SQL Questions

Why a database is called as relational database model?

A database model represents the relationship between one or more databases. The relationship is known as the
relational database model. It is an extension of the normal databases without relations. It provides flexibility
and allows one database to be in relation with another database. It can access the data from many databases at
one time over the network.

What are entities and attributes referring to?

- Table consists of some properties that are known as attributes.
- These consist of the representation of entity in the table.
- They are represented by columns in the table.
- Entity is referred to the store data about any particular thing.
- It is the smallest unit inside the table.

What do you understand by relation in relational database model?

Relation in the relational database model is defined as the set of tuples that have the same attributes.
Tuple represents an object and also the information that the object contains. Objects are basically instances
of classes and used to hold the larger picture. Relation is described as a table and is organized in rows and
columns. The data referenced by the relation come in the same domain and have the same constraints as well.
Relations in the relational database model can be modified using the commands like insert, delete etc.

Why domain is of high importance?

- Domain describes possible values grouped together that can be given for an attribute. It is considered the same way as a constraint on the value of attribute.

- A domain can be attached to an attribute but only if the attribute is an element of specified set.

For example: XYZ doesn’t fulfill the domain constraint but the integer value as 899 fulfills the criteria of domain constraint. Hence, domain is of high importance.

What is the difference between base and derived relation?

- Relational database means the relationship between different databases. In relational database user can store and access all the data through the tables which are related to each other.
- Relationship between the store data is called base relations and implementation of it is called as tables. Whereas, relations which don’t store the data, but can be found out by applying relational operations on other relations are called as derived relations. When these are implemented they are termed as views or queries.
- Derived relations are more useful then base relation, as they can have more information from many relations, but they act as a single relation.

What are constraints in database?

Constraints are kind of restrictions that are applied to the database or on the domain of an attribute. For example an integer attribute is restricted from 1-10 and not more than that. They provide the way to implement the business logic and the rules in database. In database it can be implemented in the form of check constraints that checks for the rules that haven’t been followed by the programmer. Constraint also used to restrict the data that can be stored in the relations. Domain constraint can be applied to check the domain functionality and keep it safe..

What are the two principles of relational database model? What is the difference between them?

The two principal rules for the relational model are as follows:

- Entity integrity: this is used to maintain the integrity at entity level
- Referential integrity: it is used to maintain integrity on all the values which have been referenced.

The differences between them are as follows:

- Entity integrity tells that in a database every entity should have a unique key; on the other hand referential integrity tells that in the database every table values for all foreign keys will remain valid.
- Referential integrity is based on entity integrity but it is not the other way around.
- For example: if a table is present and there is a set of column out of which one column has parent key set then to ensure that the table doesn't contain any duplicate values, a unique index is defined on the column that contains the parent key.

What is the difference between primary and foreign key?

- Primary key uniquely identify a relationship in a database, whereas foreign key is the key that is in other relation and it has been referenced from the primary key from other table.
- Primary key remains one only for the table, whereas there can be more than one foreign key.
- Primary key is unique and won't be shared between many tables, but foreign key will be shared between more than one table and will be used to tell the relationship between them.

Why stored procedures are called as executable code?

Stored procedure stored inside the database. This also includes the executable code that usually collects and customizes the operations like insert, encapsulation, etc. These stored procedures are used as APIs for simplicity and security purposes. The implementation of it allows the developers to have procedural extensions to the standard SQL syntax. Stored procedure doesn't come as a part of relational database model, but can be included in many implementations commercially.

What is an index represent in relational database model?

- Index is a way to provide quick access to the data and structure. It has indexes maintain and can be created to combine attributes on a relation. Index allows the queries to filter out the searches faster and matching data can be found earlier with simplicity.
- For example: It is same as the book where by using the index you can directly jump to a defined section. In relational database there is a provision to give multiple indexing techniques to optimize the data distribution.

What are the relational operations that can be performed on the database?

