

# dbms (database management system)

- It is a software design to store, retrieve, define & manage the data in database.

# types —

- 1) RDBMS
- 2) hierarchical dbms
- 3) network dbms.

# RDBMS — (Relational dbms)

- Those database which can store its data in the form of tables. called as rdbms

column — attribute / schema

row — tuple / record

# Database - database is the collection of organized tables. called as database

(means database contain the number of tables or more than one tables).

\* IF we want to work on rdbms database we have one language name as SQL

**SQL**

- Structured query Lang.
- SQL works with relational database
- Using SQL we can perform database operations  $\Rightarrow$  like as :- (insertion, deletion etc, view etc)

# IF we want to perform SQL operation we need to use some external softwares those are working with dbms

# external software  $\Rightarrow$  MySQL, Oracle, db2 etc



If we want to work with sql we need to install mysql database software.]

If we want to work with sql we have 4 types of lang/command

#### ① DDL :- (Data definition Lang)

this type work on table structure like as [1) create table  
2) table drop  
3) alter or modify the table]

#### ② DML :- Data manipulation Lang is used for to perform [i) insert ii) delete iii) update iv) select etc. operations]

#### ③ DCL :- (Data control lang)

It is used for, to give the privileges to user or administrative command like as grant etc.

#### ④ TCL :- Transaction control language commit, revoke etc

x How to work with database.

1) install mysql

5) create your own

2) open mysql

database.

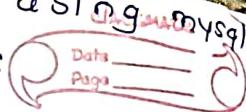
enter password  
3) login mysql

6) use your own database

4) use default of mysql

7) create table in default database to us

(4) If we want to enter in database using MySQL  
we have SQL the command name as  
e.g. use database name [use mysql]



(5) Create your own database -

If we want to create your own database we have command name as

create database database-name

e.g. create database mysql onlineexam

- use onlineexam

(6) Use your own database :-

use database name

e.g. use onlineexam.

(7) Create the table in database :-

If we want to store your information permanently in your database we need to create its table.

# How to create table in database :-

We have command

create table tablename(columnname datatype  
(size), columnname datatype(size));

mysql> create table question (qid int(5), question varchar(200),  
option1 varchar(200), option2 varchar(200),  
option3 varchar(200), option4 varchar(200)  
answer varchar(200));

mysql> insert into question values (1, 'What is OOP',  
'OOP is concept', 'abc', 'mno', 'pqr', 'oop is concept');

mysql>

qid

qid question

1 what is

# Data

3 Type

① char(

length)

and

⇒ It's

②

mysq> select \* from question;

qid question option1 option2 option3 option4

id	question	options	option2	option3	option4	answer
1	what is oop	oop is concept	abc	mno	pqr	oop is concept

## # Datatypes in SQL :-

3 Types :-

① string datatype

② numeric datatype

③ date and time types

① char(size) :- It is used for specify fixed length string that contain number, letter and specified characters.

⇒ It's size can be 0 to 255.

② varchar(size) :- It is used to specify the variable length of string that contain numbers, letter and special character

varchar(200), ⇒ It's size can be 0 to 65535

rechar(200),

archar(200),

③ binary(size) :- It is equal with char, but

store binary byte string it size parameter specify the column length in bytes-

**text (size)** :- It hold string that contain maximum length.

e.g - If we want to store the text document file in database.

**blob (size)** :- It is used for binary long object. It can hold up to 65535.

e.g - If we want to store image in database then we can use the blob.

**int (size)** :- It is used to store integer value. (range -2147483648 to 2147483647)

**float (size)** :- It is used for specify floating point number.

**parameter** :-

**size** : parameter specify number of digit

**d** : parameter specify number of digit after point.

**Date** :- It is used for specify the date format.

yyyy-mm-dd support range from

1000-01-01 to 9999-12-31

**DATETIME** :- It is used for specify date and time combination.

The format is yyyy-mm-dd hh:mm:ss

range - (1000-01-01 00:00:00) to (9999-12-31 23:59:59)

## Timestamp

→ used for specify time stamp  
It value store as number of second since unix.

1970-01-01 00:00:00 IST etc.

## Operators in SQL

3 Types of operator in SQL :-

### ① Arithmetic Operator

+ - \* / %

### ② SQL Comparison Operator

= equal

!= not equal

< check operand values equal or not if values are not equal then

> greater than

< less than

<= less than equal

>= greater than equal.

!< check left value is not less than right value.

>! check left value is greater than right value.

### ③ SQL Logical Operator

Logical Operator → combine more than one condition & check it.

