Database Management Systems Homework 2 submission Tyrus Malmstrom :: 829947336

Here are the my “Create Table” scripts for homework 2. I was originally going to upload this as a seperate .sql in conjunction with this file, however canvas didn't accept the .sql file. Thus, I have just added the scripts into this file.

```
/* Tyrus Malmstrom :: CS430 --> Database Managment Systems :: Homework 2 */
/* Script that will setup all the tables required from the ER Diagram */

-- Good Etique --

\echo '\nDropping schema if it already exists.'

DROP SCHEMA IF EXISTS thisMusicProduction CASCADE;

\echo '\nDropping tables if they exist.'

DROP TABLE IF EXISTS thisMusicProduction.Musicians  CASCADE;
DROP TABLE IF EXISTS thisMusicProduction.Instrument CASCADE;
DROP TABLE IF EXISTS thisMusicProduction.Album      CASCADE;
DROP TABLE IF EXISTS thisMusicProduction.Songs      CASCADE;
DROP TABLE IF EXISTS thisMusicProduction.Place      CASCADE;
DROP TABLE IF EXISTS thisMusicProduction.Telephone  CASCADE;

-----------------------------------------------

\echo '\nCreating schema \'thisMusicProduction\' in which all my relational tables will reside.'

-- Creating the Schema in which all my relational tables will live:
CREATE SCHEMA thisMusicProduction;

\echo '\nCreating Musicians table.'

-- Creating the Musicians table:
CREATE TABLE thisMusicProduction.Musicians(

   ssn  varchar (11) PRIMARY KEY,
   name varchar (15)
);

\echo '\nInserting values into Musicians table.'

-- Insterting values into the Musicians
INSERT INTO thisMusicProduction.Musicians(
       ssn,
       name
```

```
)
VALUES
('147-45-4789','John'),
('256-69-7858', 'Smith'),
('123-45-6789','Tyrus'),
('789-45-6123','Bob'),
('159-52-1563','deadmau5');

-----------------------------------------------------

-----------------------------------------------------

\echo '\nCreating Instrument table.'

-- Creating the instrument table:
CREATE TABLE thisMusicProduction.Instrument(

    instrid integer PRIMARY KEY,
    dname varchar(15),
    key   varchar(15)

);

\echo '\nInserting values into Instrument table.'

-- Inserting values into the Instruments table:
INSERT INTO thisMusicProduction.instrument(
    instrid,
    dname,
    key
)
VALUES
(1,'Violin','C#'),
(2,'Piano', 'A Minor'),
(3,'Guitar', 'G'),
(4,'Electronic DAW','All keys'),
(5,'Harmonica','A Major');

-------------------------------------------------

-------------------------------------------------

\echo '\nCreating Album table.  Also including the \"Producer\" relationship with the \"Musicians\"
table'.

-- Creating the Album table:
CREATE TABLE thisMusicProduction.Album(
```

```
    albumIdentifier INTEGER PRIMARY KEY,
    copyrightDate   DATE NOT NULL,
    speed           INTERVAL MINUTE,
    title           varchar(25),
    Musician_ssn    varchar(11) NOT NULL REFERENCES thisMusicProduction.Musicians( ssn ) ON
DELETE CASCADE -- Producer relationship

);

\echo '\nInserting values into Album table.'

-- Inserting values for the Album table:
INSERT INTO thisMusicProduction.Album(
    albumIdentifier,
    copyrightDate,
    speed,
    title,
    Musician_ssn
)
VALUES
( 1,'2010-07-29','10:30','For Lack of a Better Name','159-52-1563' ),
( 2,'2008-04-29','5:00', 'While(1<2)',               '159-52-1563' ),
( 3,'2009-08-29','6:50', 'Random Ablum Title',       '159-52-1563' ),
( 4,'2016-06-29','2:00', '>Album Title Goes Here<',  '159-52-1563' ),
( 5,'2007-05-29','8:30', 'W:/2016 Ablum',            '159-52-1563' );

-- SELECT * FROM thisMusicProduction.Album;

-------------------------------------------------------

-------------------------------------------------------

\echo '\nCreating Songs table. Also including the \"Appears\" relationship with the Album table.'

-- Creating the Songs table:
CREATE TABLE thisMusicProduction.Songs(

    songId integer PRIMARY KEY,
    title  varchar(30),
    suthor varchar(10),
    Album_albumIdentifier INTEGER NOT NULL REFERENCES
thisMusicProduction.Album( albumIdentifier ) ON DELETE CASCADE -- Appears relationship.

);

\echo '\nInserting values into Songs table.'
```

```
-- Inserting values for the Songs table:
INSERT INTO thisMusicProduction.Songs(
    songId,
    title,
    suthor,
    Album_albumIdentifier
)
VALUES
(1,'Strobe','deadmau5'              , 1),
(2,'My Pet Coelacanth','deadmau5'   , 2),
(3,'I Remeber ft. Kaskade','deadmau5', 3),
(4,'The Vedlt','deadmau5'           , 4),
(5,'4ware','deadmau5'               , 5);

-- SELECT * FROM thisMusicProduction.Songs;

-------------------------------------------------

-------------------------------------------------

\echo '\nCreating Place table.'

-- Creating Place table:
CREATE TABLE thisMusicProduction.Place(

    address varchar(45) PRIMARY KEY

);

\echo '\nInserting values into Place table.'

-- Inserting values for the Places table;
INSERT INTO thisMusicProduction.Place(
    address
)
VALUES
('3140. Ontario drive'),
('5148. 7 Panagioti Gyanapolou'),
('2250 W. South Drive'),
('1337 Leet Drive'),
('7331 A. This address is cool');

-------------------------------------------------

-------------------------------------------------
```

```
\echo '\nCreating Telephone table.'

-- Creating Telephone table:
CREATE TABLE thisMusicProduction.Telephone(
   phone_no TEXT UNIQUE NOT NULL
);

\echo '\nInserting values into Telephone table.'

INSERT INTO thisMusicProduction.Telephone(
   phone_no
)
VALUES
('303-653-5478'),
('654-251-4789'),
('125-458-7963'),
('111-454-7888'),
('111-222-3334');

-----------------------------------------------------------

\echo '\nNow, creating and setting up the relational tables between entities. Some may be many-to-many while others 1-to-many.\n'

-- Identifiying relationships (tables):

-- Creating the relationship talbe between Musicians and Instruments:
-- This is a *m*any to *m*any relationship --> m:m
-- Introducing foreign key constraints:

\echo '\nCreating Perform table: m:m'

CREATE TABLE thisMusicProduction.Perform(

   Musician_ssn varchar(11) REFERENCES thisMusicProduction.Musicians( ssn ) ON UPDATE CASCADE, -- update automatically if upddated in referenced / parent table.
   Songs_songId integer  REFERENCES thisMusicProduction.Songs( songId ) ON UPDATE CASCADE,
   PRIMARY KEY( Musician_ssn,Songs_songId )

);

\echo '\nInserting values into Perfom table. Values are *dependent* on values from other previously defined tables. Specifically Musicians and Songs.'

INSERT INTO thisMusicProduction.Perform(
   Musician_ssn,
```

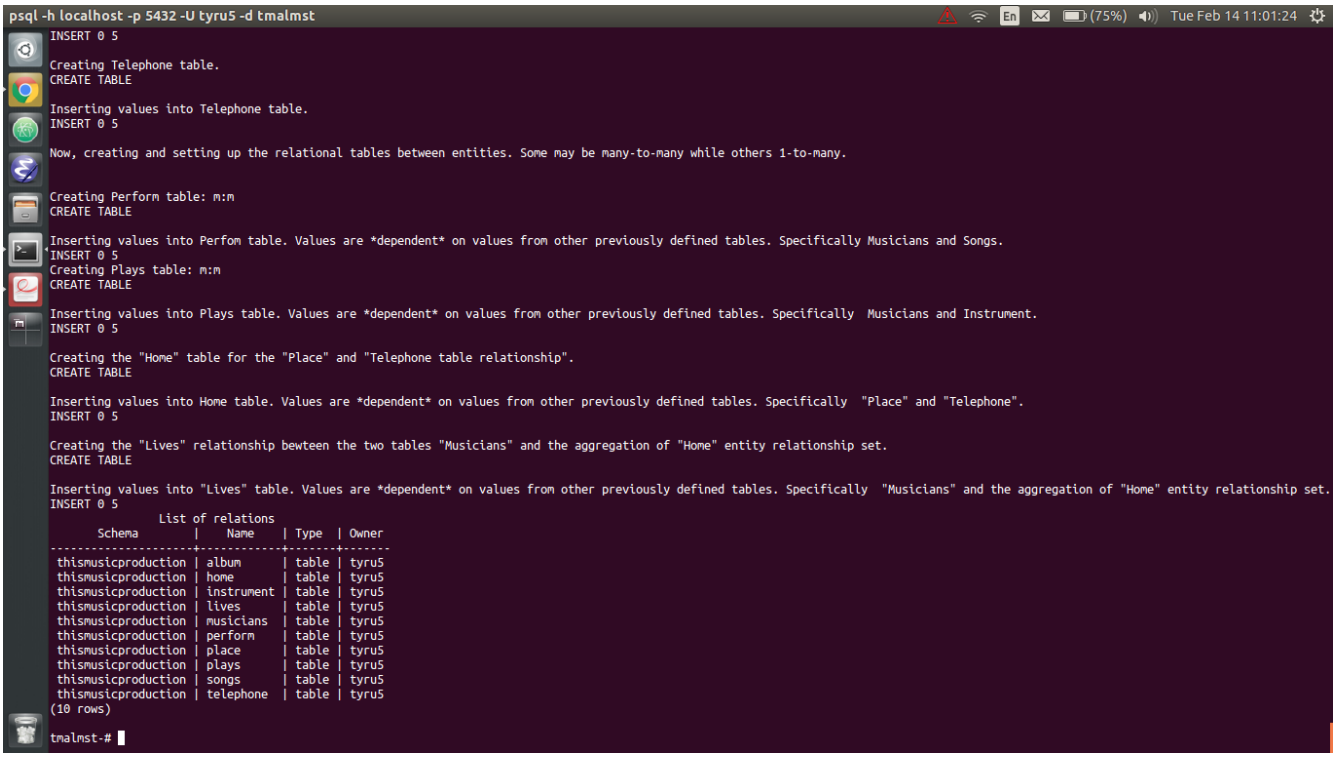
```
   Songs_songId
)
Values
('147-45-4789',1),
('256-69-7858',3),
('123-45-6789',2),
('789-45-6123',5),
('159-52-1563',4);

----------~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~----------

-- Creating the relationship talbe between Musicians and Instruments:
-- This is a *m*any to *m*any relationship --> m:m
-- Introducing foreign key constraints:\

\echo 'Creating Plays table: m:m'

CREATE TABLE thisMusicProduction.Plays(

   Musician_ssn varchar(11) REFERENCES thisMusicProduction.Musicians( ssn ) ON UPDATE
CASCADE,
   Instrument_instrid integer REFERENCES thisMusicProduction.Instrument( instrid ) ON UPDATE
CASCADE,
   PRIMARY KEY( Musician_ssn, Instrument_instrid )

);

\echo '\nInserting values into Plays table. Values are *dependent* on values from other previously
defined tables. Specifically  Musicians and Instrument.'

