

MYSQL REVISION TOUR

Date: 1-04-21

***A database software basically store the related information
In the digitalize manner .***

File processing system earlier use to save the data / records.

Disadvantages:

- 1. Data redundancy (repetition)***
- 2. Data inconsistency***
- 3. Un-sharable data***
- 4. Unstandardized data***
- 5. Insecure data***
- 6. Incorrect data***

A database Management system is answer to all these problems.

- 1. Data redundancy reduced (repetition)***
- 2. Data consistent***
- 3. sharable data***
- 4. standardized data***
- 5. secure data***
- 6. integrity can be achieved***
- 7. centralized database system***

What is centralized database system?

***A sharable records in a database is called centralized database
System(We can access the data from anywhere)***

02-04-22

RELATIONAL Data Model: In relational data model data is arranged in tables where it has rows and columns..

Table represents a relation where rows represent the relationship

- **Relation:** A table storing logically related data in a cell
- **Domain :** It is a pool of values in which it can appear in a column
- **Tuple :** It is also called rows
- **Attributes :** It is also called columns
- **Degree :** it refers to the attributes /column (DC)
- **Cardinality :** It refers to the no. Of tuples.
- **View :** It is a virtual table that does not exist , which is derived from another table.
- **Key:** A value which can be identified in the table uniquely
- **Primary key :** It is the set of one or more attributes to identify the tuples in a relation.
- **Candidate key :** An attribute inside the relation which can serve as primary key are called Candidate key.
- **Alternate Key :** A candidate key that is not primary key is called Alternate key
- **Foreign key :** It is used to join two different tables

Note: if we delete some record from the “child” , it is not possible if it is linked by Foreign key.
If we delete from the base table separately we have to delete.

Referential Integrity Rule

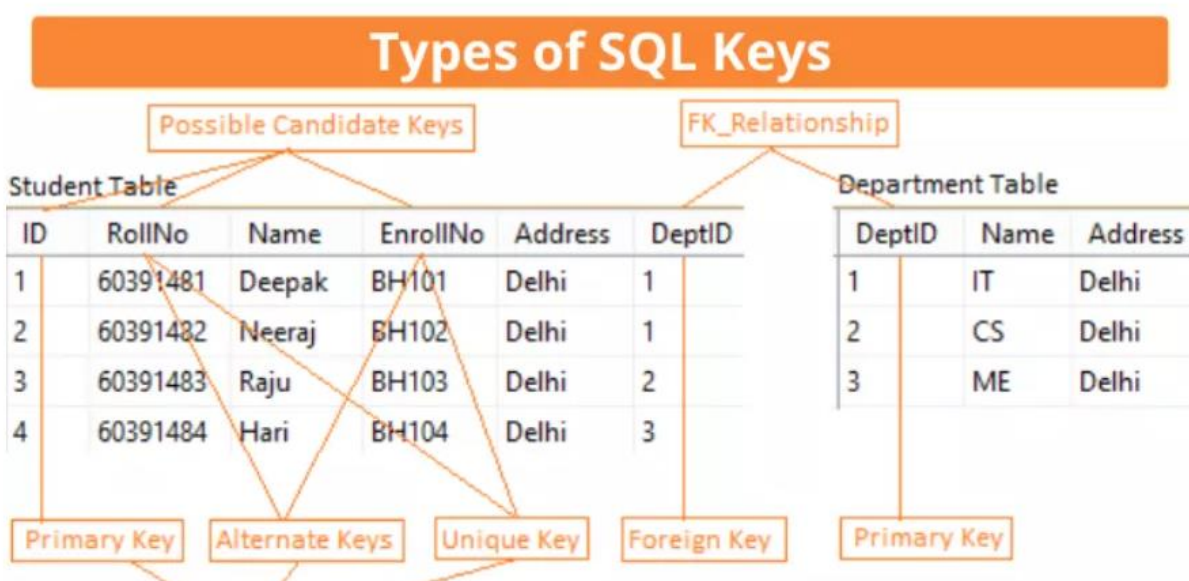
It ensures the relationships between the records in related tables..

