Order Entry Tables

---

stocntrc Order Entry Control Table

```
    disc_frght char(1),      - Apply trade discounts to freight

    tax_frght char(1),       - Compute sales tax on freight

    taxable char(1),         - Obsolete, field no longer used

    st_tx_code char(6),      - Obsolete, field no longer used
    co_tx_code char(6),      - Obsolete, field no longer used
    ci_tx_code char(6),      - Obsolete, field no longer used

    warehouse_code char(3),  - Default warehouse code

    pay_method char(6),      - Default payment method
                               This is used to default the pay_method in
                               the quick-add customer screen.
                               It is also used if the pay_method in the
                               customer table is null.

    fob_point char(15),      - Default FOB point

    ship_via char(15),       - Default shipping carrier

    retention_days smallint, - Days until purge of completed orders

    due_days smallint,       - Default due days from invoicing date

    cm_reason char(3),       - Default credit memo reason code

    dm_reason char(3),       - Default debit memo reason code

    ar_acct_no integer,      - Default a/r account number for a/r sales

    cash_acct_no integer,    - Cash account number for cash sales

    visa_acct_no integer,    - Credit card account number for cc sales

    sales_acct_no integer,   - Default sales account

    disc_acct_no integer,    - Trade discount account number

    frght_acct_no integer,   - Freight account number

    inv_acct_no integer,     - Default inventory account number

    cog_acct_no integer,     - Default cost of goods account number

    scrap_acct_no integer,   - Inventory scrappage account number
```

```
use_department char(1),   - Use order dept for asset/liability

oe_doc_no integer,        - Last order document no. (incremented)

oe_inv_doc_no integer,    - Last invoice document no. (incremented)

oe_post_no integer        - Last o/e posting number (incremented)

order_type char(3),       - Default order type

line_type char(3),        - Default line type

terms_code char(6),       - Default terms_code if customer's is null

change_terms char(1),     - Ok to change billing terms?(Y/N/Approval)
                            Field not in use - For Future Use

cod_ok char(1),           - Credit limit overrides ok for COD's
                            Field not in use - For Future Use

one_time_cust char(20),   - cust_code for the 'One-Time' customer
                            Field not in use - For Future Use

ack_kit_exp char(1),      - Expand kits on acknowledgements?

pic_kit_exp char(1),      - Expand kits on picking tickets?

mfs_kit_exp char(1),      - Expand kits on shipping manifests?

inv_kit_exp char(1),      - Expand kits on invoices?

ack_note char(1),         - Show order notes on the acknowledement?

pic_note char(1),         - Show order notes on the picking ticket?

shp_note char(1),         - Show order notes on shipping manifest?

inv_note  char(1)         - Show order notes on invoice ?

mtaxg_code char(6)        - Default multilevel tax group code

use_batch_inv char(1)     - Use Batching for Invoices?

use_approv_post char(1)   - Use Approval Code to Batch?

approval_code char(8)     - Approval Code to Post

ship_terms char(15)       -

req_profit_pct decimal(6)- price margin warning

inv_stage char(3)         - Invoice Stage 'ORD' 'PIC' 'SHP'
```

```
    truck_bol char(20)       - Truck Bill of Lading Number

    init_ord_stage char(3),  - Initial Order Stage

    rel_hld_auth char(10)    - Release hold authorization

    restock_acct_no integer  - RMA Restock account number

    profit_approval char(1), - Profit Approval Y/N

    profit_override char(10),- Profit Override password

    restock_percent decimal(6,2), - Restock Fee

    ordr_printpt char(1),         - Streamline Order Entry SO Print PT?

    ordr_createinv char(1),       - Streamline Order Entry SO Create Inv?

    pick_createinv char(1),       - Streamline Order Entry PT Create Inv?

    ship_createinv char(1),       - Streamline Order Entry Ship Create
Inv?

    invce_print char(1),          - Streamline Order Entry Inv print?

    invce_post char(1),           - Streamline Order Entry Inv post?

    bko_printpt char(1),          - Streamline Order Entry BKO print PT?

    ship_type char(6)             - Ship Type

    pick_createpl char(1),        - Real Time Processing Create Pick

    ship_createpl char(1),        - Real Time Processing Create Ship

    auto_invce_print char(1),     - Real Time Processing Create Invoice

    auto_invce_post char(1),      - Real Time Processing Auto Invoice

    create_bko_po char(1),        - Real Time Processing Auto BKO PO

    auto_invce_post_d char(1),    - Real Time Processing Auto Invoice

    invce_post_d char(1)          - Real Time Processing Auto Invoice
Direct


--------------------------------------------------------------------------
stootypr          Order types

Type control:
```

order_type char(3),
Unique order type. This is the key to this table.

description char(30),
Short description of this order type. (a longer description of the specifics of this order type should be stored as notes keyed to this order_type)

like_type char(3),
The OE programs know of several hardcoded order types. Every order type must be "like" an order type that OE knows about. The known order types are: (.iX order_types)

REG: Regular order		
DIR: Direct ship	aka (DPS: Drop ship)	
SWC: Ship when complete	Field for future use.	
T&H: Tag & hold	Field for future use.	
WIL: Will Call	Field for future use.	
CRM: Credit Memo		
DBM: Debit Memo		
BLO: Blanket Order.	Field for future use.	
RCR: Recurring order.	Field for future use.	
QUO: Quotation.		
FUT: Future order.	Field for future use.	
MOR: Master order release.	Field for future use.	

Non-modifyable order type characteristics:
many order types share characteristics, but if they are changed, it would make the program react in unpredictable ways. Those characteristics are stored here, but they don't show on any screen, so they can't be modified by the user.

master_order char(1),
is this a master order? master orders are by definition reference type orders. They are designed, however, to be released from, with a new order being made. the new order's sales order number is the original sales order number with an extension as to the number of releases from the master. example of a master order release sales order number: 1233-04 (being the 4th release from master order #1233)

reference_order char(1),
is this a reference order? a reference order is one that is not allowed to advance passed the 'NEW' stage. because lines can't advance, pickers won't be printed, the items won't ship, and therefore nothing will be billed. users aren't able to mark anything as being shipped until they change the order type to one that isn't a reference order.

Modifyable order type characteristics:
these order type characteristics are shown on the "define order

types" screen, and can be changed without compromising the integrity of the line type.

```
  print_ack char(1),
      are acknowledgements printed for this order type?

  print_pic char(1),
      are picking tickets printed for this order type?

  print_mfs char(1),
      are shipping manifests printed for this order type?

  pay_method_req char(1),
      Is the payment method required at the time of order taking?

  po_no_req
      are customer purchase order numbers required at the time of
      order taking?

  fob_point_req
      is the fob point information required at the time of order
      taking?

  shp_via_req
      is the ship via information required at the time of order
      taking?

  frt_doc_no_req
      is the freight document number required when entering
      shipment information?

      this field currently is not defined in the stoordre table,
      this number does exist in stoinvce.

  fact_ord_begin       char(6)   - Field not in use
  fact_ord_end         char(6)   - Field not in use
  fact_ord_next        char(6)   - Field not in use
  bill_only            char(1)   - Field not in use
  inv_pre_apprv        char(1)   - Field not in use
  tmp_order            char(1)   - Field not in use
  intl_order           char(1)   - Field not in use
  resale_cust          char(1)   - Field not in use

  base_sales_order     char(1),
      Attach to Base Sales Order No?

  multiple_rma_type    char(1)
      Allow multiple Debit/Credit/RMA Types

  fixed_price char(1),
      Fixed Price

  salesperson_req char(1),
```

Salesperson Required

credit_check char(1)
Credit Check

---

stoltypr Order Line Types

Type control:

line_type char(3),
unique line type. this is the key to this table.

description char(30),
short description of this line type (a longer description of the specifics of this line type should be stored as notes keyed to this line_type).

like_type char(3),
the OE programs know of several hardcoded line types. every line type must be "like" a line type that OE knows about. The known line types are:

STK: Stock
NON: Non-stock
STN: Stock - Handle as a nonstock
SUR: Surplus
FOU: Found item

Non-modifyable line type characteristics:
line types share characteristics, but if they are changed, it would make the program react in unpredictable ways. Those characteristics are stored here, but they don't show on any screen, so they can't be modified by the user.

stock_item char(1),
Is this a stocking item? If this is set to 'Y', then the item number is validated from the inventory tables. If there is stock in inventory for this item, it will commit that stock at order time. The inventory control tables are updated for lines that the stock_item = "Y".

Modifyable line type characteristics:
these line type characteristics are shown on the "define order types" screen, and can be changed without compromising the integrity of the line type.

desc_update char(1),
Should the user be allowed to override the description brought in from the inventory tables (Y/N)? This is only applicable if the item description is defaulted from the inventory tables.

price_update char(1)
Should the user be allowed to override the unit price brought in from the inventory tables? (Y/N/A) [Y]es, [N]o, [A]pproval required. This is only applicable if the item price is defaulted from the inventory tables.

rnd_dollar char(1) - Field not in use.

stage_to_bko char(1) - Field not in use.

---

stoordre Order Entry header Table

Order numbers:

doc_no integer not null,
doc number, assigned at entry time. if order is canceled, it is not deleted - just marked as canceled. once shipped, cannot be canceled.

orig_doc_no integer,
for master order releases, this represents the document number of the master order. for credit/debit memos, this contains the INVOICE document number that is being credited/debited. Otherwise it is null.

order_no char(20),
sales order number, assigned at entry time. can be entered by operator. defaults to doc_no. no dup checking.

inv_doc_no integer,
For credit/debit memos, this is the invoice document number that is being credited or debited. This is NOT the inv_doc_no for the current document. This is null for orders.

inv_no char(10),
For credit/debit memos, this is the invoice number that is being credited or debited. This is null for orders.

po_no char(20),
purchase order number. used for referencing the customers po number on the order.

pic_ticket_no smallint,
This column contains the total number of picking tickets printed. re-prints don't assign new pic ticket numbers. picking tickets are numbered with the sales order and the pick_ticket_no appendage. example: 3386-02. when a new picking ticket is printed, it uses this column to determine the next picking ticket number. it also uses this column to show on the screen the total number of picking tickets printed for this order. pic_ticket_no is initialized to null. if picking tickets aren't to be printed (based on print_pic),

then it is set to 0.

next_kit_group smallint,
This column contains the highest kit line number for the order. It is used to determine the next kit_group for the order lines (when entering a new kit). It is an internal counter and doesn't show on any screen or report.

ack_printed char(1),
Has an order acknowledgement been printed for this order (Y/N)? A NULL value means that there is no acknowledgement needed for this order type. A "N" value means that an acknowledgement needs to be printed, but hasn't been printed yet for this order. A "Y" value means that the acknowledgement has already been printed for this order.

Order Control:

order_type char(3),
These are entered at order entry time. They are validated from the stootypr table. The order types define process characteristics that affect the order. Default order types includes:

REG:	Regular order	
DIR:	Direct ship	aka (DPS: Drop ship)
SWC:	Ship when complete	For Future Use
T&H:	Tag & hold	For Future Use
WIL:	Will Call	For Future Use
EVL:	Evaluation	For Future Use
CRM:	Credit Memo	
DBM:	Debit Memo	
BLO:	Blanket Order	For Future Use
RCR:	Recurring order	For Future Use
QUO:	Quotation	
FUT:	Future order	For Future Use
MOR:	Master order release	For Future Use

There is 1 Hardcoded order type for processing purposes:
CAN: This is the mechanism for cancelling an order. The order type is changed back to it's original type, but all line stages & the order stage & status is set to 'CAN'. Orders can't be canceled if the highest line stage is at or above the stage of SHP (shipped). Allocated inventory is unallocated.

Screen documentation:
This screen field is validated from the stootypr table. If the field is left blank, then it defaults to the value stored in the O/E defaults setup screen. If you leave this field blank by pressing [TAB] to go to the detail lines, the order type will default to a quotation ("QUO").

like_type char(3),
an order type can be defined by the user, yet there are many controls that are needed based on the order type. to accommodate this, when the user creates an order type, it must act "like" one of the types known to the system. the order_type may be "DPS" (because the industry knows a direct shipment as a drop shipment), but the DPS order type is "like" the hardcoded "DIR" (direct ship).

order_status char(3) not null,
hardcoded status codes. determined by the computer. can be used for order selection. statuses include:

REF - Reference order.
Reference orders cannot have any order line advance passed the "NEW" stage.

ACT - Active Order

CRH - Order on Credit Hold. For Future Use - field not currently used.
Credit hold occurs for non-reference type orders if this order amount + other outstanding orders + current a/r balance exceeds the customers credit limit. the only way to remove a credit hold is to increase the customer's credit limit or reduce their outstanding orders (including this one), or to reduce their a/r balance (by having them send you a check). orders on credit hold will not allow picking or shipping documents printed for the order. credit hold overrides staging hold. when credit hold is removed, it can either go to ACT or STH if staging hold is appropriate for the order.

