Project Design Report:

InternshipFinder Database

Group 4

March 3, 2020

Efaz Khan
Rachid Bodson
Rachel Abraham

Introduction

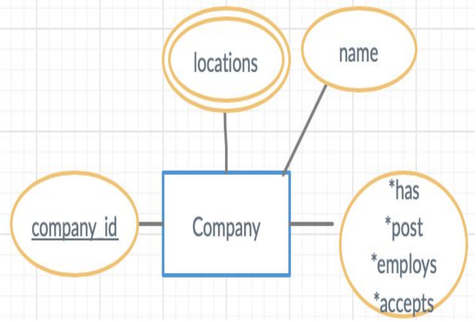
Our designed database, InternshipFinder, is about a unique environment where all kinds of internships are made easy to find without having to browse through several websites. In InternshipFinder, we gather all the best internships from different companies around the country, that are right for all students of all majors. Our motivation behind this database is that, as students, we always find it difficult to find the best internships that are right for us, we spend a vast amount of time searching in many places to find the perfect internship; therefore, to make it easier for a student to find the best internship, we created the InternshipFinder to minimize the amount of time and perhaps some frustrations that come with looking for a best internship. With this, students can devote extra time to working on their school assignments/projects. Some of the important features will include the company's information, location for the specific internship, salaries, and requirements. It will also include a brief description of the company's internship as well as the preferred majors for this company's specific internship.

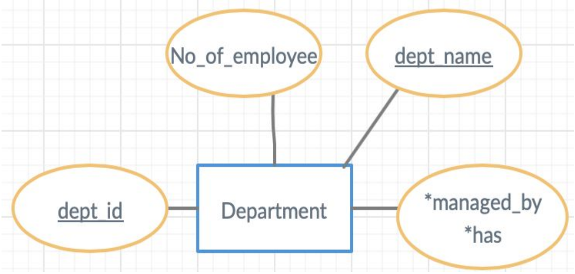
Requirements Analysis

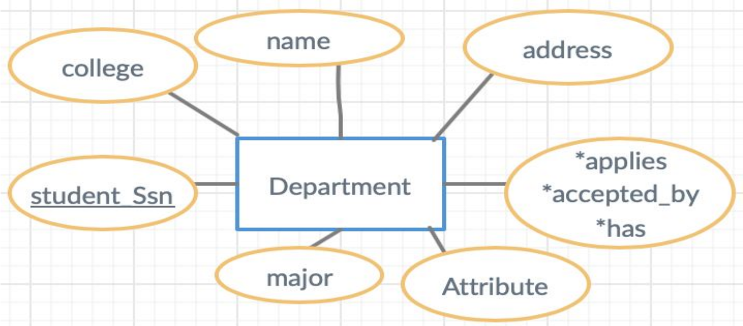
Data Requirements:

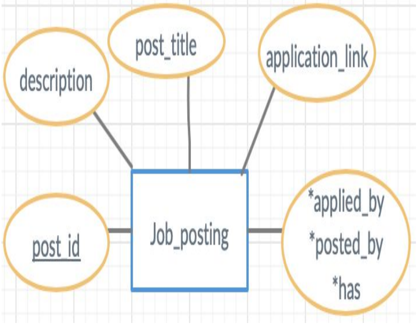
- InternshipFinder contains many companies (Ex: Google, Facebook, Microsoft, Bank of America, Home Depot etc). Each company has multiple locations, names and unique company_id.
- Each company has different departments. Departments are identified by department name(Ex: IT, HR, Research and Development, Business Development etc) , unique department id, and number of employees.
- Company posts a job_posting. Job_posting includes a unique post_id, description, application_link and post_title.
- Job_posting has several job_locations. Each job_location is associated with its unique location_id, and address (zip_code, street, state and city)
- A student can apply through the job_posting of each company.

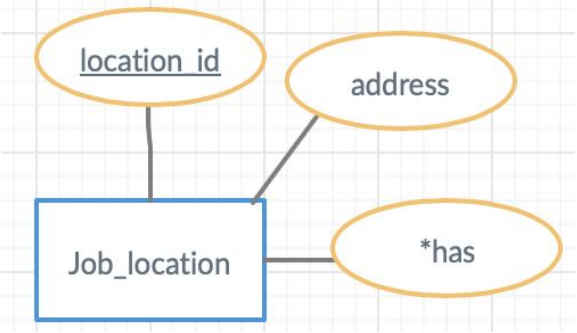
- A student can have an applied_job that stores the student_id, company_id and the post_id.
- A student can also have an apply_later jobs that stores the post_id.
- A company can accept or reject a student for an interview after reviewing their applications.
- A student has a student_id, name(Fname,Lname), gpa, college, major, and address.
- Each company employs employees and indicates the employment's date. Employees are identified by an employee_id, name(Fname,Lname), age, and employee_id.
- Employees are categorized as an intern, a manager, IT specialists, Business developers.
- Each manager manages only one department
- An Employee works on projects. Project is identified by its unique name, unique number and location.
- IT specialists have software skills.
- Business Developers have financial skills and payroll systems.

