

Database

Collection of logically related data.

Data → Unprocessed form or collection of raw facts & figures.

Information → Processed form of data.

Meta Data → Data which represent another data is known as meta data. (Data about data)

M1	M2	M3	M4	Total	Avg.	H. Ticks
60	20	90	40			
50	60	20	80			
40	50	60	85			
80	40	80	90			

→ Record (Horizontal)

Field Field ⇒ Collection of same data content is known as field.

Record ⇒ Collection of different data fields.

Table ⇒ Collection of rows and columns in a systematic manner.

Functional Dependency

If one data value depends upon another data value is known as functional dependency.

Types

(i) Partial dependency

(ii) Transitive dependency

1 Partial dependency \Rightarrow Values which depends on Primary key values is known as partial dependency.

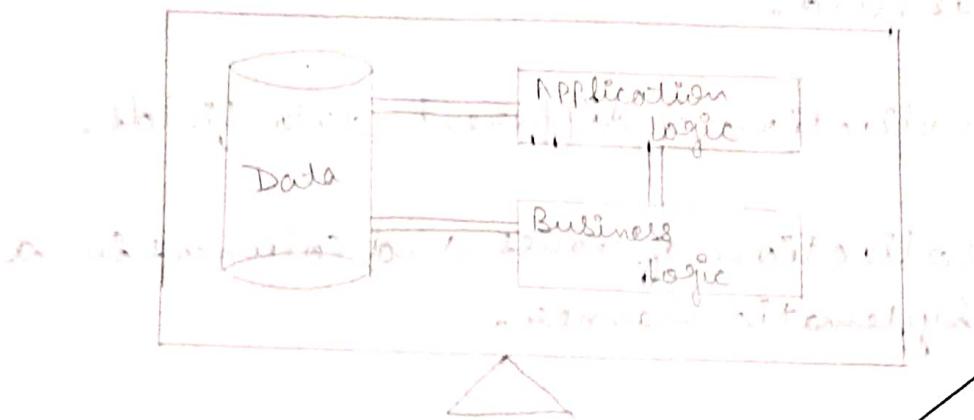
2 Transitive dependency \Rightarrow Non-primary key value dependent is known as Transitive dependency.

Primary key \rightarrow Unique Identification no. in a table.

Types Of Database

1 Centralised Database

2 Distributed Database



Business Logic \Rightarrow To fetch data from database

or store data and make changes in the database

Application Logic \Rightarrow Depends on Operating System

1 Centralised Database

Storing the data in the centralised place.

2 Distributed Database

Sharing data among different locations is known as distributed Database.

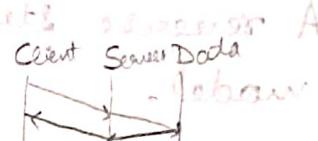
Disadvantage → Data Redundancy → Repetition of data

System Architecture

1 Centralised

2 File Server

3 Client Server



Client → requests → Application Logic

Server → response → Business Logic

Data Model

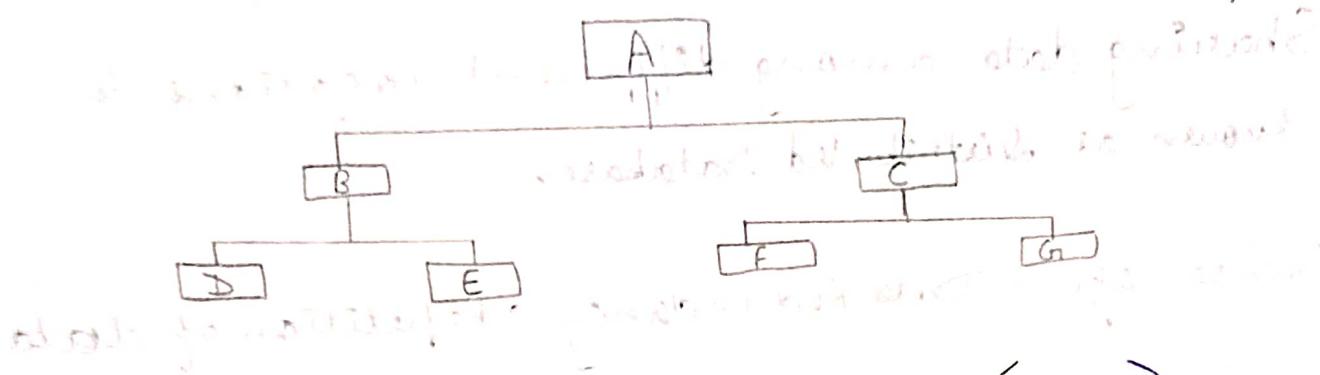
The Hierarchical / Graphical / Structural representation of data.

