Homework Assignment 1
Chapter 3

1. Define the following terms: entity, attribute, attribute value, relationship instance, composite attribute, multivalued attribute, derived attribute, complex attribute, key attribute, and value set (domain).

=> Answer:-

i) **Entity**:

An entity is an object, person, place, or concept that can store data about it. In a database, an entity is represented as a table, with each row representing an instance of the entity.

For example, a “Student” is an entity, and each student is an instance of the entity.

ii) **Attribute**:

An attribute is a characteristic or property of an entity. In databases, attributes are equivalent to columns in a table.

For example, a “Student” entity might have attributes like “Student_ID”, “Name”, and “Date_of_Birth”.

iii) **Attribute Value**:

An attribute value is the actual data stored for an attribute in a specific instance of an entity.

For example, for the attribute “Name”, the value could be “John Doe” for a particular student entity.

iv) **Relationship Instance**:

A relationship instance is a specific relationship occurrence between two or more entities.

For example, in a “Student” and “Course” relationship, a relationship instance would be the specific enrollment of a student in a course.

v) **Composite Attribute**:

A composite attribute is an attribute that can be divided into smaller subparts, which represent more basic attributes with independent meanings.

For example, an address might be a composite attribute made up of street, city, state, and ZIP code.

vi) **Multivalued Attribute:**

A multivalued attribute is one that can have multiple values for a single entity.

For example, an attribute like “Phone Numbers” could hold more than one phone number for a person.

vii) **Derived Attribute:** A derived attribute is one whose value can be calculated from other attributes.

For example, “Age” can be derived from the “Date of Birth” attribute.

viii) **Complex Attribute:**

A complex attribute is an attribute that can be both composite and multivalued.

For example, an “Address” can have multiple subparts (composite) and a person might have multiple addresses (multivalued).

ix) **Key Attribute:**

A key attribute is an attribute that uniquely identifies an entity instance in a database. For example, “Student_ID” would be a key attribute for the “Student” entity because it uniquely identifies each student.

x) **Value Set (Domain):**

A value set (also known as a domain) is the set of possible values that an attribute can take.

For example, the domain of the “Gender” attribute might be {Male, Female, Other}. The domain defines what values are valid for that particular attribute.

2. When is the concept of a weak entity used in data modeling? Define the terms owner entity type, weak entity type, identifying relationship type, and partial key.

=> Answer:-

- The concept of a weak entity is used in data modeling when an entity cannot be uniquely identified by its own attributes alone. It requires a relationship with another entity, called the owner, to form a unique identifier.
- A weak entity typically exists in cases where the entity's existence is dependent on another entity, such as a "Dependent" being dependent on an "Employee".
- The weak entity concept is applied when certain entities rely on another for identification.

TERMS:-

i) **Owner Entity Type:**

The owner entity type is the entity that a weak entity depends on for its identification. It provides the primary key that, when combined with the weak entity's partial key, uniquely identifies instances of the weak entity.

For example, in a relationship between "Employee" and "Dependent", the "Employee" is the owner entity.

ii) **Weak Entity Type:**

A weak entity type is an entity that cannot be uniquely identified by its own attributes alone. Instead, it must use attributes from another (owner) entity along with its own partial key to form a unique identifier.

For example, "Dependent" is a weak entity because its identity is tied to the "Employee" (owner entity) it depends on.

iii) **Identifying Relationship Type:**

The identifying relationship type is a relationship between a weak entity and its owner entity. In the case of an "Employee" and "Dependent", the identifying relationship is the association between these two entities.

iv) **Partial Key:**

A partial key is an attribute (or set of attributes) of a weak entity that can uniquely identify the weak entity’s instances, but only in conjunction with the primary key of the owner entity.

For example, a “Dependent” entity might have a partial key like “Dependent Name”, which, when combined with the “Employee_ID” of the owning “Employee”, uniquely identifies each dependent.

3. Composite and multivalued attributes can be nested to any number of levels. Suppose we want to design an attribute for a STUDENT entity type to keep track of previous college education. Such an attribute will have one entry for each college previously attended, and each such entry will be composed of college name, start and end dates, degree entries (degrees awarded at that college, if any), and transcript entries (courses completed at that college, if any). Each degree entry contains the degree name and the month and year the degree was awarded, and each transcript entry contains a course name, semester, year, and grade. Design an attribute to hold this information.

Use the conventions in Figure 3.5:-

{Address_phone({Phone(Area_code,Phone_number)},Address(Street_address(Number,Street,Apartment_number),City,State,Zip))}

A complex attribute: Address_phone.

=> Answer:-

Using the conventions provided, we can represent the complex and nested attribute for previous college education for a STUDENT entity as follows:

```
Previous_college_education({
    College(
        College_name,
        Start_date,
        End_date,
        Degree({Degree_name, Degree_awarded_month,
Degree_awarded_year}),
        Transcript({
            Course(Course_name, Semester, Year, Grade)
        })
    )
})
```

Explanation:

i) Previous_college_education: This is the multivalued and composite attribute that will hold one entry for each college the student previously attended.

ii) College: Each college entry includes:

- College_name: The name of the college.
- Start_date: The date the student started at the college.
- End_date: The date the student finished at the college.

iii) Degree: A multivalued composite attribute because a student may have received multiple degrees from the same college. Each degree includes:

- Degree_name: The name of the degree awarded (e.g., “Bachelors of Science”).
- Degree_awarded_month: The month the degree was awarded.
- Degree_awarded_year: The year the degree was awarded.

