SDS Resource Management database

Reflection assignment 3

In assignment 2 I had better time management than this assignment and first assignment didn’t really take the time to focus on fixing mistakes in assignment 2 I have put specifically put time into the creation of the physical database and plan to revise the course content focus that Assignment 2 go over in the exam period .

Requirement Analysis

Data requirements

Catalogue Search

A search process that investigates a collection and the location to find a resource using keywords the Catalogue Search will have

* keywords
* unique code

Category

Every resource is put into a category such as the computer category.

* Unique code
* Duration in hours
* Name
* Description

Member

Members can be categorised into two different groups students and staff. each member is either enrolled or employed at the SDS at the University of Northwest Technology

* Unique id
* Name
* Address
* Phone
* Email
* Status
* Comments field

Privilege

A member is assigned a privilege with a name, description, and maximum items students and staff will have a different privilege depending on their requirements at the SDS at the University of Northwest Technology these privileges will determine what resources and how many can be reserved or borrowed

* Unique id
* Description
* Maximum items

Acquisition Request

A member can request or updated new acquisitions to the school. An acquisition request contains this data and will let the admin of SDS know what the members would want/need in the school.

* acquisition
* item name
* make manufacturer
* model
* year
* description
* urgency
* status
* vendor code
* price
* fund code
* Notes

Reservation

Members can reserve resources that their privileges allow them to borrow. Once a reservation is made, the resource will be booked for use by the member on the requested date and time and won’t end until the end date and time is past reservations will be made of first come first serve basis .

* Unique code
* Date time start
* Date time end

Loan

SDS allows each member to take resources on loan that their privileges these resources range from movable and immovable e.g., computer ,rooms and cameras.

* Borrowed
* due
* Returned resource

Location

Where collections of resources are located each will have

* unique code
* room

Resources

The School of Data Science (SDS) at the University of Northwest Technology has many different resources e.g., rooms, cameras, speakers, software, and phones data associated with the resources.

* Unique id
* Status
* Description

Movable resources

Resources that are mobile data associated with these resources

* Year
* Model
* Name
* Manufacturer
* Value
* Make

Immovable resources

Resources that aren’t mobile data associated with these resources

* Capacity

Student

Members who are study in one course offered by SDS data associated with these resources

* Points

Staff

Employees of SDS data associated with these resources

* Position

Course

A class administer by the SDS data associated with these resources

* Unique code
* Name

Course offering

The way members are entered into the SDS Courses data associated with these resources

* Unique code
* Name
* Start date
* End data
* Semester offered
* Year offered

Business rules

1. A member's borrowing privileges are automatically taken away when the current date is later than the end date of all his/her enrolled courses. The status of the member is set to "disabled."
2. Maximum items loaned or reserved at any one time. A member cannot borrow, or reserve more than the maximum number of items specified in his/her privileges at any given time
3. The duration of borrowing/reservation periods (either number of days or hours) are determined by the category to which the item belongs
4. The members should be able to hold items from the library
5. first come-first-serve basis for reservation requests
6. When an item on hold is not collected it will be returned to category a within a week
7. Students are lower than staff in priority when using the loan service
8. Students have a set amount points these points are controlled by administrators’ points will be taken away as a penalty of overdue resources
9. Reservations should not overlap each over

