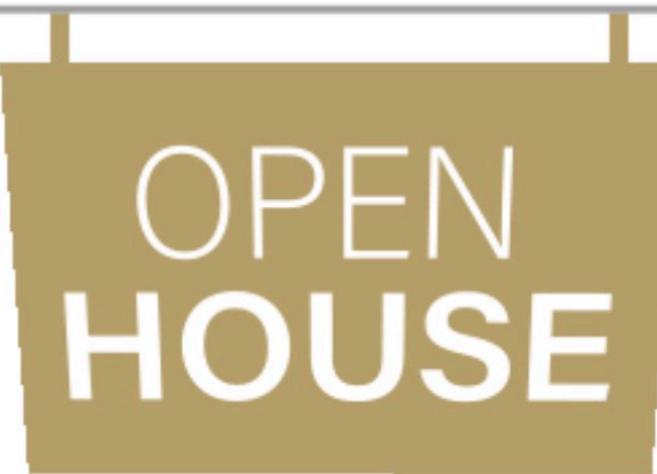




BlueFin: A Realty Application



By Matthew A., Renee C., Daniel S.



The idea behind BlueFin

BlueFin is a real estate application that helps property management companies manage realtor performance, stay up to date on housing market trends, and keep consumers happy. It includes several factors, including property overviews, transaction histories, and a detailed listing of homeowners and those searching for housing. It focuses on property in Massachusetts and detailing the complex relations that exist within this sphere.



Boston



Our Database

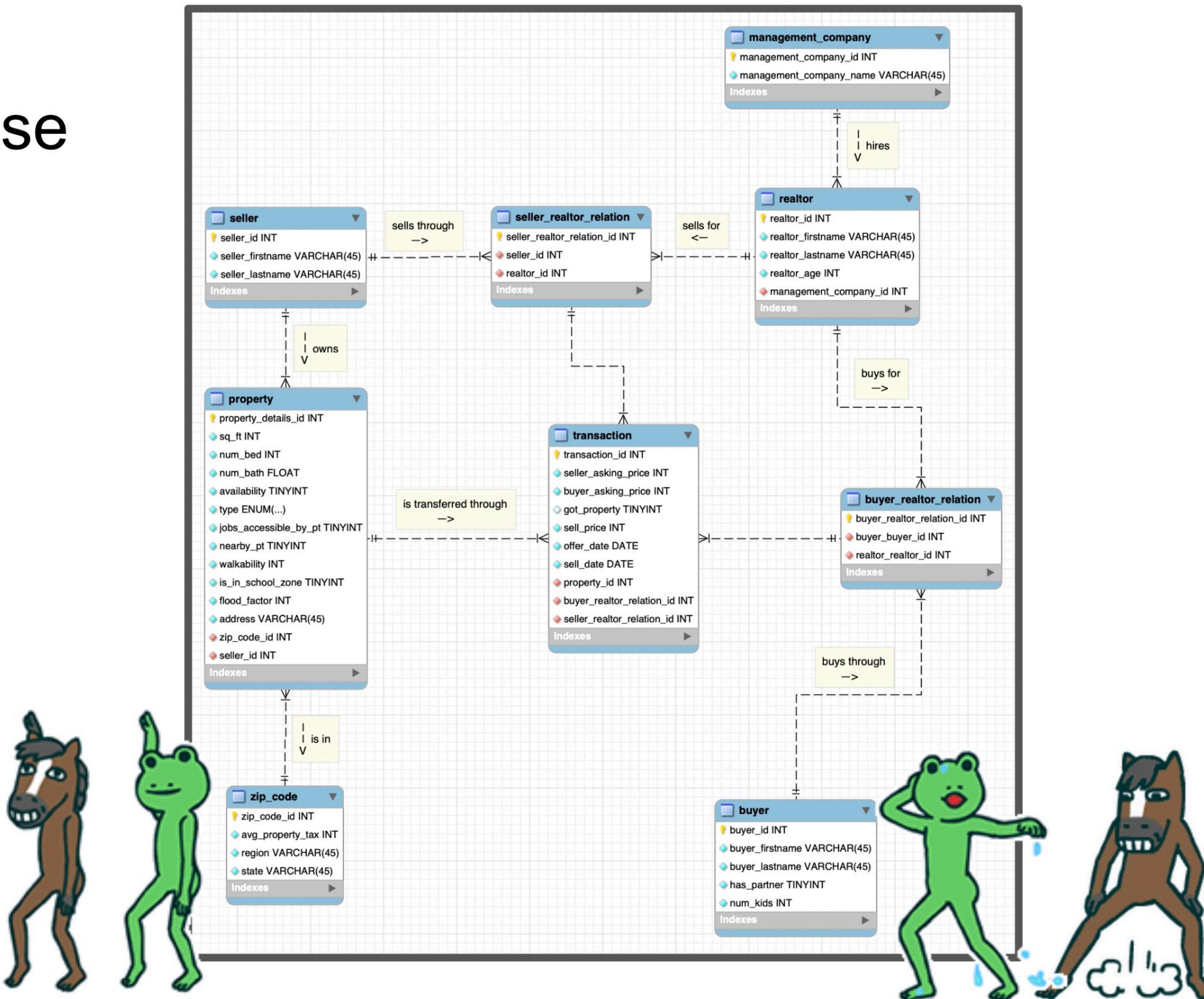


Table Creation:

```
1  -- ===== Create Database, Tables, and Attributes =====
2 • DROP DATABASE IF EXISTS realty;
3 • CREATE DATABASE IF NOT EXISTS realty;
4
5 • USE realty;
6
7 • DROP TABLE IF EXISTS management_company;
8 • ⊖ CREATE TABLE management_company(
9     management_company_id INT PRIMARY KEY AUTO_INCREMENT,
10    management_company_name VARCHAR(45));
11
12
13 • DROP TABLE IF EXISTS realtor;
14 • ⊖ CREATE TABLE realtor(
15     realtor_id INT PRIMARY KEY AUTO_INCREMENT,
16     realtor_firstname VARCHAR (45),
17     realtor_lastname VARCHAR (45),
18     realtor_age VARCHAR (45),
19     management_company_id INT,
20     FOREIGN KEY (management_company_id) REFERENCES management_company(management_company_id));
21
22 • DROP TABLE IF EXISTS seller;
23 • ⊖ CREATE TABLE seller(
24     seller_id INT PRIMARY KEY AUTO_INCREMENT,
25     seller_firstname VARCHAR (45),
26     seller_lastname VARCHAR (45));
27
28 • DROP TABLE IF EXISTS buyer;
29 • ⊖ CREATE TABLE buyer(
30     buyer_id INT PRIMARY KEY AUTO_INCREMENT,
31     has_partner TINYINT,
32     num_kids INT,
33     buyer_firstname VARCHAR (45),
34     buyer_lastname VARCHAR (45));
35
```