STH - Order on Staging Hold. For Future Use - field not currently used.
this can only occur if it's a 'tag and hold' or a 'ship when complete' type order. staging hold means to hold the inventory in it's staging area without shipping. picking lists are printed differently for orders on staging hold. the ship-to address is omitted, and replaced with: "Do not ship. Place into staging area ABC-88" after the order is taken off staging hold, a picking list is printed showing the bin location as the staging area, and this picking list shows the ship-to address. (when staging hold is removed, all shipment lines at "PIC" status change back to "ORD" so they can be picked again - this time FROM the staging area). for "ship when complete" orders, staging hold is removed when all items have been picked and placed into the staging_area (unless there exists a to_ship_date, and that date hasn't been met). for "tag and hold" type

orders, staging hold isn't removed UNTIL there exists
a to_ship_date, and that date has been met
(regardless if all items have been picked).

HLD - Manual Hold. For Future Use - field not currently used.
Manually set and removed. Orders on manual hold are
treated like those on credit hold(see above). Manual
hold is set and removed via the options menu.
When manual hold is set, the user is asked to enter hold
notes.

PST - Order Posted

CAN - Order Canceled

hi_stage char(3) not null,
lo_stage char(3) not null,
actually, orders don't have stages, order shipment lines have
stages. these columns only show the highest and lowest of
all of the line stages in the order. The lo_stage column is
shown on the screens as the order stage. stages are
hardcoded and not operator entered. available hi/lo order
stages:

NEW: New - Waiting (for some reason) to be put on order
BKO: On Backorder - Waiting to arrive
ORD: Ordered (and committed) - Waiting to pick
PIC: Has been picked - Waiting to ship
SHP: Has been shipped - Waiting for invoice approval
INV: OK to invoice. Ready to post after invoice is printed.
PST: Has been posted - OK to archive (when age is met)
CAN: Has been canceled

bo_allowed char(1),
are backorders allowed for this order Y/N? this column is
null until the first line that needs a backorder. at that
time, the user is asked if backorders are allowed, and that
sets this column. subsequent lines will automatically
backorder if necessary, or not allow backordering based on
this column.

Master order information:
This information is kept for master orders only.

recur_unit char(1), For Future Use - field not currently used.
This specifies the unit for automatically recurring orders
(contract billing). Data in this column can include:
D - Days
W - Weeks
M - Months

recur_every smallint,For Future Use - field not currently used.
This is used in conjunction with recur_unit (above). It

contains the number of units that need to pass before automatically recurring the order. Example: If recur_unit = "W" (weeks) and recur_every = 3, then the order would recur every 3 weeks.

recur_times smallint, For Future Use - field not currently used.
This specifies the total number of times this master order should recur. If the num_releases (below) is less than recur_times, then a new "next_recur" date will be calculated when the order recurs. If the order is able to recur any number of times, then this column will be null.

recur_through date, For Future Use - field not currently used.
This is the contract ending date. It specifies the last date that automatic recurring can take place. If the contract specifies a NUMBER of recurrence vs. a contract ending date, then the recur_times (above) will be filled, and the recur_through column will be null. If recur_times (above) and recur_through are BOTH specified, then both tests will be applied to determine if the order is capable of recurring again. If either test fails, the order will not automatically recur. (NOTE: the contract starting date is stored in the order_date column)

prev_recur date, For Future Use - field not currently used.
The date this master order last recurred.

next_recur date, For Future Use - field not currently used.
The next recur date. This column is automatically filled based on the "recur_unit" and "recur_every" columns. It will be null if the master order is not marked for recurring (not a recurring type order or the contract period has ended based on the recur_times and recur_through columns.

num_releases smallint,
The number of releases from this master order (master order types only). This number is used to generate the release sales order number from the master sales order number with this as an appendage (ex: 1234-04). It is also used in conjunction with recur_times (above) to specify the number of times the order has recurred.

release_type char(3),
This specifies the type of order that is generated from this master order. If this column is null, the order will be released as a master order release type: "MOR". If the release_type is not null, then it will be placed into the order_type column of the new order. In addition, if that new order type is NOT a reference type order, then the system will automatically advance the order line stages to ORD. If there is not enough stock in inventory to move the order line to ORD, it will automatically backorder the amount needed and advance the line stage to BKO.

Date stamps:

order_date date,
This is the date this order is accepted. It defaults to entry date. It is used for informational purposes only. It is not used for any A/R or G/L postings. For contract type master orders, this is the contract starting date.

to_ship_date date,
This is the date that the shipment is to be made for this order. It is for "future", "tag & hold, and "ship when complete" order types. All other (non-reference) type orders fill this column with the order date. Picking lists won't print ship-to addresses (only staging areas) and shipping manifests won't print at all until this date occurs.

alloc_date date,
for future orders only. this is the date that the future order becomes a "ship when complete" type order with a to_ship_date. this is the date that the order is required to be allocated. this date defaults to the longest lead time for all order lines + 14 days (2 week buffer). it is designed to allow for enough time to order any line (with a 2 week buffer) in the case that the item needs to be backordered and can't be allocated right away.

ship_date date,
this is the date of the last shipment. either the shipping manifest sets this, or it is set at billing time when the order line is marked as shipped. it is used for printing on the invoice.

complete_date date,
set at posting time when the entire order has been completely posted. Also set when the order is canceled. used with "retention_days" from the order entry control table to determine when to purge the order from the system. old orders are purged automatically at posting time.

Line default information:

warehouse_code char(3),
this defaults to the warehouse code in the order entry control table. it can be overridden by the operator. the warehouse_code is used as the default warehouse_code on the order lines. it can be overridden on the order lines.

department char(3),
default g/l department to use. defaulted to the department in the customer table. If that is null, or no customer exists, then this is defaulted to "000". default department

code to used on the order lines for revenue and cost of goods department. also used to default the department code in the header for trade discount, and freight amounts. If the control table's "use_department" flag is set to 'Y', then this code is also used to default the liabilities(taxes) and assets(cash/ar/card) departments. if the "use_department" flag is set to 'N', then the liabilities(taxes) and assets(cash/ar/card) departments are defaulted to "000".

sls_psn_code char(6),
sales person code. defaulted to the salesperson code in the customer record. if that is null, then it is defaulted to the login name (if it can be validated in stxinfor). if the salesperson code is changed on any line of the order, the changed salesperson code is recorded here so subsequent added order lines will default to the new salesperson code.

Customer sell-to/ship-to/bill-to information:

cust_code char(20),
This is the sell-to customer code. Orders can have different sell-to and bill-to customers. Sales analysis information is posted to the sell-to customer. Billing is posted to the bill-to customer. Normally, they are the same. Exceptions include credit card sales and 3rd party (leasing company) sales. If the cust_code refers to a "bridge" type customer, then there may be several different sell-to codes for this order. They will all belong to the same bridge customer. If it is not a bridge type customer, then there can only be one sell-to customer for the order.

ship_to_code char(6),
Shipping address code for the customer. This is validated from the customer/ship-to tables. If the code contains "SHIPTO" then the system uses the customer's billing address as the shipping address. The ship-to code is always attached to the sell-to customer, not the bill-to customer.

bill_to_code char(20),
This is the code for the customer that is going to be billed for the order. Usually it is the same as the cust_code. If the customer buys the item using a credit card, then the bill_to_code will contain the customer code for the credit card company. This allows for credit card reconciliation and discount application via the cash receipts module. If the payment terms are to '3rd Party' (as in a lease), then the 3rd party's customer code is placed in this column.

bus_name char(30), See Note Below
contact char(20), See Note Below
address1 char(30), See Note Below
address2 char(30), See Note Below
city char(20), See Note Below

```
    state char(2),       See Note Below
    zip char(10),        See Note Below
    country char(2),     See Note Below
```

These fields are not currently used, they are for future use. These name and address fields are filled only if the customer code matches the one_time_cust code in the OE control table. It provides a mechanism for allowing one time customers to buy without setting them up in A/R. The credit_limit of the one_time customer in the customer table should be set to 0 so no A/R type orders can be taken for this customer. If one-time customers aren't allowed, then don't define a one-time customer code in the OE setup screen. One-time customer orders cannot have multiple sell-to/ship-to/bill-to codes, but they can have different sell-to/bill-to customer codes. The ship-to address for one time customers will be the same as the billing address, and the ship_to_code will be filled with "SHIPTO".

Order terms:

These terms are first entered into the stoordre table, then copied to each invoice when order lines are marked as ok for billing. The real terms approved by the billing department are stored in stoinvce.

terms_code char(6),

A/R terms code. Retrieved from the bill-to customer record. If the OE setup file says it's ok to override this, then the order entry person may change the terms_code. They may be required to provide an override code. The terms_code may be set to "COD" if the customer's credit limit is exceeded and the OE setup file says it's ok to process COD orders exceeding the customer's credit limit.

terms_approval char(6),

Approval code for terms_code override. If the OE setup file requires an approval code for terms override, the entry clerk must type in an approval code. This approval code isn't checked against anything, but it will show up on an override exception report.

pay_method char(6)

This code is defaulted from the customer table. It is defaulted from the stocntrc table and validated from the stxinfor table.
CASH/VISA/AMEX/MC/ONACCT/3RDPTY are pay_method examples.

payment char(1),

This code determines which of the 3 different types of payment method used. It is looked up from the stxinfor table based on the key entered in pay_method (above).

A - accounts receivable

C - cash
V - credit card
3 - 3rd party billing, Field for Future Use.

card_no char(20)
Used to store the credit card number if paying by card. it is defaulted from the customer table, but can be overridden. this data is only valid for credit card type payments

exp_date char(5)
Expiration date for credit card payments.

card_holder char(20),
Name on the credit card.

check_no char(8)
If paying via cash, this would be the check number used for payment. If paying via credit card, this column contains the credit card companies' authorization code for this purchase.

trd_ds_code char(6),
Trade discount code. This is defaulted from the customer/shipto table. Trade discounts don't affect product pricing. the trade discount is taken from a total of all discountable lines invoiced. (.iX discounts)

trd_ds_type char(1),
Trade discount type. This is null if trd_ds_code is null. Otherwise, it is "D" if the discount type is "discount" or "M" if it is "markup". ("MARKUP" and "DISCNT" are possible values in stxinfor.src_char_desc where src_type = "I" and src_key = trd_ds_code.)

When the value is "D", trd_ds_type affects pricing two ways: if trd_ds_rate is not zero, then a trade discount is computed from a total of all discountable lines invoiced. Whether zero or not, trd_ds_code will be used as part of the key to retrieve the quantity discount information for each line item.

When the value is "M", prices for all stock items are computed from the standard cost (stilocar.purch_unit_cost) using the trd_ds_rate as a markup rate. The pricing table is not used in this case. (.iX discounts)

trd_ds_rate decimal(6),
When trd_ds_type = "D", this is a rate to apply to the sum of the discountable order lines to determine the amount of trade discount to apply to the order. When trd_ds_type = "M", this is a markup rate used to compute the price of all the stock line items. 20% would be stored as .2 (.iX discounts)

multi_shipto char(1), This field not in use - For Future Use
Are there multiple ship-to's for this order? If this column

contains a 'Y', then a ship_to_code was entered on an order line that isn't the same as the main ship_to code for the order. If there are multiple ship-tos for the order, the order tax calculation changes slightly. (.iX taxes)

tax_rate decimal(6), Obsolete, Field no longer used.
This is the sum of the state, county, and city rates for the default shipto address. It is used to calculate the tax_amount if there is only one ship-to address for this order. 8.2% would be stored as .082

mtaxg_code char(6)
this is the default tax code to use for each order line that is subject to tax. the code is defaulted from the strshipr/strcustr/stocntrc.

staging_area char(6), Field not in use - For Future Use
This is here for "tag and hold" and "ship when complete" type orders only. it is the designated "staging" area where the items are picked and placed for future shipping. The 'ship-to' address on the picking document shows the staging area instead of the customers ship-to address (so orders aren't mistakenly shipped). For "ship when complete" type orders, when all lines of the order have been picked (and staged), another picking ticket is printed showing the picker to pick from the "bin location" of this staging_area. For "tag and hold" type orders, this final picking ticket isn't printed until the to_ship_date is met (regardless of if all items have been picked). Order types that stage cannnot ship from multiple warehouses.

fob_point char(15),
Free On Board point. Printed on the order acknowledgement, picking and shipping documents, and invoice. The FOB point is where the title to the goods is transferred. The customer is responsible for freight charges from the FOB point to the shipment destination. (.iX shipping)

ship_via char(15),
Default shipment carrier. This is a required field of entry for non reference type orders. Since an order can have many shipments (and many shipping carriers), the REAL shipment carrier is stored with the invoice totals (in the stoinvce table).

ship_weight decimal(14),
This column represents the total weight for the order. It is calculated by adding the individual weights of the order lines. It is currently not being used for anything except informational purposes. In the future, it will be used as the basis for automatically calculating the shipping fees.

freight_doc_no char(15)

the freight document number.

