MySQL

Mukesh A Pund

Principal Scientist

NISCAIR, New Delhi

Database Management System (DBMS)

Database: Collection of interrelated data

Set of programs to access the data

DBMS contains information about a particular enterprise

DBMS provides an environment that is both *convenient* and *efficient* to use.

Database Applications:

Banking: all transactions

Library: Acquisition, Circulation, OPAC, Stock Verification, Serial control, Digital Library, Bibliographic databases, Union Catalogue etc.

Airlines: reservations, schedules

Universities: registration, grades

Sales: customers, products, purchases

Manufacturing: production, inventory, orders, supply chain

Human resources: employee records, salaries, tax deductions

Databases touch all aspects of our lives

Examples of RDBMS

- Proprietary RDBMS
  - ★ Informix
  - ★ Sybase
  - ★ MS SQL Server
  - ★ Visual FoxPro
  - ★ MS Access
  - ★ Oracle
  - ★ DB2
- Open Source RDBMS
  - ★ **MySQL**
  - ★ Postgresql
  - ★ HSQLDB (**H**yper **SQL D**ata**b**ase)
  - ★ SQLite

Data Definition Language (DDL)

- Specification notation for defining the database schema

  E.g.

  - **Create database** *library*
  - **create table** book (
    accno bigint(10),
    title char(50)
    author char(35))
  - **Drop table** *book*
- DDL compiler generates a set of tables stored in a *data dictionary*
- Data dictionary contains metadata (i.e., data about data)
  - database schema
  - Data *storage and definition* language
    - ✔ language in which the storage structure and access methods used by the database system are specified
    - ✔ Usually an extension of the data definition language

Data Manipulation Language (DML)

- Language for accessing and manipulating the data organized by the appropriate data model
  - ★ DML also known as query language
- Two classes of languages
  - ★ Procedural – user specifies what data is required and how to get those data
  - ★ Nonprocedural – user specifies what data is required without specifying how to get those data
- SQL is the most widely used query language

SQL

- This is a language used only for retrieving information only
- The syntax (grammar) of SQL is mostly common in different vendors of SQL like SQL server, PL SQL
- Easy to understand, least experienced users can issue commands to retrieve information
- However lacks commands for developing software
- SQL: widely used non-procedural language
  - ★ E.g. find the titles of the book written by Henry Korth

    select *book.title*
    from *book*
    where book.*author* = ‘Henry Korth’

- Application programs generally access databases through one of
  - ★ Language extensions to allow embedded SQL
  - ★ Application program interface (e.g. ODBC/JDBC) which allow SQL queries to be sent to a database

Database Users

- Users are differentiated by the way they expect to interact with the system
- Application programmers – interact with system through DML calls
- Sophisticated users – form requests in a database query language
- Specialized users – write specialized database applications that do not fit into the traditional data processing framework
- Naïve users – invoke one of the permanent application programs that have been written previously
  - ★ E.g. people accessing database over the web, bank tellers, clerical staff

What to learn in coming sessions?

Mainly Structured Query Language (SQL)

- DDL
  - ★ Data types
  - ★ Creation of databases, tables, views etc
  - ★ Modification of table structures
  - ★ Deletion of databases, tables, views etc
- DML
  - ★ Insertion of data into tables
  - ★ Applying constraints
  - ★ Retrieval of information from single and multiple tables (Joins)
  - ★ Modification of data
  - ★ Ordering of data
- Security of database

Structured Query Language (SQL) (in MySQL)

Mukesh A Pund

Principal Scientist

NISCAIR, New Delhi

SQL is a Standard - BUT....

- SQL is an ANSI (American National Standards Institute) standard computer language for accessing and manipulating database systems. SQL statements are used to retrieve and update data in a database. SQL works with database programs like MS Access, DB2, Informix, MS SQL Server, Oracle, Sybase, MySQL etc.
- Unfortunately, there are many different versions of the SQL language, but to be in compliance with the ANSI standard, they must support the same major keywords in a similar manner (such as SELECT, UPDATE, DELETE, INSERT, WHERE, and others).
- **Note:** Most of the SQL database programs also have their own proprietary extensions in addition to the SQL standard!

