Create table emp

(

Empno char(4) **primary key**,

Ename varchar(25),

Sal float,

Deptno int,

Primary key(deptno, empno) //(parent,child)

);

* If you drop the constraint, then the index is dropped automatically

Alter table emp drop primary key;

To add primary key afterwards to an existing table:

Alter table emp add primary key(deptno, empno);

To change the primary key column:-

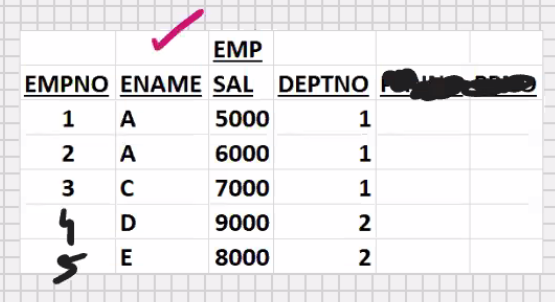
1. Drop the original primary key constraint
2. Add a new primary key

**Constrains are of 2 types:-**

a.column level constraint (specified on single column)

b.table level constraint (composite) (specified on combination of 2 or more columns) (has to be specified at the end of the table structure)

**NOT NULL CONSTRAINT**

* Null values are not allowed (it’s a mandatory column) (similar to primary key)
* ****Duplicate values are allowed (unlike primary key)
* Can have any number of not null constraints per table (unlike primary key)
* You cannot have composite not null constraint (unlike primary key)
* Always a column level constraint (cannot have table level not null constraint)

Create table emp

(

Empno char(4) **primary key**,

Ename varchar(25) not null,

Sal float not null,

Deptno int

);

* In MySQL, nullability is a part of the datatype

**To find out which are the not null columns:-**

Desc emp;

**To remove the not null constraint afterwards:-**

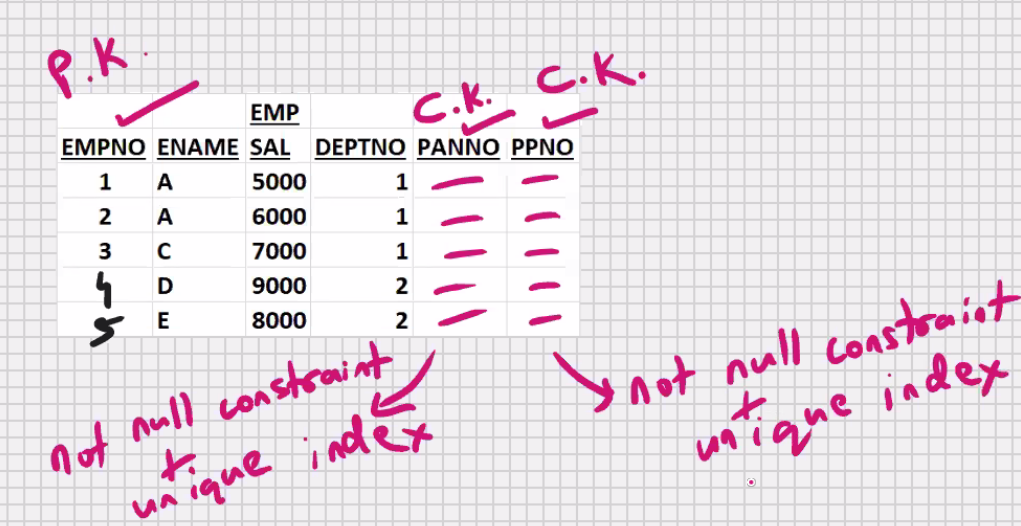
Alter table emp modify ename varchar(25) null;

