University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: July 15, 2024

Group Number: 42

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Dominic Leo	66545906	t6b8l	dpaleo22@gmail.com
Robert Burns	27053263	j4w1e	rdburns96@gmail.com
Shane Mander	38316287	s1u2u	shane.mander31@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.

University of British Columbia, Vancouver
Department of Computer Science

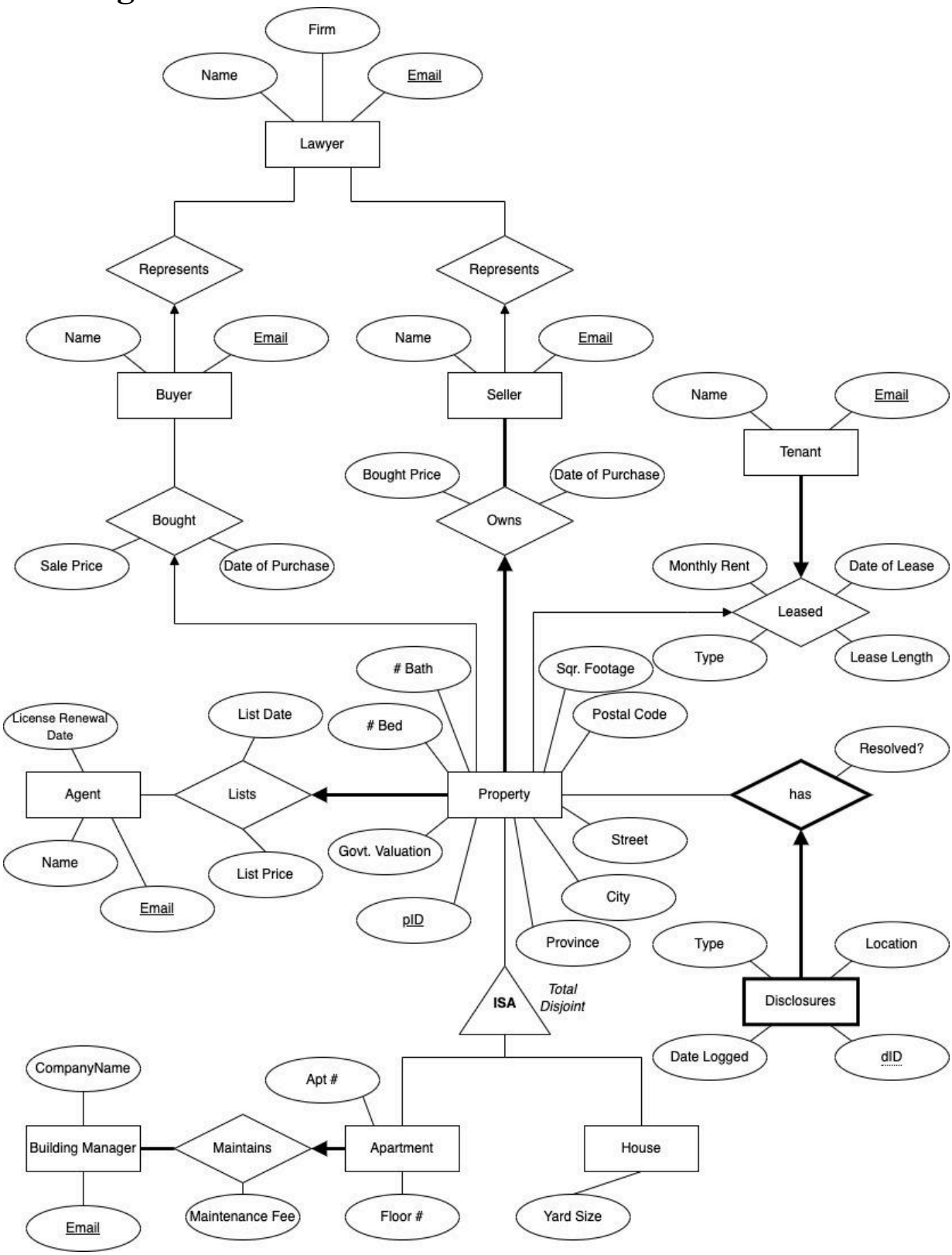
GROUP 42 - Real Estate Management System

