Database Design and Implementation

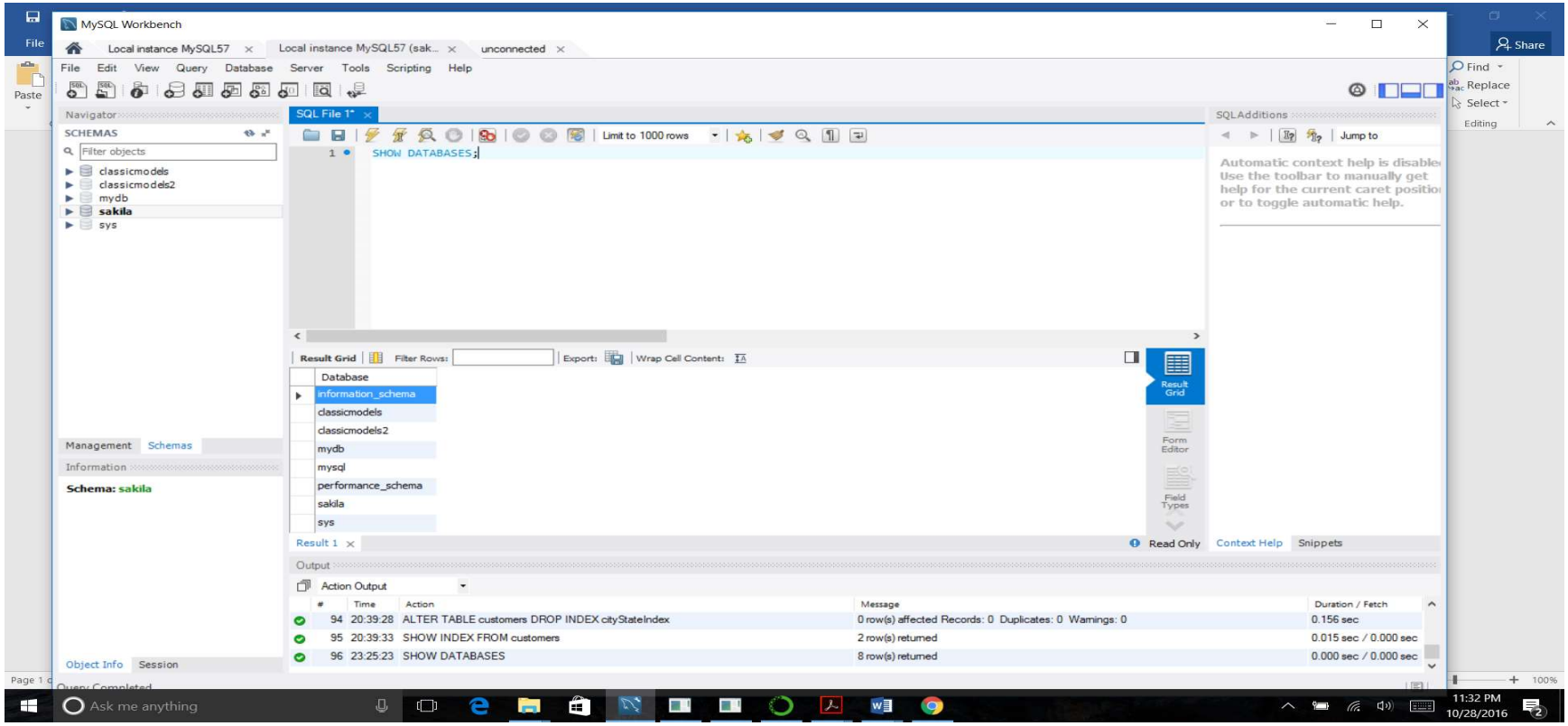
MSCA 31005

Manivassakam M

Assignment 4

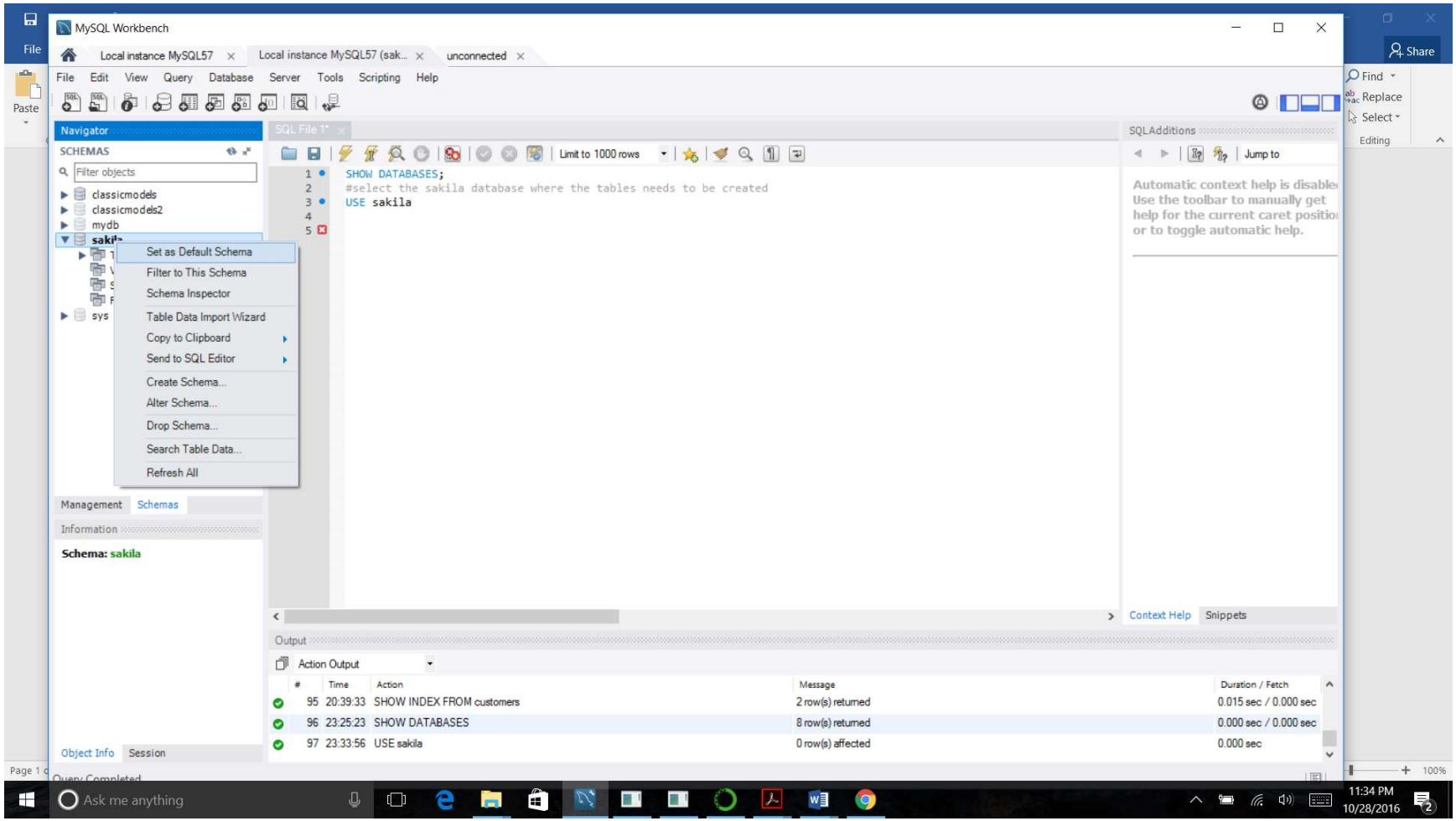
QUESTION 1

 # a) Show the list of databases.
SHOW DATABASES;



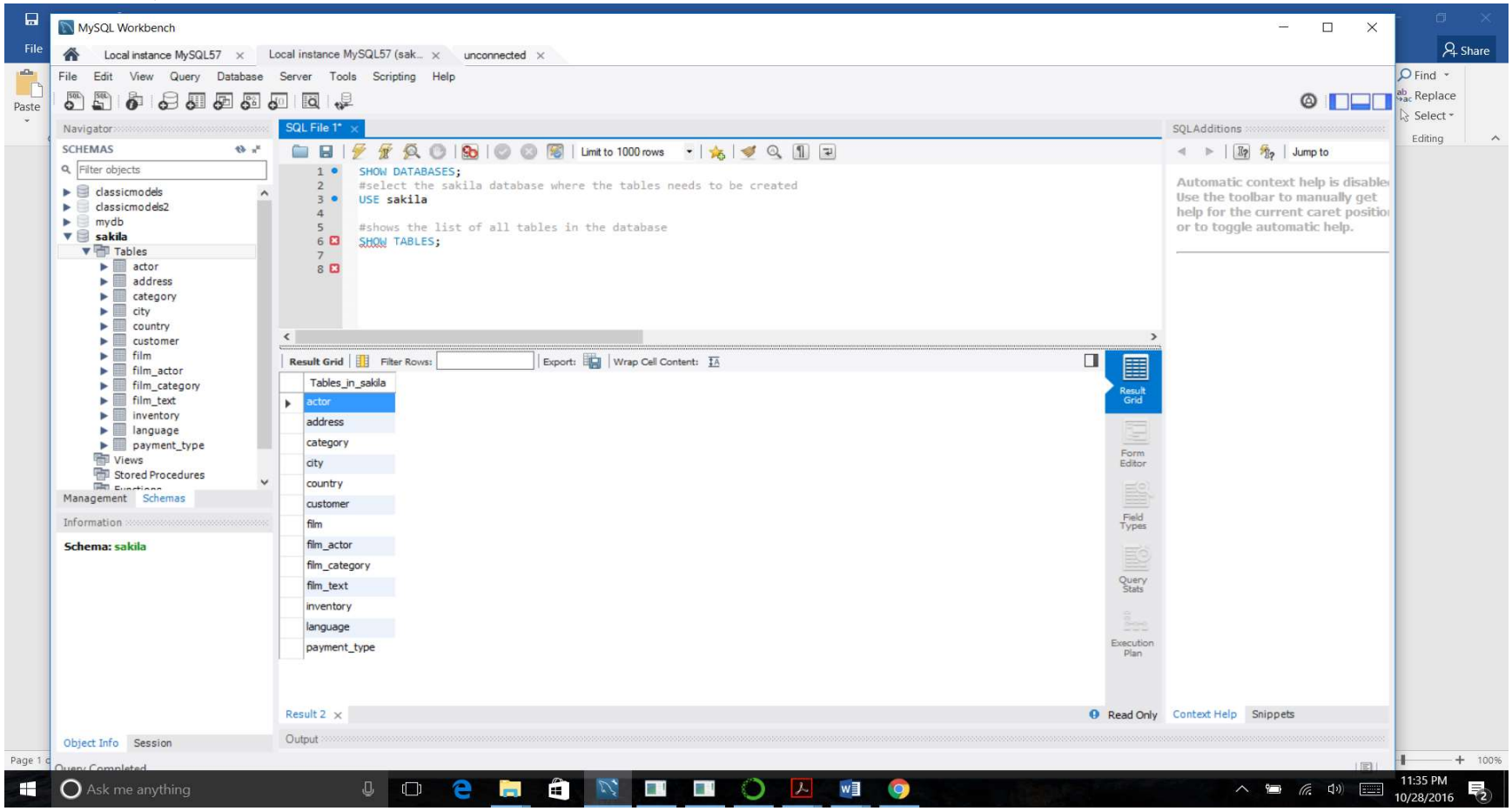
 # b) Select sakila database.

USE sakila



 # c) Show all tables in the sakila database.

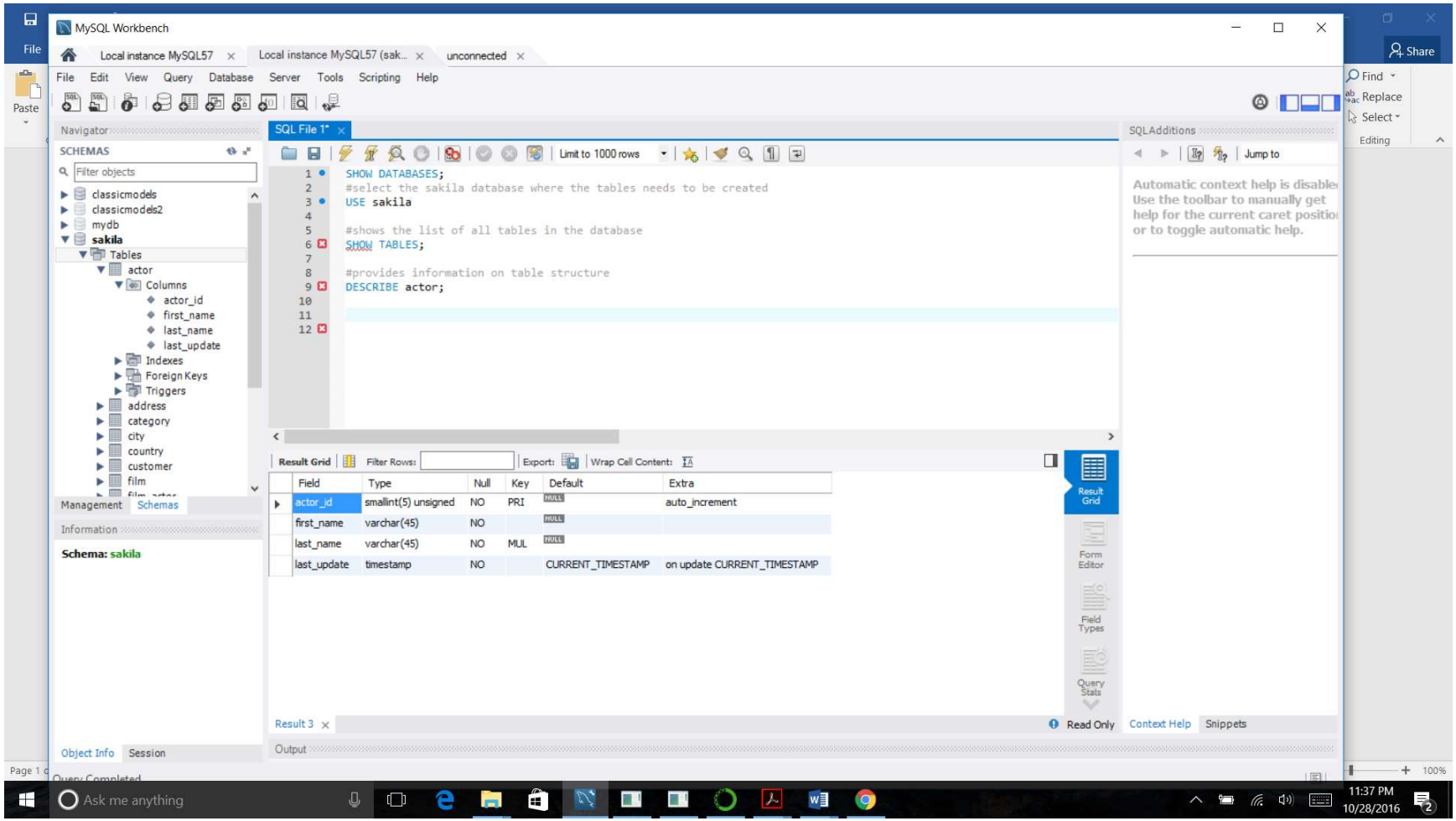
SHOW TABLES;



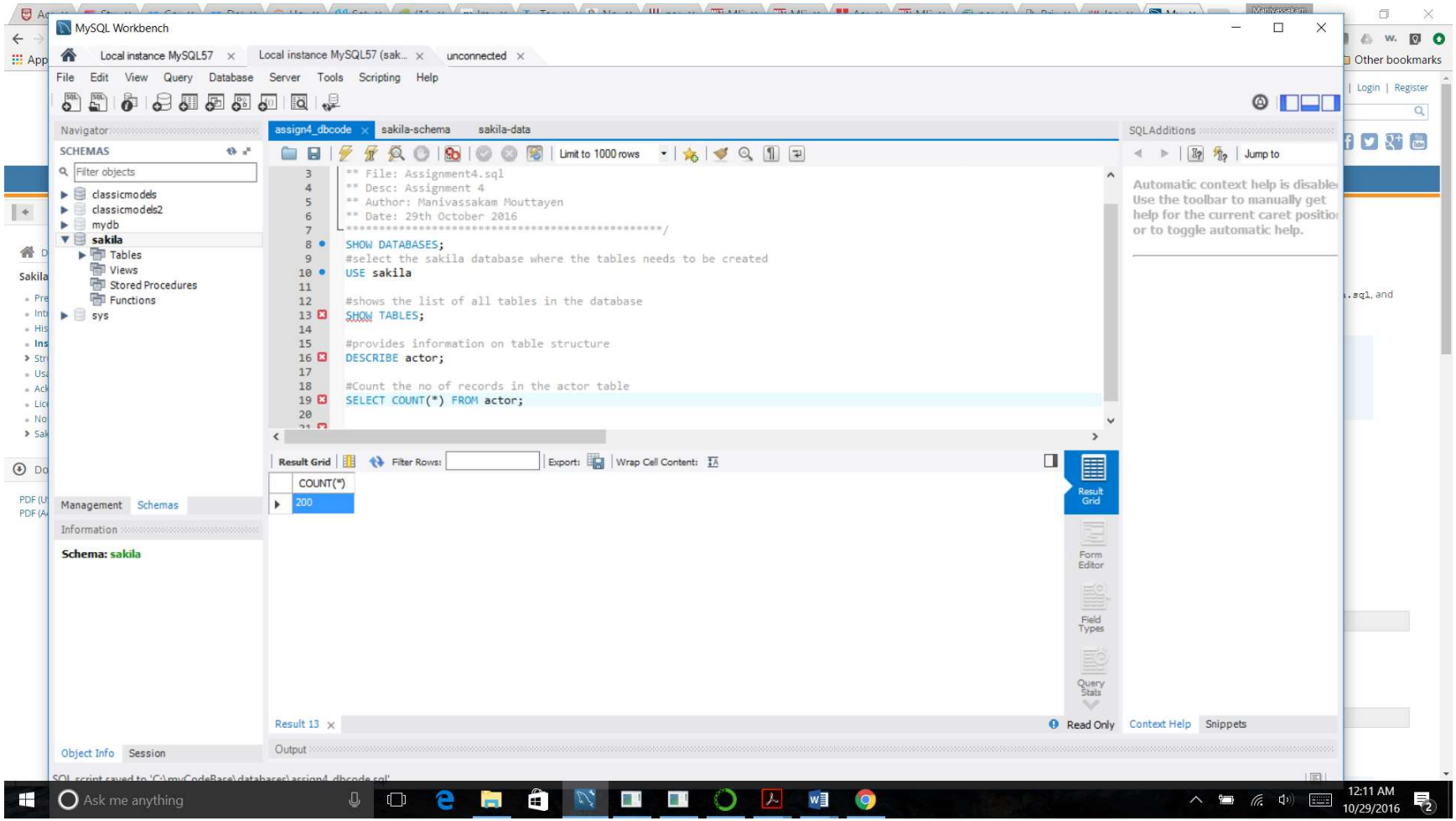
Tables_in_sakila
actor
actor_info

address
category
city
country
customer
customer_list
film
film_actor
film_category
film_list
film_text
inventory
language
nicer_but_slower_film_list
payment
rental
sales_by_film_category
sales_by_store
staff
staff_list
store

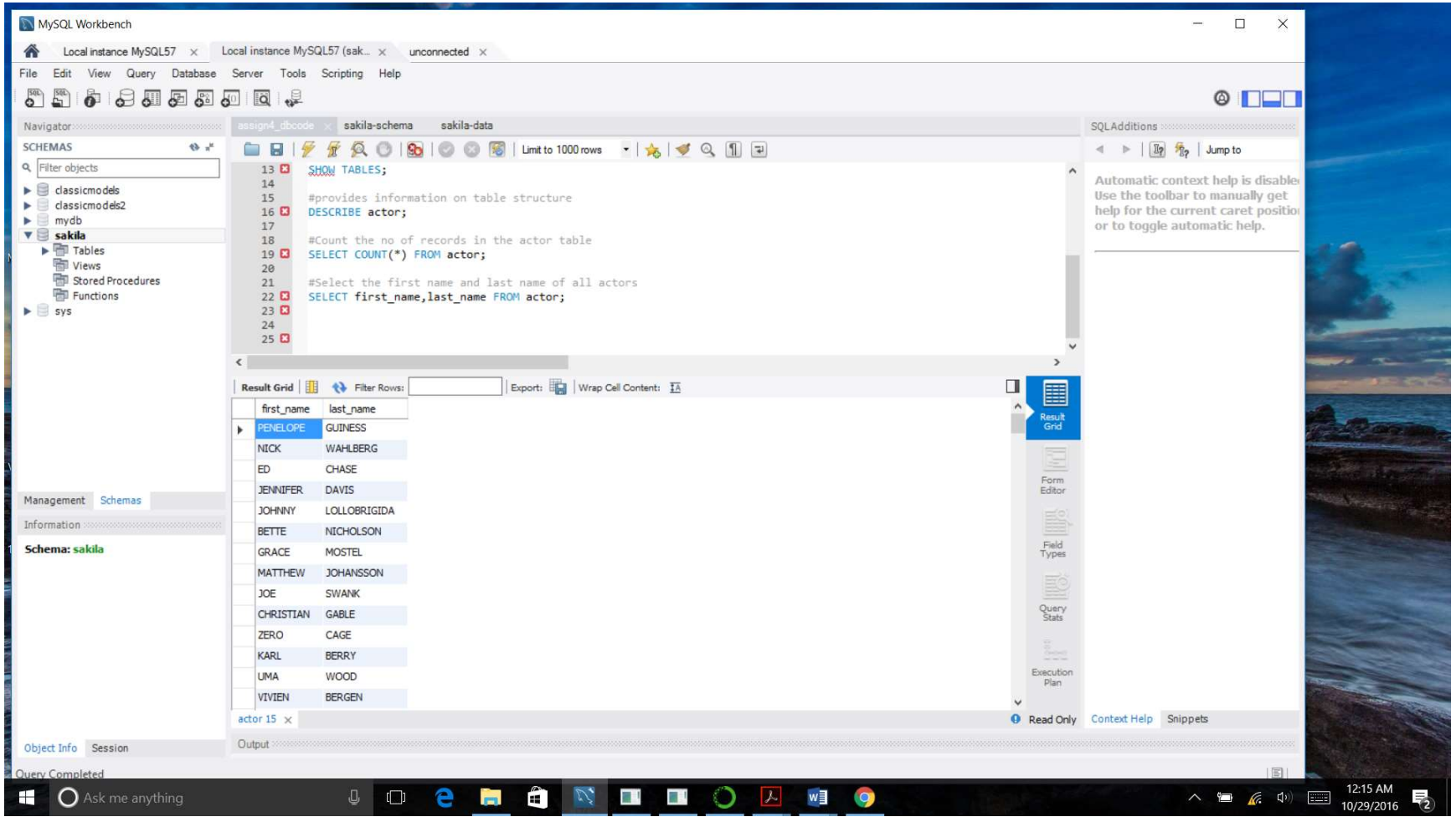
 # d) Show each of the columns along with their data types for the actor table.
DESCRIBE actor;



