University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 4

Date: November 29th, 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	psnovais@student.ubc.ca

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

University of British Columbia, Vancouver
Department of Computer Science

***Description*:**

Our project for CPSC 304 is a database for the purpose of living quarters management (examples include apartments, or condos). This would include (but is not limited to) logging and referencing maintenance requests, current renters in their respective units, mail services and tracking of such, supplies on hand, and room statuses. Authorized people using our database will be able to view this data and add to the database. Authorization is done using string matching on private fields as well as implementing a maximum login attempts variable to encourage safe coding principles.

***Schema*:**

- Addition of triggers in mail and request
- Renaming of tables for brevity (primarily in the borrowing and returning relations)
- Added new tables for borrowing from previous schema (it was an ISA so we added the borrowing and returning of both types of supplies which were games and tools)
    - StaffGameBorrowing
    - StaffGameReturning
    - StaffToolSuppliesBorrowing
    - StaffToolSuppliesReturning
    - TenantGameBorrowing
    - TenantGameReturning
    - TenantToolSuppliesBorrowing
    - TenantToolSuppliesReturning

Below you will see the schema produced by oracle db (basically the schema that our code results in). Screenshots can be found below the schema.

SCHEMA START:
Building(BuildingName, Address, NumberOfStaff, City, Province, **Postalcode**)

BuildingRecievesMail(**MailID**, DateRecieved)

GameEquipment(EquipmentID, Cost, LoadPeriod, LateFee, EquipmentName, NumberOfParts)

Mail(MailID, **TenantSIN**, **RoomNumber**, **BuildingName**, MailType)

PostalCode(Postalcode, City, Province)

Request(RequestID, **BuildingName**, **RoomNumber**, RequestType, Complete, SubmissionDate, **StaffID**, **SIN**)

Room(RoomNumber, **BuildingName**, CleanStatus, OccupancyLimit, NumberOfOccupants)

Staff(StaffID, StaffName, **VIN**)

StaffGameBorrowing(BorrowDate, **StaffID**, **EquipmentID**)

StaffGameReturning(ReturnDate, **StaffID**, **EquipmentID**)

StaffToolSuppliesBorrowing(ReturnDate, **StaffID**, **EquipmentID**)

StaffToolSuppliesReturning(ReturnDate, **StaffID**, **EquipmentID**)

Tenant(SIN, TenantName, Birthdate, email)

TenantGameBorrowing(BorrowDate, **SIN**, **EquipmentID**)

TenantGameReturning(ReturnDate, **SIN**, **EquipmentID**)

TenantRentsRoom(**SIN**, RentalRate, ContractEndDate, **RoomNumber**, **BuildingName**)

TenantToolSuppliesBorrowing(BorrowDate, **SIN**, **EquipmentID**)

TenantToolSuppliesReturning(ReturnDate, **SIN**, **EquipmentID**)

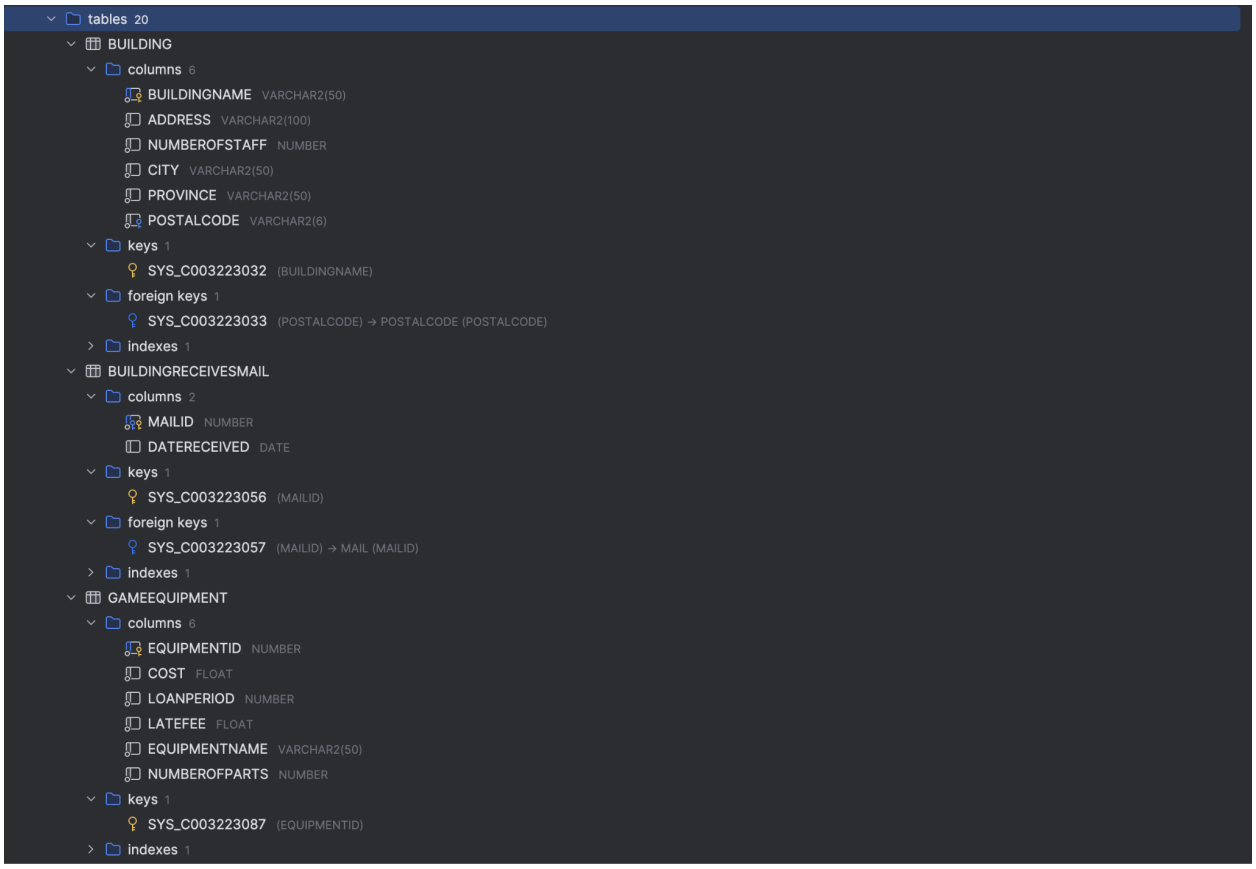
ToolSuppliesEquipment(EquipmentID, Cost, LoanPeriod, LateFee, EquipmentName, DateLastTested)