**ALL** → compare all values to another value set

**AND** → combine more than one cond'n and check

**ANY** → if all cond'n are true then cond'n is true otherwise false.

**ANY** → this operator is used to compare value in list according to condn.

**between** → used for search values within set of values.

**IN** → the operator compare value in list value.

**not** → used for reverse meaning of any logical operator.

**exists** → used to search present row in specified table.

**like** → used for search patterns.

**or** → It is used for combine more than one condns and check it if any condn is true then condn is true otherwise condn is false.

# **DDL** Operations with database — i) create ii) alter iii) describe

⇒ Data definition lang work with table structure.

\* using ddl we can perform following actions with database :-

① **creating table** :-

before that we will see how to create database -

create database databasename;

e.g

= create database revise;

⇒ Once we create database and if we want to create the tables in database we need to use database ;  
database use means we need to enter database

→ If we want to use database we have command  
use database;

classmate

eg use reverse;

once we enter in database then we can work with database.

If we want to create table in database we have command like as :-

→ create table tablename(columnname datatype(size),  
columnname datatype(size));

(table name as register)  
name, email, contact

(create table register(name varchar(200), email varchar(200) contact varchar(200));

eg →

## ② alter table :-

(modify the table, add new column in table,  
drop column table etc)

⇒ alter table command can change the table structure.

⇒ How to modify column using alter command

Syntax :-

alter table tablename modify columnname  
column type

eg :- In our example email having 200 character size but we want to change it from 200 to 100

Mysql > alter table register modify email varchar(100);

# How to Add the column using alter command  
we have command :-  
alter table tablename add (column datatype(size),  
column datatype(size));

- \* as per our ex. we want to add the address of student in table

e.g. alter table register add (address varchar(200))

# How to modify column name using alter command  
we use rename option

Syntax :-

alter table tablename rename column oldcolumn to newcolumn name

e.g. in our table we have the email column just we want to change useremail

Example :-

mysql > alter table register rename column email to useremail;

# How to Drop the column using alter command.  
we have command like as :-

⇒ alter table tablename drop columnname;

mysql > alter table register drop address;

xox xox xox xox xox

③ Describe the table, :- if we want to see the

table structure means see the column name its datatype, its size etc.

for that we have command `desc`

Syntax :- `desc tablename`

or

`describe tablename`

e.g `desc register;`

#### ④ Drop the table :-

IF we want to delete the table from database we can drop it for that we have command name as -

`drop table tablename`

e.g `drop table register;`

xox — xox — xox — xox — xox

### < DML command >

- data manipulation lang.
- using this command we can perform the following operation with database table.

- =>
- i) `insert`
  - ii) `select`
  - iii) `delete`
  - iv) `update`

#### ① Insert :- `insert` command is used to insert record in database table.

### Syntax

`insert into tablename values(column1, column2, column3 ... column)`

e.g :-

`mysql> insert into register values('ram', 'ram@gmail.com', '1234567891');`

mysql> insert into register values('shyam', 'shyam@gmail.com', '45545');

\* sometime if we want to insert specified column in database table.

Syntax :- insert into tablename(column1, column2)  
values(value for column1, value for  
column2)

e.g :-

mysql> insert into register(name, contact) values('raju', '343434');

② **Select** :- this is used for fetch or retrieve data from database table.

Syntax -

select \* from tablename;

\* fetch all column value from database table.

mysql> select \* from register;

name	email	contact
ram	ram@gmail.com	12345678
shyam	shyam@gmail.com	9101112
raju	NULL	343434

③ **Delete** :- used for delete record from database table.

Syntax :-

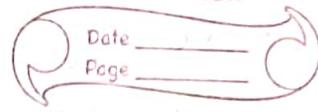
delete from register

If we want to delete the particular record we have command like as:

⇒ delete from tablename where columnname = value;

delete from register where email = 'ram@gmail.com'

name	email	contact
shyam	shyam@gmail.com	343434
rajesh	NULL	356789



④ **update** :-

IF we want to modify the column values in database table we have the update command.

Syntax :-

update tablename set columnname=value,  
columnname=value where <sup>column</sup> name = value;

e.g.  
mysql > update register set email = 'rajesh@gmail.com'  
where name = 'rajesh';

mysql > select \* from register

name	email	contact
shyam	shyam@gmail.com	45455
rajesh	rajesh@gmail.com	343434

## # Constraint in SQL #

⇒ SQL constraint are rules used to limit the type of data that can go in to table or

or  
=

⇒ SQL constraint are used to specify of database tables.

### Types of constraint

① Not Null :- By default column can hold null values if user not provide it.

If you do not want to a column have null value use null constraint.

i) It restrict column having null value.

ii) we use the alter or modify statement ; specify this constraint.

