**Introduction**

## Entertainment Virginia is a small business that is located in Northern Virginia which presents several plays for the community throughout the year. Due to the success of the company over the past 5 years, there has been a great demand for more performances. The success has also attracted several new sponsors to fund the performances. To accommodate this high demand of performances, Entertainment Virginia decided to create a database to store, retrieve, analyze, and manage their data on plays, performers, directors, and sponsors. The purpose of this project is to help Entertainment Virginia create a database management information system to better manage the company’s large amount of data. In order to create the database, the first step was to gather important data such as performers full name and contact information, directors full name and contact information, play name, type, and length, sponsor’s company name, type of business, and contact information. This large amount of data was provided to us by Entertainment Virginia. By using the data provided to us, we were able to create distinct tables that contained crucial information for the performers, sponsors, directors, and plays using SQL statements. Once the data was entered in to the database, a test run was performed to ensure data integrity and to make sure that the database ran smoothly. After the test run was performed, the finished database design was presented to the president for review and to make sure the new database fulfilled and satisfied the need and requirements of entertainment Virginia.

## We had two parts to implement the relevant SQL statements. In the first part, we used all SQL statements to create and populate the designed database. In the second part, we used relevant SQL statements to reproduce sample reports 1, 2, 3, and 4.

1st part: (see MYSQL table 1 - 12)

1. To create the database container called ‘entertainment’, we followed the statement-

create database entertainment;

2. We checked that the database was created by using the MySQL show command to list all databases.

show databases;

3. To access (use) the entertainment database, we followed the MySQL statement -

use entertainment;

4. Then we created the six tables director, play, performer, sponsor, playperformer and playsponsor (the data type for all fields is varchar except sponsorid, which is smallint) by following these MYSQL statements -

create table director

(directorid varchar(4) primary key,

directorfn varchar(8),

directorln varchar(12),

directorphone varchar(12),

directoremail varchar(20));

create table play

(playid varchar(4) primary key,

playname varchar(15),

playlength varchar(7),

playtype varchar(7),

directorid varchar(4) not null);

create table performer

(performerid varchar(4) primary key,

performerfn varchar(7),

performerln varchar(10),

performerphone varchar(12),

performeremail varchar(20));

create table sponsor

(sponsorid smallint(4) unsigned not null,

companyname varchar(15),

sponsorfn varchar(5),

sponsorln varchar(6),

sponsorphone varchar(12),

typeofbusiness varchar(15)

Primary key(sponsorid));

create table playperformer

(playid varchar(4) not null,

performerid varchar(4) not null);

create table playsponsor

(playid varchar(4) not null,

sponsorid smallint(4) unsigned not null);

5. After creating all the tables, we checked that the tables were created by using the MySQL show command to list all tables.

show tables;

6. Then we checked that the tables were created as planned using MySQL desc statements. It showed metadata for each table like the names of the fields, data types, primary key field.

desc director;

desc play;

desc performer;

desc sponsor;

desc playperformer;

desc playsponsor;

7. We Inserted data into the appropriate tables by following this statement

insert into director values

("D100", "William", "Shakespear", "800-555-1678", "shakes@directs.com"),

("D200", "Theodore", "Limited", "800-555-0000", "theo@directs.com"),

("D300", "Sheila", "Patterson", "800-555-1414", "sheila@directs.com"),

("D400", "Julie", "Wonderful", "800-555-8888", "julie@directs.com"),

("D500", "Benjamin", "Pretend", "800-555-7890", "[pretend@directs.com](mailto:pretend@directs.com)");

insert into play values

("P111", "Hello Dolly", "2 hours", "Musical", "D100"),

("P222", "The King and I", "2 hours", "Musical", "D100"),

("P333", "Serious Skits", "1 hour", "Drama", "D200"),

("P444", "Happy Hour", "1 hour", "Comedy", "D300"),

("P555", "Happy Days", "3 hours", "Drama", "D100"),

("P666", "On a Carousel", "2 hours", "Comedy", "D300"),

("P777", "Rent", "2 hours", "Musical", "D400"),

("P888", "West Side Story", "2 hours", "Musical", "D500"),

("P999", "Home Run!", "1 hour", "Drama", "D500");

insert into performer values

("A101", "Beth", "Activision", "703-555-1888", "beth@beth.com"),

("A102", "Mark", "Activision", "703-555-1888", "mark@mark.com"),

("A103", "Seth", "Adams", "703-555-1288", "seth@seth.com"),

("A104", "Peter", "Zimmers", "703-555-3434", "peter@peter.com"),

("A105", "Susan", "Smith", "703-555-4343", "sue@sue.com"),

("A106", "Susan", "peters", "703-555-5454", "susan@susan.com"),

("A107", "William", "Henry", "703-555-5555", " william@william.com"),

("A108", "Lou", "Smith", "703-555-0001", "lou@lou.com"),

("A109", "Pat", "Bender", "703-555-9000", "pat@pat.com"),

("A110", "Sally", "Fence", "703-555-9030", "sally@sally.com"),

("A111", "John", "Jacobs", "703-555-9031", "john@john.com"),

("A112", "Adam", "Quinn", "703-555-8991", "adam@adam.com");

insert into sponsor values

(1020, "Best Eats Here", "Sean", "Bud", "703-555-2222", "Fast Food"),

(1040, "Burgers R us", "Bill", "Rogers", "703-555-1212", "Fast Food"),

(1060, "Business Buy", "Joe", "Dream", "703-555-3456", "Electronics"),

(1070, "Coasters Haven", "Mary", "Clark", "703-555-3456", "Amusement Park"),

(1010, "Gadgets Inc.", "Joe", "Smith", "703-555-1234", "Electronics"),

(1050, "The Tax Store", "Kim", "Lee", "703-555-8888", "Tax Services"),

(1030, "Water Fun", "Helen", "Smith", "703-555-3333", "Amusement Park");

insert into playperformer values

("P555", "A101"),

("P555", "A102"),

("P555", "A103"),

("P555", "A104"),

("P555", "A105"),

("P555", "A106"),

("P555", "A107"),

("P555", "A108"),

("P555", "A109"),

("P444", "A103"),

("P444", "A104"),

("P444", "A105"),

("P111", "A101"),

("P111", "A102"),

("P111", "A106"),

("P111", "A107"),

("P111", "A105"),

("P111", "A111"),

("P111", "A112"),

("P999", "A110"),

("P999", "A111"),

("P999", "A112"),

("P666", "A103"),

("P666", "A104"),

("P666", "A107"),

("P666", "A110"),

("P666", "A111"),

("P666", "A112"),

("P666", "A101"),

("P666", "A102"),

("P666", "A104"),

("P666", "A107"),

("P666", "A112"),

("P333", "A110"),

("P333", "A111"),

("P333", "A112"),

("P222", "A103"),

("P222", "A104"),

("P222", "A105"),

("P222", "A106"),

("P222", "A107"),

("P222", "A109"),

("P888", "A107"),

("P888", "A105"),

("P888", "A111"),

("P888", "A112");

insert into playsponsor values

("P555", 1040),

("P555", 1060),

("P444", 1050),

("P444", 1070),

("P444", 1010),

("P111", 1050),

("P111", 1030),

("P111", 1010),

("P999", 1070),

("P999", 1010),

("P999", 1030),

("P999", 1050),

("P999", 1020),

("P666", 1040),

("P666", 1050),

("P666", 1040),

("P333", 1040),

("P333", 1070),

("P333", 1060),

("P333", 1030),

("P222", 1060),

("P222", 1020),

("P222", 1070),

("P888", 1010),

("P888", 1070);

8. To ensure that the data was entered correctly, we used MySQL select statement to list all of the data in each of the tables.

select \* from director;

select \* from play;

select \* from performer;

select \* from sponsor;

select \* from playperformer;

select \* from playsponsor;

2nd part: (see MYSQL table 13- 16)

To reproduce the sample reports 1, 2, 3, and 4, we followed 4 SQL statements.

