Faculty of
Computers and Artificial Intelligence

Cairo University

IS211 - Introduction to Database Systems

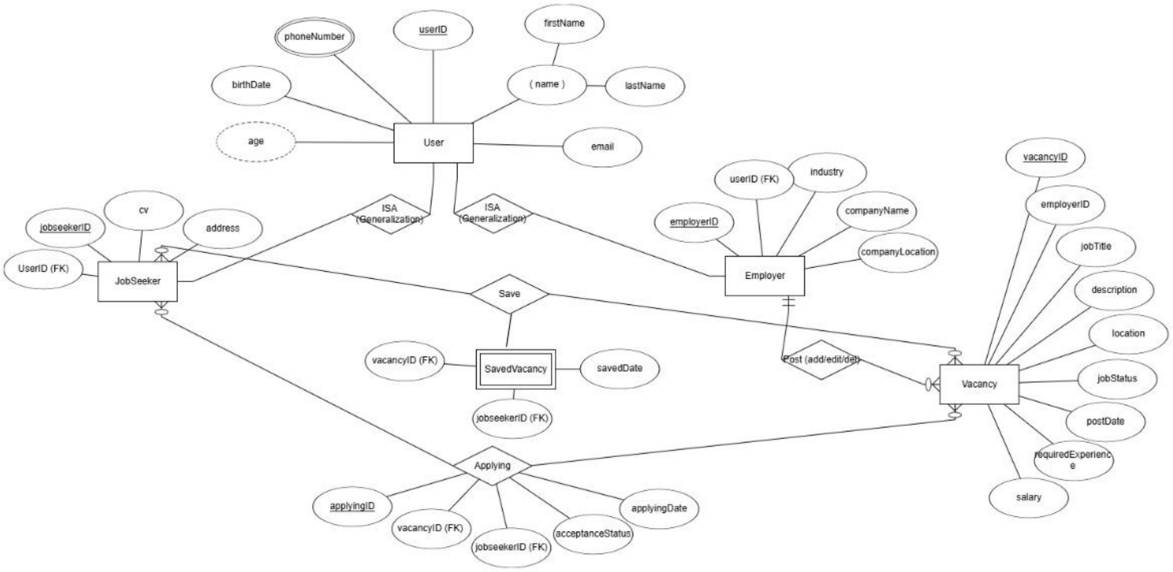
Level: 2

Assignment 1 - Phase 2

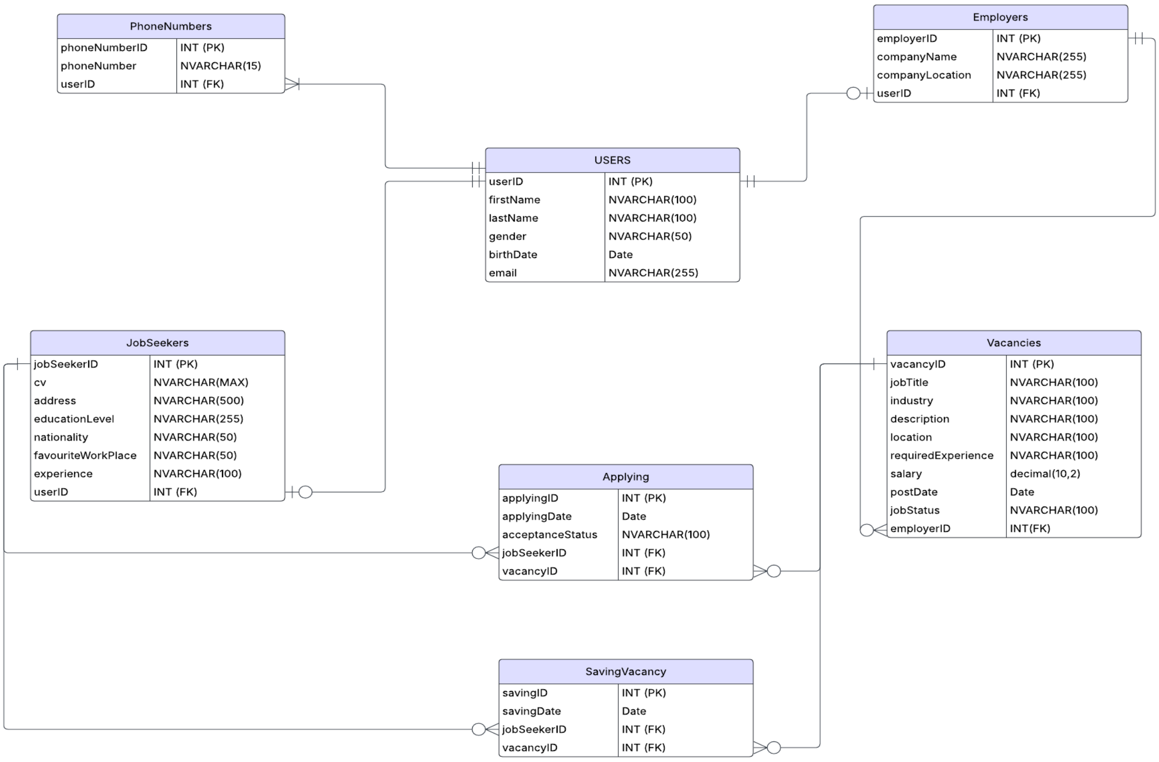
Section: S9
Project title: Online Recruitment Project
Project ID: 61
Program: General

