SQL Overview

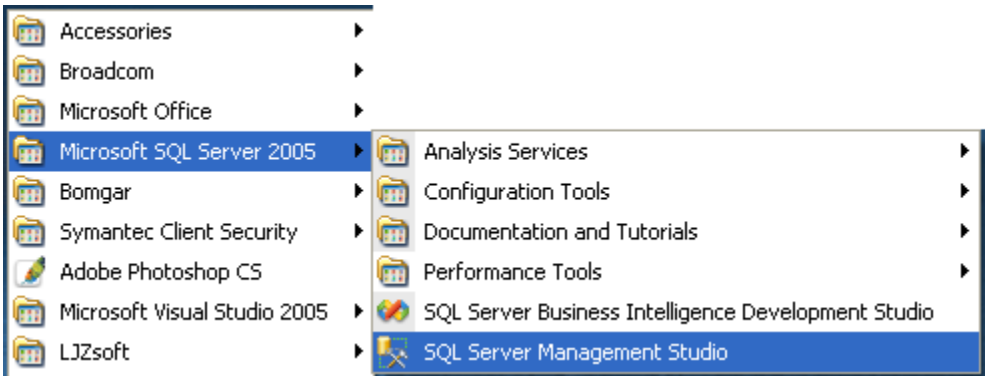
Contents

Microsoft SQL Server 2005 Studio Manager

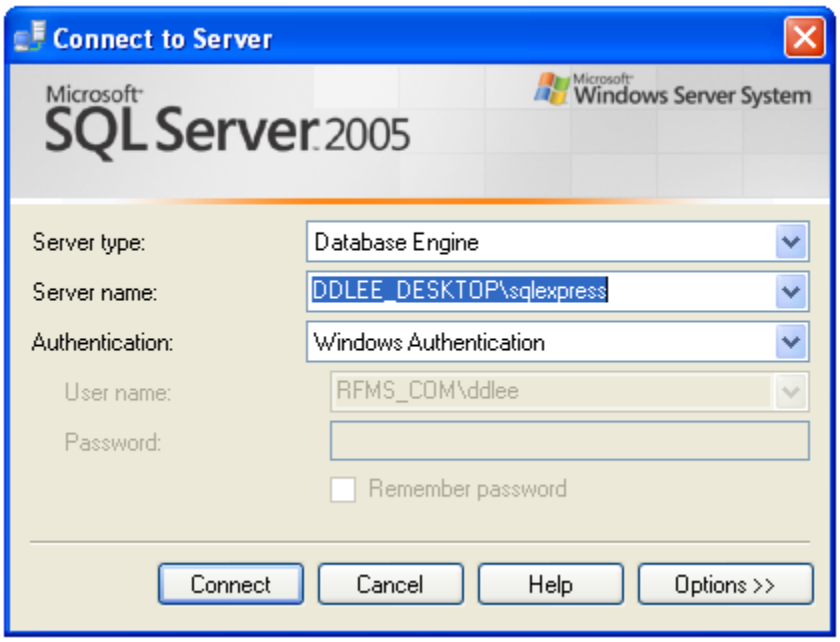
We will begin our study of the RFMS database by becoming familiar with the tool that Microsoft has given us to Query the Data and Metadata. The documentation will show examples from SQL Server 2005. If you are using SQL Server 2000 the methods will be the same but the interface will be different.

Note: In most installations the Studio Manager is not installed on individual workstations but can be found on the data server. Because of this any help that you may need with examples of your data can only be given if access to the server is available.

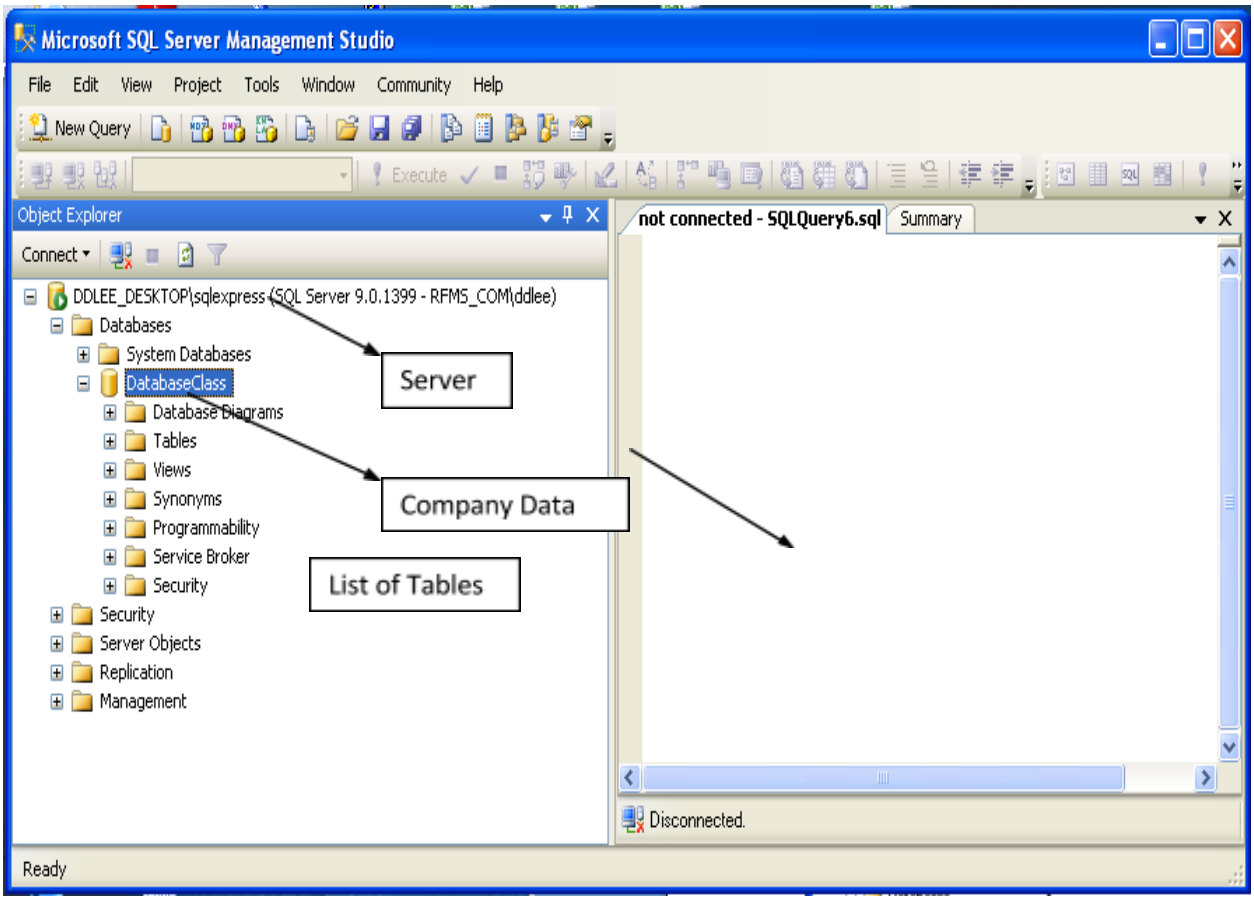
To run the Studio Manager select it from the Start button.



Window Authentication is the easiest way to log in. This requires that you are logged into the data server as the administrator.



Let's take a tour of the Studio Manager.



Server: The Studio Manager can manage many servers on your network. Only one is shown here but we could connect to other servers and list the databases being managed by those servers.

Database/Company Data: This is the database that stores the data that is entered when inventory is received, orders entered, vendor paid, journals closed etc. Almost everything done in RFMS is stored in a database on the data server.

List of Tables: Every database contains a group of tables. These tables hold the data and it is the knowledge of these tables, their structure and their relationship to each other that will enable the creation of meaningful report from RFMS data.

Metadata

Metadata refers to the data that describes the tables in the database. The database is self describing meaning that questions as to column names, data types, precision of data (size) etc. can all be found by asking (Querying) the database. Let's take a moment and define some more terms that will be needed.

Column: Refers to a piece of data that is stored in the database. Examples of columns are Invoice Number, Customer Name, Order Total etc.

Row: Refers to a group of columns.

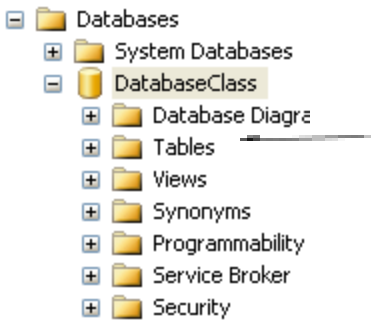
Below is an example of rows and columns from one of the APDetail table in RFMS.

	SeqNum	APSeqNum	Act	Sub	Amt	Store_Loc
1	1	1	233	0	523.01	32
2	2	2	140	1	874.8	32
3	3	3	140	4	33049.71	32
4	4	3	140	7	8700	32
5	5	4	140	4	2211.72	32
6	6	4	140	7	26700	32
7	8	6	565	0	75	32
8	9	7	565	0	65	32
9	10	8	527	0	25	32
10	11	9	538	0	218	32

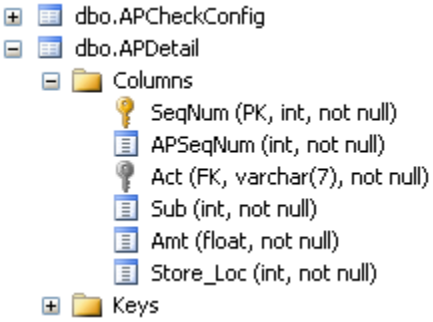
SeqNum,APSeqNum,Act,Sub,Amt,Store_Loc are all COLUMNS.

1,1,233,0,523.01,32 is a ROW.

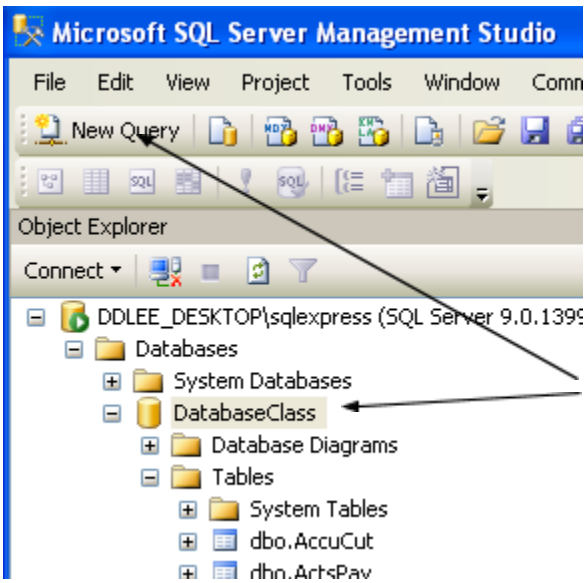
You can use the Studio Manager to find the Tables and the Columns and to display select Rows of data. To find the table names click on Tables in the tree.



