**What is a database**

A database is a system of software and capabilites that make validating, storing, searching, filtering, aggregating, grouping, and administering data possible. In enterprise applications databases fall into 2 main categories SQL and NoSQL.

**SQL databases**

SQL databases are a type of RDBMS which use the standard Structured Query Language to administer the data. Data in a SQL database are stored in objects called tables. Tables provide the relational information for the data stored in the database.

**NoSQL databases**

NoSQL (Not Only SQL) databases are not necessarily based on the relational model, unlike RDBMS. NoSQL databases typically use some other means or DSL (domain-specific language) for administering data and use different structures for storing data and relational information.

SQL

Structured Query Language or SQL is the standard language for working with RDBM systems. SQL is used to administer and manipulate SQL servers. SQL is a scripting language that is interpreted by the database server. SQL is used to...

* Define database structure
* Manipulate stored data
* Define data access permissions
* Control concurrent data access
* Query stored data

RDBMS

Relational Database Management System or RDBMS is a set of software and capabilities that enable IT teams to create, update, administer and interact with a relational database.

The use of RDBMS is essential in large scale application development. The use is so widespread that it would be nearly impossible to find an enterprise application that doesn't utilize an RDBMS for data persistence. Below is a list of some reasons for using an RDBMS.

* Structured data model
* Large scale concurrent data access
* Fault tolerance
* Distributed data storage
* Enforced data integrity
* Support for multiple client types

Schema

The term schema comes from Greek and is defined as 'form, figure'. In the world of databases, schema gives shape to the data stored in the database. In a DBMS, schema is the structure of a database which is described in a formal language which is supported by the DBMS.

SQL Data types

annual\_income decimal(10,2);

age **int**;

dob **date**;

received\_at **time**;

created\_at **timestamp**;

Multiplicity and cardinality

**Cardinality** and **multiplicity** are closely related terms used in data modeling (like ER diagrams), databases, and object-oriented modeling (like UML).

Cardinality refers to the number of elements in a set or the number of relationships between entities.

**One-to-One (1:1)**: Each student has one ID card.

**One-to-Many (1:N)**: One teacher can teach many students.

**Many-to-Many (M:N)**: Students can enroll in many courses, and each course can have many students.

Normalization

Normalization is the process of organizing data within a database ([**relational database**](https://www.datacamp.com/tutorial/sql-database)) to eliminate data anomalies, such as redundancy.

In simpler terms, it involves breaking down a large, complex table into smaller and simpler tables while maintaining data relationships.

Normalization is commonly used when dealing with large datasets.

**Why is Normalization in SQL Important?**

Normalization plays a crucial role in database design. Here are several reasons why it’s essential:

* **Reduces redundancy:** [**Redundancy**](https://campus.datacamp.com/courses/creating-postgresql-databases/database-normalization?ex=1) is when the same information is stored multiple times, and a good way of avoiding this is by splitting data into smaller tables.
* **Improves query performance:**You can perform faster query execution on smaller tables that have undergone normalization.
* **Minimizes update anomalies:**With normalized tables, you can easily update data without affecting other records.
* **Enhances data integrity:** It ensures that data remains consistent and accurate.

A database anomaly is a flaw in the database that occurs because of poor planning and redundancy.

Insertion anomalies, Updation anomalies, Deletion anomalies

Types of Normal Forms:

First Normal Form (1NF)

--A relation is in 1NF if every attribute is a single -valued attribute or it does not contain any multi-values or composite attribute, ie every attribute is an atomic attribute.

|  |  |  |
| --- | --- | --- |
| Employee code | Employee Name | Employee phone number |
| 101 | John | 987979,980800 |
| 102 | Sam | 980800980 |
| 103 | Arun | 989808080 |

|  |  |  |
| --- | --- | --- |
| Employee code | Employee Name | Employee phone number |
| 101 | John | 987979 |
| 101 | John | 980800 |
| 102 | Sam | 980800980 |
| 103 | Arun | 989808080 |

Second Normal Form

For a relational table to be in second normal form,it must satisfy the following rules:

1. The table must be in first normal form
2. It must not contain any partial dependency, ie, all non-prime attributes are fully functionally dependent on the primary key.

|  |  |  |  |
| --- | --- | --- | --- |
| Employee Code | Project ID | Employee Name | Project Name |
| 101 | P03 | John | Project123 |
| 101 | P01 | John | Project101 |
| 102 | P02 | Ryan | Project102 |
|  |  |  |  |
|  |  |  |  |

|  |  |
| --- | --- |
| Employee code | Employee name |
| 101 | John |
| 101 | John |

|  |  |
| --- | --- |
| Project code | Project Name |
| P03 | Project103 |
| Po4 | Project104 |

A screenshot of a computer

AI-generated content may be incorrect.

Third Normal Form

For a relational table to be in third normal form,it must satisfy the following rules:

1. The table must be in the second normal form
2. No non-prime attribute is transitively dependent on the primary key.
3. For each functional dependency X->Y at least one of the following conditions hold:
   1. X is a super key of the table
   2. X is a prime attribute of the table.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Boyce-Codd Normal Form

For a relational table to be in Boyce-Codd normal form,it must satisfy the following rules

1.The table must be in the third normal form.

2. For every non-trivial functional dependency X->Y,

X is the superkey of the table. That means X cannot be a non-prime attribute if Y is a prime attribute.



A screenshot of a computer

AI-generated content may be incorrect.