EXPERIMENT 4

Student Name: Vinay Tiwari **UID: 24BDA70919**
Branch: AIT-CSE **Section/Group: 24_AIT_KRG_G2**
Semester: 4
Subject Name: Database Management System **Subject Code: 24CSH-298**

Experiment 4 – Data Analysis Using SQL and PL/SQL

Experiment
Experiment 4: Creating tables, inserting data, performing conditional queries, and using PL/SQL blocks to analyze schema violations and student grades. This experiment demonstrates table creation, updates, conditional logic, and ordering in Oracle SQL and PL/SQL.

Aim
The aim of this experiment is to practice working with Oracle SQL tables, using conditional logic to determine status and grades, and displaying results using SELECT queries and PL/SQL blocks.

Objective

- To create and populate tables in Oracle SQL.
- To use CASE statements for conditional evaluation of violation counts and student grades.
- To add and update columns based on conditions.
- To use PL/SQL anonymous blocks for status messages.
- To sort query results based on defined criteria.

Software Requirements

- Database: Oracle XE or Oracle Live SQL

Practical / Experiment Steps

1. Create a table schema_violations with columns id, schema_name, and violation_count.
2. Insert data for various departments into the schema_violations table.
3. Select violation status for each department using a CASE statement.

EXPERIMENT 4

4. Add a new column approval_status to schema_violations.
5. Update approval_status based on violation count using a CASE statement.
6. Display the updated schema_violations table.
7. Execute a PL/SQL block to print a system status message based on a variable v_count.
8. Create a students table with columns name and marks.
9. Insert student data into the students table.
10. Display student grades using a CASE statement based on marks.
11. Order schema_violations by severity using a CASE statement in ORDER BY.

Procedure of the Experiment

1. Open Oracle XE or Live SQL and connect to the database.
2. Create the schema_violations and students tables.
3. Insert sample data into both tables.
4. Execute SELECT queries with CASE statements to analyze violation and grade data.
5. Alter and update tables using conditional logic.
6. Write and execute a PL/SQL anonymous block for dynamic status messages.
7. Sort and retrieve data based on defined severity.
8. Observe outputs at each step and take screenshots for documentation.

Input / Output Details

Input

- schema_violations table: id, schema_name, violation_count
- students table: name, marks
- PL/SQL block variable: v_count
- Conditional logic in SELECT and UPDATE statements

Step-wise Output

EXPERIMENT 4

Step 1 – Create schema_violations table

[SQL Worksheet]*

```
1  CREATE TABLE schema_violations (
2      id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
3      schema_name VARCHAR2(50),
4      violation_count NUMBER
5  );
```

Query result | Script output | DBMS output | Explain Plan | SQL history

```
        schema_name VARCHAR2(50),
        violation_count NUMBER...
Show more...


Table SCHEMA_VIOLATIONS created.

Elapsed: 00:00:00.017
```

EXPERIMENT 4

Step 2 – Insert data into schema_violations

[SQL Worksheet]*

```
1  INSERT INTO schema_violations (schema_name, violation_count) VALUES ('Finance', 0);
2  INSERT INTO schema_violations (schema_name, violation_count) VALUES ('HR', 2);
3  INSERT INTO schema_violations (schema_name, violation_count) VALUES ('Sales', 5);
4  INSERT INTO schema_violations (schema_name, violation_count) VALUES ('Security', 9);
5  INSERT INTO schema_violations (schema_name, violation_count) VALUES ('Admin', 1);
```