**To add the not null constraint afterwards for an existing table:-**

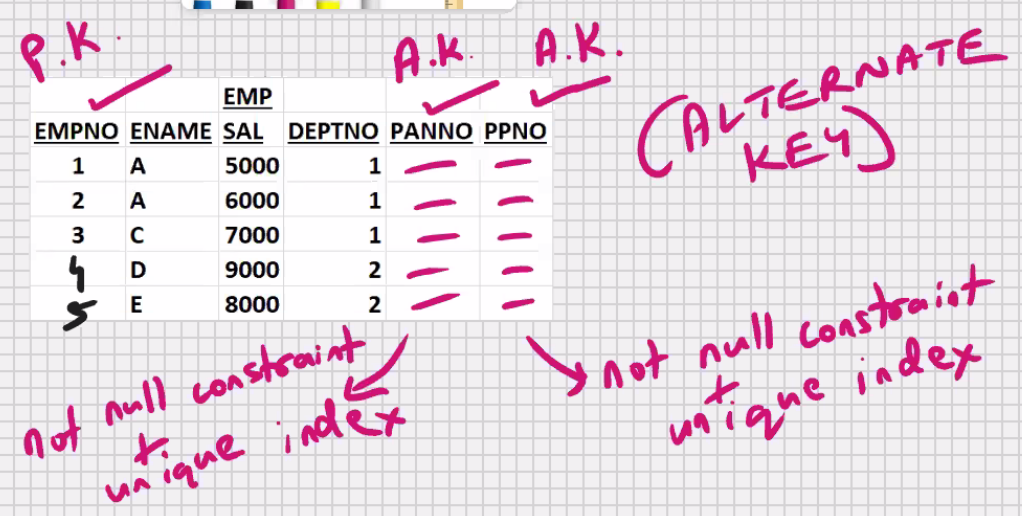
Alter table emp modify ename varchar(25) not null;

**Solution for candidate key column:-**

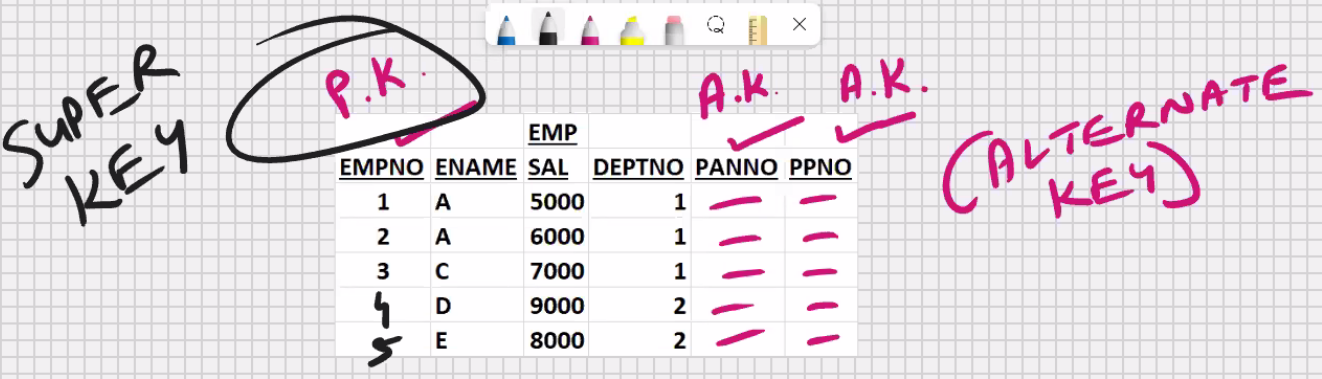
Not null constraint + unique index



* With the help of above , indirectly, you can have multiple primary keys in the table
* Alternate key is not a constraint

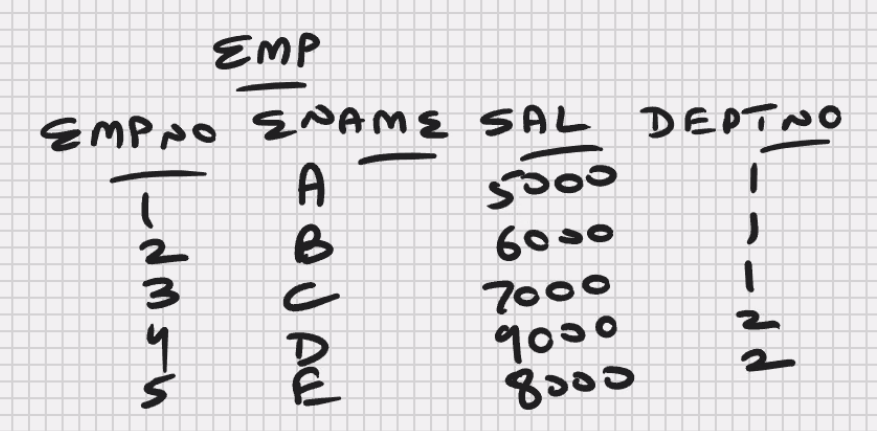


* Alternate key is a definition
* Alternate key for the candidate key column, if you specify a not null constraint and create an unique index, then it becomes an alternative to primary key, then such a candidate key column is known as **alternate key**
* **Super key :** is not a constraint
* **Super key** is a definition
* Super key if you have an alternate key in the table, then primary key is known as super key



