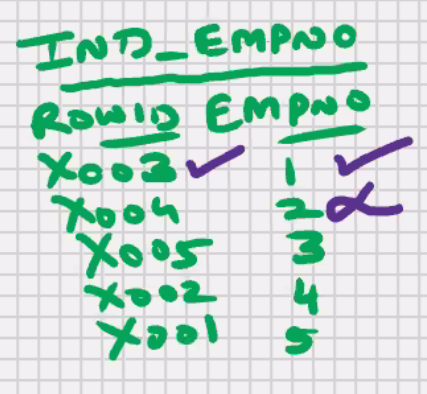
**INDEXES**



* Present in all RDBMS, all DBMS, and some of the programming languages also
* Used to speed up the search operations (for faster access)
* Used to speed up the select statement with a where clause
* In MySQl and Oracle the indexes are automatically invoked as and when required.
* Indexes are automatically updated by MySQL and Oracle for all your DML Operations.
* MySQL and Oracle are self -managing RDBMS
* Duplicate values are stored in an index
* Null values are not stored in an index

Select \* from emp where empno is null;

MySQL will do a full table scan

Select \* from emp where empno = 0;

In other RDBMS:-

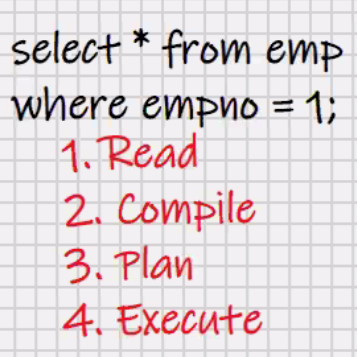
Use index ind\_empno;

Select \* from emp where empno = 1;

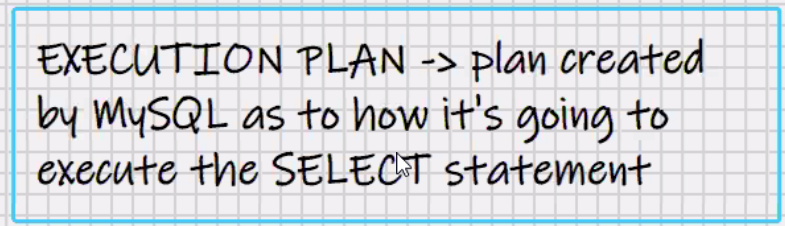
In other RDBMS:

Insert/update/delete . . . . . . . . . . . . :

REINDEX;

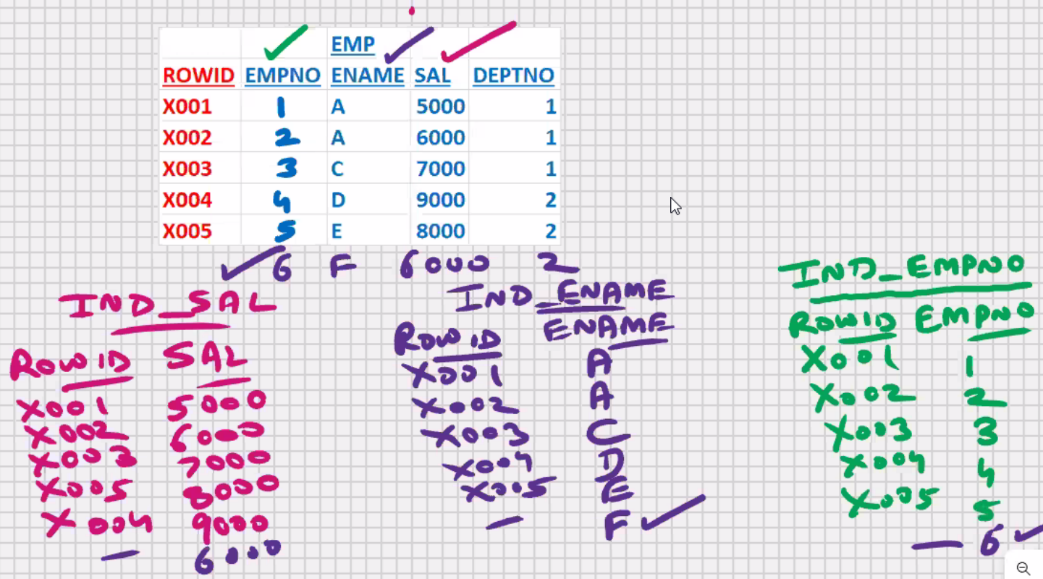
Select \* from emp Where empno = 1;