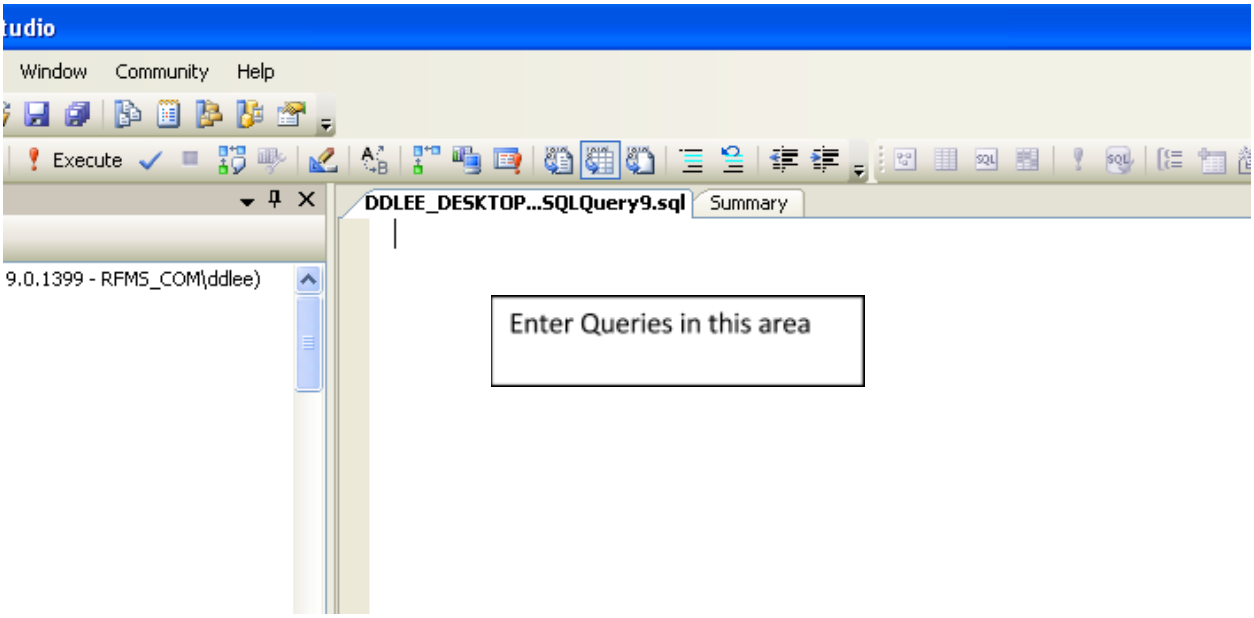
This will show the list of tables and to see the columns of a given table click on the table name and then click on columns.



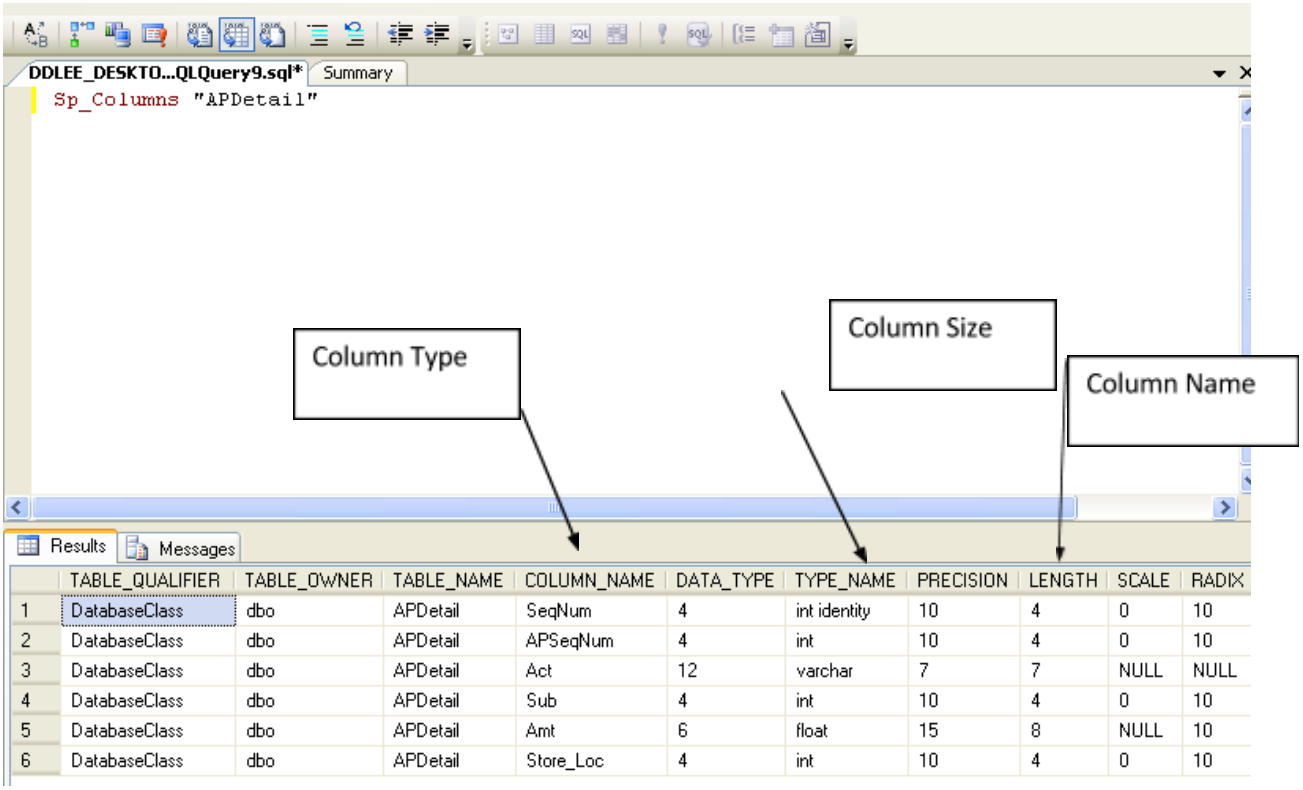
Another way to see the columns names would be to use the SP_COLUMNS “TableName” stored procedure. This is used by Querying the database. To Query the database we make use of the query window. To open a query window first make sure that the database that you want to work with is selected. Then click on “New Query”



This gives a window similar to:



If we enter SP_COLUMNS “APDetail” We will get this output.



	TABLE_QUALIFIER	TABLE_OWNER	TABLE_NAME	COLUMN_NAME	DATA_TYPE	TYPE_NAME	PRECISION	LENGTH	SCALE	RADIX
1	DatabaseClass	dbo	APDetail	SeqNum	4	int identity	10	4	0	10
2	DatabaseClass	dbo	APDetail	APSeqNum	4	int	10	4	0	10
3	DatabaseClass	dbo	APDetail	Act	12	varchar	7	7	NULL	NULL
4	DatabaseClass	dbo	APDetail	Sub	4	int	10	4	0	10
5	DatabaseClass	dbo	APDetail	Amt	6	float	15	8	NULL	10
6	DatabaseClass	dbo	APDetail	Store_Loc	4	int	10	4	0	10

Select Statement

The select statement is the heart of data reporting in SQL Server environment. It is used to ask the database questions and it outputs the data into the Studio Manager Results pane or to Crystal Reports or Excel or whatever application created the Query. For our purposes we will use the Studio Manager get familiar with the Select statement and its parts which are: FROM,WHERE,ORDER BY, GROUP BY and HAVING.

Parts of a Select Statement

SELECT column list FROM table WHERE conditions ORDER BY columns

Example:

Select Inv_Num,Cust_Name,Balance From Header Where Balance <> 0 Order By Cust_Name

	Inv_Num	Cust_Name	Balance
1	CG402254	BARRETT, MITCHELL	3727.56
2	CG200221	CABINESS, JIMMIE A.	165.86
3	CG200264	CAUSEY, MR. & MRS. LAWSON	352.43
4	CG101343	CHRIS	4.13
5	CG101727	COLLINS, MR. & MRS. TIM	1389.84
6	CG200278	DUCKS, CHARLES	2048.15
7	CG200279	DUCKS, CHARLES	600
8	CG200280	DUCKS, CHARLES	115.27
9	CG200222	FARNETTI, CHARLES	555.58
10	CG200276	GULF STATE PAPER CORPORATION	26000
11	CG100928	HASKINS, ALFRED	107.29

This statement will return the three columns specified for orders that have a non-zero balance and they will be in alphabetical order. What is we had needed that data in reverse alphabetical order? We would they add DESC after the order by. Example below:

Select Inv_Num,Cust_Name,Balance From Header Where Balance <> 0 Order By Cust_Name DESC

	Inv_Num	Cust_Name	Balance
1	CG101684	WOODS, DEBRA	403.17
2	CG200152	WILSON, STEVE	76.57
3	CG000784	WHEATS	878.74
4	CG602742	WHEAT, STEVE	82.02
5	CG700093	WHEAT, STEVE	23.75
6	CG800716	WHEAT, STEVE	363.96
7	CG000563	WHEAT'S	232.87
8	CG200277	TOWNSEND TOYOTA	8000
9	CG100168	THORNE, BURRELL M.	191.07
10	CG200112	STEVE	19.36
11	CG200252	ROHRER, MR. & MRS. STEPHEN D.	1604.04

Using the above example where the data is sorted from A to Z notice that there are three orders for "DUCKS, CHARLES".

Group By

What if we would like to have one row for each customer that showed the total balance for the customer? The GROUP BY will do this for us.

```
Select Cust_Name,Sum(Balance) From Header Where Balance <> 0
Group By Cust_Name
Order By Cust_Name
```

	Cust_Name	(No column name)
1	BARRETT, MITCHELL	3727.56
2	CABINESS, JIMMIE A.	165.86
3	CAUSEY, MR. & MRS. LAWSON	352.43
4	CHRIS	4.13
5	COLLINS, MR. & MRS. TIM	1389.84
6	DUCKS, CHARLES	2763.42
7	FARNETTI, CHARLES	555.58
8	GULF STATE PAPER CORPORATION	26000
9	HASKINS, ALFRED	107.29
10	HENRY, PATTY	145.89
11	HERRING	196.51

Notice now that there is one row for "DUCKS, CHARLES" and that the total is 2763.42 which is the 2048.15, 600.00 and 115.27 from his three invoices. Also notice that the Inv_Num column is not a part of the column list. If we added it back to the list the GROUP BY would have to include it and it would have listed all of the rows anyway showing the balance on each order.

