**ADDIS ABABA UNIVERSITY**

**Addis Ababa Institute of Technology**

**School of electrical and computer**

**Engineering**

**Database System Project**

            Project title: - **Job Portal System (Hire and find a job)**

**phase-II**

**Group 4**

**Group Members Name**

Henok Wondimu   ATR/6564/10

Abeniezer Kifle   ATR/2834/10

Samiya Hamid    ATR/5017/10

Yordanos Teku    ATR/5802/10

Girmachew Azanaw   ATR/7667/10

Submitted to Mr. Tesfamichael

Submission data May 3, 2021

**Contents**

1. ER to Relational Mapping----------------------------------------------------------------3

        1.1.  Mapping of regular Entity type---------------------------------------------4

                   1.2.  Mapping of Binary 1:1 relations---------- ----------------------------------5

       1.3.  Mapping of Binary 1:N relations---------------------------------------------6

* 1. Mapping of Binary M:N relations--------------------------------------------7

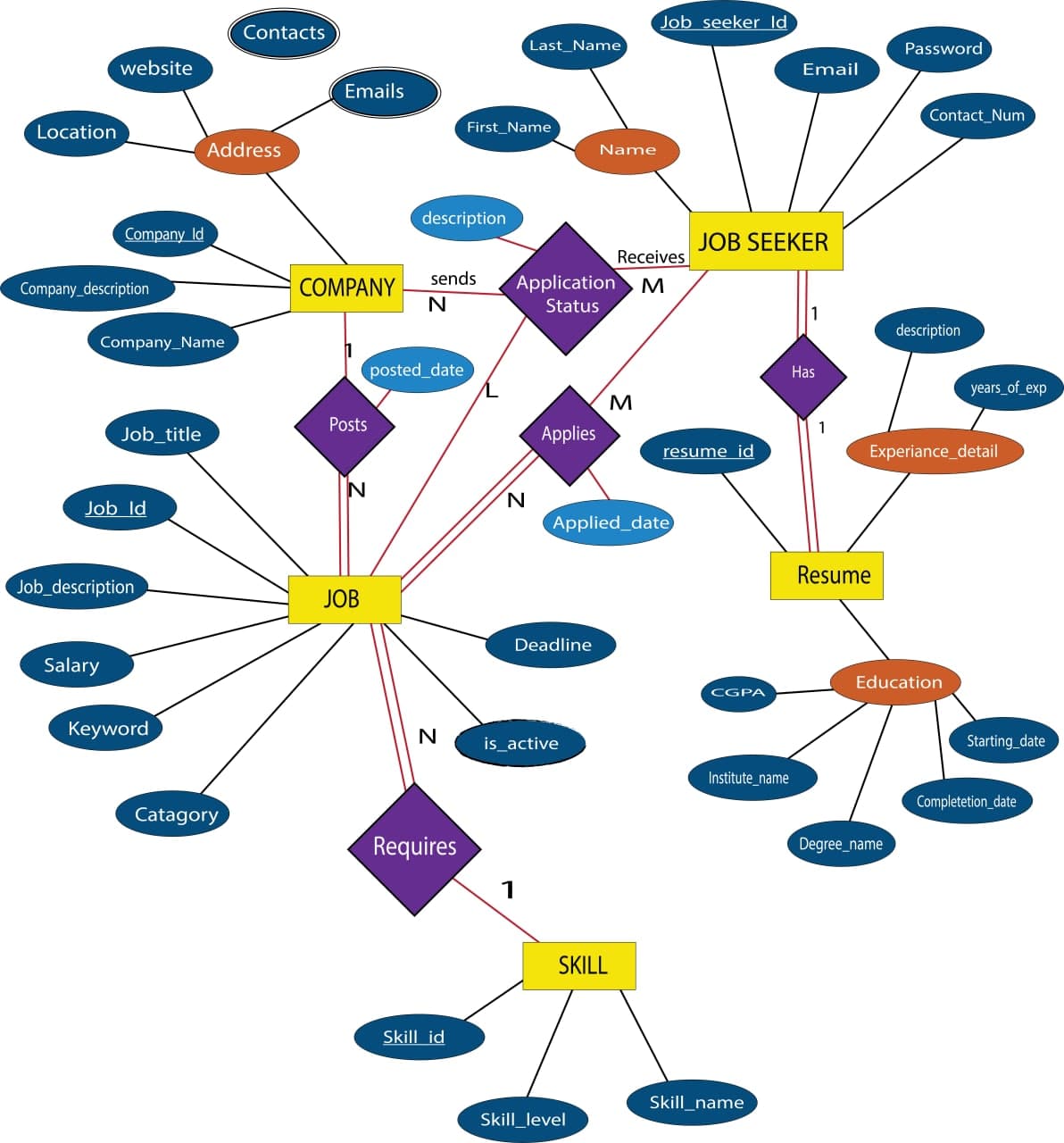
    1.5.  Mapping of Multivalued attributes------------------------------------------9

       1.6.  Overall mapped relational diagram----------------------------------------10

1. Functional Dependency -----------------------------------------------------------------11
2. Normalization -----------------------------------------------------------------------------14

1. ER to Relational Mapping

ER diagram from our previous phase:



**1.1 Mapping of Regular Entity types**

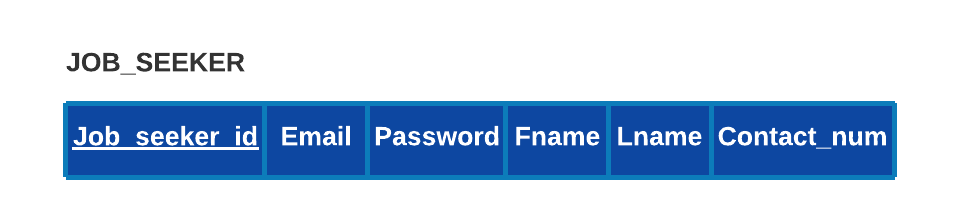
From the above ER diagram we can clearly see that there are five regular type entities. COMPANY, JOB\_SEEKER, JOB, RESUME, and SKILL.

1. **COMPANY**

 A COMPANY entity has three simple attributes and one composite attribute. So, we create a relation that includes all the simple attributes of COMPANY including only the simple component attributes of a composite attribute. Finally, the new relation become:

1. **JOB\_SEEKER**

JOB\_SEEKER entity has four simple attributes and one composite attributes. So, We create a relation that includes all the simple attributes of JOB\_ SEEKER including only the simple component attributes of a composite attribute. Finally, the new relation become:



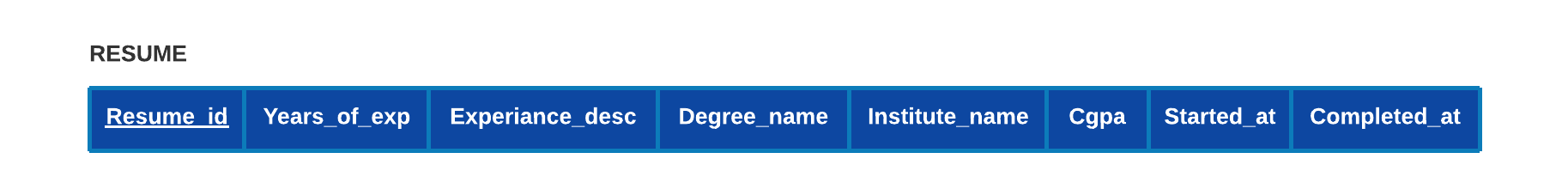
1. **JOB**

From the ER diagram, JOB entity has eight simple attributes, So, We create a relation that includes all the simple attributes of JOB. Finally, the new relation become:



1. **RESUME**

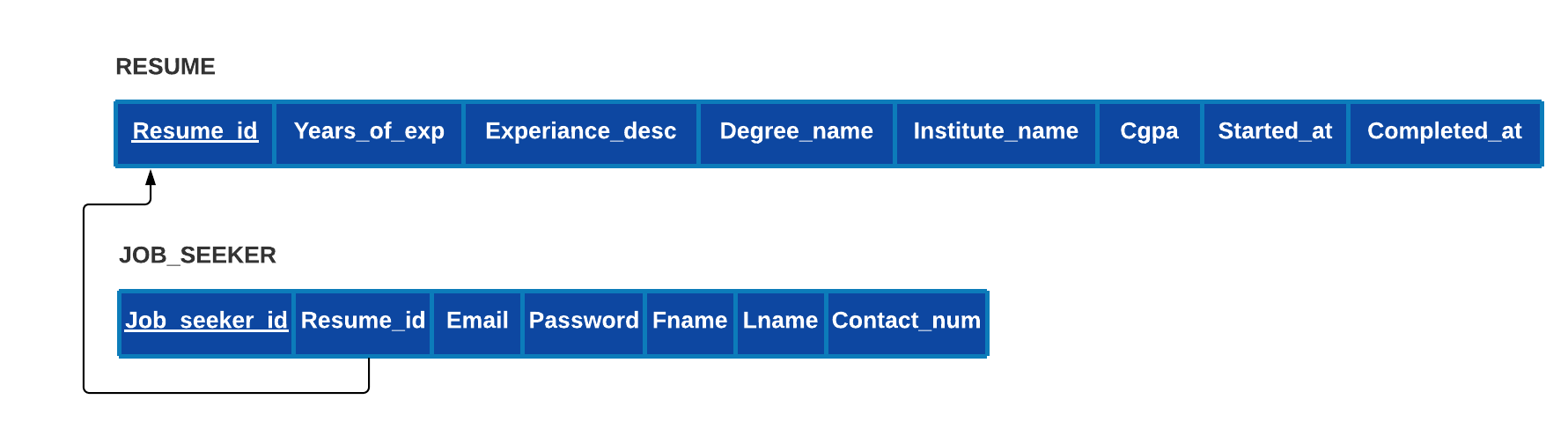
From the ER diagram, RESUME entity has one simple attributes, and two compoiste attributes. So, We create a relation that includes all the simple attributes of JOB including only the simple component attributes of a composite attributes. Finally, the new relation become:



1. **SKILL**

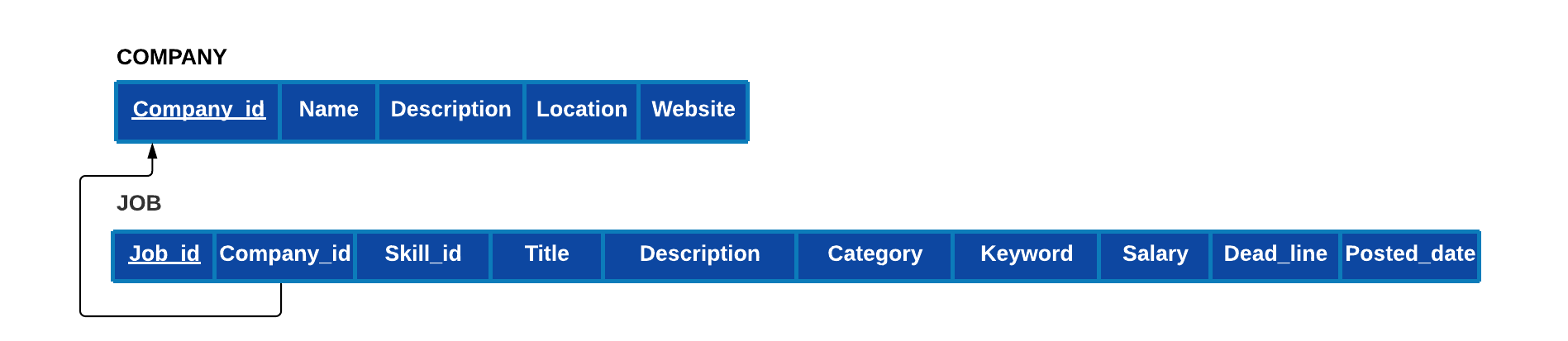
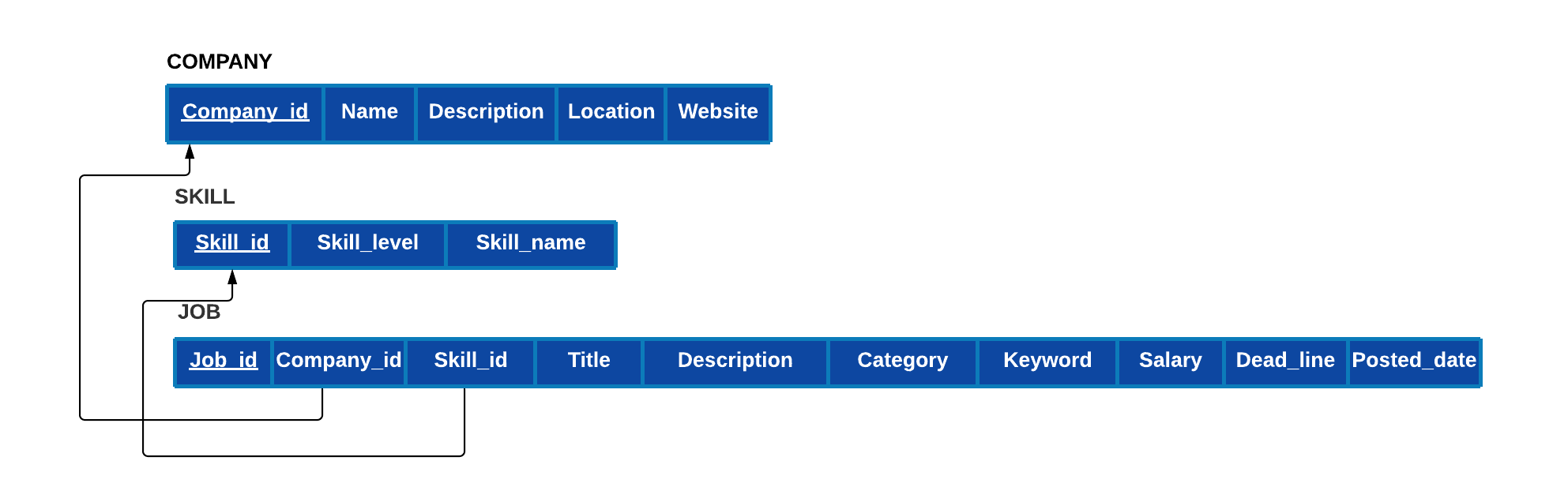
From the ER diagram, SKILL entity has three simple attributes, So, We create a relation that includes all the simple attributes of SKILL. Finally, the new relation become 

**1.2 Mapping of Binary 1:1 relationships**

 From our ER diagram we can clearly see that, there is only a single binary 1:1 relationship, which is a relation between JOB\_SEEKER and RESUME. For mapping we use the most power full approch, which is Forign key approch. Since both enitites participate totally we can choose one of the relations, in this case JOB\_SEEKER and include as a foreign key the primary key of the other entity, which is RESUME. After this step our updated JOB\_SEEKER relation is:

**1.3 Mapping of Binary 1:N relationships**

From our ER diagram we can clearly see that, there are two binary 1:N relation ship, which are a relation between COMPANY and JOB and between JOB and SKILL. For each regular binary 1:N relationship type R, we identify the relation S that represents the participating entity type at the N-side of the relationship type. And include as foreign key in S the primary key of the relation T that represents the other entity type participating in R.