Query result | Script output | DBMS output | Explain Plan | SQL history

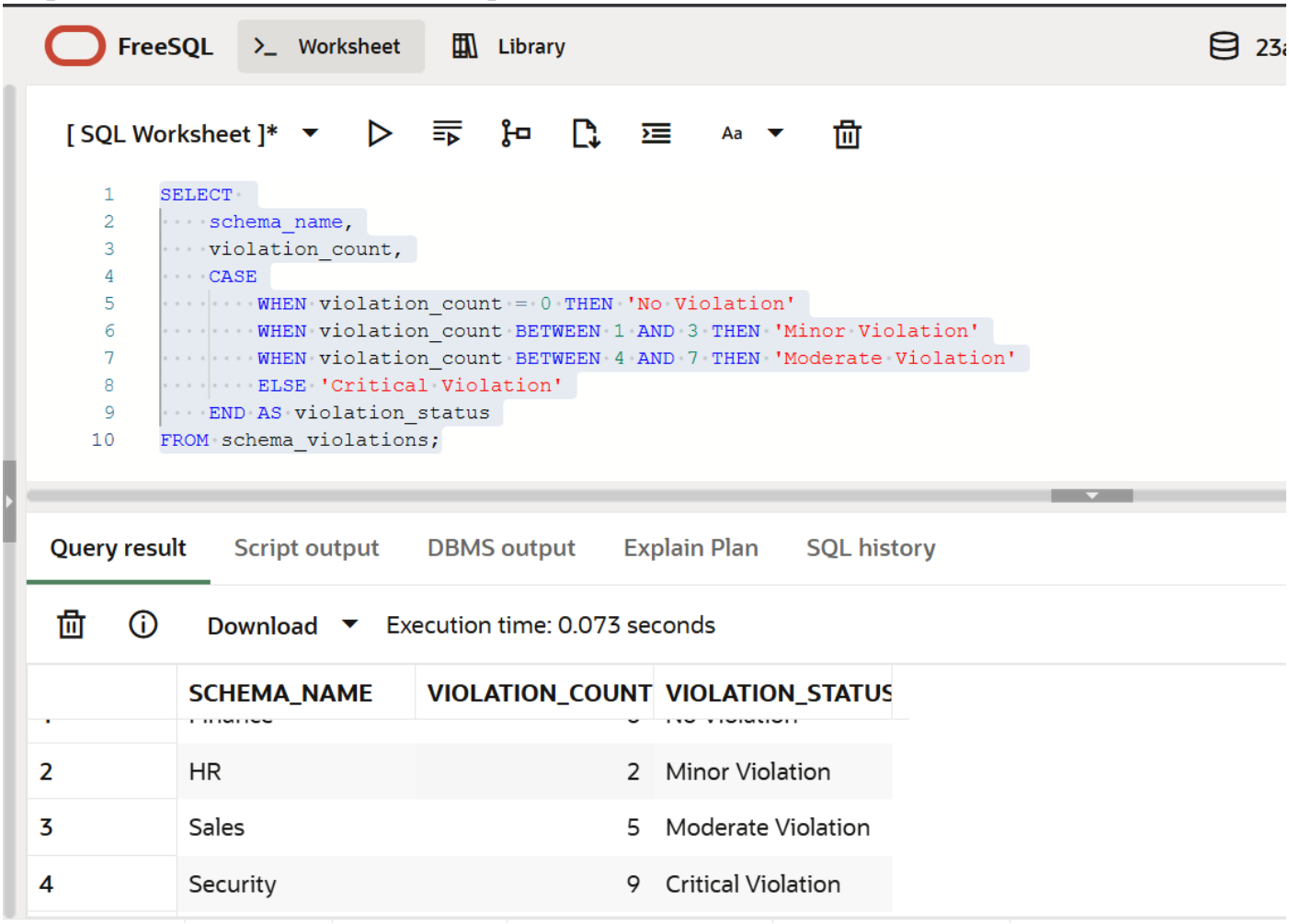
```
SQL> INSERT INTO schema_violations (schema_name, violation_count) VALUES ('Admin', 1)


1 row inserted.

Elapsed: 00:00:00.001
```

EXPERIMENT 4

Step 3 – Violation status of each department



schema_name violation_count violation_status

Finance	0	No Violation
HR	2	Minor Violation
Sales	5	Moderate Violation
Security	9	Critical Violation
Admin	1	Minor Violation

EXPERIMENT 4

Step 4 – Add approval_status column

[SQL Worksheet]*

```
1   -- Add approval_status column
2   ALTER TABLE schema_violations ADD (approval_status VARCHAR2(20));
```

Query result | Script output | DBMS output | Explain Pl

core.util.apex_layout.resize

```
SQL> ALTER TABLE schema_violations ADD (approval_status VARCHAR2(20))


Table SCHEMA_VIOLATIONS altered.

Elapsed: 00:00:00.025
```

EXPERIMENT 4

Step 5 – Update approval_status based on violation_count

[SQL Worksheet]*

```
1   UPDATE schema_violations
2   SET approval_status =
3       CASE
4           WHEN violation_count = 0 THEN 'Approved'
5           WHEN violation_count BETWEEN 1 AND 5 THEN 'Needs Review'
6           ELSE 'Rejected'
7       END;
8
```

