Day 1

Monday, 23 September 2024 9:48 pm

Data - raw information
Database is a server where we store retrieve manipulate data .
DBMS - Database Mgmt sys - store in file format - can small amount of data- cannot maintain relation - NoSQL-ex -MongoDB
RDBMS - Relational DBMS- Tabular -large data-maintain relations - SQL-ex-MySQL , ORACLE

If Oracle is paid ,why we use MySQl -?

BigData - to process large amount of data - can store structure ,semi structure and unstructured
Ex - Haddop

Cloud - Azure and AWS

SQL is a language to work on MySQL.
Comments -
1.DDL- Data Definition Language -It affects the table structure- create , alter, drop
2.DML - Data Manipulation Language - it affects the data - insert ,update,delete
3.DQL - Data Query Language - select
4.TCL - Transaction Control Language - commit, save point, rollback
5.DCL - Data Control Language - Grand and Revoke

Day 2 - Basic Commands

Wednesday, 25 September 2024 3:27 pm

Int, double ,float
For string - varchar

And we need to give size;

create database dbname;
create database v418_mubarra;

Show databases;

Use databasename;
use v418_mubarra;

create table tablename(columnname datatype(size),columnname
datatype(size)----------N);
| | |
Keyword name of the table

create table Employee (id int,name varchar(20),salary double(8,2));

Double (8,2) means 100000.00

Show tables;

desc employee;

To add column
alter table tablename add newcolumnname datatype(size);

alter table employee add desig varchar(20);

To remove a column

alter table employee drop desig;
The ALTER TABLE statement is used to add, delete, or modify columns in an existing table

To change name of a table entry

alter table Employee change name empname varchar(20);

To change datatype
alter table Employee modify salary int;

Insert into Employee value(101,"mubarra",100000,"Developer"); ///need to give value to all the table columns compulsary

Insert into employee(id,name)value(102,"leesa");
The INSERT INTO statement is used to insert a new row in a table.

mysql> select * from employee where id = 101;
The SELECT statement is probably the most used SQL command. The SELECT statement is used for retrieving rows from the database and enables the selection of one or many rows or columns from one or many tables in the database

select * from employee where desig = NULL

mysql> select * from employee where salary between 9000 and 15000;

mysql> select * from employee where salary between 10000 and 15000;

mysql> create table dept1(select * from department where salary IN(10000,15000));

Day 3

Thursday, 26 September 2024 3:44 pm

Constraint (apply to column)
1] NOT NULL - compulsory value to be provided - it does not allow null value

2]Primary Key - it does not allow duplicate and NULL value (one table can contain only one primary key)

3] Unique Key - it does not allow duplicate value but allow NULL value.

What is the difference between unique key and primary key?

4] Foreign Key -

5] Check Key - if creating column on conditional basis (ex - if age<18 ---it should display not eligible)

Aggregate function- is a inbuilt function(min ,max,avg,count,sum,distinct)
select max (salary) from employee;

mysql> alter table emp_1 modify name varchar(20);
To remove NOT NULL from table;

mysql> alter table emp_1 modify name varchar(20) NOT NULL;
To add NOT NULL to particular column.

mysql> alter table emp_1 drop primary key;
To remove primary key.(because a table have only one primary key)

mysql> alter table emp_1 add primary key(id);
To add primary key to column id.

```
mysql> select * from employee limit 0,2;
+------+---------+-----------+-----------+
| id   | name    | salary    | desig     |
+------+---------+-----------+-----------+
|  101 | mubarra | 100000.00 | Developer |
|  102 | leesa   |      NULL | NULL      |
+------+---------+-----------+-----------+
2 rows in set (0.00 sec)
```
-->to fetch data from table where 0 index is first row. And display two row.

```
mysql> select * from employee order by salary;
+------+---------+-----------+-----------+
| id   | name    | salary    | desig     |
+------+---------+-----------+-----------+
|  102 | leesa   |      NULL | NULL      |
|  108 | Aniket  |  10000.00 | Designer  |
|  106 | Aniket  |  10000.00 | Designer  |
|  107 | Sayli   |  10000.00 | Designer  |
|  109 | Akash   |  15000.00 | Tester    |
|  110 | Radhika |  15000.00 | Tester    |
|  111 | Rahat   |  15000.00 | Tester    |
|  101 | mubarra | 100000.00 | Developer |
|  103 | Geeta   | 100000.00 | Developer |
|  104 | Neeta   | 100000.00 | Developer |
|  105 | Seeta   | 100000.00 | Developer |
+------+---------+-----------+-----------+
11 rows in set (0.00 sec)
```
Salary in ascending oder

mysql> select * from employee order by salary desc;
For descending

To get top 3 distinct salary
mysql> select distinct(salary) from employee order by salary desc limit 0,3;

```
+-----------+
| salary    |
+-----------+
| 100000.00 |
|  15000.00 |
|  10000.00 |
+-----------+
3 rows in set (0.00 sec)
```

Day - 4

Monday, 30 September 2024 3:18 pm

1.GroupBy -The GROUP BY clause in SQL is used to group rows that have the same values in specified columns into summary rows, like "counting the number of items in each category" or "calculating the average salary for each department." It's often used with aggregate functions like COUNT(), SUM(), AVG(), MIN(), and MAX() to perform calculations on each group of rows.
select desig, sum(salary) from Employee Group by desig
mysql> select desig, sum(salary) from Employee Group by desig;

2.having
mysql> select desig, sum(salary) from Employee Group by desig having sum(salary>50000);

3.scaler function - (uppercase ,lower case,)
mysql> select ucase(name)from employee;
mysql> select lcase(name)from employee;

4.like - (to find name start with s,inbetween s and ends with s)
mysql> select * from employee where name like "%s"; //ends with s
mysql> select * from employee where name like "s%"; //starts with s
mysql> select * from employee where name like "%s%"; //s comes in between s
mysql> select * from employee where name like "l___a";
mysql> select * from employee where name like "__e_a";

5. To create copy of a table - nested query (means query inside query)
   create table stdu _copy(select * from student);
   create table stdu _copy(select * from student);

   When we use nested query?
   To make copy

Day 5

Tuesday, 1 October 2024 3:41 pm

Revsion -
Select * from sales where sid = (select* from employee where name = "Ramesh");

Select * from sales where sid = (select sid from employee where salary=(select max (salary) from employee);

UPDATE :
update employee set salary = salary+3000 where desig = "Tester";

DELETE:
Delete from employee where name = "Aniket";

Day 6

Thursday, 3 October 2024 3:28 pm

Foreign key:
Why we use foreign key -only the data in parent table is used in child table .now new data is allowed.

