**Project Scenario: Library Management System**

**Scenario Description**

You have been tasked with designing a Library Management System

for a public library. The system should manage the borrowing and

returning of books by members, track the availability of books, and

keep records of all transactions. The goal is to create a logical model

diagram that captures the essential entities and relationships within the

system.

**Requirements**

1. **Books**:

o Each book has a unique identifier (Book ID), title,

author, publisher, and publication year.

o Books can have multiple copies, each with a unique

Copy ID.

2. **Members**:

o Each member has a unique Member ID, name, address,

phone number, and email.

o Members can borrow multiple books, but there is a limit

to the number of books a member can borrow at one

time.

3. **Borrowing Transactions**:

o Each borrowing transaction has a unique Transaction

ID, Member ID, Copy ID, borrow date, and due date.

o A member can borrow multiple books in a single

transaction, and each book borrowed will have a

separate entry.

4. **Returning Transactions**:

o Each returning transaction has a unique Transaction ID,

Member ID, Copy ID, and return date.

o Books returned late incur a fine based on the number of

late days.

10

CO1

BL65. **Fines**:

o Fines have a unique Fine ID, Transaction ID, amount,

and payment status.

o The fine amount is calculated based on the number of

late days and a fixed daily fine rate.

**Entities and Attributes**

1. **Book**

o Book ID (Primary Key)

o Title

o Author

o Publisher

o Publication Year

2. **Copy**

o Copy ID (Primary Key)

o Book ID (Foreign Key)

3. **Member**

o Member ID (Primary Key)

o Name

o Address

o Phone Number

o Email

4. **BorrowingTransaction**

o Transaction ID (Primary Key)

o Member ID (Foreign Key)

o Copy ID (Foreign Key)

o Borrow Date

o Due Date

5. **ReturningTransaction**

o Transaction ID (Primary Key)

o Member ID (Foreign Key)

o Copy ID (Foreign Key)

o Return Date

6. **Fine**

o Fine ID (Primary Key)

o Transaction ID (Foreign Key)

o Amount

o Payment Status

**Relationships**

1. **A Book can have multiple Copies**.

2. **A Member can have multiple BorrowingTransactions**.

3. **A BorrowingTransaction involves one Member and one**

**Copy**.

4. **A ReturningTransaction involves one Member and one**

**Copy**.

5. **A Fine is associated with one BorrowingTransaction**.

**Logical Model Diagram**You can now create the logical model diagram based on the entities

and relationships described above. Use an Entity-Relationship (ER)

diagram to visually represent the model. The diagram should include:

• Entities (represented as rectangles)

• Attributes (represented as ovals)

• Primary keys (underlined attributes)

• Relationships (represented as diamonds or lines connecting

entities)

• Foreign keys (represented as lines connecting related entities)

ANSWER:

**Logical Model Diagram for Library Management System**Here is the Entity-Relationship (ER) diagram for the Library Management System:

text

+-------------+ +-------------+

| Book | | Copy |

+-------------+ +-------------+

| BookID (PK) | | CopyID (PK) |

| Title | | BookID (FK)|

| Author | +-------------+

| Publisher |

| PublicationYear|

+-------------+

+-------------+ +-------------+ +-------------+

| Member | |BorrowingTransaction| Fine |

+-------------+ +-------------+ +-------------+

| MemberID (PK)| |TransactionID (PK)| FineID (PK)|

| Name | | MemberID (FK) |TransactionID (FK)|

| Address | | CopyID (FK) | Amount |

| PhoneNumber | | BorrowDate | PaymentStatus|

| Email | | DueDate +-------------+

+-------------+ +-------------+

+-------------+

|ReturningTransaction|

+-------------+

|TransactionID (PK)|

| MemberID (FK) |

| CopyID (FK) |

| ReturnDate |

+-------------+

**Explanation of the ER Diagram**

1. **Books**: The Book entity represents the books in the library. Each book has a unique BookID (primary key), Title, Author, Publisher, and PublicationYear.
2. **Copies**: The Copy entity represents the individual copies of each book. Each copy has a unique CopyID (primary key) and is associated with a BookID (foreign key).
3. **Members**: The Member entity represents the library members. Each member has a unique MemberID (primary key), Name, Address, PhoneNumber, and Email.
4. **Borrowing Transactions**: The BorrowingTransaction entity represents the transactions when members borrow books. Each transaction has a unique TransactionID (primary key), is associated with a MemberID (foreign key) and a CopyID (foreign key), and has a BorrowDate and DueDate.
5. **Returning Transactions**: The ReturningTransaction entity represents the transactions when members return books. Each transaction has a unique TransactionID (primary key), is associated with a MemberID (foreign key) and a CopyID (foreign key), and has a ReturnDate.
6. **Fines**: The Fine entity represents the fines incurred by members for late returns. Each fine has a unique FineID (primary key), is associated with a TransactionID (foreign key), and has an Amount and PaymentStatus.

