

JobNotificationSystem - Sample Result Queries and Examples

1. Active Users and Their Preferences

This query retrieves all active users along with their job preferences including location, job type, and preferred role.

```
SELECT u.FirstName, u.LastName, u.Email, up.PreferredLocation,  
up.JobType, up.UserPreferredRole  
FROM Users u  
JOIN UserPreferences up ON u.UserID = up.UserID  
WHERE u.UserStatus = 'Active';
```

The screenshot displays the MySQL Workbench interface. The central query editor contains the following SQL query:

```
-- Active Users and Their Preferences--  
SELECT u.FirstName, u.LastName, u.Email, up.PreferredLocation, up.JobType, up.UserPreferredRole  
FROM Users u  
JOIN UserPreferences up ON u.UserID = up.UserID  
WHERE u.UserStatus = 'Active';
```

The results are shown in a table with the following columns: FirstName, LastName, Email, PreferredLocation, JobType, and UserPreferredRole. The table contains 20 rows of data.

FirstName	LastName	Email	PreferredLocation	JobType	UserPreferredRole
Susan	Taylor	susan.taylor1@example.com	Austin	Part Time	Machine Learning Engineer
William	Moore	william.moore1@example.com	Boston	Internship	Front End Developer
John	Doe	john.doe2@example.com	New York	Full Time	Full Stack Developer
Jane	Smith	jane.smith2@example.com	Los Angeles	Part Time	Machine Learning Engineer
Michael	Jones	michael.jones2@example.com	Chicago	Internship	Back End Developer
Lisa	Brown	lisa.brown2@example.com	Houston	Full Time	Data Scientist
Lisa	Brown	lisa.brown2@example.com	Houston	Part Time	Data Analyst
Emily	Davis	emily.davis2@example.com	Phoenix	Internship	Mobile Developer
Chris	Wilson	chris.wilson2@example.com	Seattle	Full Time	Full Stack Developer
Chris	Wilson	chris.wilson2@example.com	Seattle	Part Time	DevOps Engineer
Maria	Garcia	maria.garcia2@example.com	San Francisco	Internship	Data Analyst
David	Miller	david.miller2@example.com	Denver	Part Time	DevOps Engineer
Susan	Taylor	susan.taylor2@example.com	Austin	Full Time	AI Specialist
Susan	Taylor	susan.taylor2@example.com	Austin	Part Time	Data Engineer
William	Moore	william.moore2@example.com	Boston	Internship	Front End Developer

The bottom panel shows the output log with the following entries:

#	Time	Action	Message	Duration / Fetch
1	07:10:07	SELECT u.FirstName, u.LastName, u.Email, up.PreferredLocation, up.JobType, up.UserPreferredRole FROM Users u...	Error Code: 1146. Table 'sys.users' doesn't exist	0.000 sec
2	07:10:33	show databases	6 row(s) returned	0.000 sec / 0.000 sec
3	07:10:59	use jobnotificationssystem	0 row(s) affected	0.000 sec
4	07:11:05	SELECT u.FirstName, u.LastName, u.Email, up.PreferredLocation, up.JobType, up.UserPreferredRole FROM Users u...	20 row(s) returned	0.000 sec / 0.000 sec

2. Jobs Matching User Skills (UserID = 6)

Lists job titles that match the skills of User 6, along with company names and skill names.

```
SELECT DISTINCT j.Jobtitle, c.CompanyName, j.Location, s.SkillName  
FROM Jobs j  
JOIN JobSkills js ON j.JobID = js.JobID  
JOIN Skills s ON js.SkillID = s.SkillID  
JOIN UserSkills us ON s.SkillID = us.SkillID  
JOIN Companies c ON j.CompanyID = c.CompanyID  
WHERE us.UserID = 6
```

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

```
1  
2 SELECT DISTINCT j.Jobtitle, c.CompanyName, j.Location, s.SkillName  
3 FROM Jobs j  
4 JOIN JobSkills js ON j.JobID = js.JobID  
5 JOIN Skills s ON js.SkillID = s.SkillID  
6 JOIN UserSkills us ON s.SkillID = us.SkillID  
7 JOIN Companies c ON j.CompanyID = c.CompanyID  
8 WHERE us.UserID = 6;  
9
```

The results are displayed in a table with the following columns: Jobtitle, CompanyName, Location, and SkillName. The table contains 12 rows of data.

