

Database Design and Implementation

MSCA 31005

Manivassakam M

Assignment 4

QUESTION 1

a) Show the list of databases.

SHOW DATABASES;

The screenshot shows the MySQL Workbench interface with the following details:

- Title Bar:** MySQL Workbench, Local instance MySQL57 (sakila), unconnected
- Toolbar:** Standard MySQL Workbench toolbar with various icons for file operations, database management, and scripting.
- Navigator:** Shows the current schema is "sakila". It lists several databases:
 - classicmodels
 - classicmodels2
 - mydb
 - sakila
 - sys
- SQL Editor:** SQL File 1* contains the query: "SHOW DATABASES;".
- Result Grid:** Displays the output of the query:

Database
information_schema
classicmodels
classicmodels2
mydb
mysql
performance_schema
sakila
sys
- Message Area:** Shows the execution log:

Action	Time	Message	Duration / Fetch
ALTER TABLE customers DROP INDEX cityStateIndex	94 20:39:28	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.156 sec
SHOW INDEX FROM customers	95 20:39:33	2 row(s) returned	0.015 sec / 0.000 sec
SHOW DATABASES	96 23:25:23	8 row(s) returned	0.000 sec / 0.000 sec
- Status Bar:** Shows the date and time: 10/28/2016 11:32 PM

b) Select sakila database.

USE sakila

The screenshot shows the MySQL Workbench interface. In the Navigator pane, under the Schemas section, the 'sakila' schema is selected and highlighted with a blue border. A context menu is open over the 'sakila' entry, with the 'Set as Default Schema' option being the active choice, indicated by a blue selection bar. The SQL Editor pane contains the following SQL code:

```
SHOW DATABASES;
#select the sakila database where the tables needs to be created
USE sakila;
```

The Output pane at the bottom displays the execution results of the SQL statements:

Action	Time	Message	Duration / Fetch
SHOW INDEX FROM customers	95 20:39:33	2 row(s) returned	0.015 sec / 0.000 sec
SHOW DATABASES	96 23:25:23	8 row(s) returned	0.000 sec / 0.000 sec
USE sakila	97 23:33:56	0 row(s) affected	0.000 sec

c) Show all tables in the sakila database.

SHOW TABLES;

The screenshot shows the MySQL Workbench interface. In the Navigator pane, the 'sakila' schema is selected, revealing its tables: actor, address, category, city, country, customer, film, film_actor, film_category, film_text, inventory, language, payment_type. The 'Tables_in_sakila' result grid displays the same list of tables. The SQL Editor pane contains the following script:

```
1 SHOW DATABASES;
2 #select the sakila database where the tables needs to be created
3 USE sakila
4
5 #shows the list of all tables in the database
6 SHOW TABLES;
```

The 'Result Grid' tab is selected in the Results pane, showing the results of the SHOW TABLES query. The results are:

Tables_in_sakila
actor
address
category
city
country
customer
film
film_actor
film_category
film_text
inventory
language
payment_type

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;

The screenshot shows the MySQL Workbench interface. In the SQL Editor tab, the following SQL code is written:

```
1 • SHOW DATABASES;
2 #select the sakila database where the tables needs to be created
3 • USE sakila
4
5 #shows the list of all tables in the database
6 • SHOW TABLES;
7
8 #provides information on table structure
9 • DESCRIBE actor;
```

The Results Grid displays the structure of the 'actor' table:

Field	Type	Null	Key	Default	Extra
actor_id	smallint(5) unsigned	NO	PRI	HULL	auto_increment
first_name	varchar(45)	NO		HULL	
last_name	varchar(45)	NO	MUL	HULL	
last_update	timestamp	NO		CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP

e) Show the total number of records in the actor table.

