

## 24.2 Relational Databases

A **relational database** is a logical representation of data that allows the data to be accessed without consideration of its physical structure. A relational database stores data in **tables**.

Figure 24.1 illustrates a sample table that might be used in a personnel system. The table name is **Employee**, and its primary purpose is to store the attributes of employees. Tables are composed of **rows**, each describing a single entity—in Fig. 24.1, an employee. Rows are composed of **columns** in which values are stored. This table consists of six rows. The **Number** column of each row is the table’s **primary key**—a column (or group of columns) with a value that is *unique* for each row. This guarantees that each row can be identified by its primary key. Good examples of primary-key columns are a social security number, an employee ID number and a part number in an inventory system, as values in each of these columns are guaranteed to be unique. The rows in Fig. 24.1 are displayed in order by primary key. In this case, the rows are listed in ascending order by primary key, but they could be listed in descending order or in no particular order at all.

Each column represents a different data attribute. Rows are unique (by primary key) within a table, but particular column values may be duplicated between rows. For example, three different rows in the **Employee** table’s **Department** column contain number 413.

Number	Name	Department	Salary	Location
23603	Jones	413	1100	New Jersey
24568	Kerwin	413	2000	New Jersey
34589	Larson	642	1800	Los Angeles
35761	Myers	611	1400	Orlando
47132	Neumann	413	9000	New Jersey
78321	Stephens	611	8500	Orlando

Row      Primary key      Column

## Fig. 24.1

Employee table sample data.

Description

## Selecting Data Subsets

Different users of a database are often interested in different data and different relationships among the data. Most users require only subsets of the rows and columns. Queries specify which subsets of the data to select from a table. You use SQL to define queries. For example, you might select data from the Employee table to create a result that shows where each department is located, presenting the data sorted in increasing order by department number. This result is shown in Fig. 24.2. SQL is discussed in Section 24.4.

Department	Location
413	New Jersey

611	Orlando
642	Los Angeles

## Fig. 24.2

Distinct Department and Location data from the Employees table.