②	Ese - custome entity set has attribute customer-id to work as primary key.	Ese - Only payment_no can not work as primary key for cashier entity set.
③	Strong entity is not dependent on any other entity.	Weak entity depends on strong entity.
④	It is represented by single rectangle.	It is represented by double rectangle.
⑤	It has either total participation or not.	It always has total participation.
⑥	Two strong entity's relationship is represented by a single diamond.	The relation bet^n one strong and one weak entity is represented by double diamond.

Qs. 10. What is participation in relationship? Explain diff types of participation with example.

→ In a relationship, participation constraint specifies the existence of an entity when it is related to another entity in relationship type.

Types -

① total participation -

The participation of an entity set E in a relationship R is said to be total

if every entity in E participates in at least one relationship in R.
Ex - We expect every loan entity to be related to at least one customer through the borrower relationship. Hence, participation of loan in relationship set borrower is total.

② partial participation -
If only some entities in E participate in relationships in R, the participation of entity set E in relationship R is said to be partial.
Ex - An individual can be a bank customer whether or not she has a loan with bank. Hence, it is possible that only some of the customer entities are related to loan entity set through borrower relationship. Hence, participation of customer in the borrower relationship is partial.

Q.11. Explain different mapping cardinalities with example.
→ Mapping cardinalities express the number of entities to which another entity can be associated via a relationship set.

① One to one -
An entity in A is associated with at most one entity in B and an entity in B is associated with at most one entity in A.
Ex - one customer - one loan account.

② one to many -
An entity in A is associated with any number (zero or more) of entities in B and an entity in B can be associated with at most one entity in A.
Ex - one customer - many loan accounts

③ many to one -
An entity in A is associated with at most one entity in B and an entity in B can be associated with any number (zero or more) of entities in A.
Ex - 3 customers - join loan account.

④ many to many -
An entity in A is associated with any number (zero or more) of entities in B and an entity in B is associated with any number (zero or more) of entities in A.
Ex - many customers - many loan accounts

Q.12. Describe superkey, candidate key, primary key & foreign key with example.

→ i] superkey -
A superkey is set of one or more attributes that taken collectively allow us to identify uniquely an entity in entity set.
Ex - The customer_id attribute of entity set customer is sufficient to distinguish one customer entity from another.

Thus, customer-id is superkey. Similarly, the combination of customer-name and customer-id is superkey for entity set customer.

2] candidate key -
Candidate key is a superkey for which no proper subset is superkey.
Ex - Suppose that a combination of customer-name and customer-street is sufficient to distinguish among members of the customer entity set.
Then, both {customer-id} and {customer-name, customer-street} are candidate keys.

3] Primary key -
Primary key denote a candidate key that is chosen by database designer as the principal means of identifying entities within an entity set.
Ex - students are ~~routinely~~ assigned unique identification (ID) numbers and all adults receive govt-assigned social security numbers.

4] Foreign key -
A foreign key is a set of attributes in a table that refers to the primary key of another table.
Ex - consider two tables Student and Department. Attributes of table Student are-

stud-id, name & course. And attribute of table Department are - dept-name & stud-id.
In the student table, the field stud-id is a primary key. In the Department table, the field stud-id is foreign key because it is acting as primary key for student table.

Q. 13. Explain - generalization, specialization, aggregation, attribute inheritance with ex.

→ 1] Generalization -
The design process which proceeds in bottom-up manner, in which multiple entity sets are synthesized into a higher-level entity set on the basis of common features is called generalization.
Generalization is a simple inversion of specialization.
They are described by the same attributes and participate in the same relationship sets.

Ex- Customer entity set with attributes name, street, city & customer-id.
Employee entity set with attributes name, street, city, emp-id & salary.
There are similarities bet^n the customer entity set & the employee entity set. They have several attributes

in common. This commonality can be expressed by generalisation.

2] specialisation -
An entity set may include subgroupings of entities that are distinct in some way from other entities in the set. The process of designating subgroupings within an entity set is called specialisation.
In E-R diagram, specialisation is depicted by a triangle component labelled ISA.
The ISA relationship may also be referred to as a superclass - subclass relationship.