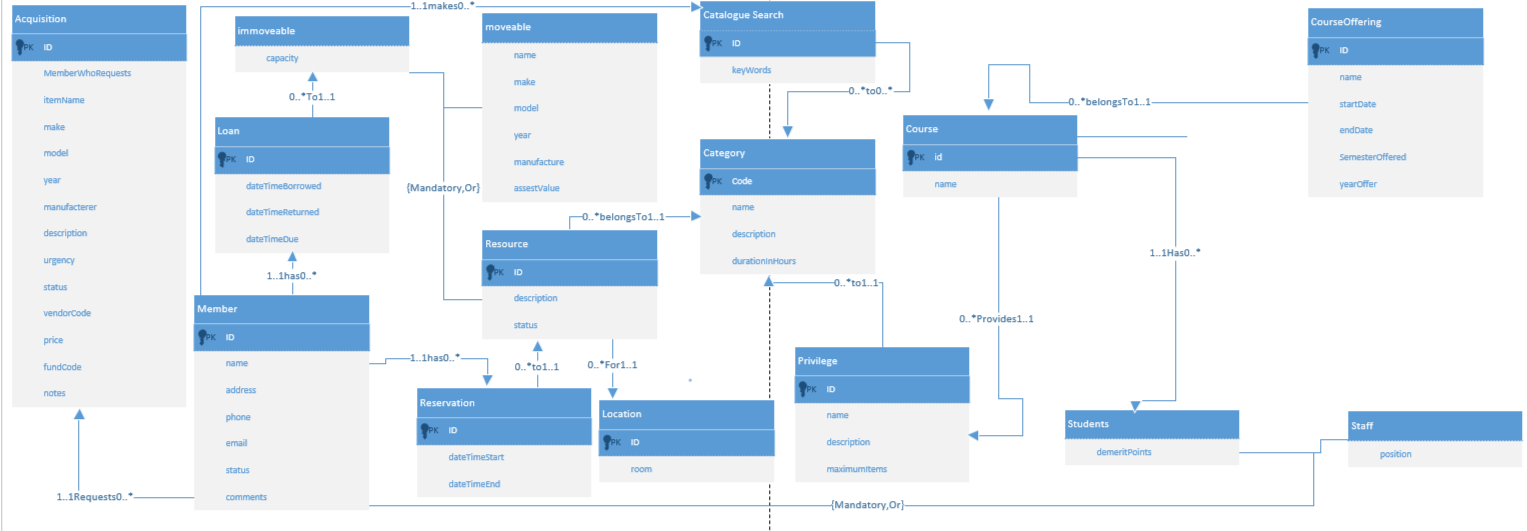
Transaction requirements

Data update/deletion

* + Update/insert/delete the details of a Resource
  + Update/insert/delete the details of Members
  + Update/insert/delete the details of a Reservation
  + Update/insert/delete the details of a Category
  + Update/insert/delete the details of a Location
  + Update/insert/delete the details of an Acquisition Request
  + Update/insert/delete the details of a Privilege
  + Update/insert/delete the details of a Catalogue Search
  + Update/insert/delete the details of a Loan
  + Update/insert/delete the details of Moveable resources

Data queries

* + Search the catalogue based on keywords
  + Search a loan resource based on a member ID
  + Search a reservation resource based on a member ID
  + List all the reservations for a particular item
  + list member privileges information
  + Display the ID of Members who have an Acquisition Request
  + Display the details of a loan
  + List the details of the location in each area.
  + Identify the total number of members
  + Identify the loans that have been taken more than the average number of times.
  + Identify the reservation that has been taken more than the average number of times
  + List data and time of available immovable resources
  + List Items returning on a specific day
  + List recent acquisitions within a two-month period



|  |  |  |  |
| --- | --- | --- | --- |
| Entity name | Description | Aliases | Occurrences |
| Member | Persons that are in the SDS system | Student and Staff | Individuals in the SDS system |
| Students | Person enrolled in a SDS courses | Member | When a person enol in an SDS courses |
| Staff | Person employed by SDS | Member | When a person is put on the payroll of SDS |
| Acquisition request | A request for SDS to get an item | resource request | When SDS get a request to get an item |
| Location | land owned and mange by SCS | room | Physical area owned by SDS |
| Category | A Collection of resources | Collection | When Resources get put into a category |
| Privilege | Members that have access to SDS resources under certain conditions | Permission | Conditions that allow for a member to access resources |
| Catalogue search | The ability to find resources | SDS search | When members initiate a search for resources in the catalogue |
| Resources | Things acquired by SDS | Items | When SDS buy resources |
| Movable resources | Mobile resources | Item | When SDS gain a portable resource |
| Immovable resources | Not mobile resources | Room | When SDS gains a non-portable resource |
| Loan | SDS allow for member to borrowed resources | Lease | When member borrow resources |
| Reservation | SDS allow for member to reserve resources | Lease | When member reserve resources |

