Tiggers

Triggers are block of code which gets executed automatically.

Trigger can be written on all DML operation in mysql

In mysql only row level triggers are allowed. Statement level triggers are not allowed in mysql.

If we want to write the trigger, then the
first step is to decide which monitoring table you want to create, and what fields will be stored in the table.

Step2 – what will be the trigger timing.

Tigger timing can either before or after

The timing can be insteadof --→ these triggers are not supported by mysql. But are supported by oracle, these triggers are only used with views.

Step3 ---- on which action the trigger can be executed.

Action can be ----→insert, update or delete

Step 4------ decide which table the trigger should monitor

Step 5-----decide trigger level

The trigger level can be row level or statement level

Statement level----→ for each DML statement only one entry will be there

Example: if a delete statement, deletes 10 rows , then we need only one entry in the table, then write statement level trigger

These triggers are not supported by mysql , but are supported by oracle.

Row level trigger--→ for each DML statement one entry will be there, for each row getting affected by the dml statement.

Example: if a delete statement, deletes 10 rows , then we need 10 entries in the table, then write row level trigger

Syntax :

```
Create trigger <name> {before|after} {action} on <table name>

For each row

Begin

   statements

End//
```

Where to use triggers:

1. For monitoring changes happening in the table by all users
2. To manage data in complex view
3. To maintain integrity of denormalized data

In trigger there are 2 special variables NEW and OLD

	OLD	NEW
insert	null	Will have row that will be created after insertion is done
delete	Will have row that exists in the table	Null
Update	Will have row that exists in the table	Will have row that will be created after changes are applied

To access data from OLD and NEW variables
Example OLD.empno, OLD.ename
NEW.sal

In mysql to get the name of current user the function is user()

```
Create table empsecurity(

Empno int,

Ename varchar(20),

Action varchar(20),

Oldsal double(9,2),

Newsal double(9,2),

Uname varchar(20),

Act_date date);
```

1. If any user insert data in emp table then insert a record in empsecurity table.

```
Delimiter //

Create trigger insertemp after insert on emp

For each row

Begin

  Insert into empsecurity values(NEW.empno,NEW.ename,'insert',null,NEW.sal,user(),curdate());

End//

Delimiter ;
```

1. If any user delete data in emp table then insert a record in empsecurity table.

```
Delimiter //
Create trigger deleterec after delete on emp
For each row
Begin
  Insert into empsecurity values(OLD.empno,OLD.ename,’delete’,OLD.sal,null,user(),curdate());
End//
Delimiter ;
```

```
Create table discounts(
Pid int,
 Disc_percent int)
```

```
  Create table product_dis(
 Pid int,
Pname varchar(20),
Price double(9,2),
Discounted_amt double(9,2))
```

discounts

Pid	Disc_percent
1	3
2	20

product_dis

pid	pname	price	Discounted_amt
1	chair	2000	1940

Write a trigger to update disounted _amt in product_dis table , as soon as we change Disc_percent in discounts table

```
Update discounts
Set  Disc_percent =7
Where pid=1;
```

	pid	Disc_percent
old	1	3
new	1	7

```
Create trigger update_discount after update on discounts

For each row

Begin

Update product_dis

Set Discounted_amt=price-price*(NEW. Disc_percent  /100) ,pid=NEW.pid

Where pid=OLD.pid;

 End //
```

Exception handling

Any error that occurs at run time, because user has entered wrong data, and these errors can be handled programmatically, then it is called as exceptions, otherwise it is called as errors.

1. Types of exception in mysql
   SQLEXCETION
   SQLSTATE 23000
   Error code
   NOT FOUND

2. Handlers
   Continue--- continue handler will handle the error and resume the execution of the procedure
   Exit--- exit handler will handle the error and stop the execution of the procedure

   To declare handler
   ```
   Declare (exit|continue) handler for SQLEXCEPTION select ‘error occurred’;
   Declare (exit|continue) handler for SQLEXCEPTION begin
   select ‘error occurred’;
   set finished=1;
   rollback;
   end;
   ```

   1. Write a procedure to insert a record in dept, if department number is duplicate then show message error occurred.

```
Delimiter //

Create procedure insertdept(in pdno int,in pdnm varchar(20),in ploc varchar(20))

Begin

   Declare exit handler for SQLEXCEPTION select “error occurred”;

   Insert into dept values(pdno,pdnm,ploc);

  Select * from dept;
```

```
End//

Delimiter ;
```

