11-10-2023
BY K.Dhanamjay

```sql
CREATE TABLE example_table (
    id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    birthdate DATE,
    is_active BOOLEAN,
    email VARCHAR(100),
    price DECIMAL(10, 2),
    interests SET('Reading', 'Cooking', 'Traveling')

);
```

we've created a table named **example_table** with the following columns and data types:

- **id**: An integer that auto-increments and serves as the primary key.
- **first_name** and **last_name**: Variable-length character strings for storing names.
- **birthdate**: Date data type for storing birthdates.
- **is_active**: Boolean data type for storing true/false values.
- **email**: Variable-length character string for storing email addresses.
- **price**: Decimal data type for storing prices with up to 10 digits and 2 decimal places.
- **interests**: Set data type for storing a set of interests.

-- Create a table with a column for each data type

```
CREATE TABLE my_table (
  id INT NOT NULL AUTO_INCREMENT,
  name VARCHAR(255) NOT NULL,
  age TINYINT NOT NULL,
  birth_date DATE NOT NULL,
  salary DECIMAL(10,2) NOT NULL,
  active BOOLEAN NOT NULL,
  location POINT NOT NULL,
  favorite_colors JSON NOT NULL,
  PRIMARY KEY (id)
);
```

-- Insert some sample data

```
INSERT INTO my_table (name, age, birth_date, salary, active, location, favorite_colors) VALUES ('John Doe', 30, '1993-08-04', 100000.00, TRUE, POINT(-71.059493, 42.360104), '[{"color": "red"}, {"color": "blue"}]');
```

-- Select all of the data from the table

```
SELECT * FROM my_table;
```

Here is an example of how to use the POINT data type to store the location of a restaurant:

```
CREATE TABLE restaurants (
  id INT NOT NULL AUTO_INCREMENT,
  name VARCHAR(255) NOT NULL,
  address VARCHAR(255) NOT NULL,
  location POINT NOT NULL,
  PRIMARY KEY (id)
);

INSERT INTO restaurants (name, address, location)
VALUES ('Pizza Hut', '123 Main Street, Anytown, CA', POINT(-122.3333, 37.7777));
```

you can use spatial functions to perform queries on the data.

