##### EXPERIMENT NO:-05

**Aim:** **Perform Simple queries, string manipulation operations and aggregate functions**

**Theory:**

SQL has many built-in functions for performing processing on string or numeric data. Following is the list of all useful SQL built-in functions –

1.Aggregate functions

2.Arithmetic functions/Numeric functions

3.Character functions / String functions

4.Date functions

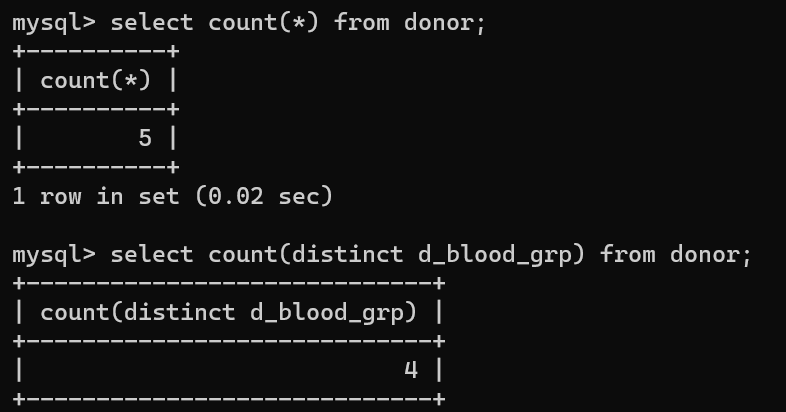
**1. Aggregate functions**

**Count :** To determine the number of rows or non NULL columns of the table

**Q37 : List the number of donors that donates blood.**

Ans : select count(\*) from donor;

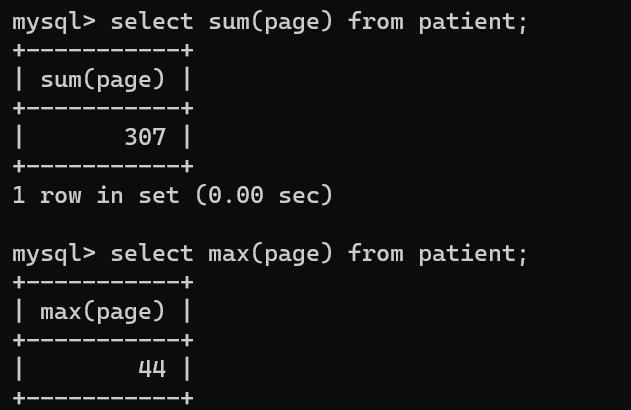
**Q38 : List the number of blood group available in the donor table.**

 Ans : select count(distinct d\_blood\_grp) from donor;

**SUM : To determine the sum of all selected column**

**Q39 : List the total sum of weight of patients.**

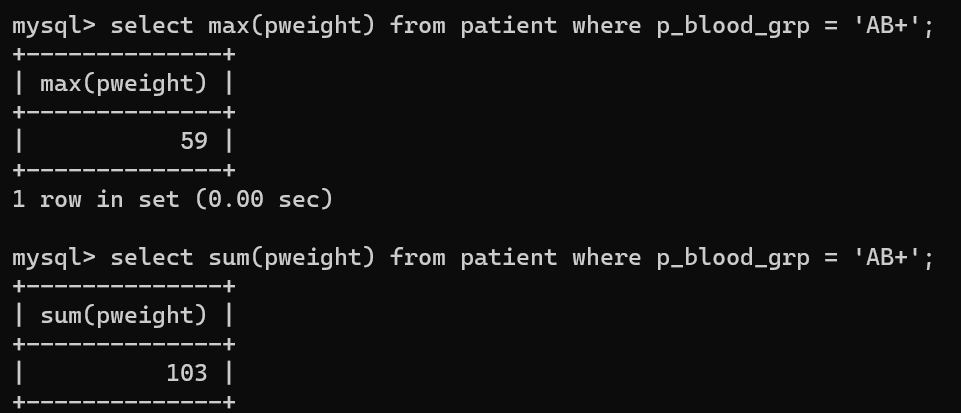
Ans : select sum(page) from patient;



