Triggers

Triggers are block of code which gets executed automatically when some events occur.

When to use triggers

1. Triggers are used to monitor table for DML operation
2. Also can be used to collect data for data analysis
3. Can be used to manage data which is denormalized

Types of triggers

1. Database level triggers(These triggers are available only in oracle and not in mysql)
   After startup
   Before shutdown
2. DDL/statement level triggers (These triggers are available only in oracle and not in mysql)
   a. Before create table
   b. After drop table
3. Row level triggers
   These triggers will get executed once for every row that gets affected by any DML operation(insert, update , delete)

1. Write a trigger to keep track of insertion and updating or deleting in product table
   Step 1:
   Create product_data table
   Create table product_data(pid int,
   Pname varchar(20),
   Oprice decimal(9,2),
   Nprice decimal(9,2),
   Uname varchar(20),
   Changedate date,
   Action varchar(20));

   Step 2
   Create trigger ins_prod
   Before insert on product
   For each row
   Begin
   Insert into product_data values
   (NEW.pid,NEW.pname,null,NEW.price,current_user(),curdate(),'before insert');
   End//

   Step 3--- to execute trigger
   Insert into product values(10,'pringles',34,120);

   To write a trigger for update
   Create trigger update_prod
   Before update

```
On product
For each row
Begin
   Insert into product_data
values(old.pid,old.pname,old.price,new.price,current_user(),curdate(),'before update');
End//
```

2. Create delete trigger

```
Create trigger del_prod before delete
On product for each row
Begin
 Insert into product_data
values(old.pid,old.pname,old.price,null,current_user(),curdate(),'before delete')
End;
```

Create table emp_audit to store information about DML operations on emp table for monitoring changes in the emp table data

```
Create table emp_audit(
Empno int,
ename varchar(20),
osal decimal(9,2),
nsal decimal(9,2),
ojob varchar(30),njob varchar(30),uname varchar(20),changedate date,action varchar(20))

-----for insert
create trigger ins_emp
after insert on emp
for each row
begin
   insert into emp_audit
values(new.empno,new.ename,null,new.sal,null,new.job,current_user(),curdate(),'after
insert')
end//


----- for update
Create trigger up_emp
After update on emp
For each row
Begin
   Insert into emp_audit
values(old.empno,old.ename,old.sal,new.sal,old.job,new.job,current_user(),curdate(),'after
update');
End//

---for delete
Create trigger del_emp
```

After delete on emp
For each row
Begin
  Insert into emp_audit
values(old.empno,old.ename,old.sal,null,old.job,null,current_user(),curdate(),'after delete');
End//

Account

a/c no	Custno	Cname	Mobile	Email	Type	balance
100	1	Kishori	34445	dff@ffnj	demat	50000
101	1	Kishori	444444	dff@ffnj	saving	60000
102	1	Kishori	444444	dff@ffnj	current	60000
103	2	Rajan	3453	Asda@asd	saving	60000
	3	Revati	65867	asdas@lsdj		-----Not possible

If the data is not normalized, then in the data lot of redundancy is there. Hence it causes following problem

1. Insertion anomaly--- in the given table, I cannot keep the information of customer, who do not open the a/c, is called as insertion anamaly.
2. Updation anomaly- if the redundant data changes at one place we may not guarantee that changes will be up to date at all places, is called as updation anamoly.
3. Deletion anomaly---if any customer closes the a/c, and if that is the only a/c then along with account information, we may lose customer information also, is clled as deleteion anamaly.

account

a/c no	Custno	Type	balance	Mobile
100	1	demat	50000	44444
101	1	saving	60000	44444
102	1	current	60000	44444
103	2	saving	60000	3453

Customer

Custno	Cname	Mobile	Email
1	Kishori	5555	dff@ffnj
2	Rajan	3453	Asda@asd
3	Revati	65867	asdas@lsdj

```
Create trigger up_cust
Before update on customer
For each row
Begin
    Update account
    Set mobile=new.mobile
     Where custno=old.custno;
Insert into cust_audit
values(old.custno,old.cname,old.mobile,new.mob,curdate(),current_user())
End//
```

To find 4 th topmost sal in emp table

```
select *
from emp e
where 3=(select count(distinct sal)
    from emp m
where m.sal>e.sal)
```

To find 4 topmost sal in emp table

```
select *
from emp e
where 4>select count(distinct sal)
    from emp m
where m.sal>e.sal)
```

Normalization

If the data is not normalized, then in the data lot of redundancy is there. Hence it causes following problem

4. Insertion anomaly--- in the given table, I cannot keep the information of customer, who do not open the a/c, is called as insertion anamaly.
5. Updation anomaly- if the redundant data changes at one place we may not guarantee that changes will be up to date at all places, is called as updation anamoly.
6. Deletion anomaly---if any customer closes the a/c, and if that is the only a/c then along with account information, we may lose customer information also, is clled as deleteion anamaly.

Types of normalization ----- 1NF, 2NF, 3Nf, BCNF(3.5NF)

First normal form (1NF)

As per the rule of first normal form, an attribute (column) of a table cannot hold multiple values. It should hold only atomic values.

Stud no	Sname	marks
1	Rajas	56,78,67
2	Rajat	45,56
3	rohit	45

Since every row does not contain atomic value so it is not in 1NF

Stud no	Sname	marks
1	Rajas	56
1	Rajas	78
1	Rajas	67
2	Rajat	45
2	Rajat	56
3	rohit	45

Second normal form (2NF)

A table is said to be in 2NF if both the following conditions hold:

- Table is in 1NF (First normal form)
- No non-prime attribute is dependent on the proper subset of any candidate key of table.

  Step1

Check whether table is in 1NF

Step 2:

Find out primary key, and if its composite key then only checks for 2NF

All the attributes which are part of primary key are called as prime attributes, and remain attributes are called as non-prime attribute

The fees is dependent on teacher and the subject

Duration is dependent on subject

teacher_id	subject	teacher_age	fees	duration
111	Maths	38	25000	20 days
111	Physics	38	30000	30 days
222	Biology	38	50000	40 days
333	Physics	40	20000	30 days
333	Chemistry	40	35000	50 days

There should not be any partial functional dependency

Primary key ---- teacherid+subject

Prime attribute

Teacherid---→ teacher age

Subject--→duration

Teacherid+subject---→ fees

teacher_id	subject	fees
111	Maths	25000
111	Physics	30000
222	Biology	50000
333	Physics	20000
333	Chemistry	35000

teacher_id	teacher_age
111	38

222	38
333	40

subject	duration
Maths	20 days
Biology	40 days
Physics	30 days
Chemistry	50 days

Third Normal form (3NF)

A table design is said to be in 3NF if both the following conditions hold:

- Table must be in 2NF

- Transitive functional dependency of non-prime attribute on any super key should be removed.

a-→b b-→c so a-→c

the following table are in 3NF

a/c no	Custno	Type	balance
100	1	demat	50000
101	1	saving	60000
102	1	current	60000
103	2	saving	60000

Transitive dependency is there so the table is not in 3NF

a/cno---→ custno---→ cname, mobile, email

Custno	Cname	Mobile	Email
1	Kishori	444444	dff@ffnj
2	Rajan	3453	Asda@asd

Normalize the given table:

One employee can work on many projects.

Proj Code	Proj Type	Proj Desc	Empno	Ename	Grade	Sal scale	Proj Join Date Time	Time allocated	
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	
005	MAI111	SHOrty	92	SMITH	A2	4	5/5/1999	6	