**Assignment - 2**

**Q1. Describe the purpose of normalizing data, functional dependency by taking your project domain as example.**

**Ans.** Normalization is the process of mitigating data redundancy through the use of additional attributes or columns that are added to an entity to define table structures properly which helps in reducing or removing duplicate data. This process can help in mitigating or eliminating data inconsistency and can also help database users to efficiently use the given storage as well. The process of normalization is usually implemented after defining entities, attributes and functional dependencies

Functional Dependency is a relationship between different attributes which is defined by their respective values. In this relation, attributes can be defined as determinants if they determine values of other attributes, or they can also be defined as dependents if their values are determined by a determinant attribute.

Taking the project topic of inventory (retail) management as example, if we have a table, customer\_details containing data of customers such as customer\_name, customer\_address, and their purchases which let’s say is denoted by attribute customer\_product. Then there may exist a chance of data to be redundant if a customer purchases multiple items and a cell may have more than one value because of multiple purchases which is not permitted in a database.

Through the process of normalization, a new attribute named customer\_id can be added to the entity in order to make sure that there are different records for same customer with multiple, different purchases. Furthermore, with the help of the customer\_id attribute, the customer\_details table can be split into multiple entities where customer\_id can be an attribute which will work as a primary key and reference key accordingly. So, this will lead to creation of two tables named customer\_details which will contain attributes customer\_id which will be the primary key, customer\_name, customer\_address and another table named customer\_purchase with attributes customer\_id as reference key and customer\_product. Also, in the first table(customer\_details), the primary key attribute customer\_id will act as a determinant, while other attributes in the same entity which are customer\_name, customer\_address will act as dependent in order to form a functional dependency relationship.

**Q2. Describe the concept of functional dependency and What are Anomalies in Relational Model and how to overcome?**

**Ans.** Functional dependency is a relationship between different attributes of the same entity which is determined by one or more values of the attributes. In this relation, attributes whose values can determine other attributes are known as determinants while attributes whose values are determined and dependent on other attributes are known as dependent.

Anomalies are problems and flaws that arise within poorly structured relational models that are riddled with inconsistencies mainly due to the presence of redundant data since with the presence of duplicate data present within a table, the chances of same data stored with different format leading to inconsistencies also increases. So, if changes being made to data will not be reflected to the same data in other parts of database due to such inconsistencies.

Normalization is a process that can mitigate such flaws since it is process which helps organize and structure the relational model properly by mitigating data redundancy which in turn helps to mitigate or eliminate data inconsistency.

**Q3. What is De-normalization and how it is helpful and explain in your own words as how it’s different than Normalization?**

**Ans.** Denormalization is the process of lowering the normal form of an entity in order to improve the performance yielded by the database by adding redundant copies of data.

However, such an improvement often comes at the expense of increased data redundancy which would further lead to inconsistencies in data as well as increase in the number of data anomalies. So, it is reversing the process of normalization since normalization is the process in which data redundancy and inconsistencies are mitigated by splitting tables, however tables are rather merged in denormalization for improved querying speed.

**Q4.** **Explain in your own words the below terms – (2M each)  
Consistency  
Concurrency  
Reliability  
Durability  
Relation (Entity)**

**Ans.**

**Consistency** in a relational database refers to the permitted changes that can take place in data within given constraints and rules. For example, in a transaction where transfer of funds is involved, the changes in data when funds are deducted from one account and added onto another account should be consistent such that there are no discrepancies in the total funds during the transaction. Any transactions being committed in the database should not be done in an inconsistent state.

**Concurrency:** In a multi-user environment, a database system may permit simultaneous execution of the same database. So multiple users can utilize, read, manipulate or analyze data in the same database at one time.

**Reliability:** Reliability of database system is defined by how well the database is structured. The database is considered reliable if the data stored in it is consistent, accurate and organized.

**Durability:** It is a transaction property which ensures that any changes that are being made to the data stored in the database does not get lost and will last permanently, even in the case of system failure.

**Relation (Entity):** It is a 2-dimensional structure comprising of tuples (rows) and attributes (columns) which intersect each other forming cells where data of a particular format is stored. This data is stored, processed and analyzed by a database user or organization for their intended purpose.

**Q5.** **What are the functions of Client server Architecture and what is 3 tier architecture?**

**Ans.** Client-Server architecture is a model that comprises of hardware and software components such as Servers that receives and services a request, client that sends in a request utilizes the response from the server and the middleware. Such an architectural model can help service the requests of multiple users or clients simultaneously with a reduced network traffic since most of the database processes are implemented at server side.

