

Payroll Tables

=====

styaccrr - accrual code reference table

```
accr_code char(6) not null,      # accrual code
accr_desc char(30),              # description of accrual code
accr_method char(1),             # method (H hourly, P period)
accr_rate decimal(18,8),         # rate
accr_freq integer,              # frequency
accr_lapse integer               # lapse
```

create unique index ilyaccrr on styaccrr (accr_code);

styactvd - payroll activity detail table

```
orig_journal char(2) not null,   # original journal
doc_no integer not null,         # document number
act_code char(6),               # income/deduction/obligation
act_type char(1) not null,      # activity type
                                # A - payroll check
                                # B - income
                                # C - deduction
                                # D - employer obligation expense
                                # E - employer obligation liability
amount decimal(12),             # amount
number decimal(18,8),           # number
hours decimal(12),              # hours
rate decimal(18,8),             # rate
acct_no integer,                # ledger account number
dept_code char(3)               # department
```

create index ilyactvd on styactvd (doc_no,act_type,act_code);

create index i2yactvd on styactvd (act_type);

styactvv - payroll transaction view table

```
orig_journal      char(2)      # original journal
doc_no            integer      # document number
post_no           integer      # posting number
post_date         date         # post date
doc_date          date         # document date
ref_code          char(6)      # reference code
doc_desc          char(30)     # document description
check_no         char(12)     # check number
pay_date         date         # pay date
eop_date         date         # end of period date
act_code         char(6)      #
income/deduction/obligation
act_type         char(1)      # activity type
```

```

# A - payroll check
# B - income
# C - deduction
# D - employer obligation expense
# E - employer obligation liability
amount          decimal(12) # amount
hours           decimal(12) # hours

```

create view styactvv

```

(orig_journal, doc_no, post_no, post_date, doc_date, ref_code,
 doc_desc, check_no, pay_date, eop_date, act_code, act_type,
 amount, hours, rate, acct_no, dept_code)

```

as

```

select x1.orig_journal ,x1.doc_no ,x1.post_no ,x1.post_date
 ,x1.doc_date ,x1.ref_code ,x1.doc_desc ,x0.check_no ,x0.pay_date
 ,x0.eop_date ,x2.act_code ,x2.act_type ,x2.amount ,x2.hours
 ,x2.rate ,x2.acct_no ,x2.dept_code from stytranr x0
 ,stxtranr x1 ,styactvd x2 where (((x0.doc_no
 = x1.doc_no ) AND (x0.doc_no = x2.doc_no ) ) AND (x0.orig_journal
 = x1.orig_journal ) ) AND (x0.orig_journal = x2.orig_journal
 ) );

```

stycntrc - payroll control table

```

post_gl char(1), # post to general ledger flag
ein_number char(10), # employer identification number
state_number char(15), # state identification number
fedtax_code char(6), # federal tax deduction code
fica_code char(6), # fica deduction code
medicare_code char(6), # medicare deduction code
statax_code char(6), # state tax deduction code
loctax_code char(6), # local tax deduction code
futa_code char(6), # futa obligation code
fica_ob_code char(6), # fica obligation code
medicare_ob_code char(6), # medicare obligation code
eic_code char(6), # earned income credit code
exp_acct integer, # expense account
liab_acct integer, # liability account
cash_acct integer, # cash account
mmedia_file char(100), # media holding file
mmedia_command char(100), # media creation command
py_doc_no integer, # last payroll document number
py_post_no integer, # last payroll posting number
immed_dest_dfi integer, # dfi (direct deposit)
immed_chk_digit smallint, # company check digit
immed_dest_name char(23), # destination name
co_bank_acct_no char(17), # company bank account number
offset_debit char(1), # offset direct deposit debit?
set_up char(1), # setup complete flag(3.90 adv)
suta_code char(6), # suta obligation code(3.9 adv)

```

Following fields added for implementation for Garnishments and Taxes

```

fed_amt_code      char(6),      #Additional Federal Withholding
                  # Deduction Code for Flat Amount
fed_percent_code char(6),      #Additional Federal Withholding
                  #Deduction Code for percentage of Net
bns_percent       decimal(12), #Bonus only percentage of net pay
                  # for Federal Withholding Tax
bns_amt_code      char(6),      #Additional Federal Withholding
                  # Deduction Code flat amount for
                  # bonus only payroll to be used with
                  # bns_percent
bns_percent_code  char(6),      #Bonus only deduction code for
                  # Federal Withholding Tax as a
                  # percentage of net pay
levy_amt_code     char(6),      #Deduction code for IRS levy as a
                  # flat amount
levy_percent_code char(6),      #Deduction code for IRS levy as a
                  # percentage of net pay
chd_sup_amt_code  char(6),      #Deduction code for child support
                  # payments as a garnishment
garn_to_ap        char(1),      #Yes or No flag to post garnishments
                  # automatically to Accounts Payable

