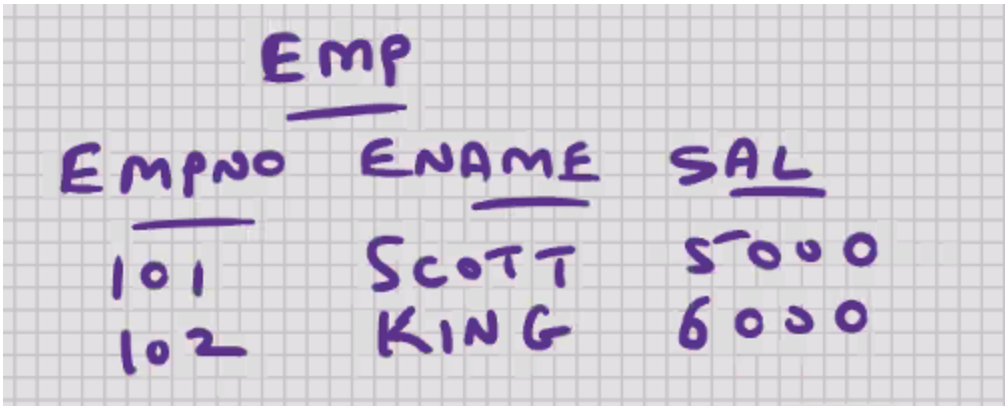
Day_8

MySQL- SQL - ALTER TABLE(DDL Command)



- Rename a table
- Add a column to the table
- Drop a column
- Increase width of column

Indirectly:

- Reduce width of column
- change the datatype
- Copy the rows from on table into another table
- Copy a table(for testing purposes)
- Copy only the structure of the table.
- Rename a column
- change the positions of columns in table structure(because of null values, for storage considerations)

select count(*)-*count(empno)*, *count()-count(ename)*, *count(*)-count(sal) from sal;←(To get null value count)

To Rename table:

- rename table emp to employees;← (Command)
- RENAME a DDL command(extra command in MySQL)

To Add column:

- alter table emp add gst float;← adds gst column at last

To Drop column:

- alter table emp drop column gst;← deletes the gst column
- if you drop a column, then indexes on that column are dropped automatically

To increase width of column

- alter table emp modify ename varchar(30);

To reduce width of column

- alter table emp modify ename varchar(2A0);
- data will get truncated

```
In Oracle:-
alter table emp modify ename varchar(20);       <- ERROR

*       you can reduce the width provided the contents are null

alter table emp add x varchar(25);
update emp set x = ename, ename = null;
alter table emp modify ename varchar(20);
/* Data testing with X column; check that none of the names > 20 chars */
update emp set ename = x;
alter table emp drop column x;

*       IT'S RECOMMENDED YOU USE ABOVE SOLUTION FOR MySQL
```

Change the datatype:

- alter table emp modify empno char(20); ←changes the datatype of empno column

Copy rows from one table to another table:

- insert into emp_kh select * from emp_jh;

- above command will work provided the structure of both the tables is the same

To copy specific rows only:

- insert into emp_kh select * from emp_jh where.....;

To Copy a table(for testing purpose):

- create table emp2 as select * from emp;← command to create a copy of the table.
- when you create table using sub-query, indexes on original table are not copied into the new table.
- If you want indexes on new table then you will have to create them manually

To copy specific rows/columns only:

- insert into emp_kh select empno, ename from emp_jh where.....;

To copy only structure of table:

- Solution #1: create table emp2 as select * from emp;

  delete from emp2;

  commit;
- Solution #2: create table emp2 as select * from emp;

  truncate table emp2;
  - Truncate is a DDL command
  - Truncate will DELETE ALL the rows and Commit
- Solution #3: create table emp2 as select * from emp

