Database: Chapter 5 Assignment 4

Complete exercises from class and the TAL Distributor exercises in text page 163-164, exercises 14, 15, 16, 17, 18, 19, 20 and 21.

Please type the query in the SQL Statement that you use in either Microsoft Access or Oracle Application Express.  Once query has run correctly and provided the results, take a screen snip of the results, and then copy into the chart below. Save this completed Word document and submit to appropriate dropbox on myHills.

20 pts.

|  |
| --- |
| **List the number and name of each customer that is represented by sales rep 15** |
| **SELECT STATEMENT:** |
|  |
| **Inner Join (3 ways)**  **List the number and name of each customer that has orders on file.**  **Join tables**  **Note: Natural join is similar to inner join without the on clause, it relates the tables automatically with common column.** |
| **SELECT STATEMENT:**  **Normal Join**  **SELECT C.CUSTOMER\_NUM, CUSTOMER\_NAME**  **FROM CUSTOMER C, ORDERS O**  **WHERE C.CUSTOMER\_NUM = O.CUSTOMER\_NUM**  **Inner Join**  **SELECT DISTINCT C.CUSTOMER\_NUM, CUSTOMER\_NAME**  **FROM CUSTOMER C INNER JOIN ORDERS O**  **ON C.CUSTOMER\_NUM = O.CUSTOMER\_NUM**  **Natural Join (Not in textbook)**  **SELECT DISTINCT CUSTOMER\_NUM, CUSTOMER\_NAME**  **FROM CUSTOMER NATURAL JOIN ORDERS** |
| **Results:** |
| **Union**  **List the number and name of each customer that is represented by sales rep 15 or currently has orders on file, or both** |
| **SELECT STATEMENT:**  **SELECT DISTINCT C.CUSTOMER\_NUM, CUSTOMER\_NAME**  **FROM CUSTOMER C, ORDERS O**  **WHERE C.CUSTOMER\_NUM = O.CUSTOMER\_NUM**  **UNION**  **SELECT CUSTOMER\_NUM, CUSTOMER\_NAME**  **FROM CUSTOMER**  **WHERE REP\_NUM = '15'** |
| **Results:** |
| **Intersect**  **List the number and name of each customer that is represented by sales rep 15 and currently has orders on file,** |
| **SELECT STATEMENT:**  SELECT DISTINCT C.CUSTOMER\_NUM, CUSTOMER\_NAME  FROM CUSTOMER C, ORDERS O  WHERE C.CUSTOMER\_NUM = O.CUSTOMER\_NUM  INTERSECT  SELECT CUSTOMER\_NUM, CUSTOMER\_NAME  FROM CUSTOMER  WHERE REP\_NUM = '15' |
| **Results:** |
| **Minus**  **List the number and name of each customer that is represented by sales rep 15 and does not currently has orders on file,** |
| **SELECT STATEMENT:**  SELECT DISTINCT C.CUSTOMER\_NUM, CUSTOMER\_NAME  FROM CUSTOMER C, ORDERS O  WHERE C.CUSTOMER\_NUM = O.CUSTOMER\_NUM  MINUS  SELECT CUSTOMER\_NUM, CUSTOMER\_NAME  FROM CUSTOMER  WHERE REP\_NUM = '15' |
| **Results:** |
| **ANY and ALL**  **Find the customer number, name, current balance and rep number of each customer whose balance exceeds the maximum balance of all customers represented by sales rep 30.** |
| **SELECT STATEMENT:**  SELECT CUSTOMER\_NUM, CUSTOMER\_NAME, BALANCE, REP\_NUM  FROM CUSTOMER  WHERE BALANCE > Any  (SELECT BALANCE  FROM CUSTOMER  WHERE REP\_NUM = '30') |
| **Results:** |
| **ANY and ALL**  **Find the customer number, name, current balance and rep number of each customer whose balance is greater than the balance of at least one customer represented by sales rep 30.** |
| **SELECT STATEMENT:** |
| **Results:** |
| **Outer Join**  **Display the customer number, customer name, order number and order date for all orders. Include all customers in the results, even if they don’t have any orders.** |