**Unique constraint**

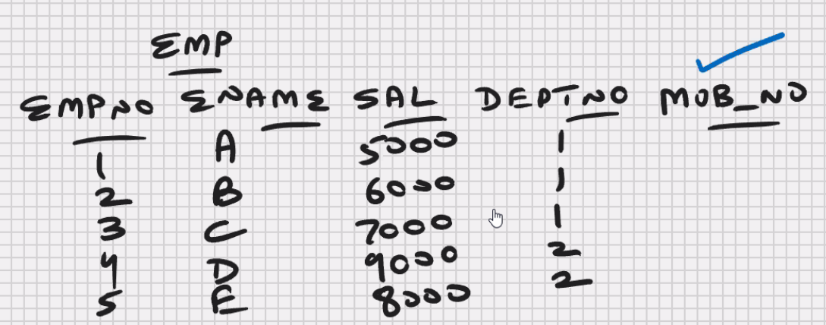
**EMP table**

****

* Duplicate values are not allowed (it has to be unique)

(similar to primary key)

* Null values are allowed (unlike key)
* You can enter any number of null values
* Unique index is automatically created (similar to primary key)
* Text and blob cannot be unique (similar to primary key)



* In MySQL, you can combine upto 32 columns in a composite unique
* **You can have any number of unique constraints per table (unlike primary key)**

Create table emp

(

Empno char(4) **,**

Ename varchar(25),

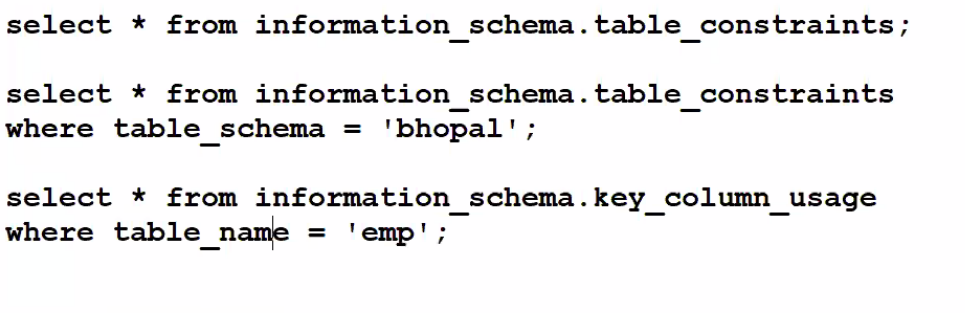
Sal float,

Deptno int,

Mob\_no char(15) unique, -> column level constraint

Unique (deptno, empno) -> table level constriant

);

****

* Unique index automatically created (similar to primary key)

Show indexes from emp;

* Unique constraint is also an index so to drop it:-

Drop index mob\_no on emp;

Drop index deptno on emp;

* To add unique constraint afterwards:-

Alter table emp

Add constraint u\_emp\_mob\_no unique (mob\_no);

Constraint u\_emp\_mob\_no -> optional

Select \* from information\_schema.table\_constraints

Where table\_schema = ‘bhopal’;

Create table emp

(

Empno char(4) **,**

Ename varchar(25),

Sal float,

Deptno int,

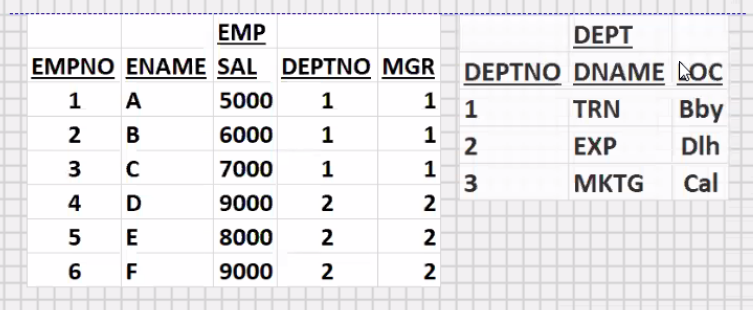
Mob\_no char(15) ,

Unique (deptno, empno) ,

Unique (mob\_no)

);

* Column level constraint can be specified at table level (at the end of structure), but a table level composite at table level (at the end of structure), except for the not null constraint which is always a column levelconstraint, and therefore the syntax will not support from specifying it at table level