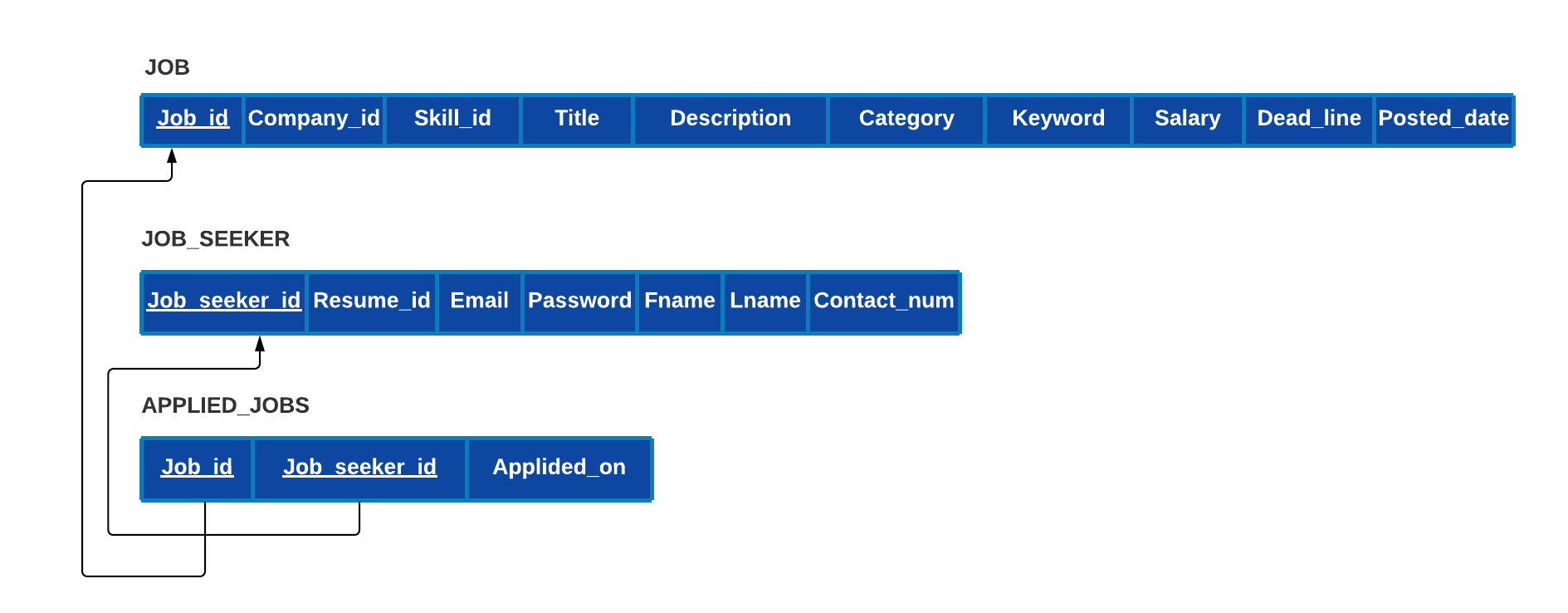
* Between COMPANY AND JOB: In this relationship JOB Is at the N-side, so we include COMPANY’s primary key as a forign key in JOB. After this step, the JOB relation well be:
* Between JOB AND SKILL: In this relationship JOB is at the N-side, so we include SKILL’s primary key as a forign key in JOB. After this step, the JOB relation well be:

**1.4 Mapping of Binary M:N relationships**

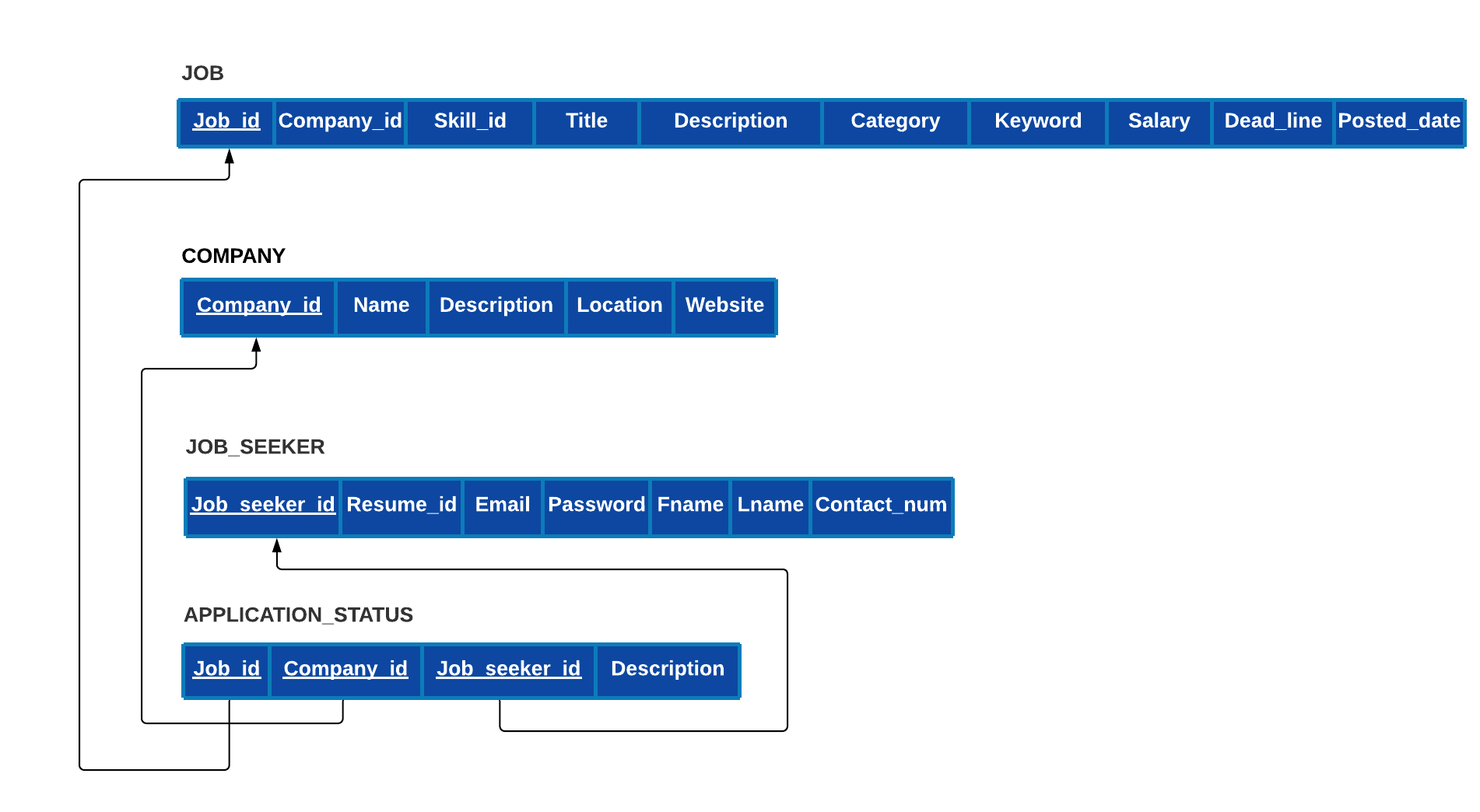
From our ER diagram we can clearly see that, there are two binary M:N relation ship, which are a relation between JOB and JOB\_SEEKER and relation among COMPANY, JOB\_SEEKER and JOB. For each binary M:N relationship type R, create a new relation S to represent R. Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; their combination will form the primary key of S. Also include any simple attributes of the M:N relationship type (or simple components of composite attributes) as attributes of S.

