Assignment 2
Instructions
Coding Challenges: CareerHub, The Job Board

- Coding Challenges submissions should be done through the partcipants’ Github repository, and the link should be shared with trainers and Hexavarsity.

Problem Statement:
Create SQL Schema from the application, use the class attributes for table column names.
SQL Schema:

Table: Companies
Attributes:

- CompanyID (Primary Key, int): Unique identifier for each company.
- CompanyName (string): The name of the hiring company.
- Location (string): The location of the company.

Table: Jobs
Attributes:

- JobID (Primary Key, int): Unique identifier for each job listing.
- CompanyID (Foreign Key, int): References the CompanyID of the hiring company.
- JobTitle (string): The title of the job.
- JobDescription (text): A detailed description of the job.
- JobLocation (string): The location where the job is based.
- Salary (decimal): The salary offered for the job.
- JobType (string): Type of job (e.g., Full-time, Part-time, Contract).
- PostedDate (datetime): Date and time when the job was posted.

Table: Applicants
Attributes:
ApplicantID (Primary Key, int): Unique identifier for each applicant.

- FirstName (string): The first name of the applicant.
- LastName (string): The last name of the applicant.
- Email (string): The email address of the applicant.
- Phone (string): The phone number of the applicant.
- Resume (text): The applicant's resume or CV (text or file reference).

Table: Applications
Attributes:

- ApplicationID (Primary Key, int): Unique identifier for each job application.
- JobID (Foreign Key, int): References the JobID of the job listing.
- ApplicantID (Foreign Key, int): References the ApplicantID of the applicant.
- ApplicationDate (datetime): Date and time when the application was submitted.
- CoverLetter (text): The applicant's cover letter for the specific job.

Tasks:

1. Provide a SQL script that initializes the database for the Job Board scenario “CareerHub”.

```
        use career;
```

2. Create tables for Companies, Jobs, Applicants and Applications.

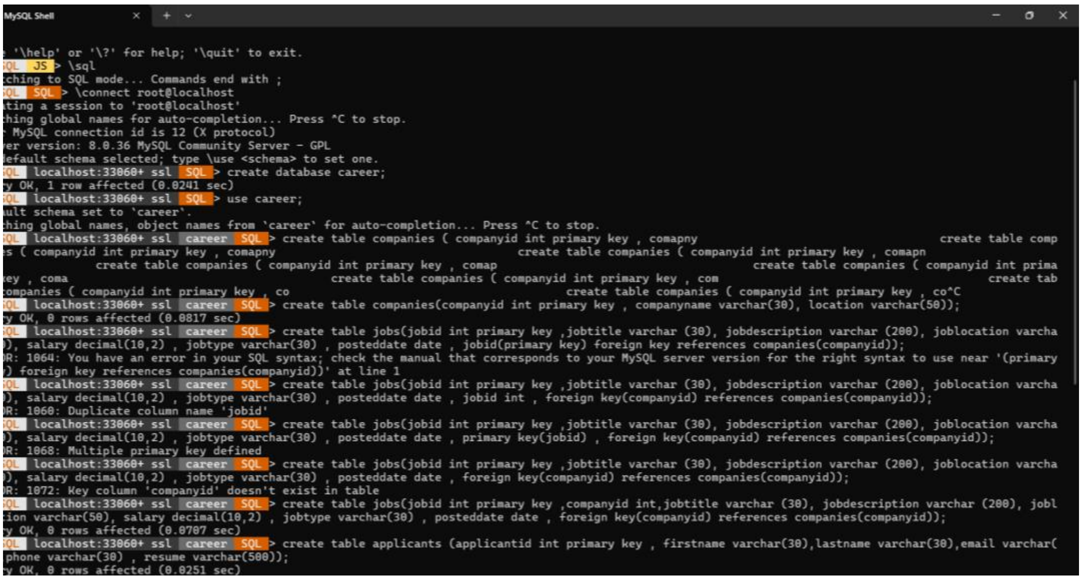
3. Define appropriate primary keys, foreign keys, and constraints.

