Blake Crowther
Alan Li
Wesley Schiller

Assignment 4

Part 1.

Business Opportunities Schema Introduction

This Neo4j schema represents a geographic and business-related data model, with a variety of nodes, relationships, and spatial layers. Here's a breakdown of its structure:

Nodes (Entities)

1. **Business**
    - Represents businesses with properties like a unique business_id, business_name, business_type (e.g., farmers market, grocery store, restaurant, bakery), location (latitude, longitude), address, rating, and price_level.
2. **BlockGroup**
    - Represents geographic block groups, including a unique ct_block_group, census_tract, block_group, object_id, and spatial data (wkt for geometry).
3. **Zipcode**
    - Represents zip code areas, with a unique zipcode_number and spatial data (wkt).
4. **City**
    - Represents cities, with properties like a unique city_id, city_name, state_name, county, and a flag is_unincorporated.
5. **Neighborhood**
    - Represents neighborhoods, identified by a unique neighborhood_id and neighborhood_name.
6. **Demographic and Index Nodes**
    - These nodes represent various demographic and socio-economic data about areas, including:
        - TotalPopulation (population levels: low, medium, high)
        - PopulationGrowth (growth rates)
        - AgeAverage (average age groups)
        - AgeGroup (age group representation)

- WealthIndex (wealth categories)
- EducationLevel (education levels)
- CrimeIndex (crime categories)
- FastFoodSpendingIndex (spending habits on fast food)

Relationships

1. **HAS_NEIGHBOR**
    - Indicates that a City or Neighborhood has a neighboring relationship with another City or Neighborhood.
2. **HAS_NEARBY**
    - Indicates a City or Neighborhood is nearby another City or Neighborhood.
3. **HAS_NEIGHBORHOOD**
    - Defines that a City has one or more Neighborhoods.
4. **IS_WITHIN**
    - Indicates spatial relationships where a City, Neighborhood, or BlockGroup is either partially or fully contained within a Zipcode, with an associated overlap_ratio.
5. **HAS_ENRICHMENT**
    - Connects BlockGroup nodes to various demographic data nodes (e.g., population, wealth index, crime index, etc.).

Spatial Layers

1. **block_group_layer**
    - Represents spatial data for BlockGroup nodes using the WKT (Well-Known Text) format for geometry.
2. **zipcode_layer**
    - Represents spatial data for Zipcode nodes, also using the WKT format for geometry.
3. **business_layer**
    - Represents Business nodes as simple point locations in space using a point encoder.

Data Model Purpose

This schema is designed to represent and analyze geographical and demographic data, with a focus on business locations and their relationships to geographic regions such as cities, neighborhoods, and zip codes to find business opportunities. The use of spatial layers and relationships allows for geographic queries and analysis, such as identifying nearby businesses, examining population characteristics, and understanding business distribution in relation to various social and economic factors.

By incorporating both location-based and socio-economic data (e.g., crime index, education level, wealth index), this model has the ability to expedite and enhance locating potential business opportunities for entrepreneurs, business owners, state, and government agencies such as the USDA.

LLM use in Schema Development and Applications

Large Language Model (LLM) like GPT were leveraged in various ways to assist in the development of the Business Opportunities schema. Below is an explanation of how an LLM were used and can be used at different stages of schema development and application.

1. Schema Design and Refinement

- **Assisting in Defining Node Types and Properties**:
    - LLMs were used to help clarify and define the purpose of each node and its properties. For example, for the Business node, the LLM suggests appropriate property types (e.g., STRING, FLOAT, POINT) based on the attributes that describe a business (such as business_name, location, rating, etc.).
    - If the schema evolves in the future, the LLM can assist by providing recommendations on the addition of new properties, such as "contact_info" for the business or "operating_hours," to enrich the model.
- **Data Validation and Integrity Constraints**:
    - The LLM was used in developing meaningful constraints for nodes and relationships, ensuring the uniqueness of business_id and block_group_id, and helping define appropriate constraints for the other fields (e.g., ensuring a rating value is between 0 and 5).
    - LLMs assist in formulating these constraints to ensure that the data stays consistent, relevant, and accurate.

