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:

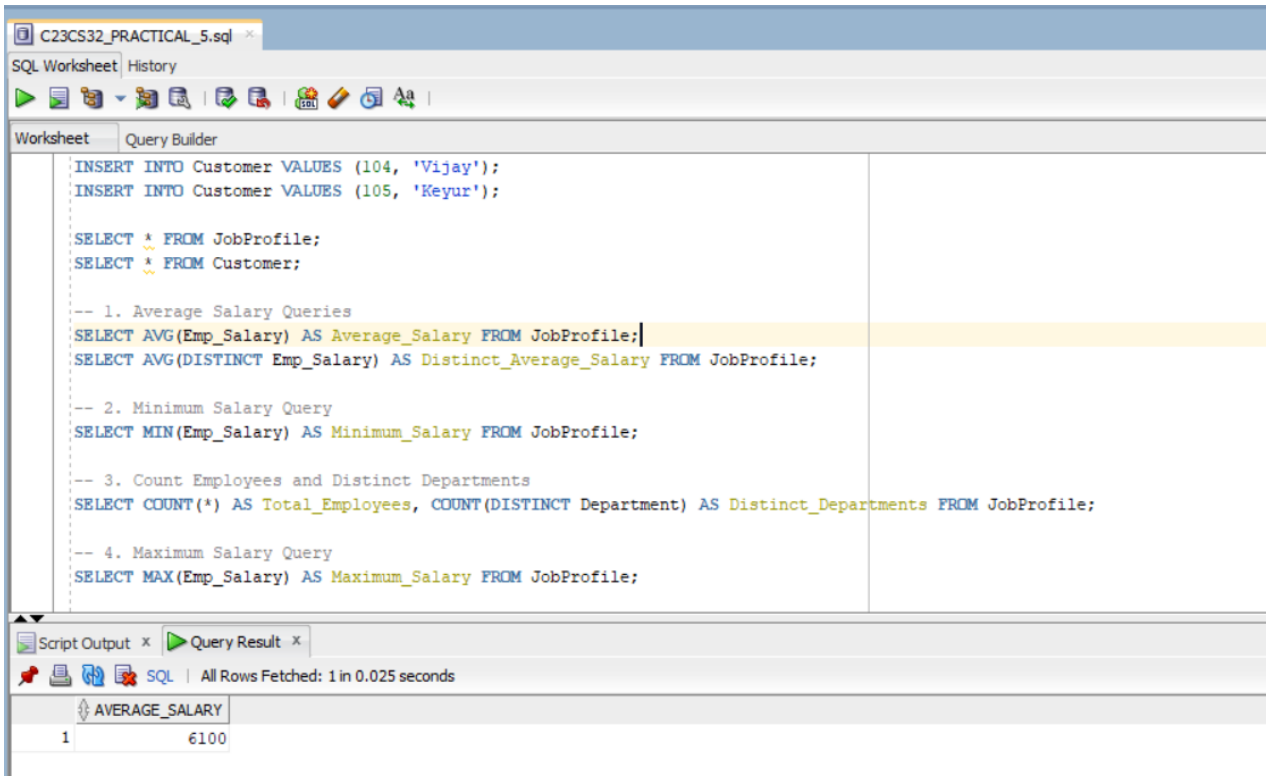
1. **JobProfile Table**: Stores details of employees and their job roles.
   - Emp_ID (Primary Key)
   - Emp_Name (Not Null)
   - Emp_Salary (Not Null, Check: Greater than zero)
   - Job_ID (Unique)
   - Department

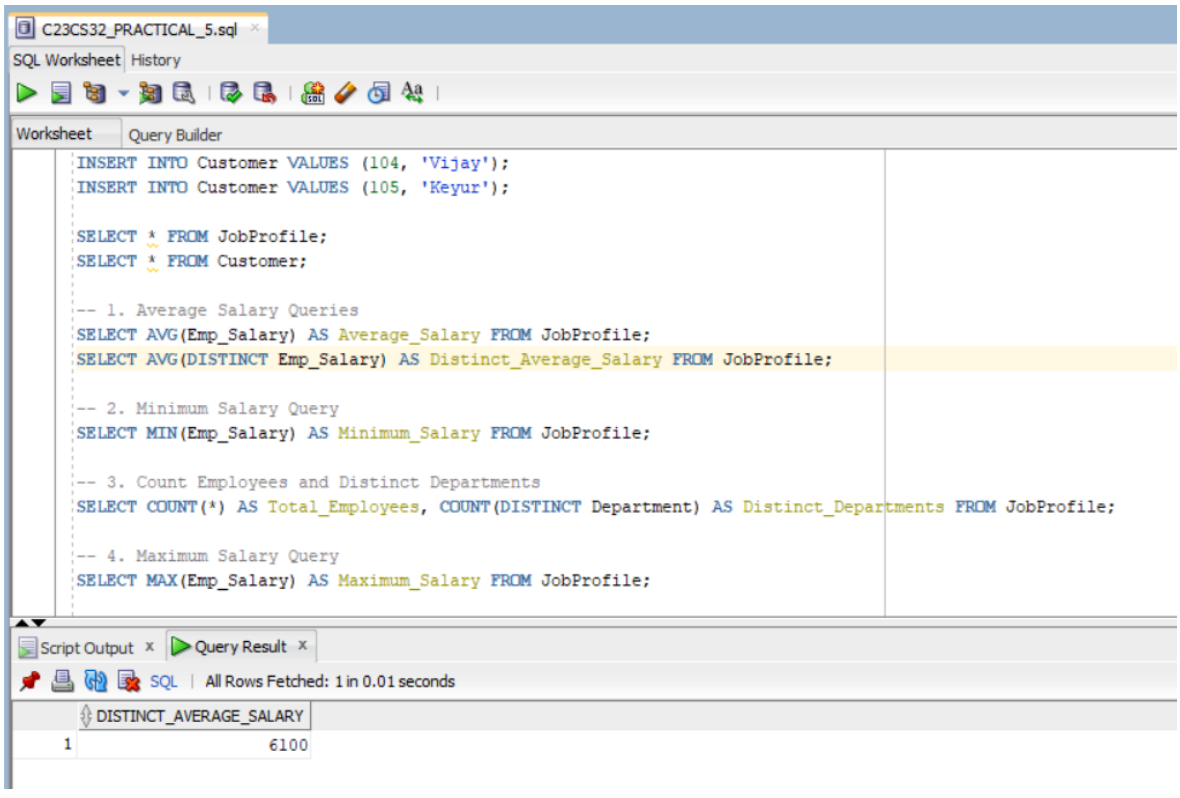
2. **Customer Table**: Stores customer details.
   - Cust_ID (Primary Key)
   - Cust_Name (Not Null)

Tasks:-

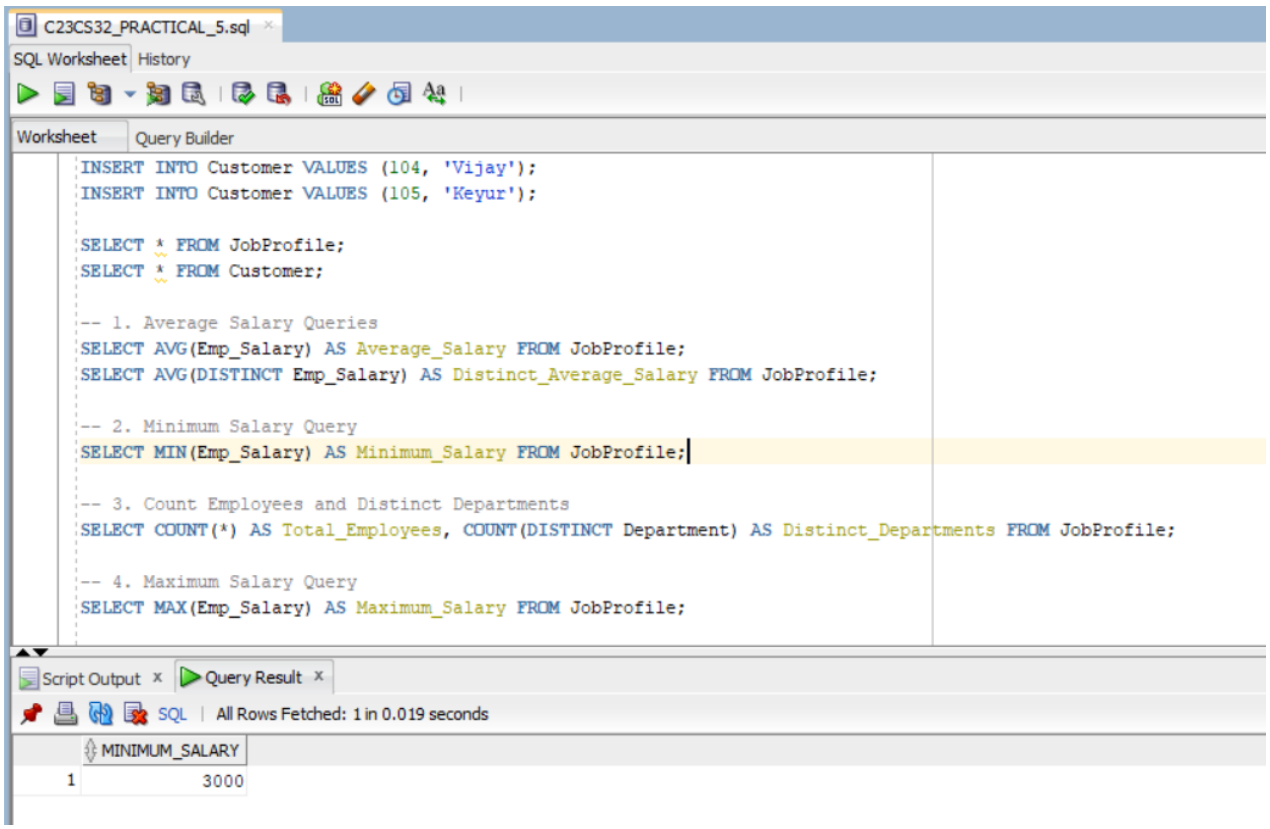
The HR department wants to analyze employee salaries for better planning. Write SQL queries to:

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

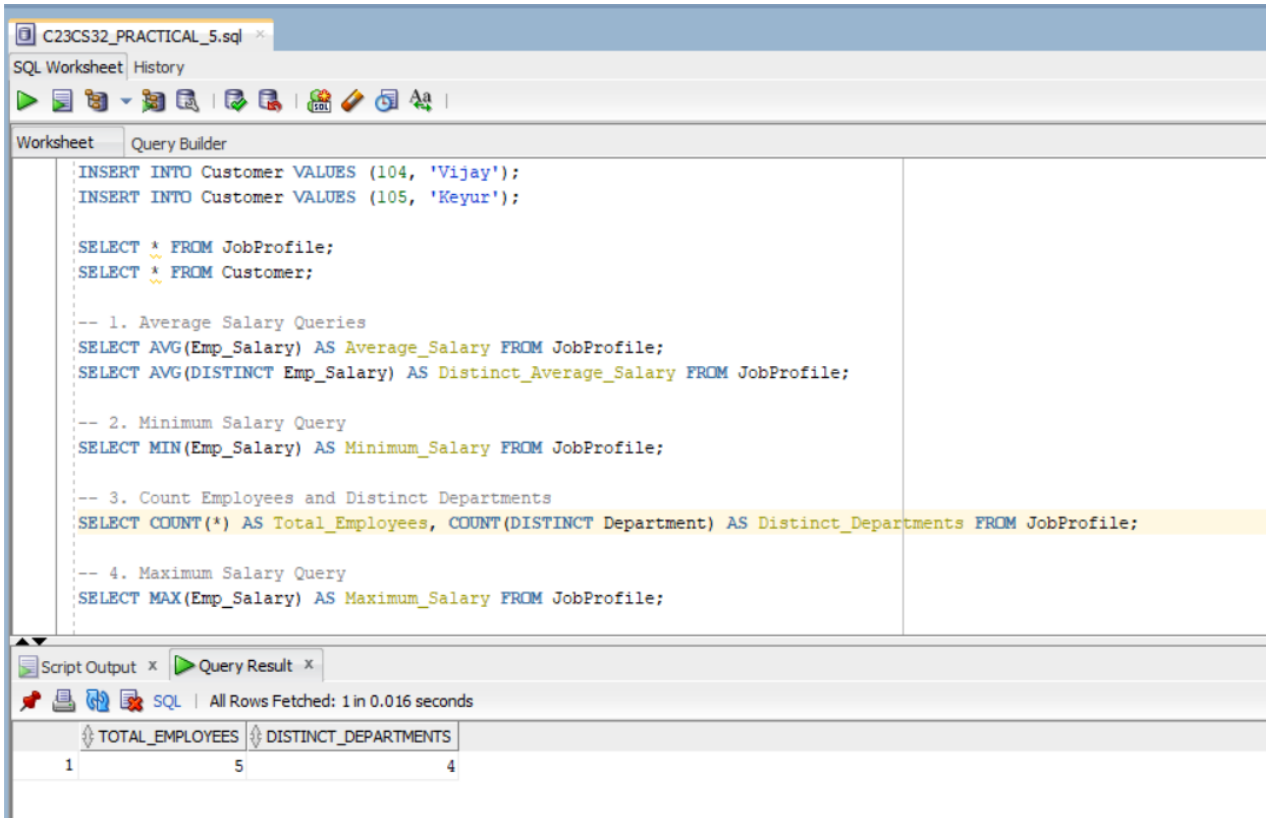




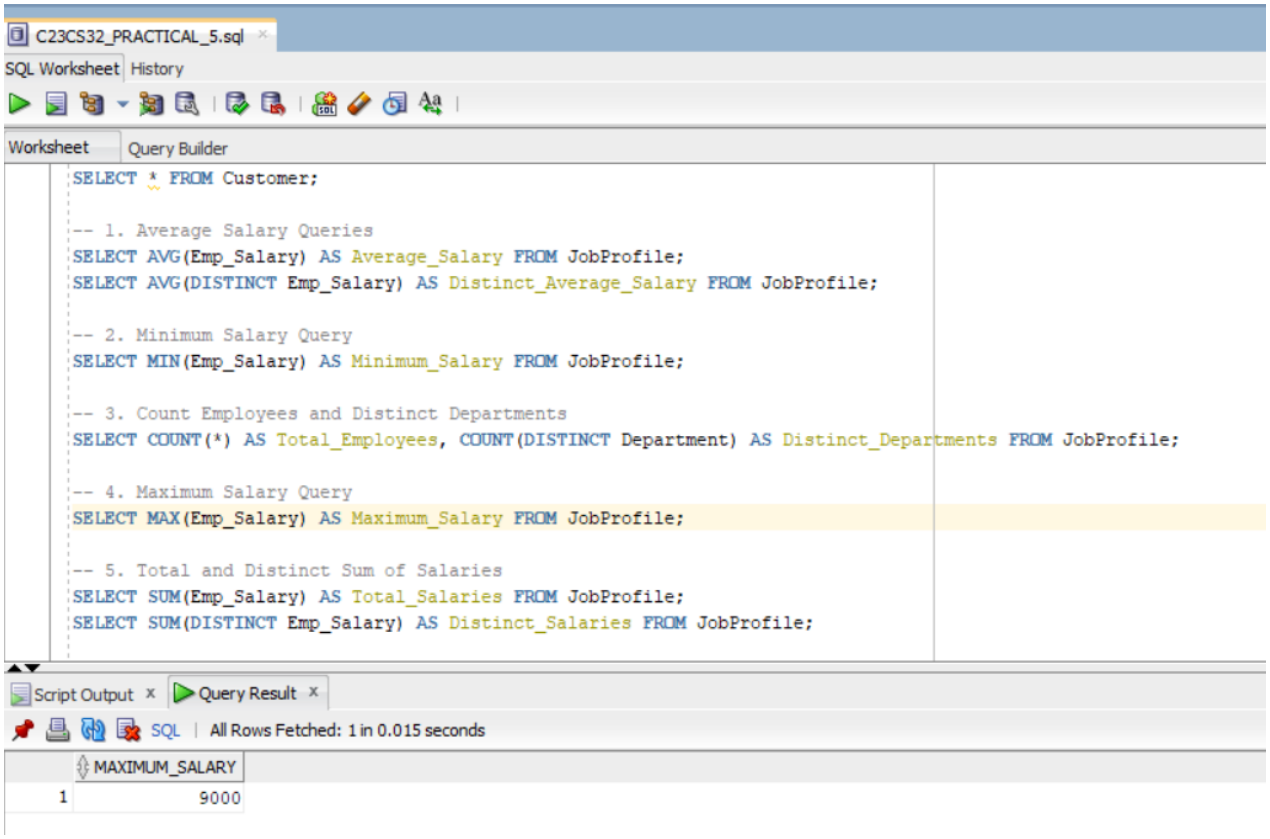
2. Retrieve the minimum salary from the JobProfile table.



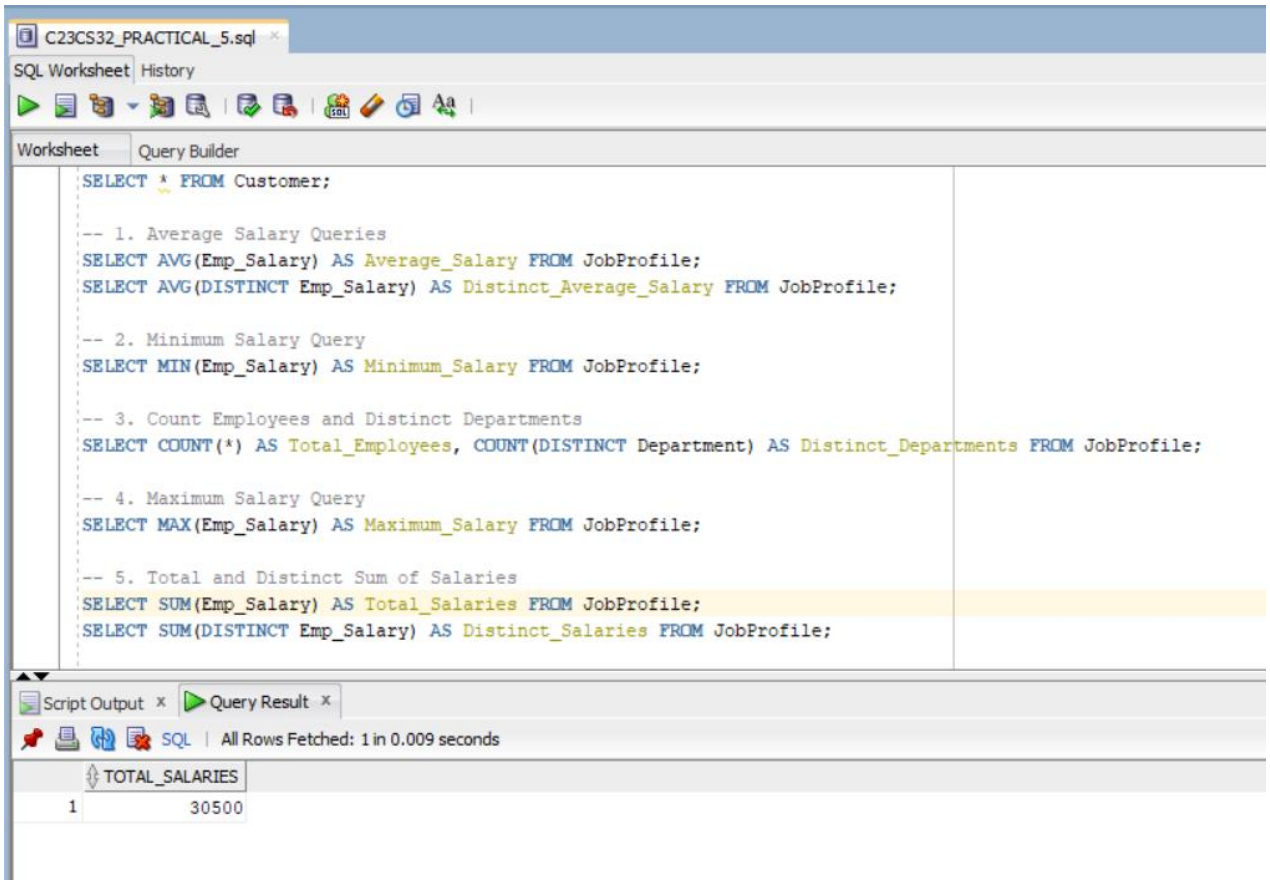
3. Count the total number of employees and distinct departments.