While deleting -we need to delete data first from child table then from parent table;
But while inserting first add data in parent table then in child table …otherwise it will throw error.

We cannot direcctly delete foreign key like we do incase of primary key.
To delete foreign key we need to use constraint while declaring it:
Ex-

```
create table book_1(bookid int primary key, bookname varchar(20),authorname varchar(20),price double(8,2),pid_fk int,constraint fk foreign key(pid_fk)references publisher(pid));
```

Day 7

Friday, 4 October 2024 3:11 pm

Unique key:
create table emp_2(id int unique key,name varchar(20)not null,salary double(8,2)not null);
At id it can accept null as entry;
What is difference between unique key and primary key?
PK doesnot allow nulll value
PK can be uk but uk cant be pk
One table have only single pk.but multiple uk.
Pk is combination of uk and not null;

Alternate key;

mysql> alter table emp_3 add primary key name;
mysql> desc emp_3;

```
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| id     |         int | NO   | UNI | NULL    |       |
| name   | varchar(20) | NO   | PRI | NULL    |       |
| salary | double(8,2) | NO   |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
```

3 rows in set (0.00 sec)
Here id is a alternate key(unique key with not null other than primary key)

Check key;
create table vote (id int,name varchar(20),age int check(age>18));

View means virtual table;
Simple view - create view emp_view as select * from employee;
Complex view - create view std_view_1 as select * from student as s left join mark as m on s.sid=m.id;

To drop
drop view emp_view;

