Faculty of Technology and Engineering

Chandubhai S. Patel Institute of Technology (CSPIT)

Department of Computer Science & Engineering

Date: / /

Laboratory Manual

Academic Year	:	2024-25	Semester	:	4
Course code	:	CSE206	Course name	:	DATABASE MANAGEMENT SYSTEM

Practical - 5

Aim: As a database administrator for a global bank, you are responsible for managing and analyzing employee and customer data stored in the bank's database. Your tasks involve using SQL functions to manipulate and retrieve critical information efficiently. These operations ensure seamless data communication and compliance with bank regulations.

Constraints

- **Not Null Constraints**: Critical fields like names and salaries must not be null.
- **Unique Constraints**: Ensure integrity of fields like Job_ID.
- **Check Constraints**: Validate positive salary values.

The bank maintains the following schemas:

1. **JobProfile Table**: Stores details of employees and their job roles.

- Emp_ID (Primary Key)
- Emp_Name (Not Null)
- Emp_Salary (Not Null, Check: Greater than zero)
- Job_ID (Unique)
- Department

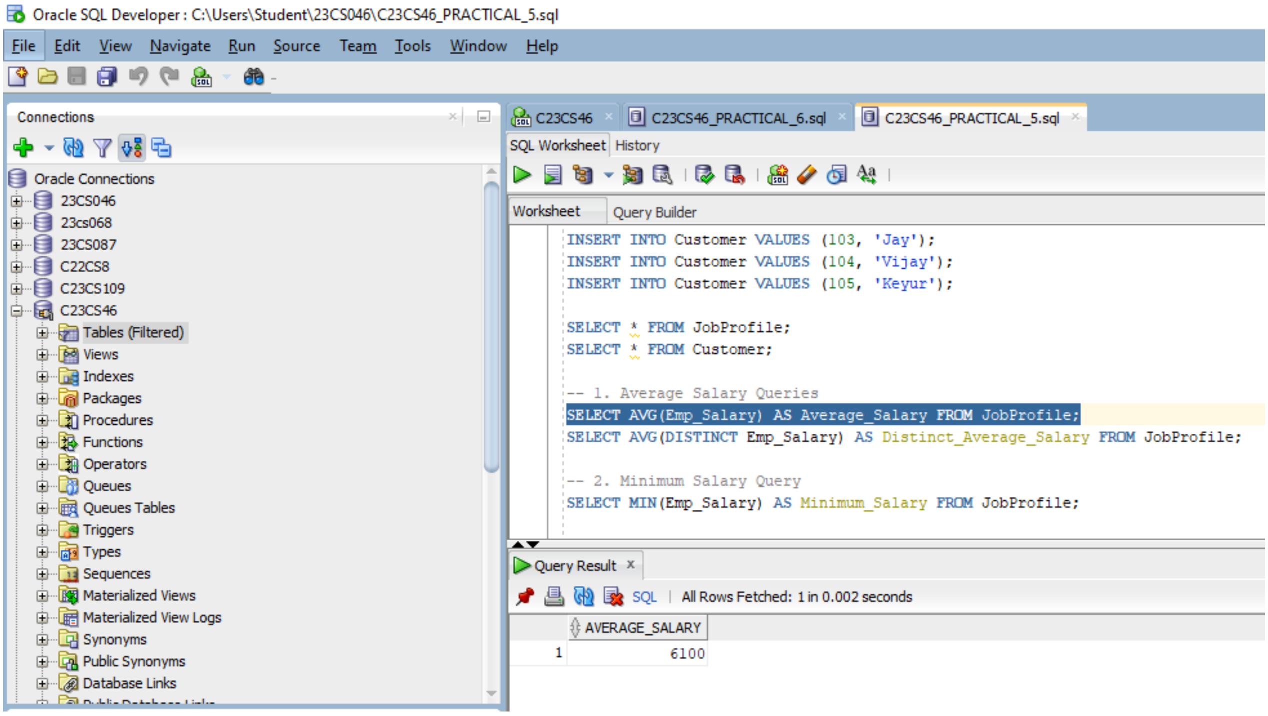
2. **Customer Table**: Stores customer details.

- Cust_ID (Primary Key)
- Cust_Name (Not Null)

Tasks:-

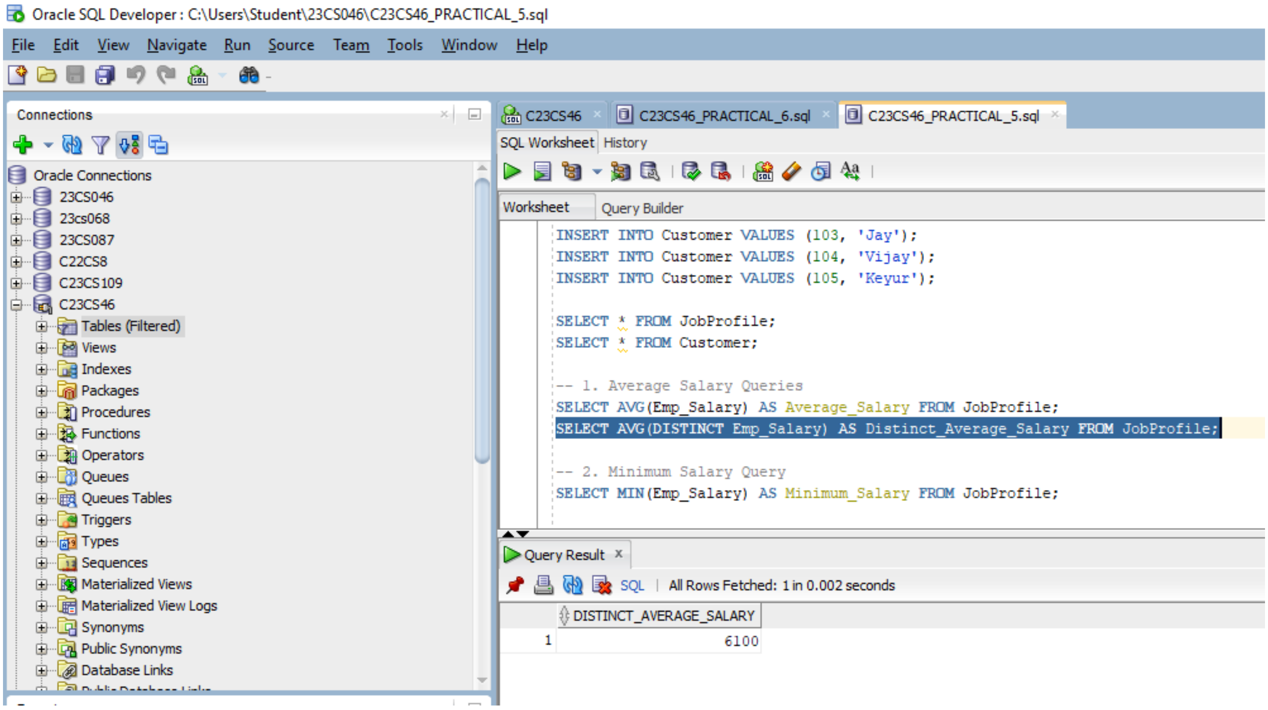
The HR department wants to analyze employee salaries for better planning. Write SQL queries to:

1. Calculate the average salary of employees (with and without duplicates).



```
INSERT INTO Customer VALUES (103, 'Jay');
INSERT INTO Customer VALUES (104, 'Vijay');
INSERT INTO Customer VALUES (105, 'Keyur');

SELECT * FROM JobProfile;
SELECT * FROM Customer;

-- 1. Average Salary Queries
SELECT AVG(Emp_Salary) AS Average_Salary FROM JobProfile;
SELECT AVG(DISTINCT Emp_Salary) AS Distinct_Average_Salary FROM JobProfile;

-- 2. Minimum Salary Query
SELECT MIN(Emp_Salary) AS Minimum_Salary FROM JobProfile;
```

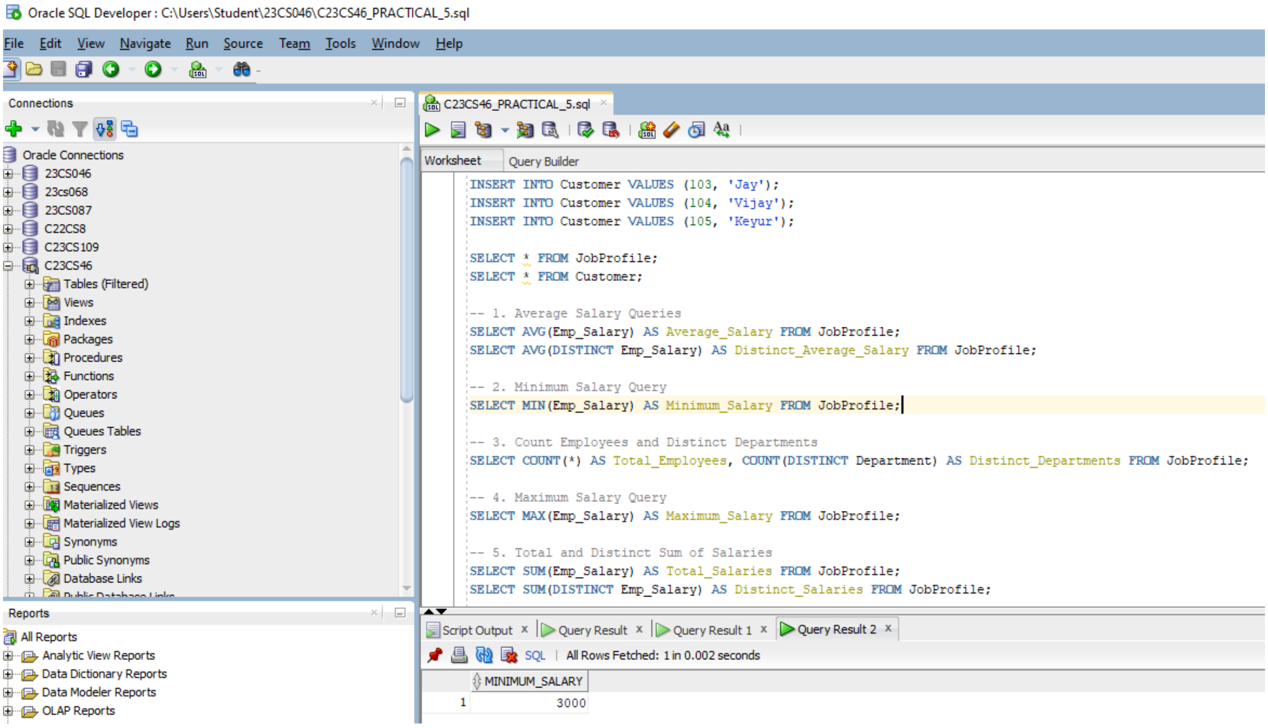
	AVERAGE_SALARY
1	6100



```
INSERT INTO Customer VALUES (103, 'Jay');
INSERT INTO Customer VALUES (104, 'Vijay');
INSERT INTO Customer VALUES (105, 'Keyur');

SELECT * FROM JobProfile;
SELECT * FROM Customer;

-- 1. Average Salary Queries
SELECT AVG(Emp_Salary) AS Average_Salary FROM JobProfile;
SELECT AVG(DISTINCT Emp_Salary) AS Distinct_Average_Salary FROM JobProfile;

-- 2. Minimum Salary Query
SELECT MIN(Emp_Salary) AS Minimum_Salary FROM JobProfile;
```

	DISTINCT_AVERAGE_SALARY
1	6100

2. Retrieve the minimum salary from the JobProfile table.

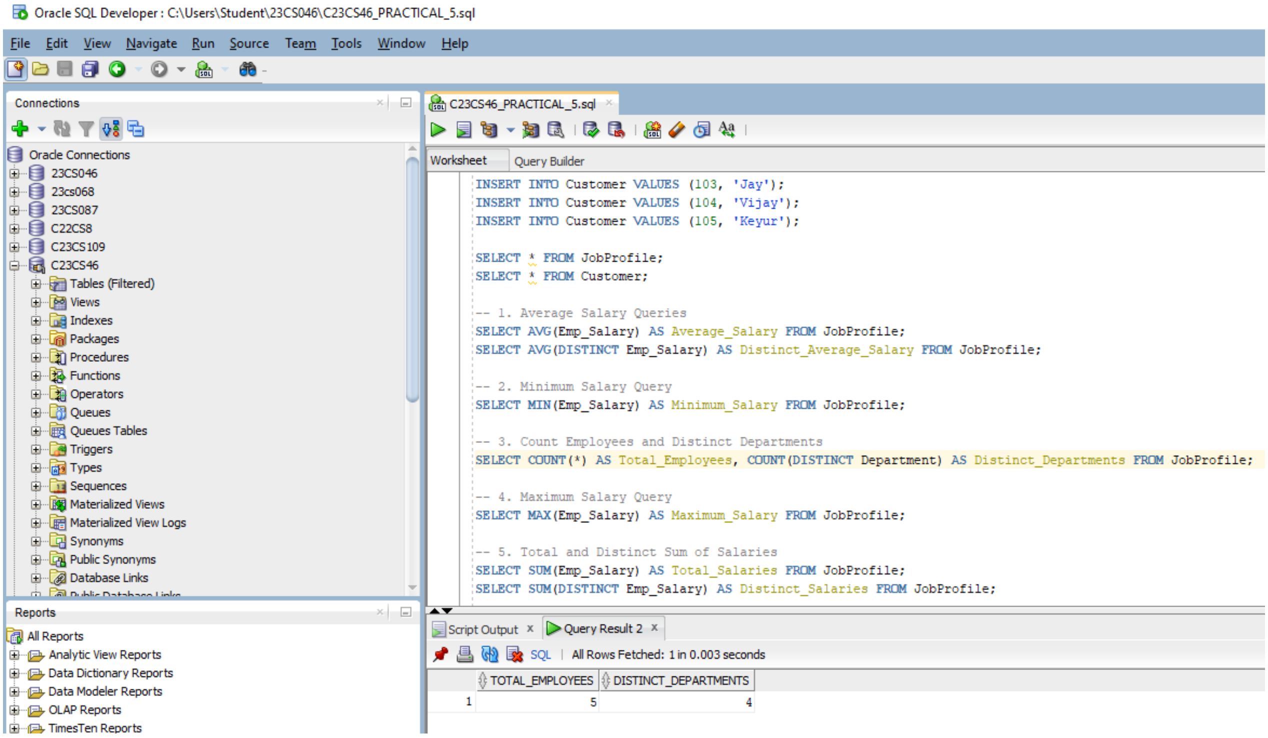


```sql
INSERT INTO Customer VALUES (103, 'Jay');
INSERT INTO Customer VALUES (104, 'Vijay');
INSERT INTO Customer VALUES (105, 'Keyur');

SELECT * FROM JobProfile;
SELECT * FROM Customer;

-- 1. Average Salary Queries
SELECT AVG(Emp_Salary) AS Average_Salary FROM JobProfile;
SELECT AVG(DISTINCT Emp_Salary) AS Distinct_Average_Salary FROM JobProfile;

-- 2. Minimum Salary Query
SELECT MIN(Emp_Salary) AS Minimum_Salary FROM JobProfile;

-- 3. Count Employees and Distinct Departments
SELECT COUNT(*) AS Total_Employees, COUNT(DISTINCT Department) AS Distinct_Departments FROM JobProfile;

-- 4. Maximum Salary Query
SELECT MAX(Emp_Salary) AS Maximum_Salary FROM JobProfile;

-- 5. Total and Distinct Sum of Salaries
SELECT SUM(Emp_Salary) AS Total_Salaries FROM JobProfile;
SELECT SUM(DISTINCT Emp_Salary) AS Distinct_Salaries FROM JobProfile;
```

