# Day_9

MySQL - SQL - constraints(5)

- primary key, not null, unique, foreign key

CHECK:

- used for validations(used for checking purposes)
- e.g sal<10000, age>25

create table emp(

empno int auto_increment,

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

sal float  default 7000 check (sal between 5001 and 2999999),  ←column level

deptno int reference dept(deptno),

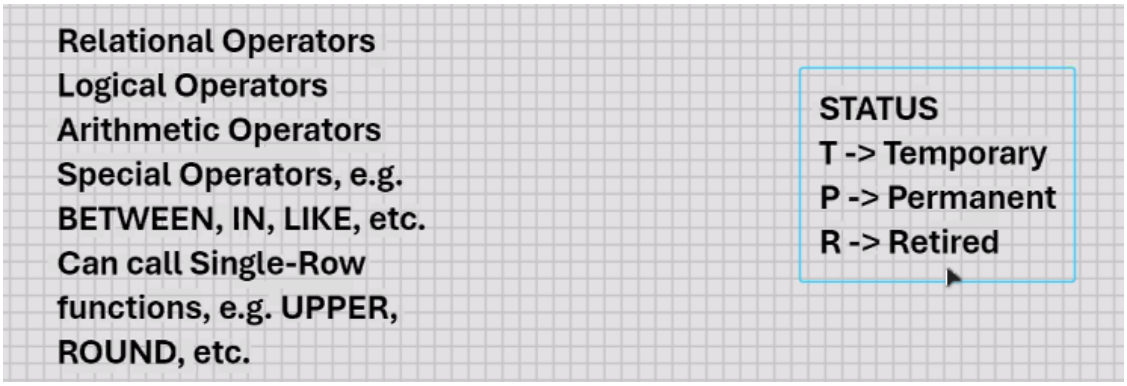
status char(1) default 'T' check (status in('T', 'P', 'R')),  ←column level

comm float not null,

mob_no char(15) unique,

check(sal+comm < 5000000)  ←table level

);




- DEFAULT is not constraint

- DEFAULT is clause that you can with CREATE TABLE
- if you enter some value, then it will take that value; if nothing is specified, then it will take default value

To make use of DEFAULT value and AUTO_INCREMENT, use the following INSERT statement:

insert into emp(ename, deptno, comm, mob_no) value(.....);

- AUTO_INCREMENT → by default is starts from 1, by default it increments by 1; upper limit = 9.9 * 10^125
- Rollback and Commit has no effect on auto_increment(it has been designed in this manner on purpose keeping in view a multi-user environment)

To avoid the problem of missing numbers:

- Do not issue the INSERT statements to the database at the time of data entry; when user does the data entry, you store the rows in an array; when user issues a Commit, you issue the INSERT statements to the database followed by Commit

MySQL - SQL - PRIVILEGES

Grant and Revoke(DCL):

root_mysql> grant select on cdacmumbai.emp to scott@localhost;

cdacmumbai.emp → dbname.tablename

cdacmumbai.emp → schemaname.tablename

- SCHEMA IS A SYNONYM FOR DATABASE

root_mysql> grant insert on cdacmumbai.emp to scott@localhost;

root_mysql> grant update on cdacmumbai.emp to scott@localhost;

root_mysql> grant delete on cdacmumbai.emp to scott@localhost;

root_mysql> grant select, insert on cdacmumbai.emp to scott@localhost;

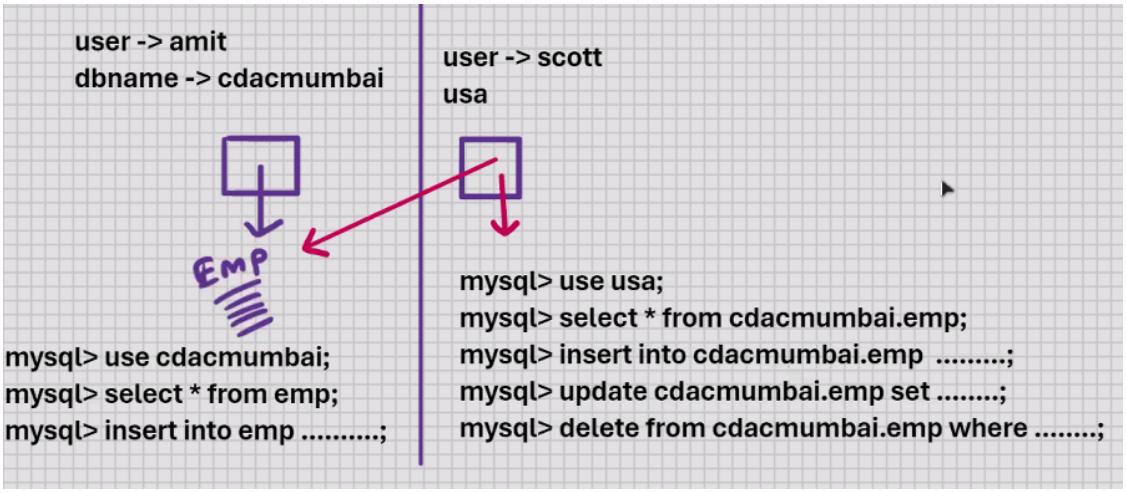
root_mysql> grant all on cdacmumbai.emp to scott@localhost;

