**Instructions:**

Please share your answers filled in line in the Word document. Submit code separately wherever applicable.

Please ensure you update all the details:

**Name: Batch ID:** 4/08/2023

**Topic: Introduction to Database**

1. Create a Supermart\_DB with the tables created from the datasets shared (Customer.csv, Sales.csv and Product.csv files)
   1. Create a new database in your database management system, and name it Supermart\_DB.
   2. Create a new table called "customers" in the Supermart\_DB database
   3. Load the data from the Customer.csv file into the customers table
   4. Create a new table called "products" in the Supermart\_DB database
   5. Load the data from the Product.csv file into the products table
   6. Create a new table called "sales" in the Supermart\_DB database
   7. Load the data from the Sales.csv file into the sales table

**ANSWERS**

1. CREATE DATABASE SUPERMART\_DB;

USE SUPERMART\_DB;

1. CREATE TABLE CUSTOMER(CUSTOMER\_ID VARCHAR(50) PRIMARY KEY, CUSTOMER\_NAME VARCHAR(50),

SEGMENT VARCHAR(50), AGE INT, COUNTRY VARCHAR(50),

CITY VARCHAR(50), STATE CHAR(20), POSTAL\_CODE INT, REGION VARCHAR (20));

SHOW VARIABLES LIKE 'SECURE\_FILE\_PRIV';

SHOW VARIABLES LIKE '%LOCAL%';

1. LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Customer.csv'

INTO TABLE Customer

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '/N'

IGNORE 1 ROWS;

DESC CUSTOMER;

SELECT \* FROM CUSTOMER LIMIT 10;

1. create table product(product\_id varchar(18),

category varchar(20), sub\_category varchar(20), product\_name text);

1. LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Product.csv'

INTO TABLE product

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '/N'

IGNORE 1 ROWS;

select \* from product;

desc product;

1. CREATE TABLE SALES(ORDER\_LINE INT PRIMARY KEY, ORDER\_DATE DATE, SHIP\_DATE DATE,

CUSTOMER\_ID VARCHAR(20), PRODUCT\_ID VARCHAR(30), SALES float(4),

QUANTITY INT, DISCOUNT FLOAT(4), PROFIT FLOAT(8));

1. LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Sales.csv'

INTO TABLE SALES

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '/N'

IGNORE 1 ROWS;

select \* from sales limit 10;

desc sales;

**SELECTION OPERATORS:- (FILTERING):- in, like, between**

**Note:** use products, customers and sales table

1. Define the relationship between the tables using constraints/keys.
2. In the database Supermart \_DB, find the following:
3. Get the list of all the cities where the region is north or east without any duplicates using the IN statement.
4. Get the list of all orders where the ‘sales’ value is between 100 and 500 using the BETWEEN operator.
5. Get the list of customers whose last name contains only 4 characters using LIKE.

ANSWERS

1)

1. Primary Key (PK) :

- A primary key is a column or a set of columns in a table that uniquely identifies each row in that table.

- It enforces the uniqueness and ensures that there are no duplicate rows.

- Primary keys are used to establish a link between tables in a one-to-many or many-to-many relationship.

- Example: `employee\_id` in an "Employees" table.

2. Foreign Key (FK) :

- A foreign key is a column or a set of columns in one table that refers to the primary key of another table.

- It enforces referential integrity, ensuring that values in the foreign key column(s) exist in the referenced table's primary key column(s).

- Foreign keys are used to create relationships between tables, such as one-to-many and many-to-many relationships.

- Example: `department\_id` in an "Employees" table referring to the `department\_id` in a "Departments" table.

3. Unique Constraint :

- A unique constraint ensures that values in a column (or a combination of columns) are unique across all rows in the table.

- It allows for uniqueness but does not create a relationship between tables.

- Example: A unique constraint on the `email` column in a "Customers" table.

4. Check Constraint :

- A check constraint defines a condition that must be true for each row in the table.

- It is used to ensure that data in a column adheres to specific rules or conditions.

- Check constraints are not typically used to define relationships but rather to enforce data integrity rules.

