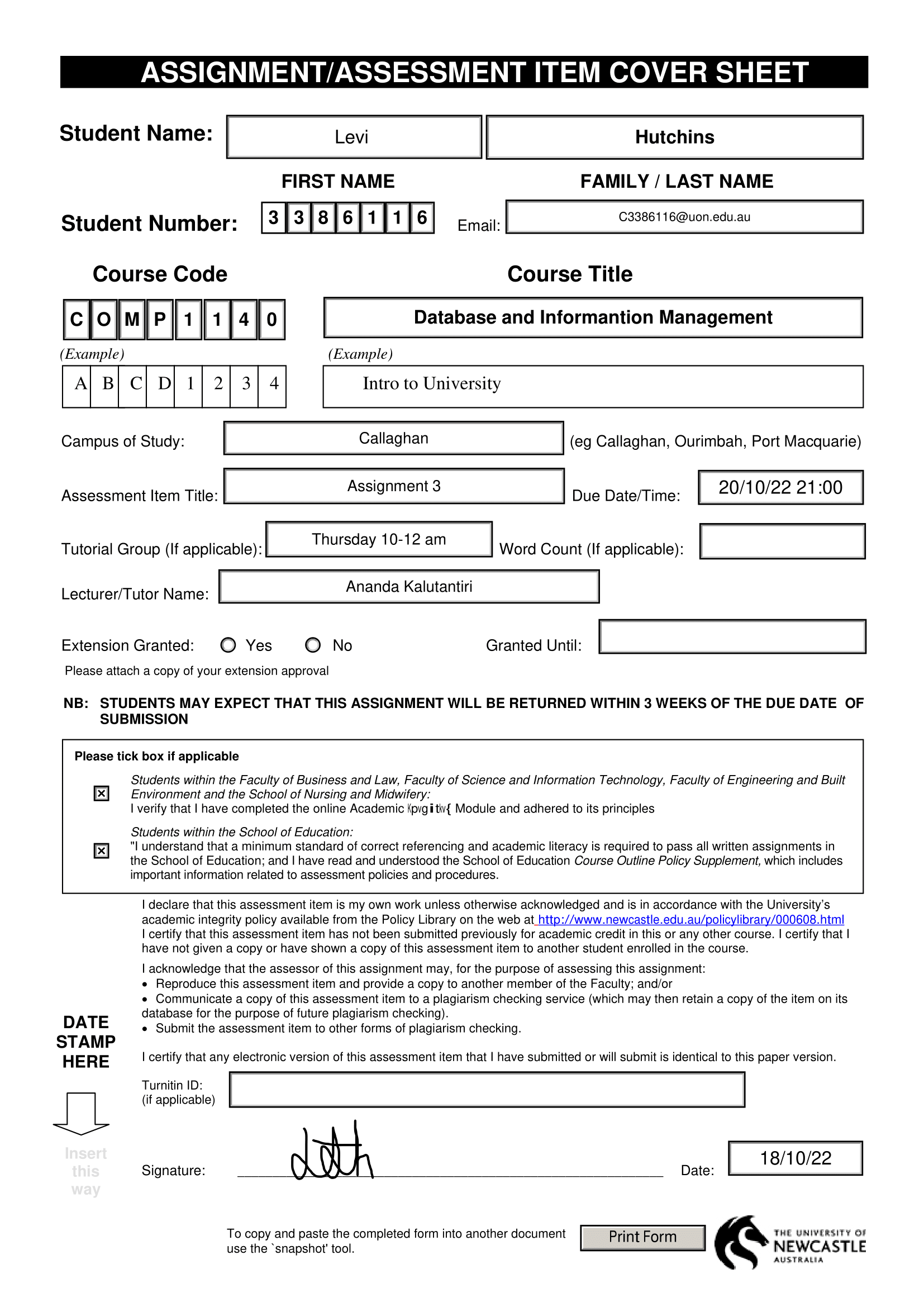
****

**COMP1140 ASSIGNMENT 2 Levi Hutchins – C3386116**

**SDS Resource Management Database Design Project-Physical Database Design**

Contents

[**Reflection Summary** 3](#_Toc117177191)

[**Revised Requirement Specification** 3](#_Toc117177192)

[**Data Requirements** 3](#_Toc117177193)

[**Transaction Requirements** 4](#_Toc117177194)

[**Business Rules** 5](#_Toc117177195)

[**Part 2 EER Model with Data Dictionary (Revised):** 6](#_Toc117177196)

[**EER Model** 6](#_Toc117177197)

[**Data Dictionary (Revised)** 7](#_Toc117177198)

[***Entities*** 7](#_Toc117177199)

[***Relationships*** 8](#_Toc117177200)

[***Attributes*** 8](#_Toc117177201)

[***Relational Model mapped from EER*** 12](#_Toc117177202)

[***Normalised Relation Schema*** 13](#_Toc117177203)

[SQL Code 17](#_Toc117177204)

# **Reflection Summary**

After viewing my Assignment 2 feedback I could quite clearly see what needed improve the most which was my relational model as well as my normalisation process. I was happy to see that my EER was pretty much correct except for one small relation missing which I could easily resolve accordingly. I believe I made all the correct adjustments to my relational design and normalisation process according to the markers feedback. I have spent several hours in writing my database design code and I am hoping to do well in this assignment. The area that needed the largest amount of my attention is the normalisation process and relational schema which I have no revised and improve and will hopefully reflect in mark I receive.

# **Revised Requirement Specification**

## **Data Requirements**

**Catalogue**

The School of Data Science (SDS) maintains and manages its online and physical resources available to students. The SDS has physical resources such as rooms, cameras, phones, and software that students can borrow or loan. SDS also provides online and software-based resources that members have access to.

