**Relational Data Model**

Nowadays, the relational model is the essential data model for commercial data processing applications, which achieved its primary position because of its simplicity, which makes the job of the programmer easy, in contrast to earlier data models such as the network model or the hierarchical model. In this chapter, you will study the essential and primary uses of the relational model. A substantial theory exists for relational databases.

The Relational Database Management System (RDBMS) has become the leading data-processing software in use nowadays with approximated new license sales of between US$6 billion and US$10 billion per year. This software signifies the second generation of DBMSs and is based on the relational data model proposed by Mr. E. F. Codd in the year 1970.

**What is Relational Model?**

The relational model is the theoretical basis of relational databases, which is a technique or way of structuring data using relations, which are grid-like mathematical structures consisting of columns and rows. You might be very familiar with the physical demonstration of a relation in a database - which is known as a table.

In the relational model, all data is logically structured within relations, i.e., tables, as mentioned above. Each relation has a name and is formed from named attributes or columns of data. Each tuple or row holds one value per attribute. The greatest strength of the relational model is the simple logical structure that it forms. Behind this simple structure is a sophisticated theoretical foundation that is lacking in the first generation of DBMSs.

**The relational model's objectives were specified as follows:**

* To allow a high degree of data independence, application programs must not be affected by alterations to the internal data representation, mostly by changes to file organizations or access paths.
* To provide considerable grounds for dealing with data semantics, reliability, and redundancy problems.

**Real Life structure of a Relational Database**

In general, a row in a table signifies a relationship among a group of values. Since a table is a collection of such relationships, there is a close connection amongst the concept of the table and the mathematical concept of relation, from which the relational data model gets its name. In mathematical terminology, a tuple is simply a sequence or list of values. A relationship between n values is indicated mathematically by an n-tuple of values, i.e., a tuple with n values, corresponds to a row in a table.

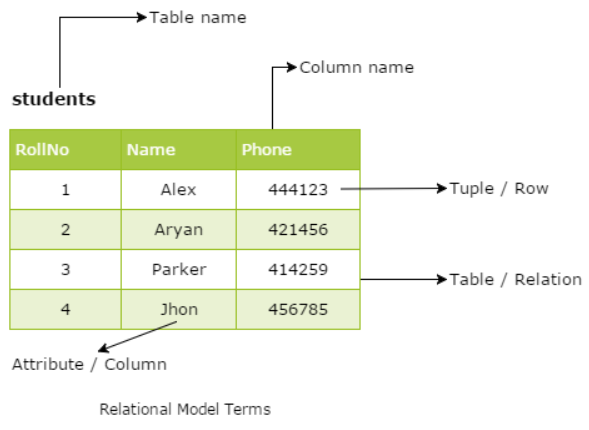
## **Database Schema**

When you talk about the database, you must distinguish between the database schema, which is the logical blueprint of the database, and the database instance, which is a snapshot of the data in the database at a given instant in time. The concept of a relation corresponds to the programming language notion of a variable. In contrast, the concept of a relation schema corresponds to the programming languages' notion of the type definition. In other words, a database schema is a skeletal structure that represents the logical view of the complete database. It describes how the data is organized and how the relations among them are associated and formulates all the constraints that are to be applied to the data.

In general, a relation schema consists of a directory of attributes and their corresponding domain.

**Some Common Relational Model Terms**

* Relation: A relation is a table with columns and rows.
* Attribute: An attribute is a named column of a relation.
* Domain: A domain is the set of allowable values for one or more attributes.
* Tuple: A tuple is a row of a relation.



**Degree**

Degree in SQL could be defined as the total number of entities that are involved in the relationship and it is usually 2 (binary relationship), and there is a possibility of Unary, and higher degree relationships can exist.

**Cardinality**

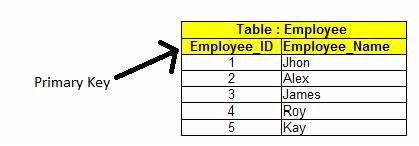
Cardinality in SQL, could be defined as the number of entities that are involved in a relationship. Basically, there are 3 types of Cardinality for binary relationships; one-to-one (1:1), one-to-many (1:n), and many-to-many (n:x).

**Keys**

A key is a single or combination of multiple fields in a table. It is used to fetch or retrieve records/data-rows from data table according to the condition/requirement. Keys are also used to create a relationship among different database tables or views.

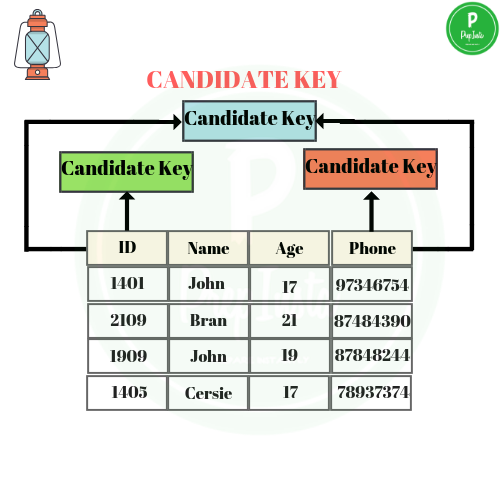
## **Primary Key**

Primary key is a set of one or more fields/columns of a table that uniquely identify a record in a database table. It can not accept null, duplicate values. Only one Candidate Key can be Primary Key.



