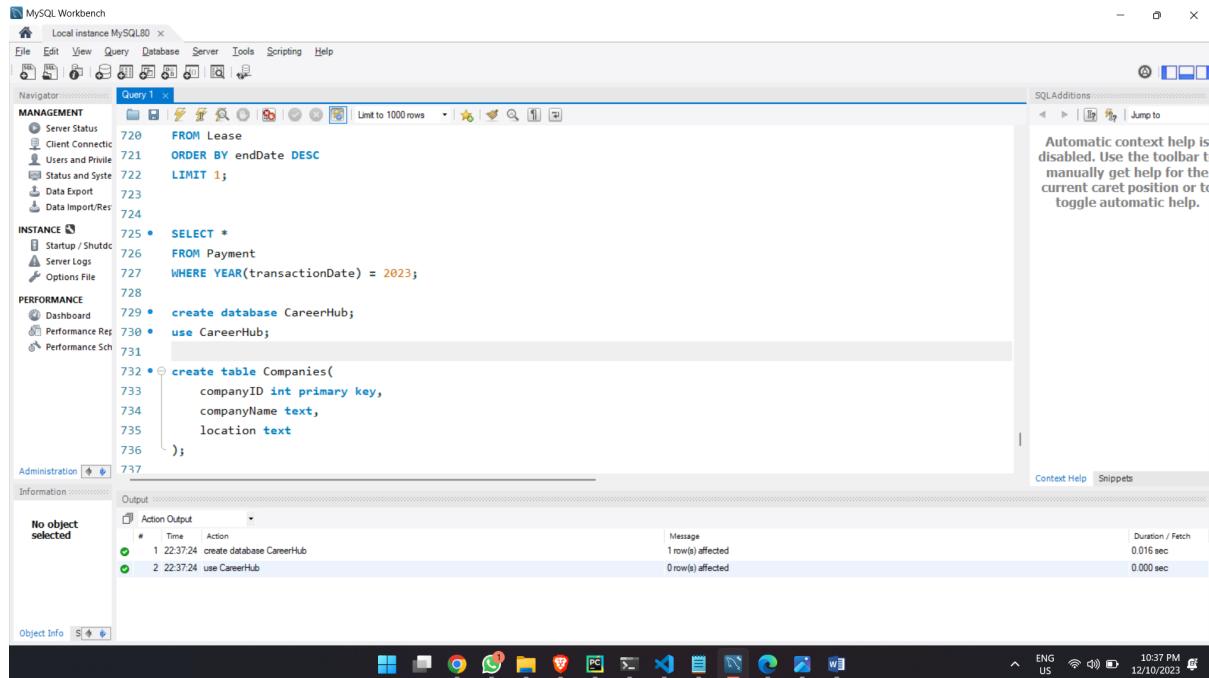


Career Hub

Q1. Provide a SQL script that initializes the database for the Job Board scenario "CareerHub".



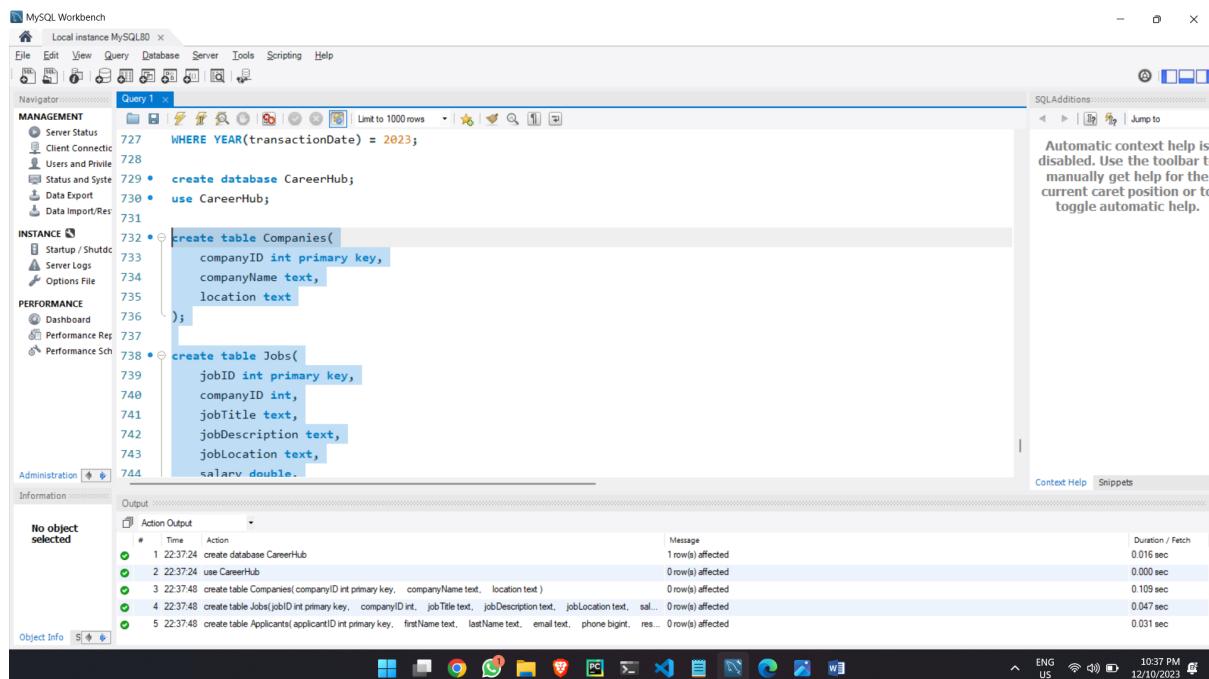
The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL script:

```
720     FROM Lease
721     ORDER BY endDate DESC
722     LIMIT 1;
723
724 •   SELECT *
725     FROM Payment
726     WHERE YEAR(transactionDate) = 2023;
727
728 •   create database CareerHub;
729 •   use CareerHub;
730
731 •   create table Companies(
732     companyID int primary key,
733     companyName text,
734     location text
735 );
736
737
```

The output pane shows the results of the commands:

Action	Time	Message	Duration / Fetch
create database CareerHub	1 22:37:24	1 row(s) affected	0.016 sec
use CareerHub	2 22:37:24	0 row(s) affected	0.000 sec

Q2. Create tables for Companies, Jobs, Applicants and Applications. 3. Define appropriate primary keys, foreign keys, and constraints.



