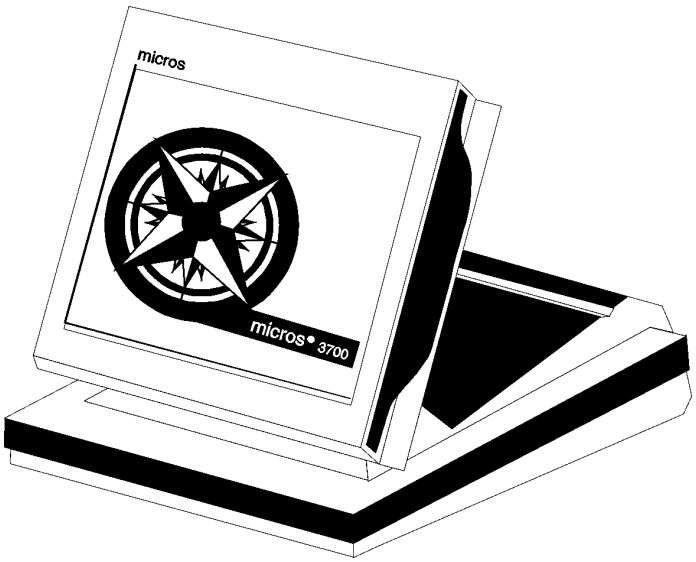
micros® Systems, Inc.

SQL / Database Access Manual

3700 POS

Restaurant Enterprise Series



Part Number:100134-507 (4th Edition)

Declarations

Warranties

Although the best efforts are made to ensure that the information in this manual is complete and correct, MICROS Systems, Inc. makes no warranty of any kind with regard to this material, including but not limited to the implied warranties of marketability and fitness for a particular purpose. Information in this manual is subject to change without notice. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information recording and retrieval systems, for any purpose other than for personal use, without the express written permission of MICROS Systems, Inc.

MICROS Systems, Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this manual.

Trademarks

CorelDraw is a registered trademark of Corel Corporation.

Crystal Reports is a registered trademark of Crystal Computer Services, Inc.

FrameMaker is a registered trademark of Adobe Corporation.

Microsoft, Microsoft Excel, Win32, Windows, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the U.S. and/or other countries.

Sybase SQL Anywhere is a registered trademark of Sybase Inc.

Visio is a registered trademark of Visio Corporation.

Windows®95 is a registered trademark of Microsoft in the U.S. and other countries.

Design and Production

This manual was written with FrameMaker 5.0 for Windows. Some illustrations and diagrams were designed in CorelDraw and/or Visio.

Printing History

New editions of this manual incorporate new and changed material since the previous edition. Minor corrections and updates may be incorporated into reprints of the current edition without changing the publication date or the edition number.

			Software Version	
Edition	**Month**	**Year**	**3700**	**Product Management**
1st	September	1996	1.00	-
2nd	April	1998	2.00	-
3rd	July	1998	2.00	1.00
4th	August	2000	2.60	2.60

Table of Contents

Chapter 9: Database Tables

Index

Preface

In this preface, you will find information about this manual. Refer to the preface if you have any questions about the organization, conventions, or contents of this manual.

In this preface

Welcome to the SQL / Database Access Manual

Welcome to the 3700 SQL Database Access Manual. This manual describes the information that is stored in the 3700 RES database and how to access this information.

How This Manual is Organized

Chapter 1: Introduction to the Database

In this first section, you will find a brief, general introduction to the 3700 database. This section provides an overview of database concepts and describes the 3700 database structure.

Chapter 2: Connecting to the Database

This section describes how to connect to a database server via ODBC and how to configure the database engine.

Chapter 3: Database Management Utilities

This section describes how ISQL and SQL Central may be used to manage an SQL Anywhere database.

Chapter 4: Accessing the Database

This section provides examples of how to access the 3700 database from the following applications:

- Crystal Reports
- Microsoft® Excel

Chapter 5: database Tables and RES

This section describes how the 3700 database tables relate to what you see in the POS Configurator module.

Chapter 6: Views

This section introduces Views and describes how they are used with the 3700 database.

Chapter 7: Stored Procedures

This section contains an overview of Stored Procedures and describes how they are used with the 3700 database.

Chapter 8: Domain Datatypes

In this section, you will find a description of the Domains and SQL data types that have been defined for the 3700 database.

Chapter 9: Database Tables

This section contains the following information for each 3700 database table:

- Logical Column Name(s)
- Physical Column Name(s)
- Datatype for each Column
- Primary Key
- Foreign Key
- Alternate Keys
- Null Option

Who Should Use This manual?

This manual is intended for use by:

- MICROS Installers/Programmers
- MICROS Dealers
- MICROS Customer Service
- MICROS Training Personnel
- MIS Personnel

This manual assumes that you have the following knowledge or expertise:

- Working knowledge of the Windows interface.
- Operational understanding of PCs.
- Understanding of basic network concepts.
- Experience with Windows; especially with Windows 95 and Windows NT.
- Basic knowledge of relational database concepts.
- Understanding of POS terminology and concepts.
- Exposure to a MICROS POS system (2400, 2700, 4700, 8700) or other POS; preferably experience programming 2700 systems.

Related Manuals

This section provides a list of the related MICROS documentation as well as a list of suggested reading materials.

MICROS Documentation

The 3700 Restaurant Enterprise Series library includes:

❑ *3700 Administration Applications Manual*	100134-503
❑ *3700 Custom Reports Design Manual*	100134-508
❑ *3700 Feature Quick Reference Manual*	100134-506
❑ *3700 Feature Reference Manual*	100134-501
❑ *3700 Hardware Installation Guide*	100134-514
❑ *3700 Hardware User's Maintenance Guide*	100134-512
❑ *PC Workstation Model 32 Setup Guide*	100016-085
❑ *3700 Reports Manual*	100134-511
❑ *3700 Site Preparation Guide*	100134-513
❑ *3700 Site Survey Manual*	100134-505
❑ *3700 SQL / Database Access Manual*	100134-507
❑ *3700 POS Configurator User's Guide*	100134-504
❑ *3700 System Interface Module User's Guide*	100134-516
❑ *3700 User's Manual*	100134-502

Suggested Reading

- ❑ Crystal Reports documentation
- ❑ Microsoft Windows 95 Resource Kit
- ❑ Sybase SQL Anywhere documentation

Conventions & Symbols

This section describes the conventions and symbols that are used throughout the printed 3700 documentation.

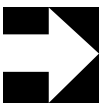
Conventions

The following conventions are used throughout this manual.

Convention	Meaning
[Key]	Keys on a PC or PC Workstation. Example: [Enter]
'Message'	Messages that may appear during the programming process. Example: 'Select Menu Item Range'
`Courier`	Information to be entered by a user. Example: `copy a:\readme.txt c:\micros`
bold	Boldface is used to indicate directory names, filenames, and menu selections. Example: **File** Menu.
[Key1]+[Key2]	The plus sign (+) between key names indicates that you should press the keys simultaneously.
|	Indicates the sequence of menu items to be selected. Example: **File | Sales | Condiments**

Symbols

The following symbols are used throughout this manual.



Note

This symbol is used to bring special attention to a related feature.

Tip

This symbol is used to point out suggestions that can save you time and difficulty.

Reader Response

As you read this, the documentation staff at MICROS is hard at work preparing the next edition of this manual. Your feedback could be instrumental in changing that next edition.

Tell us what you think— we'd like to hear from you!

We are very interested in hearing from you about:

- ❑ Good ideas -

  Tell us about some part of this manual that you think works well– we'll be sure to maintain it.

- ❑ Ideas that need work -

  Tell us about an area that needs to be improved– we'll punch it up.

- ❑ Information not included -

  Did we miss something? Let us know so we can add it.

- ❑ Information that's not clear -

  Did you find something hard to follow? We'll rethink it and rewrite it.

- ❑ Information that's not correct -

  Did something get past our arduous tech edit process? Help us fix it.

How to reach us

Postal Address

If you can offer any criticisms or suggestions about this manual, please mail a note or postcard to:

MICROS Systems, Inc.
7031 Columbia Gateway Drive
Columbia, MD 21046-2289 USA

Attn: Documentation Group Manager

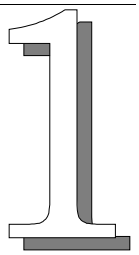
E-mail Address

As an alternative, you may address criticisms and suggestions regarding this manual by electronic mail to:

microsdoc@micros.com.

Please use the name of the manual as the subject line.

Chapter



Introduction to the Database

This chapter provides an overview of database concepts and an introduction to the 3700 Database.

In this chapter

Database Concepts

This section introduces some of the basic terms and concepts that are important in accessing the 3700 database. Note that the details of many of these concepts are beyond the scope of this manual.

What is a Database?

A database is a shared, integrated collection of data. A database can be modeled conceptually to define the way in which the database is structured. For instance, SQL Anywhere is modeled as a relational database.

Relational Database

In a relational database model, information is viewed in sets of tables. Each table contains a fixed number of columns and a variable number of rows.

Each column can contain values which are dependent on a domain (or datatype). A domain defines the set of allowable values for a particular column.

Keys

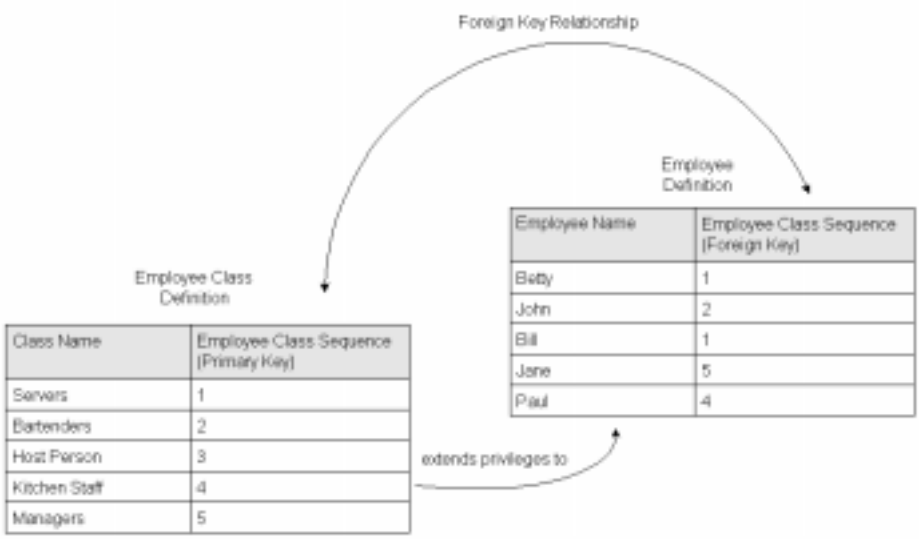
Keys are columns that help to uniquely identify specific rows in a table and define relationships between tables. There are three types of keys: primary, foreign, and alternate.

Primary Key

A primary key is any column or set of columns where the values uniquely identify a row in the table. In the 3700 database, most tables have a column for a sequence number. This sequence number is the primary key and can not be modified using the POS Configurator module.

Foreign Key

A foreign key defines the relationships between database tables. A foreign key in one table contains a value corresponding to the primary key of another table. For instance, in the 3700 database, a relationship has been defined between the Employee Class and Employee tables. The Employee table contains the Employee Class Sequence column which is a foreign key to the Employee Class Sequence column in the Employee Class table.



In the 3700 database, privileges are defined for Employee Classes and are extended to each employee who is a member of a particular Employee Class. In the diagram above, the employee Paul is a member of the Kitchen Staff class. Therefore, Paul will be extended the privileges that have been assigned to the Kitchen Staff class.

When the foreign key of a table contains the same values as the primary key of another table, there is referential integrity. Referential integrity ensures that database information remains accurate and usable. In the above example, the employee Paul is a member of the Kitchen Staff class. Since referential integrity exists between the Employee Class and Employee tables, you can not delete the Kitchen Staff class while Paul is still a member.

Alternate Key

An alternate key provides a way of uniquely identifying the rows in a table in addition to the primary key. For instance, the payroll_id column in the emp_def table is an alternate key. Since no two employees will have the same payroll ID number, the payroll_id column is designated as an alternate key and is therefore unique.

What is SQL?

Structured Query Language (SQL) is a query and programming language that is used by many relational database systems to query, update, and manage databases.

Although the exact syntax may vary from vendor to vendor, the basic concepts and structures for SQL are the same. For detailed information about using SQL commands to access a database, refer to your *Sybase SQL Anywhere User's Guide.*

What is Client/Server Computing?

Client/Server computing is a form of distributed processing in which a computer acts as a Server, providing database, application, and systems management functionality to clients. In the 3700 environment, the Server is the database engine, such as SQL Anywhere. The clients are applications, such as POS Configurator or POS Operations.

The Server handles requests for data and returns only the data required to answer the request. If the client and server are on different network nodes, unnecessary network traffic is eliminated because only the requested data is transferred.

In the networked client/server environment, the workload is split between the client computers that request services, such as printing, information retrieval, or updating of a customer account, and the server computer that processes the request.

Two of the benefits of client/server computing are described below:

Improved Performance

Client/Server processing is based on the use of a database server, which significantly reduces LAN traffic and improves the overall system performance.

Centralized Data Administration

All of the databases are managed and accessed by a central database server. The database server is responsible for various functions, such as data integrity, security, and backup.

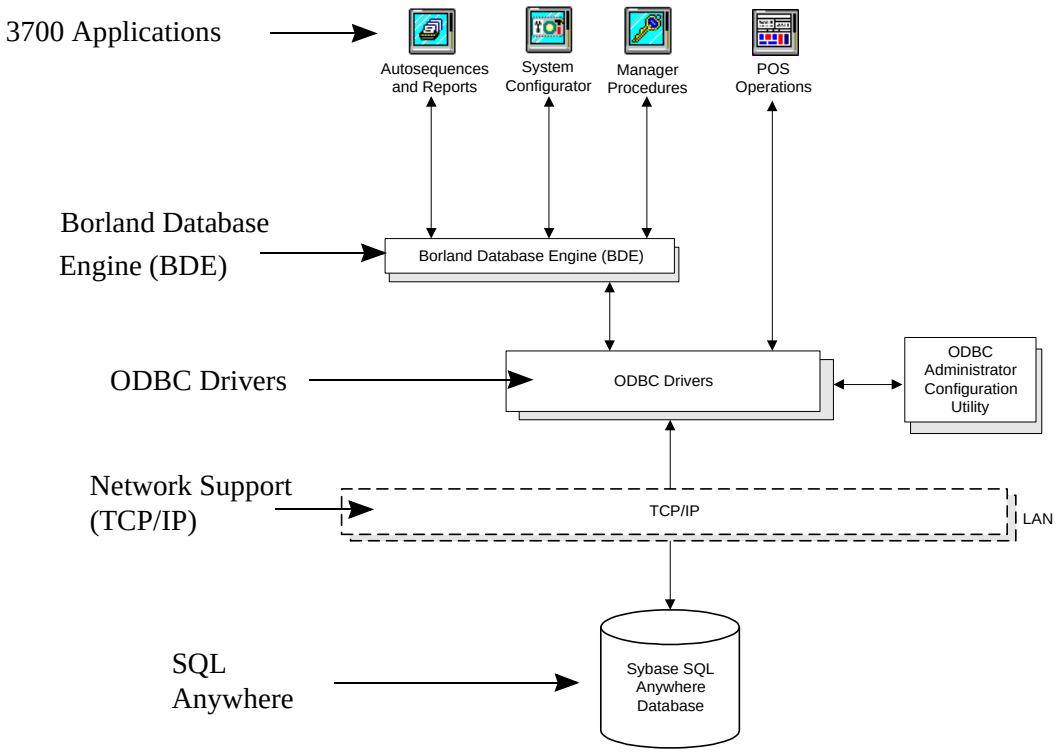
Database System Overview

This section describes the 3700 database system components and introduces the types of database tables.

3700 Database System Components

The 3700 database system consists of the following components:

- SQL Anywhere
- Network Support (TCP/IP)
- Open Database Connectivity (ODBC) Drivers
- Borland Database Engine (BDE)
- 3700 Applications



SQL Anywhere

SQL Anywhere is an SQL-based, relational database server. SQL Anywhere can be used in a client/server environment where processing is divided between the client and server components.

Network Support (TCP/IP)

TCP/IP is an interface that allows two different processes on different computers to communicate.

Open Database Connectivity (ODBC)

ODBC is an interface standard developed by Microsoft Corporation. ODBC provides an interface to database management systems through drivers provided by the database manufacturers. For instance, Sybase supplies an ODBC driver for SQL Anywhere.

Borland Database Engine (BDE)

The Borland Database Engine processes the requests that are made by certain client applications. These requests are in the form of SQL statements, such as a SELECT statement which is used to extract information from the database. The client application sends the SQL statement, the database engine processes them and sends the results back to the client application.

3700 Applications

The 3700 applications include any applications that access the 3700 database, such as POS Configurator.

Types of Database Tables

The tables in the 3700 database fall into one of the following categories: definition, status, transaction, or totals tables.

Definition Tables (*_def)

The definition tables define various aspects of the 3700 system, such as menu items, employees, devices, etc. These definition tables can be programmed using the POS Configurator module.

Status Tables (*_status)

The status tables are updated automatically in response to system events. For instance, the Employee Status table contains a column (Time Clock Status) that keeps track of when an employee clocks in or out. This table is updated whenever an employee clocks in or out.

Detail (*_dtl)

The detail tables contain information about POS transactions. For instance, as POS transactions occur, the type of transaction is written to the Transaction (trans_dtl) table.

Totals (*_ttl)

The totals tables contain accumulated totals information that reflect transaction activity, such as menu item sales and time period sales.

Chapter 2

Connecting to the Database

This chapter describes how to manually configure the connection needed to access the 3700 database.

In this chapter

Connection Overview

In order for client applications, such as POS Configurator or Autosequences and Reports to access the 3700 database, an ODBC connection must be established to the database and the Borland Database Engine must be configured.

ODBC
ODBC is an interface developed by Microsoft Corporation which provides a standard interface for connecting client applications to database management systems. ODBC defines data sources. Each data source description contains the parameters required for connecting to the database. This data source can be created using the ODBC Administrator.

The ODBC interface is automatically configured during installation of the 3700 system CD-ROM. Refer to "ODBC Administrator" on page 2-3 for information on how to manually configure the ODBC data source.

BDE
The Borland Database Engine processes the requests that are made by the client application. These requests are in the form of SQL statements, such as a SELECT statement which is used to extract information from the database. The client application sends the SQL statement, the database engine processes them and sends the results back to the client application.

The BDE is automatically configured during installation of the 3700 system CD-ROM.

ODBC Administrator

The ODBC Administrator allows you to define the data source required to connect to the 3700 database. The following steps describe how to configure the data source for 3700. Note that you must have SQL Anywhere installed in order to configure the data source.

1. From the Start Menu, select **Programs | Sybase SQL Anywhere 5.0 | ODBC Administrator.** The Data Sources dialog box appears.
2. Click the Add button. The Add Data Source dialog box appears.
3. Highlight Sybase SQL Anywhere 5.0 in the Installed ODBC Drivers box and click OK. The SQL Anywhere ODBC Configuration dialog box appears.
4. Enter the appropriate information as shown and described below:

For this field:	Enter:
Data Source Name	micros
User ID	
Password	
Server Name	The name assigned to the Server PC.
Database Name	micros
Database File	The path and filename for the micros.db database file.

5. Click the Network button in the Sybase SQL ODBC Configuration dialog box.
6. Click the Microsoft Applications (Keys in SQLStatistics) check box.
7. Click OK on the Sybase SQL ODBC Configuration dialog box.
8. Close the Data Sources dialog box.

Chapter 3

Database Management Utilities

This chapter describes how to use SQL Anywhere's ISQL and SQL Central utilities for database management.

In this chapter

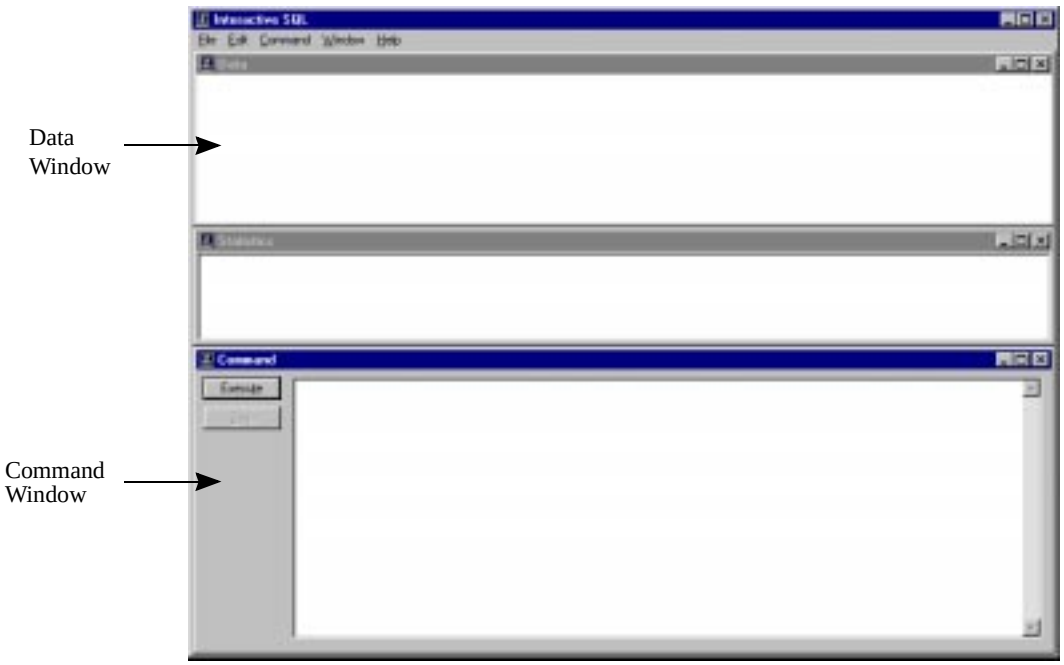
ISQL

What is ISQL?

Interactive Structured Query Language (ISQL) is the utility provided with SQL Anywhere that allows you to execute SQL statements and view the results.

Starting SQL Anywhere ISQL

To start SQL Anywhere ISQL utility, select **Programs | Sybase SQL Anywhere 5.0 | ISQL** from the Start menu. Enter the appropriate User ID and Password. The following window displays:



Enter SQL statements in the Command Window that appears at the bottom of the ISQL screen. The Data Window at the top of the ISQL screen displays the query results.

Displaying Data Using ISQL

ISQL can be used to view information in the 3700 database. For instance, to display the information that is currently in the Menu Item Definition Table, enter the following line into the Command window and press the **Execute** button.

```
select * from mi_def
```

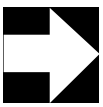
This command displays all columns (*) in the Menu Item Definition (mi_def) table.

You can also use the select command to display specific columns from a table or group of tables as shown in the following select statement.

```
select name_1, preset_amt_1
from mi_def, mi_price_def
where mi_price_def.mi_seq = mi_def.mi_seq
```

The select statement shown above selects the name_1 column from the mi_def table and the preset_amt_1 column from the mi_price_def table based on the search-condition that the mi_seq numbers are the same in both tables.

Refer to the *Sybase SQL User's Guide* for detailed information on using the select command.



Note

If ISQL or any other tool places an update lock on a row without committing or rolling back, other applications including POS Operations may stop when the locked row is encountered. Refer to the Using Transactions and Locks section of the *Sybase SQL User's Guide* for more information.

SQL Central

What is SQL Central?

SQL Central is a graphical database management tool that allows you to view the database structure and monitor database performance.

Starting SQL Central

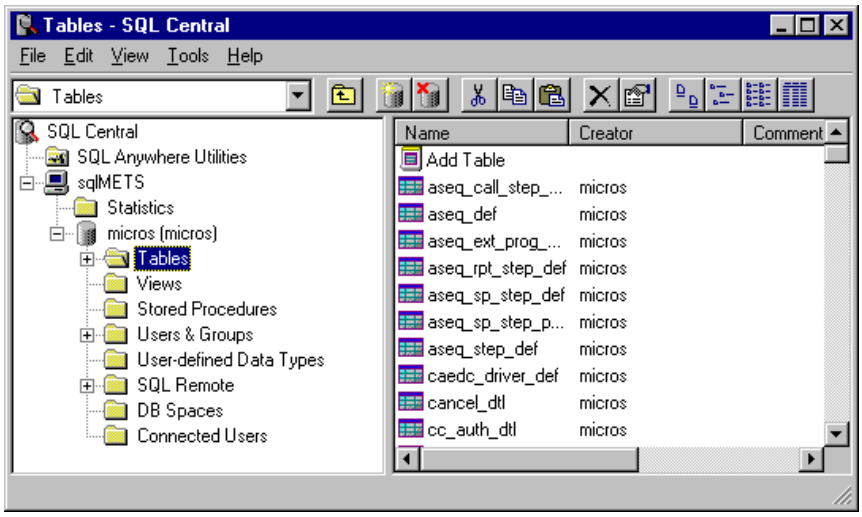
To start the SQL Central utility:

1. Select **Programs | Sybase SQL Anywhere 5.0 | SQL Central** from the Start menu. The SQL Central window displays.
2. Select **Tools | Connect.** The SQL Anywhere Logon dialog box displays.
3. Enter the appropriate user ID and password. The database appears in the SQL Central window.

Viewing the Database Structure

To examine the database structure:

1. Expand the database server that appears on the left side of the SQL Central window.
2. Expand the 3700 database folder as shown below:



3. Double-click the Tables folder. The database tables appear on the right side of the SQL Central window.
4. Double-click the chk_dtl table that appears on the right side of the SQL Central window.

5. Double-click the Columns folder. The columns in the chk_dtl table appear.

Note that the views and stored procedures in the database can be viewed in the same manner as the tables.

Refer to the *Sybase SQL User's Guide* for more information on using SQL Central.

Chapter 4

Accessing the Database

This chapter provides examples of how to access the 3700 database using Crystal Reports and Microsoft® Excel.

In this chapter

Crystal Reports

Crystal Reports is an application by Crystal Computer Services, Inc. that allows you to create custom reports using information stored in the 3700 database. Crystal Reports Professional is required to create custom reports.

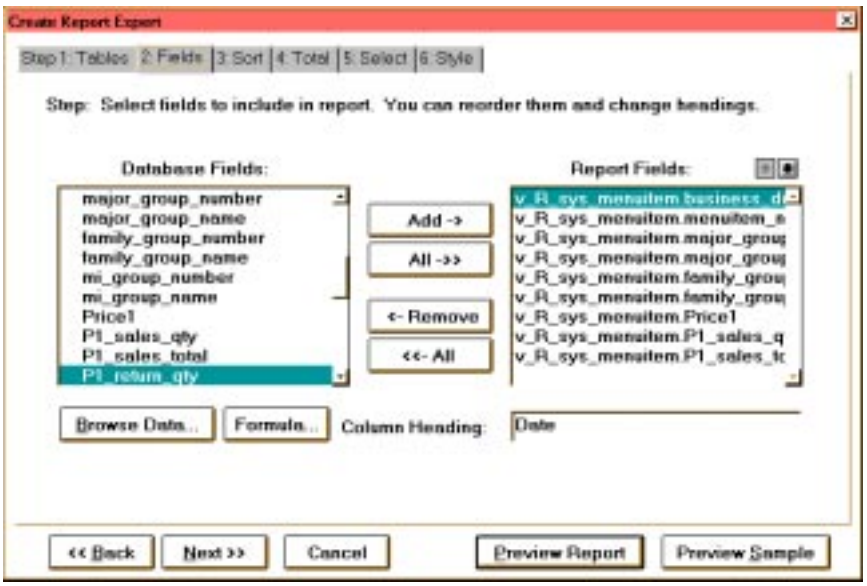
MICROS will set all standard reports at object number 9999 and below. Any custom reports should be given an object number above 9999. This will ensure that MICROS can add changes to reports without overwriting a custom report

The following steps provide an example of how to create a custom report. Refer to the *3700 Custom Reports Design Manual* for more details on how to create custom reports.

1. Select **File | New | Standard Expert** from Crystal Reports. The Create Report Expert appears.
2. Click the SQL/ODBC button. The Choose SQL Table dialog box appears.
3. Highlight the database view or table that you would like to use in creating the report and click the Add button.

   Example: v_R_sys_menuitem
4. Repeat Step 3 for each view or table you would like to use in reporting.
5. Click the Done button from the Choose SQL Table dialog box. The Step 1:Tables tab appears.
6. Click the Next button. The Step 2:Fields tab appears.

7. Highlight the database field and click the Add button for each field that you would like to appear in the report. The selected fields appear in the Report Fields box as shown below:



   Example: business_date, menuitem_number, menuitem_name1, major_group_number, major_group_name, family_group_number, family_group_name, Price1, P1_sales_qty, P1_sales_total

8. Click the Next button. The Step 3:Sort tab appears.

9. Highlight the field(s) that you would like to sort by and click the Add button.

   Example: business_date

10. Select the appropriate sort order from the Order drop-box and click the Next button. The Step 4:Total tab appears.

    Example: in ascending order

11. Select the fields to subtotal, count, etc. and click the Next button. The Step 5:Select tab appears.

    Example: P1_sales_qty (sum), P1_sales_total (sum)

12. Select the fields and filter options as desired and click the Next button. The Step 6:Style tab appears.

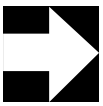
13. Enter the report title, select the desired report style, and click the Preview Report button.