Entities

Entities Relationships

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity name | Multiplicity | Relationship | Multiplicity | Entity name |
| Staff |  | {Mandatory Or} |  | Member |
| Student |  | {Mandatory Or} |  | Member |
| Member | 1..1 | Makes | 0..\* | Acquisition request |
| Catalogue Search | 0..\* | Makes | 0..\* | Category |
| Resource | 0..\* | BelongTo | 1..1 | Category |
| Resource | 0..\* | For | 1..1 | Location |
| Course | 0..\* | Provides | 1..1 | Privilege |
| Course | 1..1 | Has | 0..\* | Students |
| Course | 1..1 | Has | 0..\* | Staff |
| Member | 1..1 | Has | 0..\* | Reservation |
| Course Offering | 0..\* | BelongTo | 1..1 | Course |
| Moveable |  | {Mandatory Or} |  | Resource |
| immoveable |  | {Mandatory Or} |  | Resource |
| Member | 1..1 | Has | 0..\* | Loan |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Entity | Attributes | Description | Date type & Length | Nulls | Multi valued | Derived | Default |
| Resource | ID No | Unique number | Char (15) | N | N | N | None |
| Resource | Status | What condition the resource is in | Varchar (25) | N | N | N | None |
| Resource | Description | The features of resource | Varchar (40) | Y | N | N | None |
| Member | ID No | Unique number | char (30) | N | N | N | None |
| Member | Name | Member name they go by | Varchar(45) | N | N | N | None |
| Member | Address | Place of residence | Varchar(45) | N | N | N | None |
| Member | Status | Whether the member is active or inactive | Varchar(45) | N | N | N | None |
| Member | Comments | Comment relevance to members | Varchar(45) | N | N | N | None |
| Member | Email | Email address of an individual | Varchar(45) | N | N | N | None |
| Member | Phone | Number of an individual | Char (15) | N | N | N | 10 |
| Catalogue Search | ID No | Unique number | Char (30) | N | N | N | None |
| Catalogue Search | Keyword used | List of Keywords used | Char(30) | Y | N | N | None |
| Location | ID No | Unique number | Char (30) | N | Y | N | None |
| Location | Room | The features of room | Char (30) | Y | N | N | None |
| Category | code | Unique number | Char (15) | N | N | N | None |
| Category | Name | a grouping of the items called a specific thing | Varchar(30) | N | N | N | None |
| Category | Description | features of collection | Varchar (30) | N | N | N | None |
| Category | Duration | Total duration of category | VARCHAR int | N | N | N | None |
| Privilege | ID No | Unique number | char (15) | N | N | N | None |
| Privilege | Name | Privileges each member has | Varchar (15) | N | N | N | None |
| Privilege | Maximum items | The features of privilege | Int() | N | N | N | None |
| Privilege | Description | Privilege description | Varchar (45) | N | N | N | None |
| Acquisition Request | Acquisition ID | Unique number | Char (15) | N | N | N | None |
| Acquisition Request | Notes | The features of requested Acquisition details | Varchar (45) | N | N | N | None |
| Acquisition Request | Status | Whether a request is active or inactive | Varchar (15) | Y | N | Y | None |
| Acquisition Request | Name | Item name | Varchar (45) | N | N | N | None |
| Acquisition Request | Fund code | Fund code | Varchar (45) | N | N | N | None |
| Acquisition Request | Price | Item cost | Decimal (5) | N | N | N | None |
| Acquisition Request | Vendor code | Vendor code | Varchar (45) | N | N | N | None |
| Acquisition Request | Description | Item description | Varchar (45) | N | N | N | None |
| Acquisition Request | Urgency | Request urgency | Varchar (45) | N | N | N | None |
| Acquisition Request | Year | Released year item | Char (4) | N | N | N | None |
| Acquisition Request | Model | Version item | Varchar (45) | N | N | N | None |
| Acquisition Request | Make | Version item | Varchar (45) | N | N | N | None |
| Acquisition Request | Manufacturer | Maker of item | Varchar (45) | Y | Y | N | None |
| Reservation | ID | Unique number | Char (45) | N | N | N | None |
| Reservation | Date time start | Started Reservation time and date | Date() | N | N | N | None |
| Reservation | Date time end | Ended Reservation time and date | Date() | N | N | N | None |
| Loan | ID | Unique number | INT (1,1) | N | Y | N | None |
| Loan | Date time borrowed | Loan details of date and time | Date Time() | N | N | N | None |
| Loan | Date time due | Due date and time | Date Time() | N | N | Y | None |
| Loan | Date time returned | Recorded of item being returned | Date Time() | N | N | N | None |
| Moveable | Name | Item name | Varchar (30) | N | N | N | None |
| Moveable | Model | Item id | Varchar (30) | Y | N | N | None |
| Moveable | Manufacturer | Item maker | Varchar (30) | Y | N | N | None |
| Moveable | Asset value | Monetary Value of item | Decimal | Y | N | N | None |
| Moveable | Year | Year item was made | char (4) | Y | N | N | None |
| Moveable | Make | Vision of item | Varchar (30) | Y | N | N | None |
| Immoveable | Capacity | Maximum amount People in a room | Int() | Y | N | N | None |
| Staff | Position | Job details | Char (15) | N | N | N | 15 |
| Students | DemeritPoints | Students Points | Int() | N | N | N | 12 |
| Course | Id | Unique code | Char (30) | N | N | N | 15 |
| Course | Name | Couse name | Varchar (30) | N | N | N |  |