There are many relational operators that are used to perform actions on relational database. These operators are as follows:

1. Union operator that combines the rows of two relations and doesn't include any duplicate. It also removes the duplicates from the result.
2. Intersection operator provides a set of rows that two relations have in common.
3. Difference operator provide the output by taking two relations and producing the difference of rows from first that don't exist in second.
4. Cartesian product is done on two relations. It acts as a cross join operator.

What do you understand by database Normalization?

- Normalization is very essential part of relational model.
- Normal forms are the common form of normalization.
- It helps in reducing redundancy to increase the information overall.
- It has some disadvantages as it increases complexity and have some overhead of processing.
- It consists of set of procedures that eliminates the domains that are non-atomic and redundancy of data that prevents data manipulation and loss of data integrity.

What are the different types of normalization that exists in the database?

There are 9 normalizations that are used inside the database. These are as follows:

1. First normal form: in this table represents a relation that has no repeating groups.
2. Second normal form: non- prime attributes are not functional dependent on subset of any candidate key.
3. Third normal form: in a table every non- prime attribute is non-transitively dependent on every candidate key
4. Elementary key normal form: superkey dependency or elementary key dependency effects the functional dependency in a table.

5. Boyce codd normal form: “every non-trivial functional dependency in the table is dependent on superkey”.
6. Fourth normal form: “Every non-trivial multivalued dependency in the table is a dependent on a superkey”.
7. Fifth normal form (5NF): “Every non-trivial join dependency in the table is implied by the superkeys of the table”.
8. Domain/key normal form (DKNF): “Every constraint on the table is a logical consequence of the table's domain constraints and key constraints”.
9. Sixth normal form (6NF): “Table features no non-trivial join dependencies at all”.

How de-normalization is different from normalization?

- Analytical processing databases are not very normalized. The operations which are used are read most databases.
- It is used to extract the data that are ancient and accumulated over long period of time. For this purpose de-normalization occurs that provide smart business applications.
- Dimensional tables in star schema are good example of de-normalized data.
- The de-normalized form must be controlled while extracting, transforming, loading and processing.
- There should be constraint that user should not be allowed to view the state till it is consistent.
- It is used to increase the performance on many systems without RDBMS platform.

What is the type of de-normalization?

Non-first normal form (NFA)

– It describes the definition of the database design which is different from the first normal form.

- It keeps the values in structured and specialized types with their own domain specific languages.
- The query language used in this is extended to incorporate more support for relational domain values by adding more operators.

How many levels of data abstraction are available?

There are three levels of data abstraction available in database model and these are as follows:

1. Physical level: It is the lowest level that describes how data is stored inside the database.
2. Logical level: It is the next higher level in the hierarchy that provides the abstraction. It describes what data are stored and the relationship between them.
3. View level: It is the highest level in hierarchy that describes part of the entire database. It allows user to view the database and do the query.

What is the difference between extension and intension?

The major difference between extension and intension is that:

- Extension is time dependent, whereas intension includes a constant value.
- Extension tells about the number of tuples presented in a table at any instance, whereas intension gives the name, structure and constraint of the table.

What are its two major subsystems of System R?

System R is being developed by IBM. Its purpose is to demonstrate the possible solution to build a relational database system. The relational database system has to be such that which can interact with the real life environment to sole real life scenarios.

The two subsystems that are included in it are:

1. Research storage: This includes the research information of the database.
2. System relational system: This includes the relational data that a system has to produce and keep everything in relation.

What do you understand by Data Independence?

Data independence tells about the independence of the data inside the application. It usually deals with the storage structure and represents the ability to modify the schema definition. It doesn’t affect the schema definition which is being written on the higher level.

There are two types of data independence:

1. Physical data independence: It allows the modification to be done in physical level and doesn't affect the logical level.
2. Logical data independence: It allow the modification to be done at logical level and affects the view level.

NOTE: Logical Data Independence is more difficult to achieve.

How view is related to data independence?

- View is a virtual table that doesn't really exist, but it remains present so that user can view their data.
- It is derived from the base table. The view is stored in the data dictionary and represents the file directly.
- The base table updation or reconstruction is not being reflected in views.
- It is related to the logical data independence as it is at the logical level and not at the physical level.