```

stydedcr - deduction code reference table

```

ded_code char(6) not null, # deduction type code
description char(30),     # description of deduction
ded_type char(1),         # (G, T, H, N, F, U) code which
                          # indicates what rate applies to
                          # (gross wages, taxable wages, hours
                          # worked, flat rate, fica wages,
                          # futa wages)
ded_taxred char(1),       # (A, B, C, D, F, N, T, U) code
                          # indicates what wage base is
                          # reduced by the deduction
dflt_rate decimal(18,8),  # rate
dflt_limit decimal(12),   # limit for the deduction
dflt_acct integer,        # liability account number
dflt_dept char(3),        # department
dflt_apply char(1),       # default code which indicates
                          # at what frequency to apply this
                          # type of deduction.
                          # M monthly, A always, Q quarterly,
                          # Y yearly
dflt_hi_ded_amt decimal(12), # high deduction exception amount
dflt_lo_ded_amt decimal(12), # low deduction exception amount
state_ein char(15),       # state identification number
tax_jur char(6),          # tax jurisdiction

```

Following fields added for implementation for Garnishments and Taxes

```

ded_class char(1),        #Deduction Class to indicate the type
                          #of deduction '1'=Taxes,
                          '2'=Garnishments,

```

```

                                # '3'=Levy, '4 ->'=other
rate_type      char(1)          #Type of rate for deduction, 'P'=rate,
                                #'C'=amount for net, 'L'=Flat amount with
                                #upper limits
ded_fee_amt    decimal(10)      #Default Fee Amount for Garnishments
ded_fee_code   char(6)          #Deduction Code for Garnishment Fee
threshold_amount decimal(8,2)  #Additional Medicare Tax

create unique index ilydedcr on stydedcr (ded_code);

```

styempdd - employee deduction detail table

```

empl_code char(6) not null,    # employee code to whom this row refers
ded_code char(6),              # reference code for deduction type
line_no serial,                # line number (for sorting purposes)
ded_rate decimal(18,8),        # rate
ded_limit decimal(12),         # annual limit for this deduction
ded_apply char(1),             # code which indicates at what
                                # frequency to apply this deduction
acct_no integer,               # liability account number
department char(3),            # department
ded_qtd1 decimal(12),          # first quarter to date accrued
ded_qtd2 decimal(12),          # second quarter to date accrued
ded_qtd3 decimal(12),          # third quarter to date accrued
ded_qtd4 decimal(12),          # fourth quarter to date accrued
ded_ytd decimal(12),           # year to date accrued
ded_date date,                 # date deduction was last taken
lo_ded_amt decimal(12),        # low deduction exception amount
hi_ded_amt decimal(12),        # high deduction exception amount

```

Following fields added for implementation for Garnishments and Taxes

```

garnishment char(1),          # Yes/No flag deduction as a garnishment
garn_cnt     smallint         # Value greater than 0 indicates that
                                # garnishment data exists in table
                                # styempgn. The value is the unique
                                # key to the table styempgn.
ded_fee_amt  decimal(10)      #Default Fee Amount for Garnishments
ded_fee_code char(6)          #Deduction Code for Garnishment Fee

```

```

create index ilyempdd on styempdd (empl_code,line_no);

```

styempid - employee income detail table

```

empl_code char(6) not null,    # employee code to whom this row refers
inc_code char(6),              # reference code for income type
line_no serial,                # line number (for sorting purposes)
inc_rate decimal(18,8),        # rate
inc_number decimal(18,8),      # number to apply to rate when
                                # calculating amount
inc_hours decimal(12),         # hours worked
acct_no integer,               # expense account

```

```

department char(3),           # department
inc_qtd1 decimal(12),        # first quarter to date accrued
inc_qtd2 decimal(12),        # second quarter to date accrued
inc_qtd3 decimal(12),        # third quarter to date accrued
inc_qtd4 decimal(12),        # fourth quarter to date accrued
inc_ytd decimal(12),         # year to date accrued
lo_inc_amt decimal(12),      # low income exception amount
hi_inc_amt decimal(12)       # high income exception amount

