1. What is normalization?

It is a process of structuring a relational database in accordance with a series of so-called normal forms in order to reduce data redundancy. It does not completely eliminate data redundancy, but instead, controls redundancy and helps properly link database tables.

1. When is a table in 1NF?

A table is in 1NF when the following are accomplished:

* 1. All key attributes are defined, and remaining attributes are dependant on the PK
  2. There are no repeating groups in the table.
  3. All attributes are dependent on the primary key.

1. When is a table in 2NF?

A table is in 2NF when the following are accomplished:

* 1. It is in 1NF
  2. It includes no partial dependencies; therefore, no attribute is dependent on only a portion of the primary key.

1. When is a table in 3NF?

A table is in 3NF when the following are accomplished:

* 1. It is in 2NF
  2. it contains no transitive dependencies.

1. What is a partial dependency? With what normal form is it associated?

A partial dependency happens when an attribute is dependant on only a part of the PK; it is associated with 1NF.

1. What three data anomalies are likely to be the result of data redundancy? How can such anomalies be eliminated?

Three data anomalies that are likely to be the result of data redundancy are:

* 1. Insertion
  2. Update
  3. Deletion

These can be eliminated by normalizing the table structure, splitting up tables to divide the info into separate relational groups can reduce redundancy.

1. Define and discuss the concept of transitive dependency.

Whenever some indirect relationship happens to cause functional dependency (FC), it is known as Transitive Dependency. Thus, if A -> B and B -> C are true, then A -> C happens to be a transitive dependency. Thus, to achieve 3NF, one must eliminate the Transitive Dependency.

Ex:

Cus\_Num, Cus\_name, Cus\_adress, **agent\_num, agent\_name, agent\_address**

In the example table the attributes in **BOLD** have a transitive dependency to the underlined attribute. The tables can be separated to eliminate transitive dependency.

1. How do you remove transitive dependency?

We remove the transitively dependent attribute(s) from the relation by placing the attribute(s) in a new relation along with a copy of the determinant.

Using the example from used in the previous question:

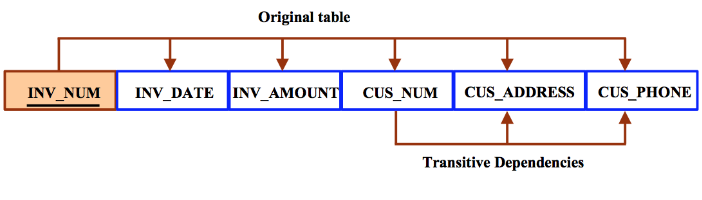
Cus\_Num, Cus\_name, Cus\_adress, agent\_num, agent\_name, agent\_address 🡪 1 Table

Becomes:

Cus\_Num, Cus\_name, Cus\_adress 🡪 Table 1

Cus\_num, agent\_num, agent\_name, agent\_address 🡪 Table 2

1. Remove the transitive dependency of the following table:



Becomes:

Inv\_num, inv\_date, inv\_amount, cus\_num

cus\_num, cus\_address, cus\_phone

1. What is a surrogate key, and when should you use one?

A surrogate key is an artificial PK created by the designer to make simplifying the assignment of primary keys to tables. Surrogate keys are usually numeric and are automatically generated by the DBMS.

1. Why is a table whose primary key consists of a single attribute automatically in 2NF when it is

in 1NF?

If the PK is a single attribute there can be no partial dependencies. A dependency based on only a part of a composite PK, is called a partial dependency

1. How would you describe a condition in which one attribute is dependent on another attribute when neither attribute is part of the primary key?

An attribute that depends on another attribute and neither attribute is part of the primary key is called a transitive dependency.