aggregate functions

- it is refers to combination or collection of values into a single entity.
- it is operate on a set of values and return a single value as a result.

sum, count, avg, max, min etc.

1) **count( )** :→ counts the number of rows

count (37, 24, 3, 8, 9) :→ 5

count (5, 3, null, 4, null) :→ 3

- generally aggregate functions ignored null values.

(2) sum( ) :

sum (3, 2, 5, 10) = 20

sum (3, 2, null, 5, null) = 10

(3) avg( ):

avg (3, 2, 5, 10) = 5

(4) max( ) → get maximum value

(5) min( ) → get minimum value.

general syntax

```
select    aggregate function ( column )
from      {table_name}
where     condition
```

students

s_id	name	age	c_id
101	Abhijit	23	202
102	Bharat	25	205
103	harish	null	202
104	Kunal	22	null
105	null	null	205
106	neeraj	23	203

Q) count number of students in students table.

```
select    count (age)
from      students ;        4
```

```
select    count (*)         6
from      students;
```

Q) calculate avg of students.

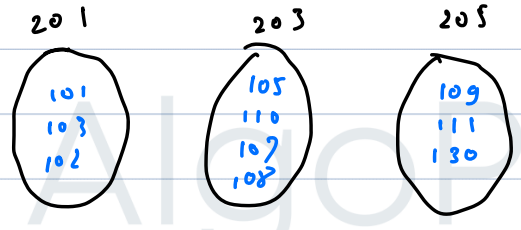
Q) find all the student whose is age maximum.

Q) count number of students who enrolled in c_id = 205 :

```
select count(*)
from students
where c_id = 205;
```

2

Q) count number of students in each course.



group by

```
select c_id, count(*)
from students
group by c_id;
```

↗ { group }

select {i) column_name , (ii) aggregate function( )

Instructor table { i_id, i_name, salary, d_id }

Q) display names of instructor whose salary is greater than avg salary of all instructors.

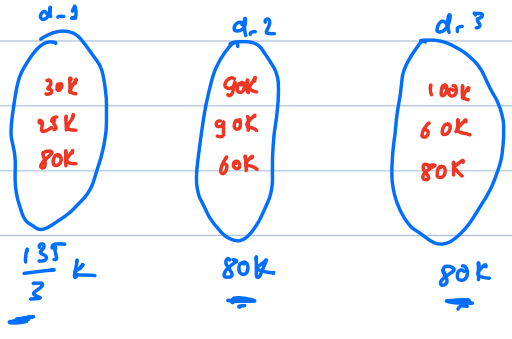
```
select   inst_name
from     instructors
where    salary > avg(salary)
```

X

- we can't use aggregate function with where.
- it is work on only individual row

```
select   inst_name
from     instructors
where    salary > ( select avg(salary)
                    from instructors);
```

Q) display the dep-id and avg salary of dep_id whose avg salary of department is greater than 60,000 ?



```
select  dep-id , avg(salary)
from    instructors

group by dept-id
```

having

- it is used after group by
- it is used to apply condition on group.
- we can use aggregate function.

```
select  dep-id , avg(salary)
from    instructors
group by dept-id
having  avg(salary) > 60000;
```

break till 9:27

Built in functions

String

(i) length()

```
select  length("abcdef");   6
```

(ii) LOWER()

```
select  Lower("ABcdDeF")
```

abcddef

(iii) UPPER()

```
select  upper("abCcdDf")
```

ABCCDDF

(iv) CONCAT()

```
select  CONCAT("first-name", "last_name");
```

(v) substring()

```
substring( string , starting_position)

select  substring("abcdefgh" , 3)
                  1 2 3 4 5 6 7 8
                  3 included
```

cdefgh

* In this language 1-based indexing is used.

```
substring( string, starting_position, length)
                                      ↓
                                  length of the
```

output string.

select substring("12345678 abcdefgh", 2, 5);

bcdef

8) find 5 characters from starting in "12abcdefgh".

select substring("12abcdefgh", 1, 5);

(vi) LEFT() → it gets characters from starting

select LEFT("12abcdefgh", 5);

12abc

8) find 5 characters from end in "12abcdefgh".

(vii) Right

select Right("12abcdefgh", 5);

defgh

(viii) trim :→ its remove leading and trailing spaces.

trim(" .... Hello .--. ") : "Hello"

leading spaces

trailing spaces.

(ix) LTRIM( ): it removes leading spaces

(x) RTRIM( ): it removes trailing spaces

(xi) LOCATION( ):

LOCATION(string, orgstring, start_pos)
↓
optional

select LOCATION("cd", "abcdcded") = 3
(1 2 3 4 5 6 7 8)

- if string is present then it gives first occurence of that string otherwise return 0;

select LOCATION("adc", "abcdcded") = 0
(1 2 3 4 5 6 7 8)

Numeric

1) abs( ); → it gives positive value.

abs(-10) = 10

2) floor( ) :→ just smaller or equal integer value.

floor(3.728) = 3

floor(3.00) = 3

3) ceil : just greater or equal integer value.

ceil (7.328) = 8

ceil (7.000) = 7

4) round( )

Round(328.497) = 329

Round (374.37245, 2) 374.37

Round (32.4578, 1) 32.5

↓

number after decimal

(5) truncate()

truncate (374.37245, 2) = 374.37

truncate (32.4578, 1) = 32.4

(6) RAND(): it generates random value between 0 to 1;

Q) general random number from 0 to 100:

FLOOR(rand() * 100);

0.345 = 34

0.476 x100

= 47

(7) power()

power(2, 5) = 32

DATE

1) now() → curr date and curr time.

2) curdate() → curr date

3) curtime() → curr time

4) dayname() →

dayname(curdate()) → friday

(5) date_add() →

date_add(curdate() interval 6 month)

7 day)

2 year)

6) date_sub (curdate( ) interval 6 month)
7 day)
2 year)

7) datediff (" 24-02-29" , " 24-03-01"); 1

8) date_format (" 24-02-29" , " %d-%m-%y");
y M D

```sql
-- number of instructors
select count(instructor_name) as count_of_inst
from instructors;

-- number of distinct instructors
select count(distinct instructor_name) as count_of_inst
from instructor;

-- calculate average salary of all instructors
select avg(salary) as avg_sal
from instructors;

-- total salary
select sum(salary) as avg_sal
from instructors;

-- find number of instructors in each department
select department_id , count(*)
from instructors
group by department_id;

-- find the all departments with avg_sal whose average salary is greater than 50000
select department_id , avg(salary)
from instructors
group by department_id
having avg(salary) > 60000;

-- - find the all departments with avg_sal where in each department number of instructor is atleast 2
select department_id , avg(salary)
from instructors
group by department_id
having count(*) >= 2;

select now();

-- get all the instructor who has more than 10 year of experience
select instructor_name
from instructors
where datediff(curdate() , joining_date) /365 > 10;

```