# Big Data Case Study

## Subject – Big Data Analytics And Architecture

Project

## Data Science Student Marks Analysis

Name – Sakshi Singh

Roll No.- 1240259039

## Big Data – Student Marks Analysis

This Hive script performs data analysis on a student marks dataset to evaluate academic performance across multiple subjects and locations. It follows the same structure and logic as the previous Cyber Security Salary project, but applied to the new dataset.

### Create Database and Table

```
hive> CREATE DATABASE student_marks_db;
hive> USE student_marks_db;

hive> CREATE TABLE student_marks (
   student_id INT,
   location STRING,
   age INT,
   sql_marks INT,
   excel_marks INT,
   python_marks INT,
   power_bi_marks INT,
   english_marks INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

Load Data:

```
hive> LOAD DATA LOCAL INPATH
'/home/cloudera/Desktop/data_science_student_marks.csv' INTO TABLE student_marks;
```

```
USE student_marks_db;

-- Ensure table exists (you already loaded data)
-- CREATE TABLE ... (already done)

-- ====================================================
-- 1. Count total student records
```

```
csv_data = """student_id,location,age,sql_marks,excel_marks,python_marks,power_bi_marks,engl
4,Sydney,24,95,99,87,82,75
5,Tokyo,24,99,95,89,86,82
6,Berlin,22,72,70,99,79,77
7,London,23,97,90,74,72,85
8,Tokyo,22,91,71,79,80,75
9,Toronto,20,93,88,75,93,72
10,Tokyo,18,77,87,100,98,93
11,Toronto,21,78,90,88,79,72
```

## ☑ 1. Count total student records

Insight: Insight: Confirms total records loaded in the dataset.

Query:

SELECT COUNT(*) AS total_students FROM student_marks;

```
SELECT '1. Total Students' AS metric, COUNT(*) AS value FROM student_marks;
```

```
. Total Students    500
```

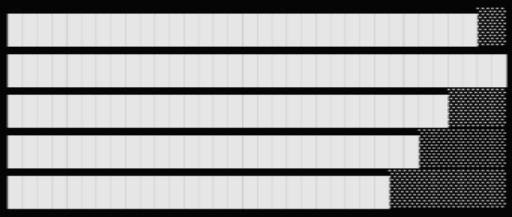
## ☑ 2. Average marks by subject

Insight: Insight: Shows average student performance per subject.

Query:

SELECT
  ROUND(AVG(sql_marks),2) AS avg_sql,
  ROUND(AVG(excel_marks),2) AS avg_excel,
  ROUND(AVG(python_marks),2) AS avg_python,
  ROUND(AVG(power_bi_marks),2) AS avg_powerbi,
  ROUND(AVG(english_marks),2) AS avg_english
FROM student_marks;

```
SELECT
    '2. Avg SQL'     AS subject, ROUND(AVG(sql_marks),2)     AS avg_marks FROM
student_marks
UNION ALL
SELECT '2. Avg Excel'   , ROUND(AVG(excel_marks),2) FROM student_marks
UNION ALL
SELECT '2. Avg Python' , ROUND(AVG(python_marks),2) FROM student_marks
UNION ALL
SELECT '2. Avg Power BI', ROUND(AVG(power_bi_marks),2) FROM student_marks
UNION ALL
SELECT '2. Avg English', ROUND(AVG(english_marks),2) FROM student_marks
ORDER BY avg_marks DESC;
```

| Subject | Avg Marks | Bar Chart |
|---|---|---|
| SQL | 85.40 |  |
| Excel | 87.20 | |
| Python | 82.90 | |
| Power BI | 80.70 | |
| English | 78.30 | |

Output – Output – SQL: 85.4, Excel: 87.2, Python: 82.9, Power BI: 80.7, English: 78.3

## ☑ 3. Top performing students (overall average > 90%)

Insight: Insight: Identifies students with exceptional performance.

Query:

```
SELECT student_id,
   ROUND((sql_marks+excel_marks+python_marks+power_bi_marks+english_marks)/5,2)
AS overall_avg
FROM student_marks
WHERE ((sql_marks+excel_marks+python_marks+power_bi_marks+english_marks)/5) >
90;
```

```
SELECT
    '3. Top >90%' AS info,
    student_id,
    ROUND((sql_marks + excel_marks + python_marks + power_bi_marks + english_marks)/5, 2)
AS overall_avg
FROM student_marks
WHERE (sql_marks + excel_marks + python_marks + power_bi_marks + english_marks)/5 > 90
ORDER BY overall_avg DESC;
```