Select \* from emp Where ename = ‘A’;



Select \* from emp Where sal > 7000 ;

* In MySQL and oracle , there no upper limit on the number of indexes per table



* Larger the number of indexes, the slower would be the DML operations
* Fast searching vs Fast DML operations? (AS per User-requirements)

Select \* from emp

Where empno = 2;

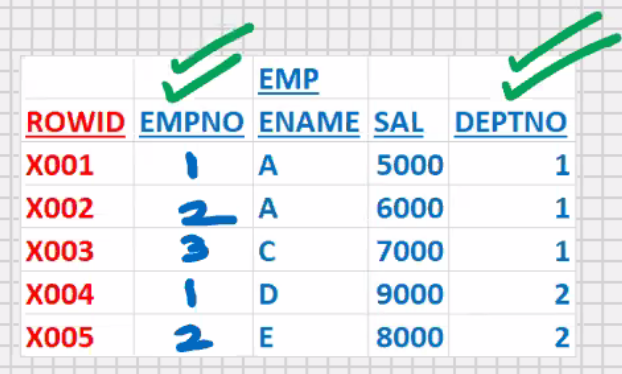
Select \* from emp

Where sal > 5000;

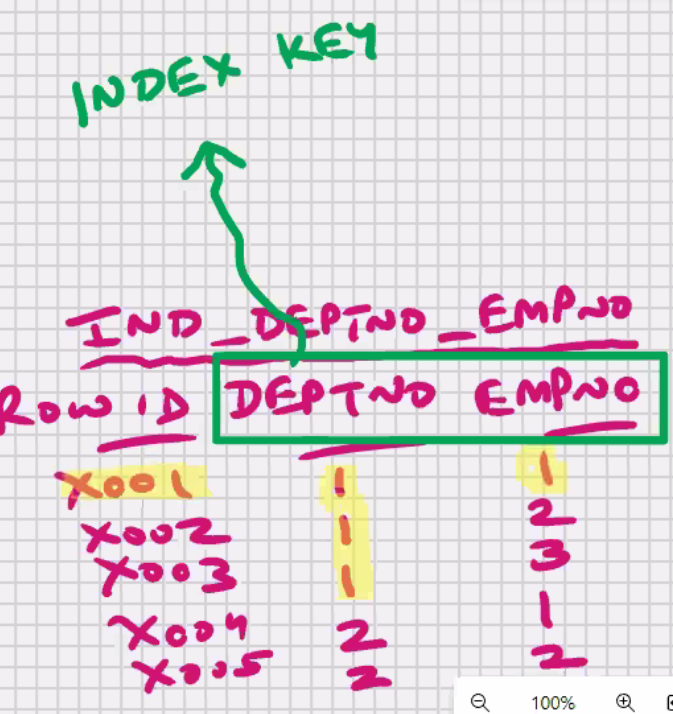
Select \* from emp

Where empno= 2 and sal >5000;

* If you have 2 or more independent columns in the where clause create indexes for all such columns; and MySQL will use all the necessary indexex as and when required