    Example: Report Example (Title), Leading Break (Style)

14. The report template is created, the report is generated and displayed as shown below.

Report Example
1/17/96

Date	Menu Item ID	Name	Major Group ID	Name	Family Group ID	Name	Price	Qty	Total
12/6/95									
	2	Quesadilla	1	Food	1	Appetizers	4.50	1	4.50
	4	Chix Ques	1	Food	1	Appetizers	5.75	2	11.50
	23	Cobb Salad	1	Food	2	Salads	6.95	1	6.95
	24	Mixed Green Sal	1	Food	2	Salads	4.75	1	4.75
	61	Pasta w/Tomato	1	Food	4	Pastas	6.75	2	13.50
	62	Pasta w/Meat	1	Food	4	Pastas	7.00	2	14.00
	63	Pasta w/Alfredo	1	Food	4	Pastas	7.00	2	14.00
	64	Chicken Alfredo	1	Food	4	Pastas	7.00	3	21.00
	65	Lasagna	1	Food	4	Pastas	8.95	4	35.80
	66	Tort. Alfredo	1	Food	4	Pastas	8.65	2	17.30
	81	Shrimp Scampi	1	Food	5	Entrees	9.85	1	9.85
	85	NY Strip Steak	1	Food	5	Entrees	12.95	1	12.95
	86	Prime Rib	1	Food	5	Entrees	15.95	1	15.95
	101	Turkey Club	1	Food	6	Sandwiches	5.95	1	5.95
	103	BLT	1	Food	6	Sandwiches	5.95	1	5.95
	106	Cheese Burger	1	Food	6	Sandwiches	6.55	4	26.20
	208	Iced Tea	1	Food	11	Beverages	1.25	2	2.50
	209	Lemonade	1	Food	11	Beverages	1.25	4	5.00
	216	Coke	1	Food	11	Beverages	1.00	8	0.00
	503	Med.	1	Food	8	Condiments		1	0.00
	504	Medium Well	1	Food	8	Condiments		2	0.00
	513	Rice	1	Food	8	Condiments		1	0.00
	514	Baked Potato	1	Food	8	Condiments		1	0.00
	516	French Fries	1	Food	8	Condiments		1	0.00
	518	Asparagus	1	Food	8	Condiments		1	0.00
	519	Green Beans	1	Food	8	Condiments		2	0.00
	522	Rasp. Vin	1	Food	8	Condiments		2	0.00
	523	Ranch	1	Food	8	Condiments		1	0.00
	524	French	1	Food	8	Condiments		1	0.00
	528	Wheat	1	Food	8	Condiments		1	0.00
	529	Rye	1	Food	8	Condiments		1	0.00
	533	Spaghetti	1	Food	8	Condiments		4	0.00
	534	Linguini	1	Food	8	Condiments		3	0.00
	536	Penne	1	Food	8	Condiments		2	0.00
	526	Thous Island	1	Food	8	Condiments		1	0.00
	1204	Sam Adams	2	Liquor	17	Draft Beer	2.25	2	4.50
	46	Cup of Crab	1	Food	3	Soups	2.75	8	0.00
	1206	Miller Lite	2	Liquor	17	Draft Beer	2.25	2	4.50
								64	236.75



Note

Refer to the *3700 Custom Reports Design Manual* for more information on creating custom reports.

Microsoft Excel

The information stored in the 3700 database can be accessed from Microsoft® Excel by using the XLODBC and Microsoft Query add-in programs. These add-in programs allow you to use Microsoft Query to select the tables and columns that you wish to bring into Excel. Once the tables and columns are selected, the information can be returned to the desired cells in Excel.

The following steps describe how to use the Excel add-in programs to bring information into Microsoft Excel from the 3700 database. Note that you must have the XLODBC and Microsoft Query add-ins loaded to complete this procedure.

1. Select **Data | Get External Data** from Microsoft Excel. The Microsoft Query application is started and the Select Data Source dialog box appears.
2. Highlight the 3700 data source and click the Use button. The Add Tables dialog box appears.
3. Highlight the database table that you would like to access and click the Add button. The selected database table and its columns appears at the top of the Microsoft Query screen.
4. Repeat Step 3 for each table you would like to access.
5. Click the Close button from the Add Tables dialog box.
6. The selected tables appear as shown below:

7. Select the desired columns from the tables that appear at the top of the Microsoft Query screen. The contents of the columns appear at the bottom of the screen as shown below:

	name_1	preset_amt_1	price_1_sls_cnt	price_1_sls_ttl
	Chix Ques	5.75	1	5.75
	Cobb Salad	6.95	1	6.95
	Mixed Green Sal.	4.75	1	4.75
	Pasta w/Tomato	6.75	2	13.50
	Pasta w/Meat	7.00	1	7.00
	Pasta w/Alfredo	7.00	1	7.00
	Chicken Alfredo	7.00	2	14.00
	Lasagna	8.95	3	26.85
	Tort. Alfredo	8.65	2	17.30
	NY Strip Steak	12.95	1	12.95
	Prime Rib	15.95	1	15.95
	Iced Tea	1.25	2	2.50
	Lemonade	1.25	1	1.25
	Sam Adams	2.25	2	4.50
	Miller Lite	2.25	2	4.50
	Coke	1.00	0	0.00
	Cup of Crab	2.75	0	0.00

8. Select **File | Return Data to Microsoft Excel.** The Excel screen appears with the Get External Data dialog box.

9. Enter the appropriate cell location in the Destination field in which you would like the selected information to appear. Click OK. The selected database information appears in the Excel spreadsheet.

Chapter 5

Database Tables and RES

This chapter introduces the Restaurant Enterprise System and describes how changes made through POS Configurator, Enterprise Configurator, and Enterprise Office Main Monitor affect the contents of the database tables.

In this chapter

Overview

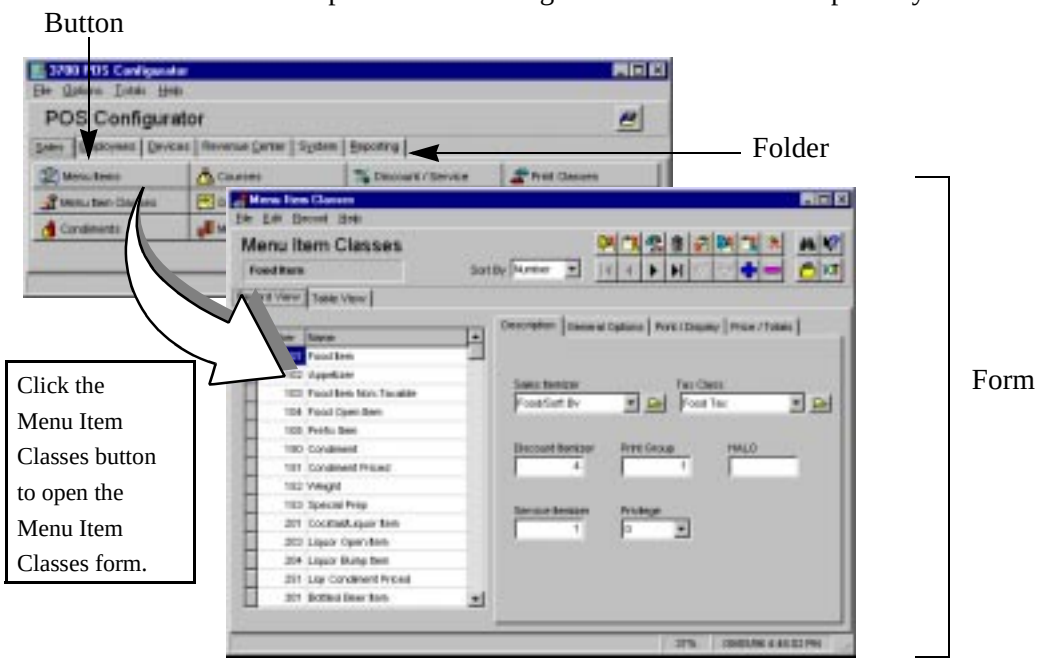
POS Configurator, Enterprise Configurator, and Enterprise Office Main Monitors are software applications which allow you to easily program the database by adding, updating, and deleting information stored in the database. For instance, with POS Configurator you can program hardware configurations, employee information, menu items, and other POS related information. Enterprise Configurator allows you to define store locations, price tiers, and enterprise level information. Enterprise Office allows you to keep track of inventory, recipes, and order information.

Refer to the Online Help for POS Configurator or the POS Configurator User's Guide for detailed information about the POS Configurator module.

POS Configurator Folders

The POS Configurator module includes the following folders: Sales, Employees, Devices, Revenue Center, System, and Reporting. Each of these folders contain buttons that allow you to access various database tables. Clicking a button on a folder displays the corresponding form.

The following diagram points out the terminology used throughout this chapter when referring to the Restaurant Enterprise System modules.



The following sections contain tables which describe each of the POS Configurator folders and how they relate to what is stored in the database. The following is an example of the tables that appear in the following sections.

The POS Configurator form	and the POS Configurator tab	updates the database table
Menu Item Classes	Description	Menu Item Type Class Definition (mi_type_class_def)

The table shown in the above example should be read as follows:

The POS Configurator button *Menu Item Classes* and the POS Configurator tab *Description* updates the database table *Menu Item Type Class Definition.*

Note that the Logical (Menu Item Type Class Definition) and Physical (mi_type_class_def) names are listed for the database table being updated.

Sales Folder

The Sales folder allows you to update the sales-related tables in the database. The POS Configurator Sales folder is shown below:

The following table shows how the POS Configurator Sales folder corresponds to the database.

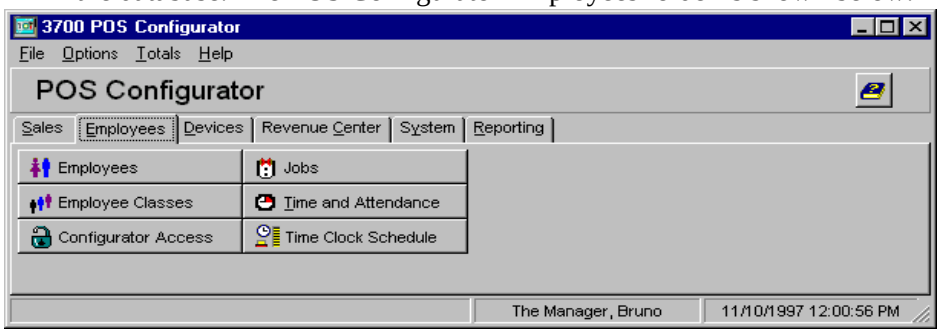
The POS Configurator form	and the POS Configurator tab	updates the database Table
Menu Items	Definition	
	General	Menu Item Definition (mi_def)
	Groups	Menu Item Group Definition (mi_grp_def)
	Options	Menu Item Options (mi_def)
	Reporting	Report Groups (mi_def)
	Prices	Menu Item Price Definition (mi_price_def)
Menu Item Classes	Description General Options Print/Display Price/Totals	Menu Item Type Class Definition (mi_type_class_def)
Condiments	Condiment Groups	Condiment Group Definition (cond_grp_def)
	Condiment Membership	Condiment Group Membership Class Definition (cond_grp_mem_def)
	Condiment Selection	Condiment Selection Class Definition (cond_grp_sel_def)

The POS Configurator form	and the POS Configurator tab	updates the database Table
Courses	Courses	Course Definition (crs_def)
	Course Membership	Course Membership Class Definition (crs_mem_def)
	Course Selection	Course Selection Class Definition (crs_sel_def)
Descriptors	Discount/Service SLU	Discount/Service Charge SLU Definition (dsvc_slu_def)
	Menu Item SLU	Menu Item SLU Definition (mi_slu_def)
	Tender/Media SLU	Tender/Media SLU (tmed_slu_def)
	Sales Itemizers	Sales Itemizer Definition (sales_itmzr_def)
	Headers	Header Definition (hdr_def)
	Trailers	Trailer Definition (trlr_def)
Menu Levels	Menu Level Classes	Menu Level Class Definition (mlvl_class_def)
	Auto Menu Levels	Auto Menu Level Definition (auto_mlvl_def)
Discount/Service	General Options Discount/Service Charge	Discount/Service Charge Definition (dsvc_def)
Tender/Media	General Tender CC Tender Credit Auth | Authorization PMS Service TTL Printing	Tender Media Definition (tmed_def)
	Presets	Tender Preset Amount (preset_tmed_def)
	Credit Auth | Preambles	Tender Media Preamble Definition (tmed_preamble_def)
Currency		Foreign Currency Definition (currency_def)
Print Classes		Print Definition Class (prn_class_def)

The POS Configurator form	and the POS Configurator tab	updates the database Table
Tax Rates	General	Tax Rate Definition (tax_rate_def)
	Breakpoints	Tax Rate Definition (tax_rate_def) Tax Breakpoint Definition (tax_brkpt_def)
	Canadian Tax	Tax Rate Definition (tax_rate_def)
Tax Classes		Tax Class Definition (tax_class_def)

Employees

The Employees folder allows you to update the employee-related tables in the database. The POS Configurator Employees folder is shown below:



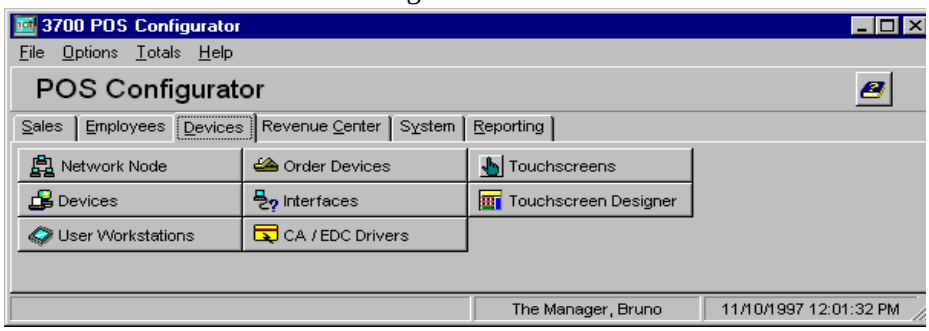
The following table shows how the POS Configurator Employees folder corresponds to the database.

The POS Configurator form	and the System Configurator tab	updates the database Table
Employees	General Options	Employee Definition (emp_def)
	Employee Status	Employee Status (emp_status)
	Job Rates	Employee Job Rate Definition (job_rate_def)
Employee Classes	Guest Checks Void/Return Options Procedures Clock In/Sign In Privileges Transactions Printing	Employee Class Definition (emp_class_def)

The POS Configurator form	and the System Configurator tab	updates the database Table
Configurator Access		Employee Class Configurator Access Definition (emp_class_cfg_access_def)
Jobs	Job Definitions	Employee Job Definition (job_def)
	Job Categories	Employee Job Category Definition (job_cat_def)
Time and Attendance	General Time Clock Schedule Overtime Breaks	Employee Time and Attendance Definition (time_clock_def)
Time Clock Schedule		Employee Time Clock Schedule Definition (time_clock_sched_def)

Devices

The Devices folder allows you to update the hardware-related tables in the database. The POS Configurator Devices folder is shown below:



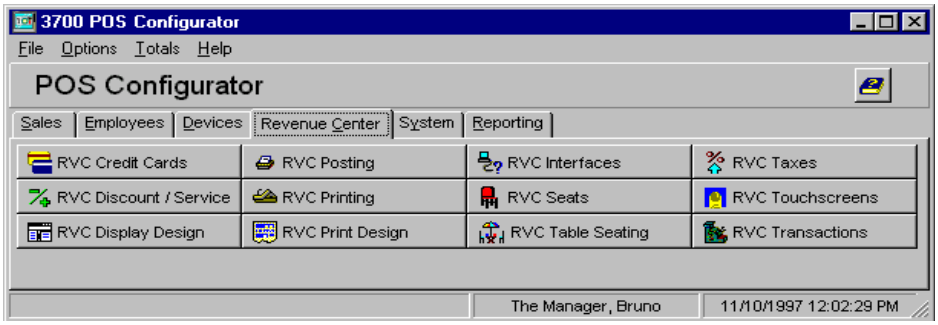
The following table shows how the POS Configurator Devices folder corresponds to the database.

The POS Configurator form	and the POS Configurator tab	updates the database Table
Network Node	Network	Lan Node Definition (lan_node_def)
	Comm Ports	Communications Port Definition (com_port_def)
Devices	General Printer Definition Printer Interface	Device Table Definition (dev_def)

The POS Configurator form	and the POS Configurator tab	updates the database Table
User Workstation	General UWS Options Cashier Order Devices Printers	User Workstation Definition (uws_def)
Order Devices	General Options	Order Device Definition (order_device_def)
Interfaces	General	Interface Definition (interface_def)
	Interface	Com Port Definition (com_port_def)
CA/EDC Drivers	Driver System Authorization Settlement	CA/EDC Driver Definition (caedc_driver_def)
Touchscreens	Touchscreens	Touchscreen Screen Definition (ts_scrn_def)
	Styles	Touchscreen Style Definition (ts_style_def)
Touchscreen Designer		Touchscreen Key Definition (ts_key_def)

Revenue Center

The Revenue Center folder allows you to update the Revenue Center characteristics of the database. The POS Configurator Revenue Center folder is shown below:



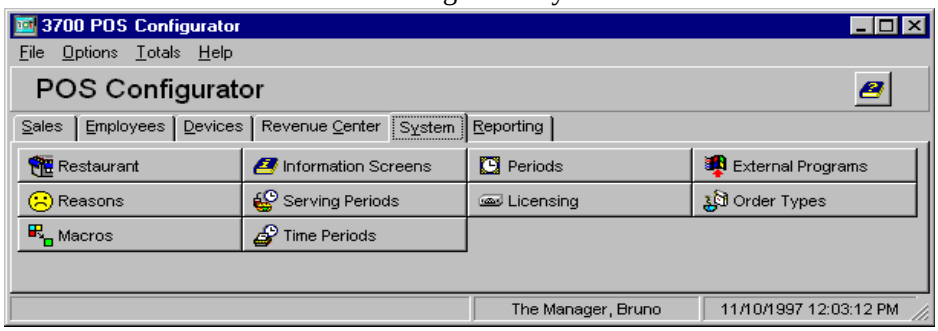
The following table shows how the POS Configurator Revenue Center folder corresponds to the database:

The POS Configurator form	and the POS Configurator tab	updates the database Table
RVC Credit Cards	General Headers Trailers Floor Limits Printing	Revenue Center Definition (rvc_def)
RVC Discount/ Service		Revenue Center Definition (rvc_def)
RVC Display Design		Revenue Center Definition (rvc_def)
RVC Posting	General Options	Revenue Center Definition (rvc_def)
RVC Printing	General Options	Revenue Center Definition (rvc_def)
RVC Print Design	General Parameters Guest Checks Options Headers Trailers	Revenue Center Definition (rvc_def)
RVC Interfaces	System Interface Module Property Management System	Revenue Center Definition (rvc_def)
RVC Seats		Revenue Center Definition (rvc_def)
RVC Table Seating		Table Definition (tbl_def)

The POS Configurator form	and the POS Configurator tab	updates the database Table
RVC Taxes	General Value Added Tax Canadian Tax Other Tax	Revenue Center Definition (rvc_def)
RVC Touchscreens	Touchscreens Styles Alerts	Revenue Center Definition (rvc_def)
RVC Transactions	General Checks/Receipts Cashier Security Menu Levels Retail	Revenue Center Definition (rvc_def)

System

The System folder allows you to update the system-wide characteristics of the database. The POS Configurator System folder is shown below:



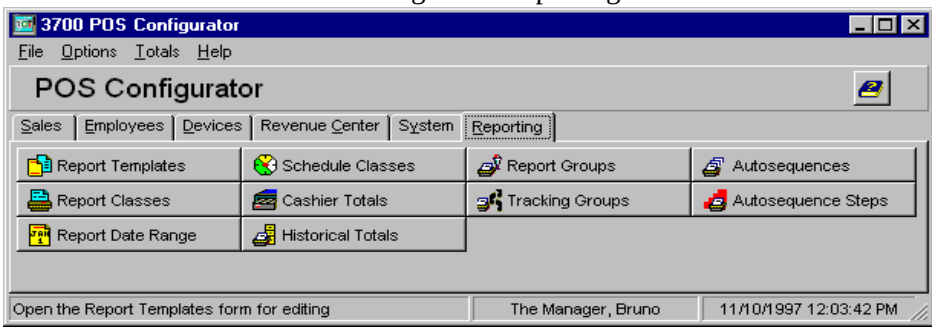
The following table shows how the POS Configurator System folder corresponds to the database:

The POS Configurator form	and the POS Configurator tab	updates the database Table
Restaurant	Descriptions Business Settings Options	Restaurant Definition (rest_def)
Reasons		Reason Definition (reason_def)
Macros		Macro Definition (macro_def) Macro Key Definition (macro_key_def)

The POS Configurator form	and the POS Configurator tab	updates the database Table
Information Screens		Information Screen Definition (info_screen_def)
Serving Periods		Serving Period Definition (srv_period_def)
Time Periods		Time Period Definition (time_period_def)
Periods		Period Definition (period_def)
Licensing		Licensing Definition (registry)
External Programs		External Programs Definition (ext_prog_def)
Order Types		Order Type Definition (order_type_def)

Reporting

The Reporting folder allows you to update the report-related features of the database. The POS Configurator Reporting folder is shown below:



The following table shows how the POS Configurator Reporting folder corresponds to the database:

The POS Configurator form	and the POS Configurator tab	updates the database Table
Report Templates		Report Templates Definition (rpt_template_def)
Report Classes		Report Classes Definition (rpt_class_def)
Report Date Range		Date Range Definition (date_range_def)
Schedule Classes	General Days/Months	Report Schedule Definition (sched_class_def)
Cashier Totals		Cashier Totals Definition (cshr_ttl_def)
Historical Totals	Descriptor	Historical Totals Definition (histo_ttl_def)
	Classes	Historical Totals Class Definition (histo_ttl_class_def)
Report Groups	Major Groups	Major Group Definition (maj_grp_def)
	Family Groups	Family Group Definition (fam_grp_def)
	Menu Item Groups	Menu Item Group Definition (mi_grp_def)

The POS Configurator form	and the POS Configurator tab	updates the database Table
Tracking Groups	Tracking Groups	Tracking Group Definition (trk_grp_def) Tracking Totals Definition (trk_ttl_def)
Autosequences		Autosequences Definition (aseq_def)
Autosequence Steps	Ranges	Autosequence Steps Definition (aseq_step_def)
	Error Actions	Autosequence Steps Definition (aseq_step_def)
	Call/Ext Prog	Autosequence External Programs Step Definition (aseq_ext_prog_step_def)
	Report	Autosequence Report Step Definition (aseq_rpt_step_def)
	Stored Proc	Autosequence Stored Procedure Step Definition (aseq_stored_proc_step_def) Autosequence Stored Procedure Step Parameter Definition (aseq_stored_proc_step_parm_def)

Adding Changes to the MICROS Database

Custom User

In version 1.02 and later, a custom user is available for making custom changes to the database. This user is a member of the DEFINITION_GROUP, and has the same privileges as applications such as POS Configurator.

The custom user (password: “custom”) has resource permission. This allows it to create its own database objects: tables, stored procedures, views and triggers for custom tables. In addition to resource permission, this user has permission to read all tables and to insert, delete, and update all definition tables. For more detailed information about permissions for this user, refer to SQL Central.

Restrictions

Be aware of these restrictions when making custom changes:

- No changes can be made to MICROS-owned objects. The custom user is only able to modify its own objects.
- Foreign keys cannot be added that reference MICROS- owned tables. Any relationship to a MICROS table needs to be implied. That relationship cannot be formed with a foreign key. Data integrity for inserts and updates to a custom table can be handled via triggers on that table.
- Conversion is not automatic. Special conversion scripts must be created (as documented below).

Custom Stored Procedures

Custom stored procedures can be included in autosequences.

All rows with object numbers above 10,000 are available for custom use in the stored_proc_def table. By inserting a row in this table for the custom stored procedure, that procedure can be setup to run in an autosequence. When adding rows to this table, the name of the procedure should be qualified with the owner name (i.e., custom.sp_ReserveRoom()).

Converting Custom Changes

To allow custom tables to be converted, additional support has been added to dbconverter. Each time it runs, dbconverter checks for the existence of preconv.bat and postconv.bat files in the 3700\scripts directory. These files will be automatically called if they exist. Although primitive, this allows sql scripts to be automatically run through ISQL.

The following example illustrates how to leverage this functionality. It is an example of a simple custom configuration that involves conference room management. This example illustrates how to set up a system to allow for custom changes to be automatically converted.

In this example, assume that the MICROS database contains the following custom database objects:

- Tables: conference_room_def, customer_account_def, current_bookings
- Stored Procedures: sp_ReserveRoom()

Converting this information involves the following steps:

- extracting information from the old database
- creating the custom objects in the new database
- loading data into the custom objects

Extracting Information

Create a sql file script to unload the data from the old database. Place the script in a directory such as 3700\sql\custom. For the purposes of this example, the sql script will be called preconv.sql.

Below is an example of a preconv.sql script used to unload data:

```
SELECT room_num, name, capacity
   FROM custom.conference_room_def;
OUTPUT TO conference_room_def.dat;

SELECT acct_num, group_name, contact_lname,
contact_fname, phone_number
   FROM custom.customer_account_def;
OUTPUT TO customer_account_def.dat;

SELECT room_num, event_date, start_time, acct_grp,
guest_cnt, setup_type, confirmation_num
  FROM custom.current_bookings;
OUTPUT TO current_bookings.dat;
```

Create a batch file in the 3700\scripts directory called preconv.bat. This batch file will be used the run the preconv.sql script. The contents of the batch file should resemble the following:

```
%SQLANY%\win32\isql -c "uid=custom;pwd=custom;dbn=mi-
crosold;dbf=%RESDBDIR%\microsold.db;
 start=dbeng50" %RESDBDIR%\custom\preconv.sql
```

Place this batch file in the 3700\scripts directory. This will automatically be run by the dbconverter.

Note that the %SQLANY% and %RESDBDIR% environment variables are created during MICROS setup.

Creating Objects and Loading Data into the New Database

The next step in setting up a custom conversion is creating a sql script to create custom objects and load data into the new database. For the purposes of this example, the script will be called postconv.sql.

Create the postconv.sql script to recreate the custom database objects in the new database.

```
%% Create custom tables and stored procedures.

CREATE TABLE custom.conference_room_def
   (room_num int,
   name char(32),
   max_capacity int,
   PRIMARY KEY(room_num));

CREATE TABLE custom.customer_account_def
   (acct_num int,
   group_name char(32),
   contact_lname char(32),
   contact_fname char(32),
   phone_number char(32),
   PRIMARY KEY(acct_num));

CREATE TABLE custom.current_bookings
    (room_num int,
    event_date date,
    start_time time,
    acct_grp int,
    guest_cnt int,
    setup_type  int,
    confirmation_num int,
    PRIMARY KEY(room_num, event_date, start_time));

CREATE PROCEDURE custom.sp_ReserveRoom
  (in p_room_num int,
   in p_acct_grp int,
   in p_date date,
   in p_time time,
   in p_num_guests int,
   inout p_confirm_num int,
   inout p_error int)
```

```
BEGIN

DECLARE @max_capacity int;

IF EXISTS (SELECT * FROM custom.current_bookings
     WHERE room_num = p_room_num
        AND event_date = p_date
        AND start_time = p_time) THEN

  %% Room is already booked!!!
  SET p_error = -1;
  RETURN;
END IF;

SELECT max_capacity
  INTO @max_capacity
  FROM custom.conference_room_def;

IF p_num_guests > @max_capacity THEN
  %% Number of guests exceeds room capacity!!!
  SET p_error = -2;
  RETURN;
END IF;

INSERT INTO custom.current_bookings
  (room_num, event_date, start_time, acct_grp,
guest_cnt,
   confirmation_num)
VALUES (p_room_num, p_date, p_time, p_acct_grp,
p_num_guests,
   room_num || minutes(now(*)));

END;

%% Reload data before adding foreign keys between custom
tables.
%% This will eliminate the need to determine a load or-
der for
%% maintaining referential integrity.

INPUT INTO custom.conference_room_def
  FROM conference_room_def.dat
(room_num, name, capacity);

INPUT INTO custom.customer_account_def
  FROM customer_account_def.dat
(acct_num,  group_name,  contact_lname,  contact_fname,
phone_number);

 INPUT INTO custom.current_bookings
  FROM current_bookings.dat
(room_num, event_date, start_time, acct_grp, guest_cnt,
setup_type, confirmation_num);

%% Restore any foreign keys between custom tables.
```

```
ALTER TABLE custom.current_bookings
  ADD FOREIGN KEY CFK1current_booking(acct_grp)
   REFERENCES custom.customer_account_def(acct_num)
   ON DELETE RESTRICT;

ALTER TABLE custom.current_bookings
  ADD FOREIGN KEY CFK2current_booking(room_num)
   REFERENCES custom.conference_room_def(room_num)
   ON DELETE RESTRICT;
```

Create a batch file in the 3700\scripts directory called postconv.bat. The contents of the batch file should resemble this:

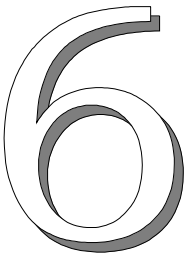
```
%SQLANY%\win32\isql -c "uid=custom;pwd=custom;dbn=mi-
cros;dbf=%RESDBDIR%\micros.db;
 start=dbeng50" %RESDBDIR%\custom\postconv.sql
```

The final directory structure should look like this:

```
C:\RES\3700\SQL\CUSTOM\preconv.sql
C:\RES\3700\SQL\CUSTOM\postconv.sql
C:\RES\3700\SCRIPTS\preconv.bat
C:\RES\3700\SCRIPTS\postconv.bat
```

Once these scripts are created, they are automatically detected on all subsequent conversions.

Chapter



Views

This chapter provides an introduction to views and describes how they are used in the 3700 environment.

In this chapter

What are Views?

A view is a logical table composed of columns from one or more database tables. Views behave in the same manner as tables, but do not really contain any data.

Views allow you to:

- Combine information from multiple tables to facilitate access.
- Limit access to data in a table.
- Define common queries that are central to the database and improve the speed of data access.

The information in a view is not stored separately in the database. Instead, each time you refer to the view, SQL Anywhere executes the associated SELECT statement to retrieve the appropriate data.

Using Views

A view is created using the CREATE VIEW command. This CREATE VIEW command allows you to define a table, known as a view, and store the results of a SELECT statement in the database under a view name.

The following is an example 3700 database view.

```
create view v_R_sys_maj_grp(business_date,
  group_number,
  group_name,
  group_category,
  sales_qty,
  sales_total,
  return_qty,
  return_total,
  discount_total, sys_sbtl_discount_total)
  as select ttl.business_date,
    ttl.maj_grp_seq,
    def.name,
    def.cat,
    ttl.sls_cnt,
    ttl.sls_ttl,
    ttl.rtn_cnt,
    ttl.rtn_ttl,
    ttl.dsc_ttl,
    (select sttl_dsc_ttl
      from dly_sys_ttl
      where business_date=ttl.business_date)
    from M3700.maj_grp_def as def
    ,M3700.dly_sys_maj_grp_ttl as ttl
    where def.maj_grp_seq=ttl.maj_grp_seq
```

This example creates a view using the System Definition, System Status, Major Group Definition, and Daily System Major Group Total tables.

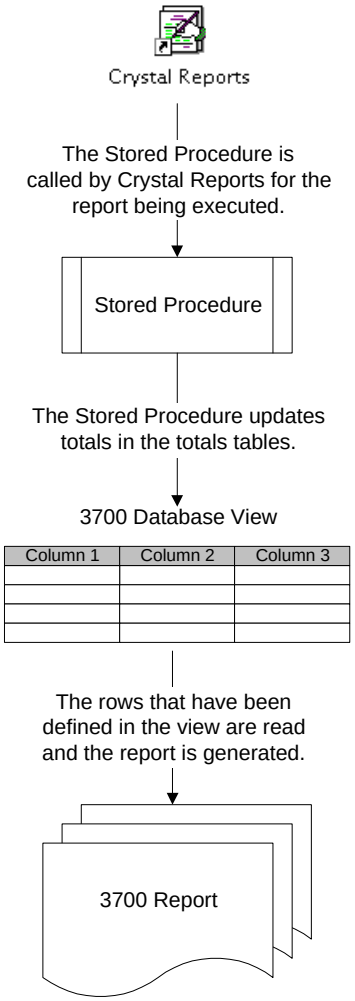
Refer to the *Sybase SQL Anywhere User's Guide* for detailed information on using views.

