Database Design Report
Freelancer Mediator
ER diagram to Relational Schema

202203034 - Parth Sorathiya
202203065 - Paawan Vala

x

Contents

1 Introduction

This report describes the relational schema for the Freelancer Mediator system, which connects clients and freelancers on a project-based platform. It documents the entities, their attributes, keys, functional dependencies, and the classification into strong and weak entities.

2 Entities and Attributes

We identify the following base tables (entities) and their attributes:

2.1 Users

Attribute	Description
`user_id`	Unique identifier (PK)
`name`	Full name
`email`	Email address (UNIQUE)
`password`	Hashed password
`dob`	Date of birth
`country`	Country of residence
`created_at`	Timestamp of account creation
`updated_at`	Timestamp of last update

2.2 Freelancers

Attribute	Description
`freelancer_id`	Unique identifier (PK)
`user_id`	References `Users`(user_id) (FK)
`description`	Profile summary
`experience`	Years of experience
`rating`	Average rating
`availability_status`	Availability flag

2.3 Clients

Attribute	Description
`client_id`	Unique identifier (PK)
`user_id`	References `Users`(user_id) (FK)
`company_name`	Client organization name
`description`	Company description
`rating`	Average rating

2.4 Skills

Attribute	Description
`skill_id`	Unique identifier (PK)
`skill_name`	Name of the skill

2.5 Freelancer_skills

Attribute	Description
`freelancer_id`	Prime Attribute of composite PK (PA)
`skill_id`	Prime Attribute of composite PK (PA)

2.6 Projects

Attribute	Description
`project_id`	Unique identifier (PK)
`client_id`	References `Clients`(client_id) (FK)
`title`	Project title
`description`	Detailed description
`budget`	Budget allocation
`deadline`	Project deadline
`status`	{open, in_progress, completed, cancelled}
`created_at`	Timestamp of creation
`updated_at`	Timestamp of last update

2.7 Required_skills

Attribute	Description
`project_id`	Prime Attribute of composite PK (PA)
`skill_id`	Prime Attribute of composite PK (PA)

2.8 Contracts

Attribute	Description
`contract_id`	Unique identifier (PK)
`project_id`	References `Projects`(project_id) (FK)
`freelancer_id`	References `Freelancers`(freelancer_id) (FK)
`start_date`	Contract start date
`end_date`	Contract end date
`amount`	Agreed payment amount

`status`	{active, completed, terminated}
`created_at`	Timestamp of creation
`updated_at`	Timestamp of last update

2.9 Reviews

Attribute	Description
`review_id`	Unique identifier (PK)
`contract_id`	References `Contracts`(contract_id) (FK)
`reviewer_id`	References `Users`(user_id) (FK)
`reviewee_id`	References `Users`(user_id) (FK)
`rating`	Numeric score
`feedback`	Text comments
`created_at`	Timestamp of creation
`updated_at`	Timestamp of last update

2.10 Proposals

Attribute	Description
`project_id`	(PA) References `Projects`(project_id) (FK)
`freelancer_id`	(PA) References `Freelancers`(freelancer_id) (FK)
`proposal`	Proposal text
`bid_amount`	Proposed compensation
`status`	{pending, accepted, rejected}
`created_at`	Timestamp of submission

2.11 Payments

Attribute	Description
`contract_id`	(PK) References `Contracts`(contract_id) (FK)
`amount`	Payment amount
`payment_at`	Timestamp of payment

3 Functional Dependencies

- `Users`:
  - *user_id* → name, email, password, dob, country, created_at, updated_at
  - *email* → user_id, name, password, dob, country, created_at, updated_at
- `Freelancers`:
  - *freelancer_id* → user_id, description, experience, rating, availability_status

- `Clients`:
  - *client_id* → user_id, company_name, description, rating
- `Skills`:
  - *skill_id* → skill_name
- `Projects`:
  - *project_id* → client_id, title, description, budget, deadline, status, created_at, updated_at
- `Proposals`:
  - (*project_id*, *freelancer_id*) → proposal, bid_amount, status, created_at
- `Contracts`:
  - *contract_id* → project_id, freelancer_id, start_date, end_date, amount, status, created_at, updated_at
  - (*project_id*, *freelancer_id*) → contract_id
- `Reviews`:
  - *review_id* → contract_id, reviewer_id, reviewee_id, rating, feedback, created_at, updated_at
- `Payments`:
  - *contract_id* → amount, payment_at

4 Entity Classification

4.1 Strong Entities

- Users
- Freelancers
- Clients
- Skills
- Projects
- Contracts
- Reviews

4.2 Weak / Associative Entities

- Freelancer_skills (associative between Freelancers and Skills)
- Required_skills (associative between Projects and Skills)
- Proposals (associative between Projects and Freelancers)
- Payments (dependent on Contracts; supports milestone payments)

5 Anomaly Analysis

Based on the current schema design, all tables are normalized to Third Normal Form (3NF). This effectively eliminates the classical relational anomalies:

- **Insertion Anomaly**: No table requires unrelated data to exist before inserting a new fact. Each entity and associative table carries only attributes fully dependent on its key.
- **Update Anomaly**: There are no repeating or derived attributes across multiple rows—updating a value (e.g., a freelancer's description) occurs in exactly one place.
- **Deletion Anomaly**: Removing a row does not unintentionally erase other facts. For example, deleting a contract does not delete user or project data, since foreign keys preserve referential integrity.

Thus, the schema is free of insertion, update, and deletion anomalies.

6 Conclusion

This schema design achieves Third Normal Form (3NF), ensures clear separation of concerns. Surrogate keys are used where appropriate, and foreign key constraints maintain referential integrity.