DATABASE

Assignment

Team No.2

1) Explain a Database system?

Ans: Database System is generally a software which is used to create and manage the database. The software generally is a tool which allows user to create, store, modify, access and perform various operations on the data to retrieve useful information. The software communicates with the operating system to work with the data. So, The user does not need to worry much about internal processing.Database system takes care of the security of the data as well.

2) Explain database?

Ans: Database is an organised collection of data which is generally stored in server Hard disk and that data can be operated using different DataBase Managegement System tools.

3) Define the benefits of DBMS?

Ans: Reducing Data Redundency
Sharing Data
Data Integrity
Data Security
Privacy
BackUp and Recovery
Data Consistency

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

Ans: 3 levels of data Abstraction:

I. Internal level/Schema

The internal schema is the lowest level of data abstraction

It helps you to keeps information about the actual representation of the entire database. Like the actual storage of the data on the disk in the form of records

The internal view tells us what data is stored in the database and how

It never deals with the physical devices. Instead, internal schema views a physical device as a collection of physical pages

II. Conceptual Level/Logical Level

Defines all database entities, their attributes, and their relationships
Security and integrity information
In the conceptual level, the data available to a user must be contained in or derivable from the physical level

III. External Level/View Level
An external level is only related to the data which is viewed by specific end users.
This level includes some external schemas.
External schema level is nearest to the user
The external schema describes the segment of the database which is needed for a certain user group and hides the remaining details from the database from the specific user group.

5) Explain durability in DBMS?

Ans: In database systems, durability is the ACID property which guarantees that transactions that have committed will survive permanently.

For example, if a flight booking reports that a seat has successfully been booked, then the seat will remain booked even if the system crashes.

Durability can be achieved by flushing the transaction's log records to non-volatile storage before acknowledging commitment.

In distributed transactions, all participating servers must coordinate before commit can be acknowledged. This is usually done by a two-phase commit protocol.

Many DBMSs implement durability by writing transactions into a transaction log that can be reprocessed to recreate the system state right before any later failure. A transaction is deemed committed only after it is entered in the log.

6) What do you mean by atomicity and aggregation?

Ans: Atomicity: Either all actions are carried out or none are. Users should not have to worry about the effect of incomplete transactions. DBMS ensures this by undoing the actions of incomplete transactions.
Aggregation: A concept which is used to model a relationship between a collection of entities and relationships.

7) Explain a checkpoint and When does it occur?

Ans: Checkpoint is a process that writes current in-memory dirty pages (modified pages) and transaction log records to physical disk. ... The following set of operations starts when checkpoint occurs: Log records from log buffer (including the last log record) are written to the disk.

8) Define the different phases of transaction?

Ans: Active state: This phase is divided into two states: - Initial phase: This phase is achieved when the transaction starts. -
Partially Committed phase: This is achieved when the transactions final statement has been executed.

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. Fortunately, most database programs such as Microsoft Access and FileMaker Pro can import flat file databases and use them in a larger relational database.

10) Explain "transparent DBMS"?

Ans: It is one, which keeps its Physical Structure hidden from user. A query with respect to DBMS relates to user commands that are used to interact with a data base. The query language can be classified into data definition language and data manipulation language.

11) Explain a query?

Ans: In database terms, a query is used to retrieve data from the database.Queries are one of the things that make databases so powerful.

A "query" refers to the action of retrieving data from your database. Usually, you will be selective with how much data you want returned.If you have a lot of data in your database, you probably don't want to see everything. More likely, you'll only want to see data that fits a certain criteria.

For example, you might only want to see how many individuals in your database live in a given city.Or you might only want to see which individuals have registered with your database within a given time period.As with many other tasks, you can query a database either programatically or via a user interface.

12) What do you mean by Correlated subquery?

Ans: SQL Correlated Subqueries are used to select data from a table referenced in the outer query. The subquery is known as a correlated because the subquery is related to the outer

query. In this type of queries, a table alias (also called a correlation name) must be used to specify which table reference is to be used.If you have a join alongwith distinct,to make it work faster,use correlated sub-query.EXISTS operator can be used in correlated subqueries also.

13) How do you communicate with an RDBMS?

Ans: We can communicate RDBMS by structural query language (SQL).You can communicate RDBMS by structural query language (SQL). You can also use SQL for Access controls & Administration.

