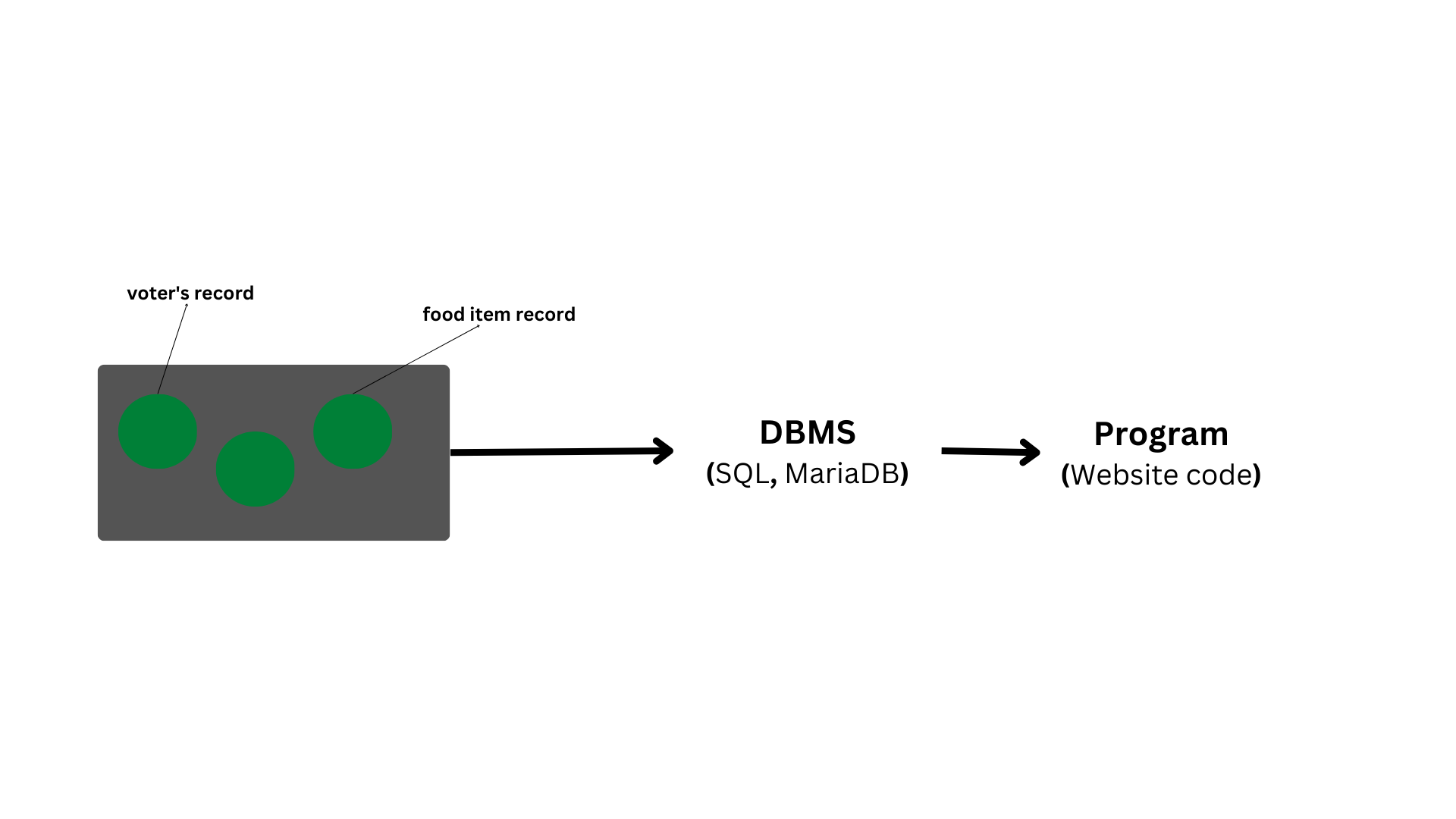
**1. What is DBMS?**

A database management system (DBMS) is a set of tools that make it easier for users to construct and maintain databases. In other words, a database management system (DBMS) provides us with an interface or tool for completing various tasks such as creating a database, entering data into it, deleting data from it, updating data, and so on. A database management system (DBMS) is software that allows data to be kept in a more secure manner than a file-based system. We can solve a variety of issues using DBMS, such data redundancy, data inconsistency, quick access, more ordered and intelligible data, and so on.  
There are some well-known Database Management Systems, such as MySQL, Oracle, SQL Server, Amazon Simple DB (Cloud-based), and so on.



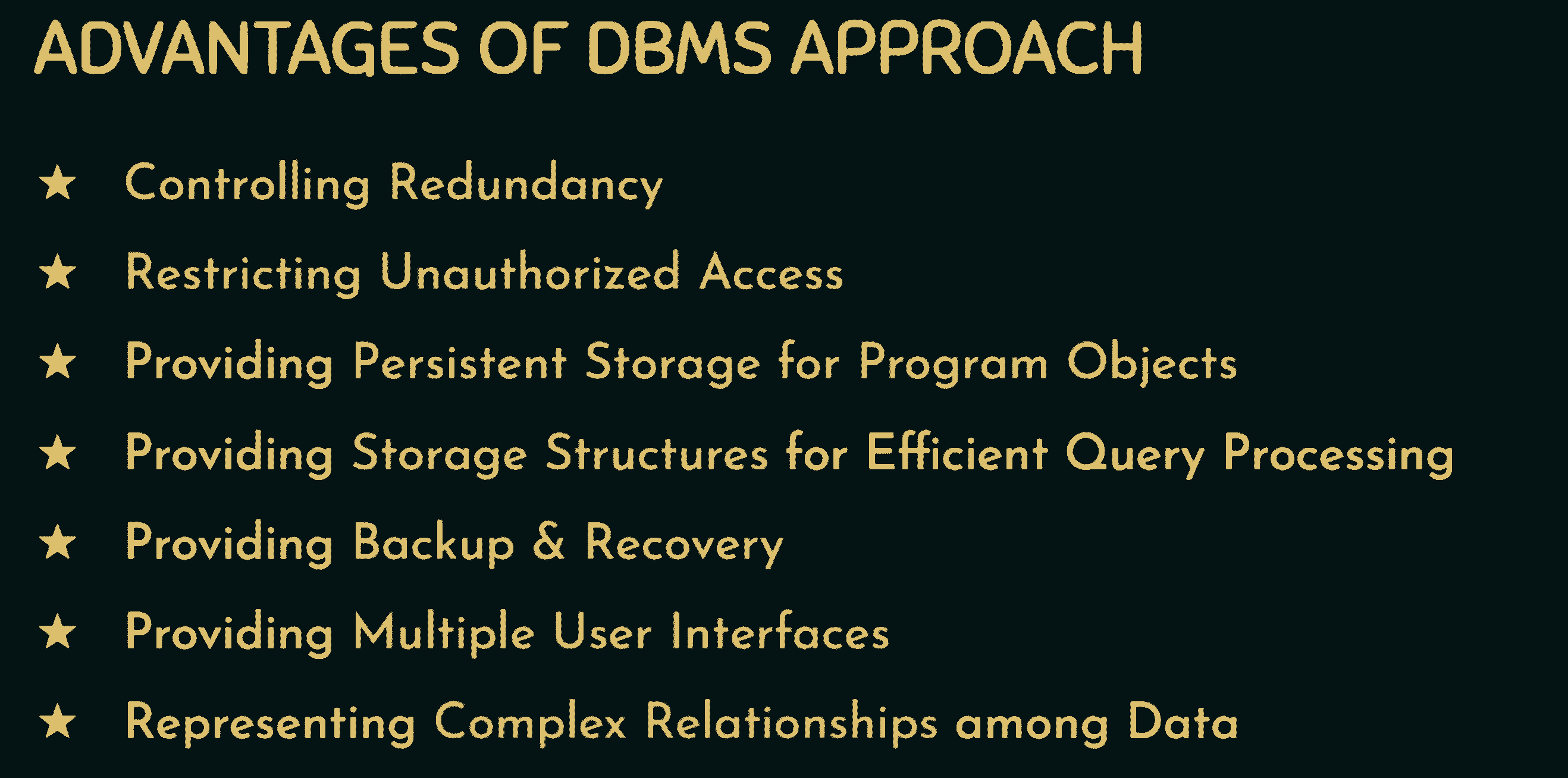
**2. What is Database?**

A database is a collection of logical, consistent, and organised data that can be easily accessed, controlled, and updated. Databases, also known as electronic databases, are structured to allow for the efficient production, insertion, and updating of data and are saved as a file or set of files on magnetic discs, tapes, and other secondary devices. Objects (tables) make up the majority of a database, and tables contain records and fields. Fields are the fundamental units of data storage, containing information on a certain element or attribute of the database's entity. A database management system (DBMS) is used to extract data from a database in the form of queries. To make it easier to access relevant information, you can organise data into tables, rows, and columns, as well as index it.  
Database handlers design a database such that all users have access to the data through a single piece of tools.  
The database's primary goal is to manage a huge amount of data by storing, retrieving, and managing it.  
Databases are used to manage a large number of dynamic websites on the Internet today. Consider a model that checks the availability of hotel rooms. It's an example of a database-driven dynamic webpage. Databases such as MySQL, Sybase, Oracle, MongoDB, Informix, PostgreSQL, SQL Server, and others are available.

