

# Web Database Concepts & Management



## **Fundamental** Web Programming

**Asst. Prof. Manop Phankokkruad, Ph.D.**

School of Information Technology

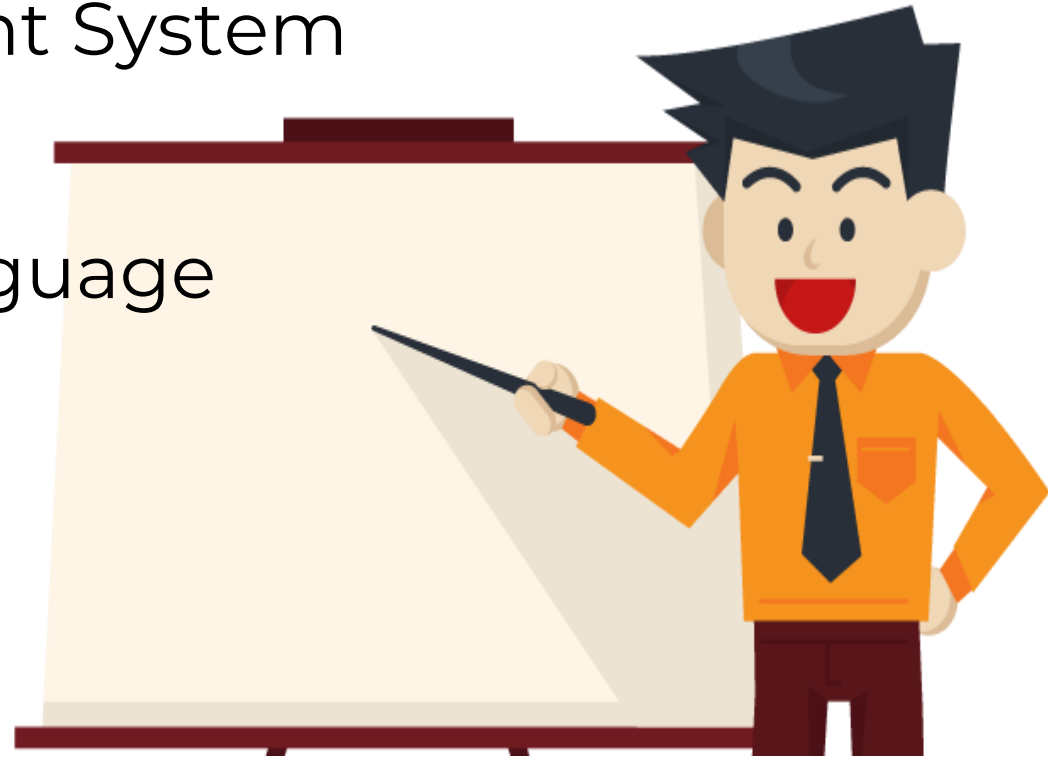
King Mongkut's Institute of Technology Ladkrabang