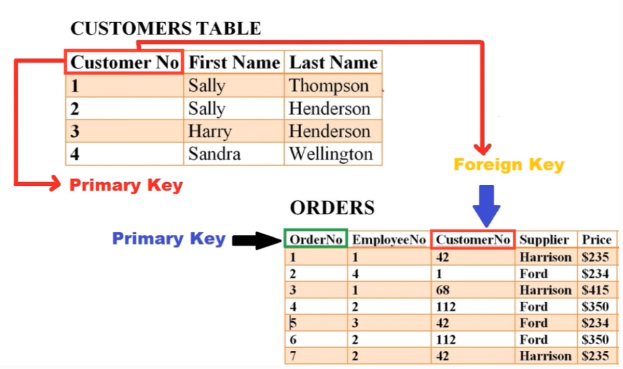
## Candidate Key

A Candidate Key is a set of one or more fields/columns that can identify a record uniquely in a table. There can be multiple Candidate Keys in one table. Each Candidate Key can work as Primary Key.



## **Foreign Key**

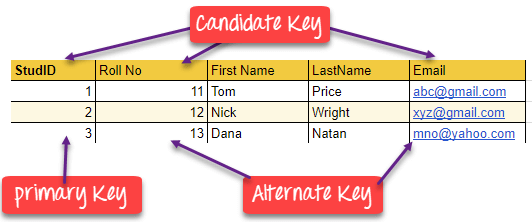
Foreign Key is a field in a database table that is Primary key in another table. It can accept multiple null, duplicate values.



## **Alternate key**

Sometimes more than one column is having the capability to uniquely identify a row but they might not be selected as primary keys this type of keys on nothing but alternate keys.

Alternate keys  are  columns present in the table which are not selected as primary keys but still, they have all the capabilities to be used as a primary key is called alternate key.



**General Concepts**

[Structural Query Language (SQL)](https://www.geeksforgeeks.org/structured-query-language/" \t "https://www.geeksforgeeks.org/advantages-and-disadvantages-of-sql/_blank) is used for accessing, manipulating, and communicating with the database. Almost every function such as retrieving data from the database, creating a new database, manipulating data and databases such as insertion, deletion and updation can be performed using SQL. It is a user-friendly and domain-specific language.

**Need of SQL**

* It is widely used in the Business Intelligence tool.
* Data Manipulation and data testing are done through SQL.
* Data Science tools depend highly on SQL. Big data tools such as Spark, Impala are dependant on SQL.
* It is one of the demanding industrial skills.

**Advantages of SQL**

SQL has many advantages which makes it popular and highly demanded. It is a reliable and efficient language used for communicating with the database. Some advantages of SQL are as follows:

1. Faster Query Processing –  
   Large amount of data is retrieved quickly and efficiently. Operations like Insertion, deletion, manipulation of data is also done in almost no time.
2. No Coding Skills –  
   For data retrieval, large number of lines of code is not required. All basic keywords such as SELECT, INSERT INTO, UPDATE, etc are used and also the syntactical rules are not complex in SQL, which makes it a user-friendly language.
3. Standardised Language –  
   Due to documentation and long establishment over years, it provides a uniform platform worldwide to all its users.
4. Portable –  
   It can be used in programs in PCs, server, laptops independent of any platform (Operating System, etc). Also, it can be embedded with other applications as per need/requirement/use.
5. Interactive Language –  
   Easy to learn and understand, answers to complex queries can be received in seconds.

**What is Data Definition Language?**

A data definition language (DDL) is a computer language used to create and modify the structure of database objects in a database. These database objects include views, schemas, tables, indexes, etc.  
  
This term is also known as data description language in some contexts, as it describes the fields and records in a database table.

Example: Create, Alter etc

**What is Data Manipulation Language?**

A data manipulation language (DML) is a family of computer languages including commands permitting users to manipulate data in a database. This manipulation involves inserting data into database tables, retrieving existing data, deleting data from existing tables and modifying existing data. DML is mostly incorporated in SQL databases.

DML resembles simple English language and enhances efficient user interaction with the system. The functional capability of DML is organized in manipulation commands like SELECT, UPDATE, INSERT INTO and DELETE FROM, as described below:

* SELECT: This command is used to retrieve rows from a table. The syntax is SELECT [column name(s)] from [table name] where [conditions]. SELECT is the most widely used DML command in SQL.
* UPDATE: This command modifies data of one or more records. An update command syntax is UPDATE [table name] SET [column name = value] where [condition].
* INSERT: This command adds one or more records to a database table. The insert command syntax is INSERT INTO [table name] [column(s)] VALUES [value(s)].
* DELETE: This command removes one or more records from a table according to specified conditions. Delete command syntax is DELETE FROM [table name] where [condition].