| **SELECT STATEMENT:**  SELECT C.CUSTOMER\_NUM, CUSTOMER\_NAME, ORDER\_NUM, ORDER\_DATE  FROM CUSTOMER C LEFT JOIN ORDERS O  ON C.CUSTOMER\_NUM = O.CUSTOMER\_NUM |
| **Results:** |
| **5-15** List the order number and order date for each order that either was placed by Almondton General Store or that contains an order line for a Fire Engine. |
| **5-15 SELECT STATEMENT:**    SELECT ORDERS.ORDER\_NUM, ORDERS.ORDER\_DATE  FROM ITEM, ORDER\_LINE, ORDERS  WHERE ORDER\_LINE.ORDER\_NUM = ORDERS.ORDER\_NUM  AND ITEM.ITEM\_NUM = ORDER\_LINE.ITEM\_NUM  AND ITEM.DESCRIPTION = 'Fire Engine'  UNION  SELECT ORDERS.ORDER\_NUM, ORDERS.ORDER\_DATE  FROM ORDERS, CUSTOMER  WHERE CUSTOMER.CUSTOMER\_NUM = ORDERS.CUSTOMER\_NUM  AND CUSTOMER.CUSTOMER\_NAME = 'Almondton General Store' |
| **5-15** **Results:** |
| **5-16** List the order number and order date for each order that was placed by Almondton General Store and that contains an order line for a Fire Engine. |
| **5-16 SELECT STATEMENT:**  SELECT ORDERS.ORDER\_NUM, ORDERS.ORDER\_DATE  FROM ITEM, ORDER\_LINE, ORDERS, CUSTOMER  WHERE ORDER\_LINE.ORDER\_NUM = ORDERS.ORDER\_NUM  AND ITEM.ITEM\_NUM = ORDER\_LINE.ITEM\_NUM  AND CUSTOMER.CUSTOMER\_NUM = ORDERS.CUSTOMER\_NUM  AND CUSTOMER.CUSTOMER\_NAME = 'Almondton General Store'  AND ITEM.DESCRIPTION = 'Fire Engine' |
| **5-16 Results:** |
| **5-17** List the order number and order date for each order that was placed by Almondton General Store but does not contains an order line for a Fire Engine. |
| **5-17** **SELECT STATEMENT:**    SELECT ORDER\_NUM, ORDER\_DATE  FROM CUSTOMER, ORDERS  WHERE CUSTOMER.CUSTOMER\_NUM = ORDERS.CUSTOMER\_NUM  AND CUSTOMER\_NAME = 'Almondton General Store'  AND ORDER\_NUM NOT IN  (SELECT ORDER\_NUM  FROM ORDER\_LINE, ITEM  WHERE ORDER\_LINE.ITEM\_NUM = ITEM.ITEM\_NUM  AND DESCRIPTION = 'Fire Engine') |
| **5-17 Results:** |
| **5-18** List the item number, description, unit price, and category for each item that has a unit price greater than the unit price of every item in category GME. Use either the ALL or the ANY operator in your query. |
| **5-18** **SELECT STATEMENT:**  SELECT ITEM\_NUM, DESCRIPTION, PRICE, CATEGORY  FROM ITEM  WHERE PRICE > ALL  (SELECT PRICE  FROM ITEM  WHERE CATEGORY = 'GME') |
| **5-18 Results:** |
| **5-19.** For each item, list the item number, description, units on hand, order number and number of units ordered. All items should be included in the results. Order the results by item number. |
| **5-19** **SELECT STATEMENT:**  **SELECT ITEM.ITEM\_NUM, DESCRIPTION, ON\_HAND, ORDER\_LINE.ORDER\_NUM, NUM\_ORDERED**  **FROM ITEM**  **LEFT JOIN ORDER\_LINE**  **ON ITEM.ITEM\_NUM= ORDER\_LINE.ITEM\_NUM**  **ORDER BY ITEM.ITEM\_NUM** |
| **5-19 Results:** |
| **5-21.** For each rep, list the customer number, customer name, rep last name and rep first name. All reps should be included in the results. Order the results by rep numbers. (Do 2 ways, 1)use outer join and SET operation.) |
| **5-21** **SELECT STATEMENT:**  SELECT REP.REP\_NUM, CUSTOMER.CUSTOMER\_NUM, CUSTOMER.CUSTOMER\_NAME, REP.LAST\_NAME, REP.FIRST\_NAME  FROM REP  LEFT JOIN CUSTOMER  ON CUSTOMER.REP\_NUM = REP.REP\_NUM  ORDER BY REP.REP\_NUM |
| **5-21 Results:** |