Types

- 1 Hierarchical data model
- 2 Network data model
- 3 ER (Entity Relationship) data model
- 4 Relational data model
- 5 Object Oriented data model

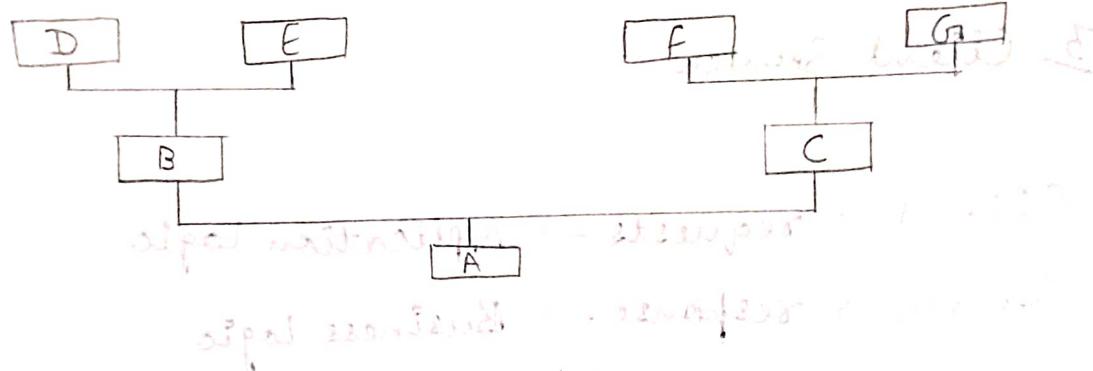
1 Hierarchical data model

Which contains a structural representation just like a tree, which begins with the root, contain the branches & leaves. leaf.



2 Network data model

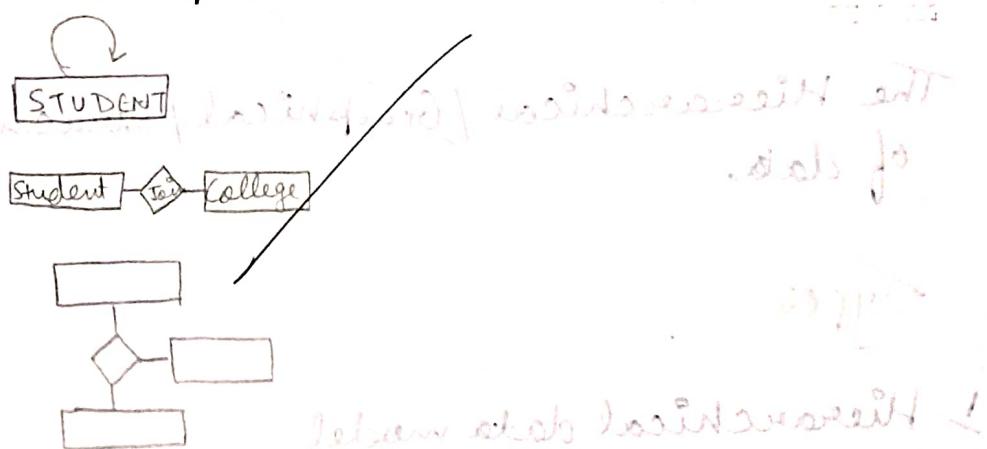
A reverse structure of the Hierarchical data model.