INSERT INTO thisMusicProduction.Plays(
   Musician_ssn,
   Instrument_instrid
)
VALUES
('159-52-1563',2),
('123-45-6789',4),
('256-69-7858',1),
('789-45-6123',3),
('147-45-4789',5);

-- SELECT COUNT(*) FROM thisMusicProduction.Plays;

----------------------------------------------------

\echo '\nCreating the \"Home\" table for the \"Place\" and \"Telephone table relationship\".'
```

```
CREATE TABLE thisMusicProduction.Home(

    Place_address varchar(45) REFERENCES thisMusicProduction.Place( address ) ON UPDATE
CASCADE,
    Telephone_phone_no TEXT UNIQUE NOT NULL REFERENCES thisMusicProduction.Telephone(
phone_no ) ON UPDATE CASCADE,
    PRIMARY KEY( Place_address )

);

\echo '\nInserting values into Home table. Values are *dependent* on values from other previously
defined tables. Specifically  \"Place\" and \"Telephone\".'

INSERT INTO thisMusicProduction.Home(
    Place_address,
    Telephone_phone_no
)
VALUES
('3140. Ontario drive',          '303-653-5478'),
('5148. 7 Panagioti Gyanapolou','654-251-4789'),
('2250 W. South Drive',         '125-458-7963'),
('1337 Leet Drive',             '111-454-7888'),
('7331 A. This address is cool','111-222-3334');

-- SELECT * FROM thisMusicProduction.Home;

------------------------------------------------------------


\echo '\nCreating the \"Lives\" relationship bewteen the two tables \"Musicians\" and the aggregation of
\"Home\" entity relationship set.'

CREATE TABLE thisMusicProduction.Lives(

    Musician_ssn varchar(11) REFERENCES thisMusicProduction.Musicians(ssn) ON UPDATE
CASCADE,
    Place_address varchar(45) REFERENCES thisMusicProduction.Home( Place_address ) ON
UPDATE CASCADE,
    PRIMARY KEY( Musician_ssn, Place_address )

);

\echo '\nInserting values into \"Lives\" table. Values are *dependent* on values from other previously
defined tables. Specifically  \"Musicians\" and the aggregation of \"Home\" entity relationship set.'

INSERT INTO thisMusicProduction.Lives(
```

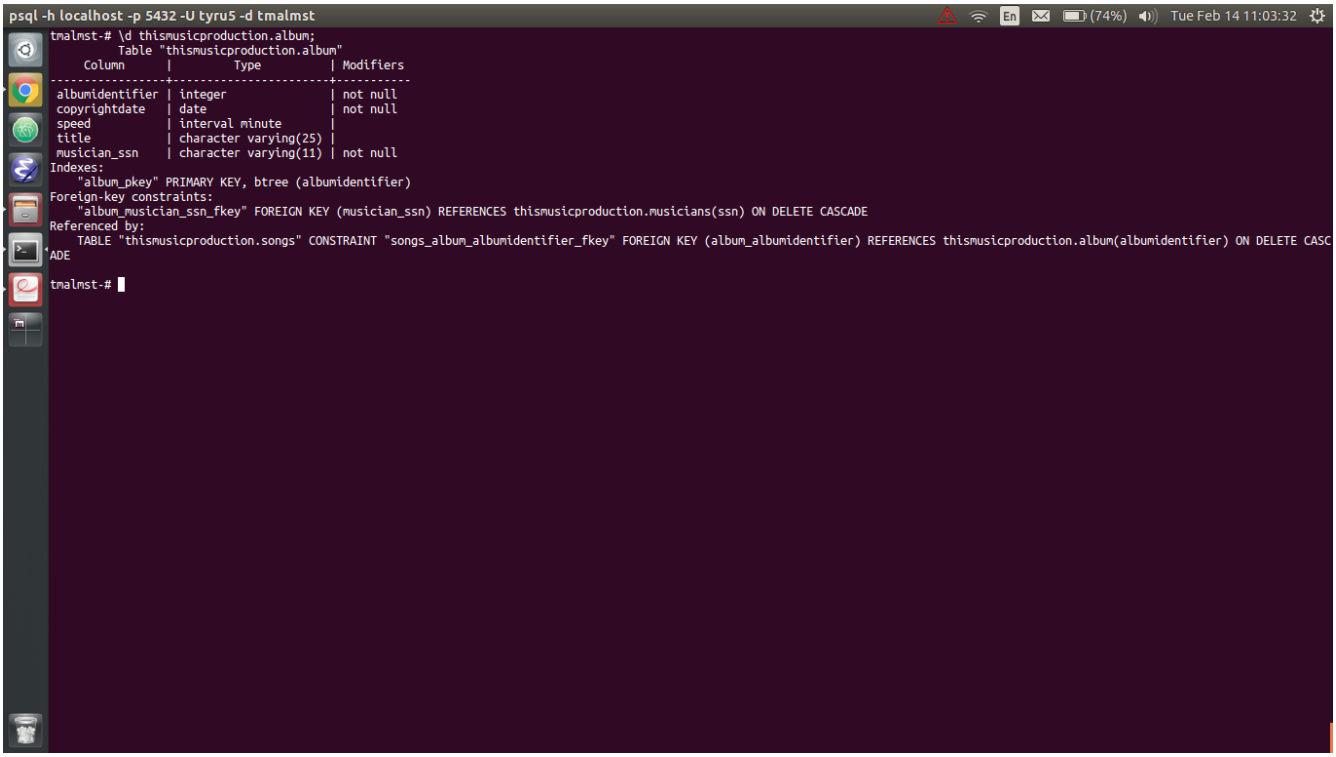
```
    Musician_ssn,
    Place_address

)
Values
('147-45-4789','2250 W. South Drive'),
('256-69-7858', '1337 Leet Drive'),
('123-45-6789','5148. 7 Panagioti Gyanapolou'),
('789-45-6123','7331 A. This address is cool'),
('159-52-1563','3140. Ontario drive');

-- SELECT * FROM thisMusicProduction.Lives;

-- Show all list of relations in PostgreSQL:
\dt thisMusicProduction.*;

-- done.
```

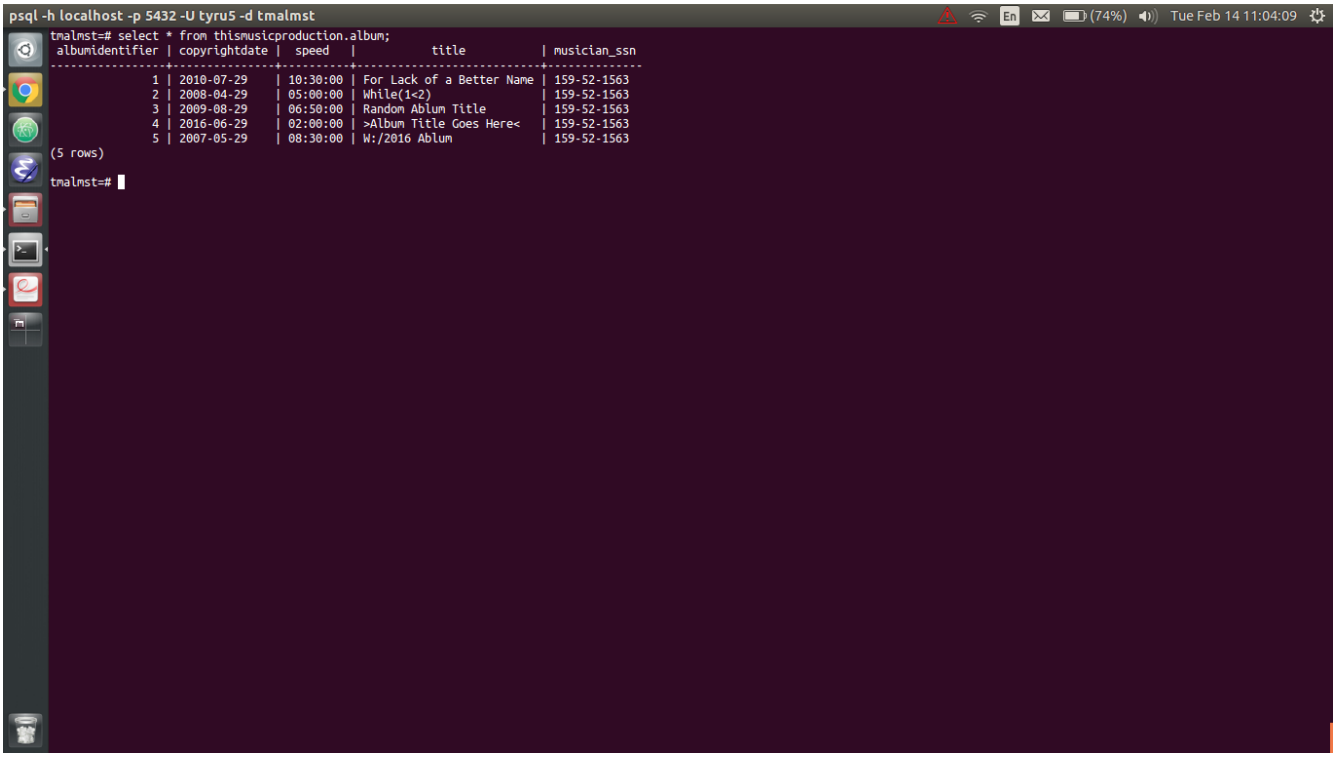
Below are all my tables. Both showing the schema of the tables, and the tables populated with data.

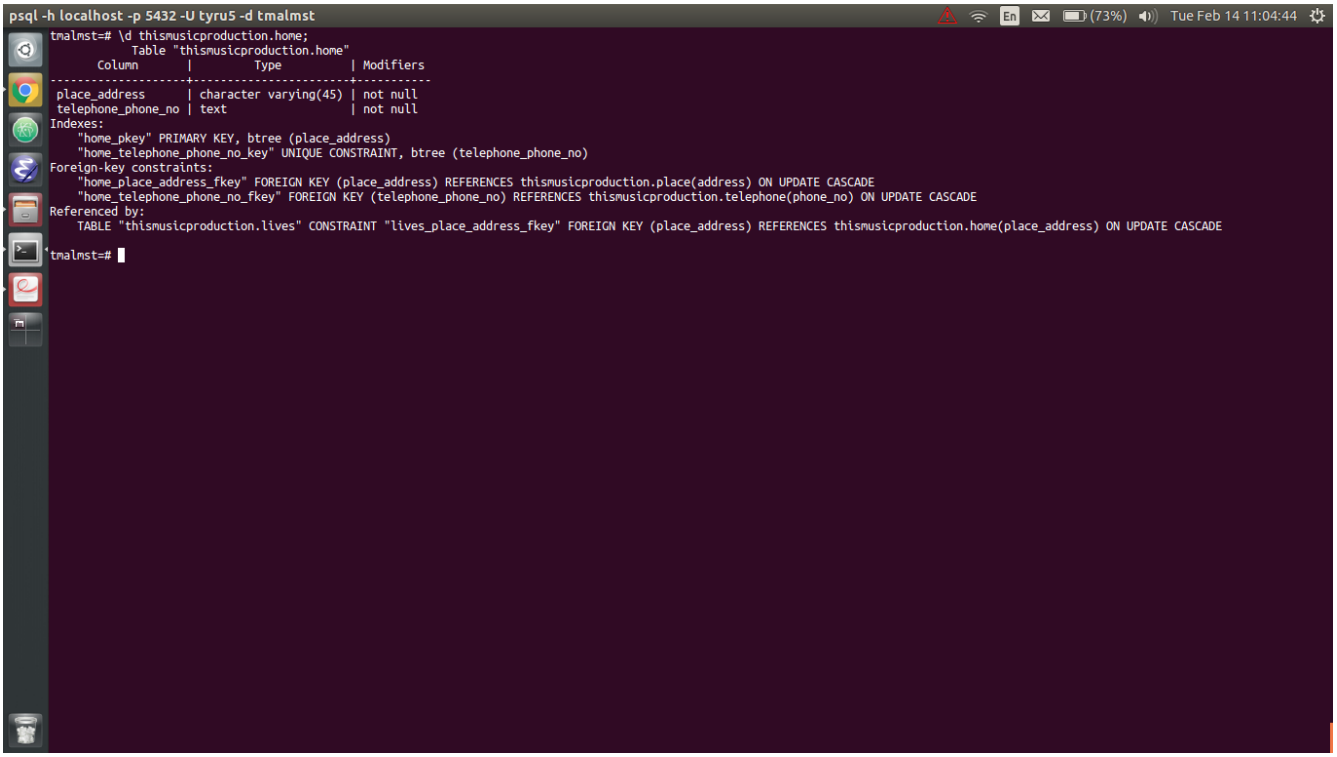
The picture below just showcases all the relations / tables within the database (specifically under the 'thismusicproduction' schema). It is the \dt <schema_name>.* command used in PostgreSQL.

