



## 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.

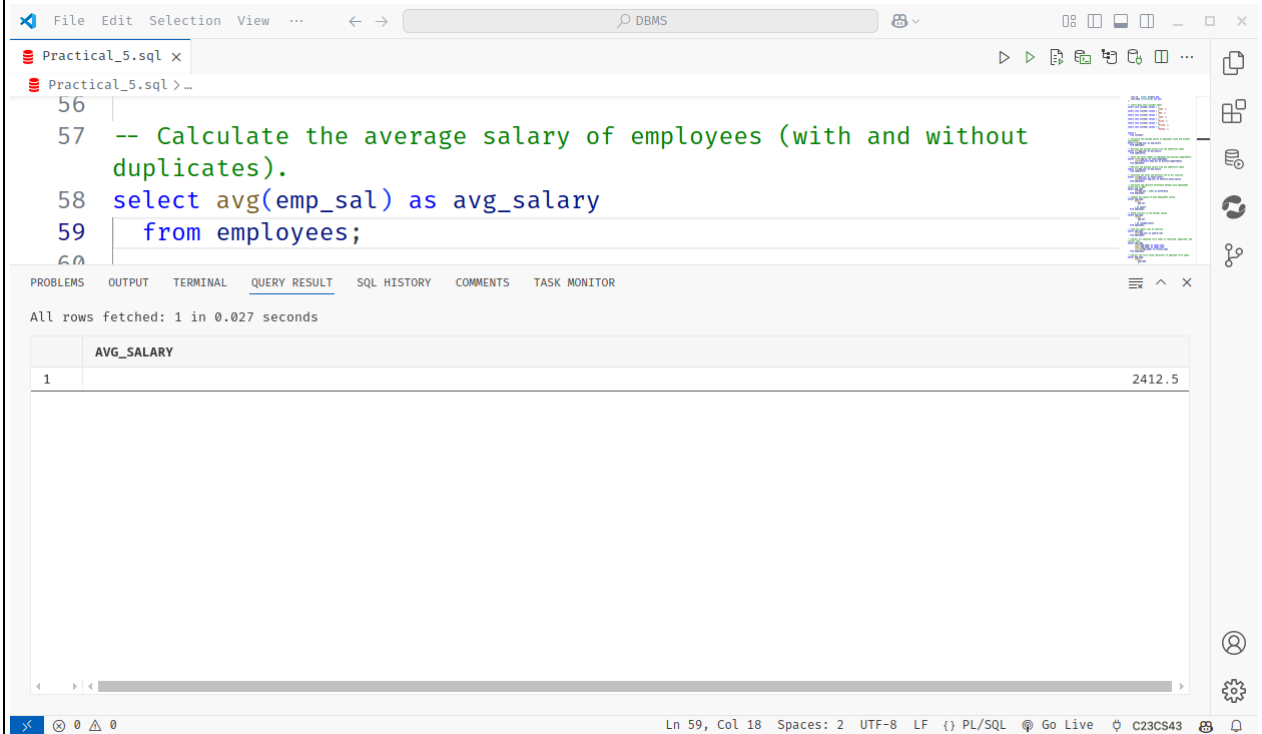
- Emp\_ID (Primary Key)
- Emp\_Name (Not Null)
- Emp\_Salary (Not Null, Check: Greater than zero)
- Job\_ID (Unique)
- Department

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

- Cust\_ID (Primary Key)

- **Cust\_Name (Not Null)**

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



The screenshot shows the SQL Developer interface with a query window titled 'Practical\_5.sql'. The query is as follows:

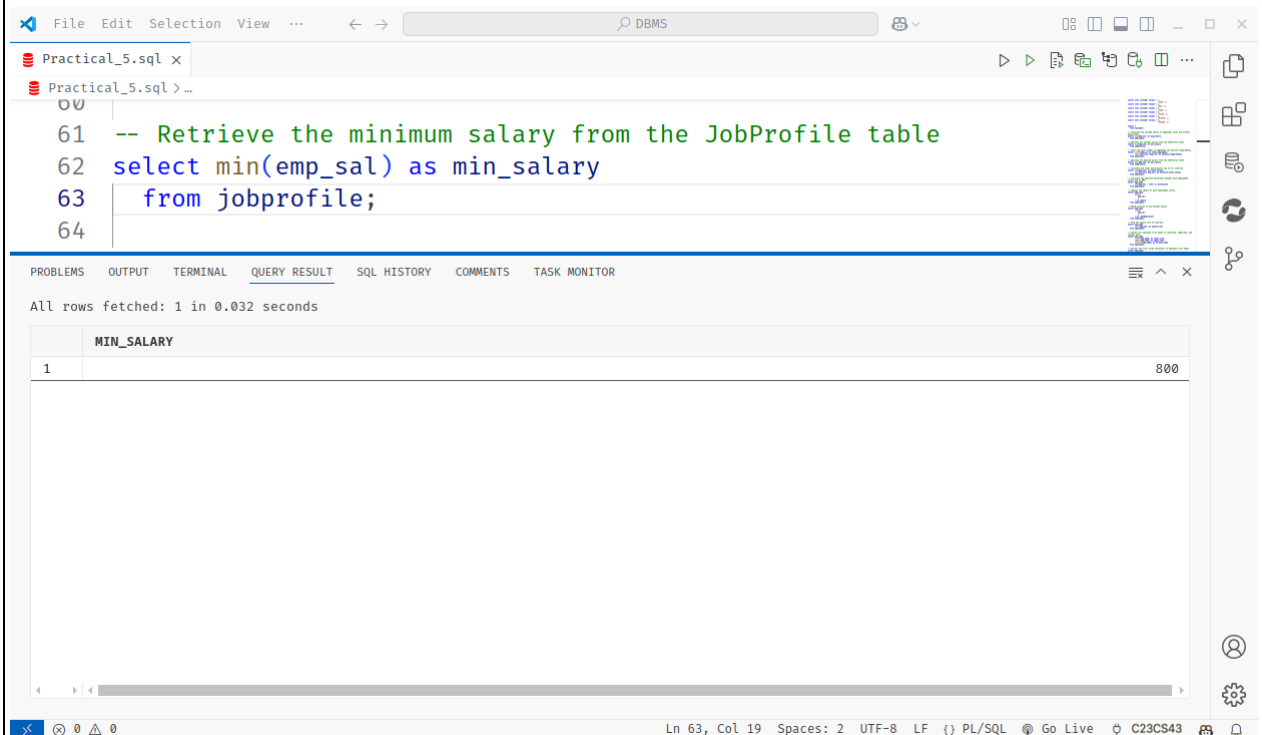
```
56  
57 -- Calculate the average salary of employees (with and without  
58 select avg(emp_sal) as avg_salary  
59 from employees;  
60
```

The query results are displayed in a table with the following structure:

	AVG_SALARY
1	2412.5

The status bar at the bottom indicates 'Ln 59, Col 18 Spaces: 2 UTF-8 LF {} PL/SQL Go Live C23CS43'.

### 2. Retrieve the minimum salary from the JobProfile table.



The screenshot shows the SQL Developer interface with a query window titled 'Practical\_5.sql'. The query is as follows:

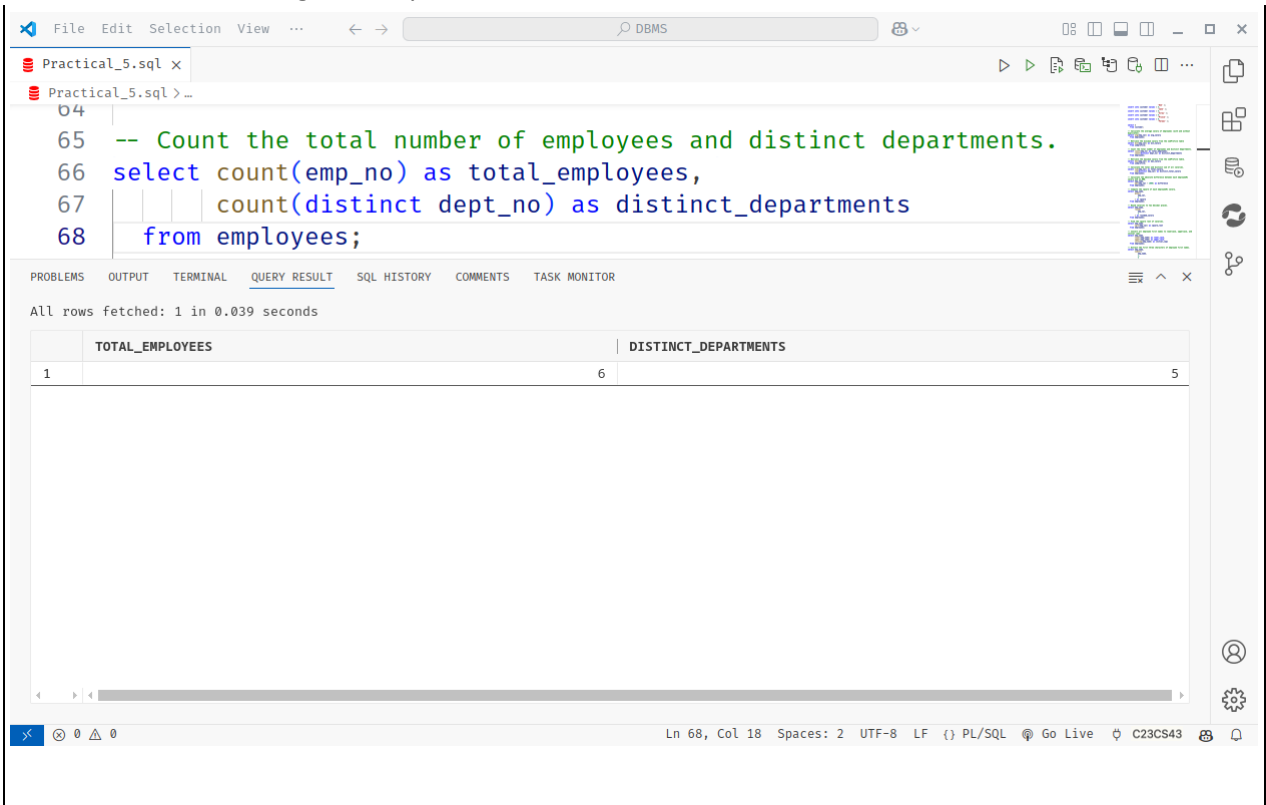
```
61 -- Retrieve the minimum salary from the JobProfile table  
62 select min(emp_sal) as min_salary  
63 from jobprofile;  
64
```

The query results are displayed in a table with the following structure:

	MIN_SALARY
1	800

The status bar at the bottom indicates 'Ln 63, Col 19 Spaces: 2 UTF-8 LF {} PL/SQL Go Live C23CS43'.

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



The screenshot shows the SQL Developer interface with a query window titled 'Practical\_5.sql'. The query is as follows:

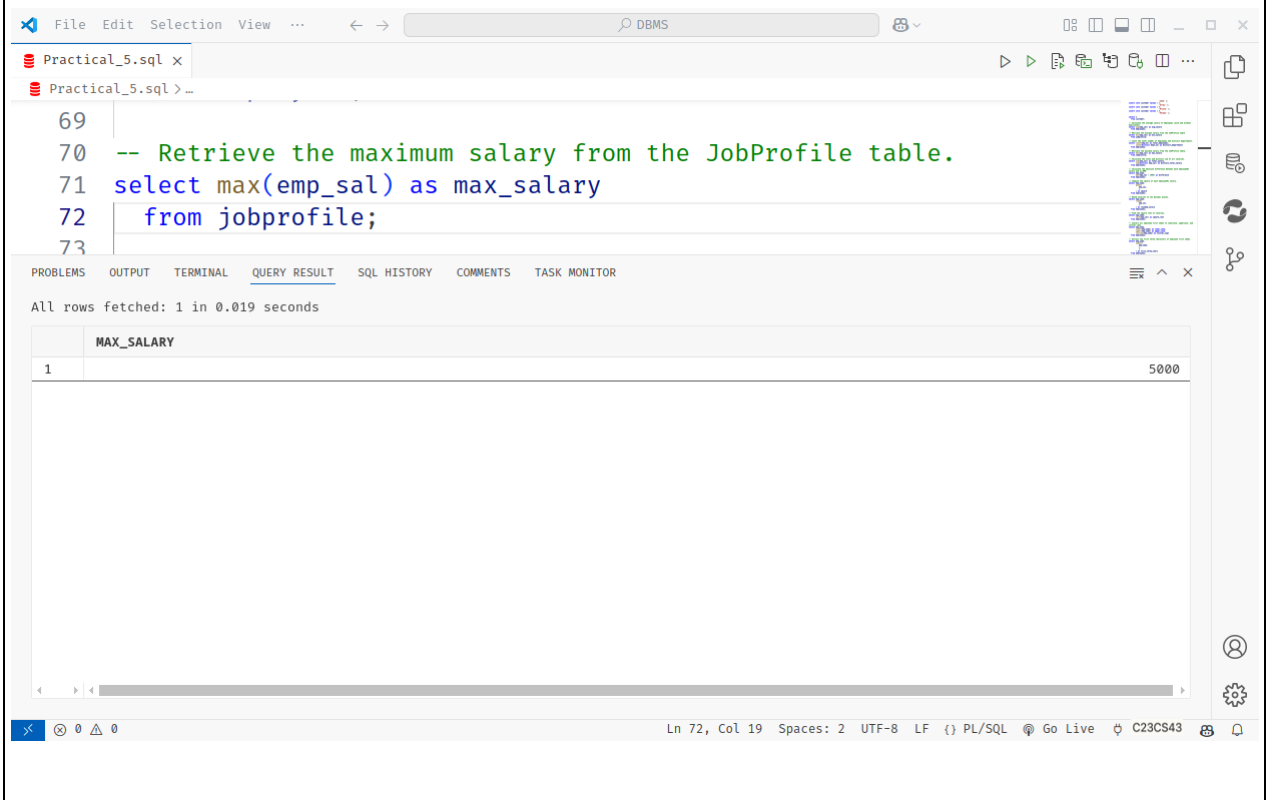
```
65 -- Count the total number of employees and distinct departments.
66 select count(emp_no) as total_employees,
67        count(distinct dept_no) as distinct_departments
68 from employees;
```

