Database Lab 05: SQL Commands and Queries

DengFeng Qiao Yi He MingYi Wang

1 Basic Tasks

1. **Total Transactions in Bank**

```
SELECT COUNT(*) AS total_transactions FROM transacti
```

2. **Number of 'CHK' Accounts**

```
SELECT COUNT(*) AS chk_accounts
FROM accounts
WHERE account_type = 'CHK';
```

3. **Job Titles and Employee Count**

```
SELECT job_title, COUNT(*) AS num_employees
FROM employees
GROUP BY job_title;
```

4. **Customers and Their Account Count**

```
SELECT customer_id, COUNT(*) AS account_count
FROM accounts
GROUP BY customer_id;
```

5. **Total Balance for Customer (James Hadley, cust_id = 1)**

```
SELECT SUM(balance) AS total_balance
FROM accounts
WHERE customer_id = 1;
```

6. **Total Balance per Customer**

```
SELECT customer_id, SUM(balance) AS total_balance
FROM accounts
GROUP BY customer_id;
```

7. **Account Types and Their Average Balance**

```
SELECT account_type, AVG(balance) AS avg_balance
FROM accounts
GROUP BY account_type;
```

2 Medium Tasks

8. **Total Balance for Woburn Branch Accounts**

```
SELECT SUM(balance) AS total_balance
FROM accounts
WHERE branch = 'Woburn';
```

9. **Highest Balance per Account Type**

```
SELECT account_type, MAX(balance) AS highest_balance
FROM accounts
GROUP BY account_type;
```

10. **Minimum Balance in Accounts**

```
SELECT MIN(balance) AS minimum_balance
FROM accounts;
```

11. **Rounded Down Total Balance per Customer**

```
SELECT customer_id, FLOOR(SUM(balance)) AS total_bal
FROM accounts
GROUP BY customer_id;
```

12. **Formatted Employee Name and Position**

    (a) **Employee Name Format**: LAST_NAME, FIRST_NAME

```
SELECT CONCAT(last_name, ', ', first_nam
FROM employees;
```

    (b) **Employee Position Format**: FIRST_NAME LAST_NAME Position: TITLE

```
SELECT CONCAT(first_name, ' ', last_name
AS employee_position
FROM employees;
```

3 Advanced Tasks

13. **Replace 'anger' with 'panic buying' in Text**

```
SELECT REPLACE('Fear leads to anger; anger leads to
'anger', 'panic buying');
```

14. **Standardize FED_ID Format in CUSTOMER Table**

```
UPDATE customer
SET fed_id = REPLACE(REPLACE(fed_id, '-', ''), ' ',
```

15. **Yearly Transaction Count**

```
SELECT EXTRACT(YEAR FROM transaction_date) AS year,
COUNT(*) AS transaction_count
FROM transactions
GROUP BY year
ORDER BY year;
```

16. **Update Job Titles to Uppercase and Count Titles**

```
UPDATE employees
SET job_title = UPPER(job_title);

SELECT COUNT(*) AS count,
CASE WHEN job_title LIKE '%TELLER%' THEN 'Cashier'
ELSE 'Other' END AS job_category
FROM employees
GROUP BY job_category;
```

17. **Customers with Balance Less Than £5000**

```
SELECT customer_id, SUM(balance) AS total_balance
FROM accounts
GROUP BY customer_id
HAVING total_balance < 5000;
```

18. **Total Staff per Branch**

```
SELECT branch, COUNT(*) AS staff_count
FROM employees
GROUP BY branch;
```

19. **Account Count by Product Codes (CHK and SAV)**

```
SELECT COUNT(*) AS count,
CASE WHEN product_code = 'CHK' THEN 'Checking Accoun
WHEN product_code = 'SAV' THEN 'Savings Account'
END AS product_type
FROM accounts
WHERE product_code IN ('CHK', 'SAV')
GROUP BY product_type;
```

4 Conclusion

This lab exercise covers fundamental to advanced SQL operations, focusing on data retrieval, formatting, aggregation, and data standardization. The tasks highlight essential SQL functions and clauses, reinforcing SQL skills for managing and querying relational databases.