Sample Report -1

To display the performer”s last name , first name and phone number of all performers in the database sorted by performer name, we followed the statement -

Select performerln, performerfn, performerphone from performer

order by performerln;

Sample Report - 2

To display companyname, first name, last name, phone number, we followed the below SQL statement -

Select companyname, sponsorfn, sponsorln, sponsorphone

from sponsor

order by company name;

Sample Report - 3

To reproduce sample report 3, we did inner join. We joined the director table and the play table on the directorid field in common. We sorted it by director last name. Therefore, our SQL statement was -

Select d. directorln, d.directorfn, p. playname, p. playtype

from director d, play p

where d. directorid = p. directorid

order by directorln;

Sample Report - 4

To reproduce sample report 4, we joined tables with intersection. We joined 3 tables which were play, sponsor and playsponsor. Table play and playsponsor had a common field that was playid and table sponsor had a common field with table playsponsor which was sponsorid. We sorted everything by play name. So, we used following SQL statement -

select p. playname, p. playtype, s. companyname, s. typeofbusiness

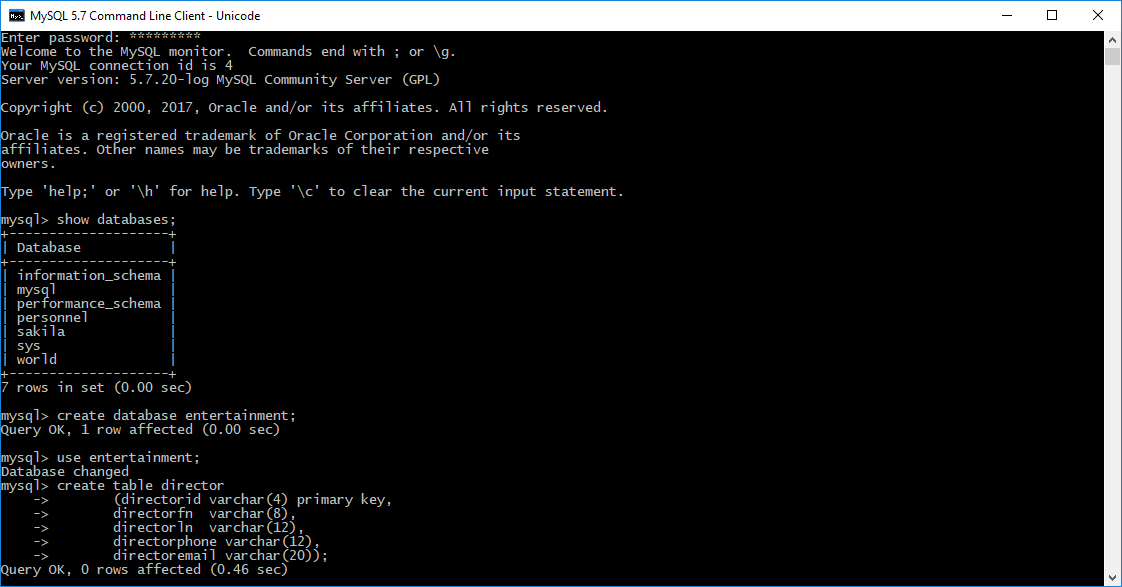
from play p, sponsor s, playsponsor ps

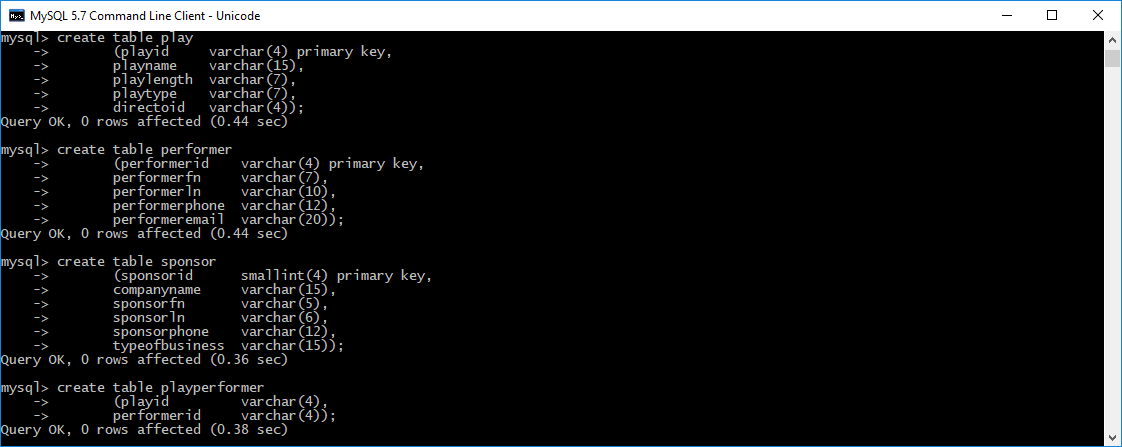
where p. playid = ps playid and s. sponsorid = ps. Sponsorid

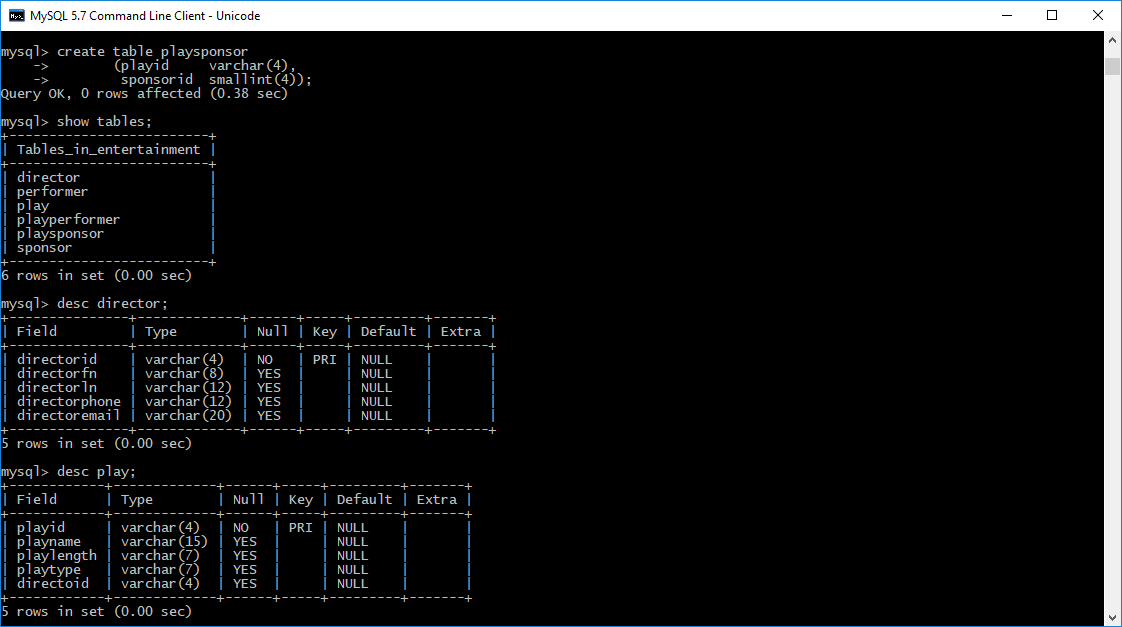
order by playname;

