

Queries

Queries for Customers

Search **ForSale** or **ForRent** properties

```
SELECT Property.*, PostalCode.*, ForRent.price
FROM Property, PostalCode, ForRent
WHERE Property.postal_code = PostalCode.postal_code
AND ForRent.property_id = Property.property_id
AND ForRent.price BETWEEN 1 AND 5000
AND Property.sq_ft BETWEEN 1 AND 1500;
```

Get the contact info of the realtor of a specific property

```
SELECT R.*, O.* FROM Realtor R
INNER JOIN Property P ON R.realtor_id = P.realtor_id
INNER JOIN RealtyOffice O ON R.branch_id =
O.branch_id
WHERE P.property_id = '3';
```

Queries for Realtors

Get the info of a specific **ForSale** or **ForRent** properties

```
SELECT Property.*, PostalCode.*, ForRent.price
FROM Property, PostalCode, ForRent
WHERE Property.postal_code = PostalCode.postal_code
AND ForRent.property_id = Property.property_id
AND property_id = 3
```

Update the info of a specific property

```
UPDATE Property p
SET
p.sq_ft = 2300,
p.num_beds = 4,
p.num_baths = 5.5,
WHERE p.property_id = '3';
```

Get the Room info of a specific property

```
SELECT Room.*
FROM INNER JOIN Property
ON Property.property_id = Room.property_id
WHERE Property.property_id = '3';
```

Update the price in ForSale or ForRent table

```
UPDATE ForRent
SET price = 5000
WHERE property_id = '6';
```

Delete a property

```
DELETE FROM Property
WHERE Property.property_id = '2';
```

NOTE: This SQL statement deletes an entry from the Property table and

also deletes the corresponding entry from the **ForSale**, **ForRent**, **Sold**, or **Rented** table.

Get the contact messages and related customer info for a specific realtor

```
SELECT CCR.contact_message, C.name, C.email, C.phone,
CCR.date
FROM CustomerContactRealtor CCR, Customer C
WHERE CCR.customer_id = C.customer_id AND
CCR.realtor_id = '2';
```

Division Query

Find the customers who has contacted all the realtors

```
SELECT DISTINCT CCR1.customer_id
FROM CustomerContactRealtor CCR1
WHERE NOT EXISTS(
    SELECT CCR2.realtor_id
    FROM CustomerContactRealtor CCR2
    WHERE CCR2.realtor_id NOT IN
        (SELECT CCR3.realtor_id
         FROM CustomerContactRealtor CCR3
         WHERE CCR3.customer_id = CCR1.customer_id)
);
```

Aggregation Query

For each property, find the property of lowest rental price

```
SELECT R1.property_id, min_rent
```

```

FROM ForRent R1, (SELECT
                    P2.property_type AS type,
                    MIN(rent)         AS min_rent
                    FROM property P2, ForRent R2
                    WHERE P2.property_id =
R2.property_id
                    GROUP BY P2.property_type) AS
MinRents
WHERE MinRents.type = 'Apt/Condo' AND R1.rent =
MinRents.min_rent;

```

Nested Aggregation with Group-by

Find the highest/lowest average price of **ForSale** properties

Using **MAX()**

```

SELECT MAX(AvgPrices.avg_price) AS max_avg_price
FROM (SELECT AVG(S.price) AS avg_price
      FROM Property P, ForSale S, PostalCode PC
      WHERE S.property_id = P.property_id AND
P.postal_code = PC.postal_code
      GROUP BY PC.province) AS AvgPrices;

```

Using **MIN()**

```

SELECT MIN(AvgPrices.avg_price) AS min_avg_price
FROM (SELECT AVG(S.price) AS avg_price
      FROM Property P, ForSale S, PostalCode PC
      WHERE S.property_id = P.property_id AND
P.postal_code = PC.postal_code
      GROUP BY PC.province) AS AvgPrices;

```

Queries for Reports

Find the total number of sales this year

```
SELECT COUNT(property_id) AS num_properties
FROM RentalDatabase.Sold
WHERE YEAR(date_sold) = YEAR(CURDATE());
```

Find the average price of **ForSale** properties in each province

```
SELECT
    AvgPrices.avg_price,
    AvgPrices.province
FROM (SELECT
        AVG(S.price) AS avg_price,
        PC.province AS province
    FROM Property P, ForSale S, PostalCode PC
    WHERE S.property_id = P.property_id AND
    P.postal_code = PC.postal_code
    GROUP BY PC.province) AS AvgPrices;
```

Triggers

Check if the customer enters a valid email address

```
DELIMITER $$
CREATE TRIGGER email_before_insert
    BEFORE INSERT
    ON Customer
    FOR EACH ROW
```

```
BEGIN
  IF new.email NOT REGEXP '^[A-Z0-9._%-]+@[A-Z0-9.-
]+\. [A-Z]{2,4}$'
  THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Email is invalid!';
  END IF;
END $$
DELIMITER ;
```