University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: 3-June-2023

Group Number: 17

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Khammy Saychaleun	40259574	v1m0l	khammyschl@gmail.com
Keshav Gopinath	61086260	z8z2b	keshavkarthikk@gmail.com
Aayush Kogar	62235312	s7j9s	18aayushk@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

Project Summary

Finding an affordable apartment is incredibly difficult in Vancouver, especially around the UBC area. The time-consuming process consists of exploring different areas, making hundreds of applications, and avoiding scams.

STUHOUSING is an application that helps students find housing in school areas by acting as a compiler of all housing available on different platforms and supporting search management. A school support contact (employee) will act as the middle-man for users and different platforms. Our current tech stack consists of:
Backend and Frontend: PHP
OS: Linux (Undergraduate Servers)
Database: Oracle

University of British Columbia, Vancouver
Department of Computer Science

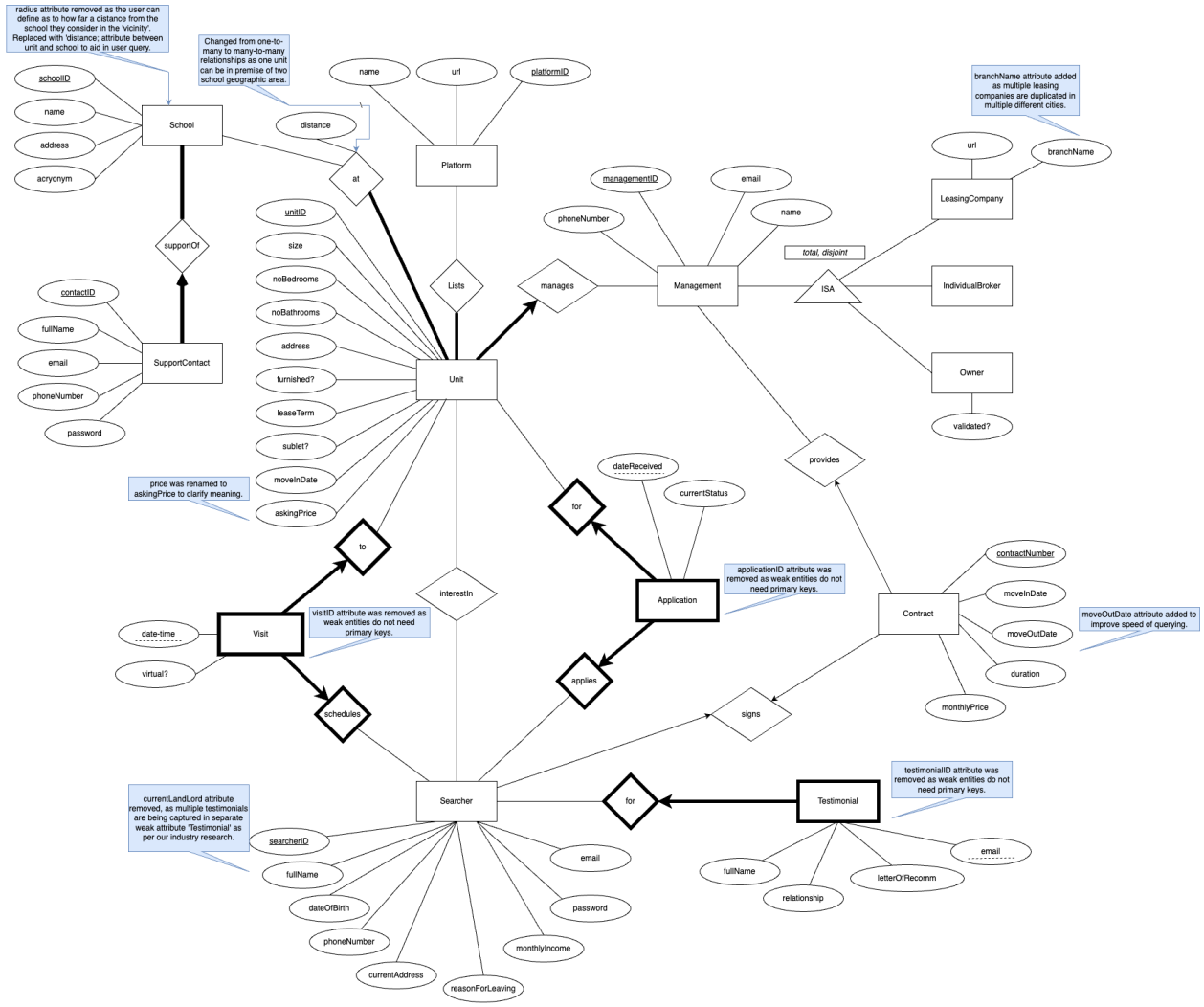
Entity Relationship Diagram Changes

Change Summary	
Change	Reasoning
Replaced ‘radius’ attribute from school to a ‘distance’ attribute at relationship between unit and school.	In speaking to users, we realized that every user has a different requirement of distance to a school. Changing to a distance field will allow users to query distance to their needs.
Modified the relationship between UNIT and SCHOOL from `many-to-one` to `many-to-many`.	To work around the edge case that one unit is in close proximity to multiple universities. This is common in downtown areas of large cities.
Delete ‘currentLandLord’ attribute from SEARCHER.	Do not need this information as it is captured in the weak entity connected called ‘Testimonial’.
Removed ‘applicationID’, ‘testimonialID’, and ‘visitID’ attributes.	As weak attributes do not need a primary key..
branchName attribute added to Property Management Leasing Company entity.	Leasing companies are found to have different branches, which should be distinguished in the database.
price was renamed to askingPrice	To clarify the meaning of this attribute.
moveOutDate attribute added to Contract entity.	To improve speed of querying by reducing calculations.