14) Explain DDL (Data Definition Language)?

Ans: 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.

15) Explain VDL (View Definition Language)?
Ans: 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.
It defines the subset of records available to classes of users.
It creates virtual tables and the view appears to users like conceptual level.
It specifies user interfaces.

16) Explain SDL(Storage Defination Language)?

Ans: SDL(Storage Definition Language):
It is used to specify the internal schema.
The mappings between the two schemas may be specified in either one of these languages.
In most relational DBMS,there is no specific language that performs the role of SDL.
Instead, the internal schema is specified by a combination of functions, parameters, and specifications related to storage of files.
These permit the DBA staff to control indexing choices and mapping of data to storage.

For a true three-schema architecture, we would need a third language, the view definition language (VDL), to specify user views and their mappings to the conceptual schema, but in most DBMSs the DDL is used to define both conceptual and external schemas.

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 "Integrity Rules"

Ans: Business rules: Obtained from users when gathering requirements and are used to determine cardinality.

Cardinality: Expresses the minimum and maximum number of entity occurrences associated with one occurrence of a related entity.

Connectivity: The relationship between two tables, e.g., one to one or one to many.

Constraints: The rules that force DBMSs to check that data satisfies the semantics.

Entity integrity: Requires that every table have a primary key; neither the primary key, nor any part of it, can contain null values.

Identifying relationship: Where the primary key contains the foreign key; indicated in an ERD by a solid line.

Integrity constraints: Logical statements that state what data values are or are not allowed and which format is suitable for an attribute.

Mandatory relationship: One entity occurrence requires a corresponding entity occurrence.

Non-identifying relationship: Does not contain the foreign key in the primary key; indicated in an ERD by a dotted line.

Optional relationship: The FK can be null or the parent table does not need to have a corresponding child table occurrence.

Orphan record: A record whose foreign key value is not found in the corresponding entity – the entity where the primary key is located.

Referential integrity: Requires that a foreign key must have a matching primary key or it must be null.

Relational database management system (RDBMS): A popular database system based on the relational model introduced by E. F. Codd of IBM's San Jose Research Laboratory.

Relationship type: The type of relationship between two tables in an ERD (either identifying or non-identifying); This relationship is indicated by a line drawn between the two tables.

19) Explain Data Independence?

Ans: Data Independence is defined as a property of DBMS That helps you to change the Database schema at one level of a database system without requiring to change the schema at the next higher level. Data independence helps you to keep data separated from all programs that make use of it.

20) Explain a view? How it related to data independence?

Ans: A View is a "Virtual Table".

It is not like a simple table, but is a virtual table
which contains columns and data from different tables (may be one or more tables).

A View does not contain any data, it is a set of queries that are applied to one or more tables that is stored within the database as an object.

After creating a view from some table(s),
it used as a reference of those tables and when executed,
it shows only those data which are already mentioned in the query during the creation of the View.

A view may be thought of as a virtual table, that is, a table that does not really exist in its own right but is instead
derived from one or more underlying base table.

There is no stored file that direct represents the view instead a definition of view is stored in data dictionary.

21) Define view? how it is related to data independence, Database Management System.

Ans: A View may be defined of as a virtual table, that is, a table that does not really exist in its own right but is instead inferred from one or more underlying base data table. There is no stored file that direct shows the view instead a definition of view is loaded in data dictionary Restructuring and growth of base tables is not shown in views. Thus the view may insulate end users from the effects of developing and growth in the database.

22) Explain Data Model?

Ans: Data models define how the logical structure of a database is modeled. Data Models are fundamental entities to introduce abstraction in a DBMS. Data models define how data is connected to each other and how they are processed and stored inside the system.
The very first data model could be flat data-models, where all the data used are to be kept in the same plane. Earlier data models were not so scientific, hence they were prone to introduce lots of duplication and update anomalies.

23) Explain E-R model?

Ans: Entity-Relationship (ER) Model is based on the notion of real-world entities and relationships among them. While formulating real-world scenario into the database model, the ER Model creates entity set, relationship set, general attributes and constraints.

ER Model is best used for the conceptual design of a database.

ER Model is based on −

Entities and their attributes.

Relationships among entities.

24.Explain Object Oriented model?

