Database Management System

---

1. What are Hash Files?
    - In the index sequential file organisation considered, the mapping from the search key value to the storage location is via index entries.
    - In direct file organisations, the key value is mapped directly to the storage location, avoiding the use of indices.
    - The usual method of direct mapping is by performing some arithmetic manipulation of the key value. This process is called hashing.
    - However hashing schemes usually give rise to collisions when two or more distinct key values are mapped to the same value.
    - Collisions are handled in a number of ways .The colliding records may be assigned to the next available free space or they may be assigned to an overflow area.
    - In using the hash function to generate a value, which is the address of a bucket where the <key, address> pair values of records are stored, we can handle limited collisions as well as re-organizations of the file without affecting the hash function.
    - In extendable hashing; the database size changes are handled by splitting or coalescing buckets.
2. What is Database?
    - Database is a collection of interrelated data with an implicit meaning.It is a collection of persistent data that is used by the application systems of some given enterprise.A database can be of any size and varying complexity.A database may be generated and maintained manually or by machine.
3. What is a Database System?
    - A database system is an integrated collection of related files, along with details of interpretation of the data contained therein.Database system is basically a computerised record keeping system i.e. a computerised system whose overall purpose is to store information and to allow users to retrieve and update that information on demand.

4. What is a Database Management System?
    - DBMS is a s/w system that allows access to data contained in a database. It is a collection of programs that enables users to create and maintain a database.
    - The objective of the DBMS is to provide a convenient and effective method of defining, storing and retrieving the information contained in the database.
    - A DBMS is hence a general purpose s/w system that facilitates the process of Defining, Constructing and Manipulating databases for various applications.
5. Advantages of DBMS
    - Reduction of Redundancies:Centralised control of data by the DBA avoids unnecessary duplication of data and effectively reduces the total amount of data storage required.Another advantage of avoiding duplication is the elimination of the inconsistencies that tend to be present in redundant data files.
    - Shared Data:A database allows the sharing of data under its control by any number of application programs or users.Eg.The application for the public relations and payroll departments could share the data contained for the record type EMPLOYEE
    - Integrity: Centralised control can also ensure that adequate checks are incorporated in the DBMS to provide data integrity, which means that the data contained in the database is both accurate and consistent.
    - Security:Data is of vital importance to an organisation and may be confidential. Such confidential data must not be accessed by unauthorised persons. The DBA who has the ultimate responsibility for the data in the DBMS can ensure that proper access procedures are followed,including proper authentication schemes for access to the DBMS and additional checks before permitting access to sensitive data.
    - Conflict Resolution:Since the database is under the control of DBA, he could resolve the conflicting requirements of various users and applications. In essence, the DBA chooses the best file structure and access method to get optimal

performance for the response critical applications, while permitting less critical applications to continue to use the database, with a relatively slower response.

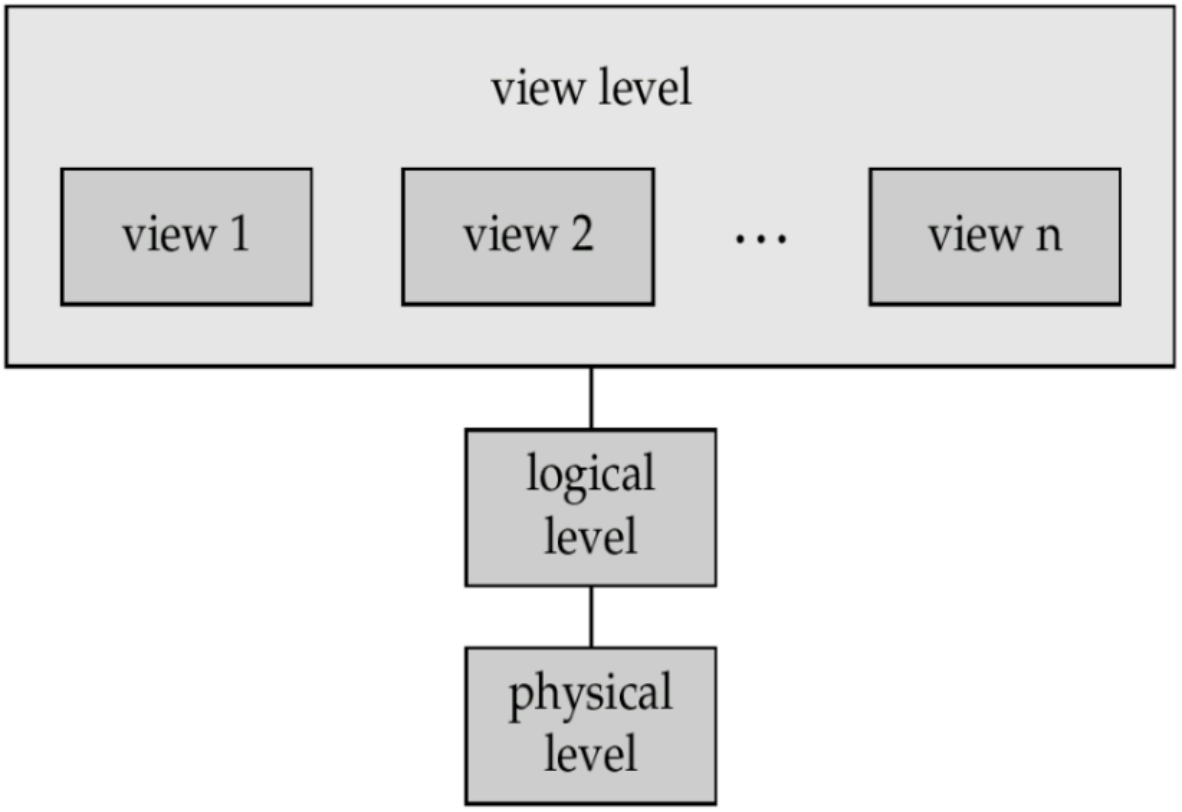
- Data Independence:Data independence is advantageous in the database environment since it allows for changes at one level of the database without affecting other levels.These changes are absorbed by the mappings between the levels. It Is usually considered from 2 points of view:
  - Physical data independence - It allows changes in the physical storage devices or organisation of files to be made without requiring changes in the conceptual view or any of the external views and hence in the application programs using the database. Thus the files may migrate from one type of physical media to another or the file structure may change without any need for changes in the application programs.
  - Logical data independence - It implies that application programs need not be changed if fields are added to an existing record, nor do they have to be changed if fields not used by application programs are deleted. Logical data independence indicates that the conceptual schema can be changed without affecting the existing external schemas.

6. Disadvantages of DBMS
   - Significant disadvantage of DBMS is cost.
   - In addition to the cost of purchasing or developing the software, the hardware has to be upgraded to allow for the extensive programs and work spaces required for their execution and storage.
   - The processing overhead introduced by the DBMS to implement security,integrity and sharing of the data causes a degradation of the response and throughput times.
   - An additional cost is that of migration from a traditionally separate application environment to an integrated one.
   - While centralization reduces duplication, the lack of duplication requires that the database be adequately backed up so that in the case of failure the data can be recovered.

Backup and recovery operations are fairly complex in the DBMS environment. Furthermore a database system requires a certain amount of controlled redundancies and duplication to enable access to related data items.

- Centralization also means that the data is accessible from a single source namely the database. This increases the potential severity of security branches and disruption of the operation of the organisation because of downtimes and failures.
- The replacement of a monolithic centralised database by a federation of independent and cooperating distributed databases resolves some of the problems resulting from failures and downtimes.

7. Explain the Architecture of DBMS



- The 3 levels of architecture are:
- Physical Level(also known as the Internal Level) - It is the closest level to the physical storage (i.e. One concerned with the way the data is physically stored). It is at a low-level representation of the entire database. It consists of many occurrences of each of the many types of internal record.The

internal view is described by means of the internal schema, which not only defines the various stored record types but also specifies what indexes exist, how stored fields are represented, what physical sequence the stored records are in and so on.

- Logical Level(also known as Conceptual level/community logical level) - It is the level of indirection between the other two levels. It is a representation of the entire information content of the database.The conceptual view is defined by means of the conceptual schema, which includes definitions of each of the various conceptual record types. The conceptual schema is written using another data definition language, the conceptual DDL.
- External Level(Also known as the User Logical Level) - It is the closest level to the users – It is the one concerned with the way the data is seen by individual users. In other words there will be many distinct “external views”, each consisting of a more or less abstract representation of some portion of the total database. It is at the individual user level. A given user can be either an application programmer or an end user of any degree of sophistication.The external view is defined by means of external schema. The external schema is written using the (data definition language)DDL portion of the user’s Data sublanguage.

8. Who is a DBA & Explain the Role of DBA

Centralised control of the database is exerted by a person or group of persons under the supervision of a high level administrator. They are referred to as the Database Administrator(DBA). They are the users who are most familiar with the database and are responsible for creating, modifying and maintaining its 3 levels.

- The DBA is the custodian of data and controls the database structures. The DBA administers the 3 levels of the database and in consultation with the overall user community, sets up the definition of the global view or conceptual level of the database.
- The DBA further specifies the external view of the various users and applications and is responsible for the definition

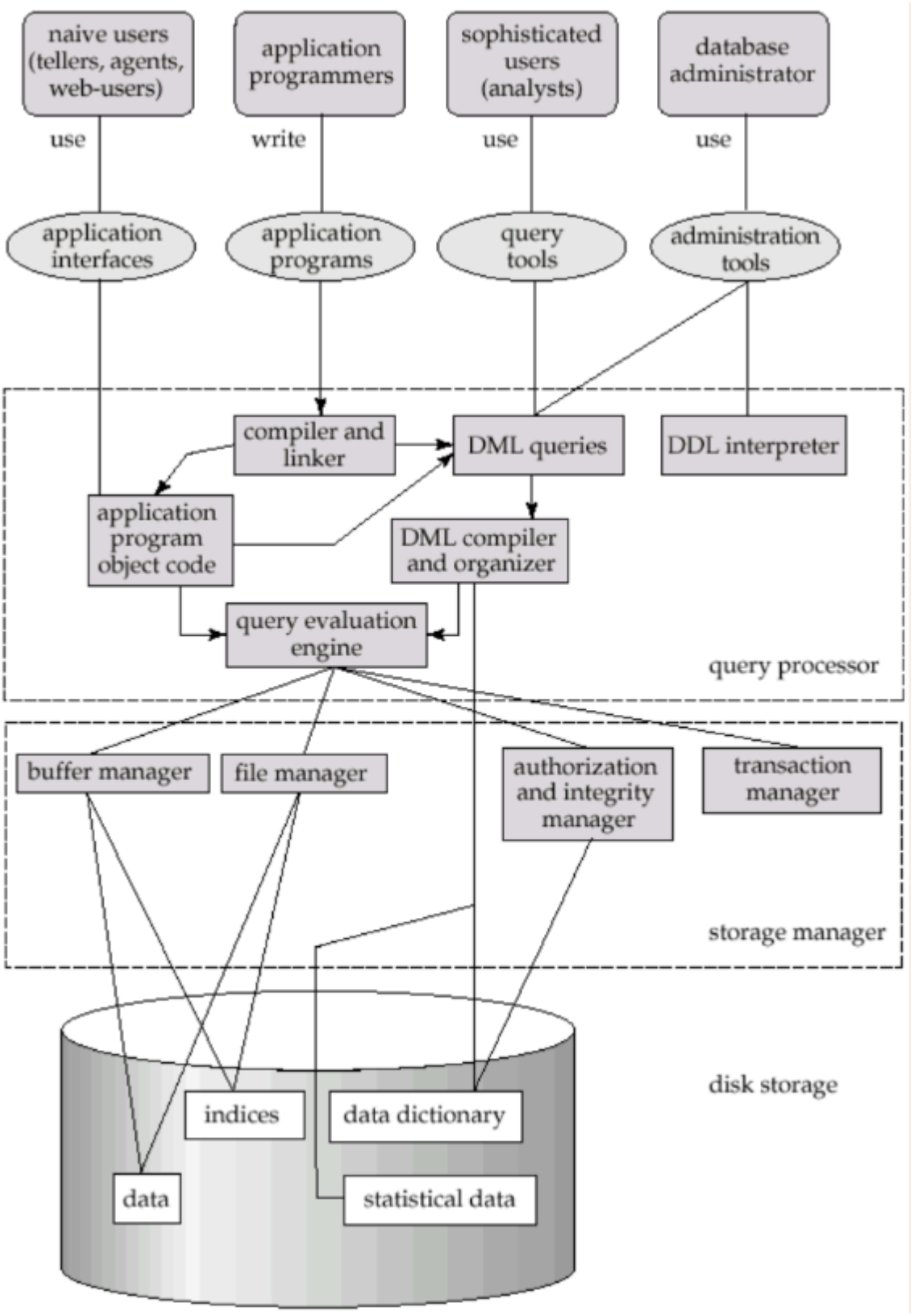
and implementation of the internal level including the storage structure and access methods to be used for the optimum performance of the DBMS.
    - Changes to any of the 3 levels necessitated by changes or growth in the organisation or emerging technology are under the control of DBA
    - Mappings between the internal and conceptual levels, as well as between the conceptual and external levels are also defined by the DBA
    - Ensuring that appropriate measures are in place to maintain the integrity of the database and that the database is not accessible to unauthorised users is another responsibility.
    - The DBA is responsible for granting permission to the users of the database and stores the profile of each user in the database.
    - The DBA is also responsible for defining procedures to recover the database from failures due to human, natural, hardware causes with minimal loss of data.
    - The Data Administrator (DA) is the person who makes the strategic and policy decisions regarding the data of the enterprise and the Database Administrator (DBA) is the person who provides the necessary technical support for implementing these decisions.