*Resources*

The SDS has an array of resources available to members which they can loan for their degree. Every resource has:

* A resource ID (unique) - A Description
* Present status (“In Use”, “maintenance”,” available”, etc.)

Resources are then separated into two groups: moveable and immovable. Movable resources include speakers, phones, laptops, cameras etc. Immovable resources include rooms, labs, lab PC, etc.

Movable Resources Immovable Resources

* Name - Capacity ( max number of person )
* Make - Room
* Manufacturer - Building
* Model - Campus
* Year
* Asset value
* Building BDS

*Category*

Each resource is categorised into an appropriate category. Each category contains:

* A unique code
* Name
* Description
* Max time allowed to borrow/ book (days and or hours)

**Loan**

The SDS students that have enrolled in a course offered by the institution and staff both have lending reservation rights to the resources provided by the SDS.

*Members*

Both staff and students are referred to as members of the SDS system. Each “member” has the following fields:

* Unique ID - Name
* Address - Phone Number
* Email - Current Status
* Comments

Student members enrol in courses off by the SDS. Course information about course offerings and student enrolments are maintained by the SDS.

*Course offering*

A course offering has the following:

* Offering ID (unique)
* Name
* Semester offered
* Year Offered
* Courses begin / end date

Staff members can borrow and reserve resources and face no limits on the number of resources they are allowed access to.

*Privilege*

A course is assigned privileges to different categories of resources. Each privilege contains the following attributes:

* Name
* Description
* A category to which the privilege is granted for
* Maximum number of resources borrow/booked at a given time

**Reservations**

Members can make reservations on resource if their privilege allows them. When a reservation is made the resource will be booked for a specific pick-up time made by the member. No two reservations made by the member should conflict. Reservations have the following:

* Collection time / date
* Due date/time

**Acquisition**

A member can request a new acquisition to the SDS. The acquisition consists of the following:

* Person requesting acquisition
* Item name
* Make
* Manufacturer
* Model
* Year
* A Description of the item
* Urgency

An administrator of the system then assigns the following values to the item of acquisition:

* Status (“acquired”, “pending”, etc.)
* A fund code
* Vendor code
* Price
* Notes pertaining to the request

## **Transaction Requirements**

**Data Manipulation Operations**

* Insert/update/delete resources in the database
* Insert/update/delete resource location(s)
* Insert/update/delete an online information source
* Insert/update/delete a copy of a resource
* Insert/update/delete the status of a resource
* Insert/update/delete student(s)/staff
* Insert/update/delete student(s)/staff information
* Insert/update/delete reservations
* Insert/update/delete acquisition of resources

**Queries**

* Search for a resource based on keyword, title, serial number, author and/or type
* List current loans by specific student(s)/staff
* List resources that are frequently borrowed
* List student(s)/staff who are overdue
* List new resource acquisitions made by the student(s)/staff
* List fines report containing information on the student(s)/staff
* Display information on the student(s)/staff privileges
* Display information about resources with reservations
* Display information about student(s)/staff loan history
* Display information about the maximum amount of loans student(s)/staff can have

## **Business Rules**

**Expiration of student(s)/staff access**

A Students borrowing privilege is automatically removed when they finish their enrolled course.

A staff member’s borrowing privilege is automatically removed when they no longer work at (SDS).

The student(s)/staff member status is set to “disabled”.

**Maximum number of items loaned or reserved at one time**

Student(s) cannot borrow, or reserve more than their corresponding maximum number of items specified in their privileges at any one time.

Staff have priority over students regarding loaning of resources

**Late returns by student(s) fines**

Student members have a predefined number of points at the beginning of each course. Each student is given 12 points at the start.

A total of 3 points is deducted from the student for each day the resource is overdue

When the member’s points equal 0, their status is then set to “disabled”, and their ability to borrow is denied.

**Cancellation of reservations**

Reserved items are cancelled if it is not collected by the student member on the day of the required due date.

Members who do not cancel their reservations receive a 1-point deduction

Administrators have permission to cancel any reservations they see fit.

**Borrowing and or reservation periods**

the period of which a resource can be burrowed either being days or hours relates to which category it belongs. E.g. (laptops x days, microphones x hours).

# **Part 2 EER Model with Data Dictionary (Revised):**

## Diagram, schematic Description automatically generated**EER Model**

## **Data Dictionary (Revised)**

### ***Entities***

|  |  |  |  |
| --- | --- | --- | --- |
| Entity Name | Description | Aliases | Occurrence |
| Resources | Items that are both physical and software based used by members | Borrowed resource | Stored in locations either physical or software libraries |
| Immovable resources | Resources such as rooms, lab pcs etc | Rooms | Loaned to members  SDS |
| Movable Resources | Resources like phones, laptops etc |  | Loaned to members of SDS |
| Category | Groups of resources of similar characteristics |  | Items are grouped into categories for the convenience of finding |
| Loan | Borrowing of resource | Borrow | Resources are borrowed by members |
| Members | Either student or staff | Student or staff | Members make reservations about resources |
| Students | Members that can enrol into courses of SDS | Member | Students make reservations for resources |
| Staff | Members that work at SDS | Member | Staff make reservations / accept acquisitions |
| Privilege | Ability to loan from SDS |  | Members have different privileges on loaning resources |
| Reservations | Resources members can request/hold to borrow |  | Members submit a form to loan from SDS |
| Acquisitions | Members request new resources for the SDS |  | Member requests acquisition which is then accepted or denied by an administrator |
| Course Offering | A course that is offered by the SDS | Subject / Class | Members can participate in offered courses |