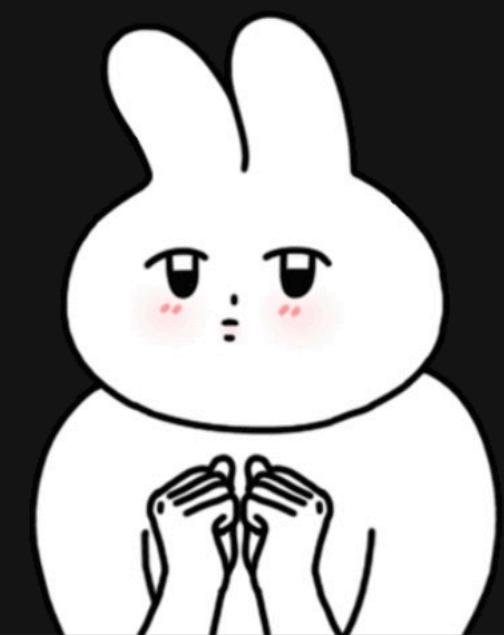


Table Creation:

```
5 • DROP TABLE IF EXISTS zip_code;
1   • ⊖ CREATE TABLE zip_code(
2   • | 3   zip_code_id INT PRIMARY KEY,
3   • | 4   avg_property_tax INT,
4   • | 5   region VARCHAR(45),
5   • | 6   state VARCHAR(45));
5   • |
6   • |
7   • | 3 • DROP TABLE IF EXISTS seller_realtor_relation;
7   • | 4 • ⊖ CREATE TABLE seller_realtor_relation(
8   • | | 5   seller_realtor_relation_id INT PRIMARY KEY AUTO_INCREMENT,
9   • | | 6   seller_id INT,
10  • | | 7   realtor_id INT,
11  • | | 8   FOREIGN KEY (seller_id) REFERENCES seller(seller_id),
12  • | | 9   FOREIGN KEY (realtor_id) REFERENCES realtor(realtor_id));
12  • |
13  • | 1 • DROP TABLE IF EXISTS buyer_realtor_relation;
14  • | 2 • ⊖ CREATE TABLE buyer_realtor_relation(
15  • | | 3   buyer_realtor_relation_id INT PRIMARY KEY AUTO_INCREMENT,
16  • | | 4   buyer_id INT,
17  • | | 5   realtor_id INT,
18  • | | 6   FOREIGN KEY (buyer_id) REFERENCES buyer(buyer_id),
19  • | | 7   FOREIGN KEY (realtor_id) REFERENCES realtor(realtor_id));
19  • |
20  • | 9 • DROP TABLE IF EXISTS property;
21  • | 0 • ⊖ CREATE TABLE property(
22  • | | 1   property_id INT PRIMARY KEY AUTO_INCREMENT,
23  • | | 2   sq_ft INT,
24  • | | 3   num_bed INT,
25  • | | 4   num_bath FLOAT,
26  • | | 5   availability TINYINT,
27  • | | 6   type ENUM('apartment', 'house', 'studio', 'condo'),
28  • | | 7   jobs_accessible_by_pt TINYINT,
29  • | | 8   nearby_pt TINYINT,
30  • | | 9   walkability INT,
31  • | | 0   in_school_zone TINYINT,
32  • | | 1   flood_factor INT,
33  • | | 2   address VARCHAR (45),
34  • | | 3   zip_code_id INT,
34  • | | 4   seller_id INT,
```

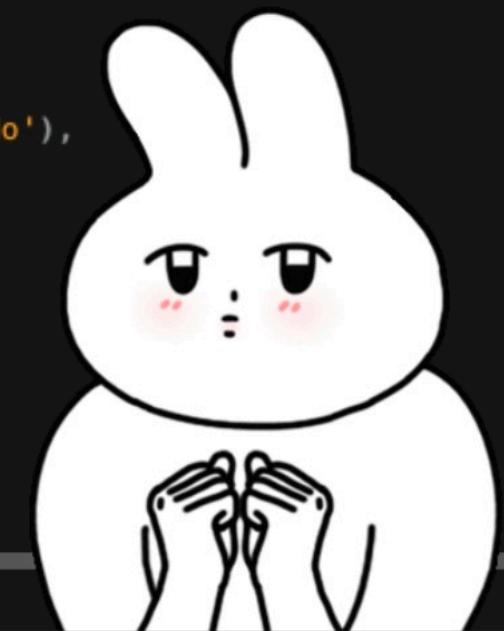


Table Creation:

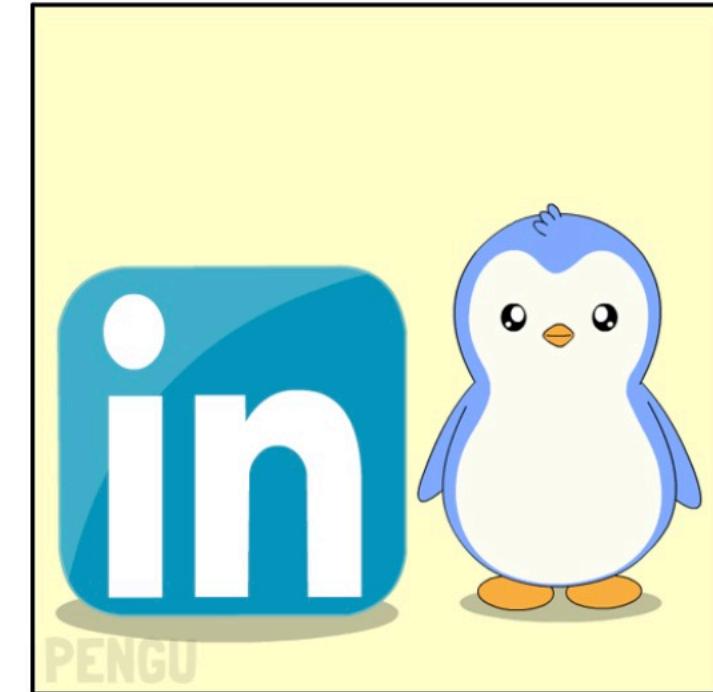
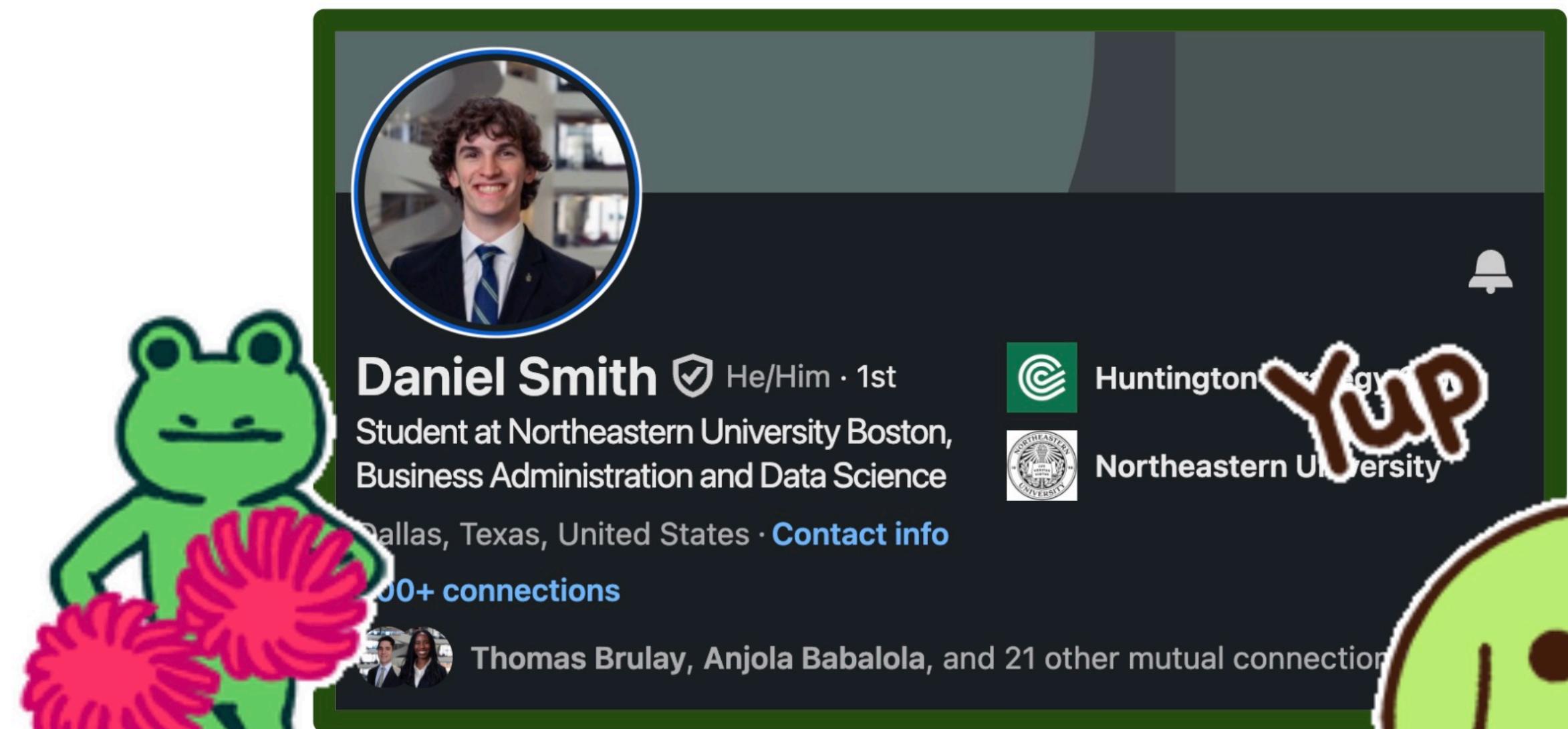
```
5 • DROP TABLE IF EXISTS zip_code;
1   • 7 • ⊕ CREATE TABLE zip_code(
2 • | 8 | zip_code_id INT PRIMARY KEY,
3 • | 9 | avg_property_tax INT,
```

```
• DROP TABLE IF EXISTS property;
• ⊕ CREATE TABLE property(
  •
  • DROP TABLE IF EXISTS transaction;
  • ⊖ CREATE TABLE transaction(
    transaction_id INT PRIMARY KEY AUTO_INCREMENT,
    seller.asking_price INT,
    buyer.asking_price INT,
    got.property TINYINT,
    sell.price INT,
    offer.date DATE,
    sell.date DATE,
    property_id INT,
    realtor.relation_id INT,
    realtor.relation_id INT,
    FOREIGN KEY (property_id) REFERENCES property(property_id),
    FOREIGN KEY (buyer_realtor_relation_id) REFERENCES buyer_realtor_realtor(realtor_id),
    FOREIGN KEY (seller_realtor_relation_id) REFERENCES seller_realtor_realtor(realtor_id)
```



Where does our Data come from?