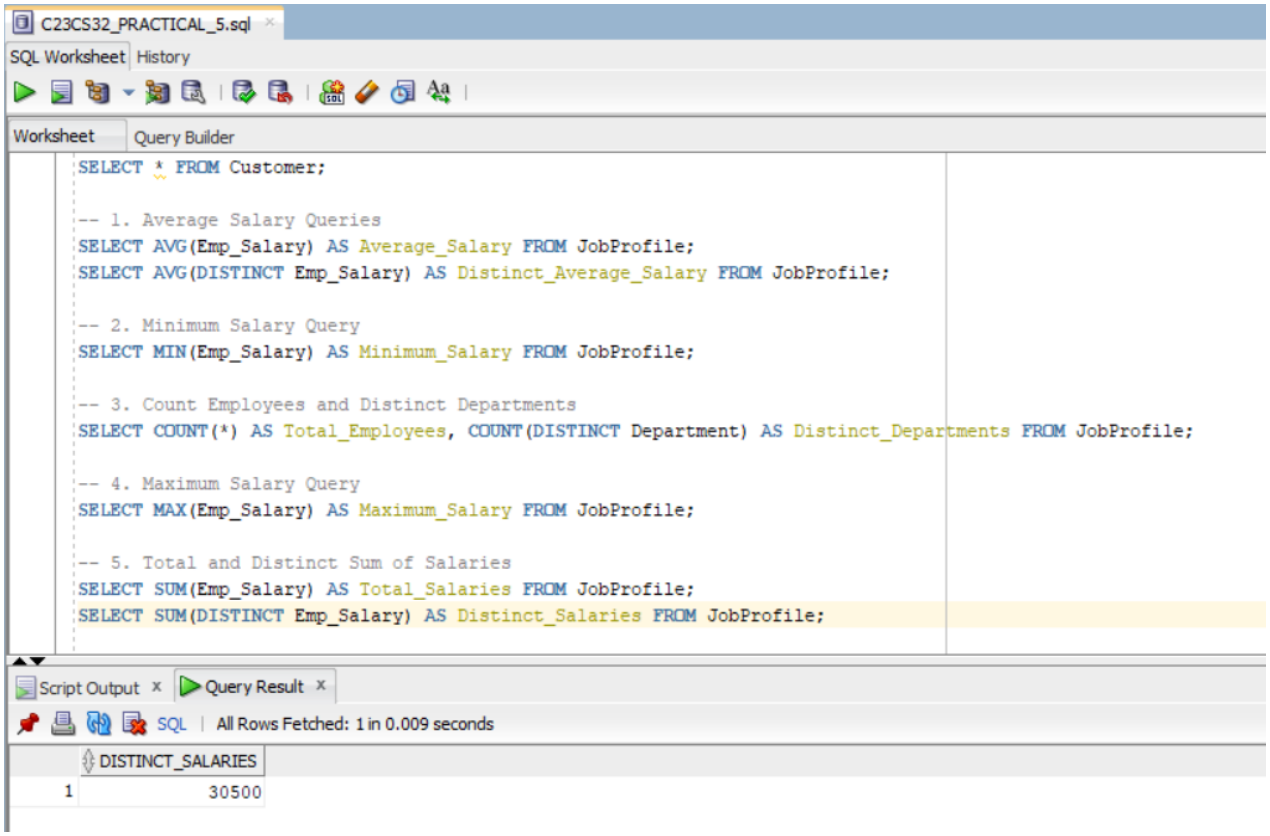


4. Retrieve the maximum salary from the JobProfile table.



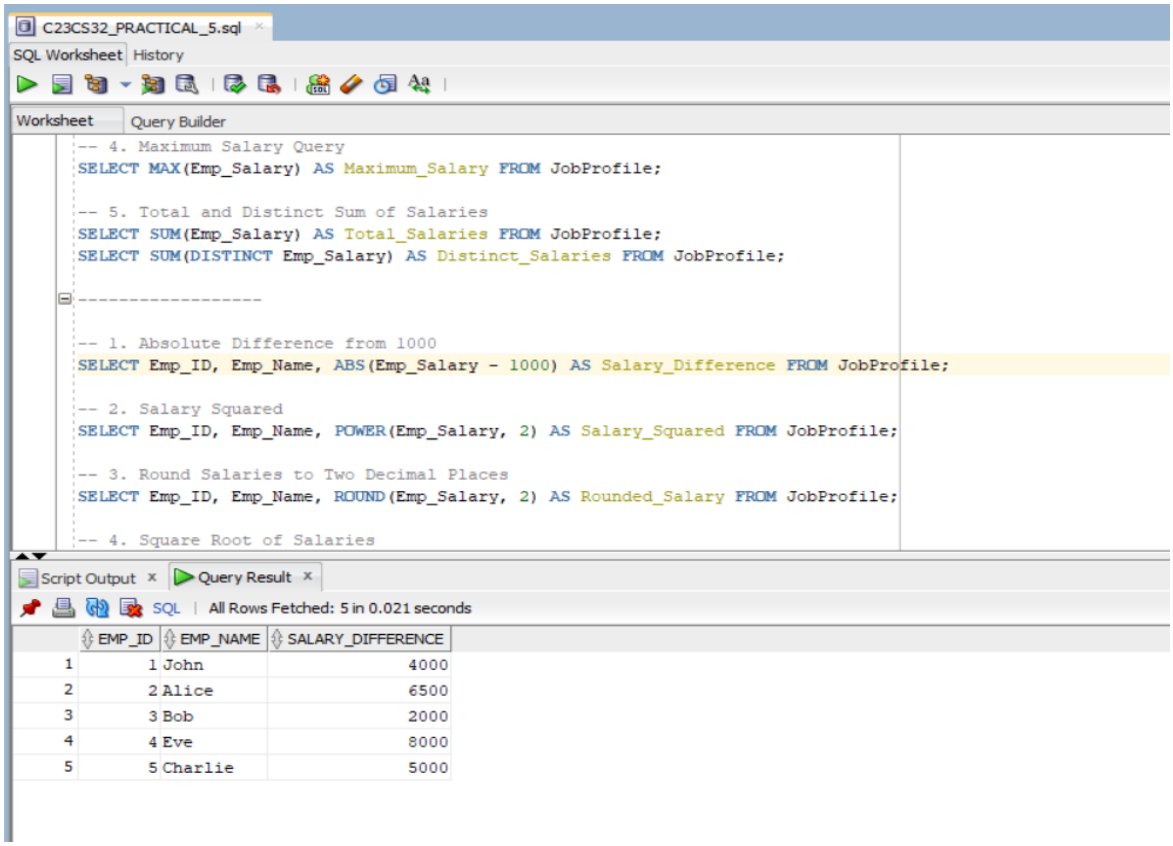
5. Calculate the total and distinct sum of all salaries.





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.



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

C23CS32_PRACTICAL_5.sql

SQL Worksheet History

Worksheet Query Builder

```
-- 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;

-- 4. Square Root of Salaries
```

Script Output | Query Result

SQL | All Rows Fetched: 5 in 0.012 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.

C23CS32_PRACTICAL_5.sql

SQL Worksheet History

Worksheet Query Builder

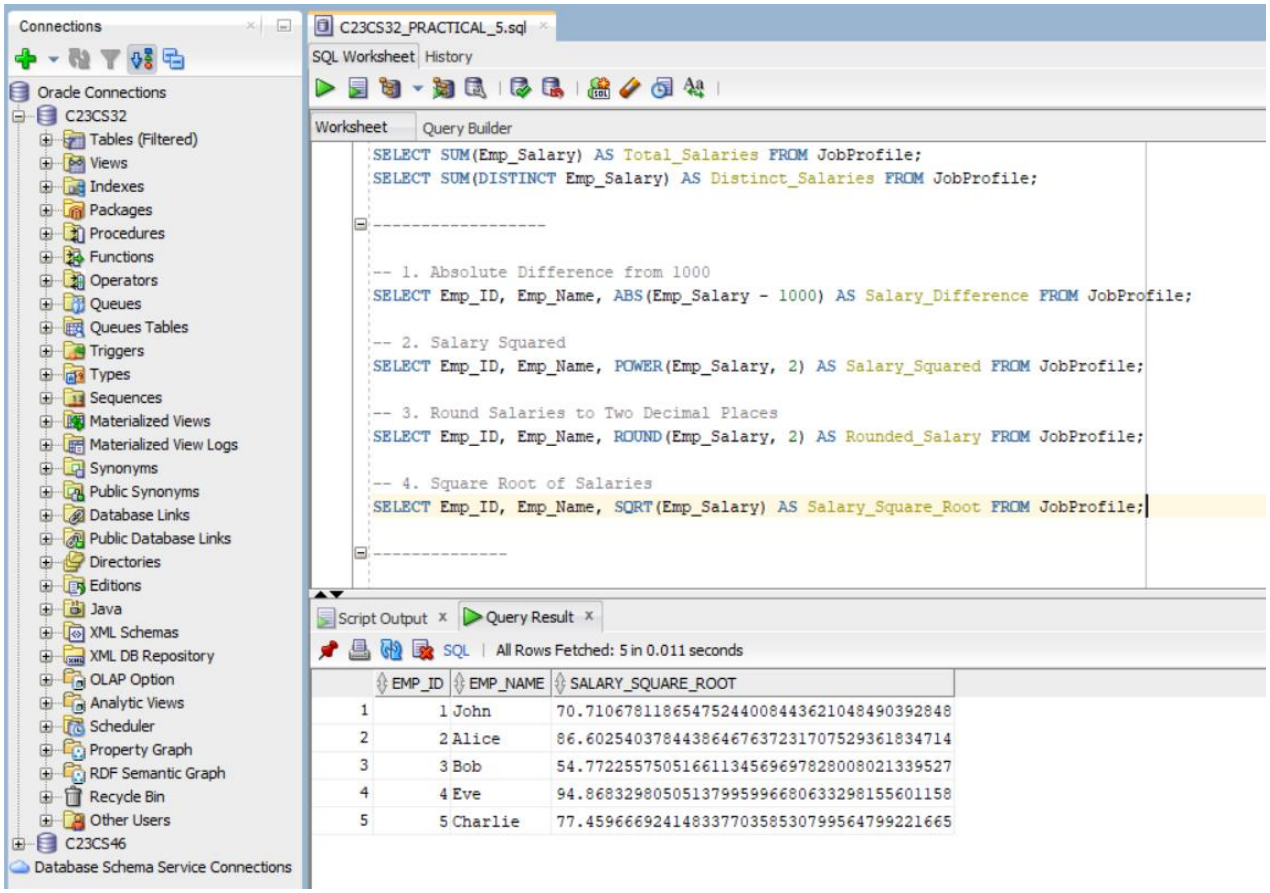
```
-- 4. Maximum Salary Query
SELECT MAX(Emp_Salary) AS Maximum_Salary FROM JobProfile;

-- 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;

-- 4. Square Root of Salaries
```

Script Output | Query Result

SQL | All Rows Fetched: 5 in 0.011 seconds

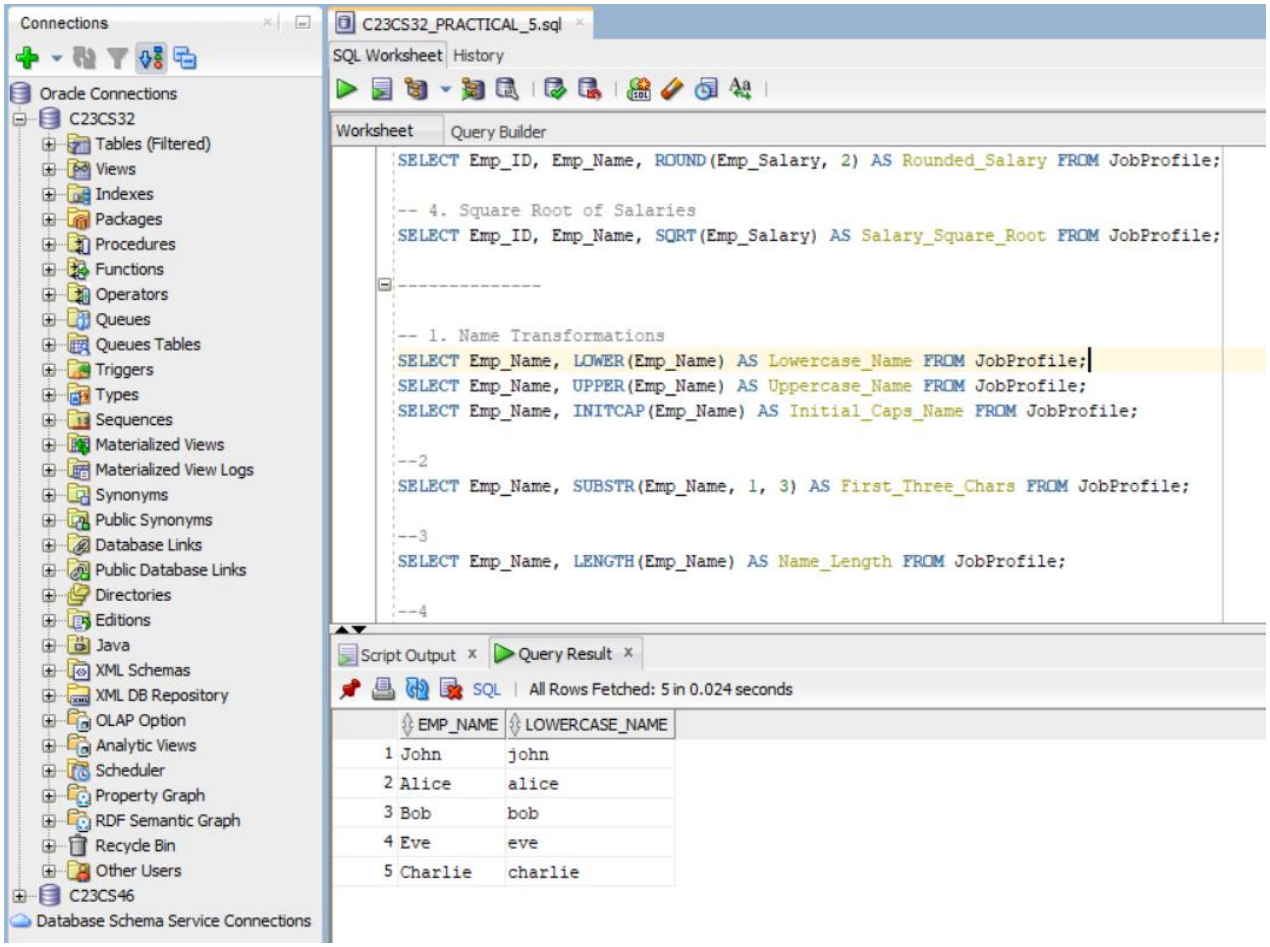
	EMP_ID	EMP_NAME	ROUNDED_SALARY
1	1	John	5000
2	2	Alice	7500
3	3	Bob	3000
4	4	Eve	9000
5	5	Charlie	6000

