Q.2a. What is a database? Describe the advantages and disadvantages of using DBMS over file system.

Answer: Database – A database is a collection of related data and/or information stored so that it is available to many users for different purposes.
Advantages Of DBMS
1. Centralized Management and Control - One of the main advantages of using a database system is that the organization can exert, via the DBA, centralized management and control over the data.
2. Reduction of Redundancies and Inconsistencies - Centralized control avoids unnecessary duplication of data and effectively reduces the total amount of data storage required. Removing redundancy eliminates inconsistencies.
3. Data Sharing - A database allows the sharing of data under its control by any number of application programs or users.
4. Data Integrity - Data integrity means that the data contained in the database is both accurate and consistent. Centralized control can also ensure that adequate checks are incorporated in the DBMS to provide data integrity.
5. Data Security - Data is of vital importance to an organization and may be confidential. Such confidential data must not be accessed by unauthorized persons. The DBA who has the ultimate responsibility for the data in the DBMS can ensure that proper access procedures are followed. Different levels of security could be implemented for various types of data and operations.
6. Data Independence - Data independence is the capacity to change the schema at one level of a database system without having to change the schema at the next level. It is usually considered from two points of view: physical data independence and logical data independence. Physical data independence is the capacity to change the internal schema without having to change conceptual schema. Logical data independence is the capacity to change the conceptual schema without having to change external schemas or application programs.
7. Providing Storage Structures for Efficient Query Processing - Database systems provide capabilities for efficiently executing queries and updates. Auxiliary files called indexes are used for this purpose.
8. Backup and Recovery - These facilities are provided to recover databases from hardware and/or software failures.
Some other advantages are:
Reduced Application Development Time
Flexibility
Availability of up-to-date Information
Disadvantages Of DBMS
1. Cost of Software/Hardware and Migration - A significant disadvantage of the DBMS system is cost.

2. Reduced Response and Throughput - 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.
3. Problem with Centralization - Centralization also means that the data is accessible from a

single source namely the database. This increases the potential of security breaches and disruption of the operation of the organization because of downtimes and failures.

b. Using a suitable example show how an E-R model construct can be mapped to a relational model.

Answer: An entity-relationship model (ERM): An entity-relationship model (ERM) is an abstract conceptual representation of structured data. Entity-relationship modeling is a relational schema database modeling method, used in software engineering to produce a type of conceptual data model (or semantic data model) of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created using this process are called entity-relationship diagrams, or ER diagrams or ERDs for short.
ER-to-Relational Mapping Algorithm:
1) Step 1: Mapping of regular entity types: For each strong entity type E, create a relation T that includes all the simple attributes of a composite attribute.
2) Step2: Mapping of weak entity types: For each weak entity type W with owner entity type E, create relation R and include all simple attributes (or simple components of composite attributes) of W as attributed of R. In addition, include as foreign key attributes of R, the primary key attribute (s) of relation(s) that correspond to the owner(s) and the partial key of the weak entity type W, if any.
3) Mapping of relationship types: form a relation R, for relationship with primary keys of participating relations A and B as foreign keys in R. In addition to this, any attributes of relationship become an attribute of R also.
4) Mapping of multivalued attributes: For each multilvalued attribute A, create a new relation R. This relation R will include an attribute corresponding to A, plus primary key attribute K-as a foreign key in R-of the relation that represents the entity type or relationship type that has A as an attribute.

Q.3a. Illustrate and explain the main phases of database design.

Answer: Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a Data Definition Language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system

Conceptual Design

Once all the requirements have been collected and analyzed, the next step is to create a conceptual shema for the database, using a high level conceptual data model. This phase is called conceptual design.

The result of this phase is an Entity-Relationship (ER) diagram or UML class diagram. It is a high-level data model of the specific application area. It describes how different entities (objects, items) are related to each other. It also describes what attributes (features) each entity has. It includes the definitions of all the concepts (entities, attributes) of the application area.

During or after the conceptual shema design, the basic data model operations can be used to specify the high-level user operations identified during the functional analysis. This also serves to confirm that the conceptual schema meets all the indenfied functional requirements.

Logical Design

The result of the logical design phase (or data model mapping phase) is a set of relation shcemas. The ER diagram or class diagram is the basis for these relation schemas.

To create the relation shemas is quite a mechanical operation. There are rules how the ER model or class diagram is transferred to relation shemas.

The relation schemas are the basis for table definitions. In this phase (if not done in previous phase) the primary keys and foreign keys are defined.

