

CPSC 304 Project Cover Page

Milestone #: 2

Date: October 13th, 2023

Group Number: 67

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Justin Prasad	78028420	c8s1o	justinm.prasad@gmail.com
Andrew Joji	28440428	x0m6d	andrewjoji71@gmail.com
Pedro de Sant'Anna Novais	41950486	g9u5j	psantnovais@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

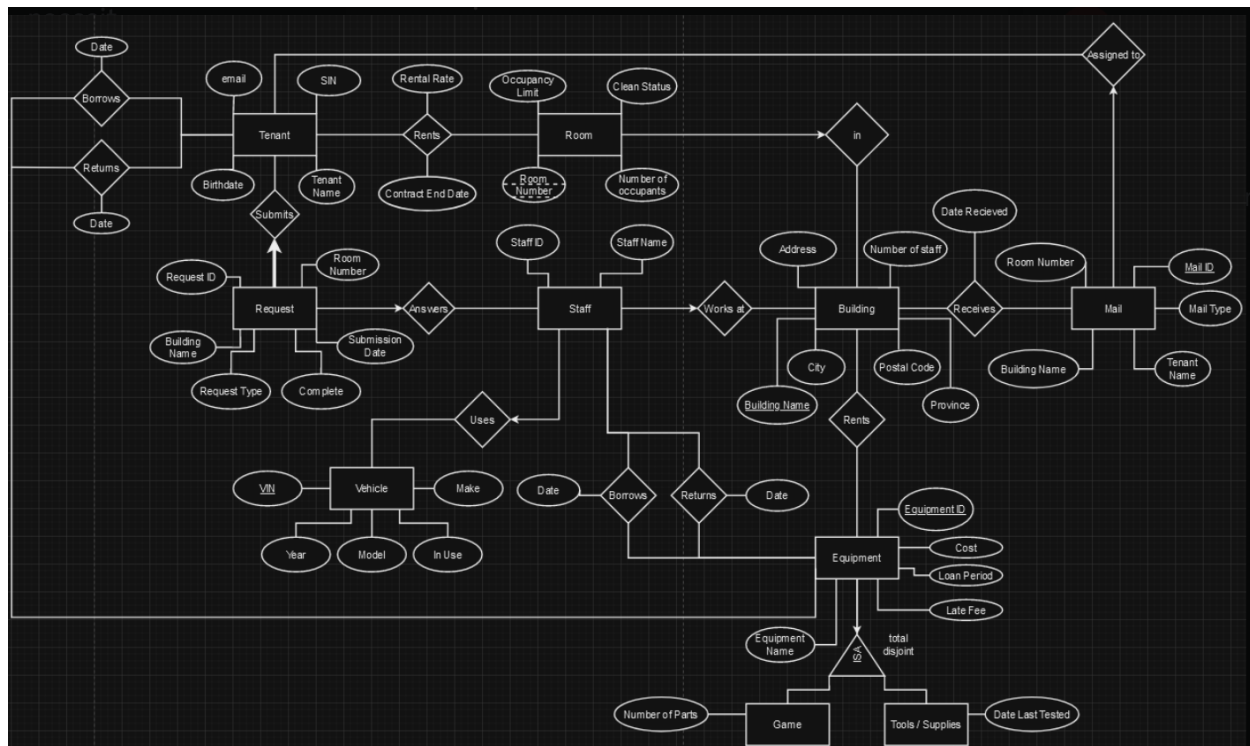
In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Brief Summary:

Our project lies in the domain of property management and aims to mimic what a real life database for a communal property would look like. Things we would like to include are logging and tracking maintenance requests, tenants who currently live in the building, equipment that can be loaned for some period of time, staff, front desk workers, and mail coming into a building and being assigned to a tenant. We have also added the Vehicles entity to record which vehicles staff have access to and track if they are available to use.

We have noted the assumptions and limitations in scope in the following sections.

