

# **Final Report:**

# **InternshipFinder Database**

Group 4

April 20, 2020

Efaz Khan  
Rachid Bodson  
Rachel Abraham

## **Introduction:**

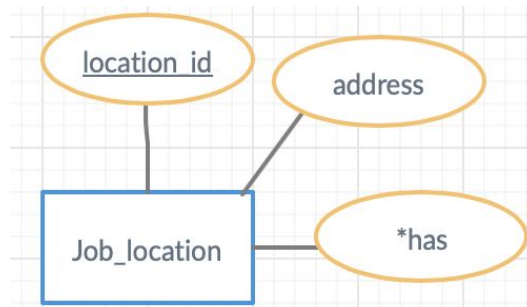
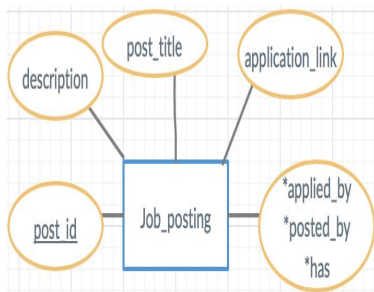
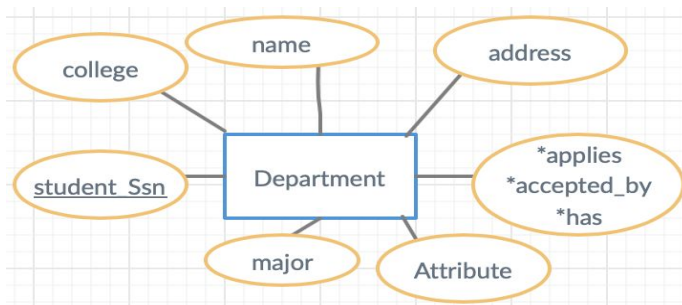
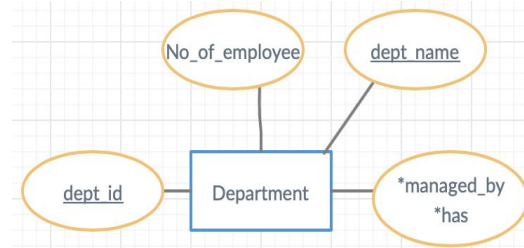
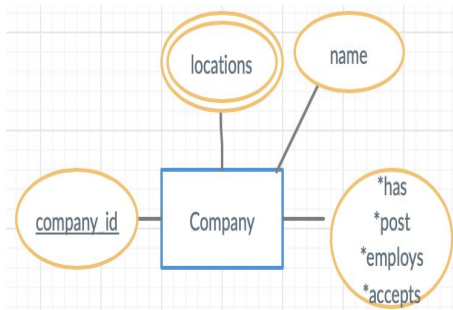
Our designed database, InternshipFinder, is about a unique environment where all kinds of internships are made easy to find without having to browse through several websites. In InternshipFinder, we gather all the best internships from different companies around the country, that are right for all students 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 the 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.

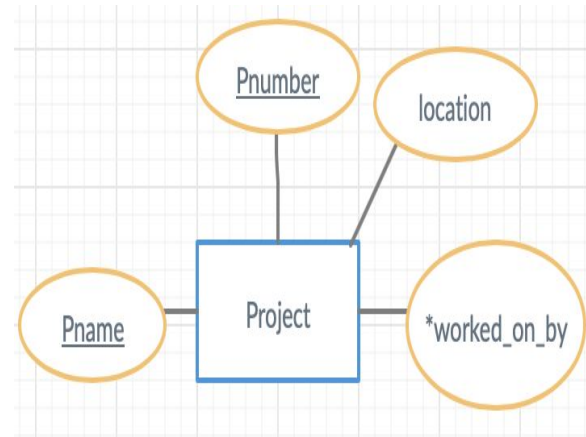
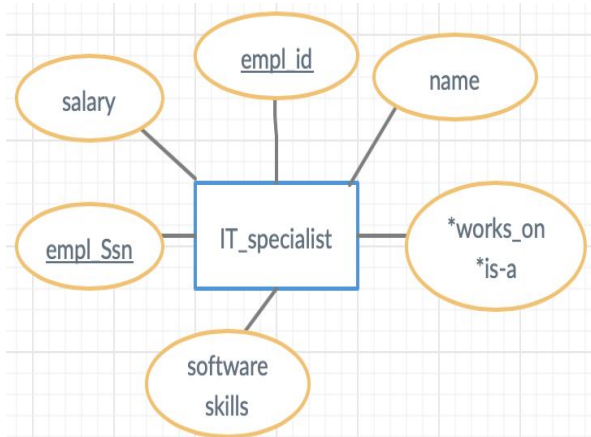
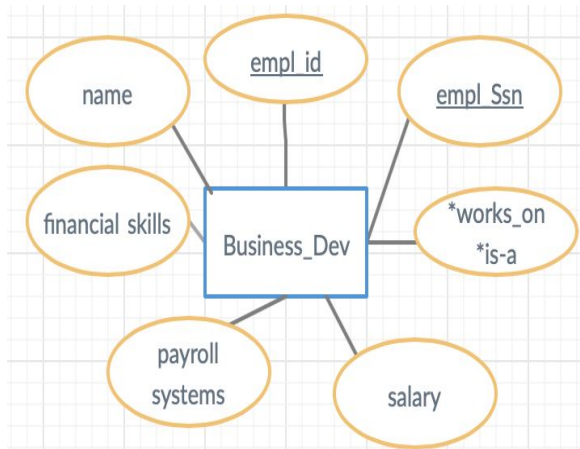
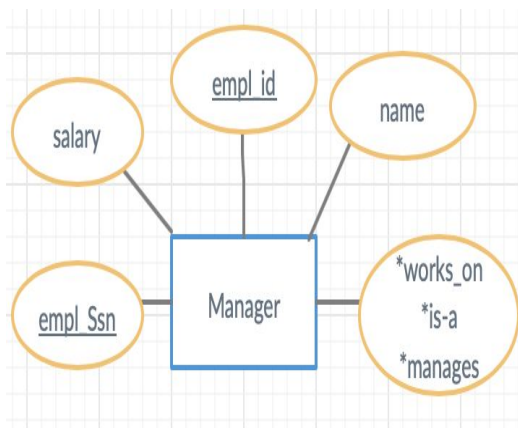
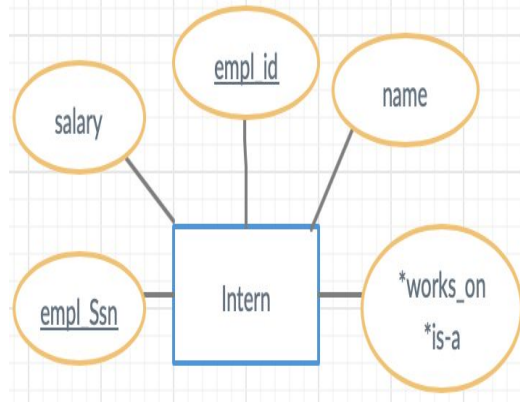
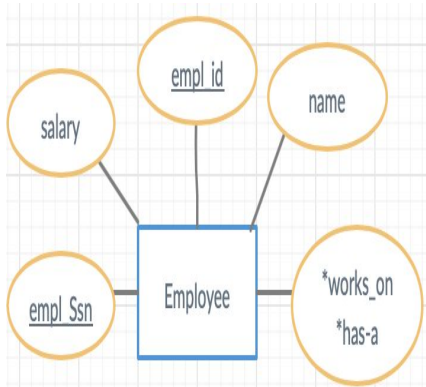
## **Requirements Analysis:**