3-tier architecture is a client-server architecture model which consists of 3 layers which are presentation tier, application tier and data tier. In the presentation tier, graphical user interface or GUI of the application is defined where information is displayed, and it forms the front-end aspect of the application. The application tier manages the business logic of the application and contains the functionality aspect of the application through which information retrieved from presentation layer is processed. The data tier comprises of the database systems which stores the information and data for processing and analysis.

**Q6. The relational set operators UNION, INTERSECT, and MINUS work properly only when the relations are union compatible. What does union-compatible mean, and how would you check for this condition? What is the difference between UNION and UNION ALL? Suppose you have two tables: EMPLOYEE and EMPLOYEE\_1. The EMPLOYEE table contains the records for three employees: Alice Cordoza, John Cretchakov, and Anne McDonald. The EMPLOYEE\_1 table contains the records for employees John Cretchakov and Mary Chen. Given that information, list the query output for the UNION query.**

**Ans.** Union-compatible is a condition that two tables must achieve in order to implement the union operation which is an operation used in relational algebra to integrate two tables into one new table. Two tables are union-compatible if both the tables have the same number of columns or attributes, and the attributes have the same domain or column value.

UNION ALL is similar to UNION in the sense that it also appends the two tables into a new single table. However, unlike UNION operator, UNION ALL selects all the attributes of tables for merging which means that also includes duplicate rows, while UNION operator drops duplicate rows when it merges the attributes of a table.

When Union query is implemented on tables EMPLOYEE and EMPLOYEE\_1 then the names and records of both tables are merged into a single table and since UNION query is implemented, the duplicate data (name and record) which in this case is for employee name ‘John Cretchakov’ will be dropped and only one record of that employee will be merged into new table with other employee names and their records.

Graphical user interface, application

Description automatically generated

Graphical user interface, application, table

Description automatically generated

**Q7. What three join types are included in the OUTER JOIN classification?**

**Ans.** The three join types included in the OUTER JOIN classification are:

* **Left Outer Join:** Left Outer Join query yields all records of left table including the ones with no matching record in other table relative to the query used while unmatched records of right table will be set to null.
* **Right Outer Join:** Right Outer Join query yields all the records of right table including the ones with no matching record in other table relative to the query used while unmatched values of left table will be set to null.
* **Full Outer Join:** Full Outer Join query returns all records from both tables on which the query is implemented. The unmatched records of both tables are set to null.

**Q8. Suppose a PRODUCT table contains two attributes, PROD\_CODE and VEND\_CODE. Those two attributes have values of ABC, 125, DEF, 124, GHI, 124, and JKL, 123, respectively. The VENDOR table contains a single attribute, VEND\_CODE, with values 123, 124, 125, and 126, respectively. (The VEND\_CODE attribute in the PRODUCT table is a foreign key to the VEND\_CODE in the VENDOR table.) Given that information, what would be the query output for: a. A UNION query based on the two tables? b. A UNION ALL query based on the two tables? c. An INTERSECT query based on the two tables? d. A MINUS query based on the two tables?**

**Ans.**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

1. **UNION QUERY:** Duplicate data is dropped while all the other records from both VENDOR AND PRODUCT tables are merged into one table

**Graphical user interface, application

Description automatically generated**

1. **UNION ALL QUERY:** Duplicate data is kept from both PRODUCT and VENDOR tables. All records of both tables merged into one table

**Graphical user interface, text, application

Description automatically generated**

1. **INTERSECT QUERY:** Only common records of VEND\_CODE attribute from both PRODUCT and VENDOR tables are retained into a single table.

**Graphical user interface, application

Description automatically generated**

1. **MINUS QUERY:** Common VEND\_CODE attribute values from tables VENDOR and PRODUCT are dropped and only the unmatched record from VENDOR table is displayed

**Graphical user interface, text, application

Description automatically generated**

**Q9. Using the provided data model, create a list of events and the corresponding package code based on that event’s cost.**

**Ans.** Select NAME, PACKAGE\_CODE from D\_EVENTS group by COST;

**Q10. Using the provided data model, create a list of all client’s names, whether they have booked an event or not, if they have, show the event name and description**

**Ans.** Select D\_CLIENTS.FIRST\_NAME, D\_CLIENTS.LAST\_NAME, D\_EVENTS.NAME, D\_EVENTS.DESCRIPTION from D\_CLIENTS LEFT OUTER JOIN D\_EVENTS;