

LinkedIn Database

Create a new database with your RegNo. Design the schema for the LinkedIn database, considering entities such as users, profiles, connections, job postings, and any other relevant entities you deem necessary. Create the necessary tables for each entity, ensuring appropriate data types, primary keys, and foreign keys. Implement the necessary integrity constraints, such as unique constraints, not-null constraints, and any other constraints deemed necessary for data consistency.

```
mysql> use sanjiv;
```

Database changed

```
mysql> CREATE TABLE 21BCI0045_Users(
```

```
    -> UserID INT PRIMARY KEY,
```

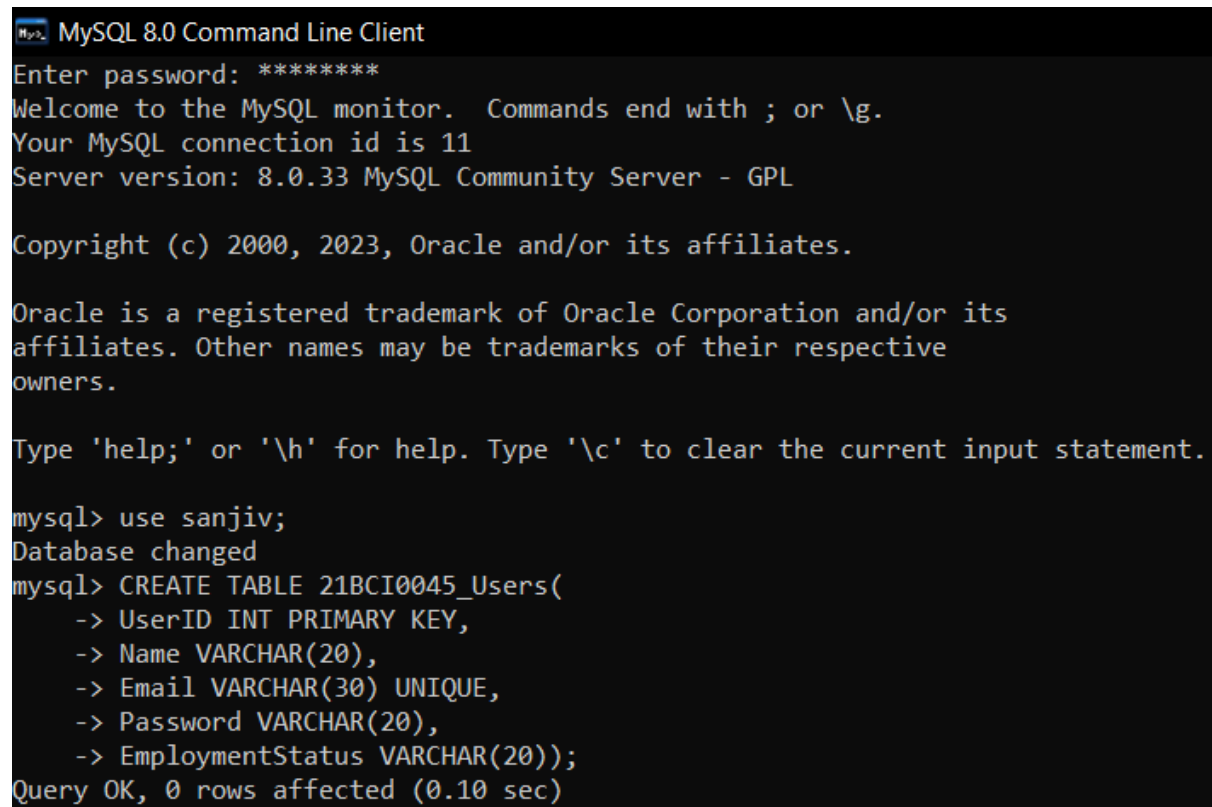
```
    -> Name VARCHAR(20),
```

```
    -> Email VARCHAR(30) UNIQUE,
```

```
    -> Password VARCHAR(20),
```

```
    -> EmploymentStatus VARCHAR(20));
```