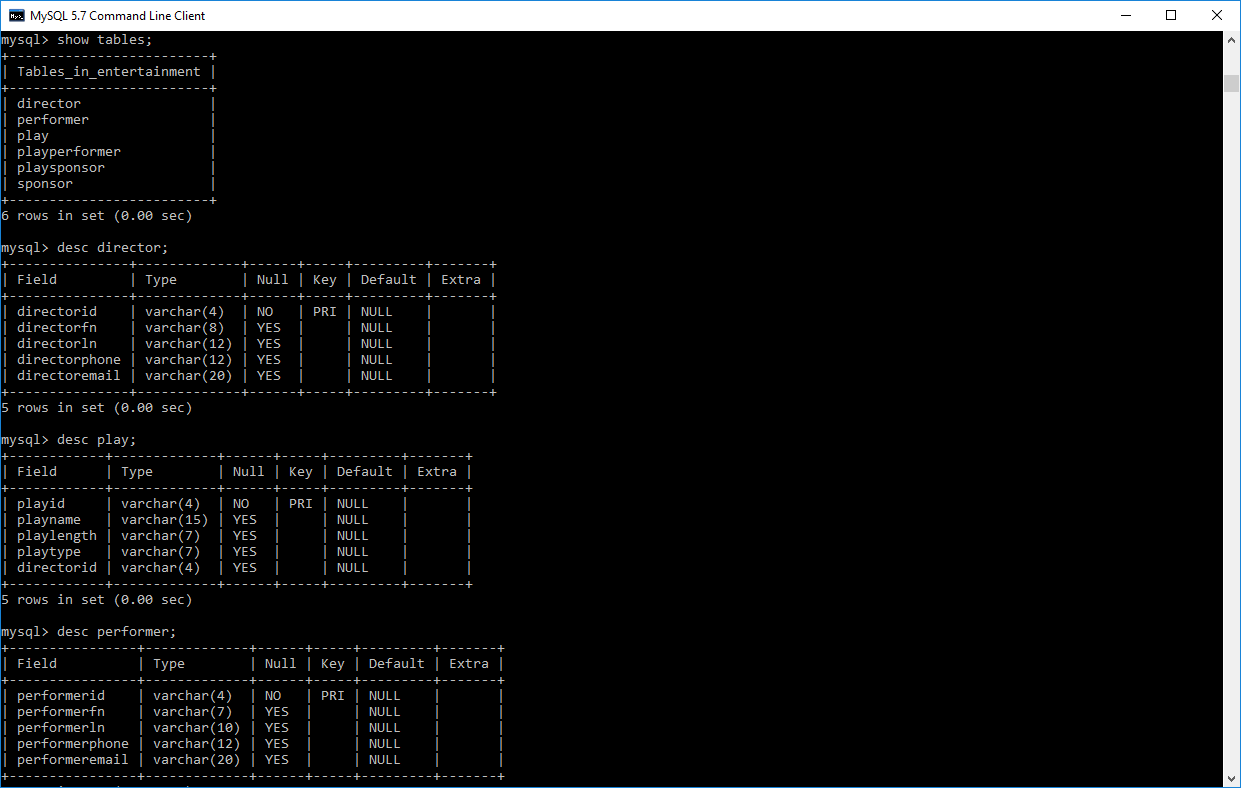
Screenshot of our MYSQL

MYSQL table -1

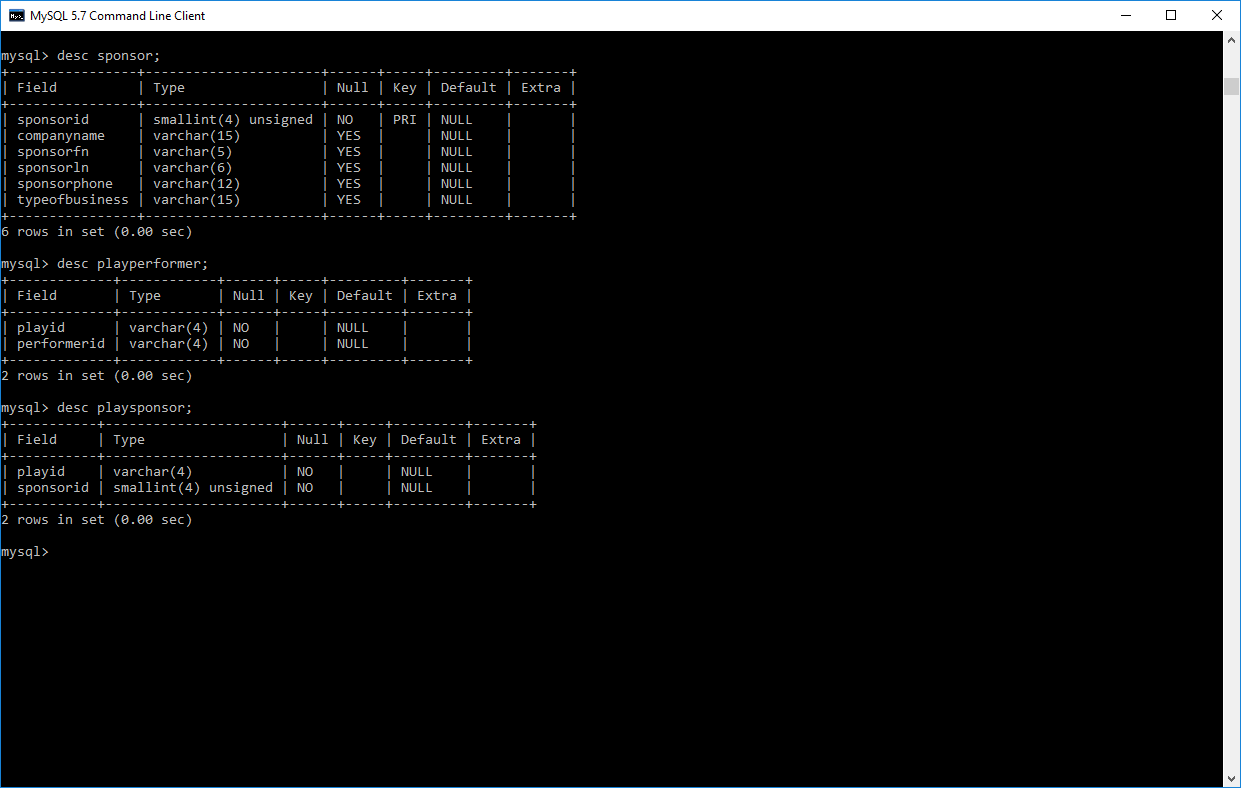




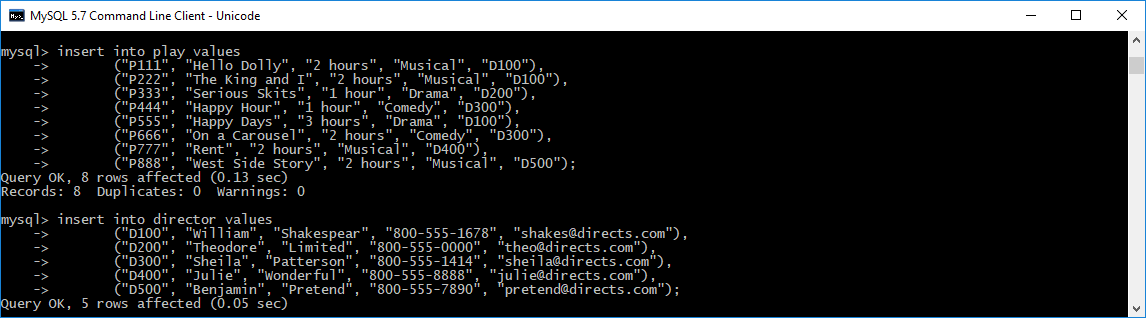