Name	ID
Adam Mohamed Gamil	20230041
Youssef Amr Mohamed	20230502
Hazem Ahmed Mohamed	20230114
Ahmed Reda Hamed	20230019
Fatma Ali Abdel Aty	20230282

ERD diagram:



Relational schema:



PhoneNumbers

phoneNumberID	INT (PK)
phoneNumber	NVARCHAR(15)
userID	INT (FK)

Employers

employerID	INT (PK)
companyName	NVARCHAR(255)
companyLocation	NVARCHAR(255)
userID	INT (FK)

USERS

userID	INT (PK)
firstName	NVARCHAR(100)
lastName	NVARCHAR(100)
gender	NVARCHAR(50)
birthDate	Date
email	NVARCHAR(255)

JobSeekers

jobSeekerID	INT (PK)
cv	NVARCHAR(MAX)
address	NVARCHAR(500)
educationLevel	NVARCHAR(255)
nationality	NVARCHAR(50)
favouriteWorkPlace	NVARCHAR(50)
experience	NVARCHAR(100)
userID	INT (FK)

Vacancies

vacancyID	INT (PK)
jobTitle	NVARCHAR(100)
industry	NVARCHAR(100)
description	NVARCHAR(100)
location	NVARCHAR(100)
requiredExperience	NVARCHAR(100)
salary	decimal(10,2)
postDate	Date
jobStatus	NVARCHAR(100)
employerID	INT(FK)

Applying

applyingID	INT (PK)
applyingDate	Date
acceptanceStatus	NVARCHAR(100)
jobSeekerID	INT (FK)
vacancyID	INT (FK)

SavingVacancy

savingID	INT (PK)
savingDate	Date
jobSeekerID	INT (FK)
vacancyID	INT (FK)

Entities and attributes:

1- User:

1-UserID (pk)

2-Name (composite attribute):

FirstName

LastName

3-PhoneNumber (multivalued attribute)

4-email

5- birthdate

6- age (derived attribute)

2- Jobseeker:

1-cv

2-jobseekerID (pk)

3- address

4- education level

5- nationality

6- favorite work place

7- experience

8- userID (fk)

3- Employer:

1-EmployerID (pk)

2-company name

3- company’s location

4- industry

5- userID (fk)

4- Vacancy:

1- vacancyID (pk)

2- employerID (fk)

3-jobTitle

4-description

5-location

6-requiredExperience

7-salary

8-postDate

9-jobStatus

10- industry

5- Application:

1-applicationID(pk)

2-vacancyID (fk)

3-jobseekerID(fk)

4-appliedDate

5-acceptanceStatus

6- SavedVacancy:

Both 2 fk will work as primary key for this table

1-vacancyID (fk)

2-jobseekerID(fk)

3-savedDate

Physical mode:

```
CREATE TABLE [USERS] (
        userID INT IDENTITY(1,1) PRIMARY KEY,
        firstName NVARCHAR(100)  NOT NULL,
        lastName NVARCHAR(100) NOT NULL,
        gender NVARCHAR(50) NOT NULL,
        birthDate DATE,
        email NVARCHAR(255) NOT NULL
);

CREATE TABLE PhoneNumbers(

        phoneNumberID INT IDENTITY(1,1) PRIMARY KEY,
        phoneNumber NVARCHAR(15),
        userID INT,
        FOREIGN KEY (userID) REFERENCES [USERS](userID)

);
```

```
CREATE TABLE JobSeekers(
	jobSeekerID INT IDENTITY(1,1) PRIMARY KEY,
	cv NVARCHAR(MAX) NOT NULL,
	[address] NVARCHAR(500),
	educationLevel NVARCHAR(255) NOT NULL,
	nationality NVARCHAR(50) NOT NULL,
	favouriteWorkPlace NVARCHAR(50) NOT NULL,
	experience NVARCHAR(100) NOT NULL,
	userID INT,

	FOREIGN KEY (userID) REFERENCES [USERS](userID)
);

CREATE TABLE Employers(

	employerID INT IDENTITY(1,1) PRIMARY KEY,
	companyName NVARCHAR(255) NOT NULL,
	companyLocation NVARCHAR(255),
	userID INT,
	FOREIGN KEY (userID) REFERENCES [USERS](userID)
);
```

```
CREATE TABLE Vacancies(
	vacancyID INT IDENTITY(1,1) PRIMARY KEY,
	jobTitle NVARCHAR(100)  NOT NULL,
	industry NVARCHAR(100)  NOT NULL,
	[description] NVARCHAR(1000)  NOT NULL,
	[location] NVARCHAR(100)  NOT NULL,
	requiredExperience NVARCHAR(100)  NOT NULL,
	salary DECIMAL(10,2) NOT NULL,
	postDate DATE,
	jobStatus NVARCHAR(100) NOT NULL,
	employerID INT,
	FOREIGN KEY (employerID) REFERENCES Employers(employerID)
);


CREATE TABLE Applying(
	applyingID INT IDENTITY(1,1) PRIMARY KEY,
	applyingDate DATE,
	acceptanceStatus NVARCHAR(100),
	jobseekerID INT,
	vacancyID INT,

	FOREIGN KEY (jobseekerID) REFERENCES JobSeekers(jobseekerID),
	FOREIGN KEY (vacancyID) REFERENCES Vacancies(vacancyID)
);
```

```
CREATE TABLE SavingVacancy(
            savingID INT IDENTITY(1,1) PRIMARY KEY,
            savingDate DATE,
            jobseekerID INT,
            vacancyID INT,

    FOREIGN KEY (jobseekerID) REFERENCES JobSeekers(jobseekerID),
    FOREIGN KEY (vacancyID) REFERENCES Vacancies(vacancyID)
);
```