Project Description

A real estate management system that would allow a real estate company to store and query its listed properties, and the parties related to the listed property. This database will allow employees of our target real estate company to store and query all property listings the company has been involved with (past and present). It will enable our agent to view the property details, which agent listed the property, the contact details of the seller or the seller's lawyer, the building manager's contact details (if the property is an apartment), and any disclosures that were made regarding the property (property damage, etc.).

University of British Columbia, Vancouver
Department of Computer Science

ER Diagram



University of British Columbia, Vancouver
Department of Computer Science

Conceptual Schema

Lawyer(name:VARCHAR, firm:VARCHAR, email:VARCHAR)
PK(email)

Buyer(name:VARCHAR, email:VARCHAR, **L_email**:VARCHAR)
PK(email)
FK L_email REFERENCES Lawyer

Seller(name:VARCHAR, email:VARCHAR, **L_email**:VARCHAR)
PK(email)
FK L_email REFERENCES Lawyer

Tenant(name:VARCHAR, email:VARCHAR, **pID**:INT UNIQUE, monthlyRent:FLOAT,
leaseType:VARCHAR, leaseSignDate:DATE, leaseLength:INT)
PK(email)
FK pID REFERENCES Property

Disclosures(dID:INT, type:INT, location:VARCHAR, dateLogged:DATE, resolved:BOOLEAN,
pID: INT UNIQUE)
PK(dID, pID)
FK pID REFERENCES Property

Agent(email:VARCHAR, name:VARCHAR, licenseRenewalDate:DATE)
PK(email)

BuildingManager(email:VARCHAR, name:VARCHAR)
PK(email)

Apartment(**pID**:INT, floorNum:INT NOT NULL, aptNum:INT NOT NULL,
maintenanceFee:DECIMAL(19,2), **BM_email**:VARCHAR NOT NULL)
PK(pID)
FK pID REFERENCES Property
FK BM_email REFERENCES BuildingManager

House(**pID**:INT, yardSize:FLOAT)
PK(pID)
FK pID REFERENCES Property

Property(pID:INT, numBed:INT NOT NULL, numBath:FLOAT NOT NULL,
govtValuation:DECIMAL(19,2), sqft:FLOAT, postalCode:VARCHAR,
street#:VARCHAR, city:VARCHAR, province:VARCHAR,
boughtPrice:DECIMAL(19,2), dateOfPurchase:DATE, salePrice:DECIMAL(19,2),
dateOfSale:DATE, listDate:DATE, listPrice:DECIMAL(19,2), B_email, S_email NOT
NULL, A_email NOT NULL)
PK(pID)
FK B_email REFERENCES Buyer
FK S_email REFERENCES Seller

FK A_email REFERENCES Agent

Functional Dependencies

Property:

pID -> numBed, numBath, govtValuation, sqft, postalCode, streetNum, city, province, listDate, listPrice, salePrice, dateOfPurchase, boughtPrice, dateOfSale, B_email, S_email, A_email

Postal Code -> city, province,

Disclosures:

dID -> location, dateLogged, type, resolved

Apartment:

aptNum -> floorNum
pID -> floorNum, aptNum, maintenance fee

House:

pID -> yardSize

Building Manager:

email -> companyName

Agent:

email -> name, licenseRenewalDate

Tenant:

email -> name, monthlyRent, dateOfLease, leaseLength, type

Buyer:

email -> name

Seller:

email -> name

Lawyer:

email -> name, firm

University of British Columbia, Vancouver
Department of Computer Science

Normalization

Buyer, Seller, Building Manager, and House are already in BCNF since they each have only two attributes. Additionally, Lawyer, Tenant, Agent, and Disclosures are also in BCNF already because the left hand sides of their only functional dependencies consist only of their keys. Property and Apartment each have two functional dependencies, notably one non-key functional dependency, and will be investigated further.