Relational model mapped with EER

**Acquisition** (ID, item name ,member who requests , make, model, year, manufacturer, description, urgency, status, vendor code, price, fund code, notes)

**Primary key** ID

**Foreign Key** member id references member (ID)

**ON UPDATE CASCADE,ON DELETE CASCADE**

**Movable (**ID, assetValue, year, manufacturer, make, model, name**)**

**Primary key** ID

**Foreign Key** ID **references** resource (ID)

**ON UPDATE CASCADE,ON DELETE CASCADE**

**Immovable (**ID, capacity)

**Primary key** ID

**Foreign Key** ID **references** resource  **(**ID**)**

**ON UPDATE CASCADE,ON DELETE CASCADE**

**Reservation (**ID, dateTimeEnd, dateTimeStart**)**

**Primary key** ID

**Foreign Key** ID **references** Member (ID)

**ON UPDATE CASCADE,ON DELETE CASCADE**

**Foreign Key** ID **references** Resources (ID)

**ON UPDATE CASCADE,ON DELETE CASCADE**

**Resource (**ID, description, status**)**

**Primary key** ID

**Foreign key** code **references** Category (code)

**Foreign Key** ID **references** Location(ID)

**ON UPDATE CASCADE,ON DELETE CASCADE**

**CourseOffering** (ID, semesterOffered, yearOffered, startDate, endDate**)**

**Primary key** ID

**Foreign key** ID **references** Course (ID)

**ON UPDATE CASCADE,ON DELETE CASCADE**

**Location** (ID,Room)

**Primary key** ID

**Category** (Code, description, duration, name)

**Primary Key** Code

**Course** (ID, name)

**Primary key** ID

**Privilege** (ID, name, description, maximumItems)

**Primary key** ID

**Foreign key** ID references course (ID)

**ON UPDATE CASCADE,ON DELETE CASCADE**

**Loan** (ID, dateTimeReturned, dateTimeBorrwed, dateTimeDue)

**Primary key** ID

**Foreign key** ID **references** Movable (ID)

**ON UPDATE CASCADE,ON DELETE CASCADE**

**Foreign key** ID **references** member (ID)

**ON UPDATE CASCADE,ON DELETE CASCADE**

**Student**  (ID, points, status, name, address, phone, comments, email)

**Primary key** ID

**Foreign key** ID **references** course (ID)

**Foreign key** ID **references** member (ID)

**ON UPDATE CASCADE,ON DELETE CASCADE**

**Staff** (ID, position, status , name, address, phone, comments, email)

**Primary key** ID

**Foreign key** ID  **references** member (ID)

**ON UPDATE CASCADE,ON DELETE CASCADE**

**Member** (ID, status, name, address, phone, comments, email)

**Primary key** ID

**ON UPDATE CASCADE,ON DELETE CASCADE**

Relation Normalization

**DBDL form before normalisation**

**Staff (**ID, position , status, name, address, phone, comments, email**)**

**Primary key** ID

**Foreign key** ID **references member** (ID)

**ON UPDATE CASCADE,ON DELETE CASCADE**

**Normalisation**

**Staff** (ID, position , status, name, address, phone, comments, email**)**

FD1: ID->ID, position , name, address, phone, comments, email

R relation of staff

R: (ID, position , name, address, phone, comments, email)

Dependency in BCNF

**Normalisation in DBDL form**

**Staff** (ID, position , status, name, address, phone, comments, email)

**Primary key** ID

**Foreign key** ID  **references member** (ID)

**ON UPDATE CASCADE,ON DELETE CASCADE**

**DBDL form before normalisation**

**Student** (ID, points, status, name, address, phone, comments, email)

**Primary key** ID

**Foreign key** ID  **references** course (ID)

**Foreign key** ID **references** member (ID)

**ON UPDATE CASCADE,ON DELETE CASCADE**

**Normalisation**

**Student** (ID, points, status, name, address, phone, comments, email)

FD1: ID-> ID points, name, status, address, phone, comments, email

R is the relation of staff

R: (ID, position , name, address, status ,phone, comments, email)

