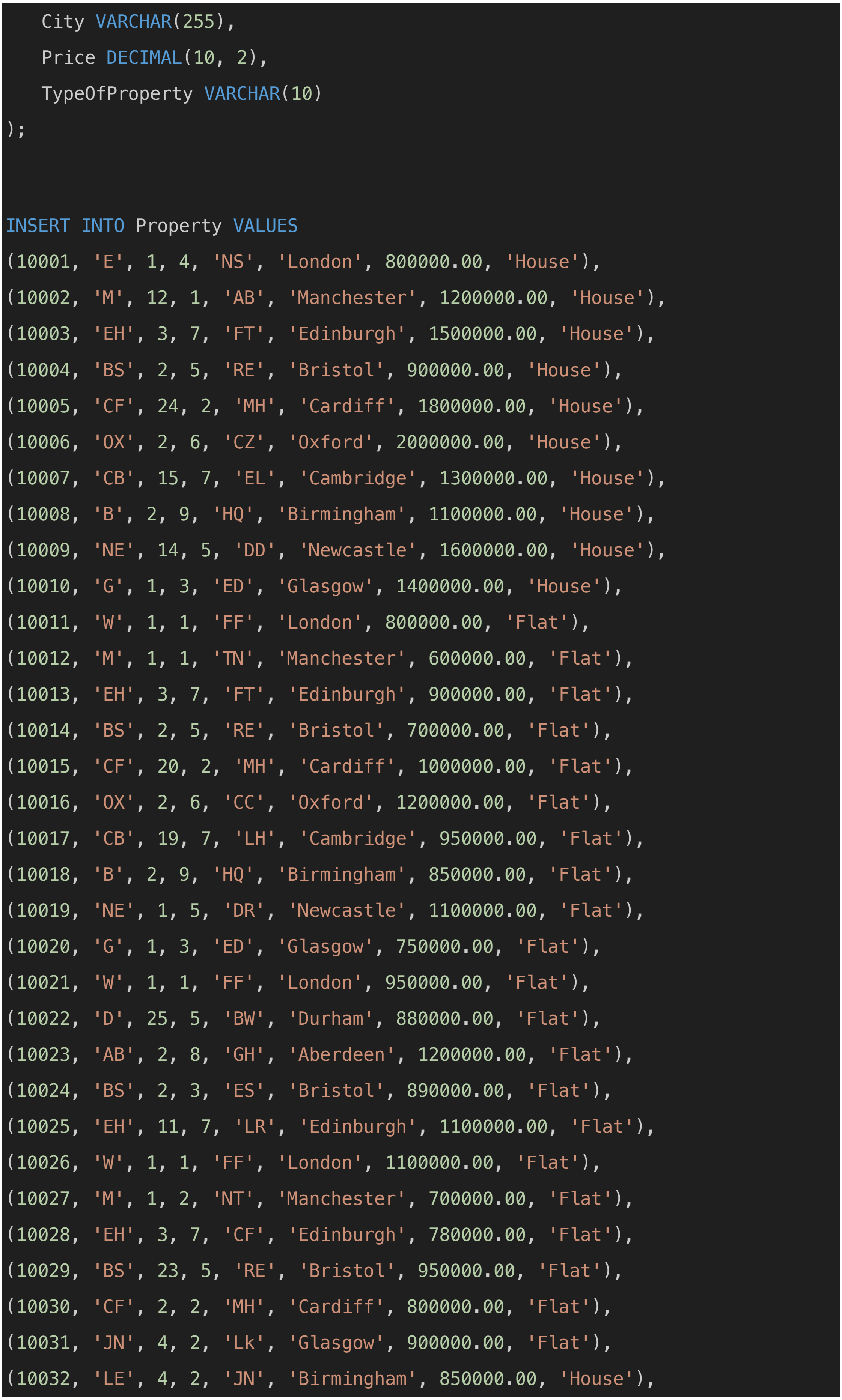
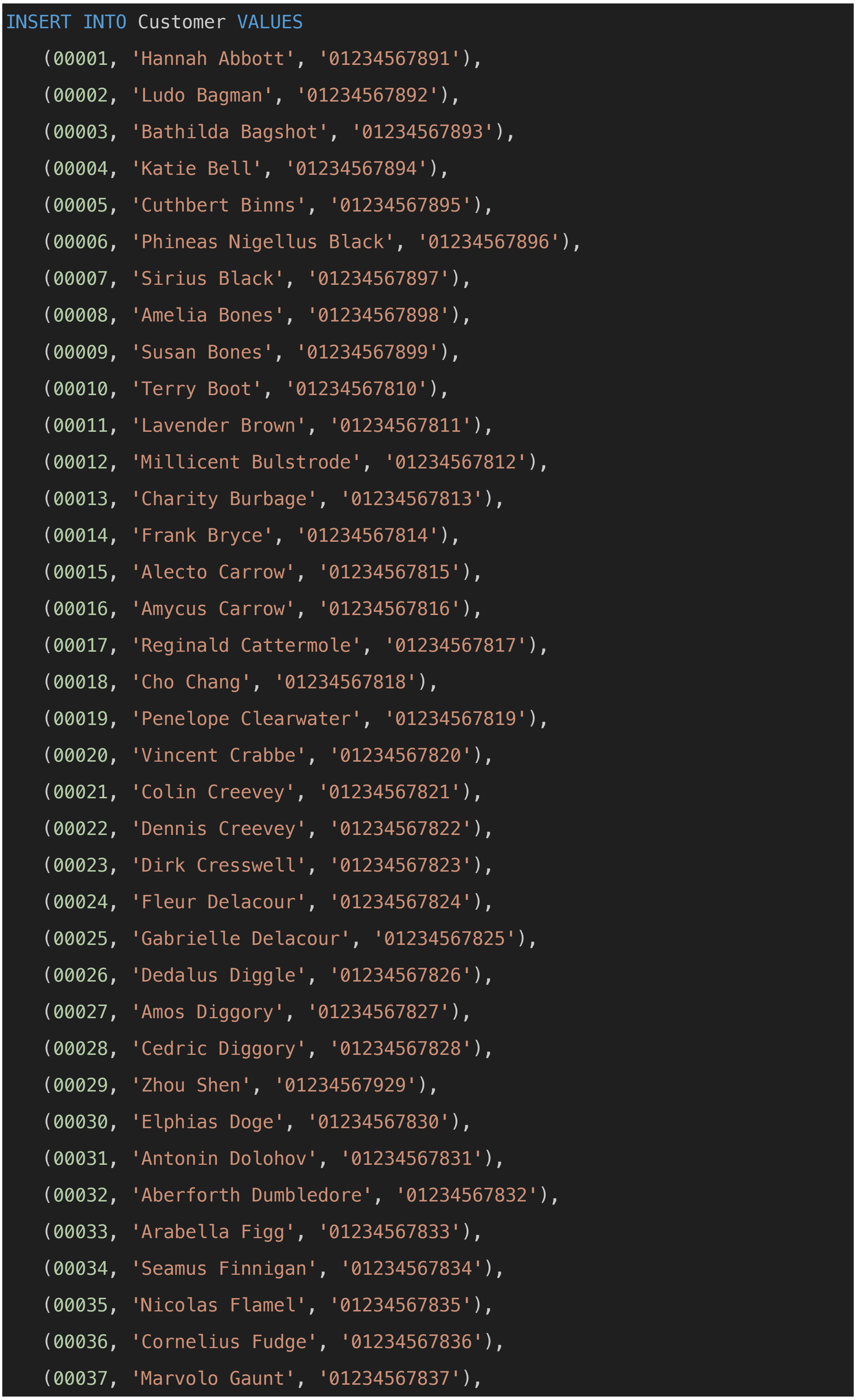
```sql
--Group 333
--Chia-Yi Chou - 220271772
--Poppy Isabella Howard - 220584535
--Ka Yi Cheng - 220566472

--Targeting MySQL

CREATE DATABASE  IF NOT EXISTS Agent;
USE Agent;
select 'Building demonstration tables.  Please wait.' as '';

DROP TABLE IF EXISTS Agency;
DROP TABLE IF EXISTS Office;
DROP TABLE IF EXISTS Property;
DROP TABLE IF EXISTS House;
DROP TABLE IF EXISTS Flat;
DROP TABLE IF EXISTS Customer;
DROP TABLE IF EXISTS forSale;
DROP TABLE IF EXISTS forRent;
DROP TABLE IF EXISTS Sold;

CREATE TABLE Agency (
   AgencyID INT PRIMARY KEY,
   AgencyName VARCHAR(255) NOT NULL,
   PhoneNumber VARCHAR(11)
);


INSERT INTO Agency (AgencyID, AgencyName, PhoneNumber) VALUES
(1, 'London Elite Estates', '07123456789'),
(2, 'Thamesview Properties', '07987654321'),
(3, 'Regency Residences', '07555123456'),
(4, 'Knightsbridge Homes', '07111222333'),
(5, 'Chelsea Charm Realty', '07444555666'),
(6, 'Westminster Realty Group', '07777888999'),
(7, 'Mayfair Mansion Management', '07888777666'),
(8, 'Kensington Key Properties', '07333222111'),
(9, 'Notting Hill Homesteads', '07555555555'),
(10, 'Camden City Homes', '07999888777');
```

```sql
CREATE TABLE Office (
    OfficeID INT PRIMARY KEY,
    AgencyID INT,
    Area VARCHAR(2),
    District INT,
    Sector INT,
    Unit VARCHAR(2),
    City VARCHAR(255),
    PhoneNumber VARCHAR(11),
    FOREIGN KEY (AgencyID) REFERENCES Agency(AgencyID)
);


INSERT INTO Office VALUES
(2001, 1, 'E', 1, 4, 'NS', 'London', '01234567890'),
(2002, 2, 'M', 1, 1, 'AB', 'Manchester', '02345678901'),
(2003, 3, 'EH', 3, 7, 'FD', 'Edinburgh', '03456789012'),
(2004, 4, 'BS', 2, 5, 'RE', 'Bristol', '04567890123'),
(2005, 5, 'CF', 2, 2, 'MH', 'Cardiff', '05678901234'),
