INFO6001 2024 T1

Assignment 2

Project: Database design of SCS Resource Management

*Logical Database Design*

Contents

[Preface 4](#_Toc162013197)

[Part 1: Reflection on Assignment 1 5](#_Toc162013198)

[Part 2: Requirements 5](#_Toc162013199)

[Data Requirements 5](#_Toc162013200)

[Transaction Requirements 6](#_Toc162013201)

[INSERT QUERY: 6](#_Toc162013202)

[UPDATE QUERY: 6](#_Toc162013203)

[DELETE QUERY: 6](#_Toc162013204)

[SOME OTHER QUERIES: 6](#_Toc162013205)

[Query 1: 6](#_Toc162013206)

[Query 2: 6](#_Toc162013207)

[Query 3: 7](#_Toc162013208)

[Query 4: 7](#_Toc162013209)

[Query 5: 7](#_Toc162013210)

[Business Rules 7](#_Toc162013211)

[Part 2: EER Model 9](#_Toc162013212)

[EER Model 9](#_Toc162013213)

[Data Dictionary 10](#_Toc162013214)

[Entity 10](#_Toc162013215)

[Relationships 11](#_Toc162013216)

[Attributes: 13](#_Toc162013217)

[Part 4: Mapping the EER to Relational Model 17](#_Toc162013218)

[Part 5: Normalising the Scheme up to BCNF 22](#_Toc162013219)

[Step 1: Identify Dependencies 22](#_Toc162013220)

[Acquisition Table: 22](#_Toc162013221)

[Member Table: 22](#_Toc162013222)

[Staff Table: 22](#_Toc162013223)

[Student Table: 22](#_Toc162013224)

[Loan Table: 22](#_Toc162013225)

[Reservation Table: 22](#_Toc162013226)

[CourseOffering Table: 22](#_Toc162013227)

[Privilege Table: 23](#_Toc162013228)

[Category Table: 23](#_Toc162013229)

[Resource Table: 23](#_Toc162013230)

[Movable Table: 23](#_Toc162013231)

[Immovable Table: 23](#_Toc162013232)

[Summary 23](#_Toc162013233)

# Preface

This is the requirements analysis, conceptual design and logical database design for the implementation of the database of *SCS Resource Management*.

The requirements analysis includes the data requirements, transaction requirements and business rules. This will ensure that all the data required to be stored is identified, can be manipulated, and is managed according to the store’s business policies.

The conceptual design includes an extended entity relationship diagram in UML which describes the required entities, their attributes, and their relationships. It is further clarified with a data dictionary for the entities, attributes, and also the relationships. This conceptual model was developed from the requirements analysis.

From the conceptual model, a relational model is established and presented in the form of DBML, which is then normalized to BCNF, with a discussion of the steps to produce the logical design of the database.

In this report, we have discussed the EER solution and the submitted EER model. We have shown and told briefly the difference between the provided solution EER and the EER submitted.

We will then present data requirements, transaction requirements, and business rules that have been used in this database. Other than this we have mapped the EER model into the relational model to create a database according to the requirement.

After presenting the database with a data dictionary, we will show all the entities with the relationships and attributes that have been used in it. In conclusion we will show the normalization in the database.

# Part 1: Reflection on Assignment 1

In this assignment we have some points that are not similar to the previous one. As we know that EER model has some special things that will make more complex database and relations. The EER use top-down (means specialized) approach and it also use bottom-top approach (means generalized) approach. Other than this the EER model uses superclass and subclass. In the submitted EER both side generalization is used. The approach of the bottom top is used but in the provided solution the specialized and generalized are used.

In the assignment all the entities are same but some attributes are missing and change in the submitted EER model. Although everything other than this is the same.

The provided EER is more specific and generalized in terms of the data and design. The EER model has acquisition , member , course offered, loan , privilege, category and resource.

# Part 2: Requirements

## Data Requirements

* **Loan Service**

Loan

Loan describes loans that members have made. It is created when a loan is made by a member.