3 Entity Relationship data model

Classification

- 1 Unary
- 2 Binary
- 3 Ternary



Notes for ER Model:
 1. Unary: self-referencing
 2. Binary: between 2 entities
 3. Ternary: between 3 entities
 4. Unary: between 1 and 2 entities
 5. Binary: between 1 and 2 entities
 6. Ternary: between 3 entities

Types

Object Oriented Types

1 1:1

2 1:M; 1:N, 1:1, ... one to many, many to one

3 M:1

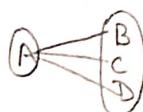
4 M:M

1 → for one
* → many

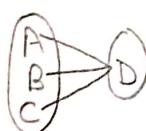
1 1:1



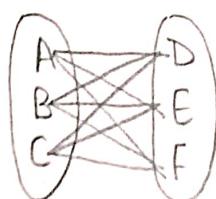
2 1:M



3 M:1



4 M:M



general purpose database systems

Relationship

Best communication path between two or more.

Relational Data Model

Relation between two or more table is known as Relational data model.

Object Oriented Data model.

It totally depends on Object Oriented Programming language i.e. Oop's Concept.

- 1 Class Collection of Data members & member fⁿ
- 2 Object
- 3 Inheritance
- 4 Polymorphism
- 5 Encapsulation
- 6 Message Communication
- 7 Abstraction

Database

My SQL
Open Source } Server side &
also Client side

S Q L (Structured Query Language) → Only Statement

P L S Q L (Procedural language Structured Query Language)

→ It contains Sub-query language.

1 DDL => Data definition language

Which defines the data of database table.

2 DML => Data Manipulation language

Steering

3 DCL => Data Control Language

Which controls the flow of data

4 TCL => Transaction Control language

Language which is mainly used for Transferring data.

Query Language

Which are going to be work on the specified Language (Question or request).

Normalization \Rightarrow No Duplication

works at Disjoint relation at repeating units of information

at the same context at repeating unit of information

• 3 normal forms

• primary key represents the end of relationship

• segment with respect of

ref (two) views or ref of been with

• stand that no turned

SQL

1 DDL Commands

- Create => It is used for creating a Database Table.
- Alter => It is used for Altering the Database Table.
- Drop => It is used for Deleting the Database Object.
- Truncate => It is used to Delete entire Database table with Auto Commit.
- Rename => To rename the Database Table Object.

2 DML Commands

- Select => To fetch the Database Content. } DQL Command
- Insert => To insert data into the database Table.
- Update => To Update the database Table Values.
- Delete => To delete the rows from the Database Table.

3 DCL Commands

- Grant => To assign privileges to Database Object to other users.
- Revoke => To Cancel the Privileges to Database object to other users.

4 TCL Commands

- Commit => To save the changes permanently.
- Roll back => To cancel the changes.
- Save Point => It is used for set a margin (Point) for Commit or Roll back.

8 Ch | 3 Ch

P.F.N	S.F.N
Primary File Name	Secondary File Name
E.F.N	Extension file Name

Computer.bat

Batch Processing => .bat

Syntax (Creating a Table)

Create table <table name> (Column 1 dt (size),

Column 2 dt (size), " " 2 " " ") ;

Example Create table Student (S.No. Number(10),

Name VarChar(20),

Address VarChar(10),

M2 VarChar(10),

Total VarChar(10));

SQL > desc Student ;

Insert values in a Database Table.

Insert is a command which is mainly used for inserting values into a database table.

We can insert a values into two ways.

Syntax

- 1 Insert into <filename> Values (number, "Cpu-name", 'C', ...);
- 2 Insert into <Table name> (<column name1>, <column2>, ...) Values (number, "Cpu-name", 'C', ...);

Example

- 1 Insert into student Values (20, "Drishti", "d@gmail.com", 9354574304);
- 2 Insert into student (Id, Name, Email, Mobile no.) Values (20, "Drishti", "d@gmail.com", 9354574304)

Select Command

→ This command is mainly used for fetching the content from the Database table.

In select command we can use conditions also by using clauses.