### ***Relationships***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity Name | Multiplicity | Relationship | Multiplicity | Entity Name |
| Loan | 0..\* | To | 1..1 | Movable Resource |
| Category | 1..1 | Belongs To | 0..\* | Resources |
| Category | 1..\* | For | 0..\* | Privilege |
| Resources | (Man, or) | Generalisation | (Man, or) | Movable Resource |
| Resources | (Man, or) | Generalisation | (Man, or) | Immovable Resource |
| Resources | 1..1 | To | 0..\* | Reservations |
| Immovable Resource | 1..1 | Contains | 1..\* | Category |
| Movable Resource | 1..1 | To | 0..\* | Loan |
| Movable Resource | (Man, Or) | Generalisation | (Man, Or) | Resources |
| Immovable Resource | (Man, Or) | Generalisation | (Man, Or) | Resources |
| Staff | (Man, Or) | Generalisation | (Man, Or) | Member |
| Staff | 1..1 | Updates | 0..\* | Acquisition |
| Reservations | 0..\* | To | 1..1 | Resources |
| Reservations | 0..\* | Has | 1..1 | Member |
| Member | 1..1 | Has | 0..\* | Reservations |
| Member | 1..1 | Requests | 0..\* | Acquisition |
| Student | 1..\* | Enrols | 0..\* | Course Offering |
| Privilege | 0..\* | Assigns | 0..\* | Course Offering |
| Privilege | 0..\* | For | 1..\* | Category |
| Acquisition | 0..\* | Requests | 1..1 | Member |

### 

### ***Attributes***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Entity Name | Attributes | Description | Data Type & Length | Nulls | Multi-valued | Derived | Default |
| Resources | Resource ID | ID of the resource | CHAR(5) | No | No | No | No |
| Description | Description of the resource | Varchar(100) | No | No | No | No |
| Status | Current status of a resource being loaned | Varchar(11) | No | No | Yes | ‘available’ |
| Course Offering | Offering ID | ID of the course offered | CHAR(5) | No | No | No | No |
| Name | Name of the course | Varchar(30) | No | No | No | No |
| Semester Offered | Semester that the course is offered | Char(1) | No | No | No | No |
| Year Offered | Year that the loan is offered | YEAR | No | No | No | No |
| Course Beginning/Ending Date | Date of the course’s beginning/ending | DATE | No | Yes | No | No |
| Immovable resources | Capacity (Max Person Capacity) | How many people can fit in the area | Varchar(30) | No | No | No | No |
| Room | Room name | VarChar(5) | No | No | No | No |
| Building | Building location | Varchar(15) | No | No | No | No |
| Campus | Campus location | Varchar(15) | No | No | No | No |
| Reservations | Reservation ID | ID of the reservations | Char(10) | No | No | No | No |
| Collection Date / Time | Date and time resources are to be given | DATE | Yes | No | No | No |
| Return Date / Time | Date and time resources are due | DATE | No | No | No | No |
| Staff | Admin Status | Is admin or Not | Varchar(15) | No | No | No | ‘Yes’ |
| Position | Position of the staff | Varchar(30) | No | No | No | No |
| Student | Points Available | Points related to borrowing | Int(2) | No | No | No | No |

***Attributes Continued.***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Entity Name | Attributes | Description | Data Type & Length | Nulls | Multi-valued | Derived | Default |
| Member | Member ID | ID of a member | CHAR(5) | No | No | No | No |
| Name | Name of member | Varchar(30) | No | No | No | No |
| Email | Email of member | Varchar(50) | Yes | No | No | No |
| Address | Address of member | Varchar(50) | Yes | No | No | No |
| Status | Status of member | Varchar(10) | No | No | No | ‘active’ |
| Phone | Phone number(s) of member | Char(8) | Yes | No | No | No |
| Comments | Comments about member | Varchar(50) | Yes | No | No | No |
| Loan | Loan ID | Status of the acquisition | CHAR(5) | No | No | No | No |
| Borrower | Person Loaning | VarChar(30) | No | No | No | No |
| Date Borrowed | Date/Time of borrowed item | DATE | No | No | No | No |
| Date Returned | Date/Time of borrowed item returned | DATE | Yes | No | No | No |
| Date Due | Date/Time of item due | DATE | No | No | No | No |
| Movable resources | Name | Name of resource | Varchar(30) | No | No | No | No |
| Make | Type of make of the resource | Varchar(25) | No | No | No | No |
| Manufacturer | Manufacturer of the resource | Varchar(30) | No | No | No | No |
| Model | Model of resource | Varchar(20) | No | No | No | No |
| Urgency | How bad the item is needed | Varchar(20) | No | No | Yes | No |
|  | Asset Value | Value of the resource | INT | No | No | No | No |

***Attributes Continued.***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Entity Name | Attributes | Description | Data Type & Length | Nulls | Multi-valued | Derived | Default |
| Privilege | Privilege ID | ID of the privilege | CHAR(5) | No | No | No | No |
|  | Description | Description of the privilege | Varchar(50) | Yes | No | No | No |
| Max number of resources borrowed at a given time | Time allowed for a member to borrow a resource | INT | No | Yes | No | No |
| Name | Name of the privilege | Varchar(30) | No | Yes | No | No |
| Acquisitions | Acquisition ID | ID of the acquisition | CHAR(5) | No | No | No | No |
| Name | Name of item | Varchar(30) | No | No | No | No |
| Make | Make of resource | Varchar(20) | No | No | No | No |
| Manufacturer | Manufacturer of item | Varchar(20) | No | No | No | No |
| Model | Model of resource | Char(12) | No | No | No | No |
| Year | Year the item was built | YEAR | No | No | No | No |
| Description | Resource description | Varchar(100) | No | No | No | No |
| Urgency | How much the item is needed | Varchar(20) | No | No | Yes | No |
| Status (“Pending”, ”Acquired”, etc. ) | Status of the acquisition | Varchar(15) | No | No | No | Acquired |
| Fund Code | Code of fund | Char(5) | No | No | No | No |
| Vendor Code | Code of vendor | Char(5) | No | No | No | No |
| Price | How much the item costs | Decimal(5,2) | No | No | No | No |
| Notes Pertaining Request | Extra Notes | Varchar(50) | No | No | No | No |
| Category | Category ID | ID of the category | CHAR(5) | No | No | No | No |
|  | Name | Name of the category | Varchar(30) | No | No | No | No |
|  | Description | Description of the category | Varchar(100) | No | No | No | No |
|  | Max time allowed to borrow in Hrs | Time a item can be borrowed | INT | No | No | No | No |