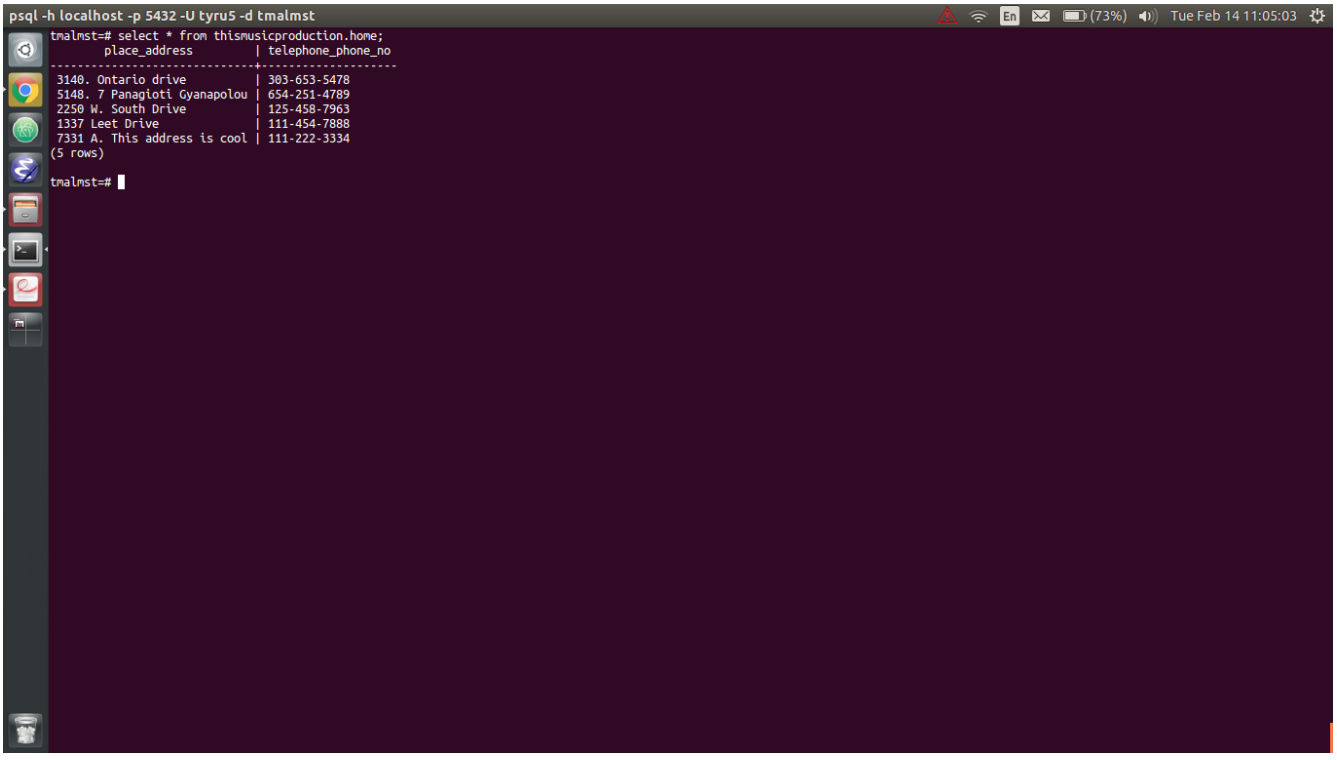


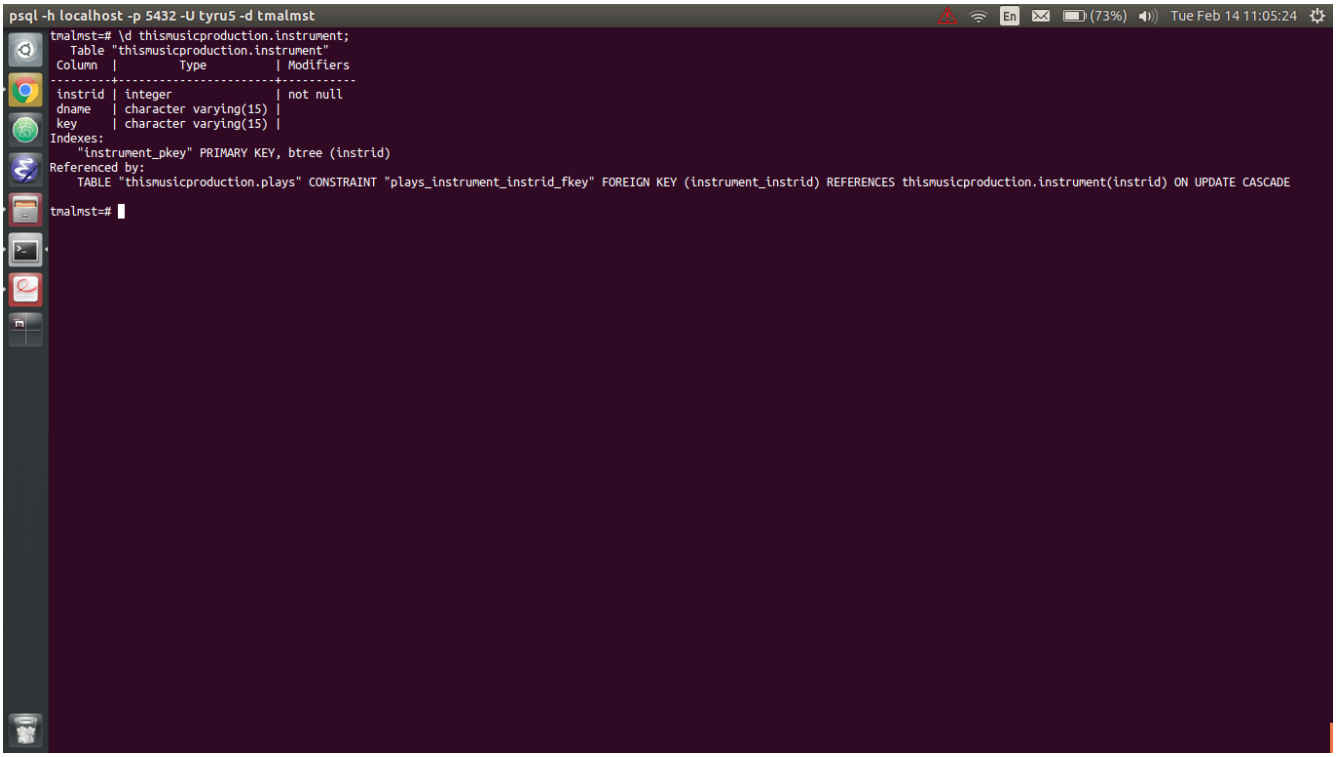
~~~~~~~~~~~~~~~~~~~~~~Now starts the listing of the relations / tables:~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~

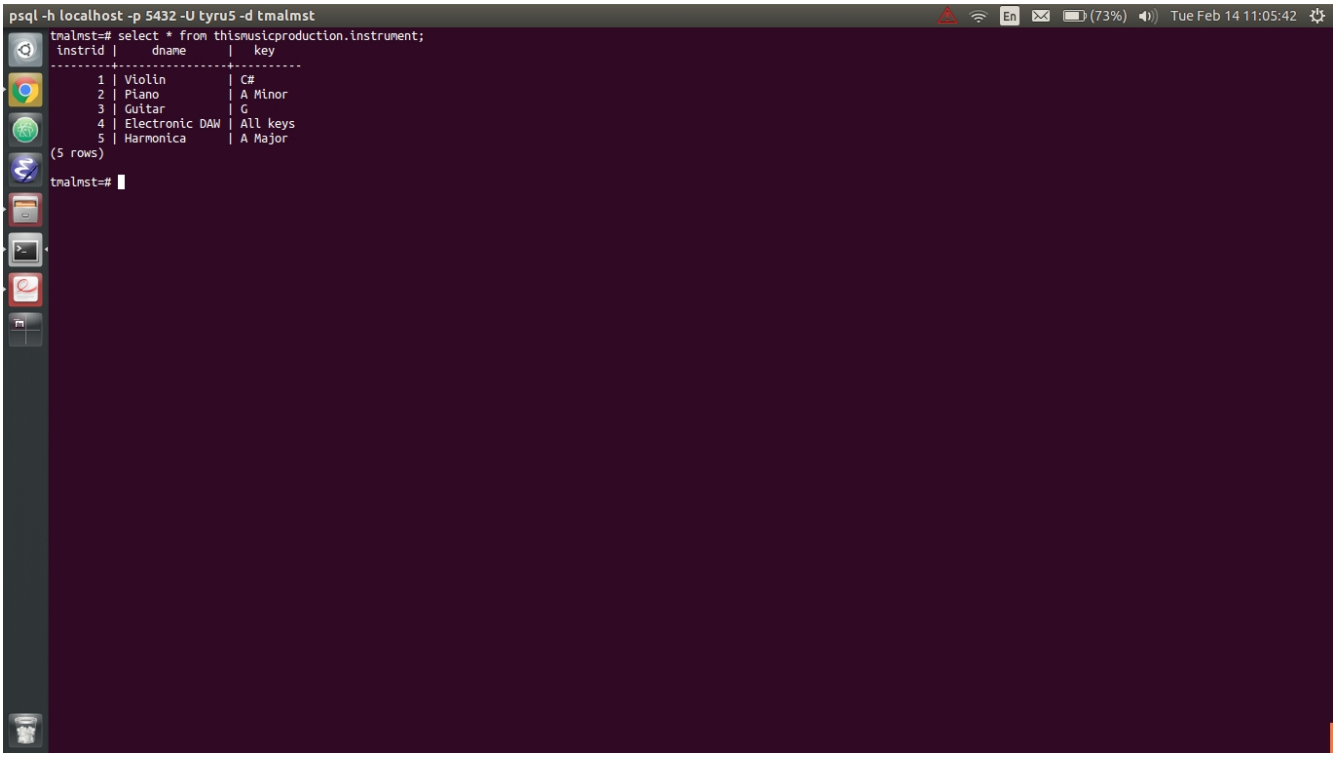

psql -h localhost -p 5432 -U tyru5 -d tmalmst
Tue Feb 14 11:03:32
tmalmst-# \d thismusicproduction.album;
Table "thismusicproduction.album"
Column | Type | Modifiers
albumidentifier | integer | not null
copyrightdate | date | not null
speed | interval minute |
title | character varying(25) |
musician_ssn | character varying(11) | not null
Indexes:
"album_pkey" PRIMARY KEY, btree (albumidentifier)
Foreign-key constraints:
"album_musician_ssn_fkey" FOREIGN KEY (musician_ssn) REFERENCES thismusicproduction.musicians(ssn) ON DELETE CASCADE
Referenced by:
TABLE "thismusicproduction.songs" CONSTRAINT "songs_album_albumidentifier_fkey" FOREIGN KEY (album_albumidentifier) REFERENCES thismusicproduction.album(albumidentifier) ON DELETE CASC
ADE
tmalmst-#

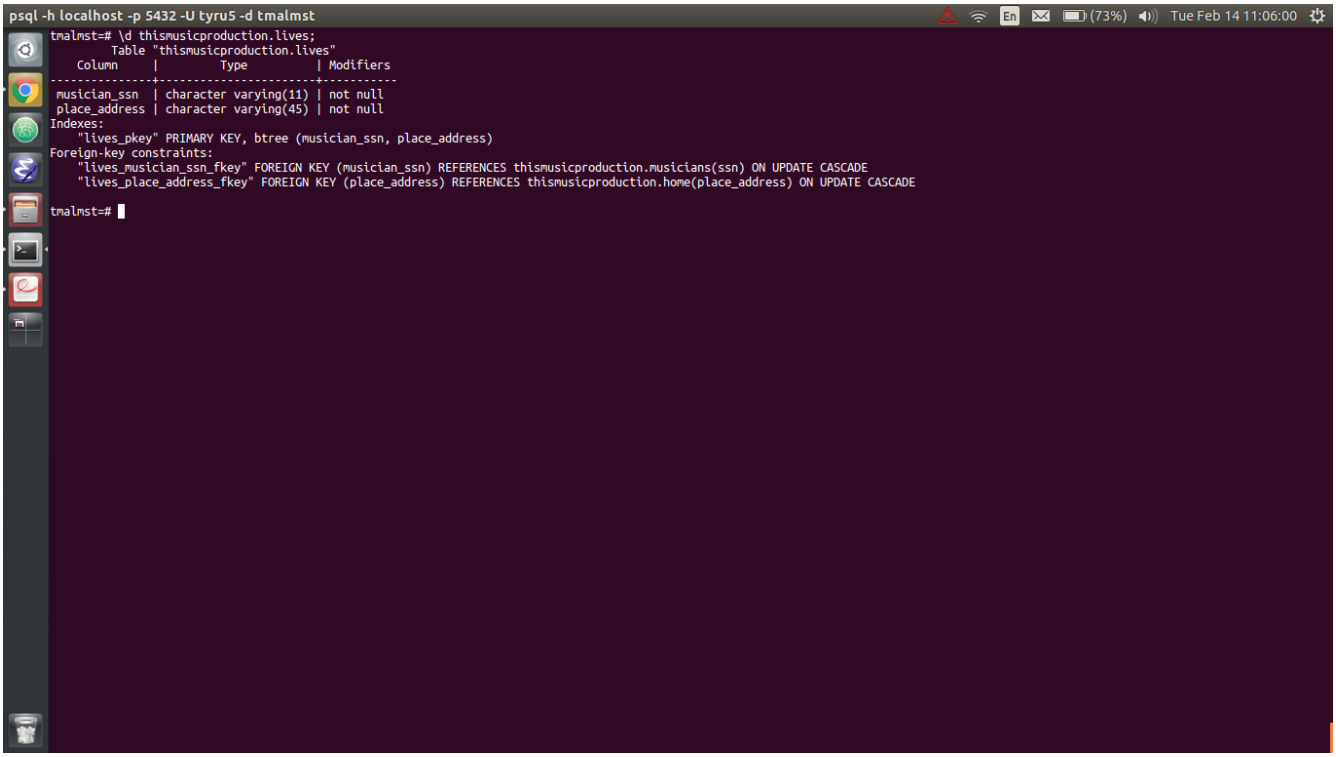

psql -h localhost -p 5432 -U tyru5 -d tmalmst
Tue Feb 14 11:04:09
tmalmst=# select * from thismusicproduction.album;
albumidentifier | copyrightdate | speed | title | musician_ssn
1 | 2010-07-29 | 10:30:00 | For Lack of a Better Name | 159-52-1563
2 | 2008-04-29 | 05:00:00 | While(1<2) | 159-52-1563
3 | 2009-08-29 | 06:50:00 | Random Ablum Title | 159-52-1563
4 | 2016-06-29 | 02:00:00 | >Album Title Goes Here< | 159-52-1563
5 | 2007-05-29 | 08:30:00 | W:/2016 Ablum | 159-52-1563
(5 rows)
tmalmst=#

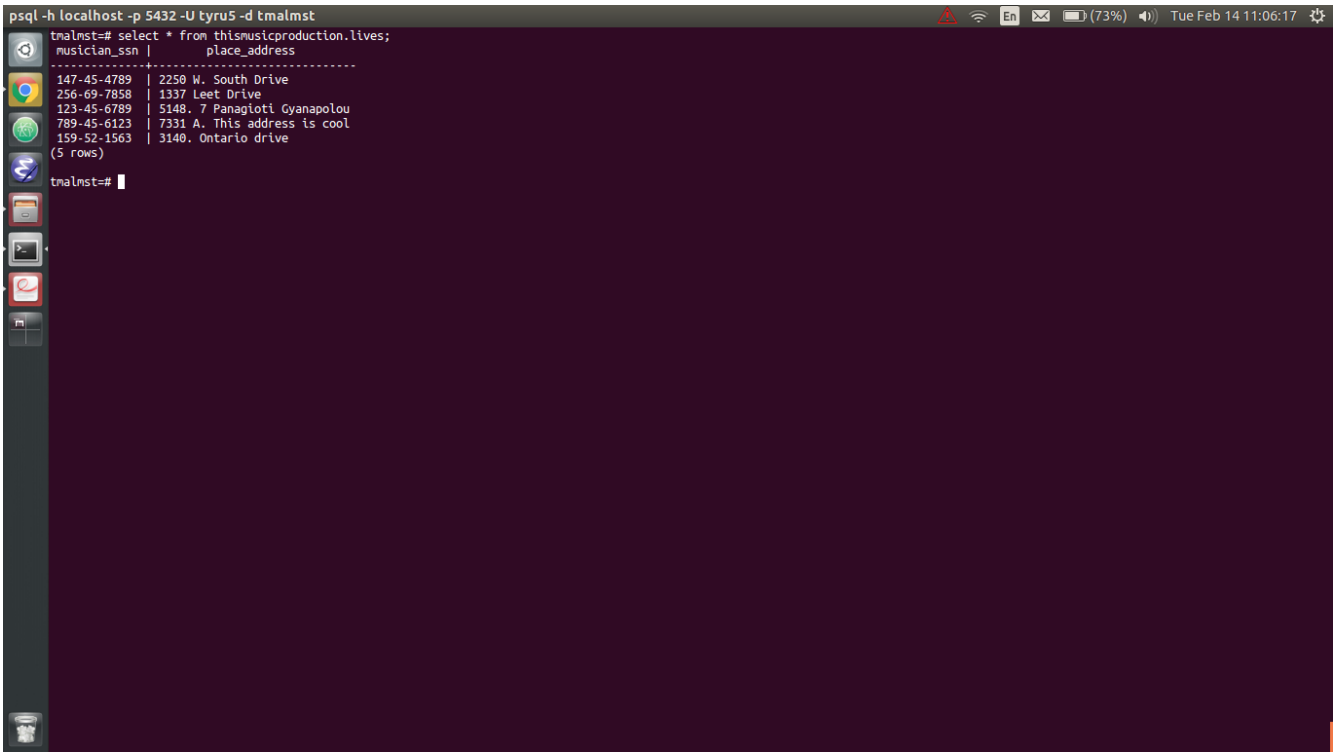

psql -h localhost -p 5432 -U tyru5 -d tmalmst
Tue Feb 14 11:04:44
tmalmst=# \d thismusicproduction.home;
Table "thismusicproduction.home"