3700 Views

The 3700 database views can be categorized according to their purpose. Some views are used internally by POS Operations. Other views are used for creating reports.

Views for Reports

In the 3700 System, views are used by the Crystal Reports templates to generate reports. When a 3700 report is run, the appropriate stored procedure is called by Crystal Reports. The stored procedure updates the totals that will be used by the report. Once the totals have been updated, the Crystal Reports template reads the rows that have been defined in the view and generates the report.



Refer to the *3700 Custom Reports Design* manual for more information on how views are used by 3700 report templates.

The information below is listed for each of the report views in the following sections.

- Purpose of the view
- Tables and columns used to create the view
- Stored procedure that updates the totals used by the view
- Report name and report template that uses the view

v_R_cashier

The v_R_cashier view provides financial information for each cashier, such as net sales totals and check activity.

Tables/Columns

The tables and columns used to create the v_R_cashier view are listed below:

Tables	Columns	
Cashier Total Definition (cshr_ttl_def)	name	obj_number
Daily Cashier Totals (shift_cshr_ttl)	shift_seq start_time item_dsc_ttl rtn_cnt svc_ttl void_ttl credit_ttl change_grand_ttl cancel_ttl mgr_void_ttl error_correct_ttl chk_begun_ttl chk_paid_ttl chk_xfer_in_ttl chk_xfer_out_ttl	end_time net_sls_ttl sttl_dsc_ttl rtn_ttl void_cnt tax_coll_ttl rounding_ttl cancel_cnt mgr_void_cnt error_correct_cnt chk_begun_cnt chk_paid_cnt chk_xfer_in_cnt chk_xfer_out_cnt no_sale_cnt
Restaurant (rest_def)	obj_num	store_id
Order Type Definition (order_type_def)	name	
Shift Cashier Order Type Totals (shift_cshr_ot_ttl)	order_type_seq net_sls_ttl cov_cnt	chk_cnt tbl_turn_cnt tbl_dine_time

Stored Procedure

sp_R_cashier

Reports

The v_R_cashier view is used to create the following reports:

Cashier Summary (CSHR_001.rpt)
Cashier Detail (CSHR_101.rpt)

v_R_cashier_trk

The v_R_cashier_trk view provides financial information for each cashier, based on the tracking groups defined.

Tables/Columns

The tables and columns used to create the v_R_cashier_trk view are listed below:

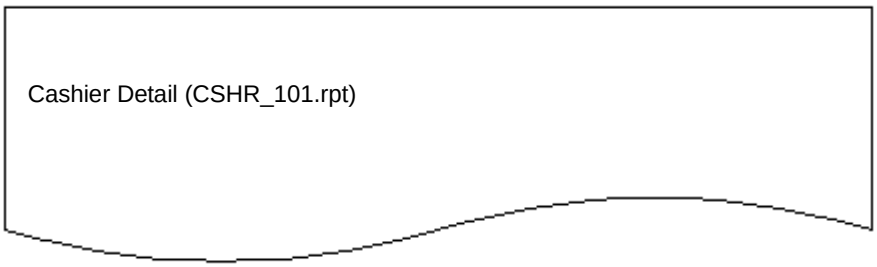
Tables	Columns	
Cashier Total Definition (cshr_ttl_def)	obj_num	name
Restaurant (rest_def)	obj_num	store_id
Tracking Group Definition (trk_grp_def)	obj_number	name
Tracking Group Total Definition Temp (trk_ttl_def_temp)	ttl_name_01 *to* ttl_name_64	
Shift Cashier Tracking Totals (shift_cshr_trk_ttl)	shift_seq trk_cnt_01 *to* trk_cnt_64\	trk_ttl_01 *to* trk_ttl_64

Stored Procedure

sp_R_cashier_sales_tracking

Reports

The v_R_cashier_trk view is used to create the following reports:



v_R_CCBatch

The v_R_CCBatch view provides credit card and credit authorization information.

Tables/Columns

The tables and columns used to create the v_R_CCBatch view are listed below:

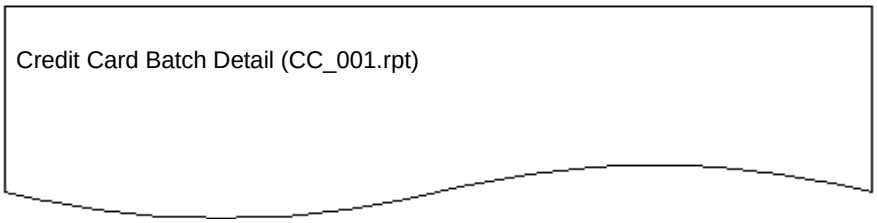
Tables	Columns	
Credit Card Batch Detail (cc_batch_dtl)	creation_emp_seq creation_time	batch_seq processor_batch_ref
Credit Card Batch Item Detail (cc_batch_item_dtl)	batch_item_seq cc_acct_num expiration_date customer_name ob_manual_entry ob_settled omitted_by alternate_tender_name	start_date issue_number check_num date_time base_ttl charge_tip_ttl num_auths ob_immed_pymnt
Credit Card Batch CA Detail (cc_batch_ca_dtl)	batch_ca_seq response_type auth_code auth_date_time base_amount tip_amount	ob_manual_auth ob_demo_driver ob_void_flag driver_data_1 driver_data_2 driver_data_3
CA/EDC Driver (caedc_driver_def)	obj_num	name
RVC/System Params (rvc_cfg_temp)	rvc_obj_num	rvc_seq
Employee (emp_def)	obj_number emp_seq	first_name last_name
Tender/Media (tmed_def)	obj_number	name
Device Table (dev_def)	name	

Stored Procedure

This view does not use any totals tables and therefore does not require a stored procedure.

Reports

The v_R_cashier_trk view is used to create the following reports:



v_R_CCBatch_Xfer_Status

The v_R_CCBatch_Xfer_Status view provides status information for each credit card batch transfer.

Tables/Columns

The tables and columns used to create the v_R_CCBatch_Xfer_Status view are listed below:

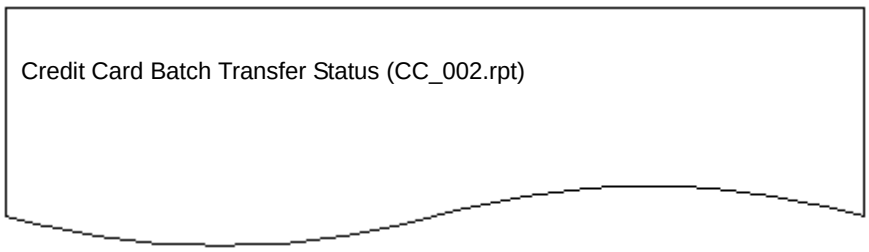
Tables	Columns	
Credit Card Batch Detail (cc_batch_dtl)	processor_batch_ref batch_seq	creation_time done_settle_time
CA/EDC Driver (caedc_driver_def)	obj_num	name
Credit Card Batch Transfer Status (cc_batch_xfer_status)	settle_attempt_seq settle_time response_text_1 response_text_2	response_text_3 response_text_4 previous_settle_cnt
Credit Card Batch Transfer Item Status (cc_batch_xfer_item_status)	item_ seq cc_acct_num expiration_date ob_manual_entry chk_num	date_time base_total charge_tip_ttl omitted_by omitted_desc
Employee (emp_def)	emp_num emp_seq	first_name last_name

Stored Procedure

This view does not use any totals tables and therefore does not require a stored procedure.

Reports

The v_R_CCBatch_Xfer_Status view is used to create the following reports:



v_R_employee

The v_R_employee view provides sales and tip information for each employee, such as net sales and charged tips.

Tables/Columns

The tables and columns used to create the v_R_employee view are listed below:

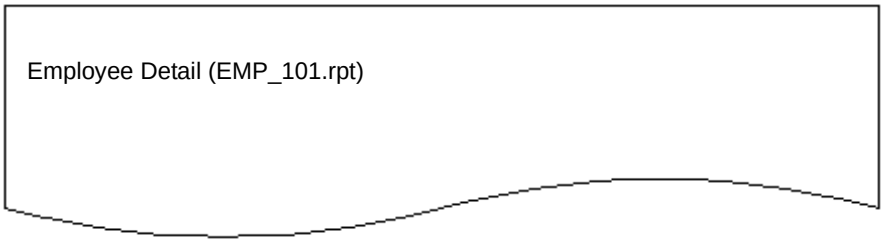
Tables	Columns	
Employee Definition (emp_def)	obj_num payroll_id	first_name last_name
Restaurant (rest_def)	obj_num	store_id
Shift Employee Totals (shift_emp_ttl)	shift_start_time sttl_dsc_ttl item_dsc_ttl rtn_cnt svc_ttl void_ttl credit_ttl change_grand_ttl cancel_ttl mgr_void_ttl error_correct_ttl chk_carry_over_ttl chk_begun_ttl chk_paid_ttl chk_xfer_in_ttl chk_xfer_out_ttl chk_cnt tbl_dine_time gross_rcpts_ttl charged_tips_ttl tips_paid_ttl tip_svc_ttl	shift_end_time shift_seq net_sls_ttl stl_dsc_ttl rtn_ttl void_cnt tax_coll_ttl rounding_ttl training_ttl cancel_cnt mgr_void_cnt error_correct_cnt chk_carry_over_cnt chk_begun_cnt chk_paid_cnt chk_xfer_in_cnt chk_xfer_out_cnt cov_cnt tbl_turn_cnt no_sale_cnt charged_rcpts_ttl tips_decl_ttl
Order Type Definition (order_type_def)	name	
Shift Employee Order Type Totals (shift_emp_ot_ttl)	order_type_seq ot_net_sales_ttl chk_cnt	cov_cnt tbl_turn_cnt tbl_dine_time

Stored Procedure

sp_R_employee

Reports

The v_R_employee view is used to create the following reports:



v_R_employee_job_code

The v_R_employee_job_code view provides job information for each employee, such as regular and overtime work hours.

Tables/Columns

The tables and columns used to create the v_R_employee_job_code view are listed below:

Tables	Columns	
Employee Definition (emp_def)	obj_num emp_seq payroll_id	last_name first_name
Job Definition (job_def)	obj_num lab_cat	name
Job Category Definition (job_cat_def)	obj_num	name
Employee Job (job_rate_def)	override_reg_pay_rate	override_otm_pay_rate
Time Clock Definition (time_clock_def)	lab_cat_1_name lab_cat_3_name	lab_cat_2_name lab_cat_4_name
Daily Employee Job Code Total (dly_emp_job_code_ttl)	reg_hrs otm_hours otm_ttl	labor_week reg_ttl

Stored Procedure

sp_R_employee_job_code

Reports

The v_R_employee_job_code view is used to create the following reports:

Employee Job Summary (TIME_003.rpt)
Consolidated Employee Job Summary (TIME_004.rpt)

v_R_employee_sales_tips

The v_R_employee_sales_tips view provides sales and tip information for each employee, such as net sales and charged tips.

Tables/Columns

The tables and columns used to create the v_R_employee_sales_tips view are listed below:

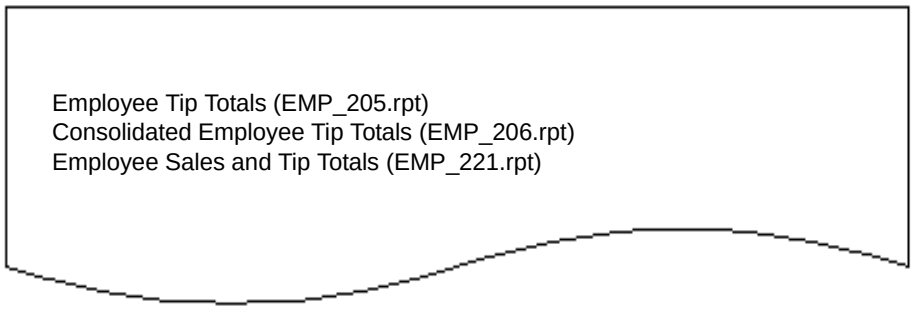
Tables	Columns	
Employee Definition (emp_def)	obj_id	first_name last_name
Shift Employee Totals (shift_emp_ttl)	shift_seq shift_start_time shift_end_time net_sls_ttl sttl_dsc_ttl iteml_dsc_ttl rtn_cnt rtn_ttl srv_ttl void_cnt void_ttl tax_coll_ttl credit_ttl rounding_ttl change_grand_ttl training_ttl cancel_cnt cancel_ttl mgr_void_cnt mgr_void_ttl error_correct_cnt error_correct_ttl	chk_carry_over_cnt chk_carry_over_ttl chk_begun_cnt chk_begun_ttl chk_paid_cnt chk_paid_ttl chk_xfer_in_cnt chk_xfer_in_ttl chk_xfer_out_cnt chk_xfer_out_ttl cov_cnt chk_cnt tbl_turn_cnt tbl_dine_time no_sale_cnt gross_rcpts_ttl chgd_rcpts_ttl chgd_tips_ttl tip_svc_ttl tips_paid_ttl tips_decl_ttl
Restaurant (rest_def)	obj_num	name

Stored Procedure

sp_R_employee

Reports

The v_R_employee_sales_tips view is used to create the following reports:



v_R_employee_time_card

The v_R_employee_time_card view is used to create the Employee Time Reports.

Tables/Columns

The tables and columns used to create the v_R_employee_time_card view are listed below:

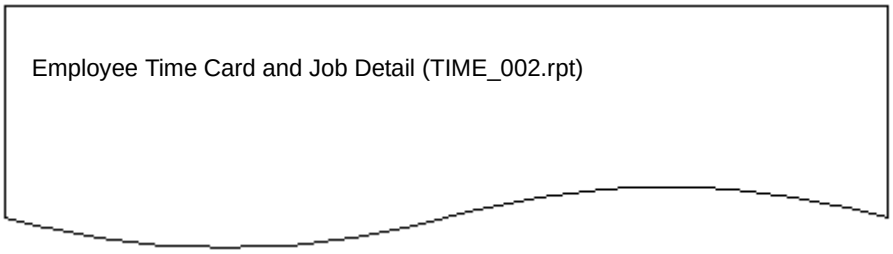
Tables	Columns	
Employee Definition (emp_def)	employee_seq payroll_id	first_name last_name
Job Definition (job_def)	obj_num name lab_cat	deflt_otm_pay_rate deflt_reg_pay_rate
Job Category Definition (job_cat_def)	obj_num	name
Job Rate Definition (job_rate_def)	override_reg_pay_rate	override_otm_pay_rate
Time Clock Definition (time_clock_def)	lab_wk_start_day hrs_before_daily_otm	hrs_before_prd_otm
Time Card Detail (time_card_dtl)	tm_card_seq tm_clk_sched_seq clk_in_date_tm clk_out_date_tm reg_hrs ovt_hrs	clock_in_status clock_out_status labor_week reg_ttl ovt_ttl
View of v_R_time_card_adj	last_name first_name	reason_name
Daily System Total (dly_sys_ttl)	business-date	
Revenue Center Configuration Temp (rvc_cfg_temp)	rvc_number	
Restaurant (rest_def)	bus_day__start_tm	

Stored Procedure

sp_R_time_card_ttls

Reports

The v_R_employee_time_card view is used to create the following reports:



v_R_employee_time_card_status

The v_R_employee_time_card_status view is used to create the Employee Time Reports.

Tables/Columns

The tables and columns used to create the v_R_employee_time_card_status view are listed below:

Tables	Columns	
Employee Definition (emp_def)	employee_seq payroll_id	first_name last_name
Job Definition (job_def)	obj_num name	lab_cat
Job Category Definition (job_cat_def)	obj_num	name
Time Card Detail (time_card_dtl)	tclk_in_date_tm clk_out_date_tm	clock_in_status clock_out_status
View of v_R_time_card_adj	last_name first_name	reason_name
Revenue Center Configuration Temp (rvc_cfg_temp)	rvc_number	

Stored Procedure

This view does not use any totals tables and therefore does not require a stored procedure.

Reports

The v_R_employee_time_card_status view is used to create the following reports:

Employee Clock IN Status (TIME_005.rpt)
Labor Availibility (TIME_006.rpt)

v_R_employee_trk

The v_R_employee_trk view provides tracking total information for each employee.

Tables/Columns

The tables and columns used to create the v_R_employee_trk view are listed below:

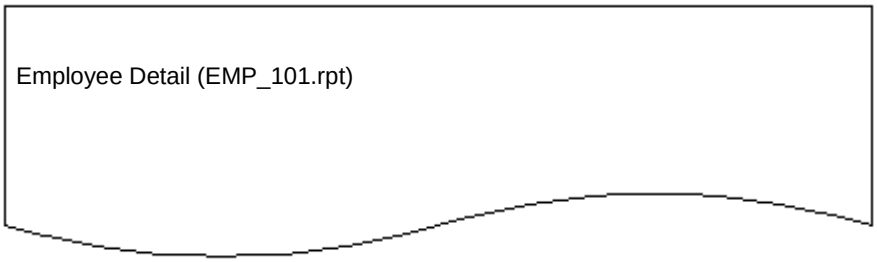
Tables	Columns	
Employee Definition (emp_def)	last_name payroll_id	obj_number first_name
Tracking Group Definition (trk_grp_def)	obj_num	name
Restaurant (rest_def)	obj_num	store_id
Shift Employee Tracking Totals (shift_emp_trk_ttl)	shift_number trk_cnt_01 *to* trk_cnt_64	trk_ttl_01 *to* trk_ttl_64
Tracking Total Definition Temp (trk_ttl_def_temp)	ttl_name_01 *to* ttl_name_64	

Stored Procedure

sp_R_employee_sales_tracking

Reports

The v_R_employee_trk view is used to create the following reports:



v_R_guest_checks

The v_R_guest_checks view provides guest check information for each employee.

Tables/Columns

The tables and columns used to create the v_R_guest_checks view are listed below:

Tables	Columns	
Revenue Center Configuration Temp (rvc_cfg_temp)	rvc_seq	
Employee Definition (emp_def)	emp_seq last_name	employee_number first_name
Check Detail (chk_dtl)	chk_num grp id chk_open ob_ccs_02_ca_check ob_ccs04_chk_added ob_ccs11_chk_edited chk_open_date_time tbl_open_date_time tax_ttl auto_svc_ttl paymnt_ttl	cov_cnt ob_ocs10_tms_created_chk ob_ocs11_suspended_chk ob_ccs10_ca_chk_batched ob_ccs03_fast_trans_chk ob_ccs05_chk_cancelled training_status chk_clsd_date_time sub_ttl other_svc_ttl amount_due_ttl chk_prntd_cnt
Table Definition (tbl_def)	table_seq	name
Order Type Definition (order_type_def)	order_type_seq	name

Stored Procedure

This view does not use any totals tables and therefore does not require a stored procedure.

Reports

The v_R_guest_checks view is used to create the following reports:

Employee Open Guest Checks (CHK_101.rpt)
Employee Closed Guest Checks (CHK_102.rpt)

v_R_job_code

The v_R_job_code view provides information for each job code, such as regular and overtime work hours.

Tables/Columns

The tables and columns used to create the v_R_job_code view are listed below:

Tables	Columns	
Job Definition (job_def)	obj_num lab_cat	name
Restaurant (rest_def)	obj_num	store_id
Job Category Definition (job_cat_def)	obj_num	name
Time Clock Definition (time_clock_def)	lab_cat_1_name lab_cat_2_name	lab_cat_3_name lab_cat_4_name
Daily Employee Job Code Total (dly_job_code_ttl)	business_date reg_hrs ovt_hrs net_sls_ttl	labor_week reg_ttl ovt_ttl
Revenue Center Configuration Temp (rvc_cfg_temp)	rvc_seq	

Stored Procedure

sp_R_job_code_totals

Reports

The v_R_job_code view is used to create the following reports:

RVC Job Summary (TIME_R011.rpt)
Consolidated RVC Job Summary (TIME_R012.rpt)
System Job Summary (TIME_S011.rpt)
Consolidated System Job Summary (TIME_S012.rpt)

v_R_rvc_cashier

The v_R_rvc_cashier view provides financial information for each cashier within an RVC, such as net sales totals and check activity.

Tables/Columns

The tables and columns used to create the v_R_rvc_cashier view are listed below:

Tables	Columns	
Cashier Total Definition (cshr_ttl_def)	name cashier_ttl_seq	obj_number
Daily RVC Cashier Totals (shift_rvc_cshr_ttl)	store_id shift_seq shift_start_time shift_end_time item_dsc_ttl rtn_cnt svc_ttl void_ttl credit_ttl cancel_ttl error_correct_ttl chk_begun_ttl chk_paid_ttl chk_xfer_in_ttl chk_xfer_out_ttl	net_sls_ttl sttl_dsc_ttl rtn_ttl void_cnt tax_coll_ttl rounding_ttl cancel_cnt mgr_void_cnt mgr_void_ttl error_correct_cnt chk_begun_cnt chk_paid_cnt chk_xfer_in_cnt chk_xfer_out_cnt no_sale_cnt
Restaurant (rest_def)	obj_num	
Order Type Definition (order_type_def)	name	
Shift Cashier Order Type Totals (shift_cshr_ot_ttl)	order_type_seq net_sls_ttl cov_cnt	chk_cnt tbl_turn_cnt tbl_dine_time

Stored Procedure

sp_R_rvc_cashier
sp_R_rvc_cashier_sales_tracking

Reports

The v_R_cashier view is used to create the following reports:

Cashier Summary by Revenue Center (CSHR_601.rpt)
Cashier Detail by Revenue Center (CSHR_701.rpt)

v_R_rvc_cashier_trk

The v_R_rvc_cashier_trk view provides tracking total information for each employee in an RVC.

Tables/Columns

The tables and columns used to create the v_R_rvc_cashier_trk view are listed below:

Tables	Columns	
Employee Definition (cshr_ttl_def)	cshr_ttl_seq name	obj_num
Restaurant (rest_def)	obj_num	store_id
Shift Employee Tracking Totals (shift_rvc_cshr_trk_ttl)	trk_cnt_01 *to* trk_cnt_64 store_id	trk_ttl_01 *to* trk_ttl_64 shift_seq
Tracking Group Definition (trk_grp_def)	obj_num	ttl_name_01 *to* ttl_name_64
RVC/System Params (rvc_cfg_temp)	obj_num	name

Stored Procedure

sp_R_rvc_cashier_tracking
sp_R_rvc_cashier_sales_tracking

Reports

The v_R_rvc_cashier_trk view is used to create the following reports:

Employee Detail by Revenue Center (EMP_701.rpt)

v_R_rvc_employee

The v_R_rvc_employee view provides sales and tip total information for each employee.

Tables/Columns

The tables and columns used to create the v_R_rvc_employee view are listed below:

Tables	Columns	
Employee Definition (emp_def)	emp_seq last_name payroll_id	obj_number first_name
Order Type Definition (order_type_def)	name	
Restaurant (rest_def)	obj_num	
Shift RVC Employee Totals (shift_rvc_emp_ttl)	store_id shift_seq shift_start_time shift_end_time item_dsc_ttl rtn_cnt svc_ttl void_ttl credit_ttl change_grand_total cancel_ttl chk_carry_over_cnt error_correct_ttl chk_begun_ttl chk_paid_ttl chk_xfer_in_ttl chk_xfer_out_ttl gross_rcpts_ttl charged_tips_ttl tips_paid_ttl cov_cnt tbl_turn_cnt	net_sls_ttl sttl_dsc_ttl tax_coll_ttl rounding_ttl rtn_ttl mgr_void_cnt mgr_void_ttl void_cnt training_ttl error_correct_cnt cancel_cnt chk_carry_over_ttl chk_begun_cnt chk_paid_cnt chk_xfer_in_cnt chk_xfer_out_cnt no_sale_cnt charged_rcpts_ttl tip_svc_ttl tips_decl_ttl chk_cnt tbl_dine_time
RVC Shift Employee Order Type Totals (shift_rvc_emp_ot_ttl)	order_type_seq net_sls_ttl chk_cnt	cov_cnt tbl_turn_cnt tbl_dine_time
RVC/System Params (rvc_cfg_temp)	obj_num	name

Stored Procedure

sp_R_rvc_employee_sales_tracking

Reports

The v_R_rvc_employee view is used to create the following reports:

Employee Detail by Revenue Center (EMP7101.rpt)

v_R_rvc_employee_sales_tips

The v_R_rvc_employee_sales_tips view provides sales and tip information for each employee in an RVC, such as net sales and charged tips.

Tables/Columns

The tables and columns used to create the v_R_rvc_employee_sales_tips view are listed below:

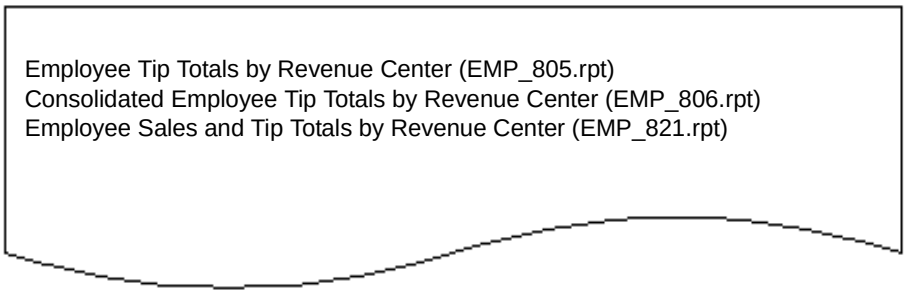
Tables	Columns	
Employee Definition (emp_def)	obj_num payroll_id	first_name last_name
Shift Employee Totals (shift_rvc_emp_ttl)	store_id shift_seq shift_start_time shift_end_time net_sls_ttl sttl_dsc_ttl iteml_dsc_ttl rtn_cnt rtn_ttl srv_ttl void_cnt void_ttl tax_coll_ttl credit_ttl rounding_ttl change_grand_ttl cancel_cnt cancel_ttl mgr_void_cnt mgr_void_ttl error_correct_cnt error_correct_ttl	chk_carry_over_cnt chk_carry_over_ttl chk_begun_cnt chk_begun_ttl chk_paid_cnt chk_paid_ttl chk_xfer_in_cnt chk_xfer_in_ttl chk_xfer_out_cnt chk_xfer_out_ttl cov_cnt chk_cnt tbl_turn_cnt tbl_dine_time no_sale_cnt gross_rcpts_ttl chgd_rcpts_ttl chgd_tips_ttl tip_svc_ttl tips_paid_ttl tips_decl_ttl training_ttl
Restaurant (rest_def)	obj_num	
RVC/System Params (rvc_cfg_temp)	obj_num	name

Stored Procedure

sp_R_rvc_employee

Reports

The v_R_rvc_employee_sales_tips view is used to create the following reports:



v_R_rvc_employee_trk

The v_R_rvc_employee_trk view provides tracking total information for each employee in an RVC.

Tables/Columns

The tables and columns used to create the v_R_rvc_employee_trk view are listed below:

Tables	Columns	
Employee Definition (emp_def)	last_name payroll_id emp_seq	obj_number first_name
Tracking Group Definition (trk_grp_def)	obj_num	name
Restaurant (rest_def)	obj_num	
Shift Employee Tracking Totals (shift_rvc_emp_trk_ttl)	shift_seq trk_cnt_01 *to* trk_cnt_64	store_id trk_ttl_01 *to* trk_ttl_64
Tracking Total Definition Temp (trk_ttl_def_temp)	ttl_name_01 *to* ttl_name_64	
RVC/System Params (rvc_cfg_temp)	obj_num	name

Stored Procedure

sp_R_rvc_employee_tracking

Reports

The v_R_rvc_employee_trk view is used to create the following reports:

Employee Detail by Revenue Center (EMP_701.rpt)

v_R_rvc_fam_grp

The v_R_rvc_fam_grp view provides a summary by revenue center of family group sales activity.

Tables/Columns

The tables and columns used to create the v_R_rvc_fam_grp view are listed below:

Tables	Columns	
Revenue Center Configuration Temp (rvc_cfg_temp)	rvc_obj_num	rvc_seq
Daily Revenue Center Family Group Total (dly_rvc_fam_grp_ttl)	business_date sls_cnt rtn_cnt dsc_ttl	group_number sls_ttl rtn_ttl
Family Group Definition (fam_grp_def)	obj_num fam_grp_seq	cat name
Restaurant (rest_def)	obj_num	store_id

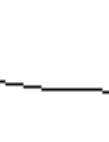
Stored Procedure

sp_R_family_group

Reports

The v_R_rvc_fam_grp view is used to create the following reports:

RVC Family Group Sales Detail (FG_R001.rpt)
RVC Family Group Sales Detail, Subtotaled By Category (FG_R010.rpt)
Consolidated RVC Family Group Sales Detail (FG_R002.rpt)
Consolidated RVC Family Group Sales Detail, Subtotaled By Family Group (FG_R012.rpt)
Revenue Center Family Group Sales Summary (FG_R101.rpt)
Consolidated Revenue Center Family Group Sales Summary (FG_R111.rpt)



Note

These report templates are also used to create the RVC Major Group Sales Detail and RVC Menu Item Group Sales Detail reports.

v_R_rvc_maj_grp

The v_R_rvc_maj_grp view provides a summary by revenue center of major group sales activity.

Tables/Columns

The tables and columns used to create the v_R_rvc_maj_grp view are listed below:

Tables	Columns	
Revenue Center Configuration Temp (rvc_cfg_temp)	rvc_obj_num	rvc_seq
Daily Revenue Center Major Group Total (dly_rvc_maj_grp_ttl)	business_date sls_cnt rtn_cnt dsc_ttl	sls_ttl rtn_ttl
Major Group Definition (maj_grp_def)	obj_num maj_grp_seq	“name” cat
Restaurant (rest_def)	obj_num	store_id

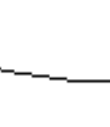
Stored Procedure

sp_R_major_group

Reports

The v_R_rvc_maj_grp view is used to create the following reports:

RVC Major Group Sales Detail (FG_R001.rpt)
RVC Major Group Sales Detail, Subtotaled By Category (FG_R010.rpt)
Consolidated RVC Major Group Sales Detail (FG_R002.rpt)
Consolidated RVC Major Group Sales Detail, Subtotaled By Major Group (FG_R012.rpt)
Revenue Center Family Group Sales Summary (FG_R101.rpt)
Consolidated Revenue Center Family Group Sales Summary (FG_R111.rpt)



Note

These report templates (*.rpt) are also used to create the RVC Family Group Sales Detail and RVC Menu Item Group Sales Detail reports.

v_R_rvc_mi_grp

The v_R_rvc_mi_grp view provides a summary by revenue center of menu item group sales activity.