University of British Columbia, Vancouver
Department of Computer Science

New ERD

The ERD diagram has been attached below and a link to the original document can be found here . You will have to go to the **`M2 ERD`** or `**M2 ERD - No Comments**`tab on the draw.io page to find the ERD diagram.



University of British Columbia, Vancouver
Department of Computer Science

Relational Schema

Note:

- The lengths of the Email and Url are based on this and this
- Each unit has a max size set in the thousands, but we also allow for 1 decimal place, therefore 5 digits in total. (Eg. 1,000 sq and 1,000.5 sq allowed)
- We use DATETIME type for DateTime in VISIT, because we need to book searchers for a visit without conflict, unlike the moveInDate where the time is not restrictive
- The max allowable length for password is recommended as follows
- Change datatype of PhoneNumber from INT to VARCHAR(20) according to here

School(
SchoolID: INT,
Name: VARCHAR(50),
Address: VARCHAR(50),
Acronym: VARCHAR(5),
)

Candidate Keys: { SchoolID, (Name, Address) }
Primary Key: SchoolID

SupportContact(
ContactID: INT
FullName: VARCHAR(50),
Email: VARCHAR(320),
PhoneNumber: VARCHAR(20),
Password: VARCHAR(128),
SchoolID: INT
)
** SchoolID cannot be null

Candidate Keys: { ContactID }
Primary Key: ContactID
Foreign Key: SchoolID

Platform(

PlatformID: INT,
Name: VARCHAR(50),
Url: VARCHAR(2048)
)

Candidate Keys: { PlatformID, URL }
Primary Key: PlatformID

Unit(
UnitID: INT,
Size: FLOAT(5),
NoBedrooms: INT,
NoBathrooms: INT,
Address: VARCHAR(50),
Furnished: BOOLEAN,
LeaseTerm: INT,
Sublet: BOOLEAN,
MoveInDate: DATE,
AskingPrice: INT,
ManagementID: INT
)

**ManagementID cannot be null

Candidate Keys: { UnitID }
Primary Key: UnitID
Foreign Key: ManagementID

*Address (thought of making unique but possibility of having different people renting out rooms in the same unit)

Management(
ManagementID: INT,
Name: VARCHAR(50),
Email: VARCHAR(320),
PhoneNumber: VARCHAR(20),
)
Candidate Keys: { ManagementID }

University of British Columbia, Vancouver
Department of Computer Science

Primary Key: ManagementID
LeasingCompany(**ManagementID: INT,** Url: VARCHAR(2048), BranchName: VARCHAR(50)) Candidate Keys: { ManagementID } Primary Key: ManagementID Foreign Key: ManagementID
IndividualBroker(**ManagementID: INT**) CandidateKeys: {ManagementID} PrimaryKey: ManagementID Foreign Key: ManagementID *IndividualBroker and Owner will have unique email and phone number in oppose to a Leasing Company where employees might use the same email and phone number
Owner(**ManagementID: INT,** Validated?: BOOLEAN) CandidateKeys: {ManagementID} PrimaryKey: ManagementID Foreign Key: ManagementID
Visit(DateTime: DATETIME, Virtual: BOOLEAN, **UnitID: INT,** **SearcherID: INT**)

University of British Columbia, Vancouver
Department of Computer Science

**UnitID and SearcherID cannot be null CandidateKeys: {(DateTime,UnitID), (DateTime, SearcherID) } Primary Key: (DateTime,UnitID) Foreign Key: UnitID, SearcherID
Application(DateReceived: DATETIME, CurrentStatus: VARCHAR(20), **UnitID: INT,** **SearcherID: INT**) **UnitID and SearcherID cannot be null CandidateKey: { (DateReceived, UnitID, SearcherID) } Primary Key: (DateReceived, UnitID, SearcherID) Foreign Key: UnitID, SearcherID
Searcher(SearcherID: INT, FullName: VARCHAR(50), Email: VARCHAR(320), PhoneNumber: VARCHAR(20), DateOfBirth: DATE, CurrentAddress: VARCHAR(50), ReasonForLeave: VARCHAR(100), MonthlyIncome: INT, Password: VARCHAR(128),) Candidate Key: {SearcherID, Email, PhoneNumber} Primary Key: SearcherID
Testimonial(LetterOfRecomm: VARCHAR(500), FullName: VARCHAR(50),

Email: VARCHAR(320),
Relationship: VARCHAR(100),
SearcherID: INT
)
**SearcherID cannot be null

Candidate Key: {(Email, SearcherID)}
Primary Key: (Email, SearcherID)
Foreign Key: SearcherID

Contract(
ContractNumber: INT,
MoveInDate: DATE,
MoveOutDate: DATE,
Duration: INT,
MonthlyPrice: INT,
SearcherID: INT,
ManagementID: INT
)

**SearcherID cannot be null
**ManagementID can be null

Candidate Key: {ContractNumber, (SearcherID, ManagementID, MoveInDate) }
Primary Key: ContractNumber
Foreign Key: SearcherID, ManagementID

Unit_Searcher(
SearcherID: INT,
UnitID: INT,
)