## ***Relational Model mapped from EER***

|  |
| --- |
| StudentMember (Member ID, Points Available)  Primary Key: Member ID  Foreign Key (Member ID) references Member (Member ID)  ON UPDATE CASCADE, ON DELETE NO ACTION |
| StaffMember (Member ID, Position, Admin status)  Primary Key: Member ID  Foreign Key Member ID references Member(Member ID)  ON UPDATE CASCADE, ON DELETE NO ACTION |
| Category (Category ID, Name, Description, Max time allowed to borrow)  Primary Key: Category ID |
| Loan (Loan ID, Member ID, Resource ID, Date Borrowed, Date Returned, Date Due)  Primary Key: Loan ID  Foreign Key: Member ID references Member (Member ID)  ON UPDATE CASCADE, ON DELETE NO ACTION  Foreign Key: Resource ID references Movable Resource (Resource ID)  ON UPDATE CASCADE ON DELETE NO ACTION |
| Resources (Resource ID, Category ID, Description, Status)  Primary Key: Resource ID  Foreign Key: Category ID references Category (Category ID)  ON UPDATE CASCADE, ON DELETE NO ACTION |
| Reservations (Reservation ID, Member ID, Resource ID, Collection Date, Return Date)  Primary Key: Reservation ID  Foreign key: Member ID references Member (Member ID)  ON UPDATE CASCADE, ON DELETE NO ACTION  Foreign Key: Resource ID references Resources (Resource ID)  ON UPDATE CASCADE ON DELETE CASCADE |
| Movable Resource (Resource ID, Name, Make, Manufacturer, Model, Year, Asset Value, Building SDS)  Primary Key: Resource ID  Foreign Key: Resource ID references Resources (Resource ID)  ON UPDATE CASCADE, ON DELETE NO ACTION |
| Immovable Resource (Resource ID, Capacity, Room, Building, Campus)  Primary Key: Resource ID  Foreign Key: Resource ID references Resources (Resource ID)  ON UPDATE CASCADE, ON DELETE NO ACTION |
| Member (Member ID, Name, Address, Phone, Email, Status, Comments)  Primary Key: Member ID |
| Acquisition (Acquisition ID, Member ID, Name, Make, Manufacturer, Model, Year, Description, Urgency, Status, Fund Code, Notes pertaining request, Price, Vendor Code)  Primary Key: Acquisition ID  Foreign Key: Member ID references Member (Member ID)  ON UPDATE CASCADE, ON DELETE NO ACTION |
| Privilege (Privilege ID, Name, Description, Max number of resources borrowed at a given time)  Primary Key: Privilege ID  Foreign Key: Category ID references Category (Category ID)  ON UPDATE CASCADE, ON DELETE NO ACTION |
| Course Offering (Offering ID, Name, Semester Offered, Year Offered, Begin Date, End Date)  Primary Key: Offering ID |
| CourseOffering\_Student (Offering ID, Member ID)  Primary Key: Offering ID  Foreign Key: Offering ID references CourseOffering (Offering ID)  ON UPDATE CASCADE, ON DELETE NO ACTION  Foreign Key: Member ID references Member (Member ID)  ON UPDATE CASCADE, ON DELETE NO ACTION |
| CourseOffering\_Privilege (Offering ID, Privilege ID)  Primary Key: Offering ID  Foreign Key: Offering ID references CourseOffering (Offering ID)  ON UPDATE CASCADE, ON DELETE NO ACTION  Foreign Key: Privilege ID references Privilege (Privilege ID)  ON UPDATE CASCADE, ON DELETE NO ACTION |

## ***Normalised Relation Schema***

**Normalisation 1**

Acquisition is currently in 2nd normal form.

**Acquisition (**Acquisition ID, Name, Make, Manufacturer, Model, Year, Description, Status, Fund Code, Vendor Code, Price, Other Notes Pertaining Request**)**

**Primary Key:** Acquisition ID

**FD1:**

Acquisition ID -> Member ID, Status, Fund Code, Vendor Code, Price, Other Notes Pertaining Request, item

**FD2:**

Item -> name, make, model, year description, price, manufacturer

There is a transitive dependency present with Acquisition ID -> item and item-> name, make, model, year, description, price, and manufacturer. To effectively remove this transitive dependency, we need to split the existing relation into two separated related tables like so:

Table 1:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Acquisition ID | Member ID | Status | FundCode | VendorCode | Price | OtherNotes | itemID |

Acquisition ( Acquisition ID, Member ID, Status, FundCode, VendorCode, Price, Other Notes pertaining to request, item)

**Primary Key:** Acquisition ID

**Foreign Key**: Member ID **references** Member (Member ID) on **update** cascade **delete** cascade

**Foreign Key**: item **references** AcquisitionItemInfo (item) on **update** cascade **delete** cascade

The table displays no transitive dependencies and is now in BCNF as the only functional dependency is the primary key.

Table 2:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ItemID | Name | Manufacturer | Year | Description | Model | Price |

AcquisitionItemInfo (item, Name, Manufacturer, year, description, model, price)

**Primary Key:** item

The table displays no transitive dependencies and is now in BCNF as the only functional dependency is the primary key.