    Thus the DBA is responsible for the overall control of the system at a technical level.
9. Explain the Users of DBMS

    The users of a database system can be classified according to the degree of expertise or the mode of their interactions with the DBMS.
    - Naive Users - Users who need not be aware of the presence of the database system or any other system supporting their usage are considered naïve users. A user of an automatic teller machine falls in this category.Other such users are end users of the database who work through menu oriented application programs where the type and range of response is always indicated to the user.

- Online Users - These are the users who may communicate with the database directly via an online terminal or indirectly via a user interface and application program.These users are aware of the presence of the database system and may have acquired a certain amount of expertise in the limited interaction with the database through the application program. The more sophisticated of these users may also use a data manipulation language to manipulate the database directly. Online users can also be naïve users requiring additional help such as menus.
- Application Programmers - Professional programmers who are responsible for developing application programs or user interfaces utilised by the naïve and online users fall into this category. The application program could be written in a general purpose programming language such as C, Assembler, FORTRAN, Pascal etc. and can include commands required to manipulate the database.
- Database Administrator - Centralised control of the database is exerted by a person or group of persons under the supervision of a high level administrator. This person or group is referred to as the Database Administrator(DBA). They are the users who are most familiar with the database and are responsible for creating, modifying and maintaining its 3 levels.

10. Components of DBMS



A database system is partitioned into modules that deal with each of the responsibilities of the overall system. The functional components

of a database system are broadly divided into the storage manager and query processor components.

- Storage Manager - A storage manager is a program module that provides the interface between the low level data stored in the database and the application programs and queries submitted to the system.
- Components of Storage Manager include:
  - Authorization and Integrity Manager - Which tests for the satisfaction of integrity constraints and check the authority of users to access data
  - Transaction Manager - Which ensures that the database system remains in a consistent state despite system failures, and that concurrent transaction execution proceed without conflicting
  - File Manager - Which manages the allocation of space on disk storage and the data structures used to represent information stored on disk.
  - Buffer Manager - Which is responsible for fetching data from disk storage into main memory and deciding what data to cache in main memory. The buffer manager is a critical part of the database system, since it enables the database to handle data sizes that are much larger than the size of the main memory.
- Query Processor - The query processor is used to interpret the online user's query and convert it into an efficient series of operations in a form capable of being sent to the data manager for execution. The query processor uses the data dictionary to find the structure of the relevant portion of the database and uses this information in modifying the query and preparing an optimal plan to access the database.
- Query Processor includes
  - DDL Interpreter - Which interprets DDL statements and records the definitions in the data dictionary
  - DML PreCompiler - Which translates DML statements in a query language into an evaluation plan consisting of low – level instructions that the query evaluation engine understands. A query can usually be translated

into any number of alternative evaluation plans that all give the same result. The DML compiler also performs query optimization, that is, it picks the lowest cost evaluation plan from among the alternatives.
        - Embedded DML Pre compiler - It converts DML statements embedded in an application program to normal procedure calls in the host language. The pre compiler must interact with the DML compiler to generate the appropriate code.
        - Query Evaluation Engine - Which executes low-level instructions generated by the DML compiler.
    - Disk Storage - The storage manager implements several data structures as part of the physical system implementation. Following Data structures are required as a part of the physical system implementation.
        - Data Files - Which stores the database itself. It contains the data portion of the database.
        - Data Dictionary - Which stores metadata about the structure of the database, in particular the schema of the database.
        - Indices - Which provides fast access to data items that hold particular values
        - Statistical Data - It stores statistical data /Information about the data in the database. This information is used by the query processor to select efficient ways to execute the query.
11. Storage Manager
    - The storage manager is important because databases typically require a large amount of storage space.
    - A storage manager is a program module that provides the interface between the low level data stored in the database and the application programs and queries submitted to the system.
    - It is also responsible for interaction with the file manager.
    - The raw data are stored on the disk using the file system, which is usually provided by a conventional operating system.

- The storage manager translates the various DML statements into low-level file-system commands.
- Thus the storage manager is responsible for storing, retrieving and updating data in the database

12. Responsibilities of Storage Manager

    Following are the responsibilities of Database Manager or Storage Manager

    - Interaction with the file Manager - Actual data is stored in the file system. The storage manager translates the various DML statements into low level file system commands. This database manager is responsible for actual storing, retrieving and updating of the data in the database.
    - Integrity Enforcement - Consistency constraints are specified by Database Administrator. But the responsibility of database manager is to enforce, implement or check these constraints.
    - Backup and Recovery - It is the responsibility of database manager to detect system failures and restore the database to a consistent state.
    - Concurrency Control - Interaction among the concurrent users is controlled by a database manager. Thus storage manager or database manager is responsible for
        - Storing the data
        - Retrieving the data
        - Updating the data in the database

13. Explain the Hierarchical Data Model
    - In a hierarchical database the data is organised in a hierarchical or ordered tree structure and the database is a collection of such disjoint trees (sometimes referred to as a forest or spanning trees).
    - The nodes of the trees represent record types. Each tree effectively represents a root record type and all of its dependent record types.
    - The hierarchical database can be represented by a structure in which the records are represented by rectangular boxes and the relationship between records by links pointing from root towards leaf.

- Such structured diagrams are called tree structured diagrams, definition trees or hierarchical definition trees.
- The hierarchical model provides a straightforward or natural method of implementing one to many relationships, However M: N relationships can't be expressed directly in the hierarchical model.
- Such a relationship can be expressed by using data replication or virtual records.
- The disadvantages of data replication are wastage of storage space and the problems of maintaining data consistencies.
- In hierarchical structure diagrams one can assume that the DBMS provides a single occurrence of a dummy record type and all the hierarchical trees can then be attached to this single dummy parent record. The roots of these trees can be treated as children of this dummy record.
- The data manipulation facility of the hierarchical model provides functions similar to the network approach; however the navigation to be provided is based on the hierarchical model.

14. Virtual Records
    -
15. Explain The Network Model
    - In the network model data is represented by collection of records and relationships among data are represented by links. Thus it is called a record based model.
    - Network database model was created to represent complex data relationships more effectively than HDM could, to improve database performance and to impose database standards.
    - Using Network database terminology, a relationship is called a set.
    - Each set is composed of at least two record types, an owner record that is equivalent to HDM's parent and member record that is equivalent to the HDM's child.
    - The difference between HDM and NDM is that the NDM might include a condition in which a record can appear (as a

member) in more than one set. In other words a member may have several owners.

16. Advantages of Network Model
    - In NDM 1: N, M: M, 1:1 relationships can be implemented. Thus NDM handles more relationship types.
    - Data duplication is avoided.
    - ER to network mapping is easier than in HDM
    - Like the HDM the conceptual view of the database is simple, thus promoting design simplicity.
    - Data access flexibility: It is superior to that found in HDM in the file system. An application can access an owner record and all the member records within a set.
    - Promotes data integrity: The NDM enforces database integrity because the user must first define the owner record and then the member (A member cannot exist without owner)
    - Data Independence: Changes in the data characteristics do not require changes in the data access portions of the application programs.
    - Conformance to standards: The network database's existence is at least partially traceable to the database standards imposed in the seventies. These standards include a DDL thus greatly facilitating administration and portability.
17. Disadvantages of Network Model
    - System Complexity - Database integrity control and the efficiency with which the network model manages relationships are sometimes short circuited by the system complexity. Like the HDM, the NDM provides a navigational data access environment, in which data are accessed one record at a time. Consequently DBA, programmers and end users must be very familiar with the internal structures in order to access the database.
    - Lacks of structural Independence -- Some structural changes are impossible to make in a network database. If changes are made to the database structure, all application programs must be revalidated before they can access the database. In

short, although the network model achieves data independence, it still does not produce structural independence.

18. Explain The Relation Data Model
    - The relational model, first developed by E.F.Codd (of IBM) in 1970 represented a major break for both designers and users. The RDM, after more than a decade has emerged from the research, development, test and trial stages as a commercial product.
    - Software systems using this approach are available for all sizes of computer systems. This model has the advantage of being simple in principle and users can express their queries in powerful query language.
    - The relational data model, like all data models consist of 3 basic components:
        - A set of domains and a set of relation
        - Operation on relations
        - Integrity rules.
    - A relational database consist of a collection of tables, each of which is assigned a unique name
    - Since a table is a collection of such relationships, there is a close correspondence between the concept of a table and mathematical concept of relation, from which relational data model takes its name.
    - The basic building blocks of relational systems are attributes and the domains on which they are defined.
    - An object or entity is characterised by its properties (or attributes) whereas we define domain Di as a set of values of the same data type.
19. Define Domains
    - We define domain Di as a set of values of the same data type.
    - A domain may be unstructured (atomic) or structured.
    - Domain Di is said to be simple if all its elements are non decomposable (i.e. atomic).

- In typical DBMS systems atomic domains are general sets, such as the set of integers, real numbers, character strings and so on.
- Atomic domains are sometimes referred to as application-Independent domains because these general sets are not dependent on a particular application.
- Structured or composite domains can be specified as consisting of nonatomic values.
- Attributes are defined on some underlying domain.That is they can assume values from the set of values in the domain.
- Attributes defined on the same domain are comparable as these attributes draw their values from the same set .It is meaningless to compare attributes defined on different domains.

20. Define Tuples
    - An entity type having n attributes can be represented by an ordered set of these attributes called an n-tuple.A tuple is comparable to a record in a conventional file systems and is used for handling relationship between entities.Tuples are generally denoted by lowercase letters r,s,t...of the alphabet. An n-tuple t can be specified as t = (ai, .an)
21. Why RDBMS?
    - The relational model frees the user from the details of storage structures and access methods.
    - It is also conceptually simple and more importantly based on sound theoretical principles that provide formal tools to tackle problems arising in database design and maintenance.
    - In this model, the relation is the only construct required to represent the associations among the attributes of an entity as well as relationships among different entities.
    - One of the main reasons for introducing this model was to increase the productivity of the application programmer by eliminating the need to change the application program when a change is made to the database.

- Users need not know exact physical structures to use the database and are protected from any changes made to these structures.
- They are however still required to know how the data has been positioned into the various relations.