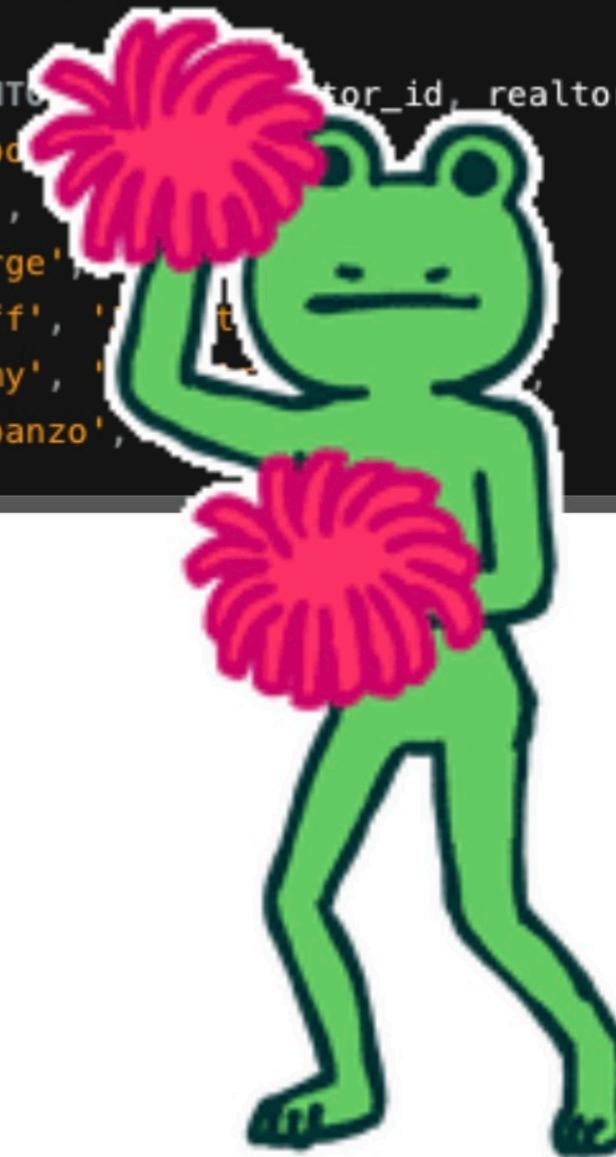
Our data was entered by our very own Daniel Smith



Where does our Data come from?

Our data was entered by our very own Daniel Smith

```
-- ===== Data Insertion =====  
INSERT INTO management_company(management_company_id, management_company_name) VALUES  
(1, 'Photon Mgmt'),  
(2, 'Real Bros Real Estate'),  
(3, 'Garbs Inc');  
  
INSERT INTO realtor(realtor_id, realtor_firstname, realtor_lastname, realtor_age, management_company_id) VALUES  
(1, 'Jimbo', 'Big', 30, 1),  
(2, 'Big', 'Jimbo', 30, 1),  
(3, 'George', 'Smith', 30, 1),  
(4, 'Geoff', 'Smith', 30, 1),  
(5, 'Timmy', 'Smith', 30, 1),  
(6, 'Garbanzo', 'Smith', 30, 1);
```



Yup



Where does our Data come from?

Our data was entered by

```
-- ===== Data =====
INSERT INTO management_company(management_company_id, man
(1, 'Photon Mgmt'),
(2, 'Real Bros Real Estate'),
(3, 'Garbs Inc');

INSERT INTO realtor(realtor_id, realtor_firstname, realto
(1, 'Jimbo', 'Jorges'),
(2, 'Big', 'Nate'),
(3, 'George', 'Jumbo'),
(4, 'Geoff', 'Jumbo'),
(5, 'Timmy', 'Jumbo'),
(6, 'Garbanzo', 'B
```

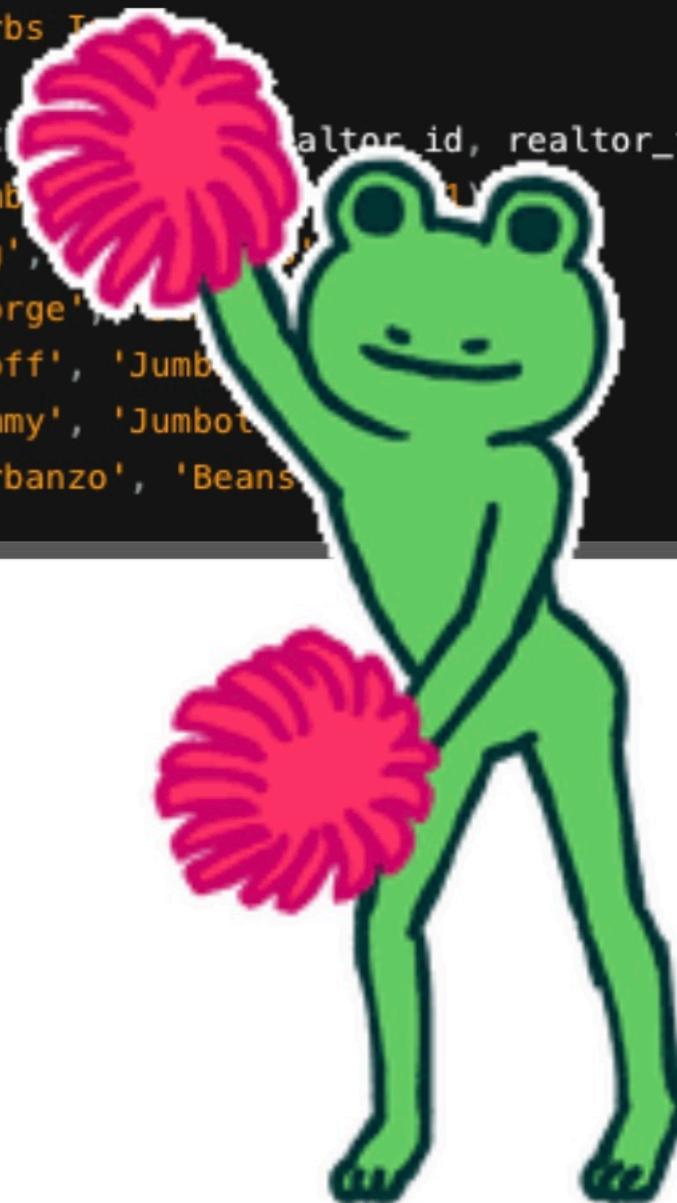
```
INSERT INTO zip_code(zip_code_id, avg_property_tax, region, state) VALUES
(02120, 8422, 'Suffolk', 'MA'),
(01740, 12636, 'Worcester', 'MA'),
(02019, 4748, 'Norfolk', 'MA'),
(02744, 10157, 'Bristol', 'MA'),
(02739, 9886, 'Plymouth', 'MA'),
(02115, 9015, 'Suffolk', 'MA');
```



Where does our Data come from?