The relationships between the entities are as follows:

* A Book can have multiple Copies (one-to-many relationship).
* A Member can have multiple BorrowingTransactions (one-to-many relationship).
* A BorrowingTransaction involves one Member and one Copy (many-to-many relationship).
* A ReturningTransaction involves one Member and one Copy (many-to-many relationship).
* A Fine is associated with one BorrowingTransaction (one-to-one relationship).

This ER diagram captures the essential entities and relationships required for the Library Management System, allowing for the efficient management of books, members, borrowing and returning transactions, and fines.

**Project Scenario: Oracle Baseball League (OBL)**

You are a small consulting company specializing in database development.

You have just been awarded the contract to develop a data model for a

database application system for a small retail store called Oracle Baseball

League (OBL).

The Oracle Baseball League store serves the entire surrounding community

selling baseball kit. The OBL has two types of customer, there are individuals

who purchase items like balls, cleats, gloves, shirts, screen printed t-shirts,

and shorts. Additionally customers can represent a team when they purchase

uniforms and equipment on behalf of the team.

Teams and individual customers are free to purchase any item from the

inventory list, but teams get a discount on the list price depending on the

number of players. When a customer places an order we record the order

items for that order in our database.

OBL has a team of three sales representatives that officially only call on

teams but have been known to handle individual customer complaints.

ANSWER:

SQL

Database language

More

Definition

A domain-specific language used in programming for managing data in a relational database management system (RDBMS) or for stream processing in a relational data stream management system (RDSMS).

Initial Release

1986

Paradigm

Declarative

**Entity-Relationship (ER) Diagram for Oracle Baseball League (OBL)**Here is the ER diagram for the Oracle Baseball League (OBL) database:

text

+-------------+ +-------------+ +-------------+

| Customer | | Order | | OrderItem |

+-------------+ +-------------+ +-------------+

| CustomerID (PK) | | OrderID (PK)| | OrderItemID (PK)|

| CustomerType | | CustomerID (FK)| | OrderID (FK) |

| Name | | OrderDate | | ProductID (FK)|

| Address | | Discount | | Quantity |

| PhoneNumber | +-------------+ +-------------+

+-------------+

+-------------+ +-------------+

| Product | | SalesRep |

+-------------+ +-------------+

| ProductID (PK)| | SalesRepID (PK)|

| ProductName| | SalesRepName |

| ProductType| | PhoneNumber |

| UnitPrice | +-------------+

+-------------+

**Explanation of the ER Diagram**

1. **Customer**: The Customer entity represents the customers of the OBL store. Each customer has a unique CustomerID (primary key), a CustomerType (individual or team), Name, Address, and PhoneNumber.
2. **Order**: The Order entity represents the orders placed by customers. Each order has a unique OrderID (primary key), is associated with a CustomerID (foreign key), has an OrderDate, and may have a Discount applied.
3. **OrderItem**: The OrderItem entity represents the individual items within an order. Each order item has a unique OrderItemID (primary key), is associated with an OrderID (foreign key) and a ProductID (foreign key), and has a Quantity.
4. **Product**: The Product entity represents the products sold by the OBL store. Each product has a unique ProductID (primary key), a ProductName, ProductType, and UnitPrice.
5. **SalesRep**: The SalesRep entity represents the sales representatives of the OBL store. Each sales representative has a unique SalesRepID (primary key), a SalesRepName, and a PhoneNumber.

The relationships between the entities are as follows:

* A Customer can place multiple Orders (one-to-many relationship).
* An Order can have multiple OrderItems (one-to-many relationship).
* An OrderItem is associated with one Product (many-to-one relationship).
* An Order is associated with one Customer (many-to-one relationship).
* A SalesRep can be associated with multiple Customers (one-to-many relationship).

This ER diagram captures the essential entities and relationships required for the Oracle Baseball League (OBL) database, allowing for the efficient management of customers, orders, order items, products, and sales representatives.