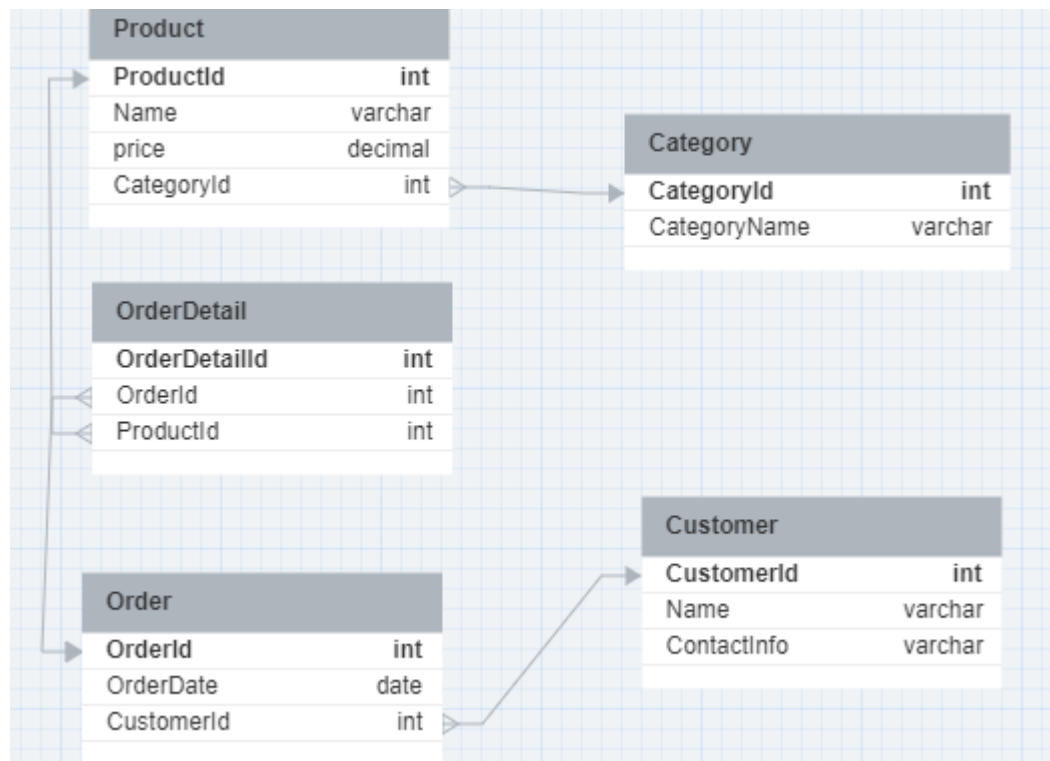| Student ID:	Name:

1. (5 points) You are tasked with designing a database for a small online store. The store sells products, and customers can place orders for these products. Each product has a name, price, and category, and each customer has a unique customer ID, name, and contact information. An order can contain multiple products, and each product in an order has a quantity.

   Design an Entity-Relationship (ER) diagram for the online store database based on the given scenario. Include entities, attributes, and relationships between entities. Be sure to indicate primary keys and any necessary cardinality constraints. You have 5 minutes to complete this task.

   Solution:

   Product
ProductId	int
Name	varchar
price	decimal
CategoryId	int

   Category
CategoryId	int
CategoryName	varchar

   OrderDetail
OrderDetailId	int
OrderId	int
ProductId	int

   Customer
CustomerId	int
Name	varchar
ContactInfo	varchar

   Order
OrderId	int
OrderDate	date
CustomerId	int

2. Consider the following table, "Employee Info" which stores information about employees in a company. The table represents an employee hierarchy within the organization, where employees are organized into departments, and each employee has a supervisor except for those at the top of the hierarchy.

EmployeeID	EmployeeName	Department	SupervisorID	SupervisorName	Salary
101	Alice	HR	201	Bob	60000
201	Bob	HR	301	Charlie	75000
301	Charlie	IT	401	David	80000
401	David	IT	NULL	NULL	90000
501	Eve	Sales	101	Alice	55000
601	Frank	Sales	201	Bob	60000

- EmployeeID: This is a unique identifier for each employee in the organization.
- EmployeeName: This field stores the name of each employee.
- Department: It indicates the department to which the employee belongs, such as HR, IT, or Sales.
- SupervisorID: This field references the EmployeeID of the supervisor or manager of each employee. It establishes a hierarchical relationship between employees and their supervisors.
- SupervisorName: This field stores the name of the supervisor of each employee. It redundantly stores information already present in the "EmployeeName" field for readability.
- Salary: This field contains the salary of each employee.

Your task is to normalize this table up to the Third Normal Form (3NF). To do this, follow these steps:

(a) (5 points) List all the functional dependencies present in the table.

> **Solution:**
> ```
> EmployeeId -> EmployeeName, Department, SupervisorId, SupervisorName, Salary
> SupervisorId -> SupervisorName
> ```

(b) (3 points) Identify any transitive dependencies.

> **Solution:**
> ```
> SupervisorId -> SupervisorName
> EmployeeId -> SupervisorId
> ```

(c) (7 points) Normalize the table up to the 3NF by creating new tables as needed.

> **Solution:**
> **1NF Schema**
> Employee(<u>EmployeeId</u>, EmployeeName, Department, SupervisorId, SupervisorName, Salary)
> All attributes are atomic
> **2NF Schema**
> Employee(<u>EmployeeId</u>, EmployeeName, Department, SupervisorId, SupervisorName, Salary)
> All attributes fully functional dependent on the Primary Key
> **3NF Schema**
> Employee(<u>EmployeeId</u>, EmployeeName, Department, SupervisorId, Salary)
> Supervisor(<u>SupervisorId</u>, SupervisorName)

Student ID:	Name:

1. You are designing a database for a small library. The library has books, each book is written by one or more authors, and each book has multiple copies available for borrowing. Each copy has a unique identification number. A library member can borrow one or more books at a time.