The query has been executed, and the results are displayed in the 'QUERY RESULT' tab. The message 'All rows fetched: 1 in 0.039 seconds' is shown. The results table has two columns: 'TOTAL\_EMPLOYEES' and 'DISTINCT\_DEPARTMENTS'.

	TOTAL_EMPLOYEES	DISTINCT_DEPARTMENTS
1	6	5

The status bar at the bottom indicates 'Ln 68, Col 18 Spaces: 2 UTF-8 LF {} PL/SQL Go Live C23CS43'.

#### 4. Retrieve the maximum salary from the JobProfile table.



The screenshot shows the SQL Developer interface with a query window titled 'Practical\_5.sql'. The query is as follows:

```
69
70 -- Retrieve the maximum salary from the JobProfile table.
71 select max(emp_sal) as max_salary
72 from jobprofile;
73
```

The query has been executed, and the results are displayed in the 'QUERY RESULT' tab. The message 'All rows fetched: 1 in 0.019 seconds' is shown. The results table has one column: 'MAX\_SALARY'.

	MAX_SALARY
1	5000

The status bar at the bottom indicates 'Ln 72, Col 19 Spaces: 2 UTF-8 LF {} PL/SQL Go Live C23CS43'.

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

The screenshot shows the SQL Developer interface with a query window titled 'Practical\_5.sql'. The query is as follows:

```
74 -- Calculate the total and distinct sum of all salaries.
75 select sum(emp_sal) as total_salary,
76        sum(distinct emp_sal) as distinct_total_salary
77 from employees;
```

The 'QUERY RESULT' tab is active, displaying the following table:

	TOTAL_SALARY	DISTINCT_TOTAL_SALARY
1	14475	14475

The status bar at the bottom indicates 'Ln 77, Col 18 Spaces: 2 UTF-8 LF {} PL/SQL Go Live C23CS43'.

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

The screenshot shows the SQL Developer interface with a query window titled 'Practical\_5.sql'. The query is as follows:

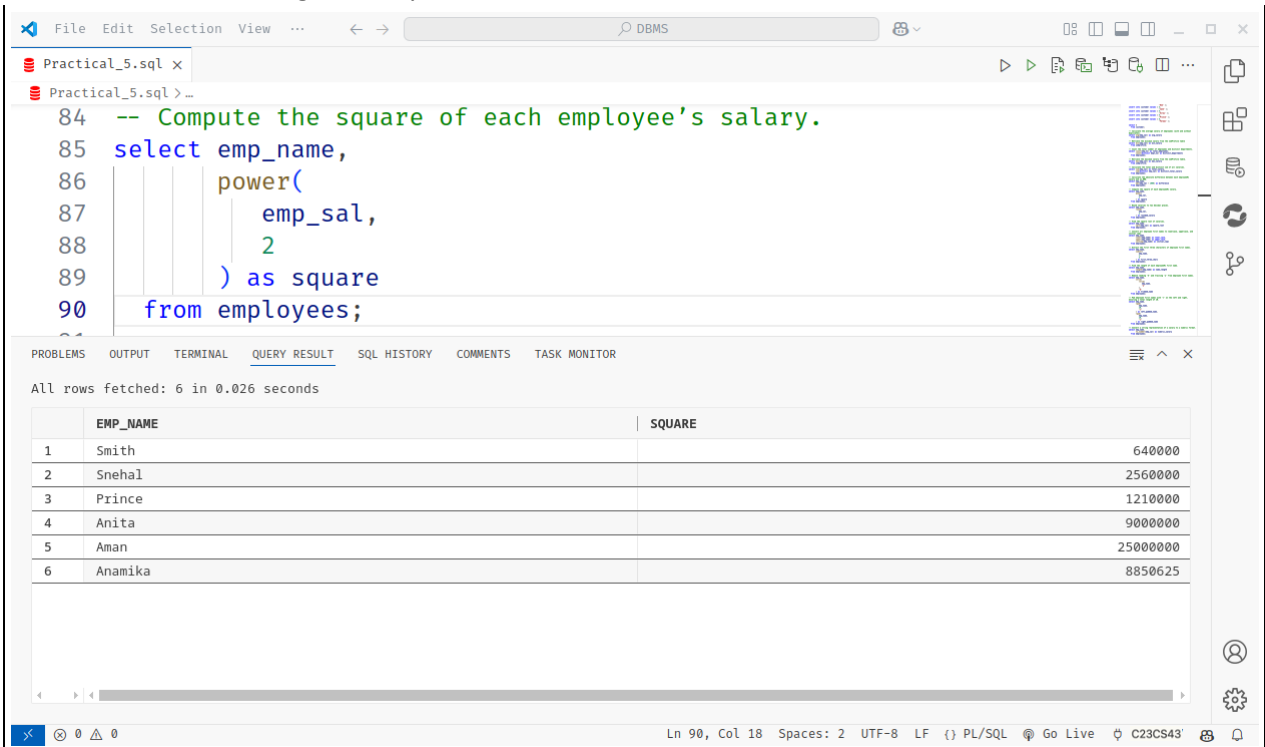
```
79 -- Calculate the absolute difference between each employee's
80 salary and ₹1,000.
81 select emp_name,
82        abs(emp_sal - 1000) as difference
82 from employees;
```

The 'QUERY RESULT' tab is active, displaying the following table:

	EMP_NAME	DIFFERENCE
1	Smith	200
2	Snehal	600
3	Prince	100
4	Anita	2000
5	Aman	4000
6	Anamika	1975

The status bar at the bottom indicates 'Ln 82, Col 18 Spaces: 2 UTF-8 LF {} PL/SQL Go Live C23CS43'.

