SOFTWARE ENGINEERING PRACTICAL DOCUMENT

PRACTICAL NO: 3

AIM: Study and implementation of Entity Relationship Diagrams.

SOLUTION

What is ER Model?

ER Model stands for Entity Relationship Model is a high-level conceptual data model diagram. ER model helps to systematically analyse data requirements to produce a well-designed database. The ER Model represents real-world entities and the relationships between them.

ER Diagrams Symbols & Notations

Entity Relationship Diagram Symbols & Notations mainly contains three basic symbols which are rectangle, oval and diamond to represent relationships between elements, entities and attributes. There are some sub-elements which are based on main elements in ERD Diagram. ER Diagram is a visual representation of data that describes how data is related to each other using different ERD Symbols and Notations.

Following are the main components and its symbols in ER Diagrams:

Rectangles: This Entity Relationship Diagram symbol represents entity types

Ellipses: Symbol represent attributes

Diamonds: This symbol represents relationship types

Lines: It links attributes to entity types and entity types with other relationship types

Primary key: attributes are underlined

Double Ellipses: Represent multi-valued attributes

ER Diagram Symbols

Components of the ER Diagram

This model is based on three basic concepts:

1) Entities

2) Attributes

3) Relationships

WHAT IS ENTITY?

A real-world thing either living or non-living that is easily recognizable and nonrecognizable. It is anything in the enterprise that is to be represented in our database. It may be a physical thing or simply a fact about the enterprise or an event that happens in the real world.

An entity can be place, person, object, event or a concept, which stores data in the database. The characteristics of entities are must have an attribute, and a unique key. Every entity is made up of some 'attributes' which represent that entity.

Examples of entities:

Person: Employee, Student, Patient

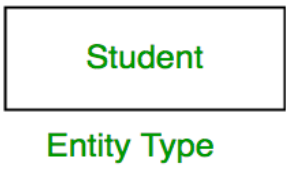
Place: Store, Building

Object: Machine, product, and Car

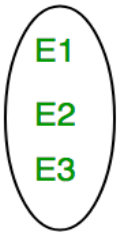
Event: Sale, Registration, Renewal

Concept: Account, Course

Entity type: It is a group of objects with the same properties that are identified by the enterprise as having an independent existence. The basic concept of the ER model is the entity type that is used to represent a group of 'objects' in the 'real world' with the same properties. An entity type has an independent existence within a database.



Entity Type



Entity Set

Types of entities based on Unique Identification –

1) Strong entity

2) Weak entity

Strong entity

Strong Entity is independent to any other entity in the schema.

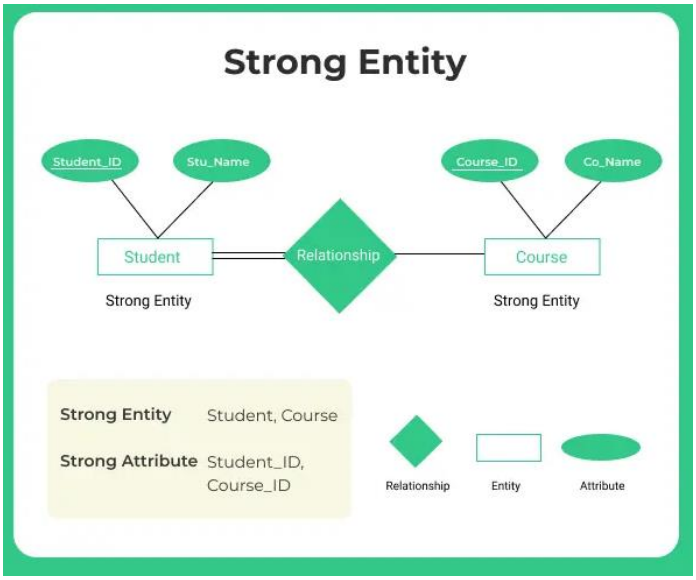
Simply strong entity is nothing but an entity set having a primary key attribute or a table which consists of a primary key column

The primary key of the strong entity is represented by underlining it.