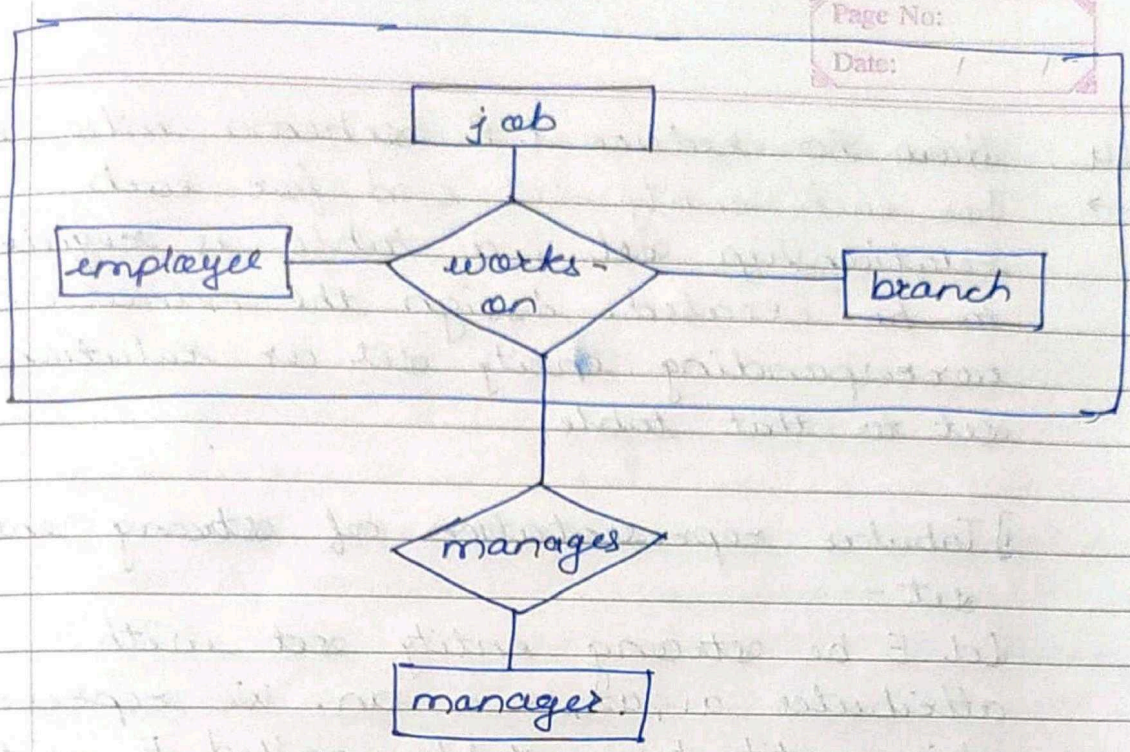
Ex - Consider an entity set person, with attributes name, street & city.
A person may be further classified as one of following:-
customer, employee.
Customer entities may be described further by the attribute customer-id.
Whereas, employee entities may be described further by the attributes employee-id and salary.

3] Aggregation -
Aggregation is an abstraction through which relationships are treated as higher-level entities. E-R model cannot express relationships among relationships.
Aggregation helps to improve this

Page No:
Date:

limitation of E-R model.

Ex - Consider ternary relationship works-on, set^ employee, branch & job. Suppose we want to record managers for tasks performed by an employee at a branch. We want to record managers for (employee, branch, job) combinations.



4] Attribute inheritance -
A crucial property of the higher and lower-level entities created by specialization & generalization is called attribute inheritance.
The attributes of higher-level entity sets are said to be inherited by lower-level entity sets.

Ex- Customer is described by its name, street, city & customer-id attribute.
Employee is described by its name, street, city & employee-id attribute.
Customer & employee inherit the attributes of person. Here customer & employee are lower-level entity sets and person is higher-level entity sets.

Q. 14. How to reduce E-R schema into tables.

→ For each entity set and for each relationship set, a table is required to be created. Assign the name of corresponding entity set or relationship set to that table.

1] Tabular representation of strong entity set -

Let E be strong entity set with attributes $a_1, a_2, \ldots a_n$. We represent this entity by a table called E with n distinct columns, each of which corresponds to one of the attributes of E.

Ex -

loan no	amount
L-11	900
L-12	1000
L-13	1100
L-14	1200

loan table.

2] Tabular representation of weak entity set -

Let A be weak entity set with attributes $a_1, a_2, \ldots, a_m$. Let B be strong entity set on which A depends. Let primary key of B consists of attributes $b_1, b_2, \ldots, b_n$. We represents this entity set by $\{a_1, a_2, \ldots a_m\} \cup \{b_1, b_2, \ldots, b_n\}$.

