E-COMMERCE DATA ANALYSIS USING MySQL

Objective: Use SQL queries to extract and analyze data from a database.

Overview:

This project demonstrates how to design and analyze an **E-commerce SQL database** using MySQL. It covers schema creation, sample data insertion, and running queries for insights such as customer orders, product sales, and revenue analysis.

Features

- Database schema with **users, products, categories, orders, order_items, reviews**
- Queries using **SELECT, WHERE, ORDER BY, GROUP BY**
- **JOINS** (INNER, LEFT, RIGHT) for relational analysis
- **Aggregates** (SUM, AVG, COUNT) for metrics
- **Subqueries** for advanced filtering
- **Views** for reusable analysis
- **Indexes & EXPLAIN** for query optimization

OUTPUT:

```
mysql> USE ecommerce;
Database changed
mysql> CREATE TABLE users (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NOT NULL, email VARCHAR(150) UNIQUE NOT NULL, created_at DATETIME DEFAULT CU
RRENT_TIMESTAMP) ENGINE=InnoDB;
Query OK, 0 rows affected (0.18 sec)

mysql> CREATE TABLE categories (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NOT NULL);
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE products (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(200) NOT NULL,
    -> category_id INT, price DECIMAL(10,2) NOT NULL DEFAULT 0.00, stock INT NOT NULL DEFAULT 0, created_at DATETIME DEFAULT CURRENT_TIMESTAMP, FOREIGN KEY
(category_id) REFERENCES categories(id) ON DELETE SET NULL);
Query OK, 0 rows affected (0.07 sec)

mysql> CREATE TABLE orders (id INT AUTO_INCREMENT PRIMARY KEY, user_id INT NOT NULL,
    -> order_date DATETIME DEFAULT CURRENT_TIMESTAMP, status ENUM('pending','shipped','delivered','cancelled') DEFAULT 'pending', total_amount DECIMAL(12,2)
 NOT NULL DEFAULT 0.00, FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE);
Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE order_items (id INT AUTO_INCREMENT PRIMARY KEY, order_id INT NOT NULL, product_id INT NOT NULL, quantity INT NOT NULL DEFAULT 1, unit_pr
ice DECIMAL(10,2) NOT NULL,
    -> FOREIGN KEY (order_id) REFERENCES orders(id) ON DELETE CASCADE, FOREIGN KEY (product_id) REFERENCES products(id) ON DELETE CASCADE);
Query OK, 0 rows affected (0.07 sec)

mysql> CREATE TABLE reviews (id INT AUTO_INCREMENT PRIMARY KEY, product_id INT NOT NULL, user_id INT NOT NULL, rating TINYINT NOT NULL CHECK (rating BETWEEN
 1 AND 5), review TEXT, created_at DATETIME DEFAULT CURRENT_TIMESTAMP, FOREIGN KEY (product_id) REFERENCES products(id) ON DELETE CASCADE, FOREIGN KEY (user
_id) REFERENCES users(id) ON DELETE CASCADE);
Query OK, 0 rows affected (0.09 sec)

mysql> CREATE INDEX idx_products_category ON products(category_id);
Query OK, 0 rows affected (0.09 sec)
```

```
mysql> CREATE INDEX idx_products_category ON products(category_id);
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> CREATE INDEX idx_orders_user ON orders(user_id);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> CREATE INDEX idx_orderitems_order ON order_items(order_id);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> CREATE INDEX idx_orderitems_product ON order_items(product_id);
Query OK, 0 rows affected (0.25 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> INSERT INTO users (name, email) VALUES
    -> ('Alice Johnson','alice@example.com'),
    -> ('Bob Kumar','bob@example.com'),
    -> ('Cara Lee','cara@example.com');
Query OK, 3 rows affected (0.03 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> INSERT INTO categories (name) VALUES ('Electronics'),('Home'),('Books');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

```
mysql> INSERT INTO reviews (product_id, user_id, rating, review) VALUES
    -> (1,1,5,'Excellent mouse'),
    -> (2,1,4,'Great sound for the price'),
    -> (4,3,5,'Well-written explanations');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from users;
+----+--------------+-----------------+---------------------+
| id | name         | email           | created_at          |
+----+--------------+-----------------+---------------------+
|  1 | Alice Johnson | alice@example.com | 2025-09-28 19:04:38 |
|  2 | Bob Kumar     | bob@example.com   | 2025-09-28 19:04:38 |
|  3 | Cara Lee      | cara@example.com  | 2025-09-28 19:04:38 |
+----+--------------+-----------------+---------------------+
3 rows in set (0.01 sec)

mysql> select * from categories;
+----+-------------+
| id | name        |
+----+-------------+
|  1 | Electronics |
|  2 | Home        |
|  3 | Books       |
+----+-------------+
3 rows in set (0.00 sec)