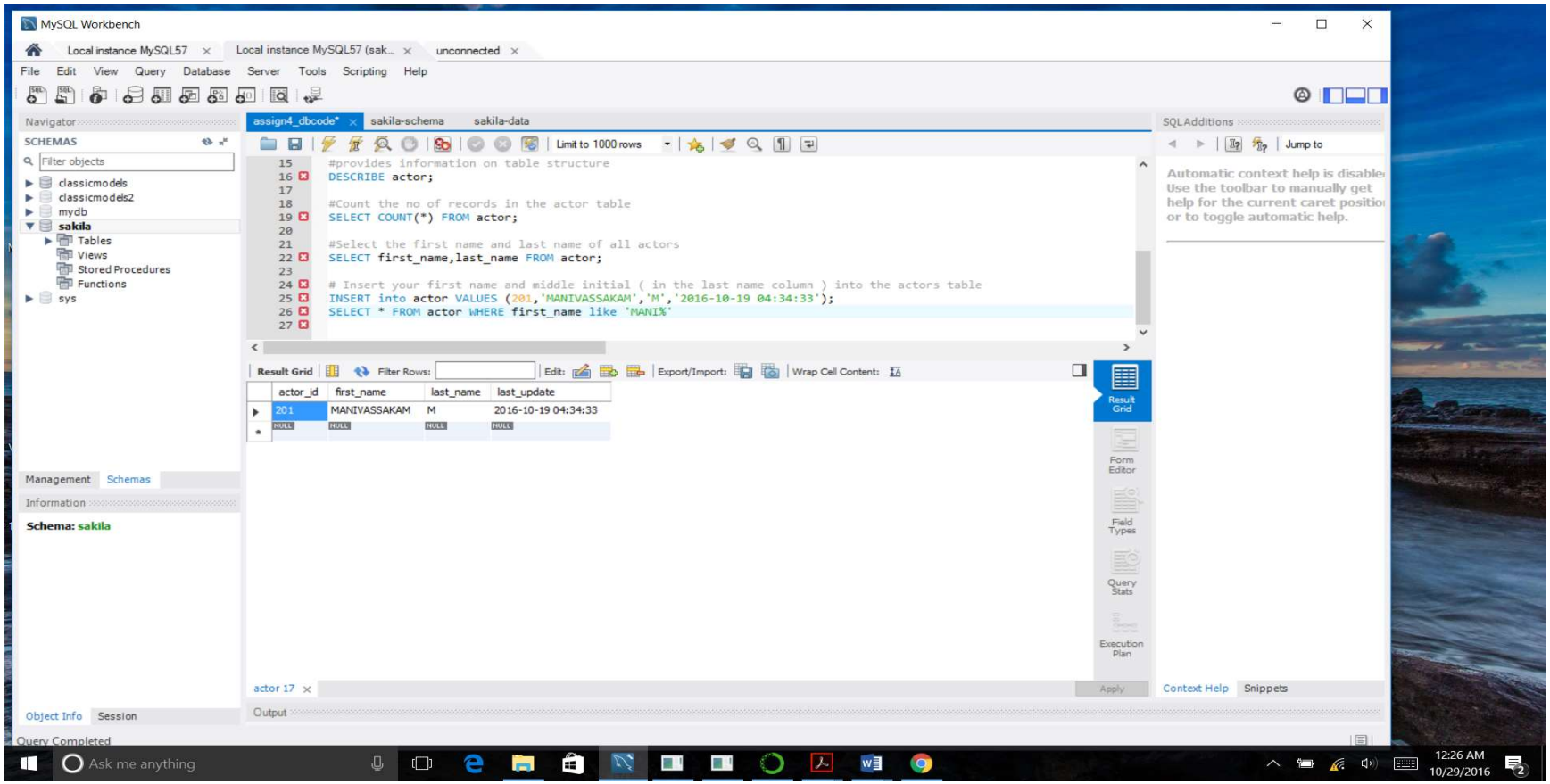
 # e) Show the total number of records in the actor table.
SELECT COUNT(*) FROM actor;



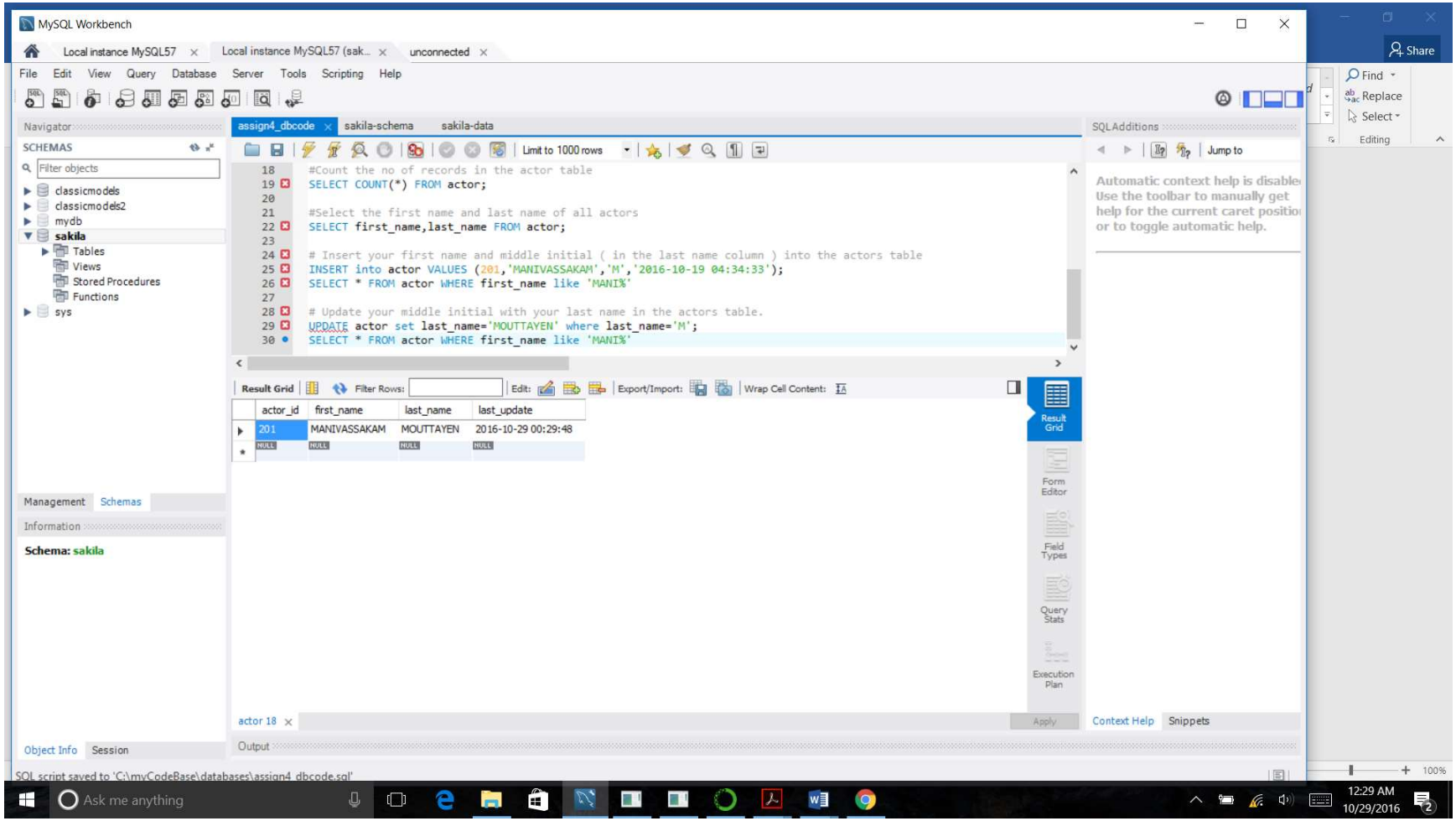
 # f) What is the first name and last name of all the actors in the actor table?
SELECT first_name,last_name FROM actor;



 # g) Insert your first name and middle initial (in the last name column) into the actors table.
INSERT into actor VALUES (201,'MANIVASSAKAM','M','2016-10-19 04:34:33');
SELECT * FROM actor WHERE first_name like 'MANI%'

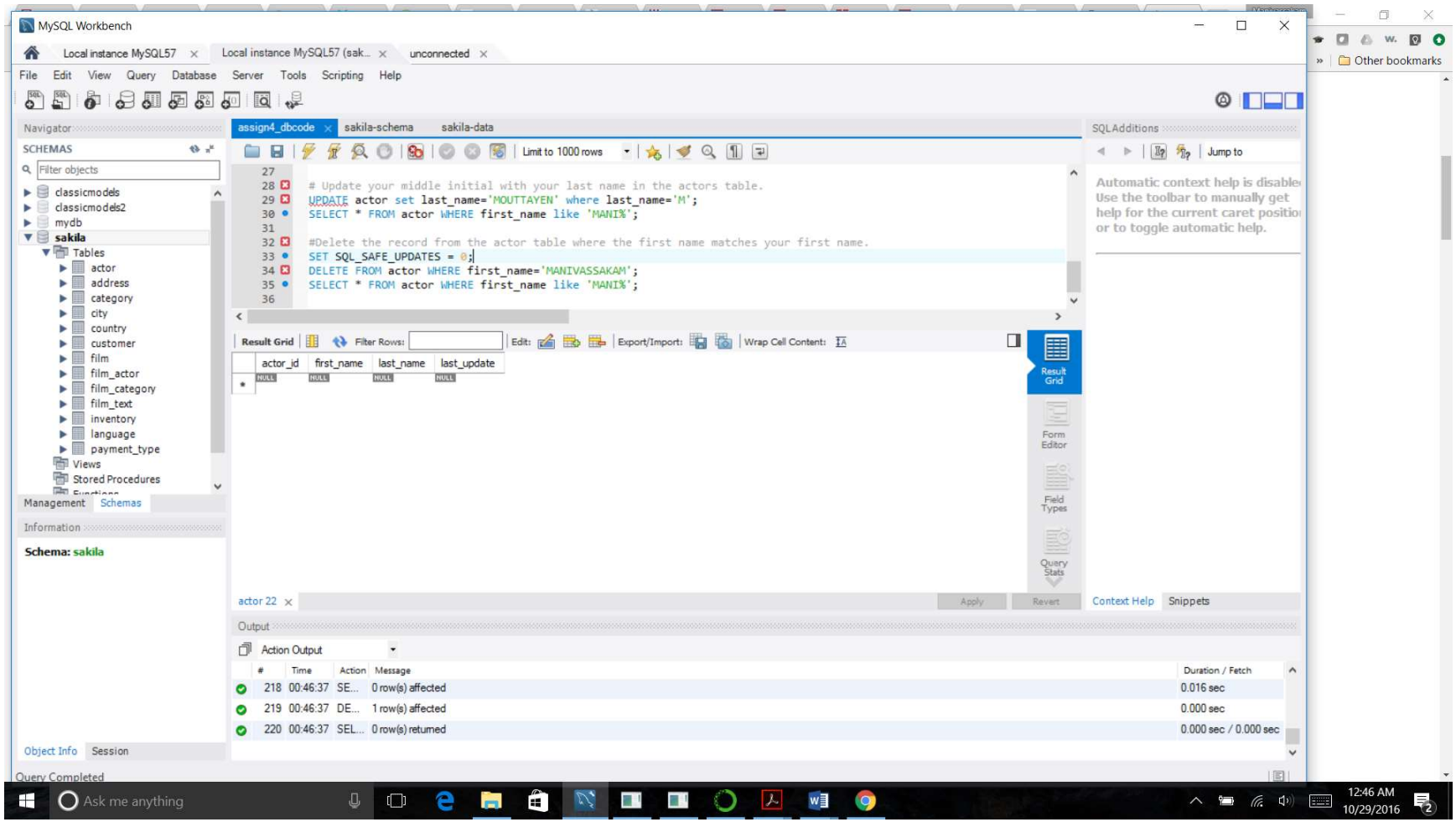


 # h) Update your middle initial with your last name in the actors table.
UPDATE actor set last_name='MOUTTAYEN' where last_name='M';
SELECT * FROM actor WHERE first_name like 'MANI%';



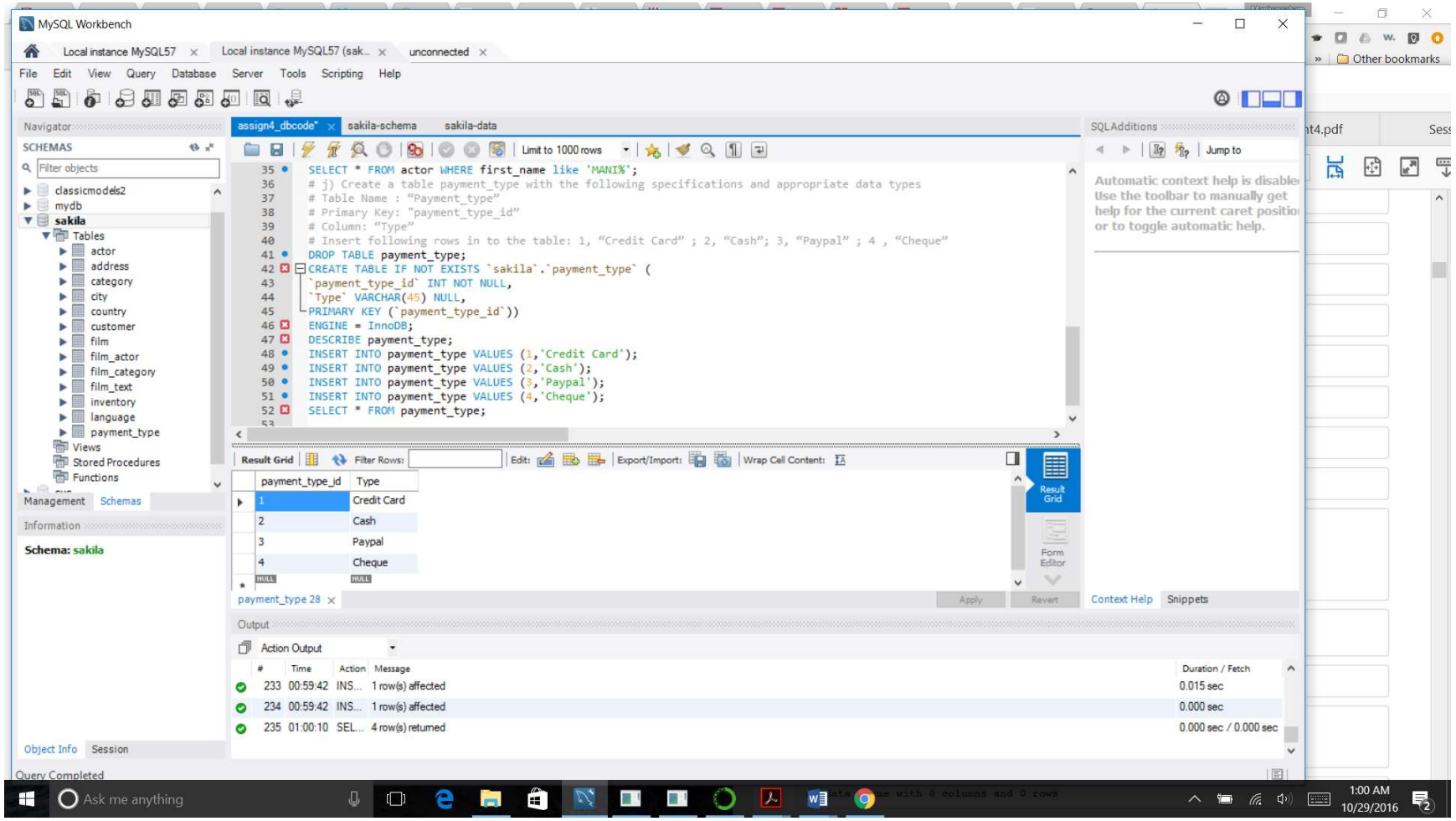
 # i) Delete the record from the actor table where the first name matches your first name.

```
SET SQL_SAFE_UPDATES = 0;
DELETE FROM actor WHERE first_name='MANIVASSAKAM';
SELECT * FROM actor WHERE first_name like 'MANI%';
```

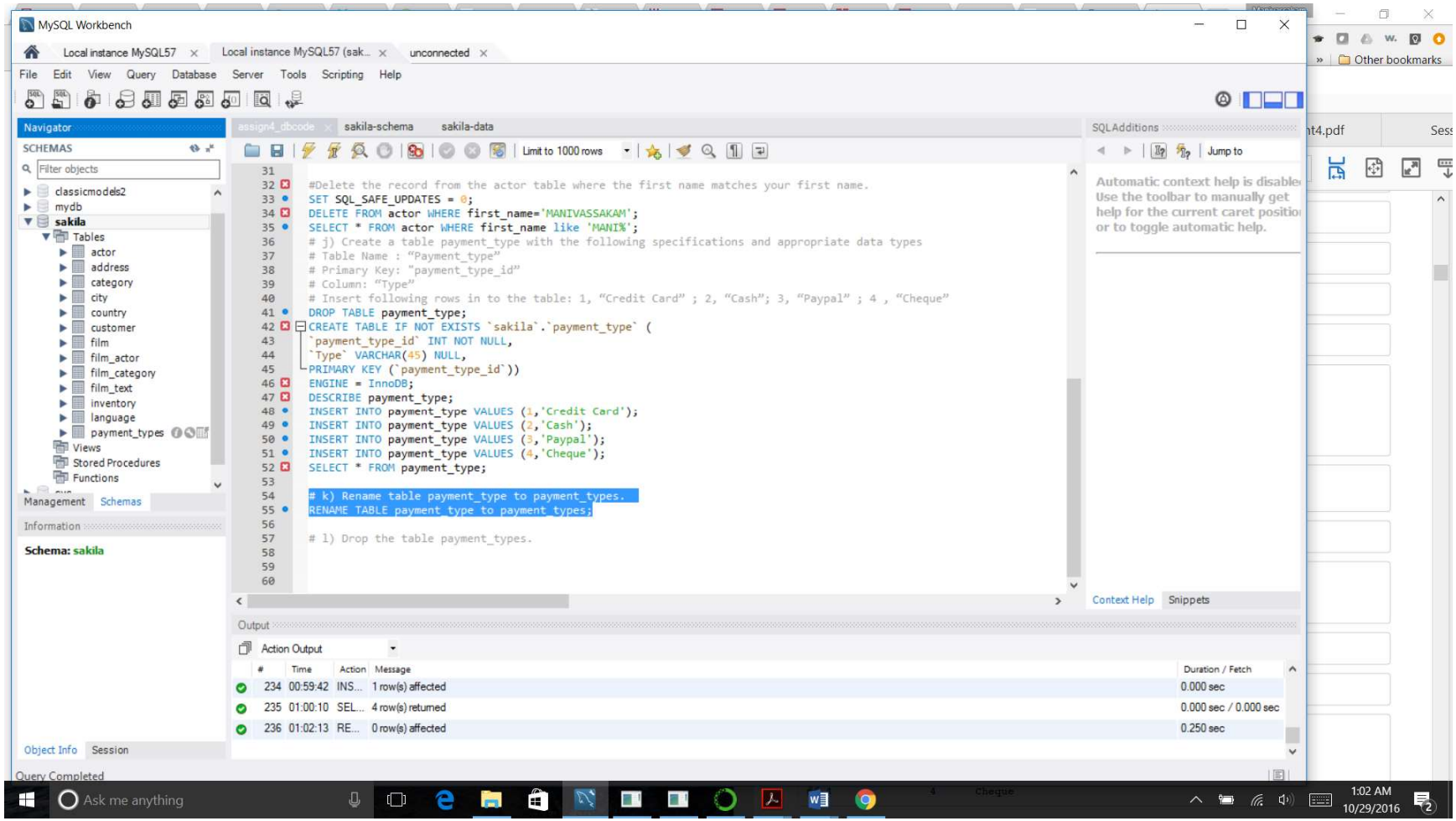


 # j) Create a table payment_type with the following specifications and appropriate data types
 # Table Name : “Payment_type”
 # Primary Key: "payment_type_id”
 # Column: “Type”
 # Insert following rows in to the table: 1, “Credit Card” ; 2, “Cash”; 3, “Paypal” ; 4 , “Cheque”

```
DROP TABLE payment_type;
CREATE TABLE IF NOT EXISTS `sakila`.`payment_type` (
`payment_type_id` INT NOT NULL,
`Type` VARCHAR(45) NULL,
PRIMARY KEY (`payment_type_id`))
ENGINE = InnoDB;
DESCRIBE payment_type;
INSERT INTO payment_type VALUES (1,'Credit Card');
INSERT INTO payment_type VALUES (2,'Cash');
INSERT INTO payment_type VALUES (3,'Paypal');
INSERT INTO payment_type VALUES (4,'Cheque');
SELECT * FROM payment_type;
```

