Introduction to DBMS

Database management system is a software that controls all the different manipulation of stored or to be stored data in a database. It allows, creation, updation, manipulation, definition of a database.

Top 30 DBMS Interview Questions

1. What is DBMS?

DBMS is a collection of programs that provides users to create and maintain a database. It manages incoming data, Organizes to provide a better and easy way to users. Or in a Simpler way, DBMS provides us an interface or tool for performing different operations such as the creation of a database, inserting data into it, deleting data from it, updating the data, etc. DBMS is a software in which data is stored in a more secure way as compared to the file-based system. Using DBMS, we can overcome many problems such as data redundancy, data inconsistency, easy access, more organized and understandable, and so on.

2. What are the advantages of DBMS?

The advantages of DBMS are as follows:

- **Data Independence:** It allows to change the structure of the data without affecting the structure of any of the running application programs.
- **Sharing of Data:** Multiple users can use data from the same database simultaneously.
- **Integrity constraints:** These constraints allow the data to be stored in a database in a refined manner.
- **Redundancy control:** Supports a mechanism to control the redundancy of data by integrating all the data into a single database.
- **Provide backup and recovery facility:** Provides a feature of ‘backup and recovery’ to automatically create the data backup and restore the data as and when required.

3. What is database partitioning?

It (Database partitioning) is a process where a logical database is divided into different independent parts. The database objects like tables, indexes are subdivided and managed, and accessed at the granular level.

4. Define Normalization?

Organizing data into a relatable Table is known as Normalization.

Normalization in DBMS

A lot of anomalies may be caused in a database because of various cause **insertion, deletion and updation operation or concurrency errors**.

There are 6 different normalisation benchmarks used which are –

- 1 NF (normal form)
- 2 NF
- 3 NF
- BCNF (Boyce-Codd Normal Form
- 4NF
- 5NF (not used anymore)
- 6NF (not used anymore)