mysql> select * from products;
+----+----------------------+-------------+-------+-------+---------------------+
| id | name                 | category_id | price | stock | created_at          |
+----+----------------------+-------------+-------+-------+---------------------+
|  1 | Wireless Mouse       |           1 | 19.99 |   120 | 2025-09-28 19:05:05 |
|  2 | Bluetooth Speaker    |           1 | 49.99 |    45 | 2025-09-28 19:05:05 |
|  3 | Ceramic Mug          |           2 |  9.95 |   200 | 2025-09-28 19:05:05 |
|  4 | Data Structures Book |           3 | 39.90 |    30 | 2025-09-28 19:05:05 |
+----+----------------------+-------------+-------+-------+---------------------+
```

```
mysql> SELECT COUNT(*) AS users_count FROM users;
+-------------+
| users_count |
+-------------+
|           3 |
+-------------+
1 row in set (0.01 sec)

mysql> SELECT COUNT(*) AS products_count FROM products;
+----------------+
| products_count |
+----------------+
|              4 |
+----------------+
1 row in set (0.00 sec)

mysql> SELECT COUNT(*) AS orders_count FROM orders;
+--------------+
| orders_count |
+--------------+
|            3 |
+--------------+
1 row in set (0.00 sec)

mysql> SELECT COUNT(*) AS order_items_count FROM order_items;
+-------------------+
| order_items_count |
+-------------------+
|                 4 |
+-------------------+
1 row in set (0.00 sec)

mysql> SELECT COUNT(*) AS order_items_count FROM order_items;
+-------------------+
| order_items_count |
+-------------------+
```

```
mysql> SELECT id, user_id, order_date, total_amount FROM orders
    -> WHERE order_date >= '2025-07-01' AND status <> 'cancelled'
    -> ORDER BY order_date DESC;
+----+---------+---------------------+--------------+
| id | user_id | order_date          | total_amount |
+----+---------+---------------------+--------------+
|  3 |       1 | 2025-08-01 09:00:00 |        39.90 |
|  2 |       2 | 2025-07-11 12:00:00 |         9.95 |
|  1 |       1 | 2025-07-10 10:25:00 |        69.98 |
+----+---------+---------------------+--------------+
3 rows in set (0.02 sec)

mysql> SELECT oi.id AS order_item_id, o.id AS order_id, u.name AS customer, p.name AS product,
    -> oi.quantity, oi.unit_price, o.order_date
    -> FROM order_items oi
    -> INNER JOIN orders o ON oi.order_id = o.id
    -> INNER JOIN products p ON oi.product_id = p.id
    -> INNER JOIN users u ON o.user_id = u.id
    -> ORDER BY o.order_date DESC;
+---------------+----------+---------------+----------------------+----------+------------+---------------------+
| order_item_id | order_id | customer      | product              | quantity | unit_price | order_date          |
+---------------+----------+---------------+----------------------+----------+------------+---------------------+
|             4 |        3 | Alice Johnson | Data Structures Book |        1 |      39.90 | 2025-08-01 09:00:00 |
|             3 |        2 | Bob Kumar     | Ceramic Mug          |        1 |       9.95 | 2025-07-11 12:00:00 |
|             1 |        1 | Alice Johnson | Wireless Mouse       |        1 |      19.99 | 2025-07-10 10:25:00 |
|             2 |        1 | Alice Johnson | Bluetooth Speaker    |        1 |      49.99 | 2025-07-10 10:25:00 |
+---------------+----------+---------------+----------------------+----------+------------+---------------------+
4 rows in set (0.01 sec)

mysql> SELECT p.id, p.name, p.price, AVG(r.rating) AS avg_rating, COUNT(r.id) AS num_reviews
    -> FROM products p
    -> LEFT JOIN reviews r ON p.id = r.product_id
    -> GROUP BY p.id, p.name, p.price
    -> ORDER BY avg_rating DESC;
+----+----------------------+-------+------------+-------------+
| id | name                 | price | avg_rating | num_reviews |
+----+----------------------+-------+------------+-------------+
|  1 | Wireless Mouse       | 19.99 |     5.0000 |           1 |
```