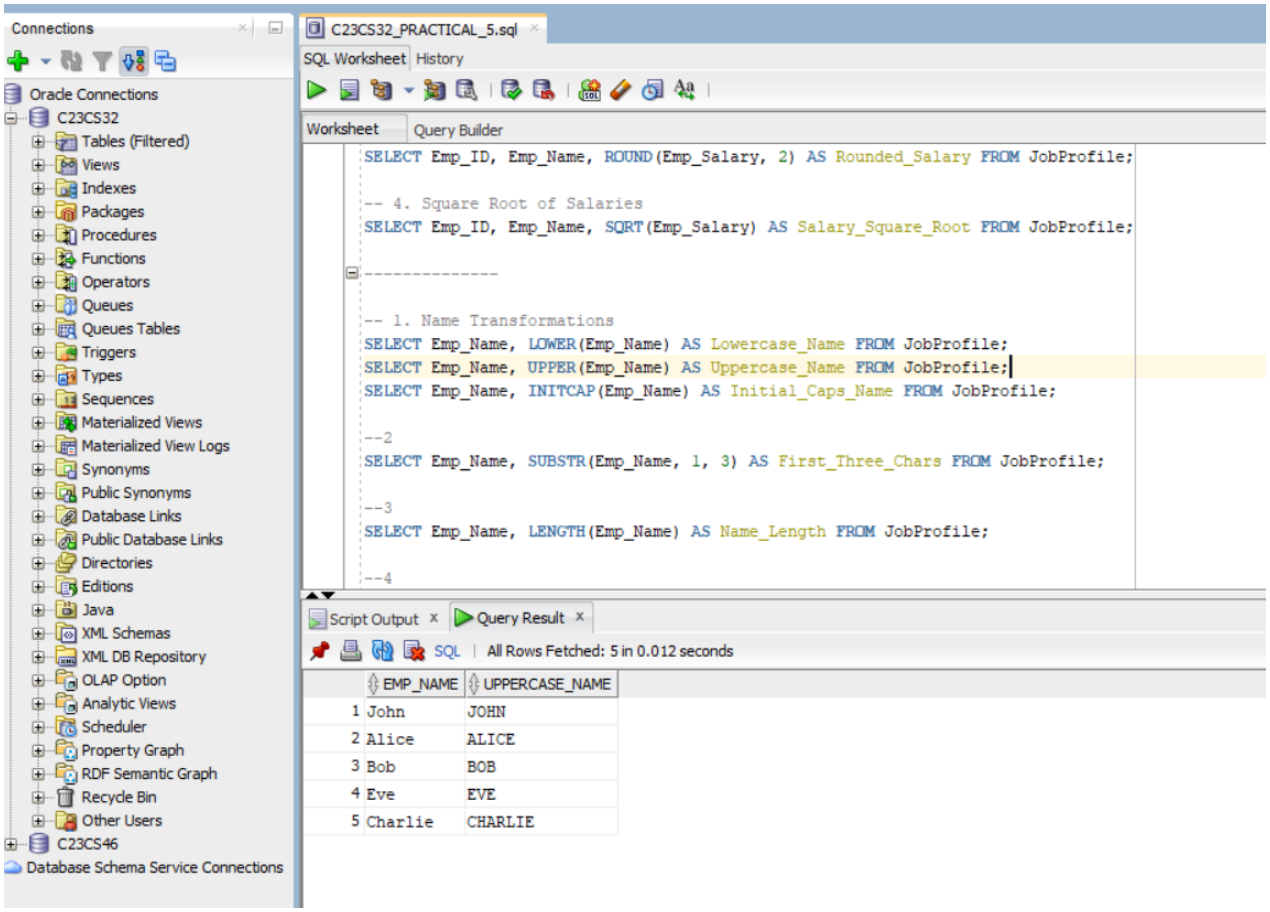
4. Find the square root of salaries.

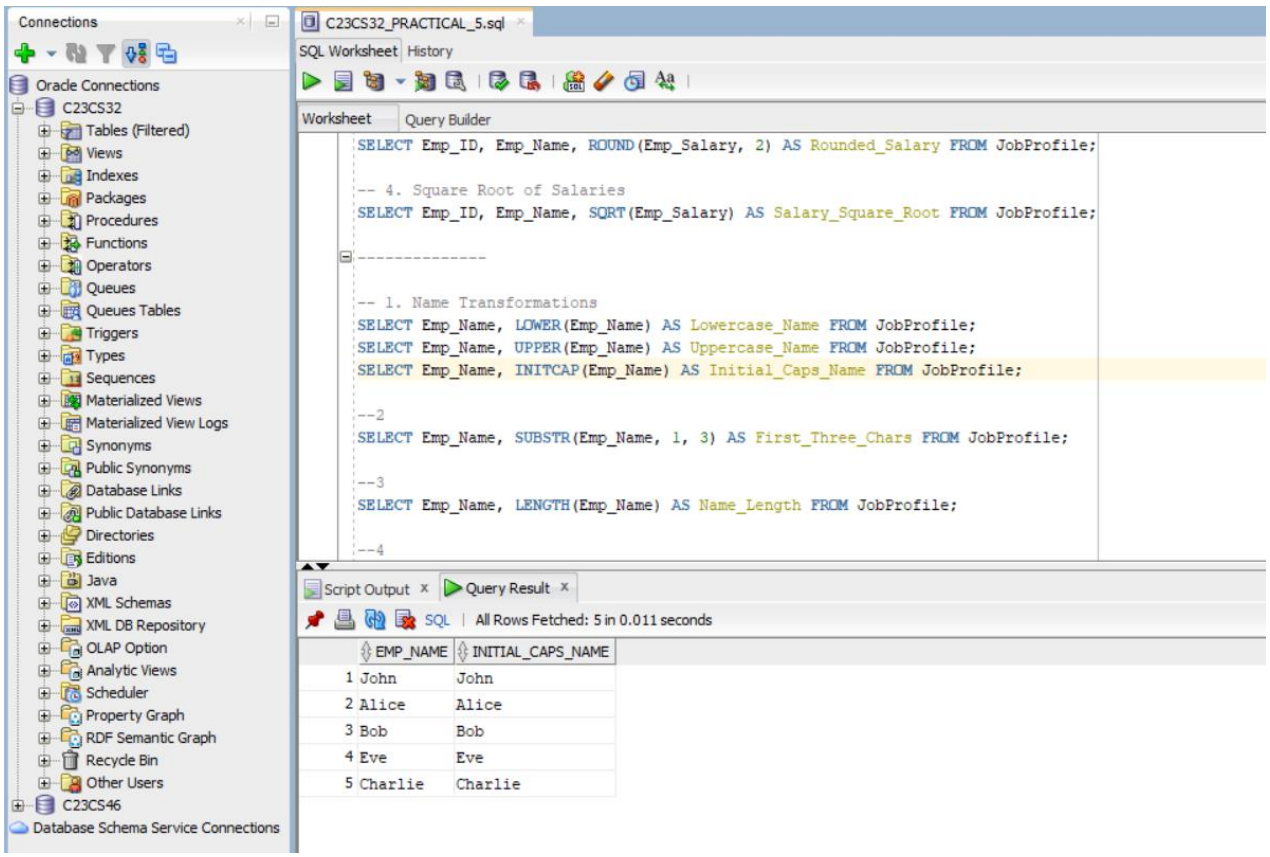


To ensure uniformity in names across systems, perform the following:

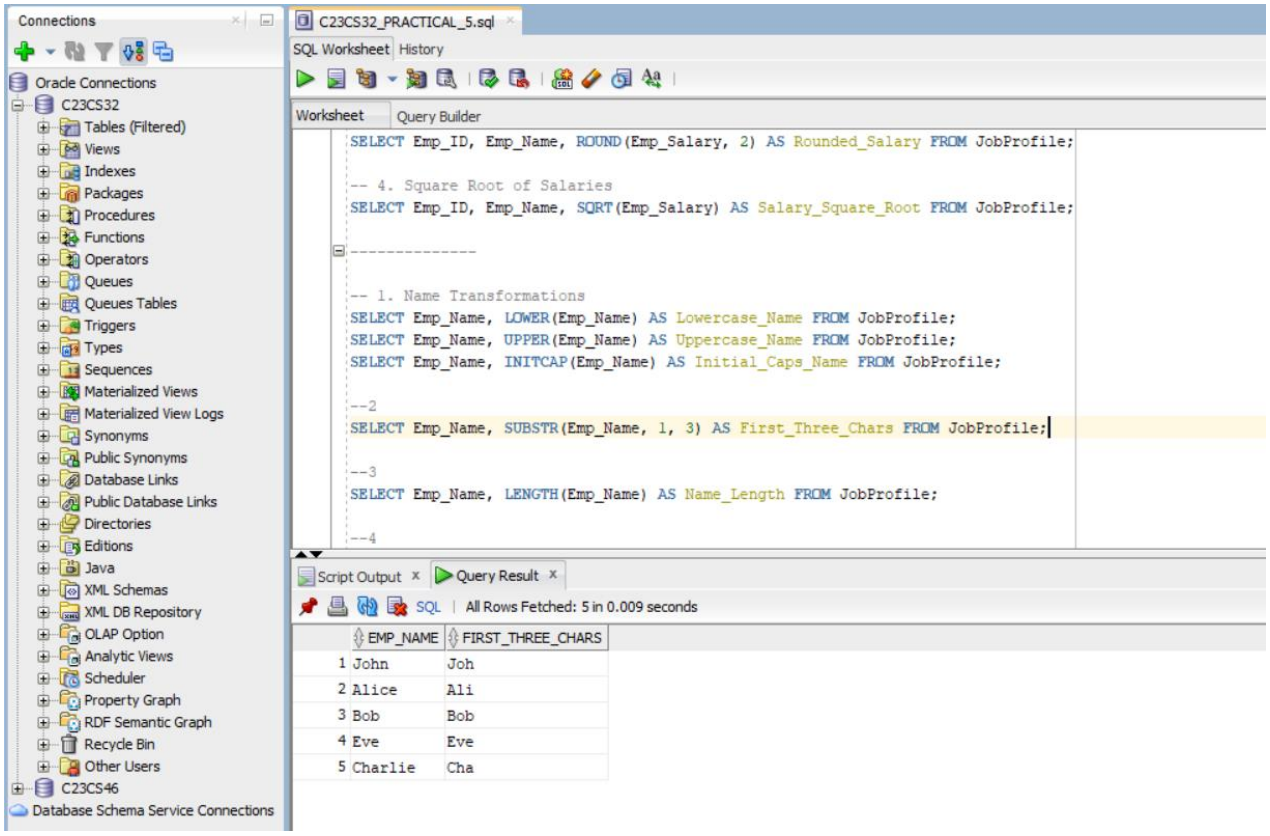
1. Convert all employee first names to lowercase, uppercase, and initial caps.



