Assessment 1 – Portfolio

Student Name	Jina	**Student Number**	
Unit Code/s & Name/s	ICTNWK514 Model preferred systems solutions		
Assessment Name	**Portfolio** – Database Project: Modelling system solution	**Assessment Task No.**	1 of 2
Assessment Due Date		**Date submitted**	

Database Design Project - TechnoWorld Recruitment Agency

ASSESSMENT INSTRUCTIONS

PART ONE – Review and confirm client requirements

a) Read the project requirements provided and identify at least four (4) areas or requirements where you need further client clarification. The purpose of this task is to identify any overlooked functionality not covered in the requirements provided.

(1) System and Technology requirement
It includes existing system and utilities to migrate it into a new system if necessary
(2) Data requirement.
It includes existing data and migration.
(3) People requirement
This requirement is the identification of roles, accessibilities, etc.
(4) Business/Process requirement
(5) Security requirement

b) Prepare a list of no less than 30 questions grouped by the four (4) areas/requirements you identified as needing clarification.

System and Technology requirement
• Hardware - How much budget for hardware? - What kind of platform is ready for the new system? - If it is not ready, do you have any preferred brand of hardware? - If it is not ready, do you have any special requirement for hardware? • Software - How much budget for software? - Do you have any preferred Operating System, for example, Windows NT, Linux, etc.? - Do you have any preferred Database Management System, for example, MySQL, MS SQL Server and Oracle?

- Do you have any special requirement for software? • TechnoWorld has a semi-automated system that involves two different databases. - What kind of software are you using? - What kind of database management systems are you using? - What kind of hardware are you using? • Do you have any plan to migrate data from existing databases to a new database? • How large would be the existing databases be, approximately?
Data Requirement
• Do you have any standard job positions and codes? • Do you have any standard language and codes for skill areas, skills and qualification, including skill levels? • Do you have any standard visa types and codes?
People Requirement– roles, access, etc.
• What can the customers who have not registered see on the site? • Who manages the skills, skill types and job types? • Can job seeker create new skills, or only they choose one of the options? • Can job seeker or Employer modify the skill level from (3 levels to 5 levels) • The suggested system is a global system. Any job seekers from the world can access and apply for any positions advertised. Does any employer from other countries not Australian can advertise a job? • Does one organisation have more than one contact points, employers? • Who posts the job position? Answer: Employer can do. • Can the Employer delete or update job positions?
Processes/business requirement
• Do you have any rank system for the best fit? • What is the most relevant thing, what is the criteria of most preferred? • Can employers set the priority among criteria? • If there are no exact proper candidates, what is need to display?
Security Requirement
• Do you have any special security feature to be considered? • What kind of software are you using to secure the current system? • Do you have any special requirement to protect the new system and database?

c) In addition to your selected four (4) areas or requirements, you must discuss with the client:

1. Potential areas are likely to create bottlenecks. Outline the proposed solution/s to resolve bottlenecks.

   Refer Part one e)

2. The expected future expansion/s of the model/project.

Jina

Refer Part one e)

d) Send an email to the client/teacher requesting a **meetin**g to clarify the requirements. In the email, you must list the areas that you need to discuss.

To: Manuela Perez, Business Development Team Manager, TechnoWorld Recruitment Agency.
From: Jina Baek, Project Manager, Website Development Department, TAFE Company.
Title: Meeting of Business Requirement.

Dear Manuela Perez

I am happy to suggest the design and implementation of the TechnoWorld Recruitment website project and ask to have a meeting.

This meeting would include a discussion of the scope and requirements precisely.
The followings are the requirement list we need to review.

(1) System and Technology requirement
It includes the existing system and utilities to migrate it into a new system if necessary
(2) Data requirement.
It includes existing data and migration.
(3) People requirement
This requirement is about the identification of roles, accessibilities, etc.
(4) Business/Process requirement
(5) Security requirement

Our data analyst and I will join the meeting.
We are available on Tuesday, Wednesday, Thursday.

If you have any time and place for the meeting or any other topics to discuss, please let me know.
I am looking forward to the meeting.

Kind regards
Jina Baek, Project Manager

e) Attend the meeting and document the findings in a report. The marking criteria outline the meeting expectations.

The report must include:

1. The four (4) areas or requirements discussed with the client. For **each** requirement, write 150 to 250 words. Keep in mind that after the discussion, **new requirements** may have been added or modified.
2. Bottlenecks and proposed solutions (150 to 250 words)
3. Expected future model/project expansions (100 to 150 words)

Report

Requirements

1) System and Technology requirement

To run TechnoWorld website, Web hosting hardware is necessary and has three options, and TechnoWorld owns a web hosting server, Web hosting Service and Cloud services. The client needs easy management and more extensibility. The Amazon cloud service would be a good solution. The client expects more than 5000 users, and enough storage and hardware performance are required.

The service software also is necessary to run, and it includes an operating system, web server, database management system and security related things.

Migration data from existing semi-automated system needs to be considered. The migration from two different databases should be no data loss, consistent and keep integrity.
The databases should be compromised to a new centralised database, and all functionalities of TechnoWorld need to be implemented on it. Database software should be a relational database and easy to retrieve and manage.

The new system needs to be designed using organisational standard modelling technics, such as Entity Relationship Diagram (ERD), Data Flow Diagram (DFD), Use Case diagrams and Action diagrams.

For a secure environment for database management, automatic regular backup system should be built. Effective recovery methods need to be presented. This backup system should be easy to recover, and if any failure happens, fast recovery is essential to continue the web service as soon as possible.

2) Data requirement.

The consistency and integrity of data are important. To provide it, standard codes for the whole TechnoWorld website.

For example, the postcode for each country needs to be adapted from its national standard. For standardisation qualification and level codes, the Australian Qualification Framework (AQF) policy will be adapted, and visa information defined by Australian Government will be used. The standards of Job position and title, skill area, skill and skill level need to be defined, if necessary. All data shall be generated and managed in accordance with these standard codes.

And technically, foreign key constraints should be applied to tables that refer to these code tables. If codes of the existing system are different from those, an additional migration is necessary.

These codes and values will be managed by TechnoWorld system manager. Employers and job seekers can choose one of the defined values and cannot input a new value or modify an existing value.

3) People requirement

Only the authorised people can register and manage code information such as job position, skill and qualification, and potential employers and job seekers can only choose those codes.

One organisation is able to register multiple employers, and each Employer can register multiple job advertisements. Employers can retrieve and update their own job positions and cannot access others. Job seekers can register their skill and qualification information and update these and cannot access others

Registered employers can search all job seekers skill and personal details, which the seeker agreed to open. Registered job seekers can search for all jobs in the system and apply for these positions. They cannot access other seekers' profiles

Non-registered employers cannot see any seeker information. Non-register job seekers can see only a brief list of job positions.

System manager manage all data in the database and generate recruiting report for statistics.

4) Business/Process requirement

The new database will be used the backend of TechnoWorld website and will be connected web services. All data and functionalities need to be considered with web functionality together.

Potential employers and job seekers will use the system. After Registered members' login, they can use main functionalities of TechnoWorld website.

The functionalities are to find out candidates for employers and to search for job positions for job seekers. Employers can register job position advertisements and find out best-fit candidates, which is based on the following criteria, skill area, Skill and skill level, Year of experience, qualification and level, and visa status. According to the best-fit result, Employer will provide interview opportunities via email and phone.

To assure quality of the system, proper test scenarios need to be generated and tested and their expected input and output also need to be specified and tested. Those scenarios include 6 queries given by TechnoWorld to test the implemented database.

5) Security requirement

Only authorised people can access to the database and the stored data in database should be secure. Any unauthorised access should be prohibited. Therefore, strong user authentication is required, and user password need to be encrypted and protected.

Data transferred on website using HTTP is plain text and anyone on internet can see it. To prevent this, it should be protected via HTTPS.

Security software including firewall is necessary. A database firewall prevents access by default and allows only specific applications or web servers to access database. Only some identified connections are permitted, any malicious access would be prohibited. It would also secure your database from any remote access if it is not necessary. It also needs web application firewall. This will protect your database from SQL injection attacks by a web application.

The system should comply with the Australian government laws and regulations to protect personal information and prevent malicious usage.

Bottlenecks and proposed solutions

Some specific time that many people connect this website can be a bottle neck. At lunch time during work days and Friday afternoon it might have much traffic. The first thing to consider is monitoring. Check the number of user connection and find out the specific traffic hours. Then, analyse the which data (database table) and which application they use most. You also need to monitor the memory usage and storage usage. Out of memory can make a serious perfomance bottle neck.
For heavy access tables, you can define the optimisation methods and create new indexes. And split the table to partition to distribute the traffics.

An application which run a specific SQL query for long time can cause a bottle neck. The monitoring the database performance schema would help to find out which query cause a delay

and how long it takes.

Security Software such as a firewall can be a bottle neck. It is important to decide appropriate level of defence between security and performance.

Expected future model/project expansions

This project is the design and implementation of database in TechnoWorld recruiting system. It needs to extend to web service implementation to provide web functionality globally. For the service, proper platform including appropriate hardware and software need to be prepared and web application should be designed and implemented. The new web application needs to be connected to the database implemented in this project. This requires additional time and costs including human resource.

After implementation of the web service, the system can be connected to external databases to provide job seekers more job opportunities from external recruiting agencies. This can be cooperated to this system or be implemented as extra functionality.

f) Identify and explain three (3) or more missed opportunities of the clients' current semi-automated system.

- The legacy system does not provide international access to job information. People who live out of Australia cannot see and apply any job position. Potential employers can have limited applications and less chances to find the best candidates. Therefore, the client loses the job seekers from overseas and the employers who want better candidates.