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

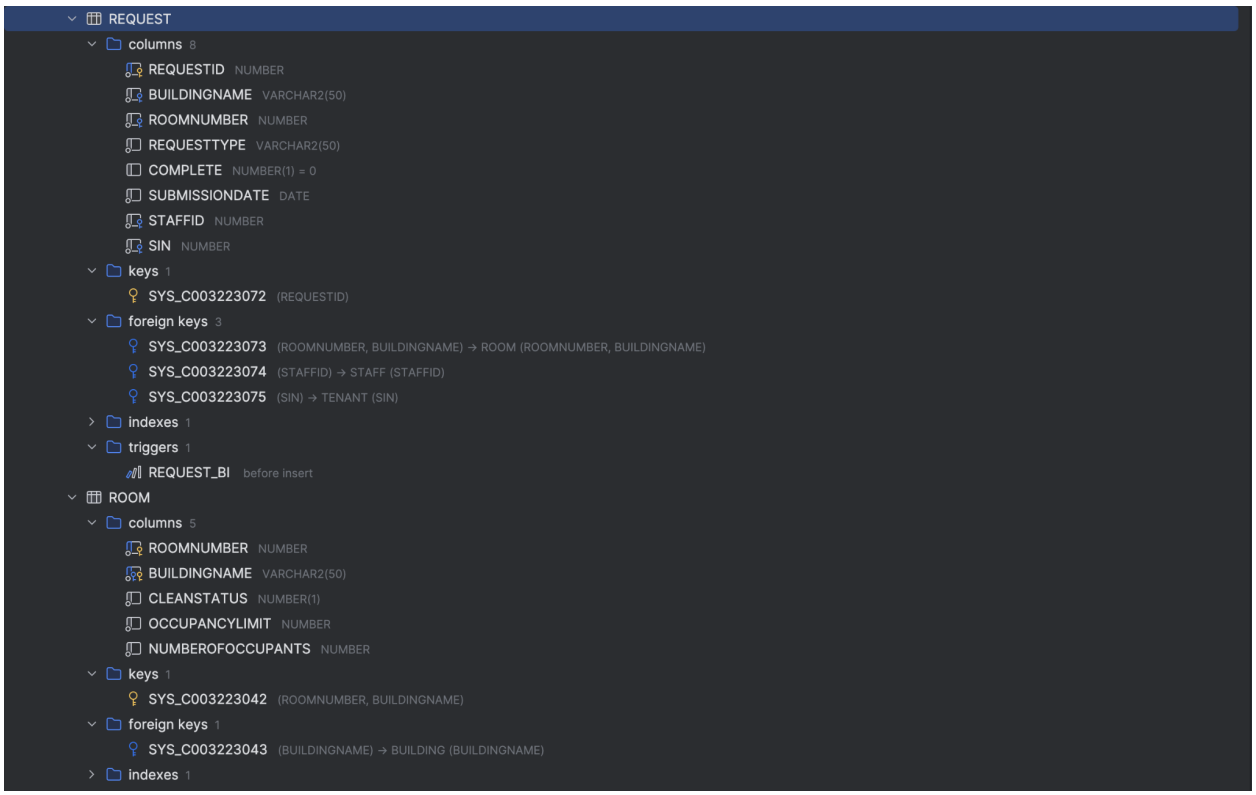
University of British Columbia, Vancouver
Department of Computer Science

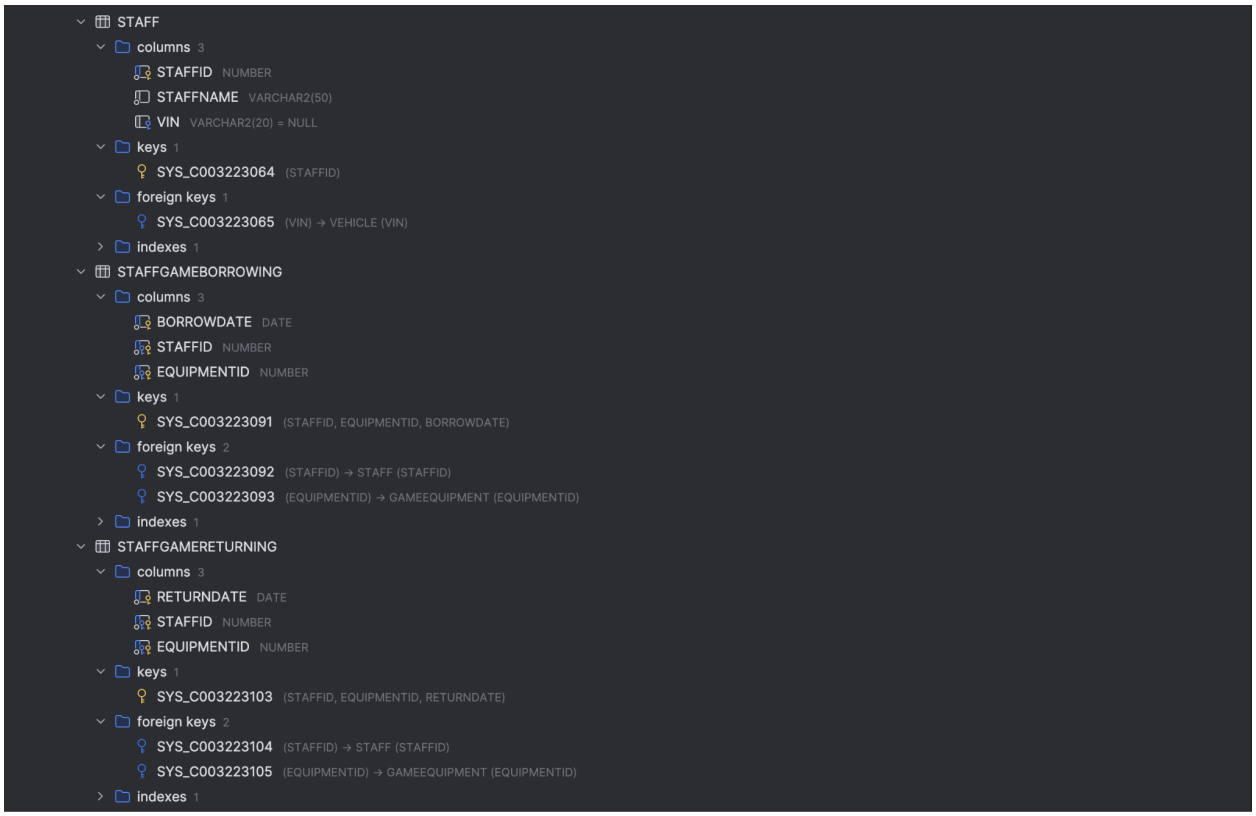
SCREENSHOTS:

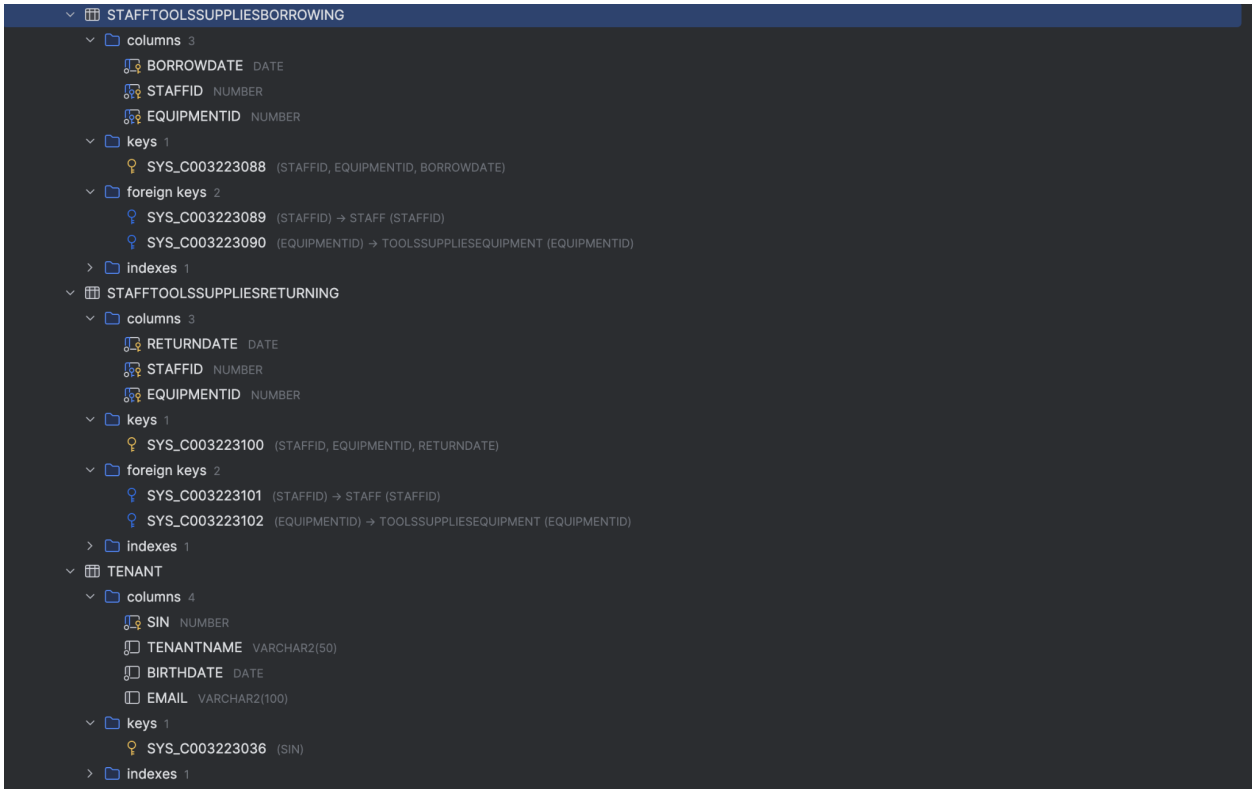
All tables

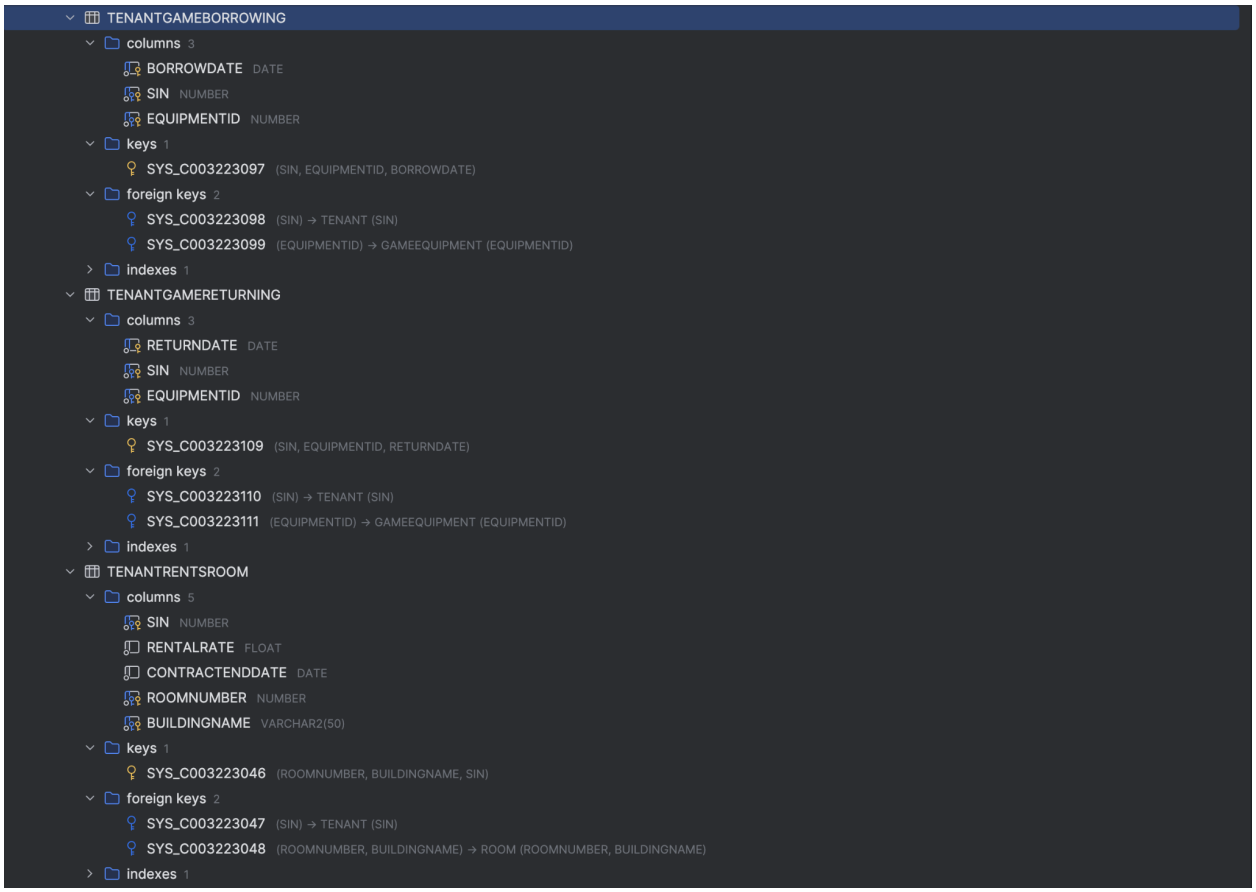


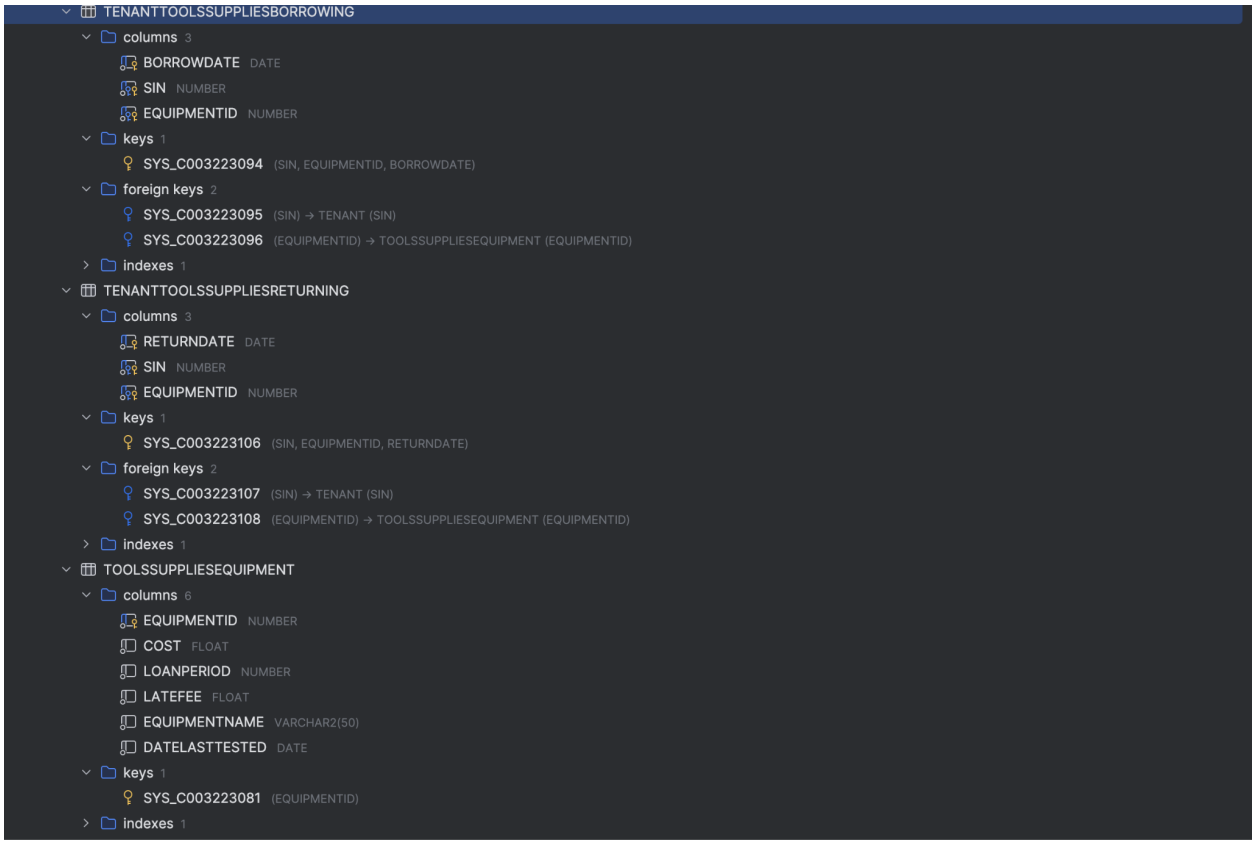

MAIL
columns 5
MAILID NUMBER
TENANTSIN NUMBER
ROOMNUMBER NUMBER
BUILDINGNAME VARCHAR2(50)
MAILTYPE VARCHAR2(50)
keys 1
SYS_C003223053 (MAILID)
foreign keys 2
SYS_C003223054 (TENANTSIN) → TENANT (SIN)
SYS_C003223055 (ROOMNUMBER, BUILDINGNAME) → ROOM (ROOMNUMBER, BUILDINGNAME)
indexes 1
triggers 1
MAIL_BI before insert
POSTALCODE
columns 3
POSTALCODE VARCHAR2(6)
CITY VARCHAR2(50)
PROVINCE VARCHAR2(50)
keys 1
SYS_C003223026 (POSTALCODE)
indexes 1