```

```
create index ilyempid on styempid (empl_code,line_no);
```

```
-----
styemplr - employee reference table
```

```

empl_code char(6) not null,   # employee id number
soc_sec_num char(11),         # social security number
type_code char(6),           # employee template code - template
                              # used to set up employee.
birthdate date,              # birthdate
first_name char(12),          # employee first name
middle_name char(12),         # employee middle name
last_name char(15),           # employee last name
address1 char(30),            # address 1
address2 char(30),            # address 2
city char(20),                # city
state char(2),                # state
zip char(10),                 # zip code
phone char(20),               # phone number
cash_acct integer,           # payroll cash account
department char(3),           # department
job_code char(6),             # job type reference code
job_title char(30),           # job title
date_hired date,              # date hired
terminated date,              # date terminated
empl_status char(1),          # full/part time status
pay_period char(1),           # pay period (weekly, bi-weekly, etc.)
allowances smallint,         # number of withholding allowances
state_allow smallint,        # number of state withholding
                              # allowances
marital_stat char(1),         # marital status (S, M)
vac_code char(6),             # vacation income code
vac_allowed decimal(12),      # vacation time allowed
vac_used decimal(12),         # vacation time used
sick_code char(6),            # sick income code
sick_allowed decimal(12),     # sick time allowed
sick_used decimal(12),        # sick time used for state tax
last_pay date,                # date last paid
hold_pymnt char(1),           # hold payment: a Y means that
                              # automatic payroll will skip this
                              # employee
statax_code char(6),          # state tax deduction code
loctax_code char(6),          # local tax deduction code
sick_accr_code char(6),       # sick accrual code

```

```

sick_accr_ctr integer,      # sick accrual counter
sick_lapse_date date,      # date the sick accrual began
                             # taking place
vac_accr_code char(6),     # vacation accrual code
vac_accr_ctr integer,      # vacation accrual counter
vac_lapse_date date,       # date the vacation accrual began
                             # taking place
dir_dept char(1),          # direct deposit, Y/N
dfi_dest integer,          # dfi destination
chk_digit smallint,       # check digit
bank_acct_no char(17),     # employee bank account number
state_udf decimal(18,8),   # flexible field for state tax
email char(50),
cell_phone char(20)
shift_em char(3),          # Labor Processing Shift
emp_team_em char(5),       # Labor Processing Team
emp_group_em char(5),      # Labor Processing Group
department_em char(3),     # Labor Processing Department
jobclass_em char(3),       # Labor Processing Job Class
machine_ovr_em char(1),    # Labor Processing Machine
cost_ctr_ovr_em char(1),   # Labor Processing Cost
department_ovr_em char(1), # Labor Processing Department
team_ovr_em char(1),       # Labor Processing Team
shift_ovr_em char(1),      # Labor Processing Shift Over
job_class_ovr_em char(1),  # Labor Processing Job Class Over
std_hour_ovr_em char(1),   # Labor Processing Standard hour Over
date_ovr_em char(1)        # Labor Processing Date Over

```

```

create index ilyemplr on styemplr (soc_sec_num);
create index i2yemplr on styemplr (last_name,first_name);
create index i3yemplr on styemplr (empl_code);

```

```

-----
styempod - employee obligation detail

```

```

empl_code char(6) not null, # employee code to whom this row refers
obl_code char(6),           # reference code for this obligation
line_no serial,             # line number (used for sorting)
obl_rate decimal(18,8),     # rate
obl_limit decimal(12),      # annual obligation limit
acct_no integer,           # payroll expense account
department char(3),         # department
bal_acct_no integer,        # balancing liability account
bal_dept char(3),           # balancing department
obl_qtd1 decimal(12),       # first quarter to date accrued
obl_qtd2 decimal(12),       # second quarter to date accrued
obl_qtd3 decimal(12),       # third quarter to date accrued
obl_qtd4 decimal(12),       # fourth quarter to date accrued
obl_ytd decimal(12)         # year to date accrued

```

```

create index ilyempod on styempod (empl_code,line_no);

```

styhistd - employee history detail table

```
doc_no integer not null,      # document number for history
line_no smallint,            # line number
hist_line char(60)           # line of text
```

```
create index ilyhistd on styhistd (doc_no,line_no);
```

styhiste - employee history header table

```
doc_no serial not null,      # document number for history
empl_code char(6),           # employee reference code
hist_date date               # history date
```

```
create index ilyhiste on styhiste (hist_date desc,doc_no);
```

styinccr - income code reference table

```
inc_code char(6) not null,   # income type code
description char(30),        # description
dflt_num decimal(18,8),      # default number
dflt_rate decimal(18,8),     # default rate
dflt_hours decimal(12),      # default hours
dflt_acct integer,          # default expense account
dflt_dept char(3),          # default department
inc_type char(1),           # income type
                             # A advance, B fica/futa exempt,
                             # E expense, F fica exempt, H hourly,
                             # N non-hourly, U futa exempt
                             # expense/advance types are not used
                             # in the calculation of deductions
dflt_lo_inc_amt decimal(12), # low income exemption amount
dflt_hi_inc_amt decimal(12), # high income exemption amount
non_qual char(1)            # non qualified
```

```
create unique index ilyinccr on styinccr (inc_code);
```