(2006, 6, 'OX', 2, 6, 'CT', 'Oxford', '06789012345'),
(2007, 7, 'CB', 1, 7, 'LA', 'Cambridge', '07890123456'),
(2008, 8, 'B', 2, 9, 'HQ', 'Birmingham', '08901234567'),
(2009, 9, 'NE', 1, 5, 'DG', 'Newcastle', '09012345678'),
(2010, 10, 'G', 1, 3, 'ED', 'Glasgow', '09123456789'),
(2011, 1, 'W', 1, 1, 'FF', 'London', '10234567890'),
(2012, 2, 'M', 1, 1, 'TN', 'Manchester', '11234567890'),
(2013, 3, 'EH', 3, 7, 'FK', 'Edinburgh', '12234567890'),
(2014, 4, 'BS', 2, 5, 'RE', 'Bristol', '13234567890'),
(2015, 5, 'CF', 2, 2, 'MH', 'Cardiff', '14234567890');


CREATE TABLE Property (
    PropertyID INT PRIMARY KEY,
    Area VARCHAR(2),
    District INT,
    Sector INT,
    Unit VARCHAR(2),
```

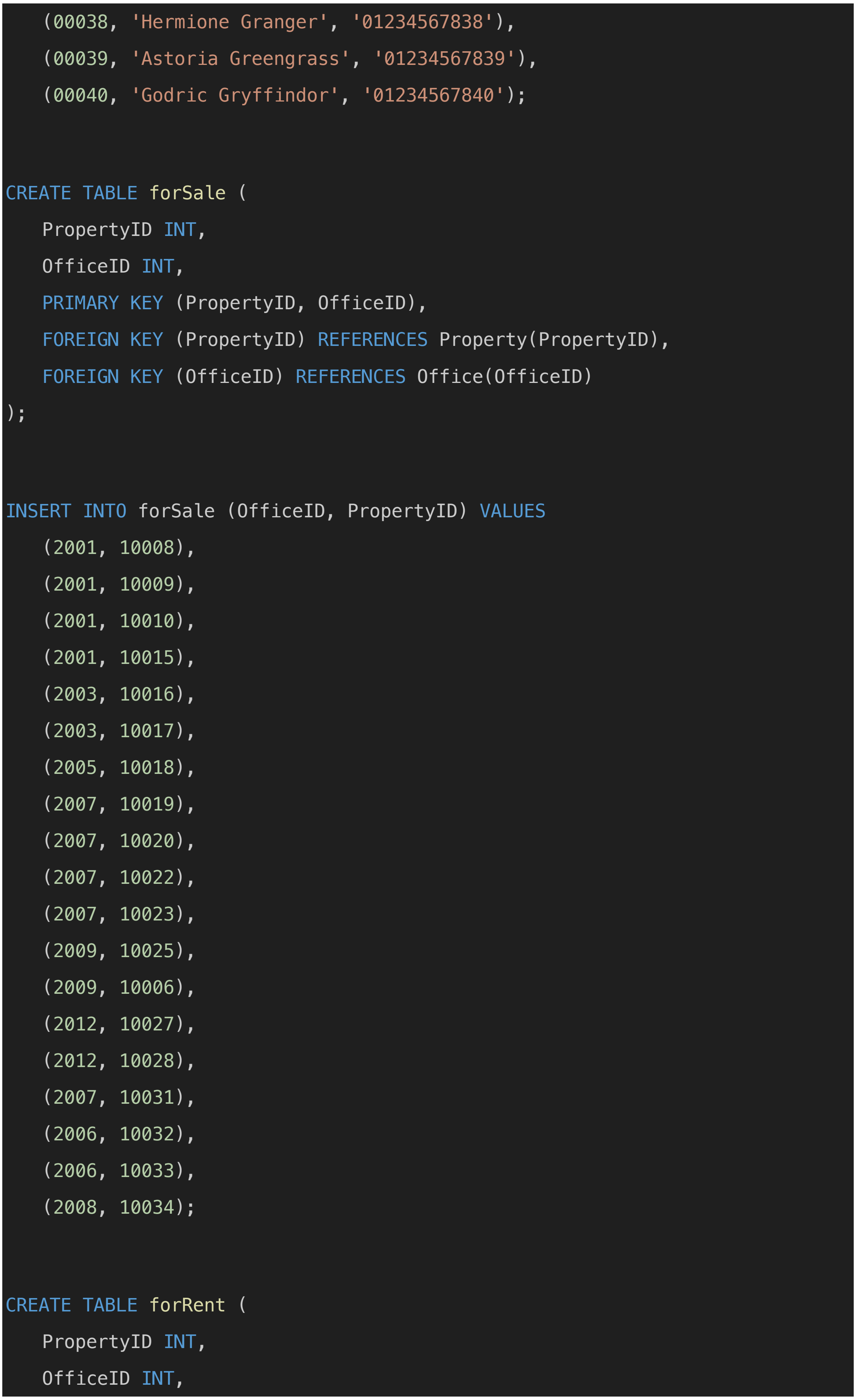


```
    City VARCHAR(255),
    Price DECIMAL(10, 2),
    TypeOfProperty VARCHAR(10)
);


INSERT INTO Property VALUES
(10001, 'E', 1, 4, 'NS', 'London', 800000.00, 'House'),
(10002, 'M', 12, 1, 'AB', 'Manchester', 1200000.00, 'House'),
(10003, 'EH', 3, 7, 'FT', 'Edinburgh', 1500000.00, 'House'),
(10004, 'BS', 2, 5, 'RE', 'Bristol', 900000.00, 'House'),
(10005, 'CF', 24, 2, 'MH', 'Cardiff', 1800000.00, 'House'),
(10006, 'OX', 2, 6, 'CZ', 'Oxford', 2000000.00, 'House'),
(10007, 'CB', 15, 7, 'EL', 'Cambridge', 1300000.00, 'House'),
(10008, 'B', 2, 9, 'HQ', 'Birmingham', 1100000.00, 'House'),
(10009, 'NE', 14, 5, 'DD', 'Newcastle', 1600000.00, 'House'),
(10010, 'G', 1, 3, 'ED', 'Glasgow', 1400000.00, 'House'),
(10011, 'W', 1, 1, 'FF', 'London', 800000.00, 'Flat'),
(10012, 'M', 1, 1, 'TN', 'Manchester', 600000.00, 'Flat'),
(10013, 'EH', 3, 7, 'FT', 'Edinburgh', 900000.00, 'Flat'),
(10014, 'BS', 2, 5, 'RE', 'Bristol', 700000.00, 'Flat'),
(10015, 'CF', 20, 2, 'MH', 'Cardiff', 1000000.00, 'Flat'),
(10016, 'OX', 2, 6, 'CC', 'Oxford', 1200000.00, 'Flat'),