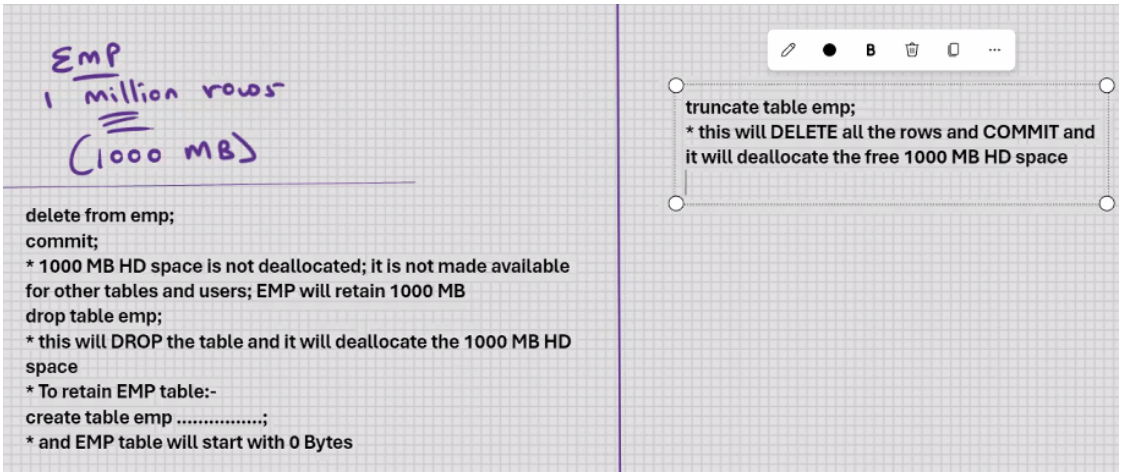
  where 1 = 2

TRUNCATE Vs. DELETE:

- DELETE:
  - Common for all RDBMS

- ANSI SQL (ANSI Stds. & ISO Stds.)
- DML command
- Requires commit
- Rollback is possible
- WHERE clause is supported, can DELETE specific rows only
- DELETE triggers on table will execute

- TRUNCATE:
  - Extra command in MySQL and Oracle
  - Not an ANSI SQL(MySQL Stds and Oracle Stds.)
  - DDL command
  - Auto-commit
  - Rollback is not possible
  - WHERE clause is not supported, TRUNCATE will DELETE all the rows and commit
  - DELETE triggers on table will not execute because TRUNCATE is not a DML command



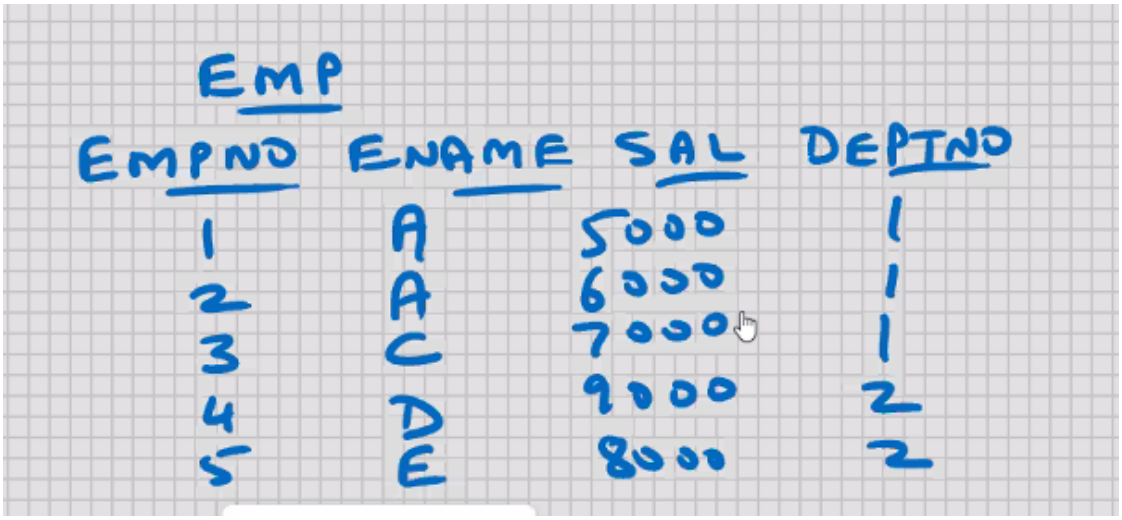
Rename a column:

```
create table emp2
as
select empno, ename, sal salary
from emp;

drop table emp;

rename table emp2 to emp;
```

Change the position of the columns:

```
create table emp2
as
select ename, sal, empno from emp;

drop table emp;

rename table emp2 to emp;
```

MySQL - SQL - Constraints:



- limitations/restrictions imposed on a table

PRIMARY KEY:

- key is another word for column
- Primary column
- column or set of columns that uniquely identifies a row e.g. EMPNO
- duplicate values are not allowed; it has to be unique
- null values are not allowed ; this is a mandatory column
- it is recommended that every table should have a Primary Key; it helps from a long-term perspective
- purpose of Primary key is row uniqueness; with the help of Primary key column you distinguish between 2 rows o the table
- TEXT and BLOB cannot be Primary Key
- Unique index is automatically created for Primary Key(because Primary Key is the best column for searching)
- COMPOSITE PRIMARY KEY→ combine 2 or more columns together to serve the purpose of Primary key; preferably they should be inter-dependent columns
- In MySQL, you can combine upto 32 columns in a Composite Primary Key
- if you have a Composite Primary Key, then a Composite Unique Index is automatically created
- YOU CAN HAVE ONLY 1 PRIMARY KEY PER TABLE.
- CANDIDATE KEY→ its not a constraint; is a definition; is the column that is Primary key in waiting; besides the Primary key, any other column in the table that could serve the purpose of Primary Key, is a good candidate for Primary Key, is known Candidate key

```
*       Steps for identifyiing Primary key:-
        1. Identify some key column and make it the Primary key of your
        table
        2. If you cannot identify some key column, then try to implement
        Composite Primary key
        3. If this is not possible, then you add an extra column to the
        table to serve the purpose of Primary key
```

- SURROGATE KEY→ is not constraint; it is a definition; When an extra column to the table to serve the purpose of Primary key, then such a Primary key is known as Surrogate key.
- Primary key is best column for searching, and with CHAR datatype the searching and retrieval is very fast; therefore for Surrogate key the CHAR datatype is recommended

Creating a table with primary key:

create table emp(

empno char(4) primary key,

ename varchar(25),

sal float,

deptno int

);

- all constraints are at Server level; you can perform the DML operations using MySQL Command line Client, or MySQL Workbench etc. or any other front-end s/w, the constraints will always be valid (this concept is known as DATA Integrity)

To see Constraints:

```
select * from information_schema.table_constraints;

select * from information_schema.table_constraints
where table_schema = 'cdacmumbai';

select * from information_schema.key_column_usage
where table_name = 'emp';
```

- unique index automatically created

show indexes from emp;

- to drop primary key constraint:

alter table emp drop primary key;

- to add primary key to an exiting table:

alter table emp add primary key(empno);

```
To change the Primary key column:-

Drop the existing Primary key constraint  & alter table & then we can add
new primary key constraint
```

Creating a table with Composite primary key:

create table emp(

empno char(4),

ename varchar(25),

sal float,

deptno int,

primary key(deptno, empno)

);

```
Constraints are of 2 types:-
a. Column level constraint
*       constraint specified on one column
b. Table level constraint
*       Composite
*       constraint 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

- duplicate values are allowed
- always a column level constraint

Creating a NOT NULL :

```
create table emp
(
empno char(4),
ename varchar(25) not null,
sal float not null,
deptno int not null
);
```

ENTITY INTEGRITY → PRIMARY KEY

REFERENTIAL INTEGRITY → FOREIGN KEY

DOMAIN INTEGRITY → Datatype, e.g. int, char(10)

- In MySQL, nullability is part of the datatype
- to see the not null columns: desc emp;

To drop the not null constraint:

```
alter table emp modify ename varchar(25) null;
```

To add the not null constraint to existing table:

```
alter table emp modify ename varchar(25) not null;
```

UNIQUE Constraint:

- duplicate values are not allowed
- null values are allowed
- can INSERT any number of null values, but no duplicate values
- unique index is automatically created

- TEXT and BLOB cannot be unique
- 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

Creating a unique constraint:

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 constraint

);

unique is also an index, so drop it:

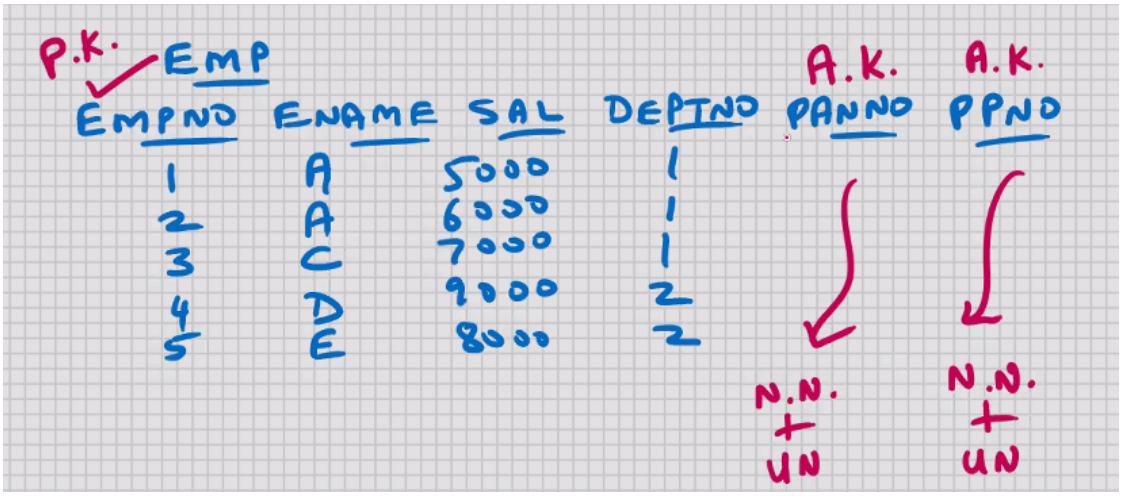
- drop index mon_no on emp;
- drop index deptno on emp;

```
To add unique constraint to an existing table:-

alter table emp add constraint u_emp_mob_no unique(mob_no);

constraint u_emp_mob_no -> constraint constraintname
constraint u_emp_mob_no -> optional
```

```
*       column level constraint can be specified at table level (end of
        structure), but a table level composite constraint can never be
        specified at column level
*       most developers prefer this, it make the CREATE table command more
        Readable
*       column level constraint can be specified at table level (end of
        structure), except for the not null constraint which is always a
        table level constraint and therefore the syntax will not support
        you from specifying it at table level
```

```
*       1 column can have more than 1 constraint
        e.g.
        primary key and foreign key
        not null and unique
        etc.

Solution for candidate key:-
not null constraint + unique index
OR
not null constraint + unique constraint
```

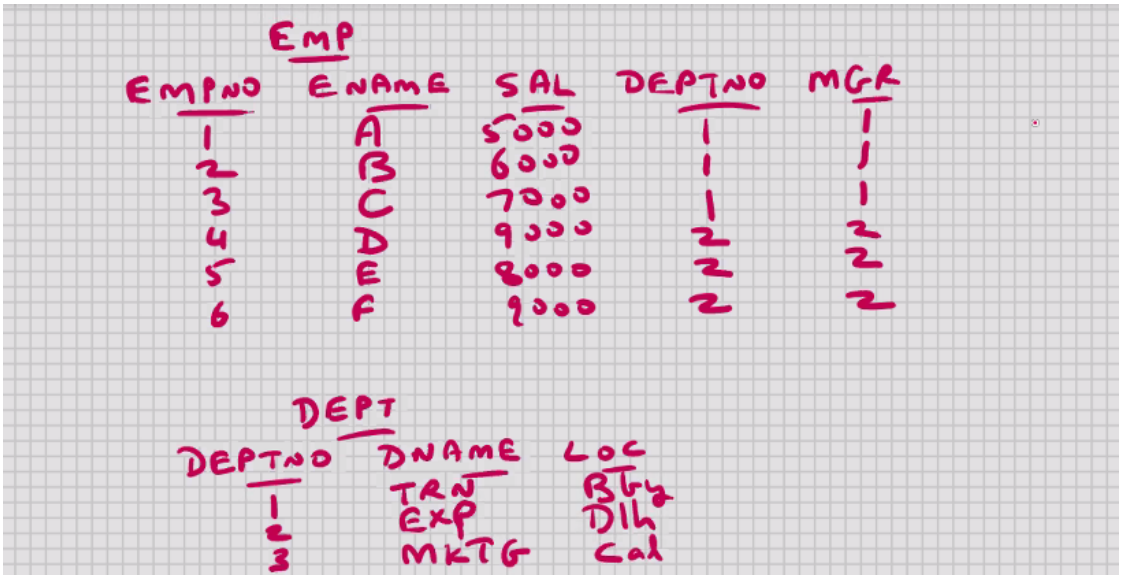
```
*       with the help of above logic, indirectly you can have
        multiple Primary keys in the table; one directly, the
        remaining indirectly
```



- ALTERNATE KEY→ is not a constraint; is a definition; for the candidate key column, if you give it a Not null constraint + unique index/constraint, then it becomes an alternative to Primary key, then such Candidate key column is known ALTERNATE key
- SUPER KEY→ is not a constraint; is a definition; if you have one or more alternate keys in table, then the Primary key is also Known as SUPER KEY

FOREIGN KEY:

- Foreign column (column that is derived from else were)



- column or set of columns that references a column or set of columns of some tables
- Foreign key constraint is specified on the child column (not the parent column)
- Parent column has to Primary key or unique(this is a pre-requisite for Foreign key)
- Foreign key will allow duplicate values
- Foreign key will allow null values also
- Foreign key may reference column of same table also(known as self-referencing)(parent column and child column both are present in the same table)

Creating a foreign key on a table:

```
 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) reference dept(deptno),

constraint fk_emp_mgr foreign key(mgr) reference emp(empno)

);

```
select * from information_schema.table_constraints;

select * from information_schema.table_constraints
where table_schema = 'cdacmumbai';

select * from information_schema.key_column_usage
where table_name = 'emp';
```

```
alter table emp drop foreign key fk_emp_deptno;

alter table emp add foreign key(deptno) references dept(deptno);
```

To disable the foreign key constraint:

- for current connection:
  - set foreign_key_checks = 0;
  - set foreign_key_checks=1;
- for all connection:
  - set global foreign_key_checks = 0;
  - set global foreign_key_checks=1;

```
*       you cannot delete the parent row when child rows exist

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) reference dept(deptno) on delete cascade,

constraint fk_emp_mgr foreign key(mgr) reference emp(empno)

);

- if you delete the parent row, then MySQL will automatically delete the child rows also ← on delete cascade

```
To retain the child rows:-

update emp set set deptno = null where deptno = 2;
delete from dept where deptno = 2;
```

create table emp(

empno char(4) primary key,

ename varchar(25),

sal float,

deptno int,

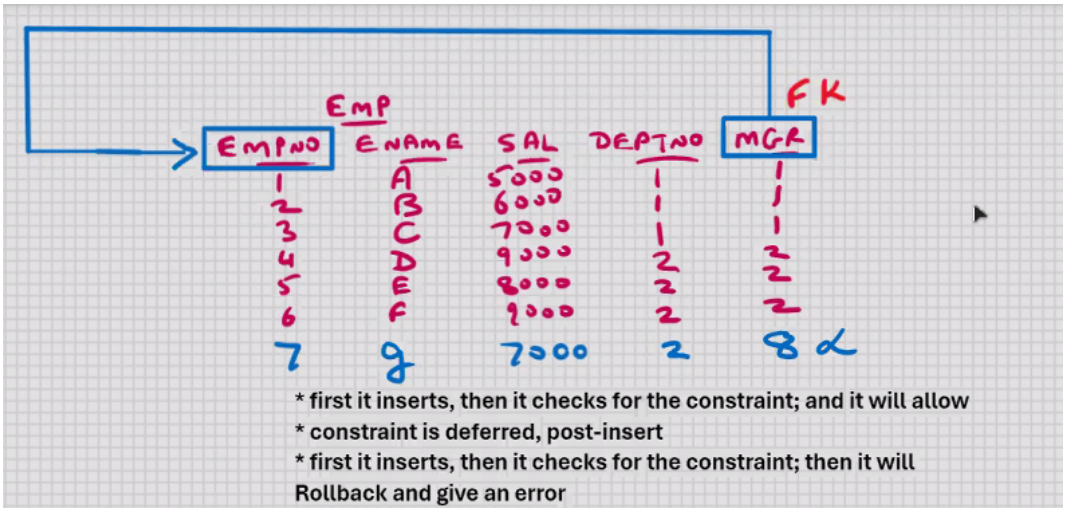
mgr char(4),

constraint fk_emp_deptno foreign key(deptno) reference dept(deptno) on delete cascade on update cascade,

constraint fk_emp_mgr foreign key(mgr) reference emp(empno)

);

- if you update the parent row, then MySQL will automatically update the child rows also ← on update cascade



Internal working of constraint