

CISC 332

QBnB Project Deliverable 2

Zachary Baum 10090150

Vinyas Harish 10089169

March 10th 2016

Table of Contents

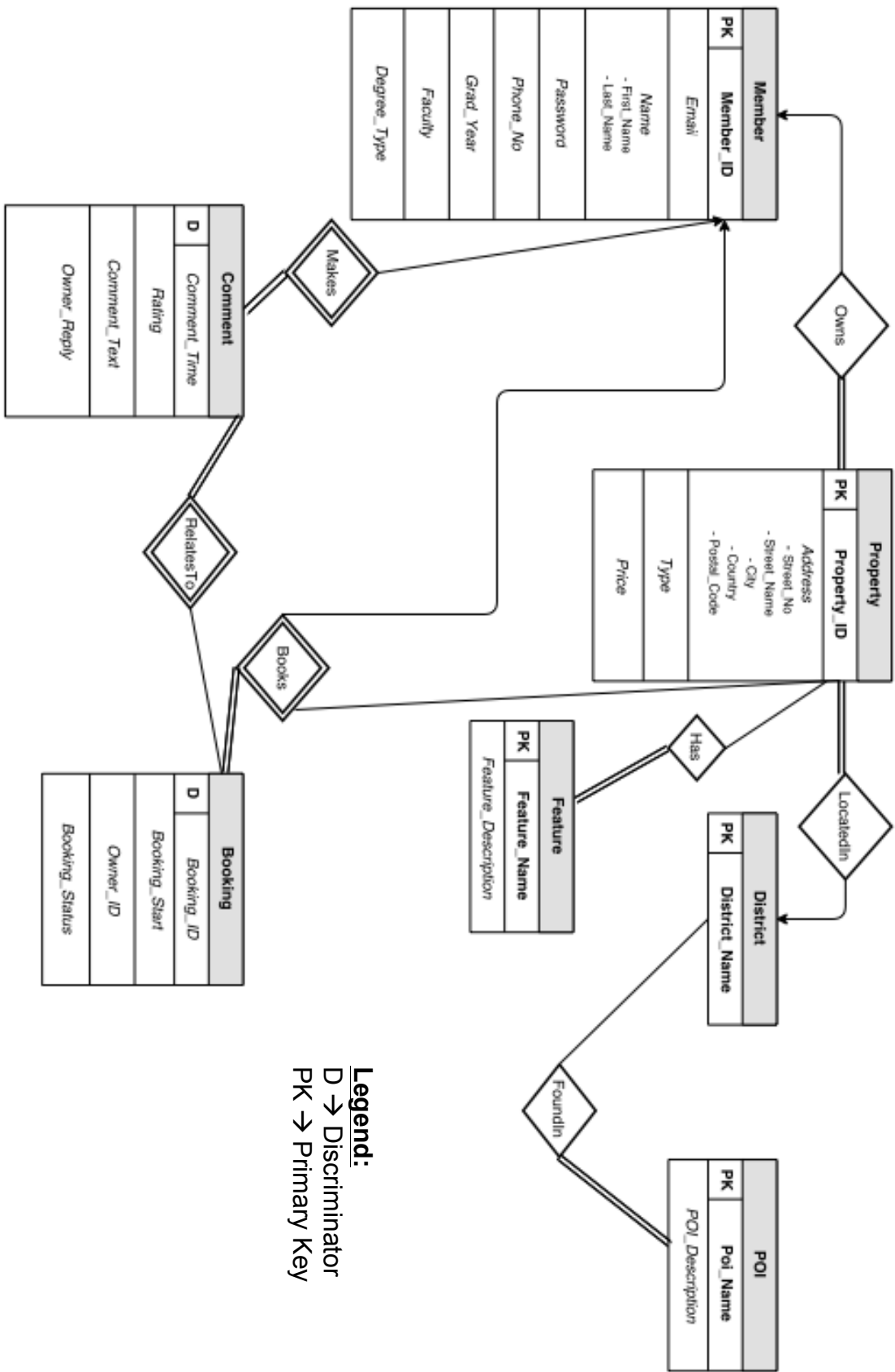
CONTINUED ASSUMPTIONS	1
UPDATED ER DIAGRAM	3
UPDATED RELATIONAL SCHEMA & TABLE DUMPS	4
INITIAL TABLE DUMPS	5
REQUIRED SQL STATEMENTS & SAMPLE OUTPUTS	7
GENERIC STATEMENTS - BY USER	7
MEMBER	7
OWNER	7
CONSUMER	8
ADMIN	9
DEMO STATEMENTS - BY USER	10
MEMBER	10
OWNER	11
CONSUMER	12
ADMIN	15
STATE MACHINE DIAGRAM	18
USER INTERFACE FLOW	19
MAIN PAGE	19
SIGN UP PAGE	19
SIGN IN PAGE	19
USER DASHBOARD	19
SETTINGS PAGE	20
ADMINISTRATOR DASHBOARD	20

