ARC Database
Project Final Report

The Best Team
Bardia Borhani
Terence Ho
Jin Kim
Wai Kwan Shum

University of Washington | Bothell
CSS 475 Database Systems
Winter 2018 | Dr. Erika Parson
Mar 12, 2018 | Version 5

Table of Contents

Database Application Area

The Activities & Recreation Center (ARC) is the center of activities and events that occur on the University of Washington Bothell (UWB) campus; it is a place for students to gather, socialize, dine, and exercise.

This database will keep track of the resources available in the ARC. Information about the equipment, activities and rooms available at the ARC will be held by the database. The purpose of the database is to be able to hold all this information and to modify it when UWB students perform many tasks at the ARC such as checking out equipment, signing up for activities, reserving rooms, etc.

Relations/Tables in Database

1. Name of Table: **Equipment**
    a. Attributes:
        i. Department name (string)
        ii. Equipment Serial # (int)
        iii. Name (string)
        iv. Type (string)
        v. Year bought (string)
2. Name of Table: **Equipment_Borrowed**
    a. Attributes:
        i. Student ID# (int)
        ii. Equipment serial number (string)
        iii. Checkout date (date)
        iv. Due date (date)
        v. Return status(boolean)
3. Name of Table: **Student**
    a. Attributes:
        i. Student ID# (int)
        ii. Name (string)
        iii. Contact Phone number (int)
4. Name of Table: **Address**
    a. Attributes:
        i. Student ID# (int)
        ii. House number(int)
        iii. Street name (string)
        iv. City (string)
        v. State (string)

vi. Zip (int)

5. Name of Table: **Activity**
    a. Attributes:
        i. Department name (string)
        ii. Activity ID# (int)
        iii. Activity type: Trips, sports, fitness classes (string)
        iv. Activity date (string)
6. Name of Table: **Activity_Participate**
    a. Attributes:
        i. Student ID (char)
        ii. Activity ID (char)
7. Name of Table: **Department**
    a. Attributes:
        i. Department Name (string)
        ii. Phone number (int)
8. Name of Table: **Room**
    a. Attributes:
        i. Department name (string)
        ii. Room # (int)
9. Name of Table: **Room_Reserve**
    a. Attributes:
        i. Student ID (int)
        ii. Department name (string)
        iii. Room # (int)
        iv. Checkin time (Datetime)
        v. Checkout time (Datetime)

Figure 1. Table/Relation Constraint Diagram

Queries

- The table can show the number of stock available for each equipment in the ARC --> Select the Table record in the ARC table
- Activity table will show activity available for users to sign-up
- Student information can be retrieved with their name or student ID, can be filtered by their institution
- Display possible equipment for checkout
- Tables can display all checkouts and equipment that is available for checkout

Roles

- Employees at the ARC will help people check out equipment, manage rental availability, and activities availability.
- Student can check the availability of equipment to rent out.
- Student can check the availability of on-going activities to participate in.
- Administrator will be able to add or remove items from checkout if new item is added or the condition of an item is no longer rentable, etc.
- Administrator will be able to extend the checkout period.

Schedule

Worked will be done through the week on each deliverable but most time will be spent during group meetings on Tuesdays and Thursdays. Deliverables will be done before the due date as late work cannot be accepted by professor. If someone cannot attend a group meeting, the team member is expected to make up the work in his/her own time.

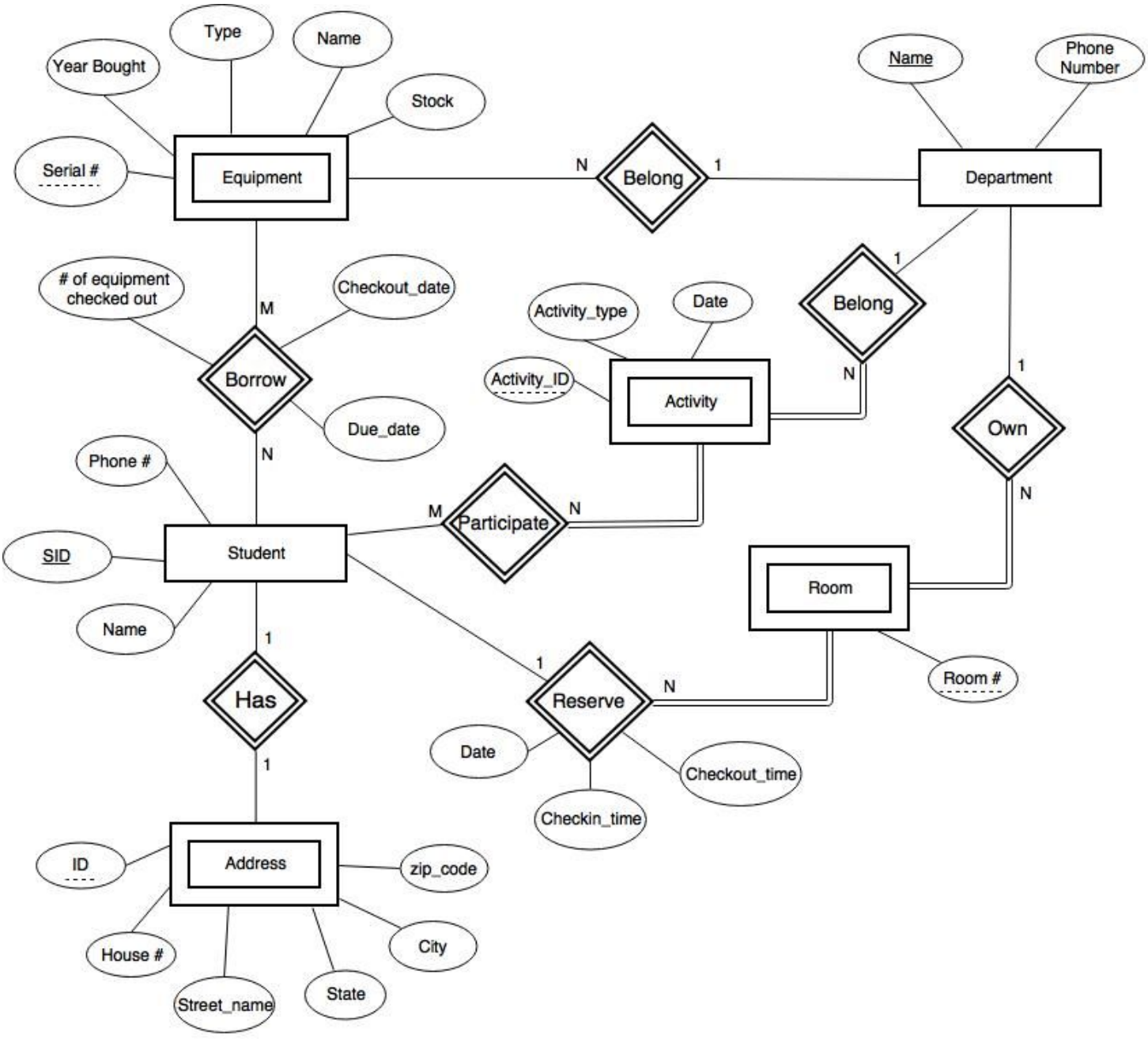
Every member will work on each part of the project

