Bluefin: Realty Made Easy

Matthew Ambrose, Renee Cai, Daniel Smith
Northeastern University, Boston, MA, USA

Abstract

BlueFin, a real estate database application, addresses a problem commonplace throughout the industry: transparent and accessible information for everyone. BlueFin's goal of providing real estate information was accomplished through the creation of nine separate tables, each providing standalone information and connecting to form a holistic view of the market. BlueFin offers insight into the relationships between management companies and realtors, the properties realtors list, and how the market responds to these listings. To accomplish our goals, we needed a comprehensive and realistic information base. We input custom data into our tables tailored to the individuals included, the zip codes listed, and the current state of housing prices. When testing the use cases of our database application, several points became apparent. Foremost was the functionality for business cases. While we predicted that realtors' performance would be important, the timeliness of management companies was an area we did not expect to explore, proving consequential as one company took on average over ten days longer than the competition to complete transactions. Overall, the database application offered a variety of information sources for multiple stakeholders, enabling the informed evaluation of every decision point in the real estate process and achieving our mission of transparency.

Introduction

The need for transparent housing pricing is a pressing concern in today's socioeconomic environment. Additionally, the need for tracking performance and analyzing growth opportunities within companies is incredibly prevalent. BlueFin offers the solution to both of these problems. As a whole, BlueFin had the following goals: to serve as an evaluator of realtor and management company performance, provide insight into the trends of the real estate market over time, and ultimately, deliver transparency to buyers, sellers, and realtors alike.

Within the real estate industry, one of the greatest challenges is pricing transparency. There is a lack of concrete information, with currently existing organizations like Zillow, Redfin, and Apartments.com only offering estimates for the pricing of different properties. With limited information, buyers are often compelled to accept an inflated price for a property. The availability of detailed information on the real estate market would allow transactions to occur at optimal prices, benefiting the market as a whole.

The motivation behind this project stems from our time as students at Northeastern. Despite being one of the hottest housing markets in the world, Boston's real estate information is not transparent and often misleading, with properties listed on sites like Apartments.com having

valuations that are significantly over the typical selling price. With nowhere left to turn, BlueFin offers a haven for those in need. Gone are the days of wondering if a 3-bedroom, 2-bath apartment is truly commensurate with the $5100 monthly rent listed online. With candid apartment, condo, studio, and housing prices, more power is given to the buyer, a much-needed change in Boston.

BlueFin, a nine-table relational database using custom data, tracks the buyers, sellers, realtors, properties, and more, detailing the complex relationships that exist between parties in the real estate industry. By tying together different relationships through primary and foreign keys, other stakeholders' actions can be linked, offering insights ranging from the performance of a management company to the typical listings within a zip code. Different attributes, such as the number of beds and baths for properties, offer further evaluation of what flows into a unit's pricing.

Another significant aspect, showcasing BlueFin's flexibility, is its focus on the corporate side of real estate. By offering listing histories and detailed relationships between realtors and their clients, management companies can evaluate the performance of their employees, locate areas for improvement, and even have a statistical backing for end-of-year bonuses. For the realtors, the detailed listing of available buyers and sellers, as well as zip codes, offers a wealth of information for the taking. Should one realtor wish to improve their performance, they could relocate to a more profitable zip code and build relationships with high-value sellers.

Ultimately, throughout this report, we plan to cover the database design and schema behind BlueFin, different examples of its usage in query form, and our conclusions.

Database Design

We knew from the start that we would be entering our own data, so we didn't have to worry about cleaning it. We had nine tables, most of which were directly related to transactions or only separated by a join table that symbolized the relation between two parties. The most important tables were for transactions, buyers, sellers, and realtors. We had some additional tables for management companies and housing zip codes so we could have more data available for interesting queries. Overall, we focused most of our attention on the transactions since that would be the most important information for users when making decisions.
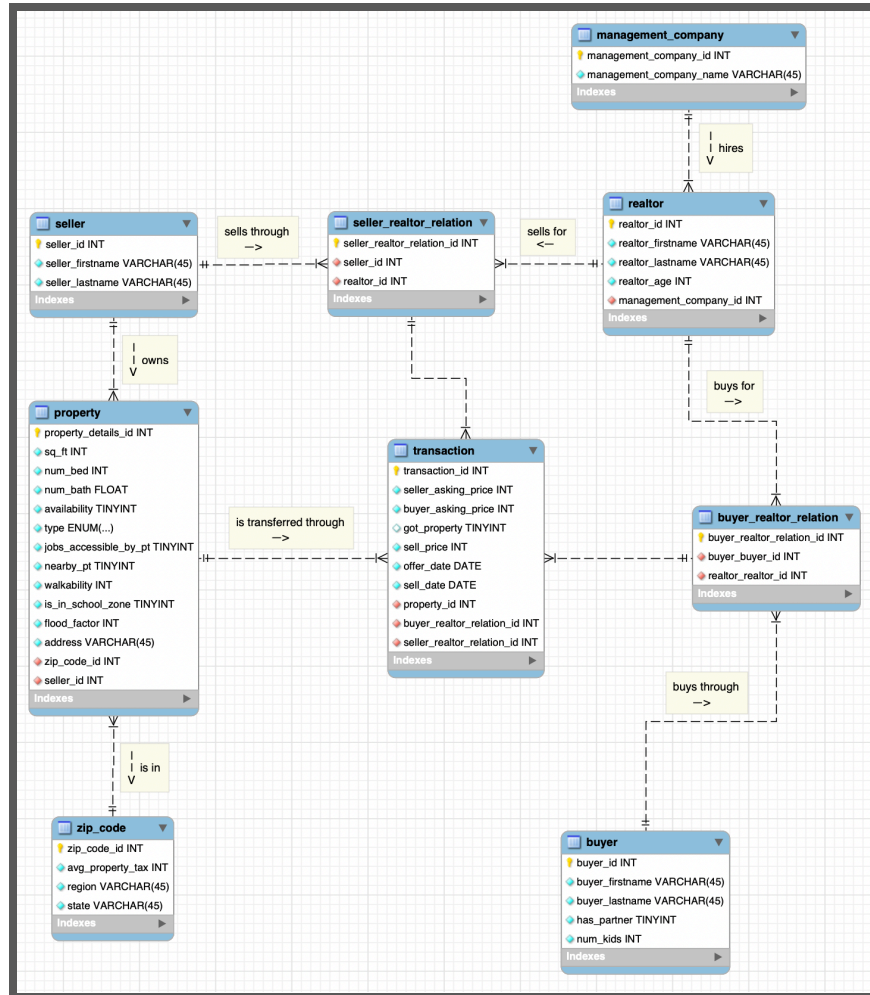
Data Source / Methods