- The current system provides only the manual matching methods, which cannot find best-fits automatically between employer requirements and job seekers owned ones. It takes a lot of time and effort to find the best matches. The client wastes their own human resources and time, and also loses potential customers who look for efficiency, saving time and cost

- The new system will provide standard skill and qualification categories and visa types. These standards will reduce matching cost. It also makes job seekers easily find out requirements for a job and employers identify their necessities well.

g) At this stage you should be able to identify what modelling must be done for the database project. Complete the table provided to ensure that all modelling tasks comply with the clients' organisational standards. As a **minimum** you need to create models for:

1. Database design (ERD)
2. Processes and functional decomposition (DFD)
3. People and processes (Use Case diagram)

REQUIRED MODELLING /SCHEMA	TOOLS/ TECHNOLOGIES	Complies with Organisational standards?
Database design	Entity Relational Diagram	Yes
Processes and functional decomposition	Data Flow Diagram	Yes
People and processes	Use Case Diagram	Yes

PART TWO – Create Modelling solutions

a) Create a Use Case diagram to demonstrate the user interactions with the system. The diagram must be detailed and accurate and must be able to be traced to the clients' requirements. – future work for 517

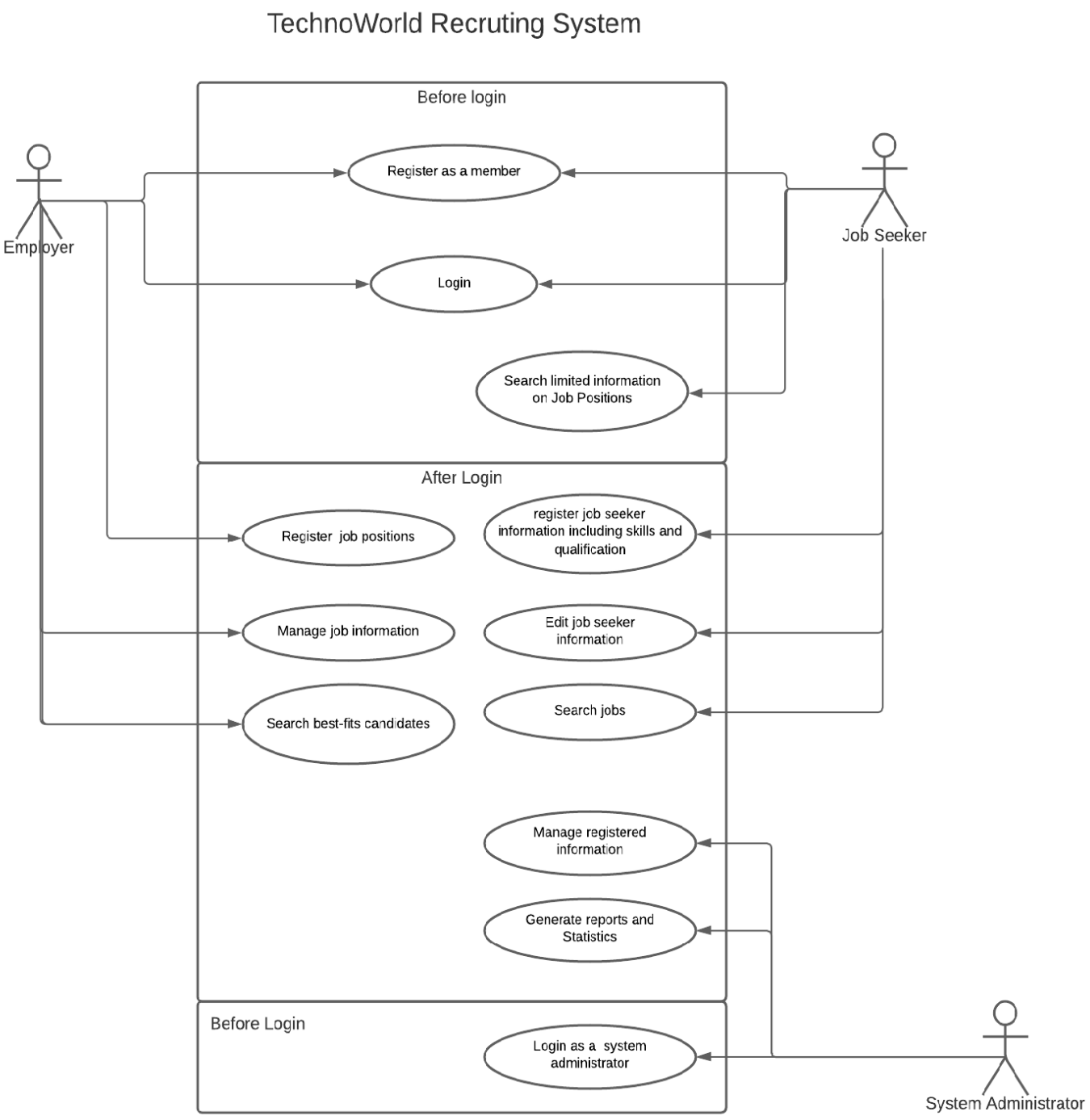


Figure 1 Data Flow Diagram

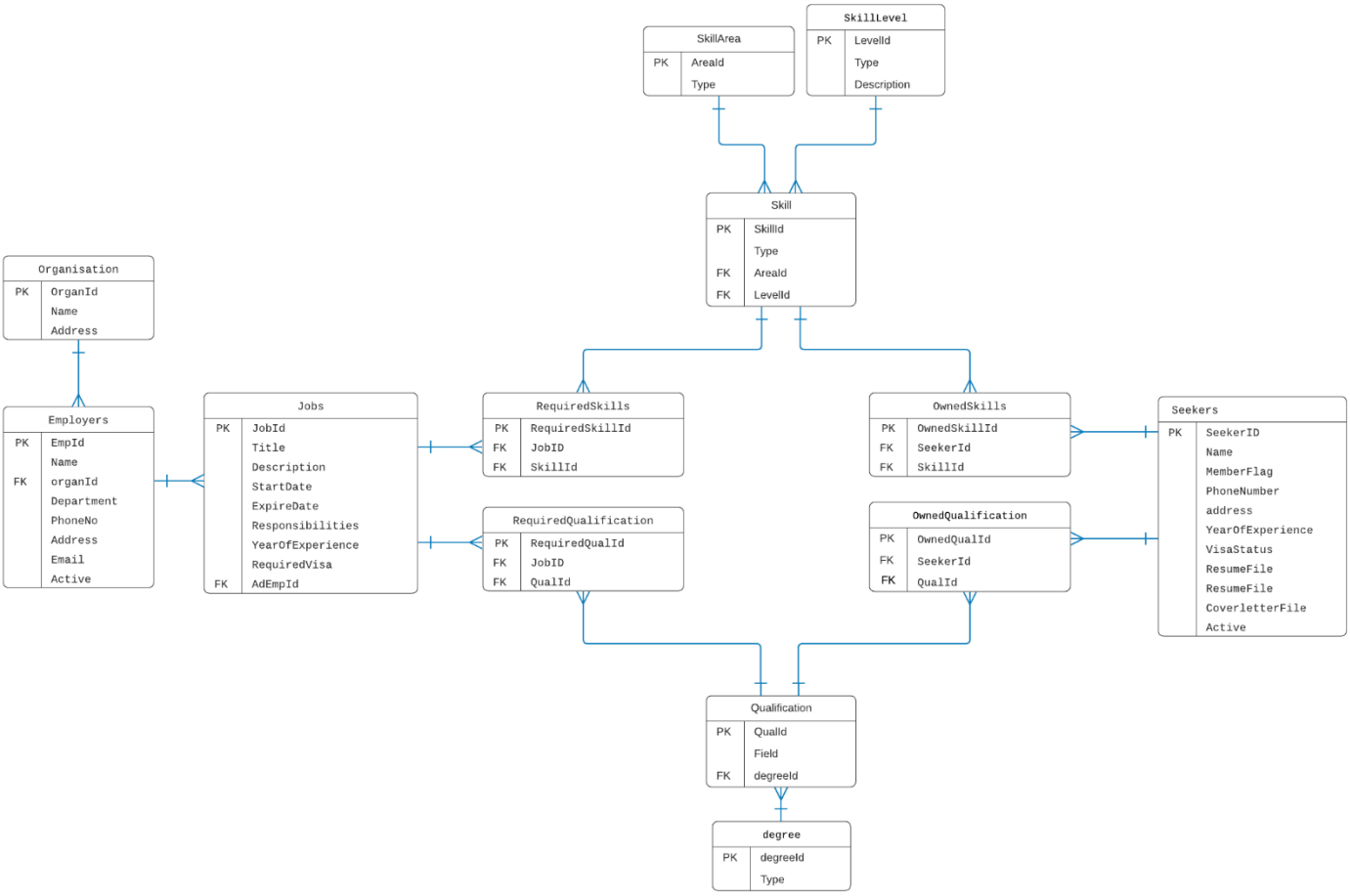
b) Create the modelling solution for the database design (ERD). Identify and document any assumptions you have made about the database project that have been incorporated into the ERD.

Assumption

1. In one organisation, multiple employers can advertise.
2. An employer can advertise multiple job positions.
3. An address of the Organisation table is head office or representative ones.
4. One job position can require multiple skills or qualifications.
5. A job seeker can have multiple skills and qualifications.
6. In the tables, SkillLevel and QualificationLevel, id means a level (lower id: lower level). For example, id 1 represents a Beginner level.
7. If a job seeker is an Australian citizen or permanent resident, the seeker does not have the visa information.

More assumption will be presented in the detailed table specifications.

ERD



c) Normalise the resulting database design diagram. Show all the normalisation steps taken. A final normalised list of tables is not sufficient. The normalised tables must include primary and foreign keys.

1NF: Primary key exists, and all attributes are atomic
2NF: A table is in 1NF.
All attributes are fully dependent on the primary key.
3NF: A table is in 2NF. All other attributes except the primary key are non-transitively dependent on the primary key.

Jina

1. **Organisation**

Name	Type	Key	Comment
OrganId	Unsigned Int(11)	PK	
Name	Varchar(50)		
Address	Varchar(30)		

The Organisation table has primary key. However, the address is not atomic. This table needs to be split. An address is composed of unit and street address, suburb, state, country and postcode. To prevent transitive-dependency on the primary key, it is split to three tables, PostGlobal, PostLocal and Organisation.