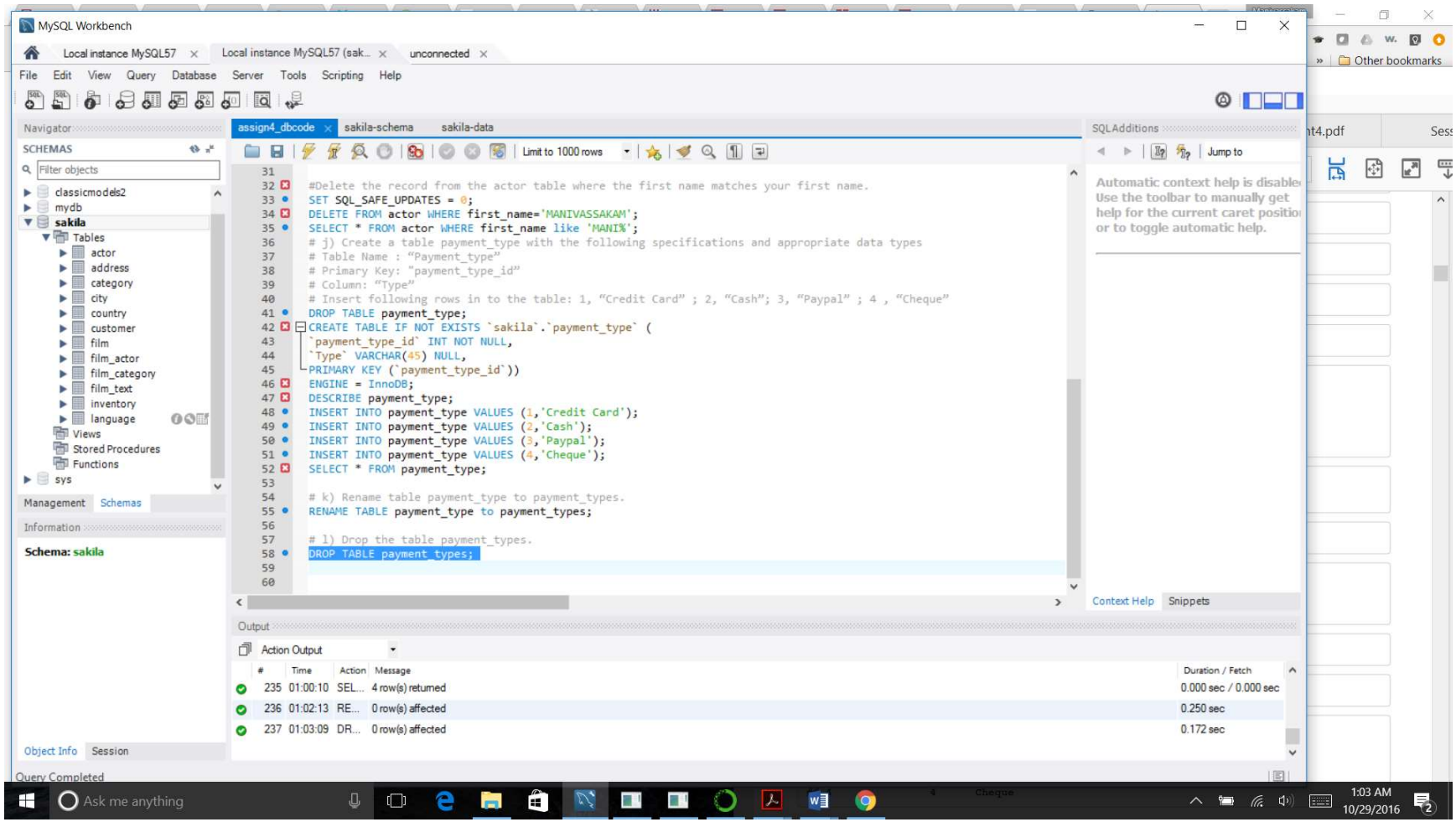

MySQL Workbench
Local instance MySQL57
Local instance MySQL57 (sak...
unconnected
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
classicmodels2
mydb
sakila
Tables
actor
address
category
city
country
customer
film
film_actor
film_category
film_text
inventory
language
payment_type
Views
Stored Procedures
Functions
Management Schemas
Information
Schema: sakila
Object Info Session
assign4_dbcode*
sakila-schema
sakila-data
Limit to 1000 rows
35 SELECT * FROM actor WHERE first_name like 'MANI%';
36 # j) Create a table payment_type with the following specifications and appropriate data types
37 # Table Name : "Payment_type"
38 # Primary Key: "payment_type_id"
39 # Column: "Type"
40 # Insert following rows in to the table: 1, "Credit Card" ; 2, "Cash"; 3, "Paypal" ; 4 , "Cheque"
41 DROP TABLE payment_type;
42 CREATE TABLE IF NOT EXISTS `sakila`.`payment_type` (
43 `payment_type_id` INT NOT NULL,
44 `Type` VARCHAR(45) NULL,
45 PRIMARY KEY (`payment_type_id`))
46 ENGINE = InnoDB;
47 DESCRIBE payment_type;
48 INSERT INTO payment_type VALUES (1,'Credit Card');
49 INSERT INTO payment_type VALUES (2,'Cash');
50 INSERT INTO payment_type VALUES (3,'Paypal');
51 INSERT INTO payment_type VALUES (4,'Cheque');
52 SELECT * FROM payment_type;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
payment_type_id Type
1 Credit Card
2 Cash
3 Paypal
4 Cheque
payment_type 28
Apply
Revert
SQLAdditions
Jump to
Automatic context help is disable
Use the toolbar to manually get
help for the current caret positio
or to toggle automatic help.
Context Help Snippets
Output
Action Output
Time Action Message Duration / Fetch
233 00:59:42 INS... 1 row(s) affected 0.015 sec
234 00:59:42 INS... 1 row(s) affected 0.000 sec
235 01:00:10 SEL... 4 row(s) returned 0.000 sec / 0.000 sec
Query Completed
Other bookmarks
Ask me anything
1:00 AM
10/29/2016

 # k) Rename table payment_type to payment_types.
RENAME TABLE payment_type to payment_types;



 # l) Drop the table payment_types.

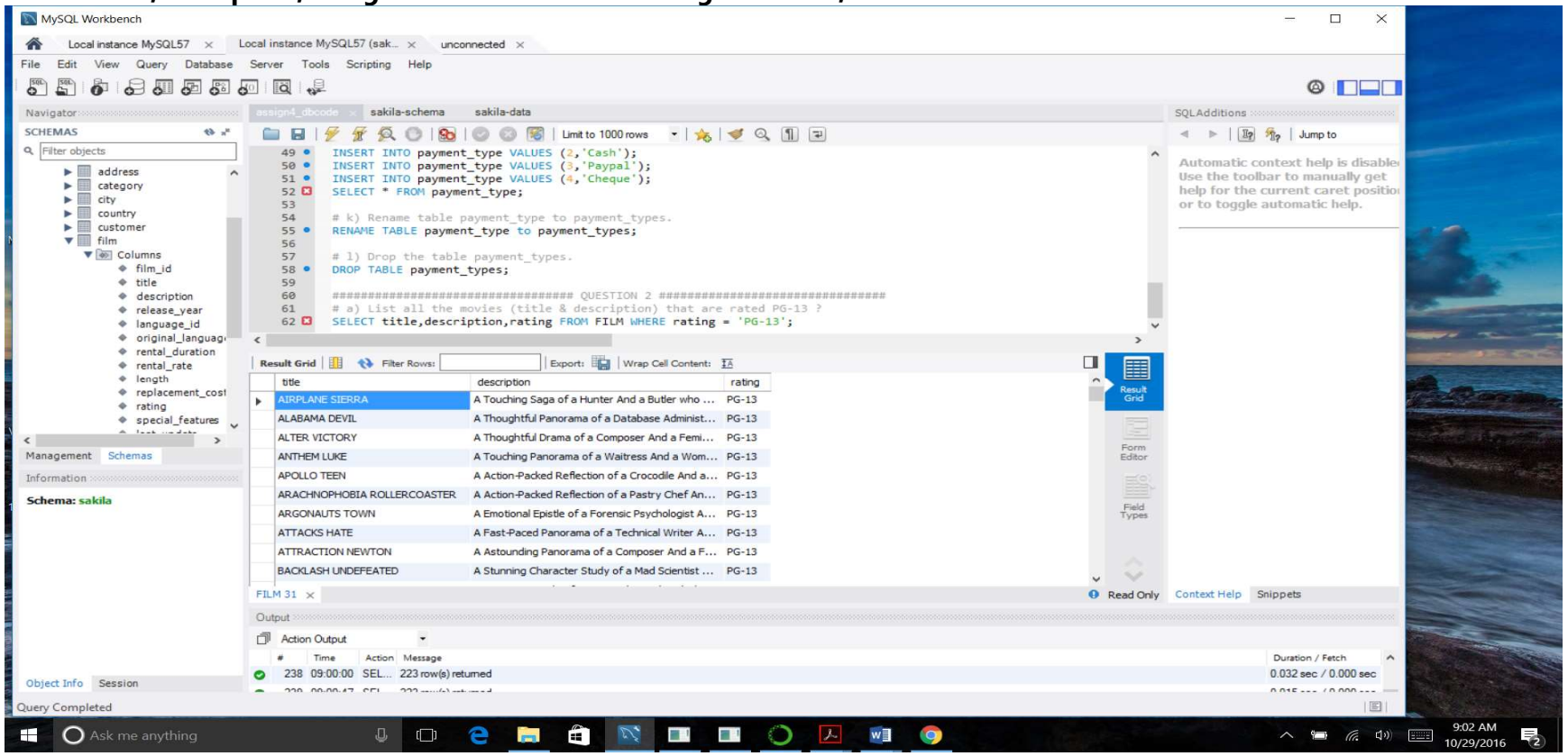
DROP TABLE payment_types;



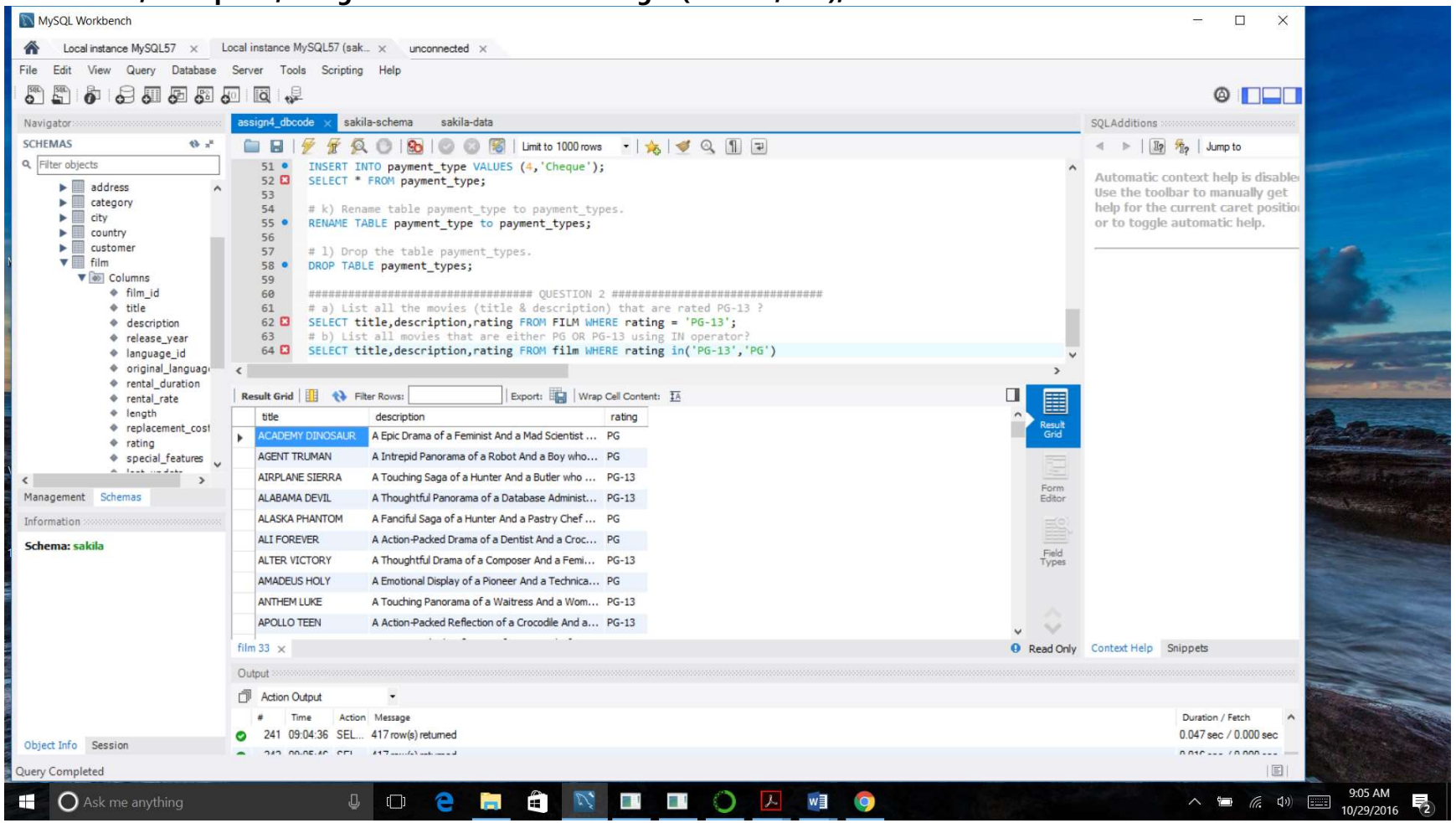
########## QUESTION 2 ##########

 # a) List all the movies (title & description) that are rated PG-13 ?

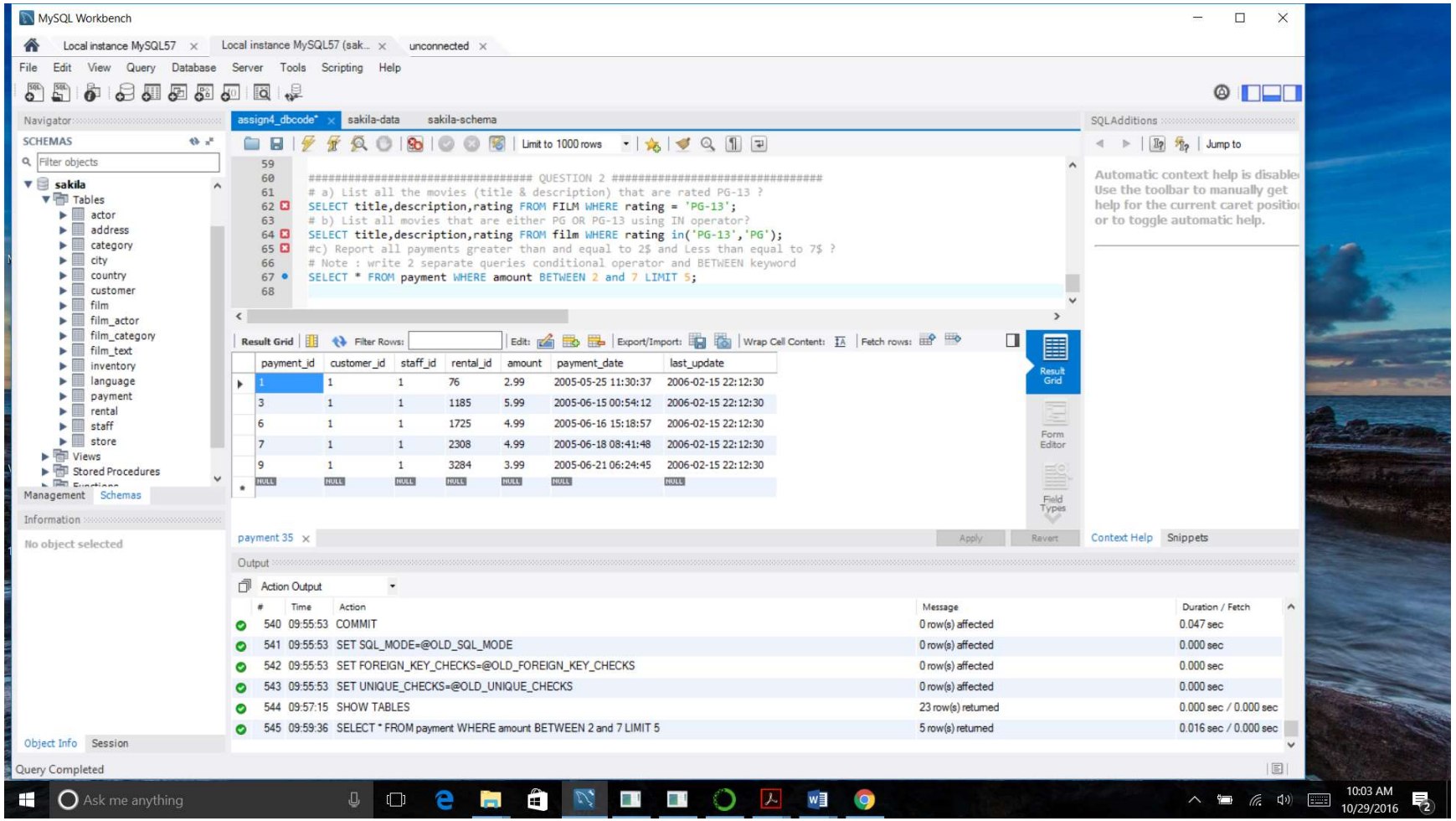
SELECT title,description,rating FROM FILM WHERE rating = 'PG-13';



 # b) List all movies that are either PG OR PG-13 using IN operator?
SELECT title,description,rating FROM film WHERE rating in('PG-13','PG');

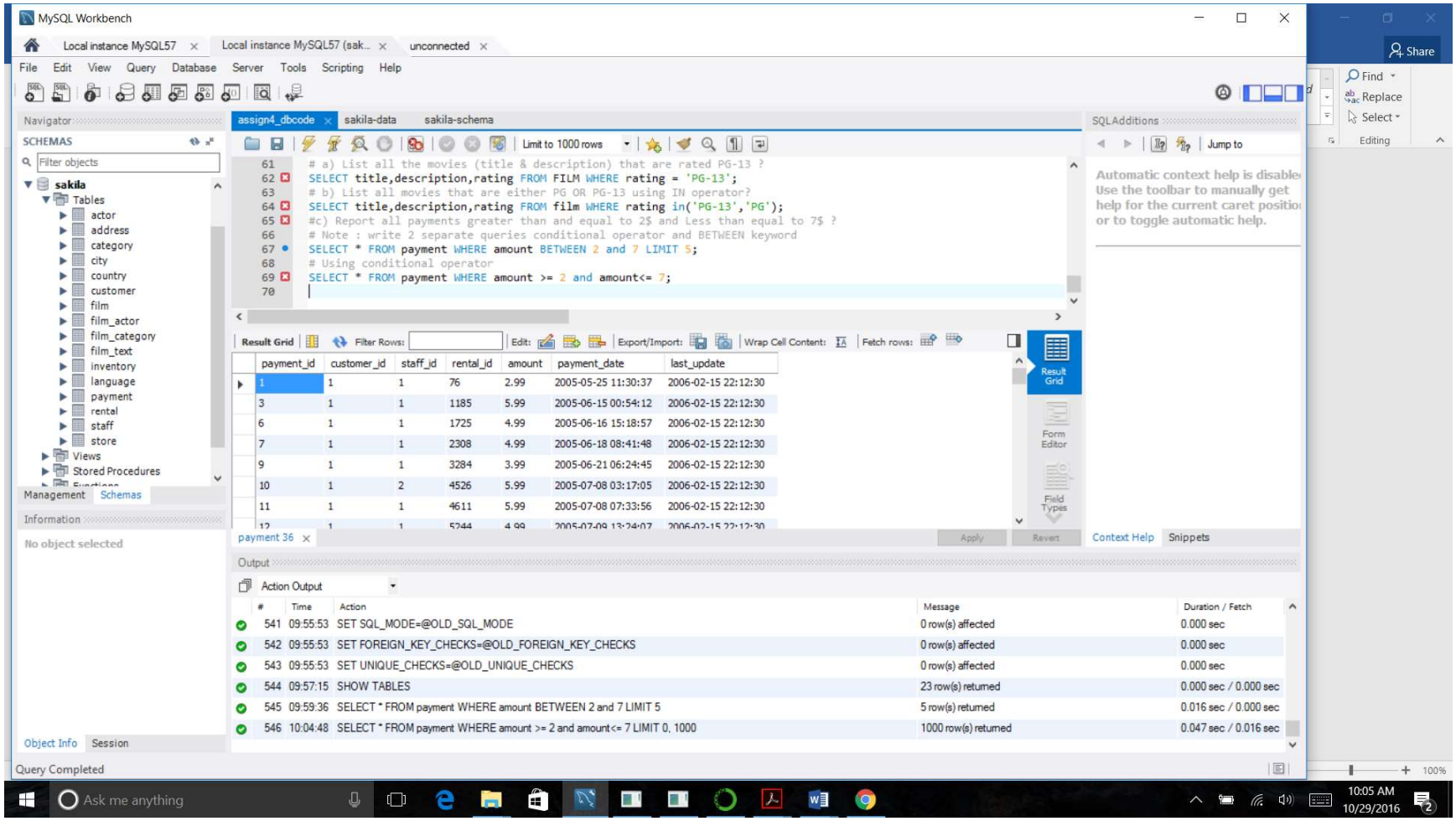


 # c) Report all payments greater than and equal to 2$ and Less than equal to 7$? #
Note : write 2 separate queries conditional operator and BETWEEN keyword
SELECT * FROM payment WHERE amount BETWEEN 2 and 7 LIMIT 5;

