

## TYPES OF KEYS :

- **Primary Key** → Uniquely identifies each record (e.g., StudentID in Students table).
- **Candidate Key** → A set of unique keys, one of which becomes the Primary Key.
- **Super Key** → A combination of attributes that uniquely identify a record (can include extra attributes).
- **Foreign Key** → A key in one table that refers to the Primary Key in another table (establishes relationships).
- **Composite Key** → A key made of multiple attributes (e.g., StudentID + CourseID for enrollments).
- **Alternate Key** → Candidate keys that are not chosen as the Primary Key.
- **Unique Key** → Ensures values are unique but allows NULLs (unlike Primary Key).

board My courses Support

Q 1 HD Edit mode

database management tool, the tool uses DDL behind the scenes. To truly master SQL, you must master DDL. If you want your DBA (database administrator) to take you seriously, master DDL. Let's begin.

## Lesson Objectives

By the end of this lesson, you will be able to:

1. Describe available SQL data types.
2. Differentiate between DDL and DML.
3. Create a database.
4. Create, edit, and delete tables.
5. Drop a database.

## MySQL Data Types

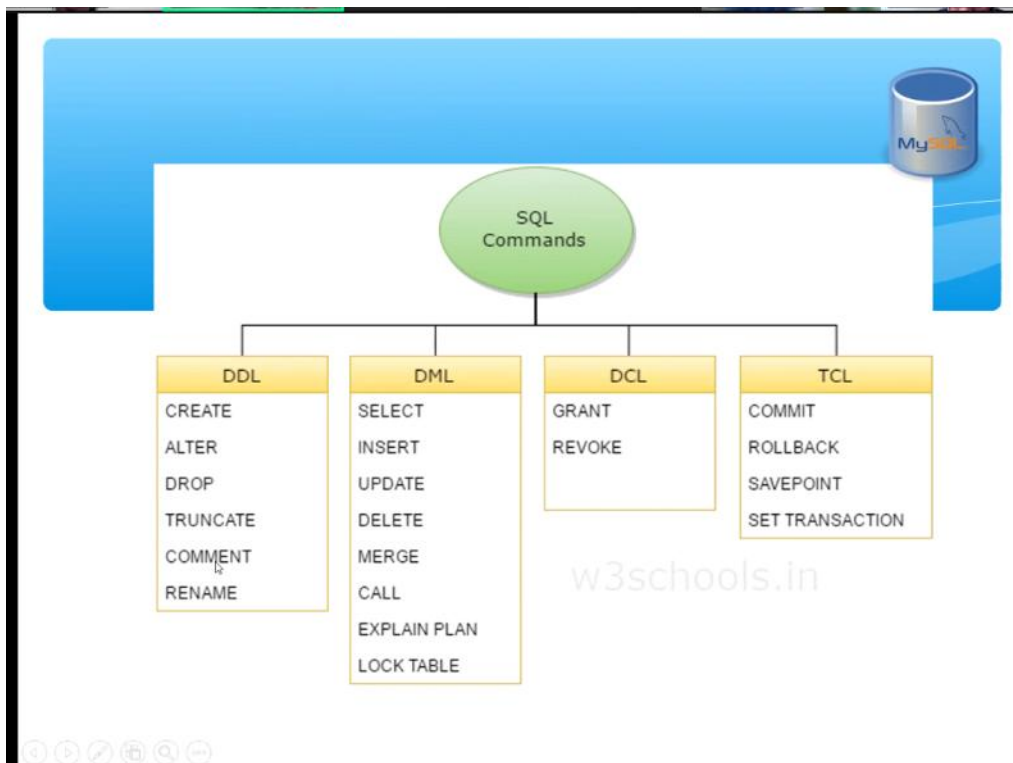
Each column in a table can store one and only one type of data. When we say "type" or "data type", we mean a formal restriction on the values allowed. Restrictions include the size and shape of a value. Consider a person's name, birth date, and height in inches. A name is a series of characters, or a string. We don't want to allow a number as a name. (Though there's nothing stopping a person from entering the value '43770' as their name. It looks numeric, but it's still a series of characters.) A birth date is a date – year, month, and day. The value 'Chris' is not a valid birth date. Height-in-inches is a number. A date is not a number, so it should not be allowed as a value for height.

## Types of SQL Language statements



\* As a language, the SQL standard has three major components:

- \* A **Data Definition Language** (DDL) for defining the database structure and controlling access to the data.
- \* A **Data Manipulation Language** (DML) for retrieving and updating data.
- \* A **Data Control Language** (DCL) concerns with rights, permissions and other controls of the database system.