2. Generating Relationships and Queries

- **Creating Relationships between Nodes**:
    - The HAS_NEIGHBOR relationship between City and Neighborhood is an essential aspect of geographic models. The LLM can help by generating the query patterns to connect relevant nodes. For instance, if a new Business is added, the LLM can help determine which Neighborhood or City it belongs to and create relationships accordingly.
- **Query Assistance**:

    - Once the schema is populated with data, the LLM assisted in generating graph queries (Cypher queries for Neo4j) to find relationships. For example, it can help write a query to find all businesses within a particular BlockGroup or locate the nearest grocery_store to a given business. Additionally, it can help with more complex queries such as analyzing the relationship between rating and price_level across neighborhoods or cities.

- **Providing Recommendations**:
    - Based on the schema, LLMs can also help suggest relationships or attributes to further develop the graph, like suggesting a new relationship type (e.g., "HAS_REVIEW" for businesses) based on user feedback or external data.

4. Data Insights and Analysis

- **Pattern Recognition**:
    - Once the schema is in use with real data, the LLM can assist in detecting patterns within the data. For example, it can identify areas where businesses with higher ratings are located in certain BlockGroups or neighborhoods. It might also suggest correlations between business type and pricing levels or geographic distribution patterns based on location data.
- **Predictive Analytics**:
    - Using historical data, LLMs can help predict trends or generate insights such as the future performance of certain business types in specific areas or the impact of location on customer ratings.
    - For example, the model could suggest that businesses in certain neighborhoods (identified by BlockGroup) with high ratings may have specific geographic or demographic characteristics, and thus predict that similar businesses could have similar success when located in other parts of the city.

5. Natural Language Interaction with Data

- **Simplified Queries for End Users**:
    - The LLM can be integrated into a system where non-technical users (e.g., business owners, city planners, or local authorities) can interact with the database using natural language. For instance, a user could ask, "Show me all the bakeries in the downtown area with a rating above 4," and the LLM could translate this into an appropriate Cypher query and retrieve the relevant results.
- **Generating Reports and Summaries**:
    - The LLM could automatically generate reports or summaries based on the data stored in the graph. For instance, it could summarize the distribution of businesses across neighborhoods, the average rating per business type, or trends in certain geographic regions.

6. Improving the Schema Over Time

- **Schema Evolution**:
    - As new use cases and requirements emerge, the LLM can assist in evolving the schema by suggesting new nodes, relationships, and properties. For example, if there is a need to capture customer feedback on businesses, the LLM can suggest adding a CustomerReview node with properties like review_id, customer_name, review_text, and a timestamp.
- **Optimization Suggestions**:
    - The LLM can help optimize the schema by suggesting performance improvements, such as the use of certain indexing strategies or refinements in how nodes and relationships are connected, especially as the dataset grows.

7. Documentation and Communication

- **Generating Documentation**:
    - LLMs can assist in creating documentation for the schema, including descriptions of each node, property, relationship, and the intended usage. This is valuable for future developers, data scientists, or analysts working with the data model.
- **Communication of Schema Changes**:
    - When the schema evolves (e.g., new types of businesses or new relationships), the LLM can generate change logs or communicate the impact of schema updates to stakeholders.

Conclusion

Incorporating an LLM into the development of this schema can enhance several aspects, from initial schema design and enrichment to querying, data analysis, and natural language interaction with the graph data. The LLM's capabilities in generating meaningful relationships, recommending data integrations, automating query generation, and providing insights ensure that the schema evolves in alignment with both user needs and available data.

Part 2.