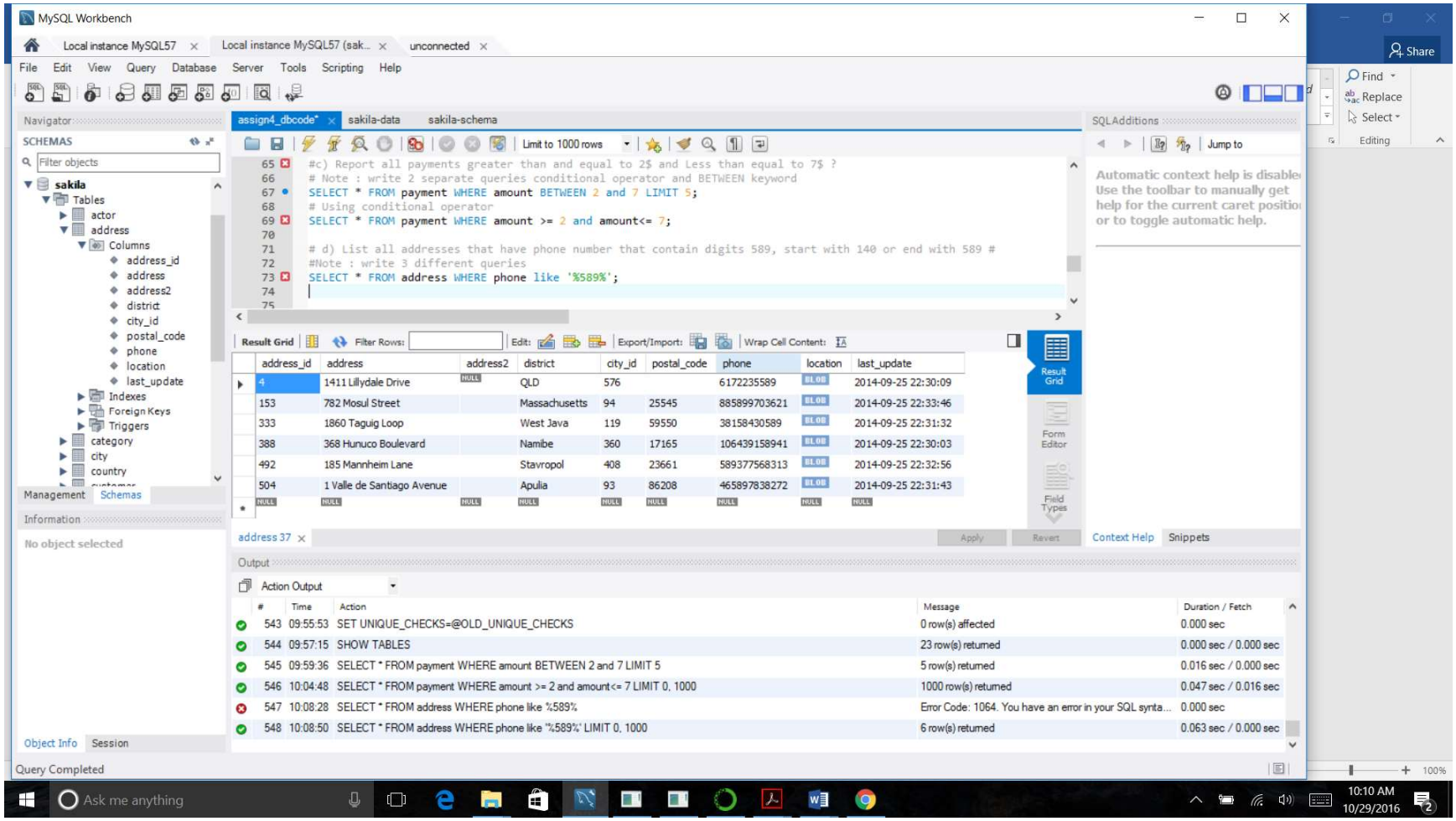


 # Using Conditional Operator

SELECT * FROM payment WHERE amount >= 2 and amount<= 7;

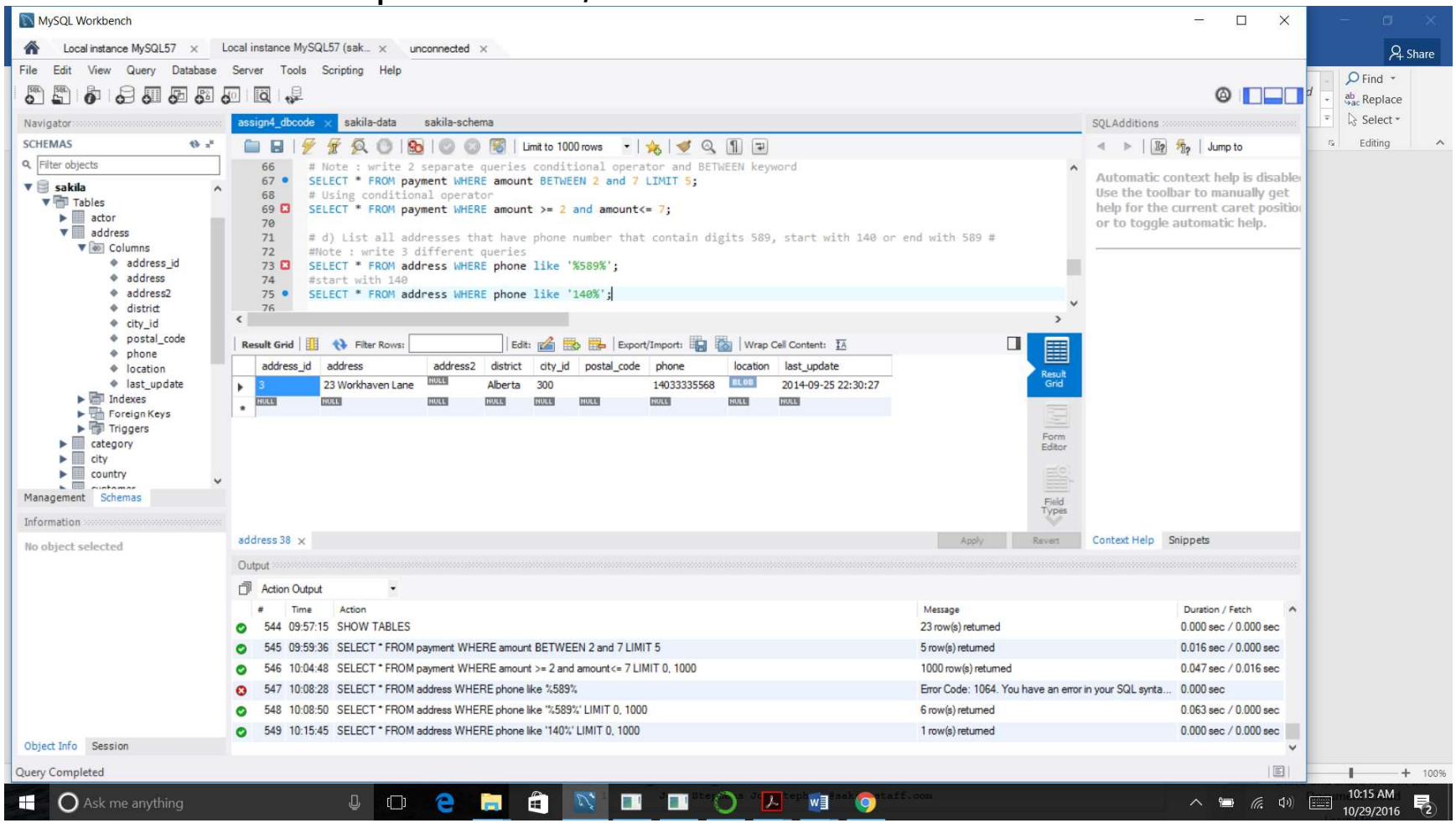


 # d) List all addresses that have phone number that contain digits 589,
#Note : write 3 different queries
SELECT * FROM address WHERE phone like '%589%';



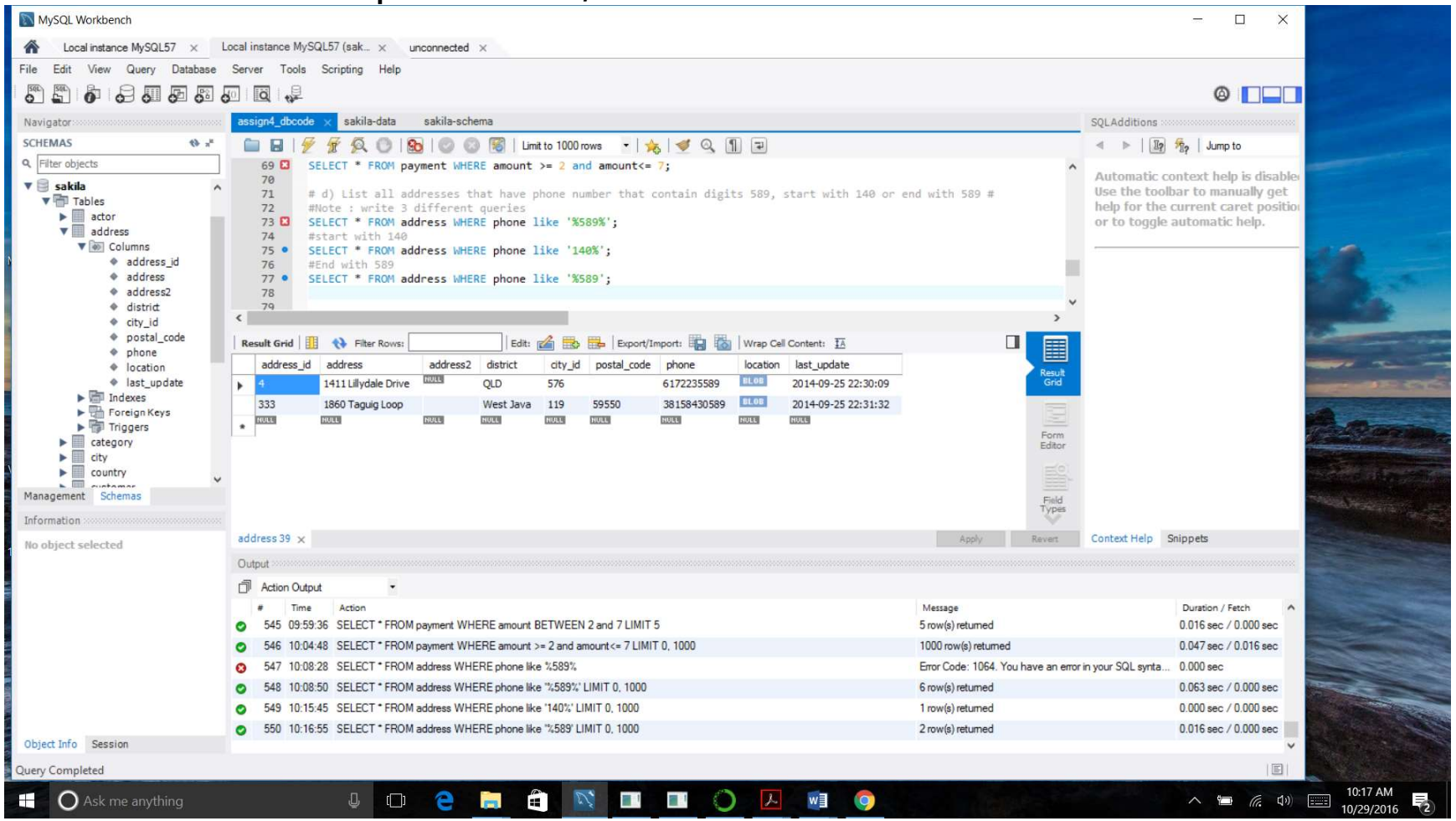
 # start with 140

SELECT * FROM address WHERE phone like '140%';

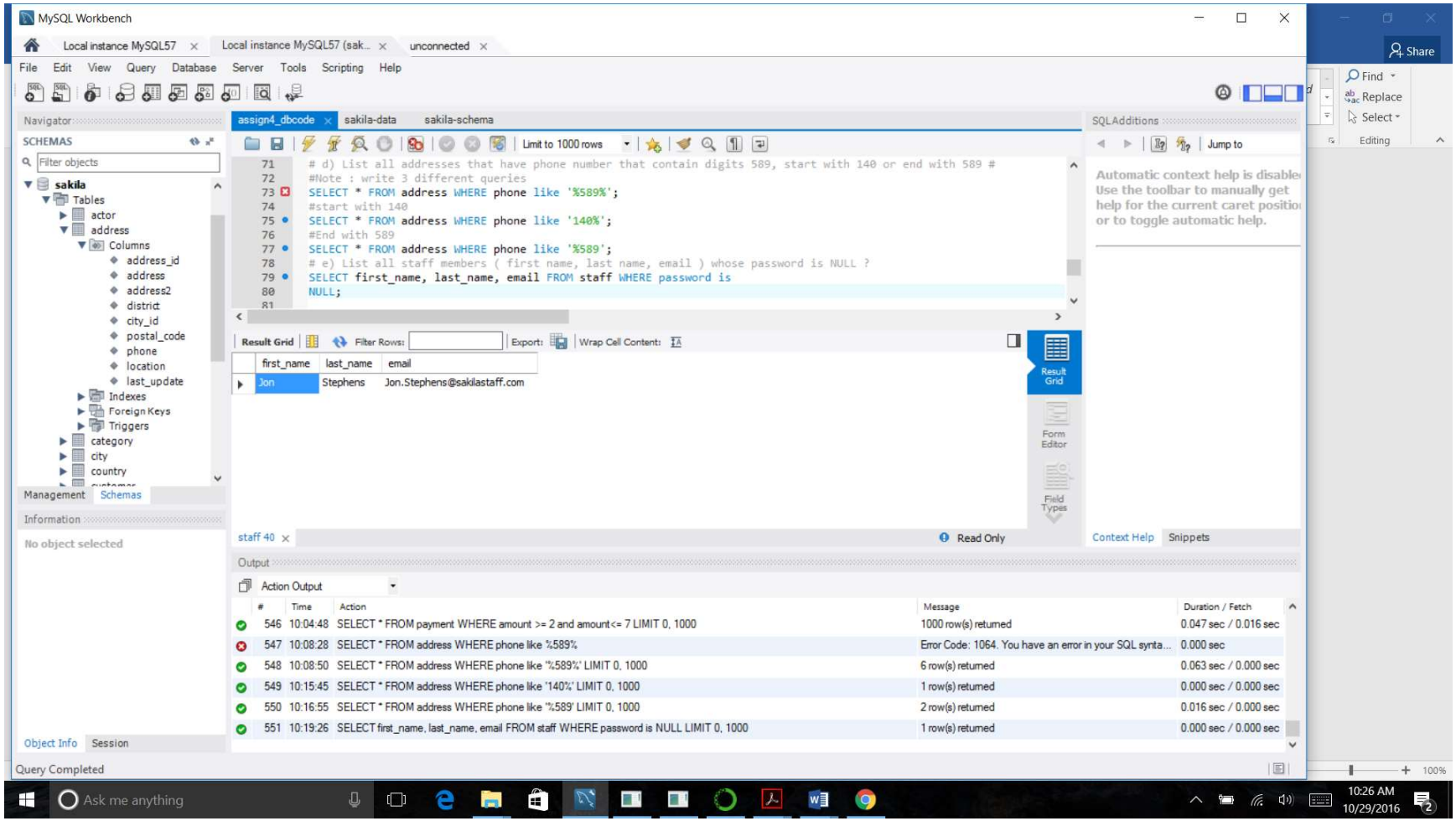


 # end with 589 #

SELECT * FROM address WHERE phone like '%589';

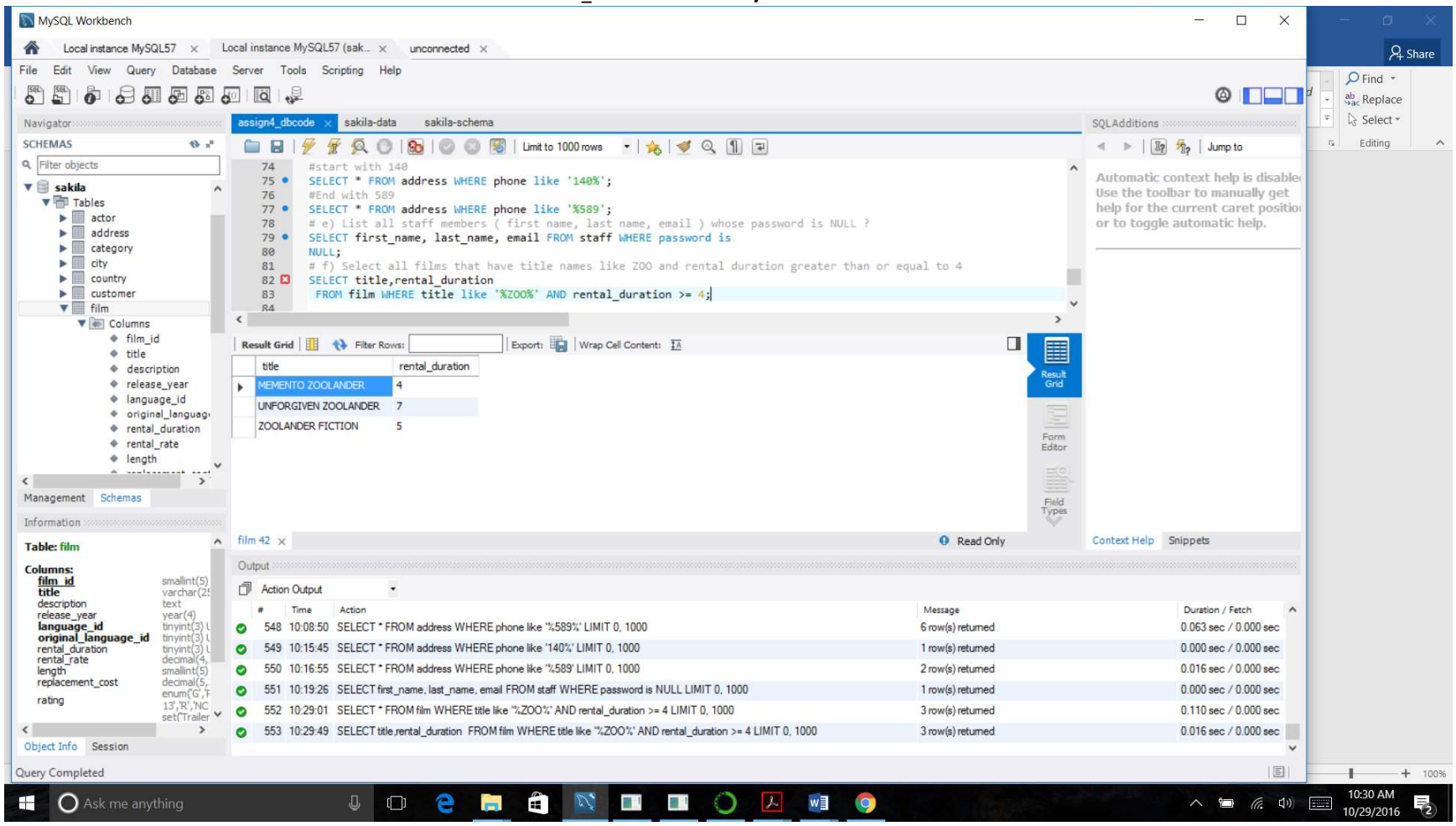


 # e) List all staff members (first name, last name, email) whose password is NULL ?
SELECT first_name, last_name, email FROM staff WHERE password is NULL;

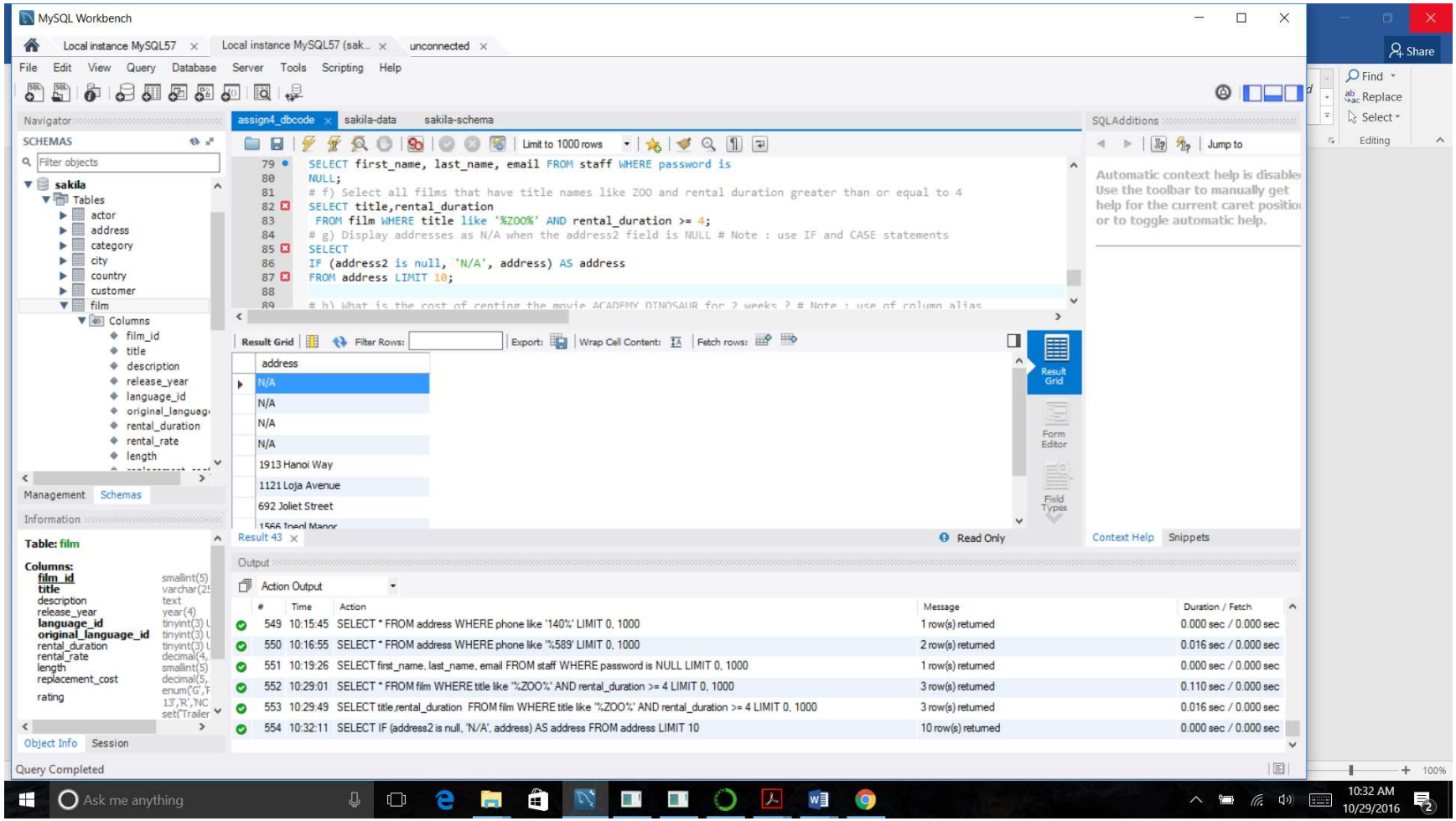


 # f) Select all films that have title names like ZOO and rental duration greater than or equal to 4

```
SELECT title,rental_duration
 FROM film WHERE title like '%ZOO%' AND rental_duration >= 4;
```

