

CPSC 304 Project Cover Page

Milestone #: 2

Date: 7/27/2023

Group Number: 19

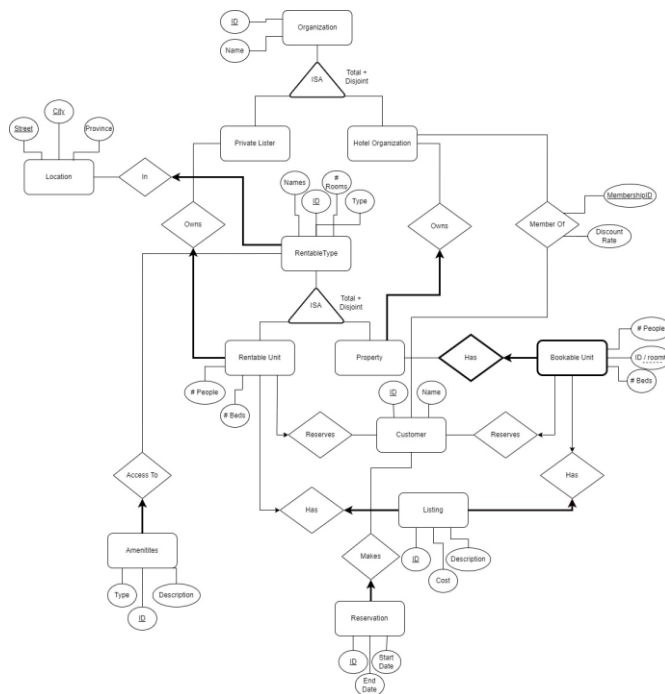
Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Vicki Zhang	97350599	S6p1j	vickiszhang@gmail.com
Ayush Bharat	15931678	u3q9w	ayshbrt210@gmail.com
Jeffrey Gao	71097984	D6w2b	Jeffrey.MagmaChu@gmail.com

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

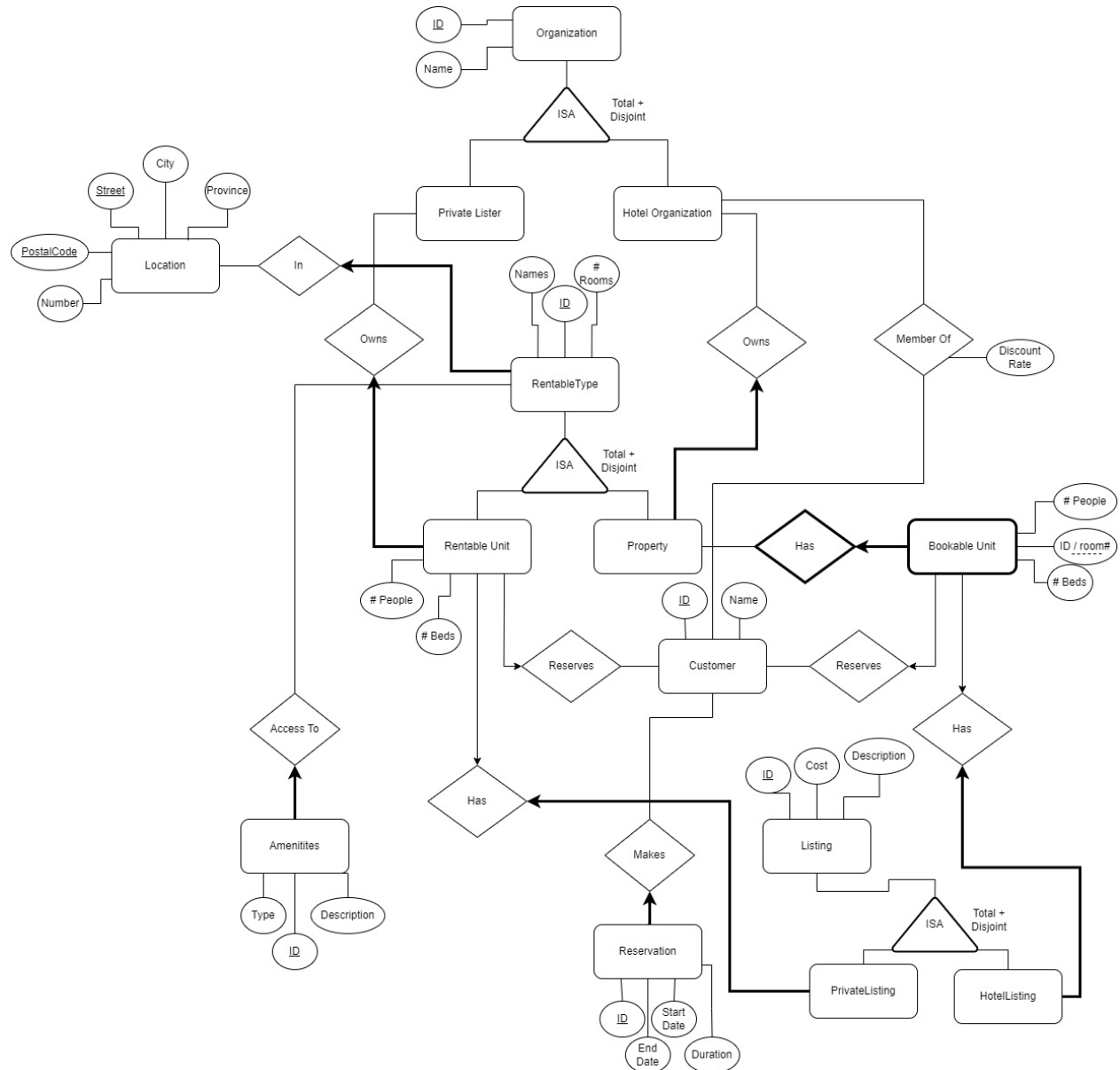
Changes:

- The ER diagram has a Location entity added to the Rentable Type entity to add up to 7 entities. A private lister or a hotel organization should be able to own multiple properties or rentable units in the same location.
- #People and #Beds attributes have been added to the Rentable Unit entity because that information should be represented when the listing is posted.
- membershipID removed from MemberOf relationship because it cannot have a key
- Split Listing entity to an ISA relationship, with the two subclasses being PrivateListing and HotelListing. This is because in our previous diagram, we had an issue where we should know either the rentable unit or bookable unit if we know the listing but not both. The new change allows us to ensure the separation of the type of listing.
- Removed “type” attribute from rentableType entity, because we are unsure of what the value of type would be.



Before:

Final:



ERD to Relational Schema

PrivateLister(PrivateOrganization_ID: integer, Name: string)

- Candidate keys: name
- Functional Dependencies: PrivateOrganization_ID -> Name, Name -> PrivateOrganization_ID

HotelOrganization(HotelOrganization_ID: integer, Name: string)

- Candidate Keys: Name
- Functional Dependencies: HotelOrganization_ID -> Name, Name -> HotelOrganization_ID
- Constraint: No two hotel organizations can have the same name, this would mean Name would be UNIQUE in the DDL

Customer(Customer_ID: integer, Name: string)

- Candidate Keys: None
- Functional Dependencies: customer_ID -> Name

MakesReservation(Reservation_ID: integer, EndDate: date, StartDate: date, Duration : integer, **Customer_ID**: integer)

- Functional Dependencies:

StartDate + Duration -> EndDate,

EndDate + duration -> StartDate,

reservation_ID -> EndDate, StartDate, Duration, Customer_ID

- Constraint: CustomerID NOT NULL

Membership(DiscountRate: integer, HotelOrganization_ID: string, Customer_ID: string)

- Functional Dependencies:

HotelOrganization_ID -> DiscountRate (organizations have a fixed discount rate for members)

HotelOrganization_ID + Customer_ID -> DiscountRate

Location(City: string, Street: string, Province: string, postalcode: string, number: integer)

- Functional Dependencies:

Street, postalCode, Number -> City, Province

University of British Columbia, Vancouver

Department of Computer Science

postal code -> city, province

Amenities(Amenities_ID: integer, Type: string, Description: string, **RentableUnit_ID**: integer, **Property_ID**: integer)

- Functional Dependencies: Amenities_ID -> Type, Description, RentableUnit_ID, Property_ID

Property(Property_ID: integer, **HotelOrganization_ID**: integer, Name: string, NumRooms: integer)

- Functional Dependencies: Property_ID -> HotelOrganization_ID, Name, NumRooms
- Constraints: HotelOrganization_ID NOT NULL

RentableUnit(RentableUnit_ID: integer, NumPeople: integer, NumBeds: integer, **PrivateListing_ID**: integer, **Customer_ID**: integer, **PrivateOrganization_ID**: integer)

- Functional Dependencies: RentableUnit_ID -> NumPeople, NumBeds, PrivateListing_ID, Customer_ID, PrivateOrganization_ID
- Constraints: PrivateOrganization_ID NOT NULL