Block

Syntax

Keywords or reserved words

- 1 Select <Column name> * from <table name>;

- 2 Select <Column name> from <table name> where <cond.>;

* → for fetch total content

Alter (Add, remove, change)

6 ways

- Alter is a command which is mainly used for alter the structure of database object.
- That means we are going to increase/decrease/add/remove etc.

Syntax :-

1 Alter table <table name> [modify column name datatype(size)]
[add <col" name> datatype(size)] [drop col" col" name]

Case I To modify the length of an existing column of a table.

→ alter table Student modify RNo number(5);

Case II Add a new column to an existing table.

→ alter table Student add Avg number(5(2));

Case III Remove an existing column of a table.

→ alter table Student drop column Avg;

Case IV To add constraint on existing col" of a table.

The following query adds constraint Primary key to the existing attribute Rno of Student table.

→ alter table Student add constraint C1 primary key(R

Case V To remove constraint on existing column of a table.

The following query removes the constraint C1 of Student table.

→ alter table student drop constraint C1;

Case VI It is to enable/disable constraint on existing column of a table.

The following query enables/disables the constraint C1 of a Student table.

→ alter table student enable constraint C1;

→ alter table student disable constraint C1;

Update

This command is mainly used for update the values of attributes.

Syntax :-

Update <Table name> set colⁿ name = Value [colⁿ name = Value]
[where condⁿ];

Case I To set a salary i.e., 8,000 employee name Smith

→ Update emp set sal = 8,000 where e_name = 'Smith';

Case II To increase the salary of employee by 10%.

→ Update emp set sal = sal + sal * 0.10;

Case III To update value of multiple attributes at a time

→ Update emp set sal=8000, comm=100 where job='Manager';

Drop

This statement is mainly used for to delete a database object.

Syntax: drop objecttype object-name ;

By using the drop command we can drop a table items, indexes & also views.

→ Drop table items ;

→ Drop index dname_idx ;

→ Drop view empview ;

(Table data &
structure both
are deleted)

Truncate

To delete a row with auto commit (not the table structure)

Syntax: Truncate Table <Table name> ;

To delete a complete information from a table

→ Truncate Table Student ;

Rename

This command is mainly used for renaming the Database object.

Syntax:- rename oldtablename to newtablename;

The following query renaming a table.

→ rename student to std

alter table oldtablename rename to newtablename

Delete

To delete a table rows from a Database table.

Syntax: delete from tablename [where cond];

To delete the records of a table

→ delete from emp where job = 'Manager';

To delete all records of a table

→ delete from std;

or

delete std;

Grant

To give Privileges on database objects to other users.

Syntax:

Grant privilege list/role on object to userlist/Public
[with grant option];

Case I

To grant all permissions on employee table to everyone.

→ Grant all on emp to Public;

Case II

To grant select, delete permission on department table to cnu.

→ Grant select, insert on dpt. to cnu;

Case III

To grant all permissions on emp table to kiran & kiran should be able to give further grants on emp to other user.

→ Grant all on emp to kiran with grant option;

Revoke

To cancel privilege on database objects to other users.

Syntax: Revoke Privilegelist/Role on object from user/Public;

To revoke update privilege from everybody on employ table.

→ Revoke update on emp from public;

To revoke all privileges on emp from cnu.

→ Revoke all on emp from cnu;

Commit

To make changes permanently

Syntax: Commit;

To delete the records of clerks & save the change on emp table.

→ delete from emp where Job = 'Clerk'

We will not be able to retrieve the records of clerk back.

Even if this is deleted, total staff

recorded in payroll file is not changed. No staff of other job is removed. Suppose old record is added

in payroll file with job as clerk no the staff

- access ratio of staff is deducted in updating record of

old staff. New staff brought in and total number of staff is added. Total number of staff is updated. Staff record of old staff is also updated.

Updating of staff is done by changing the status of
new staff from working to non-working. Staff record of
non-working staff will be stored in

idle (non-working) segment or show of
(turns) - temporary

segment and it needs to store all details of
old staff and no

Rele back

To cancel the changes (that means restore the data)

Syntax: Rele back;

To delete records of clerks & retrieve them back.

