The different types of keys in an RDBMS are as follows:

KEY	DESCRIPTION
Primary Key	A primary key is an attribute or a group of attributes that can uniquely identify tuples within the relation.
Candidate Key	A candidate key is one that is capable of becoming the primary key (*i.e.*, candidate for primary key position).
Alternate Key	A candidate key that is not the primary key is called an alternate key.
Foreign Key	A non-key attribute whose value is derived from the primary key of some other table is known as foreign key in its current table.

Let us discuss these keys in detail.

1. **Primary Key:** A primary key is a set of one or more attributes/fields which uniquely identifies a tuple/row in a table. The salient features of a primary key are as follows:

   (a) It must contain unique values, *i.e.*, non-redundant.

   (b) It arranges the table in its own order.

   (c) It cannot be re-declared or left null.

   (d) One table can have only one primary key; however, primary key can be a combination of more than one field. For example, roll_number along with admission_no can be combined together and can be declared as a primary key in the relation Student. In the table Item given below, Item_id is the primary key while Supp_id (supplier id) is the primary key in the table Supplier.
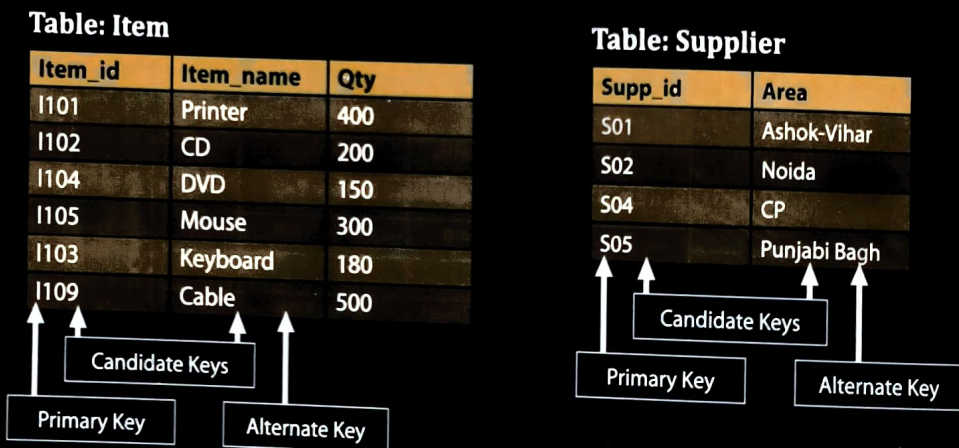

Fig. 7.12: Keys in a Database (Tables)

2. **Candidate Key:** A candidate key refers to all the attributes in a relation that are candidates or are capable of becoming a primary key. We mark it virtually.

   In the given Item table, Item_id and Item_name are the candidate keys. Out of these keys, Item_id is the primary key and Item_name becomes the alternate key. Similarly, in the case of Supplier relation, Supp_id and Area are the candidate keys, Supp_id is the primary key and Area becomes the alternate key. Thus, the equation becomes:

<div align="center">

Candidate Keys – Primary Key = Alternate Key

</div>

3. **Alternate Key:** A candidate key that is not the primary key is called an alternate key. In other words, any attribute that is a candidate for the primary key, *i.e.*, which is capable of becoming a primary key but is not a primary key, is an alternate key. For example, in a Customer table, cust_name is the alternate key. Similarly, in the given table Item, Item_name becomes the alternate key.

4. **Foreign Key:** A foreign key is a non-key attribute whose value is derived from the primary key of another table; in other words, a primary key in some other table having relationship with the current or original table.

Foreign Key

Table: Employee

Item_id	Emp_name	Desig_code
E01	Ankur Mehta	Mgr
E02	Deepika Gupta	Dir
E04	Arnav Bansal	Asst_mgr
E03	Harshit Singh	Acc
E05	Kirti Dubey	Mgr

Table: Department

Desig_code	Designation
Mgr	Manager
Dir	Director
Acc	Accountant
Asst_mgr	Assistant Manager

In the above table, Desig_code is the primary key in the table Department which, when related with the table Employee, becomes a foreign key to it.

This is to keep in mind that as per the Referential integrity constraint, any value which does not exist as a primary key in the parent table (**Department**) cannot be taken in the child table (**Employee**). As shown alongside, you cannot insert a new employee record with Desig_code as Sales_hd (Sales Head) in the Employee table, as there is no such Desig_code existing in the base table **Department**, otherwise, it will violate the Referential integrity constraint and shall result in an error.

