In PLSQL, if you want to return single value as output,then use function

to write functions, it I necessary to assign following variable

set global log_bin_trust_function_creators=1;

1. write a function to generate email, by concatenating first 3 letters of name and last 3 letters of job and concatenate '@mycompany.com'.

```
delimiter //
create function generateEmail(nm varchar(20),
   ejob varchar(20)) returns varchar(30)
begin
 declare vem varchar(30);
 set vem=concat(substr(nm,1,3),'.',
       right(ejob,3),'@mycompany.com');
 return vem;
end//
```

2. calculate experience of a employee

```
delimiter //
create function calcexp(edate date) returns int
begin
 declare vexp int;
 set vexp=timestampdiff(year,edate,curdate());
 return vexp;
end//
delimiter ;
```

exception handling

In exception handling we can write 2 types of handler

1. continue ----it will continue execution of the code after handling the exception.
2. exit ----it will exit and stop the execution of the code after handling the exception.

exception condition

1. SQLEXCEPTION
2. NOT FOUND
3. sql error code

```
delimiter //
create procedure insertdept1(dno int,dnm varchar(30),
dloc varchar(30))
begin
        declare continue handler for 1062
          select "deptno is duplicate pls reenter";
 declare exit handler for SQLEXCEPTION
 select "error occured";
 insert into dept values(dno,dnm,dloc);
 select dno,dnm,dloc;
end//
delimiter ;
```

triggers

When to write the trigger

1. to monitor operation on a table
2. gathering of data for data analysis
3. to manage correctness of denormalize
4. to manage dml operation on some complex views, but the triggers on views are not available in mysql, but available in oracle.

It is task which gets executed automatically.

1. every trigger receives 2 special variables old and new

	old	new
insert	null	new updated values
delete	old existing values from table	null
update	old existing values from table	new updated values in table

```
create table emp_audit(
eno int,
oldename varchar(20),
newename varchar(20),
oldsal float(9,2),
```

```
newsal float(9,2),
uname varchar(20),
changedate date,
action varchar(20));


create trigger insertemptr before insert on emp
for each row
 insert into emp_audit values(new.empno,null,new.ename,null,
 new.sal,user(),curdate(),'insert')

create trigger updateemptr after update on emp
for each row
 insert into emp_audit values(new.empno,old.ename,new.ename,old.sal,
 new.sal,user(),curdate(),'update')



create trigger deleteemptr before delete on emp
for each row
    insert into emp_audit values(old.empno,old.ename,null,old.sal,
null,user(),curdate(),'update')
```

Normalize the data
If the data is normalized then redundancy of data will be as less as possible

accid	custid	cname	mobile	balance	type
100	1001	Kishori	33333	455666	saving
101	1001	Kishori	33333	55555	current
102	1001	Kishori	33333	55666	demat
103	1002	Rajan	44444	4444444	saving
	1003	Revati	555444		

updation anamoly→ if a customer tries to change the mobile number then it may get change only at one place, and will not get changed in all account

insertion anamoly--→if the customer donot open the account, then we will not be able to store the data of probable customers

deletion anamoly--→If any customer has only one account in the bank, and if customer closes the account, then along with account details, we will loose customer details also
account

accid	custid	balance	mobile	type
100	1001	455666	5555	saving
101	1001	55555	5555	current
102	1001	55666	5555	demat
103	1002	4444444	44444	saving

customer

custid	cname	mobile
1001	Kishori	33333

1002	Rajan	44444
1003	Revati	555444

update customer
set mobile=6666
where custid=1001;
old

1001	Kishori	33333

new

1001	Kishori	6666

create trigger before update on customer
for each row
  update account
set mobile=new.mobile
where custid=old.custid