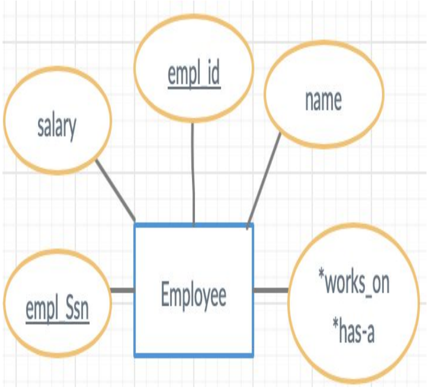


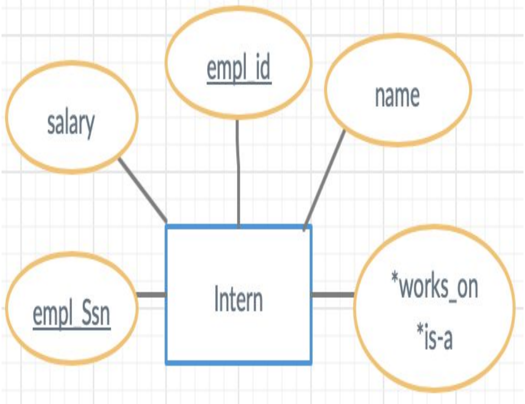


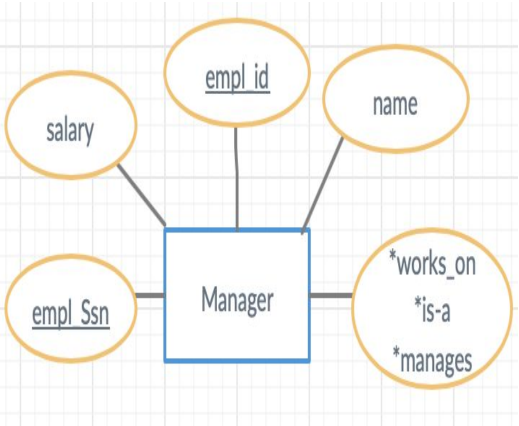


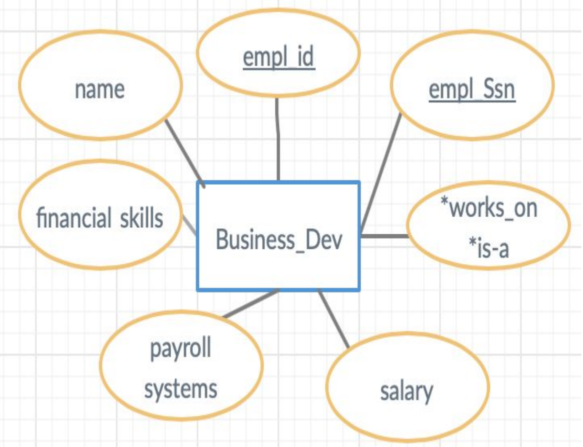

description
post_title
application_link
post id
Job_posting
*applied_by
*posted_by
*has