Order total amounts: these columns contain the order total amounts in each category. If the order has not been fully processed, then these amounts contain projected totals. If the order's lo_stage = "PST", then these are the orders actual totals for these categories. If you need these totals for any one invoice, the data is stored in stoshipd in the negative line_no's (see stoshipd).

item_amount decimal(14),
Total of the invoiceable order lines.

discountable decimal(12),
Total of the amounts that are discountable for this invoice. items in lines may or may not be discountable (this is set in the item row in the inventory control module). also, freight may or may not be discountable. this is set in the order entry control table.

trd_ds_amount decimal(14),
amount of the trade discount for this invoice. computed by taking the discountable amount multiplied by the trd_ds_pct. this amount is DEDUCTED from the order total.

taxable decimal(12),
Note: this column is no longer used, the tax group calculation will handle the amounts to tax. (.iX taxes)

total of invoice amounts that are taxable. items in lines may or may not be taxable (this is set in the item row in the inventory control module). also, freight may or may not be taxable. this is set in the order entry control table.

tax_amount decimal(14),
This is the estimated amount of taxes that will be charged for the order. It is the exact amount if there is only one invoice cut for the order. If there are many invoices cut from this order (which may be unknown at the time of order taking), then this estimated tax amount may differ slightly from the sum of the individual invoices due to rounding. (.iX taxes)

frght_amount decimal(14),
amount of freight to charge for the invoice. entered by the operator.

total_amount decimal(14),
invoice total amount. this consists of this sum:

```
    item_amount
  - trd_ds_amount
  + tax_amount
  + frght_amount
```

= total_amount

System maintained columns: these columns cannot be entered or changed by the operator. they are all printed on the edit and posting lists.

create_date date,
the date the order was first entered.

create_time char(8),
the time the order was first entered.

create_id char(8),
the user id of the person who first entered the order.

l_mod_date date,
the date the order was most recently modified or billed.

l_mod_time char(8),
the time the order was most recently modified or billed.

l_mod_id char(8)
the user id of the person who most recently modified the order.

system_order char(1) Field not in use
spr_no char(7) Field not in use
cust_ord_date date Field not in use
cust_po_date date Field not in use
fact_ack_date date Field not in use
fact_rec_date date Field not in use
moto_rec_date date Field not in use
sent_to_wwop char(1) Field not in use
intl_order char(1) Field not in use
intl_lic_no char(30) Field not in use

currency_code char(3)
multi-currency code

curr_rate_type char(6)
multi-currency rate type

currency_rate decimal(16)
multi-currency rate

edi_sent char(1) Field not in use

blo_exp_date date Field not in use

dpas_rating char(2) Field not in use
Used by the DPAS .ext it is the DPAS rating for the order

```
resale_cust          char(6)  Field not in use
     Used by the resale .ext, it is the
     strcustr.cust_code of the customer who is the
     end user of this order

resale_po            char(30)  Field not in use
     This is the resale customer's purchase order number

resale_price         decimal(18) Field not in use
     This is the price being charge the resale

actual_frght_amt decimal(12)

orig_frght_amt decimal(12)

ship_terms char(15)

residential_cust char(1)

email char(50)

ups_account char(10)

mtax_freight char(6)

auth_amt decimal(10,2),
    Credit Card authorization amount

auth_code char(8),
    Credit Card authorization code

auth_date date,
    Credit Card authorization date

decline_code char(8),
    Creidit Card decline code

decline_message char(60),
    Creidit Card decline message

ship_complete char(1),
    [Y/N]?

contract_no char(20) ,
    Contract Number

multiple_orders char(1)
    [Y/N]
    If user enter 'Y' then oe.4gm/i_order will create
    multiple sales orders and the grand total will
    divided equally by the number of orders user is
    creating.
```

```
deposit decimal(10,2),
   Deposit

docs_sent date,

destination char(30),
   Destination

consignee_name char(20),
   Consignee Name

consignee_addr1 char(30),
   Consignee Address

consignee_addr2 char(30),
   Consignee Address

consignee_city char(20),
   Consignee City

consignee_state char(2),
   Consignee State

consignee_zip char(10),
   Consignee Zip

consignee_country char(20),
   Consignee Country

notify_name char(20),
   Consignee Name

notify_info char(240),
   Notify Information

truck_bol char(20)
   Bill of Lading

route_code char(10)
   Route Code

resale_no char(15),
   Resale Number

resale_expiry date);
   Resale Expiry Date

rma_reason char(6),
   RMA Reason

base_doc_no integer,
   Base Doc Number
```

```
 rma_doc_no integer,
    RMA Doc Number

 order_description char(80),
    RMA Description

 ready_to_invoice char(1),
    Ready to Invoie (Y/N)

 required_date date,
    Required Date

 default_rma_type char(3)
    Default RMA Reason

 restock_fee decimal(8,2)
    Restocking Fee

 ship_to_name char(30),
    Ship To Name

 one_time_cust char(1),
    One Time Customer?

 ship_type char(6),
    Ship Type

 restock_percent decimal(6,2),
    Restock Percent

 restock_amount decimal(8,2),
    Restock Amount

 rma_status_code char(20),
    RMA Stuatus Code

 handling_fee decimal(8,4),
    Handling Fee

 split_terms_code char(6),
    Split Terms Code

 phone char(12),
    Customer phone

split_terms_code char(6),
    Split Terms Code

contact_name char(20),
    Contact Name

contact_phone char(20),
```

```
        Contact Phone

    fixed_price char(1),
        Check if Fixed Price

    rlse_no integer,
        Release Number

    hold_code char(6),
        Hold Code

    credit_approved char(1),
        Credit Approved

    approved_by char(8),
        Approved By

    date_approved date,
        Date Approved

constraint check (ship_complete IN ("Y" ,"N" )) constraint c1oordre

------------------------------------------------------------------------
stoordrd           Order Entry Detail Table

Order header & shipment line join criteria:

    doc_no integer not null,
        Order document number.  Ties the lines with the order
        header.

    line_no smallint,
        This is a sequence number starting at 1 for the order. It is
        used with doc_no to uniquely identify the line, and to
        provide line ordering.  Line numbers less than 0 store
        invoice totals (.iX negative_line_numbers)

    kit_group smallint,
        This is a number that is used to group together all order
        lines that are a part of an exploded kit.  It is an internal
        grouping number, and not displayed or reported anywhere.  The
        order header contains a counter of the number of kits on the
        order in 'last_kit_group'.  This provides for an easy
        mechanism of grouping the kit lines together for kit
        compression on documents.  This column will be null if the
        line is not part of a kit.

    kit_line_no smallint,
        This column, when used with alias_code forms a unique join to
        the kit line that this order line makes reference to.  It is
        used only for order lines that have been made up from kit
        lines.
```

last_ship_line smallint,
This column contains the total number of shipment lines attached to the order line. The system uses this column to determine the next shipment line number. As shipment lines are added to the order line, this number must be incremented.

price_lock char(1),
If the user overrides the price that was automatically retrieved by the system, the price becomes locked. This means that it is NOT automatically looked up again when prices are recalulated.

Line control:

line_type char(3),
These are entered at line entry time. They are validated from the stoltypr table. The line types define process characteristics that affect the line. line_type cannot be changed (except to 'CAN') if the stage is greater than NEW. Default line types includes:

STK: Stock
NON: Non-stock
STN: Stock - Handle as a nonstock
SUR: Surplus - No history posting
FOU: Found item
There are 2 Hardcoded line types for processing purposes:
KIT: Kit - this converts the line (and subsequent lines) into the breakdown of the kit. The line types of the converted lines are set to their type in the kit definition.
CAN: This is the mechanism for cancelling a line. The line type is changed back to it's original type, but the line_stage is set to 'CAN'. Lines can't be canceled if they are on or above the stage of SHP (shipped). Allocated inventory is unallocated.

like_type char(3),
a line type can be defined by the user, yet there are many controls that are needed based on the line type. to accommodate this, when the user creates a new line type, it must act "like" one of the types known to the system.

hi_stage char(3),
lo_stage char(3),
actually, lines don't have stages, order shipment lines have stages. these columns only show the highest and lowest of all of the line shipment stages for this line. The lo_stage column is shown on the screens as the line stage. if there are no order shipment lines, the lo_stage (and hi_stage) will be "NEW". stages are hardcoded and not operator entered. available hi/lo line stages:

NEW: New - Waiting (for some reason) to be put on order
BKO: On Backorder - Waiting to arrive
ORD: Ordered (and allocated) - Waiting to pick
PIC: Has been picked - Waiting to ship
SHP: Has been shipped - Waiting for invoice approval
INV: OK to invoice. Ready to post after invoice is printed.
PST: Has been posted - OK to archive (when age is met)
CAN: Has been canceled

cm_dm_reason char(3),
used for credit and debit memos only. reason codes are kept in the stxinfor reference table. the text from the reference record is displayed on the cm/dm forms and on the edit list and posting reports. the reason type is used to determine what accounts to update and what to do with inventory.

	sales	cog	invent
1) goods returned and scrapped	decrease (scrappage acct increase)	decrease	no chg
2) goods returned and restocked	decrease	decrease	increase
3) overpriced, not returned	decrease	no chg	no chg
4) underpriced, not returned	increase	no chg	no chg

the default codes for credit and for debit memos in the order entry control table.

our_po_no char(20),
This is our purchase order number that the backorder quantity is purchased on. It is used for information as well as for knowing that an actual po has been created from the bko_qty. At first, the column will contain "RQ# ABCD" where ABCD is the purchase order request number. When the PO request is turned into a real PO, then this column will contain the real PO number. In the rare case that this order line has several backorders posted to it, this column will contain the backorder reference number of the last backorder processed.

sls_psn_code char(6),
sales person code. defaulted to the salesperson code in the order header. if the salesperson code is changed on any line of the order, the changed salesperson code is recorded in the header so subsequent added order lines will default to the new salesperson code.

Inventory item information:

warehouse_code char(3),
This specifies the default warehouse that this item will be shipped from. The actual warehouse it is shipped from is stored in the shipment record. This warehouse code is here

only to provide a default for the shipment record.

item_code char(20),
Code for inventory item. This must be entered at order time. Keyed to the inventory table except for non-stock items. For stocking items, when this is entered or changed, the item information (descriptions, costs, prices, etc) is re-loaded.

desc1 char(30),
desc2 char(30),
two lines of item description. can be overridden at order entry time. if more lines of item description are required, they can be entered as order/line notes.

alias_code char(20),
This is any alias that the item_code may have been entered as. If the customer is willing to interchange another item for the one he ordered, the original item_code will be stored in the alias_code column, and the interchanged column will be set to 'Y' instructing the sales history for the original item to be posted to vs. The item that was actually sold. If the line type is 'KIT', then the alias_code becomes the kit ID, and the kit_group column is filled with a code that is shared by this and the other lines of the kit. If the item_code was found to be a customer alias, then the entered alias will be transferred to this column, and the real item code will be placed in the item_code column.

vend_code char(20),
When backordering, if the item is a non-stocking item, the system will ask for the vendor of the merchandise. This is not required, but is usually known at the time the order is taken, so it is recorded here. If the purchasing module is installed, the vendor code is passed so the purchasing agent doesn't have to decide on a vendor before creating the purchase order. For stocking items, the vend_code is retrieved from the default vendor in the item location record.

interchanged char(1),
marked 'Y' if this alias_code was the original requested stocking item, and the customer accepted an interchange. if this is marked 'Y', then the sales history for the original requested item (stored in alias_code) will be updated vs. the sales history for the item_code on the order. This column is for internal use. It is not shown on the screen.

serialized char(1),
Marked 'Y' if this stocking item is kept track of via lots or serial numbers in the inventory control module. If the item is marked as serialized, the picking ticket will print a message to have the picking clerk pencil in the serial

numbers of the items picked. When the item is marked in the system as picked, a winow will open for the entry of those serial and/or lot numbers.

td_disc_allowed char(1),
trade discount allowed indicator. it comes from the item location record for stocking items, and is a field of entry for non-stocking items. it is used to determine whether this item is subject to the customer's trade discount.

tax char(1),
indicator as to whether this order line is taxable. this is defaulted from the item location record, but can be overridden by the operator.

Quantities: (all quantities and prices are stored in selling units unless otherwise noted)

ordr_qty decimal(14),
Quantity that has been ordered for this line. For credit/debit memos, this is the quantity credited/debited.

back_qty decimal(14),
Quantity that has been backordered for this line.

commit_qty decimal(14),
The quantity of stock committed for this line. This commit_qty is for display purposes only & is stored in selling units so the user can quickly see how many of the items that they ordered are actually committed. The actual item commitment (in stock keeping units) is stored in the stoshipd table.