# Outline

---

1. Introduction to Web Database
2. Database Management System
3. Relational Data Model
4. Structured Query Language



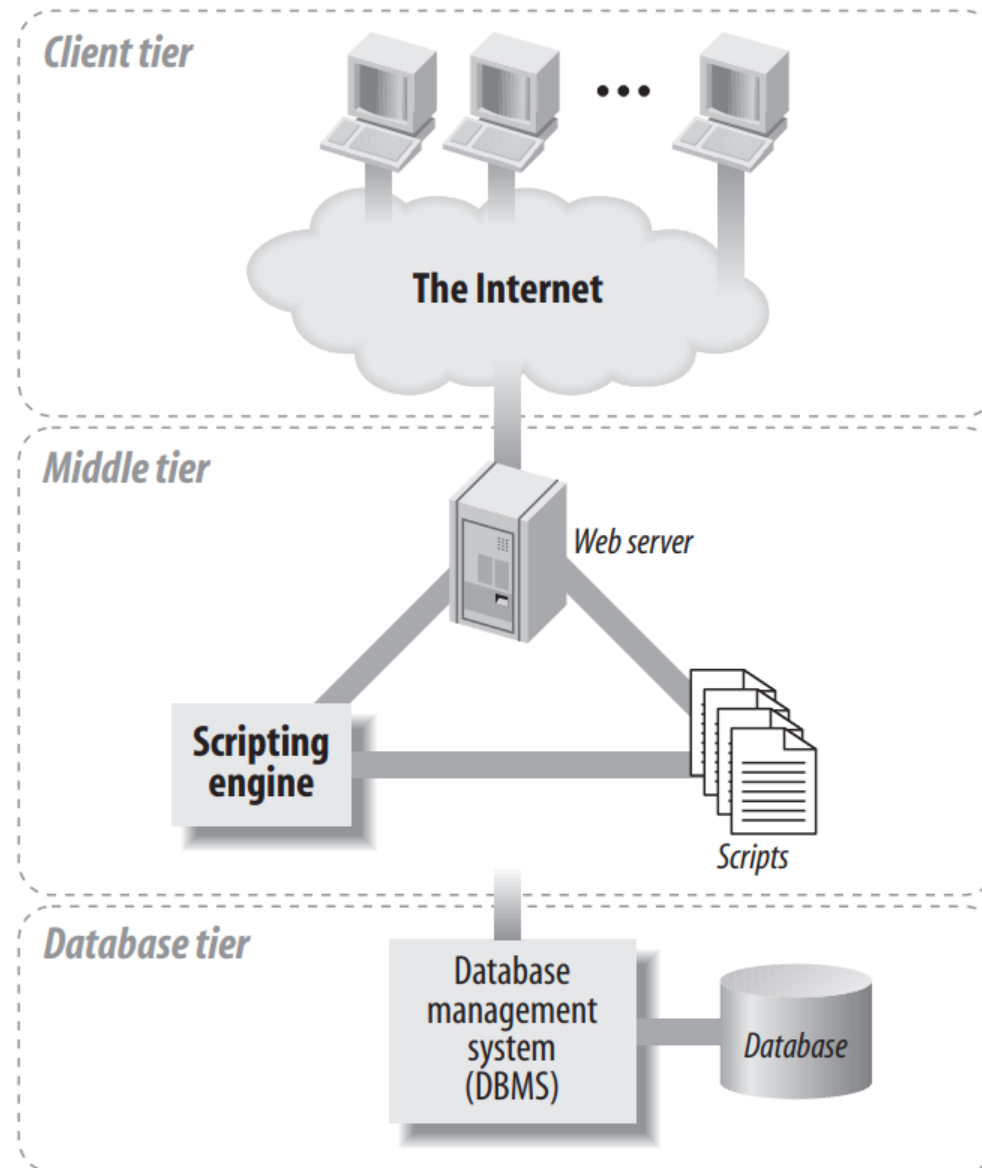
# Introduction

A **web database** is a system for storing information that can then be accessed via a website. For example, an online community may have a database that stores the username, password, and other details of all its members.

- One of the types of web databases that users may be more familiar with is a relational database.
- The **most commonly used** database system is **MySQL, SQL Server, Oracle, and PostgreSQL** due to its integration with the widely used server-side programming languages.

# Introduction : Three-Tier Architectures

- The client tier, usually web browser software that interacts with the application.
- The middle tier contains most of the application logic.
- At the base of an application is the database tier.



A **database management system** (DBMS) is a software that can be used to create and manage databases. DBMS lets users to create a database, store, manage, update/modify and retrieve data from that database by users or application programs. Some examples of DBMS include MySQL, Oracle, PostgreSQL, SQL Server, Microsoft Access, MongoDB.

single file db

NoSQL

- Retrieving data from a database through special type of commands is called querying the database.
- In addition, users can modify the structure of the database itself through a DBMS.



# Database Management System

---

To store and manage data efficiently in the database let us understand some key terms:

1. **Database Schema:** It is a design of the database. It is a skeleton of the database that is used to represent the structure, types of data will be stored in the rows and columns, constraints, relationships between the tables..
2. **Data Constraint:** - Sometimes we put some restrictions on the table that what type of data can be stored in one or more columns of the table, it can be done by using constraints.



# Database Management System

คำอธิบายว่าข้อมูลนั้นคืออะไร

3. **Data dictionary or Metadata:** Metadata is known as the data about the data. The database schema along with different types of constraints on the data is stored by DBMS in the dictionary is known as metadata.
4. **Database instance:** The Database instance refers to the information stored in the database at a given point of time. Thus, it is a dynamic value which keeps on changing. An instance is made up of the background processes needed by the database software.