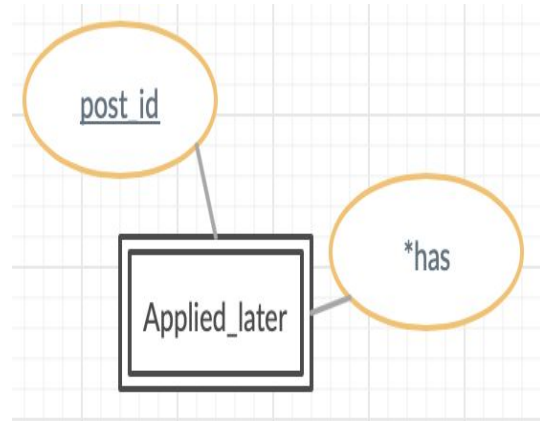
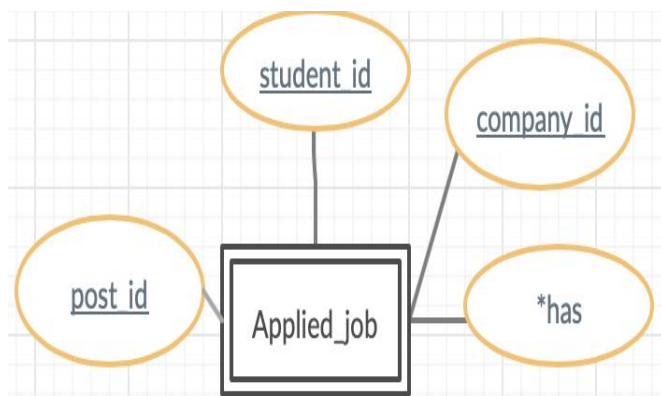
### **Data Requirements:**

- 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 the number of employees.
- The 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 job 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 date. Employees are identified by an employee\_id, name(Fname, Lname), age, and employee\_id.

- Employees are categorized as an intern, manager, IT specialists, business developers.
- Each manager manages only one department
- An Employee works on projects. The project is identified by its unique name, unique number, and location.
- IT specialists have software skills.
- Business Developers have financial skills and payroll systems.







## 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 that a particular company or all companies in the database have received.
- Functional Requirement 5: Students should be able to add/remove an internship to their wishlist, favorite or apply\_later files.
- Functional Requirement 6: Students should be able to access the number of interns that are currently interning for a particular company.
- Functional Requirement 7: Should be able to access the type of projects that interns tend to work on in a particular company.
- Functional Requirement 8: Should return all the internships specific to Computer Science
- Functional Requirement 9: Admin can see the number of applications for each internship in descending order
- Functional Requirement 10: Admin can see the number of students in each major

## 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 the 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 employee\_ssn, 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 employee\_ssn 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:*

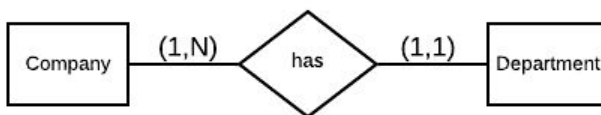
- \*has: Related to STUDENT, it shows the job that will be applied later by a student.

*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*



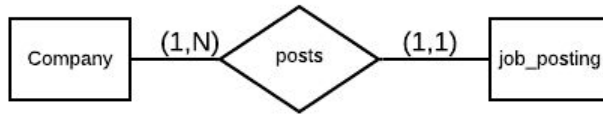
*Relation:* Shows the relation between company and its department

*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*



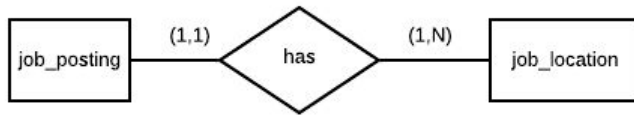
*Relation:* Shows the relation between a company and a student that can accept an offer

*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*



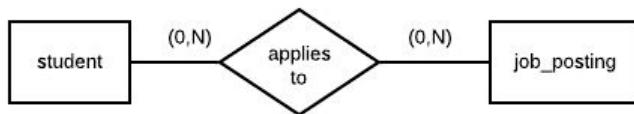
*Relation:* Shows the relation between a job posting and its location

*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*



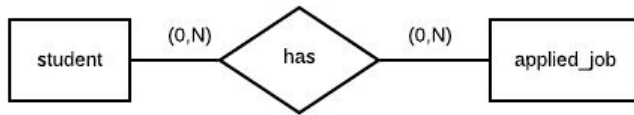
*Relation:* Shows the relation between a student and a job posted by a company

*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*



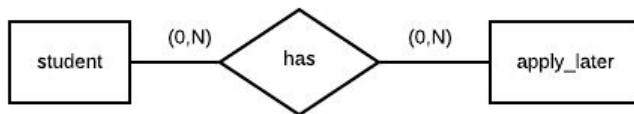
*Relation:* Shows the relation between a student and jobs the student has applied to.

*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*



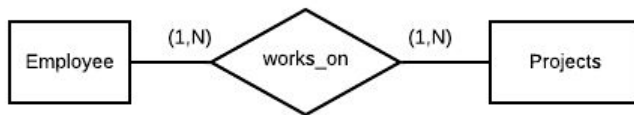
*Relation:* Shows the relation between company and the employees that a company employs.

*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*



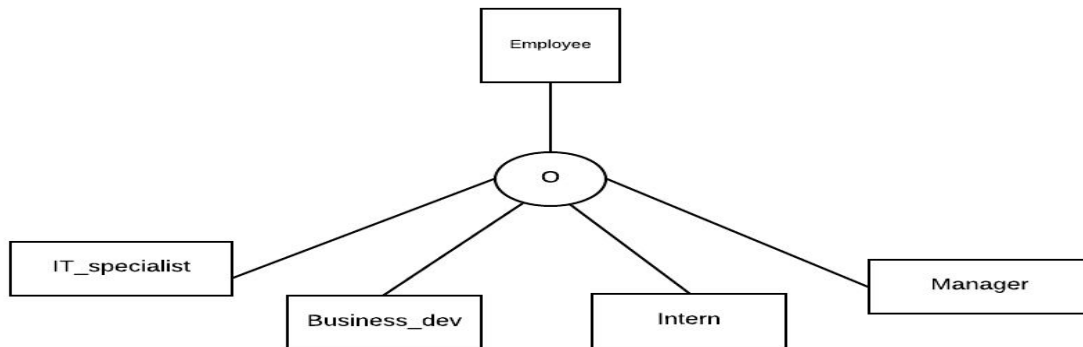
*Relation:* Shows the relation between employees and the projects they work on.