Ex - loan
payment

loan-no.	payment-no	payment-amt
L-11	53	100
L-12	60	200
L-13	91	300
L-14	92	400
L-15	93	500

3] Tabular representation of relationship set -
Let R be relationship set, let $a_1, a_2, \dots a_m$ be the set of attributes. Let the descriptive attributes of R be $b_1, b_2, \dots b_n$.
We represent this relationship set by $\{a_1, a_2, \dots a_m\} \cup \{b_1, b_2, \dots b_n\}$.

Ex - relationship - borrower.
customer - primary key customer_id
loan - primary key loan-no.

customer_id	loan-no.
11-22-33	L-11
12-22-32	L-12
13-23-33	L-13
14-24-34	L-14

borrower table.

4] Redundancy of table -
Redundancy must be avoided in multiple tables. Only key attributes are stored redundantly in multiple tables. Non-key attributes should be stored only at one place. Redundancy occurs mostly in case of weak entity set.

5] Composite attributes -
It is better to split the composite attribute into simple attributes & represent as a column of table.
Eg - Name composed of first name, middle name, last name.
If name is stored as a composite attribute, it is difficult to retrieve first name only.

6] multivalued attributes -
Diff ways to represent multivalued attributes.
Better to apply normalization & create one more table.

7] Tabular representation of aggregation-
Separate table required to be created for each of the entity set taking part in aggregation.

Q.15. Describe database system structure.
→ - A database system is partitioned into modules that deal with each of the responsibilities of the overall system.
- The functional components of database system are -
① The storage manager
② query processor.

- ① Storage manager
• It is a program module that

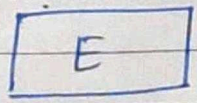
Page No:
Date: / /

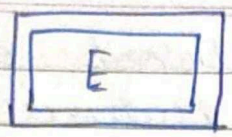
⑤ object - relational data model -
- It combines features of object - oriented data model and relational data model.
- It allows designers to incorporate objects into familiar table structure.

⑥ semi - structured data model -
- In semi - structured model no separation between the data and the schema and the structure used depend on the purpose.
- It provides a flexible format for data exchange between different types of databases.
- The extensible markup language (XML) is widely used to represent semi - structured data.

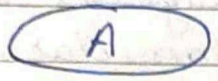
⑦ The Entity - Relationship model -
- Entities - collection of basic objects.
- Entities are described in database by a set of attributes.
- An entity set is set of entities of the same type that share same properties or attributes.
- Relationship - association among several entities.

Q.7. What are different notations used in E-R diagram?

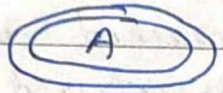
→ Rectangle -  - represent entity set

double rectangle - 

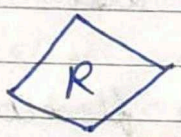
- represent weak entity set

ellipse - 

- represent attribute

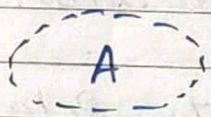
double ellipse - 

- represent multivalued attribute

diamond - 

- represent relationship among entity set.

lines - link attribute to entity set and entity set to relationship

dashed ellipse - 

- derived attribute

double lines - indicate total participation of entity in relationship set.

Q. 8. What are attributes ? Explain diff types of attributes.

→ Attributes are descriptive properties possessed by each member of an entity set. For each attribute, there is a set of permitted values called the domain.

simple attributes -
They are not divided in subparts.

composite attributes -
They can be divided into subparts.
Ex - Name can be divided into first name, middle name and last name.

single - valued attribute -
Attribute which always has only one value.

multi - valued attribute -
Attribute can have more than one value.
Ex - An employee may have zero, one or several phone no.

derived attribute -
value of the attribute can be derived from the values of other related attributes.
Ex - age of person can be derived from date of birth.

Q.9. compare strong entity set & weak entity set.

→

Strong entity set	Weak entity set.
① An entity set having sufficient attributes to form a primary key is called strong entity set.	An entity set which does not have sufficient attributes to form a primary key is called weak entity set.

U1 Introduction to DBMS

Friday, October 29, 2021 11:56

1. Describe the terms – Data, Information, Database, Database Management System.

a. Data- Raw, unorganized facts that need to be processed which can be something simple and seemingly random and useless until it is organized.
b. Information - When data is processed, organized, structured or presented in a given context, to make it useful, it is called information.
c. Database - Collection of related data
d. Database Management System (DBMS) -
    i. Software that manages and controls access to the database.
    ii. Collection of interrelated data (database) and set of programs to access those data.

