interviewbit.com

15+ Top DBMS Interview Questions (2021) - Interviewbit

Kings Gambit Labs

13-16 minutes

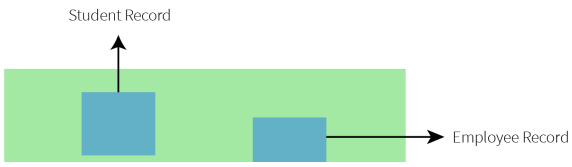
To consolidate your knowledge and concepts in DBMS, here we've listed the most commonly asked DBMS interview questions to help you ace your interview!

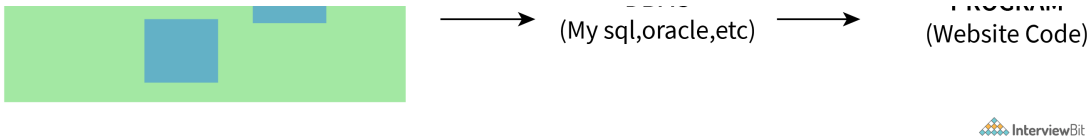
1. What is meant by DBMS and what is its utility?

Explain RDBMS with examples.

As the name suggests DBMS or Database Management System is a set of applications or programs that enable users to create and maintain a database. DBMS provides a tool or an interface for performing various operations such as inserting, deleting, updating, etc. into a database. It is software that enables the storage of data more compactly and securely as compared to a file-based system. A DBMS system helps a user to overcome problems like data inconsistency, data redundancy, etc. in a database and makes it more convenient and organized to use it.

Examples of popular DBMS systems are file systems, XML, Windows Registry, etc.





RDBMS stands for Relational Database Management System and was introduced in the 1970s to access and store data more efficiently than DBMS. RDBMS stores data in the form of tables as compared to DBMS which stores data as files. Storing data as rows and columns makes it easier to locate specific values in the database and makes it more efficient as compared to DBMS.

Examples of popular RDBMS systems are MySQL, Oracle DB, etc.

2. What is meant by a database?

A Database is an organized, consistent, and logical collection of data that can easily be updated, accessed, and managed. Database mostly contains sets of tables or objects (anything created using create command is a database object) which consist of records and fields. A tuple or a row represents a single entry in a table. An attribute or a column represents the basic units of data storage, which contain information about a particular aspect of the table. DBMS extracts data from a database in the form of queries given by the user.

3. Mention the issues with traditional file-based systems that make DBMS a better choice?

The absence of indexing in a traditional file-based system leaves us with the only option of scanning the full page and hence making the access of content tedious and super slow. The other issue is redundancy and inconsistency as files have many duplicate and redundant data and changing one of them makes all of them inconsistent. Accessing data is harder in traditional file-based systems because data is

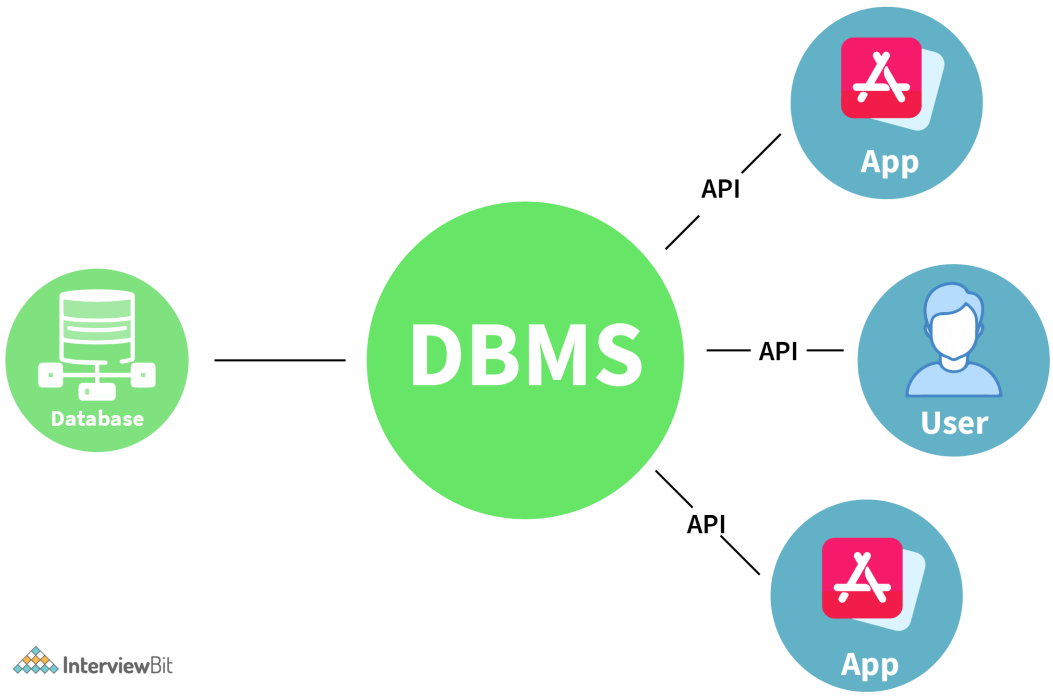
unorganized in them.

