ASSIGNMENT-2
RELATIONAL DATABASES AND SQL
NAME: BHAVYA BHARDWAJ
BLEND ALL STARS

Q)

Design and Query a Retail Sales Database
Goal:
Convert the dataset into a relational database schema and perform analytical queries.
Tasks:
Design tables: Customers, Products, Orders, OrderDetails.
Normalize to 3NF and draw an ER diagram.
Write queries for:
Top-selling product by month.
Sales by region using JOIN + GROUP BY.
Customers with total spend > threshold (use subqueries).
(Optional) Implement using SQLite or MySQL.
Tech: SQL, MySQL / PostgreSQL / SQLite
Deliverables: .sql schema, ER diagram, query outputs + screenshots (ER diagram, query results, console output)

PS. I DID MY WORK ON THE CSV FILE AMAZON SALE REPORT.CSV AS IT CONTAINED ALL THE REQUIRED DATA

```
mysql> CREATE DATABASE retail;
Query OK, 1 row affected (0.033 sec)

mysql> USE retail;
Database changed
mysql> CREATE TABLE staging_amazon_sales (
    ->     idx INT,
    ->     order_id VARCHAR(50),
    ->     order_date VARCHAR(20),
    ->     status VARCHAR(100),
    ->     fulfilment VARCHAR(100),
    ->     sales_channel VARCHAR(100),
    ->     ship_service_level VARCHAR(100),
    ->     style VARCHAR(100),
    ->     sku VARCHAR(100),
    ->     category VARCHAR(100),
    ->     size VARCHAR(50),
    ->     asin VARCHAR(50),
    ->     courier_status VARCHAR(100),
    ->     qty INT,
    ->     currency VARCHAR(10),
    ->     amount DECIMAL(10,2),
    ->     ship_city VARCHAR(100),
    ->     ship_state VARCHAR(100),
    ->     ship_postal VARCHAR(50),
    ->     ship_country VARCHAR(10),
    ->     promotion_ids TEXT,
    ->     b2b VARCHAR(10),
    ->     fulfilled_by VARCHAR(100),
    ->     dummy VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.037 sec)
```

CREATING DATABASE

```
mysql> USE retail;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> LOAD DATA LOCAL INFILE '/Users/bhavya/Downloads/Sales Dataset-2/Amazon Sale Report.csv'
    -> INTO TABLE staging_amazon_sales
    -> FIELDS TERMINATED BY ','
    -> ENCLOSED BY '"'
    -> IGNORE 1 ROWS;
Query OK, 128975 rows affected, 7795 warnings (2.655 sec)
Records: 128975  Deleted: 0  Skipped: 0  Warnings: 7795
```

LOADING THE DATA FROM THE CSV FILE FROM OUR COMPUTER AND PUTTING IT INTO staging_amazon_sale

```
mysql>
mysql> CREATE TABLE Customers(
    ->     customer_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     ship_city VARCHAR(100),
    ->     ship_state VARCHAR(100),
    ->     ship_country VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.027 sec)

mysql>
mysql> CREATE TABLE Products(
    ->     product_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     SKU VARCHAR(50),
    ->     ASIN VARCHAR(50),
    ->     Category VARCHAR(50),
    ->     Size VARCHAR(20)
    -> );
Query OK, 0 rows affected (0.013 sec)

mysql>
mysql> CREATE TABLE Orders(
    ->     order_id VARCHAR(50) PRIMARY KEY,
    ->     order_date DATE,
    ->     Status VARCHAR(100),
    ->     Sales_Channel VARCHAR(50),
    ->     Fulfilment VARCHAR(50),
    ->     ship_service_level VARCHAR(50),
    ->     B2B BOOLEAN,
    ->     fulfilled_by VARCHAR(50),
    ->     customer_id INT,
    ->     FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
    -> );
Query OK, 0 rows affected (0.021 sec)

mysql>
mysql> CREATE TABLE OrderDetails(
    ->     order_detail_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     order_id VARCHAR(50),
    ->     product_id INT,
    ->     Qty INT,
    ->     currency VARCHAR(10),
    ->     Amount DECIMAL(10,2),
    ->     FOREIGN KEY(order_id) REFERENCES Orders(order_id),
    ->     FOREIGN KEY(product_id) REFERENCES Products(product_id)
    -> );
Query OK, 0 rows affected (0.023 sec)
```

Creating Customer ,Product,Orders,OrderDetail table with primary key and foreign key

```
mysql> INSERT INTO Customers (ship_city, ship_state, ship_country)
    -> SELECT DISTINCT ship_city, ship_state, ship_country
    -> FROM staging_amazon_sales
    -> WHERE ship_city IS NOT NULL;
Query OK, 7400 rows affected (0.242 sec)
Records: 7400  Duplicates: 0  Warnings: 0

mysql> INSERT INTO Products (SKU, ASIN, Category, Size)
    -> SELECT DISTINCT SKU, ASIN, Category, Size
    -> FROM staging_amazon_sales
    -> WHERE SKU IS NOT NULL;
Query OK, 7200 rows affected (0.196 sec)
Records: 7200  Duplicates: 0  Warnings: 0
```