#### SAMPLE WAY TO CREATE A DATABASE & A TABLE (with INSERTION)

```

1 • CREATE DATABASE sindhu;
2
3 • USE sindhu;
4
5 • CREATE TABLE info (
6     age INT(3),
7     name VARCHAR(20)
8 );
9 • INSERT INTO info (age, name) VALUES (13, 'sindhuja');
10 • select * from info;

```

RESULT after executing line 3 & 10

Line	Time	Command	Rows Affected	Time
3	23:44:33	USE sindhu	0 row(s) affected	0.015 sec
4	23:44:38	select * from info LIMIT 0, 1000	3 row(s) returned	0.015 sec / 0.000 sec

**Result Grid** | Filter Rows:

	age	name
▶	13	sindhuja
	14	sandy
	15	sindhulu

```

create database mthree;
show databases;

use mthree;
CREATE TABLE `Client` (
  ClientId CHAR(36) PRIMARY KEY,
  FirstName VARCHAR(50) NOT NULL,
  LastName VARCHAR(50) NOT NULL,
  BirthDate DATE NULL,
  Address VARCHAR(256) NULL,
  City VARCHAR(100) NULL,
  StateAbbr CHAR(2) NULL,
  PostalCode VARCHAR(10) NULL
);
select * from Client;

```

RR • select \* from Worker;

ClientId	FirstName	LastName	BirthDate	Address	City	StateAbbr	PostalCode
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```

34 • DROP DATABASE IF EXISTS TrackIt;
35 • show databases;
36
37 • CREATE DATABASE TrackIt;
38 • use TrackIt;
39
40 • CREATE TABLE Project (
41     ProjectId CHAR(50) PRIMARY KEY,
42     `Name` VARCHAR(100) NOT NULL,
43     Summary VARCHAR(2000) NULL,
44     DueDate DATE NOT NULL,
45     IsActive BOOL NOT NULL DEFAULT 1
46 );
47
48 • CREATE TABLE Worker (
49     WorkerId INT PRIMARY KEY AUTO_INCREMENT,
50     FirstName VARCHAR(50) NOT NULL,
51     LastName VARCHAR(50) NOT NULL
52 );
53
54 • CREATE TABLE ProjectWorker (
55     ProjectId CHAR(50) NOT NULL,
56     WorkerId INT NOT NULL,
57     PRIMARY KEY pk_ProjectWorker (ProjectId, WorkerId),
58     FOREIGN KEY fk_ProjectWorker_Project (ProjectId)
59         REFERENCES Project(ProjectId),
60     FOREIGN KEY fk_ProjectWorker_Worker (WorkerId)
61         REFERENCES Worker(WorkerId)
62 );
63
64 • CREATE TABLE Task (
65     TaskId INT PRIMARY KEY AUTO_INCREMENT,
66     Title VARCHAR(100) NOT NULL,
67     Details TEXT NULL,
68     DueDate DATE NOT NULL,
69     EstimatedHours DECIMAL(5, 2) NULL,
70     ProjectId CHAR(50) NOT NULL,
71     WorkerId INT NOT NULL,
72     FOREIGN KEY fk_Task_ProjectWorker (ProjectId, WorkerId)
73         REFERENCES ProjectWorker(ProjectId, WorkerId)
74 );
75
76 • show tables;
77 • select * from project;
78 • use TrackIt;
79 • INSERT INTO Worker (WorkerId, FirstName, LastName)
80     VALUES (1, 'Rosamund', 'Pike');
81 • INSERT INTO Worker (WorkerId, FirstName, LastName)
82     VALUES (2, 'Cate', 'Blanchett');
83 • select * from Worker;
84 • RENAME TABLE Worker TO Actresses;
85 • select * from Actresses;

```

```

86 • ALTER TABLE Actresses CHANGE COLUMN WorkerId ActressId INT;
87 • RENAME TABLE Actresses TO Worker;
88 • select * from Worker;
89 • INSERT INTO Worker (FirstName, LastName)
90   VALUES ('Kiera', 'Knightley');
91 • select * from Worker;
92 • INSERT INTO Worker (FirstName, LastName) VALUES
93   ('Sandra', 'Bullock'),
94   ('Demi', 'Moore'),
95   ('Pamela', 'Anderson');
96 • select * from Worker;
97 • INSERT INTO Worker (WorkerId, FirstName, LastName)
98   VALUES (50, 'Sarah', 'Paulson');
99 • select * from Worker;
100 • INSERT INTO Worker (FirstName, LastName)
101   VALUES ('Natalie', 'Portman');
102 • INSERT INTO Project (ProjectId, 'Name', DueDate)
103   VALUES ('db-milestone', 'Database Material', '2018-12-31');
104 • INSERT INTO ProjectWorker (ProjectId, WorkerId)
105   VALUES ('db-milestone', 2);
106 • select * from ProjectWorker;
107 • INSERT INTO Project (ProjectId, 'Name', DueDate)
108   VALUES ('kitchen', 'Kitchen Remodel', '2025-07-15');
109
110 • INSERT INTO ProjectWorker (ProjectId, WorkerId) VALUES
111   ('db-milestone', 1),

```

On executing :

```

77 • select * from project;
78 • use TrackIt;

```

ProjectId	Name	Summary	DueDate	IsActive
db-milestone	Database Material	NULL	2018-12-31	1

```

115 • select * from Worker;

```

WorkerId	FirstName	LastName
1	Rosamund	Pike
2	Cate	Blanchett
3	Kiera	Knightley
4	Sandra	Bullock
5	Demi	Moore
6	Pamela	Anderson
50	Sarah	Paulson
51	Natalie	Portman

```

116 • select * from ProjectWorker;
117

```

ProjectId	WorkerId
db-milestone	2