Another issue is the lack of concurrency control, which leads to one operation locking the entire page, as compared to DBMS where multiple operations can work on a single file simultaneously.

Integrity check, data isolation, atomicity, security, etc. are some other issues with traditional file-based systems for which DBMSs have provided some good solutions.

4. Explain a few advantages of a DBMS.

Following are the few advantages of using a DBMS.



- **Data Sharing:** Data from a single database can be simultaneously shared by multiple users. Such sharing also enables end-users to react to changes quickly in the database environment.
- **Integrity constraints:** The existence of such constraints allows storing of data in an organized and refined manner.
- **Controlling redundancy in a database:** Eliminates

redundancy in a database by providing a mechanism that integrates all the data in a single database.

- **Data Independence:** This allows changing the data structure without altering the composition of any of the executing application programs.
- **Provides backup and recovery facility:** It can be configured to automatically create the backup of the data and restore the data in the database whenever required.
- **Data Security:** DBMS provides the necessary tools to make the storage and transfer of data more reliable and secure. Authentication (the process of giving restricted access to a user) and encryption (encrypting sensitive data such as OTP, credit card information, etc.) are some popular tools used to secure data in a DBMS.

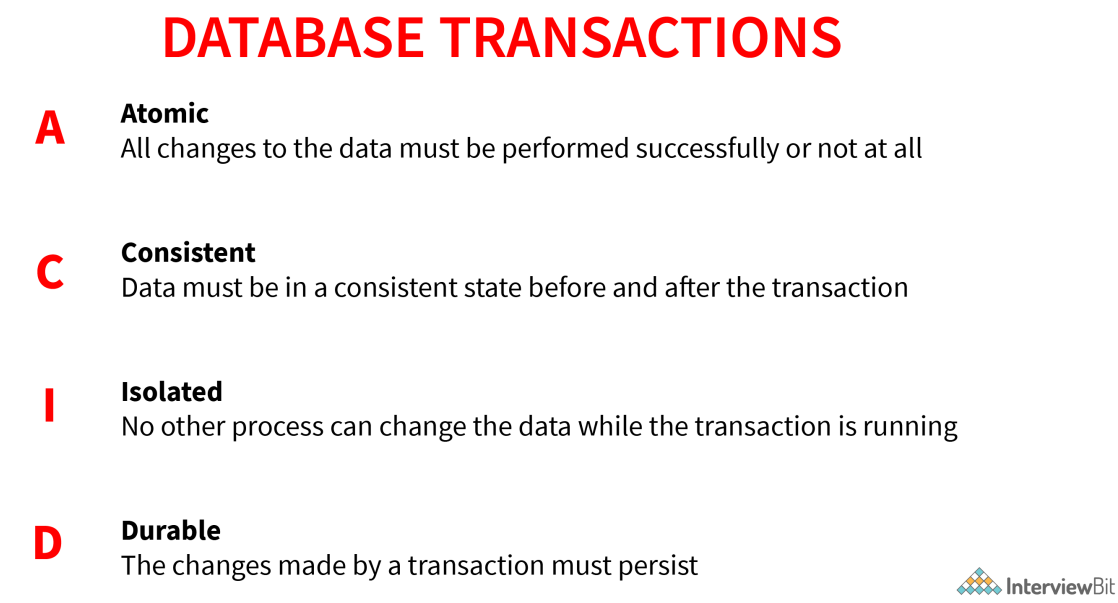
5. Explain different languages present in DBMS.