1NF

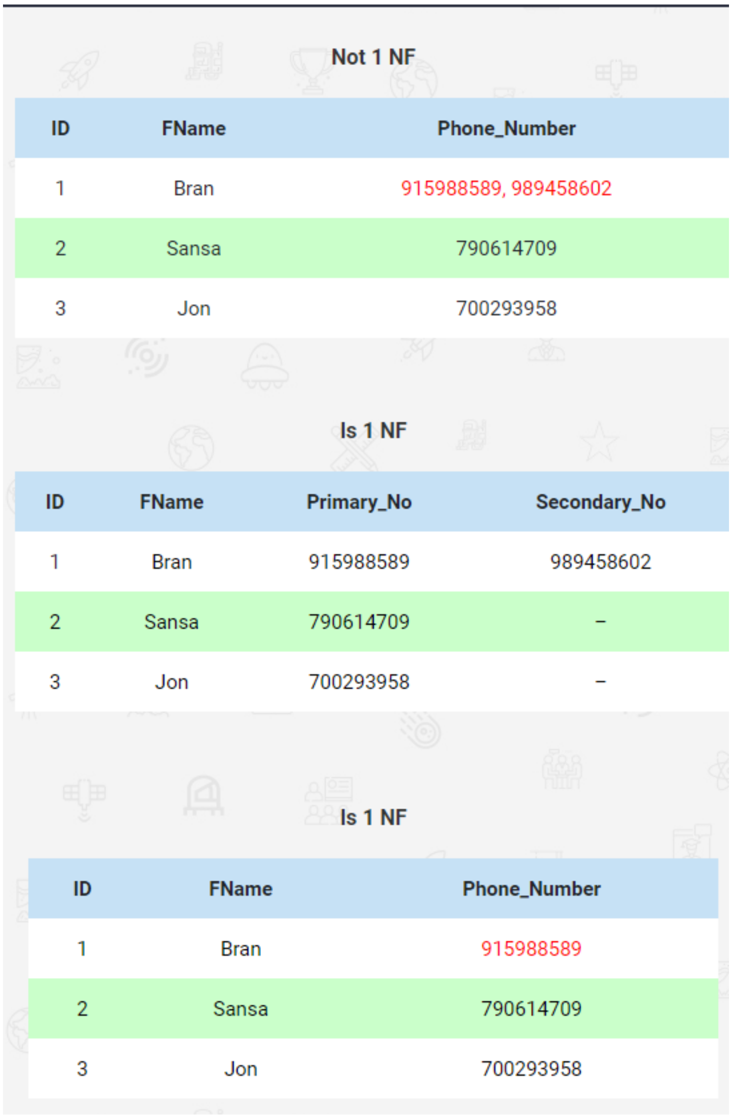
Name – 1 Normal Form

Prerequisite –

The relation **should not have any multivalued attribute** at all in it. For example, for Student table in the database (relation). In the phone number column there must only be one phone number (value).

How to solve –

Decompose the table into single values attributes. Either the second phone number must be deleted or it can be decomposed into primary and secondary number.



Not 1 NF

ID	FName	Phone_Number
1	Bran	915988589, 989458602
2	Sansa	790614709
3	Jon	700293958

Is 1 NF

ID	FName	Primary_No	Secondary_No
1	Bran	915988589	989458602
2	Sansa	790614709	–
3	Jon	700293958	–

Is 1 NF

ID	FName	Phone_Number
1	Bran	915988589
2	Sansa	790614709
3	Jon	700293958

2NF

To understand the conditions for 2NF, one must know the following –

- What is non prime attribute
- What is an candidate Key

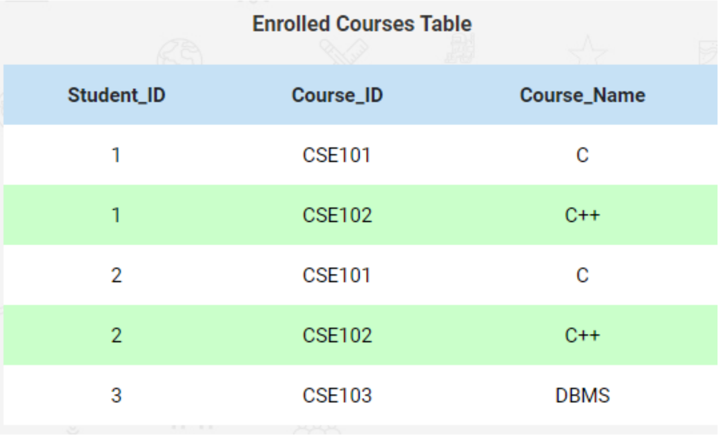
Candidate Key – A candidate key is a column, or set of columns, in a table that can uniquely identify any database record without referring to any other data. The best set of columns which identity it uniquely is chosen as candidate key.

Non Prime Attribute – Is the row that doesn't belong to the candidate key for the table.

Example –

In the table below, if you want to identify any row of the table uniquely, we need both {StudentID, CourseID} thus, it becomes candidate key.

And since CourseName is not part of candidate key. It becomes non-prime attribute.



Enrolled Courses Table

Student_ID	Course_ID	Course_Name
1	CSE101	C
1	CSE102	C++
2	CSE101	C
2	CSE102	C++
3	CSE103	DBMS

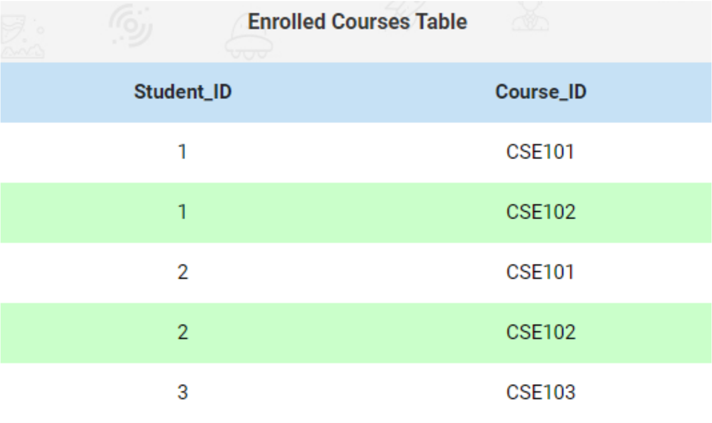
What is 2NF Finally?