Dependency in BCNF

**Normalisation in DBDL form**

**Student** (ID, points, name, address, status, phone, comments, email)

**Primary key** ID

**Foreign key** ID  **references** course (ID)

**Foreign key** ID **references** member (ID)

**ON UPDATE CASCADE,ON DELETE CASCADE**

**Database code**

drop table Courseoffing

drop table CatalogueSearch

drop table Course

drop table Reservation

drop table Privilege

drop table Loan

drop table Acquisition

drop table ImmoveableResource

drop table MoveableResource

drop table Resource

drop table Location

drop table Category

drop table Staff

drop table Student

drop table Member

CREATE TABLE Member

(

memberId CHAR(30) NOT NULL,

name VARCHAR(45) NOT NULL,

address VARCHAR(45) NOT NULL,

phone CHAR(15) NOT NULL,

email VARCHAR(45) CHECK(email LIKE '%\_@\_\_%.\_\_%') NOT NULL,

status VARCHAR(45) CHECK(status IN('Disabled', 'Enabled')) DEFAULT 'Enabled' NOT NULL,

comments VARCHAR (45) NULL,

primary key(memberId)

)

CREATE TABLE Staff

(

memberId CHAR(30) NOT NULL,

Position CHAR(15) NOT NULL,

Foreign Key (memberId) REFERENCES Member(memberId) ON UPDATE CASCADE ON DELETE CASCADE,

primary key(memberId)

)

CREATE TABLE Student

(

memberId CHAR(30) NOT NULL,

demeritPoints INT DEFAULT 12 NOT NULL,

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

primary key(memberId)

)

CREATE TABLE Category

(

code CHAR(15) NOT NULL,

name VARCHAR(30) NOT NULL,

description VARCHAR(30) NOT NULL,

durationInHours int NOT NULL,

PRIMARY KEY (code)

)

CREATE TABLE CatalogueSearch

(

CatalogueId CHAR(30) NOT NULL,

keywords CHAR (30) NOT NULL,

PRIMARY KEY (CatalogueId)

)

CREATE TABLE Location

(

locationId CHAR(30) NOT NULL,

room CHAR(30) NULL,

PRIMARY KEY (locationId)

)

CREATE TABLE Resource

(

resourceId CHAR(15) NOT NULL,

description VARCHAR(40) NULL,

status VARCHAR(25) CHECK(status IN('Not Loaned', 'Loaned', 'Loan Cancelled')) DEFAULT 'Not Loaned ' NOT NULL,

code CHAR(15) NOT NULL,

locationId CHAR(30) NOT NULL,

FOREIGN KEY (code) REFERENCES Category(code) ON UPDATE CASCADE ON DELETE NO ACTION,

FOREIGN KEY (locationId) REFERENCES Location(locationId) ON UPDATE CASCADE ON DELETE NO ACTION,

PRIMARY KEY (resourceId)

)

CREATE TABLE Privilege

(

privilegeId CHAR(15) NOT NULL,

description VARCHAR(45) NOT NULL,

maximumItems INT NOT NULL,

code CHAR(15) NOT NULL,

FOREIGN KEY (code) REFERENCES Category(code) ON UPDATE CASCADE ON DELETE NO ACTION,

PRIMARY KEY(privilegeId)

)

CREATE TABLE Course

(

courseId CHAR(30) NOT NULL,

name VARCHAR(30) NOT NULL,

primary key(courseId)

)

CREATE TABLE ImmoveableResource

(

resourceId CHAR(15) NOT NULL,

capacity INT NULL,

FOREIGN KEY (resourceId) REFERENCES Resource(resourceId) ON UPDATE CASCADE ON DELETE CASCADE,

primary key(resourceId)

)

CREATE TABLE MoveableResource

(

resourceId CHAR(15) NOT NULL,

name VARCHAR(30) NOT NULL,

make VARCHAR(30) NULL,

model VARCHAR(30) NULL,

year CHAR(4) NULL,

manufacturer VARCHAR(30) NULL,

assetValue MONEY NULL,

FOREIGN KEY (resourceId) REFERENCES Resource(resourceId) ON UPDATE CASCADE ON DELETE CASCADE,

PRIMARY KEY(resourceId)

)