Following are various languages present in DBMS:

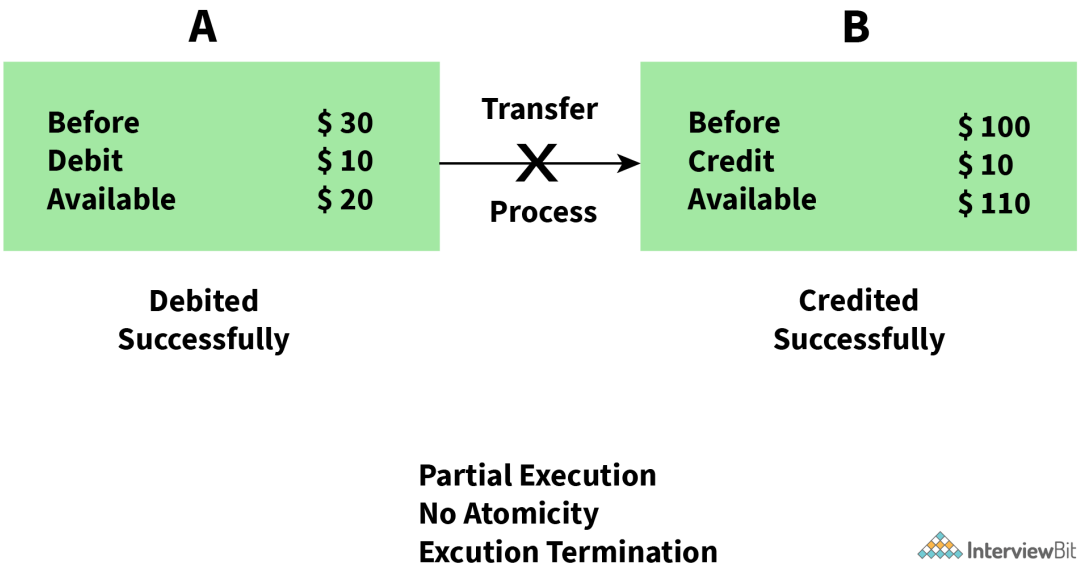
- **DDL(Data Definition Language):** It contains commands which are required to define the database.
  E.g., CREATE, ALTER, DROP, TRUNCATE, RENAME, etc.
- **DML(Data Manipulation Language):** It contains commands which are required to manipulate the data present in the database.
  E.g., SELECT, UPDATE, INSERT, DELETE, etc.
- **DCL(Data Control Language):** It contains commands which are required to deal with the user permissions and controls of the database system.
  E.g., GRANT and REVOKE.
- **TCL(Transaction Control Language):** It contains commands which are required to deal with the transaction of the database.
  E.g., COMMIT, ROLLBACK, and SAVEPOINT.

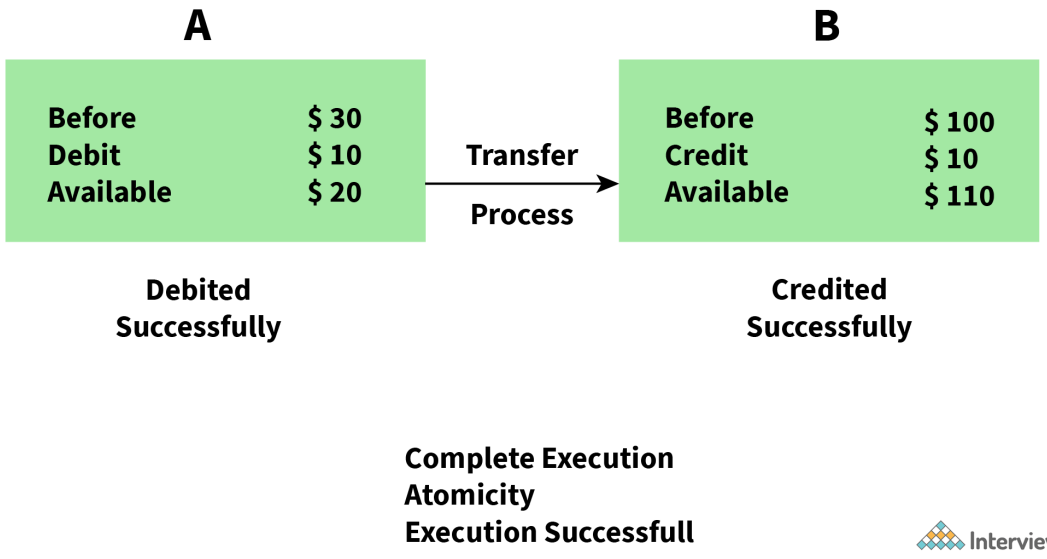
6. What is meant by ACID properties in DBMS?

ACID stands for Atomicity, Consistency, Isolation, and Durability in a DBMS these are those properties that ensure a safe and secure way of sharing data among multiple users.