### Syntax :-

```
create table tablename (columnname datatype  
                      (size) not null, columnname datatype  
                      (size) not null)
```

### e.g :-

```
mysql > create table player (name varchar(200)  
                           not null, matches int(5), runs int(5));
```

### ② Primary key - constraint

⇒ primary key constraint uniquely identify each record in database

⇒ primary key must be contain unique values & not null value.

- imp**
- ① primary key not accept null value.
  - ② primary cannot be duplicated in table
  - ③ more than one column cannot mark as primary key
  - ④ primary key column may be auto increment.

- How we can create primary key constraint?

Syntax :-

create table tablename (columnname datatype(size)  
primary key, columnname datatype(size));

eg :-

mysql> create table player (pid int(5) primary  
key, name varchar(200), runs  
int(5), matches int(5));

\* How to create primary key with auto increment

value :-

Syntax :- create table tablename (columnname data-  
type(size) primary key auto\_in-  
crement, columnname datatype(size));

eg :-

create table employee (ep1 eid int(5) primary key  
auto\_increment, name varchar(200)  
email varchar(200), sal int(5));

\* When we have auto-incremented primary key  
value then we need to specify the primary  
key column value manually.

System will generate the primary automatically.

\* When we insert primary key value with auto-  
incrementation then we need to specify the  
'0' in database column where we want to  
insert value.

Example :- (auto-increment)

mysql> insert into employee values ('0', 'ganesh',  
'ganesh@gmail.com', 2000);

classmate  
Date

mysql> insert into employee values ('0', 'manesh',  
'manesh@gmail.com', 20000);

mysql> select \* from employee;

eid	name	email	sal
1	ganesh	ganesh@gmail.com	20000
2	manesh	manesh@gmail.com	20000

-----xox-----xox-----xox-----xox-----

③

Unique constraint :-

- It ensures that column will only have unique value.
- unique constraint field cannot have any duplicated data.

# unique constraint prevent two record having identical values in column.

Syntax :-

Create table tablename(columnname datatype unique,  
columnname datatype);

mysql> create table employee (eid int(5) primary  
key auto\_increment, name varchar(200),  
email varchar(200) not null unique, sal int(5));

mysql> commit;

mysql> insert into employee values ('0', 'ram', 'ram@gmail.com',  
10000);

```
mysql> insert into employee values('0','shyam','shyam@gmail.com',  
mysql> insert into employee values('0','dinesh','shyam@gmail.com', 20000);
```

error

⇒ Duplicate entry 'shyam@gmail.com' for key  
'employee.email'

DIFF<sup>n</sup> :-

primary key constraint	unique key constraint
① more than one column cannot declare as primary	① more than one column can have unique key
② select using primary key can create cluster index.	② select using unique key can create non-cluster index.
③ primary key is uniquely identifier for row in table.	③ unique constraint is uniquely identify for row in table if primary key is not present.
④ primary key cannot accept the null value	④ unique key can accept null value

④ Foreign key constraint :-

⇒ foreign key constraint is used for provide relationship bet<sup>n</sup> two tables.

⇒ primary key of one table work as foreign key in another table.

why we need to relate two tables with each other

\* discuss in this point in normalization

## How to create Foreign key constraint in table

syntax :-

```
create table tablename( columnname datatype(size), column  
name datatype(size), references (columnname),  
foreign key tablename (primary key));
```

```
create table course( cid int(5) primary key auto_  
increment , cname varchar(200),  
Fees int(5));
```

```
create table student ( sid int(5) primary key auto_  
increment , name varchar(200),  
email varchar(200) not null unique,  
contact varchar(200) not null unique , cid  
int(5) , references (cid) foreign key course(cid),  
Foreign key (cid) references course (cid));
```

## Behaviour of Foreign key constraint

⇒ Two ways to maintain the integrity of data in child table.

⇒ When a particular record is deleted in main table when two tables connected with Foreign key and certain data in main table is deleted for which record exist in child table then must have some mechanism to solve the integrity constraint.

e.g %

### pk course

cid	name	fees
1	Java	20000
2	PHP	20000

main table

Date \_\_\_\_\_  
Page \_\_\_\_\_

### pk of student

sid	name	email	contact	cid	fk
1	ram	ram@	14545	1	child table
2	shyam	shyam@	35348	1	
3	dinesh	dinesh@	231843	2	

# delete from course where cid=2 || It is not possible . course cannot be delete bcz cid use by student table , so it is not possible to delete the course table bcz cid refer by some other tables.

so if we want to delete course info from course table.

before that we need to delete the record from student table associated with course.

e.g

delete student from cid=1;

delete course from cid=1;

**Note:-** It is not possible in real time app to delete manually entry from every child.

for that we have :-

- ① on delete cascade
- ② on delete null

on delete cascade - this will remove the record from child table if table value of foreign key from main table.

on delete null -

this will be set all the values in the record of the child table as null for which the value of foreign key is deleted from main table.