Exercise

Friday, 4 October 2024 4:40 pm

```
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.34 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use v418_mubarra;
Database changed
mysql> create table emp_2(id int unique key,name varchar(20)not null,salary double(8,2)not null);
Query OK, 0 rows affected, 1 warning (0.05 sec)

mysql> insert into emp_2 value(101,"Gita",10000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into emp_2 value(102,"babita",null);
ERROR 1048 (23000): Column 'salary' cannot be null
mysql> insert into emp_2(id,name) value(,"babita");
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near ',"babita")' at line 1
mysql> insert into emp_2(id,name) value('',"babita");
ERROR 1366 (HY000): Incorrect integer value: '' for column 'id' at row 1
mysql> insert into emp_2(id,name) value(null,"babita");
ERROR 1364 (HY000): Field 'salary' doesn't have a default value
mysql> insert into emp_2 value(null,"babita",13000);
Query OK, 1 row affected (0.00 sec)

mysql> select
   -> * from emp_2;
+------+--------+----------+
| id   | name   | salary   |
+------+--------+----------+
|  101 | Gita   | 10000.00 |
| NULL | babita | 13000.00 |
+------+--------+----------+
2 rows in set (0.00 sec)

mysql> create table emp_3(id int unique key not null,name varchar(20)not null,salary double(8,2)not
null);
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> desc emp_3;
```

```
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| id     | int         | NO   | PRI | NULL    |       |
| name   | varchar(20) | NO   |     | NULL    |       |
| salary | double(8,2) | NO   |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)

mysql> alter table emp_3 add primary key name;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near '' at line 1
mysql> alter table emp_3 add primary key (name);
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc emp_3;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| id     | int         | NO   | UNI | NULL    |       |
| name   | varchar(20) | NO   | PRI | NULL    |       |
| salary | double(8,2) | NO   |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> create table vote (id int,name varchar(20),age int check(age>18));
Query OK, 0 rows affected (0.02 sec)

mysql> desc vote;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int         | YES  |     | NULL    |       |
| name  | varchar(20) | YES  |     | NULL    |       |
| age   | int         | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> create table sports(id int,name varchar(30),age int check(age>18) and (age<25));
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near 'and (age<25))' at line 1
mysql> create table sports(id int,name varchar(30),age int check((age>18) and (age<25)));
Query OK, 0 rows affected (0.02 sec)

mysql> insert into sports(101,Mubarra Shaikh,22);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near '101,Mubarra Shaikh,22)' at line 1
mysql> insert into sports(101,"Mubarra Shaikh",22);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near '101,"Mubarra Shaikh",22)' at line 1
mysql> insert into sports value(101,"babita",19);
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into sports value(101,"babita",13);
ERROR 3819 (HY000): Check constraint 'sports_chk_1' is violated.
mysql>  create view emp_view as select * from employee;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from emp_view;
+------+---------+-----------+-----------+
| id   | name    | salary    | desig     |
+------+---------+-----------+-----------+
|  101 | mubarra | 300000.00 | Designer  |
|  102 | leesa   | 200000.00 | Accounts  |
|  103 | Gita    | 350000.00 | Developer |
|  104 | Neeta   | 150000.00 | Developer |
|  105 | Seeta   |  50000.00 | Developer |
|  106 | Akash   |  18000.00 | Accounts  |
|  107 | Radhika | 200000.00 | Tester    |
|  108 | Rahat   |  45000.00 | Tester    |
|  109 | Sayli   |  10000.00 | Designer  |
+------+---------+-----------+-----------+
9 rows in set (0.00 sec)

mysql> insert into employee value(110,"Natraj",30000,"IT");
Query OK, 1 row affected (0.00 sec)

mysql> select * from emp_view;
+------+---------+-----------+-----------+
| id   | name    | salary    | desig     |
+------+---------+-----------+-----------+
|  101 | mubarra | 300000.00 | Designer  |
|  102 | leesa   | 200000.00 | Accounts  |
|  103 | Gita    | 350000.00 | Developer |
|  104 | Neeta   | 150000.00 | Developer |
|  105 | Seeta   |  50000.00 | Developer |
|  106 | Akash   |  18000.00 | Accounts  |
|  107 | Radhika | 200000.00 | Tester    |
|  108 | Rahat   |  45000.00 | Tester    |
|  109 | Sayli   |  10000.00 | Designer  |
|  110 | Natraj  |  30000.00 | IT        |
+------+---------+-----------+-----------+
10 rows in set (0.00 sec)

mysql> delete from emp_view where name="Neeta";
Query OK, 1 row affected (0.01 sec)

mysql> select * from employee;
+------+---------+-----------+-----------+
| id   | name    | salary    | desig     |
+------+---------+-----------+-----------+
|  101 | mubarra | 300000.00 | Designer  |
|  102 | leesa   | 200000.00 | Accounts  |
|  103 | Gita    | 350000.00 | Developer |
|  105 | Seeta   |  50000.00 | Developer |
```

```
| 106 | Akash   |  18000.00 | Accounts  |
| 107 | Radhika | 200000.00 | Tester    |
| 108 | Rahat   |  45000.00 | Tester    |
| 109 | Sayli   |  10000.00 | Designer  |
| 110 | Natraj  |  30000.00 | IT        |
+------+---------+-----------+-----------+
9 rows in set (0.00 sec)

mysql> update employee set salary = 180000 where id=106;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from employee;
+------+---------+-----------+-----------+
| id   | name    | salary    | desig     |
+------+---------+-----------+-----------+
| 101 | mubarra | 300000.00 | Designer  |
| 102 | leesa   | 200000.00 | Accounts  |
| 103 | Gita    | 350000.00 | Developer |
| 105 | Seeta   |  50000.00 | Developer |
| 106 | Akash   | 180000.00 | Accounts  |
| 107 | Radhika | 200000.00 | Tester    |
| 108 | Rahat   |  45000.00 | Tester    |
| 109 | Sayli   |  10000.00 | Designer  |
| 110 | Natraj  |  30000.00 | IT        |
+------+---------+-----------+-----------+
9 rows in set (0.00 sec)

mysql> select * from emp_view;
+------+---------+-----------+-----------+
| id   | name    | salary    | desig     |
+------+---------+-----------+-----------+
| 101 | mubarra | 300000.00 | Designer  |
| 102 | leesa   | 200000.00 | Accounts  |
| 103 | Gita    | 350000.00 | Developer |
| 105 | Seeta   |  50000.00 | Developer |
| 106 | Akash   | 180000.00 | Accounts  |
| 107 | Radhika | 200000.00 | Tester    |
| 108 | Rahat   |  45000.00 | Tester    |
| 109 | Sayli   |  10000.00 | Designer  |
| 110 | Natraj  |  30000.00 | IT        |
+------+---------+-----------+-----------+
9 rows in set (0.00 sec)

mysql> drop table employee;
Query OK, 0 rows affected (0.02 sec)

mysql> desc emp_view;
ERROR 1356 (HY000): View 'v418_mubarra.emp_view' references invalid table(s) or column(s) or
function(s) or definer/invoker of view lack rights to use them
mysql> desc students;
ERROR 1146 (42S02): Table 'v418_mubarra.students' doesn't exist
mysql> create table mark(mid int, sid int,mark int);
```

```
Query OK, 0 rows affected (0.01 sec)

mysql> create table student(sid int, name varchar(20),sub varchar(20));
ERROR 1050 (42S01): Table 'student' already exists
mysql> create table student_1(sid int, name varchar(20),sub varchar(20));
Query OK, 0 rows affected (0.01 sec)

mysql> desc student_1;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| sid   | int         | YES  |     | NULL    |       |
| name  | varchar(20) | YES  |     | NULL    |       |
| sub   | varchar(20) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> desc mark;
+-------+------+------+-----+---------+-------+
| Field | Type | Null | Key | Default | Extra |
+-------+------+------+-----+---------+-------+
| mid   | int  | YES  |     | NULL    |       |
| sid   | int  | YES  |     | NULL    |       |
| mark  | int  | YES  |     | NULL    |       |
+-------+------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> insert into student_1 value(101,"x","c");
Query OK, 1 row affected (0.01 sec)

mysql> insert into student_1 value(108,"y","c++");
Query OK, 1 row affected (0.00 sec)

mysql> insert into student_1 value(106,"z","python");
Query OK, 1 row affected (0.00 sec)

mysql> select * from student_1;
+------+------+--------+
| sid  | name | sub    |
+------+------+--------+
|  101 | x    | c      |
|  108 | y    | c++    |
|  106 | z    | python |
+------+------+--------+
3 rows in set (0.00 sec)

mysql> insert into mark value(103,1,65);
Query OK, 1 row affected (0.01 sec)

mysql> insert into mark value(108,2,75);
Query OK, 1 row affected (0.00 sec)

mysql> insert into mark value(104,3,95);
```

```
Query OK, 1 row affected (0.01 sec)

mysql> insert into mark value(106,4,85);
Query OK, 1 row affected (0.01 sec)

mysql> select * from mark;
+------+------+------+
| mid  | sid  | mark |
+------+------+------+
|  103 |    1 |   65 |
|  108 |    2 |   75 |
|  104 |    3 |   95 |
|  106 |    4 |   85 |
+------+------+------+
4 rows in set (0.00 sec)

mysql> SELECT id
   -> FROM table1
   -> ;
ERROR 1146 (42S02): Table 'v418_mubarra.table1' doesn't exist
mysql> select sid from mark where sid IN (select sid from student_1);
Empty set (0.01 sec)

mysql> select sid from mark where sid IN (select sid from student_1);
Empty set (0.00 sec)

mysql> select * from marks where sid=(select * from student_1);
ERROR 1146 (42S02): Table 'v418_mubarra.marks' doesn't exist
mysql> select * from mark where sid=(select * from student_1);
ERROR 1241 (21000): Operand should contain 1 column(s)
mysql> create table copy_11(select * from mark where sid = (select * from student_1));
ERROR 1241 (21000): Operand should contain 1 column(s)
mysql> select mark.sid,student_1.sid,mark.mid,mark.mark,student_1.name,student_1.sub from
mark,student_1;
+------+------+------+------+------+--------+
| sid  | sid  | mid  | mark | name | sub    |
+------+------+------+------+------+--------+
|    1 |  106 |  103 |   65 | z    | python |
|    1 |  108 |  103 |   65 | y    | c++    |
|    1 |  101 |  103 |   65 | x    | c      |
|    2 |  106 |  108 |   75 | z    | python |
|    2 |  108 |  108 |   75 | y    | c++    |
|    2 |  101 |  108 |   75 | x    | c      |
|    3 |  106 |  104 |   95 | z    | python |
|    3 |  108 |  104 |   95 | y    | c++    |
|    3 |  101 |  104 |   95 | x    | c      |
|    4 |  106 |  106 |   85 | z    | python |
|    4 |  108 |  106 |   85 | y    | c++    |
|    4 |  101 |  106 |   85 | x    | c      |
+------+------+------+------+------+--------+
12 rows in set (0.00 sec)

mysql> select * from student,mark where student_1.sid=mark.sid;
```

```
ERROR 1054 (42S22): Unknown column 'student_1.sid' in 'where clause'
mysql> select * from student_1,mark where student_1.sid=mark.sid;
Empty set (0.00 sec)

mysql> select * from student_1,mark where student_1.sid=mark.mid;
+------+------+--------+------+------+------+
| sid  | name | sub    | mid  | sid  | mark |
+------+------+--------+------+------+------+
|  108 | y    | c++    |  108 |    2 |   75 |
|  106 | z    | python |  106 |    4 |   85 |
+------+------+--------+------+------+------+
2 rows in set (0.00 sec)

mysql> select * from student_1,mark where student_1.sid=mark.mid check(student_1.sid=108 and
mark.mid=108);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near 'check(student_1.sid=108 and mark.mid=108)' at
line 1
mysql> select * from student_1,mark where student_1.sid=108 and mark.mid=108;
+------+------+------+------+------+------+
| sid  | name | sub  | mid  | sid  | mark |
+------+------+------+------+------+------+
|  108 | y    | c++  |  108 |    2 |   75 |
+------+------+------+------+------+------+
1 row in set (0.00 sec)
```