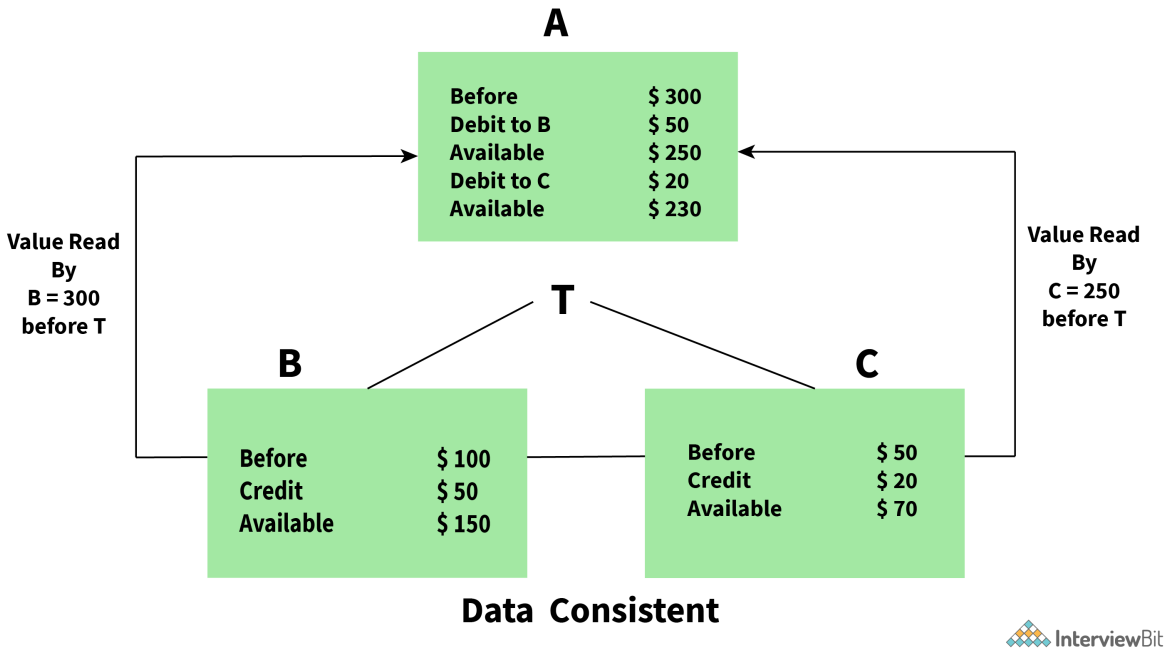


- **Atomicity**: This property reflects the concept of either executing the whole query or executing nothing at all, which implies that if an update occurs in a database then that update should either be reflected in the whole database or should not be reflected at all.

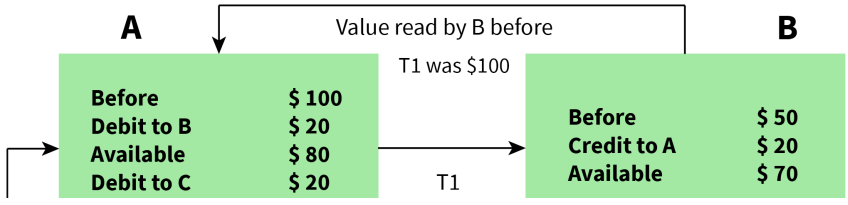


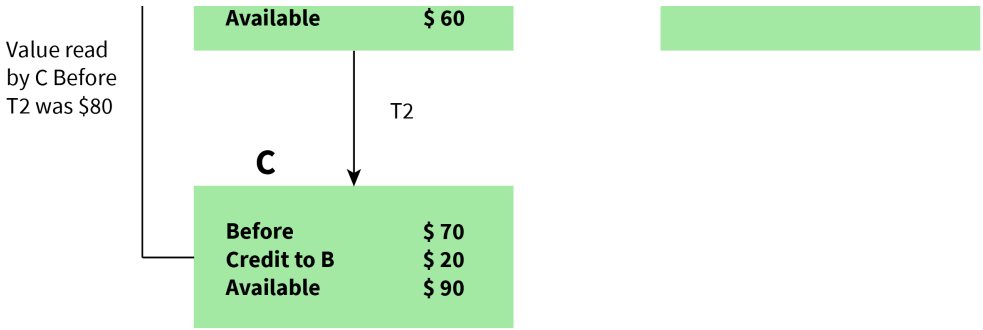


- **Consistency:** This property ensures that the data remains consistent before and after a transaction in a database.