Candidate Key: {(SearcherID, UnitID)}
Primary Key: (SearcherID, UnitID)
Foreign Key: SearcherID, UnitID

Unit_Platform(
UnitID: INT ,

PlatformID: INT,
)

Candidate Key: {(UnitID, PlatformID)}
Primary Key: (UnitID, PlatformID)
Foreign Key: UnitID, PlatformID

Unit_School(
UnitID: INT,
SchoolID: INT,
Distance: FLOAT)

CandidateKey: {(UnitID, SchoolID)}
PrimaryKey: (UnitID, SchoolID)
ForeignKey: UnitID, SchoolID

FUNCTIONAL DEPENDENCY

School = (SchoolID, Name, Address, Acronym)

Candidate Keys: { SchoolID, (Name, Address) }
Primary Key: SchoolID

FDs observed in the schema:

- SchoolID → Name, Address, Acronym
- Name, Address → SchoolID, Acronym
- Name → Acronym

SupportContact = (ContactID, FullName, Email, PhoneNumber, Password, SchoolID)

Candidate Keys: { ContactID }
Primary Key: ContactID

FDs observed in the schema:

- ContactID → FullName, Email, PhoneNumber, Password, SchoolID

University of British Columbia, Vancouver
Department of Computer Science

- Email → Password

Platform = (PlatformID, Name, Url)

Candidate Keys: { PlatformID, URL }
Primary Key: PlatformID

FDs observed in the schema:

- PlatformID → Name, URL
- Url → PlatformID, Name

Unit = (UnitID, Size, NoBedrooms, NoBathrooms, Address, Furnished, LeaseTerm, Sublet, MoveInDate, AskingPrice, ManagementID)

Candidate Keys: { UnitID } // We may want to be more explicit here
Primary Key: UnitID
Foreign Key: ManagementID

FDs observed in the schema:

- UnitID → Size, NoBedrooms, NoBathrooms, Address, Furnished, LeaseTerm, Sublet, MoveInDate, Price, ManagementID

Management = (ManagementID, Name, Email, PhoneNumber)

Candidate Keys: { ManagementID } // We may want to be more explicit here
Primary Key: ManagementID

FDs observed in the schema:

ManagementID → Name, Email, PhoneNumber

LeasingCompany = (ManagementID, Url, BranchName)

Candidate Keys: { ManagementID }
Primary Key: ManagementID
ForeignKey : ManagementID

University of British Columbia, Vancouver
Department of Computer Science

FDs observed in the schema:

- ManagementID⟶ URL, BranchName

Broker = (ManagementID)

Candidate Keys: { ManagementID }
Primary Key: ManagementID
ForeignKey : ManagementID

FDs observed in the schema:

- No FDs

Owner = (ManagementID, Validated)

Candidate Keys: { ManagementID }
Primary Key: ManagementID
ForeignKey : ManagementID

FDs observed in the schema:

- ManagementID ⟶ Validated

Visit = (DateTime, Virtual, UnitID, SearcherID)

Candidate Keys: { (DateTime, UnitID), (DateTime, SearcherID) }
Primary Key: (DateTime, UnitID)
Foreign Key: SearcherID, UnitID

FDs observed in the schema:

- DateTime, SearcherID, UnitID ⟶ Virtual
- UnitID, DateTime ⟶ SearcherID
- SearcherID, DateTime ⟶ UnitID

Application = (DateReceived, CurrentStatus, UnitID, SearcherID)

CandidateKey: { (DateReceived, UnitID, SearcherID) }
PrimaryKey: DateReceived, UnitID, SearcherID

University of British Columbia, Vancouver
Department of Computer Science

FDs observed in the schema:

- DateRecieved, SearcherID, UnitID → CurrentStatus

Searcher = (SearcherID, FullName, Email, PhoneNumber, DateOfBirth, CurrentAddress, ReasonForLeave, MonthlyIncome, Password)

CandidateKey: {SearcherID, Email, PhoneNumber}
PrimaryKey: SearcherID

FDs observed in the schema:

- SearcherID → FullName, Email, PhoneNumber, DateOfBirth, CurrentAddress, ReasonForLeave, MonthlyIncome, Password,
- Email → SearcherID, FullName, PhoneNumber, DateOfBirth, CurrentAddress, ReasonForLeave, MonthlyIncome, Password
- PhoneNumber → SearcherID, FullName, Email, DateOfBirth, CurrentAddress, ReasonForLeave, MonthlyIncome, Password

Testimonial = (FullName, Email, Relationship, LetterOfRecomm, SearcherID)

CandidateKey: {(Email, SearcherID)}
PrimaryKey: (Email, SearcherID)
ForeignKey: SearcherID

FDs observed in the schema:

- Email, SearcherID → FullName, LetterOfRecomm, Relationship

Contract = (ContractNumber, MoveInDate,MoveOutDate, Duration, MonthlyPrice, SearcherID, ManagementID)

CandidateKey: {ContractNumber, (SearcherID, ManagementID, MoveInDate) }
PrimaryKey: ContractNumber
ForeignKey: SearcherID, ManagementID

University of British Columbia, Vancouver
Department of Computer Science

FDs observed in the schema:

- ContractNumber → MoveInDate, MoveOutDate, Duration, MonthlyPrice, SearcherID, ManagementID
- (SearcherID, ManagementID, MoveInDate) → ContractNumber, Duration, MoveOutDate, MonthlyPrice
- MoveInDate, MoveOutDate → Duration
- MoveInDate, Duration → MoveOutDate
- MoveOutDate, Duration → MoveInDate