*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)*



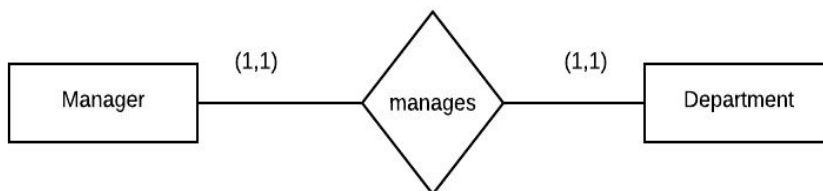
*Relation:* Shows the relation between employee (superclass) and its subclasses (IT\_specialist, Business\_developer, manager, intern)

*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.

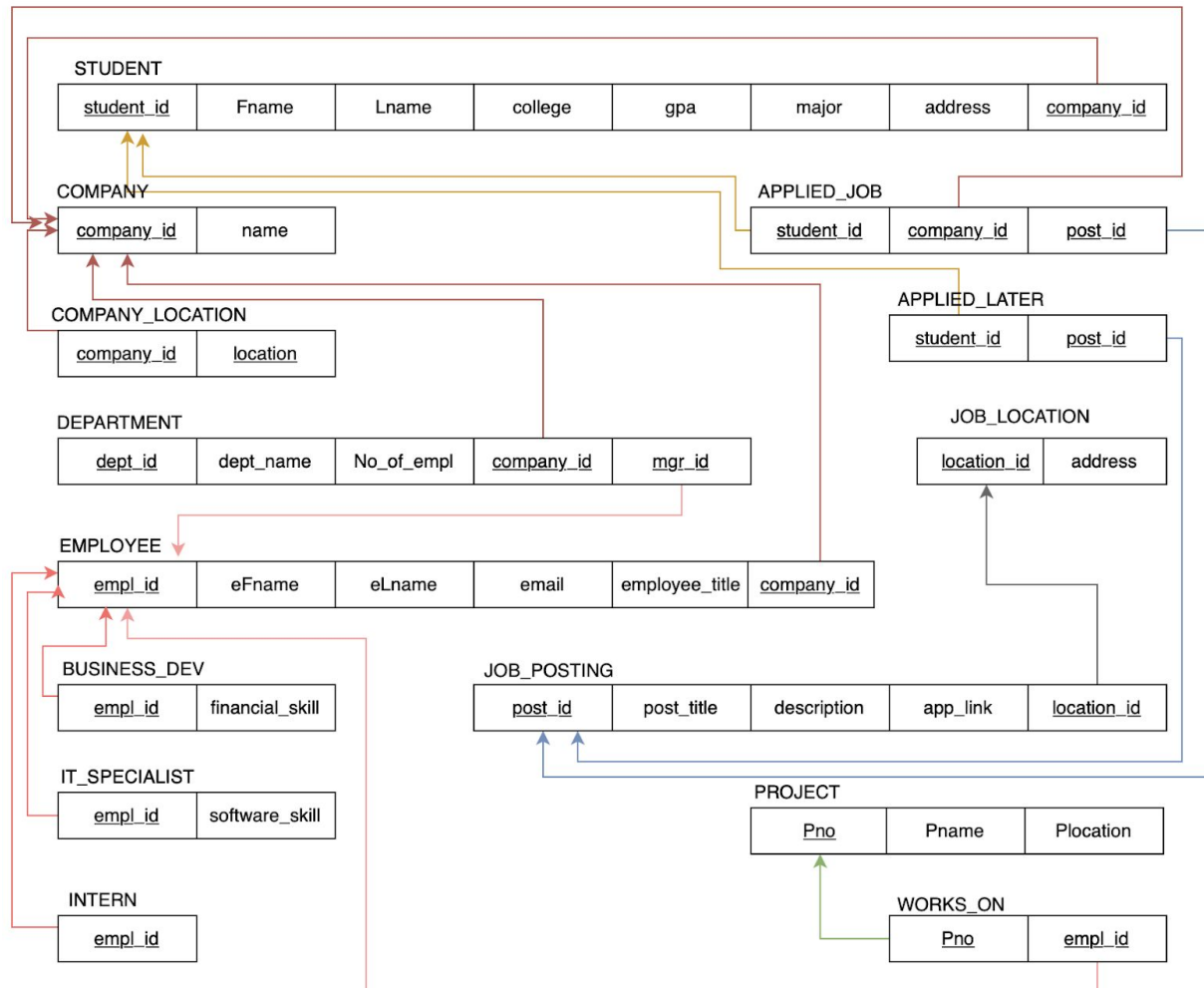
*Attributes:*

- 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:



### *The process of obtaining the 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:

### STUDENT\_PERSONAL\_INFO

<u>student_id</u>	Fname	Lname	college	gpa	major
-------------------	-------	-------	---------	-----	-------

### STUDENT\_LOCATION\_INFO

<u>student_id</u>	location_id
-------------------	-------------

### APPLIED\_JOB

<u>student_id</u>	post_id
-------------------	---------

### APPLY\_LATER

<u>student_id</u>	post_id
-------------------	---------

## STUDENT\_COMPANY

<u>student_id</u>	company_id
-------------------	------------

## COMPANY

<u>company_id</u>	Name
-------------------	------

## COMPANY\_LOCATION

<u>company_id</u>	location
-------------------	----------

## JOB\_LOCATION

<u>location_id</u>	Address
--------------------	---------

## BUSINESS\_DEV

<u>empl_id</u>	financial_skill
----------------	-----------------

## IT\_SPECIALIST

<u>empl_id</u>	software_skill
----------------	----------------

## INTERN

<u>empl_id</u>
----------------

## PROJECT

<u>Pno</u>	Pname	Plocation
------------	-------	-----------

**WORKS\_ON**

<u>Pno</u>	empl_id
------------	---------

**DEPARTMENT\_DETAILS\_INFO**

<u>dept_id</u>	dept_name	No_of_empl
----------------	-----------	------------

**DEPARTMENT\_EMP**

<u>dept_id</u>	mgr_id
----------------	--------

**DEPARTMENT\_COMPANY\_INFO**

<u>dept_id</u>	company_id
----------------	------------

**EMPLOYEE\_INFO**

<u>empl_id</u>	eFname	Lname	email	employee_title
----------------	--------	-------	-------	----------------

**EMPLOYEE\_COMPANY**

<u>empl_id</u>	company_id
----------------	------------

**JOB\_POSTING\_INFO**

<u>post_id</u>	post_title	description	app_link
----------------	------------	-------------	----------