- **Isolation:** This property ensures that each transaction is occurring independently of the others. This implies that the state of an ongoing transaction doesn't affect the state of another ongoing transaction.





Isolation - Independent execution T1 and T2 by A

InterviewBit

- **Durability:** This property ensures that the data is not lost in cases of a system failure or restart and is present in the same state as it was before the system failure or restart.

7. Are NULL values in a database the same as that of blank space or zero?

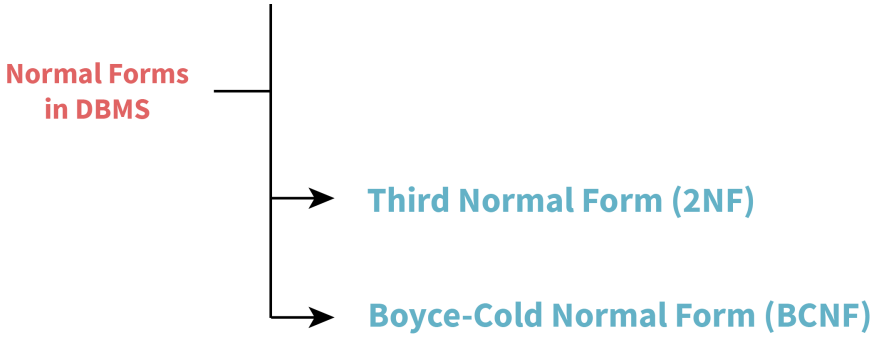
No, a NULL value is very different from that of zero and blank space as it represents a value that is assigned, unknown, unavailable, or not applicable as compared to blank space which represents a character and zero represents a number.

Example: NULL value in “number_of_courses” taken by a student represents that its value is unknown whereas 0 in it means that the student hasn’t taken any courses.

Advanced DBMS Interview Questions

16. Explain different types of Normalization forms in a DBMS.

Following are the major normalization forms in a DBMS:



Full Names	Physical Address	Movies Rented	Salutation
Janet Jones	First street Plot No 4	Pirates. the Caribbean Clash of the Titans	Ms.
Robert Phil	3rd street 34	Forgetting Sarah Marshal Daddy's Little Girls	Mr.
Robert Phil	5th Avenue	Clash of the Titans	Mr.

InterviewBit

Considering the above Table-1 as the reference example for understanding different normalization forms.

- 1NF: It is known as the first normal form and is the simplest type of normalization that you can implement in a database. A table to be in its first normal form should satisfy the following conditions:
- Every column must have a single value and should be atomic.
- Duplicate columns from the same table should be removed.
- Separate tables should be created for each group of related data and each row should be identified with a unique column.

Full Names	Physical Address	Movies Rented	Salutation
Janet Jones	First street Plot No 4	Pirates. the Caribbean	Ms.

Janet Jones	First street Plot No 4	Clash of the Titans	Ms.
Robert Phil	3rd street 34	Forgetting Sarah Marshal	Mr.
Robert Phil	3rd street 34	Daddy's Little Girls	Mr.
Robert Phil	5th Avenue	Clash of the Titans	Mr.

InterviewBit

Table-1 converted to 1NF form

- **2NF:** It is known as the second normal form. A table to be in its second normal form should satisfy the following conditions:
- The table should be in its 1NF i.e. satisfy all the conditions of 1NF.
- Every non-prime attribute of the table should be fully functionally dependent on the primary key i.e. every non-key attribute should be dependent on the primary key in such a way that if any key element is deleted then even the non_key element will be saved in the database.

Membership ID	Full Names	Physical Address	Salutation
1	Janet Jones	First street Plot No 4	Ms.
2	Robert Phil	3rd street 34	Mr.
3	Robert Phil	5th Avenue	Mr.

InterviewBit

Membership ID	Movies Rented
1	Pirates. the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

InterviewBit

Breaking Table-1 into 2 different tables to move it to 2NF.

- 3NF: It is known as the third normal form. A table to be in its second normal form should satisfy the following conditions:
- The table should be in its 2NF i.e. satisfy all the conditions of 2NF.
- There is no transitive functional dependency of one attribute on any attribute in the same table.

Membership ID	Full Names	Physical Address	SalutationID
1	Janet Jones	First street Plot No 4	2
2	Robert Phil	3rd street 34	1
3	Robert Phil	5th Avenue	1

InterviewBit

Membership ID	Movies Rented
1	Pirates. the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

InterviewBit

Salutation ID	Salutation
1	Mr.
2	Ms.
3	Mrs.
4	Dr.