Tables/Columns

The tables and columns used to create the v_R_rvc_mi_grp view are listed below:

Tables	Columns	
Revenue Center Configuration Temp (rvc_cfg_temp)	rvc_seq	
Daily Revenue Center Menu Item Group Total (dly_rvc_mi_grp_ttl)	sls_cnt sls_ttl rtn_cnt	rtn_ttl dsc_ttl
Menu Item Group Definition (mi_grp_def)	obj_num mi_grp_seq	“name” cat
Restaurant (rest_def)	obj_num	store_id

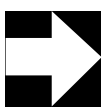
Stored Procedure

sp_R_menuitem_group

Report Templates

The v_R_rvc_mi_grp view is used to create the following reports:

RVC Menu Item Group Sales Detail (FG_R001.rpt)
RVC Menu Item Sales Detail, Subtotaled By Category (FG_R010.rpt)
Consolidated RVC Menu Item Sales Detail (FG_R002.rpt)
Consolidated RVCMenu Item Sales Detail, Subtotaled By Major Group (FG_R012.rpt)
Revenue Center Family Group Sales Summary (FG_R101.rpt)
Consolidated Revenue Center Family Group Sales Summary (FG_R111.rpt)



Note

Note that these report templates (*.rpt) are also used to create the RVC Family Group Sales Detail and RVC Major Group Sales Detail reports.

v_R_rvc_menuitem

The v_R_rvc_menuitem view may be used to create a Revenue Center Menu Item Report. This view is not used by any of the standard 3700 reports.

Tables/Columns

The tables and columns used to create the v_R_rvc_menuitem view are listed below:

Tables	Columns	
Revenue Center Configuration Temp (rvc_cfg_temp)	rvc_obj_number rvc_seq main_mlvl_1_name sub_mlvl_1_name main_mlvl_2_name	sub_mlvl_2_name main_mlvl_3_name sub_mlvl_3_name main_mlvl_4_name sub_mlvl_4_name
Daily Revenue Center Menu Item Total (dly_rvc_mi_ttl)	business_date price1_sls_ttl price1_dsc_ttl price2_sls_ttl price2_dsc_ttl price3_slsttl price3_dsc_ttl price4_sls_ttl price4_dsc_ttl	price1_sls_cnt price1_rtn_cnt price2_sls_cnt price2_rtn_cnt price3_sls_cnt price3_rtn_cnt price4_sls_cnt price4_rtn_cnt
Menu Item Definition (mi_def)	obj_number mi_seq nlu_num	name_1 name_2
Major Group Definition (maj_grp_def)	obj_number	name
Family Group Definition (fam_grp_def)	obj_number	name
Menu Item Group Definition (mi_grp_def)	obj_number	name
Menu Item Price Definition (mi_price_def)	preset_amt_1 preset_amt_2	preset_amt_3 preset_amt_4
Menu Item Type Class Definition (mi_type_class_def)	ob_mi20_sub_lvl_for_prices	
Restaurant (rest_def)	obj_num	name

Stored Procedure

sp_R_menuitem

Reports

This view is not used by any of the standard 3700 reports.

v_R_rvc_menuitem_fam_grp

The v_R_rvc_menuitem_fam_grp view provides detailed sales information by revenue center for each of the menu items in a family group.

Tables/Columns

The tables and columns used to create the v_R_rvc_menuitem_fam_grp view are listed below:

Tables	Columns	
Revenue Center Configuration Temp (rvc_cfg_temp)	rvc_obj_num rvc_seq main_mlvl_1_name sub_mlvl_1_name main_mlvl_2_name	sub_mlvl_2_name main_mlvl_3_name sub_mlvl_3_name main_mlvl_4_name sub_mlvl_4_name
Daily Revenue Center Menu Item Total (dly_rvc_mi_ttl)	business_date price1_sls_ttl price1_dsc_ttl price2_sls_ttl price2_dsc_ttl price3_slsttl price3_dsc_ttl price4_sls_ttl price4_dsc_ttl	price1_sls_cnt price1_rtn_cnt price2_sls_cnt price2_rtn_cnt price3_sls_cnt price3_rtn_cnt price4_sls_cnt price4_rtn_cnt
Menu Item Definition (mi_def)	obj_num mi_seq nlu_num	name1 name2
Family Group Definition (fam_grp_def)	obj_number	name
Menu Item Price Definition (mi_price_def)	preset_amt_1 preset_amt_2	preset_amt_3 preset_amt_4
Menu Item Type Class Definition (mi_type_class_def)	ob_mi20_sub_lvl_for_prices	
Restaurant (rest_def)	obj_num	store_id

Stored Procedure

sp_R_menuitem

Reports

The v_R_rvc_menuitem_fam_grp view is used to create the following reports:

RVC Menu Item Sales Detail, Subtotaled By Family Group (MI_R001.rpt)
Consolidated RVC Menu Item Sales Detail, Subtotaled By Family Group (MI_R002.rpt)
RVC Menu Item Sales Summary, Subtotaled By Family Group (MI_R101.rpt)
Consolidated RVC Menu Item Sales Summary, Subtotaled By Family Group (MI_R102.rpt)

v_R_rvc_menuitem_maj_grp

The v_R_rvc_menuitem_maj_grp view provides detailed sales information by revenue center for each of the menu items in a major group.

Tables/Columns

The tables and columns used to create the v_R_rvc_menuitem_maj_grp view are listed below:

Tables	Columns	
Revenue Center Configuration Temp (rvc_cfg_temp)	rvc_obj_num rvc_seq main_mlvl_1_name sub_mlvl_1_name main_mlvl_2_name	sub_mlvl_2_name main_mlvl_3_name sub_mlvl_3_name main_mlvl_4_name sub_mlvl_4_name
Daily Revenue Center Menu Item Total (dly_rvc_mi_ttl)	business_date price1_sls_ttl price1_dsc_ttl price2_sls_ttl price2_dsc_ttl price3_slsttl price3_dsc_ttl price4_sls_ttl price4_dsc_ttl	price1_sls_cnt price1_rtn_cnt price2_sls_cnt price2_rtn_cnt price3_sls_cnt price3_rtn_cnt price4_sls_cnt price4_rtn_cnt
Menu Item Definition (mi_def)	obj_num mi_seq nlu_num	name1 name2
Major Group Definition (maj_grp_def)	obj_number	name
Menu Item Price Definition (mi_price_def)	preset_amt_1 preset_amt_2	preset_amt_3 preset_amt_4
Menu Item Type Class Definition (mi_type_class_def)	ob_mi20_sub_lvl_for_prices	
Restaurant (rest_def)	obj_num	store_id

Stored Procedure

sp_R_menuitem

Reports

The v_R_rvc_menuitem_maj_grp view is used to create the following reports:

RVC Menu Item Detail, Subtotaled By Major Group (MI_R001.rpt)
Consolidated RVC Menu Item Detail, Subtotaled By Major Group (MI_R002.rpt)
RVC Menu Item Summary, Subtotaled By Major Group (MI_R101.rpt)
Consolidated RVC Menu Item Summary, Subtotaled By Major Group (MI_R102.rpt)

v_R_rvc_menuitem_mi_grp

The v_R_rvc_menuitem_mi_grp view provides detailed sales information by revenue center for each of the menu items in a menu item group.

Tables/Columns

The tables and columns used to create the v_R_rvc_menuitem_mi_grp view are listed below:

Tables	Columns	
Revenue Center Configuration Temp (rvc_cfg_temp)	rvc_obj_num rvc_seq main_mlvl_1_name sub_mlvl_1_name main_mlvl_2_name	sub_mlvl_2_name main_mlvl_3_name sub_mlvl_3_name main_mlvl_4_name sub_mlvl_4_name
Daily Revenue Center Menu Item Total (dly_rvc_mi_ttl)	business_date price1_sls_ttl price1_dsc_ttl price2_sls_ttl price2_dsc_ttl price3_slsttl price3_dsc_ttl price4_sls_ttl price4_dsc_ttl	price1_sls_cnt price1_rtn_cnt price2_sls_cnt price2_rtn_cnt price3_sls_cnt price3_rtn_cnt price4_sls_cnt price4_rtn_cnt
Menu Item Definition (mi_def)	obj_num mi_seq nlu_num	name1 name2
Major Group Definition (maj_grp_def)	obj_number	name
Menu Item Price Definition (mi_price_def)	preset_amt_1 preset_amt_2	preset_amt_3 preset_amt_4
Menu Item Type Class Definition (mi_type_class_def)	ob_mi20_sub_lvl_for_prices	
Restaurant (rest_def)	obj_num	store_id

Stored Procedure

sp_R_menuitem

Reports

The v_R_rvc_menuitem_mi_grp view is used to create the following reports:

RVC Menu Item Detail, Subtotaled By Menu Item Group (MI_R001.rpt)
Consolidated RVC Menu Item Detail, Subtotaled By Menu Item Group (MI_R002.rpt)
RVC Menu Item Summary, Subtotaled By Menu Item Group (MI_R101.rpt)
Consolidated RVC Menu Item Summary, Subtotaled By Menu Item Group (MI_R102.rpt)

v_R_rvc_sales

The v_R_rvc_sales view provides sales and tip information for each revenue center, such as net sales and charged tips.

Tables/Columns

The tables and columns used to create the v_R_rvc_sales view are listed below:

Tables	Columns	
Revenue Center Configuration Temp (rvc_cfg_temp)	rvc_seq	
Daily Revenue Center Total (dly_rvc_ttl)	business_date net_sales_ttl item_discount_ttl sttl_discount_ttl rtn_cnt rtn_ttl svc_ttl void_cnt void_ttl tax_coll_ttl credit_ttl rounding_ttl change_grand_ttl grand_total non_txbl_ttl cancel_cnt cancel_ttl mgr_void_cnt mgr_void_ttl error_correct_cnt error_correct_ttl chk_carry_over_cnt chk_carry_over_ttl chk_begun_cnt chk_begun_ttl chk_paid_cnt chk_paid_ttl	chk_xfer_in_cnt chk_xfer_in_ttl chk_xfer_out_cnt chk_xfer_out_ttl gross_receipts_ttl chgd_receipts_ttl chgd_tips_ttl tip_svc_ttl tips_paid_ttl tips_dcl_ttl labor_cat_1_reg_hrs labor_cat_1_ovt_hrs labor_cat_1_reg_ttl labor_cat_1_ovt_ttl labor_cat_2_reg_hrs labor_cat_2_ovt_hrs labor_cat_2_reg_ttl labor_cat_2_ovt_ttl labor_cat_3_reg_hrs labor_cat_3_ovt_hrs labor_cat_3_reg_ttl labor_cat_3_ovt_ttl labor_cat_4_reg_hrs labor_cat_4_ovt_hrs labor_cat_4_reg_ttl labor_cat_4_ovt_ttl
Daily Revenue Center Order Type Totals (dly_rvc_ot_ttl)	order_type_seq ot_net_sls_ttl chk_count	cov_count tbl_turn_count tbl_dine_time
Order Type Definition (order_type_def)	name	
Restaurant (rest_def)	obj_num store_id	

Stored Procedure

sp_R_rvc_totals, sp_R_time_card_totals

Reports

The v_R_rvc_sales view is used to create the following reports:

RVC Sales Detail (RVC_101.rpt)
Consolidated RVC Sales Detail (RVC_102.rpt)

v_R_rvc_sales_tips

The v_R_rvc_sales_tips view provides sales and tip information for each revenue center, such as net sales and charged tips.

Tables/Columns

The tables and columns used to create the v_R_rvc_sales_tips view are listed below:

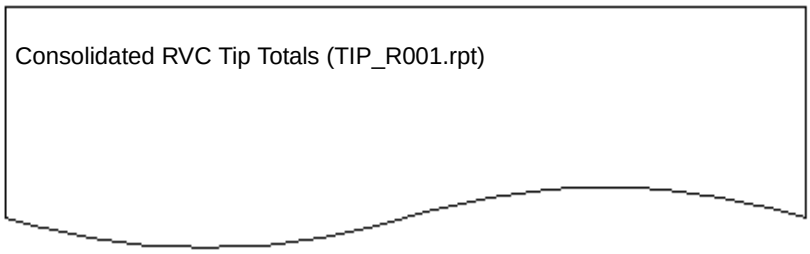
Tables	Columns	
Restaurant (rest_def)	obj_num	store_id
Daily Revenue Center Total (dly_rvc_ttl)	business_date net_sls_ttl item_dsc_ttl sttl_discount_ttl rtn_cnt rtn_ttl svc_ttl void_cnt void_ttl tax_coll_ttl credit_ttl rounding_ttl change_grand_ttl grand_total training_ttl non_txbl_ttl cancel_cnt cancel_ttl mgr_void_cnt mgr_void_ttl error_correct_cnt error_correct_ttl chk_carry_over_cnt chk_carry_over_ttl chk_begun_cnt chk_begun_ttl chk_paid_cnt	chk_paid_ttl chk_xfer_in_cnt chk_xfer_in_ttl chk_xfer_out_cnt chk_xfer_out_ttl gross_rcpts_ttl chgd_rcpts_ttl chgd_tips_ttl tip_svc_ttl tips_paid_ttl tips_decl_ttl labor_cat_1_reg_hrs labor_cat_1_ovt_hrs labor_cat_1_reg_ttl labor_cat_1_ovt_ttl labor_cat_2_reg_hrs labor_cat_2_ovt_hrs labor_cat_2_reg_ttl labor_cat_2_ovt_ttl labor_cat_3_reg_hrs labor_cat_3_ovt_hrs labor_cat_3_reg_ttl labor_cat_3_ovt_ttl labor_cat_4_reg_hrs labor_cat_4_ovt_hrs labor_cat_4_reg_ttl labor_cat_4_ovt_

Stored Procedure

sp_R_rvc_totals

Reports

The v_R_rvc_sales_tips view is used to create the following reports:



v_R_rvc_tax

The v_R_rvc_tax view provides tax information for each revenue center.

Tables/Columns

The tables and columns used to create the v_R_rvc_tax view are listed below:

Tables	Columns	
Revenue Center Configuration Temp (rvc_cfg_temp)	rvc_seq	
Daily Revenue Center Tax Total (dly_rvc_tax_ttl)	tax_coll_ttl exempt_ttl	tax_rate_seq txbl_sls_ttl
Daily Revenue Center Total (dly_rvc_ttl)	net_sls_ttl business_date	non_txbl_sales
Tax Rate Definition (tax_rate_def)	name percentage txbl_sls_name net_sls_name	type tax_coll_name tax_exempt_name
Restaurant (rest_def)	obj_num	store_id

Stored Procedure

sp_R_tax

Reports

The v_R_rvc_tax view is used to create the following reports:

RVC Tax Totals (TAX_R001.rpt)
Consolidated RVC Tax Totals (TAX_R002.rpt)
Canadian Revenue Center Tax Totals (TAX_cr01.rpt)
Canadian Consolidated Revenue Center Tax Totals (TAX_cr02.rpt)

v_R_rvc_time_period

The v_R_rvc_time_period view provides financial information for each revenue center based on time periods.

Tables/Columns

The tables and columns used to create the v_R_rvc_time_period view are listed below:

Tables	Columns	
Daily RVC Fixed Period Total (dly_rvc_fixed_prd_ttl)	rvc_seq store_number store_id business_date net_sales_ttl time_period_number time_period_seq time_period_name period_name period_number period_name period_start_time period_end_time labor_cat_1_reg_hrs labor_cat_1_ovt_hrs	labor_cat_1_reg_ttl labor_cat_1_ovt_ttl labor_cat_2_reg_hrs labor_cat_2_ovt_hrs labor_cat_2_reg_ttl labor_cat_2_ovt_ttl labor_cat_3_reg_hrs labor_cat_3_ovt_hrs labor_cat_3_reg_ttl labor_cat_3_ovt_ttl labor_cat_4_reg_hrs labor_cat_4_ovt_hrs labor_cat_4_reg_ttl labor_cat_4_ovt_ttl
Daily RVC Fixed Period Order Type Total (dly_rvc_fixed_prd_ot_ttl)	order_type_seq	
Order Type Definition (order_type_def)	“name”	

Stored Procedure

sp_R_time_period
sp_R_time_prd_time_card_ttls

Reports

The v_R_rvc_time_period view is used to create the following reports:

RVC Time Period Totals (TP_R001.rpt)
RVC Time Period Summary (TP_R103.rpt)
RVC Time Period Detail (TP_R101.rpt)

v_R_rvc_time_period_trk

The v_R_rvc_time_period_trk view provides financial information based on time periods for each revenue center.

Tables/Columns

The tables and columns used to create the v_R_rvc_time_period_trk view are listed below:

Tables	Columns	
Revenue Center Configuration Temp (rvc_cfg_temp)	rvc_seq	
Daily RVC Fxed Period Tracking Total (dly_rvc_fixed_prd_trk_ttl)	business_date trk_cnt_01 *to* trk_cnt_64	trk_ttl_01 *to* trk_ttl_64
Tracking Group Definition (trk_grp_def)	obj_number	“name”
Tracking Totals Definition Temp (trk_ttl_def_temp)	ttl_name_01 *to* ttl_name_64	
Time Period Definition (time_period_def)	time_period_seq obj_num	name
Period Definition (period_def)	obj_number name	start_time end_time
Restaurant (rest_def)	obj_num	store_id

Stored Procedure

sp_R_rvc_time_prd_sales_trk

Reports

The v_R_rvc_time_period_trk view is used to create the following reports:

RVC Time Period Detail (TP_R101.rpt)

v_R_rvc_trk

The v_R_rvc_trk view may be used to create a Revenue Center Tracking Group report.

Tables/Columns

The tables and columns used to create the v_R_rvc_trk view are listed below:

Tables	Columns
Revenue Center Configuration Temp (rvc_cfg_temp)	rvc_seq
Daily Revenue Center Tracking Totals (dly_rvc_trk_ttl)	business_date
Tracking Group Definition (trk_grp_def)	obj_num name
Tracking Totals Definition Temp (trk_ttl_def_temp)	ttl_name_01 *to* ttl_name_64
Restaurant (rest_def)	obj_num store_id

Stored Procedure

v_R_rvc_sales_tracking

Reports

The v_R_rvc_trk view is used to create the following reports:

RVC Sales Detail (RVC_101.rpt)
Revenue Center Sales Detail (RVC_101_2.rpt)

v_R_sys_fam_grp

The v_R_sys_fam_grp view provides a system-wide summary of family group sales activity.

Tables/Columns

The tables and columns used to create the v_R_sys_fam_grp view are listed below:

Tables	Columns	
Daily System Family Group Total (dly_sys_fam_grp_ttl)	business_date sls_ttl rtn_ttl	dsc_total sls_cnt rtn_cntl
Family Group Definition (fam_grp_def)	obj_num fam_grp_seq	cat name
Daily System Total (dly_sys_ttl)	sttl_dsc_ttl	
Restaurant (rest_def)	obj_num	store_id

Stored Procedure

sp_R_family_group

Reports

The v_R_sys_fam_grp view is used to create the following reports:

System Family Group Sales Detail (FG_S001.rpt)
System Family Group Sales Detail, Subtotaled By Category (FG_S010.rpt)
Consolidated System Family Group Sales Detail (FG_S002.rpt)
Consolidated SYS Family Group Sales Detail, Subtotaled By Category (FG_S012.rpt)
Consolidated System Family Group Sales Detail with Graph (FG_S202.rpt)
System Family Group Sales Summary (FG_S101.rpt)
Consolidated System Family Group Sales Summary (FG_S111.rpt)

v_R_sys_maj_grp

The v_R_sys_maj_grp view provides a system-wide summary of major group sales activity.

Tables/Columns

The tables and columns used to create the v_R_sys_maj_grp view are listed below:

Tables	Columns	
Daily System Major Group Total (dly_sys_maj_grp_ttl)	business_date sls_cnt rtn_cnt dsc_ttl	group_number sls_ttl rtn_ttl
Major Group Definition (maj_grp_def)	obj_num maj_grp_seq	name cat
Daily System Total (dly_sys_ttl)	sttl_dsc_ttl	
Restaurant (rest_def)	obj_num	store_id

Stored Procedure

sp_R_major_group

Reports

The v_R_sys_maj_grp view is used to create the following reports:

System Major Group Sales Detail (FG_S001.rpt)
System Major Group Sales Detail, Subtotaled By Category (FG_S010.rpt)
Consolidated System Major Group Sales Detail (FG_S002.rpt)
Consolidated SYS Major Group Sales Detail, Subtotaled By Category (FG_S012.rpt)
Consolidated System Major Group Sales Detail with Graph (FG_S202.rpt)
System Major Group Sales Summary (FG_S101.rpt)
Consolidated System Major Group Sales Summary (FG_S111.rpt)

v_R_sys_mi_grp

The v_R_sys_mi_grp view provides a system-wide summary of menu item group sales activity.

Tables/Columns

The tables and columns used to create the v_R_sys_mi_grp view are listed below:

Tables	Columns	
Daily System Menu Item Group Total (dly_sys_mi_grp_ttl)	business_date sls_ttl rtn_ttl dsc_ttl	sls_cnt rtn_cntl
Menu Item Group Definition (mi_grp_def)	obj_num mi_grp_seq	name cat
Daily System Totals (dly_sys_ttl)	sttl_dsc_ttll	
Restaurant (rest_def)	obj_num	store_id

Stored Procedure

sp_R_menuitem_group

Reports

The v_R_sys_mi_grp view is used to create the following reports:

System Menu Item Group Sales Detail (FG_S001.rpt)
System Menu Item Group Sales Detail, Subtotaled By Category (FG_S010.rpt)
Consolidated System Menu Item Group Sales Detail (FG_S002.rpt)
Consolidated SYS Menu Item Group Sales Detail, Subtotaled By Category (FG_S012.rpt)
Consolidated System Menu Item Group Sales Detail with Graph (FG_S202.rpt)
System Menu Item Group Sales Summary (FG_S101.rpt)
Consolidated System Menu Item Group Sales Summary (FG_S111.rpt)

v_R_sys_menuitem

The v_R_sys_menuitem view may be used to create a System Menu Item report. This view is not used by any of the standard 3700 reports.

Tables/Columns

The tables and columns used to create the v_R_sys_menuitem view are listed below:

Tables	Columns	
Daily System Menu Item Total (dly_sys_mi_ttl)	business_date price1_sls_ttl price1_dsc_ttl price2_sls_ttl price2_dsc_ttl price3_sls_ttll price3_dsc_ttl price4_sls_ttl price4_dsc_ttl	price1_sls_cnt price1_rtn_cnt price2_sls_cnt price2_rtn_cnt price3_sls_cnt price3_rtn_cnt price4_sls_cnt price4_rtn_cnt
Menu Item Definition (mi_def)	obj_num mi_seq name_1	nlu_num name_2
Major Group Definition (maj_grp_def)	obj_num	name
Family Group Definition (fam_grp_def)	obj_num	name
Menu Item Group Definition (mi_grp_def)	obj_num	name
Menu Item Price Definition (mi_price_def)	preset_amt_1 preset_amt_2	preset_amt_3 preset_amt_4
Menu Item Type Class Definition (mi_type_class_def)	ob_mi20_sub_lvl_for_prices	
Daily System Total (dly_sys_ttl)	sys_sbtl_discount_total	
Revenue Center Configuration Temp (rvc_cfg_temp)	main_mlvl_name_1 sub_mlvl_1_name main_mlvl_2_name sub_mlvl_2_name	main_mlvl_3_name sub_mlvl_3_name main_mlvl_4_name sub_mlvl_5_name
Restaurant (rest_def)	obj_num	store_id

Stored Procedure

sp_R_menuitem

Reports

This view is not used by any of the standard 3700 reports.

v_R_sys_menuitem_fam_grp

The v_R_sys_menuitem_fam_grp view provides system-wide detailed sales information for each of the menu items in a family group.

Tables/Columns

The tables and columns used to create the v_R_sys_menuitem_fam_grp view are listed below:

Tables	Columns	
Daily System Menu Item Total (dly_sys_mi_ttl)	business_date price1_sls_ttl price1_dsc_tal price2_sls_ttl price2_dsc_ttl price3_sls_ttl price3_dsc_ttl price4_sls_ttl price4_dsc_ttl	price1_sls_cnt price1_rtn_cnt price2_sls_cnt price2_rtn_cnt price3_sls_cnt price3_rtn_cnt price4_sls_cnt price4_rtn_cnt
Menu Item Definition (mi_def)	obj_number mi_seq nlu_num	name_1 name_2
Family Group Definition (fam_grp_def)	obj_num	name
Menu Item Price Definition (mi_price_def)	preset_amt_1 preset_amt_2	preset_amt_3 preset_amt_4
Menu Item Type Class Definition (mi_type_class_def)	ob_mi20_sub_lvl_for_prices	
Daily System Totals (dly_sys_ttl)	sttl_dsc_ttl	
Revenue Center Configuration Temp (rvc_cfg_temp)	main_mlvl_1_name sub_mlvl_1_name main_mlvl_2_name sub_mlvl_2_name	main_mlvl_3_name sub_mlvl_3_name main_mlvl_4_name sub_mlvl_4_name
Restaurant (rest_def)	obj_num	store_id

Stored Procedure

sp_R_menuitem

Reports

The v_R_sys_menuitem_fam_grp view is used to create the following reports:

System Menu Item Detail, Subtotaled By Menu Item Group (MI_S001.rpt)
Consolidated System Menu Item Detail, Subtotaled By Menu Item Group (MI_S002.rpt)
System Menu Item Summary, Subtotaled By Menu Item Group (MI_S101.rpt)
Consolidated System Menu Item Summary, Subtotaled By Menu Item Group

v_R_sys_menuitem_maj_grp

The v_R_sys_menuitem_maj_grp view provides system-wide detailed sales information for each of the menu items in a major group.

Tables/Columns

The tables and columns used to create the v_R_sys_menuitem_maj_grp view are listed below:

Tables	Columns	
Daily System Menu Item Total (dly_sys_mi_ttl)	business_date price1_slss_ttl price1_dsc_ttl price2_sls_ttl price2_dsc_ttl price3_sls_ttl price3_dsc_ttl price4_sls_ttl price4_dsc_ttl	price1_sls_cnt price1_rtn_cnt price2_sls_cnt price2_rtn_cnt price3_sls_cnt price3_rtn_cnt price4_sls_cnt price4_rtn_cnt
Menu Item Definition (mi_def)	obj_num mi_seq nlu_num	name_1 name_2
Major Group Definition (maj_grp_def)	obj_num	name
Menu Item Price Definition (mi_price_def)	preset_amt_1 preset_amt_2	preset_amt_3 preset_amt_4
Menu Item Type Class Definition (mi_type_class_def)	ob_mi20_sub_lvl_for_prices	
Daily System Totals (dly_sys_ttl)	sttl_dsc_ttl	
Revenue Center Configuration Temp (rvc_cfg_temp)	main_mlvl_1_name sub_mlvl_1_name main_mlvl_2_name sub_mlvl_2_name	main_mlvl_3_name sub_mlvl_3_name main_mlvl_4_name sub_mlvl_4_name
Restaurant (rest_def)	obj_num	store_id

Stored Procedure

sp_R_menuitem

Reports

The v_R_sys_menuitem_maj_grp view is used to create the following reports:

System Menu Item Detail, Subtotaled By Major Group (MI_S001.rpt)
Consolidated System Menu Item Detail, Subtotaled By Major Group (MI_S002.rpt)
System Menu Item Summary, Subtotaled By Major Group (MI_S101.rpt)
Consolidated SYS Menu Item Summary, Subtotaled By Major Group (MI_S102.rpt)

v_R_sys_menuitem_mi_grp

The v_R_sys_menuitem_mi_grp view provides system-wide detailed sales information for each of the menu items in a menu item group.

Tables/Columns

The tables and columns used to create the v_R_sys_menuitem_mi_grp view are listed below:

Tables	Columns	
Daily System Menu Item Total (dly_sys_mi_ttl)	business_date price1_slss_ttl price1_dsc_ttl price2_sls_ttl price2_dsc_ttl price3_sls_ttl price3_dsc_ttl price4_sls_ttl price4_dsc_ttl	price1_sls_cnt price1_rtn_cnt price2_sls_cnt price2_rtn_cnt price3_sls_cnt price3_rtn_cnt price4_sls_cnt price4_rtn_cnt
Menu Item Definition (mi_def)	obj_num mi_seq nlu_num	name_1 name_2
Menu Item Group Definition (mi_grp_def)	obj_num	name
Menu Item Price Definition (mi_price_def)	preset_amt_1 preset_amt_2	preset_amt_3 preset_amt_4
Menu Item Type Class Definition (mi_type_class_def)	ob_mi20_sub_lvl_for_prices	
Daily System Totals (dly_sys_ttl)	sttl_dsc_ttl	
Revenue Center Configuration Temp (rvc_cfg_temp)	main_mlvl_1_name sub_mlvl_1_name main_mlvl_2_name sub_mlvl_2_name	main_mlvl_3_name sub_mlvl_3_name main_mlvl_4_name sub_mlvl_4_name
Restaurant (rest_def)	obj_num	store_id

Stored Procedure

sp_R_menuitem

Reports

The v_R_sys_menuitem_mi_grp view is used to create the following reports:

System Menu Item Detail, Subtotaled By Menu Item Group (MI_S001.rpt)
Consolidated SYS Menu Item Detail, Subtotaled By Menu Item Group (MI_S002.rpt)
System Menu Item Summary, Subtotaled By Menu Item Group (MI_S101.rpt)
Consolidated SYS Menu Item Summary, Subtotaled By Menu Item Group (MI_S102.rpt)

v_R_sys_sales

The v_R_sys_sales view provides system-wide sales and tip information, such as net sales and charged tips.

Tables/Columns

The tables and columns used to create the v_R_sys_sales view are listed below:

Tables	Columns	
Daily System Total (dly_sys_ttl)	business_date_start_time business_date_end_time item_dsc_ttl rtn_cnt svc_chg_ttl void_ttl credit_ttl change_grand_ttl grand_total training_total cancel_cnt mgr_void_cnt error_correct_cnt chk_carry_over_cnt chk_begun_cnt chk_paid_cnt gross_rcpts_ttl chgd_tips_ttl tips_paid_ttl net_sls_ttl sttl_dsc_ttl rtn_ttl void_cnt tax_coll_ttl rounding_ttl non_txbl_ttl	cancel_ttl mgr_void_ttl error_correct_ttl chk_carry_over_ttl chk_begun_ttl chk_paid_ttl chgd_rcpts_ttl tip_svc_ttl tips_declared_ttl labor_cat_1_reg_hrs labor_cat_1_ovt_hrs labor_cat_1_reg_ttl labor_cat_1_ovt_ttl labor_cat_2_reg_hrs labor_cat_2_ovt_hrs labor_cat_2_reg_ttl labor_cat_2_ovt_ttl labor_cat_3_reg_hrs labor_cat_3_ovt_hrs labor_cat_3_reg_ttl labor_cat_3_ovt_ttl labor_cat_4_reg_hrs labor_cat_4_ovt_hrs labor_cat_4_reg_ttl labor_cat_4_ovt_ttl
Daily System Order Type Totals (dly_sys_ot_ttl)	order_type_seq net_sls_ttl chk_cnt	cov_cnt tbl_turn_cnt tbl_dine_time
Order Type (order_type_def)	name	
Restaurant (rest_def)	obj_num	store_id

Stored Procedure

sp_R_sys_totals
sp_R_time_card_ttls

Reports

The v_R_sys_sales view is used to create the following reports:

System Sales Detail (SYS_101.rpt)
System Sales Detail (SYS_101_1.rpt)
System Sales Detail (SYS_101_2.rpt)
System Sales Detail (SYS_101_3.rpt)

v_R_sys_sales_tips

The v_R_sys_sales_tips provides system-wide sales and tip information, such as net sales and charged tips.