To normalize data

1. the data can be in 1NF,2NF,3NF,BCNF

1NF--- 1 normal form

1. all the values in the table should be atomic
2. each row and each column should have single value

studid	sname	marks	subject
100	Rohit	78,88,99	java,c++,DBMS
101	amit	77,87,97	java,c++,DBMS

since in marks and subject column multiple values are stored, hence the table is not in 1NF

studid	marks	subject
100	78	java
100	88	c++
100	99	DBMS
100	78	java
100	88	c++
100	99	DBMS

studid	sname
100	Rohit
101	amit

since in both tables each single row, single column contains single value so both of them are in 1NF

3. second normal form(2NF)
   The table is in 2NF, if it is in 1NF
   and no partial functional dependency should be there.

prime attribute---- all attributes which are part of candidate key
non prime attribute-----all attribute which are not part of candidate key are called as non prime attributes

studid	courseid	sname	cname	marks
123	100	Rohit	JAVA	89
123	101	Rohit	C++	89
124	100	Ramesh	JAVA	85
124	101	Ramesh	C++	87

It is in 1NF
to check whether it is 2NF
prime attributes--→ stidid, courseid
nonprime attributes-→sname,cname,marks
find functional dependency.
studid--→sname
courseid-→cname
studid+courseid-→marks

if any non prime attribute is dependent on portion of the primary key, then partial functional dependency is there, and hence the table is not in 2NF

studid	sname
123	Rohit
124	Ramesh

courseid	cname
100	JAVA
101	C++

studid	courseid	marks
123	100	89
123	101	89
124	100	85
124	101	87

4. Third normal form(3NF)
   if the table is in 2NF
   if there is no transitive dependency is there, then it is in 3NF

   a-→b-→c then a-→c
   prime attribute-→ non prime attribute--→non prime attribute

accid	custid	cname	mobile	balance	type
100	1001	Kishori	33333	455666	saving
101	1001	Kishori	33333	55555	current
102	1001	Kishori	33333	55666	demat
103	1002	Rajan	44444	4444444	saving

   accid---→custid--→cname
   accid---→custid--→mobile

custid	cname	mobile
1001	Kishori	33333
1002	Rajan	44444

accid	custid	balance	type
100	1001	455666	saving
101	1001	55555	current
102	1001	55666	demat
103	1002	4444444	saving

BCNF (boyce codd normal form)

Boyce-Codd Normal Form (BCNF) is the advance version of the third normal form (3NF) that's why it is also known as a **3.5NF**
According to the E.F. Codd, a relation is in **Boyce-Codd normal form (3NF)** if it satisfies the following conditions:

- A relation is in 3NF.
- And, for every functional dependency, **X → Y,** L.H.S of the functional dependency (X) be the super key of the table.

Example 3:

In this example, we have a relation R with three columns: Id, Subject, and Professor. We have to find the highest normalization form, and also, if it is not in BCNF, we have to decompose it to satisfy the conditions of BCNF.

Id	Subject	Professor
101	Java	Mayank
101	C++	Kartik
102	Java	Sarthak
103	C#	Lakshay
104	Java	Mayank

Interpreting the table:

- One student can enroll in more than one subject.
  - Example: student with **Id 101** has enrolled in **Java and C++**.
- Professor is assigned to the student for a specified subject, and there is always a possibility that there can be multiple professors teaching a particular subject.

Finding the solution:

- Using Id and Subject together, we can find all unique records and also the other columns of the table. Hence, the Id and Subject together form the primary key.
- The table is in 1NF because all the values inside a column are atomic and of the same domain.

- We can't uniquely identify a record solely with the help of either the Id or the Subject name. As there is no partial dependency, the table is also in 2NF.
- There is no transitive dependency because the non-prime attribute i.e., Professor, is not deriving any other non-prime attribute column in the table. Hence, the table is also in 3NF.
- There is a point to be noted that the table is not in **BCNF (Boyce-Codd Normal Form)**.

Why is the table not in BCNF?

As we know that each professor teaches only one subject, but one subject may be taught by multiple professors. This shows that there is a dependency between the subject & the professor, and the subject is always dependent on the professor **(professor -> subject).** As we know that the professor column is a non-prime attribute, while the subject is a prime attribute. This is not allowed in BCNF in DBMS. **For BCNF, the deriving attribute (professor here) must be a prime attribute.**

How to satisfy BCNF?