22. Relational Database Characteristics
    - The relational model is an example of a record-based model. Record based models are so named because the database is structured in fixed format records of several types.
    - Each table contains records of a particular type.
    - Each record type defines a fixed number of fields, or attributes.
    - The columns of the table correspond to the attributes of the record types.
    - The relational data model is the most widely used data model, and a vast majority of current database systems are based on the relational model.
23. Explain Codd's Rules
    - Rule 0: The Foundation Rule - For any system to be called a RDBMS, it must be able to manage databases entirely through its relational capabilities.
    - Rule 1: The Information Rule - All information in a RDBMS is represented explicitly (at the logical level) in exactly one way, by values in table
    - Rule 2: The Guaranteed Access Rule - Each and every datum (atomic values) is logically accessible through a combination of table name, column name and primary key value.
    - Rule 3: Systematic Treatment Of NULL Value - Inapplicable or missing information can be represented through null values. This rule specifies that RDBMS should support representation of null in place of column values to represent 'unknown values' and/or 'inapplicable values'.
    - Rule 4: Active/Dynamic Online Catalog Based On The Relational Model - The description of the database is held in the same way as ordinary data that is in tables and columns and is accessible to authorised users.

- Rule 5: The Comprehensive Data Sublanguage Rule - There must be at least one language which is comprehensive in supporting data definition, view definition, data manipulation, integrity constraints, and authorization and transaction control.
- Rule 6: The View Updating Rule - All views that are theoretically updateable are updated by the system.
- Rule 7: High-Level Insert, Update & Delete Rule - The capability of handling a base or derived table as a single operand applies not only to the retrieval of data but also to the insertion, updation and deletion of data. This means that all SELECT, UPDATE, DELETE, INSERT or their equivalents must be available and operate on a set of rows in any relation.
- Rule 8: Physical Data Independence - Application programs and terminal activity remain logically unimpaired whenever any changes are made in the storage representation and access method.
- Rule 9: Logical Data Independence - Application programs and terminal activities remain logically unimpaired when information persevering changes of any kind that theoretically permit impairment are made to the base table.
- Rule 10: Integrity Independence - This specifies that RDBMS should support all integrity constraints such as Entity Integrity & Referential Integrity
- Rule 11: Distribution Independence - The system must have a data sublanguage which can support distributed databases without impairing application programs or terminal activities.
- Rule 12: The Non-subversion Rule - If the system has a low level (record at a time) language, this language cannot be used to bypass the integrity rules and constraints expressed in the higher level relational language.

24. Relational Database Objects

Various types of objects can be found in a relational database. Some of the most common objects found in a relational database include

- Table – A table is the primary object used to store data in a relational database. When data is queried and accessed for modification, it is usually found in a table. A table is defined by columns. One occurrence of all columns in a table is called a row of data.
- View - A view is a virtual table, in that it looks like and acts like a table. A view is defined based on the structure and data of a table. A view can be queried and sometimes updated.
- Constraint - A constraint is an object used to place rules on data. Constraints are used to control the allowed data in a column. Constraints are created at the column level and also used to enforce referential integrity (parent and child table relationships)
- Index - An index is an object that is used to speed the process of data retrieval on a table. For example, an index might be created on a customer's name if users tend to search for customers by name. The customer names would be stored alphabetically in the index. The rows in the index would point to the corresponding rows in the table, much like an index in a book points to a particular page.
- Trigger - A trigger is a stored unit of programming code in the database that is fired based on an event that occurs in the database. When a trigger is fired, data might be modified based on other data that is accessed or modified. Triggers are useful for maintaining redundant data.
- Procedure - A procedure is a program that is stored in the database. A procedure is executed at the database level. Procedures are typically used to manage data and for batch processing.

The first four objects deal with the definition of the database, whereas the last two objects deal with methods for accessing database objects. Objects in a relational database provide users with a logical representation of data, such that the physical location of the data is immaterial to the user.

25. Entity Relationship Model

- ER model is a popular high level conceptual data model. Earlier commercial systems were based on the hierarchical and network approach. The ER model is a generalisation of these models.
- It allows the representation of explicit constraints as well as relationships.
- Even though the ER model has some means of describing the physical database model, it is basically useful in the design and communication of the logical database model.
- In this model, objects of similar structure are collected into entity sets.
- The relationship between entity sets is represented by a named ER relationship and is 1:1, 1:M or M:N., mapping from one entity set to another.
- The database structure, employing the ER model is usually shown pictorially using ERDs.
- ER model consist of basic objects, called entities and relationship among these objects. It represents the overall logical structure of a database.
- ER model was introduced by Chen in 1976 and refined in various ways by Chen and numerous others since that time.It is one of the several semantic data models

26. Define Entity Set
    - An Entity is a thing or object in the real world that is distinguishable from all other objects. It is an object of interest to an organisation.An Entity Set is a set of entities of the same type that share the same properties or attributes.
27. Define Relationship Sets
    - A relationship is an association among several entities.A relationship set is a set of relationships of the same type.
    - When 2 entities are involved in a relationship it is called a binary relationship. A binary relationship can be one-to-one, one-to-many, many-to-many.
    - When 3 entities are involved in it it is ternary relationship. When N entities are involved then it is a N-ary relationship.
28. Define Attributes

- An entity is represented by a set of attributes. Attributes are properties of each member of entity set.Formally, an attribute of an entity set is a function that maps from the entity set into a domain.For each attribute, there is a set of permitted values, called the domain, or value set of that attribute.

29. Types of Attributes

An attribute as used in ER model can be characterised by the following attribute types.

- Simple and Composite Attributes -
  - Simple attributes are attributes which are not further divided into subparts.
  - On other hand, composite attributes can be divided into subparts. (I.e. other attributes).Thus composite attributes help us to group together related attributes.
  - E.g. Attribute student-name could be structured as composite attribute consisting of first-name, middle-initial and last name.
- Single Valued and Multivalued Attributes
  - Attributes having single value for a particular entity are single valued.
  - Attributes that have a set of values for a specific entity are multivalued.
- Derived Attributes
  - The value for this type of attribute can be derived from the values of other related attributes or entities.An attribute takes null value when an entity does not have a value for it.

30. Define Strong & Weak Entities

- An entity set that has a primary key is termed as a strong entity set. E.g.Employee
- An entity that does not have sufficient attributes to form a primary key is termed as a weak entity set. E.g. Dependent.

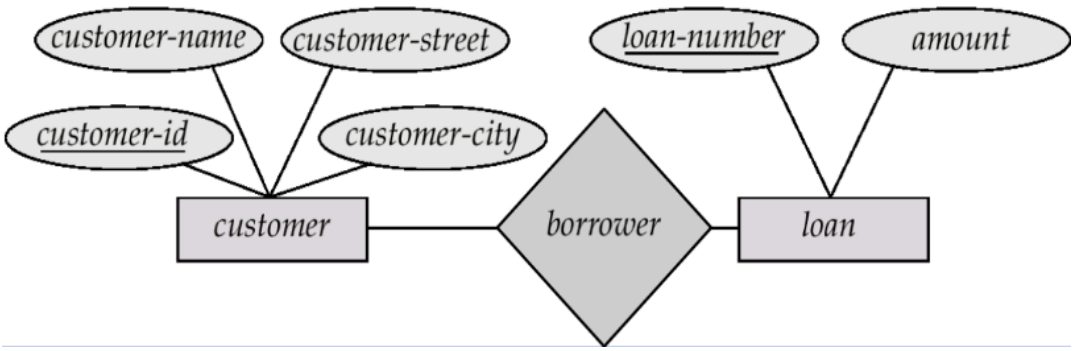
31. Types of Keys

- A key is a single attribute or combination of two or more attributes of an entity set that is used to identify one or more instances of the set.

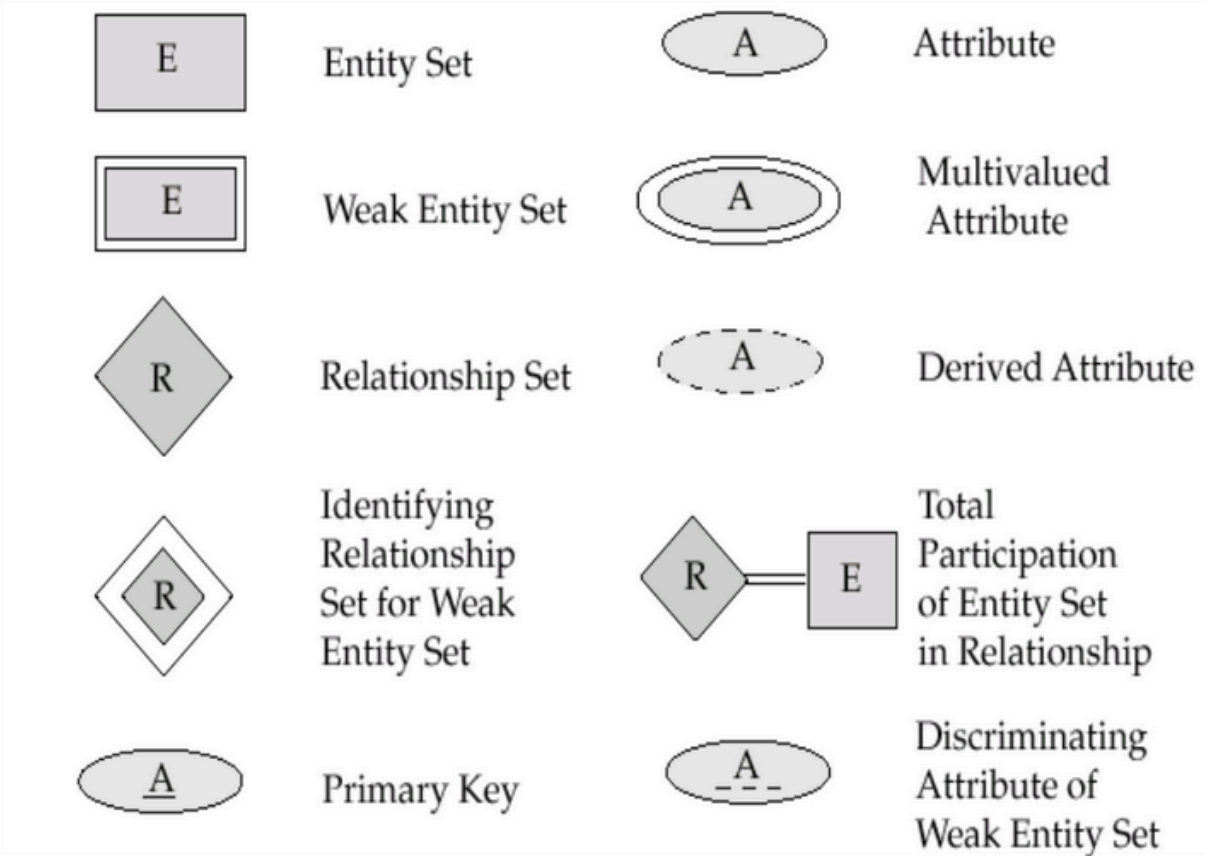
- Primary Key - The attribute of entity set which identifies and distinguishes instances of entity set is called primary key.A primary key is a minimal super key
- Super Key - If we add additional attributes to a primary key, the resulting combination would still uniquely identify an instance of the entity set. Such keys are called super keys.
- Candidate Keys - There may be two or more attributes or combinations of attributes that uniquely identify an instance of an entity set. These attributes or combinations of attributes that uniquely identify an instance of an entity set are called candidate keys.A super key is a superset of a candidate key.
- Alternate Keys - There may be two or more attributes or combinations of attributes that uniquely identify an instance of an entity set,In such a case we must decide which of the candidate keys will be used as the primary key. The remaining candidate keys would be considered alternate keys.
- Secondary Key - A secondary key is an attribute or combination of attributes that may not be a candidate key but that classifies the entity set on a particular characteristic.

