Project Assignment-06

Name: Sachin Bansal, SID: 101318958

Freelancer Management Database Project

Youtube Video Link: https://youtu.be/ErwT2w_BcYI

1. **Background:**

Freelancing has become a popular trend that offers great flexibility to both workers and organizations, more importantly, provides exciting opportunities to the young aspiring freelancers. Although platforms like Upwork, Freelancer, and Fiverr provide a wide range of freelancers specializing in different fields, managing freelancer information has become complex and proper tracking of a freelancing talent pool available to any organization at any given time has become crucial for proper project management and its success.

My project aims to build a Freelancer Management Database that streamline the management of freelancers. It would provide key information about freelancers, including their skills, rates, availability, performance reviews, and the projects they have previously completed. By improving the trackability of completed projects and utilizing the past experience, the system will enable better matching of freelancers to future work opportunities.

2. **Application Requirements:**

Some of the required features of the Freelancer Management database are:

1. The database will provide detailed freelancer profiles, including key details like names, skills, houly rates, work history, availability, and reviews from past clients.
2. A freelancer can have multiple skills and a skill can be listed under multiple freelancers.
3. The database would enable organisations to assign freelancers to multiple projects and each project can involve multiple freelancers depending on the project requirements.
4. A search function implementation that allows organizations to find freelancers based on specific skills, availablility, and past performance.
5. The system would maintain a detailed project history for each freelancer and use this history for future project matching.
6. The application would track freelancer availability and allows when a freelancers are available for future work.
7. The database would allow managers to submit and update performance reviews based on the freelancers performance.
8. The system would also allow organisations to view and manage freelancer rates, and helping ensure projects stay within budget.

3. **Due Diligence:**

As this Freelancer Management Database will store only public and provided freelancer information and project details, we believe that there are no intellectual property or legal concerns regarding the creation and public use of this system, provided that the data is collected with proper consent from individuals or organizations.

CPY1.1 – It is essential that each freelancer or organization must consent prior to storing their information on this database. All the data stored in the database will be provided by the freelancers themselves or organizations employing them.

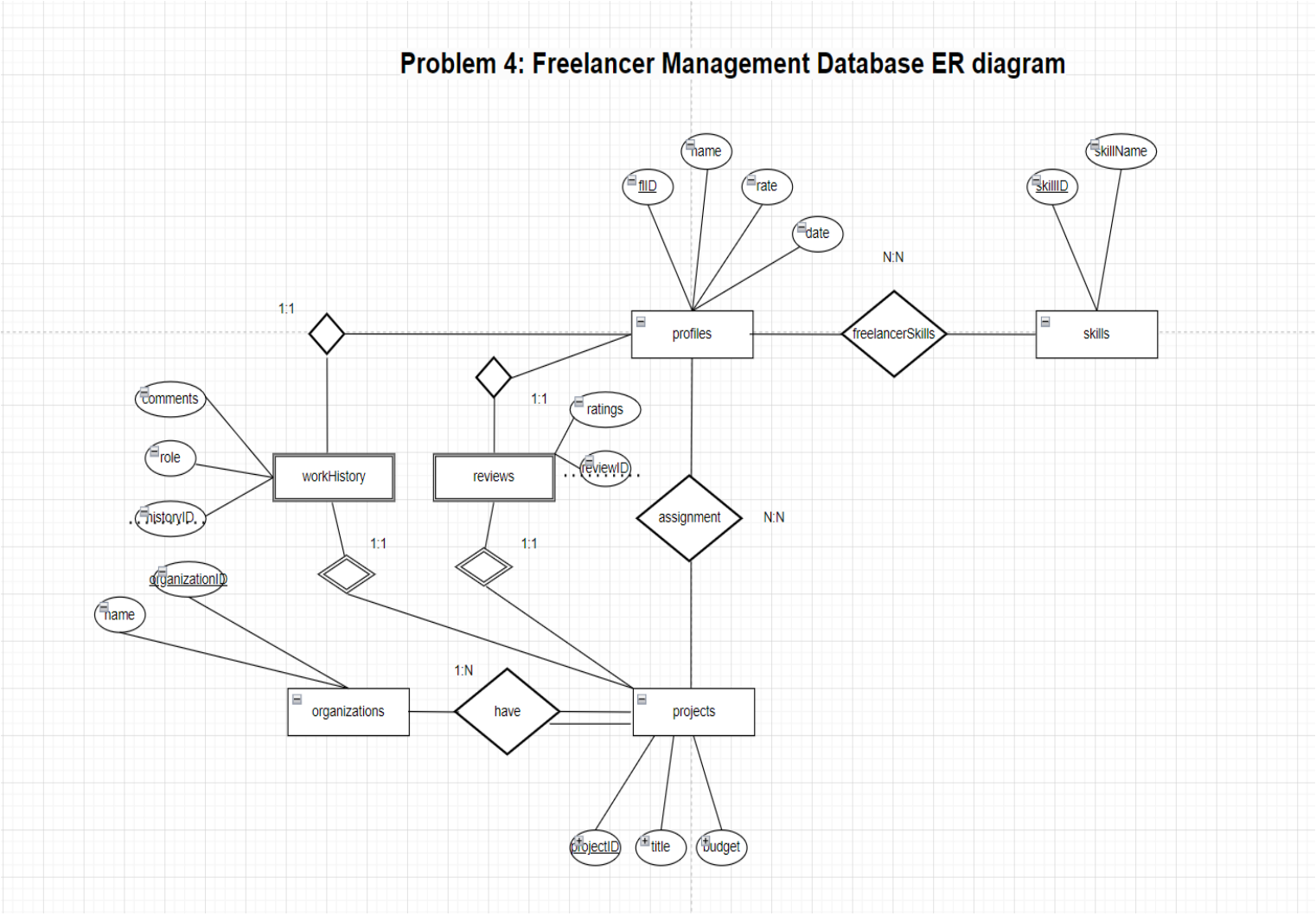
CPY1.2 – The database will not store any proprietary information work, source code, or confidential documents created by freelancers during their projects.