Representation

The strong entity is represented by a single rectangle.

Relationship between two strong entities is represented by a single diamond.



Examples for the strong entity

Consider the ER diagram which consists of two entities student and course

Student entity is a strong entity because it consists of a primary key called student id which is enough for accessing each record uniquely

The same way, course entity contains of course ID attribute which is capable of uniquely accessing each row.

Weak entity

A weak entity is an entity set that does not have sufficient attributes for Unique Identification of its records

Simply a weak entity is nothing but an entity which does not have a primary key attribute

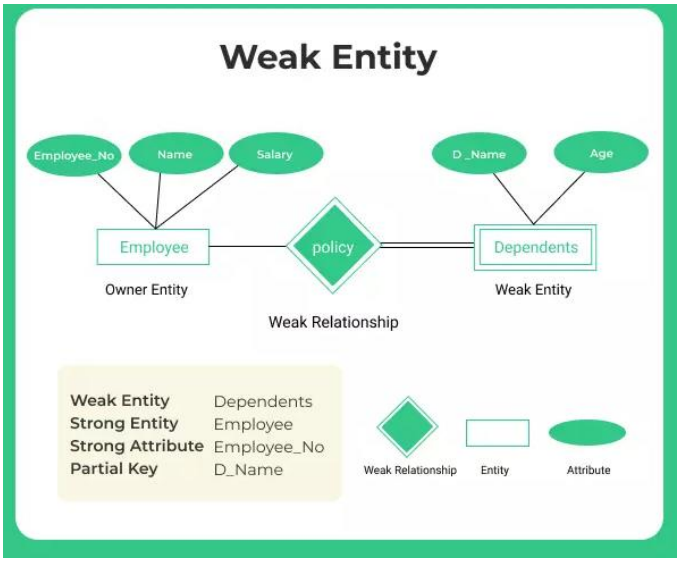
It contains a partial key called as discriminator which helps in identifying a group of entities from the entity set

Discriminator is represented by underlining with a dashed line

Representation

A double rectangle is used for representing a weak entity set

The double diamond symbol is used for representing the relationship between a strong entity and weak entity which is known as identifying relationship.



Example for weak entity

In the ER diagram, we have two entities Employee and Dependents.

Employee is a strong entity because it has a primary key attribute called Employee number (Employee No) which is capable of uniquely identifying all the employee.

Unlike Employee, Dependents is weak entity because it does not have any primary key.

D_Name along with the Employee No can uniquely identify the records of Depends. So here the D_Name (Depends Name) is partial key.

Attribute(s):

Attributes are the properties which define the entity type. For example, Roll No, Name, DOB, Age, Address, Mobile No are the attributes which defines entity type Student. In ER diagram, attribute is represented by an oval.

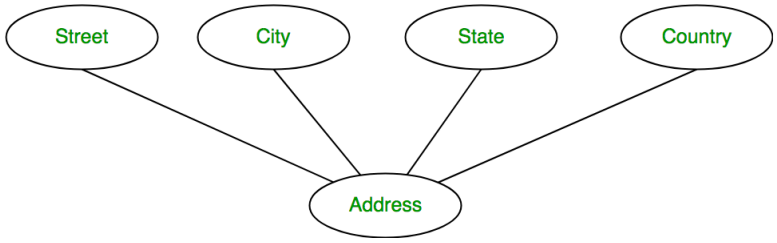
Types of Attributes:

1. Key Attribute –

The attribute which uniquely identifies each entity in the entity set is called key attribute. For example, Roll No will be unique for each student. In ER diagram, key attribute is represented by an oval with underlying lines.

2. Composite Attribute –

An attribute composed of many other attributes is called as composite attribute. For example, Address attribute of student Entity type consists of Street, City, State, and Country. In ER diagram, composite attribute is represented by an oval comprising of ovals.