root_mysql> grant delete on cdacmumbai.emp to scott@localhost, king@localhost;

root_mysql> grant delete on cdacmumbai.emp to public;

root_mysql> revoke select on cdacmumbai.emp from scott@localhost; ← no longer permission;

- we cannot grant a drop permission




```
root_mysql> create user amit@localhost identified by 'student';

root_mysql> create user amit@'%' identified by 'student';

root_mysql> create user amit@'d%' identified by 'student';

root_mysql> grant all privileges on cdacmumbai.* to amit@localhost;
root_mysql> grant all privileges on cdacmumbai.* to amit@'%';
root_mysql> grant all privileges on cdacmumbai.* to amit@'d%';
```

MySQL - System tables

- MySQL created
- automatically created when you install MySQL
- 78 System tables in MySQL v9
- set of System tables is also known as DATABASE CATALOG
- set of System tables is known as DATA DICTIONARY

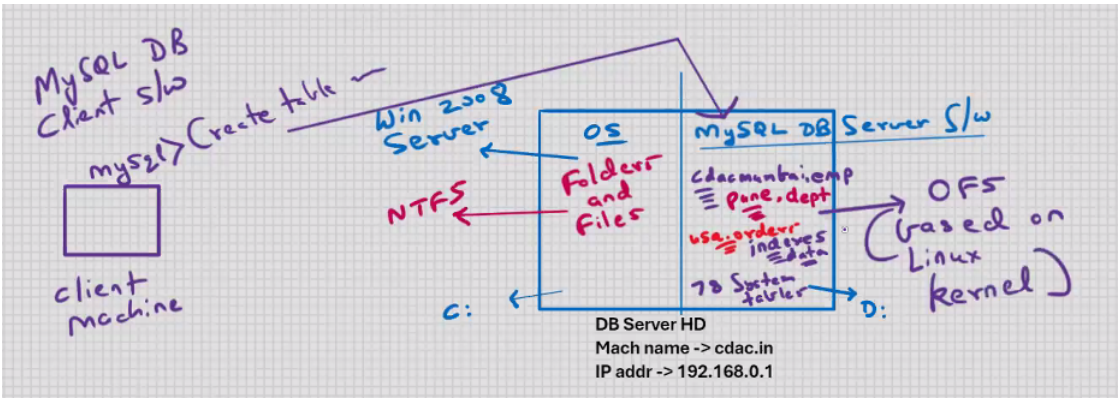
- stored in information_schema
- System tables store complete information about the database
- e.g. statistics (for indexes), table_constraints, key_column_usage, table_priviledges, etc.

```
Login as root user:-
mysql> use information_schema;
mysql> show tables;
```

- all the system tables are READ_ONLY
- DDL FOR USER IS DML FOR SYSTEM TABLE

Data is of 2 types:

1. user data
    - user created
    - user tables and indexes
2. system data (Metadata)
    - MySQL created
    - data in System Tables
    - also known as Metadata(data about data)

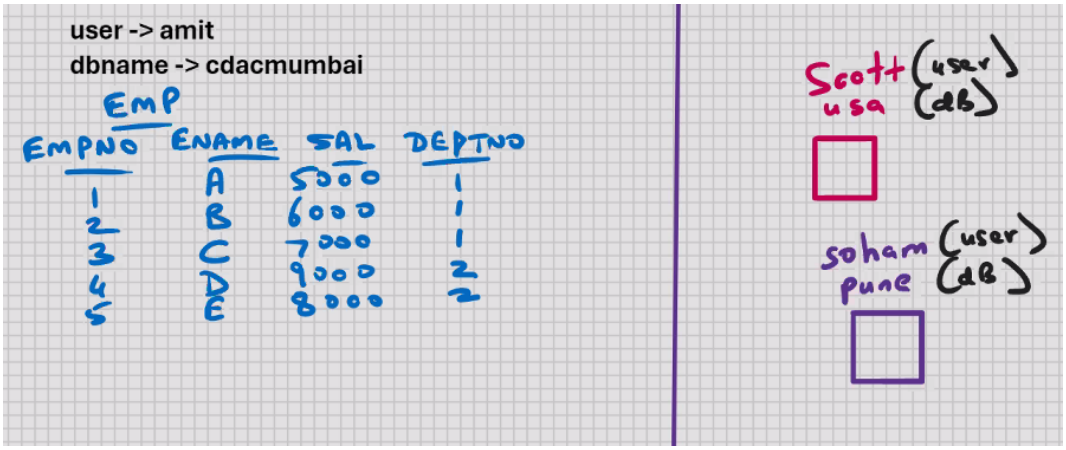


MySQL - STORED OBJECTS

- objects that are stored in the database
- e.g. CREATE ..... tables, indexes
- anything that you do with CREATE command is stored object

