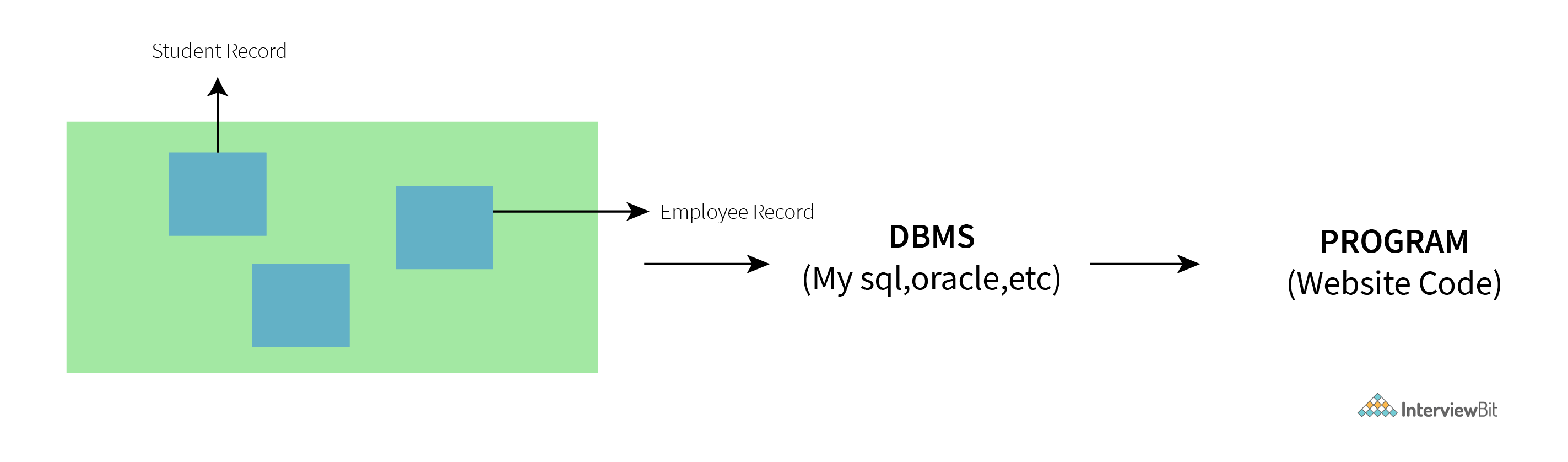
**DBMS Basic Interview Questions**

**1. What is DBMS and what is its utility? Explain RDBMS with examples.**

DBMS stands for [**Database Management System**](https://www.interviewbit.com/blog/components-of-dbms/)**,** 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. Check this [DBMS Tutorial by Scaler Topics](https://www.scaler.com/topics/dbms/).

**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.

[**Learn More**](https://www.interviewbit.com/blog/difference-between-dbms-and-rdbms/)

**2. What is 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.

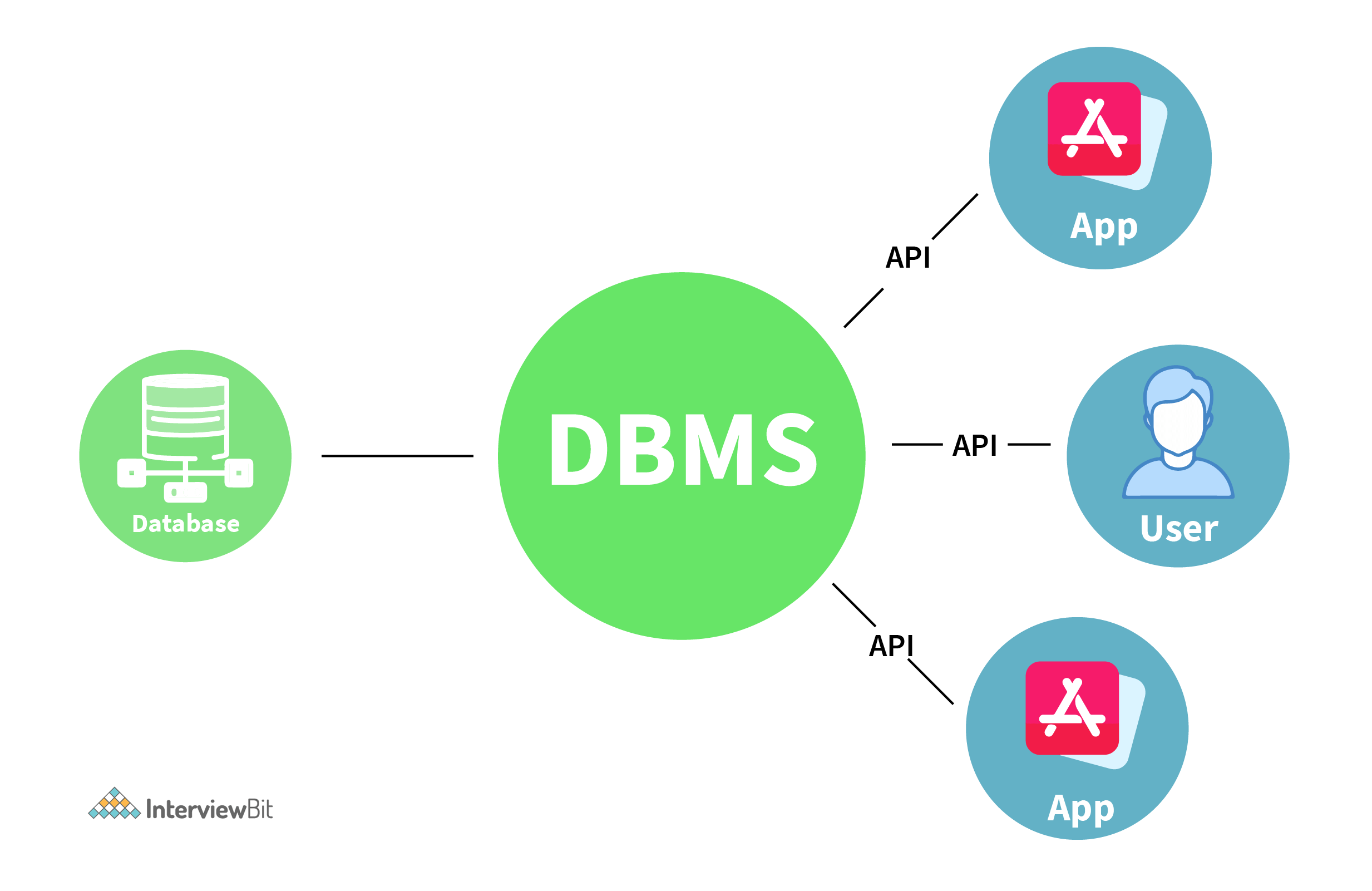
Integrity check, data isolation, atomicity, security, etc. are some other issues with traditional file-based systems for which DBMSs have provided some good solutions.

**You can download a PDF version of Dbms Interview Questions.**

[**Download PDF**](javascript:void(0))

**4. Explain a few advantages of a DBMS.**

Following are the few [**advantages of using a DBMS**](https://www.interviewbit.com/blog/features-of-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.

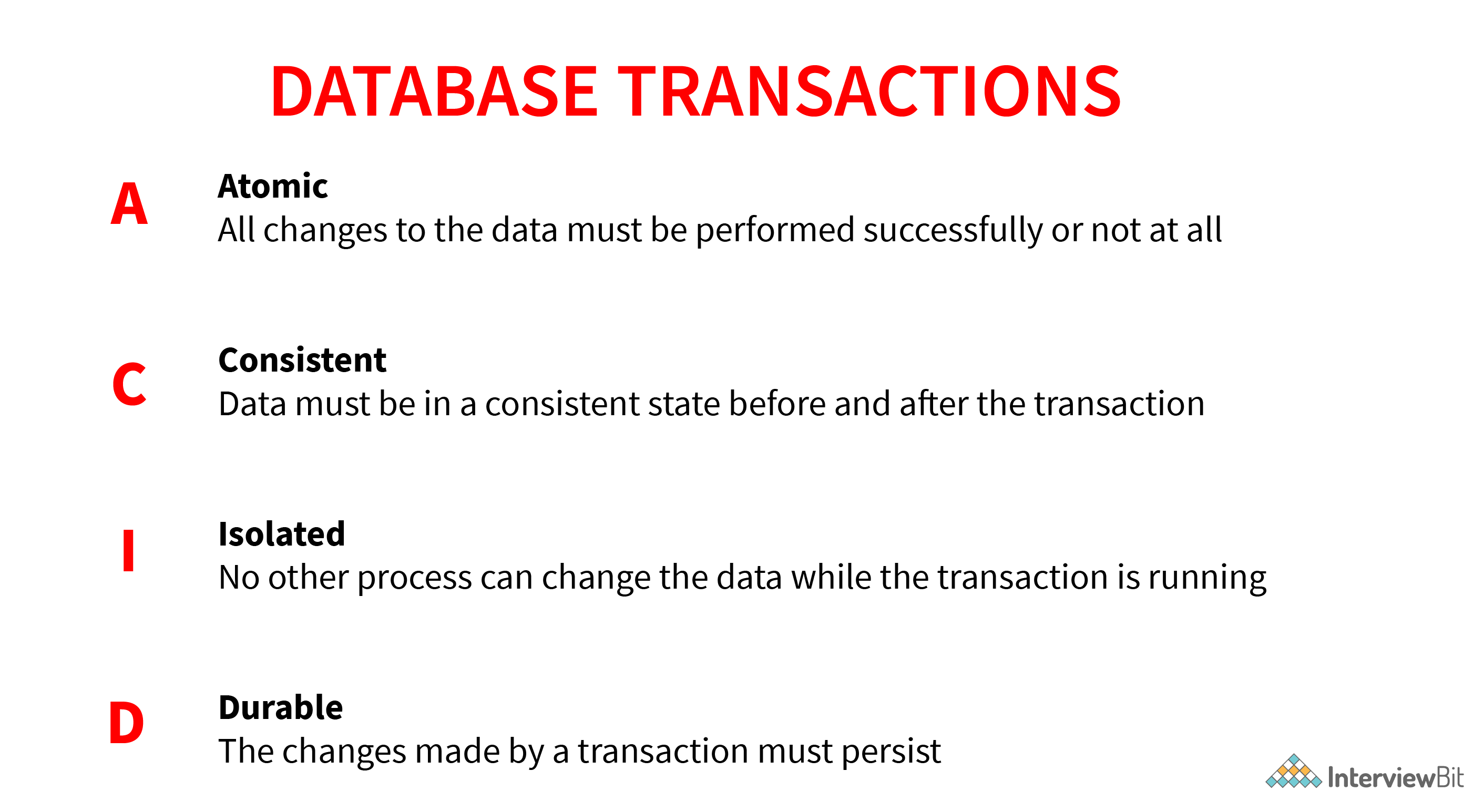
**5. Explain different languages present in DBMS.**

Following are various languages present in DBMS:

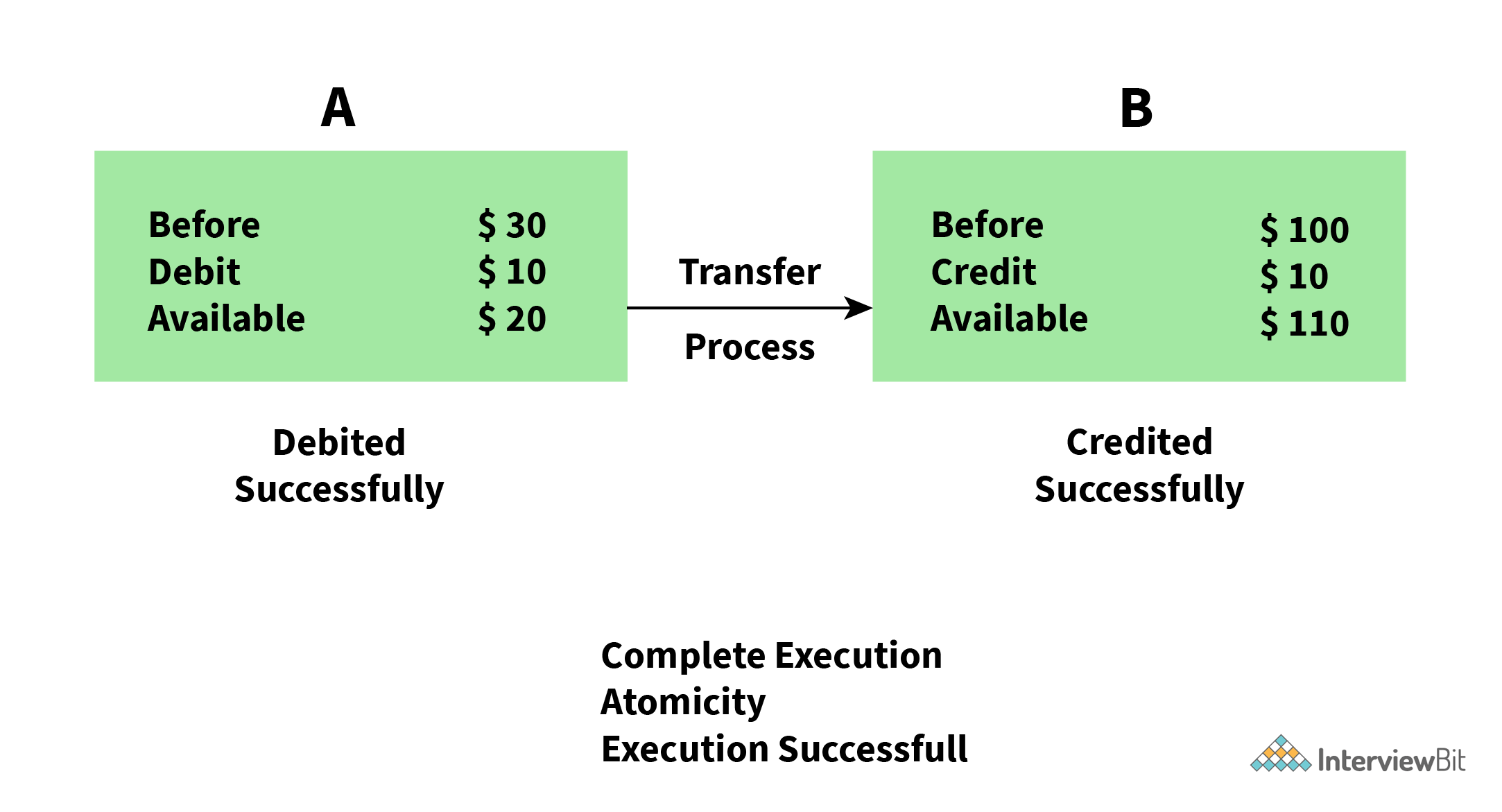
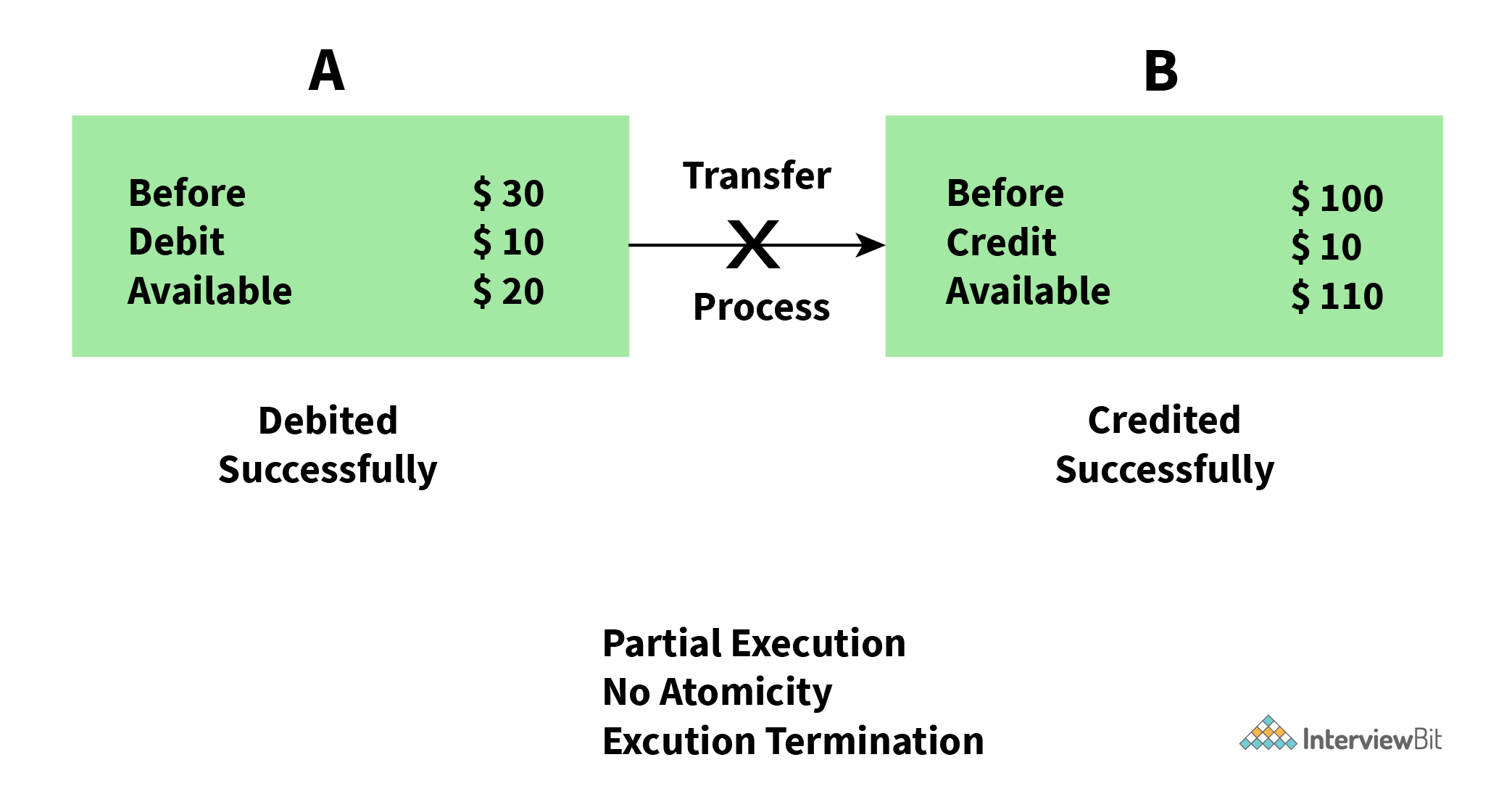
* **DDL(Data Definition Language):**  It contains commands which are required to define the database.  
  E.g., CREATE, ALTER, DROP, TRUNCATE, RENAME, etc.
* **DML(Data Manipulation Language):** It contains commands which are required to manipulate the data present in the database.  
  E.g., SELECT, 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?**

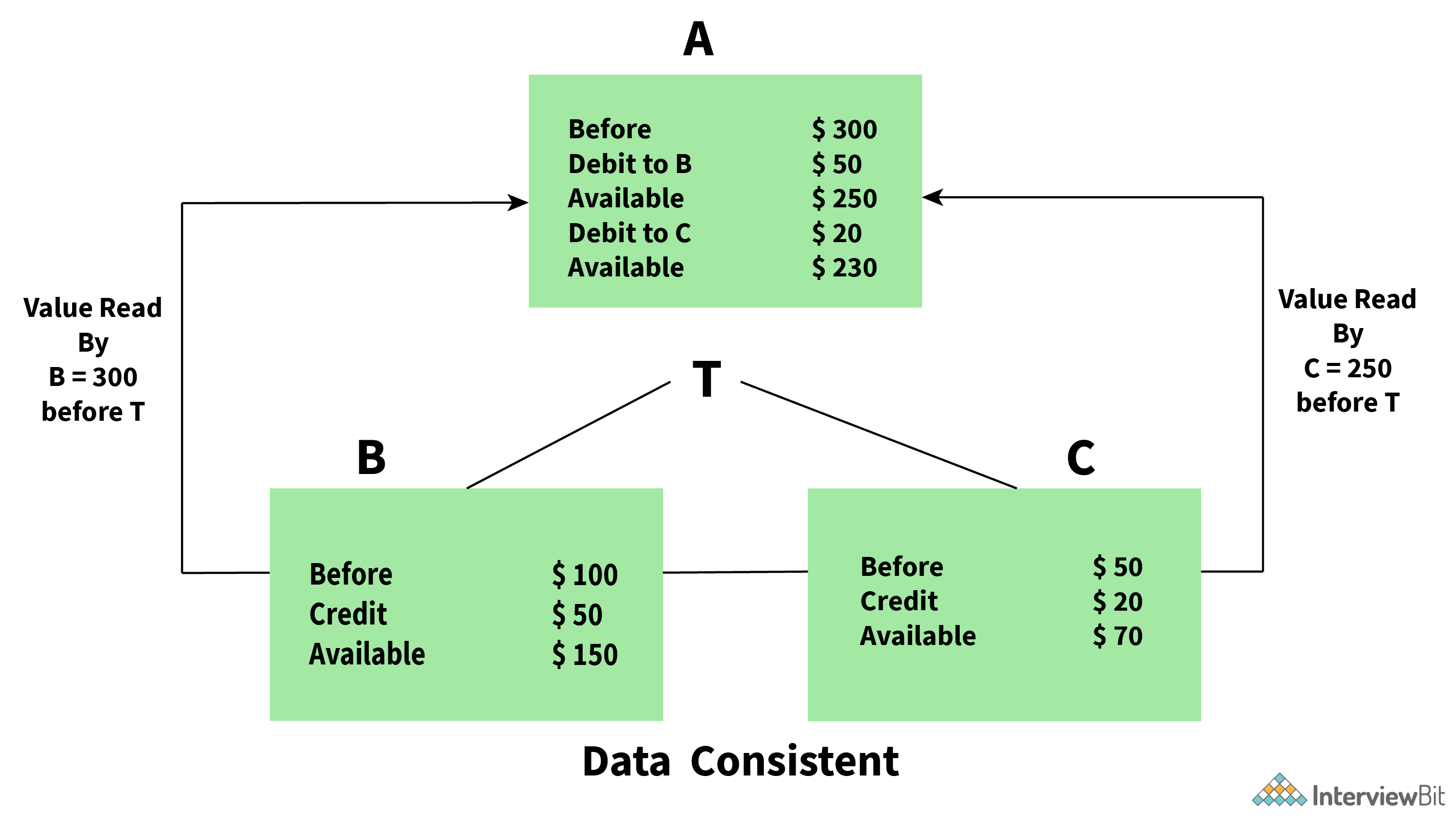
ACID stands for Atomicity, Consistency, Isolation, and Durability in a DBMS these are those properties that ensure a safe and secure way of sharing data among multiple users.



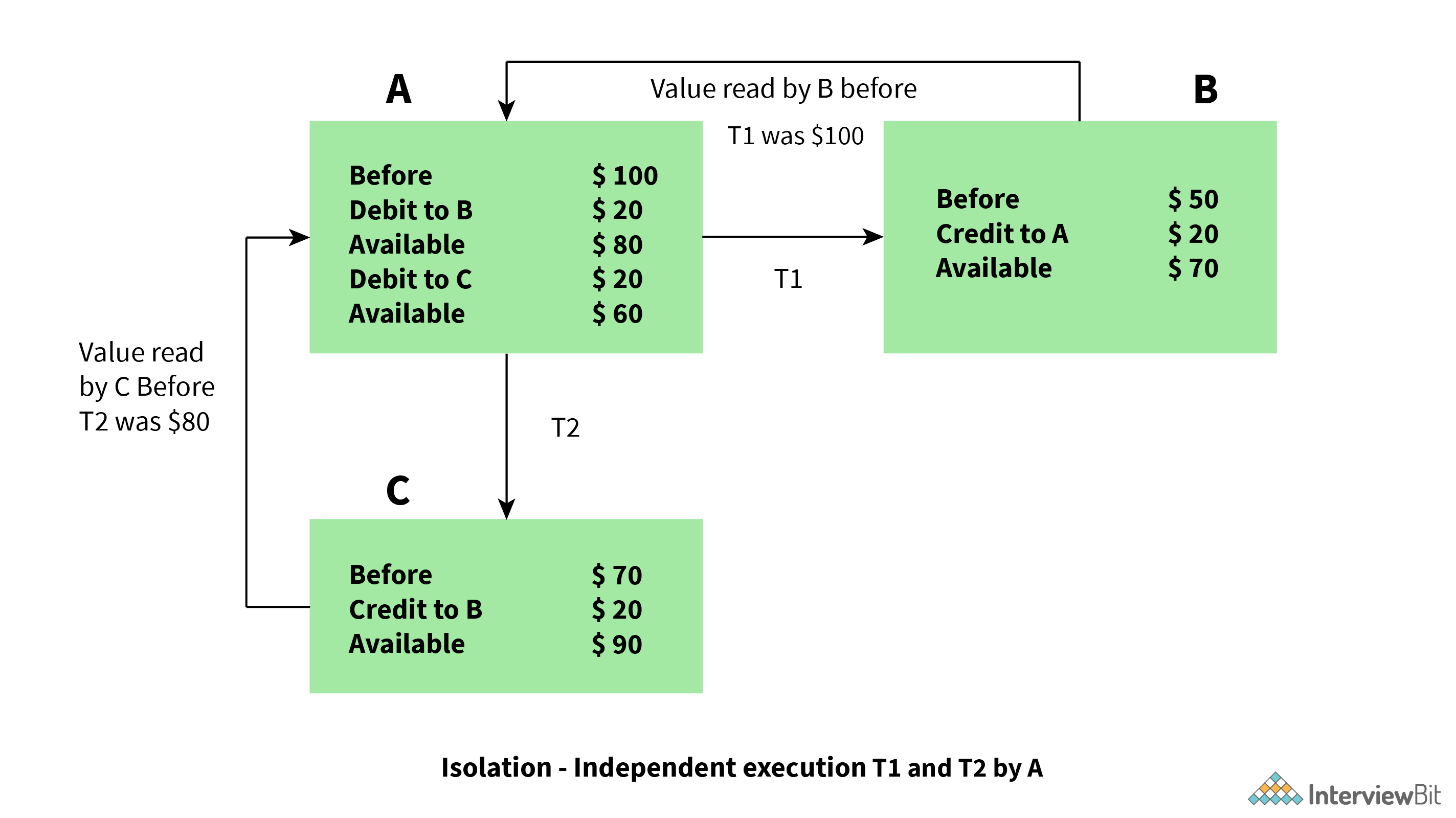
* **Atomicity**: This property reflects the concept of either executing the whole query or executing nothing at all, which implies that if an update occurs in a database then that update should either be reflected in the whole database or should not be reflected at all.



* **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.



* **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?**

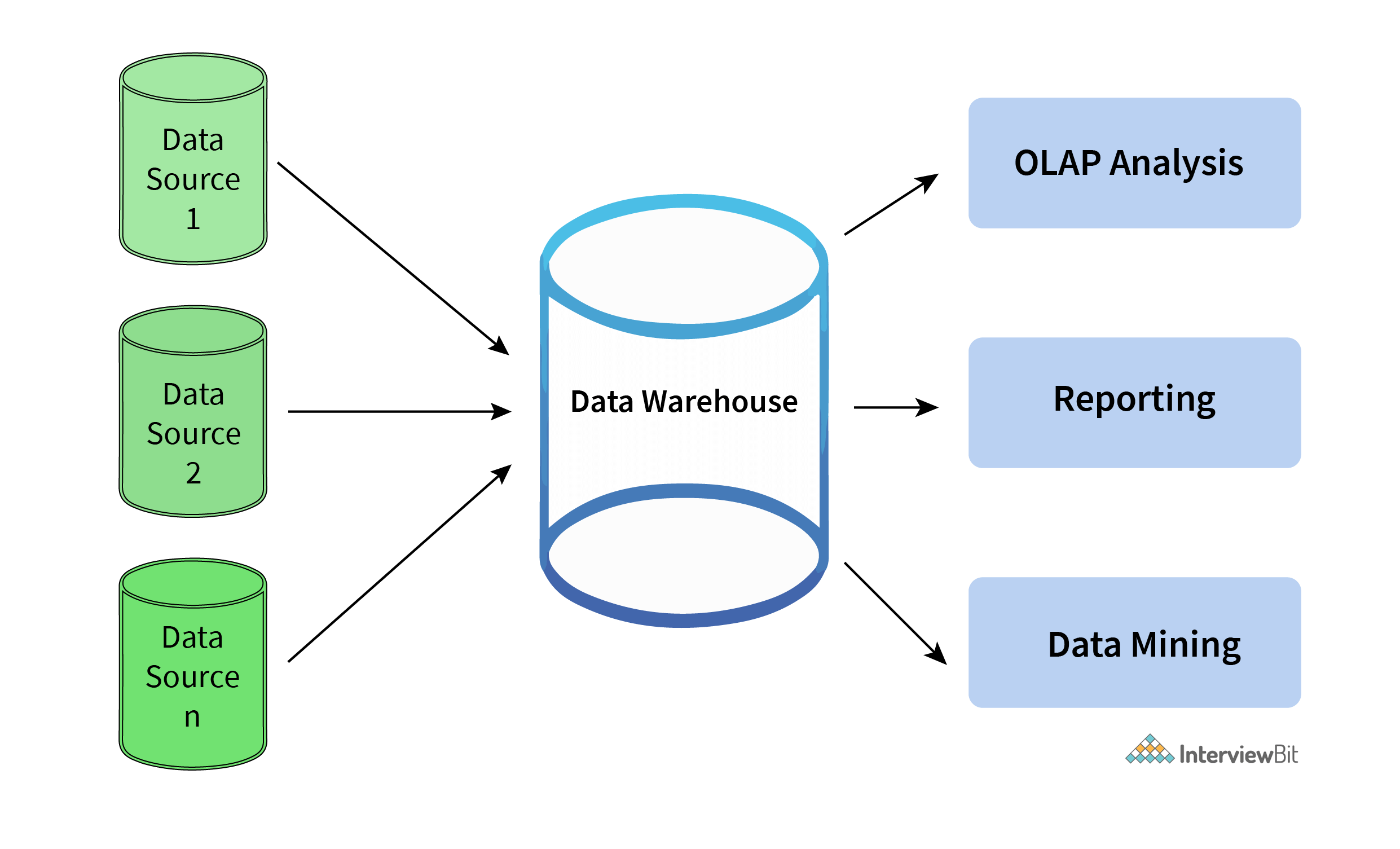
No, a NULL value is very different from that of zero and blank space as it represents a value that is assigned, unknown, unavailable, or not applicable as compared to blank space which represents a character and zero represents a number.