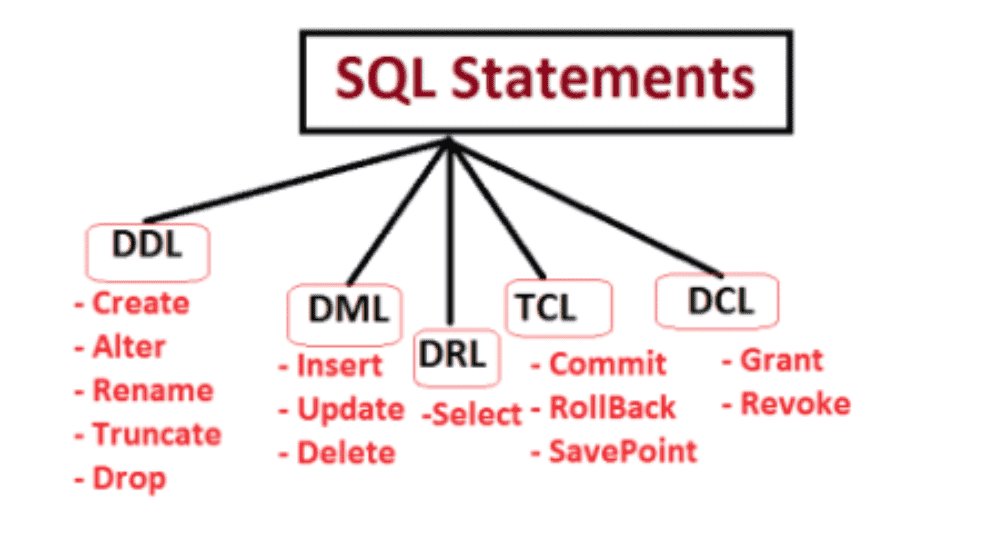
**3. Mention the issues with traditional file-based systems that make DBMS a better choice?**

The lack of indexing in a traditional file-based system leaves us with little choice but to scan the entire page, making content access time-consuming and slow. The other issue is redundancy and inconsistency, as files often include duplicate and redundant data, and updating one causes all of them to become inconsistent. Traditional file-based systems make it more difficult to access data since the data is disorganised.  
Another drawback is the lack of concurrency management, which causes one operation to lock the entire page, as opposed to DBMS, which allows several operations to work on the same file at the same time.  
Other concerns with traditional file-based systems that DBMSs have addressed include integrity checks, data isolation, atomicity, security, and so on.

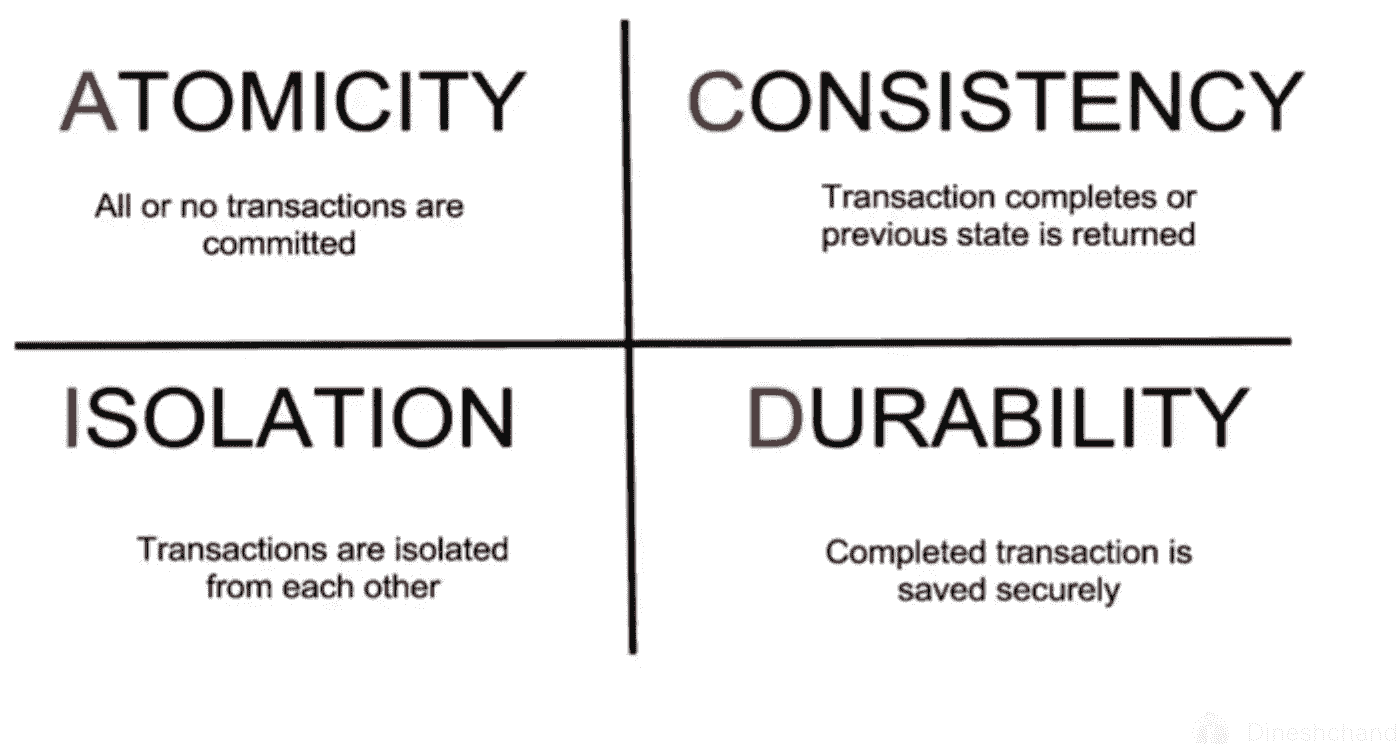
**4. Explain a few advantages of a DBMS.**

The following are some of the benefits of using a database management system:  
  
🚀 Data sharing: Data from the same database can be accessed by multiple people at the same time.  
🚀 Integrity restrictions: These limitations allow for more refined data storage in a database.  
🚀 Data redundancy control: Supports a system for controlling data redundancy by combining all data into a single database.  
🚀 Data Independence: Allows the structure of the data to be changed without affecting the structure of any running application applications.  
  
🚀 Backup and recovery feature: Provides a 'backup and recovery' feature that automatically creates a data backup and restores the data as needed.  
🚀 Data Security: A database management system (DBMS) provides the capabilities needed to make data storage and transfer more dependable and secure. Some common technologies used to safeguard data in a DBMS include authentication (the act of granting restricted access to a user) and encryption (encrypting sensitive data such as OTP, credit card information, and so on).

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

The following are some of the DBMS languages:  
  
🚀 DDL (Data Definition Language) is a language that contains commands for defining databases. CREATE, ALTER, DROP, TRUNCATE, RENAME, and so on.  
🚀 DML (Data Manipulation Language) is a set of commands that can be used to manipulate data in a database. SELECT, UPDATE, INSERT, DELETE, and so on.  
  
🚀 DCL (Data Control Language): It offers commands for dealing with the database system's user permissions and controls. GRANT and REVOKE, for example.  
🚀 TCL (Transaction Control Language) is a programming language that offers commands for dealing with database transactions. COMMIT, ROLLBACK, and SAVEPOINT, for example.

