Data Modelling Peer Learning Document

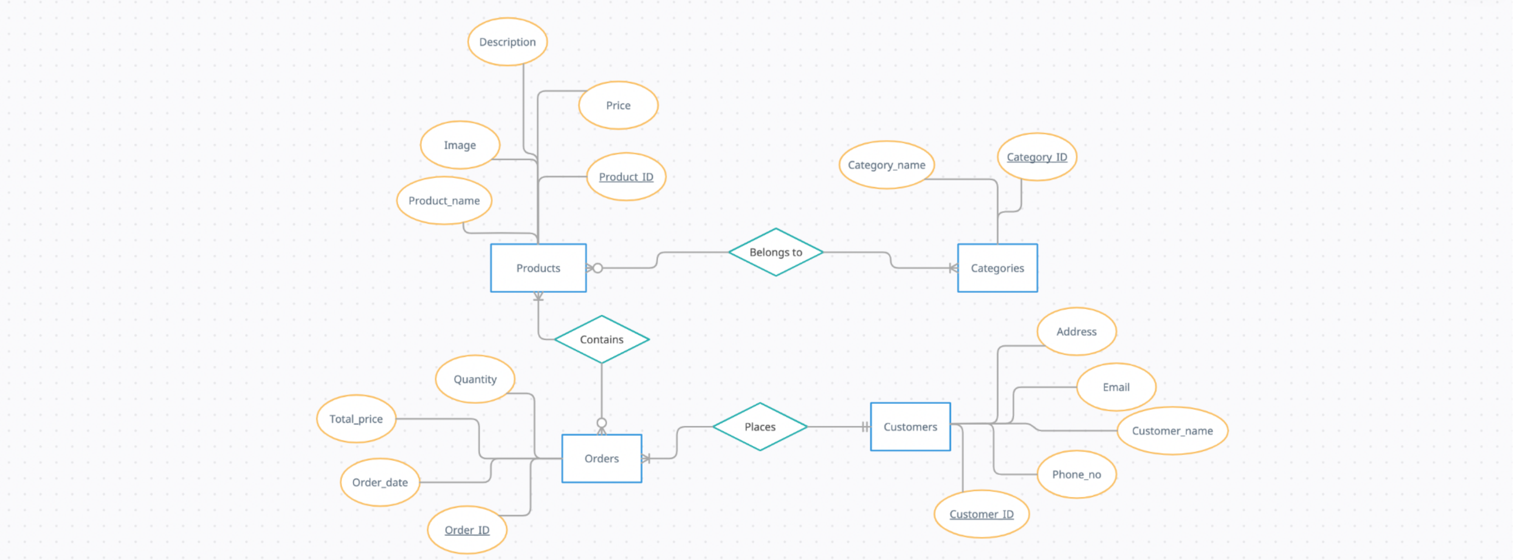
Question 1: Design a data model for a simple e-commerce system (duration: 1.5-2 hours).

Instructions:

1. Identify the entities in an e-commerce system, including products, categories, customers, and orders.
2. Determine the attributes for each entity, such as product name, price, customer name, order date, etc.
3. Identify the relationships between entities, such as a product can belong to multiple categories and a customer can place multiple orders.
4. Determine the primary keys for each entity, such as product ID for products, customer ID for customers, etc.
5. Create an ER diagram to visually represent the data model.
6. Write a brief description of the data model, including its purpose, entities, relationships, and any assumptions or constraints.

Solution:

ER Diagram:



Description:

This data model represents a simple e-commerce system that sells products to customers. The system contains four entities: products, categories, customers, and orders. Products have attributes such as product id, name, description, price, and image. Categories have attributes such as name and category id. Customers have attributes such as customer id, name, email, phone number, and address. Orders have attributes such as order id, order date, total price, quantity.

Purpose:

The purpose of the above ER model is to provide a data model for a simple e-commerce system. The model describes the various entities, attributes, and relationships that exist within the system. The entities in the model include products, categories, customers, and orders.

Overall, the purpose of the ER model is to provide a clear and structured representation of the data that is required to build and operate an e-commerce system. It serves as a blueprint for the design and development of the system, ensuring that all necessary data is captured and stored appropriately

Entities:

Products: A product that can be sold on the e-commerce system.

