COMP1140: Database and Information Management

Assignment 2: SCS Resource Access Database Design Project
Logical Database design

Reflection summary of assignment 2.

Looking back on Assignment 2, I did a better job then in assignment 1. I was expecting this result as a lot of work was put into getting assignment 2 correct. The business rules have been revised and structured into their respective groupings and very little negative feedback regarding them and data dictionary. I had a better understand of the EER model but still not perfect yet. I did a good job of mapping relational model with EER but I need to work on and improve my normalisation as several errors were pointed out to me and this is the area where most of my feedback came from. Fixing the normalisation component of the report will be my main concern. Taking all of this into consideration my database for assignment 3 is working well and I expect similar marks for this assignment.

Data Requirements Revised content

Acquisition Service

The acquisition module will provide a means for admin staff to capture and respond to requests for new and/or updated versions of resources made by members that they would like to see added to the SCS resource collection. The data being stored for an acquisition will need to be the following

- Acquisition Id
- Member Id
- Item Name
- Make
- Model
- Year
- Manufacturer
- Description
- Urgency
- Status
- VendorCode
- Price
- FundCode
- Notes

Category

The category is a collection of a specified resource. Eg category of cameras or a category of software. The data being stored in category is

- Code
- Name
- Description
- Duration in hours

Course

Student members must be enrolled in courses offered by university to be able to reserve and borrow the resources. The data being stored on course is

- Course id
- name

Enrolment

Enrolment is the process in which a student must be enrolled into a course. The data being stored on enrolment is

- Course id
- Name
- Start Date
- End date
- Semester offered
- Year offered
- Privilege id
- Member id

Immovable Resources:

Immoveable resources are study rooms, studios, and laboratories. The data being stored on these items are

- Resource id
- Capacity of people allowed in the room.

Loan Service

The SCS resource collection consists of a variety of items available for loan and physical spaces available for booking by University of Sunshine staff and students, known as members. Through the loan service module, admin staff will manage the issuing and returning of resources to borrowers and the booking of rooms. The data being stored in the loan service module will be

- Loan Id.
- date and time borrowed.
- date and time due.
- date and time returned.
- Moveable resource id
- Member id

Location

All resources have a specified location. The data being stored on a resource's location is

- Location id
- Room
- Level
- Building
- Campus

Members

The SCS service is offered to staff who are employed by the university or to students who are enrolled in courses offered by the university. These students and staff are known members. The following data being stored on members is

- Member id
- name
- Address which consists of street name, street number, suburb, and postcode.
- Phone number
- Email
- Status
- comments

Moveable Resources:

Moveable resources are Speakers, phones, cameras. The data being stored on these items are

- name
- Make
- Model
- Year
- Manufacturer
- Asset value

Privilege

Privileges are based on what type of member you are. If you are a student member you are granted privileges based on what courses you are enrolled in, this will determine what resources you can reserve and borrow. If you are a staff member you are allowed to reserve and borrow resources without a limit. The data being stored on privileges is

- Privilege id
- Description
- Maximum Items
- Code

Reservation Service

The reservation module will be used to keep track of all resources requested by members and access will be based on availability and privileges. This will also be a 'first-come-first-serve' basis. The data being stored in the reservation module is

- Reservation id
- Date and time start
- Date and time end
- Member id
- Resource id

Resources

The SEC department has many resources to be borrowed or used by it's members either being physical like cameras or speakers or online resources such as software. The following data being stored on resources is

- Resource id
- Description
- Status which would be for example on loan, lost, damaged and on order.

Student

Student is a person who will enrol into the university to undertake courses. The data being stored on a student will be

- points

Staff

Staff is a person who is employed at the university. The data being stored on staff is

- Is position

Transaction Requirements:

Data Manipulation Operations

- Update/delete the details of admin staff
- Update/delete the details of members
- Update/delete the details of resource categories
- Update/delete the details of loans
- Update/delete the details of acquisition requests
- Update/delete the details of reservations
- Update/delete the details of moveable resources
- Update/delete the location of a resource
- Update/delete the status of an item
- Update/delete members privileges
- Update/delete the catalogue