Updated ER Diagram:



Change Log:

- Added Vehicle Entity
- Removed ISA from Requests and added RequestType Attribute
- FrontDesk Entity changed to Building and added attributes as shown below
- Added attribute Occupants to Room entity
- Made Room Weak Entity of Building
- A change was made such that a request can only exist if a tenant submits it
- The bolded attributes were removed since they served no purpose

- For the ISA being derived from the Requests attribute, we have maintenance and cleaning requests being split from it.
 - We have now added a RoomType attribute for CleaningRequest which be one of ("Bathroom", "Unit", "Kitchen")
 - And added attribute MaintenanceType for MaintenanceRequest which will be one of ("Plumbing", "Electrical", "Furniture")
- Mail table Size attribute changed to Type which is either "Parcel" or "Letter"
- Mail name is removed due to redundancy
- Mail's attributes "Building, Room Number, SIN" are no longer keys due to redundancy with "Mail ID"
- Removed SIN from Mail Entity

Assumptions/Limitations:

- o Many staff work at each Building, but **not all staff work at the Building**
- o Rooms can be empty, but **all Tenants must have a Room**
- o Room is a Weak Entity, Room Number alone **cannot** identify Building Name (e.g. Room 100 can exist in Marine Drive 1 and Marine Drive 2)
- o **Multiple Tenants** can Rent the **same Room**
- o **Multiple Tenants** in the same room **cannot** have the **same Name**
- o 1 member of Staff answers a given Request
- o **Only 1 staff uses a Vehicle at a time** (one to one)
- o **Mail ID uniquely identifies the mail**, no two buildings can assign the same Mail ID
- o **All mail received** is addressed to a **valid** Tenant, Room and Building
- o Email will be sent to Tenant once Mail has been assigned
- o No 2 vehicles have the same Model if the Make is different (Model → Make)
- o For Building we can have the same Address for different Buildings (e.g. Main Street in Toronto and Main Street in Vancouver) so it is not a candidate key
- o Vehicle can exist without Staff (i.e its not in use and empty), and one Staff member can only borrow one car at a time.

Clarification of Attributes

- Room
 - o Room Number: Room numbers are integers that identify a Room within a Building
 - o Number of Occupants: Number of occupants in the room currently
 - o Occupancy Limit: Number of occupants allowed in the room
 - o Clean Status: Either "Clean" or "Dirty" can be changed from clean to dirty by submitting cleaning request and dirty to clean once request completed
- Building
 - o Building Name (e.g. "Marine Drive 1")
 - o Address (e.g. "2075 West Mall")

- Number of staff: Number of staff assigned to building
- Vehicle
 - Make (e.g. “Toyota”)
 - Model (e.g. “Prius”)
 - InUse: Boolean to flag if the vehicle unavailable to use
- Request
 - Submission Date: Date request was submitted
 - Request Type: Either “Cleaning” or “Maintenance”
 - Complete: Boolean to flag if request is fulfilled
- Equipment
 - Cost: Cost of replacing the Equipment
 - Loan Period: Number of days the equipment can be borrowed for
 - Equipment Name (e.g. “Ping Pong Set”, “Drill”, “Vacuum Cleaner”)
- (ISA) GameEquipment
 - Number of Parts Parts the equipment has (e.g. PingPong set would have 3 parts)
- (ISA) ToolsSuppliesEquipment
 - DateLastTested: Date the equipment was last verified to work and be safe to use
- Mail
 - Tenant Name: Name of tenant mail is addressed to
 - MailType: Type of mail either “Parcel” or “Letter”
 - Room Number: Indicating the room number mail is addressed to
 - Building Name: Indicating the building mail is addressed to

Schema

```
Building(  
  BuildingName: string PK,  
  Address: string NOT NULL,  
  NumberOfStaff: integer NOT NULL,  
  City: string NOT NULL,  
  Province: string NOT NULL,  
  PostalCode: string NOT NULL,  
)
```

```
Tenant(  
  SIN: integer PK,  
  TenantName: string NOT NULL,  
  Birthdate: date NOT NULL,  
  Email: string CK  
)
```

University of British Columbia, Vancouver

Department of Computer Science

Request(

RequestID: integer PK,
BuildingName: string FK REFERENCES Building(BuildingName),
RoomNumber: integer NOT NULL,
RequestType: string NOT NULL,
Complete: boolean DEFAULT FALSE,
SubmissionDate: date NOT NULL,
StaffID: integer FK REFERENCES Staff(StaffID),
SIN: integer FK REFERENCES Tenant(SIN)

