<http://www.studytonight.com/dbms/orderby-clause.php>

Types of databases:

1. **Hierarchical**: this type of DBMS employs the "parent-child" relationship of storing data. Its structure is like a tree with nodes representing records and branches representing fields. The windows registry used in Windows XP is an example of a hierarchical database. Configuration settings are stored as tree structures with nodes.
2. **Network DBMS** - this type of DBMS supports many-to many relations. RDM Server is an example
3. **Relational DBMS** - this type of DBMS defines database relationships in form of tables, also known as relations. does not support many to many relationships. Examples MySQL, Oracle, and Microsoft SQL Server database.
4. **Object Oriented Relation DBMS** this type supports storage of new data types. The data to be stored is in form of objects. PostgreSQL

**What is SQL?**

Structured Query language (SQL) **pronounced as "S-Q-L" or sometimes as "See-Quel**"is actually the standard language for dealing with Relational Databases.

**MySQL**

1. Open source relational database.
2. cross platform
3. MySQL has high performance compared to other relation database systems.
4. Cost effective, it's relatively cheaper in terms of cost when compared to other relational databases

In order to interact with MySQL, you will need a **server access tool**that can communicate with MySQL server. MySQL supports multiple user connections.

Composite values and Atomic values

### 

A **schema** is a collection of database objects (as far as this hour is concerned—tables) associated with one particular database username

Say you have been issued a database username and password by the database administrator. Your username is USER1. Suppose you log on to the database and then create a table called EMPLOYEE\_TBL. According to the database, your table's actual name is USER1.EMPLOYEE\_TBL

* **Data Definition Language (DDL)** - These SQL commands are used for creating, modifying, and dropping the structure of database objects. The commands are CREATE, ALTER, DROP, RENAME, and TRUNCATE.

CADRaT

**create** is a DDL command used to create a table or a database.

**create** database *database-name*;

create database Test;

create table Student(id int, name varchar, age int);

**alter command**

*alter* command is used for alteration of table structures. There are various uses of *alter* command, such as,

* to add a column to existing table
* to rename any existing column
* to change datatype of any column or to modify its size.
* *alter* is also used to drop a column
* alter table Student add(address char);

Using alter command you can rename an existing column. Following is the Syntax,

**alter** table *table-name* **rename** old-column-name to column-name;

Here is an Example for this,

alter table Student rename address to Location;

The above command will rename *address* column to *Location*.

#### To Add column with Default Value

alter command can add a new column to an existing table with default values. Following is the Syntax,

**alter** table *table-name* add(**column-name1** *datatype1* **default** *data*);

Here is an Example for this,

alter table Student add(dob date default '1-Jan-99');

The above command will add a new column with default value to the **Student** table

#### To Drop a Column

alter command is also used to drop columns also. Following is the Syntax,

**alter** table *table-name* drop(column-name);

Here is an Example for this,

alter table Student drop(address);

The above command will drop *address* column from the **Student table**

#### truncate command

*truncate* command removes all records from a table. But this command will not destroy the table's structure. When we apply truncate command on a table its Primary key is initialized. Following is its Syntax,

**truncate** table *table-name*

Here is an Example explaining it.

truncate table Student;

The above query will delete all the records of **Student** table.

**truncate** command is different from **delete** command. delete command will delete all the rows from a table whereas truncate command re-initializes a table(like a newly created table).

**For eg.** If you have a table with 10 rows and an auto\_increment primary key, if you use *delete* command to delete all the rows, it will delete all the rows, but will not initialize the primary key, hence if you will insert any row after using delete command, the auto\_increment primary key will start from 11. But in case of *truncate*command, primary key is re-initialized.

CHAR

1. Used to store character string value of **fixed length**.
2. The maximum no. of characters the data type can hold is **255 characters**.
3. It's **50% faster** than VARCHAR.
4. Uses **static memory allocation**.

VARCHAR

1. Used to store **variable length** alphanumeric data.
2. The maximum this data type can hold is up to
3. Pre-MySQL 5.0.3: **255 characters**.
4. In MySQL 5.0.3+: **65,535 characters** shared for the row.
5. It's **slower** than CHAR.
6. Uses **dynamic memory allocation\***.