All Rows Fetched: 1 in 0.002 seconds

	MINIMUM_SALARY
1	3000

3. Count the total number of employees and distinct departments.

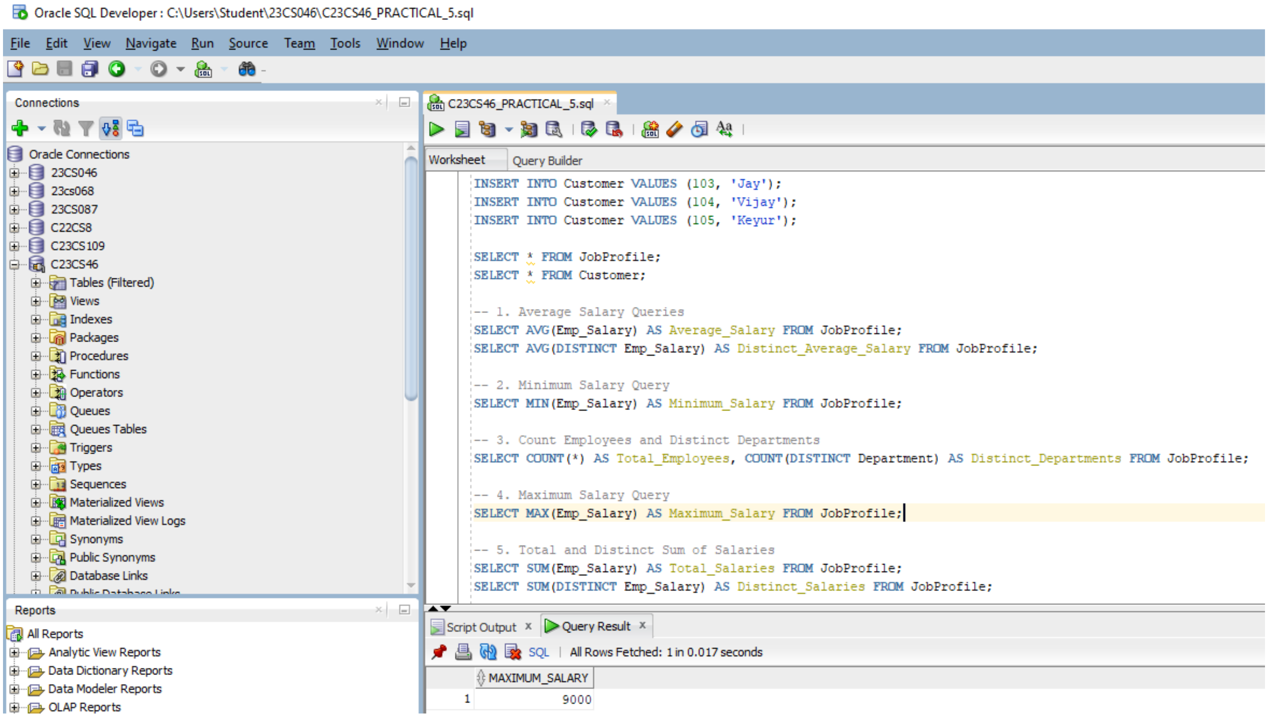


```sql
INSERT INTO Customer VALUES (103, 'Jay');
INSERT INTO Customer VALUES (104, 'Vijay');
INSERT INTO Customer VALUES (105, 'Keyur');

SELECT * FROM JobProfile;
SELECT * FROM Customer;

-- 1. Average Salary Queries
SELECT AVG(Emp_Salary) AS Average_Salary FROM JobProfile;
SELECT AVG(DISTINCT Emp_Salary) AS Distinct_Average_Salary FROM JobProfile;

-- 2. Minimum Salary Query
SELECT MIN(Emp_Salary) AS Minimum_Salary FROM JobProfile;

-- 3. Count Employees and Distinct Departments
SELECT COUNT(*) AS Total_Employees, COUNT(DISTINCT Department) AS Distinct_Departments FROM JobProfile;

-- 4. Maximum Salary Query
SELECT MAX(Emp_Salary) AS Maximum_Salary FROM JobProfile;

-- 5. Total and Distinct Sum of Salaries
SELECT SUM(Emp_Salary) AS Total_Salaries FROM JobProfile;
SELECT SUM(DISTINCT Emp_Salary) AS Distinct_Salaries FROM JobProfile;
```

All Rows Fetched: 1 in 0.003 seconds

	TOTAL_EMPLOYEES	DISTINCT_DEPARTMENTS
1	5	4

4. Retrieve the maximum salary from the JobProfile table.

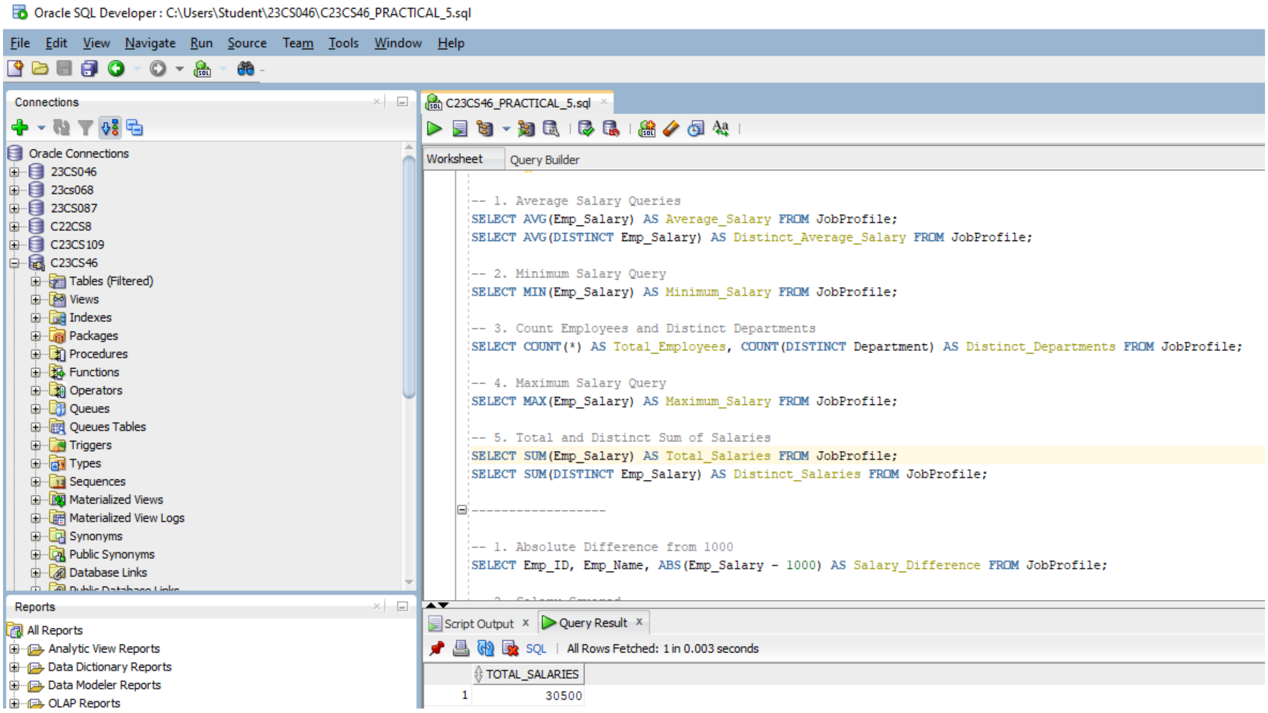


```sql
INSERT INTO Customer VALUES (103, 'Jay');
INSERT INTO Customer VALUES (104, 'Vijay');
INSERT INTO Customer VALUES (105, 'Keyur');

SELECT * FROM JobProfile;
SELECT * FROM Customer;

-- 1. Average Salary Queries
SELECT AVG(Emp_Salary) AS Average_Salary FROM JobProfile;
SELECT AVG(DISTINCT Emp_Salary) AS Distinct_Average_Salary FROM JobProfile;

-- 2. Minimum Salary Query
SELECT MIN(Emp_Salary) AS Minimum_Salary FROM JobProfile;

-- 3. Count Employees and Distinct Departments
SELECT COUNT(*) AS Total_Employees, COUNT(DISTINCT Department) AS Distinct_Departments FROM JobProfile;

-- 4. Maximum Salary Query
SELECT MAX(Emp_Salary) AS Maximum_Salary FROM JobProfile;

-- 5. Total and Distinct Sum of Salaries
SELECT SUM(Emp_Salary) AS Total_Salaries FROM JobProfile;
SELECT SUM(DISTINCT Emp_Salary) AS Distinct_Salaries FROM JobProfile;
```

All Rows Fetched: 1 in 0.017 seconds

	MAXIMUM_SALARY
1	9000

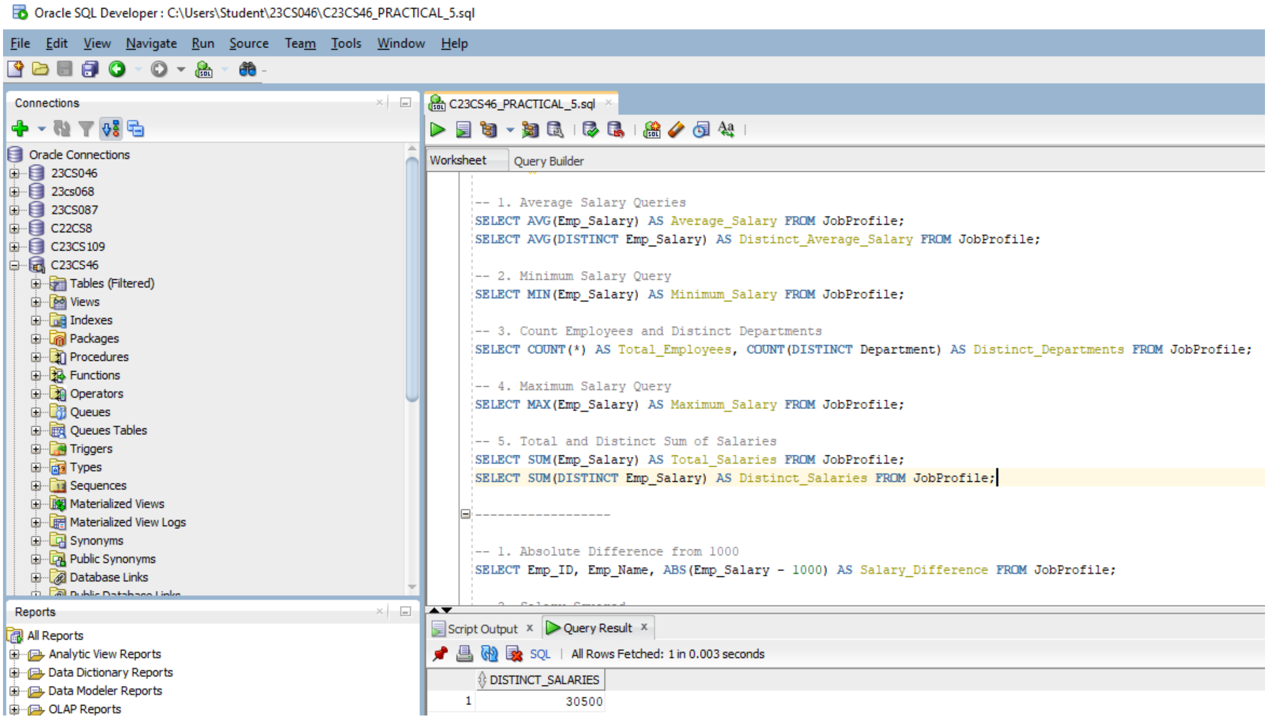
5. Calculate the total and distinct sum of all salaries.



```sql
-- 1. Average Salary Queries
SELECT AVG(Emp_Salary) AS Average_Salary FROM JobProfile;
SELECT AVG(DISTINCT Emp_Salary) AS Distinct_Average_Salary FROM JobProfile;

-- 2. Minimum Salary Query
SELECT MIN(Emp_Salary) AS Minimum_Salary FROM JobProfile;

-- 3. Count Employees and Distinct Departments
SELECT COUNT(*) AS Total_Employees, COUNT(DISTINCT Department) AS Distinct_Departments FROM JobProfile;

-- 4. Maximum Salary Query
SELECT MAX(Emp_Salary) AS Maximum_Salary FROM JobProfile;

-- 5. Total and Distinct Sum of Salaries
SELECT SUM(Emp_Salary) AS Total_Salaries FROM JobProfile;
SELECT SUM(DISTINCT Emp_Salary) AS Distinct_Salaries FROM JobProfile;

------------------

-- 1. Absolute Difference from 1000
SELECT Emp_ID, Emp_Name, ABS(Emp_Salary - 1000) AS Salary_Difference FROM JobProfile;

-- 2. Salary Squared
```

