



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CHANDIGARH  
UNIVERSITY

Discover. Learn. Empower.

## Experiment 2

**Student Name:** Ravi Ranjan Kumar

**UID:** 25MCA20098

**Branch:** MCA general

**Section/Group:** 25MCA\_KAR-1

**Semester:** II

**Date of Performance:** 13-01-2026

**Subject Name:** Technical Training

**Subject Code:** 25CAP-652

### **1. Aim:**

To analyze student performance using SQL by identifying high-performing students (percentage > 95) and evaluating their distribution across different cities using CASE statements and aggregate functions.

### **2. Objective:**

1. To create and analyze a Students database table containing student ID, name, city, and percentage.
2. To determine the count of students whose percentage is greater than 95:
  - a. Using a CASE statement
  - b. Without using a CASE statement
3. To calculate the average percentage of students (percentage > 95) in each city:
  - a. Using a CASE statement
  - b. Without using a CASE statement
4. To sort the cities in descending order of average percentage to identify cities with the highest-performing students.
5. To understand the use of CASE statements, aggregate functions (COUNT, AVG), GROUP BY, and ORDER BY clauses in SQL for data analysis.

### **3. Implementation/Code:**

```
CREATE TABLE Students (student_id INT, name VARCHAR(50), city VARCHAR(50), percentage DECIMAL(5,2));
```

```
INSERT INTO Students VALUES(1, 'Amit', 'Delhi', 96.5);
```

```
INSERT INTO Students VALUES(2, 'Riya', 'Mumbai', 94.2);
```

```
INSERT INTO Students VALUES(3, 'Rahul', 'Delhi', 97.8);
```

```
INSERT INTO Students VALUES(4, 'Sneha', 'Mumbai', 98.1);
```

```
INSERT INTO Students VALUES(5, 'Ankit', 'Chandigarh', 95.6);
```

```
INSERT INTO Students VALUES(6, 'Pooja', 'Delhi', 93.4);
```

```
INSERT INTO Students VALUES(7, 'Karan', 'Chandigarh', 96.2);
```

```
--count
```

```
--without case statement
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CHANDIGARH  
UNIVERSITY

Discover. Learn. Empower.

SELECT city, COUNT(\*) AS students\_count FROM Students WHERE percentage>95 GROUP BY city;

--with case statement

SELECT city, SUM(CASE WHEN percentage>95 THEN 1 ELSE 0 END) AS students\_count FROM Students GROUP BY city;

--average

--without case statement

SELECT city, AVG(percentage) AS students\_avg FROM Students WHERE percentage>95 GROUP BY city ORDER BY students\_avg DESC;

--with case statement

SELECT city, AVG(CASE WHEN percentage>95 THEN percentage ELSE NULL END) AS students\_avg FROM Students GROUP BY city ORDER BY students\_avg DESC;

## 4. Output:

```
17 ---Without Case Statement
18 SELECT CITY , COUNT(*) AS STUDET_COUNT FROM Students
19 WHERE percentage> 95
20 GROUP BY city;
```

Data Output Messages Notifications

Showing rows: 1 to 3 Page No: 1 of 1

	city character varying (50)	studet_count bigint
1	Delhi	2
2	Mumbai	1
3	Chandigarh	2

```
33 SELECT CITY, AVG(CASE WHEN PERCENTAGE>95 THEN PERCENTAGE ELSE
34 NULL END) AS STUDENT_AVG FROM Students
35 GROUP BY city
36 ORDER BY STUDENT_AVG DESC;
```

Data Output Messages Notifications

Showing rows: 1 to 3 Page No: 1 of 1

	city character varying (50)	student_avg numeric
1	Mumbai	98.100000000000000000
2	Delhi	97.150000000000000000
3	Chandigarh	95.900000000000000000



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CHANDIGARH  
UNIVERSITY

Discover. Learn. Empower.

```
25  SELECT CITY, SUM(CASE WHEN percentage > 95 THEN 1
26  ELSE 0 END) AS STUDENT_COUNTS FROM Students
27  GROUP BY city
28
```

Data Output Messages Notifications

The screenshot shows a SQL query execution interface. At the top, there is a code editor with four lines of SQL. Below the code editor is a toolbar with various icons. Underneath the toolbar, it says "Showing rows: 1 to 3" and "Page No: 1 of 1". The main area displays a table with three rows of data. The table has two columns: "city" and "student\_counts". The data is as follows:

	city	student_counts
1	Mumbai	1
2	Delhi	2
3	Chandigarh	2

## 5. Learning Outcomes:

1. Learn how to use the CASE statement in SQL to count students based on specific conditions.
2. Understand how to calculate the average percentage for selected students using SQL.
3. Know how to group data by city and filter results to focus on high-performing students.
4. Be able to sort results to easily identify which cities have the best-performing students.