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL script:

```
727     WHERE YEAR(transactionDate) = 2023;
728
729 •   create database CareerHub;
730 •   use CareerHub;
731
732 •   create table Companies(
733     companyID int primary key,
734     companyName text,
735     location text
736 );
737
738 •   create table Jobs(
739     jobID int primary key,
740     companyID int,
741     jobTitle text,
742     jobDescription text,
743     jobLocation text,
744     salary double.
745
746
747
```

The output pane shows the results of the commands:

Action	Time	Message	Duration / Fetch
create database CareerHub	1 22:37:24	1 row(s) affected	0.016 sec
use CareerHub	2 22:37:24	0 row(s) affected	0.000 sec
create table Companies(companyID int primary key, companyName text, location text)	3 22:37:48	0 row(s) affected	0.109 sec
create table Jobs(jobID int primary key, companyID int, jobTitle text, jobDescription text, jobLocation text, salary double.)	4 22:37:48	0 row(s) affected	0.047 sec
create table Applicants(applicantID int primary key, firstName text, lastName text, email text, phone bigint, res...	5 22:37:48	0 row(s) affected	0.031 sec

Q3. Define appropriate primary keys, foreign keys, and constraints.

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Local instance MySQL80, showing Server Status, Client Connectic, Users and Privile, Status and Syst, Data Export, Data Import/Res, and Administration.
- Query Editor:** Query 1, displaying SQL code for creating a database and tables. The code includes:

```
727 WHERE YEAR(transactionDate) = 2023;
728
729 • create database CareerHub;
730 • use CareerHub;
731
732 • create table Companies(
    companyID int primary key,
    companyName text,
    location text
);
733
734
735
736 );
737
738 • create table Jobs(
    jobID int primary key,
    companyID int,
    jobTitle text,
    jobDescription text,
    jobLocation text,
    salary double.
739
740
741
742
743
744
```
- Output Window:** Action Output table showing the execution results of each query. The table has columns: Action, Time, Message, Duration / Fetch. The results are:

Action	Time	Message	Duration / Fetch
1 22:37:24 create database CareerHub	22:37:24	1 row(s) affected	0.016 sec
2 22:37:24 use CareerHub	22:37:24	0 row(s) affected	0.000 sec
3 22:37:48 create table Companies(companyID int primary key, companyName text, location text)	22:37:48	0 row(s) affected	0.109 sec
4 22:37:48 create table Jobs(jobID int primary key, companyID int, jobTitle text, jobDescription text, jobLocation text, salary double.)	22:37:48	0 row(s) affected	0.047 sec
5 22:37:48 create table Applicants(applicantID int primary key, firstName text, lastName text, email text, phone bigint, res...	22:37:48	0 row(s) affected	0.031 sec
- System Bar:** Shows icons for various applications and the system clock (10:37 PM, 12/10/2023).

Q4. Ensure the script handles potential errors, such as if the database or tables already exist.

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Local instance MySQL80, showing Server Status, Client Connectic, Users and Privile, Status and Syst, Data Export, Data Import/Res, and Administration.
- Query Editor:** Query 1, displaying SQL code for creating a database and tables. The code includes:

```
727 WHERE YEAR(transactionDate) = 2023;
728
729 • create database CareerHub;
730 • use CareerHub;
731
732 • create table Companies(
    companyID int primary key,
    companyName text,
    location text
);
733
734
735
736 );
737
738 • create table Jobs(
    jobID int primary key,
    companyID int,
    jobTitle text,
    jobDescription text,
    jobLocation text,
    salary double.
739
740
741
742
743
744
```
- Output Window:** Action Output table showing the execution results of each query. The table has columns: Action, Time, Message, Duration / Fetch. The results are:

Action	Time	Message	Duration / Fetch
1 22:37:24 create database CareerHub	22:37:24	1 row(s) affected	0.016 sec
2 22:37:24 use CareerHub	22:37:24	0 row(s) affected	0.000 sec
3 22:37:48 create table Companies(companyID int primary key, companyName text, location text)	22:37:48	0 row(s) affected	0.109 sec
4 22:37:48 create table Jobs(jobID int primary key, companyID int, jobTitle text, jobDescription text, jobLocation text, salary double.)	22:37:48	0 row(s) affected	0.047 sec
5 22:37:48 create table Applicants(applicantID int primary key, firstName text, lastName text, email text, phone bigint, res...	22:37:48	0 row(s) affected	0.031 sec
- System Bar:** Shows icons for various applications and the system clock (10:37 PM, 12/10/2023).