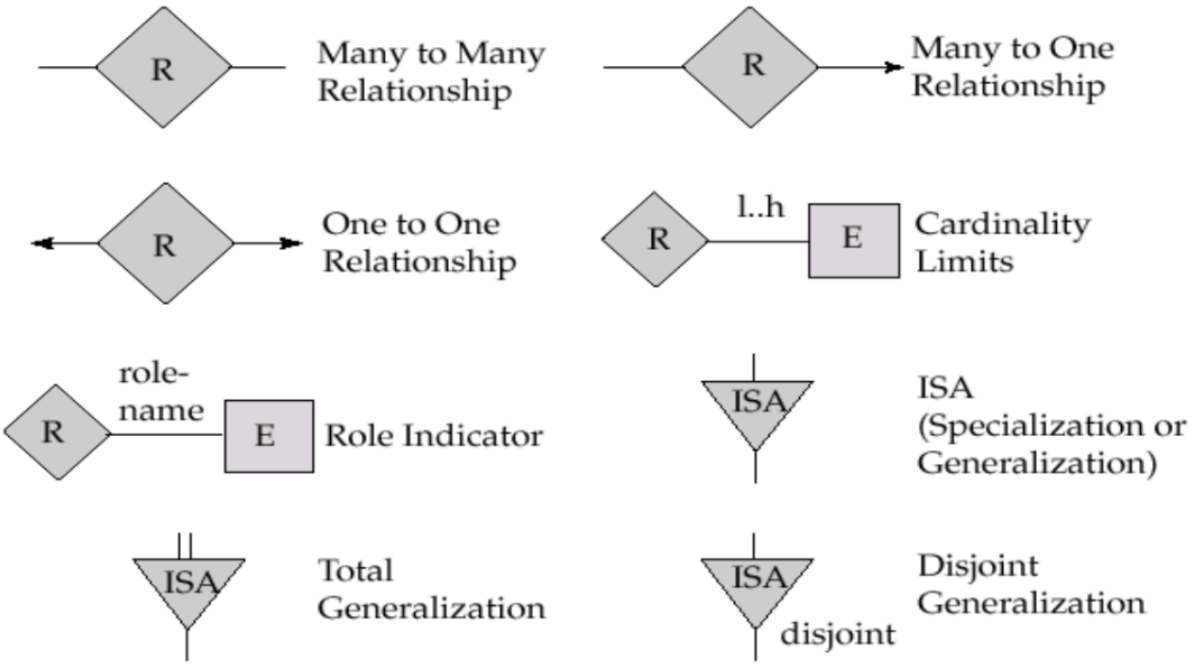
32. Conventions for Drawing ERD
    - An entity is shown as a rectangle
    - A diamond represents the relationship among a number of entities, which are connected to the diamond by lines.
    - The attributes, shown as ovals, are connected to the entities or relationship by lines.
    - Diamonds, ovals and rectangles are labelled.
    - The type of relationship existing between the entities is represented by giving the cardinality of the relationship on the line joining the relationship to the entity.

- Ex.



33. Symbols used while drawing ERD

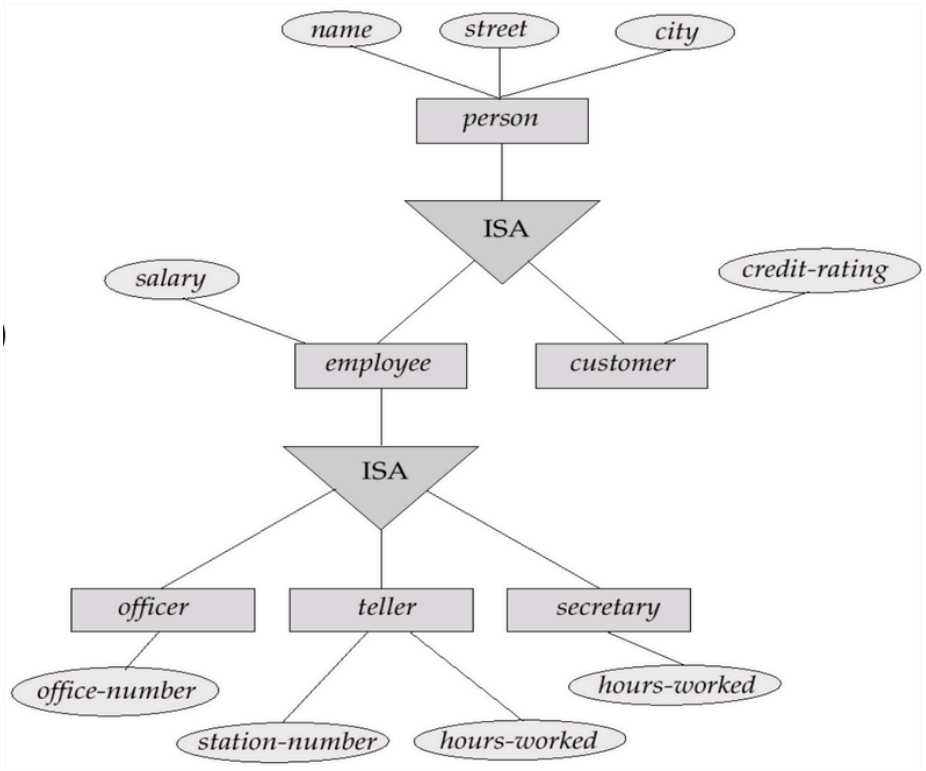




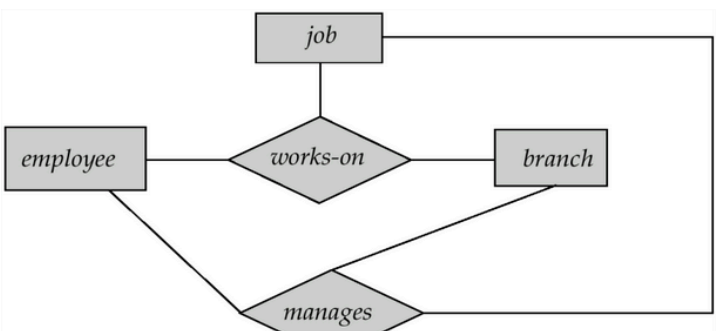
34. Explain Abstraction

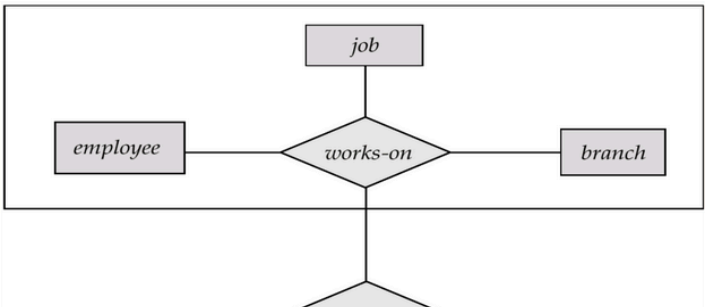
- Abstraction is the simplification mechanism used to hide superfluous details of a set of objects; it allows one to concentrate on the properties that are of interest to the application.
- Following are the Main Abstraction Mechanisms used to model information:
- Generalisation -
  - Generalisation is the abstract process of viewing a set of objects as a single general class by concentrating on general characteristics of the constituent sets while suppressing or ignoring their differences.
  - It is the union of a number of lower level entity types for the purpose of producing a higher level entity type.
  - Generalisation is a containment relationship that exists between a higher level entity set and one or more low-level entity sets.
  - E.g. Employee is a generalisation of the classes of objects: cook,waiter,cashier etc. Student is a generalisation of graduate or undergraduate, full-time or part- time students
- Specialisation -

- The process of designating sub groupings within an entity set is specialisation.
- It is the abstracting process of introducing new characteristics to an existing class of lower level entities that also inherits the characteristics of higher level entities.
- Specialisation may be seen as the reverse process of generalisation.
- Additional specific properties are introduced at a lower level in a hierarchy of objects.



- Aggregation -
  - One limitation of the ER model is that it is not possible to express relationships among relationships.
  - Aggregation is an abstraction through which relationships are treated as higher level entities.
  - It helps us to express relationships among relationships.
  - It is the process of compiling information on an object, thereby abstracting a higher-level object.





Entity Relationship Diagram with aggregation

35. Explain ACID Properties
    - To ensure the integrity of data, database system maintains following properties of transaction. These are called ACID properties.
    - Atomicity - Atomicity property ensures that at the end of transaction, either no changes have occurred to the database or the database has been changed in a consistent manner. At the end of the transaction, the updates made by the transaction will be accessible to other transactions and processes outside the transaction.
    - Consistency - Consistency property of transaction implies that if the database was in consistent state before the start of transaction, then on termination of transaction the database will also be in a consistent state
    - Isolation - This property indicates that action performed by a transaction will be hidden from outside the transaction until the transaction terminates. Thus each transaction is unaware of another transaction executing concurrently in the system.
    - Durability - Durability ensures that once a transaction completes, i.e. successfully commits, the changes it has made to the database persist, even if there are system failures
36. What is Concurrency and list the problems that may occur.
    - In a multi-user system, at a time more than one user tries to make transactions is called as concurrency

- So when several transactions are executed concurrently, the corresponding schedule is called concurrent schedule .But due to concurrency following problems may occur:
  - Lost Update Problem
  - Inconsistent/Dirty Read Problem
  - Semantics of Concurrent Transaction
  - The Phantom Phenomenon

37. List Concurrency Control Techniques
    - Time Stamp Ordering
    - Locking Protocol
    - Validation Techniques
    - Multiversion Techniques(MVT)
38. Explain Recovery Mechanisms in DBMS
    - The types of failures that the computer system is likely to be subjected to include failures of components or subsystems, s/w failures, power outages, accidents, unforeseen situations and the natural or manmade disasters.
    - Database recovery techniques are the methods of making the database fault tolerant.
    - The aim of the recovery scheme is to allow database operations to be resumed after a failure with minimum loss of information at an economically justifiable cost.
    - Different Recovery Mechanisms in centralised DBMS are:
      - Log Based Recovery - A log, which is usually written to the stable storage, contains the redundant data required to recover from volatile storage failures and also from errors discovered by the transaction or the database system.
      - Checkpoints - In an on-line database system, for example in airline reservation system, there could be hundreds of transactions handled per minute. The log for this type of database contains a very large volume of information. A scheme called checkpoint is used to limit the volume of log information that has to be handled and processed in the event of a system failure involving loss of volatile information. The checkpoint

scheme is an additional component of the logging scheme described above.

- Shadow Page Table - Shadow paging is an alternative to log-based recovery; this scheme is useful if transactions execute serially. Idea: maintain two page tables during the lifetime of a transaction – the current page table, and the shadow page table.Store the shadow page table in nonvolatile storage, such that state of the database prior to transaction execution may be recovered.Shadow page table is never modified during execution

39. Explain View in DBMS
    - It is not desirable for all users to see the entire logical model. Security considerations may require that certain data be hidden from users.
    - Any relation that is not part of the logical model, but is made visible to the user as a virtual relation, is called a view.
    - It is possible to support a large number of views on top of any given set of actual relations.
    - We define a view in SQL by using the create view command. To define a view, we must give the view a name and must state the query that computes the view. The form of the create view command is Create view v as <query expression>
    - A view can hide data that a user does not need to see. The ability of views to hide data serves both to simplify usage of the system and to enhance security.
    - Views simplify system usage because they restrict the user's attention to the data of interest.
    - Although a user may be denied direct access to a relation, that user may be allowed to access part of that relation through a view.
    - Thus a combination of relational-level security and view-level security limits a user's access to precisely the data that the user needs.
40. Integrity Constraints

- Integrity constraints provide a means of ensuring that changes made to the database by authorised users do not result in a loss of data consistency. Thus integrity constraints guard against accidental damage to the database.
- (Integrity Rule 1)Entity Integrity - Integrity rule 1 is concerned with primary key values.Integrity rule 1 specifies that instances of the entities are distinguishable and thus no prime attribute (component of a primary key) value may be null. This rule is also referred to as the entity rule. We could state this rule formally as If attribute A of relation R(R) is a prime attribute of R(R), then A cannot accept null values.
- (Integrity Rule 2)Referential Integrity - It is concerned with foreign keys i.e. with attributes of a relation having domains that are those of the primary key of another relation.Given two relations R and S, suppose R refers to the relation S via a set of attributes that forms the primary key of S and this set of attributes forms a foreign key in R. Then the value of the foreign key in a tuple R must either be equal to the primary key of a tuple of S or be entirely null
- Domain or Data Item value Integrity Rule - One aspect that has to be dealt with by the integrity subsystem is to ensure that only valid values can be assigned to each data item. This is referred to as Domain integrity. Domain integrity rules are simply the definition of the domains of the attributes or the value set for the data-items.