styoblcr - obligation code reference table

```
obl_code char(6) not null,   # deduction type code
description char(30),        # description
obl_type char(1),           # the wage base that the obligation
                             # rate is applied to, same as deduction
dflt_rate decimal(18,8),     # rate
dflt_limit decimal(12),      # annual obligation limit
dflt_acct integer,          # expense account
dflt_dept char(3),          # department
dflt_bacct integer,         # balancing liability account
dflt_bdept char(3)          # balancing department
```

```
create unique index ilyoblcr on styoblcr (obl_code);
```

```
-----  
stypaydd - payroll entry deduction detail table
```

```
doc_no integer not null,      # payroll document number  
line_no smallint,           # line number (for sorting purposes)  
ded_code char(6),           # deduction code  
ded_rate decimal(18,8),     # deduction rate (if applicable)  
amount decimal(12),         # deduction amount  
acct_no integer,            # liability account  
department char(3),         # department  
mod_flag smallint,         # flag to mark manual modification  
add_code char(1),          # flag to mark code as new to employee  
lo_ded_amt decimal(12),     # low deduction exemption amount  
hi_ded_amt decimal(12)     # high deduction exemption amount
```

```
create index ilypaydd on stypaydd (doc_no,line_no);
```

```
-----  
stypayid - payroll entry income detail table
```

```
doc_no integer not null,      # payroll document number  
line_no smallint,           # line number (for sorting purposes)  
inc_code char(6),           # income code  
inc_rate decimal(18,8),     # income rate  
number decimal(18,8),       # income number  
                                # (used to calculate amount)  
hours decimal(12),          # hours associated with this income  
amount decimal(12),         # amount  
acct_no integer,            # expense account  
department char(3),         # department  
mod_flag smallint,         # flag to mark manual modification  
add_code char(1),          # flag to mark code as new to employee  
lo_inc_amt decimal(12),     # low income exemption amount  
hi_inc_amt decimal(12)     # high income exemption amount
```

```
create index ilypayid on stypayid (doc_no,line_no);
```

```
-----  
stypayod - payroll entry obligation detail table
```

```
doc_no integer not null,      # payroll document number  
line_no smallint,           # line number (for sorting purposes)  
obl_code char(6),           # obligation code  
obl_rate decimal(18,8),     # obligation rate  
amount decimal(12),         # obligation amount  
acct_no integer,            # expense account  
department char(3),         # department  
bal_acct_no integer,        # balancing liability account  
bal_dept char(3),          # department  
mod_flag smallint,         # flag to mark manual modification
```



```

    add_code char(1)          # flag to mark code as new to employee

create index ilypayod on stypayod (doc_no,line_no);

```

stypayre - payroll entry reference table

```

doc_no serial not null,      # payroll document number
empl_code char(6),          # employee code
doc_date date,              # date document was created
pay_date date,              # payroll date
eop_date date,              # end of period date
print_check char(1),        # Y/N: N if not to print check
                             # or if check has already printed
cash_acct_no integer,       # payroll cash account
department char(3),         # department
cash_amount decimal(12),    # amount of check
check_no integer,           # check number
inc_gross decimal(12),      # gross income amount
ded_fica decimal(12),       # fica deduction amount
inc_taxable decimal(12),    # taxable income amount
ded_medicare decimal(12),   # medicare deduction amount
ded_fedtax decimal(12),     # federal tax deduction amount
ded_statax decimal(12),    # state tax deduction amount
ded_loctax decimal(12),    # local tax deduction amount
ded_other decimal(12),     # amount of all other deductions
obl_futa decimal(12),       # futa obligation amount
obl_fica decimal(12),       # fica obligation amount
obl_medicare decimal(12),   # medicare obligation amount
obl_other decimal(12),     # amount of all other obligations
obl_total decimal(12),     # total amount of obligations
inc_net decimal(12),        # net income amount
inc_expense decimal(12),    # expense income amount
total_hours decimal(12),    # hours worked
ok_to_post char(1),         # flag to mark document ready to post
accrue_sick char(1),        # Y/N to accrue sick time
accrue_vac char(1),         # Y/N to accrue vacation time
bonus char(1),              # Y/N bonus check
deposit char(1),            # Y/N direct deposit check
orig_journal char(2),
trans_doc_no integer,
posted char(1)

```

```

create unique index ilypayre on stypayre (empl_code,doc_no);
create index i2ypayre on stypayre (doc_no);
create index i3stypayre on stypayre(orig_journal, trans_doc_no);

```

stypdatd - holiday date detail table (used for direct deposit)

```

dd_year char(4),            # year of detail row, relates to header
dd_date date,               # holiday date