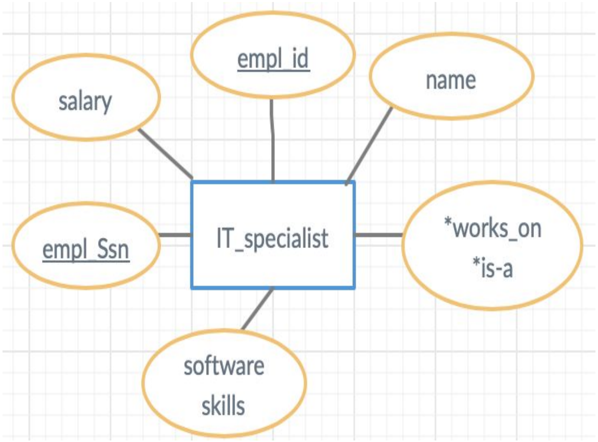

location id
address
Job_location
*has

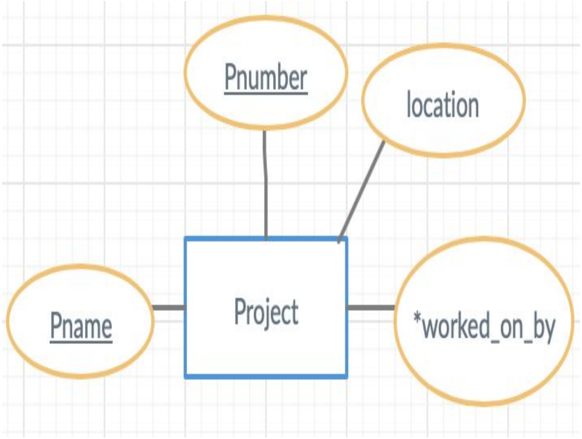

salary
empl id
name
empl Ssn
Employee
*works_on
*has-a

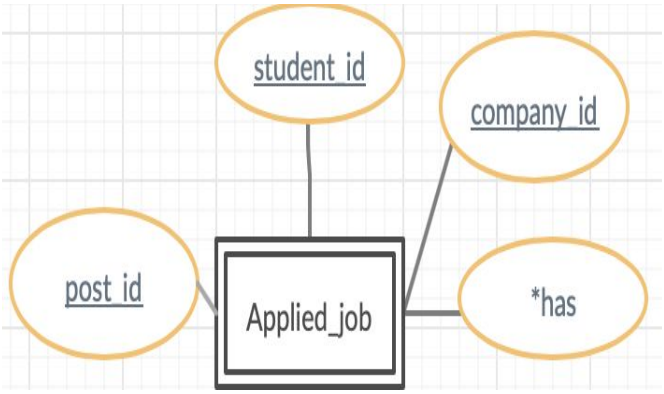

salary
empl id
name
empl Ssn
Intern
*works_on
*is-a

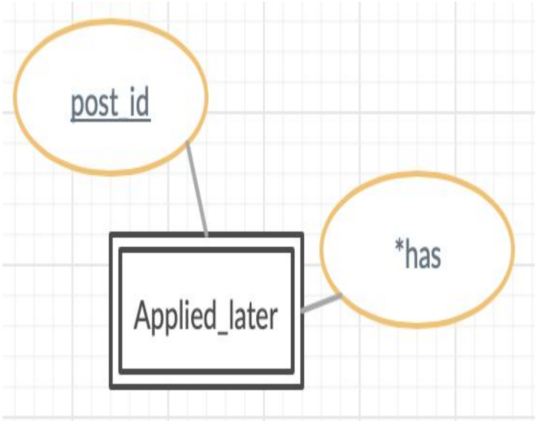

salary
empl id
name
empl Ssn
Manager
*works_on
*is-a
*manages


name
empl id
empl Ssn
financial skills
Business_Dev
*works_on
*is-a
payroll systems
salary









Functional Requirements:

- Functional Requirement 1: Students should be able to create an account with information about themselves
- Functional Requirement 2: Students should be taken to an external link when a student wants to apply to an internship
- Functional Requirement 3: Students will be able to mark what internships they have applied for.
- Functional Requirement 4: InternshipFinder should generate the number of applications per week.

- Functional Requirement 5: InternshipFinder should be able to add more companies and remove some companies (if no student has applied for the last 12 months).
- Functional Requirement 6: Students should be able to add/remove an internship to their wishlist, favorite or apply_later files.
- Functional Requirement 7: Students should be able to sort and filter through different companies
- Functional Requirement 8: InternshipFinder should generate the salary of an entry_level of all companies.
- Functional Requirement 9: InternshipFinder should be able to inform and update the application statues of a student.
- Functional Requirement 10: Student should be able to upload/remove their resume from the database
- Functional Requirement 11: Account passwords must be hashed before storing in the database for security.

Conceptual Design:

Entities and Attributes

Entity 1: Company

Description: It describes the company's information
Attributes:

- company_id_: it is the identification and also the primary key for this entity.
- name: it is the name of the company.
- location: it is a multivalued attribute describing the company's location(s).

Relationships:

- *accepts : Related to STUDENT, it indicates how many students a company accepts.
- *has : Related to DEPARTMENT, it shows how many departments a company has.
- *employs: Related to EMPLOYEE, it shows the employment of employees of a company.
- *posts : Related to JOB_POSTING, it indicates the jobs that a company posts.

Primary key:

- It is identified by company_id attribute, because it shows the uniqueness of a company.

Entity 2: Department

Description: It describes the departments in a company
Attributes:

- dept_name: it is the name of a department .
- dept_id : it is the identification of a department and also the primary key for the entity.
- Number of employees: the number of employees in a department.

Relationships:

- *is managed by: Related to MANAGER, it determines the managers of departments.
- *has: Related to COMPANY, it shows how many departments that company has.

Primary key:

- It is identified by dept_id, because each department has a unique id.

Entity 3: Student

Description: It describes the students that apply to an internship in a company.
Attributes:

- Student_id: it is the identification of a student and also the primary key of the entity.
- Name: is it the name of the student and it is also a composite attribute with first and last name.
- Gpa: it is the grade point average of the student.
- College: it is the college that the student attends.
- Major: it is the field of study of the student.
- Address: it is the address of a student.

Relationships:

- *applies: Related to JOB_POSTING, it shows how many students apply for the job of an internship.
- *accepted: Related to COMPANY, it reports the number of student acceptance of a company.
- *has: Related to APPLIED_JOB, it reports the jobs that a student has applied for.

Primary key:

- It is identified by student_id, because each student has a unique identification .

Entity 4: Job_posting

Description: It describes the job/internship that a company has posted.
Attributes:

- Post_id: it is the identification of the post and also the primary key of the entity.

- Description: it describes the job that has been posted.
- post _title: it is the title of the post.
- application _link: it is the link from which application form can be found.

Relationships:

- *has applied: Related to STUDENT, it indicates how many students have applied for a job/internship.
- *has: Related to JOB_LOCATION, it indicates how many locations that a job has.

Primary key:

- It is identified by post_id, because it separates the jobs posted by each company.

Entity 5: Job_location

Description: It describes the location of each job posted.

