LAB – 3 PRACTICALS

Instructions:

1. Students should use “bank” database for Task-1 and Task-2.
2. No requirement of any other database for Task-3 and Task-4 but, queries (to create table and to insert data) are given for Task-3 and Task-4 separately. Students should use them before proceeding to the question.

****

TASK-1

Select distinct customer name, id, account number, total loan amount of all female customer having exactly 3 loans.

****

TASK-2

Find the time that passed between a payment and all payments occurring within year(365 days) later on the same payment number and amount paid on first date using cross join AND in ascending order of date1.

Hint:

- expected output - payment_number, date1 , date2 , date_diff , amount_paid_on_date1
- {DATEDIFF(date1, date2)} – gives difference between the dates in days.

****

TASK-3

We have two following tables (STORE_INFO and GEOGRAPHY) for this task.

```
CREATE TABLE STORE_INFO (CITY VARCHAR(255), SALES_DATE DATE, SALES INT);
INSERT INTO STORE_INFO (CITY, SALES_DATE, SALES) VALUES ('MUMBAI', '2018-02-06', '125');
INSERT INTO STORE_INFO (CITY, SALES_DATE, SALES) VALUES ('CHENNAI', '2018-02-01', '135');
INSERT INTO STORE_INFO (CITY, SALES_DATE, SALES) VALUES ('DELHI', '2018-01-07', '100');
INSERT INTO STORE_INFO (CITY, SALES_DATE, SALES) VALUES ('KOLKATA', '2018-02-03', '187');
INSERT INTO STORE_INFO (CITY, SALES_DATE, SALES) VALUES ('AHMEDABAD', '2018-02-02', '112');
INSERT INTO STORE_INFO (CITY, SALES_DATE, SALES) VALUES ('BANGALORE', '2018-02-07', '108');
INSERT INTO STORE_INFO (CITY, SALES_DATE, SALES) VALUES ('DELHI', '2018-01-26', '113');
INSERT INTO STORE_INFO (CITY, SALES_DATE, SALES) VALUES ('HYDERABAD', '2018-02-06', '125');
INSERT INTO STORE_INFO (CITY, SALES_DATE, SALES) VALUES ('JAIPUR', '2018-02-06', '125');
INSERT INTO STORE_INFO (CITY, SALES_DATE, SALES) VALUES ('BANGALORE', '2018-02-01', '100');
```

```
CREATE TABLE GEOGRAPHY (REGION_NAME VARCHAR(255), CITY VARCHAR(255));
INSERT INTO GEOGRAPHY (REGION_NAME, CITY) VALUES ('NORTH', 'JAIPUR');
INSERT INTO GEOGRAPHY (REGION_NAME, CITY) VALUES ('WEST', 'MUMBAI');
INSERT INTO GEOGRAPHY (REGION_NAME, CITY) VALUES ('NORTH', 'DELHI');
INSERT INTO GEOGRAPHY (REGION_NAME, CITY) VALUES ('SOUTH', 'CHENNAI');
INSERT INTO GEOGRAPHY (REGION_NAME, CITY) VALUES ('EAST', 'KOLKATA');
INSERT INTO GEOGRAPHY (REGION_NAME, CITY) VALUES ('WEST', 'AHMEDABAD');
INSERT INTO GEOGRAPHY (REGION_NAME, CITY) VALUES ('SOUTH', 'BANGALORE');
INSERT INTO GEOGRAPHY (REGION_NAME, CITY) VALUES ('SOUTH', 'HYDERABAD');
INSERT INTO GEOGRAPHY (REGION_NAME, CITY) VALUES ('WEST', 'PUNE');
INSERT INTO GEOGRAPHY (REGION_NAME, CITY) VALUES ('WEST', 'NAGPUR');
INSERT INTO GEOGRAPHY (REGION_NAME, CITY) VALUES ('NORTH', 'GURUGRAM');
```

Question: We want to see the results for all west region cities regardless whether there is a sale (with total amount) in the STORE_INFO table.