Data Queries

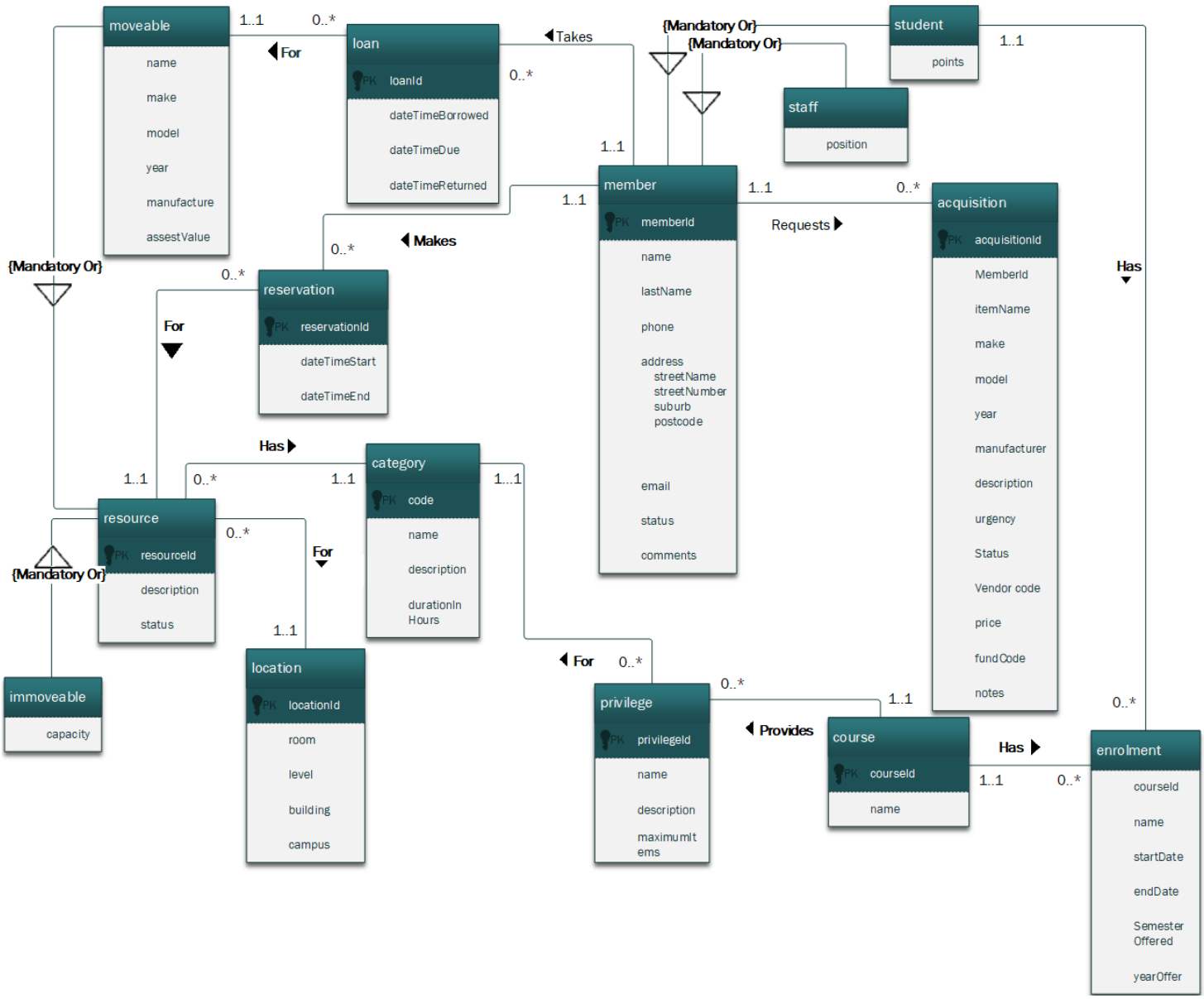
- List details of all current members enrolled in or teaching a particular course
- List all current non active members (including any information regarding why status is non active which will be found in the members comments field)
- Identify details of current courses on offer (displaying the conditions under which resources can be loaned based on the course requirements)
- List all items due for returning today, by rank of the greatest number of loans to least
- Identify which rooms have bookings over the next seven days
- Display a member's outstanding reservations
- List all acquisitions purchased in the past three months
- List all resource available in a category, e.g. cameras
- Display a member's loan privileges, e.g. how many resources can be loaned at any given time
- Display an immovable resource availability and capacity
- List all available rooms on a particular date/time
- List information about members privileges

SCS Business Rules

Business Rules:

1. **Student Member**
    - A student's borrowing privileges are taken away when they finish their enrolled course.
    - The student members current status is set to disabled.
2. **Maximum number of items loaned or reserved at any one time**
    - A student member cannot borrow, or reserve more than the maximum number of items specified at any given time
    - Staff members have priority over student members when loaning resources
3. **Late returns**
    - Student members have by default 12 points at the beginning of their course.
    - For every day that a resource is overdue a three point penalty will be given.
    - Once a student has all of their points deducted the status of that member is disabled, and all rights and privileges are cancelled, including borrowing and reserving.
    - A system administrator has the right to amend and restore a member's points.
4. **Reservations**
    - A member can reserve an immoveable resource if available for a set period of time.
    - A reserved item is cancelled if it is not collected by the required due date causing the requesting member to incur a one point deduction of their total points.
    - Reservations are on a first come, first serve basis.
    - Administrators reserve the right to cancel any reservation
5. **Borrowing/Reservation Periods**
    - The duration of the borrowing/reservation periods being days or hours are determined by the category to which the item belongs
    - No reservations with the same resource id should overlap with their dates or times.

Revised EER From Assignment 2



Data dictionary Revised

Entity Types			
Entity Name	**Entity Description**	**Aliases**	**Occurrence**
Students	Student is an enrolled person of the university and is a Subclass of members.	Member	A new student is created when a student enrols into the university and commences classes.
Staff	A university staff is a person who is employed by the university. University staff is a subclass of members.	Member	University Staff is created when a University Staff becomes employed by the university and commences work.
Member	Member is a person who uses the SCS department services.	Member	A member is created when A new student is created when a student whose details are not in the database goes to use the service.
Loans	Loans is a record of a member borrowing a resource.	Borrow	A loan is created when a member borrows a resource.
Acquisitions	An Acquisition is a record of the process of a request and purchasing of an asset or object brought by the SCS department used for learning and development of university and SCS members.	Request	An acquisition is created when a member submits a request.
Resources	Resource is a physical item that can be loaned to members.	Moveable/Immoveable resource	A resource is created when the SCS department has purchased or obtained the item and it is ready for member use.

