Unit-2
Relational Database

A relational database is a collection of information that organizes data points with defined relationships for easy access. In the relational database model, the data structures -- including data tables, indexes and views -- remain separate from the physical storage, allowing administrators to edit the physical data storage without affecting the logical data structure.

The data tables used in a relational database store information about the related objects. Each row holds a record with a unique identifier -- known as a key -- and each column contains the attributes of the data. Each record assigns a value to each feature, making relationships between data points easy to identify.

The standard user and application program interface (API) of a relational database is the Structured Query Language (SQL). SQL statements are used both for interactive queries for information from a relational database and for gathering data for reports. Defined data integrity rules must be followed to ensure the relational database is accurate and accessible.

Relational databases are beneficial to all types and sizes of organizations where related data points must be consistently managed and secured. For example, e-commerce businesses can use the relational model to process purchases and track inventory. The relational database is the most widely accepted database model.

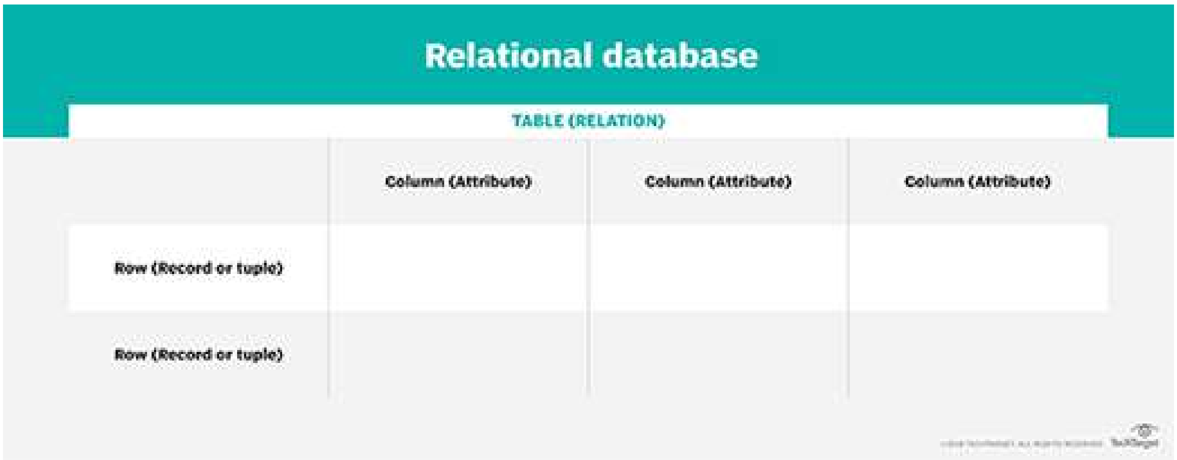
What is in a relational database model?

The relational database was invented in 1970 by E. F. Codd, then a young programmer at IBM. In his paper, "A Relational Model of Data for Large Shared Data Banks," Codd proposed shifting from storing data in hierarchical or navigational structures to organizing data in tables containing rows and columns.

Each table -- sometimes called a *relation* -- in a relational database contains one or more data categories in columns -- or *attributes*. Each row -- also called a *record* or *tuple* -- contains a unique instance of data -- or *key* -- for the categories defined by the columns. Each table has a unique primary key, which identifies the information in a table. The relationship between tables can then be set via the use of foreign keys -- a field in a table that links to the primary key of another table.

For example, a typical business order entry database would include a table that described a customer with columns for name, address, phone number and so forth. Another table would describe an order -- including information like the product, customer, date and sales price.

A user of a relational database can then obtain a view of the database to fit their needs. For example, a branch office manager might like a view or report on all customers that bought products after a certain date. A financial services manager in the same company could, from the same tables, obtain a report on accounts that need to be paid.



Relational database

TABLE (RELATION)

	Column (Attribute)	Column (Attribute)	Column (Attribute)
Row (Record or tuple)			
Row (Record or tuple)			

A relational database includes tables containing rows and columns.

When creating a relational database, you can define the domain of possible values in a data column and further constraints that may apply to that data value. For example, a domain of possible customers could allow up to 10 possible customer names but is limited in one table to allowing only three of these customer names to be specifiable.

