* Constraints (keys)
  + Primary key:

A primary key is a Column of table which uniquely identify each row in that table.

Primary key does not accept any duplicate and null values.

The primary key value in a table change very rarely, so it is chosen with care.

A primary key of one table can be referenced by foreign key of another table.

* + Unique key:

Unique key also identifies an individual table uniquely in a relation or table.

Unique constrains are also referenced by the foreign key of another table.

* + Foreign key:

When one table’s primary key is field is added to related many tables to order to create the common field which relates the two tables, it Is called foreign key to table.

Foreign key constrain is used to prevent actions that would destroy links between two tables.

Foreign key is a field in one table that refers to the primary key in another table.

DDL (Data Definition language):

1 Create: it creates new table / database

Syntax 🡪 create database database\_name;

Example 🡪 create database Kashyap;

2 Use: it allows to be in the particular batabase

Syntax 🡪 use database\_name

Example 🡪 use Kashyap;

3 Drop: it is used to delete the database

Syntax 🡪 drop database database\_name;

Example 🡪 drop database Kashyap;

Tables 🡪 Data are stored in the table

Create – it is used to make table

* Table Create
  + syntax

**Create table table\_name(Column0 datatype auto\_increment,column1,2,3,..,**

**Column from anothrt table which is going to be foreign key,**

**Primary key (column0),**

**Foreign key(column from another table) REFERENCE another table**

**(column from another table));**

Syntax 🡪 Create table table\_name(

strudentID int **auto\_increment**,

student\_name varchar(20),

city varchar(10),

age int,

**primary key(studentID)**);

Delete Table – to delete the entire table

Syntax🡪 drop table table\_name;

* Commands:
* --ALTER--

1. Alter 🡪 to add column in the table
   * Alter table table\_name add new\_column name;
2. Rename 🡪 to change the name of the table
   * Alter table table\_name rename new\_name; (for rename table)
3. Drop 🡪 to delete column in table
   * Alter table table\_name drop column name; (to delete column in table)

* --INSERT--

Syntax🡪Insert into table\_name(column\_name) values(value1,value2,value,3);

* **To add multiple values**

Syntax 🡪 INSERT INTO table\_name(column 1, column 2,column 3, column 4)VALUES

("Kashyap","surat",50000,"gujarat"),

("Kashish","vadodara",30000,"gujarat"),

("Shivam","Ahemedabad",90000,"gujarat");

Truncate –

it is used to clear the data in the table whereas the drop is used to delete the entire table from the database.

* + Syntax 🡪 truncate table table\_name;

Update –

it is used to update the value in the table.(first column is what we want to update and second column is reference colmn.)

* + Syntax 🡪 update table\_name set column\_name=value where column\_name=value;

Delete statement—

it is used to delete particular statement (in block wise)

* + Syntax 🡪 delete from table\_name where (condition)

Select – It is used to show entire table

* + Syntax 🡪 select \* from table\_name; (to select entire table)
  + Syntax 🡪 select \* from table\_name where column\_name=value; (it is used to show particular values in block)

Select Distinct –

it is used to show the values in particular column and do not show duplicate value

* Syntax 🡪 select DISTINCT Column1.column2... FROM Table\_name;

Select distinct with condition- the where clause is used to filter records.it is used to extract only those records that fullfill a specified condition

* Syntax 🡪 select Column1, column2… FROM table\_name WHERE condition;