1.1 PostGlobal

Name	Type	Key	Comment
PostGlobalId	Unsigned Int(11)	PK	
City	Varchar(30)		Is Null, Optional
State	Varchar(30)		
Country	Varchar(30)		

1NF: Satisfied
2NF: Satisfied
3NF: Satisfied

1.2 PostLocal

Name	Type	Key	Comment
PostLocalId	Unsigned Int(11)	PK	
Suburb	Varchar(30)		
Postcode	Varchar(10)		
PostGlobalId	Unsigned Int(11)	FK	

1NF: Satisfied
2NF: Satisfied
3NF: Satisfied

1.3 Organisation

Name	Type	Key	Comment
OrganId	Unsigned Int(11)	PK	
Name	Varchar(50)		
Address	Varchar(30)		Unit number and Street name
PostLocalId	Unsigned Int(11)	FK	

1NF: Satisfied
2NF: Satisfied
3NF: Satisfied

2. **Employer**

Name	Type	Key	Comment
EmployerId	Unsigned Int(11)	PK	
Name	Varchar(30)		
Email	Varchar(30)	U	Login ID
Password	Varchar(100)		
Department	Varchar(30)		
PhoneNo	Varchar(20)		(61)0444111222
Address	Varchar(100)		
OrganId	Unsigned Int(11)	FK	
Active	Unsigned Int(1)		1: Confirmed as an existing employer 0: is not

Jina

The Employer table has primary key. However, the address is not atomic. This table needs to be split. An address is composed of unit and street address, suburb, state, country and postcode. To prevent transitive-dependency on the primary key, it is split to three tables, PostGlobal, PostLocal and Employer.

2.1 PostGlobal: uses the same table as 1.1 PostGlobal
2.2 PostLocal: uses the same table as 1.2 PostLocal
2.3 Employer

Name	Type	Key	Comment
EmployerId	Unsigned Int(11)	PK	
Name	Varchar(30)		
Email	Varchar(30)	U	Login ID
Password	Varchar(100)		
Department	Varchar(30)		
PhoneNo	Varchar(20)		(61)0444111222
Address	Varchar(30)		Unit number and Street name
PostLocalId	Unsigned Int(11)	FK	
OrganId	Unsigned Int(11)	FK	
Active	Unsigned Int(1)		Confirmed as an existing employer

1NF: Satisfied
2NF: Satisfied
3NF: Satisfied

3. **Job**

Name	Type	Key	Comment
JobId	Unsigned Int(11)	PK	
Title	Varchar(30)		
Description	Varchar(500)		
Responsibilities	Varchar(500)		
YearOfExperience	Unsigned Int(2)		
SalaryMin	Unsigned Int(7)		
SalaryMax	Unsigned Int(7)		
PostLocalId	Unsigned Int(11)		Suburb (Location)
ContractType	Varchar(30)		Casual, Permanent
StartDate	Date		Advertisement start date
ExpiryDate	Date		Advertisement end date
EmployerId	Unsigned Int(11)	FK	

1NF: Satisfied
2NF: Satisfied
3NF: Satisfied

4. **Seeker**

Name	Type	Key	Comment
SeekerId	Unsigned Int(11)	PK	
Name	Varchar(30)		
Email	Varchar(30)	U	Login ID
Password	Varchar(100)		
PhoneNo	Varchar(20)		
Address	Varchar(100)		
YearOfExperience	Unsigned Int(2)		
CVFile	Varchar(100)		
CoverletterFile	Varchar(100)		
JoinedDate	Date		

The Seeker table has primary key. However, the address is not atomic. This table needs to be split. An address is composed of unit and street address, suburb, state, country and postcode.

To prevent transitive-dependency on the primary key, it is split to three tables, PostGlobal, PostLocal and Seeker.

4.1 PostGlobal: uses the same table as 1.1 PostGlobal
4.2 PostLocal: uses the same table as 1.2 PostLocal
4.3 Seeker

Name	Type	Key	Comment
SeekerId	Unsigned Int(11)	PK	
Name	Varchar(30)		
Email	Varchar(30)	U	Login ID
Password	Varchar(100)		
PhoneNo	Varchar(20)		
Address	Varchar(100)		Unit number and street name
PostLocalID	Unsigned Int(11)	FK	
YearOfExperience	Unsigned Int(2)		
CVFile	Varchar(100)		
CoverletterFile	Varchar(100)		
JoinedDate	Date		

5. **SkillArea**

Name	Type	Key	Comment
AreaId	Unsigned Int(11)	PK	
Type	Varchar(30)	U	Programming Lang, CMS, DBMS

1NF: Satisfied
2NF: Satisfied
3NF: Satisfied

6. **SkillLevel**: only three records (beginner, intermediate, expert)

Name	Type	Key	Comment
SkillLevelId	Unsigned Int(1)	PK	1, 2, 3 matches level type
Type	Varchar(15)	U	Beginner, Intermediate, expert

1NF: Satisfied
2NF: Satisfied
3NF: Satisfied

7. **Skill**

Name	Type	Key	Comment
SkillId	Unsigned Int(5)	PK	
Type	Varchar(30)	U	C#, Python, Java, Javascript
Description	Varchar(200)		
AreaID	Unsigned Int(3)	FK	Programming language

1NF: Satisfied
2NF: Satisfied
3NF: Satisfied

8. **OwnedSkill**

Name	Type	Key	Comment
OwnedSkillId	Unsigned Int(11)	PK	
SeekerId	Unsigned Int(11)	FK	
SkillId	Unsigned Int(3)	FK	
SkillLevelId	Unsigned Int(1)	FK	

1NF: Satisfied

2NF: Satisfied
3NF: Satisfied

9. **RequiredSkill**

Name	Type	Key	Comment
OwnedSkillId	Unsigned Int(11)	PK	
SeekerId	Unsigned Int(11)	FK	
SkillId	Unsigned Int(3)	FK	
LevelId	Unsigned Int(1)	FK	

1NF: Satisfied
2NF: Satisfied
3NF: Satisfied

10. **QualificationDegree**

Name	Type	Key	Comment
DegreeId	Unsigned Int(11)	PK	1, 2, 3 matches to level type
Type	Varchar(20)		Cert III, IV, Diploma, Bachelor

1NF: Satisfied
2NF: Satisfied
3NF: Satisfied

11. **Qualification**

Name	Type	Key	Comment
QualId	Unsigned Int(11)	PK	
Area	Varchar(20)		Web Dev, Computer science, IT

1NF: Satisfied
2NF: Satisfied
3NF: Satisfied

12. **OwnedQualification**

Name	Type	Key	Comment
OwnedQuailId	Unsigned Int(11)	PK	
SeekerId	Unsigned Int(11)	FK	
Organisation	Varchar(30)		Queensland TAFE, UQ, QUT
QualId	Unsigned Int(11)	FK	
DegreeId	Unsigned Int(11)	FK	

1NF: Satisfied
2NF: Satisfied
3NF: Satisfied

13. **RequiredQualification**

Name	Type	Key	Comment
RequiredQuailId	Unsigned Int(11)	PK	
EmployerId	Unsigned Int(11)	FK	
QualId	Unsigned Int(11)	FK	
DegreeId	Unsigned Int(11)	FK	

1NF: Satisfied
2NF: Satisfied
3NF: Satisfied

I deleted the requiredVisa table related to Employer.

14. **VisaInfo**

Name	Type	Key	Comment
VisaInfoId	Unsigned Int(11)	PK	
Type	Varchar(50)		
Description	Varchar(200)		

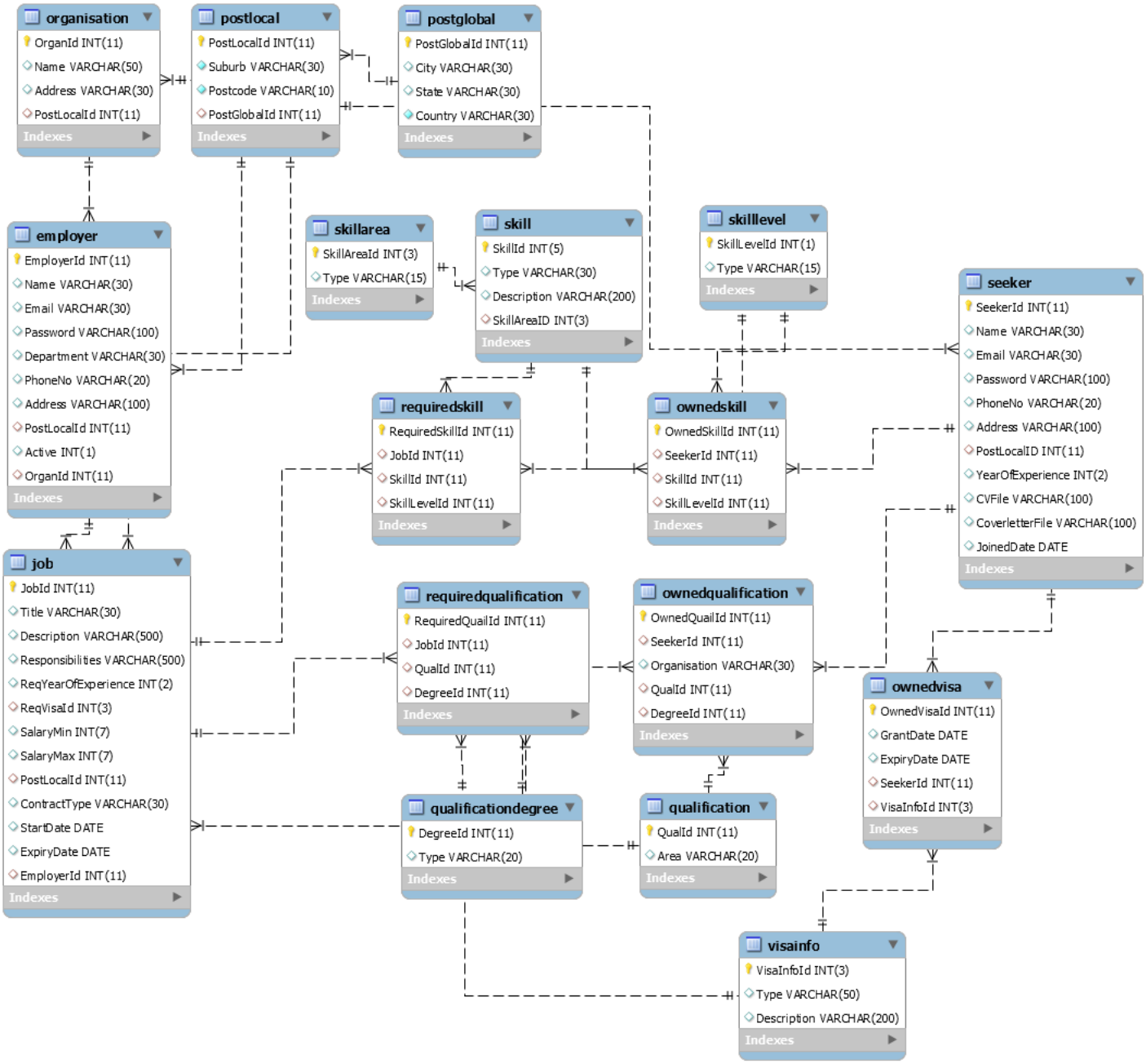
1NF: Satisfied
2NF: Satisfied
3NF: Satisfied