Item_id	Item_name	Desig_code
E01	Ankur Mehta	Mgr
E02	Deepika Gupta	Dir
E04	Arnav Bansal	Asst_mgr
E03	Harshit Singh	Acc
E05	Kirti Dubey	Mgr
E06	Geetika	Sales_hd

X
Not Permitted

Thus, the following **points are to be remembered while defining Primary key and Foreign key**:

- Records cannot be inserted in the Child table if the foreign key value does not exist as the primary key in the Parent table, as explained in the above example for Desig_code as Sales_hd (Sales Head).
- Records in the Parent table cannot be deleted if the corresponding records already exist in the Child table. In other words, you cannot delete a record from the Parent table (say, Acc as Desig_code) as its corresponding record (for Harshit Singh) is already defined in the Child table. If you wish to do this, you have to delete the corresponding record from Child table as well and then delete it from the Parent table.
- This is because the purpose of the foreign key is to ensure the referential integrity of the data values that are supposed to appear in the tables, or subsequently in the database.

CTM: Referential Integrity is a constraint in the database that enforces the relationship between two tables. The Referential Integrity constraint requires that values in a foreign key column must either be present in the primary key that is referenced by the foreign key or they must be null.

This was all about the database concepts. We will now learn about their implementation using SQL.

7.6 INTRODUCTION TO SQL

Database Management System (DBMS) is not a new concept and was first implemented in the 1960's. With time, database technologies evolved a lot while usage and expected functionalities of databases increased immensely. In the current scenario, there is no organization which does not manage its data and records using some type of DBMS. An online telephone directory uses DBMS to store data pertaining to people, phone numbers and other contact details. Apart from this, your electricity service provider uses a DBMS to manage billing, client-related issues, to handle fault data, etc., not to forget Facebook—it needs to store, manipulate and present data related to its members, their friends, member activities, messages, advertisements and a lot more.

Online Telephone Directory

Electricity Billing System

All these real-life applications require a DBMS to manipulate and handle this enormous data. A DBMS requires some language to handle and manipulate its data, which is known as Structured Query Language (SQL). The following topic deals exclusively with relational databases, their tables and retrieving data using SQL.

7.7 OVERVIEW OF SQL AND MySQL

SQL (Structured Query Language) is a standard language for accessing and manipulating databases. SQL commands are used to create, transform and retrieve information from Relational Database Management Systems and are also used to create interface between a user and database. By using SQL commands, one can search for any data in the database and perform other functions like creating tables, adding records, modifying data, removing rows, dropping tables, etc.

Fig. 7.13: Structured Query Language

MySQL is an open-source and freely-available Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). It provides excellent features for creating, storing, maintaining and accessing data, stored in the form of databases and their respective tables. A single MySQL database can store several tables at a time and can store thousands of records in it.

Being an open-source software, it can be freely and easily downloaded from the site www.mysql.org. It is fully secured, reliable and fast, and possesses far better functionalities than many other commercial RDBMSs available in the market. Originally, MySQL was developed and supported by a Sweden-based company, MySQL AB, which was bought by Sun Microsystems. In 2010, Oracle acquired Microsystems. The chief inventor of MySQL was Michael 'Monty' Widenius.

7.8 FEATURES OF SQL

SQL is the most common language used to create, operate, update, manipulate and communicate with a database.

In 1970, SQL was developed by Donald D. Chamberlin and Raymond F. Boyce at IBM. Thus, SQL is a fourth generation non-procedural language that is used to create, manipulate and process the databases (relations). It has the following salient features and strong processing capabilities:

➤ It can retrieve data from a database through Query processing.

➤ It can insert records in a database.

➤ It can update records in a database.

➤ It can create new databases and modify the existing ones.

➤ It can create new tables in a database.

➤ It can create views in a database.

➤ It allows modifying the security settings of the system.

> **CTM:** SQL (Structured Query Language) is a unified, non-procedural language used for **creating, accessing,** handling and managing data in relational databases.

7.9 ADVANTAGES OF SQL

SQL has the following advantages:

1. **Ease of use:** It is very easy to learn and use and does not require high-end professional training to work upon it.

2. Large volume of databases can be handled quite easily.

3. **No coding required:** It is non-procedural and a unified language, *i.e.*, we need not specify the procedures to accomplish a task but only need to give a command to perform the activity.

