Login Credentials:

Graphical user interface, application

Description automatically generated

1. 1. Run the LAB4 sample script to create table GameTeams, Teamplayers and Teamfields table. Create a query that shows teamID,teamname,fieldname, address who have shirtcolor as white.

**ANS: SELECT teamid, teamname, fieldname, address**

**FROM gameteams INNER JOIN teamfields**

**ON gameteams.homefield = teamfields.fieldname**

**WHERE LOWER(gameteams.shirtcolor) = 'white';**

**Graphical user interface, text, application, email

Description automatically generated**

1. . Create a query that shows teamID,teamname,firstname and lastname who belong to gameteam as well as not in a gameteam.

**ANS:** **SELECT teamid, teamname, firstname, lastname**

**FROM gameteams RIGHT OUTER JOIN teamplayers**

**USING (teamid);**

**Graphical user interface, text

Description automatically generated**

1. Create a query that shows employee number, first name, last name, city, phone number

and postal code for all employees in France.

a. Answer this question using an ANSI-89 Join

b. Answer this question using an ANSI-92 Join

**ANS:**

**A.) SELECT employeenumber, firstname, lastname, city, phone, postalcode**

**FROM offices, employees**

**WHERE employees.officecode = offices.officecode**

**AND LOWER (offices.country) LIKE 'france';**

**Graphical user interface, text

Description automatically generated**

**B.) SELECT employeenumber,firstname, lastname, city, phone, postalcode**

**FROM offices INNER JOIN employees**

**ON employees.officecode = offices.officecode**

**WHERE LOWER (offices.country) LIKE 'france';**

**Graphical user interface, text

Description automatically generated**

4.) Create a query that displays all payments made by customers from Canada.

a. Sort the output by Customer Number.

b. Only display the Customer Number, Customer Name, 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, ….)

**ANS: SELECT C.customernumber, C.customername, TO\_CHAR(p.paymentdate,'MON**

**DD,YYYY')paymentdate, P.amount**

**FROM payments P INNER JOIN customers C**

**ON P.customernumber = C.customernumber**

**AND LOWER(C.country) LIKE 'canada'**

**ORDER BY customernumber;**

**Text

Description automatically generated**

1. Create a query that shows all USA customers who have not made a payment. Display only the customer number and customer name sorted by customer number

**ANS: SELECT customernumber,customername**

**FROM customers FULL OUTER JOIN payments USING (customernumber)**

**WHERE payments.amount IS NULL AND LOWER (customers.country) LIKE 'usa'**

**ORDER BY customernumber DESC;**

**Graphical user interface, text, application, email

Description automatically generated**

1. a) Create a view (vwCustomerOrder) to list all orders with the following data for all

customers:

Customer Number, Order number, order date, product name, quantity ordered, and

price for each product in every order.

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

**ANS:**

**A.) Create view vwcustomerorder**

**AS**

**Select**

**orders.customernumber,**

**orders.ordernumber,**

**orders.orderdate,**

**products.productname,**

**orderdetails.quantityordered,**

**orderdetails.priceeach**

**from orders**

**Join orderdetails on orders.ordernumber = orderdetails.ordernumber**

**join products on orderdetails.productcode = products.productcode;**

**Graphical user interface, text, application, email

Description automatically generated**

B.) **SELECT \* FROM vwcustomerorder;**

**Graphical user interface, table

Description automatically generated**

7.) Using the vwCustomerOrder view, display the order information for customer number

124. Sort the output based on order number and then order line number. (Yes, I

know orderLineNumber is not in the view)

**ANS: SELECT DISTINCT**

**customernumber,**

**orderlinenumber,**

**vwcustomerorder.ordernumber**

**FROM**

**vwcustomerorder**

**INNER JOIN orderdetails ON vwcustomerorder.ordernumber= orderdetails.ordernumber**

**WHERE**

**customernumber =124**

**ORDER BY**

**ordernumber,**

**orderlinenumber;**

**Graphical user interface, table

Description automatically generated**

1. Create a query that displays the customer number, first name, last name, phone, and

credit limits for all customers who do not have any orders.

**ANS: SELECT DISTINCT(customers.customernumber), customername, contactfirstname, contactlastname, phone, creditlimit**

**FROM customers LEFT OUTER JOIN orders ON customers.customernumber = orders.customernumber;**

**Graphical user interface, table

Description automatically generated**

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

anyone.

**ANS: CREATE VIEW vwemployeemanager AS**

**SELECT**

**employ.employeenumber,**

**employ.lastname,**

**employ.firstname,**

**employ.extension,**

**employ.email,**

**employ.officecode,**

**employ.reportsto,**

**employ.jobtitle,**

**em.firstname AS managerfirst,**

**em.lastname AS managerlast**

**FROM employees employ**

**INNER JOIN employees em ON employ.reportsto = em.employeenumber;**

**Table

Description automatically generated**

1. Modify the employee\_manager view so the view returns only employee information for employees who have a manager. Do not DROP and recreate the view – modify it. (Google is your friend).

**ANS: CREATE OR REPLACE VIEW vwemployeemanager AS**

**SELECT \***

**FROM employees**

**WHERE reportsto IS NOT NULL;**

**Graphical user interface, text, application, email

Description automatically generated**

1. Drop both customer\_order and employee\_manager views.

**ANS: DROP VIEW vwcustomerorder;**

**DROP VIEW vwemployeemanager;**

**Graphical user interface, text, application, email

Description automatically generated**