**Homework #1:**

**Name:** Abdulrahman Abdulrazak Alnajlat

**Linkedin:** www.linkedin.com/in/aboodalnajlat03

**Q1) DDL:**

CREATE TABLE Employee (

ID INT PRIMARY KEY,

person\_name VARCHAR(255) NOT NULL,

street VARCHAR(255) NOT NULL,

city VARCHAR(255) NOT NULL

);

CREATE TABLE Company (

company\_name VARCHAR(255) PRIMARY KEY,

city VARCHAR(255) NOT NULL

);

CREATE TABLE Works (--

ID INT,

company\_name VARCHAR(255),

salary DECIMAL(8, 3) NOT NULL,

PRIMARY KEY (ID, company\_name),

FOREIGN KEY (ID) REFERENCES Employee(ID) ON DELETE CASCADE,

FOREIGN KEY (company\_name) REFERENCES Company(company\_name) ON DELETE CASCADE

);

CREATE TABLE Manages (

ID INT,

manager\_id INT,

PRIMARY KEY (ID),

FOREIGN KEY (ID) REFERENCES Employee(ID) ON DELETE CASCADE,

FOREIGN KEY (manager\_id) REFERENCES Employee(ID) ON DELETE SET NULL

);

**Q2) DML:**

**A. Find the ID of each customer of the bank who has an account but not a loan:**

SELECT DISTINCT D.ID

FROM Depositor D

WHERE D.ID NOT IN (

SELECT B.ID

FROM Borrower B

);

**B. Find the ID of each customer who lives on the same street and in the same city as customer '12345'.**

SELECT C1.ID

FROM Customer C1

WHERE C1.customer\_street = (

SELECT customer\_street

FROM Customer

WHERE ID = '12345'

)

AND C1.customer\_city = (

SELECT customer\_city

FROM Customer

WHERE ID = '12345'

)

AND C1.ID != '12345';

**C. Find the name of each branch that has at least one customer who has an account in the bank and who lives in “Harrison”.**

**SELECT branch\_name**

**FROM Branch**

**WHERE branch\_name IN (**

**SELECT DISTINCT A.branch\_name**

**FROM Account A**

**JOIN Depositor D ON A.account\_number = D.account\_number**

**JOIN Customer C ON D.ID = C.ID**

**WHERE C.customer\_city = 'Harrison'**

**);**

**Q3) Using SQL Window Functions.**

**A. From the demand table, find the cumulative total sum for qty.**

**SELECT day, qty, SUM(qty) OVER (ORDER BY day) AS cumQty**

**FROM demand;**

**B. Extract the two worst performing days of each product in terms of number of qty sold. Paraphrasing it: Get the days corresponding to the two minimum most values of qty for each product.**

WITH RankedDays AS (

SELECT

product,day,qty,

RANK() OVER (PARTITION BY product ORDER BY qty ASC) AS rn

FROM demand

)

SELECT

product,day,qty,rn

FROM RankedDays

WHERE rn <= 2

ORDER BY product, rn;