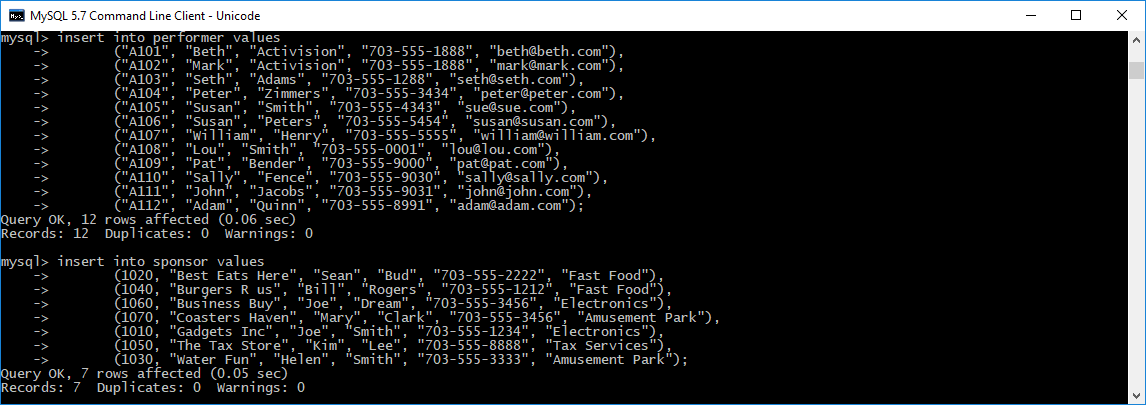
MYSQL table - 2



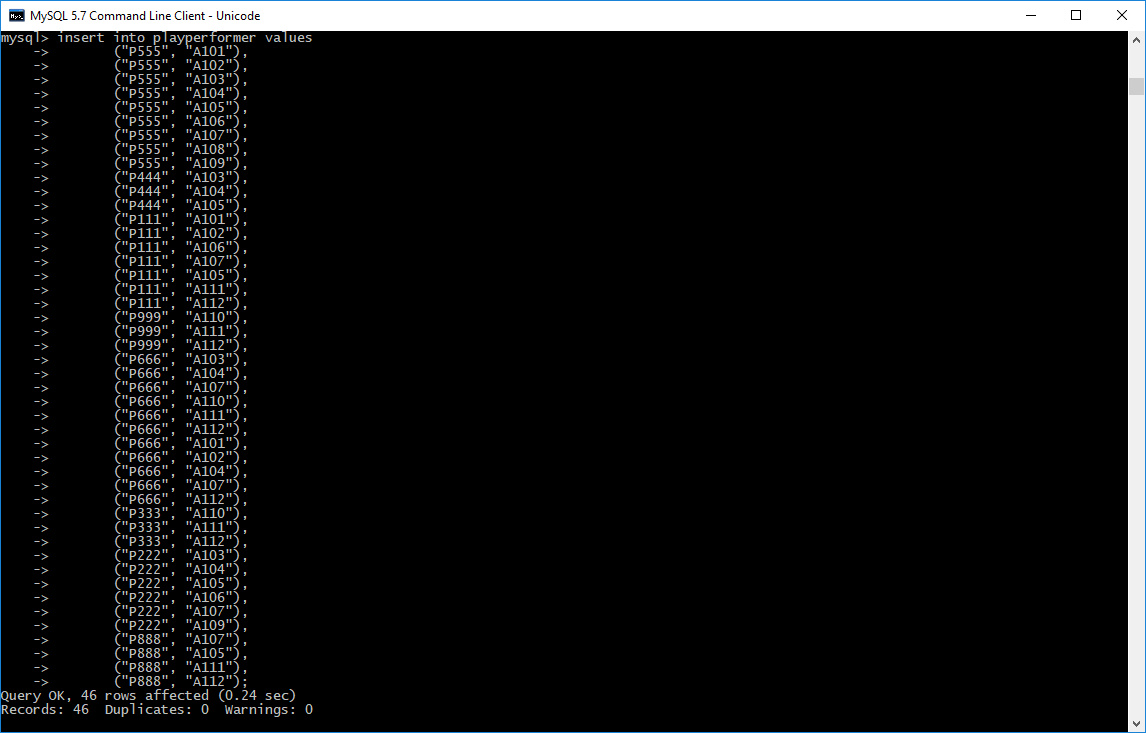
MYSQL table - 3



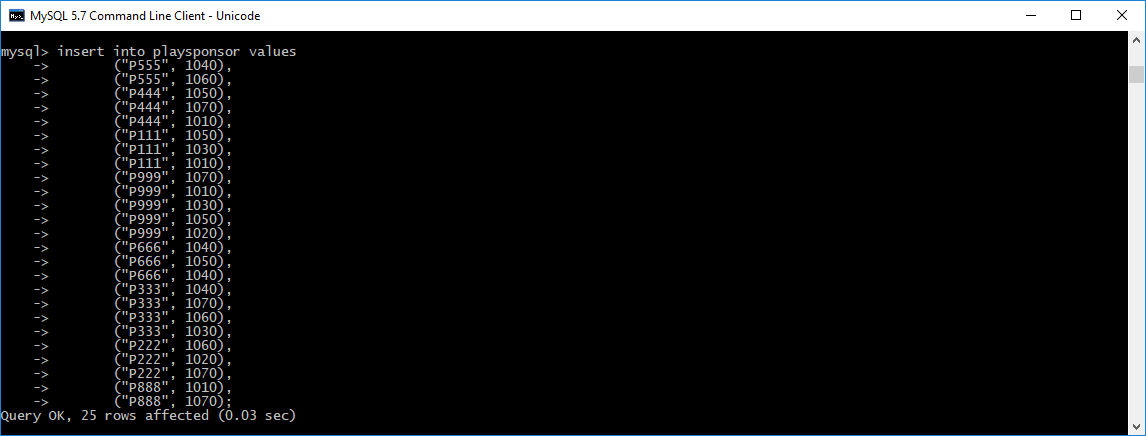
