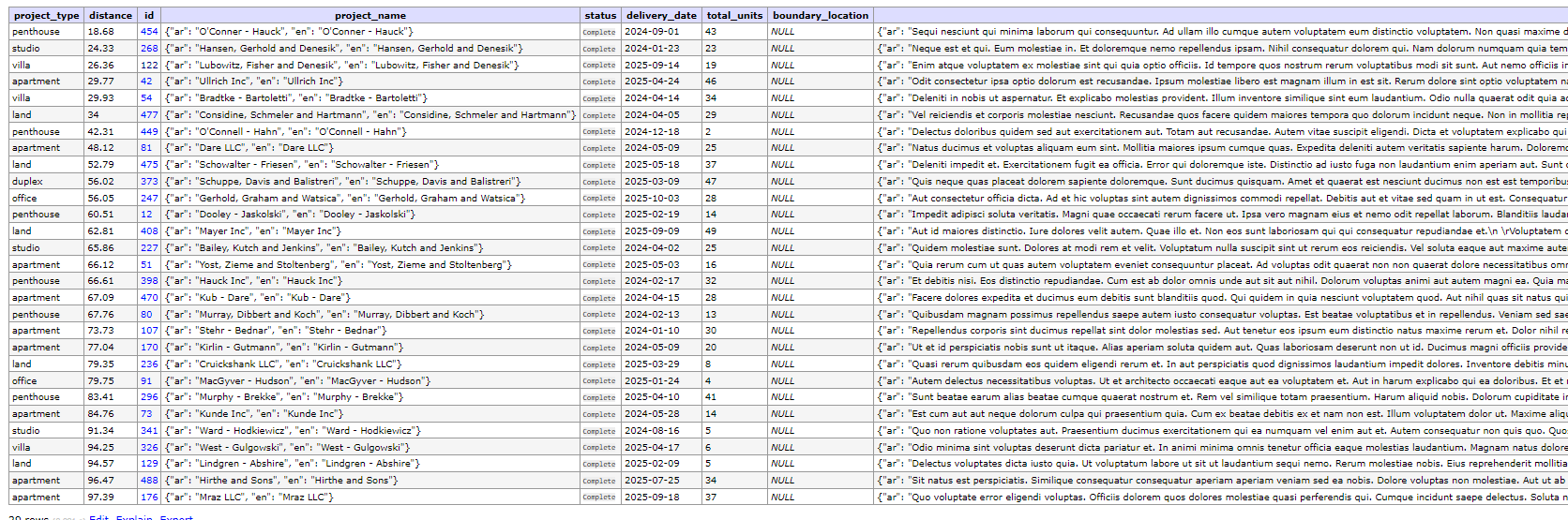
**Documentation: SQL Query and Table Structure**

****

**SQL Query**

**Description**

This SQL query retrieves project data from the database based on certain criteria such as project type, status, and distance from a specific location. It calculates the distance between project locations and a specified latitude and longitude using the Haversine formula.

**Query:**

sqlCopy code

SELECT JSON\_UNQUOTE(JSON\_EXTRACT(p.project\_name, '$.en')) AS project\_type, ROUND(6371 \* acos(cos(radians(22.85857000)) \* cos(radians(p.latitude)) \* cos(radians(p.longitude) - radians(51.82114000)) + sin(radians(22.85857000)) \* sin(radians(p.latitude))), 2) AS distance, p.\* FROM projects p JOIN project\_types pt ON p.project\_type\_id = pt.id WHERE JSON\_UNQUOTE(JSON\_EXTRACT(pt.type\_name, '$.en')) IN ('penthouse', 'studio', 'villa', 'apartment', 'land', 'duplex', 'office') AND p.status = 'Complete' AND 6371 \* acos(cos(radians(22.85857000)) \* cos(radians(p.latitude)) \* cos(radians(p.longitude)) - radians(51.82114000)) + sin(radians(22.85857000)) \* sin(radians(p.latitude)) <= 100 ORDER BY distance ASC;

**Parameters:**

* **p.project\_name**: Name of the project in JSON format.
* **p.latitude**: Latitude of the project location.
* **p.longitude**: Longitude of the project location.
* **p.status**: Status of the project (Complete or Under Construction).
* **pt.type\_name**: Type of the project in JSON format.
* **pt.id**: Unique identifier of the project type.
* **distance**: Calculated distance between project location and specified coordinates.

