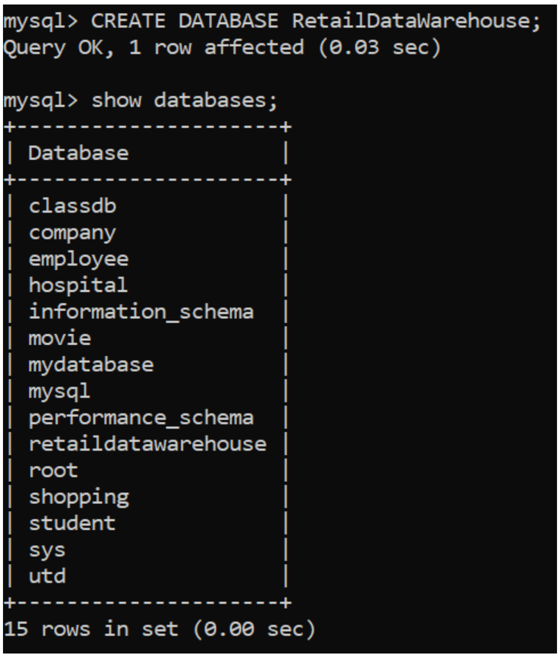
Experiment 2

Aim: Students will design and implement a star schema data model for a given business scenario, focusing on creating fact and dimension tables.

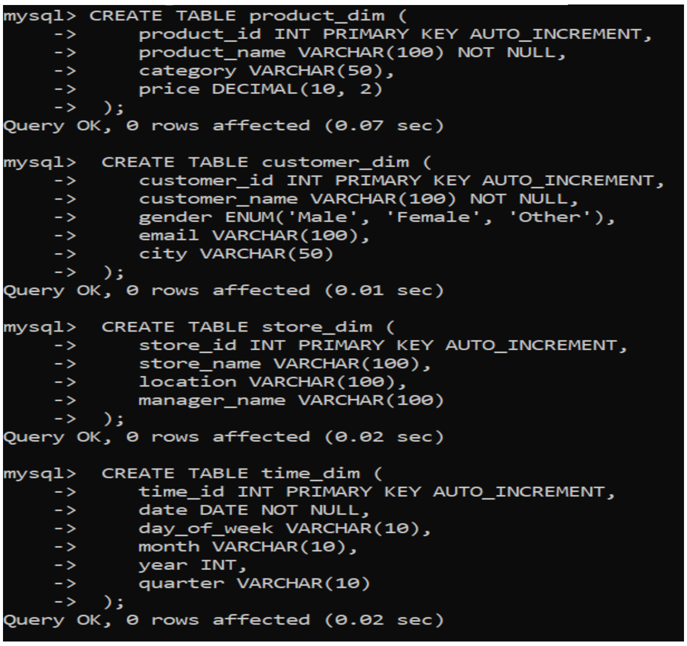
Creating Database and Using in MySQL



```
mysql> CREATE DATABASE RetailDataWarehouse;
Query OK, 1 row affected (0.03 sec)

mysql> show databases;
+---------------------+
| Database            |
+---------------------+
| classdb             |
| company             |
| employee            |
| hospital            |
| information_schema  |
| movie               |
| mydatabase          |
| mysql               |
| performance_schema  |
| retaildatawarehouse |
| root                |
| shopping            |
| student             |
| sys                 |
| utd                 |
+---------------------+
15 rows in set (0.00 sec)
```

Fig 1: Database in MySQL

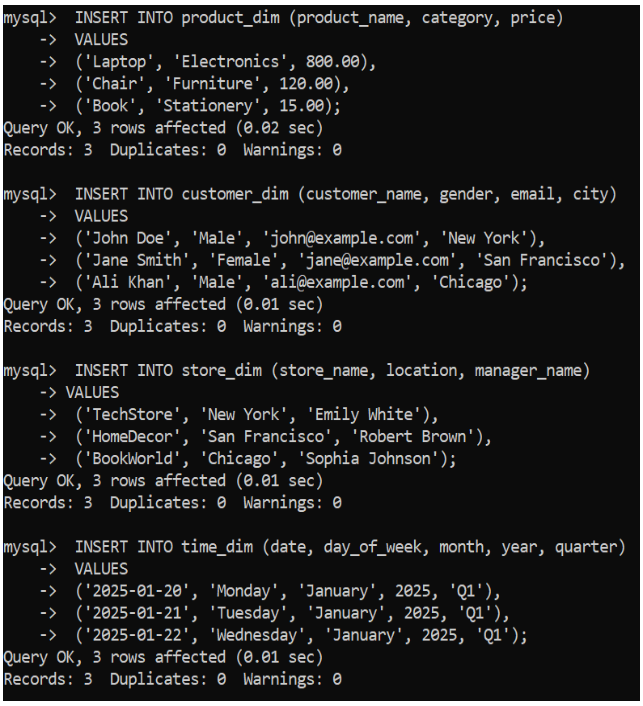
Creating Tables



```
mysql> CREATE TABLE product_dim (
    ->     product_id INT PRIMARY KEY AUTO_INCREMENT,
    ->     product_name VARCHAR(100) NOT NULL,
    ->     category VARCHAR(50),
    ->     price DECIMAL(10, 2)
    ->  );
Query OK, 0 rows affected (0.07 sec)

mysql>  CREATE TABLE customer_dim (
    ->     customer_id INT PRIMARY KEY AUTO_INCREMENT,
    ->     customer_name VARCHAR(100) NOT NULL,
    ->     gender ENUM('Male', 'Female', 'Other'),
    ->     email VARCHAR(100),
    ->     city VARCHAR(50)
    ->  );
Query OK, 0 rows affected (0.01 sec)

mysql>  CREATE TABLE store_dim (
    ->     store_id INT PRIMARY KEY AUTO_INCREMENT,
    ->     store_name VARCHAR(100),
    ->     location VARCHAR(100),
    ->     manager_name VARCHAR(100)
    ->  );
Query OK, 0 rows affected (0.02 sec)

mysql>  CREATE TABLE time_dim (
    ->     time_id INT PRIMARY KEY AUTO_INCREMENT,
    ->     date DATE NOT NULL,
    ->     day_of_week VARCHAR(10),
    ->     month VARCHAR(10),
    ->     year INT,
    ->     quarter VARCHAR(10)
    ->  );
Query OK, 0 rows affected (0.02 sec)
```