Each member can loan movable resources, the number of resources the member can loan is dependent on the privileges. As staff does not have privilege, the amount of resource they can loan are not restricted. Information stored include (the resource loaned, the member lending it), date and time loaned, due date and time and date and time returned. All loans have a unique loan id. The due date will depend on the duration allowed by the category.

Reservation

Reservations are the reservations that members make. In the reservation, members reserve it for themselves and the resources. Reservations will be categorized by category. Reservations has a unique reservation id. That will be used to keep the data unique in all the rows. However, the time and the date of the reservation also store inside the database.

Course Offering

Course will be offered to the students. The student can enroll in many offered courses. Course has some privilege. The course has course id. That keeps the data unique in the database. The course contains some name. Course offered also take information about the semester in which student is enrolling and the year. It also keep the track of the start date of the semester and the end date of the semester or course.

## Transaction Requirements

Data Manipulation

### INSERT QUERY:

INSERT INTO Acquisition (resourceName, description, make, model, year, urgency, status, fundCode, vendorCode, price, notes, member\_id)

VALUES ('Printer', 'High-speed laser printer', 'HP', 'LaserJet Pro', 2022, 'High', 'Available', 'FC123', 'VC456', 599.99, 'Office use', 1);

### UPDATE QUERY:

UPDATE Acquisition

SET status = 'In Progress'

WHERE acqID = 1;

### DELETE QUERY:

DELETE FROM Member

WHERE member\_id = 2;

## SOME OTHER QUERIES:

### Query 1:

Select \* from loan;

### Query 2:

SELECT \*

FROM Member

INNER JOIN Acquisition ON Member.member\_id = Acquisition.member\_id;

### Query 3:

SELECT Loan.\*, Member.name AS member\_name, Resource.description AS resource\_description

FROM Loan

INNER JOIN Member ON Loan.member\_id = Member.member\_id

INNER JOIN Movable ON Loan.loan\_id = Movable.loan\_id

INNER JOIN Resource ON Movable.resource\_id = Resource.resource\_id;

### Query 4:

SELECT \*

FROM Acquisition

WHERE status = 'In Progress'

ORDER BY year DESC;

### Query 5:

SELECT Loan.\*, Member.name AS member\_name, Resource.description AS resource\_description

FROM Loan

INNER JOIN Member ON Loan.member\_id = Member.member\_id

INNER JOIN Movable ON Loan.loan\_id = Movable.loan\_id

INNER JOIN Resource ON Movable.resource\_id = Resource.resource\_id

WHERE Loan.datetime\_borrowed BETWEEN '2024-01-01' AND '2024-03-31'

ORDER BY Loan.datetime\_borrowed DESC;

## Business Rules

* Student member set to ‘Disabled’ if the current date is later than end date of his course offering.
* Each acquisition should have a unique combination of resource name and member.
* The status of an acquisition should be one of a predefined set of values.
* The price of an acquisition should be non-negative.
* The year of acquisition should be within a valid range

# Part 2: EER Model

## EER Model



Data Dictionary

## Entity

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity Name** | **Description** | **Aliases** | **Occurrence** |
| Loan | describing loans that members have made | Resource borrowed | When a loan is made by a member |

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity Name** | **Description** | **Aliases** | **Occurrence** |
| Reservations | describing reservations that members have made | Resource borrowed | When a reservation is made by a member |

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity Name** | **Description** | **Aliases** | **Occurrence** |
| Member | Members are important for acquisition | The person made reservations and loans | Members are the entities that will relate with each other to make it meaningful. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity Name** | **Description** | **Aliases** | **Occurrence** |
| Course Offering | describing enroll courses that members have made | Enroll by students | Occur when students enroll into the courses. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity Name** | **Description** | **Aliases** | **Occurrence** |
| Privilege | describing privilege of the items | Resource borrowed | When a student enroll into the course. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity Name** | **Description** | **Aliases** | **Occurrence** |
| Resources | describing resources that have been reserved by members | Resource borrowed | When a resource is reserved by a member |

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity Name** | **Description** | **Aliases** | **Occurrence** |
| Category | Category is the thing that save about the privilege and the for the time resource is reserved | Resource borrowed | Resource category reserve |

