**Default Constraint-**

alter table tblperson modify salary default 50000;

insert into tblperson(id,name,location) values(97,'mamata','patna')-here salary would be inserted as 5000.

**Referential Integrity Constraint-**(Options-Cascade, Set Null, Set Default)(u can set options from UI)

2 tables-emp and empSal-empSal has empId references emp. If from emp record is deleted and empSal has corresponding empId data, then if- If a)Cascade-u cannot delete data b)Set Null – data deleted with empSal corresponding empId becomes null

c)Set Default - data deleted with empSal corresponding empId becomes Default constraint value.

**Check Constraint-**

alter table tblperson add constraint salary\_check check(salary >700)

alter table tblPerson drop constraint salary\_check

**Sequences-**

CREATE SEQUENCE tblperson\_SEQ INCREMENT BY 1 START WITH 1 MAXVALUE

9999999999999999999999999999 MINVALUE 1 NOCYCLE CACHE 20 ORDER

Insert into tblperson(id,name,location,salary) values(tblperson\_SEQ.NEXTVAL,'priya','patna',8)

Q)**Identity insert column like SQL Server is present in oracle?**

**Unique Key Constraint-**

We use UNIQUE constraint to enforce uniqueness of a column i.e. the column shouldnot allow any duplicate values.

Both primary key and unique key are used to enforce uniqueness of a column.SO when do you choose one over the other?

A table can have only one primary key.If you wan to enforce uniqueness on 2 or more columns, then we use unique constraint.

Difference between Primary key And Unique Key Constraint?

1.A table can have only one PK, but more than one unique key.

2.PK does not allow nulls, where as unique key allows one one null.

PK example-empId….Unique Key Example-emailid, employeeID

alter table tblPerson

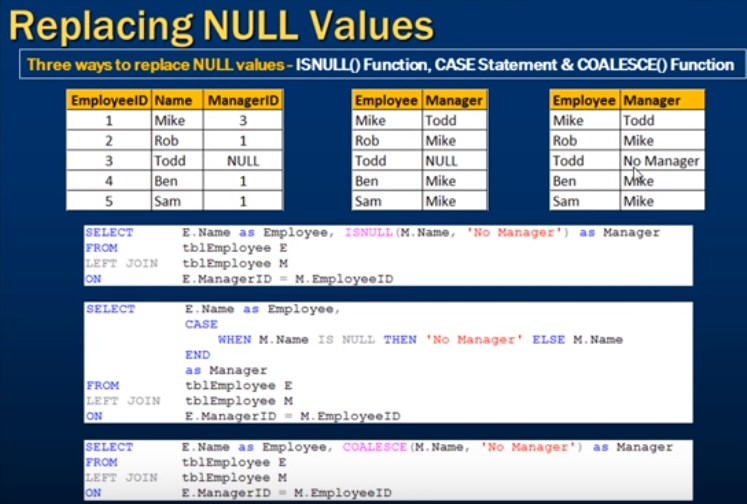
add constraint UQ\_tblPerson\_emailid UNIQUE(emailid)

alter table tblPerson

drop constraint UQ\_tblPerson\_emailid

**Different ways to replace NULL values:**

1. ISNULL()----- (NVL() in Oracle)—same thing
2. CASE STATEMENTS---CASE WHEN expression THEN ‘ ’ ELSE ‘ ’ END
3. COALESCE—powerful function

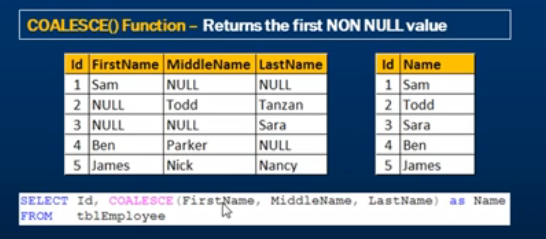


select e.name as employee,CASE WHEN M.NAME IS NULL THEN 'no manager' else m.name end,NVL(m.managerid,0) as managerId

