

Assignment 1

1. Describe the purpose of normalizing data

The purpose of normalizing data is to remove anomalies so that the analysis of the data can be made more streamlined. Other than that normalizing of data can also help to improve data visualization other than that if we normalize the data the total space taken by the database is also reduced and the performance is increased. The normalization also removes unnecessary information furthermore updating the database is also easier after normalization. Other than that data redundancy is reduced, and data integrity is improved by normalization.

2. Describe the concept of functional dependency

A functional dependency is used to do the normalization of the database. When in a table an attribute uniquely identifies another attribute in the same table then those attributes are called functionally dependant on each other. For example, we have NUID which can uniquely determine the two students who have the same name then we can say the student name is functionally dependent on NUID in the database.

3. Describe the concept of transitive dependency and describe how this concept relates to 3NF. Provide an example to illustrate your answer.

The transitive dependency can occur when an attribute defines other attribute and that attribute define another than transitive dependency is formed btw those attribute.

For example, we have attributes X, Y, Z. $X \rightarrow Y$, and $Y \rightarrow Z$, hence we can say there is transitive dependency btw X and Z

In 3nf also called the third normal form is a relation which is in 1nf and 2nf in which none of the non-primary key attributes has a transitive dependency with the primary key.

4. Describe the approach to converting a UNF table to 1NF relation(s)

The Unnormalized form also called non first normal form NF^2 can be converted into a 1NF form using several different approaches

We first have to identify the primary key

Then we can either split the multiple values into the separate tuples

Another way is we can add a different column for the repeating group.

5. The second normal form (2NF) is realized by removing partial dependencies from 1NF relations. Briefly describe the term partial dependency

To achieve normalization, we should have a minimum no. of attributes in determinants that are used to maintain functional dependency with right-hand side attributes. These requirements here are called full functional dependency.

In full functional dependency, x and y are attributed in the relation where y is fully dependent on X only if y is functionally dependent on X, but Y is not dependent on a subset of X.

In full functional dependency if we remove an attribute from X, the result will cause the dependency to be non-existent.

6. What are Anomalies in Relational Model and how do overcome them?

There are generally three types of anomaly

Insertion anomaly:- this anomaly can occur when the new value is inserted in the database at a later point in time this could to a null value in the attribute of the database and hence cause the anomaly.

For example, a worker added to a company there is a column in a database about workers like his age salary address, and performance report but the new worker which was joined later will have a null value in his performance report. Thus causing anomaly

Update anomaly:- this anomaly can occur when a piece of information In a dataset is updated suppose there are 5 columns and for a student, now student changes his address this would lead to an error in a database if the data is not normalized.

Deletion Anomaly

Deletion anomaly can occur when data is removed from the table. This can lead to unnecessary deletion of information that is important for example an employee leave the company do his information is to be removed now if info about his project of also deleted I would be a loss of company as project info should be available to a company for future updating purposes.

We can solve these anomalies by the process of normalization where we can divide data into multiple tables and form keys between them using multiple keys.

7. What is De-normalization and how it is helpful and explain in your own words how it's different than Normalization

Denormalization is the process, which is different from normalization, in Denormalization multiple tables are combined into one single table. So that query can be applied to it in a faster way.

Normalization aims to reduce data redundancy whereas in denormalization faster query execution aimed to be achieved by introducing redundancy

In normalization memory is optimized whereas in denormalization wastage of memory is introduced. Unlike normalization, denormalization does not maintain data integrity.

8. Explain in your own words the below terms – (2M each)

1. Consistency

Consistency means the data in the database must follow all rules across different tables across all tables such as constraints, cascades, triggers, etc.

2. Concurrency

In database n no. of the user should be able to perform n no. of transactions in the database and the database should be able to support it.

3. Reliability

Database reliability is the amalgamation of different terms which generally means data should be reliable and should be able to perform the task without any problems

A reliable database must have

Data Integrity

Meaning data in the database should be accurate and should be consistent throughout the entire database.

Data recoverability

The database should be recoverable in case of some mishaps such as data loss. Or mock data lost drill.

4. Durability

The transaction committed in the database should be able to survive permanently in the database such as if the user books a movie ticket that transaction should not be deleted in case of a mishap such as a system crash.

5. Relation (Entity)

An entity is an object in the database which could be a place, person, unit, etc., and can be stored in the form of a table, workflow, and properties.

(There are a different kind of relationship btw different entity)

They are:-

a) One to One Relationship

In one to one such as person and driver license has on to one relation

b) One to Many

In one to many, a customer can have both saving and checking account at the bank

c) Many to many

Many to many relationships can be formed btw student and subject where multiple students have the same subject and vice versa.)

9. Explain below -

1. What is EF Cood's RULE Zero? Explain EF Cood's 12 Rules. (5M)

EF code rule is set of rule which can be followed in Relational Database Management System. To better manage the database

The 12 rules are:-

0 -foundation rule:-The database should store data with the help of relational capabilities

1 – Information rule tr: - the data stored in the database should be in form of relations and value can be stored as the entity in table cells.

2 – Guaranteed Access: - the data in the database should be able to access without pointers, data entity could be accessed by the table, primary key, etc.

3 Systematic treatment of Null Value

In the Relation database, the null value should be supported and they should be accessible by the user and they should not be dependent on any data type.

4 Online Active Catalog

There should be a database dictionary that should describe the structure of the database and should be stored online

5 Powerful and Well-Structured Language

The language used in the database should be well-structured robust language and should provide access to data in a database. For example SQL.

6: View Updating Rule

The view created inside the database that can be updatable by a user should be updatable by the system as well.

7: Relational Level Operation

There should be commands in the database like Insert, update, delete, and other operation such as intersection and unions

8: Physical Data Independence

The updating in physical data storage should not affect the system suppose a table is stored in file X when that file is renamed to file y it should not affect the database

9: Logical Data Independence

If the logical structure of a database is changed the view of that database should not be changed.

10: Integrity Independence

The database should be able to form its integrity principle with the help of keys, trigger, etc. and store that in a dictionary

Rule 11: Distribution Independence

Suppose database is stored at different places like azure cloud DB which stores data at multiple locations now if the different component of data is stored at different places it should not affect data performance.

Rule 12: Nonsubversion Rule

If the database can be accessed from a low level it should not compromise the data integrity rules of the database.

2. What is the purpose of the like operator in SQL? (5M)

Like operator in SQL is one of the advanced functions of SQL which help in the management of data. Like operator is used in where clause to find a pattern in an attribute.

There are wild cards in like operator % and _ which help in finding data.

%- This symbol represents zero to multiple characters to find

_ this symbol represents only one character find

There is a certain variation of like a character like

In X% it will show value starting with Z

In %X it will show value ending with a

In %x% find the value that has x in the middle

Same rule with x_

The like command is used with code

WHERE Colln LIKE pattern.

10. What are the functions of Client-server Architecture and what is 3 tier architecture?

client-server architecture in a computer network has many clients that are also called remote processors which do request to a centralized server and receive services in return. In other words, we can say that client will request info of the database from the server on the net and responds.

We have 3 types of client-server architecture.

They are

- A. Tier 1
- B. Tier 2
- C. Tier 3

3 tier Architecture

3 tier architecture is the most commonly used in this architecture there are 3 layers known as

Client layer – this layer request information from the server

Application layer – This layer act as a middleman btw two it has functions such as security, logic, etc.

Database server layer- this is the final layer that store all the data in the database

The advantage of tier architecture is that different tiers can run individually on their infrastructure and work can be done on the same project by different teams, along with it scaling and updating of project work can be performed easily with this model. Tier 3 architecture is useful for cloud services such a dockers and microservices