Immoveable resources	An immovable resource is a type of study room or studio that can be booked for use.	Room	Immoveable resource is created when the SCS department has a new physical structure or area for member use and is ready to be used by members.
Moveable resources	A moveable resource is a portable item that can be taken from the SCS department by a member.	Item	A moveable resource is created when the SCS department has purchased an item and it is ready to be loaned to members.
Reservations	A reservation is a record of the request submitted by a member to the SCS department ask to be allocated a resource over a defined time period.	Hold	A reservation is created when a member submits a reservation request.
Category	A category is a record of a collection of like resources defined by the item's attributes and functions. Example phone or camera.	Class	A category is created when a physical item becomes property of the SCS department. And it is the first of its type.
Privilege	Borrowing and access rights given to SCS members	Right	Members borrowing rights and access to resources
Course	A course offered by the University.	Subject	A course is created when the university roles out approved course syllabus
Location	Unique location with university.	Place	A position within a building which is located on a campus. A location is created when a new resource is acquired.

Attributes							
Entity Name	**Attributes**	**Description**	**Data type & length**	**Nulls**	**Multi-Valued**	**Derived**	**Default**
Acquisition	acquisitionId	Item identification number or name	CHAR(15)	N	N	N	15
Acquisition	ItemName	Name of the item	VARCHAR(30)	N	N	N	
Acquisition	make	Make of item	VARCHAR(30)	N	N	N	
Acquisition	model	Model of item	VARCHAR(30)	N	N	N	
Acquisition	year	Year item was released	Year CHAR (4)	N	N	N	
Acquisition	manufacturer	Manufacturer of item	VARCHAR(30)	Y	Y	N	
Acquisition	description	Description for item	VARCHAR(30)	N	N	N	
Acquisition	urgency	Priority of the request	VARCHAR(30)	N	N	N	
Acquisition	vendorCode	Code of the vendor	VARCHAR(30)	N	N	N	
Acquisition	Price	Cost of the item	Decimal(5)	N	N	N	5
Acquisition	fundCode	Code of the fund	VARCHAR(30)	N	N	N	
Acquisition	status	Current status of an acquisition	VARCHAR(30)	N	N	N	
Acquisition	notes	Any required comments about the item	VARCHAR(30)	N	N	N	
Category	code	The category identification code for the category a resource belongs	CHAR(15)	N	N	N	15
Category	name	The name of the category a resource belongs to.	VARCHAR (30)	N	N	N	
Category	description	The description of the category a resource belongs to.	VARCHAR (30)	N	N	N	
Category	duration	Time allowed	VARCHAR int	N	N	N	

Course	courseId	Identification number or name of the course	CHAR(30)	N	N	N	15
Course	name	Name of the course	VARCHAR(30)	N	N	N	
Enrolment	courseId						
Enrolment	Name	Name of the course	VARCHAR(30)	N	N	N	
Enrolment	startDate	Starting date of the course	date()				
Enrolment	endDate	End date of the course	date()	N	N	N	
Enrolment	semesterOffered	Semester the course is offered	VARCHAR(30)	N	N	N	
Enrolment	yearOffered	Year the course is running	CHAR(4)	N	N	N	
Immoveable	capacity	The amount of people allowed in a room	Int()	Y	N	N	
Loan	loanId	Identification number/name of the loan	int	N	N	N	15
Loan	dateTimeBorrowed	Date and time of the loan	dateTime()	N	N	N	
Loan	dateTimeDue	Date and time item is due back	dateTime()	N	N	Y	
Loan	dateTimeReturned	The date and time recorded of item being returned	dateTime()	N	N	N	
Location	locationId	Identification of the item's location	CHAR(15)	N	N	N	15
Location	room	The room the item is located	CHAR(15)	Y	N	N	15
Location	level	The level of the building the item is located	VARCHAR(30)	Y	N	N	
Location	building	The building the item is located	VARCHAR(30)	Y	N	N	
Location	campus	The campus the item is located	VARCHAR(30)	N	N	N	
Member	memberId	Identification number/name of the member	CHAR(15)	N	N	N	15
Member	name	The first name of the member	VARCHAR(30)	N	N	N	
Member	address						
Member	streetName	The street name of the members address	VARCHAR(30)	N	N	N	

