| **DBMS** | **RDBMS** |
| --- | --- |
| [**DBMS**](https://www.geeksforgeeks.org/introduction-of-dbms-database-management-system-set-1/)**stores data as file.** | [**RDBMS**](https://www.geeksforgeeks.org/rdbms-architecture/)**stores data in tabular form.** |
| **Data elements need to access individually.** | **Multiple data elements can be accessed at the same time.** |
| No relationship between data. | Data is stored in the form of tables which are related to each other. |
| Normalization is not present. | Normalization is present. |
| DBMS does not support distributed database. | RDBMS supports distributed database. |
| It stores data in either a navigational or hierarchical form. | It uses a tabular structure where the headers are the column names, and the rows contain corresponding values. |
| It deals with small quantity of data. | It deals with large amount of data. |
| Data redundancy is common in this model. | Keys and indexes do not allow Data redundancy. |
| It is used for small organization and deal with small data. | It is used to handle large amount of data. |
| Not all Codd rules are satisfied. | All 12 Codd rules are satisfied. |
| Security is less | More security measures provided. |
| It supports single user. | It supports multiple users. |
| Data fetching is slower for the large amount of data. | Data fetching is fast because of relational approach. |
| The data in a DBMS is subject to low security levels with regards to data manipulation. | There exists multiple levels of data security in a RDBMS. |
| Low software and hardware necessities. | Higher software and hardware necessities. |
| Examples:[XML](https://www.geeksforgeeks.org/xml-basics/), Window Registry, Forxpro, dbaseIIIplus etc. | Examples: [MySQL](https://www.geeksforgeeks.org/architecture-of-mysql/), [PostgreSQL](https://www.geeksforgeeks.org/what-is-postgresql-introduction/), [SQL](https://www.geeksforgeeks.org/what-is-sql/) Server, Oracle, Microsoft Access etc. |

1. **Data Storage**:
   * **DBMS**: Stores data as files.
   * [**RDBMS**: Stores data in tabular form, with rows and columns](https://www.geeksforgeeks.org/difference-between-rdbms-and-dbms/)
2. **Data Relationships**:
   * **DBMS**: Does not support relationships between data.
   * [**RDBMS**: Supports relationships between tables using foreign keys1](https://www.geeksforgeeks.org/difference-between-rdbms-and-dbms/)[2](https://www.coursera.org/articles/difference-between-dbms-and-rdbms).
3. **Normalization**:
   * **DBMS**: Normalization is not typically supported.
   * [**RDBMS**: Supports normalization to reduce data redundancy1](https://www.geeksforgeeks.org/difference-between-rdbms-and-dbms/)[2](https://www.coursera.org/articles/difference-between-dbms-and-rdbms).
4. **Data Redundancy**:
   * **DBMS**: Higher data redundancy.
   * [**RDBMS**: Reduced data redundancy due to relational structure1](https://www.geeksforgeeks.org/difference-between-rdbms-and-dbms/)[2](https://www.coursera.org/articles/difference-between-dbms-and-rdbms).
5. **Users**:
   * **DBMS**: Generally supports single users.
   * [**RDBMS**: Supports multiple users](https://www.geeksforgeeks.org/difference-between-rdbms-and-dbms/)[1](https://www.geeksforgeeks.org/difference-between-rdbms-and-dbms/)[2](https://www.coursera.org/articles/difference-between-dbms-and-rdbms).
6. **Security**:
   * **DBMS**: Lower security measures.
   * [**RDBMS**: Higher security measures, including access controls](https://www.geeksforgeeks.org/difference-between-rdbms-and-dbms/)[1](https://www.geeksforgeeks.org/difference-between-rdbms-and-dbms/)[2](https://www.coursera.org/articles/difference-between-dbms-and-rdbms).
7. **Examples**:
   * **DBMS**: Examples include XML, Windows Registry, and dBase.
   * [**RDBMS**: Examples include MySQL, PostgreSQL, SQL Server, and Oracle](https://www.geeksforgeeks.org/difference-between-rdbms-and-dbms/)

**Codd’s 12 Rules** are a set of thirteen rules (numbered from zero to twelve) proposed by Edgar F. Codd, a pioneer of the relational model for databases. [These rules define what is required for a database management system to be considered relational, i.e., a Relational Database Management System (RDBMS)1](https://en.wikipedia.org/wiki/Codd%27s_12_rules)[2](https://www.geeksforgeeks.org/codds-rules-in-dbms/).