# Database Management System

---

- 5. Query:** A query is used to access data from the database. Users have to write queries to retrieve or manipulate data from the database.
- 6. Data manipulation:** Users can easily manipulate data using the three main operations that is Insertion, Deletion, and Updation.
- 7. Data Engine:** The set of programs used by a DBMS to create database and handle various queries for data retrieval and manipulation.





# Database Management System

## Advantages of Database

1. Reduces database data redundancy.
2. The database can control data inconsistency.
3. The database facilitates sharing of data.
4. Database enforce standards.
5. The database can ensure data security.
6. Integrity can be maintained through databases.

ความสมบูรณ์ของข้อมูล เช่นชื่อนามสกุลไม่ควรว่าง บางอย่างต้องใส่ข้อมูลให้ครบ



# Relational Data Model

---

# 3

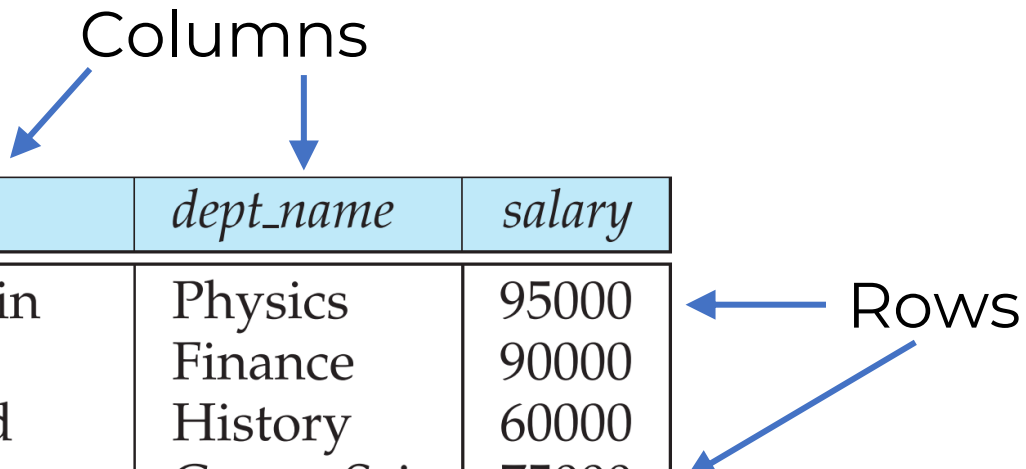
A data model describes the structure of the database, including how data are defined and represented, relationships among data, and the constraints. The most commonly used data model is Relational Data Model.

The relational model is the primary data model for commercial data processing applications. It attained its primary position because of its simplicity, which eases the job of the programmer, compared to earlier data models.



# Relational Data Model - Tables

A relational database is based on the relational model and uses a collection of **tables** to represent both data and the relationships among those data. Each table has multiple columns, and each column has a unique name.



<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000



# Relational Data Model - Tables

**Figure:** Relation containing CD\_NAME, ARTIST\_NAME, and COPYRIGHT\_YEAR attributes

Attribute ↓ ARTIST_NAME:FullName	Attribute name (CD_NAME) ↓ CD_NAME:Title	Domain name (Year) ↓ COPYRIGHT_YEAR:Year
Jennifer Warnes	Famous Blue Raincoat	1991
Joni Mitchell	Blue	1971
William Ackerman	Past Light	1983
Kitaro	Kojiki	1990
Bing Crosby	That Christmas Feeling	1993
Patsy Cline	Patsy Cline: 12 Greatest Hits	1988