→ delete from emp where job = 'Clerk'

Rele back;

Save Point

To set a margin for Commit or releback.

Syntax: Save Point SavePoint name;

Delete from emp where job = 'Manager';

Save Point P1;

Delete from emp where job = 'Clerk';

Rele back to P1;

Commit;

We will be able to restore the clerk records back but not manager records because we have cancel the changes upto P1 only.

P1 is a save point set after deleting manager records.

Function \Rightarrow It is a specific task performed.

Type of Function

→ Pre-defined

(~~comes with standard library~~)

→ User-defined

Function

Pre-defined

with return type with Parameters [int add(a, b)]

User-defined

with return type without Parameters [int add ()]

without return type with Parameters [add (a, b)]

without return type without Parameters

Pre Pre-defined \Rightarrow Already existing functions.

User-defined \Rightarrow It is defined by user, depend up on the requirement.

Cardinality

The min. & max. instances associated with an entity.

My SQL Views

In My SQL view is a virtual table created by a query by joining one or more tables.

- 1 Create View \Rightarrow A view is created by select statements. Select statements are used to take data from the source table to make a view.

Syntax

~~Create table table name (Column1 data type (size),
Column2 data type (size),
Column3 data type (size));~~

Eg Create table emp(emp_id int (5), emp_name varchar(20),
DOB date, basic_sal int (8));

insert into emp values (101, 'Drishti', '2000-09-24', 50000);

insert into emp values (102, 'Bhaati', '1999-09-13', 50000);

Syntax

Select * from table name ;

Select * from emp ;

Syntax

Create [OR replace] view, view name as

select column

from tables

[Where cond"] ;

Parameters

OR Replace \Rightarrow It is optional. It is used when a view already exists. If you don't specify this clause and the view already exists, the create view statement will return an error.

View Name => It specifies the name of the view that you want to create in My SQL where conditions. It is also optional. It specifies the conditions that must be met for the records to be included in the view.

Eg Create View trainer as

Select Course name, Course trainee
from Courses;

To see the Created View

Syntax Select * from View name;

Eg Select * from V-trainer;

Update View

In My SQL the alter view statement is used to modify or update the already created view without dropping it.

Syntax

alter View View_name as

select Columns

from table

where cond";

adding a new column to already created view name "trainer"

Eg alter view trainer as

select Course name, Course-trainee,
Course_id
from Courses;

To see the altered view

Select * from trainer;

My SQL drop View

We can drop the view by using drop view statement.

Syntax drop view [if exists] view-name

View name

It specifies the name of the view that you want to drop.

If exists

It is optional. If you do not specify this clause & the view does not exist, the drop view statement will return an error.

Eg `Drop view trainer;`

MySQL Queries

A list of commonly used MySQL queries to create database, use database, create table, insert record, update record, delete record, select record, truncate table & drop the table.

MySQL Create database

MySQL Create database is used for creating the database.

Syntax `Create database database name;`

Eg `Create database employee;`

MySQL Select or Use Database

Which is used for use the database.

Syntax

`Use database name;`

MySQL Create query

Which is mainly used for create a table, view, procedure & function.

Eg `Create table customers (Id int(10), name varchar(50), City varchar(50), Primary key (Id));`

My SQL alter query

My SQL alter query is used to add, modify, delete or drop columns of a table.

Eg

alter table customers add age Varchar(50);

My SQL insert query

This query is used to insert the records into a table.

Eg

insert into customers values (1223, 'anu', 'MNCI');

My SQL Update query

It is used to update records of a table.

Eg Update Customers set name = 'Hyd', City = 'BPL' where Id = 1223;

My SQL delete query

It is used to delete records of a table from a database.

Eg

delete from customers where Id = 1224.

My SQL Select query

It is used to fetch records from database.

Eg

Select * from Customers.

My SQL Truncate Table query

It is used to Truncate or remove records of a table.

It is used to remove the record of table not the structure.