Here is a brief overview of Codd’s 12 Rules:

Here is a brief overview of Codd’s 12 Rules:

1. **Rule 0: The Foundation Rule**: For any system to qualify as an RDBMS, it must be able to manage databases entirely through its relational capabilities.
2. **Rule 1: The Information Rule**: All information in a relational database is represented explicitly at the logical level and in exactly one way – by values in tables.
3. **Rule 2: The Guaranteed Access Rule**: Each and every datum (atomic value) in a relational database is guaranteed to be logically accessible by resorting to a combination of table name, primary key value, and column name.
4. **Rule 3: Systematic Treatment of Null Values**: Null values must be uniformly treated as “missing information,” not as empty strings, blanks, or zeros.
5. **Rule 4: Dynamic Online Catalog Based on the Relational Model**: The database description is represented at the logical level in the same way as ordinary data, so that authorized users can apply the same relational language to its interrogation as they apply to the regular data.
6. **Rule 5: The Comprehensive Data Sublanguage Rule**: A relational system may support several languages and various modes of terminal use, but there must be at least one language that supports data definition, view definition, data manipulation, integrity constraints, authorization, and transaction boundaries.
7. **Rule 6: The View Updating Rule**: All views that are theoretically updatable must also be updatable by the system.
8. **Rule 7: High-level Insert, Update, and Delete**: The system must support set-level insert, update, and delete operations.
9. **Rule 8: Physical Data Independence**: Changes to the physical storage of data should not affect the application programs.
10. **Rule 9: Logical Data Independence**: Changes to the logical structure of the database should not affect the user’s ability to access data.
11. **Rule 10: Integrity Independence**: Integrity constraints must be specified separately from application programs and stored in the catalog.
12. **Rule 11: Distribution Independence**: The distribution of data across multiple locations should be invisible to users.

1. **[Rule 12: Non-Subversion Rule](https://en.wikipedia.org/wiki/Codd%27s_12_rules" \t "_blank)**[: If a system provides a low-level (record-at-a-time) interface, it must not be able to bypass the integrity rules and constraints expressed in the higher-level relational language](https://en.wikipedia.org/wiki/Codd%27s_12_rules" \t "_blank)[1](https://en.wikipedia.org/wiki/Codd%27s_12_rules)[2](https://www.geeksforgeeks.org/codds-rules-in-dbms/).

[These rules ensure that a database system maintains data integrity, consistency, and usability, making it a true RDBMS](https://en.wikipedia.org/wiki/Codd%27s_12_rules" \t "_blank)

[**Data redundancy** occurs when the same piece of data is stored in multiple places within a database or across different databases1](https://www.talend.com/resources/what-is-data-redundancy/)[2](https://coresignal.com/blog/data-redundancy/). This can happen intentionally or unintentionally and has both advantages and disadvantages.

**Key Points about Data Redundancy:**

1. **Types**:
   * [**Intentional Redundancy**: Used for data backup and recovery purposes, ensuring data availability in case of system failures1](https://www.talend.com/resources/what-is-data-redundancy/).
   * [**Unintentional Redundancy**: Often results from poor database design, leading to duplicate data entries2](https://coresignal.com/blog/data-redundancy/).
2. **Advantages**:
   * **Data Backup**: Provides an extra layer of data protection.
   * [**Data Security**: Enhances data security by having multiple copies1](https://www.talend.com/resources/what-is-data-redundancy/).
3. **Disadvantages**:
   * [**Increased Storage Costs**: Storing duplicate data consumes more storage space2](https://coresignal.com/blog/data-redundancy/).
   * [**Data Inconsistency**: Can lead to inconsistencies if one copy of the data is updated while others are not](https://www.talend.com/resources/what-is-data-redundancy/)[2](https://coresignal.com/blog/data-redundancy/).
   * [**Complexity**: Makes database management more complex and can slow down query performance](https://www.talend.com/resources/what-is-data-redundancy/)[2](https://coresignal.com/blog/data-redundancy/).