**6. What is meant by ACID properties in DBMS?**

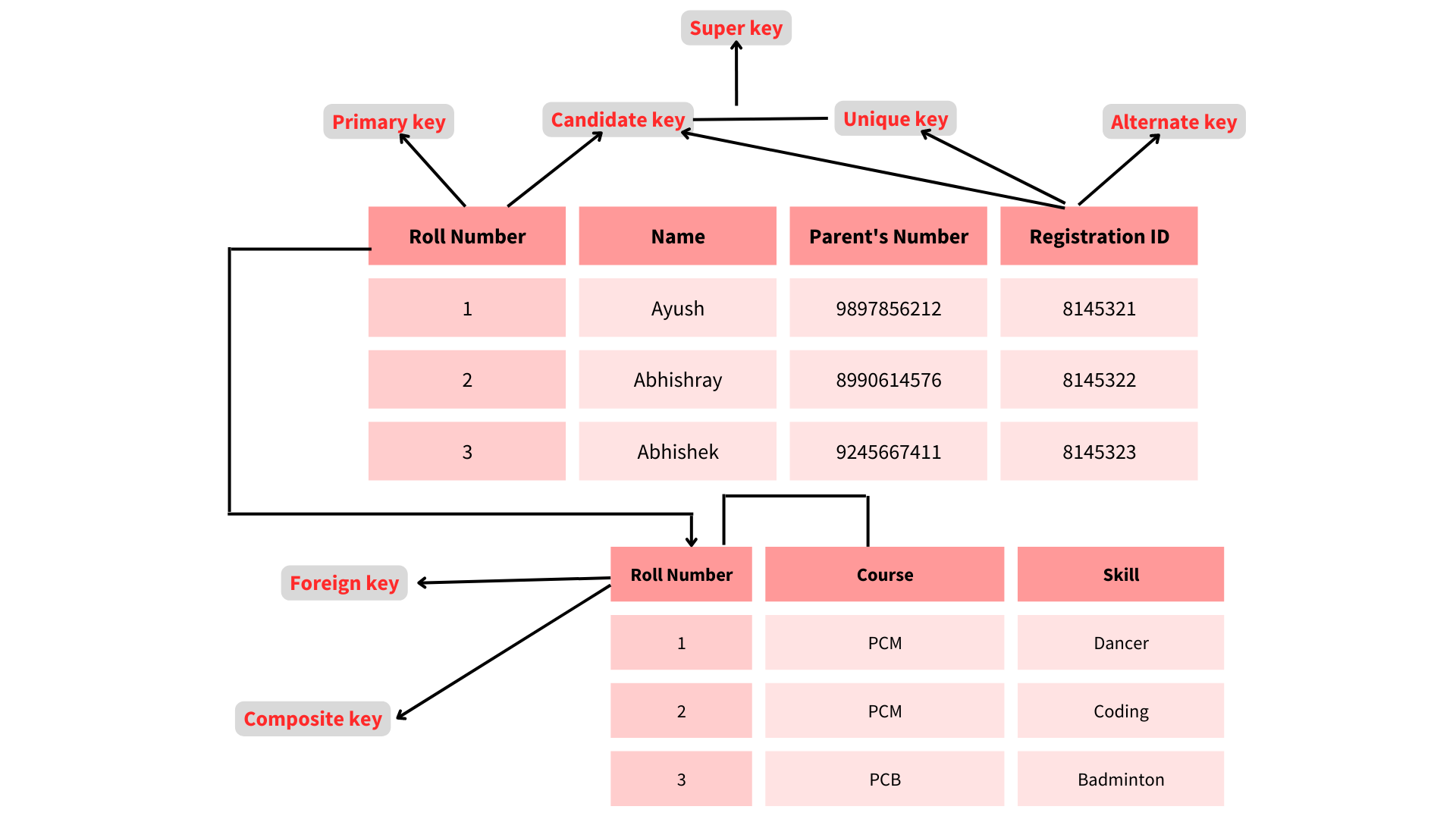
The ACID properties of a database management system are the basic principles that must be observed in order to maintain data integrity. They are as follows:  
  
🚀Atomicity - Also known as the "all or nothing" rule, atomicity states that everything evaluated as a single unit is either executed to completion or not at all.  
🚀 Consistency - This attribute indicates that the database's data is consistent before and after each transaction.  
🚀  
Isolation - This characteristic specifies that several transactions can be conducted at the same time.  
🚀 Durability - This characteristic ensures that each transaction is saved in non-volatile memory once it has been finished.

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

No, a NULL value is distinct from zero and blank space in that it denotes a value that is assigned, unknown, unavailable, or not applicable, as opposed to blank space, which denotes a character, and zero, which denotes a number.  
For instance, a NULL value in "number of courses" taken by a student indicates that the value is unknown, but a value of 0 indicates that the student has not taken any courses.

**8. What are super, primary, candidate, and foreign keys?**

🚀A super key is a set of relation schema attributes that all other schema attributes are functionally dependent on. The values of super key attributes cannot be the identical in any two rows.  
🚀A Candidate key is a minimum superkey, which means that no suitable subset of Candidate key properties may be used to create a superkey.  
🚀One of the candidate keys is the Primary Key.  
🚀One of the candidate keys is chosen as the primary key and becomes the most important. In a table, there can only be one main key.  
🚀A foreign key is a field (or set of fields) in one table that is used to uniquely identify a row in another table.



**9. What is the difference between primary key and unique constraints?**

Although the primary key cannot have a NULL value, the unique constraints can. A table has just one main key, but it might have numerous unique constraints.

**10. What is meant by DBMS and what is its utility? Explain RDBMS with examples.**

The Database Management System, or DBMS, is a collection of applications or programmes that allow users to construct and maintain databases. A database management system (DBMS) provides a tool or interface for executing various database activities such as inserting, removing, updating, and so on. It is software that allows data to be stored in a more compact and secure manner than a file-based system. A database management system (DBMS) assists a user in overcoming issues such as data inconsistency, data redundancy, and other issues in a database, making it more comfortable and organised to use.  
File systems, XML, the Windows Registry, and other DBMS systems are examples of prominent DBMS systems.  
  
RDBMS stands for Relational Database Management System, and it was first introduced in the 1970s to make it easier to access and store data than DBMS. In contrast to DBMS, which stores data as files, RDBMS stores data as tables. When opposed to DBMS, storing data as rows and columns makes it easier to locate specific values in the database and makes it more efficient.  
MySQL, Oracle DB, and other prominent RDBMS systems are examples.

**11. What is a checkpoint in DBMS?**

The Checkpoint is a technique that removes all previous logs from the system and stores them permanently on the storage drive.  
Preserving the log of each transaction and maintaining shadow pages are two methods that can assist the DBMS in recovering and maintaining the ACID properties. When it comes to a log-based recovery system, checkpoints are necessary. Checkpoints are the minimal points from which the database engine can recover after a crash as a specified minimal point from which the transaction log record can be utilised to recover all committed data up to the moment of the crash.

**12. What is a database system?**

A database system is a collection of database and database management system software. We can execute a variety of tasks using the database system, including:  
The data can be easily stored in the database, and there are no concerns about data redundancy or inconsistency.  
When necessary, data will be pulled from the database using DBMS software. As a result, using database and DBMS software together allows you to store, retrieve, and access data with precision and security.

**13. What do you mean by Data Model?**

A data model consists of a set of tools for describing data, semantics, and constraints. They also assist in the description of the relationship between data entities and their attributes. Hierarchical data models, network models, entity relationship models, and relational models are some of the most prevalent data models. You may also learn more about data models by looking at other data modelling interview questions.

**14. When does checkpoint occur in DBMS?**

A checkpoint is a snapshot of the database management system's current state. The DBMS can use checkpoints to limit the amount of work that needs to be done during a restart in the event of a subsequent crash. After a system crash, checkpoints are utilised to recover the database. The log-based recovery solution employs checkpoints. When we need to restart the system because of a system crash, we use checkpoints. As a result, we won't have to execute the transactions from the beginning.

**15. What is the difference between an entity and an attribute?**

In a database, an entity is a real-world thing. Employee, designation, department, and so on are examples of different entities in an employee database.  
A trait that describes an entity is called an attribute. For example, the entity "employee" can have properties such as name, ID, and age.

16. **What are the various kinds of interactions catered by DBMS?**

DBMS can handle a variety of interactions, including:  
  
🚀Data definition  
🚀 Update  
🚀 Retrieval  
🚀 Administration

**17. What do you understand by query optimization?**

Query optimization is the phase in which a plan for evaluating a query with the lowest estimated cost is identified. When there are numerous algorithms and approaches to perform the same goal, this phase emerges.  
The following are some of the benefits of query optimization:

 • The output is delivered more quickly.

 • In less time, a higher number of queries can be run.

 • Reduces the complexity of time and space

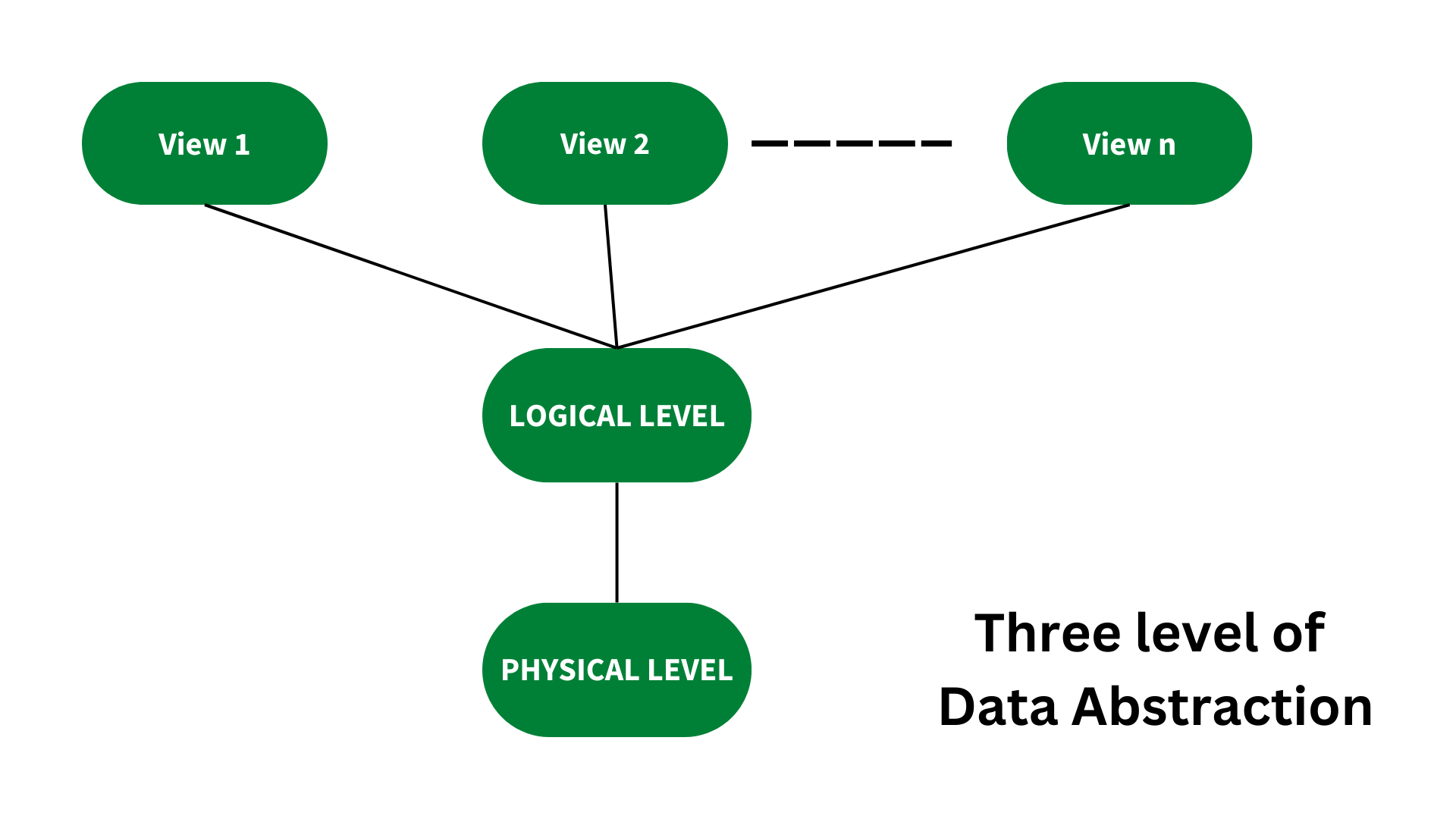
**18. Do we consider NULL values the same as that of blank space or zero?**