(10017, 'CB', 19, 7, 'LH', 'Cambridge', 950000.00, 'Flat'),
(10018, 'B', 2, 9, 'HQ', 'Birmingham', 850000.00, 'Flat'),
(10019, 'NE', 1, 5, 'DR', 'Newcastle', 1100000.00, 'Flat'),
(10020, 'G', 1, 3, 'ED', 'Glasgow', 750000.00, 'Flat'),
(10021, 'W', 1, 1, 'FF', 'London', 950000.00, 'Flat'),
(10022, 'D', 25, 5, 'BW', 'Durham', 880000.00, 'Flat'),
(10023, 'AB', 2, 8, 'GH', 'Aberdeen', 1200000.00, 'Flat'),
(10024, 'BS', 2, 3, 'ES', 'Bristol', 890000.00, 'Flat'),
(10025, 'EH', 11, 7, 'LR', 'Edinburgh', 1100000.00, 'Flat'),
(10026, 'W', 1, 1, 'FF', 'London', 1100000.00, 'Flat'),
(10027, 'M', 1, 2, 'NT', 'Manchester', 700000.00, 'Flat'),
(10028, 'EH', 3, 7, 'CF', 'Edinburgh', 780000.00, 'Flat'),
(10029, 'BS', 23, 5, 'RE', 'Bristol', 950000.00, 'Flat'),
(10030, 'CF', 2, 2, 'MH', 'Cardiff', 800000.00, 'Flat'),
(10031, 'JN', 4, 2, 'Lk', 'Glasgow', 900000.00, 'Flat'),
(10032, 'LE', 4, 2, 'JN', 'Birmingham', 850000.00, 'House'),
(10033, 'E', 1, 1, 'EE', 'Newcastle', 750000.00, 'Flat'),
(10034, 'J', 4, 2, 'KK', 'Oxford', 600000.00, 'House');


--Generalized from property table
CREATE TABLE House (
    PropertyID INT PRIMARY KEY,
    NumBedrooms INT,
    NumFloors INT,
    HasBackyard VARCHAR(3),
    GarageCapacity INT,
    Description TEXT,
    FOREIGN KEY (PropertyID) REFERENCES Property(PropertyID)
);


INSERT INTO House VALUES
(10001, 4, 2, 'Yes', 2, 'Modern house with spacious rooms'),
(10002, 3, 1, 'No', 1, 'Cozy cottage-style house'),
(10003, 5, 3, 'Yes', 2, 'Luxury mansion with a beautiful backyard'),
(10004, 4, 2, 'No', 2, 'Classic house with vintage charm'),
(10005, 3, 1, 'Yes', 1, 'Contemporary house with city views'),
(10006, 4, 2, 'Yes', 2, 'Family-friendly house with play area'),
(10007, 3, 1, 'No', 1, 'Rustic house with natural surroundings'),
(10008, 5, 3, 'Yes', 2, 'Elegant estate with a pool'),