InterviewBit

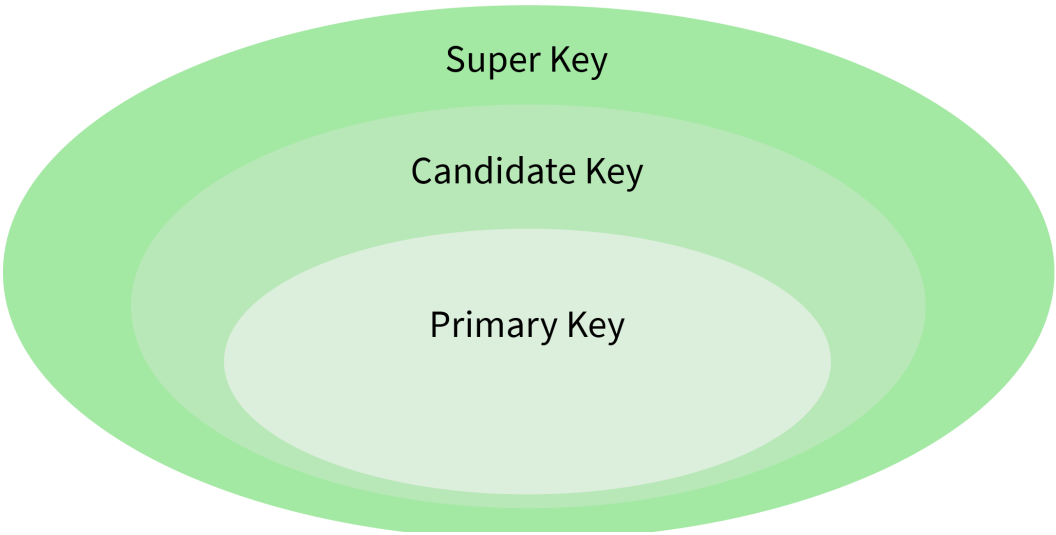
Breaking Table-1 into 3 different tables to move it to 3NF.

- **BCNF:** BCNF stands for Boyce-Codd Normal Form and is an advanced form of 3NF. It is also referred to as 3.5NF for the same reason. A table to be in its BCNF normal form should satisfy the following conditions:
- The table should be in its 3NF i.e. satisfy all the conditions of 3NF.
- For every functional dependency of any attribute A on B (A->B), A should be the super key of the table. It simply implies that A can't be a non-prime attribute if B is a prime attribute.

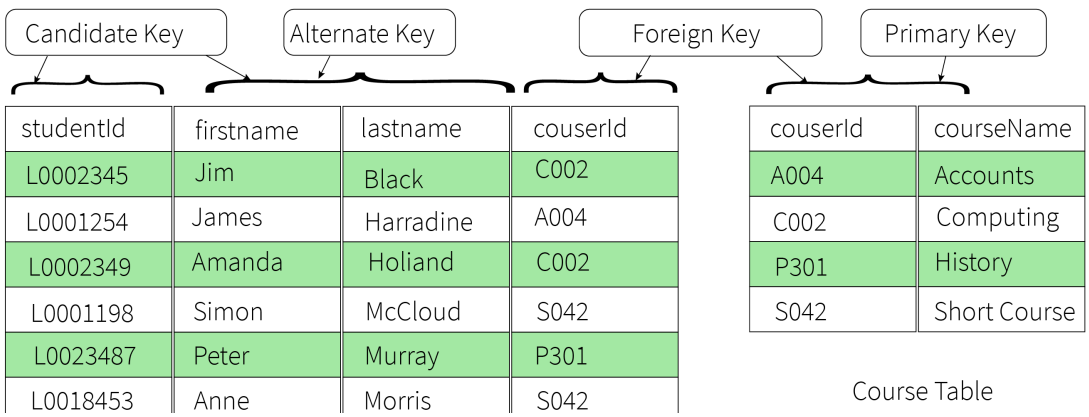
17. Explain different types of keys in a database.

There are mainly 7 types of keys in a database:

- **Candidate Key:** The candidate key represents a set of properties that can uniquely identify a table. Each table may have multiple candidate keys. One key amongst all candidate keys can be chosen as a primary key. In the below example since studentId and firstName can be considered as a Candidate Key since they can uniquely identify every tuple.
- **Super Key:** The super key defines a set of attributes that can uniquely identify a tuple. Candidate key and primary key are subsets of the super key, in other words, the super key is their superset.