Column | Type | Modifiers
place_address | character varying(45) | not null
telephone_phone_no | text | not null
Indexes:
"home_pkey" PRIMARY KEY, btree (place_address)
"home_telephone_phone_no_key" UNIQUE CONSTRAINT, btree (telephone_phone_no)
Foreign-key constraints:
"home_place_address_fkey" FOREIGN KEY (place_address) REFERENCES thismusicproduction.place(address) ON UPDATE CASCADE
"home_telephone_phone_no_fkey" FOREIGN KEY (telephone_phone_no) REFERENCES thismusicproduction.telephone(phone_no) ON UPDATE CASCADE
Referenced by:
TABLE "thismusicproduction.lives" CONSTRAINT "lives_place_address_fkey" FOREIGN KEY (place_address) REFERENCES thismusicproduction.home(place_address) ON UPDATE CASCADE
tmalmst=#

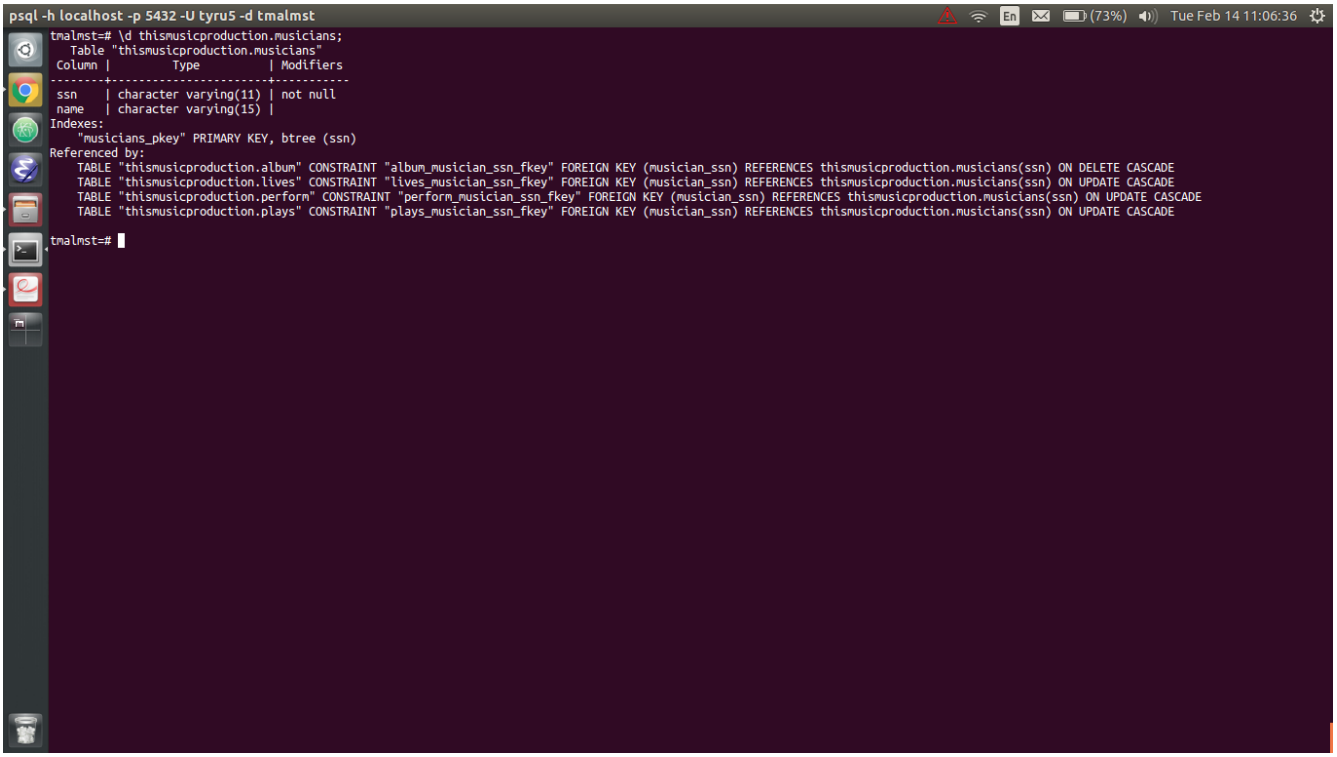

psql -h localhost -p 5432 -U tyru5 -d tmalmst
Tue Feb 14 11:05:03
tmalmst=# select * from thismusicproduction.home;
place_address | telephone_phone_no
3140. Ontario drive | 303-653-5478
5148. 7 Panagioti Gyanapolou | 654-251-4789
2250 W. South Drive | 125-458-7963
1337 Leet Drive | 111-454-7888
7331 A. This address is cool | 111-222-3334
(5 rows)
tmalmst=#

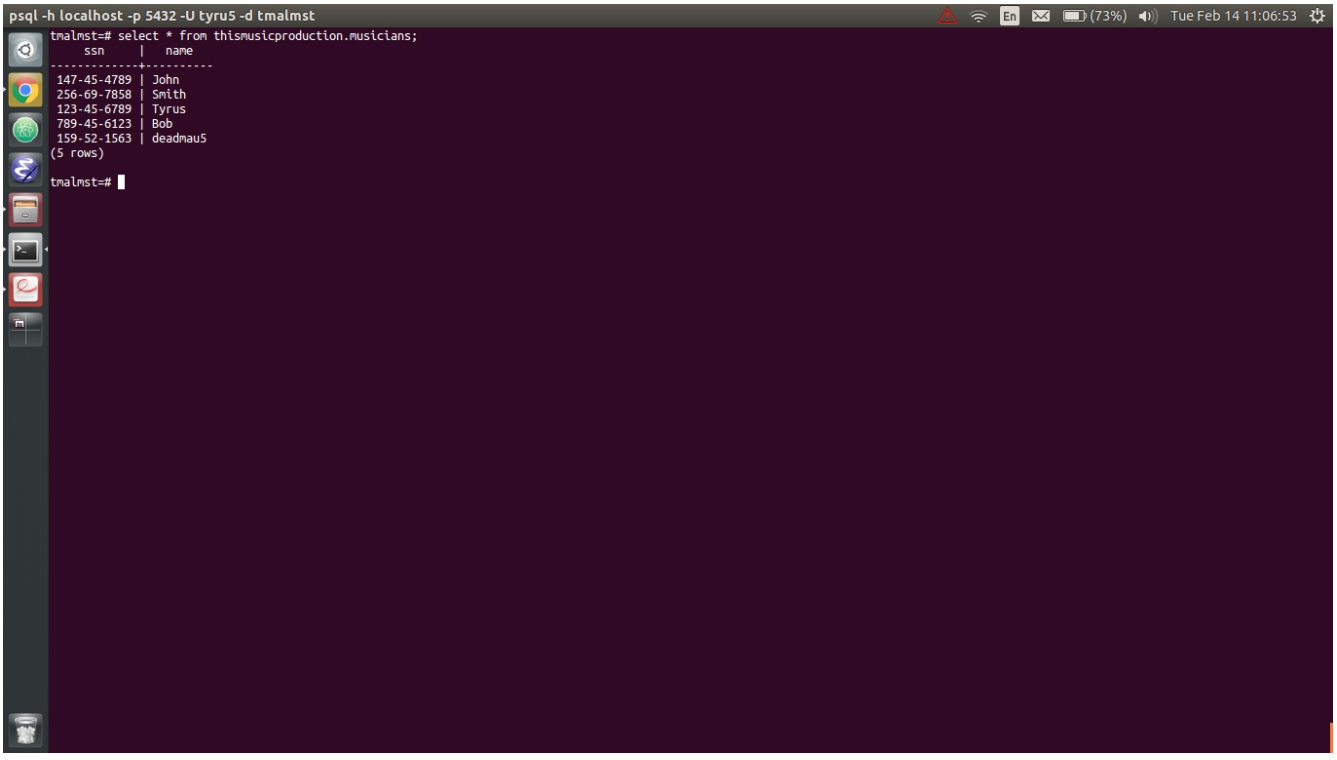

psql -h localhost -p 5432 -U tyru5 -d tmalmst
(73%) Tue Feb 14 11:05:24
tmalmst=# \d thismusicproduction.instrument;
Table "thismusicproduction.instrument"
Column | Type | Modifiers
instrid | integer | not null
dname | character varying(15) |
key | character varying(15) |
Indexes:
"instrument_pkey" PRIMARY KEY, btree (instrid)
Referenced by:
TABLE "thismusicproduction.plays" CONSTRAINT "plays_instrument_instrid_fkey" FOREIGN KEY (instrument_instrid) REFERENCES thismusicproduction.instrument(instrid) ON UPDATE CASCADE
tmalmst=#

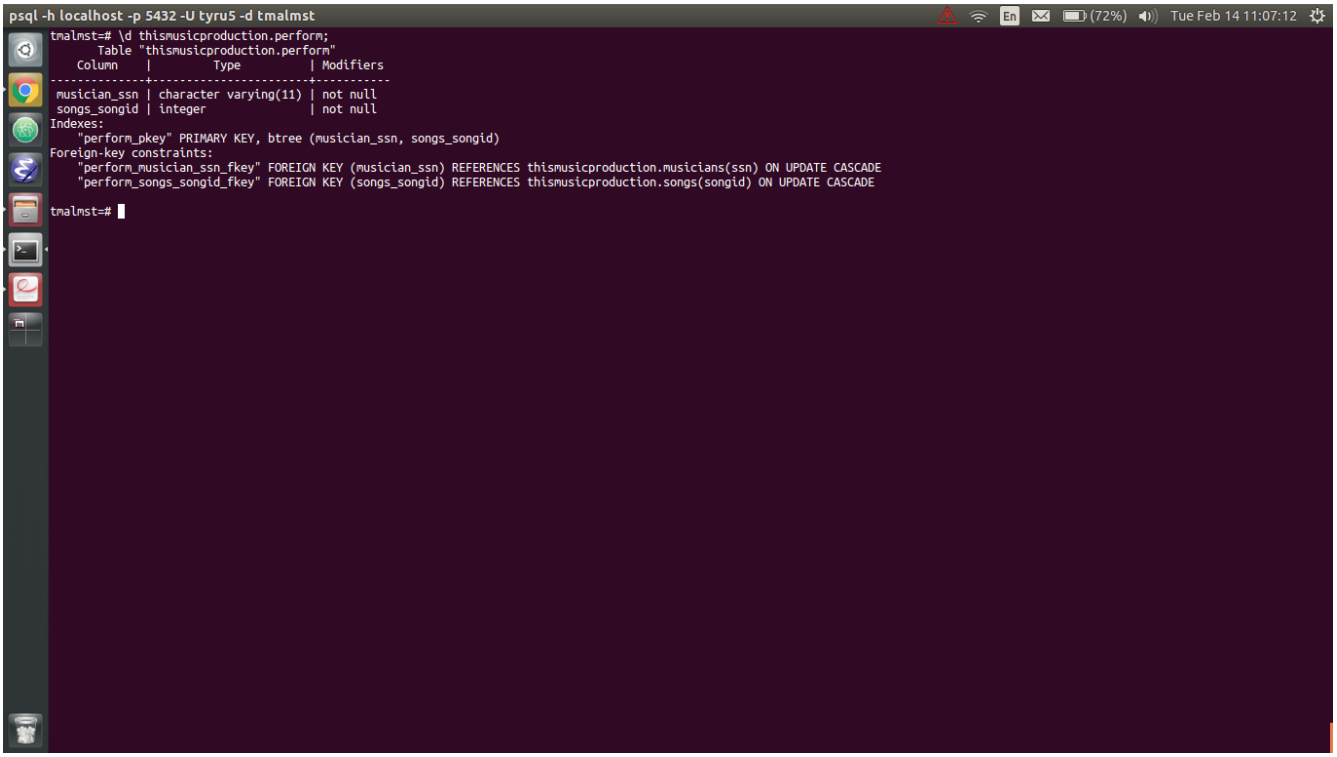