**MAX : To determine the largest of all selected values of a column**

**Q40 : List the maximum weight of patient as a blood group AB+.**

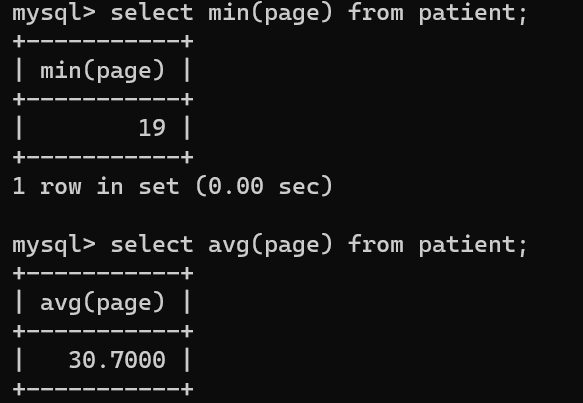
Ans : select max(pweight) from patient where p\_blood\_grp = ‘AB+’;



**Min : To determine the smallest of all selected values of a column**

**Q41 : List the minimum age from patient table.**

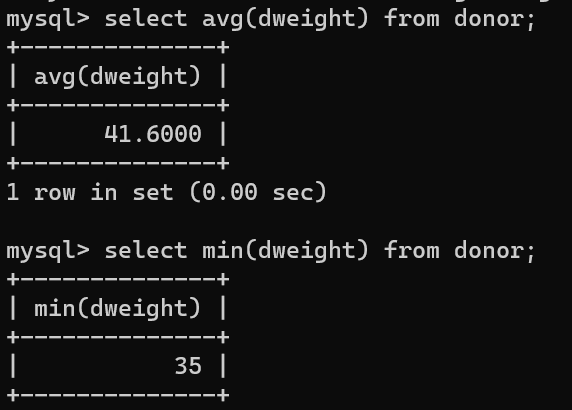
Ans : select min(page) from patient;

****

**AVG : To determine the average of all selected values of a column**

**Q42 : List the average weight of donor.**

Ans : select avg(dweight) from donor;



**2. Numeric functions**

MS SQL Server numeric functions can be applied on numeric data and will return numeric data.

Following is the list of Numeric functions with examples.

## ABS()

Absolute value will come as output for numeric expression.

### Example

### The following query will give the absolute value.

**Q43:** SELECT ABS(-10);

**Output:** 10

**CEIL()**

The CEIL() function returns the smallest integer value greater than or equal to a given number.

**Example**

CEIL(5.2) would return 6.

**Q44:** SELECT CEIL(5.2);

**Output:** 6

## FLOOR()

## The FLOOR() function returns the largest integer value less than or equal to a given number.

**Example**

FLOOR(5.8) would return 5.

**Q45:** SELECT FLOOR(5.8);

**Output: 5**

## ROUND()

## The ROUND() function rounds a given number to a specified number of decimal places.

**Example**

ROUND(3.14159, 2) would return 3.14.

**Q46:** SELECT ROUND(3.14159, 2);

**Output:** 3.14

**SQRT()**

The SQRT() function returns the square root of a given number.

**Example**

SQRT(25) would return 5.

**Q47:** SELECT SQRT(25);

**Output:** 5

## RAND()

## The RAND() function generates a random number between 0 and 1.

**Example**

RAND() would return a value such as 0.643726.

**Q48: SELECT RAND();**

**Output:** returns a random value between 0 and 1, such as 0.643726

**3. String functions**

MS SQL Server String functions can be applied on string value or will return string value or numeric data.

Following is the list of String functions with examples.

**ASCII():** This function is used to find the ASCII value of a character.

**Q49:** SELECT ascii('t');

**Output:** 116

**CONCAT():** This function is used to add two words or strings.

**Q50:** SELECT 'Geeks' || ' ' || 'forGeeks' FROM dual;

**Output:** ‘GeeksforGeeks’

**INSERT():** This function is used to insert the data into a database.

**Q51:** INSERT INTO database (geek\_id, geek\_name) VALUES (5000, 'abc');

**Output:** successfully updated

**INSTR():** This function is used to find the occurrence of an alphabet.

**Q52:** INSTR('geeks for geeks', 'e');

**Output:** 2 (the first occurrence of ‘e’)

**Q53:** INSTR('geeks for geeks', 'e', 1, 2 );

**Output:** 3 (the second occurrence of ‘e’)

**REVERSE():** This function is used to reverse a string.

**Q54:** SELECT REVERSE('geeksforgeeks.org');

**Output:** ‘gro.skeegrofskeeg’

**RIGHT():** This function is used to SELECT a sub string from the right end of the given size.

**Q55:** SELECT RIGHT('geeksforgeeks.org', 4);

**Output:** ‘.org’

**Group by clause:**

The GROUP BY statement groups rows that have the same values into summary rows.