15. **OwnedVisa**

Name	Type	Key	Comment
OwnedVisaId	Unsigned Int(11)	PK	
GrantDate	Date		
ExpiryDate	Date		
SeekerId	Unsigned Int(11)	FK	
VisaInfoId	Unsigned Int(11)	FK	

1NF: Satisfied
2NF: Satisfied
3NF: Satisfied

Jina


organisation
OrganId INT(11)
Name VARCHAR(50)
Address VARCHAR(30)
PostLocalId INT(11)
Indexes
postlocal
PostLocalId INT(11)
Suburb VARCHAR(30)
Postcode VARCHAR(10)
PostGlobalId INT(11)
Indexes
postglobal
PostGlobalId INT(11)
City VARCHAR(30)
State VARCHAR(30)
Country VARCHAR(30)
Indexes
employer
EmployerId INT(11)
Name VARCHAR(30)
Email VARCHAR(30)
Password VARCHAR(100)
Department VARCHAR(30)
PhoneNo VARCHAR(20)
Address VARCHAR(100)
PostLocalId INT(11)
Active INT(1)
OrganId INT(11)
Indexes
skillarea
SkillAreaId INT(3)
Type VARCHAR(15)
Indexes
skill
SkillId INT(5)
Type VARCHAR(30)
Description VARCHAR(200)
SkillAreaID INT(3)
Indexes
skilllevel
SkillLevelId INT(1)
Type VARCHAR(15)
Indexes
seeker
SeekerId INT(11)
Name VARCHAR(30)
Email VARCHAR(30)
Password VARCHAR(100)
PhoneNo VARCHAR(20)
Address VARCHAR(100)
PostLocalID INT(11)
YearOfExperience INT(2)
CVFile VARCHAR(100)
CoverletterFile VARCHAR(100)
JoinedDate DATE
Indexes
requiredskill
RequiredSkillId INT(11)
JobId INT(11)
SkillId INT(11)
SkillLevelId INT(11)
Indexes
ownedskill
OwnedSkillId INT(11)
SeekerId INT(11)
SkillId INT(11)
SkillLevelId INT(11)
Indexes
job
JobId INT(11)
Title VARCHAR(30)
Description VARCHAR(500)
Responsibilities VARCHAR(500)
ReqYearOfExperience INT(2)
ReqVisaId INT(3)
SalaryMin INT(7)
SalaryMax INT(7)
PostLocalId INT(11)
ContractType VARCHAR(30)
StartDate DATE
ExpiryDate DATE
EmployerId INT(11)
Indexes
requiredqualification
RequiredQualId INT(11)
JobId INT(11)
QualId INT(11)
DegreeId INT(11)
Indexes
ownedqualification
OwnedQualId INT(11)
SeekerId INT(11)
Organisation VARCHAR(30)
QualId INT(11)
DegreeId INT(11)
Indexes
ownedvisa
OwnedVisaId INT(11)
GrantDate DATE
ExpiryDate DATE
SeekerId INT(11)
VisaInfoId INT(3)
Indexes
qualificationdegree
DegreeId INT(11)
Type VARCHAR(20)
Indexes
qualification
QualId INT(11)
Area VARCHAR(20)
Indexes
visainfo
VisaInfoId INT(3)
Type VARCHAR(50)
Description VARCHAR(200)
Indexes

Jina

a) After normalisation, create a data dictionary including correct datatypes and precision parameters and nullability, and key and other constraints. Use the template provided.

   See the separate data dictionary file.

b) Create a Data flow Diagram (DFD) for the project. You need to prepare the context level (or level 0) and level 1. Comply with the rules and guidelines of the notation chosen. Identify and document any assumptions you have made about the database project that have been incorporated into the DFD.

   1) External entities

      - employers
      - job seekers
      - System manager

   2) Actions

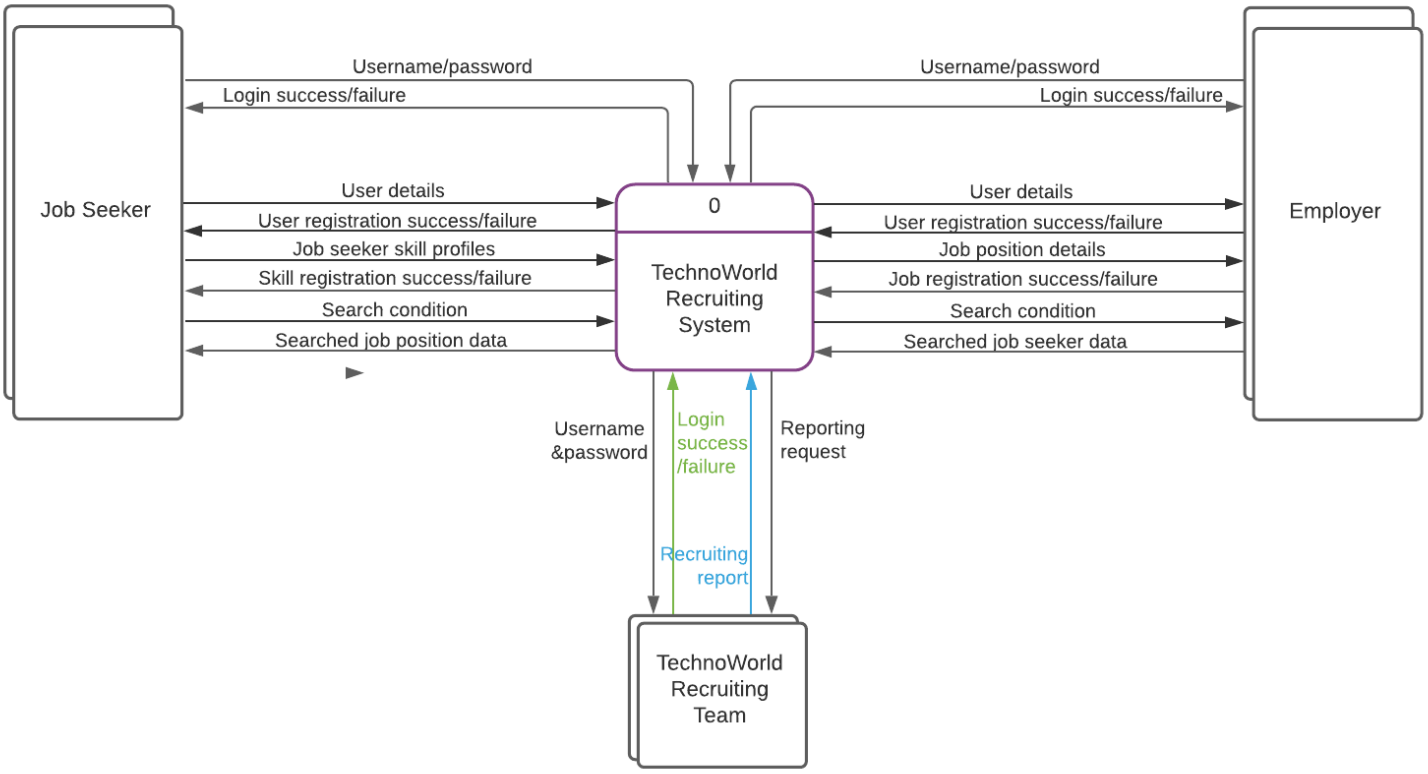
      - job seekers can
        - login
        - register account and skill profile
        - edit profile
        - search jobs

      - employers can
        - login
        - register account and job advertisement
        - manage registered jobs
        - find out best-fit candidates