Attributes:

- Location_id: it is the identification of a job location and also the primary key of the entity.
- Address: it is the address of the location and it is also a composite attribute with the city, state, zip code and street.

Relationships:

- *has : Related to JOB_POSTING, it shows how many jobs a location has.

Primary key:

- It is identified location_id, because it separates the locations of jobs.

Entity 6: Employee

Description: It describes the employees who work for a company.

Attributes:

- Employee_id: it is the identification of an employee.
- Name: it is the name of an employee and it is also a composite attribute with first and last name.
- Salary: it is the amount of money that an employee makes.

Relationships:

- *employed: Related to COMPANY, it shows the number of employees employed in a company.
- *works on: Related to PROJECT, it indicates the projects that employees work on.

Primary key:

- It is identified by employee_ssn, because each employee has a unique social security number.

Entity 7: Intern

Description: It describes interns in a company

Attributes:

- Employee_id: it is the identification of an intern.
- Name: it is the name of an intern and it is also a composite attribute with first and last name.
- Salary: it is the amount of money that an intern makes (if paid intern).

Relationships:

- *is-a: Related to EMPLOYEE, it is the subclass of a superclass employee.
- *employed: Related to COMPANY, it shows the number of interns employed in a company.
- *works on: Related to PROJECT, it indicates the projects that interns work on.

Primary key:

- It is identified by employee_ssn, because each employee has a unique social security number.

Entity 8: Manager

Description: It describes the managers of a company.

Attributes:

- Employee_id: it is the identification of a manager
- Name: it is the name of a manager and it is also a composite attribute with first and last name.
- Salary: it is the amount of money that a manager makes.

Relationships:

- *is-a: Related to EMPLOYEE, it is the subclass of a superclass employee.
- *manages: Related to DEPARTMENT, it shows how many managers manage a department.
- *employed: Related to COMPANY, it shows the number of managers employed in a company.
- *works on: Related to PROJECT, it indicates the projects that managers work on.

Primary key:

- It is identified by employee_ssn, because each employee has a unique social security number.

Entity 9: Business Developers

Description: It describes the employees who deal with the business side of a company.
Attributes:

- Employee_id: it is the identification of a business developer.
- Name: it is the name of a business developer and it is also a composite attribute with first and last name.
- Salary: it is the amount of money that a business developer makes.
- Financial skills: it is a skill that a business developer has.
- Payroll systems: business developer distributes(takes charge) of the payroll.

Relationships:

- *is-a: Related to EMPLOYEE, it is the subclass of a superclass employee.
- *employed: Related to COMPANY, it shows the number of business developers employed in a company.
- *works on: Related to PROJECT, it indicates the projects that business developers work on.

Primary key:

- It is identified by <u>employee_ssn</u>, because each employee has a unique social security number.

Entity 10: IT Specialist

Description: It describes the employees who deal with information technology in a company.
Attributes:

- Employee_id: it is the identification of an IT specialist
- Name: it is the name of an IT specialist and it is also a composite attribute with first and last name.
- Salary: it is the amount of money that an IT specialist makes.
- Software skills: it is a skill that an IT specialist has.

Relationships:

- *is-a: Related to EMPLOYEE, it is the subclass of a superclass employee.
- *employed: Related to COMPANY, it shows the number of IT specialists employed in a company.
- *works on: Related to PROJECT, it indicates the projects that IT specialists work on.

Primary key:

- It is identified by <u>employee_ssn</u>, because each employee has a unique social security number.

Entity 11: Project

Description: It describes the projects that a company handles.
Attributes:

- Project_number: it is the number of a project and also the primary key of the entity.
- Name: it is the name of the project.
- Location: it is the location of the project.

Relationships:

- *worked on by: Related to EMPLOYEE, it shows the projects and the employees who worked on them .

Primary key:

- It is identified as PNo, because it indicates the uniqueness of a project.

Entity 12: Applied_job

Description: It describes the job that a student has applied for and it is also a weak entity.
Attributes:

- Student_id: it is the identification of a student.
- Company_id: it is the identification of a company.
- Post_id: it is the identification of the posted job.

Relationships:

- *has : Related to STUDENT, it shows the job has been applied by a student.

Primary key:

- It is identified by student_id and company_id, because they are both partial keys, which together form a primary key.

Entity 13: Applied_later

Description: It describes the job that a student has marked down to apply later and it is also a weak entity.
Attributes:

- Post_id: it is the identification of the posted job.

Relationships:

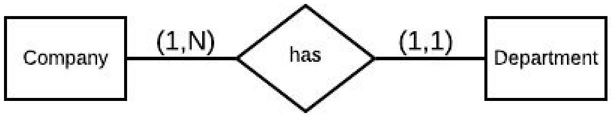
- *has : Related to STUDENT, it shows the job that will be applied later by a student.

Primary key:

- It is identified by post_id and it is also a partial key, because it is a weak entity with only one key.

Relationships

Relationship 1: Company - Department



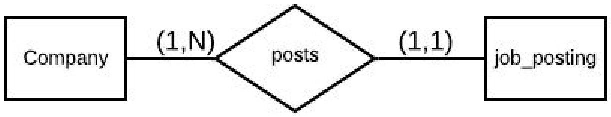
Relation: Shows the relation between company and its department

Attributes:

- Dept_id (Foreign key from company) : it is the unique identification of a department.
- Company_id (Foreign key from department) : it is the unique identification of company.