**Normalisation 2**

Movable Resource is currently in 2nd normal form

Movable Resource (Resource ID, Name, Make, Manufacturer, Model, Year, Asset Value, Building SDS)

**Primary Key:** Resource ID

**FD1:**

Resource ID -> Name, Make, Manufacturer, Model, Year, Asset Value, Building SDS

**FD2:**

Moveable Item ID -> Name, Make, Manufacturer, Year, Asset Value, Building SDS.

We can clearly identify a transitive dependency with Resource ID -> Movable Item ID and Movable Item ID-> Name, Make, Manufacturer, Model, Year, Asset Value, and Building SDS. To effectively remove this transitive dependency, we need to split the existing relation into two separated related tables like so:

Table 1:

|  |  |  |
| --- | --- | --- |
| Resource ID | Movable Item ID | Model |

Movable Resource (Resource ID, Movable Item ID)

**Primary Key:** Resource ID

**Foreign Key:** Item ID **references** MovableItemInfo(Movable item ID) ON UPDATE CASCADE DELETE NO ACTION

This table now has no current transitive dependencies and is in BCNF as we can see the only functional dependency is the primary key.

Table 2:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Movable Item ID | Name | Make | Manufacturer | Year | Asset Value | Building SDS |

MovableItemInfo (Movable Item ID, Name, Make, Manufacturer, Model Year, Asset Value, Building SDS)

**Primary Key:** Movable Item ID

The table displays no transitive dependencies and is now in BCNF as the only functional dependency is the primary key.

We can now display all the relations in BCNF:

|  |
| --- |
| StudentMember(Member ID, Points Available)  Primary Key: Member ID |
| StaffMember(Member ID, Position, Admin Status)  Primary Key: Member ID |
| Category (Category ID, Name, Description, Max Time Allowed To Borrow)  Primary Key: Category ID |
| Loan (Loan ID, Member ID, Resource ID, Date Time Borrowed, Date Time Returned, Date Time Due)  Primary Key: Loan ID  Foreign Key Resource ID references Movable Resource (Resource ID) ON UPDATE CASCADE DELETE NO ACTION  Foreign Key Member ID references Member (Member ID) ON UPDATE CASCADE DELETE NO ACTION |
| Resources (Resource ID, Category ID, Description, Status)  Primary Key: Resource ID  Foreign Key Category ID references Category (Category ID) ON UPDATE CASCADE DELETE NO ACTION |
| Reservations (Reservation ID, Member ID, Resource ID, CollectionDateTime, ReturnDateTime)  Primary Key: Reservation ID  Foreign Key Member ID references Member (Member ID) ON UPDATE CASCADE DELETE NO ACTION  Foreign Key Resource ID references Resource (Resource ID) ON UPDATE CASCADE DELETE NO ACTION |
| Movable Resource (Resource ID, Model, Movable Item ID)  Primary Key: Resource ID  Foreign Key Resource ID references Resource (Resource ID) ON UPDATE CASCADE DELETE NO ACTION  Foreign Key Movable Item ID references MovableItemInfo (Movable Item ID) ON UPDATE CASCADE DELETE NO ACTION  MovableItemInfo(Movable Item ID, Name, Description, Manufacturer, Make, Asset Value, Building SDS)  Primary Key: Moveable Item ID |
| Immovable Resource (Resource ID, Capacity, Room, Building, Campus)  Primary Key: Resource ID  Foreign Key Resource ID references Resource (Resource ID) ON UPDATE CASCADE DELETE NO ACTION |
| Member (Member ID, Name, Address, Phone, Email, Email, Status, Comments)  Primary Key: Member ID |
| Acquisition (Acquisition ID, Member ID, Item ID, FundCode, VendorCode, Status, Urgency, Notes Pertaining request)  Primary Key: Acquisition ID  Foreign Key Member ID references Member (Member ID) ON UPDATE CASCADE DELETE NO ACTION  Foreign Key Item ID references AcquisitionItemInfo (Item ID) ON UPDATE CASCADE DELETE NO ACTION  AquisitionItemInfo(Item ID, Name, Make, Model, Manufacturer, Year, Price)  Primary Key: Item ID |
| Privilege (Privilege ID, Category ID, Name, Description, Max Number of resources borrow at a given time)  Primary Key: Privilege ID  Foreign Key Category ID references Category (Category ID) ON UPDATE CASCADE DELETE NO ACTION |
| CourseOffering (Offering ID, Name, Semester Offered, Year Offered, Begin Date End Date)  Primary Key: Offering ID |
| CourseOffering\_Student(Member ID, Offering ID)  Primary Key: Member ID, Offering ID  Foreign Key Member ID references Member (Member ID) ON UPDATE CASCADE DELETE CASCADE  Foreign Key Offering ID references CourseOffering (Offering ID) ON UPDATE CASCADE DELETE CASCADE |
| CourseOffering\_Privilege(Offering ID, Privilege ID)  Primary Key: Offering ID, Privilege ID  Foreign Key Offering ID references CourseOffering (Offering ID) ON UPDATE CASCADE NO ACTON  Foreign Key Privilege ID references Privilege (Privilege ID) ON UPDATE CASCADE DELETE CASCADE |

# SQL Code

-- Author Levi Hutchins C3386116

DROP TABLE Reservations

DROP TABLE Acquisition

DROP TABLE AcquisitionItemInfo

DROP TABLE CourseOffering\_Privilege

DROP TABLE CourseOffering\_Student

DROP TABLE CourseOffering

DROP TABLE StudentMember

DROP TABLE StaffMember

DROP TABLE Privilege

DROP TABLE MovableResource

DROP TABLE MovableItemInfo

DROP TABLE ImmovableResource

DROP TABLE Loan

DROP TABLE Resources

DROP TABLE Category

DROP TABLE Member

CREATE TABLE Member(

MemberID CHAR(5) PRIMARY KEY,

Name VARCHAR(30) NOT NULL,

Email VARCHAR(50) NOT NULL,

Address VARCHAR(50) NOT NULL,

Status VARCHAR(10) DEFAULT 'active' CHECK (Status IN ('active','expired')) NOT NULL,

