Conrad Lewin
CS 340
8/6/2016

## Project Summary

My wife recently secured a software engineering job in Raleigh, North Carolina, so we will be moving there at the end of this term. This project aims to provide a tool with which we can use to survey this new environment. The application supplies its user with the ability to track employers, jobs and their required skill sets, both leisure and work related activities and various local venues. Aside from tracking these opportunities, the user is also able to assign a rating to each data element with the intention of attributing a numeric, in the case of venues, activities, jobs and employers, or lexical, insofar as skills are concerned, enjoyment/interest value rating to each element. At the moment, seeing as I will be unemployed upon our arrival in Raleigh, only jobs can be searched for and updated; however, all entities can be inserted into the database and subsequently viewed in tabular format.

The hope is that this tool will provide an easy way for my wife and I to survey the general opportunities Raleigh and the surrounding cities have to offer its residents. The organized collection of this data should provide a quick and easy way to log these opportunities, allowing us to view simple reports that will aid in any decisions we make regarding the various jobs and activities we are bound to discover.

## Database Description

<u>Main Entities</u>

- Employer
  - primary key constraint on id column
  - the id, name and rating columns cannot be null
- Job
  - primary key constraint on id column
  - unique key constraint on title and description columns
  - foreign key constraint on employerID column that references the id column in the employer table (set to on delete cascade)
  - only the pay column is allowed a null value
- Skill
  - primary key constraint on id column
  - unique key constraint on name and description columns
  - no column is allowed a null value
- Activity
  - primary key constraint on id column
  - unique key constraint on description column
  - no column is allowed a null value
- Venue
  - primary key constraint on id column
  - the id, name, description and rating columns cannot be null

<u>Relational Entities</u>

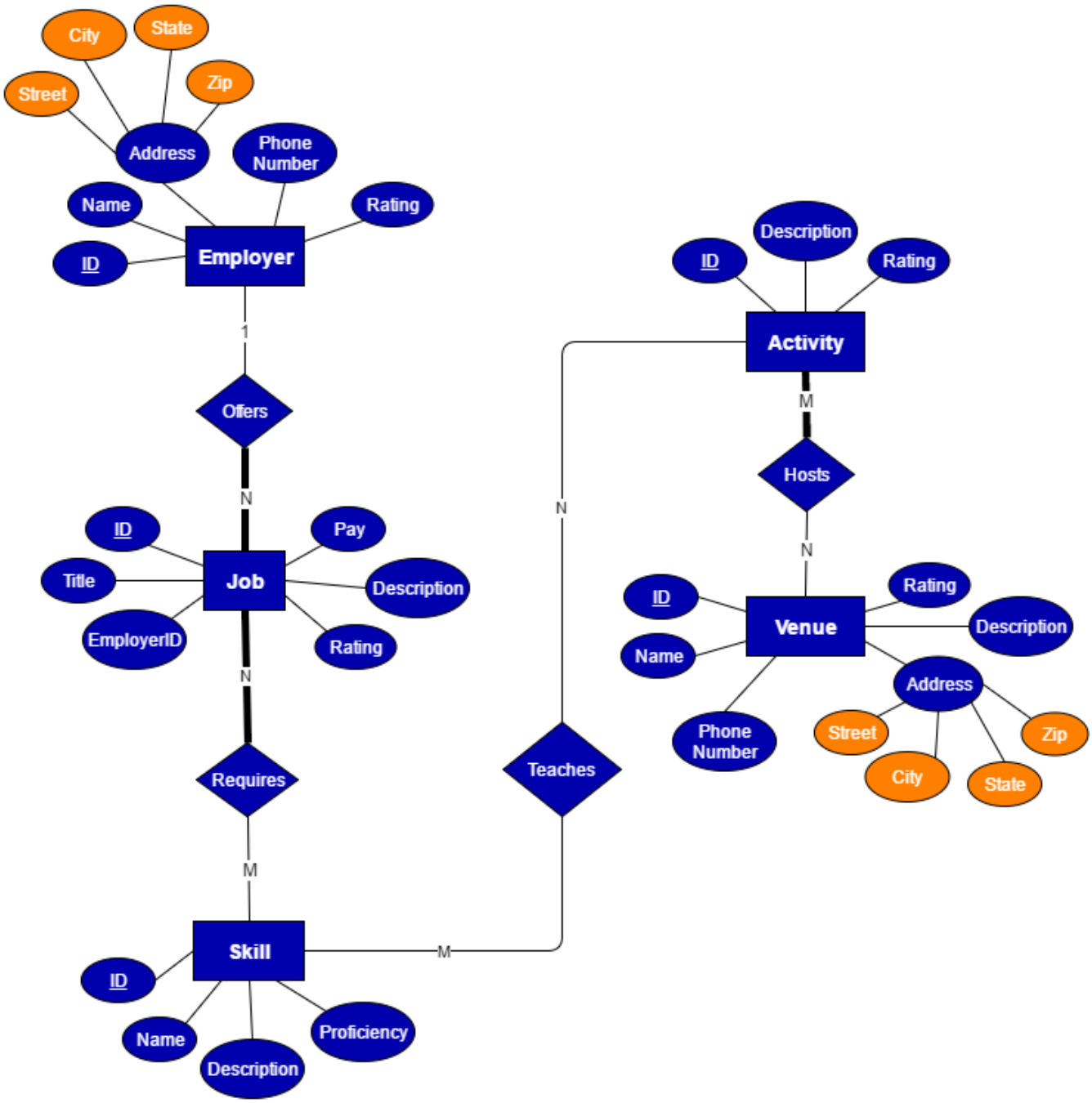
- Job_Skill
  - primary key constraint on jobID and skillID columns