Cardinalities: the reason for the cardinalities is that we assume that at least one or at most many company(s) has at least one or at most many departments in order to be called a company.

Relationship 2: Company - job_posting



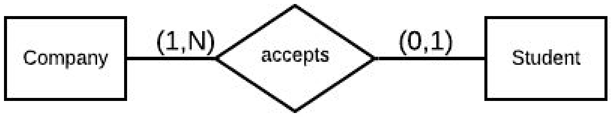
Relation: Shows the relation between a company and its job posting

Attributes:

- Post_id (Foreign key from job_posting) : it is the unique identification of a job posting
- Company_id (Foreign key from Company) : it is the unique identification of company.

Cardinalities: the reason for the cardinalities is that a company may post multiple jobs, but a specific job will be posted by one company only.

Relationship 3: Company - Student



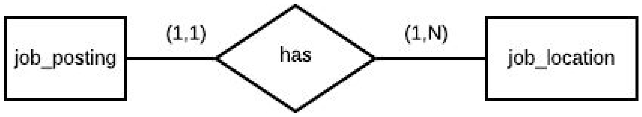
Relation: Shows the relation between a company and a student that can accept an offer

Attributes:

- Student_id (Foreign key from Student): it is the unique identification of a student.
- Company_id (Foreign key from Company): it is the unique identification of company.

Cardinalities: the reason for the cardinalities is that a company might hire one or multiple interns, but a student can accept one company only.

Relationship 4: job_posting - job_location



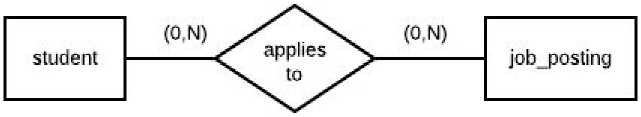
Relation: Shows the relation between a job posting and its location

Attributes:

- post_id (Foreign key from job_posting) : it is the unique identification of job posted by a company
- Location_id (Foreign key from job_location) : it is the unique identification of a job location

Cardinalities: the reason for the cardinalities is a specific job posting can be in one location only, but a location may have multiple job openings.

Relationship 5: Student - job_posting



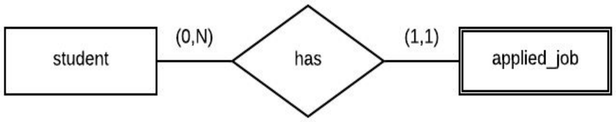
Relation: Shows the relation between a student and a job posted by a company

Attributes:

- Student_Id(Foreign key from Student) : it is the unique identification of a student.
- post_id(Foreign key from job_posting) : it is the unique identification of a job posting

Cardinalities: the reason for the cardinalities is a student can apply to zero or more companies. Similarly, a job post applied to by zero or more students.

Relationship 6: Student - applied_job



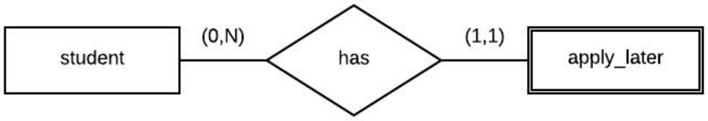
Relation: Shows the relation between a student and jobs the student has applied to.

Attributes:

- Student_id (Foreign key from Student) : it is the unique identification of a student
- Post_id (Foreign key from job_posting) : it is the unique identification of a job posting
- company_id (Foreign key from company) : it is the unique identification of company

Cardinalities: the reason for the cardinalities is that a student can apply to zero or more companies and similarly, a job can be applied by zero or more students.

Relationship 7: Student - apply_later



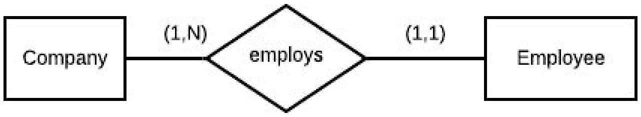
Relation: Shows the relation between student and job posts that the student plans to apply in future

Attributes:

- Student_id (Foreign key from Student) : it is the unique identification of a student
- Post_id (Foreign key from job_posting) : it is the unique identification of a job posting

Cardinalities: the reason for the cardinalities is a student can decide to add zero or more job posts to apply_later.

Relationship 8: Company-Employee



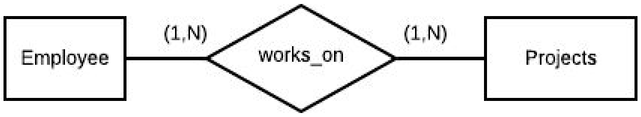
Relation: Shows the relation between company and the employees that a company employs.

Attributes:

- Company_id (Foreign key from company) : it is the unique identification of a company.
- Employee_id (Foreign key from employee) : it is the unique identification of an employee

Cardinalities: the reason for the cardinalities is that a company can employ one or many employees but exactly one employee can work for one company.