REQUEST
columns 8
REQUESTID NUMBER
BUILDINGNAME VARCHAR2(50)
ROOMNUMBER NUMBER
REQUESTTYPE VARCHAR2(50)
COMPLETE NUMBER(1) = 0
SUBMISSIONDATE DATE
STAFFID NUMBER
SIN NUMBER
keys 1
SYS_C003223072 (REQUESTID)
foreign keys 3
SYS_C003223073 (ROOMNUMBER, BUILDINGNAME) → ROOM (ROOMNUMBER, BUILDINGNAME)
SYS_C003223074 (STAFFID) → STAFF (STAFFID)
SYS_C003223075 (SIN) → TENANT (SIN)
indexes 1
triggers 1
REQUEST_BI before insert
ROOM
columns 5
ROOMNUMBER NUMBER
BUILDINGNAME VARCHAR2(50)
CLEANSTATUS NUMBER(1)
OCCUPANCYLIMIT NUMBER
NUMBEROFOCCUPANTS NUMBER
keys 1
SYS_C003223042 (ROOMNUMBER, BUILDINGNAME)
foreign keys 1
SYS_C003223043 (BUILDINGNAME) → BUILDING (BUILDINGNAME)
indexes 1


STAFF
columns 3
STAFFID NUMBER
STAFFNAME VARCHAR2(50)
VIN VARCHAR2(20) = NULL
keys 1
SYS_C003223064 (STAFFID)
foreign keys 1
SYS_C003223065 (VIN) → VEHICLE (VIN)
indexes 1
STAFFGAMEBORROWING
columns 3
BORROWDATE DATE
STAFFID NUMBER
EQUIPMENTID NUMBER
keys 1
SYS_C003223091 (STAFFID, EQUIPMENTID, BORROWDATE)
foreign keys 2
SYS_C003223092 (STAFFID) → STAFF (STAFFID)
SYS_C003223093 (EQUIPMENTID) → GAMEEQUIPMENT (EQUIPMENTID)
indexes 1
STAFFGAMERETURNING
columns 3
RETURNDATE DATE
STAFFID NUMBER
EQUIPMENTID NUMBER
keys 1
SYS_C003223103 (STAFFID, EQUIPMENTID, RETURNDATE)
foreign keys 2
SYS_C003223104 (STAFFID) → STAFF (STAFFID)
SYS_C003223105 (EQUIPMENTID) → GAMEEQUIPMENT (EQUIPMENTID)
indexes 1


STAFFTOOLSSUPPLIESBORROWING
columns 3
BORROWDATE DATE
STAFFID NUMBER
EQUIPMENTID NUMBER
keys 1
SYS_C003223088 (STAFFID, EQUIPMENTID, BORROWDATE)
foreign keys 2
SYS_C003223089 (STAFFID) → STAFF (STAFFID)
SYS_C003223090 (EQUIPMENTID) → TOOLSSUPPLIESEQUIPMENT (EQUIPMENTID)
indexes 1
STAFFTOOLSSUPPLIESRETURNING
columns 3
RETURNDATE DATE
STAFFID NUMBER
EQUIPMENTID NUMBER
keys 1
SYS_C003223100 (STAFFID, EQUIPMENTID, RETURNDATE)
foreign keys 2
SYS_C003223101 (STAFFID) → STAFF (STAFFID)
SYS_C003223102 (EQUIPMENTID) → TOOLSSUPPLIESEQUIPMENT (EQUIPMENTID)
indexes 1
TENANT
columns 4
SIN NUMBER
TENANTNAME VARCHAR2(50)
BIRTHDATE DATE
EMAIL VARCHAR2(100)
keys 1
SYS_C003223036 (SIN)
indexes 1


TENANTGAMEBORROWING
columns 3
BORROWDATE DATE
SIN NUMBER
EQUIPMENTID NUMBER
keys 1
SYS_C003223097 (SIN, EQUIPMENTID, BORROWDATE)
foreign keys 2
SYS_C003223098 (SIN) → TENANT (SIN)
SYS_C003223099 (EQUIPMENTID) → GAMEEQUIPMENT (EQUIPMENTID)
indexes 1
TENANTGAMERETURNING
columns 3
RETURNDATE DATE
SIN NUMBER
EQUIPMENTID NUMBER
keys 1
SYS_C003223109 (SIN, EQUIPMENTID, RETURNDATE)
foreign keys 2
SYS_C003223110 (SIN) → TENANT (SIN)
SYS_C003223111 (EQUIPMENTID) → GAMEEQUIPMENT (EQUIPMENTID)
indexes 1
TENANTRENTSROOM
columns 5
SIN NUMBER
RENTALRATE FLOAT
CONTRACTENDDATE DATE
ROOMNUMBER NUMBER
BUILDINGNAME VARCHAR2(50)
keys 1
SYS_C003223046 (ROOMNUMBER, BUILDINGNAME, SIN)
foreign keys 2
SYS_C003223047 (SIN) → TENANT (SIN)
SYS_C003223048 (ROOMNUMBER, BUILDINGNAME) → ROOM (ROOMNUMBER, BUILDINGNAME)
indexes 1