A NULL value is not to be confused with a value of zero or a blank space. While zero is a number and blank space is a character, NULL denotes a value that is unavailable, unknown, assigned, or not applicable.

**19. What do you understand by aggregation and atomicity?**

|  |  |
| --- | --- |
| **Aggregation** | **Atomicity** |
| This is an E-R model feature that allows one relationship set to interact with another relationship set. | This attribute specifies that a database alteration must either adhere to all of the rules or not at all. As a result, if one portion of the transaction fails, the transaction as a whole fails. |

**20. What are the different levels of abstraction in the DBMS?**

In DBMS, there are three degrees of data abstraction.  
  
They are as follows:  
  
🚀Physical Level : The physical level of abstraction specifies how data is stored and is the lowest degree of abstraction.  
🚀Logical Layer : After the Physical level, there is the Logical level of abstraction. This layer decides what data is saved in the database and how the data pieces relate to one another.  
🚀 View Level: The greatest level of abstraction, the View Level describes only a portion of the entire database.

**21. What is an entity-relationship model?**

It's a diagrammatic approach to database architecture in which real-world things are represented as entities and relationships between them are mentioned. This method allows the DBA staff to quickly grasp the schema.

**22. What do you understand by the terms Entity, Entity Type, and Entity Set in DBMS?**

🚀Entity: An entity is a real-world object with attributes, which are nothing more than the object's qualities. An employee, for example, is a type of entity. This entity can have attributes like empid, empname, and so on.  
🚀Entity Type: An entity type is a collection of entities with similar attributes. An entity type, in general, refers to one or more related tables in a database. As a result, entity type can be thought of as a trait that uniquely identifies an entity. Employees can have attributes such as empid, empname, department, and so on.  
🚀Entity Set: In a database, an entity set is a collection of all the entities of a specific entity type.An entity set can include, for example, a group of employees, a group of companies, and a group of persons.

**23. What do you mean by transparent DBMS?**

The transparent DBMS is a form of database management system that conceals its physical structure from users. Physical structure, also known as physical storage structure, refers to the DBMS's memory manager and explains how data is saved on disc.

**24. What are the unary operations in Relational Algebra?**

In relational algebra, the unary operations are PROJECTION and SELECTION. Single-operand operations are known as unary operations. SELECTION, PROJECTION, and RENAME are unary operations.  
Relational operators, like as =,=,>=, and others, are employed in SELECTION.

**25. What is RDBMS?**

Relational Database Management Systems (RDBMS) is an acronym for Relational Database Management Systems. It's used to keep track of data records and table indices. RDBMS is a type of database management system that employs structure to identify and access data about other pieces of data in the database. RDBMS is a database management system that allows you to update, insert, delete, manipulate, and administer a relational database with minimal effort. The SQL language is utilised by RDBMS the majority of the time since it is simple to grasp and is frequently employed.

**26. What are the differnt data models?**

There are number of data modesl and they are :  
  
🚀 Hierarchical data model  
🚀Network model  
🚀Relational model  
🚀Entity-Relationship model

**27. Define a Relation Schema and a Relation.**

A Relation Schema is a collection of properties that define a relationship. Table schema is another name for it. It specifies the name of the table. The blueprint with which we may explain how data is grouped into tables is known as the relation schema. There is no data in this blueprint.  
A set of tuples is used to define a relation. A connection is a collection of connected attributes with key attributes that identify them.  
Consider the following scenario:  
Let r be the relation containing set tuples (t1, t2, t3, ..., tn). Each tuple consists of an ordered list of n-values (t=1) (v1,v2, ...., vn).

**28. What is Degree of relation?**

The degree of a relationship is one of its relation schema's attributes. A degree of connection, also known as Cardinality, is defined as the number of times one entity occurs in relation to the number of times another entity occurs. One-to-one (1:1), one-to-many (1:M), and many-to-one (1:N) are the three degrees of relation (M:M).

**29. What is Relationship?**

An association between two or more entities is characterised as a relationship. In a database management system, there are three types of relationships:  
  
🚀One-to-One: In this case, one record of any object can be linked to another object's record.  
🚀One-to-Many (many-to-one): In this case, one record of any object can be linked to many records of other objects, and vice versa.  
🚀Many-to-many: In this case, multiple records of one item can be linked to n records of another object.

**30. What are the disadvantages of file processing systems?**

tThe disadvantages of file processing systems are :  
  
🚀 Data redundancy  
🚀 Not secure  
🚀 Inconsistent  
🚀 Difficult in accessing data  
🚀 Limited data sharing  
🚀 Data integrity  
🚀 Concurrent access is not possible  
🚀 Data isolation  
🚀 Atomicity problem

**31. What is Data Abstraction in DBMS?**

In a database management system, data abstraction is the process of hiding unimportant facts from users. Because database systems are made up of complicated data structures, user interaction with the database is made possible.  
For example, we know that most users prefer systems with a simple graphical user interface (GUI), which means no sophisticated processing. As a result, data abstraction is required to keep the user engaged and to make data access simple. Furthermore, data abstraction divides the system into layers, allowing the job to be stated and properly defined.

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

The following are some of the primary advantages of DBMS:  
  
🚀Controlled Redundancy: DBMS enables a way to control data redundancy inside the database by integrating all data into a single database and preventing duplication of data because data is stored in just one location.  
🚀Data Sharing: In a DBMS, data can be shared among several users at the same time because the same database is shared across all users and by various application applications.  
🚀Backup and Recovery Facility: DBMS eliminates the burden of producing data backups over and over by including a 'backup and recovery' function that automatically produces data backups and restores them as needed.  
🚀Integrity Constraints Must Be Enforced: Integrity Constraints must be enforced. So That The Refined Data Is Stored In The Database And This Is Followed By DBMS  
🚀Data independence: It basically means that you can modify the data structure without affecting the structure of any application applications.

**33. What is the difference between having and where clause?**

In a select statement, HAVING is used to establish a condition for a group or an aggregate function. Before grouping, the WHERE clause picks. After grouping, the HAVING clause picks rows. The WHERE clause, unlike the HAVING clause, cannot contain aggregate functions.

**34. What is a transaction? What are ACID properties?**

A database transaction is a collection of database operations that must be handled as a whole, meaning that all or none of the actions must be executed. A bank transaction from one account to another is a good illustration. Either both debit and credit operations must be completed, or none of them must be completed. The ACID qualities (Atomicity, Consistency, Isolation, and Durability) ensure that database transactions are processed reliably.

**35. What is Join?**

An SQL Join is a technique for combining data from two or more tables based on a shared field.

**36. What is Identity?**

Identity (or AutoNumber) is a column that creates numeric values automatically. It is possible to set a start and increment value, however most DBAs leave them at 1. A GUID column generates numbers as well, but the value cannot be changed. There is no need to index the identity/GUID columns.

**37. What is view in SQL?**

A view is a virtual table created from a SQL statement's result set. We can use the create view syntax to do so.

**38. What are uses of view?**

The uses of view are as follows :  
  
  
1. Views can represent a subset of the data in a table; as a result, a view can limit the extent to which the underlying tables are exposed to the outside world: a user may be allowed to query the view but not the whole of the base database.  
2. Views allow you to combine and simplify numerous tables into a single virtual table.  
3. Views can be used as aggregated tables, in which the database engine aggregates data (sum, average, and so on) and displays the generated results alongside the data.  
4. Views can obscure data complexity.  
5. Views take up extremely minimal storage space; the database simply maintains the specification of a view, not a copy of all the data it displays.  
6. Views can give additional security depending on the SQL engine utilised.