Continued Assumptions

- 1) We have used Member_ID as the primary key for our Member table, restricting each email to being associated with one account on QBnB. We are assuming since this is a Queen's alumni database we will not have an integer overflow with the number of registered users.
- 2) Users will not enter invalid information or forms that are incomplete when using our application.
- 3) All Properties in the Property table must be owned by some member of QBnB.
- 4) One member may own multiple properties, but multiple members cannot own the same property.
- 5) Each property belongs to one and only one district, but a district may encompass multiple properties.
- 6) All bookings must be approved (or denied). Users can only cancel bookings if they are "pending". All bookings are for one week in duration.
- 7) The member that owns a rentable property is able to approve (or deny) any bookings, and all bookings need only be approved once.
- 8) Owner_ID corresponds to the Member_ID of the owner of the property in question.
- 9) Any member may comment as many times as they want on a property, but the owner of the property is the only user who may reply directly to those comments on their property.
- 10) In order to comment on a property, a member must have booked it at some point in time. The property being commented on must exist.
- 11) Administrator accounts will be limited to five, and will be the first five accounts created in the Member table (Member_IDs 1, 2, 3, 4 and 5).
- 12) When leaving feedback on a property, a comment is required, but a rating (out of five) is optional.
- 13) An average rating will only be shown if the property being shown has at least one rating given to it.
- 14) When in the user dashboard, administrators will have the same abilities as an equivalent user (consumer if they do not own any properties or supplier if they own at least one).
- 15) Any fields can be updated by a member except for the Member_ID.
- 16) A property is available for booking if and only if there is no booking for it listed at that time, or the booking(s) that is/are currently made on it are listed as rejected or pending for that time.
- 17) When browsing properties, users will only be able to filter their searches by one of District, Price, Feature ,or Type at a time.

- 18) Any fields can be updated to a property except for the Property_ID.
- 19) When deleting an accommodation, the comments and bookings associated with it are also deleted.
- 20) When deleting a user account, all properties and the comments and bookings with those properties are deleted.

Updated ER Diagram



Updated Relational Schema & Table Dumps

```
CREATE DATABASE QBnB;
USE QBnB;
CREATE TABLE Member(
    Member_ID int NOT NULL AUTO_INCREMENT,
    F_Name varchar(20) NOT NULL,
    L_Name varchar(20) NOT NULL,
    Email varchar(50) NOT NULL,
    Phone_No bigint(15) NOT NULL,
    Grad_Year int(4) NOT NULL,
    Faculty varchar(20) NOT NULL,
    Degree_Type char(10) NOT NULL,
    Password varchar(180) NOT NULL,
    PRIMARY KEY (Member_ID),
    UNIQUE (Email)
);

CREATE TABLE District(
    District_Name varchar(30) NOT NULL,
    PRIMARY KEY (District_Name)
);

CREATE TABLE POI(
    District_Name varchar(30) NOT NULL,
    POI_Name varchar (30) NOT NULL,
    POI_Description varchar(500) DEFAULT NULL,
    PRIMARY KEY (District_Name, POI_Name),
    FOREIGN KEY (District_Name) references District(District_Name) ON UPDATE CASCADE
    ON DELETE CASCADE
);

CREATE TABLE Property(
    Property_ID int NOT NULL AUTO_INCREMENT,
    Owner_ID int NOT NULL,
    Street_No int NOT NULL,
    Street_Name varchar(30) NOT NULL,
    City varchar(30) NOT NULL,
    Country varchar(30) NOT NULL,
    Postal_Code varchar(10) NOT NULL,
    District_Name varchar(30) NOT NULL,
    Type varchar(15) NOT NULL,
    Price decimal NOT NULL,
    PRIMARY KEY (Property_ID),
    FOREIGN KEY (District_Name) references District(District_Name) ON UPDATE CASCADE
    ON DELETE CASCADE
    FOREIGN KEY (Owner_ID) references Member(Member_ID) ON UPDATE CASCADE
    ON DELETE CASCADE
);

CREATE TABLE Feature(
    Property_ID int NOT NULL,
    Feature_Name varchar(100) NOT NULL,
    Feature_Description varchar(500) DEFAULT NULL,
    PRIMARY KEY (Property_ID, Feature_Name),
    FOREIGN KEY (Property_ID) references Property(Property_ID) ON UPDATE CASCADE
    ON DELETE CASCADE
);
```

```

CREATE TABLE Booking(
  Booking_ID int NOT NULL AUTO_INCREMENT,
  Property_ID int NOT NULL,
  Member_ID int NOT NULL,
  Owner_ID int NOT NULL,
  Booking_Start datetime NOT NULL,
  Booking_Status char(15) NOT NULL,
  PRIMARY KEY (Booking_ID, Property_ID, Member_ID),
  FOREIGN KEY (Property_ID) references Property(Property_ID) ON UPDATE CASCADE
  ON DELETE CASCADE,
  FOREIGN KEY (Member_ID) references Member(Member_ID) ON UPDATE CASCADE
  ON DELETE CASCADE,
  FOREIGN KEY (Owner_ID) references Property(Owner_ID) ON UPDATE CASCADE
  ON DELETE CASCADE
);

CREATE TABLE Comment(
  Booking_ID int NOT NULL,
  Comment_Time timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  Member_ID int NOT NULL,
  Rating int DEFAULT NULL,
  Comment_Text varchar(500) NOT NULL,
  Owner_Reply varchar(500) DEFAULT NULL,
  PRIMARY KEY (Booking_ID, Comment_Time),
  FOREIGN KEY (Booking_ID) references Booking(Booking_ID) ON UPDATE CASCADE
  ON DELETE CASCADE,
  FOREIGN KEY (Member_ID) references Member(Member_ID) ON UPDATE CASCADE
  ON DELETE CASCADE
);

