// editAccount

// Function: Update USER table with new password and email for the current user.

// Required values: password(New password), email (New email), userName (current logged in user's ID).

// Output: Nothing

// createUserCheckDupe

// function: Select count(\*) from USER table to check if a user ID already exists.

// Required values: user\_id(The new user ID)

// Output: COUNT(\*) returns as 0 if current user\_id does not exist, > 0 if it does.

// createUser

// function: Insert a new user into USER table.

// Required values: user\_id(new user ID), password(user's password), email(user's email address), employee\_id(user's employee ID, if he has one), role(user's role), supplier\_id(user's associated supplier ID, if he has one)

// Output: None.

// viewUsers

// function: select count(\*) from USER table for calculating number of page tabs required.

// Required values: none.

// Output: COUNT(\*)

// viewUsersPage

// Function: Select from USER table a list of users for the current page tab.

// Required values: pagenum(page tab number)

// Output: User Login(USER\_ID), Email Address(EMAIL), Employee ID(EMPLOYEE\_ID), Role(ROLE), Associated Supplier(NAME)

// editUser

// function: Update USER table with new user information for row USER\_ID.

// required values: user\_id, email, employee\_id, role, old\_user\_id (original user ID)

// Output: None.

// deleteUser

// Function: Delete from USER table the selected user.

// Required values: user\_id

// Output: None.

// createItem

// function: Insert a new item into ITEM table.

// Required values: item\_name, receipt\_name, category, unit, item\_type, comment, supplier\_id

// Output: None.

// viewItems

// function: select count(\*) from ITEM table for calculating number of page tabs required.

// Required values: none.

// Output: COUNT(\*)

// viewItemsPage

// Function: Select from ITEM table a list of items for the current page tab.

// Required values: pagenum(page tab number)

// Output: Item ID(ITEM\_ID), Distribution Code(DIST\_CODE), Item Name(ITEM\_NAME), Receipt Name(RECEIPT\_NAME), Item Category(CATEGORY), Sales Unit(UNIT), Department(ITEM\_TYPE), Comments(COMMENT), Latest Price(PRICE), Supplier(NAME)

// editItem

// function: Update ITEM table with new item information for row ITEM\_ID.

// required values: dist\_code, item\_name, receipt\_name, category, unit, item\_type, comment, supplier\_id, item\_id

// Output: None.

// deleteItem

// Function: Delete from ITEM table the selected item.

// Required values: item\_id

// Output: None.

// createPrice

// function: Insert a new price into priceHistory table, and then update the associated item in ITEM table with new price ID.

// Required values: item\_id, price(new price), userName(current logged in user ID)

// Output: None.

// viewPrice

// Function: Select from PRICE\_HISTORY table the latest 20 price entries for current item.

// Required values: item\_id

// Output: PRICE, LOG\_DATE

// createSupplierCheckDupe

// function: Select count(\*) from SUPPLIER table by name to check if supplier already exists.

// Required values: name(new supplier's name)

// Output: COUNT(\*) returns as 0 if current user\_id does not exist, > 0 if it does.

// createSupplier

// function: Insert a new supplier into SUPPLIER table.

// Required values: name(new supplier's name), legal\_name, lead\_time, supplier\_comment, special\_comment

// Output: None.

// viewSuppliers

// function: select count(\*) from SUPPLIER table for calculating number of page tabs required.

// Required values: none.

// Output: COUNT(\*)

// viewSuppliersPage

// Function: Select from SUPPLIER table a list of suppliers for the current page tab.

// Required values: pagenum(page tab number)

// Output: Supplier Name(NAME), Legal Name(LEGAL\_NAME), Lead Time(LEAD\_TIME), Supplier Comments(SUPPLIER\_COMMENT), Special Comments(SPECIAL\_COMMENT)

// editSupplier

// function: Update SUPPLIER table with new supplier information for row SUPPLIER\_ID.

// required values: name(supplier's name), legal\_name, lead\_time, supplier\_comment, special\_comment, supplier\_id

// Output: None.

// deleteSupplier

// Function: Delete from SUPPLIER table the selected supplier.

// Required values: supplier\_id

// Output: None.

// createContactPerson

// function: Insert a new contact person into CONTACT\_PERSON table.

// Required values: supplier\_id, last\_name, first\_name, phone\_number, email

// Output: None.

// viewContactPerson

// Function: Select from CONTACT\_PERSON table a list of contact persons for the current supplier.

// Required values: supplier\_id

// Output: Last Name(LAST\_NAME), First Name(FIRST\_NAME), Phone Number(PHONE\_NUMBER), Email Address(EMAIL)

// editContactPerson

// function: Update CONTACT\_PERSON table with new contact person information for row CONTACT\_PERSON\_ID.

// required values: last\_name, first\_name, phone\_number, email, contact\_person\_id

// Output: None.

// deleteContactPerson

// Function: Delete from CONTACT\_PERSON table the selected contact person.

// Required values: contact\_person\_id

// Output: None.

// createSupplierAddress

// function: Insert a new supplier address into SUPPLIER\_ADDRESS table.

// Required values: supplier\_id, address\_line\_1, address\_line\_2, city, prov\_state, country, postal\_zip, phone\_number

// Output: None.

// viewSupplierAddress

// Function: Select from SUPPLIER\_ADDRESS table a list of addresses for the current supplier.

// Required values: supplier\_id

// Output: ADDRESS\_ID, Address Line 1(ADDRESS\_LINE\_1), Address\_Line\_2(ADDRESS\_LINE\_2), City(CITY), Province/State(PROV\_STATE), Country(COUNTRY), Postal/Zip Code(POSTAL\_ZIP), Phone Number(PHONE\_NUMBER)

// editSupplierAddress

// function: Update SUPPLIER\_ADDRESS table with new address information for row ADDRESS\_ID.