```

```

        dd_desc char(30),          # description of holiday
create unique index ilypdatsd on stypdatd (dd_date);

-----
stypdate - holiday date header table (used for direct deposit)

        dd_year char(4)          # year, relates to detail
create unique index ilypdate on stypdate (dd_year);

-----
stypddrd - direct deposit detail table

        doc_no serial not null,   # document number, relates to header
                                   # identifier of batch
        empl_code char(6),        # employee code
        dfi_dest integer,        # employee's bank routing number
        chk_digit smallint,      # employee's bank account check digit
        pay_date date,           # payroll date
        trans_code smallint,     # code thant tells bank what kind
                                   # of entry this is.
                                   # 23 prenotification debit to checking
                                   # 22 - regular debit to checking
        bank_acct_no char(17),    # employee's bank account number
        amount decimal(10),      # amount of entry
        empl_name char(22),      # employee name
        trace_number decimal(15,0) # concatenation of immediate dest dfi
                                   # and a sequential integer. Uniquely
                                   # identifes an entry on a tape.

-----
stypddre - direct deposit header table

        doc_no serial not null,   # document number, relates to detail
                                   # identifier of batch
        company_name char(16),    # company name
        entry_desc char(10),     # Payroll (hardcoded)
        dfi_immed integer,       # company's bank (where disk is sent)
        svc_class smallint,      # 3-digit code defining the entries as:
                                   # 200 - all credits
                                   # 220 - mixed debit/credit
                                   # if offset_debit is "Y" (meaning that
                                   # company debits its cash account to
                                   # pay for transaction), we use 220.
        batch_no integer,        # sequential numbering of unused batch
        batch_date date,         # date batch created
        create_date date,       # date tape created. left null until
                                   # user runs o_magmed
        file_id char(1),         # value identifies this file from
                                   # others if user sends more than one
                                   # tape in one day. Up to 64 tapes can
                                   # be sent per day. Value is stamped

```

```

used char(1)          # when o_automag.4gs is run.
                    # marked Y when o_automag is run

```

stystwahr - state tax withholding table

```

state_code char(6),      # state tax deduction code
wage_base char(1),      # wage base
ann_seq char(2),        # sequence for annualization
inc_lim char(1),        # basis for the income limit
                        # A based on filing status
                        # B based on allowances
                        # C based on both
inc_lim_seq char(2),    # sequence for income limit
inc_lim_ex_mult char(1), # income limit multiplier
inc_lim_marr decimal(12), # income limit for married 0/1 allow.
inc_lim_marr2 decimal(12), # income limit for married 2 allow.
inc_lim_sing decimal(12), # income limit for single person
inc_lim_hh decimal(12), # income limit for head of household
inc_lim_other decimal(12), # income limit for others
inc_lim_0 decimal(12), # income limit for 0 allowances
inc_lim_1 decimal(12), # income limit for 1 allowance
inc_lim_2 decimal(12), # income limit for 2 allowance
inc_lim_x decimal(12), # income limit for more allowance
std_code char(1),      # basis for standard deduction
                        # A based on filing status
                        # B based on allowances
                        # C based on both
std_seq char(2),      # sequence for standard deduction
std_ex_mult char(1), # standard deduction multiplier
std_marr decimal(12), # std deduction for married 0/1 allow.
std_marr2 decimal(12), # std deduction for married 2 allow.
std_sing decimal(12), # std deduction for single person
std_hh decimal(12), # std deduction for head of household
std_other decimal(12), # std deduction for others
std_0 decimal(12), # std deduction for 0 allowances
std_1 decimal(12), # std deduction for 1 allowances
std_2 decimal(12), # std deduction for 2 allowances
std_x decimal(12), # std deduction for more allowances
std_l char(1),      # basis for standard deduction limit
                        # A based on filing status
                        # B based on allowances
                        # C based on both
std_l_mult char(1), # std ded limit multiplier
std_l_sing decimal(12), # std ded limit for single person
std_l_marr decimal(12), # std ded limit for married 0/1 allow
std_l_marr2 decimal(12), # std ded limit for married 2 allow
std_l_hh decimal(12), # std ded limit head of household
std_l_other decimal(12), # std ded limit for others
std_l_0 decimal(12), # std ded limit for 0 allowances
std_l_1 decimal(12), # std ded limit for 1 allowances
std_l_2 decimal(12), # std ded limit for 2 allowances
std_l_x decimal(12), # std ded limit for more allowances

