Lab 04 – Multi-Table Queries and Views

Objective:

The purpose of this lab is to introduce students to querying data from multiple tables. Relationships are used in relational databases to reduce redundant and repetitive data, but it is necessary to reconnect these tables when extracting data and obtaining information. Student will be able to:

- produce query results containing data from multiple tables using ANSI-92 joins and demonstrate their knowledge of inner, outer and full joins.
- To actively troubleshoot queries to handle potentially ambiguous fields across multiple tables through the use of aliases
- Students learn to create and modify views.

Submission:

Your submission will be a single WORD file with the solutions provided.

Your submission needs to follow the same question order and clearly indicate the answers to each question. Make sure every SQL statement terminates with a semicolon.

ALL questions must be answered using ANSI-92 JOINs unless otherwise stated. ANSI-89 are obsolete and should not be used in new query derivations. We only teach them in case you see them in the workplace, that you know what they are and how they work.

Tasks:

Select data from multiple tables

1. Create a query that shows retail customers first name and last name along with their sales rep employee number and their first name, last name, city, phone numberand postal code for all retail customers who live in Singapore.
    a. Answer this question using an ANSI-89 Join

    ```
    SELECT
        CONTACTFIRSTNAME,CONTACTLASTNAME,SALESREPEMPLOYEENUMBER,CITY,PH
        ONE,POSTALCODE
    FROM RETAILCUSTOMERS,RETAILEMPLOYEES
        WHERE CITY='Singapore';
    ```

    b. Answer this question using an ANSI-92 Join

    ```
    SELECT
        CONTACTFIRSTNAME,CONTACTLASTNAME,SALESREPEMPLOYEENUMBER,CITY,PH
        ONE,POSTALCODE
    FROM RETAILCUSTOMERS LEFT OUTER JOIN RETAILEMPLOYEES ON
        RETAILCUSTOMERS.CUSTOMERNUMBER=RETAILEMPLOYEES.EMPLOYEENUMBER
        WHERE CITY='Singapore';
    ```

2. Create a query that displays all retail payments made by retail customers from USA.
    a. Sort the output by Customer Number.

    ```
    SELECT *
    ```

```
FROM RETAILCUSTOMERS RIGHT OUTER JOIN RETAILPAYMENTS ON
    RETAILCUSTOMERS.CUSTOMERNUMBER=RETAILPAYMENTS.CUSTOMERNUMBER
WHERE COUNTRY='USA'
    ORDER BY RETAILCUSTOMERS.CUSTOMERNUMBER;
```

b. Only display the Customer Number, Customer Name,Country, Payment Date and Amount.

1. Make sure the date is displayed clearly to know what date it is. (i.e. what date is 02-04-19??? – Feb 4, 2019, April 2, 2019, April 19, 2002, ….)

```
SELECT
    RETAILCUSTOMERS.CUSTOMERNUMBER,CUSTOMERNAME,RETAILCUSTOMERS.CO
    UNTRY,TO_DATE(PAYMENTDATE,'MM DD, YYYY') AS PAYMENTDATE,AMOUNT
FROM RETAILCUSTOMERS RIGHT OUTER JOIN RETAILPAYMENTS ON
    RETAILCUSTOMERS.CUSTOMERNUMBER=RETAILPAYMENTS.CUSTOMERNUMBER
WHERE RETAILCUSTOMERS.COUNTRY='USA'
    ORDER BY RETAILCUSTOMERS.CUSTOMERNUMBER;
```

3. Create a query that shows all Canada customers who have not made a payment. Display only the customer number ,customer name, amount sorted by customer number.

```
SELECT RETAILCUSTOMERS.CUSTOMERNUMBER,CUSTOMERNAME,AMOUNT
FROM RETAILCUSTOMERS RIGHT OUTER JOIN RETAILPAYMENTS ON
    RETAILCUSTOMERS.CUSTOMERNUMBER=RETAILPAYMENTS.CUSTOMERNUMBER
WHERE RETAILCUSTOMERS.COUNTRY='Canada'
    ORDER BY RETAILPAYMENTS.CUSTOMERNUMBER;
```