psql -h localhost -p 5432 -U tyru5 -d tmalmst
(73%) Tue Feb 14 11:05:42
tmalmst=# select * from thismusicproduction.instrument;
instrid | dname | key
1 | Violin | C#
2 | Piano | A Minor
3 | Guitar | G
4 | Electronic DAW | All keys
5 | Harmonica | A Major
(5 rows)
tmalmst=#

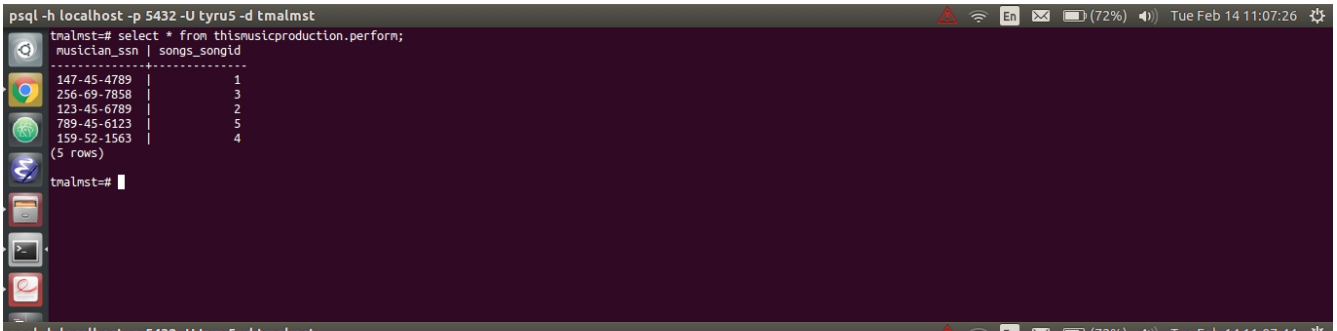

psql -h localhost -p 5432 -U tyru5 -d tmalmst
Tue Feb 14 11:06:00
tmalmst=# \d thismusicproduction.lives;
Table "thismusicproduction.lives"
Column | Type | Modifiers
musician_ssn | character varying(11) | not null
place_address | character varying(45) | not null
Indexes:
"lives_pkey" PRIMARY KEY, btree (musician_ssn, place_address)
Foreign-key constraints:
"lives_musician_ssn_fkey" FOREIGN KEY (musician_ssn) REFERENCES thismusicproduction.musicians(ssn) ON UPDATE CASCADE
"lives_place_address_fkey" FOREIGN KEY (place_address) REFERENCES thismusicproduction.home(place_address) ON UPDATE CASCADE
tmalmst=#

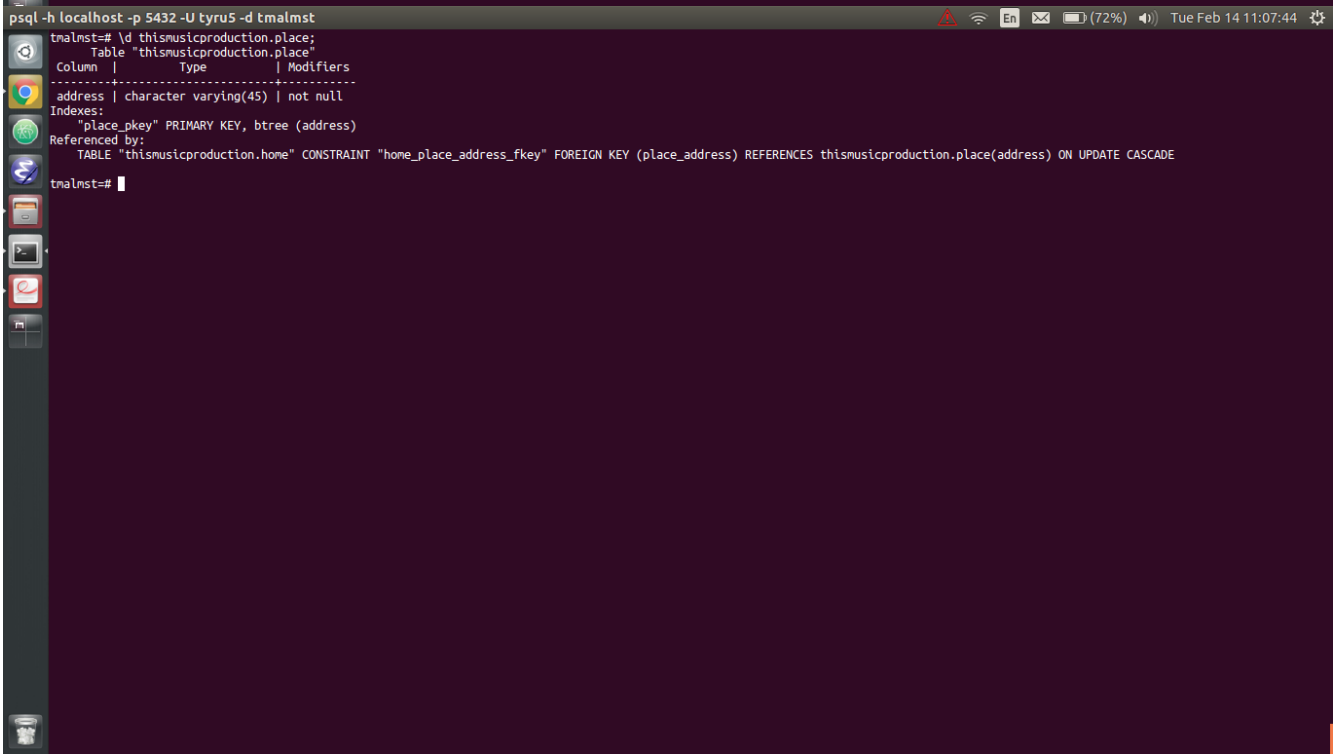

psql -h localhost -p 5432 -U tyru5 -d tmalmst
Tue Feb 14 11:06:17
tmalmst=# select * from thismusicproduction.lives;
musician_ssn | place_address
147-45-4789 | 2250 W. South Drive
256-69-7858 | 1337 Leet Drive
123-45-6789 | 5148. 7 Panagioti Gyanapolou
789-45-6123 | 7331 A. This address is cool
159-52-1563 | 3140. Ontario drive
(5 rows)
tmalmst=#

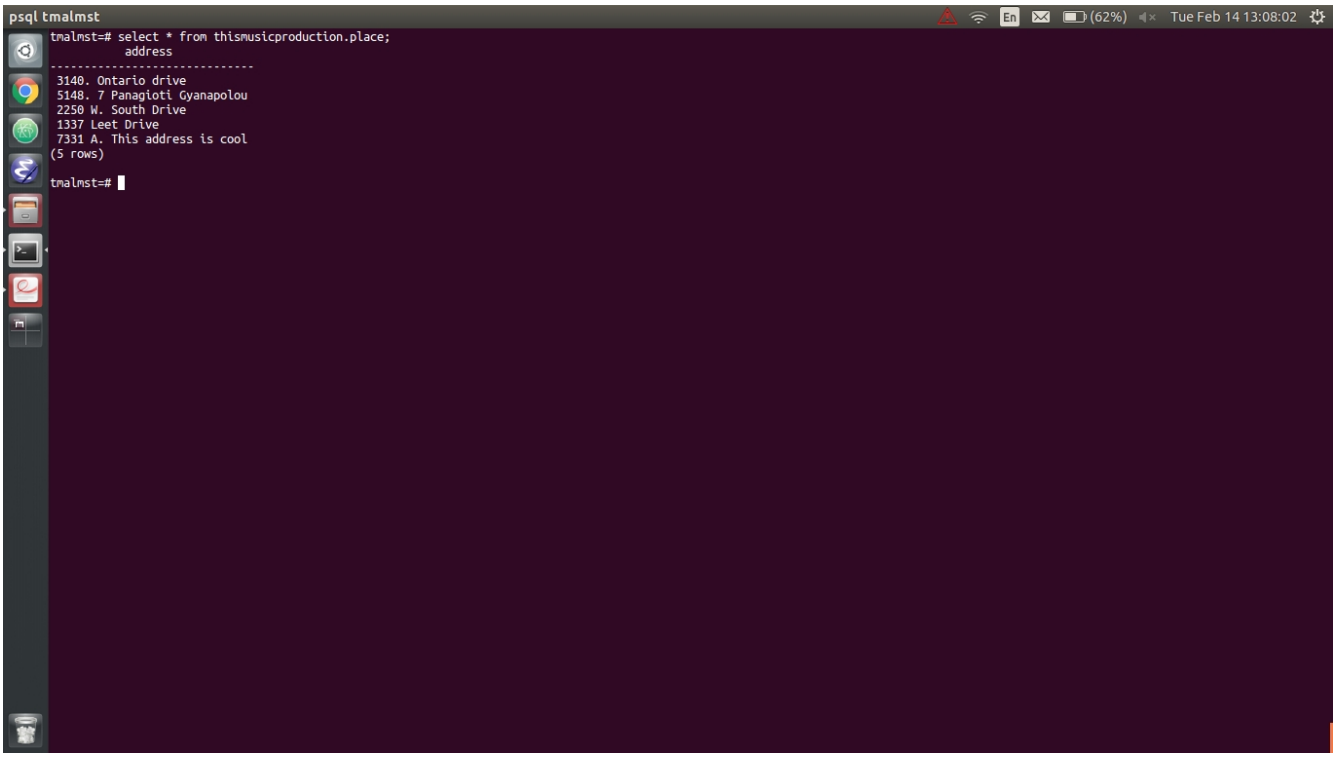

psql -h localhost -p 5432 -U tyru5 -d tmalmst
Tue Feb 14 11:06:36
tmalmst=# \d thismusicproduction.musicians;
Table "thismusicproduction.musicians"
Column | Type | Modifiers
ssn | character varying(11) | not null
name | character varying(15) |
Indexes:
"musicians_pkey" PRIMARY KEY, btree (ssn)
Referenced by:
TABLE "thismusicproduction.album" CONSTRAINT "album_musician_ssn_fkey" FOREIGN KEY (musician_ssn) REFERENCES thismusicproduction.musicians(ssn) ON DELETE CASCADE
TABLE "thismusicproduction.lives" CONSTRAINT "lives_musician_ssn_fkey" FOREIGN KEY (musician_ssn) REFERENCES thismusicproduction.musicians(ssn) ON UPDATE CASCADE
TABLE "thismusicproduction.perform" CONSTRAINT "perform_musician_ssn_fkey" FOREIGN KEY (musician_ssn) REFERENCES thismusicproduction.musicians(ssn) ON UPDATE CASCADE
TABLE "thismusicproduction.plays" CONSTRAINT "plays_musician_ssn_fkey" FOREIGN KEY (musician_ssn) REFERENCES thismusicproduction.musicians(ssn) ON UPDATE CASCADE
tmalmst=#

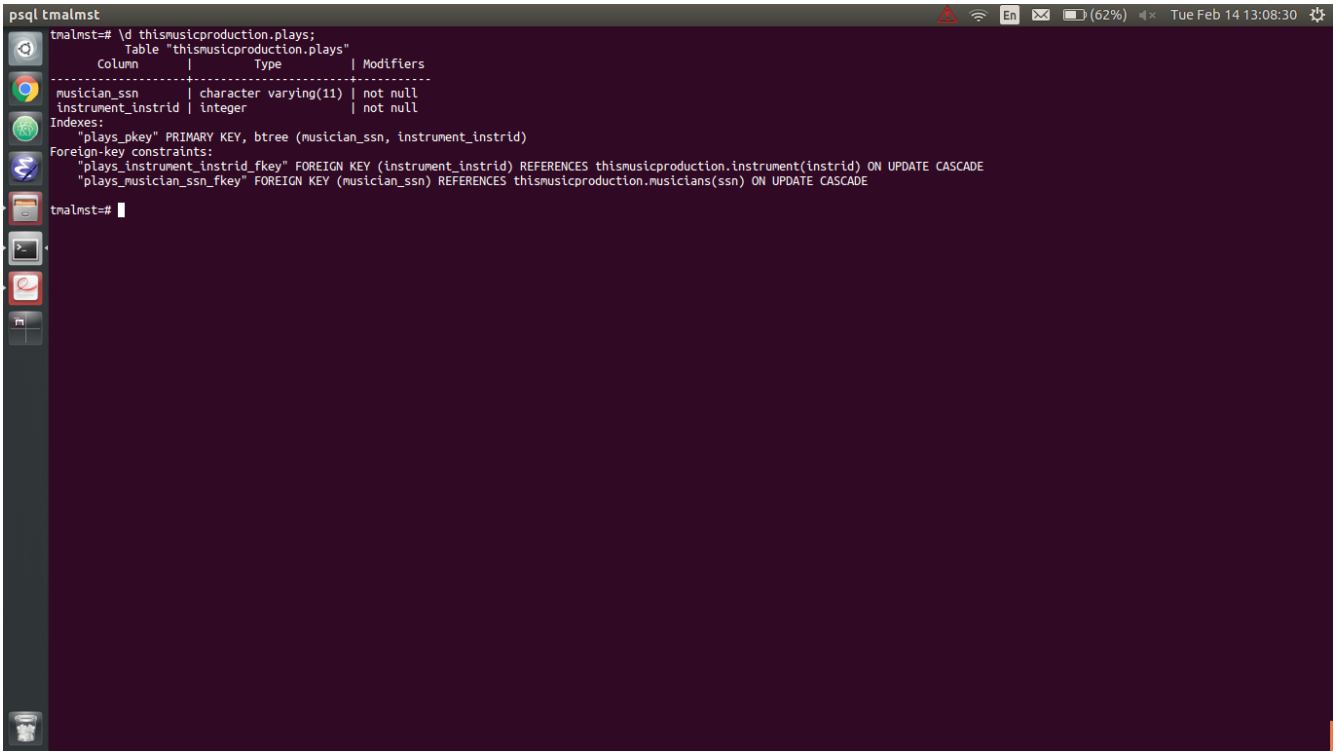

psql -h localhost -p 5432 -U tyru5 -d tmalmst
Tue Feb 14 11:06:53
tmalmst=# select * from thismusicproduction.musicians;
ssn | name
147-45-4789 | John
256-69-7858 | Smith
123-45-6789 | Tyrus
789-45-6123 | Bob
159-52-1563 | deadmau5
(5 rows)
tmalmst=#

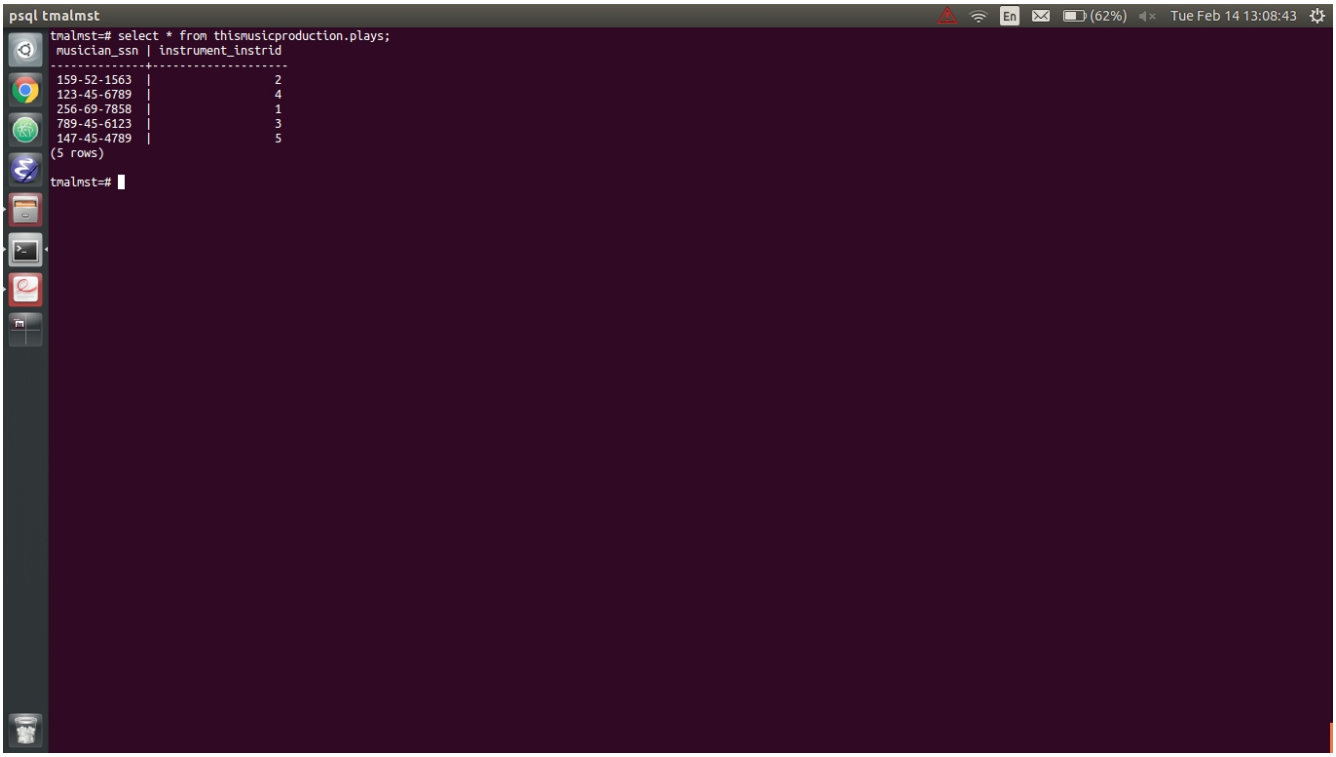

psql -h localhost -p 5432 -U tyru5 -d tmalmst
Tue Feb 14 11:07:12
tmalmst=# \d thismusicproduction.perform;
Table "thismusicproduction.perform"
Column | Type | Modifiers
musician_ssn | character varying(11) | not null
songs_songid | integer | not null
Indexes:
"perform_pkey" PRIMARY KEY, btree (musician_ssn, songs_songid)
Foreign-key constraints:
"perform_musician_ssn_fkey" FOREIGN KEY (musician_ssn) REFERENCES thismusicproduction.musicians(ssn) ON UPDATE CASCADE
"perform_songs_songid_fkey" FOREIGN KEY (songs_songid) REFERENCES thismusicproduction.songs(songid) ON UPDATE CASCADE
tmalmst=#

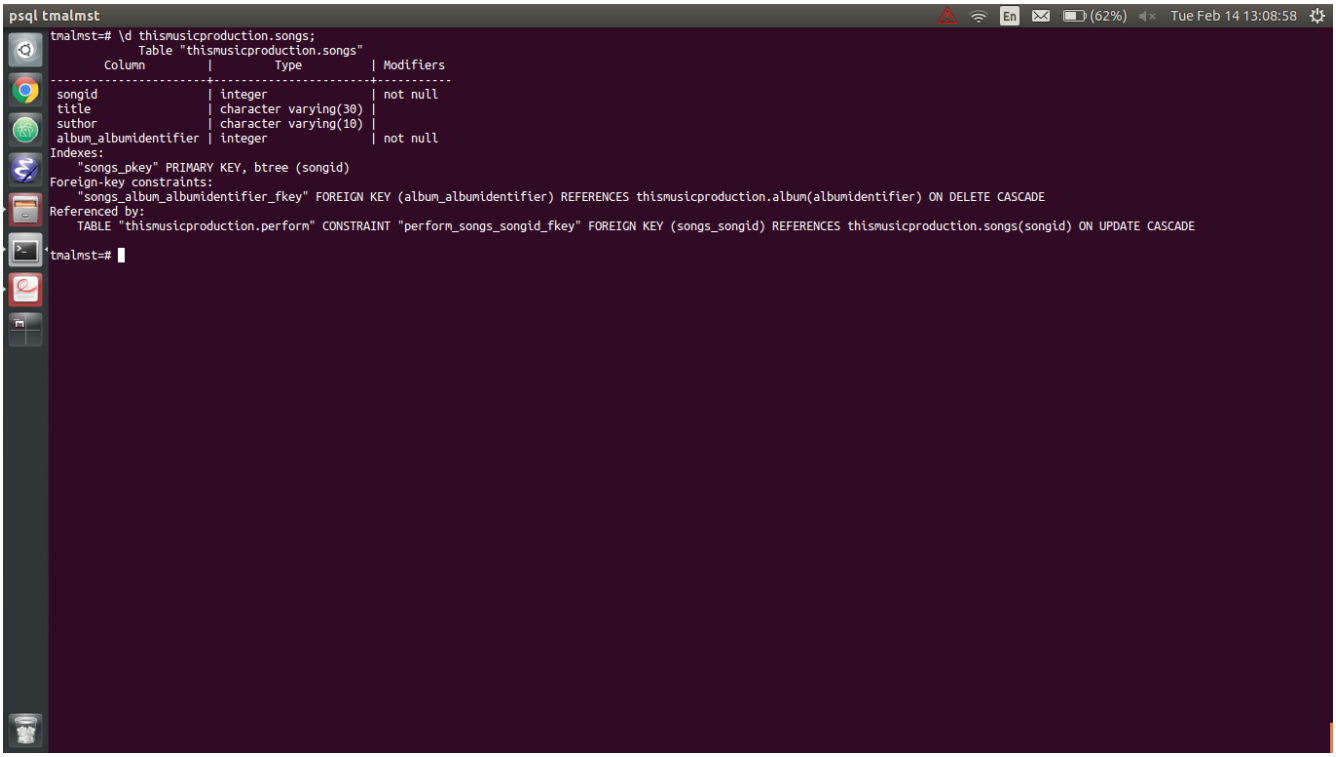

psql -h localhost -p 5432 -U tyru5 -d tmalmst
Tue Feb 14 11:07:26
tmalmst=# select * from thismusicproduction.perform;