SQL Script (can be found in databaseSetup.sql in src/sql for a cleaner look):

```
-- Check if the MailIDSeq sequence exists and then drop it
BEGIN
   EXECUTE IMMEDIATE 'DROP SEQUENCE MailIDSeq';
EXCEPTION
   WHEN OTHERS THEN
       IF SQLCODE != -2289 THEN
           RAISE;
       END IF;
END;

-- Check if the RequestIDSeq sequence exists and then drop it
BEGIN
   EXECUTE IMMEDIATE 'DROP SEQUENCE RequestIDSeq';
EXCEPTION
   WHEN OTHERS THEN
       IF SQLCODE != -2289 THEN
```

```
            RAISE;
        END IF;
END;

-- Create a sequence
CREATE SEQUENCE MailIDSeq
    START WITH 10000
    INCREMENT BY 1
    NOCACHE
    NOCYCLE;

-- Create a sequence
CREATE SEQUENCE RequestIDSeq
    START WITH 50000
    INCREMENT BY 1
    NOCACHE
    NOCYCLE;


BEGIN
    FOR t IN (SELECT table_name FROM user_tables) LOOP
            EXECUTE IMMEDIATE 'DROP TABLE ' || t.table_name || ' CASCADE
CONSTRAINTS';
        END LOOP;
END;

CREATE TABLE PostalCode(PostalCode VARCHAR(6) PRIMARY KEY, City VARCHAR(50) NOT
NULL, Province VARCHAR(50) NOT NULL);

CREATE TABLE Building(BuildingName VARCHAR(50) PRIMARY KEY, Address
VARCHAR(100) NOT NULL, NumberOfStaff INTEGER NOT NULL, City VARCHAR(50) NOT
NULL, Province VARCHAR(50) NOT NULL, PostalCode VARCHAR(6) NOT NULL, FOREIGN
KEY (PostalCode) REFERENCES PostalCode(PostalCode) ON DELETE CASCADE);

CREATE TABLE Tenant(SIN INTEGER PRIMARY KEY, TenantName VARCHAR(50) NOT NULL,
Birthdate DATE NOT NULL, Email VARCHAR(100));

CREATE TABLE Room(RoomNumber INTEGER NOT NULL, BuildingName VARCHAR(50) NOT
NULL, CleanStatus NUMBER(1) NOT NULL, OccupancyLimit INTEGER NOT NULL,
NumberOfOccupants INTEGER NOT NULL, PRIMARY KEY (RoomNumber, BuildingName),
FOREIGN KEY (BuildingName) REFERENCES Building(BuildingName));

CREATE TABLE TenantRentsRoom(SIN INTEGER, RentalRate FLOAT NOT NULL,
ContractEndDate DATE NOT NULL, RoomNumber INTEGER, BuildingName VARCHAR(50),
PRIMARY KEY (RoomNumber, BuildingName, SIN), FOREIGN KEY (SIN) REFERENCES
Tenant(SIN), FOREIGN KEY (RoomNumber, BuildingName) REFERENCES Room(RoomNumber,
BuildingName));

CREATE TABLE Mail(MailID INTEGER PRIMARY KEY, TenantSIN INTEGER NOT NULL,
RoomNumber INTEGER NOT NULL, BuildingName VARCHAR(50) NOT NULL, MailType
VARCHAR(50) NOT NULL, FOREIGN KEY (TenantSIN) REFERENCES Tenant(SIN), FOREIGN
KEY (RoomNumber, BuildingName) REFERENCES Room(RoomNumber, BuildingName));

-- Trigger to automatically assign MailID values from the sequence
CREATE OR REPLACE TRIGGER Mail_BI
    BEFORE INSERT ON Mail
```

```
   FOR EACH ROW
BEGIN
   SELECT MailIDSeq.NEXTVAL INTO :NEW.MailID FROM DUAL;
END;


CREATE TABLE BuildingReceivesMail(MailID INTEGER, DateReceived DATE, PRIMARY
KEY (MailID), FOREIGN KEY (MailID) REFERENCES Mail(MailID));

CREATE TABLE Vehicle(VIN VARCHAR(20) PRIMARY KEY, Make VARCHAR(50) NOT NULL,
Year INTEGER NOT NULL, InUse NUMBER(1) NOT NULL, Model VARCHAR(50) NOT NULL);

CREATE TABLE Staff(StaffID INTEGER PRIMARY KEY, StaffName VARCHAR(50) NOT NULL,
VIN VARCHAR(20) DEFAULT NULL, FOREIGN KEY (VIN) REFERENCES Vehicle(VIN));

CREATE TABLE Request(RequestID INTEGER PRIMARY KEY, BuildingName VARCHAR(50)
NOT NULL, RoomNumber INT NOT NULL, RequestType VARCHAR(50) NOT NULL, Complete
NUMBER(1) DEFAULT 0, SubmissionDate DATE NOT NULL, StaffID INTEGER NOT NULL,
SIN INTEGER NOT NULL, FOREIGN KEY (RoomNumber, BuildingName) REFERENCES
Room(RoomNumber, BuildingName), FOREIGN KEY (StaffID) REFERENCES
Staff(StaffID), FOREIGN KEY (SIN) REFERENCES Tenant(SIN));

-- Trigger to automatically assign MailID values from the sequence
CREATE OR REPLACE TRIGGER Request_BI
   BEFORE INSERT ON Request
   FOR EACH ROW
BEGIN
   SELECT RequestIDSeq.NEXTVAL INTO :NEW.RequestID FROM DUAL;
END;

CREATE TABLE ToolsSuppliesEquipment(EquipmentID INTEGER PRIMARY KEY, Cost FLOAT
NOT NULL, LoanPeriod INTEGER NOT NULL, LateFee FLOAT NOT NULL, EquipmentName
VARCHAR(50) NOT NULL, DateLastTested DATE NOT NULL);

CREATE TABLE GameEquipment(EquipmentID INTEGER PRIMARY KEY, Cost FLOAT NOT
NULL, LoanPeriod INTEGER NOT NULL, LateFee FLOAT NOT NULL, EquipmentName
VARCHAR(50) NOT NULL, NumberOfParts INTEGER NOT NULL);

CREATE TABLE StaffToolsSuppliesBorrowing(BorrowDate DATE, StaffID INTEGER,
EquipmentID INTEGER, PRIMARY KEY (StaffID, EquipmentID, BorrowDate), FOREIGN
KEY (StaffID) REFERENCES Staff(StaffID), FOREIGN KEY (EquipmentID) REFERENCES
ToolsSuppliesEquipment(EquipmentID));

CREATE TABLE StaffGameBorrowing(BorrowDate DATE, StaffID INTEGER, EquipmentID
INTEGER, PRIMARY KEY (StaffID, EquipmentID, BorrowDate), FOREIGN KEY (StaffID)
REFERENCES Staff(StaffID), FOREIGN KEY (EquipmentID) REFERENCES
GameEquipment(EquipmentID));

CREATE TABLE TenantToolsSuppliesBorrowing(BorrowDate DATE, SIN INTEGER,
EquipmentID INTEGER, PRIMARY KEY (SIN, EquipmentID, BorrowDate), FOREIGN KEY
(SIN) REFERENCES Tenant(SIN), FOREIGN KEY (EquipmentID) REFERENCES
ToolsSuppliesEquipment(EquipmentID));

CREATE TABLE TenantGameBorrowing(BorrowDate DATE, SIN INTEGER, EquipmentID
INTEGER, PRIMARY KEY (SIN, EquipmentID, BorrowDate), FOREIGN KEY (SIN)
```

```
REFERENCES Tenant(SIN), FOREIGN KEY (EquipmentID) REFERENCES
GameEquipment(EquipmentID));

CREATE TABLE StaffToolsSuppliesReturning(ReturnDate DATE, StaffID INTEGER,
EquipmentID INTEGER, PRIMARY KEY (StaffID, EquipmentID, ReturnDate), FOREIGN
KEY (StaffID) REFERENCES Staff(StaffID), FOREIGN KEY (EquipmentID) REFERENCES
ToolsSuppliesEquipment(EquipmentID));

CREATE TABLE StaffGameReturning(ReturnDate DATE, StaffID INTEGER, EquipmentID
INTEGER, PRIMARY KEY (StaffID, EquipmentID, ReturnDate), FOREIGN KEY (StaffID)
REFERENCES Staff(StaffID), FOREIGN KEY (EquipmentID) REFERENCES
GameEquipment(EquipmentID));

CREATE TABLE TenantToolsSuppliesReturning(ReturnDate DATE, SIN INTEGER,
EquipmentID INTEGER, PRIMARY KEY (SIN, EquipmentID, ReturnDate), FOREIGN KEY
(SIN) REFERENCES Tenant(SIN), FOREIGN KEY (EquipmentID) REFERENCES
ToolsSuppliesEquipment(EquipmentID));

CREATE TABLE TenantGameReturning(ReturnDate DATE, SIN INTEGER, EquipmentID
INTEGER, PRIMARY KEY (SIN, EquipmentID, ReturnDate), FOREIGN KEY (SIN)
REFERENCES Tenant(SIN), FOREIGN KEY (EquipmentID) REFERENCES
GameEquipment(EquipmentID));


INSERT INTO PostalCode (PostalCode, City, Province) VALUES ('V6T1Z2',
'Vancouver', 'BC');
INSERT INTO PostalCode (PostalCode, City, Province) VALUES ('V6T1K2',
'Vancouver', 'BC');
INSERT INTO PostalCode (PostalCode, City, Province) VALUES ('V6T1W9',
'Vancouver', 'BC');
INSERT INTO PostalCode (PostalCode, City, Province) VALUES ('V6T1Z4',
'Vancouver', 'BC');
INSERT INTO PostalCode (PostalCode, City, Province) VALUES ('V6T2G9',
'Vancouver', 'BC');

INSERT INTO Building (BuildingName, Address, NumberOfStaff, City, Province,
PostalCode) VALUES ('Ponderosa', '2075 West Mall', 10, 'Vancouver', 'BC',
'V6T1Z2');
INSERT INTO Building (BuildingName, Address, NumberOfStaff, City, Province,
PostalCode) VALUES ('Walter Gage', '5959 Student Union Blvd', 20, 'Vancouver',
'BC', 'V6T1K2');
INSERT INTO Building (BuildingName, Address, NumberOfStaff, City, Province,
PostalCode) VALUES ('Totem Park', '2525 West Mall', 30, 'Vancouver', 'BC',
'V6T1W9');
INSERT INTO Building (BuildingName, Address, NumberOfStaff, City, Province,
PostalCode) VALUES ('Marine Drive', '2205 Lower Mall', 40, 'Vancouver', 'BC',
'V6T1Z4');
INSERT INTO Building (BuildingName, Address, NumberOfStaff, City, Province,
PostalCode) VALUES ('Thunderbird', '6335 Thunderbird Crescent', 50,
'Vancouver', 'BC', 'V6T2G9');

INSERT INTO Tenant (SIN, TenantName, Birthdate, Email) VALUES (1001, 'John
Doe', TO_DATE('1990-01-01', 'YYYY-MM-DD'), 'john.doe@example.com');
INSERT INTO Tenant (SIN, TenantName, Birthdate, Email) VALUES (1002, 'Jane
Smith', TO_DATE('1992-02-02', 'YYYY-MM-DD'), 'jane.smith@example.com');
```

```
INSERT INTO Tenant (SIN, TenantName, Birthdate, Email) VALUES (1003, 'Alice
Johnson', TO_DATE('1994-03-03', 'YYYY-MM-DD'), 'alice.johnson@example.com');
INSERT INTO Tenant (SIN, TenantName, Birthdate, Email) VALUES (1004, 'Bob
Brown', TO_DATE('1996-04-04', 'YYYY-MM-DD'), 'bob.brown@example.com');
INSERT INTO Tenant (SIN, TenantName, Birthdate, Email) VALUES (1005, 'Charlie
Davis', TO_DATE('1998-05-05', 'YYYY-MM-DD'), 'charlie.davis@example.com');

INSERT INTO Room (RoomNumber, BuildingName, CleanStatus, OccupancyLimit,
NumberOfOccupants) VALUES (101, 'Ponderosa', 1, 4, 3);
INSERT INTO Room (RoomNumber, BuildingName, CleanStatus, OccupancyLimit,
NumberOfOccupants) VALUES (102, 'Ponderosa', 0, 1, 1);
INSERT INTO Room (RoomNumber, BuildingName, CleanStatus, OccupancyLimit,
NumberOfOccupants) VALUES (101, 'Marine Drive', 0, 2, 2);
INSERT INTO Room (RoomNumber, BuildingName, CleanStatus, OccupancyLimit,
NumberOfOccupants) VALUES (102, 'Marine Drive', 0, 4, 4);
INSERT INTO Room (RoomNumber, BuildingName, CleanStatus, OccupancyLimit,
NumberOfOccupants) VALUES (101, 'Thunderbird', 1, 4, 4);
INSERT INTO Room (RoomNumber, BuildingName, CleanStatus, OccupancyLimit,
NumberOfOccupants) VALUES (102, 'Walter Gage', 1, 4, 4);
INSERT INTO Room (RoomNumber, BuildingName, CleanStatus, OccupancyLimit,
NumberOfOccupants) VALUES (101, 'Totem Park', 1, 2, 2);


INSERT INTO Vehicle (VIN, Make, Year, InUse, Model) VALUES ('VIN001', 'Toyota',
2010, 1, 'Prius');
INSERT INTO Vehicle (VIN, Make, Year, InUse, Model) VALUES ('VIN002', 'Toyota',
2011, 0, 'Corolla');
INSERT INTO Vehicle (VIN, Make, Year, InUse, Model) VALUES ('VIN003', 'Tesla',
2020, 1, 'S');
INSERT INTO Vehicle (VIN, Make, Year, InUse, Model) VALUES ('VIN004', 'Tesla',
2021, 0, '3');
INSERT INTO Vehicle (VIN, Make, Year, InUse, Model) VALUES ('VIN005', 'Tesla',
2020, 1, 'X');

INSERT INTO Staff (StaffID, StaffName, VIN) VALUES (2001, 'Elon Musk',
'VIN003');
INSERT INTO Staff (StaffID, StaffName, VIN) VALUES (2002, 'Elon Ma', 'VIN004');
INSERT INTO Staff (StaffID, StaffName, VIN) VALUES (2003, 'Jeff Bay',
'VIN002');
INSERT INTO Staff (StaffID, StaffName, VIN) VALUES (2004, 'Gregor Karl',
'VIN001');
INSERT INTO Staff (StaffID, StaffName, VIN) VALUES (2005, 'John Doe',
'VIN005');

INSERT INTO ToolsSuppliesEquipment (EquipmentID, Cost, LoanPeriod, LateFee,
EquipmentName, DateLastTested) VALUES (3001, 100.0, 2, 5.0, 'Power Drill',
TO_DATE('2023-11-21', 'YYYY-MM-DD'));
INSERT INTO ToolsSuppliesEquipment (EquipmentID, Cost, LoanPeriod, LateFee,
EquipmentName, DateLastTested) VALUES (3002, 100.0, 1, 10.0, 'Basic Vacuum',
TO_DATE('2023-09-21', 'YYYY-MM-DD'));
INSERT INTO ToolsSuppliesEquipment (EquipmentID, Cost, LoanPeriod, LateFee,
EquipmentName, DateLastTested) VALUES (3003, 200.0, 1, 15.0, 'Dyson Vacuum',
TO_DATE('2023-10-21', 'YYYY-MM-DD'));
INSERT INTO ToolsSuppliesEquipment (EquipmentID, Cost, LoanPeriod, LateFee,
EquipmentName, DateLastTested) VALUES (3004, 50.0, 3, 20.0, 'Flat-head
Screwdriver', TO_DATE('2023-11-21', 'YYYY-MM-DD'));
```