Views and Joins

4. Display all the retail orders with quantity ordered, price of each item, who have their order shipped and who live in Denmark

```
        SELECT QUANTITYORDERED,PRICEEACH
FROM ORDERDETAILS JOIN RETAILORDERS USING (ORDERNUMBER)
  JOIN RETAILCUSTOMERS ON
RETAILORDERS.CUSTOMERNUMBER=RETAILCUSTOMERS.CUSTOMERNUMBER
WHERE RETAILORDERS.STATUS='Shipped' AND RETAILCUSTOMERS.COUNTRY='Denmark';
```

5. Create a view (***vwProductOrder***) to list all the retail products with the following data:
   Product code, product name, msrp, buyprice, quantity ordered, andprice for each product in every order.

```
CREATE VIEW vwProductOrder AS
  SELECT
RETAILPRODUCTS.PRODUCTCODE,RETAILPRODUCTS.PRODUCTNAME,MSRP,BUYPRICE,Q
UANTITYORDERED,PRICEEACH
  FROM RETAILPRODUCTS,ORDERDETAILS;
```

   b) Write a statement to view the results of the view just created.

```
SELECT * FROM vwProductOrder;
```

6. Using the ***vwProductOrder*** view, display the product order information with product name, buyprice ,order line number anwhose buy price is in the range from $30 to $40 and whose product code starts with 's32'. Sort the output based on product name and then buy price. (Hint: orderLineNumber is not in the view then how can you get in this query?)

```
CREATE OR REPLACE VIEW vwProductOrder AS
  SELECT
RETAILPRODUCTS.PRODUCTNAME,RETAILPRODUCTS.BUYPRICE,ORDERDETAILS.ORDERLINENUMBER
  FROM RETAILPRODUCTS,ORDERDETAILS
  WHERE RETAILPRODUCTS.BUYPRICE IN(30,40) AND RETAILPRODUCTS.PRODUCTCODE LIKE '%s32';
```

7. Create a query that displays the customer order information with customer number, first name, last name, phone, and credit limits for all customers who do not have any orders.

```
 SELECT
    CUSTOMERNUMBER,CONTACTFIRSTNAME,CONTACTLASTNAME,PHONE,CREDITLIMI
    T
FROM RETAILCUSTOMERS LEFT OUTER JOIN ORDERDETAILS ON
    RETAILCUSTOMERS.CUSTOMERNUMBER=ORDERDETAILS.ORDERNUMBER
    WHERE ORDERNUMBER IS NULL;
```

8. Create a view (***vwEmployeeManager***) to display the information of all retail employees first name and last name and their managers first name and managers last name if there is any manager that the employee reports to. Include all employees, including those who do not report to anyone.

```
CREATE VIEW vmEmployeeManager AS
SELECT r.FIRSTNAME,r.LASTNAME,m.FIRSTNAME AS
MANAGERFIRSTNAME,m.LASTNAME AS MANAGERLASTNAME
FROM RETAILEMPLOYEES r FULL OUTER JOIN RETAILEMPLOYEES m USING
(EMPLOYEENUMBER)
WHERE r.REPORTSTO=EMPLOYEENUMBER OR r.REPORTSTO IS NULL;
```

9. Modify the ***vwEmployeeManager*** view so the view returns only employee information foremployees who have a manager. Do not DROP and recreate the view – modify it. (Google is your friend).

```
CREATE OR REPLACE VIEW vwEmployeeManager
 SELECT
EMPLOYEENUMBER,LASTNAME,FIRSTNAME,REPORTSTO,JOBTITLE,OFFICECODE,EMAIL,EXTENSI
ON
 FROM RETAILEMPLOYEES
 WHERE REPORTSTO IS NOT NULL;
```

10. Drop both ***vwProductOrder*** and ***vwEmployeeManager*** views.

```
DROP VIEW IF EXISTS
        vwProductOrder,
        vwEmployeeManager;
```