**DATABASE**

**Assignment 5**

**Submission Date: 28th Aug 2020**

1. Explain a Database system?

Ans : **Database Management System (DBMS)** is a software for storing and retrieving users' data while considering appropriate security measures. It consists of a group of programs which manipulate the database. The DBMS accepts the request for data from an application and instructs the operating system to provide the specific data.

DBMS software primarily functions as an interface between the end user and the database, simultaneously managing the data, the database engine, and the database schema in order to facilitate the organization and manipulation of data.

2. Explain database?

Ans : Database is a collection of inter-related data which helps in efficient retrieval, insertion and deletion of data from database and organizes the data in the form of tables, views, schemas, reports etc. For Example, university database organizes the data about students, faculty, and admin staff etc. which helps in efficient retrieval, insertion and deletion of data from it.

A database is a collection of related data which represents some aspect of the real world. A database system is designed to be built and populated with data for a certain task.

3. Define the benefits of DBMS?

**Ans :** Reducing Data Redundancy
Sharing of Data
Data Integrity
Data Security
Privacy
Backup and Recovery
Data Consistency

4. Write in brief the three levels of data abstraction.

Ans : There are mainly three levels of data abstraction:

1. Internal Level: Actual PHYSICAL storage structure and access paths.
2. Conceptual or Logical Level: Structure and constraints for the entire database
3. External or View level: Describes various user views

5. Explain durability in DBMS?

**Ans : <u>1. Atomicity-</u>**

- This property ensures that either the transaction occurs completely or it does not occur at all.

- In other words, it ensures that no transaction occurs partially.
- That is why, it is also referred to as "**All or nothing rule**".
- It is the responsibility of Transaction Control Manager to ensure atomicity of the transactions.

**2. Consistency-**

- This property ensures that integrity constraints are maintained.
- In other words, it ensures that the database remains consistent before and after the transaction.
- It is the responsibility of DBMS and application programmer to ensure consistency of the database.

**3. Isolation-**

- This property ensures that multiple transactions can occur simultaneously without causing any inconsistency.
- During execution, each transaction feels as if it is getting executed alone in the system.
- A transaction does not realize that there are other transactions as well getting executed parallely.
- Changes made by a transaction becomes visible to other transactions only after they are written in the memory.
- The resultant state of the system after executing all the transactions is same as the state that would be achieved if the transactions were executed serially one after the other.
- It is the responsibility of concurrency control manager to ensure isolation for all the transactions.

**4. Durability-**

- This property ensures that all the changes made by a transaction after its successful execution are written successfully to the disk.
- It also ensures that these changes exist permanently and are never lost even if there occurs a failure of any kind.
- It is the responsibility of recovery manager to ensure durability in the database.

6. What do you mean by atomicity and aggregation?

Ans : **Atomicity**

By this, we mean that either the entire transaction takes place at once or doesn't happen at all. There is no midway i.e. transactions do not occur partially. Each transaction is considered as one unit and either runs to completion or is not executed at all. It involves the following two operations.

—**Abort**: If a transaction aborts, changes made to database are not visible.

—**Commit**: If a transaction commits, changes made are visible.

Atomicity is also known as the 'All or nothing rule'.

**Aggregation –**

In aggregation, the relation between two entities is treated as a single entity. In aggregation, relationship with its corresponding entities is aggregated into a higher level entity.An ER diagram is not capable of representing relationship between an entity and a relationship which may be required in some scenarios. In those cases, a relationship with its corresponding entities is aggregated into a higher level entity. For Example, Employee working for a project may require some machinery. So, REQUIRE relationship is needed between relationship WORKS_FOR and entity MACHINERY. Using aggregation, WORKS_FOR relationship with its entities EMPLOYEE and PROJECT is aggregated into single entity and relationship REQUIRE is created between aggregated entity and MACHINERY.

7. Explain a checkpoint and When does it occur?

Ans : Checkpoint is a mechanism where all the previous logs are removed from the system and stored permanently in a storage disk. Checkpoint declares a point before which the DBMS was in consistent state, and all the transactions were committed.