Jobtitle	CompanyName	Location	SkillName
Full Stack Developer	FullStack Corp	Los Angeles	AWS
Data Engineer	DataArch Architects	Boston	AWS
AI Specialist	AI Pioneers	Houston	AWS
Data Engineer	DataArch Architects	Portland	AWS
Big Data Specialist	Portland Data Builders	Salt Lake City	AWS
AI Specialist	AI Pioneers	San Jose	AWS
Mobile Developer	Mobile Masters	Seattle	GCP
Data Engineer	DataArch Architects	Portland	GCP
Mobile Developer	Mobile Masters	Charlotte	GCP
AI Specialist	AI Pioneers	San Jose	GCP
DevOps Engineer	DevOps Dynamics	Denver	Ansible
DevOps Engineer	DevOps Dynamics	Atlanta	Ansible

The bottom panel shows the output of the query, including the time taken to execute the query and the number of rows returned.

#	Time	Action	Message	Duration / Fetch
2	07:10:33	show databases	6 row(s) returned	0.000 sec / 0.000 sec
3	07:10:59	use jobnotificationsystem	0 row(s) affected	0.000 sec
4	07:11:05	SELECT u.FirstName, u.LastName, u.Email, u.PreferredLocation, u.JobType, u.UserPreferredRole FROM ...	20 row(s) returned	0.000 sec / 0.000 sec
5	07:14:46	SELECT DISTINCT j.Jobtitle, c.CompanyName, j.Location, s.SkillName FROM Jobs j JOIN JobSkills js ON j.JobID = js.JobID JOIN Skills s ON js.SkillID = s.SkillID JOIN UserSkills us ON s.SkillID = us.SkillID JOIN Companies c ON j.CompanyID = c.CompanyID WHERE us.UserID = 6;	12 row(s) returned	0.000 sec / 0.000 sec

3. Notifications Sent to Users for Matching Jobs

Displays all notifications sent to users about job matches, including job title and message content.

```
SELECT u.Email, j.Jobtitle, n.Message, n.Notification_sent_at  
FROM Notifications n  
JOIN Users u ON n.UserID = u.UserID  
JOIN Jobs j ON n.JobID = j.JobID;
```

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

```
1 SELECT u.Email, j.Jobtitle, n.Message, n.Notification_sent_at
2 FROM Notifications n
3 JOIN Users u ON n.UserID = u.UserID
4 JOIN Jobs j ON n.JobID = j.JobID;
```

The query results are displayed in the Result Grid, showing a list of notifications sent to users. The columns are Email, Jobtitle, Message, and Notification_sent_at. The results are as follows:

Email	Jobtitle	Message	Notification_sent_at
chris.wilson1@example.com	Front End Developer	New job opening for Front End Developer in Ne...	2024-01-11 08:00:00
maria.garcia1@example.com	Data Analyst	New job opening for Data Analyst in San Franci...	2024-01-15 13:00:00
david.miller1@example.com	DevOps Engineer	Exciting opportunity for DevOps Engineer in De...	2024-01-16 14:00:00
susan.taylor1@example.com	Machine Learning Engineer	Machine Learning Engineer role open in Austin...	2024-01-17 15:00:00
william.moore1@example.com	Front End Developer	Front End Developer position available in San Di...	2024-01-20 17:00:00
john.doe2@example.com	Game Developer	Exciting role for Game Developer in Las Vegas...	2024-01-28 09:00:00
jane.smith2@example.com	Data Engineer	New job for Data Engineer in Portland. Don't mi...	2024-01-22 09:30:00
Michael.Jones2@example.com	Big Data Specialist	Big Data Specialist position available in Salt Lake...	2024-01-29 11:30:00
isa.brown2@example.com	Full Stack Developer	Full Stack Developer needed in Phoenix. Apply t...	2024-01-23 10:00:00
emily.davis2@example.com	Mobile Developer	Mobile Developer role open in Charlotte. Check L...	2024-01-24 11:00:00
chris.wilson1@example.com	Full Stack Developer	Full Stack Developer position open in Los Angele...	2024-01-12 12:00:00
maria.garcia1@example.com	Data Engineer	Data Engineer opportunity in Boston. Apply now!	2024-01-16 15:00:00
david.miller1@example.com	DevOps Engineer	DevOps Engineer needed in Atlanta. Don't miss ...	2024-01-25 13:30:00
susan.taylor1@example.com	Machine Learning Engineer	Machine Learning Engineer opening in Philadelp...	2024-01-26 14:30:00
william.moore1@example.com	AI Specialist	AI Specialist role available in San Jose. Apply to...	2024-01-27 15:30:00

The Output pane shows the execution of the query, indicating that 30 rows were returned.

4. Applications Submitted with Status

Shows which users applied to which jobs, along with their application status and timestamps.

```
SELECT u.FirstName, u.LastName, j.Jobtitle, a.Applicationstatus,  
a.ApplicationappliedAT  
FROM Applications a  
JOIN Users u ON a.UserID = u.UserID  
JOIN Jobs j ON a.JobID = j.JobID;
```

The screenshot displays the MySQL Workbench interface. The central query editor contains the following SQL query:

```
1 SELECT u.FirstName, u.LastName, j.Jobtitle, a.Applicationstatus, a.ApplicationappliedAT
2 FROM Applications a
3 JOIN Users u ON a.UserID = u.UserID
4 JOIN Jobs j ON a.JobID = j.JobID;
```

The 'Result Grid' tab is active, showing the following data:

FirstName	LastName	Jobtitle	Applicationstatus	ApplicationappliedAT
Chris	Wilson	Mobile Developer	Pending	2023-12-15 15:00:00
Chris	Wilson	DevOps Engineer	Accepted	2023-12-18 15:30:00
Maria	Garcia	Data Analyst	Pending	2024-01-16 10:00:00
Maria	Garcia	Data Engineer	Rejected	2024-01-18 11:00:00
David	Miller	Machine Learning Engineer	Accepted	2024-02-01 17:00:00
Susan	Taylor	AI Specialist	Pending	2024-02-15 18:00:00
William	Moore	Front End Developer	Pending	2024-03-01 19:00:00
William	Moore	Back End Developer	Rejected	2024-03-05 19:30:00
John	Doe	Game Developer	Accepted	2024-03-25 18:30:00
Jane	Smith	Data Analyst	Pending	2024-04-01 11:30:00
Michael	Jones	Data Engineer	Accepted	2024-05-01 12:30:00
Lisa	Brown	Big Data Specialist	Rejected	2024-05-15 13:30:00
Emily	Davis	Full Stack Developer	Pending	2024-06-01 14:30:00
Chris	Wilson	Mobile Developer	Pending	2024-07-01 15:30:00
Chris	Wilson	DevOps Engineer	Accepted	2024-07-15 16:30:00

The 'Output' tab at the bottom shows the execution log:

#	Time	Action	Message	Duration / Fetch
4	07:11:05	SELECT u.FirstName, u.LastName, u.Email, u.PreferredLocation, u.JobType, u.UserPreferredRole FROM ...	20 row(s) returned	0.000 sec / 0.000 sec
5	07:14:46	SELECT DISTINCT j.Jobtitle, c.CompanyName, j.Location, s.SkillName FROM Jobs j JOIN JobSkills js ON j.Job...	12 row(s) returned	0.000 sec / 0.000 sec
6	07:17:13	SELECT u.Email, j.Jobtitle, n.Message, n.Notification_sent_at FROM Notifications n JOIN Users u ON n.User...	30 row(s) returned	0.000 sec / 0.000 sec
7	07:18:28	SELECT u.FirstName, u.LastName, j.Jobtitle, a.Applicationstatus, a.ApplicationappliedAT FROM Applications a ...	20 row(s) returned	0.016 sec / 0.000 sec

5. Suggest Skills for Job Matching (UserID = 20)

Recommends skills to User 20 that they do not currently have, but are required in job postings.

```
SELECT DISTINCT s.SkillName, s.SkillsourceURL  
FROM JobSkills js  
JOIN Skills s ON js.SkillID = s.SkillID  
WHERE js.SkillID NOT IN (  
    SELECT us.SkillID  
    FROM UserSkills us  
    WHERE us.UserID = 20  
);
```

The screenshot displays the MySQL Workbench interface. The central query editor contains the following SQL query:

```
1 SELECT DISTINCT s.SkillName, s.SkillsourceURL  
2 FROM JobSkills js  
3 JOIN Skills s ON js.SkillID = s.SkillID  
4 WHERE js.SkillID NOT IN (  
5     SELECT us.SkillID  
6     FROM UserSkills us  
7     WHERE us.UserID = 20  
8 );  
9  
10
```

The left sidebar shows the 'Navigator' pane with sections for 'MANAGEMENT' (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), 'INSTANCE' (Startup / Shutdown, Server Logs, Options File), and 'PERFORMANCE' (Dashboard, Performance Reports, Performance Schema Setup). The 'Administration' section is selected, showing 'Schemas' and 'Information'. A message states 'No object selected'.

The 'Result Grid' pane at the bottom displays the query results in a table with two columns: 'SkillName' and 'SkillsourceURL'. The results are as follows:

SkillName	SkillsourceURL
HTML	https://www.coursera.org/learn/html
CSS	https://www.udemy.com/course/css-for-beginners/
JavaScript	https://www.pluralsight.com/courses/javascript
Python	https://www.coursera.org/learn/python
Java	https://www.udemy.com/course/java-programming/
SQL	https://www.datacamp.com/courses/sql
AWS	https://aws.amazon.com/training/
GCP	https://cloud.google.com/training/
Kubernetes	https://www.edx.org/course/kubernetes
Docker	https://www.pluralsight.com/courses/docker
Pandas	https://www.coursera.org/learn/pandas-data-analysis
NumPy	https://www.udemy.com/course/numpy
TensorFlow	https://www.tensorflow.org/learn
PyTorch	https://www.udacity.com/course/pytorch
Tableau	https://www.tableau.com/learn/training

The 'Output' pane at the bottom shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
5	07:14:46	SELECT DISTINCT j.JobTitle, c.CompanyName, j.Location, s.SkillName FROM Jobs j JOIN JobSkills js ON j.JobID = js.JobID	12 row(s) returned	0.000 sec / 0.000 sec
6	07:17:13	SELECT u.Email, j.JobTitle, n.Message, n.Notification_sent_at FROM Notifications n JOIN Users u ON n.UserID = u.UserID	30 row(s) returned	0.000 sec / 0.000 sec
7	07:18:28	SELECT u.FirstName, u.LastName, j.JobTitle, a.ApplicationStatus, a.ApplicationAppliedAt FROM Applications a JOIN Users u ON a.UserID = u.UserID	20 row(s) returned	0.016 sec / 0.000 sec
8	07:19:10	SELECT DISTINCT s.SkillName, s.SkillsourceURL FROM JobSkills js JOIN Skills s ON js.SkillID = s.SkillID WHERE js.SkillID NOT IN (SELECT us.SkillID FROM UserSkills us WHERE us.UserID = 20)	19 row(s) returned	0.016 sec / 0.000 sec