**JOB\_POSTING\_LOCATION**

<u>post_id</u>	location_id
----------------	-------------

## Data Dictionary:

Table	Attribute	Data Type	Primary Key	Foreign Key	Constraints
STUDENT_PERSONAL_INFO	Student_id	INT	YES		7 digits, positive
STUDENT_PERSONAL_INFO	Fname	VARCHAR(20)			
STUDENT_PERSONAL_INFO	Lname	VARCHAR(20)			
STUDENT_PERSONAL_INFO	College	VARCHAR(40)			
STUDENT_PERSONAL_INFO	Gpa	FLOAT			
STUDENT_PERSONAL_INFO	Major	VARCHAR(20)			
STUDENT_LOCATION_INFO	Student_id	INT		STUDENT_PERSONAL_INFO(Student_id)	7 digits, positive
STUDENT_LOCATION_INFO	location_id	VARCHAR(25)			
COMPANY	Company_id	INT	YES		4 digit, positive
COMPANY	Name	VARCHAR(20)			

COMPANY_LOCATION	Company_id	INT		COMPANY(Company_id)	4 digit, positive
COMPANY_LOCATION	Location	VARCHAR(25)			
STUDENT_COMPANY	Student_id	INT		STUDENT_PERSONAL_INFO(Student_id)	7 digits, positive
STUDENT_COMPANY	Company_id	INT		COMPANY(Company_id)	4 digit, positive
DEPARTMENT_DETAIL_INFO	Dept_id	INT	YES		5 digit, positive
DEPARTMENT_DETAIL_INFO	Dept_name	VARCHAR(25)			
DEPARTMENT_DETAIL_INFO	No_of_empl	INT			positive
DEPARTMENT_EMP	Dept_id	INT		DEPARTMENT_DETAIL_INFO(dept_id)	5 digits, positive
DEPARTMENT_EMP	Mgr_id	INT		EMPLOYEE_INFO(empl_id)	7 digits, positive
DEPARTMENT_COMPANY_INFO	Dept_id	INT		DEPARTMENT_DETAIL_INFO(dept_id)	5 digits, positive
DEPARTMENT_COMPANY_INFO	Company_id	INT		COMPANY(Company_id)	4 digit, positive
EMPLOYEE_INFO	Empl_id	INT	YES	EMPLOYEE_INFO(empl_id)	7 digit, positive
EMPLOYEE_INFO	eFname	VARCHAR(20)			
EMPLOYEE_INFO	Lname	VARCHAR(20)			
EMPLOYEE_INFO	Email	VARCHAR(45)			

EMPLOYEE_INFO	Empl_title	VARCHAR(20)			
EMPLOYEE_COMPANY	Empl_id	INT		EMPLOYEE_INFO(empl_id)	7 digit, positive
EMPLOYEE_COMPANY	Company_id	INT		COMPANY(Company_id)	4 digit, positive
INTERN	Empl_id	INT		EMPLOYEE_INFO(empl_id)	7 digit, positive
BUSINESS_DEV	Empl_id	INT		EMPLOYEE_INFO(empl_id)	7 digit, positive
BUSINESS_DEV	Financial_skill	VARCHAR(25)			
IT SPECIALIST	Empl_id	INT		EMPLOYEE_INFO(empl_id)	7 digit, positive
IT SPECIALIST	Software_skill	VARCHAR(25)			
PROJECT	Pno	INT	YES		3 digits, positive
PROJECT	Pname	VARCHAR(20)			
PROJECT	Plocation	VARCHAR(25)			
JOB_POSTING_INFO	Post_id	INT	YES		2 digits, positive
JOB_POSTING_INFO	Post_title	VARCHAR(40)			
JOB_POSTING_INFO	Description	VARCHAR(40)			
JOB_POSTING_INFO	App_link	VARCHAR(40)			
JOB_POSTING_INFO	Location_id	INT		JOB_LOCATION(Location_id)	
JOB_POSTING_LOCATION	Post_id	INT		JOB_POSTING_INFO(post_id)	2 digits, positive
JOB_POSTING_LOCATION	Location_id	INT	YES		4 digits, positive

JOB_LOCATION	Location_id	INT	YES		4 digits, positive
JOB_LOCATION	address	VARCHAR(50)			
APPLIED_JOB	student_id	INT		STUDENT_PERSONAL_INFO(student_id)	7 digits, positive
APPLIED_JOB	post_id	INT		JOB_POSTING_INFO(post_id)	2 digits, positive
APPLIED_LATER	student_id	INT		STUDENT_PERSONAL_INFO(student_id)	7 digits, positive
APPLIED_LATER	post_id	INT		JOB_POSTING_INFO(post_id)	
WORKS_ON	Pno	INT	YES	PROJECT(Pno)	3 digits, positive
WORKS_ON	empl_id	INT		EMPLOYEE_INFO(empl_id)	7 digits, positive

## Implementation:

Functional Requirement 1: Students should be able to create an account with information about themselves. This was implemented by inserting tuples into the student account and then we retrieved a student's info. PS: only accessible to companies (whoever is in control of reviewing application)

Functional Requirement 2: Students should be taken to an external link when a student wants to apply to an internship. We implemented this by retrieving a job's external link.

Functional Requirement 3: Students will be able to mark what internships they have applied for. This was implemented retrieving jobs/internships that a student has applied for.

Functional Requirement 4: InternshipFinder should generate the number of applications that a particular company or all companies in the database have received. We implemented this by retrieving the number of applicants(students) that all the companies have received to determine the most applied company.



Functional Requirement 5: Students should be able to add/remove an internship to their wishlist, favorite or apply\_later files. We implemented this by retrieving the job(s) that a student wants to apply later.

Functional Requirement 6: Students should be able to access the number of interns that are currently interning for a particular company to determine their chances of being called for an interview. We implemented this by retrieving the number of interns that are interning in a particular company.

Functional Requirement 7: Should be able to access the type of projects that interns tend to work on in a particular company to determine the type of projects that they might work on if hired. This was implemented by retrieving projects that interns work on.

Functional Requirement 8: Should return all the internships specific to Computer Science. We implemented this by retrieving all the internships/jobs that are specifically for computer science major

Functional Requirement 9: Admin can see the number of applications for each internship along with the job post title in descending order. We implemented this by retrieving the number of applications in descending order and joining the applied\_job table and the job\_posting table.

Functional Requirement 10: Admin can see the number of students in each major to determine the most dominant major in the database. This was implemented by retrieving the total number of students in each major.

## **Summary:**

All in all, the InternshipFinder is a perfect place for a student to find the best internship. Unlike any other place to find an internship, InternshipFinder gives a student the opportunity to explore a company of their choice as it gives the student access to see the type of workers who work for a particular company which also includes interns, the number of interns, and the project that interns work on. These types of information might help students to well prepare themselves for an internship with the company if hired. It also provides students with the current interns' emails so they could ask for help on how to apply for an internship with a particular company and how to prepare for an interview to better their chances of getting the internship they desire.