```

```

prs_code char(1),          # basis for personal exemption
                           # A based on filing status
                           # B based on allowances
                           # C based on both
prs_seq char(2),          # sequence for personal exemption
prs_ex_mult char(1),      # prs exmp multiplier
prs_marr decimal(12),     # prs exmp married 0/1 allowances
prs_marr2 decimal(12),   # prs exmp married 2 allowances
prs_sing decimal(12),     # prs exmp single
prs_hh decimal(12),       # prs exmp head of household
prs_other decimal(12),   # prs exmp others
prs_0 decimal(12),        # prs exmp 0 allowances
prs_1 decimal(12),        # prs exmp 1 allowance
prs_2 decimal(12),        # prs exmp 2 allowances
prs_x decimal(12),        # prs exmp more allowances
prs_l char(1),            # basis for prs exmp limit
prs_l_mult char(1),       # prx exmp limit multiplier
prs_l_sing decimal(12),   # prx exmp limit single
prs_l_marr decimal(12),   # prx exmp limit married 0/1 allow
prs_l_marr2 decimal(12),  # prx exmp limit married 2 allow
prs_l_hh decimal(12),     # prx exmp limit head of household
prs_l_other decimal(12),  # prx exmp limit others
prs_l_0 decimal(12),      # prx exmp limit 0 allowances
prs_l_1 decimal(12),      # prx exmp limit 1 allowance
prs_l_2 decimal(12),      # prx exmp limit 2 allowances
prs_l_x decimal(12),      # prx exmp limit more allowances
fed_calc_seq char(2),     # sequence for federal tax calculation
fed_red_seq char(2),      # sequence to reduce by fedtax
dep_code char(1),         # basis for dependent deduction
                           # A based on filing status
                           # B based on allowances
                           # C based on both
dep_seq char(2),          # sequence for dependent deduction
dep_ex_mult char(1),      # dependent deduction multiplier
dep_marr decimal(12),     # dependent for married 0/1 allow
dep_marr2 decimal(12),   # dependent for married 2 allow
dep_sing decimal(12),     # dependent for single
dep_hh decimal(12),       # dependent for head of household
dep_other decimal(12),   # dependent for others
dep_0 decimal(12),        # depended ded for 0 allowances
dep_1 decimal(12),        # depended ded for 1 allowances
dep_2 decimal(12),        # depended ded for 2 allowances
dep_x decimal(12),        # depended ded for more allowances
dep_l char(1),            # basis for dependent limit
                           # A based on filing status
                           # B based on allowances
                           # C based on both
dep_l_mult char(1),       # dependent limit multiplier
dep_l_sing decimal(12),   # dependent limit for single
dep_l_marr decimal(12),   # dependent limit married 0/1 allow
dep_l_marr2 decimal(12),  # dependent limit married 2 allow
dep_l_hh decimal(12),     # dependent limit head of household
dep_l_other decimal(12),  # dependent limit others

```

```

dep_l_0 decimal(12), # dependent limit 0 allowances
dep_l_1 decimal(12), # dependent limit 1 allowances
dep_l_2 decimal(12), # dependent limit 2 allowances
dep_l_x decimal(12), # dependent limit more allowances
tc_code char(1), # basis for tax credit
# A based on filing status
# B based on allowances
# C based on both
tc_seq char(2), # sequence for tax credit
tc_ex_mult char(1), # tax credit multiplier
tc_marr decimal(12), # tax credit married 0/1 allow
tc_marr2 decimal(12), # tax credit married 2 allow
tc_sing decimal(12), # tax credit single
tc_hh decimal(12), # tax credit head of household
tc_other decimal(12), # tax credit others
tc_0 decimal(12), # tax credit 0 allowances
tc_1 decimal(12), # tax credit 1 allowances
tc_2 decimal(12), # tax credit 2 allowances
tc_x decimal(12), # tax credit more allowances
tc_l char(1), # basis for tax credit limit
# A based on filing status
# B based on allowances
# C based on both
tc_l_mult char(1), # tax credit limit multiplier
tc_l_sing decimal(12), # tax credit limit single
tc_l_marr decimal(12), # tax credit limit married 0/1 allow
tc_l_marr2 decimal(12), # tax credit limit married 2 allow
tc_l_hh decimal(12), # tax credit limit head of household
tc_l_other decimal(12), # tax credit limit others
tc_l_0 decimal(12), # tax credit limit 0 allowances
tc_l_1 decimal(12), # tax credit limit 1 allowances
tc_l_2 decimal(12), # tax credit limit 2 allowances
tc_l_x decimal(12), # tax credit limit more allowances
st_tax_seq char(2), # sequence for state tax
fed_tax_seq char(2), # sequence for custom rate/federal tax
fed_tax_rate decimal(12), # custom rate to apply to federal tax
adjust_seq char(2), # sequence for de-annualizing
nsq_marr decimal(12), # non-sequenced married 0/1 allow
nsq_marr2 decimal(12), # non-sequenced married 2 allow
nsq_sing decimal(12), # non-sequenced single
nsq_hh decimal(12), # non-sequenced head of household
nsq_other decimal(12), # non-sequenced others
nsq_0 decimal(12), # non-sequenced 0 allowances
nsq_1 decimal(12), # non-sequenced 1 allowance
nsq_2 decimal(12), # non-sequenced 2 allowances
nsq_x decimal(12), # non-sequenced more allowances
nsq_l_sing decimal(12), # non-sequenced limit single
nsq_l_marr decimal(12), # non-sequenced limit married 0/1 allow
nsq_l_marr2 decimal(12), # non-sequenced limit married 2 allow
nsq_l_hh decimal(12), # non-sequenced limit head of household
nsq_l_other decimal(12), # non-sequenced limit others
nsq_l_0 decimal(12), # non-sequenced limit 0 allowances
nsq_l_1 decimal(12), # non-sequenced limit 1 allowance