MYSQL table - 4



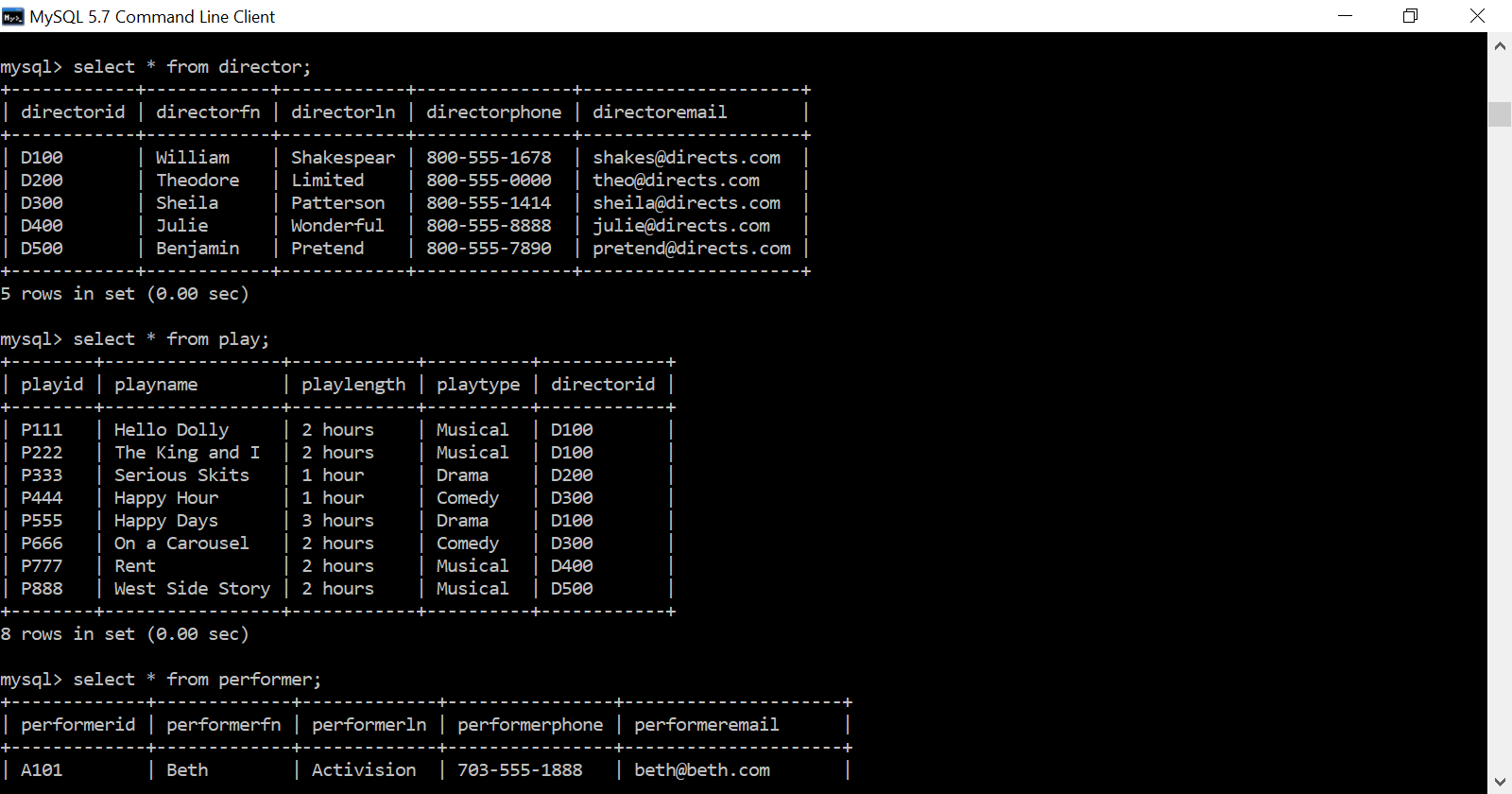
MYSQL table - 5



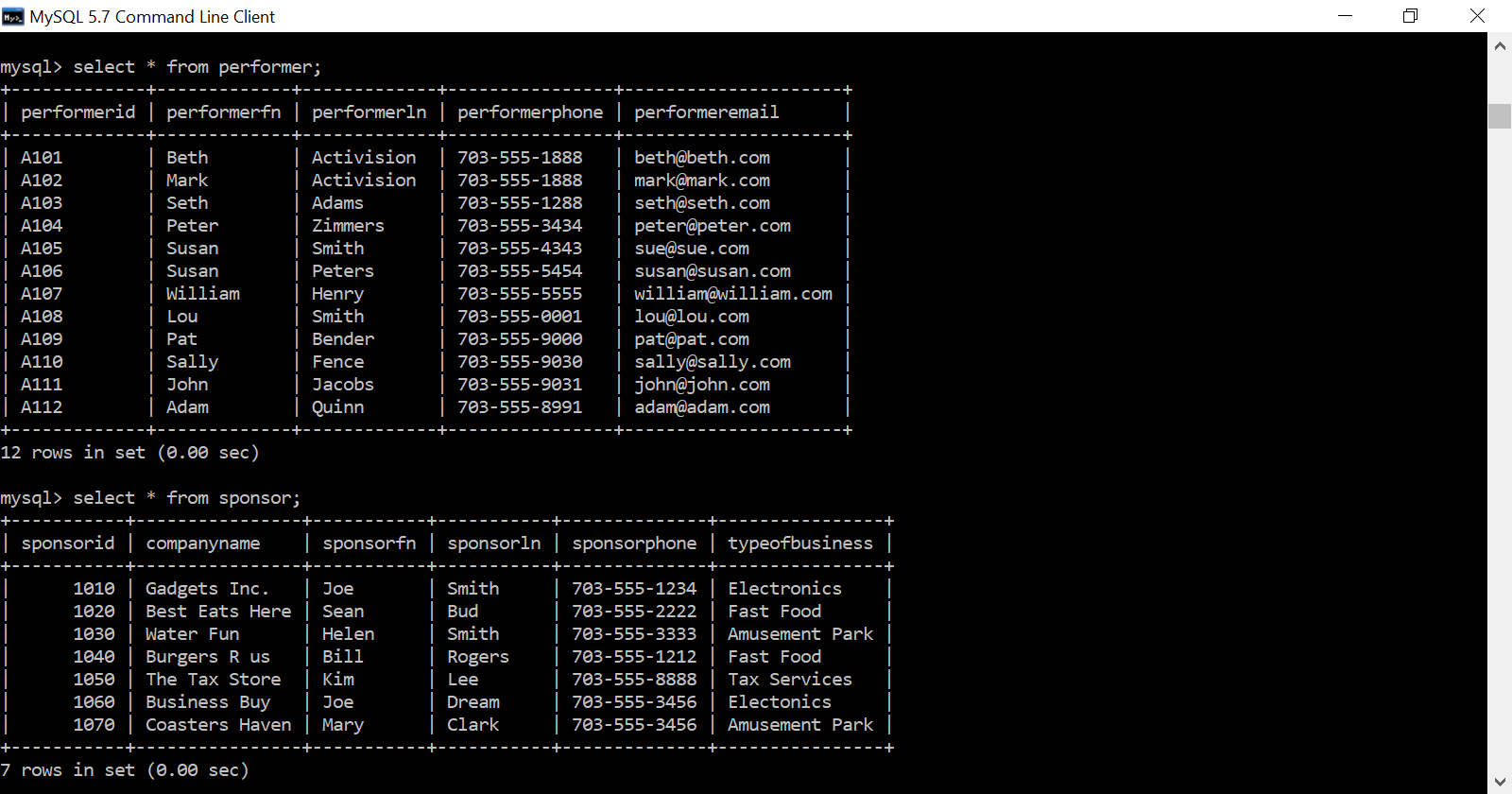
MYSQL table - 6