)

Room(

RoomNumber: integer PK ,
BuildingName: string FK REFERENCES Building(BuildingName),
CleanStatus: boolean NOT NULL,
OccupancyLimit: integer NOT NULL,
NumberOfOccupants: integer NOT NULL

)

Staff(

StaffID: integer PK,
StaffName: string NOT NULL,
VIN: string FK REFERENCES Vehicle(VIN) DEFAULT NULL

)

GameEquipment(

EquipmentID: integer PK,
Cost: float NOT NULL,
LoanPeriod: integer NOT NULL,
LateFee: float NOT NULL,
EquipmentName: string NOT NULL,
NumberOfParts: integer NOT NULL

)

Vehicle(

VIN: string PK,
Make: string NOT NULL,
Year: integer NOT NULL,
InUse: boolean NOT NULL,
Model: string NOT NULL

)

University of British Columbia, Vancouver

Department of Computer Science

ToolsSuppliesEquipment(
EquipmentID: integer PK,
Cost: float NOT NULL,
LoanPeriod: integer NOT NULL,
LateFee: float NOT NULL,
EquipmentName: string NOT NULL,
DateLastTested: date NOT NULL
)

Mail(
MailID: integer PK,
TenantName: string REFERENCES Tenant(Name) NOT NULL,
RoomNumber: integer FK REFERENCES Room(RoomNumber) NOT NULL,
BuildingName: string FK REFERENCES Building(BuildingName) NOT NULL,
MailType: string NOT NULL,
Name: integer FK REFERENCES Tenant(Name)
)

TenantRentsRoom(
SIN: integer FK REFERENCES Tenant(SIN),
RentalRate: float NOT NULL,
ContractEndDate: date NOT NULL,
RoomNumber: integer FK REFERENCES Room(RoomNumber),
BuildingName: string FK REFERENCES Building(BuildingName)
PK (RoomNumber, BuildingName, SIN)
)

BuildingReceivesMail(
MailID: integer PK REFERENCES Mail(MailID),
Date Received: date,
)

StaffReturnsEquipment(
Date: date,
StaffID: integer FK REFERENCES Staff(StaffID),
EquipmentID: integer FK REFERENCES Equipment(EquipmentID)
PK (StaffID, EquipmentID, Date)
)

StaffBorrowsEquipments(
Date: date,

University of British Columbia, Vancouver

Department of Computer Science

StaffID: integer FK REFERENCES Staff(StaffID),
EquipmentID: integer FK REFERENCES Equipment(EquipmentID)
PRIMARY KEY (StaffID, EquipmentID, Date)

)

TenantReturnsEquipment(
Date: date,
SIN: integer FK references Tenant(SIN),
EquipmentID: integer FK references Equipment(EquipmentID)
PRIMARY KEY (EquipmentID, Date, SIN)

)

TenantBorrowsEquipment(
Date: date,
SIN: integer K references Tenant(SIN),
EquipmentID: integer FK references Equipment(Equipment),
PRIMARY KEY (EquipmentID, Date, SIN)

)

Normalization

Building

FDs:

Building Name \rightarrow Address, NumberOfStaff, City, Province, Postal Code, StaffID

Postal Code \rightarrow City, Province

(City, Province, Address) \rightarrow Postal Code

Step 1 Find Closures of FDs:

$\{\text{Building Name}\}^+ = \{\text{Building Name, Address, NumberOfStaff, City, Province, Postal Code}\}$

$\{\text{Postal Code}\}^+ = \{\text{Postal Code, City, Province}\}$

$\{\text{City, Province, Address}\}^+ = \{\text{City, Province, Address, Postal Code}\}$

Step 2 Create Relations for Each FD:

R1(Building Name, Address, NumberOfStaff, City, Province, Postal Code, StaffID)

R2(Postal Code, City, Province)

R3(~~City, Province, Address, Postal Code~~)

Step 3 Remove Redundant Relations:

R3 is redundant as (City, Province, Address) \rightarrow Postal Code can be derived from R1 and R2

Step 4 Ensure all FDs are preserved: They are preserved

Step 5 Ensure 3NF: R1 Building is a superkey, R2 Postal Code is a superkey

University of British Columbia, Vancouver

Department of Computer Science

Tenant

FDs:

SIN \rightarrow Tenant Name, birth date, email

Email \rightarrow Tenant Name, birth date, email, SIN

Step 1 Find Closures of FDs:

$\{SIN\}^+ = \{SIN, Tenant\ Name, Birthdate, Email\}$

$\{Email\}^+ = \{SIN, Tenant\ Name, Birthdate, Email\}$

Step 2 Create Relations for Each FD:

R1(SIN, Tenant Name, Birthdate, Email)

R2(~~Email, Tenant Name, Birthdate, SIN~~)

Step 3 Remove Redundant Relations: R2 is redundant as SIN \rightarrow Tenant Name, birth date, email

Step 4 Ensure all FDs are preserved: It is preserved

Step 5 Ensure 3NF: This relation is in 3NF as SIN is a superkey

Request

FDs:

RequestID \rightarrow Building Name, Room Number, Request Type, Complete, Submission Date, StaffID, SIN

Step 1 Find Closures of FDs:

$\{RequestID\}^+ = \{RequestID, Building\ Name, Room\ Number, Request\ Type, Complete, Submission\ Date, StaffID, SIN\}$

Step 2 Create Relations for Each FD:

R1(RequestID, Building Name, Room Number, Request Type, Complete, Submission Date, StaffID, SIN)

Step 3 Remove Redundant Relations: No redundant relations

Step 4 Ensure all FDs are preserved: All FDs are preserved

Step 5 Ensure 3NF: Because RequestID is a superkey, it is in 3NF

Room

FDs: RoomNumber, BuildingName \rightarrow CleanStatus, OccupancyLimit, NumberOfOccupants

Step 1 Find Closures of FDs:

$\{RoomNumber, BuildingName\}^+ = \{RoomNumber, BuildingName, CleanStatus, OccupancyLimit, NumberOfOccupants\}$

Step 2 Create Relations for Each FD:

R1(RoomNumber, BuildingName, CleanStatus, OccupancyLimit, NumberOfOccupants)

Step 3 Remove Redundant Relations: No redundant relations

University of British Columbia, Vancouver

Department of Computer Science

Step 4 Ensure all FDs are preserved: All FDs are preserved

Step 5 Ensure 3NF: This relation is in 3NF as RoomNumber, BuildingName is a superkey

Staff

FDs:

StaffID \rightarrow StaffName, VIN

Step 1 Find Closures of FDs:

$\{\text{StaffID}\}^+ = \{\text{StaffID}, \text{StaffName}, \text{VIN}\}$

Step 2 Create Relations for Each FD: R1(StaffID, StaffName, VIN)

Step 3 Remove Redundant Relations: None

Step 4 Ensure all FDs are preserved: All FDs are preserved

Step 5 Ensure 3NF : Because StaffID is a superkey in the relation it is in 3NF

GameEquipment

FDs:

EquipmentID \rightarrow Cost, LoanPeriod, LateFee, EquipmentName, NumberOfParts

Step 1 Find Closures of FDs:

$\{\text{EquipmentID}\}^+ = \{\text{EquipmentID}, \text{Cost}, \text{LoanPeriod}, \text{LateFee}, \text{EquipmentName}, \text{NumberOfParts}\}$

Step 2 Create Relations for Each FD:

R1(EquipmentID, Cost, LoanPeriod, LateFee, EquipmentName, NumberOfParts)

Step 3 Remove Redundant Relations: No redundant relations to be removed

Step 4 Ensure all FDs are preserved All FDs are preserved

Step 5 Ensure 3NF: Because EquipmentID is a superkey in the relation it is in 3NF

Vehicle

FDs:

VIN \rightarrow Make, Year, InUse, Model

Model \rightarrow Make

Step 1 Find Closures of FDs:

$\{\text{VIN}\}^+ = \{\text{VIN}, \text{Make}, \text{Year}, \text{InUse}, \text{Model}\}$

$\{\text{Model}\}^+ = \{\text{Model}, \text{Make}\}$

Step 2 Create Relations for Each FD:

R1(VIN, Make, Year, InUse, Model)

R2(Model, Make)