Tables/Columns

The tables and columns used to create the v_R_sys_sales_tips view are listed below:

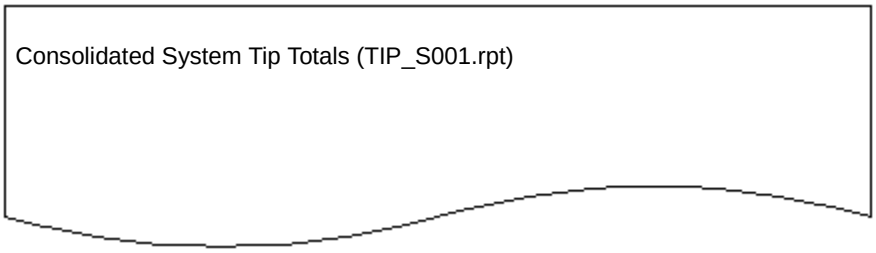
Tables	Columns	
Daily System Total (dly_sys_ttl)	business_date_start_time business_date_end_time item_dsc_ttl rtn_cnt svc_chg_ttl void_ttl credit_ttl change_grand_ttl grand_total training_total cancel_cnt mgr_void_cnt error_correct_cnt chk_carry_over_cnt chk_begun_cnt chk_paid_cnt gross_rcpts_ttl chgd_tips_ttl tips_paid_ttl net_sls_ttl sttl_dsc_ttl rtn_ttl void_cnt tax_coll_ttl rounding_ttl non_txbl_ttl	cancel_ttl mgr_void_ttl error_correct_ttl chk_carry_over_ttl chk_begun_ttl chk_paid_ttl chgd_rcpts_ttl tip_svc_ttl tips_declared_ttl labor_cat_1_reg_hrs labor_cat_1_ovt_hrs labor_cat_1_reg_ttl labor_cat_1_ovt_ttl labor_cat_2_reg_hrs labor_cat_2_ovt_hrs labor_cat_2_reg_ttl labor_cat_2_ovt_ttl labor_cat_3_reg_hrs labor_cat_3_ovt_hrs labor_cat_3_reg_ttl labor_cat_3_ovt_ttl labor_cat_4_reg_hrs labor_cat_4_ovt_hrs labor_cat_4_reg_ttl labor_cat_4_ovt_ttl
Restaurant (rest_def)	obj_num	store_id

Stored Procedure

sp_R_sys_totals
sp_R_time_card_ttls

Reports

The v_R_sys_sales_tips view is used to create the following reports:



v_R_sys_tax

The v_R_sys_tax view provides system-wide tax information.

Tables/Columns

The tables and columns used to create the v_R_sys_tax view are listed below:

Tables	Columns	
Daily System Tax Total (dly_sys_tax_ttl)	coll_ttl exempt_ttl	tax_rate_seq txbl_sls_ttl
Tax Rate Definition (tax_rate_def)	type percentage txbl_sls_name net_sls_name	name tax_coll_name exempt_name
Daily System Totals (dly_sys_ttl)	business_date net_sls_ttl	non_txbl_sls
Restaurant (rest_def)	obj_num	store_id

Stored Procedure

sp_R_tax

Reports

The v_R_sys_tax view is used to create the following reports:

System Tax Totals (TAX_S001.rpt)
Consolidated System Tax Totals (TAX_S002.rpt)
Canadian System Tax Totals(TAX_cs01.rpt)
Canadian Consolidated System Tax Totals (TAX_cs02.rpt)

v_R_sys_time_period

The v_R_sys_time_period view provides system-wide financial information based on time periods.

Tables/Columns

The tables and columns used to create the v_R_sys_time_period view are listed below:

Tables	Columns	
Daily System Fixed Period Total (dly_sys_fixed_prd_ttl)	business_date store_number store_id time_period_number time_period_seq period_number period_name period_start_time period_end_time labor_cat_1_reg_hrs labor_cat_1_reg_ttl labor_cat_2_reg_hrs labor_cat_2_reg_ttl labor_cat_3_reg_hrs labor_cat_3_reg_ttl labor_cat_4_reg_hrs labor_cat_4_reg_ttl	net_sales_ttl labor_cat_1_ovt_hrs labor_cat_1_ovt_ttl labor_cat_2_ovt_hrs labor_cat_2_ovt_ttl labor_cat_3_ovt_hrs labor_cat_3_ovt_ttl labor_cat_4_ovt_hrs labor_cat_4_ovt_ttl
Daily System Fixed Period Order Type Total (dly_sys_fixed_prd_ot_ttl)	order_type_seq	
Order Type (order_type_def)	name	

Stored Procedure

sp_R_time_period
sp_R_sys_time_prd_time_card_ttls

Reports

The v_R_sys_time_period view is used to create the following reports:

System Time Period Totals (TP_S001.rpt)
System Time Period Summary (TP_S103.rpt)
System Time Period Detail (TP_S101.rpt)

v_R_sys_time_period_trk

The v_R_sys_time_period_trk view provides system-wide financial information based on time periods.

Tables/Columns

The tables and columns used to create the v_R_sys_time_period_trk view are listed below:

Tables	Columns	
Tracking Group Definition (trk_grp_def)	obj_num	"name"
Tracking Totals Definition Temp (trk_ttl_def_temp)	ttl_name_01 *to* ttl_name_64	
Daily System Time Period Tracking Totals (dly_sys_fixed_prd_trk_ttl)	business_date	
Time Period Definition (time_period_def)	time_period_seq obj_num	name
Period Definition (period_def)	obj_num start_time	name end_time
Restaurant (rest_def)	obj_num	name

Stored Procedure

sp_R_sys_time_prd_sales_trk
sp_R_time_period_tracking

Reports

The v_R_sys_time_period_trk view is used to create the following reports:

System Time Period Detail (TP_S101.rpt)

v_R_sys_trk

The v_R_sys_trk view may be used to create a System Tracking Group report.

Tables/Columns

The tables and columns used to create the v_R_sys_trk view are listed below:

Tables	Columns
Daily System Tracking Total (dly_sys_trk_ttl)	business_date
Tracking Group Definition (trk_grp_def)	obj_num name
Tracking Total Definition Temp (trk_ttl_def_temp)	ttl_name_01 *to* ttl_name_64
Restaurant (rest_def)	obj_num name

Stored Procedure

sp_R_sys_sales_tracking

Reports

The v_R_sys_trk view is used to create the following reports:

System Sales Detail (SYS_101.rpt)
(SYS_101_2.rpt)

v_R_time_card_adj

The v_R_time_card_adj view is used by the v_R_employee_time_card view. The v_R_time_card_adj view is not used by any of the standard 3700 reports.

Tables/Columns

The tables and columns used to create the v_R_time_card_adj view are listed below:

Tables	Columns	
Time Card Adjust Detail (time_card_adjust_dtl)	tm_card_seq	tm_card_adjmt_seq
Employee Definition (emp_def)	last_name	first_name
Reason Definition (reason_def)	name	

Stored Procedure

This view does not require a stored procedure.

Reports

This view is not used by any of the standard 3700 reports.

Chapter 7

Stored Procedures

This chapter provides an introduction to stored procedures and describes how they are used in the 3700 environment.

In this chapter

What are Stored Procedures?

Stored procedures are collections of SQL statements that are precompiled and stored in the database.

Stored procedures are used in the 3700 system to retrieve, process, and update data in the database tables and are used internally by the 3700 system.

MICROS will set all standard reports and stored procedures at object number 9999 and below. Any custom reports or stored procedures must be given an object number above 9999. This will ensure that MICROS can add changes to reports without overwriting a custom report. For more information, see "Adding Changes to the MICROS Database" on page 5-14.

Using Stored Procedures

In order to use stored procedures, you need to understand how to do the following:

- Create stored procedures
- Drop or remove stored procedures
- Call stored procedures from a database application

Creating Stored Procedures

Stored procedures are created using the CREATE PROCEDURE statement. Note that you must have RESOURCE authority in order to create a procedure.

Refer to the *Sybase SQL Anywhere User's Guide* for the syntax of the CREATE PROCEDURE statement.

Dropping or Removing Stored Procedures

Once a custom procedure is created, it remains in the database until it is explicitly removed. It is recommended that custom stored procedures be saved in a text file. A database conversion will not preserve any custom objects. Note that only the owner of the procedure can drop the procedure from the database.

Calling Stored Procedures

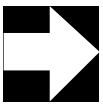
A stored procedure can be invoked with a CALL statement, by an application program, or by other procedures. Refer to the *Sybase SQL Anywhere User's Guide* for the syntax of the CALL statement.

3700 Stored Procedures

The 3700 Stored Procedures are used internally by the 3700 system to update totals that are used to generate reports. Some stored procedures are called through autosequences.

Stored Procedures Used to Generate Reports

The following stored procedures are used to update the totals that are used in generating reports.



Note

There is no need to call several stored procedures if they update the same tables. You need only call the one with the most table updates.

.

Stored Procedure	Purpose	Tables Read	Tables Updated
sp_R_cashier	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_cshr_cum and sp_post_cshr_ot_cum.		shift_cshr_ttl shift_cshr_ot_ttl
sp_R_cashier_sales_tracking	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_cshr_cum, sp_post_cshr_ot_cum, and sp_post_cshr_trk_cum.		shift_cshr_ttl shift_cshr_ot_ttl shift_cshr_trk_ttl
sp_R_cashier_tracking	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_cshr_trk_cum.		shift_cshr_trk_ttl
sp_R_cons_cshr_ttls	Consolidates totals for a range of business days and returns a result set.	shift_cshr_ot_ttl shift_cshr_ttl cshr_ttl_def shift_cshr_trk_ttl	
sp_R_cons_emp_ttls	Consolidates totals for a range of business days and returns a result set.	shift_emp_ot_ttl shift_emp_ttl shift_emp_trk_ttl emp_def	
sp_R_cons_rvc_cashier_ttls	Consolidates totals for a Revenue Center for a range of business days and returns a result set.	shift_rvc_cshr_ttl shift_rvc_cshr_ot_ttl shift_rvc_cshr_trk_ttl	
sp_R_cons_rvc_emp_ttls	Consolidates totals for a Revenue Center for a range of business dates and returns a set.	shift_rvc_emp_ttl shift_rvc_emp_ot_ttl shift_rvc_emp_trk_ttl	

Stored Procedure	Purpose	Tables Read	Tables Updated
sp_R_cons_rvc_time_prd_ttls	Consolidates totals for a range of business days and returns a result set.	dly_rvc_fixed_prd_ot_ttl dly_rvc_fixed_prd_ttl dly_rvc_fixed_prd_trk_ttl time_period_def period_def	
sp_R_cons_rvc_ttls	Consolidates totals for a range of business days and returns a result set.	dly_rvc_ot_ttl dly_rvc_ttl dly_rvc_trk_ttl	
sp_R_cons_sys_time_prd_ttls	Consolidates totals for the system for a range of business dates and returns a result set.	dly_rvc_fixed_prd_ot_ttl dly_rvc_fixed_prd_ttl dly_rvc_fixed_prd_trk_ttl	
sp_R_cons_sys_ttls	Consolidates totals for the system for a range of business dates and returns a result set.	dly_sys_ttl dly_sys_ot_ttl dly_sys_trk_ttl	
sp_R_employee	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_emp_cum and sp_post_emp_ot_cum.		shift_emp_ttl shift_emp_ot_ttl
sp_R_employee_job_code	Calls sp_post_emp_job_code_labor		dly_emp_job_code_ttl dly_job_code_ttl
sp_R_employee_sales_tracking	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_emp_cum, sp_post_emp_ot_cum, sp_post_transpose_trk_ttl, and sp_post_emp_trk_cum.		shift_emp_ttl shift_emp_ot_ttl shift_emp_trk_ttl
sp_R_employee_tracking	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_emp_trk_cum.		shift_emp_trk_ttl
	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_fam_grp_cum.		dly_rvc_fam_grp_ttl dly_rvc_ttl dly_sys_fam_grp_ttl dly_sys_ttl
sp_R_job_code_totals	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_job_code_sales. This stored procedure also calls sp_post_time_card_dtl and sp_post_job_code_labor.		dly_emp_job_code_ttl dly_job_code_ttl
sp_R_major_group	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_maj_grp_cum.		dly_rvc_maj_grp_ttl dly_rvc_ttl dly_sys_maj_grp_ttl dly_sys_ttl

Stored Procedure	Purpose	Tables Read	Tables Updated
sp_R_menuitem	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_mi_grp_cum, sp_post_cum, and sp_post_mi_cum.		dly_rvc_mi_ttl dly_rvc_ttl dly_sys_mi_ttl dly_sys_ttl
sp_R_menuitem_group	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_mi_grp_cum.		dly_rvc_mi_grp_ttl dly_rvc_ttl dly_sys_mi_grp_ttl dly_sys_ttl
sp_R_rvc_cashier	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_rvc_cshr_cum and sp_post_rvc_cshr_ot_cum.		shift_rvc_cshr_ttl shift_rvc_cshr_ot_ttl
sp_R_rvc_cashier_sales_tracking	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_rvc_cshr_cum, sp_post_rvc_cshr_ot_cum, sp_post_rvc_cshr_trk_cum, and sp_transpose_trk_ttl_names.		shift_rvc_cshr_ttl shift_rvc_cshr_ot_ttl shift_rvc_cshr_trk_ttl
sp_R_rvc_cashier_tracking	Calls the sp_post stored procedure with parameters that specify for it to call sp_transpose_trk_ttl_names and sp_post_rvc_cshr_trk_cum.		shift_rvc_cshr_trk_ttl
sp_R_rvc_employee	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_rvc_emp_cum and sp_post_rvc_emp_ot_cum.		shift_rvc_emp_ttl shift_rvc_emp_ot_ttl
sp_R_rvc_employee_sales_trk	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_rvc_emp_cum, sp_post_rvc_emp_ot_cum, sp_transpose_trk_ttl_names, and sp_post_rvc_emp_trk_cum.		shift_rvc_emp_ttl shift_rvc_emp_ot_ttl shift_rvc_emp_trk_ttl
sp_R_rvc_employee_tracking	Calls the sp_post stored procedure with parameters that specify for it to call sp_transpose_trk_ttl_names and sp_post_rvc_emp_trk_cum.		shift_rvc_emp_trk_ttl
sp_R_rvc_sales_tracking	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_cum, sp_post_ot_cum, and sp_post_rvc_trk_cum.		dly_rvc_ttl dly_rvc_ot_ttl dly_sys_ttl dly_sys_ot_ttl dly_rvc_trk_ttl

Stored Procedure	Purpose	Tables Read	Tables Updated
sp_R_rvc_time_prd_sales_trk	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_tm_prd_cum, sp_post_rvc_tm_prd_trk_cum, and sp_post_tm_prd_ot_cum.		dly_rvc_fixed_prd_ttl dly_sys_fixed_prd_ttl dly_rvc_fixed_prd_ot_ttl dly_sys_fixed_prd_ot_ttl dly_rvc_fixed_prd_trk_ttl
sp_R_rvc_time_period_tracking	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_rvc_tm_prd_trk_cum.		dly_rvc_fixed_prd_trk_ttl
sp_R_rvc_totals	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_cum.		dly_rvc_ttl dly_rvc_ot_ttl dly_sys_ttl dly_sys_ot_ttl
sp_R_rvc_tracking	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_rvc_trk_cum.		dly_rvc_trk_ttl
sp_R_sys_sales_tracking	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_cum, sp_post_ot_cum, and sp_post_sys_trk_cum.		dly_rvc_ot_ttl dly_sys_ot_ttl dly_rvc_ttl dly_sys_ttl dly_sys_trk_ttl
sp_R_sys_time_prd_sales_trk	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_tm_prd_cum, sp_post_rvc_tm_prd_trk_cum, and sp_post_sys_tm_prd_trk_cum.		dly_rvc_fixed_prd_ttl dly_sys_fixed_prd_ttl dly_rvc_fixed_prd_ot_ttl dly_sys_fixed_prd_ot_ttl dly_sys_fixed_prd_trk_ttl
sp_R_sys_time_period_tracking	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_sys_tm_prd_trk_cum.		dly_sys_fixed_prd_trk_ttl
sp_R_sys_totals	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_cum.		dly_sys_ttl dly_rvc_ttl dly_sys_ot_ttl dly_rvc_ot_ttl
sp_R_sys_tracking	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_sys_trk_cum.		dly_sys_trk_ttl
sp_R_tax	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_tax_cum.		dly_rvc_tax_ttl dly_rvc_ttl dly_sys_tax_ttl dly_sys_ttl
sp_R_time_card_detail	Calls sp_post_emp_time_card_dtl with NULL parameter to post all employees.		time_card_dtl

Stored Procedure	Purpose	Tables Read	Tables Updated
sp_R_time_card_totals	Posts transaction totals to the dly_rvc_ttl and dly_sys_ttl tables.	time_card_dtl job_def	dly_rvc_ttl dly_sys_ttl
sp_R_time_period	Calls the sp_post stored procedure with parameters that specify for it to call sp_post_tm_prd_cum and sp_post_tm_prd_ot_cum.		dly_rvc_fixed_prd_ttl dly_sys_fixed_prd_ttl
sp_R_time_prd_time_card_ttls		time_card_dtl time_period_def period_def job_def	dly_rvc_fixed_prd_ttl dly_sys_fixed_prd_ttl

Stored Procedures available through Autosequences

The following stored procedures can be called through an autosequence.

Clock Out an Employee sp_ClockOutEmployee

This stored procedure is usually run as part of an end of night autosequence. It forces a clock out of all clocked in employees. It can be used with the Employee Clocked In stored procedure.

Tables Read

- ❑ time_card_dtl
- ❑ emp_status

Tables Updated

- ❑ time_card_dtl
- ❑ emp_status

Related Reports

There are no related reports.

Clock Out and In All Employees sp_ClockOutInAllEmployees

This stored procedure is usually run as part of an end of night autosequence when employees are on the clock and end of night is run. It clocks out all employees and clocks them back in. This stored procedure is a good way to split employees' payroll across two pay periods.

Tables Read

- ❑ time_card_dtl
- ❑ emp_status

Tables Updated

- ❑ time_card_dtl

- ❑ emp_status

Related Reports
There are no related reports

Clock Out and In an Employee sp_ClockOutInEmployee

This stored procedure is usually run as part of an end of night autosequence when employees are on the clock and end of night is run. It clocks out an employee and clocks the employee back in. This stored procedure is a good way to split an employee's payroll across two pay periods.

Tables Read

- ❑ time_card_dtl
- ❑ emp_status

Tables Updated

- ❑ time_card_dtl
- ❑ emp_status

Related Reports
There are no related reports.

Credit Card Batch Create sp_CreateCCBatch

This stored procedure will create a batch of all the unbatched credit card transactions in the system. This is typically run once a day as part of the end of night autosequence.

Tables Read

- ❑ cc_index_dtl
- ❑ caedc_driver_def
- ❑ tmed_dtl
- ❑ rest_status
- ❑ ccbatch_item_dtl

Tables Updated

- ❑ rest_status
- ❑ cc_batch_item_dtl
- ❑ chk_dtl

Related Reports
There are no related reports.

Employee Clocked In
sp_EmployeeClockedIn

This stored procedure is usually run as part of the end of night autosequence. The procedure checks to see if any employees are clocked in. If employees are clocked in, the stored procedure branches to print a status report and can be programmed to clock out all clocked in employees using the Clock Out an Employee stored procedure.

Tables Read

- emp_status

Tables Updated

This stored procedure does not update any tables.

Related Reports

There are no related reports.

Employee Open Checks
sp_EmployeeOpenChecks

This stored procedure is usually run as part of the end of night autosequence to check for open employee checks. If there are open checks, the stored procedure can be programmed to branch to another step such as printing an open check report. If there are no open checks the stored procedure will complete without any errors.

Tables Read

- chk_dtl

Tables Updated

This stored procedure does not update any tables

Related Reports

There are no related reports.

Increment Cashier Shift
sp_IncrementCshrShift

This procedure can be run as part of an end of shift, end of night, or Clock In autosequence. This stored procedure will increment a cashier's shift. Once the shift is incremented all subsequent transactions will be posted to the new shift.

Tables Read

- chsr_ttl_def
- cshr_ttl_status

Tables Updated

- shift_cshr_dtl
- shift_cshr_ttl

- ❑ cshr_ttl_status

Related Reports
There are no related reports

Increment Employee Shift
sp_IncrementEmpShift

This procedure will increment an employee's shift. It is usually called as part of an employee's close out report for the end of a shift. After the shift is incremented all subsequent transactions for the employee will be posted to the new shift.

Tables Read

- ❑ emp_def
- ❑ emp_status

Tables Updated

- ❑ shift_emp_dtl
- ❑ shift_emp_ttl
- ❑ emp_status
- ❑ rvc_sign_in_status

Related Reports
There are no related reports.

Post Cashier Sales
sp_R_cashier

This stored procedure can be used to post totals throughout the day so related reports will have a quick run time.

This stored procedure calls the sp_post stored procedure with parameters that specify for it to call sp_post_cshr_cum and sp_post_cshr_ot_cum.

Tables Updated

- ❑ shift_cshr_ttl
- ❑ shift_cshr_ot_ttl

Related Reports

- ❑ Cashier Summary Totals (CSHR_001.RPT)

Post Cashier Sales and Tracking
sp_R_cashier_sales_tracking

This stored procedure can be used to post totals throughout the day so related reports will have a quick run time.

This stored procedure calls the sp_post stored procedure with parameters that specify for it to call sp_post_cshr_cum, sp_post_cshr_ot_cum, and sp_post_cshr_trk_cum.

Tables Updated

- shift_cshr_ttl
- shift_cshr_ot_ttl
- shift_cshr_trk_ttl

Related Reports

- Cashier Detail Totals (CSHR_101.RPT)

Post Cashier Sales and Tracking by Revenue Center sp_R_rvc_cashier_sales_tracking

This stored procedure can be used to post totals throughout the day so related reports will have a quick run time.

This stored procedure calls the sp_post stored procedure with parameters that specify for it to call sp_post_rvc_cshr_cum, sp_post_rvc_cshr_ot_cum, sp_post_rvc_cshr_trk_cum, and sp_transpose_trk_ttl_names.

Tables Updated

- shift_rvc_cshr_ttl
- shift_rvc_cshr_ot_ttl
- shift_rvc_cshr_trk_ttl

Related Reports

- Cashier Detail Totals by RVC (CSHR_701)
- Consolidated Cashier Detail Totals (CSHR_702)

Post Cashier Sales by Revenue Center sp_R_rvc_cashier

This stored procedure can be used to post totals throughout the day so related reports will have a quick run time.

This stored procedure calls the sp_post stored procedure with parameters that specify for it to call sp_post_rvc_cshr_cum and sp_post_rvc_cshr_ot_cum.

Tables Updated

- shift_rvc_cshr_ttl
- shift_rvc_cshr_ot_ttl

Related Reports

- Cashier Summary Totals by RVC (CSHR_601.RPT)

Post Employee Job Code Totals sp_R_employee_job_code

This stored procedure can be used to post totals throughout the day so related reports will have a quick run time.

Tables Read

- sp_post_emp_job_code_labor

Tables Updated

- dly_emp_job_code_ttl
- dly_job_code_ttl

Related Reports

- Employee Job Summary (Time_003.RPT)
- Consolidated Employee Job Summary (Time_004.RPT)

Post Employee Sales and Tracking sp_R_employee_sales_tracking

This stored procedure can be used to post totals throughout the day so related reports will have a quick run time.

This stored procedure calls the sp_post stored procedure with parameters that specify for it to call sp_post_emp_cum, sp_post_emp_ot_cum, sp_post_transpose_trk_ttl, and sp_post_emp_trk_cum.

Tables Updated

- shift_emp_ttl
- shift_emp_ot_ttl
- shift_emp_trk_ttl

Related Reports

- Employee Detail Totals (EMP_101.RPT)
- Consolidated Employee Detail Totals (EMP_102.RPT)

Post Employee Sales and Tracking by Revenue Center sp_R_rvc_employee_sales_tracking

This stored procedure can be used to post totals throughout the day so related reports will have a quick run time.

This stored procedure calls the sp_post stored procedure with parameters that specify for it to call sp_post_rvc_emp_cum, sp_post_rvc_ot_emp_cum, sp_transpose_trk_ttl_names, and sp_post_rvc_emp_trk_cum.

Tables Updated

- shift_rvc_emp_ttl
- shift_rvc_emp_ot_ttl

- ❑ shift_rvc_emp_trk_ttl

Related Reports

- ❑ Employee Detail Totals by Revenue Center (EMP_701.RPT)
- ❑ Consolidated Employee Detail Totals by Revenue Center (EMP_702.RPT)

Post Employee Sales by Revenue Center sp_R_rvc_employee

This stored procedure can be used to post totals throughout the day so related reports will have a quick run time.

This stored procedure calls the sp_post stored procedure with parameters that specify for it to call sp_post_rvc_emp_cum and sp_post_rvc_emp_ot_cum.

Tables Updated

- ❑ shift_rvc_emp_ttl
- ❑ shift_rvc_emp_ot_ttl

Related Reports

- ❑ Employee Tip Totals by Revenue Center (EMP_805.RPT)
- ❑ Consolidated Employee Tip Totals by Revenue Center (EMP_806.RPT)

Post Family Group Sales sp_R_family_group

This stored procedure can be used to post totals throughout the day so related reports will have a quick run time.

This stored procedure calls the sp_post stored procedure with parameters that specify for it to call sp_post_fam_grp_cum.

Tables Updated

- ❑ dly_rvc_fam_grp_ttl
- ❑ dly_rvc_ttl
- ❑ dly_sys_fam_grp_ttl
- ❑ dly_sys_ttl

Related Reports

- ❑ Daily RVC Group Sales Detail Subtotal by Category (FG_R010.RPT)
- ❑ Consolidated RVC Group Sales Detail Subtotal by Category (FG_R012.RPT)
- ❑ Daily SYS Group Sales Detail Subtotal by Category (FG_S010.RPT)
- ❑ Consolidated System Group Sales Detail Subtotal by Category (FG_S012.RPT)

- Daily RVC Group Sales Detail (FG_R001.RPT)
- Consolidated RVC Group Sales Detail (FG_R002.RPT)
- Daily RVC Group Sales Summary (FG_R101.RPT)
- Consolidated RVC Group Sales Summary (FG_R111.RPT)
- Daily System Group Sales Summary (FG_S101.RPT)
- Consolidated System Group Sales Summary (FG_S111.RPT)
- Daily SYS Group Sales Detail (FG_S001.RPT)
- Consolidated System Group Sales Detail (FG_S002.RPT)
- Consolidated System Group Sales Detail w/Graph (FG_S202.RPT)

Post Job Code Totals
sp_R_job_code_totals

This stored procedure can be used to post totals throughout the day so related reports will have a quick run time.

This stored procedure calls the sp_post stored procedure with parameters that specify for it to call sp_post_job_code_sales, sp_post_time_card_dtl, and sp_post_job_code_labor.

Tables Updated

- dly_emp_job_code_ttl
- dly_job_code_ttl

Related Reports

- Revenue Center Job Summary (Time_R011.RPT)
- Consolidated Revenue Center Job Summary (Time_R012.RPT)
- System Job Summary (Time_S011.RPT)
- Consolidated System Job Summary (Time_S012.RPT)

Post Major Group Sales
sp_R_major_group

This stored procedure can be used to post totals throughout the day so related reports will have a quick run time.

This stored procedure calls the sp_post stored procedure with parameters that specify for it to call sp_post_maj_grp_cum.

Tables Updated

- dly_rvc_maj_grp_ttl
- dly_rvc_ttl
- dly_sys_maj_grp_ttl
- dly_sys_ttl

Related Reports

This stored procedure uses the family group reports.

Post Menuitem Group Sales
sp_R_menuitem_group

This stored procedure can be used to post totals throughout the day so related reports will have a quick run time.

This stored procedure calls the sp_post stored procedure with parameters that specify for it to call sp_post_mi_grp_cum.

Tables Updated

- dly_rvc_mi_grp_ttl
- dly_rvc_ttl
- dly_sys_mi_grp_ttl
- dly_sys_ttl

Related Reports

This stored procedure uses the family group reports.

Post Menuitem Sales
sp_R_menuitem

This stored procedure can be used to post totals throughout the day so related reports will have a quick run time.

This stored procedure calls the sp_post stored procedure with parameters that specify for it to call sp_post_mi_grp_cum, sp_post_cum, and sp_post_mi_cum.

Tables Updated

- dly_rvc_mi_ttl
- dly_rvc_ttl
- dly_sys_mi_ttl
- dly_sys_ttl

Related Reports

- Daily Revenue Center Menu Item Sales Detail (MI_R001.RPT)
- Consolidated Revenue Center Menu Item Sales Detail (MI_R002.RPT)
- Daily Revenue Center Menu Item Sales Summary (MI_R101.RPT)
- Consolidated Revenue Center Menu Item Sales Summary (MI_R102.RPT)
- Daily System Menu Item Sales Detail (MI_S001.RPT)
- Consolidated System Menu Item Sales Detail (MI_S002.RPT)
- Daily System Menu Item Sales Summary (MI_S101.RPT)

- Consolidated System Menu Item Sales Summary (MI_S102.RPT)

Post RVC Sales and Tracking
sp_R_rvc_sales_tracking

This stored procedure can be used to post totals throughout the day so related reports will have a quick run time.

This stored procedure calls the sp_post stored procedure with parameters that specify for it to call sp_post_cum, sp_post_ot_cum and sp_post_rvc_trk_cum.

Tables Updated

- dly_rvc_ttl
- dly_rvc_ot_ttl
- dly_sys_ttl
- dly_sys_ot_ttl
- dly_rvc_trk_ttl

Related Reports

- Daily Revenue Center Sales Detail (RVC_101.RPT)
- Consolidated Revenue Center Sales Detail (RVC_102.RPT)

Post RVC Time Period Sales and Tracking
sp_R_rvc_time_prd_sales_trk

This stored procedure can be used to post totals throughout the day so related reports will have a quick run time.

