

1. What is relational database management system?

RDBMS is a DBMS that is based on relational model as introduced by Codd.

Rules

Rule 0: The Foundation Rule

Rule 1: Information Rule

The data stored in a database, may it be user data or metadata, must be a value of some table cell. Everything in a database must be stored in a table format.

Rule 2: Guaranteed Access Rule

Every single data element (value) is guaranteed to be accessible logically with a combination of table-name, primary-key (row value), and attribute-name (column value). No other means, such as pointers, can be used to access data.

Rule 3: Systematic Treatment of NULL Values

The NULL values in a database must be given a systematic and uniform treatment. This is a very important rule because a NULL can be interpreted as one the following – data is missing, data is not known, or data is not applicable.

Rule 4: Active Online Catalog

The structure description of the entire database must be stored in an online catalog, known as data dictionary, which can be accessed by authorized users. Users can use the same query language to access the catalog which they use to access the database itself.

Rule 5: Comprehensive Data Sub-Language Rule

A database can only be accessed using a language having linear syntax that supports data definition, data manipulation, and transaction management operations. This language can be used directly or by means of some application. If the database allows access to data without any help of this language, then it is considered as a violation.

Rule 6: View Updating Rule

All the views of a database, which can theoretically be updated, must also be updatable by the system.

Rule 7: High-Level Insert, Update, and Delete Rule

A database must support high-level insertion, updation, and deletion. This must not be limited to a single row, that is, it must also support union, intersection and minus operations to yield sets of data records.

Rule 8: Physical Data Independence

The data stored in a database must be independent of the applications that access the database. Any change in the physical structure of a database must not have any impact on how the data is being accessed by external applications.

Rule 9: Logical Data Independence

The logical data in a database must be independent of its user's view (application). Any change in logical data must not affect the applications using it. For example, if two tables are merged or one is split into two different tables, there should be no impact or change on the user application. This is one of the most difficult rule to apply.

Rule 10: Integrity Independence

A database must be independent of the application that uses it. All its integrity constraints can be independently modified without the need of any change in the application. This rule makes a database independent of the front-end application and its interface.

Rule 11: Distribution Independence

The end-user must not be able to see that the data is distributed over various locations. Users should always get the impression that the data is located at one site only. This rule has been regarded as the foundation of distributed database systems.

Rule 12: Non-Subversion Rule

If a system has an interface that provides access to low-level records, then the interface must not be able to subvert the system and bypass security and integrity constraints.

2. Define concepts of Entity and Attributes

Entity

Real world object, either animate or inanimate, that can be easily identifiable. For example: in school database; students, teachers, classes, courses offered can be considered as entities.

Entity Set

Is a collection of similar types of entities.

Types of Entities

Weak Entity

It is entity type that is existence dependent and has a primary key that is totally and partially constructed from the entity it depends on.

Mandatory and Optional Relationship

Attributes

Entities are represented by means of their properties called attributes. Represented with Oval Shape.

Types of Attributes

Simple Attribute

Atomic values which cannot be divided further. Example: Phone number

Composite Attribute

Made up of more than one simple attribute value. Example: Name might contain First, Middle, Last Name.

Derived Attribute

Don't exist in physical database, but their values are derived/calculated from other attributes present in the database. Example: Age can be derived from DateOfBirth. Represented by DOTTED oval

Multi Value Attribute

Can contain more than one value. Example: Education can have multiple values like Intermediate, Bachelors, and Masters. Represented by Double Oval

Keys

Super Key

A set of attributes (one or more) that collectively identifies an entity in an entity set

Candidate Key

A minimal super key is called a candidate key. An entity set may have more than more candidate key

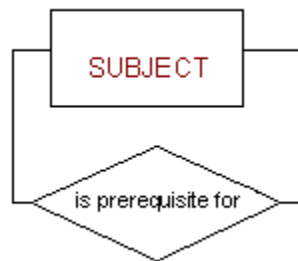
Primary Key

A primary key is one of the candidate chosen by the database designer to uniquely identify the entity set.

Relationship – Types

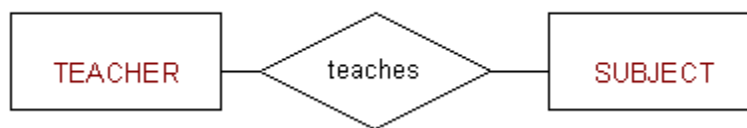
Unary

When both participants in the relationship are of same entity.



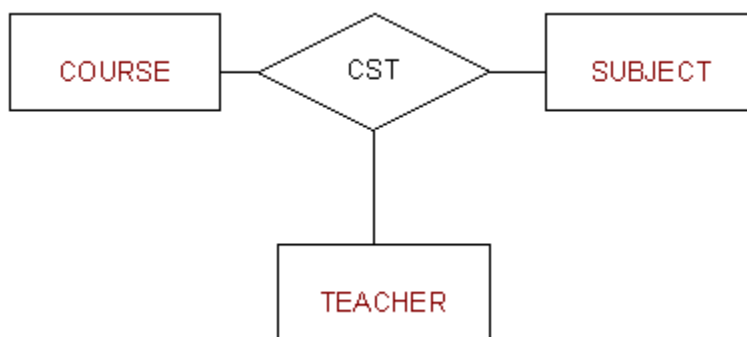
Binary

When two entities participate in relationship



Ternary

When three entities participate in relationship



Cardinality

One-to-One

A Principal teacher manages one Department

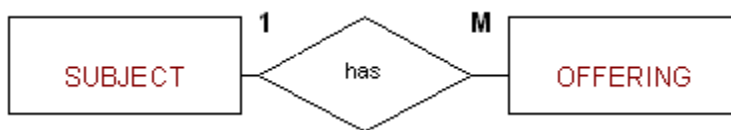
Each Department is managed by One Principal Teacher



One-to-Many

A subject can be offered Many Times

Each Offering belongs to One Subject



Many-to-Many

A Teacher can teach many Different Subjects

Each Subject can be taught by Many Teachers

