**FCIS-310 Database Design**

**Assignment #8**

**60 points**

Student Name: Mikhaela Mueller

After you have successfully completed Assignments 4 and 7 as well as Small Group Activities 7 and 8, the Premiere Products database containing five tables (REP, CUSTOMER, ORDERS, PART, and ORDER\_LINE) populated with the data and the entity relationship diagram (ERD) should already be in your SQL Server account. You will find ERD and the five database tables at the end of this document.

**Part A (10 points).**

In Assignment 4, we defined the primary key(s) PK(s) for each table in the CREATE TABLE command. The foreign key (FK) constraints were defined while you were creating the ERD for the Premiere Products database. You should have the ERD in the database diagram folder.

In this assignment we will take a different approach to enforcing referential integrity constraints on the FKs. We will do it by writing and running the SQL code. The code is pretty much self-explanatory and the SQL commands that you will se are described in Chapter 8.

Open in the SQL Server the SQL code given in a file named ***Assignment 8 – Fall 2021 - SQL Code*** that resides in the Assignment 8 folder. The code first drops the five tables, if they exist. (They do exist in your database.) The associated ERD that you created in Assignment 4 will be dropped as well. Note that the tables have to be dropped in the right order to avoid the referential integrity violations. First, we drop the ORDER\_LINE table which has two FKs (ORDER\_NUM pointing to ORDERS and PART\_NUM pointing to PART. Then we drop the tables PART, ORDERS and then CUSTOMER, each having one FK. Lastly, we drop table REP that does not have the FK. Next, the SQL creates the structure for each of the five tables, one at a time. Each table gets a unique name (REP, CUSTOMER, etc.) and we name the PK attribute(s) and non-prime attributes and their data types for each table. Then, the SQL code adds and names constraints on the FK(s) for the CUSTOMER, ORDERS and ORDER\_LINE tables. It alters the metadata in the database dictionary. Finally, the last segment of the code populates the five tables, one at a time. Again, the order in which each of the tables are populated matters. They are populated in the reverse sequence to that how they were dropped. So, table REP that does not have the FK will be populated first followed by CUSTOMER, ORDERS, PART, and ORDER\_LINE will be populated last.

**Part B (50 points).**

Using the Premiere Products database write the SQL queries for ten of the following problems. Save the ten queries in a single SQL file or in ten separate files in your account on J drive. Run each query. Paste each query and the output it generated after each of the ten problems.

**After you paste the queries and the output save this document as Word or pdf file named Assignment8\_YourFirstName\_YourLastName and submit via Blackboard. See the Assignments/Assignments/Assignment 8 folder.**

Problem 1.

Use the EXISTS operator to find the customer number and customer name of each customer that placed an order before Oct 23 2020. Run the query on SQL Server. Paste the query and the output from the query below.

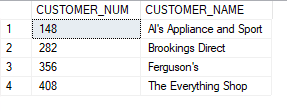
SELECT CUSTOMER\_NUM, CUSTOMER\_NAME

FROM CUSTOMER

WHERE CUSTOMER\_NUM IN (SELECT CUSTOMER\_NUM

FROM ORDERS

WHERE ORDER\_DATE < '2020-10-23');



Problem 2.

Find the description of each part included in the order number 21610 or the order number 21613. Run the query on SQL Server. Paste the query and the output from the query below. (Note that the ORDER\_NUM attribute is of the character type and values such as 21610 must be enclosed in the single quotes, '21610' for comparison.)

SELECT DESCRIPTION

FROM PART, ORDERS, ORDER\_LINE

WHERE ORDERS.ORDER\_NUM = ORDER\_LINE.ORDER\_NUM

AND ORDER\_LINE.PART\_NUM = PART.PART\_NUM

AND ORDERS.ORDER\_NUM = '21610'

AND ORDERS.ORDER\_NUM = '21613';

(I had a hard time getting this one to run/ couldn’t pinpoint my error)

Problem 3.

Find the order number and order date for each order that includes a part located in warehouse 3. Run the query on SQL Server. Paste the query and the output from the query below.