Eg

Truncate Table Customers;

My SQL drop query

It is used to drop a table, view or database. It is used to remove the structure & data of the table.

Eg

Drop table Customers;

My SQL Insert Statement

Insert statement is used to insert data in My SQL table within the Database. We can insert single or multiple records using a single query.

The SQL insert into command is used to insert data in My SQL Table.

Syntax

Insert into Table name (field1, field2, fieldn)
Values (Value1, Value2,, Value n);

Syntax

Insert into Table name Values (Value1, Value2, ..., Value n);

If you have to store all field values either specify all field name or don't specify any field.

Eg

Insert into emp Values (23, 'Cru', 80000);

OR

Insert into emp (Id, name, salary)
Values (23, 'Cru', 80000);

Insert Eg 2 for Partial fields

It is mandatory to specify the field names.

Eg - 2

Insert into emp (id, name)
Values (7, 'Cru');

Eg - 3 Insert Multiple records

We are going to insert record in the customer table of a customer database.

Insert into Customer table (Customer_id, Customer-first-name, Customer-last-name)

Values

(1221, 'Sushma', 'Patil'), (1222, 'Vikas', 'Yadav'),
(1223, 'Anu', 'Rao');

My SQL Select command for the multiple table

Select statement can also be used to retrieve records from multiple tables by using joint statement.

Eg let take two table student & officer.

Student's table

fields:- Student-id, std-name, Course-name

Officer's table

fields:- off-id, off-name, address

Query

Select Officers.officer-id, student.Student-id
from student

Inner join Officers

On student.Student-id = officers.officer-id

Ordered by student.id;

06/09/2019

Where Clauses

In MySQL where clause is used with select, insert, update and delete. The clause is mainly used for filter the results & which specifies a specific position where you have to do the operation.

Syntax

Where "Cond";

Where Clause with single condition

Eg Select * from officers where address = 'MNCL';

Where Clause with AND Condition

By this we are retrieving data from the table officers with AND condn.

Eg Select * from officers where address = 'MNCL'
AND sal > 20000

Where Clause with Combination of AND & OR

We can use the AND & OR conditions all together with the where clause.

Eg Select * from officers where (address = 'MNCL' AND off-name = 'Cnu') OR (off-sal > 20000)

Where Clause with OR condition

Select * from officers where address = 'MNCL' OR
sal > 20000

Distinct Clause

Distinct clause is used to remove duplicate records from the table & fetch only unique records.

The distinct clause is only used with the select statement.

Syntax

Select distinct expressions

from tables

[where] cond";

Expression

Which specifies the columns or calculations that you want to retrieve.

Where Condition

It is optional. It specifies the conditions that must be met for the records to be selected.

Note

- If we put only one expression in the distinct clause. The query will return the unique values for that expression.
- If we put more than one expression in the distinct clause the query will retrieve unique combinations for the expressions listed.
- In MySQL the distinct clause doesn't ignore null values.
- So if you are using the distinct clause in your SQL statement your result set will include null as a distinct value.

Eg-1 Select distinct address from officers;

Eg-2 Select distinct_id, salary from officers;

Having Conditions

It is used to restrict the groups returned the rows. It shows only the groups in result set whose condⁿ are true.

Having Clauses with Sum function

Select emp name, Sum(Working hours) as "total working hours"
 from employee
 Group by emp name
 having sum(Working hours > 5);

MySQL like conditions

In MySQL like conditions used to perform match pattern to find the correct result it is used in select, insert, update & delete statement with condⁿ of where clauses.

Syntax Expression like Pattern [escape "Esc char"]

Pattern

Escape Characters => It is optional. It allows you to test for literal instances of a wild card character such as % or -. If we don't provide the escape character MySQL assumes that '\' is the escape character.

By Using % Wild Card

Select officers name
 from officer
 where address like "luck%" ;

Using - Wild Card

We are using the "-" for address

Select officers name
 from officer
 where address like 'luc-now' ;

Using NOT Operator => We can also use NOT operator with MySQL like condition. We can use %, wild card with NOT operator.