Unit_Searcher = (SearcherID, UnitID)

CandidateKey: {(SearcherID, UnitID)}
PrimaryKey: (SearcherID, UnitID)
ForeignKey: SearcherID, UnitID

FDs observed in the schema:

- No FDs

Unit_Platform = (UnitID, PlatformID)
CandidateKey: {(UnitID, PlatformID)}
PrimaryKey: (UnitID, PlatformID)
ForeignKey: UnitID, PlatformID

FDs observed in the schema:

- No FDs

Unit_School = (UnitID, SchoolID,Distance)

CandidateKey: {(UnitID, SchoolID)}
PrimaryKey: (UnitID, SchoolID)
ForeignKey: UnitID, SchoolID

FDs observed in the schema:

- (UnitID, SchoolID) → Distance

University of British Columbia, Vancouver
Department of Computer Science

Normalization

School = (SchoolID, Name, Address, Acronym)

We describe the relation as R(Sid, N, A, Ac) with the following FDs

- Sid → N, A, Ac
- N, A→ Sid, Ac
- N → Ac

Candidate Keys: { Sid, (N, A) }
Primary Key: Sid

The following Relation violates both BCNF and 3NF.

In the FD: [N → Ac]
N is not a SuperKey
Ac is not part of the Candidate Key

Performing Decomposition, we get:
R(Sid, N, A, Ac) → R1(Sid, N, A) + R2(N, Ac) [Decomposing on N → Ac]

The Normalised Relations are: {R1(Sid, N, A), R2(N, Ac)}

SupportContact = (ContactID, FullName, Email, PhoneNumber, Password, SchoolID)

We describe the relation as R(Cid, Fn, E, Pn, P, Sid) with the following FDs:

- Cid → Fn, E, Pn, P, Sid
- E → P

Candidate Keys: { Cid }
Primary Key: Cid

The following Relation violates both BCNF and 3NF.

In the FD: [E → P]
E is not a SuperKey

P is not part of the Candidate Key

Performing Decomposition, we get:
R(Cid, Fn, E, Pn, P, Sid) ⟶ R1(E, P) + R2(E, Pn, Cid, Fn, Sid) [Decomposing on E ⟶ P]

The Normalised Relations are: {R1(E, P), R2(E, Pn, Cid, Fn, Sid)}

Platform = (PlatformID, Name, Url)

We describe the relation as R(Pid, N, U) with the following FDs:

- Pid⟶ N, U
- U ⟶ Pid, N

Candidate Keys: { Pid, U}
Primary Key: Pid

The Relation is in BCNF, no normalization is required

Unit = (UnitID, Size, NoBedrooms, NoBathrooms, Address, Furnished, LeaseTerm, Sublet, MoveInDate, AskingPrice, ManagementID)

We describe the relation as R(Uid, S, NBe, NBa, A, F, LTe, S, M, P, Mid) with the following FDs:

- Uid -> S, NBe, NBa, A, F, LTe, S, M, P, Mid

Candidate Keys: { Uid }
Primary Key: Uid
Foreign Key: Mid

The Relation is already in BCNF, no normalization is required

Management = (ManagementID, Name, Email, PhoneNumber)

We describe the relation as R(Mid, N, E, Pn) with the following FDs:

- Mid ⟶ N, E, Pn

Candidate Keys: { Mid}
Primary Key: Mid

The Relation is already in BCNF, no normalization required

LeasingCompany = (ManagementID, Url, BranchName)

We describe the relation as R(Mid, U, Bn) with the following FDs:

- Mid → U, Bn

Candidate Keys: { Mid}
Primary Key: Mid
ForeignKey: Mid

The Relation is already in BCNF, no normalization required

Broker = (ManagementID)

We describe the relation as R(Mid) with the following FDs:

- No FDs

Candidate Keys: { Mid}
Primary Key: Mid
ForeignKey: Mid

The Relation is already in BCNF, no normalization required

Owner = (ManagementID, Validated)

We describe the relation as R(Mid,Validated) with the following FDs:

- Mid -> Validated

Candidate Keys: { Mid}
Primary Key: Mid
ForeignKey: Mid
The Relation is already in BCNF, no normalization required

Visit = (DateTime, Virtual, UnitID, SearcherID)

University of British Columbia, Vancouver
Department of Computer Science

We describe the relation as R(DT, V, Uid, Sid) with the following FDs:

- DT, Sid, Uid →V
- DT, Uid → Sid
- DT, Sid → Uid

Candidate Keys: { (DT, Uid), (DT, Sid) }
Primary Key: (DT, Uid)
ForeignKeys: {Sid,Uid}

The relation is already in BCNF, no normalization is required

---

Application = (DateReceived, CurrentStatus, UnitID, SearcherID)

We describe the relation as R(DR, Cs, Uid, Sid) with the following FDs:

- DR, Sid, Uid → Cs

Candidate Keys: { (DR, Sid, Uid) }
Primary Key: (DR, Sid, Uid)
ForeignKeys: {Sid,Uid}

The relation is already in BCNF, no normalization is required

---

Searcher = (SearcherID, FullName, Email, PhoneNumber, DateOfBirth, CurrentAddress, ReasonForLeave, MonthlyIncome, Password)

