Business Insights Database

Goal: Create a database to analyze sales and customer behavior for decision-making.

Tables:

- `customers(customer_id, age, region, income_level)`
- `transactions(transaction_id, customer_id, product_id, amount, date)`
- `products(product_id, category, price)`

Queries:

- Aggregate sales by region and income level.
- Identify top 5 products contributing to revenue growth.

a. Create database

```
CREATE DATABASE business_insights;
USE business_insights;
```

```
mysql> CREATE DATABASE business_insights;
Query OK, 1 row affected (0.10 sec)

mysql> USE business_insights;
Database changed
```

b. Create tables

```
-- Customer information
CREATE TABLE customers (
    customer_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    age INT,
    region VARCHAR(50),
    income_level VARCHAR(50)
);

-- Product catalog
CREATE TABLE products (
    product_id INT AUTO_INCREMENT PRIMARY KEY,
    category VARCHAR(50),
    price DECIMAL(10,2)
);

-- Transactions
CREATE TABLE transactions (
    transaction_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_id INT,
    product_id INT,
    amount DECIMAL(10,2),
    date DATE,
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id),
    FOREIGN KEY (product_id) REFERENCES products(product_id)
);
```

```
mysql> -- Customer information
mysql> CREATE TABLE customers (
    ->     customer_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(100),
    ->     age INT,
    ->     region VARCHAR(50),
    ->     income_level VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.09 sec)

mysql>
mysql> -- Product catalog
mysql> CREATE TABLE products (
    ->     product_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     category VARCHAR(50),
    ->     price DECIMAL(10,2)
    -> );
Query OK, 0 rows affected (0.09 sec)

mysql>
mysql> -- Transactions
mysql> CREATE TABLE transactions (
    ->     transaction_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     customer_id INT,
    ->     product_id INT,
    ->     amount DECIMAL(10,2),
    ->     date DATE,
    ->     FOREIGN KEY (customer_id) REFERENCES customers(customer_id),
    ->     FOREIGN KEY (product_id) REFERENCES products(product_id)
    -> );
Query OK, 0 rows affected (0.15 sec)
```

c. Insert sample data

```
INSERT INTO customers (name, age, region, income_level)
VALUES
('Alice', 28, 'Jakarta', 'High'),
('Bob', 35, 'Bandung', 'Medium'),
('Charlie', 22, 'Surabaya', 'Low');

INSERT INTO products (category, price)
VALUES
('Electronics', 500.00),
('Groceries', 50.00),
('Clothing', 120.00);

INSERT INTO transactions (customer_id, product_id, amount, date)
VALUES
(1, 1, 500.00, '2025-09-01'),
(2, 2, 50.00, '2025-09-02'),
(3, 3, 120.00, '2025-09-03'),
(1, 2, 50.00, '2025-09-04'),
(2, 1, 500.00, '2025-09-05');
```

```
mysql> INSERT INTO customers (name, age, region, income_level)
    -> VALUES
    -> ('Alice', 28, 'Jakarta', 'High'),
    -> ('Bob', 35, 'Bandung', 'Medium'),
    -> ('Charlie', 22, 'Surabaya', 'Low');
Query OK, 3 rows affected (0.05 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql>
mysql> INSERT INTO products (category, price)
    -> VALUES
    -> ('Electronics', 500.00),
    -> ('Groceries', 50.00),
    -> ('Clothing', 120.00);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql>
mysql> INSERT INTO transactions (customer_id, product_id, amount, date)
    -> VALUES
    -> (1, 1, 500.00, '2025-09-01'),
    -> (2, 2, 50.00, '2025-09-02'),
    -> (3, 3, 120.00, '2025-09-03'),
    -> (1, 2, 50.00, '2025-09-04'),
    -> (2, 1, 500.00, '2025-09-05');
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

d. Example queries

```
-- Total sales by region
SELECT region, SUM(amount) AS total_sales
FROM customers c
JOIN transactions t ON c.customer_id = t.customer_id
GROUP BY region;

-- Top 3 products by revenue
SELECT p.category, SUM(t.amount) AS revenue
FROM products p
JOIN transactions t ON p.product_id = t.product_id
GROUP BY p.category
ORDER BY revenue DESC
LIMIT 3;

-- Average spending by income level
SELECT income_level, AVG(amount) AS avg_spending
FROM customers c
JOIN transactions t ON c.customer_id = t.customer_id
GROUP BY income_level;
```

```
mysql> -- Total sales by region
mysql> SELECT region, SUM(amount) AS total_sales
    -> FROM customers c
    -> JOIN transactions t ON c.customer_id = t.customer_id
    -> GROUP BY region;
+----------+-------------+
| region   | total_sales |
+----------+-------------+
| Jakarta  |      550.00 |
| Bandung  |      550.00 |
| Surabaya |      120.00 |
+----------+-------------+
3 rows in set (0.03 sec)

mysql>
mysql> -- Top 3 products by revenue
mysql> SELECT p.category, SUM(t.amount) AS revenue
    -> FROM products p
    -> JOIN transactions t ON p.product_id = t.product_id
    -> GROUP BY p.category
    -> ORDER BY revenue DESC
    -> LIMIT 3;
+-------------+---------+
| category    | revenue |
+-------------+---------+
| Electronics | 1000.00 |
| Clothing    |  120.00 |
| Groceries   |  100.00 |
+-------------+---------+
3 rows in set (0.00 sec)
```

```
mysql> -- Average spending by income level
mysql> SELECT income_level, AVG(amount) AS avg_spending
    -> FROM customers c
    -> JOIN transactions t ON c.customer_id = t.customer_id
    -> GROUP BY income_level;
+--------------+--------------+
| income_level | avg_spending |
+--------------+--------------+
| High         |   275.000000 |
| Medium       |   275.000000 |
| Low          |   120.000000 |
+--------------+--------------+
3 rows in set (0.01 sec)
```