Step 3 Remove Redundant Relations: No redundant relations

Step 4 Ensure all FDs are preserved: All FDs are preserved

Step 5 Ensure 3NF: This relation is in 3NF as both VIN, Model is a superkey

ToolsSuppliesEquipment

FDs:

EquipmentID \rightarrow Cost, LoanPeriod, LateFee, EquipmentName, DateLastTested

Step 1 Find Closures of FDs:

$\{\text{EquipmentID}\}^+ = \{\text{EquipmentID}, \text{Cost}, \text{LoanPeriod}, \text{LateFee}, \text{EquipmentName}, \text{DateLastTested}\}$

Step 2 Create Relations for Each FD:

R1(EquipmentID, Cost, LoanPeriod, LateFee, EquipmentName, DateLastTested)

Step 3 Remove Redundant Relations: No redundant relations to be removed

Step 4 Ensure all FDs are preserved: All FDs are preserved

Step 5 Ensure 3NF: Because ToolsSuppliedEquipment is a superkey, it is in 3NF

Mail

FDs:

MailID \rightarrow TenantName, RoomNumber, BuildingName, MailType

Step 1 Find Closures of FDs:

$\{\text{MailID}\}^+ = \{\text{MailID}, \text{TenantName}, \text{RoomNumber}, \text{BuildingName}, \text{MailType}\}$

Step 2 Create Relations for Each FD:

R1(MailID, TenantName, RoomNumber, BuildingName, MailType)

Step 3 Remove Redundant Relations: No redundant to be removed

Step 4 Ensure all FDs are preserved: They are preserved

Step 5 Ensure 3NF: Because MailID is a superkey, it is in 3NF

TenantRentsRoom

FDs: RoomNumber, BuildingName, SIN \rightarrow RentalRate, ContractEndDate

Step 1 Find Closures of FDs: $\{\text{RoomNumber}, \text{BuildingName}, \text{SIN}\}^+ = \{\text{RoomNumber}, \text{BuildingName}, \text{SIN}, \text{RentalRate}, \text{ContractEndDate}\}$

Step 2 Create Relations for Each FD: R1(RoomNumber, BuildingName, SIN, RentalRate, ContractEndDate)

Step 3 Remove Redundant Relations: No redundancies

Step 4 Ensure all FDs are preserved: Preserved

Step 5 Ensure 3NF: Because RoomNumber, BuildingName is a superkey, it is in 3NF

University of British Columbia, Vancouver

Department of Computer Science

BuildingReceivesMail

FDs:

MailID \rightarrow DateReceived

Step 1 Find Closures of FDs:

$\{\text{MailID}\}^+ = \{\text{MailID}, \text{DateReceived}\}$

Step 2 Create Relations for Each FD:

R1(MailID, DateReceived)

Step 3 Remove Redundant Relations: No redundant to be removed

Step 4 Ensure all FDs are preserved: They are preserved

Step 5 Ensure 3NF: Because MailID is a superkey, it is in 3NF

TenantRentsRoom

FDs: RoomNumber, BuildingName, SIN \rightarrow RentalRate, ContractEndDate

Step 1 Find Closures of FDs: $\{\text{RoomNumber}, \text{BuildingName}, \text{SIN}\}^+ = \{\text{RoomNumber}, \text{BuildingName}, \text{SIN}, \text{RentalRate}, \text{ContractEndDate}\}$

Step 2 Create Relations for Each FD: R1(RoomNumber, BuildingName, SIN, RentalRate, ContractEndDate)

Step 3 Remove Redundant Relations: No redundancies

Step 4 Ensure all FDs are preserved: Preserved

Step 5 Ensure 3NF: Because RoomNumber, BuildingName is a superkey, it is in 3NF

The following are comprised of all keys (No FDs) so the relation includes all attributes:

StaffReturnsEquipment

R1(StaffID, EquipmentID, Date)

StaffBorrowEquipments

R1(StaffID, EquipmentID, Date)

TenantReturnsEquipment

R1(SIN, EquipmentID, Date)

TenantBorrowsEquipment

R1(SIN, EquipmentID, Date)