**AND OR & NOT operator –**

* The where clause can be combined with AND, OR & NOT operators
* The AND & OR operator are used to filter records based on more than one condition.
* AND operator Display record, if all the conditions separated by AND are true.
  + Syntax 🡪 select \* FROM table\_name WHERE condition1 AND conditon2 AND..;
  + Syntax 🡪 select \* column1, column2… FROM table\_name WHERE condition1 AND conditon2 AND..;
  + IT shows if both the values are true in particular column.
* OR operator display record, if any of the condition separated by OR are true.
  + Syntax 🡪 select \* FROM table\_name WHERE condition1 OR conditon2 OR..;
  + Syntax 🡪 select \* column1, column2… FROM table\_name WHERE condition1 OR conditon2 OR..;
  + IT shows if either the values are true in particular column and show everything about it.
* NOT operator display record, if the condition is not true.
  + Syntax 🡪 Select \*~~column1,counm2~~, FROM table\_name where NOT condition
  + Select \* FROM Table\_name WHERE NOT column\_name = value
* AND and NOT – it will not give values from both column with both value
  + Syntax🡪select \* FROM Table\_name WHERE NOT Column\_name1=value AND NOT Column\_name2=value;

IN Operator—

it is Allows you to specify multiple values in where clause.IN Operator is short hand for multiple OR condition.

* + Syntax 🡪 Select column\_name1, coulmn2.. FROM Table\_name WHERE column\_name1 IN (value1,value2,..) AND column\_name2 IN (value1,value2);
  + Syntax 🡪 Select \* FROM Table\_name WHERE column\_name IN (Value1,value2..);

NOT IN Operator

* Syntax 🡪 Select \* FROM table\_name WHERE column\_name NOT IN (value1,value2..);

BETWEEN –

the between operator selects values within a given range. The values can be numbers,text or dates.the between operator is inclusive: begin AND end value are included

* Syntax 🡪select column\_name FROM table\_name where column\_name BETWEEN value1 AND value2;

BETWEEN NOT – it uses the to display the product outside the range of the previous example.

ORDER BY -- the order by keyword is used to sort the results at in ascending and descending order.

The order by keywork sorts the records in ascending order by default.

To sort the record in descending order, use the DESC keyword.

* Syntax 🡪 select column\_name1,column\_name2 FROM Table\_name

ORDER BY column\_name1,column\_name2… ACS|DESC.

Group By –

The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country".

GROUP BY Syntax

SELECT *column\_name(s)*  
FROM *table\_name*  
WHERE *condition*  
GROUP BY *column\_name(s)*ORDER BY *column\_name(s);*

**SQL Statement types**

Avg()- average

Syntax 🡪 select avg(column\_name) FROM table\_name;

Count()- count the rows

Syntax 🡪 select count(column\_name) FROM table\_name;

First()- give first data

Syntax 🡪 select First(column\_name) FROM table\_name;

Last()- give last data

Syntax 🡪 select last(column\_name) FROM table\_name;

Max()- give max from column

Syntax 🡪 Select max(column\_name) FROM table\_name;

Min()-give min from column

Syntax 🡪 select min(column\_name) from table\_name;

* // Entity=Table

One to one Relationship (1:1) – One entity has a relationship with only one entity

One to Many (1:M)- one entity has a relationship with more than one entity.

Many to Many (M:M)- May entities has a relationship with Many entities.

* What is E-R Model?
* The entity relationship diagram represents relationship among entities in diagram.
* E-R diagram is structured design of the Database.
* E-R Model is a concise as a best practise to complete the ER modelling before implementing your database
* This diagram displays the relationship of the entity set stored in database, in short we can say that ER diagram helps your to explain the logical structure of database.

TCL – Transaction control

Commit- the commit command is transactional command used to save changes invoked by a transaction to the database.

Rollback- it is used to rollback or to undo the command saved to the database

This command is only be used to undo the transactions since the last commit or rollback command are issued.

Join-

* It is used to combine data or rows from two or more tables based on a common field between them.

Inner Join –

* it is most frequently used and important of the joints. It creates a new result table by combining column values of two tables (table1 and table 2) based on join predicate.

Join quarry compares each row of table1 with each row of table 2 to find all pairs of rows with satisfy the join-predicate. When the join-predicate satisfy column values of each matched pair of rows of A and B are combined into result row

* Syntax🡪 select columns from table1 inner join table2 on table1.column=table2.column

Procedure:

A stored procedure is a prepared SQL code that can save, so the code can be reused over and over again.so, if you have SQL quarry that you write over and over again then you can use procedure quary.