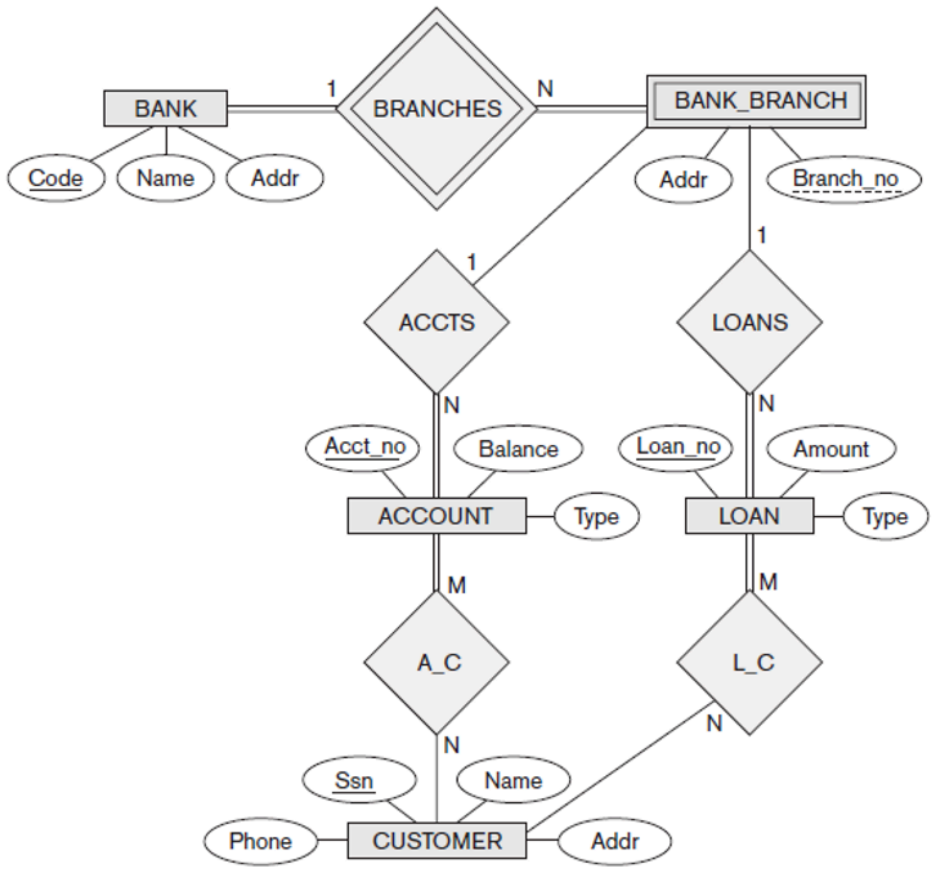
iv) Transcript: A multivalued composite attribute containing the courses the student completed at that college. Each transcript includes:

- Course_name: The name of the course.
- Semester: The semester during which the course was completed.
- Year: The year the course was completed.
- Grade: The grade the student received in the course.

This attribute design allows for a detailed and nested structure to store information about previous college education, degrees awarded, and transcripts.

4. Consider the ER diagram shown in Figure 3.22 for part of a BANK database. Each bank can have multiple branches, and each branch can have multiple accounts and loans. a. List the strong (nonweak) entity types in the ER diagram. b. Is there a weak entity type? If so, give its name, partial key, and identifying relationship. c. What constraints do the partial key and the identifying relationship of the weak entity type specify in this diagram? d. List the names of all relationship types, and specify the (min, max) constraint on each participation of an entity type in a relationship type. Justify your choices. e. List concisely the user requirements that led to this ER schema design. f. Suppose that every customer must have at least one account but is restricted to at most two loans at a time, and that a bank branch cannot have more than 1,000 loans. How does this show up on the (min, max) constraints?

Figure 3.22
An ER diagram for a BANK database schema.



=> Answer:-

Analyzing the ER Diagram:

i) **Strong Entity Types**

The strong entity types in the ER diagram are:

- BANK
- BRANCHES
- ACCTS
- LOANS
- ACCOUNT
- LOAN
- CUSTOMER

ii) **Weak Entity Type**

Yes, there is a weak entity type:

Name: A_C (Account-Customer)

Partial Key: Ssn (Social Security Number)

Identifying Relationship: ACCOUNT (Account)

iii) **Constraints of Partial Key and Identifying Relationship**

The partial key (Ssn) and identifying relationship (ACCOUNT) specify that an account-customer association can only exist if it is associated with an account.

The Ssn, combined with the account number, uniquely identifies an account-customer association.

iv) **Relationship Types and Constraints**

- **BANK-BRANCHES:** (1, N) - A bank can have one or more branches.
- **BANK-BANK_BRANCH:** (1, 1) - A bank branch is associated with exactly one bank.
- **BRANCHES-ACCTS:** (1, N) - A branch can have one or more accounts.
- **BRANCHES-LOANS:** (1, N) - A branch can have one or more loans.
- **ACCOUNT-A_C:** (1, N) - An account can have one or more account-customer associations.
- **LOAN-L_C:** (1, N) - A loan can have one or more loan-customer associations.

- **A_C-CUSTOMER:** (1, 1) - An account-customer association is associated with exactly one customer.
- **L_C-CUSTOMER:** (1, 1) - A loan-customer association is associated with exactly one customer.

v) **User Requirements**

- Based on the ER diagram, the user requirements likely include:
- Tracking information about banks, branches, accounts, loans, and customers.
- Associating accounts and loans with specific banks and branches.
- Associating customers with accounts and loans.
- Maintaining information about customers (name, SSN, phone number, address).
- Tracking the account balance and loan amount.

vi) **Updated Constraints**

Given the additional constraints:

- **ACCOUNT-A_C:** (1, 2) - A customer must have at least one account but is restricted to at most two loans.
- **BRANCHES-LOANS:** (1, 1000) - A bank branch cannot have more than 1,000 loans.