We describe the relation as R(Sid, Fn, E, Pn, Dob, Ca, Rl, MI, P, Cn) with the following FDs:

- Sid → Fn, E, Pn, Dob, Ca, Rl, MI, P
- E → Sid, Fn, Pn, Dob, Ca, Rl, MI, P
- Pn → Sid, Fn, E, Dob, Ca, Rl, MI, P

CandidateKey: {Sid, E, Pn}
PrimaryKey: Sid

The relation is already in BCNF, no normalization is required

---

Testimonial = (FullName, Email, Relationship, LetterOfRecomm, SearcherID)

University of British Columbia, Vancouver
Department of Computer Science

We describe the relation as R(Fn, E, R, LoR, Sid) with the following FDs:

- E, Sid⟶ Fn, E, R,LoR

CandidateKey: {(E, Sid)}
PrimaryKey: (E, Sid)
ForeignKey: Sid

The relation is already in BCNF, no normalization is required

Contract = (ContractNumber, MoveInDate, MoveOutDate, Duration, MonthlyPrice, SearcherID, ManagementID)

We describe the relation as R(Cn, MID, MOD, D, Mp, Sid, MAid) with the following FDs:

- Cn ⟶ MID, MOD, D, Mp, Sid, MAid
- Sid, MAid, MID ⟶ Cn, D, MOD, Mp
- MID, MOD ⟶ D
- MID, D ⟶ MOD
- MOD, D ⟶ MID

CandidateKey: {Cn, (Sid, MAid, MID) }
PrimaryKey: Cn
ForeignKey: Sid, MAid

The Relation violates BCNF and 3NF

We see that in the FD: [MID, MOD ⟶ D]
MID, MOD is not a super key and D is not part of the Candidate Key

We also observe the FDs: [MID, D ⟶ MOD] and [MOD, D ⟶ MID] also violates BCNF for the same reason.

Normalization on the FDs that violate BCNF: **[decomposing MID, MOD ⟶ D]**

R(Cn, MID, MOD, D, Mp, Sid, MAid) ⟶ R1(Cn, MID, MOD, Mp, Sid, MAid)
+ R2(MID, MOD, D)

University of British Columbia, Vancouver
Department of Computer Science

The information provided by the FDs: [MID, D → MOD] and [MOD, D → MID] can be found in the relation R2(MID, MOD, D) so no further decomposition is needed

The Normalized relations are : {R1(Cn, MID, MOD, Mp, Sid, MAid), R2(MID,MOD,D)}

Unit_Searcher = (SearcherID, UnitID)

We describe the relation as R(Sid,Uid) with the following FDs:

- No FDs

CandidateKey: {(Sid, Uid)}
PrimaryKey: (Sid,Uid)
ForeignKey: (Sid,Uid)

Relation is already in BCNF, no normalization is required

Unit_Platform = (UnitID, PlatformID)

We describe the relation as R(Sid,Pid) with the following FDs:

- No FDs

CandidateKey: {(Sid, Pid)}
PrimaryKey: (Sid,Pid)
ForeignKey: (Sid,Pid)

Relation is already in BCNF, no normalization is required

Unit_School = (UnitID, SchoolID, Distance)

We describe the relation as R(Sid, Uid, D) with the following FDs:

- (Sid, Uid) → D

CandidateKey: {(Sid, Uid)}
PrimaryKey: (Sid, Uid)
ForeignKey: Sid, Uid

Relation is already in BCNF, no normalization is required

University of British Columbia, Vancouver
Department of Computer Science

SQL DDL

```
CREATE TABLE NameToAcronym(
        Name VARCHAR(50),
        Acronym VARCHAR(5),
        PRIMARY KEY(Name)
)

CREATE TABLE SchoolDetails(
        SchoolID INT,
        Name VARCHAR(50),
        Address VARCHAR(20),
        PRIMARY KEY(SchoolID),
        FOREIGN KEY(Name) REFERENCES NameToAcronym(Name)
)
```

```
CREATE TABLE SupportContact(
        ContactID: INT,
        FullName VARCHAR(50),
        Email VARCHAR(320),
        PhoneNumber VARCHAR(20),
        Password VARCHAR(128),
        SchoolID INT NOT NULL,
        PRIMARY KEY(ContactID),
        FOREIGN KEY(SchoolID) REFERENCES School(SchoolID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
)

CREATE TABLE SupportContactPasswords(
        Email VARCHAR(320),
        Password VARCHAR(128),
)
```

```
CREATE TABLE Platform(
        PlatformID INT,
```

```
        Name VARCHAR(50),
        Url VARCHAR(2048),
        PRIMARY KEY(PlatformID)
)
```

```
CREATE TABLE Unit(
        UnitID INT,
        Size FLOAT(5),
        NoBedrooms INT,
        NoBathrooms INT,
        Address VARCHAR(50),
        Furnished BOOLEAN,
        LeaseTerm INT,
        Sublet BOOLEAN,
        MoveInDate DATE,
        AskingPrice INT,
        ManagementID INT NOT NULL,
        PRIMARY KEY(UnitID),
        FOREIGN KEY (ManagementID) REFERENCES Management(ManagementID)
        ON DELETE NO ACTION
        ON UPDATE CASCADE
)
```

```
CREATE TABLE Management(
        ManagementID INT,
        Name VARCHAR(50),
        Email VARCHAR(320),
        PhoneNumber VARCHAR(20),
        PRIMARY KEY(ManagementID)
)
```

```
CREATE TABLE LeasingCompany(
        ManagementID INT,
        Url VARCHAR(2048),
        BranchName VARCHAR(50),
        PRIMARY KEY(ManagementID)
        FOREIGN KEY(ManagementID) REFERENCES Management(ManagementID)
        ON DELETE CASCADE
```

```
        ON UPDATE CASCADE,
)
```

```
CREATE TABLE Broker(
        ManagementID INT,
        PRIMARY KEY(ManagementID)
        FOREIGN KEY(ManagementID) REFERENCES Management(ManagementID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
)
```

```
CREATE TABLE Owner(
        ManagementID,
        Validated BOOLEAN,
        PRIMARY KEY(ManagementID)
        FOREIGN KEY(ManagementID) REFERENCES Management
        ON DELETE CASCADE
        ON UPDATE CASCADE,
)
```

```
CREATE TABLE Visit(
        DateTime DATETIME,
        Virtual BOOLEAN,
        UnitID INT,
        SearcherID INT
        PRIMARY KEY(UnitID, SearcherID),
        FOREIGN KEY(UnitID) REFERENCES Unit(UnitID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY(SearcherID) REFERENCES Searcher(SearcherID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
)
```

```
CREATE TABLE Application(
         DateReceived DATETIME,
        CurrentStatus VARCHAR(20),
        UnitID INT,
        SearcherID INT
```

```
    PRIMARY KEY(UnitID, SearcherID),
    FOREIGN KEY(UnitID) REFERENCES Unit(UnitID)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    FOREIGN KEY(SearcherID) REFERENCES Searcher(SearcherID)
    ON DELETE CASCADE
    ON UPDATE CASCADE
)
```

```
CREATE TABLE Searcher(
    SearcherID INT,
    FullName VARCHAR(50),
    Email VARCHAR(320),
    PhoneNumber VARCHAR(20),
    DateOfBirth DATE,
    CurrentAddress VARCHAR(50),
    ReasonForLeave VARCHAR(100),
    MonthlyIncome INT,
    Password VARCHAR(128),
    PRIMARY KEY(SearcherID),
    ON DELETE NO ACTION
    ON UPDATE CASCADE
)
```

```
CREATE TABLE Testimonial(
    LetterOfRecomm VARCHAR(500),
    FullName VARCHAR(50),
    Email VARCHAR(320),
    Relationship VARCHAR(100),
    SearcherID INT,
    PRIMARY KEY(Email, SearcherID),
    FOREIGN KEY(SearcherID) REFERENCES Searcher(SearcherID)
    ON DELETE CASCADE
    ON UPDATE CASCADE
)
```

```
CREATE TABLE Contract(
    ContractNumber INT,
```

```
        SearcherID INT,
        MoveInDate DATE,
        MoveOutDate DATE,
        MonthlyPrice INT,
        ManagementID INT,
        PRIMARY KEY(ContractNumber),
        FOREIGN KEY(SearcherID) REFERENCES Searcher(SearcherID)
        ON DELETE CASCADE
        ON DELETE UPDATE,
        FOREIGN KEY(ManagementID)
        REFERENCES Management(ManagementID),
        ON DELETE CASCADE
        ON UPDATE CASCADE
)

CREATE TABLE Contract_Duration(
         MoveInDate DATE,
         MoveOutDate DATE,
         Duration INT,
         FOREIGN KEY(MoveInDate) REFERENCES Contract(MoveInDate),
         FOREIGN KEY(MoveOutDate) REFERENCES Contract(MoveOutDate),
)
```

```
CREATE TABLE Unit_Searcher(
        SearcherID INT,
        UnitID INT,
        PRIMARY KEY(SearcherID, UnitID),
        FOREIGN KEY(SearcherID) REFERENCES Searcher(SearcherID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY(UnitID) REFERENCES Unit(UnitID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
)
```

```
CREATE TABLE Unit_Platform(
        UnitID INT,
```

```
    PlatformID INT,
    PRIMARY KEY(UnitID, PlatformID),
    FOREIGN KEY(UnitID) REFERENCES Unit(UnitID)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    FOREIGN KEY(PlatformID) REFERENCES Platform(PlatformID)
    ON DELETE CASCADE
    ON UPDATE CASCADE
)
```

```
CREATE TABLE Unit_School(
    UnitID INT,
    SchoolID INT,
    Distance FLOAT,
    PRIMARY KEY(UnitID, SchoolID),
    FOREIGN KEY(UnitID) REFERENCES Unit(UnitID)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    FOREIGN KEY(SchoolID) REFERENCES School(SchoolID)
    ON DELETE CASCADE
    ON UPDATE CASCADE
)
```