Neo4J Competency Queries

1. List Businesses in a Specific Zipcode

This query will return a list of businesses located in a specific Zipcode. It will match the Zipcode and then fetch the businesses connected through the LOCATED_IN relationship, returning the business name, business type, rating, and price level, ordering businesses by ratings.

```
MATCH (b:Business)-[:LOCATED_IN]->(z:Zipcode)
WHERE z.zipcode_number = '92131'  // Replace with the target Zipcode
RETURN b.business_name, b.business_type, b.rating, b.price_level
```

```
ORDER BY b.rating DESC
```

Output:

The query will return:

- business_name: The name of the business.
- business_type: The type of business (e.g., "grocery_store", "fast_food_restaurant").
- business_address: The address of the business.
- business_rating: The rating of the business (if available).

This will provide a list of businesses located in the specified Zipcode.

b.business_name	b.business_type	b.rating	b.price_level
"Donutopolis Scripps Ranch Villages"	"bakery"	5.0	0

2. Average Rating of Businesses in a Zipcode

This query calculates the average rating of businesses within a specific zipcode.

```
MATCH (b:Business)-[:LOCATED_IN]->(z:Zipcode)
WHERE z.zipcode_number = '92131'  // Replace with the target Zipcode
RETURN AVG(b.rating) AS avg_rating
```

This query returns zipcode and avg_rating pairs for all zipcodes in San Diego county.

```
MATCH (b:Business)-[:LOCATED_IN]->(z:Zipcode)
WITH z.zipcode_number AS zipcode, avg(b.rating) AS avg_rating
RETURN zipcode, avg_rating
ORDER BY zipcode
```

Output:

The query will return a result with two columns:

- zipcode: The Zipcode number.
- avg_rating: The average rating of the businesses in that Zipcode.

This result will be a key-value pair, where zipcode_number is the key and avg_rating is the value.

zipcode	avg_rating
"91901"	2.91

3. Number of Businesses by Type in a Zipcode

This query will return the number of businesses per type within a given Zipcode.

```
MATCH (b:Business)-[:LOCATED_IN]->(z:Zipcode)
WHERE z.zipcode_number = '92131'  // Replace with the target Zipcode
RETURN b.business_type, COUNT(b) AS business_count
ORDER BY business_count DESC
```

This query will return zipcode, with business type and count for all zipcodes.

```
MATCH (b:Business)-[:LOCATED_IN]->(z:Zipcode)
WITH z.zipcode_number AS zipcode, b.business_type AS business_type, count(b) AS business
_count
RETURN zipcode, business_type, business_count
ORDER BY zipcode, business_type
```

Output:

The query will return a result with three columns:

- zipcode: The Zipcode number.
- business_type: The type of the business (e.g., "farmers_market", "grocery_store", etc.).
- business_count: The count of businesses of that type in that Zipcode.

This will allow you to see the number of businesses per type within each Zipcode node.

zipcode	business_type	business_count
"91901"	"bakery"	2

4. Crime Index and Businesses in a Zipcode

You can link businesses with the crime index within a zipcode. This can provide insights into how safe areas are, potentially influencing business decisions like where to open new stores.

```
MATCH (b:Business)-[:LOCATED_IN]->(z:Zipcode)
MATCH (bg:BlockGroup)-[:IS_WITHIN]->(z)
MATCH (bg)-[:HAS_ENRICHMENT]->(ci:CrimeIndex)
RETURN z.zipcode_number AS zipcode, ci.category AS crime_index, collect(b.business_name)
 AS businesses
ORDER BY z.zipcode_number
```

Output:

The result will show the following for each Zipcode:

- zipcode: The Zipcode number.
- crime_index: The crime index category for that Zipcode (as determined by the associated CrimeIndex node).
- businesses: A list of business names located within that Zipcode.