All Rows Fetched: 1 in 0.003 seconds

	TOTAL_SALARIES
1	30500



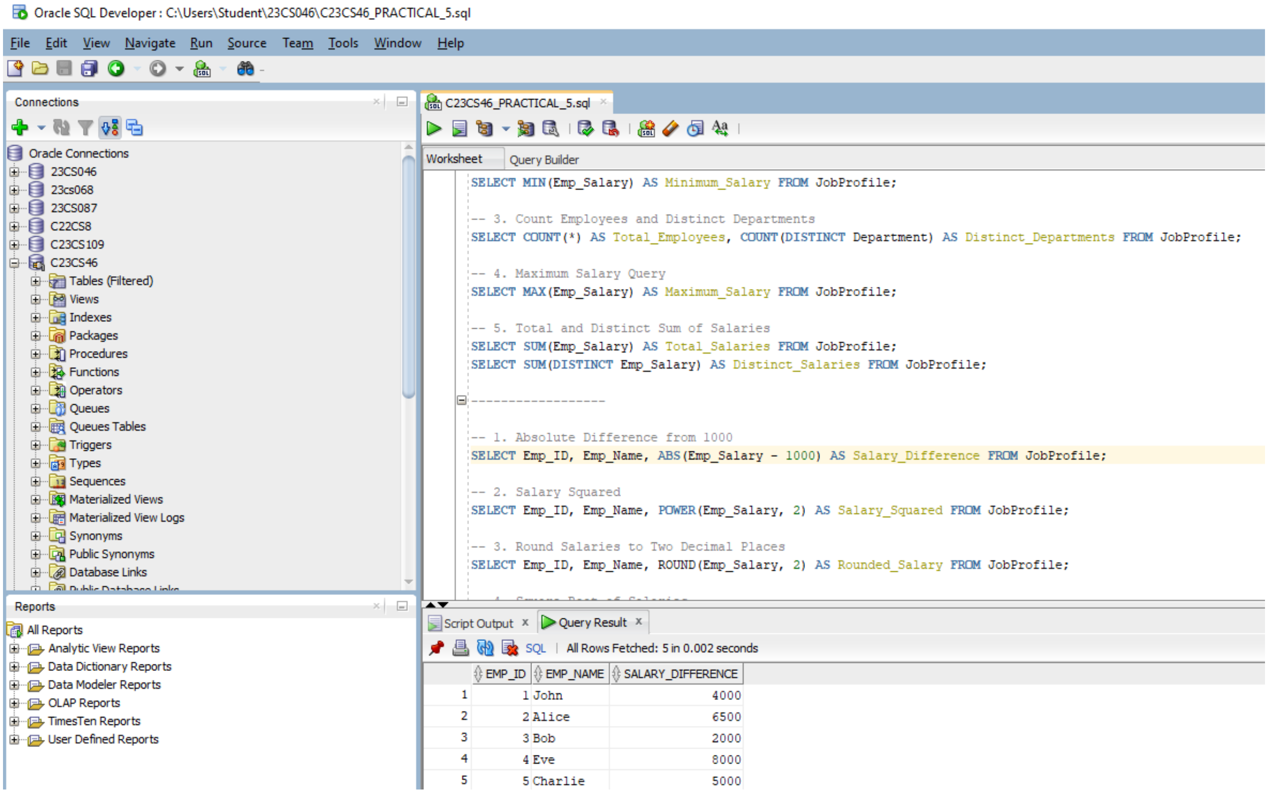
```
-- 1. Average Salary Queries
SELECT AVG(Emp_Salary) AS Average_Salary FROM JobProfile;
SELECT AVG(DISTINCT Emp_Salary) AS Distinct_Average_Salary FROM JobProfile;

-- 2. Minimum Salary Query
SELECT MIN(Emp_Salary) AS Minimum_Salary FROM JobProfile;

-- 3. Count Employees and Distinct Departments
SELECT COUNT(*) AS Total_Employees, COUNT(DISTINCT Department) AS Distinct_Departments FROM JobProfile;

-- 4. Maximum Salary Query
SELECT MAX(Emp_Salary) AS Maximum_Salary FROM JobProfile;

-- 5. Total and Distinct Sum of Salaries
SELECT SUM(Emp_Salary) AS Total_Salaries FROM JobProfile;
SELECT SUM(DISTINCT Emp_Salary) AS Distinct_Salaries FROM JobProfile;

------------------

-- 1. Absolute Difference from 1000
SELECT Emp_ID, Emp_Name, ABS(Emp_Salary - 1000) AS Salary_Difference FROM JobProfile;
```

All Rows Fetched: 1 in 0.003 seconds

	DISTINCT_SALARIES
1	30500

The finance team needs specific salary calculations for tax and benefits. Write SQL queries to:

1. Calculate the absolute difference between each employee’s salary and ₹1,000.

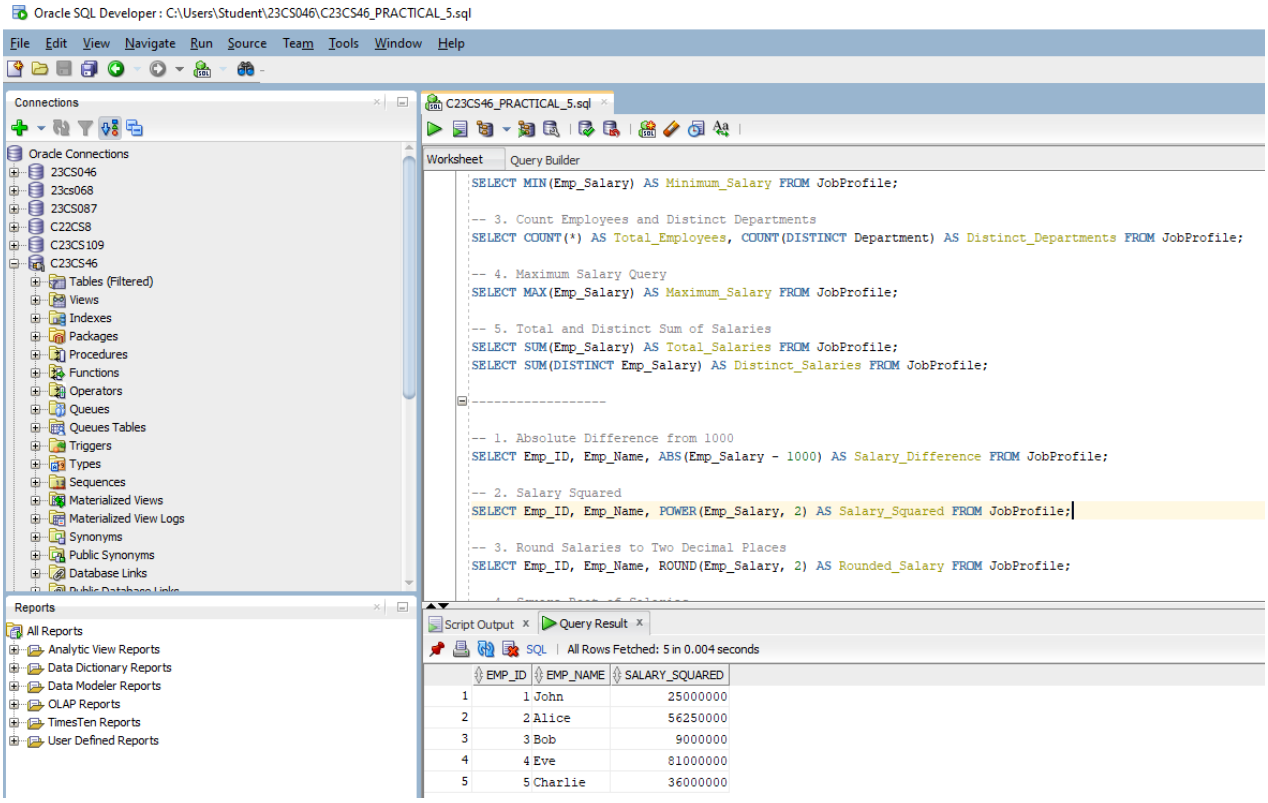


```
SELECT MIN(Emp_Salary) AS Minimum_Salary FROM JobProfile;

-- 3. Count Employees and Distinct Departments
SELECT COUNT(*) AS Total_Employees, COUNT(DISTINCT Department) AS Distinct_Departments FROM JobProfile;

-- 4. Maximum Salary Query
SELECT MAX(Emp_Salary) AS Maximum_Salary FROM JobProfile;

-- 5. Total and Distinct Sum of Salaries
SELECT SUM(Emp_Salary) AS Total_Salaries FROM JobProfile;
SELECT SUM(DISTINCT Emp_Salary) AS Distinct_Salaries FROM JobProfile;

------------------

-- 1. Absolute Difference from 1000
SELECT Emp_ID, Emp_Name, ABS(Emp_Salary - 1000) AS Salary_Difference FROM JobProfile;

-- 2. Salary Squared
SELECT Emp_ID, Emp_Name, POWER(Emp_Salary, 2) AS Salary_Squared FROM JobProfile;

-- 3. Round Salaries to Two Decimal Places
SELECT Emp_ID, Emp_Name, ROUND(Emp_Salary, 2) AS Rounded_Salary FROM JobProfile;
```

All Rows Fetched: 5 in 0.002 seconds

	EMP_ID	EMP_NAME	SALARY_DIFFERENCE
1	1	John	4000
2	2	Alice	6500
3	3	Bob	2000
4	4	Eve	8000
5	5	Charlie	5000

2. Compute the square of each employee’s salary.

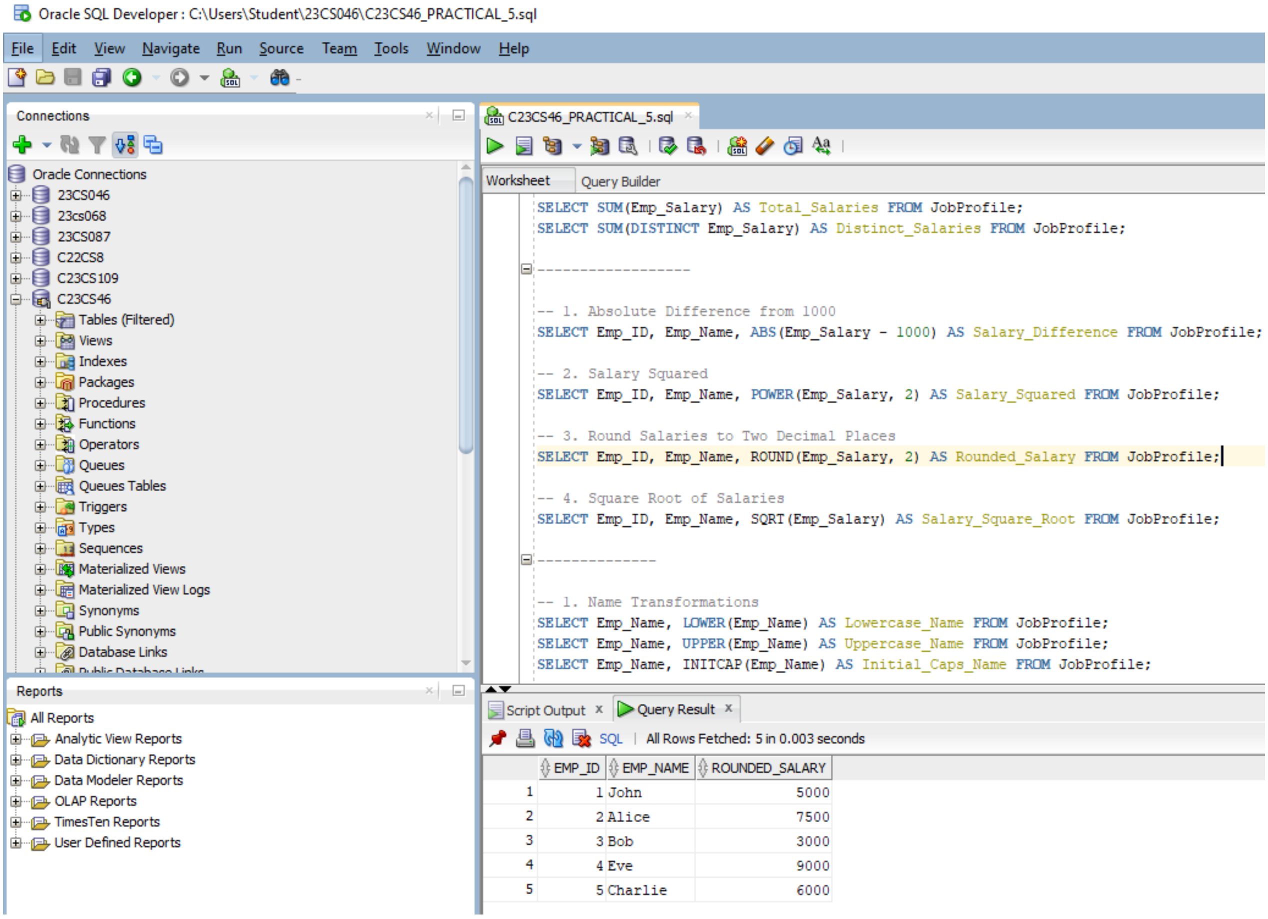


```
SELECT MIN(Emp_Salary) AS Minimum_Salary FROM JobProfile;

-- 3. Count Employees and Distinct Departments
SELECT COUNT(*) AS Total_Employees, COUNT(DISTINCT Department) AS Distinct_Departments FROM JobProfile;

-- 4. Maximum Salary Query
SELECT MAX(Emp_Salary) AS Maximum_Salary FROM JobProfile;

-- 5. Total and Distinct Sum of Salaries
SELECT SUM(Emp_Salary) AS Total_Salaries FROM JobProfile;
SELECT SUM(DISTINCT Emp_Salary) AS Distinct_Salaries FROM JobProfile;

------------------

-- 1. Absolute Difference from 1000
SELECT Emp_ID, Emp_Name, ABS(Emp_Salary - 1000) AS Salary_Difference FROM JobProfile;

-- 2. Salary Squared
SELECT Emp_ID, Emp_Name, POWER(Emp_Salary, 2) AS Salary_Squared FROM JobProfile;

-- 3. Round Salaries to Two Decimal Places
SELECT Emp_ID, Emp_Name, ROUND(Emp_Salary, 2) AS Rounded_Salary FROM JobProfile;
```

All Rows Fetched: 5 in 0.004 seconds

	EMP_ID	EMP_NAME	SALARY_SQUARED
1	1	John	25000000
2	2	Alice	56250000
3	3	Bob	9000000
4	4	Eve	81000000
5	5	Charlie	36000000

3. Round salaries to two decimal places.

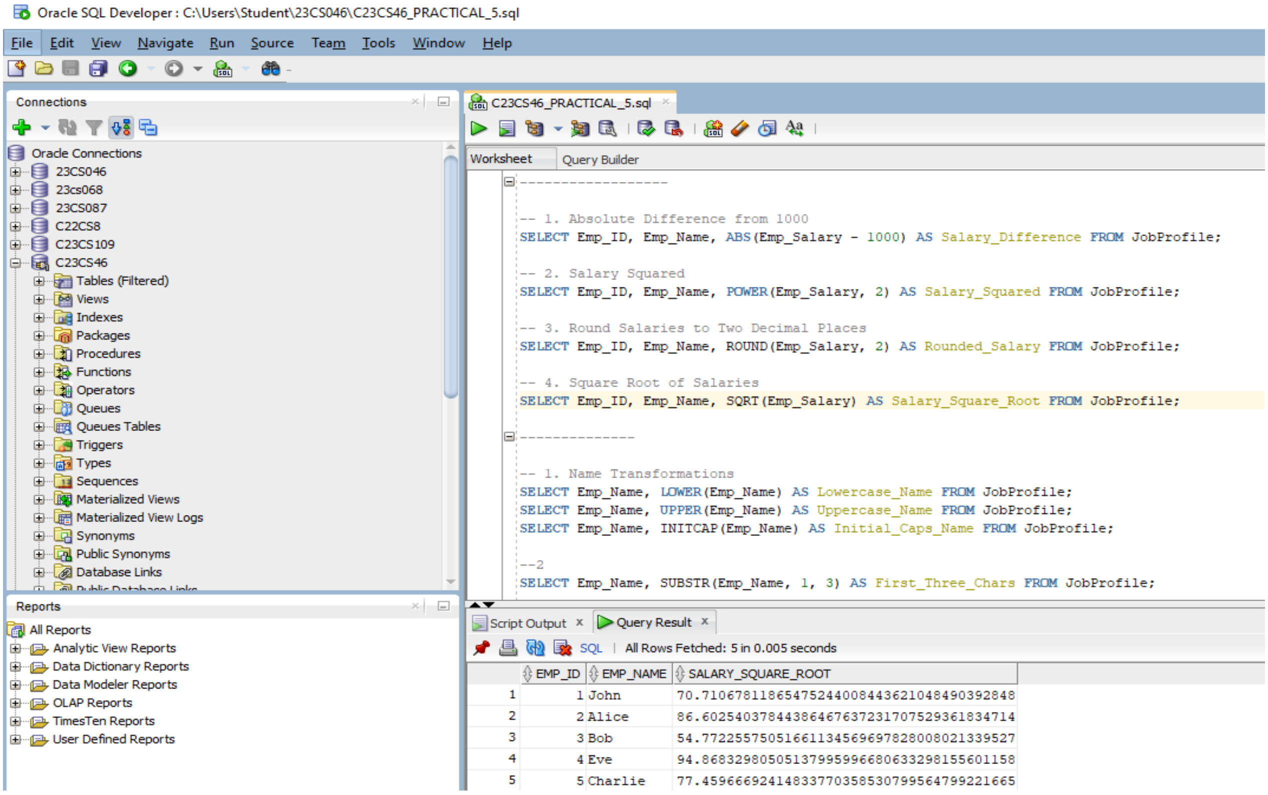


```
SELECT SUM(Emp_Salary) AS Total_Salaries FROM JobProfile;
SELECT SUM(DISTINCT Emp_Salary) AS Distinct_Salaries FROM JobProfile;

------------------

-- 1. Absolute Difference from 1000
SELECT Emp_ID, Emp_Name, ABS(Emp_Salary - 1000) AS Salary_Difference FROM JobProfile;

-- 2. Salary Squared
SELECT Emp_ID, Emp_Name, POWER(Emp_Salary, 2) AS Salary_Squared FROM JobProfile;

-- 3. Round Salaries to Two Decimal Places
SELECT Emp_ID, Emp_Name, ROUND(Emp_Salary, 2) AS Rounded_Salary FROM JobProfile;

-- 4. Square Root of Salaries
SELECT Emp_ID, Emp_Name, SQRT(Emp_Salary) AS Salary_Square_Root FROM JobProfile;

--------------

-- 1. Name Transformations
SELECT Emp_Name, LOWER(Emp_Name) AS Lowercase_Name FROM JobProfile;
SELECT Emp_Name, UPPER(Emp_Name) AS Uppercase_Name FROM JobProfile;
SELECT Emp_Name, INITCAP(Emp_Name) AS Initial_Caps_Name FROM JobProfile;
```