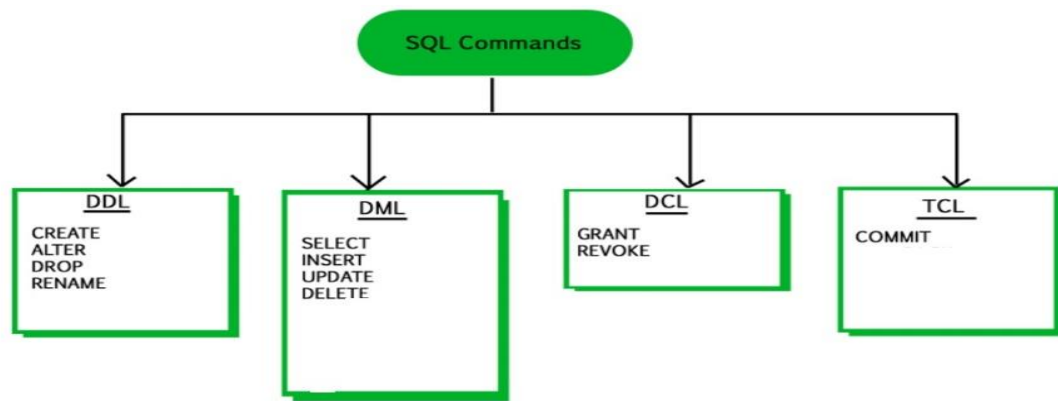
You can set referential integrity when all the following condition is met.

- 1) One of the field in the table should have unique value.*
- 2) The related field have same data type.*
- 3) Both tables belong to same database.*

Referential integrity enforced

- We can't enter a value in the foreign key of the related table does not exist in the primary key of the primary table*
- We can delete a record from the base or primary table if matching record does not exist in the secondary or child table*
- We can't change the primary key value if it is exist in the child table*





Classification of SQL statements:

- 1) DDL : Data Definition language**
- 2) DML : Data Manipulation language**
- 3) DCL : Data control language**
- 4) TCL: Transaction Control Language**

To download the WampServer

- 1) To create database**

>> create database <name-of-database>;

- 2) To get inside into the database**

>> use <name-of-database>;

- 3) To see all the database available in wampserver**

>>show databases;

- 4) To create the table**

**CREATE TABLE ACTOR
(ANO INTEGER,
ANAME CHAR(20),
AADDRESS VARCHAR(22),
ADOJ DATE,
ASALARY FLOAT);**

5) To see the structure of the table

>> desc <table-name>

6) To insert the records into the table

>>insert into <table-name> values(field1 , field2 , field3);

**Note: alphabet , alphanumeric and date values should be in
Inverted comma**

7) To see the records from the table

>> select * from <table-name>

Note: To enter any null value in the record

insert into <table-name> values(field1 , null , field3);

**# where clause: it is used to retrieve the specific
record/records from the table**

SQL QUERIES

| | |
|----------|---|
| <u>1</u> | Display the records of an employee whose job is salesman. select * from emp where job = 'salesman'; |
| <u>2</u> | Display the ename , salary and deptno of employees Whose deptno is 20. select ename , sal , deptno from emp where deptno = 20; |
| <u>3</u> | Display the records of an employee whose salary is greater and equal to 1300. select * from emp where sal >= 1300 ; |
| <u>4</u> | Display the records of an employee whose salary between 2000 to 3000. select * from emp where sal >=2000 and sal <=3000; |
| <u>5</u> | Display the records of the employee who is not getting commission select * from emp WHERE COMM IS NULL; OR WHERE COMM IS NOT NULL |
| <u>6</u> | Display the records of an employees whose job is 'salesman' or 'manager' select * from emp where job='salesman' and job='manager'; Note : we can't use "and" bcz employee not satisfy both condition of 'salesman' and 'manager' i.e we use "or" |