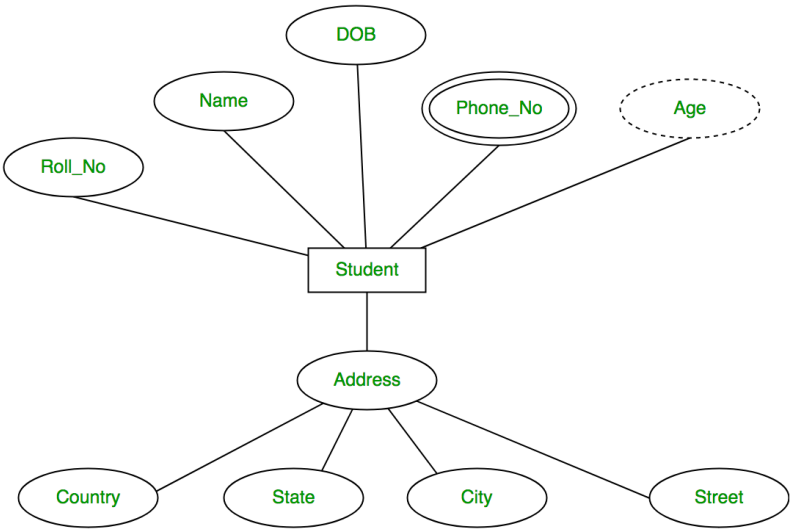
3. Multivalued Attribute –

An attribute consisting more than one value for a given entity. For example, Phone No (can be more than one for a given student). In ER diagram, multivalued attribute is represented by double oval.

4. Derived Attribute –

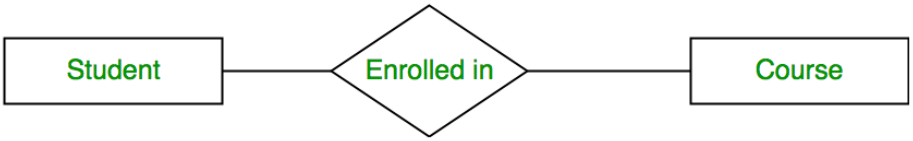
An attribute which can be derived from other attributes of the entity type is known as derived attribute. e.g.; Age (can be derived from DOB). In ER diagram, derived attribute is represented by dashed oval.

The complete entity type **Student** with its attributes can be represented as:

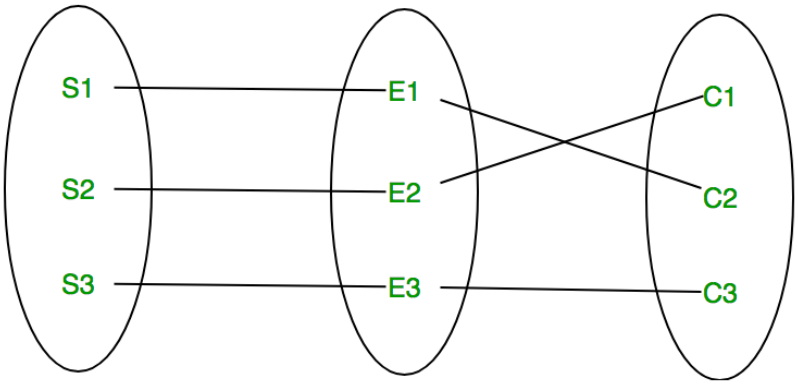


Relationship Type and Relationship Set:

A relationship type represents the association between entity types. For example,' Enrolled in' is a relationship type that exists between entity type Student and Course. In ER diagram, relationship type is represented by a diamond and connecting the entities with lines.



A set of relationships of same type is known as relationship set. The following relationship set depicts S1 is enrolled in C2, S2 is enrolled in C1 and S3 is enrolled in C3.