# How to apply the on delete cascade practically with Foreign key

Syntax :-

```
create table childtablename(columnname datatype(size),  
columnname datatype(size), foreign key (keyname),  
references parenttablename (reference column  
primary key) on delete cascade);
```

How to apply on delete null with Foreign key,

e.g :- create table course (cid int(5) primary key  
auto-increment, name varchar(200), fees int  
(5));

mysql > create table student (sid int(5) primary  
key auto-increment, name varchar(200),  
email varchar(200), contact varchar(200),  
cid int(5), foreign key (cid) references  
course(cid) on delete cascade)

mysql > desc student;

mysql > insert into course values ('0', 'java', 2000);

mysql > insert into course values ('0', 'ram@', '12345');

mysql > insert into student values ('0', 'shyam', 'shyam@', 56);

mysql> commit;  
mysql> select \* from course;

cid	name	fees
1	java	20000

mysql> select \* from student;

sid	name	email	contact	cid
1	ram	ram@gmail.com	12345	1
2	shyam	shyam@gmail.com	454545	1

mysql> delete from course where cid=1;

mysql> select \* from course

Empty set (0.00 sec)

mysql> select \* from student

Empty set (0.00 sec)

on delete Null

mysql> create table course (cid int(5) primary key auto\_increment, name varchar(200), fees int(5));

mysql> create table student (sid int(5) primary key auto\_increment, name varchar(200), fees int(5), email varchar(200), contact varchar(200), cid int(5), foreign key (cid) references course (cid) on delete set null;

mysql> desc course;

Field	Type	Null	Key	Default	Extra
cid	int	NO	PRI	NULL	auto-increment
name	varchar(200)	YES		NULL	
fees	int	YES		NULL	

mysql> desc student;

field	Type	Null	key	Default	Extra
sid	int	No	PRI	NULL	auto-increment
name	varchar(200)	Yes		NULL	
email	varchar(200)	Yes		NULL	
contact	varchar(200)	Yes		NULL	
cid	int	Yes	MUL	NULL	

mysql> insert into course values ('0', 'Java', 20000);

mysql> insert into student values ('0', 'ram', 'ram@', '5555', 1);

mysql> insert into student values ('0', 'shyam', 'shyam@', '4545', 1);

mysql> insert into student values ('0', 'ghanshyam', 'ghanshyam@', '3333', 1);

mysql> select \* from student;

sid	name	email	contact	cid
1	ram	ram@	5555	1
2	shyam	shyam@	4545	1
3	ghanshyam	ghanshyam@	3333	1

mysql> delete select \* from course;

cid	name	fees
1	Java	20000

mysql> delete from course where cid=1;

mysql> select \* from student;

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

sid	name	email	contact	cic
1	ram	ram@	6555	NULL
2	shyam	shyam@	4565	NULL
3	ghanshyam	ghanshyam@	3333	NULL

(S)

### check constraint

- check constraint is used to restrict the value of a column bet^n range.
- It is like as cond'n before saving data into the column.

Syntax :-

```
create table tablename(columnname datatype (size) check  
                           (condition), columnname datatype  
(size) check(condition));
```

e.g :-

```
mysql> create table employee (eid int(5) primary key auto-  
increment, name varchar(200), email  
varchar(200), contact varchar(200), sal int  
(5) not null check (sal > 0));
```

mysql> desc employee;

```
mysql> insert into employee values ('0', 'shyam',  
'shyam@', '12345', -10000);
```

[It will give error]  $\Rightarrow$  check constraint 'employee-chk-1' is violated

## default constraint

- default constraint is used to set the default value for column.
- default value will be added to all new records if no other value is specified.

syntax :-

```
create table tablename ( columnname datatype (size),  
columnname datatype (size) default value
```

example :-

```
mysql > create table employee ( eid int (5)  
primary key auto_increment , name  
varchar(200) , email varchar(200) , contact  
varchar(200) , sal int (5) default 5000 );
```

```
mysql > insert into employee values ('0' , 'ram' , 'ram@' , '12345'  
10,000);
```

```
mysql > insert into employee values ('0' , 'shyam'  
'shyam@' , '4545');
```

```
mysql > insert into employee ( eid , name , email , contact )  
values ('0' , 'shyam' , 'shyam@' , 12345 );
```

```
mysql > select * from employee;
```

eid	name	email	contact	sal	
1	ram	ram@	12345	10,000	
2	shyam	shyam@	12345	5000	default set value 5000