Phone CHAR(10),

Comments VARCHAR(50),

);

go

CREATE TABLE Category(

CategoryID CHAR(5) PRIMARY KEY NOT NULL,

Name VARCHAR(30),

Description VARCHAR(100),

MaxTimeAllowedToBorrowInHrs INT NOT NULL,

);

go

CREATE TABLE Resources(

ResourceID CHAR(5) PRIMARY KEY NOT NULL,

CategoryID CHAR(5),

Description VARCHAR(100),

Status VARCHAR(11) DEFAULT 'available' CHECK (Status IN ('available','unavailable')) NOT NULL,

FOREIGN KEY (CategoryID) REFERENCES Category(CategoryID) ON UPDATE CASCADE ON DELETE CASCADE,

);

go

CREATE TABLE Loan(

LoanID CHAR(5) PRIMARY KEY,

ResourceID CHAR(5) NOT NULL,

MemberID CHAR(5) NOT NULL,

Borrower VARCHAR(30) NOT NULL,

DateBorrowed DATE NOT NULL,

DaeReturned DATE,

DateDue DATE NOT NULL,

FOREIGN KEY (MemberID) REFERENCES Member(MemberID) ON UPDATE CASCADE ON DELETE NO ACTION,

FOREIGN KEY (ResourceID) REFERENCES Resources ON UPDATE CASCADE ON DELETE NO ACTION

);

go

CREATE TABLE ImmovableResource(

ResourceID CHAR(5) PRIMARY KEY,

Capacity VARCHAR(30) NOT NULL,

Room VARCHAR(5),

Building VARCHAR(15),

Campus VARCHAR(15)

);

go

CREATE TABLE MovableItemInfo(

MovableItemID CHAR(5) PRIMARY KEY,

Name VARCHAR(30) NOT NULL,

Make VARCHAR(25) NOT NULL,

Manufacturer VARCHAR(30) ,

Urgency VARCHAR(20) ,

Year CHAR(4),

AssetValue INT NOT NULL,

BuildingSDS VARCHAR(30),

);

go

CREATE TABLE MovableResource(

ResourceID CHAR(5) PRIMARY KEY NOT NULL,

MovableItemID CHAR(5) NOT NULL,

Model VARCHAR(20) ,

FOREIGN KEY (ResourceID) REFERENCES Resources ON UPDATE CASCADE ON DELETE NO ACTION,

FOREIGN KEY (MovableItemID) REFERENCES MovableItemInfo(MovableItemID) ON UPDATE CASCADE ON DELETE NO ACTION

);

go

CREATE TABLE Privilege(

PrivilegeID CHAR(5) PRIMARY KEY,

CategoryID CHAR(5) NOT NULL,

Name VARCHAR(30) NOT NULL,

Description VARCHAR(50),

BorrowedAtOneTime INT NOT NULL,

FOREIGN KEY (CategoryID) REFERENCES Category ON UPDATE CASCADE ON DELETE CASCADE

);

go

CREATE TABLE StaffMember(

MemberID CHAR(5) PRIMARY KEY,

Position VARCHAR(30),

Status VARCHAR(15) DEFAULT 'Yes' CHECK (Status IN ('Yes','No')) NOT NULL,

FOREIGN KEY (MemberID) REFERENCES Member(MemberID) ON UPDATE CASCADE ON DELETE NO ACTION,

);

go

CREATE TABLE StudentMember(

MemberID CHAR(5) PRIMARY KEY NOT NULL,

PointsAvailable INT,

FOREIGN KEY (MemberID) REFERENCES Member(MemberID) ON UPDATE CASCADE ON DELETE NO ACTION,

);

go

CREATE TABLE CourseOffering(

OfferingID CHAR(5) PRIMARY KEY,

Name VARCHAR(30),

SemesterOffered CHAR(1),

YearOffered CHAR(4),

BeginDate DATE,

EndDate DATE

);

go

CREATE TABLE CourseOffering\_Student(

OfferingID CHAR(5) PRIMARY KEY,

MemberID CHAR(5) NOT NULL,

FOREIGN KEY (OfferingID) REFERENCES CourseOffering ON UPDATE CASCADE ON DELETE NO ACTION,

FOREIGN KEY (MemberID) REFERENCES StudentMember ON UPDATE CASCADE ON DELETE NO ACTION

);

go

CREATE TABLE CourseOffering\_Privilege(

OfferingID CHAR(5) PRIMARY KEY,

PrivilegeID CHAR(5) NOT NULL,

FOREIGN KEY (OfferingID) REFERENCES CourseOffering(OfferingID) ON UPDATE CASCADE ON DELETE NO ACTION,

FOREIGN KEY (PrivilegeID) REFERENCES Privilege(PrivilegeID) ON UPDATE CASCADE ON DELETE NO ACTION,

);

go

CREATE TABLE AcquisitionItemInfo(

ItemID CHAR(5) PRIMARY KEY,

Name VARCHAR(30) NOT NULL,

Manufacturer VARCHAR(50) NOT NULL,

YEAR CHAR(4) NOT NULL,

Description VARCHAR(50),

Model VARCHAR(20),

Price INT NOT NULL,

);

go

CREATE TABLE Acquisition(

AcquisitionID CHAR(5) PRIMARY KEY,

MemberID CHAR(5),

ItemID CHAR(5),

Urgency VARCHAR(20),

Status VARCHAR(15) DEFAULT 'Acquired' CHECK (Status IN ('Acquired','Pending')) NOT NULL,