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.

**Intermediate DBMS Interview Questions**

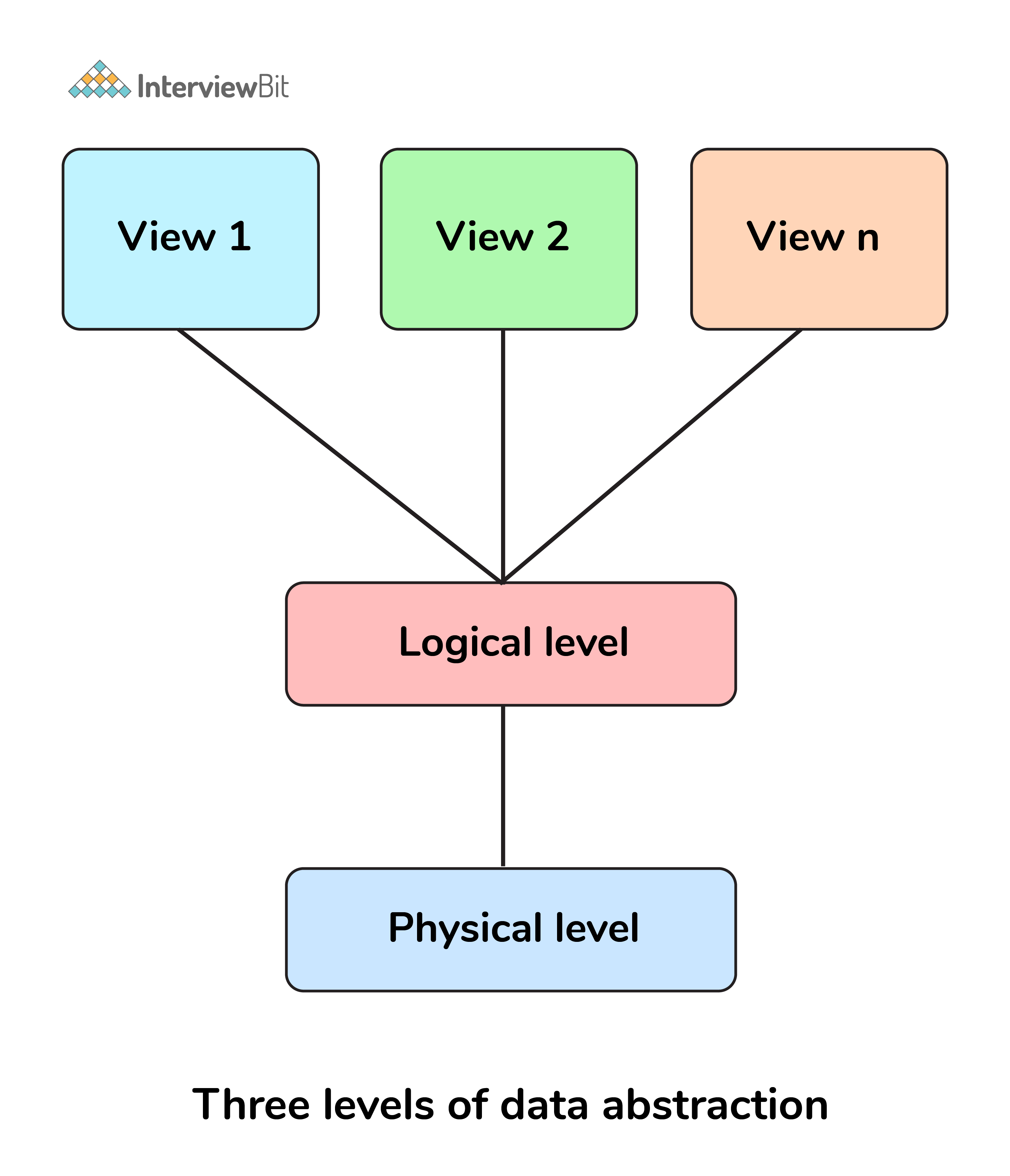
**8. What is Data Warehousing?**

The process of collecting, extracting, transforming, and loading data from multiple sources and storing them in one database is known as [**data warehousing**](https://www.interviewbit.com/data-warehouse-interview-questions/). A data warehouse can be considered as a central repository where data flows from transactional systems and other relational databases and is used for data analytics. A data warehouse comprises a wide variety of an organization’s historical data that supports the decision-making process in an organization.



**9. Explain different levels of data abstraction in a DBMS.**

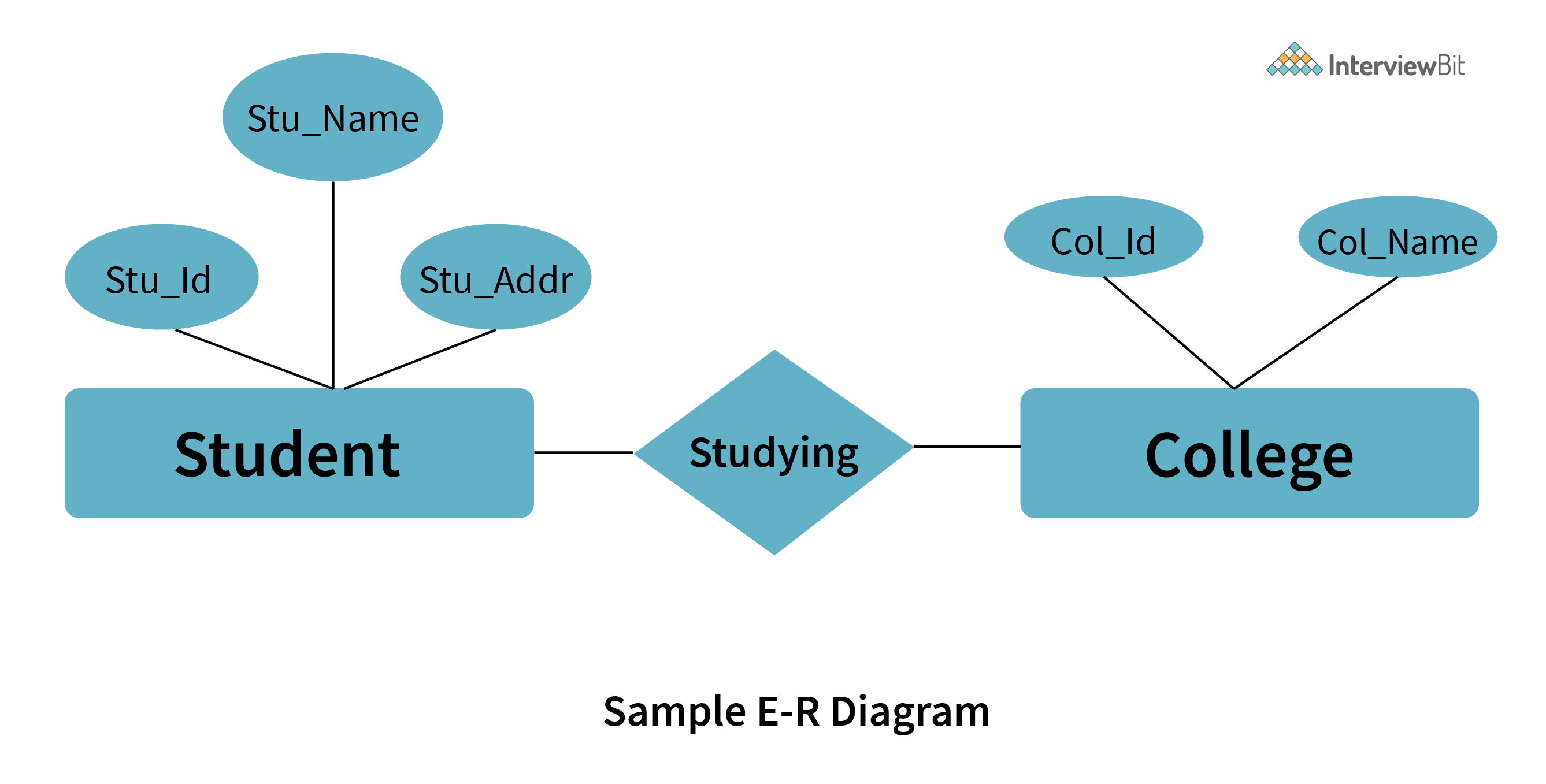
The process of hiding irrelevant details from users is known as data abstraction. Data abstraction can be divided into 3 levels:



* **Physical Level:**  it is the lowest level and is managed by DBMS. This level consists of data storage descriptions and the details of this level are typically hidden from system admins, developers, and users.
* **Conceptual or Logical level:**  it is the level on which developers and system admins work and it determines what data is stored in the database and what is the relationship between the data points.
* **External or View level:** it is the level that describes only part of the database and hides the details of the table schema and its physical storage from the users. The result of a query is an example of View level data abstraction.  A view is a virtual table created by selecting fields from one or more tables present in the database.

**10. What is meant by an entity-relationship (E-R) model? Explain the terms Entity, Entity Type, and Entity Set in DBMS.**

An [**entity-relationship model**](https://www.interviewbit.com/blog/er-model-in-dbms/) is a diagrammatic approach to a database design where real-world objects are represented as entities and relationships between them are mentioned.

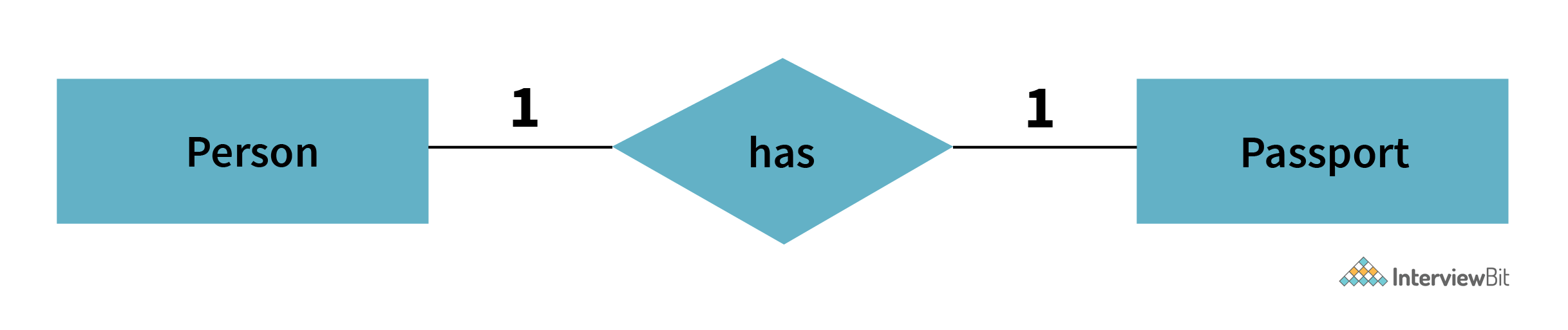


* **Entity:** An entity is defined as a real-world object having attributes that represent characteristics of that particular object. For example, a student, an employee, or a teacher represents an entity.
* **Entity Type:** An entity type is defined as a collection of entities that have the same attributes. One or more related tables in a database represent an entity type. Entity type or attributes can be understood as a characteristic which uniquely identifies the entity.  For example, a student represents an entity that has attributes such as student\_id, student\_name, etc.
* **Entity Set:** An entity set can be defined as a set of all the entities present in a specific entity type in a database. For example, a set of all the students, employees, teachers, etc. represent an entity set.