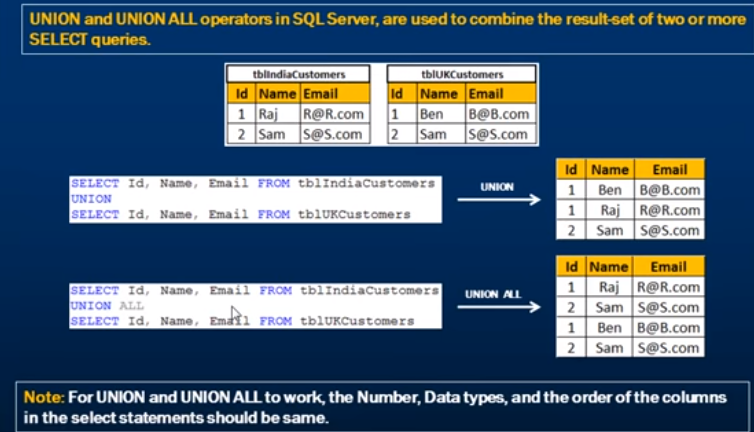
from employee E

left join employee M on e.managerid=m.id

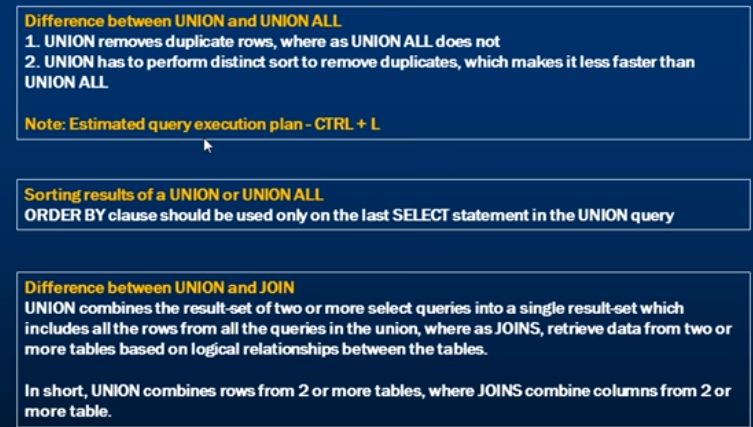
**COALESCE FUNCTION-**



**UNION and UNION ALL**



**UNION UNION ALL AND JOIN**



select id, name from person

union all

select id,firstame from employee

union all

select id,lastname from doctors

group by name //used in last

**STRING FUNCTIONS-**

Char(65) -> A

ASCII(‘A’)->65

LTRIM(lastname)—(select ltrim(lastname) from employee)

RTRIM(lastname)

TRIM(lastname)

LOWER(lastname)

UPPER(TRIM(lastname))

LEFT(‘abcdef’, 3)🡪abc

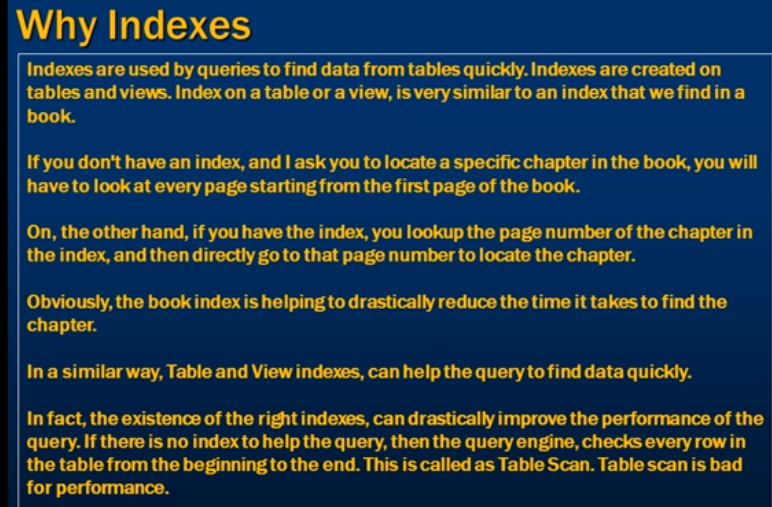
RIGHT(‘abcdef’,3)->def

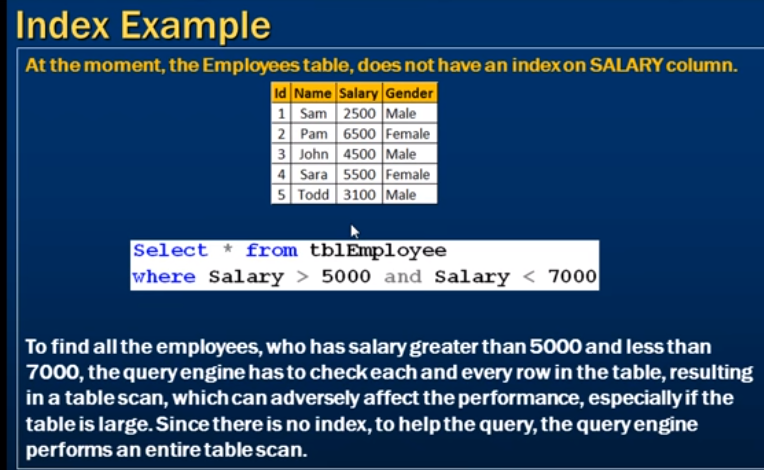
INSTR(‘@’,swati@optum.com)->6(gives position of character)