Units, Prices & Extensions:

sell_unit char(2),
Selling unit of measure. The inventory control system allows three different units of measure. They are arbitrarily called the stockkeeping unit, selling unit, and purchasing unit. When the item code is entered, the selling unit is retrieved, and this column is filled. Selling and purchasing units are related to the stockkeeping unit by a ratio or factor which is used in price computation. Note that inventory control keeps quantities and amounts in terms of stockkeeping units. (.iX units_of_measure)

unit_factor decimal(6),
This is the factor that converts selling units into stock keeping units (SKU's) and vise versa. It is not displayed on the screen. Example: stock_unit = BX, sell_unit = EA. If 5 items are in a box, 0.2 would be stored as the unit_factor. (.iX units_of_measure)

price decimal(14,4),
Price is computed based on the pricing table mechanism. The operator can override the computed price if authorized to do so. If quantity, item code, warehouse code, or unit of measure is changed, the price will be recomputed. The price stored at the line level is for informational uses only. It represents the latest price used for the item. The actual price used on the invoice, posting, and sales reports is stored in the line/shipment record for the actual shipment. It may be different for each shipment (see the documentation on price in the item/shipment record).

price_code integer,
If this column is not null, then it will contain the unique price_code from the pricing table that was used to determine the price of this item.

tax_amount decimal(14),
NOT currently being used. 6/26/92
This is the sum of the tax_amount columns in stoshipd for this line. The stoordrd.tax_amounts are summed up to determine the total order tax amount if there are more than one ship-to addresses. It contains 0 if the line isn't taxable.

net_amount decimal(14)
Extended total line amount. This column represents the sum of all shipment lines net_amount columns. It is used for order totaling. Because there may be differences in price for different shipments, this column cannot be calculated taking the price column (above) * the order_qty column.

ship_weight decimal(14),
This column represents the total individual weights of the shipment lines (see stoshipd.ship_weight). It is currently not being used for anything except informational purposes. In the future, it will be used as the basis for automatically calculating the shipping fees.

General ledger codes: these are the general ledger accounts and departments for the amounts on this line that will post to the ledger.

inv_acct integer,
Inventory g/l account number. Defaulted from the item table unless null, then defaulted from the o/e control table. Null if nonstock.

inv_dept char(3),
Inventory g/l department.
if the inventory control table indicates that warehouse department should be used for the inventory account, then get it and use it. otherwise, if order entry control table

```
          says use order department for asset/liability accounts,
          use the order department.  otherwise, use department "000".

     sls_acct integer,
          sales g/l account number.  if item is a stocking inventory
          item, then defaulted from the item table.  if item table
          sls_acct is null or if the item is a non-stocking item, then
          defaulted from the o/e control table.

     sls_dept char(3),
          sales g/l department.
          warehouse department should be used for the
          sales account, then get it and use it, if the warehouse
          does not have a department defined, then use the
          order department.

     cog_acct integer,
          cost of goods g/l account number.  defaulted from the item
          table unless null, then defaulted from the o/e control
          table.  null if nonstock.

     cog_dept char(3),
          cost of goods g/l department.
          warehouse department should be used for the
          cog account, then get it and use it, if the warehouse
          does not have a department defined, then use the
          order department.

     intl_lic_no          char(30)
     price_lock           char(1)
     release_qty          decimal(14)

     resale_price         decimal(18)
         This is the price being charge the resale

     mtaxg_code           char(6)

     blanket_doc_no integer,
         Blanket doc number

index1: doc_no, line_no (unique)    index2: item_code
-------------------------------------------------------------------------
stoshipd      Order Line Shipment Detail Table

  For every order line, there may be up to 99 shipments of that
  item.  This table keeps track of each shipment for the order line.
  it is used to record the quantity for the picking ticket, shipping
  manifest, invoice, and the postings.

Shipment join criteria:

     doc_no integer not null,
         Order document number.  Ties the shipment to the order.
```

line_no smallint not null,
Order line number. Ties the shipment to the order line.

ship_no smallint,
This is a sequence number starting at 1. It is used with doc_no and line_no to uniquely identify the shipment, and to provide shipment history ordering.

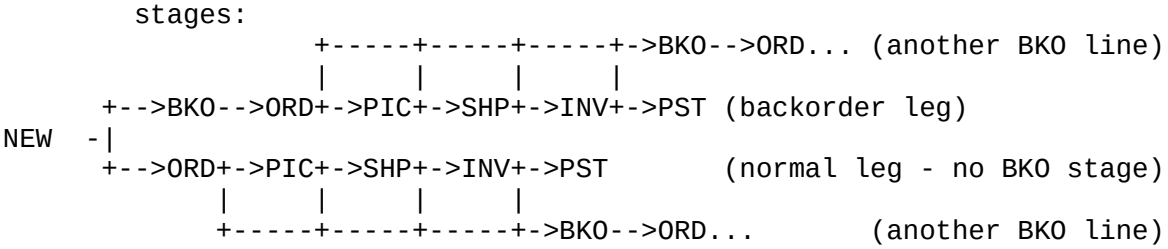
kit_group smallint,
This is a number that is used to group together all shipment lines that are a part of an exploded kit. It is an internal grouping number, and not displayed or reported anywhere. This column will be null if the order line is not part of a kit.

Shipment control:

stage char(3),
these stages are hardcoded and determined by the computer. the stage of a shipment line is where it is in it's life cycle. a shipment record always starts at ORD or BKO and ends up as PST or CAN. Hardcoded shipment line stages:

NEW: New - Waiting (for some reason) to be put on order
BKO: On Backorder - Waiting to arrive
ORD: Ordered (and allocated) - Waiting to pick
PIC: Has been picked - Waiting to ship
SHP: Has been shipped - Waiting for invoice approval
INV: OK to invoice. Ready to post after invoice is printed.
PST: Has been posted - OK to archive (when age is met)
CAN: Has been canceled

Stages only advance forward. If at any time throught the stage cycle, it is determined that a BKO is needed (due to not enough stock to process the qty ordered), a new BKO line is created for the backordered amount. Flow diagram for stages:



ship_qty decimal(14),
This is the quantity for the shipment line. depending on the shipment stage, it is the quantity ordered, backordered, picked, shipped, billed, posted, or cancelled.

sell_unit char(2),
Selling unit of measure. This is directly copied from the

stoordrd table. It is stored in stoshipd for convenience & speed. There can not be different sell_units on an order line, so sell_unit will be the same for all shipd lines on this order line. See stoordre.sell_unit for a further description of this column. (.iX units_of_measure)

ship_weight decimal(14),
This column is either directly entered, or calculated using the following formula: (item's sku_weight * unit_factor) * ship_qty. It is currently not being used for anything except informational purposes. In the future, it will be used as the basis for automatically calculating the shipping fees. (.iX shipping) (.iX units_of_measure)

commit_qty decimal(14),
The quantity of stock committed for this item. Stock is committed upon reaching the stage of "ORD" for stocking items only. commit_qty is stored in stockkeeping units, not selling units. the committed quantity is calculated as: ship_qty * unit_factor. The commit_qty will always be non-null (for SQL summing). It will be reset to 0 when the stage is set to PST. (.iX commit) (.iX units_of_measure)

pic_ticket_no smallint
picking ticket number that this item's quantity was shown on. picking tickets are numbered with the sales order and the pick_ticket_no appendage. example: 3386-02. When entering the amounts picked, this number shows on the screen to verify that you are entering the amount from the correct picking ticket (there may be several picking tickets for the order printed the same day). This column is used to verify that a picking document has been printed for this shipment line. If no pick ticket is necessary for the order, then this column is filled with 99 (.iX direct_ship).

mfs_printed char(1),
Has a shipping manifest been printed for this line (Y/N)? A NULL value means that there is no shipping manifest needed for this order type. A "N" value means that a shipping manifest needs to be printed, but hasn't been printed yet for this shipment line. A "Y" value means that the shipping manifest has already been printed for this shipment line.

inv_doc_no integer,
assigned at invoicing time. this is the document number used for various and sundry postings. it is obtained from the a/r control file if a/r is installed. otherwise, it is obtained from inv_doc_no in the order entry control table. It is used to uniquely join to the stoinvce table.

Multiple Customer/Ship-to/Bill-to Storage

sell_to_code char(20),

Normally this is the same as the order header's cust_code column. But, if the order header's cust_code points to a "bridge" type customer, then this sell_to_code may be any one of the customers that belong to that bridge account. Any one order line may have sales to up to 99 bridge customers. The order line quantity may be 10,000 units (for a price break), but sold to 10 different customers that are under the same "bridge" buying umbrella.

ship_to_code char(6),
This is the same as the order header's ship_to_code column. In the future the program will be as follows:there are multiple shipping addresses for the order, the different ship-to codes are stored as separate shipment lines. There can be up to 99 different ship_to addresses for every order line. The ship-to code is always attached to the sell-to customer, not the bill-to customer.

bill_to_code char(20),
Normally this is the same as the order header's bill_to_code column.Examples of exceptions are:1) A bridge buy order where a group of companies combine buying power but they're all billed separately. 2) That same scenario except the group authority is billed in total. 3) A combination of the above. 4) A customer barely goes over his credit limit, so he places some of the order on his credit card to keep the A/R portion under the limit. 5) A customer places an order where 4 items are leased (and billed to the leasing company), but the other 2 items are billed to the customer, not the leasing company. See the order header description of the bill_to_code for further explanation of its use.

Item & Warehouse information:

item_code char(20),
Inventory item code. It is stored in this table for quick access to items on order for a warehouse. It is stored as data vs. joined to the stoordrd table because the 'available for sale' process requires an index to item/warehouse and joins cannot be indexed.

warehouse_code char(3),
warehouse the item is to be shipped from. this is defaulted to the warehouse code in the order header, but can be overridden by the operator for the line when entering orders or credit/debit memos. it is also used with the item_code to find the item warehouse information. the warehouse code cannot be changed if the stage is greater than "ORD".

stock_location char(12),
This is defaulted from the inventory location record. For orders that have been put into a staging area and are now ready to ship, this column is filled with the staging area

(for those items that have been picked).

Pricing & Costing information:
Pricing information is stored at the shipment level because the price may change from one shipment to the next. This is a normal occurrence for 'market price' inventory items. It can also happen if a price is manually overridden for some reason. Example: A customer had to wait too long for the backordered items, so the first items were at regular price but the backordered items were sold at a 5% price reduction due to late shipment. Costing information also needs to be stored at the shipment level because the cost of a backordered item may be more than the cost of the same item sold from inventory.

price_group char(6),
This is the price_group column from the inventory table. It contains the group code that is used for group pricing. When items are 'group priced', the quantity of all items on the order (in the same group) are added to determine the quantity price break.

price decimal(14,4),
Price is computed based on the pricing table mechanism. It is stored in selling units. The operator can override the computed price if authorized to do so. If quantity, item code, warehouse code, or unit of measure is changed, the price will be recomputed. This price represents the actual price used in the billing.

orig_price decimal(14,4),
This price is the one that is defaulted from the pricing scheme for this order. If the order entry person changes the price, this original price is kept intact. This allows for exception reporting on price overrides. If the order entry person does not change the price retrieved for the item, this will match the price column. It represents the price for one selling unit. The final result here is that prices are not recalculated if orig_price differs from price, because the user must have manually overidden the price field.

retail_price decimal(14,4),
This is the item's retail price (from the inventory tables - modified by unit_factor) that the pricing scheme uses as the basis for calculating the orig_price. It serves to fix the basis price in time. If the order line needs to re-calculate the price, it uses this one as the basis vs. a potentially different price in the inventory tables. It represents the price for one selling unit.