Alias

* SQL Alias are used to give a table, or a column name in a table temporary name
* Alias is often used to make column names more readable form. Alias are only exist for the duration of that quarry.
* Alias is created with the AS key word
* Column\_name 🡪 je column nu name temp change karavu che e
* Syntax 🡪 select column\_name AS alias\_name FROM table\_name;

Join

Join clause is used to combine row from two or more tables, based on a related column between them

1. Inner join – left and right join
2. Outer join
3. Full join

Inner Join 🡪 Return records that have marching values in both tables.

Syntax 🡪 Select column\_name from table1 inner join table2 on table1.coumn\_name = table2.column\_name;

Syntax 🡪 select table1.column1,table1.column2,table2.colum1 from table1 inner join table2 on table1.matching column=table2.matching column;

Table1: first table.

table2: second table.

LEFT JOIN 🡪 left join keyword returns all records from left table(table1) and the matching records from the right table(table2), the result us 0 records from the right table, if there is no match.

Syntax 🡪 select column\_name from table\_name left join table2 on table1.column\_name = table2.column\_name;

Syntax 🡪 select table1.column1,table1.column2, table2,column1 from table1 left join table2

On table1.matching column=table2.matching column;

Table1: first table.

table2: second table.

RIGHT JOIN 🡪 left join keyword returns all records from right table(table1) and the matching records from the left table(table2), the result us 0 records from the left table, if there is no match.

Syntax 🡪 select column\_name from table\_name right join table2 on table1.column\_name = table2.column\_name;

Syntax 🡪 select table1.column1,table1.column2, table2,column1 from table1 right join table2 On table1.matching column=table2.matching column;

Table1: first table.

table2: second table.

FULL JOIN 🡪 returns all records when there is a match in either left or right table.

Syntax 🡪 select column\_name1 from table1 FULL OUTER join table2 on table1.column\_name=table2.column\_name where condition;

Syntax 🡪 select table1.column1,table1.column2,table2.column1.. from table1 full join table2 on table1.matching\_column=table2.matching\_column;

Grant –

* it is enables system admistrators to assign to assign privilages and roles to the MySQL user so that they can use the assigned permission on the databse whenever required.

Syntax 🡪 Grant privilege\_name(s)

ON object

TO user\_account\_name;

* Privilege\_name(s) it specifies the access rights or grant privialge to user accounts. If we want to give multiple privilage, then use comma to separate.
* User\_account\_name: It determines the account name of the user from which we want to revoke the access rights

Revoke –

* it enables system administrators to revoke privileges and roles to the MySQL user account so that that cannot use the assigned permission on the database in the past.
* Syntax 🡪 Revoke privilege\_name(s)

ON object

FROM user\_account\_name;

Trigger –

* a Trigger in MySQL is a set of SQL statements that reside in a system catalog. IT IS A SPECIAL TYPE OF SOTRED PROCEDURE THAT IS INVOKED AUTOMETICALLY IN RESPONSE TO AN EVENT . Each trigger is associated with a table, which is activated any DML statements such as INSERT,UPDTAE or DELETE.
* Trigger helps us to validate data even before they inserted.
* Syntax 🡪 Create Trigger Trigger\_name trigger\_time Trigger event

On table name for each row

Begin

Variables declaration

Trigger code

End;

* MySQL client programme such as mySQL workbench used the delimiter(;) to separate statements and executes each statements and execute each satament speratly;
* Trigger name: which you want to give name of event
* Trigger time: Before/After
* Trigger Event: Insert/update/delete
* End: Stop the process
* Start: start the process
* Example:

Delimiter//

Create trigger TIG

Before insert ON course

For each ROW

BEGIN

If New.Totalfees<0 then set New.Totalfees=0;

End if;

End//