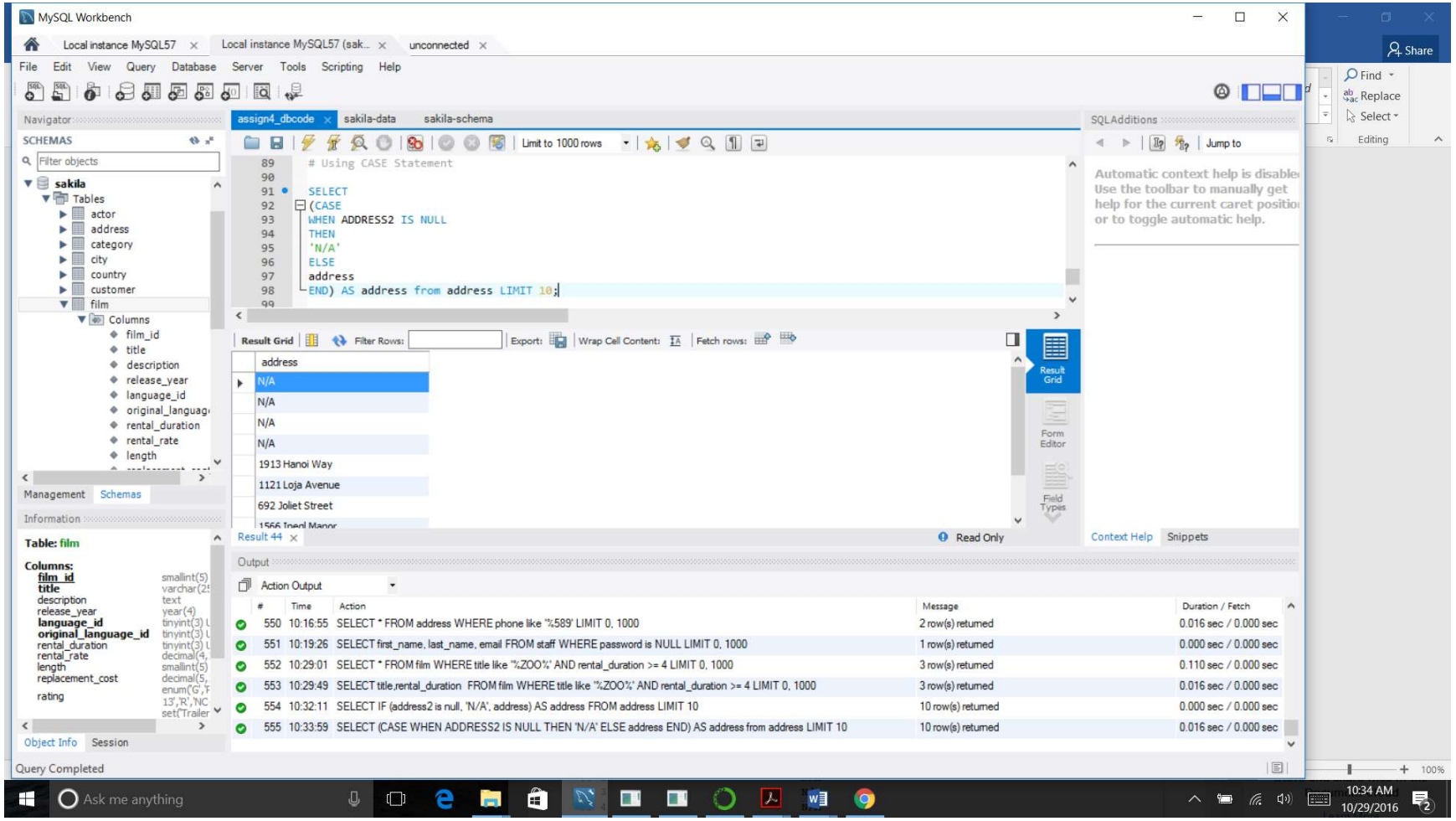


 # g) Display addresses as N/A when the address2 field is NULL # Note: use IF and CASE statements
SELECT
IF (address2 is null, 'N/A', address) AS address
FROM address LIMIT 10;



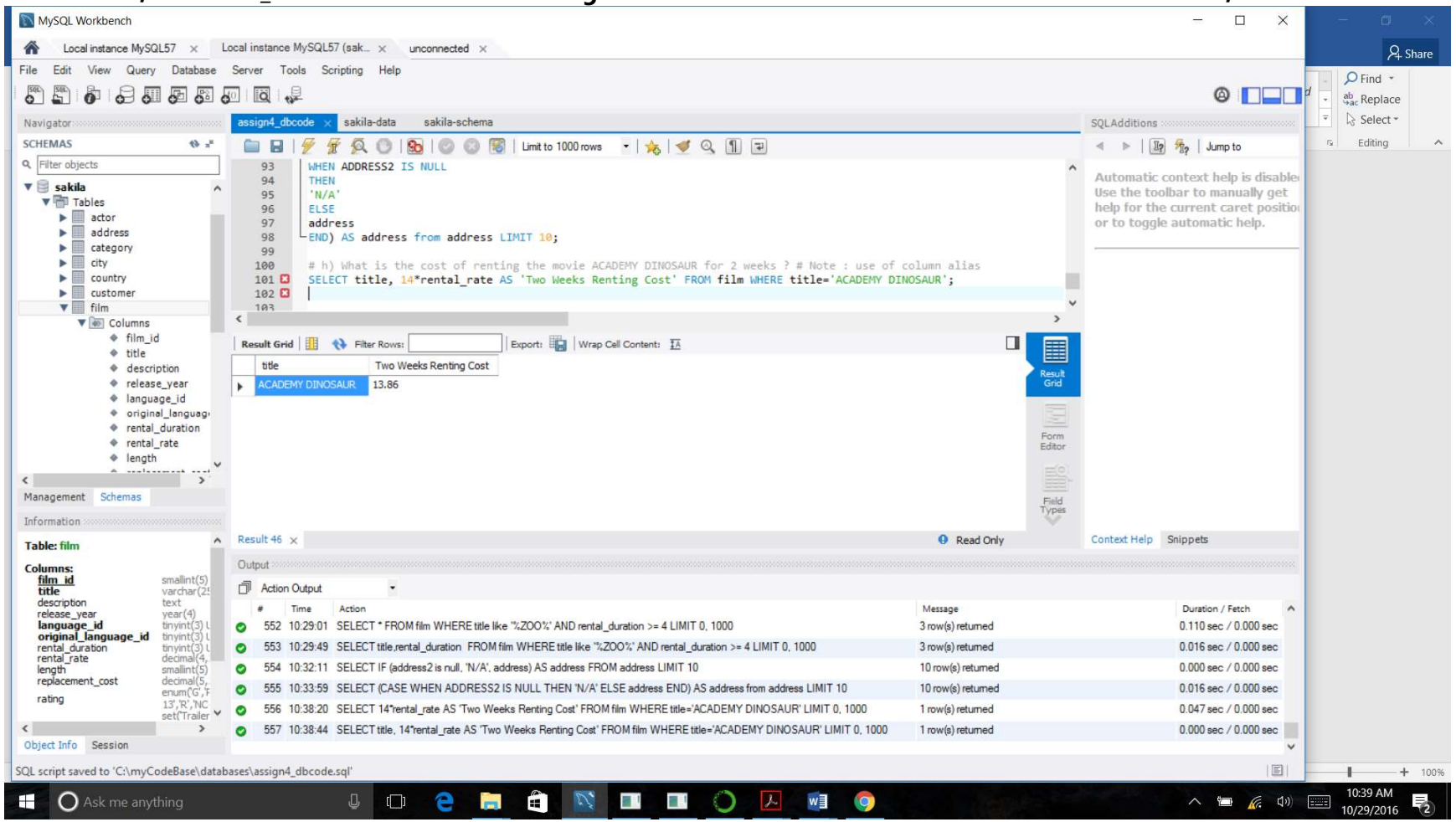
 # Using Case statement

```
SELECT
(CASE
WHEN ADDRESS2 IS NULL
THEN
'N/A'
ELSE
address
END) AS address from address LIMIT 10;
```

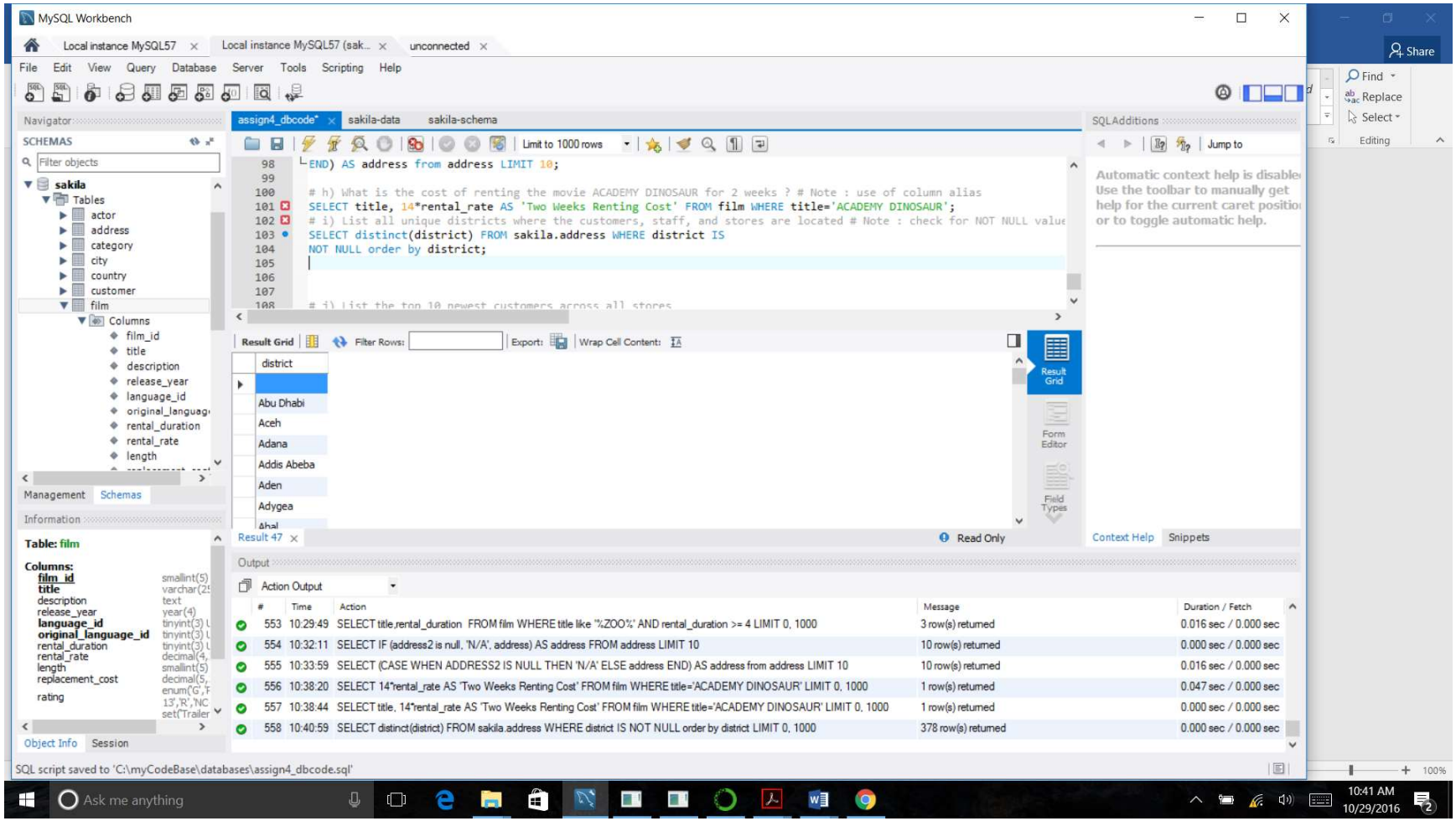

MySQL Workbench
Local instance MySQL57
Local instance MySQL57 (sak...
unconnected
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sakila
Tables
actor
address
category
city
country
customer
film
Columns
film_id
title
description
release_year
language_id
original_languag
rental_duration
rental_rate
length
Management Schemas
Information
Table: film
Columns:
film_id smallint(5)
title varchar(25
description text
release_year year(4)
language_id tinyint(3) U
original_language_id tinyint(3) U
rental_duration tinyint(3) U
rental_rate decimal(4,
length smallint(5)
replacement_cost decimal(5,
rating enum('G','F
13','R','NC
set('Trailer
Object Info Session
assign4_dbcode
sakila-data
sakila-schema
Limit to 1000 rows
89 # Using CASE Statement
90
91 SELECT
92 (CASE
93 WHEN ADDRESS2 IS NULL
94 THEN
95 'N/A'
96 ELSE
97 address
98 END) AS address from address LIMIT 10;
99
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Fetch rows:
address
N/A
N/A
N/A
N/A
1913 Hanoi Way
1121 Loja Avenue
692 Joliet Street
1566 Inegl Manor
Result 44
Read Only
Result Grid
Form Editor
Field Types
SQLAdditions
Jump to
Automatic context help is disable
Use the toolbar to manually get
help for the current caret positio
or to toggle automatic help.
Context Help Snippets
Output
Action Output
Time Action Message Duration / Fetch
550 10:16:55 SELECT * FROM address WHERE phone like '%589' LIMIT 0, 1000 2 row(s) returned 0.016 sec / 0.000 sec
551 10:19:26 SELECT first_name, last_name, email FROM staff WHERE password is NULL LIMIT 0, 1000 1 row(s) returned 0.000 sec / 0.000 sec
552 10:29:01 SELECT * FROM film WHERE title like '%ZOO%' AND rental_duration >= 4 LIMIT 0, 1000 3 row(s) returned 0.110 sec / 0.000 sec
553 10:29:49 SELECT title,rental_duration FROM film WHERE title like '%ZOO%' AND rental_duration >= 4 LIMIT 0, 1000 3 row(s) returned 0.016 sec / 0.000 sec
554 10:32:11 SELECT IF (address2 is null, 'N/A', address) AS address FROM address LIMIT 10 10 row(s) returned 0.000 sec / 0.000 sec
555 10:33:59 SELECT (CASE WHEN ADDRESS2 IS NULL THEN 'N/A' ELSE address END) AS address from address LIMIT 10 10 row(s) returned 0.016 sec / 0.000 sec
Query Completed
Share
Find
Replace
Select
Editing
100%
Ask me anything
10:34 AM
10/29/2016

 # h) What is the cost of renting the movie ACADEMY DINOSAUR for 2 weeks? # Note: use of column alias

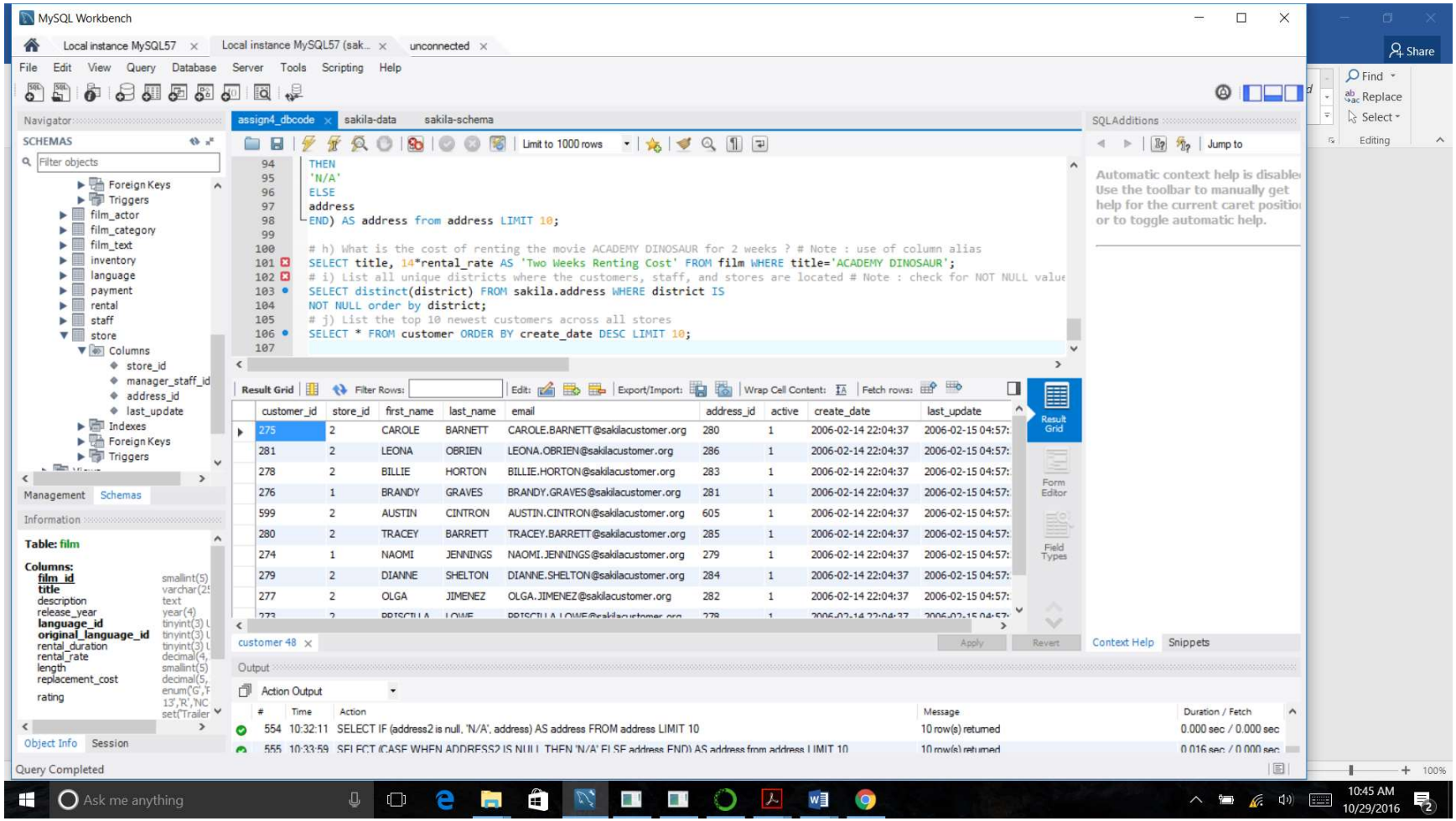
```
SELECT title, 14*rental_rate AS 'Two Weeks Renting Cost' FROM film WHERE title='ACADEMY DINOSAUR';
```



 # i) List all unique districts where the customers, staff, and stores are located # Note: check for NOT NULL values
SELECT distinct(district) FROM sakila.address WHERE district IS NOT NULL order by district;