**Data Manipulation Language (DML)** - These SQL commands are used for storing, retrieving, modifying, and deleting data.   
These Data Manipulation Language commands are: [SELECT](http://beginner-sql-tutorial.com/sql-select-statement.htm), [INSERT](http://beginner-sql-tutorial.com/sql-insert-statement.htm), [UPDATE](http://beginner-sql-tutorial.com/sql-update-statement.htm), and [DELETE](http://beginner-sql-tutorial.com/sql-delete-statement.htm).

#### INSERT command

Insert command is used to insert data into a table. Following is its general syntax,

**INSERT** into *table-name* values(data1,data2,..)

INSERT into Student values(101,'Adam',15);

INSERT into Student values(102,'Alex',null);

#### Example to Insert Default value to a column

INSERT into Student values(103,'Chris',default)

#### 2) UPDATE command

update Student set age=18 where s\_id=102;

UPDATE Student set s\_name='Abhi',age=17 where s\_id=103;

#### 3) Delete command

#### Example to Delete all Records from a Table

DELETE from Student;

DELETE from Student where s\_id=103;

The above command will delete the record where s\_id is 103 from **Student** table.

**Transaction Control Language (TCL)** - These SQL commands are used for managing changes affecting the data. These commands are COMMIT, ROLLBACK, and SAVEPOINT.

#### Commit command

Commit command is used to permanently save any transaaction into database.

Following is Commit command's syntax,

***commit***;

#### Rollback command

This command restores the database to last commited state. It is also use with savepoint command to jump to a savepoint in a transaction.

Following is Rollback command's syntax,

**rollback** to *savepoint-name*;

#### Savepoint command

**savepoint** command is used to temporarily save a transaction so that you can rollback to that point whenever necessary.

Following is savepoint command's syntax,

**savepoint** *savepoint-name*

#### Example of Savepoint and Rollback

Following is the **class** table,

|  |  |
| --- | --- |
| **ID** | **NAME** |
| 1 | abhi |
| 2 | adam |
| 4 | alex |

Lets use some SQL queries on the above table and see the results.

INSERT into class values(5,'Rahul');

commit;

UPDATE class set name='abhijit' where id='5';

savepoint **A**;

INSERT into class values(6,'Chris');

savepoint **B**;

INSERT into class values(7,'Bravo');

savepoint **C**;

SELECT \* from class;

The resultant table will look like,

|  |  |
| --- | --- |
| **ID** | **NAME** |
| 1 | abhi |
| 2 | adam |
| 4 | alex |
| 5 | abhijit |
| 6 | chris |
| 7 | bravo |

Now **rollback** to **savepoint B**

rollback to B;

SELECT \* from class;

The resultant table will look like

|  |  |
| --- | --- |
| **ID** | **NAME** |
| 1 | abhi |
| 2 | adam |
| 4 | alex |
| 5 | abhijit |
| 6 | chris |

* **Data Control Language (DCL)** - These SQL commands are used for providing security to database objects. These commands are GRANT and REVOKE.
* **Grant :** Gives user access privileges to database.
* **Revoke :** Take back permissions from user

#### To Allow a User to create Table

**grant** create table to *username*;

#### To Grant all privilege to a User

**grant** sysdba to *username*

#### To take back Permissions

**revoke** create table from *username*

#### Example to Perform Simple Calculations using Select Query

SELECT eid, name, salary+3000 from Employee;

The above command will display a new column in the result, showing 3000 added into existing salaries of the employees.

### Like clause

**Like** clause is used as condition in SQL query. **Like** clause compares data with an expression using wildcard operators. It is used to find similar data from the table.

* **Percent sign %** : represents zero, one or more than one character.
* **Underscore sign \_** : represents only one character.
* SELECT \* from Student where s\_name like 'A%';

return all records where **s\_name** starts with character 'A'.

SELECT \* from Student where s\_name like '\_d%';

return all records from **Student** table where **s\_name** contain 'd' as second character.

SELECT \* from Student where s\_name like '%x';

The above query will return all records from **Student** table where **s\_name** contain 'x' as last character.

### Order By Clause

Order by clause is used with **Select** statement for arranging retrieved data in sorted order. The **Order by**clause by default sort data in ascending order. To sort data in descending order **DESC** keyword is used with**Order by** clause.

SELECT \* from Emp **order by** salary

SELECT \* from Emp order by salary DESC;