The GROUP BY statement is often used with aggregate functions (COUNT, MAX, MIN, SUM, AVG) to group the result-set by one or more columns.

Example :

Detpno No of Employees

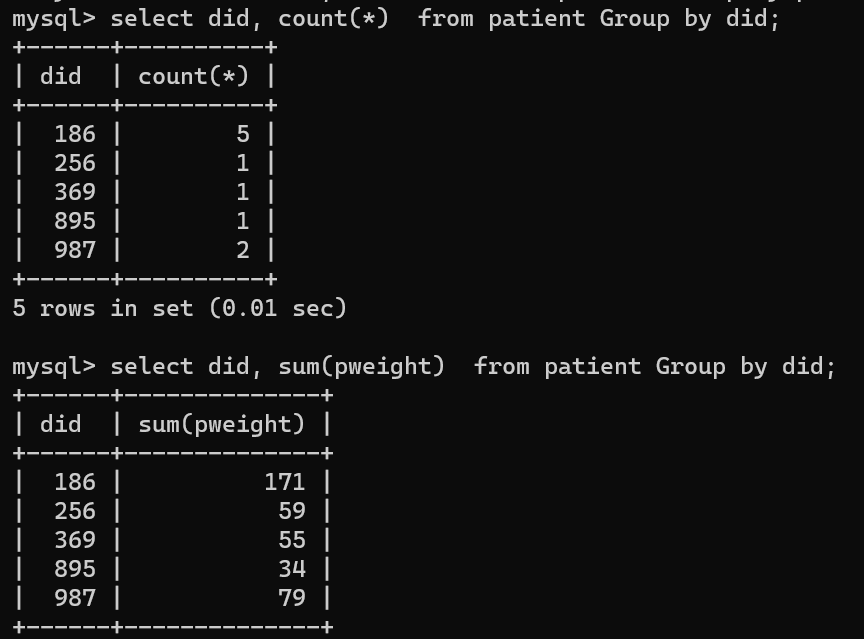
10 3

20 5

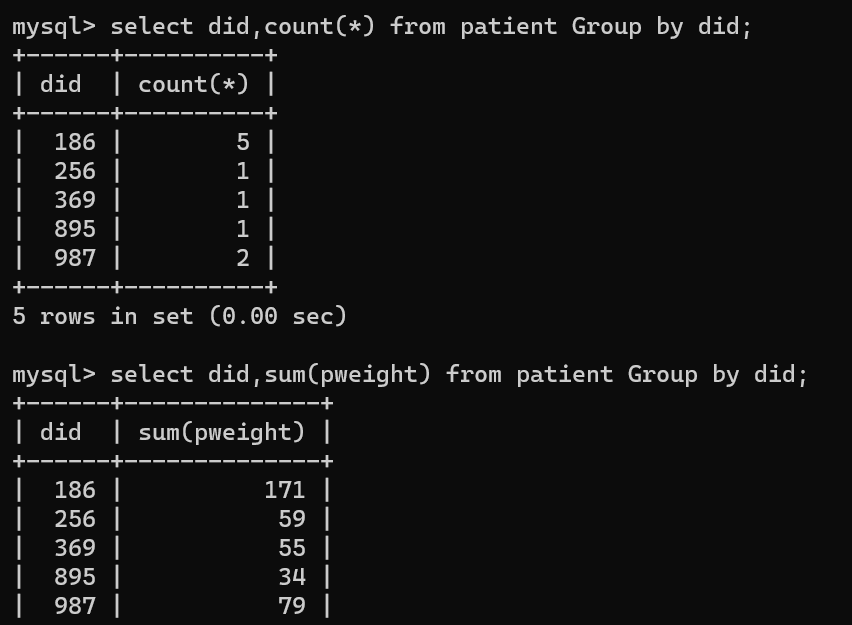
30 6

**Q56 : List the donor id and number of patients that receive the blood from donor.**

Ans : select did, count(\*) from patient Group by did;