CREATE TABLE Courseoffing

(

courseId CHAR(30) NOT NULL,

name VARCHAR(30) NOT NULL,

startDate Date NOT NULL,

endDate Date NOT NULL,

semesterOffered VARCHAR(30) CHECK(semesterOffered IN('Summer','Winter', 'Summer and Winter')) NOT NULL,

yearOffered CHAR(4) NOT NULL,

privilegeId CHAR(15) NOT NULL,

memberId CHAR(30) Not NULL,

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

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

FOREIGN KEY (courseId) REFERENCES Course(courseId) ON UPDATE CASCADE ON DELETE NO ACTION,

PRIMARY KEY (courseId)

)

CREATE TABLE Acquisition

(

acquisitionId CHAR(15) NOT NULL,

memberId CHAR(30) Not NULL,

itemName VARCHAR(45) Not NULL,

make VARCHAR(45) Not NULL,

model VARCHAR(45) Not NULL,

year CHAR(4) Not NULL,

manufacturer VARCHAR(45) NULL,

description VARCHAR(45) Not NULL,

urgency VARCHAR(45) CHECK(urgency IN('High','Low')) NULL,

status VARCHAR(15) CHECK(status IN('Approved','Denied', 'In-process')) DEFAULT 'In-process' NULL,

vendorCode VARCHAR(45) Not NULL,

price Decimal(5) Not NULL,

fundCode VARCHAR(45) Not NULL,

notes VARCHAR(45) Not NULL,

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

PRIMARY KEY (acquisitionId)

)

CREATE TABLE Loan

(

loanId INT IDENTITY(1,1) NOT NULL,

dateTimeBorrowed dateTime NOT NULL,

dateTimeReturned dateTime NULL,

dateTimeDue dateTime NOT NULL,

moveableResourceId CHAR(15) NOT NULL,

memberId CHAR(30) NOT NULL,

FOREIGN KEY (moveableResourceId) REFERENCES MoveableResource(resourceId) ON UPDATE CASCADE ON DELETE NO ACTION,

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

PRIMARY KEY (loanId)

)

CREATE TABLE Reservation

(

reservationId CHAR(45) NOT NULL,

dateTimeStart date NOT NULL,

DateTimeEnd date NOT NULL,

memberId CHAR(30) NOT NULL,

resourceId CHAR(15) NOT NULL,

FOREIGN KEY (resourceId) REFERENCES Resource(resourceId) ON UPDATE CASCADE ON DELETE NO ACTION,

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

PRIMARY KEY(reservationId)

)

INSERT INTO Member VALUES ('S0000001','Fake M one', 'Fake S one','0000000001','FakeMone@hotmail.com', 'Enabled', '')

INSERT INTO Member VALUES ('S0000002','Fake M two', 'Fake S two','0000000002','FakeMtwo@hotmail.com', 'Enabled', '')

INSERT INTO Member VALUES ('S0000003','Fake M third', 'Fake S third','0000000003','FakeMthird@hotmail.com', 'Enabled', '')

INSERT INTO Member VALUES ('S0000004','Fake M fourth', 'Fake S fourth','0000000004','FakeMfourth@hotmail.com', 'Disabled', 'Overdue fines')

INSERT INTO Member VALUES ('S0000005','Fake M fifth', 'Fake S fifth','0000000005','FakeMfifth@hotmail.com', 'Enabled', '')

INSERT INTO Member VALUES ('S0000006','Fake M sixth', 'Fake S sixth','0000000006','FakeMsixth@hotmail.com', 'Enabled', '')

INSERT INTO Member VALUES ('S0000007','Fake M seventh', 'Fake S seventh','0000000007','FakeMseventh@hotmail.com', 'Enabled', '')

INSERT INTO Member VALUES ('S0000008','Fake M eighth', 'Fake S eighth','0000000008','FakeMeighth@hotmail.com', 'Enabled', '')

SELECT \*

FROM Member;

INSERT INTO Student VALUES ('S0000001', 5)

INSERT INTO Student VALUES ('S0000002', 10)

INSERT INTO Student VALUES ('S0000003', 15)

INSERT INTO Student VALUES ('S0000004', 20)

SELECT \*

FROM Student;

INSERT INTO Staff VALUES ('S0000005','Admin')

INSERT INTO Staff VALUES ('S0000006','Tutor')

INSERT INTO Staff VALUES ('S0000007','Researcher')

INSERT INTO Staff VALUES ('S0000008','Former Staff')

SELECT \*

FROM Staff;

INSERT INTO Location VALUES ('new0001','new11')