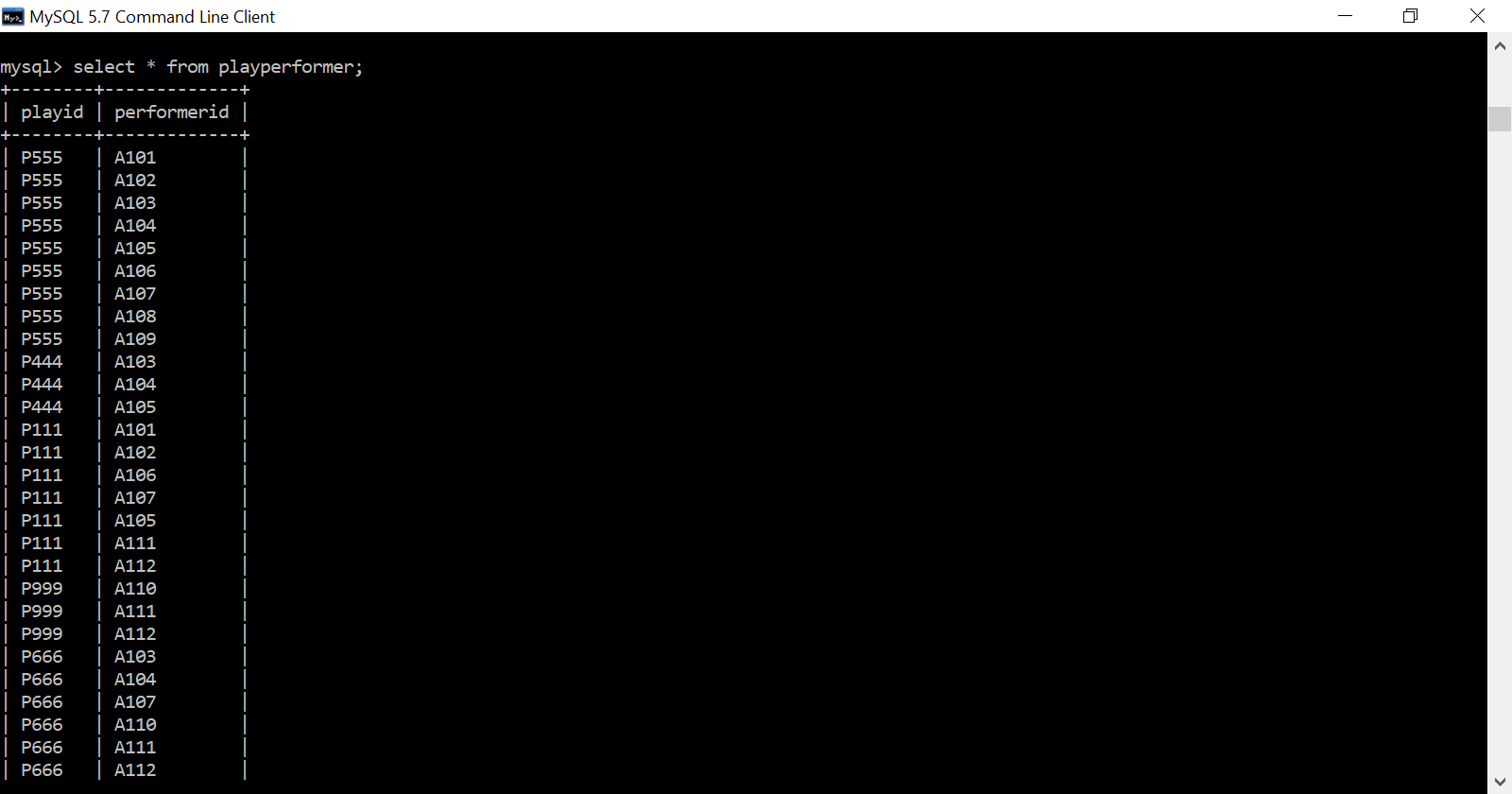
MYSQL table - 7



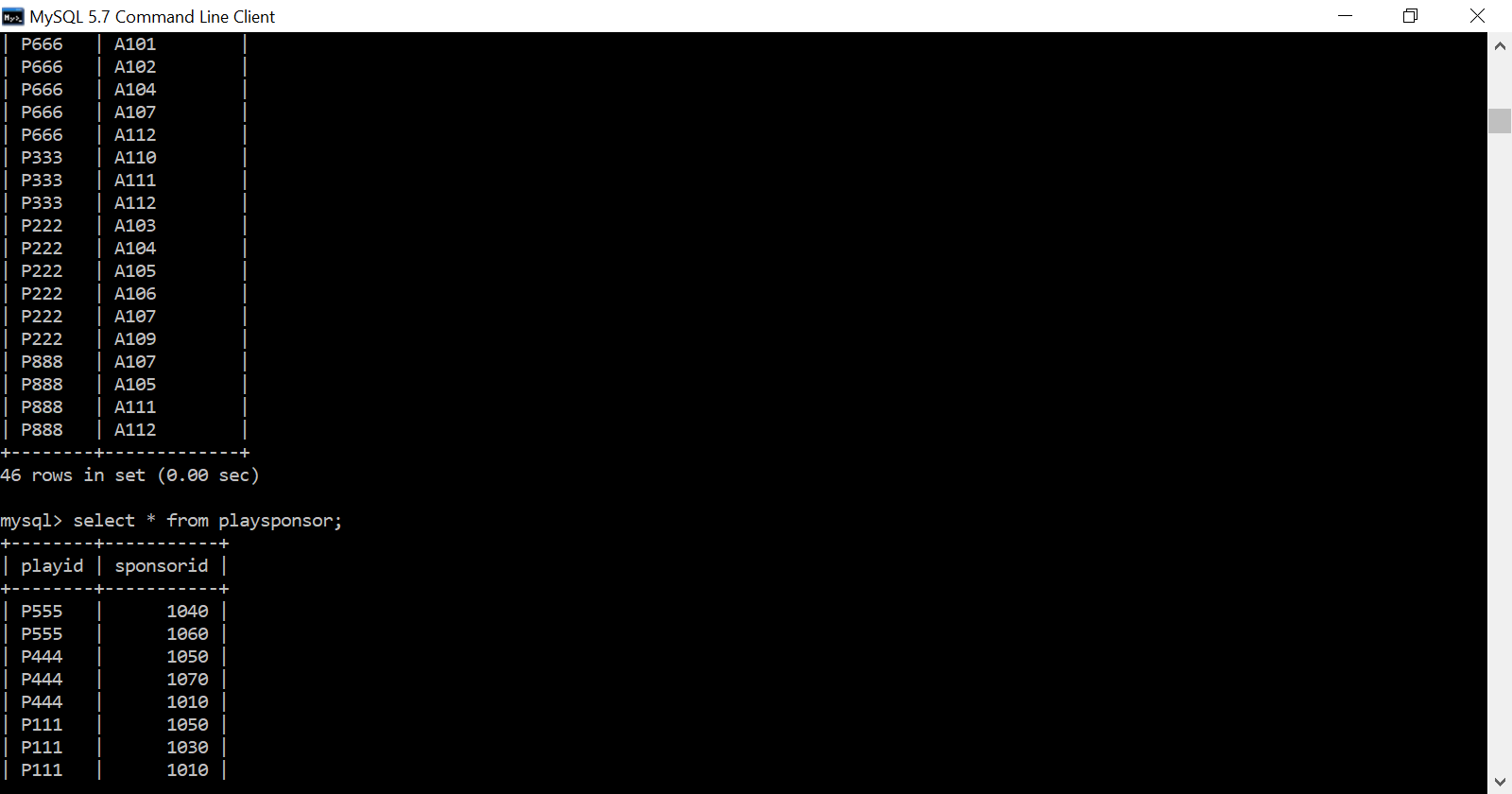
MYSQL table - 8



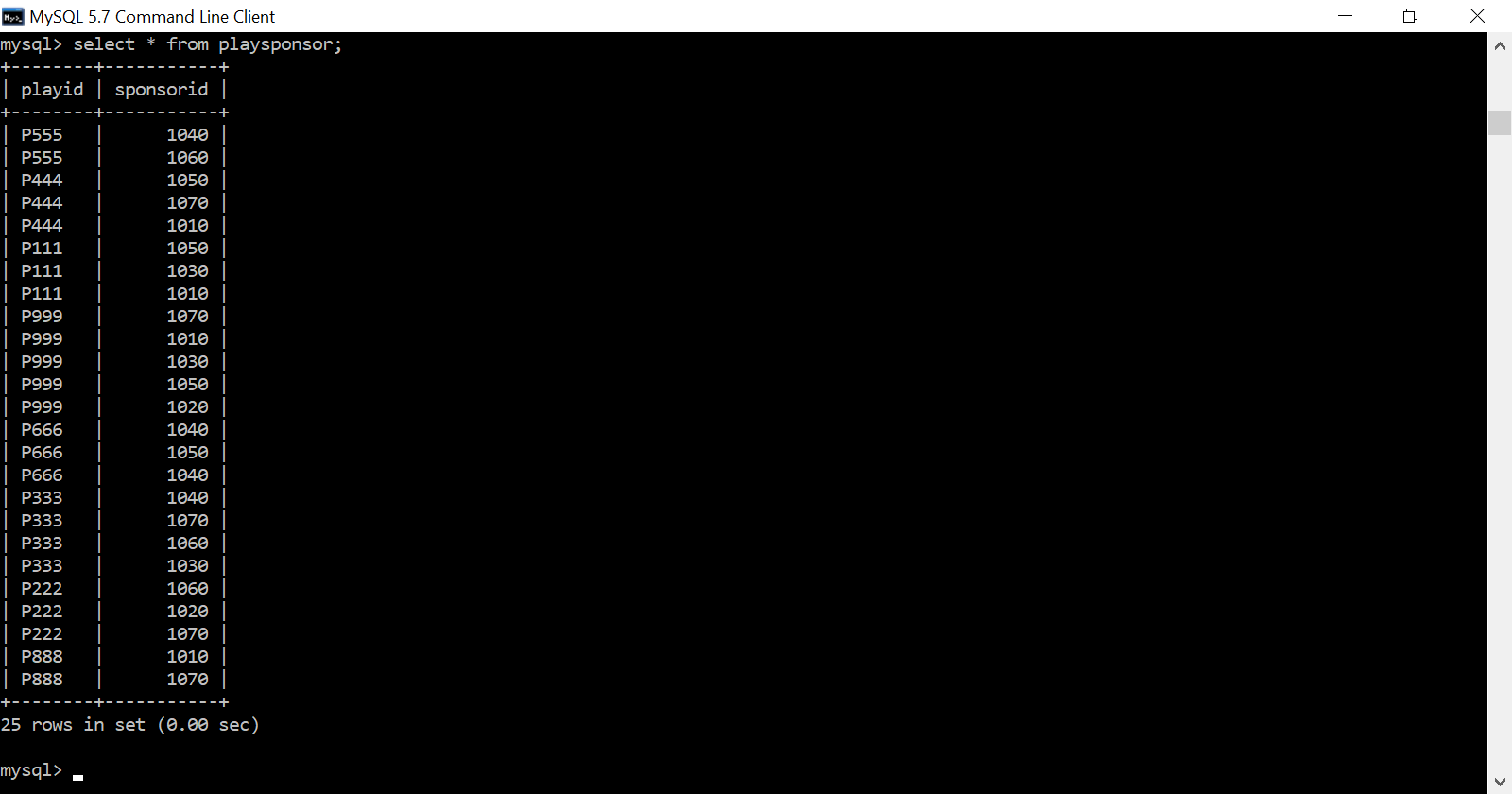