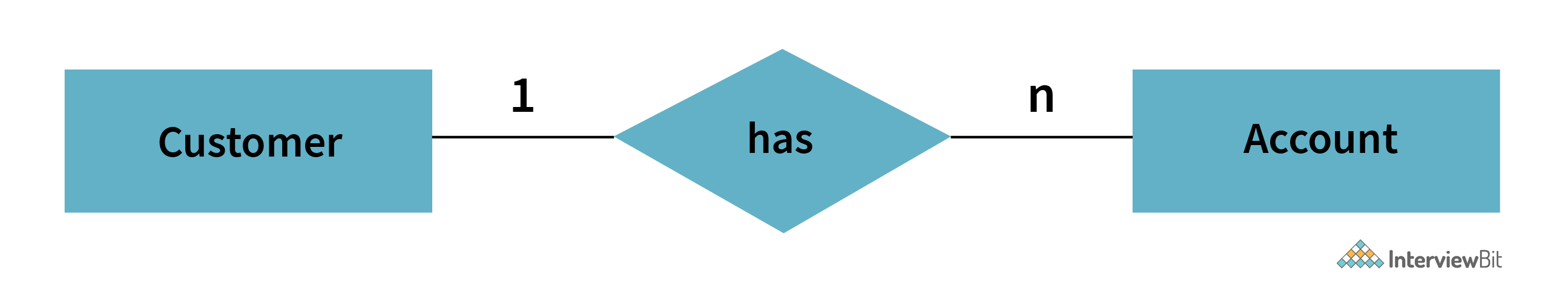
**11. Explain different types of relationships amongst tables in a DBMS.**

Following are different types of relationship amongst tables in a DBMS system:

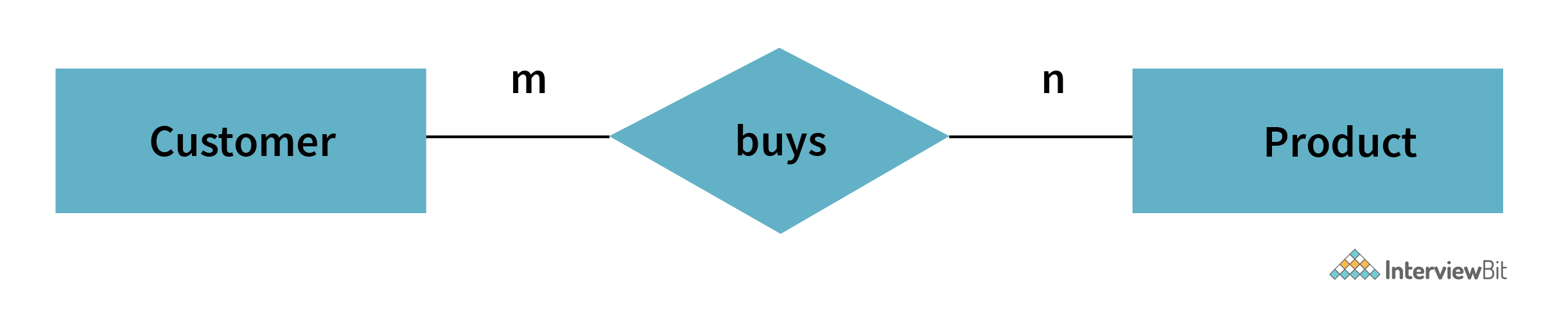
* **One to One Relationship:** This type of relationship is applied when a particular row in table X is linked to a singular row in table Y.



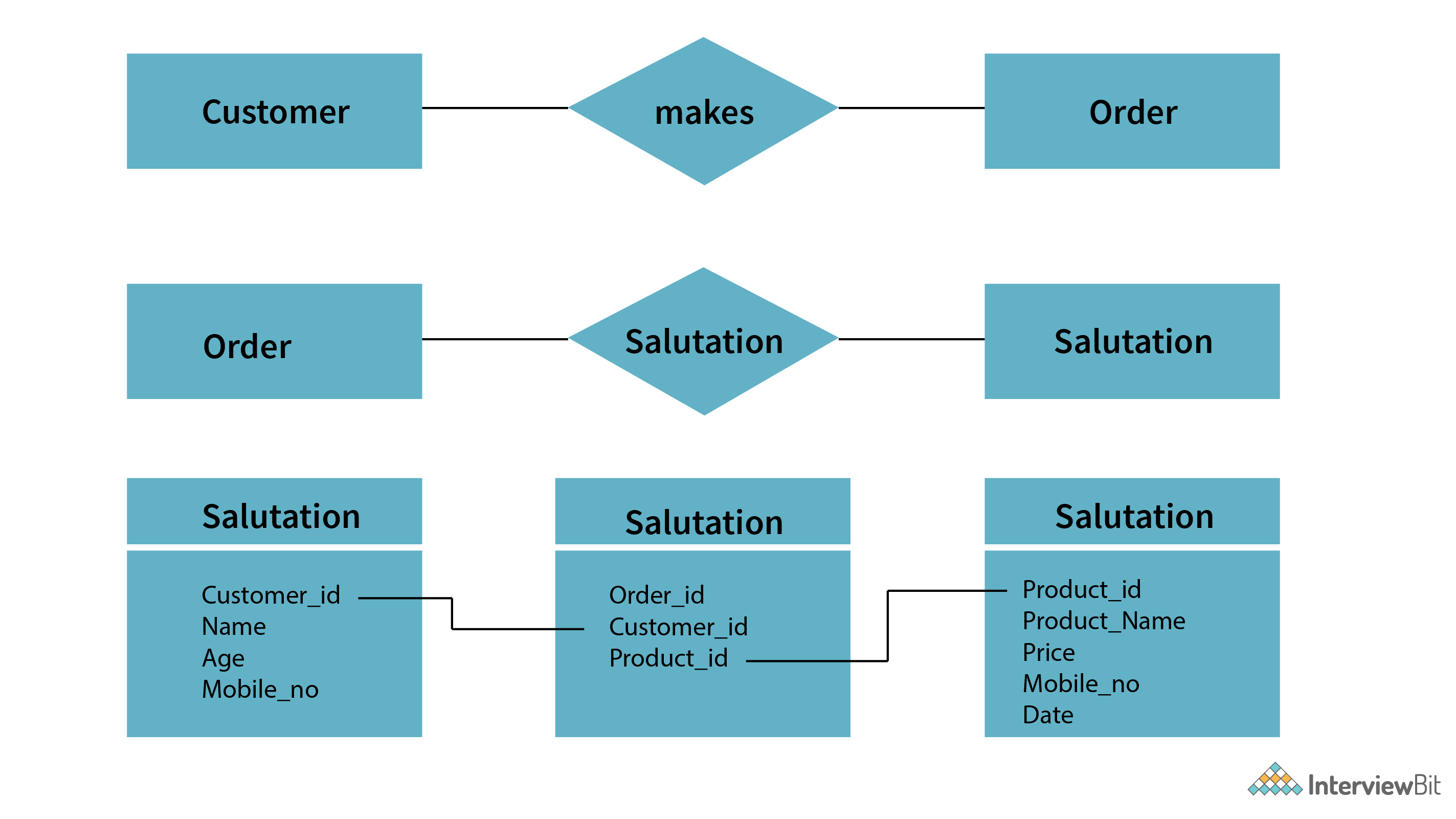
* **One to Many Relationship:** This type of relationship is applied when a single row in table X is related to many rows in table Y.



* **Many to Many Relationship:**This type of relationship is applied when multiple rows in table X can be linked to multiple rows in table Y.



* **Self Referencing Relationship:**This type of relationship is applied when a particular row in table X is associated with the same table.



**12. Explain the difference between intension and extension in a database.**

Following is the major difference between intension and extension in a database:

* **Intension:** Intension or popularly known as database schema is used to define the description of the database and is specified during the design of the database and mostly remains unchanged.
* **Extension:** Extension on the other hand is the measure of the number of tuples present in the database at any given point in time. The extension of a database is also referred to as the snapshot of the database and its value keeps changing as and when the tuples are created, updated, or destroyed in a database.

**13. Explain the difference between the DELETE and TRUNCATE command in a DBMS.**

**DELETE command:**this command is needed to delete rows from a table based on the condition provided by the WHERE clause.

* It deletes only the rows which are specified by the WHERE clause.
* It can be rolled back if required.
* It maintains a log to lock the row of the table before deleting it and hence it’s slow.

**TRUNCATE command:** this command is needed to remove complete data from a table in a database. It is like a DELETE command which has no WHERE clause.

* It removes complete data from a table in a database.
* It can't be rolled back even if required. ( truncate can be rolled back in some databases depending on their version but it can be tricky and can lead to data loss). Check this [link](https://codingsight.com/rollback-truncate-in-sql-server/) for more details
* It doesn’t maintain a log and deletes the whole table at once and hence it’s fast.

**14. What is a lock. Explain the major difference between a shared lock and an exclusive lock during a transaction in a database.**

A database lock is a mechanism to protect a shared piece of data from getting updated by two or more database users at the same time. When a single database user or session has acquired a lock then no other database user or session can modify that data until the lock is released.

* **Shared Lock:**A shared lock is required for reading a data item and many transactions may hold a lock on the same data item in a shared lock. Multiple transactions are allowed to read the data items in a shared lock.
* **Exclusive lock:** An exclusive lock is a lock on any transaction that is about to perform a write operation. This type of lock doesn’t allow more than one transaction and hence prevents any inconsistency in the database.

**15. What is meant by normalization and denormalization?**

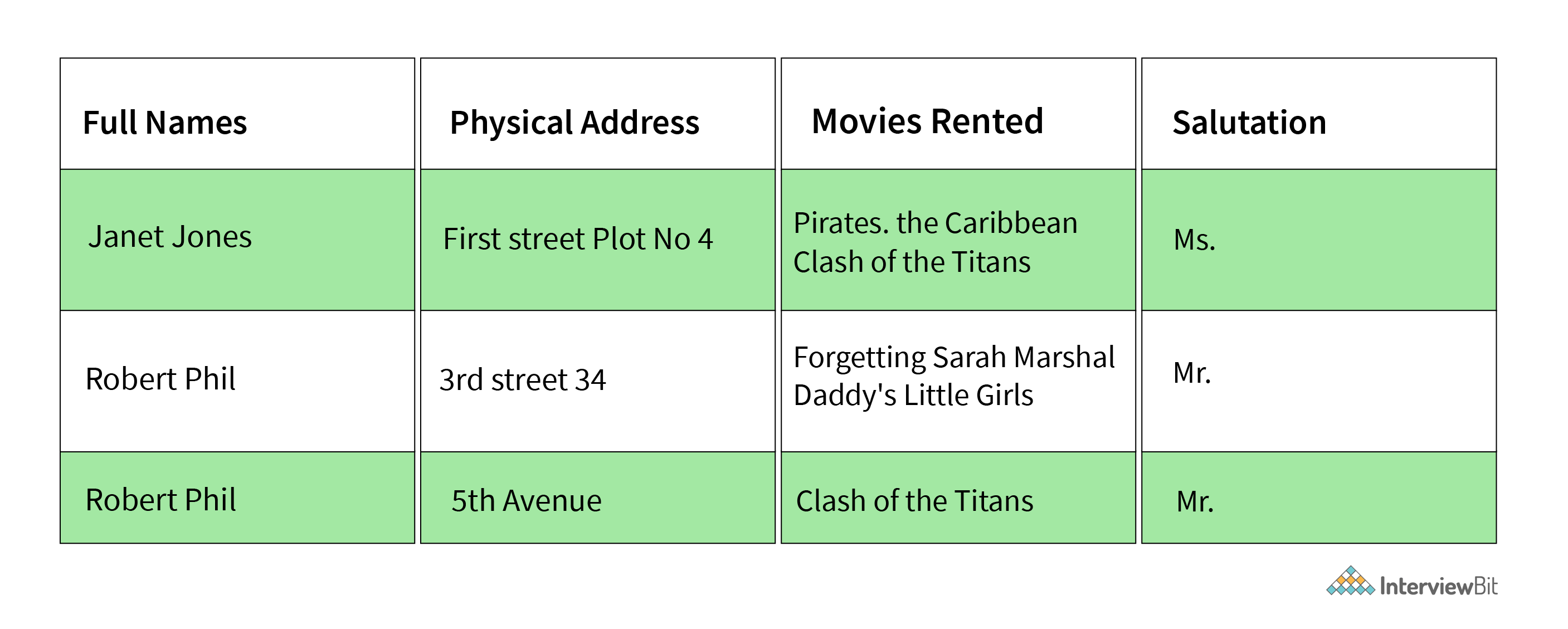
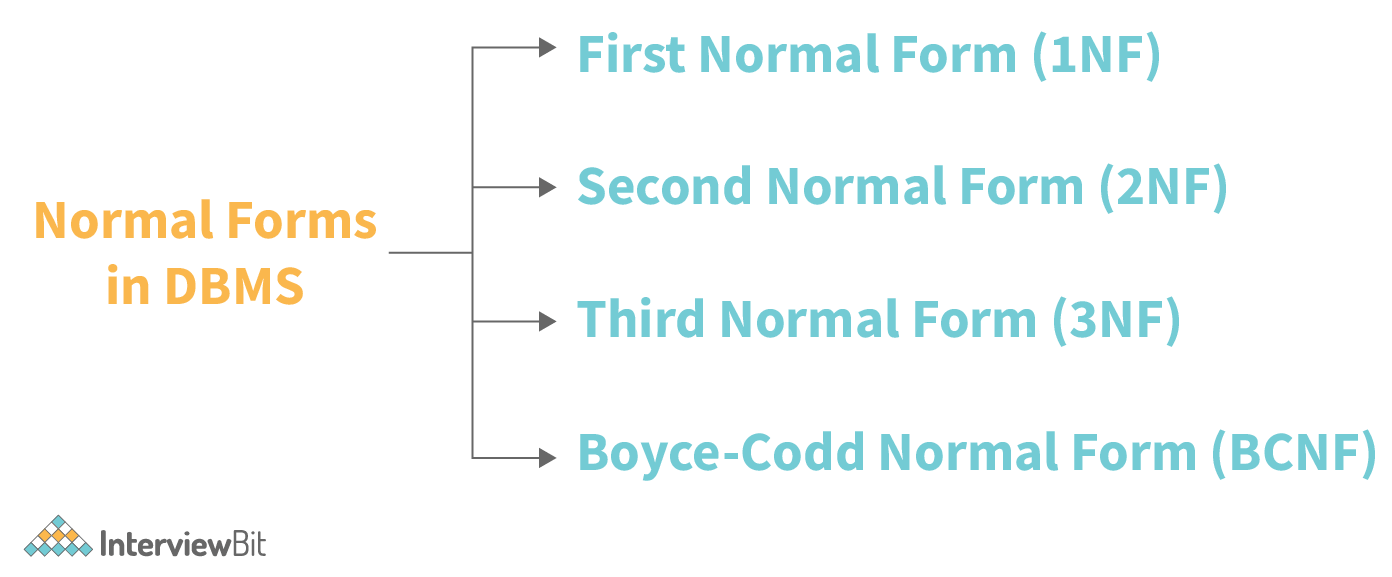
[**Normalization**](https://www.scaler.com/topics/normalization-in-dbms/) is a process of reducing redundancy by organizing the data into multiple tables. Normalization leads to better usage of disk spaces and makes it easier to maintain the integrity of the database.

**Denormalization** is the reverse process of normalization as it combines the tables which have been normalized into a single table so that data retrieval becomes faster. JOIN operation allows us to create a denormalized form of the data by reversing the normalization.