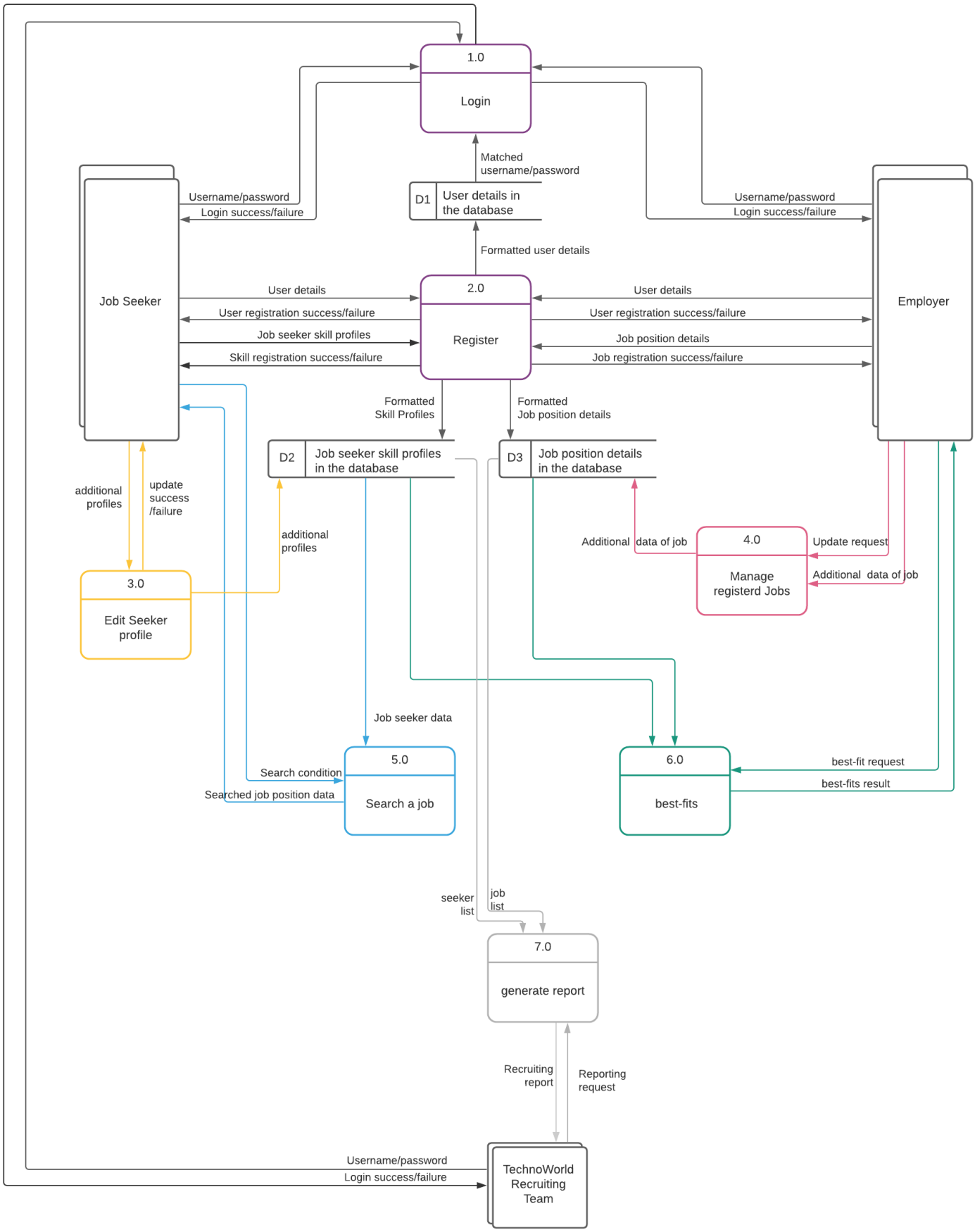
      - manager can
        - login
        - generate reports

   3) Processes
      1. Login (employer/job seeker)
      2. Register (employer/job seeker)
      3. Edit profile (job seeker)
      4. Manage jobs (Employer)
      5. Search jobs (job seeker)
      6. Find out best-fits (Employer)
      7. Generate reports (system manager)

4) Context Diagram



5) Level 1 Diagram



Jina

PART THREE – Goals, objectives and tasks

a) In preparation for the Project Plan, you need to complete the table provided to identify the clients' goals, the objectives derived from each goal and the tasks that must be completed to achieve each objective.

You need to write all requirements

GOALS	OBJECTIVES	REQUIREMENTS	TASKS
Well-organised database solution for TechnoWorld Recruiting Website	Well-organised database for the centralised management of data	*Analysis and migrate existing data*	See the project plan attached
		Design with consistency and integrity of data	
		adapting standardised code values such as post code, qualification and visa information.	
		secure and authorised access to the database	
	Automated matching functionality to find appropriate candidates	Search best-fits candidates to a given job/position.	
	User friendly interface to access the recruiting service from all over the world	Job seekers can manage their profiles easily	
		Employer can manage their personal and job information easily	

b) Create a Project Plan using acceptable organisational standards tools. The project plan will be organised into appropriate life-cycle stages. In addition to the tasks identified in the table created in point a), the plan will include all the necessary process tasks required to complete the project. The project timeline must match the one provided in the Business Requirements.

See the attached project plan file

PART FOUR – Build and test the model

In order to fix the problems of the existing architecture (refer to Business requirements) complete the following tasks:

a) Create a database prototype from the normalised tables and the data dictionary (PART 2 points (c) and (d)).

b) Populate the tables with sufficient testing data to run the queries/reports listed in the section: **Automated list of queries/reports to test prototype** included in the project requirements.

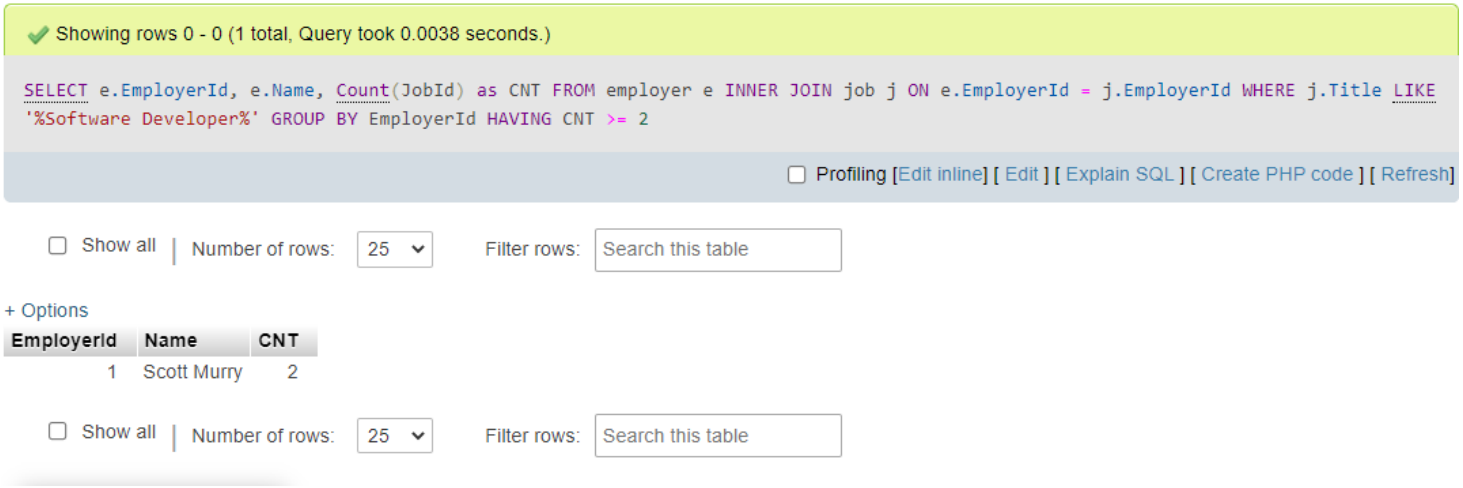
c) Export or dump the SQL file including the database structure and data.

See the exported file, "TechnoWorld.sql"

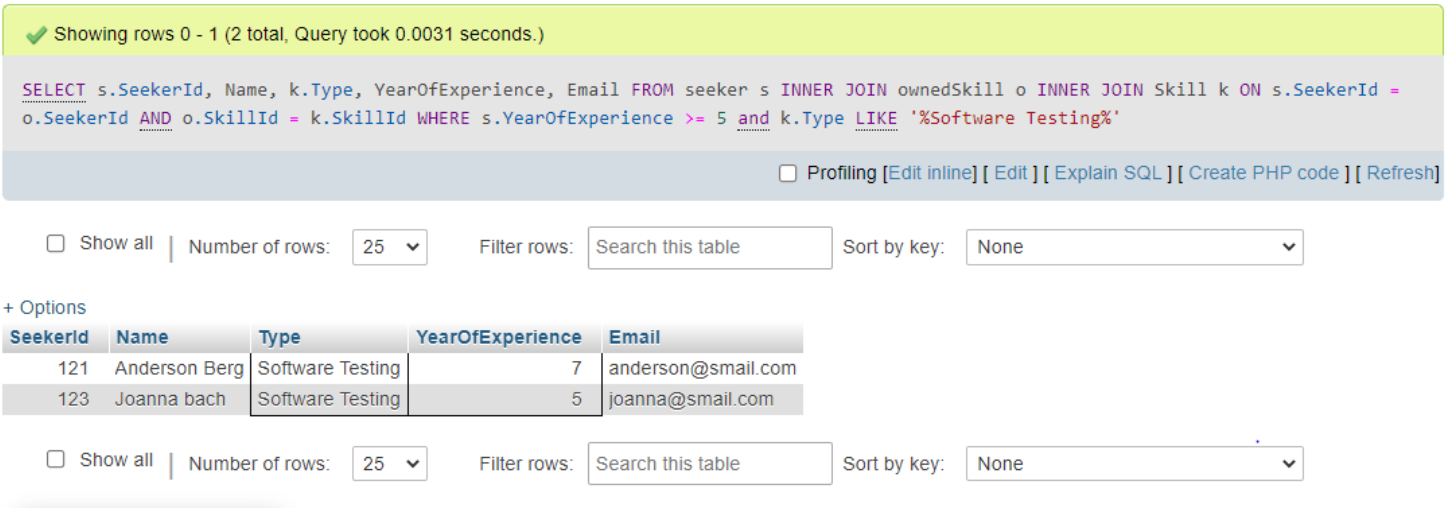
Jina

d) Create and run the queries listed in the section: **Automated list of queries/reports to test prototype** included in the project requirements.

Query 1: List all employers that have more than 2 positions available for software developers



Query 2: List all candidates that have more than 5 years' experience in software testing



Query 3: List all the positions available for software developers including the qualifications and years of experience required.