Member	streetNumber	The street number of the members address	Int()	N	N	N	
Member	suburb	The suburb of the members address	VARCHAR(30)	N	N	N	
Member	postcode	The postcode of the members address	VARCHAR(30)	N	N	N	
Member	phone	The phone number of the member	VARCHAR(10)	N	N	N	10
Member	email	The email address of the member	VARCHAR(30)	N	N	N	
Member	status	The members status	VARCHAR(30)	N	N	Y	
Member	comments	Any comments regarding the member	VARCHAR(30)	Y	N	N	
Moveable	name	The name of the moveable item	VARCHAR(30)	N	N	N	
Moveable	make	The make or brand name of the moveable item	VARCHAR(30)	Y	N	N	
Moveable	model	Model id of the moveable item	VARCHAR(30)	Y	N	N	
Moveable	year	The year the moveable item was released	CHAR(4)	Y	N	N	
Moveable	manufacturer	Name of the Manufacturer who made the moveable item	VARCHAR(30)	Y	N	N	
Moveable	assetsValue	Current market value of the moveable item	Mone)	Y	N	N	
Privilege	privilegeId	Identification number/name of a privilege	CHAR(15)	N	N	N	15
Privilege	name	Name of the privilege	VARCHAR(30)	N	N	N	
Privilege	description	Description of the given privilege	VARCHAR(30)	N	N	N	
Privilege	maximumItems	Most items that a particular privilege can borrowed	Int()	N	N	N	

Reservation	reservationId	Identification number/name of a reservation	CHAR(15)	N	N	N	15
Reservation	dateTimeStart	Date and time a reservation began	date()	N	N	N	
Reservation	dateTimeEnd	Date and time a reservation ended	date()	N	N	N	
Resource	resourceId	Identification number/name of a resource	CHAR(15)	N	N	N	15
Resource	description	Description of the resource	VARCHAR(30)	Y	N	N	
Resource	Status	Current status of the resource	VARCHAR(30)	N	N	N	
Staff	position	Job position	CHAR(15)	N	N	N	15
Student	points	Points allocated to student	Int()	N	N	N	12

Relationship Types				
Entity name	**Multiplicity**	**Relationship**	**Multiplicity**	**Entity name**
Course	1..1	Provides	0..*	Privileges
Course	1..1	Has	0..*	Enrolment
Immoveable		{Mandatory Or}		Resource
Loan	0..*	For	1..1	Moveable
Location				
Member	1..1	Requests	0..*	Acquisition
Member	1..1	Takes	0..*	Loan
Member	1..1	Makes	0..*	Reservation
Moveable		{Mandatory Or}		Resource
Privilege	0..*	For	1..1	Category
Reservation	0..*	For	1..1	Resource
Resource	0..*	For	1..1	Location
Resource	0..*	Has	1..1	Category
Staff		{Mandatory Or}		Member
Student		{Mandatory Or}		Member
Student	1..1	Has	0..*	Enrolment

Revised Relational model mapped with EER From Assignment 2

Acquisition(AcquisitionId, itemName, make, model, year, manufacturer, description, urgency, status, vendorCode, price, fundCode, notes)

Primary Key AcquisitionID

Foreign Key memberId **references** Member(memberId)

ON UPDATE CASCADE, ON DELETE CASCADE

Category (Code, name, description, durationInHours)
Primary Key Code

Course (courseId, name)
Primary Key coursed

Enrolment (courseId, name, startDate, endDate, semesterOffered, yearOffered, privilegeId, memberId)

Primary course Id

Foreign Key courseId **references** course(courseId)

ON UPDATE CASCADE, ON DELETE NO ACTION

Foreign Key memberId **references** Member(memberId)

ON UPDATE CASCADE, ON DELETE NO ACTION

Foreign Key privilegeId **references** Privilege(privilegeId)

ON UPDATE CASCADE, ON DELETE NO ACTION

Immovable Resource(resourceId, description, status, capacity)

Primary Key resource Id

Foreign Key resource Id **references** Resource(resourceId)

ON UPDATE CASCADE, ON DELETE CASCADE

Loan (loanId, dateTimeBorrowed, dateTimeDue, dateTimeReturned, resourceId, memberId)

Primary Key loanId

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

Location (LocationId, Room, level, building, campus)

Primary Key locationId

Member (memberId, firstName, lastName , address, streetName , streetNumber, suburb, postcode, phone, email, status, comments)

Primary Key memberId

ON UPDATE NO ACTION, ON DELETE NO ACTION

Movable (resourceId, name, make, model, year, manufacturer, assetValue)

Primary key resourceId

Foreign Key resourceId **references** Resource(resourceId)

ON UPDATE CASCADE, ON DELETE CASCADE

Privilege (privilegeId, name, description, maxItems)

Primary Key privilegeId

Foreign Key courseId **references** course(courseId)

ON UPDATE CASCADE, ON DELETE CASCADE, NO ACTION

Reservation (reservationId, dateTimeStart, dateTimeEnd)

Primary Key reservationId

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

Resource(resourceId, description, status)

Primary Key resource Id

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

Staff (memberId, firstName, lastName , address, streetName , streetNumber, suburb, postcode, phone, email, status, comments, position)

Primary Key memberId

Foreign Key memberId **references** Member(memberId)

ON UPDATE CASCADE, ON DELETE CASCADE.