| | |
|-----------|---|
| <u>7</u> | <p>Display the records of an employee whose job is 'analyst' and deptno =20</p> <pre>SELECT * FROM EMP WHERE JOB='ANALYST' OR DEPTNO=20;</pre> |
| <u>8</u> | <p>Display the distinct jobs in the table emp.</p> <pre>Select distinct(job) from emp;</pre> |
| <u>9</u> | <p>Display the current date in sql</p> <pre>Select curdate() from dual</pre> |
| <u>10</u> | <p>Display the current time in sql</p> <pre>Select curtime() from dual</pre> |
| <u>11</u> | <p>Write the command to give alias (temporary) column name in the table</p> <pre>Select ename , sal as "salary" from emp</pre> |
| <u>12</u> | <p>Write a query which gives "alias" to the column sal to "salary"</p> <pre>Select ename , job , sal as "salary" from emp;</pre> |
| <u>13</u> | <p>Display the records of an employee who is getting salary between 1250 to 2850</p> <pre>Select * from emp Where sal between 1250 and 2850</pre> |
| <u>14</u> | <p>Display the records of an employee whose job is 'clerk' and 'salesman'</p> <pre>Select * from emp Where job in('clerk' , 'salesman')</pre> <p>Or</p> <pre>Where job not in('clerk' , 'salesman')</pre> |
| <u>15</u> | <p>Display the records of an employee whose name start by letter 's'</p> <pre>Select * from emp Where ename like 's%'</pre> |
| <u>16</u> | <p>Display the records of an employee whose third letter is 's'</p> <pre>Select * from emp Where ename like '__s%';</pre> |

| | |
|-----------|--|
| | <p>Alter command :</p> <p>The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.</p> <p>The ALTER TABLE statement is also used to add and drop various constraints on an existing table.</p> |
| <u>17</u> | <p>Write the command to add any column in the existing table</p> <p>Alter table <table-name> add <column-name> <data Type></p> |
| <u>18</u> | <p>Write the command to modify the column data type in the existing table</p> <p>Alter table <table-name> Modify <column-name> <update – data type> it should be similar TYPE</p> |
| <u>19</u> | <p>Write a command to drop the column from the existing table</p> <p>Alter table <table-name> Drop column <column-name></p> |
| <u>20</u> | <p>Write a command to change the column name in the existing table</p> <p>Alter table <table-name> Change column <old-column> <new column> <data type></p> |
| <u>21</u> | <p>Write a command to update the records in the table</p> <p>Update <table-name> set <column-name> = "new value" Where <condition></p> |
| <u>22</u> | <p>Write a command to delete the specific record/s from the table</p> <p>Delete from <table-name> Where <condition></p> |
| <u>23</u> | <p>View : It is a virtual table created from base table or derived from base table</p> <p>CREATE VIEW <VIEW-NAME> AS SELECT * FROM <TABLE-NAME> (SPECIFIC COLUMNS NEEDED) WHERE <CONDITION></p> |

CONSTRAINTS: It is a protocol(conditions) which we can apply on during the table creation.

Types:

1. Primary key :

In this constraint each value in the tuple must be unique
And it cannot has null value

2. Not null constraint

This constraint not allowed null values in the tuples

3. Unique

In this every tuple has unique value but we can put one or more null values

4. Default

In this constraint we assign default value in the tuples

5. Check

In this constraint we can assign the values in the tuples by checking certain conditions.

6. Foreign key constraint

It is used to join two different tables with one common column

Base table:

Create table student1

(sid int(3) primary key,
Sname varchar(20) unique,
Saddress varchar(20) not null)

Child table or linked table (by foreign key)

Create table student2

(sid int(3) ,
Sbusno int(2) not null,
foreign key(sid) references student1(sid));