## Relationships

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity Name | Multiplicity | Relationship | Multiplicity | Entity Name |
| Loan | 0..\* | is of | 1..1 | Member |
| 0..\* | to | 1..1 | Movable |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity Name | Multiplicity | Relationship | Multiplicity | Entity Name |
| Member | 0..\* | is of | 1..1 | loan |
| 0..\* | to | 1..1 | reservations |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity Name | Multiplicity | Relationship | Multiplicity | Entity Name |
| Reservations | 0..\* | is of | 1..1 | Member |
| 0..\* | to | 1..1 | Resource |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity Name | Multiplicity | Relationship | Multiplicity | Entity Name |
| Resource | 0..\* | is of | 1..1 | Reservations |
| 0..\* | to | 1..1 | category |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity Name | Multiplicity | Relationship | Multiplicity | Entity Name |
| Category | 0..\* | is of | 1..1 | Resource |
| 0..\* | Is of | 1..1 | Privilige |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity Name | Multiplicity | Relationship | Multiplicity | Entity Name |
| Privilege | 0..\* | is of | 1..1 | Course Offering |
| 0..\* | to | 1..1 | Category |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity Name | Multiplicity | Relationship | Multiplicity | Entity Name |
| Course Offering | 1..\* | is of | 0..\* | Student |
| 0..\* | to | 0..\* | privilege |