Query OK, 0 rows affected (0.10 sec)



```
MySQL 8.0 Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.33 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use sanjiv;
Database changed
mysql> CREATE TABLE 21BCI0045_Users(
    -> UserID INT PRIMARY KEY,
    -> Name VARCHAR(20),
    -> Email VARCHAR(30) UNIQUE,
    -> Password VARCHAR(20),
    -> EmploymentStatus VARCHAR(20));
Query OK, 0 rows affected (0.10 sec)
```

```
mysql> CREATE TABLE 21BCI0045_Profiles(
```

```
-> ProfileID INT PRIMARY KEY,
```

```
-> UserID INT,
```

```
-> Title VARCHAR(20),
```

```
-> Summary TEXT,
```

```
-> Experience TEXT,
```

```
-> Education TEXT,
```

```
-> FOREIGN KEY(UserID) REFERENCES 21BCI0045_Users(UserID));
```

Query OK, 0 rows affected (0.03 sec)

```
mysql> CREATE TABLE 21BCI0045_Profiles(  
  -> ProfileID INT PRIMARY KEY,  
  -> UserID INT,  
  -> Title VARCHAR(20),  
  -> Summary TEXT,  
  -> Experience TEXT,  
  -> Education TEXT,  
  -> FOREIGN KEY(UserID) REFERENCES 21BCI0045_Users(UserID));  
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> CREATE TABLE 21BCI0045_Connections(
```

```
-> ConnectionID INT PRIMARY KEY,
```

```
-> UserID1 INT,
```

```
-> UserID2 INT,
```

```
-> Status VARCHAR(20),
```

```
-> FOREIGN KEY(UserID1) REFERENCES 21BCI0045_Users(UserID),
```

```
-> FOREIGN KEY(UserID2) REFERENCES 21BCI0045_Users(UserID));
```

Query OK, 0 rows affected (0.03 sec)

```
mysql> CREATE TABLE 21BCI0045_Connections(  
  -> ConnectionID INT PRIMARY KEY,  
  -> UserID1 INT,  
  -> UserID2 INT,  
  -> Status VARCHAR(20),  
  -> FOREIGN KEY(UserID1) REFERENCES 21BCI0045_Users(UserID),  
  -> FOREIGN KEY(UserID2) REFERENCES 21BCI0045_Users(UserID));  
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> CREATE TABLE 21BCI0045_Jobpostings(
    -> JobpostingID INT PRIMARY KEY,
    -> UserID INT,
    -> Title VARCHAR(30),
    -> Description TEXT,
    -> Location VARCHAR(20),
    -> Industry VARCHAR(20),
    -> FOREIGN KEY(UserID) REFERENCES 21BCI0045_Users(UserID));
```

Query OK, 0 rows affected (0.02 sec)

```
mysql> CREATE TABLE 21BCI0045_Jobpostings(
    -> JobpostingID INT PRIMARY KEY,
    -> UserID INT,
    -> Title VARCHAR(30),
    -> Description TEXT,
    -> Location VARCHAR(20),
    -> Industry VARCHAR(20),
    -> FOREIGN KEY(UserID) REFERENCES 21BCI0045_Users(UserID));
Query OK, 0 rows affected (0.02 sec)
```

Populate the tables with sample data to demonstrate the functionality of the LinkedIn database.

```
mysql> INSERT INTO 21BCI0045_Users(UserID,Name,Email>Password,EmploymentStatus) VALUES
    -> (1,'Ram Babu','ram.babu@yahoo.com','pass123','Employed'),
    -> (2,'Vijay Shankar','vijay.shankar@yahoo.com','pass456','Employed'),
    -> (3,'Shyam Singh','shyam.singh@yahoo.com','pass789','Unemployed'),
    -> (4,'Rahul Sharma','rahul.sharma@yahoo.com','pass890','Employed'),
    -> (5,'Somesh Kapoor','somesh.kapoor@yahoo.com','pass012','Employed');
```

Query OK, 5 rows affected (0.01 sec)

Records: 5 Duplicates: 0 Warnings: 0

```
mysql> INSERT INTO 21BCI0045_Users(UserID,Name,Email>Password,EmploymentStatus) VALUES
    -> (1,'Ram Babu','ram.babu@yahoo.com','pass123','Employed'),
    -> (2,'Vijay Shankar','vijay.shankar@yahoo.com','pass456','Employed'),
    -> (3,'Shyam Singh','shyam.singh@yahoo.com','pass789','Unemployed'),
    -> (4,'Rahul Sharma','rahul.sharma@yahoo.com','pass890','Employed'),
    -> (5,'Somesh Kapoor','somesh.kapoor@yahoo.com','pass012','Employed');
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