**Composite index** -> combine 2 or more inter dependent columns in

 a single index

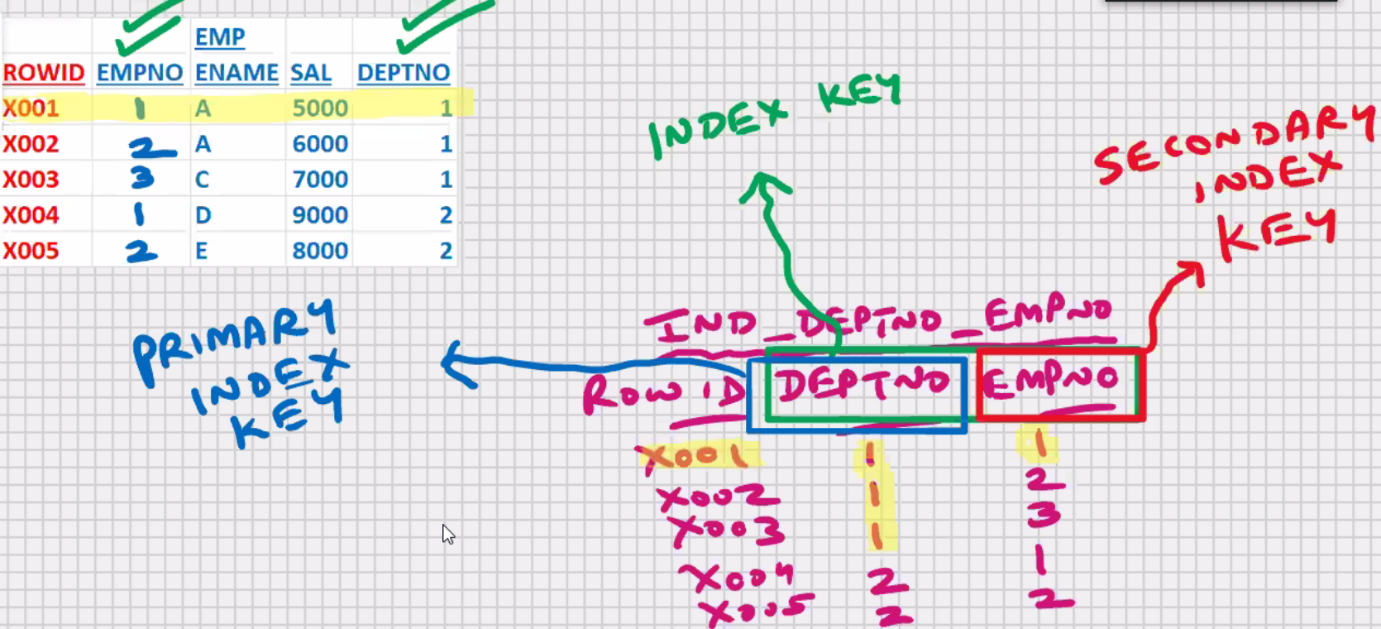
Select \* from emp

Where deptno = 1 and empno = 1;

**Index key** -> column or set of columns

On whose basis the index has been

Created



* In **MySQL** you can **combine upto 32 columns** in a composite index and in oracle upper limit is 16 columns
* You cannot index text and blob

Select \* from emp

where empnp = 1;

**condition when an index should be created:**

* Used to speed up the select statement with a where clause, order by clause, distinct, group by clause, union, intersect, minus
* If your select statements retrieves < 25% of the table data

Select \* from emp where empno = 1;

Select \* from emp Order by empno;

Select distinct empno from emp;

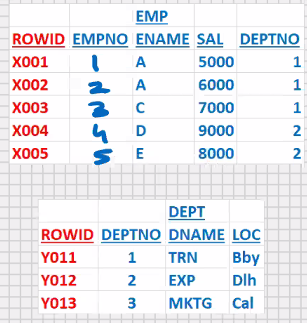
Select deptno, sum(sal) from emp Group by deptno;

Select \* from emp where empno = 1;

Select \* from emp where empno = 5;

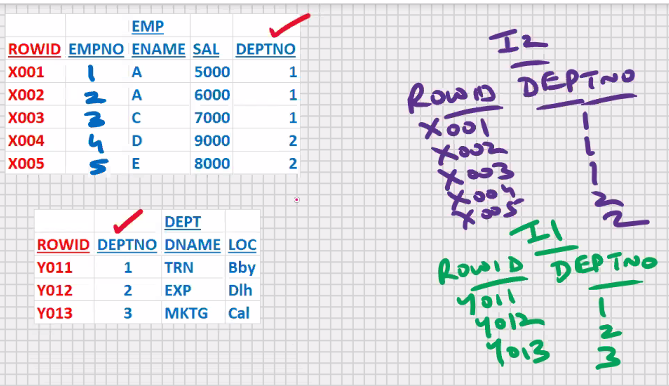
Select \* from emp where empno != 1;