Day 8

Wednesday, 9 October 2024 3:27 pm

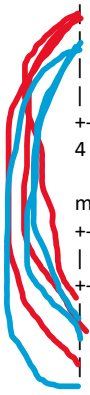
Join
Inner join
Outer join
Left join
Right join
Cross join
Self join
Natural join

Also we learnt about alias

CROSS JOIN:

- Returns the Cartesian product of both tables, combining every row from the first table with every row from the second.
- **Example:** If Students has 10 rows and Courses has 5 rows, the result of a CROSS JOIN will be 50 rows.

These are the foundational concepts that make SQL essential for relational database management.



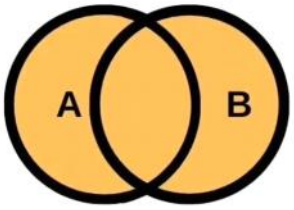
```
| sid  | mid  | marks |
+------+------+-------+
| 106 |   1 |   85 |
| 102 |   2 |   75 |
| 109 |   3 |   60 |
| 105 |   4 |   65 |
+------+------+-------+
4 rows in set (0.01 sec)

mysql> select * from student;
+------+--------+---------+
| sid  | name   | course  |
+------+--------+---------+
| 101 | Ramesh | python  |
| 106 | Geeta  | java    |
| 109 | Seeta  | Testing |
```

Outer join is not supported in mysql.

- **Full Outer Join**

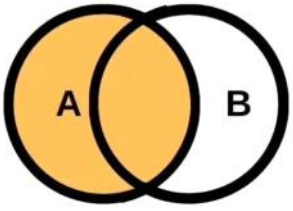
This type of join returns all rows from both tables, and any matching rows. If there is no match




```
SELECT *
FROM TableA
FULL OUTER JOIN TableB ON
TableA.ID = TableB.ID;
```

- **Left join**

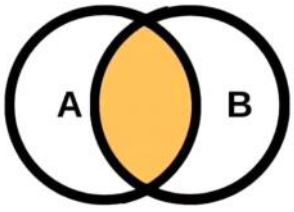
This type of join returns all rows from the left table (table A), and any matching rows from the right table (table B). If there is no match, NULL values will be returned for right table's columns.




```
SELECT *
FROM TableA
LEFT JOIN TableB ON
TableA.ID = TableB.ID;
```

- # Inner Join

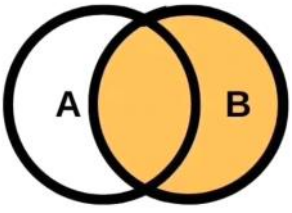
This type of join returns only the rows that have matching values in both tables.




```
SELECT *
FROM TableA
JOIN TableB ON TableA.ID =
TableB.ID;
```

- # Right Join

This type of join returns all rows from the right table (table B), and any matching rows from the left table (table A). If there is no match, NULL values will be returned for left table's columns

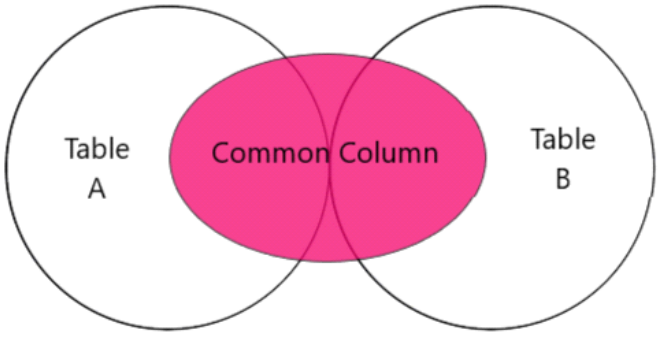



```
SELECT *
FROM TableA
RIGHT JOIN TableB ON
TableA.ID = TableB.ID;
```

NATURAL JOIN

[Natural Join in MySQL](#)
[123]

NATURAL JOIN



-

```
mysql> select * from student as s natural join Mark as m ;
+------+-------+---------+------+-------+
| sid  | name  | course  | mid  | marks |
+------+-------+---------+------+-------+
|  106 | Geeta | java    |    1 |    85 |
|  109 | Seeta | Testing |    3 |    60 |
+------+-------+---------+------+-------+
```

Natural Join in MySQL
[1][2][3]

A **natural join** in MySQL is a type of join operation that automatically combines tables based on columns with the same name and data type. It is similar to an INNER JOIN or LEFT JOIN but does not require the use of the ON or USING clause. Instead, it implicitly matches columns with the same name and data type from the tables being joined[1][2].

Syntax and Usage

The basic syntax for a natural join is as follows:

```
SELECT column_names
FROM table1
NATURAL JOIN table2;
```

In this syntax, you specify the columns to be included in the result set after the SELECT keyword. If you want to select all columns from both tables, you can use the * operator. The NATURAL JOIN clause is used between the table names to perform the join[2].

Example

Consider two tables, `table111` and `table113`, with a common column `id`. To perform a natural join on these tables, you can use the following query:

```
SELECT id, aval1, cval1
FROM table111
NATURAL JOIN table113;
```

This query will return rows where the `id` column matches in both tables, and it will include the columns `id`, `aval1`, and `cval1` in the result set[1].

