BDTM: Big Data Tools for Managers
2nd Internal Question Paper [Set-A]

Q1. Demonstrate UPDATE & DELETE statements in MySQL. [10]

Table: **EMPLOYEE**

ID	NAME	CITY	STATE	COUNTRY
1	AAA	PUNE	MAH	INDIA
2	BBB	MUMBAI		
3	CCC	TUMKUR	KAR	
4	DDD	BANGALORE		
5	EEE	GANDHINAGAR		

A. Update STATE as MAH for city MUMBAI

```
UPDATE EMPLOYEE SET STATE=`MAH` WHERE CITY = `MUMBAI`;
```

B. Update value IN for Country for all the records, after updating Country should contains ‘IN’ values for entire table.

```
UPDATE EMPLOYEE SET COUNTRY=`IN` ;
```

C. DELETE the records where NAME is EEE

```
DELETE FROM EMPLOYEE WHERE NAME = `EEE`;
```

D. DELETE the records where STATE is MAH

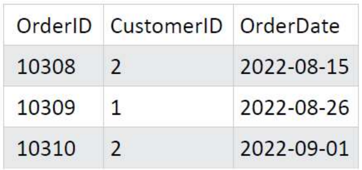
```
DELETE FROM EMPLOYEE WHERE STATE = `MAH`;
```

E. Display all available records in EMPLOYEE table

```
SELECT * FROM EMPLOYEE;
```

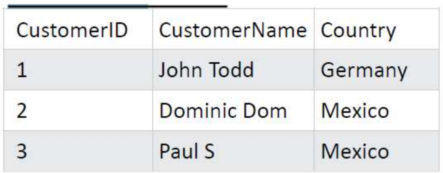
Q2. Demonstrate JOINS in MySQL for the given Table. [10]

Table: **Orders**



OrderID	CustomerID	OrderDate
10308	2	2022-08-15
10309	1	2022-08-26
10310	2	2022-09-01

Table: **Customers**



CustomerID	CustomerName	Country
1	John Todd	Germany
2	Dominic Dom	Mexico
3	Paul S	Mexico

A. Perform Inner Join with Orders & Customer Table

```
SELECT Orders.*, Customers.* FROM Orders
INNER JOIN Customers
ON Orders.CustomerID = Customers.CustomerID;
```

B. Create a MySQL View for left join with Orders & Customer Table

```
CREATE VIEW my_view AS
SELECT Orders.*, Customers.* FROM Orders
LEFT JOIN Customers
ON Orders.CustomerID = Customers.CustomerID;
```

C. Use a View which created in Q2[B] to display data for left join queries

```
SELECT * FROM my_view;
```

D. Delete a MySQL View which created in Q2[B]

```
DROP VIEW my_view;
```

Q3. Perform SET operations on given MySQL Tables [10]

Tables: **CUSTOMERS_1**, **CUSTOMERS_2**

A. Display all the records including duplicate records from CUSTOMER_1 and CUSTOMER_2 using MySQL SET operators.

```
SELECT * FROM CUSTOMER_1
UNION ALL
SELECT * FROM CUSTOMER_2;
```

B. Display common records from CUSTOMER_1 and CUSTOMER_2 tables.

```
SELECT * FROM CUSTOMER_1
INTERSECT
SELECT * FROM CUSTOMER_2;
```

Q4. [A] Perform basic operations on R Vector [5]

1. Create R vector for given elements (99, 86, 63, 81, 48)

```
my_vector <- c(99, 86, 63, 81, 48)
```

2. Display the vector elements

```
print(my_vector)
```

3. Display the length of R vector

```
print(my_vector)
            or
length(my_vector)
```

4. Sort vector elements in ascending order

```
sort(my_vector)
```

Q4. [B] Create and Display R Matrix for below given elements [5]

$$\begin{bmatrix} 8 & 4 & 3 \\ -5 & 6 & -2 \\ 7 & 9 & -8 \end{bmatrix}$$

1. Create Matrix with given elements

```
my_matrix <- matrix(c(8, 4, 3, -5, 6, -2, 7, 9, -8), nrow = 3, byrow = TRUE)
```

2. Display Matrix elements in R

```
print(my_matrix)
```

Q5. Read Vehicle Park dataset and write R code for following statements. [10]

About Dataset:

- Vehicle Park data contains the no of vehicles are present on road or market in India from Year 2000 to 2022
- **Columns:**
  - YEAR: Vehicle Sales Year
  - VEHICLE_TYPE: Type of vehicle sold to the market (Truck, Bus, Four & Two-wheeler, Others)
  - BRAND: Vehicle brand & Manufacturer
  - VEHICLE_COUNT: No of vehicle sold in market for a year
  - AGE_GROUP: Age group of the vehicle
  - AGE: Vehicle age represent how old vehicle
  - RTO_REGISTRATION_YEAR: Year of vehicle registration

Write R Code for following statements:

A. Read (VEHICLE_PARK.csv) CSV file

```
vehicle_park <- read.csv("VEHICLE_PARK.csv")
```

B. Display top 20 and bottom 55 records using head and tail command

```
head(vehicle_park,20)
tail(vehicle_park, 55)
```

C. Display quick summary of all the columns.

```
summary(vehicle_park)
```

D. Display all the vehicles which have been registered on Year 2010

```
vehicle_park[vehicle_park$RTO_REGISTRATION_YEAR == 2010, ]
```

E. Display AGE_GROUP single columns from Vehicle Park data.

```
age_group <- vehicle_park$AGE_GROUP
```