- **Primary Key:** The primary key defines a set of attributes that are used to uniquely identify every tuple. In the below example studentId and firstName are candidate keys and any one of them can be chosen as a Primary Key. In the given example studentId is chosen as the primary key for the student table.
- **Unique Key:** The unique key is very similar to the primary key except that primary keys don't allow NULL values in the column but unique keys allow them. So essentially unique keys are primary keys with NULL values.
- **Alternate Key:** All the candidate keys which are not chosen as primary keys are considered as alternate Keys. In the below example, firstname and lastname are alternate keys in the database.
- **Foreign Key:** The foreign key defines an attribute that can only take the values present in one table common to the attribute present in another table. In the below example courseId from the Student table is a foreign key to the Course table, as both, the tables contain courseId as one of their attributes.
- **Composite Key:** A composite key refers to a combination of two or more columns that can uniquely identify each tuple in a table. In the below example the studentId and firstname can be grouped to uniquely identify every tuple in the table.



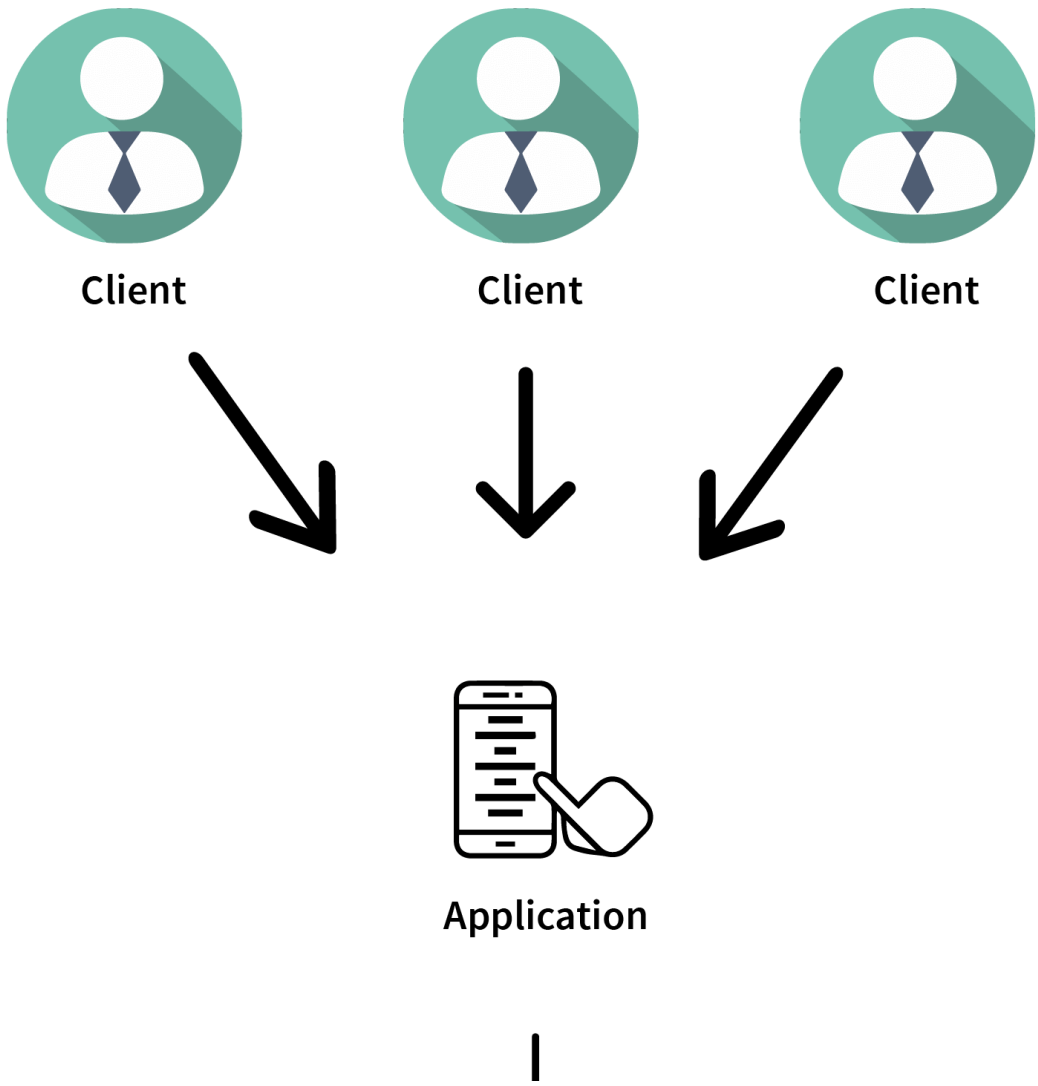
Course Table

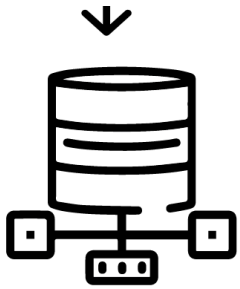
Relationship Between Keys

InterviewBit

18. Explain the difference between a 2-tier and 3-tier architecture in a DBMS.

The **2-tier architecture** refers to the client-server architecture in which applications at the client end directly communicate with the database at the server end without any middleware involved.