4. Ensure the script handles potential errors, such as if the database or tables already exist.

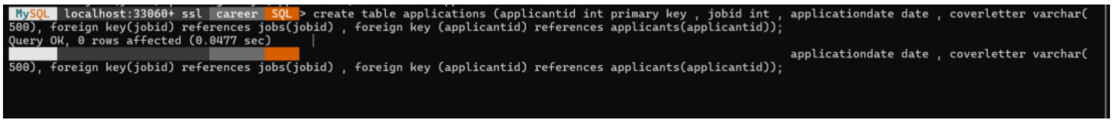
```
        create table companies(companyid int primary key , companyname varchar(30), location
varchar(50));
        create table jobs(jobid int primary key ,companyid int,jobtitle varchar (30), jobdescription
varchar (200), joblocation varchar(50), salary decimal(10,2) , jobtype varchar(30) , posteddate date ,
foreign key(companyid) references companies(companyid));
```

create table applicants (applicantid int primary key , firstname varchar(30),lastname varchar(30),email varchar(40),phone varchar(30) , resume varchar(500));

create table applications (applicantid int primary key , jobid int , applicationdate date , coverletter varchar(500), foreign key(jobid) references jobs(jobid) , foreign key (applicantid) references applicants(applicantid));



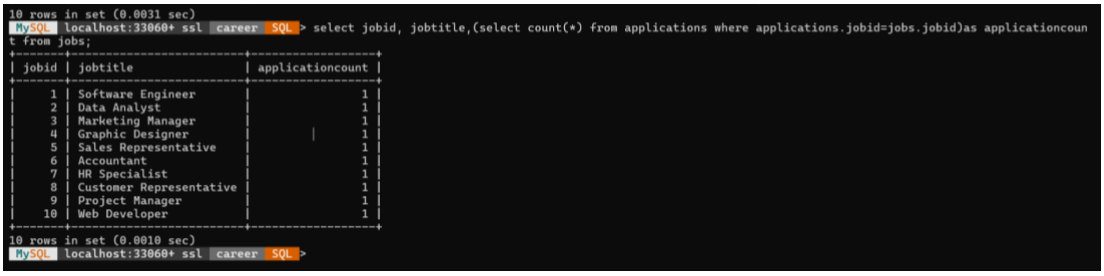
```
e '\help' or '\?' for help; '\quit' to exit.
QL  JS > \sql
tching to SQL mode... Commands end with ;
QL  SQL > \connect root@localhost
ating a session to 'root@localhost'
ching global names for auto-completion... Press ^C to stop.
r MySQL connection id is 12 (X protocol)
ver version: 8.0.36 MySQL Community Server - GPL
default schema selected; type \use <schema> to set one.
QL  localhost:33060+ ssl  SQL > create database career;
ry OK, 1 row affected (0.0241 sec)
QL  localhost:33060+ ssl  SQL > use career;
ult schema set to `career`.
ching global names, object names from `career` for auto-completion... Press ^C to stop.
QL  localhost:33060+ ssl  career  SQL > create table companies ( companyid int primary key , comapny                        create table comp
es ( companyid int primary key , comapny                                  create table companies ( companyid int primary key , comapn
                create table companies ( companyid int primary key , comap                                  create table companies ( companyid int prima
key , coma                                  create table companies ( companyid int primary key , com                                  create tab
ompanies ( companyid int primary key , co                                  create table companies ( companyid int primary key , co^C
QL  localhost:33060+ ssl  career  SQL > create table companies(companyid int primary key , companyname varchar(30), location varchar(50));
ry OK, 0 rows affected (0.0817 sec)
QL  localhost:33060+ ssl  career  SQL > create table jobs(jobid int primary key ,jobtitle varchar (30), jobdescription varchar (200), joblocation varcha
), salary decimal(10,2) , jobtype varchar(30) , posteddate date , jobid(primary key) foreign key references companies(companyid));
OR: 1064: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '(primary
) foreign key references companies(companyid))' at line 1
QL  localhost:33060+ ssl  career  SQL > create table jobs(jobid int primary key ,jobtitle varchar (30), jobdescription varchar (200), joblocation varcha
), salary decimal(10,2) , jobtype varchar(30) , posteddate date , jobid int , foreign key(companyid) references companies(companyid));
OR: 1060: Duplicate column name 'jobid'
QL  localhost:33060+ ssl  career  SQL > create table jobs(jobid int primary key ,jobtitle varchar (30), jobdescription varchar (200), joblocation varcha
), salary decimal(10,2) , jobtype varchar(30) , posteddate date , primary key(jobid) , foreign key(companyid) references companies(companyid));
OR: 1068: Multiple primary key defined
QL  localhost:33060+ ssl  career  SQL > create table jobs(jobid int primary key ,jobtitle varchar (30), jobdescription varchar (200), joblocation varcha
), salary decimal(10,2) , jobtype varchar(30) , posteddate date , foreign key(companyid) references companies(companyid));
OR: 1072: Key column 'companyid' doesn't exist in table
QL  localhost:33060+ ssl  career  SQL > create table jobs(jobid int primary key ,companyid int,jobtitle varchar (30), jobdescription varchar (200), jobl
tion varchar(50), salary decimal(10,2) , jobtype varchar(30) , posteddate date , foreign key(companyid) references companies(companyid));
ry OK, 0 rows affected (0.0707 sec)
QL  localhost:33060+ ssl  career  SQL > create table applicants (applicantid int primary key , firstname varchar(30),lastname varchar(30),email varchar(
 phone varchar(30) , resume varchar(500));
ry OK, 0 rows affected (0.0251 sec)
```



```
MySQL  localhost:33060+ ssl  career  SQL > create table applications (applicantid int primary key , jobid int , applicationdate date , coverletter varchar(
500), foreign key(jobid) references jobs(jobid) , foreign key (applicantid) references applicants(applicantid));
Query OK, 0 rows affected (0.0477 sec)
                                                                                                          applicationdate date , coverletter varchar(
500), foreign key(jobid) references jobs(jobid) , foreign key (applicantid) references applicants(applicantid));
```