We originally considered using a Python module called Faker to create example data; however, we decided not to follow through on this as we realized forming data by hand would offer more room for customization. As the data we utilized was strictly for portraying the functionality of our database application, we felt confident using our internalized knowledge of Boston's housing market from our time as students living in the area. The customization of data included custom names, listings, and transactions tailored for specific individuals, offering an element of realism.

To create the tables, we followed the ER diagram to establish complete relationships and focused on providing relevant information that could offer insight when queried.

Some assumptions we included were the following:

1. Center-city Boston zip codes, such as 02115 and 02120, were more likely to contain apartments than houses due to the relative lack of space compared to other zip codes

2. Every buyer and seller would only have one realtor, and would not be attributed to multiple for simplicity's sake
3. Houses or apartments could be bid on and lost, providing further realism to the application
4. School zones, walkability scores, nearby transportation, and jobs accessible by public transportation were all assigned with the assumption that inner city zip codes are more walkable, have more public transit, and are less likely to be in school zones. The flood factor was a randomized integer, although in real life, it might be higher in cities than in suburbs.
5. Pricing information is separated by unit type, with houses being the most expensive, and condos, apartments, and studios all being listed at a rental price. Houses were listed at the purchase price.
6. Sellers might not have to sell, and buyers do not necessarily have to buy; they are only in the pool of potential transactors
7. Buyers with more people in their household would naturally attempt to acquire property with more space, beds, and bathrooms
8. All information is made up and offers no discernible insight into the real Boston housing market

User cases

Question 1: How many sales are made by each realtor?

Query summary: Number of sales each realtor made.

```
-- 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;
```

realtor_firstna...	realtor_lastna...	num_sales
Big	Nate	5
Timmy	Jumbotron	5
Jimbo	Jorges	3
Geoff	Jumbotron	2
George	Jumbotron	1
Garbanzo	Beans	1

Question 2: Which realtor negotiates for their buyers the best?
Query summary: How close can a realtor get a sell price to the buyer's offer price, rather than the seller's asking price?

```sql
-- 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;
```

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

Question 3: Which zip code has the most sales?
Query summary: Number of sales per zip code.

```sql
-- 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;
```

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

Question 4: Which management company processes transactions the quickest?
Query summary: Average number of days each management company takes to process transactions.

```
-- 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;
```

management_company...	AVG_DAYS_TAKEN	
2	18.25	
3	19.00	
1	33.25	

Question 5: How are different housing types distributed among zip codes?

Query summary: How many of each housing type (apartment, house, studio, condo) are in each zip code, and how much do they cost on average?

```sql
-- 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;
```

zip_code_id	type	num_type	avg_listing_price
2120	apartment	1	5200.00
1740	house	4	700625.00
2115	studio	4	3318.75
2019	condo	2	4525.00
2739	apartment	3	784983.33
2744	apartment	1	3500.00
2019	house	1	10000000.00
2115	apartment	1	3000.00
2744	house	1	875000.00

Question 6: What are the wealthiest zip codes?

Query summary: Average housing sale price by zip code.

```
-- 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;
```

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

Question 7: In which zip code does each realtor perform the best?

Query summary: Where does each realtor make the most money, assuming they take in 15% of their total sales as earnings?

```
-- 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;
```

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

Conclusions

With the Bluefin app, we were able to demonstrate all of the relations we wanted to and use them to answer other questions. It can help management companies clearly evaluate realtor effectiveness and identify where they are underperforming or performing well. We were able to see how zip codes affected house sales and how many of each housing type were going on sale, providing insight into which zip codes would be the best for management companies to send their realtors to maximize profits. Finally, we were able to determine which management companies were the best for their customers, based on how quickly they processed transactions, as well as which of their realtors performed the best.

We didn't quite utilize Bluefin to its fullest extent, though. We focused on homes in Boston, which helped us concentrate our efforts, but we neglected housing types that might be found outside of Boston. However, we did have a good variety of housing types and a wide price range, so users should be able to use Bluefin to find any kind of housing that fits their needs.

Author Contributions

The authors contributed as follows:

Daniel hand-created the database information, writing the realty_creation.sql file and collaborating with Matthew and Renee as to the relationships included and any potential changes needed. He also participated in the query writing, focusing on realtor performance. Finally, Daniel wrote the abstract, introduction, and data source sections of the final report.

Matthew created and continuously updated the diagram in MySQL Workbench, basing the properties on a list that all three group members added to and worked with Renee and Daniel to incorporate more properties throughout the project. He also wrote the query and conclusion sections of the final report and polished up the report overall.

Renee participated in the majority of the query writing. She focused on specific aspects of the initial diagramming in the early planning stages. Additionally, she set up the GitHub repository and created the presentation slides. She compiled the project material and edited the final video presentation. She also elaborated on the questions in the SQL file and did a final report grammar check.