Ans: The ODBMS which is an abbreviation for object oriented database management system, is the data model in which data is stored in form of objects, which are instances of classes. These classes and objects together makes an object oriented data model.

Components of Object Oriented Data Model: The OODBMS is based on three major components, namely: Object structure, Object classes, and Object identity. These are explained as following below.

25) What is entity?

Ans: An entity can be a real-world object, either animate or inanimate, that can be easily identifiable. For example, in a school database, students, teachers, classes, and courses offered can be considered as entities. All these entities have some attributes or properties that give them their identity.

An entity set is a collection of similar types of entities. An entity set may contain entities with attribute sharing similar values. For example, a Students set may contain all the students of a school; likewise a Teachers set may contain all the teachers of a school from all faculties. Entity sets need not be disjoint.

26) Explain Entity Set?

Ans: In dbms, entity set is set of entities of same type...

Attributes are associated with an entity set.

Attributes describe the properties of entities in the entity set.

Based on the values of certain attributes, an entity can be identified uniquely.

entity set has two types:-

1)Strong entity set

2)Weak entity set

STRONG ENTITY SET:-

A strong entity set is an entity set that contains sufficient attributes to uniquely identify all its entities.

In other words, a primary key exists for a strong entity set.

Primary key of a strong entity set is represented by underlining it.

A single rectangle is used for representing a strong entity set.

A diamond symbol is used for representing the relationship that exists between two strong entity sets.

A single line is used for representing the connection of the strong entity set with the relationship set.

A double line is used for representing the total participation of an entity set with the relationship set.

WEAK ENTITY SET:-

A weak entity set is an entity set that does not contain sufficient attributes to uniquely identify its entities.

27) Explain weak Entity set?

Ans: A weak entity set is an entity set that does not contain sufficient attributes to uniquely identify its entities.

1. In other words, a primary key does not exist for a weak entity set.

2. However, it contains a partial key called as a discriminator.

3. Discriminator can identify a group of entities from the entity set.

4. Discriminator is represented by underlining with a dashed line.

SYMBOL USED TO SHOW WEAK ENTITY SET:-

1. A double rectangle is used for representing a weak entity set.

2. A double diamond symbol is used for representing the relationship that exists between the strong and weak entity sets and this relationship is known as identifying relationship.

3. A double line is used for representing the connection of the weak entity set with the relationship set.

28) Explain Attribute?

Ans: In RDBMS, a table organizes data in rows and columns. The columns are known as attributes whereas the rows are known as records.

Type of attribute:-

1. Simple attributes

2. Composite attributes

3. Single valued attributes

4. Multi valued attributes

5. Derived attributes

6. Key attributes

1) simple attribute:-Simple attributes are those attributes which can not be divided further.

2) Composite attributes:-Composite attributes are those attributes which are composed of many other simple attributes.

NAME can further devided into first name,last name and middle name..

address can be further diveded as road no,building name,plot no..

3) Single valued attributes:-Single valued attributes are those attributes which can take only one value for a given entity from an entity set.

4) Multi valued attributes:-Multi valued attributes are those attributes which can take more than one value for a given entity from an entity set.

5) Derived attributes:- Derived attributes are those attributes which can be derived from other attribute(s).

29) What is Relation schema and Relation?

Ans: A relation schema means it refers to organisation of data as blueprint as how database is constructed.. and relation is set of touples(d1,d2..dn) in which each element(d1) is related to data domain(D1).

30) What is Degree of Relationship?

Ans: The degree of a relationship is the number of entity types that participate(associate) in a relationship. The number of an entity type that is connected to a relationship is the degree of the relationship.

31) Explain Relationship?

Ans: A relationship, in the context of databases, is a situation that exists between two relational database tables when one table has a foreign key that references the primary key of the other table. Relationships allow relational databases to split and store data in different tables, while linking disparate data items.

32) Explain Relationship set?

Ans: A relationship set is a set of relationships of same type.

Degree of a Relationship Set- The number of entity sets that participate in a relationship set is termed as the degree of that relationship set. Thus, Degree of a relationship set = Number of entity sets participating in a relationship set

Types of Relationship Sets-(On the basis of degree of a relationship set, a relationship set can be classified into the following types)

-Unary relationship set

-Binary relationship set

-Ternary relationship set

-N-ary relationship set

33) Explain normalization?

Ans: Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.