5. Write an SQL query to count the number of applications received for each job listing in the "Jobs" table. Display the job title and the corresponding application count. Ensure that it lists all jobs, even if they have no applications.

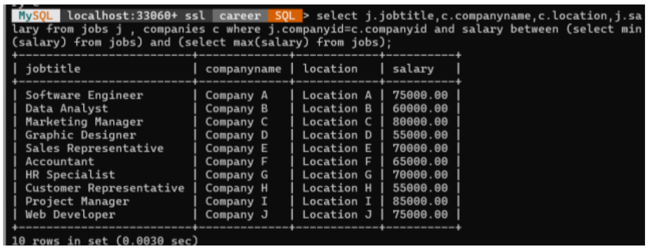
select jobid, jobtitle,(select count(*) from applications where applications.jobid=jobs.jobid)as applicationcount from jobs;



```
10 rows in set (0.0031 sec)
MySQL  localhost:33060+ ssl  career  SQL > select jobid, jobtitle,(select count(*) from applications where applications.jobid=jobs.jobid)as applicationcoun
t from jobs;
+-------+--------------------------+------------------+
| jobid | jobtitle                 | applicationcount |
+-------+--------------------------+------------------+
|     1 | Software Engineer        |                1 |
|     2 | Data Analyst             |                1 |
|     3 | Marketing Manager        |                1 |
|     4 | Graphic Designer         |                1 |
|     5 | Sales Representative     |                1 |
|     6 | Accountant               |                1 |
|     7 | HR Specialist            |                1 |
|     8 | Customer Representative  |                1 |
|     9 | Project Manager          |                1 |
|    10 | Web Developer            |                1 |
+-------+--------------------------+------------------+
10 rows in set (0.0010 sec)
MySQL  localhost:33060+ ssl  career  SQL >
```

6. Develop an SQL query that retrieves job listings from the "Jobs" table within a specified salary range. Allow parameters for the minimum and maximum salary values. Display the job title, company name, location, and salary for each matching job.