Two constraints relate to data integrity and the primary and foreign keys:

- **Entity integrity** ensures that the primary key in a table is unique and that the value is not set to null.
- **Referential integrity** requires that every value in a foreign key column will be found in the primary key of the table from which it originated.

In addition, relational databases possess physical data independence. This refers to a system's capacity to make changes to the inner schema without altering the external schemas or application programs. Inner schema alterations may include:

- the use of new storage devices;
- modifying indexes;
- changing from a specific access method to a different one;
- utilizing different data structures; and
- using various storage structures or file organizations.

Logical data independence is a system's ability to manage the conceptual schema without altering the external schema or application programs. Conceptual schema alterations may include the addition or deletion of new relationships, entities or attributes without altering existing external schemas or rewriting application programs.

Examples of relational databases

Standard relational databases enable users to manage predefined data relationships across multiple databases. Popular examples of standard relational databases include Microsoft SQL Server, Oracle Database, MySQL and IBM DB2.

Cloud-based relational databases, or database as a service (DBaaS), are also widely used because they enable companies to outsource database maintenance, patching and infrastructure support requirements. Cloud relational databases include Amazon Relational Database Service (RDS), Google Cloud SQL, IBM DB2 on Cloud, SQL Azure and Oracle Cloud.

Types of databases

There are a number of database categories, from basic flat files that aren't relational to NoSQL and newer graph databases that are considered even more relational than standard relational databases.

A flat file database consists of a single table of data that has no interrelation -- typically text files. This type of file enables users to specify data attributes, such as columns and data types.

A NoSQL database is an alternative to relational databases that's especially useful for working with large sets of distributed data. These databases can support a variety of data models, including key-value, document, columnar and graph formats.

A graph database expands beyond traditional column- and row-based relational data models; this NoSQL database uses nodes and edges that represent connections between data relationships and can discover new relationships between the data. Graph databases are more sophisticated than relational databases, and thus, their uses include fraud detection or web recommendation engines.

An object relational database (ORD) is composed of both a relational database management system (RDBMS) and an object-oriented database management system (OODBMS). An ORD contains characteristics of both the RDBMS and OODBMS models. In an ORD, a traditional database is used to store the data. It is then accessed and manipulated using queries written in a query language, such as SQL. Therefore, the basic approach of an ORD is based on a relational database.

However, an ORD can also be considered object storage, particularly for software written in the object-oriented programming language (OOP), thus pulling on object-oriented characteristics. In this situation, APIs are utilized in the storage and retrieval of data.

Advantages of relational databases

The main advantage of relational databases is that they enable users to easily categorize and store data that can later be queried and filtered to extract specific information for reports. Relational databases are also easy to extend and aren't reliant on physical organization. After the original database creation, a new data category can be added without all existing applications being modified.

Other relational database advantages include:

- **Accuracy:** Data is stored just once, eliminating data deduplication.
- **Flexibility:** Complex queries are easy for users to carry out.
- **Collaboration:** Multiple users can access the same database.
- **Trust:** Relational database models are mature and well-understood.
- **Security:** Data in tables within a RDBMS can be limited to allow access by only specific users.

Differences between a database and a relational database

The majority of software products in today's market incorporate both relational database and regular database compliances. Therefore, they can manage databases in the relational tabular form as well as in file form, or both. Essentially, in today's market, a relational database is a database and vice versa; however, there are still major differences in data storage between the two systems.

The most important difference is that a relational database stores data in a tabular form -- or arranged in a table with rows and columns -- while a database stores data as files. Other differences include:

- Database normalization is present in a relational database while it is not present in a database.
- A relational database supports a distributed database while a database does not support a distributed database.
- In a relational database, the data values are stored in the form of tables and each table possesses a primary key. In a database, data is normally stored in hierarchical or navigational form.
- Since data is stored in the form of tables in a relational database, then the relationship between these data values is stored as well. Since a database stores data as files, then there is no relationship between the values or tables.
- In a relational database, the integrity constraints are defined for the purpose of ACID. On the other hand, a database does not utilize any security to protect against data manipulation.
- While a relational database is designed to support large amounts of data and multiple users, a database is designed to deal with small amounts of data and one single user.

One final, major distinction is that the data storage in a relational database is accessible, meaning the value can be updated by the system. Furthermore, the data within an RDBMS is physically and logically independent.