In Example 3, we will decompose the table into two tables: the Student table and the Professor table to satisfy the conditions of BCNF.

Student Table

P_Id	S_Id	Professor
1	101	Mayank
2	101	Kartik
3	102	Sarthak
4	103	Lakshay
5	104	Mayank

Professor Table

Professor	Subject
Mayank	Java
Kartik	C++
Sarthak	Java
Lakshay	C#

Professor is now the primary key and the prime attribute column, deriving the subject column. **Hence, it is in BCNF.**

In ER diagram 3 types of relations can be there

one-one	If the relation is one-one then any one side key can be added in other

one-Many	if the relation is one to many, then add key of one side to many side table
Many-Many	if relation is many to many then create 3 rd table and add primary key of both sides, and add all relation attributes

Proj Code	Proj Type	Proj Desc	Empno	Ename	Grade	Sal scale	Proj Join Date	Alloc Time
001	APP	LNG	46	JONES	A1	5	12/1/1998	24
001	APP	LNG	92	SMITH	A2	4	2/1/1999	24
001	APP	LNG	96	BLACK	B1	9	2/1/1999	18
004	MAI	SHO	72	JACK	A2	4	2/4/1999	6
004	MAI	SHO	92	SMITH	A2	4	5/5/1999	6

Table is in 1 NF

is it in 2NF

project code---→project type, project description,

empno----→ename,grade, sal scale

project code+empno--→ project join date, alloc time

since partial functional dependency is there so the table is not in 2NF

Proj Code	Proj Type	Proj Desc	Empno	Ename	Grade	Sal scale	Proj Join Date	Alloc Time
001	APP	LNG	46	JONES	A1	5	12/1/1998 24	
001	APP	LNG	92	SMITH	A2	4	2/1/1999	24
001	APP	LNG	96	BLACK	B1	9	2/1/1999	18
004	MAI	SHO	72	JACK	A2	4	2/4/1999	6

project

Proj Code	Proj Type	Proj Desc
001	APP	LNG
004	MAI	SHO

Empno	Ename	Grade	Sal scale

46	JONES	A1	5
92	SMITH	A2	4
96	BLACK	B1	9
72	JACK	A2	4

empno--→grade-→sal scale

proj_emp

Proj Code	Empno	Proj Join Date	Alloc Time
001	46	12/1/1998 24	
001	92	2/1/1999	24
001	96	2/1/1999	18
004	72	2/4/1999	6