musician_ssn | songs_songid
147-45-4789 | 1
256-69-7858 | 3
123-45-6789 | 2
789-45-6123 | 5
159-52-1563 | 4
(5 rows)
tmalmst=#

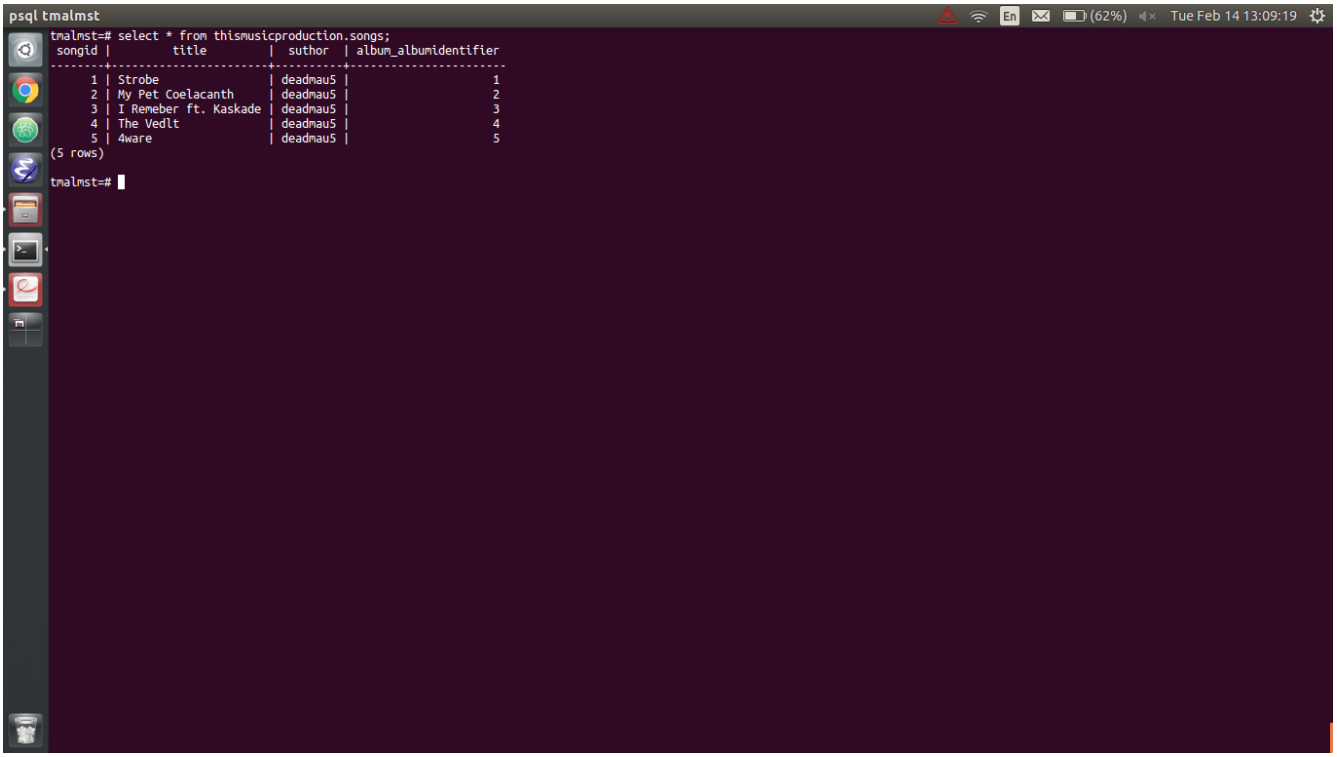

psql -h localhost -p 5432 -U tyru5 -d tmalmst
Tue Feb 14 11:07:44
tmalmst=# \d thismusicproduction.place;
Table "thismusicproduction.place"
Column | Type | Modifiers
address | character varying(45) | not null
Indexes:
"place_pkey" PRIMARY KEY, btree (address)
Referenced by:
TABLE "thismusicproduction.home" CONSTRAINT "home_place_address_fkey" FOREIGN KEY (place_address) REFERENCES thismusicproduction.place(address) ON UPDATE CASCADE
tmalmst=#

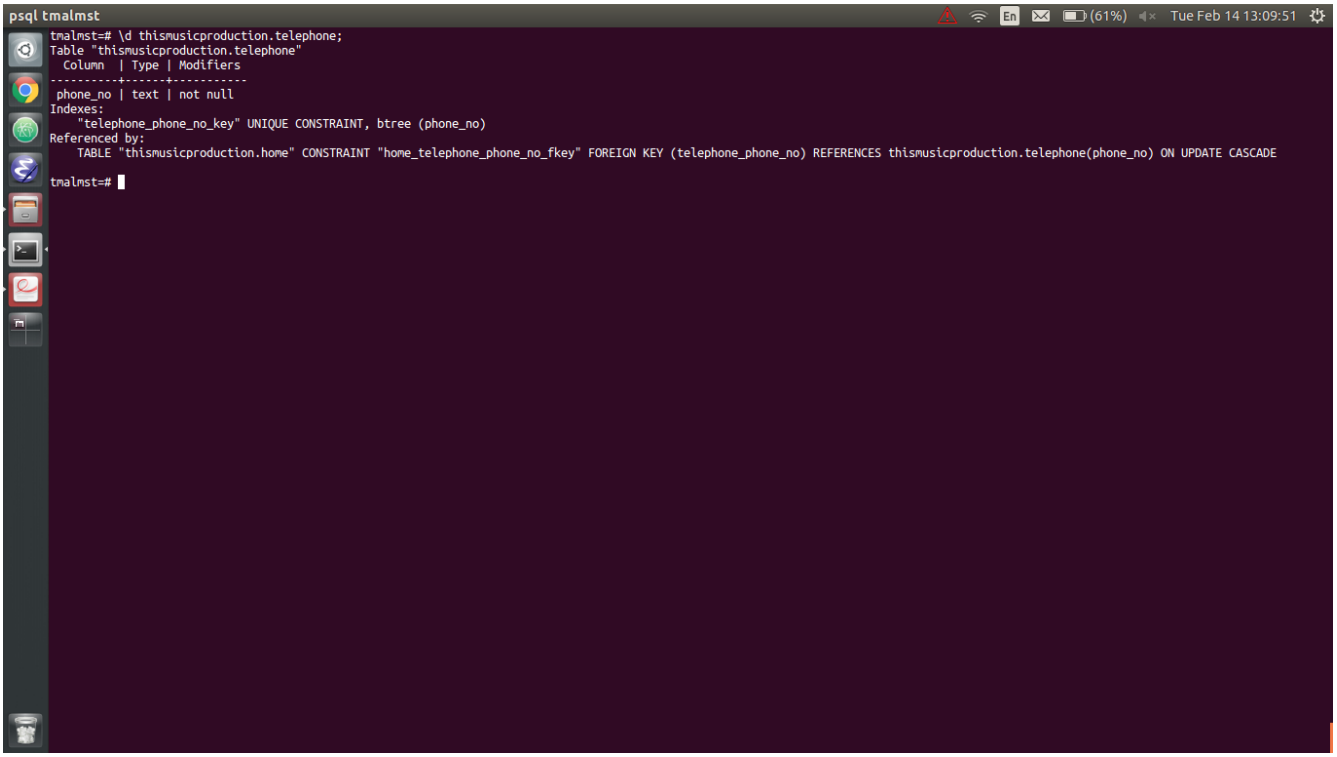

psql tmalmst
(62%) Tue Feb 14 13:08:02
tmalmst=# select * from thismusicproduction.place;
address
3140. Ontario drive
5148. 7 Panagioti Gyanapolou
2250 W. South Drive
1337 Leet Drive
7331 A. This address is cool
(5 rows)
tmalmst=#


psql tmalmst
(62%) Tue Feb 14 13:08:30
tmalmst=# \d thismusicproduction.plays;
Table "thismusicproduction.plays"
Column | Type | Modifiers
musician_ssn | character varying(11) | not null
instrument_instrid | integer | not null
Indexes:
"plays_pkey" PRIMARY KEY, btree (musician_ssn, instrument_instrid)
Foreign-key constraints:
"plays_instrument_instrid_fkey" FOREIGN KEY (instrument_instrid) REFERENCES thismusicproduction.instrument(instrid) ON UPDATE CASCADE
"plays_musician_ssn_fkey" FOREIGN KEY (musician_ssn) REFERENCES thismusicproduction.musicians(ssn) ON UPDATE CASCADE
tmalmst=#

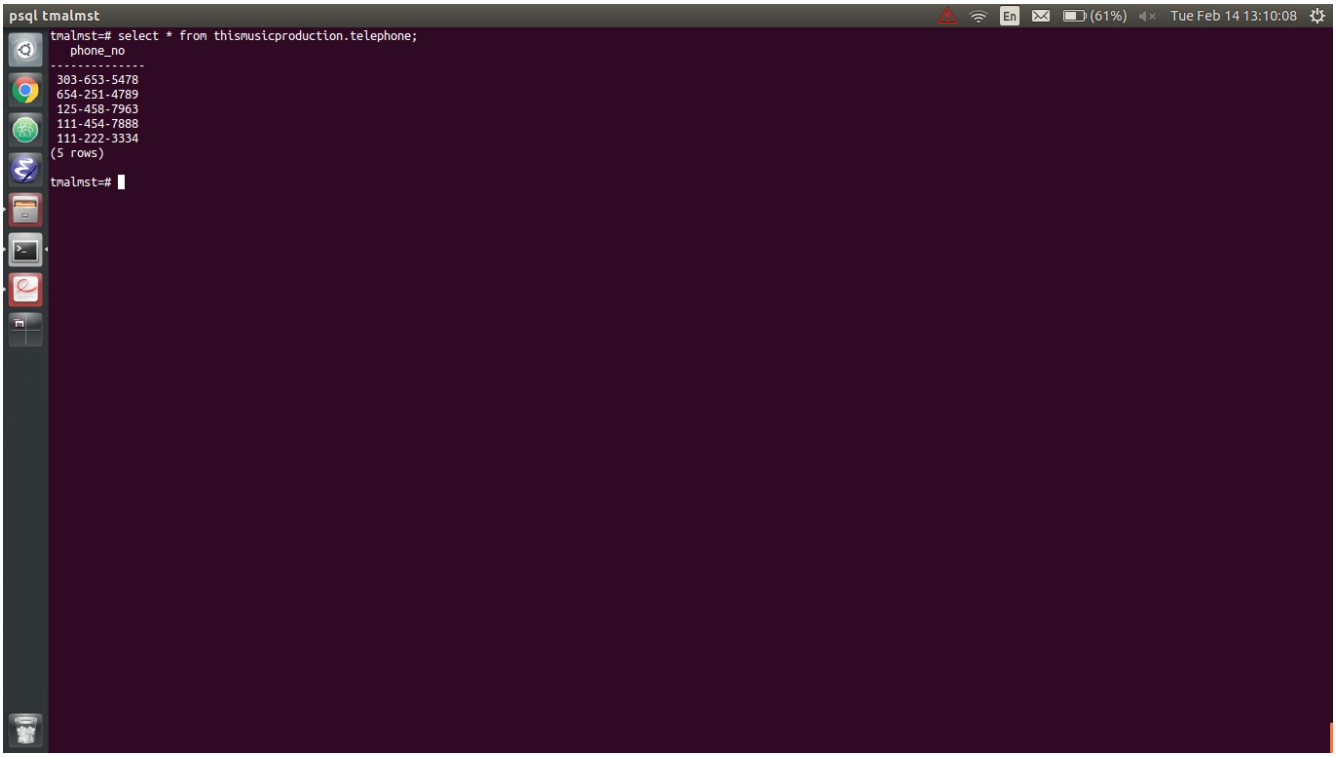

psql tmalmst
(62%) Tue Feb 14 13:08:43
tmalmst=# select * from thismusicproduction.plays;
musician_ssn | instrument_instrid
159-52-1563 | 2
123-45-6789 | 4
256-69-7858 | 1
789-45-6123 | 3
147-45-4789 | 5