Q5. 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.

```

MySQL Workbench
Local instance MySQL80 x
File Edit View Query Database Server Tools Scripting Help
Navigator Query 1 x
MANAGEMENT
    Server Status
    Client Connector
    Users and Privileges
    Status and System
    Data Export
    Data Import/Restore
1218     FROM Jobs
1219     WHERE jobTitle LIKE '%Developer%' OR jobTitle LIKE '%Engineer%';
1220
1221
1222 •     SELECT J.jobID, J.jobTitle, COUNT(A.applicationID) AS applicationCount
1223     FROM Jobs J
1224     LEFT JOIN Applications A ON J.jobID = A.jobID
1225     GROUP BY J.jobID, J.jobTitle;
INSTANCE
    Startup / Shutdown
    Server Logs
    Options File
PERFORMANCE
    Dashboard
    Performance Report
    Performance Schema
Result Grid | Filter Rows: Export: Wrap Cell Content: 
jobID jobTitle applicationCount
1 Software Engineer 1
2 Data Scientist 1
3 Product Manager 1
4 Sales Representative 1
5 UX/UI Designer 1
6 Marketing Specialist 1
7 Network Administrator 1
8 Financial Analyst 1
9 Research Scientist 1
10 Project Manager 1
Administration
Information
No object selected
Object Info Session
Result 276 x
Output
Action Output
# Time Action Message Duration / Fetch
1 17:28:47 use CareerHub 0 rows affected 0.000 sec
2 17:29:12 SELECT J.jobID, J.jobTitle, COUNT(A.applicationID) AS applicationCount FROM Jobs J LEFT JOIN Applications A ON ... 10 rows returned 0.015 sec / 0.000 sec
12:13:2023 5:29 PM

```

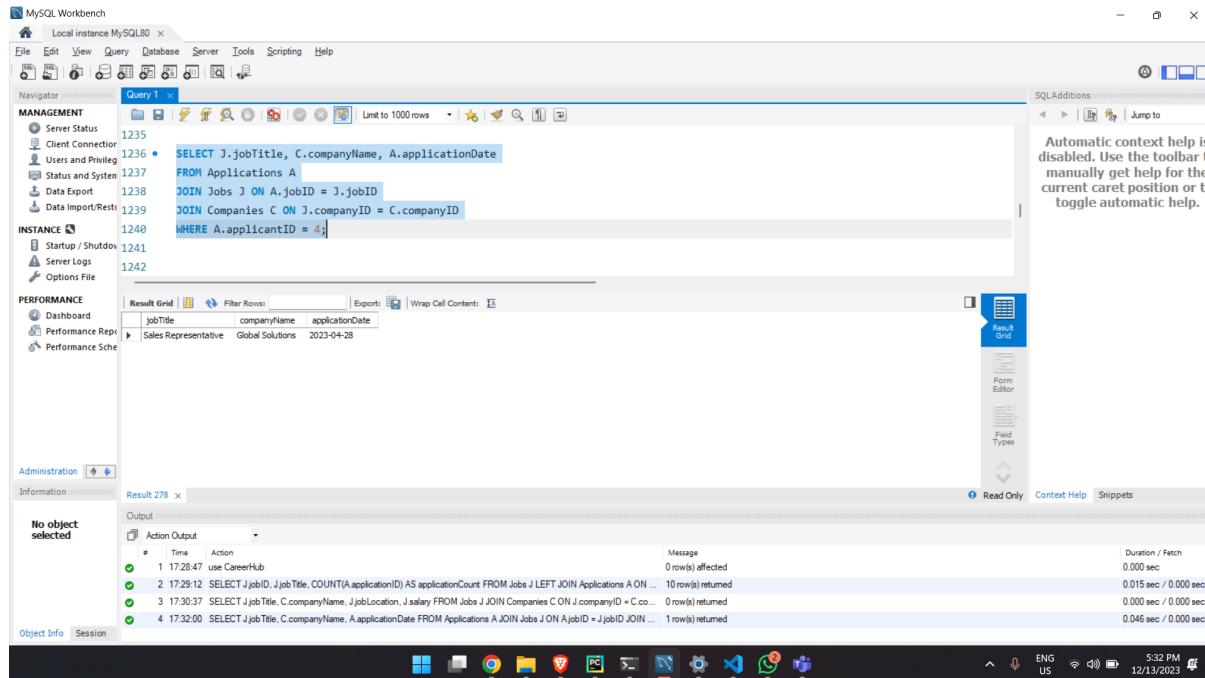
Q6. Develop an SQL query that retrieves job listings from the "Jobs" table within a specified salary range. Allow parameters for the minimum and maximum salary values. Display the job title, company name, location, and salary for each matching job.

```

MySQL Workbench
Local instance MySQL80 x
File Edit View Query Database Server Tools Scripting Help
Navigator Query 1 x
MANAGEMENT
    Server Status
    Client Connector
    Users and Privileges
    Status and System
    Data Export
    Data Import/Restore
1229 •     SELECT J.jobTitle, C.companyName, J.jobLocation, J.salary
1230     FROM Jobs J
1231     JOIN Companies C ON J.companyID = C.companyID
1232     WHERE J.salary BETWEEN 8000 AND 12000;
1233
1234
1235
1236
INSTANCE
    Startup / Shutdown
    Server Logs
    Options File
PERFORMANCE
    Dashboard
    Performance Report
    Performance Schema
Result Grid | Filter Rows: Export: Wrap Cell Content: 
jobTitle companyName jobLocation salary
Administration
Information
No object selected
Object Info Session
Result 277 x
Output
Action Output
# Time Action Message Duration / Fetch
1 17:28:47 use CareerHub 0 rows affected 0.000 sec
2 17:29:12 SELECT J.jobID, J.jobTitle, COUNT(A.applicationID) AS applicationCount FROM Jobs J LEFT JOIN Applications A ON ... 10 rows returned 0.015 sec / 0.000 sec
3 17:30:37 SELECT J.jobTitle, C.companyName, J.jobLocation, J.salary FROM Jobs J JOIN Companies C ON J.companyID = C.co... 0 rows returned 0.000 sec / 0.000 sec
12:13:2023 5:30 PM

```

Q7. 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.