Our data was entered by

```
-- ===== Data =====  
INSERT INTO management_company(management_company_id, man  
(1, 'Photon Mgmt'),  
(2, 'Real Bros Real Estate'),  
(3, 'Garbs T  
  
INSERT INTO realtor(realtor_id, realtor_fi  
(1, 'Jimb  
(2, 'Big'  
(3, 'George'  
(4, 'Geoff', 'Jumb  
(5, 'Timmy', 'Jumbo  
(6, 'Garbanzo', 'Beans')
```



```
INSERT INTO zip_code(zip_code_id, avg_property_tax, region, state) VALUES  
(02120, 8422, 'Suffolk', 'MA'),  
(01740, 12636, 'Worcester', 'MA'),  
(02019, 4748, 'Norfolk', 'MA'),  
(02744, 10157, 'Bristol', 'MA')
```

```
• ⊕ INSERT INTO transaction(transaction_id, seller.asking_price, buyer.asking_price, got.property, sell.price,  
offer.date, sell.date, property.id, buyer.realtor.relation.id, seller.realtor.relation.id) VALUES  
(1, 5200, 4770, 1, 4800, '2024-12-08', '2024-12-30', 1, 3, 1),  
(2, 875000, 625000, 1, 675000, '2025-02-22', '2025-02-28', 2, 12, 2),  
(3, 1700, 1600, 1, 1650, '2024-11-18', '2025-01-01', 3, 1, 3),  
(4, 4850, 4850, 1, 4850, '2022-05-11', '2022-07-22', 4, 21, 4),  
(5, 1175000, 650000, 0, 1250000, '2025-02-07', '2025-02-23', 6, 12, 6),  
(6, 1175000, 1250000, 1, 1250000, '2025-02-12', '2025-02-23', 6, 4, 6),  
(7, 5200, 5300, 1, 5250, '2024-08-27', '2024-09-11', 7, 17, 7),  
(8, 612500, 570000, 1, 575000, '2025-02-01', '2025-02-17', 14, 18, 14),  
(9, 3500, 3200, 1, 3475, '2024-07-29', '2024-08-29', 15, 9, 1),  
(10, 765000, 740000, 1, 750000, '2025-09-20', '2025-09-23', 10, 19, 10),  
(11, 2575, 2420, 1, 2500, '2025-10-22', '2025-10-23', 12, 20, 12),  
(12, 10000000, 6500000, 1, 7650000, '2025-01-24', '2025-02-25', 19, 21, 15),  
(13, 3000, 2925, 1, 3000, '2024-08-12', '2024-08-30', 20, 14, 18),  
(14, 4200, 3950, 1, 4100, '2024-09-15', '2024-10-05', 9, 22, 9),  
(15, 550000, 525000, 1, 540000, '2025-03-10', '2025-05-19', 24, 29, 24),  
(16, 3800, 3600, 1, 3725, '2024-12-01', '2024-12-20', 23, 27, 20),  
(17, 875000, 850000, 1, 862500, '2025-01-15', '2025-02-10', 22, 29, 22),  
(18, 4950, 4800, 1, 4900, '2024-10-18', '2024-11-12', 25, 30, 23);
```



Our queries



Our queries



```
-- 1. Number of sales made by each realtor

SELECT r.realtor_firstname, r.realtor_lastname, IFNULL(sum(t.got_property), '0') AS "num_sales"
FROM realtor r
LEFT JOIN seller_realtor_relation s ON r.realtor_id = s.realtor_id
LEFT JOIN transaction t ON s.seller_realtor_relation_id = t.seller_realtor_relation_id
GROUP BY r.realtor_firstname, r.realtor_lastname
ORDER BY num_sales DESC;
```



Our queries



```
-- 1. Number of sales
```

```
SELECT r.realtor_first
FROM realtor r
LEFT JOIN seller_realtor sr ON r.realtor_id = sr.realtor_id
LEFT JOIN transaction t ON sr.seller_realtor_id = t.seller_realtor_id
GROUP BY r.realtor_first
ORDER BY num_sales DESC
```

	realtor_firstname	realtor_lastname	num_sales
	Big	Nate	5
	Timmy	Jumbotron	5
	Jimbo	Jorges	3
	Geoff	Jumbotron	2
	George	Jumbotron	1
	Garbanzo	Beans	1

'0') AS "num_sales"

relation_id



Our queries



-- 2. Which realtor negotiates for their buyers the best?

```
SELECT ROUND(AVG((sell_price - buyer.asking_price) / (t.seller.asking_price - t.buyer.asking_price)),4)
      AS "pct_of_bid_ask_spread_realized",
r.realtor_firstname, r.realtor_lastname, m.management_company_name
FROM transaction t
INNER JOIN buyer_realtor_relation b ON t.buyer_realtor_relation_id = b.buyer_realtor_relation_id
INNER JOIN realtor r ON b.realtor_id = r.realtor_id
INNER JOIN management_company m ON r.management_company_id = m.management_company_id
GROUP BY r.realtor_firstname, r.realtor_lastname, m.management_company_name
ORDER BY pct_of_bid_ask_spread_realized ASC;
```