8. Define the different phases of transaction?

Ans : The different phases of transaction :-

- **Active** − In this state, the transaction is being executed. This is the initial state of every transaction.
- **Partially Committed** − When a transaction executes its final operation, it is said to be in a partially committed state.
- **Failed** − A transaction is said to be in a failed state if any of the checks made by the database recovery system fails. A failed transaction can no longer proceed further.
- **Aborted** − If any of the checks fails and the transaction has reached a failed state, then the recovery manager rolls back all its write operations on the database to bring the database back to its original state where it was prior to the execution of the transaction. Transactions in this state are called aborted. The database recovery module can select one of the two operations after a transaction aborts −
    - Re-start the transaction
    - Kill the transaction
- **Committed** − If a transaction executes all its operations successfully, it is said to be committed. All its effects are now permanently established on the database system.

9. What do you mean by flat file database?

Ans : A flat file database is a database that stores data in a plain text file. Each line of the text file holds one record, with fields separated by delimiters, such as commas or tabs. While it uses a simple structure, a flat file database cannot contain multiple tables like a relational database can.

10. Explain "transparent DBMS"?

Ans : It is one, which keeps its Physical Structure hidden from user. Transparent hides implementation details from the user.

11. Explain a query?

Ans : A query is a request for data or information from a database table or combination of tables. This data may be generated as results returned by Structured Query Language (SQL) or as pictorials, graphs or complex results, e.g., trend analyses from data-mining tools.

12. What do you mean by Correlated subquery?

Ans : Correlated subqueries are used for row-by-row processing. Each subquery is executed once for every row of the outer query.

A correlated subquery is evaluated once for each row processed by the parent statement. The parent statement can be a **SELECT**, **UPDATE**, or **DELETE** statement.

13. How do you communicate with an RDBMS?

Ans : You can communicate RDBMS by structural query language (SQL).

The SQL is used for inserting/modifying/deleting & Retrieving data from Database.

You can also use SQL for Access controls & Administration.

SQL is classified into three categories, those categories are listed below.

Data Manipulation Language (DML)

14. Explain DDL (Data Definition Language)?

Ans : **DDL(Data Definition Language) :** DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.

**Examples of DDL commands:**

- **CREATE** – is used to create the database or its objects (like table, index, function, views, store procedure and triggers).
- **DROP** – is used to delete objects from the database.
- **ALTER**-is used to alter the structure of the database.
- **TRUNCATE**–is used to remove all records from a table, including all spaces allocated for the records are removed.
- **COMMENT** –is used to add comments to the data dictionary.
- **RENAME** –is used to rename an object existing in the database.

15. Explain VDL (View Definition Language)?

Ans : VDL: View Definition Language is used to specify user view and their mappings to the conceptual schema but in most DBMs's the DDL is used to define both conceptual and external schemas. In relational DBMS’s SQL is used in the sale of VDL to define user are application views as results of predefined queries.

16. Explain SDL (Storage Definition Language)?

Ans : This language is used to define internal schema. It defines that what will be the Physical structure of database, How many bites per field will be used, what will be the order of fields, and how records will be accesses etc.

17. Explain DML (Data Manipulation Language)?

Ans : **DML(Data Manipulation Language) :** The SQL commands that deals with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements.

**Examples of DML:**

- **INSERT** – is used to insert data into a table.
- **UPDATE** – is used to update existing data within a table.
- **DELETE** – is used to delete records from a database table.

18. Explain the "integrity rules"?

Ans : **Data integrity** is the maintenance of, and the assurance of the accuracy and consistency of data over its entire life-cycle,[1] and is a critical aspect to the design, implementation and usage of any system which stores, processes, or retrieves data.

19. Explain Data Independence?

Ans : A database system normally contains a lot of data in addition to users' data. For example, it stores data about data, known as metadata, to locate and retrieve data easily. It is rather difficult to modify or update a set of metadata once it is stored in the database. But as a DBMS expands, it needs to change over time to satisfy the requirements of the users. If the entire data is dependent, it would become a tedious and highly complex job.

20. Explain a view? How it is related to data independence?

Ans : A database view is a searchable object in a database that is defined by a query. Though a view doesn't store data, some refer to a views as "virtual tables," you can query a view like you can a table. A view can combine data from two or more table, using joins, and also just contain a subset of information. This makes them convenient to abstract, or hide, complicated queries.

Views in SQL are kind of virtual tables. A view also has rows and columns as they are in a real table in the database. We can create a view by selecting fields from one or more tables present in the database. A View can either have all the rows of a table or specific rows based on certain condition.

