**Module-3 Operators**

1. Create a Student\_DB with the tables created from the datasets shared (students.csv files)

**Sol:** To create a Student\_DB with the tables created from the "students.csv" file, we can follow the steps below:

1. Create a new database called "Student\_DB" using the following SQL command:

CREATE DATABASE Student\_DB;

1. Create a new table called "students" in the "Student\_DB" database using the following SQL command:

USE Student\_DB;

CREATE TABLE students (

enrollment\_no INT PRIMARY KEY,

name VARCHAR(255),

class VARCHAR(255),

section VARCHAR(255),

marks INT

);

1. Import the data from the "students.csv" file into the "students" table using the following SQL command:

LOAD DATA INFILE '/path/to/students.csv'

INTO TABLE students

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS;

1. Write a SQL query to display all the unique values in the "section" column of the "students" table.

**Sol:**

SELECT DISTINCT section

FROM students;

**Explanation:** The above SQL query will select all distinct values from the "section" column of the "students" table using the "DISTINCT" keyword. This will remove any duplicate values and display only the unique values.

The query result will show all the unique values in the "section" column of the "students" table. The first column will display the unique values of the "section" column.

1. Write a SQL query to display the top 5 highest marks from the "students" table.

**Sol:**

SELECT student\_name, marks

FROM students

ORDER BY marks DESC

LIMIT 5;

**Explanation:** The above SQL query will select the "student\_name" and "marks" columns from the "students" table and sort the results in descending order of marks using the "ORDER BY" keyword. The "LIMIT" keyword limits the number of rows returned to 5, which will give us the top 5 highest marks.

The query result will show the names and marks of the top 5 students in the "students" table. The first column will display the names of the students, and the second column will display their marks.

1. Write a SQL query to display the number of students in each class from the "students" table.

**Sol:**

SELECT class, COUNT(\*) AS num\_students

FROM students

GROUP BY class;

**Explanation:** The above SQL query will count the number of rows in the "students" table for each unique value in the "class" column using the "COUNT" aggregate function. The "GROUP BY" clause is used to group the results by the "class" column. The "AS" keyword is used to rename the resulting column to "num\_students".

The query result will show the number of students in each class in the "students" table. The first column will display the unique values of the "class" column, and the second column will display the number of students in each class.

1. Write a SQL query to display the average marks of all in each section from the "students" table.

**Sol:**

SELECT section, AVG(marks) AS avg\_marks

FROM students

GROUP BY section;

The above SQL query will calculate the average of the "marks" column for each unique value in the "section" column using the "AVG" aggregate function. The "GROUP BY" clause is used to group the results by the "section" column. The "AS" keyword is used to rename the resulting column to "avg\_marks".

The query result will show the average marks of all students in each section in the "students" table. The first column will display the unique values of the "section" column, and the second column will display the average marks for each section.

1. Write a SQL query to display the names and marks of all students in the "students" table in descending order of enrollment\_no.

**Sol:**

SELECT name, marks

FROM students

ORDER BY enrollment\_no DESC;

**Explanation:** The above SQL query will select the "name" and "marks" columns from the "students" table and order the result set by the "enrollment\_no" column in descending order using the "ORDER BY" clause.

The query result will show the names and marks of all students in the "students" table in descending order of enrollment\_no.

1. Write a SQL query to display the names of all students who scored in the "science\_marks" and get marks greater than 60.

**Sol:**

SELECT name

FROM students

WHERE science\_marks > 60;

**Explanation:** The above SQL query will select the "name" column from the "students" table where the "science\_marks" column is greater than 60.

The query result will show the names of all students who scored in the "science\_marks" and got marks greater than 60.