To be 2NF in DBMS. The relation should be –

- Already 1NF
- No Non Prime attribute should be dependent on any candidate key element, which is called not having partial dependency. This maybe is difficult to understand from definitions but easier from example.

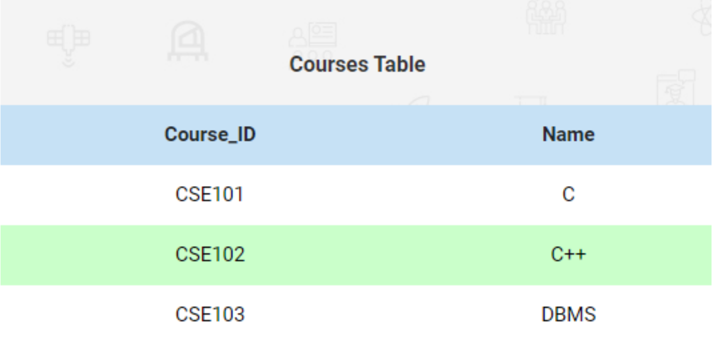
In table, the candidate key is {StudentID, CourseID}, the name of the course is obviously dependent on the CourseID right? CourseID, is non prime also. Thus there is partial dependency in it. Thus, the relation is not 2NF.

To make it 2NF we can break the table as follows –



Enrolled Courses Table

Student_ID	Course_ID
1	CSE101
1	CSE102
2	CSE101
2	CSE102
3	CSE103



Courses Table

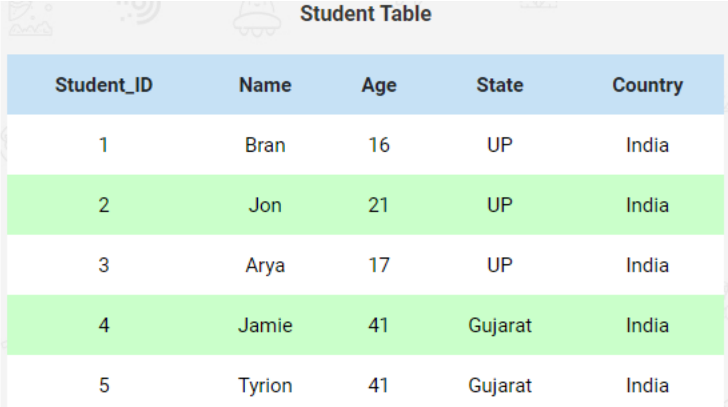
Course_ID	Name
CSE101	C
CSE102	C++
CSE103	DBMS

3NF

For a relation to be in 3NF form the following must hold true –

- Should be 2NF already
- There should not be any transitive dependency for non prime attributes

Don't worry if you don't know what transitive dependency is, we will understand this with an example.



Student Table

Student_ID	Name	Age	State	Country
1	Bran	16	UP	India
2	Jon	21	UP	India
3	Arya	17	UP	India
4	Jamie	41	Gujarat	India
5	Tyrion	41	Gujarat	India

Now, for the table above. The primary key obviously is StudentID. However, the following dependencies are there –

1. StudentID -> Name
2. StudentID -> Age
3. **StudentID -> State**
4. **State -> Country**

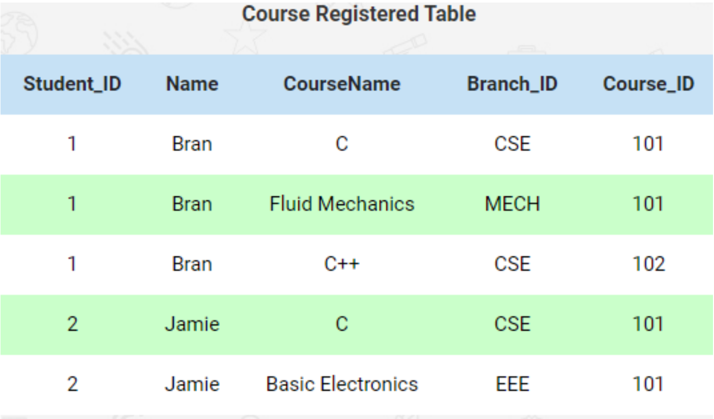
Now, when we look at point 3 and 4 they form transitive dependency as they are of format A -> B and B -> C. Thus, this forms transitive dependency is not in 3NF.