VIEWS:

- present in all RDBMS and some of the DBMS.




- view is a handle to a table
- stores the address of the table
- view is a HD pointer (stores the address of the table) (Known as LOCATOR)
- used for indirect access to the table
- used for SECURITY purposes
- used to restrict the access of the users

Create View:

mysql> create view viewname .....;

amit_mysql> create view v1

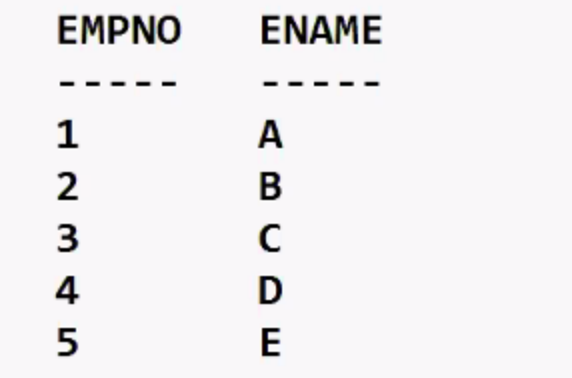
as

select empno, ename from emp;

root_mysql> grant select on cdacmumbai.v1 to scott@localhost;

scott_mysql> select * from cdacmumbai.emp; ← ERROR

scott_mysql> select * from cdacmumbai.v1;



| EMPNO | ENAME |
|---|---|
| 1 | A |
| 2 | B |
| 3 | C |
| 4 | D |
| 5 | E |

- used to restrict the column access
- VIEWS DOES NOT CONTAIN DATA
- only the definition is stored; data is not stored
- view is stored query (stored in the database)
- SELECT statement on which the view is based, it is stored in the DB in the COMPILED FORMAT
- view is an executable format of SELECT statement
- hence the execution will be very fast
- hiding source code from end user

root_mysql> grant select, insert on cdacmumbai.v1 to scott@localhost;

scott_mysql> insert into cdacmumbai.v1 values(6, ‘F’);

- DML operations can be performed on a view
- DML operations done on a view will affect the base table
- Constraints that are specified on the table will enforced even whenn you insert via view.
- ENTIRE APPLICATION IS BASED ON VIEWS

Dropping a View:

amit_mysql> drop view v1;

create view v2

as

select * from emp where deptno = 1;

select * from v2;

| EMPNO | ENAME | SAL | DEPTNO |
|---|---|---|---|
| 1 | A | 5000 | 1 |
| 2 | B | 6000 | 1 |
| 3 | C | 7000 | 1 |

- used to restrict the row access

insert into v2 values(6, 'F', 6000, 2);  ←It will Allow

create view v2

as

select * from emp where deptno = 1 with check option;

select * from v2;

insert into v2 values(6, 'F', 6000, 2);  ←ERROR

- view with check option is similar to check constraint
- used to enforce different checks for different users

To change the SELECT statement of the view:

drop view v1;

create view v1 as .....;

create or replace view v1

as

select ename, sal from emp;

---

show tables; ← will show tables and views but it won;t tell you which is a table and which is a view

---

To find out which is a table and which is a view:

show full tables;

desc emp;

desc v1;

---

create or replace view v1

as

select upper(ename), “u_ename”  sal*12 “Annual” from emp;

select * from v1;

- view based on computed column, expression, function, order by clause, group by clause, etc.
- you can SELECT from this view
- DML operations are not allowed
- common for all RDBMS

---

create or replace view v1

as

select dname, ename from emp, dept

where dept.deptno = emp.deptno;

- view based on join, sub-query, UNION, etc.
- you can SELECT from this view
- DML operations are not allowed

- common for all RDBMS

---

To see SELECT statement on which the view is based:

show create view v1;

v1 = select .....

v2 = select.....

---

- view based on view is allowed
- uses:
  - To exceed the limits of SQL:
    - UNION > 255 statements
    - sub-query > 255 levels
    - function within function >255 levels
  - To simplify the writing of complex SELECT statements
    - Join of 20 Tables
  - complex queries can be stored in view definition

---