****

TASK-4

We have two following tables (ORDERS and EMPLOYEE) for this task.

```
CREATE TABLE ORDERS (ORDER_ID INT NOT NULL PRIMARY KEY, CUSTOMER_ID INT, EMPLOYEE_ID INT, ORDER_DATE DATE);
INSERT INTO ORDERS (ORDER_ID, CUSTOMER_ID, EMPLOYEE_ID, ORDER_DATE) VALUES ('10251', '90', '5', '2018-02-06');
INSERT INTO ORDERS (ORDER_ID, CUSTOMER_ID, EMPLOYEE_ID, ORDER_DATE) VALUES ('10253', '91', '4', '2018-01-22');
INSERT INTO ORDERS (ORDER_ID, CUSTOMER_ID, EMPLOYEE_ID, ORDER_DATE) VALUES ('10252', '80', '5', '2018-02-03');
INSERT INTO ORDERS (ORDER_ID, CUSTOMER_ID, EMPLOYEE_ID, ORDER_DATE) VALUES ('10255', '70', '3', '2018-01-16');
INSERT INTO ORDERS (ORDER_ID, CUSTOMER_ID, EMPLOYEE_ID, ORDER_DATE) VALUES ('10254', '85', '5', '2018-01-26');
INSERT INTO ORDERS (ORDER_ID, CUSTOMER_ID, EMPLOYEE_ID, ORDER_DATE) VALUES ('10256', '25', '2', '2018-01-25');
INSERT INTO ORDERS (ORDER_ID, CUSTOMER_ID, EMPLOYEE_ID, ORDER_DATE) VALUES ('10258', '75', '5', '2018-01-30');
INSERT INTO ORDERS (ORDER_ID, CUSTOMER_ID, EMPLOYEE_ID, ORDER_DATE) VALUES ('10259', '45', '4', '2018-01-24');
INSERT INTO ORDERS (ORDER_ID, CUSTOMER_ID, EMPLOYEE_ID, ORDER_DATE) VALUES ('10257', '29', '4', '2018-02-07');
```

INSERT INTO ORDERS (ORDER_ID, CUSTOMER_ID, EMPLOYEE_ID, ORDER_DATE) VALUES ('10261', '38', '1', '2018-02-05');

INSERT INTO ORDERS (ORDER_ID, CUSTOMER_ID, EMPLOYEE_ID, ORDER_DATE) VALUES ('10260', '36', '1', '2018-01-21');

INSERT INTO ORDERS (ORDER_ID, CUSTOMER_ID, EMPLOYEE_ID, ORDER_DATE) VALUES ('10262', '37', '2', '2018-01-20');

INSERT INTO ORDERS (ORDER_ID, CUSTOMER_ID, EMPLOYEE_ID, ORDER_DATE) VALUES ('10263', '34', '1', '2018-02-07');

CREATE TABLE EMPLOYEE (EMPLOYEE_ID INT NOT NULL PRIMARY KEY, NAME VARCHAR(255), CITY VARCHAR(255));

INSERT INTO EMPLOYEE (EMPLOYEE_ID, NAME, CITY) VALUES ('1', 'JHON', 'LONDON');

INSERT INTO EMPLOYEE (EMPLOYEE_ID, NAME, CITY) VALUES ('2', 'WENG', 'PARIS');

INSERT INTO EMPLOYEE (EMPLOYEE_ID, NAME, CITY) VALUES ('3', 'ZHONG', 'NEW YORK');

INSERT INTO EMPLOYEE (EMPLOYEE_ID, NAME, CITY) VALUES ('4', 'ZHU', 'SAN DEIGO');

INSERT INTO EMPLOYEE (EMPLOYEE_ID, NAME, CITY) VALUES ('5', 'WEI', 'SINGAPORE');

Question: We want to see the results for two employees (whose name is either WENG or ZHONG) only regardless whether he has booked less than 4 orders. A student should display three columns (employee ID, employee name, total # of orders) for the results and it should be sorted based on employee ID.

##########