MYSQL table - 9



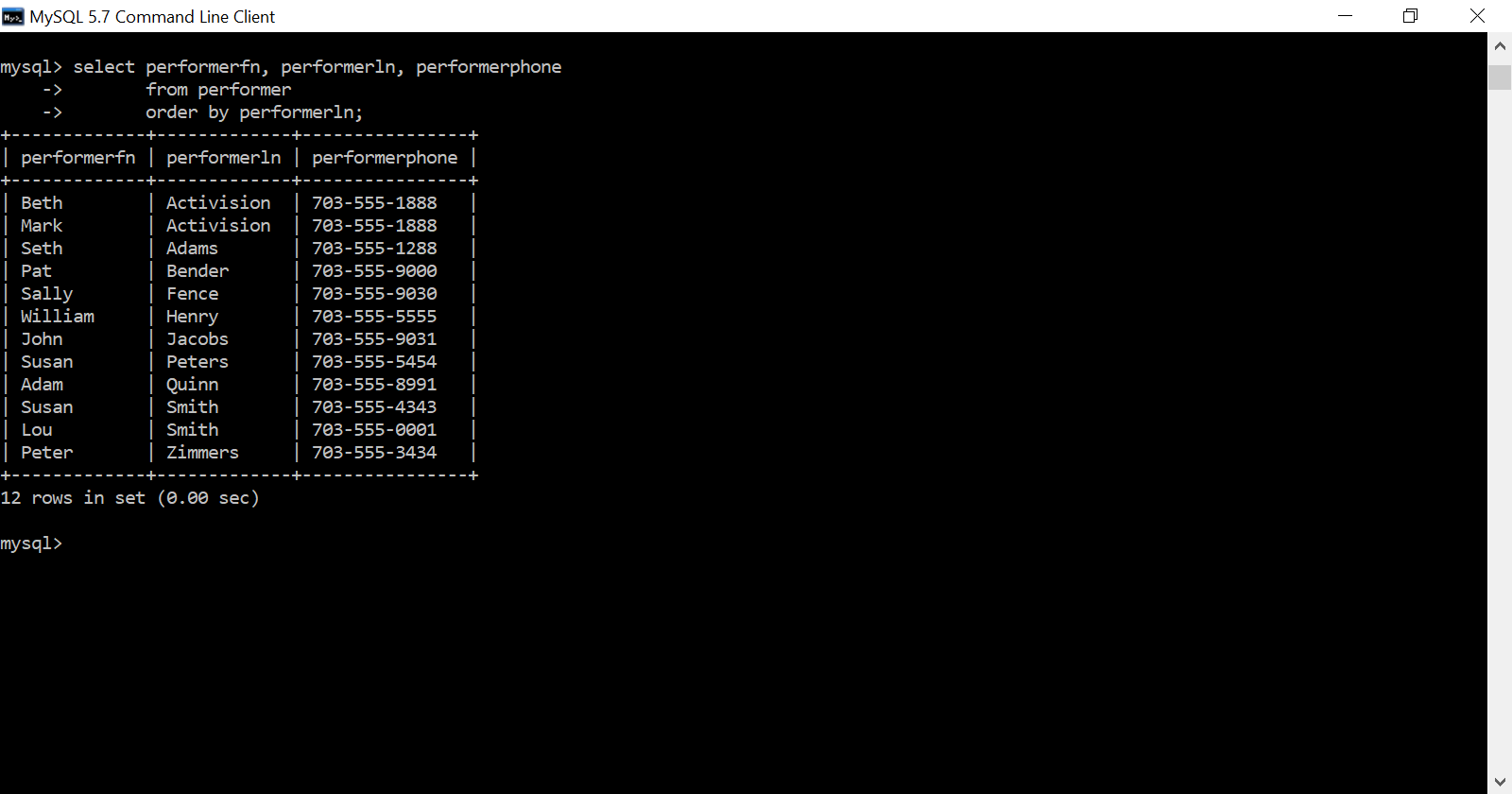
MYSQL table - 10



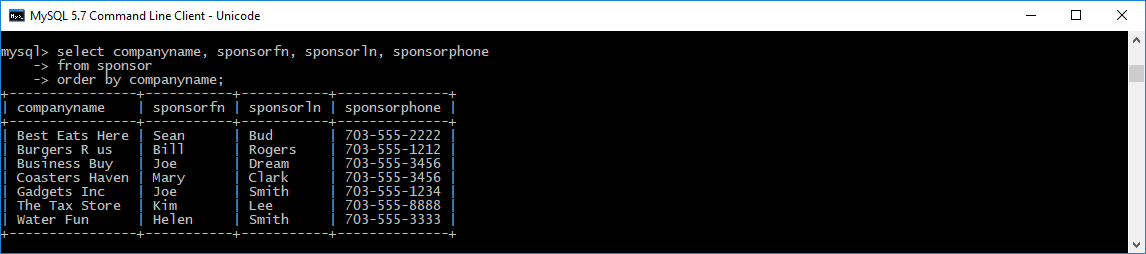
MYSQL table - 11



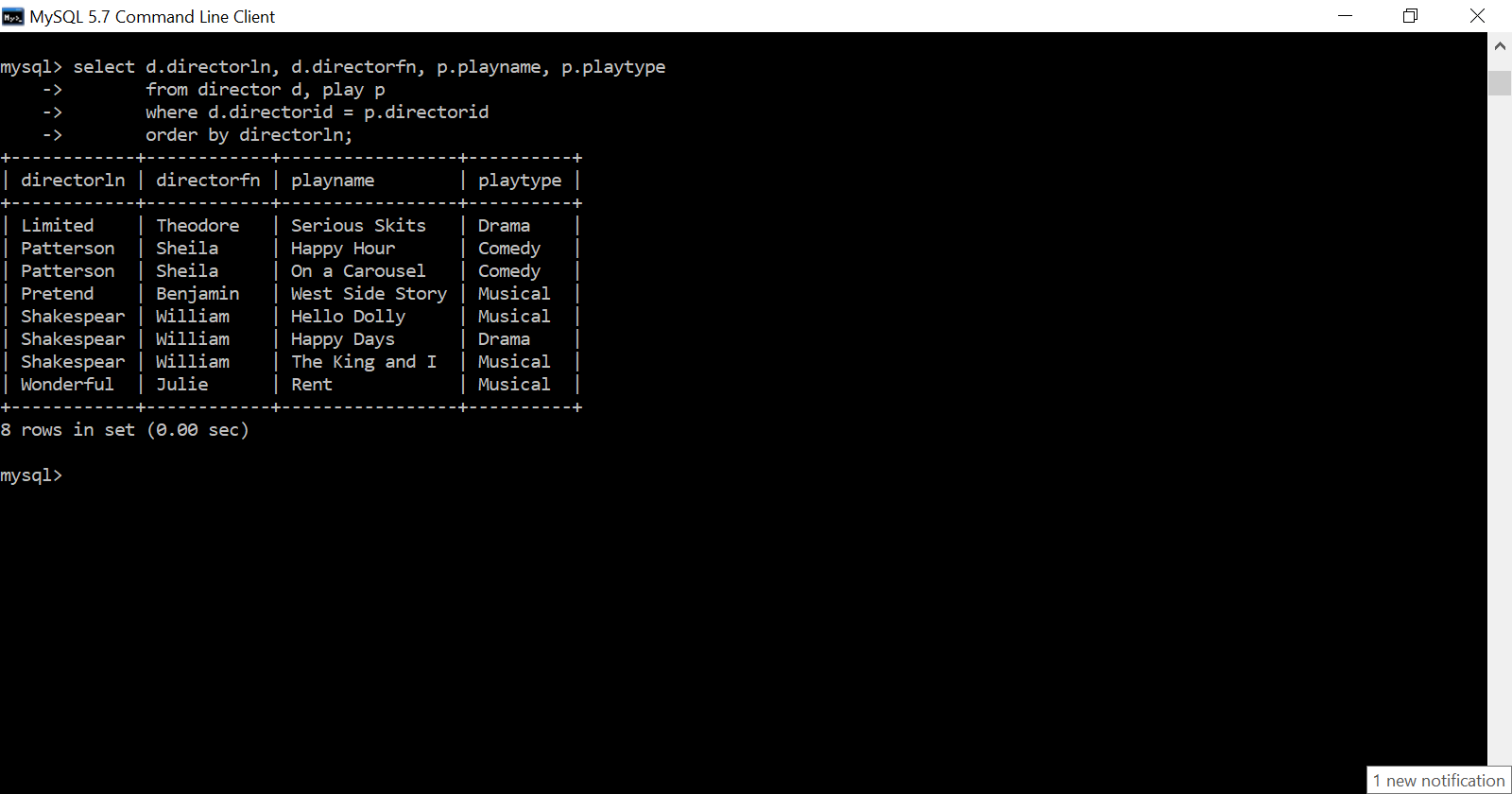