Deliverables	**Approximate schedule**
~~Proposal~~	~~Jan 12th~~
~~Tooling~~	~~Jan. 30th - Feb. 15th (meetings on Jan. 30 and Feb. 1, 6, 8, 13)~~
~~Design documentation~~	~~Jan 12 - Jan 26 (meetings on Jan. 16, 18, 23, 25)~~

~~Populated database~~	~~Feb 3~~
~~SQL query statements~~	~~Feb 10~~
~~Normalization~~	~~Feb 17~~
~~User Interface strategy~~	~~Feb 24~~
~~Documentation and Poster presentation~~	~~Feb 27 - Mar 8(meetings on Feb 27 and Mar 1, 6, 8)~~

Entity-Relationship Diagram

I. Figure 2. ER diagram



II. Description

This ER diagram outlines our database design, and it represents all the tables that are needed for our database. There are two strong entities: Student and Department, and there are four weak entities: Address, Room, Activity and Equipment. This alongside with Reserve, Borrow, and Participate relationships make a total of nine relations in the database.

Every student at UW Bothell that borrows an equipment, reserves a room, or participate in an activity at any time is in the Student relation. A student ID is used as a primary key to uniquely identify a student. A student's name serves as an attribute to clearly display to the user the name of the student associated with the student ID. The phone number attribute serves as a way of contacting a student in case of emergencies. The address entity contains each students full home address. This is used to send mail to a student's address in case of updates to the ARC. Each address relation requires a Student ID (SID) to associate the address with a student.

Each department in the ARC has equipment, activities, and rooms associated with it. The departments in the ARC are Intramurals, Outdoor Wellness, ARC fitness front desk, ARC front desk, and ARC rooms. These three cannot exist with the departments they are associated with; that is why they're identified as weak entities. The rooms represent physical rooms that students can reserve for a certain period of time. The Checkin_time and Checkout_time and Data attributes attached to a room represent the day and times that a student has reserved a room. The departments "own" the rooms and the students "reserve" the rooms.

III. Assumption

- Equipment, Room and Activity cannot exist without Department
- Room is not a strong entity because it serves no purpose without the Department and a Student reserving it
- All Activities, Reservations, and Equipments can only be checked out by Students
- Address is a weak entity because it depends on the student. There is no need for an address in the database if it does not have a student's ID attached to it.
- Every attribute in the Address relation needs to be filled with information. For example, there cannot be an address without a City being indicated. Another example is that an address cannot be identified with a House Number even if all the other attributes are correctly filled in.

IV. Limitations/Constraints

Name of Entity	Attribute	Domain Constraint
Equipment	Department name (string)	Length of the string can be up to 30 characters
	Equipment Serial Number (int)	Integer from #0001-9999
	Name (string)	Up to 25 characters long
	Type (string)	{Gaming, Sporting, Cables}
	Year bought (int)	>0 and <2018 (current date) and in the format YYYY. Example: 2017
Equipment_Borrowed	Student ID Number (int)	Numbers up to 7 digits
	Department name (string)	Intramurals, Outdoor Wellness, ARC Fitness Front Desk, ARC Front Desk, or ARC Rooms.
	Equipment serial number (string)	Integer from #0001-9999
	Checkout date (date)	MM/DD/YYYY
	Due date (date)	MM/DD/YYYY
	Returned (boolean)	(True or false) or (1 or 0)
Student	Student ID Number (int)	Numbers is 7 digits
	Name (string)	Name is formatted as “FirstName LastName”