All Rows Fetched: 5 in 0.003 seconds

	EMP_ID	EMP_NAME	ROUNDED_SALARY
1	1	John	5000
2	2	Alice	7500
3	3	Bob	3000
4	4	Eve	9000
5	5	Charlie	6000

4. Find the square root of salaries.



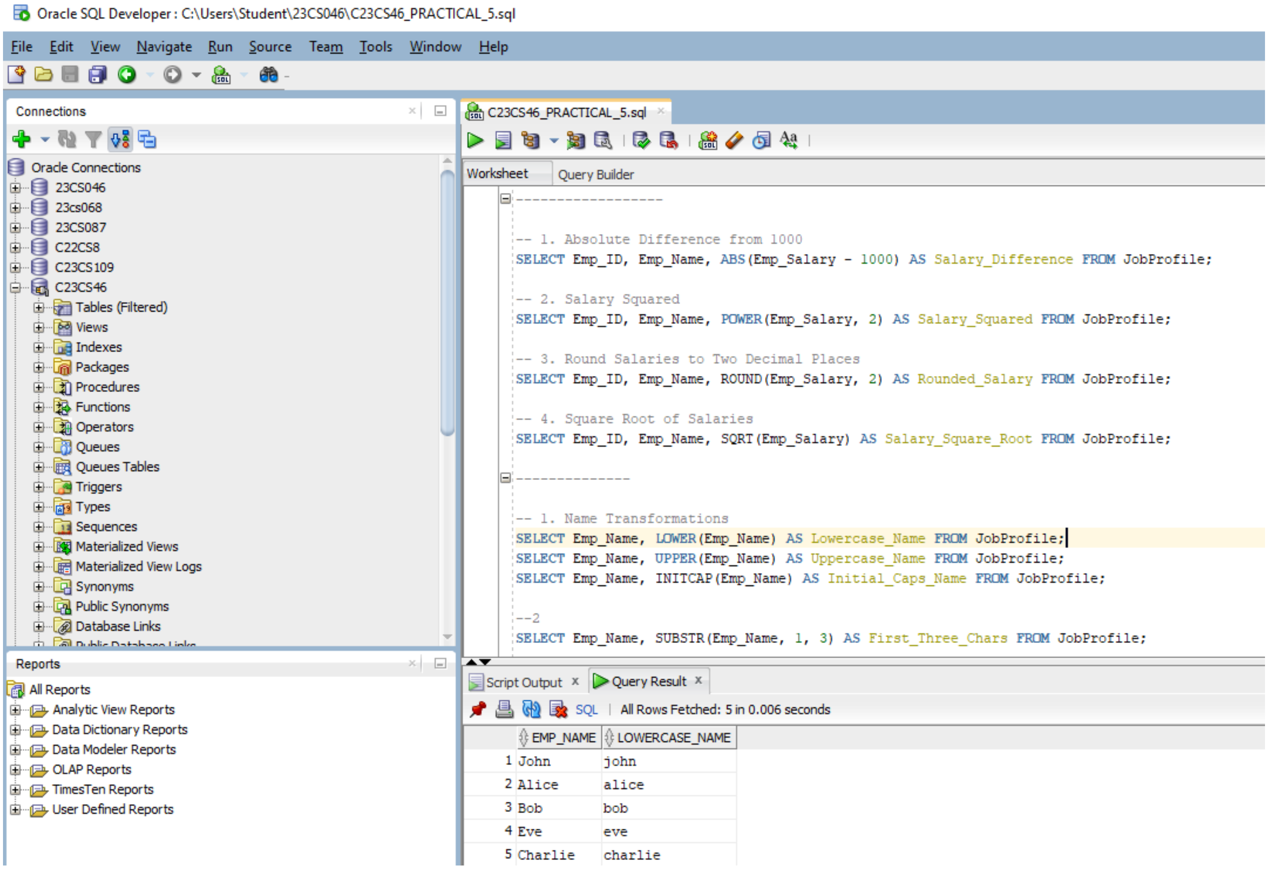
```
------------------

-- 1. Absolute Difference from 1000
SELECT Emp_ID, Emp_Name, ABS(Emp_Salary - 1000) AS Salary_Difference FROM JobProfile;

-- 2. Salary Squared
SELECT Emp_ID, Emp_Name, POWER(Emp_Salary, 2) AS Salary_Squared FROM JobProfile;

-- 3. Round Salaries to Two Decimal Places
SELECT Emp_ID, Emp_Name, ROUND(Emp_Salary, 2) AS Rounded_Salary FROM JobProfile;

-- 4. Square Root of Salaries
SELECT Emp_ID, Emp_Name, SQRT(Emp_Salary) AS Salary_Square_Root FROM JobProfile;

-------------

-- 1. Name Transformations
SELECT Emp_Name, LOWER(Emp_Name) AS Lowercase_Name FROM JobProfile;
SELECT Emp_Name, UPPER(Emp_Name) AS Uppercase_Name FROM JobProfile;
SELECT Emp_Name, INITCAP(Emp_Name) AS Initial_Caps_Name FROM JobProfile;

--2
SELECT Emp_Name, SUBSTR(Emp_Name, 1, 3) AS First_Three_Chars FROM JobProfile;
```

All Rows Fetched: 5 in 0.005 seconds

	EMP_ID	EMP_NAME	SALARY_SQUARE_ROOT
1	1	John	70.710678118654752440084436210484903928 48
2	2	Alice	86.602540378443864676372317075293618347 14
3	3	Bob	54.772255750516611345696978280080213395 27
4	4	Eve	94.868329805051379959966806332981556011 58
5	5	Charlie	77.459666924148337703585307995647992216 65

To ensure uniformity in names across systems, perform the following:

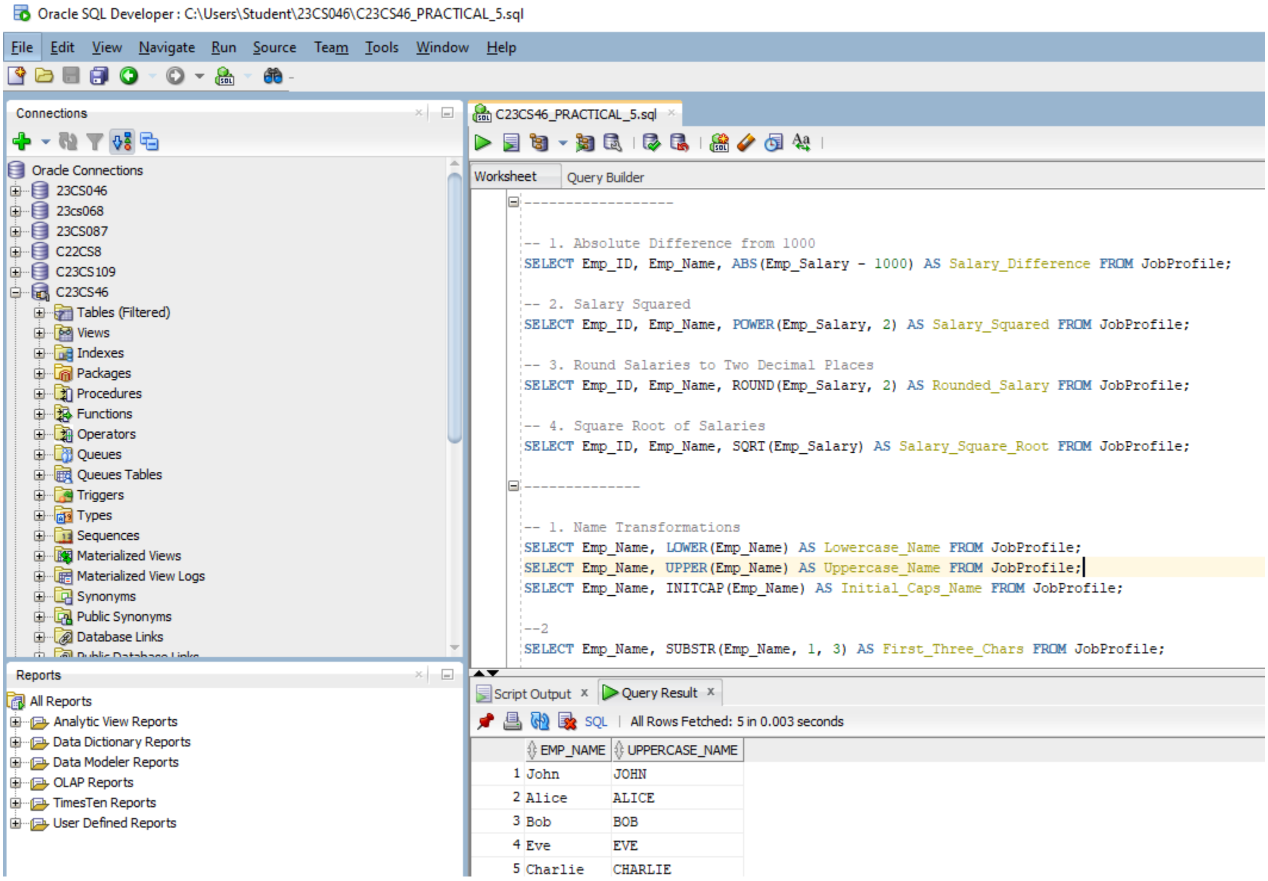
1. Convert all employee first names to lowercase, uppercase, and initial caps.



```
------------------

-- 1. Absolute Difference from 1000
SELECT Emp_ID, Emp_Name, ABS(Emp_Salary - 1000) AS Salary_Difference FROM JobProfile;

-- 2. Salary Squared
SELECT Emp_ID, Emp_Name, POWER(Emp_Salary, 2) AS Salary_Squared FROM JobProfile;

-- 3. Round Salaries to Two Decimal Places
SELECT Emp_ID, Emp_Name, ROUND(Emp_Salary, 2) AS Rounded_Salary FROM JobProfile;

-- 4. Square Root of Salaries
SELECT Emp_ID, Emp_Name, SQRT(Emp_Salary) AS Salary_Square_Root FROM JobProfile;

-------------

-- 1. Name Transformations
SELECT Emp_Name, LOWER(Emp_Name) AS Lowercase_Name FROM JobProfile;
SELECT Emp_Name, UPPER(Emp_Name) AS Uppercase_Name FROM JobProfile;
SELECT Emp_Name, INITCAP(Emp_Name) AS Initial_Caps_Name FROM JobProfile;

--2
SELECT Emp_Name, SUBSTR(Emp_Name, 1, 3) AS First_Three_Chars FROM JobProfile;
```