Categories: A category that a product can belong to.

Customers: A customer who can place orders on the e-commerce system.

Orders: A record of an order placed by a customer that contains information about the products, quantities, and total price.

Attributes:

Products: ID, name, description, price, image.

Categories: ID, name.

Customers: ID, name, email, phone number, address.

Orders: ID, order date, total price, quantity.

Relationships:

A product can belong to multiple categories and a category can have multiple products (many-to-many).

A customer can place multiple orders, but an order can only be placed by one customer (one-to-many).

An order can contain multiple products and a product can be in multiple orders (many-to-many).

The ER diagram visually represents the relationships between the entities, with arrows indicating the direction of the relationships. The ‘Contains’ is a junction that allows for a many-to-many relationship between orders and products. The ‘Belongs to’ is a junction that allows for a many-to-many relationship between categories and products. The ‘Places’ is a junction that allows for a many-to-many relationship between orders and customers.

Assumptions:

The data model assumes that each product can belong to multiple categories, each customer can place multiple orders, and each order can contain multiple products. The primary keys for each entity are ID for products, categories, and customers, and ID for orders.

Constraints:

Primary Keys (Entity Integrity)

Products: product ID

Categories: category ID

Customers: customer ID

Orders: order ID

Foreign Keys (Referential Integrity)

Belongs to (Intersection Table between Products and Categories):

Composite Key (Product ID, Category ID)

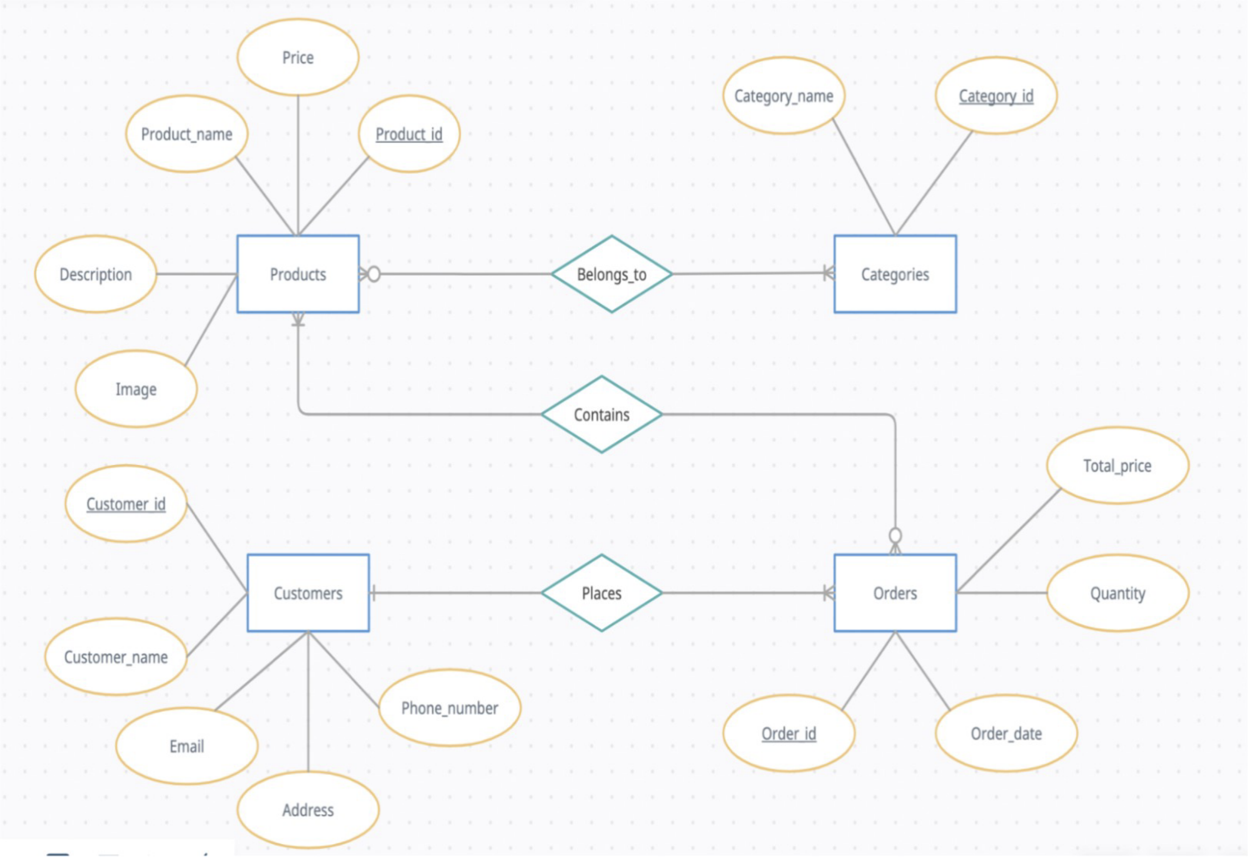
Contains(Intersection Table between Products and Orders):