Select officer_name
from officers
where address not like 'luck%';

NOTE ⇒ In this ~~exp~~ example we can see the address is not like
luck%. ~~are~~ only shown.

19/09/2019

MySQL Joins

The MySQL Joins are used with Select statement. It is used to retrieve data from multiple tables. It is performed whenever you need to fetch records from two or more tables.

There are three types of MySQL Joins :-

- 1 MySQL Inner Join (Simple Join)
- 2 MySQL Left Outer Join (Left Join)
- 3 MySQL Right Outer Join (Right Join)

1 MySQL Simple Join

The MySQL inner join is used to return all rows from multiple tables where the join condition is satisfied. It is the most common type of Join.

Syntax

Select Columns

from table 1

inner join table 2

On table 1. Column = table 2. Column;

Eg

Consider two tables Officers & Students having the following data, by that project a inner join :-

Select Officers. Off-name, Officers. address, Students. Course_name
from Officers
inner join Students
On Officers. off-id = Students. Student-id;

2 MySQL Left Outer Join

The left Outer Join returns all rows from the left hand table specified in the ON condition & only those rows from the other table where the join condition is fulfilled.

Syntax

Select column
from table 1

left [Outer] Join table 2

ON table 1. Colⁿ = table 2. Colⁿ;

Eg

Select Officers.Off-name, Officers.address, Students.CourseName
from Officers

left Join Students

ON Officers.Off-id = Students.Std-id;

3 Right Outer Join

Which returns all rows from right hand table specified in the ON condⁿ & only those rows from the other table where join condⁿ is fulfilled.

Syntax

Select Colⁿ
from table 1

Right [Outer] Join Table 2

ON table 1. Colⁿ = table 2. Colⁿ;

Eg

Select Officers.Off-name, Officers.address, Students.CourseName
from Officers

Right Join Students

ON Officers.Off-id = Students.Std-id;

My SQL aggregate fⁿ

There are various aggregate fⁿ that can be used with select statement. A list of commonly used aggregate fⁿ is given below:-

Aggregate fⁿ

1 Count()

Description

Return the count of expⁿ.

2 Sum()

Which returns the total summed value of expⁿ.

3 Avg()

Returns the average value of expⁿ.

4 Min()

Returns the min. value of expⁿ.

5 Max()

Returns the max. value of expⁿ.

First → Which returns the first value of expⁿ.

Last → Which returns the last value of expⁿ.

MySQL Count f"

Syntax

```
Select Count (aggregate expn)
from table name
[Where] condn ;
```

Aggregate Expⁿ

It specifies the colⁿ or expⁿ those non null values will be counted.

Table Name

It specifies the table from where you want to retrieve records. There must be atleast one table listed in the from Clause.

Where Condⁿ

It is Optional.

Eg

```
Select Count (Off-name)
from Officers
where Off-add = 'MNCL' ;
```

Sum f"

```
Select Sum (aggregate fn)
from table name
[Where] condn ;
```

Eg

```
Select Sum (Working hours) as "total Working hours"
from employees
Where working hours > 5 ;
```

Average f"

```
Select Avg (aggregate expn)
from table name
[Where] condn ;
```

Eg

Select avg(working hours) as "avg working hours"
 from employees
 Where working hours > 5;

Min fn

Select min (aggregate fn)
 from table name;

Eg

Select min (working hours) as "min working hours"
 from employees;

Max fn

Select max (aggregate exp)
 from table name;

Eg

Select max(working hours) as "max working hours"
 from employees;

First fn

The MySQL first fn is used to return first value of the selected col. We use limit clause to select first record or more.

Syntax

Select Colⁿ name
 from table name
 limit 1;

Eg

Select Off-name
 from Officers
 limit 1;

If we want to select first two records the limit no. is going to be increased.

Eg

Select Off-name
 from Officers
 limit 2;

Last f"

Select Col" name
from table name
Order by Col" name desc
limit 1;

Eg

Select Off-name
from Officers
Order by Off-id desc
limit 1;