SQL

DDL (Create table):

```
CREATE TABLE PostalCode(  
    PostalCode VARCHAR PRIMARY KEY,  
    City VARCHAR NOT NULL,  
    Province VARCHAR NOT NULL  
);  
  
CREATE TABLE Building(  
    BuildingName VARCHAR PRIMARY KEY,  
    Address VARCHAR NOT NULL,  
    NumberOfStaff INTEGER NOT NULL,  
    City VARCHAR NOT NULL,  
    Province VARCHAR NOT NULL,  
    PostalCode VARCHAR FOREIGN KEY REFERENCES PostalCode(PostalCode)  
NOT NULL,  
    ON DELETE CASCADE  
);  
  
CREATE TABLE Tenant(  
    SIN INTEGER PRIMARY KEY,  
    TenantName VARCHAR NOT NULL,  
    Birthdate DATE NOT NULL,  
    Email VARCHAR  
);  
  
CREATE TABLE Room(  
    RoomNumber INTEGER PRIMARY KEY ,  
    BuildingName VARCHAR FOREIGN KEY REFERENCES Building(BuildingName),  
    CleanStatus BOOLEAN NOT NULL,  
    OccupancyLimit INTEGER NOT NULL,  
    NumberOfOccupants INTEGER NOT NULL  
);
```

```
CREATE TABLE Vehicle(
  VIN VARCHAR PRIMARY KEY,
  Make VARCHAR NOT NULL,
  Year INTEGER NOT NULL,
  InUse BOOLEAN NOT NULL,
  Model VARCHAR NOT NULL
);

CREATE TABLE Staff(
  StaffID INTEGER PRIMARY KEY,
  StaffName VARCHAR NOT NULL,
  VIN VARCHAR FOREIGN KEY REFERENCES Vehicle(VIN) DEFAULT NULL
);

CREATE TABLE Request(
  RequestID INTEGER PRIMARY KEY,
  BuildingName VARCHAR FOREIGN KEY REFERENCES Building(BuildingName),
  RoomNumber INT FOREIGN KEY REFERENCES Room(RoomNumber),
  RequestType VARCHAR NOT NULL,
  Complete BOOLEAN DEFAULT FALSE,
  SubmissionDate DATE NOT NULL,
  StaffID INTEGER FOREIGN KEY REFERENCES Staff(StaffID),
  SIN INTEGER FOREIGN KEY REFERENCES Tenant(SIN)
);

CREATE TABLE GameEquipment(
  EquipmentID INTEGER PRIMARY KEY,
  Cost FLOAT NOT NULL,
  LoanPeriod INTEGER NOT NULL,
  LateFee FLOAT NOT NULL,
  EquipmentName VARCHAR NOT NULL,
  NumberOfParts INTEGER NOT NULL
);
```

```
CREATE TABLE ToolsSuppliesEquipment(  
    EquipmentID INTEGER PRIMARY KEY,  
    Cost FLOAT NOT NULL,  
    LoanPeriod INTEGER NOT NULL,  
    LateFee FLOAT NOT NULL,  
    EquipmentName VARCHAR NOT NULL,  
    DateLastTested DATE NOT NULL  
);  
  
CREATE TABLE Mail(  
    MailID INTEGER PRIMARY KEY,  
    TenantName VARCHAR REFERENCES Tenant(TenantName) NOT NULL,  
    RoomNumber INTEGER FOREIGN KEY REFERENCES Room(RoomNumber) NOT  
NULL,  
    BuildingName VARCHAR FOREIGN KEY REFERENCES Building(BuildingName)  
NOT NULL,  
    MailType NOT NULL  
);  
  
CREATE TABLE TenantRentsRoom(  
    SIN INTEGER FOREIGN KEY REFERENCES Tenant(SIN),  
    RentalRate FLOAT NOT NULL,  
    ContractEndDate DATE NOT NULL,  
    RoomNumber INTEGER FOREIGN KEY REFERENCES Room(RoomNumber),  
    BuildingName VARCHAR FOREIGN KEY REFERENCES Building(BuildingName),  
    PRIMARY KEY (RoomNumber, BuildingName, SIN)  
);  
  
CREATE TABLE BuildingReceivesMail(  
    MailID INTEGER FOREIGN KEY REFERENCES Mail(MailID),  
    Date Received DATE,  
    PRIMARY KEY (MailID)  
);  
  
CREATE TABLE StaffReturnsEquipment(  
    Date DATE,  
    StaffID INTEGER FOREIGN KEY REFERENCES Staff(StaffID),  
    EquipmentID INTEGER FOREIGN KEY REFERENCES Equipment(EquipmentID),
```

```
PRIMARY KEY (StaffID, EquipmentID, Date)
);

CREATE TABLE StaffBorrowsEquipments(
    Date DATE,
    StaffID INTEGER FOREIGN KEY REFERENCES Staff(StaffID),
    EquipmentID INTEGER FOREIGN KEY REFERENCES Equipment(EquipmentID),
    PRIMARY KEY (StaffID, EquipmentID, Date)
);

CREATE TABLE TenantReturnsEquipment(
    Date DATE,
    SIN INTEGER FOREIGN KEY references Tenant(SIN),
    EquipmentID INTEGER FOREIGN KEY references Equipment(EquipmentID),
    PRIMARY KEY (EquipmentID, Date, SIN)
);

CREATE TABLE TenantBorrowsEquipment(
    Date DATE,
    SIN INTEGER FOREIGN KEY REFERENCES Tenant(SIN),
    EquipmentID INTEGER FOREIGN KEY REFERENCES Equipment(EquipmentID),
    PRIMARY KEY (EquipmentID, Date, SIN)
);
```