BookableUnit(Property_ID: integer, RoomNum: integer, NumPeople: integer, NumBeds: integer, **CustomerID**: integer, **HotelListingID**: integer)

- Functional Dependencies: Property_ID + RoomNum -> NumPeople, NumBeds, CustomerID, HotelListingID

PrivateListing(PrivateListing_ID: integer, Cost: integer, Description: string, **RentableUnit_ID**: integer)

- Functional Dependencies: PrivateListing_ID -> Cost, Description, RentableUnit_ID
- Constraint: RentableUnit_ID NOT NULL

HotelListing(HotelListing_ID: integer, Cost: integer, Description: string, **Property_ID**: integer, **RoomNumber**: integer)

University of British Columbia, Vancouver

Department of Computer Science

- Functional Dependencies: HotelListing_ID_ -> Cost, Description, Property_ID, RoomNumber
- Constraints: RoomNumber, Property_ID NOT NULL

Normalized Tables

Location split into two tables:

Location(City: string, Street: string, Province: string, postalcode: string, number: integer)

Functional Dependencies:

- Street, PostalCode, Number -> City
 - Street, PostalCode -> Province
 - PostalCode -> City
 - PostalCode -> Province
1. Location_1(PostalCode, City, Province)
 2. Location_2(Street, Number, PostalCode)
-
3. PrivateLister(PrivateOrganization ID: integer, Name: string)
 - CK: name
 4. HotelOrganization(HotelOrganization ID: integer, Name: string)
 - CK: name
 5. Customer(Customer ID: integer, Name: string)

MakesReservation split into two tables:

- MakesReservation(reservation_ID: integer, EndDate: date, StartDate: date, duration : integer, **Customer_ID**: integer)
- FD's:
- (1)StartDate,duration -> EndDate,
- (2)EndDate,duration -> StartDate
- (3) reservation_ID -> EndDate, StartDate, duration, CustomerID

Split into two tables using (1):

6. MakesReservation_1(reservation_id: int, **customerID**: int, **StartDate**: date, **duration**: int)
 - FK's: **customerID**, **StartDate + duration**
7. MakesReservation_2(StartDate: date, duration: int, EndDate: date)

Membership split into two tables:

- Membership(DiscountRate: int, **HotelOrganization_ID**: string, **CustomerID**: string)

Split using fd: organization_id -> DiscountRate:

8. Membership_1(HotelOrganization_id: integer, discount_rate: integer)
9. Membership_2(**HotelOrganization_ID**: integer, **customerID**: integer)

10. Amenities(Amenities_ID: integer, Type: string, Description: string, **RentableUnit_ID**: integer, **Property_ID**: integer)

11. Property(Property_ID: integer, **HotelOrganization_ID**: integer, Name: string, NumRooms: integer)

University of British Columbia, Vancouver

Department of Computer Science

12. RentableUnit(RentableUnit_ID: integer, NumPeople: integer, NumBeds: integer, **PrivateListing_ID**: integer, **Customer_ID**: integer, **PrivateOrganization_ID**: integer)

13. BookableUnit(Property_ID: integer, RoomNum: integer, NumPeople: integer, NumBeds: integer, **CustomerID**: integer, **HotellistingID**: integer)

14. PrivateListing(PrivateListing_ID: integer, Cost: integer, Description: string, **RentableUnit_ID**: integer)

15. HotelListing(Hotellisting_ID: integer, Cost: integer, Description: string, **Property_ID**: integer, **RoomNumber**: integer)

SQL DDL

```
CREATE TABLE Location_1(  
PostalCode STRING PRIMARY KEY,  
City STRING,  
Province STRING  
)
```

```
CREATE TABLE Location_2(  
PostalCode STRING,  
Street STRING,  
Number INTEGER,  
PRIMARY KEY (PostalCode, Street, Number)  
)
```

```
CREATE TABLE PrivateLister (
```



```
        ID INTEGER PRIMARY KEY,
        Name CHAR(30)
    )

CREATE TABLE HotelOrganization(
    ID INTEGER PRIMARY KEY,
    Name CHAR(30) UNIQUE
)

CREATE TABLE Customer (
    ID INTEGER PRIMARY KEY,
    Name CHAR(30)
)

CREATE TABLE MakesReservation_1(
    StartDate DATE,
    EndDate DATE,
    duration INTEGER,
    PRIMARY KEY(StartDate, EndDate)
)

CREATE TABLE MakesReservation_2(
    reservation_ID INTEGER PRIMARY KEY,
    StartDate DATE,
    EndDate DATE,
    CustomerID INTEGER NOT NULL,
    FOREIGN KEY (CustomerId) REFERENCES Customer
        ON DELETE CASCADE,
        ON UPDATE CASCADE,
```