INSERT INTO Location VALUES ('new0002','new22')

INSERT INTO Location VALUES ('new0003','new33')

SELECT \*

FROM Location;

INSERT INTO Category VALUES ('code0001','room', 'Computer room', 5)

INSERT INTO Category VALUES ('code0002','speaker', 'speakers', 10)

INSERT INTO Category VALUES ('code0003','Laptop','Desktops', 15)

INSERT INTO Category VALUES ('code0004','room','Meeting room', 20)

INSERT INTO Category VALUES ('code0005','room','Study room', 25)

INSERT INTO Category VALUES ('code0006','Camera','Cameras', 30)

SELECT \*

from Category;

INSERT INTO CatalogueSearch VALUES ('CS000001','K00000001')

INSERT INTO CatalogueSearch VALUES ('CS000002','K00000002')

INSERT INTO CatalogueSearch VALUES ('CS000004','K00000003')

INSERT INTO CatalogueSearch VALUES ('CS000005','K00000004')

SELECT \*

from CatalogueSearch;

INSERT INTO RESOURCE VALUES ('ID000001','Camera', 'Loaned','code0006','new0001')

INSERT INTO Resource VALUES ('ID000002','Speaker','Loaned','code0002','new0002')

INSERT INTO Resource VALUES ('ID000003','Laptop','Not Loaned', 'code0003','new0003')

INSERT INTO Resource VALUES ('ID000004','Meeting Room','Not Loaned','code0004','new0001')

INSERT INTO Resource VALUES ('ID000005','Computer Room', default, 'code0005','new0002')

INSERT INTO Resource VALUES ('ID000006','Study room', default,'code0006','new0003')

SELECT \*

FROM Resource;

INSERT INTO MoveableResource (resourceId, name, make, model, year, manufacturer, assetValue) VALUES ('ID000001', 'Camera', 'fakecamera', 'C001', 2020,'fake1',0001)

INSERT INTO MoveableResource (resourceId, name, make, model, year, manufacturer, assetValue) VALUES ('ID000002', 'Speaker', 'fakespeaker', 'S001', 2020,'fake2',0002)

INSERT INTO MoveableResource (resourceId, name, make, model, year, manufacturer, assetValue) VALUES ('ID000003', 'Laptop', 'fakelaptop', 'L001', 2020,'fake3',0003)

SELECT \*

FROM MoveableResource;

INSERT INTO ImmoveableResource (resourceId, capacity) VALUES ('ID000004', 5)

INSERT INTO ImmoveableResource (resourceId, capacity) VALUES ('ID000005', 10)

INSERT INTO ImmoveableResource (resourceId, capacity) VALUES ('ID000006', 15)

SELECT \*

FROM ImmoveableResource;

INSERT INTO Course VALUES ('FAKE1','Class1010')

INSERT INTO Course VALUES ('FAKE2','Class1020')

INSERT INTO Course VALUES ('FAKE3','Class1030')

INSERT INTO Course VALUES ('FAKE4','Class1040')

INSERT INTO Course VALUES ('FAKE5','Class1050')

SELECT \*

FROM Course;

INSERT INTO Acquisition VALUES ('Acq00001','S0000001','Producttwo','F2','F3', 2022,'F4', 'F5', 'high', 'Approved', 'v01', 5, 'a01', 'Order approved')

INSERT INTO Acquisition VALUES ('Acq00002','S0000005','Productthird','F22','F33', 2022,'F44', 'F55', 'low', 'In-process', 'v02', 10, 'a02', 'Order Denied')

INSERT INTO Acquisition VALUES ('Acq00003','S0000003','Productfour','F222','F333', 2022,'F444', 'F555', 'high', 'In-process', 'v03', 15, 'a03', 'Order approved')

INSERT INTO Acquisition VALUES ('Acq00004','S0000004','Productfive','F2222 ','F3333', 2022,'F4444', 'F5555', 'low', 'Denied', 'v04', 20, 'a04', 'Order Denied')

SELECT \*

FROM Acquisition;

INSERT INTO Reservation VALUES ('Res00001', '2022-1-1', '2022-12-1' , 'S0000005', 'ID000001')

INSERT INTO Reservation VALUES ('Res00002', '2022-1-1', '2022-12-1' , 'S0000005', 'ID000003')

INSERT INTO Reservation VALUES ('Res00003', getDate(), DATEADD(DAY, +14, getDate()) , 'S0000003', 'ID000002')