Relation

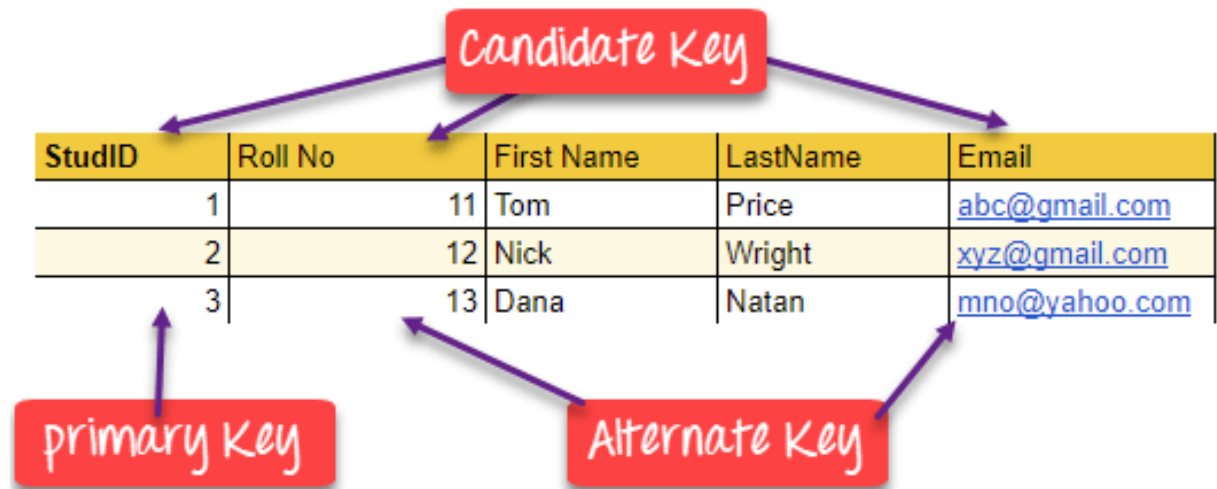
Tuple



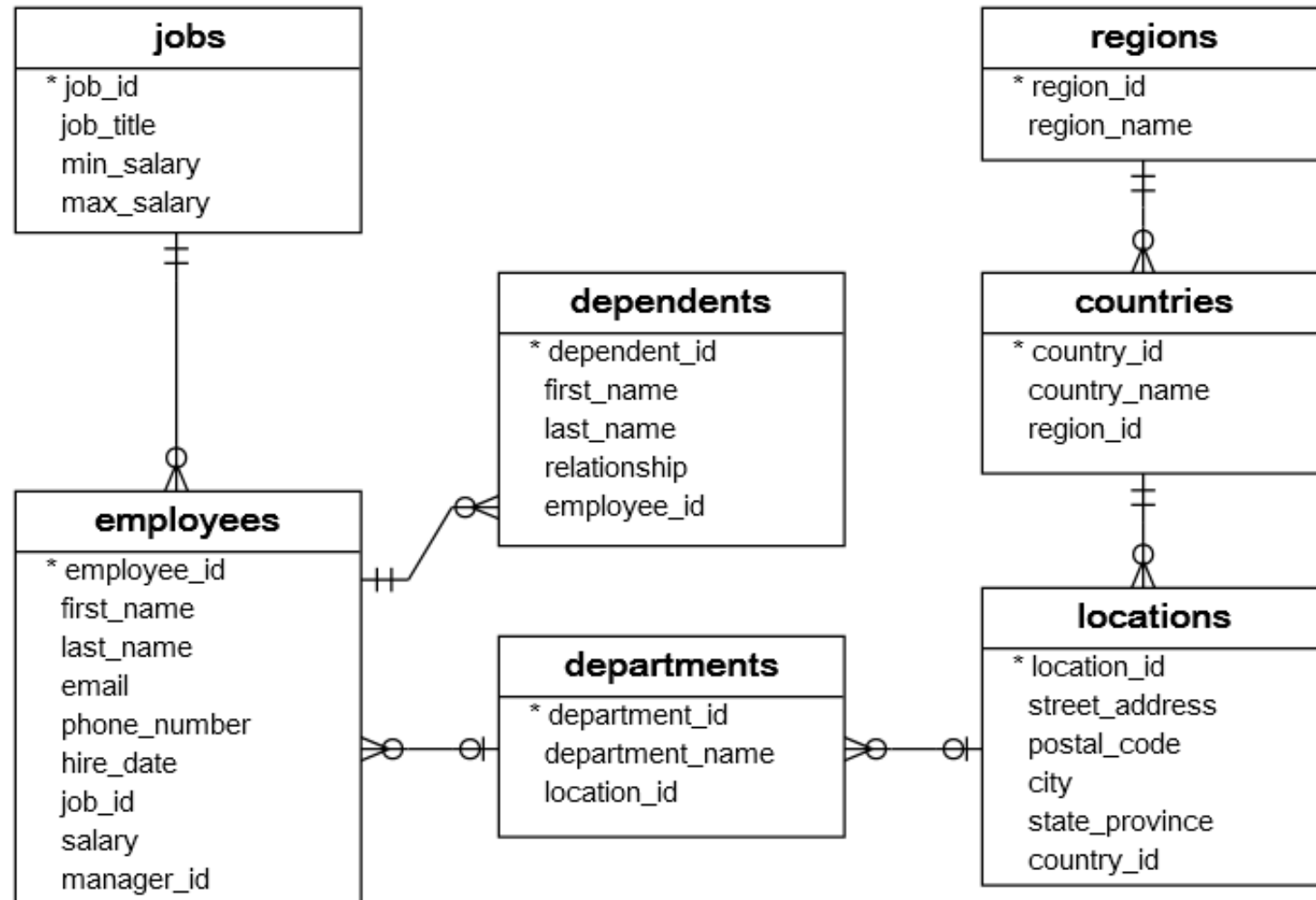
# Relational Data Model - Keys

KEYS in DBMS is an attribute or set of attributes which helps user to identify a row(tuple) in a relation(table). They allow user to find the relation between two tables. Keys help user uniquely identify a row in a table by a combination of one or more columns in that table.

Types of Keys in DBMS includes Super Key, Primary Key, Candidate Key, Alternate Key, Foreign Key, Compound Key, Composite Key, Surrogate Key