41. Data Security
    - Data Security is nothing but protection from malicious attempts to steal or modify data.
    - Database system level
        - Authentication and authorization mechanisms to allow specific users access only to required data
        - We concentrate on authorization in the rest of this chapter
    - Operating system level
        - Operating system super-users can do anything they want to the database
        - Good operating system level security is required.

- Network level: must use encryption to prevent
  - Eavesdropping (unauthorised reading of messages)
  - Masquerading (pretending to be an authorised user or sending messages supposedly from authorised users)
- Physical level
  - Physical access to computers allows destruction of data by Intruders; traditional lock-and-key security is needed
  - Computers must also be protected from floods, fire, etc.
- Human level
  - Users must be screened to ensure that authorised users do not Give access to intruders
  - Users should be trained on password selection and secrecy

42. What are the undesirable properties of Bad Database Design?
    - A database that is not normalised can include data that is contained in one or more different tables for no apparent reason.
    - This is not optional design with regard to security reasons, disk space usage, query retrieval speed, efficiency of database updates and most importantly data integrity.
    - Ex. Consider foll database design

EMPLOYEE

Empno	Fname	Lname	Deptname	Location
1	Chetan	Kelkar	Sales	Pune
2	Pushkar	Kate	Marketing	Mumbai
3	Mukta	Kunte	Sales	Pune
4	Deepti	Joshi	Marketing	Mumbai
5	Medha	Pawar	Purchase	Banglore
6	Amit	Sathe	Purchase	Banglore
7	Rajesh	Mohite	Accounts	Nasik

    - Repetition of Information or Redundancy - The aim of the database system is to reduce the redundancy i.e. the information stored must not be repeated. Repetition of information wastes space.In above data Deptname (i e

Department name) and Location information is repeated for a number of times.

- Various types of Anomalies -
  - Insertion Anomaly - Insertion of new department is not possible in above data if no corresponding employee is working. And vice versa (i.e. if new Administration Department is to be added and no single employee is assigned to it then we cannot enter information of Administration department unless and until some employee is attached to it)
  - Updating anomaly - If any of the department or location information is to be updated, then corresponding change has to be done for number of times. (I.e. if purchase department get shifted from Bangalore to pune, corresponding change has to be done for number of times, depending on number of records)
  - Deletion anomaly - Deletion of employee's record having corresponding unique department information (I.e. In above data if Rajesh's record is deleted corresponding account department's information will get lost). This leads to deletion anomalies.
- Thus the above database design has following undesirable properties

43. Short note on Functional Dependencies

- Functional dependencies are constraints on the set of legal relations.
- They allow us to express facts about the enterprise that we are modelling with our database.
- The notion of functional dependency generalises the notion of super key.
- Functional dependency can be described as follows. If column2 is functionally dependent on column1, it means that each value in column1 is associated with one and only one value in column2 at a particular point in time. A particular value for column1 will always have the same value in column2. The reverse is not true, however. A particular value

for column2 can have multiple corresponding values in column1.
    - In other words functional dependency can be described as, in relation R if there is a set of attributes called x and other set called y.
    - For each occurrence of x there will be only one occurrence of y. In such cases y is called as functionally dependent on x.
    - R=X→ Y

44. Explain Normalisation
    - Any DBMS strives to reduce the anomalies in the database resulting in duplication or loss of data.
    - Thus a scientific technique called Normalization is used to design a file which will represent database design in a normal form.
    - Thus normalization represents good database design which eliminates the anomalies and inconsistencies.
    - Normalization of data can be looked on as a process during which unsatisfactory relation schemas are decomposed by breaking up their attributes into smaller relation schemas that possess desirable properties
    - Normalization is the process of reducing redundancy of data in a relational database. Redundant data refers to the data that is stored in more than one location in the database.
    - Normalization was defined by E.F.Codd as a mathematical basis for deriving tables (rows and columns of data) using relational calculus and a set of transformations that could be applied to data structures to ensure that they met the requirements for a Relational DBMS.
    - There are two approaches to normalization.
        - Decomposition
        - Synthesis
    - The process of converting one entity into multiple entities in order to normalize the original is called decomposition. An unnormalized single entity will be decomposed into two or more normalized entities via the application of normalization rules.

- The decomposition approach starts with one relation and associated set of constraints in the form of
  - Functional dependencies
  - Multivalued dependencies
  - Join dependencies
- A relation that has any undesirable properties in the form of insertion, deletion or update anomalies, repetition of information or loss of information and replaced by its projections. This results in normal form.
- The second approach is the synthesis approach. It starts with a set of functional dependencies on a set of attributes. It then synthesises relations of third normal form(3NF).It is also known as transitivity dependency.

45. Advantages of Normalisation

Normalization provides numerous benefits to the design of a database, the design and implementation of an application.

- It gives refined file design and greater overall database organization will be gained.
- It removes undesirable elements (fields) i.e. the amount of unnecessary redundant data is reduced.
- It helps in understanding the system
- It avoids anomalies
- Data integrity is easily maintained within the database.
- The database and application design processes are much more flexible.
- Security is easier to manage.

46. Disadvantages of Normalisation

- The disadvantages of normalization are that it produces a lot of tables with a relatively small number of columns. These columns then have to be joined using their primary/foreign key relationships in order to put the information back together so we can use it.
- Decomposition of tables has two primary impacts.
- The first is performance. All the joins required to merge data slow processing down and place additional stress on your hardware.

- The second impact challenges developers to code queries that return desired information, without experiencing the impact of the relational database's insistence on returning a row for every possible combination of matching values if the tables are not properly joined by the developer.
- Additional rows that are returned because of tables that are not properly joined (using their key values) are not useful.

47. Rules of Normalisation
    - Some normalization rules are there which are used as guidelines to normalization
    - First Normal Form
        - The objective of the first normal form is to divide the base data into logical units called entities, or tables.
        - When each entity has been designed, a primary key is assigned to it. It is also called the 'Atomic value rule'.
        - Thus 1st NF states that each attribute in a database must have an atomic value. Multivalued attributes are not allowed in 1st NF.
        - A Relation schema is said to be in 1 NF (First normal form) if the values in the domain of each attribute of the relation are atomic. In other words, only one value is associated with each attribute and the value is not a set of values or list of values.
        - A table is said to be in first normal form if it has no repeating groups of attributes. In other words, if there is a group of attributes that is repeated, remove it to a separate table.
        - A database schema is in First Normal Form if every relation schema included in the database schema is in 1st NF.
        - A 1st NF disallows having a set of values, a tuple of values or combination of both as an attribute value for a single tuple. In other words 1st NF disallows "relations within relations" or "relations as attributes of tuples". The only attribute values permitted by 1st NF are single atomic (or indivisible) values.

        - The 1st NF also disallows composite attributes that are themselves multivalued. These are called nested relations because each tuple can have a relation within it.
    - Second Normal Form
        - It is based on functional dependency .It says that your file design must be in 1st NF, and all non key attributes must be fully functionally dependent on prime attribute(s) or key attribute(s).
        - A relation schema is in second normal form if it is in 1 NF and if all nonprime attributes are fully functionally dependent on the relation key(s).i.e. Prime attributes or key attributes.
        - A database schema is in 2nd NF if every relation schema included in the database schema is in second normal form.
        - Each non key attribute within a table with a multiple attribute key is examined to transform a set of tables into second normal form.
        - Thus the objective of the Second Normal Form is to take data that is only partly dependent on the primary key and enter it into another table
    - Third Normal Form(Transitivity Dependency)
        - It says that the file design must be in 2nd NF. It also states that non key attributes must be fully fully functionally and non transitively dependent on primary key attribute(s)
        - A table is said to be in 3rd NF if it is in second normal form and every non key attribute is dependent on the entire primary key.
        - A relation schema in 3 NF does not allow partial or transitional dependency.
        - A database schema is in 3rd NF if every relational schema included in the database schema is in 3rd NF.
        - The 3rd NF schema, like the 2nd NF does not allow partial dependencies. Furthermore, unlike the 2nd NF schema, it does not allow any transitive dependency.

        - Thus the THIRD NORMAL FORM's objective is to remove data in a table that is not dependent on the primary key.The entity is in SECOND
        - NORMAL FORM and a non-UID attribute can't depend on another non-UID attribute. All non-UID attributes should depend directly on the whole UID and not on each other.
        - An attribute dependency on the UID- which is not direct but only passes through another attribute that is dependent on the UID- is called transitive dependency. Transitive dependencies are unacceptable in THIRD NORMAL FORM.
    - Fourth Normal Form
        - Fourth Normal Form is in effect when an entity is in Boyce-Codd Normal Form, it has no multivalued dependencies, and neither of the following conditions exists.
            1. The dependent attribute(s) is not a subset of the attribute(s) upon which it depends (the determinant)
            2. The determinant in union (combination) with the dependent attribute(s) includes the entire entity.
    - Fifth Normal Form
        - FIFTH NORMAL FORM (also called project-join normal form) is in effect when FOURTH NORMAL FORM is in effect and the entity has no join dependencies that are not related to that entity's candidate keys.
        - An entity is in FIFTH NORMAL FORM when it cannot be decomposed into several smaller entities, which have different keys from the original without data losses. If data was lost after decomposition, a lossless-join would not be possible.

48. Explain Boyce Codd Normal Form
    - BCNF was developed to overcome some problems with 3rd NF.
    - A relation meets the requirements of BCNF if every determinant is a candidate key.

- Remember a determinant is an attribute on which another attribute is fully functionally dependent.
- “A relation R is in BCNF if and only if every determinant is a candidate key”
- Any relation R can be non-loss decomposed into an equivalent collection of BCNF relation.
- Thus Boyce-codd NORMAL FORM is in effect when an entity is in THIRD NORMAL FORM and every attribute or combination of attributes (a determinant) upon which any attribute is functionally dependent is a candidate key (that is, unique).
- An attribute or combination of attributes upon which any attribute is functionally dependent is also called a determinant.
- If there is only one determinant upon which other attributes depend and it is a candidate key (such as a primary key), THIRD NORMAL FORM and Boyce-codd NORMAL FORM are identical.
- The difference between THIRD NORMAL FORM and Boyce-codd NORMAL FORM is that Boyce-codd requires all attributes that have other attributes with functional dependencies on them(are determinants) to be candidate keys and THIRD NORMAL FORM does not. Boyce-codd, with this subtle difference, is in effect a stronger version of THIRD NORMAL FORM.

49. Explain Domain Key Normal Form
    - The idea behind domain key normal form is to specify the “ultimate normal form” that takes into account all possible types of dependencies and constraints.
    - A relation is said to be in DKNF if all constraints and dependencies that should hold on the relation can be enforced simply by enforcing the domain constraints specified on the relation.
    - For a relation in DKNF, it becomes very straightforward to enforce the constraints by simply checking that each attribute value in a tuple is of the appropriate domain and that every key constraint on the relation is enforced.

- However it seems unlikely that complex constraints can be included in a DKNF relation, hence its practical utility is limited.

50. Explain Denormalisation
    - As opposed to normalisation, denormalization allows some redundancy in fields (for user requirements). It also allows derived fields.
    - When we denormalize data it creates redundancy but it must be controlled redundancy.
    - Occasionally database designers choose a schema that has redundant information, that is it is not normalised. They use the redundancy to improve performance for specific applications.
    - The penalty paid for not using a normalised schema is the extra work (in terms of coding time and execution time) to keep redundant data consistent.
    - The process of taking a normalised schema and making it non normalised is called denormalization.
    - Thus Denormalization is the process of taking a normalised database and modifying table structures to allow controlled redundancy for increased database performance.
    - Denormalization generally involves one of these three tasks.
        - Replicating some data columns to several tables in order to have them more easily accessible without multi-table joins.
        - Pre-calculation and storage of derived data such as summary calculations or concatenations can speed processing.
        - Undoing some decomposition of entities to avoid the price of multiple joins. This would primarily be undertaken for tables that have one-to-many relationships.
51. Explain Decomposition & Properties of Decomposition
    - The process of converting one entity into multiple entities in order to normalise the original is called decomposition.