From <https://www.bing.com/search?q=natural+join+in+mysql&FORM=HDRSC1>

Todays exercise

Wednesday, 9 October 2024 3:47 pm

Inner join

```
mysql> select * from mark;
+------+------+-------+
| sid  | mid  | marks |
+------+------+-------+
|  106 |    1 |    85 |
|  102 |    2 |    75 |
|  109 |    3 |    60 |
|  105 |    4 |    65 |
+------+------+-------+
4 rows in set (0.01 sec)

mysql> select * from student;
+------+--------+---------+
| sid  | name   | course  |
+------+--------+---------+
|  101 | Ramesh | python  |
|  106 | Geeta  | java    |
|  109 | Seeta  | Testing |
+------+--------+---------+
3 rows in set (0.00 sec)

mysql> select * from student as s inner join Mark as m on s.sid=m.sid;
+------+-------+---------+------+------+-------+
| sid  | name  | course  | sid  | mid  | marks |
+------+-------+---------+------+------+-------+
|  106 | Geeta | java    |  106 |    1 |    85 |
|  109 | Seeta | Testing |  109 |    3 |    60 |
+------+-------+---------+------+------+-------+
```

Day 9

Thursday, 10 October 2024 3:31 pm

Find min max and avg from same salary.

```
mysql> select desig, min(salary),max(salary),avg(salary) from employee group by desig;
+-----------+-------------+-------------+---------------+
| desig     | min(salary) | max(salary) | avg(salary)   |
+-----------+-------------+-------------+---------------+
| it        |    10000.00 |    13000.00 |  11333.333333 |
| ac        |    16000.00 |   200000.00 |  79000.000000 |
| developer |    30000.00 |   150000.00 | 110000.000000 |
+-----------+-------------+-------------+---------------+
```

Difference between truncate and delete.

1.delete can be use to delete single entry or complete table whereas truncate is used to delete complete table.

2.incase of delete - we can rollback data; if truncate is used rollback not possible.

3.

There are three types of view 1.simple view

2.complex view

```
create view std_view_1 as select * from student as s left join mark as m
on s.sid=m.id;
```

3.materialized view- home work

How to create procedure

```
delimiter #
mysql> create procedure pro()
   -> begin
   -> select st.id,st.name,st_1.id,st_1.name from st,st_1;
   -> end;
   -> #
```

Parameterized procedure.

Day 10

Friday, 11 October 2024 3:42 pm

Prce doesnot return value but function does
Function is used to apply logic

Function syntax

```
Delimiter #
Create function fun()
Returns  varchar(20) DETERMINISTIC
Begin
Declare msg varchar(20);
Set _____;
return  msg;
End;
#
```

```
mysql> create function demo() returns varchar(20) DETERMINISTIC begin declare msg
varchar(20); set msg = "hello"; return msg;end;#
Query OK, 0 rows affected (0.02 sec)

mysql> select demo()#
+--------+
| demo() |
+--------+
| hello  |
+--------+
1 row in set (0.01 sec)
```

Mysql query with if else

```
mysql> create function even(num int) returns varchar(20) DETERMINISTIC begin declare msg
varchar(20); if num%2 = 0 then set msg = "ev
en"; else set msg = "odd";end if; return msg; end;#
Query OK, 0 rows affected (0.01 sec)

mysql> select even(30)#
+----------+
| even(30) |
+----------+
| even     |
+----------+
1 row in set (0.00 sec)

mysql> select even(333)#
+-----------+
| even(333) |
+-----------+
| odd       |
+-----------+
1 row in set (0.00 sec)
```

exercise

Friday, 11 October 2024 3:44 pm

```
 create procedure pro_ins6(sid int ,sname varchar(20))
   -> begin
   -> insert into st(id,name) value(sid,sname);
   -> end;
   -> #
Query OK, 0 rows affected (0.02 sec)

mysql> call pro_ins6(7,"amaira")#
Query OK, 1 row affected (0.01 sec)
```

```
mysql> create function disc(num int) returns varchar(20) DETERMINISTIC begin declare msg
varchar(20);if num>50000 then set msg="20% discount";elseif num>25000 then set
msg="15% discount";elseif num>10000 then set msg="10% discount";elseif num>5000 then
set msg="5% d
iscount";elseif num>2000 then set msg="2% discount";else set msg="no discount";end
if;return msg;end;#
```

```
 create procedure pro_insert()
   -> begin
   -> insert into st(id,name) value(5,"anna");
   -> end;
   -> #
Query OK, 0 rows affected (0.06 sec)

mysql> call pro_insert()#
Query OK, 1 row affected (0.01 sec)

mysql> select * from st#
+------+--------+
| id   | name   |
+------+--------+
|  101 | aman   |
|  102 | sara   |
|  105 | ayeshs |
|  105 | ayeshs |
|    5 | anna   |
+------+--------+
5 rows in set (0.01 sec)

mysql> create procedure pro_ins2()
   -> begin
   -> #
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near '' at line 2
mysql> create procedure pro_ins3(sid,sname)
   -> begin
   -> insert into st(id,name) value(sid,sname);
   -> end;
   -> #
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near ',sname)
begin
insert into st(id,name) value(sid,sname);
end' at line 1
mysql> create procedure pro_ins4(sid,sname)
   -> begin
   -> insert into st(id,name) value(sid=id,sname=name);
   -> end;
   -> #
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near ',sname)
begin
insert into st(id,name) value(sid=id,sname=name);
end' at line 1
mysql> create procedure pro_ins5(sid,sname)
   -> #
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near ',sname)' at line 1
mysql> create procedure pro_ins6(sid int ,sname varchar(20))
```

```
   -> begin
   -> insert into st(id,name) value(sid,sname);
   -> end;
   -> #
Query OK, 0 rows affected (0.02 sec)

mysql> call pro_ins6(7,"amaira")#
Query OK, 1 row affected (0.01 sec)

mysql> create function demo() returns varchar(20) DETERMINISTIC begin declare msg varchar(20); set
msg = "hello"; return msg;end;#
Query OK, 0 rows affected (0.02 sec)

mysql> select demo()#
+--------+
| demo() |
+--------+
| hello  |
+--------+
1 row in set (0.01 sec)

mysql> create function even(num int) returns varchar(20) DETERMINISTIC begin declare msg
varchar(20); if num%2 = 0 then set msg = "ev
en"; else set msg = "odd";end if; return msg; end;#
Query OK, 0 rows affected (0.01 sec)

mysql> select even(30)#
+----------+
| even(30) |
+----------+
| even     |
+----------+
1 row in set (0.00 sec)

mysql> select even(333)#
+-----------+
| even(333) |
+-----------+
| odd       |
+-----------+
1 row in set (0.00 sec)

mysql> create function dby_five(num int) returns varchar(20) DETERMINISTIC begin declare msg
varchar(20); if num%5=0 then msg="yes";s
lse set msg="no"; end if; return msg; end;#
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near '="yes";slse set msg="no"; end if; return msg; end'
at line 1
mysql> create function dby_five(num int) returns varchar(20) DETERMINISTIC begin declare msg
varchar(20); if num%5=0 then msg="yes";else set msg="no"; end if; return msg; end;#
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near '="yes";else set msg="no"; end if; return msg; end'
at line 1
mysql> create function dbyfive(num int) returns varchar(20) DETERMINISTIC begin declare msg
varchar(20); if num%5=0 then set msg="yes
";else set msg="no"; end if; return msg; end;#
Query OK, 0 rows affected (0.01 sec)

mysql> select dbyfive(333)#
+--------------+
| dbyfive(333) |
+--------------+
| no           |
+--------------+
1 row in set (0.00 sec)

mysql> select dbyfive(330)#
+--------------+
| dbyfive(330) |
+--------------+
| yes          |
```

```
+--------------+
1 row in set (0.00 sec)

mysql> create function grade(num int) returns varchar(20) DETERMINISTIC begin declare msg
varchar(20); if num>90 then set msg="A Grade"; elseif num>80 then set msg = "B Grade"; elseif num>65
then set msg = "C Grade"; elseif num>45 then set msg = "D Grade"; else set
msg="Fail"; end if; return msg;end;#
Query OK, 0 rows affected (0.01 sec)

mysql> select grade(98)#
+-----------+
| grade(98) |
+-----------+
| A Grade   |
+-----------+
1 row in set (0.00 sec)

mysql> select grade(35)#
+-----------+
| grade(35) |
+-----------+
| Fail      |
+-----------+
1 row in set (0.00 sec)

mysql> select grade(47)#
+-----------+
| grade(47) |
+-----------+
| D Grade   |
+-----------+
1 row in set (0.00 sec)

mysql> create function disc(num int) returns varchar(20) DETERMINISTIC begin declare msg
varchar(20);if num>50000 then set msg="20% discount";elseif num>25000 then set msg="15%
discount";elseif num>10000 then set msg="10% discount";elseif num>5000 then set msg="5% d
iscount";elseif num>2000 then set msg="2% discount";else set msg="no discount";end if;return
msg;end;#
Query OK, 0 rows affected (0.01 sec)

mysql> select disc(12000)#
+--------------+
| disc(12000)  |
+--------------+
| 10% discount |
+--------------+
1 row in set (0.00 sec)

mysql> select disc(120000)#
+--------------+
| disc(120000) |
+--------------+
| 20% discount |
+--------------+
1 row in set (0.00 sec)

mysql> select disc(16000)#
+--------------+
| disc(16000)  |
+--------------+
| 10% discount |
+--------------+
1 row in set (0.00 sec)

mysql> select disc(25000)#
+--------------+
| disc(25000)  |
+--------------+
| 10% discount |
+--------------+
1 row in set (0.00 sec)
```

```
mysql> select disc(27000)#
+--------------+
| disc(27000)  |
+--------------+
| 15% discount |
+--------------+
1 row in set (0.00 sec)
```