* Key is another name for column
* Primary key and unique columns should always be indexed
* Primary key column is the best column for searching



Select dname, ename from emp, dept

Where dept.deptno = emp.deptno;



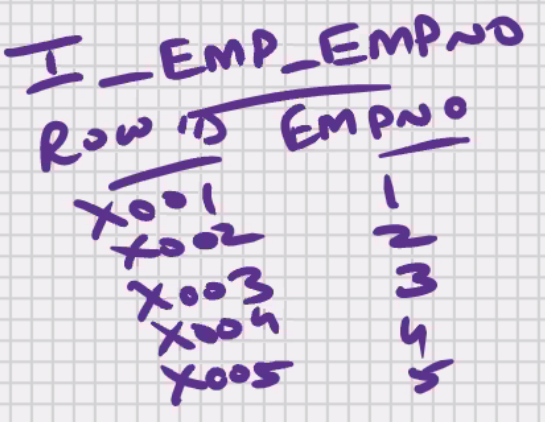
* Common column In join operations should always be indexed

In MySQL :

**Create index indexname on table (column)**

Create index i\_emp\_empno on emp(empno);

Select \* from emp where empno = 1;



Select \* from i\_emp\_empno; -> error

Create index i\_emp\_ename on emp(ename);

Create index i\_emp\_sal on emp(sal);

Create index i\_emp\_deptno\_empno on emp(deptno,empno);

Create index i\_emp\_empno on emp(empno desc);//descendingorder

Create index i\_order\_onum on orders(onum desc);

Create index i\_emp\_deptno\_empno on emp(deptno desc,empno);

Create index i\_emp\_deptno\_empno

on emp(deptno desc,empno desc);

**To see which all indexes are created for specific tables:-**

show indexes from emp;

**To see all indexes on all tables in all schemas:-**

use information\_schema;

select \* from statistics; <- this is a system table

**To drop the index:-**

Drop index i\_emp\_empno on emp;

**Unique index:**

Create unique index i\_emp\_empno on emp(empno);

* performs one extra function, it wont allow the user to insert duplicate values in empno
* At the time of creating the unique index, if you already have duplicate values in empno, then MySQL will not allow you to create the unique index

**Types of Indexex:**

1.Normal Index

2.Unique Index

3.bitmap Index

4.Index Organized table

5.index partitioning (global and Local)

6.etc

**MySQL – SQL – PRIVILEGES**

GRANT/REVOKE (DCL COMMANDS)

Dishi\_mysql> grant select on emp to vaibhai;

Dishi\_mysql> grant insert on emp to vaibhai;

Dishi\_mysql> grant update on emp to vaibhai;

Dishi\_mysql> grant delete on emp to vaibhai;

Dishi\_mysql> grant select,insert on emp to vaibhai;

Dishi\_mysql> grant all on emp to vaibhai;