	Contact Phone number (int)	Up to 15 integers
Student_Address	Student ID Number (int)	Numbers is 7 digits
	House Number (int)	Integer above 0 and below 99999. House numbers cannot be more than 5 digits long
	Street_name (string)	Up to 20 characters long
	City (string)	Up to 10 characters long
	State (string)	Up to 10 characters long
	Zip_code (int)	Must be 5 characters long
Department	Department Name (string)	{Intramurals, Outdoor Wellness, ARC fitness front desk, ARC front desk, or ARC rooms}
	Department Phone Number (int)	No hyphens or parentheses allows - only numbers. Range is between 1111111111 and 9999999999 (10 digits)
Activity	Department Name (string)	Intramurals, Outdoor Wellness, ARC fitness front desk, ARC front desk, or ARC rooms.
	Activity Name (string)	Up to 15 characters long
	Activity ID (int)	Must be 4 characters long
	Activity_Type (string)	Up to 10 characters long
	Activity_Date (string)	MM/DD/YYYY
Activity_Participate	Student ID (int)	Numbers up to 7 digits Integer above 0

	Activity ID (int)	Must be 4 characters long
Room	Department name (string)	{Intramurals, Outdoor Wellness, ARC fitness front desk, ARC front desk, or ARC rooms}
	Room # (int)	3 digit, integers above 0
Room_Reserve	Student ID Number (string)	Numbers up to 7 digits Integer above 0
	Department_Name (string)	{Intramurals, Outdoor Wellness, ARC fitness front desk, ARC front desk, or ARC rooms}
	Room Number (int)	Integer above 0
	Checkin_time (datetime)	8am-12am
	Checkout_time (datetime)	8am-12am, (must be anytime after check in)

Key constraints

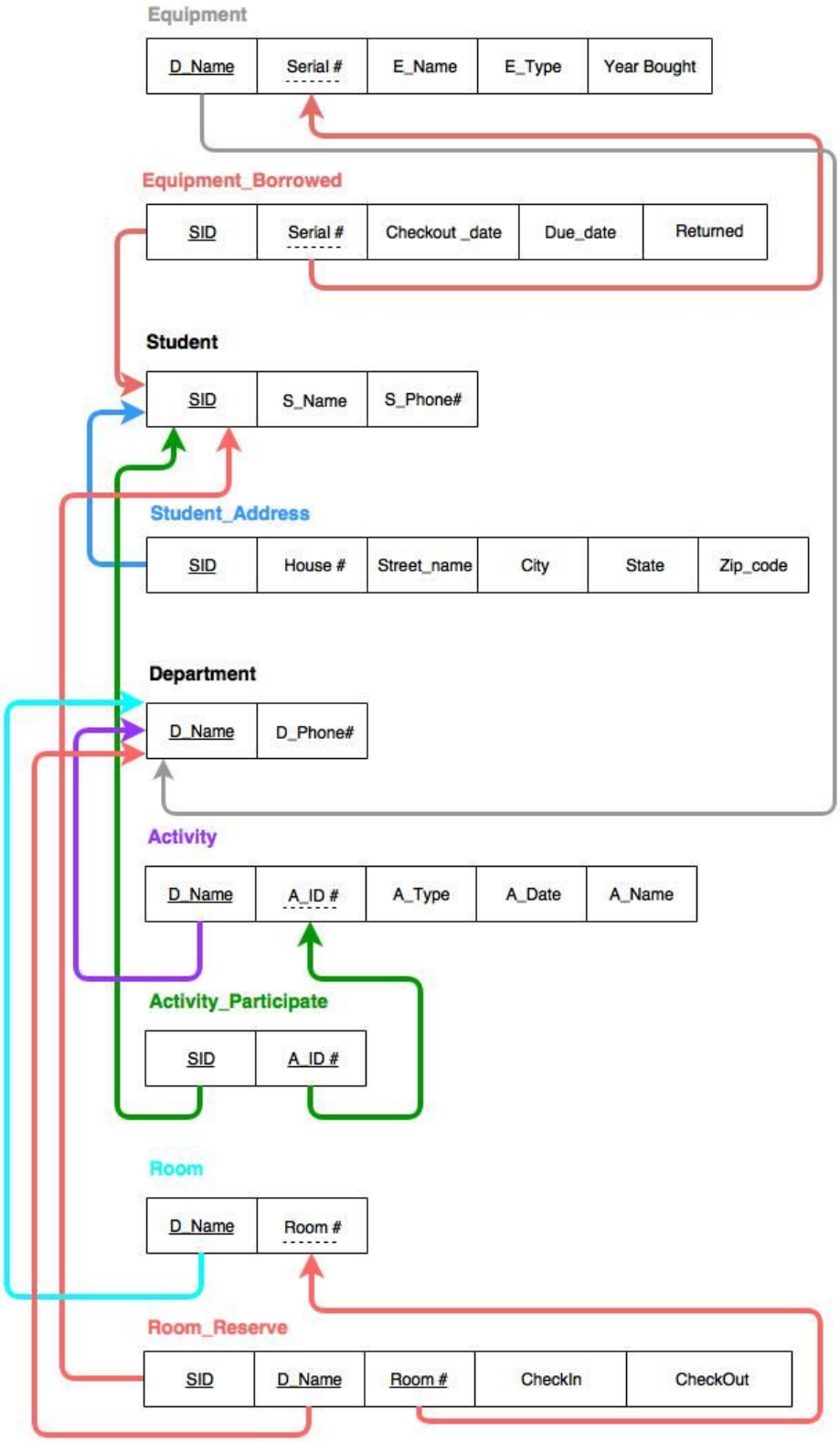
- Student ID number should be unique for each Student
- Department Name should be unique for each Department
- Serial number for each Equipment should be unique only with a Department
- Activity number for each Activity should be unique only with a Department
- Room number for each Room should be unique only with a Department