Day 11

Monday, 14 October 2024 4:14 pm

While do

```
mysql> create function loop_5(value int) returns int deterministic begin declare sal int;set sal=0;WHILE
sal<=30000 DO set sal=sal+va
lue;end WHILE;return sal;end;#
Query OK, 0 rows affected (0.01 sec)

mysql> select loop_5(34500);
   -> #
+---------------+
| loop_5(34500) |
+---------------+
|         34500 |
+---------------+
1 row in set (0.01 sec)
```

Trigger - use on insert update delete
We can drop trigger
Trigger can be replaced
Show trigger shows triggers…

Create trigger tr after insert/before insert/after delete/before delete/after update/before update

```
mysql> CREATE TRIGGER calculate_total_price
   -> BEFORE INSERT ON invoice
   -> FOR EACH ROW
   -> BEGIN
   ->     -- Calculate total_price by multiplying price and qty
   ->     SET NEW.total_price = NEW.price * NEW.qty;
   -> END;#
Query OK, 0 rows affected (0.01 sec)

mysql> INSERT INTO invoice (id, name, price, qty) VALUES (101, 'Neha', 100, 2)#
Query OK, 1 row affected (0.01 sec)

mysql> select * from invoice#
+-----+------+--------+------+-------------+
| id  | name | price  | qty  | total_price |
+-----+------+--------+------+-------------+
| 101 | Neha | 100.00 |    2 |      200.00 |
+-----+------+--------+------+-------------+
1 row in set (0.00 sec)
```

exercise

Monday, 14 October 2024 4:39 pm

```
mysql> create trigger tr_2 after delete on st1 for each row begin insert into st3
values(old.id,old.name);end;#
Query OK, 0 rows affected (0.01 sec)

mysql> select * from st1;
   -> #
+------+--------+
| id   | name   |
+------+--------+
|  101 | neha   |
|  102 | sitara |
|  103 | kainat |
+------+--------+
3 rows in set (0.00 sec)

mysql> delete from st1#
Query OK, 3 rows affected (0.01 sec)

mysql> select * from st3;
   -> #
+------+--------+
| id   | name   |
+------+--------+
|  101 | neha   |
|  102 | sitara |
|  103 | kainat |
+------+--------+
3 rows in set (0.00 sec)
```

Day 12

Tuesday, 15 October 2024 4:13 pm

create trigger trmsg before update on employee for each row begin if new.salary > old.salary + 10000 then set new.msg=concat('warning: Employee ID',old.id,'salary increased by more than $10000.');end if;end;#

Trigger with function

NOW()- date function current date is fetched.

Day 13

Wednesday, 16 October 2024 3:23 pm

Loop

```
delimiter #
create procedure pro_2()
begin
declare i int default 0;
declare output varchar(255) default "";
set i=1;
set output = "";
loop_1 : loop
set i = i+1;
if i>=10 then leave loop_1;
end if;
set output=concat(output,i,",");
end loop;
select output;
end;
#
call pro_2()#
```

Day 14

Thursday, 17 October 2024 3:12 pm

- ✧ What is computer program

Components -source code()
Compiler interpreter
Types of computer prog-system software
-application software
-embedded system software
-categories of program language
Low level prg
High level prog
Special purpose language-html,sql

Sql- expressions
Timestamp,float ,decimal,bigInt,char,varchar,
Arithmatic,logical,

Constraints- default , auto increment

Alter - on existing table

Create
Modification of structure of table
Drop the table
Indexing the table(not yet done)
Er diag and normalization(not yet done)
Select
Comparison
Range operators
List operators(in ,not in)
Like
Order by , distinct, Top
Not null,is null
Insert update delete
String function math(not yet done)
Aggregate,gropu by,having
Sub query and types
Joins(self join ,equi join)

View(types)

Function and procedures

Trigger(how to get the error message on trigger)-

cursor(not yet done)