**FOREIGN KEY**



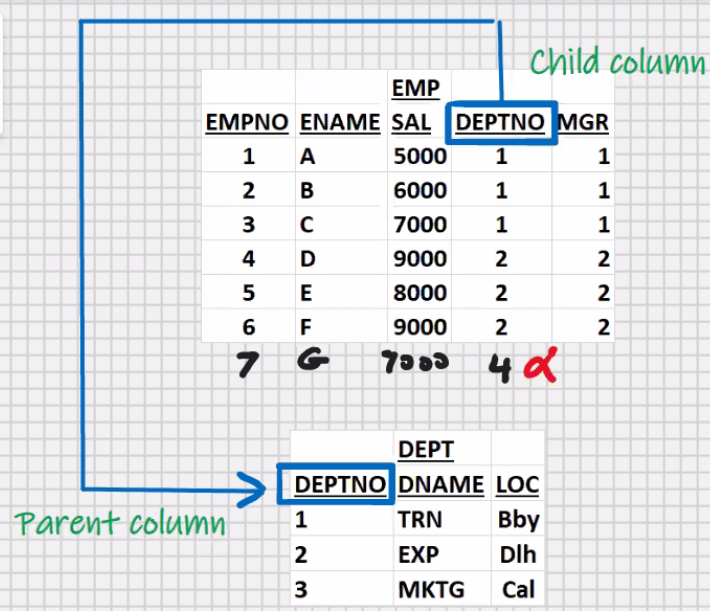
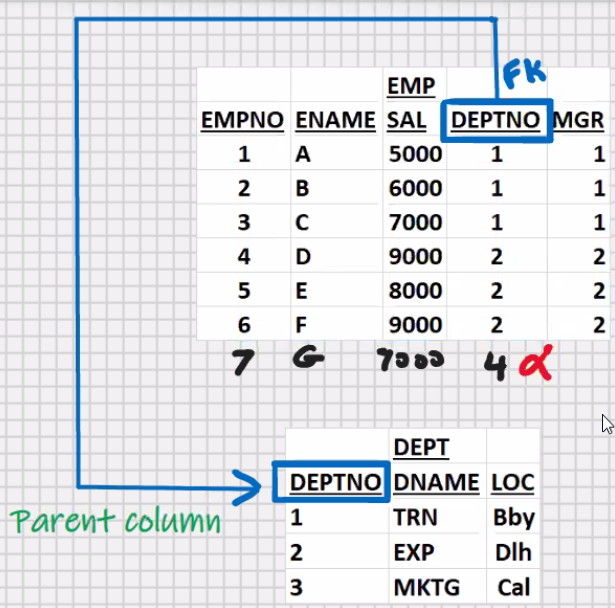
Foreign key is also known as foreign column