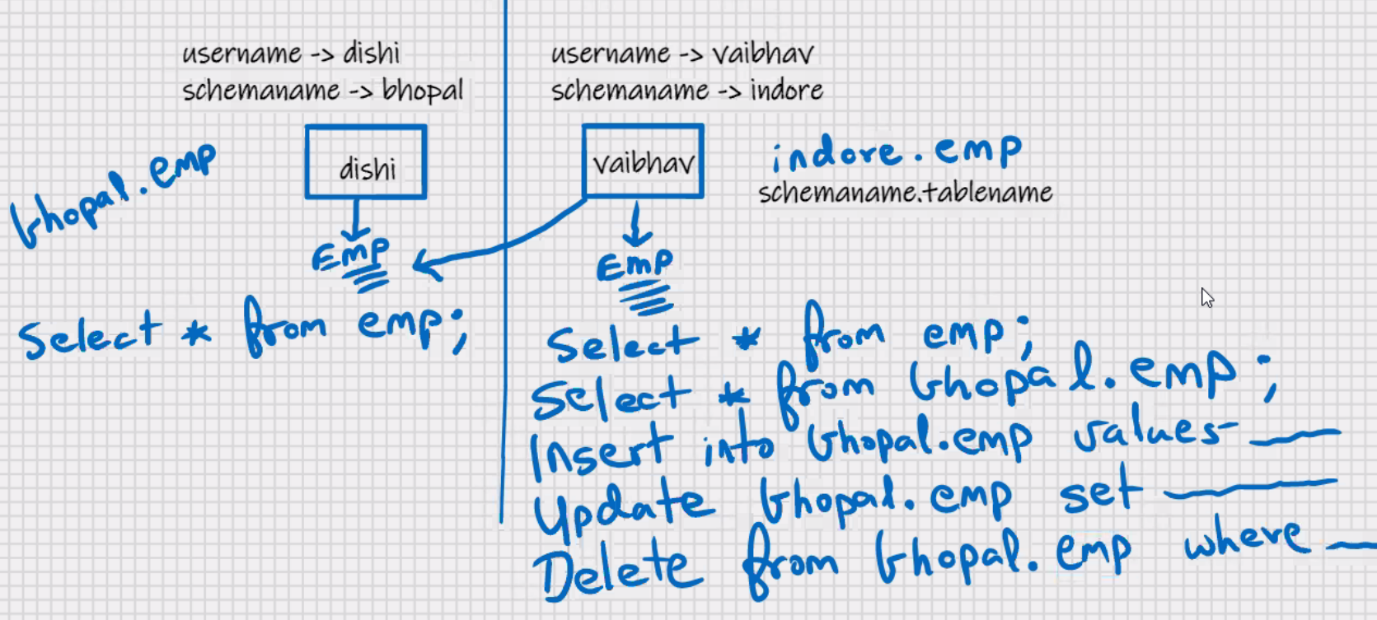
Dishi\_mysql> grant select on emp to vaibhai, manoj;

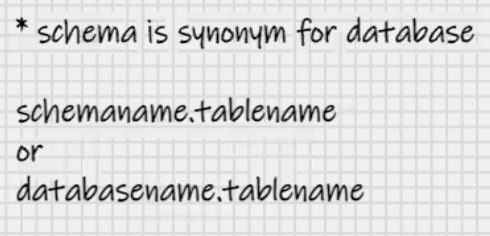
Dishi\_mysql> grant select on emp to public; //all users

Revoke select on emp from Vaibhav;

To see which all permissions have been granted and received:

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





Dishi\_mysql> grant select on emp to Vaibhav with grant option;

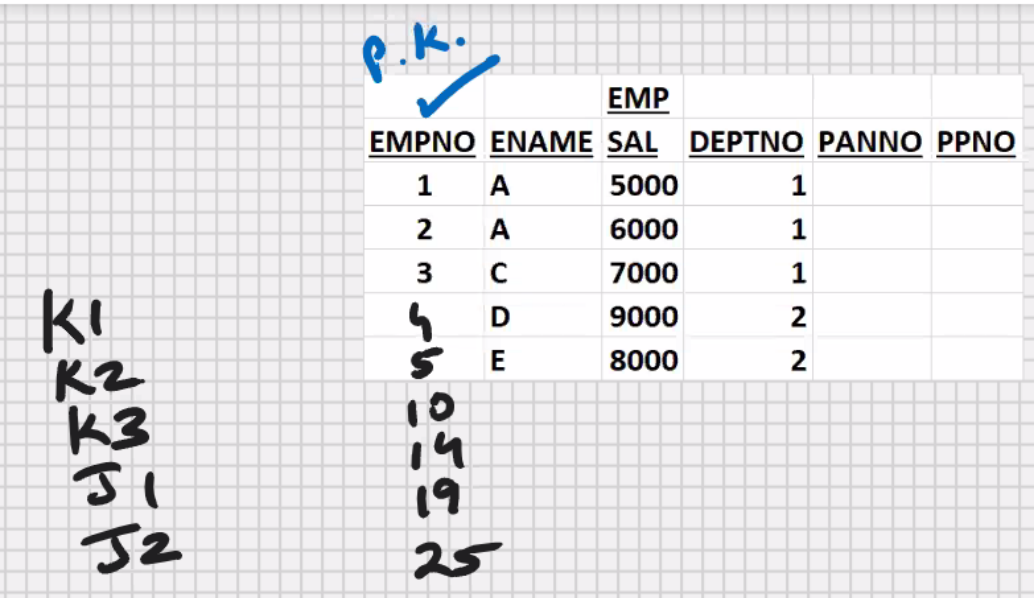
Vaibhav mysql> grant select on Bhopal.emp to manoj;