University of British Columbia, Vancouver

Department of Computer Science

```
        FOREIGN KEY (StartDate, Duration) REFERENCES MakesReservation_1
ON DELETE CASCADE,

        ON UPDATE CASCADE,

)
```

```
CREATE TABLE Membership_1(

        DiscountRate INTEGER,

        HotelOrganization_ID INTEGER PRIMARY KEY

)
```

```
CREATE TABLE Membership_2(

        MembershipID INTEGER,

        DiscountRate INTEGER,

        organization_ID INTEGER,

        CustomerID INTEGER,

        PRIMARY KEY (HotelOrganization_ID, CustomerID),

        FOREIGN KEY (HotelOrganization_ID) REFERENCES Membership_1,

        FOREIGN KEY (CustomerID) REFERENCES Customer

)
```

```
CREATE TABLE Amenities (

        Amenities_ID INTEGER PRIMARY KEY

        Type CHAR(30),

        Description CHAR(200)

        RentableUnit_ID INTEGER,

        FOREIGN KEY (RentableUnit_ID) REFERENCES RentableUnit

        ON DELETE CASCADE,
```

University of British Columbia, Vancouver

Department of Computer Science

```
ON UPDATE CASCADE,  
FOREIGN KEY (Property_ID) REFERENCES Property_ID  
ON DELETE CASCADE,  
ON UPDATE CASCADE,  
)
```

```
CREATE TABLE Property(  
    Property_ID INTEGER PRIMARY KEY,  
    HotelOrganization_ID INTEGER NOT NULL,  
    Name CHAR(30),  
    NumRooms INTEGER,  
    FOREIGN KEY (HotelOrganization_ID) REFERENCES HotelOrganization,  
ON DELETE CASCADE,  
ON UPDATE CASCADE,  
)
```

```
CREATE TABLE RentableUnit(  
    RentableUnit_ID INTEGER PRIMARY KEY,  
    NumPeople INTEGER,  
    NumBeds INTEGER,  
    PrivateListing_ID INTEGER,  
    HotelOrganization_ID INTEGER NOT NULL,  
    Name CHAR(30),  
    NumRooms INTEGER,  
    FOREIGN KEY (HotelOrganization_ID) REFERENCES HotelOrganization,
```

ON DELETE CASCADE,

ON UPDATE CASCADE,

FOREIGN KEY (PrivateListing_ID) REFERENCES PrivateListing,

ON DELETE SET NULL,

ON UPDATE CASCADE,

FOREIGN KEY (Customer_ID) REFERENCES Customer,

ON DELETE SET NULL,

ON UPDATE CASCADE,

)

CREATE TABLE BookableUnit(

Property_ID INTEGER,

RoomNum INTEGER,

NumPeople INTEGER,

NumBeds INTEGER,

CustomerID INTEGER,

HotelListingID INTEGER,

PRIMARY KEY (Property_ID, RoomNum),

FOREIGN KEY (CustomerID) REFERENCES Customer,

ON DELETE SET NULL,

ON UPDATE CASCADE,

FOREIGN KEY (HotelListingID) REFERENCES HotelListing,

ON DELETE SET NULL,

ON UPDATE CASCADE,

)

University of British Columbia, Vancouver

Department of Computer Science

PrivateListing(PrivateListing_ID: integer, Cost: integer, Description: string, **RentableUnit_ID**: integer)

```
CREATE TABLE PrivateListing_ID(  
    PrivateListing_ID INTEGER PRIMARY KEY,  
    Cost INTEGER,  
    Description CHAR(200),  
    RentableUnit_ID INTEGER NOT NULL,  
    FOREIGN KEY (RentableUnit_ID) REFERENCES RentableUnit,  
    ON DELETE CASCADE,  
    ON UPDATE CASCADE,  
)
```

```
CREATE TABLE HotelListing(  
    HotelListing_ID INTEGER PRIMARY KEY,  
    Cost INTEGER,  
    Description CHAR(30),  
    Property_ID INTEGER NOT NULL,  
    RoomNumber INTEGER NOT NULL,  
    FOREIGN KEY (Property_ID, RoomNumber) REFERENCES BookableUnit,  
    ON DELETE CASCADE,  
    ON UPDATE CASCADE,  
)
```

7. Populate each table with at least 5 tuples, and probably more so that you can issue

University of British Columbia, Vancouver

Department of Computer Science

meaningful queries later on. Show the instance of each relation after inserting the tuples.