Log-in into MySQL from Linux Terminal

First Check MySQL is running?

- $mysqladmin -u root -p status

If this is showing error then Log-in root terminal and start it

- #service mysqld start OR
- #/etc/init.d/mysqld start

Log-in into MySQL

$mysql -u root –p //syntax: mysql -u user_name -p password

Enter the user password

Now MySQL terminal start in Linux Terminal

- mysql>

Create/Drop user and database

Log-in into MySQL terminal:

$ mysql -u root -p mysql

$Enter Password:

Create Database: mysql> CREATE DATABASE newdb;

Create User and Grant Permission:

mysql> create user 'new_user'@'localhost' IDENTIFIED BY 'password';

mysql>GRANT ALL ON *.* TO 'new_user'@'localhost';

Delete database: mysql> DROP DATABASE newdb;

Delete user: mysql> DROP USER new_user@localhost;

Alt. command:

mysql> DELETE FROM mysql.user WHERE user='new_user' and host='localhost';

mysql> FLUSH PRIVILEGES;

Backup and Restore

Backup of single database

- $mysqldump -u root -p database_name > /home/root/Desktop/db_backup.sql

Restore of single database

- $mysql -u root -p new_database_name < /home/root/Desktop/db_backup.sql

Backup of All databases

```
$mysqldump -u root -p  --add-drop-database --all-databases > full.dump
```

Restore of All databases

- $ mysql -u root -p < full.dump

Creation of database

CREATE DATABASE database_name

Eg. CREATE DATABASE library;

Example of tables

- Below is an example of “Book” table

Book

Accno	Title	Author	Price	isbn
101	Social Processes	S. Sharma	300.00	81-266-2078-1
102	Fundamentals of IT	P. Dubey	600.00	71-265-2077-2
103	Management Information System	Lauden & Lauden	700.00	72-265-2076-3

Example of tables

- Below is an example of “Members” table

Members

Memno	Name	Maxbooks	Maxdays
M01	Gian Singh	4	15
M02	A S Rao	4	15

Example of tables

- Below is an example of “book_hold” table

Book_hold

Book_accno	Members_memo
101	M01
104	M02

Various data types in MySQL

Data type	Description	Example
bigint(size) int(size)	Hold integers only. The maximum number of digits are specified in parenthesis.	accno
decimal(size,d) numeric(size,d)	Hold numbers with fractions. The maximum number of digits are specified in "size". The maximum number of digits to the right of the decimal is specified in "d".	Price of book
char(size)	Holds a fixed length string (can contain letters, numbers, and special characters). The fixed size is specified in parenthesis. Can store up to 255 characters	ISBN Number
varchar(size)	Holds a variable length string (can contain letters, numbers, and special characters). The maximum size is specified in parenthesis. Can store up to 255 characters. Note: If you put a greater value than 255 it will be converted to a TEXT type	Title, author name etc
Date (yyyy-mm-dd) Datetime (yyyy-mm-dd hh:mm:ss)	Holds a date	Date of joinig, issue date etc

Data Integrity with constraints

- Data Integrity
  - ★ Refers to the consistency and accuracy of the data that is stored in a database
- Constraints
  - ★ Constraints offers a way to have SQL server enforces the integrity of the database automatically
  - ★ Constraints defines rules regarding the values allowed in columns and are the standard mechanism for enforcing integrity

Classes of constraints

- **Not Null** – specifies that the column does not accept null values
- **Check** – enforces to accept only the specified values / values in range in a set of column
- **Default** – if no value is supplied to a column, then the column gets the value set as DEFAULT
- **Primary Key** – Identify a column or a set of column whose values uniquely identify a row in table
- **Foreign Key** – Identify the relationship between the tables
- **Unique Key** – The UNIQUE constraint in Mysql does not allow to insert a duplicate value in a column.

Note: Major difference between Primary Key and Unique Key is that Unique key has a Null value and more than one column but Primary Key is not.