All Rows Fetched: 5 in 0.006 seconds

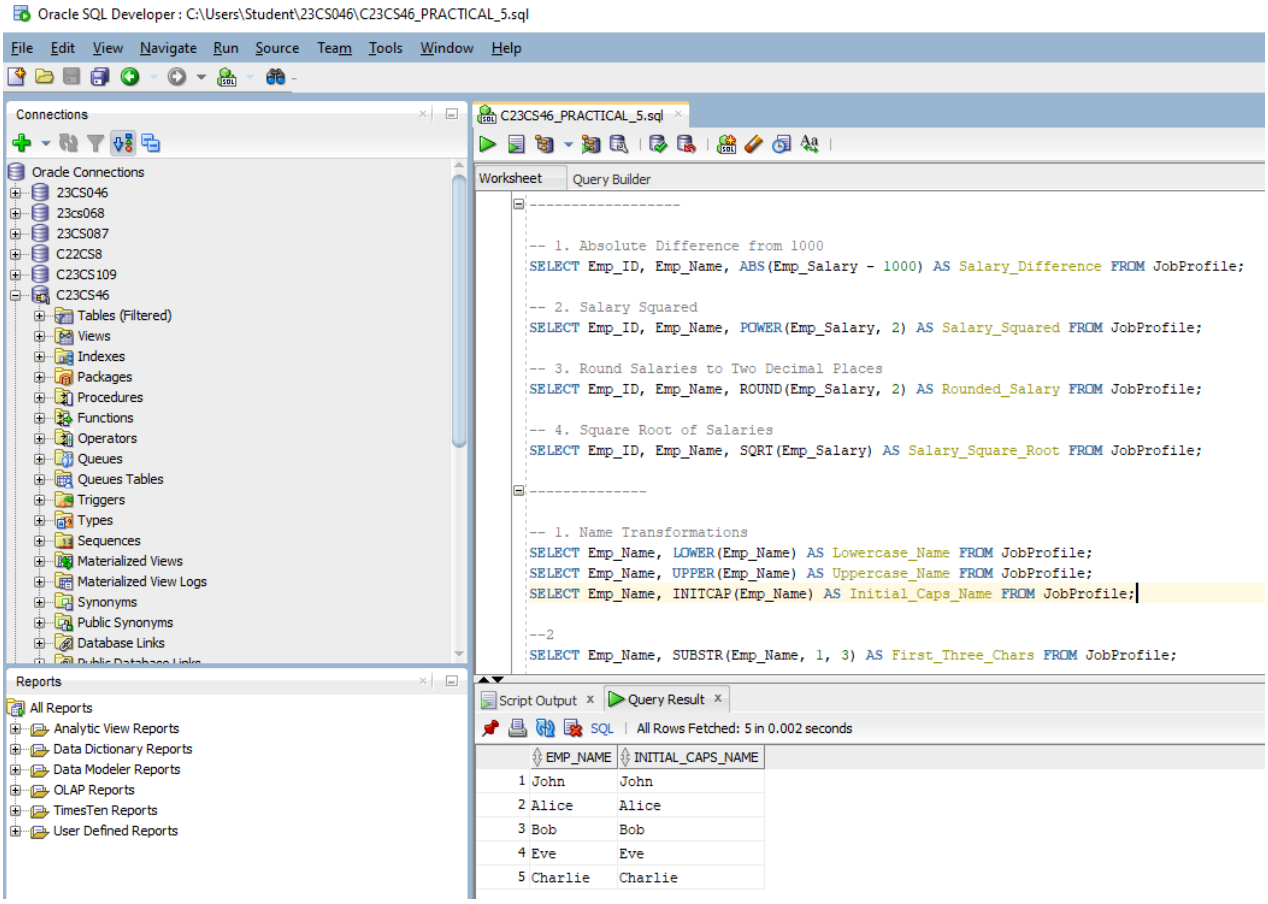
	EMP_NAME	LOWERCASE_NAME
1	John	john
2	Alice	alice
3	Bob	bob
4	Eve	eve
5	Charlie	charlie



```
------------------

-- 1. Absolute Difference from 1000
SELECT Emp_ID, Emp_Name, ABS(Emp_Salary - 1000) AS Salary_Difference FROM JobProfile;

-- 2. Salary Squared
SELECT Emp_ID, Emp_Name, POWER(Emp_Salary, 2) AS Salary_Squared FROM JobProfile;

-- 3. Round Salaries to Two Decimal Places
SELECT Emp_ID, Emp_Name, ROUND(Emp_Salary, 2) AS Rounded_Salary FROM JobProfile;

-- 4. Square Root of Salaries
SELECT Emp_ID, Emp_Name, SQRT(Emp_Salary) AS Salary_Square_Root FROM JobProfile;

--------------

-- 1. Name Transformations
SELECT Emp_Name, LOWER(Emp_Name) AS Lowercase_Name FROM JobProfile;
SELECT Emp_Name, UPPER(Emp_Name) AS Uppercase_Name FROM JobProfile;
SELECT Emp_Name, INITCAP(Emp_Name) AS Initial_Caps_Name FROM JobProfile;

--2
SELECT Emp_Name, SUBSTR(Emp_Name, 1, 3) AS First_Three_Chars FROM JobProfile;
```

All Rows Fetched: 5 in 0.003 seconds

	EMP_NAME	UPPERCASE_NAME
1	John	JOHN
2	Alice	ALICE
3	Bob	BOB
4	Eve	EVE
5	Charlie	CHARLIE

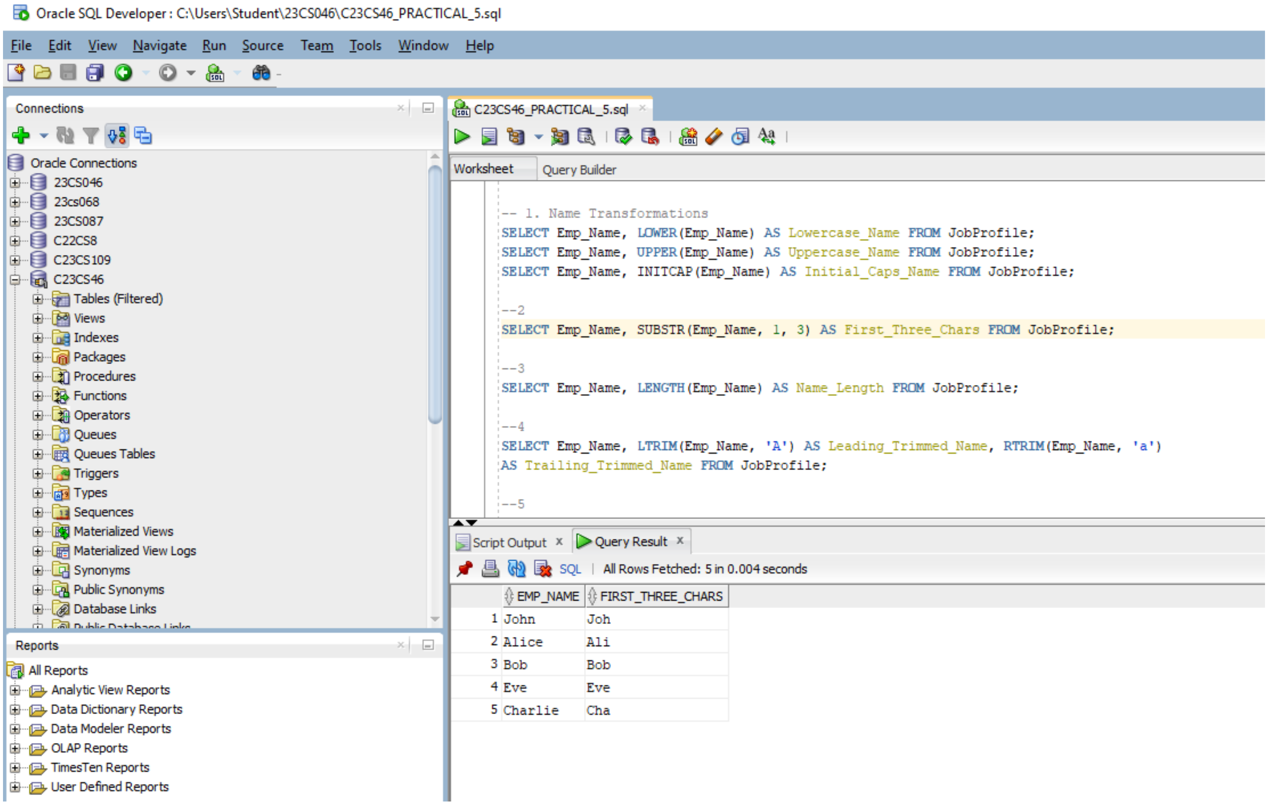


```
------------------

-- 1. Absolute Difference from 1000
SELECT Emp_ID, Emp_Name, ABS(Emp_Salary - 1000) AS Salary_Difference FROM JobProfile;

-- 2. Salary Squared
SELECT Emp_ID, Emp_Name, POWER(Emp_Salary, 2) AS Salary_Squared FROM JobProfile;

-- 3. Round Salaries to Two Decimal Places
SELECT Emp_ID, Emp_Name, ROUND(Emp_Salary, 2) AS Rounded_Salary FROM JobProfile;

-- 4. Square Root of Salaries
SELECT Emp_ID, Emp_Name, SQRT(Emp_Salary) AS Salary_Square_Root FROM JobProfile;

--------------

-- 1. Name Transformations
SELECT Emp_Name, LOWER(Emp_Name) AS Lowercase_Name FROM JobProfile;
SELECT Emp_Name, UPPER(Emp_Name) AS Uppercase_Name FROM JobProfile;
SELECT Emp_Name, INITCAP(Emp_Name) AS Initial_Caps_Name FROM JobProfile;

--2
SELECT Emp_Name, SUBSTR(Emp_Name, 1, 3) AS First_Three_Chars FROM JobProfile;
```

All Rows Fetched: 5 in 0.002 seconds

	EMP_NAME	INITIAL_CAPS_NAME
1	John	John
2	Alice	Alice
3	Bob	Bob
4	Eve	Eve
5	Charlie	Charlie

2. Extract the first three characters of employee first names.

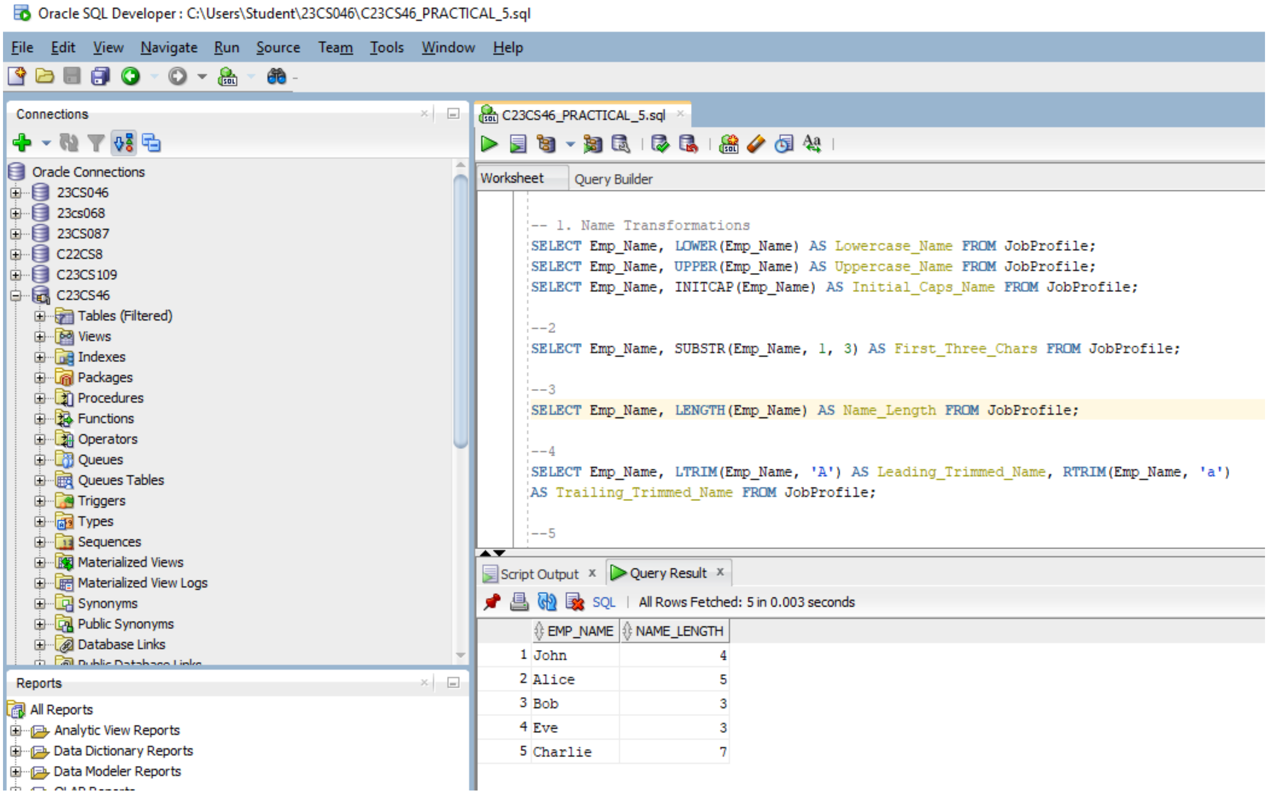


```
-- 1. Name Transformations
SELECT Emp_Name, LOWER(Emp_Name) AS Lowercase_Name FROM JobProfile;
SELECT Emp_Name, UPPER(Emp_Name) AS Uppercase_Name FROM JobProfile;
SELECT Emp_Name, INITCAP(Emp_Name) AS Initial_Caps_Name FROM JobProfile;

--2
SELECT Emp_Name, SUBSTR(Emp_Name, 1, 3) AS First_Three_Chars FROM JobProfile;

--3
SELECT Emp_Name, LENGTH(Emp_Name) AS Name_Length FROM JobProfile;

--4
SELECT Emp_Name, LTRIM(Emp_Name, 'A') AS Leading_Trimmed_Name, RTRIM(Emp_Name, 'a')
AS Trailing_Trimmed_Name FROM JobProfile;

--5
```

All Rows Fetched: 5 in 0.004 seconds

	EMP_NAME	FIRST_THREE_CHARS
1	John	Joh
2	Alice	Ali
3	Bob	Bob
4	Eve	Eve
5	Charlie	Cha

3. Find the length of each employee's first name.

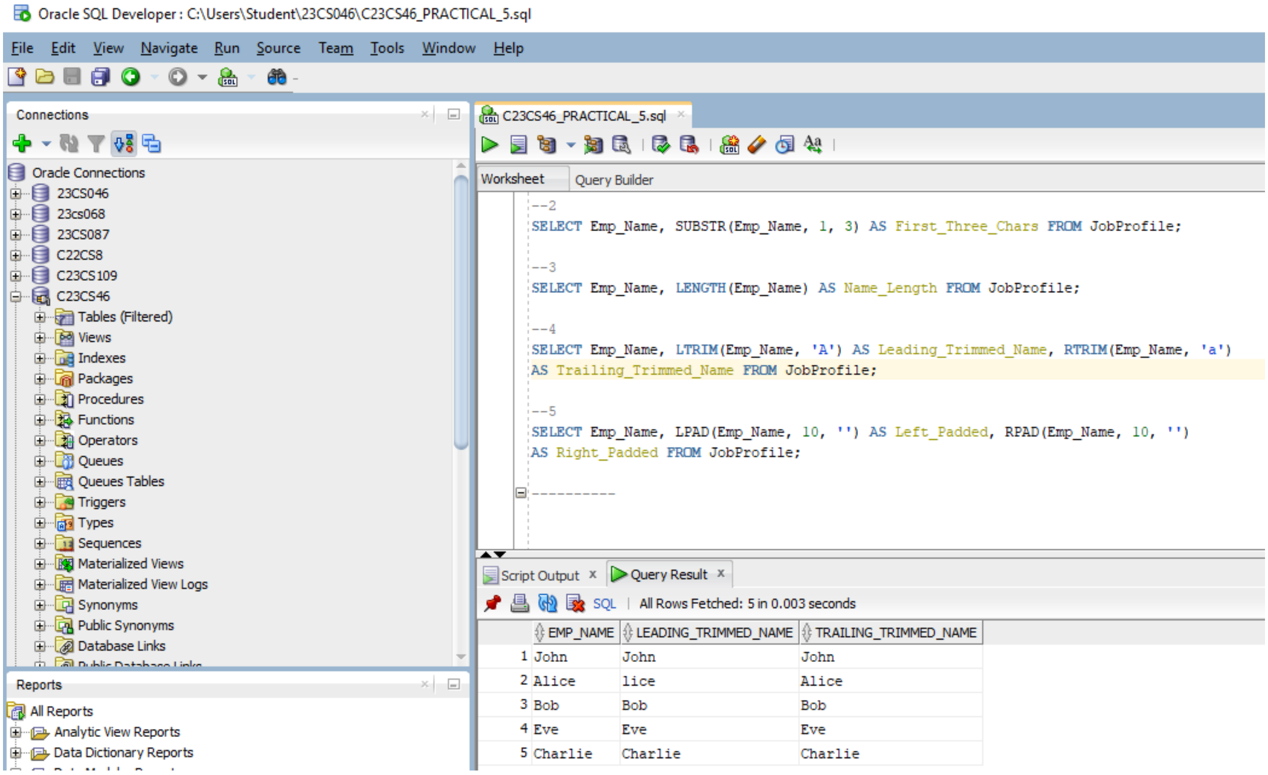


```
-- 1. Name Transformations
SELECT Emp_Name, LOWER(Emp_Name) AS Lowercase_Name FROM JobProfile;
SELECT Emp_Name, UPPER(Emp_Name) AS Uppercase_Name FROM JobProfile;
SELECT Emp_Name, INITCAP(Emp_Name) AS Initial_Caps_Name FROM JobProfile;

--2
SELECT Emp_Name, SUBSTR(Emp_Name, 1, 3) AS First_Three_Chars FROM JobProfile;

--3
SELECT Emp_Name, LENGTH(Emp_Name) AS Name_Length FROM JobProfile;

--4
SELECT Emp_Name, LTRIM(Emp_Name, 'A') AS Leading_Trimmed_Name, RTRIM(Emp_Name, 'a')
AS Trailing_Trimmed_Name FROM JobProfile;

--5
```