**MySQL- SQL- CONSTRAINTS**

* Limitation / restrictions imposed on a table

**Primary Key (Primary column)**

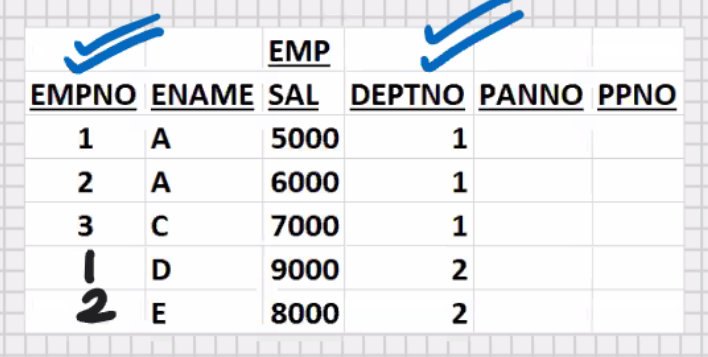
* Column or set of set of column that uniquely identifies a row(e.g. empno)
* Duplicate values are not allowed
* Null values are not allowed (mandatory column)
* Its recommended that every table should have a primary key



* Purpose of primary key is row uniqueness
* With the help of primary key you can distinguish between 2 rows of a table
* Text and blob cannot be primary key
* Unique index is automatically created

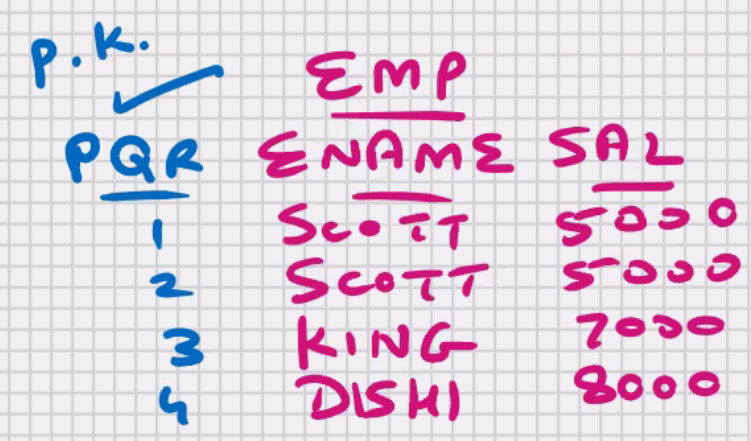
**COMPOSITE PRIMARY KEY** -> combine 2 or more columns together to serve the purpose of primary key

