Module 3 Assignment Solution

1. Case study Solution

1. SELECT * FROM customers ORDER BY age ASC;
2. SELECT * FROM customers ORDER BY balance DESC LIMIT 10;
3. SELECT region, AVG(balance) AS avg_balance FROM customers GROUP BY region HAVING avg_balance >60000;
4. SELECT job, SUM(balance) AS total_balance FROM customers GROUP BY job ORDER BY total_balance DESC LIMIT 3;
5. SELECT region, SUM(balance) AS total_balance FROM customers GROUP BY region ORDER BY total_balance DESC LIMIT 5;
6. SELECT SUM(balance) AS total_balance FROM customers WHERE age BETWEEN 25 AND 35;
7. SELECT * FROM customers WHERE gender = 'male' AND balance > 50000;

2. Case Study Solution

```
CREATE TABLE Products (
product_id INT PRIMARY KEY,
product_name VARCHAR(255),
price DECIMAL(10, 2),
category VARCHAR(100)
);

CREATE TABLE Customers (
```

```
customer_id INT PRIMARY KEY,
customer_name VARCHAR(255),
email VARCHAR(255),
phone VARCHAR(20)
);

CREATE TABLE Orders (
order_id INT PRIMARY KEY,
customer_id INT,
product_id INT,
quantity INT,
order_date DATE,
FOREIGN KEY (customer_id) REFERENCES Customers(customer_id),
FOREIGN KEY (product_id) REFERENCES Products(product_id)
);

INSERT INTO Products (product_id, product_name, price, category)
VALUES
(1, 'Laptop', 1200.00, 'Electronics'),
(2, 'Smartphone', 800.00, 'Electronics'),
(3, 'Headphones', 100.00, 'Electronics'),

(4, 'Mouse', 20.00, 'Accessories'),
(5, 'Keyboard', 30.00, 'Accessories');

INSERT INTO Customers (customer_id, customer_name, email, phone)
VALUES
(1, 'John Doe', 'john@example.com', '123-456-7890'),
(2, 'Jane Smith', 'jane@example.com', '987-654-3210'),
```

(3, 'Michael Johnson', 'michael@example.com', '456-789-0123');

INSERT INTO Orders (order_id, customer_id, product_id, quantity, order_date)
VALUES
(1, 1, 1, 2, '2024-04-15'),
(2, 2, 2, 1, '2024-04-16'),
(3, 3, 3, 3, '2024-04-17'),
(4, 1, 4, 1, '2024-04-18'),
(5, 2, 5, 2, '2024-04-19');

3. Case study Solution

1. UPDATE customers SET job = 'Data Scientist' WHERE customerid = 101;
2. UPDATE customers SET balance = balance * 1.1;
3. UPDATE customers SET balance = 0 WHERE age < 30;
4. UPDATE customers SET region = 'Wales' WHERE name = 'Thomas' AND surname = 'Lawrence';
5. UPDATE customers SET job = 'Engineer' WHERE gender = 'Male';
6. ROLLBACK;
7. COMMIT;
8. UPDATE customers SET balance = balance * 2 WHERE name LIKE 'J%';
9. UPDATE customers SET job = 'Manager' ORDER BY balance DESC LIMIT 1;
10. UPDATE customers SET balance = 50000 WHERE region = 'England';

4. Case Study Solution

1. DELETE FROM customers WHERE age > 50;
2. DELETE FROM customers WHERE customerid = 300000812;
3. DELETE FROM customers WHERE job = 'Other';
4. DELETE FROM customers WHERE gender = 'Male' AND age > 40;
5. DELETE FROM customers WHERE region = 'Scotland' AND balance < 50000;

5. Case Study Solution

1. SELECT gender, COUNT(customerid) AS total_customers FROM customers GROUP BY gender;
2. SELECT MAX(age) AS max_age, MIN(age) AS min_age FROM customers;
3. SELECT region, COUNT(customerid) AS total_customers FROM customers GROUP BY region ORDER BY total_customers DESC LIMIT 1;

4. SELECT * FROM customers WHERE age = (SELECT MAX(age) FROM customers) OR age = (SELECT MIN(age) FROM customers);

5. SELECT gender, SUM(balance) AS total_balance FROM customers GROUP BY gender;

6. SELECT job, AVG(balance) AS avg_balance FROM customers GROUP BY job ORDER BY avg_balance DESC LIMIT 1;
SELECT job, AVG(balance) AS avg_balance FROM customers GROUP BY job ORDER BY avg_balance ASC LIMIT 1;

7. SELECT job, COUNT(customerid) AS total_customers FROM customers GROUP BY job ORDER BY total_customers DESC LIMIT 1;

8. SELECT region, AVG(balance) AS avg_balance FROM customers GROUP BY region;
9. SELECT region, SUM(balance) AS total_balance FROM customers GROUP BY region ORDER BY total_balance DESC LIMIT 3;