```

Initial Table Dumps

Member_ID	F_Name	L_Name	Email	Phone_No	Grad_Year	Faculty	Degree_Type	Password
1	QBnB	Admin	admin@qbnb.ca	6135336666	2000	Computing	BComp	admin
2	Vinyas	Harish	v.harish@queensu.ca	6477675831	2017	Computing	BComp	vin
3	Zac	Baum	zac.baum@queensu.ca	4163203344	2017	Computing	BComp	zac
4	Patrick	Martin	martin@cs.queensu.ca	6131113333	2000	Computing	BComp	cisc332prof
5	Laura	Brooks	laura.brooks@queensu.ca	4164359453	2016	Computing	BComp	cisc332ta

Property_ID	Owner_ID	Street_No	Street_Name	City	Country	Postal_Code	District_Name	Type	Price
1	1	12	Brock Street	Toronto	Canada	M9A 4X6	Entertainment District	Apartment	35
2	2	12	Bay Street	Toronto	Canada	M4A 9X6	Financial District	Loft	3000
3	1	50	Gerrard Street	Toronto	Canada	M9A 4X1	Danforth	Basement	50
4	3	41	Yonge Street	Toronto	Canada	M9A 4X9	North Toronto	Apartment	75
5	4	1891	Quebec Avenue	Toronto	Canada	M6T 4Q9	High Park	House	15

Booking_ID	Comment_Time	Member_ID	Rating	Comment_Text	Owner_Reply
1	2016-03-08 11:07:58	2	5	Great little spot very reasonably priced!	Glad to hear! Thank you for your visit!
4	2016-03-08 11:07:58	5	1	Worst experience I have ever had!	NULL
5	2016-03-08 11:07:58	3	4	Great place with an even better view!	NULL

District_Name
Annex
Beaches
Cabbagetown
Chinatown
Danforth
Distillery District
Entertainment District
Financial District
Forest Hill
Gerrard Street East
Harbourfront
High Park
North Toronto
Rosedale
Scarborough City Centre
St. Lawrence
Yonge and Dundas
York Mills

Property_ID	Feature_Name	Feature_Description
1	1 bathroom	NULL
2	Full kitchen	NULL
3	Full kitchen	NULL
3	Mahogany floors	NULL
4	2 bathrooms	NULL
4	2 kitchens	NULL
4	Study	NULL
5	1 bathroom	NULL
5	2 bath	NULL
5	Full kitchen	NULL

Booking_ID	Property_ID	Member_ID	Owner_ID	Booking_Start	Booking_Status
1	5	2	4	2016-03-11 12:00:00	Approved
2	4	5	3	2016-06-11 12:30:00	Pending
3	2	3	2	2016-02-14 12:00:00	Rejected
4	4	5	3	2016-06-11 12:30:00	Approved
5	2	3	2	2016-02-14 12:00:00	Approved

District_Name	POI_Name	POI_Description
Annex	Jewish Community Centre	NULL
Annex	Koreatown	NULL
Annex	University of Toronto	NULL
Beaches	Ashbridges Bay	NULL
Beaches	Kew Gardens	NULL
Beaches	The Boardwalk	NULL
Beaches	The Goof	NULL
Beaches	Woodbine Beach	NULL
Cabbagetown	Riverdale Farm	NULL
Cabbagetown	The School of Toronto Dance Th	NULL
Chinatown	Chinatown Markets	NULL
Chinatown	Spadina Ave.	NULL
Danforth	Greektown	NULL
Danforth	Taste of the Danforth	NULL
Distillery District	Gooderham and Worts Distillery	NULL
Distillery District	Mill Street Brewery	NULL
Distillery District	Toronto Christmas Market	NULL
Entertainment District	Air Canada Centre	NULL
Entertainment District	CN Tower	NULL
Entertainment District	Ripleys Aquarium	NULL
Entertainment District	Rogers Centre	NULL
Financial District	Fairmont Place	NULL
Financial District	First Canadian Place	NULL
Financial District	Royal York Hotel	NULL
Financial District	Trump Tower	NULL
Financial District	Union Station	NULL
Forest Hill	Branksome Hall	NULL
Forest Hill	Eglinton Theatre	NULL

Gerrard Street East	Gerrard India Bazaar	NULL
Harbourfront	Billy Bishop Airport	NULL
Harbourfront	Harbourfront Centre	NULL
Harbourfront	Jack Layton Ferry Terminal	NULL
Harbourfront	Queens Quay	NULL
High Park	High Park	NULL
High Park	Sunnyside Docks	NULL
North Toronto	Chaplin Estates	NULL
North Toronto	Yonge and Eglinton	NULL
Rosedale	The Integral House	NULL
Rosedale	The Studio Building	NULL
Scarborough City Centre	Scarborough Town Centre	NULL
St. Lawrence	Sony Centre for the Performing	NULL
St. Lawrence	St. Lawrence Hall	NULL
St. Lawrence	St. Lawrence Market	NULL
Yonge and Dundas	Eatons Centre	NULL
Yonge and Dundas	Four Seasons Centre for the Pe	NULL
York Mills	Don Valley Golf Course	NULL
York Mills	York Mills Shopping Centre	NULL

Required SQL Statements & Sample Outputs

Generic Statements - by User

MEMBER

1.

#Register as member

```
INSERT INTO `Member`  
(`F_Name`,`L_Name`,`Email`,`Phone_No`,`Grad_Year`,`Faculty`,`Degree_Type`,`Password`,`Member_ID`)  
VALUES('value1','value2','value3',value4,value5,'value6','value7','value8',value9)
```

2.