This query will allow you to see both the crime index and the businesses present within a specific Zipcode.

	zipcode	crime_index	businesses
1	"91901"	"SAFEST"	["Steph's Donut Hole", "Carl's Jr.", "Jack in the Box",

5. Population Growth and Business Activity in a Zipcode

This query ties together the population growth rate and business activity in a specific zipcode.

```
MATCH (b:Business)-[:LOCATED_IN]->(z:Zipcode)

MATCH (bg:BlockGroup)-[:IS_WITHIN]->(z)

MATCH (bg)-[:HAS_ENRICHMENT]->(pg:PopulationGrowth)

WITH z.zipcode_number AS zipcode, pg.growth_rate AS population_growth, COUNT(b) AS busi
ness_count

RETURN zipcode, population_growth, business_count

ORDER BY zipcode
```

Output:

The query will return a list of Zipcodes with the following columns:

- **zipcode**: The identifier for the Zipcode.
- **population_growth**: The population growth rate in the Zipcode (e.g., "NEGATIVE", "LOW", "MODERATE", "HIGH", "VERY_HIGH").
- **business_count**: The number of businesses located within that Zipcode.

zipcode	population_growth	business_count
"91901"	"NEGATIVE"	130

6. Which zip codes have the most number of low-income block groups?

This query returns the top 5 zip codes with the count of low-income block groups, sorted in descending order.

```
MATCH (z:Zipcode)<-[:IS_WITHIN]-(b:BlockGroup)-[:HAS_ENRICHMENT]->(w:WealthIndex
{category: "LOW"})

RETURN z.zipcode_number AS Zipcode, count(b) AS LowIncomeBlockGroups

ORDER BY LowIncomeBlockGroups DESC

LIMIT 5
```

Output

	Zipcode	LowIncomeBlockGroups
1	"92105"	63
2	"92154"	54
3	"91911"	53
4	"92115"	49
5	"92104"	46

7. Which zip codes have the most number of high-income block groups?

This query returns the top 5 zip codes with the count of high-income block groups, sorted in descending order.

```
MATCH (z:Zipcode)<-[:IS_WITHIN]-(b:BlockGroup)-[:HAS_ENRICHMENT]->(w:WealthIndex {ca
tegory: "HIGH"})
RETURN z.zipcode_number AS Zipcode, count(b) AS HighIncomeBlockGroups
ORDER BY HighIncomeBlockGroups DESC
LIMIT 5
```

Output

Zipcode	HighIncomeBlockGroups
"92067"	4
"92130"	4
"92009"	3
"92024"	3
"92014"	3

8. Which zip codes have the most number of middle-income block groups?

This query returns the top 5 zip codes with the count of middle-income block groups, sorted in descending order.

```
MATCH (z:Zipcode)<-[:IS_WITHIN]-(b:BlockGroup)-[:HAS_ENRICHMENT]->(w:WealthIndex {category: "MIDDLE"})
RETURN z.zipcode_number AS Zipcode, count(b) AS MiddleIncomeBlockGroups
ORDER BY MiddleIncomeBlockGroups DESC
LIMIT 5
```

Zipcode	MiddleIncomeBlockGroups
"92024"	18
"92129"	18
"92064"	15
"92009"	15
"92127"	12

9. Count block groups in each wealth index category

This query shows the population distribution by block group categories

```
MATCH (b:BlockGroup)-[:HAS_ENRICHMENT]->(w:WealthIndex)
RETURN w.category AS WealthCategory, count(b) AS BlockGroupCount
ORDER BY BlockGroupCount DESC
```

Output

WealthCategory	BlockGroupCount
"LOW"	1136
"LOWER_MIDDLE"	580
"MIDDLE"	238
"UPPER_MIDDLE"	94
"HIGH"	20

10. Show me Block Group Population Distribution by Zipcode (91901)?

This query shows the population distribution by block group categories for a specific zip code, in this case 91901.

```
MATCH (z:Zipcode {zipcode_number: "91901"})<-[:IS_WITHIN]-(b:BlockGroup)-[:HAS_ENRICHMENT]->(t:TotalPopulation)
RETURN z.zipcode_number AS Zipcode,
    t.level AS PopulationCategory,
    count(b) AS BlockGroupCount
ORDER BY PopulationCategory
```

Output

Zipcode	PopulationCategory	BlockGroupCount
"91901"	"HIGH"	8
"91901"	"LOW"	2
"91901"	"MEDIUM"	7

11. Which zip codes have the most block groups with fast food super fan?

This query returns top 5 zip codes with the count of block groups where residents are fast food super fans.