- An unnormalized single entity will be decomposed into two or more normalised entities via the application of normalisation rules.
- Loss Less Join -

52. Short note on SQL
    - Ideally, a database language must enable us to create database and table structures, to perform basic data management chores (add, delete and modify), and to perform complex queries designed to transform the raw data into useful information
    - Structured query language, more commonly referred to as SQL, is the standard language used to issue commands to and communicate with a relational database.
    - With SQL, you can easily enter data into the database, modify data, delete and retrieve data.
    - SQL is a non-procedural language, which means that you tell the database server what data to access, not necessarily what methods to access the data.
    - SQL contains the following three sublanguages that allow you to perform nearly any operation desirable within a relational database.They are:
        - DDL - Data Definition Language
        - DML - Data Manipulation Language
        - DCL - Data Control Language
    - SQL is relatively easy to learn. Its command set has a basic vocabulary of less than 100 words.
    - It uses a combination of Relational algebra and Relational calculus.

53. Data types in SQL
    - char - Values of this data type are fixed length characters strings of maximum length 255 characters.
    - varchar/varchar2 - Values of this data type are variable length character strings of maximum length 2000.
    - number - It is used to store numbers (fixed or floating point).Numbers of virtually any magnitude may be stored upto 38 digits of precision.

- date - The standard format is DD-MM-YY. To change this format we use appropriate functions
- Long - It can store variable length character strings containing up to 65,535 characters. Long data can be used to store arrays of binary data in ASCII format.

54. Explain DDL
    - DBMS provides a facility known as Data Definition Language (DDL), which can be used to define the conceptual scheme and also give some details about how to implement this scheme in the physical devices used to store the data.
    - This definition includes all the entity sets and their associated attributes as well as the relationships among the entity sets.
    - The definition also contains any constraints that have to be maintained, including the constraints on the value that can be assigned to a given attribute and the constraints on the values assigned to different attributes in the same or different records.
    - The DDL, just like any other programming language, gets as input some instructions (statements) and generates some output. The output of the DDL is placed in the data dictionary, which contains metadata – that is data about data.
    - Data Definition Language is a part of SQL that is responsible for the creation, updation and deletion of tables. It is responsible for creation of views and indexes also.
    - Ex. CREATE TABLE, ALTER TABLE, DROP TABLE, CREATE VIEW, CREATE INDEX

55. Commands in DDL
    - The list of DDL commands is given below:
        - CREATE TABLE
          Syntax: Create table table name (columnname datatype (size), columnname datatype (size)...);
        - ALTER TABLE
          Syntax: For addition of new column
          1) Alter table tablename add (newcolumnname datatype (size), newcolumnname datatype (size),....);

2) For modifying existing column information Alter table tablename modify (columnname newdatatype (size));

- DROP TABLE
  Syntax: for deleting table
  Drop table tablename;
- CREATE VIEW
  Syntax: Create view viewname as <query expression>
- CREATE INDEX
  Syntax: Create index indexname on tablename (columnname);

56. Explain DML

- The language used to manipulate data in the database is called Data Manipulation Language (DML).
- Data manipulation involves retrieval of data from the database, insertion of new data into the database, and deletion or modification of existing data.
- The first of these data manipulation operations is called a Query. A query is a statement in the DML that requests the retrieval of data from the Database.
- The subset of the DML used to pose a query is known as Query Language.
- The DML provides commands to select and retrieve data from the database.
- Commands are also provided to insert, update and delete records.
- They could be used in an interactive mode or embedded in conventional programming languages such as Assembler, COBOL, FORTRAN, PASCAL or PL/I.
- The data manipulation functions provided by the DBMS can be invoked in application programs directly by procedure calls or by preprocessor statements.
- There are two types of DMLs:-
  - Procedural DML:-It requires a user to specify what data is needed and how to get this data.

        - Non Procedural DML:-It requires a user to specify what data are needed without specifying how to get those data.
    - Commands Include SELECT, UPDATE, DELETE, INSERT

57. Commands in DML
    - Data Manipulation commands manipulate (insert, delete, update and retrieve) data. The DML language includes commands that run queries and changes in data. It includes following commands.
    - SELECT
        - Syntax: for global data extract
          Select * from tablename;
        - Syntax: The retrieval of specific columns from a table
          Select columnname, columnname from tablename;
        - Syntax: Elimination of duplicates from the select statement
          Select distinct columnname, columname from tablename;
        - Syntax: Sorting of data in a table
          Select columnname, columnname from tablename order by columnname,columnname;
        - Syntax: Selecting a data set from table data
          select columnname, columnname from tablename where search condition;
    - UPDATE
        - Syntax: for updating the contents of a table
          update tablenameset columnname = expression, columnname = expression where columnname = expression ;
    - DELETE
        - Syntax :Deletion of all rows
          delete from tablename ;
        - Syntax:Deletion of a specified number of rows
          delete from tablename where search condition;
    - INSERT
        - Syntax: Inserting data into table from another table

Insert into tablename select columnname, columnname from tablename;

- Syntax: Insertion of selected data into table from another table
  Insert into tablename select columnname, columnname from tablename where column = expression;

58. Define Query
    - The language used to manipulate data in the database is called Data Manipulation Language (DML).
    - The first of these data manipulation operations is called a Query.
    - A query is a statement in the DML that requests the retrieval of data from the Database.
    - The subset of the DML used to pose a query is known as Query Language.
    - The Query language of SQL is nonprocedural.
    - It takes as an input several tables (possibly only one) and always returns a single table.
    - Queries may involve information from more than one table.
59. Explain DCL
    - The data control language(DCL) commands are related to the security of the database.
    - They perform tasks of assigning privileges.
    - So users can access certain objects in the database.
    - The DCL commands are:
    - GRANT
        - The grant statement provides various types of access to database objects such as tables, views and sequence to other users, which is done by the owner of those objects.
        - Syntax :- GRANT { object privileges}
          ON object Name
          To UserName
          {with GRANT OPTION}
    - REVOKE
        - The revoke statement is used to deny the grant given on an object

        - Syntax :- REVOKE {object privilege}
          ON object Name
          FROM User Name ;
    - COMMIT
        - Commit command is used to permanently record all changes that the user has made to the database since the last commit command was issued or since the beginning of the database session.
        - Syntax :- commit ;
    - ROLLBACK
        - The Rollback statement does the exact opposite of the commit statement .It ends the transaction but undoes any changes made during the transaction.
        - Rollback is useful for following 2 reasons.
            1. If you have made a mistake such as deleting the wrong row for a table ,you can use rollback to restore the original data.
            2. Rollback is useful if you have started a transaction that you cannot complete.
        - Syntax: rollback[work][to[savepoint]savepoint]

60. Define Privilege
    - We may assign a user several forms of authorization on parts of the database.
    - For example
        - Authorization to read data
        - Authorization to insert new data
        - Authorization to update data
        - Authorization to delete data
    - Each of these types of authorizations is called a privilege.
    - We may authorize the user all, none, or a combination of these types of privileges on specified parts of a database, such as a relation or a view.
    - The SQL standard includes the privileges select, insert, update and delete.

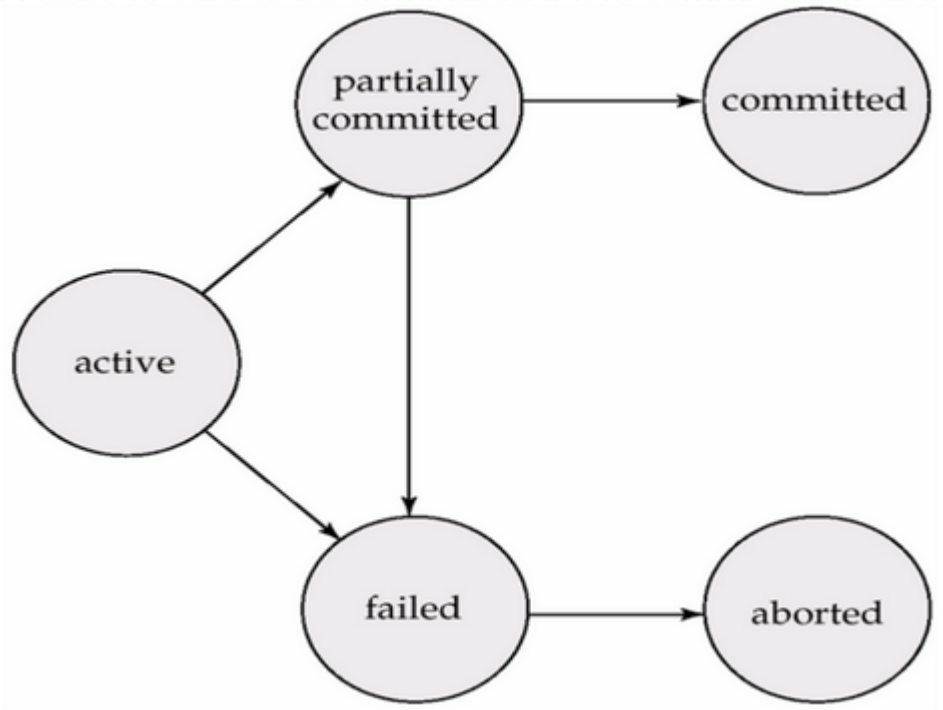
61. REVOKE statement
    - To revoke an authorization, we use the revoke statement. It takes a form almost identical to that of grant.

revoke <privilege list> on <relation name or view name > from <user /role list>[restrict/cascade]

- Revocation of privileges is more complex if the user from whom the privilege is revoked has granted the privilege to another user.
- The revocation of a privilege from a user/role may cause other users/roles also to lose that privilege.
- This behaviour is called cascading of the revoke.
- In most database systems, cascading is the default behaviour, the keyword cascade can thus be omitted.

62. Explain Transaction Model
    - A transaction is a unit of program execution that accesses and possibly updates various data items.
    - If the database was in consistent state before a transaction, then on execution of transaction, the database will be in consistent state.
    - Transaction thus basically is reading information from database and writing information to database.
    - A transaction must see a consistent database.
    - During transaction execution the database may be inconsistent.
    - When the transaction is committed, the database must be consistent.
    - Two main issues to deal with:
        - Failures of various kinds, such as hardware failures and system crashes
        - Concurrent execution of multiple transactions

63. States of Transaction



- Active - Transaction is active when it is executing. This is the initial state of the transaction.
- Partially committed - When a transaction completes its final statement but it is to be returned to the database, then it is in a partially committed state.
- Failed - If the system decides that the normal execution of the transaction can no longer proceed, the transaction is termed as failed.
- Committed - When the transaction completes its execution successfully it enters committed state from partially committed state (i.e transaction is returned to db)
- Aborted /Terminated - To ensure the atomicity property, changes made by failed transaction are undone i.e The transaction is rolled back. After roll back, that transaction enters in an aborted state. A state is said to be terminated if it is committed or aborted.

64. Concurrency
    - Transaction processing systems usually allow multiple transactions to run concurrently.
    - Allowing multiple transactions to update data concurrently causes several complications with consistency of data.
    - Ensuring concurrency in spite of concurrent execution of transactions requires extra work; it is far easier to insist that

transactions run serially – that is one at a time, each starting only after the previous one has completed.

- However there are two good reasons for allowing concurrency :
  - Improved throughput and resource utilization - Throughput means the number of transactions executed in a given amount of time Correspondingly, the processor and disk utilization also increase; in other words, the processor and disk spend less time idle, and not performing any useful work.
  - Reduced waiting time - If transactions run serially, a short transaction may have to wait for a preceding long transaction to complete, which can lead to unpredictable delays in running a transaction.
- If the transactions are operating on different parts of the database, it is better to let them run concurrently, sharing the CPU cycles and disk accesses among them.
- Concurrent execution reduces the unpredictable delays in running transactions .
- Moreover, it also reduces the average response time: the average time for a transaction to be completed after it has been submitted.
- The motivation of using concurrent execution in a database is essentially the same as the motivation for using multiprogramming in an operating system.

65. Schedules
    - Schedules are execution sequences which represent the chronological order in which instructions are executed in the system
    - Clearly, a schedule for a set of transactions must consist of all instructions of those transactions, and must preserve the order in which the instructions appear in each individual transaction.
    - A serial schedule consists of a sequence of instructions from various transactions, where the instructions belonging to one single transaction appear together in that schedule.