Composite Key (Product ID, Order ID)

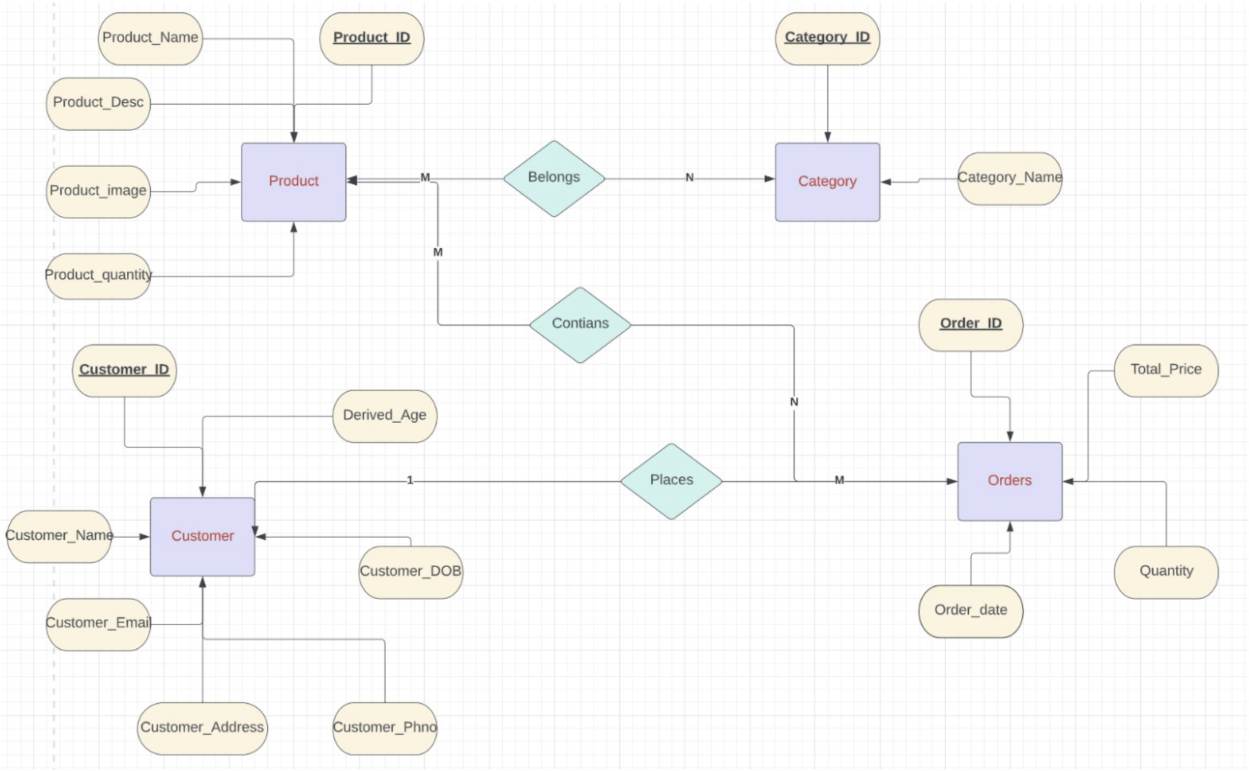
Places(Relationship between Customers and Orders):

Foreign Key (Customer ID migrates from Customer table to Order table)

Purushottam’ Solution:



Atul’ solution:



Difference:

- The Entities and number of entities are similar for all of us.
- The relationship between products and categories (many to many) and Order and Products (many to many) is the same for all of us.
- Assumption is that each customer places at least one order.
- Some attributes of product and order are different from Srinivas,
    - There is a **product_quantity** field in Atul’s product entity to track the quantity available but there is no **price** as in my solution.
    - There are ***dob and derived_age*** fields in Atul’s customer entity.