Jina

Showing rows 0 - 5 (6 total, Query took 0.0017 seconds.)

```
SELECT j.JobId, j.title, j.ReqYearOfExperience, q.Area, d.Type, j.salaryMin, j.salaryMax FROM job j INNER JOIN
RequiredQualification rq INNER JOIN Qualification q INNER JOIN QualificationDegree d ON j.JobId = rq.JobId AND rq.QualId = q.QualId
AND rq.DegreeId = d.DegreeId WHERE j.Title like '%Software Developer%'
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

JobId	title	ReqYearOfExperience	Area	Type	salaryMin	salaryMax
1	Senior Software Developer	5	Information Technolo	Bachelor	70000	80000
1	Senior Software Developer	5	Computer Science	Bachelor	70000	80000
4	Software Developer	1	Information Technolo	Diploma	60000	70000
4	Software Developer	1	Computer Science	Diploma	60000	70000
11	Junior Software Developer	1	Information Technolo	Bachelor	60000	70000
11	Junior Software Developer	1	Computer Science	Bachelor	60000	70000

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Query 4: Display the 'best fit' list of candidates for a specific database analyst position

Query 1 | SQL File 3* | seeker | ownedvisa

Limit to 1000 rows

```
53          INNER JOIN ownedVisa ov
54          ON s.SeekerId = ov.SeekerId
55      ) s
56      INNER JOIN
57      (
58          SELECT j.jobid, j.title, j.ReqYearOfExperience, j.ReqVisaId, q.QualId, d.DegreeId
59          FROM
60          ( SELECT *
61              FROM job
62              WHERE Title like '%Database Analyst%'
63              LIMIT 1  -- This can be changed into another WhERE clause, like " AND JobId = :jobid"
64          ) j
65          INNER JOIN requiredqualification rq INNER JOIN qualification q INNER JOIN qualificationdegree d
66          ON j.JobId = rq.JobId AND rq.QualId = q.QualId AND rq.DegreeId = d.DegreeId
67      ) j
68      ON s.QualId = j.QualId AND s.DegreeId >= j.DegreeId
69      WHERE s.YearOfExperience >= j.ReqYearOfExperience AND s.VisaInfoId >= j.ReqVisaId
70  ) o
71  ON s.SeekerId = o.SeekerId
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

SeekerId	Name	JobId	Title	SeekerId	Name	jobid	title
123	Joanna bach	12	Database Analyst	123	Joanna bach	12	Database Analyst

Result 1

The following figure is the whole SQL statement for query 4.

Jina

```
SELECT s.SeekerId, s.Name, s.JobId, s.Title,  o.SeekerId, o.Name, o.JobId, o.Title
FROM
( -- Compare Skill, Skill Area, Skill level between  job and seeker
    SELECT s.SeekerId, s.Name, j.JobId, j.Title, SeekerSkillCNT
    FROM
    (
        SELECT s.SeekerId, s.Name, COUNT(s.SeekerId) SeekerSkillCNT
        FROM
        (
            SELECT s.SeekerId, s.Name, a.SkillAreaId, k.SkillId, l.SkillLevelId
            FROM Seeker s INNER JOIN OwnedSkill os INNER JOIN Skill k INNER JOIN SkillArea a INNER JOIN SkillLevel l
            ON s.SeekerId = os.SeekerId AND os.SkillId = k.SkillId AND k.SkillAreaId = a.SkillAreaId AND os.SkillLevelId = l.SkillLevelId
        ) s
        RIGHT JOIN
        (
            SELECT j.jobid, j.title, a.SkillAreaId, k.SkillId, l.SkillLevelId
            FROM
            ( SELECT *
                FROM job
                WHERE Title like '%Database Analyst%'
                LIMIT 1  -- This can be changed into another WhERE clause, like " AND JobId = :jobid"
            ) j
            INNER JOIN RequiredSkill r INNER JOIN Skill k INNER JOIN SkillArea a INNER JOIN SkillLevel l
                ON j.JobId = r.JobId AND r.SkillId = k.SkillId AND k.SkillAreaId = a.SkillAreaId AND r.SkillLevelId = l.SkillLevelId
        ) j
        ON s.SkillAreaId = j.SkillAreaId AND s.SkillId = j.SkillId AND s.SkillLevelId >= j.SkillLevelId
        GROUP BY s.SeekerID
    ) s
    INNER JOIN
    (
        SELECT COUNT(j.jobid) JobSkillCNT, j.jobid, j.title
        FROM
        ( SELECT *
            FROM job
            WHERE Title like '%Database Analyst%'
            LIMIT 1  -- This can be changed into another WhERE clause, like " AND JobId = :jobid"
        ) j
        INNER JOIN RequiredSkill r INNER JOIN Skill k INNER JOIN SkillArea a INNER JOIN SkillLevel l
            ON j.JobId = r.JobId AND r.SkillId = k.SkillId AND k.SkillAreaId = a.SkillAreaId AND r.SkillLevelId = l.SkillLevelId
    ) j
    ON s.SeekerSkillCNT = j.JobSkillCNT
) s
INNER JOIN
( -- Compare Qualification, YearOfExperience, VisaId job and seeker
    SELECT s.SeekerId, s.Name, s.YearOfExperience, s.QualId, s.DegreeId,  s.VisaInfoId,
    j.ReqYearOfExperience, j.ReqVisaId, j.JobId, j.Title
    FROM
    (
        SELECT s.SeekerId, s.Name, s.YearOfExperience, q.QualId, d.DegreeId,  ov.VisaInfoId
        FROM Seeker s INNER JOIN ownedqualification oq INNER JOIN qualification q INNER JOIN qualificationdegree d
        ON s.SeekerId = oq.SeekerId AND oq.QualId = q.QualId AND oq.DegreeId = d.DegreeId
        INNER JOIN ownedVisa ov
        ON s.SeekerId = ov.SeekerId
    ) s
    INNER JOIN
    (
        SELECT j.jobid, j.title, j.ReqYearOfExperience, j.ReqVisaId, q.QualId, d.DegreeId
        FROM
        ( SELECT *
            FROM job
            WHERE Title like '%Database Analyst%'
            LIMIT 1  -- This can be changed into another WhERE clause, like " AND JobId = :jobid"
        ) j
        INNER JOIN requiredqualification rq INNER JOIN qualification q INNER JOIN qualificationdegree d
        ON j.JobId = rq.JobId AND rq.QualId = q.QualId AND rq.DegreeId = d.DegreeId
    ) j
    ON s.QualId = j.QualId AND s.DegreeId >= j.DegreeId
    WHERE s.YearOfExperience >= j.ReqYearOfExperience AND s.VisaInfoId >= j.ReqVisaId
) o
ON s.SeekerId = o.SeekerId
```

Jina

Query 5: Display the visa status for all web designers. Include the candidate name, surname and phone number.

Query 1 | SQL File 3* | seeker | ownedvisa

Limit to 1000 rows

```
52          ON s.SeekerId = ov.SeekerId
53      ) s
54      INNER JOIN
55      (
56          SELECT j.jobid, j.title, j.ReqYearOfExperience, j.ReqVisaId, q.QualId, d.DegreeId
57          FROM
58          ( SELECT *
59              FROM job
60              WHERE Title like '%Web Design%'
61          ) j
62          INNER JOIN requiredqualification rq INNER JOIN qualification q INNER JOIN qualificationdegree d
63          ON j.JobId = rq.JobId AND rq.QualId = q.QualId AND rq.DegreeId = d.DegreeId
64      ) j
65      ON s.QualId = j.QualId AND s.DegreeId >= j.DegreeId
66      WHERE s.YearOfExperience >= j.ReqYearOfExperience AND s.VisaInfoId >= j.ReqVisaId
67  ) o
68  ON s.SeekerId = o.SeekerId AND s.jobId = o.JobId
69  INNER JOIN OwnedVisa ov INNER JOIN VisaInfo v
70  ON s.SeekerId = ov.SeekerId AND ov.VisaInfoId = v.VisaInfoId
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

SeekerId	Name	JobId	Title	VisaInfoId	VisaType	PhoneNo
111	Gilbert Hornet	9	Junior Web Designer	5	working holiday	+1)0121222111
112	Kenethe Kim	9	Junior Web Designer	7	sponser	+1)0123222333
123	Joanna bach	8	Web Designer	7	sponser	+49)043333444
123	Joanna bach	9	Junior Web Designer	7	sponser	+49)043333444

Result 4

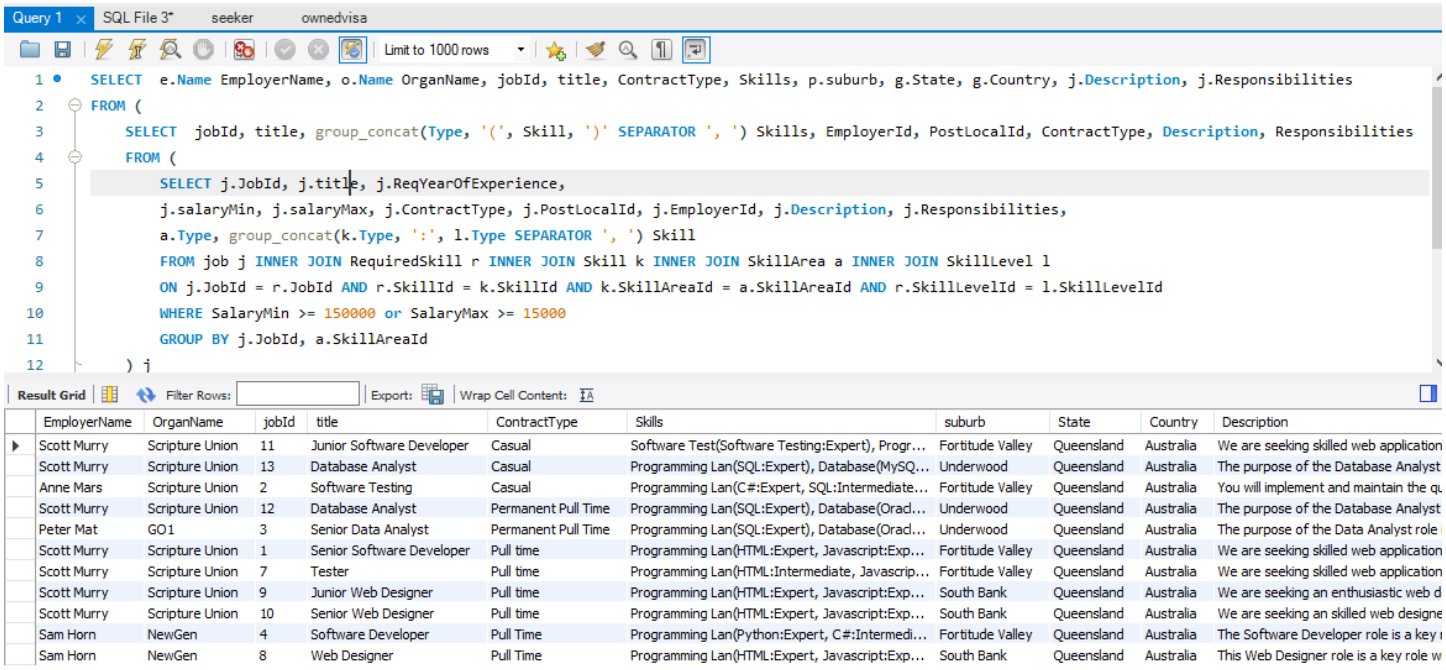
The following figure is the whole SQL statement for query 5.

Jina