This stored procedure calls the sp_post stored procedure with parameters that specify for it to call sp_post_tm_prd_cum, sp_post_rvc_tm_prd_trk_cum, sp_post_tm_prd_ot_cum.

Tables Updated

- dly_rvc_fixed_prd_ttl
- dly_sys_fixed_prd_ttl
- dly_rvc_fixed_prd_ot_ttl
- dly_sys_fixed_prd_ot_ttl
- dly_rvc_fixed_prd_trk_ttl

Related Reports

- Revenue Center Time Period Detail (TP_R101.RPT)

Post RVC Totals
sp_R_rvc_totals

This stored procedure can be used to post totals throughout the day so related reports will have a quick run time.

This stored procedure calls the sp_post stored procedure with parameters that specify for it to call sp_post_cum.

Tables Updated

- dly_rvc_ttl
- dly_rvc_ot_ttl
- dly_sys_ttl
- dly_sys_ot_ttl

Related Reports

- Consolidated Revenue Center Tip Totals (Tip_R001.RPT)

Post SYS Sales and Tracking
sp_R_sys_sales_tracking

This stored procedure can be used to post totals throughout the day so related reports will have a quick run time.

This stored procedure calls the sp_post stored procedure with parameters that specify for it to call sp_post_ot_cum and sp_post_rvc_trk_cum.

Tables Updated

- dly_rvc_ot_ttl
- dly_sys_ot_ttl
- dly_rvc_ttl
- dly_sys_ttl
- dly_sys_trk_ttl

Related Reports

- Daily System Sales Detail (SYS_101.RPT)
- Consolidated System Sales Detail (SYS_102.RPT)

Post SYS Time Period Sales and Tracking
sp_R_sys_time_prd_sales_trk

This stored procedure can be used to post totals throughout the day so related reports will have a quick run time.

This stored procedure calls the sp_post stored procedure with parameters that specify for it to call sp_post_tm_prd_cum, sp_post_rvc_rvc_tm_prd_trk_cum, and sp_post_sys_tm_prd_trk_cum.

Tables Updated

- dly_rvc_fixed_prd_ttl
- dly_sys_fixed_prd_ttl
- dly_rvc_fixed_prd_ot_ttl
- dly_sys_fixed_prd_ot_ttl
- dly_sys_fixed_prd_trk_ttl

Related Reports

- Revenue Center Time Period Detail (TP_R101.RPT)

Post SYS Totals
sp_R_sys_totals

This stored procedure can be used to post totals throughout the day so related reports will have a quick run time.

This stored procedure calls the sp_post stored procedure with parameters that specify for it to call sp_post_cum.

Tables Updated

- dly_sys_ttl
- dly_rvc_ttl

Related Reports

- Consolidated System Tip Totals (Tip_S001.RPT)

Post Tax Totals
sp_R_tax

This stored procedure can be used to post totals throughout the day so related reports will have a quick run time.

This stored procedure calls the sp_post stored procedure with parameters that specify for it to call sp_post_tax_cum.

Tables Updated

- dly_rvc_tax
- dly_rvc_ttl
- dly_sys_tax
- dly_sys_ttl

Related Reports

- Daily System Tax Totals (Tax_S001.RPT)
- Consolidated System Tax Totals (Tax_S002.RPT)
- Daily Revenue Center Tax Totals (Tax_R001.RPT)
- Consolidated Revenue Center Tax Totals (Tax_S001.RPT)

Post Time Card Totals
sp_R_time_card_ttls

This stored procedure can be used to post totals throughout the day so related reports will have a quick run time.

This stored procedure calls the sp_post_time_card_dtl stored procedure.

Tables Read

- time_card_dtl

- job_def

Tables Updated

- dly_rvc_ttl

Related Reports

- Employee Time Card and Job Detail (Time_002.RPT)
- Daily System Sales Detail (SYS_101.RPT)
- Consolidated System Sales Detail (SYS_102.RPT)
- Daily Revenue Center Sales Detail (RVC_101.RPT)
- Consolidated Revenue Center Sales Detail (RVC_102.RPT)

Post Time Period Sales
sp_R_time_period

This stored procedure can be used to post totals throughout the day so related reports will have a quick run time.

This stored procedure calls the sp_post stored procedure with parameters that specify for it to call sp_post_tm_prd_cum and sp_post_tm_prd_ot_cum.

Tables Updated

- dly_rvc_fixed_prd_ttl
- dly_sys_fixed_prd_ttl

Related Reports

- System Time Period Totals (TP_S001.RPT)
- Consolidated System Time Period Detail (TP_S002.RPT)
- System Time Period Summary (TP_S103.RPT)
- Consolidated System Time Period Summary (TP_S104.RPT)
- Revenue Center Time Period Summary (TP_R103.RPT)
- Revenue Center Time Period Totals (TP_R001.RPT)

Purge Historical Totals
sp_PurgeHistory

This stored procedure is called as part of the end of night autosequence. It deletes old detail and totals records from the database, which keep the database size manageable and improves system performance.

Tables Read

- rest_status
- time_clock_def
- rest_def
- hist_ttl_def

Tables Updated

There are no tables updated with this stored procedure.

Related Reports

There are no related reports.

Reset Cash Drawer Link
sp_ResetCashDrawerLink

This stored procedure clears the link between an employee and a cash drawer. The link is most commonly found behind the bar when two or more bartenders are using one cash drawer. It is usually run as part of an end of shift autosequence.

Tables Read
This stored procedure does not read any tables.

Tables Updated

- ❑ emp_def

Related Reports
There are no related reports.

Reset Employee Cashier Link
sp_ResetCashierLink

This stored procedure clears all links for cashiers that are linked to employees. This procedure is usually run as part of an end of shift autosequence.

Tables Read
This stored procedure does not read any tables.

Tables Updated

- ❑ emp_status

Related Reports
There are no related reports.

Reset One Touch Sign In
sp_ResetOneTouchSignIn

This stored procedure clears all one touch sign in keys. One touch sign in allows an employee to sign in one time with their ID. From that point on a key will display on the touchscreen with their name. This stored procedure is run by the revenue center, usually at the end of the day, to clear all one touch sign in keys.

Tables Read
This report does not read any tables.

Tables Updated

- ❑ rvc_sign_in_status

Related Reports
There are no related reports.

Reset User Workstation Cashier Link by RVC
sp_ResetRVC_UWSCashierLink

This stored procedure is run as part of the end of shift autosequence. It resets the link for a cashier to a UWS to zero. This stored procedure prevents totals from accumulating for the first cashier.

Tables Read

- ❑ uws_def

Tables Updated

- uws_def

Related Reports

There are no related reports.

RVC Open Checks
sp_RVCOpenChecks

This stored procedure is usually run as part of a pre close autosequence. It checks for open checks within a revenue center. If there are open checks the stored procedure can be programmed to print an RVC open check report. If the stored procedure returns no errors the next step in the current autosequence or the next autosequence or will run.

Tables Read

- chk_dtl

Tables Updated

This stored procedure does not update any tables.

Related Reports

There are no related reports.

Set the business date
sp_SetBusinessDate

This stored procedure is used to set the business date. It is usually called as part of the end of night autosequence. After this procedure is called all subsequent transactions will be posted to the new date.

Tables Read

- rest_status
- dly_sys_ttl
- rest_def

Tables Updated

- rest_status
- dly_sys_ttl

Related Reports

There are no related reports.

System Open Checks
sp_SystemOpenChecks

This stored procedure is usually run as part of a pre close autosequence. It checks for open checks within the system. If there are open checks the stored procedure can be programmed to print a system open check report. If the stored procedure returns no errors the next step in the current autosequence or the next autosequence will run.

Tables Read

- chk_dtl

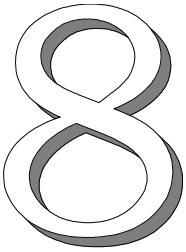
Tables Updated

This stored procedure does not update any tables.

Related Reports

There are no related reports.

Chapter



Domain Datatypes

This chapter describes the attributes of each datatype in the 3700 Database.

In this chapter

Datatype Overview

A datatype is a specification that designates the type of information accepted in a column and the manner in which data is stored. Each column in the 3700 Database is assigned a specific datatype.

The datatype definition for each column depends on the SQL DBMS being used. For instance, the 3700 domain Breakpoint uses the SMALLINT datatype in a SQL Anywhere Database.

SQL Anywhere Datatypes

The following list describes the SQL Anywhere datatypes that are used in mapping the 3700 user-defined datatypes to the SQL Anywhere database.

Datatype	Description
char(**size**)	Fixed-length character data with maximum length of **size**. The maximum size allowed is 32,767.
varchar(**size**)	Variable-length character data with maximum length of **size**. The maximum size allowed is 32,767.
integer	Datatype that holds values of -2,147,483,647 through 2,147,483,647. The storage size is 4 bytes.
datetime	A timestamp containing both the calendar date and time of day for a point in time. The useful range of values is 1600-02-28 23:59:59 to 7911-01-01 00:00:00. The storage size is 8 bytes.
smallint	Datatype that holds values of -32,767 through 32,767. The storage size is 2 bytes.
numeric (**precision**, **scale**)	Decimal number where **precision** is the total number of digits and **scale** is the number of digits after the decimal point.

3700 Domain Datatypes

This section describes the datatypes defined for the 3700 database. The following information is provided for each datatype:

- 3700 Domain Datatype

  Lists the user-defined datatype that is being described.
- SQL Anywhere Datatype

  Provides the SQL Anywhere Datatype that has been defined for the corresponding 3700 datatype.
- Null?

  Designates whether or not a NULL value is allowed for the column that uses this datatype.
- Default Value

  Designates the default value to be inserted into the column if a value has not been provided.
- Allowed Values

  Provides the values that can be entered into the column that uses the datatype.

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
ACCT_CLASS	numeric(1)	Not Null		1-Asset 2-Liability 3-Owners Equity 4-Income 5-Expense
ACCT_PERIOD	numeric(1)	Null		1-Week 2-Period 3-Quarter 4-Year
ACCTNG_CLOSE_TYPE	numeric(1)	Null		0-Soft 1-Hard
ACCTNG_CYCLE_FRQ_TYPE	numeric(1)	Null		0-Monthly 1-4/5/4 2-4/4/5 3-5/4/4 4-13-4 5-Event
ACTIVATION_ID	char(8)	Null		
ADDRESS_CITY	char(25)	Null		
ADDRESS_COUNTRY	char(25)	Null		
ADDRESS_LINE	char(30)	Null		
ADDRESS_NAME	char(20)	Null		
ADDRESS_NUM	char(10)	Null		

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
ADDRESS_POSTAL_CODE	char(15)	Null		
ADDRESS_STATE	char(2)	Null		
ADJUSTMENT_TYPE	numeric(1)	Null	0	0-Discount 1-Extra Charge
ALLOCATION_TYPE	numeric(1)	Null	1	1-Do Not Distribute 2-Distribute By Quantity 3-Distribute By Adjusted Total 4-Distribute By Unadjusted Total 5-Distribute By Price 6-Distribute Equally
AMT_PCT_TYPE	char(1)	Null	A	A-Amount P-Percent
AP_ACCOUNT_ID	char(10)	Null		
APP_CODE	int	Not Null	0	
AREA_CITY_COUNTRY_CODE	char(9)	Null		
ASEQ_ERROR_ACTION	char(1)	Not Null	S	B-ERROR_ACTION_BRANCH I-ERROR_ACTION_IGNORE S-ERROR_ACTION_STOP
ASEQ_STEP_TYPE	char(1)	Null		C-STEP_TYPE_CALL_ASEQ R-STEP_TYPE_REPORT S-STEP_TYPE_STORED_PROCEDURE X-STEP_TYPE_EXTERNAL_PROGRAM

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
AUDIT_TYPE	numeric(1)	Not Null		0-Internal 1-External 2-Corporate 3-Management Turnover 4-Special Audit A 5-Special Audit B 6-Special Audit C
BAUD_RATE	numeric(1)	Null		0-BAUD_RATE_300 1-BAUD_RATE_1200 2-BAUD_RATE_2400 3-BAUD_RATE_4800 4-BAUD_RATE_9600 5-BAUD_RATE_19200 6-BAUD_RATE_38400 7-BAUD_RATE_57600
BOOLEAN	numeric(1)	Null	0	0-FALSE 1-TRUE
BREAKPOINT	numeric(2)	Null		1-99
BUS_DATE	datetime	Null		
CA_RESPONSE	char(1)	Not Null		A-CA_RESPONSE_APPROVED C-CA_RESPONSE_CONDITIONALLY_APPROVED D-CA_RESPONSE_DECLINED E-RESPONSE_ERROR M-CA_RESPONSE_MANUAL_AUTH_REQUESTED
CAL_DATE	datetime	Null		

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
CASH_DRAWER	numeric(1)	Not Null	0	0-NO_CASH_DRAWER 1-CASH_DRAWER_1 2-CASH_DRAWER_2
CC_PREAMBLE	char(8)	Null		
CFG_FORM	smallint	Null		
CHECK_COMPLETE	char(1)	Not Null	T	I-CHECK_COMPLETE_INCOMPLETE C-CHECK_COMPLETE_COMPLETE F-CHECK_COMPLETE_FIXED D-CHECK_COMPLETE_DAMAGED X-CHECK_COMPLETE_UNKNOWN
CHECK_NUM	numeric(4)	Null		-9999-+9999
CLOCK_DTL_STATUS	char(1)	Null		A-TCS_EARLY_FROM_BREAK B-TCS_ON_BREAK C-TCS_LATE_FROM_BREAK D-TCS_SCHEDULE_DISABLED E-TCS_EARLY L-TCS_LATE M-TCS_MGR_CLK_OUT N-TCS_NOT_SCHEDULED P-TCS_ON_PAID_BREAK T-TCS_ON_TIME
CLOSED_DAY_OPT	numeric(1)	Null	2	0-Order Before Holiday 1-Order After Holiday 2-Skip Altogether
CNT_SHT_SORT_ORDR	numeric(1)	Not Null		1-by location 2-by task

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
CNV_FCTR	numeric(8,4)	Null		1-9999.9999
COMMENT_LINE	varchar(256)	Null		
CONNECTION_TYPE	char(1)	Null		C-CONNECTION_TYPE_COM_PORT M-CONNECTION_TYPE_MICROS_PCWS O-CONNECTION_TYPE_OPOS U-CONNECTION_TYPE_UNIVERSAL_SERIAL_BUS
CONTRAST	numeric(2)	Null		-99-+99
CONVRATE	numeric(12,6)	Null		1-999999.999999
COUNT_STATE	numeric(1)	Not Null	0	0-Definition in Progress 1-Defined 2-Count in Progress 3-Entry in Progress 4-Reconciled 5-Canceled
COUNT_TYPE	numeric(1)	Null		1-On-Hand Count of Selected Locations for Order Placement 2-Key Item Count of Selected Locations 3-Complete Count of All Locations
COUNT4	numeric(4)	Null		-9999-+9999
COUNT6	numeric(6)	Null		-999999-+999999
COURSE_NUM	numeric(2)	Null		1-16
DECIMAL_CHAR	char(1)	Null		(.) (,)
DECIMAL_DIGITS	numeric(1)	Not Null	2	0-3

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
DENSITY_FCTR	numeric(15,4)	Null		1-99999999999.9999
DENSITY_FCTR_VOL_WGHT	numeric(7,3)	Null		1-9999.999
DETAIL_TYPE	char(1)	Null		D-DETAIL_TYPE_DSC M-DETAIL_TYPE_MI T-DETAIL_TYPE_TMED R-DETAIL_TYPE_REF C-DETAIL_TYPE_CA N-DETAIL_TYPE_NULL V-DETAIL_TYPE_VOUCHER
DEVICE_TYPE	char(1)	Not Null		D-DVC_TYPE_DISK_FILE R-DVC_TYPEROLL_PRNTR S-DVC_TYPE_SLIP_PRNTR W-DVC_TYPE_UWS F-DVC_TYPE_FISCAL_CASH_REGISTER P-DVC_TYPE_PARALLEL_PRNTR_BOARD V-DVC_TYPE_VIDEO_DISPLAY_UNIT C-DVC_TYPE_COMM_INTF_BOARD H-DVC_TYPE_RS232_HOST I:DVC_TYPE_PC_ISN L-DVC_TYPE_LDS
DOC_SORT_ORDER	numeric(1)	Null	1	1-Vendor Item ID 2-My Item ID 3-As Entered
DOC_TYPE	numeric(1)	Null	1	1-Invoice 2-Packing Slip
DSC_ITMZR_NUM	numeric(1)	Not Null	1	

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
DSVC_CATEGORY	numeric(1)	Null	0	
DSVC_TYPE	char(1)	Null		D-DSVC_TYPE_DSC S-DSVC_TYPE_SVC
DTL_RECORD_TYPE	char(1)	Null		A-DTL_RECORD_TYPE_APPEND D-DTL_RECORD_TYPE_DELETE I-DTL_RECORD_TYPE_INSERT R-DTL_RECORD_TYPE_REPLACE S-DTL_RECORD_TYPE_SKIP
EM_CHG_SET_STATUS	char(1)	Not Null	I	A-Active I-Inactive R-Received S-Sent
EM_CHG_SET_TYPE	char(1)	Null		G-Global Table N-Notification S-Synchronization T-Totals Table U-User-defined
EM_MSG_TYPE	char(1)	Not Null	D	D-Database Update via Change Sets S-Database Synchronization
EM_REPL_STATUS_FLAG	char(1)	Null		C-Receipt Confirmed F-Not Sent T-Sent
EM_REST_TYPE	char(1)	Not Null		C-EM_REST_TYPE_EM_CENTRAL R-EM_REST_TYPE_STANDALONE_RESTAURANT T-EM_REST_TYPE_EM_STORE

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
EM_SYNC_TYPE	char(1)	Null		C-Centralize Data D-Distribute Data
EM_TTL_SEQ_NUM	numeric(7)	Null	auto-increment	-9999999-+9999999
EMPL_ID	numeric(10)	Null		1-9999999999
EO_BUS_OP_ACCESS	numeric(1)	Null	0	0-None 1-Full Access 2-Full Access
EO_BUS_OP_ID	smallint	Null		
EO_CFG_OP_ID	smallint	Null		
EO_RPT_ID	smallint	Null		
EXPLORER_DATA_TYPE	numeric(1)	Null		0-Unknown 1-Group 2-Executable 3-Auto Sequence 4-Auto Sequence Step
EXT_PROG_RUN_STYLE	char(1)	Null		M-RUN_STYLE_MINIMIZED N-RUN_STYLE_NORMAL X-RUN_STYLE_MAXIMIZED
EXT_PROG_UI_TYPE	char(1)	Null		C-UI_TYPE_CONSOLE D-UI_TYPE_DOS N-UI_TYPE_NONE W-UI_TYPE_WINDOWS
EX_SEQ_NUM	numeric(20,0)	Null		1-99999999999999999999

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
FD_NAME	char(50)	Null		
FEDERAL_TAX_ID	char(10)	Null		
FILEPATH	varchar(255)	Null		
FISCAL_PERIOD_TYPE	char(1)	Not Null	1	1-FISCAL_PERIOD_MONTHLY 3-FISCAL_PERIOD_13_PERIODS 4-FISCAL_PERIOD_4_4_5 5-FISCAL_PERIOD_4_5_4 6-FISCAL_PERIOD_5_4_4
FORECAST METHOD	numeric(1)	Null		0-None 1-Recipe Explosion 2-Historical Usage
FOREIGN_CNCY	numeric(15,3)	Null		1-99999999999.9999
FORM_SIZE	numeric(1)	Not Null	0	0-640 X 480 1-800 X 600 2-1024 X 768
FREQUENCY_METHOD	numeric(1)	Null		0-Day 1-Specific

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
FREQUENCY_SPECIFIC_DAY	numeric(2)	Null		0-Day 1-Weekday 2-Weekend Day 3-Sunday 4-Monday 5-Tuesday 6-Wednesday 7-Thursday 8-Friday 9-Saturday
GL_ACCT_ID	numeric(9)	Null		1-999999999
HHMM	numeric(4)	Null		0-2359
HOURS	numeric(2)	Null		0-99
HOURS8_2	numeric(8,2)	Null		1-999999.99
ICON_ID	smallint	Null		
ID_FIELD	char(16)			
IDN_ID	smallint	Null		1-15
INFO_PRINT_FMT	char(1)	Null		A-CHK_INFO_NO_PRINT B-CHK_INFO_BEFORE_HEADER C-CHK_INFO_AFTER_HEADER D-CHK_INFO_AFTER_TRAILER
INTERFACE_TYPE	char(1)	Null		C-INTERFACE_TYPE_CIB P-INTERFACE_TYPE_PCISN R-INTERFACE_TYPE_RS232 T-INTERFACE_TYPE_TCP

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
INV_SELN_TYPE	numeric(1)	Null	1	1-Groups 2-Lists 3-Items 4-Location 5-Location Group
INV_UNIT_QTY	numeric(18,6)	Null		1-999999999999.999999
INVCE_PSLIP_STATUS	numeric(1)	Null	1	1-Deleted 2-Partial Receipt 3-Received 4-Reconciled to Receipts 5-Reconciled to Invoice 6-Approved 7-Sent to A/P
INVEN_CLOSE_MTHD	numeric(1)	Null	1	1-Physical Count 2-Default to Zero 3-Default to Minimum Stock 4-Default to Maximum Stock 5-Default to Par Level
INVEN_MTHD	numeric(1)	Null		1-FIFO 2-Weighted Average Cost 3-Last Cost
INVEN_TRANS_TYPE	numeric(1)	Not Null		
INVOICE_ID	char(15)	Null		
KDS_CLIENT_TYPE	smallint	Null		1-KDS_CLIENT_TYPE_WIN32 2-KDS_CLIENT_TYPE_EMBEDDED

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
KDS_KEY_NUM	numeric(7)	Null		-9999999-+9999999
KDS_ORDER_TYPE	char(1)	Null		M-KDS_ORDER_TYPE_MENU_ITEM C-KDS_ORDER_TYPE_CONDIMENT R-KDS_ORDER_TYPE_REFERENCE
KDS_STATION_STATUS	smallint	Not Null		1-KDS_STATION_STATUS_WAITING 2-KDS_STATION_STATUS_STARTED 3-KDS_STATION_STATUS_DONE 4-KDS_STATION_STATUS_REOPENED
KDS_STATION_TYPE	smallint	Null		1-KDS_STATION_TYPE_PREP 2-KDS_STATION_TYPE_EXPO 3-KDS_STATION_TYPE_PICKUP
KEY_FUNC_NUM	numeric(7)	Null		-9999999-+9999999
KEY_NUM	numeric(7)	Null		1-99999999
KEY_TYPE	numeric(2)	Null		1-18
KYBD_TYPE	char(1)	Null		F-KYBD_TYPE_FULL H- KYBD_TYPE_HALF
LABOR_CATEGORY	numeric(1)	Not Null	1	1-4
LABOR_TOTAL	numeric(23,5)	Null		1-999999999999999999.99999
LAN_ADDRESS	char(15)	Null		
LAN_NODE_TYPE	char(1)	Null		P-LAN_NODE_PCWS D-LAN_NODE_DTWS O-LAN_NODE_PCWS_64 A-LAN_NODE_PCWS_64_REV_D

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
LANGID	smallint	Null		>0
LARGE_CMNT_LN	varchar(2048)	Null		
LOCAL_PHONE	char(10)	Null		
LOGICAL	char(1)	Not Null	F	F-FALSE T-TRUE
MI_CATEGORY	numeric(2)	Not Null	0	0-9
MI_NLU_GROUP	numeric(2)	Null		1-64
MINUTES	numeric(2)	Null		0-99
MLVL	numeric(1)	Null		1-4
MONEY12	numeric(12,2)	Null		1-9999999999.99
MONEY12_3	numeric(13,3)	Null		1-9999999999.999
MONEY18	numeric(18,2)	Null		1-9999999999999999.99
MONEY18_4	numeric(20,4)	Null		1-99999999999999.9999
MONTHS_OF_YEAR	numeric(2)	Null		1-12
NAME12	char(12)	Null		
NAME16	char(16)	Null		
NAME20	char(20)	Null		
NAME24	char(24)	Null		
NAME25	char(25)	Null		

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
NAME32	char(32)	Null		
NAME4	char(4)	Null		
NAME8	char(8)	Null		
NLU12	numeric(12)	Null		1-999999999999
NLU4	numeric(4)	Null		1-9999
NODE_TYPE	numeric(1)	Null		0-Branch
NOTIFY_ACTION	char(1)	Not Null		C-CREATE D-DELETE U-UPDATE
NUM_CHK_INFO_LINES	numeric(2)	Not Null	0	0-16
NUM_COLOR_BITS	numeric(2)	Null		1-COLOR_BLACK_AND_WHITE 4-COLOR_16_COLORS 8-COLOR_256_COLORS 16-COLOR_64K_COLORS
NUM_OF_DRAWERS	numeric(1)	Not Null	0	0-2
NUM_SUB_CNT_UNIT	numeric(1)	Not Null	1	1-One 2-Two 3-Three 4-Four 5-Five
OBJ_NAME	char(32)	Null		

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
OBJ_NUM	numeric(7)	Null		1-9999999

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
OBJECT_TYPE	numeric(2)	Null		0-Inventory Location Cost Groups 1-Ingredients 2-Recipes 3-Inventory Items 4-Inventory Locations 5-Store Groups 6-Stores 7-Employees 8-Vendors 9-Invoices 10-Par Levels 11-Units 12-Inventory Cost Groups 13-Physical Count Number 14-Future Unused 14 15-Future Unused 15 16-Future Unused 16 17-Future Unused 17 18-Future Unused 18 19-Future Unused 19 20-Future Unused 20 21-Future Unused 21 22-Future Unused 22 23-Future Unused 23 24-Future Unused 24 25-Future Unused 25 26-Future Unused 26 27-Future Unused 27 28-Future Unused 28 29-Future Unused 29 30-Future Unused 30

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
OMITTED_BY_TYPE	char(1)	Null		U-OMITTED_BY_USER D-OMITTED_BY_DRIVER
OPERATION_TYPE	char(1)	Null		*'-Multiply /'-Divide
OPOS_HIGHLIGHT	char(1)	Null	A	A-OPOS_HIGHLIGHT_AUTO B-OPOS_HIGHLIGHT_BOLD C-OPOS_HIGHLIGHT_ALTERNATE_COLOR I-OPOS_HIGHLIGHT_ITALICS R-OPOS_HIGHLIGHT_REVERSE_VIDEO U-OPOS_HIGHLIGHT_UNDERLINE
ORDER_DETAIL_STATUS	numeric(1)	Null		1-Added 2-Modified 3-Deleted 4-Created
ORDER_METHOD	numeric(1)	Null	1	1-Call In Over Phone 2-FAX 3-EDI 4-Other
ORDER_QTY_CALC_MTHD	numeric(1)	Null	3	1-Forecast 2-Par Level 3-Manual

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
ORDER_STATUS	numeric(2)	Null		1-Started 2-Placed 3-Completed 4-Cancelled 5-Modified 6-Partial Receipt 7-Received 8-Reconciled 9-Approved 10-Sent to A/P
PACKING_SLIP_ID	char(15)	Null		
PARITY_TYPE	char(1)	Not Null	N	N-PARITY_TYPE_NONE E-PARITY_TYPE_EVEN O-PARITY-TYPE_ODD
PARM_DATA_TYPE	char(1)	Null		N-DATA_TYPE_NUMERIC S-DATA_TYPE_STRING D-DATA_TYPE_DATE_TIME Q-DATA_TYPE_SEQ_NUM
PARM_VALUE_SOURCE	char(1)	Null		N-PARM_VALUE_NUMERIC S-PARM_VALUE_STRING R-PARM_VALUE_RVC_SEQ O-PARM_VALUE_OBJ_SEQ B-PARM_VALUE_BEGIN_DATE_TIME E-PARM_VALUE_END_DATE_TIME
PATHNAME	varchar(255)	Null		

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
PAYMENT_TERMS	numeric(1)	Null	1	1-COD 3-Check 4-Net X Days
PAYROLL_ID	char(16)	Null		
PAYROLL_TYPE	char(1)	Null		W-PAYROLL_TYPE_WEEKLY B-PAYROLL-TYPE_BI-WEEKLY S-PAYROLL_TYPE_SEMI-MONTHLY
PCNT	numeric(8,4)	Null		0-9999.9999
PERIPHERAL_TYPE	char(1)	Not Null		C-PERIPHERAL_TYPE_COIN_DISPENSER D-PERIPHERAL_TYPE_CASH_DRAWER_1 E-PERIPHERAL_TYPE_CASH_DRAWER_2 N-PERIPHERAL_TYPE_SCANNER M-PERIPHERAL_TYPE_MAGNETIC_STRIPE_READER R-PERIPHERAL_TYPE_REAR_DISPLAY S-PERIPHERAL_TYPE_SCALE
PERPETUAL_INVEN_MTHD	numeric(1)	Not Null	0	0-None 1-Recipe Explosion 2-Historical Usage
PHONE_EXT	char(4)	Null		
PO_NUMBER	numeric(10)	Null		1-9999999999
PO_PREFIX	char(6)	Null		

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
PO_STATUS	numeric(1)	Not Null		1-Placed 2-Cancelled 3-Modified 4-Partial Received 5-Received 6-Reconciled 7-Closed 8-Merged
POS_TRANS_TYPE	numeric(1)	Null		0-Sale 1-Return 2-Void Sale 3-Void Return
POS_TYPE	char(1)	Null	3	2-POS_TYPE_2800 3-POS_TYPE_3700 O-POS_TYPE_OTHER
POS2700_NODE_NUMBER	smallint	Not Null	0	
POS2800_CIB_PORT_ FUNCTION_1	numeric(2)	Null		1-POS2800_CIB_PORT_FUNCTION_1_HOST_COMPUTER 2-POS2800_CIB_PORT_FUNCTION_1_FCR 3-POS2800_CIB_PORT_FUNCTION_1_LDS
POS2800_CIB_PORT_ FUNCTION_2	numeric(2)	Null		1-POS2800_CIB_PORT_FUNCTION_2_IDN_COMM 2-POS2800_CIB_PORT_FUNCTION_2_GLOBAL_ISN_COMM
POS2800_DRAWER_TIMEOUT	numeric(3)	Null		0-255
POS2800_KEY_FUNC_NUM	numeric(7)	Null		
POS2800_KEY_TYPE	numeric(2)	Null		