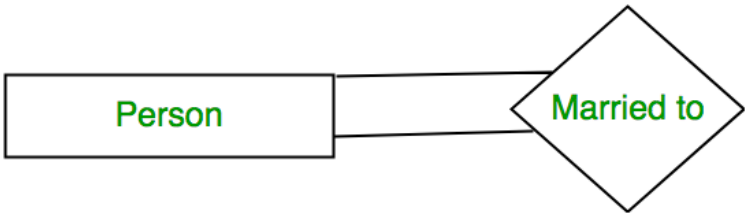


Degree of a relationship set:

The number of different entities sets participating in a relationship set is called as degree of a relationship set.

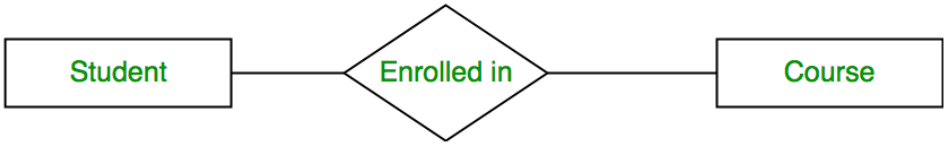
1. Unary Relationship –

When there is only ONE entity set participating in a relation, the relationship is called as unary relationship. For example, one person is married to only one person.



2. Binary Relationship –

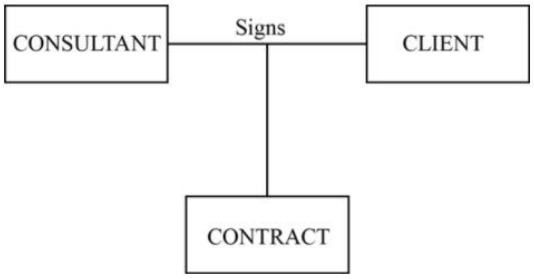
When there are TWO entities set participating in a relation, the relationship is called as binary relationship. For example, Student is enrolled in Course.



3. Ternary Relationship –

When there are THREE entities set participating in a relation, the relationship is called as ternary relationship. For example, Consultant, client and contract are three different entities with different attributes.

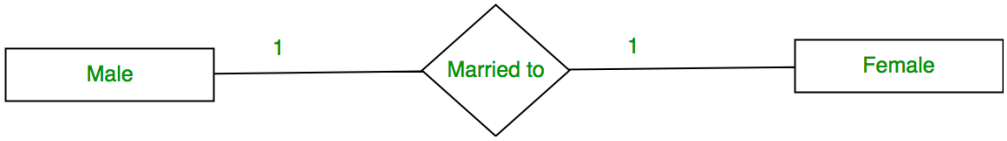
These three entities are related with a single relationship called “signs”.



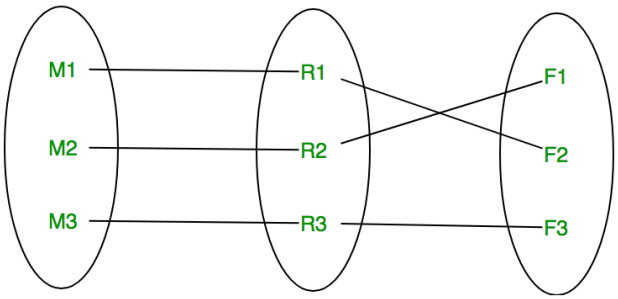
Cardinality:

The number of times an entity of an entity set participates in a relationship set is known as cardinality. Cardinality can be of different types:

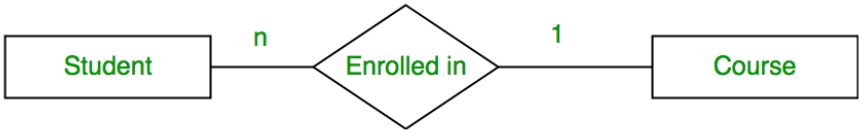
1. One to one – When each entity in each entity set can take part only once in the relationship, the cardinality is one to one. Let us assume that a male can marry to one female and a female can marry to one male. So, the relationship will be one to one.