price_approval char(6),
Approval code for a price override. If the line type requires an approval code for price override, the entry clerk must type in an approval code. This code isn't checked against anything, but it will show up on a price override

```
        exception report.

    tax_rate decimal(6), Obsolete, Field no longer used.
        This is the summed state, county, and city tax rates for this
        ship-to address.  It is used to calculate the tax_amount for
        this shipment line.  8.2% would be stored as .082 (.iX taxes)

    tax_amount decimal(14), Obsolete, Field no longer used.
        If this item is taxable, this column contains the 3 tax rates
        (summed) * net_amount.  It contains 0 if the line isn't
        taxable.

    mtaxg_code char(6)
        Multilevel Tax group code assign to this shipment.

    net_amount decimal(14),
        Extended total shipment amount.  price * ship_qty = net
        amount.

    item_cost decimal(14,4),
        This is the cost of the item (in selling units).  It is
        brought over as the average unit cost from the inventory
        tables, or entered by the operator on a non-inventory type
        item.  This cost is adjusted to the actual cost (based on the
        LIFO/FIFO or Serialized Inventory) at the time the order line
        is posted.

    gross_margin decimal(10),
        This is the calculation:   (price - item_cost) * ship_qty.
        It is stored for informational purposes & easy access to
        selecting and sorting on this data via SQL.

Date stamps:
    new_date date,
    bko_date date,
    ord_date date,
    pic_date date,
    shp_date date,
    inv_date date,
    pst_date date,
    can_date date,
        These are the dates that the shipment record changed to the
        indicated stage.

Shipment Dates:
        these dates are for reference only.

    proj_ship_date date,    - projected ship date
    request_date date,      - date requested
    reject_date date,       - date rejected
    actual_ship_date date,  - actual ship date
    fact_sched_date date
    ship_via_cd char(3)     - ship via code
```

```
    po_doc_no integer       - purchase order document number
    ship_lookup char(30)    - Interface to Fedex and UPS web services
    atp_date date           - ATP Date
    index1: doc_no, line_no, ship_no (unique)    index2: stage

-------------------------------------------------------------------------
stoinvce        Invoice Header Table

  One row for each invoice and  credit/debit memo produced.  There will
  always be exactly one row for an order of credit or debit memo type.
  Any one invoice represents one shipment to a ship-to location.  If
  there are multiple ship-to locations for an order, several invoices
  are generated - one for each shipment to each ship-to location.

    doc_no integer not null,
        Order document number.

    order_no char(20),
        Sales order number.

    bill_to_code char(20),
        This is the code for the customer that is going to be billed
        on this invoice.  Before the invoice is printed, there is no
        inv_doc_no, and this column forms the unique join to the
        invoice.

    sell_to_code char(20),
        This is the sell_to_code associated with the ship_to_code for
        this shipment.

    ship_to_code char(6),
        Billing customer's shipping code for the invoice.  This is
        validated from the customer/ship-to tables.  If the code
        contains "SHIPTO" then the system uses the billing customer's
        billing address.

    inv_doc_no integer,
        Assigned at invoice/memo print time.  If this column is NULL,
        then this row contains the invoice/memo totals for the NEXT
        invoice for this customer (on this order or credit/debit
        memo).  If the column is not null, then the invoice or memo
        has already been printed and this column serves to uniquely
        join the order to the specific invoice.

Invoice information:

    stage char(3),
        The stoinvce row is created when an stoshipd line becomes
        stage INV.  Also, you cannot cancel an order once it has been
        approved for billing (stage INV).  Therefore, the only 2
        possible stages for this table are INV and PST. (.iX stages)

    inv_no char(10),
```

Invoice number, assigned at invoice/memo print or billing entry.
Normally, this is a copy of inv_doc_no. If the invoice number is assigned outside of the OE system, it is stored here.

inv_date date,
This is the date that the invoice was printed. It is set by the invoicing function.

inv_printed
Set to "Y" when an invoice is printed.
If inv_printed is set to "Y", the user cannot update the invoice.

ok_to_post char(1) not null,
Set to "Y" in edit run of posting program. Marks this invoice as postable. Reset to N if the invoice is changed in any fashion.

Payment information:

terms_code char(6),
A/R terms code. Retrieved from the bill-to customer record.

Future programming change will be as follows:
If the OE setup file says it's ok to override this, then the order entry person may change the terms_code. They may be required to provide an override code. The terms_code may be set to "COD" if the customer's credit limit is exceeded and the OE setup file says it's ok to process COD orders exceeding the customer's credit limit.

terms_approval char(6), For Future Use
Approval code for terms_code override. If the OE setup file requires an approval code for terms override, the entry clerk must type in an approval code. This approval code isn't checked against anything, but it will show up on an override exception report.

pay_method char(6)
This code is defaulted from the customer table. It is validated from the stxinfor table, and to provide the G/L account number for the payment.
CASH/VISA/AMEX/MC/ONACCT are pay_method examples.

payment char(1),
This code determines which of the 3 different types of payment method used. It is looked up from the stxinfor table based on the key entered in pay_method (above).

A - accounts receivable
C - cash
V - credit card

3 - 3rd party billing - Field for Future Use

card_no char(20)
    used to store the credit card number if paying by card. it is defaulted from the customer table, but can be overridden. this data is only valid for credit card type payments

exp_date char(5)
    expiration date for credit card payments.

check_no char(8)
    if paying via cash, this would be the check number used for payment. if paying via credit card, this column contains the credit card companies' authorization code for this purchase.

Shipping information

fob_point char(15),
    Free On Board point. Printed on the order acknowledgement, picking and shipping documents, and invoice. The FOB point is where the title to the goods is transferred. The customer is responsible for freight charges from the FOB point to the shipment destination.

ship_via char(15),
    Default shipment carrier. This is a required field of entry for non reference type orders. Since an order can have many shipments (and many shipping carriers), the REAL shipment carrier is stored with the invoice totals (in the stoinvce table).

ship_weight decimal(14),
    This column represents the total weight for the order. It is calculated by adding the individual weights of the order shipment lines. It is currently not being used for anything except informational purposes. In the future, it will be used as the basis for automatically calculating the shipping fees.

freight_doc char(15),
    Freight document number.

Taxes & discounts:

st_tx_code char(6), Obsolete, Field no longer used
co_tx_code char(6), Obsolete, Field no longer used
ci_tx_code char(6), Obsolete, Field no longer used

st_tx_rate decimal(6), Obsolete, Field no longer used
co_tx_rate decimal(6), Obsolete, Field no longer used
ci_tx_rate decimal(6), Obsolete, Field no longer used

tax_rate decimal(6), Obsolete, Field no longer used

This is the sum of the state, county, and city rates for this shipto address. It is used to calculate the tax_amount based on the total taxable amount for this invoice. 8.2% tax would be stored as .082 (.iX taxes)

trd_ds_rate decimal(6),
This is a rate to apply to the sum of the discountable invoice lines to determine the amount of trade discount to apply to the invoice. 20% discount would be stored as .2 (.iX discounts)

mtax_freight char(8),
Multilevel tax code for freight

Invoice total information:

item_amount decimal(14),
Total of the invoiceable order lines.

discountable decimal(12),
Total of the amounts that are discountable for this invoice. items in lines may or may not be discountable (this is set in the item row in the inventory control module). also, freight may or may not be discountable. this is set in the order entry control table.

trd_ds_amount decimal(14),
amount of the trade discount for this invoice. computed by taking the discountable amount multiplied by the trd_ds_pct. this amount is DEDUCTED from the order total.

taxable decimal(12), Obsolete, Field no longer in use.
total of invoice amounts that are taxable. items in lines may or may not be taxable (this is set in the item row in the inventory control module). also, freight may or may not be taxable. this is set in the order entry control table.

st_tx_amount decimal(12), Obsolete, Field no longer in use.
amount of state tax for the invoice. computed by taking the taxable amount multiplied by st_tx_pct. on the order entry screen, these three are shown as a sum.

co_tx_amount decimal(12), Obsolete, Field no longer in use.
amount of county tax for the invoice. computed by taking the taxable amount multiplied by co_tx_pct.

ci_tx_amount decimal(12), Obsolete, Field no longer in use.
amount of city tax for the invoice. computed by taking the taxable amount multiplied by ci_tx_pct.

frght_amount decimal(14),
amount of freight to charge for the invoice. entered by the operator.

```
    total_amount decimal(14),
        invoice total amount.  this consists of this sum:
            item_amount
          - trd_ds_amount
          + ci_tx_amount
          + co_tx_amount
          + st_tx_amount
          + frght_amount
          = total_amount

General ledger codes:  these are the general ledger accounts and
 departments for the invoice.

    td_ds_acct integer,
        Trade discount contra-income account number.  Defaulted with
        the disc_acct_no in the oe control table.

    st_tx_acct integer, Obsolete, Field no longer used
    co_tx_acct integer, Obsolete, Field no longer used
    ci_tx_acct integer, Obsolete, Field no longer used

    freight_acct integer,
        Outgoing freight income account number.  Defaulted with the
        frght_acct_no in the oe control table.

    asset_acct integer,
        a/r, cash, or credit_card asset account number.  Defaulted
        with the ar|cash|visa|_acct_no in the oe control table.
        the acct to use is determine by the payment column[A|V|C].

    td_ds_dept char(3),
        Trade discount department.  Defaulted from the order header
        department.

    st_tx_dept char(3), Obsolete, Field no longer used
    co_tx_dept char(3), Obsolete, Field no longer used
    ci_tx_dept char(3), Obsolete, Field no longer used

    freight_dept char(3),
        Freight department.   Default from the order header
        department.

    asset_dept char(3),
        a/r, cash, or credit_card department.  If
        stocntrc.use_department = "Y" then default from department.
        If stocntrc.use_department = "N" then default "000".

    mtaxg_code char(6),  - the default tax code to use for each
        detail line.

    tax_amount decimal(14) - Obsolete, Field no longer in use.
    The total tax amount for this invoice.
```

```
    batch_id  integer - Batch Control ID

    ship_terms char(15)

    mtax_freight char(6)

    settle_decl_code char(8),
        Decline Code in settlement

    decline_message char(60),
        Decline message in settlement

    settled char(1),
        Has invoice been settled?

    cc_batch_id char(12)
        Credit Card Batch Id in SkipJack

    deposit_applied decimal(12,2)
        Deposit Applied

    restock_fee decimal(8,2)
        Restocking Fee

    handling_fee decimal(8,4));
        Special Handling charges

    split_terms_code char(6);
        Split Payment Terms

-------------------------------------------------------------------------
stolockr           Order Entry Lock Tables

  Each row in this table contains the notes on how the table is
  used during posting, and the effects of locking it during posting.

    tabname char(18),
        table name

    lock_required char(1)
        Y/N.  if "Y" is specified, then during posting the specified
        table will be locked for the duration of the begin/commit
        work for each document.  the only reason you would want to
        lock a table is if you run out of unix system resources (too
        many open files, or too many record locks).  the sample data
        comes with all tables that are affected during posting, and
        each of them have a lock_required of "N".  If during posting
        you run out of system resources, you can do one of 3 things:

          1) re-compile your unix kernel for more locks/open files
          2) consider moving your database to informix's "online"
             (formerly "turbo") engine.  this does not use up unix
```

resources for open tables & file locks.
3) mark certain tables for locking at the time of posting

---

stopricr Special Pricing Header

This table overrides the normal inventory price stored in the I/C tables. The system finds all pricing records that match any of the item, item class, customer, customer class, order type, and date range criteria. It then groups them by price_level. The highest price_level records make the final cut. If there is more than one record in the final cut (within the same price_level), then they are default ordered as follows:

1) Order type
2) Inventory item
3) Item class
4) Customer
5) Customer class
6) Selling unit
7) None of the above (date match with all other columns null)

Once the correct pricing record is selected, the order quantity is found in the quantity/price break detail table, and the corresponding pricing discount code is determined. The system then looks up that discount code to determine if it is a markup from cost or a markdown from price, and the percentage of markup/markdown.

We suggest assigning different price levels based on your pricing priorities. This assures you get the pricing you want if more than one match occurs.

price_code serial,
This is the unique identifier for this pricing record.

description char(30),
Verbal description of this pricing record.

price_level smallint,
Arbitrary pricing levels 0-9. If more than one pricing record matches the pricing criteria, they are grouped together and ordered on this column. The record in the group with the highest price_level wins (ie: that is the price used). If a pricing scheme has a low priority, assign it a lower number. If it has a high priority (ie: it should override other pricing matches), then it should be assigned a higher price_level. For example, the basic, everyday pricing scheme would be level 0, while level 1 (used in conjunction with a date range) would be used for special promotion.

item_code char(20),
Item code for the special pricing (if specifying pricing based on inventory items). item_code and item_class are mutually exclusive. Only one may be specified.

item_class char(6),
    Inventory class for the special pricing (if specifying
    pricing based on class of inventory). item_code and
    item_class are mutually exclusive. Only one may be
    specified.

cust_code char(20),
    Customer code for the special pricing (if specifying pricing
    based on customer codes). cust_code and trd_ds_code are
    mutually exclusive. Only one may be specified.

trd_ds_code char(6),
    Customer class for the special pricing (if specifying pricing
    based on customer class, the class is stored in the
    trd_ds_code column of the customer table). cust_code and
    trd_ds_code are mutually exclusive. Only one may be
    specified.

order_type char(3),
    Order type for the special pricing (if specifying pricing
    based on sales order types)

sell_unit char(2),
    Unit of measure for this item_code. This defaults to null.
    when null, the unit of measure is assumed to be the sell_unit
    for the item_code. It may be overridden with any of the
    other unit measures that are valid for the item_code.

begin_date date,
end_date date,
    Begin/end date for the special pricing (if any)

disc_type char(1),
    This column is described in detail under stopricd. The value
    in this column is also stored in disc_type in the quantity
    price break detail table as a programming convenience.