The screenshot shows the MySQL Workbench interface with the following details:

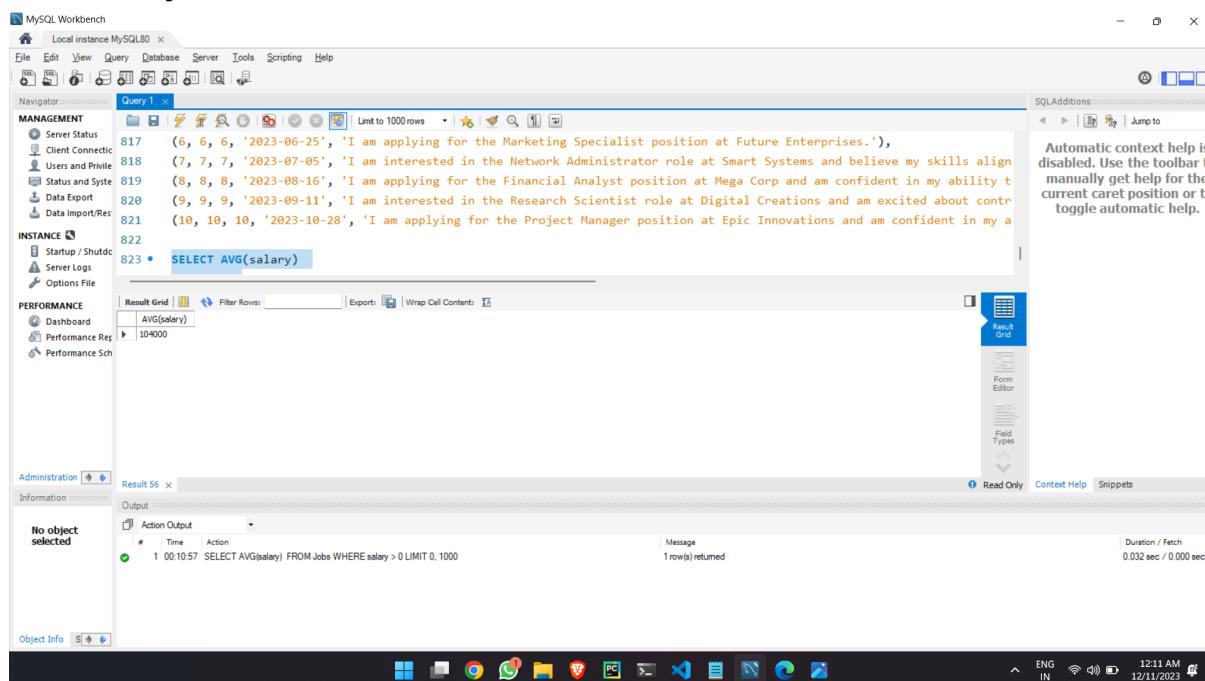
- Query Editor:** Contains the following SQL code:


```
1235
1236 • SELECT J.jobTitle, C.companyName, A.applicationDate
1237   FROM Applications A
1238     JOIN Jobs J ON A.jobID = J.jobID
1239     JOIN Companies C ON J.companyID = C.companyID
1240   WHERE A.applicantID = 4;
```
- Result Grid:** Displays the output of the query:

jobTitle	companyName	applicationDate
Sales Representative	Global Solutions	2023-04-28
- Output Window:** Shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
1	17:28:47	use CareerHub	0 row(s) affected	0.000 sec
2	17:29:12	SELECT jobID, jobTitle, COUNT(applicationID) AS applicationCount FROM Jobs J LEFT JOIN Applications A ON J.jobID = A.jobID WHERE A.applicantID = 4	10 rows(s) returned	0.015 sec / 0.000 sec
3	17:30:37	SELECT J.jobTitle, C.companyName, J.jobLocation, J.salary FROM Jobs J JOIN Companies C ON J.companyID = C.companyID WHERE J.jobID = 10	0 row(s) returned	0.000 sec / 0.000 sec
4	17:32:00	SELECT J.jobTitle, C.companyName, A.applicationDate FROM Applications A JOIN Jobs J ON A.jobID = J.jobID JOIN Companies C ON J.companyID = C.companyID WHERE A.applicantID = 4	1 row(s) returned	0.046 sec / 0.000 sec

Q8. 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.



The screenshot shows the MySQL Workbench interface with the following details:

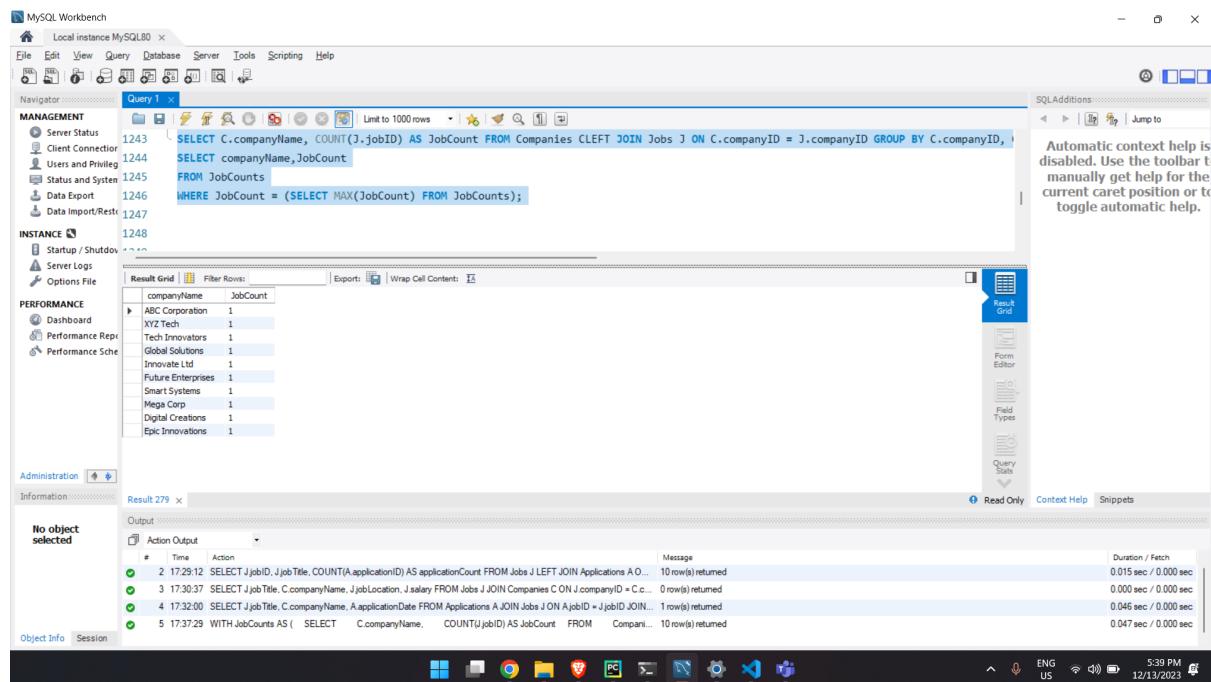
- Query Editor:** Contains the following SQL code:


```
817 (6, 6, 6, '2023-06-25', 'I am applying for the Marketing Specialist position at Future Enterprises.')
818 (7, 7, 7, '2023-07-05', 'I am interested in the Network Administrator role at Smart Systems and believe my skills align with the requirements.')
819 (8, 8, 8, '2023-08-16', 'I am applying for the Financial Analyst position at Mega Corp and am confident in my ability to handle complex financial data.')
820 (9, 9, 9, '2023-09-11', 'I am interested in the Research Scientist role at Digital Creations and am excited about contributing to their innovative projects.')
821 (10, 10, 10, '2023-10-28', 'I am applying for the Project Manager position at Epic Innovations and am confident in my leadership skills.')
822
823 • SELECT AVG(salary)
```
- Result Grid:** Displays the output of the query:

Avg(salary)
104000
- Output Window:** Shows the execution log with the following entry:

#	Time	Action	Message	Duration / Fetch
1	00:10:57	SELECT AVG(salary) FROM Jobs WHERE salary > 0 LIMIT 0, 1000	1 row(s) returned	0.032 sec / 0.000 sec

Q9. 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.



```

MySQL Workbench
Local instance MySQL80 x
File Edit View Query Database Server Tools Scripting Help
Navigator Query 1 x
MANAGEMENT
    Server Status
    Client Connector
    Users and Privileges
    Status and System
    Data Export
    Data Import/Restore
INSTANCE
    Startup / Shutdown
    Server Logs
    Options File
PERFORMANCE
    Dashboard
    Performance Reports
    Performance Schema
Administration
Information
Result 279 x
No object selected
Output
Action Output
    Time Action
    2 17:29:12 SELECT J.jobID, J.jobTitle, COUNT(A.applicationID) AS applicationCount FROM Jobs J LEFT JOIN Applications A ON J.jobID = A.jobID GROUP BY J.jobID ORDER BY applicationCount DESC LIMIT 1; 10 row(s) returned
    3 17:30:37 SELECT J.jobTitle, C.companyName, J.jobLocation, J.salary FROM Jobs J JOIN Companies C ON J.companyID = C.companyID WHERE J.jobTitle = 'Software Developer'; 0 row(s) returned
    4 17:32:00 SELECT J.jobTitle, C.companyName, A.applicationDate FROM Applications A JOIN Jobs J ON A.jobID = J.jobID JOIN Companies C ON J.companyID = C.companyID WHERE J.jobTitle = 'Software Developer' AND C.companyName = 'Tech Innovators'; 1 row(s) returned
    5 17:37:29 WITH JobCounts AS ( SELECT C.companyName, COUNT(J.jobID) AS JobCount FROM Companies C JOIN Jobs J ON C.companyID = J.companyID GROUP BY C.companyName ) SELECT C.companyName, JobCount FROM JobCounts WHERE JobCount = (SELECT MAX(JobCount) FROM JobCounts);
    Duration / Fetch
    0.015 sec / 0.000 sec
    0.000 sec / 0.000 sec
    0.046 sec / 0.000 sec
    0.047 sec / 0.000 sec
    0.047 sec / 0.000 sec
Object Info Session
ENG US 5:39 PM 12/13/2023