(10009, 4, 2, 'No', 2, 'Artistic house with unique design'),
(10010, 3, 1, 'Yes', 1, 'Suburban retreat with a garden'),
(10032, 4, 3, 'Yes', 2, 'Complete with hardwood floors'),
(10034, 4, 3, 'No', 1, 'A charming flat with a warm ambiance with a
comfortable bedroom');


--Generalized from property table
CREATE TABLE Flat (
    PropertyID INT PRIMARY KEY,
    NumBedrooms INT,
    FloorNum INT,
    HasElevator VARCHAR(3),
    HasBalcony VARCHAR(3),
```

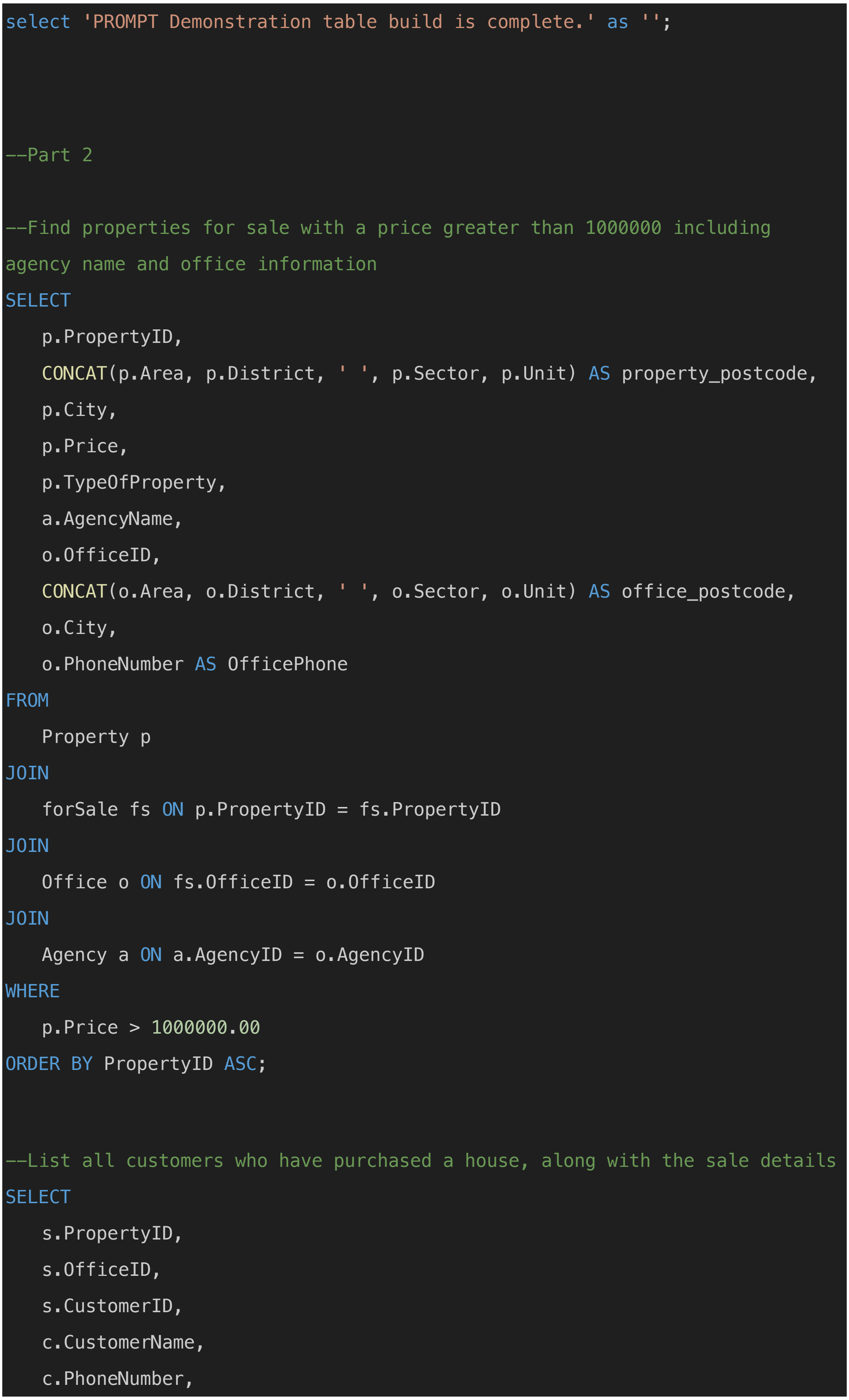
```sql
    Description TEXT,
    FOREIGN KEY (PropertyID) REFERENCES Property(PropertyID)
);


INSERT INTO Flat VALUES
(10011, 2, 5, 'Yes', 'No', 'Modern flat with city view'),
(10012, 1, 2, 'No', 'Yes', 'Cozy apartment with a balcony'),
(10013, 3, 8, 'Yes', 'Yes', 'Luxury penthouse on the top floor'),
(10014, 2, 3, 'No', 'No', 'Simple flat with essential amenities'),
(10015, 1, 6, 'Yes', 'Yes', 'Elegant flat with a spacious balcony'),
(10016, 3, 4, 'Yes', 'No', 'Family-friendly apartment on a mid-level floor'),
(10017, 2, 7, 'No', 'Yes', 'Chic loft-style flat with a balcony'),
(10018, 4, 12, 'Yes', 'Yes', 'Exclusive penthouse with panoramic views'),
(10019, 3, 5, 'No', 'No', 'Comfortable flat in a residential building'),
(10020, 2, 9, 'Yes', 'Yes', 'Urban living with a balcony and city vibes'),
(10021, 1, 3, 'Yes', 'No', 'Stylish studio with elevator access'),
(10022, 2, 10, 'No', 'Yes', 'Spacious flat with a scenic balcony'),
(10023, 3, 6, 'Yes', 'Yes', 'Modern apartment with high-rise living'),
(10024, 1, 4, 'No', 'No', 'Quaint flat with vintage charm'),
(10025, 2, 8, 'Yes', 'Yes', 'Contemporary flat with a private balcony'),
(10026, 3, 5, 'Yes', 'No', 'Executive apartment with cityscape views'),
(10027, 2, 7, 'No', 'Yes', 'Sleek and modern flat on an upper floor'),