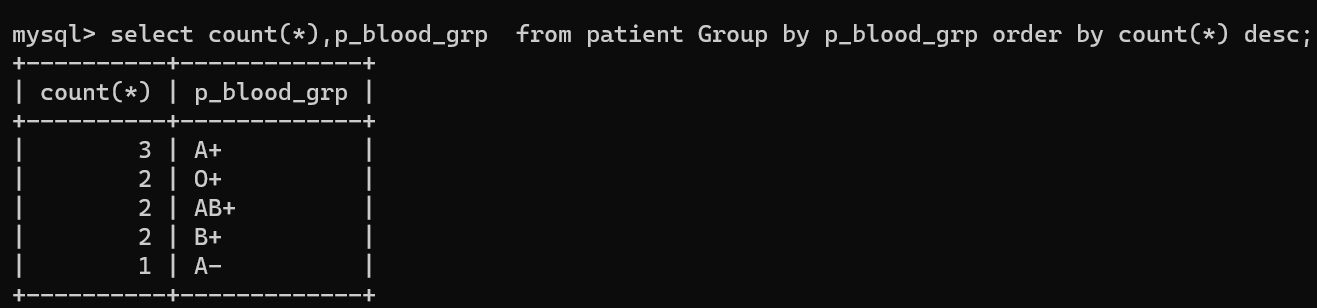


**Q57 : List the donor id and the total weight from donor table.**



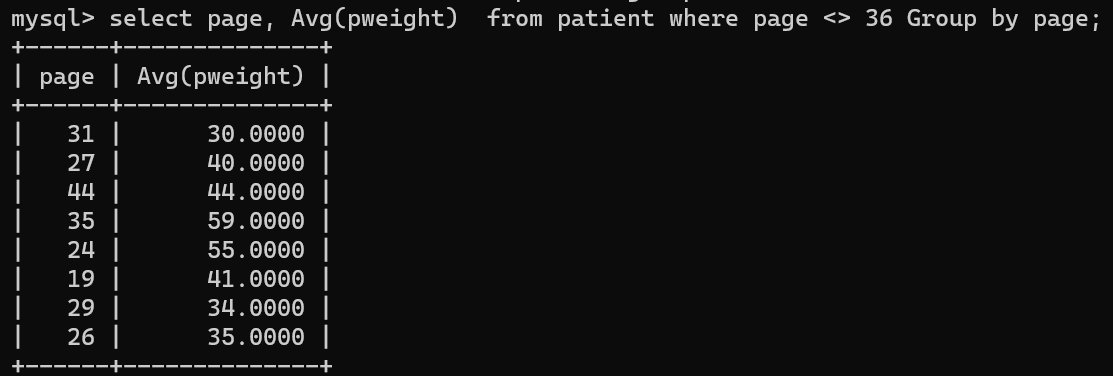
**Q58 : List the blood group and the number of patient having each blood group. The result should be in descending order of the number of patient.**

Ans : select count(\*) from patient Group by p\_blood\_grp order by 2 desc;



**Q59 : List the average weight from each blood group excluding AB+.**

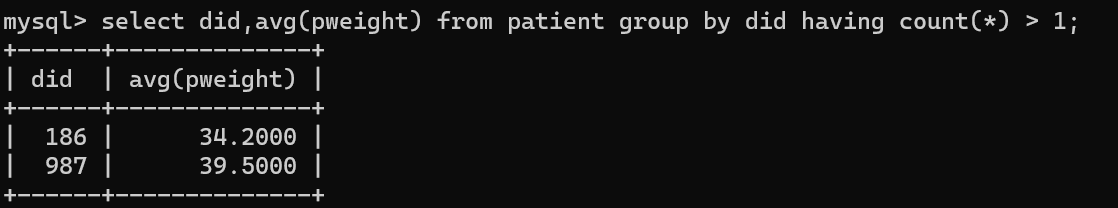
Ans : select page, Avg(pweight) from patient where page <> 36 Group by page;