21. Explain Data Model?

Ans : A **data model** (or **datamodel**)[1][2][3][4][5] is an abstract model that organizes elements of data and standardizes how they relate to one another and to the properties of real-world entities. For instance, a data model may specify that the data element representing a car be composed of a number of other elements which, in turn, represent the color and size of the car and define its owner.

22. Explain E-R model?

Ans : **Entity Relationship Model**(ER Modeling) is a graphical approach to database design. It is a high-level data model that defines data elements and their relationship for a specified software system. An ER model is used to represent real-world objects.

23. Explain Object Oriented model?

Ans : Object oriented data model is based upon real world situations. These situations are represented as objects, with different attributes. All these object have multiple relationships between them.

**Elements of Object oriented data model**

**Objects**

The real world entities and situations are represented as objects in the Object oriented database model.

**Attributes and Method**

Every object has certain characteristics. These are represented using Attributes. The behaviour of the objects is represented using Methods.

**Class**

Similar attributes and methods are grouped together using a class. An object can be called as an instance of the class.

**Inheritance**

A new class can be derived from the original class. The derived class contains attributes and methods of the original class as well as its own.

24. Explain an Entity?

Ans : An entity is any singular, identifiable and separate object. It refers to individuals, organizations, systems, bits of data or even distinct system components that are considered significant in and of themselves.

**An entity is a real-world object that are represented in database. It can be any object,place,person or class.Data are stored about such entities.**

25. Explain an Entity type?

Ans : The *entity type* is the fundamental building block for describing the structure of data with the Entity Data Model (EDM). In a conceptual model, an entity type represents the structure of top-level concepts, such as customers or orders. An entity type is a template for entity type instances.

The entity type is a collection of the entity having similar attributes.

26. Explain an Entity set?

Ans : Entity Set is a collection of entities of the same entity type.

27. Explain Weak Entity set?

Ans : A **weak entity** is an **entity** that cannot exist in a database unless another type of **entity** also exists in that database. **Weak entities** cannot exist without the identifying **relationship**.

28. Explain an attribute?

Ans : In general, an attribute is a characteristic. In a database management system (DBMS), an attribute refers to a database component, such as a table. It also may refer to a database field. Attributes describe the instances in the row of a database

29. Explain a Relation Schema and a Relation?

Ans : A relation Schema denoted by R(A1, A2, ..., An) is made up of the relation name R and the list of attributes Ai that it contains. A relation is defined as a

set of tuples. Let r be the relation which contains set tuples (t1, t2, t3, ..., tn). Each tuple is an ordered list of n-values t=(v1,v2, ..., vn).

30. Explain degree of a Relation?

Ans : Degree of a Relation: The number of attributes in the relation is known as degree of the relation

31. Explain Relationship?

Ans : The association among entities is called a relationship. For example, an employee works_at a department, a student enrolls in a course. Here, Works_at and

Enrolls are called relationships.

32. Explain Relationship set?

Ans : A set of relationships of similar type is called a relationship set. Like entities, a relationship too can have attributes. These attributes are called

descriptive attribute.

33. Explain normalization?

Ans : Database Normalization is a technique of organizing the data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy(repetition) and undesirable characteristics like Insertion, Update and Deletion Anomalies. It is a multi-step process that puts data into tabular form, removing duplicated data from the relation tables.

Normalization is used for mainly two purposes,

- Eliminating redundant(useless) data.
- Ensuring data dependencies make sense i.e data is logically stored.

34. Explain Functional Dependency?

Ans : The functional dependency is a relationship that exists between two attributes. It typically exists between the primary key and non-key attribute within a table.

35. Explain Fully Functional dependency?

Ans : An attribute is fully functional dependent on another attribute, if it is Functionally Dependent on that attribute and not on any of its proper subset.

36. Explain 1 NF (Normal Form)?

Ans : First Normal Form (1NF)

A relation will be 1NF if it contains an atomic value.

It states that an attribute of a table cannot hold multiple values. It must hold only single-valued attribute.

First normal form disallows the multi-valued attribute, composite attribute, and their combinations.

37. Explain 2NF?

Ans : Second Normal Form (2NF)

In the 2NF, relational must be in 1NF.

In the second normal form, all non-key attributes are fully functional dependent on the primary key

Example: Let's assume, a school can store the data of teachers and the subjects they teach. In a school, a teacher can teach more than one subject.