(10028, 4, 11, 'Yes', 'Yes', 'Penthouse living with luxury amenities'),
(10029, 3, 6, 'No', 'No', 'Classic flat with a homely atmosphere'),
(10030, 2, 9, 'Yes', 'Yes', 'Urban retreat with a spacious balcony'),
(10031, 4, 3, 'No', 'No', 'Experience modern elegance in this newly renovated flat'),
(10033, 2, 2, 'Yes', 'No', 'This cozy flat features a spacious living room with modern furnishings');


CREATE TABLE Customer (
    CustomerID INT PRIMARY KEY,
    CustomerName VARCHAR(255),
    PhoneNumber VARCHAR(11)
);
```



```
INSERT INTO Customer VALUES
    (00001, 'Hannah Abbott', '01234567891'),
    (00002, 'Ludo Bagman', '01234567892'),
    (00003, 'Bathilda Bagshot', '01234567893'),
    (00004, 'Katie Bell', '01234567894'),
    (00005, 'Cuthbert Binns', '01234567895'),
    (00006, 'Phineas Nigellus Black', '01234567896'),
    (00007, 'Sirius Black', '01234567897'),
    (00008, 'Amelia Bones', '01234567898'),
    (00009, 'Susan Bones', '01234567899'),
    (00010, 'Terry Boot', '01234567810'),
    (00011, 'Lavender Brown', '01234567811'),
    (00012, 'Millicent Bulstrode', '01234567812'),
    (00013, 'Charity Burbage', '01234567813'),
    (00014, 'Frank Bryce', '01234567814'),
    (00015, 'Alecto Carrow', '01234567815'),
    (00016, 'Amycus Carrow', '01234567816'),
    (00017, 'Reginald Cattermole', '01234567817'),
    (00018, 'Cho Chang', '01234567818'),
    (00019, 'Penelope Clearwater', '01234567819'),
    (00020, 'Vincent Crabbe', '01234567820'),
    (00021, 'Colin Creevey', '01234567821'),
    (00022, 'Dennis Creevey', '01234567822'),
    (00023, 'Dirk Cresswell', '01234567823'),
    (00024, 'Fleur Delacour', '01234567824'),
    (00025, 'Gabrielle Delacour', '01234567825'),
    (00026, 'Dedalus Diggle', '01234567826'),
    (00027, 'Amos Diggory', '01234567827'),
    (00028, 'Cedric Diggory', '01234567828'),