We recommend writing INSERT statements and submitting a .SQL file. However, screenshots of an excel tabular like format are accepted as well.

PrivateLister

PrivateOrganization_ID	Name
1	Vicki
2	Ayush
3	Jeffrey
4	Frank
5	Sam

HotelOrganization

HotelOrganization_ID	Name
1	Fairmont Hotel
2	Hyatt Regency
3	Sandman Hotel
4	Marriott
5	Rosewood

Customer

University of British Columbia, Vancouver

Department of Computer Science

Customer_ID	Name
1	Bob
2	John
3	Alex
4	Marie
5	Ken

MakesReservation_1

Reservation_ID	customer_ID	StartDate	Duration
1	2	13-2-23	7
2	3	18-6-23	2
3	5	23-12-23	4
4	4	8-9-23	10
5	2	20-2-23	5

MakeReservation_2

StartDate	Duration	EndDate
13-2-23	7	20-2-23
18-6-23	2	20-6-23
23-12-23	4	27-12-23

University of British Columbia, Vancouver

Department of Computer Science

8-9-23	10	18-9-23
20-2-23	5	25-2-23

RentableUnit

Property_ID	RoomNum	NumPeople	NumBeds	CustomerID	HotelListingID
1	142	1	1		1
2	364	2	2		2
3	2000	5	2		3
4	234	2	1		4
5	446	2	1		5

BookableUnit

Property_ID	RoomNum	NumPeople	NumBeds	CustomerID	HotelListingID
1	142	1	1		1
2	364	2	2		2
3	2000	5	2		3
4	234	2	1		4
5	446	2	1		5

PrivateListing_ID

University of British Columbia, Vancouver

Department of Computer Science

PrivateListing_ID	Cost	Description	RentableUnit_ID
1	600	small house, 3 bedrooms, kitchen	1
2	100	singular room with shared bathroom	2
3	200	singular room with private bathroom	3
4	300	small apartment, 2 bedroom 1 bathroom	4
5	200	treehouse	5

HotelListing

HotelListing_ID	Cost	Description	Property_ID	RoomNumber
1	200	Twin bed	1	142
2	250	Two twin	2	364
3	1000	2x king bed + jacuzzi + balcony	3	2000
4	300	water bed	4	234
5	200	bunk bed and canoe	5	446

Location_1

PostalCode	City	Province
A5B3C7	Richmond Knoll	Allofit
B2D3S6	Eastminster	Twotario

University of British Columbia, Vancouver

Department of Computer Science

S2B5D3	Boronto	French Columbia
B3G6S2	Vanscooter	Princess Edwardo Land
B2S5D3	Canada	Oldfoundland

Location_2

PostalCode	Street	Number
A5B3C7	Second St	2
B2D3S6	Dinger Rd	2633
S2B5D3	Schule St	562
B3G6S2	West St E	69
B2S5D3	Lane Rd	555

Membership_1

HotelOrganization_ID	DiscountRate
1	20
2	45
3	12
4	15
5	15

University of British Columbia, Vancouver

Department of Computer Science

Membership_2

HotelOrganization_ID	CustomerID
1	2
2	2
3	4
4	3
5	1

Amenities

Amentites_ID	Type	Description	RentableUnit_ID	Property_ID
1	Pool	20 sq foot pool in lobby	1	Null
2	Lounge	Amazing lounge in the lobby of our building	Null	2
3	Cafeteria	Free breakfast	Null	3
4	Pool	Outdoor large pool	3	Null
5	Theater	In house movie theater	1	Null

Property

Property_ID	HotelOrganization_ID	Name	NumRooms
1	1	Fairmont Vancouver	100

University of British Columbia, Vancouver

Department of Computer Science

2	1	Fairmont Whistler	200
3	3	Hyatt Vancouver	150
4	4	Marriott Vancouver	100
5	5	Rosewood Burnaby	50

RentableUnit

RentableUnit_ID	NumPeople	NumBeds	PrivateListing_ID	Customer_ID	PrivateOrganization_ID
1	6	3	1	2	1
2	3	2	2	3	Null
3	1	1	Null	1	3
4	3	2	4	5	Null
5	3	2	5	4	Null