2. Compare Database System with File System. What is the purpose of Database System?

a. 3

	Basis	File System	DBMS
1	Structure	File system is a software that manages and organizes the files in a storage medium within a computer.	DBMS is a software for managing the database.
2	Data Redundancy	Redundant data can be present in a file system.	In DBMS there is no redundant data.
3	Backup and Recovery	It doesn't provide backup and recovery of data if it is lost.	It provides backup and recovery of data even if it is lost.
4	Query processing	There is no efficient query processing in file system.	Efficient query processing is there in DBMS.
5	Consistency	There is less data consistency in file system.	There is more data consistency because of the process of normalization.
6	Complexity	It is less complex as compared to DBMS.	It has more complexity in handling as compared to file system.
7	Security Constraints	File systems provide less security in comparison to DBMS.	DBMS has more security mechanisms as compared to file system.
8	Cost	It is less expensive than DBMS.	It has a comparatively higher cost than a file system.
9	Data Independence	There is no data independence.	In DBMS data independence exists.
10	Data Access	Is done by using programming languages on text files	Is done by SQL on database application

b. dv

Point of Comparison	File System	DataBase System
	In Early days, database applications were built on top of file systems	
	It suffered from lot of limitations	
Drawbacks	Data isolation • Multiple files and formats • Inconvenience in data sharing Data redundancy and inconsistency • Multiple file formats • duplication of information in different files Difficulty in accessing data – • Need to write a new program to carry out each new task Concurrent access by multiple users • Concurrent access needed for performance • Uncontrolled concurrent accesses can lead to inconsistencies • E.g. two people reading a balance and updating it at the same time Integrity problems • Integrity constraints (e.g. account balance < 0) become part of program code • Hard to add new constraints or change existing ones Atomicity of updates • Failures may leave database in an inconsistent state with partial updates carried out • E.g. transfer of funds from one account to another should either complete or not happen at all Security problems • Due to concurrent access and no access control mechanism	Control of data redundancy • Traditional file-based systems waste space by storing the same information in more than one file. Data consistency • By eliminating redundancy, reduce the risk of inconsistencies occurring. • If a data item is stored only once in the database, any update to it has to be performed only once and the new value is available immediately to all. More information from the same amount of data • With the integration of the operational data, it may be possible for the organization to derive additional information from the same data. Sharing of data • Database belongs to the entire organization and can be shared by all authorized users. Improved data integrity • Database integrity refers to the validity and consistency of stored data. • Integrity is expressed in terms of constraints • Constraints are consistency rules that the database is not permitted to violate. • Constraints may apply to data items within a single record or they may apply to relationships between records. Improved security • Database security is the protection of the database from unauthorized users. Enforcement of standards • Integration allows the DBA to define and enforce the necessary standards. • These may include departmental, organizational, national, or international standards.

Economy of scale
- Combining all the organization's operational data into one database and creating a set of applications that work on this one source of data, can result in cost savings.

Improved data accessibility and responsiveness
- As a result of integration, data that crosses departmental boundaries is directly accessible to the end-users.
- This provides a system with potentially much more functionality

Increased productivity
- DBMS provides many of the standard functions that the programmer would normally have to write in a file based application.
- At a basic level, the DBMS provides all the low-level file-handling routines that are typical in application programs.

Increased concurrency
- Many users can access database simultaneously
- Many DBMS manage concurrent database access and ensure problems cannot occur.

Improved backup and recovery services
- Modern DBMSs provide facilities to minimize the amount of processing that is lost due to failure.
- Take backup regularly.
- Assure minimum down-time

The limitations of file-based approach can be attributed to two factors:

1. The definition of the data is embedded in the application programs, rather than being stored separately and independently
2. There is no control over the access and manipulation of data beyond that imposed by the application programs.

Disadvantages of DBMS

- Complexity
  - The provision of the functionality expected from a good DBMS makes the DBMS an extremely complex piece of software.
  - Database designers, developers, database administrators, and end-users must understand this functionality to take full advantage of it.
  - Failure to understand the system can lead to bad design decisions, which can have serious consequences for an organization.
- Memory Size
  - The complexity and breadth of functionality makes the DBMS an

extremely large piece of software,
- occupying many megabytes of disk space and requiring substantial amounts of memory to run efficiently