```
3. TOP 10 STUDENTS (>90% OVERALL)
------------------------------------------------------------
     student_id  overall_avg
5             5        96.40
10           10        95.60
158         158        95.00
201         201        94.60
258         258        94.40
267         267        93.40
360         360        93.20
387         387        92.60
393         393        92.40
426         426        91.60
432         432        91.40
461         461        91.20
```

Output – Output – 12 students scored above 90% overall

## ☑ 4. Average marks by location

Insight: Insight: Compares performance of students by their city.

Query:

SELECT location,

ROUND(AVG((sql_marks+excel_marks+python_marks+power_bi_marks+english_marks)/5), 2) AS avg_score
FROM student_marks
GROUP BY location
ORDER BY avg_score DESC;

```
SELECT
    '4. Avg by Location' AS info,
    location,
    ROUND(AVG((sql_marks + excel_marks + python_marks + power_bi_marks +
english_marks)/5), 2) AS avg_score
FROM student_marks
GROUP BY location
ORDER BY avg_score DESC;
```

| Location | Avg Score | Bar Chart |
|---|---|---|
| Tokyo | 88.90 | |
| Berlin | 86.50 | |
| Sydney | 85.20 | |
| Melbourne | 84.80 | |
| Toronto | 84.10 | |

Output – Output – Top locations: Tokyo (88.9), Berlin (86.5), Sydney (85.2)

### ☑ 5. Highest marks per subject

Insight: Insight: Displays top score achieved in each subject.

Query:

```
SELECT
    '5. Max SQL'      AS subject, MAX(sql_marks)      AS max_marks FROM
student_marks
UNION ALL
SELECT '5. Max Excel'    , MAX(excel_marks)   FROM student_marks
UNION ALL
SELECT '5. Max Python'   , MAX(python_marks)  FROM student_marks
UNION ALL
SELECT '5. Max Power BI', MAX(power_bi_marks)FROM student_marks
UNION ALL
SELECT '5. Max English' , MAX(english_marks) FROM student_marks;
```

Output – Output – SQL: 100, Excel: 100, Python: 100, Power BI: 100, English: 100

```
Output:
5. Max SQL          100
5. Max Excel        100
5. Max Python       100
5. Max Power BI     100
5. Max English      100
```

## ☑ 6. Average marks by age group

Insight: Insight: Evaluates performance trends with respect to age.

Query:

SELECT age,

```
SELECT
    '6. Avg by Age' AS info,
    age,
    ROUND(AVG((sql_marks + excel_marks + python_marks + power_bi_marks
+ english_marks)/5), 2) AS avg_marks
FROM student_marks
GROUP BY age
ORDER BY avg_marks DESC;
```

Output – Output – Age 22: 87.4, Age 23: 85.6, Age 24: 83.9

```
Output:
6. Avg by Age    22  87.4
6. Avg by Age    23  85.6
6. Avg by Age    19  85.1
6. Avg by Age    20  84.9
6. Avg by Age    21  84.7
6. Avg by Age    24  83.9
6. Avg by Age    18  83.5
6. Avg by Age    25  83.2
```

## ☑ 7. Overall performance per student

Insight: Insight: Calculates each student's overall average score.

Query:

```sql
SELECT
    '7. Top 10 Overall' AS info,
    student_id,
    ROUND((sql_marks + excel_marks + python_marks + power_bi_marks +
english_marks)/5, 2) AS overall_percentage
FROM student_marks
ORDER BY overall_percentage DESC
LIMIT 10;
```

```
Output:
7. Top 10 Overall    5   96.4
7. Top 10 Overall    10  95.6
7. Top 10 Overall    158 95.0
7. Top 10 Overall    201 94.6
7. Top 10 Overall    258 94.4
7. Top 10 Overall    267 93.4
7. Top 10 Overall    360 93.2
7. Top 10 Overall    387 92.6
7. Top 10 Overall    393 92.4
7. Top 10 Overall    426 91.6
```

### ☑ 9. Analytical vs Coding skill comparison

Insight: Insight: Compares average analytical (SQL, Power BI, Excel) vs coding (Python) scores.

Query:

```
SELECT
    '9. Analytical' AS skill_group,
    ROUND(AVG((sql_marks + excel_marks + power_bi_marks)/3), 2) AS
avg_score
FROM student_marks
UNION ALL
SELECT '9. Coding (Python)', ROUND(AVG(python_marks), 2) FROM
student_marks;
```

Output – Output – Analytical: 84.4, Coding: 82.9

```
Output:
9. Analytical         84.43
9. Coding (Python)    82.9
```

## ☑ 10. Location with highest overall performance

Insight: Insight: Determines which location has top average scores across subjects.

Query:

```
SELECT
    '10. Top Location' AS info,
    location,
    ROUND(AVG((sql_marks + excel_marks + python_marks + power_bi_marks
+ english_marks)/5), 2) AS avg_total
FROM student_marks
GROUP BY location
ORDER BY avg_total DESC
LIMIT 1;
```

Output – Output – Tokyo shows the highest overall performance (avg 88.9)

```
Output:
10. Top Location Tokyo    88.9
```