```
mysql> SELECT p.id, p.name, SUM(oi.quantity * oi.unit_price) AS total_sales,
    -> SUM(oi.quantity) AS total_units_sold
    -> FROM order_items oi
    -> JOIN products p ON oi.product_id = p.id
    -> GROUP BY p.id, p.name
    -> ORDER BY total_sales DESC;
+----+----------------------+-------------+------------------+
| id | name                 | total_sales | total_units_sold |
+----+----------------------+-------------+------------------+
|  2 | Bluetooth Speaker    |       49.99 |                1 |
|  4 | Data Structures Book |       39.90 |                1 |
|  1 | Wireless Mouse       |       19.99 |                1 |
|  3 | Ceramic Mug          |        9.95 |                1 |
+----+----------------------+-------------+------------------+
4 rows in set (0.01 sec)

mysql> SELECT u.id, u.name, COUNT(o.id) AS orders_count, AVG(o.total_amount) AS avg_order_value,
    -> SUM(o.total_amount) AS lifetime_value
    -> FROM users u
    -> LEFT JOIN orders o ON u.id = o.user_id
    -> GROUP BY u.id, u.name
    -> ORDER BY lifetime_value DESC;
+----+---------------+--------------+-----------------+----------------+
| id | name          | orders_count | avg_order_value | lifetime_value |
+----+---------------+--------------+-----------------+----------------+
|  1 | Alice Johnson |            2 |       54.940000 |         109.88 |
|  2 | Bob Kumar     |            1 |        9.950000 |           9.95 |
|  3 | Cara Lee      |            0 |            NULL |           NULL |
+----+---------------+--------------+-----------------+----------------+
3 rows in set (0.00 sec)

mysql> SELECT DATE_FORMAT(order_date, '%Y-%m') AS month, SUM(total_amount) AS monthly_revenue, COUNT(id) AS orders_count
    -> FROM orders
    -> GROUP BY month
    -> ORDER BY month;
+---------+-----------------+--------------+
| month   | monthly_revenue | orders_count |
```

```
mysql> SELECT * FROM products p
    -> WHERE p.id NOT IN (SELECT DISTINCT product_id FROM order_items);
Empty set (0.01 sec)

mysql> SELECT * FROM products p
    -> WHERE p.id NOT IN (SELECT DISTINCT product_id FROM order_items);
Empty set (0.00 sec)

mysql> SELECT u.id, u.name, (
    -> SELECT IFNULL(SUM(o2.total_amount),0) FROM orders o2 WHERE o2.user_id = u.id
    -> ) AS user_spent
    -> FROM users u
    -> WHERE (
    -> SELECT IFNULL(SUM(o3.total_amount),0) FROM orders o3 WHERE o3.user_id = u.id
    -> ) > (
    -> SELECT AVG(total_spent) FROM (
    -> SELECT SUM(total_amount) AS total_spent FROM orders GROUP BY user_id
    -> ) as per_user
    -> );
+------+---------------+------------+
| id   | name          | user_spent |
+------+---------------+------------+
|    1 | Alice Johnson |     109.88 |
+------+---------------+------------+
1 row in set (0.01 sec)

mysql> SELECT c.id AS category_id, c.name AS category, p.id AS product_id, p.name AS product_name, p.price
    -> FROM products p
    -> JOIN categories c ON p.category_id = c.id
    -> WHERE p.id = (
    -> SELECT oi.product_id FROM order_items oi
    -> JOIN products p2 ON oi.product_id = p2.id
    -> WHERE p2.category_id = c.id
    -> GROUP BY oi.product_id
    -> ORDER BY SUM(oi.quantity * oi.unit_price) DESC
    -> LIMIT 1
    -> );
+-------------+-------------+------------+---------------------+-------+
| category_id | category    | product_id | product_name        | price |
```

```
mysql> CREATE OR REPLACE VIEW vw_product_sales AS
    -> SELECT p.id AS product_id, p.name, p.category_id,
    -> SUM(oi.quantity * oi.unit_price) AS total_sales,
    -> SUM(oi.quantity) AS total_units
    -> FROM products p
    -> LEFT JOIN order_items oi ON p.id = oi.product_id
    -> GROUP BY p.id, p.name, p.category_id;
Query OK, 0 rows affected (0.02 sec)

mysql> SELECT * FROM vw_product_sales ORDER BY total_sales DESC LIMIT 10;
+------------+---------------------+-------------+-------------+-------------+
| product_id | name                | category_id | total_sales | total_units |
+------------+---------------------+-------------+-------------+-------------+
|          2 | Bluetooth Speaker   |           1 |       49.99 |           1 |
|          4 | Data Structures Book |           3 |       39.90 |           1 |
|          1 | Wireless Mouse      |           1 |       19.99 |           1 |
|          3 | Ceramic Mug         |           2 |        9.95 |           1 |
+------------+---------------------+-------------+-------------+-------------+
4 rows in set (0.01 sec)

mysql> ALTER TABLE products ADD INDEX idx_products_cat_price (category_id, price);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE order_items ADD INDEX idx_orderitems_order_product (order_id, product_id);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE reviews ADD FULLTEXT INDEX ft_reviews_text (review);
Query OK, 0 rows affected, 1 warning (0.34 sec)
Records: 0  Duplicates: 0  Warnings: 1
```