Relationship 9: Employee-Project



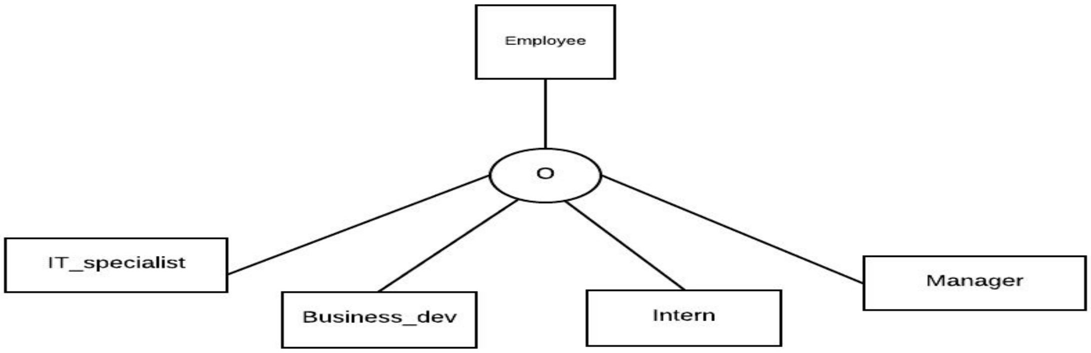
Relation: Shows the relation between employees and the projects they work on.

Attributes:

- Project_number (Foreign key from company) : it is the unique number of a project.
- Employee_id (Foreign key from employee) : it is the unique identification of an employee

Cardinalities: the reason for the cardinalities is that one or more employees can work on one or many projects.

Relationship 9: Employee-(IT_specialist, Business_developer, manager,intern)



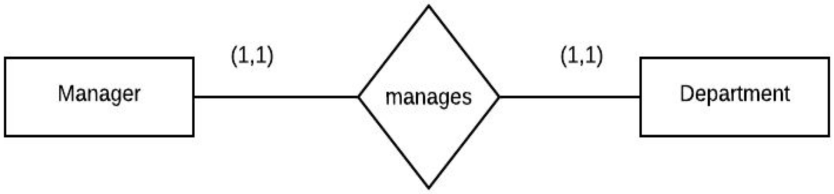
Relation: Shows the relation between employee (superclass) and it's subclasses (IT_specialist, Business_developer, manager,intern)

Attributes:

- Employee_id (Foreign key from employee) : it is the unique identification of an employee

Cardinalities: this is a specialization/generalization case, it is partial and overlapping because there might be more other categories of employees and each employee might be involved in two or more categories.

Relationship 9: Manager-Department



Relation: Shows the relation between a department and the manager who manages it.

Attributes:

- Manager_id (Foreign key from employee) : it is the unique identification of a manager.
- Department_id(Foreign key from department) : it is the unique identification of a department.

Cardinalities: the reason for the cardinalities is that exactly one manager manages exactly one department.

Relational Model

Mapping:

We started with a step one which was to map all the regular entities,STUDENT, DEPARTMENT, COMPANY, JOB_LOCATION, JOB_POSTING AND PROJECT.
In step two, we mapped the weak entities, APPLIED_JOBS AND APPLIED_LATER.
In step three, we mapped the binary 1:1 relation types; with the cardinalities of MANAGER(1,1) and DEPARTMENT(1,1) being total participation, we merged the two entities into one with all their attributes. In step four, we mapped 1:N relation types; with the cardinalities of DEPARTMENT(1,1) and COMPANY(1,N), the primary key in company becomes the foreign key in department. The same 1:N concept applies to relationships STUDENT-COMPANY, COMPANY-EMPLOYEE, JOB_POSTING-JOB_LOCATION, STUDENT-APPLIED_LATER and STUDENT-APPLY_LATER. In step five, we mapped M:N relation types; with EMPLOYEE(1,N) and PROJECT(1,N), we created a new entity as WORKS_ON with the

primary keys of both employee and project entities. The same concept goes to STUDENT-JOB_POSTING. In step six, we mapped the multivalued attributes; in our case we have location as a multivalued attribute in the company entity, therefore, we created a new entity as COMPANY_LOC with the primary key of company as a foreign key in the entity company_loc and the attribute(location) itself as a partial key. In step seven, we mapped the specialization/generalization; with EMPLOYEE being a super class and BUSINESS_DEV, IT_SPECIALIST, INTERN and MANAGER being subclasses with overlapping and partial participation, we created a relation for all the entities with their foreign keys being the primary key of the superclass(employee).