MATCH (z:Zipcode)<-[:IS_WITHIN]-(b:BlockGroup)-[:HAS_ENRICHMENT]->(f:FastFoodSpendingIndex {category: "SUPER_FAN"})

RETURN z.zipcode_number AS Zipcode, count(b) AS SuperFan

ORDER BY SuperFan DESC

LIMIT 5

Output

Zipcode	SuperFan
"92130"	6
"92014"	3
"92127"	2
"92067"	2
"92129"	1

12. Which zip codes have the most grocery stores?

This query lists top 5 zip codes with the count of grocery stores, sorted in descending order.

```
MATCH (z:Zipcode)<-[:LOCATED_IN]-(b:Business {business_type: "grocery_store"})
RETURN z.zipcode_number AS Zipcode, count(b) AS GroceryStores
ORDER BY GroceryStores DESC
LIMIT 5
```

Output

Zipcode	GroceryStores
"91950"	28
"92113"	27
"92025"	27
"92115"	26
"92105"	26

13. Which zip codes have the most bakery?

This query lists top 5 zip codes with the count of bakeries, sorted in descending order.

```
MATCH (z:Zipcode)<-[:LOCATED_IN]-(b:Business {business_type: "bakery"})
RETURN z.zipcode_number AS Zipcode, count(b) AS Bakery
ORDER BY Bakery DESC
LIMIT 5
```

Output

Zipcode	Bakery
"91950"	32
"92101"	26
"92111"	21
"92024"	19
"92126"	18

14. Which zip codes have the highest average business ratings?

This query returns top 5 zip codes with the highest average ratings of businesses, sorted in descending order.

```
MATCH (z:Zipcode)<-[:LOCATED_IN]-(b:Business)
RETURN z.zipcode_number AS Zipcode, avg(b.rating) AS AvgRating
ORDER BY AvgRating DESC
LIMIT 5
```

Output

Zipcode	AvgRating
"92067"	4.8
"92059"	4.733333333333333
"91906"	4.7333333333333325
"92061"	4.7
"92070"	4.7

15. Which zip codes have the most number of safest block groups?

This query lists top 5 zip codes with the count of block groups in the "safest" category.

```
MATCH (z:Zipcode)<-[:IS_WITHIN]-(b:BlockGroup)-[:HAS_ENRICHMENT]->(ci:CrimeIndex {category: "SAFEST"})
RETURN z.zipcode_number AS Zipcode, count(b) AS SafestBlockGroups
ORDER BY SafestBlockGroups DESC
LIMIT 5
```

Output

Zipcode	SafestBlockGroups
"92056"	41
"92064"	39
"92040"	38
"92129"	38
"92071"	37

16. Which zip codes have the most number of unsafe block groups?

This query lists top 5 zip codes with the count of block groups in the “unsafe” or "most unsafe" category.

```
MATCH (z:Zipcode)<-[:IS_WITHIN]-(b:BlockGroup)-[:HAS_ENRICHMENT]->(ci:CrimeIndex)
WHERE ci.category IN ["UNSAFE", "MOST_UNSAFE"]
RETURN z.zipcode_number AS Zipcode, count(b) AS MostUnsafeBlockGroups
ORDER BY MostUnsafeBlockGroups DESC
LIMIT 5
```

Output

Zipcode	MostUnsafeBlockGroups
"92101"	35
"92113"	17
"91950"	14
"92108"	13
"92154"	12

