Chapter: MySQL ReVision Tour

Introduction (Advantage of Database)

Database:-

(i) It is basically a computer based record keeping system

(ii) The collection of data, usually referred to as the database, contains information about one particular enterprise.

Various advantages of database systems

(i) It reduce data redundancy(Data duplication) to a large extent.

(ii) It control data inconsistency to a large extent .

(iii) Database facilitate sharing of data.

(iv) Databases enforce standards.

(v) Centralized databases can ensure data security.

Integrity can be maintained through databases

Relational Data Model

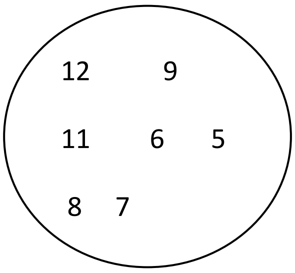
- In relational data model, the data is organized into tables(i.e , rows and columns).
- Tables are called relations.
- A row in a table represents a relationship among a set of values.
- Note:--- In table rows represents a relationship among a set of values therefore it is generally referred to
- using the mathematical term relation , from which the relational data model derives its name.

Terms used In SQL

- Relation :- A table storing logically related data , data must be atomic in cell, all rows of the table should be distinct, ordering of rows and columns is immaterial.
- Domain:- This is a pool of values which the actual values appearing in a given column are drawn

Table 1

Sl	Class	name



1	11	Raman
2	12	Tilak
3	6	Ashok
4	8	Mohan
5	9	Ram

class

Table 3

Table 2

Class	Add	House
6	Guwahati	LO
7	Barpata	LA
8	Goalpara	LO
9	Patna	EKL
10	Hajipur	LO

- Tuple :- A row of a table(relation) is generally referred to as tuple.
- Attribute :- A column of table(relation) is generally referred to as an attribute.

- Degree :-This refers to the number of attributes in a table(relation)
- Cardinality :-This refers to the number of tuples in a table(relation.)
- View :- It is a virtual table that does not really exist in its own right but is instead derived from one or more underlying base tables.
- Primary Key :--This refers to a set of one or more attributes that can uniquely identify tuples with the relation
- Candidate Key:- All attribute combinations inside a table(relation) that can serve as primary key are candidates keys as these are candidates for primary key position.
- Alternate Key :-A candidate key that is not primary key, is called an alternate key.

- Foreign Key:- A non primary key attribute, whose values are derived from the primary key of some other table, is known as foreign key in its

Table:2

Primary Key

TABLE:1

Primary Key

s	Add	House
6	Guwahati	LO
7	Barpata	LA
8	Goalpara	LO
9	Patna	EKL
10	Hajipur	LO

Sl	Class	name
1	6	Raman
2	7	Tilak
3	8	Ashok
4	9	Mohan
5	10	Ram

In Table 2 :

Primary Key---Sl no

In Table 1 :

Primary Key---Class

Non Primary Key—Add House

- Non primary key Class of table 2 is Foreign key because its values derived from the primary key of a table1(another table) therefore we can say that Class is a foreign key.

Brief History of MySQL.

- MySQL was created and is supported by MySQL AB, a company based in Sweden(www.mysql.com).
- This company is now a subsidiary of Sun Microsystems, which holds the copyright to most of the codebase.
- The chief inventor of MySQL was Michael Widenius(a.k.a. Monty)

MySQL Database System

- The Key role of a database management system is information management.
- A database server is the key to solving the problems of information management.
- MySQL operates using **client /server** architecture in which the server runs on the machine containing the databases and clients connect to the server over a network.
- **The server**(MySQL server) listens for client requests coming in over the network and accesses database contents according to those requests and provides that to the clients.
- **Clients** are programs that connect to the database server and issue queries in a pre-specified format.
- **Key features are** speed, Ease of use, Cost, Query Language Support, MySQL, Portability, Data Types, Security, Scalability and Limits, Connectivity, Localization, Clients and Tools.

MySQL and SQL

- In order to access data within the MySQL database, all programs and users must use, SQL.
- SQL is the set of commands that is recognised by nearly all RDBMSs.
- Usage of SQL has become a standard for most of RDBMSs.
- There are numerous version of SQL. The original version was developed at IBM's San Jose Research Laboratory(now the Almandan Research Centre).
- This language, originally called Sequel.