**Advanced DBMS Interview Questions**

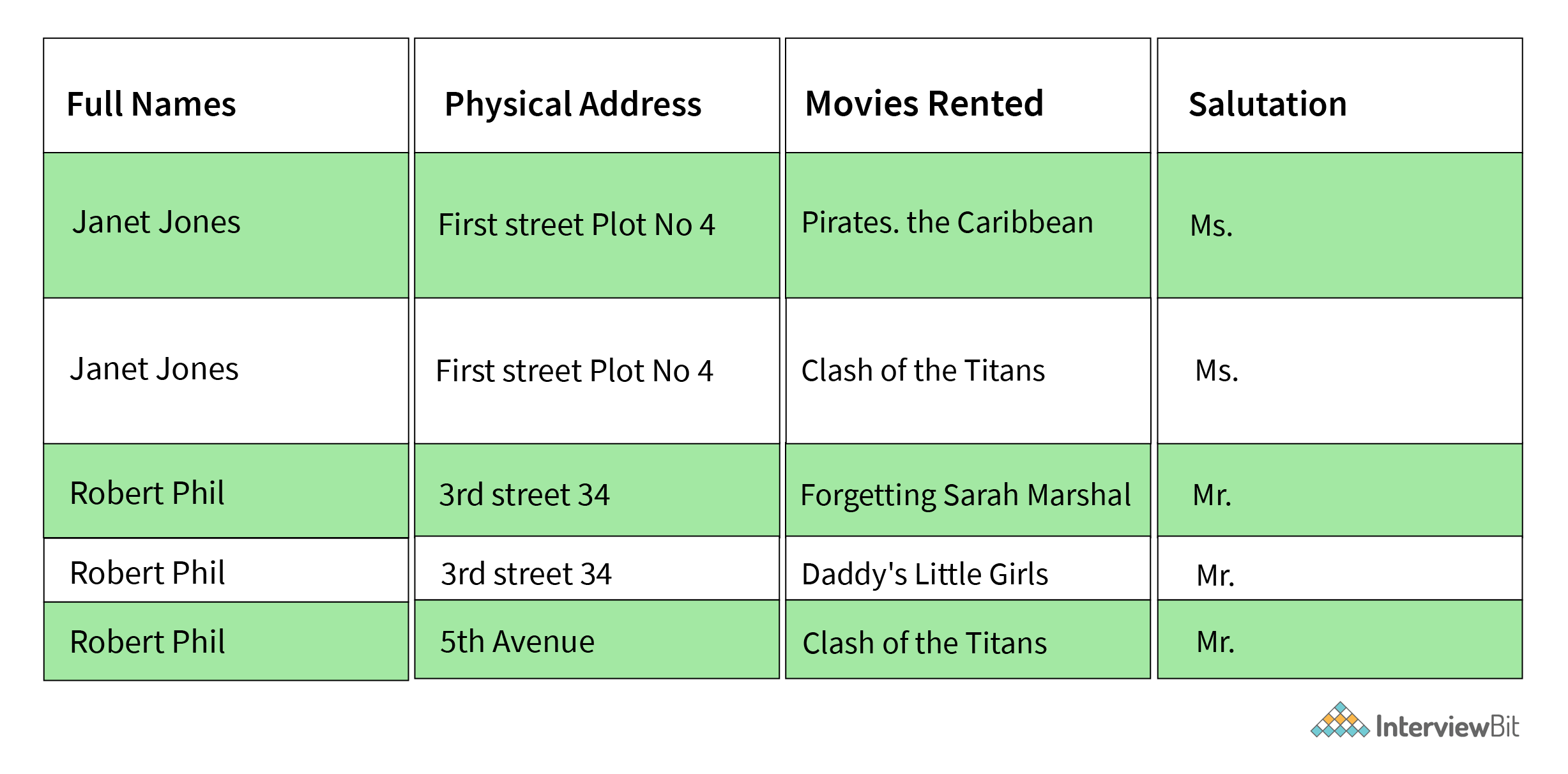
**16. Explain different types of Normalization forms in a DBMS.**

Following are the major normalization forms in a DBMS:



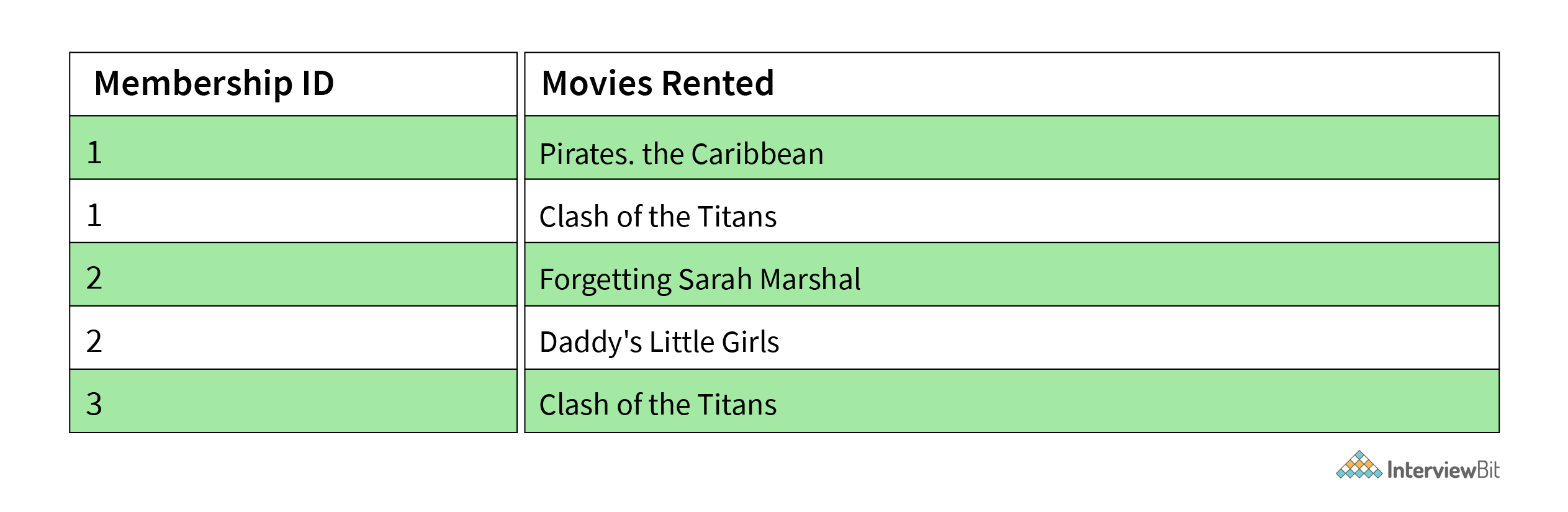
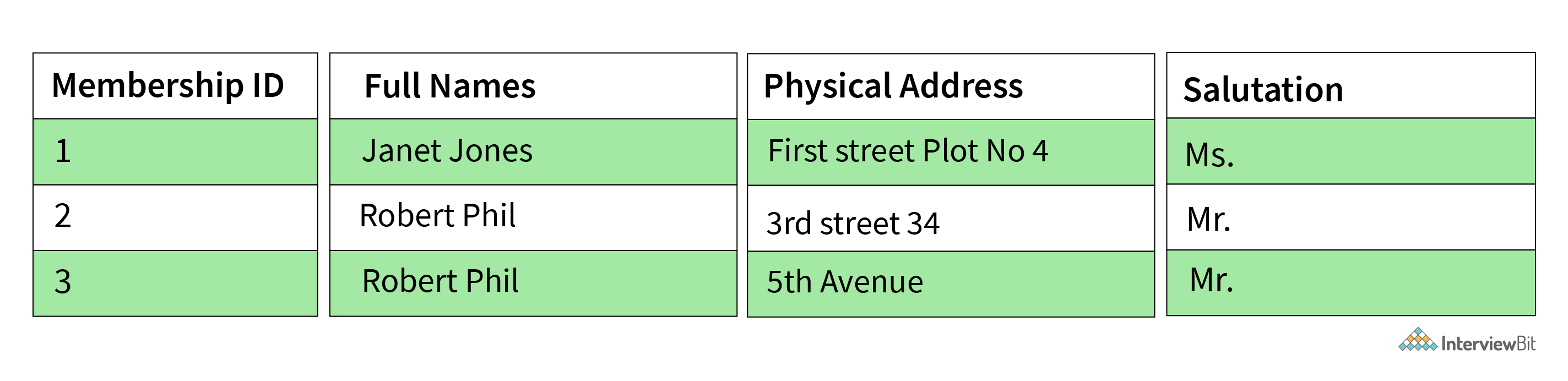
**Considering the above Table-1 as the reference example for understanding different normalization forms.**

* 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.



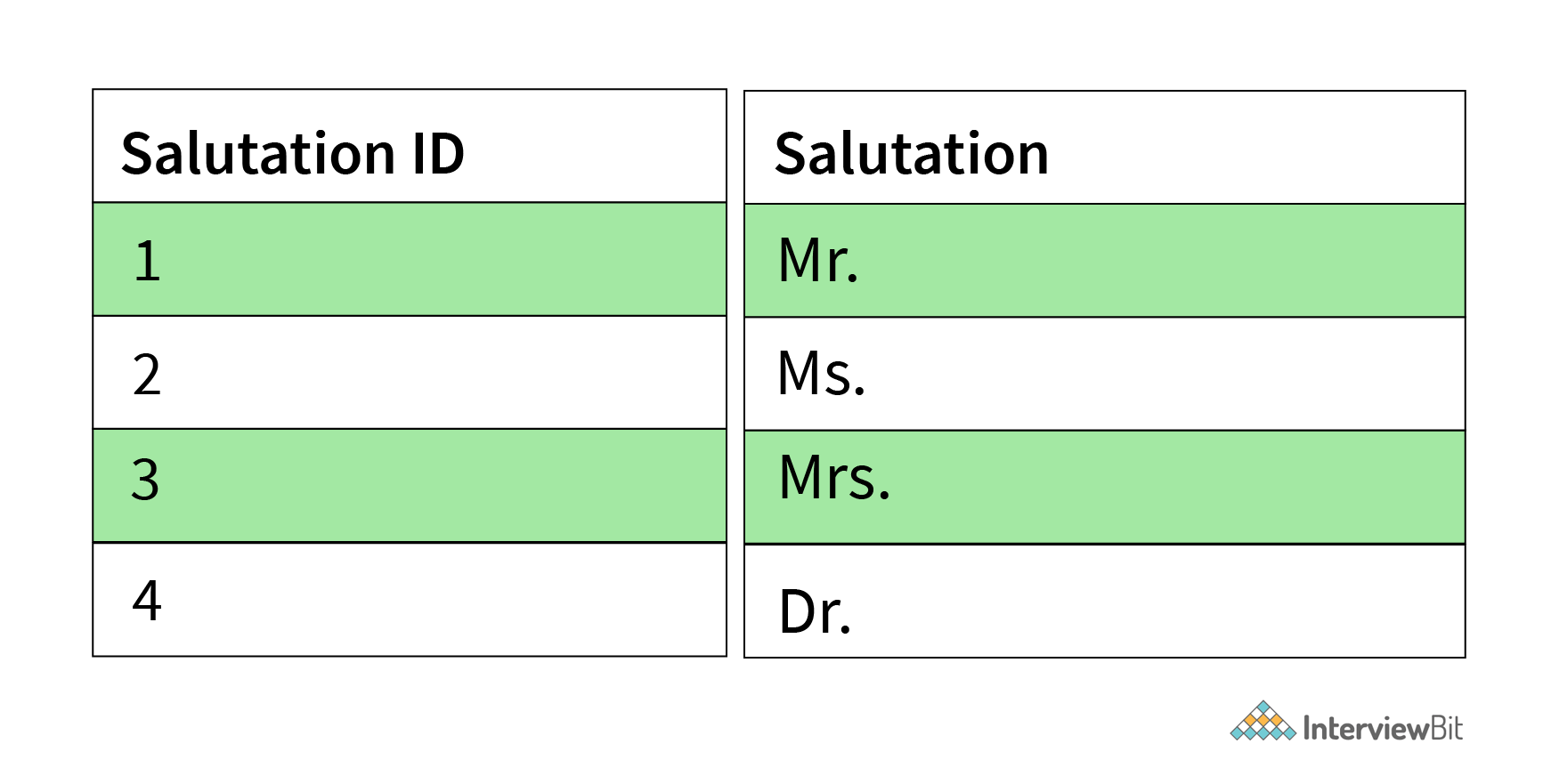
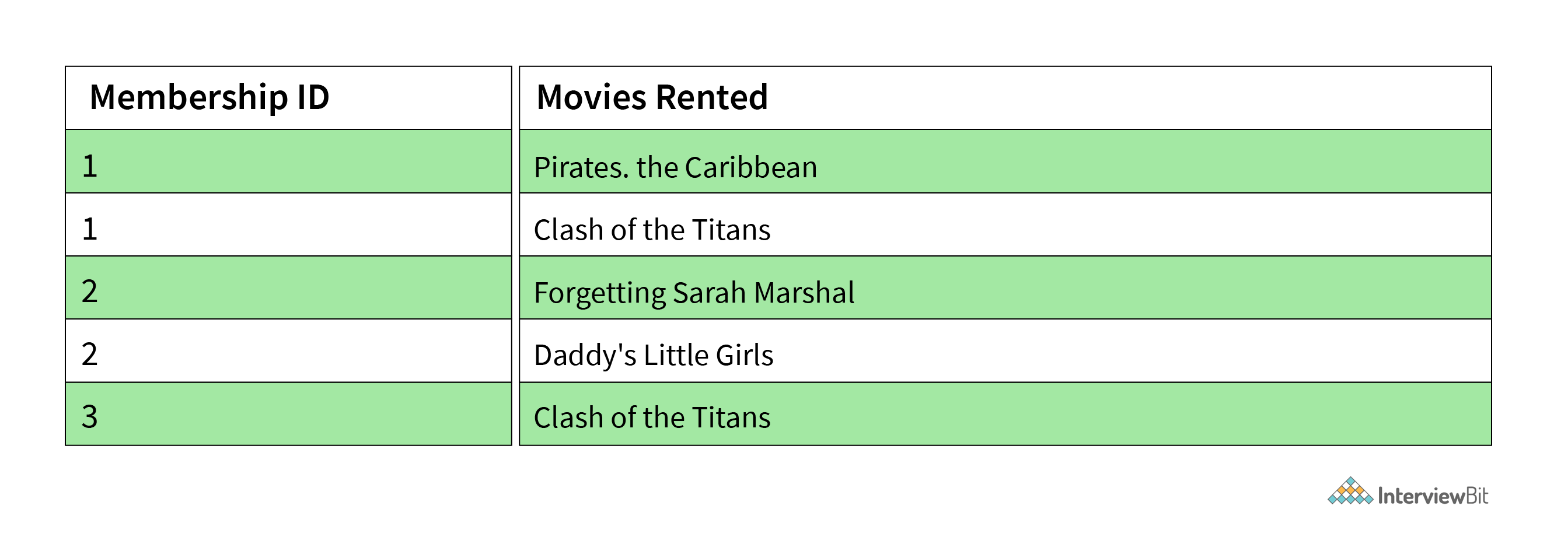
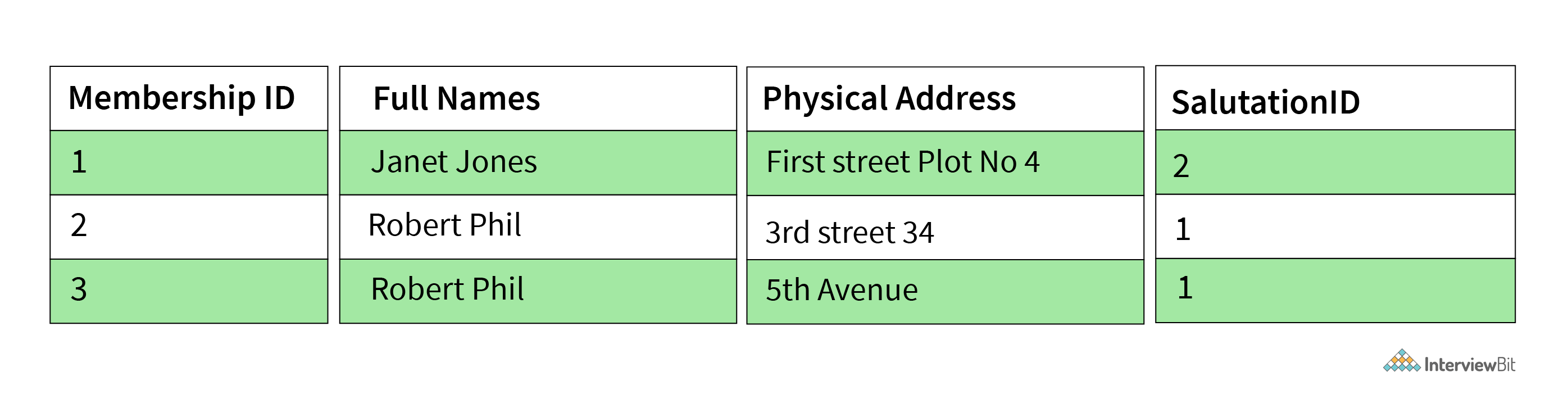
***Table-1 converted to 1NF form***

* **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.