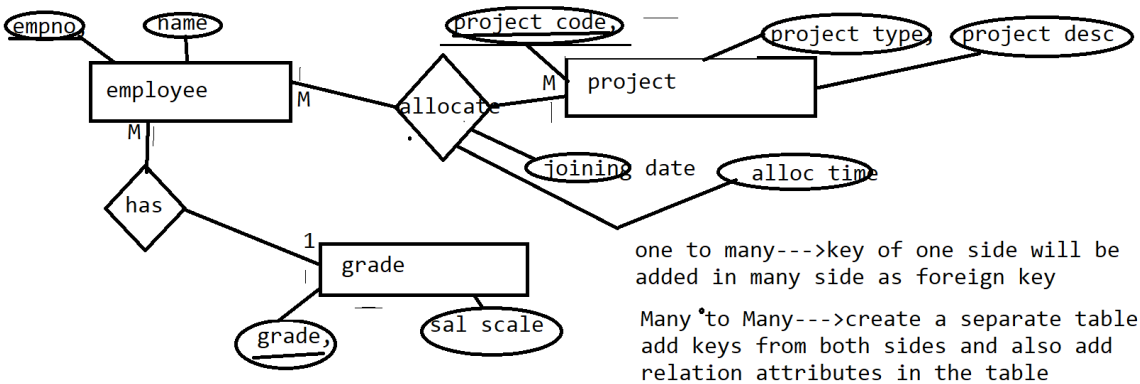
Is it in 3 NF

Employee

Empno	Ename	Grade
46	JONES	A1
92	SMITH	A2
96	BLACK	B1
72	JACK	A2

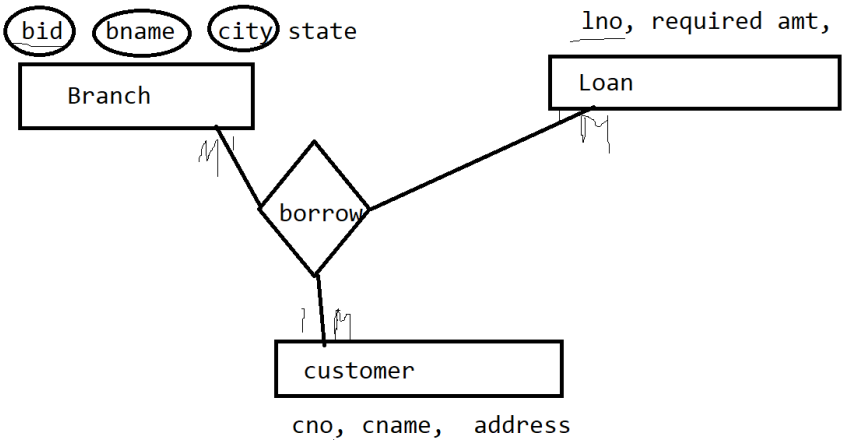
grade

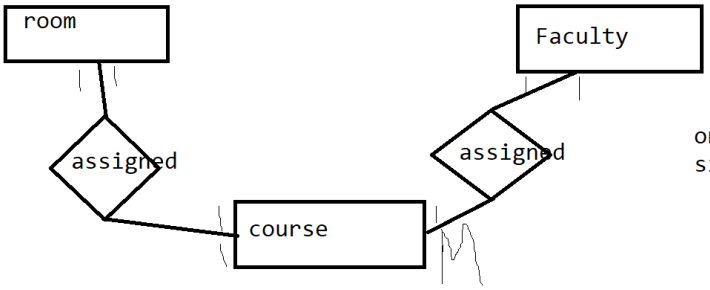
Grade	Sal scale
A1	5
A2	4
B1	9



one to many--->key of one side will be added in many side as foreign key

Many to Many--->create a separate table, add keys from both sides and also add relation attributes in the table





one to one ---> then add any one side key to another side

- **Orderno**
- **Orderdate**
- **Itemno**
- **Qty**
- **Price**
- **Cname**
- **Custno**
- **Email**
- **Orderamt**
- **Salespersonno**
- **Salespersonname**
- **Locationid** ----------**location from where item dispatched**
- **Location name**

One customer can place many order
One order contains many items
One order will be managed by one salesperson
One order belong to one customer
One order can be dispatched from different location

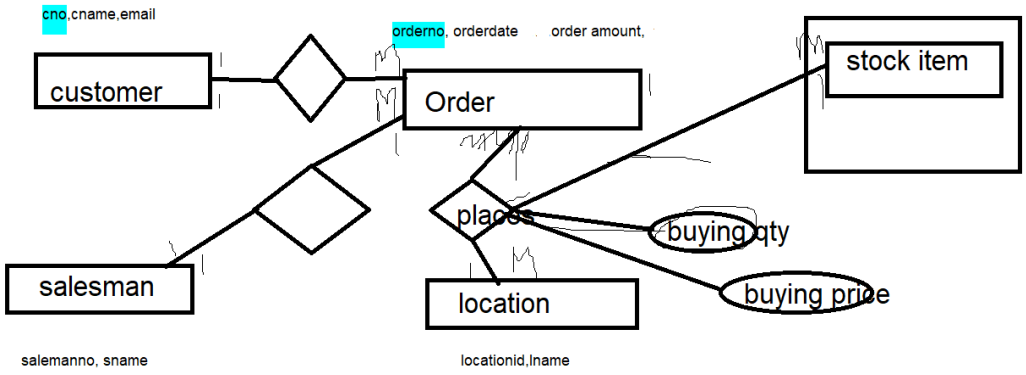
stock item table --→ item no, stock qty, stock price,date,