```
mysql> INSERT INTO 21BCI0045_Profiles(ProfileID,UserID,Title,Summary,Experience,Education)
VALUES
```

-> (1,1,'Data Scientist','Specialized in analyzing large datasets','3+ years','Bachelor of Science in Statistics'),

-> (2,2,'HR Manager','Specialized in managing people','9+ years','Master of Business Administration'),

-> (3,3,'Graphic Designer','Specialized in using tools like canva, flutter','no experience','Bachelor of Science in Computer Science'),

-> (4,4,'Software Engineer','Specialized in web development','4+ years','Bachelor of Science in Computer Science'),

-> (5,5,'Marketing Manager','Speialized in digital marketing strategies','6+ years','Bachelor of Business Administration');

Query OK, 5 rows affected (0.01 sec)

Records: 5 Duplicates: 0 Warnings: 0

```
mysql> INSERT INTO 21BCI0045_Profiles(ProfileID,UserID,Title,Summary,Experience,Education) VALUES
-> (1,1,'Data Scientist','Specialized in analyzing large datasets','3+ years','Bachelor of Science in Statistics'),
-> (2,2,'HR Manager','Specialized in managing people','9+ years','Master of Business Administration'),
-> (3,3,'Graphic Designer','Specialized in using tools like canva, flutter','no experience','Bachelor of Science in Computer Science'),
-> (4,4,'Software Engineer','Specialized in web development','4+ years','Bachelor of Science in Computer Science'),
-> (5,5,'Marketing Manager','Speialized in digital marketing strategies','6+ years','Bachelor of Business Administration');
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

mysql> INSERT INTO 21BCI0045_Connections(ConnectionID,UserID1,UserID2,Status) VALUES

-> (1,1,2,'Connected'),

-> (2,1,3,'Connected'),

-> (3,2,4,'Connected'),

-> (4,3,5,'Connected'),

-> (5,4,5,'Connected');

Query OK, 5 rows affected (0.01 sec)

Records: 5 Duplicates: 0 Warnings: 0

```
mysql> INSERT INTO 21BCI0045_Connections(ConnectionID,UserID1,UserID2,Status) VALUES
-> (1,1,2,'Connected'),
-> (2,1,3,'Connected'),
-> (3,2,4,'Connected'),
-> (4,3,5,'Connected'),
-> (5,4,5,'Connected');
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

mysql> INSERT INTO 21BCI0045_Jobpostings(JobpostingID,UserID,Title,Description,Location,Industry) VALUES

-> (1,1,'Data Scientist','Join our team and derive valuable insights from data','Bangalore','Analytics'),

-> (2,2,'HR Manager','Lead our team and ensure tasks are completed','Mumbai','Management'),

-> (3,3,'Graphic Designer Intern','Opportunity to learn and grow','Kolkata','Design'),

-> (4,4,'Senior Software Engineer','Join our team and gain valuable experience','Chennai','Technology'),

-> (5,5,'Digital Marketing Specialist','Drive digital marketing initiatives','Delhi','Marketing');

Query OK, 5 rows affected (0.00 sec)

Records: 5 Duplicates: 0 Warnings: 0

```
mysql> INSERT INTO 21BCI0045_Jobpostings(JobpostingID,UserID,Title,Description,Location,Industry) VALUES
-> (1,1,'Data Scientist','Join our team and derive valuable insights from data','Bangalore','Analytics'),
-> (2,2,'HR Manager','Lead our team and ensure tasks are completed','Mumbai','Management'),
-> (3,3,'Graphic Designer Intern','Opportunity to learn and grow','Kolkata','Design'),
-> (4,4,'Senior Software Engineer','Join our team and gain valuable experience','Chennai','Technology'),
-> (5,5,'Digital Marketing Specialist','Drive digital marketing initiatives','Delhi','Marketing');
Query OK, 5 rows affected (0.00 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

Write SQL queries to perform the following tasks:

Retrieve a user's profile information, including their connections and job history.

```
mysql> SELECT
21BCI0045_Users.UserID,21BCI0045_Users.Name,21BCI0045_Profiles.Title,21BCI0045_Profiles.Experience,21BCI0045_Connections.Status,21BCI0045_Jobpostings.Title AS JobHistory
```