Example – Contact Management System created using MS-Access or Railway Reservation System, etc.





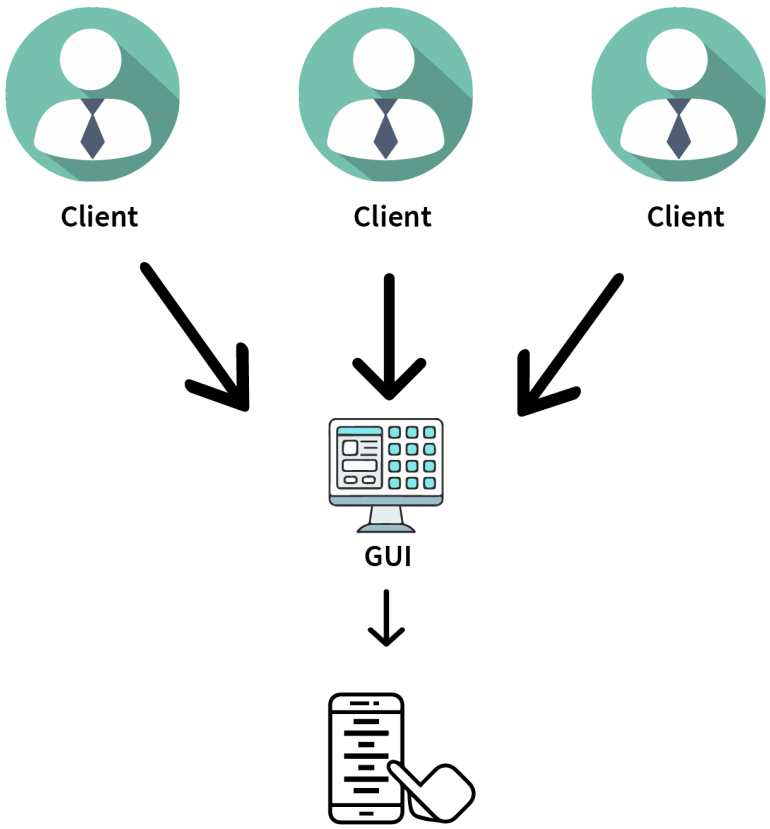
Database System

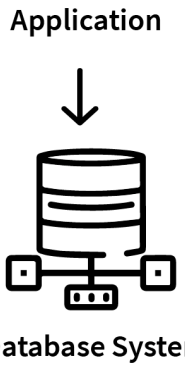
InterviewBit

The above picture represents a 2-tier architecture in a DBMS.

The **3-tier architecture** contains another layer between the client and the server to provide GUI to the users and make the system much more secure and accessible. In this type of architecture, the application present on the client end interacts with an application on the server end which further communicates with the database system.

Example – Designing registration form which contains a text box, label, button or a large website on the Internet, etc.





The above picture represents a 3-tier architecture in a DBMS.

Recommended Tutorials:

SQL Interview Questions
SQL Server Interview Questions
MySQL Interview Questions
MongoDB Interview Questions
PL SQL Interview Questions

Dbms-interview MCQs

Circle

Rectangle

Ellipse

Diamond Box

Attribute

Tuple

Relation

Criteria

Oral Database Connectivity

Oracle Database Connectivity

Open Database Connectivity

Object Database Connectivity

Many teachers have one class

Many teachers have many classes

One teacher has many classes

One teacher has one class

Directory

Sub Data

Warehouse

Meta Data

Data Administrator

Database Administrator

Data Bank Administrator

None of the above

Ternary Control Language

Transmission Control Language

Transaction Central Language

Transaction Control Language