4. SQL can be linked to most of the other high-level languages which makes it the first choice for database programmers.

5. **Portable:** It is compatible with other database programs like Dbase IV, FoxPro, MS Access, DB2, MS SQL Server, Oracle, Sybase, etc.

6. SQL is not a case-sensitive language, *i.e.*, both capital and small letters are recognized.

7.10 CLASSIFICATION OF SQL STATEMENTS

SQL is the language used to interact with the database. The SQL statements or commands that we type are the statements that are regarded as the instructions to the database. SQL provides different types of statements or commands for different purposes. These statements are classified into the following categories:
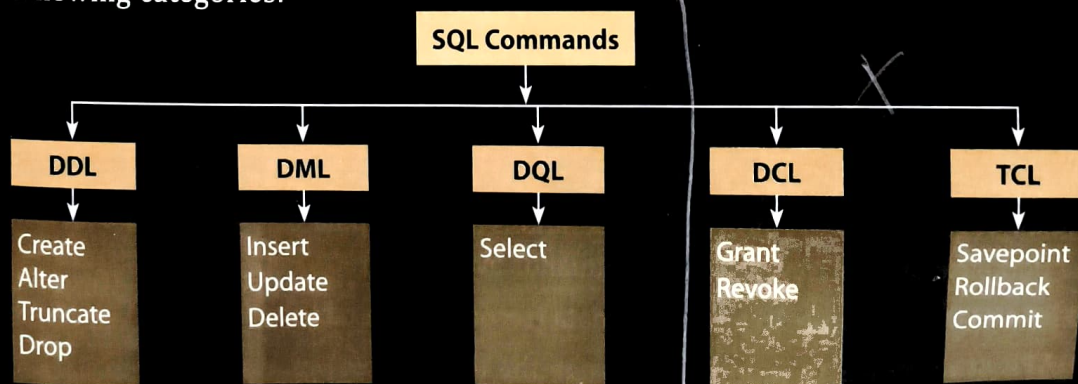
SQL Commands				
DDL	DML	DQL	DCL	TCL
Create Alter Truncate Drop	Insert Update Delete	Select	Grant Revoke	Savepoint Rollback Commit

Fig. 7.14: Classification of SQL Statements

DCL and TCL are beyond the scope of this book. So, we shall be discussing only DDL, DML and DQL commands in detail.

7.10.1 Data Definition Language (DDL) Commands

The DDL part of SQL permits database tables to be created or deleted. It also defines indices (keys), specifies links between tables and imposes constraints on tables. It contains necessary statements for creating, manipulating, altering and deleting the table.
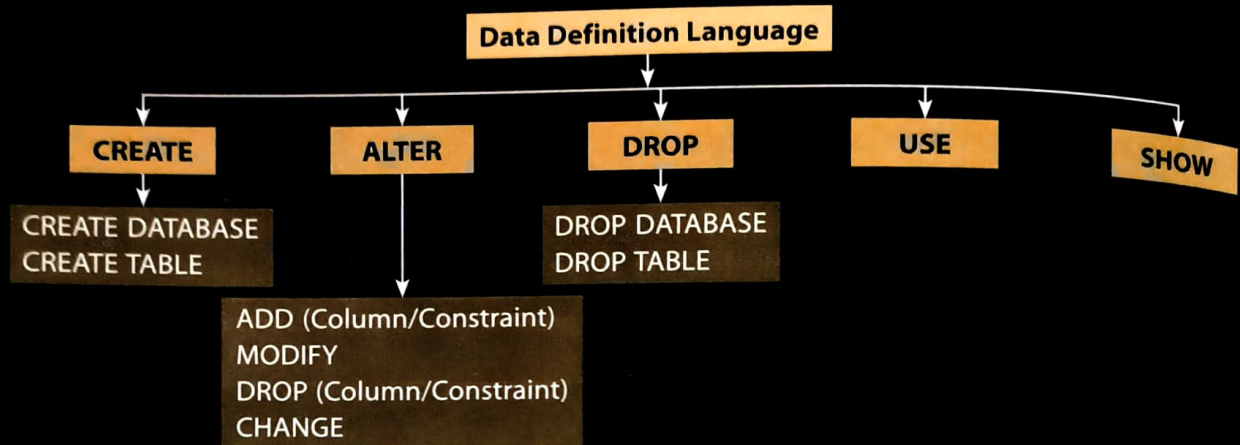

Fig. 7.15: DDL Commands in SQL

Examples of DDL commands in SQL are:

➢ **CREATE DATABASE:** creates a new database.

➢ **USE command:** to select and open an already existing database.