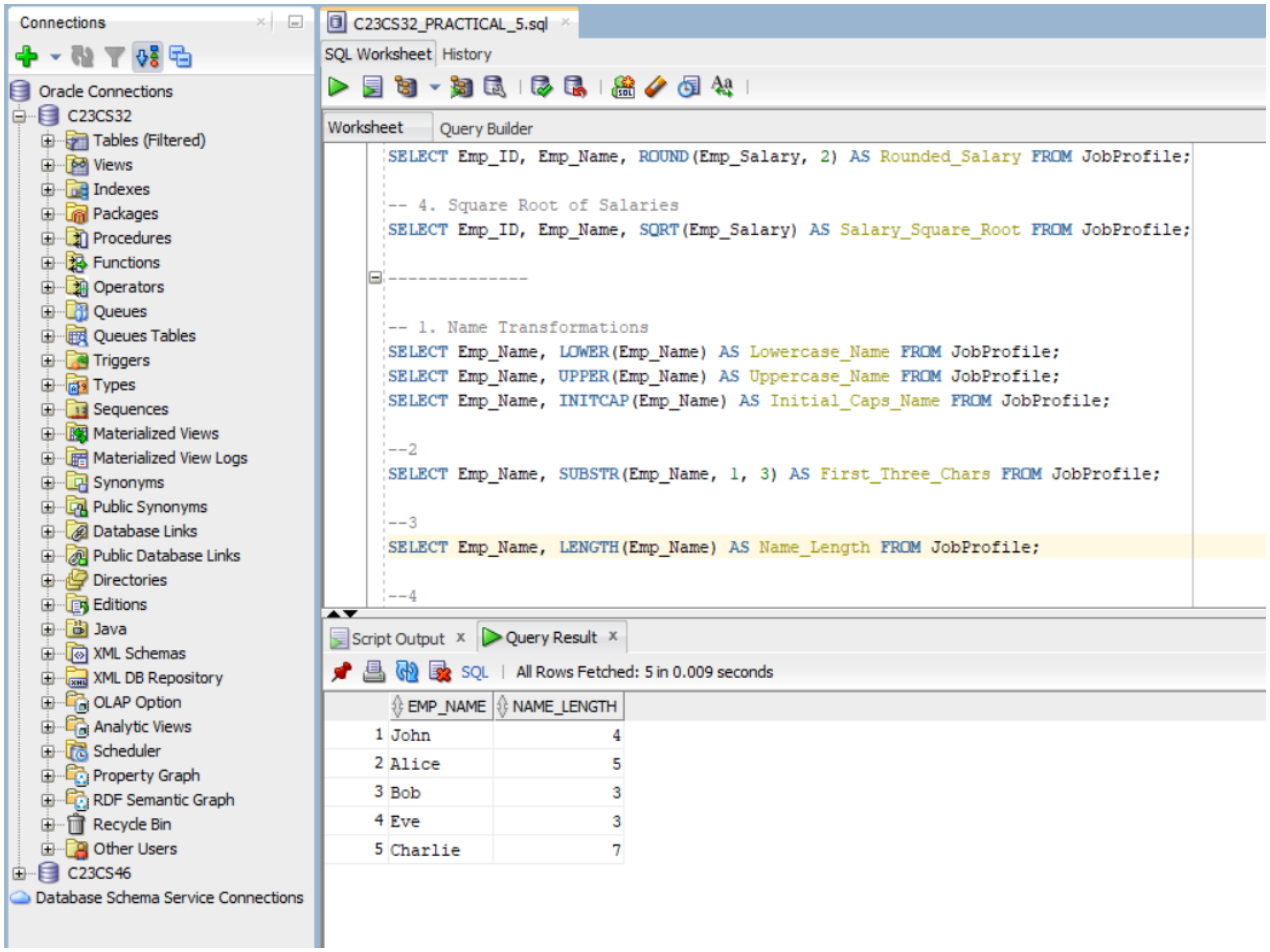

Connections
Oracle Connections
C23CS32
Tables (Filtered)
Views
Indexes
Packages
Procedures
Functions
Operators
Queues
Queues Tables
Triggers
Types
Sequences
Materialized Views
Materialized View Logs
Synonyms
Public Synonyms
Database Links
Public Database Links
Directories
Editions
Java
XML Schemas
XML DB Repository
OLAP Option
Analytic Views
Scheduler
Property Graph
RDF Semantic Graph
Recycle Bin
Other Users
C23CS46
Database Schema Service Connections
C23CS32_PRACTICAL_5.sql
SQL Worksheet
History
Worksheet
Query Builder
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;
--3
SELECT Emp_Name, LENGTH(Emp_Name) AS Name_Length FROM JobProfile;
--4
Script Output
Query Result
SQL
All Rows Fetched: 5 in 0.012 seconds
EMP_NAME
UPPERCASE_NAME
1 John JOHN
2 Alice ALICE
3 Bob BOB
4 Eve EVE
5 Charlie CHARLIE


Connections
Oracle Connections
C23CS32
Tables (Filtered)
Views
Indexes
Packages
Procedures
Functions
Operators
Queues
Queues Tables
Triggers
Types
Sequences
Materialized Views
Materialized View Logs
Synonyms
Public Synonyms
Database Links
Public Database Links
Directories
Editions
Java
XML Schemas
XML DB Repository
OLAP Option
Analytic Views
Scheduler
Property Graph
RDF Semantic Graph
Recycle Bin
Other Users
C23CS46
Database Schema Service Connections
C23CS32_PRACTICAL_5.sql
SQL Worksheet
History
Worksheet
Query Builder
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;
--3
SELECT Emp_Name, LENGTH(Emp_Name) AS Name_Length FROM JobProfile;
--4
Script Output
Query Result
SQL
All Rows Fetched: 5 in 0.011 seconds
EMP_NAME
INITIAL_CAPS_NAME
1 John John
2 Alice Alice
3 Bob Bob
4 Eve Eve
5 Charlie Charlie

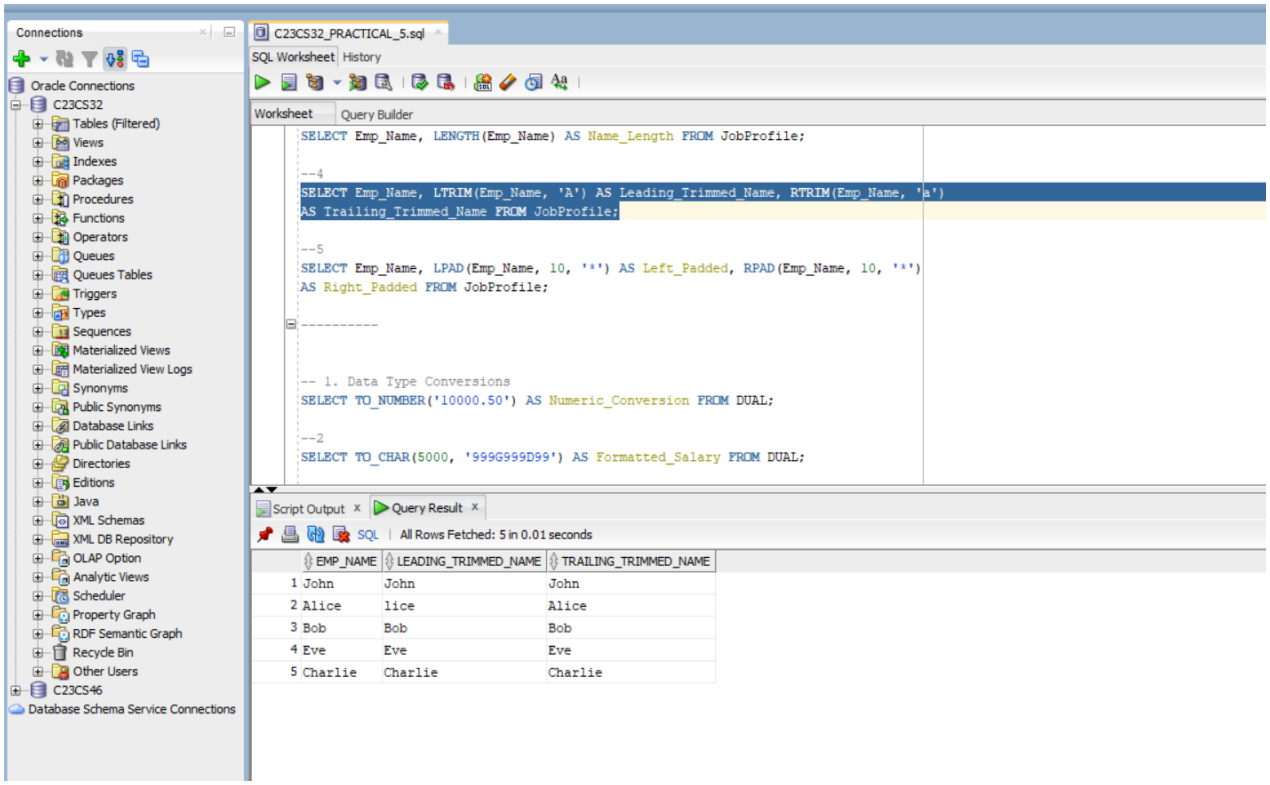
2. Extract the first three characters of employee first names.



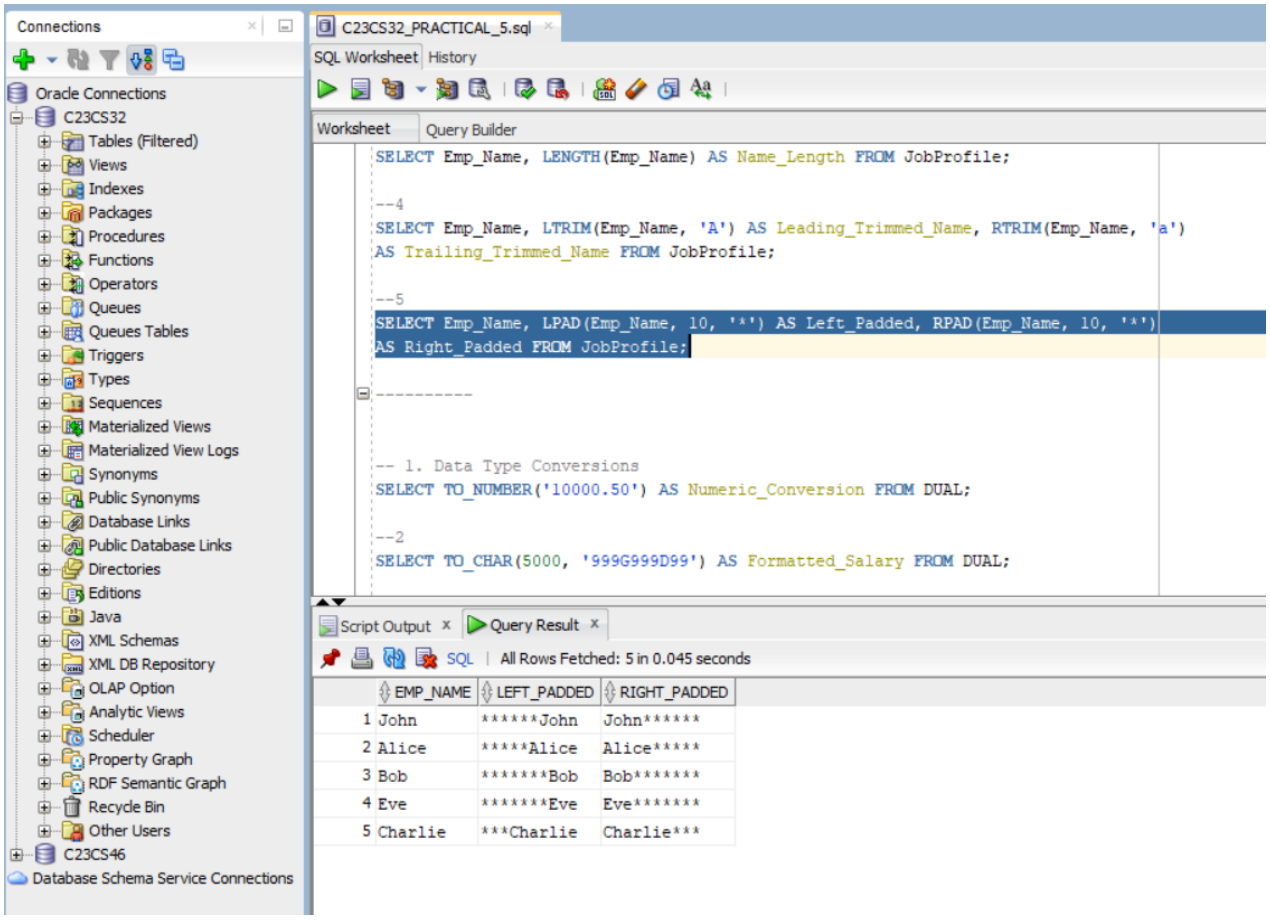
3. Find the length of each employee's first name.