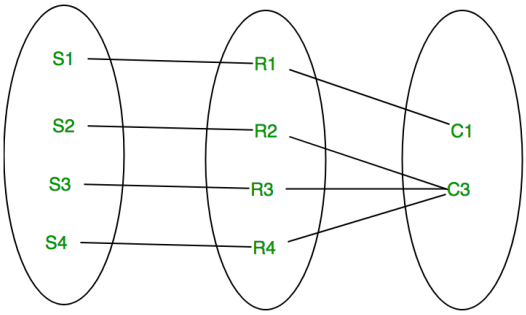
Using Sets, it can be represented as:



2. Many to one – When entities in one entity set can take part only once in the relationship set and entities in other entity set can take part more than once in the relationship set, cardinality is many to one. Let us assume that a student can take only one course but one course can be taken by many students. So, the cardinality will be n to 1. It means that for one course there can be n students but for one student, there will be only one course.



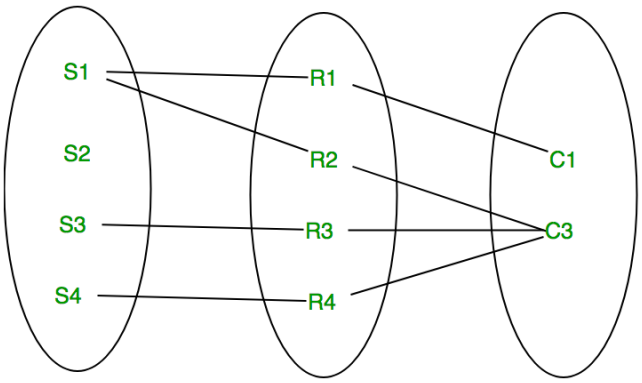
Using Sets, it can be represented as:



In this case, each student is taking only 1 course but 1 course has been taken by many students.

3. Many to many – When entities in all entity sets can take part more than once in the relationship cardinality is many to many. Let us assume that a student can take more than one course and one course can be taken by many students. So, the relationship will be many to many.

Using sets, it can be represented as:



In this example, student S1 is enrolled in C1 and C3 and Course C3 is enrolled by S1, S3 and S4. So it is many to many relationships.

Keys

Keys play an important role in the relational database.

It is used to uniquely identify any record or row of data from the table. It is also used to establish and identify relationships between tables.

For example, ID is used as a key in the Student table because it is unique for each student. In the PERSON table, passport_number, license_number, SSN are keys since they are unique for each person.

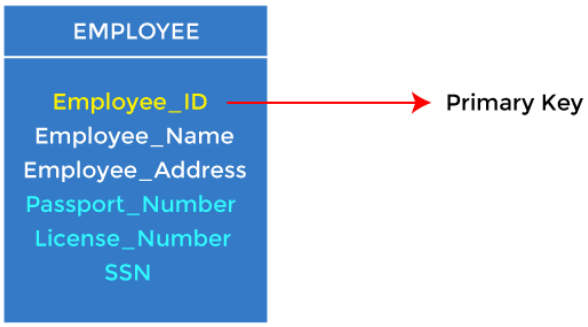
Types of keys:

1. Primary key

It is the first key used to identify one and only one instance of an entity uniquely. An entity can contain multiple keys, as we saw in the PERSON table. The key which is most suitable from those lists becomes a primary key.

In the EMPLOYEE table, ID can be the primary key since it is unique for each employee. In the EMPLOYEE table, we can even select License_Number and Passport_Number as primary keys since they are also unique.

For each entity, the primary key selection is based on requirements and developers.