- Cost of DBMS
  - The cost of DBMS varies significantly, depending on the environment and functionality provided.
  - A single-user DBMS for a personal computer may only cost US$100
- Additional hardware costs
  - The disk storage requirements for the DBMS and the database may necessitate the purchase of additional storage space.
  - To achieve the required performance, it may be necessary to purchase a higher configuration machine
- Cost of conversion
  - In some situations, the cost of the DBMS and extra hardware may be insignificant compared with the cost of converting existing applications to run on the new DBMS and hardware. –
  - This cost also includes the cost of training staff to use these new systems.
- Performance
  - A file-based system is written for a specific application, such as invoicing.
  - As a result, performance is generally very good.
  - However, the DBMS is written to be more general, to cater for many applications rather than just one.
  - The effect is that some applications may not run as fast as they used to.
- Higher impact of a failure
  - The centralization of resources increases the vulnerability of the system.
  - Since all users and applications rely on the availability of the DBMS, the failure of certain components can bring operations to a halt.

- The purpose of the db systems

3. Give advantages and disadvantages of DBMS.

a. Advantages of DBMS

1. Control of data redundancy
   - Traditional file-based systems waste space by storing the same information in more than one file.
2. Data consistency
   - By eliminating redundancy, reduce the risk of inconsistencies occurring.
   - If a data item is stored only once in the database, any update to it has to be performed only once and the new value is available immediately to all.
3. More information from the same amount of data
   - With the integration of the operational data, it may be possible for the organization to derive additional information from the same data.
4. Sharing of data
   - Database belongs to the entire organization and can be shared by all authorized users.
5. Improved data integrity
   - Database integrity refers to the validity and consistency of stored data.
   - Integrity is expressed in terms of constraints
   - Constraints are consistency rules that the database is not permitted to violate.
   - Constraints may apply to data items within a single record or they may apply to relationships between records.
6. Improved security
   - Database security is the protection of the database from unauthorized users.
7. Enforcement of standards
   - Integration allows the DBA to define and enforce the necessary standards.
   - These may include departmental, organizational, national, or international standards.
8. Economy of scale
   - Combining all the organization's operational data into one database and creating a set of applications that work on this one source of data, can result in cost savings.
9. Improved data accessibility and responsiveness
   - As a result of integration, data that crosses departmental boundaries is directly accessible to the end-users.
   - This provides a system with potentially much more functionality
10. Increased productivity
    - DBMS provides many of the standard functions that the programmer would normally have to write in a file based application.
    - At a basic level, the DBMS provides all the low-level file-handling routines that are typical in application programs.
11. Increased concurrency
    - Many users can access database simultaneously
    - Many DBMS manage concurrent database access and ensure problems cannot occur.
12. Improved backup and recovery services
    - Modern DBMSs provide facilities to minimize the amount of processing that is lost due to failure.
    - Take backup regularly.
    - Assure minimum down-time

b. Disadvantages of DBMS

1. Complexity
    - The provision of the functionality expected from a good DBMS makes the DBMS an extremely complex piece of software.
    - Database designers, developers, database administrators, and end-users must understand this functionality to take full advantage of it.
    - Failure to understand the system can lead to bad design decisions, which can have serious consequences for an organization.
2. Memory Size
    - The complexity and breadth of functionality makes the DBMS an extremely large piece of software,
    - occupying many megabytes of disk space and requiring substantial amounts of memory to run efficiently
3. Cost of DBMS
    - The cost of DBMS varies significantly, depending on the environment and functionality provided.
    - A single-user DBMS for a personal computer may only cost US$100
4. Additional hardware costs
    - The disk storage requirements for the DBMS and the database may necessitate the purchase of additional storage space.
    - To achieve the required performance, it may be necessary to purchase a higher configuration machine
5. Cost of conversion
    - In some situations, the cost of the DBMS and extra hardware may be insignificant compared with the cost of converting existing applications to run on the new DBMS and hardware. –
    - This cost also includes the cost of training staff to use these new systems.
6. Performance
    - A file-based system is written for a specific application, such as invoicing.
    - As a result, performance is generally very good.
    - However, the DBMS is written to be more general, to cater for many applications rather than just one.
    - The effect is that some applications may not run as fast as they used to.
7. Higher impact of a failure
    - The centralization of resources increases the vulnerability of the system.
    - Since all users and applications rely on the availability of the DBMS, the failure of certain components can bring operations to a halt.