**39. What is a Trigger?**

A trigger is a code that is connected with inserting, updating, or deleting data. When a table's associated query is run, the code is automatically performed. Triggers are useful for maintaining database integrity.

**40. What is a stored procedure?**

A stored procedure is similar to a function in that it contains a collection of operations that have been put together. It includes a set of procedures that are frequently used in applications to perform database activities.

**41. What is the difference between Trigger and Stored Procedure?**

Triggers, unlike Stored Procedures, cannot be called directly. Only inquiries can be linked to them.

**42. What is database normalization?**

It is a method for assessing relation schemas based on their functional dependencies and primary keys in order to obtain the following desirable properties:  
  
1. Keeping Redundancy to a Minimum  
2. Reducing Insertion, Deletion, and Update Inconsistencies  
Relation schemas that don't meet the properties are broken down into smaller relation schemas that might meet the requirements.

**43. What are indexes?**

A database index is a data structure that improves the speed of data retrieval operations on a database table at the expense of more writes and storage space to keep the extra copy of data. On a disc, data can only be stored in one order. Faster search, such as binary search for different values, is sought to provide faster access according to different values. Indexes on tables are constructed for this purpose. These indexes take up more disc space, but they allow for speedier searches based on several frequently queried parameters.

**44. What are clustered and non-clustered Indexes?**

Clustered indexes are the indexes that determine how data is stored on a disc. As a result, each database table can only have one clustered index. Non-clustered indexes define logical ordering rather than physical ordering of data. In most cases, a tree is generated, with the leaves pointing to disc records. For this, a B-tree or B+ tree is utilised.

**45. What is Denormalization?**

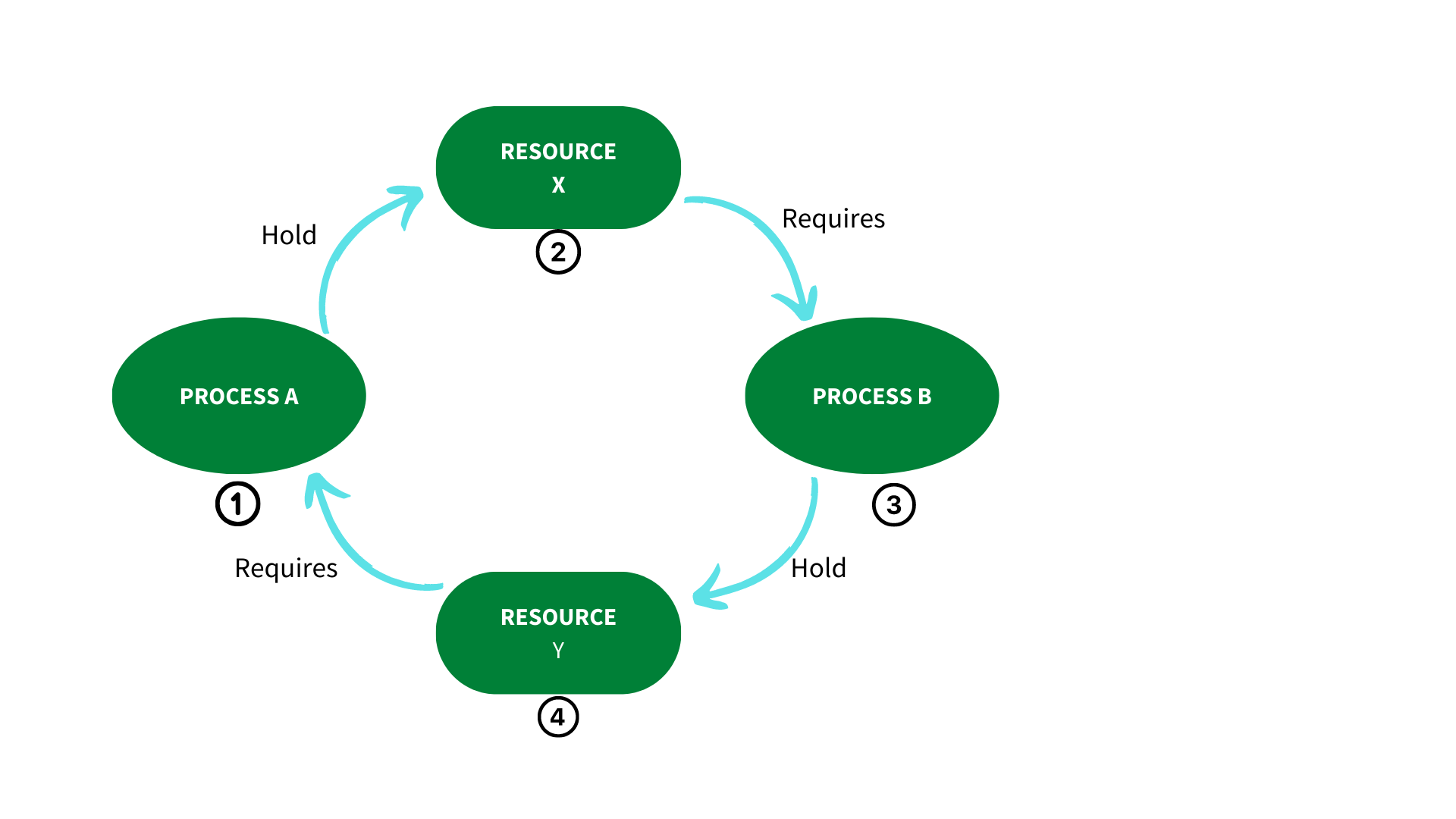
Denormalization is a database optimization method in which duplicated data is added to one or more tables.

**46. What is CLAUSE in SQL?**

In SQL, a clause is a portion of a query that allows you to filter or personalise how your data is queried for you.

**47. What is LiveLock?**

When two or more processes repeatedly repeat the same interaction in reaction to changes in the other processes without producing any beneficial work, this is known as a livelock situation. These processes are not in a condition of waiting, and they are all executing at the same time. This is distinct from a stalemate, which occurs when all processes are in a state of waiting.



**48. What is QBE?**

Query-by-example is a visual/graphical technique to obtaining information in a database by using skeleton tables as query templates. It's used to express what has to be done by explicitly entering example values into a query template. Many database systems for personal computers use QBE. QBE is a very strong tool that allows the user to access the information they want without having to know any programming languages. Skeleton tables are used to express queries in QBE. QBE has two distinguishing characteristics:  
Queries in QBE use a two-dimensional syntax, which makes them look like tables.

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

A cursor is an object that stores the outcome of a query so that application programmes can process it row by row. SQL statements are statements that operate on a collection of data and return another set of data. Host language programmes, on the other hand, work in a row at a time. Cursors are used to move through a set of rows produced by a SQL SELECT statement included in the code. A cursor is similar to a pointer.

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

The practise of structuring the attributes of a database in order to decrease or remove data redundancy is known as database normalisation (having the same data but at different places).  
  
Normalization's purpose:  
  
1. It is used to clean up the relational table by removing duplicate data and database oddities.  
2. By assessing new data types utilised in the table, normalisation helps to decrease redundancy and complexity.  
3. It's a good idea to break down a huge database table into smaller tables and use relationships to connect them.  
4. It prevents duplicate data from being entered into a database, as well as no recurring groups.  
5. It lowers the likelihood of anomalies in a database.

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

Database state refers to the collection of data kept in a database at a specific point in time, whereas database schema refers to the database's overall design.

**52. What is the purpose of SQL?**

SQL stands for Structured Query Language, and its primary purpose is to interact with relational databases by entering, updating, and altering data in the database.

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

Primary Key is used to uniquely identify records in a database table, whereas Foreign Key is used to connect two or more tables together, since it is a specific field(s) in one database table that is the primary key of another table.  
Employee and Department are two tables, for example. Both tables have a similar field/column called 'ID,' which is the primary key for the Employee table and the foreign key for the Department table.

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

A few discrepancies are listed below:  
  
The primary distinction between the Primary and Unique keys is that the Primary key can never contain a null value, whereas the Unique key can.  
There can only be one main key in each table, although there can be multiple unique keys in a table.

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

A sub-query is a query that is contained within another query. It is also known as an inner query because it is found within the outer query.

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

The DROP command is a DDL command that deletes an existing table, database, index, or view from a database.  
The following are the main differences between the DROP, TRUNCATE, and DELETE commands:  
  
The DDL commands DROP and TRUNCATE are used to delete tables from the database, and after the table is gone, all rights and indexes associated with the table are likewise deleted. Because these two actions cannot be reversed, they should only be utilised when absolutely required.  
The DELETE command, on the other hand, is a DML command that may also be rolled back to delete rows from a table.

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

UNION and UNION ALL are both used to connect data from two or more tables, but UNION removes duplicate rows and selects the distinct rows after merging the data from the tables, whilst UNION ALL does not remove duplicate rows and simply selects all of the data from the tables.

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

A nested query, or a query written inside another query, is known as a subquery. A Correlated Subquery is defined as a Subquery that is conducted for each row of the outer query.

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

In a database management system, there are two major integrity rules.  
  
