

Relational database management system RDBMS in SQL

What is RDBMS? RDBMS stands for Relational Database Management System. RDBMS is the foundation for SQL as well as for all other database systems such as MS SQL Server, IBM DB2, Oracle, MySQL and Microsoft Access.

RDBMS is an indispensable part when learning about SQL. So what is RDBMS? What ingredients does it include? We will find out together in this lesson.

What is RDBMS?

RDBMS stands for **R**elational **D**atabase Management **S**ystem (**Relational Database Management System**). RDBMS is the foundation for SQL as well as for all other database systems such as MS SQL Server, IBM DB2, Oracle, MySQL and Microsoft Access.

A relational database management system is a database management system (DBMS) based on the relational model introduced by EFCodd.

Data in an RDBMS is stored in database objects called tables. A table is a collection of related data entries, it consists of columns and rows.

What is a table?

If you know SQL, you probably already know the data tables in it. It can be said that, data in RDBMS is stored in database objects called tables. This table is essentially a collection of relational data items, and it consists of multiple columns and rows.

Remember, tables are the simplest and most common form of data storage in relational databases. Here is an example of the **SAN_PHAM** table :

```
-- Vi du ve bang SAN_PHAM +-----+-----+-----+-----+-----+ | ID | TEN
```

What is school?

Every table can be divided into smaller sections called fields. Fields in the SAN_PHAM table include: ID, TEN, SL, GIA and THANH_TIEN. A field is a column in a table, designed to maintain specific information about every record in the table.

What is a record or row?

A record is often called a row of data, which is each individual item that exists in the table. For example, there are 5 records in the SAN_PHAM table above. And here is a row of data or record in the SAN_PHAM table:

```
-- Hang trong bang SAN_PHAM +-----+-----+-----+-----+-----+ | 1 |
```

A record is a horizontal row in the table.

What is a column?

A column is a vertical entity in a table, containing all the information associated with a field in the table. For example, a column in the SAN_PHAM table is TEN, which describes the product name and is displayed as shown below:

```
-- Cot trong bang SAN_PHAM +-----+ | TEN | +-----+ | Tui01 | | Tui02 |
```

What is NULL value?

The NULL value in the table is the value in the blank field, or in other words, the field does not contain a value, it was left blank when creating the record.

You need to remember that NULL is a blank field, completely different from the value 0 or a field containing white space. A field with a NULL value is a field that has been left blank since the record creation process.

Constraints in SQL

Constraints are rules enforced on the data columns of a table. They are used to limit the type of data that can be entered into a table. These constraints help ensure the accuracy and reliability of the data in the database.

There are two main types of constraints: column-level constraints (applied only in columns) and table-level constraints (applied throughout the data table).

Here is a list of the most commonly used constraints in SQL:

NOT NULL : Ensures that the column does not contain NULL values.

DEFAULT : Provides a default value for a column when no value is specified for the column.

UNIQUE : Ensures that all values in a column are different.

PRIMARY Key : We often call the primary key, this constraint uniquely identifies a record in a database table.

FOREIGN Key : Also known as foreign key, this constraint uniquely identifies a record in any other database table.

You can simply understand that a foreign key is one or a group of columns in a database table, acting as a bridge between data in two different tables. It acts as a cross-reference between tables because it references the primary

key of another table, thereby establishing a link between the tables. The majority of tables in a database system follow the concept of foreign keys. In complex databases and data warehouses, data in one domain must be added to multiple tables, thereby maintaining relationships between them. The concept of referential integrity also originates from this foreign key theory. Implementing foreign keys is often more complicated than primary keys. I find this the most difficult to understand so I'll explain further. If you still don't understand, don't rush, in the next articles there will be detailed explanations :).

CHECK : The CHECK constraint ensures that all values in a column meet certain conditions.

INDEX : Often used to quickly create and extract data from a database.

Data integrity in SQL

Data integrity is the completeness, accuracy and consistency of the data. It ensures that new and modified values must ensure accuracy and reasonableness with the overall data. Integrity is typically established during the database design phase through the use of standard procedures and rules.

Various error checking methods and validation procedures will be implemented to maintain data integrity. If the data intended to be included in the table violates data integrity rules and procedures, it will not be saved in the database.

Here are some types of data integrity that exist in every RDBMS :

Entity Integrity - Ensures that there are no duplicate records in the table.

Domain Integrity - Value domain integrity : Ensures data entered into a specified column must comply with certain restrictions such as data type, format or data range.

Referential integrity - Referential integrity : Refers to the accuracy and consistency of data in a relationship. In a relational database, data is linked between two or more tables. This is achieved through foreign keys. Therefore, referential integrity ensures that the row is not deleted when it is used by another record, meaning the data must remain intact on both sides of the reference.

User-Defined Integrity - Applies some user-defined data integrity rules, as long as the rules are different from the data integrity types above.

Standardize the database

Database normalization is the process of organizing data effectively in a database. Why is this process necessary? There are two reasons:

1. One is to eliminate redundant data, such as identical data stored across multiple tables.
2. Second, ensure meaningful data dependencies, for example, data is stored properly, related data is stored on the same table.

These two reasons are well worth implementing database normalization, as they reduce the amount of storage space used by the database and ensure that data is stored logically. Normalization includes a series of instructions that help you create a good database structure.

Normalization instructions are divided into Normal Forms (formats or how the database structure is laid out) whose purpose is to organize the database structure so that it complies with the rules of First Normal Form (1NF), then Second Normal Form (2NF) and finally Third Normal Form (3NF). We will learn specifically about these Normal Forms in more in-depth articles.

This article is full of theory, the concepts are like a jumble, but you try to grasp the main ideas because it is the basis for implementing the next content in SQL. The next lesson is also a theoretical lesson, talking about Database

You finished reading the article "**Relational database management system RDBMS in SQL**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.