HOMEWORK ASSIGNMENT 05

1.

The main purpose of logical database design is to construct a conceptual image of the database that does not depend on any database management system. It focuses on creating a conceptual representation of the data model that aligns with the business requirements and accurately reflects the relationships among different entities in the system for which we need to identify the different entities, their attributes and the relationships among those entities.

2.

Relations are described as two-dimensional data tables. However, not every table qualifies as a relation. There are specific properties that set relations apart from tables that lack a relational structure. Some of these properties are as follow:

a. Every relation (or table) within a database should have a unique name.
b. Each attribute on a specific row is associated with only one value; a relation does not allow multivalued attributes.
c. No two rows within a relation are identical. i.e., Each row should be unique.
d. Every attribute (or column) in a table should be assigned a unique name.
e. The arrangement of columns in a relation can be altered without impacting the meaning or utility of the relation. i.e., The order of columns (from left to right) holds no significance.
f. Similarly, the sequence of rows (from top to bottom) is inconsequential. The order of rows in a relation may be modified or stored in any sequence without affecting its meaning or use.

3.

A candidate key is an attribute that can uniquely identify the tuple in a relation. This characteristic guarantees that no two tuples can share an identical set of values. thereby eliminating the presence of duplicate data in the relation. The candidate keys can’t have null values, which further ensures the unique identification of a tuple.

4.

A referential integrity constraint is a rule that uphold consistency across the rows of two relations. According to this rule, if a relation contains a foreign key, it is a must that each foreign key value either corresponds to a primary key value in another relation, or the foreign key value must be null.

With referential integrity, the database management system (DBMS) blocks actions that could jeopardize the connections between tables. For instance, it prohibits the deletion of a row from the

table in cases if its primary key is used as foreign key in another table. This approach safeguards the integrity of data by ensuring the validity of relationships and preventing the introduction of inconsistent or corrupted data.

5.

A multi-valued attribute is an attribute that can have multiple values for a single entity within a relational database. Unlike simple attributes, which can have only one value for a specific entity, multi-valued attributes can have a set of values associated with them.

For e.g. Student and Extracurricular Activities:

- In a student information system, an entity representing students might have a multi-valued attribute named "Extracurricular Activities." Students can be involved in multiple activities, such as "Chess Club," "Debate Team," or "Science Olympiad," making it a multi-valued attribute.

To represent multi-valued attributes in a relational database, a separate table is often created. This table establishes a relationship between the main entity and its multi-valued attribute. For above example, there might be a separate table which indicates the relation between student and the extra curriculum activities they are involves in.

6.

The relational data model encompasses various constraints, which are rules that restrict acceptable values and operations. These rules are designed to maintain the data accuracy and integrity within the database. The primary categories of integrity constraints include domain constraints, entity integrity, and referential integrity.

a. Entity Integrity Constraint: It enforces the rule of having unique primary keys, which ensures that each row in a table is uniquely identified by its primary key. This prevents duplicate or null values in primary key columns.

b. Referential Integrity Constraint: It maintains consistency between tables by ensuring that foreign key relationships between tables are valid. It restricts a foreign key column must either match a primary key in another table or be null. Also, ensures the illegitimate deletion of any primary key if it has a reference to another table.

c. Domain Integrity Constraint: It defines the values allowed for a specific column, ensuring that data entered into the database falls within the specified range or set of values. This includes data types, check constraints, and default values.

There are numbers of reasons why Data integrity constraints play a crucial role in a database:

a. Ensuring Accuracy and Consistency: Constraints are crucial in preserving accurate and consistent data by prohibiting the values that are either invalid or inconsistent.

b. Preventing Data Anomalies: it reduces the likelihood of data anomalies, including errors during insertion, updating, or deletion.

c. Enforcing Business Rules: At the database level, constraints enforce the business rules and requirements. This ensures that the data aligns with predetermined standards, promoting adherence to organizational guidelines.

7.

Both primary key and unique key constraints enforce uniqueness, but the primary key is more restrictive. Although, both constraints serve different purposes, they both enforce data integrity in relational database.

Primary Key Constraint:

- The primary key constraint is responsible uniquely identify a tuple in a relation. It enforces the entity integrity by ensuring that there are no duplicate or null values in the primary key.
- The primary key should be unique.
- There should not be any null value.
- A relation can have only one primary key.

Unique Key Constraint:

- The unique key constraint ensures that values in a column or combination of columns are unique across all records in a table.
- It enforces the uniqueness of data but does not restrict null values.
- Also, a table can possess multiple unique keys.

8.

A foreign key constraint is used to enforce the relationship between two tables in a relational database, a column or a set of in one table acting as foreign key should correspond to the values in the primary key of another table.