Having

If we would like to see the Customers who have a balance OVER $500.00 we would use the HAVING statement. The same query but with the HAVING added would be:

```
Select Cust_Name,Sum(Balance) From Header Where Balance <> 0
Group By Cust_Name
```

Having Sum(Balance) > 500.00
Order By Cust_Name

	Cust_Name	(No column name)
1	BARRETT, MITCHELL	3727.56
2	COLLINS, MR. & MRS. TIM	1389.84
3	DUCKS, CHARLES	2763.42
4	FARNETTI, CHARLES	555.58
5	GULF STATE PAPER CORPORATION	26000
6	HUGHES, DEBBIE	677.54
7	KILBOURNE, DAVID	1844
8	KIMBRELL, MR. & MRS. R. O.	559.78
9	LAKE, JERRELL L.	970.99

Notice that "CABINESS, JIMMIE A.","CAUSEY, MR. & MRS. LAWSON", and "CHRIS" are now gone. Each of these customers had balances less than $500.00.

Join

There are times when you need data from more than just one table. For example, what if you needed to see the material sold to a customer and you want the customer's name also. The material is stored in one table and the customer information is stored in another. This is known as a Parent/Child relationship. In this example the Header table is the Parent and the Custline table is the child. One Parent and have many children. A single Invoice Number can have many material lines. We will later show the RFMS table relationships but for now let's look at an example.

Select Cust_Name,A.Inv_Num,Pr_Code,Style,Color From Header A Join Custline B On A.Inv_Num = B.Inv_Num

	Cust_Name	Inv_Num	Pr_Code	Style	Color
1	MARSHALL, MR. & MRS. GENE	CG000018	01	SP-42	300 BRAMBLE
2	MARSHALL, MR. & MRS. GENE	CG000018	04	PAD - $3.00 6LB.	$3.00 6LB. REBOND
3	MARSHALL, MR. & MRS. GENE	CG000018	02	AURORA	41099 SAHRA DUSK
4	MARSHALL, MR. & MRS. GENE	CG000018	08	SEAM SEALER MANNINGT	LOW GLOSS
5	SMALLEY, JIMMY	CG000180	01	MACK	00026 NEW SUEDE
6	SMALLEY, JIMMY	CG000180	04	PAD $3.00 6LB.	$3.00 6LB. REBOND
7	SMALLEY, JIMMY	CG000180	02	TRILLICK TILE	14034 BEIGE
8	HUMPHRIES, ELIZABETH J.	CG000295	01	SIERRA	734 BEIGE
9	HUMPHRIES, ELIZABETH J.	CG000295	04	PAD $3.00 6LB.	$3.00 6LB. REBOND

Notice that Invoice Number "CG000018" is listed four times; once for each material (or service) line on that invoice.

In the query statement above we had A.Inv_Num and Header A and Custline B. When we add an identifier after a table name we can then refer to columns from that table using the id.column syntax. This is known as giving the table name an alias. Another valid way to do this query would be:

Select Cust_Name,Header.Inv_Num,Pr_Code,Style,Color From Header Join Custline On Header.Inv_Num =Custline.Inv_Num

Here we use the table name as an ID. One of these two methods has to be used when the tables contain columns that are named the same. It is the way to let the SQL Server know which of the columns to return or to join on or to use in the where clause.

Overview of Supporting Tables

If you look under the Tables tab of the RFMS database you will see that there are dozens of tables. Some of these tables hold the core data i.e. Orders and Payables. Others hold information to make those core tables complete. For example, the Header (order) table contains a column called Job_Type. This columns links to a table called JobType and more specifically to a column called JobType.JobType. These tables are referred to a supporting tables. These tables are:

STORE This is used to hold the Store Code and it's supporting information. This information includes the store's address and phone numbers as well as how to charge tax and how much tax to charge.

SLMNMAST This is a list of Sales Reps. It also includes which commission schedule to use a default for this sales person.

JOBTYPE This table contains the job/customer types that can be assigned to orders.

These three tables are outlined in the Visio Database Structure Document on the Supporting Tables 2 Tab.

CHTACTS Holds the Accounting codes used in the RFMS system.

SUPPLIER Supplier information including address and telephone. Also contains YTD numbers for the supplier.

PRODCODE Holds the descriptions for the inventory Product Codes

These three tables are outlined in the Visio Database Structure Document on the Supporting Tables 2 Tab.

CUSTOMER Holds information related to the customer including name, address and telephone information. A default store, customer type and sales person are on this table and that information is selected from the STORE, JOBTYPE and SLMNMAST tables.

This table is outlined in the Visio Database Structure Document on the Supporting Tables 3 Tab.

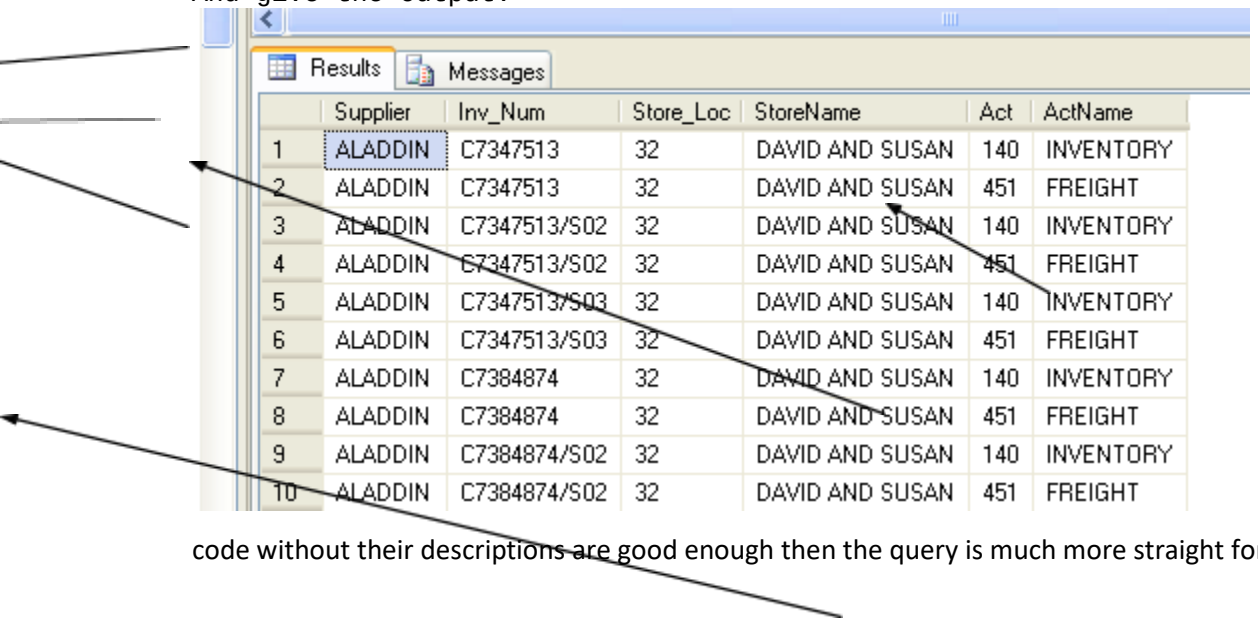
Overview of Core RFMS Tables

A/P

The a/p module primarily consist of two tables ACTSPAY (Parent) and APDETAIL (Child 1-M) and APDETAIL uses STORE and CHTACTS as supporting tables. The ACTSPAY table uses SUPPLIER as a supporting table although it is not mandatory that the Supplier on the ACTSPAY table be in the SUPPLIER table. Lets write a query that Joins ACTSPAY with APDETAIL and prints the Store Name and the Account Code Description for a given supplier, namely ALADDIN. The APDETAIL is related to the ACTSPAY table by ACTSPAY.SEeqNum = APDETAIL.APSeqNum. Using this the query would take the form:

```
Select Supplier,Inv_Num,Store_Loc,StoreName,Act,ActName From Actspay A
Join APDetail B On A.SeqNum = B.APSeqNum Join Store C On B.Store_Loc = C.Store
Join ChtActs D On B.Act = D.ActCode
Where A.Supplier = 'ALADDIN'
```

And give the output:



Results | Messages

	Supplier	Inv_Num	Store_Loc	StoreName	Act	ActName
1	ALADDIN	C7347513	32	DAVID AND SUSAN	140	INVENTORY
2	ALADDIN	C7347513	32	DAVID AND SUSAN	451	FREIGHT
3	ALADDIN	C7347513/S02	32	DAVID AND SUSAN	140	INVENTORY
4	ALADDIN	C7347513/S02	32	DAVID AND SUSAN	451	FREIGHT
5	ALADDIN	C7347513/S03	32	DAVID AND SUSAN	140	INVENTORY
6	ALADDIN	C7347513/S03	32	DAVID AND SUSAN	451	FREIGHT
7	ALADDIN	C7384874	32	DAVID AND SUSAN	140	INVENTORY
8	ALADDIN	C7384874	32	DAVID AND SUSAN	451	FREIGHT
9	ALADDIN	C7384874/S02	32	DAVID AND SUSAN	140	INVENTORY
10	ALADDIN	C7384874/S02	32	DAVID AND SUSAN	451	FREIGHT

From APDETAIL

From STORE

From APDETAIL

From CHTACTS

From ACTSPAY

If store code and account code without their descriptions are good enough then the query is much more straight forward:

```
Select Supplier,Inv_Num,Store_Loc,Act From Actspay A
Join APDetail B On A.SeqNum = B.APSeqNum
Where A.Supplier = 'ALADDIN'
```

Giving:

	Supplier	Inv_Num	Store_Loc	Act
1	ALADDIN	C7347513	32	140
2	ALADDIN	C7347513	32	451
3	ALADDIN	C7347513/S02	32	140
4	ALADDIN	C7347513/S02	32	451
5	ALADDIN	C7347513/S03	32	140
6	ALADDIN	C7347513/S03	32	451
7	ALADDIN	C7384874	32	140
8	ALADDIN	C7384874	32	451
9	ALADDIN	C7384874/S02	32	140

Some columns of interest in ACTSPAY are:

Disc_Rate : A percentage number to multiply with Discountable to get the Discount amount for paying the A/P by the Due_Date

Due_Date: See above. All but a few dates in RFMS are stored as string data in the form of YYYYMMDD even is using the English Date format.

Inv_Date: This is the date that the A/P will hit the General Ledger.