* Column or set of columns that

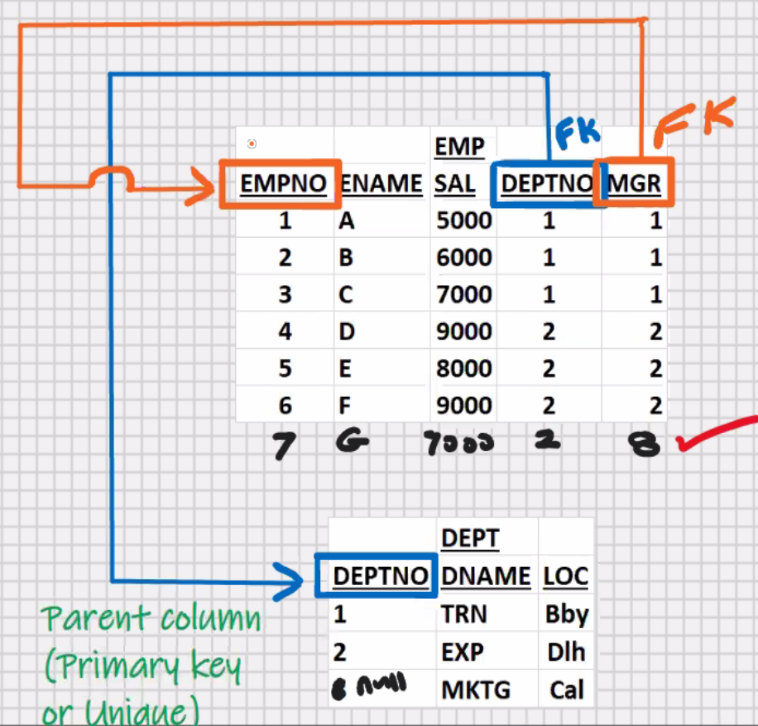
References a column or set of columns

of some table

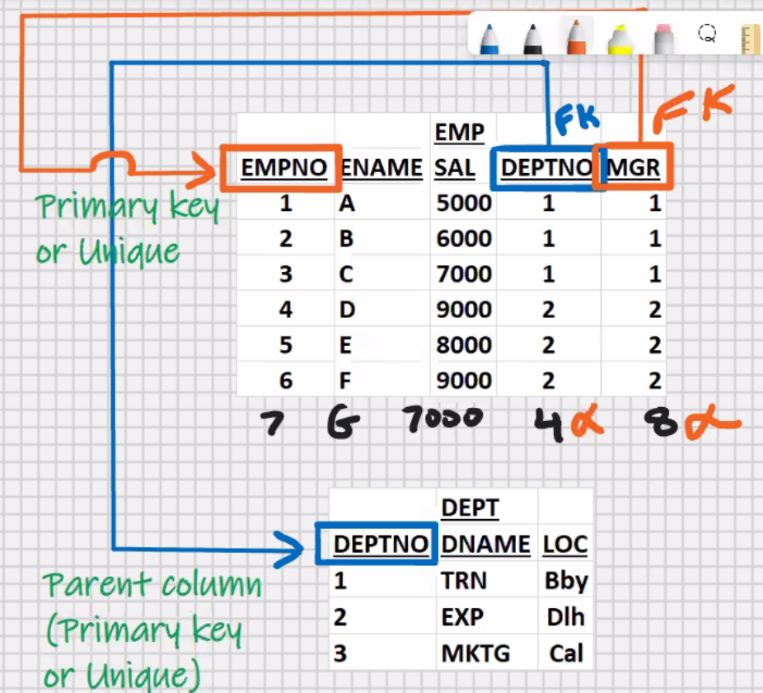
* foreign key constraint is not required to write a join
* foreign key constraint is specified on the child column (not the parent column)



* parent column has to be primary key or unique (this is a prerequisite for foreign key)
* foreign key will allow duplicate values (unless specified otherwise)
* foreign key will allow null values also (unless specified otherwise)



* foreign key may reference column of same table also (known as self-referencing)

create table dept

(

Deptno int primary key,

Dname varchar(15),

Loc varchar(10)

);

Create table emp

(empno char(4) primary key,

Ename varchar(25),

Sal float,

Deptno int,

Mgr char(4),

Constraint fk\_emp\_deptno foreign key(deptno)

References dept(deptno),

Constraint fk\_emp\_mgr foreign key(mgr)

References emp (empno)

);

Constraint fk\_emp\_deptno -> optional

Constraint fk\_emp\_mgr -> optional

Select \* from information\_schema.table\_constraints;

Select \* from information\_schema.table\_constraints

Where table\_schema = ‘bhopal’;

Select \* from information\_schema.key\_column\_usage

Where table\_name = ‘emp’;

Alter table emp drop foreign key fk\_emp\_deptno;

**To add the constraint afterwards:**

Alter table emp add constraint fk\_emp\_deptno foreign key(deptno) references dept(deptno);

Delete from dept where deptno = 3;

* you can delete the parent row provided child rows don’t exist

Delete from dept where deptno = 2;

* you cannot delete the parent row when child rows exist

but still if you want to delete then solution is:

1.delete from emp where deptno = 2;

2. delete from dept where deptno = 2;

Create table emp

(empno char(4) primary key,

Ename varchar(25),

Sal float,

Deptno int,

Mgr char(4),

Constraint fk\_emp\_deptno foreign key(deptno)

References dept(deptno) **on delete cascade**,

Constraint fk\_emp\_mgr foreign key(mgr)

References emp (empno)

);

**On delete cascade** -> if you delete the parent row, then MySQL will automatically delete the child rows also

**To preserve the child rows :**

Update emp set deptno = null where deptno = 2;

Delete from dept where deptno = 2;

* you can update the parent column provided child rows don’t exist

Update dept set deptno = 4 where deptno = 3;

* you cannot update the parent column when child column is exist

update dept set deptno = 4 where deptno = 2; -> error

Create table emp

(empno char(4) primary key,

Ename varchar(25),

Sal float,

Deptno int,

Mgr char(4),

Constraint fk\_emp\_deptno foreign key(deptno)

References dept(deptno) **on delete cascade** **on update cascade**,

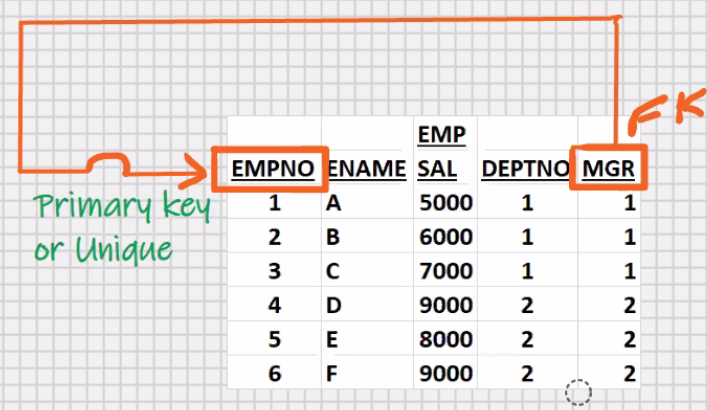
Constraint fk\_emp\_mgr foreign key(mgr)