Student (memberId, firstName, lastName , address, streetName , streetNumber, suburb, postcode, phone, email, status, comments, points)

Primary Key memberId

Foreign Key memberId **references** Member(memberId)

Foreign Key course **references** Course(courseId)

ON UPDATE CASCADE, ON DELETE CASCADE

Relation Normalization

Example 1:

valid DBDL form before normalisation

Student (memberId, name , address, streetName , streetNumber, suburb, postcode, phone, email, status, comments, points)

Primary Key memberId

Foreign Key memberId **references** Member(memberId)

ON UPDATE CASCADE, ON DELETE CASCADE

Normalisation process:

Student (memberId, name, address, streetName , streetNumber, suburb, postcode, phone, email, status, comments, points)

FD1: memberId -> memberId, name , address, streetName , streetNumber, suburb, postcode, phone, email, status, comments, points

R is the relation of student

R: (memberId, firstName, lastName , address, streetName , streetNumber, suburb, postcode, phone, email, status, comments points)

FD2: suburb -> postcode

Suburb and postcode are known as a functional dependency. Without a suburb you do not have a postcode.

R - postcode: (memberId, Name, Email, Address, City, Status, Phone Number, Comments, points)

postcode is removed from R to decompose the functional dependency.

The dependency is now in BCNF.

After normalisation in DBDL form

NStudent (memberId, firstName, lastName , address, streetName , streetNumber, suburb, phone, email, status, comments, points)

Primary Key memberId

Foreign Key memberId **references** Member(memberId)

ON UPDATE CASCADE, ON DELETE CASCADE

Example 2:

valid DBDL form before normalisation

Staff (memberId, name , address, streetName , streetNumber, suburb, postcode, phone, email, status, comments, position)

Primary Key memberId

Foreign Key memberId **references** Member(memberId)

ON UPDATE CASCADE, ON DELETE CASCADE.

Normalisation process:

Staff (memberId, name , address, streetName , streetNumber, suburb, postcode, phone, email, status, comments, position)

FD1: memberId -> memberId, firstName, lastName , address, streetName , streetNumber, suburb, postcode, phone, email, status, comments, position

R is the relation of staff

R: (memberId, firstName, lastName , address, streetName , streetNumber, suburb, postcode, phone, email, status, comments position)

FD2: suburb -> postcode

suburb and postcode are known as a functional dependency. Without a suburb you do not have a postcode.

R - postcode: (memberId, Name, Email, Address, City, Status, Phone Number, Comments, asAdmin)

postcode is removed from R to decompose the functional dependency.

postcode(suburb, postcode)

The dependency is now in BCNF

After normalisation in DBDL form

Staff (memberId, firstName, lastName , address, streetName , streetNumber, suburb, phone, email, status, comments, asAdmin)

Primary Key memberId

Foreign Key memberId **references** Member(memberId)

ON UPDATE CASCADE, ON DELETE CASCADE.

Example 3

valid DBDL form before normalisation

Acquisition(AcquisitionId, itemName, make, model, year, manufacturer, description, urgency, status, vendorCode, price, fundCode, notes)

Primary Key AcquisitionId

Foreign Key memberId **references** Member(memberId)

ON UPDATE CASCADE, ON DELETE CASCADE

Normalisation process:

Acquisition(AcquisitionId, itemName, make, model, year, manufacturer, description, urgency, status, vendorCode, price, fundCode, notes)

FD1: Acquisition -> AcquisitionId, memberId itemName, make, model, year, manufacturer, description, urgency, status, vendorCode, price, fundCode, notes

All the attributes inside of Acquisition are known as R

R: Acquisition(AcquisitionId, itemName, make, model, year, manufacturer, description, urgency, status, vendorCode, price, fundCode, notes)

FD2: vendorCode -> price, fundCode

vendorCode and price, fundCode are known as a functional dependency. Without a vendorCode you do not have a price and fundCode.

R – price, fundCode (AcquisitionId, memberId itemName, make, model, year, manufacturer, description, urgency, vendorCode, notes)

price and fundCode is removed from R to decompose the functional dependency.

vendorCode(vendorCode, price, fund code)

The functional dependency is now in BCNF

After normalisation in DBDL form

Acquisition(AcquisitionId, itemName, make, model, year, manufacturer, description, urgency, status, vendorCode, notes)

vendorCode(price, fundCode)

Primary Key AcquisitionId

Foreign Key memberId **references** Member(memberId)