BCNF

The BCNF is the short for Boyce-Codd Normal Form the following are the conditions for a relation to be in BCNF form –

- Should be 3NF
- If every non-trivial functional dependency A –> B, A is a super key, i.e. LHS should always be a super key.

Let's have a look onto an example to understand this better. Let's say a student enrolling a course can also register for courses from other branches and the course registration table for student is stored in the following way –



Course Registered Table

Student_ID	Name	CourseName	Branch_ID	Course_ID
1	Bran	C	CSE	101
1	Bran	Fluid Mechanics	MECH	101
1	Bran	C++	CSE	102
2	Jamie	C	CSE	101
2	Jamie	Basic Electronics	EEE	101

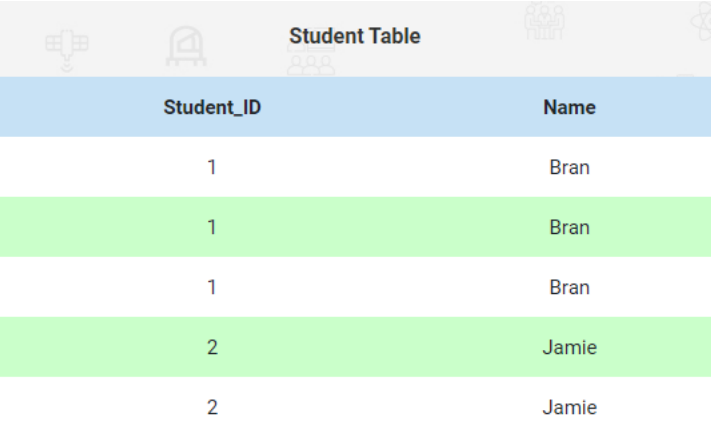
Candidate Key – {StudentID, CourseName}

- To identify each row uniquely we need {StudentID, CourseName}
- Thus, its the candidate key

Functional dependencies are –

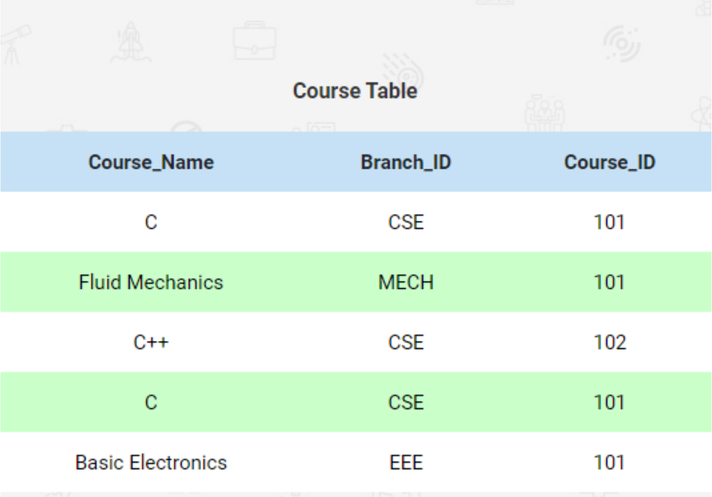
1. StudentID -> Name
2. CourseName -> {BranchID, CourseID}

Now, neither the 2nd is a dependency of type X -> Y, nor CourseName is a super key. **Note –** Set of attributes that can help us uniquely identify a record or tuple in the database is called super key.



Student Table

Student_ID	Name
1	Bran
1	Bran
1	Bran
2	Jamie
2	Jamie



Course Table

Course_Name	Branch_ID	Course_ID
C	CSE	101
Fluid Mechanics	MECH	101
C++	CSE	102
C	CSE	101
Basic Electronics	EEE	101

4NF

To be in 4NF form the follwing must be true

1. Should be in BCNF form
2. Should not have any Multi-Valued Dependency.

Example –

A -> BC

5. Enlist the advantages of a normalizing database?

Advantages of the normalizing database are:

- No duplicate entries
- Saves storage space
- Boasts the query performances.