Entity integrity constraints

- Each Department must have a value for Department Name, which cannot be NULL
- Each Student must have a value for Student ID number, which cannot be NULL
- Each Equipment must have a value for Serial #, which cannot be NULL
- Each Activity must have a value for A_ID, which cannot be NULL
- Each Room must have a value for Room #, which cannot be NULL

Relational Data Model

I. Figure 3. Relation Model Schema



II. Description

- Student
    - SID: - a valid UW Bothell student ID - this is this relation's primary key.
    - S_name: a student's full name
    - S_phone#: a student's phone number
- Student_Address
    - A student's home address
- Department
    - A department located in the ARC: Intramurals, Outdoor Wellness, ARC fitness front desk, ARC front desk, or ARC rooms.
    - Department Phone Number:
- Equipment
    - Equipment attributes like Name and serial number
- Equipment_Borrowed (Relation)
    - Information about what students borrowed and what they borrowed
- Activity
    - Activity name, their unique identifier, when and what department the activity belongs to
- Activity_Participate (Relation)
    - Student IDs and the activity that they are participating in
- Room
    - Name of department the room is under and room number
- Room_Reserve (Relation)
    - Shows who reserved the room, which room it is and what time the check_in and check_out time is for that room

Tools

These are the tools we plan to use to create the database

- **RDBMS:** MySQL
    - MySQL will be used to code and store the database
    - It is chosen because it is one of the most popular free RDBMSs - It is commonly used by software developers around the world
    - It has friendly access for Windows and Mac users
- **UI Tools:** PHP
    - PHP will be used to create the user interface on the AWS EC2 instance. Combining it with Apache, a website is used to display the UI.
    - This language is chosen because it is compatible with MySQL and it follows the LAMP software bundle
- **Database Hosting:** AWS
    - AWS will be used to host the database and the user interface connecting to the database
    - It is chosen because it is free, for the scale of our project, and is commonly used in the software industry. Working with AWS will give us valuable insight as to how to use
    - The AWS RDS (Relational Database Service) will host an instance of the database. It will be set to public so all group members can access it
- **Version Control:** GitHub
    - GitHub will be used to hold the SQL code. It will be used as a source to gain access to different versions of the code throughout the project.
    - It is free and one of the most popular version control services

Access to Database

An AWS cloud server is created to host the ARC database. It serves as an instance under Amazon RDS (Relational Database Service). The instance is named “arcdatabase” and can be accessed using the following credentials:

MySQL Hostname: arcdatabase.cgfmhhwmrqat.us-east-1.rds.amazonaws.com

Username: arcuser

Password: bestteam

If needed, these are the credentials for the AWS EC2 machine that is connected to the RDS instance stated above:

SSH Hostname: ec2-54-163-17-76.compute-1.amazonaws.com

SSH Username: ec2-user

SSH Key File: Attached in submission under name “ArcDatabaseEC2Key.pem”

The link below gives access to the website that has sample queries to the database

Website: http://ec2-54-163-17-76.compute-1.amazonaws.com/home.php

NOTE: The main part of the code for the UI is on the home.php file

Test Data

When creating test data for the ARC database, we received much help from team member Terence Ho. Terence works at the ARC and knows of real world examples of data that would be inserted into a database for the ARC. Using Terence’s knowledge of real world knowledge of the system, we wrote out data by hand into our database and manually added them into the database using insert statements.

MySQL Workbench program was used to enter MySQL code and connect the code to the Amazon RDS instance hosted on AWS. The MySQL code is in the attached document “ARC Database Code”.