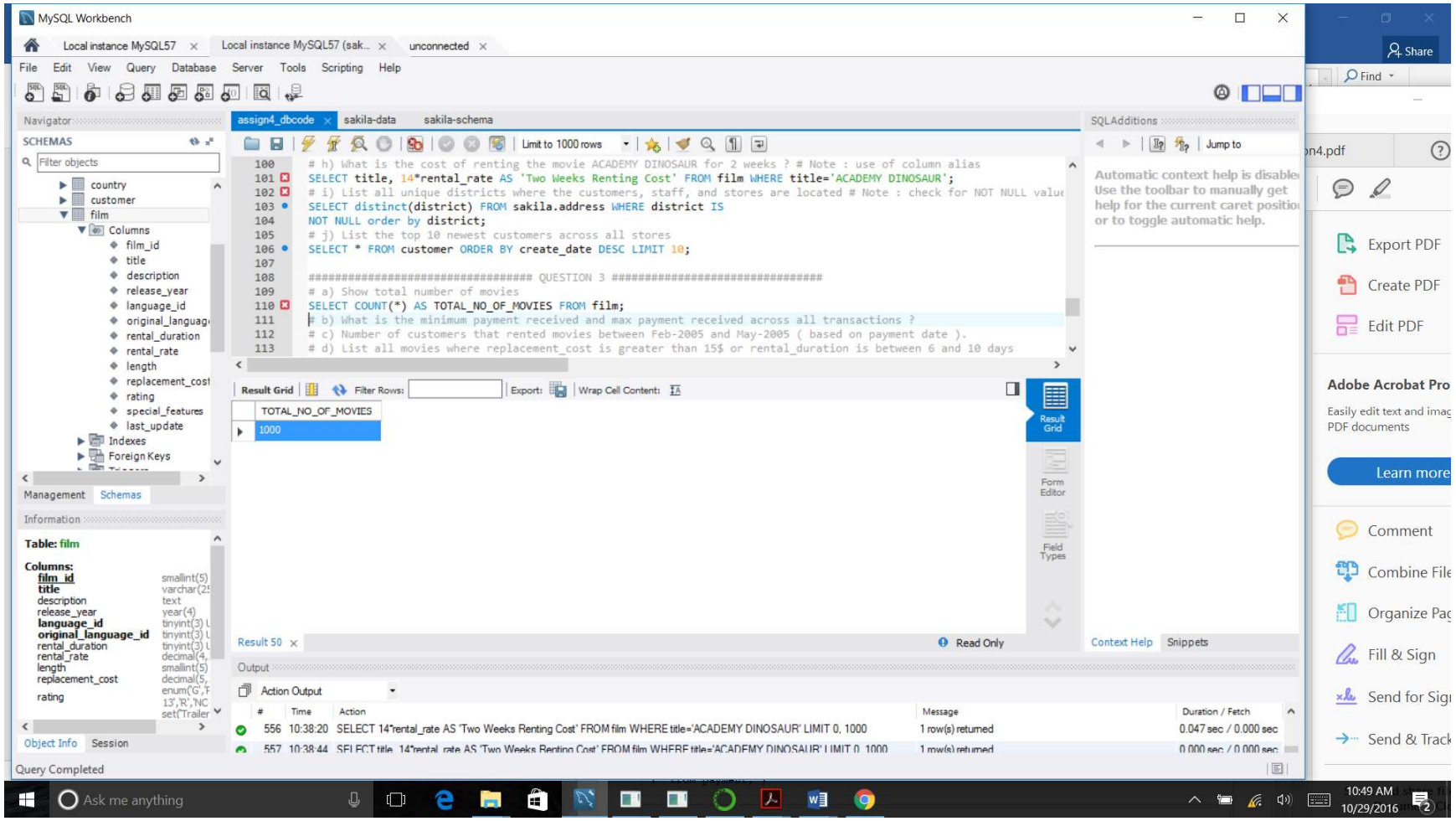
 # j) List the top 10 newest customers across all stores
SELECT * FROM customer ORDER BY create_date DESC LIMIT 10;



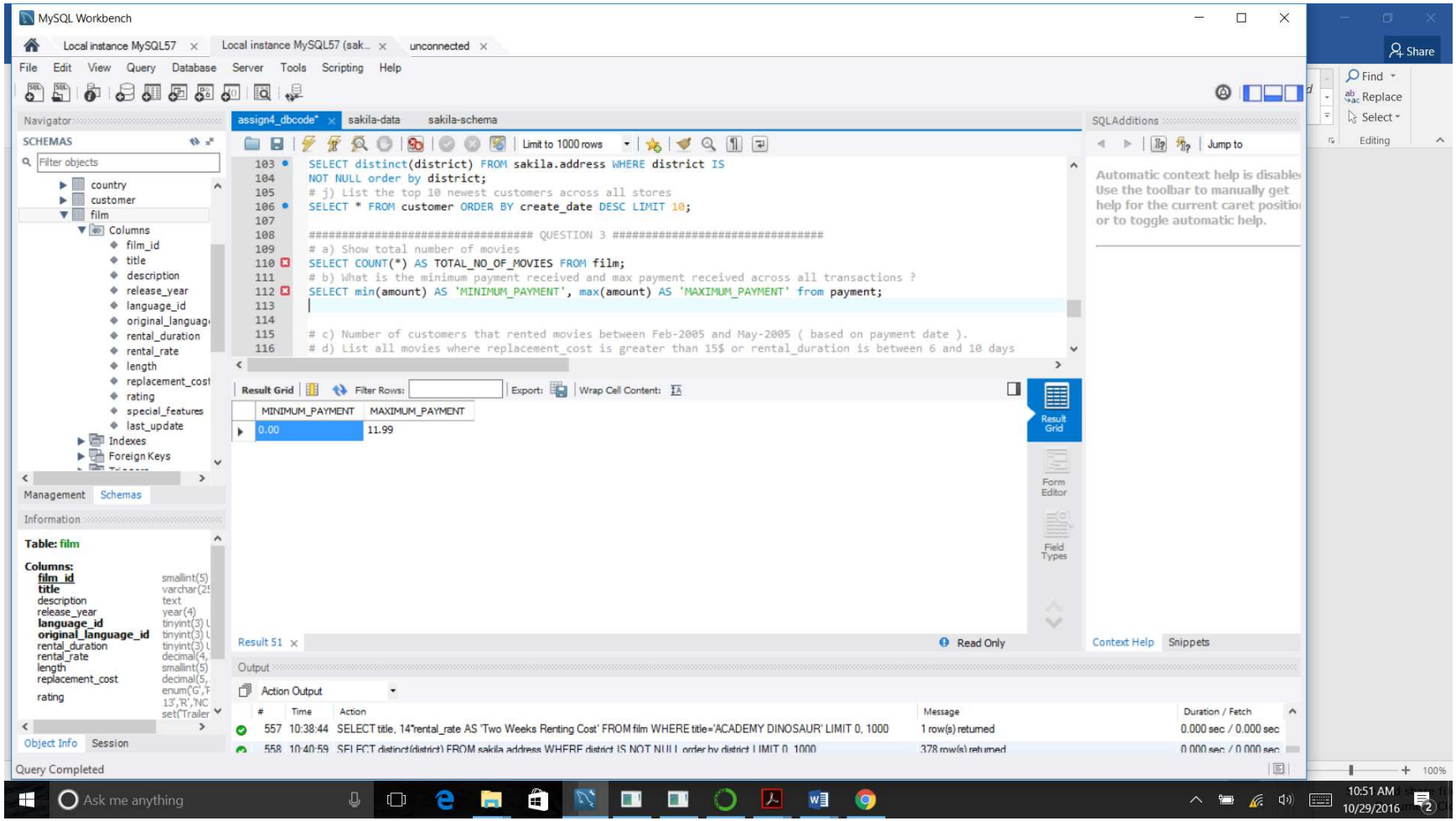
################################## QUESTION 3 ##################################

 # a) Show total number of movies

SELECT COUNT(*) AS TOTAL_NO_OF_MOVIES FROM film;

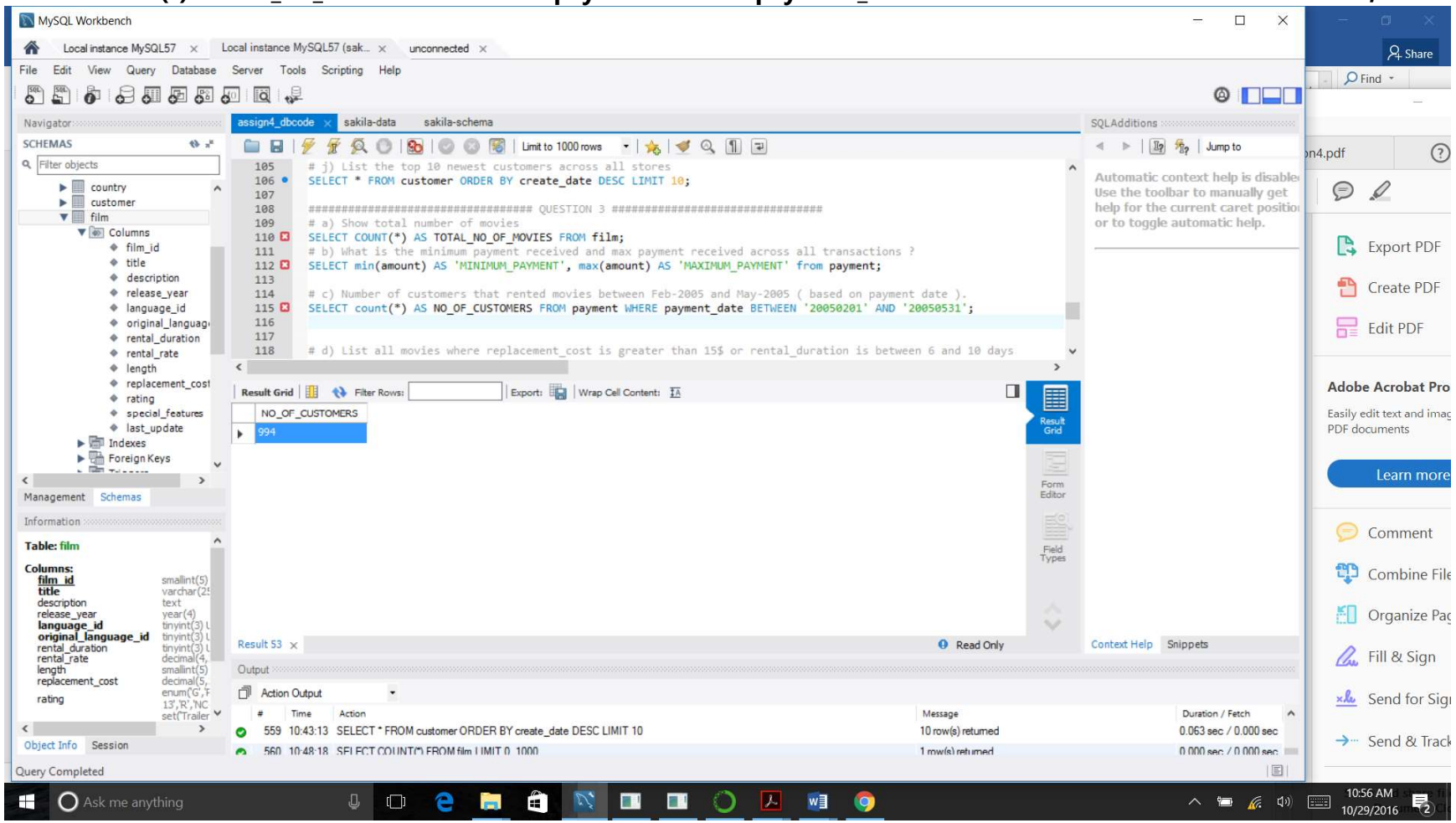


 # b) What is the minimum payment received and max payment received across all transactions?
SELECT min(amount) AS 'MINIMUM_PAYMENT', max(amount) AS 'MAXIMUM_PAYMENT' from payment;

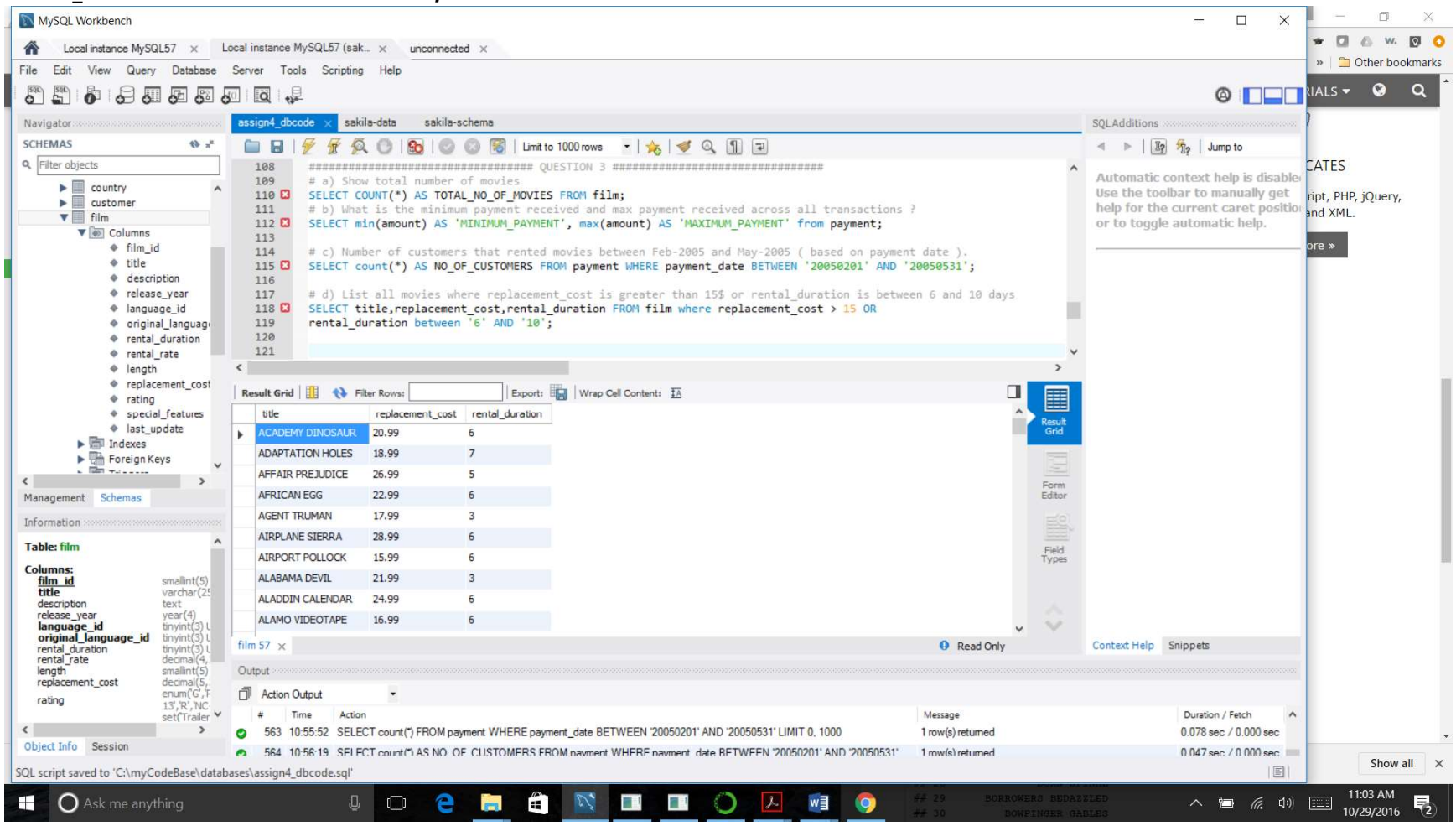


 # c) Number of customers that rented movies between Feb-2005 and May-2005 (based on payment date).

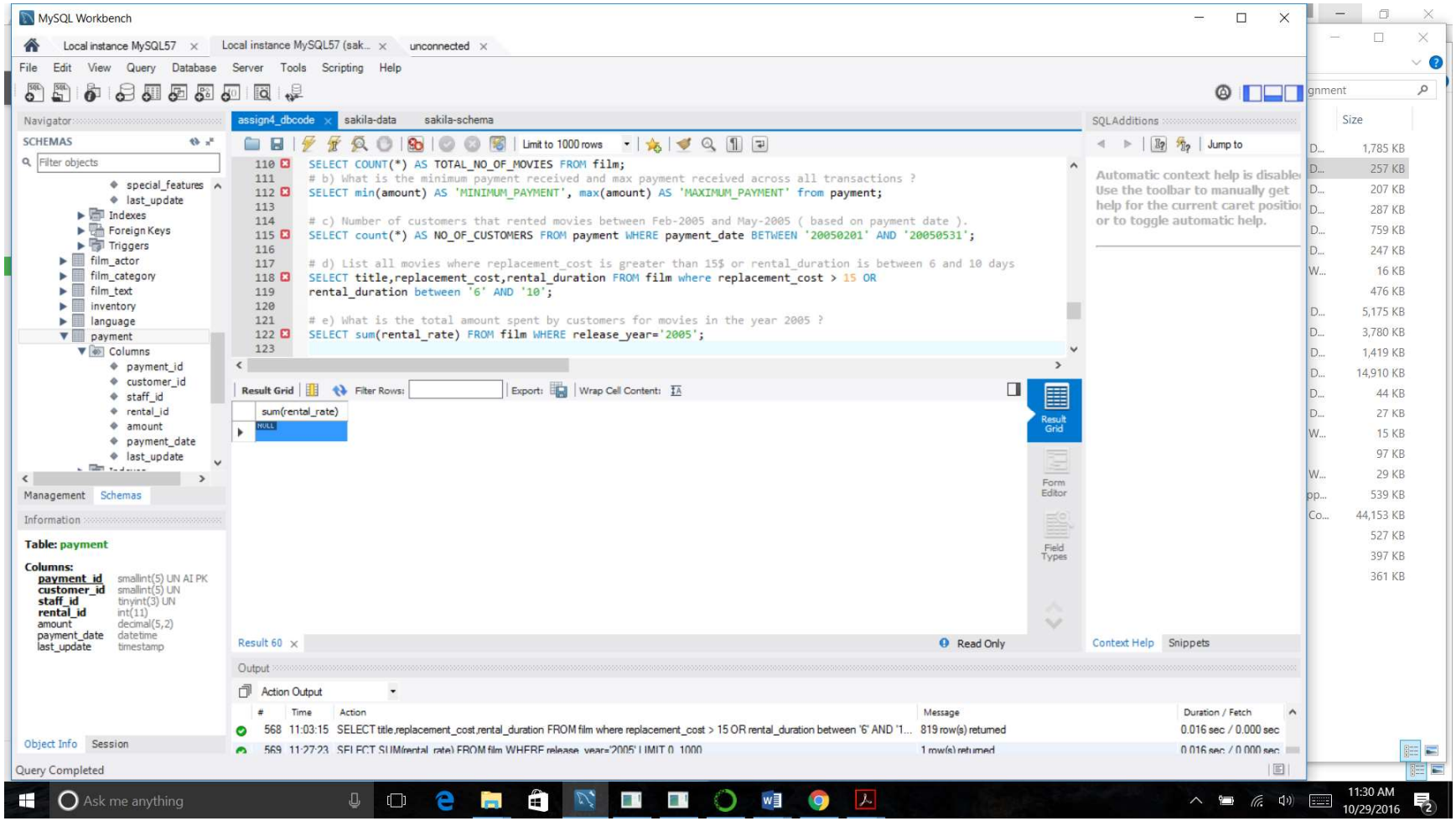
SELECT count(*) AS NO_OF_CUSTOMERS FROM payment WHERE payment_date BETWEEN '20050201' AND '20050531';



 # d) List all movies where replacement_cost is greater than 15$ or rental_duration is between 6 and 10 days
SELECT title,replacement_cost,rental_duration FROM film where replacement_cost > 15 OR rental_duration between '6' AND '10';

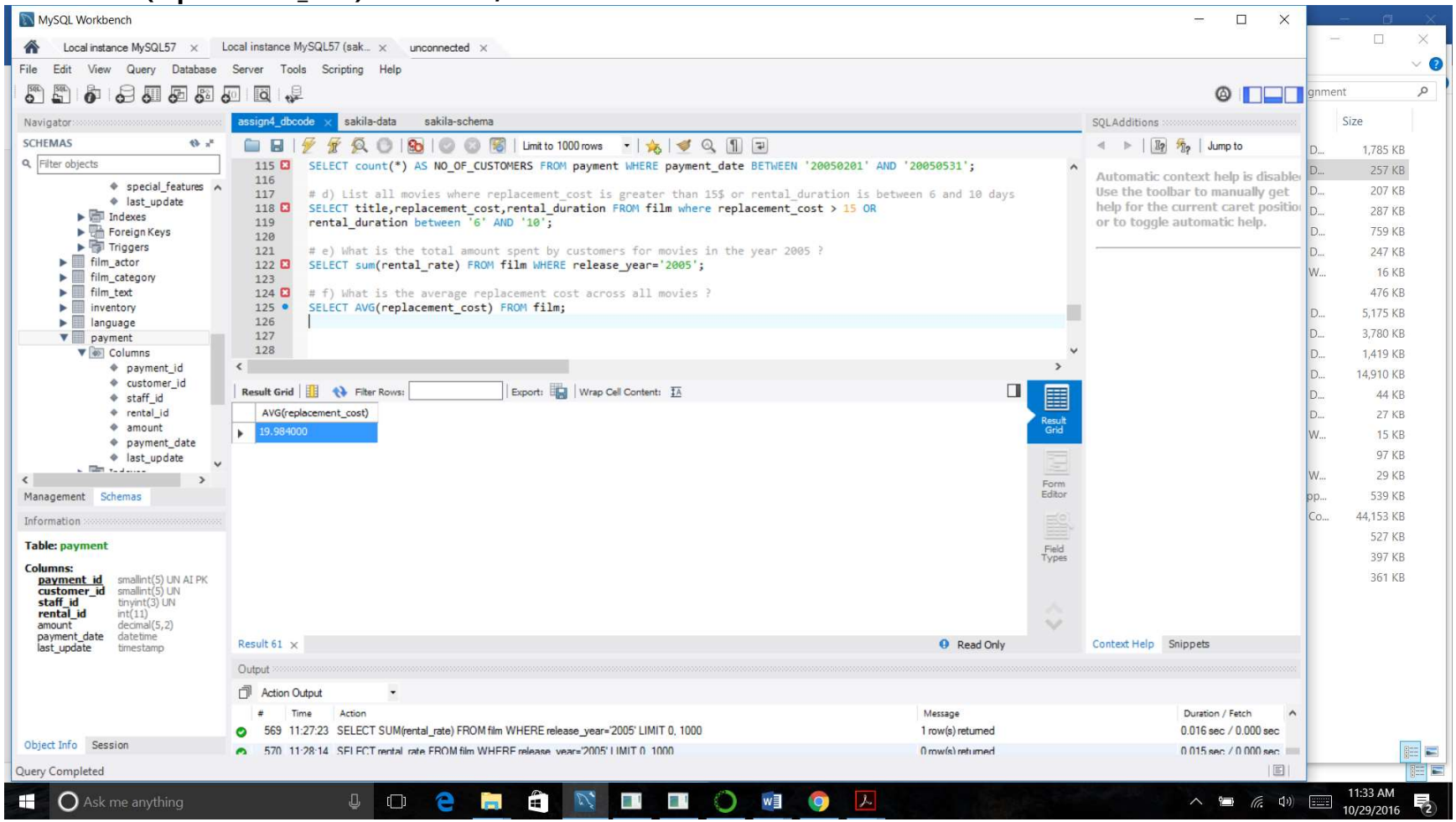


 # e) What is the total amount spent by customers for movies in the year 2005?
SELECT sum(rental_rate) FROM film WHERE release_year='2005';