ON UPDATE CASCADE, ON DELETE CASCADE

SQL scripts that create the normalised SCS database, including all necessary tables, SQL statements satisfying the transaction requirements and 6 Queries required to be ran.

```
drop table Enrolment
drop table Course
drop table Privilege
drop table Acquisition
drop table Reservation
drop table Loan
drop table MoveableResource
drop table ImmoveableResource
drop table Resource
drop table Location
drop table Category
drop table Staff
drop table Student
drop table Member
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 Location
(
	locationId CHAR(15) NOT NULL,
	room CHAR(15) NULL,
	level VARCHAR(30) NULL,
	building VARCHAR(30) NOT NULL,
	campus VARCHAR(30) CHECK(campus IN('Newcastle','Ourimbah','Sydney','Coffs Harbour')) NOT NULL,
	PRIMARY KEY (locationId)
)
```

```
CREATE TABLE Resource
(
	 resourceId CHAR(15) NOT NULL,
	description VARCHAR(30) NULL,
	status VARCHAR(20) CHECK(status IN('Not collected', 'Collected', 'Cancelled')) DEFAULT 'Not collected'
	NOT NULL,
	code CHAR(15) NOT NULL,
	locationId CHAR(15) 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 Member
(
	memberId CHAR(15) NOT NULL,
	name VARCHAR(30) NOT NULL,
	streetName VARCHAR(30) NOT NULL,
	streetNumber INT NOT NULL,
	suburb VARCHAR(30) NOT NULL,
	postCode VARCHAR(30) NOT NULL,
	phone CHAR(10) NOT NULL,
	email VARCHAR(30) CHECK(email LIKE '%_@__%.__%') NOT NULL,
	status VARCHAR(30) CHECK(status IN('Disabled', 'Active')) DEFAULT 'Active' NOT NULL,
	comments VARCHAR (30) NULL,
	primary key(memberId)
)
CREATE TABLE Student
(
	memberId CHAR(15) NOT NULL,
	points INT DEFAULT 12 NOT NULL,
	FOREIGN KEY (memberId) REFERENCES Member(memberId) ON UPDATE CASCADE ON DELETE CASCADE,
	primary key(memberId)
)
```

```
CREATE TABLE Staff
(
	memberId CHAR(15) NOT NULL,
	Position CHAR(15) NOT NULL,
	Foreign Key (memberId) REFERENCES Member(memberId) ON UPDATE CASCADE ON DELETE CASCADE,
	primary key(memberId)
)

CREATE TABLE Course
(
	courseId CHAR(30) NOT NULL,
	name VARCHAR(30) NOT NULL,
	primary key(courseId)
)

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 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 Privilege
(
	privilegeId CHAR(15) NOT NULL,
	description VARCHAR(30) 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 Enrolment
(
	courseId CHAR(30) NOT NULL,
	name VARCHAR(30) NOT NULL,
	startDate Date NOT NULL,
	endDate Date NOT NULL,
	semesterOffered VARCHAR(30) CHECK(semesterOffered IN('1','Winter','2','Summer', '1 AND 2')) NOT NULL,
	yearOffered CHAR(4) NOT NULL,
	privilegeId CHAR(15) NOT NULL,
	memberId CHAR(15) 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(15) Not NULL,
	itemName VARCHAR(30) Not NULL,
	make VARCHAR(30) Not NULL,
	model VARCHAR(30) Not NULL,
	year CHAR(4)  Not NULL,
	manufacturer VARCHAR(30) NULL,
	description		VARCHAR(30)	Not NULL,
	urgency  VARCHAR(30)	CHECK(urgency IN('High','Low')) NULL,
```

```
	status	VARCHAR(10)	CHECK(status IN('Pending','Approved', 'Denied')) DEFAULT 'Pending' NULL,
	vendorCode VARCHAR(30) Not NULL,
	price Decimal(5) Not NULL,
	fundCode VARCHAR(30) Not NULL,
	notes VARCHAR(30) 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,
	dateTimeDue	dateTime NOT NULL,
	dateTimeReturned dateTime NULL,
	moveableResourceId CHAR(15) NOT NULL,
	memberId CHAR(15) NOT NULL,
	FOREIGN KEY (moveableResourceId) REFERENCES MoveableResource(resourceId) ON UPDATE CASCADE O
	N DELETE NO ACTION,
	FOREIGN KEY (memberId) REFERENCES Member(memberId) ON UPDATE CASCADE ON DELETE NO ACTION,

	PRIMARY KEY (loanId)
)

CREATE TABLE Reservation
(
	reservationId CHAR(15) NOT NULL,
	dateTimeStart date NOT NULL,
	DateTimeEnd date NOT NULL,
	memberId CHAR(15) 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 Category VALUES ('cam10287','Camera', 'Camera types', 14)

INSERT INTO Category VALUES ('bos1237','speaker', 'Bluetooth speakers', 14)

INSERT INTO Category VALUES ('mthb002615','book','Math books', 14)

INSERT INTO Category VALUES ('mr1264','room','Meeting room', 3)

INSERT INTO Category VALUES ('srm1304','room','Study room', 3)

INSERT INTO Category VALUES ('st2003','room','Studio room', 4)


SELECT *

FROM Category;


INSERT INTO Location VALUES ('syd1001','blm1406',1,'blossom','Sydney')

INSERT INTO Location VALUES ('new0785','lsm100',2,'math','Newcastle')

INSERT INTO Location VALUES ('our1501','es1206',1,'engineering','Ourimbah')

SELECT *

FROM Location;


INSERT INTO RESOURCE VALUES ('Cam01874','Sony Camera', 'Collected','cam10287','new0785')

INSERT INTO Resource VALUES ('Spkr12031','Bose bluetrooth speaker','Collected','bos1237','Syd1001')

INSERT INTO Resource VALUES ('mthBook101','math basics textBook','Not collected', 'mthb002615','our1501')

INSERT INTO Resource VALUES ('SydSR001','Study room for students', default,'srm1304','Syd1001')

INSERT INTO Resource VALUES ('OurMR002','Meeting Room', default, 'mr1264','our1501')

INSERT INTO Resource VALUES ('newMR003','Studio room', default,'st2003','new0785')

SELECT *

FROM RESOURCE;

INSERT INTO Member VALUES ('S3334685','Jill Smith', 'Fake Street', 37, 'Lake
Mac','2259','0412762918','JSmith82@Gmail.com', 'Active', '')

INSERT INTO Member VALUES ('S3331125','Dean Hughes', 'Leggee Place', 145, 'Green
Point','2250','0417198473','Deano27@Gmail.com', 'Active', 'None')

INSERT INTO Member VALUES ('S3330695','Matthew Parker', 'Timmins Avenue', 145, 'Birmingham
Gardens','2287','0482657361','PraxM@Gmail.com', 'Disabled', 'None')

INSERT INTO Member VALUES ('S3334463','John Coyle', 'Boulvard Street', 24, 'Surry Hills',
'2002','0411673609','JohnyCoyleM@Gmail.com', 'Active', 'Fines due')

INSERT INTO Member VALUES ('S3332275','Chris Watson', 'SunShine Street', 6, 'Coffs
Harbour','2450','0465283760','CDW@Gmail.com', 'Disabled', 'None')