SQL code for creating database

```
USE arc;

SET FOREIGN_KEY_CHECKS = 0;
DROP TABLE IF EXISTS DEPARTMENT;
DROP TABLE IF EXISTS  STUDENT;
DROP TABLE IF EXISTS  EQUIPMENT;
DROP TABLE IF EXISTS  EQUIPMENT_BORROWED;
DROP TABLE IF EXISTS  STUDENT_ADDRESS;
DROP TABLE IF EXISTS  ACTIVITY;
DROP TABLE IF EXISTS  ACTIVITY_PARTICIPATE;
DROP TABLE IF EXISTS  ROOM;
DROP TABLE IF EXISTS  ROOM_RESERVE;
SET FOREIGN_KEY_CHECKS = 1;


CREATE TABLE DEPARTMENT
(
d_name VARCHAR(30) NOT NULL,
d_phone VARCHAR(15),
PRIMARY KEY (d_name),
UNIQUE (d_name),
CHECK (d_name = "Intramurals" OR d_name = "Fitness" OR  d_name = "Game" OR d_name =
"Camp" OR d_name = "Sport")
);


CREATE TABLE STUDENT
(
s_id CHAR(7) NOT NULL,
s_name VARCHAR(30) NOT NULL,
s_phone_num VARCHAR(15),
unique(s_id),
PRIMARY KEY (s_id),
CHECK (char_length(s_id) = 7)
);

Create Table EQUIPMENT
(
e_dname VARCHAR(25) NOT NULL,
```

```
e_serialNumber CHAR(4) NOT NULL,
e_name VARCHAR(25) NOT NULL,
e_type VARCHAR(10) NOT NULL,
e_yearBought INT,
Primary key (e_dname, e_serialNumber),
FOREIGN KEY (e_dname) REFERENCES DEPARTMENT(d_name) ON DELETE CASCADE
ON UPDATE CASCADE,
unique(e_serialNumber),
CHECK (e_serialNumber >= 1 and e_serialNumber <= 9999),
CHECK (e_type = "Gaming" OR e_type = "Sporting" OR e_type = "Camping"),
CHECK  (e_stock  >= 0),
CHECK (e_yearBought >= 0 and e_yearBought <= YEAR(CURDATE()))
);

CREATE TABLE EQUIPMENT_BORROWED
(
eb_sid CHAR(7) NOT NULL,
eb_serialNumber CHAR(4) NOT NULL,
eb_checkedOut DATE NOT NULL,
eb_dueDate DATE NOT NULL,
eb_returned BOOLEAN NOT NULL DEFAULT 0,
PRIMARY KEY(eb_sid, eb_serialNumber),
FOREIGN KEY (eb_sid)  REFERENCES STUDENT(s_id) ON DELETE CASCADE ON
UPDATE CASCADE,
FOREIGN KEY (eb_serialNumber) REFERENCES EQUIPMENT(e_serialNumber) ON DELETE
CASCADE ON UPDATE CASCADE,
CHECK (eb_serialNumber <= 9999 AND eb_serialNumber >= 0001),
CHECK(eb_checkedOut <= CURDATE()),
CHECK(eb_dueDate >= eb_checkedOut)
);



CREATE TABLE STUDENT_ADDRESS
(
sa_sid CHAR(7) NOT NULL,
sa_houseNumber INT,
sa_streetName VARCHAR(20),
sa_city VARCHAR(10),
sa_state VARCHAR(10),
sa_zip CHAR(5),
PRIMARY KEY(sa_sid),
```

```
FOREIGN KEY (sa_sid) REFERENCES STUDENT(s_id) ON DELETE CASCADE ON UPDATE
CASCADE,
unique(sa_sid),
CHECK (char_length(sa_sid) = 7),
CHECK(sa_zip > 00000)
);

CREATE TABLE ACTIVITY
(
a_dname  VARCHAR(15) NOT NULL,
a_name          VARCHAR(20) NOT NULL,
a_id CHAR(4) NOT NULL,
a_type VARCHAR(20) NOT NULL,
a_date DATE NOT NULL,
PRIMARY KEY(a_dname, a_id),
FOREIGN KEY(a_dname) REFERENCES DEPARTMENT(d_name) ON DELETE CASCADE
ON UPDATE CASCADE,
CHECK (a_id >= 0001 AND a_id <= 9999)
);

CREATE TABLE ROOM
(
r_dname VARCHAR(15) NOT NULL,
r_number VARCHAR(3) NOT NULL,
Primary Key(r_dname, r_number),
Foreign Key (r_dname) REFERENCES DEPARTMENT(d_name) ON DELETE CASCADE ON
UPDATE CASCADE,
CHECK (char_length(r_number) = 10)
);


CREATE TABLE ACTIVITY_PARTICIPATE
(
ap_sid CHAR(7) NOT NULL REFERENCES STUDENT(s_id) ON DELETE CASCADE ON
UPDATE CASCADE,
ap_aid CHAR(4) NOT NULL REFERENCES ACTIVITY(a_id) ON DELETE CASCADE ON
UPDATE CASCADE,
PRIMARY KEY(ap_sid, ap_aid)
);

CREATE TABLE ROOM_RESERVE
(
```

```
rr_sid CHAR(7) NOT NULL REFERENCES STUDENT(s_id) ON DELETE CASCADE ON
UPDATE CASCADE,
rr_dname VARCHAR(30) NOT NULL REFERENCES DEPARTMENT(d_name) ON DELETE
CASCADE ON UPDATE CASCADE,
rr_rnumber VARCHAR(10) NOT NULL REFERENCES ROOM(r_number) ON DELETE
CASCADE ON UPDATE CASCADE,
rr_checkIn DATETIME NOT NULL,
rr_checkOut DATETIME NOT NULL,
PRIMARY KEY(rr_sid, rr_dname, rr_rnumber)
);
```