POPULATION (Insert statements):

```
INSERT INTO PostalCode (PostalCode, City, Province) VALUES
('10001', 'New York', 'NY'),
('90001', 'Los Angeles', 'CA'),
('60601', 'Chicago', 'IL'),
('77001', 'Houston', 'TX'),
('85001', 'Phoenix', 'AZ');
```

```
INSERT INTO Building (BuildingName, Address, NumberOfStaff, City,
Province, PostalCode) VALUES
('Building A', '123 Main St', 10, 'New York', 'NY', '10001'),
('Building B', '456 Market St', 15, 'Los Angeles', 'CA', '90001'),
('Building C', '789 Broad St', 20, 'Chicago', 'IL', '60601'),
('Building D', '101 Elm St', 5, 'Houston', 'TX', '77001'),
('Building E', '202 Oak St', 8, 'Phoenix', 'AZ', '85001');
```

```
INSERT INTO Tenant (SIN, TenantName, Birthdate, Email) VALUES
(123456789, 'John Doe', '1980-01-01', 'john.doe@email.com'),
(987654321, 'Jane Doe', '1990-02-02', 'jane.doe@email.com'),
(111222333, 'Mike Johnson', '1975-03-03', 'mike.j@email.com'),
(444555666, 'Sarah Smith', '1985-04-04', 'sarah.s@email.com'),
(777888999, 'Brian Lee', '1995-05-05', 'brian.lee@email.com');
```

```
INSERT INTO Room (RoomNumber, BuildingName, CleanStatus,
OccupancyLimit, NumberOfOccupants) VALUES
(101, 'Building A', TRUE, 2, 1),
(102, 'Building A', TRUE, 2, 2),
(201, 'Building B', FALSE, 3, 3),
(202, 'Building B', TRUE, 3, 2),
(301, 'Building C', FALSE, 4, 4);
```

```
INSERT INTO Vehicle (VIN, Make, Year, InUse, Model) VALUES
('1HGCM82633A123456', 'Honda', 2003, FALSE, 'Accord'),
('1HGCM82633A123457', 'Toyota', 2005, TRUE, 'Camry'),
('1HGCM82633A123458', 'Ford', 2010, FALSE, 'Focus'),
('1HGCM82633A123459', 'Chevrolet', 2015, TRUE, 'Malibu'),
('1HGCM82633A123450', 'Nissan', 2020, FALSE, 'Altima');
```


