

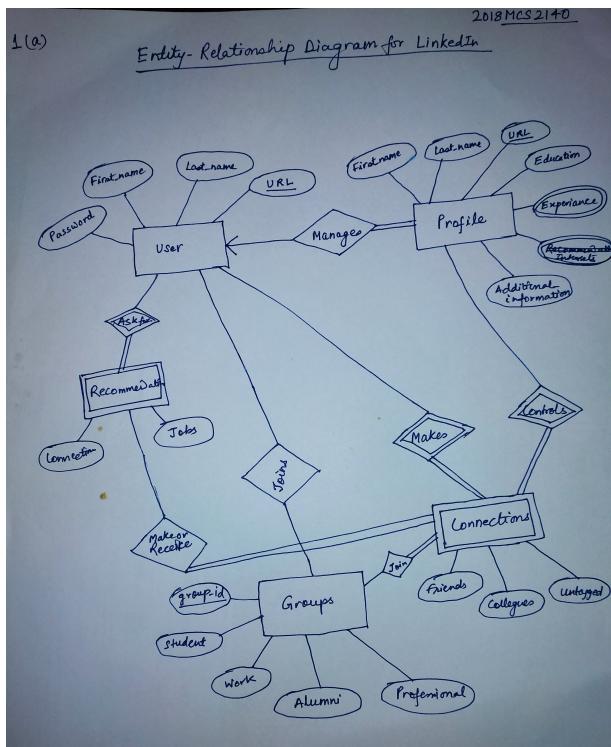
Assignment 1

Nitish Raj, 2018MCS2140

Due date: January 14, 2019, 11:55pm IST

1 Answer 1

1.1 ER Diagram



1.2 Set of relations

Relations

User			
URL	First-name	Last-name	password

Profile				
URL	First-name	Last-name	Education	Experience

Connections		
Friends	Colleagues	untagged

Groups				
group_id	student	work	Alumni	Professional

Recommendation	
Connection	Job

1.3 Keys and FDs

L.C) URL is primary key for relation "User" and "Profile".
 Consider it as key for relation "Groups".
 Relation "Connections" & "Friends" have no key in design.
 As they are weak entities in ER model.

1.4 Sample Data

1(d) User

URL	Firstname	Lastname	PassWord
https://www.linkedin.com/in/nitch-ray-53012465/	Nitch	Raj	ABC123
https://www.linkedin.com/in/pandeep-bansal-1aa0a57f/	Pandeep	Bansal	XYZ456
https://www.linkedin.com/in/avish-patil-2438811/	AMIT	PASRICHA	QPL783
https://www.linkedin.com/in/mamanath/	Maya	Ramanath	QUERTY123
https://www.linkedin.com/in/abhishek-dey-083a035/	Abhishek	Dey	QRSTAB12

VRC	Firstname	Lastname	Education	Experience	Interest	Additional info
Same as User Relation	Nitch	Raj	B.Tech IITB	Scientist	IITD	Vanc avancered
	Pandeep	Bansal	B.Tech	Sr. SW Devloper	Danmagi	Learning job along
	AMIT	PASRICHA	B.Tech	Sr. SDE Engg.	Adobe	10M certified
	Maya	Ramanath	IITSC	Assistant Engg	IIT Ropar	Attended MaxPlaza
	Abhishek	Dey	IIT KGP 1998-2002 offcr.	Harvard Business School	Stanley at Jamnagar	

Connections

friends	colleagues	un>tagged
True	True	False
True	True	False
True	True	False
True	True	True
True	True	False

1.5 List of various relationships

1. Manages
2. Controls
3. Makes
4. Joins
5. Ask for

2 Answer 2

3 Answer 3

3.1 Schema

I have used barcelona data sets. I have created schema called barcelona_data_sets using following command.

```
CREATE SCHEMA barcelona_data_sets;
```

Furthermore, Table immigrants_by_nationality is created using following command.

```
CREATE TABLE barcelona_data_sets.immigrants_by_nationality(year smallint,DistrictCode small-int, DistrictName VARCHAR(50),NeighborhoodCode smallint, NeighborhoodName VARCHAR(50), Nationality TEXT, Number smallint );
```