1. List Businesses in a Specific Zipcode

This query will return a list of businesses located in a specific Zipcode. It will match the Zipcode and then fetch the businesses connected through the LOCATED_IN relationship, returning the business name, business type, rating, and price level, ordering businesses by ratings.

```
MATCH (b:Business)-[:LOCATED_IN]->(z:Zipcode)
WHERE z.zipcode_number = '92131'  // Replace with the target Zipcode
RETURN b.business_name, b.business_type, b.rating, b.price_level
ORDER BY b.rating DESC
```

Output:

The query will return:

- business_name: The name of the business.
- business_type: The type of business (e.g., "grocery_store", "fast_food_restaurant").
- business_address: The address of the business.
- business_rating: The rating of the business (if available).

This will provide a list of businesses located in the specified Zipcode.

b.business_name	b.business_type	b.rating	b.price_level
"Donutopolis Scripps Ranch Villages"	"bakery"	5.0	0

2. Average Rating of Businesses in a Zipcode

This query calculates the average rating of businesses within a specific zipcode.

```
MATCH (b:Business)-[:LOCATED_IN]->(z:Zipcode)
WHERE z.zipcode_number = '92131'  // Replace with the target Zipcode
RETURN AVG(b.rating) AS avg_rating
```

This query returns zipcode and avg_rating pairs for all zipcodes in San Diego county.

```
MATCH (b:Business)-[:LOCATED_IN]->(z:Zipcode)
WITH z.zipcode_number AS zipcode, avg(b.rating) AS avg_rating
RETURN zipcode, avg_rating
ORDER BY zipcode
```

Output:

The query will return a result with two columns:

- zipcode: The Zipcode number.
- avg_rating: The average rating of the businesses in that Zipcode.

This result will be a key-value pair, where zipcode_number is the key and avg_rating is the value.

zipcode	avg_rating
"91901"	2.91

3. Number of Businesses by Type in a Zipcode

This query will return the number of businesses per type within a given Zipcode.

```
MATCH (b:Business)-[:LOCATED_IN]->(z:Zipcode)
WHERE z.zipcode_number = '92131'  // Replace with the target Zipcode
RETURN b.business_type, COUNT(b) AS business_count
ORDER BY business_count DESC
```

This query will return zipcode, with business type and count for all zipcodes.

```
MATCH (b:Business)-[:LOCATED_IN]->(z:Zipcode)
WITH z.zipcode_number AS zipcode, b.business_type AS business_type, count(b) AS business
_count
RETURN zipcode, business_type, business_count
ORDER BY zipcode, business_type
```

Output:

The query will return a result with three columns:

- zipcode: The Zipcode number.
- business_type: The type of the business (e.g., "farmers_market", "grocery_store", etc.).
- business_count: The count of businesses of that type in that Zipcode.

This will allow you to see the number of businesses per type within each Zipcode node.

zipcode	business_type	business_count
"91901"	"bakery"	2

4. Crime Index and Businesses in a Zipcode

You can link businesses with the crime index within a zipcode. This can provide insights into how safe areas are, potentially influencing business decisions like where to open new stores.

```
MATCH (b:Business)-[:LOCATED_IN]->(z:Zipcode)
MATCH (bg:BlockGroup)-[:IS_WITHIN]->(z)
```

```
MATCH (bg)-[:HAS_ENRICHMENT]->(ci:CrimeIndex)

RETURN z.zipcode_number AS zipcode, ci.category AS crime_index, collect(b.business_name)
 AS businesses

ORDER BY z.zipcode_number
```

Output:

The result will show the following for each Zipcode:

- zipcode: The Zipcode number.
- crime_index: The crime index category for that Zipcode (as determined by the associated CrimeIndex node).
- businesses: A list of business names located within that Zipcode.