Exercise

Friday, 18 October 2024 4:26 pm

To find even numbers till a user input

```
DELIMITER $$

CREATE FUNCTION is_even(num INT)
RETURNS BOOLEAN
DETERMINISTIC
BEGIN
   -- Returns TRUE if the number is even (i.e., divisible by 2), otherwise FALSE
   RETURN MOD(num, 2) = 0;
END $$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE display_even_numbers_concat1(IN max_number INT)
BEGIN
 DECLARE i INT DEFAULT 1;
 DECLARE even_numbers TEXT DEFAULT '';  -- Variable to store the concatenated result

 -- Loop through numbers from 1 to max_number
 WHILE i <= max_number DO
  -- Call the is_even function to check if the current number is even
  IF is_even(i) THEN
    -- Concatenate the even number to the result string
    SET even_numbers = CONCAT(even_numbers, i, ', ');
  END IF;
  -- Increment the counter
  SET i = i + 1;
 END WHILE;

 -- Remove the trailing comma and space from the concatenated string
 SET even_numbers = TRIM(TRAILING ', ' FROM even_numbers);

 -- Return the concatenated result
 SELECT even_numbers AS Even_Numbers_List;

END $$

DELIMITER ;
CALL display_even_numbers_concat1(20);
```

Day 15

Friday, 18 October 2024 3:28 pm

Inbuilt functions of string
1.CONCAT - concatination of string
select concat("Good Morning ","Thanks");
2.Lower(),UPPER()
3.replace()
select replace("Good Morning","Good","Bad");
Output - Bad Morning

4. SUBSTR("",2,7)--> here index starts with 1;(start index ,and second number is how many character you want);

Math Function
1.abs()
2.select ceil(9.8); output - -> 10
3.floor(); --> o/p --> 9
4.select MOD(9,2);--> we use it in loop to find even nummber--gives remainder
5.select ROUND(-23.8);-->round off
6.select TRUNCATE(1.658,2); -->o/p 1.65
select TRUNCATE(1.658,1); -->o/p 1.6
7.select POW(2,2);
8.select sqrt(4);
9.now();-->
2024-10-18 15:52:57
10.curdate();
2024-10-18
11.sysdate(); same as now()
12.
select last_day(curdate());
select last_day('2024-05-12');-->last day of a given month

13. select date_format('2017-08-28',%y);
    select date_format('2017-08-28',"%y");
    select date_format('2017-08-28',"%d %M %y"); --
    28 August 17

19.select year(now());
20.select DATEDIFF('2007-11-30 23:59:59','2007-12-31');

Day 16

Monday, 21 October 2024 3:17 pm

Cursor: A cursor is a work area where the result of a SQL query is stored at server side.

Implicit Cursor
Explicit Cursor

```
delimiter #
create procedure pro_cur()
begin
declare i  int;
declare n varchar(20);
declare cur CURSOR for select id,s_name from student;
open cur;
fetch cur into i,n;
select i,n;
close cur;
end;
#

call pro_cur()#
```

Update using cursor

```
CREATE TABLE Employees (
   employee_id INT PRIMARY KEY,
   employee_name VARCHAR(50),
   salary DECIMAL(10, 2)
);

INSERT INTO Employees (employee_id, employee_name, salary)
VALUES (1, 'Alice', 5000.00),
     (2, 'Bob', 4500.00),
     (3, 'Charlie', 6000.00),
     (4, 'David', 3000.00);
DELIMITER $$

CREATE PROCEDURE update_salaries_with_cursor()
BEGIN
   DECLARE emp_id INT;
   DECLARE emp_salary DECIMAL(10, 2);
   DECLARE finished INT DEFAULT 0;

   -- Declare the cursor to select employee_id and salary from the Employees table
   DECLARE emp_cursor CURSOR FOR
   SELECT employee_id, salary FROM Employees;

   -- Declare a handler for when there are no more rows to fetch
   DECLARE CONTINUE HANDLER FOR NOT FOUND SET finished = 1;

   -- Open the cursor
   OPEN emp_cursor;

   -- Start the loop to fetch data row-by-row
   fetch_loop: LOOP
      -- Fetch the next row from the cursor
      FETCH emp_cursor INTO emp_id, emp_salary;

      -- Exit the loop if there are no more rows
      IF finished = 1 THEN
         LEAVE fetch_loop;
      END IF;
```

```
DELIMITER $$

CREATE PROCEDURE update_salary(new_id INT, new_fees
double(10, 2))
BEGIN

   IF EXISTS (SELECT 1 FROM student WHEREnew_id = id) THEN
      -- Update the employee's salary
      UPDATE Employees
      SET salary = new_salary
      WHERE employee_id = emp_id;

      -- Display a message indicating the update was successful
      SELECT CONCAT('Salary updated for employee ID: ',
emp_id) AS Result;
   ELSE
      -- Display a message if the employee doesn't exist
      SELECT CONCAT('Employee with ID ', emp_id, ' does not
exist.') AS Result;
   END IF;
END $$

DELIMITER ;
```

```
        -- Check if the salary is below 5000.00, then update the salary by 10%
        IF emp_salary < 5000.00 THEN
            UPDATE Employees
            SET salary = emp_salary * 1.10
            WHERE employee_id = emp_id;

            -- Optionally, show a message indicating that the salary was updated
            SELECT CONCAT('Salary updated for employee ID: ', emp_id) AS Message;
        END IF;

    END LOOP fetch_loop;

    -- Close the cursor when done
    CLOSE emp_cursor;
END $$

DELIMITER ;
CALL update_salaries_with_cursor();
SELECT * FROM Employees;
```

Day 17

Tuesday, 22 October 2024 3:13 pm

Indexing-

Create index index_name on tablename(column name);
Drop index indexname on tablename;

Types- -
1.single index - for single column
2.Composite - on multiple columns
3.Unnique index - only on primary key

ER Diagram

Table without primary key is called weak entity.
Identifying relationship - relationship between two table and one table doesn't have primary key