Answering requirements’ questions:

a. What was the most interesting job “title” that had maximum number of applicants?

```
SELECT TOP 1 jobTitle, COUNT(Applying.applyingID) AS Total_Applies
FROM Vacancies
INNER JOIN  Applying
ON Vacancies.vacancyID = Applying.vacancyID
GROUP BY jobTitle
ORDER BY Total_Applies DESC;
```

b. What was the announced job “title” that hadn’t any applicants last month?

```
SELECT TOP 1 jobTitle, COUNT(Applying.applyingID) AS Total_Applies

FROM Vacancies

LEFT JOIN  Applying

ON Vacancies.vacancyID = Applying.vacancyID

AND Applying.applyingDate >= DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE())-1, 0)

AND Applying.applyingDate <  DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE()), 0)

GROUP BY jobTitle

HAVING  COUNT(Applying.applyingID) = 0

ORDER BY Total_Applies ASC;
```

c. Who was the employer with the maximum announcements last month ?

```
SELECT TOP 1 firstName, lastName, COUNT(Vacancies.employerID) AS Total_Announcements

FROM USERS

INNER JOIN Employers

ON USERS.userID = Employers.userID

LEFT JOIN Vacancies

ON Employers.employerID = Vacancies.employerID

AND Vacancies.postDate >= DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE())-1, 0)

AND Vacancies.postDate <  DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE()), 0)

GROUP BY firstName, lastName

ORDER BY Total_Announcements DESC;
```

d. Who were the employers didn’t announce any job last month?

```
SELECT USERS.firstName, USERS.lastName, COUNT(Vacancies.vacancyID) AS Total_Announcements

FROM USERS

INNER JOIN Employers

ON USERS.userID = Employers.userID

LEFT JOIN Vacancies

ON Employers.employerID = Vacancies.employerID

AND Vacancies.postDate >= DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE()) - 1, 0)

AND Vacancies.postDate <  DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE()), 0)

WHERE Vacancies.vacancyID IS NULL

GROUP BY USERS.firstName, USERS.lastName

ORDER BY Total_Announcements ASC;
```

e. What were the available positions at each employer last month?

```
SELECT E.employerID, U.firstName, U.lastName, V.vacancyID, V.jobTitle, V.postDate

FROM Vacancies V

INNER JOIN Employers E ON V.employerID = E.employerID

INNER JOIN USERS U ON E.userID = U.userID

WHERE V.postDate >= DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE()) - 1, 0)

  AND V.postDate < DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE()), 0)
```

f. For each seeker, retrieve all his/her information and the number of jobs he applied for

```
SELECT USERS.userID, firstName, lastName, birthDate, email, JobSeekers.jobSeekerID, JobSeekers.address, JobSeekers.cv, ISNULL(COUNT(Applying.applyingID), 0) AS total_apply

FROM USERS

INNER JOIN JobSeekers

ON USERS.userID = JobSeekers.userID

LEFT JOIN Applying

ON JobSeekers.jobSeekerID = Applying.jobseekerID

GROUP BY USERS.userID, firstName, lastName, birthDate, email, JobSeekers.jobSeekerID, JobSeekers.address, JobSeekers.cv
```

Requirements:

Entity	Functional requirement
User	• Sign up as a job seeker or employer • Log in with credentials • Update basic user details (name, email, birthdate)
JobSeeker	• Complete profile with CV and address • View and apply to vacancies • Save vacancies • View vacancies filtered by criteria
Employer	• Complete profile with company info • Add, update, or hide job vacancies • View applicants for their vacancies • Search for job seekers
Vacancy	• Added and updated by employer • Has attributes like title, location, industry, experience, salary • Can be hidden (jobStatus = hidden)
Application	• Created when a job seeker applies to a vacancy • Stores application date and status • Used to track job seeker's application history
SavingVacancy	• Created when job seeker saves a vacancy • Allows listing saved jobs later
PhoneNumber	• Add or update user's phone number