This query will allow you to see both the crime index and the businesses present within a specific Zipcode.

	zipcode	crime_index	businesses
1	"91901"	"SAFEST"	["Steph's Donut Hole", "Carl's Jr.", "Jack in the Box",

5. Population Growth and Business Activity in a Zipcode

This query ties together the population growth rate and business activity in a specific zipcode.

```
MATCH (b:Business)-[:LOCATED_IN]->(z:Zipcode)

MATCH (bg:BlockGroup)-[:IS_WITHIN]->(z)

MATCH (bg)-[:HAS_ENRICHMENT]->(pg:PopulationGrowth)

WITH z.zipcode_number AS zipcode, pg.growth_rate AS population_growth, COUNT(b) AS busi
ness_count

RETURN zipcode, population_growth, business_count

ORDER BY zipcode
```

Output:

The query will return a list of Zipcodes with the following columns:

- **zipcode**: The identifier for the Zipcode.
- **population_growth**: The population growth rate in the Zipcode (e.g., "NEGATIVE", "LOW", "MODERATE", "HIGH", "VERY_HIGH").
- **business_count**: The number of businesses located within that Zipcode.

zipcode	population_growth	business_count
"91901"	"NEGATIVE"	130

6. Which zip codes have the most number of low-income block groups?

This query returns the top 5 zip codes with the count of low-income block groups, sorted in descending order.

```
MATCH (z:Zipcode)<-[:IS_WITHIN]-(b:BlockGroup)-[:HAS_ENRICHMENT]->(w:WealthIndex {category: "LOW"})
RETURN z.zipcode_number AS Zipcode, count(b) AS LowIncomeBlockGroups
ORDER BY LowIncomeBlockGroups DESC
LIMIT 5
```

Output

	Zipcode	LowIncomeBlockGroups
1	"92105"	63
2	"92154"	54
3	"91911"	53
4	"92115"	49
5	"92104"	46

7. Which zip codes have the most number of high-income block groups?

This query returns the top 5 zip codes with the count of high-income block groups, sorted in descending order.

```
MATCH (z:Zipcode)<-[:IS_WITHIN]-(b:BlockGroup)-[:HAS_ENRICHMENT]->(w:WealthIndex {ca
tegory: "HIGH"})
RETURN z.zipcode_number AS Zipcode, count(b) AS HighIncomeBlockGroups
ORDER BY HighIncomeBlockGroups DESC
LIMIT 5
```

Output

Zipcode	HighIncomeBlockGroups
"92067"	4
"92130"	4
"92009"	3
"92024"	3
"92014"	3

8. Which zip codes have the most number of middle-income block groups?

This query returns the top 5 zip codes with the count of middle-income block groups, sorted in descending order.

```
MATCH (z:Zipcode)<-[:IS_WITHIN]-(b:BlockGroup)-[:HAS_ENRICHMENT]->(w:WealthIndex
{category: "MIDDLE"})
RETURN z.zipcode_number AS Zipcode, count(b) AS MiddleIncomeBlockGroups
ORDER BY MiddleIncomeBlockGroups DESC
LIMIT 5
```

Output

Zipcode	MiddleIncomeBlockGroups
"92024"	18
"92129"	18
"92064"	15
"92009"	15
"92127"	12

9. Count block groups in each wealth index category

This query shows the population distribution by block group categories

```
MATCH (b:BlockGroup)-[:HAS_ENRICHMENT]->(w:WealthIndex)
RETURN w.category AS WealthCategory, count(b) AS BlockGroupCount
ORDER BY BlockGroupCount DESC
```

Output

WealthCategory	BlockGroupCount
"LOW"	1136
"LOWER_MIDDLE"	580
"MIDDLE"	238
"UPPER_MIDDLE"	94
"HIGH"	20

10. Show me Block Group Population Distribution by Zipcode (91901)?

This query shows the population distribution by block group categories for a specific zip code, in this case 91901.

```
MATCH (z:Zipcode {zipcode_number: "91901"})<-[:IS_WITHIN]-(b:BlockGroup)-[:HAS_ENRICHMENT]->(t:TotalPopulation)
RETURN z.zipcode_number AS Zipcode,
    t.level AS PopulationCategory,
    count(b) AS BlockGroupCount
ORDER BY PopulationCategory
```