Sample Data entered to Database:

Below table represents how freelancer information might be organized in the database.

Name	Skills	Hourly Rates	Availability Date	Client Rating	Work history (Completed Work)	**Comments**
Adrian Smith	Web Development, Graphic Design	$50/hr	2025-01-10	4.8/5.0	Website Redesign for Alpha Techs	Great Work
Dung Long	Content Writing, Data Analysis	$45/hr	2025-01-22	4.5/5.0	Content Creation for IREP Inc	Good Work
David Hong	Data Analysis, Digital Marketing, Content Writing	$60/hr	2025-01-10	4.9/5.0	Marketing Materials for Nokia Inc	No Complaints
Mike Brom	Data Analysis, Digital Marketing	$45/hr	2025-02-01	4.7/5.0	Marketing Materials for Nokia Inc	Excellent Work
Andrew Kumar	Software Developmt Web Development	$55/hr	2025-02-22	4.3/5.0	Website Redesign for Alpha Techs	Top Notch Work

4. **PartProject ER Diagram**



4.1 Attributes Table:

Attribute	Comments
flID	Unqiue id for each freelancer
name	freelancer's name
rate	Hourly rate for work like 50 $/h is implied
availability (date)	When the freelancer is availabe for new projects
skillID	Unique ID for each skill
skillName	Name of the skill like programming
organizationID	Unique id for each organization
organizationName	Name of the organization hosting the project or hiring the freelancer
projectID	Unique ID for each project
title	Project official title
budget	Project's allocated budget
historyID	Unique ID for each work history entry
comments	Comments regarding freelancer's work on the project
role	Role of the freelancer while associated with the project
reviewID	Unique ID for each review
rating	Rating given to the freelancer for a specific project like 4/5

4.2 Assumptions and Constraints:

Profiles table: freelancer profiles

- flID: each freelancer must have a unique flID serving as PK
- name: freelancers name
- rate: hourly rate
- availability: start date
- each freelancer can be assigned many projects and vice versa where each project can have many freelancers (N:N relation)

skills table:

- skillID: each skill must have an unique ID serving as PK
- skillName: storing skill name

organizations:

- organizationID: each organization hunting for freelancer's must register with a unique ID serving as PK
- name: offical organization name
- An organization can have multiple projects but each project must be linked to one organization N:1 relationship

freelancerSkills: N:N relationship mapping between freelancer and skills as each freelancer can have multiple skills and each skill is known by multiple freelancers