    (00029, 'Zhou Shen', '01234567929'),
    (00030, 'Elphias Doge', '01234567830'),
    (00031, 'Antonin Dolohov', '01234567831'),
    (00032, 'Aberforth Dumbledore', '01234567832'),
    (00033, 'Arabella Figg', '01234567833'),
    (00034, 'Seamus Finnigan', '01234567834'),
    (00035, 'Nicolas Flamel', '01234567835'),
    (00036, 'Cornelius Fudge', '01234567836'),
    (00037, 'Marvolo Gaunt', '01234567837'),
```

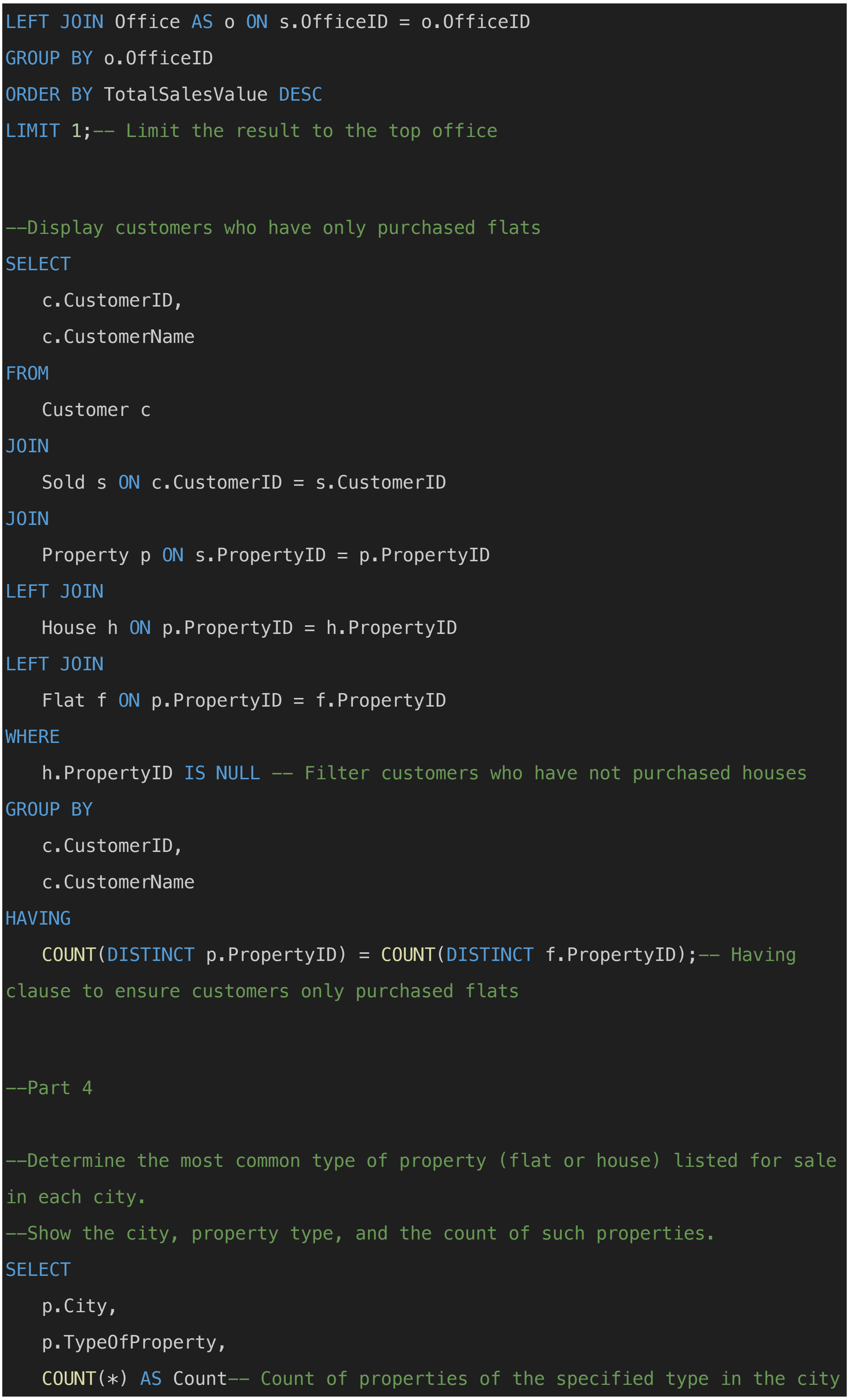


```sql
    (00038, 'Hermione Granger', '01234567838'),
    (00039, 'Astoria Greengrass', '01234567839'),
    (00040, 'Godric Gryffindor', '01234567840');


CREATE TABLE forSale (
    PropertyID INT,
    OfficeID INT,
    PRIMARY KEY (PropertyID, OfficeID),
    FOREIGN KEY (PropertyID) REFERENCES Property(PropertyID),
    FOREIGN KEY (OfficeID) REFERENCES Office(OfficeID)
);


INSERT INTO forSale (OfficeID, PropertyID) VALUES
    (2001, 10008),
    (2001, 10009),
    (2001, 10010),
    (2001, 10015),
    (2003, 10016),
    (2003, 10017),
    (2005, 10018),
    (2007, 10019),
    (2007, 10020),
    (2007, 10022),
    (2007, 10023),
    (2009, 10025),
    (2009, 10006),
    (2012, 10027),
    (2012, 10028),
    (2007, 10031),
    (2006, 10032),
    (2006, 10033),
    (2008, 10034);


CREATE TABLE forRent (
    PropertyID INT,
    OfficeID INT,
    PRIMARY KEY (PropertyID, OfficeID),
    FOREIGN KEY (PropertyID) REFERENCES Property(PropertyID),
    FOREIGN KEY (OfficeID) REFERENCES Office(OfficeID)
);


INSERT INTO forRent (OfficeID, PropertyID) VALUES
    (2002, 10002),
    (2002, 10003),
    (2002, 10004),
    (2004, 10005),
    (2006, 10011),
    (2008, 10012),
    (2008, 10013),
    (2010, 10021),
    (2010, 10024),
    (2011, 10029);


CREATE TABLE Sold (
    PropertyID INT,
    OfficeID INT,
    CustomerID INT,