38. Explain 3NF?

Ans : Third Normal Form (3NF)

A relation will be in 3NF if it is in 2NF and not contain any transitive partial dependency.

3NF is used to reduce the data duplication. It is also used to achieve the data integrity.

If there is no transitive dependency for non-prime attributes, then the relation must be in third normal form.

A relation is in third normal form if it holds atleast one of the following conditions for every non-trivial function dependency X → Y.

X is a super key.

Y is a prime attribute, i.e., each element of Y is part of some candidate key.

39. Explain BCNF (Boyce-Codd Normal Form)?

Ans : Boyce Codd normal form (BCNF) BCNF is the advance version of 3NF. It is stricter than 3NF.

A table is in BCNF if every functional dependency X → Y, X is the super key of the table.

For BCNF, the table should be in 3NF, and for every FD, LHS is super key.

40. Explain 4NF?

Ans : Fourth normal form (4NF) A relation will be in 4NF if it is in Boyce Codd normal form and has no multi-valued dependency.

For a dependency A → B, if for a single value of A, multiple values of B exists, then the relation will be a multi-valued dependency.

41. Explain Domain-Key Normal Form?

Ans : A relation is in DKNF when insertion or delete anomalies are not present in the database. Domain-Key Normal Form is the highest form of Normalization.

The reason is that the insertion and updation anomalies are removed. The constraints are verified by the domain and key constraints.

A table is in Domain-Key normal form only if it is in 4NF, 3NF and other normal forms

42. Define partial, alternate, artificial, compound and natural key?

Ans : Partial Key:It is a set of attributes that can uniquely identify weak entities and that are related to same owner entity. It is sometime called as Discriminator.

Alternate Key:All Candidate Keys excluding the Primary Key are known as Alternate Keys.

Artificial Key: If no obvious key, either stand alone or compound is available, then the last resort is to simply create a key, by assigning a unique number to each record or occurrence. Then this is known as developing an artificial key.

Compound Key:If no single data element uniquely identifies occurrences within a construct, then combining multiple elements to create a unique identifier for

the construct is known as creating a compound key.

Natural Key:When one of the data elements stored within a construct is utilized as the primary key,then it is called the natural key.

43. Explain indexing and define the different kinds of indexing?

Ans : INDEXING is a data structure technique which allows you to quickly retrieve records from a database file. An Index is a small table having only two columns.

The first column comprises a copy of the primary or candidate key of a table. Its second column contains a set of pointers for holding the address of the disk

block where that specific key value stored.

An index -

Takes a search key as input

Efficiently returns a collection of matching records.

Two main types of indexing methods are:

44. Write in brief the four types of indexes.

Ans : The four types of indexes.

1)Primary Indexing:Primary Index is an ordered file which is fixed length size with two fields.

The primary Indexing is also further divided into two types.

1)Dense Index:In a dense index, a record is created for every search key valued in the database. This helps you to search faster but needs more space to store index records. In this Indexing, method records contain search key value and points to the real record on the disk.

2)Sparse Index:It is an index record that appears for only some of the values in the file. Sparse Index helps you to resolve the issues of dense Indexing. In this method of indexing technique, a range of index columns stores the same data block address, and when data needs to be retrieved, the block address will be fetched.

2)Secondary Index:The secondary Index can be generated by a field which has a unique value for each record, and it should be a candidate key. It is also known as a non-clustering index.This two-level database indexing technique is used to reduce the mapping size of the first level. For the first level, a large range of numbers is selected because of this; the mapping size always remains small.

3) Clustering Index

In a clustered index, records themselves are stored in the Index and not pointers. Sometimes the Index is created on non-primary key columns which might not be unique for each record. In such a situation, you can group two or more columns to get the unique values and create an index which is called clustered Index. This also helps you to identify the record faster.

4) Multilevel Indexing is created when a primary index does not fit in memory. In this type of indexing method, you can reduce the number of disk accesses to short any record and kept on a disk as a sequential file and create a sparse base on that file.

45. Explain system catalog or catalog relation? How is better known as?

Ans : A RDBMS maintains a description of all the data that it contains, information about every relation and index that it contains. This information is stored in a collection of relations maintained by the system called metadata. It is also called data dictionary.

46. Explain meant by query optimization?

Ans : Query Optimization: A single query can be executed through different algorithms or re-written in different forms and structures. Hence, the question of query optimization comes into the picture – Which of these forms or pathways is the most optimal? The query optimizer attempts to determine the most efficient way to execute a given query by considering the possible query plans.