6. What are the different levels of abstraction in the DBMS?

There are 3 levels of data abstraction in the DBMS.
They include:

- Physical Level: This is the lowest level of the data abstraction which states how the data is stored in the database.

- Logical Level: This is the next level of the data abstraction which states the type of the data and the relationship among the data that is stored in the database.

- View Level: This is the highest level in the data abstraction which shows/states only a part of the database.

7. What is a checkpoint in DBMS and when does it occur?

A checkpoint is a mechanism where all the previous logs are removed from the system and are permanently stored on the storage disk. So, basically, checkpoints are those points from where the transaction log record can be used to recover all the committed data up to the point of crash.

8. What is the Relationship?

The Relationship is defined as an association among two or more entities. There are three types of relationships in DBMS-

- One-To-One: Here one record of any object can be related to one record of another object.
- One-To-Many (many-to-one): Here one record of any object can be related to many records of other objects and vice versa.
- Many-to-many: Here more than one record of an object can be related to n number of records of another object.

9. What is RDBMS?

RDBMS is the Relational Database Management System which contains data in the form of the tables and data is accessed on the basis of the common fields among the tables.

10. Mention the different types of DBMS?

The different types of DBMS are as follows:

- **Relational DBMS (RDBMS):** This type of DBMS, uses a structure that allows the users to access data in relation to another piece of data in a database. In this type of DBMS, data is stored in the form of tables.

- **Hierarchical DBMS:** As the name suggests, this type of DBMS has a structure similar to that of a tree, wherein the nodes represent records and the branches of the tree represent fields.
- **Network DBMS:** This type of DBMS supports many-to-many relations wherein multiple member records can be linked.
- **Object-oriented DBMS:** Uses small individual software called to object to store pieces of data and the instructions for the actions to be done with the data**.**

11. Enlist the disadvantages of the query?

The disadvantages of the query are:

- No indexes
- Stored procedures are excessively compiled.
- Triggers and procedures are without SET NOCOUNT ON.
- Complicated joins making up inadequately written queries.
- Cursors and temporary tables showcase a bad presentation.

12. What is the purpose of normalization in DBMS?

Normalization is the process of analyzing the relational schemas which are based on their respective functional dependencies and the primary keys to fulfilling certain properties.

The properties include:

- To minimize the redundancy of the data.
- To minimize the Insert, Delete, and Update Anomalies.

13. What is an ER diagram in DBMS?

An entity-relationship model or an entity-relationship diagram is a visual representation of data that is represented as entities, attributes, and relationships are set between entities.

14. What are the integrity rules in DBMS?

Data integrity is one significant aspect while maintaining the database. So, data integrity is enforced in the database system by imposing a series of rules. Those set of integrity is known as the integrity rules.

There are two integrity rules in DBMS**:**

Entity Integrity: It specifies that the “Primary key cannot have a NULL value.”

Referential Integrity: It specifies that the “Foreign Key can be either a NULL value or should be the Primary Key value of other relation

15. Define Atomicity and Aggregation?

- **Atomicity:** It’s an all or none concept which enables the user to be assured of incomplete transactions to be taken care of. The actions involving incomplete transactions are left undone in DBMS.
- **Aggregation:** The collected entities and their relationship are aggregated in this model. It is mainly used in expressing relationships within relationships.

16. What is an entity-relationship model?

It is a diagrammatic approach to database design, where you represent real-world objects as entities and mention relationships between them. This approach helps the team of DBA’s to understand the schema easily.

17. Popular DBMS Software?

Here, is the list of some popular DBMS systems:

- MySQL
- Microsoft Access
- Oracle
- PostgreSQL
- dBASE
- FoxPro
- SQLite
- IBM DB2
- LibreOffice Base
- MariaDB
- Microsoft SQL Server etc.