SELECT ORDERS.ORDER\_NUM, ORDERS.ORDER\_DATE

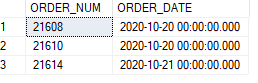
FROM ORDERS, CUSTOMER, PART, ORDER\_LINE

WHERE CUSTOMER.CUSTOMER\_NUM = ORDERS.CUSTOMER\_NUM

AND ORDERS.ORDER\_NUM = ORDER\_LINE.ORDER\_NUM

AND ORDER\_LINE.PART\_NUM = PART.PART\_NUM

AND WAREHOUSE = '3';



Problem 4.

List the customer number and customer name for each customer who placed an order on the Gas Range. (Note that you have to link several tables here.) Run the query on SQL Server. Paste the query and the output from the query below.

SELECT CUSTOMER.CUSTOMER\_NUM, CUSTOMER.CUSTOMER\_NAME

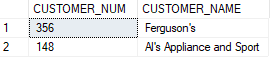
FROM CUSTOMER, ORDERS, ORDER\_LINE,PART

WHERE CUSTOMER.CUSTOMER\_NUM = ORDERS.CUSTOMER\_NUM

AND ORDERS.ORDER\_NUM = ORDER\_LINE.ORDER\_NUM

AND ORDER\_LINE.PART\_NUM = PART.PART\_NUM

AND PART.DESCRIPTION = 'Gas Range';



Problem 5.

List the part number, part description, unit price, and item class for each part that has a unit price greater than the unit price of every part in item class AP. Use the ALL operator in your query. Run the query on SQL Server. Paste the query and the output from the query below.

SELECT PART.PART\_NUM, PART.DESCRIPTION, PART.PRICE, PART.CLASS

FROM PART

WHERE NOT (CLASS = 'AP')

AND PRICE > 595.00;



Problem 6.

List the part number of any part with an unknown description. (Note that if a part has a description no part numbers will be returned.) Use the NULL operator. Run the query on SQL Server. Paste the query and the output from the query below.

SELECT PART.PART\_NUM

FROM PART, ORDER\_LINE

WHERE DESCRIPTION IS NULL;



Problem 7.

List the order number and order date for each order that was placed by Ferguson’s and that contains an order line for a Gas Range. Run the query on SQL Server. Paste the query and the output from the query below.

SELECT ORDER\_NUM, ORDER\_DATE

FROM CUSTOMER, ORDERS

WHERE CUSTOMER.CUSTOMER\_NUM = ORDERS.CUSTOMER\_NUM

AND CUSTOMER\_NAME = 'Ferguson''s'

AND ORDER\_NUM IN

(SELECT ORDER\_NUM

FROM ORDER\_LINE, PART

WHERE ORDER\_LINE.PART\_NUM = PART.PART\_NUM

AND DESCRIPTION = 'Gas Range');



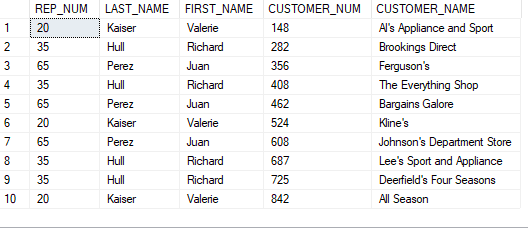
Problem 8.

List the number, last name, and first name for each sales rep together with the number and name for each customer the sales rep represents. Use aliases R and C for REP and CUSTOMER. Run the query on SQL Server. Paste the query and the output from the query below.

SELECT R.REP\_NUM, R.LAST\_NAME, R.FIRST\_NAME, C.CUSTOMER\_NUM, C.CUSTOMER\_NAME

FROM REP R, CUSTOMER C

WHERE R.REP\_NUM = C.REP\_NUM



Problem 9.

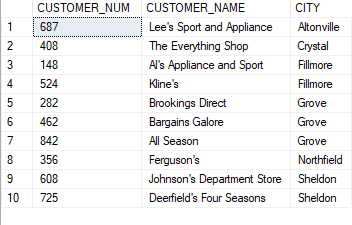
For each pair of customers located in the same city, display the customer number, customer name, and city. You need to join the table to itself. Run the query on SQL Server. Paste the query and the output from the query below.