Creation of tables

To create a table in a database (Without Constraints):

CREATE TABLE table_name(column_name1 data_type, column_name2 data_type,.....)

Example:

CREATE TABLE book *(accno bigint(5), title varchar(50),author varchar(30),publisher varchar(50)*)

To create a table in a database (With Constraints):

CREATE TABLE table_name(column_name1 data_type ([Size]) *[Constraint]*, column_name2 data_type([Size]) *[Constraint]*,.....)

Example:

CREATE TABLE book *(accno numeric(5) primary key, title varchar(50),* ***authid int (4), FOREIGN KEY (authid) REFERENCES author (auth_id),*** *publisher varchar(50) not null*)

Inserting data into tables

- INSERT INTO table_name VALUES (value1, value2,....)

  OR

- INSERT INTO table_name

  (column_name1, column_name2,...)
  VALUES (value1, value2,....)

- Example1:

  INSERT INTO book values (101,”Social Processes”,”S. Sharma”,300,”81-266-2078-1”)

- Example2:

  Insert into members values (“M01”,”Gian Singh”,4,15)

- Example3:

  Insert into book_hold values (101,”M01”,’2007/04/05’,’2007/04/20’)

- Example4: (For values in specified column)

  INSERT INTO members

  (memo, name) VALUES (“m06”,”Mukesh Pund”)

Modifying structure of tables

- Adding column
- ALTER TABLE table_name
  ADD column_name datatype
- Ex. 1
- Alter table book add publisher varchar(50);

Modifying structure of tables

- Deleting column
- ALTER TABLE table_name
  drop column_name
- Ex. 1
- Alter table book drop publisher;

Specifying conditions with Where Clause

- WHERE clause is used to specify conditions in ALTER, UPDATE, SELECT etc.

Example:

```
UPDATE book
    SET author='P Sharma'
    WHERE accno=101;
```

- Here value “S Sharma” is being stored in author attribute in record with accno 101 and update with “P Sharma”

Comparison Operators

Following operators are used while specifying conditions

Operator	Operator
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
=	Equal to
<>	Not equal to
BETWEEN	Between an inclusive range
IN	If you know the exact value you want to return for at least one of the columns
LIKE	Search for a pattern

Modifying Data in tables

- UPDATE clause is used to modify and delete data in the tables
- UPDATE table_name
  SET column_name=new_value
  [, column_name=new_value]
  WHERE column_name=some_value
- Example:
- UPDATE book
  SET author='S Sharma'
  WHERE accno=101 ;

Modifying Data in tables

- Deleting data from table:
- DELETE FROM table_name
  (**Note:** Deletes the entire table!!)

OR

- DELETE FROM table_name
  WHERE condition

Example1:

- delete from book WHERE accno=101;

Retrieval of data

- **The SQL SELECT Statement**
- The SELECT statement is used to select data from a table. The tabular result is stored in a result table (called the result-set).

SELECT column_name(s) FROM table_name

SQL SELECT Example

- To select the content of columns named “Title" and “Author", from the database table called “Book", use a SELECT statement like this

```
SELECT author, title FROM book;
```

- To select all columns from the "Persons" table, use a * symbol instead of column names, like this:

```
SELECT * FROM book;
```

The SELECT DISTINCT Statement

- The DISTINCT keyword is used to return only distinct (different) values.
- With SQL, all we need to do is to add a DISTINCT keyword to the SELECT statement:
- SELECT DISTINCT column_name(s) FROM table_name
- Ex: display all unique titles which are available in library

  SELECT DISTINCT title FROM book;

SELECT statement with WHERE clause

- To conditionally select data from a table, a WHERE clause can be added to the SELECT statement.

- SELECT column FROM table WHERE column operator value

SELECT statement with WHERE clause

The operators that can be used with WHERE clause:

Operator	Operator
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
=	Equal to
<>	Not equal to
BETWEEN	Between an inclusive range
IN	If you know the exact value you want to return for at least one of the columns
LIKE	Search for a pattern

SELECT statement with WHERE clause

- Example1 : Display all the book with accession number greater than 400

- SELECT * FROM book WHERE price >600;
- Example2: Display all the books written by “Roger S Presman”
- SELECT title FROM book

  WHERE author =”Roger S Presman”;

- For numeric (int) value don’t put quotes for other values put quotes

The LIKE Condition

- The LIKE condition is used to specify a search for a pattern in a column.