Normalization:

Data Dictionary

Table	**Attribute**	**Data Type**	**Primary Key**	**Foreign Key**	**Constraints**
STUDENT	Student_id	INT	YES		9 digits, positive
STUDENT	Fname	VARCHAR(25)			
STUDENT	Lname	VARCHAR(25)			
STUDENT	College	VARCHAR(50)			
STUDENT	Gpa	FLOAT			
STUDENT	Address	VARCHAR(100)			
STUDENT	Company_id	INT		COMPANY(Company_id)	5 digit, positive

COMPANY	Company_id	INT	YES		5 digit, positive
COMPANY	Name	VARCHAR(50)			
COMPANY_LOC	Company_id	INT		COMPANY(Company_id)	5 digit, positive
COMPANY_LOC	Location	VARCHAR(50)			
DEPARTMENT	Dept_id	INT	YES		5 digit, positive
DEPARTMENT	No_of_empl	INT			positive
DEPARTMENT	Company_id	INT		COMPANY(Company_id)	5 digit, positive
DEPARTMENT	Mgr_id	INT		EMPLOYEE(empl_id)	9 digit, positive
EMPLOYEE	Empl_id	INT	YES		9 digit, positive
EMPLOYEE	eFname	VARCHAR(25)			
EMPLOYEE	eLname	VARCHAR(25)			
EMPLOYEE	Email	VARCHAR(100)			
EMPLOYEE	Empl_title	VARCHAR(50)			
EMPLOYEE	Company_id	INT		COMPANY(Company_id)	5 digit, positive
INTERN	Empl_id	INT		EMPLOYEE(empl_id)	9 digit, positive
BUSINESS_DEV	Empl_id	INT		EMPLOYEE(empl_id)	9 digit, positive
BUSINESS_DEV	Financial_skill	VARCHAR(50)			
IT SPECIALIST	Empl_id	INT		EMPLOYEE(empl_id)	9 digit, positive
IT SPECIALIST	Software_skill	VARCHAR(50)			

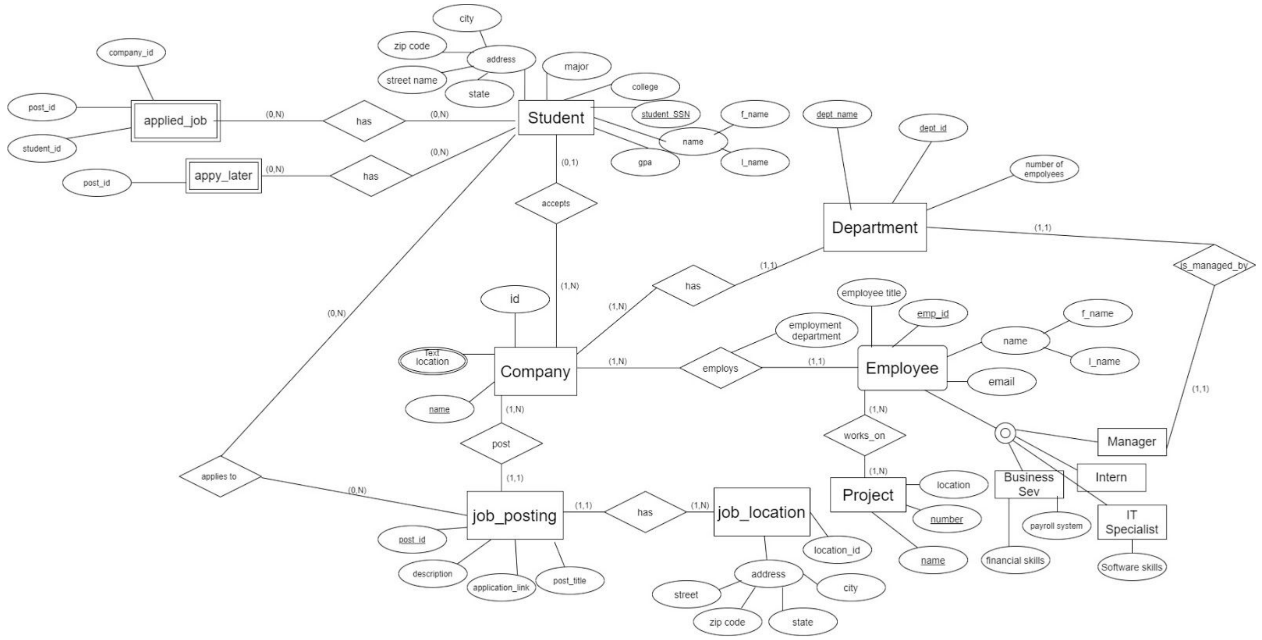
PROJECT	Pno	INT	YES		positive
PROJECT	Pname	VARCHAR(50)			
PROJECT	Plocation	VARCHAR(50)			
JOB_POSTING	Post_id	INT	YES		positive
JOB_POSTING	Post_title	VARCHAR(50)			
JOB_POSTING	Description	VARCHAR(100)			
JOB_POSTING	App_link	VARCHAR(100)			
JOB_POSTING	Location_id	INT		JOB_LOCATION(Location_id)	
JOB_LOCATION	Location_id	INT	YES		
JOB_LOCATION	address	VARCHAR(100)			
APPLIED_JOB	student_id	INT	YES	STUDENT(stuendent_id)	9 digits, positive
APPLIED_JOB	company_id	INT		COMPANY(Company_id)	5 digit, positive
APPLIED_JOB	post_id	INT		JOB_POSTING(post_id)	
APPLIED_LATER	student_id	INT	YES	STUDENT(stuendent_id)	9 digits, positive
APPLIED_LATER	post_id	INT		JOB_POSTING(post_id)	
WORKS_ON	Pno	INT	YES	PROJECT(Pno)	positive
WORK	empl_id	INT		EMPLOYEE(empl_id)	9 digits, positive