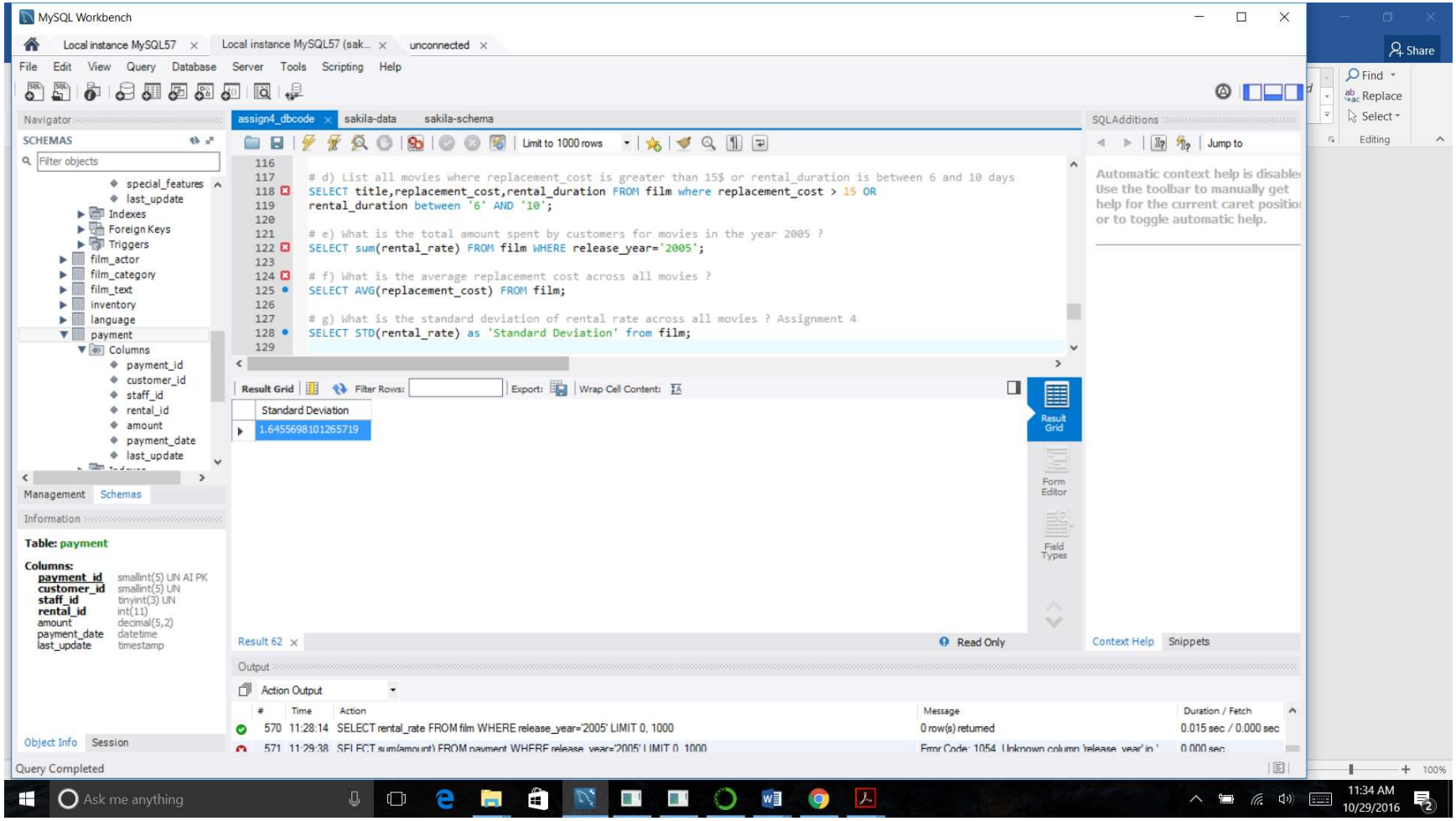
 # f) What is the average replacement cost across all movies?

SELECT AVG(replacement_cost) FROM film;



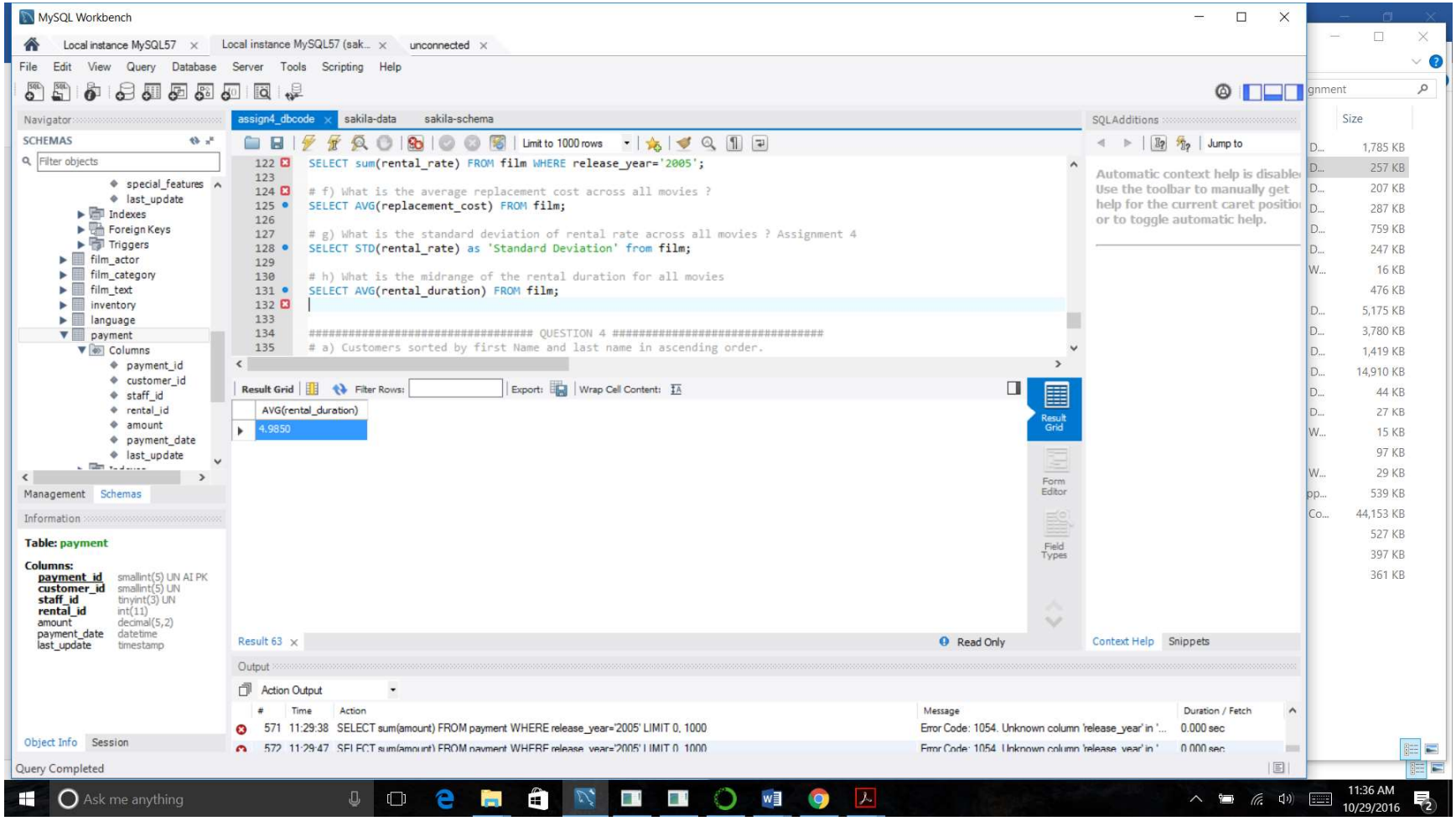
 # g) What is the standard deviation of rental rate across all movies?

SELECT STD(rental_rate) as 'Standard Deviation' from film;



 # h) What is the midrange of the rental duration for all movies

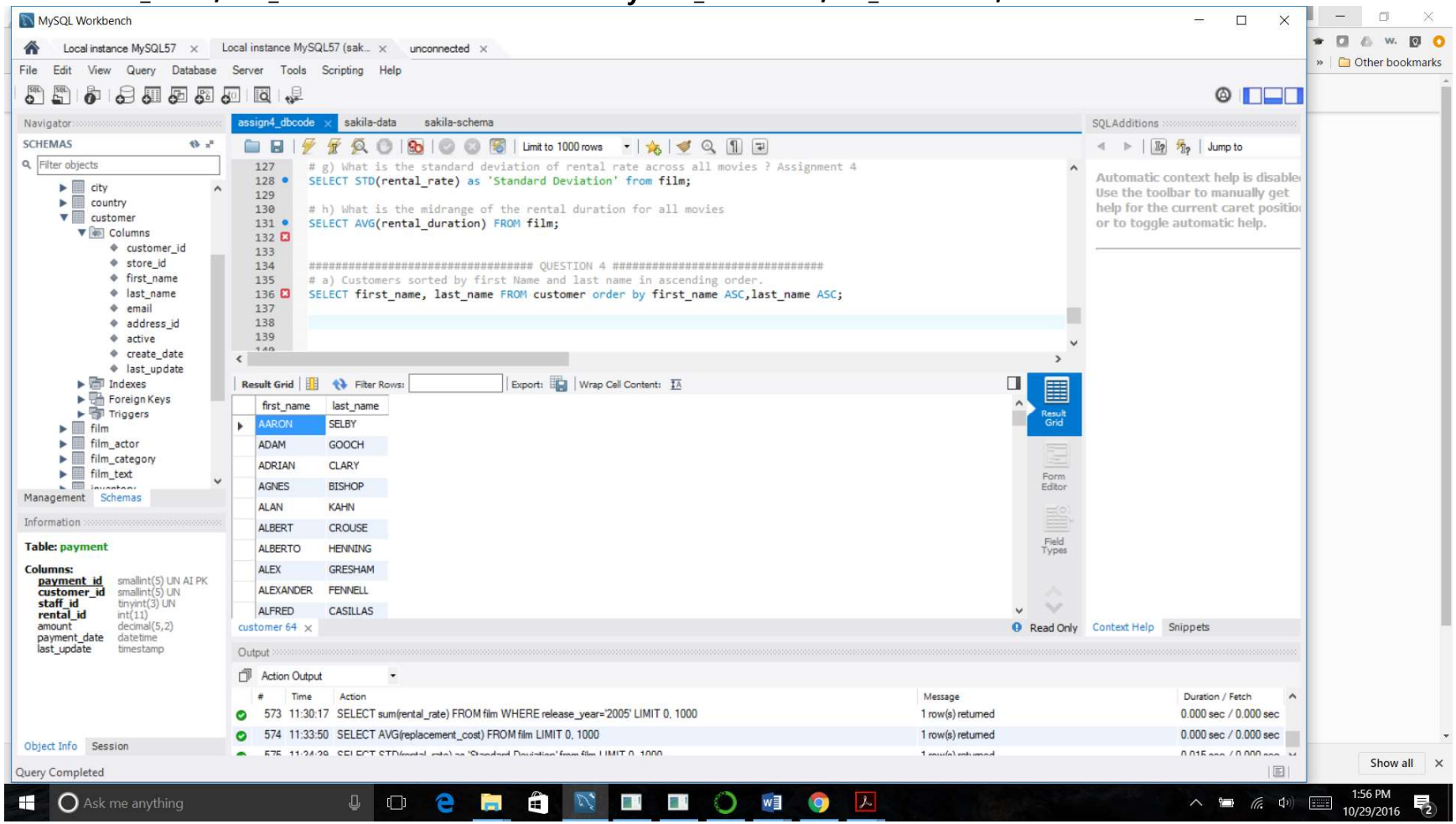
SELECT AVG(rental_duration) FROM film;



################################## QUESTION 4 ##################################

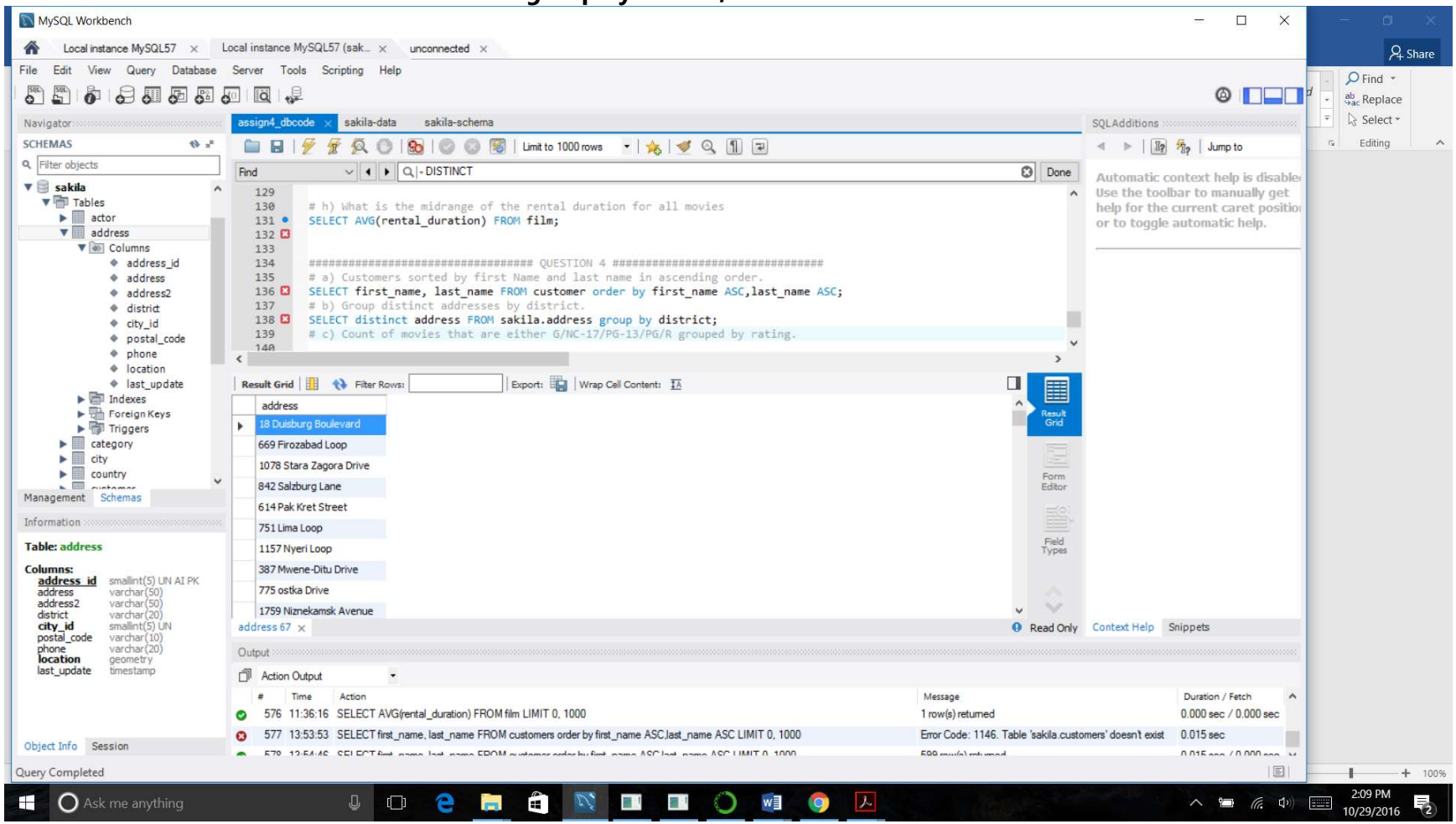
 # a) Customers sorted by first Name and last name in ascending order.

SELECT first_name, last_name FROM customer order by first_name ASC,last_name ASC;

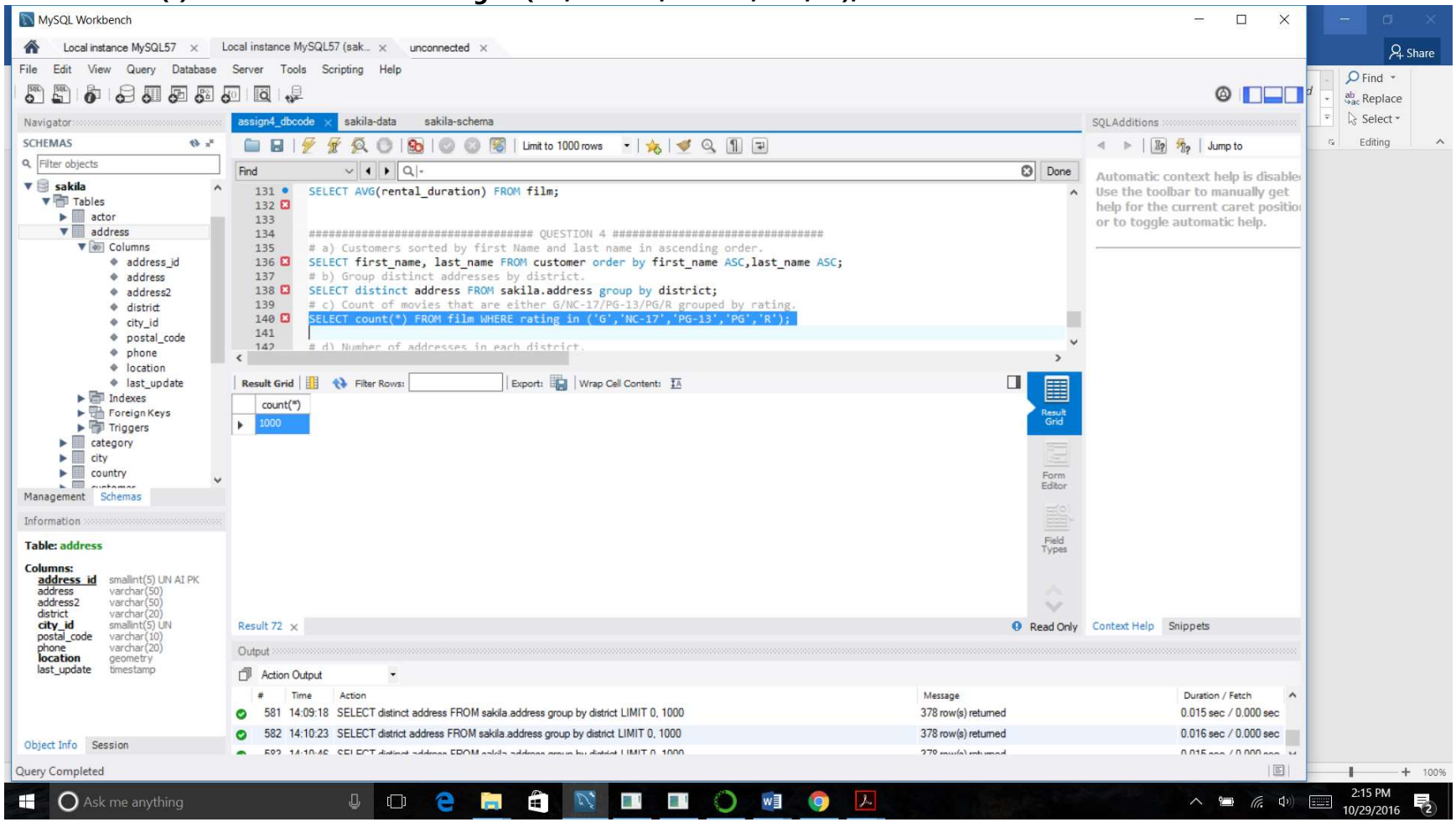


 # b) Group distinct addresses by district.

SELECT distinct address FROM sakila.address group by district;