The Property relation has two functional dependencies:

1) pID -> numBed, numBath, govtValuation, sqft, postal code, streetNum, city, province, listDate, listPrice, salePrice, dateOfPurchase, boughtPrice, dateOfSale, B_email, S_email, A_email

2) postalCode -> city, province

Functional dependency 2) violates BCNF, hence the Property relation is not in BCNF and so it will be normalized through BCNF decomposition.
After decomposing once we get the following two relations:

- Property(<u>pID</u>:INT, numBed:INT, numBath:INT, govtValuation:DECIMAL(19,2), sqft:FLOAT, **postalCode**:VARCHAR, streetNum:VARCHAR, boughtPrice:DECIMAL(19,2), dateOfPurchase:DATE, salePrice:DECIMAL(19,2), dateOfSale:DATE, listDate:DATE, listPrice:DECIMAL(19,2), **B_email**:VARCHAR, **S_email**:VARCHAR, **A_email**:VARCHAR),
  PK(pID), FK(B_email) REFERENCES Buyer(email), FK(S_email) REFERENCES Seller(email), FK(postalCode) REFERENCES PropertyAddress(postalCode),
  FK(A_email) REFERENCES Agent(email)

- PropertyAddress(<u>postalCode</u>:VARCHAR, city:VARCHAR, province:VARCHAR)
  PK(postalCode)

In the new Property relation, pID is a key for the relation so it is in BCNF. Similarly, in Property2, postalCode is a key for the relation so it is also in BCNF. Hence, we have completed a lossless-join decomposition.

The Apartment relation has two functional dependencies:

1) aptNum -> floorNum

2) pID -> floorNum, aptNum, maintenance fee

Functional dependency 1) violates BCNF, hence the Apartment relation is not in BCNF and so it will be normalized through BCNF decomposition.
After decomposing once we get the following two relations:

- Apartment1(**pID**:INT, **aptNum**:INT, maintenanceFee:DECIMAL(19,2), **BM_email**:VARCHAR)
  PK(pID), FK(BM_email) REFERENCES BuildingManager(email), FK(pID) REFERENCES Property(pID), FK(aptNum) REFERENCES Apartment2(aptNum)

- Apartment2(aptNum:INT, floorNum:INT)
  PK(aptNum)

In Apartment1, pID is a key for the relation so it is in BCNF. Similarly, in Apartment2, postalCode is a key for the relation so it is also in BCNF. Hence, we have completed a lossless-join decomposition of the Apartment relation.