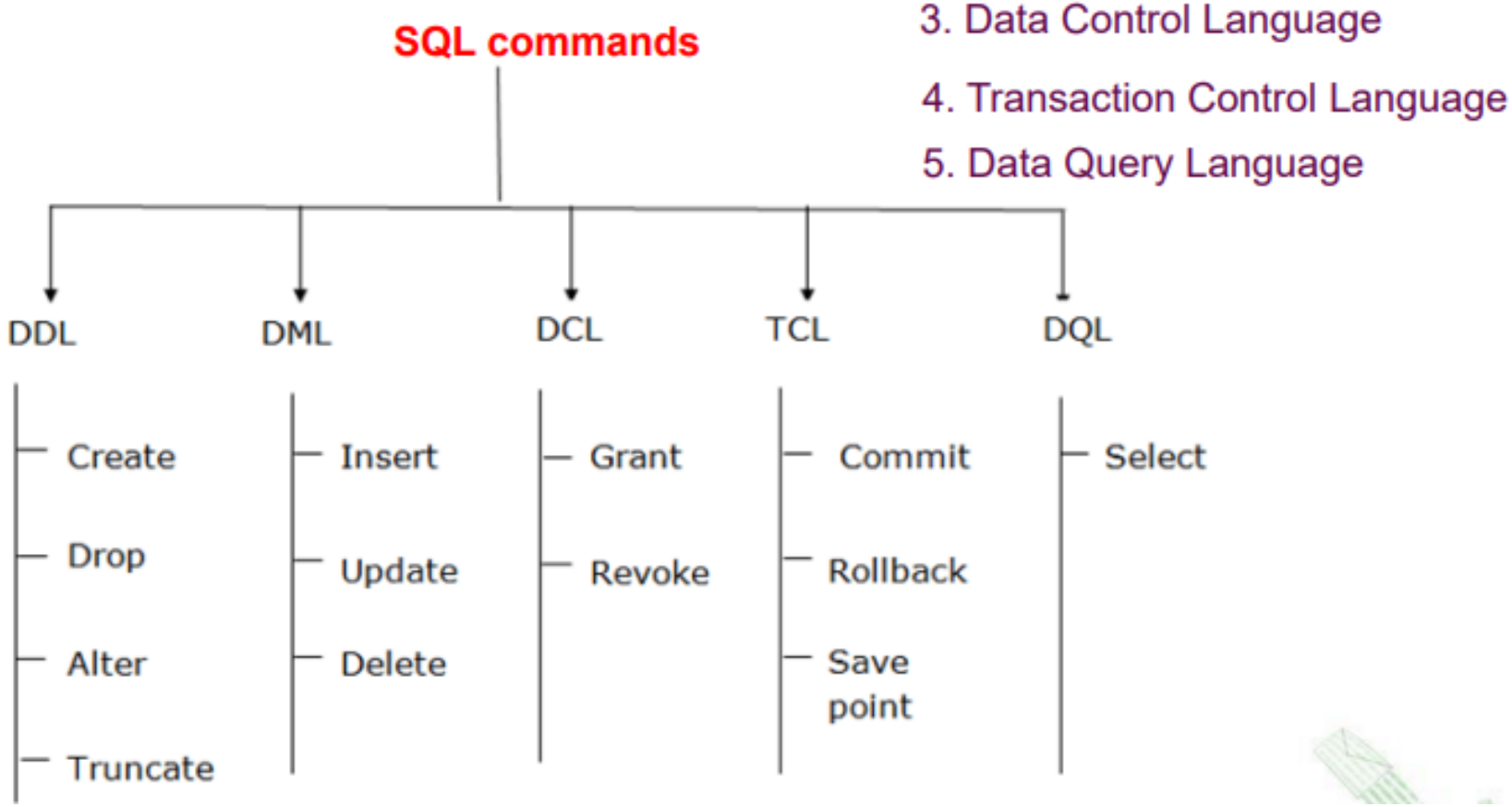
For example, the following query will find all of the restaurants that are within 10 miles of the Eiffel Tower in Paris:

```
SELECT * FROM restaurants
WHERE ST_Distance(location, POINT(2.294481, 48.858370)) < 10000;
```

This query will return a result set that contains all of the restaurants that are within 10 miles of the Eiffel Tower.

Types of SQL Commands

1. Data Definition Language (DDL)
2. Data Manipulation Language
3. Data Control Language
4. Transaction Control Language
5. Data Query Language



There are five types of SQL commands: DDL, DML, DCL, TCL, and DQL

DDL

DDL stands for Data Definition Language.

•It is a subset of SQL (Structured Query Language) used to define and manage the structure of a database.

• DDL statements are used to create, alter, and delete database objects such as tables, views, indexes, and schemas.

• DDL statements do not manipulate or retrieve data; their primary focus is on the database structure.

•DDL statements are generally executed by database administrators or users with appropriate privileges.

Examples of DDL statements include CREATE, ALTER,TRUNCATE and DROP.

CREATE TABLE: Used to create a new table in the database.

Example:

```
CREATE TABLE Customers (id INT, name VARCHAR(50), email VARCHAR(100));
```

ALTER TABLE: Used to modify the structure of an existing table.

Example:

```
ALTER TABLE Customers ADD COLUMN age INT;
```

DROP TABLE: Used to delete a table from the database.

Example:

```
DROP TABLE Customers;
```

DML

DML stands for Data Manipulation Language.

- It is a subset of SQL (Structured Query Language) used to manipulate and retrieve data within a database.
- DML statements are used to insert, update, delete, and retrieve data from database tables.
- DML statements focus on the manipulation and modification of data rather than the database structure
- DML statements are commonly used by application developers and users to interact with the database and perform operations on the data.
- DML statements can be combined with conditions, joins, and other SQL clauses to retrieve or manipulate specific data subsets.

INSERT: Used to add new records or rows into a table.

Example: INSERT INTO Customers (name, email) VALUES ('John Smith', 'john@example.com');

UPDATE: Used to modify existing data in a table.

Example: UPDATE Customers SET email = 'newemail@example.com' WHERE id = 1;

DELETE: Used to remove records or rows from a table.

Example: DELETE FROM Customers WHERE id = 1;

SELECT: Used to retrieve data from one or more tables.

Example: SELECT * FROM Customers WHERE age > 18;