**Example:**

Imagine a customer database where the same customer’s information is stored in multiple tables. If the customer’s address changes, it must be updated in all tables to maintain consistency. Failure to do so can lead to data anomalies and inconsistencies.

[**Data redundancy** refers to the situation where **the same pieces of data are stored in multiple places within a database or data storage system**](https://www.bing.com/ck/a?!&&p=7e91c80c6b38c55bJmltdHM9MTcyMjgxNjAwMCZpZ3VpZD0zMDE0ZjgzNy0xYTZlLTY5MGUtMjgzNy1lY2ExMWJkYzY4MDUmaW5zaWQ9NTc3Ng&ptn=3&ver=2&hsh=3&fclid=3014f837-1a6e-690e-2837-eca11bdc6805&psq=5.+What+do+you+understand+By+Data+Redundancy%3f&u=a1aHR0cHM6Ly93d3cubWluaXRvb2wuY29tL2xpYi9kYXRhLXJlZHVuZGFuY3kuaHRtbA&ntb=1)[**1**](https://www.bing.com/ck/a?!&&p=d36a4a0f7afe0589JmltdHM9MTcyMjgxNjAwMCZpZ3VpZD0zMDE0ZjgzNy0xYTZlLTY5MGUtMjgzNy1lY2ExMWJkYzY4MDUmaW5zaWQ9NTc3Nw&ptn=3&ver=2&hsh=3&fclid=3014f837-1a6e-690e-2837-eca11bdc6805&psq=5.+What+do+you+understand+By+Data+Redundancy%3f&u=a1aHR0cHM6Ly93d3cubWluaXRvb2wuY29tL2xpYi9kYXRhLXJlZHVuZGFuY3kuaHRtbA&ntb=1)[**2**](https://www.bing.com/ck/a?!&&p=4dcd38ed8b3848ccJmltdHM9MTcyMjgxNjAwMCZpZ3VpZD0zMDE0ZjgzNy0xYTZlLTY5MGUtMjgzNy1lY2ExMWJkYzY4MDUmaW5zaWQ9NTc3OA&ptn=3&ver=2&hsh=3&fclid=3014f837-1a6e-690e-2837-eca11bdc6805&psq=5.+What+do+you+understand+By+Data+Redundancy%3f&u=a1aHR0cHM6Ly93d3cuaW5kZWVkLmNvbS9jYXJlZXItYWR2aWNlL2NhcmVlci1kZXZlbG9wbWVudC93aGF0LWlzLWRhdGEtcmVkdW5kYW5jeQ&ntb=1)[**3**](https://www.bing.com/ck/a?!&&p=ca822b4aabddc8e7JmltdHM9MTcyMjgxNjAwMCZpZ3VpZD0zMDE0ZjgzNy0xYTZlLTY5MGUtMjgzNy1lY2ExMWJkYzY4MDUmaW5zaWQ9NTc3OQ&ptn=3&ver=2&hsh=3&fclid=3014f837-1a6e-690e-2837-eca11bdc6805&psq=5.+What+do+you+understand+By+Data+Redundancy%3f&u=a1aHR0cHM6Ly9kYXRhY29ub215LmNvbS8yMDIyLzAzLzA5L3doYXQtaXMtZGF0YS1yZWR1bmRhbmN5Lw&ntb=1). [This can happen intentionally or accidentally. Redundancy can be useful for data recovery in case of corruption or loss](https://www.bing.com/ck/a?!&&p=cd7327777a1de4c9JmltdHM9MTcyMjgxNjAwMCZpZ3VpZD0zMDE0ZjgzNy0xYTZlLTY5MGUtMjgzNy1lY2ExMWJkYzY4MDUmaW5zaWQ9NTc4MA&ptn=3&ver=2&hsh=3&fclid=3014f837-1a6e-690e-2837-eca11bdc6805&psq=5.+What+do+you+understand+By+Data+Redundancy%3f&u=a1aHR0cHM6Ly9kYXRhY29ub215LmNvbS8yMDIyLzAzLzA5L3doYXQtaXMtZGF0YS1yZWR1bmRhbmN5Lw&ntb=1)[**3**](https://www.bing.com/ck/a?!&&p=6e7c81f79516b667JmltdHM9MTcyMjgxNjAwMCZpZ3VpZD0zMDE0ZjgzNy0xYTZlLTY5MGUtMjgzNy1lY2ExMWJkYzY4MDUmaW5zaWQ9NTc4MQ&ptn=3&ver=2&hsh=3&fclid=3014f837-1a6e-690e-2837-eca11bdc6805&psq=5.+What+do+you+understand+By+Data+Redundancy%3f&u=a1aHR0cHM6Ly9kYXRhY29ub215LmNvbS8yMDIyLzAzLzA5L3doYXQtaXMtZGF0YS1yZWR1bmRhbmN5Lw&ntb=1). [In computer memory and storage, data redundancy allows for error correction](https://www.bing.com/ck/a?!&&p=faf71a64b3f9c537JmltdHM9MTcyMjgxNjAwMCZpZ3VpZD0zMDE0ZjgzNy0xYTZlLTY5MGUtMjgzNy1lY2ExMWJkYzY4MDUmaW5zaWQ9NTc4Mg&ptn=3&ver=2&hsh=3&fclid=3014f837-1a6e-690e-2837-eca11bdc6805&psq=5.+What+do+you+understand+By+Data+Redundancy%3f&u=a1aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvRGF0YV9yZWR1bmRhbmN5&ntb=1)[**4**](https://www.bing.com/ck/a?!&&p=bf6bd99f1786dbdfJmltdHM9MTcyMjgxNjAwMCZpZ3VpZD0zMDE0ZjgzNy0xYTZlLTY5MGUtMjgzNy1lY2ExMWJkYzY4MDUmaW5zaWQ9NTc4Mw&ptn=3&ver=2&hsh=3&fclid=3014f837-1a6e-690e-2837-eca11bdc6805&psq=5.+What+do+you+understand+By+Data+Redundancy%3f&u=a1aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvRGF0YV9yZWR1bmRhbmN5&ntb=1).