Why E-R models are used?

E-R model stands for entity-relationship model and it is used to represent a model with their relationships. This is an object oriented approach and it is based on real world that consists of objects which are called entities and relationship between them. Entities are further used inside the database in the form of attributes.

What is the purpose of acid properties?

- ACID stands for Atomicity, Consistency, Isolation and durability and it plays an important role in the database.
- These properties allow the database to be more convenient to access and use. This allows data to be shared more safely in between the tables.
- If these properties are not being implemented then the data will become inconsistent and inaccurate.
- It helps in maintaining the accuracy of the data in the database.

What do you understand by cardinality and why it is used?

- Cardinality is important and used to arrange the data inside the database.
- It is related to the design part and need to be properly used in database.

- It is used in E-R diagrams and used to show the relationship between entities/tables.
- It has many forms like the basic is one to one, which associate one entity with another.
- Second is one to many: which relates one entity with many entities in a table.
- Third is many to many M: N that allows many entities to be related to many more.
- Last is many to one that allows the many entities to be associated with one entity.

What is the difference between DBMS and RDBMS?

- DBMS is persistent and accessible when the data is created or exists, but RDBMS tells about the relation between the table and other tables.
- RDBS supports a tabular structure for data and relationship between them in the system whereas DBMS supports only the tabular structure.
- DBMS provide uniform methods for application that has to be independently accessed, but RDBMS doesn’t provide methods like DBMS but provide relationship which link one entity with another.

What is Index?

- A pointer to data having physical representation is called as Index.
- Record can be located quickly and efficiently by creating Indices on existing tables.
- Each index in a table has some valid name and we can have more than one index in different columns of a table.
- We can speed up queries by setting up index in a column of a table.
- In a table , each row is examined by sql server to fulfil our query is known as table scan and it only happen when there is no index available to help the query.
- On large tables, the table scan has huge impact on performance.
- Clustered and Non clustered indexes are the most widely used indexes in a database.

What is Trigger?

- A Trigger is a process of firing an action when some event like Insert, Update or Delete occurs.
- A trigger can’t be called or even executed rather they are automatically become active by the DBMS whenever some modification in associated table occur.
- Triggers are event driven and can attached to particular table in a database.
- Triggers are implicitly executed and stored procedures are also executed by triggers.
- Referential integrity is maintained by the trigger and they are managed and stored by DBMS.
- Triggers can be nested also, in which Insert, Update or Delete logic can be fired from the trigger itself.

What is a NOLOCK?

- NOLOCK is used to improve concurrency on a busy system.
- On data read, no lock can be taken on SELECT statement.
- When some other process is updating the data on the same time you are reading it is known as dirty read.
- Read (Shared) locks are taken by SELECT Statements.
- Simultaneous access of multiple SELECT statements is allowed in Shared lock but modification process is not allowed.
- The result to your system is blocking.
- Update will start on completion of all the reads.

What is the STUFF function and how does it differ from the REPLACE function?

- Using STUFF function we can overwrite the specified characters of a string.
  The syntax of STUFF function is:
  STUFF (stringToChange, startIndex, length, new_characters)

where stringToChange is the string which will have the characters those we want to overwrite, startIndex is the starting position, length is the number of characters in the string that are to be overwrited, and new_characters are the new characters to write into the string.

- While REPLACE function is used to replace specified character at all its existing occurrences.
- The syntax of REPLACE function is REPLACE (string_to_change, string_to_Replace, new_tring).
- Every occurrence of string_to_change will be replaced by new_string.

What are Self Join and Cross Join?

- When we want to join a table to itself then SELF JOIN is used.
- We can give one or more aliases to eliminate the confusion.
- A self join can be used as any type, if both the tables are same.
- The simple example where we can use SELF JOIN is if in a company have a hierarchal reporting structure and an employee reports to another.
- A cross join give the number of rows in the first table multiplied by the number of rows in second table.
- The simple example where we can use CROSS JOIJ is if in an organization wants to combine every Employee with family table to see each Employee with each family member.