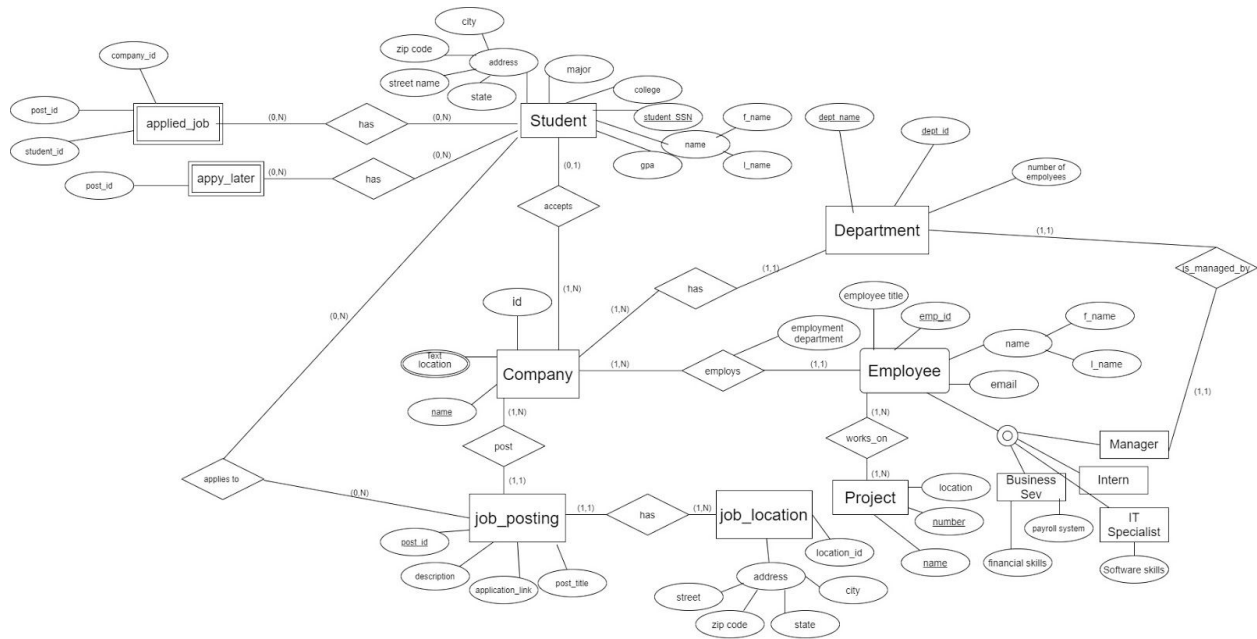
## Teamwork:

Efaz Khan - Relationships between Entities and Functional Requirements,  
Normalization, Implementation, Final Report, Presentation

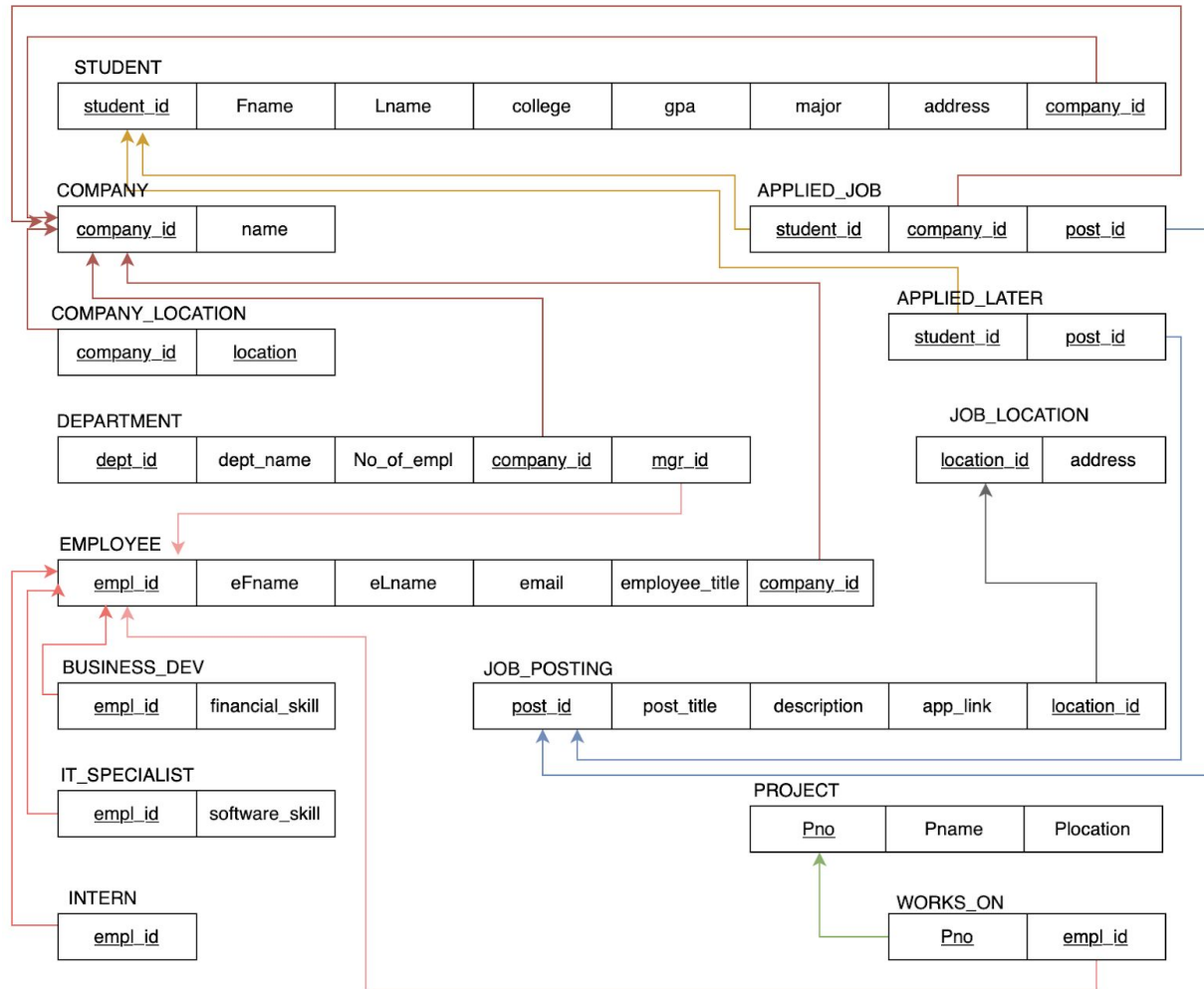
Rachid Bodson - Introduction, Data Requirements, and Functional  
Requirements, Mapping, Data Dictionary, Implementation, Final Report, Presentation

Rachel Abraham - Creating Entities, Attributes, and the ER diagram, Data Dictionary,  
Implementation, Final Report, Presentation