* Between JOB\_SEEKER and JOB (APPLIES relationship)

For this relationship we create a new realtion called APPLIED\_JOBS and include as a foreign key the primary keys of JOB AND JOB\_SEEKER including relationship’s simple attribute. After this step, the JOB relation well be:

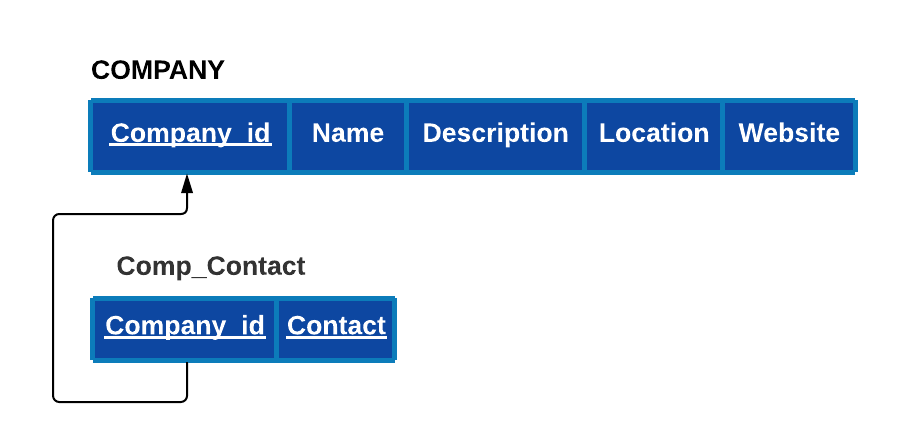


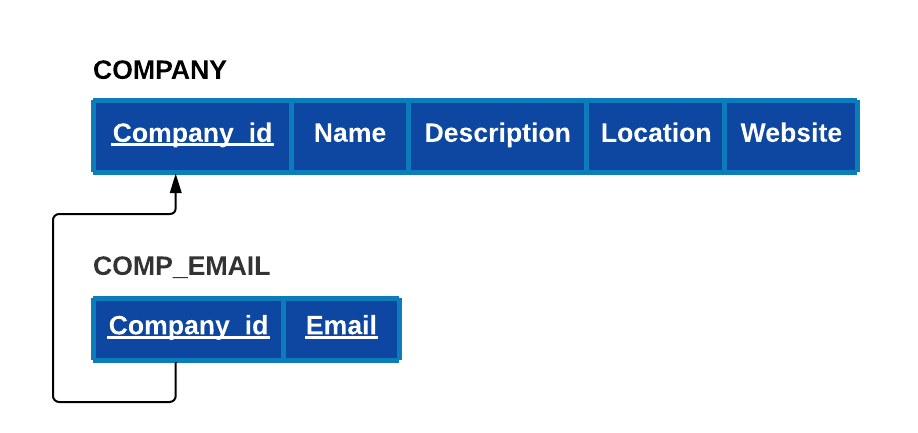
* Among COMPANY, JOB\_SEEKER and JOB

(APPLICATION\_STATUS): For this relationship we create a new realtion called APPLICATION\_STATUS and include as a foreign key the primary keys of COMPANY, JOB\_SEEKER AND JOB including relationship’s simple attribute. After this step, the JOB relation well be:

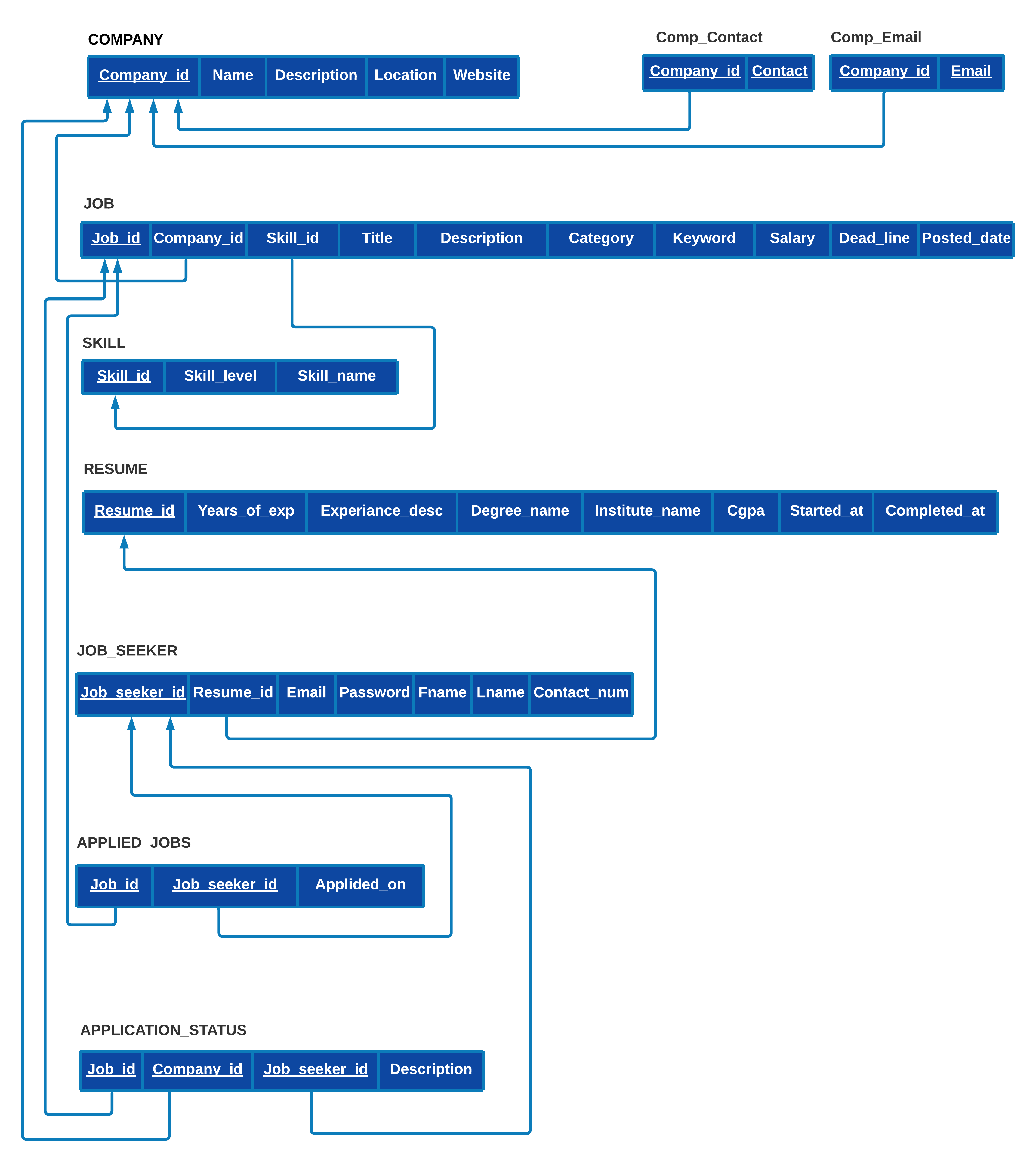
**1.5 Mapping of Multivalued attributes**

From our ER diagram we can clearly see that, there are two mulitvalued attributes, which are Contacts and Emails. For each multivalued attribute A, we create a new relation R. This relation R will include an attribute corresponding to A, plus the primary key attribute K—as a foreign key in R—of the relation that represents the entity type or relationship type that has A as a multivalued attribute. The primary key of R is the combination of A and K.

* Contacts : For this multivalued attribute we created a new relation called COMP\_CONTACT including primary key of the COMPANY as its forign key and corresponding attribute for Contact. Finaly, newly created relation will be :
* Emails : For this multivalued attribute we created a new relation

called COMP\_EMAIL including primary key of the COMPANY as its forign key and corresponding attribute for email. Finaly, newly created relation will be :

* 1. Overall mapped relational diagram will be :



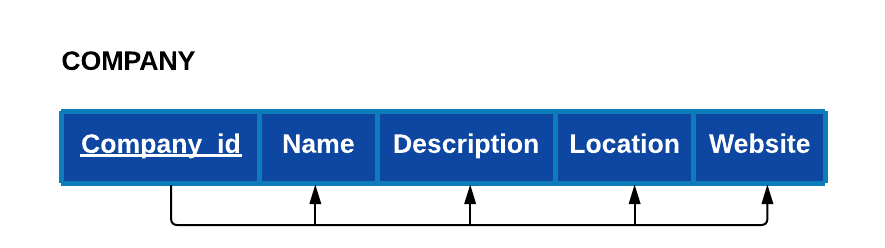
**2. Functional Dependencies**

**COMPANY**

In the Company Relation the company Id is the primary key that determines every other attribute; so, we have only one functional dependency.