MYSQL table - 12



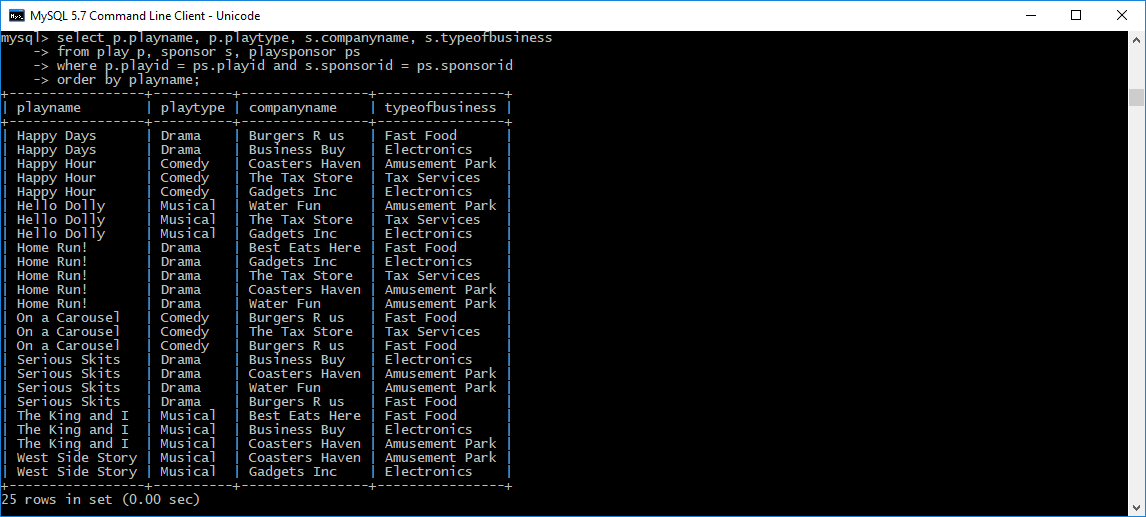
MYSQL table - 13

****

MYSQL table - 14



MYSQL table - 15



MYSQL table - 16