Normalization

Normalization is the last part of the logical design. The goal of normalization is to eliminate redundancy and potential update anomalies.

Redundancy means that the same data is saved more than once in a database. Update anomaly is a consequence of redundancy. If a piece of data is saved in more than one place, the same data must be updated in more than one place.

Normalization is a technique by which one can modify the relation schema to reduce the redundancy. Each normalization phase adds more relations (tables) into the database.

Physical Design

The goal of the last phase of database design, physical design, is to implement the database. At this phase one must know which database management system (DBMS) is used. For example, different DBMS's have different names for data types and have different data types.

b.What are Weak entity types? Explain with the help of examples.

Answer: In a relational database, a **Weak Entity** is an entity that cannot be uniquely identified by its attributes alone; therefore, it must use a foreign key in conjunction with its attributes to create a primary key. The foreign key is typically a primary key of an entity it is related to.

In entity relationship diagrams a weak entity set is indicated by a bold rectangle (the entity) connected by a bold type arrow to a bold diamond (the relationship). An identifying relationship is one where the primary key is populated to the child weak entity as a primary key in that entity.

In general (though not necessarily) a weak entity does not have any items in its primary key other than its inherited primary key and a sequence number. There are two types of weak entities: associative entities and subtype entities. The latter represents a crucial type of normalization, where the super-type entity inherits its attributes to subtype entities based on the value of the discriminator.

An example of a weak entity without a sub-type relationship would be the "header/detail' records in many real world situations such as claims, orders and invoices, where the header captures information common across all forms and the detail captures information specific to individual items.

Q.4a. What do you mean by integrity constraints? Explain the two integrity constraints, check and foreign key in SQL, with an example for each. Give the syntax.

Ans:Integrity Constraints –An integrity constraint is a condition specified on a database schema and restricts the data that can be stored in an instance of the database. If a database instance satisfies all the integrity constraints specified on the database schema, it is a legal instance. A DBMS enforces integrity constraints, in that it permits only legal instances to be stored in the database.
CHECK constraint – CHECK constraint specifies an expression that must always be true for every row in the table. It can't refer to values in other rows.
Syntax:
ALTER TABLE <table_name>
ADD CONSTRAINT <constraint_name> CHECK(<expression>);
FOREIGN KEY constraint – A foreign key is a combination of columns with values based on the primary key values from another table. A foreign key constraint, also known as referential integrity constraint, specifies that the values of the foreign key correspond to actual values of the primary or unique key in other table. One can refer to a primary or unique key in the same table also.
Syntax:
ALTER TABLE <table_name>
ADD CONSTRAINT <constraint_name> FOREIGN KEY(<column_name(s)>) REFERENCES <base_table>(<column_name>) ON {DELETE | UPDATE} CASCADE;

b.Explain the operation of a two-tier client/server architecture for DBMS.

Answer: Page-47 of textbook (5th edition

Q.5 a. Describe Entity integrity and Referential integrity. Give an example for both.

Answer: Entity Integrity Rule – If the attribute A of relation R is a prime attribute of R then A cannot accept null values.

Referential Integrity Rule – In referential integrity, it is ensured that a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation. For example:

STUDENT

Enrl No	Roll No	Name	City	Mobile
11	17	Ankit Vats	Delhi	9891663808
15	16	Vivek Rajput	Meerut	9891468487
6	6	Vanita	Punjab	
33	75	Bhavya	Delhi	9810618396

GRADE

Roll No	Course	Grade
6	C	A
17	VB	C
75	VB	A
6	DBMS	B
16	C	B

Roll No is the primary key in the relation STUDENT and Roll No + Course is the primary key of the relation GRADE. (Entity Integrity)
Roll No in the relation GRADE (child table) is a foreign key, which is referenced from the relation STUDENT (parent table). (Referential Integrity).

b.Define Relational Algebra. Discuss traditional set operations on relations.

relational algebra is an offshoot of first-order logic and of algebra of sets concerned with operations over finitary relations, usually made more convenient to work with by identifying the components of a tuple by a name (called attribute) rather than by a numeric column index, which is called a relation in database terminology.

The main application of relational algebra is providing a theoretical foundation for relational databases, particularly query languages for such databases, chief among which is SQL.

he relational algebra uses set union, set difference, and Cartesian product from set theory, but adds additional constraints to these operators.

For set union and set difference, the two relations involved must be union-compatible—that is, the two relations must have the same set of attributes. Because set intersection can be defined in terms of set difference, the two relations involved in set intersection must also be union-compatible.