# Relational Data Model

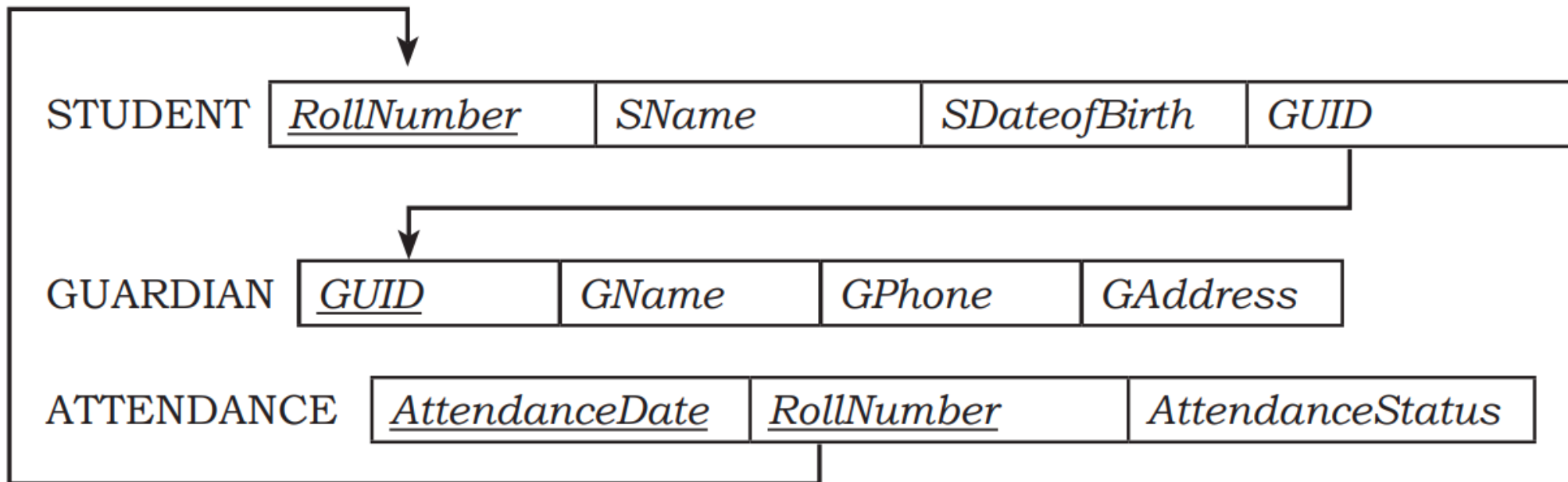


**Figure:** Database using Relational Data Model



# Relational Data Model

**Figure:** StudentAttendance Database with the Primary and Foreign keys



# Structured Query Language

**Structured Query Language** (SQL) is a standard language for storing, manipulating and retrieving data in a relational database management system (RDBMS). It is particularly useful in handling structured data, i.e. data incorporating relations among entities and variables.