```sql
SELECT s.SeekerId, s.Name, s.JobId, s.Title, v.VisaInfoId, v.Type VisaType, s.PhoneNo -- o.SeekerId, o.Name, o.JobId, o.Title
FROM
( -- Compare Skill, Skill Area, Skill level
    SELECT s.SeekerId, s.Name, j.JobId, j.Title, SeekerSkillCNT, s.PhoneNo
    FROM
    (
        SELECT s.SeekerId, s.Name, COUNT(s.SeekerId) SeekerSkillCNT, j.jobId, j.title, s.PhoneNo
        FROM
        (
            SELECT s.SeekerId, s.Name, a.SkillAreaId, k.SkillId, l.SkillLevelId, s.PhoneNo
            FROM Seeker s INNER JOIN OwnedSkill os INNER JOIN Skill k INNER JOIN SkillArea a INNER JOIN SkillLevel l
            ON s.SeekerId = os.SeekerId AND os.SkillId = k.SkillId AND k.SkillAreaId = a.SkillAreaId AND os.SkillLevelId = l.SkillLevelId
        ) s
        RIGHT JOIN
        (
            SELECT j.jobid, j.title, a.SkillAreaId, k.SkillId, l.SkillLevelId
            FROM
            ( SELECT *
                FROM job
                WHERE Title like '%Web Design%'
            ) j
            INNER JOIN RequiredSkill r INNER JOIN Skill k INNER JOIN SkillArea a INNER JOIN SkillLevel l
                ON j.JobId = r.JobId AND r.SkillId = k.SkillId AND k.SkillAreaId = a.SkillAreaId AND r.SkillLevelId = l.SkillLevelId
        ) j
        ON s.SkillAreaId = j.SkillAreaId AND s.SkillId = j.SkillId AND s.SkillLevelId >= j.SkillLevelId
        GROUP BY s.SeekerID, j.jobid
    ) s
    INNER JOIN
    (
        SELECT COUNT(j.jobid) JobSkillCNT, j.jobid, j.title
        FROM
        ( SELECT *
            FROM job
            WHERE Title like '%Web Design%'
        ) j
        INNER JOIN RequiredSkill r INNER JOIN Skill k INNER JOIN SkillArea a INNER JOIN SkillLevel l
            ON j.JobId = r.JobId AND r.SkillId = k.SkillId AND k.SkillAreaId = a.SkillAreaId AND r.SkillLevelId = l.SkillLevelId
        GROUP BY j.jobId
    ) j
    ON s.SeekerSkillCNT = j.JobSkillCNT AND s.jobId = j.jobId
) s
INNER JOIN
( -- Compare Qualification, YearOfExperience, VisaId
    SELECT s.SeekerId, s.Name, s.YearOfExperience, s.QualId, s.DegreeId,  s.VisaInfoId,
    j.ReqYearOfExperience, j.ReqVisaId, j.JobId, j.Title
    FROM
    (
        SELECT s.SeekerId, s.Name, s.YearOfExperience, q.QualId, d.DegreeId,  ov.VisaInfoId
        FROM Seeker s INNER JOIN ownedqualification oq INNER JOIN qualification q INNER JOIN qualificationdegree d
        ON s.SeekerId = oq.SeekerId AND oq.QualId = q.QualId AND oq.DegreeId = d.DegreeId
        INNER JOIN ownedVisa ov
        ON s.SeekerId = ov.SeekerId
    ) s
    INNER JOIN
    (
        SELECT j.jobid, j.title, j.ReqYearOfExperience, j.ReqVisaId, q.QualId, d.DegreeId
        FROM
        ( SELECT *
            FROM job
            WHERE Title like '%Web Design%'
        ) j
        INNER JOIN requiredqualification rq INNER JOIN qualification q INNER JOIN qualificationdegree d
        ON j.JobId = rq.JobId AND rq.QualId = q.QualId AND rq.DegreeId = d.DegreeId
    ) j
    ON s.QualId = j.QualId AND s.DegreeId >= j.DegreeId
    WHERE s.YearOfExperience >= j.ReqYearOfExperience AND s.VisaInfoId >= j.ReqVisaId
) o
ON s.SeekerId = o.SeekerId AND s.jobId = o.JobId
INNER JOIN OwnedVisa ov INNER JOIN VisaInfo v
ON s.SeekerId = ov.SeekerId AND ov.VisaInfoId = v.VisaInfoId
```

Jina

Query 6: List the position title, description, responsibilities, skills requirements, and years of experience, salary range, location, and contract type for all positions where the salary is over $150 000. Group the results by position type (contract type: Full-time, Casual, etc).



```
SELECT  e.Name EmployerName, o.Name OrganName, jobId, title, ContractType, Skills, p.suburb, g.State, g.Country, j.Description, j.Responsibilities
FROM (
    SELECT  jobId, title, group_concat(Type, '(', Skill, ')' SEPARATOR ', ') Skills, EmployerId, PostLocalId, ContractType, Description, Responsibilities
    FROM (
        SELECT j.JobId, j.title, j.ReqYearOfExperience,
        j.salaryMin, j.salaryMax, j.ContractType, j.PostLocalId, j.EmployerId, j.Description, j.Responsibilities,
        a.Type, group_concat(k.Type, ':', l.Type SEPARATOR ', ') Skill
        FROM job j INNER JOIN RequiredSkill r INNER JOIN Skill k INNER JOIN SkillArea a INNER JOIN SkillLevel l
        ON j.JobId = r.JobId AND r.SkillId = k.SkillId AND k.SkillAreaId = a.SkillAreaId AND r.SkillLevelId = l.SkillLevelId
        WHERE SalaryMin >= 150000 or SalaryMax >= 15000
        GROUP BY j.JobId, a.SkillAreaId
    ) j
```

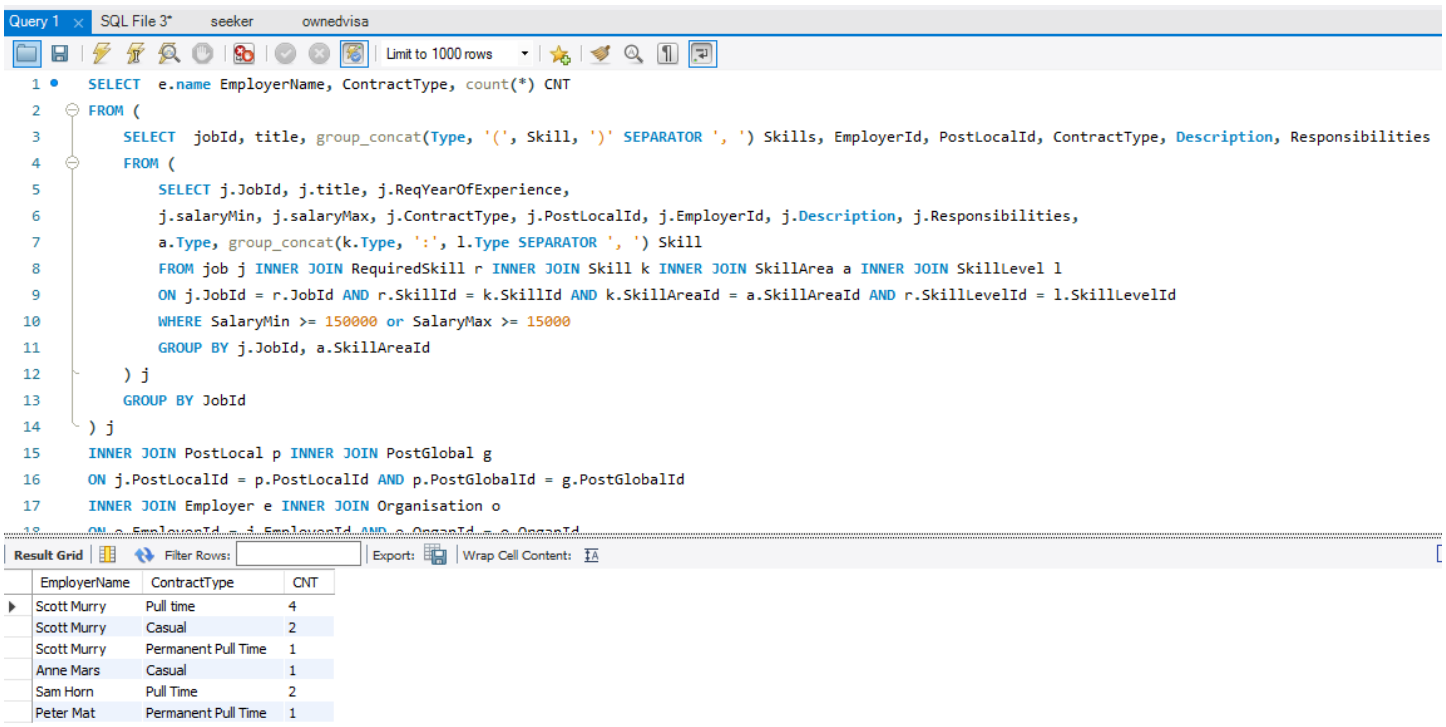
EmployerName	OrganName	jobId	title	ContractType	Skills	suburb	State	Country	Description
Scott Murry	Scripture Union	11	Junior Software Developer	Casual	Software Test(Software Testing:Expert), Progr...	Fortitude Valley	Queensland	Australia	We are seeking skilled web application
Scott Murry	Scripture Union	13	Database Analyst	Casual	Programming Lan(SQL:Expert), Database(MySQ...	Underwood	Queensland	Australia	The purpose of the Database Analyst
Anne Mars	Scripture Union	2	Software Testing	Casual	Programming Lan(C#:Expert, SQL:Intermediate...	Fortitude Valley	Queensland	Australia	You will implement and maintain the qu
Scott Murry	Scripture Union	12	Database Analyst	Permanent Pull Time	Programming Lan(SQL:Expert), Database(Oracl...	Underwood	Queensland	Australia	The purpose of the Database Analyst
Peter Mat	GO1	3	Senior Data Analyst	Permanent Pull Time	Programming Lan(SQL:Expert), Database(Oracl...	Underwood	Queensland	Australia	The purpose of the Data Analyst role
Scott Murry	Scripture Union	1	Senior Software Developer	Pull time	Programming Lan(HTML:Expert, Javascript:Exp...	Fortitude Valley	Queensland	Australia	We are seeking skilled web application
Scott Murry	Scripture Union	7	Tester	Pull time	Programming Lan(HTML:Intermediate, Javascrip...	Fortitude Valley	Queensland	Australia	We are seeking skilled web application
Scott Murry	Scripture Union	9	Junior Web Designer	Pull time	Programming Lan(HTML:Expert, Javascript:Exp...	South Bank	Queensland	Australia	We are seeking an enthusiastic web d
Scott Murry	Scripture Union	10	Senior Web Designer	Pull time	Programming Lan(HTML:Expert, Javascript:Exp...	South Bank	Queensland	Australia	We are seeking an skilled web designe
Sam Horn	NewGen	4	Software Developer	Pull Time	Programming Lan(Python:Expert, C#:Intermedi...	Fortitude Valley	Queensland	Australia	The Software Developer role is a key
Sam Horn	NewGen	8	Web Designer	Pull Time	Programming Lan(HTML:Expert, Javascript:Exp...	South Bank	Queensland	Australia	This Web Designer role is a key role w

The following figure is the whole SQL statement for query 6.