SQL code for inserts

```
-- DEPARTMENT INSERTS
INSERT INTO DEPARTMENT VALUES ('Fitness', '425-329-1111');
INSERT INTO DEPARTMENT VALUES ('Game', '425-329-1112');
INSERT INTO DEPARTMENT VALUES ('Camp', '425-329-1113');
INSERT INTO DEPARTMENT VALUES ('Sport', '425-329-1114');
INSERT INTO DEPARTMENT VALUES ('Intramurals', '425-329-1115');

-- EQUIPMENT INSERTS
INSERT INTO EQUIPMENT VALUES ('Fitness', '9999', 'Dumbbell 5LB', 'Sporting', 2016);
INSERT INTO EQUIPMENT VALUES ('Fitness','1121', 'Jump Rope', 'Sporting', 2015);
INSERT INTO EQUIPMENT VALUES ('Game', '5678', 'Monopoly', 'Gaming', 2017);
INSERT INTO EQUIPMENT VALUES ('Game', '2345', 'Mario Kart for Wii', 'Gaming', 2017);
INSERT INTO EQUIPMENT VALUES ('Game', '2346', 'Wii', 'Gaming', 2017);
INSERT INTO EQUIPMENT VALUES ('Camp', '2478', '4 Person Tent', 'Camping', 2014);
INSERT INTO EQUIPMENT VALUES ('Camp', '2477', 'Hiking poles', 'Camping', 2014);
INSERT INTO EQUIPMENT VALUES ('Intramurals', '8567', 'Soccer Ball', 'Sporting', 2016);
INSERT INTO EQUIPMENT VALUES ('Intramurals', '8547', 'Soccer Ball', 'Sporting', 2016);
INSERT INTO EQUIPMENT VALUES ('Intramurals', '8569', 'Basketball', 'Sporting', 2016);
INSERT INTO EQUIPMENT VALUES ('Intramurals', '8568', 'Volleyball', 'Sporting', 2016);
INSERT INTO EQUIPMENT VALUES ('Sport', '7890', 'Keiser Power Racks', 'Sporting', 2018);
INSERT INTO EQUIPMENT VALUES ('Sport', '7891', 'Precor Ellipticals', 'Sporting', 2018);

-- STUDENT INSERTS
INSERT INTO STUDENT VALUES ('1234567', 'WAI KWAN SHUM', '425-320-1234');
INSERT INTO STUDENT VALUES ('1111111', 'TERENCE HO', '425-212-1111');
INSERT INTO STUDENT VALUES ('2222222', 'BARDIA BORHANI', '206-123-4567');
INSERT INTO STUDENT VALUES ('3333333', 'JIN KIM', '425-325-3333');
INSERT INTO STUDENT VALUES ('4444444', 'JOHN MCDONALD', '206-111-4567');
```

```
INSERT INTO STUDENT VALUES ('7654321', 'MARY JANE', '206-222-4567');

-- STUDENT ADDRESS INSERTS
INSERT INTO STUDENT_ADDRESS VALUES ('1234567', 1234, 'WAI STREET', 'LAKE CITY',
'WA', '98125');
INSERT INTO STUDENT_ADDRESS VALUES ('1111111', 1111, 'TERENCE DRIVE',
'BOTHELL', 'WA', '98011');
INSERT INTO STUDENT_ADDRESS VALUES ('2222222', 2323, 'BARDIA BOULEVARD',
'REDMOND', 'WA', '98052');
INSERT INTO STUDENT_ADDRESS VALUES ('3333333', 1033, 'JIN AVENUE', 'EVERETT',
'WA', '98204');
INSERT INTO STUDENT_ADDRESS VALUES ('4444444', 4343, 'MCNUGGET STREET',
'SEATTLE', 'WA', '98134');
INSERT INTO STUDENT_ADDRESS VALUES ('7654321', 9876, 'QUEEN ANNE STREET',
'QUEEN ANNE', 'WA', '98256');


-- ACTIVITY INSERTS
INSERT INTO ACTIVITY VALUES ('Fitness', 'Spin Class', '1234', 'Workout', '2018-03-20');
INSERT INTO ACTIVITY VALUES ('Fitness', 'Yoga', '2222', 'Workout', '2018-03-25');
INSERT INTO ACTIVITY VALUES ('Game', 'Mario Kart party', '3456', 'Video Game',
'2018-04-01');
INSERT INTO ACTIVITY VALUES ('Game', 'Monopoly party', '3457', 'Boardgame',
'2018-04-05');
INSERT INTO ACTIVITY VALUES ('Camp', 'Camp Long', '2345', 'Camping', '2018-04-22');
INSERT INTO ACTIVITY VALUES ('Camp', 'Mountain Rainer', '2346', 'Camping', '2018-04-25');
INSERT INTO ACTIVITY VALUES ('Sport', 'DodgeBall', '1345', 'Tournament', '2018-03-22');
INSERT INTO ACTIVITY VALUES ('Sport', '5v5 Basketball', '1346', 'Sports Leagues',
'2018-03-15');

-- ACTIVITY PARTICIPATE INSERTS
INSERT INTO ACTIVITY_PARTICIPATE VALUES ('1234567', '1234');
INSERT INTO ACTIVITY_PARTICIPATE VALUES ('1111111', '2222');
INSERT INTO ACTIVITY_PARTICIPATE VALUES ('2222222', '3456');
INSERT INTO ACTIVITY_PARTICIPATE VALUES ('3333333', '2345');
INSERT INTO ACTIVITY_PARTICIPATE VALUES ('1111111', '2345');
INSERT INTO ACTIVITY_PARTICIPATE VALUES ('4444444', '1345');
INSERT INTO ACTIVITY_PARTICIPATE VALUES ('2222222', '1345');
INSERT INTO ACTIVITY_PARTICIPATE VALUES ('7654321', '1345');
INSERT INTO ACTIVITY_PARTICIPATE VALUES ('4444444', '1346');
INSERT INTO ACTIVITY_PARTICIPATE VALUES ('7654321', '1346');
INSERT INTO ACTIVITY_PARTICIPATE VALUES ('1111111', '1346');
INSERT INTO ACTIVITY_PARTICIPATE VALUES ('1111111', '3457');
```

```
INSERT INTO ACTIVITY_PARTICIPATE VALUES ('3333333', '3457');
INSERT INTO ACTIVITY_PARTICIPATE VALUES ('7654321', '3457');



-- EQUIPMENT BORROWED INSERTS
INSERT INTO EQUIPMENT_BORROWED VALUES ('1234567', '5678', '2018-02-28',
'2018-03-08', 1);
INSERT INTO EQUIPMENT_BORROWED VALUES ('1111111', '2478', '2018-02-26',
'2018-03-05', 1);
INSERT INTO EQUIPMENT_BORROWED VALUES ('2222222', '9999', '2018-02-23',
'2018-03-09', 0);
INSERT INTO EQUIPMENT_BORROWED VALUES ('3333333', '2345', '2018-02-28',
'2018-03-05', 0);
INSERT INTO EQUIPMENT_BORROWED VALUES ('4444444', '1121', '2018-02-14',
'2018-02-22', 1);
INSERT INTO EQUIPMENT_BORROWED VALUES ('1234567', '8567', '2018-02-14',
'2018-02-22', 0);
INSERT INTO EQUIPMENT_BORROWED VALUES ('7654321', '2477', '2018-03-11',
'2018-03-14', 0);
INSERT INTO EQUIPMENT_BORROWED VALUES ('7654321', '7890', '2018-03-02',
'2018-03-14', 1);



-- ROOM INSERTS
INSERT INTO ROOM VALUES ('Fitness', '001');
INSERT INTO ROOM VALUES ('Fitness', '002');
INSERT INTO ROOM VALUES ('Game', '001');
INSERT INTO ROOM VALUES ('Game', '002');
INSERT INTO ROOM VALUES ('Sport', '001');
INSERT INTO ROOM VALUES ('Sport', '002');

-- ROOM RESERVE INSERTS
INSERT INTO ROOM_RESERVE VALUES ('1234567', 'Fitness','001', '2018-01-20 15:00:00',
'2018-01-20 19:00:00');
INSERT INTO ROOM_RESERVE VALUES ('1111111', 'Fitness','002', '2018-01-22 12:00:00',
'2018-01-22 14:00:00');
INSERT INTO ROOM_RESERVE VALUES ('2222222', 'Sport','001', '2018-02-15 13:30:00',
'2018-02-15 14:30:00');
INSERT INTO ROOM_RESERVE VALUES ('3333333', 'Game','001', '2018-03-03 15:30:00',
'2018-03-03 17:00:00');
```