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
POS2800_MGR_PROC	numeric(2)	Null		
POS_NODE_NUMBER	smallint	Not Null	0	0-32
POS2800_OMIT_ZEROS	numeric(1)	Not Null		0-POS2800_OMIT_ZEROS_PROMPT 1-POS2800_OMIT_ZEROS_NO 2-POS2800_OMIT_ZEROS_YES 3-POS2800_OMIT_ZEROS_INHERIT
POS2800_OPTION_BOARD_TYPE	numeric(1)	Null	1	1-POS2800_OPTION_BOARD_TYPE_CIB 2-POS2800_OPTION_BOARD_TYPE_PIB 3-POS2800_OPTION_BOARD_TYPE_TIB
POS2800_PERIOD	numeric(1)	Null		0-POS2800_PERIOD_PROMPT 1-POS2800_PERIOD_CURRENT 2-POS2800_PERIOD_INHERIT
POS2800_PIB_KYBD_TERMINATOR	numeric(3)	Null	221	-999-+999
POS2800_PIB_PORT_FUNCTION_1	numeric(2)	Null		1-POS2800_PIB_PORT_FUNCTION_1_NCI_400_SCALE 2-POS2800_PIB_PORT_FUNCTION_1_METROLOGIC_BARCODE 3-POS2800_PIB_PORT_FUNCTION_1_GP_KYBD 4-POS2800_PIB_PORT_FUNCTION_1_SYMBOL_LASER_SCANNER

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
POS2800_PIB_PORT_ FUNCTION_2	numeric(2)	Null		1-POS2800_PIB_PORT_FUNCTION_2_BRANDT_COIN_ CHANGER 2-POS2800_PIB_PORT_FUNCTION_2_METROLOGIC_ BARCODE 3-POS2800_PIB_PORT_FUNCTION_2_GP_KEYBOARD 4-POS2800_PIB_PORT_FUNCTION_2_SYMBOL_LASER_ SCANNER
POS2800_PRINTER	numeric(1)	Null		0-POS2800_PRINTER_PROMPT 1-POS2800_PRINTER_REPORT_PRINTER_1 2-POS2800_PRINTER_REPORT_PRINTER_2 3-POS2800_PRINTER_NO_PRINT 4-POS2800_PRINTER_INHERIT
POS2800_RANGE_TYPE	numeric(1)	Null		0-POS2800_RANGE_TYPE_PROMPT 1-POS2800_RANGE_TYPE_ALL 2-POS2800_RANGE_TYPE_RANGE 3-POS2800_RANGE_TYPE_REPORT_GROUP 4-POS2800_RANGE_TYPE_INHERIT 5-POS2800_RANGE_TYPE_COMPOUND
POS2800_RESET	numeric(1)	Null		0-POS2800_RESET_PROMPT 1-POS2800_RESET_NO 2-POS2800_RESET_YES 3-POS2800_RESET_INHERIT
POS2800_SCOPE	numeric(1)	Null		0-POS2800_SCOPE_PROMPT 1-POS2800_SCOPE_SYSTEM_UNIT 2-POS2800_SCOPE_REVENUE_CENTER 3-POS2800_SCOPE_SYSTEM 4-POS2800_SCOPE_ALL_SYSTEMS 5-POS2800_SCOPE_INHERIT

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
POS2800_SCOPE_VALUE	numeric(2)	Null		0-16
POS2800_SU_ASEQ_RPT_ TYPE	numeric(2)	Null		1-POS2800_SU_ASEQ_RPT_TYPE_SYSTEM_BALA NCE 2-POS2800_SU_ASEQ_RPT_TYPE_TAX_SUMMAR Y 3-POS2800_SU_ASEQ_RPT_TYPE_TIP_SUMMARY 4-POS2800_SU_ASEQ_RPT_TYPE_RVC_BALANCE 5-POS2800_SU_ASEQ_RPT_TYPE_SU_BALANCE 6-POS2800_SU_ASEQ_RPT_TYPE_EMPLOYEE_BALANCE 7-POS2800_SU_ASEQ_RPT_TYPE_EMPLOYEE_ OPEN_CHK 8-POS2800_SU_ASEQ_RPT_TYPE_EMPLOYEE_ CLSD_CHK 9-POS2800_SU_ASEQ_RPT_TYPE_EMPLOYEE_TIP 10-POS2800_SU_ASEQ_RPT_TYPE_CASHIER_BALANCE 11-POS2800_SU_ASEQ_RPT_TYPE_MAJ_GRP_DTL 12-POS2800_SU_ASEQ_RPT_TYPE_MAJ_GRP_SUMMARY 13-POS2800_SU_ASEQ_RPT_TYPE_FAM_GRP_DTL 14-POS2800_SU_ASEQ_RPT_TYPE_FAM_GRP_ SUMMARY 15-POS2800_SU_ASEQ_RPT_TYPE_MI_DTL 16-POS2800_SU_ASEQ_RPT_TYPE_MI_SUMMARY 17-POS2800_SU_ASEQ_RPT_TYPE_TIME_PERIOD 18-POS2800_SU_ASEQ_RPT_TYPE_CLOCK_IN_STATUS 19-POS2800_SU_ASEQ_RPT_TYPE_LDS 20-POS2800_SU_ASEQ_RPT_TIME_PERIOD_SUMMARY 110-POS2800_SU_ASEQ_RPT_EMPLOYEE_LABOR _DTL 111-POS2800_SU_ASEQ_RPT_EMPLOYEE_LABOR_ SUMMARY 121-POS2800_SU_ASEQ_RPT_LABOR_AVAILABILITY 122-POS2800_SU_ASEQ_RPT_JOB_CODE_LABOR

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
POS2800_SU_STATUS	char(1)	Not Null	A	A-POS2800_SU_STATUS_ACTIVE I-POS2800_SU_STATUS_INACTIVE T-POS2800_SU_STATUS_PC_TTLS_ONLY
POS2800_THOUSANDS_ PLACE	char(1)	Null	(,)	(,) (.)
POS2800_TRK_TRANSFER	numeric(2)	Null		0-POS2800_TRK_TRANSFER_OUT 1-POS2800_TRK_TRANSFER_OUT
POS2800_TRK_TTL_TYPE	numeric(2)	Null		1-POS2800_TRK_TTL_TYPE_MAJ_GRP 2-POS2800_TRK_TTL_TYPE_FAM_GRP 3-POS2800_TRK_TTL_TYPE_MI 4-POS2800_TRK_TTL_TYPE_DSVC 5-POS2800_TRK_TTL_TYPE_TMED 6-POS2800_TRK_TTL_TYPE_SUBTTL_PRN_ONLY 7-POS2800_TRK_TTL_TYPE_TTL_PRN_ONLY 8-POS2800_TRK_TTL_TYPE_PRN_NAME_ONLY 9-POS2800_TRK_TTL_TYPE_CURRENCY_CONVE RSION 10-POS2800_TRK_TTL_TYPE_TAX 11-POS2800_TRK_TTL_TYPE_SUBTTL_DISCOUN T 12-POS2800_TRK_TTL_TYPE_OT_SALES 13-POS2800_TRK_TTL_TYPE_VOID/RETURN_REASON_ CODE 14-POS2800_TRK_TTL_TYPE_INSUFFICIENT_BEV 15-POS2800_TRK_TTL_TYPE_BEV_ADDED_GST_CHK 16-POS2800_TRK_TTL_TYPE_MEDIA_DECLARATION 17-POS2800_TRK_TTL_TYPE_COV_CNT 18-POS2800_TRK_TTL_TYPE_VOIDS 19-POS2800_TRK_TTL_TYPE_CHK_XFER

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
POS2800_TRK_VOIDS	numeric(2)	Null		1-POS2800_TRK_VOIDS_EMPLOYEE 2-POS2800_TRK_VOIDS_MANAGER 3-POS2800_TRK_VOIDS_CURRENT_ROUND 4-POS2800_TRK_VOIDS_PREVIOU_ROUND
POS2800_VDU_CFG	char(1)	Null	A	A-POS2800_VDU_CFG_SINGLE_QUEUE_MAX_4 B-POS2800_VDU_CFG_SINGLE_QUEUE_MAX_8 C-POS2800_VDU_CFG_DUAL_QUEUE_LOWER D-POS2800_VDU_CFG_DUAL_QUEUE_LOWER
POS3700_DEVICE_TYPE	char(1)	Not Null		D-POS3700_DVC_TYPE_DISK_FILE R-POS3700_DVC_TYPE_ROLL_PRNTR S-POS3700_DVC_TYPE_SLIP_PRNTR W-POS3700_DVC_TYPE_UWS F-POS3700_DVC_TYPE_FISCAL_CASH_REGISTER
POS3700_INTERFACE_TYPE	char(1)	Null		R-POS3700_INTERFACE_TYPE_RS232 T-POS3700_INTERFACE_TYPE_TCP
PRINT_GRP	numeric(1)	Not Null	1	1-8
PRINTER_INTERFACE_TYPE	char(1)	Null		I-PRINTER_INTERFACE_TYPE_IDN P-PRINTER_INTERFACE_TYPE_PARALLEL R-PRINTER_INTERFACE_TYPE_RS232 O-PRINTER_INTERFACE_TYPE_OPOS

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
PRINTER_TYPE	char(1)	Null		A-PRINTER_TYPE_AUTOCUT_ROLL C-PRINTER_TYPE_EPSON_TM_300 D-PRINTER_TYPE_EPSON_TM_200 E-PRINTER_TYPE_EXPRESS_THERMAL J-PRINTER_TYPE_OPOS_JOURNAL K-PRINTER_TYPE_OPOS_RECEIPT L-PRINTER_TYPE_OPOS_SLIP R-PRINTER_TYPE_STANDALONE_ROLL S-PRINTER_TYPE_EPSON_TM_290 T-PRINTER_TYPE_EPSON_TM_T80 U-PRINTER_TYPE_EPSON_TM_T85 V-PRINTER_TYPE_EPSON_TM_T88
PRINTER_TYPE_IDN	char(1)	Null		A-PRINTER_TYPE_AUTOCUT_ROLL C-PRINTER_TYPE_EPSON_TM_300 D-PRINTER_TYPE_EPSON_TM_200 E-PRINTER_TYPE_EXPRESS_THERMAL J-PRINTER_TYPE_OPOS_JOURNAL K-PRINTER_TYPE_OPOS_RECEIPT L-PRINTER_TYPE_OPOS_SLIP R-PRINTER_TYPE_STANDALONE_ROLL S-PRINTER_TYPE_EPSON_TM_290 T-PRINTER_TYPE_EPSON_TM_T80 U-PRINTER_TYPE_EPSON_TM_T85 V-PRINTER_TYPE_EPSON_TM_T88

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
PRINTER_TYPE_OPOS	char(1)	Null		A-PRINTER_TYPE_AUTOCUT_ROLL C-PRINTER_TYPE_EPSON_TM_300 D-PRINTER_TYPE_EPSON_TM_200 E-PRINTER_TYPE_EXPRESS_THERMAL J-PRINTER_TYPE_OPOS_JOURNAL K-PRINTER_TYPE_OPOS_RECEIPT L-PRINTER_TYPE_OPOS_SLIP R-PRINTER_TYPE_STANDALONE_ROLL S-PRINTER_TYPE_EPSON_TM_290 T-PRINTER_TYPE_EPSON_TM_T80 U-PRINTER_TYPE_EPSON_TM_T85 V-PRINTER_TYPE_EPSON_TM_T88
PRINTER_TYPE_RS232	char(1)	Null		A-PRINTER_TYPE_AUTOCUT_ROLL C-PRINTER_TYPE_EPSON_TM_300 D-PRINTER_TYPE_EPSON_TM_200 E-PRINTER_TYPE_EXPRESS_THERMAL J-PRINTER_TYPE_OPOS_JOURNAL K-PRINTER_TYPE_OPOS_RECEIPT L-PRINTER_TYPE_OPOS_SLIP R-PRINTER_TYPE_STANDALONE_ROLL S-PRINTER_TYPE_EPSON_TM_290 T-PRINTER_TYPE_EPSON_TM_T80 U-PRINTER_TYPE_EPSON_TM_T85 V-PRINTER_TYPE_EPSON_TM_T88
PRIORITY	numeric(4)	Null		0-9999
PRIV_LEVEL	numeric(1)	Not Null	0	0-3

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
PROD_TYPE	numeric(1)	Null	0	0-None 1-Passive 2-Auto 3-Active
PRODUCT_TYPE	char(1)	Not Null		I-PRODUCT_TYPE_INVENTORY_ITEM R-PRODUCT_TYPE_RECIPE
RANGE_DATE_TYPE	char(1)	Null		B-RANGE_BUSINESS_DATE C-RANGE_CALENDAR_DATE F-RANGE_START_OF_FISCAL_PERIOD M-RANGE_START_OF_MONTH P-RANGE_START_OF_PAYROLL_PERIOD S-RANGE_SPECIFIED_DATE W-RANGE_START_OF_WEEK Y-RANGE_START_OF_YEAR f-RANGE_END_OF_FISCAL_PERIOD m-RANGE_END_OF_MONTH p-RANGE_END_OF_PAYROLL_PERIOD w-RANGE_END_OF_WEEK y-RANGE_END_OF_YEAR
RANGE_SOURCE	char(1)	Null		P-RANGE_SOURCE_PROMPT I-RANGE_SOURCE_INHERIT S-RANGE_SOURCE_SPECIFIED A-RANGE_SOURCE_ALL
RANGE_SOURCE_SHIFT	char(1)	Null		P-RANGE_SOURCE_PROMPT I-RANGE_SOURCE_INHERIT C-RANGE_SOURCE_CURRENT_SHIFT V-RANGE_SOURCE_PREVIOUS_SHIFT A-RANGE_SOURCE_ALL_SHIFTS

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
RECEIVE_ITM_STATUS	numeric(1)	Null	1	1-Received 2-Posted to Inventory
RECIPE_COST_CALC	numeric(1)	Null		2-Weighted Average Cost 3-Last Cost 4-Standard Cost
RECIPE_COST_METHOD	numeric(1)	Null		0-Calculate From Ingredients 1-Use Standard Cost
RECIPE_TYPE	char(1)	Null		B-Batch Recipe S-Serving Recipe
REQUIREMENTS_METHOD	numeric(1)	Null		0-Forecast 1-Par Level 2-Manual
RES_PRODUCT_TYPE	char(1)	Null	3	2-RES_PRODUCT_TYPE_POS_2800 3-RES_PRODUCT_TYPE_POS_3700 E-RES_PRODUCT_TYPE_EO G-RES_PRODUCT_TYPE_GSS
REVISED_ORDER_SEND_FORMAT	numeric(1)	Null		0-Resend Complete Order 1-Resend the items that have changed
RGB_COLOR	numeric(8)	Null		0-1677215
ROUND_QTY_TYPE	numeric(1)	Not Null	3	0-Do Not Round 1-Round Up 2-Round Down 3-Round Nearest
RPT_CATEGORY	numeric(2)	Not Null	0	0-13

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
RPT_OBJ_TYPE	char(1)	Null		C-OBJ_TYPE_CASHIER E-OBJ_TYPE_EMPLOYEE G-OBJ_TYPE_MENU_ITEM_GROUP M-OBJ_TYPE_MENU_ITEM S-OBJ_TYPE_SERVING_PERIOD T-OBJ_TYPE_TIME_PERIOD
SCANNER_DATA_SIZE	numeric(2)	Null		1-20
SCANNER_FORMAT_TYPE	char(1)	Null		A-SCANNER_FORMAT_TYPE_UPC_A B-SCANNER_FORMAT_TYPE_UPC_E C-SCANNER_FORMAT_TYPE_CUSTOM D-SCANNER_FORMAT_TYPE_EAN_8 E-SCANNER_FORMAT_TYPE_EAN_13
SCHED_FREQUENCY	numeric(1)	Not Null		0-Daily 1-Weekly 2-Monthly 3-Period 4-Yearly
SCHED_INVEN_CNT_MTHD	numeric(1)		0	0-None 1-Required Task 2-Use Perpetual Stock On Hand
SCHED_MEMBER_TYPE	numeric(1)	Not Null		0-Item 1-List 2-Group 3-Task

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
SCHED_ORDINAL	numeric(1)	Null		0-First 1-Second 2-Third 3-Fourth 4-Last
SCHED_TYPE	numeric(1)	Null	0	0-Inventory 1-Ordering
SCHEDULE_DATE_TYPE	char(1)	Null		B-SCHEDULE_BIWEEKLY D-SCHEDULE_DAILY F-SCHEDULE_FIRST_DAY_OF_FISCAL_PERIOD L-SCHEDULE_LAST_DAY_OF_FISCAL_PERIOD M-SCHEDULE_DATES_IN_MONTH P-SCHEDULE_PER_PAYROLL_PERIOD S-SCHEDULE_SPECIFIED_DATE W-SCHEDULE_DAYS_IN_WEEK Y-SCHEDULE_MONTHS_IN_YEAR
SECONDS	numeric(7)	Null		0-9999999
SEQ_NUM	numeric(7)	Null		1-9999999
STORE_ID	numeric(7)	Null		1-9999999
SUMM_OPTION_TYPE	numeric(1)	Not Null	0	0-Summarize To Account 1-Do Not Summarize
SVC_ITMZR_NUM	numeric(1)	Not Null	1	1-4
TAX_TYPE	char(1)	Null	P	B-TAX_TYPE_BREAKPOINT P-TAX_TYPE_PERCENT I-TAX_TYPE_INCLUSIVE

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
TEXT20	char(20)	Null		
TEXT24	char(24)	Null		
TEXTLINE16	varchar(16)	Null		
TEXTLINE32	varchar(32)	Null		
TEXTLINE40	varchar(40)	Null		
THAW_HRS	numeric(5,2)	Null		0-999.99
TIME_CLOCK_STATUS	char(1)	Null	O	B-ETS_ON_BREAK O-ETS_CLK_OUT B-ETS_ON_BREAK P-ETS_ON_PAID_BREAK I-ETS_CLK_IN S-ETS_CLK_IN_W_SCHED
TMED_CATEGORY	numeric(1)	Null	1	1-4
TMED_TYPE	char(1)	Not Null		T-TMED_TYPE_PAYMENT S-TMED_TYPE_SVC_TTL P-TMED_TYPE_PICKUP L-TMED_TYPE_LOAN
TRAINING_STATUS	numeric(1)	Null		0-2
TRANS_TYPE	char(1)	Null		S-TRANS_TYPE_SALE C-TRANS_TYPE_CANCEL N-TRANS_TYPE_NO_SALE X-TRANS_TYPE_XFER T-TRANS_TYPE_TIP_DECLARED P-TRANS_TYPE_PICKUP_LOAN B-TRANS_TYPE_BALANCE_CORRECTION

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
TRK_TTL_NUM	numeric(7)	Null		0-+9999999
TRK_TTL_NUM_CHK_XFER	numeric(7)	Null		0-CHK_XFER_IN 1-CHK_XFER_OUT
TRK_TTL_NUM_EMP_MEAL	numeric(7)	Null		1-EMP_MEAL_DISCOUNT 2-EMP_MEAL_TENDER
TRK_TTL_TYPE	numeric(7)	Null		-99-+99
TS_ARRANGE_TYPE	char(1)	Null		C-TS_ARRANGE_TYPE_CENTER H-TS_ARRANGE_TYPE_HORIZONTAL V-TS_ARRANGE_TYPE_VERTICAL
TS_COLOR	numeric(2)	Not Null	1	-99-+99
TS_COLUMN	smallint	Null		1-30
TS_FONT	numeric(2)	Not Null	1	-99-+99
TS_HEIGHT	smallint	Null		2-12
TS_ICON_PLACEMENT	char(1)	Null		L-TS_ICON_PLACEMENT_LEFT R-TS_ICON_PLACEMENT_RIGHT T-TS_ICON_PLACEMENT_TOP B-TS_ICON_PLACEMENT_BOTTOM C-TS_ICON_PLACEMENT_CENTER
TS_ROW	smallint	Null		1-12
TS_WIDTH	smallint	Null		2-30
TYPE_AND_SEQ	char(8)	Null		

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
UNIT_TYPE	char(1)	Not Null	U	U'-Unit V'-Volume W'-Weight
USER_ALPHA_FIELD	char(32)	Null		
USER_DATA_TYPE	smallint	Null		0-USER_DATA_TYPE_DATE 1-USER_DATA_TYPE_STRING 2-USER_DATA_TYPE_NUMBER 3-USER_DATA_TYPE_EMPLOYEE_LIST 4-USER_DATA_TYPE_RVC_LIST 5-USER_DATA_TYPE_STORE_LIST 6-USER_DATA_TYPE_VENDOR_LIST 7-USER_DATA_TYPE_RECIPE_LIST 8-USER_DATA_TYPE_INVENTORY_ITEM_LIST 9-USER_DATA_TYPE_INGREDIENT_LIST 10-USER_DATA_TYPE_BUSINESS_DATE
USER_INTEGER_FIELD	int	Null		
USER_NUMERIC_FIELD	numeric(16,4)	Null		0-999999999999.9999
VENDOR_ACCT_ID	char(10)	Null		
VENDOR_ITEM_NUMBER	char(10)	Null		
VENDOR_SEL_MTHD	numeric(1)	Null	4	1-Assign Contracted Vendor 2-Select Preferred Vendor with Best Price 3-Select Vendor with Best Price 4-Manually Select Vendor
VNDR_PERF_INDX	numeric(3)	Null		

3700 Domain Datatype	SQL Anywhere Datatype	Null?	Default Value	Allowed Values
WEEKDAY	numeric(1)	Null	1	1-WEEKDAY_SUNDAY 2-WEEKDAY_MONDAY 3-WEEKDAY_TUESDAY 4-WEEKDAY_WEDNESDAY 5-WEEKDAY_THURSDAY 6-WEEKDAY_FRIDAY 7-WEEKDAY_SATURDAY
WEIGHT10	numeric(10,3)	Null		0-9999999.999
WEIGHT6	numeric(6,3)	Null		0-999.999
XFER_TYPE	char(1)	Not Null		A-XFER_TYPE_ADD B-XFER_TYPE_BLOCK_XFER M-XFER_TYPE_MEMO_TENDER S-XFER_TYPE_SPLIT X-XFER_TYPE_XFER
YIELD	numeric(4,3)	Null		0-9999.999

Database Domain Report

The MICROS Database Domain Report lists the domain names currently in the database along with a valid set of values for each. The report is based on a pre-defined template and performs no database query when run. Therefore, changes made to the database will not affect the structure of the report template.

Like all Crystal Reports, the Database Domain Report is setup through the POS Configurator and accessed via the Autosequences and Reports module. The process consists of the following steps:

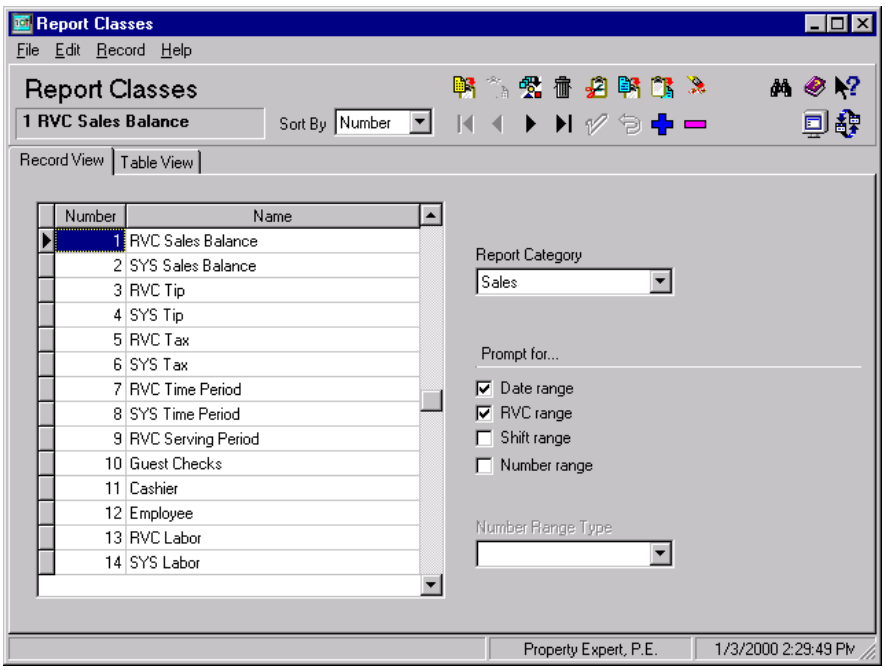
- ❑ Adding a Report Class
- ❑ Assigning a Report Template
- ❑ Running the Report

Adding a Report Class

Report classes allow you to assign similar attributes to a group of reports. The 3700 database is installed with a set of pre-defined report classes applicable to most POS operations. For some reports, however, the available options may not be applicable and a new report class may be required.

Follow these steps to add a new Report Class:

1. Open the POS Configurator.
2. Select **Reporting | Report Classes**.



3. Create a new report class by adding a record number and assigning it a value of 10000 or higher.

Note

MICROS assigns pre-defined report classes a number of 9999 or below. By setting the new report class at 10000 or greater, you ensure that upgrades or reinstallation of the POS system will not overwrite your custom entries.

4. Enter a **Name** for the report class.
5. Select *Other* from the **Report Category** drop-down list.
6. Make sure that all of the **Prompt for...** checkboxes are blank. Clear the boxes, if necessary.
7. Save the changes and close the Report Classes form.

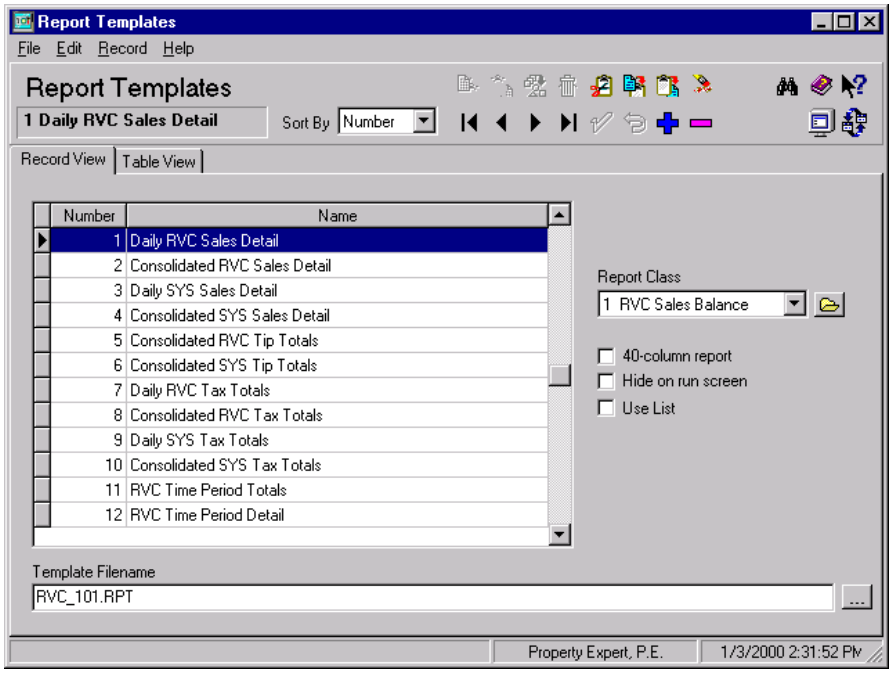
Assigning a Report Template

Report templates are pre-defined files that filter the data and determine how it will be displayed or printed when a report is generated. The 3700 System ships with a number of standard reports and templates, which can be accessed and configured to meet specific customer needs. To do this, the report template is linked to a report class, a new report name is added, and the template is assigned.

Follow these steps to add a new report and assign the appropriate template:

1. Open the POS Configurator.

2. Select **Reporting | Report Templates**.



3. Add a record number and assign it a value of 10000 or higher.

Note

MICROS assigns pre-defined reports a number of 9999 or below. By assigning the report record number a value of 10000 or greater, you ensure that upgrades or reinstallation of the POS system will not overwrite your entries.

4. Enter a **Name** for the report. *Be precise!* This is the name that will be displayed on the Reports folder in the Autosequences and Reports module. It is also used as a default header for the generated report.

5. Select a **Report Class** from the drop-down list. For this report, choose the report class that was added in the preceding section.

6. Make sure that the **40-column report**, **Hide on run screen**, and **Use List** checkboxes are blank. Clear the boxes, if necessary.

7. Enter the **Template Filename** where the pre-defined MICROS report is located. For this report, the filename is domain.rpt. You may browse to locate the file.

> ***Note***
>
> Report templates are normally stored in the Micros/res/pos/reports directory. If you store this template elsewhere, you must modify the path accordingly. Failure to properly specify the template location may result in an error message when the report is run.

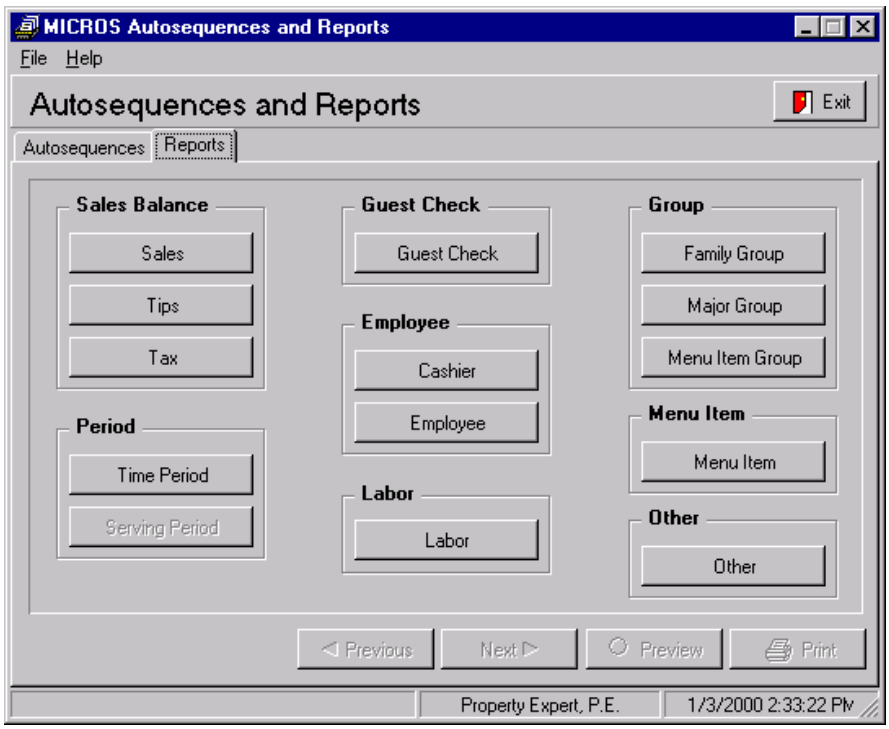
8. Save the changes and close the Report Template form.