```
SELECT  e.Name EmployerName, o.Name OrganName, jobId, title, ContractType, Skills, p.suburb, g.State, g.Country, j.Description, j.Responsibilities
FROM (
    SELECT  jobId, title, group_concat(Type, '(', Skill, ')' SEPARATOR ', ') Skills, EmployerId, PostLocalId, ContractType, Description, Responsibilities
    FROM (
        SELECT j.JobId, j.title, j.ReqYearOfExperience,
        j.salaryMin, j.salaryMax, j.ContractType, j.PostLocalId, j.EmployerId, j.Description, j.Responsibilities,
        a.Type, group_concat(k.Type, ':', l.Type SEPARATOR ', ') Skill
        FROM job j INNER JOIN RequiredSkill r INNER JOIN Skill k INNER JOIN SkillArea a INNER JOIN SkillLevel l
        ON j.JobId = r.JobId AND r.SkillId = k.SkillId AND k.SkillAreaId = a.SkillAreaId AND r.SkillLevelId = l.SkillLevelId
        WHERE SalaryMin >= 150000 or SalaryMax >= 15000
        GROUP BY j.JobId, a.SkillAreaId
    ) j
    GROUP BY JobId
) j
INNER JOIN PostLocal p INNER JOIN PostGlobal g  ON j.PostLocalId = p.PostLocalId AND p.PostGlobalId = g.PostGlobalId
INNER JOIN Employer e INNER JOIN Organisation o ON e.EmployerId = j.EmployerId AND e.OrganId = o.OrganId
ORDER BY ContractType
```

Jina

6.1 This query count per group of an employer and contract type for Query6.

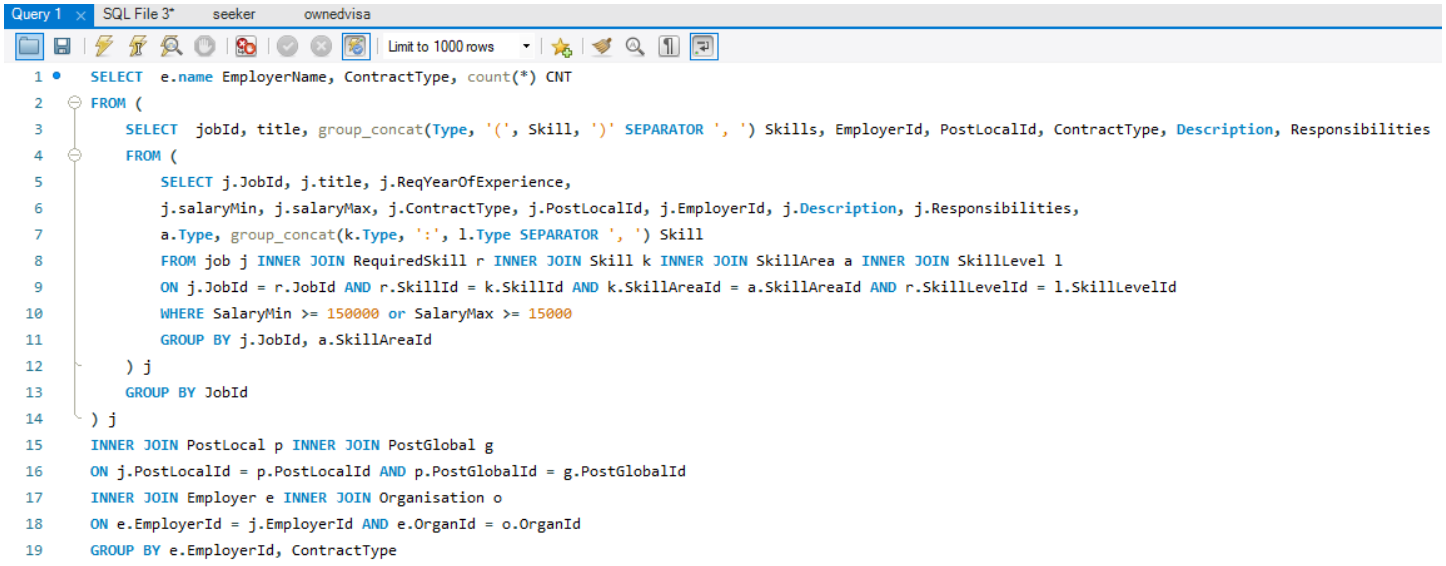


```
SELECT  e.name EmployerName, ContractType, count(*) CNT
FROM (
    SELECT  jobId, title, group_concat(Type, '(', Skill, ')' SEPARATOR ', ') Skills, EmployerId, PostLocalId, ContractType, Description, Responsibilities
    FROM (
        SELECT j.JobId, j.title, j.ReqYearOfExperience,
        j.salaryMin, j.salaryMax, j.ContractType, j.PostLocalId, j.EmployerId, j.Description, j.Responsibilities,
        a.Type, group_concat(k.Type, ':', l.Type SEPARATOR ', ') Skill
        FROM job j INNER JOIN RequiredSkill r INNER JOIN Skill k INNER JOIN SkillArea a INNER JOIN SkillLevel l
        ON j.JobId = r.JobId AND r.SkillId = k.SkillId AND k.SkillAreaId = a.SkillAreaId AND r.SkillLevelId = l.SkillLevelId
        WHERE SalaryMin >= 150000 or SalaryMax >= 15000
        GROUP BY j.JobId, a.SkillAreaId
    ) j
    GROUP BY JobId
) j
INNER JOIN PostLocal p INNER JOIN PostGlobal g
ON j.PostLocalId = p.PostLocalId AND p.PostGlobalId = g.PostGlobalId
INNER JOIN Employer e INNER JOIN Organisation o
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

EmployerName	ContractType	CNT
Scott Murry	Pull time	4
Scott Murry	Casual	2
Scott Murry	Permanent Pull Time	1
Anne Mars	Casual	1
Sam Horn	Pull Time	2
Peter Mat	Permanent Pull Time	1

The following figure is the whole SQL statement for query 6.1.



```
SELECT  e.name EmployerName, ContractType, count(*) CNT
FROM (
    SELECT  jobId, title, group_concat(Type, '(', Skill, ')' SEPARATOR ', ') Skills, EmployerId, PostLocalId, ContractType, Description, Responsibilities
    FROM (
        SELECT j.JobId, j.title, j.ReqYearOfExperience,
        j.salaryMin, j.salaryMax, j.ContractType, j.PostLocalId, j.EmployerId, j.Description, j.Responsibilities,
        a.Type, group_concat(k.Type, ':', l.Type SEPARATOR ', ') Skill
        FROM job j INNER JOIN RequiredSkill r INNER JOIN Skill k INNER JOIN SkillArea a INNER JOIN SkillLevel l
        ON j.JobId = r.JobId AND r.SkillId = k.SkillId AND k.SkillAreaId = a.SkillAreaId AND r.SkillLevelId = l.SkillLevelId
        WHERE SalaryMin >= 150000 or SalaryMax >= 15000
        GROUP BY j.JobId, a.SkillAreaId
    ) j
    GROUP BY JobId
) j
INNER JOIN PostLocal p INNER JOIN PostGlobal g
ON j.PostLocalId = p.PostLocalId AND p.PostGlobalId = g.PostGlobalId
INNER JOIN Employer e INNER JOIN Organisation o
ON e.EmployerId = j.EmployerId AND e.OrganId = o.OrganId
GROUP BY e.EmployerId, ContractType
```

e) Using the template provided, create an appropriate number of tests and corresponding test cases to test the prototype. Use all sections of the template.

See the attached test file.

f) Prepare a presentation for the client/teacher and class members/stakeholders outlining the proposed modelling solution, the prototype and its suitability to meet client requirements. If necessary, incorporate changes suggested by the stakeholders.

Jina

See the attached presentation file

g) Submit final completed project deliverables for sign-off (grading)

User	Actions	Comments
Employer	Register	Register with Organisation name, address, Employer name, phone number and email address
	Create job	Create a job with required skills, the least required year of experience, required qualification and required visa for working.
	Update job	details as above
	Delete job	
	Best-fits	Find out best-fit candidates that match a job.
Job Seeker	Register profiles	Register their personal profiles including qualification and skill
	Manage profiles	
	Search jobs/positions	
Admin	Generate reports	Report including statistics

Reference

[1] "Working In Australia". 2021. Immi.Homeaffairs.Gov.Au. https://immi.homeaffairs.gov.au/visas/working-in-australia.