Fig 2: Creating Tables with datatypes

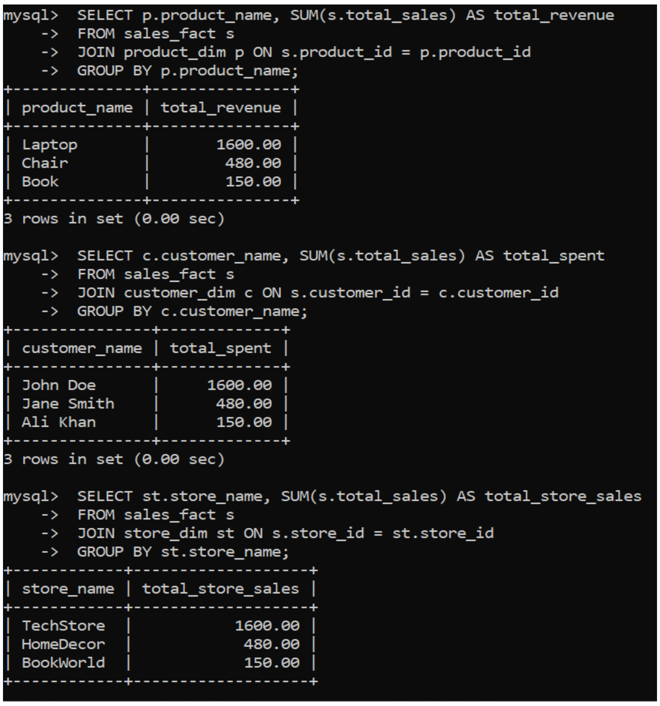
Inserting Data



```
mysql>  INSERT INTO product_dim (product_name, category, price)
    ->  VALUES
    ->  ('Laptop', 'Electronics', 800.00),
    ->  ('Chair', 'Furniture', 120.00),
    ->  ('Book', 'Stationery', 15.00);
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql>  INSERT INTO customer_dim (customer_name, gender, email, city)
    ->  VALUES
    ->  ('John Doe', 'Male', 'john@example.com', 'New York'),
    ->  ('Jane Smith', 'Female', 'jane@example.com', 'San Francisco'),
    ->  ('Ali Khan', 'Male', 'ali@example.com', 'Chicago');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql>  INSERT INTO store_dim (store_name, location, manager_name)
    -> VALUES
    ->  ('TechStore', 'New York', 'Emily White'),
    ->  ('HomeDecor', 'San Francisco', 'Robert Brown'),
    ->  ('BookWorld', 'Chicago', 'Sophia Johnson');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql>  INSERT INTO time_dim (date, day_of_week, month, year, quarter)
    ->  VALUES
    ->  ('2025-01-20', 'Monday', 'January', 2025, 'Q1'),
    ->  ('2025-01-21', 'Tuesday', 'January', 2025, 'Q1'),
    ->  ('2025-01-22', 'Wednesday', 'January', 2025, 'Q1');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Fig 3: Inserting data into tables

Aggregate Queries



```
mysql>  SELECT p.product_name, SUM(s.total_sales) AS total_revenue
    ->  FROM sales_fact s
    ->  JOIN product_dim p ON s.product_id = p.product_id
    ->  GROUP BY p.product_name;
+--------------+---------------+
| product_name | total_revenue |
+--------------+---------------+
| Laptop       |       1600.00 |
| Chair        |        480.00 |
| Book         |        150.00 |
+--------------+---------------+
3 rows in set (0.00 sec)

mysql>  SELECT c.customer_name, SUM(s.total_sales) AS total_spent
    ->  FROM sales_fact s
    ->  JOIN customer_dim c ON s.customer_id = c.customer_id
    ->  GROUP BY c.customer_name;
+---------------+-------------+
| customer_name | total_spent |
+---------------+-------------+
| John Doe      |     1600.00 |
| Jane Smith    |      480.00 |
| Ali Khan      |      150.00 |
+---------------+-------------+
3 rows in set (0.00 sec)

mysql>  SELECT st.store_name, SUM(s.total_sales) AS total_store_sales
    ->  FROM sales_fact s
    ->  JOIN store_dim st ON s.store_id = st.store_id
    ->  GROUP BY st.store_name;
+------------+-------------------+
| store_name | total_store_sales |
+------------+-------------------+
| TechStore  |           1600.00 |
| HomeDecor  |            480.00 |
| BookWorld  |            150.00 |
+------------+-------------------+
```

Fig 4: Aggregate Queries in tables

Madhurima Rawat DS 42