Tutorial for setting up assignment environment

Purpose of this tutorial is to guide you towards setting up the working environment for the assignment. Before the class, you need to finish the **Virtual Machine** section. During the class, we will start from the **Postgres** section.

PS. If you are using your system without using VM provided by us, you need to make sure your system has **Postgres** and **Python** installed and they are compatible with the following set up:

- python 2.7,
- Ubuntu 16.04,
- PostgreSQL 9.5

The provided VM should have the required setup. You can check it to make sure.

Virtual Machine

1 . Download all the files from the link: https://drive.google.com/open?id=1_efzBidI7waVEjVxJw2OJ3kv2PZO-dNm. Put all three files in the same directory. The file is only accessible for ASU Google Driveaccount. Make sure you are logging in as asu account.
2 . Load files into either VMWare or VirtualBox.

- VMWare (recommended). It provides a free trial for CIDSE student for one year. The link is https://e5.onthehub.com/WebStore/OfferingsOfMajorVersionList.as pmv=e7d3dd90-8b51-e511-940f-b8ca3a5db7a1&cmi_mnuMain=16a020b5-ed3c-df11-b4ab-0030487d8897&cmi_mnuMain_child=aafc5891-884f-e511-940f-b8ca3a5db7a1&cmi_mnuMain_child_child=6130e417-ad1a-e511-940d-b8ca3a5db7a1&ws=dbcd06b7-86b0-e411-9408-b8ca3a5db7a1&vsro=. Use File->Open to load the .ovf file.
- VirtualBox. It can be downloaded online. Use File->Import Appliance to load the .ovf file.
  The default set up of the machine is Memory: 4GB, Processor: 4, Disk: 15GB. You can modify them in the settings for your purpose.

3 . Start the machine and log into the default user “cse512assignment”. The login password is: user

Postgres with Terminal (For your playing with Postgres, not required)

Postgres has already been installed in the virtual machine. You can directly try with it.

1 . Open a terminal. Log into Postgres with the default root user: postgres

```
sudo -su postgres psql
```

2 . Create a test database and try some operations.

create a test database and switch to it

```
create database test;
\c test
```

create a table:

```
CREATE TABLE USERS(
   USERID INT PRIMARY KEY NOT NULL,
   NAME TEXT NOT NULL
);
CREATE TABLE MOVIES(
   MOVIEID INT PRIMARY KEY NOT NULL,
   TITLE TEXT NOT NULL,
   Genre TEXT
);
CREATE TABLE RATINGS(
   USERID INT REFERENCES USERS(USERID),
   MOVIEID INT REFERENCES MOVIES(MOVIEID),
   RATING NUMERIC NOT NULL CHECK(RATING>=0 AND RATING<=5)
);
```

check the schema:

```
select * from users;
select * from movies;
select * from ratings;
```

insert some data:

```
# users table
INSERT INTO users VALUES (1, 'David');
INSERT INTO users VALUES (2, 'Eric');
INSERT INTO users VALUES (3, 'Kevin');

# movies table
COPY MOVIES FROM 'movies.dat' delimiter '_';

# ratings table
insert into ratings VALUES (1, 122, 5.0);
insert into ratings VALUES (1, 185, 4.5);
insert into ratings VALUES (2, 231, 4.0);
insert into ratings VALUES (2, 292, 3.5);
insert into ratings VALUES (3, 316, 3.0);
```

perform search:

```
select * from ratings where userid=1;
select * from ratings where rating>=3;


select * from users, movies, ratings
where users.name='David'
    and users.userid=ratings.userid
    and movies.movieid=ratings.movieid;


select users.name, movies.title
from users, movies, ratings
where users.name='David'
    and users.userid=ratings.userid
    and movies.movieid=ratings.movieid;


select users.name, avg(ratings.rating)
from users, ratings
    where users.userid=ratings.userid
    group by users.name;
```

switch back to postgres db and delete the test db

```
\c postgres
drop database test
\l
```

Connect to Postgres in Python

We use psycopg2 package to connect to Postgres in python. It has been installed in the vm. The password of the user “postgres” has been modified to “1234”.

1 . Go to python shell and connect to the database named with “postgres”. Create a test database:

```
# python shell
python
# connect to Postgres
import psycopg2
con = psycopg2.connect("dbname='postgres' user='postgres'
host='localhost' password='1234'")
con.set_isolation_level(psycopg2.extensions.ISOLATION_LEVE
L_AUTOCOMMIT)
cur = con.cursor()
cur.execute("create database test")
# disconnect from Postgres
cur.close()
con.close()
```

2 . Connect to the test database and build the same table:

```
con = psycopg2.connect("dbname='test' user='postgres' host
='localhost' password='1234'")
con.set_isolation_level(psycopg2.extensions.ISOLATION_LEVE
L_AUTOCOMMIT)
cur = con.cursor()
cur.execute("create table ratings (userid integer, movieid
 integer, rating float)")

insert_command="insert into ratings VALUES (1, 122, 5.0);"
insert_command+="insert into ratings VALUES (1, 185, 4.5);
"
insert_command+="insert into ratings VALUES (2, 231, 4.0);
"
insert_command+="insert into ratings VALUES (2, 292, 3.5);
"
insert_command+="insert into ratings VALUES (3, 316, 3.0);
"
cur.execute(insert_command)
```

3 . Perform queries

```
cur.execute("select * from ratings")
res=cur.fetchall()
print res

cur.execute("select * from ratings where userid=1")
res=cur.fetchall()
print res

cur.execute("select * from ratings where rating>=3")
res=cur.fetchall()
print res

```

```
cur.close()
con.close()
```

4 . Drop the test database

```
con = psycopg2.connect("dbname='postgres' user='postgres'
host='localhost' password='1234'")
con.set_isolation_level(psycopg2.extensions.ISOLATION_LEVE
L_AUTOCOMMIT)
cur = con.cursor()
cur.execute("drop database test")
cur.close()
con.close()
```

<!–stackedit_data:
eyJoaXN0b3J5IjpbLTE4NTg0MDk3NjgsLTcyMjM1Mzk3NiwtNz
c3MzU3ODUzLDc0NTAwMTExNiwtNjI1ODE1NTIwLDgwNzAxOTQx
OCwtMTU1OTIzMjUzMCw1NDczNzMzNTgsLTE0NjE0NTc1OTQsLT
E3MTQ1NjU0NjEsMTE3NDM5NzA0OSw3MDY1NzkwMjgsMTA2NzA5
ODEwLC01MDA5MjIxNjAsLTYzNDk2NjQxNCwtNzgzNjk1NjI2XX
0=
–>