* In MySQL you can combine upto 32 columns in a composite primary key

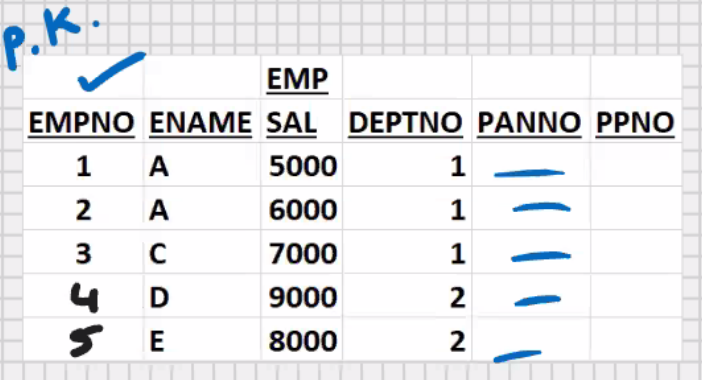


**SURROGATE KEY** ->

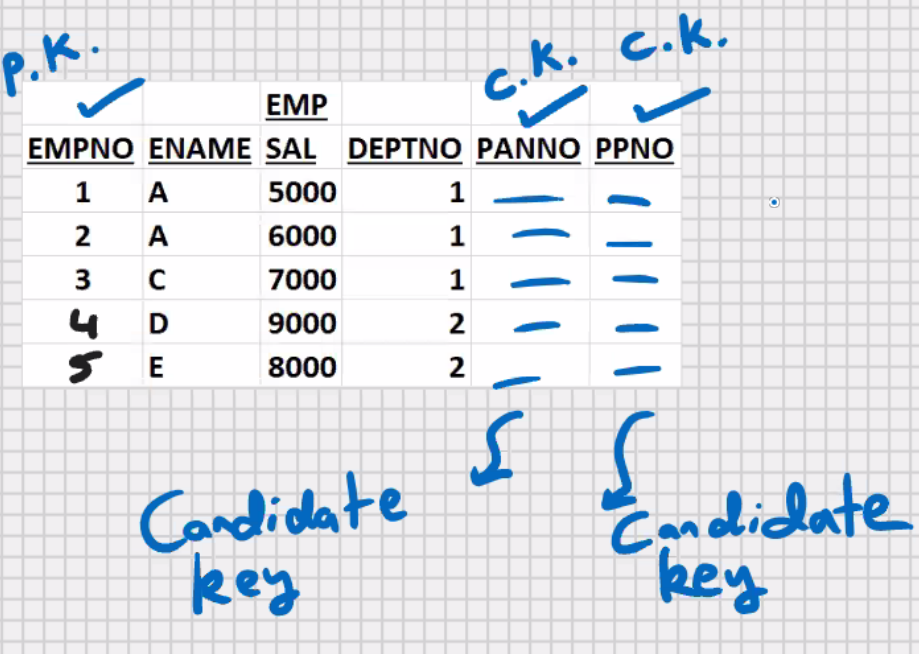
* If you cannot identify primary key in your table , add an extra column to the table, and make it the primary key of your table
* such a primary key which is not an original column table, such a primary key is known as surrogate key
* surrogate key is not a constraint
* surrogate key is a definition
* primary key column in the best column for searching and with char datatype the searching a retrieval is very fast
* for (surrogate key) primary key, char datatype is recommended



* if you declare a composite primary key, then mysql automatically creates a composite unique index
* YOU CAN HAVE ONLY 1 PRIMARY KEY PER TABLE



**CANDIDATE KEY** ->



* Candidate key is not a constraint
* Candidate key is a definition
* Besides the primary key, any other column in the table that can also serve the purpose of primary key, is a good candidate for primary key, is known as candidate key.
* Its good to have a couple of candidate keys in your table; in future if you drop the primary key (empno )column, then your table is left without a primary key; then you can immediately make PANNO or PPNO as the new primary key

Create table emp

(

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

Ename varchar(25),

Sal float,

Deptno int

);

Insert into emp values(‘5’,’F’,5000,2); -> error

Insert into emp values(null,’F’,5000,2); -> error

* All constraints are at server level
* You can insert using MySQL command line client, MySQL Workbench, MySQL – PL program, java, MS .Net, etc.; the constraints will always be valid
* You can perform DML operations using any front-end s/w, the constraints will always be valid
* This is known as DATA INTEGRITY
* Internally a constraint is a mysql created function, it performs the validations

To find out which all constraints you have created:-

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’;

* Unique index is automatically created

Show indexes from emp;

**To drop the primary key constraint:-**

Alter table emp drop primary key;

**To add the primary key afterwards to an existing table:-**

Alter table emp add primary key (empno);

**To change the primary key column:-**

1. drop the original primary key constraint
2. add a new primary key constraint