Output

Zipcode	PopulationCategory	BlockGroupCount
"91901"	"HIGH"	8
"91901"	"LOW"	2
"91901"	"MEDIUM"	7

11. Which zip codes have the most block groups with fast food super fan?

This query returns top 5 zip codes with the count of block groups where residents are fast food super fans.

```
MATCH (z:Zipcode)<-[:IS_WITHIN]-(b:BlockGroup)-[:HAS_ENRICHMENT]->(f:FastFoodSpendingIndex {category: "SUPER_FAN"})
RETURN z.zipcode_number AS Zipcode, count(b) AS SuperFan
ORDER BY SuperFan DESC
LIMIT 5
```

Output

Zipcode	SuperFan
"92130"	6
"92014"	3
"92127"	2
"92067"	2
"92129"	1

12. Which zip codes have the most grocery stores?

This query lists top 5 zip codes with the count of grocery stores, sorted in descending order.

```
MATCH (z:Zipcode)<-[:LOCATED_IN]-(b:Business {business_type: "grocery_store"})
RETURN z.zipcode_number AS Zipcode, count(b) AS GroceryStores
ORDER BY GroceryStores DESC
LIMIT 5
```

Output

Zipcode	GroceryStores
"91950"	28
"92113"	27
"92025"	27
"92115"	26
"92105"	26

13. Which zip codes have the most bakery?

This query lists top 5 zip codes with the count of bakeries, sorted in descending order.

```
MATCH (z:Zipcode)<-[:LOCATED_IN]-(b:Business {business_type: "bakery"})
RETURN z.zipcode_number AS Zipcode, count(b) AS Bakery
ORDER BY Bakery DESC
LIMIT 5
```

Output

Zipcode	Bakery
"91950"	32
"92101"	26
"92111"	21
"92024"	19
"92126"	18

14. Which zip codes have the highest average business ratings?

This query returns top 5 zip codes with the highest average ratings of businesses, sorted in descending order.

```
MATCH (z:Zipcode)<-[:LOCATED_IN]-(b:Business)
RETURN z.zipcode_number AS Zipcode, avg(b.rating) AS AvgRating
ORDER BY AvgRating DESC
LIMIT 5
```

Output

Zipcode	AvgRating
"92067"	4.8
"92059"	4.733333333333333
"91906"	4.7333333333333325
"92061"	4.7
"92070"	4.7

15. Which zip codes have the most number of safest block groups?

This query lists top 5 zip codes with the count of block groups in the "safest" category.

MATCH (z:Zipcode)<-[:IS_WITHIN]-(b:BlockGroup)-[:HAS_ENRICHMENT]->(ci:CrimeIndex {category: "SAFEST"})

RETURN z.zipcode_number AS Zipcode, count(b) AS SafestBlockGroups

ORDER BY SafestBlockGroups DESC

LIMIT 5

Output

Zipcode	SafestBlockGroups
"92056"	41
"92064"	39
"92040"	38
"92129"	38
"92071"	37

16. Which zip codes have the most number of unsafe block groups?

This query lists top 5 zip codes with the count of block groups in the “unsafe” or "most unsafe" category.

```
MATCH (z:Zipcode)<-[:IS_WITHIN]-(b:BlockGroup)-[:HAS_ENRICHMENT]->(ci:CrimeIndex)
WHERE ci.category IN ["UNSAFE", "MOST_UNSAFE"]
RETURN z.zipcode_number AS Zipcode, count(b) AS MostUnsafeBlockGroups
ORDER BY MostUnsafeBlockGroups DESC
LIMIT 5
```

Output

Zipcode	MostUnsafeBlockGroups
"92101"	35
"92113"	17
"91950"	14
"92108"	13
"92154"	12