18. What is concurrency control?

This is a process managing simultaneous operations in a database so that database integrity is not compromised. The following are the two approaches involved in concurrency control:

- Optimistic approach – Involves versioning
- Pessimistic approach – Involves locking

19. What is the purpose of SQL?

SQL stands for **Structured Query Language** whose main purpose is to interact with the relational databases in **the** form of inserting and updating/modifying the data in the database.

20. Disadvantage of DBMS?

DBMS may offer plenty of advantages but, it has certain flaws-

- Cost of Hardware and Software of a DBMS is quite high which increases the budget of your organization.
- Most database management systems are often complex systems, so training for users to use the DBMS is required.

- In some organizations, all data is integrated into a single database which can be damaged because of electric failure or database is corrupted on the storage media
- Use of the same program at a time by many users sometimes leads to the loss of some data.
- DBMS can't perform sophisticated calculations

21. What are the unary operations in Relational Algebra?

Unary operations are operations that use single operands. Unary operations in relational algebra are PROJECTION and SELECTION.

-=,>=,<=,+= are the relational operators used in SELECTION.

22. How many types of database languages are?

There are four types of database languages:

- **Data Definition Language (DDL)**: These commands are used for updating the data. CREATE, ALTER, DROP, TRUNCATE, RENAME are some examples of DDL commands.
- **Data Manipulation Language (DML)**: These commands are used for the manipulation of already updated data. SELECT, UPDATE, INSERT, DELETE are some examples of DML commands.
- **Data Control Language (DCL) :** These commands are used for giving and removing user access to the database. GRANT and REVOKE are the examples of DCL commands.
- **Transaction Control Language (TCL) :** These are the commands used for managing transactions in the database. TCL is used for managing the changes made by DML. COMMIT, ROLLBACK, and SAVEPOINT are the examples of TCL commands.

23. Define a Relation Schema?

A relation schema is known as the blueprint with the help of which we can explain how the data is organized into tables.

24. What is a degree of Relation?

A degree of relation is also known as Cardinality it is defined as the number of occurrence of one entity which is connected to the number of occurrence of other entity.

25. What are the disadvantages of file processing systems?

There are some disadvantages of file processing system :

- Inconsistent
- Not secure
- Data redundancy
- Difficult in accessing data
- Data isolation
- Data integrity
- Concurrent access is not possible
- Limited data sharing
- Atomicity problem

26. What is data abstraction in DBMS?

Data abstraction in DBMS is a process of hiding irrelevant details from users. Because database systems are made of complex data structures so, it makes accessible the user interaction with the database.

27. What is Relational Algebra?

Relational Algebra is a Procedural Query Language that contains a set of operations that take one or two relations as input and produce a new relationship. Relational algebra is the basic set of operations for the relational model. The decisive point of relational algebra is that it is similar to the algebra which operates on the number.

There are a few fundamental operations of relational algebra:

- select
- project
- set difference
- union
- rename, etc.

28. What is Relational Calculus?

Relational Calculus is a Non-procedural Query Language that uses mathematical predicate calculus instead of algebra. Relational calculus doesn't work on mathematics fundamentals such as algebra, differential, integration, etc. That's why it is also known as predicate calculus.

There is two types of relational calculus:

- Tuple relational calculus
- Domain relational calculus

29. Explain ACID properties?

ACID properties are some basic rules, which have to be satisfied by every transaction to preserve the integrity. These properties and rules are:

ATOMICITY: Atomicity is more generally known as ? all or nothing rule.' This implies all are considered as one unit, and they either run to completion or are not executed at all.

CONSISTENCY: This property refers to the uniformity of the data. Consistency implies that the database is consistent before and after the transaction.

ISOLATION: This property states that the number of transactions can be executed concurrently without leading to the inconsistency of the database state.

DURABILITY: This property ensures that once the transaction is committed it will be stored in the non-volatile memory and a system crash can also not affect it anymore.

30. What is the 3-Tier architecture?

The 3-Tier architecture contains another layer between the client and server. The introduction of 3-tier architecture is for the ease of the users as it provides the GUI, which, makes the system secure and much more accessible. In this architecture, the application on the client-end interacts with an application on the server which further communicates with the database system.