FundCode CHAR(5),

VendorCode CHAR(5),

Notes VARCHAR(50),

FOREIGN KEY (ItemID) REFERENCES AcquisitionItemInfo(ItemID) ON UPDATE CASCADE ON DELETE NO ACTION,

FOREIGN KEY (MemberID) REFERENCES Member(MemberID) ON UPDATE CASCADE ON DELETE NO ACTION

);

go

CREATE TABLE Reservations(

ReservationID CHAR(5) PRIMARY KEY,

MemberID CHAR(5) NOT NULL,

ResourceID CHAR(5) NOT NULL,

CollectionDateTime DATE,

ReturnDateTime DATE NOT NULL,

FOREIGN KEY (MemberID) REFERENCES Member(MemberID) ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY (ResourceID) REFERENCES Resources(ResourceID) ON UPDATE CASCADE ON DELETE CASCADE

);

go

INSERT INTO Member VALUES ('MB001','Levi Hutchins','C0001@uon','First Street','active','0422338791',NULL);

INSERT INTO Member VALUES ('MB002','Bob Smith','C2000@uon','Charles Av','expired','0622339101',NULL);

INSERT INTO Member VALUES ('MB003','John Wayne','C3201@uon','March Close','active','0299199191',NULL);

INSERT INTO Member VALUES ('MB004','Olivia Jane','C4221@uon','David Parade','expired','0213568190',NULL);

INSERT INTO Member VALUES ('MB200','Marge Swan','C3111@uon','Swan Close','Active','0482710391',NULL);

INSERT INTO Member VALUES ('MB201','Harry Gray','C1010@uon','Charles Kay Dr','expired','0456192371',NULL);

INSERT INTO Member VALUES ('MB202','Ben Roberts','C0000@uon','Penns Rd','active','0462904718',NULL);

INSERT INTO Member VALUES ('MB203','Bart Simpson','C0020@uon','Evergreen Terrace','active','1234567890',NULL);

INSERT INTO Category VALUES('CR100','Laptop','Fast laptops',24);

INSERT INTO Category VALUES('CR200','Speaker','Genuine JBL speakers',48);

INSERT INTO Category VALUES('CR250','Camera','HD Camera',48);

INSERT INTO Category VALUES('CR300','Room','Collaboration Rooms',5);

INSERT INTO Category VALUES('CR400','Desktop','Alienware desktops',3);

INSERT INTO Resources VALUES('RS100','CR100','Fast laptop great for data science students','available');

INSERT INTO Resources VALUES('RS200','CR100','Portable laptop for simple document work','available');

INSERT INTO Resources VALUES('RS300','CR200','Loud speaker with great bass','unavailable');

INSERT INTO Resources VALUES('RS350','CR250','HD Photography Camera','unavailable');

INSERT INTO Resources VALUES('RS400','CR300','Collaboration room for math students','unavailable');

INSERT INTO Resources VALUES('RS500','CR300','Collaboration room for computer science students','unavailable');

INSERT INTO Resources VALUES('RS600','CR300','Collaboration room for science students','unavailable');

INSERT INTO Resources VALUES('RS700','CR300','Collaboration room for law students','unavailable');

INSERT INTO Loan VALUES('LN001','RS100','MB001','Levi Hutchins','10/12/2021',NULL,'10/18/2022')

INSERT INTO Loan VALUES('LN002','RS200','MB002','Bob Smith','10/8/2021',NULL,'12/11/2022')

INSERT INTO Loan VALUES('LN003','RS100','MB001','Levi Hutchins','10/1/2021',NULL,'10/10/2022')

INSERT INTO Loan VALUES('LN004','RS500','MB004','Olivia Jane','10/8/2021',NULL,'12/11/2022')

INSERT INTO Loan VALUES('LN005','RS350','MB002','Bob Smith','10/7/2022',NULL,'12/7/2022')

INSERT INTO Loan VALUES('LN006','RS350','MB001','Levi Hutchins','8/7/2022',NULL,'12/7/2022')

INSERT INTO ImmovableResource VALUES ('RS400','30',NULL,NULL,'Callaghan');

INSERT INTO ImmovableResource VALUES ('RS500','50','ES100',NULL,'Callaghan');

INSERT INTO ImmovableResource VALUES ('RS600','20',NULL,NULL,'Callaghan');

INSERT INTO ImmovableResource VALUES ('RS700','10',NULL,NULL,'Callaghan');

INSERT INTO MovableItemInfo VALUES('MI001','Fast laptop','ASUS',NULL,NULL,'2021',1500,NULL);

INSERT INTO MovableItemInfo VALUES('MI002','Portable speaker','JBL',NULL,'Urgent','2018',100,NULL);

INSERT INTO MovableItemInfo VALUES('MI003','Camera','Sony',NULL,'Not Urgent','2016',1200,NULL);

INSERT INTO MovableItemInfo VALUES('MI004','Portable Laptop','Apple',NULL,NULL,'2022',1500,NULL);

INSERT INTO MovableResource VALUES('RS100','MI001',NULL);

INSERT INTO MovableResource VALUES('RS200','MI004',NULL);

INSERT INTO MovableResource VALUES('RS300','MI002',NULL);

INSERT INTO MovableResource VALUES('RS350','MI003','SONY AX500');

INSERT INTO Privilege VALUES('PR001','CR100','StudentMember','Comp Sci Student',4);

INSERT INTO Privilege VALUES('PR002','CR200','StudentMember','Math Student',4);

INSERT INTO Privilege VALUES('PR003','CR300','StaffMember',NULL,6);

INSERT INTO Privilege VALUES('PR004','CR400','StaffMember',NULL,6);

INSERT INTO StaffMember VALUES('MB200','Admin','Yes');