- flID: FK from profiles table

- skillID: FK from skills table

projects: all, past and available projects

- projectID: unique and PK
- title: project's title
- budget: allocated budget by the organization
- organizationID: FK from organizations table
- One organization can many projects said earlier – one to many relationship with projects
- Project's existence rely on organization, they must be linked to exactly one organization (projects mandatory participation in relation with organizations)

assignment: N:N relationship between freelancerProfiles and projects

- flID: FK from freelaners profiles
- projectID: FK from projects
- Freelancer profiles have been assigned to projects. As said earlier its N:N relation

workHistory:

- historyID: PK also weak key
- comments: comments from previous work done
- role: role on the project while employed
- flID: FK from profiles
- projectID: FK from project
- One to one relation with profiles table, means each profile has one workHistory entry
- each flID profile can have one work history linked to that particular projectID, 1:1 relation with profiles as well
- workhistory is weak identity requiring projectID from projects table (workHistory existence rely on projects existence if there is no projects worked on before, the wH to projects relation can not form)

reviews:

- reviewID: PK, also weak key
- rating: ratings in decimal like 4/5
- flID: FK from profiles
- projectID: FK from project
- form one to one relation with profiles, 1:1 relation where each profile have one review entry.
- each flID profile have one review/rating linked to particular project ID
- reviews is weak identity requiring projectID from projects table

5.1 Functional Dependencies:

Minimal Cover: the functional dependencies below are already in minimal cover.

flID -> name, rate, avialability_date

skillID -> skillName

organizationID -> organization_name

projectID -> title, budget, organizationID

historyID -> comments, role, flID, projectID

reviewID -> rating, flID, projectID

flID, skillID -> temp1(N:N)

flID, projectID -> temp2 (N:N)

3NF Set of Tables:

Freelancers table: flID (PK), name, rate, date (availability_date)

FD: flID -> name, rate, avialability_date

Skills table: skillID (PK), skillName

FD: skillID -> skillName

Organizations table: organizationID (PK), organization_name

FD: organizationID -> organization_name

Projects table: projectID (PK), title, budget, organizationID(FK)

FD: projectID -> title, budget, organizationID

Work History table: historyID (PK), projectID(FK), comments, role, flID(FK)

FD: historyID -> comments, role, flID, projectID

Reviews table:reviewID (PK), rating, flID (FK), projectID(FK)

FD: reviewID -> rating, flID, projectID

FreelancerSkills table (N:N): flID and skillID (PK and FK)

FD: flID, skillID -> temp1(N:N)

Assignments table (N:N): flID and projectID (PK and FK)

FD: flID, projectID -> temp2 (N:N)

5.2 Schemas:

profiles:

flID	name	rate	availability

```
sqlite> select * from profiles;
flID  name          rate  availability
----  ------------  ----  ------------
1     Adrian Smith  50.0  2025-01-10
2     Dung Long     45.0  2025-01-22
3     David Hong    60.0  2025-01-10
4     Mike Brom     45.0  2025-02-01
5     Andrew Kumar  55.0  2025-02-22
```

skills:

skillID	skillName

```
sqlite> select * from skills;
skillID  skillName
-------  --------------------
1        Web Development
2        Graphic Design
3        Content Writing
4        Data Analysis
5        Digital Marketing
6        Software Development
```

organizations:

organizationID	name

```
sqlite> select * from organizations;
organizationID  name
--------------  -----------
1               Alpha Techs
2               IREP Inc
3               Nokia Inc
```

freelancerSkills: primary key is composite of {flID and skillID}

flID (Fk from profiles flID key)	skillID (Fk from skills, skillID key)

```
sqlite> select * from freelancerSkills;
flID  skillID
----  -------
1     1
1     2
2     3
2     4
3     3
3     4
3     5
4     4
4     5
5     1
5     6
```

projects:

projectID	title	budget	organizationID (FK from organizations, organizationID key)

```
sqlite> select * from projects;
projectID  title              budget   organizationID
---------  -----------------  -------  --------------
1          Website Redesign   10000.0  1
2          Content Creation   5000.0   2
3          Marketing Materials  7000.0   3
```

assgnment:

flID(PK and FK from profiles, flID)	projectID (PK and FK from projects, projectID key)

```
sqlite> select * from assignments;
flID  projectID
----  ---------
1     1
2     2
3     3
4     3
5     1
```

workHistory:

historyID	comments	role	flID(FK from profiles, flID key)	projectID(FK from projects, projectID key)

reviews:

reviewID	rating	flID (FK from profiles, flID key)	projectID (FK from projects, projectID key)

```
sqlite> select * from workHistory;
historyID  comments        role                flID  projectID
---------  -------------  ------------------  ----  ---------
1          Great Work     UI/UX Designer      1     1
2          Good Work      Content Writer      2     2
3          No Complaints  Data Analyst        3     3
4          Excellent Work Marketing Lead      4     3
5          Top-notch Work Software Developer  5     1

sqlite> select * from reviews;
reviewID  rating  flID  projectID
--------  ------  ----  ---------
1         4.8     1     1
2         4.5     2     2
3         4.9     3     3
4         4.7     4     3
5         4.3     5     1
```

6.1 Use Cases:

1. The Main use case of my app would be to track freelancer project assignments. My app store project that belong each organization (each project is lied to an organization, ie mandatory participation in the relationship).

Assign Freelancer to Project tab can help add existing freelancers (to add more freelancers profile visit Add Freelancer tab), to the existing projects in my database. This helps track each freelancer job assignments. The Following query also involves profiles and projects table JOIN with assignments (N:N relation between profiles and projects table) and perform the match based on matching flID and projectID

```
"SELECT profiles.name AS freelancer, projects.title AS project "
"FROM assignments "
"JOIN profiles ON assignments.flID = profiles.flID "
"JOIN projects ON assignments.projectID = projects.projectID"
```

2. The second best thing my database help track is each freelancer skills, where there is N:N relation between profiles and skills table. We can add freelancer to skills mapping by using Add Freelancer Skills tab, where you map existing profiles to exisitng skills. Currently, llists of skils can only be edited using manual databaase manipulation as I don't want to give access to change these to users, similarly with organizations, which can only be edited by making manual database additions.

This query also form a join of freelancerSkills, profiles and skills table.

```
"SELECT profiles.name as freelancer, skills.skillName "
"FROM freelancerSkills "
"JOIN profiles ON freelancerSkills.flID = profiles.flID "
"JOIN skills ON freelancerSkills.skillID = skills.skillID"
```

3. Also there is a query where I refinement the search of all the freelancers and the project assignment if there rate is more than 50 bucks. We can similarly build more refine queries.

This refinement helps print the profiles who's rate are more than 50+ and their project assignments. To form this query, first I fetched all the freelancer profiles and on select each project assignment where the flID matches for both the user and the assignments.

```
 "SELECT title FROM projects "
 "JOIN assignments ON assignments.projectID = projects.projectID "
 "WHERE assignments.flID = ?",
 (freelancer["flID"],),
```

7 Web Application:

Freelancer Management System

Home | View Freelancers | Add Freelancer | Projects | Add Project | Freelancer Skills | Add Freelancer Skills | Project Assignments | Work History | Reviews | Refine Query

Assignments

Existing Assignments

Freelancer	Project
Adrian Smith	Website Redesign
Dung Long	Content Creation
David Hong	Marketing Materials
Mike Brom	Marketing Materials
Andrew Kumar	Website Redesign
Sachin Bansal	Sample Project

Assign Freelancer to Project

Select Freelancer: Adrian Smith
Select Project: Website Redesign
Assign

Freelancer Management System

Home | View Freelancers | Add Freelancer | Projects | Add Project | Freelancer Skills | Add Freelancer Skills | Project Assignments | Work History | Reviews | Refine Query

Freelancer Skills

Freelancer	Skill
Adrian Smith	Web Development
Adrian Smith	Graphic Design
Dung Long	Content Writing
Dung Long	Data Analysis
David Hong	Content Writing
David Hong	Data Analysis
David Hong	Digital Marketing
Mike Brom	Data Analysis
Mike Brom	Digital Marketing
Andrew Kumar	Web Development
Andrew Kumar	Software Development
Sachin Bansal	Software Development
Sachin Bansal	Data Analysis