  R={Company\_Id, Name, Description, Location, Website}

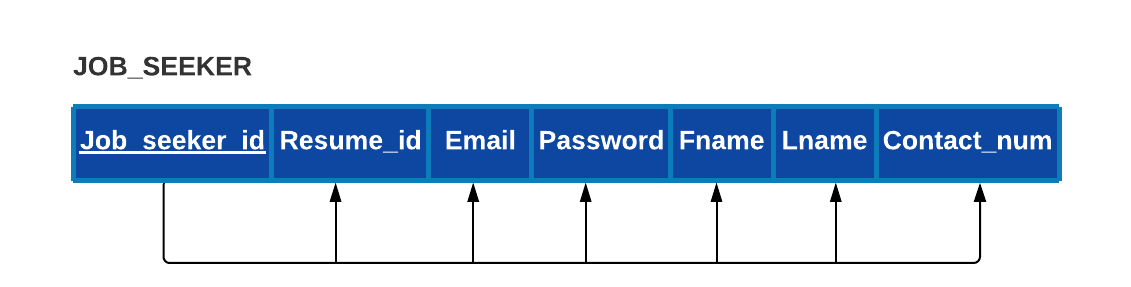
  F={Company\_id =>{ Name, Description, Location, Website}}



**Job Seeker:**

In the Job\_seeker relation the Job\_seeker\_id is the primary key and it determines all other attributes; so, we have only one functional dependency.

R={Job\_seeker\_id, Resume\_id, Email, Password, Fname, Name, Contact\_num}

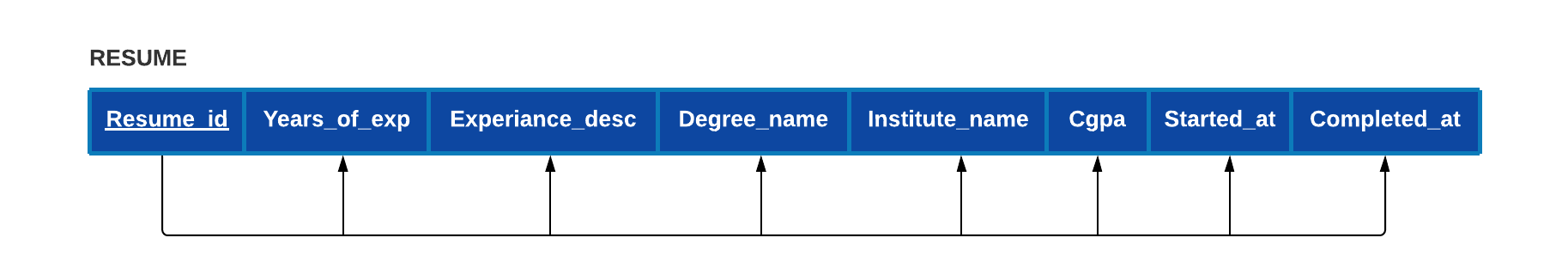
F={ Job\_seeker\_id=> {Resume\_id, Email, Password, Fname, Name , Contact\_num}}

**Resume:**

In Resume Relation the Resume\_id is the primary key and it determines all other attributes and so we have the following functional dependency.

R={Resume\_id, year\_of\_exp, Experiance\_desc, Degree\_name, Institute\_name, cgpa, started\_at, completed\_date}

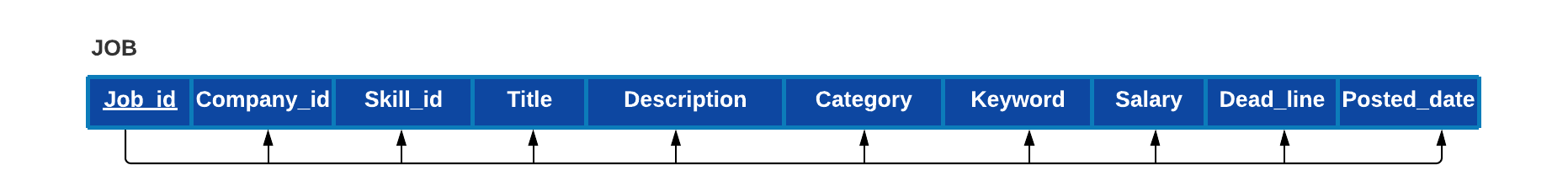
F={ Resume\_id => { year\_of\_exp, Experiance\_desc, Degree\_name, institute\_name , cgpa, started\_at, completed\_at}}



**Job:**

Like the others in this relation Job\_id can determine other attributes and this follows the following functional dependency.