Processing Capabilities of SQL

- Data Definition Language(DDL)
- Interactive Data Manipulation Language(DML)

- Embedded Data Manipulation Language
- View Definition
- Authorization
- Integrity
- Transaction Control

Classification of SQL Statements

- SQL, technically speaking, is a data sublanguage. That is, it is a language used to interact with databse.
- In other words, all SQL statements are instructions to the database only.

SQL commands can be divided into following categories

- Data Definition Language(DDL) Commands
- Data Manipulation Language(DML) Commands
- Transaction Control Language (TCL) Commands eg **COMMIT, ROLLBACK, SAVEPOINT, SET TRANSACTION**
- Session Control Commands
- System Control Commands

SOME MySQL SQL ELEMENT

- Literals – It generally, refer to a fixed data value which may be character type or numeric literal.
- Datatypes – Are means to identify the type of data and associated operations for handling it.
- Nulls – If a column in a row has no value, then columnis said to be null, or to contain a null.
  Any arithmetic expression containing a null, always evaluates to null.
- Comments- A comment is a text that is not executed; it is only for documentation purpose.

SQL Command Syntax

- The SQL provides a predefined set of commands that help us work on relational database.
- In commands different terminology used are **keywords, commands, or statements.**
- **Clauses –** Commands consist of one or more logically distinct parts called **clauses.**
  “FROM sales” AND “WHERE value=1500.00” here from and where are clauses.
- Commands in SQL are not case –sensitive.

Making Simple Queries

- To fully use the power of an RDBMS, we need to communicate with it.A powerful way of communicating with it is making queries.
- We can see desired data in desired format with help of **SELECT** command.
- **Other commands are :-**
  - ✓ Distinct command

- ✓ All keyword
- ✓ Show tables- Viewing Structure of Table
- ✓ Simple calculations(eg select 1+6)
- ✓ Scalar Expressions with Selected Fields
- ✓ Using Column Aliases
- ✓ Handling nulls
- ✓ Putting Text in the Query Output
- ✓ Selecting Specific Rows-**Where** clause
- ✓ Relational Operators
- ✓ Logical Operators
- ✓ Condition based on a Range (eg BETWEEN)
- ✓ Condition Based on a List
- ✓ Condition Based on Pattern Matches (%, _, LIKE, NOT LIKE)
- ✓ Searching for NULL
- ✓ Operator Precedence
- ✓ Sorting Results—ORDER BY clause
- ✓ Sorting by Column Alias

Accessing Database

- **1st we need to open the database for use.**
- **Command--- use <databasename>**
  Eg;- USE MENAGERIE

Creating Tables in MySQL

- Before issuing a CREATE TABLE COMMAND, MAKE SURE THAT ITS PARENT DATABASE HAS BEEN OPENED USING **USE<DATABASE> COMMAND**.
  - Tables are defined with the CREATE TABLE command.

  - When tables is created , its columns are named, data types and sizes are supplied for eachcolumn.

  - Each table must have at least one column.

➢ To create an employee table whose schema is as follows:

employee (ecode, ename, sex, grade, gross)

SQL command will be

```
CREATE TABLE employee
 ( ecode   integer,
ename   char(20),
sex     char(1),
grade   char(2),
gross   decimal);
```

Inserting Daata

- The rows (tuples) are added to relations using INSERT command of SQL.
- In it simplest form, Insert takes the following syntax:-

```
INSERT INTO <TABLENAME> [<COLUMN LIST>]
VALUES(<VALUE>,<VALUE>.....);
```

EXAMPLE

```
INSERT INTO employee
VALUES(1001, ‘Ravi’,’M’,’E4’,4670.00);

            OR
INSERT INTO employee( ecode, ename, sex, grade , gross)
VALUES(1001,’Ravi’,’M’,’E4’,4670.00);
```

- The INSERT statement adds a new row to employee giving a value for every column in therow.
- The data values are in the same order as the column names in the table.
- Data can be added only to some columns in a row by specifying the columns and their data.

  Eg

  ```
  INSERT INTO employee(ecode, ename, sex)
  VALUES(2014, ‘Manju’,’F’);
  ```

- The columns that are not listed in the INSERT command will have their default value, if it is defined for them, otherwise, MULLvalue .
- If any other column (that does not have a default value and is defined NOT NULL)is skipped or omitted, an error message is generated and the row is not added.