🚀Entity Integrity: This is a crucial rule that stipulates that the value of a primary key can never be NULL.  
🚀Referential Integrity: This rule is connected to the Foreign key and stipulates that a Foreign key's value must be NULL or it must be the primary key of another relation.

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

In relational databases, the E-R model is known as an Entity-Relationship model, and it is built on the concept of Entities and the relationships that exist between them.

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

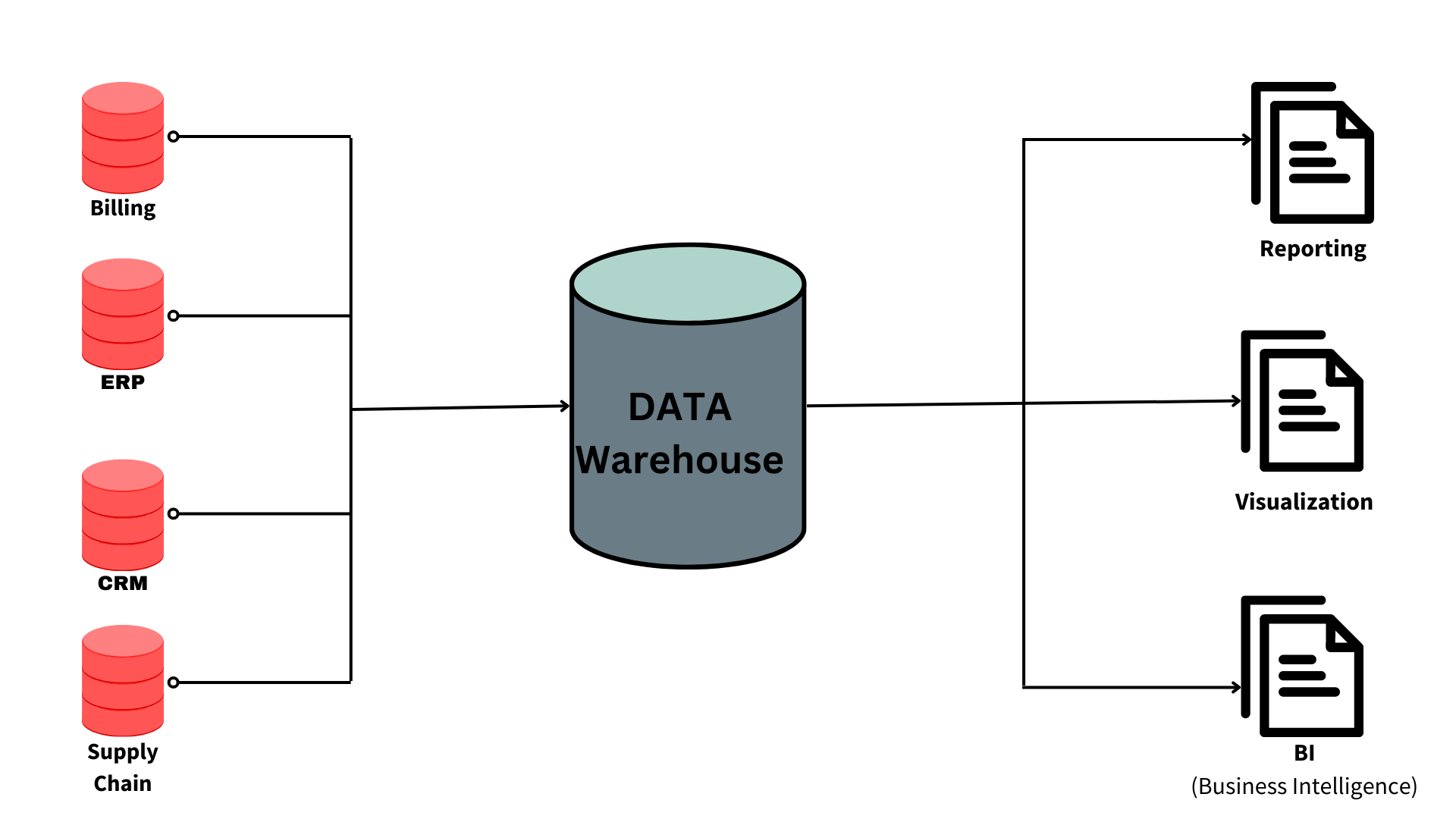
This is a constraint that can be used to describe the relationship between the various attributes of a relation.  
For example, if a relation 'R1' has two attributes, Y and Z, the functional dependency between these two qualities can be represented as Y->Z, indicating that Z is functionally dependent on Y.

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

Pattern matching is feasible in SQL with the help of the LIKE operator. When the LIKE operator matches 0 or more characters, 'percent' is used, while '\_' is used when it matches a single character.

**63. What is meant by Data Warehousing?**

Data warehousing is the process of collecting, extracting, processing, and importing data from numerous sources and storing it in a single database. A data warehouse can be thought of as a central repository for data analytics that receives data from transactional systems and other relational databases. A data warehouse is a collection of historical data from an organisation that aids in decision-making.



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

The main distinction between intension and extension in a database is as follows:  
  
**Intension**: Intension, also known as database schema, is used to define the database's description. It is specified throughout the database's design and typically remains unmodified.  
**Extension**, on the other hand, is a measurement of the number of tuples in a database at any particular point in time. The snapshot of a database is also known as the extension of a database, and its value changes as tuples are created, updated, or destroyed in the database.

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

**DELETE command:** This command is used to delete rows from a table based on the WHERE clause's condition.  
  
🚀It only deletes the rows that the WHERE clause specifies.  
🚀 If necessary, it can be rolled back.  
🚀 It keeps a log to lock the table row before deleting it, which makes it slow.  
  
**TRUNCATE command :**The TRUNCATE command is used to remove all of the data from a table in a database. It's similar to a DELETE command without a WHERE clause.  
  
🚀 It deletes all of the data from a database table.  
🚀It can be rolled back if necessary. (Depending on the database version, truncate can be rolled back, but it's hard and can result in data loss.)  
🚀 It doesn't keep a log and deletes the entire table at once, so it's quick.

**66. 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 method that prevents two or more database users from updating the same piece of data at the same time. When a lock is acquired by a single database user or session, no other database user or session can edit the data until the lock is released.  
  
🚀 Shared Lock :A shared lock is necessary for reading a data item, and in a shared lock, multiple transactions can hold a lock on the same data item. A shared lock allows many transactions to read the data items.  
🚀Exclusive Lock :A lock on any transaction that is about to perform a write operation is known as an exclusive lock.This form of lock prevents any database inconsistency by allowing just one transaction at a time.

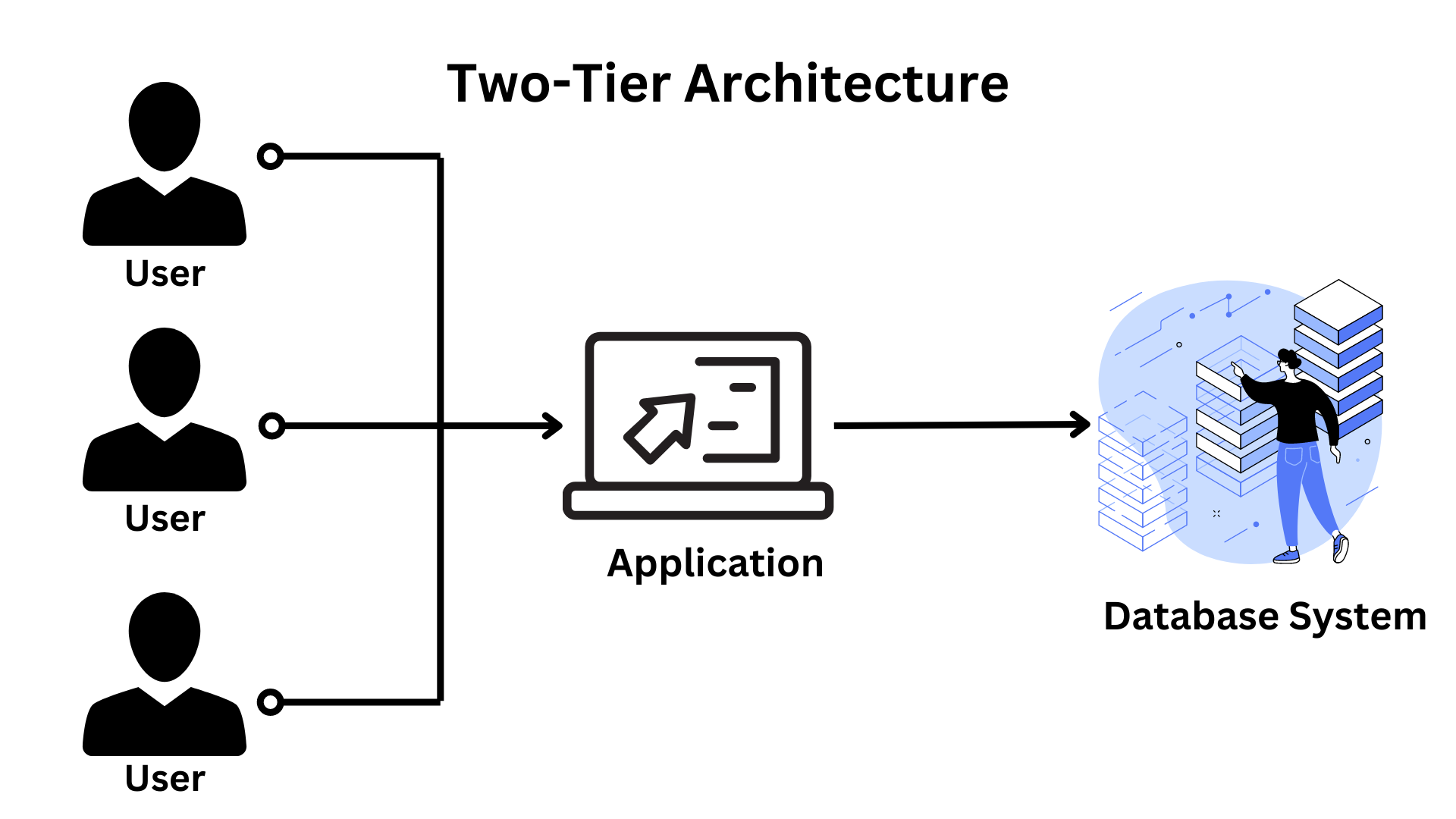
**67. Explain different types of Normalization forms in a DBMS.**