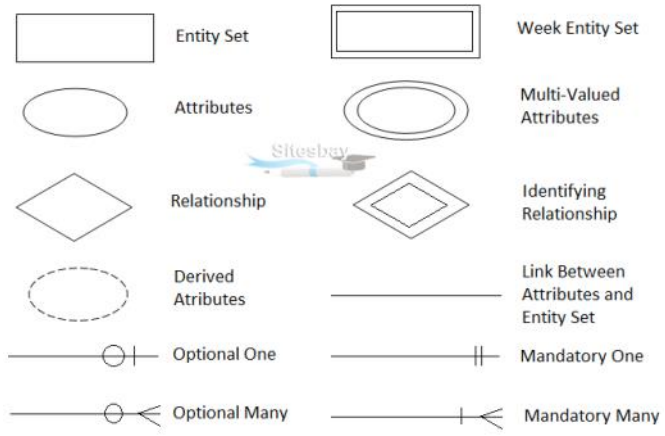
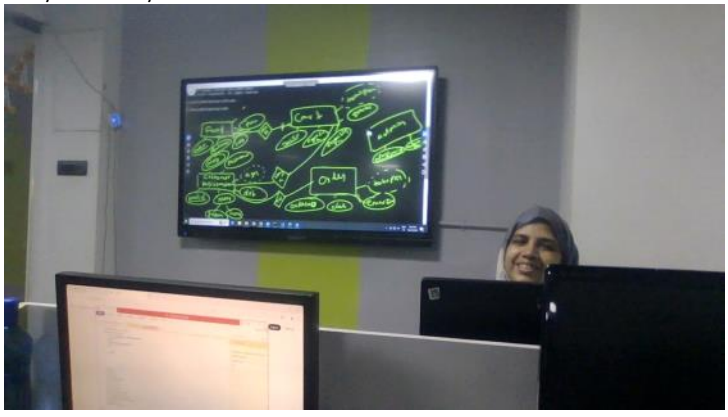


Fig: ER Diagram Symbols

Attributes:
1.Single Vs Multivalued attributes
2.simple vs Composite attributes
3.stored vs derived attributes
4.Key vs Non key attributes



Food - food id(pk),food name,type,price
Customer registration - email id(pk),name-->fname,lname,dob,age(derived)
Cart - cart id,food id(fk),email id(fk),quantity,total(derived)
Order - order id(pk),date,email id,total qty(derived)
Payment - payment id(pk), emai id(linked to customer),date,total price(linked to order)

Final complete syllabus revise

Tuesday, 22 October 2024 9:16 pm

Module 1: Introduction to Databases and SQL (Basic)

Objective: Learn what databases are, how SQL is used to interact with databases, and basic operations in MySQL.

- **1.1**: Introduction to Databases
  - What is a database?
  - Types of databases: Relational vs. Non-relational.
  - Introduction to SQL and MySQL.
- **1.2**: Setting Up MySQL Environment
  - Installing MySQL Server and MySQL Workbench.
  - Basic navigation of MySQL Workbench.
- **1.3**: Basic SQL Queries
  - **SELECT** statement.
  - **WHERE** clause for filtering data.
  - Using **DISTINCT** to avoid duplicates.
  - Sorting data with **ORDER BY**.
- **1.4**: Basic Data Types and Table Creation
  - Common data types (INT, VARCHAR, DATE, etc.).
  - Creating simple tables using **CREATE TABLE**.
- **1.5**: Inserting Data into Tables
  - **INSERT INTO** statement.
  - Inserting multiple rows.
- **1.6**: Basic Data Retrieval and Manipulation
  - Using **UPDATE** to modify data.
  - Using **DELETE** to remove data.
  - Differences between **TRUNCATE** and **DELETE**.

Day 18

Thursday, 24 October 2024 3:31 pm

Candidate key - collection of columns of unique values
Super key/Composite key - more than one column declared as primary key

Both Natural Join and Inner Join are used to combine records from two or more ta bles, but they do it differently:
Natural Join:

- Automatically joins tables based on columns with the same name and datatype.
- Doesn't require you to specify the columns to join on; it uses common columns.
- Can be convenient but also risky if tables have unrelated columns with the same n ame.

Inner Join:

- Requires you to specify the columns to join on using the ON clause.
- Provides more control by letting you define exactly how tables are related.
- Ensures that only rows with matching values in both tables are returned.

In practice, Inner Joins are more commonly used because they offer better clarity a nd control. It's like choosing between a pre-made meal and a custom order— you get more

From <https://copilot.microsoft.com/?sendquery=1&FORM=SCCODX&showconv=1>

Normalization - is used to avoid redundancy,helpful for data visualization
Row level redundancy - same rows data 2 times,using primary key on column we can avoid row level redundancy
Column level Redundancy - rows are same but in column level because of sid is primary key but columns are same..

Anomaly -

Insert anomalies-occurs when certain attributes cannot be inserted into the database w/o the presence of other attributes
Delete Anomalies - exists when certain attributes are lost because of the deletion of other attributes.
Update Anomaly: exists when one or more instances of duplicated data is updated but,but not all.

Rules of 1Nf
The official qualifications of 1NF are:
1.Each attribute name must be unique.
2.Each attribute value must be single
3.Eac row must be unique.
Additional: choose a primary key

Second Normal Form

Prime and Non-Prime attributes:

Functional Dependency: values of Y are determined by value of X.
Partial Dependency:

Partial dependency not allowed in second normal form
Example-
A table

sid	pid	sname	Pm=name

Sid	Sname	Pid

pid	pname	

Order no.	cname	caddress	Order total

Cnamefk	Order id	Total order

cname	Caddress

Third Normal Form:

Transitive dependencies
All transitive dependencies are removed in 3rd normal form

Day 19

Monday, 28 October 2024 3:15 pm

2 projects:

1.ER diagram
2.

BCNF(Boyce Code Normal Form)

Roll no	Name	Voter id	Age
1	Ravi	K0123	20
2	Varun	M034	21
3	Ravi	K786	23
4	Rahul	D236	21

Ck:{roll no,voter id}

Functional Dependency: {roll no --> name ,
Roll no --> voter id,
Voter id --> age,
Voter id --> roll no}

BCNF, or **Boyce-Codd Normal Form**, is an advanced normal form used in database normalization, which aims to reduce redundancy and dependency within a database. It is a stronger version of the third normal form (3NF). A relational table is in BCNF if it satisfies the following conditions:

1. **It is in 3NF**: This means the table must already meet all criteria for the third normal form.
2. **Every determinant is a candidate key**: In other words, for any functional dependency \(X \rightarrow Y \) (where \(X \) determines \(Y \)), \(X \) must be a superkey (a key that uniquely identifies a row).

A table is usually converted to BCNF to eliminate certain types of anomalies (insert, update, and delete) and ensure data integrity.

Example
Consider a table with the following structure:

StudentID	Course	Instructor
1	Math	Prof. A
2	Science	Prof. B
3	Math	Prof. A

In this table:
- `StudentID` and `Course` together form the primary key.
- But we have a dependency: `Course` determines `Instructor`.

Since `Course` is not a superkey, this table does not satisfy BCNF. To convert it into BCNF, we could decompose it into two tables:

1. **Student_Course** (to store which student is enrolled in which course):
   - Columns: `StudentID`, `Course`
2. **Course_Instructor** (to store the instructor for each course):
   - Columns: `Course`, `Instructor`

This decomposition removes the dependency and brings the schema into BCNF.

*******4th Normal form:**********