***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 third 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.



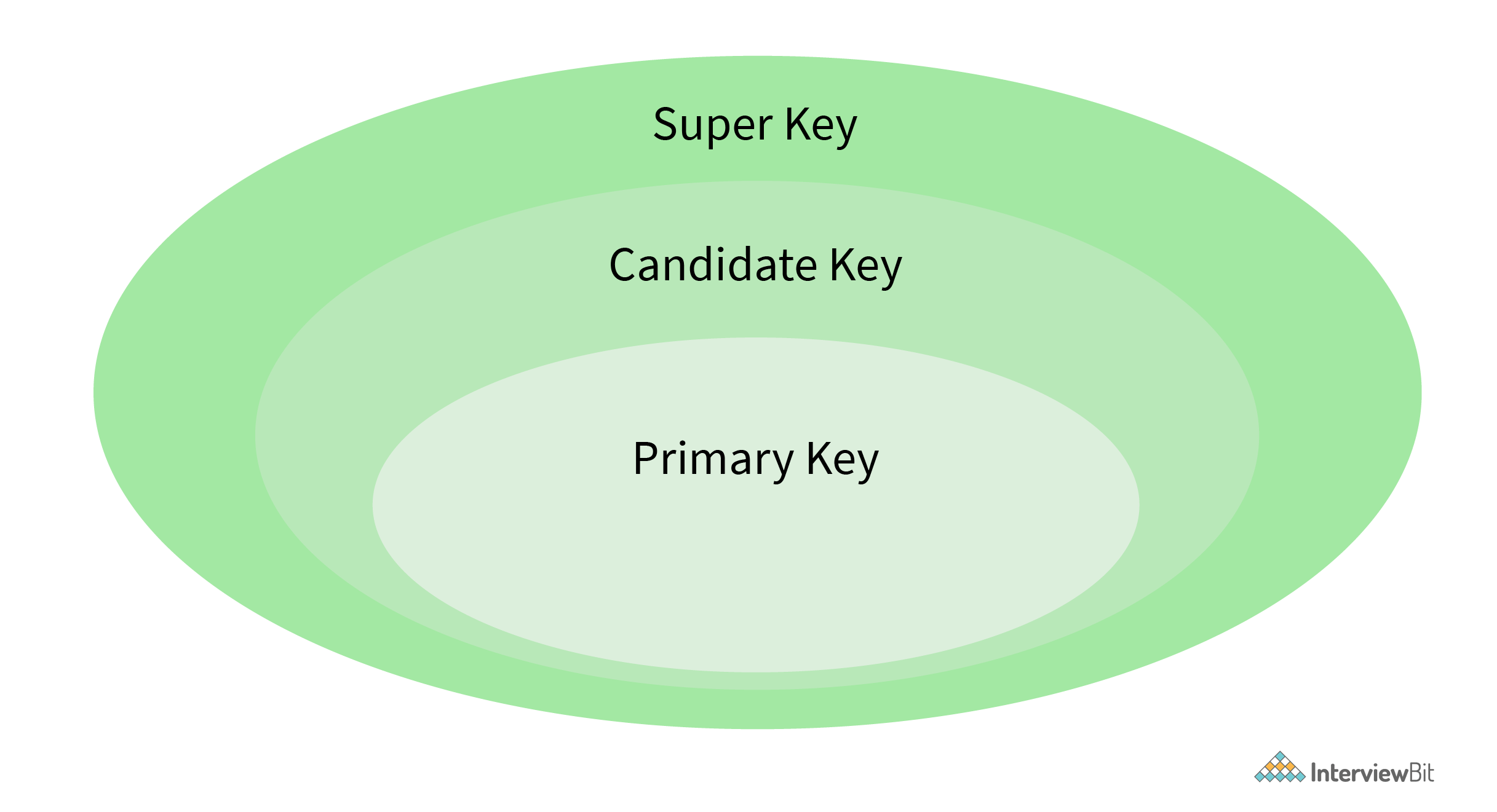
***Breaking Table-1 into 3 different tables to move it to 3NF.***

* **BCNF:** BCNF stands for [Boyce-Codd Normal Form](https://www.scaler.com/topics/bcnf-in-dbms/) 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.

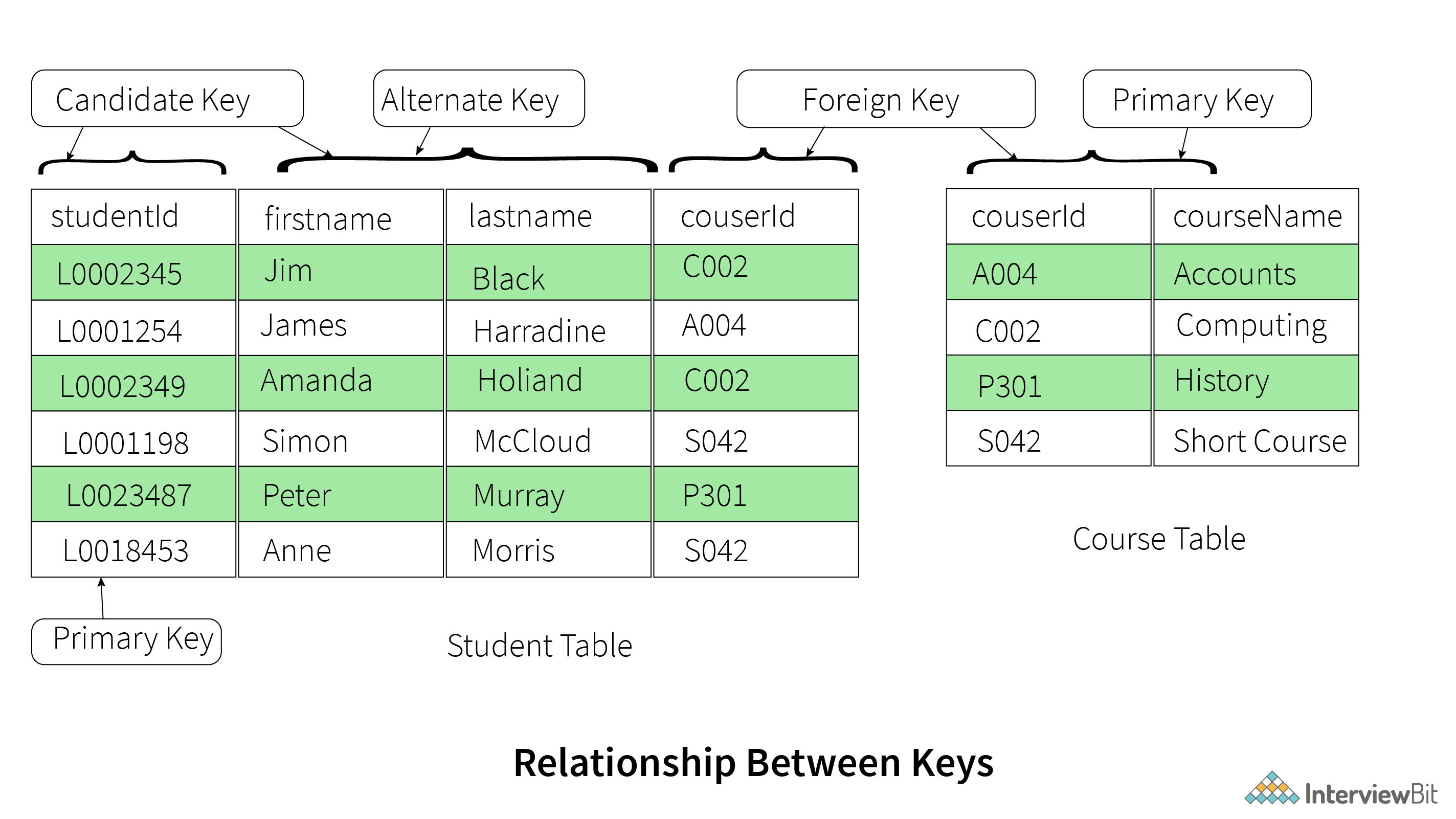
**17. Explain different types of keys in a database.**

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.

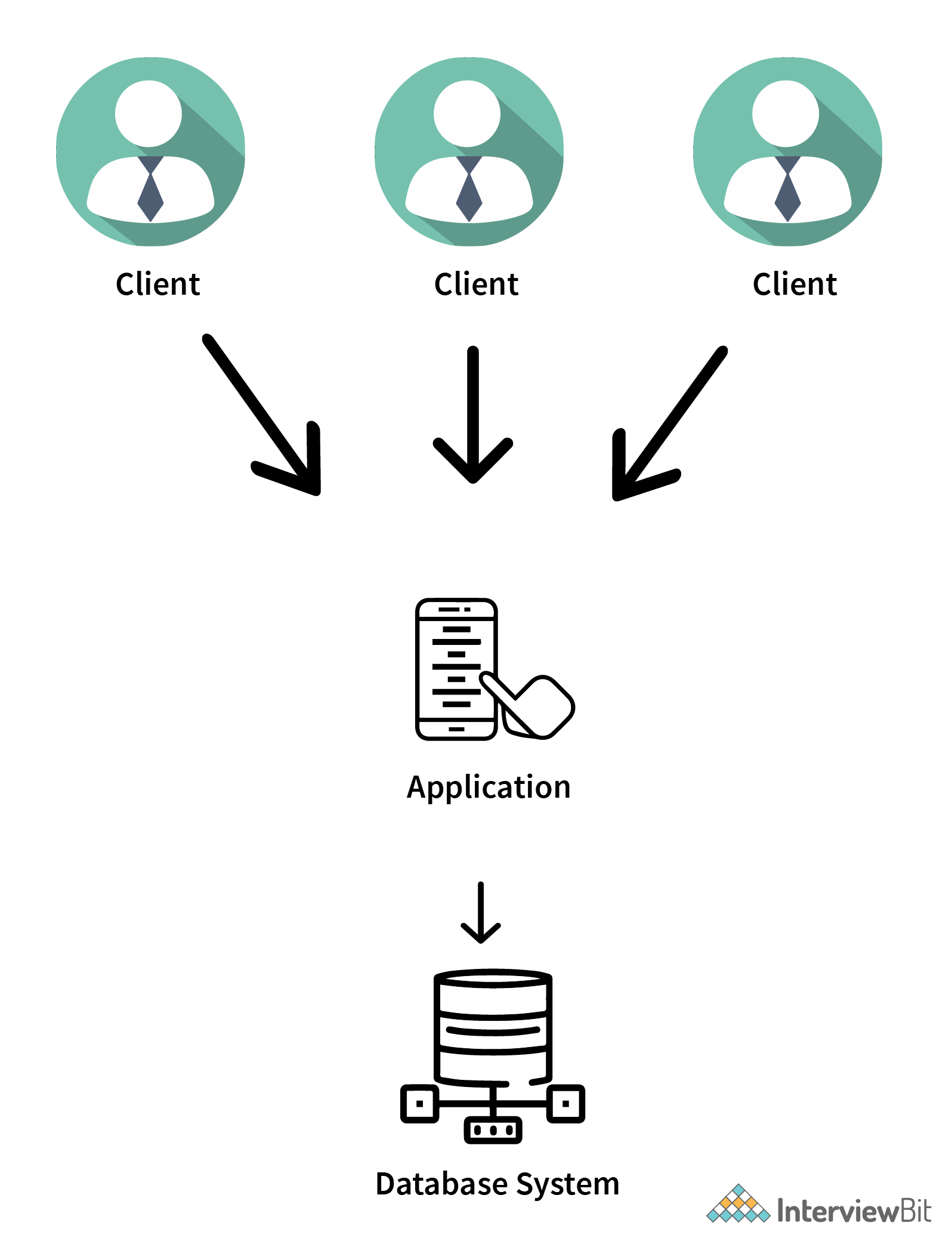


* **Primary Key:** The primary key defines a set of attributes that are used to uniquely identify every tuple. In the below example studentId and firstName are candidate keys and any one of them can be chosen as a Primary Key. In the given example studentId is chosen as the primary key for the student 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.



