Number function.

EMP

SAL(float)
1234.567
1852.019
1375.825
1748.616

SQL - float.
oracle - number (7,3)
(implied decimal point.

- select round (sal) from emp;
  This will roundup the sal
  This function is overloaded.

  O/P. 1235
  1852
  1376
  1749

- select round (sal, 1) from emp;
  O/P - 1234.6, 1852, 1375.8, 1748.6.
- select round (sal, 2) from emp;
  roundup 2 digits after decimal.
  O/P - 1234.57, 1852.02, 1375.83, 1748.62
- select round (sal, -2) from emp;
  O/P - 1200, 1900, 1400, 1700.
- select round (sal, -3) from emp;
  O/P - 1000, 2000, 1000, 2000
  round up to nearest 1000, 10000, etc.
- select truncate (sal, 0) from emp;
  O/P - 1234, 1852, 1375, 1748.
  It removes the decimals.
  normally used in date calculation..(age calculation)
- select truncate (sal, 1) from emp;
  O/P - 1234.56, 1852.01, 1375.82, 1748.61
- select truncate (sal, -2) from emp;
  O/P - 1200, 1800, 1300, 1700.
- select ceil(sal) from emp; Ceiling
  O/P - 1235, 1853, 1376, 1749.
  add 1 to number if there is anything after decimal.
  Practical use is, it is use in billing / bill payments.

- select floor(sal) from emp;

o/p -

- select truncate(3.6,0), floor(3.6), truncate(-3.6,0) Floor(-3.6) from dual;

o/p - 3 , 3 , -3 , -4

- select sign (-15) from dual;

o/p - -1 -1 for negative no.
1 for +ve no.
0 for equal to zero.

source code:-
```
if x>0 then
    return 1;
elseif x<0 then
    return -1;
else
    return 0;
endif;
```


use:- 1. check if num is +ve or -ve.
2. sign (sp-cp) +1 = profit
-1 = loss
0 = No profit / No loss.
3. sign (temperature)
4. sign (bank - balance).
5. sign (blood-group)
6. sign (corona - test).
7. sign (medical -test).
8. sign (marks).
9. sign (a-b) ← to find greater of 2 no.
10. sign (gender)
11. sign (base- sen sex)
12. sign (imaginary -no)

- Lateral Thinking (Books & videos)
- select mod(9,5) from dual; o/p :- 4.
- select mod(8.22, 2.2) from dual; o/p :- 1.62
- select sqrt(81) from dual; o/p = 9.
(works only for +ve no.)

- select power(10, 3) from dual; o/p - 1000.
- select power(1000, 1/3) from dual; o/p - 10.
  (cube root calculation. ((cube root).
- select abs(-10) from dual; ← absolute value.
  always returns +ve no.
  o/p - 10.
- sin(x), cos(x), tan(x)
  x → radians.
- ln(y), log(n, m).
  ln(y) → log$_e$(y).
  log(n, m) → log$_n$(m)

Date functions.

① 'YYYY/MM/DD'
② '1000/01/01' to '9999/12/31'
③ date1 - date2.
④ internally date is stored as a fixed-length no
   no. of days since '01/01/1000' AD.
⑤ it occupies 7 bytes of storage.
⑥ date and time is stored together.
⑦ default value for time is 12 am midnight
⑧ 1970 is the cut-off year.

EMP.

HIRE DATE.

1st Jan 1582 - 10th oct 1500.

2019-10-15
2019-12-31
2020-01-15

Snehal Sawant

- select sysdate() from dual;
  returns system date / server date & time.

'2021-11-24' → o/p = '2021-11-24 15:15:16'
'2021-11-24'
'2021-11-24'

returns server date and time when the statement executed.

- select now() from dual;
  returns server date and time when statemen began to execute.
- select sysdate(), now(), sleep(10), sysdate(), now() from dual;

  15:21:54, 15:21:54
  15:22:04 , 15:21:54.