### EER Model



## Relation model



## **Create.sql**

```
use internship_finder;
```

```
create table student_personal_info(  
    student_id int primary key,  
    Fname varchar(20),  
    Lname varchar(20),  
    college varchar(40),  
    gpa float,  
    major varchar(20),  
    CONSTRAINT CHECK (length(student_id) = 7 AND student_id > 1)  
);
```

```
create table student_location_info(  
    student_id int,  
    foreign key(student_id) references student_personal_info(student_id),  
    location varchar(25),  
    CONSTRAINT CHECK (length(student_id) = 7 AND student_id > 1)  
);
```

```
create table company(  
    company_id int primary key,  
    name varchar(20),  
    CONSTRAINT CHECK (length(company_id) = 4 AND company_id > 1)  
);
```

```
create table company_location(  
    company_id int,  
    foreign key(company_id) references company(company_id),  
    location varchar(25)  
);
```

```
create table student_company(  
    student_id int, foreign key(student_id) references student_personal_info(student_id),  
    company_id int,
```

```
foreign key(company_id) references company(company_id),
CONSTRAINT CHECK (length(student_id) = 7 AND student_id > 1 AND
(length(company_id) = 4 AND company_id > 1))
);
```

```
create table job_location(
    location_id int primary key,
    address varchar(45),
    CONSTRAINT CHECK (length(location_id) = 4 AND location_id > 1)
);
```

```
create table job_posting_info(
    post_id int primary key,
    post_title varchar(30),
    description varchar(45),
    app_link varchar(250),
    CONSTRAINT CHECK (length(post_id) = 2 AND post_id > 1 )
);
```

```
create table applied_job(
    student_id int,
    foreign key(student_id) references student_personal_info(student_id),
    post_id int,
    foreign key(post_id) references job_posting_info(post_id),
    CONSTRAINT CHECK (length(student_id) = 7 AND student_id > 1 AND length(post_id) =
2 AND post_id > 1)
);
```

```
create table employee_info(
    empl_id int primary key,
    eFname varchar(20),
    Lname varchar(20),
    email varchar(45),
    employee_title varchar(20),
    CONSTRAINT CHECK (length(empl_id) = 7 AND empl_id > 1 )
);
```

```
create table business_dev(
```

```
    empl_id int,  
    foreign key(empl_id) references employee_info(empl_id),  
    financial_skills varchar(45),  
    CONSTRAINT CHECK (length(empl_id) = 7 AND empl_id > 1 )  
);
```

```
create table it_specialist(  
    empl_id int,  
    foreign key(empl_id) references employee_info(empl_id),  
    software_skill varchar(45),  
    CONSTRAINT CHECK (length(empl_id) = 7 AND empl_id > 1 )  
);
```

```
create table intern(  
    empl_id int, foreign key(empl_id) references employee_info(empl_id),  
    CONSTRAINT CHECK (length(empl_id) = 7 AND empl_id > 1 )  
);
```

```
create table project(  
    Pno int primary key,  
    Pname varchar(30),  
    Plocation varchar(30),  
    CONSTRAINT CHECK (length(Pno) = 3 AND Pno > 1 )  
);
```

```
create table works_on(  
    Pno int, foreign key(Pno) references project(Pno),  
    empl_id int, foreign key(empl_id) references employee_info(empl_id),  
    CONSTRAINT CHECK (length(empl_id) = 7 AND empl_id > 1 AND length(Pno) = 3 AND  
Pno > 1)  
);
```

```
create table department_details_info(  
    dept_id int primary key,  
    dept_name varchar(20),  
    no_of_empl int,  
    CONSTRAINT CHECK (length(dept_id) = 5 AND dept_id > 1 AND no_of_empl>1)  
);
```

```
create table department_emp(  
    dept_id int,  
    foreign key(dept_id) references department_details_info(dept_id),  
    mgr_id int, foreign key(mgr_id) references employee_info(empl_id),  
    CONSTRAINT CHECK (length(dept_id) = 5 AND dept_id > 1 AND length(mgr_id) = 7  
AND mgr_id >1)  
);
```

```
create table department_company_info(  
    dept_id int,  
    foreign key(dept_id) references department_details_info(dept_id),  
    company_id int,  
    foreign key(company_id) references company(company_id),  
    CONSTRAINT CHECK (length(dept_id) = 5 AND dept_id > 1 AND length(company_id) = 4  
AND company_id >1)  
);
```

```
create table employee_company(  
    empl_id int,  
    foreign key(empl_id) references employee_info(empl_id),  
    company_id int, foreign key(company_id) references company(company_id),  
    CONSTRAINT CHECK (length(empl_id) = 7 AND empl_id > 1 AND length(company_id) =  
4 AND company_id > 1)  
);
```

```
create table apply_later(  
    student_id int,  
    foreign key(student_id) references student_personal_info(student_id),  
    post_id int,  
    foreign key(post_id) references job_posting_info(post_id),  
    CONSTRAINT CHECK (length(student_id) = 7 AND student_id > 1 AND length(post_id) =  
2 AND post_id > 1)  
);
```

```
create table job_posting_location(  
    post_id int,  
    foreign key(post_id) references job_posting_info(post_id),
```



```

location_id int,
foreign key(location_id) references job_location(location_id),
CONSTRAINT CHECK (length(post_id) = 2 AND post_id > 1 AND length(location_id) = 4
AND location_id > 1)
);

```

## **Insert.sql**

```

use internship_finder;
INSERT INTO student_personal_info values ('1111222','Rachid','Bodson','Georgia State
University','3.30','Computer Science');
INSERT INTO student_personal_info values ('1111333','Alain','Bodson','Georgia State
University','3.00','Computer Science');
INSERT INTO student_personal_info values ('1111444','Rita','Bodson','Georgia State
University','3.50','Nursing');
INSERT INTO student_personal_info values ('2222222','Rachel','Abraham','Georgia State
University','3.50','Computer Science');
INSERT INTO student_personal_info values ('2222333','Efaz','Khan','Georgia State
University','3.50','Computer Science');
INSERT INTO student_personal_info values ('1212122','Daryl','Johnson','Georgia Southern
University','3.10','Business');
INSERT INTO student_personal_info values ('1313133','Rick','Grims','Georgia Southern
University','3.20','Psychology');
INSERT INTO student_personal_info values ('1414144','Carol','Anderson','Kennesaw State
University','3.40','Nursing');
INSERT INTO student_personal_info values ('1515155','Arezoo','Hosseini','Georgia State
University','3.15','Biology');
INSERT INTO student_personal_info values ('1616166','Issak','Taiwo','Georgia Piedmont
college','3.25','Applied Math');
INSERT INTO student_personal_info values ('2552333','Carl','Grims','Florida State
University','2.50','Geology');
INSERT INTO student_personal_info values ('1414122','Alicia','Zaleya','West Point
University','3.33','Criminal science');
INSERT INTO student_personal_info values ('1515133','Keyanna','Wallace','Georgia Gwinnett
College','3.25','History');
INSERT INTO student_personal_info values ('1616144','Israel','Seka','Kennesaw State
University','3.65','Sociology');

```

INSERT INTO student\_personal\_info values ('1717155','Mehrnush','Hosseini','Georgia State University','3.15','Biology');  
INSERT INTO student\_personal\_info values ('1818166','Marjaneh','Hosseini','Georgia Piedmont college','3.25','Physics');

INSERT INTO student\_location\_info values ('1818166','Atlanta');  
INSERT INTO student\_location\_info values ('1717155','Miami');  
INSERT INTO student\_location\_info values ('1616144','Los Angeles');  
INSERT INTO student\_location\_info values ('1515133','Tampa');  
INSERT INTO student\_location\_info values ('1414122','Atlanta');  
INSERT INTO student\_location\_info values ('2552333','Savanna');  
INSERT INTO student\_location\_info values ('1616166','Orlando');  
INSERT INTO student\_location\_info values ('1515155','Newark');  
INSERT INTO student\_location\_info values ('1414144','New York City');  
INSERT INTO student\_location\_info values ('1111222','Atlanta');  
INSERT INTO student\_location\_info values ('2222222','Atlanta');  
INSERT INTO student\_location\_info values ('2222333','Atlanta');