All Rows Fetched: 5 in 0.003 seconds

	EMP_NAME	NAME_LENGTH
1	John	4
2	Alice	5
3	Bob	3
4	Eve	3
5	Charlie	7

4. Remove leading 'A' and trailing 'a' from employee first names.

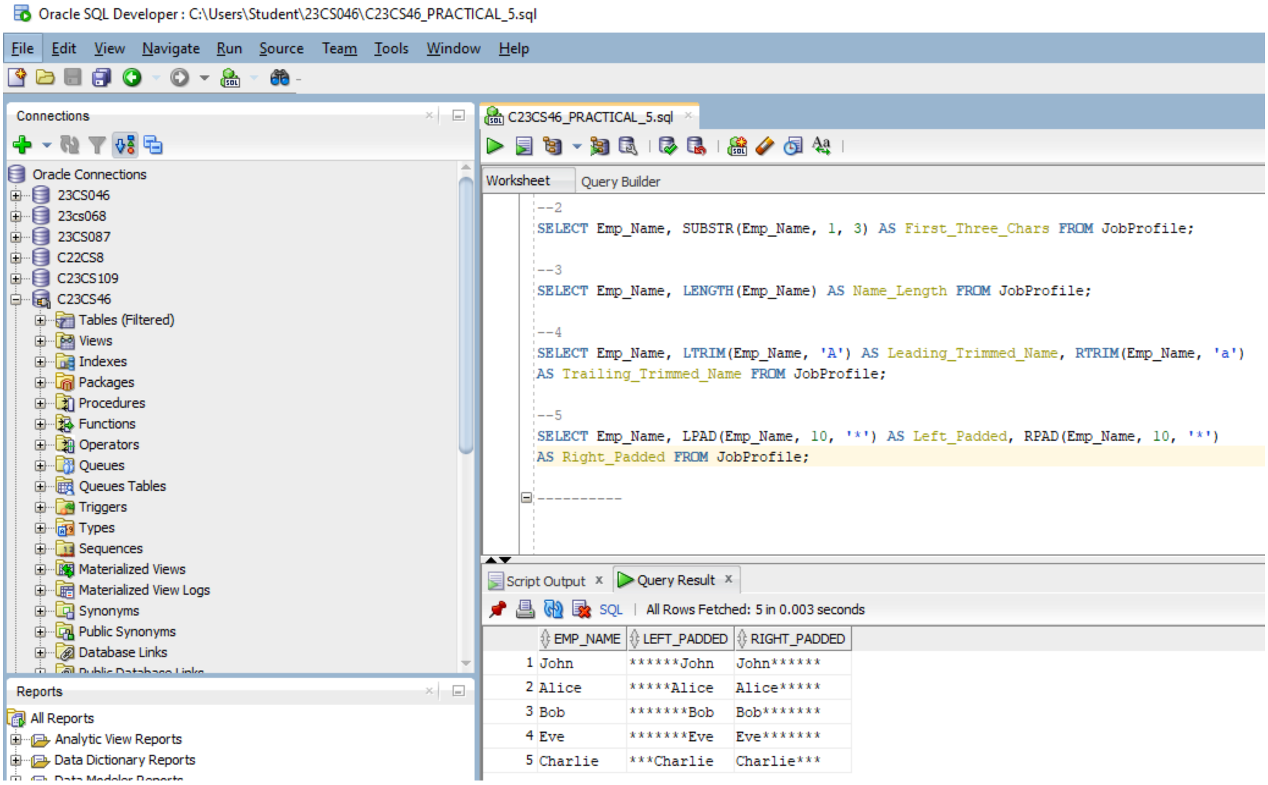


```sql
--2
SELECT Emp_Name, SUBSTR(Emp_Name, 1, 3) AS First_Three_Chars FROM JobProfile;

--3
SELECT Emp_Name, LENGTH(Emp_Name) AS Name_Length FROM JobProfile;

--4
SELECT Emp_Name, LTRIM(Emp_Name, 'A') AS Leading_Trimmed_Name, RTRIM(Emp_Name, 'a')
AS Trailing_Trimmed_Name FROM JobProfile;

--5
SELECT Emp_Name, LPAD(Emp_Name, 10, '') AS Left_Padded, RPAD(Emp_Name, 10, '')
AS Right_Padded FROM JobProfile;

----------
```

All Rows Fetched: 5 in 0.003 seconds

	EMP_NAME	LEADING_TRIMMED_NAME	TRAILING_TRIMMED_NAME
1	John	John	John
2	Alice	lice	Alice
3	Bob	Bob	Bob
4	Eve	Eve	Eve
5	Charlie	Charlie	Charlie

5. Pad employee first names with '*' on the left and right, ensuring a total length of 10.



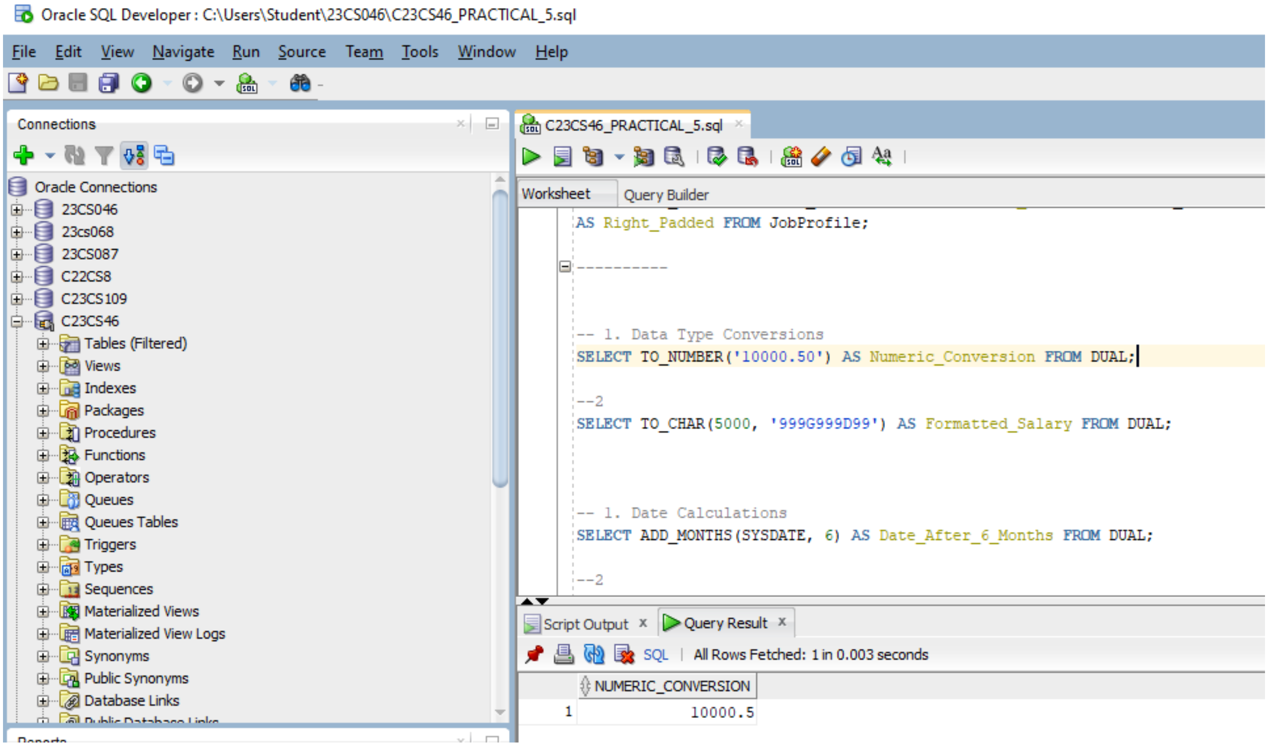
```sql
--2
SELECT Emp_Name, SUBSTR(Emp_Name, 1, 3) AS First_Three_Chars FROM JobProfile;

--3
SELECT Emp_Name, LENGTH(Emp_Name) AS Name_Length FROM JobProfile;

--4
SELECT Emp_Name, LTRIM(Emp_Name, 'A') AS Leading_Trimmed_Name, RTRIM(Emp_Name, 'a')
AS Trailing_Trimmed_Name FROM JobProfile;

--5
SELECT Emp_Name, LPAD(Emp_Name, 10, '*') AS Left_Padded, RPAD(Emp_Name, 10, '*')
AS Right_Padded FROM JobProfile;

----------
```

All Rows Fetched: 5 in 0.003 seconds

	EMP_NAME	LEFT_PADDED	RIGHT_PADDED
1	John	******John	John******
2	Alice	*****Alice	Alice*****
3	Bob	*******Bob	Bob*******
4	Eve	*******Eve	Eve*******
5	Charlie	***Charlie	Charlie***

The data migration team requires conversions between different data types:

1. Convert a string representation of a salary to a numeric format.

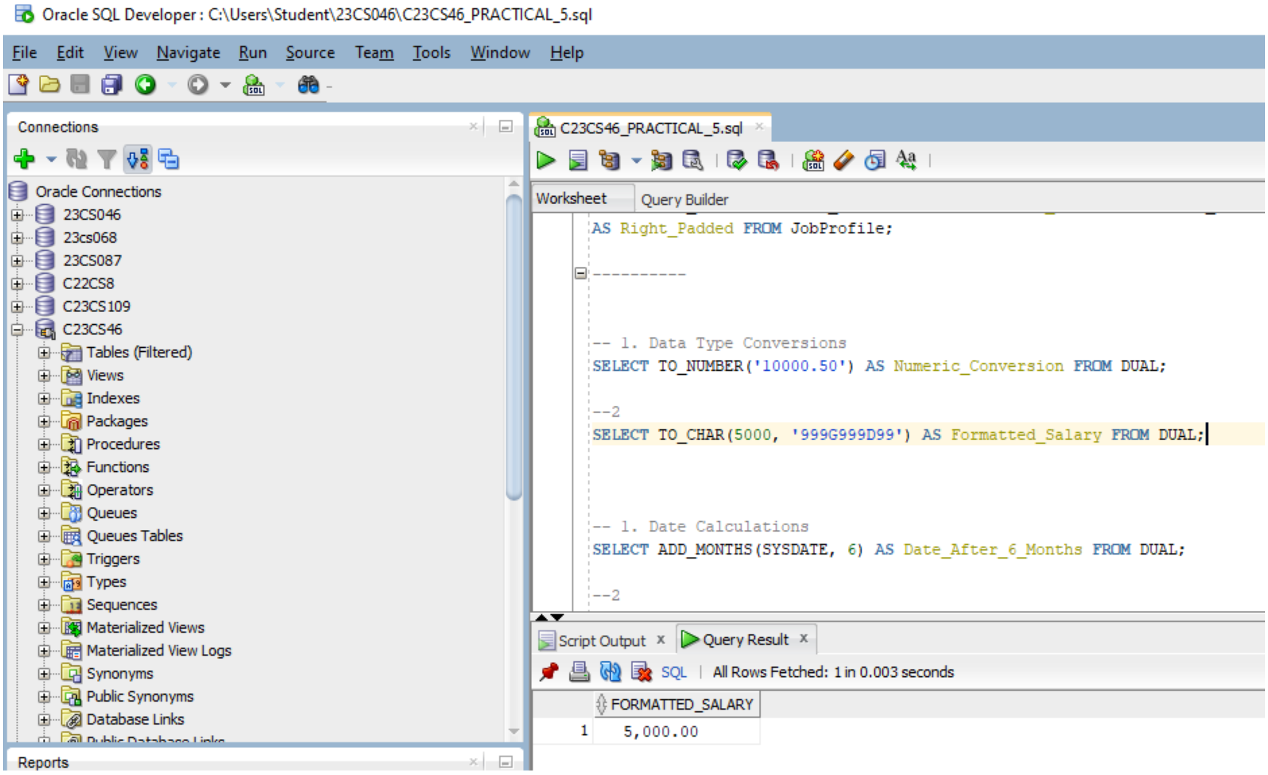


```sql
AS Right_Padded FROM JobProfile;

----------

-- 1. Data Type Conversions
SELECT TO_NUMBER('10000.50') AS Numeric_Conversion FROM DUAL;

--2
SELECT TO_CHAR(5000, '999G999D99') AS Formatted_Salary FROM DUAL;

-- 1. Date Calculations
SELECT ADD_MONTHS(SYSDATE, 6) AS Date_After_6_Months FROM DUAL;

--2
```

All Rows Fetched: 1 in 0.003 seconds

	NUMERIC_CONVERSION
1	10000.5

2. Format a numeric salary value into a string with specific formatting.



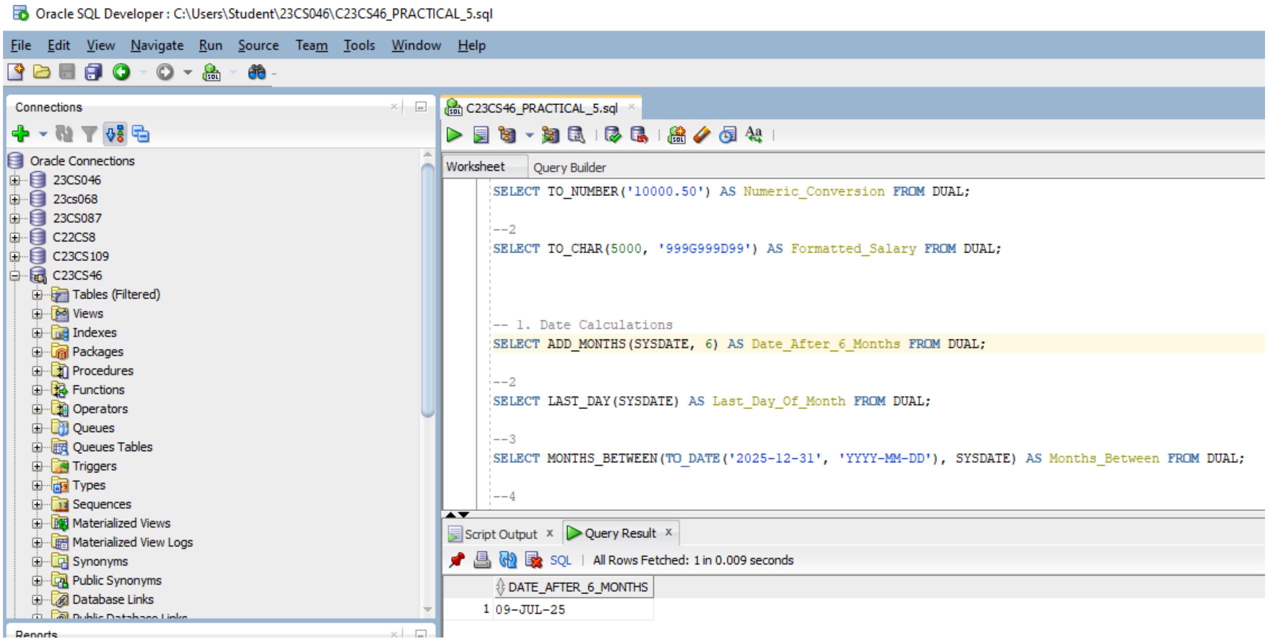
```sql
AS Right_Padded FROM JobProfile;

----------

-- 1. Data Type Conversions
SELECT TO_NUMBER('10000.50') AS Numeric_Conversion FROM DUAL;

--2
SELECT TO_CHAR(5000, '999G999D99') AS Formatted_Salary FROM DUAL;

-- 1. Date Calculations
SELECT ADD_MONTHS(SYSDATE, 6) AS Date_After_6_Months FROM DUAL;

--2
```