- Thus for a set of n transactions, there exists a number of different valid serial schedules.

66.

- When the database system executes several transactions concurrently, the corresponding schedule no longer needs to be serial.
- If two transactions are running concurrently, the operating system may execute one transaction for a little while, then perform a context switch, execute the second transaction for some time, and then switch back to the first transaction for some time, and so on. With multiple transactions, the CPU time is shared among all the transactions.
- Several execution sequences are possible, since the various instructions from both transactions may now be interleaved.
- In general, it is not possible to predict exactly how many instructions of a transaction will be executed before the CPU switches to another transaction.
- Not all concurrent executions result in a correct state.

67. What does Concurrency Control Component do?

- If control of concurrent execution is left entirely to the operating system, many possible schedules, including ones that leave the database in an inconsistent state such as the one just described, are possible.
- It is the job of the database system to ensure that any schedule that gets executed will leave the database in a consistent state.
- The concurrency control component of the database system carries out this task.

68. Serializability

- The only significant operations of a transaction, from a scheduling point of view, are its read and write instructions.
- A transaction, when executed, generates a schedule in memory. This schedule is nothing but sequential operations of the transaction.
- A serializable schedule is defined as : Given (an interleaved execution)a concurrent schedule for n transactions, the following conditions hold for each transaction in a set:

        - All transactions are correct. i.e If any of the transactions is executed on a consistent database, the resulting database is also consistent.
        - Any serial execution of the transactions is also correct and preserves the consistency of the database.
    - There are two forms of serializability:
        - Conflict Serializability
        - View Serializability
    - Precedence graph is used to test the serializability i.e it selects the schedule which will guarantee that the database will be in a consistent state, the most reliable schedule is selected, a cyclic schedule is not desirable as it may lead to deadlock.
    - Thus serializability is the result of the concurrent execution of transactions is the same as that of the transactions being executed in serial order.
    - This property is important in multiuser and distributed databases, where several transactions are likely to be executed concurrently.
    - Serializability of schedules generated by concurrently executing transactions can be ensured through one of a variety of mechanisms called concurrency control schemes.
69. Conflict Serializability
70. View Serializability
71. Types of Failures
    - Hardware Failures - Design errors, poor quality control (poor connections and defective subsystems), over utilisation and overloading, wear out (Insulation on wire could undergo chemical changes)
    - Software Failures - Design errors, poor quality control(undetected errors in entering program code), over utilisation and overloading(Buffers and stacks may be overloaded), wear out(usefulness of s/w system may become absolute due to introduction of new versions with additional features).
    - Storage Medium Failure - Storage Media can be classified as volatile, Non volatile, permanent/stable storage

72. Types of Storage

- Volatile storage -
  - A volatile storage failure can occur due to spontaneous shutdown of the computer system sometimes referred to as a system crash.
  - The cause of the shutdown could be a failure in the power supply unit or a loss of power.
  - A system crash will result in loss of information stored in the volatile storage medium.
  - Another source of data loss from volatile storage can be due to parity errors in more bits than could be corrected by the parity checking unit, such errors will cause partial loss of data..
  - Eg of this type of storage is semiconductor memory requiring uninterruptible power supply for correct operation.
- Non Volatile Storage
  - These types of storage devices do not require power supply for maintaining the stored information.
  - A power failure or system shutdown will not result in loss of information stored on such devices.
  - It is vital that failures that can cause the loss of ordinary data should not also cause the loss of redundant data that is to be used for recovery of ordinary data.
  - One method of avoiding this double loss is store the recovery data on separate storage devices.
  - Examples of this type of storage are magnetic tape and magnetic disk.
- Permanent or Stable Storage
  - It is achieved by redundancy. Thus instead of having a single copy of data on a non volatile storage medium, multiple copies of the data are stored. Each such copy is made on a separate non volatile storage device.
  - Since these independent storage devices have independent failure modes, it is assumed that at least

one of these multiple copies will survive any failure and be usable.
        - The amount and type of data stored in stable storage depends on the recovery scheme used in particular DBMS.
        - Failure of stable storage could be due to natural or manmade disasters.
        - Manually assisted database regeneration is the only possible remedy to permanent stable storage failure.

73. Database Recovery Techniques
    - The types of failures that the computer system is likely to be subjected to include failures of components or subsystems, s/w failures, power outages, accidents, unforeseen situations and the natural or manmade disasters.
    - Database recovery techniques are the methods of making the database fault tolerant.
    - The aim of the recovery scheme is to allow database operations to be resumed after a failure with minimum loss of information at an economically justifiable cost.
    - Different Techniques are:
        - Log Based Recovery
        - Checkpoints
        - Shadow Page Table

74. Log Based Recovery
    - A log, which is usually written to the stable storage, contains the redundant data required to recover from volatile storage failures and also from errors discovered by the transaction or the database system.
    - The Log is written before any updates are made to the database. This is called the write ahead log strategy.
    - In this strategy a transaction is not allowed to modify the physical database until the undo portion of the log (i.e portion of the log that contains the previous values of the modified data) is written to the stable storage
    - Furthermore the log write-ahead strategy requires that a transaction is allowed to commit only after the redo portion of

the log and the commit transaction marker are written to the log.
    - In effect both the undo and redo portions of the log will be written to stable storage before a transaction commit.
    - Using this strategy, the partial updates made by an uncommitted transaction can be undone using the undo portion of the log, and a failure occurring between the writing of the log and the completion of updating the database corresponding to the actions implied by the log can be redone.

75. Data recorded for each transaction in Log Based Recovery?

For each transaction the following data is recorded on the log.

- A start of transaction marker
- The transaction identifier, which could include the who and where information
- The record identifiers, which include the identifiers for the record occurrences.
- The operation(s) performed on the records (insert, delete, modify)
- The previous values of the modified data. This information is required for undoing the changes made by a partially completed transaction, it is called the undo log .Where the modification made by the transaction is the insertion of a new record, the previous values can be assumed to be null.
- The updated values of the modified record(s). This information is required for making sure that the changes made by a committed transaction are in fact reflected in the database and can be used to redo these modifications. This information is called the redo part of the log. In case the modification made by the transaction is the deletion of a record, the updated values can be assumed to be null.
- A commit transaction marker if the transaction is committed, otherwise an abort or rollback transaction marker.

76. Checkpoints

- In an on-line database system, for example in an airline reservation system, there could be hundreds of transactions

handled per minute. The log for this type of database contains a very large volume of information.
    - A scheme called checkpoint is used to limit the volume of log information that has to be handled and processed in the event of a system failure involving loss of volatile information.
    - The checkpoint scheme is an additional component of the logging scheme described above.
    - In the case of a system crash, the log information being collected in buffers will be lost.
    - A checkpoint operation performed periodically, copies log information onto stable storage.
    - For all transactions active at checkpoint, their identifiers and their database modification actions, which at that time are reflected only in the database buffers, will be propagated to the appropriate storage.
    - The frequency of checkpointing is a design consideration of the recovery system.
77. Information stored at each checkpoint

    The information and operations performed at each checkpoint consist of the following.
    - A start of checkpoint record giving the identification that it is a checkpoint along with the time and date of the checkpoint is written to the log on a stable storage device.
    - All log information from the buffers in the volatile storage is copied to the log on stable storage.
    - All database updates from the buffers in the volatile storage are propagated to the physical database.
    - An end of checkpoint record is written and the address of the checkpoint record is saved on a file accessible to the recovery routine on start-up after a system crash.
78. Different Alternatives to Checkpoint problem
79. Shadow Page Table
    - Shadow paging is an alternative to log-based recovery; this scheme is useful if transactions execute serially
    - Idea: maintain two page tables during the lifetime of a transaction – the current page table, and the shadow page table.Store the shadow page table in nonvolatile storage,

such that the state of the database prior to transaction execution may be recovered.
    - Shadow page table is never modified during execution
    - To start with, both the page tables are identical. Only the current page table is used for data item accesses during execution of the transaction.
    - In the shadow page scheme, the database is considered to be made up of logical units of storage called pages.
    - The pages are mapped into physical blocks of storage (of the same size as the logical pages) by means of a page table of the database.
    - This entry contains the block number of the physical storage where this page is stored.
    - The virtual or logical pages are mapped onto physical memory blocks of the same size as the pages, and the mapping is provided by means of a table known as page table.
    - To recover from system crashes during the life of a transaction, all we have to do is revert to the shadow page table so that the database remains accessible after the crash.
80. Advantages & Disadvantages of shadow-paging over log-based schemes
    - Advantages:
        - no overhead of writing log records
        - recovery is trivial
    - Disadvantages :
        - Copying the entire page table is very expensive
            1. Can be reduced by using a page table structured like a B+-tree
            2. No need to copy entire tree, only need to copy paths in the tree that lead to updated leaf nodes
        - Commit overhead is high even with above extension
            1. Need to flush every updated page, and page table
        - Data gets fragmented (related pages get separated on disk)

        - After every transaction completion, the database pages containing old versions of modified data need to be garbage collected
        - Hard to extend algorithm to allow transactions to run concurrently
            1. Easier to extend log based schemes

81. How is a transaction committed in Shadow Page Table
    - Flush all modified pages in main memory to disk
    - Output current page table to disk
    - Make the current page table the new shadow page table, as follows:
        - keep a pointer to the shadow page table at a fixed (known) location on disk.
        - to make the current page table the new shadow page table, simply update the pointer to point to current page table on disk
    - Once a pointer to the shadow page table has been written, a transaction is committed.
    - No recovery is needed after a crash — new transactions can start right away, using the shadow page table.
    - Pages not pointed to from the current/shadow page table should be freed (garbage collected).
82. Concurrency and Problems occurring due to Concurrency and their control mechanisms
    - Concurrency occurs in a multi-user system where at a time more than one user tries to make transactions and it is called concurrency. So when several transactions are executed concurrently, the corresponding schedule is called concurrent schedule.
    - But due to concurrency following problems may occur.
        - Lost Update Problem
        - Inconsistent Read Problem
        - The Phantom Phenomenon
        - Semantics of Concurrent Transaction
    - Concurrency Control Mechanisms include
        - Timestamp Ordering
        - Locking Protocol

- Validation Techniques
- Multiversion Technique(MVT)

83. Timestamp Ordering
    - In the timestamp based method, a serial order is created among the concurrent transaction by assigning to each transaction a unique non decreasing number.
    - The usual value assigned to each transaction is the system clock value at the start of the transaction, hence name Timestamp ordering.
    - A transaction with a smaller timestamp value is considered to be an ‘older’ transaction than another transaction with a larger timestamp value.
    - The contention problem between two transactions in the timestamp ordering system is resolved by rolling back one of the conflicting transactions.
    - A conflict is said to occur when an older transaction tries to read a value that is written by a younger transaction or when an older transaction tries to modify avalue already read or written by a younger transaction.
    - Sometimes rollback could be of type cascading rollback where a single transaction failure leads to a series of transaction rollbacks.
    - Thus the timestamp-ordering protocol guarantees serializability since all the arcs in the precedence graph are of the form:

- There will be no cycles in the precedence graph
- Timestamp protocol ensures freedom from deadlock as no transaction ever waits.
- But the schedule may not be cascade-free, and may not even be recoverable.