#Update anything in profile

```
UPDATE `Member`  
SET attributel = value1, ...  
WHERE Member_ID = theirMemberID
```

3.

#Sign a member in

```
SELECT Member_ID  
FROM `Member`  
WHERE Email = theirEmailID AND Password = theirPassword
```

4.

#Remove their membership

```
DELETE FROM `Member`  
WHERE Member_ID = theirMemberID
```

OWNER

5.

#Create new accommodation

```
INSERT INTO `Property`  
(`Street_No`,`Street_Name`,`City`,`Country`,`Postal_Code`,`District_Name`,`Type`,`Price`,`Property_ID`,`Owner_ID`)  
VALUES (value1,'value2','value3','value4','value5','value6','value7', value8, value9)
```

6.

#Update existing accommodation

```
UPDATE `Property`  
SET attributel = value1, ...  
WHERE Property_ID = thePropertyID
```

7.

#Remove accommodation

```
DELETE FROM `Property`  
WHERE Property_ID = thePropertyID
```

8.

#Approve booking

```
UPDATE `Booking`  
SET Booking_Status = 'Approved'  
WHERE Booking_ID = theBookingID AND Booking_Start = theBookingStart
```

```

9.
#Reject booking
UPDATE `Booking`
SET Booking_Status = 'Rejected'
WHERE Booking_ID = theBookingID AND BookingStart = theBookingStart

10.
#Reply to comment
UPDATE `Comment`
SET Owner_Reply = ownerCommentText
WHERE Member_ID = commentersMemberID AND Booking_ID = theBookingID AND Comment_Time =
theCommentTime

11.
#List all owned properties + comments / ratings
SELECT Street_No, Street_Name, City, Price, Property.Property_ID, Comment_Text, Rating, Owner_Reply
FROM Property
LEFT JOIN (`Comment` NATURAL JOIN `Booking`)
ON Property.Property_ID = Booking.Property_ID

12.
#Add feature to property
INSERT INTO `Feature`(Property_ID, Feature_Name, Feature_Description)
VALUES (Value1, "Value2", "Value3")

CONSUMER

13.
#Search by district
SELECT DISTINCT Street_No, Street_Name, City, Country, District_Name, Type, Price, Property_ID
FROM Property
WHERE District_Name = selectedDistrict

14.
#Search by type
SELECT DISTINCT Street_No, Street_Name, City, Country, District_Name, Type, Price, Property_ID
FROM Property
WHERE Type = selectedType

15.
#Search by features
SELECT DISTINCT Street_No, Street_Name, City, Country, District_Name, Type, Price, Property_ID
FROM Property NATURAL JOIN Feature
WHERE Feature_Name = selectedFeature

16.
#Search by price
SELECT DISTINCT Street_No, Street_Name, City, Country, District_Name, Type, Price, Property_ID
FROM Property
WHERE Price <= maxPriceEntered AND Price >= minPriceEntered

17.
#List all ratings and comments for a listing
SELECT DISTINCT Property_ID, Comment_Text, Owner_Reply, Rating
FROM Comment NATURAL JOIN Booking
WHERE Booking.Property_ID = thePropertyID
ORDER BY Comment_Time

18.
#List availability of rental
SELECT Booking_ID, Booking_Status
FROM Booking
WHERE Booking_Start = currentWeekStart AND Property_ID = thePropertyID AND (Booking_Status =
"Approved")

```

```

19.
#Show owner details
SELECT DISTINCT F_Name, L_Name, Email, Phone_No
FROM Member NATURAL JOIN Property
WHERE Member_ID = ownerOfSelectedProperty

20.
#Place booking request
INSERT INTO `Booking` (`Property_ID`,`Booking_Start`,`Booking_Status`,`Member_ID`,`Owner_ID`)
VALUES (value1, value2,'Pending',value3,value4)

21.
#List all bookings by one member
SELECT Booking_ID, Member_ID, Property_ID, Street_No, Street_Name, City, Booking_Start,
Booking_Status
FROM Booking NATURAL JOIN Property
WHERE Member_ID = theMemberID
GROUP BY Booking_Status
ORDER BY Booking_Start

22.
#Show details of one booking
SELECT DISTINCT Street_No, Street_Name ,City, District_Name, Type, Price, Booking_Start,
Booking_Status, Owner_ID
FROM Booking NATURAL JOIN Property
WHERE Booking_ID = consumerMemberID AND Property_ID = thePropertyID

23.
#Add comment and rating for an accommodation
INSERT INTO `Comment` (`Member_ID`,`Property_ID`,`Rating`,`Comment_Text`)
VALUES (commenterMemberID, thePropertyID, ratingValue, 'commentText')

24.
#Cancel a booking
DELETE FROM `Booking`
WHERE Booking_ID = theBookingID

ADMIN

25.
#Delete a member and associated listings
DELETE FROM `Member`
WHERE Member_ID = theirMemberID

26.
#Delete an accommodation
DELETE FROM `Property`
WHERE Property_ID = thePropertyID

27.
#Summarize bookings and ratings per accomodation
SELECT Booking_Start, Booking_Status, Avg(Rating)
FROM Booking NATURAL JOIN Comment
WHERE Property_ID = thePropertyID
GROUP BY Booking_Status
ORDER BY Booking_Start

28.
#Summarize bookings and ratings per owner
SELECT Booking_Start, Booking_Status, Avg(Rating)
FROM Booking NATURAL JOIN Comment
WHERE Owner_ID = supplierMemberID
GROUP BY Booking_Status
ORDER BY Booking_Start

```

29.

```
#Summarize booking activity per customer
SELECT Booking_Start, Booking_Status
FROM Booking
WHERE Member_ID = customerMemberID
GROUP BY Booking_Status
ORDER BY Booking_Start
```

Demo Statements - by User

Note that the timestamps on comments may vary from those shown in the original table dumps because the database sometimes had to be dropped as part of the process in order to get the queries working.