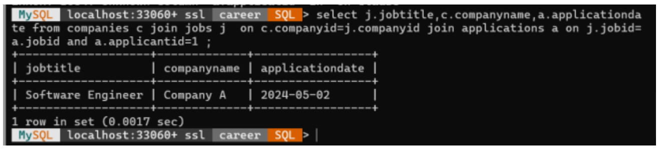
select j.jobtitle,c.companyname,c.location,j.salary from jobs j , companies c where j.companyid=c.companyid and salary between (select min(salary) from jobs) and (select max(salary) from jobs);



```
MySQL  localhost:33060+ ssl  career  SQL > select j.jobtitle,c.companyname,c.location,j.sa
lary from jobs j , companies c where j.companyid=c.companyid and salary between (select min
(salary) from jobs) and (select max(salary) from jobs);
+-------------------------+-------------+------------+----------+
| jobtitle                | companyname | location   | salary   |
+-------------------------+-------------+------------+----------+
| Software Engineer       | Company A   | Location A | 75000.00 |
| Data Analyst            | Company B   | Location B | 60000.00 |
| Marketing Manager       | Company C   | Location C | 80000.00 |
| Graphic Designer        | Company D   | Location D | 55000.00 |
| Sales Representative    | Company E   | Location E | 70000.00 |
| Accountant              | Company F   | Location F | 65000.00 |
| HR Specialist           | Company G   | Location G | 70000.00 |
| Customer Representative | Company H   | Location H | 55000.00 |
| Project Manager         | Company I   | Location I | 85000.00 |
| Web Developer           | Company J   | Location J | 75000.00 |
+-------------------------+-------------+------------+----------+
10 rows in set (0.0030 sec)
```

7. Write an SQL query that retrieves the job application history for a specific applicant. Allow a parameter for the ApplicantID, and return a result set with the job titles, company names, and application dates for all the jobs the applicant has applied to.

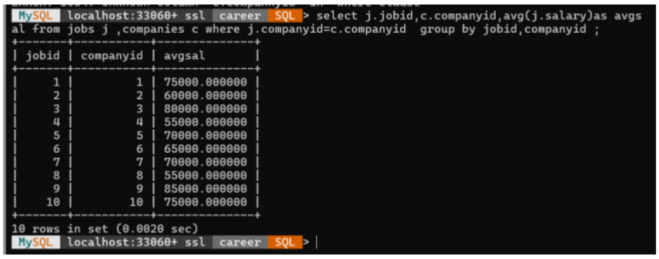
select j.jobtitle,c.companyname,a.applicationdate from companies c join jobs j on c.companyid=j.companyid join applications a on j.jobid=a.jobid where a.applicantid=1 ;



```
MySQL  localhost:33060+ ssl  career  SQL > select j.jobtitle,c.companyname,a.applicationda
te from companies c join jobs j  on c.companyid=j.companyid join applications a on j.jobid=
a.jobid and a.applicantid=1 ;
+-------------------+-------------+-----------------+
| jobtitle          | companyname | applicationdate |
+-------------------+-------------+-----------------+
| Software Engineer | Company A   | 2024-05-02      |
+-------------------+-------------+-----------------+
1 row in set (0.0017 sec)
MySQL  localhost:33060+ ssl  career  SQL > |
```

8. Create an SQL query that calculates and displays the average salary offered by all companies for job listings in the "Jobs" table. Ensure that the query filters out jobs with a salary of zero.

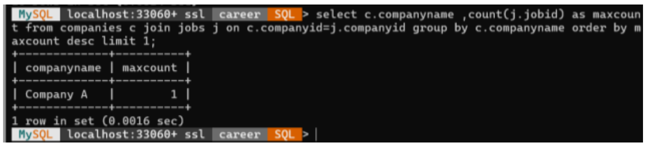
select j.jobid,c.companyid,avg(j.salary)as avgsal from jobs j ,companies c where j.companyid=c.companyid group by jobid,companyid ;



```
MySQL  localhost:33060+ ssl  career  SQL > select j.jobid,c.companyid,avg(j.salary)as avgs
al from jobs j ,companies c where j.companyid=c.companyid  group by jobid,companyid ;
+-------+-----------+--------------+
| jobid | companyid | avgsal       |
+-------+-----------+--------------+
|     1 |         1 | 75000.000000 |
|     2 |         2 | 60000.000000 |
|     3 |         3 | 80000.000000 |
|     4 |         4 | 55000.000000 |
|     5 |         5 | 70000.000000 |
|     6 |         6 | 65000.000000 |
|     7 |         7 | 70000.000000 |
|     8 |         8 | 55000.000000 |
|     9 |         9 | 85000.000000 |
|    10 |        10 | 75000.000000 |
+-------+-----------+--------------+
10 rows in set (0.0020 sec)
MySQL  localhost:33060+ ssl  career  SQL > |
```

9. Write an SQL query to identify the company that has posted the most job listings. Display the company name along with the count of job listings they have posted. Handle ties if multiple companies have the same maximum count.