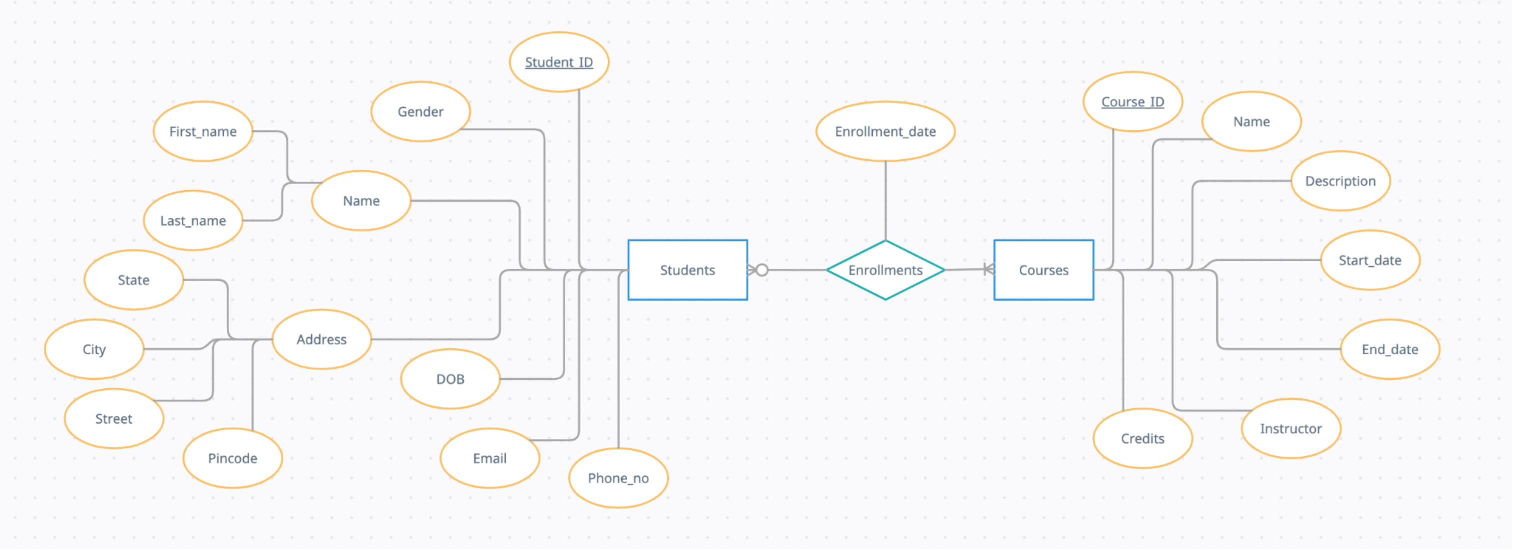
- The constraints for all of our data models are the same where we have an intersection table between *product and category* entities and product and order entities.
- And Atul's solution uses **Chen notation** to present cardinality in ER diagrams whereas I have used **Crow's foot notation** to do the same.

Question 2: Design a data model for a student enrollment system (duration: 1.5-2 hours).

Instructions:

1. Identify the entities in a student enrollment system, including students, courses, and enrollments.
2. Determine the attributes for each entity, such as student name, course name, enrollment date, etc.
3. Identify the relationships between entities, such as a student can enroll in multiple courses and a course can have multiple students.
4. Determine the primary keys for each entity, such as student ID for students, course ID for courses, etc.
5. Create an ER diagram to visually represent the data model.
6. Write a brief description of the data model, including its purpose, entities, relationships, and any assumptions or constraints.

Solution:



Description:

The purpose of this data model is to represent a student enrollment system, where students can enroll in multiple courses and courses can have multiple students. The model consists of two entities: students, courses, and a relationship named enrollments. Students and courses have their own attributes, such as student_id, gender, name, address, dob, email, phone number and course id, name, course description, start date, end date, instructor, credits respectively. Enrollments represent the relationship between students and courses and have attributes such as enrollment date. The primary keys are student ID, course ID.