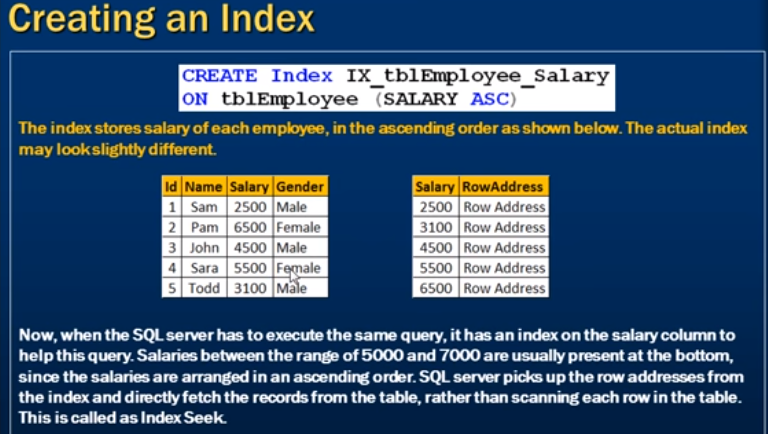
SUBSTR(expression,startindex,length)

**Q)Temporary table in oracle?????**

**INDEXS----(for quick search)**



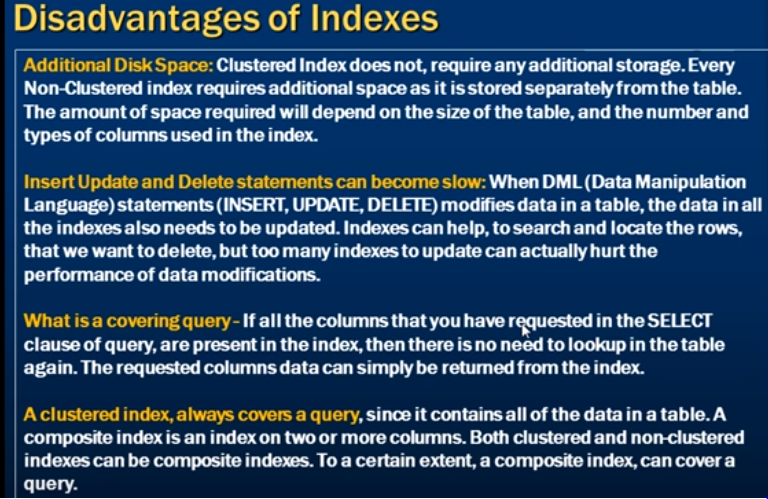




**create index index1 on perma(sal desc)**

**select \* from perma where sal>0->**without index this would give output with sal column in unsorted order .But after creating index the output is sorted

**Q)Clustered and Non Clustered index ,Unique and Non unique index in oracle??**



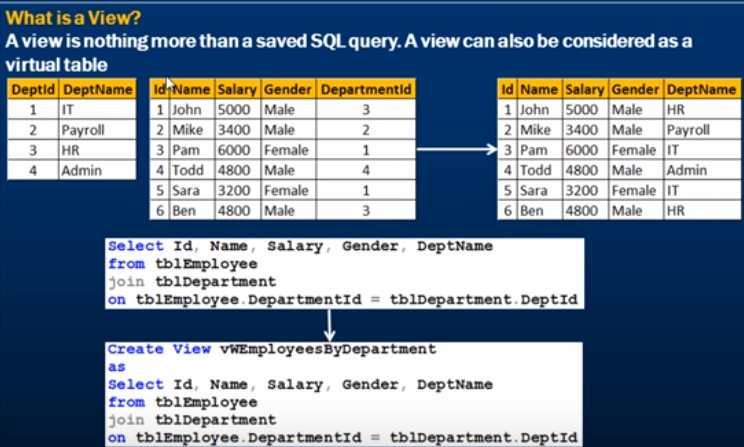
**VIEWS:-**

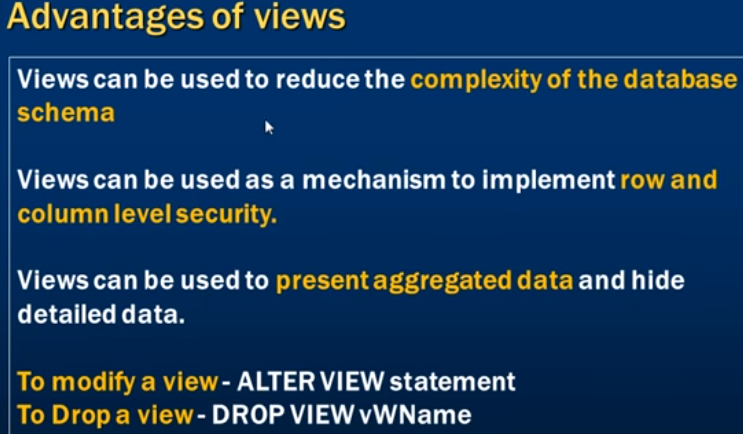
create view myview as

select e.id as empid,e.name as empname,m.name as managername

from employee e

left join employee m on e.managerid=m.id





Count(\*) is faster than count(1)..otherwise results are same

CAST function converts one built-in data type into another built-in data type with the specified precision and length.