Ddl

A **DDL Interpreter** is a component of a Database Management System (DBMS) that processes Data Definition Language (DDL) statements. [These statements are used to define and modify the structure of database objects such as tables, indexes, and views1](https://www.geeksforgeeks.org/structure-of-database-management-system/)[2](https://www.geeksforgeeks.org/data-storage-and-querying-in-dbms/).

**Key Functions of a DDL Interpreter:**

1. [**Interpreting DDL Statements**: It interprets DDL commands like CREATE, ALTER, DROP, and TRUNCATE to define or modify the database schema1](https://www.geeksforgeeks.org/structure-of-database-management-system/)[2](https://www.geeksforgeeks.org/data-storage-and-querying-in-dbms/).
2. **Generating Metadata**: The interpretation of DDL statements results in the creation of metadata, which is data about the data structure. [This metadata is stored in the data dictionary1](https://www.geeksforgeeks.org/structure-of-database-management-system/)[2](https://www.geeksforgeeks.org/data-storage-and-querying-in-dbms/).
3. [**Schema Management**: It helps in managing the database schema by ensuring that the structure of the database is correctly defined and maintained1](https://www.geeksforgeeks.org/structure-of-database-management-system/)[2](https://www.geeksforgeeks.org/data-storage-and-querying-in-dbms/).

**Example:**

When you execute a CREATE TABLE statement, the DDL Interpreter processes this command and creates the necessary metadata entries in the data dictionary to define the new table’s structure.

SELECT date\_format(Joining\_date,'%d-%b-%y %h:%i:%s %p') as Joining\_date FROM `employee` ;