Inserting NULL values

- To inset value NULL in a specific column. We can type NULL without quotes and NULL will be inserted in that column.
- Eg

  ```
  INSERT INTO EMPL(Empno, Ename, Job, Mgr, Hiredate, Sal, Comm, Deptno)
  VALUES(8100, ‘YASH’, ‘ANALYST’, NULL, ’10-MAY-03’, 6000, NULL, 20);
  ```

Inserting Dates

- Dates are by default entered in 'YYYY-MM-DD' format i.e 1st four digits depicting year, followed by a hyphen, followed by 2 digits of month followed by a hypen and a two digit day.

- All this is enclosed in single quotes.

Inserting Daata

- The rows (tuples) are added to relations using INSERT command of SQL.
- In it simplest form, Insert takes the following syntax:-

  ```
  INSERT INTO <TABLENAME> [<COLUMN LIST>]
  VALUES(<VALUE>,<VALUE>.....);
  ```

  EXAMPLE

  ```
  INSERT INTO employee
  VALUES(1001, 'Ravi','M','E4',4670.00);

            OR
  INSERT INTO employee( ecode, ename, sex, grade , gross)
  VALUES(1001,'Ravi','M','E4',4670.00);
  ```

- The INSERT statement adds a new row to employee giving a value for every column in therow.
- The data values are in the same order as the column names in the table.
- Data can be added only to some columns in a row by specifying the columns and their data.

  Eg INSERT INTO employee(ecode, ename, sex)
  VALUES(2014, 'Manju','F');
- The columns that are not listed in the INSERT command will have their default value, if it is defined for them, otherwise, MULLvalue .
- If any other column (that does not have a default value and is defined NOT NULL)is skipped or omitted, an error message is generated and the row is not added.

Inserting NULL values

- To inset value NULL in a specific column. We can type NULL without quotes and NULL will be inserted in that column.
- Eg

  INSERT INTO EMPL(Empno, Ename, Job, Mgr, Hiredate, Sal, Comm, Deptno)

VALUES(8100, ‘YASH’, ‘ANALYST’, NULL, ’10-MAY-03’, 6000, NULL, 20);

Inserting Dates

- Dates are by default entered in ‘YYYY-MM-DD’ format i.e 1st four digits depicting year, followed by a hyphen, followed by 2 digits of month followed by a hypen and a two digit day.
- All this is enclosed in single quotes.
- Making Simple Queries Through Select Command
- General form
- Test

Sl	Class	name	Date_ of _admission
1	11	Raman	11/07/05
2	12	Tilak	13/08/02
3	6	Ashok	24/09/01
4	8	Mohan	17/05/99
5	9	Ram	19/09/00

- SELECT what_to_select
- FROM which_table
- WHERE conditions_to_satisfy;
- Selecting particular Rows
- Select sl, class
- From Test
- Where name=Ashok
- Select *
- From test
- Where name= Ashok

Sl	Class
3	6

Sl	Class	name	Date_ of _admission
3	6	Ashok	24/09/01

Select *

From Test

Where Date_of_admission>=’13/08/02’;

Sl	Class	name	Date_ of _admission
1	11	Raman	11/07/05
2	12	Tilak	13/08/02

```
Select  name, date_of_admission

From test

Where Date_of_admission>='13/08/02';
```

name	Date_ of _admission
Raman	11/07/05
Tilak	13/08/02

Example of some select statement

```
(I)      Select  *
         from  pet
         where species='dog' & sex='f';
(II)     Select *
         From pet
         Where species ='snake' or species='bird';
(iii)    Select *
         from pet
         where(species='cat' and sex='m');
```

SEECTING Particular Columns

Ex_ select class , name

From test

Eliminating Redundant Data (with keyword Distinct)

- We always use "DISTINCT" keyword to remove redundant(duplicate) data.
- The DISTICT KEYWORD eliminates duplicate rows from the results of a select statement

  Example:

  ```
  select  distinct  name
  from test
  ```

name
Raman
Tilak
Mohan

Selecting From All the Rows ---(**All** keyword)

In place of keyword Distinct , we can give keyword ALL then the result retains the duplicate

name
Raman
Tilak
Mohan
Mohan
Mohan

(i) Viewing Structure of a table

DESCRIBE/DESC<table name;