SELECT CUSTOMER\_NUM, CUSTOMER\_NAME, CITY

FROM CUSTOMER

WHERE CUSTOMER.CITY = CUSTOMER.CITY

ORDER BY CITY;



Problem 10.

Use the IN operator or the INTERSECT set operator to list the number and name of each customer that is represented by sales rep 65 and that currently has orders on file.

SELECT CUSTOMER\_NUM, CUSTOMER\_NAME

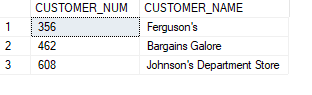
FROM CUSTOMER

WHERE REP\_NUM IN

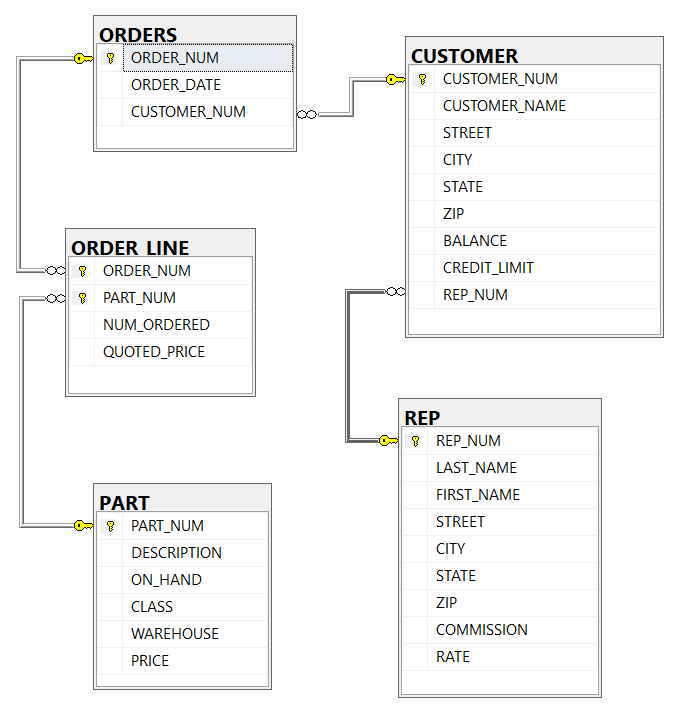
(SELECT REP\_NUM

FROM CUSTOMER

WHERE REP\_NUM = '65');

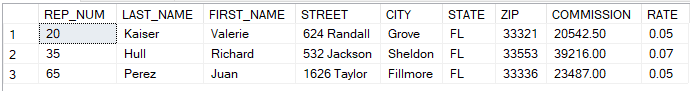


ERD – PREMIERE PRODUCTS DATABASE



PREMIERE PRODUCTS SCHEMA

TABLE REP



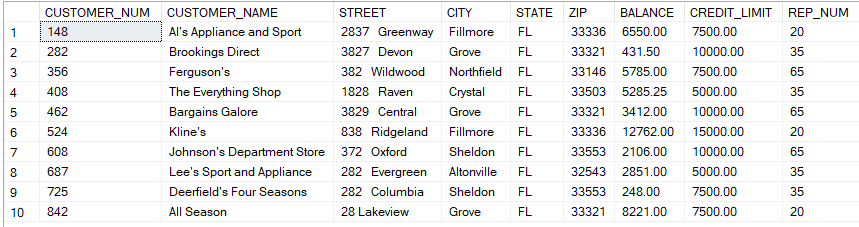
TABLE CUSTOMER

TABLE ORDERS

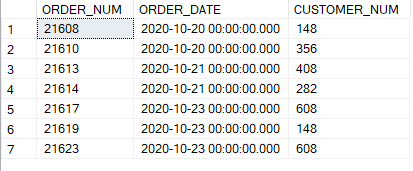


TABLE PART

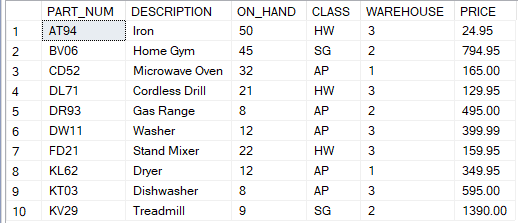


TABLE ORDER\_LINE