MEMBER

1. #Register new member

```
INSERT INTO `Member`
(`F_Name`,`L_Name`,`Email`,`Phone_No`,`Grad_Year`,`Faculty`,`Degree_Type`,`Password`)
VALUES ('Elon','Musk','e.musk@tesla.com',4567866787,1985,'Commerce','BComm','money')
```

Member_ID	F_Name	L_Name	Email	Phone_No	Grad_Year	Faculty	Degree_Type	Password
1	QBnB	Admin	admin@qbnb.ca	6135336666	2000	Computing	BComp	admin
2	Vinyas	Harish	v.harish@queensu.ca	6477675831	2017	Computing	BComp	vin
3	Zac	Baum	zac.baum@queensu.ca	4163203344	2017	Computing	BComp	zac
4	Patrick	Martin	martin@cs.queensu.ca	6131113333	2000	Computing	BComp	cisc332prof
5	Laura	Brooks	laura.brooks@queensu.ca	4164359453	2016	Computing	BComp	cisc332ta
6	Elon	Musk	e.musk@tesla.com	4567866787	1985	Commerce	BComm	money

2. #Update anything in profile

```
UPDATE `Member`
SET Password = 'cash'
WHERE Member_ID = 6
```

6	Elon	Musk	e.musk@tesla.com	4567866787	1985	Commerce	BComm	cash
---	------	------	------------------	------------	------	----------	-------	------

3. #Sign in

```
SELECT Member_ID
FROM `Member`
WHERE Email = 'e.musk@tesla.com' AND Password = 'cash'
```

Member_ID
6

4. #Cancel membership

```
#Note: Elon Musk was still in the table prior to this statement
DELETE FROM `Member`
WHERE Member_ID = 6
```

Member_ID	F_Name	L_Name	Email	Phone_No	Grad_Year	Faculty	Degree_Type	Password
1	QBnB	Admin	admin@qbnb.ca	6135336666	2000	Computing	BComp	admin
2	Vinyas	Harish	v.harish@queensu.ca	6477675831	2017	Computing	BComp	vin
3	Zac	Baum	zac.baum@queensu.ca	4163203344	2017	Computing	BComp	zac
4	Patrick	Martin	martin@cs.queensu.ca	6131113333	2000	Computing	BComp	cisc332prof
5	Laura	Brooks	laura.brooks@queensu.ca	4164359453	2016	Computing	BComp	cisc332ta

OWNER

```
5. #Create new property
INSERT INTO `Property`
(`Street_No`,`Street_Name`,`City`,`Country`,`Postal_Code`,`District_Name`,`Type`,`Price`,`Owner_ID`)
VALUES (15,'Wynd Drive','Toronto','Canada','M7T G6H','Forest Hill','House',1000, 2)
```

Property_ID	Owner_ID	Street_No	Street_Name	City	Country	Postal_Code	District_Name	Type	Price
1	1	12	Brock Street	Toronto	Canada	M9A 4X6	Entertainment District	Apartment	35
2	2	12	Bay Street	Toronto	Canada	M4A 9X6	Financial District	Loft	3000
3	1	50	Gerrard Street	Toronto	Canada	M9A 4X1	Danforth	Basement	50
4	3	41	Yonge Street	Toronto	Canada	M9A 4X9	North Toronto	Apartment	75
5	4	1891	Quebec Avenue	Toronto	Canada	M6T 4Q9	High Park	House	15
6	2	15	Wynd Drive	Toronto	Canada	M7T G6H	Forest Hill	House	1000

```
6. #Update existing property information
UPDATE `Property`
SET Price = 1500
WHERE Property_ID = 6
```

6	2	15	Wynd Drive	Toronto	Canada	M7T G6H	Forest Hill	House	1500
---	---	----	------------	---------	--------	---------	-------------	-------	------

```
7. #Delete property
DELETE FROM `Property`
WHERE Property_ID = 6
```

Property_ID	Owner_ID	Street_No	Street_Name	City	Country	Postal_Code	District_Name	Type	Price
1	1	12	Brock Street	Toronto	Canada	M9A 4X6	Entertainment District	Apartment	35
2	2	12	Bay Street	Toronto	Canada	M4A 9X6	Financial District	Loft	3000
3	1	50	Gerrard Street	Toronto	Canada	M9A 4X1	Danforth	Basement	50
4	3	41	Yonge Street	Toronto	Canada	M9A 4X9	North Toronto	Apartment	75
5	4	1891	Quebec Avenue	Toronto	Canada	M6T 4Q9	High Park	House	15

```
8. #Approve booking
UPDATE `Booking`
SET Booking_Status = 'Approved'
WHERE Property_ID = 4 AND Booking_Start = "2016-06-11 12:30:00"
```

2	4	5	3	2016-06-11 12:30:00	Approved
---	---	---	---	---------------------	----------

```
9. #Reject booking
UPDATE `Booking`
SET Booking_Status='Rejected'
WHERE Booking_ID = 2 AND Booking_Start = "2016-06-11 12:30:00"
```