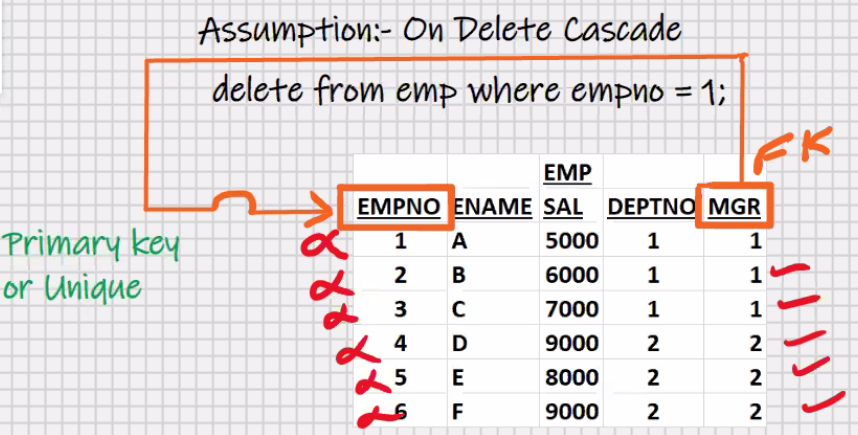
References emp (empno)

);

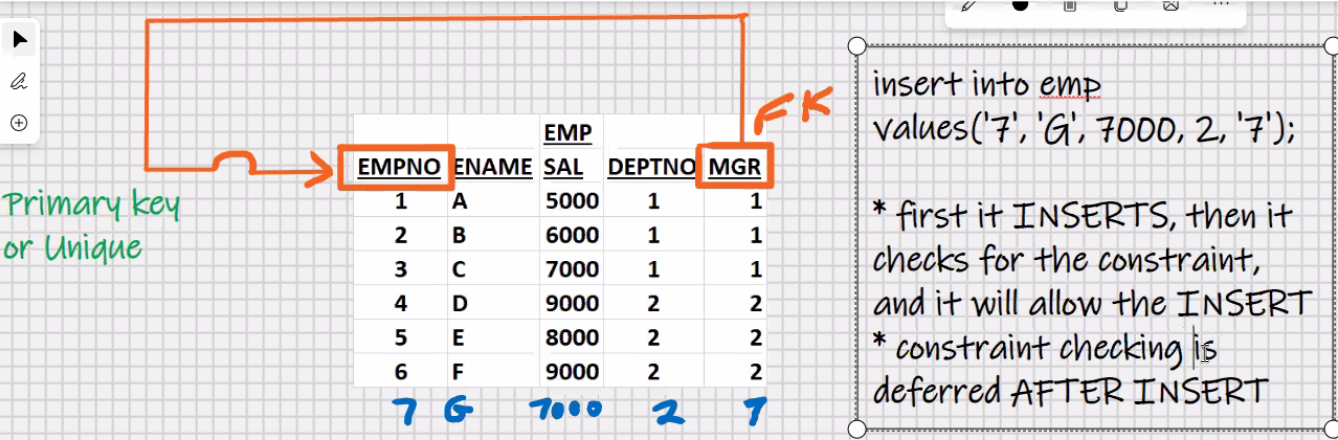
**On update cascade** -> if you update the parent column, then the child rows are updated automatically

Update dept set deptno = 4 where deptno = 2;





* avoid on delete cascade in the event of self referencing; you may delete more rows than expected