  SELECT column FROM table WHERE column LIKE pattern

- A "%" sign can be used to define wildcards (missing zero or more letters in the pattern) both before and after the pattern.
- A "_" sign can be used to define wildcards (missing one letter in the pattern) both before and after the pattern.

- Example1: Display all the members whose name start will letter "S"

```
SELECT * FROM members   WHERE name LIKE "S%" ;
```

- Example2: Display all the book titles containing "software" word

```
SELECT title FROM book
  WHERE title LIKE "%software%" ;
```

The IN condition

IN

- The IN operator may be used if you know the exact value you want to return for at least one of the columns.
- SELECT column_name FROM table_name WHERE column_name IN (value1,value2,..)

- Example: Display all the books written by “Roger S Presman” or “D W Patterson”

SELECT title, author FROM book WHERE author IN (“Roger S Presman”,” D W Patterson”);

The BETWEEN condition

- **BETWEEN ... AND**
- The BETWEEN ... AND operator selects a range of data between two values. These values can be numbers, text, or dates.
- It selects fields between the test values, including the first test value and excluding the last test value
- SELECT column_name FROM table_name WHERE column_name BETWEEN value1 AND value2

- Example: display accession and title of the books in which accession number is between 101 to 105

SELECT accno,title FROM book WHERE accno BETWEEN 101 AND 105;

Where clause with multiple conditions

- **AND & OR**

  AND and OR join two or more conditions in a WHERE clause.

  The AND operator displays a row if ALL conditions listed are true. The OR operator displays a row if ANY of the conditions listed are true.

- Example1: Display all the books of “computer discipline” published by “Mc Graw Hills”

  SELECT * FROM book

  WHERE author=“Roger S Presman” AND accno=105;

- Example2: Display all the books published by “Mc Graw Hills” or is of “Computer Discipline”

  SELECT * FROM book

  WHERE author=“Roger S Presman” OR accno=105;

Working with Functions

- **SQL has a lot of built-in functions for counting and calculations.**
- **Types of Functions:**
  - ★ Aggregate Functions
  - ★ Scalar functions
- **Aggregate functions**
  - ★ Aggregate functions operate against a collection of values, but return a single value.
  - ★ **Note:** If used among many other expressions in the item list of a SELECT statement, the SELECT must have a GROUP BY clause!!

Some Aggregate Functions in SQL Server

AVG(column)	Returns the average value of a column
COUNT(column)	Returns the number of rows (without a NULL value) of a column
COUNT(*)	Returns the number of selected rows
COUNT(DISTINCT column)	Returns the number of distinct results
MAX(column)	Returns the highest value of a column
MIN(column)	Returns the lowest value of a column
SUM(column)	Returns the total sum of a column

Some Scalar Functions in MySQL

- Scalar functions operate against a single value, and return a single value based on the input value

Function Name	Description
UCASE(c)	Converts a field to upper case
LCASE(c)	Converts a field to lower case
SUBSTRING(c,start,end)	Extract characters from a text field
LENGTH(c)	Returns the length of a text field
MID(c, pos, length)	Extract characters from a text field
ROUND(c,[D])	Rounds a numeric field to the number of decimals specified

Some Scalar Functions in MySQL

Function Name	Description
DATEDIFF (start_date,end_date)	Start_date is subtracted from end_date

Example: Display the days between return date and issue date of all books which were issued

SELECT members_memo, book_accno, datediff(end_date,start_date) AS no_of_days FROM book_hold;

Examples of Aggregate Functions

- Example1: Display the total cost of holdings in library

  SELECT sum(price) from book;
- Example2: Display an average cost of books in your library

  SELECT avg(price) from book;
- Example3: Display total number of members in your library

  SELECT count(*) from members;
- Example4: Display number of unique titles of books in your library