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

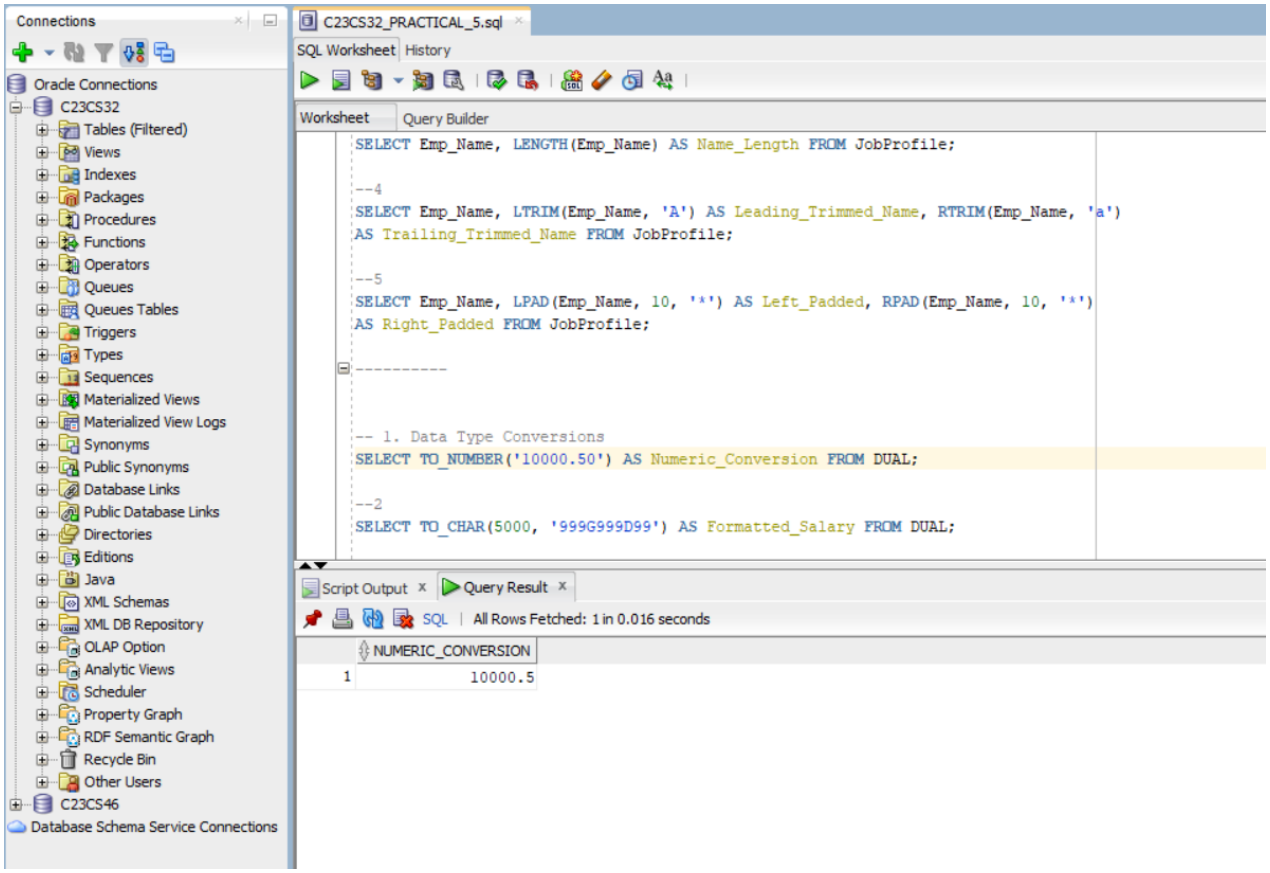


5. Pad employee first names with '*' on the left and right, ensuring a total length of 10.

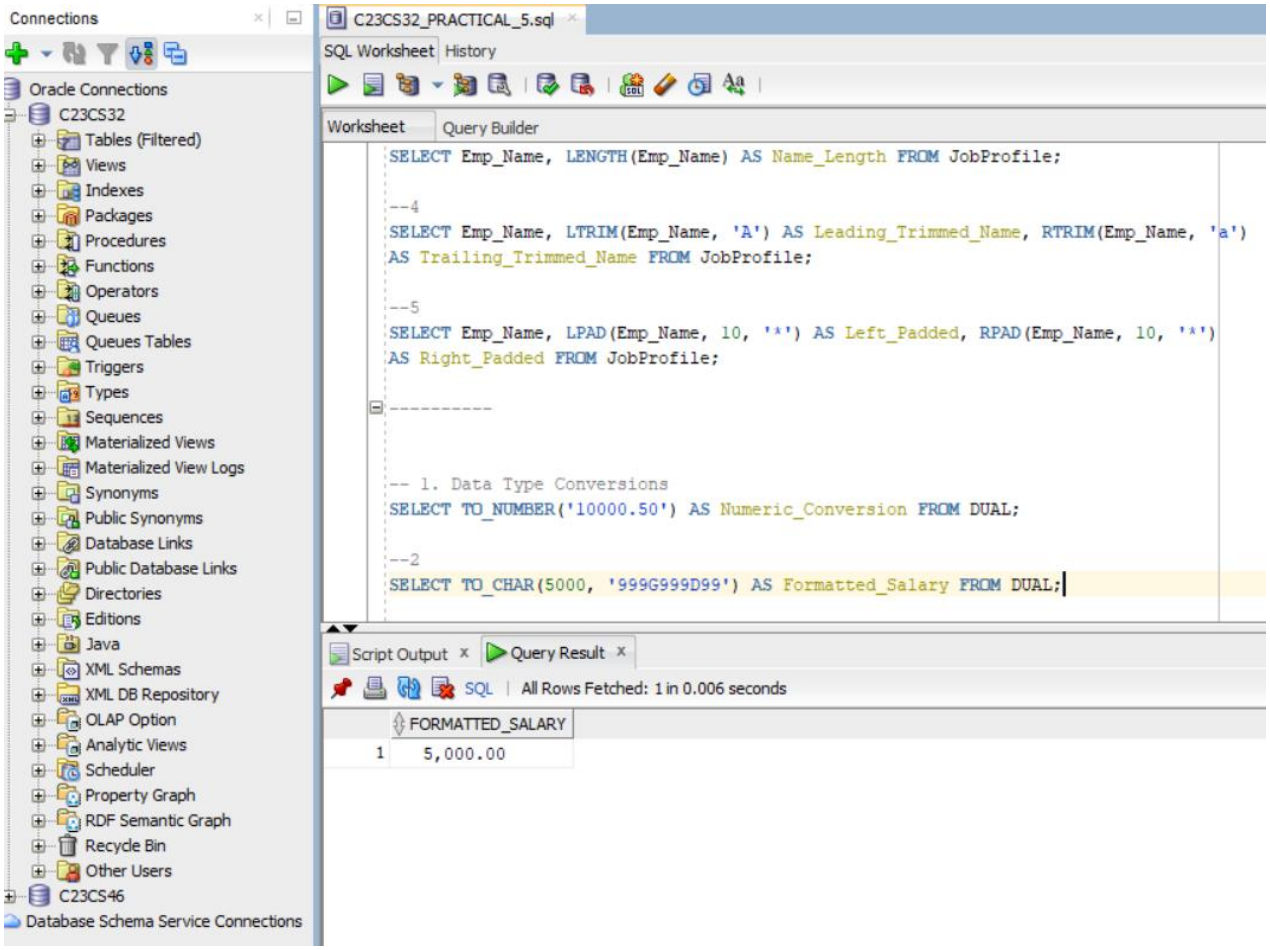


The data migration team requires conversions between different data types:

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

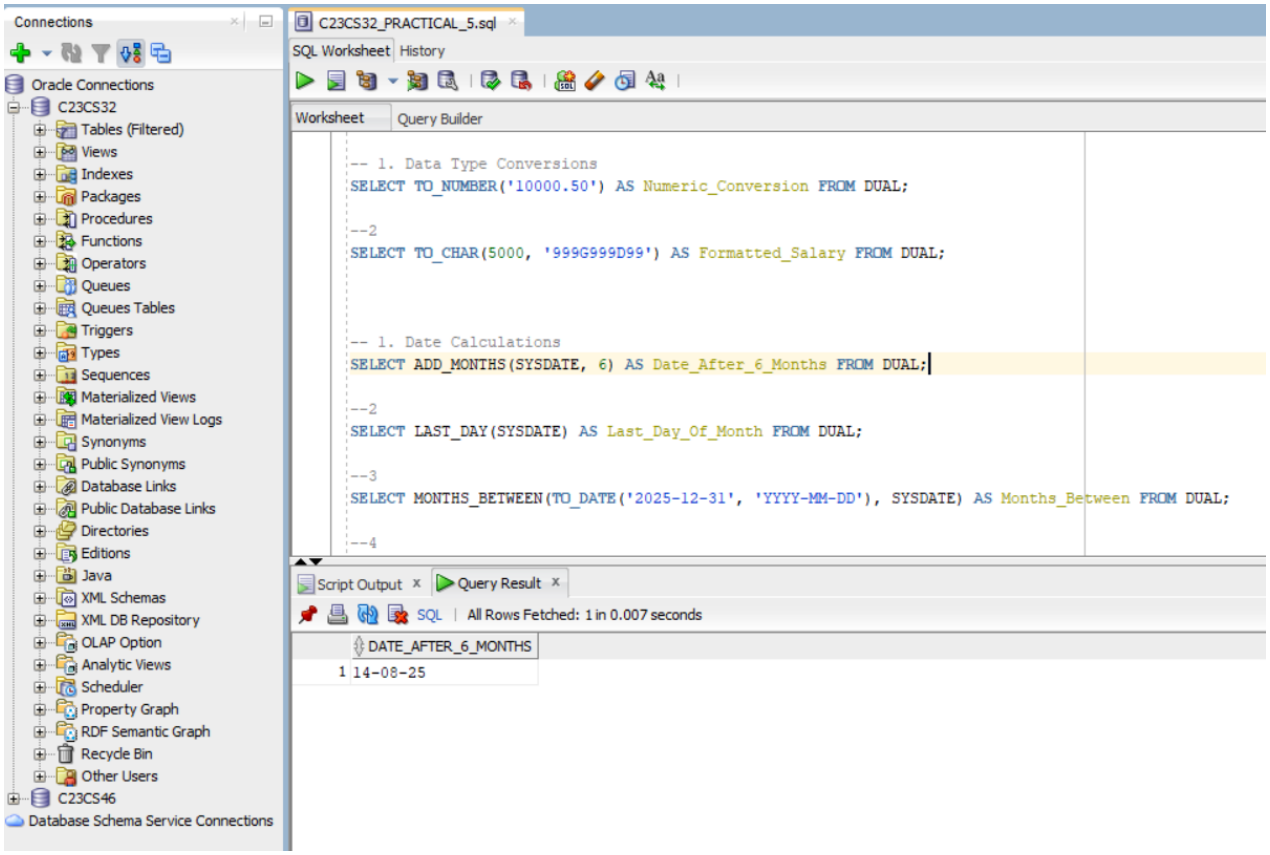


2. Format a numeric salary value into a string with specific formatting.

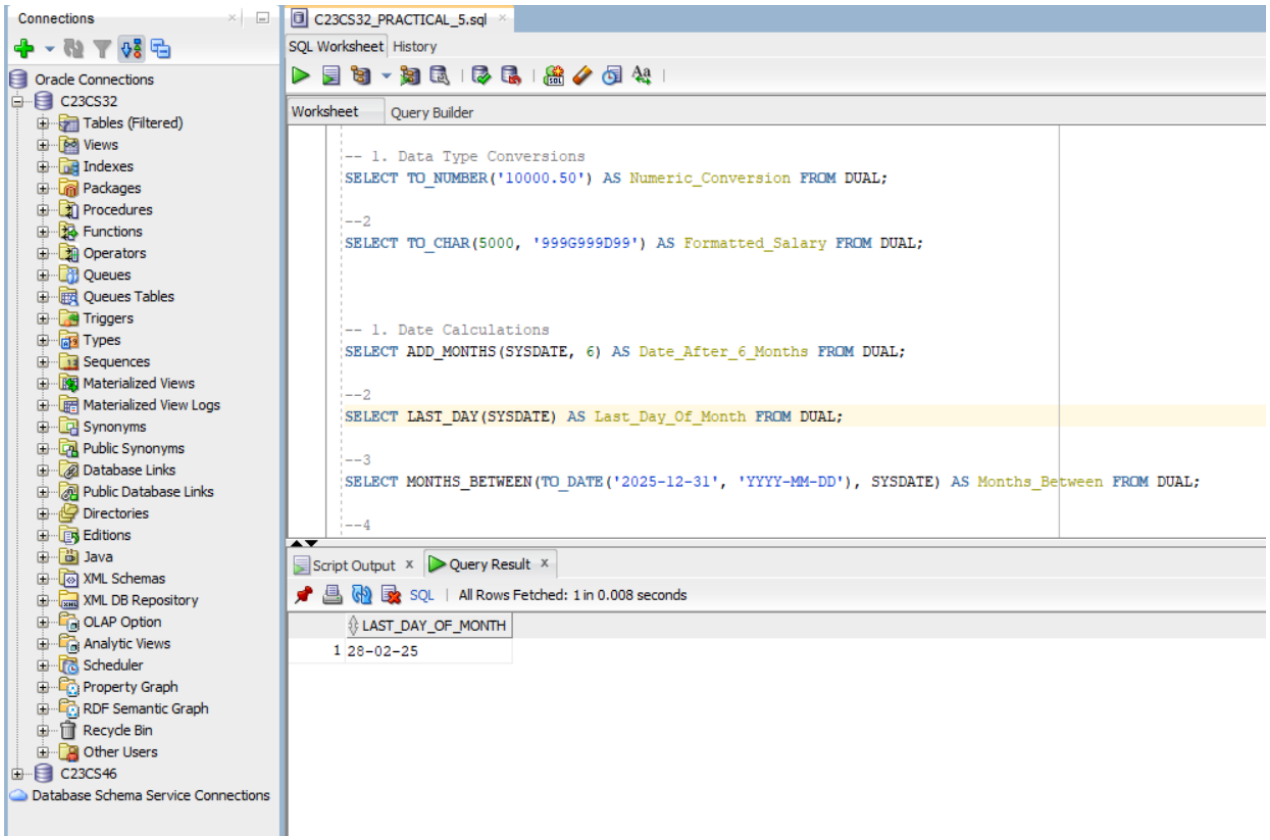


To assist in employee scheduling, perform the following:

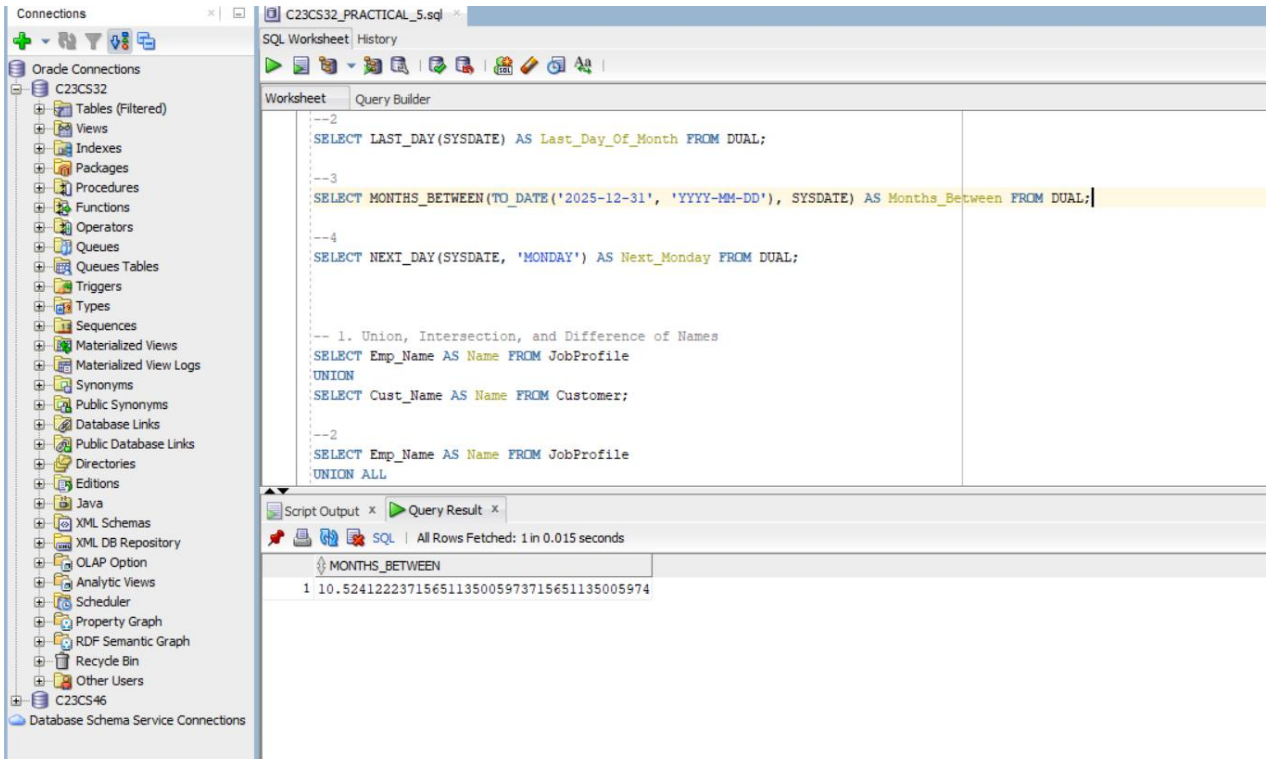
1. Calculate the date after adding 6 months to the current date.



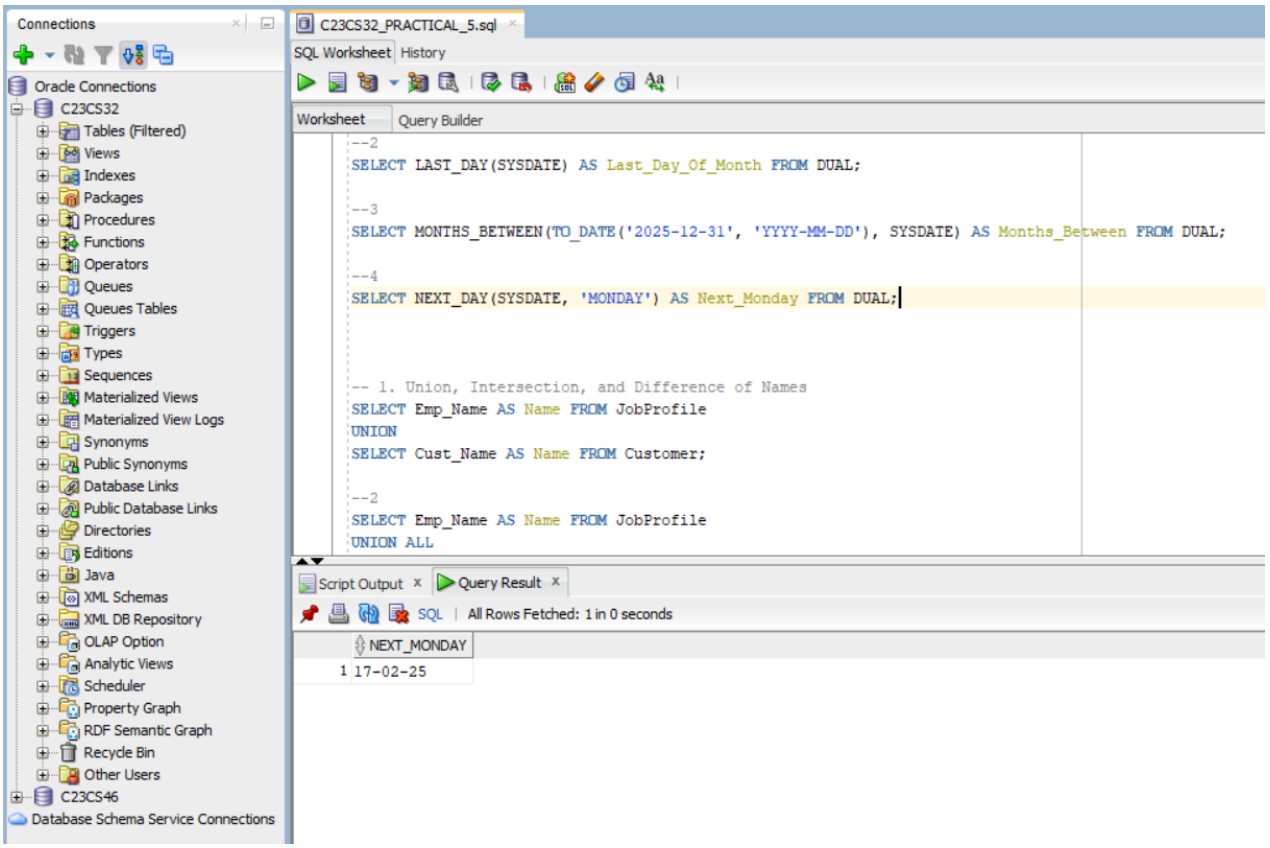
2. Retrieve the last day of the current month.