-> FROM 21BCI0045_Users

-> LEFT JOIN 21BCI0045_Profiles ON 21BCI0045_Users.UserID = 21BCI0045_Profiles.UserID

-> LEFT JOIN 21BCI0045_Connections ON 21BCI0045_Users.UserID = 21BCI0045_Connections.UserID1 OR 21BCI0045_Users.UserID = 21BCI0045_Connections.UserID2

-> LEFT JOIN 21BCI0045_Jobpostings ON 21BCI0045_Users.UserID = 21BCI0045_Jobpostings.UserID

-> WHERE 21BCI0045_Users.UserID = 1;

```
mysql> SELECT 21BCI0045_Users.UserID,21BCI0045_Users.Name,21BCI0045_Profiles.Title,21BCI0045_Profiles.Experience,21BCI0045_Connections.Status,21BCI0045_Jobpostings.Title AS JobHistory
-> FROM 21BCI0045_Users
-> LEFT JOIN 21BCI0045_Profiles ON 21BCI0045_Users.UserID = 21BCI0045_Profiles.UserID
-> LEFT JOIN 21BCI0045_Connections ON 21BCI0045_Users.UserID = 21BCI0045_Connections.UserID1 OR 21BCI0045_Users.UserID = 21BCI0045_Connections.UserID2
-> LEFT JOIN 21BCI0045_Jobpostings ON 21BCI0045_Users.UserID = 21BCI0045_Jobpostings.UserID
-> WHERE 21BCI0045_Users.UserID = 1;
+----+-----+-----+-----+-----+-----+
| UserID | Name   | Title           | Experience | Status | JobHistory |
+----+-----+-----+-----+-----+-----+
| 1      | Ram Babu | Data Scientist | 3+ years  | Connected | Data Scientist |
| 1      | Ram Babu | Data Scientist | 3+ years  | Connected | Data Scientist |
+----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

Search for job postings based on specific criteria (e.g., job title, location, industry).

```
mysql> SELECT *
```

-> FROM 21BCI0045_Jobpostings

-> WHERE Title LIKE '%Senior Software Engineer%' AND Location = 'Chennai' AND Industry = 'Technology';

```
mysql> SELECT *
-> FROM 21BCI0045_Jobpostings
-> WHERE Title LIKE '%Senior Software Engineer%' AND Location = 'Chennai' AND Industry = 'Technology';
+----+-----+-----+-----+-----+-----+
| JobpostingID | UserID | Title           | Description | Location | Industry |
+----+-----+-----+-----+-----+-----+
| 4            | 4      | Senior Software Engineer | Join our team and gain valuable experience | Chennai | Technology |
+----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Insert new user profiles, job postings, and connection requests into the respective tables.

```
mysql> INSERT INTO 21BCI0045_Users(UserID,Name,Email>Password,EmploymentStatus) VALUES
```

```
-> (6,'Ashok Mahendra','ashok.mahendra@yahoo.com','pass901','Employed');
```

Query OK, 1 row affected (0.00 sec)

```
mysql> INSERT INTO 21BCI0045_Profiles(ProfileID,UserID,Title,Summary,Experience,Education) VALUES
```

```
-> (6,6,'Security Analyst','Specialized in information security','7+ years','Bachelor of Science in Computer Science');
```

Query OK, 1 row affected (0.00 sec)

```
mysql> INSERT INTO 21BCI0045_Jobpostings(JobpostingID,UserID,Title,Description,Location,Industry) VALUES
```

```
-> (6,6,'Security Analyst','Participate in analysis and audits','Hyderabad','Security');
```

Query OK, 1 row affected (0.00 sec)

```
mysql> INSERT INTO 21BCI0045_Connections(ConnectionID,UserID1,UserID2>Status) VALUES
```

```
-> (6,3,6,'Pending');
```

Query OK, 1 row affected (0.00 sec)

```
mysql> INSERT INTO 21BCI0045_Users(UserID,Name,Email>Password,EmploymentStatus) VALUES
```

```
-> (6,'Ashok Mahendra','ashok.mahendra@yahoo.com','pass901','Employed');
```