    PRIMARY KEY (PropertyID, OfficeID, CustomerID),
    FOREIGN KEY (PropertyID) REFERENCES Property(PropertyID),
    FOREIGN KEY (OfficeID) REFERENCES Office(OfficeID),
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
);


INSERT INTO Sold (OfficeID, PropertyID, CustomerID) VALUES
    (2013, 10001, 38),
    (2013, 10026, 32),
    (2014, 10007, 12),
    (2015, 10014, 29),
    (2015, 10030, 40);

COMMIT;
```



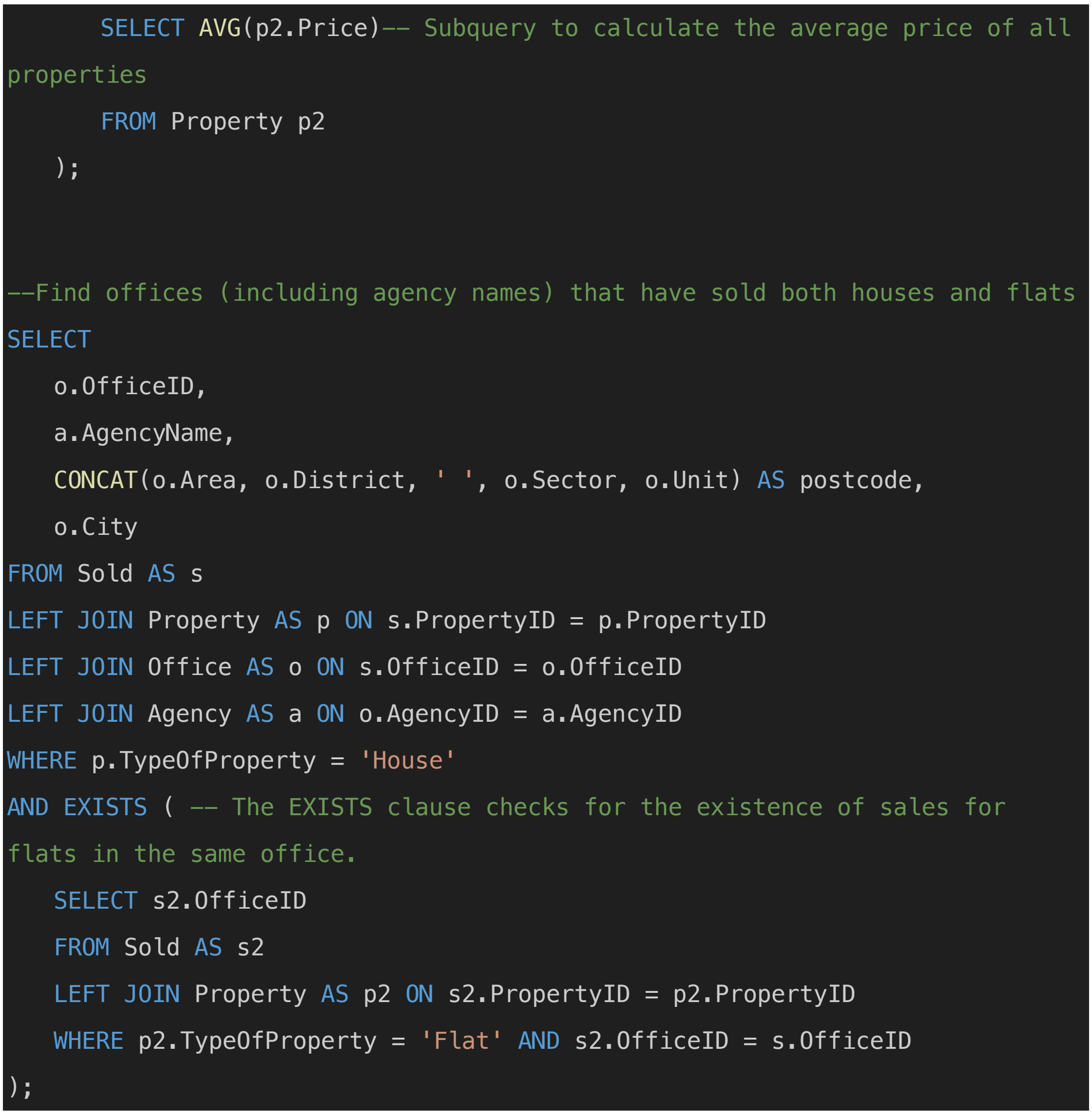
```
select 'PROMPT Demonstration table build is complete.' as '';



--Part 2

--Find properties for sale with a price greater than 1000000 including agency name and office information
SELECT
    p.PropertyID,
    CONCAT(p.Area, p.District, ' ', p.Sector, p.Unit) AS property_postcode,
    p.City,
    p.Price,
    p.TypeOfProperty,
    a.AgencyName,
    o.OfficeID,
    CONCAT(o.Area, o.District, ' ', o.Sector, o.Unit) AS office_postcode,
    o.City,
    o.PhoneNumber AS OfficePhone
FROM
    Property p
JOIN
    forSale fs ON p.PropertyID = fs.PropertyID
JOIN
    Office o ON fs.OfficeID = o.OfficeID
JOIN
    Agency a ON a.AgencyID = o.AgencyID
WHERE
    p.Price > 1000000.00
ORDER BY PropertyID ASC;


--List all customers who have purchased a house, along with the sale details
SELECT
    s.PropertyID,
    s.OfficeID,
    s.CustomerID,
    c.CustomerName,
    c.PhoneNumber,
```

```sql
    CONCAT(p.Area, p.District, ' ', p.Sector, p.Unit) AS postcode,
    p.City,
    p.Price,
    p.TypeOfProperty,
    h.NumBedrooms,
    h.NumFloors
FROM Sold AS s
JOIN Property AS p ON s.PropertyID = p.PropertyID
JOIN Customer AS c ON s.CustomerID = c.CustomerID
JOIN House AS h ON h.PropertyID = p.PropertyID
WHERE p.TypeOfProperty = 'House';


--Part 3

--Which real estate agencies have sold properties with a total value exceeding 1750000?