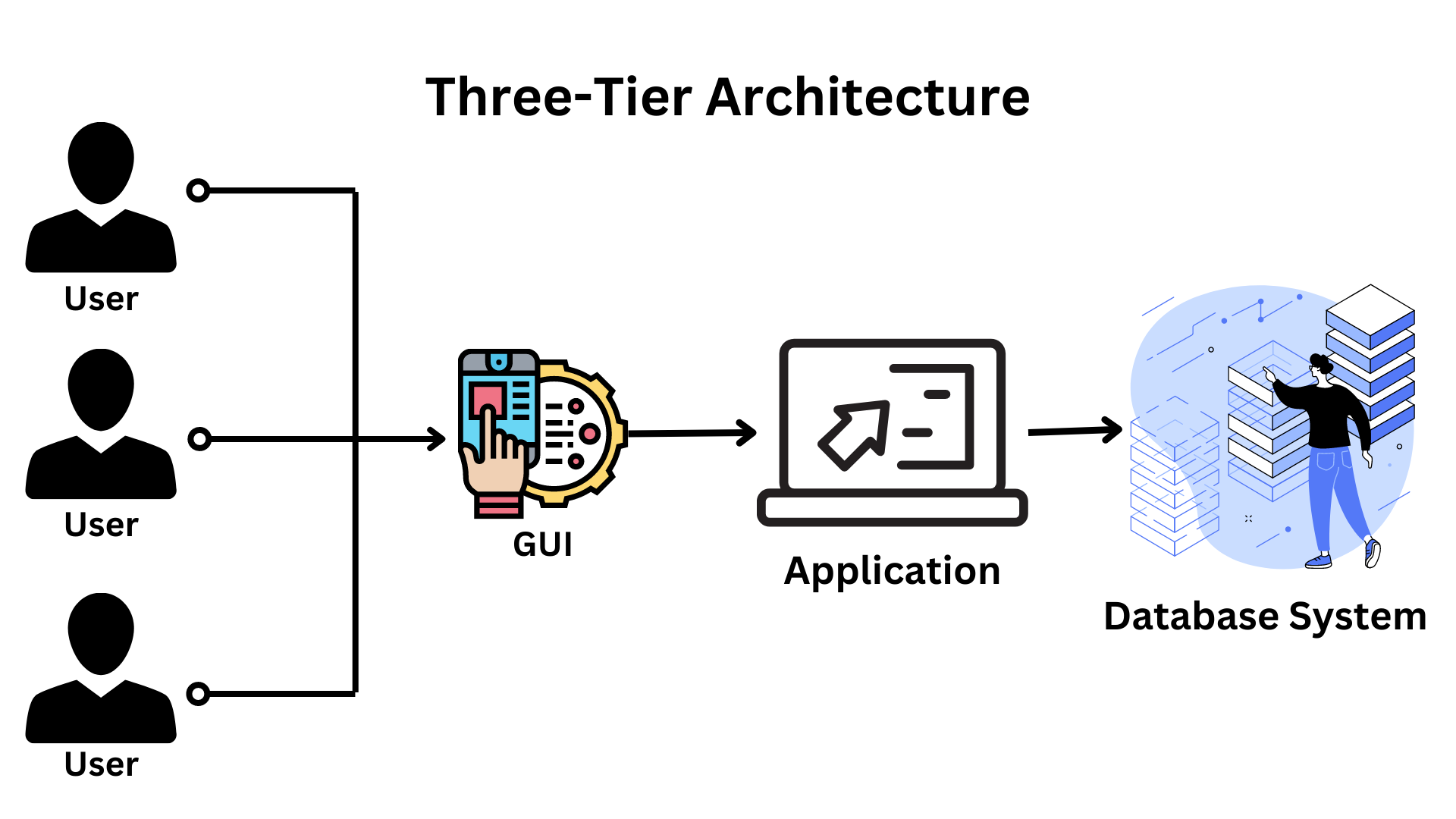
The major normalisation forms in a database management system are as follows:  
  
1. 1 NF :The first normal form, sometimes known as the first normal form, is the most basic sort of database normalisation. The following conditions must be met for a table to be in its initial normal form:  
  
🚀 Every column should have only one value and be atomic.  
🚀 Columns from the same table that are duplicated should be eliminated.  
🚀 For each category of linked data, separate tables should be constructed, and each entry should be recognised by a distinct column.  
  
2. 2NF : The second normal form is referred to as 2NF. The following conditions must be met for a table to be in its second normal form:  
  
🚀 The table should be in its 1NF state, which means it must meet all of the 1NF requirements.  
🚀 Every non-prime attribute of the table should be entirely functionally reliant on the main key, i.e., every non-key attribute should be dependant on the primary key in such a way that if any key element is destroyed, the database will still save the non key element.  
  
3. 3NF : The third normal form is referred to as 3NF. The following conditions must be met for a table to be in its second normal form:  
  
🚀 The table should be in its 2NF state, which means it must meet all of the 2NF requirements.  
🚀 One property does not have a transitive functional dependency on any other attribute in the same table.  
  
4. BCNF :BCNF is an advanced form of 3NF that stands for Boyce-Codd Normal Form. For the same reason, it's also known as 3.5NF. The following conditions must be met for a table to be in its BCNF normal form:  
  
🚀 The table should be in its 3NF state, which means it must meet all of the 3NF requirements.  
🚀 A should be the table's super key for any functional dependency of any attribute A on B (A->B).  
🚀It simply means that if B is a prime attribute, A cannot be a non-prime attribute.

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

In a database, there are primarily seven types of keys:  
  
🚀Candidate Key: A set of features that can be used to uniquely identify a table is referred to as a candidate key. There may be numerous candidate keys in each table. A primary key can be chosen from among all candidate keys.  
🚀Super Key : The super key is a set of properties that can be used to uniquely identify a tuple. Candidate and main keys are subsets of the super key, with the super key being their superset.  
🚀Primary Key :The main key specifies a collection of characteristics that are used to uniquely identify each tuple. StudentId and firstName are candidate keys in the example below, and any of them can be used as a Primary Key.  
🚀 Unique Key: A unique key is identical to a primary key, with the exception that primary keys do not accept NULL values in columns, whereas unique keys do. Unique keys are just primary keys with NULL values.  
🚀 Alternate Key: All candidate keys that were not selected as primary keys are referred to as alternate keys.  
🚀Foreign Key: A foreign key is an attribute that can only take values from one table and is shared by an attribute in another table.  
🚀Composite Key : A composite key is a combination of two or more columns that allows each tuple in a table to be uniquely identified.

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

The 2-tier architecture : It is a client-server architecture in which client applications interface directly with the database on the server end without the use of any middleware.  
For instance, a Contact Management System built in MS-Access or a Railway Reservation System, for example.  
  
  
  
The 3-tier design : It adds a layer between the client and the server to give users a graphical user interface and make the system more safe and accessible. In this form of design, the client-side application communicates with a server-side application, which then communicates with the database system.  
Designing a registration form with a text box, label, button, or a large website on the Internet, as an example.



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

|  |  |
| --- | --- |
| **Logical database design** | **Physical database design** |
| The conceptual schema (or an ER schema) from the high-level data model is mapped or transformed into a relational database structure. | The specifications for the stored database are created, including physical storage structures, record placement, and indexes. |
| The mapping process can be divided into two stages:  🚀 Data model-dependent but system-independent mapping 🚀 Adapting the schemas to a certain database management system | The criteria listed below are frequently used to inform the selection of physical database design options:  🚀 Space Utilization 🚀Response Time 🚀 Throughput of transactions |
| DDL statements in the chosen DBMS language that define the database system's conceptual and external level schemas. However, if the DDL statements incorporate physical design parameters, the DDL specification must be performed after the physical database design step is completed. | An first determination of database file storage architecture and access pathways. In terms of Data Storage Definition Language, this relates to defining the internal schema. |

There are various stages to database design. Two of them are logical database design and physical database design. This separation is usually based on the three-level design of a database management system (DBMS), which ensures data independence. As a result, we may argue that this separation leads to data independence since the logical database design's output is the database system's conceptual and external level schemas, which are independent of the physical database design's output, which is an internal schema.