## Attributes:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Data Type & Length** | **Nulls** | **Multi-valued** | **Derived** | **Default** |
| Loan | LoanID |  |  |  |  |  |  |
|  | dateOfLoan | The date the resource is loaned out | date | FALSE | FALSE | FALSE | current date |
|  | dateOfDue | The date the resource is due for return | date | FALSE | FALSE | FALSE |  |
|  | dateOfReturn | The date the resource is returned | date | TRUE | FALSE | TRUE |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Data Type & Length** | **Nulls** | **Multi-valued** | **Derived** | **Default** |
| Resource | Resource ID |  |  |  |  |  |  |
|  | description | Description of the source | Varchar(100) | FALSE | FALSE | FALSE | current date |
|  | status | Status of the resource | Varchar(100) | FALSE | FALSE | FALSE |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Data Type & Length** | **Nulls** | **Multi-valued** | **Derived** | **Default** |
| Immovable | Resource ID |  | int |  |  |  |  |
|  | Capacity | Capacity of the room | Int | FALSE | FALSE | FALSE |  |
|  | Room | Room no | VARCHAR(50) | FALSE | FALSE | FALSE |  |
|  | Building | Building | Varchar | TRUE | FALSE | TRUE |  |
| … | Campus | And the campus in which these resources are | Varchar |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Data Type & Length** | **Nulls** | **Multi-valued** | **Derived** | **Default** |
| MOvable | Resource ID |  | int |  |  |  |  |
|  | manufacturer | The date the resource is loaned out | date | FALSE | FALSE | FALSE |  |
|  | model | Model of the resources | Varchar(100) | FALSE | FALSE | FALSE |  |
|  | year | Year it is built | date | TRUE | FALSE | TRUE |  |
|  | AssetValue |  | Varchar(50) |  |  |  |  |
|  | Building ID |  | Varchar(50) |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Data Type & Length** | **Nulls** | **Multi-valued** | **Derived** | **Default** |
| Category | Code |  | int |  |  |  |  |
|  | name | Name of the category | Varchar(100) | FALSE | FALSE | FALSE |  |
|  | description | Description of the category | Varchar(100) | FALSE | FALSE | FALSE |  |
|  | duration days | For days they have reserved | int | TRUE | FALSE | TRUE |  |
|  | duration Hours | For hours they have reserved | time |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Data Type & Length** | **Nulls** | **Multi-valued** | **Derived** | **Default** |
| Privilege | Privi ID | Primary key | INt |  |  |  |  |
|  | name | Name of the privilege | Varchar(100) | FALSE | FALSE | FALSE | current date |
|  | description | Description of the privilege | Varchar(100) | FALSE | FALSE | FALSE |  |
|  | Max Items | Max item of the privilege | int | TRUE | FALSE | TRUE |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Data Type & Length** | **Nulls** | **Multi-valued** | **Derived** | **Default** |
| Course Offering | Offer ID | Primary key | int |  |  |  |  |
|  | cid | Course id | Int | FALSE | FALSE | FALSE | current date |
|  | course | Course name | Varchar (100) | FALSE | FALSE | FALSE |  |
|  | semester | Semester in which it is offered | Int | TRUE | FALSE | TRUE |  |
| … | year | Year in which it is offered | date |  |  |  |  |
|  | dateBegin | The day it is begin | date |  |  |  |  |
|  | dateEnd | The day it is end | date |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Data Type & Length** | **Nulls** | **Multi-valued** | **Derived** | **Default** |
| Reservations | Reservation ID |  | int |  |  |  |  |
|  | Date time reserved | The time made reservations | date | FALSE | FALSE | FALSE | current date |
|  | Date time due | Reservations till | date | FALSE | FALSE | FALSE |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Data Type & Length** | **Nulls** | **Multi-valued** | **Derived** | **Default** |
| Member | Member id | Primary key | int |  |  |  |  |
|  | name | The date the resource is loaned out | Varchar(100) | FALSE | FALSE | FALSE | current date |
|  | address | The date the resource is due for return | Varchar(100) | FALSE | FALSE | FALSE |  |
|  | phone | The date the resource is returned | date | TRUE | FALSE | TRUE |  |
|  | email | Email of the memeber | Varchar(100) |  |  |  |  |
|  | status | Status of that person | Varchar(50) |  |  |  |  |
|  | comment |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Data Type & Length** | **Nulls** | **Multi-valued** | **Derived** | **Default** |
| Acquisition | ACq ID | Primary key | int |  |  |  |  |
|  | Resource name | Name of the resource | Varchar(100) | FALSE | FALSE | FALSE | current date |
|  | description | Description of the acquisition | Varchar(100) | FALSE | FALSE | FALSE |  |
|  | make | Make of the aquisition | Varchar(50) | TRUE | FALSE | TRUE |  |
| … | model | Acquisition model | Varchar(50) |  |  |  |  |
|  | year | Year it is made | int |  |  |  |  |
|  | Urgency |  | Varchar(100) |  |  |  |  |
|  | Status | Status of the acuisition | Varchar(100) |  |  |  |  |
|  | fundCode | The code at which it is fund | Int |  |  |  |  |
|  | VendorCode |  | Int |  |  |  |  |
|  | price | The price of the item | Int |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Data Type & Length** | **Nulls** | **Multi-valued** | **Derived** | **Default** |
| STAFF | Member ID |  |  |  |  |  |  |
|  | Staff ID the staff id | The staff contain ID | int | FALSE | FALSE | FALSE | current date |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Data Type & Length** | **Nulls** | **Multi-valued** | **Derived** | **Default** |
| Student | Student ID |  |  |  |  |  |  |
|  | Point earned | The point student get | intnt | FALSE | FALSE | FALSE | current date |

# Part 4: Mapping the EER to Relational Model

Using the mapping rules, got the following relations for all entities in EER.

-- Create database

CREATE DATABASE IF NOT EXISTS ResourceManagement;

-- Use the database

USE ResourceManagement;

-- Create tables

CREATE TABLE Acquisition (

acqID INT PRIMARY KEY AUTO\_INCREMENT,

resourceName VARCHAR(255),

description TEXT,

make VARCHAR(255),

model VARCHAR(255),

year INT,

urgency VARCHAR(50),

status VARCHAR(50),

fundCode VARCHAR(50),

vendorCode VARCHAR(50),

price DECIMAL(10,2),

notes TEXT,

member\_id INT,

FOREIGN KEY (member\_id) REFERENCES Member(member\_id)

);

CREATE TABLE Member (

member\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(255),

address TEXT,

phone\_no VARCHAR(20),

email VARCHAR(100),

status VARCHAR(50),

comments TEXT

);