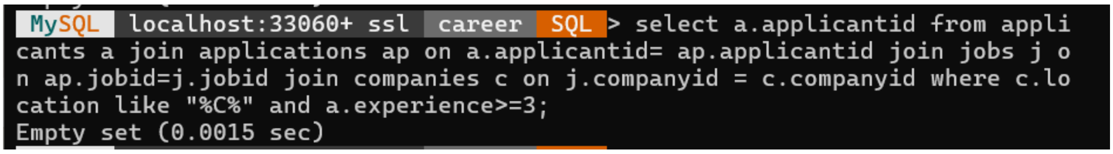
select c.companyname ,count(j.jobid) as maxcount from companies c join jobs j on c.companyid=j.companyid group by c.companyname order by maxcount desc limit 1;



```
MySQL  localhost:33060+ ssl  career  SQL > select c.companyname ,count(j.jobid) as maxcoun
t from companies c join jobs j on c.companyid=j.companyid group by c.companyname order by m
axcount desc limit 1;
+-------------+----------+
| companyname | maxcount |
+-------------+----------+
| Company A   |        1 |
+-------------+----------+
1 row in set (0.0016 sec)
MySQL  localhost:33060+ ssl  career  SQL > |
```

10. Find the applicants who have applied for positions in companies located in 'CityX' and have at least 3 years of experience.

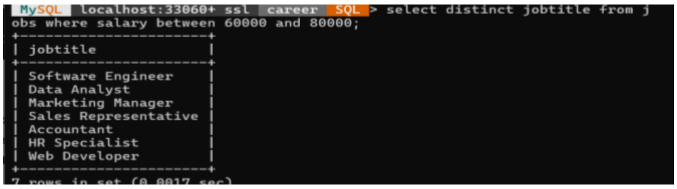
select a.applicantid from applicants a join applications ap on a.applicantid= ap.applicantid join jobs j on ap.jobid=j.jobid join companies c on j.companyid = c.companyid where c.location like "%C%" and a.experience>=3;



```
 MySQL  localhost:33060+ ssl  career  SQL > select a.applicantid from appli
cants a join applications ap on a.applicantid= ap.applicantid join jobs j o
n ap.jobid=j.jobid join companies c on j.companyid = c.companyid where c.lo
cation like "%C%" and a.experience>=3;
Empty set (0.0015 sec)
```

11. Retrieve a list of distinct job titles with salaries between $60,000 and $80,000.

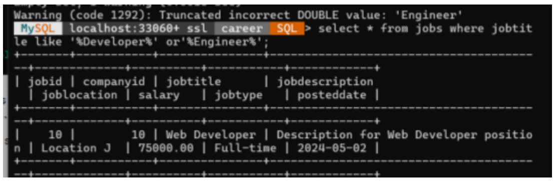
select distinct jobtitle from jobs where salary between 60000 and 80000;



```
 MySQL  localhost:33060+ ssl  career  SQL > select distinct jobtitle from j
obs where salary between 60000 and 80000;
+----------------------+
| jobtitle             |
+----------------------+
| Software Engineer    |
| Data Analyst         |
| Marketing Manager    |
| Sales Representative |
| Accountant           |
| HR Specialist        |
| Web Developer        |
+----------------------+
7 rows in set (0.0017 sec)
```