2	4	5	3	2016-06-11 12:30:00	Rejected
---	---	---	---	---------------------	----------

```

10. #Reply to comment
UPDATE `Comment`
SET Owner_Reply = "Glad you liked it!"
WHERE Member_ID = 3 AND Booking_ID = 5 AND Comment_Time = "2016-03-08 22:33:41"

```

5	2016-03-08 22:43:25	3	4 Great place with an even better view!	Glad you liked it!
---	---------------------	---	---	--------------------

```

11. #List all owned properties with comments
SELECT Street_No, Street_Name, City, Price, Property.Property_ID, Comment_Text, Rating,
Owner_Reply
FROM Property
LEFT JOIN (`Comment` NATURAL JOIN `Booking`)
ON Property.Property_ID = Booking.Property_ID

```

Street_No	Street_Name	City	Price	Property_ID	Comment_Text	Rating	Owner_Reply
12	Brock Street	Toronto	35	1		NULL	NULL
12	Bay Street	Toronto	3000	2		NULL	NULL
50	Gerrard Street	Toronto	50	3		NULL	NULL
41	Yonge Street	Toronto	75	4	Worst experience I have ever had!	1	NULL
1891	Quebec Avenue	Toronto	15	5	Great little spot very reasonably priced!	5	Glad to hear! Thank you for your visit!

```

12. #Add feature to property
INSERT INTO `Feature`(Property_ID, Feature_Name, Feature_Description)
VALUES (3, "Basketball court", "The floors of the basketball court are also mahogany!")

```

Property_ID	Feature_Name	Feature_Description
1	1 bathroom	NULL
2	Full kitchen	NULL
3	Basketball court	The floors of the basketball court are also mahoga...
3	Full kitchen	NULL
3	Mahogany floors	NULL
4	2 bathrooms	NULL
4	2 kitchens	NULL
4	Study	NULL
5	1 bathroom	NULL
5	2 bath	NULL
5	Full kitchen	NULL

CONSUMER

```

13. #Search by district
SELECT DISTINCT Street_No, Street_Name, City, Country, District_Name, Type, Price,
Property_ID
FROM Property
WHERE District_Name = "Entertainment District"

```

Street_No	Street_Name	City	Country	District_Name	Type	Price	Property_ID
12	Brock Street	Toronto	Canada	Entertainment District	Apartment	35	1

14.

```
#Search by type
SELECT DISTINCT Street_No, Street_Name, City, Country, District_Name, Type, Price, Property_ID
FROM Property
WHERE Type = "Loft"
```

Street_No	Street_Name	City	Country	District_Name	Type	Price	Property_ID
12	Bay Street	Toronto	Canada	Financial District	Loft	3000	2

15.

```
#Search by feature
SELECT DISTINCT Street_No, Street_Name, City, Country, District_Name, Type, Price,
Property_ID
FROM Property NATURAL JOIN Feature
WHERE Feature_Name = "Basketball court"
```

Property_ID	Owner_ID	Street_No	Street_Name	City	Country	Postal_Code	District_Name	Type	Price
3	1	50	Gerrard Street	Toronto	Canada	M9A 4X1	Danforth	Basement	50

16. #Search by price

```
SELECT DISTINCT Street_No, Street_Name, City, Country, District_Name, Type, Price,
Property_ID
FROM Property
WHERE Price <= 75 AND Price >= 15
```

Street_No	Street_Name	City	Country	District_Name	Type	Price	Property_ID
12	Brock Street	Toronto	Canada	Entertainment District	Apartment	35	1
50	Gerrard Street	Toronto	Canada	Danforth	Basement	50	3
41	Yonge Street	Toronto	Canada	North Toronto	Apartment	75	4
1891	Quebec Avenue	Toronto	Canada	High Park	House	15	5

17. #List all ratings and comments for a listing

```
SELECT DISTINCT Property_ID, Comment_Text, Owner_Reply, Rating
FROM Comment NATURAL JOIN Booking
WHERE Booking.Property_ID = 5
ORDER BY Comment_Time
```

Property_ID	Comment_Text	Owner_Reply	Rating
5	Great little spot very reasonably priced!	Glad to hear! Thank you for your visit!	5

18. #List availability of rental

```
SELECT Booking_ID, Booking_Status
FROM Booking
WHERE Booking_Start = "2016-06-11 12:30:00" AND Property_ID = 4 AND (Booking_Status =
"Approved")
```

4	Approved
---	----------

19. #Provide owner details

```
SELECT DISTINCT F_Name, L_Name, Email, Phone_No
FROM Member NATURAL JOIN Property
WHERE Member_ID = 2
```

F_Name	L_Name	Email	Phone_No
Vinyas	Harish	v.harish@queensu.ca	6477675831

20. #Place booking request

```
INSERT INTO `Booking` (`Property_ID`, `Booking_Start`, `Booking_Status`, `Member_ID`,
`Owner_ID`)
VALUES (2, '2016-08-17 16:30:00', 'Pending', 5, 2)
```

Booking_ID	Property_ID	Member_ID	Owner_ID	Booking_Start	Booking_Status
1	5	2	4	2016-03-11 12:00:00	Approved
2	4	5	3	2016-06-11 12:30:00	Rejected
3	2	3	2	2016-02-14 12:00:00	Rejected
4	4	5	3	2016-06-11 12:30:00	Approved
5	2	3	2	2016-02-14 12:00:00	Approved
6	2	5	2	2016-08-17 16:30:00	Pending

