**What is a relational database? A deep dive -** [The Educative Team](https://educative-inc.medium.com/?source=post_page-----3247d26869a6--------------------------------) Aug 10, 2022

Relational databases are a useful way to work with structured data. They represent data in an organized and clear way, making it easy to see and understand relationships between data structures. Today, we’ll dive deeper into relational databases and discuss their advantages, uses, and more.

**We’ll cover**:

* What is a relational database?
* Advantages of relational databases
* Relational database uses
* What is SQL?
* Properties of relational tables
* Relational database example
* Wrapping up and next steps

**What is a relational database?**

A relational database, or SQL database, **stores data in tables and allows easy access to related data points**. In a relational database, each row in the database table is a record with a unique identifier called a *key*, and each column holds attributes of the data.

The tables in a relational database are **based on the relational model**, which is a simple, intuitive way to represent data. Let’s take a look at some of the fundamental concepts of the relational data model in relation to a sample table. The relation is *Student* and the attributes are *First\_Name*, *Student\_ID*, and *Student\_Age*.

**Student**

Table

Description automatically generated

* **Attributes**: The columns in the table. Attributes define a relationship, like *First\_Name* or *Student\_ID*.
* **Relation schema**: Represents the name of the relation with its attributes. For example, *Student (First\_Name, Student\_ID, Student\_Age)* is a relation schema for *Student*.
* **Tuples**: A single row of a table containing a single record. The table above has three tuples, and one of them is *Elliot 17 20*.
* **Degree**: The number of attributes in the relation. The above relation has three.
* **Carindality**: The number of tuples in the relation. The above relation has three.

**Relational Database Management System**

A relational database management system (RDBMS) is the **software that is used to record, manipulate, and retrieve data in a relational database**. This software is a great way to work with related data points that need to be maintained in a safe and consistent way.

MySQL is the most popular open-source relational database management system (RDBMS). Some other popular relational database software includes:

* MongoDB
* PostgreSQL
* Microsoft SQL Server
* Oracle Database
* Amazon Relational Database Service (RDS)
* IBM Informix

**Non-relational databases**

Non-relational databases, or NoSQL databases, **store data in a non-tabular way**. They use a variety of different data models to access and manage types of data. NoSQL databases are **great for applications with large data sets** as they are responsive, scalable, and adaptive.

NoSQL databases provide highly functional APIs and data types built for their corresponding data models and optimized for higher performance.

There are **four main types of NoSQL databases**:

* Key-value stores
* Graph databases
* Column-oriented
* Document store

**Note**: NoSQL stands for *not only SQL* because you can either use them with or without SQL.

**Advantages of relational databases**

There are **many advantages** to using the relational database model for data management and data storage, including:

* **Flexibility**: It’s easy to expand, update, and delete data whenever needed.
* **Atomicity**: Atomic transactions ensure that you either commit to the entire transaction or have no transaction at all. If there’s a lost connection, the relational database goes back to its previous state.
* **Durability**: The changes that are made to the database will survive permanently, even in the event of a system crash.
* **Consistency**: The only data allowed to be written into the database is data that follows data validation rules.

**Relational database uses**

Relational databases are useful for many different things, such as:

* Application development
* Data warehousing
* Data maintenance in applications
* Storing structured data
* Logging applications

They’re also **useful in many different industries**, such as:

* Retail and e-commerce
* Finance
* Banking
* Insurance
* IT
* Telecom
* Health
* Manufacturing

**What is SQL?**

[SQL](https://www.educative.io/blog/what-are-database-schemas-examples?eid=5082902844932096), or Structured Query Language, is the programming **language used to work with data in a relational database**. All major relational database management systems use SQL as their db language.

SQL is the **standard choice because of its benefits**. With SQL, users can:

* Access data in their RDMS
* Describe, define, and manipulate data
* Use SQL modules, libraries, and pre-compilers to embed it into other programming languages
* Make and delete databases and tables
* Set permissions on tables, processes, and views

You can **use SQL statements to interact with your relational database**. These commands are organized into three different groups:

* **Data Definition Language (DDL):**
* CREATE
* ALTER
* DROP
* **Data Manipulation Language (DML):**
* SELECT
* INSERT
* UPDATE
* DELETE
* **Data Control Language (DCL):**
* GRANT
* REVOKE

Let’s take a look at how to use one of these commands. Here’s how to use the CREATE DATABASE command:

CREATE DATABASE DatabaseName;

If you want to create two databases and then display the list of your databases, you can do this:

CREATE DATABASE sampleDB1;  
CREATE DATABASE sampleDB2;  
SHOW DATABASES;--->   
Database  
information\_schema  
mysql  
performance\_schema  
ri\_db  
sampleDB1  
sampleDB2  
sys

**Note**: The SHOW DATABASE command is used to display your databases.

**Properties of relational tables**

Let’s explore some of the characteristics of relational tables:

**Each row is unique**

No two rows in a table are the same.

**Values are atomic**

Atomic values can’t be broken down into smaller pieces. Relational tables don’t contain repeating groups or multi-valued attributes. This simplifies data manipulation.

**Column values are of the same type**

All values in a column come from the same domain based on their data type. This allows for simplified data access because you can be sure of the type of data contained within a given column.

**The sequence of columns doesn’t matter**

Columns can be retrieved in any order. This allows users to share the same table without worrying about its organization.

**The sequence of rows doesn’t matter**

The rows of a relational table can be retrieved in any order. Adding new data to a table doesn’t affect existing queries.

**Each column has a different name**

Since the sequence of columns doesn’t matter, so they must be referred to by name. A column name doesn’t need to be unique within an entire database, but only within the relation it belongs in.

**Relational database example**

Relational databases work in many different scenarios. Let’s say there’s a college that wants to maintain information about its students, departments, and instructors. We can do that easily with a relational database. The table below shows the relational database structure and some sample data records.

**Student table**:

Table

Description automatically generated

**Department table**:

Table

Description automatically generated

**Instructor table**:

Graphical user interface, text, application

Description automatically generated

This database has three tables that store data records of the same type in an organized way. Once all the data is defined and constructed into the database, you can begin working with a database management system (DBMS) to retrieve information from the different tables.

**Wrapping up and next steps**

Congratulations on taking your first steps with relational databases! They are a great tool for working with structured data. There are still many things to learn about databases, such as:

* Primary keys and foreign keys
* Normalization
* Boyce-Codd Normal Form

To learn these concepts and more, check out Educative’s course [**Database Design Fundamentals for Software Engineers**](https://www.educative.io/courses/database-design-fundamentals?eid=5082902844932096). In this course, you’ll learn about the fundamentals of databases and SQL. After completing the course, you’ll be ready to move on to more advanced concepts involving database systems and SQL.

*Happy learning!*

**Continue reading about databases and SQL on Educative**

* [The complete guide to system design in 2022](https://www.educative.io/blog/complete-guide-to-system-design?eid=5082902844932096)
* [What is a Database Foreign Key? A beginner’s tutorial](https://www.educative.io/blog/what-is-foreign-key-database?eid=5082902844932096)
* [What is a Database Query? SQL and NoSQL queries explained](https://www.educative.io/blog/what-is-database-query-sql-nosql?eid=5082902844932096)
* [MySQL Workbench tutorial: A complete guide to the RDBMS tool](https://www.educative.io/blog/mysql-workbench-tutorial?eid=5082902844932096)

**Start a discussion**

Where else would a relational database be useful? Was this article helpful? Let us know in the comments below!