(5 rows)
tmalmst=#


psql tmalmst
(62%) Tue Feb 14 13:08:58
tmalmst=# \d thismusicproduction.songs;
Table "thismusicproduction.songs"
Column | Type | Modifiers
songid | integer | not null
title | character varying(30) |
suthor | character varying(10) |
album_albumidentifier | integer | not null
Indexes:
"songs_pkey" PRIMARY KEY, btree (songid)
Foreign-key constraints:
"songs_album_albumidentifier_fkey" FOREIGN KEY (album_albumidentifier) REFERENCES thismusicproduction.album(albumidentifier) ON DELETE CASCADE
Referenced by:
TABLE "thismusicproduction.perform" CONSTRAINT "perform_songs_songid_fkey" FOREIGN KEY (songs_songid) REFERENCES thismusicproduction.songs(songid) ON UPDATE CASCADE
tmalmst=#


psql tmalmst
(62%) Tue Feb 14 13:09:19
tmalmst=# select * from thismusicproduction.songs;
songid | title | suthor | album_albumidentifier
1 | Strobe | deadmau5 | 1
2 | My Pet Coelacanth | deadmau5 | 2
3 | I Remeber ft. Kaskade | deadmau5 | 3
4 | The Vedlt | deadmau5 | 4
5 | 4ware | deadmau5 | 5
(5 rows)
tmalmst=#


psql tmalmst
(61%) Tue Feb 14 13:09:51
tmalmst=# \d thismusicproduction.telephone;
Table "thismusicproduction.telephone"
Column | Type | Modifiers
phone_no | text | not null
Indexes:
"telephone_phone_no_key" UNIQUE CONSTRAINT, btree (phone_no)
Referenced by:
TABLE "thismusicproduction.home" CONSTRAINT "home_telephone_phone_no_fkey" FOREIGN KEY (telephone_phone_no) REFERENCES thismusicproduction.telephone(phone_no) ON UPDATE CASCADE
tmalmst=#

Database Management Systems Homework 2 submission Tyrus Malmstrom :: 829947336



Those are all my tables, including all entities, attributes, relationships, and primary, foreign key constraints stated within the assignment description.

Best,
Tyrus Malmstrom.