University of British Columbia, Vancouver

Department of Computer Science

```
INSERT INTO Staff (StaffID, StaffName, VIN) VALUES
```

```
(1, 'Alice Johnson', '1HGCM82633A123456'),  
(2, 'Bob Smith', '1HGCM82633A123457'),  
(3, 'Charlie Brown', '1HGCM82633A123458'),  
(4, 'Diana Ross', '1HGCM82633A123459'),  
(5, 'Edward Norton', '1HGCM82633A123450');
```

```
INSERT INTO Request (RequestID, BuildingName, RoomNumber,  
RequestType, SubmissionDate, StaffID, SIN) VALUES
```

```
(1, 'Building A', 101, 'Maintenance', '2023-01-01', 1, 123456789),  
(2, 'Building B', 201, 'Cleaning', '2023-02-01', 2, 987654321),  
(3, 'Building C', 301, 'Repair', '2023-03-01', 3, 111222333),  
(4, 'Building D', 101, 'Maintenance', '2023-04-01', 4, 444555666),  
(5, 'Building E', 202, 'Cleaning', '2023-05-01', 5, 777888999);
```

```
INSERT INTO GameEquipment (EquipmentID, Cost, LoanPeriod, LateFee,  
EquipmentName, NumberOfParts) VALUES
```

```
(1, 100.50, 7, 2.50, 'Board Game', 5),  
(2, 200.75, 14, 5.00, 'Video Game Console', 1),  
(3, 150.00, 7, 3.00, 'Card Game Set', 3),  
(4, 250.25, 30, 10.00, 'Sports Equipment', 10),  
(5, 50.00, 3, 1.00, 'Puzzle', 1);
```

```
INSERT INTO ToolsSuppliesEquipment (EquipmentID, Cost, LoanPeriod,  
LateFee, EquipmentName, DateLastTested) VALUES
```

```
(6, 300.00, 30, 15.00, 'Drill', '2023-01-01'),  
(7, 100.00, 7, 5.00, 'Saw', '2023-02-01'),  
(8, 150.00, 14, 7.50, 'Hammer', '2023-03-01'),  
(9, 50.00, 3, 2.50, 'Screwdriver', '2023-04-01'),  
(10, 200.00, 21, 10.00, 'Wrench', '2023-05-01');
```

```
INSERT INTO Mail (MailID, TenantName, RoomNumber, BuildingName,  
MailType) VALUES
```

```
(1, 'John Doe', 101, 'Building A', 'Package'),  
(2, 'Jane Doe', 201, 'Building B', 'Letter'),  
(3, 'Mike Johnson', 301, 'Building C', 'Postcard'),  
(4, 'Sarah Smith', 102, 'Building A', 'Magazine'),  
(5, 'Brian Lee', 202, 'Building B', 'Document');
```

University of British Columbia, Vancouver

Department of Computer Science

```
INSERT INTO TenantRentsRoom (SIN, RentalRate, ContractEndDate,
RoomNumber, BuildingName) VALUES
(123456789, 1000.00, '2024-01-01', 101, 'Building A'),
(987654321, 1200.00, '2024-02-02', 201, 'Building B'),
(111222333, 1100.00, '2024-03-03', 301, 'Building C'),
(444555666, 1050.00, '2024-04-04', 102, 'Building A'),
(777888999, 1250.00, '2024-05-05', 202, 'Building B');
```

```
INSERT INTO BuildingReceivesMail (MailID, DateReceived) VALUES
(1, '2023-01-01'),
(2, '2023-02-01'),
(3, '2023-03-01'),
(4, '2023-04-01'),
(5, '2023-05-01');
```

```
INSERT INTO StaffBorrowsEquipments (Date, StaffID, EquipmentID)
VALUES
('2023-01-01', 1, 1),
('2023-02-01', 2, 6),
('2023-03-01', 3, 2),
('2023-04-01', 4, 7),
('2023-05-01', 5, 3);
```

```
INSERT INTO StaffReturnsEquipment (Date, StaffID, EquipmentID) VALUES
('2023-01-10', 1, 1),
('2023-02-10', 2, 6),
('2023-03-10', 3, 2),
('2023-04-10', 4, 7),
('2023-05-10', 5, 3);
```

```
INSERT INTO TenantReturnsEquipment (Date, SIN, EquipmentID) VALUES
('2023-01-15', 123456789, 4),
('2023-02-15', 987654321, 8),
('2023-03-15', 111222333, 5),
('2023-04-15', 444555666, 9),
('2023-05-15', 777888999, 10);
```