Query OK, 1 row affected (0.00 sec)

```
mysql> INSERT INTO 21BCI0045_Profiles(ProfileID,UserID,Title,Summary,Experience,Education) VALUES
```

```
-> (6,6,'Security Analyst','Specialized in information security','7+ years','Bachelor of Science in Computer Science');
```

Query OK, 1 row affected (0.00 sec)

```
mysql> INSERT INTO 21BCI0045_Jobpostings(JobpostingID,UserID,Title,Description,Location,Industry) VALUES
```

```
-> (6,6,'Security Analyst','Participate in analysis and audits','Hyderabad','Security');
```

Query OK, 1 row affected (0.00 sec)

```
mysql> INSERT INTO 21BCI0045_Connections(ConnectionID,UserID1,UserID2>Status) VALUES
```

```
-> (6,3,6,'Pending');
```

Query OK, 1 row affected (0.00 sec)

Update user profile information and job postings.

```
mysql> UPDATE 21BCI0045_Profiles SET Title = 'Manager' WHERE UserID = 4;
```

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> UPDATE 21BCI0045_Jobpostings SET Description = 'Join our growing team and make an impact' WHERE JobpostingID = 4;
```

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> UPDATE 21BCI0045_Profiles SET Title = 'Manager' WHERE UserID = 4;
```

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> UPDATE 21BCI0045_Jobpostings SET Description = 'Join our growing team and make an impact' WHERE JobpostingID = 4;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

Delete user profiles and job postings.

```
mysql> DELETE FROM 21BCI0045_Profiles WHERE UserID = 6;
```

Query OK, 1 row affected (0.00 sec)

```
mysql> DELETE FROM 21BCI0045_Jobpostings WHERE JobpostingID = 6;
```

Query OK, 1 row affected (0.00 sec)

```
mysql> DELETE FROM 21BCI0045_Profiles WHERE UserID = 6;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> DELETE FROM 21BCI0045_Jobpostings WHERE JobpostingID = 6;
Query OK, 1 row affected (0.00 sec)
```

Retrieve users who have the most common connections with a specific user

```
mysql> SELECT 21BCI0045_Users.UserID, 21BCI0045_Users.Name, COUNT(*) AS
CommonConnections
```

-> FROM 21BCI0045_Users

-> JOIN 21BCI0045_Connections ON 21BCI0045_Users.UserID = 21BCI0045_Connections.UserID1
OR 21BCI0045_Users.UserID = 21BCI0045_Connections.UserID2

-> WHERE 1 IN (21BCI0045_Connections.UserID1, 21BCI0045_Connections.UserID2) AND
21BCI0045_Users.UserID != 1

-> GROUP BY 21BCI0045_Users.UserID, 21BCI0045_Users.Name

-> ORDER BY CommonConnections DESC

-> LIMIT 10;

```
mysql> SELECT 21BCI0045_Users.UserID, 21BCI0045_Users.Name, COUNT(*) AS CommonConnections
-> FROM 21BCI0045_Users
-> JOIN 21BCI0045_Connections ON 21BCI0045_Users.UserID = 21BCI0045_Connections.UserID1 OR 21BCI0045_Users.UserID = 21BCI0045_Connections.UserID2
-> WHERE 1 IN (21BCI0045_Connections.UserID1, 21BCI0045_Connections.UserID2) AND 21BCI0045_Users.UserID != 1
-> GROUP BY 21BCI0045_Users.UserID, 21BCI0045_Users.Name
-> ORDER BY CommonConnections DESC
-> LIMIT 10;
+-----+-----+-----+
| UserID | Name       | CommonConnections |
+-----+-----+-----+
| 2      | Vijay Shankar | 1                |
| 3      | Shyam Singh  | 1                |
+-----+-----+-----+
2 rows in set (0.02 sec)
```

Retrieve users who possess a unique combination of skills

```
mysql> ALTER TABLE 21BCI0045_Profiles
```

-> ADD COLUMN Skills VARCHAR(20);

Query OK, 0 rows affected (0.05 sec)

Records: 0 Duplicates: 0 Warnings: 0


```
mysql> UPDATE 21BCI0045_Profiles
```

```
-> SET Skills='Java'
```

```
-> WHERE UserID=1;
```

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> UPDATE 21BCI0045_Profiles
```