Trans_Date: Date the A/P was entered into RFMS.

Discountable: Amount to apply Disc_Rate against.

Non_Disc: Fixed Amount of Payable

Inv_Total: Amount of payable after discount has been applied. (Discountable + Non_Disc) – (Discountable * Disc_Rate)

Chk_Num: Check number from the banking system that was used to satisfy this payable. Many payables can be satisfied with one check. Is this field is empty then the payable has not yet been paid.

Chk_Act: Account number from CHTACTS that the check was written from. If blank then payable has not yet been paid.

Staus: Empty AND Chk_Num Empty = Open A/P, “V” = Void, “F” = Flagged for check writing, “H” hold.

Date_Paid: Date Check Created to satisfy A/P

Example Query to show all open A/Ps.

Select * From Actspay Where status = '' AND Chk_Num = ''

	SeqNum	Tagg...	Inv_Num	Inv_Date	Supplier	Due_Date	Disc_Rate	Discountable	Inv_Tot	Non_Disc	Da
1	2	0	758209	20040305	SHAW INDUSTRIES-CORE	20040305	0	0	874.8	874.8	
2	3	0	78E90	20040305	CARPENTER CO.	20040305	0	0	41749.71	41749.71	
3	4	0	897-8	20040319	CARPET WHOLESALE	20040408	0	0	28911.72	28911.72	

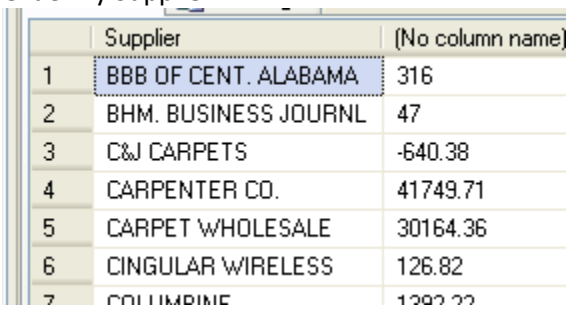
Show all Voided A/P

Select * From Actspay Where status = 'V'

	SeqNum	Tagg...	Inv_Num	Inv_Date	Supplier	Due_Date	Disc_Rate	Discountable	Inv_Tot	Non_Disc
1	18	0	1ST QT*VD-01	20010515	ALABAMA HOMEBUILDERS	20010515	0	-1053.52	-1053.52	0
2	19	0	1ST QTR 01	20010515	ALABAMA HOMEBUILDERS	20010515	0	1053.52	1053.52	0
3	85	0	400171*VD-01	20010403	BBB OF CENT. ALABAMA	20010322	0	-310	-310	0
4	86	0	4001710	20010220	BBB OF CENT. ALABAMA	20010322	0	310	310	0
5	112	0	020101	20010201	CARPET ONE ROYALTY	20010201	0	1230.88	1230.88	0

Show Outstanding Payables summed by supplier

```
Select Supplier,Sum((Discountable + Non_Disc) - (Discountable * Disc_Rate))
From Actspay Where status = '' AND Chk_Num = ''
Group By Supplier
Order By Supplier
```



	Supplier	(No column name)
1	BBB OF CENT. ALABAMA	316
2	BHM. BUSINESS JOURNL	47
3	C&J CARPETS	-640.38
4	CARPENTER CO.	41749.71
5	CARPET WHOLESALE	30164.36
6	CINGULAR WIRELESS	126.82
7	COLUMBINE	1382.22

All Invoices that where posted to the Inventory Account Code (140 in this data set)

```
Select Supplier,Inv_Num From Actspay A
Join APDetail B On A.SeqNum = B.APSeqNum
Where Act = 140 AND Status <> 'V'
```

	Supplier	Inv_Num
1	SHAW INDUSTRIES-CORE	758209
2	CARPENTER CO.	78E90
3	CARPENTER CO.	78E90
4	CARPET WHOLESALE	897-8
5	CARPET WHOLESALE	897-8
6	ALADDIN	C7347513
7	ALADDIN	C7347513/S02
8	ALADDIN	C7347513/S03
9	ALADDIN	C7384874
10	ALADDIN	C7384874/S02

Notice that no data is shown from the APDETAIL table. It had to be joined so that the WHERE clause could look at the ACT column.

Banking (CK tables)

There is a list of tables in the RFMS database that are named CK followed by an account number such as CK110. These are the banking tables. These tables contain checks, deposits, bank charges and transfers for a given bank account. CK Tables are supported by the SUPPLIER table. It is possible to have a check where the supplier is not in the CK table. Columns of interest include:

Chk_Num: Check number left padded with spaces. 1001 would be stored as ‘ 1001’. That is two leading spaces.
Chk_Date: Date check was posted. Not the transaction date but rather a GL date.
Payee:
Supplier:
Amount: Check is for this amount
Amount_Type: ‘ C’=Check,’D’=Deposit
Trans_Type: ‘C’=Check,’D’=Desposit,’S’=Bank Charge.
Status: Empty not Reconciled, ‘R’ = Reconciled.
Balance: Running balance of the check register.

Journal

The journal table contains rows of data that come from two separate streams. The first are transactions that post directly into the Journal. These include single check postings, deposits, check voids and journal batch entries. The second are month end batch entries and examples of these include the expense debits for payables, A/R, COGS,inventory etc. Columns include:

Act_Code: Code that the Journal transaction is posted to.
Trans_Amt: Amount
Trans_Type: D=Debit,C=Credit
Description: Varies
_Cycle: Empty=Posted directly to Journal; Char(255) = Posted by month-end close
Trans_Date: Date the posting occurred
Store_Loc: Store Code

GL

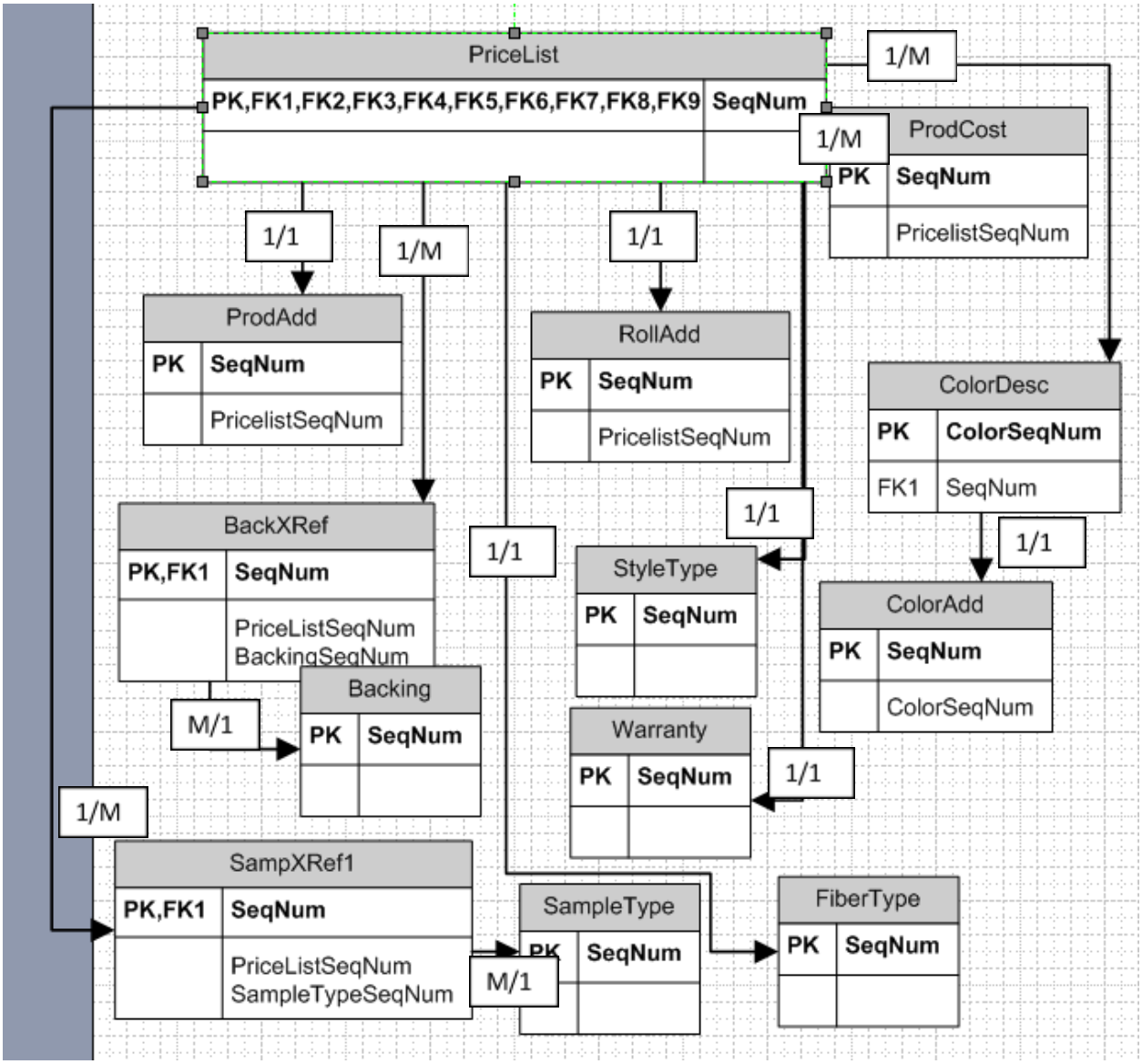
The General Ledger can only be closed after the month has been closed and that is triggered by the journal close. The GL table contain summary of the closed journal table. There is one record per account code/store code combination. Columns:

Trans_Date: This will always be the last day of the month for the month being closed. The close for March 2005 would be '20050331'
Act_Code: Account code
Description: Either Blank or 'END OF YEAR – MM/YY'. The End Of Year marks a year end close. An example is below to get the last year end close.
Beg_Bal: Beginning balance for this Account/Store/ Trans_Date
Debit: All Debits for this Account/Store/ Trans_Date
Credit: All Credits for this Account/Store/ Trans_Date
End_Bal: Ending balance for this Account/Store/ Trans_Date
Store_Loc: Store

To get the last year end close:
Select Max(Trans_Date) From GL Where Description Like 'END OF YEAR%'

Products Module

The products module is made up of many tables. A diagram showing the tables and the relationships is show below.



Some notable columns from the Pricing Module are:

Pricelist.Supplier_Style: What the supplier call this product. Purchase Orders and Products are the only places RFMS show this data. This is referred to as the Cross Over Data. Many stores keep this information away from not management employees.

Pricelist.Private_Style: What this product is called in your store. Order Entry and Inventory show this information.