SQL/DDL Statements

```
CREATE TABLE Lawyer(
        name            VARCHAR,
        firm            VARCHAR,
        email           VARCHAR PRIMARY KEY);

CREATE TABLE Buyer(
        email           VARCHAR PRIMARY KEY,
        name            VARCHAR,
        L_email         VARCHAR,
        FOREIGN KEY(L_email)
                REFERENCES Lawyer(email)
                 ON DELETE CASCADE);

CREATE TABLE Seller(
        email           VARCHAR PRIMARY KEY,
        name            VARCHAR,
        L_email         VARCHAR,
        FOREIGN KEY(L_email)
                REFERENCES Lawyer(email)
                 ON DELETE CASCADE);

CREATE TABLE Tenant(
        email           VARCHAR PRIMARY KEY,
        name            VARCHAR,
        pID             INT UNIQUE,
        monthlyRent     DECIMAL (19,2)
        leaseType       VARCHAR,
        leaseSignDate   DATE,
        leaseLength     INT,
        FOREIGN KEY(pID)
                REFERENCES Property(pID));

CREATE TABLE Disclosures(
```

```
        dID             INT,
        type            INT,
        location        VARCHAR,
        dateLogged      DATE,
        resolved        BOOLEAN,
        pID             INT,
        PRIMARY KEY(dID, pID)
        FOREIGN KEY (pID) REFERENCES Property);

CREATE TABLE Agent(
        email           VARCHAR PRIMARY KEY,
        name            VARCHAR,
        licenseRenewalDate  DATE);

CREATE TABLE BuildingManager(
        email           VARCHAR PRIMARY KEY,
        companyName     VARCHAR);

CREATE TABLE Apartment1(
        pID             INT PRIMARY KEY,
        floorNum        INT NOT NULL,
        aptNum          INT NOT NULL,
        maintenanceFee  DECIMAL(19,2),
        BM_email        VARCHAR NOT NULL,
        FOREIGN KEY (pID)       REFERENCES Property
        FOREIGN KEY (BM_email)
                REFERENCES BuildingManager(email)
        FOREIGN KEY (aptNum)
                REFERENCES ApartmentAddress);

CREATE TABLE Apartment2(
        aptNum          INT PRIMARY KEY,
        floorNum        INT NOT NULL);

CREATE TABLE House(
        pID             INT PRIMARY KEY,
        yardSize        FLOAT,
        FOREIGN KEY (pID) REFERENCES Property);

CREATE TABLE Property(
        pID             INT PRIMARY KEY,
        numBed          INT NOT NULL,
        numBath         FLOAT NOT NULL,
        govtValuation   DECIMAL(19,2),
        sqft            FLOAT NOT NULL,
        postalCode      CHAR(6) NOT NULL,
        streetNum       VARCHAR NOT NULL,
        boughtPrice     DECIMAL(19,2),
```

```
    dateofPurchase      DATE,
    salePrice           DECIMAL(19,2),
    dateOfSale          DATE,
    listDate            DATE NOT NULL,
    listPrice           DECIMAL(19,2) NOT NULL,
    B_email             VARCHAR,
    S_email             VARCHAR NOT NULL,
    A_email             VARCHAR NOT NULL,
    FOREIGN KEY (postalCode) REFERENCES PropertyAddress
        ON DELETE NO ACTION)
    FOREIGN KEY (B_email) REFERENCES Buyer
        ON DELETE SET NULL
    FOREIGN KEY (S_email) REFERENCES Seller
        ON DELETE SET DEFAULT
    FOREIGN KEY (A_email) REFERENCES Agent
        ON DELETE SET NULL);

CREATE TABLE PropertyAddress(
    postalCode          CHAR(6) PRIMARY KEY,
    city                VARCHAR NOT NULL,
    province            VARCHAR NOT NULL);
```

Populate Tables

```
INSERT INTO Lawyer(name, firm, email)
    VALUES('John Smith', 'Baker McKenzie', 'johnsmith2@bakermckenzie.com')
    VALUES('John Smith', 'Baker McKenzie', 'johnsmith1@bakermckenzie.com')
    VALUES('Sara Smith', 'Baker McKenzie', 'sarasmith@bakermckenzie.com')
    VALUES('Donald Trump', 'Temple Chambers', 'donaldtrump@templechambers.com')
    VALUES('Donald Duck', 'Mickey Mouse & Son', 'donaldduck@mickeymouse.com')

INSERT INTO Buyer(email, name, L_email)
    VALUES('johnsmith@gmail.com', 'John Smith', 'johnsmith2@bakermckenzie.com')
    VALUES('joeforte@gmail.com', 'Joe Forte', 'johnsmith2@bakermckenzie.com')
    VALUES('johnsmith42@gmail.com', 'John Smith', 'sarasmith@bakermckenzie.com')
    VALUES('donalduck@gmail.com', 'Donald Duck', donaldduck@mickeymouse.com)
    VALUES('alicebaker@gmail.com', 'Alice Baker', 'johnsmith1@bakermckenzie.com')

INSERT INTO Seller(email, name, L_email)
    VALUES('alicesmith@gmail.com', 'Alice Smith, 'johnsmith1@bakermckenzie.com')
    VALUES('bobshaw@gmail.com', 'Bob Shaw', 'johnsmith2@bakermckenzie.com')
    VALUES('xialong@baidu.com', 'Cherry Li', 'sarasmith@bakermckenzie.com')
    VALUES('donalduck@gmail.com', 'Donald Duck', donaldduck@mickeymouse.com)
    VALUES('emmalau@outlook.com', 'Emma Lau', NULL)

INSERT INTO Tenant(email, name, pID, monthlyRent, leaseType, leaseSignDate, leaseLength)
    VALUES('adamjames@hotmail.com', 'Adam James', 100, 5040.12, 'Monthly',
    '03-06-2019', 14)
```