Here are the most commonly used normal forms:

-First normal form(1NF)

-Second normal form(2NF)

-Third normal form(3NF)

-Boyce & Codd normal form (BCNF)

34) Explain Functional Dependency?

Ans: A functional dependency (FD) is a relationship between two attributes, typically between the PK and other non-key attributes within a table.

OR

The attributes of a table is said to be dependent on each other when an attribute of a table uniquely identifies another attribute of the same table.

For example: Suppose we have a student table with attributes: Stu_Id, Stu_Name, Stu_Age. Here Stu_Id attribute uniquely identifies the Stu_Name attribute of student table because if we know the student id we can tell the student name associated with it. This is known as functional dependency and can be written as Stu_Id->Stu_Name or in words we can say Stu_Name is functionally dependent on Stu_Id.

35) Explain Fully Functional dependency?

Ans: A full functional dependency is a state of database normalization that equates to the normalization standard of Second Normal Form (2NF). In brief, this means that it meets the requirements of First Normal Form (1NF), and all non-key attributes are fully functionally dependent on the primary key.

36) Explain 1 NF (Normal Form)?

Ans: If a relation contain composite or multi-valued attribute, it violates first normal form, or a relation is in first normal form if it does not contain any composite or multi-valued attribute. A relation is in first normal form if every attribute in that relation is singled valued attribute.

A table is in 1 NF iff:

1. There are only Single Valued Attributes.
2. Attribute Domain does not change.
3. There is a Unique name for every Attribute/Column.
4. The order in which data is stored, does not matter.
5. A relation will be 1NF if it contains an atomic value.
6. It states that an attribute of a table cannot hold multiple values. It must hold only single-valued attribute.
7. First normal form disallows the multi-valued attribute, composite attribute, and their combinations.

Example: Relation EMPLOYEE is not in 1NF because of multi-valued attribute EMP_PHONE.

37) Explain 2NF?

Ans: Second Normal Form (2NF) is based on the concept of full functional dependency. Second Normal Form applies to relations with composite keys, that is, relations with a primary key composed of two or more attributes. A relation with a single-attribute primary key is automatically in at least 2NF. A relation that is not in 2NF may suffer from the update anomalies.

In other words, A relation that is in First Normal Form and every non-primary-key attribute is fully functionally dependent on the primary key, then the relation is in 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: A relation is in third normal form, if there is no transitive dependency for non-prime attributes as well as it is in second normal form.

A relation is in 3NF if at least one of the following condition holds in every non-trivial function dependency X –> Y:

1. X is a super key.
2. Y is a prime attribute (each element of Y is part of some candidate key).

In other words,

A relation that is in First and Second Normal Form and in which no non-primary-key attribute is transitively dependent on the primary key, then it is in Third Normal Form (3NF).

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

Ans: Boyce–Codd normal form (or BCNF or 3.5NF) is a normal form used in database normalization. It is a slightly stronger version of the third normal form (3NF). ... If a relational schema is in BCNF then all redundancy based on functional dependency has been removed, although other types of redundancy may still exist.

Rules For BCNF:

1. It should be in the Third Normal Form.
2. And, for any dependency A → B, A should be a super key.

40) Explain 4NF?

Ans: Fourth normal form (4NF) is a level of database normalization where there are no non-trivial multivalued dependencies other than a candidate key.

It builds on the first three normal forms (1NF, 2NF and 3NF) and the Boyce-Codd Normal Form (BCNF). It states that, in addition to a database meeting the requirements of BCNF, it must not contain more than one multivalued dependency.

O A relation will be in 4NF if it is in Boyce Codd normal form and has no multi-valued dependency.

O 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: Domain-key Normal Form :-

* It is basically a process in database to organize data efficiently. Basically there are two goals of doing normalization these are as follows:

1. To remove repeated data or in simple words we can say to remove redundant data.

2. Second one is to ensure that there will be data dependencies.

* DKNF is a normal form used in database normalization which requires that the database contains no constraints other than domain

constraints and key constraints.

* Reason to use DKNF are as follows:

1. To avoid general constraints in the database that are not clear key constraints.

2. Most database can easily test or check key constraints on attributes.

* However, because of the difficulty of including complex constraints in a DKNF relation its practical utility is limited

means that they are not in practical use, since it may be quite difficult to specify general integrity constraints.