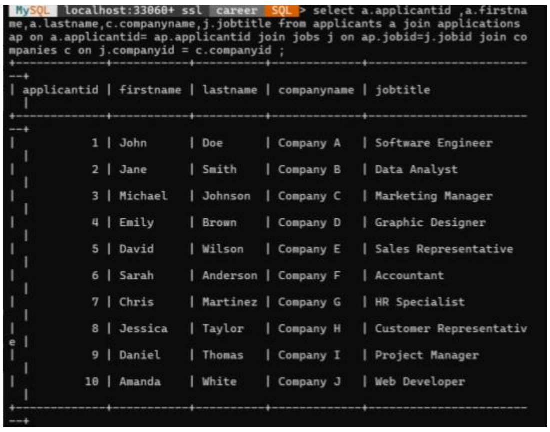
12. Find the jobs that have not received any applications.

select * from jobs where jobid not in(select jobid from applications);



```
Warning (code 1292): Truncated incorrect DOUBLE value: 'Engineer'
 MySQL  localhost:33060+ ssl  career  SQL > select * from jobs where jobtit
le like '%Developer%' or'%Engineer%';
+-------+-----------+---------------+--------------------------------------
--+-------------+----------+-----------+------------+
| jobid | companyid | jobtitle      | jobdescription
  | joblocation | salary   | jobtype   | posteddate |
+-------+-----------+---------------+--------------------------------------
--+-------------+----------+-----------+------------+
|    10 |        10 | Web Developer | Description for Web Developer positio
n | Location J  | 75000.00 | Full-time | 2024-05-02 |
+-------+-----------+---------------+--------------------------------------
--+-------------+----------+-----------+------------+
```

13. Retrieve a list of job applicants along with the companies they have applied to and the positions they have applied for.

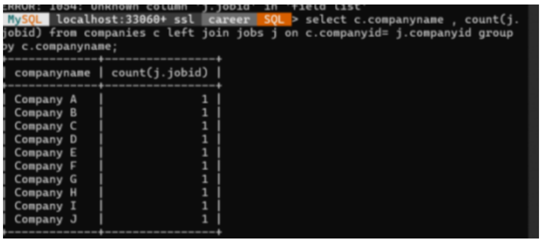


```
 MySQL  localhost:33060+ ssl  career  SQL > select a.applicantid ,a.firstna
me,a.lastname,c.companyname,j.jobtitle from applicants a join applications
ap on a.applicantid= ap.applicantid join jobs j on ap.jobid=j.jobid join co
mpanies c on j.companyid = c.companyid ;
+-------------+-----------+----------+-------------+-----------------------
--+
| applicantid | firstname | lastname | companyname | jobtitle
  |
+-------------+-----------+----------+-------------+-----------------------
--+
|           1 | John      | Doe      | Company A   | Software Engineer
  |
|           2 | Jane      | Smith    | Company B   | Data Analyst
  |
|           3 | Michael   | Johnson  | Company C   | Marketing Manager
  |
|           4 | Emily     | Brown    | Company D   | Graphic Designer
  |
|           5 | David     | Wilson   | Company E   | Sales Representative
  |
|           6 | Sarah     | Anderson | Company F   | Accountant
  |
|           7 | Chris     | Martinez | Company G   | HR Specialist
  |
|           8 | Jessica   | Taylor   | Company H   | Customer Representativ
e |
|           9 | Daniel    | Thomas   | Company I   | Project Manager
  |
|          10 | Amanda    | White    | Company J   | Web Developer
  |
+-------------+-----------+----------+-------------+-----------------------
--+
```

select a.applicantid ,a.firstname,a.lastname,c.companyname,j.jobtitle from applicants a join applications ap on a.applicantid= ap.applicantid join jobs j on ap.jobid=j.jobid join companies c on j.companyid = c.companyid ;

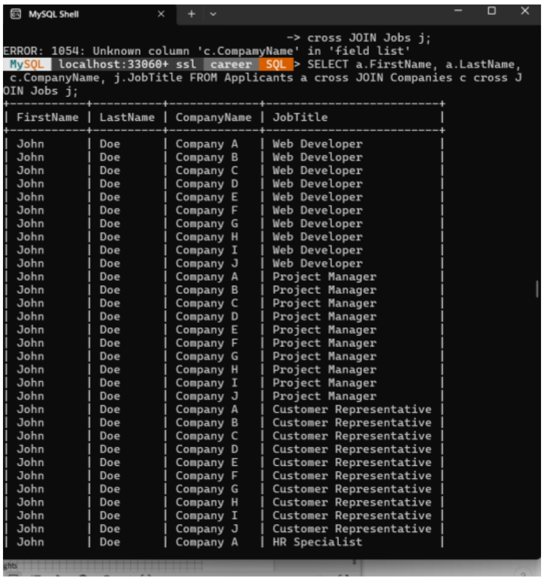
14. Retrieve a list of companies along with the count of jobs they have posted, even if they have not received any applications.