(ii) Performing Simple Calculations

To perform simple calculation on a table we have to write expression/formula to be calculated next to keyword SELECT, e.g.,

```
Eg   select  4*6*6
     Select curdate()
```

(iii) Using Column Aliases

The columns that we select in a query can be given a different name i.e column alias name for output purpose. As per following syntax:

(primary key)

Sl	class	name	Date_ of _admission
1	11	Raman	11/07/05
2	12	Tilak	13/08/02
3	6	Mohan	24/09/01
4	8	Mohan	17/05/99
5	9	Mohan	19/09/00

ssgclass
11
12
6
8
9

(iv) Condition Based on a Range

- The BETWEEN operator defines a range of values that the column values must fall in to make the condition true.
- The range includes both lower value and the upper value.
- Eg Select sl, name, date_of_admission

  From test

  Where class between 6 and 9

Sl	name	Date_ of _admission
3	Mohan	24/09/01
4	Mohan	17/05/99
5	Mohan	19/09/00

(v) Condition Based on a list

- To specify a list of values , IN operator is used.
- The IN operator selects values that match any value in a given list of value .
- Select *

  From test

  Where class in(11,12);

Sl	class	name	Date_ of _admission
1	11	Raman	11/07/05
2	12	Tilak	13/08/02

(vi) Condition Based on Pattern Matches

- SQL includes a string –matching operator , LIKE for comparisons on character strings using patterns .
- Two special wildcard characters

  - Percent (%) . The character matches any substrings

    Eg

    ```
    select firstname , lastname, city
    from members
    where pin like’13%’ ;
    ```

Pin
130006
130007
240000
139676
119870

    ```
    Select name
    From  emp
    Where name like  ’%y’
    ```

  - Underscore(_).The_ character matches any character.

(vii) Searching for Null

- The NULL value in a column can be searched in a table using IS NULL IN THE WHERE CLAUSE

EG

```
SELECT EMPNO, ENAME, JOB
FROM EMP
WHERE DEPTNO IS NULL
```

(viii) Creating Tables with SQL Constraints

- To apply conditions on columns , SQL constraints are used
- Common types of constraints are
- NOT NULL (It ensures that a column cannot have NULL value)
- DEFAULT(Provides a default value for a column when none is specified.
- UNIQUE(Ensures that all values in a column are different)
- CHECK(Makes sure that all values in a column satisfy certain criteria)
- PRIMARY KEY(used to uniquely identify a row in the table.)
- FOREIGN KEY(used to ensure referential integrity of the data.

Student

```
Eg.  Create  table student

   ( Student_ID integer  check (Student_ID>0), Unique

     Student_Rollno  integer   NOT NULL, Primary Key
```

```
Last_Name  varchar(30),

Score  Default  80);
```

Student_Rollno	Student_ID	Last_Name	Score
1003	2	Das	67
1002	2	Deka	80
1003	3	Singh	40
1005	23	Hajarika	87

(ix) Inserting Data Into another Table

Eg

```
 INSERT INTO EMPLOYEE
 VALUES(1001,'Rahman','M','E4',1400.00 );
Or
INSERT INTO STUDENT
VALUES(1005,23, HAJARIKA,87)
```

Eg

```
Insert into branch1
Select * from branch2
Where gross>7000.00
```

(x) Modifying Data in Tables

- We can modify data in tables using UPDATE COMMAND OF SQL
- The UPDATE command specifies the rows to be changed using WHERE clause
- The new data using the SET keyword.

Eg

```
UPDATE  STUDENTS
 SET Student_Rollno=1009
  Where (Score=40)

UPDATE ITEMS
SET ROL=400,QOh=700
Where  icode<'I040';
```

(xi) Deleting Data from Tables

- To delete some data from tables , we can use SQL delete commands.
- Delete fro emp
- Where gross<2200

(xii) Altering Tables

To change the existing column namely First_Name of table Student, to FirstName

```
Alter TABLE customers
Change First_Name    FirstName  VARCHAR(20)
```

(xiii) Dropping Table

The DROPTABLE USE FOR DROPING A TABLE FROM THE DATABASE.

Syntax:---   Name of table.

Eg DROP TABLE items