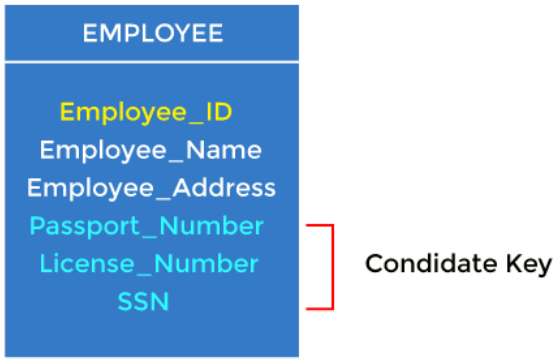


2. Candidate key

A candidate key is an attribute or set of attributes that can uniquely identify a tuple.

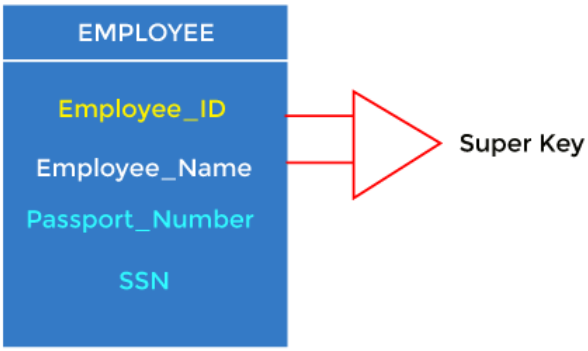
Except for the primary key, the remaining attributes are considered a candidate key. The candidate keys are as strong as the primary key.

For example: In the EMPLOYEE table, id is best suited for the primary key. The rest of the attributes, like SSN, Passport_Number, License_Number, etc., are considered a candidate key.



3. Super Key

Super key is an attribute set that can uniquely identify a tuple. A super key is a superset of a candidate key.



For example: In the above EMPLOYEE table, for (EMPLOEE_ID, EMPLOYEE_NAME), the name of two employees can be the same, but their EMPLYEE_ID can't be the same. Hence, this combination can also be a key.

The super key would be EMPLOYEE-ID (EMPLOYEE_ID, EMPLOYEE-NAME), etc.

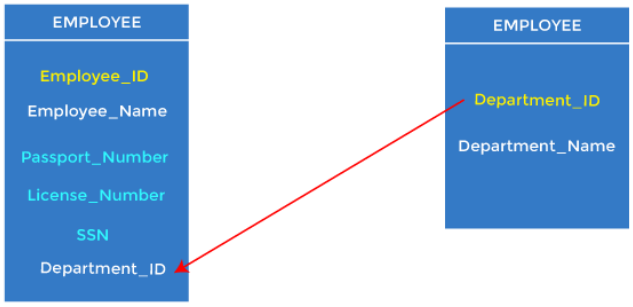
4. Foreign key

Foreign keys are the column of the table used to point to the primary key of another table.

Every employee works in a specific department in a company, and employee and department are two different entities. So, we can't store the department's information in the employee table. That's why we link these two tables through the primary key of one table.

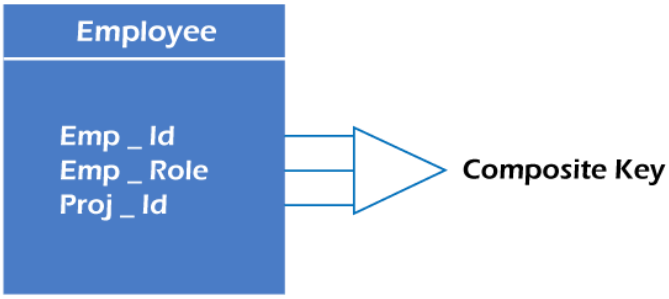
We add the primary key of the DEPARTMENT table, Department_Id, as a new attribute in the EMPLOYEE table.

In the EMPLOYEE table, Department_Id is the foreign key, and both the tables are related.



5. Composite key

Whenever a primary key consists of more than one attribute, it is known as a composite key. This key is also known as Concatenated Key.



For example, in employee relations, we assume that an employee may be assigned multiple roles, and an employee may work on multiple projects simultaneously. So, the primary key will be composed of all three attributes, namely Emp_ID, Emp_role, and Proj_ID in combination. So, these attributes act as a composite key since the primary key comprises more than one attribute.