--List these agencies along with the total value of properties they've sold
SELECT
    a.AgencyID,
    a.AgencyName,
    SUM(p.Price) AS TotalSales
FROM Agency a
JOIN Office o ON a.AgencyID = o.AgencyID
JOIN Sold s ON o.OfficeID = s.OfficeID
JOIN Property p ON s.PropertyID = p.PropertyID
GROUP BY a.AgencyID, a.AgencyName
HAVING SUM(p.Price) > 1750000;


--Find the top one offices in terms of total sales value of properties.
--Display the officeID along with the total number of properties sold and the total sales value
SELECT
    o.OfficeID,
    COUNT(s.PropertyID) AS TotalPropertiesSold,
    SUM(p.Price) AS TotalSalesValue
FROM Sold AS s
LEFT JOIN Property AS p ON s.PropertyID = p.PropertyID
```



```sql
LEFT JOIN Office AS o ON s.OfficeID = o.OfficeID
GROUP BY o.OfficeID
ORDER BY TotalSalesValue DESC
LIMIT 1;-- Limit the result to the top office


--Display customers who have only purchased flats
SELECT
    c.CustomerID,
    c.CustomerName
FROM
    Customer c
JOIN
    Sold s ON c.CustomerID = s.CustomerID
JOIN
    Property p ON s.PropertyID = p.PropertyID
LEFT JOIN
    House h ON p.PropertyID = h.PropertyID
LEFT JOIN
    Flat f ON p.PropertyID = f.PropertyID
WHERE
    h.PropertyID IS NULL -- Filter customers who have not purchased houses
GROUP BY
    c.CustomerID,
    c.CustomerName
HAVING
    COUNT(DISTINCT p.PropertyID) = COUNT(DISTINCT f.PropertyID);-- Having clause to ensure customers only purchased flats


--Part 4

--Determine the most common type of property (flat or house) listed for sale in each city.
--Show the city, property type, and the count of such properties.
SELECT
    p.City,
    p.TypeOfProperty,
    COUNT(*) AS Count-- Count of properties of the specified type in the city
```

```sql
FROM
    forSale AS s
JOIN
    Property AS p ON s.PropertyID = p.PropertyID
GROUP BY
    p.City, p.TypeOfProperty
HAVING
    COUNT(*) = (
        SELECT
            COUNT(*)-- Count of properties of each type in the city
        FROM
            forSale AS s2-- forSale table (inner query)
        JOIN
            Property AS p2 ON s2.PropertyID = p2.PropertyID-- Join with Property table based on PropertyID from forSale (inner query)
        WHERE
            p.City = p2.City
        GROUP BY
            p2.TypeOfProperty
        ORDER BY
            COUNT(*) DESC
        LIMIT 1-- Limit to the top type of property in the city
    )
ORDER BY
    p.City, Count DESC;


--List all properties for rent that are priced above the average price of all properties.
SELECT
    p.PropertyID,
    p.District,
    p.Price
FROM
    Property p
JOIN
    forRent fr ON p.PropertyID = fr.PropertyID
WHERE
    p.Price > (
```



```sql
        SELECT AVG(p2.Price)-- Subquery to calculate the average price of all properties
        FROM Property p2
    );


--Find offices (including agency names) that have sold both houses and flats
SELECT
    o.OfficeID,
    a.AgencyName,
    CONCAT(o.Area, o.District, ' ', o.Sector, o.Unit) AS postcode,
    o.City
FROM Sold AS s
LEFT JOIN Property AS p ON s.PropertyID = p.PropertyID
LEFT JOIN Office AS o ON s.OfficeID = o.OfficeID
LEFT JOIN Agency AS a ON o.AgencyID = a.AgencyID
WHERE p.TypeOfProperty = 'House'
AND EXISTS ( -- The EXISTS clause checks for the existence of sales for flats in the same office.
    SELECT s2.OfficeID
    FROM Sold AS s2
    LEFT JOIN Property AS p2 ON s2.PropertyID = p2.PropertyID
    WHERE p2.TypeOfProperty = 'Flat' AND s2.OfficeID = s.OfficeID
);
```