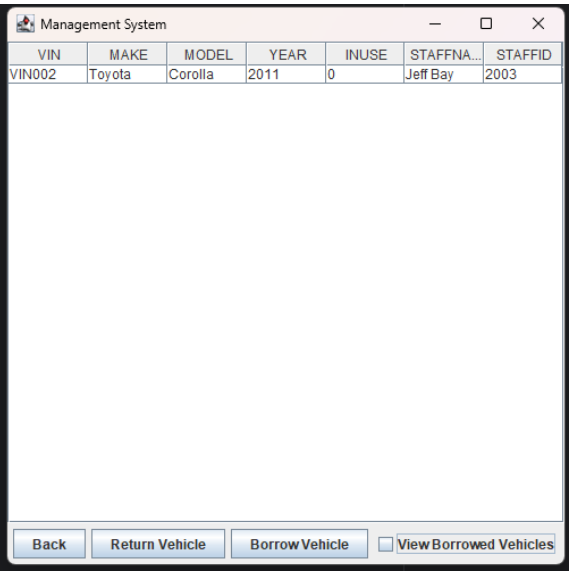
```
INSERT INTO ToolsSuppliesEquipment (EquipmentID, Cost, LoanPeriod, LateFee,
EquipmentName, DateLastTested) VALUES (3005, 65.0, 3, 25.0, 'Star-head
Screwdriver', TO_DATE('2021-09-01', 'YYYY-MM-DD'));

INSERT INTO GameEquipment (EquipmentID, Cost, LoanPeriod, LateFee,
EquipmentName, NumberOfParts) VALUES (4001, 250.0, 5, 10.0, 'Chess Set', 32);
INSERT INTO GameEquipment (EquipmentID, Cost, LoanPeriod, LateFee,
EquipmentName, NumberOfParts) VALUES (4002, 500.0, 1, 20.0, 'Pool Table Set',
16);
INSERT INTO GameEquipment (EquipmentID, Cost, LoanPeriod, LateFee,
EquipmentName, NumberOfParts) VALUES (4003, 150.0, 1, 5.0, 'Soccer Ball', 1);
INSERT INTO GameEquipment (EquipmentID, Cost, LoanPeriod, LateFee,
EquipmentName, NumberOfParts) VALUES (4004, 300.0, 2, 15.0, 'Tennis Racket',
2);
INSERT INTO GameEquipment (EquipmentID, Cost, LoanPeriod, LateFee,
EquipmentName, NumberOfParts) VALUES (4005, 350.0, 2, 12.0, 'Basketball', 1);

INSERT INTO StaffToolsSuppliesBorrowing (BorrowDate, StaffID, EquipmentID)
VALUES (TO_DATE('2023-11-01', 'YYYY-MM-DD'), 2001, 3001);
INSERT INTO StaffToolsSuppliesBorrowing (BorrowDate, StaffID, EquipmentID)
VALUES (TO_DATE('2023-11-02', 'YYYY-MM-DD'), 2002, 3002);
INSERT INTO StaffToolsSuppliesBorrowing (BorrowDate, StaffID, EquipmentID)
VALUES (TO_DATE('2023-11-03', 'YYYY-MM-DD'), 2003, 3003);
INSERT INTO StaffToolsSuppliesBorrowing (BorrowDate, StaffID, EquipmentID)
VALUES (TO_DATE('2022-11-04', 'YYYY-MM-DD'), 2004, 3001);
INSERT INTO StaffToolsSuppliesBorrowing (BorrowDate, StaffID, EquipmentID)
VALUES (TO_DATE('2023-11-05', 'YYYY-MM-DD'), 2005, 3001);

INSERT INTO TenantGameBorrowing (BorrowDate, SIN, EquipmentID) VALUES
(TO_DATE('2023-11-01', 'YYYY-MM-DD'), 1001, 4004);
INSERT INTO TenantGameBorrowing (BorrowDate, SIN, EquipmentID) VALUES
(TO_DATE('2023-11-02', 'YYYY-MM-DD'), 1002, 4004);
INSERT INTO TenantGameBorrowing (BorrowDate, SIN, EquipmentID) VALUES
(TO_DATE('2023-11-03', 'YYYY-MM-DD'), 1003, 4004);
INSERT INTO TenantGameBorrowing (BorrowDate, SIN, EquipmentID) VALUES
(TO_DATE('2023-11-01', 'YYYY-MM-DD'), 1004, 4005);
INSERT INTO TenantGameBorrowing (BorrowDate, SIN, EquipmentID) VALUES
(TO_DATE('2023-11-02', 'YYYY-MM-DD'), 1005, 4001);

INSERT INTO StaffToolsSuppliesReturning (ReturnDate, StaffID, EquipmentID)
VALUES (TO_DATE('2023-11-03', 'YYYY-MM-DD'), 2001, 3001);
INSERT INTO StaffToolsSuppliesReturning (ReturnDate, StaffID, EquipmentID)
VALUES (TO_DATE('2023-11-03', 'YYYY-MM-DD'), 2002, 3002);
INSERT INTO StaffToolsSuppliesReturning (ReturnDate, StaffID, EquipmentID)
VALUES (TO_DATE('2023-11-04', 'YYYY-MM-DD'), 2003, 3003);
INSERT INTO StaffToolsSuppliesReturning (ReturnDate, StaffID, EquipmentID)
VALUES (TO_DATE('2023-11-05', 'YYYY-MM-DD'), 2004, 3001);
INSERT INTO StaffToolsSuppliesReturning (ReturnDate, StaffID, EquipmentID)
VALUES (TO_DATE('2023-11-06', 'YYYY-MM-DD'), 2005, 3001);

INSERT INTO TenantToolsSuppliesReturning (ReturnDate, SIN, EquipmentID) VALUES
(TO_DATE('2023-11-02', 'YYYY-MM-DD'), 1001, 3004);
INSERT INTO TenantToolsSuppliesReturning (ReturnDate, SIN, EquipmentID) VALUES
(TO_DATE('2023-11-03', 'YYYY-MM-DD'), 1002, 3004);
INSERT INTO TenantToolsSuppliesReturning (ReturnDate, SIN, EquipmentID) VALUES
(TO_DATE('2023-11-04', 'YYYY-MM-DD'), 1003, 3004);
```

```
INSERT INTO TenantToolsSuppliesReturning (ReturnDate, SIN, EquipmentID) VALUES
(TO_DATE('2023-11-02', 'YYYY-MM-DD'), 1004, 3005);
INSERT INTO TenantToolsSuppliesReturning (ReturnDate, SIN, EquipmentID) VALUES
(TO_DATE('2023-11-06', 'YYYY-MM-DD'), 1005, 3001);

INSERT INTO TenantGameReturning (ReturnDate, SIN, EquipmentID) VALUES
(TO_DATE('2023-11-02', 'YYYY-MM-DD'), 1001, 4004);
INSERT INTO TenantGameReturning (ReturnDate, SIN, EquipmentID) VALUES
(TO_DATE('2023-11-03', 'YYYY-MM-DD'), 1002, 4004);
INSERT INTO TenantGameReturning (ReturnDate, SIN, EquipmentID) VALUES
(TO_DATE('2023-11-04', 'YYYY-MM-DD'), 1003, 4004);
INSERT INTO TenantGameReturning (ReturnDate, SIN, EquipmentID) VALUES
(TO_DATE('2023-11-02', 'YYYY-MM-DD'), 1004, 4005);
INSERT INTO TenantGameReturning (ReturnDate, SIN, EquipmentID) VALUES
(TO_DATE('2023-11-06', 'YYYY-MM-DD'), 1005, 4001);

INSERT INTO TenantRentsRoom (SIN, RentalRate, ContractEndDate, RoomNumber,
BuildingName) VALUES (1001, 1200.0, TO_DATE('2024-01-01', 'YYYY-MM-DD'), 101,
'Ponderosa');
INSERT INTO TenantRentsRoom (SIN, RentalRate, ContractEndDate, RoomNumber,
BuildingName) VALUES (1002, 1000.0, TO_DATE('2024-01-01', 'YYYY-MM-DD'), 102,
'Ponderosa');
INSERT INTO TenantRentsRoom (SIN, RentalRate, ContractEndDate, RoomNumber,
BuildingName) VALUES (1003, 1100.0, TO_DATE('2023-12-31', 'YYYY-MM-DD'), 101,
'Marine Drive');
INSERT INTO TenantRentsRoom (SIN, RentalRate, ContractEndDate, RoomNumber,
BuildingName) VALUES (1004, 950.0, TO_DATE('2023-12-31', 'YYYY-MM-DD'), 102,
'Marine Drive');
INSERT INTO TenantRentsRoom (SIN, RentalRate, ContractEndDate, RoomNumber,
BuildingName) VALUES (1005, 1300.0, TO_DATE('2024-01-02', 'YYYY-MM-DD'), 101,
'Thunderbird');

INSERT INTO Mail (MailID, TenantSIN, RoomNumber, BuildingName, MailType) VALUES
(NULL, 1001, 101, 'Ponderosa', 'Parcel');
INSERT INTO Mail (MailID, TenantSIN, RoomNumber, BuildingName, MailType) VALUES
(NULL, 1002, 102, 'Ponderosa', 'Letter');
INSERT INTO Mail (MailID, TenantSIN, RoomNumber, BuildingName, MailType) VALUES
(NULL, 1003, 101, 'Marine Drive', 'Letter');
INSERT INTO Mail (MailID, TenantSIN, RoomNumber, BuildingName, MailType) VALUES
(NULL, 1004, 102, 'Marine Drive', 'Parcel');
INSERT INTO Mail (MailID, TenantSIN, RoomNumber, BuildingName, MailType) VALUES
(NULL, 1005, 101, 'Thunderbird', 'Letter');

INSERT INTO BuildingReceivesMail (MailID, DateReceived) VALUES (10000,
TO_DATE('2023-11-22', 'YYYY-MM-DD'));
INSERT INTO BuildingReceivesMail (MailID, DateReceived) VALUES (10001,
TO_DATE('2023-11-23', 'YYYY-MM-DD'));
INSERT INTO BuildingReceivesMail (MailID, DateReceived) VALUES (10002,
TO_DATE('2023-11-24', 'YYYY-MM-DD'));
INSERT INTO BuildingReceivesMail (MailID, DateReceived) VALUES (10003,
TO_DATE('2023-11-25', 'YYYY-MM-DD'));
INSERT INTO BuildingReceivesMail (MailID, DateReceived) VALUES (10004,
TO_DATE('2023-11-26', 'YYYY-MM-DD'));
```