**Purpose:**

This query is designed to retrieve project data from the database based on specific criteria such as project type, status, and distance from a given location. It's useful for querying projects within a certain radius of a location and filtering them based on their type and status.

**Table Structure**

**Table Name: project**

| **Column Name** | **Data Type** | **Description** | **Default Value** |
| --- | --- | --- | --- |
| id | int unsigned | Auto-incremented unique identifier for the project |  |
| project\_name | json | Name of the project in JSON format |  |
| status | enum | Status of the project (Under Construction or Complete) | Under Construction |
| delivery\_date | date | Date of project delivery |  |
| total\_units | int | Total number of units in the project |  |
| boundary\_location | json | Location boundaries of the project in JSON format |  |
| project\_description | json | Description of the project in JSON format |  |
| agencies\_id | int unsigned | ID of the agency associated with the project |  |
| cover\_image | text | Image representing the project |  |
| project\_type\_id | int unsigned | ID of the project type |  |
| price\_per\_sq\_feet | int | Price per square foot of the project |  |
| buy\_plan | json | Buy plans for the project in JSON format |  |
| rent\_plan | json | Rent plans for the project in JSON format |  |
| is\_active | tinyint(1) | Flag indicating whether the project is active or not | 1 |
| created\_at | timestamp | Timestamp of when the project was created |  |
| updated\_at | timestamp | Timestamp of when the project was last updated |  |
| address | varchar(255) | Address of the project |  |
| starting\_price | double | Starting price of the project |  |
| emirate\_id | int unsigned | ID of the emirate where the project is located |  |
| area\_community\_id | int unsigned | ID of the community where the project is located |  |
| latitude | decimal(10,8) | Latitude of the project location |  |
| longitude | decimal(10,8) | Longitude of the project location |  |

**Table Name: project\_types**

| **Column Name** | **Data Type** | **Description** | **Default Value** |
| --- | --- | --- | --- |
| id | int unsigned | Auto-incremented unique identifier for the project type |  |
| type\_name | json | Name of the project type in JSON format |  |
| is\_active | tinyint(1) | Flag indicating whether the project type is active or not | 1 |
| created\_at | timestamp | Timestamp of when the project type was created |  |
| updated\_at | timestamp | Timestamp of when the project type was last updated |  |

**Indexes:**

* **type\_name**: JSON object with keys 'en' (English) and 'ar' (Arabic) representing the name of the project type in different languages.

**Purpose:**

These tables store information related to real estate projects and their types. The **project** table contains detailed information about each project, while the **project\_types** table contains information about different types of projects. The structure allows for efficient organization and retrieval of project data.

**Additional Notes:**

**JSON Data Format:**

* The JSON data format is utilized for storing certain information such as project names, descriptions, and project types. This allows for flexibility in handling multilingual data and structured information.

**Enumerated Data:**

* The **status** column in the **project** table is defined as an enumeration type, allowing only specific values (**Under Construction** or **Complete**). This ensures data integrity and consistency in representing the status of projects.

**Geospatial Data:**

* Latitude and longitude coordinates (**latitude** and **longitude** columns) are stored as decimal values with precision to eight decimal places, enabling accurate representation of project locations.

**Distance Calculation:**

* The SQL query employs the Haversine formula to calculate the distance between project locations and a specified latitude and longitude. This facilitates filtering projects based on proximity to a given location.

**Localization:**

* The **type\_name** column in the **project\_types** table is stored as a JSON object with keys for different languages (**en** for English and **ar** for Arabic). This allows for localization of project type names, catering to users from different language backgrounds.

**Indexing:**

* Indexes can be created on relevant columns to improve query performance, especially for frequently queried fields such as project names, project types, and geographical coordinates.

**Foreign Key Constraint:**

* The **project\_type\_id** column in the **project** table serves as a foreign key referencing the **id** column in the **project\_types** table. This maintains referential integrity and ensures that only valid project types are associated with projects.

**Default Values:**

* Default values are specified for certain columns (**status**, **is\_active**) to provide initial values if not explicitly provided during data insertion.

**Data Integrity:**

* Data integrity is enforced through constraints such as foreign key relationships and enumerated types, ensuring that only valid data is stored in the database.