    - foreign key constraint on jobID column that references the id column in the job table (set to on delete cascade)
    - foreign key constraint on skillID column that references the id column in the skill table
    - no column is allowed a null value
- Skill_Activity
    - primary key constraint on skillID and activityID columns
    - foreign key constraint on skillID column that references the id column in the skill table
    - foreign key constraint on activityID column that references the id column in the activity table (set to on delete cascade)
    - no column is allowed a null value
- Activity_Venue
    - primary key constraint on activityID and venueID columns
    - foreign key constraint on activityD column that references the id column in the activity table (set to on delete cascade)
    - foreign key constraint on venueID column that references the id column in the venue table (set to on delete cascade)
    - no column is allowed a null value

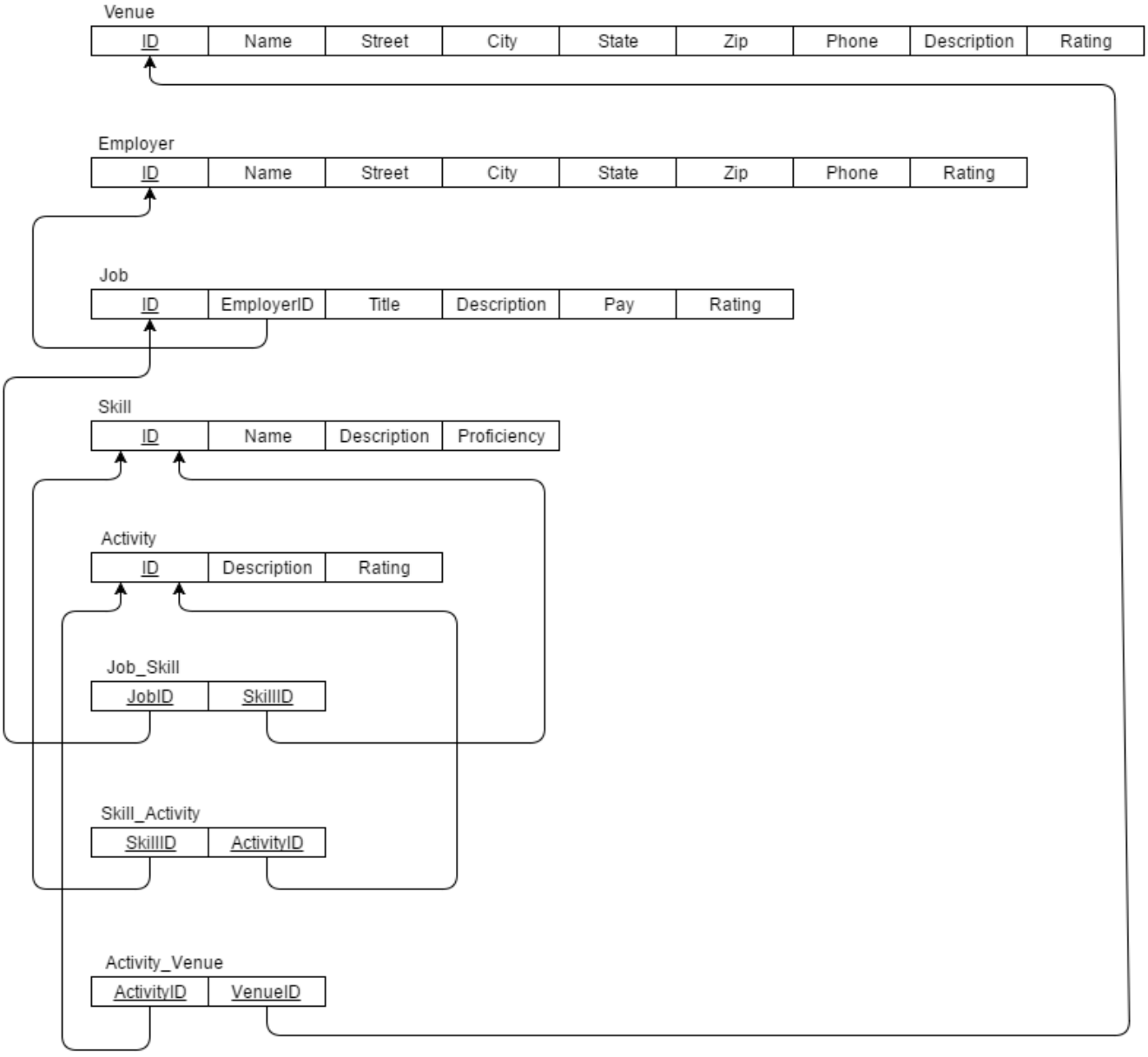
<u>Relationships</u>

- One-to-Many
    - At most 1 employer can offer many different jobs.
    - Each job must be offered by at most 1 employer.
- Many-to-Many
    - Each job must require at least one skill.
    - Each skill may be associated with 0 or more jobs.
    - Each skill may be associated with 0 or more activities.
    - Each activity may teach 0 or more skills.
    - Each activity must be hosted by at least 1 venue.
    - Each venue can host 0 or more activities.

# Entity Relationship Diagram



## Schema


Venue
ID
Name
Street
City
State
Zip
Phone
Description
Rating
Employer
ID
Name
Street
City
State
Zip
Phone
Rating
Job
ID
EmployerID
Title
Description
Pay
Rating
Skill
ID
Name
Description
Proficiency
Activity
ID
Description
Rating
Job_Skill
JobID
SkillID
Skill_Activity
SkillID
ActivityID
Activity_Venue
ActivityID
VenueID