Inserting into Customer and Product and OrderDetails using 3NF NORMALISATION ,also using JOIN AND GROUP BY KEY WORDS

```
mysql> INSERT INTO OrderDetails(order_id, product_id, quantity, item_subtotal)
    -> SELECT
    ->     s.order_id,
    ->     p.product_id,
    ->     s.qty,
    ->     s.item_subtotal
    -> FROM staging_amazon_sales s
    -> JOIN Products p
    ->     ON p.sku = s.sku;
ERROR 1054 (42S22): Unknown column 'item_subtotal' in 'field list'
mysql> INSERT INTO OrderDetails(order_id, product_id, quantity, Amount)
    -> SELECT
    ->     s.order_id,
    ->     p.product_id,
    ->     s.qty,
    ->     s.amount
    -> FROM staging_amazon_sales s
    -> JOIN Products p
    ->     ON p.sku = s.sku;
Query OK, 129077 rows affected (1.792 sec)
Records: 129077  Duplicates: 0  Warnings: 0
```

OUTPUT

```
mysql> SELECT COUNT(*) FROM OrderDetails;
+----------+
| COUNT(*) |
+----------+
|   129077 |
+----------+
1 row in set (0.040 sec)

mysql> SELECT COUNT(*) FROM staging_amazon_sales;
+----------+
| COUNT(*) |
+----------+
|   128975 |
+----------+
1 row in set (0.058 sec)

mysql> SELECT COUNT(*) FROM OrderDetails;
+----------+
| COUNT(*) |
+----------+
|   129077 |
+----------+
1 row in set (0.037 sec)

mysql> SELECT COUNT(DISTINCT order_id) FROM Orders;
+--------------------------+
| COUNT(DISTINCT order_id) |
+--------------------------+
|                   120378 |
+--------------------------+
1 row in set (0.283 sec)
```

Write queries for:
Top-selling product by month.

```
mysql> SELECT
    ->     DATE_FORMAT(o.order_date, '%Y-%m') AS Month,
    ->     p.product_desc,
    ->     SUM(od.Quantity) AS Total_Quantity
    -> FROM OrderDetails od
    -> JOIN Orders o ON od.order_id = o.order_id
    -> JOIN Products p ON p.product_id = od.product_id
    -> GROUP BY Month, p.product_desc
    -> ORDER BY Month, Total_Quantity DESC;
+---------+-------------+----------------+
| Month   | product_desc | Total_Quantity |
+---------+-------------+----------------+
| 2022-03 | JNE3160      |              6 |
| 2022-03 | JNE3405      |              5 |
| 2022-03 | J0003        |              5 |
| 2022-03 | JNE3373      |              3 |
| 2022-03 | SET268       |              3 |
| 2022-03 | JNE3368      |              3 |
| 2022-03 | J0230        |              3 |
| 2022-03 | SET044       |              3 |
| 2022-03 | SET073       |              3 |
| 2022-03 | SET172       |              3 |
| 2022-03 | SET187       |              2 |
| 2022-03 | SET183       |              2 |
| 2022-03 | PJNE3373     |              2 |
| 2022-03 | J0341        |              2 |
| 2022-03 | JNE3567      |              2 |
| 2022-03 | JNE3546      |              2 |
| 2022-03 | SET264       |              2 |
| 2022-03 | JNE3721      |              2 |
| 2022-03 | SET239       |              2 |
| 2022-03 | JNE3797      |              2 |
| 2022-03 | J0119        |              2 |
| 2022-03 | J0346        |              2 |
| 2022-03 | JNE3724      |              2 |
| 2022-03 | J0127        |              2 |
| 2022-03 | JNE3291      |              2 |
| 2022-03 | JNE2270      |              2 |
| 2022-03 | J0212        |              2 |
| 2022-03 | J0328        |              2 |
| 2022-03 | J0151        |              1 |
| 2022-03 | J0281        |              1 |
| 2022-03 | J0126        |              1 |
| 2022-03 | SET182       |              1 |
| 2022-03 | JNE3510      |              1 |
| 2022-03 | JNE3686      |              1 |
| 2022-03 | SET252       |              1 |
| 2022-03 | JNE3684      |              1 |
| 2022-03 | JNE3457      |              1 |
| 2022-03 | JNE3415      |              1 |
| 2022-03 | SET132       |              1 |
| 2022-03 | PSET264      |              1 |
| 2022-03 | JNE3758      |              1 |
| 2022-03 | BL111        |              1 |
| 2022-03 | SET317       |              1 |
| 2022-03 | JNE3528      |              1 |
| 2022-03 | J0345        |              1 |
| 2022-03 | J0299        |              1 |
| 2022-03 | JNE3633      |              1 |
| 2022-03 | JNE3440      |              1 |
```