21. #List all bookings for one Member

```
SELECT Booking_ID, Member_ID, Property_ID, Street_No, Street_Name, City, Booking_Start,
Booking_Status
FROM Booking NATURAL JOIN Property
WHERE Member_ID = 3
GROUP BY Booking_Status
ORDER BY Booking_Start
```

Booking_ID	Member_ID	Property_ID	Street_No	Street_Name	City	Booking_Start	Booking_Status
3	3	2	12	Bay Street	Toronto	2016-02-14 12:00:00	Rejected
5	3	2	12	Bay Street	Toronto	2016-02-14 12:00:00	Approved

22. #Get details on one booking

```
SELECT DISTINCT Street_No, Street_Name, City, District_Name, Type, Price, Booking_Start,
Booking_Status, Owner_ID
FROM Booking NATURAL JOIN Property
WHERE Booking_ID = 1 AND Property_ID = 5
```

Street_No	Street_Name	City	District_Name	Type	Price	Booking_Start	Booking_Status	Owner_ID
1891	Quebec Avenue	Toronto	High Park	House	15	2016-03-11 12:00:00	Approved	4

23. #Comment on booking

```
INSERT INTO `Comment` (`Booking_ID`, `Member_ID`, `Rating`, `Comment_Text`)
VALUES (6, 5, 5, 'Never wanted to leave! Wonderful home.')
```

Booking_ID	Comment_Time	Member_ID	Rating	Comment_Text	Owner_Reply
1	2016-03-08 22:33:41	2	5	Great little spot very reasonably priced!	Glad to hear! Thank you for your visit!
4	2016-03-08 22:33:41	5	1	Worst experience I have ever had!	NULL
5	2016-03-08 22:43:25	3	4	Great place with an even better view!	Glad you liked it!
6	2016-03-08 23:46:20	5	5	Never wanted to leave! Wonderful home.	NULL

24. #Delete booking

```
DELETE FROM `Booking`
WHERE Booking_ID = 5
```

Booking_ID	Property_ID	Member_ID	Owner_ID	Booking_Start	Booking_Status
1	5	2	4	2016-03-11 12:00:00	Approved
2	4	5	3	2016-06-11 12:30:00	Pending
3	2	3	2	2016-02-14 12:00:00	Rejected
4	4	5	3	2016-06-11 12:30:00	Approved

ADMIN

25. #Delete member and associated listings

```
DELETE FROM `Member`
WHERE Member_ID = 4
```

Member_ID	F_Name	L_Name	Email	Phone_No	Grad_Year	Faculty	Degree_Type	Password
1	QBnB	Admin	admin@qbnb.ca	6135336666	2000	Computing	BComp	admin
2	Vinyas	Harish	v.harish@queensu.ca	6477675831	2017	Computing	BComp	vin
3	Zac	Baum	zac.baum@queensu.ca	4163203344	2017	Computing	BComp	zac
5	Laura	Brooks	laura.brooks@queensu.ca	4164359453	2016	Computing	BComp	cisc332ta

Property_ID	Owner_ID	Street_No	Street_Name	City	Country	Postal_Code	District_Name	Type	Price
1	1	12	Brock Street	Toronto	Canada	M9A 4X6	Entertainment District	Apartment	35
2	2	12	Bay Street	Toronto	Canada	M4A 9X6	Financial District	Loft	3000
3	1	50	Gerrard Street	Toronto	Canada	M9A 4X1	Danforth	Basement	50
4	3	41	Yonge Street	Toronto	Canada	M9A 4X9	North Toronto	Apartment	75

Property_ID	Feature_Name	Feature_Description
1	1 bathroom	NULL
2	Full kitchen	NULL
3	Full kitchen	NULL
3	Mahogany floors	NULL
4	2 bathrooms	NULL
4	2 kitchens	NULL
4	Study	NULL

Booking_ID	Property_ID	Member_ID	Owner_ID	Booking_Start	Booking_Status
2	4	5	3	2016-06-11 12:30:00	Pending
3	2	3	2	2016-02-14 12:00:00	Rejected
4	4	5	3	2016-06-11 12:30:00	Approved
5	2	3	2	2016-02-14 12:00:00	Approved

Booking_ID	Comment_Time	Member_ID	Rating	Comment_Text	Owner_Reply
4	2016-03-09 22:46:43	5	1	Worst experience I have ever had!	NULL
5	2016-03-09 22:46:43	3	4	Great place with an even better view!	NULL

```
26. #Delete property
DELETE FROM `Property`
WHERE Member_ID = 5
```

Member_ID	F_Name	L_Name	Email	Phone_No	Grad_Year	Faculty	Degree_Type	Password
1	QBnB	Admin	admin@qbnb.ca	6135336666	2000	Computing	BComp	admin
2	Vinyas	Harish	v.harish@queensu.ca	6477675831	2017	Computing	BComp	vin
3	Zac	Baum	zac.baum@queensu.ca	4163203344	2017	Computing	BComp	zac
4	Patrick	Martin	martin@cs.queensu.ca	6131113333	2000	Computing	BComp	cisc332prof
5	Laura	Brooks	laura.brooks@queensu.ca	4164359453	2016	Computing	BComp	cisc332ta