DEAL



Our queries

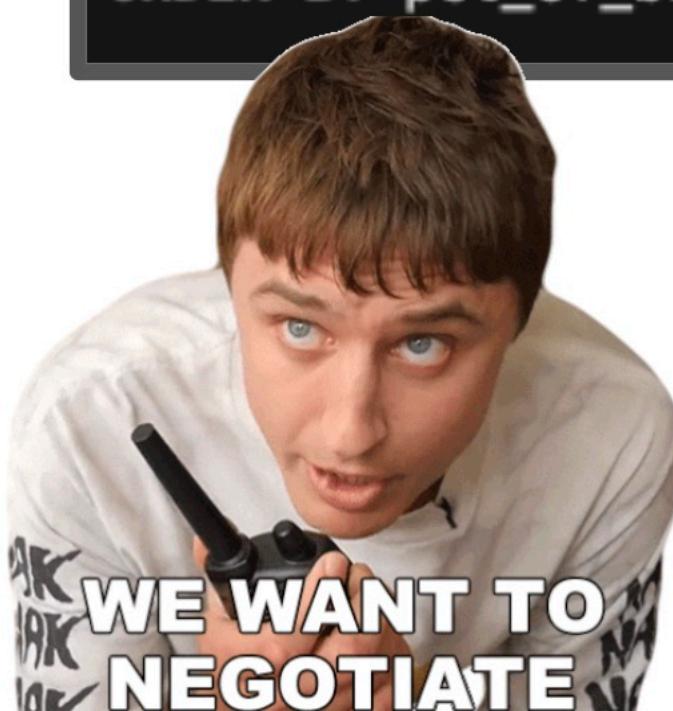


-- 2. Which realtor negotiates for their buyers the best?

```
SELECT ROUND(AVG((sell_price - buyer.asking_price) / (t.seller.asking_price - t.buyer.asking_price)),4)
AS "p
r.realtor
FROM trans
INNER JOIN
INNER JOIN
INNER JOIN
GROUP BY r.realtor_firstname, r.realtor_lastname, m.management_company_name
ORDER BY pct_of_bid_ask_spread_realized ASC;
```

pct_of_bid_ask_spread_realized	realtor_firstname	realtor_lastname	management_company_name
0.4765	Big	Nate	Photon Mgmt
0.5000	Geoff	Jumbotron	Real Bros Real Estate
0.5081	Jimbo	Jorges	Photon Mgmt
0.5139	Timmy	Jumbotron	Real Bros Real Estate
0.5500	George	Jumbotron	Real Bros Real Estate
0.6667	Garbanzo	Beans	Garbs Inc

DEAL



Our queries

Sale

-- 3. Which zip code has the most sales sorted in order

```
SELECT  
    p.zip_code_id AS "Zip Code",  
    COUNT(*) AS FREQUENCY  
FROM transaction t  
    JOIN property p  
        ON p.property_id = t.property_id  
GROUP BY p.zip_code_id  
ORDER BY FREQUENCY DESC;
```



WHAT WERE
YOUR SALES



Sale

Our queries

-- 3. Which zip code has the most sales sorted in order

```
SELECT  
    p.zip_code,  
    COUNT(*)  
FROM transactions  
JOIN properties  
ON p.property_id = t.property_id  
GROUP BY p.zip_code  
ORDER BY COUNT(*) DESC;
```

	Zip Code	Num_Sales
	2115	5
	1740	4
	2019	3
	2739	3
	2744	2
	2120	1



WHAT WERE
YOUR SALES



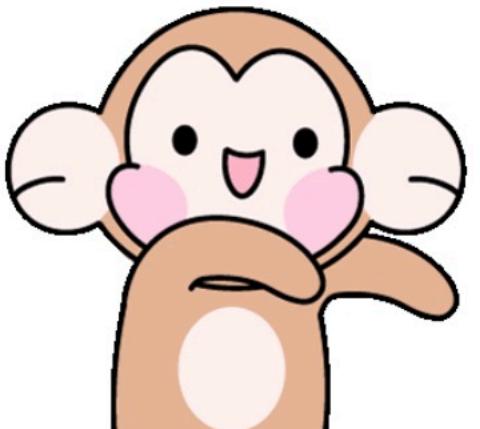


Our queries

```
-- 4. Which management company processes transactions the quickest (sell-date - offerdate)

SELECT
    TEMP.management_company_id,
    ROUND(AVG(days_taken), 2) AS AVG_DAYS_TAKEN
FROM
    (SELECT
        mc.management_company_id,
        DATEDIFF(t.sell_date, t.offer_date) AS days_taken
    FROM seller_realtor_relation srr
    JOIN realtor r ON srr.realtor_id = r.realtor_id
    JOIN management_company mc ON mc.management_company_id = r.management_company_id
    JOIN transaction t ON t.seller_realtor_relation_id = srr.seller_realtor_relation_id
    WHERE t.got_property = 1) TEMP
GROUP BY TEMP.management_company_id
ORDER BY AVG_DAYS_TAKEN ASC;
```

Hurry up! Hurry up!





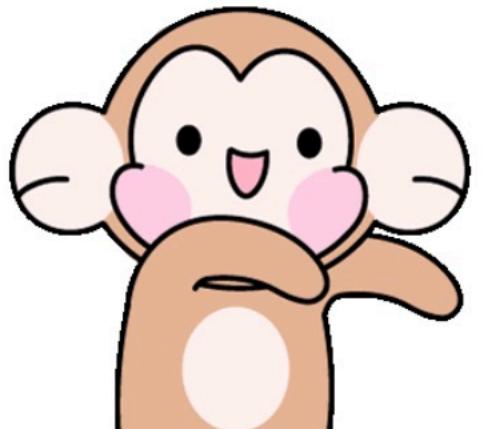
Our queries

```
-- 4. Which management company processes transactions the quickest (sell-date - offerdate)

SELECT
    TEMP.management_company_id,
    ROUND(AVG(days_taken), 2) AS AVG_DAYS_TAKEN
FROM
    (SELECT
        management_company_id,
        AVG(sell_date - offerdate) AS days_taken
    FROM
        transaction t
    JOIN
        seller_realtor_relation sr ON t.setter_realtor_relation_id = sr.setter_realtor_relation_id
    WHERE t.got_property = 1) TEMP
GROUP BY TEMP.management_company_id
ORDER BY AVG_DAYS_TAKEN ASC;
```



Hurry up! Hurry up!