Query and Output

Sales by region using JOIN + GROUP BY.

```
mysql> SELECT
    ->     c.ship_state AS Region,
    ->     SUM(od.Amount) AS Total_Sales
    -> FROM OrderDetails od
    -> JOIN Orders o ON od.order_id = o.order_id
    -> JOIN Customers c ON c.customer_id = o.customer_id
    -> GROUP BY Region
    -> ORDER BY Total_Sales DESC;
+------------------------+-------------+
| Region                 | Total_Sales |
+------------------------+-------------+
| MAHARASHTRA            | 13349360.14 |
| KARNATAKA              | 10489439.51 |
| TELANGANA              |  6920802.65 |
| UTTAR PRADESH          |  6820096.08 |
| TAMIL NADU             |  6522107.11 |
| DELHI                  |  4347125.46 |
| KERALA                 |  3832461.58 |
| WEST BENGAL            |  3511348.44 |
| ANDHRA PRADESH         |  3222639.72 |
| HARYANA                |  2882883.99 |
| Gujarat                |  2732245.82 |
| RAJASTHAN              |  1759694.16 |
| MADHYA PRADESH         |  1592382.98 |
| BIHAR                  |  1416521.32 |
| ODISHA                 |  1386250.39 |
| PUNJAB                 |  1211961.84 |
| ASSAM                  |  1019284.20 |
| UTTARAKHAND            |   974441.55 |
| JHARKHAND              |   919676.21 |
| Goa                    |   637685.85 |
| CHHATTISGARH           |   570485.83 |
| HIMACHAL PRADESH       |   503364.51 |
| JAMMU & KASHMIR        |   456932.74 |
| MANIPUR                |   214335.99 |
| CHANDIGARH             |   211740.67 |
| PUDUCHERRY             |   192632.24 |
| ANDAMAN & NICOBAR      |   158723.62 |
| NAGALAND               |   144094.67 |
| SIKKIM                 |   140828.66 |
| MEGHALAYA              |   119871.81 |
| ARUNACHAL PRADESH      |    97772.00 |
| TRIPURA                |    92548.40 |
| New Delhi              |    47109.95 |
| DADRA AND NAGAR        |    42138.92 |
| MIZORAM                |    41948.71 |
| LADAKH                 |    38388.43 |
|                        |    18671.00 |
| LAKSHADWEEP            |     3175.29 |
| orissa                 |     1737.00 |
| Rajshthan              |     1126.00 |
| RJ                     |     1040.00 |
| rajsthan               |      964.00 |
| NL                     |      961.00 |
| Punjab/Mohali/Zirakpur |      568.00 |
| Pondicherry            |      529.00 |
| AR                     |      493.00 |
| PB                     |      399.00 |
| APO                    |        0.00 |
+------------------------+-------------+
48 rows in set (1.018 sec)
```

Query and Output

Customers with total spend > threshold (use subqueries).

```
mysql> SELECT customer_id, total_spend
    -> FROM (
    ->         SELECT
    ->             o.customer_id,
    ->             SUM(od.Amount) AS total_spend
    ->         FROM Orders o
    ->         JOIN OrderDetails od ON o.order_id = od.order_id
    ->         GROUP BY o.customer_id
    ->       ) AS t
    -> WHERE total_spend > 1000;   -- threshold
+-------------+-------------+
| customer_id | total_spend |
+-------------+-------------+
|        8192 |    76208.05 |
|        8193 |   102713.57 |
|        8194 |   122567.91 |
|        8195 |    21424.38 |
|        8196 |   114935.67 |
|        8197 |     2205.00 |
|        8198 |    44796.62 |
|        8199 |   176447.17 |
|        8200 |    95895.96 |
|        8201 |    83222.76 |
|        8202 |    77796.76 |
|        8204 |   162194.02 |
|        8205 |    10247.00 |
|        8206 |    77540.91 |
|        8207 |    55722.76 |
|        8208 |     6475.00 |
|        8209 |    76882.76 |
|        8210 |     3286.00 |
|        8211 |    53692.71 |
|        8212 |   274890.27 |
|        8213 |    11066.62 |
|        8214 |    23090.23 |
|        8215 |    70555.91 |
|        8216 |    40248.67 |
|        8217 |    38086.96 |
|        8218 |   333519.44 |
|        8219 |    33471.29 |
|        8221 |   144904.35 |
|        8222 |    43747.76 |
|        8223 |     4774.00 |
|        8224 |    92919.15 |
|        8225 |   114332.67 |
|        8226 |    39973.24 |
|        8227 |   171759.75 |
|        8228 |    24869.55 |
```