tolerance_level   smallint
    Tolerance Level

warehouse_code char(10)
    Warehouse Code

---

stopricd   Special Pricing Detail Table

This is the quantity/price break detail table. Once the correct pricing record is selected from the price header table, the order quantity is found in this table, and the corresponding pricing discount code is determined. The system then looks up that discount code to determine if it is a markup from cost or a markdown from price, and the percentage of markup/markdown. The

```
  disc_code may be an amount or a markup/discount percentage rather
  than a code.  This is determined by the value of disc_type.

Price header & quantity price break detail join criteria:
    price_code integer,
        This is the join column that matches the detail with the header.

Discount information:
    disc_type char(1),
        This column contains the same disc_type as the header as a
        programming convenience.  This column contains a type code that
        describes the data in the disc_code column.  If this column
        contains a "C" (code), then the data in the disc_code column is
        a code that's validated in the stxinfor table.  The discount
        rate is found in stxinfor.  If the disc_type column contains a
        "D", then disc_code contains a percent of discount that should
        be applied.  If the disc_type column contains a "M", then
        disc_code contains a percent of markup over cost that should be
        applied.  If the disc_type column contains an "A", then the
        disc_code column will contain an exact amount to charge for the
        specified quantity (item_code is required when specifying a
        disc_type of "A").

    disc_qty decimal(10),
        Pricing quantity level.  If the order quantity is greater than
        or equal to this level, but less than the next higher level,
        then the disc_code from this row is used to compute price.

    disc_code char(12),
        Either the discount code, the discount percent, markup percent,
        or the specific item price based on the contents of the
        disc_type column.

-------------------------------------------------------------------------
stokitre       Kit Header Table

     kit_code char(15) not null,   # uniq code to identify the kit
     desc1 char(30),               # kit description line 1
     desc2 char(30)                # kit description line 2

-------------------------------------------------------------------------
stokitrd       Kit Detail Table

    kit_code char(15),             # uniq code tied to stokitre
    line_no smallint,              # order of expansion
    item_code char(20),            # inventory item code
    ordr_qty decimal(12),          # quantity in shipping units
    include_price char(1)          # include price during expansion
-------------------------------------------------------------------------
storecur - Recurring Usage Cross Reference Table

    This table allows the user to define what order and
    line types are treated as a recurring order in the
```

```
    replenishment system.

    order_type char(3),  # order type/any order type(stootypr.order_type)
    line_type char(3),   # line type/must of "stock item"
    recurr_usage char(1) # is the combination a recurring

    ie: if order type is filled and line type is null, then
    all orders of that order type will track as recurring usage.

    the reason to have recurring usage flag is to do exceptions.
    ie: if i want all "REG" orders to be track as recurring but
        i want "REG"/"STN" to be non-recurring, then i would
        have a record of "REG"/""/"Y" and another record as
        "REG"/"STN"/"N".
-------------------------------------------------------------------------
stoakasr  - Alias Reference Table

   cust_code char(20),        # Customer Code
   alias  char(20)            # Customer's Alias Name for Inventory Item
   item_code char(20)         # Inventory Item Number

index i1oakasr on stoakasr (cust_code, alias)
-------------------------------------------------------------------------
stocmitd - Committed Detail Table

   doc_no integer not null    # Document Number
   line_no smallint not null  # Line Number
   ship_no smallint           # Shipment Number
   item_code char(20)         # Item Code
   warehouse_code char(10)    # Warehouse Code
   commit_qty decimal(14)     # Committed Quantity

index i1stocmitd on stocmitd(doc_no, line_no, ship_no)
-------------------------------------------------------------------------
stoshpar -  UPS/FEDEX

   ship_lookup_key char(40),
   ship_custcode char(20),
   ship_name char(30),
   ship_addr1 char(30),
   ship_addr2 char(30),
   ship_city char(20),
   ship_state char(2),
   ship_zip char(10),
   ship_country char(20),
   ship_contact char(20),
   ship_phone char(20),
   ship_send_type char(30),
   billing_option char (20),
   ups_account char(10),
   po_no char(24),
   order_no char(20),
   email_address char(50),
```

```
   qvn_option char(1),
   residential_cust char(1),
   fax_email char(6),
   ship_notify char(1)
   ship_type char(6),
   package_type char(30)

index i1oshpar on stoshpar(ship_lookup_key)
--------------------------------------------------------------------------
stoshpfr - UPS/FEDEX

   pack_lookup_key char(40),
   pack_tracking_no char(40),
   pack_freight decimal(10,2),
   pack_shp_date char(10),
   pack_shp_boxes char(4),
   pack_void_flag char(1),
   pack_del_date date,
   pack_del_city char(20),
   pack_del_state char(2),
   pack_del_type char(20),
   pack_timestamp DATETIME year TO second
                  default current year to second,
   order_no char(20),
   shp_weight decimal(9,1),
   shp_cost decimal(14,4)
   ship_type char(6),
   package_type char(30),
   stage char(5),
   inv_doc_no integer

index i1oshpfr on stoshpfr(pack_lookup_key)
--------------------------------------------------------------------------
stoshptr - UPS

   ship_terms       char(15),
   description      char(50),
   invoice_freight  char(1),
   order_limit      decimal(12,2)

unique index i1oshptr on stoshptr(ship_terms)
--------------------------------------------------------------------------
stoupstr - UPS/FEDEX

    ship_type        char(6),
    ship_code        char(15),
    ship_desc        char(25)
    ship_code        char(30)

unique index i1stoupstr on stoupstr(ship_type, ship_code)
--------------------------------------------------------------------------
stoshtxd - sales tax rate at line level
```

```
    doc_no integer not null ,    # Order doc number
    line_no smallint not null ,  # Order line number
    ship_no smallint,            # Order Ship number
    tax_code char(6),            # Tax Code
    tax_rate decimal(6),         # Tax Rate
    net_amt decimal(12),         # Net amount
    tax_amt decimal(12)          # Tax Amount

unique index i1oshtxd on stoshtxd (doc_no,line_no,ship_no,tax_code)
------------------------------------------------------------------------
stocontr - THIS TABLE IS NOT IN USE

   cust_code            char(20) not null
   contract_no          char(30) not null
   expire_date          date
------------------------------------------------------------------------
stocustr - THIS TABLE IS NOT IN USE

   cust_code            char(20)
   cust_type            char(3)
   sales_commit         decimal(12)
   sales_year           date
   price_protect        char(1)
   gross_sales_amt      decimal(12)
   co_code              char(3)
   region_code          char(3)
   division_code        char(2)
   terr_code            char(4)
   ship_complete        char(1)
   open_po              char(1)
   open_po_no           char(15)
   open_po_expire       date
   early_ship           char(1)
   stop_ship            char(1)
   stop_ship_begin      date
   stop_ship_end        date
   edi_recvd            char(1)
   invoice_format       char(3)
   invoice_sort         char(1)
   discount_qual        char(1)
   letter_of_cred       char(1)
   intl_lic_no          char(30)
   fed_excpt_flag       char(1)
   fed_excpt_date       date
   credit_hold          char(1)
------------------------------------------------------------------------
stofedce - THIS TABLE IS NOT IN USE

   doc_no               integer not null
   fed_cert             char(1)
   contact              char(30)
   fed_agency           char(1)
   fed_contract_no      char(30)
```

```
   fed_eu_support       char(1)
   fed_comrcl_tc        char(1)
------------------------------------------------------------------------
stoholdc - THIS TABLE IS NOT IN USE

   crdt_hold            char(3)
   note_hold            char(3)
   ship_hold            char(3)
   gov_hold             char(3)
   intl_hold            char(3)
   crd1_hold            char(3)
   aval_hold            char(3)
   tod_hold             char(3)
   sit_hold             char(3)
   conf_hold            char(3)
   temp_hold            char(3)
   appv_hold            char(3)
   frej_hold            char(3)
   price_hold           char(3)
   nshp_hold            char(3)
   cust_hold            char(3)
------------------------------------------------------------------------
stoholdd - THIS TABLE IS NOT IN USE

   doc_no               integer  not null
   line_no              smallint
   ship_no              smallint
   hold_type            char(3)
   like_type            char(3)
   user_name            char(20)
   hld_date             date
   hld_time             char(8)
   progid               char(17)
   reason               char(30)
   hld_rls_date         date
   stage                char(3)
------------------------------------------------------------------------
stoholdr - THIS TABLE IS NOT IN USE

   hold_type            char(3)
   like_type            char(3)
   description          char(30)
   dept                 char(5)
   stage                char(3)
------------------------------------------------------------------------
storegir - THIS TABLE IS NOT IN USE

   region_code          char(3)
   description          char(40)
   mkup_pct             smallint
   kit_group            smallint
   stage                char(3)
   ship_qty             decimal(10)
```

```
--------------------------------------------------------------------------
stormacr - THIS TABLE IS NOT IN USE

   rma_code             char(3) not null
--------------------------------------------------------------------------
storpair - THIS TABLE IS NOT IN USE

   order_type           char(3)
--------------------------------------------------------------------------
stocclog - Credit card batch log

    html_serialno char(12),
    cc_batch_id char(12),
    cc_batch_date date

unique index i1occlog on stocclog (html_serialno, cc_batch_id);
--------------------------------------------------------------------------
stoupsrc - Interface to Fedex and UPS web services

    ups_account  char(40), # UPS Account
    ups_login    char(40), # UPS Login
    ups_password char(40), # UPS Password
    ups_version  char(1)   # (X)ML or (H)TML
    ship_type char(6),
    meter_no char(15),
    version_id_svc char(10),
    version_id_major char(10),
    version_id_int char(10),
    version_id_minor char(10),
    authent_key char(30),
    authent_pwd char(30),
    svc_available char(1)
  );

unique index i1oupsrc on stoupsrc (ups_account);
constraint check (ups_version IN ("X" ,"H")) constraint c1oupsrc ;
--------------------------------------------------------------------------
stoccard - Credit Card Master Table

    cust_code char(20) not null ,  # Customer Code
    card_name char(20) not null ,  # Card Name
    card_number char(4) not null , # Last Four digits credit card number
    exp_mo char(2) not null ,      # Expiration month
    exp_year char(4) not null ,    # Expiration year
    first_name char(20) not null , # First name
    middle_initial char(1),        # Middle name
    last_name char(30) not null ,  # Last name
    address1 char(30) not null ,   # address
    address2 char(30),             # address
    city char(20) not null ,       # City
    state char(2) not null ,       # State
    zip char(10) not null ,        # Zip Code
    country char(20),              # Country
```

```
    phone char(20) not null ,      # Phone
    email char(50) not null ,      # Email
    fax char(20)                   # Fax

create unique index i1occard on stoccard(cust_code,card_name,card_number);
------------------------------------------------------------------------

stoordsd - Table used to insert or delete order lines in i_order

    doc_no integer,
    line_no smallint,
    dock_receipt date,
    to_repair_ctr date,
    repair_complete date,
    to_customer date,
    turn_time smallint,
    sched_ship date,
    rep_rep_prt date,
    symptom char(40),
    serial_no char(20),
    last_chg_user char(8),
    last_chg_date date

create unique index i1oordsd on stoordsd(doc_no,line_no);
---------------------------------------------------------------------------
stotrckd - Track master table
    contract_no char(20),     - Contract
    doc_no integer,           - Sales order Doc number
    line_no smallint,         - Sales order Line number
    po_doc_no integer,        - PO Doc number
    fwdr_invoice char(20),    - forwarder's invoice
    sales_basis char(5),      - Sales Basis