Our queries

```
-- 5. Breakdown of the zipcode and housing types
```

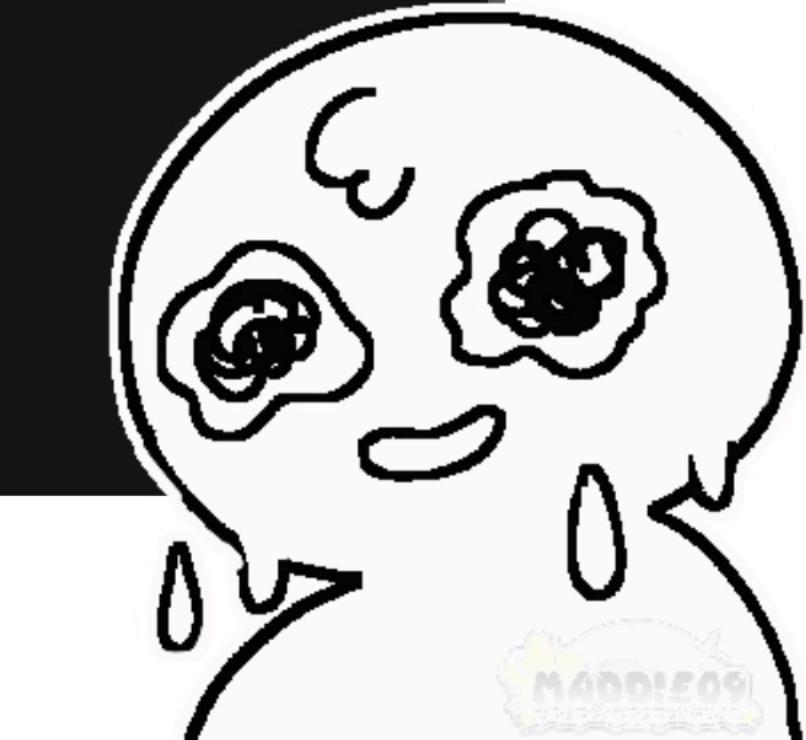
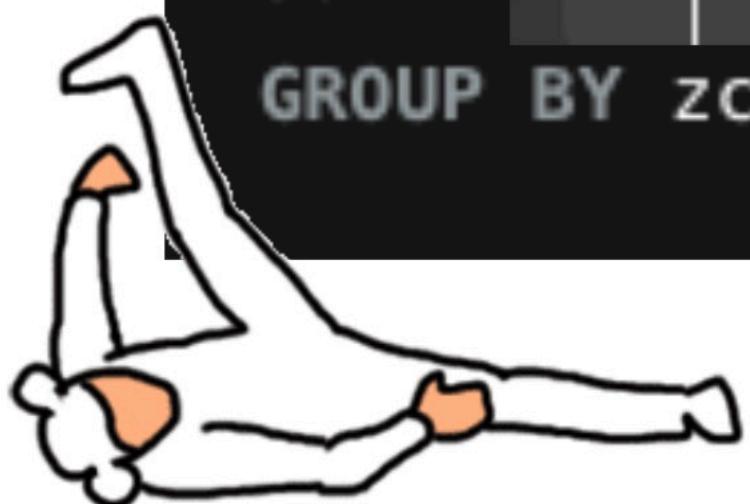
```
SELECT  
    zc.zip_code_id,  
    p.type,  
    COUNT(p.type) AS num_type,  
    ROUND(AVG(t.seller.asking_price), 2) AS avg_listing_price  
FROM zip_code zc  
JOIN property p ON zc.zip_code_id = p.zip_code_id  
JOIN transaction t ON p.property_id = t.property_id  
GROUP BY zc.zip_code_id, p.type;
```



Our queries

-- 5. Breakdown of the zipcode and housing types

```
SELECT zip_code_id, type, num_type, avg_listing_price
FROM zip_code AS zc
JOIN property AS p ON zc.zip_code_id = p.zip_code_id
JOIN listing AS l ON p.property_id = l.property_id
GROUP BY zc.zip_code_id, p.type;
```



Our queries

```
-- 6. What are the wealthiest zip codes? (this is based on average highest sell price of the homes)
```

```
SELECT
```

```
    zc.zip_code_id,  
    zc.region,  
    ROUND(AVG(t.sell_price), 2) AS AVG_SELL_PRICE
```

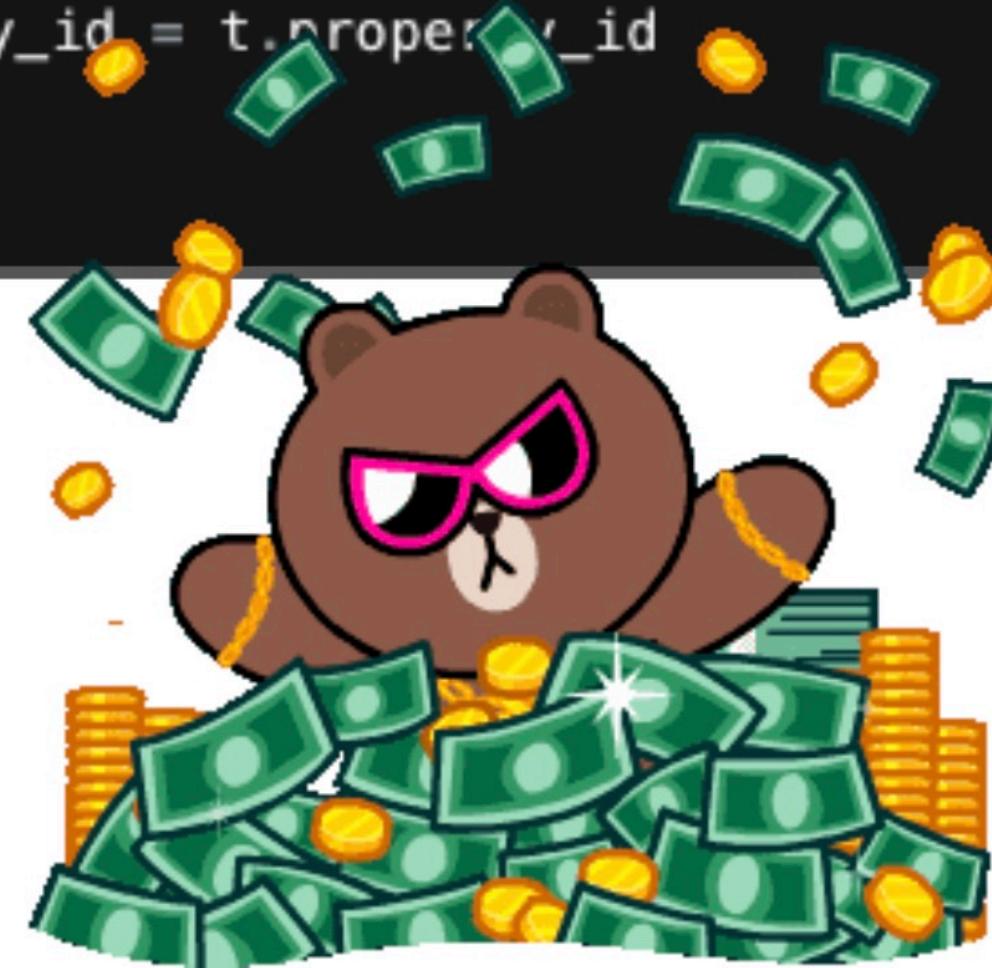
```
FROM property p
```

```
LEFT JOIN zip_code zc on zc.zip_code_id = p.zip_code_id
```

```
LEFT JOIN transaction t on p.property_id = t.property_id
```

```
GROUP BY zc.zip_code_id, zc.region
```

```
ORDER BY AVG_SELL_PRICE DESC;
```