4. Explain the view of data. What are the different levels of data abstraction?

a. View of data

1. Complex data structures used for efficiency
2. Hides certain details of how data are stored and maintained
3. Provides abstract view
4. Hides complexity

b. levels of data abstraction

1. Physical Level

1) Lowest level of abstraction
2) Describes how data are actually stored
3) Describes complex low level data structures in detail
4) Blocks of consecutive storage locations- words, byte etc

5) Administrator may be aware of certain details of the physical organization

2. Logical Level

1) Next higher level of abstraction
2) Describes what data are stored in the database
3) And what relationship exists among those data
4) Describes entire database in terms of a small number of relatively simple structures
5) Do not bother with complex physical storage
6) Database administrator uses logical level
7) Who decides what information to keep in database
8) Each record is described by a type definition
9) Programmers also work at this level

3. View Level

1) Highest level of abstraction
2) Describes only part of entire database
3) Exists to simplify interaction with the system
4) System may provide many views for the same database
5) Set of application programs that hide details of the data types
6) Views also provides security mechanisms to prevent users from accessing certain parts of database

5. Describe the terms – instance and schema.

a. Database changes over time as information is inserted and deleted
b. These are similar to data types and variables in programming languages

1. Schema –

1) The logical structure of the database
2) The overall design of the database
3) Schemas are changed infrequently
4) A schema contains schema objects like table, foreign key, primary key, views, columns, data types, stored procedure, etc.
5) Analogous to type information of a variable in a program
6) corresponds to the variable declaration
7) A database schema can be represented by using the visual diagram. That diagram shows the database objects and relationship with each other.
8) – e.g. the database consists of information about a set of customers and accounts and the relationship between them
9) Database systems have several schemas
10) Partitioned according to levels of abstraction
    a) Physical Schema – describes database design at physical level
    b) Logical Schema – describes database design at logical level
    c) Several schemas at view level, called as sub-schemas

2. Instance

1) Collection of information stored in the database at a particular moment
2) The value of the variable in a program at a point in time correspond to an instance of a database schema

6. Describe different data models – relational, object-oriented, object-relational, semi-structured, network, hierarchical What is data model? Explain different data models.

a. Underlying the structure of a database is data model
b. A collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints

1. Relational Model

1) uses a collection of tables to represent both data and the relationships among those data.
2) Each table has multiple columns, and each column has a unique name.
3) relational model is an example of a record-based model.
4) Record-based models are so named because the database is structured in fixed-format records of several types.
5) Each table contains records of a particular type.

2. Hierarchical Data Model

1) The hierarchical model organizes data into a tree- like structure, where each record has a single parent or root.
2) Sibling records are sorted in a particular order.
3) That order is used as the physical order for storing the database.
4) This model is good for describing many real-world relationships.
5) Hierarchical model was primarily used by IBM's Information Management Systems(IMS) in the 60s and 70s,
6) but they are rarely seen today due to certain operational inefficiencies.

3. Network Data Model

1) The network model builds on the hierarchical model by allowing many-to-many relationships between linked records, implying multiple parent records.
2) Based on mathematical set theory, the model is constructed with sets of related records.
3) Each set consists of one owner or parent record and one or more member or child records.
4) A record can be a member or child in multiple sets, allowing this model to convey complex relationships.
5) It was most popular in the 70s after it was formally defined by the Conference on Data Systems Languages (CODASYL).

4. Object-oriented Data Model

1) This model defines a database as a collection of objects.
2) Uses Object-oriented programming language concepts like
    a) Class, Object, Encapsulation, Inheritance

5. Object-relational Data Model

1) It combines features of the object-oriented data model and relational data model
2) It allows designers to incorporate objects into the familiar table structure.

6. Semi-structured Data Model

1) In semi-structured model no separation between the data and the schema, and the structure used depends on the purpose.
2) It permit the specification of data where individual data items of the same type may have different sets of attributes.
3) This is in contrast with the data models mentioned earlier where every data item of a particular type must have the same set of attributes
4) It provides a flexible format for data exchange between different types of databases.
5) The schema can easily be changed.
6) The data transfer format may be portable.
7) The primary trade-off being made in using a semi-structured database model is that queries cannot be made as efficiently as in a more constrained structure, such as in the relational model. •
8) The extensible markup language (XML) is widely used to represent semi-structured data.

7. The Entity-Relationship Model (E-R)

1) Collection of basic objects – entities
2) And relationships among these objects
3) Entity – things or objects in the real world
4) Entities are described in database by a set of attributes
5) Some attributes used to uniquely identify the entities
6) Relationship – association among several entities
7) Attribute - An entity is represented by a set of attributes

7. What are the different notations used in ER diagram?

8. What are the different types of attributes? Give the notations used in E-R model for different types of attributes. Describe simple, composite, Single valued, Multivalued, Derived Attributes.