select c.companyname , count(j.jobid) from companies c left join jobs j on c.companyid= j.companyid group by c.companyname;



```
ERROR: 1054: Unknown column 'j.jobid' in 'field list'
MySQL  localhost:33060+ ssl  career  SQL > select c.companyname , count(j.
jobid) from companies c left join jobs j on c.companyid= j.companyid group
by c.companyname;
+-------------+---------------+
| companyname | count(j.jobid) |
+-------------+---------------+
| Company A   |             1 |
| Company B   |             1 |
| Company C   |             1 |
| Company D   |             1 |
| Company E   |             1 |
| Company F   |             1 |
| Company G   |             1 |
| Company H   |             1 |
| Company I   |             1 |
| Company J   |             1 |
+-------------+---------------+
```

15. List all applicants along with the companies and positions they have applied for, including those who have not applied.

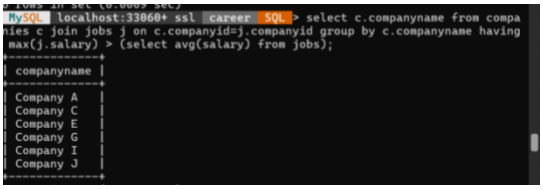


```
                                 -> cross JOIN Jobs j;
ERROR: 1054: Unknown column 'c.CompamyName' in 'field list'
 MySQL  localhost:33060+ ssl  career  SQL > SELECT a.FirstName, a.LastName,
 c.CompanyName, j.JobTitle FROM Applicants a cross JOIN Companies c cross J
OIN Jobs j;
+-----------+----------+-------------+-------------------------+
| FirstName | LastName | CompanyName | JobTitle                |
+-----------+----------+-------------+-------------------------+
| John      | Doe      | Company A   | Web Developer           |
| John      | Doe      | Company B   | Web Developer           |
| John      | Doe      | Company C   | Web Developer           |
| John      | Doe      | Company D   | Web Developer           |
| John      | Doe      | Company E   | Web Developer           |
| John      | Doe      | Company F   | Web Developer           |
| John      | Doe      | Company G   | Web Developer           |
| John      | Doe      | Company H   | Web Developer           |
| John      | Doe      | Company I   | Web Developer           |
| John      | Doe      | Company J   | Web Developer           |
| John      | Doe      | Company A   | Project Manager         |
| John      | Doe      | Company B   | Project Manager         |
| John      | Doe      | Company C   | Project Manager         |
| John      | Doe      | Company D   | Project Manager         |
| John      | Doe      | Company E   | Project Manager         |
| John      | Doe      | Company F   | Project Manager         |
| John      | Doe      | Company G   | Project Manager         |
| John      | Doe      | Company H   | Project Manager         |
| John      | Doe      | Company I   | Project Manager         |
| John      | Doe      | Company J   | Project Manager         |
| John      | Doe      | Company A   | Customer Representative |
| John      | Doe      | Company B   | Customer Representative |
| John      | Doe      | Company C   | Customer Representative |
| John      | Doe      | Company D   | Customer Representative |
| John      | Doe      | Company E   | Customer Representative |
| John      | Doe      | Company F   | Customer Representative |
| John      | Doe      | Company G   | Customer Representative |
| John      | Doe      | Company H   | Customer Representative |
| John      | Doe      | Company I   | Customer Representative |
| John      | Doe      | Company J   | Customer Representative |
| John      | Doe      | Company A   | HR Specialist           |
```

SELECT a.FirstName, a.LastName, c.CompanyName, j.JobTitle FROM Applicants a cross JOIN Companies c cross JOIN Jobs j;

16. Find companies that have posted jobs with a salary higher than the average salary of all jobs.

select c.companyname from companies c join jobs j on c.companyid=j.companyid group by c.companyname having max(j.salary) > (select avg(salary) from jobs);

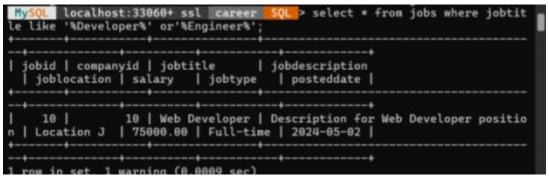


```
 MySQL  localhost:33060+ ssl  career  SQL > select c.companyname from compa
nies c join jobs j on c.companyid=j.companyid group by c.companyname having
 max(j.salary) > (select avg(salary) from jobs);
+-------------+
| companyname |
+-------------+
| Company A   |
| Company C   |
| Company E   |
| Company G   |
| Company I   |
| Company J   |
+-------------+
```

17. Display a list of applicants with their names and a concatenated string of their city and state.
select firstname,lastname , concat(city , ' ', state) as location from applicants;

Empty set

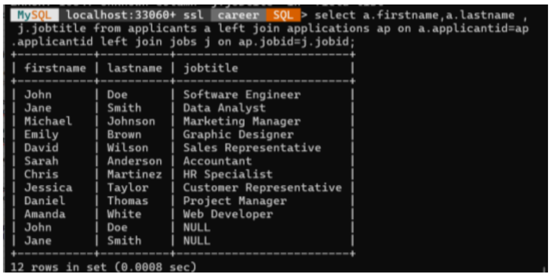
18. Retrieve a list of jobs with titles containing either 'Developer' or 'Engineer'.
select * from jobs where jobtitle like '%Developer%' or'%Engineer%';



```
MySQL  localhost:33060+ ssl  career  SQL > select * from jobs where jobtit
le like '%Developer%' or'%Engineer%';
+-------+-----------+---------------+--------------------------------------
--+-------------+----------+-----------+------------+
| jobid | companyid | jobtitle      | jobdescription
  | joblocation | salary   | jobtype   | posteddate |
+-------+-----------+---------------+--------------------------------------
--+-------------+----------+-----------+------------+
|    10 |        10 | Web Developer | Description for Web Developer positio
n | Location J  | 75000.00 | Full-time | 2024-05-02 |
+-------+-----------+---------------+--------------------------------------
--+-------------+----------+-----------+------------+
1 row in set, 1 warning (0.0009 sec)
```

19. Retrieve a list of applicants and the jobs they have applied for, including those who have not applied and jobs without applicants.

select a.firstname,a.lastname , j.jobtitle from applicants a left join applications ap on a.applicantid=ap.applicantid left join jobs j on ap.jobid=j.jobid;



```
MySQL  localhost:33060+ ssl  career  SQL > select a.firstname,a.lastname ,
 j.jobtitle from applicants a left join applications ap on a.applicantid=ap
.applicantid left join jobs j on ap.jobid=j.jobid;
+-----------+----------+-------------------------+
| firstname | lastname | jobtitle                |
+-----------+----------+-------------------------+
| John      | Doe      | Software Engineer       |
| Jane      | Smith    | Data Analyst            |
| Michael   | Johnson  | Marketing Manager       |
| Emily     | Brown    | Graphic Designer        |
| David     | Wilson   | Sales Representative    |
| Sarah     | Anderson | Accountant              |
| Chris     | Martinez | HR Specialist           |
| Jessica   | Taylor   | Customer Representative |
| Daniel    | Thomas   | Project Manager         |
| Amanda    | White    | Web Developer           |
| John      | Doe      | NULL                    |
| Jane      | Smith    | NULL                    |
+-----------+----------+-------------------------+
12 rows in set (0.0008 sec)
```

20. List all combinations of applicants and companies where the company is in a specific city and the applicant has more than 2 years of experience. For example: city=Chennai

select a.firstname,a.lastname , j.jobtitle ,c.companyname from applicants a join applications ap on a.applicantid=ap.applicantid join jobs j on ap.jobid=j.jobid join companies c on c.companyid=j.companyid where c.location like '%C%' and a.experience >2;

Empty set