```
-> SET Skills='Python'
```

```
-> WHERE UserID=2;
```

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> SELECT 21BCI0045_Users.UserID, 21BCI0045_Users.Name
```

```
-> FROM 21BCI0045_Users
```

```
-> JOIN 21BCI0045_Profiles ON 21BCI0045_Users.UserID = 21BCI0045_Profiles.UserID
```

```
-> WHERE Skills IN (
```

```
-> SELECT Skills
```

```
-> FROM 21BCI0045_Profiles
```

```
-> GROUP BY Skills
```

```
-> HAVING COUNT(*) = 1
```

```
-> );
```

```
mysql> ALTER TABLE 21BCI0045_Profiles
-> ADD COLUMN Skills VARCHAR(20);
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> UPDATE 21BCI0045_Profiles
-> SET Skills='Java'
-> WHERE UserID=1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE 21BCI0045_Profiles
-> SET Skills='Python'
-> WHERE UserID=2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> SELECT 21BCI0045_Users.UserID, 21BCI0045_Users.Name
-> FROM 21BCI0045_Users
-> JOIN 21BCI0045_Profiles ON 21BCI0045_Users.UserID = 21BCI0045_Profiles.UserID
-> WHERE Skills IN (
-> SELECT Skills
-> FROM 21BCI0045_Profiles
-> GROUP BY Skills
-> HAVING COUNT(*) = 1
-> );
+-----+-----+
| UserID | Name |
+-----+-----+
| 1      | Ram Babu |
| 2      | Vijay Shankar |
+-----+-----+
2 rows in set (0.01 sec)
```


Calculate the average number of connections per user

```
mysql> SELECT AVG(ConnectionCount) AS AverageConnections
```

```
-> FROM (
```

```
-> SELECT COUNT(*) AS ConnectionCount
```

```
-> FROM 21BCI0045_Connections
```

```
-> GROUP BY UserID1
```

```
-> ) AS ConnectionCounts;
```

```
mysql> SELECT AVG(ConnectionCount) AS AverageConnections
-> FROM (
-> SELECT COUNT(*) AS ConnectionCount
-> FROM 21BCI0045_Connections
-> GROUP BY UserID1
-> ) AS ConnectionCounts;
+-----+
| AverageConnections |
+-----+
|          1.5000    |
+-----+
1 row in set (0.01 sec)
```

Retrieve the users who have not posted any job openings

```
mysql> SELECT 21BCI0045_Users.UserID, 21BCI0045_Users.Name
```

```
-> FROM 21BCI0045_Users
```

```
-> LEFT JOIN 21BCI0045_Jobpostings ON 21BCI0045_Users.UserID =
21BCI0045_JobPostings.UserID
```

```
-> WHERE 21BCI0045_Jobpostings.JobPostingID IS NULL;
```

```
mysql> SELECT 21BCI0045_Users.UserID, 21BCI0045_Users.Name
-> FROM 21BCI0045_Users
-> LEFT JOIN 21BCI0045_Jobpostings ON 21BCI0045_Users.UserID = 21BCI0045_JobPostings.UserID
-> WHERE 21BCI0045_Jobpostings.JobPostingID IS NULL;
+-----+-----+
| UserID | Name          |
+-----+-----+
|      6 | Ashok Mahendra |
+-----+-----+
1 row in set (0.00 sec)
```

Retrieve users who have connections in a specific industry

```
mysql> SELECT 21BCI0045_Users.UserID, 21BCI0045_Users.Name
```

```
-> FROM 21BCI0045_Users
```

```
-> JOIN 21BCI0045_Jobpostings ON 21BCI0045_Users.UserID = 21BCI0045_Jobpostings.UserID
```

-> WHERE 21BCI0045_Jobpostings.Industry = 'Technology';

```
mysql> SELECT 21BCI0045_Users.UserID, 21BCI0045_Users.Name
-> FROM 21BCI0045_Users
-> JOIN 21BCI0045_Jobpostings ON 21BCI0045_Users.UserID = 21BCI0045_Jobpostings.UserID
-> WHERE 21BCI0045_Jobpostings.Industry = 'Technology';
+-----+-----+
| UserID | Name      |
+-----+-----+
|      4 | Rahul Sharma |
+-----+-----+
1 row in set (0.00 sec)
```

Modify a job posting's salary range

```
mysql> ALTER TABLE 21BCI0045_Jobpostings
```

-> ADD COLUMN SalaryRange VARCHAR(20);