The database server applications interface is accessed using SQL. It's a standard query language that's used to define and manipulate databases and data, and it's supported by all popular database servers.





# Structured Query Language

---

**SQL statement** is used to interact with the database with some operations. It is also used to perform specific tasks, functions, and queries of data. SQL can perform various tasks like creating a table, adding data to tables, dropping the table, modifying the table, set permission for users. These SQL statements are categorized into 5 categories:

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



# Structured Query Language - DCL

**Data Control Language (DCL)** statements allow user to control who or what has access to specific objects in the database. With DCL, user can grant or restrict access by using the GRANT and REVOKE statements, the two primary DCL commands.

The DCL statements also allow user to control the type of access each user has to database objects.

## Syntax:

```
GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER;
```

```
REVOKE SELECT, UPDATE ON MY_TABLE FROM USER1, USER2;
```



# Structured Query Language - DDL

**Data Definition Language**(DDL) statements are used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database. List of DDL statements:

- **CREATE**: to create the database or its objects.
- **DROP**: to delete objects from the database.
- **ALTER**: alter the structure of the database.
- **TRUNCATE**: to remove all records from a table, including all spaces allocated for the records.
- **RENAME**: to rename an object existing in the database.



# Structured Query Language - DDL

---

For instance, the following SQL DDL statement defines the department table:

```
CREATE TABLE department (  
    dept_name char(20),  
    building char(15),  
    budget numeric(12,2)  
);
```



# Structured Query Language - DDL

---

The SQL DDL supports a variety of built-in types, including:

- **char**: A fixed-length character string.
- **varchar**: A variable-length character string.
- **int**: An integer.
- **numeric**: A fixed-point number with user-specified precision.
- **real** : Floating-point numbers with machine-dependent precision.
- **float**: A floating-point number with precision.



# Structured Query Language - DQL

**Data Query Language**(DQL) statements are used for performing queries on the data within schema objects. The purpose of the DQL Command is to get some schema relation based on the query passed to it. It includes the SELECT statement. List of DQL:

- **SELECT**: to retrieve data from the database.

```
SELECT instructor.name  
FROM instructor  
WHERE Cinstructor.dept_name = 'History';
```



# Structured Query Language - DML

---

**Data-Manipulation Language** (DML) statements deal with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements. It is the component of the SQL statement that controls access to data and to the database. List of DML commands:

- **INSERT**: to insert data into a table.
- **UPDATE**: to update existing data within a table.
- **DELETE**: to delete records from a database table.



# Structured Query Language - DML

---

The SQL statement is nonprocedural. A query takes as input several tables (possibly only one) and always returns a single table.

Here is an example of an SQL DML statement.