INSERT INTO StaffMember VALUES('MB201','Lecturer','No');

INSERT INTO StaffMember VALUES('MB203','CEO','No');

INSERT INTO StaffMember VALUES('MB202','Coordinator','Yes');

INSERT INTO StudentMember VALUES('MB001',12);

INSERT INTO StudentMember VALUES('MB002',12);

INSERT INTO StudentMember VALUES('MB003',12);

INSERT INTO StudentMember VALUES('MB004',12);

INSERT INTO CourseOffering VALUES('OF101','Data Structures','2','2022','6/1/2022','10/31/2022');

INSERT INTO CourseOffering VALUES('OF202','Discrete Mathematics','2','2022','6/1/2022','10/31/2022');

INSERT INTO CourseOffering VALUES('OF303','Database Information','2','2022','6/1/2022','10/31/2022');

INSERT INTO CourseOffering VALUES('OF404','Web Technologies','2','2022','6/1/2022','10/31/2022');

INSERT INTO CourseOffering\_Student VALUES('OF202','MB001')

INSERT INTO CourseOffering\_Student VALUES('OF404','MB003')

INSERT INTO CourseOffering\_Student VALUES('OF303','MB004')

INSERT INTO CourseOffering\_Student VALUES('OF101','MB002')

INSERT INTO CourseOffering\_Privilege VALUES('OF202','PR001');

INSERT INTO CourseOffering\_Privilege VALUES('OF101','PR002');

INSERT INTO CourseOffering\_Privilege VALUES('OF404','PR003');

INSERT INTO CourseOffering\_Privilege VALUES('OF303','PR004');

INSERT INTO AcquisitionItemInfo VALUES('AIT01','Norton 360','Norton','2022',NULL,NULL,99);

INSERT INTO AcquisitionItemInfo VALUES('AIT02','Surface Pro','Microsoft','2021',NULL,NULL,1500);

INSERT INTO AcquisitionItemInfo VALUES('AIT03','Ipad Pro','Aplle','2022',NULL,NULL,1100);

INSERT INTO AcquisitionItemInfo VALUES('AIT04','Azure Software','Microsoft','2019',NULL,NULL,10000);

INSERT INTO Acquisition VALUES('A0001','MB200','AIT01',NULL,'Acquired',NULL,NULL,NULL);

INSERT INTO Acquisition VALUES('A0002','MB201','AIT02',NULL,'Pending',NULL,NULL,NULL);

INSERT INTO Acquisition VALUES('A0003','MB001','AIT04',NULL,'Acquired',NULL,NULL,NULL);

INSERT INTO Acquisition VALUES('A0004','MB002','AIT03',NULL,'Pending',NULL,NULL,NULL);

INSERT INTO Reservations VALUES('REV01','MB001','RS100','2/1/2021','3/3/2021');

INSERT INTO Reservations VALUES('REV02','MB002','RS400','6/6/2018','6/6/2018');

INSERT INTO Reservations VALUES('REV03','MB203','RS300','1/1/2022','3/3/2022');

INSERT INTO Reservations VALUES('REV04','MB203','RS400','12/3/2022','12/3/2022');

INSERT INTO Reservations VALUES('REV05','MB203','RS400','12/3/2022','12/3/2022');

INSERT INTO Reservations VALUES('REV06','MB200','RS200',NULL,'3/3/2023');

INSERT INTO Reservations VALUES('REV07','MB003','RS500','2022/06/05','2022/06/05');

-- Question 1

SELECT Name

FROM Member

WHERE MemberID =

(SELECT MemberID FROM

CourseOffering\_Student

WHERE OfferingID = 'OF202');

-- Question 2

SELECT BorrowedAtOneTime

FROM Privilege

WHERE CategoryID =

(SELECT CategoryID FROM

Category WHERE CategoryID ='CR200');

-- Question 3

SELECT m.Name AS 'Name', m.Phone AS 'Phone',

COUNT(r.MemberID) AS 'Reservations'

FROM Member m, Reservations r, StaffMember s

WHERE r.MemberID = s.MemberID AND s.MemberID = m.MemberID AND s.MemberID = 'MB203'

AND YEAR(r.CollectionDateTime) = '2022'

GROUP BY m.Name, m.Phone;

-- Question 4

SELECT DISTINCT(m.Name) AS 'Sony AX500 Borrowed By'

FROM Member m, StudentMember sm, Category c, Loan l, MovableResource mrs, Resources rc

WHERE l.MemberID = m.MemberID AND m.MemberID = sm.MemberID AND

l.ResourceID = rc.ResourceID AND rc.CategoryID = c.CategoryID AND c.Name = 'Camera' AND mrs.Model = 'SONY AX500'

AND YEAR(l.DateBorrowed) = YEAR(SYSDATETIME());

-- Question 5

SELECT res.resourceID, item.Name

FROM MovableResource res, MovableItemInfo item, Loan l

WHERE res.MovableItemID = item.MovableItemID AND l.ResourceID = res.ResourceID AND Month(l.DateBorrowed) = Month(SYSDATETIME())

AND Year(l.DateBorrowed) = YEAR(SYSDATETIME())

GROUP BY res.ResourceID, item.Name;

-- Question 6

SELECT res.CollectionDateTime, im.Room,

COUNT(DISTINCT(res.ReservationID)) AS 'Num of Reservations'

FROM ImmovableResource im, Resources r, Reservations res

WHERE r.CategoryID = 'CR300' AND r.ResourceID = 'RS500' AND im.Room = 'ES100' AND res.CollectionDateTime = '2022/06/05' OR res.CollectionDateTime = '2022/05/01' OR res.CollectionDateTime = '2022/09/19'

GROUP BY res.CollectionDateTime, im.Room, r.CategoryID, res.ReservationID;