Running the Report

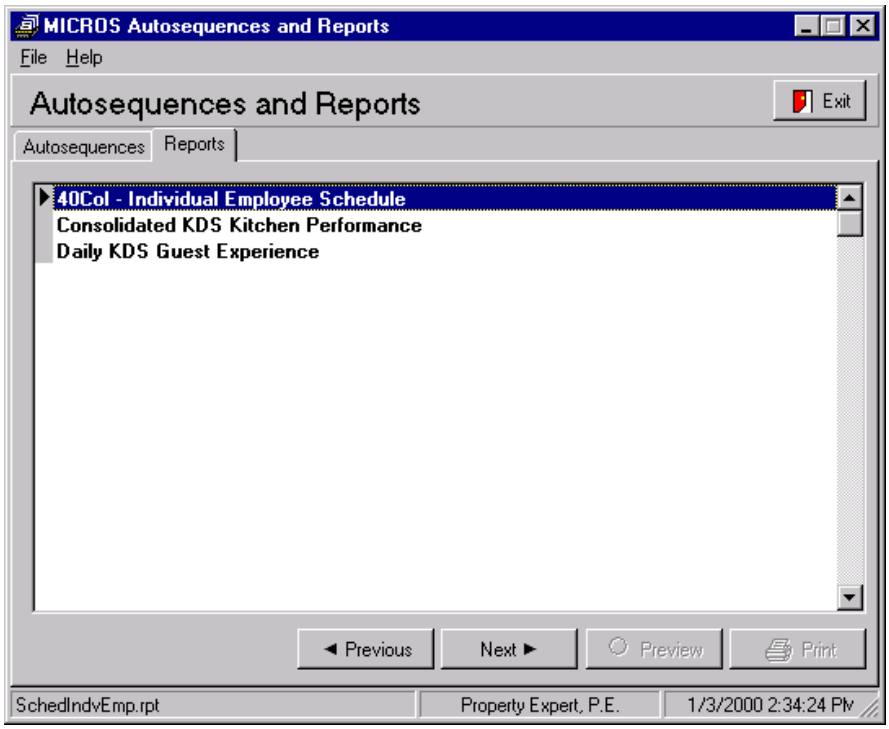
Reports are run from the Autosequences and Reports module. Once defined, they can be run manually or as part of an automated sequence.

Follow these steps to run the Database Domain Report:

1. Open the Autosequences and Reports module.
2. Select the **Reports** tab.



3. Click **Other**. The following screen displays:

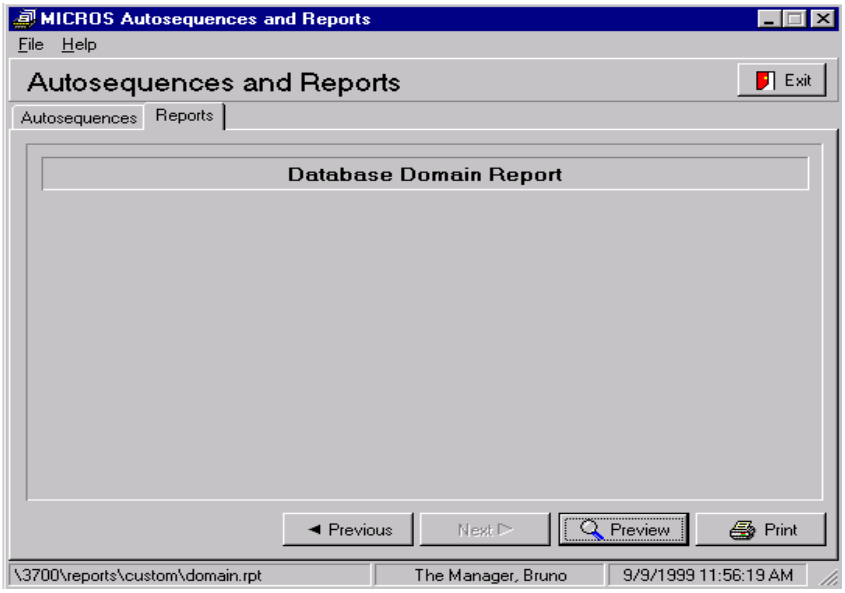


4. Select the report name from the list and click **Next**.

Note

For illustration, the report was named **Database Domain Report** in Step 4 of the preceding section. If you enter a different name, that name will appear in this list.

5. Since the report template neither requires nor accepts range values, the following screen may be ignored.



6. Click **Preview** or **Print** to run the report.

Note

The report will take a few minutes to generate. There are approximately 1200 rows that must be read into the Autosequences and Reports module before the report is displayed.

The Domain Report Template

This section describes the structure and output generated from the Domain Report Template as shown in the following sample:

MICROS Unified Database Domain Report
V2.60C
9/20/1999

Domain	Datatype	NULL Option	Default Value	Valid Value : Definition
ACCT_CLASS				
	smallint	N		1 : Asset 2 : Liability 3 : Owners Equity 4 : Income 5 : Expense
ACCT_PERIOD				
	smallint	Y		1 : Week 2 : Period 3 : Quarter 4 : Year
ACCTNG_CLOSE_TYPE				
	smallint	Y		0 : Soft 1 : Hard
ACCTNG_CYCLE_FRQ_TYPE				
	smallint	Y		0 : Monthly 1 : 4/4/5 2 : 5/4/4 3 : 4/5/4 4 : 13-4 5 : Event
ADDRESS_COUNTRY				
	char	Y		USA : USA CAN : CAN
ADDRESS_STATE				
	char	Y		AB : Alberta AL : Alabama AK : Alaska AS : American Samoa AZ : Arizona AR : Arkansas CA : California CO : Colorado CT : Connecticut DE : Delaware

The report includes the following columns of data:

- Domain (in bold)—This column lists the name of the database entry.
- Datatype—This column indicates the Sybase-defined data type for the entry.
- NULL Option—This column specifies whether a null value is allowed for the database entry. A ‘Y’ indicates the null is allowed. A ‘N’ indicates that it is not allowed.
- Default Value—This column displays the default value for the entry (if any).
- Valid Value: Definition—This column lists the accepted values for the entry.

Chapter 9

Database Tables

This chapter contains a list of the 3700 database tables. The table name, column names, datatype, primary key, foreign key, alternate key, and null options are provided for each table.

In this chapter

Types of Database Tables

The tables in the 3700 database fall into one of the following categories: definition, status, transaction, totals, or temp tables.

Definition Tables (*_def)

The definition tables define various aspects of the 3700 System, such as menu items, employees, devices, etc. These definition tables can be programmed using the POS Configurator module.

Status Tables (*_status)

The status tables are updated automatically in response to system events. For instance, the Employee Status table contains a column (Time Clock Status) that keeps track of when an employee clocks in or out. This table is updated whenever an employee clocks in or out.

Detail (*_dtl)

The detail tables contain information about POS transactions. For instance, as POS transactions occur, the type of transaction is written to the Transaction (trans_dtl) table.

Totals (*_ttl)

The totals tables contain accumulated totals information that reflect transaction activity, such as menu item sales and time period sales.

Temp (*_temp)

The temp tables contain information that is used internally by the 3700 system.

Overview

This chapter provides the following information for each 3700 database table.

- Logical Table Name

  The logical name for the database table. This name appears at the top of the page.

- Physical Table Name

  The physical name for the database table. This name appears just below the logical table name. Note that the Physical Table Name is recognized by the database.

- Logical Column Name

  The logical column name for each column contained in the database table. This name appears in the first column of the tables in this chapter.

- Physical Column Name

  The physical name for each column contained in the database table. This name appears in the second column of the tables in this chapter. Note that the Physical Column Name is recognized by the database.

- Column Datatype

  The datatype for the physical table column names. This name appears in the third column of the tables in this chapter.

- Primary Key (PK)

  PK is designated for every database table column that is a primary key. PK appears in the fourth column of the tables in this chapter.

- Foreign Key (FK)

  FK is designated for every database table column that is a foreign key. FK appears in the fifth column of the tables in this chapter.

- Alternate Key

  AK is designated for every database table column that is an alternate key and can be uniquely indexed. IE (index entry) is designated for every column that has a non-unique index. AK or IE followed by a number appears in the sixth column of the tables in this chapter.

- Null Option

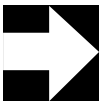
  NULL or NOT NULL is designated for every database table column. This option appears in the seventh column of the tables in this chapter.

Database Schema Report

The MICROS Database Schema Report allows you to view or print out a public schema of a MICROS database which can be used to write other custom reports, stored procedures, and database views.

Like all Crystal Reports, the Database Schema Report is setup through the POS Configurator and may be accessed via the Autosequences and Reports module. The process consists of the following steps:

- ❑ Adding a Report Class
- ❑ Assigning a Report Template
- ❑ Running the Report



Note

The report executable will not operate unless you have Crystal Reports v6.0 Professional software installed.

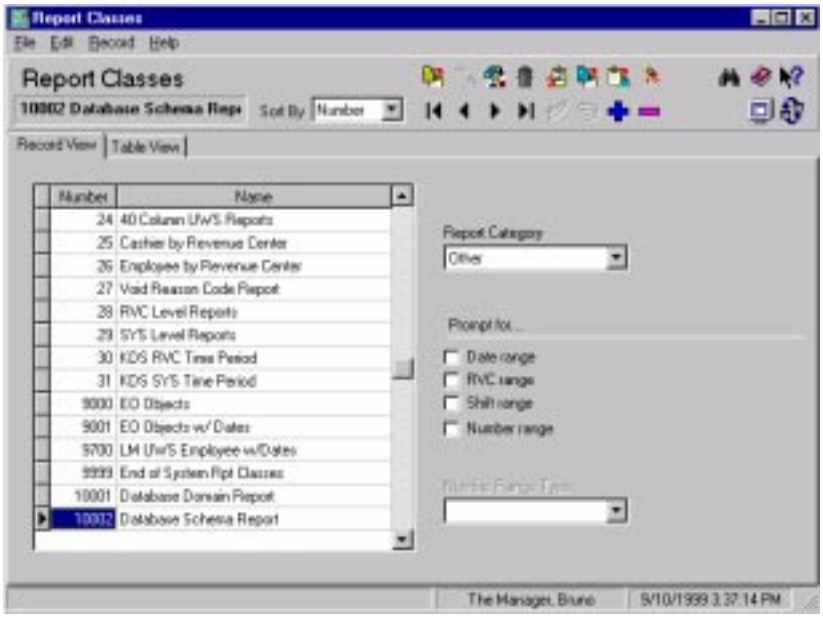
Adding a Report Class

Report classes allow you to assign similar attributes to a group of reports. The 3700 database is installed with a set of pre-defined report classes applicable to most POS operations. For some reports, however, the available options may not be relevant and a new report class may be required.

Follow these steps to add a new Report Class:

1. Open the POS Configurator.

2. Select **Reporting | Report Classes**.



3. Create a new report class by adding a record number and assigning it a value of 10000 or higher.

> ***Note***
>
> MICROS assigns pre-defined report classes a number of 9999 or below. By setting the new report class at 10000 or greater, you ensure that upgrades or reinstallation of the POS system will not overwrite your entries.

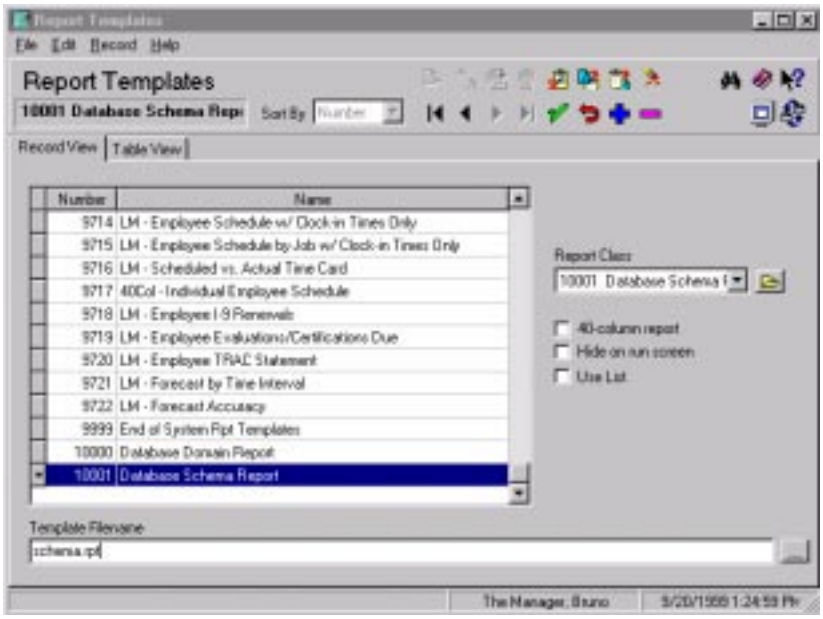
4. Enter a **Name** for the report class.
5. Select *Other* from the **Report Category** drop-down list.
6. Make sure that all of the **Prompt for...** checkboxes are blank. Clear the boxes, if necessary.
7. Save the changes and close the Report Classes form.

Assigning a Report Template

Report templates are pre-defined files that filter the data and determine how it will be displayed or printed when a report is generated. The 3700 System ships with a number of standard reports and templates, which can be accessed and configured to meet specific customer needs. To do this, the report template is linked to a report class, a new report name is added, and the template is assigned.

Follow these steps to create a new report and assign the appropriate template:

1. Open the POS Configurator.
2. Select **Reporting | Report Templates**.



3. Add a record number and assign it a value of 10000 or higher.

Note

MICROS assigns pre-defined reports a number of 9999 or below. By assigning the report record number a value of 10000 or greater, you ensure that upgrades or reinstallation of the POS system will not overwrite your custom entries.

---

4. Enter a **Name** for the report. *Be precise!* This is the name that will be displayed on the Reports folder in the Autosequences and Reports module. It is also used as a default header for the generated report.
5. Select a **Report Class** from the drop-down list. For this report, choose the report class that was added in the preceding section.
6. Make sure that the **40-column report**, **Hide on run screen**, and **Use List** checkboxes are blank. Clear the boxes, if necessary.

7. Enter the **Template Filename** where the pre-defined MICROS report is located. For this report, the filename is schema.rpt. You may browse to locate the file.

> ***Note***
>
> Report templates are normally stored in the MICROS\Res\Pos\Reports directory. If you store this template elsewhere, you must modify the path accordingly. Failure to properly specify the template location may result in an error message when the report is run.

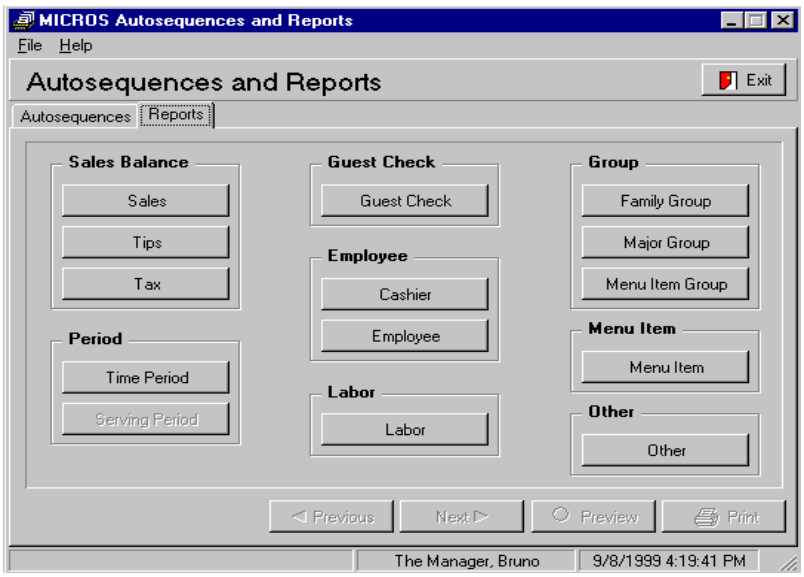
8. Save the changes and close the Report Template form.

Running the Report

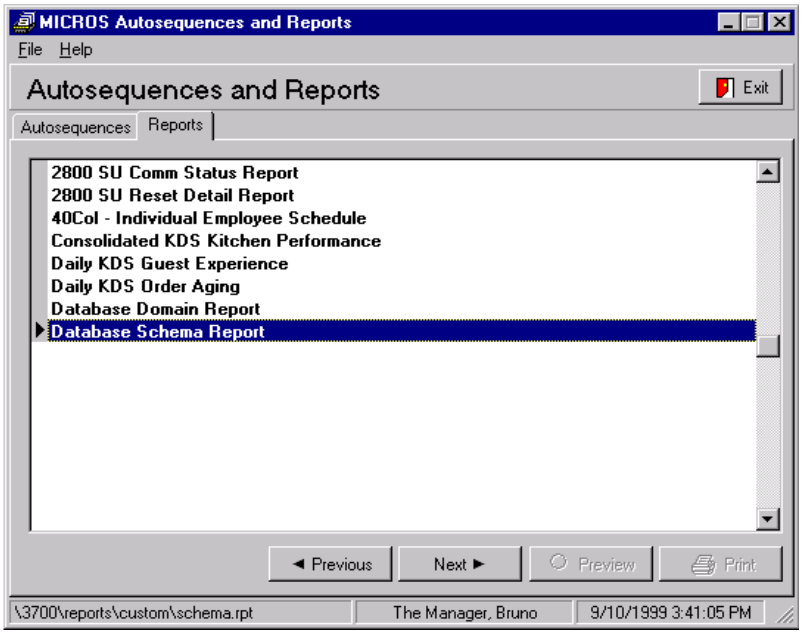
Reports are run from the Autosequences and Reports module. Once defined, they can be run manually or as part of an automated sequence.

Follow these steps to run the Database Schema Report:

1. Open the Autosequences and Reports module.
2. Select the **Reports** tab.



3. Click **Other**. The following screen displays:

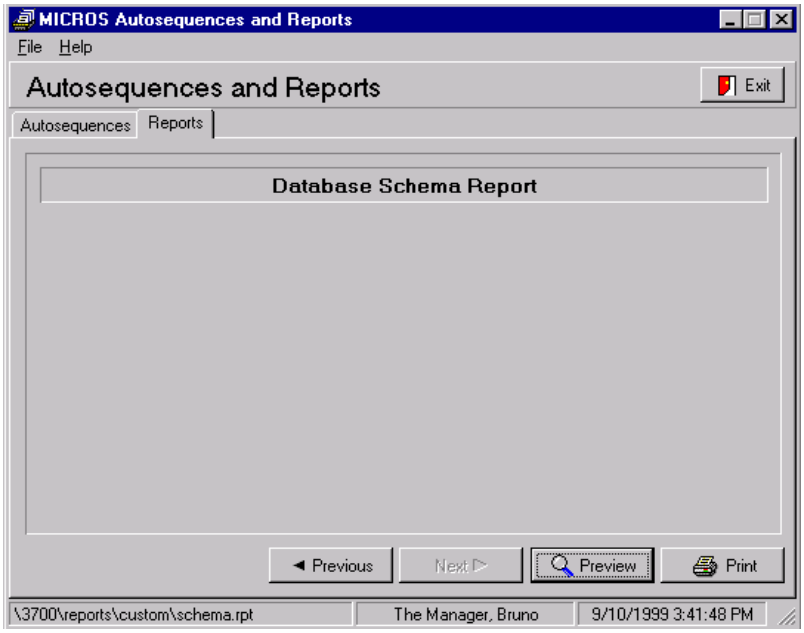


4. Select the report name from the list and click **Next**.

Note

For illustration, the report was named **Database Schema Report** in Step 4 of the preceding section. If you enter a different name, that name will appear in this list.

5. Since the report template neither requires nor accepts range values, the following screen may be ignored.



6. Click **Preview** or **Print** to run the report.

Note

The report will take a few minutes to generate. There are approximately 8400 rows that must be read into the Autosequences and Reports module before the report is displayed.

Using the Report Executable

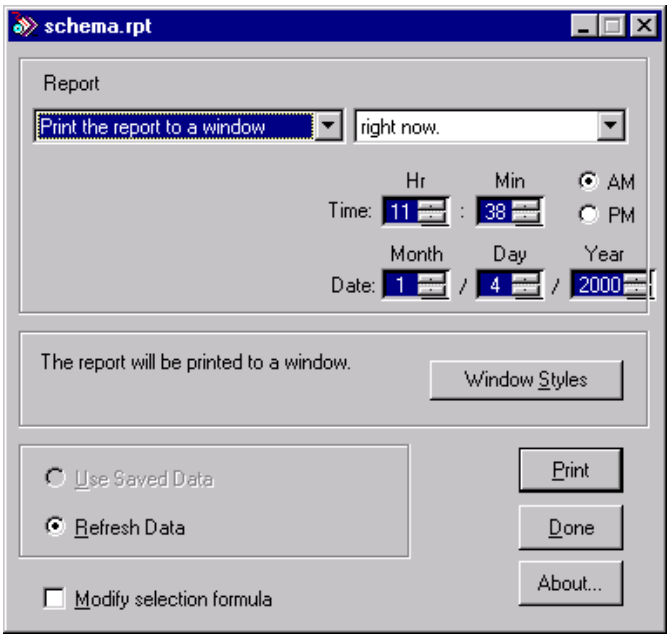
The Schema Report can be accessed directly from the Windows Start Menu by launching the report executable.

Note

To run the Schema Report executable, you must have Crystal Reports v6.0 Professional software already installed.

Follow these steps to run the report from the MICROS Application shortcut:

1. From the Windows Start menu, click on **Programs | MICROS Applications | Reports | Database schema report**.



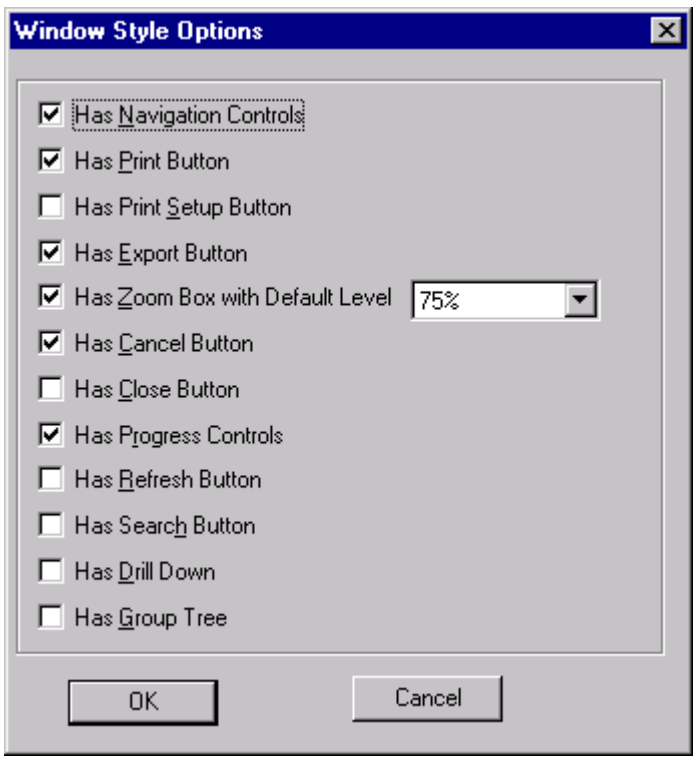
2. Select a **Report** display from the drop-down list. The options are:
   - ❑ Print the report to a window—previews the report on the screen
   - ❑ Export the report—saves the results to a file
   - ❑ Print the report to a printer—sends the report to a designated printer
3. Select a run-time from the drop-down list. The options are:
   - ❑ right now
   - ❑ in one hour
   - ❑ at 10:00 tonight
   - ❑ at midnight
   - ❑ at 2:00 tomorrow morning

- ❑ at 4:00 tomorrow morning
- ❑ at 6:00 tomorrow morning
- ❑ at a specific time

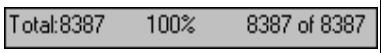
4. (Optional) If the report is to be run later, set the **Time** and **Date** using the appropriate list boxes.

5. (Optional) Click **Window Styles** to open the options screen. This screen allows you to choose the buttons and other features that will be displayed when the report is printed to a preview screen (see Step 2 on page 9-10).

   Check the appropriate boxes to make your selections. Click **OK** to save the selections.



The following table provides a sample icon as well as an explanation of the selected feature:

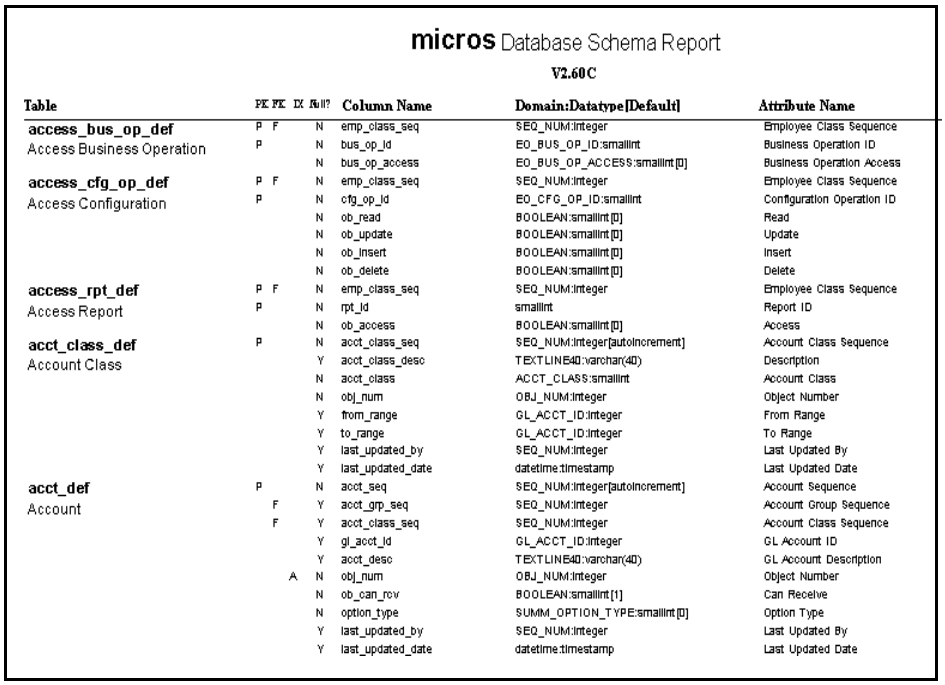
Icon	Description
	**Has Navigation Controls** Provides a scroll bar that allows you to move through the report, forward and backwards, one page at a time.
	**Has Print Button** Brings up a print dialog box that allows you to print the report.
	**Has Print Setup Button** Brings up a print setup dialog box that allows you to select the paper size, orientation, and default printer options.
	**Has Export Button** Exports the report to a variety of file formats.
	**Has Zoom Box with Default Level** Provides a drop-down list of percentages and allows you to set the default zoom level when the report is previewed.
	**Has Cancel Button** Cancels the current query and allows you to either close the schema report or refresh the database and continue.
	**Has Close Button** Closes the report query and returns you to the main screen.
	**Has Progress Controls** Shows the status of the current report query.

Icon	Description
	**Has Refresh Button** Rebuilds the report incorporating any changes made to the schema since the last query was submitted.
	**Has Search Button** Allows the user to search the report for a specified data string.
(not available)	**Has Drill Down**
	**Has Group Tree** Displays the entire schema on the left pane and allows the user to quickly jump to a database column instead of using the navigation controls.

6. Ignore the **Use Saved Data** and **Refresh Data** radio buttons as well as the **Modify Selection Formula** checkbox. (These options are a part of the underlying Sybase utility which is not accessible through this application.)
7. Click **Print** to run the report.
8. Click **Done** to exit the Database Schema report program.

The Schema Report

This section describes the structure and output generated from the Schema Report template as shown in the following sample:



micros Database Schema Report

V2.60C

Table	PK	FK	IX	Null?	Column Name	Domain:Datatype[Default]	Attribute Name
access_bus_op_def	P	F		N	emp_class_seq	SEQ_NUM:Integer	Employee Class Sequence
Access Business Operation	P			N	bus_op_id	EO_BUS_OP_ID:smallint	Business Operation ID
				N	bus_op_access	EO_BUS_OP_ACCESS:smallint[0]	Business Operation Access
access_cfg_op_def	P	F		N	emp_class_seq	SEQ_NUM:Integer	Employee Class Sequence
Access Configuration	P			N	cfg_op_id	EO_CFG_OP_ID:smallint	Configuration Operation ID
				N	ob_read	BOOLEAN:smallint[0]	Read
				N	ob_update	BOOLEAN:smallint[0]	Update
				N	ob_insert	BOOLEAN:smallint[0]	Insert
				N	ob_delete	BOOLEAN:smallint[0]	Delete
access_rpt_def	P	F		N	emp_class_seq	SEQ_NUM:Integer	Employee Class Sequence
Access Report	P			N	rpt_id	smallint	Report ID
				N	ob_access	BOOLEAN:smallint[0]	Access
acct_class_def	P			N	acct_class_seq	SEQ_NUM:Integer[autoIncrement]	Account Class Sequence
Account Class				Y	acct_class_desc	TEXTLINE40:varchar(40)	Description
				N	acct_class	ACCT_CLASS:smallint	Account Class
				N	obj_num	OBJ_NUM:Integer	Object Number
				Y	from_range	GL_ACCT_ID:Integer	From Range
				Y	to_range	GL_ACCT_ID:Integer	To Range
				Y	last_updated_by	SEQ_NUM:Integer	Last Updated By
				Y	last_updated_date	datetime:timestamp	Last Updated Date
acct_def	P			N	acct_seq	SEQ_NUM:Integer[autoIncrement]	Account Sequence
Account		F		Y	acct_grp_seq	SEQ_NUM:Integer	Account Group Sequence
		F		Y	acct_class_seq	SEQ_NUM:Integer	Account Class Sequence
				Y	gl_acct_id	GL_ACCT_ID:Integer	GL Account ID
				Y	acct_desc	TEXTLINE40:varchar(40)	GL Account Description
			A	N	obj_num	OBJ_NUM:Integer	Object Number
				N	ob_can_rcv	BOOLEAN:smallint[1]	Can Receive
				N	option_type	SUMM_OPTION_TYPE:smallint[0]	Option Type
				Y	last_updated_by	SEQ_NUM:Integer	Last Updated By
				Y	last_updated_date	datetime:timestamp	Last Updated Date

This report provides an alphabetical listing of all of the database tables grouped with the following columns of data:

- Table (in bold)—This column provides an alphabetical list of the names of all the database tables currently in the system.
- PK—A ‘P’ in the column indicates that the entry is used as a primary key.
- FK—An ‘F’ in the column indicates that the entry is used as a foreign key.
- IX—An ‘A’ in the column indicates that the entry is used as a index.
- Null?—This column specifies whether a null value is allowed for the database column entry. A ‘Y’ indicates the null is allowed. A ‘N’ indicates it is not allowed.
- Column Name—This column lists the names of all the columns found in the database table.
- Domain:Datatype [Default]—This column indicates the Sybase-defined data type for the entry.
- Attribute—This column provides a proper (or logical) name for each of the column names found in the database table.

Index

T

V

X