```

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query is:

```

SELECT C.companyName, COUNT(J.jobID) AS JobCount FROM Companies C LEFT JOIN Jobs J ON C.companyID = J.companyID GROUP BY C.companyName, JobCount;
SELECT companyName, JobCount
FROM JobCounts
WHERE JobCount = (SELECT MAX(JobCount) FROM JobCounts);

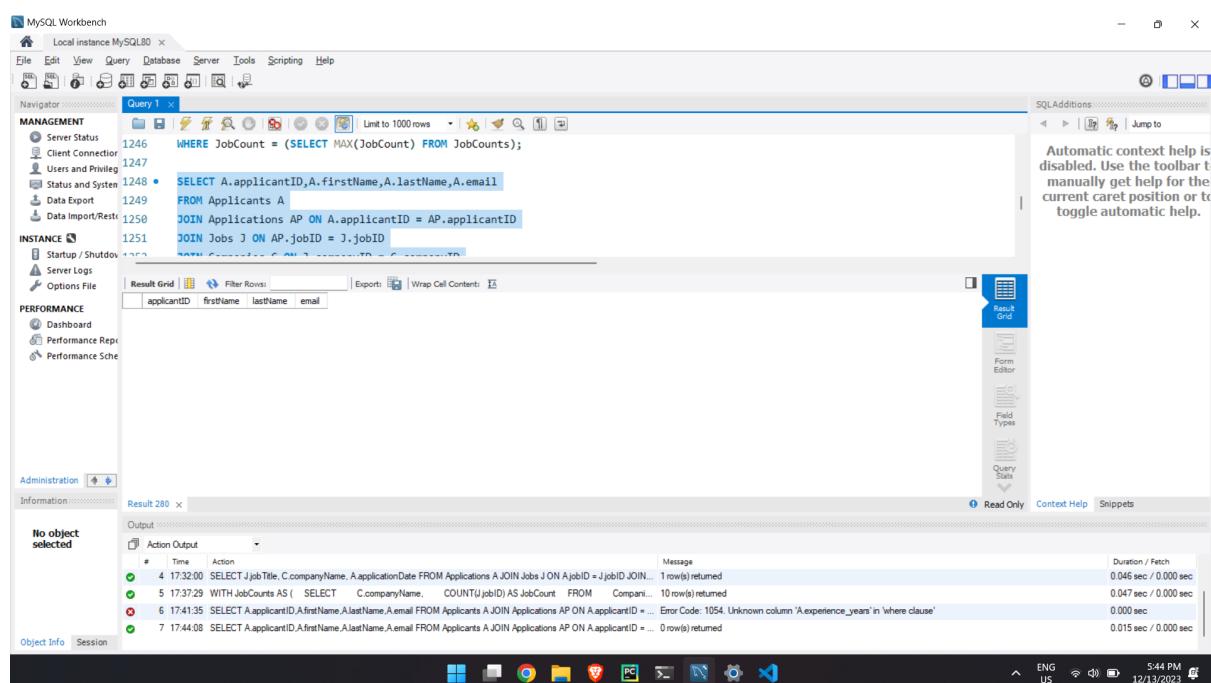
```

The results grid displays the following data:

companyName	JobCount
ABC Corporation	1
XYZ Tech	1
Tech Innovators	1
Global Solutions	1
Innovate Ltd.	1
Future Enterprises	1
Smart Systems	1
Mega Corp	1
Digital Creations	1
Epic Innovations	1

The status bar at the bottom right indicates the time as 5:39 PM and the date as 12/13/2023.

Q10. Find the applicants who have applied for positions in companies located in 'CityX' and have at least 3 years of experience.



```

MySQL Workbench
Local instance MySQL80 x
File Edit View Query Database Server Tools Scripting Help
Navigator Query 1 x
MANAGEMENT
    Server Status
    Client Connector
    Users and Privileges
    Status and System
    Data Export
    Data Import/Restore
INSTANCE
    Startup / Shutdown
    Server Logs
    Options File
PERFORMANCE
    Dashboard
    Performance Reports
    Performance Schema
Administration
Information
Result 280 x
No object selected
Output
Action Output
    Time Action
    4 17:32:00 SELECT JobTitle, C.companyName, A.applicationDate FROM Applications A JOIN Jobs J ON A.jobID = J.jobID JOIN Companies C ON J.companyID = C.companyID WHERE C.companyName = 'Tech Innovators'; 10 row(s) returned
    5 17:37:29 WITH JobCounts AS ( SELECT C.companyName, COUNT(J.jobID) AS JobCount FROM Companies C JOIN Jobs J ON C.companyID = J.companyID GROUP BY C.companyName ) SELECT C.companyName, JobCount FROM JobCounts WHERE JobCount = (SELECT MAX(JobCount) FROM JobCounts);
    Duration / Fetch
    0.046 sec / 0.000 sec
    0.047 sec / 0.000 sec
    0.000 sec
    0.015 sec / 0.000 sec
Object Info Session
ENG US 5:44 PM 12/13/2023

```

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query is:

```

WHERE JobCount = (SELECT MAX(JobCount) FROM JobCounts);
SELECT A.applicantID, A.firstName, A.lastName, A.email
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.companyName = 'Tech Innovators' AND A.experience_years >= 3;

```

The results grid displays the following data:

applicantID	firstName	lastName	email
1	John	Doe	john.doe@example.com
2	Jane	Doe	jane.doe@example.com
3	Mike	Smith	mike.smith@example.com
4	Sarah	Johnson	sarah.johnson@example.com
5	David	Wilson	da...@example.com

The status bar at the bottom right indicates the time as 5:44 PM and the date as 12/13/2023.

Q11. Retrieve a list of distinct job titles with salaries between \$60,000 and \$80,000.

The screenshot shows the MySQL Workbench interface. In the Query Editor (Query 1), the following SQL code is executed:

```
820   (9, 9, 9, '2023-09-11', 'I am interested in the Research Scientist role at Digital Creations and am excited about contr
821   (10, 10, 10, '2023-10-28', 'I am applying for the Project Manager position at Epic Innovations and am confident in my a
823 • SELECT AVG(salary)
824   FROM Jobs
825   WHERE salary > 0;
826
```

The Result Grid shows the output:

JobTitle	salary
Sales Representative	80000

In the Output pane, the Action Output table shows the execution details:

#	Time	Action	Message	Duration / Fetch
1	00:10:57	SELECT AVG(salary) FROM Jobs WHERE salary > 0 LIMIT 0, 1000	1 row(s) returned	0.032 sec / 0.000 sec
2	00:14:48	SELECT DISTINCT jobTitle, salary FROM Jobs WHERE salary BETWEEN 60000 AND 80000 LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec

Q12. Find the jobs that have not received any applications.