## Project Queries

Table Creation Queries

```
create table if not exists employer
(id int primary key auto_increment,
name varchar(255) not null,
street varchar(255),
city varchar(255),
state varchar(2) default "NC",
zip varchar(10),
phone varchar(10),
rating enum('1','2','3','4','5','6','7','8','9','10') not null)
engine = innodb;


create table if not exists job
(id int primary key auto_increment,
employerID int not null,
title varchar(255) not null,
description varchar(255) not null,
pay decimal(10,2) unsigned,
rating enum('1','2','3','4','5','6','7','8','9','10') not null,
unique(title,description),
foreign key (employerID) references employer(id)
on delete cascade)
engine = innodb;


create table if not exists skill
(id int primary key auto_increment,
name varchar(255) not null,
description varchar(255) not null,
proficiency enum('very poor','poor','acceptable','good','very good','excellent') not null,
unique(name,description))
engine = innodb;


create table if not exists activity
(id int primary key auto_increment,
description varchar(255) unique not null,
rating enum('1','2','3','4','5','6','7','8','9','10') not null)
engine = innodb;


create table if not exists venue
(id int primary key auto_increment,
name varchar(255) not null,
street varchar(255),
```

```
city varchar(255),
state varchar(2) default "NC",
zip varchar(10),
phone varchar(10),
description varchar(255) not null,
rating enum('1','2','3','4','5','6','7','8','9','10') not null)
engine = innodb;


create table if not exists job_skill
(jobID int not null,
skillID int not null,
primary key (jobID,skillID),
foreign key (jobID) references job(id)
on delete cascade,
foreign key (skillID) references skill(id))
engine = innodb;


create table if not exists skill_activity
(skillID int not null,
activityID int not null,
primary key (skillID,activityID),
foreign key (skillID) references skill(id),
foreign key (activityID) references activity(id)
on delete cascade)
engine = innodb;


create table if not exists activity_venue
(activityID int not null,
venueID int not null,
primary key (activityID,venueID),
foreign key (activityID) references activity(id)
on delete cascade,
foreign key (venueID) references venue(id)
on delete cascade)
engine = innodb;
```

General Use Queries

```
insert into employer (name,street,city,zip,phone,rating) values ([empName],[empStreet],
[empCity],[empZip],[empPhone],[empRating]);

insert into skill (name,description,proficiency) values ([skillName],[skillDesc],[skillProf]);

insert into job (employerID,title,description,pay,rating) values ([employerID],[jobTitle],
[jobDesc],[jobPay],[jobRating]);
```

```
insert into job_skill (jobID,skillID) values ([jobID],[skillID]);

insert into venue (name,street,city,zip,phone,description,rating) values ([venueName],
[venueStreet],[venueCity],[venueZip],[venuePhone],[venueDesc],[venueRating]);

insert into activity (description,rating) values ([activityDesc],[activityRating]);

insert into skill_activity (skillID,activityID) values ([skillID],[activityID]);

insert into activity_venue (activityID,venueID) values ([‘activityID],[venueID]);

update job set
employerID = [employerID],
title = [jobTitle],
description = [jobDesc],
pay = [jobPay],
rating = [jobRating]
where id = [jobID]’

delete from job
where id = [jobID];

select e.name,j.title,j.description,j.pay,j.rating
from employer e
inner join job j
on e.id=j.employerID
where {j.employerID = [employerID] | and j.title like % [jobTitle] | and j.pay
[relationalOperator1] [jobPay] | and j.rating [relationalOperator2] [jobRating]};

select name,street,city,state,zip,phone,rating from employer;

select description,rating from activity;

select a.description, v.name
from activity a
inner join activity_venue av
on a.id=av.activityID
inner join venue v
on av.venueID=v.id;

select e.name,j.title,j.description,j.pay,j.rating
from employer e
inner join job j
on e.id=j.employerID;
```

```
select e.name,j.title,s.name
from job j
inner join job_skill js
on j.id=js.jobID
inner join skill s
on js.skillID=s.id
inner join employer e
on j.employerID=e.id;

select name, description, proficiency from skill;

select s.name, a.description, v.name
from skill s
inner join skill_activity sa
on s.id=sa.skillID
inner join activity a
on sa.activityID=a.id
inner join activity_venue av
on av.activityID=a.id
inner join venue v
on av.venueID=v.id;

select name,street,city,state,zip,phone,description,rating from venue;

select id, name from [table];

select id,name from employer;

select j.id, j.employerID, e.name, j.title, j.description, j.pay, j.rating
from job j
inner join employer e
on j.employerID=e.id;
```