

Student's Name

Professor's Name

Course

Date

Question-Answer Assignment

Task 1: Document and formalize your final design

Identify the entities and attributes. List the entity and attribute name for each entity.

Identify the relationships among the entities. List the relationship type in terms of cardinality and specify the business rules (e.g. 1:M between Library and Branch: a library has one or many branches; a branch belongs to one and only one library).

For each entity, identify the primary key and foreign key if applicable. If you use a surrogate key for PK, also specify a unique business key (natural key). For FK, also specify the parent entity and the matching attribute in parent entity.

Create ERD using crow's foot notation via an ER Assistant or Visio. The ERD should incorporate all items mentioned in 2.1, 2.2, and 2.3 (NOTE: 2.1, 2.2, and 2.3 still need to be answered separately in narrative format).

Solution:

Library: holds info about a library

Branch: info about each branch and concern library

Catalog: holds the details of availability and other general details about items

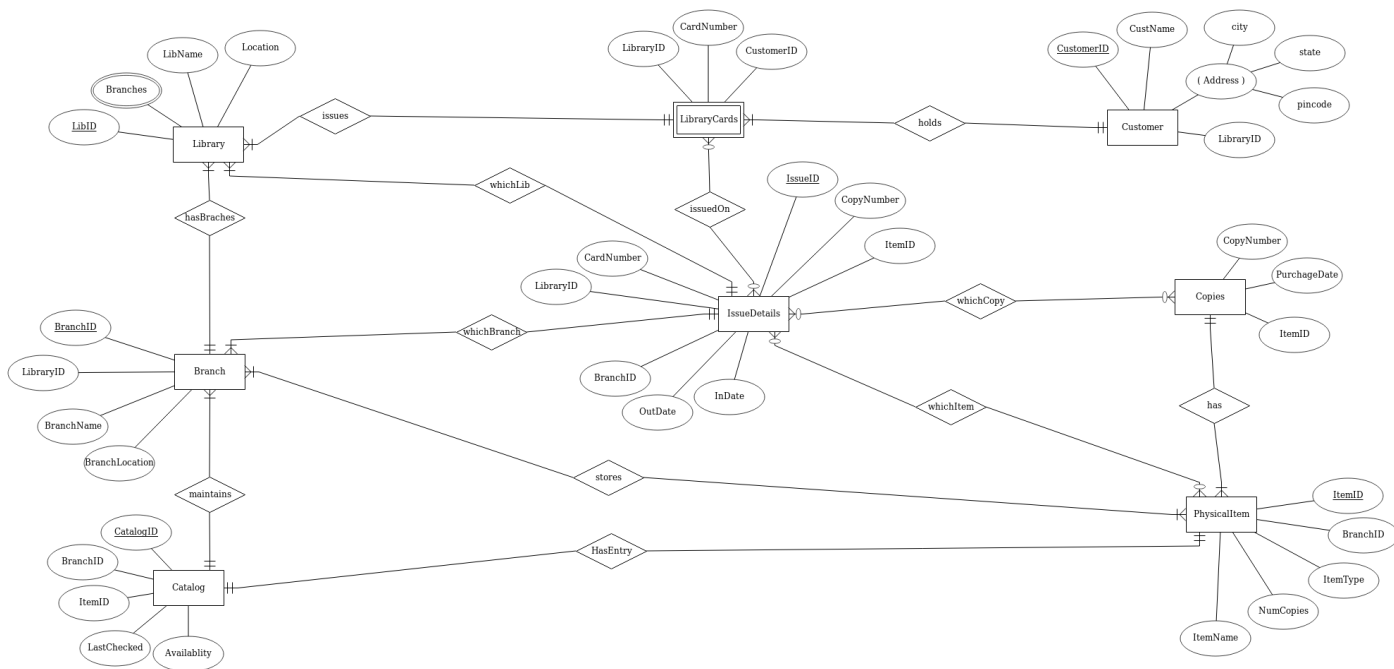
IssueDetails: hold info about check in/out of items by customers

LibraryCard: holds a library card number and the concerned library and customer

Customer: basic details about a customer

PhysicalItem: general details about physical items

Copies: holds details about copies of a particular item



The underlined fields are the primary keys (Zoom to view)

Task 2: Implement the database

Write SQL DDL statements to create tables in Oracle to implement all entities and attributes with the correct data type and data size, primary keys, foreign keys, and constraints such as NOT NULL. Create indexes on all foreign key and business key columns. Write SQL DML statements to insert all data shown in the sample spreadsheet into the new database.