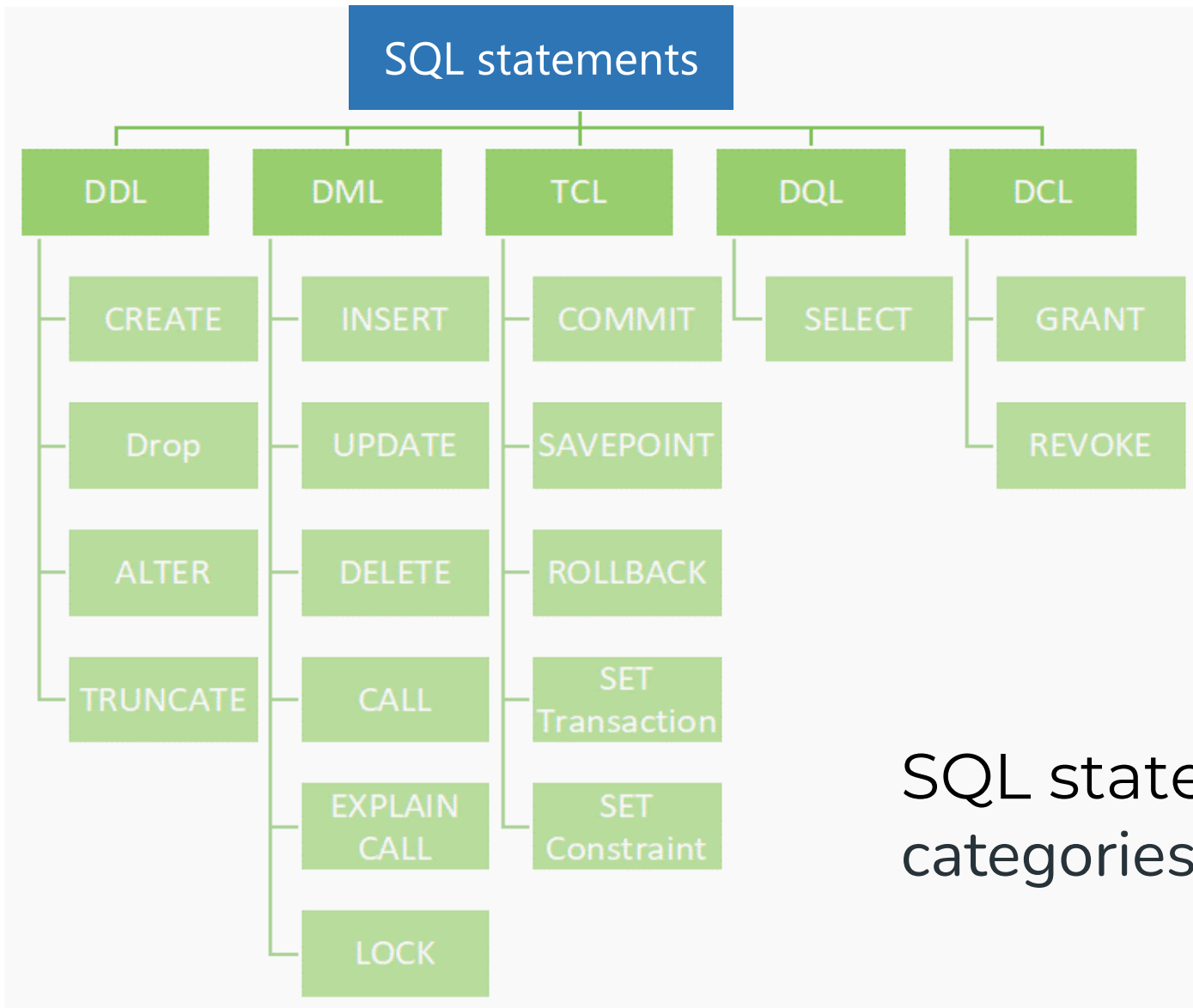
```
insert into course (course_id, title, dept_name, credits)  
  values ('CS-437', 'Database Systems', 'Comp. Sci.', 4);
```

```
insert into course (title, course_id, credits, dept_name)  
  values ('Database Systems', 'CS-437', 4, 'Comp. Sci.');
```





# Structured Query Language



SQL statements has five categories:



# SELECT Statement

The SELECT statement is used to select data from a database. The data returned is stored in a result table, called the result-set.

## SELECT Syntax

```
SELECT column1, column2, ... FROM Table_Name;
```

## Example:

```
SELECT CustomerID , CustomerName, Address FROM Customers;
```

The following SQL statement selects all the records in the Customers table:

```
SELECT * FROM Customers;
```



# INSERT INTO Statement

The INSERT INTO statement is used to insert new records in a table.

## INSERT INTO Syntax

```
INSERT INTO Table_Name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

## Example:

```
INSERT INTO Customers (CustomerName, ContactName, Address,
City, PostalCode, Country)
VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger',
'4006', 'Norway');
```



# UPDATE Statement

The UPDATE statement is used to modify the existing records in a table.

## UPDATE Syntax

```
UPDATE Table_Name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

## Example:

```
UPDATE Customers  
SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'  
WHERE CustomerID = 1;
```



# DELETE Statement

---

The DELETE statement is used to delete existing records in a table.

## DELETE Syntax

```
DELETE FROM Table_Name WHERE condition;
```

## Example:

```
DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';
```



# More Information & References

---

- Avi Silberschatz, Henry F. Korth, and S. Sudarshan, “Database System Concepts”, 7th Edition, McGraw-Hill, 2019.
- Carlos Coronel, Steven Morris, and Peter Rob. “Database Systems: Design, Implementation, and Management”, 9th Edition, Cengage Learning, 2011.