- Example: Ensuring that the `age` column in an "Employees" table is greater than or equal to 18.

5. Composite Key :

- A composite key consists of two or more columns that together form a unique identifier for each row in a table.

- It is used when a single column cannot uniquely identify rows, but the combination of columns does.

- Composite keys are often used in junction tables for many-to-many relationships.

6. Primary Key and Foreign Key Combination :

- In a one-to-one or one-to-many relationship, the primary key of one table becomes a foreign key in another table.

- This establishes a direct relationship between the two tables.

7. Many-to-Many Relationships :

- Many-to-many relationships are established using junction tables.

- The junction table typically contains two or more foreign keys, each referring to a primary key in other tables.

- Example: A junction table linking "Students" and "Courses" tables to represent which students are enrolled in which courses.

2A) SELECT DISTINCT CITY FROM CUSTOMER WHERE REGION IN ('EAST','NORTH');

2B) ANSWER; select \* from sales where Sales between 100 and 500 order by sales;

Output;

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Order id |  |  |  |  |  | sales |  |  |  |
| 1729 | CA-2016-106894 | 07-02-2016 | 07-02-2016 | Same Day | CA-12265 | TEC-AC-10003063 | 100 | 4 | 0 | 21 |
| 4034 | CA-2016-169957 | 26-09-2016 | 30-09-2016 | Standard Class | SN-20710 | TEC-AC-10003063 | 100 | 4 | 0 | 21 |
| 5624 | CA-2016-121748 | 24-10-2016 | 28-10-2016 | Second Class | VW-21775 | TEC-AC-10003063 | 100 | 4 | 0 | 21 |
| 330 | US-2016-141544 | 30-08-2016 | 01-09-2016 | First Class | PO-18850 | OFF-LA-10001074 | 100.24 | 10 | 0.2 | 33.831 |
| 6473 | CA-2014-151897 | 06-06-2014 | 10-06-2014 | Standard Class | VT-21700 | OFF-LA-10001074 | 100.24 | 10 | 0.2 | 33.831 |

**2C)** SELECT CUSTOMER\_NAME from CUSTOMER WHERE CUSTOMER\_NAME LIKE '% \_\_\_\_' ORDER BY CUSTOMER\_NAME;

**SELECTION OPERATORS:- ordering**

1. Retrieve all orders where the ‘discount’ value is greater than zero ordered in descending order basis ‘discount’ value
2. Limit the number of results in the above query to the top 10.

**Answers;**

1a). SELECT \* FROM SALES WHERE DISCOUNT >0 ORDER BY DISCOUNT desc;

2a). SELECT \* FROM SALES WHERE DISCOUNT >0 ORDER BY DISCOUNT desc LIMIT 10 ;

**Aggregate operators:-**

1. Find the sum of all ‘sales’ values.
2. Find count of the number of customers in the north region with ages between 20 and 30
3. Find the average age of east region customers
4. Find the minimum and maximum aged customers from Philadelphia

Answers;

1. SELECT sum(SALES) FROM SALES;
2. SELECT COUNT(CUSTOMER\_NAME) FROM CUSTOMER WHERE REGION = 'NORTH' AND AGE between 20 AND 30;
3. SELECT AVG(AGE) FROM CUSTOMER WHERE REGION = "EAST";
4. SELECT MIN(AGE), MAX(AGE) FROM CUSTOMER WHERE CITY = 'PHILADELPHIA' ;

**GROUP BY OPERATORS:-**

1. Create a display with the information below for each product ID.
2. Total sales (in $) order by this column in descending
3. Total sales quantity
4. The number of orders
5. Max Sales value
6. Min Sales value
7. Average sales value
8. Get the list of product ID’s where the quantity of product sold is greater than 10

**ANSWERS; 1**

1. select product\_id , sales from sales order by sales desc;
2. select sum(quantity) from sales;
3. select distinct count(order\_id) from sales;
4. select max(sales) from sales;
5. select min(sales) from sales;
6. select avg(sales) from sales;

2) select product\_id, quantity from sales where quantity >10 order by quantity desc;