9. Compare – strong entity set and weak entity set

10. d

		Strong Entity	Weak Entity
	Key	Strong entity always has a primary key.	Weak entity have a foreign key referencing primary key of strong entity.
	Dependency	Strong entity is independent of other entities.	Weak entity depends on strong entity.
	Represented by	Strong entity is represented by a single rectangle.	Weak entity is represented by a double rectangle.
	Relationship Representation	Two strong entity's relationship is represented by a single diamond.	While the relation between one strong and one weak entity is represented by a double diamond.
	Participation	Strong entities have either total participation or not. i.e. may or may not participate in entity relationships.	While weak entity always has total participation. i.e. always participates in entity relationships.

11. D

	BASIS OF COMPARISON	Strong Entity	Weak Entity
	description	A strong entity is an entity that is independent of any other entity in a schema. It has sufficient attributes to form a primary key.	A weak entity is an entity set that cannot be uniquely identified by its attributes alone. It does not have sufficient attributes to form a primary key.
	Name for member	The member of a strong entity set is referred to as a dominant entity set.	The member of a weak entity set is referred to as a subordinate entity set.
	Symbol	A strong entity is denoted with a single rectangle.	A weak entity is denoted with a double rectangle.
	Dependency	Strong entity is not dependent on any other	Weak entity is dependent on strong entity for its existence.

		entity in a schema.	
	Primary Key Structure	The primary key is one of its attributes which uniquely identifies its member.	The primary key of the weak entity set is a combination of partial key and primary key of the strong entity.
	Primary Key	A strong entity always has a primary key represented by an underline.	A weak entity has a partial key or discriminator, which is a list of attributes that identify weak entities related to the same owner entity. The partial key is represented by a dashed underline.
	ER Diagram	In the ER diagram, the relationship between two strong entity set is represented by a diamond symbol.	In the ER diagram, both the weak entity and its corresponding relationship are represented using a double line and the partial key is underlined with a dotted line.
	Connection Line	The connecting line of the strong entity set with the relationship is single.	The line connecting the weak entity set for identifying relationship is double.
	Total Participation	The strong entity may or may not show the total participation in its relations.	Weak entity has a total participation constraint with respect to its identifying relationship because it cannot be identified independently of its own identity.

	BASIS OF COMPARISON	Strong Entity set	Weak Entity set
		Has its own primary key	It does not have sufficient attributes to form a primary key on its own
		It is represented by a rectangle	It is represented by a double rectangle
		It contains a primary key represented by an underline	It contains a partial key or discriminator represented by a dashed underline
		The primary key is one of its attributes which uniquely identifies its member	The primary key of weak entity set is a combination of partial key and primary key of the strong entity set.
		The relationship between two strong entity set is represented by a diamond symbol	The relationship between one strong and a weak entity set is represented by a double diamond sign. It is known as identifying relationship
		The line connecting strong entity set with the relationship is single	The line connecting weak entity set with the identifying relationship is double
		Total participation in the relationship may or may not exist	Total participation in the identifying relationship always exists

		Strong Entity Sets	Weak Entity Sets
		An entity set having	An entity set may not have sufficient

		sufficient attributes to form a primary key.	attributes to form a primary key
			For a weak entity set to be meaningful, it must be associated with another entity set, called the identifying or owner entity set.
			The weak entity set is said to be existence dependent on the identifying entity set
			The identifying entity set is said to own the weak entity set that it identifies.
			The relationship associating the weak entity set with the identifying entity set is identifying relationship.
			The identifying relationship is many to one

12. What is participation in relationship? Explain different types of participation with example.
    a. Participation:
        1. The association between entity sets is referred to as participation
    b. Total participation
    c. Partial participation
13. Explain different mapping cardinalities/cardinality ratios with example
    a. express the number of entities to which another entity can be associated via a relationship set.
14. Describe the terms – super key, candidate key, primary key and foreign key with example.
15. Explain – generalization, specialization, aggregation with example.
16. How to reduce E-R schema into tables?
    a. For each entity set and for each relationship set, a table is required to be created.
    b. Assign the name of the corresponding entity set or relationship set to that table.
    c. Each table has multiple columns, each of which has unique name.
17. Describe Database System Structure.
    a. A database system is partitioned into modules
    b. that deal with each of the responsibilities of the overall system.
    c. The functional components of a database system can be broadly divided into
        1. The storage manager
            1) It 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.
            2) The storage manager is responsible for the interaction with the file manager.
            3) The raw data are stored on the disk using the file system, which is usually provided by a conventional operating system.
            4) The storage manager translates the various DML statements into low-level file-system commands.
            5) The storage manager is responsible for storing, retrieving, and updating data in the database.