the given table is in 1NF
Is it in 2 NF
orderno---→orderdate,cname,cno,email,order amount,salesno,name
itemno
orderno+itemno->qty,price,locationid,lname

order
orderno, orderdate,cname,cno,email,order amount,salesno,sname
order-item
orderno,itemno,qty,price,locationid,lname
orderno-→locationid-→lname
These are no in 3NF, because we have transitive dependencies
orderno-→ cno-→cname, email
order no-→ salesno-→sname

order
orderno, orderdate,cno,order amount,salesno
customer
cno,cname,email
saleman
salemanno, sname

location
locationid,lname
order-item
orderno,itemno,qty,price,locationid



Movie ticket booking
movieid, mname, show time,duration, date,
price,capacity,seatno,screeno,screen_name, cname, cid, caddress, age, qty, mobile

screen no	seatno	date	show-time			
1	1		morning			
1	2					
1	3					

it is in 1 NF
Is it in 2NF
screenno--→capacity, screen_name
date
showtime
seatno
screenno+date
screen num+showtime
screen num+seatno
date+showtime
date+seatno
screennum+showtime+date-→movieid,mname,duration,price
screenum+showtime+date+seatno-→ cname, cid, caddress, age,qty,mobile

screen
screenno,capacity, screen_name

screen_movie
screennum+showtime+date,movieid,mname,duration,price

screen-cust
screenum+showtime+date+seatno,cname, cid, caddress, age,qty,mobile

to check it whether it is in 3 NF
screen_movie
screennum+showtime+date,movieid,mname,duration,price
screennum+showtime+date-→movieid-→ mname,duration,price

screen_movie
screennum+showtime+date,movieid,

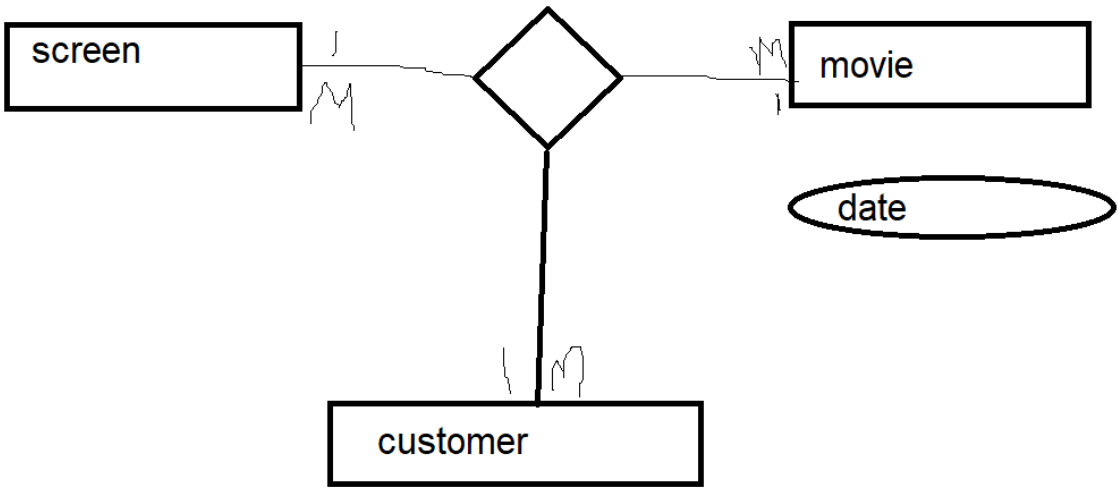
movie
movieid,mname,duration,price

screen-cust
screenum+showtime+date+seatno,cname, cid, caddress, age,qty,mobile

screenum+showtime+date+seatno-→cid-→cname, caddress, age ,mobile

screen-cust
screenum+showtime+date+seatno, cid, qty

customer
cid,cname, caddress, age ,mobile


screen
movie
date
customer