| | |
|----|---|
| | 23-04-22 |
| 24 | <p>QUERIES RELATED FOR TWO TABLES</p> <p>A JOIN clause is used to combine rows from two or more tables, based on a related column between them.</p> <p>Types of joins</p> <h2>Different Types of SQL JOINS</h2> <p>The join clause is used to combine tables based on a common column and a join condition. An equi join is a type of join that combines tables based on matching values in specified columns.</p> <p>Please remember that:</p> <ul style="list-style-type: none"> • The column names do not need to be the same. • The resultant table contains repeated columns. • It is possible to perform an equi join on more than two tables. <p>DISPLAY THE ENAME , JOB , DEPTNO WHOES DNAME IS 'ACCOUNTING'</p> <pre> SELECT ENAME, JOB , DNAME FROM EMP E , DEPT D WHERE E.DEPTNO = D.DEPTNO AND DNAME = 'ACCOUNTING'; </pre> |
| | 25-04-22 |
| | <h3>Natural Join</h3> <p>A NATURAL JOIN is a JOIN operation that creates an implicit join clause for you based on the common columns in the two tables being joined. Common columns are columns that have the same name in both tables.</p> <p>A NATURAL JOIN can be an INNER join, a LEFT OUTER join, or a RIGHT OUTER join. The default is INNER join.</p> |

| | |
|------------------|--|
| <p><u>25</u></p> | <p>select * from table1 natural join table2;</p> <p>MAKING THE DUPLICATE COPY OF THE TABLE (SPECIFIC COLUMNS) FROM THE EXISTING TABLE</p> <p>INSERT INTO USA(EMPNO, JOB , SAL , COMM , DEPTNO) SELECT EMPNO , JOB , SAL , COMM , DEPTNO FROM EMP WHERE SAL BETWEEN 800 AND 1500;</p> |
| <p><u>26</u></p> | <p>26-04-22 (Chapter 13 “More On SQL”)</p> <p>Order by clause (Arranging the records in ascending or descending)</p> <p>Display the records of an employee by ascending and descending order by ename</p> <p>Select * from emp Order by ename ASC ;</p> <p>Select * from emp Order by ename DESC;</p> |

| | |
|-----------|--|
| | <p>Aggregate/ group functions</p> <p>It will apply on group of column values</p> |
| <u>27</u> | <p>Write a command to find the total salary giving to an employee</p> <p>Select sum(sal) from emp;</p> |
| <u>28</u> | <p>Write a command to find the average salary to an employs</p> <p>Select avg(sal) from emp;</p> |
| <u>29</u> | <p>Write a command to count the no of salesman available in the emp table.</p> <p>Select job , count(job) from emp</p> <p>Where job = 'salesman';</p> |
| <u>30</u> | <p>Write a command to find out the total salary giving to 'salesman' in employee table.</p> <p>Select job , sum(sal) from emp</p> <p>Where job = 'salesman'</p> |
| <u>31</u> | <p>Write a query to find the maximum salary giving to an employee in emp table</p> <p>Select ename , sal from emp</p> <p>Where sal = (select max(sal) from emp)</p> |
| <u>32</u> | <p>Write a query to find the minimum salary giving to an employee in emp table</p> <p>Select ename , sal from emp</p> <p>Where sal = (select min(sal) from emp)</p> |
| <u>33</u> | <p>Group by clause : it is applicable for multiples rows but in specific columns</p> <p>Display the number of SALESMAN jobs available in the emp table.</p> <p>SELECT JOB , COUBT(JOB) FROM EMP</p> <p>WHERE JOB = 'SALESMAN';</p> |

| | |
|-----------|---|
| | |
| | |
| <u>34</u> | Display the number of jobs individually available in emp TABLE SELECT JOB , COUNT(JOB) FROM EMP GROUP BY JOB |
| <u>35</u> | Display the no of deptno available in emp table Select deptno , count(deptno) from emp Group by deptno |
| <u>36</u> | Display the no of employee doing job in different jobs profile SAME AS 34 |
| <u>37</u> | Display the total salary giving to individual job profile. Select job , count(job) , sum(sal) From emp Group by job |
| <u>38</u> | Display the total salary giving to salesman in the emp table Select job , count(job) , sum(sal) From emp Group by job Having job = 'salesman'; |
| <u>39</u> | Write a query to count the no of jobs count more than 2 select job , count(job) from emp group by job having count(job) > 2; |
| <u>40</u> | Display the sum of salary giving to 'salesman; and 'manager' From emp table select job, count(job) , sum(sal) from emp group by job having job in('salesman' , 'manager'); |