84. Solution to the Cascading Rollback Problem

- A transaction is structured such that its writes are all performed at the end of its processing
- All writes of a transaction form an atomic action; no transaction may execute while a transaction is being written
- A transaction that aborts is restarted with a new timestamp

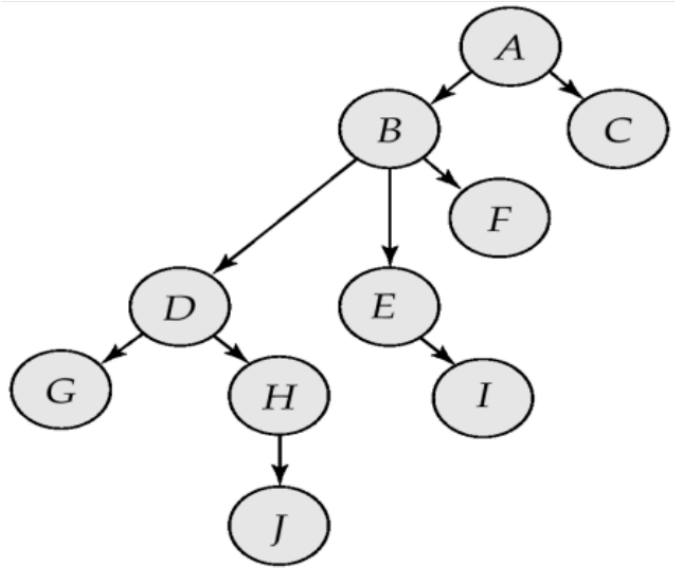
85. Locking Protocol
    - From the point of view of a locking scheme a database can be considered as being made up of a set of data items. A lock is a mechanism to control concurrent access to a data item
    - A lock is a variable associated with each data item which gives status about the availability of the data item.
    - The value of the lock variable is used in the locking scheme to control the concurrent access and manipulation of data associated with the data item.
    - The locking is done by a subsystem of the database management system called Lock Manager
    - Lock requests are made to concurrency-control manager.
    - Transaction can proceed only after request is granted.
    - A locking protocol is a set of rules followed by all transactions while requesting and releasing locks.
    - Locking protocols restrict the set of possible schedules.
    - Three types of locks are there.
        - Exclusive Locks : It is also called update or write lock. The intension of this mode of locking is to provide exclusive use of the data item to one transaction. If a transaction T locks the data item Q in an exclusive mode, no other transaction can access Q, not even to read Q until the lock is released by T.
        - Shared Lock : It is also called as a read lock. Any no of transactions can concurrently lock and access a data item in a shared mode .But none of these transactions can modify the data item .A data item locked in a shared mode can’t be locked in exclusive mode, until the shared lock is released.

        - wait lock: It is used when a data item is locked by some other transaction. Normally a wait queue is maintained to keep track of such transactions
    - To Implement the locking concept, following protocols are used
        - Two Phase Protocol
        - Graph Based or Intension Locking Protocol

86. Two Phase Locking Protocol
    - It uses two phases for locking . i.e release of the locks is delayed until all the locks on all data items required by the transaction have been acquired.
    - Two phases are:
        - Growing phase - In this the number of locks increases from zero to maximum for the transaction. In this phase the transaction may obtain lock but may not release any lock
        - Shrinking phase - In this the number of locks decreases from maximum to zero. A transaction may release locks, but may not obtain any new lock.
    - Thus, this is a protocol which ensures conflict-serializable schedules.
    - In Growing Phase,transaction may obtain locks and transaction may not release locks
    - In Shrinking Phase, Transaction may release locks and transaction may not obtain locks
    - The protocol ensures serializability.
    - It can be proved that the transactions can be serialised in the order of their lock points (i.e. the point where a transaction acquired its final lock).

87. Graph Based or Intension Locking Protocol
    - It provides explicit locking at the lower level of the tree and intension locks are applied in all ancestors.
    - Advantage of this protocol is that the lock manager knows that somewhere the lower level node is locked without having the actual examination of lower level nodes.
    - Thus Graph-based protocols are an alternative to two-phase locking

- The tree-protocol is a simple kind of graph protocol.



  - Only exclusive locks are allowed.
  - The first lock by Ti may be on any data item. Subsequently, a data Q can be locked by Ti only if the parent of Q is currently locked by Ti
  - Data items may be unlocked at any time.
  - The tree protocol ensures conflict serializability as well as freedom from deadlock.
  - Advantages
    - Unlocking may occur earlier in the tree-locking protocol than in the two-phase locking protocol.
    - shorter waiting times, and increase in concurrency
    - protocol is deadlock-free, no rollbacks are required
  - Disadvantages:
    - a transaction may have to lock data items that it does not access.
    - increased locking overhead, and additional waiting time
    - potential decrease in concurrency
    - Schedules not possible under two-phase locking are possible under tree protocol, and vice versa.
    - the abort of a transaction can still lead to cascading rollbacks.

88. Validation Techniques
  - In validation techniques or optimistic scheduling scheme it is assumed that all data items can be successfully updated at

the end of the transaction and to read in the data values without locking.

- Reading is done if the data item is found to be inconsistent (with respect to value read in) at the end of the transaction, then the transaction is rolled back.
- There are 3 phases in optimistic scheduling:
  - Read phase: Various data items are read and stored in temporary local variables. All operations are performed on these variables without actually updating the database.
  - validation phase: All concurrent data items in the database are checked. i.e It checks the serializability. If serializability is violated the transaction is aborted and started again.
  - Write phase: If the transaction passes validation phase the modification made by the transaction are committed i.e Written to the database. Optimistic scheduling does not use any lock hence it is deadlock free. However it faces the problem of data starvation.
- Thus in validation based protocol, Execution of transaction Ti is done in three phases.
  - Read and execution phase: Transaction Ti writes only to temporary local variables
  - Validation phase: Transaction Ti performs a “validation test” to determine if local variables can be written without violating serializability.
  - Write phase: If Ti is validated, the updates are applied to the database; otherwise, Ti is rolled back.
- The three phases of concurrently executing transactions can be interleaved, but each transaction must go through the three phases in that order.
- Also called as optimistic concurrency control since transaction executes fully in the hope that all will go well during validation

89. Multiversion Technique
    - In MVT each write and update of the data item is achieved by making a new copy (version) of the data item.

- MVT is called the Time Domain Addressing Scheme. It follows the accounting principle of “Never overwrite a transaction”. Hence the effect of the latest version is taken into consideration.
- History of evolution of data items is recorded in the database. In case of read operations, the database management system selects one of the versions for processing.
- Multiversion schemes keep old versions of data items to increase concurrency.
- Multiversion Timestamp Ordering
- Multiversion Two-Phase Locking
- Each successful write results in the creation of a new version of the data item written.
- Use timestamps to label versions.
- When a read(Q) operation is issued, select an appropriate version of Q based on the timestamp of the transaction, and return the value of the selected version.
- Reads never have to wait as an appropriate version is returned immediately.
- There are two major problems in the case of MVT.
  - It is a little bit slower in operation as every time a new version is generated.
  - Rollbacks are expensive(cascading)

90. Define Deadlock
- A deadlock is said to occur when there is a circular chain of transactions, each waiting for the release of a data item held by the next transaction in the chain

91. What is wait-for-graph?
- The algorithm to detect a deadlock is based on the detection of such a circular chain in the current system wait-for-graph.
- The wait-for-graph is a directed graph and contains nodes and directed arcs; the nodes of the graph are active transactions.
- An arc of the graph is inserted between two nodes if there is a data item required by the node at the tail of the arc, which is being held by the node at the head of the arc.

- Consider the following two transactions:

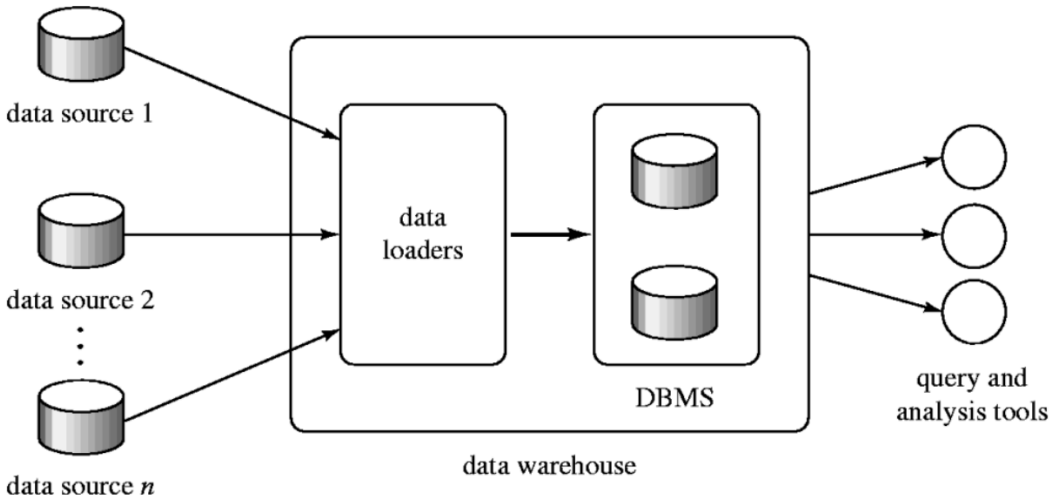
T_1:	write (X)	T_2:	write(Y)
	write(Y)		write(X)

- **Schedule with deadlock**

T_1	T_2
lock-X on X write (X)	
	lock-X on Y write (X) wait for **lock-X** on X
wait for **lock-X** on Y	

  - System is deadlocked if there is a set of transactions such that every transaction in the set is waiting for another transaction in the set.

92. Data Warehousing
  - A data warehouse is a repository (or archive) of information gathered from multiple sources, stored under a unified schema, at a single site.
  - Once gathered, the data are stored for a long time ,permitting access to historical data.
  - Thus data warehouses provide the user a single consolidated interface to data, making decision support queries easier to write.
  - Moreover, by accessing information for decision support from a data warehouse, the decision maker ensures that online transaction processing systems are not affected by the decision support workload.



- Following figure shows the architecture of a typical data warehouse, and illustrates the gathering of data, the storage of data, and the querying and data analysis support.
- The different steps involved in getting data into a data warehouse are called as extract, transform, and load or ETL tasks; extraction refers to getting data from the sources, while load refers to loading the data into the data warehouse.

93. Issues to be addressed in building a warehouse
    - Among the issues to be addressed in building a warehouse are the following.
    - When and how to gather data
        - In a source driven architecture for gathering data, the data sources transmit new information, either continually (as transaction processing takes place), or periodically (nightly, for example).
        - In a destination – driven architecture, the data warehouse periodically sends requests for new data to the sources.
        - Unless updates at the sources are replicated at the warehouse via two- phase commit, the warehouse will never be quite up-to-date with the sources
        - Two phase commit is usually far too expensive to be an option, so data warehouses typically have slightly out-of-date data.
    - What Schema to use
        - Data sources that have been constructed independently are likely to have different schemas.

        - Part of the task of a warehouse is to perform schema integration, and to convert data to the integrated schema before they are stored.
        - As a result, the data stored in the warehouse are not just a copy of the data at the sources.
        - Instead, they can be thought of as a materialised view of the data at the sources.
    - Data transformation and cleansing
        - The task of connecting and preprocessing data is called data cleansing.
        - Data sources often deliver data with numerous minor inconsistencies, which can be corrected.
        - The approximate matching of data required for this task is referred to as fuzzy lookup.
    - How to propagate updates
        - Updates on relations at the data sources must be propagated to the data warehouse.
        - If the relations at the data warehouse are exactly the same as those at the data source, the propagation is straight – forward.
        - If they are not, the problem of propagating updates is basically the view maintenance problem.
    - What data to summarise
        - The raw data generated by a transaction processing system may be too large to store online.
        - However, we can answer many queries by maintaining just summary data obtained by aggregation on a relation, rather than maintain the entire relation.

94. Explain Data Mining

- The term data mining refers loosely to the process of semi automatically analyzing large databases to find useful patterns.
- Data mining differs from machine learning and statistics in that it deals with large volumes of data, stored primarily on disk.
- Data mining deals with ”knowledge discovery in databases”

- Some types of knowledge discovered from a database can be represented by a set of rules.
- There are a variety of possible types of patterns that may be useful, and different techniques are used to find different types of patterns.
- Usually there is a manual component to data mining, consisting of preprocessing data to a form acceptable to the algorithms and post processing of discovered patterns to find novel ones that could be useful.

95. Applications of Data Mining
    - The most widely used applications are those that require some sort of prediction.The prediction is to be based on known attributes
    - Another class of applications looks for associations.
    - Associations are an example of descriptive patterns. Clusters are another example of such patterns.
    - Classification deals with predicting the class of test instances ,by using attributes of the test instances ,based on attributes of training instances, and the actual class of training instances.
    - Association rules identify items that co-occur frequently, for instance items that tend to be bought by the same customer. Correlations look for deviations from expected levels of associations.