Purpose:

The purpose of the above data model is to represent a student enrollment system. The model is designed to store and manage information related to students, courses, and enrollments. It provides a structure for organizing and storing data related to these entities, including their attributes and relationships. The model can be used to track which courses a student has enrolled in, which students are enrolled in a particular course, when enrollments were made. The data model can be used by educational institutions, such as schools or universities, to manage student enrollment and course registration processes. It can also be used to generate reports, analyze enrollment trends, and make informed decisions about course offerings and student performance.

Entities:

Students: This entity represents the students who enroll in courses. Its attributes include student ID, name, email, address, gender, dob and phone number.

Courses: This entity represents the courses that students can enroll in. Its attributes may include course ID, name, description, instructor, start date, end date and credits.

Attributes:

Students: student ID, name, email, address, gender, dob and phone number

Courses: course ID, name, description, instructor, start end, end date and credit

Relationships:

A student can enroll in multiple courses.

A course can have multiple students.

The ER diagram visually represents the relationships between the entities, with arrows indicating the direction of the relationships. The ‘Enrollments’ is a junction that allows for a many-to-many relationship between students and courses.

Assumptions:

This data model assumes that a student can enroll in multiple courses and a course can have multiple students. It also assumes that each enrollment is unique and can be identified by a composite key referencing student id and course id.

Constraints:

Primary Keys (Entity Integrity)

