Database Schema SQL Commands

1. Cities Table

sql

**CREATE** **TABLE** cities (

id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

name **VARCHAR**(255) NOT NULL,

state **VARCHAR**(255),

country **VARCHAR**(255)

);

2. Localities Table

sql

**CREATE** **TABLE** localities (

id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

city\_id **INT**,

name **VARCHAR**(255) NOT NULL,

landmark **VARCHAR**(255),

**FOREIGN** **KEY** (city\_id) **REFERENCES** cities(id) **ON** **DELETE** **CASCADE**

);

3. Developers Table

sql

**CREATE** **TABLE** developers (

id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

name **VARCHAR**(255) NOT NULL,

about **TEXT**,

contact\_info **VARCHAR**(255)

);

4. Projects Table

sql

**CREATE** **TABLE** projects (

id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

locality\_id **INT**,

developer\_id **INT**,

name **VARCHAR**(255) NOT NULL,

address **TEXT**,

rera\_status **VARCHAR**(255),

rera\_id **VARCHAR**(255),

launch\_date **DATE**,

possession\_start **DATE**,

overview **TEXT**,

project\_size **INT**, *-- Total number of units.*

why\_to\_choose **TEXT**,

about\_project **TEXT**,

**FOREIGN** **KEY** (locality\_id) **REFERENCES** localities(id) **ON** **DELETE** **CASCADE**,

**FOREIGN** **KEY** (developer\_id) **REFERENCES** developers(id) **ON** **DELETE** **CASCADE**

);

5. Configurations Table

sql

**CREATE** **TABLE** configurations (

id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

project\_id **INT**,

**type** **VARCHAR**(50), *-- E.g., 1BHK, 2BHK, etc.*

carpet\_area **INT**, *-- Sq. ft.*

price **DECIMAL**(10, 2), *-- Price of the configuration.*

emi **DECIMAL**(10, 2), *-- EMI starting from.*

segment\_floor\_image **VARCHAR**(255), *-- Image path.*

floor\_section **VARCHAR**(255), *-- Optional*

**FOREIGN** **KEY** (project\_id) **REFERENCES** projects(id) **ON** **DELETE** **CASCADE**

);

6. Amenities Table

sql

**CREATE** **TABLE** amenities (

id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

name **VARCHAR**(255) NOT NULL,

icon **VARCHAR**(255)

);

7. Project\_Amenities Table (Many-to-Many relationship)

sql

**CREATE** **TABLE** project\_amenities (

project\_id **INT**,

amenity\_id **INT**,

**PRIMARY** **KEY** (project\_id, amenity\_id),

**FOREIGN** **KEY** (project\_id) **REFERENCES** projects(id) **ON** **DELETE** **CASCADE**,

**FOREIGN** **KEY** (amenity\_id) **REFERENCES** amenities(id) **ON** **DELETE** **CASCADE**

);

8. Pricing Table

sql

**CREATE** **TABLE** pricing (

id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

project\_id **INT**,

avg\_price **DECIMAL**(10, 2),

emi\_starts **DECIMAL**(10, 2),

price\_floor\_plan **VARCHAR**(255), *-- Path to the image.*

**FOREIGN** **KEY** (project\_id) **REFERENCES** projects(id) **ON** **DELETE** **CASCADE**

);

9. Reviews Table

sql

**CREATE** **TABLE** reviews (

id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

project\_id **INT**,

user\_name **VARCHAR**(255),

rating **INT** **CHECK** (rating BETWEEN 1 AND 5), *-- Rating out of 5.*

review **TEXT**,

created\_at **TIMESTAMP** **DEFAULT** **CURRENT\_TIMESTAMP**,

**FOREIGN** **KEY** (project\_id) **REFERENCES** projects(id) **ON** **DELETE** **CASCADE**

);

10. Brochures Table

sql

**CREATE** **TABLE** brochures (

id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

project\_id **INT**,

brochure\_path **VARCHAR**(255),

**FOREIGN** **KEY** (project\_id) **REFERENCES** projects(id) **ON** **DELETE** **CASCADE**

);

11. Payment Plans Table

sql

**CREATE** **TABLE** payment\_plans (

id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

project\_id **INT**,

plan\_details **TEXT**,

**FOREIGN** **KEY** (project\_id) **REFERENCES** projects(id) **ON** **DELETE** **CASCADE**

);

12. Offers Table

sql

**CREATE** **TABLE** offers (

id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

project\_id **INT**,

offer\_details **TEXT**,

**FOREIGN** **KEY** (project\_id) **REFERENCES** projects(id) **ON** **DELETE** **CASCADE**

);

13. Maps Table

sql

**CREATE** **TABLE** maps (

id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

project\_id **INT**,

map\_view **VARCHAR**(255),

**FOREIGN** **KEY** (project\_id) **REFERENCES** projects(id) **ON** **DELETE** **CASCADE**

);

14. Developers\_Projects Table (Many-to-Many relationship)

sql

**CREATE** **TABLE** developers\_projects (

developer\_id **INT**,

project\_id **INT**,

**PRIMARY** **KEY** (developer\_id, project\_id),

**FOREIGN** **KEY** (developer\_id) **REFERENCES** developers(id) **ON** **DELETE** **CASCADE**,

**FOREIGN** **KEY** (project\_id) **REFERENCES** projects(id) **ON** **DELETE** **CASCADE**

);

15. Similar Projects Table

sql

**CREATE** **TABLE** similar\_projects (

id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

project\_id **INT**,

similar\_project\_id **INT**, *-- References another project in the same table.*

**FOREIGN** **KEY** (project\_id) **REFERENCES** projects(id) **ON** **DELETE** **CASCADE**,

**FOREIGN** **KEY** (similar\_project\_id) **REFERENCES** projects(id) **ON** **DELETE** **CASCADE**

);

Summary of Relationships

* **Cities to Localities**: One city can have many localities.
* **Localities to Projects**: One locality can have many projects.
* **Developers to Projects**: One developer can have many projects.
* **Projects to Configurations**: One project can have multiple configurations.
* **Projects to Amenities**: Many-to-many relationship through project\_amenities.
* **Projects to Reviews**: One project can have multiple reviews.
* **Developers to Projects**: Many-to-many relationship through developers\_projects.
* **Projects to Similar Projects**: One project can be linked to multiple similar projects.

Conclusion

This database schema provides a comprehensive structure for managing a real estate platform's data effectively. It captures essential entities and their relationships while ensuring scalability and maintainability.