  SELECT count(distinct title) from book;
- Example4: Display highest cost of books in your library

  SELECT max(price) from book;

GROUP BY and HAVING

- Aggregate functions (like SUM, COUNT) often need an added GROUP BY functionality
- **GROUP BY...**
- GROUP BY... was added to SQL because aggregate functions (like SUM) return the aggregate of all column values every time they are called, and without the GROUP BY function it was impossible to find the sum for each individual group of column values.
- SELECT column,SUM(column) FROM table GROUP BY column
- Example: Display number of copies of each book
- SELECT isbn, count(isbn) AS number_of_copies FROM book GROUP BY isbn;

HAVING...

- HAVING... was added to SQL because the WHERE keyword could not be used against aggregate functions (like SUM), and without HAVING... it would be impossible to test for result conditions.

```
SELECT column,aggregate_function(column) FROM table
GROUP BY column
HAVING condition
```

- Example: Display isbn numbers of all books having two or more copies

```
SELECT isbn,count(isbn) AS number_of_books FROM book
GROUP BY isbn
HAVING count(isbn)>=2;
```

Sorting the result - ORDER BY

- The ORDER BY keyword is used to sort the result.
- The default order is in ascending order of values
- We can specify the keyword **DESC** if we want a descending order; the keyword **ASC** can be used to explicitly specify ascending order, even though it is the default
- SELECT column_names FROM table

  [WHERE condition]

  ORDER BY column_name
- Example1: Display the title, accno in alphabetical order of title

  SELECT title,accno FROM book ORDER BY title;
- Example2: Display the title, accno in descending order of title

  SELECT title,accno FROM book ORDER BY title DESC;

Now can you predict the outcome?

- SELECT title,COUNT(isbn) AS numer_of_copies FROM book GROUP BY title,isbn HAVING COUNT(isbn)>=2 ORDER BY title;

Retrieving Data From Multiple Tables

- To retrieve data from two or more tables we need to join them by specifying join condition
- SELECT column1,column2.. FROM table1,table2

  WHERE table1.column = table2.column

- Example: Display the accno, title and name of the members who have issued books at any point of time

  Ans: Here title is available in 'book' table, name of the member is available in 'members' table and issue information is available in 'issue' table. Therefore we need to join all these three table

```
SELECT book.accno, book.title, members.name
FROM book,book_hold,members
WHERE book.accno=book_hold.book_accno AND
    book_hold.members_memo=members.memo;
```

Retrieving Data From Multiple Tables

- NESTED QUERIES: - A complete SELECT query, called a *nested query* , can be specified within the WHERE-clause of another query, called the *outer query*
- Many of the previous queries can be specified in an alternative form using nesting
- IN and EXISTS can be used for nested query (query inside query)
- In general, we can have several levels of nested queries
- Example: Display the list of members who have not issued any book

```
SELECT memo,name FROM members WHERE memo NOT IN
    (SELECT  members_memo FROM book_hold);
```

Retrieving Data From Multiple Tables by JOIN

- An SQL JOIN clause is used to combine rows from two or more tables, based on a common field between them.
- Some Common join:
  - INNER JOIN: Keep only matched rows on both side
  - LEFT JOIN: Keep only matched rows with left side of table
  - RIGHT JOIN: Keep only matched rows with right side of table
  - FULL JOIN: Keep all rows of both side of table, where not matched show as NULL
- **Example**: Display the list of members name who have issued any book

```
SELECT book.accno, book.title, memb.name
FROM book INNER JOIN book_hold bkh ON
    book.accno=bkh.book_accno
INNER JOIN members memb ON memb.memo=bkh.members_memo;
```

Create View

In SQL, a view is a virtual table based on the result-set of an SQL statement.
A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

- **Example**: Display the list of members name who have issued any book

```
CREATE VIEW view_name AS

SELECT book.accno, book.title, memb.name

FROM book INNER JOIN book_hold bkh ON
book.accno=bkh.book_accno

INNER JOIN members memb ON memb.memo=bkh.members_memo;
```

Call View:

```
SELECT * FROM view_name;
```


Thanks