All Rows Fetched: 1 in 0.003 seconds

	FORMATTED_SALARY
1	5,000.00

To assist in employee scheduling, perform the following:

1. Calculate the date after adding 6 months to the current date.

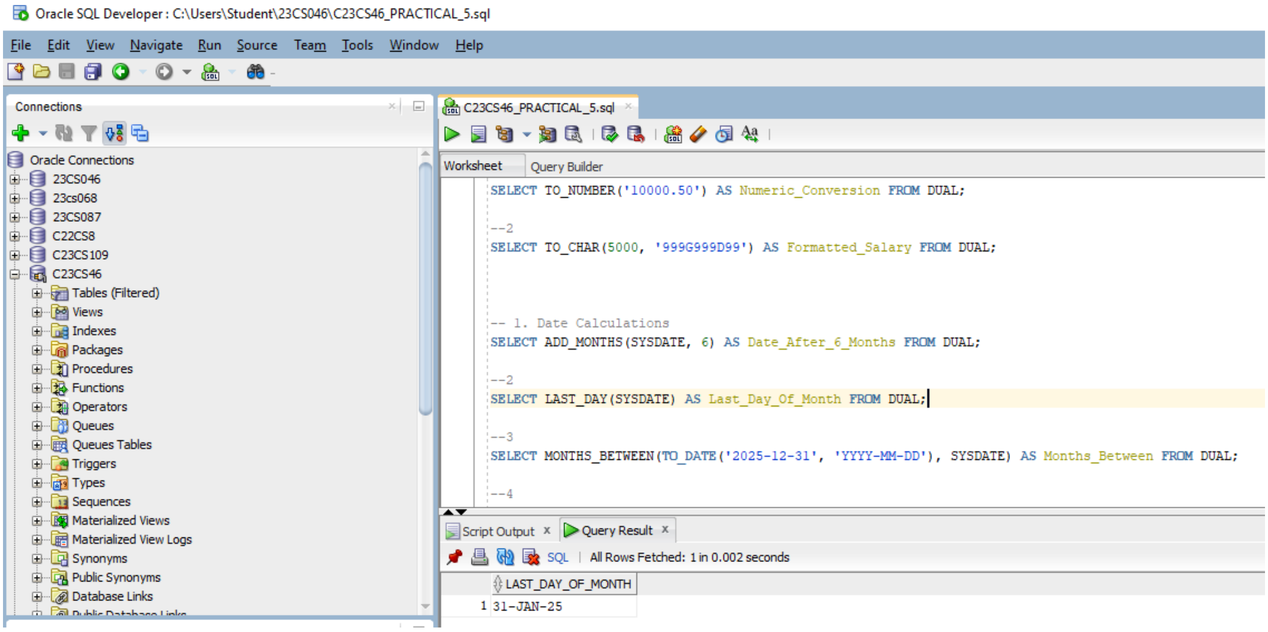


```
SELECT TO_NUMBER('10000.50') AS Numeric_Conversion FROM DUAL;

--2
SELECT TO_CHAR(5000, '999G999D99') AS Formatted_Salary FROM DUAL;


-- 1. Date Calculations
SELECT ADD_MONTHS(SYSDATE, 6) AS Date_After_6_Months FROM DUAL;

--2
SELECT LAST_DAY(SYSDATE) AS Last_Day_Of_Month FROM DUAL;

--3
SELECT MONTHS_BETWEEN(TO_DATE('2025-12-31', 'YYYY-MM-DD'), SYSDATE) AS Months_Between FROM DUAL;

--4
```

All Rows Fetched: 1 in 0.009 seconds

```
  DATE_AFTER_6_MONTHS
1 09-JUL-25
```

2. Retrieve the last day of the current month.

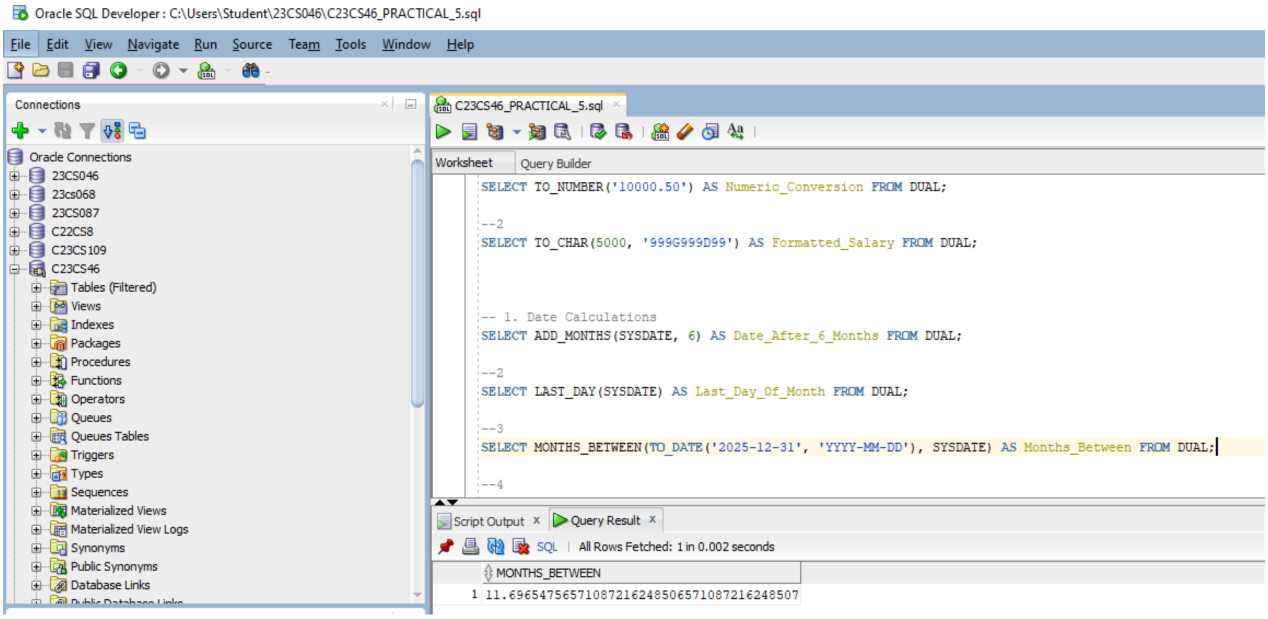


```
SELECT TO_NUMBER('10000.50') AS Numeric_Conversion FROM DUAL;

--2
SELECT TO_CHAR(5000, '999G999D99') AS Formatted_Salary FROM DUAL;


-- 1. Date Calculations
SELECT ADD_MONTHS(SYSDATE, 6) AS Date_After_6_Months FROM DUAL;

--2
SELECT LAST_DAY(SYSDATE) AS Last_Day_Of_Month FROM DUAL;

--3
SELECT MONTHS_BETWEEN(TO_DATE('2025-12-31', 'YYYY-MM-DD'), SYSDATE) AS Months_Between FROM DUAL;

--4
```

All Rows Fetched: 1 in 0.002 seconds

```
  LAST_DAY_OF_MONTH
1 31-JAN-25
```

3. Calculate the number of months between two dates.

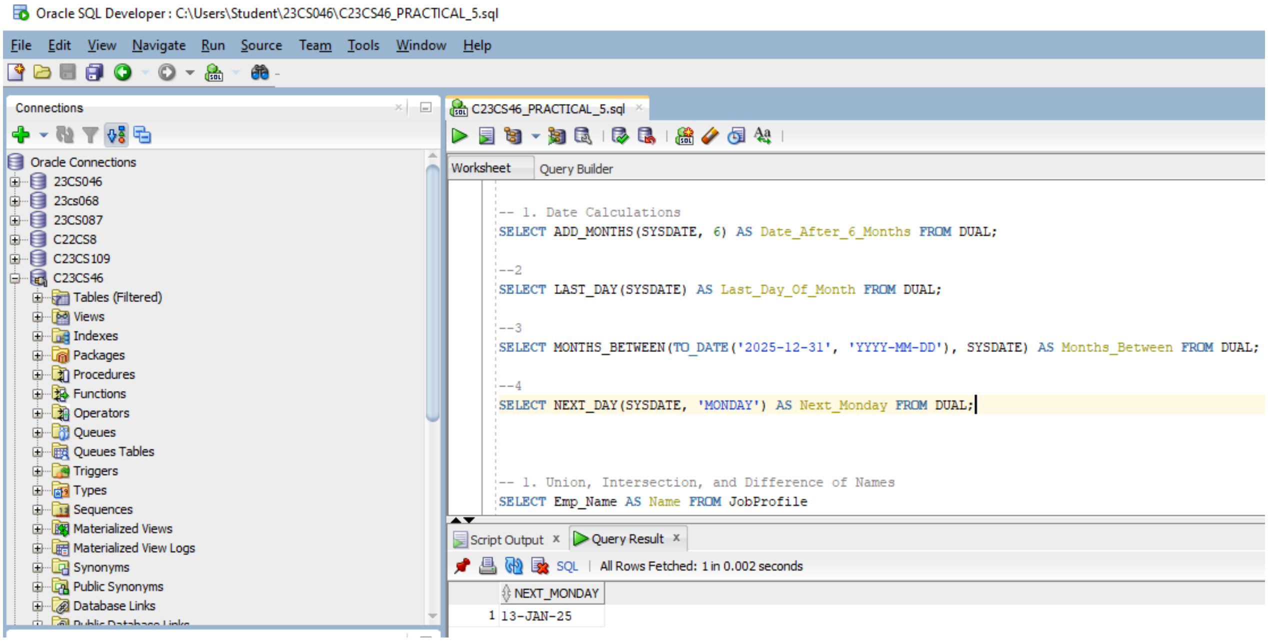


```sql
SELECT TO_NUMBER('10000.50') AS Numeric_Conversion FROM DUAL;

--2
SELECT TO_CHAR(5000, '999G999D99') AS Formatted_Salary FROM DUAL;



-- 1. Date Calculations
SELECT ADD_MONTHS(SYSDATE, 6) AS Date_After_6_Months FROM DUAL;

--2
SELECT LAST_DAY(SYSDATE) AS Last_Day_Of_Month FROM DUAL;

--3
SELECT MONTHS_BETWEEN(TO_DATE('2025-12-31', 'YYYY-MM-DD'), SYSDATE) AS Months_Between FROM DUAL;

--4
```

All Rows Fetched: 1 in 0.002 seconds

```
  MONTHS_BETWEEN
1 11.696547565710872162485065710872162485 07
```

4. Find the next Monday from the current date.



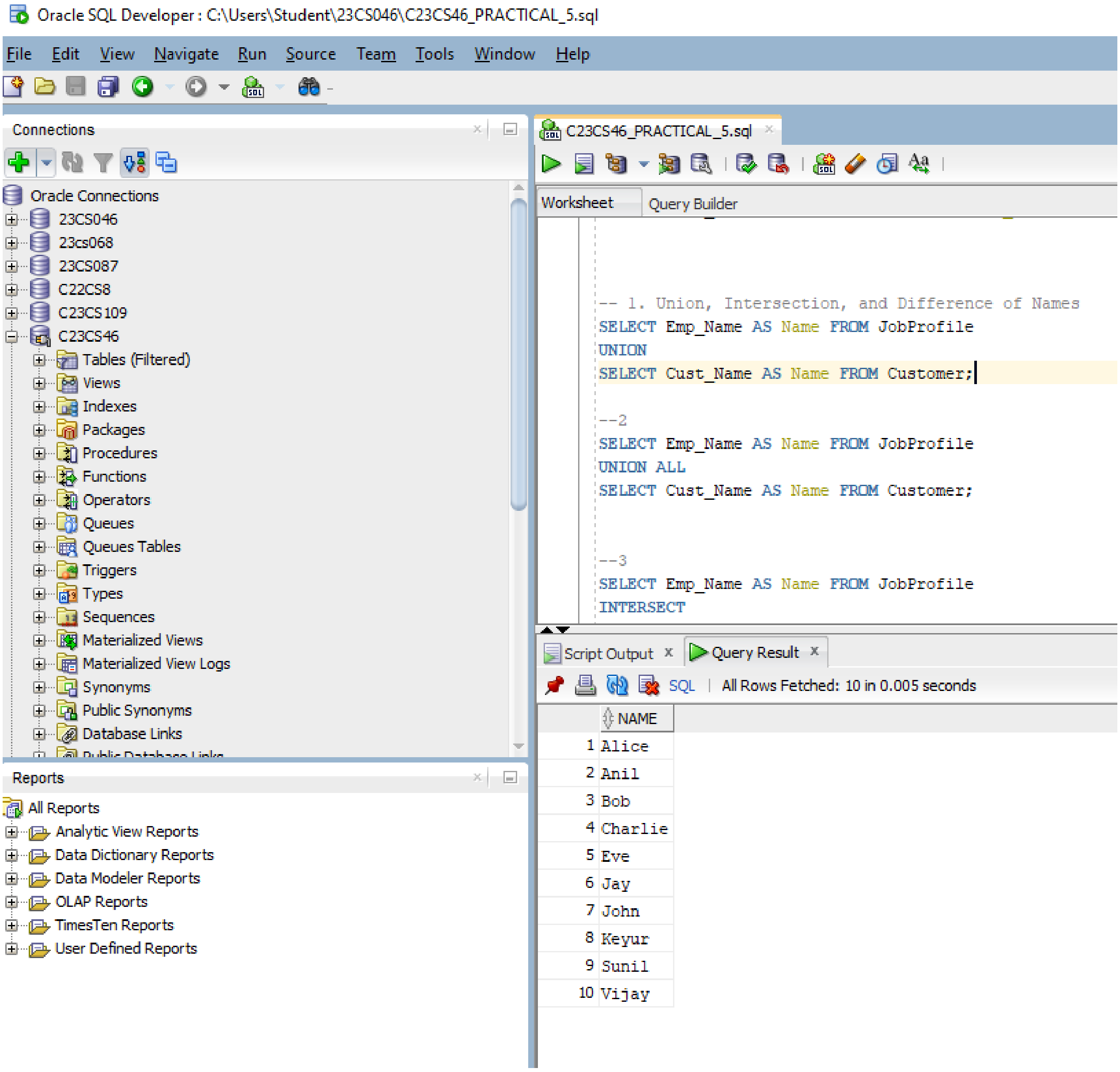
```sql

-- 1. Date Calculations
SELECT ADD_MONTHS(SYSDATE, 6) AS Date_After_6_Months FROM DUAL;

--2
SELECT LAST_DAY(SYSDATE) AS Last_Day_Of_Month FROM DUAL;

--3
SELECT MONTHS_BETWEEN(TO_DATE('2025-12-31', 'YYYY-MM-DD'), SYSDATE) AS Months_Between FROM DUAL;

--4
SELECT NEXT_DAY(SYSDATE, 'MONDAY') AS Next_Monday FROM DUAL;



-- 1. Union, Intersection, and Difference of Names
SELECT Emp_Name AS Name FROM JobProfile
```