    loadg_location char(30),  - Location where loaded
    port_of_exit char(30),    - Port to Exit
    gross_weight decimal(10), - Gross Weight
    net_weight decimal(10),   - Net Weight
    tare_weight decimal(10),  - Tare Weight
    container_no char(20),    - Container Number
    bill_lading char(20),     - Bill of Lading
    ocean_bol char(20),       - Ocean bill of lading
    booking_no char(20),      - Booking number
    vessel char(25),          - Vessel
    voyage char(25),          - Voyage
    cutoff_date date,         - Cut off Date
    loadg_date date,          - Loading Date
    release_date date,        - Release Date
    est_depart date,          - Estimated Departure
    est_arrive date,          - Estimated Arrive
    pymt_due_date date,       - Payment Due Date
    est_demurrage decimal(10,2), - Est Demurrage
    insurance decimal(10,2),  - Insurance
    relse_rqstd date,         - Release requested date
    relse_recvd date,         - Release Receviced date
```

```
    draft_obl_rcvd date,      - Draft OBL date
    consignee_name char(20),  - Consignee Name
    consignee_addr1 char(30), - Consignee Addr
    consignee_addr2 char(30), - Consignee Addr
    consignee_city char(20),  - Consignee City
    consignee_state char(2),  - Consignee State
    consignee_zip char(10),   - Consignee Zip
    consignee_country char(20), - Consignee Country
    notify_name char(20),     - Notify Name
    notify_info char(240),    - Notify Info
    gross_mt decimal(8,2),    - Gross Amount
    net_mt decimal(8,2),      - Net Amount
    tare_mt decimal(8,2),     - Tare Amount
    forwarder_name char(20),  - Forwarder Name
    carrier char(20),         - Carrier
    send_docs char(20),       - Send Original Documents
    container_size char(20),  - Container Size
    transhipment1 char(20),   - Transhipment
    vessel1 char(20),         - Vessel 1
    voyage1 char(10),         - Voyage 1
    eta1 date,                - ETA Date
    etd1 date,                - ETD date
    transhipment2 char(20),
    vessel2 char(20),
    voyage2 char(10),
    eta2 date,
    etd2 date,
    transhipment3 char(20),
    vessel3 char(20),
    voyage3 char(10),
    eta3 date,
    etd3 date,
    transhipment4 char(20),
    vessel4 char(20),
    voyage4 char(10),
    eta4 date,
    etd4 date,
    transhipment5 char(20),
    vessel5 char(20),
    voyage5 char(10),
    eta5 date,
    etd5 date,
    formula_type char(1),       - Formula Type
    formula_value decimal(6,2), - Formula Value
    comex_price decimal(6,2),   - Comex Price
    formula_status char(1)      - Formula Status
    port1 char(10),
    port2 char(10),
    port3 char(10),
    port4 char(10),
    port5 char(10),
    seal_no char(10)
-------------------------------------------------------------------------
```

```
stoorwte - Weight Information (header)
    doc_no      integer,  - Order doc number
    line_no     smallint, - Order line number
    ship_no     smallint, - Order ship number
    po_doc_no   integer,  - PO order number for DIR
    po_line_no  smallint, - PO line number for DIR
    total_units smallint  - total of units "sum(stoorwtd.unit_no)

index i1oorwte on stoorwte (doc_no, line_no, ship_no);

--------------------------------------------------------------------------
stoorwtd - Weight Information (warehouse shipment) detail
    doc_no      integer,  - Order doc number
    line_no     smallint, - Order line number
    ship_no     smallint, - Order ship number
    po_doc_no   integer,  - PO order number for DIR
    po_line_no  smallint, - PO line number for DIR
    seq_no      smallint, - sequential number
    unit_no     smallint, - number of units
    uom         char(6),  - unit of measure
    gross       decimal(12), - gross weight
    tare        decimal(12), - tare
    net         decimal(12)  - gross - tare

index i1oorwtd on stoorwtd (doc_no, line_no, ship_no, seq_no);

--------------------------------------------------------------------------
stormarr - RMA Reason reference table
    rma_reason char(6),      - RMA Reason Code
    rma_description char(30) - Reason Description
--------------------------------------------------------------------------
stoarctr - archive order entry tables into history tables
    sys_table_name char(20),   - System Table Name
    arch_table_name char(20),  - Archive Table Name
    join_column char(20)       - Join Column

--------------------------------------------------------------------------

stoitmpd - Item Notes with Selective Print
     prog_no smallint,   - Program Number
     item_code char(20), - Item Code
     prt_notes char(1)   - Print Notes

--------------------------------------------------------------------------
stoitmpr - Item Notes with Selective Print
    prog_no smallint,     - Program Number
    prt_module char(8),   - Program Module
    prt_program char(8),  - Program
    prt_description char(30), - Description

    ROWS LOADED
   (1, "oe", "o_quote", "Sales Quotations");
   (2, "oe", "o_order", "Order Acknowledgements");
```

```
   (3, "oe", "o_picker", "Picking Documents");
   (4, "oe", "o_shipr", "Packing Slips");
   (5, "oe", "o_invce", "Invoices And Memos");
   (6, "pu", "o_order", "Purchase Orders");
-------------------------------------------------------------------------

stomlsoe - Multiple Orders Streamline process (use internally)
    doc_no            integer,  - Doc Number
    init_doc_no       integer,  - Init Doc Number
    next_doc_no       integer,  - Next Doc Number
    inv_doc_no        integer   - Invoice Doc Number

-------------------------------------------------------------------------
stoordrh Interface to Fedex and UPS web services
     doc_no integer,              - Document Number
     ship_type char(6),           - Ship Type
     service_type char(50),       - Service
     package_type char(30),       - Package Type
     weight decimal(9,2),         - Weight
     weight_unit char(5),         - Unit
     cost_amount decimal(9,2),    - Cost
     charge_amount decimal(9,2),  - Charge Amount
     estimate_date date,          - Estimate Date
     commit_days smallint         - Commit Days

-------------------------------------------------------------------------
stopicke - Streamline order entry  (Use in the PT reprint functionality -
Internal)
    doc_no integer,    - Doc Number
    line_no smallint,  - Line Number
    ship_no smallint,  - Ship Number
    orig_ship_qty decimal(14) - Original Ship Qty

-------------------------------------------------------------------------
stopromp - Streamline order entry (Use in prompt fuctionality - Internal)
    prompt_type char(10),
    doc_no_list char(500)


-------------------------------------------------------------------------
storecoh - Recurring Orders. (Header)
   reorder_doc_no integer,   - Reorder Doc Number
   order_doc_no integer,     - Order Doc Number
   create_id char(8),        - Create ID
   duplication_date date,    - Duplication Date
   duplication_time char(8)  - Duplication Time

-------------------------------------------------------------------------
storecod - Recurring Orders. (Detail)
   reorder_doc_no serial,    - Reorder Doc Number
   order_doc_no  integer,    - Order Doc Number
   cycle_frequency  char(1), - Cycle Freq.
   cycle_code  char(1),      - Cycle Code
```

```
   start_date date,          - Start Date
   end_date  date,           - End Date
   current_price  char(1),   - Current Price
   next_reorder_date date    - Next Reorder Date


------------------------------------------------------------------------
stormasr - RMA Status
    status_code       char(20), - Status Code
    description       char(40)  - Description

------------------------------------------------------------------------
stoshrbd - Create rebate import mapping tables
    doc_no integer,            - Doc Number
    line_no integer,           - Line Number
    ship_no integer,           - Ship Number
    item_code char(20),        - Item Code
    rebate_cost decimal(11,4), - Rebate Cost
    vend_code char(20),        - Vendor Code
    posted char(1)             - Posted

------------------------------------------------------------------------
stospltd - Split Payment Terms
     order_doc_no integer,    - Order Doc Number
     orig_inv_doc_no integer, - Original Invoice
     split_inv_doc_no integer - Split Invoice

------------------------------------------------------------------------
stotermd - Split Payment Terms
     split_terms_code char(6), -- join to stotermh
     line_no smallint,         -- for ordering of entries
     terms_code char(6),       -- terms code from strtermr
     terms_percent decimal(6)  -- percent of order at this term

stotermh - Split Payment Terms
     split_terms_code char(6),  - Split Terms Code
     split_terms_desc char(30), - Split Terms Desc
     cm_memo_acct integer,      - for debit account for credit memo
attached to the
                                  original invoice, the credit account for
the split
                                  out invoices if not left blank
     split_ar_acct integer      - Debit account for split out invoices if
not left blank.


------------------------------------------------------------------------
stoupsws - Interface to Fedex and UPS web services
     ship_type char(6),
     service_name char(30),
     version_id_svc char(10),
     version_id_major char(10),
     version_id_int char(10),
```

```
     version_id_minor char(10)

--------------------------------------------------------------------------
aroinvce - Archive stoinvce
    doc_no integer not null ,
    order_no char(20),
    bill_to_code char(20),
    sell_to_code char(20),
    ship_to_code char(6),
    inv_doc_no integer,
    stage char(3),
    inv_no char(10),
    inv_date date,
    inv_printed char(1),
    ok_to_post char(1) not null ,
    terms_code char(6),
    terms_approval char(6),
    pay_method char(6),
    payment char(1),
    card_no char(20),
    exp_date char(5),
    check_no char(8),
    fob_point char(15),
    ship_via char(15),
    ship_weight decimal(10),
    freight_doc char(15),
    st_tx_code char(6),
    co_tx_code char(6),
    ci_tx_code char(6),
    st_tx_rate decimal(6),
    co_tx_rate decimal(6),
    ci_tx_rate decimal(6),
    tax_rate decimal(6),
    trd_ds_rate decimal(6),
    item_amount decimal(12),
    discountable decimal(12),
    trd_ds_amount decimal(12),
    taxable decimal(12),
    st_tx_amount decimal(12),
    co_tx_amount decimal(12),
    ci_tx_amount decimal(12),
    frght_amount decimal(12),
    total_amount decimal(12),
    td_ds_acct integer,
    st_tx_acct integer,
    co_tx_acct integer,
    ci_tx_acct integer,
    freight_acct integer,
    asset_acct integer,
    td_ds_dept char(3),
    st_tx_dept char(3),
    co_tx_dept char(3),
    ci_tx_dept char(3),
```

```
    freight_dept char(3),
    asset_dept char(3),
    mtaxg_code char(6),
    tax_amount decimal(12),
    currency_code char(3),
    curr_rate_type char(6),
    currency_rate decimal(16),
    batch_id integer,
    ship_terms char(15),
    mtax_freight char(6),
    settle_decl_code char(8),
    decline_message char(60),
    settled char(1),
    cc_batch_id char(12),
    deposit_applied decimal(12,2),
    restock_fee decimal(8,2),
    handling_fee decimal(8,4)


--------------------------------------------------------------------------
aroordrd - Archive stoordrd
    doc_no integer not null ,
    line_no smallint,
    kit_group smallint,
    kit_line_no smallint,
    last_ship_line smallint,
    line_type char(3),
    like_type char(3),
    hi_stage char(3),
    lo_stage char(3),
    cm_dm_reason char(3),
    our_po_no char(20),
    sls_psn_code char(6),
    warehouse_code char(10),
    item_code char(20),
    desc1 char(30),
    desc2 char(30),
    alias_code char(20),
    vend_code char(20),
    interchanged char(1),
    serialized char(1),
    td_disc_allowed char(1),
    tax char(1),
    ordr_qty decimal(14),
    back_qty decimal(14),
    commit_qty decimal(14),
    sell_unit char(2),
    unit_factor decimal(6),
    price decimal(12),
    price_code integer,
    tax_amount decimal(12),
    net_amount decimal(14),
    ship_weight decimal(14),
```

```
    inv_acct integer,
    inv_dept char(3),
    sls_acct integer,
    sls_dept char(3),
    cog_acct integer,
    cog_dept char(3),
    intl_lic_no char(30),
    price_lock char(1),
    release_qty decimal(14),
    resale_price decimal(18),
    mtaxg_code char(6)

--------------------------------------------------------------------------
aroordre - Archive stoordre
    doc_no integer not null,
    orig_doc_no integer,
    order_no char(20),
    inv_doc_no integer,
    inv_no char(10),
    po_no char(24),
    pic_ticket_no smallint,
    next_kit_group smallint,
    ack_printed char(1),
    order_type char(3) not null,
    like_type char(3) not null,
    order_status char(3) not null,
    hi_stage char(3) not null,
    lo_stage char(3) not null,
    bo_allowed char(1),
    recur_unit char(1),
    recur_every smallint,
    recur_times smallint,
    recur_through date,
    prev_recur date,
    next_recur date,
    num_releases smallint,
    release_type char(3),
    order_date date,
    to_ship_date date,
    alloc_date date,
    ship_date date,
    complete_date date,
    warehouse_code char(10),
    department char(3),
    sls_psn_code char(6),