The screenshot shows the MySQL Workbench interface. In the Query Editor (Query 1), the following SQL code is executed:

```
828   FROM Jobs
829   WHERE salary BETWEEN 60000 AND 80000;
830
831 • SELECT *
832   FROM Jobs
833   WHERE jobID NOT IN (SELECT DISTINCT jobID FROM Applications);
834
```

The Result Grid shows the output:

jobID	companyID	jobTitle	jobDescription	jobLocation	salary	jobType	postedDate

In the Output pane, the Action Output table shows the execution details:

#	Time	Action	Message	Duration / Fetch
1	00:10:57	SELECT AVG(salary) FROM Jobs WHERE salary > 0 LIMIT 0, 1000	1 row(s) returned	0.032 sec / 0.000 sec
2	00:14:48	SELECT DISTINCT jobTitle, salary FROM Jobs WHERE salary BETWEEN 60000 AND 80000 LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
3	00:17:13	SELECT jobID, jobTitle, jobDescription, jobLocation, salary, jobType, postedDate FROM Jobs WHERE jobID NOT IN (S...)	0 row(s) returned	0.000 sec / 0.000 sec
4	00:17:33	SELECT * FROM Jobs WHERE jobID NOT IN (SELECT DISTINCT jobID FROM Applications) LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec

Q13. Retrieve a list of job applicants along with the companies they have applied to and the positions they have applied for.

```

MySQL Workbench - Local instance MySQL80.x
File Edit View Query Database Server Tools Scripting Help
Navigator Query1 x SQL Additions
MANAGEMENT
INSTANCE
PERFORMANCE
Administration
Information
No object selected
Object Info Session
Result 281 x
Output
Action Output
# Time Action Duration / Fetch
1 5 17:37:29 WITH JobCounts AS ( SELECT C.companyName, COUNT(J.jobID) AS JobCount FROM Companies C JOIN Jobs J ON AP.jobID = J.jobID 0.047 sec / 0.000 sec
2 6 17:41:35 SELECT A.applicantID, A.firstName, A.lastName, A.email FROM Applicants A JOIN Applications AP ON A.applicantID = AP.applicantID ... Error Code: 1054. Unknown column 'A.experience_years' in 'where clause' 0.000 sec
3 7 17:44:08 SELECT A.applicantID, A.firstName, A.lastName, A.email FROM Applicants A JOIN Applications AP ON A.applicantID = AP.applicantID ... 0 rows(s) returned 0.015 sec / 0.000 sec
4 8 17:45:54 SELECT A.applicantID, A.firstName, A.lastName, C.companyName, J.jobTitle FROM Applicants A JOIN Applications AP ON A.applicantID = AP.applicantID ... 10 row(s) returned 0.062 sec / 0.000 sec
Result Grid | Filter Rows: Export: Wrap Cell Content: 
applicantID firstName lastName companyName jobTitle
1 John Doe ABC Corporation Software Engineer
2 Jane Smith XYZ Tech Data Scientist
3 Robert Johnson Tech Innovators Product Manager
4 Sarah Brown Global Solutions Sales Representative
5 David Lee Innovate Ltd UX/UI Designer
6 Laura Hall Future Enterprises Marketing Specialist
7 Michael Davis Smart Systems Network Administrator
8 Emma White Mega Corp Financial Analyst
9 William Taylor Digital Creations Research Scientist
Read Only Context Help Snippets
ENG US 5:46 PM 12/13/2023

```

Q14. Retrieve a list of companies along with the count of jobs they have posted, even if they have not received any applications.

```

MySQL Workbench - Local instance MySQL80.x
File Edit View Query Database Server Tools Scripting Help
Navigator Query1 x SQL Additions
MANAGEMENT
INSTANCE
PERFORMANCE
Administration
Information
No object selected
Object Info Session
Result 282 x
Output
Action Output
# Time Action Duration / Fetch
1 6 17:41:35 SELECT A.applicantID, A.firstName, A.lastName, A.email FROM Applicants A JOIN Applications AP ON A.applicantID = AP.applicantID ... Error Code: 1054. Unknown column 'A.experience_years' in 'where clause' 0.000 sec
2 7 17:44:08 SELECT A.applicantID, A.firstName, A.lastName, A.email FROM Applicants A JOIN Applications AP ON A.applicantID = AP.applicantID ... 0 rows(s) returned 0.015 sec / 0.000 sec
3 8 17:45:54 SELECT A.applicantID, A.firstName, A.lastName, C.companyName, COUNT(J.jobID) AS jobCount FROM Applicants A JOIN Applications AP ON A.applicantID = AP.applicantID ... 10 row(s) returned 0.062 sec / 0.000 sec
4 9 17:47:41 SELECT C.companyID, C.companyName, COUNT(J.jobID) AS jobCount FROM Companies C LEFT JOIN Jobs J ON C.companyID = J.companyID ... 10 row(s) returned 0.000 sec / 0.000 sec
Result Grid | Filter Rows: Export: Wrap Cell Content: 
companyID companyName jobCount
1 ABC Corporation 1
2 XYZ Tech 1
3 Tech Innovators 1
4 Global Solutions 1
5 Innovate Ltd 1
6 Future Enterprises 1
7 Smart Systems 1
8 Mega Corp 1
9 Digital Creations 1
Read Only Context Help Snippets
ENG US 5:48 PM 12/13/2023

```

Q15. List all applicants along with the companies and positions they have applied for, including those who have not applied.

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query is:

```
1264 GROUP BY C.companyID, C.companyName;
1265
1266
1267 • SELECT A.applicantID, A.firstName, A.lastName, C.companyName, J.jobTitle, COALESCE(AP.applicationDate, 'Not Applied') AS applicationDate
  FROM Applicants A
     CROSS JOIN Companies C
     CROSS JOIN Jobs J
    LEFT JOIN Applications AP ON A.applicantID = AP.applicantID AND J.jobID = AP.jobID;
1271
1272
1273
1274
```

The results grid displays the following data:

applicantID	firstName	lastName	companyName	jobTitle	applicationDate
1	John	Doe	ABC Corporation	Data Scientist	Not Applied
1	John	Doe	ABC Corporation	Financial Analyst	Not Applied
1	John	Doe	ABC Corporation	Marketing Specialist	Not Applied
1	John	Doe	ABC Corporation	Network Administrator	Not Applied
1	John	Doe	ABC Corporation	Product Manager	Not Applied
1	John	Doe	ABC Corporation	Project Manager	Not Applied
1	John	Doe	ABC Corporation	Research Scientist	Not Applied
1	John	Doe	ABC Corporation	Sales Representative	Not Applied

The output pane shows the execution log:

- 7 17:44:08 SELECT A.applicantID, A.firstName, A.lastName, A.email FROM Applicants A JOIN Applications AP ON A.applicantID = AP.applicantID WHERE AP.jobID = 10; 0 row(s) returned
- 8 17:45:54 SELECT A.applicantID, A.firstName, A.lastName, C.companyName, J.jobTitle FROM Applicants A JOIN Companies C ON A.companyID = C.companyID JOIN Jobs J ON A.applicantID = J.applicantID WHERE J.jobID = 10; 10 row(s) returned
- 9 17:47:41 SELECT C.companyID, C.companyName, COUNT(J.jobID) AS jobCount FROM Companies C LEFT JOIN Jobs J ON C.companyID = J.companyID GROUP BY C.companyID; 10 row(s) returned
- 10 17:49:36 SELECT A.applicantID, A.firstName, A.lastName, C.companyName, J.jobTitle, COALESCE(AP.applicationDate, 'Not Applied') AS applicationDate FROM Applicants A CROSS JOIN Companies C CROSS JOIN Jobs J LEFT JOIN Applications AP ON A.applicantID = AP.applicantID AND J.jobID = AP.jobID; 1000 row(s) returned

Q16. Find companies that have posted jobs with a salary higher than the average salary of all jobs.

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query is:

```
1274
1275 • SELECT DISTINCT C.companyName
  FROM Companies C
     JOIN Jobs J ON C.companyID = J.companyID
 WHERE J.salary > (SELECT AVG(salary) FROM Jobs);
1277
1278
1279
1280
1281
1282 • show databases;
1283 • use TechShop;
1284 • show tables;
```

The results grid displays the following data:

CompanyName
XYZ Tech
TechInnovators
Mega Corp
Digital Creations
Epic Innovations

The output pane shows the execution log:

- 1 17:52:15 SELECT DISTINCT C.companyName FROM Companies C JOIN Jobs J ON C.companyID = J.companyID WHERE J.salary > (SELECT AVG(salary) FROM Jobs); 5 row(s) returned

Q18. Retrieve a list of jobs with titles containing either 'Developer' or 'Engineer'.

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query is:

```

829 WHERE salary BETWEEN 60000 AND 80000;
830
831 • SELECT *
832   FROM Jobs
833 WHERE jobID NOT IN (SELECT DISTINCT jobID FROM Applications);
834
835 • SELECT *
  
```

The results grid shows one row of data:

jobID	companyId	jobTitle	jobDescription	joblocation	salary	jobType	postedDate
1	NULL	Software Engineer	Develop and maintain software applications	New York	90000	Full-time	2023-01-10

The output pane shows the executed SQL query and the result message:

```

1 00:24:28 SELECT * FROM Jobs WHERE jobTitle LIKE '%Developer%' OR jobTitle LIKE '%Engineer%' LIMIT 0, 1000
Message
1 row(s) returned
Duration / Fetch
0.000 sec / 0.000 sec
  
```

Q19. Retrieve a list of applicants and the jobs they have applied for, including those who have not applied and jobs without applicants.

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query is:

```

1280
1281 • SELECT A.applicantID,A.firstName,A.lastName,J.jobID,J.jobTitle,C.companyName,J.jobLocation,J.salary,Apl.applicationDate,Apl.coverLetter
FROM Applicants A
CROSS JOIN Jobs J
JOIN Companies C ON J.companyID = C.companyID
LEFT JOIN Applications Apl ON A.applicantID = Apl.applicantID AND J.jobID = Apl.jobID
ORDER BY A.applicantID, J.jobID;
1282
1283
1284
1285
1286
1287
1288
1289 • show databases;
1290 • use TechShop;
  
```

The results grid shows multiple rows of data:

applicantID	firstName	lastName	jobID	jobTitle	companyName	jobLocation	salary	applicationDate	coverLetter
1	John	Doe	1	Software Engineer	ABC Corporation	New York	90000	2023-01-05	I am excited about the opportunity to work as a...
1	John	Doe	2	Data Scientist	XYZ Tech	San Francisco	110000	NULL	NULL
1	John	Doe	3	Product Manager	Tech Innovators	Seattle	120000	NULL	NULL
1	John	Doe	4	Sales Representative	Global Solutions	London	80000	NULL	NULL
1	John	Doe	5	UX/UI Designer	Innovate Ltd	Tokyo	95000	NULL	NULL
1	John	Doe	6	Marketing Specialist	Future Enterprises	Berlin	85000	NULL	NULL
1	John	Doe	7	Network Administrator	Smart Systems	Sydney	100000	NULL	NULL
1	John	Doe	8	Financial Analyst	Mega Corp	Toronto	110000	NULL	NULL

The output pane shows the executed SQL query and the result message:

```

1 17:52:15 SELECT DISTINCT C.companyName FROM Companies C JOIN Jobs J ON C.companyID = J.companyID WHERE J.salary > 100000
5 row(s) returned
2 17:53:31 SELECT A.applicantID, A.firstName, A.lastName, J.jobID, J.jobTitle, C.companyName, J.jobLocation, ...
100 row(s) returned
Message
Duration / Fetch
0.000 sec / 0.000 sec
0.032 sec / 0.000 sec
  
```