Query result | Script output | DBMS output | Explain Plan | SQL history

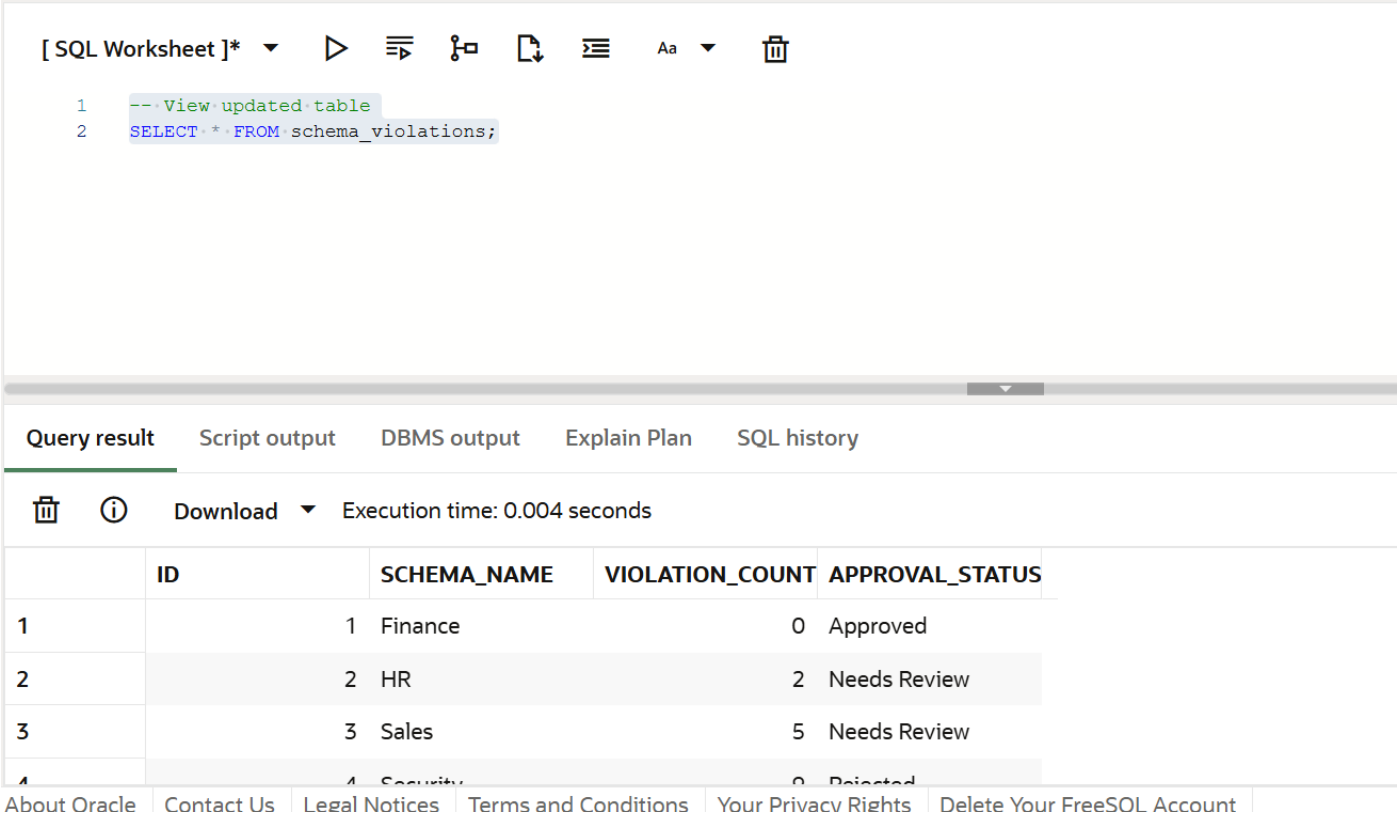
```
        CASE
            WHEN violation_count = 0 THEN 'Approved'...
Show more...


5 rows updated.

Elapsed: 00:00:00.005
```

EXPERIMENT 4

Step 6 – View updated schema_violations table



id	schema_name	violation_count	violation_status	approval_status
1	Finance	0	No Violation	Approved
2	HR	2	Minor Violation	Needs Review
3	Sales	5	Moderate Violation	Needs Review
4	Security	9	Critical Violation	Rejected
5	Admin	1	Minor Violation	Needs Review

Step 7 – PL/SQL anonymous block for status message

Output:

EXPERIMENT 4

[SQL Worksheet]*

```
1   CREATE TABLE students (
2       name VARCHAR2(50),
3       marks NUMBER
4   );
```

Query result | Script output | DBMS output | Explain Plan | SQL history

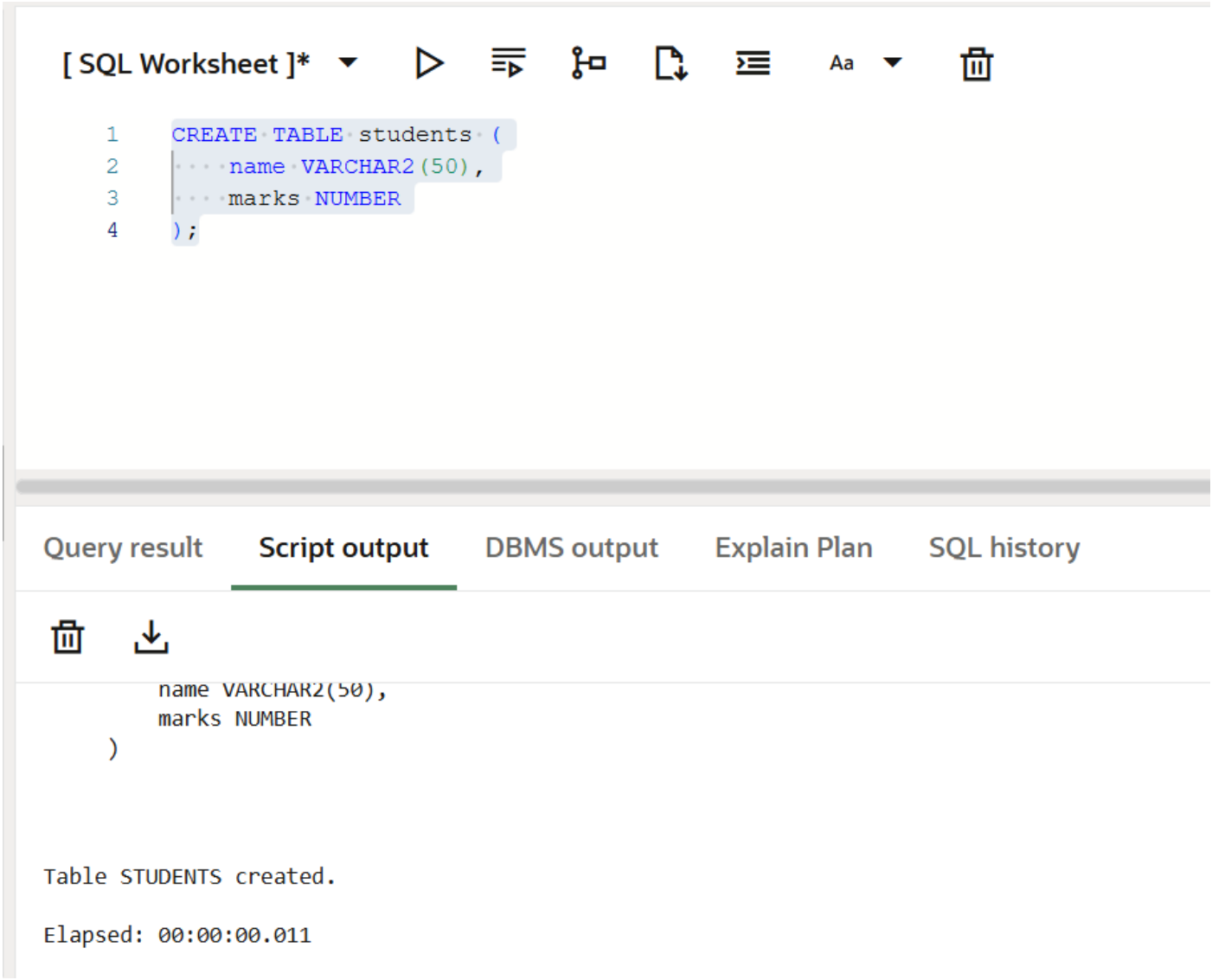
```
        name VARCHAR2(50),
        marks NUMBER
    )


Table STUDENTS created.

Elapsed: 00:00:00.011
```