VALUES('bethanylau@gmail.com', 'Bethany Lau', 102, 6000.00, 'Annually', '10-06-2024', 24)
VALUES('cassiebones@yahoo.com', 'Cassandra Bones', 104, 8000, 'Monthly', '25-12-2023', 6)
VALUES('lepetitdominique@hotmail.com', 'Dominique Rousseau', 106, 5040.12, 'Annually', '28-06-2020', 36)
VALUES('thegreatbanana@gmail.com', 'Esther James', 108, 1234.56, 'Monthly', '03-02-2019', 2)

INSERT INTO Disclosures(dID, type, location, dateLogged, resolved, pID)
VALUES(1, 1001, 'bathroom #1', '03-05-2019', TRUE, 100)
VALUES(2, 1002, 'kitchen', '03-05-2019', FALSE, 100)
VALUES(3, 1001, 'bathroom #2', '03-05-2019', , TRUE, 100)
VALUES(1, 1005, 'master bedroom', '06-06-2024', FALSE, 102)
VALUES(2, 6000, 'yard', 104, '06-06-2024', FALSE, 102)

INSERT INTO Agent(email, name, licenseRenewalDate)
VALUES('augustusjulius@manderburnsleo.com', 'Augustus Julius', 01-01-2022)
VALUES('tiberiusclaudius@manderburnsleo.com', 'Tiberius Claudius', 01-01-2023)
VALUES('domitianusflavius@manderburnsleo.com', 'Domitianus Flavius', 01-01-2024)
VALUES('genghiskhan@manderburnsleo.com', 'Genghis Khan', 01-01-2024)
VALUES('commodusaurelius@manderburnsleo.com', 'Commodus Aurelius', 01-01-2020)

INSERT INTO BuildingManager(email, companyName)
VALUES('johnsmith@bluemanagement.com', 'Blue Management')
VALUES('michaelbay@bluemanagement.com', 'Blue Management')
VALUES('cassius_clay@hamptonplace.com', 'Hampton Place')
VALUES('drexler_clyde@trailblazer.mgmt.com', 'Trailblazer Management')
VALUES('clark.caitlin@atira.com', 'Atira Property Management')

INSERT INTO Apartment1(pID, aptNum, maintenanceFee, BM_email)
VALUES(100, 302, 49.99, 'johnsmith@bluemanagement.com')
VALUES(102, 006, 35.50, 'michaelbay@bluemanagement.com')
VALUES(104, 808, 72.00, 'cassius_clay@hamptonplace.com')
VALUES(110, 100, 27.25, 'drexler_clyde@trailblazer.mgmt.com')
VALUES(112, 501, 49.99, 'clark.caitlin@atira.com')

INSERT INTO Apartment2(aptNum, floorNum)
VALUES(302, 3)
VALUES(006, 0)
VALUES(808, 8)
VALUES(100, 1)
VALUES(501, 5)

INSERT INTO House(pID, yardSize)
VALUES(106, 80)
VALUES(108, 112)

VALUES(114, 55)
VALUES(116, 200)
VALUES(118, 344)

INSERT INTO Property(pID, numBed, numBath, govtValuation, sqft, postalCode, streetNum, boughtPrice, dateOfPurchase, salePrice, dateOfSale, listDate, listPrice, B_email, S_email, A_email)

VALUES(100, 2, 2, 220 000, 800, 'V6T 1Z4', '105 W 10th', 150 000, '02-09-2017', NULL, NULL, '03-05-2019', 270 000, NULL, 'alicesmith@gmail.com', 'genghiskhan@manderburnsleo.com')