2. Create procedure to insert record in product table, if any error occurred because of duplicate pid show error message duplicate key, if error occurred because of -ve qty or -ve price then show error message values cannot be -ve, otherwise show error message error occurred.

```
Delimiter //

Create procedure insertproduct(in ppid int, pnm varchar(20), pqty int,pprice
double(9,2),pcid int)

Begin

    Declare continue handler for 1062 select 'duplicate key' msg;

      Declare exit handler for 3819 select 'value should be > 0' msg;

    Declare continue handler for SQLEXCEPTION select 'error occurred' msg;

    Insert into product values(ppid,pnm,pqty,pprice,pcid);

     Select * from product;

End//
```

Normalization

For proper data modelling we use rules of normalization and ER diagram

Acid	cid	Cname	address	balance	type	Relmgrid	relmgrname
1	100	Kishori	Aundh	4567	saving	120	ANIL
2	100	Kishori	Baner	5555	curret	120	ANIL
3	100	Kishori	Baner	6666	demat	120	ANIL
5	101	Rajan	Baner	7777	saving	121	Bhavika
null	102	Atharva	Aundh			122	Revati

Insertion anamoly—

In the above table primary key is acid.

Hence if any new relation manager joins the bank, then unless I assign any account to the manager, we will not be able to add the record

If any customer comes for enquiry, and does not open account still I will not be able to add customer details in the table

This problem is called as insertion anamoly

Updation anamoly

In above table if Kishori submits the request for change in the address with a/c no 1, then the change may happen only in one account,and may keep old address for account 2 and 3, which creates a problem.

This is called as updation anamoly.

Deletion anamoly:

In the above table if Rajan closes the a/c, so we will delete the record from the table,

Along with that record we will loose the customer information, and also looe information of relation manager Bhavika.

This is called as deletion anamoly.

To remove these problems we need to divide the table into multiple tables

Acid	cid	balance	type
1	100	4567	saving
2	100	5555	curret
3	100	6666	demat
5	101	7777	saving

cid	Cname	address	Relmgrid
100	Kishori	baner	120
101	Rajan	Baner	121
102	Atharva	Aundh	122

Relmgrid	relmgrname
120	ANIL
121	Bhavika
122	Revati

Rules of normalization

1NF

According to the E.F. Codd, a relation will be in 1NF, if each cell of a relation contains only an atomic value.

studid	name	marks	course
1	Revati	99,89,96	Java,.net,c++
2	Ashu	88,89,95	Java,.net,c++

Since marks column contains multiple values so the given table is not in 1NF

studid	name	course	marks
1	Revati	java	99
1	Revati	.net	89

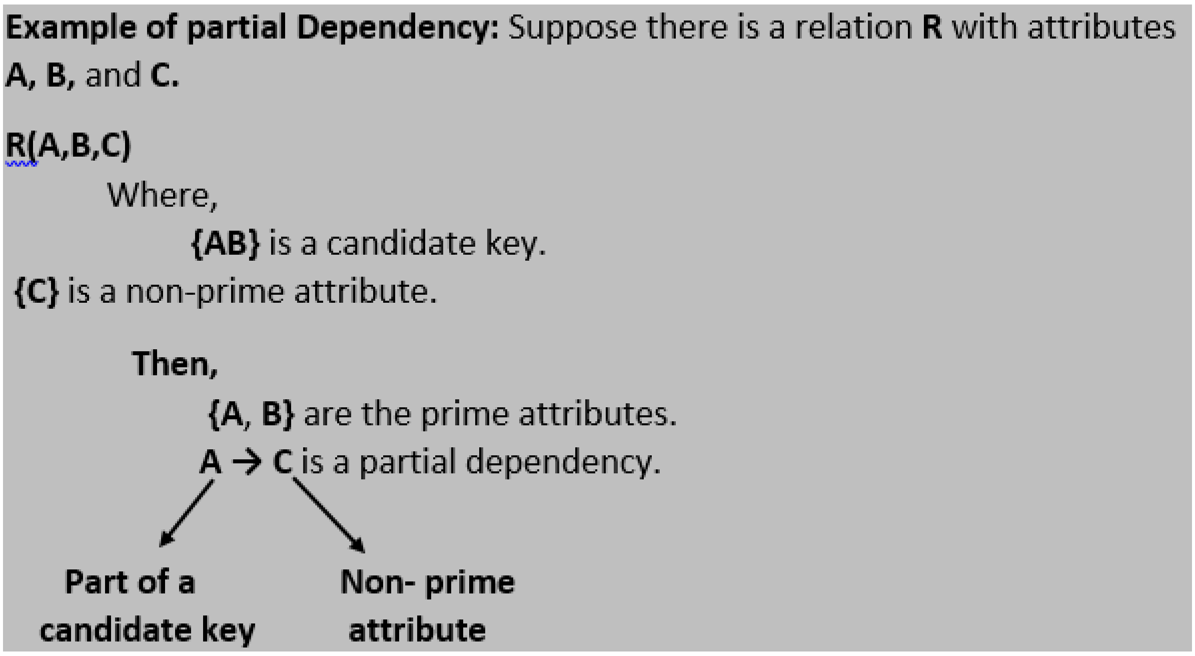
1	Revati	C++	96
2	Ashu	java	88
2	Ashu	.net	89
2	Ashu	C++	95