Sysdate() → date and time display.
now() → use to maintain logs of operations
eg. insert, update, delete, etc)

- select adddate(sysdate(),1) from dual;
  returns tomorrows date.
- select adddate(sysdate(),2) from dual;
  day after tomorrows.
- select adddate(sysdate(),-1) from dual;
  previous date.
- select datediff(sysdate(), hiredate) from emp;
  - returns no. of days beth the 2 dates.
- select date_add(hiredate, interval 2 month) from dual;
- select date_add(hire date, interval -2 month) from dual;
- select date_add(hire date, interval 1 year) from emp;
- ~~last~~
  select last_day(hire date) from emp;
  This function is available only in MySQL & oracle.
  → atterdance calculation.
  → interest calculation.
- select dayname(sysdate()) from dual;
  'Wednesday'

Snehal Sawant

- select upper (dayname (sysdate())) from dual;
  'WEDNESDAY'.
- select substr (dayname (sysdate(), 1, 3) from dual;
  'Wed'.
- select addtime ('2010-01-15 10:00:00', 1') from dual;
  O/P '2010-01-15 11:00:01' (substract sec)/ (add sec).
- select addtime ('2010-01-15 11:00:00', '-1:00:00') from dual;
  substract sec.
- select addtime ('2010-01-15 11:00:00', '1:30:45') from dual;
- select addtime ('2010-01-15 11:00:00.000000',
  '1:30:45.123456') from dual;

List functions :
independent of data type.

EMP

ENAME	SAL	comm
A	5000	5000
B	6000	
C		700

- select * from emp where comm = null;
  O/P = No rows selected.
  - Null value has ascii value 0
  - any comparision done with null, returns null.

Pessimistic query → Searching for nullvalue.
- select * from emp where comm is null;

is null → it's a special operator.

Snehal Sawant

- select * from emp where comm != null;
  No output.
  - any comparision done with null, returns null.
- select * from emp where comm is not null;
- select sal+comm from emp; no output.
  - any operation done with null, returns null.
- select sal+ ~~co~~ ifnull(comm,0) from emp;

source code:- if comm is null then
return 0;
else
return comm;
end if;

o/p: 5500
6000
.

- select ifnull(sal,0) + ifnull(comm,0) from emp;

o/p: 5500
6000
700.

- ifnull(comm, 100)
- ifnull(city, 'Mumbai')
- ifnull(orderdate, '2021-04-01').

EMP.

SAL
1000
2000
3000
4000
5000.

- select greatest(sal, 3000) from emp;
  return greater of two values.
  Use:-
  1. used to set a lower limit on some value.
  e.g. Bonus = 10% sal, min Bonus = 300.
- select greatest(sal*0.1, 300) from emp;
- greatest(val1, val2, val3, ..., val255).
  max = 255.
- greatest(num1, num2, num3).
- greatest('str1', 'str2', 'str3', 'str4');

greatest(date1, date2, date3).
all these calculations are compared using asciival.

- set x = greatest(a,b,c,d);
- select least(sal, 3000) from emp;
  returns the small of the two
  use:- use to set an upper limit on some value
  e.g. cbashback = 10% amt, max cbasback = 1000.
- select least(amt*0.1, 1000) from orders;
  least(val1, val2, val3, . . . , val255);
  least('str1','str2','str3'. . . ,'str255');

independent of datatypes:

- # case expression:

EMP.

SAL	DEPTNO.
1000	10
2000	10
3000	20
4000	30
5000.	40.

. select
case
when deptno = 10 then 'Training'
when deptno = 20 then 'Exports'
when deptno = 30 then 'Sales'
else 'others' (else is optional)
end "DEPTNAME"
from emp;

o/p: DEPTNAME
Ten
Ten
Twenty
.

use:-
a. Encoding and Decoding.
select
case.
when dept no = 10 then 'Ten'
when dept no = 20 then 'Twenty'
when dept no = 30 then 'Thirty'
end "DEPTNAME"
from emp;

Snehal Sawant

- If you don't supply else it will return null value.

Environment functions:-

- select user() from dual; o/p - username@localhost
- show character set;

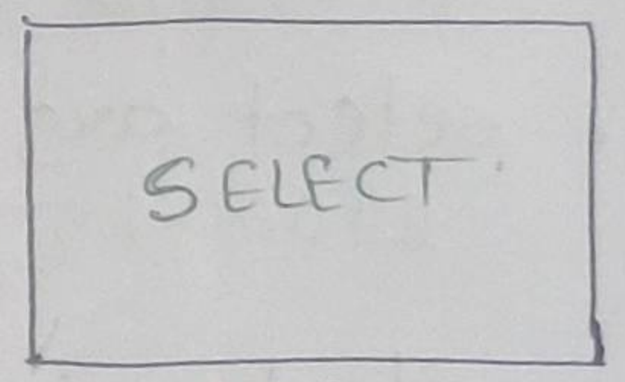
Group functions / Aggregate function:

DB SERVER HD.