INSERT INTO ROOM_RESERVE VALUES ('4444444', 'Game','002', '2018-02-15 14:30:00', '2018-02-15 16:00:00');
INSERT INTO ROOM_RESERVE VALUES ('7654321', 'Game','002', '2018-02-15 17:00:00', '2018-02-15 18:00:00');

Normalization

Department	
d_name	d_phone

Department is normalized as BCNF as it satisfy conditions of
1NF, 2NF, 3NF and has X as a superkey of R

Room	
r_dname	r_number

Room is normalized as BCNF as it satisfy conditions of
1NF, 2NF, 3NF and has X as a superkey of R

Room_Reserve				
rr_sid	rr_dname	rr_rnumber	rr_checkIn	rr_checkOut

Room_Reserve is normalized as BCNF as it satisfy conditions of
1NF, 2NF, 3NF and has X as a superkey of R

Activity				
a_dname	a_name	a_id	a_type	a_date

Activity is normalized as BCNF as it satisfy conditions of
1NF, 2NF, 3NF and has X as a superkey of R

Activty_Participate	
ap_sid	ap_aid

Activty_Participate is 1NF as this relation has no non-primary-attribute that is fully dependent on X. We can't further normalize this table as there are no participating attribute. However if we can add an attribute, it will still need to be normalized to 2NF and check the condition for 3NF and BCNF.

Equipment				
e_dname	e_serialNumber	e_name	e_type	e_yearBought

Equipment is normalized as BCNF as it satisfy conditions of
1NF, 2NF, 3NF and has X as a superkey of R

Equipment_Borrowed				
eb_sid	eb_serialNumber	eb_checkedOut	eb_dueDate	eb_returned

Equipment_Borrowed is normalized as BCNF as it satisfy conditions of
1NF, 2NF, 3NF and has X as a superkey of R

Student		
s_id	s_name	s_phone_num

Student is normalized as BCNF as it satisfy conditions of
1NF, 2NF, 3NF and has X as a superkey of R

Student_Address					
sa_sid	sa_houseNumber	sa_streetName	sa_city	sa_state	sa_zip

Student_Address is in 2NF as it sa_city, sa_state and sa_zip can be decomposed further to make 3NF. As of now, sa_state sa_zip can be considered transitivity depend to sa_sid, as sa_sid->sa_city and sa_city->sa_state,sa_zip to change Student_Address as 3NF and then BCNF.

Most of our current relations are normalized to BCNF with exception of Activity_Participate and Student_Address. Activity_Participate can't be further decomposed to 3NF, but if we had normalized Student_Address, it would make a better schema of our design and also save us a lot of storage. Reason being, State and Zip code will always follow the city (assuming that all students are living in State of Washington), and by identifying the city, state and zip code can be easily referenced on a separate relation.