VALUES(102, 1, 1.5, 120 000, 650, 'V5J 0A3', '#11 Dunbar', 120 000, '03-05-2024', NULL, NULL, '06-06-2024', 200 000, NULL, 'donalduck@gmail.com', 'genghiskhan@manderburnsleo.com')

VALUES(104, 2, 2.5, 400 000, 1024, 'V5K 0BT', '#3 W 49', 325 000, '09-01-2021', NULL, NULL, '08-12-2023', 270 000, NULL, 'donalduck@gmail.com', 'genghiskhan@manderburnsleo.com')

VALUES(106, 2, 1, 620 000, 900, 'V6T 2H2', '#28 EB 13th', 450 000, '03-05-2018', NULL, NULL, '23-02-2021', 720 000, NULL, 'emmalau@outlook.com', 'genghiskhan@manderburnsleo.com')

VALUES(108, 3, 2, 780 000, 1300, 'S0K 0Y0', '#2 Sunny Rd', 660 000, '05-03-2017', NULL, NULL, '02-11-2018', 850 000, NULL, bobshaw@gmail.com, 'genghiskhan@manderburnsleo.com')

VALUES(110, 1, 2, 230 000, 1200, 'V5K 004', '#123 Main St', 150 000, '03-05-2020', 300 000, '21-07-24', '02-01-2024', 330 000, 'johnsmith@gmail.com', xialong@baidu.com, 'genghiskhan@manderburnsleo.com')

VALUES(112, 3, 3, 1 000 000, 1500, 'V3S 0H2', '34 52nd Ave', 400 000, '05-03-2006', NULL, 'NULL', '11-07-2023', 1 200 000, NULL, 'donalduck@gmail.com', 'genghiskhan@manderburnsleo.com')

VALUES(114, 4, 3.5, 900 000, 1350, 'V3Z 04T', '2842 32nd Ave', 825 000, '09-10-2022', NULL, NULL, '01-04-2024', 925 000, NULL, emmalau@outlook.com, commodusaurelius@manderburnsleo.com)

VALUES(116, 4, 4, 1 800 000, 2120, 'H1A 0A1', '#123 W 4th', 1 000 000, '10-09-2013', NULL, NULL, '31-07-2022', 2 000 000, NULL, donalduck@gmail.com, 'augustusjulius@manderburnsleo.com')

VALUES(118, 4, 4.5, 2 800 000, 3140, 'M4C 1B5', '#54 193 Ave', 1 100 000, '06-02-2010', 3 000 000, '06-02-2024', '05-16-2024', 3 200 000, joeforte@gmail.com, bobshaw@gmail.com, 'augustusjulius@manderburnsleo.com')

INSERT INTO PropertyAddress(postalCode, city, province)

VALUES('V6T 1Z4', Vancouver, BC)
VALUES('V5J 0A3', Vancouver, BC)
VALUES('V5K 0BT', Vancouver, BC)
VALUES('V5K 004', Vancouver, BC)
VALUES('S0K 0Y0', Saskatoon, SK)
VALUES('V6T 2H2', Vancouver, BC)
VALUES('V3S 0H2', Surrey, BC)
VALUES('V3Z 04T', Surrey, BC)

VALUES(‘H1A 0A1’ Montreal, QC)
VALUES(‘M4C 1B5’, Toronto, ON)

Comments

- Changed attribute names when translating ER diagram to conceptual schema to accommodate for SQL.
- Changed the ER Diagram to include a total participation constraint for Apartment to the Maintains relationship as every apartment (should) have a Building Manager who maintains the apartment building.
- In our conceptual schema for Property, dateOfPurchase is the attribute representing the date in which our seller bought the property, and dateOfSale is the date when the buyer bought the property. The same convention holds for salePrice, and boughtPrice
- In Disclosures, type refers to an in-house code system used to encode the type of disclosure being referred to. E.g. Code #1001 indicates water damage, and Code #1003 indicates a scuffed wall.