47. Explain SQL and state the differences among SQL and other conventional programming Languages.

Ans : SQL is a nonprocedural language that is designed specifically for data access operations on normalized relational database structures. The primary difference between SQL and other conventional programming languages is that SQL statements specify what data operations should be performed rather than how to perform them.

48. Explain database Trigger?

Ans : Trigger is a statement that a system executes automatically when there is any modification to the database. In a trigger, we first specify when the trigger is to be executed and then the action to be performed when the trigger executes. Triggers are used to specify certain integrity constraints and referential constraints that cannot be specified using the constraint mechanism of SQL.

49. Name four applications for triggers.

Ans : 1) providing default values,

(2) enforcing data constraints,

(3) updating views and

(4) enforcing referential integrity.

50. Define stored-procedures? And Define the Benifits of using them?

Ans : -Stored Procedures are created to perform one or more DML operations on Database. It is nothing but the group of SQL statements that accepts some input in the form of parameters and performs some task and may or may not returns a value.

Syntax : Creating a Procedure

```
CREATE or REPLACE PROCEDURE name(parameters)
IS
variables;
BEGIN
//statements;
END;
```

A Stored Procedure is a type of code in SQL that can be stored for later use and can be used many times. So, whenever you need to execute the query, instead of

calling it you can just call the stored procedure. You can also pass parameters to a stored procedure, so that the stored procedure can act based on the parameter values that is passed.

The main advantages of stored procedure are given below:

1)Better Performance –The procedure calls are quick and efficient as stored procedures are compiled once and stored in executable form.Hence the response is quick. The executable code is automatically cached, hence lowers the memory requirements.

2)Maintainability–Maintaining a procedure on a server is much easier then maintaining copies on various client machines, this is because scripts are in one location.

3)Security–Access to the Oracle data can be restricted by allowing users to manipulate the data only through stored procedures that execute with their definer’s privileges.

## DATABASE

## Assignment 6

## Submission Date: 4th Aug 2020

51. Define cursors give different types of cursors?
52. Define data and information, and how are they related in a database?
53. Explain Enterprise Resource Planning (ERP), and what kind of a database is used in an ERP application?
54. Write an SQL SELECT statement to display all the columns of the STUDENT table but only those rows where the Grade column is greater than or equal to 90.
55. Name and briefly Write in brief the five SQL built-in functions.
56. Write an SQL SELECT statement to count the number of rows in STUDENT table and display the result with the label NumStudents.
57. Explain an SQL subquery?
58. Explain a foreign key, and explain it used for?
59. Define the steps for transforming an entity into a table?
60. Explain a surrogate key, Write in brief the ideal primary key and Write in brief how surrogate keys meet this ideal
61. Explain a cascading update?
62. Explain a SQL view? Briefly Write in brief the use of views.
63. Write in brief how to add a NOT NULL column to a table.
64. You have two tables, EMPLOYEE and COMPUTER that are in a one-to-one relationship. The foreign key is EmpNumber in COMPUTER which references EmpNumber as the primary key of EMPLOYEE. Write in brief what must be done to convert the one-to-one EMPLOYEE-COMPUTER relationship to a one-to-many relationship where one employee can have more than one computer.
65. Write in brief what we mean by an ACID transaction.
66. Write in brief what needs to happen to convert a relation to third normal form.
67. Explain denormalizations and why would someone consider doing so?
68. Compare a hierarchical and network database model?
69. Write in brief the difference among a dynamic and materialized view.
70. Briefly Write in brief the three types of SQL commands.
71. Define the steps to follow when preparing to create a table?
72. Write in brief a join among tables
73. Write in brief and contrast a trigger and a procedure.
74. Briefly Write in brief an outer join.
75. Write in brief a subquery.

76. Write in brief the difference among two and three-tier architectures.

77. Write in brief a data warehouse.

78. Write in brief the differences among a data warehouse and data mart.

79. Write in brief the difference among data and database administration.

80. Define some of the important security features of a DBMS?

81. Write in brief the difference among homogeneous and heterogeneous distributed database.

82. Explain a distributed database?

83. Write a query to print Second highest salary.

84. Define different types of clauses?

85. Explain inner join? Write it’s query

86. Types of joins, slowest and fastest amongst them.