Query OK, 0 rows affected (0.02 sec)

Records: 0 Duplicates: 0 Warnings: 0

```
mysql> UPDATE 21BCI0045_Jobpostings
```

-> SET SalaryRange = '80000-100000'

-> WHERE JobpostingID = 1;

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> ALTER TABLE 21BCI0045_Jobpostings
-> ADD COLUMN SalaryRange VARCHAR(20);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> UPDATE 21BCI0045_Jobpostings
-> SET SalaryRange = '80000-100000'
-> WHERE JobpostingID = 1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

Modify a user's employment status.

```
mysql> UPDATE 21BCI0045_Users
```

-> SET EmploymentStatus = 'Unemployed'

-> WHERE UserID = 1;

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> UPDATE 21BCI0045_Users
-> SET EmploymentStatus = 'Unemployed'
-> WHERE UserID = 1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

Retrieve the number of job postings per industry

```
mysql> SELECT Industry, COUNT(*) AS JobPostingCount
-> FROM 21BCI0045_Jobpostings
-> GROUP BY Industry;
```

```
mysql> SELECT Industry, COUNT(*) AS JobPostingCount
-> FROM 21BCI0045_Jobpostings
-> GROUP BY Industry;
```

Industry	JobPostingCount
Analytics	1
Management	1
Design	1
Technology	1
Marketing	1

5 rows in set (0.00 sec)

Identify the essential transactions in the LinkedIn database.

Essential Transactions in the LinkedIn database

Essential transactions in the LinkedIn database could include creating a user profile, establishing a connection with another user, posting a job opening, applying for jobs, updating profile information, sending connection requests, etc. Each of these transactions would involve multiple database operations and need to be handled as a single unit to maintain data consistency.

Discuss the ACID properties and transaction states.

ACID PROPERTIES

Atomicity: Atomicity guarantees that either all the changes made by a transaction are committed, or none of them are. In the LinkedIn database, various transactions occur, such as creating a user profile, establishing connections, posting job openings, and updating profile information. For example, when creating a user profile, if any part of the transaction fails (e.g., due to an error or constraint violation), the entire transaction is rolled back, ensuring that no partial changes are left in the database. This helps maintain data consistency and prevents incomplete or inconsistent user profiles or connections.

Consistency: Consistency ensures that a transaction brings the database from one valid state to another valid state. In the LinkedIn database, consistency is maintained by enforcing data integrity

constraints and rules. For instance, when updating a user's employment status or modifying a job posting, the database ensures that the changes made comply with the defined rules and constraints. Consistency also ensures that the relationships between entities (e.g., connections between users) remain valid and consistent throughout the transaction.

Isolation: Isolation ensures that concurrent transactions do not interfere with each other and are executed in isolation. In the LinkedIn database, multiple users can perform various actions simultaneously, such as creating profiles, connecting with others, and posting job openings. Isolation prevents data anomalies and conflicts by controlling how transactions access and modify data concurrently.

Durability: Durability guarantees that once a transaction is committed, its changes are permanent and survive any subsequent failures. In the LinkedIn database, committed transactions, such as creating profiles or posting job openings, persist even in the event of power outages, system crashes, or other failures. Durability is achieved by ensuring that the committed data is stored reliably and can be recovered in case of failures. This typically involves techniques like write-ahead logging, transaction logging, and periodic backups to ensure data durability and recoverability.

TRANSACTION STATES

Active: The transaction is in progress and executing a series of operations. For example, when a user creates a profile, the transaction enters the active state while inserting data into the User and Profile tables.

Partially Committed: The transaction has executed successfully, and its changes are not yet visible to other transactions. At this stage, the database system is preparing to commit the transaction. For example, when a user establishes connections with other users, the changes are not yet visible to other users until the transaction is partially committed.

Committed: The transaction has executed successfully, and its changes are permanently saved and visible to other transactions. In the LinkedIn database, when a transaction is committed, the created user profiles, connections, and job postings become visible and can be accessed by other users.

Aborted: The transaction has encountered an error or been explicitly rolled back, and its changes have been discarded. This can happen when an error occurs during a transaction or when a user decides to cancel or rollback a transaction. In such cases, the database ensures that any changes made by the aborted transaction are reverted, and the data returns to its previous consistent state.

ER Diagram:

