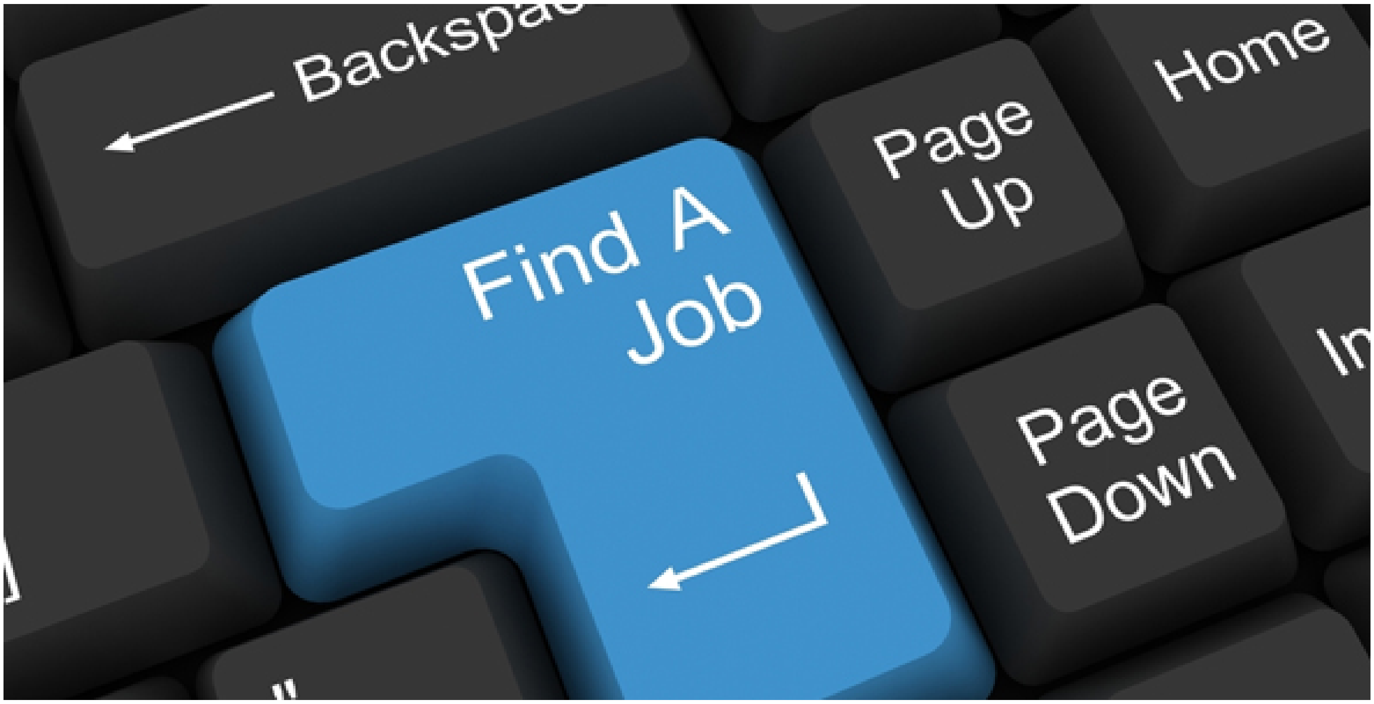


FINAL PROJECT
JOB APPLICATION DATABASE

Live database:

http://web.engr.oregonstate.edu/~balaba/project/main.php

Authors:
Brett Irvin
Aleksandr Balab

INTRODUCTION TO DATABASES (CS 340)
Oregon State University
Fall 2015

I. Outline

For our 340 project, we chose to construct a database of jobs that we have applied to. As computer science students, we will likely apply to several jobs before being hired. This database will help us keep a close eye on the job applications and their status. It will serve as a fast reference if we need to react on a professional email or a phone call. Additionally it should keep everything organized and will help us out with follow up calls.

Since we are new to the industry, we will probably apply to a broader variety of jobs than more experienced and specialized applicants would. The types of jobs we will likely encounter can be sorted by many different criteria, making them complex enough to justify using a database.

This database will consist of the most essential information about the position that one is applying to e.g. company, job(position), source, job requirements, application status and most importantly contact information.

II. Database Outline in Words

The Job Application database consists of five entities. They are designed as follows:

1. **Job**. Job is written as “job” when accessing the table. This is a job that the company(potential employer) is offering to the employment market.This is the information that we are usually presented by the source(e.g. jobs.com). Job table has the following attributes:
    a. **j_id** : The job identification key/ primary key. This is an auto incrementing integer that represents title, address,employment option (full-time,part-time, internship), job requirements, status of job application and potential salary.
    b. **title**: The official job title as presented by the source of application. This is varchar(255) and may be entered as any string of user choice. The title is an essential part of the job description and therefore cannot be null.
    c. **emp_option**: The employment option. This is varchar(255) and may be entered as any string of choice. Most likely it will be entered as values (full-time, part-time, internship). This cannot be null.
    d. **salary**: Potential salary of the position. This is an integer and can be null.
    e. **experience**: Preferred experience. This is varchar(255) and may be entered as any string of choice(e.g. 6 months, 1 year, 2 years). This can be null.
    f. **app_id**: The status of job application. This is an integer that is a foreign key reference to application entity(“application” table). The app_id cannot be null.
    g. **company_id**: Company id. This is an integer that is a foreign key reference to company entity(“company” table). The company_id cannot be null.

2. **Application.** Written as “application” when accessing the table. This entity contains the current status of the application, method of submission(electronically via internet or via physical resume), the date of the submission and notes.
    a. **a_id**: Application status id is an identification key/primary key. This is an auto incrementing integer. It cannot be null.
    b. **method**: Method of submission. This is varchar(255) and may be entered as any string of choice(most likely as values “internet” or “physical”). This can be null.
    c. **app_st**: The status of the application. This is varchar(255) and can be entered as any string of choice. It cannot be null.
    d. **sub_date**: Date of submission. This is varchar(255) and may be entered as any string of choice. Since the date of submission is always available for inquiry, this cannot be null.
    e. **notes**: Some additional information about the application. This is varchar(255) and can be entered as any string of choice. It cannot be null.

3. **Company**. Written as “company” when accessing the table. This entity contains the essential information about the company that posted a job.
    a. **co_id**. Company identification number also is a company primary key. This is auto incrementing integer. It cannot be null.
    b. **name**: The name of the company. This is varchar (255) and may be entered as any string of choice. This cannot be null since this is essential information.
    c. **industry**: The industry that this company belongs to (e.g. tech, financial services, non-profit, government etc.). This is varchar (255) and may be entered as any string of choice. This can be null.
    d. **department**: The department/wing of the company. If the company is big enough it might be useful to have more specific information what part of the company the position is referring to. This is varchar (255) and may be entered as any string of choice. This can be null since this info is not always available.
    e. **contact_id**: This refers to the contact in that company. It’s an integer that is a foreign key reference to contact entity (“contact” table). This cannot be null.

4. **Contact**. Written as “contact” when accessing the table. This is information of the person who is our contact within the company.
    a. **c_id**. Contact identification number also is a contact primary key. This is an auto incrementing integer. This cannot be null.
    b. **fname**: The first name of the contact. This is varchar (255) and may be entered as any string of choice. This can be null, since we can just have an email to HR department.
    c. **lname**: The last name of the contact. This is varchar (255) and may be entered as any string of choice. This can be null, since we can just have an email to HR department.

d. **title**: The title that the person that we are in communications with. This is varchar (255) and may be entered as any string of choice. This can be null, since this information is not always provided.
e. **phone**: The phone number of a contact. This is varchar (255) and may be entered as any string of choice. This can be null, since this information is not always provided.
f. **email**: The email of a contact. This is varchar (255) and may be entered as any string of choice. This can be null, since this information is not always provided.

5. **Source**. Written as “source” when accessing the table. This entity represents the name of the source(e.g. Careerbuilder.com, indeed.com, recruiter’s company(name) and the date when the job was published on the source).
    a. **s_id**: An identification id of the table called source and its primary key. This is an auto incrementing integer. It cannot be null.
    b. **name**: The name of the source. This is varchar (255) and may be entered as any string of choice. This cannot be null.

This database also implements the following relationships between entities:

1. Job entity has (is associated with) exactly one application entity and each application entity is associated with exactly one job entity. (one to one relationship).

2. Application has exactly one job entity.

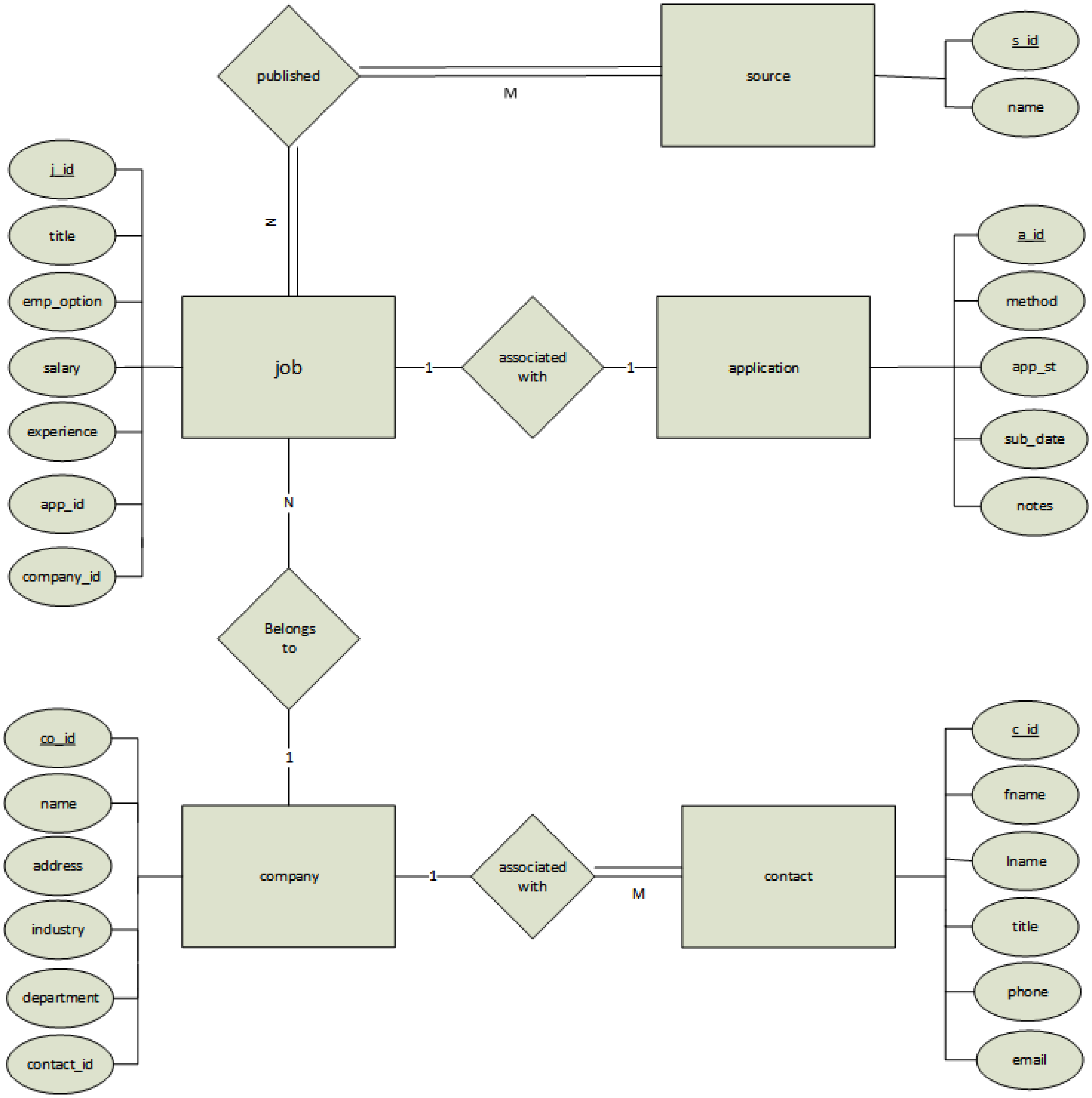
3. Job can be published in at least one or more different sources and each source can publish at least one or more different jobs. (many to many relationship).
    a. In order to implement this relationship we’ve created a joining table. This table is called “published” table. It consists of the primary identification key from the “source” table(s_id) and primary identification key from the “job” table(j_id).
    b. This will be implemented to see what source has the most replies from our submitted applications, so that we can analyze the data and track the most efficient source and concentrate on it for future job applications.

4. A company is attached to zero or more job openings, but each job is attached to exactly one company. One to many relationship.

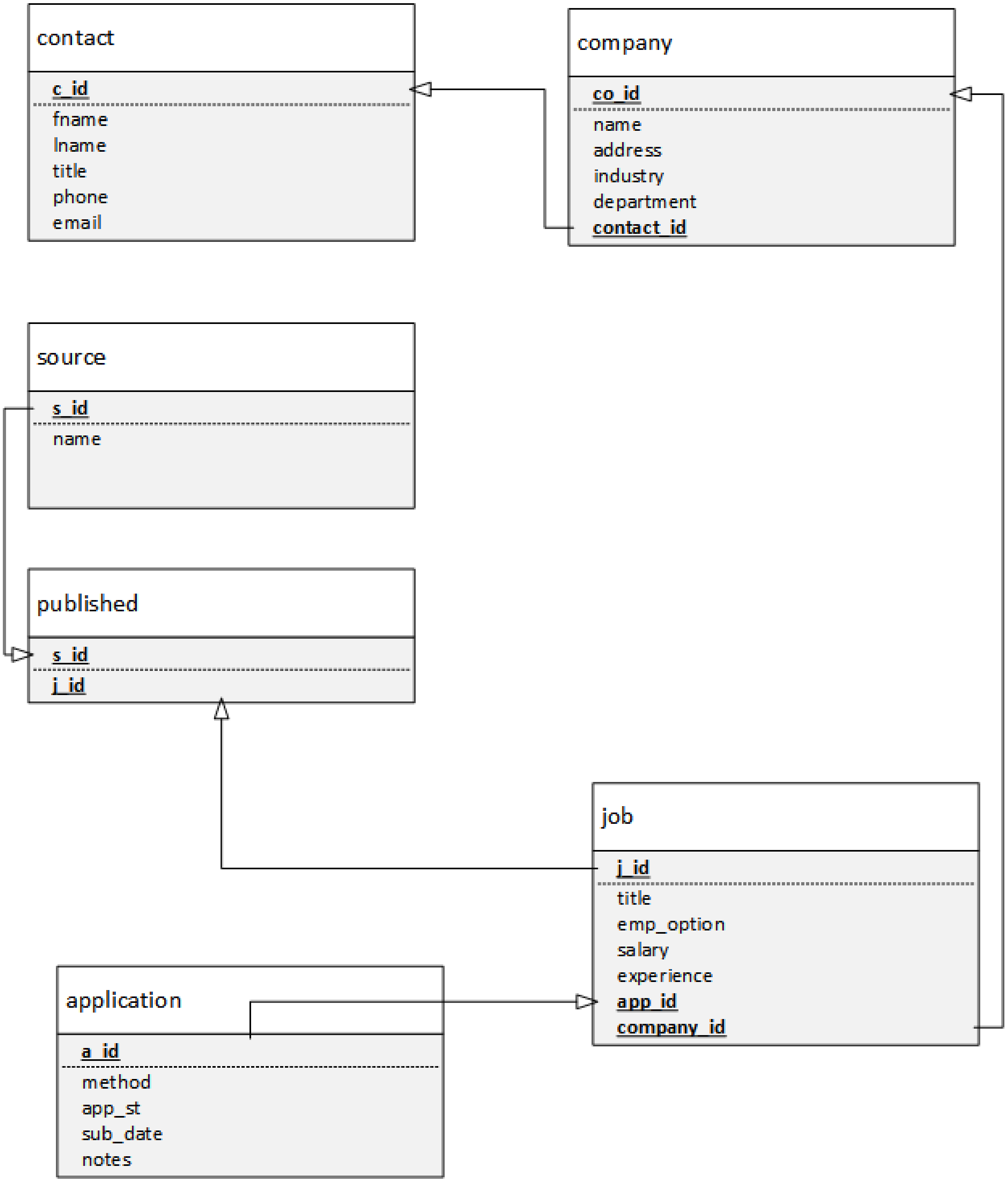
5. A company is associated with at least one or more contact(person) who is responsible for communication, and each contact is associated with one company.

III. ER Diagram of the Database

- **job table is related (specifies) to exactly one application**
  - each job has exactly one application and therefore the status of it.
- **job** table (position) is published in at least one or more sources(publications)
  - every job has to belong to some (at least one) source.
- **source** table can have publications(sources) that have multiple jobs.
- **company** table can (belongs to) have zero or more jobs (from the job table).
  - a company might not have a position available (but it makes sense to keep it for potential future position if we have contact information or we are interested in pursuing employment from specific company).
- **job table (position) belongs to exactly one company.**
  - each job can only be from company.
- **company table** is associated with at least one or more contacts(from contact table).
  - we might be dealing with multiple contacts from the same company.
- **contact** table is associated with exactly one company.


published
source
s_id
name
M
N
j_id
title
emp_option
salary
experience
app_id
company_id
job
1
associated with
1
application
a_id
method
app_st
sub_date
notes
N
Belongs to
1
co_id
name
address
industry
department
contact_id
company
1
associated with
M
contact
c_id
fname
lname
title
phone
email

IV. Database Schema



V. Queries

Section One: Table Creation Queries

Entity Table Creation

```
CREATE TABLE contact (
c_id int AUTO_INCREMENT,
fname varchar(255),
lname varchar(255),
title varchar(255),
phone varchar(255),
email varchar(255),
PRIMARY KEY(c_id)
)ENGINE = InnoDB;
```

```
CREATE TABLE application (
a_id int AUTO_INCREMENT NOT NULL,
method varchar(255),
app_st varchar(255),
sub_date varchar(255) NOT NULL,
notes varchar(255) NOT NULL,
PRIMARY KEY(a_id),
UNIQUE KEY application (sub_date, notes)
)ENGINE = InnoDB;
```

```
CREATE TABLE source (
s_id int AUTO_INCREMENT NOT NULL,
name varchar(255) NOT NULL,
PRIMARY KEY(s_id)
)ENGINE = InnoDB;
```

```
CREATE TABLE company (
co_id int AUTO_INCREMENT NOT NULL,
name varchar(255) NOT NULL,
address varchar(255),
industry varchar(255),
department varchar(255),
contact_id int,
FOREIGN KEY(contact_id) REFERENCES contact(c_id) ON DELETE SET NULL ON UPDATE
CASCADE,
PRIMARY KEY(co_id)
)ENGINE = InnoDB;
```

```
CREATE TABLE job (
j_id int AUTO_INCREMENT NOT NULL,
title varchar(255) NOT NULL,
emp_option varchar(255),
salary int,
experience varchar(255),
app_id int NOT NULL,
company_id int NOT NULL,
FOREIGN KEY(app_id) REFERENCES application(a_id),
FOREIGN KEY(company_id) REFERENCES company(co_id),
PRIMARY KEY(j_id)
)ENGINE = InnoDB;
```

Many-to-Many Table Creation

```
CREATE TABLE published(
s_id INT NOT NULL ,
j_id INT NOT NULL ,
FOREIGN KEY (s_id) REFERENCES source(s_id) ,
FOREIGN KEY (j_id) REFERENCES job(j_id)
) ENGINE = INNODB;
```

Initial Starting Data: Table Insertion Code

Contact Table General Code:

```
INSERT INTO contact (fname, lname, title, phone, email) VALUES ([First Name],[Last Name],[Title],[Phone number],[Email address]);
```

Contact Table Code Used to Insert Starting Data:

```
INSERT INTO contact (fname, lname, title, phone, email) VALUES ("Diana","Johnson","HR Manager","201-645-6512","company@gmail.com");

INSERT INTO contact (fname, lname, title, phone, email) VALUES ("Val","Smith","HR Manager","201-645-6512","ups@gmail.com");

INSERT INTO contact (fname, lname, title, phone, email) VALUES ("William","Miller","Owner","212-645-5555","fedex@gmail.com");

INSERT INTO contact (fname, lname, title, phone, email) VALUES ("Tylor","Martin","HR Manager","201-645-6544","usps@gmail.com");

INSERT INTO contact (fname, lname, title, phone, email) VALUES ("Walter","White","Owner","674-645-3132","notech@gmail.com");
```

Application Table General Code:

```
INSERT INTO application (method, app_st, sub_date,notes) VALUES ([Internet/manual submission],[Status of application],[submission date],[relevant info]);
```

Application Table Code Used to Insert Starting Data:

```
INSERT INTO application (method, app_st, sub_date,notes) VALUES ("Internet","No-reply","11-14-2015","Need to follow up");

INSERT INTO application (method, app_st, sub_date,notes) VALUES ("Internet","Scheduled Phone Interview","12-14-2015", "Prepare for C#");

INSERT INTO application (method, app_st, sub_date,notes) VALUES ("Internet","No-reply","11-28-2015","Reapply in 2 months");

INSERT INTO application (method, app_st, sub_date,notes) VALUES ("Internet","Recieved confirmation","01-14-2016","Need to follow up");
INSERT INTO application (method, app_st, sub_date,notes) VALUES
("Internet","No-reply","02-14-2016","Deadend");
```

Source Table General Code:

```
INSERT INTO source (name) VALUES ([Name of the source]);
```

Source Table Code Used to Insert Starting Data:

```
INSERT INTO source (name) VALUES ("indeed.com");

INSERT INTO source (name) VALUES ("careerbuilder.com");

INSERT INTO source (name) VALUES ("craigslist.com");

INSERT INTO source (name) VALUES ("jobs.com");

INSERT INTO source (name) VALUES ("monster.com");

INSERT INTO source (name) VALUES ("dice.com");

INSERT INTO source (name) VALUES ("culintro.com ");

INSERT INTO source (name) VALUES ("efinancialcareers.com ");

INSERT INTO source (name) VALUES ("jobsonthemenu.com");

INSERT INTO source (name) VALUES ("healthcarejobsite.com ");

INSERT INTO source (name) VALUES ("stackoverflow.com");
```

Company Table General Code:

```
INSERT INTO company (name,address, industry, department, contact_id)
VALUES ([Name of the Company],[Address City and State],[Type of Industry],[Department],
(SELECT c_id FROM contact WHERE fname=[first name] AND lname=[last name]);
```

Company Table Code Used to Insert Starting Data:

```
INSERT INTO company (name,address, industry, department, contact_id)
VALUES ("Metus Aenean Corp.","Hexham, NY"," Medical","Public Relations",
(SELECT c_id FROM contact WHERE fname="Diana" AND lname="Johnson"));
```

```
INSERT INTO company (name,address, industry, department, contact_id)
VALUES ("Lobortis Class Ltd","Rocky Mountain House, CA","Tech","Asset Management",
(SELECT c_id FROM contact WHERE fname="Val" AND lname="Smith"));

INSERT INTO company (name,address, industry, department, contact_id)
VALUES ("Egestas Ltd","Monacilioni, NY"," Financial","Accounting",
(SELECT c_id FROM contact WHERE fname="William" AND lname="Miller"));

INSERT INTO company (name,address, industry, department, contact_id)
VALUES ("Condimentum Eget Volutpat Ltd","Cumberland, CA","Tech","Tech Support",
(SELECT c_id FROM contact WHERE fname="Tylor" AND lname="Martin"));

INSERT INTO company (name,address, industry, department, contact_id)
VALUES ("Phasellus Dapibus LLP","Hagen"," Medical ","Tech Support",
(SELECT c_id FROM contact WHERE fname="Walter" AND lname="White"));
```

Job Table General Code:

```
INSERT INTO job (title, emp_option, salary, experience, app_id, company_id)
VALUES ([Name of Position],[Full-time/Part-time/Internship],[Potential Salary],[level of work
experience],
(SELECT a_id FROM application WHERE sub_date=[Date of submission] AND notes=[Relevant
Info],
(SELECT co_id FROM company WHERE name=[Name of the company]);
```

Job Table Code Used to Insert Starting Data:

```
INSERT INTO job (title, emp_option, salary, experience, app_id, company_id)
VALUES ("Junior .NET developer","Full-time",45000,"6 months",
(SELECT a_id FROM application WHERE sub_date="11-14-2015" AND notes="Need to follow
up"),
(SELECT co_id FROM company WHERE name="Metus Aenean Corp."));

INSERT INTO job (title, emp_option, salary, experience, app_id, company_id)
VALUES ("C# developer","Full-time",47000,"1 year",
(SELECT a_id FROM application WHERE sub_date="12-14-2015" AND notes="Prepare for
C#"),
(SELECT co_id FROM company WHERE name="Lobortis Class Ltd"));

INSERT INTO job (title, emp_option, salary, experience, app_id, company_id)
VALUES ("Junior programmer","Part-time",30000,"6 months",
(SELECT a_id FROM application WHERE sub_date="11-28-2015" AND notes="Reapply in 2
months"),
(SELECT co_id FROM company WHERE name="Egestas Ltd"));

INSERT INTO job (title, emp_option, salary, experience, app_id, company_id)
VALUES ("Entry level .NET developer","Internship",NULL ," ",
(SELECT a_id FROM application WHERE sub_date="01-14-2016" AND notes="Need to follow
up"),
(SELECT co_id FROM company WHERE name="Condimentum Eget Volutpat Ltd"));

INSERT INTO job (title, emp_option, salary, experience, app_id, company_id)
VALUES ("Web Developer","Full-time",60000,"2 years",
(SELECT a_id FROM application WHERE sub_date="02-14-2016" AND notes="Deadend"),
(SELECT co_id FROM company WHERE name="Phasellus Dapibus LLP"));
```

Published Table General Code:

```
INSERT INTO published (s_id,j_id)
VALUES ((SELECT s_id FROM source WHERE name=[Source's Name] ,
(SELECT j_id from job WHERE title=["TITLE OF THE JOB"]));
```

Published Table Code Used to Insert Starting Data:

```
INSERT INTO published (s_id,j_id)
VALUES ((SELECT s_id FROM source WHERE name="indeed.com") ,
(SELECT j_id from job WHERE title="Web Developer"));

INSERT INTO published (s_id,j_id)
VALUES ((SELECT s_id FROM source WHERE name="jobs.com") ,
(SELECT j_id from job WHERE title="Entry level .NET developer"));

INSERT INTO published (s_id,j_id)
VALUES ((SELECT s_id FROM source WHERE name="careerbuilder.com") ,
(SELECT j_id from job WHERE title="C# developer"));

INSERT INTO published (s_id,j_id)
VALUES ((SELECT s_id FROM source WHERE name="indeed.com"),
(SELECT j_id from job WHERE title="Junior .NET developer"));

INSERT INTO published (s_id,j_id)
VALUES ((SELECT s_id FROM source WHERE name="indeed.com"),
(SELECT j_id from job WHERE title="Junior programmer"));
```

Section Two: General Use Queries

Basic Queries

Get all Job Information: Search by Job Title (Full Query)

```
SELECT j.title, j.emp_option, j.salary,j.experience, a.method,a.app_st,a.sub_date, a.notes,
co.name as 'Company Name',co.address,co.industry,co.department, c.fname,c.lname,
c.title,c.phone,c.email, s.name as 'Name of Source' FROM job j
INNER JOIN published p ON p.j_id = j.j_id
INNER JOIN source s ON s.s_id = p.s_id
INNER JOIN application a ON j.j_id=a.a_id
INNER JOIN company co ON j.j_id = co.co_id
INNER JOIN contact c ON c.c_id=co.co_id
WHERE [table.attribute] = [SearchValue];
```

Get all Jobs

```
SELECT j.title, j.emp_option, j.salary,j.experience, a.notes,a.app_st,
co.name,co.address,co.industry,c.fname,c.lname,
c.phone,c.email, s.name FROM job j
INNER JOIN published p ON p.j_id = j.j_id
INNER JOIN source s ON s.s_id = p.s_id
INNER JOIN application a ON j.j_id=a.a_id
INNER JOIN company co ON j.j_id = co.co_id
INNER JOIN contact c ON c.c_id=co.co_id
```

View the most recently 3 added jobs

```
SELECT j.title, j.emp_option, j.salary,j.experience, a.notes,a.app_st,
co.name,co.address,co.industry,c.fname,c.lname,
c.phone,c.email, s.name FROM job j
INNER JOIN published p ON p.j_id = j.j_id
INNER JOIN source s ON s.s_id = p.s_id
INNER JOIN application a ON j.j_id=a.a_id
INNER JOIN company co ON j.j_id = co.co_id
INNER JOIN contact c ON c.c_id=co.co_id
ORDER BY j.j_id DESC
LIMIT 3
```

All the jobs sorted by the application status

```
SELECT
j.title,j.emp_option,j.salary,j.experience,a.notes,a.app_st,co.address,co.name,co.industry,
c.fname,c.lname,c.phone,c.email,s.name from job j
INNER JOIN company co on co.co_id=j.j_id
INNER JOIN application a on a.a_id=j.app_id
INNER JOIN contact c on c.c_id=co.contact_id
INNER JOIN published p on p.j_id=j.j_id
INNER JOIN source s on s.s_id=p.s_id
WHERE a.app_st = [Application Status Selection]
```

All the jobs sorted by the Employment option

```
SELECT
j.title,j.emp_option,j.salary,j.experience,a.notes,a.app_st,co.address,co.name,co.industry,
c.fname,c.lname,c.phone,c.email,s.name from job j
INNER JOIN company co on co.co_id=j.j_id
INNER JOIN application a on a.a_id=j.app_id
INNER JOIN contact c on c.c_id=co.contact_id
INNER JOIN published p on p.j_id=j.j_id
INNER JOIN source s on s.s_id=p.s_id
WHERE j.emp_option =  [Employment Option Selection]
```

All the jobs sorted by the Required Job experience

```
SELECT
j.title,j.emp_option,j.salary,j.experience,a.notes,a.app_st,co.address,co.name,co.industry,
c.fname,c.lname,c.phone,c.email,s.name from job j
INNER JOIN company co on co.co_id=j.j_id
INNER JOIN application a on a.a_id=j.app_id
INNER JOIN contact c on c.c_id=co.contact_id
INNER JOIN published p on p.j_id=j.j_id
INNER JOIN source s on s.s_id=p.s_id
WHERE j.experience =[Required  Experience Selection]
```

All the jobs sorted by the Industry

```
SELECT
j.title,j.emp_option,j.salary,j.experience,a.notes,a.app_st,co.address,co.name,co.industry,
c.fname,c.lname,c.phone,c.email,s.name from job j
INNER JOIN company co on co.co_id=j.j_id
INNER JOIN application a on a.a_id=j.app_id
INNER JOIN contact c on c.c_id=co.contact_id
INNER JOIN published p on p.j_id=j.j_id
INNER JOIN source s on s.s_id=p.s_id
WHERE j.experience =[Industry Selection]
```

All the jobs sorted by the salary More Than.

```
SELECT
j.title,j.emp_option,j.salary,j.experience,a.notes,a.app_st,co.address,co.name,co.industry,
c.fname,c.lname,c.phone,c.email,s.name from job j
INNER JOIN company co on co.co_id=j.j_id
INNER JOIN application a on a.a_id=j.app_id
INNER JOIN contact c on c.c_id=co.contact_id
INNER JOIN published p on p.j_id=j.j_id
INNER JOIN source s on s.s_id=p.s_id
WHERE j.salary > =[Amount]
```

All the jobs sorted by the salary within a range of salary.

```
SELECT
j.title,j.emp_option,j.salary,j.experience,a.notes,a.app_st,co.address,co.name,co.industry,
c.fname,c.lname,c.phone,c.email,s.name from job j
INNER JOIN company co on co.co_id=j.j_id
INNER JOIN application a on a.a_id=j.app_id
INNER JOIN contact c on c.c_id=co.contact_id
INNER JOIN published p on p.j_id=j.j_id
INNER JOIN source s on s.s_id=p.s_id
WHERE j.salary BETWEEN [lower Amount] AND [Higher Amount]
```

Complex queries:

Select average salaries by Industry

```
SELECT
co.industry, AVG(j.salary)
from job j
INNER JOIN company co on co.co_id=j.j_id
INNER JOIN application a on a.a_id=j.app_id
INNER JOIN contact c on c.c_id=co.contact_id
INNER JOIN published p on p.j_id=j.j_id
INNER JOIN source s on s.s_id=p.s_id
WHERE co.industry = [Industry Selection];
```

Filter Jobs by with [Employment Option Selection] and [Source Name] (Many-to-Many)

```
SELECT
j.title,j.emp_option,j.salary,j.experience,a.notes,a.app_st,co.address,co.name,co.industry,
c.fname,c.lname,c.phone,c.email,s.name from job j
INNER JOIN company co on co.co_id=j.j_id
INNER JOIN application a on a.a_id=j.app_id
INNER JOIN contact c on c.c_id=co.contact_id
INNER JOIN published p on p.j_id=j.j_id
INNER JOIN source s on s.s_id=p.s_id
WHERE j.emp_option=[Employment Option Selection]  AND  s.name=[Name of Job’s Source]
```

Filter Number of Jobs per Source

```
SELECT s.name,j.emp_option, COUNT(j.j_id)
FROM job j
INNER JOIN published p ON p.j_id = j.j_id
INNER JOIN source s ON s.s_id = p.s_id
where s.name=[name of Source]
GROUP BY s.name;
```

All jobs with Employment Option Selection and Company Option Selection and Industry Option Selection combination.

```
SELECT
j.title,j.emp_option,j.salary,j.experience,a.notes,a.app_st,co.address,co.name,co.industry,
c.fname,c.lname,c.phone,c.email,s.name from job j
INNER JOIN company co on co.co_id=j.j_id
INNER JOIN application a on a.a_id=j.app_id
INNER JOIN contact c on c.c_id=co.contact_id
INNER JOIN published p on p.j_id=j.j_id
INNER JOIN source s on s.s_id=p.s_id
HERE j.emp_option=[Employment Option Selection]  AND  co.address=[Company Option
Selection] AND co.Industry=[Industry Option Selection]
```

All jobs with Specific Salary Range and that are not Employment Option(Full/Part-time, Internship)
NOT IN OPTION.

```
SELECT
j.title,j.emp_option,j.salary,j.experience,a.notes,a.app_st,co.address,co.name,co.industry,
c.fname,c.lname,c.phone,c.email,s.name from job j
INNER JOIN company co on co.co_id=j.j_id
INNER JOIN application a on a.a_id=j.app_id
INNER JOIN contact c on c.c_id=co.contact_id
INNER JOIN published p on p.j_id=j.j_id
INNER JOIN source s on s.s_id=p.s_id
WHERE j.salary > [Salary selection] AND j.j_id NOT IN
(SELECT j.j_id FROM job j WHERE j.emp_option =[Employment option )
```

Update Application Status

```
UPDATE application SET app_st=["NEW VALUE"] WHERE sub_date=[Date of Job Submission]
AND notes=[notes for the job];
```

Update Application Notes

```
UPDATE application set notes=[NEW VALUE] where a_id in ( select app_id from job j where
j.title=[Title Of the JOB])
```

Update Application Status

```
UPDATE application set app_st=[NEW VALUE] where a_id in ( select app_id from job j where
j.title=[Title Of the JOB])
```

Update Job’s Title

```
UPDATE job set title=[NEW Title Of the JOB] where title=[Title Of the JOB]
```

Delete Contact from contact Table

```
DELETE FROM contact where fname=[First Name Selection] and lname=[Last Name Selection]
```