#REGULAR EXPRESSION

#META CHAR ^ . \* + - \\ { } $-- THEY ALL HAVE SPECIAL MEANING

-- $ ending with

-- ^ starting with

-- + 1 to infinite

select ename from employee\_personal\_details;

select \* from emp\_details;

rename table employee\_personal\_details to emp\_details;

select \* from emp\_details where ename regexp '\_\_f';

select \* from emp\_details where ename like '\_\_f';

select \* from emp\_details where ename regexp '..[f]';

select \* from emp\_details where ename regexp 'a{1}'; #a appears atleast 1 time

select \* from emp\_details where ename not regexp 'a'; #a is not present

select \* from emp\_details where ename not regexp '\\\*'; # \* if star is present then we use \\ to nullify the effect of special char

select \* from emp\_details where ename regexp '[a-z]'; #display all the names

select \* from emp\_details where ename regexp '.\*[a]';

select \* from emp\_details where ename regexp '';

#EXTRACT FUNCTION

select extract(week from curdate()) as week\_;

select extract(week from curdate()) as week\_;

select extract(year from curdate()) as year\_;

select extract( minute from curtime()) as min\_;

select \* from emp\_details;

select extract(year from dob) from emp\_details;

select extract(day\_minute from current\_timestamp()) as day\_\_minute;

select extract(day\_hour from current\_timestamp()) as day\_\_minute;

select extract(year\_month from curdate()) as day\_\_minute;

#cast() doesn't change the orignal data,it changes datatype for the fetched data;

select cast(25.66 as char ) from dual;

#WINDOW FUNCTION .........ANAYLITICAL FUNCTION

# PARTITION BY AND ORDER BY ARE OPTIONAL

#ROW\_NUMBER() OVER()

#EID ,AMT

select \* from sal;

select eid,amt,row\_number() over(partition by eid) as rownum from sal;

select eid,amt,row\_number() over(partition by eid,amt) as rownum from sal;

select eid ,max(amt) ,row\_number()over(order by max(amt) desc) from sal group by eid;

#get me the total amt

select eid ,amt ,row\_number()over(order by amt desc) from sal group by eid;

select eid ,amt ,rank()over(order by amt desc) from sal group by eid;

select eid ,amt ,dense\_rank()over(order by amt desc) from sal group by eid;

insert into sal values(5,4,40000,'2022-08-10');

insert into sal values(6,2,4000,'2022-09-15');

select eid,sum(amt) from sal group by eid;

#Detail Partition by concept

create table Traveller(Tid int,sports varchar(10),score int,guide char(10));

insert into Traveller values(1,'Treaking',70,'Sherpa');

insert into Traveller values(2,'Treaking',71,'Sherpa');

insert into Traveller values(3,'climbing',50,'climber');

update Traveller set guide='Sherpa' where Tid=3;

insert into Traveller values(1,'climbing',30,'climber');

insert into Traveller values(2,'climbing',80,'climber\_2');

insert into Traveller values(4,'climbing',55,'climber\_3');

select \* from Traveller;

select \* from Traveller group by score;

use newclass;

create table learner (

lid int,

sub\_name varchar(30),

marks int,

client\_name varchar(10));

insert into learner values (1,'PYTHON',70,'ABC'),(2,'PYTHON',71,'ABC'),(3,'SQL',50,'ABC'),(1,'SQL',30,'DEF'),(2,'SQL',80,'GHI'),

(4,'SQL',55,'DEF');

select distinct(client\_name),dense\_rank() over(order by client\_name) as final from learner;

select \* from (select client\_name,lid,marks, row\_number() over(partition by client\_name order by marks desc) as final from learner) k order by client\_name desc;

update learner

set marks=55 where lid=3;

select \* from learner;

select sub\_name,marks , row\_number() over(partition by sub\_name order by marks) as final ,

rank()over(partition by sub\_name order by marks) as final\_1,

dense\_rank()over(partition by sub\_name order by marks) as final\_2

from learner;