SQL query statements

SQL statement	Purpose
SELECT e_serialNumber FROM EQUIPMENT WHERE e_name = "Soccer Ball" AND e_serialNumber IN (SELECT e.e_serialNumber FROM EQUIPMENT as e WHERE e.e_serialNumber NOT IN (SELECT eb_serialNumber FROM EQUIPMENT_BORROWED) UNION ALL (SELECT eb_serialNumber FROM EQUIPMENT_BORROWED WHERE eb_returned = 1));	Check which soccer balls are available for loan
SELECT COUNT(*) as Num_of_items_on_loan FROM EQUIPMENT_BORROWED WHERE eb_returned = 0;	Display number of items on loan
SELECT eb.eb_sid as SID, s.s_name as Student_name, eb.eb_serialNumber as Equipment_serialNumber, e.e_name as Equipment_name FROM EQUIPMENT_BORROWED as eb, EQUIPMENT as e, STUDENT as s WHERE eb.eb_returned = 0 AND eb.eb_serialNumber = e.e_serialNumber AND eb.eb_sid = s.s_id;	Display all students who have items checked out
SELECT eb.eb_sid as SID, s.s_name as Student_name, eb.eb_serialNumber as Equipment_serialNumber, e.e_name as Equipment_name, eb.eb_dueDate as Due_date FROM EQUIPMENT_BORROWED as eb, EQUIPMENT as e, STUDENT as s WHERE eb.eb_returned = 0 AND eb.eb_dueDate < CURDATE() AND eb.eb_serialNumber = e.e_serialNumber AND eb.eb_sid = s.s_id ORDER BY eb.eb_dueDate;	Display all student who have items overdue
SELECT a.a_dname as Department, a.a_date as Date, a.a_name as Activity_name, COUNT(*) as Num_of_participants FROM ACTIVITY as a, ACTIVITY_PARTICIPATE as ap WHERE a.a_dname = "Camp" AND a.a_date = "2018-04-22" AND ap.ap_aid = a.a_id GROUP BY a_name;	Display the number of participants in an Activity held by Camp Dept on April 22, 2018.

SELECT s.s_id as SID, s.s_name as Student_name, rr.rr_rnumber as Room_no, rr.rr_dname as Dept FROM STUDENT as s, ROOM_RESERVE as rr WHERE rr.rr_checkIn >= '2018-02-15 00:00:00' AND rr.rr_checkOut <= '2018-02-15 23:59:59' AND rr.rr_sid = s.s_id;	Display all student who have reserved a room on Feb 15, 2018
UPDATE STUDENT as s SET s.s_phone_num = '425-321-3832' WHERE s.s_id = '3333333';	Update the phone number of student who has ID number “3333333” to 425-321-3832
INSERT INTO EQUIPMENT_BORROWED(eb_sid, eb_serialNumber, eb_checkedOut, eb_dueDate,eb_returned) VALUES ('1111111', '8547', '2018-03-11', '2018-03-15', 0);	Student “1111111” wants to check out soccer ball “8567”.
SELECT s.s_id as SID, s.s_name as Student_name, s.s_phone_num as Phone_no, sa.sa_houseNumber as House_no, sa.sa_streetName as St_name, sa.sa_city as City, sa.sa_state as State, sa.sa_zip as Zip FROM STUDENT as s, STUDENT_ADDRESS as sa, ACTIVITY_PARTICIPATE as ap WHERE ap.ap_aid = 2222 AND ap.ap_sid = s.s_id AND s.s_id = sa.sa_sid;	Display detailed information of students who has signed up activity with activity ID “2222”
INSERT INTO ACTIVITY_PARTICIPATE(ap_sid, ap_aid) VALUES ('2222222', '2345');	Student “2222222” wants to sign up for camping trip “2345”.
INSERT INTO EQUIPMENT_BORROWED(eb_sid, eb_serialNumber, eb_checkedOut, eb_dueDate,eb_returned) VALUES ('2222222', (SELECT e_serialNumber FROM EQUIPMENT WHERE e_name = 'Monopoly'), '2018-03-11', '2018-03-15', 0);	Student “2222222” wants to check out Monopoly game.
DELETE FROM EQUIPMENT_BORROWED WHERE eb_serialNumber = '5678';	Student returns game with serialNumber ‘5678’.

Data Testing

We populated the database manually with made-up data that matched the realistic use of our database. We tested the database with queries that staff at the ARC would use based on our team member, Terence, who works at ARC. New data can also be inserted in the website we made in order to further populate the database.

Project evaluation

Overall, our ARC database has been properly designed, and went through the steps in ANSI-SPARC architecture for a proper build up. Everytime we were introduced a new schema, we had to go back to previous step to make design choices to be reflected and updated. Every group member had put in 3-4 hours of effort every week, and we had met at least once or twice after the class to re-evaluate and discuss our project.

Our group was great at discussing for ideas, and giving opinions on how to write the design and how we can change the overall database. We also excelled at listening to everyone's comment and also giving feedback in real time. Everyone shown much enthusiasm toward the project and cooperate together without causing and creating any hindrance or obstacle.

The biggest challenge we faced for our project was the ambiguity of some of the design choices, and the database. It was difficult to come up with a definite answer, and thus lot of discussion was made everytime we were making a new design. As we got deeper into the implementations, we realized that there needed to be some design changes such as adding a boolean to keep track of returned items so that overdue items are still shown. Another challenge was the difficulty of using the new software we were not familiar with. Google did help setting up the programs, but our lack of experience made it difficult to understand and actually incorporate the proper tools for the project.

If we had another week to work on the project, we would have polished the UI a bit more and implemented more SQL queries for the user. We would also have implemented seperate pages for different users so, for example, a front desk staff doesn't have the rights to delete everything on the database or necessarily see all the student data, while the administrator can see all this information if they need to verify information.

Changes made

- Updated SQL code for creating database and inserts
- Updated ER diagram and Relational model based on the latest version of the database