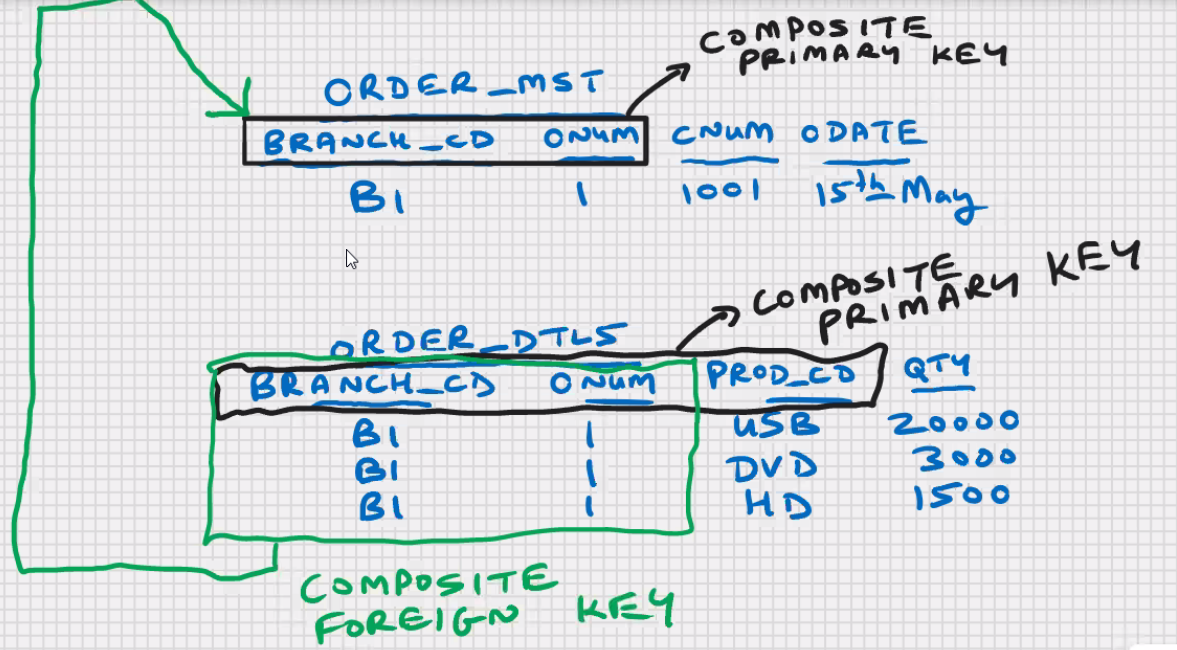


Insert into emp

Values (‘7’,’G’,7000,2,’8’);

* first it inserts, then it checks for the constraint, then it will rollback and give an error message

**COMPOSITE FOREIGN KEY**



Create table order\_mst …..;

Create table order\_dtls

(

Branch\_cd char(4),

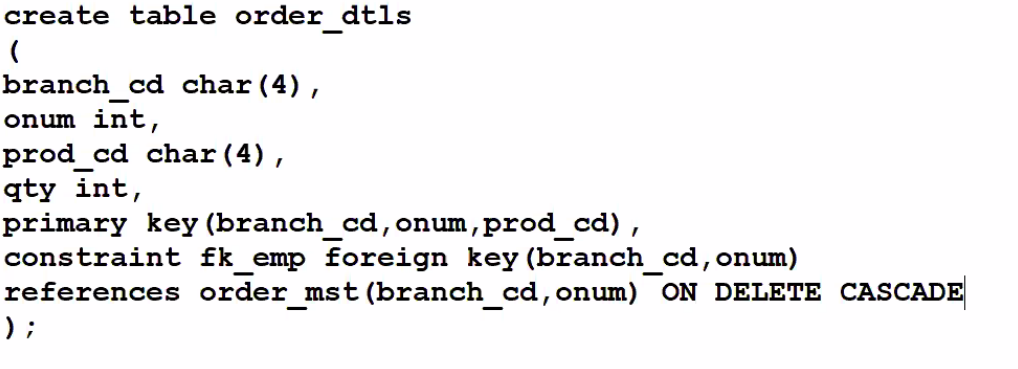
Onum int,

Prod\_cd char (4),

Qty int,

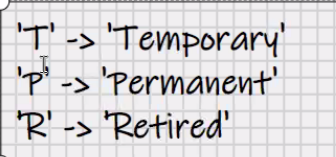
Primary key(branch\_cd, onum,prod\_cd),

Foreign key(branch\_cd,onum) references order\_mst (branch\_cd,onum)



**CHECK CONSTRAINT**

* used for validation (used for checking purposes)
* e.g. sal < 10000 ,age >21,etc
* in check constraint you can use relational operators,

logical operators,arithmetic operators,

special operators(in, between, like),

call single\_row functions

e.g. upper, round, etc.

create table emp

(empno int auto\_increament primary key,

Ename varchar (25) check (ename = upper(ename)),

Sal float **default 7000**

**check (sal > 5000 and sal < 300000),**

Deptno int,

Status char(1) **default ‘T’**

**check (status in (‘T’,’P’,’R’)),**

Comm flat not null,

Mob\_no char(15) unique,

Check (sal+comm < 500000),

Constraint fk\_emp foreign key(deptno) references dept (deptno)

);