3.2 Various insert modes

3.2.1 Bulk Load

Bulk Load is done using following command

```
COPY barcelona_data_sets.immigrants_by_nationality FROM 'D:\Study\HITD IIInd Semester\\COL 632\barcelona-data-sets\immigrants_by_nationality.csv' with CSV HEADER;
```

Terminal o/p

COPY 70448

Time: 594.143 ms

3.2.2 Insert statements

```
package csvToSql;

import java.io.*;

public class csvToSqlConverter {

    public static void main (String s[]) throws IOException
    {
        File ipFile = new File("immigrants_by_nationality.csv");
        File opFile = new File("immigrants_by_nationality.sql");

        try { Convert(ipFile ,opFile); }
        finally { }

    }

    public static void Convert(File source , File dest) throws IOException
    {
        BufferedReader in = new BufferedReader(new FileReader(source));
        Writer out = new BufferedWriter(new FileWriter(dest));

        String tuple = null;
        String [] myString = null;

        in.readLine(); // for schema head mentioned in datasets

        while ((tuple = in.readLine()) != null )
        {

            myString = tuple.replace("\r\n", "\r").split(",");
            out.write("INSERT INTO barcelona_data_sets.immigrants_
_by_nationality VALUES(" + myString[0] + "," + myString[1] +
",\r" + myString[2] + "\r," + myString[3] + ",\r" + myString[4] +
"\r,\r" + myString[5] + "\r," + myString[6] + ");\r");
        }

        in.close();
        out.close();

    }
}
```

3.2.3 using JDBC

BULK MODE USING JDBC CONNECTION

```
import java.io.*;
import java.sql.*;

public class bulkJdbcInsert {

    public static void main (String s[]) throws IOException , ClassNotFoundException
        , SQLException
    {
        try {
            bulkInsertJDBC();
        }
        finally
    }

    public static void bulkInsertJDBC() throws IOException , ClassNotFoundException
        , SQLException
    {

        Class.forName("org.postgresql.Driver");
        Connection connection = DriverManager.getConnection(
            "jdbc:postgresql://127.0.0.1:5432/dbmsAssignment_db" , "postgres" , "1234");

        connection.createStatement().executeUpdate("COPY barcelona_data_sets.
immigrants_by_nationality FROM 'D:\\Study\\IITD IInd Semester\\COL 632
\\barcelona-data-sets\\immigrants_by_nationality.csv' with CSV HEADER;");

    }
}
```

PER TUPLE INSERTION USING JDBC CONNECTION

```
import java.io.*;
import java.sql.*;

public class singleJDBCInsert {

    public static void main (String s[]) throws IOException , ClassNotFoundException
        , SQLException
    {
        File ipFile = new File("immigrants_by_nationality.csv");

        try {
            singleInsertJDBC(ipFile);
        }
        finally
    }

    public static void singleInsertJDBC(File source) throws IOException ,
        ClassNotFoundException , SQLException
    {
        BufferedReader in = new BufferedReader(new FileReader(source));
    }
}
```

```

        Class.forName("org.postgresql.Driver");
        Connection connection = DriverManager.getConnection("jdbc
                :postgresql://127.0.0.1:5432/dbmsAssignment_db", "postgres",
                "postgres");

        String tuple = null;
        String[] myString = null;

        in.readLine(); // for schema head mentioned in datasets

        while ((tuple = in.readLine()) != null)
        {
            Statement stmt;
            myString = tuple.replace("\r\n", "\r").split(",");
            connection.createStatement().executeUpdate("INSERT INTO
                barcelona_data_sets.immigrants_by_nationality VALUES(" + myString[0] + "\r
                \n" + myString[1] + "\r" + myString[2] + "\r" + myString[3] + "\r" + myString[4] + "\r" + myString[5] + "\r\n" + myString[6] + "\r\n");
        }

    }
}

```

3.3 Statistics

Processor : Intel Core i7 8th Gen
 Memory : 8 GB RAM
 BulkLoad for three data sets of Barcelona is 300 ms
 .SQL took 4 sec
 JDBC-Bulk 1 sec
 JDBC-Single 9 min

4 Answer 4

Barcelona data is copied multiple times and created more than 5GB file.
 BulkLoad taking 15 minute
 sql more than 1 day
 JDBC- bulk : 16 minute around
 JDBC single : more than a day