**Having Clause:**

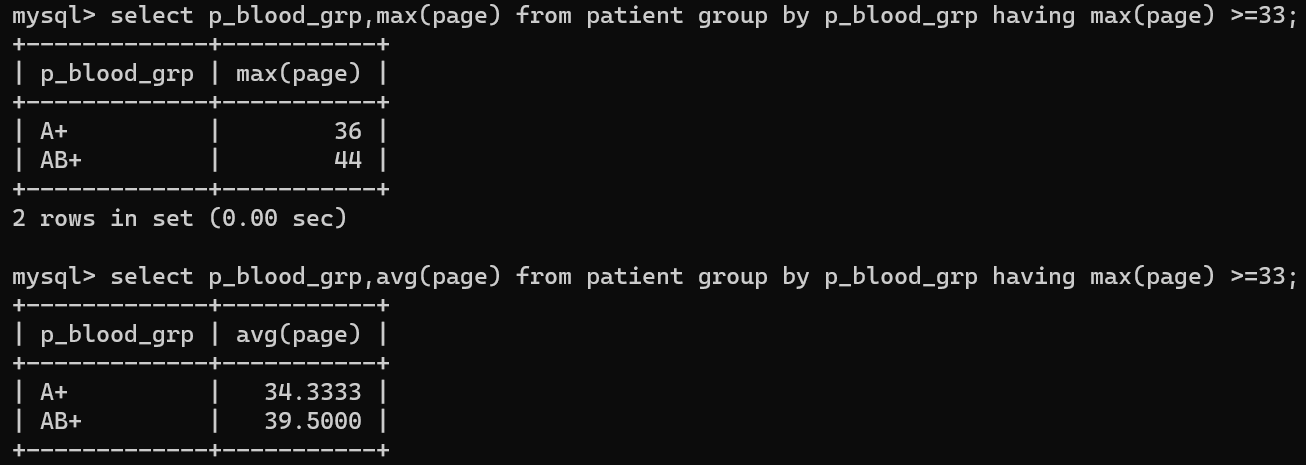
**Q60 : List the average weight for all blood groups more than one patient.**

Ans : select did,avg(pweight) from patient group by did having count(\*) > 1;



**Q61 : List blood groups of all the patient where maximum age is greater than or equal to 33.**

Ans : select p\_blood\_grp,max(page) from patient group by p\_blood\_grp having max(page) >=33;



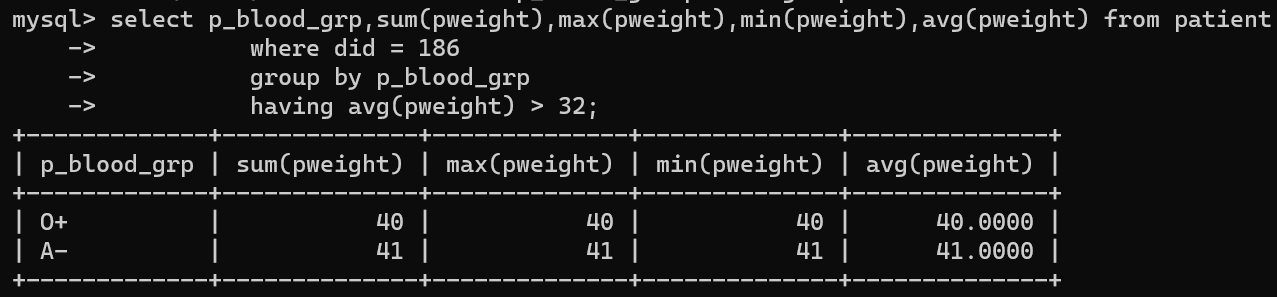
**Q62: List total, maximum ,minimum and average weight of patient blood group wise for did no 186 and display only those rows whose average weight is greater than 32.**

Ans : select p\_blood\_grp,sum(pweight),max(pweight),min(pweight),avg(pweight) from patient

where did = 186

group by p\_blood\_grp

having avg(pweight) > 32;



**Conclusion:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **R1** | **R2** | **R3** | **R4** | **Total** | **Sign with Date** |
| **(3)** | **(5)** | **(4)** | **(3)** | **(15)** |  |
|  |  |  |  |  |