For the Cartesian product to be defined, the two relations involved must have disjoint headers—that is, they must not have a common attribute name.

In addition, the Cartesian product is defined differently from the one in set theory in the sense that tuples are considered to be "shallow" for the purposes of the operation. That is, the Cartesian product of a set of n-tuples with a set of m-tuples yields a set of "flattened" (n + m)-tuples (whereas basic set theory would have prescribed a set of 2-tuples, each containing an n-tuple and an m-tuple). More formally, R × S is defined as follows:

$$R \times S = \{(r_1, r_2, ..., r_n, s_1, s_2, ..., s_m) \mid (r_1, r_2, ..., r_n) \in R, (s_1, s_2, ..., s_m) \in S\}$$

The cardinality of the Cartesian product is the product of the cardinalities of its factors, i.e., $|R \times S| = |R| \times |S|$.

c.Explain the difference between 1NF and 2NF.

1NF is the First normal form, which provides the minimum set of requirements for normalizing a relational database. A table that complies with 1NF assures that it actually represents a relation (i.e. it does not contain any records that are repeating), but there is no universally accepted definition for 1NF. One important property is that a table that comply with 1NF could not contain any attributes that are relational valued (i.e. all the attributes should have atomic values).

2NF

2NF is the Second normal form used in relational databases. For a table to comply with 2NF, it should be complied with 1NF and any attribute that is not a part of any candidate key (i.e. non-prime attributes) should fully depend on any of the candidate keys in the table.

Q.6a. What do you mean by indexing? What are the different types of indexing?

Answer: Page 514 of text book(5th edition)

b.What is hash file organization? What are the causes of bucket overflow in a hash file organization? What can be done to reduce the occurrence of bucket overflow?

Answer: Hashing involves computing the address of a data item by computing a function on the search key value.
A hash function h is a function from the set of all search key values K to the set of all bucket addresses B.
We choose a number of buckets to correspond to the number of search key values we will have stored in the database.
To perform a lookup on a search key value K_i, we compute $h(K_i)$, and search the bucket with that address.
If two search keys i and j map to the same address, because, $h(K_i)=h(K_j)$ then the bucket at the address obtained will contain records with both search key values.
The hash functions selected should be such that it should result in uniform distribution of search keys.

Q.7 a. Define Armsrong's axioms. Why are these called sound and complete?

b.Differentiate between the following:

(i) Theta Join
(ii) Equi Join
(iii) Natural Join
(iv) Outer Join

Ans:(i) Theta Join – The theta join operation is an extension to the natural-join operation that allows us to combine selection and a Cartesian product into a single operation. Consider relations r(R) and s(S), and let θ be a predicate on attributes in the schema R ∪S. The theta join operation r $_\theta$ s is defined as follows:
r $_\theta$ s = σ_θ (r x s)
(ii) Equi Join – It produces all the combinations of tuples from two relations that satisfy a join condition with only equality comparison (=).
(iii) Natural Join - Same as equi-join except that the join attributes (having same names) are not included in the resulting relation. Only one sets of domain compatible attributes involved in the natural join are present.
(iv) Outer Join - If there are any values in one table that do not have corresponding value(s) in the other, in an equi-join that will not be selected. Such rows can be forcefully selected by using the outer join. The corresponding columns for that row will have NULLs. There are actually three forms of the outer-join operation: left outer join (X), right outer join (X) and full outer join (X).

Q.8 a. What are the differences between Functional, Multivalued and Join dependencies? Give examples.

Answer: Page 355 of text book(5th edition)

b.What are the basic data types available for attributes in SQL?

Answer: Primary Key – Primary key is one of the candidate keys. It should be chosen such that its attribute values are never, or very rarely, changed.
b) Data Manipulation Language (DML) – A data manipulation language is a language that enables users to access or manipulate data as organized by the appropriate data model.
c) Multivalued Attribute – Multivalued attribute may have more than one value for an entity. For example, PreviousDegrees of a STUDENT.
d) Relationship Instance – A relationship is an association among two or more entities. An instance of relationship set is a set of relationships.

Q.9 a. Explain cost based query optimization. What are the cost components for Query Execution?

Answer: Page 580 of text book(5th edition)

b. Explain how Aggregate Operations and Outer Joins are implemented.

Answer: Page 568 of text book(5th edition)

<u>Textbook</u>

1. **Fundamentals of Database Systems, Elmasri, Navathe, Somayajulu, Gupta, Pearson Education, 2006**