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 : Indexing is a way to optimize the performance of a database by minimizing the number of disk accesses required when a query is processed. It is a data structure technique which is used to quickly locate and access the data in a database.

There are primarily three methods of indexing:

-> Clustered Indexing :

When more than two records are stored in the same file these types of storing known as cluster indexing. By using the cluster indexing we can reduce the cost of searching reason being multiple records related to the same thing are stored at one place and it also gives the frequent joing of more than two tables(records).

->Non-Clustered or Secondary Indexing :

A non clustered index just tells us where the data lies, i.e. it gives us a list of virtual pointers or references to the location where the data is actually stored. Data is not physically stored in the order of the index.

->Multilevel Indexing :

With the growth of the size of the database, indices also grow. As the index is stored in the main memory, a single-level index might become too large a size to store with multiple disk accesses. The multilevel indexing segregates the main block into various smaller blocks so that the same can stored in a single block.

44) Write in brief the four types of indexes.

Ans: There are four types of indexes :

Unique Index:

This index does not allow the field to have duplicate values if the column is unique indexed. If a primary key is defined, a unique index can be applied automatically.

Clustered Index:

This index reorders the physical order of the table and searches based on the basis of key values. Each table can only have one clustered index.

Non-Clustered Index:

Non-Clustered Index does not alter the physical order of the table and maintains a logical order of the data. Each table can have many nonclustered indexes.

Bitmap index :

A bitmap index is a special kind of indexing that stores the bulk of its data as bit arrays (bitmaps) and answers most queries by performing bitwise logical operations on these bitmaps. The most commonly used indexes, such as B+ trees, are most efficient if the values they index do not repeat or repeat a small number of times. In contrast, the bitmap index is designed for cases where the values of a variable repeat very frequently.

Dense index :

A dense index in databases is a file with pairs of keys and pointers for every record in the data file. Every key in this file is associated with a particular pointer to a record in the sorted data file. In clustered indices with duplicate keys, the dense index points to the first record with that key.[3]

Sparse index :

A sparse index in databases is a file with pairs of keys and pointers for every block in the data file. Every key in this file is associated with a particular pointer to the block in the sorted data file. In clustered indices with duplicate keys, the sparse index points to the lowest search key in each block.

Reverse index :

A reverse-key index reverses the key value before entering it in the index.

Primary index :

The primary index contains the key fields of the table and a pointer to the non-key fields of the table. The primary index is created automatically when the table is created in the database.

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

Ans: System catalog or catalog :

* A system catalog is a group of tables and views that incorporate vital details regarding a database. Every database comprised of a system catalog and the information in the system catalog specifies the framework of the database.

The system catalog is a set of objects, which includes information that defines:

1. Other objects included in the database

2. The database structure itself

3. Several other vital pieces of information

* A user generally looks up the system catalog to gain information regarding the user's own objects as well as privileges, while the database admin must be capable of inquiring about any event or structure inside the database.

* In certain implementations, one can find system catalog objects, which can be accessible only by the administrator of the database.

* A system catalog is extremely important to database admins or all other database users who wish to understand the nature and structure of a database.

* The system catalog lets order to be kept, not just by the users and database administrator, but also by the database server as well.

46) Explain meant by query optimization?

Ans: A query optimizer is a critical database management system (DBMS) component that analyzes Structured Query Language (SQL) queries and determines efficient execution mechanisms. A query optimizer generates one or more query plans for each query, each of which may be a mechanism used to run a query. The most efficient query plan is selected and used to run the query.

The phase that identifies an efficient execution plan for evaluating a query that has the least estimated cost is referred to as query optimization.

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: A database trigger is special stored procedure that is run when specific actions occur within a database. Most triggers are defined to run when changes are made to a table's data. Triggers can be defined to run instead of or after DML (Data Manipulation Language) actions such as INSERT, UPDATE, and DELETE.

49) Name four applications for triggers.

Ans: 1.Modify table data when DML statements are issued against views

2.Publish information about database events, user events, and SQL statements to subscribing applications

3.Restrict DML operations against a table to those issued during regular business hours

4.Enforce security authorizations

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

Ans: Stored procedures are database objects that perform a user defined operation. A stored procedure can have a set of compound SQL statements. A stored procedure executes the SQL commands and returns the result to the client. Stored procedures are used to reduce network traffic.