6) The storage manager implements several data structures as part of the physical system implementation:
    a) Data files - which store the database itself.
    b) Data dictionary - which store metadata about the structure of the database, in particular the schema of the database.
    c) Indices - which provide fast access to data items that hold particular values.
    d) Statistical Data – Statistical observations related to size of database, number of users, peak time etc.
7) The storage manager components include:
    a) Authorization and integrity manager
        i) Tests for the satisfaction of integrity constraints
        ii) Helps to maintain the integrity of DBMS
        iii) Checks the authority of users to access data.
        iv) Helps to prevent unauthorized data access.
    b) Transaction manager
        i) Ensures that the database always remains in a consistent (correct) state despite system failures
        ii) Ensures concurrent transaction executions proceed without conflicting.
        iii) Helps to maintain ACID properties of Transaction
        iv) A – Atomicity
        v) C – Consistency
        vi) I – Isolation
        vii) D – Durability

    c) File manager
        i) Data logically represented in form of table get converted into records.
        ii) Records are stored in form of files.
        iii) Blocks of information in files stored in sectors and tracks of disk storage
        iv) It manages the allocation of space on disk storage and the data structures used to represent information stored on disk.
    d) Buffer manager
        i) It is responsible for fetching data from disk storage into main memory
        ii) Decide what data to cache in main memory.
        iii) 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 main memory

2. The query processor

   The query processor components include

   1) DDL Interpreter
       a) Receive the Commands in form of DDL statements like Create, Alter, Rename, Drop, Truncate
       b) Interprets DDL statements and records the definitions in the data dictionary.
   2) DML Compiler and Organizer
       a) It receives command in form of DML Statements.
       b) It translates the DML statements in a query language into an evaluation plan consisting of low level instructions that the query evaluation engine understands.

3) Query Evaluation Engine
    a) It executes low-level instructions generated by the DML Compiler and received from Application Program Object Code
4) Compiler and Linker
    a) It is used to separate application program part and DML statements from application program received as input.
    b) It is used to separate application program part and DML statements from application program received as input.

18. What are the different users of database?

a. People who work with a database can be categorized as –
    1. Database Users
    2. Database Administrators
b. There are four different types of database-system users
c. differentiated by the way they expect to interact with the system.
d. Different types of user interfaces have been designed for the different types of users.
    1. Naive Users
        1) Unsophisticated users who interact with the system by invoking one of the application programs that have been written previously.
        2) For example, a bank teller who needs to transfer $50 from account A to account B invokes a program called transfer.
    2. Application Programmers
        1) They are computer professionals who write application programs.
        2) Application programmers can choose from many tools to develop user interfaces.
        3) They can write application program using any application development tool
    3. Sophisticated Users
        1) Interact with the system without writing programs.
        2) Instead, they form their requests in a database query language.
        3) They submit each such query to a query processor, whose function is to break down DML statements into instructions that the storage manager understands.
        4) Analysts who submit queries to explore data in the database fall in this category.
    4. Specialized Users
        1) They are sophisticated users who write specialized database applications that do not fit into the traditional data-processing framework.
        2) Among these applications are computer-aided design systems, knowledge base and expert systems, systems that store data with complex data types (for example, graphics data and audio data), and environment-modeling systems.
e. Database Administrator
    1. One of the main reasons for using DBMS is to have central control of both the data and the programs that access those data.
    2. A person who has such central control over the system is called a database administrator (DBA).

19. Draw E-R diagram for college database, with following assumptions. A college contains many departments. Each department can offer any number of courses. Many instructors can work in a department. An instructor can work only in one department. For each department there is a Head. An instructor can be head of only one department. Each instructor can take any

number of courses. A course can be taken by only one instructor. A student can enroll for any number of courses. Each course can have any number of students

20. Draw E-R diagram for banking application, with following assumptions. There are multiple banks and each bank has many branches. Each branch has multiple customers. Customers have various types of accounts. Some Customers also had taken different types of loans from these bank branches. One customer can have multiple accounts and Loans
21. Draw E-R diagram for National Hockey League (NHL), with following assumptions. The NHL has many teams, each team has a name, a city, a coach, a captain, and a set of players, each player belongs to only one team, each player has a name, a position (such as left wing or goalie), a skill level, and a set of injury records, a team captain is also a player, a game is played between two teams (referred to as host_team and guest_team) and has a date and a score.