Example of a Sequence Diagram

(ONLINE SHOPPING SYSTEM):

This Online Shopping system database was made based on online shopping management requirements. The system can encode the shoppers' information upon the buying/ordering. The admin can have access to the shoppers' information as well as their transactions. They can handle the data needed in managing information and records the request made by the customers.

The features included in the system ER diagram were the security and monitoring of the seller information and customers' information status. These features were also listed and recorded in reports that served as the history of transactions done in the system.

Here's the database design for Online Shopping system with ER Diagram of Online Shopping system that includes Tables, Schema, PDF, etc.

Shopping Management

This feature gives the system or the user the customers' shopping information. This will serve as the basis of the admin on what to prepare or invokes the system to give the order of the customer.

Customer Management

This feature plays a big role for the system because this gathers the important information of the customers. This information was used for determining the choices of the customer orders and their transactions.

Manage Transactions

In this feature, the admin can have and monitor the transactions that are made by the customers. This will save the transaction information as well as the important matters that were done during shopping like ordering and paying.

Transaction and Reports Management

This feature will store the transactions made including their information and the reports every transaction in the corresponding timetables.

Online Shopping System ER Diagram Tables

These tables below provide the complete database tables details such as Field Name, Descriptions, data types, character lengths.

Table Name: Customer

Field	Description	Type	Length
Customer_ID (PK)	Customer ID	Int	11
name	Customer Name	Varchar	255
contact_add	Customer Contact	Int	11
address	Customer Address	Text	

Table Name: Categories

Field	Description	Type	Length
category_ID (PK)	Category ID	Int	11
category_name	Category Name	Varchar	255
category_type	Category Type	Varchar	255

Table Name: Shopping Order

Field	Description	Type	Length
order_ID (PK)	Order ID	Int	11
customer_ID (FK)	Customer ID	Int	11
date	Date of Order	Date	

Table Name: Deliveries

Field	Description	Type	Length
acc_ID (PK)	Account ID	Int	11
customer_ID (FK)	Customer ID	Int	11
Date	Date of Delivery	Date	

Table Name: Products

Field	Description	Type	Length
product_ID (PK)	Product ID	Int	11
category_ID (FK)	Category ID	Int	11
product_name	Product Name	Varchar	255

Table Name: Seller

Field	Description	Type	Length
seller_ID (PK)	Seller ID	Int	11
product_ID (FK)	Product Name	Int	11
seller_name	Seller Name	Varchar	255