Teamwork:

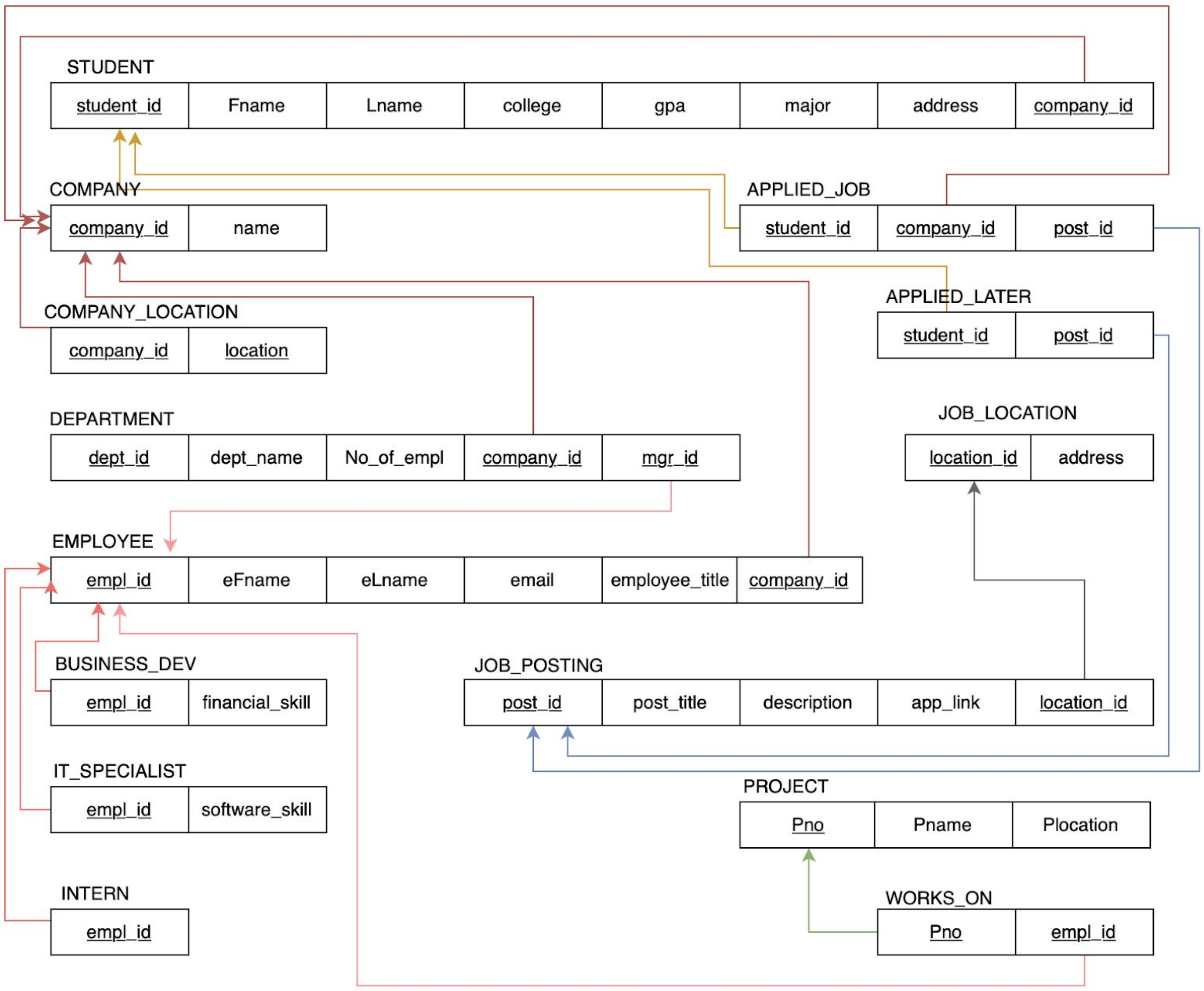
Efaz Khan - Relationships between Entities and Functional Requirements.
Rachid Bodson - Introduction, Data Requirements and Functional Requirements.
Rachel Abraham - Creating Entities, Attributes and the ER diagram.

Appendix:

ER MODEL



RELATIONAL MODEL



STUDENT

student_id	Fname	Lname	college	gpa	major	address	company_id

COMPANY

company_id	name

APPLIED_JOB

student_id	company_id	post_id

APPLIED_LATER

student_id	post_id

COMPANY_LOCATION

company_id	location

DEPARTMENT

dept_id	dept_name	No_of_empl	company_id	mgr_id

JOB_LOCATION

location_id	address

EMPLOYEE

empl_id	eFname	eLname	email	employee_title	company_id

BUSINESS_DEV

empl_id	financial_skill

JOB_POSTING

post_id	post_title	description	app_link	location_id

IT_SPECIALIST

empl_id	software_skill

PROJECT

Pno	Pname	Plocation

INTERN

empl_id

WORKS_ON

Pno	empl_id