EMP

EMPNO	ENAME	SAL	DEPT NO	JOB	MGR (manager).
1	Arun	8000	1	C	4
2	Ali	7000	1	C	1
3	kiran	3000	1	C	1
4	Jack	9000	2	M	
5	Thomas	8000	2	C.	4.

- select
  case
  when job 'M' then 'Manager'
  when job 'C' then 'Clerk'
  end "JOB"
  from emp;



Server Ram.
32 gb Ram Gdac.

- we have an relation bet^n EMPNO & MGR.

Single row function:-

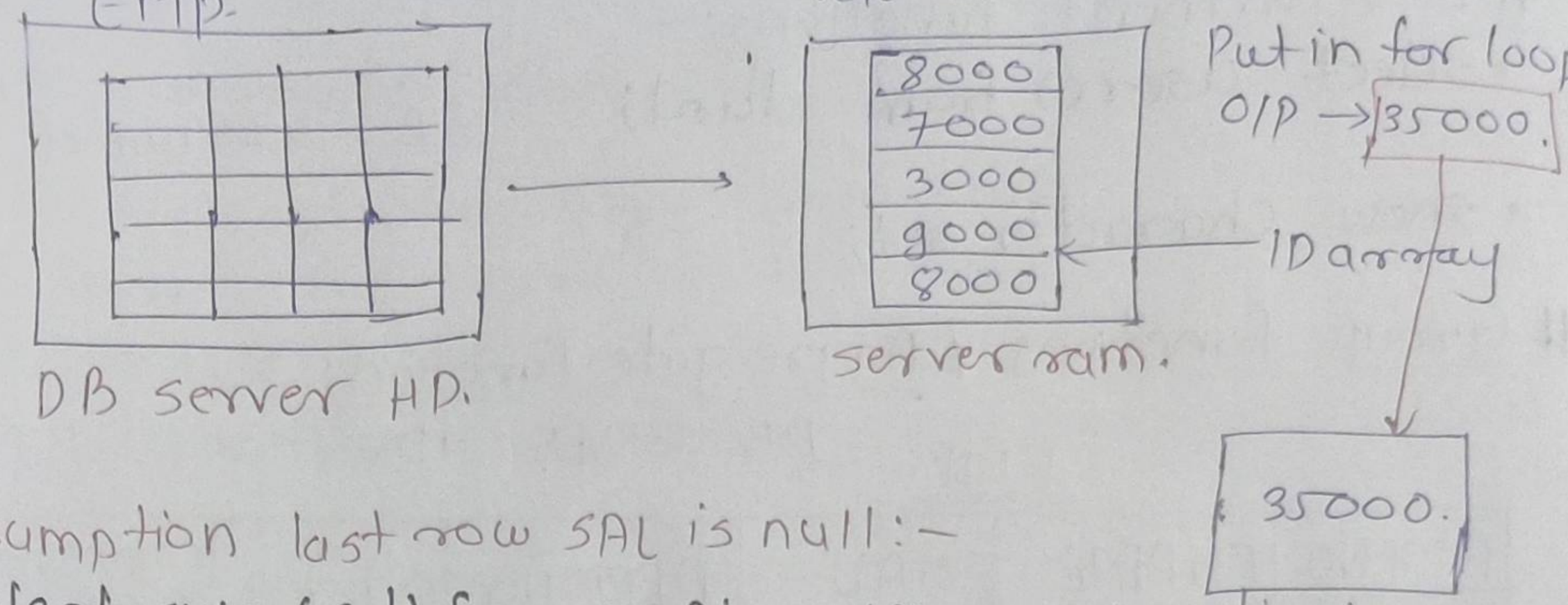
- will operate on 1 row at a time.
- character, Number, date, List, Environment function
- eg. upper(ename), round(sal), etc.

Multi row functions:-

- will operate on multiple at a time.
- Group functions.
- eg. sum(sal), etc.

Snehal Sawant

• select sum(sal) from emp; o/p 35000



#Assumption last row SAL is null:-

• select sum(sal) from emp; o/p:- ~~Null~~ 27000.