INSERT INTO Reservation VALUES ('Res00004', getDate(), DATEADD(DAY, +14, getDate()) , 'S0000004', 'ID000004')

INSERT INTO Reservation VALUES ('Res00005', '2022-1-1', '2022-12-1' , 'S0000005', 'ID000005')

SELECT \*

FROM Reservation;

INSERT INTO Privilege VALUES ('P0001','Laptop', 5,'code0001')

INSERT INTO Privilege VALUES ('P0002','Speaker', 10, 'code0002')

INSERT INTO Privilege VALUES ('P0003','Camera', 15,'code0003')

INSERT INTO Privilege VALUES ('P0004','Meeting room', 20, 'code0004')

SELECT \*

FROM Privilege

INSERT INTO Courseoffing VALUES ('FAKE1', 'Class1010', '2022-1-1', '2022-12-30','Summer and Winter', '2022','P0001', 'S0000001')

INSERT INTO Courseoffing VALUES ('FAKE2', 'Class1020', '2022-1-1', '2022-12-30','Summer and Winter', '2022', 'P0002', 'S0000002')

INSERT INTO Courseoffing VALUES ('FAKE3', 'Class1030', '2022-1-1', '2022-12-30','Summer and Winter', '2022','P0003', 'S0000003')

INSERT INTO Courseoffing VALUES ('FAKE4', 'Class1040', '2021-12-1', '2022-2-28','Summer','2022', 'P0004', 'S0000004')

INSERT INTO Courseoffing VALUES ('FAKE5', 'Class1050', '2022-6-1', '2022-8-31','Winter','2022','P0004', 'S0000002')

SELECT \*

FROM Courseoffing

INSERT INTO Loan(memberId, dateTimeBorrowed, dateTimeDue, dateTimeReturned, moveableResourceId) VALUES ('S0000001', getDate(), DATEADD(HOUR, +336, getDate()) , DATEADD(DAY, +9, getDate()) , 'ID000001')

INSERT INTO Loan(memberId, dateTimeBorrowed, dateTimeDue, dateTimeReturned, moveableResourceId) VALUES ('S0000002', getDate(), DATEADD(HOUR, +336, getDate()) , DATEADD(DAY, +13, getDate()) , 'ID000002')

INSERT INTO Loan(memberId, dateTimeBorrowed, dateTimeDue, dateTimeReturned, moveableResourceId) VALUES ('S0000003', getDate(), DATEADD(HOUR, +336, getDate()) , DATEADD(DAY, +6, getDate()) , 'ID000003')

SELECT \*

FROM Loan;

--Question 1

SELECT m.name

FROM Courseoffing s join Member m ON s.memberId = m.memberId

WHERE s.courseId = 'FAKE2';

--Question 2

SELECT SUM(p.maximumItems) AS 'sum Of Speaker Privileges'

FROM Privilege p, Courseoffing c, Courseoffing m

WHERE p.code = (SELECT code FROM Category cc WHERE cc.name = 'Speaker')

AND p.privilegeId = c.privilegeId

AND p.privilegeId = c.privilegeId AND c.courseId = m.CourseId

AND m.memberId = (SELECT m.memberId FROM Member m WHERE m.name = 'Fake M two')

--Question 3

select m.name,m.phone,

count(DISTINCT r.reservationId) as 'Reservations total'

from member m

join Acquisition a on m.memberId = a.memberId

join Reservation r on m.memberId = r.memberId

where year = '2022' and m.memberId = 'S0000005'

group by m.name,m.phone,a.acquisitionId

--Question 4

SELECT DISTINCT(m.name) FROM Member m

JOIN Student s ON m.memberId = s.memberId

JOIN Loan l ON s.memberId = l.memberId JOIN MoveableResource m2 ON l.moveableResourceId = m2.resourceId

JOIN Resource r ON m2.resourceId = r.resourceId JOIN Category c ON r.code = c.code

WHERE YEAR(l.dateTimeBorrowed) = '2022' AND c.name = 'Camera' AND m2.model = 'C001'

--Question 5

SELECT TOP (1) name, l.moveableResourceId FROM MoveableResource m RIGHT JOIN Loan l ON m.resourceId = l.moveableResourceId

WHERE MONTH(l.dateTimeBorrowed) = MONTH(GETDATE())

GROUP BY name, l.moveableResourceId ORDER BY COUNT(l.moveableResourceId) DESC;