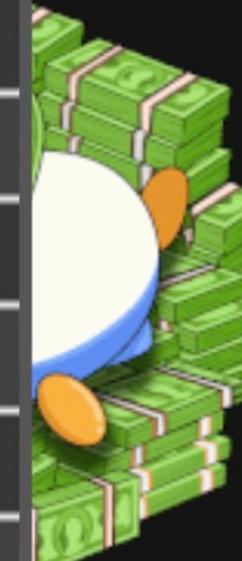
Our queries

-- 6. What are the we

```
SELECT  
    zc.zip_code_id,  
    zc.region,  
    ROUND(AVG(t.sell_  
FROM property p  
LEFT JOIN zip_code zc  
LEFT JOIN transaction t  
GROUP BY zc.zip_code_  
ORDER BY AVG_SELL_PRICE DESC;
```

	zip_code_id	region	AVG_SELL_PRICE
	2019	Norfolk	2552983.33
	2739	Plymouth	834966.67
	1740	Worcester	635000.00
	2744	Bristol	432987.50
	2120	Suffolk	4800.00
	2115	Suffolk	3225.00

rice of the homes)



Our queries



```
-- 7. What is each realtor's most successful zip code (most sales) assuming they all take a 15% stake in the final offer price?

SELECT
    TEMP.realtor_id,
    TEMP.realtor_firstname,
    TEMP.realtor_lastname,
    TEMP.zip_code_id,
    0.15 * TEMP.total_sales AS earnings
FROM (
    SELECT
        r.realtor_id,
        r.realtor_firstname,
        r.realtor_lastname,
        z.zip_code_id,
        SUM(t.sell_price) AS total_sales,
        ROW_NUMBER() OVER ( PARTITION BY r.realtor_id ORDER BY SUM(t.sell_price) DESC
        ) AS row_num
    FROM property p
    JOIN seller_realtor_relation srr ON srr.seller_id = p.seller_id
    JOIN realtor r ON srr.realtor_id = r.realtor_id
    JOIN transaction t ON p.property_id = t.property_id
    JOIN zip_code z ON z.zip_code_id = p.zip_code_id
    GROUP BY r.realtor_id, z.zip_code_id
    AS TEMP
    WHERE row_num = 1
    ORDER BY earnings DESC;
```



SUCCESS



Our queries

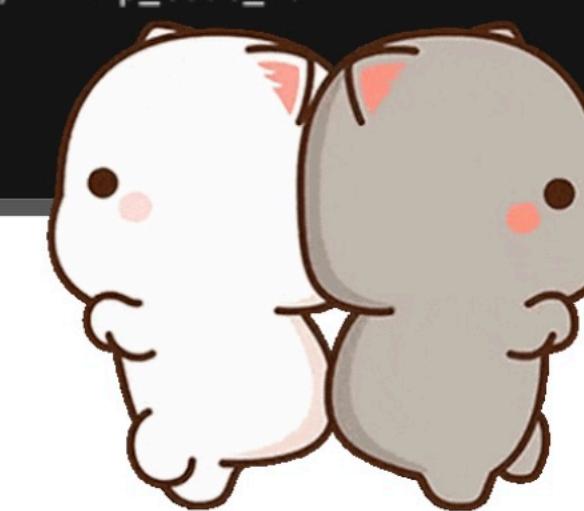


```
-- 7. What is each realtor's most successful zip code (most sales) assuming they all take a 15% stake in the final offer price?

SELECT
    TEMP.realtor_id,
    TEMP.realtor_firstname,
    TEMP.realtor_lastname,
    TEMP.zip_code_id,
    0.15 * TEMP.total_sales AS earnings
FROM (
    SELECT
        p.realtor_id,
        p.realtor_firstname,
        p.realtor_lastname,
        z.zip_code_id,
        COUNT(p.property_id) AS total_sales
    FROM property p
    JOIN seller_realtor_relation srr ON srr.seller_id = p.seller_id
    JOIN realtor r ON srr.realtor_id = r.realtor_id
    JOIN transaction t ON p.property_id = t.property_id
    JOIN zip_code z ON z.zip_code_id = p.zip_code_id
    GROUP BY r.realtor_id, z.zip_code_id
) AS TEMP
ORDER BY earnings DESC;
```

realtor_id	realtor_firstname	realtor_lastname	zip_code_id	earnings
6	Garbanzo	Beans	2019	1147500.00
5	Timmy	Jumbotron	2739	375000.00
2	Big	Nate	1740	187500.00
1	Jimbo	Jorges	1740	81000.00
4	Geoff	Jumbotron	2120	720.00

SUCCESS



Thank you for viewing our presentation.