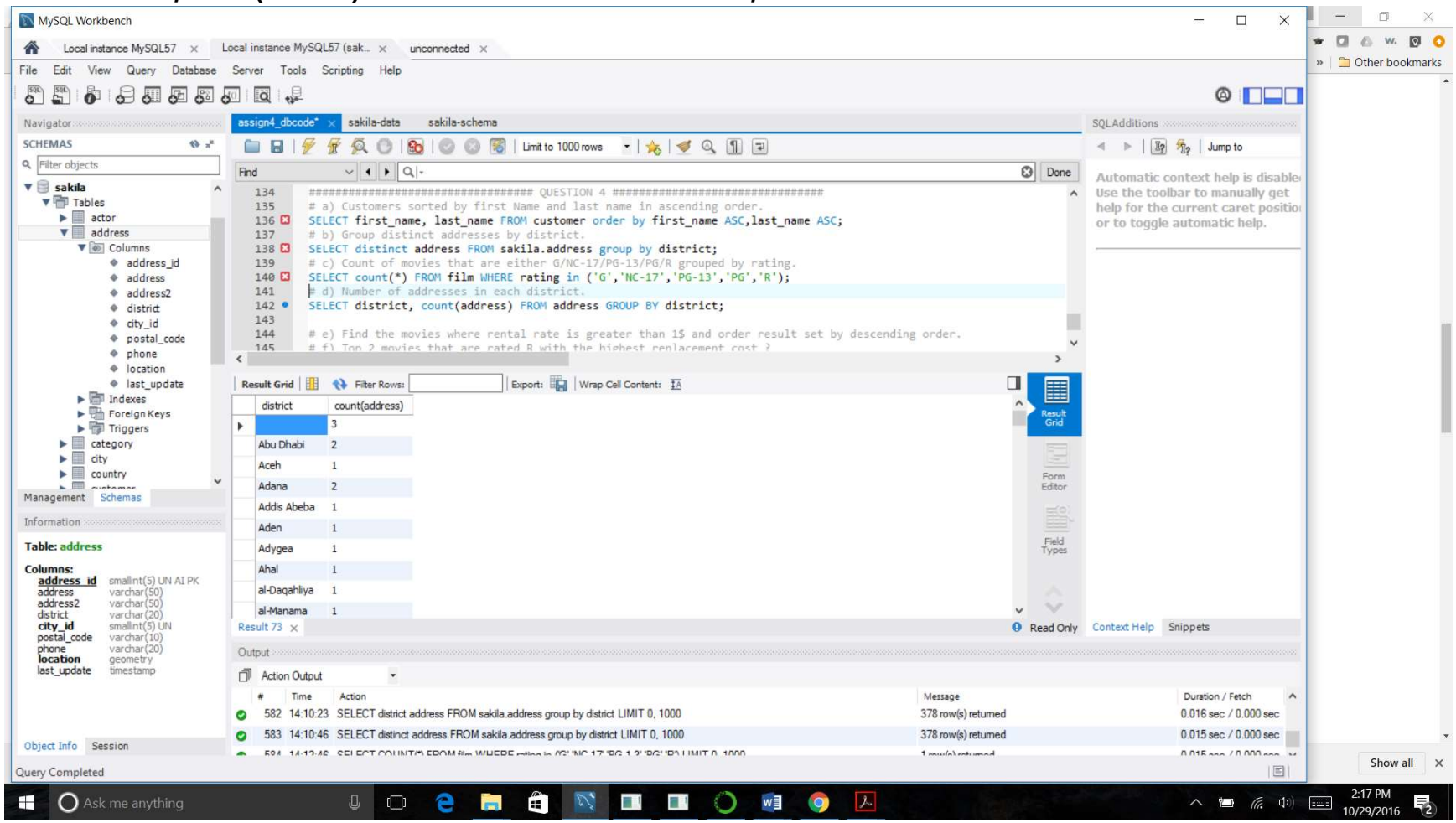


 # c) Count of movies that are either G/NC-17/PG-13/PG/R grouped by rating.
SELECT count(*) FROM film WHERE rating in ('G','NC-17','PG-13','PG','R');

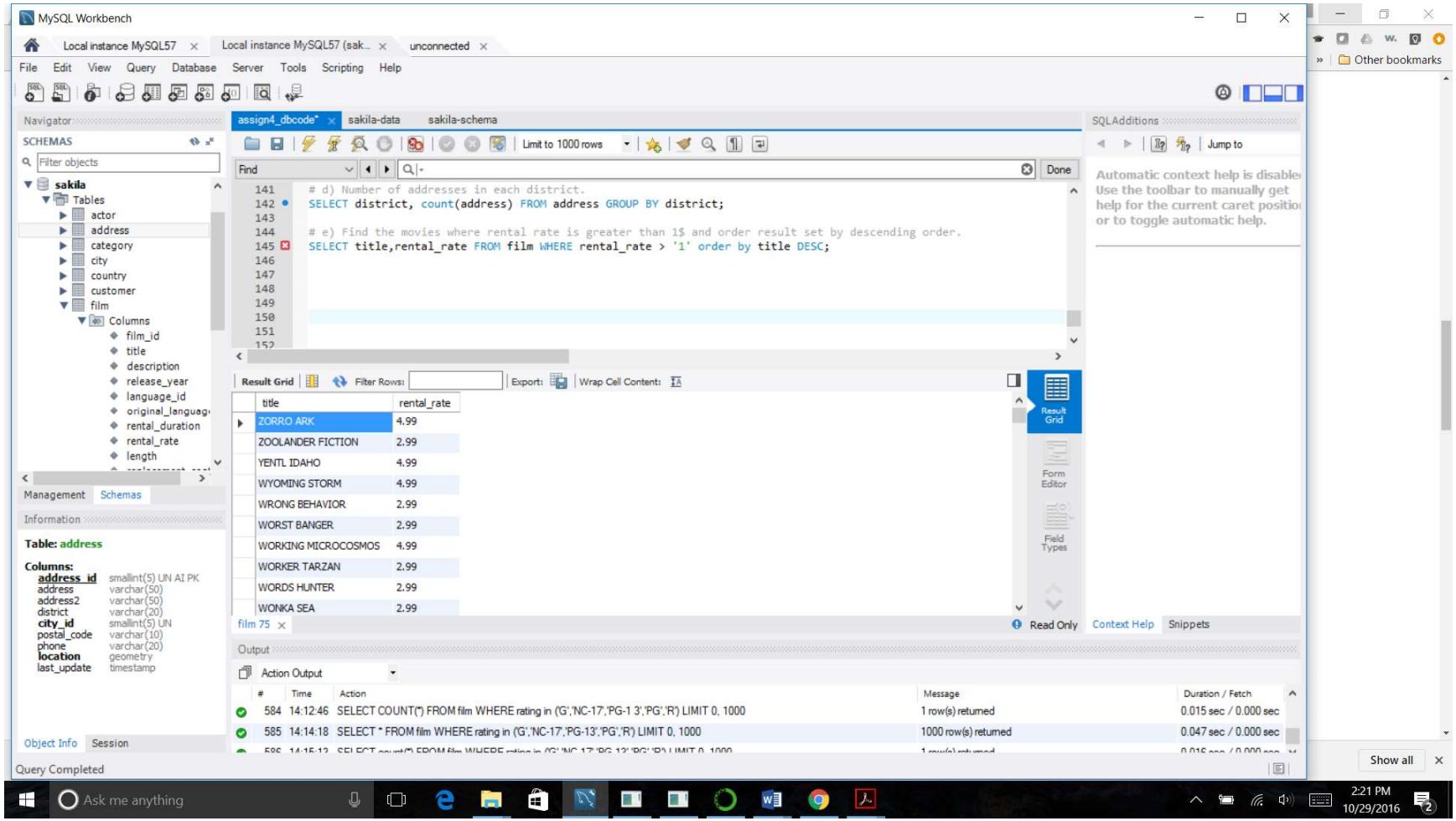


 # d) Number of addresses in each district.

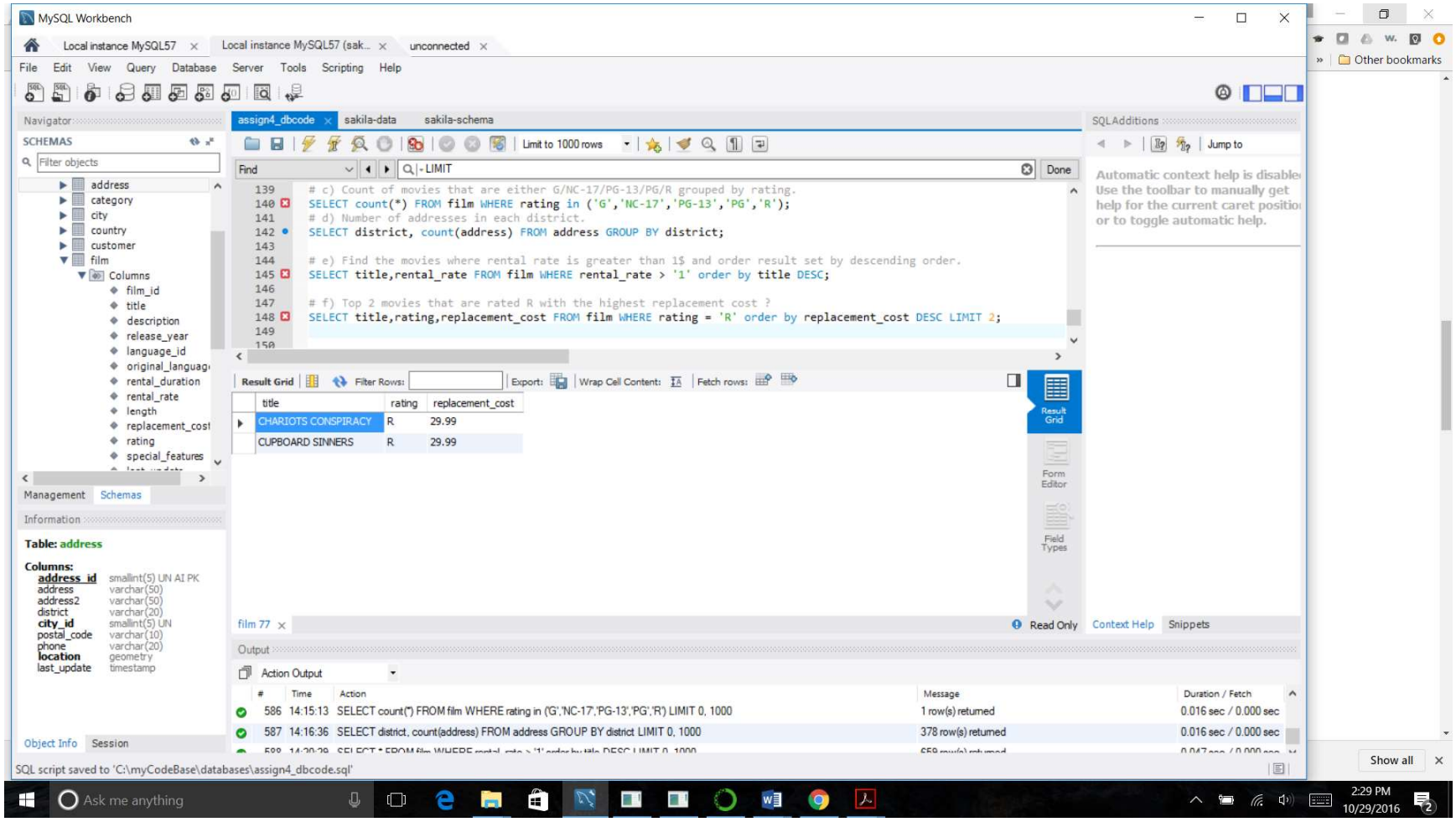
SELECT district, count(address) FROM address GROUP BY district;



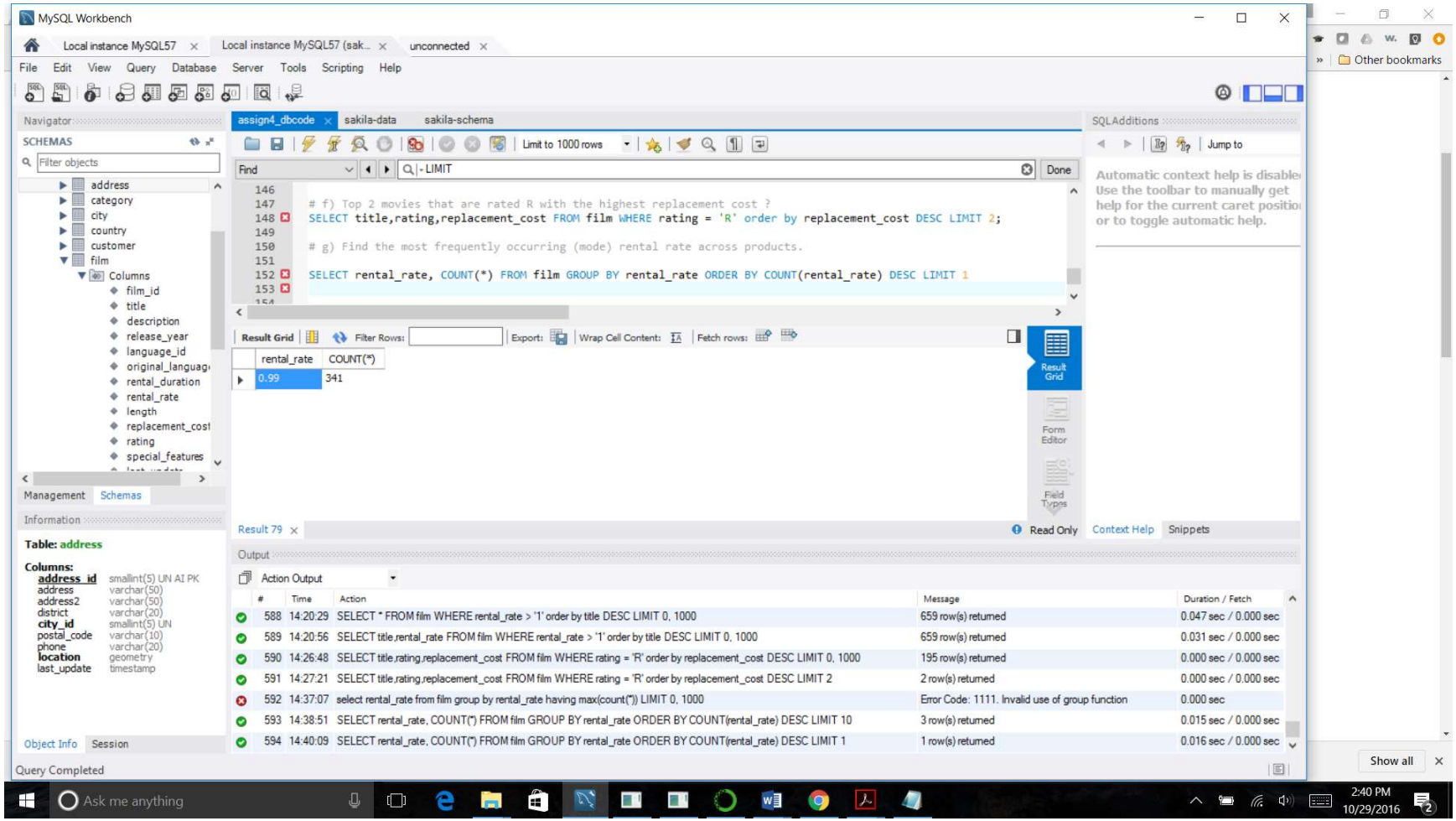
 # e) Find the movies where rental rate is greater than 1$ and order result set by descending order.
SELECT title,rental_rate FROM film WHERE rental_rate > '1' order by title DESC;



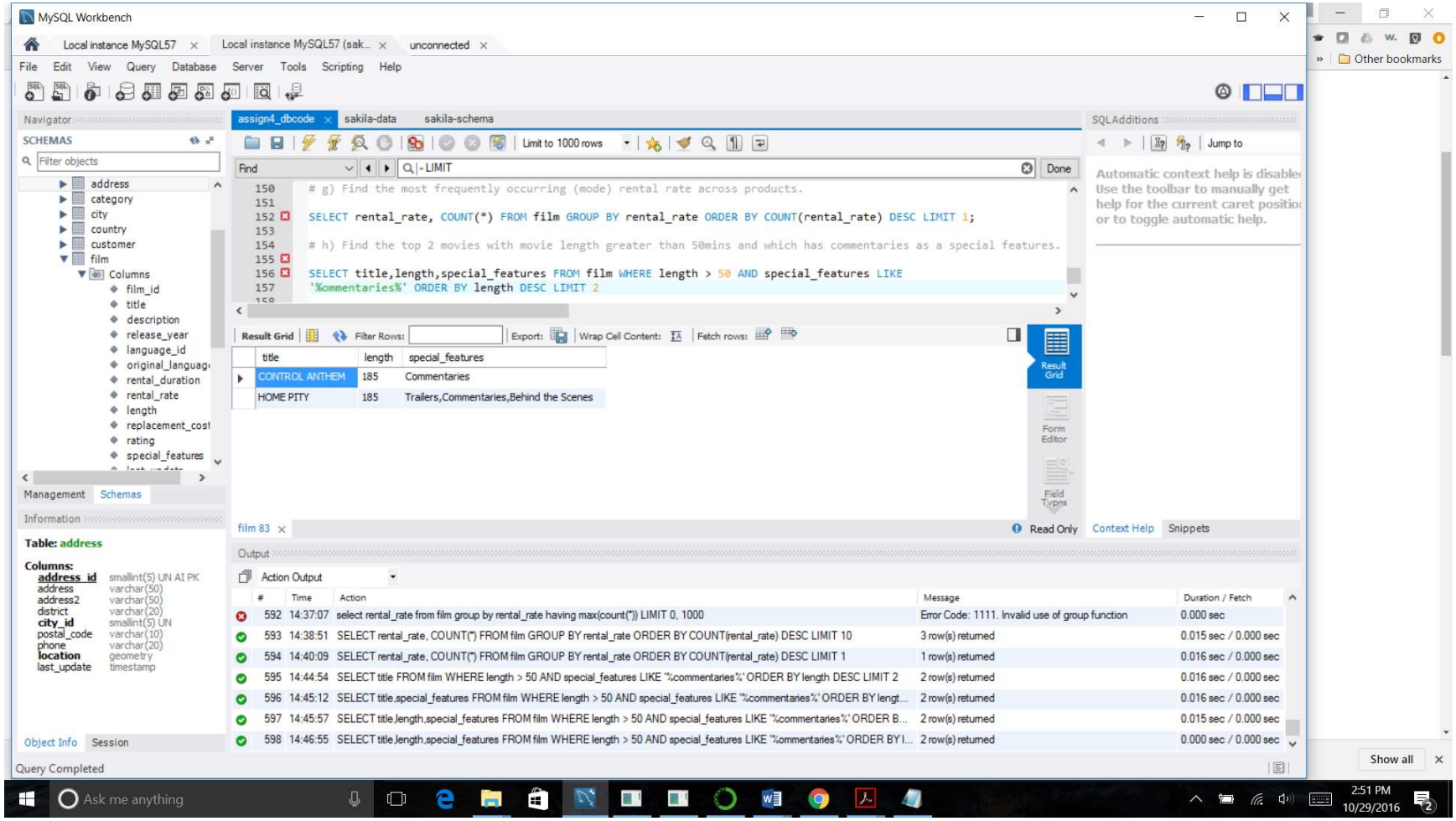
 # f) Top 2 movies that are rated R with the highest replacement cost?
SELECT title,rating,replacement_cost FROM film WHERE rating = 'R' order by replacement_cost DESC LIMIT 2;



 # g) Find the most frequently occurring (mode) rental rate across products.
SELECT rental_rate, COUNT(*) FROM film GROUP BY rental_rate ORDER BY COUNT(rental_rate) DESC LIMIT 1;



 # h) Find the top 2 movies with movie length greater than 50mins and which has commentaries as a special feature.
SELECT title,length,special_features FROM film WHERE length > 50 AND special_features LIKE '%ommentaries%' ORDER BY length DESC LIMIT 2;


MySQL Workbench
Local instance MySQL57
Local instance MySQL57 (sak...
unconnected
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
address
category
city
country
customer
film
Columns
film_id
title
description
release_year
language_id
original_languag
rental_duration
rental_rate
length
replacement_cost
rating
special_features
Management Schemas
Information
Table: address
Columns:
address_id smallint(5) UN AI PK
address varchar(50)
address2 varchar(50)
district varchar(20)
city_id smallint(5) UN
postal_code varchar(10)
phone varchar(20)
location geometry
last_update timestamp
Object Info Session
assign4_dbcode
sakila-data
sakila-schema
Limit to 1000 rows
Find
LIMIT
Done
150 # g) Find the most frequently occurring (mode) rental rate across products.
151
152 SELECT rental_rate, COUNT(*) FROM film GROUP BY rental_rate ORDER BY COUNT(rental_rate) DESC LIMIT 1;
153
154 # h) Find the top 2 movies with movie length greater than 50mins and which has commentaries as a special features.
155
156 SELECT title,length,special_features FROM film WHERE length > 50 AND special_features LIKE
157 '%ommentaries%' ORDER BY length DESC LIMIT 2
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Fetch rows:
title length special_features
CONTROL ANTHEM 185 Commentaries
HOME PITY 185 Trailers,Commentaries,Behind the Scenes
Result Grid
Form Editor
Field Types
film 83
Read Only
Context Help
Snippets
SQLAdditions
Jump to
Automatic context help is disable
Use the toolbar to manually get
help for the current caret positio
or to toggle automatic help.
Output
Action Output
Time Action Message Duration / Fetch
592 14:37:07 select rental_rate from film group by rental_rate having max(count(*)) LIMIT 0, 1000 Error Code: 1111. Invalid use of group function 0.000 sec
593 14:38:51 SELECT rental_rate, COUNT(*) FROM film GROUP BY rental_rate ORDER BY COUNT(rental_rate) DESC LIMIT 10 3 row(s) returned 0.015 sec / 0.000 sec
594 14:40:09 SELECT rental_rate, COUNT(*) FROM film GROUP BY rental_rate ORDER BY COUNT(rental_rate) DESC LIMIT 1 1 row(s) returned 0.016 sec / 0.000 sec
595 14:44:54 SELECT title FROM film WHERE length > 50 AND special_features LIKE '%commentaries%' ORDER BY length DESC LIMIT 2 2 row(s) returned 0.016 sec / 0.000 sec
596 14:45:12 SELECT title,special_features FROM film WHERE length > 50 AND special_features LIKE '%commentaries%' ORDER BY lengt... 2 row(s) returned 0.016 sec / 0.000 sec
597 14:45:57 SELECT title,length,special_features FROM film WHERE length > 50 AND special_features LIKE '%commentaries%' ORDER B... 2 row(s) returned 0.015 sec / 0.000 sec
598 14:46:55 SELECT title,length,special_features FROM film WHERE length > 50 AND special_features LIKE '%ommentaries%' ORDER BY l... 2 row(s) returned 0.000 sec / 0.000 sec
Query Completed
Other bookmarks
Show all
Ask me anything
2:51 PM
10/29/2016

 # i) List the years with more than 2 movies released.
SELECT release_year, title FROM film;
SELECT distinct(release_year), count(*) FROM film GROUP BY release_year HAVING COUNT(*) > 2;