University of British Columbia, Vancouver
Department of Computer Science

SQL DDL INSERT STATEMENTS

```
INSERT INTO NameToAcronym (Name, Acronym) VALUES
(“The University of British Columbia - Vancouver”, “UBCV”),
(“The University of British Columbia - Okanagan”, “UBCO”),
(“University of Southern California”, “USC”),
(“University of South Carolina”, “USC”),
(“Simon Fraser University”, “SFU”);

INSERT INTO SchoolDetails (SchoolID, Name, Address) VALUES
(1, “The University of British Columbia - Vancouver”, “2329 West Mall, Vancouver, BC, Canada”),
(2, “The University of British Columbia - Okanagan”, “3287 University Way, Kelowna, BC, Canada”),
(3, “University of Southern California”, “Los Angeles, CA 90007, USA”),
(4, “University of South Carolina”, “Columbia, SC 29208, USA”),
(5, “Simon Fraser University”, “8888 University Dr W, Burnaby, Canada”);
```

```
INSERT INTO SupportContact (ContactID, FullName, Email, PhoneNumber, SchoolID) VALUES
(1, “Aayush Kogar”, “18aayushk@gmail.com”, “7783173657”, “abcdefgh1234”, 1),
(2, “Joe Doe”, “jdoe@gmail.com”, “7783173847”, “dsofhoise123”, 1),
(3, “Lebron James”, “lbj@gmail.com”, “7783109373”, “ahfoaih3432”, 2),
(4, “Russell Westbrook”, “rs@gmail.com”, “7283104373”, “aiobifroih243”, 3),
(5, “Carmelo Anthony”, “melo@gmail.com”, “2392394234”. “123hoih38f”, 4),
(6, “JR Smith”, “clutchcity@gmail.com”, “2312352349”, “afhoieoc32c”, 5);

INSERT INTO SupportContactPasswords (Email, Password) VALUES
(“18aayushk@gmail.com”, “abcdefgh1234”),
(“jdoe@gmail.com”, “dsofhoise123”),
(“lbj@gmail.com”, “ahfoaih3432”),
(“rs@gmail.com”,“aiobifroih243”),
(“melo@gmail.com” ,“123hoih38f”),
(“clutchcity@gmail.com”“afhoieoc32c”);
```

```
INSERT INTO Platform (PlatformID, Name, Url) VALUES
(1, “Rentals Canada”, “https://rentals.ca”),
(2, “Zumper”, “https://zumper.com”),
```

(3, “Craiglist”, “https://craiglist.org”),
(4, “Apartments”, “https://apartments.com”),
(5, “First Choice Housing”, “https://firstchoicehousing.com”);

INSERT INTO Unit (UnitID, Size, NoBedrooms, NoBathrooms, Address, Furnished, LeaseTerm, Sublet, MoveInDate, AskingPrice, SchoolID, ManagementID) VALUES
(1, 1439.5, 2, 2, “5410 Shortcut Rd, Vancouver, BC, Canada”, TRUE, 365, FALSE, 2023-09-01, 3400, 1, 1),
(2, 852, 2, 1, “2233 Allison Rd, Vancouver, BC, Canada”, FALSE, 730, FALSE, 2023-08-05, 2800, 1, 2),
(3, 950.5, 1, 1, “5410 Shortcut Rd, Vancouver, BC, Canada”, TRUE, 365, FALSE, 2023-11-10, 3400, 1, 1),
(4, 1100, 3, 1, “1211 W 37th Dr, CA 90007, USA”, TRUE, 730, FALSE, 2023-09-01, 3400, 3, 5),
(5, 646, 2, 2, “9266 University Crescent #901, Burnaby, BC, Canada”, TRUE, 365, TRUE, 2023-09-01, 3500, 5, 4);

INSERT INTO Management (ManagementID, Name, Email, PhoneNumber) VALUES
(1, NULL, NULL, “604-727-4608”),
(2, “Cindy Zhang”, NULL, “844.243.9986”),
(3, “Jovi Realty Inc”, “rentalhousevan@gmail.com”, NULL),
(4, “First Choice Housing”, “leasing@firstchoicehousing.com”, “(213)-765-3330”),
(5, “Don Kim”, NULL, “8443265877”),
(6, “Sitings Realty”, NULL, “604-684-6767”),
(7, “Terence May”, NULL, “310-270-5928”),
(8, “James”, NULL, “6046032726 ext. 0873”),
(9, “Western Rental Property Management Group”, “info@westernrental.ca”, “6042999680”)
(10, “Tarence Wong”, “twong@gmail.com”, “6673943434”),
(11, “Tony Kim”, “tkim@gmail.com”, “231.747.0378”),
(12, “Chana Firestone”, NULL, “(213) 603-9293”),
(13, “Nick Phillips”, “nphilips@gmail.com”, “338-474-2293”)
(14, “J Tatem”, “jt@gmail.com”, NULL)
(15, “Corey Wong”, “cwong238@gmail.com”, “778-374-0989”);
(16, “Khammy Kate”, “khammysah@gmail.com”, “778-177-0303”);

INSERT INTO LeasingCompany (ManagementID, Url, BranchName) VALUES

```
(2, “https://realtor.ca”, “Burnaby”),
(4, “https://firstchoicehousing.com”, NULL),
(5, “https://realtor.ca”, “Burnaby”),
(6, “https://wesbrookproperties.com”, NULL),
(9, “https://sutton1stwest.com”, “Richmond”),
(12, “https://compass.com/homes-for-sale/”, “Los Angeles”;,
```

```
INSERT INTO IndividualBroker (ManagementID) VALUES
(3),
(7),
(10),
(11),
(3);
```

```
INSERT INTO Owner (ManagementID, Validated) VALUES
(1, FALSE),
(8, TRUE),
(14, TRUE),
(15, FALSE),
(16, TRUE);
```

```
INSERT INTO Visit (DateTime, Virtual, UnitID, SearcherID) VALUES
(2023-08-05 17:00:00, FALSE, 1, 3),
(2023-08-05 17:15:00, FALSE, 1, 4),
(2023-09-04 08:15:00, TRUE, 2, 4),
(2023-09-04 08:45:00, TRUE, 3, 4),
(2023-10-04 09:45:00, TRUE, 5, 4);
```

```
INSERT INTO Application (DateReceived, CurrentStatus, UnitID, SearcherID) VALUES
(2023-07-04, “PENDING”, 1,1),
(2023-05-04, “REJECTED”, 1,2),
(2023-05-17, “ACCEPTED-AWAITING CONTRACT SIGNING”, 1,3),
(2023-06-05, “PENDING”, 2,2),
(2023-06-07, “ACCEPTED-AWAITING CONTRACT SIGNING”, 3,1);
```

```
INSERT INTO Searcher (SearcherID, FullName, Email, PhoneNumber, DateOfBirth,
CurrentAddress, ReasonForLeave, MonthlyIncome, Password) VALUES
(1, “Rahul Dravid”, “rdravid1@gmail.com”, “778-333-3333”, 2000-07-06, “3945
Student Union Blvd, Vancouver, BC, Canada”, “Roommates”, 3700, “abcd4343”),
```