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

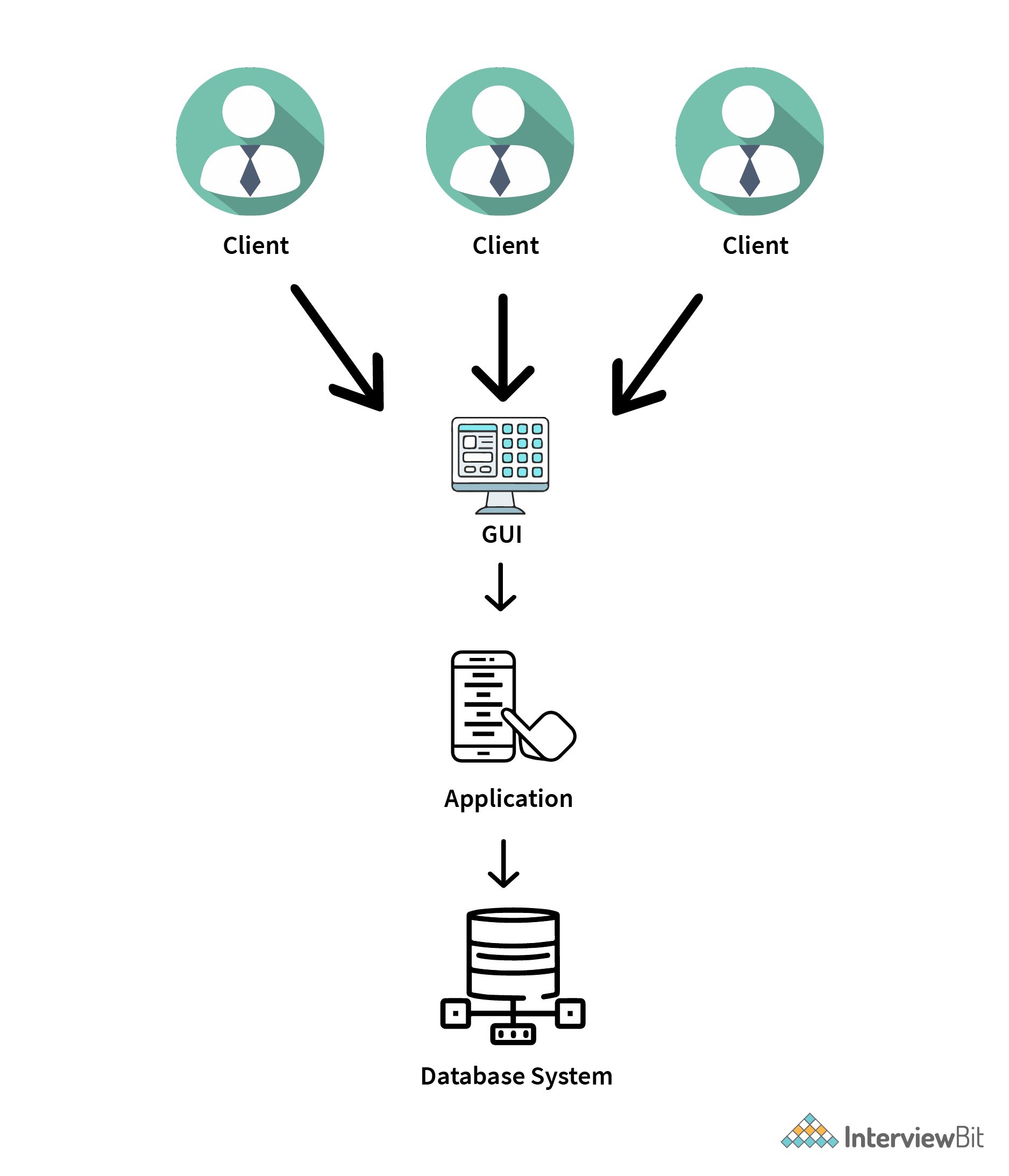
The **2-tier architecture** refers to the client-server architecture in which applications at the client end directly communicate with the database at the server end without any middleware involved.  
**Example** – Contact Management System created using MS-Access or Railway Reservation System, etc.



***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:***

* [DBMS MCQ with Answers](https://www.interviewbit.com/dbms-mcq/)
* [Top Applications of DBMS](https://www.interviewbit.com/blog/applications-of-dbms/)
* [SQL Interview Questions](https://www.interviewbit.com/sql-interview-questions/)
* [SQL Server Interview Questions](https://www.interviewbit.com/sql-server-interview-questions/)
* [MySQL Interview Questions](https://www.interviewbit.com/mysql-interview-questions/)
* [MongoDB Interview Questions](https://www.interviewbit.com/mongodb-interview-questions/)
* [PL SQL Interview Questions](https://www.interviewbit.com/pl-sql-interview-questions/)
* [Database Testing Interview Questions](https://www.interviewbit.com/database-testing-interview-questions/)
* [File System vs DBMS](https://www.interviewbit.com/blog/file-system-vs-dbms/)
* [Interview Preparation Guide](https://www.interviewbit.com/technical-interview-questions/)

DBMS MCQ

1.

An entity set is represented by what in an E-R diagram?

Circle

Rectangle

Ellipse

Diamond Box

2.

A record in a relational DBMS is also known as:

Attribute

Tuple

Relation

Criteria

3.

What does ODBC in a database stand for:

Oral Database Connectivity

Oracle Database Connectivity

Open Database Connectivity

Object Database Connectivity

4.

Which of the following represents a one-to-many relationship?

Many teachers have one class

Many teachers have many classes

One teacher has many classes

One teacher has one class

5.

Which of the following contains information about the data stored in a DBMS?

Directory

Sub Data

Warehouse

Meta Data

6.

Which of the following does DBA stand for?

Data Administrator

Database Administrator

Data Bank Administrator

None of the above

7.

 In a DBMS, TCL means:

Ternary Control Language

Transmission Control Language

Transaction Central Language

Transaction Control Language

**1. What are the advantages of DBMS over traditional file-based systems?**

Database management systems were developed to handle the following difficulties of typical File-processing systems supported by conventional operating systems.   
1. Data redundancy and inconsistency   
2. Difficulty in accessing data   
3. Data isolation – multiple files and formats   
4. Integrity problems   
5. Atomicity of updates   
6. Concurrent access by multiple users   
7. Security problems 

**2. What are super, primary, candidate**,**and foreign keys?**   
A [super key](https://www.geeksforgeeks.org/difference-between-super-key-and-candidate-key/) is a set of attributes of a relation schema upon which all attributes of the schema are functionally dependent. No two rows can have the same value of super key attributes.   
A [Candidate key](https://www.geeksforgeeks.org/difference-between-super-key-and-candidate-key/) is a minimal superkey, i.e., no proper subset of Candidate key attributes can be a superkey.   
A [Primary Key](https://www.geeksforgeeks.org/difference-between-primary-and-candidate-key/) is one of the candidate keys. One of the candidate keys is selected as most important and becomes the primary key. There cannot be more than one primary key in a table..  
A **Foreign key** is a field (or collection of fields) in one table that uniquely identifies a row of another table.

**3. What is the difference between primary key and unique constraints?**   
The primary key cannot have NULL value, the unique constraints can have NULL values. There is only one primary key in a table, but there can be multiple unique constrains.

**4.What is database normalization?**   
It is a process of analyzing the given relation schemas based on their functional dependencies and primary keys to achieve the following desirable properties:   
1. Minimizing Redundancy   
2. Minimizing the Insertion, Deletion, And Update Anomalies Relation schemas that do not meet the properties are decomposed into smaller relation schemas that could meet desirable properties.

**5. Why is the use of DBMS recommended? Explain by listing some of its major advantages?**

Some of the major advantages of DBMS are as follows:

[](https://practice.geeksforgeeks.org/courses/complete-interview-preparation?utm_source=geeksforgeeks&utm_medium=inarticle_cip_computersubjects&utm_campaign=inbound_promotions)

* **Controlled Redundancy:** DBMS supports a mechanism to control the redundancy of data inside the database by integrating all the data into a single database and as data is stored in only one place, the duplicity of data does not happen.
* **Data Sharing:**Sharing of data among multiple users simultaneously can also be done in DBMS as the same database will be shared among all the users and by different application programs.
* **Backup and Recovery Facility:**DBMS minimizes the pain of creating the backup of data again and again by providing a feature of ‘backup and recovery’ which automatically creates the data backup and restores the data whenever required.
* **Enforcement of Integrity Constraints:**Integrity Constraints are very important to be enforced on the data so that the refined data after putting some constraints are stored in the database and this is followed by DBMS.
* **Independence of Data:** It simply means that you can change the structure of the data without affecting the structure of any of the application programs.

**6. What are the differences between DDL, DML**,**and DCL in SQL?**  
Following are some details of three :  
**DDL** stands for Data Definition Language. SQL queries like CREATE, ALTER, DROP, TRUNCATE and RENAME come under this.   
**DML** stands for Data Manipulation Language. SQL queries like SELECT, INSERT, DELETE and UPDATE come under this.   
**DCL** stands for Data Control Language. SQL queries like GRANT and REVOKE come under this.

**7. What is the difference between having and where clause?**   
HAVING is used to specify a condition for a group or an aggregate function used in a select statement. The WHERE clause selects before grouping. The HAVING clause selects rows after grouping. Unlike the HAVING clause, the WHERE clause cannot contain aggregate functions. (See [this](http://newtonapples.com/difference-clause-clause/)for examples). See [Having vs Where Clause?](https://www.geeksforgeeks.org/having-vs-where-clause-in-sql/) for more details

**8.How to print duplicate rows in a table?**   
See <https://www.geeksforgeeks.org/how-to-print-duplicate-rows-in-a-table/>

**9. What is Join?**  
An SQL Join is used to combine data from two or more tables, based on a common field between them. For example, consider the following two tables.

**Table –** Student Table 

|  |  |  |
| --- | --- | --- |
| **EnrollNo** | **StudentName** | **Address** |
| **1000** | **geek1** | **geeksquiz1** |
| **1001** | **geek2** | **geeksquiz2** |
| **1002** | **geek3** | **geeksquiz3** |

**Table –** StudentCourse Table

|  |  |
| --- | --- |
| **CourseID** | **EnrollNo** |
| **1** | **1000** |
| **2** | **1000** |
| **3** | **1000** |
| **1** | **1002** |
| **2** | **1003** |

Following is a join query that shows the names of students enrolled in different courseIDs.

SELECT StudentCourse.CourseID, Student.StudentName

FROM StudentCourse

INNER JOIN Student

ON StudentCourse.EnrollNo = Student.EnrollNo

ORDER BY StudentCourse.CourseID;

The above query would produce the following result.

|  |  |
| --- | --- |
| **CourseID** | **StudentName** |
| **1** | **geek1** |
| **1** | **geek3** |
| **2** | **geek1** |
| **3** | **geek1** |

**9. What is Identity?**   
Identity (or AutoNumber) is a column that automatically generates numeric values. A start and increment value can be set, but most DBA leave these at 1. A GUID column also generates numbers; the value of this cannot be controlled. Identity/GUID columns do not need to be indexed.

**10.What is a view in SQL? How to create a view?**  
 A [view](http://en.wikipedia.org/wiki/View_(SQL))is a virtual table based on the result-set of an SQL statement. We can create it using create view syntax. 

CREATE VIEW view\_name AS

SELECT column\_name(s)

FROM table\_name

WHERE condition

**11.What are the uses of view?**   
1. Views can represent a subset of the data contained in a table; consequently, a view can limit the degree of exposure of the underlying tables to the outer world: a given user may have permission to query the view, while denied access to the rest of the base table.   
2. Views can join and simplify multiple tables into a single virtual table.  
3. Views can act as aggregated tables, where the database engine aggregates data (sum, average, etc.) and presents the calculated results as part of the data.  
4. Views can hide the complexity of data.  
5. Views take very little space to store; the database contains only the definition of a view, not a copy of all the data which it presents.   
6. Depending on the SQL engine used, views can provide extra security.

**12. What is a Trigger?**   
A Trigger is a code associated with insert, update or delete operations. The code is executed automatically whenever the associated query is executed on a table. Triggers can be useful to maintain integrity in the database.

**13. What is a stored procedure?**   
A stored procedure is like a function that contains a set of operations compiled together. It contains a set of operations that are commonly used in an application to do some common database tasks.

**14. What is the difference between Trigger and Stored Procedure?**   
 Unlike Stored Procedures, Triggers cannot be called directly. They can only be associated with queries.

**15. What is a transaction? What are ACID properties?**   
A Database Transaction is a set of database operations that must be treated as a whole, which means either all operations are executed or none of them. An example can be a bank transaction from one account to another account. Either both debit and credit operations must be executed or none of them. ACID(Atomicity, Consistency, Isolation, Durability) is a set of properties that guarantee that database transactions are processed reliably.

**16. What are indexes?**  
A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of additional writes and the use of more storage space to maintain the extra copy of data. Data can be stored only in one order on a disk. To support faster access according to different values, faster search like binary search for different values is desired, For this purpose, indexes are created on tables. These indexes need extra space on the disk, but they allow faster search according to different frequently searched values.

**17. What are clustered and non-clustered Indexes?**   
Clustered indexes are the index according to which data is physically stored on a disk. Therefore, only one clustered index can be created on a given database table.   
Non-clustered indexes don’t define the physical ordering of data, but logical ordering. Typically, a tree is created whose leaf point to disk records. B-Tree or B+ tree are used for this purpose.

**18. What is Denormalization?**

Denormalization is a database optimization technique in which we add redundant data to one or more tables.

**19. What is CLAUSE in SQL?**

A clause in SQL is a part of a query that lets you filter or customize how you want your data to be queried to you.

**20. What is a Live Lock?**

Livelock situation can be defined as when two or more processes continually repeat the same interaction in response to changes in the other processes without doing any useful work These processes are not in the waiting state, and they are running concurrently. This is different from a deadlock because in a deadlock all processes are in the waiting state.

**21. What is QBE?**

Query-by-example represents a visual/graphical approach for accessing information in a database through the use of query templates called skeleton tables. It is used by entering example values directly into a query template to represent what is to be achieved. QBE is used by many database systems for personal computers. QBE is a very powerful facility that gives the user the capability to access the information a user wants without the knowledge of any programming language. Queries in QBE are expressed by skeleton tables. QBE has two distinct features:

QBE has the two-dimensional syntax: Queries look like tables.

**22. Why are cursors necessary in embedded SQL?**

A cursor is an object used to store the output of a query for row-by-row processing by the application programs. SQL statements operate on a set of data and return a set of data. On other hand, host language programs operate on a row at a time. The cursors are used to navigate through a set of rows returned by an embedded SQL SELECT statement. A cursor can be compared to a pointer.

**23. What is the purpose of normalization in DBMS?**

Database normalization is the process of organizing the attributes of the database to reduce or eliminate data redundancy (having the same data but at different places).

**Purpose of normalization:**

It is used to remove duplicate data and database anomalies from the relational table.

Normalization helps to reduce redundancy and complexity by examining new data types used in the table.

It is helpful to divide the large database table into smaller tables and link them using relationships.

It avoids duplicate data or no repeating groups into a table.