R={Job\_id, Company\_id, Skill\_id, Title, Description, Category, Keyword, Salary, deadline, Posted\_date}

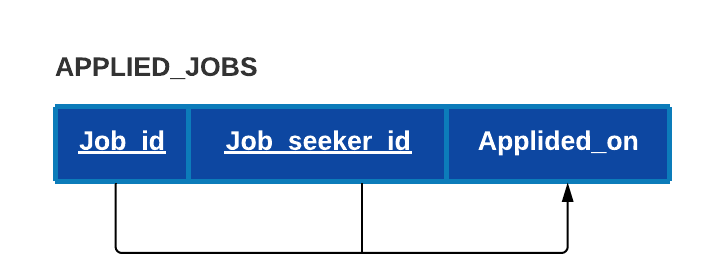
F={Job\_id=>{Company\_id, skill\_id, Title, Description, Category, keyword, salary, deadline, posted\_date}}

**Applied\_Jobs:**

In the Applied\_Jobs relation we have a Primary key which is composed of Job\_id and Job\_seeker\_id. The primary key can determine the other attribute as the following functional dependency shows below:

R={Job\_id, Job\_seeker\_id, Applied\_on}

F={Job\_id, Job\_seeker\_id=>{Applied\_on}}

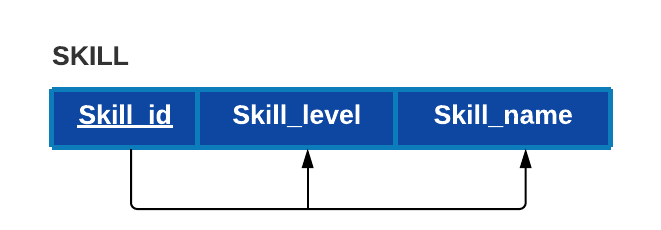


**Skill:**

In Skill Relation skill\_id is the primary key and it can determine the other two attributes. So we have the following functional dependency.

R={Skill\_id, Skill\_level, Kill\_name}

F={skill\_Id=> {skill\_level, skill\_name}}

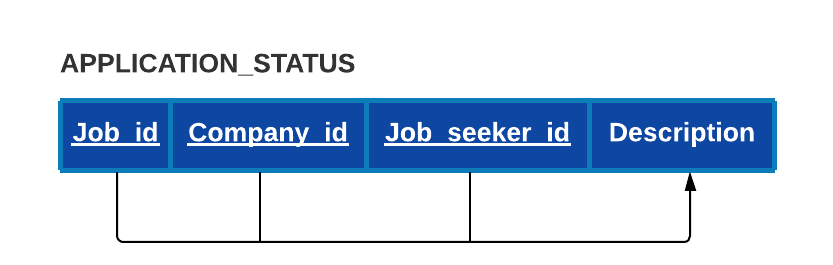


**APPLICATION\_STATUS:**

In Application Status relation we have a Primary key which is composed of three attributes and they all determine the other non prime attribute on the relation. so, we have the following functional dependency.

R={Job\_id, Company\_id, Job\_seeker\_id, description}

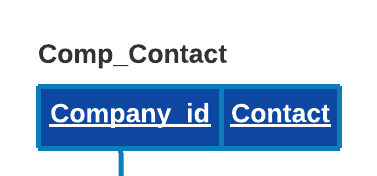
F={ Job\_id, Company\_id, Job\_seeker\_id=> {description}}



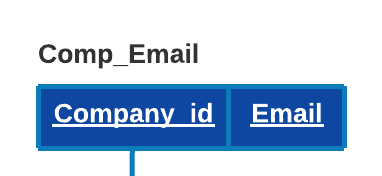
**3. Normalizations**



COMPANY relation is already normalized to 3NF form since there is only one functional dependency in the relation.



COMP\_CONTACT relation is already normalized to 3NF form because the realtion doesn’t have any functional dependency.



COMP\_EMAIL relation is already normalized to 3NF form.



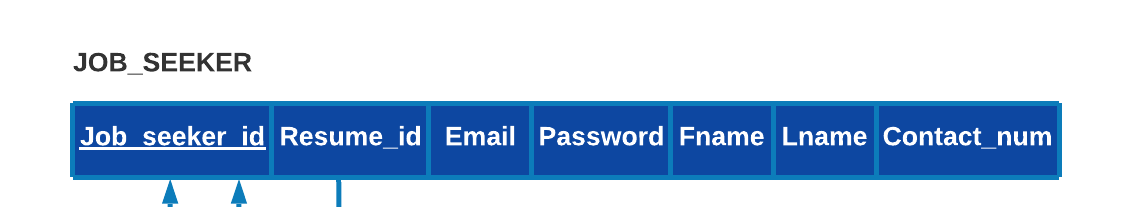
JOB relation is already normalized to 3NF form because there is only one functional dependency in the relation.



SKILL relation is already normalized to 3NF form since there is only one functional dependency in the relation.



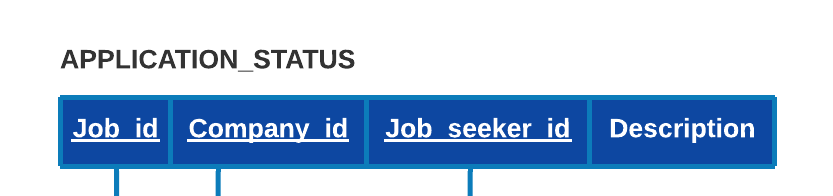
RESUME relation is already normalized to 3NF form.



JOB\_SEEKER relation is already normalized to 3NF form.



APPLIED\_JOBS relation is already normalized to 3NF form.



APPLICATIN\_STATUS relation is already normalized to 3NF form.