```

```
nsq_l_2 decimal(12),      # non-sequenced limit 2 allowances
nsq_l_x decimal(12)      # non-sequenced limit more allowances
```

stytaxtd - tax table detail

```
tax_year char(4),        # tax year
ded_code char(6) not null, # tax table reference code
                           # (matches a deduction code)
pay_period char(1),      # code for type of pay period
marital_stat char(1),    # marital status (S/M)
over_amt decimal(12),    # over amount
base_amt decimal(12),    # base amount
tax_rate decimal(18,8),  # rate (enter 25% as .25)
order_no smallint       # ordering number associated with
                           # pay period code
```

```
create index ilytaxtd on stytaxtd
(marital_stat desc,ded_code,order_no desc);
```

stytaxtr - tax table header

```
tax_year char(4),        # tax year
ded_code char(6) not null, # tax table reference code
                           # (matches a deduction code)
week_allow decimal(12),   # weekly amount per allowance
biweek_allow decimal(12), # bi-weekly amount per allowance
smmonth_allow decimal(12), # semi-monthly amount per allowance
month_allow decimal(12),  # monthly amount per allowance
quarter_allow decimal(12), # quarterly amount per allowance
syear_allow decimal(12),  # semi-annual amount per allowance
year_allow decimal(12),   # annual amount per allowance
misc_allow decimal(12)    # daily/misc amount per allowance
```

stytimed - time card detail table

```
card_no integer not null, # time card id number
line_no smallint,         # line number (used for sorting)
inc_code char(6),         # income reference code
inc_rate decimal(18,8),   # rate
inc_number decimal(18,8), # number (used to calculate amount)
inc_hours decimal(12)     # hours
```

```
create index ilytimed on stytimed (card_no,line_no);
```

stytimee - time card header table

```
card_no serial not null, # time card id number
empl_code char(6),       # employee reference code
empl_name char(30),      # employee name
```

```

        start_date date,           # starting date for timecard
        end_date date,           # ending date for timecard
        used_flag char(1)       # flag to archive timecard

create index ilytimee on stytimee (empl_code,start_date,card_no);
create index i2ytimee on stytimee (card_no);

-----
stytranr - payroll transaction table

        orig_journal char(2) not null, # original journal
        doc_no integer not null,       # document number
        check_no char(12),            # check number
        pay_date date,                # payroll date
        eop_date date                 # end of period date

create index ilytranr on stytranr (pay_date,doc_no);
create index i2ytranr on stytranr (doc_no);

-----
stytranv - payroll transaction view table

        orig_journal      char(2)      # original journal
        doc_no            integer      # document number
        post_no           integer      # posting number
        post_date         date         # post date
        doc_date          date         # document date
        ref_code          char(6)      # reference code
        doc_desc          char(30)     # document description
        check_no          char(12)     # check number
        pay_date          date         # pay date
        eop_date          date         # end of period date

create view stytranv
( orig_journal, doc_no, post_no, post_date, doc_date, ref_code,
  doc_desc, check_no, pay_date, eop_date )
as
select x1.orig_journal ,x1.doc_no ,x1.post_no ,x1.post_date
,x1.doc_date ,x1.ref_code ,x1.doc_desc ,x0.check_no ,x0.pay_date
,x0.eop_date from stytranr x0 ,stxtranr x1
where ((x0.doc_no = x1.doc_no ) AND (x0.orig_journal = x1.
orig_journal ) ) ;

-----
stytypdd - employee type deduction detail table

        type_code char(6) not null, # template code to whom this row refers
        ded_code char(6),           # reference code for this deduction
        line_no smallint,           # line number (used for sorting)
        ded_rate decimal(18,8),     # rate
        ded_limit decimal(12),     # annual limit
        ded_apply char(1),          # code which indicates the
                                     # frequency to apply the deduction

```

```

    acct_no integer,           # liability account number
    department char(3),       # department
    lo_ded_amt decimal(12),    # low deduction exception amount
    hi_ded_amt decimal(12)    # high deduction exception amount

```

```
create index ilytypdd on stytypdd (type_code,line_no);
```

```
-----
stytyper - employee type reference table
```

```

    type_code char(6) not null, # template code
    description char(30),       # type description
    cash_acct integer,         # payroll cash account
    department char(3),        # department
    empl_status char(1),       # full/part-time status
    pay_period char(1),        # pay period (weekly/bi-weekly,etc)
    vac_code char(6),          # vacation income code
    vac_allowed decimal(12),    # vacation time allowed
    sick_code char(6),         # sick income code
    sick_allowed decimal(12),  # sick time allowed
    hold_pymnt char(1),        # hold payment: a "Y" in this column
                                # tells the automatic payroll generation
                                # program to skip this employee
    statax_code char(6),       # state tax deduction code
    loctax_code char(6),       # local tax deduction code
    sick_accr_code char(6),    # sick time accrual code
    vac_accr_code char(6)     # vacation time accrual code