What are the advantages of using Stored Procedures?

- Procedure can reduce network traffic and latency, and can enhance application performance.
- Procedure execution plans can be reused, staying cached in the management tool's memory, reducing its overhead.
- Procedures provide the benefit of code reuse.
- The logic can be encapsulated using procedures and can help to change procedure's code without interacting to application.
- Procedures give more security to our data.

What is RANK function?

- RANK function can be used to give a rank to each row returned from a SELECT statment.
- For using this function first specify the function name, followed by the empty parentheses.

- Then mention the OVER function. For this function, you have to pass an ORDER BY clause as an argument. The clause identifies the column on which you are going to apply the RANK function.

For Example:

```
SELECT RANK() OVER(ORDER BY BirthDate DESC) AS [RowNumber], FirstName, BirthDate FROM EmpDetails
```

- In the result you will see that the eldest employee got the first rank and the youngest employee got the last rank. Here the rows with equal age will get same ranks.
- The rank depends on the row's position in the result set, but not on the sequential number of the row.

What are cursors and when they are useful?

- When we execute any SQL operations, SQL Server opens a work area in memory which is called Cursor.
- When it is required to perform the row by row operations which are not possible with the set-based operations then cursor is used.

There are two of cursors:

1. Implicate Cursor

- SQL Server automatically manages cursors for all data manipulation statements. These cursors are called implicit cursors.

1. Explicit Cursor

- When the programmer wants to perform the row by row operations for the result set containing more than one row, then he explicitly declare a cursor with a name.
- They are managed by OPEN, FETCH and CLOSE.
- %FOUND, %NOFOUND, %ROWCOUNT and %ISOPEN attributes are used in both types of cursors.

What is Similarity and Difference between Truncate and Delete in SQL?

- Similarity

- Both Truncate and Delete command will delete data from given table and they will not delete the table structure from the database.

- Difference

1. TRUNCATE is a DDL (data definition language) command whereas DELETE is a DML (data manipulation language) command.

2. We can’t execute a trigger with TRUNCATE whereas with DELETE command, a trigger can be executed.

3. We can use any condition in WHERE clause using DELETE but it is not possible with TRUNCATE.

4. If table is referenced by any foreign key constraints then TRUNCATE cannot work.

5. TRUNCATE is faster than DELETE, because when you use DELETE to delete the data, at that time it store the whole data in rollback space from where you can get the data back after deletion, whereas TRUNCATE will not store data in rollback space and will directly delete it. You can’t get the deleted data back when you use TRUNCATE.

What are COMMIT and ROLLBACK in SQL?

- COMMIT statement is used to end the current transaction and once the COMMIT statement is exceucted the transaction will be permanent and undone.

- Syntax: COMMIT;

- Example:

```
BEGIN
UPDATE EmpDetails SET EmpName = ‘Arpit’ where Dept = ‘Developer’
COMMIT;
END;
```

- ROLLBACK statement is used to end the current transaction and undone the changes which was made by that transaction.

- Syntax: ROLLBACK [TO] Savepoint_name;

- Example:

```
BEGIN
Statement1;
```

```
SAVEPOINT mysavepoint;
BEGIN
Statement2;
EXCEPTION
WHEN OTHERS THEN
ROLLBACK TO mysavepoint;
Statement5;
END;
END;
```

What is a WITH(NOLOCK)?

- WITH(NOLOCK) is used to unlock the data which is locked by the transaction that is not yet committed. This command is used before SELECT statement.
- When the transaction is committed or rolled back then there is no need to use NOLOCK function because the data is already released by the committed transaction.
- Syntax: WITH(NOLOCK)
- Example:

```
SELECT * FROM EmpDetails WITH(NOLOCK)
WITH(NOLCOK) is similar as READ UNCOMMITTED.
```

What is difference between Co-related sub query and nested sub query?