INSERT INTO company values ('1234','Google');  
INSERT INTO company values ('1122','Facebook');  
INSERT INTO company values ('1244','Home Depot');  
INSERT INTO company values ('1342','T-Mobile');  
INSERT INTO company values ('1111','JP Morgan');  
INSERT INTO company values ('1254','Oscar Health');  
INSERT INTO company values ('1994','Kohls');  
INSERT INTO company values ('1995','CBS');  
INSERT INTO company values ('2000','Apex');  
INSERT INTO company values ('1846','Chanel');  
INSERT INTO company values ('2020','Amazon');

INSERT INTO company\_location values ('1234','Los Angeles');  
INSERT INTO company\_location values ('1122','Miami');  
INSERT INTO company\_location values ('1244','Atlanta');  
INSERT INTO company\_location values ('1342','Tampa');  
INSERT INTO company\_location values ('1111','Orlando');  
INSERT INTO company\_location values ('1254','Atlanta');  
INSERT INTO company\_location values ('1994','Atlanta');  
INSERT INTO company\_location values ('1995','Atlanta');  
INSERT INTO company\_location values ('2000','Birmingham');

```
INSERT INTO company_location values ('1846','Miami');
INSERT INTO company_location values ('2020','Atlanta');
```

```
INSERT INTO student_company values ('1818166','1234');
INSERT INTO student_company values ('1717155','1122');
INSERT INTO student_company values ('1616144','1244');
INSERT INTO student_company values ('1515133','1342');
INSERT INTO student_company values ('1414122','1342');
INSERT INTO student_company values ('2552333','1342');
INSERT INTO student_company values ('1616166','1994');
INSERT INTO student_company values ('1515155','1846');
INSERT INTO student_company values ('1414144','2020');
INSERT INTO student_company values ('1111222','2000');
INSERT INTO student_company values ('2222222','1234');
INSERT INTO student_company values ('2222333','1234');
```

```
INSERT INTO job_location values ('1010','234 Flat Shoal,Atlanta,GA');
INSERT INTO job_location values ('2222','135 Memorial,Atlanta,GA');
INSERT INTO job_location values ('3333','245 Panthersville,Miami,FL');
INSERT INTO job_location values ('4444','555 Lion Mountain,Orlando,FL');
INSERT INTO job_location values ('5555','565 Columbia Drive,Birmingham,AL');
INSERT INTO job_location values ('6666','666 Riverdale,Atlanta,GA');
INSERT INTO job_location values ('7777','142 Columbus,Atlanta,GA');
INSERT INTO job_location values ('8888','777 Rice,Tampa,FL');
INSERT INTO job_location values ('9999','818 Fire Oat,Miami,FL');
INSERT INTO job_location values ('1011','404 River Shoal,Atlanta,GA');
```

```
INSERT INTO job_posting_info values ('12','Computer Science','analyzing big
data','https://www.google.com/data');
INSERT INTO job_posting_info values ('33','Business Admin','handling insurance
finances','https://www.JP Morgan.com/finances');
INSERT INTO job_posting_info values ('45','Biology','controlling
diseases','https://www.CDC.com/diseases');
INSERT INTO job_posting_info values ('60','Journalism','reporting on
coronavirus','https://www.CBS.com/covid-19');
INSERT INTO job_posting_info values ('20','Computer Science','testing
designs','https://www.facebook.com/testing');
```

```
INSERT INTO applied_job values ('1818166','45');
INSERT INTO applied_job values ('1717155','20');
INSERT INTO applied_job values ('1616144','45');
INSERT INTO applied_job values ('1515133','33');
INSERT INTO applied_job values ('1414122','60');
INSERT INTO applied_job values ('2552333','60');
INSERT INTO applied_job values ('1616166','45');
INSERT INTO applied_job values ('1515155','33');
INSERT INTO applied_job values ('1414144','33');
INSERT INTO applied_job values ('1111222','20');
INSERT INTO applied_job values ('2222222','12');
INSERT INTO applied_job values ('2222333','12');
```

```
INSERT INTO employee_info values
('3333222','Rasheed','Hamed','rasheedhamed@gmail.com','web designer');
INSERT INTO employee_info values ('4444333','Atayi','Carlors','atayicarlors@yahoo.com','web
programmer');
INSERT INTO employee_info values ('5555444','Rita','Juan','rita12@gmail.com','Nurse');
INSERT INTO employee_info values
('8485838','Rochella','Koutop','koutop1234@icloud.com','Database analyzer');
INSERT INTO employee_info values ('2222444','Eyaz','Davids','eyazdav@gmail.com','App
designer');
INSERT INTO employee_info values ('1313122','Rick','Johnson','rickjl@gmail.com','Business
Advisor');
INSERT INTO employee_info values
('1414133','Mehdi','Azar','mehdiaz@gmail.com','Psychologist');
INSERT INTO employee_info values
('1515144','Carol','Peterson','carolpeterson@icloud.com','Nurse');
INSERT INTO employee_info values
('1616155','Merlon','Wayan','merlon344@yahoo.com','Biologist');
INSERT INTO employee_info values
('1717166','Mussa','Abdul','mussaabd14@gmail.com','Finance analyzer');
INSERT INTO employee_info values
('2818333','Carl','Carlors','carls5764@gmail.com','Geologist');
INSERT INTO employee_info values
('1919222','Tigist','Zaleya','tg473@icloud.com','Receptionist');
INSERT INTO employee_info values
('1404133','Ruth','Wallace','ruthwallace@yahoo.com','Coder');
```

INSERT INTO employee\_info values ('1505144','Judith','Seka','seka8489@icloud.com','Coding tester');

INSERT INTO employee\_info values

('1606155','Azaar','Hosseini','azaarhossei15@gmail.com','Game designer');

INSERT INTO employee\_info values

('1707166','Afia','Makou','afiamakou@icloud.com','Physicist');

INSERT INTO business\_dev values ('1717166','strategistical and communication skill');

INSERT INTO business\_dev values ('1313122','advisory and communication skill');

INSERT INTO it\_specialist values ('1505144','technical coding and typing skill');

INSERT INTO it\_specialist values ('1606155','video gaming skill');

INSERT INTO intern values ('3333222');

INSERT INTO intern values ('5555444');

INSERT INTO project values ('123','ProjectX','Los Angeles');

INSERT INTO project values ('112','ProjectY','Miami');

INSERT INTO project values ('124','Blue','Atlanta');

INSERT INTO project values ('134','Lion','Tampa');

INSERT INTO project values ('111','T-virus','Orlando');

INSERT INTO project values ('125','Falcon','Atlanta');

INSERT INTO project values ('199','Diamond','Atlanta');

INSERT INTO project values ('191','LionHeart','Atlanta');

INSERT INTO project values ('200','ProjectZ','Bermingham');