Solution:

```
DROP TABLE IF EXISTS Library;
```

```
CREATE TABLE Library
```

```
(LibID varchar NOT NULL PRIMARY KEY,  
Branches varchar NULL,  
LibName varchar NOT NULL,  
Location varchar NOT NULL);
```

```
DROP TABLE IF EXISTS Branch;
```

```
CREATE TABLE Branch
```

```
(BranchID varchar NOT NULL PRIMARY KEY,  
BranchName varchar NULL,  
LibraryID varchar NOT NULL FOREIGN KEY REFERENCES Library(LibID),  
BranchLocation varchar NOT NULL,  
);
```

```
CREATE INDEX idx_LibraryID
```

```
ON Branch (LibraryID);
```

```
DROP TABLE IF EXISTS Catalog;
```

```
CREATE TABLE Catalog
```

```
(CatalogID varchar NOT NULL PRIMARY KEY,  
BranchID varchar NOT NULL FOREIGN KEY REFERENCES Branch(BranchID),  
ItemID varchar NOT NULL FOREIGN KEY REFERENCES PhysicalItem(ItemID),  
LastChecked varchar NOT NULL,  
Availability varchar NOT NULL,
```

);

DROP TABLE IF EXISTS LibraryCards;

CREATE TABLE LibraryCards

(

LibraryID varchar NOT NULL FOREIGN KEY REFERENCES Library(LibID),

CardNumber varchar NOT NULL,

CustomerID varchar NOT NULL FOREIGN KEY REFERENCES Customer(CustomerID),

);

DROP TABLE IF EXISTS IssueDetails;

CREATE TABLE IssueDetails

(IssueID varchar NOT NULL PRIMARY KEY,

BranchID varchar NOT NULL FOREIGN KEY REFERENCES Branch(BranchID),

ItemID varchar NOT NULL FOREIGN KEY REFERENCES PhysicalItem(ItemID),

CopyNumber varchar NOT NULL,

CardNumber varchar NOT NULL,

LibraryID varchar NOT NULL FOREIGN KEY REFERENCES Library(LibID),

OutDate varchar NOT NULL,

InDate varchar NOT NULL,

);

DROP TABLE IF EXISTS Customer;

```
CREATE TABLE Customer
(CustomerID varchar NOT NULL PRIMARY KEY,
CustName varchar NULL,
City varchar NOT NULL,
Address varchar NOT NULL,
LibraryID varchar NOT NULL FOREIGN KEY REFERENCES Library(LibID),
);
```

```
DROP TABLE IF EXISTS Copies
```

```
CREATE TABLE CopyNumber
```

```
(
CopyNumber varchar NOT NULL,
PurchaseDate varchar NOT NULL,
ItemID varchar NOT NULL FOREIGN KEY REFERENCES PhysicalItem(ItemID),
);
```

```
DROP TABLE IF EXISTS PhysicalItem;
```

```
CREATE TABLE PhysicalItem
```

```
(ItemID varchar NOT NULL PRIMARY KEY,
BranchID varchar NOT NULL FOREIGN KEY REFERENCES Branch(BranchID),
ItemType varchar NOT NULL,
NumCopies varchar NOT NULL,
ItemName varchar NOT NULL,
);
```

Task 3: Use the database and create a report

Write SQL queries against each table so as to display the number of records in each table. Write a minimum of 10 queries to query the database. Add a comment to each query to describe its purpose. Write an SQL statement to create a database view so that a report can be generated by querying the view to create the same set of data as shown in the original spreadsheet.

Solution:

```
SELECT * FROM Library; // display all columns from table library
SELECT * FROM Branch; // display all columns from table Branch
SELECT * FROM Catalog; // display all columns from table Catalog
SELECT * FROM LibraryCards; // display all columns from table LibraryCards
SELECT * FROM IssueDetails; // display all columns from table IssueDetails
SELECT * FROM Customer; // display all columns from table Customer
SELECT * FROM Copies; // display all columns from table Copies
SELECT * FROM PhysicalItem; // display all columns from table PhysicalItem
SELECT * FROM Branch; // display all columns from table Branch
SELECT * FROM Catalog; // display all columns from table Catalog

CREATE VIEW [Customer_view] AS
SELECT CustName, city, Address
FROM Customer
```