INSERT INTO Member VALUES ('S3330047','Bryce Harvard', 'Anita Place', 109, 'AdamsTown',
'2587','0428871652','HarvardB84@Gmail.com', 'Active', '')

INSERT INTO Member VALUES ('S3335569','Alex Griffin', 'Plateau Street', 334, 'Springfield',
'2265','0418726345','Griffin34@Gmail.com', 'Active', '')
```

```
INSERT INTO Member VALUES ('S3330052','Bree Sherd', 'Latern Street', 97, 'HomeBush',
'2025','0410756734','BreeSherd@Gmail.com', 'Active', '')

SELECT *
FROM Member;

INSERT INTO Student (memberId, points) VALUES ('S3331125', default)
INSERT INTO Student (memberId, points) VALUES ('S3334463', 9)
INSERT INTO Student (memberId, points) VALUES ('S3330047', default)
INSERT INTO Student (memberId, points) VALUES ('S3330052', 6)

SELECT *
FROM Student;

INSERT INTO Staff VALUES ('S3334685','Lecture')
INSERT INTO Staff VALUES ('S3330695','Lab Tutor')
INSERT INTO Staff VALUES ('S3332275','Admin')
INSERT INTO Staff VALUES ('S3335569','Former Staff')

SELECT *
FROM Staff;

INSERT INTO Course VALUES ('ENG2500','Sustainable Engineering')
INSERT INTO Course VALUES ('SENG1120','Data Structures')
INSERT INTO Course VALUES ('COMP1140','Database Management')
INSERT INTO Course VALUES ('MATH1110','Mathamatics for Engineering')
INSERT INTO Course VALUES ('HUBS1410','Medical terminalogy')

SELECT *
FROM Course;
```

```
INSERT INTO MoveableResource (resourceId, name, make, model, year, manufacturer, assetValue) VALUES
('Cam01874', 'camera1003', 'sony camera', '1522ci', 2017,'sony',1300)

INSERT INTO MoveableResource (resourceId, name, make, model, year, manufacturer, assetValue) VALUES
('Spkr12031', 'speaker1609', 'bose camera', '2270', 2019,'bose',1495)

INSERT INTO MoveableResource (resourceId, name, make, model, year, manufacturer, assetValue) VALUES
('mthBook101', 'mathBasic', 'dep edu', 'ver 3', 2015,'dep edu',345)


SELECT *

FROM MoveableResource;


INSERT INTO ImmoveableResource (resourceId, capacity) VALUES ('SydSR001', 15)

INSERT INTO ImmoveableResource (resourceId, capacity) VALUES ('OurMR002', 12)

INSERT INTO ImmoveableResource (resourceId, capacity) VALUES ('newMR003', 6)


SELECT *

FROM ImmoveableResource;


INSERT INTO Privilege VALUES ('PrMthb002615','Mathamatics Books', 10,'mthb002615')

INSERT INTO Privilege VALUES ('PrEng0023','sony Camera', 1, 'cam10287')

INSERT INTO Privilege VALUES ('PrBos1237','Speaker', 2,'bos1237')

INSERT INTO Privilege VALUES ('PrMR1264','Meeting room', 2, 'mr1264')


SELECT *

FROM Privilege


INSERT INTO Enrolment VALUES ('SENG1120', 'Data Structures', '2020-4-20', '2020-7-20','1 AND 2',
'2020','PrBos1237', 'S3331125') -- dean,

INSERT INTO Enrolment VALUES ('COMP1140', 'Database Management', '2020-4-20', '2020-7-20','1', '2020',
'PrEng0023', 'S3334463')

INSERT INTO Enrolment VALUES ('ENG2500', 'Sustainable Engineering', '2020-7-20', '2020-11-20','2',
'2020','PrMR1264', 'S3330047')

INSERT INTO Enrolment VALUES ('Hubs1410', 'Medical terminalogy', '2020-7-20', '2020-11-20','1 AND 2','2020',
'PrBos1237', 'S3330052')

INSERT INTO Enrolment VALUES ('Math1110', 'Mathamatics for Engineering', '2020-4-20', '2020-7-
20','Summer','2021','PrMthb002615', 'S3330052')

SELECT *

FROM Enrolment
```

```
INSERT INTO Acquisition (acquisitionId, memberId, itemName, make, model, year, manufacturer, description, urgency, status, vendorCode, price, fundCode, notes) VALUES ('Acq10004','S3332275','Pencils','HB','HB2', 2020,'Officeworks', 'lead pencils', 'low', 'Approved', 'ven9889675', 24.50, 'fc356475', 'Order approved')

INSERT INTO Acquisition (acquisitionId, memberId, itemName, make, model, year, manufacturer, description, urgency, status, vendorCode, price, fundCode, notes) VALUES ('Acq10001','S3334685','headphones','1790E','JBL', 2019,'JBL', 'Bluetooth Headphones', 'low', 'pending', 'ven1140594', 730, 'fc20034', 'N/A')

INSERT INTO Acquisition (acquisitionId, memberId, itemName, make, model, year, manufacturer, description, urgency, status, vendorCode, price, fundCode, notes) VALUES ('Acq10002','S3330695','Glasses','wrkSfe2378','SafeHouse', 2018,'SafeHouse', 'Saftey glasses', 'high', 'approved', 'ven1986746', 120, 'fc678473', 'saftey glasses engineering')

INSERT INTO Acquisition (acquisitionId, memberId, itemName, make, model, year, manufacturer, description, urgency, status, vendorCode, price, fundCode, notes) VALUES ('Acq10003','S3335569','Laptop','Surface pro ','pro seven', 2020,'Microsoft', 'portable laptop/tablet', 'high', 'Denied', 'ven1465096', 1700, 'fc361524', 'denied Not current employee')

SELECT *

FROM Acquisition;


INSERT INTO Loan(memberId, dateTimeBorrowed, dateTimeDue, dateTimeReturned, moveableResourceId) VALUES ('S3331125', getDate(), DATEADD(HOUR, +336, getDate()) , DATEADD(DAY, +9, getDate()) , 'Cam01874')

INSERT INTO Loan(memberId, dateTimeBorrowed, dateTimeDue, dateTimeReturned, moveableResourceId) VALUES ('S3334463', getDate(), DATEADD(HOUR, +336, getDate()) , DATEADD(DAY, +13, getDate()) , 'Spkr12031')

INSERT INTO Loan(memberId, dateTimeBorrowed, dateTimeDue, dateTimeReturned, moveableResourceId) VALUES ('S3330047', getDate(), DATEADD(HOUR, +336, getDate()) , DATEADD(DAY, +6, getDate()) , 'mthBook101')

SELECT *

FROM Loan;


INSERT INTO Reservation VALUES ('Res10001', '2019-6-19', '2019-7-3' , 'S3334685', 'Cam01874')

INSERT INTO Reservation VALUES ('Res10002', '2019-5-4', '2019-5-18' , 'S3330047', 'mthBook101')

INSERT INTO Reservation VALUES ('Res10003', getDate(), DATEADD(DAY, +14, getDate()) , 'S3331125', 'Spkr12031')

INSERT INTO Reservation VALUES ('Res10004', getDate(), DATEADD(DAY, +14, getDate()) , 'S3331125', 'OurMR002')

INSERT INTO Reservation VALUES ('Res10005', '2019-4-13', '2019-4-27' , 'S3334685', 'newMR003')


SELECT *

FROM Reservation;
```

