 records could be considered rows and fields could be considered columns. A database stores a collection of records (rows) and a record is made up of a number of fields (columns). Fields describe what data a record contains.

## ORDER BY Several Columns

The following SQL statement selects all customers from the "Customers" table, sorted by the "Country" and the "CustomerName" column. This means that it orders by Country, but if some rows have the same Country, it orders them by CustomerName:

### Example

SELECT \* FROM Customers  
ORDER BY Country, CustomerName;

## Using Both ASC and DESC

The following SQL statement selects all customers from the "Customers" table, sorted ascending by the "Country" and descending by the "CustomerName" column:

### Example

SELECT \* FROM Customers  
ORDER BY Country ASC, CustomerName DESC;

SQL, or Structured Query Language, is the standard language used to communicate with relational database management systems. It's divided into several sublanguages, each designed to handle specific aspects of database manipulation and interaction. The most commonly referenced SQL sublanguages include DDL, DQL, DML, DCL, and TCL. Here's an overview of each:

DDL (Data Definition Language)

DDL is used for defining and modifying the structure of database objects (like tables, indexes, and schemas). The commands included in DDL allow you to create, alter, and delete database objects. Some common DDL commands are:

CREATE: Used to create a new table or database.

ALTER: Used to modify an existing database object, such as adding, deleting, or modifying columns in an existing table.

DROP: Used to delete an entire table, database, or view.

TRUNCATE: Used to delete all records from a table but does not delete the table itself.

DQL (Data Query Language)

DQL is all about querying the data from the databases. It uses the SELECT statement to retrieve data from one or more tables. The SELECT statement can be elaborated with various clauses like WHERE, GROUP BY, HAVING, and ORDER BY to filter, group, and sort the data, respectively.

DML (Data Manipulation Language)

DML deals with the manipulation of data. It allows you to insert, update, or delete data within existing tables. The primary DML commands are:

INSERT: Adds new rows to a table.

UPDATE: Modifies existing data within a table.

DELETE: Removes existing rows from a table.

DCL (Data Control Language)

DCL includes commands that deal with permissions and access controls to database objects. The main DCL commands are:

GRANT: Gives a user permission to perform certain actions on database objects.

REVOKE: Removes previously granted permissions from a user.

TCL (Transaction Control Language)

TCL is used to manage transactions in a database. These commands can help ensure data integrity by grouping multiple tasks into a single transaction, which is then either committed or rolled back as a whole. Key TCL commands include:

COMMIT: Saves all changes made in the current transaction.

ROLLBACK: Undoes changes made in the current transaction.

SAVEPOINT: Sets a savepoint within a transaction, which can be rolled back to without affecting the entire transaction.

Understanding the distinctions between these sublanguages can help you better grasp the full capabilities of SQL and how it can be used to define, manipulate, query, and control data in relational databases.



You

give examples for every sublanguages in mysql

ChatGPT

ChatGPT

Sure, let's go through examples for each of the sublanguages within the context of MySQL, a popular open-source relational database management system:

DDL (Data Definition Language)

CREATE: Creates a new table named users.

sqlCopy code

CREATE TABLE users (   
id INT AUTO\_INCREMENT PRIMARY KEY,   
name VARCHAR(100),   
email VARCHAR(100) UNIQUE,   
registration\_date DATE   
); 

ALTER: Adds a new column age to the users table.

sqlCopy code

ALTER TABLE users ADD COLUMN age INT; 

DROP: Deletes the users table.

sqlCopy code

DROP TABLE users; 

TRUNCATE: Deletes all rows in the users table, but not the table itself.

sqlCopy code

TRUNCATE TABLE users; 

DQL (Data Query Language)

SELECT: Retrieves all rows and columns from the users table.

sqlCopy code

SELECT \* FROM users; 

SELECT with WHERE: Retrieves only the rows from users where age is greater than 25.

sqlCopy code

SELECT \* FROM users WHERE age > 25; 

DML (Data Manipulation Language)

INSERT: Adds a new row to the users table.

sqlCopy code

INSERT INTO users (name, email, registration\_date, age) VALUES ('John Doe', '[john.doe@example.com](mailto:john.doe@example.com" \t "/Users/karthik/Documents\\x/_blank)', '2023-01-01', 30); 

UPDATE: Updates the age for all users named John Doe.

sqlCopy code

UPDATE users SET age = 31 WHERE name = 'John Doe'; 

DELETE: Deletes all users named John Doe.

sqlCopy code

DELETE FROM users WHERE name = 'John Doe'; 

DCL (Data Control Language)

GRANT: Gives the SELECT permission on the users table to the user john.

sqlCopy code

GRANT SELECT ON database.users TO 'john'@'localhost'; 

REVOKE: Removes the SELECT permission on the users table from the user john.

sqlCopy code

REVOKE SELECT ON database.users FROM 'john'@'localhost'; 

TCL (Transaction Control Language)

COMMIT: Commits a transaction, saving all changes.

sqlCopy code

START TRANSACTION;   
INSERT INTO users (name, email, registration\_date, age) VALUES ('Jane Doe', '[jane.doe@example.com](mailto:jane.doe@example.com" \t "/Users/karthik/Documents\\x/_blank)', '2023-02-01', 28);   
COMMIT; 

ROLLBACK: Rolls back the current transaction, undoing all changes made in the transaction.

sqlCopy code

START TRANSACTION;   
INSERT INTO users (name, email, registration\_date, age) VALUES ('Joe Bloggs', '[joe.bloggs@example.com](mailto:joe.bloggs@example.com" \t "/Users/karthik/Documents\\x/_blank)', '2023-03-01', 29);   
ROLLBACK; 

SAVEPOINT: Sets a savepoint within a transaction to which you can later roll back.

sqlCopy code

START TRANSACTION;   
INSERT INTO users (name, email, registration\_date, age) VALUES ('Alice Smith', '[alice.smith@example.com](mailto:alice.smith@example.com" \t "/Users/karthik/Documents\\x/_blank)', '2023-04-01', 27);   
SAVEPOINT my\_savepoint;   
DELETE FROM users WHERE name = 'Alice Smith';   
ROLLBACK TO my\_savepoint;   
COMMIT;

**Inner Join:**

The INNER JOIN keyword selects records that have matching values in both tables.

**Note:** The INNER JOIN keyword returns only rows with a match in both tables. Which means that if you have a product with no CategoryID, or with a CategoryID that is not present in the Categories table, that record would not be returned in the result.