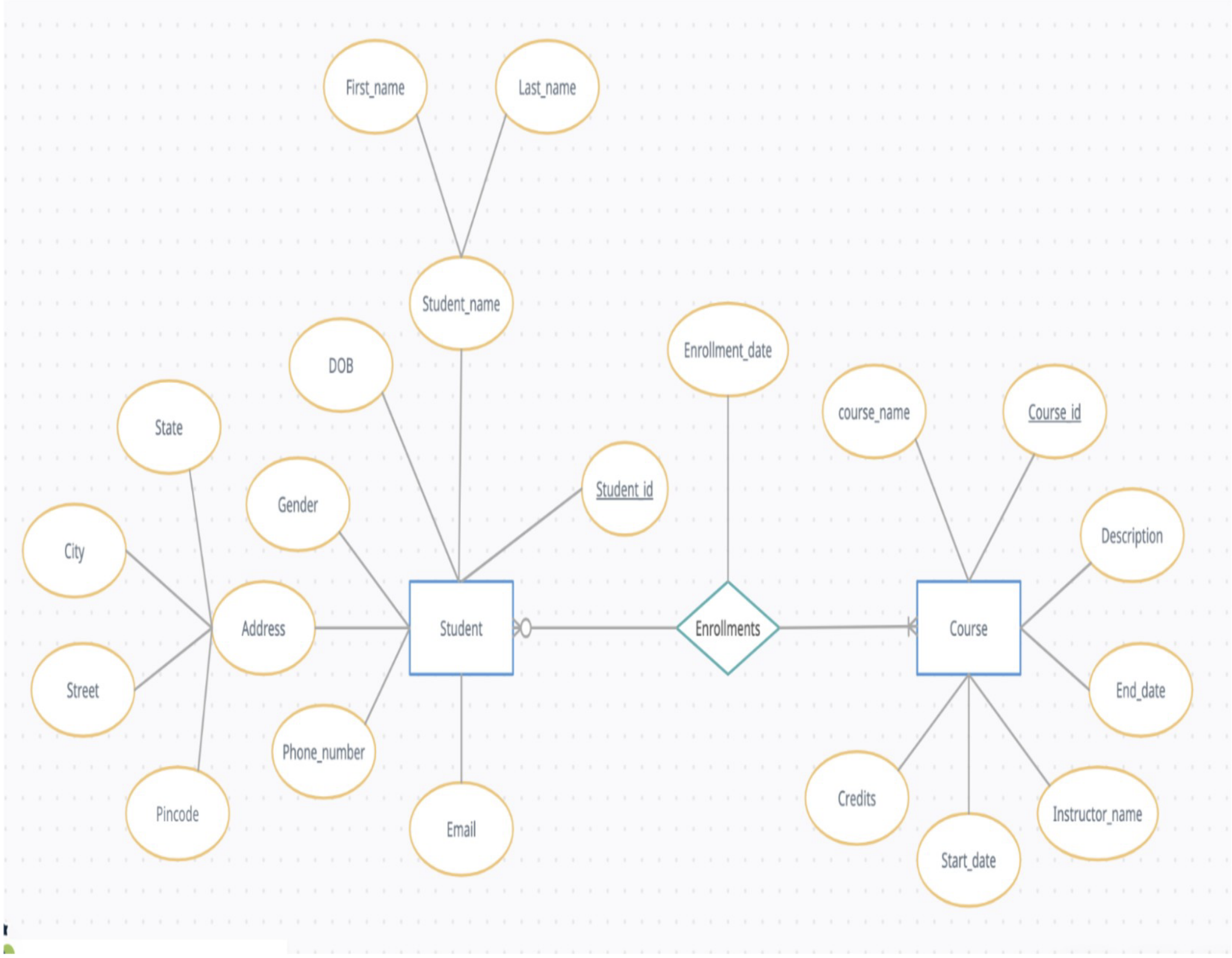
Student: student ID

Course: course ID

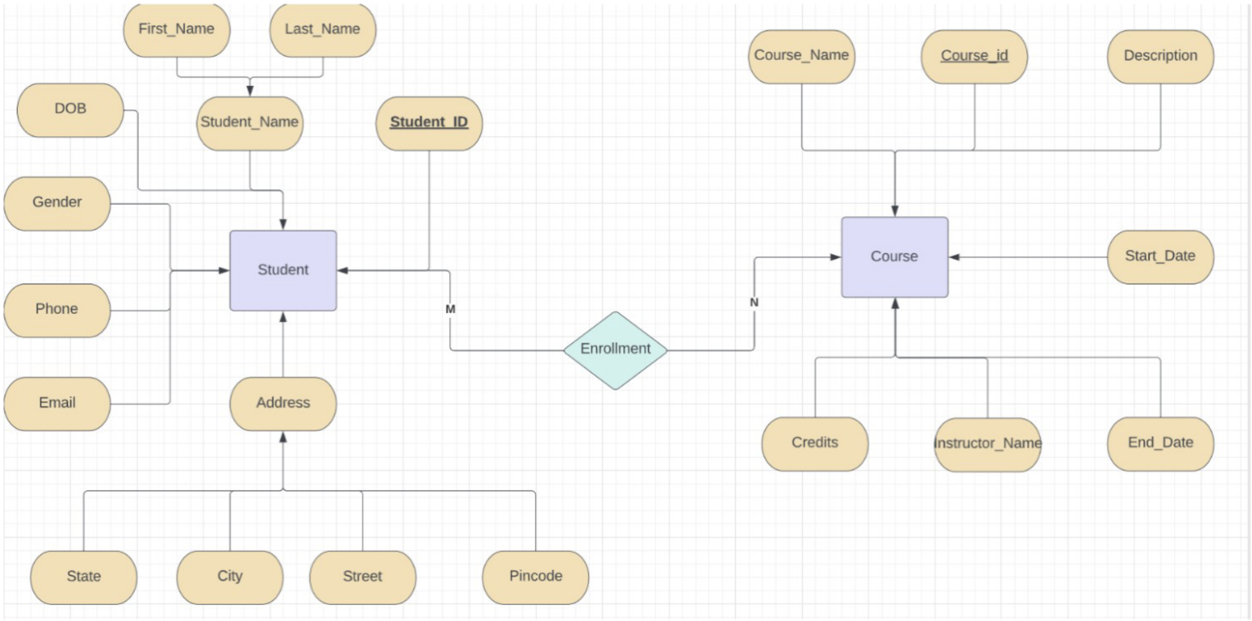
Foreign Keys (Referential Integrity)

Enrollments(Intersection Table between Students and Courses):
Composite Key (student ID, course ID)

Purushottam’ Solution:



Atul' solution:



Difference:

- The Entities and number of entities are similar for all of us.
- The relationship between **student** and **course** (many to many) is the same for all of us.
- All the attributes of both the entities are also the same.
- But Atul's solution doesn't have an **enrollment_date** field for the intersection table.
- And Atul's solution uses **Chen notation** to present cardinality in ER diagrams whereas I have used **Crow's foot notation** to do the same.