Queries to be ran

```
--1
--Courses in database are HUBS1410, SENG1120, ENG2500, COMP1140, MATH1110
SELECT m.name
FROM Enrolment s join Member m ON s.memberId = m.memberId
WHERE s.courseId = 'HUBS1410';


--2
--Only Dean hughes and Bree Sherd are allowed speakers
SELECT SUM(p.maximumItems) AS 'sum Of Speaker Privileges'
FROM Privilege p, Enrolment ci, Enrolment m
WHERE p.code = (SELECT code FROM Category c WHERE c.name = 'Speaker')
AND p.privilegeId = ci.privilegeId
AND p.privilegeId = ci.privilegeId AND ci.courseId = m.CourseId
AND m.memberId = (SELECT m.memberId FROM Member m WHERE m.name = 'Dean Hughes')


--3
-- -- Jill is the staff member who has acquisitions and Reservations
select  m.name,m.phone, count(DISTINCT a.acquisitionId) 'total acquisitions',
count(DISTINCT r.reservationId) as 'total Reservations'
 from  member m
join Acquisition a on m.memberId = a.memberId
join Reservation r on m.memberId = r.memberId
where year = '2019' and m.memberId = 'S3334685'
group by m.name,m.phone,a.acquisitionId


--4
-- Dean is only one who borrowed camera
SELECT DISTINCT(m.name) FROM Member m
RIGHT JOIN Student s ON m.memberId = s.memberId
LEFT JOIN Loan l ON s.memberId = l.memberId LEFT JOIN MoveableResource m2 ON l.moveableResourceId =
m2.resourceId
LEFT JOIN Resource r ON m2.resourceId = r.resourceId LEFT JOIN Category c ON r.code = c.code
WHERE YEAR(l.dateTimeBorrowed) = '2020' AND c.name = 'Camera' AND m2.model = '1522ci'
```

--5

-- MathBook is most loaned

```
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;
```