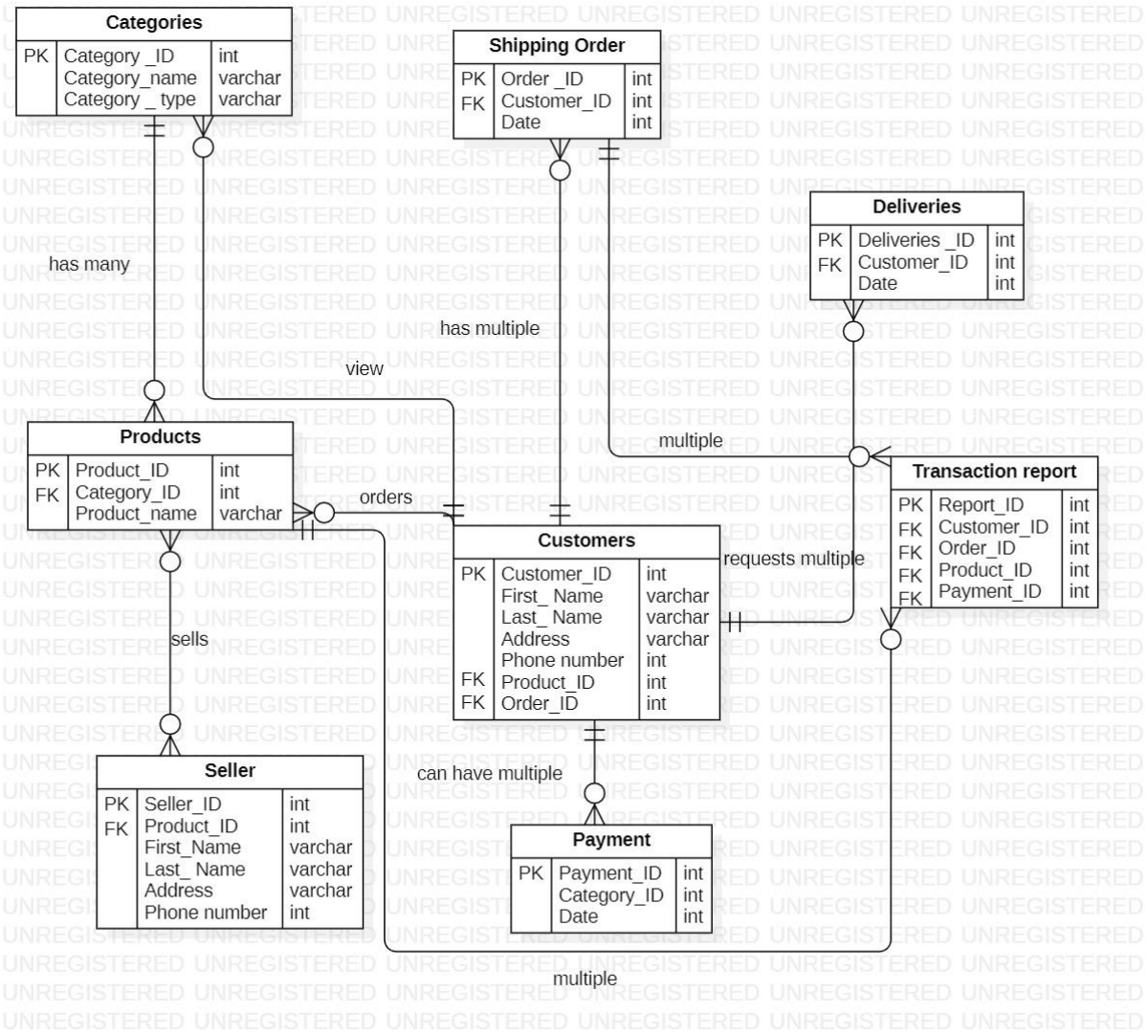
Table Name: Payment

Field	Description	Type	Length
payment_ID (PK)	Payment ID	Int	11
customer_ID (FK)	Customer ID	Int	11
date	Date of Payment	Date	

Table Name: Transaction Report

Field	Description	Type	Length
report_ID (PK)	Report ID	Int	11
customer_ID (FK)	Customer ID	Int	11
order_ID (FK)	Order Id	Int	11
product_ID (FK)	Product Id	Int	11
payment_Id (FK)	Payment Id	Int	11

Entity Relationship Diagram for Online Shopping System is as follows:


Categories
PK Category _ID int
Category_name varchar
Category _ type varchar
Shipping Order
PK Order _ID int
FK Customer_ID int
Date int
Deliveries
PK Deliveries _ID int
FK Customer_ID int
Date int
has many
has multiple
view
multiple
Products
PK Product_ID int
FK Category_ID int
Product_name varchar
orders
Transaction report
PK Report_ID int
FK Customer_ID int
FK Order_ID int
FK Product_ID int
FK Payment_ID int
Customers
PK Customer_ID int
First_ Name varchar
Last_ Name varchar
Address varchar
Phone number int
FK Product_ID int
FK Order_ID int
requests multiple
sells
Seller
PK Seller_ID int
FK Product_ID int
First_Name varchar
Last_ Name varchar
Address varchar
Phone number int
can have multiple
Payment
PK Payment_ID int
Category_ID int
Date int
multiple