Null values are not counted by group function.

• select sum(ifnull(sal,0)) from emp; o/p:- 27000
but ifnull is not required.

• select avg(sal) from emp;
27000/4 → 6750. (default)

• select avg(ifnull(sal,0)) from emp;
27000/5 → 5400;

• select min(sal) from emp; o/p → 3000

• select min(ifnull(sal,0) from emp; o/p → 0

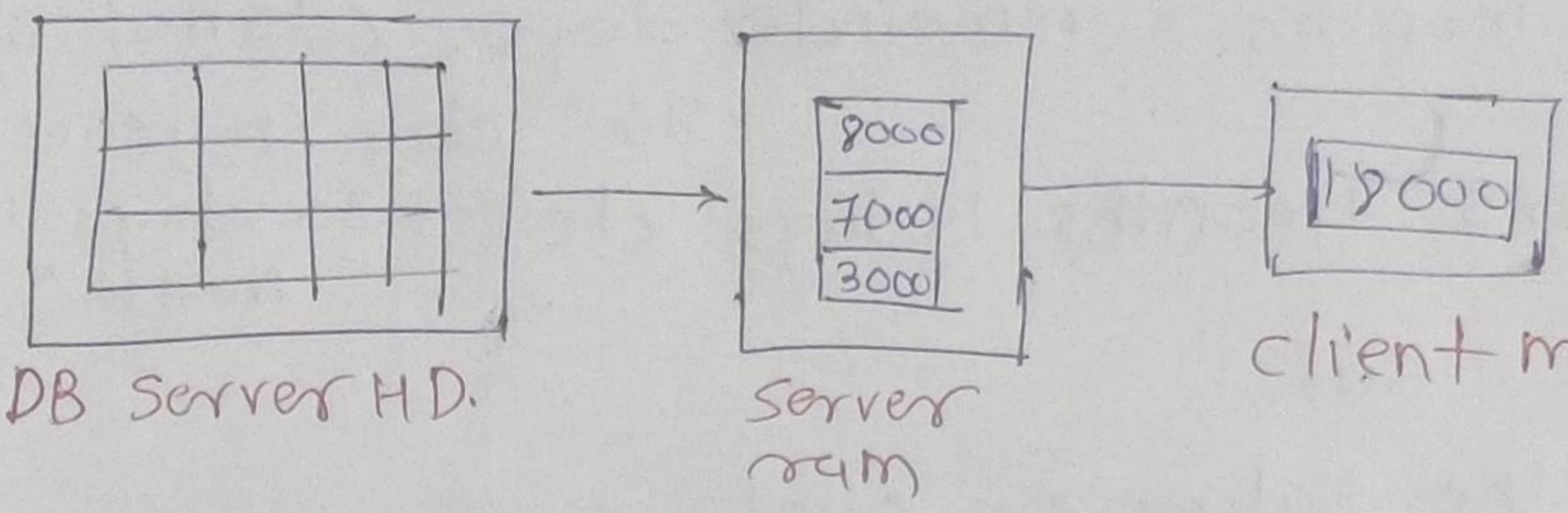
• select max(sal) from emp; o/p → 9000

• select count(sal) from emp; o/p → 4
returns a count of no. of rows where sal is not having a null value.

• select count(*) from emp; o/p → 5
no. of rows in table.

• if considered/assumed last row is null it still counts the null row.

- select sum(sal) from emp where deptno = 1; o/p → 18000



- select max(sal) from emp where job = 'c'; o/p → 7000.

count-query (counting no. of query hits):-

- select count (*) from emp where sal > 7000; o/p → 2.

- select max(sal)/min(sal) from emp;
  9000/3000 → 3

- select sum(sal)/count (*) from emp; ↑ faster.
  27000/5 →

- select avg(ifnull (sal,0)) from emp;
  27800/5 → ↑ function within function (slow)

- sum(column)
  avg(column)
  min (column), min(sal), min(ename), min(hiredate)
  max (column) max(sal), max(ename), max(hiredate)
  count (*) count(sal), count(ename), count(hiredate)
  stddevc column)
  variance (column).

Snehal Sawant

Assumption last row SAL is 8000:-

select count(*), min(sal), max(sal), sum(sal), avg(sal) from emp;
(summary report).

SQL EX = 1 to 4
assig → 1 to 8.