➢ **SHOW command:** to display all the tables in an existing database.

➢ **CREATE TABLE:** creates a new table in an existing database.

➢ **ALTER TABLE:** modifies the structure of a table.

➢ **DROP TABLE:** deletes a table.

CTM: The DDL command lets us define the database structure and its related operations.

The DDL provides a set of definitions to specify the storage structure and access methods used by the database system and also defines proper and relevant data types.

> **Learning Tip:** The query and update commands form the DML part of SQL. They enable the user to access or manipulate data stored in a database.

7.10.2 Data Manipulation Language (DML) Commands

A Data Manipulation Language (DML) is a part of SQL that helps a user manipulate data. The DML statements are executed in the form of queries which are handled by the DML compiler. It contains the set of statements to:

1. Insert data into the tables of the database.
2. Delete data from the tables of the database.
3. Update data among the rows/records in the tables of the database.

DML commands carry out query-processing operations and manipulate data in the database objects. Several DML commands available are:

1. **INSERT INTO statement:** To insert new data (record) into a table.

2. **UPDATE statement:** To modify or change the data (tuple) in a table (not modifying the data type of column).

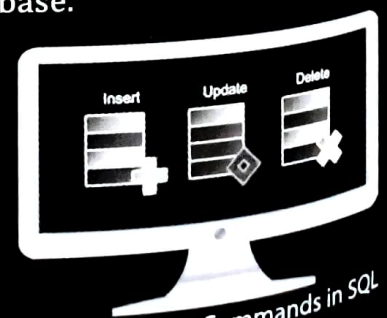3. **DELETE:** To delete data (tuple) from a table (not deleting a column).

Fig. 7.16: DML Commands in SQL

Table 7.2: Difference between DDL and DML Commands

DDL Commands	DML Commands
1. DDL stands for Data Definition Language.	1. DML stands for Data Manipulation Language.
2. These commands allow us to perform tasks related to data definition, *i.e.*, related to the structure of the database objects (relations/databases).	2. These commands are used to manipulate data, *i.e.*, records or rows in a table or relation.
3. The examples of DDL commands are Create, Alter, Drop, etc.	3. The examples of DML commands are Insert into, Update, Delete, etc.
4. DDL is not further classified.	4. DML commands are further classified into two types: (a) Procedural DMLs (b) Non-Procedural DMLs

7.10.3 Data Query Language (DQL) – SELECT Command

One of the most important tasks when working with SQL is to generate queries and retrieve data. A query is a command given to get a desired result from the database table. The SELECT command is used to query or retrieve data from a table in the database. It is used to retrieve a subset of records from one or more tables. The SELECT command can be used in various forms:

Syntax of **SELECT** command:

   SELECT <column-list>FROM<table-name>;

column-list includes one or more columns from which data is retrieved.

We will be discussing the **SELECT command** in detail in the successive sections.

Let us start implementing SQL using MySQL as the platform.

7.11 MySQL

MySQL is an open-source and freely available Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). It provides excellent features for creating, storing, maintaining and accessing data, stored in the form of databases and their respective tables. A single MySQL database can store several tables at a time and can store thousands of records in it.

Being an open-source software, it can be freely and easily downloaded from the site *www.mysql.org*. MySQL is developed and supported by a Sweden-based company, MySQL AB. It is fully secured, reliable, and fast, and possesses far better functionalities than many other commercial RDBMS available in the market. MySQL database system works upon Client/Server architecture. It constitutes a MySQL server which runs on a machine containing the databases and MySQL databases (clients), which are connected to these server machines over a network.

☞ **Advantages of MySQL:** MySQL provides the following salient features and advantages:
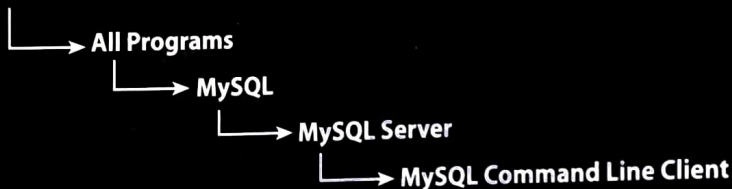
1. **Reliability and Performance:** MySQL is a very reliable and high-performance Relational Database Management System.

2. **Modifiable:** Being an open-source software, MySQL comes with its source code; so, it is easily modifiable and we can recompile its associated source code.

3. **Multi-Platform Support:** MySQL supports several different platforms like UNIX, Linux, Mac OS and Microsoft Windows.