// required values: address\_line\_1, address\_line\_2, city, prov\_state, country, postal\_zip, phone\_number, address\_id

// Output: None.

// deleteSupplierAddress

// Function: Delete from SUPPLIER\_ADDRESS table the selected supplier.

// Required values: address\_id

// Output: None.

// createPurchaseOrder

// function: Insert a new purchase order into PURCHASE\_ORDER table with 'Queued' status.

// Required values: delivery\_date, deliver\_time, ref\_number, comment

// Output: None.

// viewPurchaseOrders

// function: select count(\*) from PURCHASE\_ORDER table for calculating number of page tabs required.

// Required values: none.

// Output: COUNT(\*)

// viewPurchaseOrdersPage

// Function: Select from PURCHASE\_ORDER table a list of purchase orders for the current page tab of a specific status.

// Required values: status, pagenum(current page tab number)

// Output: PO Number(PO\_ID), Current Status(STATUS), Date Created(CREATE\_DATE), Date Submitted(SUBMIT\_DATE), Delivery Date(DELIVERY\_DATE), Delivery Time(DELIVER\_TIME), Date Received(RECEIVE\_DATE), Reference Number(REF\_NUMBER), Comments(COMMENT), Supplier(NAME)

// editPurchaseOrder

// function: Update PURCHASE\_ORDER table with new purchase order information for row PO\_ID (Leaving status as is).

// required values: delivery\_date, delivery\_time, ref\_number, comment, supplier\_id, po\_id

// Output: None.

// submitPurchaseOrder

// function: Update PURCHASE\_ORDER table with 'Submitted' status for selected PO ID.

// required values: po\_id

// Output: None.

// cancelPurchaseOrder

// function: Update PURCHASE\_ORDER table with 'Cancelled' status for selected PO ID.

// required values: po\_id

// Output: None.

// returnPurchaseOrder

// function: Update PURCHASE\_ORDER table with 'Returned' status for selected PO ID.

// required values: po\_id

// Output: None.

// receivePurchaseOrder

// function: Update PURCHASE\_ORDER table with 'Received' status for selected PO ID.

// required values: po\_id

// Output: None.

// createOrderLine

// function: Insert a new order line into PO\_LINE table.

// Required values: po\_id, po\_line\_id, item\_id, qty\_ordered, comment, userName(current logged in user ID), price\_id

// Output: None.

// viewOrderLines

// Function: Select from ORDER\_LINE table a list of order lines for the current po.

// Required values: po\_id

// Output: Line(PO\_LINE\_ID), Item Name(ITEM\_NAME), Qty Ordered(QTY\_ORDERED), qty Received(QTY\_RECEIVED), Comments(COMMENT), Created By(AUTHOR), Latest Price(PRICE)

// editOrderLine

// function: Update PO\_LINE table with new Order line information for row PO\_LINE\_ID.

// required values: item\_id, qty\_ordered, qty\_received, comment, userName (current logged in user ID), price\_id, po\_id, po\_line\_id

// Output: None.

// deleteOrderLine

// Function: Delete from PO\_LINE table the selected order line.

// Required values: po\_id, po\_line\_id

// Output: None.

// createReturnLine

// function: Insert a new return line into RETURN\_LINE table.

// Required values: po\_id, return\_line\_id, po\_line\_id, return\_date, qty\_returned, credit\_memo\_num, comment, userName(current logged in user ID)

// Output: None.

// getSupplierList

// Function: select from SUPPLIER table supplier's name and supplier ID.

// Required values: None.

// Output: SUPPLIER\_ID, NAME

// getCategoryList

// Function: select from CATEGORY table all the categories.

// Required values: None.

// Output: CAT\_NAME

// getItemList

// Function: select from ITEM table item name and item ID for a specific supplier.

// Required values: None.

// Output: ITEM\_NAME, ITEM\_ID

// getUser

// Function: select from USER table all information of a single user.

// Required values: userName(current logged in user ID).

// Output: USER\_ID, PASSWORD, EMAIL, EMPLOYEE\_ID, ROLE

// getItem

// Function: select from ITEM table all information of a single item.

// Required values: item\_id

// Output: ITEM\_ID, DIST\_CODE, ITEM\_NAME, RECEIPT\_NAME, CATEGORY, UNIT, ITEM\_TYPE, COMMENT, LATEST\_PRICE, SUPPLIER\_ID, U\_ACTIVE\_INA, U\_BIZERBA, U\_BRAND, U\_CASE\_SIZE, U\_MINOR\_REPO, U\_COOKING\_IN, U\_COUNTRY, U\_DATE\_CREAT, U\_DATE\_MOTIF, U\_DESCRIPTO, U\_EXPIRY\_DAT, U\_INGREDIENT, U\_KEYWORDS, U\_NOTES, U\_ORDER, U\_PLU, U\_PRICE, U\_SILVERWARE, U\_SKU, U\_STORAGE, U\_STORAGE\_TY, U\_TYPE, U\_UPC\_CODE, U\_USERNAME\_C, U\_USERNAME\_M, U\_PRICE\_PER, U\_TAX, U\_SCALE

// getPurchaseOrder

// Function: select from PURCHASE\_ORDER table all information of a single purchase order.

// Required values: po\_id

// Output: PO\_ID, STATUS, CREATE\_DATE, SUBMIT\_DATE, DELIVERY\_DATE, DELIVERY\_TIME, RECEIVE\_DATE, REF\_NUMBER, COMMENT, SUPPLIER\_ID