ColorDesc.ColorDesc: Supplier Color
ColorDesc.PrivateColor: Private Label Color

If we want to find all of the Products that have WHITE as part of the Private Color we would execute this query:
Select A.Private_Style,B.PrivateColor From Pricelist A Join ColorDesc B On A.SeqNum = B.SeqNum
Where PrivateColor Like '%WHITE%' which produces:

	Private_Style	PrivateColor
1	4249	TULIP WHITE
2	8106	STERLING WHITE
3	8106	ANTIQUE WHITE
4	ADELAIDE	WHITE WHEAT
5	AESTHETICS (B)	WHITE SHELL
6	ALL AMERICAN	ALMOND WHITE
7	ALLENWOOD	WHITE FOSSIL
8	ALLENWOOD	LANCASTER WHITEWASH
9	ALPENAH	ANTIQUE WHITE
10	AMBASSADOR	VINTAGE WHITE
11	AMBASSADOR	CLASSIC WHITE

If we wanted to limit this to Private_Styles that start with 'CA' we would write the query as:

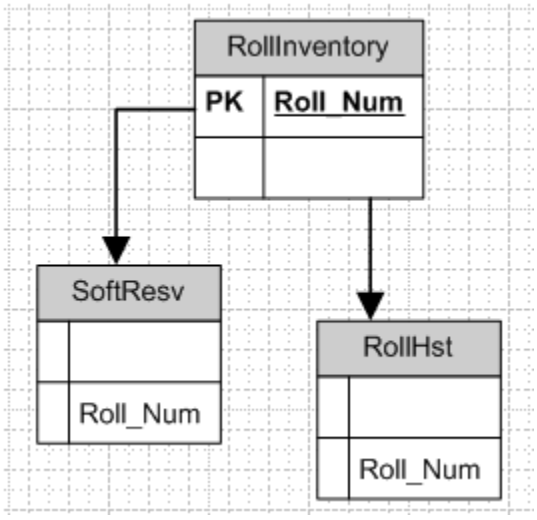
```
Select A.Private_Style,B.PrivateColor From Pricelist A Join ColorDesc B On
A.SeqNum = B.SeqNum
Where PrivateColor Like '%WHITE%' AND Private_Style Like 'CA%'
```

Which would produce:

	Private_Style	PrivateColor
1	CABERNET	BONE WHITE
2	CARNIVAL (WOR)	GLAZED WHITE
3	CAPTIVATING	WHITE FLURRY
4	CASTLE PINES	MILKY WHITE
5	CARNIVAL -CO6040	GLAZED WHITE
6	CALM CREEK	ANTIQUE WHITE
7	CANCUN	DREAM WHITE
8	CASIENDA	WHITE CHOCOLATE
9	CASTLE PINES	MILKY WHITE
10	CASUAL TOUCH	PANDA WHITE

Roll Inventory

The roll inventory module is comprised of three tables; ROLLINVENTORY, ROLLHST, and SOFTRESV. The last table is also used by the Item Inventory Module. There is only on record per roll number. The cut information is stored in the ROLLHST table.

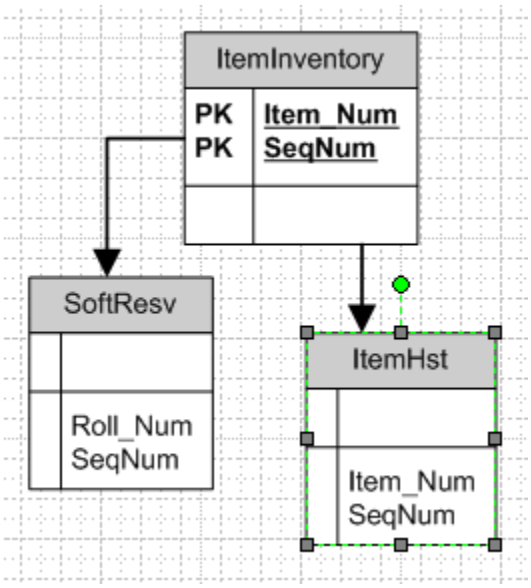


Columns of interest.

RollInventory.Roll_Num: The Roll Number that was used during receiving. This data moves with the roll throughout the entire system. It is used on the CustLine table in Order Entry and reported on from various points in RFMS.
RollInventory.Style: The private style of the roll.
RollInventory.Color: The private color of the roll.
RollInventory.PONumber: The concatenation of PO/Line number from the Purchase Order Module.
RollInventory.Sidemark: This is a column that RFMS may auto populate if the user choose to.
RollInventory.Supplier / Inv_Num / Inv_Date: The Invoice Information from Costing. If Inv_Date is empty then the Row has not been costed (vouchered) .
RollInventory.Date_Rcvd: Posted from Receiving.
RollInventory.Beg_Amt / Amt_Used / Amt_Resv / Amt_Avail: Describes roll balance information.
RollInventory.Location: Warehouse Location
HISTORY
RollHst.Roll_Num: This is the roll number field and it ties this history record to the roll.
RollHst.Status: 'O' on order, 'C' Cut, 'R' Reserved, 'D' Delivered
RollHst.Amt_Used: Amount on this cut

Item Inventory

The item inventory module is comprised of three tables; ITEMINVENTORY,ITEMHST , and SOFTRESV. The last table is also used by the Roll Inventory Module. There is only on record per roll number. The usage information is stored in the ITEMHST table.



Columns of interest.

ItemInventory.Item_Num / SeqNum: The Item Number that was used during receiving. This data moves with the Item throughout the entire system. It is used on the CustLine table in Order Entry and reported on from various points in RFMS.
ItemInventory.Style: The private NAME of the item.
ItemInventory.Color: The private DESCRIPTION of the item.
ItemInventory.PONumber: The concatenation of PO/Line number from the Purchase Order Module.
ItemInventory.Sidemark: This is a column that RFMS may auto populate if the user choose to.
ItemInventory.Supplier / Inv_Num / Inv_Date: The Invoice Information from Costing. If Inv_Date is empty then the Row has not been costed (vouchered) .
ItemInventory.Date_Rcvd: Posted from Receiving.
ItemInventory.Qty_Recv / Qty_Used / Qty_Resv / Qty_Avail: Describes item balance information.
ItemInventory.SerialNo: This is a column that RFMS allows to be user defined.
ItemInventory.Location: Warehouse Location

HISTORY

ItemHst.Trans_Type: [Physical A=New, B=Adjustment,C =Sold,D=Returned] [On Order H=New,I=Adjustment,J=Sold,K=Returned]

ItemHst.Inv_Number: May refer to Supplier Invoice or Customer Invoice based on Trans_Type. If Trans_Type = 'A'or 'H' then it is Supplier Inv_Num if Trans_Type anything else it is Customer Invoince Number.

ItemHst.Grs_Cost: If this is a Sold history then this is the cost at the time it happened. Updating Item Inventory during the Costing process can change the cost of the item but it will not change this cost.

Soft Reserve

The soft reserve table holds soft reserves placed by sales people. This table handles both items and rolls. There can be multiple soft reserves on a single roll or item.

Columns of interest.

SoftResv.Roll_Num: The roll number if Product is a Roll OR the Item_Num is it is an item.
SoftResv.ItemSeqNum: The SeqNum from ItemInventory unless it is a roll then it is zero.
SoftResv.Salesman: The sales person that has an potential sales using this inventory.

Purchase Order

The PurOrder table contains the information needed to order and receive inventory. The process of creating a PO brings data from the products table (both supplier and private information) and pushes it into the Roll or Item tables (private only). The purchase order DOES contain the "cross over" information as does the products module. The rest of the system (Inventory,OE) only present private information to the users.

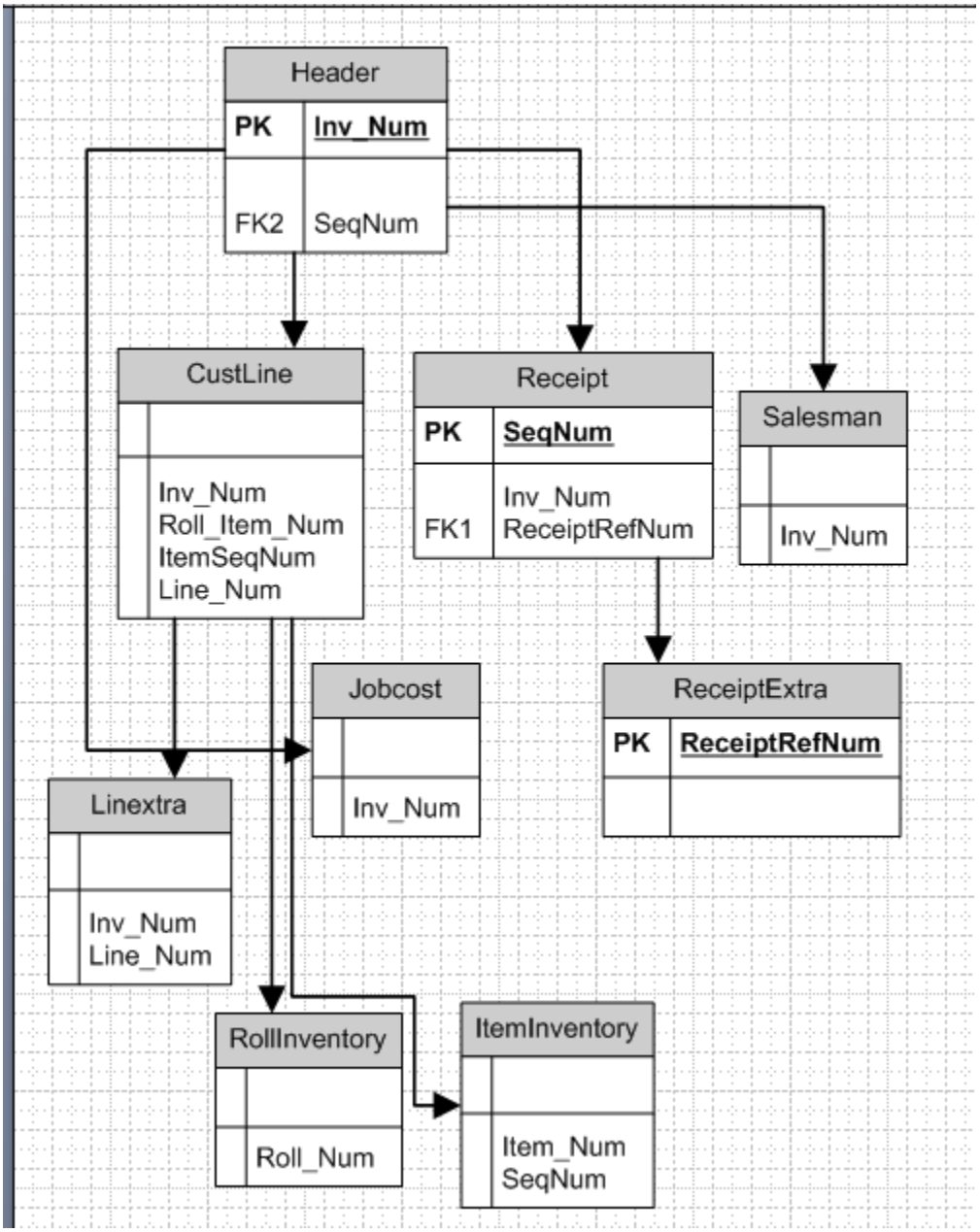
Columns of interest.