All Rows Fetched: 1 in 0.002 seconds

```
  NEXT_MONDAY
1 13-JAN-25
```

To identify overlaps and differences between employees and customers, write SQL queries to:

1. Retrieve the union of first names from employees and customers.

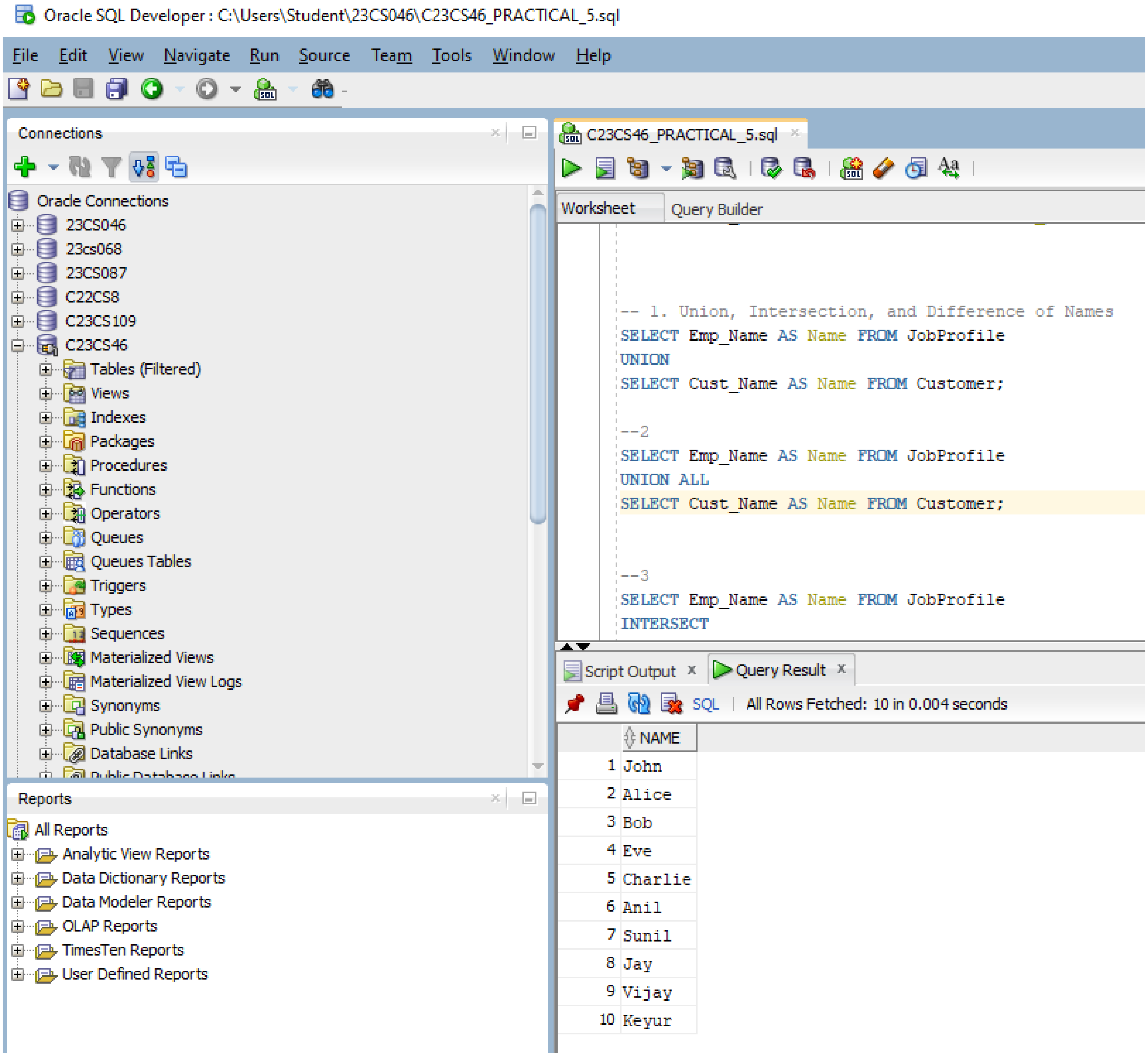


```
-- 1. Union, Intersection, and Difference of Names
SELECT Emp_Name AS Name FROM JobProfile
UNION
SELECT Cust_Name AS Name FROM Customer;

--2
SELECT Emp_Name AS Name FROM JobProfile
UNION ALL
SELECT Cust_Name AS Name FROM Customer;


--3
SELECT Emp_Name AS Name FROM JobProfile
INTERSECT
```

All Rows Fetched: 10 in 0.005 seconds

	NAME
1	Alice
2	Anil
3	Bob
4	Charlie
5	Eve
6	Jay
7	John
8	Keyur
9	Sunil
10	Vijay

2. Retrieve the union of first names (including duplicates).

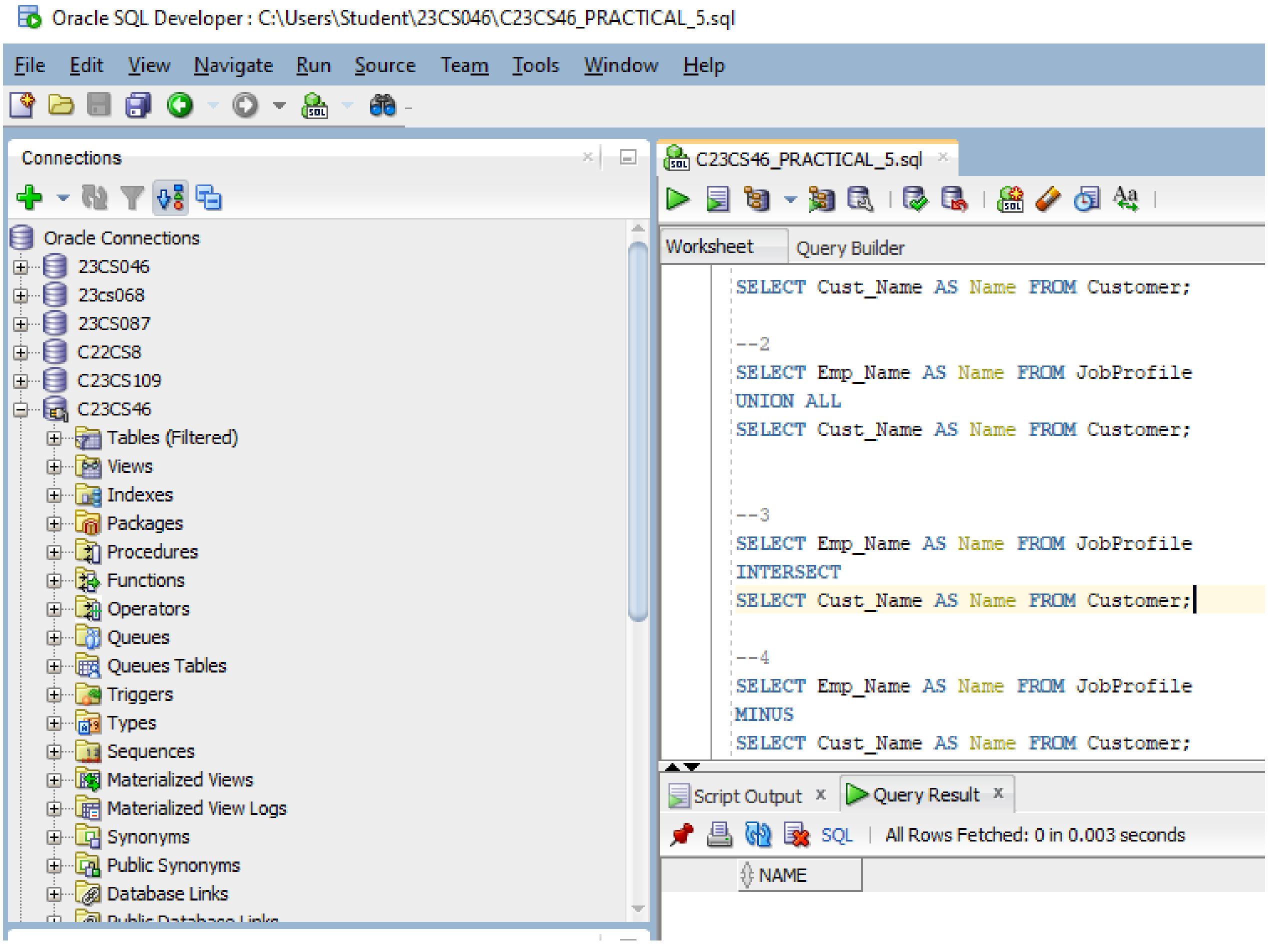


```sql
-- 1. Union, Intersection, and Difference of Names
SELECT Emp_Name AS Name FROM JobProfile
UNION
SELECT Cust_Name AS Name FROM Customer;

--2
SELECT Emp_Name AS Name FROM JobProfile
UNION ALL
SELECT Cust_Name AS Name FROM Customer;


--3
SELECT Emp_Name AS Name FROM JobProfile
INTERSECT
```

All Rows Fetched: 10 in 0.004 seconds

	NAME
1	John
2	Alice
3	Bob
4	Eve
5	Charlie
6	Anil
7	Sunil
8	Jay
9	Vijay
10	Keyur

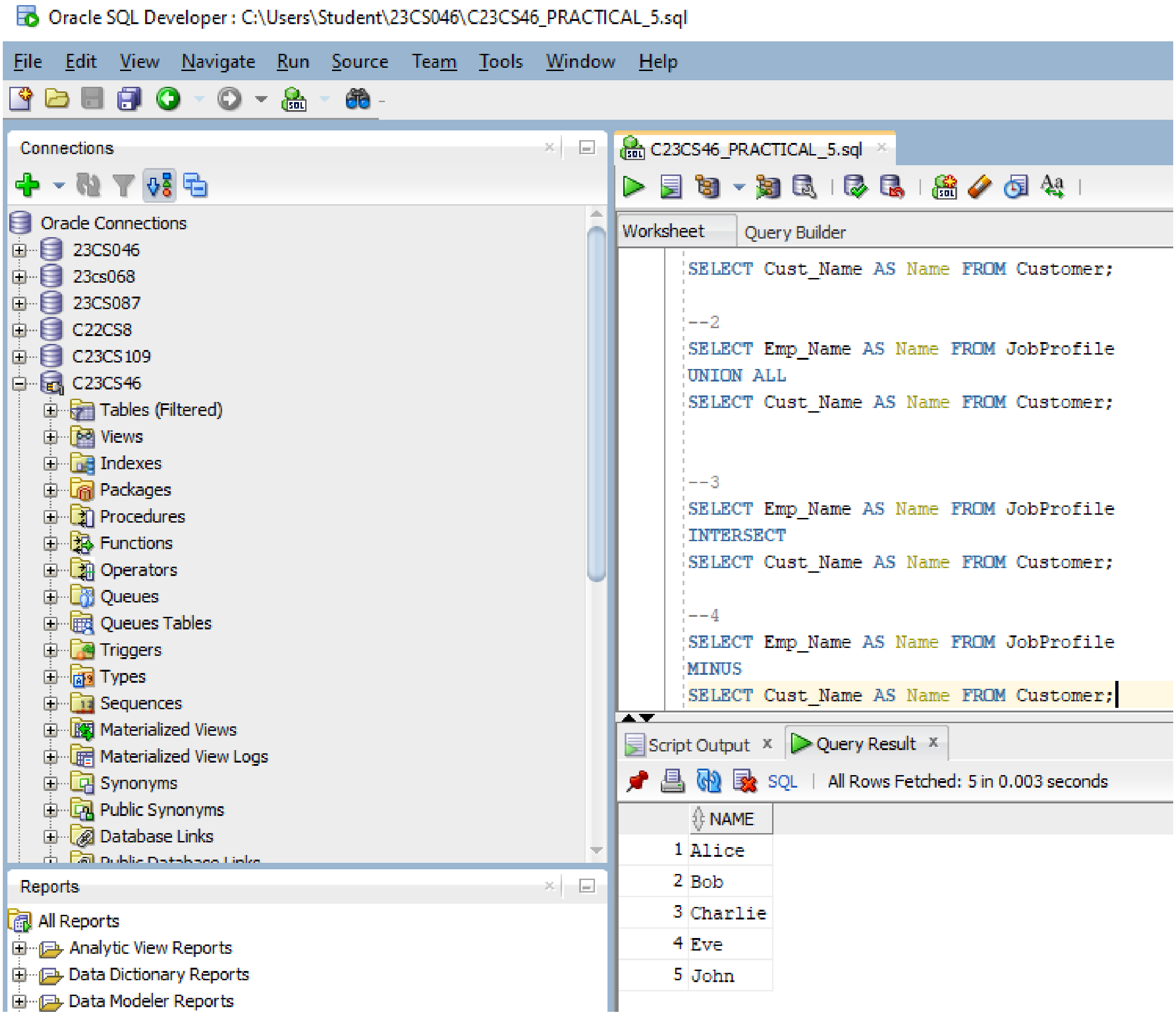
3. Find the intersection of first names from employees and customers.



```
SELECT Cust_Name AS Name FROM Customer;

--2
SELECT Emp_Name AS Name FROM JobProfile
UNION ALL
SELECT Cust_Name AS Name FROM Customer;


--3
SELECT Emp_Name AS Name FROM JobProfile
INTERSECT
SELECT Cust_Name AS Name FROM Customer;

--4
SELECT Emp_Name AS Name FROM JobProfile
MINUS
SELECT Cust_Name AS Name FROM Customer;
```

All Rows Fetched: 0 in 0.003 seconds

4. Identify first names present in the employees table but not in customers.



```sql
SELECT Cust_Name AS Name FROM Customer;

--2
SELECT Emp_Name AS Name FROM JobProfile
UNION ALL
SELECT Cust_Name AS Name FROM Customer;


--3
SELECT Emp_Name AS Name FROM JobProfile
INTERSECT
SELECT Cust_Name AS Name FROM Customer;

--4
SELECT Emp_Name AS Name FROM JobProfile
MINUS
SELECT Cust_Name AS Name FROM Customer;
```

All Rows Fetched: 5 in 0.003 seconds

	NAME
1	Alice
2	Bob
3	Charlie
4	Eve
5	John