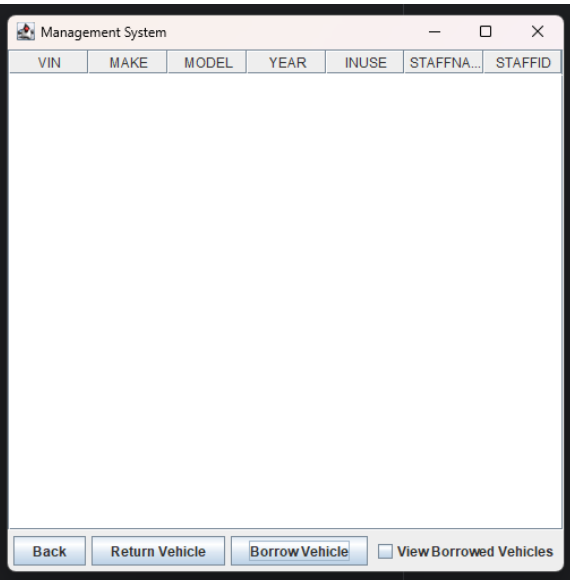
```
INSERT INTO Request (RequestID, BuildingName, RoomNumber, RequestType,
Complete, SubmissionDate, StaffID, SIN) VALUES (NULL, 'Ponderosa', 101,
'Maintenance', 0, TO_DATE('2023-11-20', 'YYYY-MM-DD'), 2001, 1001);
INSERT INTO Request (RequestID, BuildingName, RoomNumber, RequestType,
Complete, SubmissionDate, StaffID, SIN) VALUES (NULL, 'Walter Gage', 102,
'Cleaning', 1, TO_DATE('2023-11-21', 'YYYY-MM-DD'), 2002, 1002);
INSERT INTO Request (RequestID, BuildingName, RoomNumber, RequestType,
Complete, SubmissionDate, StaffID, SIN) VALUES (NULL, 'Totem Park', 101,
'Maintenance', 0, TO_DATE('2023-11-22', 'YYYY-MM-DD'), 2003, 1003);
INSERT INTO Request (RequestID, BuildingName, RoomNumber, RequestType,
Complete, SubmissionDate, StaffID, SIN) VALUES (NULL, 'Marine Drive', 102,
'Maintenance', 1, TO_DATE('2023-11-23', 'YYYY-MM-DD'), 2004, 1004);
INSERT INTO Request (RequestID, BuildingName, RoomNumber, RequestType,
Complete, SubmissionDate, StaffID, SIN) VALUES (NULL, 'Thunderbird', 101,
'Cleaning', 0, TO_DATE('2023-11-24', 'YYYY-MM-DD'), 2005, 1005);
```