   Design an ER diagram diagram for the library database based on the given scenario. Include entities, attributes, and relationships between entities. Be sure to indicate primary keys and any necessary cardinality constraints. List all necessary assumptions. These assumptions should align with the stated requirements and not contradict them.

   Solution:
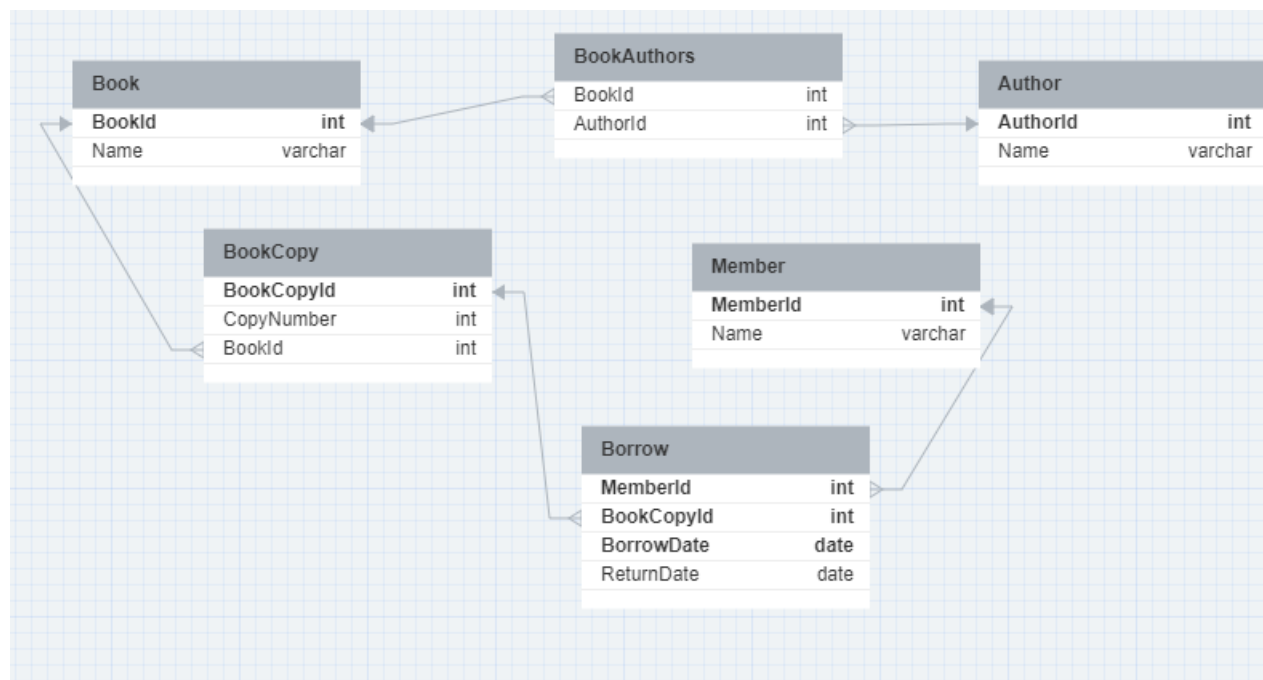
2. (5 points) This table allows the retail store to manage its product inventory efficiently. It tracks essential information such as product names, categories, suppliers, and prices, which can be used for various purposes, including inventory management, pricing, and supplier relations.

- ProductID: A unique identifier for each product.
- ProductName: The name of the product.
- Category: The categories to which the product belongs (e.g., Electronics, Clothing, Home Decor).
- SupplierID: A unique identifier for each supplier.
- Price: The price of the product.
- SupplierCountry: The country supplier belongs to.

ProductID	Product Name	Category	SupplierID	Price	SupplierCountry
101	A	Electronics, Gadgets	201	29.99	USA
102	B	Clothing, Fashion	202	19.99	Canada
103	C	Electronics, Gadgets	201	39.99	USA
104	D	Home Decor, Lifestyle	203	49.99	China
105	E	Clothing, Fashion	202	29.99	Canada
106	F	Home Decor, Lifestyle	203	59.99	China
107	G	Electronics, Gadgets	204	79.99	Japan
108	H	Clothing, Fashion	202	24.99	Canada
109	I	Home Decor, Lifestyle	203	69.99	China
110	J	Electronics, Gadgets	204	89.99	Japan

Your task is to normalize this table up to the Third Normal Form (3NF). To do this, follow these steps:

(a) (5 points) List all the functional dependencies present in the table.

> **Solution:**
> ProductId, Category -> ProductName, SupplierId, Price, SupplierCountry
> ProductId -> ProductName, SupplierId, Price, SupplierCountry

(b) (3 points) Identify any transitive dependencies.

> **Solution:**
> SuppliedId -> SupplierCountry
> ProductId -> SupplierId

(c) (7 points) Normalize the table up to the 3NF by creating new tables as needed.

> **Solution: 1NF Schema**
> Product (<u>ProductId</u>, <u>Category</u>, ProductName, SupplierId, Price, SupplierCountry)
>
> **2NF Schema**
> Product (<u>ProductId</u>, ProductName, SupplierId, Price, SupplierCountry)
> ProductCategory (<u>ProductId</u>, <u>Category</u>)
> **3NF Schema**
> Product (<u>ProductId</u>, ProductName, SupplierId, Price)
> ProductCategory (<u>ProductId</u>, <u>Category</u>)
> Supplier (<u>SupplierId</u>, SupplierCountry)