1.)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 BL6 5. 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:

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

| Book |

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

| Book ID |

| Title |

| Author |

| Publisher|

|Pub. Year |

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

|

|

|

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

| Copy |

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

| Copy ID |

| Book ID |

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

|

|

|

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

| Member |

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

| Member ID|

| Name |

| Address |

|Phone No. |

| Email |

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

|

|

|

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

|Borrowing |

|Transaction|

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

|Transaction ID|

| Member ID |

| Copy ID |

| Borrow Date |

| Due Date |

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

|

|

|

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

|Returning |

|Transaction|

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

|Transaction ID|

| Member ID |

| Copy ID |

| Return Date |

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

|

|

|

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

| Fine |

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

| Fine ID |

|Transaction ID|

| Amount |

|Payment Status|

|  |
| --- |
|  |

Relationships

1. **One-to-Many Relationship**: A Book can have multiple Copies, but each Copy is associated with only one Book.
2. **One-to-Many Relationship**: A Member can have multiple Borrowing Transactions, but each Borrowing Transaction is associated with only one Member.
3. **One-to-One Relationship**: A Borrowing Transaction is associated with one Copy, and a Copy is associated with one Borrowing Transaction.
4. **One-to-One Relationship**: A Returning Transaction is associated with one Copy, and a Copy is associated with one Returning Transaction.
5. **One-to-One Relationship**: A Fine is associated with one Borrowing Transaction, and a Borrowing Transaction is associated with one Fine.

The logical model diagram captures the essential entities and relationships within the Library Management System, allowing for efficient management of book borrowing, returning, and fine tracking.

2.) 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:

Based on the requirements for Oracle Baseball League (OBL), we can outline the initial steps and considerations for developing a data model for their database application system. Here’s a structured approach:

### Entities Identified:

1. **Customer**
   * Attributes: Customer ID, Name, Type (Individual or Team)
2. **Team**
   * Attributes: Team ID, Team Name, Contact Person, Contact Phone, Discount Percentage
3. **Individual Customer**
   * Inherits from Customer: Individual-specific attributes
4. **Sales Representative**
   * Attributes: Rep ID, Name
5. **Item**
   * Attributes: Item ID, Description, Price
6. **Order**
   * Attributes: Order ID, Order Date, Customer ID (foreign key), Total Amount
7. **Order Item**
   * Attributes: Order Item ID, Order ID (foreign key), Item ID (foreign key), Quantity, Unit Price

### Relationships:

* **Customer to Team**: Many-to-Many (A customer can be associated with multiple teams, and a team can have multiple customers)
* **Customer to Order**: One-to-Many (A customer can place multiple orders)
* **Order to Order Item**: One-to-Many (An order can have multiple order items)
* **Item to Order Item**: One-to-Many (An item can appear in multiple order items)

### Business Rules and Logic:

* **Discount Calculation**: Teams receive a discount based on the number of players.
* **Sales Representatives**: Can handle orders from teams and occasionally handle individual customer complaints.

### Proposed Schema:

plaintext

Copy code

Customer (CustomerID, Name, Type, TeamID, ...)

Team (TeamID, TeamName, ContactPerson, ContactPhone, DiscountPercentage, ...)

IndividualCustomer (CustomerID, AdditionalAttributes...)

SalesRepresentative (RepID, Name, ...)

Item (ItemID, Description, Price, ...)

Order (OrderID, OrderDate, CustomerID, TotalAmount, ...)

OrderItem (OrderItemID, OrderID, ItemID, Quantity, UnitPrice, ...)

### Considerations:

* **Normalization**: Ensure the database design is normalized to minimize redundancy and anomalies.
* **Indexes**: Use indexes on frequently queried columns (e.g., CustomerID, OrderDate) for performance.
* **Data Integrity**: Enforce referential integrity constraints using foreign keys.
* **Security**: Implement appropriate access controls to protect sensitive data.

### Next Steps:

1. **Detailed Requirements Gathering**: Further refine requirements with OBL stakeholders.
2. **Schema Refinement**: Define data types, constraints, and relationships more precisely.
3. **Implementation Plan**: Plan the database creation, including scripts for schema creation and initial data population.
4. **Testing and Validation**: Test the database schema thoroughly against use cases and edge cases.
5. **Deployment and Maintenance**: Deploy the database schema and establish a maintenance plan for updates and backups.

By following these steps and considerations, the data model for Oracle Baseball League’s database application system can effectively support their operations and business requirements.

Top of Form

Bottom of Form