4. **Powerful Processing Capabilities:** MySQL is a powerful, easy, compatible and fast Relational Database Management System. It can handle complicated corporate applications and processing requirements.

5. **Integrity (Checks):** MySQL provides various integrity checks/constraints in order to restrict the user input and processing.

6. **Authorization:** MySQL provides DDL commands to check for user authentication and authorization by restricting access to relations and views.

7. **Powerful Language:** All SQL operations are performed at a prescribed and fixed level, *i.e.,* one SELECT command can retrieve data from multiple rows and one MODIFY command can edit multiple rows at a time. These features make SQL a very powerful language as compared to other languages where one command can process only a single record at a time.

8. **Reliable:** SQL provides a high level of well-defined set of commands that provides the desirable results without any ambiguity.

9. **Freedom of Data Abstraction:** SQL provides a greater degree of abstraction freedom compared to any other procedural language.

10. **Complete Language for a Database:** Apart from being a strong query processing language, it can also be used to create, insert, delete and control access to data in databases.

7.11.1 Starting MySQL Database

MySQL is an open-source database system. You can download and install it directly from the internet. After installing, you need to start working with MySQL by following the given steps:

Start
 └──▶ **All Programs**
   └──▶ **MySQL**
     └──▶ **MySQL Server**
       └──▶ **MySQL Command Line Client**

Alternatively, (for Windows 7 and above)

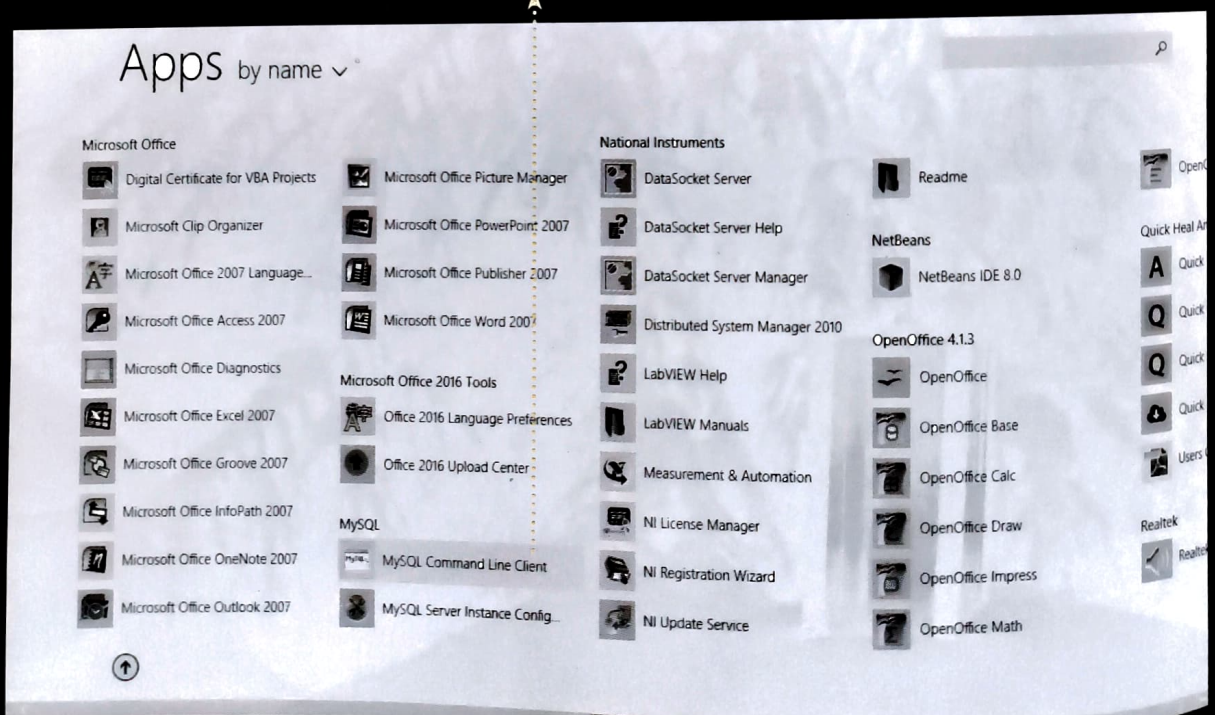Start → Apps by name → MySQL Command Line Client

Fig. 7.17: Steps to Start MySQL

After opening MySQL, the screen of the MySQL command prompt appears where you need to specify a password to work with it.