```

```
create unique index ilytyper on stytyper (type_code);
```

```
-----
stytypid - employee type income detail table
```

```

    type_code char(6) not null, # template code to whom this row refers
    inc_code char(6),           # reference code for this income
    line_no smallint,          # line number (used for sorting)
    inc_rate decimal(18,8),     # rate (applied to number to calculate
                                # amount)
    inc_number decimal(18,8),   # number (applied to rate)
    inc_hours decimal(12),     # number of hours
    acct_no integer,           # payroll expense account
    department char(3),        # department
    lo_inc_amt decimal(12),    # low income exception amount
    hi_inc_amt decimal(12)    # hi income exception amount

```

```
create index ilytypid on stytypid (type_code,line_no);
```

```
-----
stytypod - employee type obligation detail table
```

```

    type_code char(6) not null, # template code to whom this row refers
    obl_code char(6),           # reference code for this income
    line_no smallint,          # line number (used for sorting)

```



```

obl_rate decimal(18,8),      # rate
obl_limit decimal(12),      # annual limit
acct_no integer,            # payroll expense account
department char(3),         # department
bal_acct_no integer,        # balancing account number
bal_dept char(3)            # balancing department

```

```
create index ilytypod on stytypod (type_code,line_no);
```

```
-----
styvoide - payroll void table
```

```

pay_doc_no integer          # payroll document number to void
void_date date              # Voided Date
sick_accr char(1)           # Reset sick accruals?
vac_accr char(1)            # Reset vacation accruals?

```

```
-----
stywtwoe - contains w2 information
```

```

empl_code char(6),          # employee reference code
control_no char(7),         # control number
state_number char(15),     # state number
statutory char(1),         # statutory box
deceased char(1),          # deceased box
pension char(1),           # pension plan box
legal_rep char(1),         # legal representative box
type_942 char(1),          # 942 employee box
subtotal char(1),          # subtotal box
deferred char(1),          # deferred compensation box
void_form char(1),         # void form box
alloc_tips decimal(12),    # allocated tips income
advanc_eic decimal(12),    # advance/eic income amounts
fedtax decimal(12),        # federal tax withheld
wages decimal(12),         # taxable wages
fica decimal(12),          # fica withheld
fica_wages decimal(12),    # fica wages
fica_tips decimal(12),     # fica tips
lbl_1miscbox char(14),     # label for miscellaneous box 1
val_1miscbox decimal(12),  # value for miscellaneous box 1
lbl_2miscbox char(14),     # label for miscellaneous box 2
val_2miscbox decimal(12),  # value for miscellaneous box 2
statax decimal(12),        # state tax withheld
state_wages decimal(12),   # state taxable wages
state_name char(9),        # state name
                             # (taken from stydedcr.tax_jur)
loctax decimal(12),        # local tax withheld
local_wages decimal(12),   # local wages
local_name char(9),        # local name
                             # (taken from stydedcr.tax_jur)
nonqual_plans decimal(12), # non-qualified plans
dep_care_ben decimal(12),  # dependent care benefits
fringe_ben decimal(12),    # fringe benefits

```

```

medicare decimal(12),          # medicare withheld
medicare_wages decimal(12),   # medicare wages
sick_pay char(1),
code_12a char(1),
amount_12a decimal(12),
code_12b char(1),
amount_12b decimal(12),
code_12c char(1),
amount_12c decimal(12),
code_12d char(1),
amount_12d decimal(12)

```

Following Table added for implementation for Garnishments and Taxes

styempgn - Garnishment Data for reporting and link to A/P

```

empl_code   char(6),          #Employee number from styempdd.empl_code
ded_code    char(6),          #Deduction code from styempdd.ded_code
garn_no     integer not null #Garnishment number, unique from
                                     # styempdd.garn_cnt
begin_date  date,            #Effective date of garnishment
garn_active char(1),         #Garnishment Active (Y/N)
rec_date    date,            #Document received date
garn_cat    char(2),         #Garnishment category,
                                     # user defined for reporting
doc_origin  char(30),        #Document origin
doc_ident1  char(40),        #Document identification,
                                     # court order docket number
doc_ident2  char(40),        #Document identification,
                                     # court order case number
plain_name  char(40),        #Plaintiff name of court order
plain_id    char(30),        #Plaintiff Identification of court order
vend_code   char(20),        #Payee Vendor code if exists
vend_name   char(40),        #Payee Name
vend_addr   char(30),        #Payee mailing address
vend_city   char(20),        #Payee mailing city
vend_state  char(2),         #Payee mailing state abbreviation
vend_zip    char(10),        #Payee mailing zip code
cr_date     datetime year to second, #Date and Time when data create
cr_user     char(15),        #User Name when data created
ch_date     datetime year to second, #Date and time of last data
change
ch_user     char(15)         #User name of last data change

```

```

create index ilyempgn(empl_code,ded_code,garn_no) using btree ;

```