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



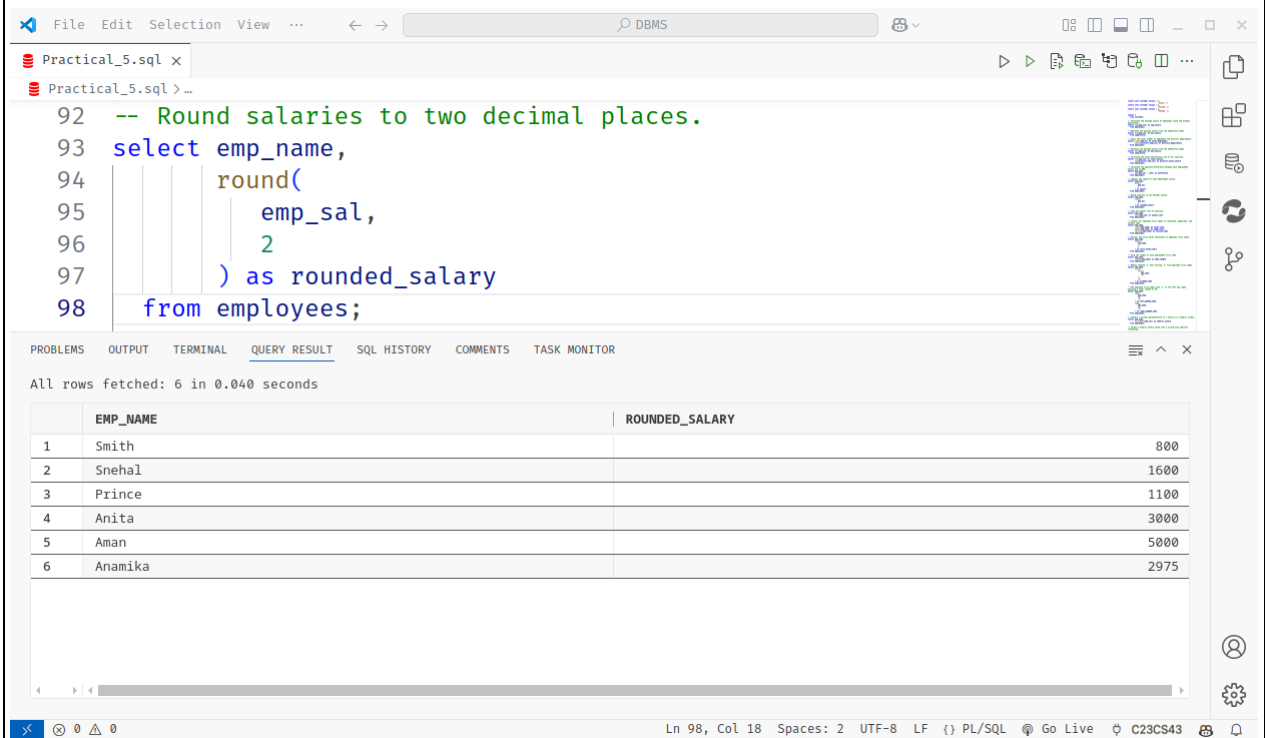
The screenshot shows the SQL Developer interface with a query window titled 'Practical\_5.sql'. The query is as follows:

```
84 -- Compute the square of each employee's salary.
85 select emp_name,
86        power(
87            emp_sal,
88            2
89        ) as square
90 from employees;
```

The 'QUERY RESULT' tab is selected, showing the results of the query. The status bar indicates 'All rows fetched: 6 in 0.026 seconds'.

	EMP_NAME	SQUARE
1	Smith	640000
2	Snehal	2560000
3	Prince	1210000
4	Anita	9000000
5	Aman	25000000
6	Anamika	8850625

#### 8. Round salaries to two decimal places.



The screenshot shows the SQL Developer interface with a query window titled 'Practical\_5.sql'. The query is as follows:

```
92 -- Round salaries to two decimal places.
93 select emp_name,
94        round(
95            emp_sal,
96            2
97        ) as rounded_salary
98 from employees;
```

The 'QUERY RESULT' tab is selected, showing the results of the query. The status bar indicates 'All rows fetched: 6 in 0.040 seconds'.

	EMP_NAME	ROUNDED_SALARY
1	Smith	800
2	Snehal	1600
3	Prince	1100
4	Anita	3000
5	Aman	5000
6	Anamika	2975

#### 9. Find the square root of salaries.

The screenshot shows a DBMS IDE interface. The top menu bar includes File, Edit, Selection, View, and a search bar labeled DBMS. The editor window displays a SQL file named 'Practical\_5.sql' with the following code:

```
100 -- Find the square root of salaries.
101 select emp_name,
102        sqrt(emp_sal) as square_root
103 from employees;
104
```

Below the editor, the 'QUERY RESULT' tab is active, showing the execution output. It states 'All rows fetched: 6 in 0.034 seconds'. The results are displayed in a table with two columns: EMP\_NAME and SQUARE\_ROOT.

	EMP_NAME	SQUARE_ROOT
1	Smith	28.28427124746190097603377448419396157139
2	Snehal	40
3	Prince	33.16624790355399849114932736670686683927
4	Anita	54.77225575051661134569697828008021339527
5	Aman	70.71067811865475244008443621048490392848
6	Anamika	54.54356057317857205751077243686450936402

The bottom status bar shows 'Ln 103, Col 18 Spaces: 2 UTF-8 LF {} PL/SQL Go Live C23CS43'.