3. Calculate the number of months between two dates.

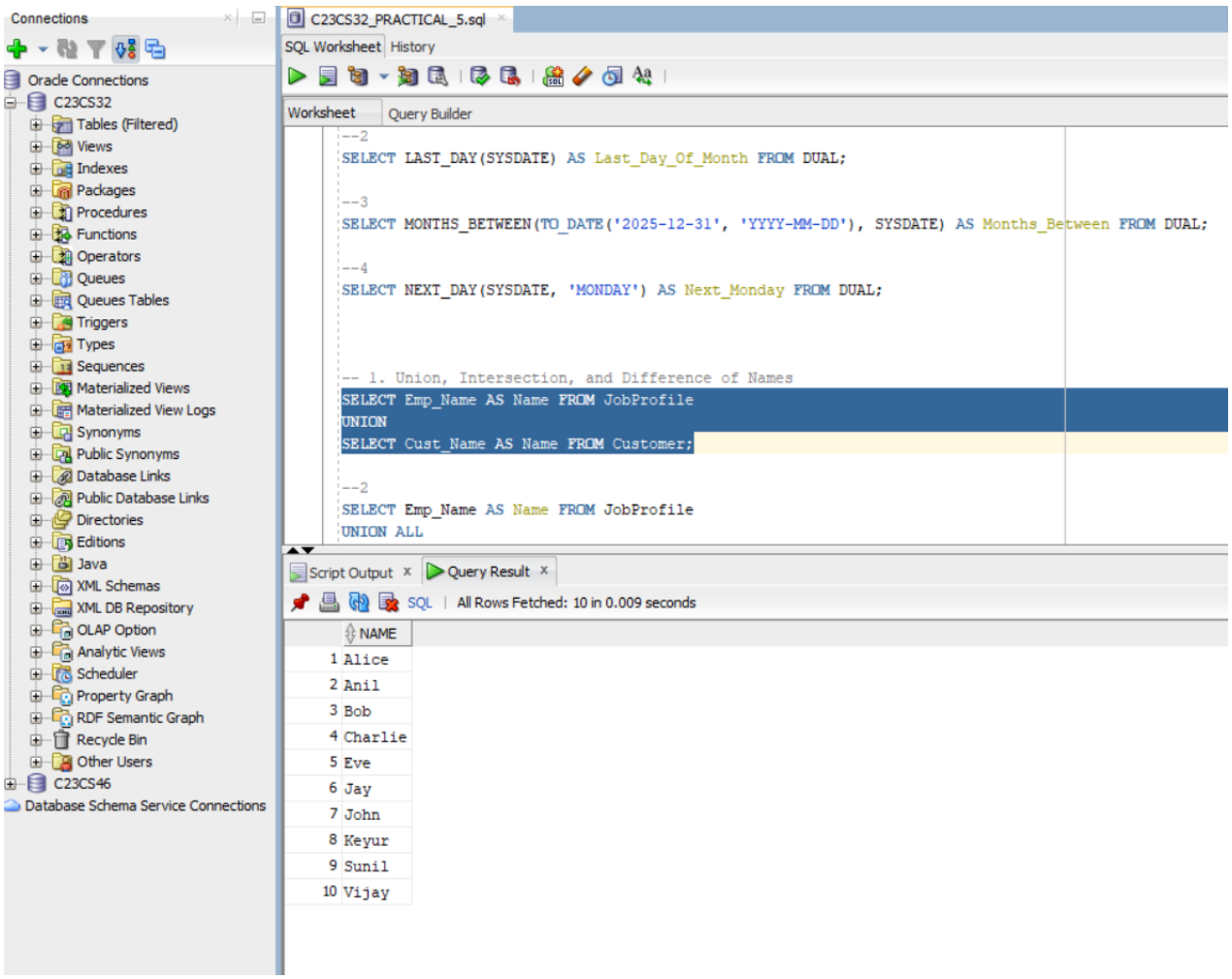


4. Find the next Monday from the current date.

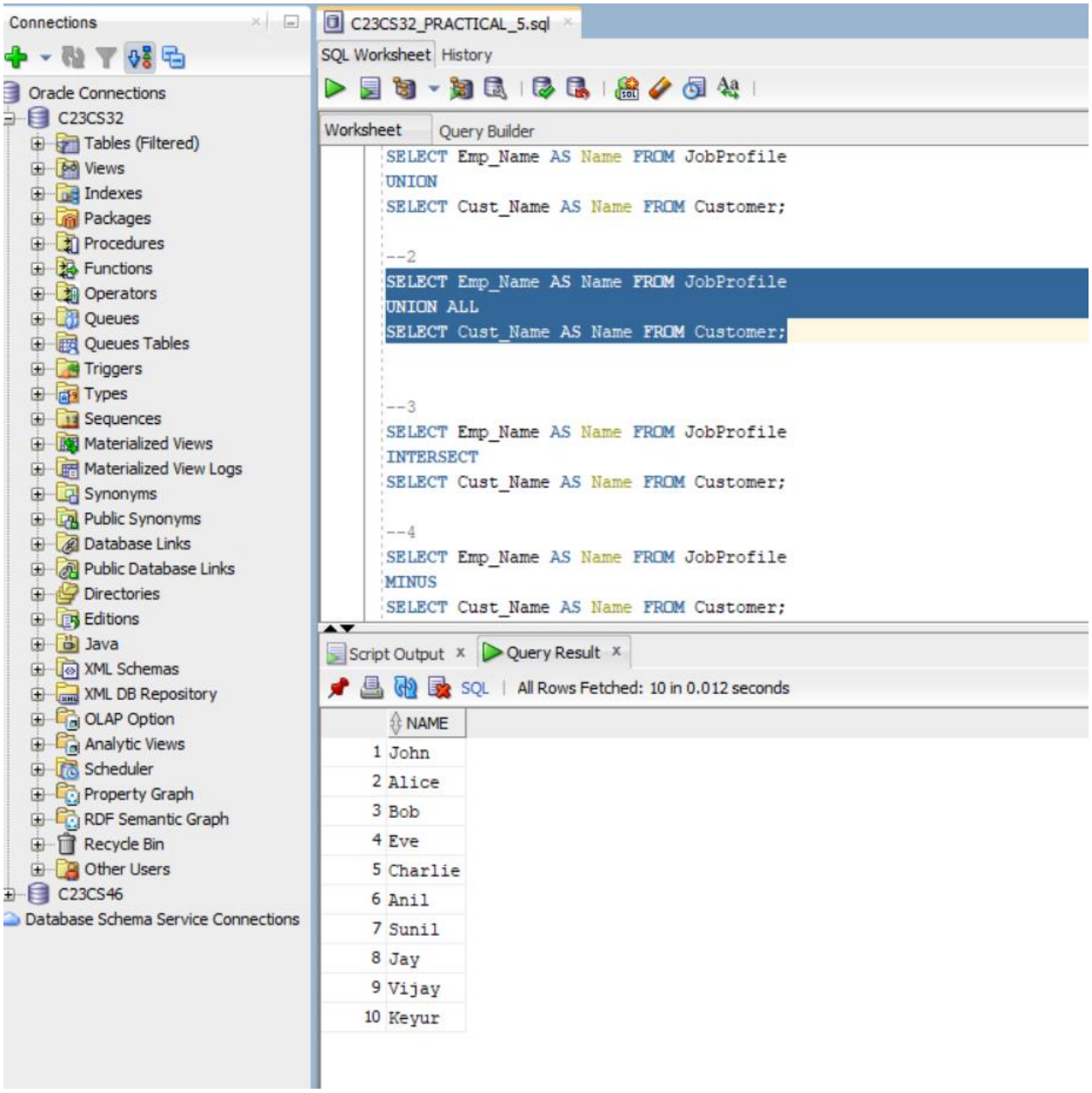


To identify overlaps and differences between employees and customers, write SQL queries to:

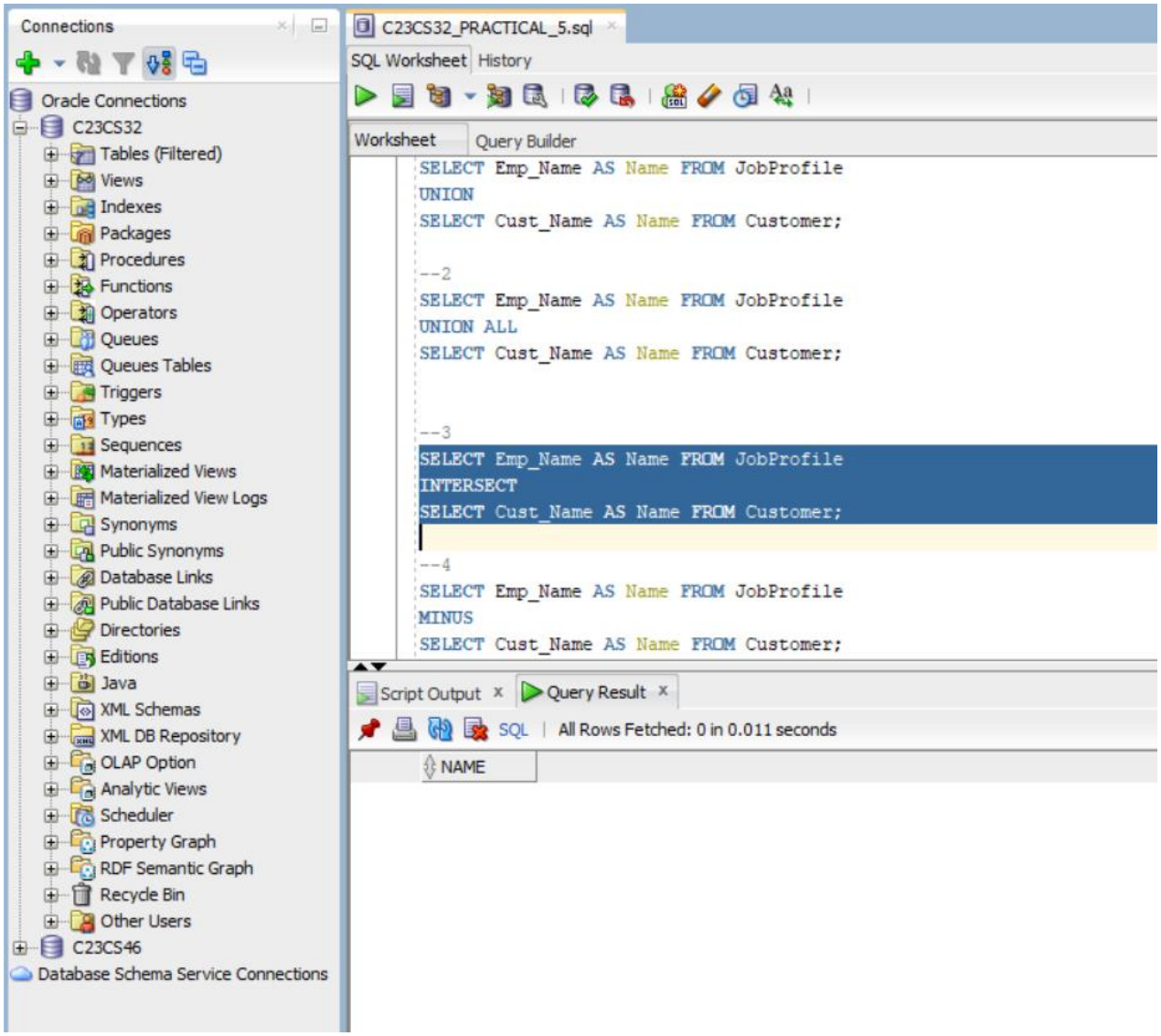
1. Retrieve the union of first names from employees and customers.



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



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



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