2NF

According to the E.F. Codd, a relation is in **2NF,** if it satisfies the following conditions:

- A relation must be in 1NF.
- And the candidate key in a relation should determine all non-prime attributes or no partial dependency should exist in the relation.
  Prime attribute: the fields which are part of candidate key(Primary key) are called as prime attribute
  Non prime attributes: the fields which are not part of candidate key(Primary key) are called as non prime attribute

Partial Dependency: If a non-prime attribute can be determined by the part of the candidate key in a relation, it is known as a partial dependency. Or we can say that, if L.H.S is the proper subset of a candidate key and R.H.S is the non-prime attribute, then it shows a partial dependency.
Example of partial Dependency: Suppose there is a relation **R** with attributes **A, B,** and **C.**



Example of partial Dependency: Suppose there is a relation **R** with attributes **A, B,** and **C.**

R(A,B,C)

Where,

{AB} is a candidate key.

{C} is a non-prime attribute.

Then,

{A, B} are the prime attributes.

A → C is a partial dependency.

Part of a candidate key
Non- prime attribute

studid	name	course	marks
1	Revati	java	99
1	Revati	.net	89
1	Revati	C++	96
2	Ashu	java	88

2	Ashu	.net	89
2	Ashu	C++	95

Primary key -→{studid,course}

Prime attributes: studid,course

Non prime attributes: name, marks

Studid-->name

Course-->

Studid+course→marks

studid	course	marks
1	java	99
1	.net	89
1	C++	96
2	java	88
2	.net	89
2	C++	95

Student

studid	name
1	Revati
2	Ashu

3NF

According to the E.F. Codd, a relation is in **third normal form (3NF)** if it satisfies the following conditions:

- A relation must be in second normal form (2NF).
- And there should be no transitive functional dependency exists for non-prime attributes in a relation.
- Third Normal Form is used to achieve data integrity and reduce the duplication of data.

A relation is in 3NF if and only if any one of the following conditions will satisfy for each non-trivial functional dependency **X→ Y:**

1. X is a super key or candidate key
2. And, Y is a prime attribute, i.e., Y is a part of candidate key.

Transitive Dependency: If **X** → **Y and Y**→ **Z** are two functional dependencies, **X** → **Z** is called as a transitive functional dependency.

Acid	cid	Cname	address	balance	type	Relmgrid	relmgrname
1	100	Kishori	Baner	4567	saving	120	ANIL
2	100	Kishori	Baner	5555	curret	120	ANIL
3	100	Kishori	Baner	6666	demat	120	ANIL
5	101	Rajan	Baner	7777	saving	121	Bhavika

a/c id---→ cid--→cname, address,relationmgrid,relname

the table is not in 3NF

cid	Cname	address	Rel id
100	Kishori	Baner	120
101	Rajan	Baner	121

Relmgrid	relmgrname
120	ANIL
121	Bhavika

Acid	cid	balance	type
1	100	4567	saving
2	100	5555	curret
3	100	6666	demat
5	101	7777	saving

BCNF(3.5 NF)

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.

Normalize the following tables upto 3 NF form

Proj	Proj	Proj	Empno	Ename	Grade	Sal	Proj	Alloc
Code	Type	Desc				scale	Join Date	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

- 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