**71. What are temporary tables? When are they useful?**

Temporary tables are tables that are only used for a single session or whose data is only kept for the duration of the transaction. Temporary tables are most commonly utilised to support unique rollups or specific application processing needs. A temporary table does not have space given to it when it is formed, unlike a permanent table. As rows are added, space for the table will be dynamically allotted. To build a temporary table in Oracle, use the CREATE GLOBAL TEMPORARY TABLE command.

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

Entity type extension is the process of combining comparable entity types into a single type that is then grouped together as an entity set.

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

The first step in the database design process is conceptual design. At this stage, the goal is to create a database that is unaffected by database software or physical details. A conceptual data model describes the primary data items, properties, relationships, and constraints of a specific issue domain as a result of this procedure.

**74. Explain different types of failures that occur in the Oracle database.**

In the Oracle database differnt types of failures can occur, some of them are given below:  
  
🚀Bad data type  
  Insufficient space 🚀 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  
🚀 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  
🚀 Statement Failure·  
🚀 Insufficient Privileges

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

The acronym RAID stands for Redundant Array of Inexpensive (or "Independent")Disks.  
  
RAID is a technique for merging many hard drives into a single logical unit (two or more discs grouped together to appear as a single device to the host system). RAID was created to overcome the fault-tolerance and performance limitations of traditional disc storage. It can provide greater fault tolerance and throughput than a single hard disc or a collection of independent hard drives. Arrays were originally thought to be complex and specialised storage solutions, but they are now simple to use and critical for a wide range of client/server applications.

**76. Explain Database partitioning and its importance.**

The process of splitting a logical database into independent components to improve availability, performance, and manageability is known as data partitioning.  
The following are some of the reasons why database partitioning is important:  
  
🚀 Allows you to access significant portions of a partition.  
🚀 Data can be stored on low-cost, slower storage.  
🚀 Improves the speed of queries

**77. Explain the functionality of DML Compiler.**

The DML Compiler converts DML statements into query language that can be understood by the query evaluation engine. Because DML is a set of grammar elements that is quite similar to other programming languages that require compilation, a DML compiler is necessary. As a result, it's critical to compile the code in a language that the query evaluation engine understands, and then work on the queries with the appropriate output.

**78. What is Relational Algebra?**

Relational Algebra is a Procedural Query Language that includes a collection of operations that accept one or two relations as input and output a new relationship. The relational model's basic set of operations is known as relational algebra. The key feature of relational algebra is that it resembles algebra that operates on numbers.  
Relational algebra has only a few basic operations:  
  
🚀 set difference  
🚀 project  
🚀 select  
🚀 union  
🚀rename,etc.

**79. What is Relational Calculus?**

Relational Calculus is a non-procedural query language that instead of algebra employs mathematical predicate calculus. Relational calculus does not work with math essentials like algebra, differential equations, integration, and so on. It's also known as predicate calculus because of this.  
  
Relational calculus is divided into two types:  
🚀 Relational calculus in a tuple  
🚀 Calculus of Domain Relationships

**80. What do you mean by durability in DBMS?**

Even if the system fails before all of the changes are reflected on disc, the effect of a transaction should endure once the DBMS alerts the user that it has completed successfully. Durability is the term for this quality. Durability guarantees that once a transaction is committed to the database, it will be stored in non-volatile memory, where it will be safe against system failure.

**81. What is System R? How many of its two major subsystems?**

From 1974 through 1979, IBM San Jose Research Centre conceived and built System R. System R was the first to demonstrate that a relational database management system (RDBMS) could give higher transaction processing performance, and it was also the first to implement SQL, the standard relational data query language. It's a working prototype that demonstrates how to create a Relational System that can be utilised in a real-world setting to address real-world problems.  
  
Two primary subsystems of System R are as follows:  
🚀 Relational Data System  
🚀 Research Storage System

**82. How do you communicate with an RDBMS?**

To interface with the RDBMS, you must utilise Structured Query Language (SQL). We can offer input to the database using SQL queries, and the database will provide us with the required output when the queries have been processed.

**83. What do you understand by Proactive, Retroactive and Simultaneous Update?**

🚀Proactive Update: These changes are made to the database before it is put into use in the actual world.  
🚀 Retroactive Updates: These updates are applied to a database after it has been operational in the real world.  
🚀 Simultaneous Update: These updates are applied to the database at the same moment as they become effective in the actual world.

**84. Explain the terms specialization and generalization**

🚀Specialization is the process of defining a set of subclasses for a particular entity type. Each subclass will contain all of the parent entity's characteristics and relationships. Aside from that, subclasses may have additional attributes and relationships that are unique to them.  
🚀 Generalization is the process of identifying relationships and common attributes among a group of entities, and then defining a common superclass for them.

**85. What does Fill Factor concept mean with respect to indexes?**

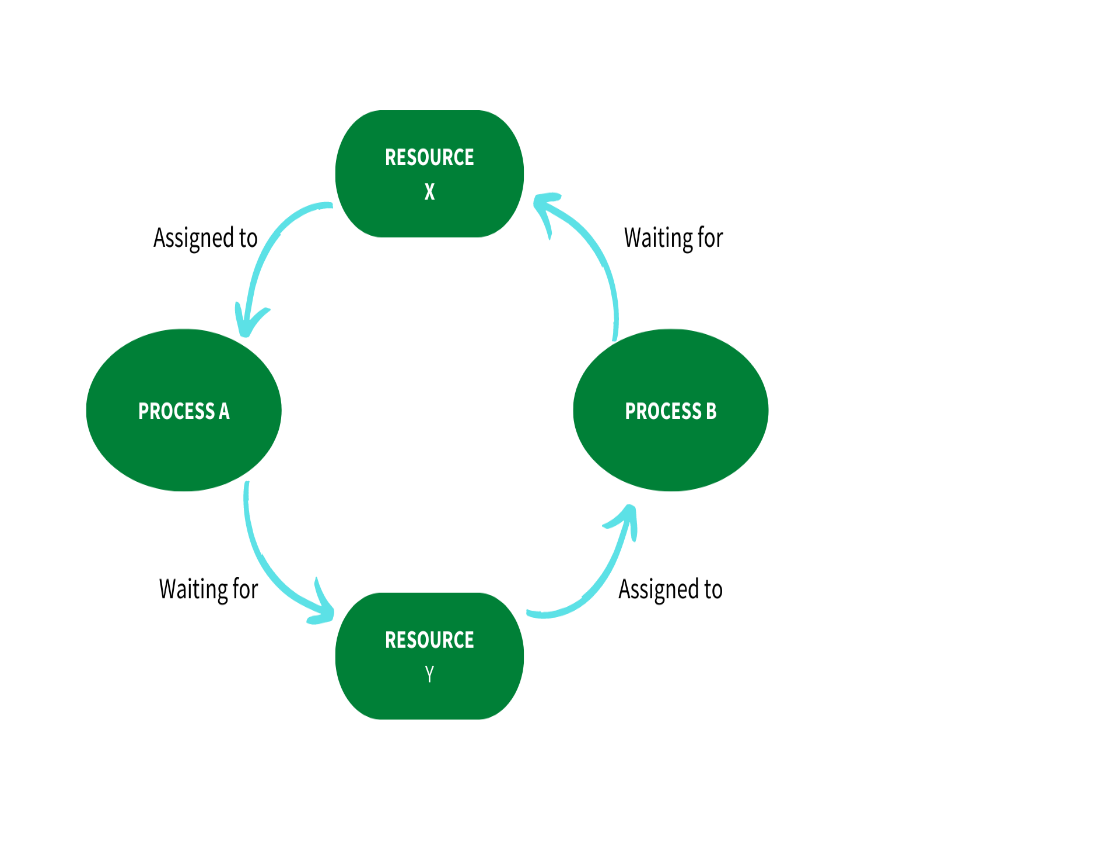
The term "fill factor" refers to the percentage of space left on each leaf-level page that is densely packed with information. The default number is usually 100.

**86. What is Index hunting and how does it help in improving query performance?**

Index hunting is the process of enhancing a collection of indices. This is done because indexes improve query performance as well as query processing time.  
  
It aids in query performance improvement in the following ways:  
🚀 Using the query optimizer, the optimal queries are suggested.  
🚀 To check the effect, measurements such as index, query distribution, and performance are used.  
🚀 Databases are optimised for a small number of problem queries.

**87. Explain what is a deadlock and mention how it can be resolved?**

A deadlock occurs when two transactions wait for a resource that is unavailable or while another operation is halted. Deadlocks can be avoided by requiring all transactions to acquire all locks at the same time. As a result, the only way to break the deadlock is to abort one of the transactions and remove the partially completed work.



**88. Under what conditions should you use an index?**

When you want to ensure that a database is unique, you can use an index. It can also be used to simplify sorting and retrieve data quickly. A frequently used column may be an excellent candidate for an index.

**Q #27) What are different types of joins in SQL?**

**Answer: 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 the 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 put and the rows which do not match hold null values.