(2, “Virat Kohli”, “vk2@gmail.com”, “378-328-3283”, 1998-07-04, “3533 Ross Drive, Vancouver, BC, Canada”, “Cost”, 2000, “aui3h34jkfe”),
(3, “Jason Kidd”, “msd_csk@gmail.com”, “238-3874-3822”, 2002-07-04, “1405 Armacost Rd, Parkton, MD 21120, USA”, NULL, 200, “DHONI233”),
(4, “Serena Williams”, “swilliamsiscool@hotmail.com”, “238-238-2378”, 2001-08-21, “100 SPAUGH LN, MOCKSVILLE NC 27028-5438, USA”, “Moving closer to campus”, 4000, “PASSWORDISPASSWORD”),
(5, “Lin Dan”, “badminton232@gmail.com”, “283-2389-4444”, 2002-04-09, “101 SPAUGH LN, MOCKSVILLE NC 27028-5438, USA”, “Moving closer to campus”, 4000, “PASSWORDISPASSWORD”);

INSERT INTO Testimonial (LetterOfRecomm, FullName, Email, Relationship, SearcherID) VALUES
("I have known Rahul for over five years as a neighbor, and I can confidently say that he is a responsible and trustworthy individual. He always takes care of his property and maintains a neat and clean living space. I have no doubt that he will treat any rental property with the same level of respect and care.", “Sarah Michell”, “smich@gmail.com”, “Neighbor”, 1),
("Rahul was my roommate for two years during college, and I couldn't have asked for a better person to live with. He was always prompt with his share of the rent, kept our apartment tidy, and was considerate of everyone's privacy. He is a reliable and respectful tenant who will be an asset to any landlord.", “Mark Coombs”, “mcoom338@gmail.com”, “Former Roommate”, 1),
("As Virat’s former supervisor at his previous job, I can attest to his exceptional work ethic and reliability. He was consistently punctual, took initiative, and demonstrated excellent communication skills. I have no doubt that Virat’s responsible nature will extend to his rental agreement, and he will be a tenant who can be trusted to fulfill his obligations.", “Mark Coombs”, “mcoom338@gmail.com”, “Former Supervisor”, 2),
("I have known Serena for many years as a close friend, and I can vouch for her integrity and character. She is a genuinely kind and considerate person who goes out of her way to help others. Serena has always been a responsible individual, and I have no doubt that he will treat any rental property with the utmost care and respect.", “Bob Marley”, “bobbym@gmail.com”, “Brother”, 4)
("As Jason's college professor, I had the pleasure of witnessing his dedication to his studies and his commitment to achieving excellence. He was a diligent student, always prepared for class, and actively engaged in discussions. Jason's responsible attitude extends beyond academics, and I believe he will approach his rental

obligations with the same level of diligence and professionalism.", “Dr. Johnson”, “jb@ubc.ca”, “College Professor”, 3);

```
INSERT INTO Contract (ContractNumber, SearcherID, MoveInDate, MoveOutDate,
MonthlyPrice, ManagementID) VALUES
(1, 5, 2023-08-21, 2024-08-21, 3400, 2)
(2, 4, 2024-08-21, 2025-08-21, 4100, 3)
(3, 3, 2024-08-21, 2024-08-21, 3330, 2)
(4, 2, 2023-08-21, 2025-08-21, 1800, 4)
(5, 1, 2023-08-21, 2024-08-21, 1100, 5)


INSERT INTO Contract_Duration (MoveInDate, MoveOutDate, Duration) VALUES
(2023-08-21, 2024-08-21, 365),
(2024-08-21, 2025-08-21, 730),
(2024-08-21, 2024-08-21, 365),
(2023-08-21, 2025-08-21, 730),
(2023-08-21, 2024-08-21, 365);
```

```
INSERT INTO Unit_Searcher (SearcherID, UnitID) VALUES
(1, 1),
(1, 2),
(1, 3),
(2, 1),
(2, 2),
(3, 1);
```

```
INSERT INTO Unit_Platform (UnitID, PlatformID) VALUES
(1, 1),
(2, 1),
(3, 1),
(4, 2),
(5, 2),
(5, 4);
```

```
INSERT INTO Unit_School (UnitID, SchoolID, Distance) VALUES
(1, 1, 3.2),
(1, 2, 12.5),
```

```
(2, 1, 6.5),
(3, 1, 7.5),
(4, 3, 12.4),
(5, 4, 22.2),
(5, 5, 0.3);
```