EXPERIMENT 4

Step 8 – Create students table



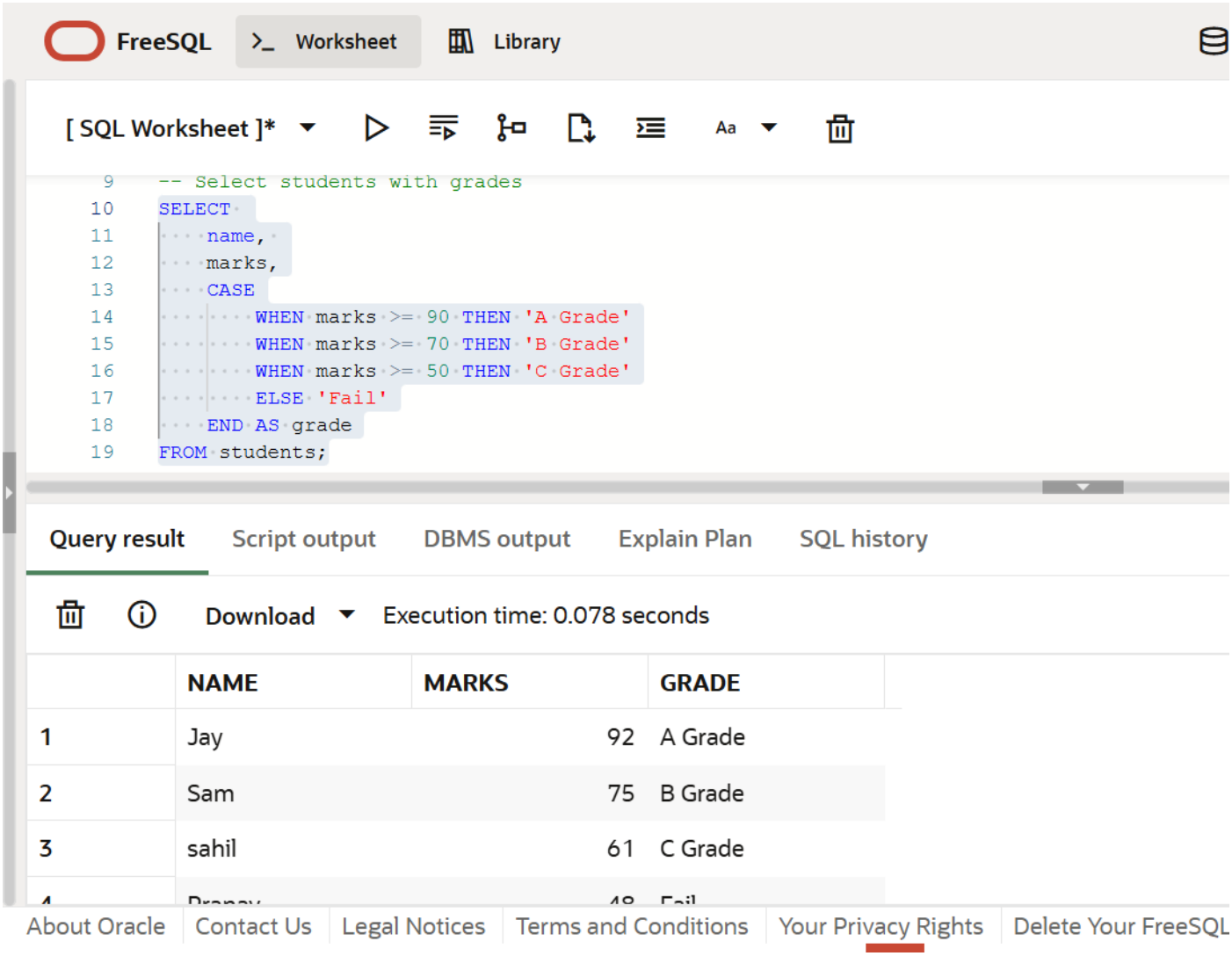
Step 9 – Insert student data

Step 10 – Student grades using CASE statement

EXPERIMENT 4

```
-- Insert student data
INSERT INTO students (name, marks) VALUES ('Meehul', 92);
INSERT INTO students (name, marks) VALUES ('Dibyendu', 75);
INSERT INTO students (name, marks) VALUES ('Vinay', 61);
INSERT INTO students (name, marks) VALUES ('Kaif', 48);

COMMIT;
```