Property_ID	Owner_ID	Street_No	Street_Name	City	Country	Postal_Code	District_Name	Type	Price
1	1	12	Brock Street	Toronto	Canada	M9A 4X6	Entertainment District	Apartment	35
2	2	12	Bay Street	Toronto	Canada	M4A 9X6	Financial District	Loft	3000
3	1	50	Gerrard Street	Toronto	Canada	M9A 4X1	Danforth	Basement	50
4	3	41	Yonge Street	Toronto	Canada	M9A 4X9	North Toronto	Apartment	75

Property_ID	Feature_Name	Feature_Description
1	1 bathroom	NULL
2	Full kitchen	NULL
3	Full kitchen	NULL
3	Mahogany floors	NULL
4	2 bathrooms	NULL
4	2 kitchens	NULL
4	Study	NULL

Booking_ID	Property_ID	Member_ID	Owner_ID	Booking_Start	Booking_Status
2	4	5	3	2016-06-11 12:30:00	Pending
3	2	3	2	2016-02-14 12:00:00	Rejected
4	4	5	3	2016-06-11 12:30:00	Approved
5	2	3	2	2016-02-14 12:00:00	Approved

Booking_ID	Comment_Time	Member_ID	Rating	Comment_Text	Owner_Reply
4	2016-03-09 22:58:14	5	1	Worst experience I have ever had!	NULL
5	2016-03-09 22:58:14	3	4	Great place with an even better view!	NULL

```
27. #Summarize bookings and ratings per property
SELECT Booking_Start, Booking_Status, Avg(Rating)
FROM Booking NATURAL JOIN Comment
WHERE Property_ID = 4
GROUP BY Booking_Status
ORDER BY Booking_Start
```

Booking_Start	Booking_Status	Avg(Rating)
2016-06-11 12:30:00	Approved	1.0000

```

28. #Summarize bookings and ratings per supplier
SELECT Booking_Start, Booking_Status, Avg(Rating)
FROM Booking NATURAL JOIN Comment
WHERE Owner_ID = 4
GROUP BY Booking_Status
ORDER BY Booking_Start

```

Booking_Start	Booking_Status	Avg(Rating)
2016-03-11 12:00:00	Approved	5.0000

```

29. #Summarize bookings and ratings per customer
SELECT Booking_Start, Booking_Status
FROM Booking
WHERE Member_ID = 2
GROUP BY Booking_Status
ORDER BY Booking_Start

```

Booking_Start	Booking_Status
2016-03-11 12:00:00	Approved



User Interface Flow

Main Page

From the Main Page, there are a few options for the user to choose from:

- Browse through the main page and view additional information on QBnB
- Enter their email into the displayed textboxes and click “Sign Me Up!”
 - This option will take them to the Sign Up Page
- Select the “Already a Member” option to sign into QBnB
 - This option will take them to the Sign In Page

Users are not able to access this page if they are logged in, and it is the main page when accessing the website via the sites’ domain.

Sign Up Page

From the Sign Up Page, the user is able to enter their information:

- Upon entering valid information in all fields of the page, the user can select Sign Up, and will be taken to the User Dashboard
- If the user does not complete all fields, or tries to create an account with an email that is already in use, the sign up process will not complete and the user will be required to enter all required, or different information

From this page, the user is also able to access the Sign In Page by clicking “Already a Member”, or to return to the home page by clicking the QBnB logo.

Sign In Page

From the Sign In Page, the user is able to enter their account credentials:

- If they enter a valid email and password, the user is taken to the User Dashboard
- If they enter incorrect information, the user is prompted to try again with correct information

If the user does not want to sign in, they are able to return to the Main Page by clicking the QBnB logo.

User Dashboard

From the User Dashboard, there are a variety of options available to the user.

- The user may click on “Settings” and will given a drop-down menu where they can logout, update or delete their profile and if they are an administrator, they may access the Administrator Dashboard.
- The user may click on the QBnB logo, and the page will refresh.
- As a regular consumer:
 - The user may add a property and list its features (which will effectively change their account to that of a supplier),

- The user may search for properties by price, district, type and features,
 - They may view detailed information about listings (owner information, availability and reviews on the property),
 - They may make booking requests for one weeks worth of time,
 - They may view their bookings, should they have any,
 - They may cancel any bookings they may have,
 - They may view full booking details when their booking is in any state (pending, approved, rejected),
 - They may comment on and rate any booking they have had that has passed.
- As a supplier:
 - The user may do all a consumer is able to and;
 - The user may add additional properties and their features,
 - The user may remove any existing properties they have listed (if they remove their last property, they will become a regular consumer),
 - The user may update any information about their properties,
 - The user can approve or reject booking requests,
 - The user can view all their properties and reply to any comments posted on their properties.
 - As an administrator, the user may perform any task that is associated with the level that they occupy out of consumer or supplier (if they own property or properties, they are a supplier, if not they are a consumer).

Settings Page

From the Settings Page, the user is able to update any parts of their account information, can delete their account. Once they have entered the new information, they can select the confirm button, and may repeat this if desired. Once the user is done updating their information, they can be taken back to the User Dashboard by clicking “Go Back”. They can also delete their account by selecting “Delete Account”.

Administrator Dashboard

Within the Administrator Dashboard, administrators have several options and controls at their disposal. The Administrator Dashboard allows Admins to:

- Remove a given accommodation from the list of all properties (and all associated features, comments and bookings)
- Remove a given user from the list of all members (and all associated accommodations, comments and bookings)
- View all bookings made by a given user
- View all bookings, comments and ratings made on a given property
- View all bookings, comments and ratings made on all properties listed by a given supplier

Additionally, the Administrators are able to leave the Administrator Dashboard and return to the User Dashboard.