It ensures referential integrity because:

- It defines the clear relationship between 2 tables where primary key of one table acts as foreign key of another table.
- It enforces referential integrity a foreign key values should either be primary key in another table or be null.
- It makes sure that there is no orphan rows in a table. An Orphaned rows are rows in a table when it contains a foreign key that references a non-existing primary key of another table.

- A foreign key constraint may allow for cascading actions such as cascading deletes or updates, if permitted.

9)

a) Draw a relational schema for PART SUPPLIER and show the functional dependencies.

1. In what normal form is this relation?
2. Develop a set of 3NF relations from PART SUPPLIER**.**

Part No	Description	Vendor Name	Address	Unit Cost
1234	Logic chip	Fast Chips	Cupertino	10.00
		Smart Chips	Phoenix	8.00
5678	Memory chip	Fast Chips	Cupertino	3.00
		Quality Chips	Austin	2.00
		Smart Chips	Phoenix	5.00

1. Relation name: PARTSUPPLIER
   PARTSUPPLIER (PartNo, Description, VendorName, Address, UnitCost)

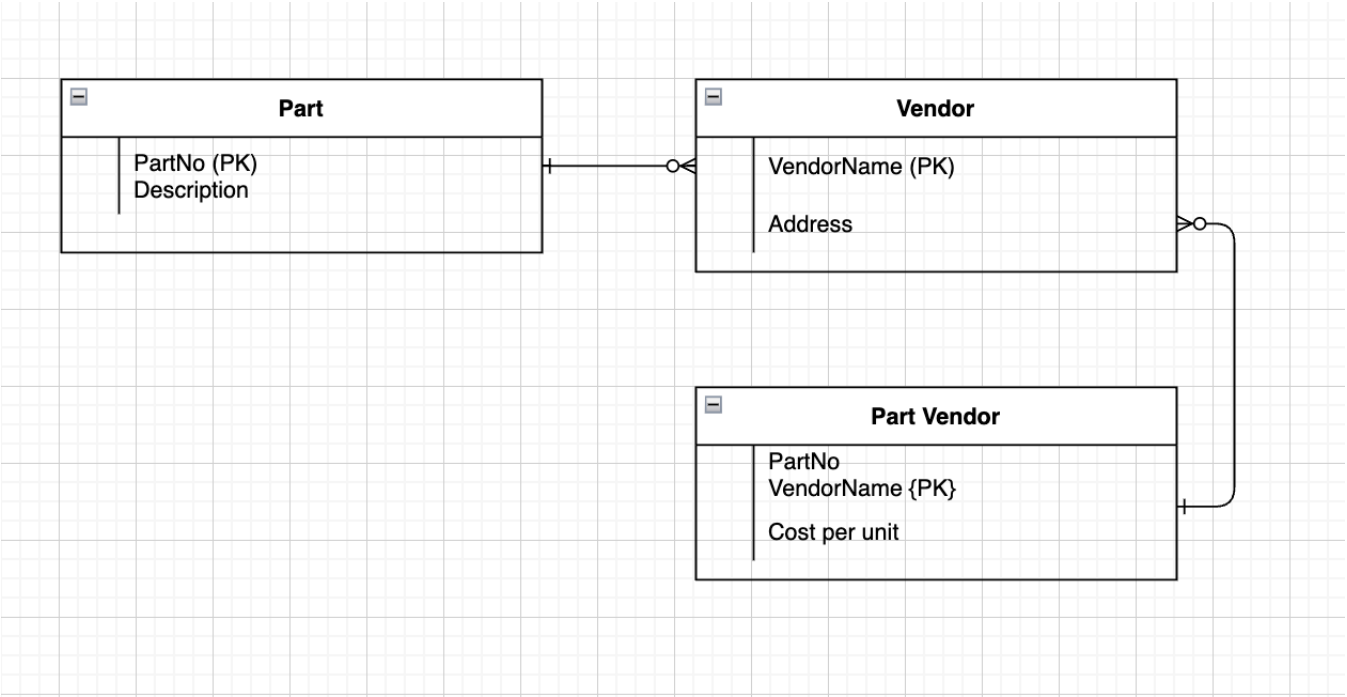
   Candidate Key: {PartNo, VendorName}

   Functional Dependency:
   PartNo -> {Description}
   VendorName -> {Address}
   {PartNo, VendorName} -> { UnitCost }

2. 1 Normal Form
3.

PartNo	Description	VendorName	Address	UnitCost
1234	Logic Chip	Fast Chips	Cupertino	10
1234	Logic Chip	Smart Chips	Phoenix	8
5678	Memory Chip	Fast Chips	Cupertino	3
5678	Memory Chip	Quality Chips	Austin	2
5678	Memory Chip	Smart Chips	Phoenix	5

ERD:


Part
PartNo (PK)
Description
Vendor
VendorName (PK)
Address
Part Vendor
PartNo
VendorName {PK}
Cost per unit