PurOrder.PONum: key field example #ST03843
PurOrder.POLine: integer example 1
PurOrder.AmtOrdered: The total amount ordered
PurOrder.AmtReceived: The amount that has already been received. It is possible that the po will be satisfied with multiple shipments.
PurOrder.ItemNum: This will be either a roll number or an item number and matches a roll record on Roll_Num or it is an item number that matches an item record on Item_Num/SeqNum (using PurOrder.ItemSeqNum)
PurOrder.ItemSeqNum: SeqNum from item inventory
PurOrder.Sidemark: This field can be auto-populated during the Auto PO Generate process.
PurOrder.ColorSeqNum: Ties the po record back to the Pricelist/Colordesc record that was used to create the po. (May not have a value).
PurOrder.TakenBy: User Entered. When this field is blank then the PO is "OPEN". Once a value is here then the PO is "On Order".
PurOrder.Status: 'O' Open is takenby is blank or On Order is takenby has a value; 'S' Satisfied,'C' Cancelled, 'B' Back Ordered.

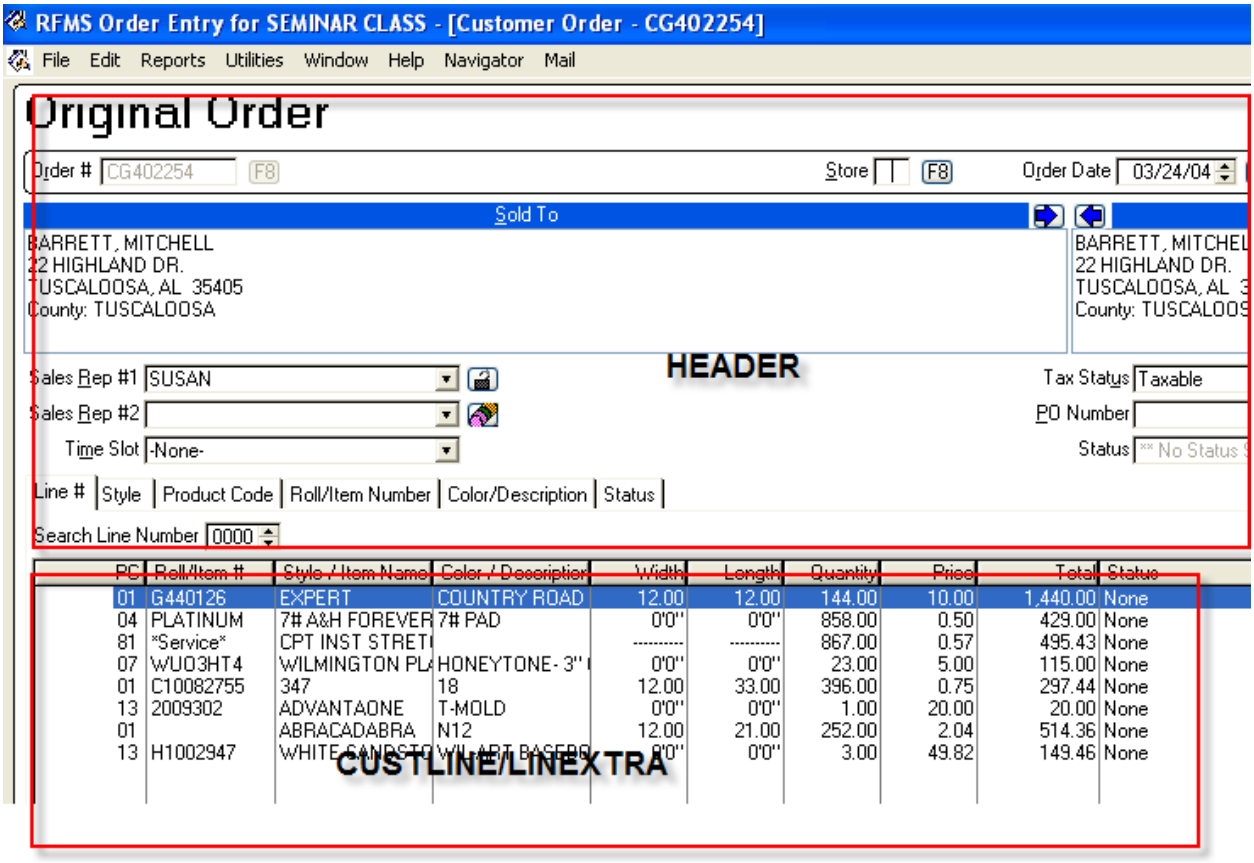
Sales

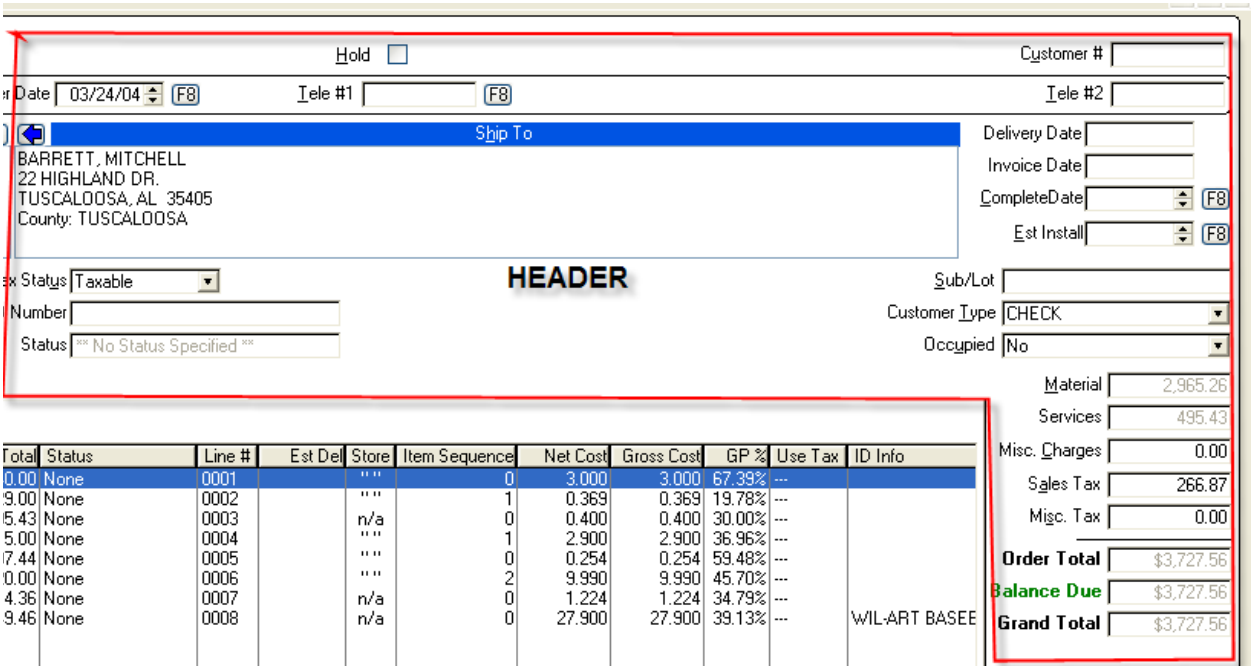
The tables that make up the sales information are probably the most report rich tables in the RFMS database. This contains the Order (HEADER), Inventory/Services sold (CUSTLINE and LINEXTRA), Payments (RECEIPT and RECEIPTEXTRA), the jobcost information (JOBCOST) and the sales commissions (SALESMAN). The general layout of this system is shown below. Keep in mind that he supporting tables

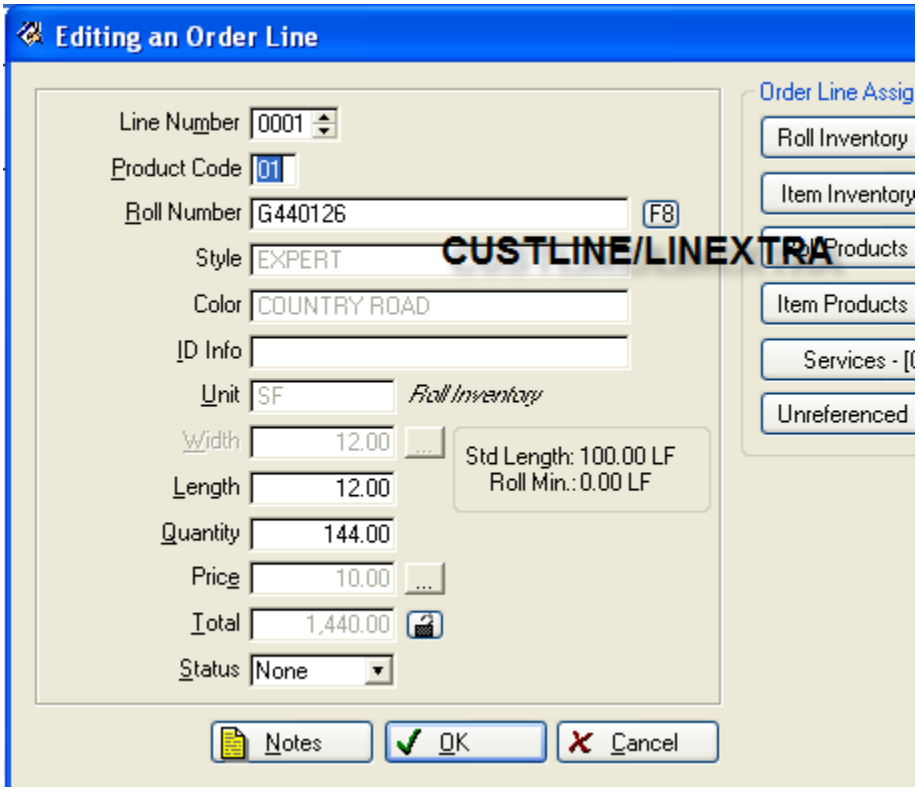
described earlier are also in play here but not shown in the diagram. The Invoice Number (INV_NUM) from the header is the column that links all of the tables together.