CREATE TABLE Staff (

member\_id INT,

staff\_id INT PRIMARY KEY AUTO\_INCREMENT,

FOREIGN KEY (member\_id) REFERENCES Member(member\_id)

);

CREATE TABLE Student (

student\_id INT PRIMARY KEY AUTO\_INCREMENT,

points\_earned INT

);

CREATE TABLE Loan (

loan\_id INT PRIMARY KEY AUTO\_INCREMENT,

datetime\_borrowed DATETIME,

datetime\_returned DATETIME,

datetime\_due DATETIME,

member\_id INT,

FOREIGN KEY (member\_id) REFERENCES Member(member\_id)

);

CREATE TABLE Reservation (

reservation\_id INT PRIMARY KEY AUTO\_INCREMENT,

datetime\_reserved DATETIME,

datetime\_due DATETIME

);

CREATE TABLE CourseOffering (

offer\_id INT PRIMARY KEY AUTO\_INCREMENT,

cid INT,

course VARCHAR(255),

semester VARCHAR(50),

year INT,

begin\_date DATE,

end\_date DATE

);

CREATE TABLE Privilege (

privil\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(255),

description TEXT,

maxItems INT

);

CREATE TABLE Category (

code VARCHAR(50) PRIMARY KEY,

name VARCHAR(255),

description TEXT,

duration\_days INT,

duration\_hours INT,

privi\_id INT,

FOREIGN KEY (privi\_id) REFERENCES Privilege(privil\_id)

);

CREATE TABLE Resource (

resource\_id INT PRIMARY KEY AUTO\_INCREMENT,

description TEXT,

status VARCHAR(50)

);

CREATE TABLE Movable (

name VARCHAR(255),

manufacturer VARCHAR(255),

model VARCHAR(255),

year INT,

asset\_value DECIMAL(10,2),

building\_ID INT,

loan\_id INT,

resource\_id INT,

FOREIGN KEY (loan\_id) REFERENCES Loan(loan\_id),

FOREIGN KEY (resource\_id) REFERENCES Resource(resource\_id)

);

CREATE TABLE Immovable (

capacity INT,

room VARCHAR(50),

building VARCHAR(255),

campus VARCHAR(255),

member\_id INT,

FOREIGN KEY (member\_id) REFERENCES Member(member\_id)

);

# Part 5: Normalising the Scheme up to BCNF

According to the definitions of 1NF, 2NF, 3NF and BCNF, it is identified that relations x1, x2, ..xn are all in BCNF, since all the attributes are atomic, and there exists only one function dependency in each table, and the left side of the FD is a PK.

### Step 1: Identify Dependencies

#### **Acquisition Table:**

* resourceName, description, make, model, year, urgency, status, fundCode, vendorCode, price, notes depend on acqID.
* member\_id depends on acqID.

### Acquisition Table:

* acqID determines resourceName, description, make, model, year, urgency, status, fundCode, vendorCode, price, notes, member\_id.
* member\_id depends on the member\_id in the Member table.

### Member Table:

* member\_id determines name, address, phone\_no, email, status, comments.

### Staff Table:

* member\_id depends on the member\_id in the Member table.

### Student Table:

* No dependencies other than the primary key.

### Loan Table:

* loan\_id determines datetime\_borrowed, datetime\_returned, datetime\_due, member\_id.
* member\_id depends on the member\_id in the Member table.

### Reservation Table:

* No dependencies other than the primary key.

### CourseOffering Table:

* No dependencies other than the primary key.

### Privilege Table:

* No dependencies other than the primary key.

### Category Table:

* No dependencies other than the primary key.

### Resource Table:

* No dependencies other than the primary key.

### Movable Table:

* loan\_id depends on the loan\_id in the Loan table.
* resource\_id depends on the resource\_id in the Resource table.

### Immovable Table:

* member\_id depends on the member\_id in the Member table.

# Summary

In this assignment we have created an EER model and convert into relational model. The relational model will be giving a DL. According to the given solution EER we have talked about the data requirement that are loan service course offered and reservation etc has been made. In the transaction requirement we have performed some query. And write down some business rules of what should happened and what should. We have performed the normalization to reduce the data redundancy.