Part 1:

a)

EER Diagram:

An enhanced entity-relationship diagram, or EER Diagram (EERD), is a graphic depiction of a data model that applies the ideas of the Enhanced Entity-Relationship Model (EERM) or Extended Entity-Relationship Model. Entity supertypes, entity subtypes, and entity clustering are some of the new semantic components that this paradigm adds to the conventional Entity-Relationship (ER) model. EER diagrams are useful for creating databases for systems that have complicated data needs because they can more fully depict complex data structures and their interactions.

Supertypes and Subtypes of Entities:

Entity supertypes and subtypes are fundamental ideas in the Extended Entity Relationship Model (EERM) that help structure and arrange data entities. Higher-level entities known as entity supertypes generalize shared characteristics among related subtypes. On the other hand, entity subtypes can contain unique properties in addition to inheriting traits from supertypes.

Example:

"Employee" can be the entity supertype in a company's database, signifying common properties such as "Name" and "Hire Date." The entity subtypes "Manager" and "Technician" may inherit these properties as well as possess unique attributes such as "Team" and "Manager ID" for managers and "Skills" and "Technician ID" and "Manager ID" for technicians. Relationships specific to each subtype are facilitated and data duplication is reduced by this structure.

b)

Entity Cluster:

An entity cluster is a fundamental idea in data modelling and computer database systems, mostly used in the construction of entity-relationship diagrams (ERD). It performs the crucial function of streamlining sophisticated ERDs that might include a large number of entities and complex interactions. An entity cluster is essentially an abstract or virtual entity type that is used to conceptually represent numerous connected entities but does not exist in the final ERD. Improving the ERD's readability and clarity is the primary goal.

An "PROGRAM" entity cluster, for example, can represent the relationships between entities like "SEMESTER," "COURSE," and "CLASS" in a university database system. It's important to keep in mind, though, that utilizing entity clusters may prevent specific entity properties from being shown, which might have an impact on important attributes and inheritance rules. In order to make the diagram more simplified at the expense of attribute visibility, it is therefore frequently advised against displaying attributes within entity clusters. To guarantee an efficient and understandable data model, this trade-off must be taken into account during the database design process.

c)

Fan Trap:

In the context of database systems and entity-relationship modeling (ERD), a particular kind of design problem that might arise during the definition of relationships between entities is referred to as a "fan trap." It's a type of design pitfall that might result in a database that inaccurately or inconsistently represents the connections in the actual world.

The primary feature of a fan trap is when one entity has two or more one-to-many (1:M) links with other entities, which leads to an unclear or poorly defined relationship between the other entities. It might be challenging to produce insightful reports or precisely extract data from the database as a result of this uncertainty.

Example can be:

When an entity, like DIVISION, is related to several other entities (like TEAM and PLAYER) via distinct one-to-many (1:M) connections, it can lead to a fan trap in database architecture. This makes it difficult to grasp how the many entities relate to one another. In a basketball league database, for instance, it would be difficult to determine which players are on which teams if DIVISION was mistakenly connected to both TEAM and PLAYER. Redesigning the database will remove ambiguity and ensure proper data representation. Division is tied to TEAM by a 1:M relationship, and TEAM is related to PLAYER via another 1:M relationship. This will fix fan traps. In order to avoid data discrepancies and enable precise querying and reporting, fan traps should be taken into consideration during database architecture.

D)

Generalization:

A specific kind of relationship in a database, known as a 1:1 (one-to-one) relationship, is called a generalization. It denotes a direct relationship between one entity and another, and vice versa. As an illustration, a department is headed by a professor, and a department has a professor as its chair. Foreign keys are used to implement this 1:1 relationship and show the relationships between the entities. It is a technique for guaranteeing data integrity and consistency in databases, but it should be applied carefully as it might result in intricate designs. Under some circumstances, generalization hierarchies—including 1:1 relationships—can be helpful for enhancing database architectures.

Specialization:

In computer database systems, specialization is arranging entities into a hierarchy in which subtypes, or lower-level things, are connected to higher-level entities (supertypes). In a business database, for example, you may have subtypes such as MANAGER, TECHNICIAN, and ACCOUNTANT under the supertype EMPLOYEE. Subtypes have distinct qualities, while entity supertypes show the shared traits of subtypes. Because of this arrangement, characteristics can be inherited by subtypes from supertypes. A subtype discriminator aids in determining an instance's subtype. Disjoint or overlapping constraints, which specify whether an instance can belong to one or more subtypes, are another aspect of specialization. Similarly, whether a subtype is required for each supertype instance depends on whether constraints are complete or partial. By effectively handling different qualities and organizing data relationships, specialization hierarchies enhance database models.

E)

Subtype Discriminator:

In the framework of entity-relationship modeling, which is frequently utilized in database architecture, a subtype discriminator is a crucial idea. It is essential for figuring out which subtype (a specialized or sub-category) a particular instance of a supertype object falls into.

In entity-relationship modeling, a "subtype discriminator" is a crucial idea that helps classify instances of a supertype entity into different subtypes based on the value of a given property. This approach maintains the distinct relationships and properties of each category while allowing the depiction of many data categories within a single entity. According to this method, a subtype is a more specialized version of the "supertype" object. Under the "EMPLOYEE" supertype, for instance, the subtypes "MANAGER," "TECHNICIAN," and "ACCOUNTANT" exist. An employee's role is determined by the subtype discriminator, the "EMP\_TYPE" property. This fundamental idea of database design makes sure that the structure of data complies with business needs and real-world circumstances.

Part 4

1)

Composite Primary Keys:

When one column in a database structure is unable to uniquely identify every record in a table, Composite Primary Keys are utilized. Instead, a unique identification is created by combining numerous columns. When distinctness cannot be ensured by individual columns alone, this is required.

Surrogate Key:

In order to guarantee that every entry in a database has a unique identifier, a fake primary key—typically an auto-incremented integer—is inserted as a surrogate key. When there isn't a good natural key or when the natural key might vary over time, it's frequently employed. Even in the event that the data they refer to changes, surrogate keys remain constant.

No Change Overtime:

With respect to the notion of "no change over time," this means that a primary key—natural or substitute—should be true throughout the duration of a record. Ensuring the consistency and integrity of data is crucial. A change in the primary key would cause data anomalies, which would compromise referential integrity and make it challenging to trace past data or build connections with other types of data. As a result, stability and uniqueness should be guaranteed by carefully selecting the main keys.

2)