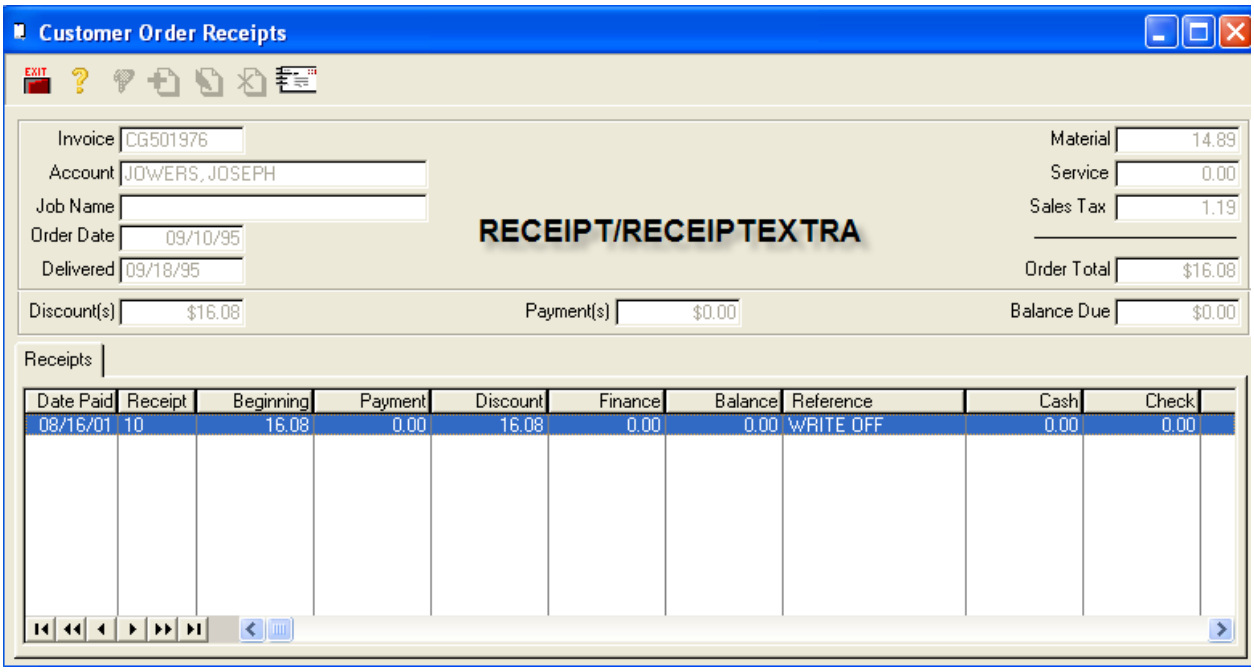


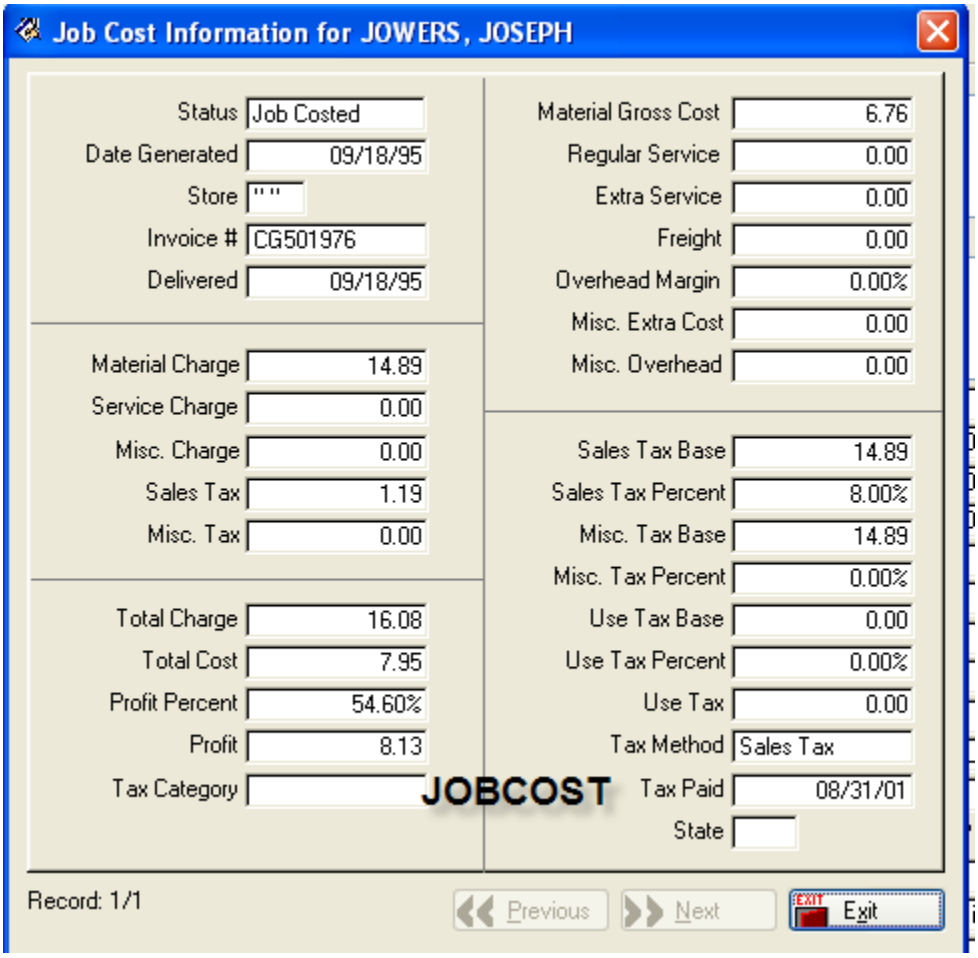
The following screens show the tables that the information comes from.


RFMS Order Entry for SEMINAR CLASS - [Customer Order - CG402254]
File Edit Reports Utilities Window Help Navigator Mail
Original Order
Order # CG402254 F8
Store F8
Order Date 03/24/04
Sold To
BARRETT, MITCHELL
22 HIGHLAND DR.
TUSCALOOSA, AL 35405
County: TUSCALOOSA
HEADER
Sales Rep #1 SUSAN
Sales Rep #2
Time Slot -None-
Tax Status Taxable
PO Number
Status
Line # Style Product Code Roll/Item Number Color/Description Status
Search Line Number 0000
PC Roll/Item # Style / Item Name Color / Description Width Length Quantity Price Total Status
01 G440126 EXPERT COUNTRY ROAD 12.00 12.00 144.00 10.00 1,440.00 None
04 PLATINUM 7# A&H FOREVER 7# PAD 0'0" 0'0" 858.00 0.50 429.00 None
81 *Service* CPT INST STRET 867.00 0.57 495.43 None
07 WUO3HT4 WILMINGTON PL HONEYTONE- 3" 0'0" 0'0" 23.00 5.00 115.00 None
01 C10082755 347 18 12.00 33.00 396.00 0.75 297.44 None
13 2009302 ADVANTAONE T-MOLD 0'0" 0'0" 1.00 20.00 20.00 None
01 ABRACADABRA N12 12.00 21.00 252.00 2.04 514.36 None
13 H1002947 WHITE SANDSTO WIL-ART BASERS 0'0" 0'0" 3.00 49.82 149.46 None
CUSTLINE/LINEXTRA


Hold
Customer #
Date 03/24/04 F8
Tele #1 F8
Tele #2
Ship To
BARRETT, MITCHELL
22 HIGHLAND DR.
TUSCALOOSA, AL 35405
County: TUSCALOOSA
Delivery Date
Invoice Date
CompleteDate F8
Est Install F8
HEADER
Tax Status Taxable
Number
Status ** No Status Specified **
Sub/Lot
Customer Type CHECK
Occupied No
Material 2,965.26
Services 495.43
Misc. Charges 0.00
Sales Tax 266.87
Misc. Tax 0.00
Order Total $3,727.56
Balance Due $3,727.56
Grand Total $3,727.56
Total Status Line # Est Del Store Item Sequence Net Cost Gross Cost GP % Use Tax ID Info
0.00 None 0001 0 3.000 3.000 67.39% ---
9.00 None 0002 1 0.369 0.369 19.78% ---
5.43 None 0003 n/a 0 0.400 0.400 30.00% ---
5.00 None 0004 1 2.900 2.900 36.96% ---
7.44 None 0005 0 0.254 0.254 59.48% ---
0.00 None 0006 2 9.990 9.990 45.70% ---
4.36 None 0007 n/a 0 1.224 1.224 34.79% ---
9.46 None 0008 n/a 0 27.900 27.900 39.13% --- WIL-ART BASEE


Editing an Order Line
Line Number 0001
Product Code 01
Roll Number G440126
F8
Style EXPERT
Color COUNTRY ROAD
ID Info
Unit SF
Roll Inventory
Width 12.00
Length 12.00
Quantity 144.00
Price 10.00
Total 1,440.00
Status None
Std Length: 100.00 LF
Roll Min.: 0.00 LF
Order Line Assig
Roll Inventory
Item Inventory
Item Products
Services - [
Unreferenced
Notes
OK
Cancel
CUSTLINE/LINEXTRA


Customer Order Receipts
Invoice CG501976
Account JOWERS, JOSEPH
Job Name
Order Date 09/10/95
Delivered 09/18/95
Material 14.89
Service 0.00
Sales Tax 1.19
Order Total $16.08
Discount(s) $16.08
Payment(s) $0.00
Balance Due $0.00
RECEIPT/RECEIPTEXTRA
Receipts
Date Paid Receipt Beginning Payment Discount Finance Balance Reference Cash Check
08/16/01 10 16.08 0.00 16.08 0.00 0.00 WRITE OFF 0.00 0.00



HEADER

The header table is the base table for the order entry system.

Columns of interest.