- Correlated subquery executes single time for every row which is selected by the outer query.
- It has a reference to a value from the row selected by the outer query.
- Nested subquery executes only once for the entire nesting (outer) query. It does not contain any reference to the outer query row.
- For example,
- Correlated Subquery:
  select e.EmpFirstName, e.Salary, e.DeptId from Employee e where e.Salary = (select max(Salary) from Employee ee where ee.DeptId = e.DeptId)
- Nested Subquery:
  select EmpFirstName, Salary, DeptId from Employee where (DeptId, Salary) in (select DeptId, max(Salary) from Employee group by DeptId)

Differentiate UNION, MINUS, UNION ALL and INTERSECT?

- INTERSECT - It will give all the distinct rows from both select queries.
- MINUS - It will give distinct rows returned by the first query but not by the second query.
- UNION - It will give all distinct rows selected by either first query or second query.
- UNION ALL - It will give all rows returned by either query with all duplicate records.

What is a join? Explain the different types of joins?

Using Join in a query, we can retrieve referenced columns or rows from multiple tables.

Following are different types of Joins:

1. JOIN: Return details from tables if there is at least one matching row in both tables.
2. LEFT JOIN: It will return all rows from the left table, even if there are no matching row in the right table.
3. RIGHT JOIN: It will return all rows from the right table, even if there is no matching row in the left table.
4. FULL JOIN: It will return rows when there is a match in either of tables.

What is DDL, DML and DCL?

SQL commands can be divided in three large subgroups.

1. DDL: The SQL commands which deals with database schemas and information of how the data will be generated in database are classified as Data Definition Language.
   -For example: CREATE TABLE or ALTER TABLE belongs to DDL.
2. DML: The SQL commands which deals with data manipulation are classified as Data Manipulation Language.
   For example: SELECT, INSERT, etc.
3. DCL: The SQL commands which deal with rights and permission over the database are classified as DCL.
   For example: GRANT, REVOKE

What is Index tuning?

- Query performance as well as speed improvement of a database can be done using Indexes.
- The process of enhancing the selection of indexes is called Index Tuning.

What is Index tuning?

Index tuning is part of database tuning for selecting and creating indexes. The index tuning goal is to reduce the query processing time. Potential use of indexes in dynamic environments with several ad-hoc queries in advance is a difficult task. Index tuning involves the queries based on indexes and the indexes are created automatically on-the-fly. No explicit actions are needed by the database users for index tuning.

How is index tuning used to improve query performance?

The Index tuning wizard can be used to improve the performance of queries and databases. It uses the following measures to do so:

- It uses the query optimizer to perform the analysis of queries with respect to the workload and based on this knowledge, it recommends the best usage of indexes.
- The changes in the usage of index, query distribution and their performance are analysed for checking the effect.
- It also recommends ways of tuning the database for a small set of problem queries.

How is index tuning used to improve query performance?

- Index tuning improves query performance by using Index Tuning Wizard. SQL profiler is used for capturing a trace of the activity, for the optimizing performance. The trace can be extended for a period of time for the purpose of capturing a wide range of activity.
- Subsequently, Enterprise manager is used for starting the Index Tuning Wizard and instructs to recommended indexes which are based on the trace that is captured. An estimation of increased performance after making changes is provided apart from appropriate columns suggestion.

Reasons of poor performance of query.

Following are the reasons for the poor performance of a query:

- No indexes.
- Excess recompilations of stored procedures.
- Procedures and triggers without SET NOCOUNT ON.
- Poorly written query with unnecessarily complicated joins.
- Highly normalized database design.
- Excess usage of cursors and temporary tables.
- Queries with predicates that use comparison operators between different columns of the same table.
- Queries with predicates that use operators, and any one of the following are true:
- There are no statistics on the columns involved on either side of the operators.
- The distribution of values in the statistics is not uniform, but the query seeks a highly selective value set. This situation can be especially true if the operator is anything other than the equality (=) operator.
- The predicate uses the not equal to (!=) comparison operator or the NOT logical operator.
- Queries that use any of the SQL Server built-in functions or a scalar-valued, user-defined function whose argument is not a constant value.
- Queries that involve joining columns through arithmetic or string concatenation operators.
- Queries that compare variables whose values are not known when the query is compiled and optimized.SQL stands for