    cust_code char(20),
    ship_to_code char(6),
    bill_to_code char(20),
    bus_name char(30),
    contact char(20),
    address1 char(30),
    address2 char(30),
    city char(20),
```

```
state char(2),
zip char(10),
country char(2),
terms_code char(6),
terms_approval char(6),
pay_method char(6),
payment char(1),
card_no char(20),
exp_date char(5),
card_holder char(20),
check_no char(8),
trd_ds_code char(6),
trd_ds_type char(1),
trd_ds_rate decimal(6),
multi_shipto char(1),
tax_rate decimal(6),
staging_area char(6),
fob_point char(15),
ship_via char(30),
ship_weight decimal(14),
item_amount decimal(14),
discountable decimal(12),
trd_ds_amount decimal(14),
taxable decimal(12),
tax_amount decimal(14),
frght_amount decimal(14),
total_amount decimal(14),
create_date date,
create_time char(8),
create_id char(8),
l_mod_date date,
l_mod_time char(8),
l_mod_id char(8),
system_order char(1),
spr_no char(7),
cust_ord_date date,
cust_po_date date,
fact_ack_date date,
fact_rec_date date,
moto_rec_date date,
sent_to_wwop char(1),
mtaxg_code char(6),
intl_order char(1),
intl_lic_no char(30),
currency_code char(3),
curr_rate_type char(6),
currency_rate decimal(16),
edi_sent char(1),
blo_exp_date date,
dpas_rating char(2),
resale_cust char(6),
resale_po char(30),
actual_frght_amt decimal(12),
```

```
    orig_frght_amt decimal(12),
    ship_terms char(15),
    residential_cust char(1),
    email char(50),
    ups_account char(10),
    mtax_freight char(6),
    auth_amt decimal(10,2),
    auth_code char(8),
    auth_date date,
    decline_code char(8),
    decline_message char(60),
    ship_complete char(1),
    contract_no char(20),
    multiple_orders char(1),
    deposit decimal(10,2),
    docs_sent date,
    destination char(30),
    consignee_name char(20),
    consignee_addr1 char(30),
    consignee_addr2 char(30),
    consignee_city char(20),
    consignee_state char(2),
    consignee_zip char(10),
    consignee_country char(20),
    notify_name char(20),
    notify_info char(240),
    truck_bol char(20),
    route_code char(10),
    resale_no char(15),
    resale_expiry date,
    rma_reason char(6),
    base_doc_no integer,
    rma_doc_no integer,
    order_description char(80),
    ready_to_invoice char(1),
    required_date date,
    default_rma_type char(3),
    restock_fee decimal(8,2),
    ship_to_name char(30),
    one_time_cust char(1),
    ship_type char(6),
    restock_percent decimal(6,2),
    restock_amount decimal(8,2),
    rma_status_code char(20),
    handling_fee decimal(8,4),
    phone char(12);

--------------------------------------------------------------------------
aroordrh - Archive
    doc_no integer,
    ship_type char(6),
    service_type char(50),
    package_type char(30),
```

```
    weight decimal(9,2),
    weight_unit char(5),
    cost_amount decimal(9,2),
    charge_amount decimal(9,2),
    estimate_date date,
    commit_days smallint

--------------------------------------------------------------------------------
aroordsd - Archive
    doc_no integer,
    line_no smallint,
    dock_receipt date,
    to_repair_ctr date,
    repair_complete date,
    to_customer date,
    turn_time smallint,
    sched_ship date,
    rep_rep_prt date,
    symptom char(40),
    serial_no char(20),
    last_chg_user char(8),
    last_chg_date date

--------------------------------------------------------------------------------
aroorwtd - Archive
    doc_no integer,
    line_no smallint,
    ship_no smallint,
    po_doc_no integer,
    po_line_no smallint,
    seq_no smallint,
    unit_no smallint,
    uom char(6),
    gross decimal(12),
    tare decimal(12),
    net decimal(12)

--------------------------------------------------------------------------------
aroshipd - Archive stoshipd
    doc_no integer not null ,
    line_no smallint not null ,
    ship_no smallint,
    kit_group smallint,
    stage char(3),
    ship_qty decimal(14),
    sell_unit char(2),
    ship_weight decimal(14),
    commit_qty decimal(14),
    pic_ticket_no smallint,
    mfs_printed char(1),
    inv_doc_no integer,
    sell_to_code char(20),
    ship_to_code char(6),
```

```
    bill_to_code char(20),
    item_code char(20),
    warehouse_code char(10),
    stock_location char(12),
    price_group char(6),
    price decimal(12),
    orig_price decimal(12),
    retail_price decimal(12),
    price_approval char(6),
    tax_rate decimal(6),
    tax_amount decimal(12),
    net_amount decimal(14),
    item_cost decimal(12),
    gross_margin decimal(10),
    new_date date,
    bko_date date,
    ord_date date,
    pic_date date,
    shp_date date,
    inv_date date,
    pst_date date,
    can_date date,
    proj_ship_date date,
    request_date date,
    reject_date date,
    actual_ship_date date,
    fact_sched_date date,
    ship_via_cd char(3),
    mtaxg_code char(6),
    po_doc_no integer,
    ship_lookup char(30)

--------------------------------------------------------------------------
aroshpfr - Archive stoshpfr
    pack_lookup_key char(40),
    pack_tracking_no char(40),
    pack_freight decimal(10,2),
    pack_shp_date char(10),
    pack_shp_boxes char(4),
    pack_void_flag char(1),
    pack_del_date date,
    pack_del_city char(20),
    pack_del_state char(2),
    pack_del_type char(20),
    pack_timestamp datetime year to second
        default current year to second,
    order_no char(20),
    shp_weight decimal(9,1),
    shp_cost decimal(10,2),
    ship_type char(6),
    package_type char(30),
    stage char(5),
    inv_doc_no integer
```

```
--------------------------------------------------------------------------
aroshtxd - Archive stoshtxd
    doc_no integer not null ,
    line_no smallint not null ,
    ship_no smallint,
    tax_code char(6),
    tax_rate decimal(6),
    net_amt decimal(12),
    tax_amt decimal(12)

--------------------------------------------------------------------------
arotrckd - Archive stotrckd
    contract_no char(20),
    doc_no integer,
    line_no smallint,
    po_doc_no integer,
    fwdr_invoice char(20),
    sales_basis char(5),
    loadg_location char(30),
    port_of_exit char(30),
    gross_weight decimal(10),
    net_weight decimal(10),
    tare_weight decimal(10),
    container_no char(20),
    bill_lading char(20),
    ocean_bol char(20),
    booking_no char(20),
    vessel char(25),
    voyage char(25),
    cutoff_date date,
    loadg_date date,
    release_date date,
    est_depart date,
    est_arrive date,
    pymt_due_date date,
    est_demurrage decimal(10,2),
    insurance decimal(10,2),
    relse_rqstd date,
    relse_recvd date,
    draft_obl_rcvd date,
    consignee_name char(20),
    consignee_addr1 char(30),
    consignee_addr2 char(30),
    consignee_city char(20),
    consignee_state char(2),
    consignee_zip char(10),
    consignee_country char(20),
    notify_name char(20),
    notify_info char(240),
    gross_mt decimal(8,2),
    net_mt decimal(8,2),
    tare_mt decimal(8,2),
```

```
    forwarder_name char(20),
    carrier char(20),
    send_docs char(20),
    container_size char(20),
    transhipment1 char(20),
    vessel1 char(20),
    voyage1 char(10),
    eta1 date,
    etd1 date,
    transhipment2 char(20),
    vessel2 char(20),
    voyage2 char(10),
    eta2 date,
    etd2 date,
    transhipment3 char(20),
    vessel3 char(20),
    voyage3 char(10),
    eta3 date,
    etd3 date,
    transhipment4 char(20),
    vessel4 char(20),
    voyage4 char(10),
    eta4 date,
    etd4 date,
    transhipment5 char(20),
    vessel5 char(20),
    voyage5 char(10),
    eta5 date,
    etd5 date,
    port1 char(10),
    port2 char(10),
    port3 char(10),
    port4 char(10),
    port5 char(10),
    seal_no char(10)

------------------------------------------------------------------------

stocompd --- FR2523 - display on report the commission reductions
   sls_psn_code  char(6),       - Sales Person
   past_due_days  smallint,     - Past Due Days
   comm_reduce_pct decimal(5,2) - Commission reduction percentage

------------------------------------------------------------------------
stofdxdt --- FR2425 - Fedex Cost Reconciliation
    bill_to_acct char(20),
    invoice_date date,
    invoice_no char(20),
    store_id char(20),
    orig_amt_due decimal(10,2),
    curr_balance decimal(10,2),
    payor char(15),
    grnd_trk_pfx char(10),
```

```
grnd_trk_id char(20),
trans_chg_amt decimal(10,2),
net_chg_amt decimal(10,2),
service_type char(30),
ground_service char(20),
shipment_date date,
pod_delvry_date date,
pod_delvry_time char(5),
pod_svc_area char(2),
pod_sign_desc char(20),
actual_weight decimal(10,2),
actual_weight_unit char(3),
rated_weight decimal(10,2),
rated_weight_unit char(3),
number_pieces integer,
bundle_number integer,
meter_number char(10),
recpnt_name char(50),
recpnt_company char(50),
recpnt_addr1 char(30),
recpnt_addr2 char(30),
recpnt_city char(30),
recpnt_state char(2),
recpnt_zip char(12),
recpnt_country char(30),
shippr_company char(50),
shippr_name char(50),
shippr_addr1 char(30),
shippr_addr2 char(30),
shippr_city char(30),
shippr_state char(2),
shippr_zip char(12),
shippr_country char(30),
cust_ref_1 char(30),
cust_ref_2 char(30),
cust_ref_3 char(30),
dept_ref char(30),
cust_ref_1u char(30),
cust_ref_2u char(30),
cust_ref_3u char(30),
dept_refu char(30),
rma_number char(20),
orig_rcpnt_addr1 char(30),
orig_rcpnt_addr2 char(30),
orig_rcpnt_city char(30),
orig_rcpnt_state char(2),
orig_rcpnt_zip char(12),
orig_rcpnt_country char(30),
zone_code char(5),
entry_date date,
entry_number integer,
customs_value decimal(10,2),
customs_val_cur char(10),
```

```
declared_value decimal(10,2),
declared_val_cur char(10),
commod_desc1 char(30),
commod_cntry1 char(10),
commod_desc2 char(30),
commod_cntry2 char(10),
commod_desc3 char(30),
commod_cntry3 char(10),
commod_desc4 char(30),
commod_cntry4 char(10),
cur_conv_date date,
cur_conv_rate decimal(12),
multi_no integer,
multi_units integer,
multi_wght decimal(10),
multi_ship_chg decimal(10,2),
multi_ship_wght decimal(10),
grnd_trk_ac_disc decimal(10,2),
grnd_trk_ac_gros decimal(10,2),
chg_desc_01 char(30),
chg_amt_01 decimal(10,2),
chg_desc_02 char(30),
chg_amt_02 decimal(10,2),
chg_desc_03 char(30),
chg_amt_03 decimal(10,2),
chg_desc_04 char(30),
chg_amt_04 decimal(10,2),
chg_desc_05 char(30),
chg_amt_05 decimal(10,2),
chg_desc_06 char(30),
chg_amt_06 decimal(10,2),
chg_desc_07 char(30),
chg_amt_07 decimal(10,2),
chg_desc_08 char(30),
chg_amt_08 decimal(10,2),
chg_desc_09 char(30),
chg_amt_09 decimal(10,2),
chg_desc_10 char(30),
chg_amt_10 decimal(10,2),
chg_desc_11 char(30),
chg_amt_11 decimal(10,2),
chg_desc_12 char(30),
chg_amt_12 decimal(10,2),
chg_desc_13 char(30),
chg_amt_13 decimal(10,2),
chg_desc_14 char(30),
chg_amt_14 decimal(10,2),
chg_desc_15 char(30),
chg_amt_15 decimal(10,2),
chg_desc_16 char(30),
chg_amt_16 decimal(10,2),
chg_desc_17 char(30),
chg_amt_17 decimal(10,2),
```

```
    chg_desc_18 char(30),
    chg_amt_18 decimal(10,2),
    chg_desc_19 char(30),
    chg_amt_19 decimal(10,2),
    chg_desc_20 char(30),
    chg_amt_20 decimal(10,2),
    chg_desc_21 char(30),
    chg_amt_21 decimal(10,2),
    chg_desc_22 char(30),
    chg_amt_22 decimal(10,2),
    chg_desc_23 char(30),
    chg_amt_23 decimal(10,2),
    chg_desc_24 char(30),
    chg_amt_24 decimal(10,2),
    chg_desc_25 char(30),
    chg_amt_25 decimal(10,2)

------------------------------------------------------------------------

stooedoc --- MR2879 - implement table stooedoc to control oe_doc_no calls.
  next_doc_no serial not null

------------------------------------------------------------------------
stoorrld -- FR3235 - Blanket SOs
    doc_no integer,          - doc number
    line_no smallint,        - line number
    ship_no smallint,        - ship number
    rlse_qty decimal(10,4),  - released qty
    balance decimal(10,4),   - balance
    ship_date date,          - ship date
    required_date date       - required date

------------------------------------------------------------------------

storlccd -- FR3323 - Customer Credit Management
    cust_code char(20),         - Customer
    rel char(1),                - Release
    doc_no integer,             - Document Number
    order_no char(20),          - Order Number
    order_date date,            - Order Date
    total_amount decimal(14),   - Total Amount
    hold_code char(6)           - Hold code

------------------------------------------------------------------------

storlcce -- FR3323 - Customer Credit Management
    cust_code char(20)          - Customer code

------------------------------------------------------------------------
storlsed -- FR3235 - Blanket SOs
    doc_no integer,           - Document Number
    line_no smallint,         - Line Number
    item_code char(20),       - Item code
```

```
    ordr_qty decimal(14),    - Order Qty
    rlse_qty decimal(14),    - Release Qty
    remain_qty decimal(14),  - Remain Qty
    qty_to_rlse decimal(14)  - Qty to release

--------------------------------------------------------------------------
storlsee -- FR3235 - Blanket SOs
    doc_no serial,           - Document Number
    so_doc_no integer,       - SO Document Number
    so_no char(20),          - SO Number
    rlse_date date,          - Release Date
    ship_date date,          - Ship Date
    required_date date       - Required Date
```