Header.Inv_Num: The invoice number of the order.
Header.Inv_Type: Invoice Type. Blank= Original Invoice,'O' = Overage, 'L'=Claim, 'C'= Credit Memo,'A' = Add on.
Header.Inv_Date: Order date
Header.Del_Date: Date order jobcosted (Accounting date)
Header.DateEntered: Date the order was entered into RFMS
Header.Prom_Date: Estimated Date for work to start.
Customer/JOB Info. This information may be imported in from the customer file or entered on each header. To link back to the customer file use CUST_NAME.
Header.JobNumber: This can be relabeled by the end user to display a different field name.
Header.Mat_Only: Total of Material Lines from CustLine
Header.Labor: Total of service lines from CustLine
Header.MiscCharge: Entered on the OE screen.
Header.Inv_Total: Total of THIS invoice. Does not include Credit Memos (CM) or Addons (AO)

Header.Grand_Inv_Total: Total of all invoices in this chain. Includes CM and AO
Header.Tot_Pay: Amount paid on the order.
Header.Balance: Balance of the order
Header.ControlNum: Inv_Num of invoice that this invoice is linked to
Header.Nexthead/PrevHead: Next/|Previous header seqnum in Invoice chain.
Header.InvPrtDate: Date the invoice was printed

CUSTLINE/LINEEXTRA

Columns of interest.

Custline.Roll_Item_Num/ItemSeqNum: Links the line to an inventory record
Custline.Qty: Qty. If ROLL GOOD then always stored in Square Yards
Custline.Grs_Cost: Cost used for G/L and sales commissions.
Custline.Net_Cost: Cost used for Use Tax calculations
Custline.JobcostDate: Date the line was jobcosted
Custline.CutDate: Date the line was assigned from inventory
LineExtra.TotalLocked: The line total will not change if Qty changes. New unit price is calculated instead.

RECEIPT/RECEIPTEXTRA

Columns of interest.

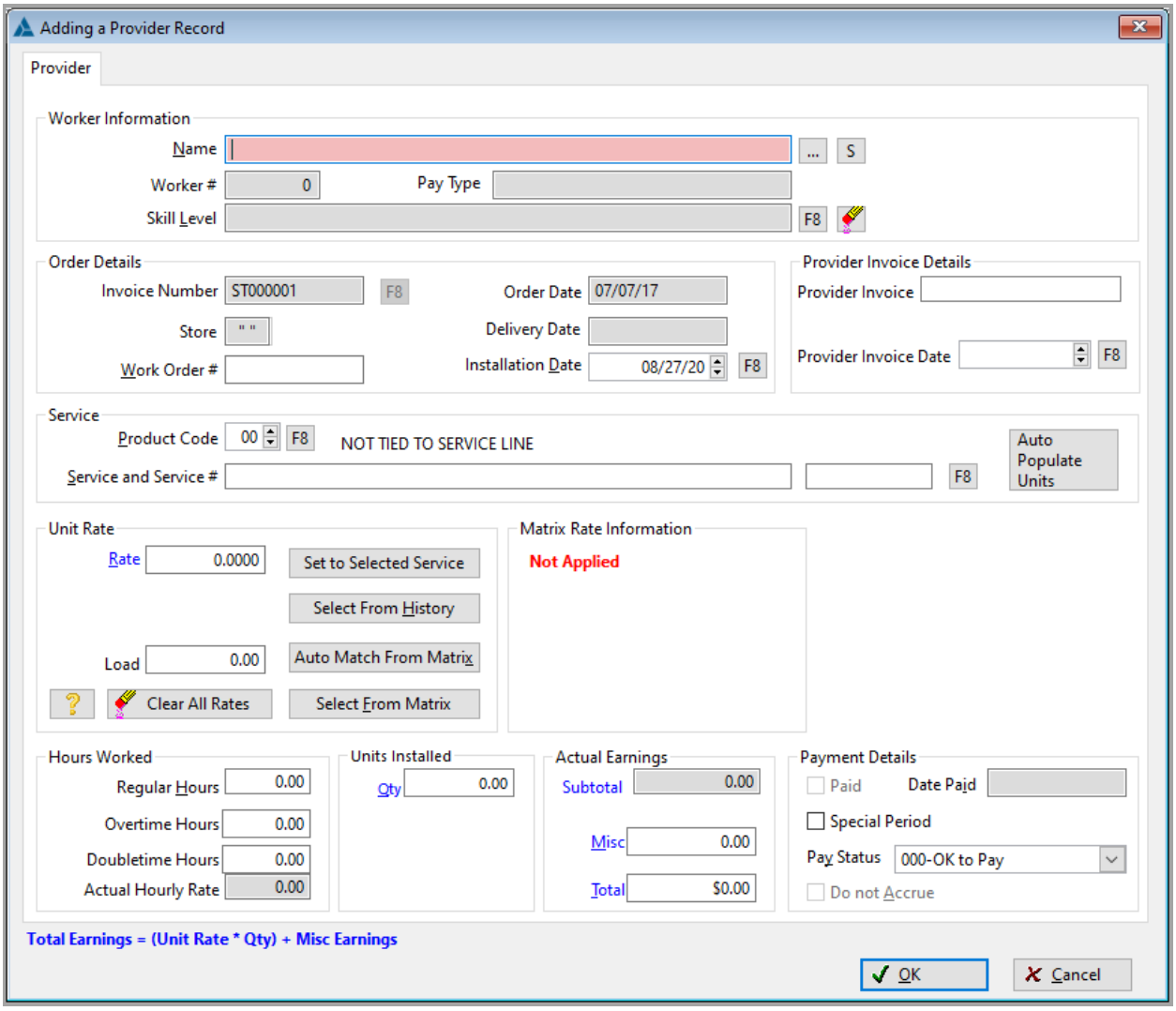
Receipt.Tot_Amt: Total of Check_Amt+Cash_Amt+Credit_Amt

Receipt.Disc_Amt: Amount posted as a customer discount.
Receipt.Recp_Type: 'F'=Finance Charge, Blank = Payment
Receipt.Pay_Date: Date the payment was made
Receipt.Del_Date: Date the payment was jobcosted
Receipt.FN_Charge: Amount of Finance Charge
Receipt.Disc_Act: Account code the discount will be applied to.
Receipt.Recp_Num: The number of the receipt. Used to balance deposits to bank with money taken in.
Receipt.SeqNum: The receipt register that this payment belongs to.
ReceiptExtra.ChkNum: Check number the customer paid with

JOBCOST

Columns of interest.

Installers (Providers)



Columns of interest

Worker Information

Pay_Type: Whether the worker is Salaried, Subcontractor, Hourly or Salaried+commission. This information originates with the Worker record.

Inv_Num: Invoice Number

Order Details

Install_Date: Actual date work happened can be pulled from the schedule pro record or entered manually

WorkOrderNum: Optional field used

Provider Invoice Details

ProviderInvNum

ProviderInvDate

System Options

Table Name SystemOptionsControl

Columns of interest

ConfigStore: Module this option relates to

ParameterType:

Parameter:

Type: Options are User, Assigned or Global

Protected:

Hidden

Category

Applies to

Option: The Option name

Private Settings

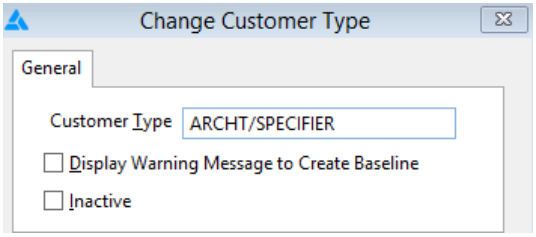
Table Name: UserPrivateOptions

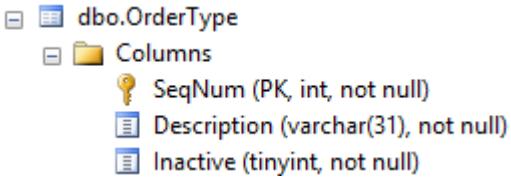
Includes “sticky” settings.

Set up Tables

Customer Type

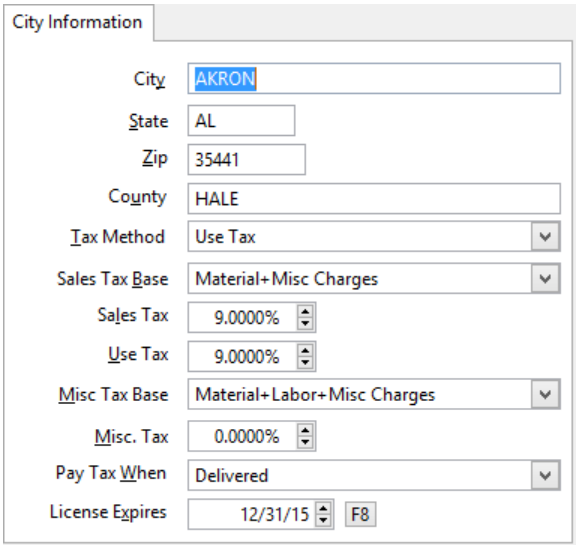
Table = Order Type

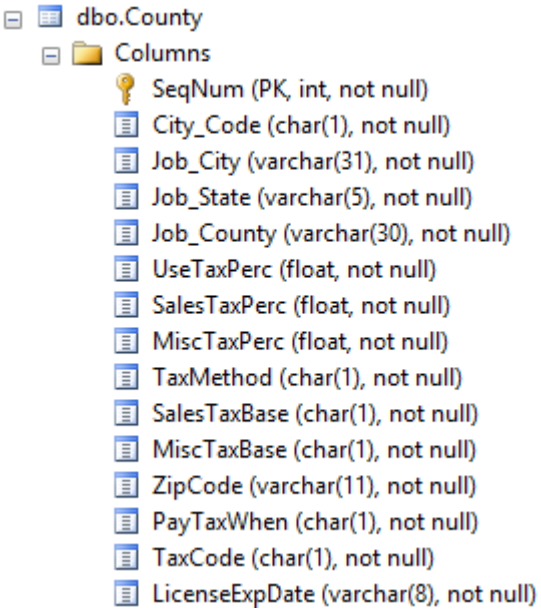




City

Table = County


City Information
City
AKRON
State
AL
Zip
35441
County
HALE
Tax Method
Use Tax
Sales Tax Base
Material+Misc Charges
Sales Tax
9.0000%
Use Tax
9.0000%
Misc Tax Base
Material+Labor+Misc Charges
Misc. Tax
0.0000%
Pay Tax When
Delivered
License Expires
12/31/15
F8


dbo.County
Columns
SeqNum (PK, int, not null)
City_Code (char(1), not null)
Job_City (varchar(31), not null)
Job_State (varchar(5), not null)
Job_County (varchar(30), not null)
UseTaxPerc (float, not null)
SalesTaxPerc (float, not null)
MiscTaxPerc (float, not null)
TaxMethod (char(1), not null)
SalesTaxBase (char(1), not null)
MiscTaxBase (char(1), not null)
ZipCode (varchar(11), not null)
PayTaxWhen (char(1), not null)
TaxCode (char(1), not null)
LicenseExpDate (varchar(8), not null)