* default is not a constraint
* default is a clause that you can use with create table
* if user is entering some, then it will take value
* if nothing is entered, then it will take default value
* to make use default value and auto increment, use the following insert statement:-

insert into emp (ename, deptno, comm, mob\_no)

values (……………………………..);

****

**Data is of 2 types:**

1. **User data**

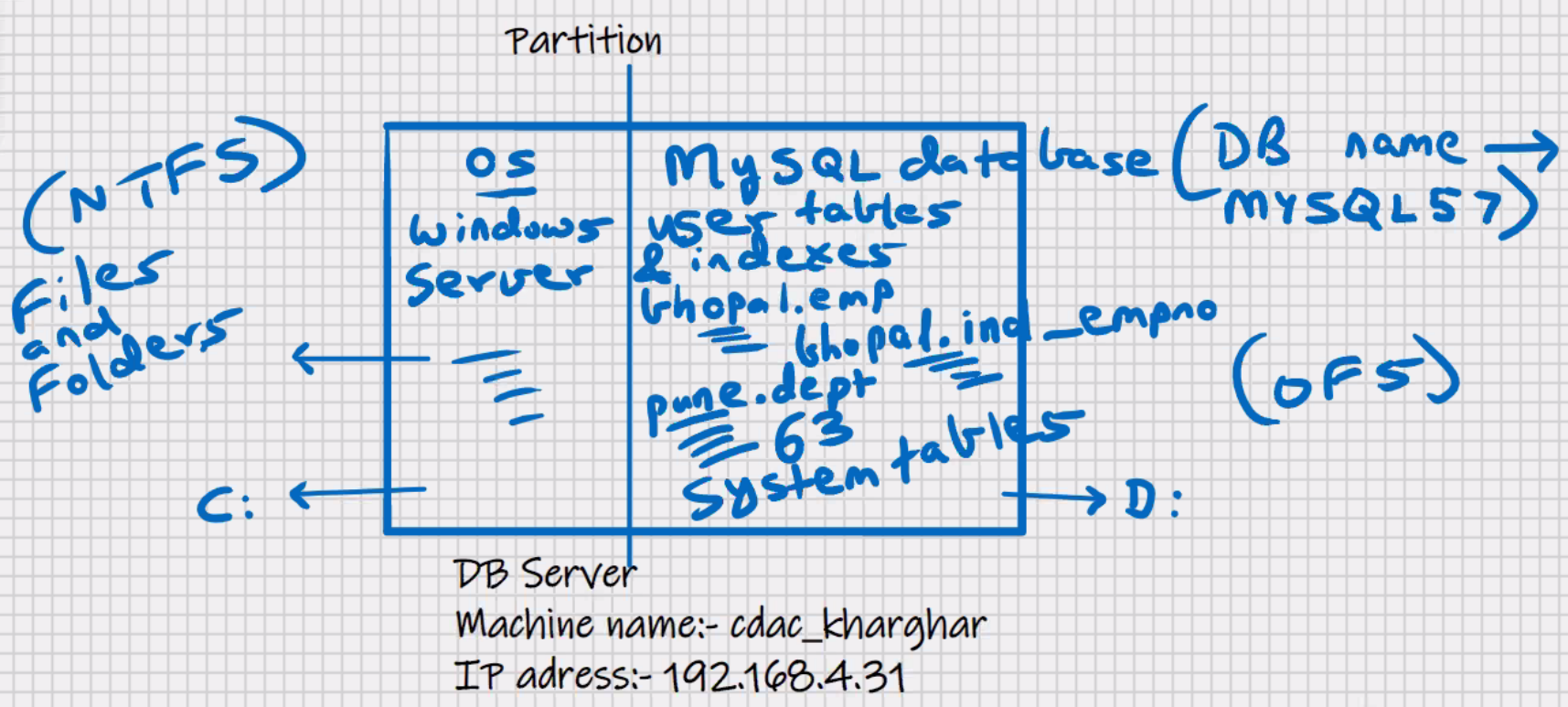
* user created
* user tables and indexes

1. **System data**

* System created (MySQL created)
* Data that is stored in system tables
* **System data** is also known as **METADATA**
* Metadata -> data about data

**SYSTEM TABLES**

* 63 system tables in MySQL
* Store complete information about the database
* System tables are stored in **information\_schema**
* **Data dictionary ->** set of system tables is known as data dictionary
* **Also known as DATABASE CATALOG**
* E.g. statistics, table\_constraints, key\_column\_usage, table\_privileges, etc.
* All system tables are read\_only
* You can only select from them (DML operations are not allowed)
* **DDL for you is DML for system tables**

****