After entering the password, the MySQL prompt appears where you start typing the SQL commands as shown in Fig. 7.18. In order to come out of MySQL application, you can type quit in front of the mysql> command prompt, as shown in Fig. 7.19.
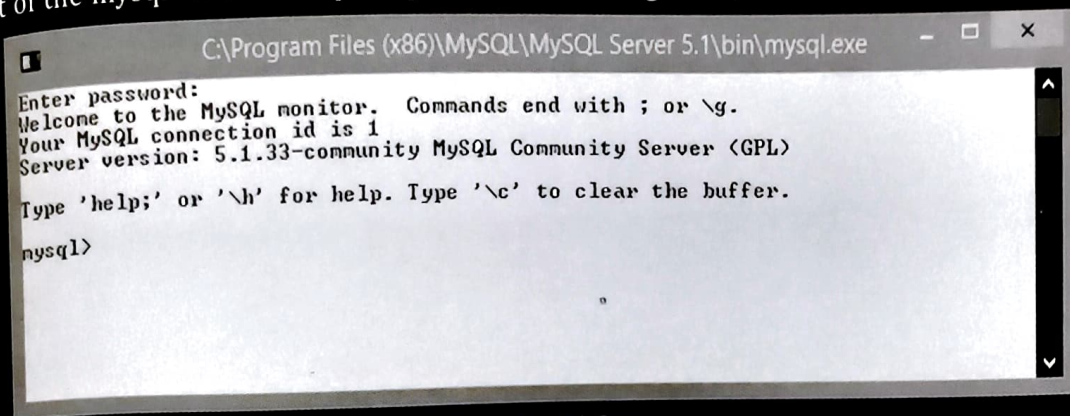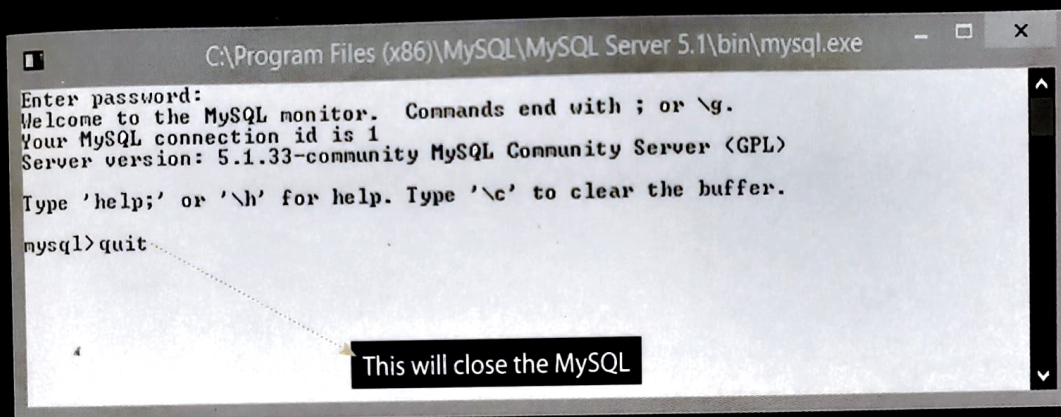

Fig. 7.18: MySQL Prompt

Fig. 7.19: Closing MySQL

Learning Tips:
1. Some database systems require a semicolon (;) at the end of each SQL statement.
2. Semicolon is the standard way to separate each SQL statement in database systems that allows more than one SQL statement to be executed in the same call to the server.
3. SQL is NOT case-sensitive; select is the same as SELECT.

7.12 SQL DATA TYPES

Just like any other programming language, the facility of defining data of various types is available in SQL also. SQL supports the following data types for the specification of various data-items or fields of a relation/table. In SQL, each column of the table is assigned a data type which conveys the kind of value that will be stored in the column.
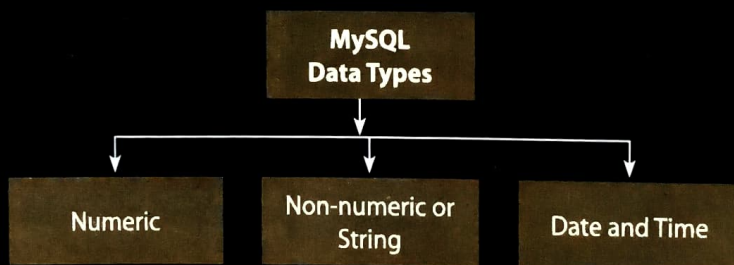

Fig. 7.20: MySQL Data types