Queries - Update found in src/ui/VehiclePage:

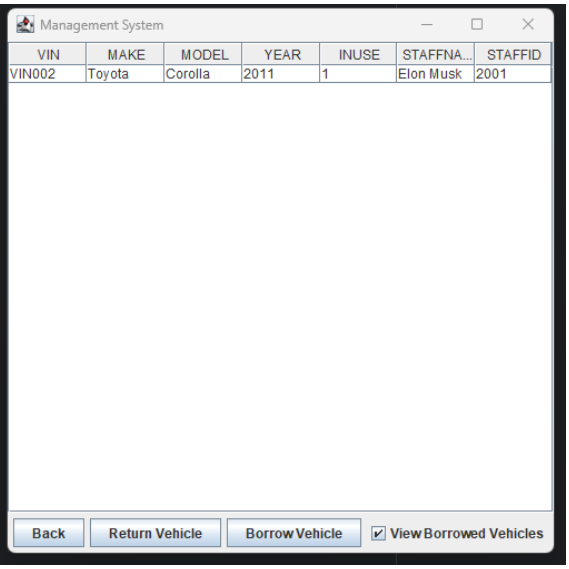
```
}

Pedro de Sant'Anna Novais
@Override
public void actionPerformed(ActionEvent e) {
    super.actionPerformed(e);
    if (currentVIN == null) return;
    if (e.getSource() == returnButton) {
        dataModifier.staffReturnVehicle(userIdentifier, currentVIN);
        refreshTable();
    } else if (e.getSource() == borrowButton) {
        dataModifier.staffBorrowVehicle(userIdentifier, currentVIN);
        refreshTable();
    }
}
```



```
    }
}

Pedro de Sant'Anna Novais
@Override
public void actionPerformed(ActionEvent e) {
    super.actionPerformed(e);
    if (currentVIN == null) return;
    if (e.getSource() == returnButton) {
        dataModifier.staffReturnVehicle(userIdentifier, currentVIN);
        refreshTable();
    } else if (e.getSource() == borrowButton) {
        dataModifier.staffBorrowVehicle(userIdentifier, currentVIN);
        refreshTable();
    }
}
```



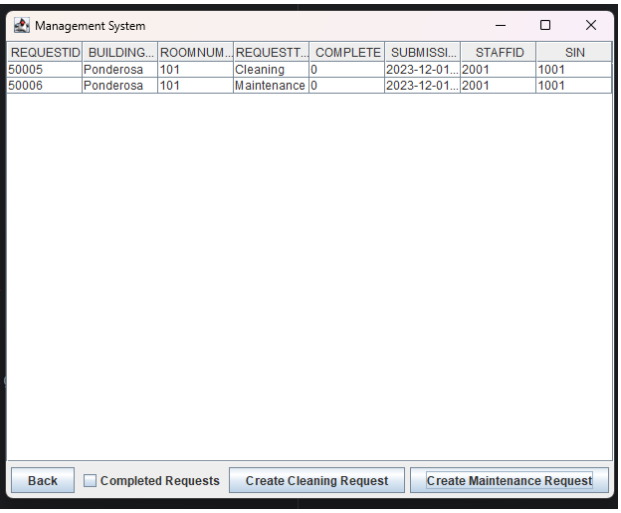

```
    }

    Pedro de Sant'Anna Novais
    @Override
    public void actionPerformed(ActionEvent e) {
        super.actionPerformed(e);
        if (currentVIN == null) return;
        if (e.getSource() == returnButton) {
            dataModifier.staffReturnVehicle(userIdentifier, currentVIN);
            refreshTable();
        } else if (e.getSource() == borrowButton) {
            dataModifier.staffBorrowVehicle(userIdentifier, currentVIN);
            refreshTable();
        }
    }
}
```



University of British Columbia, Vancouver
Department of Computer Science

Queries - Insert found in src/ui/RequestPage:

```
@Override
public void actionPerformed(ActionEvent e) {
    super.actionPerformed(e);
    if (e.getSource() == createCleaningRequestButton) {
        dataModifier.tenantRequests(userIdentifier, requestType: "Cleaning");
        refreshTable();
    } else if (e.getSource() == createMaintenanceRequestButton) {
        dataModifier.tenantRequests(userIdentifier, requestType: "Maintenance");
        refreshTable();
    } else if (e.getSource() == comboBox) {
        JComboBox cb = (JComboBox) e.getSource();
        String choice = (String) cb.getSelectedItem();

        applyDateFilter = java.util.Arrays.asList(dateFilters).indexOf(choice);
        updateTableQuery();
        refreshTable();
    } else if (e.getSource() == completeButton) {
        dataModifier.staffCompleteRequest(((BigDecimal) table.getValueAt(table.
        refreshTable();
    }
}
```




```
@Override
public void actionPerformed(ActionEvent e) {
    super.actionPerformed(e);
    if (e.getSource() == createCleaningRequestButton) {
        dataModifier.tenantRequests(userIdentifier, requestType: "Cleaning");
        refreshTable();
    } else if (e.getSource() == createMaintenanceRequestButton) {
        dataModifier.tenantRequests(userIdentifier, requestType: "Maintenance");
        refreshTable();
    } else if (e.getSource() == comboBox) {
        JComboBox cb = (JComboBox) e.getSource();
        String choice = (String) cb.getSelectedItem();

        applyDateFilter = java.util.Arrays.asList(dateFilters).indexOf(choice);
        updateTableQuery();
        refreshTable();
    } else if (e.getSource() == completeButton) {
        dataModifier.staffCompleteRequest(((BigDecimal) table.getValueAt(table.
        refreshTable();
    }
}
```