It reduces the chances for anomalies to occur in a database.

**24. What is the difference between a database schema and a database state?**

The collection of information stored in a database at a particular moment in time is called database state while the overall design of the database is called the database schema.

**25. 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, deleting and updating/modifying the data in the database.

**26. Explain the concepts of a Primary key and Foreign Key.**

Primary Key is used to uniquely identify the records in a database table while Foreign Key is mainly used to link two or more tables together, as this is a particular field(s) in one of the database tables which are the primary key of some other table.

Example: There are 2 tables – Employee and Department. Both have one common field/column as ‘ID’ where ID is the primary key of the Employee table while this is the foreign key for the Department table.

**27.What are the main differences between Primary key and Unique Key?**

Given below are few differences:

The main difference between the Primary key and the Unique key is that the Primary key can never have a null value while the Unique key may consist of a null value.

 In each table, there can be only one primary key while there can be more than one unique key in a table.

**28. What is the concept of sub-query in terms of SQL?**

Sub-query is basically the query that is included inside some other query and can also be called an inner query which is found inside the outer query.

**29. What is the use of the DROP command and what are the differences between DROP, TRUNCATE and DELETE commands?**

DROP command is a DDL command which is used to drop/delete the existing table, database, index, or view from the database.

The major difference between DROP, TRUNCATE and DELETE commands are:

DROP and TRUNCATE commands are the DDL commands which are used to delete tables from the database and once the table gets deleted, all the privileges and indexes that are related to the table also get deleted. These 2 operations cannot be rolled back and so should be used only when necessary.

DELETE command, on the other hand, is a DML Command which is used to delete rows from the table and this can be rolled back.

**30. What is the main difference between UNION and UNION ALL?**

UNION and UNION ALL are used to join the data from 2 or more tables but UNION removes duplicate rows and picks the rows which are distinct after combining the data from the tables whereas UNION ALL does not remove the duplicate rows, it just picks all the data from the tables.

**31. What is Correlated Subquery in DBMS?**

A Subquery is also known as a nested query i.e. a query written inside some query. When a Subquery is executed for each of the rows of the outer query then it is termed as a Correlated Subquery.

An example of Non-Correlated Subquery is:

|  |
| --- |
| SELECT \* from EMP WHERE ‘RIYA’ IN (SELECT Name from DEPT WHERE EMP.EMPID=DEPT.EMPID); |

Here, the inner query is not executed for each of the rows of the outer query.

**32.  Explain Entity, Entity Type, and Entity Set in DBMS?**

The entity is an object, place, or thing which has its independent existence in the real world and about which data can be stored in a database. For Example, any person, book, etc.

Entity Type is a collection of entities that have the same attributes. For Example, the STUDENT table contains rows in which each row is an entity holding the attributes like name, age, and id of the students, hence STUDENT is an Entity Type that holds the entities having the same attributes.

Entity Set is a collection of entities of the same type. For Example, A collection of the employees of a firm.

**33. 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.

**34 . What integrity rules exist in the DBMS?**

There are two major integrity rules that exist in the DBMS.

**Entity Integrity:** This states a very important rule that the value of a Primary key can never have a NULL value.

**Referential Integrity:** This rule is related to the Foreign key which states that either the value of a Foreign key is a NULL value or it should be the primary key of any other relation.

**35. What is E-R model in the DBMS?**

 E-R model is known as an Entity-Relationship model in the DBMS which is based on the concept of the Entities and the relationship that exists among these entities.

**36. What is a functional dependency in the DBMS?**

 This is basically a constraint that is useful in describing the relationship among the different attributes in a relation.

Example: If there is some relation ‘R1’ which has 2 attributes as Y and Z then the functional dependency among these 2 attributes can be shown as Y->Z which states that Z is functionally dependent on Y.

**37. What is 1NF in the DBMS?**

 1NF is known as the First Normal Form.

This is the easiest form of the normalization process which states that the domain of an attribute should have only atomic values. The objective of this is to remove the duplicate columns that are present in the table.

**38.  What is 2NF in the DBMS?**

 2NF is the Second Normal Form.

Any table is said to have in the 2NF if it satisfies the following 2 conditions:

A table is in the 1NF.

 Each non-prime attribute of a table is said to be functionally dependent in totality on the primary key.

**39.  What is 3NF in the DBMS?**

 3NF is the Third Normal Form.

Any table is said to have in the 3NF if it satisfies the following 2 conditions:

 A table is in the 2NF.

Each non-prime attribute of a table is said to be non-transitively dependent on every key of the table.

**40.  What is BCNF in the DBMS?**

 BCNF is the Boyce Codd Normal Form which is stricter than the 3NF.

Any table is said to have in the BCNF if it satisfies the following 2 conditions:

A table is in the 3NF.

For each of the functional dependencies X->Y that exists, X is the super key of a table.

**41. What is a CLAUSE in terms of SQL?**

 This is used with the SQL queries to fetch specific data as per the requirements on the basis of the conditions that are put in the SQL. This is very helpful in picking the selective records from the complete set of records.

For Example, There is a query that has a WHERE condition or the query with the HAVING clause.

**42.How can you get the alternate records from the table in the SQL?**

 If you want to fetch the odd numbers then the following query can be used:

|  |
| --- |
| SELECT EmpId from (SELECT rowno,EmpId from Emp) WHERE mod(rowno,2)=1; |

If you want to fetch the even numbers, then the following query can be used:

|  |
| --- |
| SELECT EmpId from (SELECT rowno,EmpId from Emp) WHERE mod(rowno,2)=0; |

**43.  How is the pattern matching done in the SQL?**

Answer: With the help of the LIKE operator, pattern matching is possible in the SQL.’%’ is used with the LIKE operator when it matches with the 0 or more characters, and ‘\_’ is used to match the one particular character.

Example:

|  |
| --- |
| SELECT \* from Emp WHERE name like ‘b%’; |

|  |
| --- |
| SELECT \* from Emp WHERE name like ‘hans\_’; |

**44. What is a join in the SQL?**

A Join is one of the SQL statements which is used to join the data or the rows from 2 or more tables on the basis of a common field/column among them.

**45. What are the different types of joins in SQL?**

There are 4 types of SQL Joins:

 Inner Join: This type of join is used to fetch the data among the tables which are common in both tables.

 Left Join: This returns all the rows from the table which is on the left side of the join but only the matching rows from the table which is on the right side of the join.

 Right Join: This returns all the rows from the table which is on the right side of the join but only the matching rows from the table which is on the left side of the join.

 Full Join: This returns the rows from all the tables on which the join condition has been put and the rows which do not match hold null values.

**46. Explain the Stored Procedure.**

A Stored Procedure is a group of SQL statements in the form of a function that has some unique name and is stored in relational database management systems(RDBMS) and can be accessed whenever required.

**47. 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.

**48. What are the different types of relationships in the DBMS?**

A Relationship in DBMS depicts an association between the tables.

Different types of relationships are:

**One-to-One**: This basically states that there should be a one-to-one relationship between the tables i.e. there should be one record in both the tables.

**One-to-Many**: This states that there can be many relationships for one i.e. a primary key table hold only one record which can have many, one, or none records in the related table.

**Many-to-Many**: This states that both the tables can be related to many other tables.

**49. What do you mean by Entity type extension?**

Compilation of similar entity types into one particular type which is grouped together as an entity set is known as entity type extension.

**50. What is conceptual design in dbms?**

Conceptual design is the first stage in the database design process. The goal at this stage is to design a database that is independent of database software and physical details. The output of this process is a conceptual data model that describes the main data entities, attributes, relationships, and constraints of a given problem domain.

**51. Differentiate between logical database design and physical database design. Show how this separation leads to data independence.**

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Logical Database Design** | **Physical Database Design** |
| Task | Maps or transforms the conceptual schema (or an ER schema) from the high-level data model into a relational database schema. | The specifications for the stored database in terms of physical storage structures, record placement, and indexes are designed. |
| Choice of criteria | The mapping can proceed in two stages:   * System-independent mapping but data model-dependent * Tailoring the schemas to a specific DBMS | The following criteria are often used to guide the choice of physical database design options:   * Response Time * Space Utilization * Transaction Throughput |
| Result | DDL statements in the language of the chosen DBMS that specify the conceptual and external level schemas of the database system. But if the DDL statements include some physical design parameters, a complete DDL specification must wait until after the physical  database design phase is completed. | An initial determination of storage structures and the access paths for the database files. This corresponds to defining the internal schema in terms of Data Storage Definition Language. |

The database design is divided into several phases. The logical database design and physical database design are two of them. This separation is generally based on the concept of the three-level architecture of DBMS, which provides data independence. Therefore, we can say that this separation leads to data independence because the output of the logical database design is the conceptual and external level schemas of the database system which is independent of the output of the physical database design that is an internal schema.

**52. What are temporary tables? When are they useful?**  
 Temporary tables exist solely for a particular session, or whose data persists for the duration of the transaction. The temporary tables are generally used to support specialized rollups or specific application processing requirements. Unlike a permanent table, space is not allocated to a temporary table when it is created. Space will be dynamically allocated for the table as rows are inserted. The CREATE GLOBAL TEMPORARY TABLE command is used to create a temporary table in Oracle.

**53. Explain different types of failures that occur in the** **Oracle database.**  
**Types of Failures –**In the Oracle database following types of failures can occur:

* **Statement Failure·**
* **Bad data type**
  + Insufficient space
* **Insufficient Privileges** (e.g., object privileges to a role)
* **User Process Failure**
  + The user performed an abnormal disconnect
  + The user’s session was abnormally terminated
  + The user’s program raised an address exception
* **User Error**
  + The user drops a table
  + User damages data by modification
* **Instance Failure**
* **Media Failure**
  + The user drops a table
  + User damages data by modification
* **Alert Logs**
  + Records informational and error messages
  + All Instance startups and shutdowns are recorded in the log

**54. What is the main goal of RAID technology?**

**RAID**stands for **R**edundant **A**rray of **I**nexpensive (or sometimes “Independent”)**D**isks.

RAID is a method of combining several hard disk drives into one logical unit (two or more disks grouped together to appear as a single device to the host system). RAID technology was developed to address the fault-tolerance and performance limitations of conventional disk storage. It can offer fault tolerance and higher throughput levels than a single hard drive or group of independent hard drives. While arrays were once considered complex and relatively specialized storage solutions, today they are easy to use and essential for a broad spectrum of client/server applications.

## 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](https://prepinsta.com/dbms/what-is-database-management-system/) 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?

* **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.](https://prepinsta.com/dbms/normalisation/)

### 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](https://prepinsta.com/dbms/set-operations/) 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](https://prepinsta.com/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.

#### Also Check:

[Most Asked Technical Interview Questions](https://prepinsta.com/interview-preparation/technical-interview/)

## FAQs on DBMS Interview Questions

## Describe the types of keys?

**There are following types of keys:**

**Primary key**: The Primary key is an attribute in a table that can uniquely identify each record in a table. It is compulsory for every table.

**Candidate key**: The Candidate key is an attribute or set of an attribute which can uniquely identify a tuple. The Primary key can be selected from these attributes.

**Super key**: The Super key is a set of attributes which can uniquely identify a tuple. Super key is a superset of the candidate key.

**Foreign key**: The Foreign key is a primary key from one table, which has a relationship with another table. It acts as a cross-reference between tables.

<https://prepinsta.com/dbms/>

https://prepinsta.com/dbms/

Introduction

• What is Data

• What is Database

• What is Database Management System

• File Systems vs DBMS (Why we shifted to DBMS)

• Tuple

• Record

Architecture and Models

DBMS Architecture

2 Level

3 Level

Need For DBMS

Data Abstraction and Data Independence

DBMS Database Models

Data Models

Hierarchical Model

Network Model

Entity-relationship Model

Relational Model

Object-Oriented Relation DBMS

ER Models

ER Data Model Basics concepts. (Attributes, relationships, Entities)

Entity-Relationship Model Advanced

Attribute

Key Attribute

Composite Attribute

Multivalued Attribute

Derived Attribute

Cardinality

Cardinality Ratio

One to One

One to Many

Many to One

Many to Many

Relationships

Types of Relationship

Unary Relationship

Binary Relationship

n – ary Relationship

Recursive Relationship

Ternary Relationship

Relationship – Cardinality

Participation Constraint

Weak Entity and Strong Entity

Generalization Specialization and Aggregation

Generalization

Specialization

Aggregation

Relational Database Model

Codd’s Rule

RDBMS

Relational Data Model

Keys in the Relational Model

Keys in DBMS

Primary Key in DBMS | Oracle | SQL

Foreign Key in DBMS | Oracle | SQL

Candidate Key in DBMS | Oracle | SQL

Super Key in DBMS | Oracle | SQL

Alternate Key in DBMS | Oracle | SQL

Composite Key in DBMS | Oracle | SQL

Non-Prime Attribute

SQL Constraints

Default in DBMS | Oracle | SQL

Not Null in DBMS | Oracle | SQL

Unique in DBMS | Oracle | SQL

Check-in DBMS | Oracle | SQL

Key Constraints in DBMS | Oracle | SQL

Domain constraints in DBMS | Oracle | SQL

Mapping constraints

ACID Properties in DBMS | Oracle | SQL

Normalisation

Introduction Normalisation

Normal Forms

1NF Form

2NF Form

3NF Form

Boyce-Codd Normal Form

4NF

5NF

Joins in DBMS | SQL

Inner Joins in DBMS | SQL

Outer Join in DBMS | SQL

Left Join in DBMS | SQL

Right Join in DBMS | SQL

SQL

SQL -Operators in DBMS | Oracle | Pl/SQL

SQL- Expressions in DBMS | Oracle | Pl/SQL

CREATE Table in DBMS | Oracle | Pl/SQL

ALTER table in DBMS | Oracle | Pl/SQL

DROP/TRUNCATE/RENAME table in DBMS | Oracle | Pl/SQL

Tuple/Record in DBMS | Oracle | Pl/SQL

SELECT in DBMS | Oracle | SQL

INSERT in DBMS | Oracle | SQL

WHERE in DBMS | Oracle | SQL

AND & OR in DBMS | Oracle | SQL

UPDATE in DBMS | Oracle | SQL

DELETE in DBMS | Oracle | SQL

LIKE in DBMS | Oracle | SQL

ORDER BY in DBMS | Oracle | SQL

GROUP BY in DBMS | Oracle | SQL

DISTINCT in DBMS | Oracle | SQL

ROW in DBMS | Oracle | SQL

TRUNCATE in DBMS | Oracle | SQL

RENAME in DBMS | Oracle | SQL

DROP in DBMS | Oracle | SQL