**Database Design Guide**

This guide will help to create a database for managing architectural projects. It will manage the following functionalities:  
1. Cities and Offices  
2. Architects  
3. Users  
4. Projects and their Categories  
5. Project Images, Comments, and Blueprints  
6. Materials and their Usage  
  
We will use SQL as the DBMS to create the database and its related operations.

**Entities and Attributes**

**Cities**

* city\_id (PK)
* city\_name

**Offices**

* office\_id (PK)
* office\_name
* city\_id (FK) references Cities(city\_id)

**Architects**

* architect\_id (PK)
* first\_name
* last\_name
* email
* phone
* office\_id (FK) references Offices(office\_id)

**Users**

* user\_id (PK)
* username
* password
* email
* role\_id (FK) references Roles(role\_id)

**Roles**

* role\_id (PK)
* role\_name

**Projects**

* project\_id (PK)
* project\_name
* project\_description
* architect\_id (FK) references Architects(architect\_id)
* category\_id (FK) references Project\_Categories(category\_id)
* estimated\_cost

**Project\_Categories**

* category\_id (PK)
* category\_name

**Project\_Images**

* image\_id (PK)
* project\_id (FK) references Projects(project\_id)
* image\_url

**Comments**

* comment\_id (PK)
* project\_id (FK) references Projects(project\_id)
* user\_id (FK) references Users(user\_id)
* comment\_text
* comment\_date

**Blueprints**

* blueprint\_id (PK)
* project\_id (FK) references Projects(project\_id)
* blueprint\_url

**Materials**

* material\_id (PK)
* material\_name
* price\_per\_unit

**Material\_Usage**

* usage\_id (PK)
* project\_id (FK) references Projects(project\_id)
* material\_id (FK) references Materials(material\_id)
* quantity
* total\_price

**Payments**

* payment\_id (PK)
* user\_id (FK) references Users(user\_id)
* project\_id (FK) references Projects(project\_id)
* material\_usage\_id (FK) references Material\_Usage(usage\_id)
* amount
* payment\_date
* payment\_status
* payment\_method

**Relationships**

**Cities to Offices**

One-to-Many: One city can have multiple offices.  
city\_id (PK) in Cities is referenced by city\_id (FK) in Offices.

**Offices to Architects**

One-to-Many: One office can have multiple architects.  
office\_id (PK) in Offices is referenced by office\_id (FK) in Architects.

**Architects to Projects**

One-to-Many: One architect can handle multiple projects.  
architect\_id (PK) in Architects is referenced by architect\_id (FK) in Projects.

**Projects to Project\_Categories**

Many-to-One: Multiple projects can belong to one category.  
category\_id (PK) in Project\_Categories is referenced by category\_id (FK) in Projects.

**Projects to Project\_Images**

One-to-Many: One project can have multiple images.  
project\_id (PK) in Projects is referenced by project\_id (FK) in Project\_Images.

**Projects to Comments**

One-to-Many: One project can have multiple comments.  
project\_id (PK) in Projects is referenced by project\_id (FK) in Comments.

**Users to Comments**

One-to-Many: One user can post multiple comments.  
user\_id (PK) in Users is referenced by user\_id (FK) in Comments.

**Projects to Blueprints**

One-to-Many: One project can have multiple blueprints.  
project\_id (PK) in Projects is referenced by project\_id (FK) in Blueprints.

**Projects to Material\_Usage**

One-to-Many: One project can have multiple material usages.  
project\_id (PK) in Projects is referenced by project\_id (FK) in Material\_Usage.

**Materials to Material\_Usage**

One-to-Many: One material can be used in multiple projects.  
material\_id (PK) in Materials is referenced by material\_id (FK) in Material\_Usage.

**Users to Payments**

One-to-Many: One user can make multiple payments.  
user\_id (PK) in Users is referenced by user\_id (FK) in Payments.

**Projects to Payments**

One-to-Many: One project can have multiple payments.  
project\_id (PK) in Projects is referenced by project\_id (FK) in Payments.

**Material\_Usage to Payments**

One-to-One: Each payment is associated with a specific material usage.  
usage\_id (PK) in Material\_Usage is referenced by usage\_id (FK) in Payments.

**Users to Roles**

Many-to-One: Multiple users can have the same role.  
role\_id (PK) in Roles is referenced by role\_id (FK) in Users.

**Summary of Relationships**

**One-to-Many (1:N) Relationships:**

* Cities to Offices
* Offices to Architects
* Architects to Projects
* Projects to Project\_Images
* Projects to Comments
* Users to Comments
* Projects to Blueprints
* Projects to Material\_Usage
* Materials to Material\_Usage
* Users to Payments
* Projects to Payments

**Many-to-One (N:1) Relationships:**

* Projects to Project\_Categories
* Users to Roles

**One-to-One (1:1) Relationship:**

* Material\_Usage to Payments

**Relationship:**

1. Cities to Offices: `city\_id (PK)` in Cities is referenced by `city\_id (FK)` in Offices.

2. Offices to Architects: `office\_id (PK)` in Offices is referenced by `office\_id (FK)` in Architects.

3. Architects to Projects: `architect\_id (PK)` in Architects is referenced by `architect\_id (FK)` in Projects.

4. Projects to Project\_Categories: `category\_id (PK)` in Project\_Categories is referenced by `category\_id (FK)` in Projects.

5. Projects to Project\_Images: `project\_id (PK)` in Projects is referenced by `project\_id (FK)` in Project\_Images.

6. Projects to Comments: `project\_id (PK)` in Projects is referenced by `project\_id (FK)` in Comments.

7. Users to Comments: `user\_id (PK)` in Users is referenced by `user\_id (FK)` in Comments.

8. Projects to Blueprints: `project\_id (PK)` in Projects is referenced by `project\_id (FK)` in Blueprints.

9. Projects to Material\_Usage: `project\_id (PK)` in Projects is referenced by `project\_id (FK)` in Material\_Usage.

10. Materials to Material\_Usage: `material\_id (PK)` in Materials is referenced by `material\_id (FK)` in Material\_Usage.

11. Users to Payments: `user\_id (PK)` in Users is referenced by `user\_id (FK)` in Payments.

12. Projects to Payments: `project\_id (PK)` in Projects is referenced by `project\_id (FK)` in Payments.

13. Material\_Usage to Payments:`usage\_id (PK)` in Material\_Usage is referenced by `usage\_id (FK)` in Payments.

14. Users to Roles:`role\_id (PK)` in Roles is referenced by `role\_id (FK)` in Users