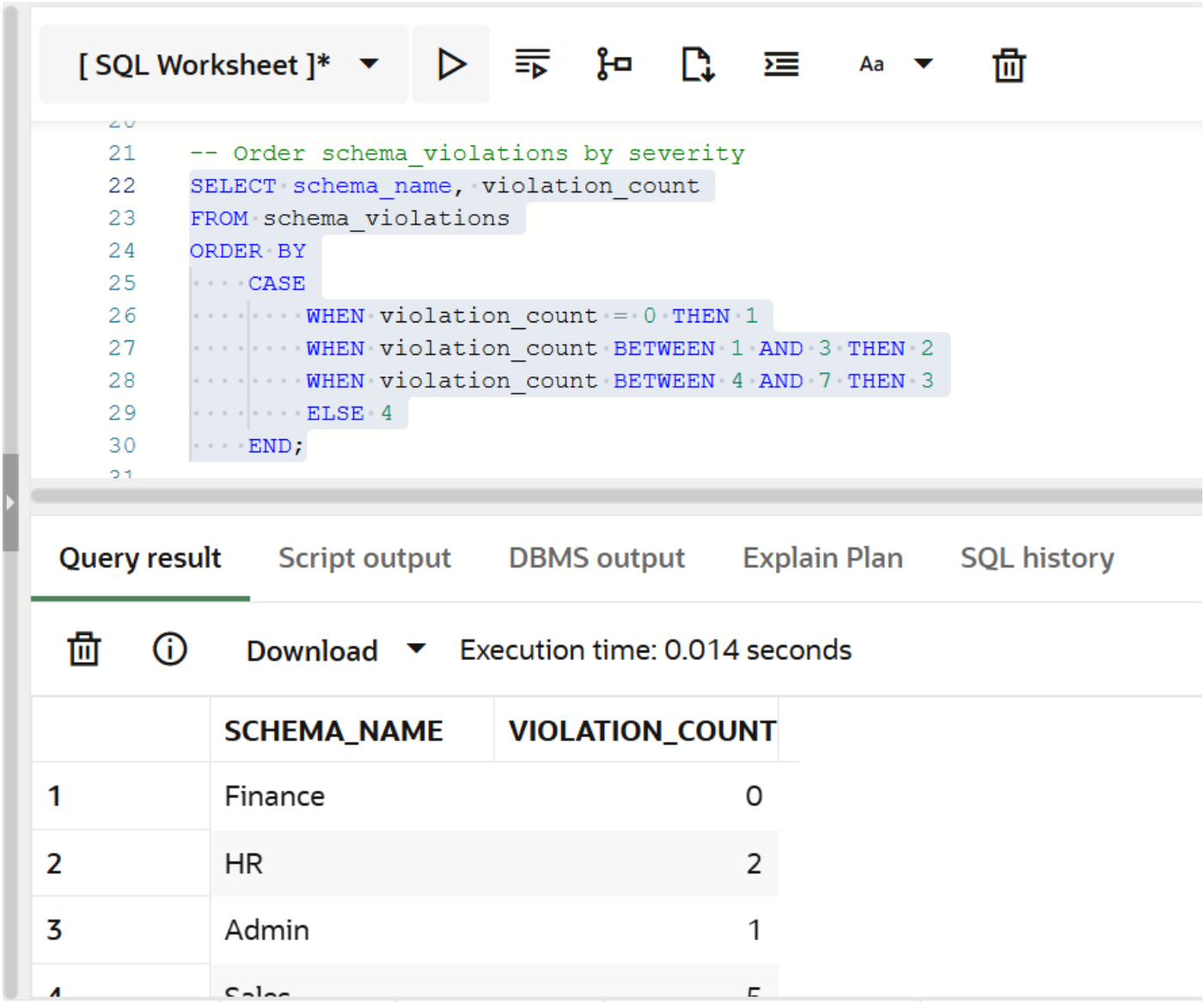
name	marks	grade
Jay	92	A Grade
Sam	75	B Grade
Sahil	61	C Grade

EXPERIMENT 4

name marks grade

Pranav 48 Fail

Step 11 – Schema violations ordered by severity



schema_name violation_count

Finance 0

HR 2

Admin 1

EXPERIMENT 4

schema_name	violation_count
Sales	5
Security	9

Learning Outcome
After completing this experiment, the student will be able to:

- Create and populate tables in Oracle SQL.
- Use CASE statements to evaluate conditions in queries.
- Update table data based on conditional logic.
- Write PL/SQL blocks for dynamic status messages.
- Sort query results using CASE statements in ORDER BY.
- Analyze data and assign grades or approval statuses automatically.
- Interpret step-wise outputs for better understanding of database operations.