INSERT INTO project values ('184','Green','Miami');

INSERT INTO project values ('202','InternshipFinder','Atlanta');

INSERT INTO works\_on values ('123','3333222');

INSERT INTO works\_on values ('112','5555444');

INSERT INTO works\_on values ('124','1313122');

INSERT INTO works\_on values ('134','1707166');

INSERT INTO works\_on values ('111','1404133');

INSERT INTO works\_on values ('125','2818333');

INSERT INTO works\_on values ('199','3333222');

INSERT INTO works\_on values ('191','1707166');

INSERT INTO works\_on values ('200','1404133');

INSERT INTO works\_on values ('184','5555444');

INSERT INTO works\_on values ('202','1717166');

```
INSERT INTO department_details_info values ('11111','Marketing','32');
INSERT INTO department_details_info values ('11224','HR','25');
INSERT INTO department_details_info values ('23111','IT services','30');
INSERT INTO department_details_info values ('67211','Research development','15');
INSERT INTO department_details_info values ('77541','Sales','50');
INSERT INTO department_details_info values ('17381','Customer support','15');
INSERT INTO department_details_info values ('36477','Production Inventory','10');
```

```
INSERT INTO department_emp values ('11111','1616155');
INSERT INTO department_emp values ('11224','1606155');
INSERT INTO department_emp values ('23111','1505144');
INSERT INTO department_emp values ('67211','1707166');
```

```
INSERT INTO department_company_info values ('11111','2020');
INSERT INTO department_company_info values ('11224','1846');
INSERT INTO department_company_info values ('23111','1234');
INSERT INTO department_company_info values ('67211','1122');
INSERT INTO department_company_info values ('77541','1244');
INSERT INTO department_company_info values ('17381','1342');
INSERT INTO department_company_info values ('36477','1994');
```

```
INSERT INTO employee_company values ('3333222','1234');
INSERT INTO employee_company values ('4444333','1122');
INSERT INTO employee_company values ('5555444','1254');
INSERT INTO employee_company values ('8485838','2020');
INSERT INTO employee_company values ('2222444','1234');
INSERT INTO employee_company values ('1313122','2020');
INSERT INTO employee_company values ('1414133','1254');
INSERT INTO employee_company values ('1515144','1254');
INSERT INTO employee_company values ('1616155','1254');
INSERT INTO employee_company values ('1717166','1244');
INSERT INTO employee_company values ('1919222','1995');
INSERT INTO employee_company values ('1404133','1122');
INSERT INTO employee_company values ('1505144','1122');
INSERT INTO employee_company values ('1606155','1234');
INSERT INTO employee_company values ('1707166','1254');
```

```
INSERT INTO apply_later values ('1212122','12');
```

```
INSERT INTO apply_later values ('1313133','33');
INSERT INTO apply_later values ('1414144','45');
INSERT INTO apply_later values ('1515155','60');
INSERT INTO apply_later values ('1616166','20');
INSERT INTO apply_later values ('2552333','20');
INSERT INTO apply_later values ('1414122','60');
INSERT INTO apply_later values ('1515133','12');
INSERT INTO apply_later values ('1616144','33');
INSERT INTO apply_later values ('1717155','45');
INSERT INTO apply_later values ('1818166','12');
```

```
INSERT INTO job_posting_location values ('12','1010');
INSERT INTO job_posting_location values ('33','2222');
INSERT INTO job_posting_location values ('45','3333');
INSERT INTO job_posting_location values ('60','4444');
INSERT INTO job_posting_location values ('20','5555');
INSERT INTO job_posting_location values ('12','6666');
INSERT INTO job_posting_location values ('45','7777');
INSERT INTO job_posting_location values ('60','8888');
INSERT INTO job_posting_location values ('33','1011');
```

### Queries.sql with output:

```
use internship_finder;
```

```
select *
```

```
from student_personal_info;
```

	student_id	Fname	Lname	college	gpa	major
►	1111222	Rachid	Bodson	Georgia State University	3.3	Computer Science
	1111333	Alain	Bodson	Georgia State University	3	Computer Science
	1111444	Rita	Bodson	Georgia State University	3.5	Nursing
	1212122	Daryl	Johnson	Georgia Southern University	3.1	Business
	1313133	Rick	Grims	Georgia Southern University	3.2	Psychology
	1414122	Alicia	Zaleya	West Point University	3.33	Criminal science
	1414144	Carol	Anderson	Kennesaw State University	3.4	Nursing
	1515133	Keyanna	Wallace	Georgia Gwinnett College	3.25	History
	1515155	Arezoo	Hosseini	Georgia State University	3.15	Biology
	1616144	Israel	Seka	Kennesaw State University	3.65	Sociology
	1616166	Issak	Taiwo	Georgia Piedmont college	3.25	Applied Math
	1717155	Mehrnush	Hosseini	Georgia State University	3.15	Biology
	1818166	Marjaneh	Hosseini	Georgia Piedmont college	3.25	Physics
	2222222	Rachel	Abraham	Georgia State University	3.5	Computer Science
	2222333	Efaz	Khan	Georgia State University	3.5	Computer Science
	2552333	Carl	Grims	Florida State University	2.5	Geology



```
select app_link  
from job_posting_info;
```

	app_link
▶	<a href="https://www.google.com/data">https://www.google.com/data</a>
	<a href="https://www.facebook.com/testing">https://www.facebook.com/testing</a>
	<a href="https://www.JP Morgan.com/finances">https://www.JP Morgan.com/finances</a>
	<a href="https://www.CDC.com/diseases">https://www.CDC.com/diseases</a>
	<a href="https://www.CBS.com/covid-19">https://www.CBS.com/covid-19</a>

```
select post_id  
from applied_job;
```

	post_id
▶	12
	12
	20
	20
	33
	33
	33
	45
	45
	45
	60
	60

```
select COUNT(student_id)
from applied_job;
```

	COUNT(student_i...
▶	12

```
select post_id
from apply_later;
```

	post_id
▶	12
	12
	12
	20
	20
	33
	33
	45
	45
	60
	60

```
select COUNT(empl_id)
from intern;
```

	COUNT(empl_id)
▶	2

```
select Pname  
from project;
```

	Pname
▶	T-virus
	ProjectY
	ProjectX
	Blue
	Falcon
	Lion
	Green
	LionHeart
	Diamond
	ProjectZ
	InternshipFinder

```
select post_id  
from job_posting_info  
where post_title = 'Computer Science';
```

	post_id
▶	12
	20

```

select job_posting_info.post_title, COUNT(applied_job.post_id)
from applied_job
JOIN job_posting_info ON job_posting_info.post_id = applied_job.post_id
group by applied_job.post_id
order by COUNT(applied_job.post_id) DESC;

```

	post_title	COUNT(applied_job.post_...
▶	Business Admin	3
	Biology	3
	Computer Science	2
	Computer Science	2
	Journalism	2

```

select COUNT(student_id)
from student_personal_info
group by major;

```

	COUNT(student_i...
▶	4
	2
	1
	1
	1
	1
	2
	1
	1
	1
	1