SELECT COUNT(*) FROM actor;

The screenshot shows the MySQL Workbench interface. The SQL editor tab contains a script named 'assign4_dbcode' with the following content:

```
3  ** File: Assignment4.sql
4  ** Desc: Assignment 4
5  ** Author: Manivassakan Mouttayen
6  ** Date: 29th October 2016
7  ****
8  * SHOW DATABASES;
9  * #select the sakila database where the tables needs to be created
10 * USE sakila
11 *
12 * #shows the list of all tables in the database
13 * SHOW TABLES;
14 *
15 * #provides information on table structure
16 * DESCRIBE actor;
17 *
18 * #Count the no of records in the actor table
19 * SELECT COUNT(*) FROM actor;
```

The results grid shows the output of the last query:

COUNT(*)
200

The status bar at the bottom right indicates the time as 12:11 AM and the date as 10/29/2016.

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%'

The screenshot shows the MySQL Workbench interface. The SQL Editor tab contains the following code:

```
15 #provides information on table structure
16 DESCRIBE actor;
17
18 #Count the no of records in the actor table
19 SELECT COUNT(*) FROM actor;
20
21 #Select the first name and last name of all actors
22 SELECT first_name,last_name FROM actor;
23
24 # Insert your first name and middle initial ( in the last name column ) into the actors table
25 INSERT into actor VALUES (201,'MANIVASSAKAM','M','2016-10-19 04:34:33');
26 SELECT * FROM actor WHERE first_name like 'MANI%'
```

The Results Grid shows the following data:

actor_id	first_name	last_name	last_update
201	MANIVASSAKAM	M	2016-10-19 04:34:33
*	NULL	NULL	NULL

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%';

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
18 #Count the no of records in the actor table
19 SELECT COUNT(*) FROM actor;
20
21 #Select the first name and last name of all actors
22 SELECT first_name,last_name FROM actor;
23
24 # Insert your first name and middle initial ( in the last name column ) into the actors table
25 INSERT into actor VALUES (201,'MANIVASSAKAM','M','2016-10-19 04:34:33');
26
27
28 # Update your middle initial with your last name in the actors table.
29 UPDATE actor set last_name='MOUTTAYEN' where last_name='M';
30
31 SELECT * FROM actor WHERE first_name like 'MANI%'
```

The result grid shows the following data:

actor_id	first_name	last_name	last_update
201	MANIVASSAKAM	MOUTTAYEN	2016-10-29 00:29:48
*	HULL	HULL	HULL

The status bar at the bottom indicates: SQL script saved to 'C:\myCodeBase\database\assign4_dbcode.sql'.

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%';
```

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema **sakila** selected, with tables like **actor**, **address**, **category**, etc.
- SQL Editor:** Contains the following code:

```
27
28 # Update your middle initial with your last name in the actors table.
29 UPDATE actor set last_name='MOUTTAYEN' where last_name='M';
30 • SELECT * FROM actor WHERE first_name like 'MANI%';
31
32 #Delete the record from the actor table where the first name matches your first name.
33 SET SQL_SAFE_UPDATES = 0;
34 DELETE FROM actor WHERE first_name='MANIVASSAKAM';
35 • SELECT * FROM actor WHERE first_name like 'MANI%';
36
```
- Result Grid:** Displays the **actor** table with columns **actor_id**, **first_name**, **last_name**, and **last_update**. The grid shows four rows, all of which are currently null.
- Output:** Shows the execution log for the queries:

#	Time	Action	Message	Duration / Fetch
218	00:46:37	SE...	0 row(s) affected	0.016 sec
219	00:46:37	DE...	1 row(s) affected	0.000 sec
220	00:46:37	SEL...	0 row(s) returned	0.000 sec / 0.000 sec
- System Tray:** Shows the Windows taskbar with various pinned icons and the system clock indicating 12:46 AM on 10/29/2016.

```
# 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 assign4_dbcode sakila-schema sakila-data

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

Query Completed

assign4_dbcode

Limit to 1000 rows

SQL Additions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
35 • SELECT * FROM actor WHERE first_name like 'MANIK';
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: | Result Grid | Form Editor | Context Help | Snippets

payment_type_id	Type
1	Credit Card
2	Cash
3	Paypal
4	Cheque
*	HULL

payment_type 28 x

Apply Revert

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

1:00 AM 10/29/2016

Ask me anything

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

The screenshot shows the MySQL Workbench interface with the following details:

- Title Bar:** MySQL Workbench, Local instance MySQL57 (sakila), unconnected.
- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Standard MySQL Workbench toolbar icons.
- Navigator:** Schemas (classicmodels2, mydb, sakila), Tables (actor, address, category, city, country, customer, film, film_actor, film_category, film_text, inventory, language, payment_types, Views, Stored Procedures, Functions). The sakila schema is selected.
- Query Editor:** A large text area containing the following SQL code:

```
31 #Delete the record from the actor table where the first name matches your first name.
32 SET SQL_SAFE_UPDATES = 0;
33 DELETE FROM actor WHERE first_name='MANIVASSAKAM';
34 SELECT * FROM actor WHERE first_name like 'MANI%';
35 # j) Create a table payment_type with the following specifications and appropriate data types
36 # Table Name : "Payment_type"
37 # Primary Key: "payment_type_id"
38 # Column: "Type"
39 # Insert following rows in to the table: 1, "Credit Card" ; 2, "Cash"; 3, "Paypal" ; 4 , "Cheque"
40 DROP TABLE payment_type;
41 CREATE TABLE IF NOT EXISTS `sakila`.`payment_type` (
42     `payment_type_id` INT NOT NULL,
43     `Type` VARCHAR(45) NULL,
44     PRIMARY KEY (`payment_type_id`)
45 ) ENGINE = InnoDB;
46 DESCRIBE payment_type;
47 INSERT INTO payment_type VALUES (1,'Credit Card');
48 INSERT INTO payment_type VALUES (2,'Cash');
49 INSERT INTO payment_type VALUES (3,'Paypal');
50 INSERT INTO payment_type VALUES (4,'Cheque');
51 SELECT * FROM payment_type;
52
53 # k) Rename table payment_type to payment_types.
54 RENAME TABLE payment_type to payment_types;
55
56
57 # l) Drop the table payment_types.
58
59
60
```
- Output Window:** Action Output table showing the results of the executed statements:

#	Time	Action	Message	Duration / Fetch
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
236	01:02:13	RE...	0 row(s) affected	0.250 sec
- System Tray:** Shows the date and time (10/29/2016, 1:02 AM).

1) Drop the table payment_types.

DROP TABLE payment_types;

The screenshot shows the MySQL Workbench interface with the following details:

- Title Bar:** MySQL Workbench, Local instance MySQL57, Local instance MySQL57 (sak...), unconnected
- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help
- Toolbar:** Standard MySQL icons for connection, schema, table, view, procedure, function, and file operations.
- Navigator:** Shows Schemas (classicmodels2, mydb, sakila, sys) and Tables (actor, address, category, city, country, customer, film, film_actor, film_category, film_text, inventory, language) under the sakila schema.
- SQL Editor:** The main area contains a script with numbered comments and SQL statements.

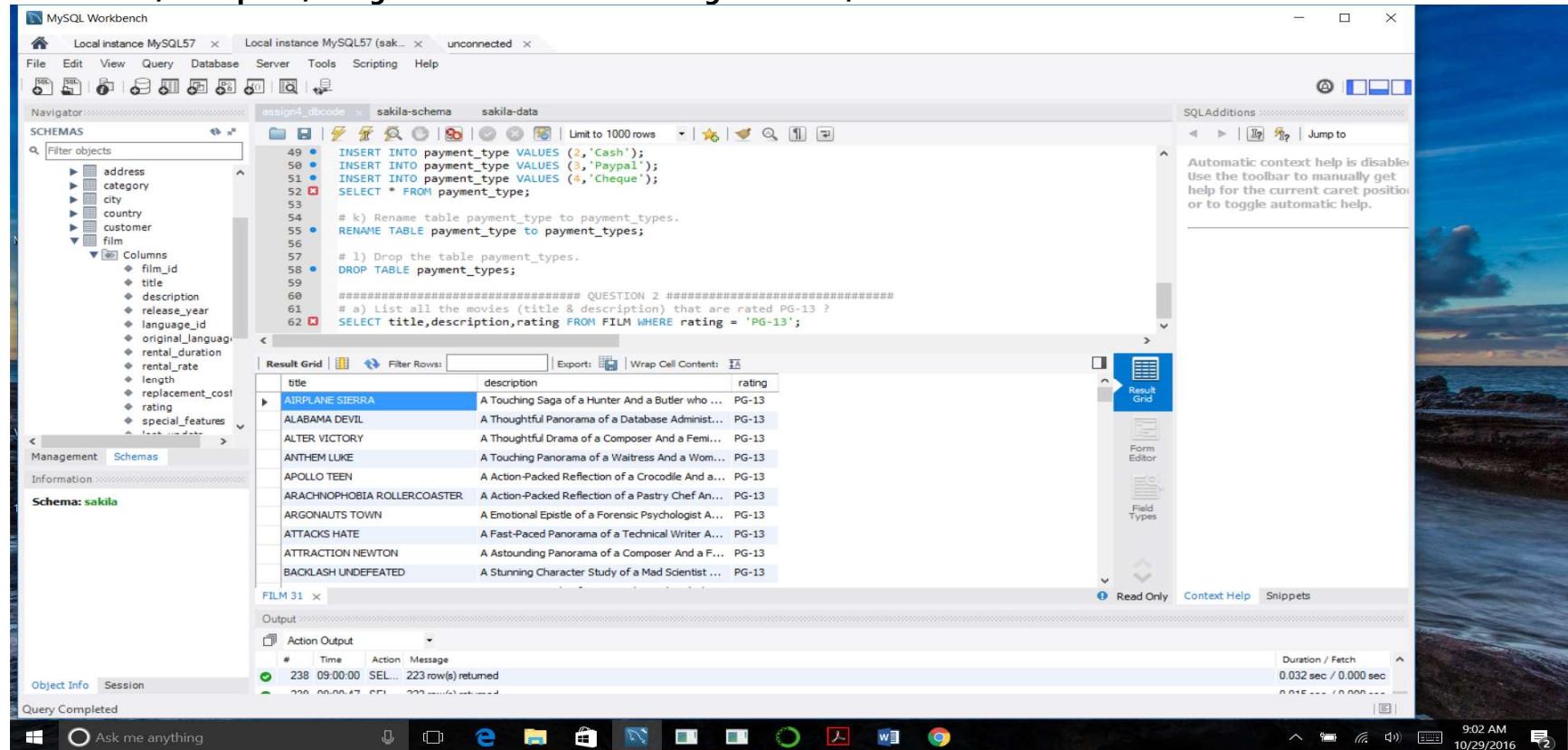
```
31 #Delete the record from the actor table where the first name matches your first name.
32 SET SQL_SAFE_UPDATES = 0;
33 DELETE FROM actor WHERE first_name='MANIVASSAKAM';
34 SELECT * FROM actor WHERE first_name like 'MANIK';
35 # j) Create a table payment_type with the following specifications and appropriate data types
36 # Table Name : "Payment_type"
37 # Primary Key: "payment_type_id"
38 # Column "Type"
39 # Insert following rows in to the table: 1, "Credit Card" ; 2, "Cash"; 3, "Paypal" ; 4 , "Cheque"
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;
53
54 # k) Rename table payment_type to payment_types.
55 RENAME TABLE payment_type to payment_types;
56
57 # l) Drop the table payment_types.
58 DROP TABLE payment_types;
59
60
```
- Output Panel:** Action Output table showing the results of the executed queries:

#	Time	Action	Message	Duration / Fetch
235	01:00:10	SEL...	4 row(s) returned	0.000 sec / 0.000 sec
236	01:02:13	RE...	0 row(s) affected	0.250 sec
237	01:03:09	DR...	0 row(s) affected	0.172 sec
- Right Panel:** SQLAdditions toolbar and a note about automatic context help.
- Bottom:** Taskbar showing various application icons and system status.

QUESTION 2

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

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



The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `sakila` with tables `address`, `category`, `city`, `country`, `customer`, and `film`. The `film` table is expanded to show columns: `film_id`, `title`, `description`, `release_year`, `language_id`, `original_language_id`, `rental_duration`, `rental_rate`, `length`, `replacement_cost`, `rating`, and `special_features`.
- SQL Editor:** Contains the following SQL code:

```
49 • INSERT INTO payment_type VALUES ('Cash');
50 • INSERT INTO payment_type VALUES ('Paypal');
51 • INSERT INTO payment_type VALUES ('Cheque');
52 ✘ SELECT * FROM payment_type;
53
54 • # k) Rename table payment_type to payment_types.
55 • RENAME TABLE payment_type TO payment_types;
56
57 • # l) Drop the table payment_types.
58 • DROP TABLE payment_types;
59
60 ##### QUESTION 2 #####
61 # a) List all the movies (title & description) that are rated PG-13 ?
62 ✘ SELECT title,description,rating FROM FILM WHERE rating = 'PG-13';
```
- Result Grid:** Displays the results of the query:

title	description	rating
AIRPLANE SIERRA	A Touching Saga of a Hunter And a Butler who ...	PG-13
ALABAMA DEVIL	A Thoughtful Panorama of a Database Administ... er's Life And a Mystery Story.	PG-13
ALTER VICTORY	A Thoughtful Drama of a Composer And a Femi... nist's Life.	PG-13
ANTHEM LUKE	A Touching Panorama of a Waitress And a Wom... an's Life.	PG-13
APOLLO TEEN	A Action-Packed Reflection of a Crocodile And a... Teen's Life.	PG-13
ARACHNOPHOBIA ROLLERCOASTER	A Action-Packed Reflection of a Pastry Chef An... d a Rollercoaster Driver.	PG-13
ARGONAUTS TOWN	A Emotional Epistle of a Forensic Psychologist A... nd a Town's History.	PG-13
ATTACKS HATE	A Fast-Paced Panorama of a Technical Writer A... nd a Hate Story.	PG-13
ATTRACTION NEWTON	A Astounding Panorama of a Composer And a F... eminist's Life.	PG-13
BACKLASH UNDEFEATED	A Stunning Character Study of a Mad Scientist ... And a Backlash.	PG-13
- Output:** Shows the action output of the query:

#	Time	Action	Message
238	09:00:00	SEL...	223 row(s) returned
239	09:00:17	SEL...	223 row(s) returned

Duration / Fetch: 0.032 sec / 0.000 sec
- Object Info:** Shows the query completed.
- Session:** Shows the user is unconnected.
- System:** Shows the system tray with icons for battery, signal, volume, and date/time (9:02 AM, 10/29/2016).

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');

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `sakila` with tables `address`, `category`, `city`, `country`, `customer`, and `film`. The `film` table is expanded to show columns: `film_id`, `title`, `description`, `release_year`, `language_id`, `original_language`, `rental_duration`, `rental_rate`, `length`, `replacement_cost`, `rating`, and `special_features`.
- SQL Editor:** Displays the following SQL code:

```
51 • INSERT INTO payment_type VALUES (4, 'Cheque');
52 ✘ SELECT * FROM payment_type;
53
54 # k) Rename table payment_type to payment_types.
55 • RENAME TABLE payment_type TO payment_types;
56
57 # l) Drop the table payment_types.
58 • DROP TABLE payment_types;
59
60 ##### QUESTION 2 #####
61 # a) List all the movies (title & description) that are rated PG-13 ?
62 ✘ SELECT title,description,rating FROM FILM WHERE rating = 'PG-13';
63 # b) List all movies that are either PG OR PG-13 using IN operator?
64 ✘ SELECT title,description,rating FROM film WHERE rating in('PG-13','PG')
```
- Result Grid:** Shows the results of the last query, listing 17 movies with their titles, descriptions, and ratings. The results are as follows:

title	description	rating
ACADEMY DINOSAUR	A Epic Drama of a Feminist And a Mad Scientist ...	PG
AGENT TRUMAN	A Intrepid Panorama of a Robot And a Boy who...	PG
AIRPLANE SIERRA	A Touching Saga of a Hunter And a Butler who ...	PG-13
ALABAMA DEVIL	A Thoughtful Panorama of a Database Administr...	PG-13
ALASKA PHANTOM	A Fanciful Saga of a Hunter And a Pastry Chef ...	PG
ALI FOREVER	A Action-Packed Drama of a Dentist And a Croc...	PG
ALTER VICTORY	A Thoughtful Drama of a Composer And a Femi...	PG-13
AMADEUS HOLY	A Emotional Display of a Pioneer And a Technica...	PG
ANTHEM LUKE	A Touching Panorama of a Waitress And a Wom...	PG-13
APOLLO TEEN	A Action-Packed Reflection of a Crocodile And a...	PG-13

Output: Shows the action output with 417 rows returned and execution time of 0.047 sec / 0.000 sec.

System Bar: Shows the Windows taskbar with various pinned icons and the system clock indicating 9:05 AM on 10/29/2016.

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;

The screenshot shows the MySQL Workbench interface. In the top-left, there are three tabs: "Local instance MySQL57", "Local instance MySQL57 (sak...)", and "unconnected". The main area contains a query editor with the following SQL code:

```
59 #####
60 ##### QUESTION 2 #####
61 # a) List all the movies (title & description) that are rated PG-13 ?
62 SELECT title,description,rating FROM film WHERE rating = 'PG-13';
63 # b) List all movies that are either PG OR PG-13 using IN operator?
64 SELECT title,description,rating FROM film WHERE rating in('PG-13','PG');
65 #c) Report all payments greater than and equal to 2$ and Less than equal to 7$?
66 # Note : write 2 separate queries conditional operator and BETWEEN keyword
67 SELECT * FROM payment WHERE amount BETWEEN 2 and 7 LIMIT 5;
```

Below the code is a "Result Grid" showing the results of the last query:

payment_id	customer_id	staff_id	rental_id	amount	payment_date	last_update
1	1	1	76	2.99	2005-05-25 11:30:37	2006-02-15 22:12:30
3	1	1	1185	5.99	2005-06-15 00:54:12	2006-02-15 22:12:30
6	1	1	1725	4.99	2005-06-16 15:18:57	2006-02-15 22:12:30
7	1	1	2308	4.99	2005-06-18 08:41:48	2006-02-15 22:12:30
9	1	1	3284	3.99	2005-06-21 06:24:45	2006-02-15 22:12:30
*	HULL	HULL	HULL	HULL	HULL	HULL

On the right side, there is a "SQLAdditions" panel with a "Result Grid" tab selected. Below the result grid is an "Output" pane showing the following log entries:

#	Time	Action	Message	Duration / Fetch
540	09:55:53	COMMIT	0 row(s) affected	0.047 sec
541	09:55:53	SET SQL_MODE=@OLD_SQL_MODE	0 row(s) affected	0.000 sec
542	09:55:53	SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS	0 row(s) affected	0.000 sec
543	09:55:53	SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS	0 row(s) affected	0.000 sec
544	09:57:15	SHOW TABLES	23 row(s) returned	0.000 sec / 0.000 sec
545	09:59:36	SELECT * FROM payment WHERE amount BETWEEN 2 and 7 LIMIT 5	5 row(s) returned	0.016 sec / 0.000 sec

Using Conditional Operator

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

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema **sakila** with tables like actor, address, category, city, country, customer, film, film_actor, film_category, film_text, inventory, language, payment, rental, staff, store.
- SQL Editor:** Contains the following SQL code:

```
61 # a) List all the movies (title & description) that are rated PG-13 ?
62 SELECT title,description,rating FROM FILM WHERE rating = 'PG-13';
63 # b) List all movies that are either PG OR PG-13 using IN operator?
64 SELECT title,description,rating FROM film WHERE rating in('PG-13','PG');
65 #c) Report all payments greater than and equal to 2$ and Less than equal to 7$ ?
66 # Note : write 2 separate queries conditional operator and BETWEEN keyword
67 SELECT * FROM payment WHERE amount BETWEEN 2 and 7 LIMIT 5;
68 # Using conditional operator
69 SELECT * FROM payment WHERE amount >= 2 and amount<= 7;
70 |
```
- Result Grid:** Displays the results of the query `SELECT * FROM payment WHERE amount >= 2 and amount<= 7;`. The results are:

payment_id	customer_id	staff_id	rental_id	amount	payment_date	last_update
1	1	1	76	2.99	2005-05-25 11:30:37	2006-02-15 22:12:30
3	1	1	1185	5.99	2005-06-15 00:54:12	2006-02-15 22:12:30
6	1	1	1725	4.99	2005-06-16 15:18:57	2006-02-15 22:12:30
7	1	1	2308	4.99	2005-06-18 08:41:48	2006-02-15 22:12:30
9	1	1	3284	3.99	2005-06-21 06:24:45	2006-02-15 22:12:30
10	1	2	4526	5.99	2005-07-08 03:17:05	2006-02-15 22:12:30
11	1	1	4611	5.99	2005-07-08 07:33:56	2006-02-15 22:12:30
12	1	1	5744	4.99	2005-07-09 13:24:07	2006-02-15 22:12:30
- Output:** Shows the action output of the query execution:

#	Time	Action	Message	Duration / Fetch
541	09:55:53	SET SQL_MODE=@OLD_SQL_MODE	0 row(s) affected	0.000 sec
542	09:55:53	SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS	0 row(s) affected	0.000 sec
543	09:55:53	SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS	0 row(s) affected	0.000 sec
544	09:57:15	SHOW TABLES	23 row(s) returned	0.000 sec / 0.000 sec
545	09:59:36	SELECT * FROM payment WHERE amount BETWEEN 2 and 7 LIMIT 5	5 row(s) returned	0.016 sec / 0.000 sec
546	10:04:48	SELECT * FROM payment WHERE amount >= 2 and amount<= 7 LIMIT 0, 1000	1000 row(s) returned	0.047 sec / 0.016 sec

d) List all addresses that have phone number that contain digits 589,

#Note : write 3 different queries

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

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains three SQL statements:
 - #c) Report all payments greater than and equal to 2\$ and Less than equal to 7\$?
 - SELECT * FROM payment WHERE amount BETWEEN 2 and 7 LIMIT 5;
 - # d) List all addresses that have phone number that contain digits 589, start with 140 or end with 589 #
#Note : write 3 different queries
SELECT * FROM address WHERE phone like '%589%';
- Result Grid:** Displays the results of the third query, showing 6 rows of address data:

address_id	address	address2	district	city_id	postal_code	phone	location	last_update
4	1411 Lilydale Drive	HULL	QLD	576	6172235589	BLOB	2014-09-25 22:30:09	
153	782 Mosul Street		Massachusetts	94	25545	885899703621	BLOB	2014-09-25 22:33:46
333	1860 Taguig Loop		West Java	119	59550	38158430589	BLOB	2014-09-25 22:13:32
388	368 Hunuco Boulevard		Namibe	360	17165	106439158941	BLOB	2014-09-25 22:30:03
492	185 Mannheim Lane		Stavropol	408	23661	589377568313	BLOB	2014-09-25 22:32:56
504	1 Valle de Santiago Avenue		Apulia	93	86208	465897838272	BLOB	2014-09-25 22:31:43

- Output:** Shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
543	09:55:53	SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS	0 row(s) affected	0.000 sec
544	09:57:15	SHOW TABLES	23 row(s) returned	0.000 sec / 0.000 sec
545	09:59:36	SELECT * FROM payment WHERE amount BETWEEN 2 and 7 LIMIT 5	5 row(s) returned	0.016 sec / 0.000 sec
546	10:04:48	SELECT * FROM payment WHERE amount >= 2 and amount<= 7 LIMIT 0, 1000	1000 row(s) returned	0.047 sec / 0.016 sec
547	10:08:28	SELECT * FROM address WHERE phone like '%589%'	Error Code: 1064. You have an error in your SQL syntax...	0.000 sec
548	10:08:50	SELECT * FROM address WHERE phone like '%589%' LIMIT 0, 1000	6 row(s) returned	0.063 sec / 0.000 sec
- System Bar:** Shows the Windows taskbar with various pinned icons and the system clock indicating 10:10 AM on 10/29/2016.

start with 140

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

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `sakila` with tables `actor`, `address`, and `category`.
- SQL Editor:** Displays the following SQL code:

```
66 # Note : write 2 separate queries conditional operator and BETWEEN keyword
67 • SELECT * FROM payment WHERE amount BETWEEN 2 and 7 LIMIT 5;
68 # Using conditional operator
69 ✘ SELECT * FROM payment WHERE amount >= 2 and amount<= 7;
70
71 # d) List all addresses that have phone number that contain digits 589, start with 140 or end with 589 #
72 #Note : write 3 different queries
73 ✘ SELECT * FROM address WHERE phone like '%589%';
74 #start with 140
75 • SELECT * FROM address WHERE phone like '140%';
```
- Result Grid:** Shows the results of the query `SELECT * FROM address WHERE phone like '140%'`.

address_id	address	address2	district	city_id	postal_code	phone	location	last_update
3	23 Workhaven Lane	HULL	Alberta	300	14033335568	BLUE	2014-09-25 22:30:27	
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL
- Action Output:** Shows the history of actions taken in the session.

#	Time	Action	Message	Duration / Fetch
544	09:57:15	SHOW TABLES	23 row(s) returned	0.000 sec / 0.000 sec
545	09:59:36	SELECT * FROM payment WHERE amount BETWEEN 2 and 7 LIMIT 5	5 row(s) returned	0.016 sec / 0.000 sec
546	10:04:48	SELECT * FROM payment WHERE amount >= 2 and amount<= 7 LIMIT 0, 1000	1000 row(s) returned	0.047 sec / 0.016 sec
547	10:08:28	SELECT * FROM address WHERE phone like '%589%'	Error Code: 1064. You have an error in your SQL syntax...	0.000 sec
548	10:08:50	SELECT * FROM address WHERE phone like '%589%' LIMIT 0, 1000	6 row(s) returned	0.063 sec / 0.000 sec
549	10:15:45	SELECT * FROM address WHERE phone like '140%' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
- System Bar:** Shows the Windows taskbar with various pinned icons and the system clock indicating 10:15 AM on 10/29/2016.

end with 589

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

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains three SQL statements:
 - Line 69: `SELECT * FROM payment WHERE amount >= 2 and amount<= 7;
 - Line 73: `#d) List all addresses that have phone number that contain digits 589, start with 140 or end with 589 #`
`#Note : write 3 different queries`
`SELECT * FROM address WHERE phone like '%589%';`
 - Line 75: `SELECT * FROM address WHERE phone like '140%';`
#End with 589
 - Line 77: `SELECT * FROM address WHERE phone like '%589';`
- Result Grid:** Displays the results of the last query, showing two rows of address data:

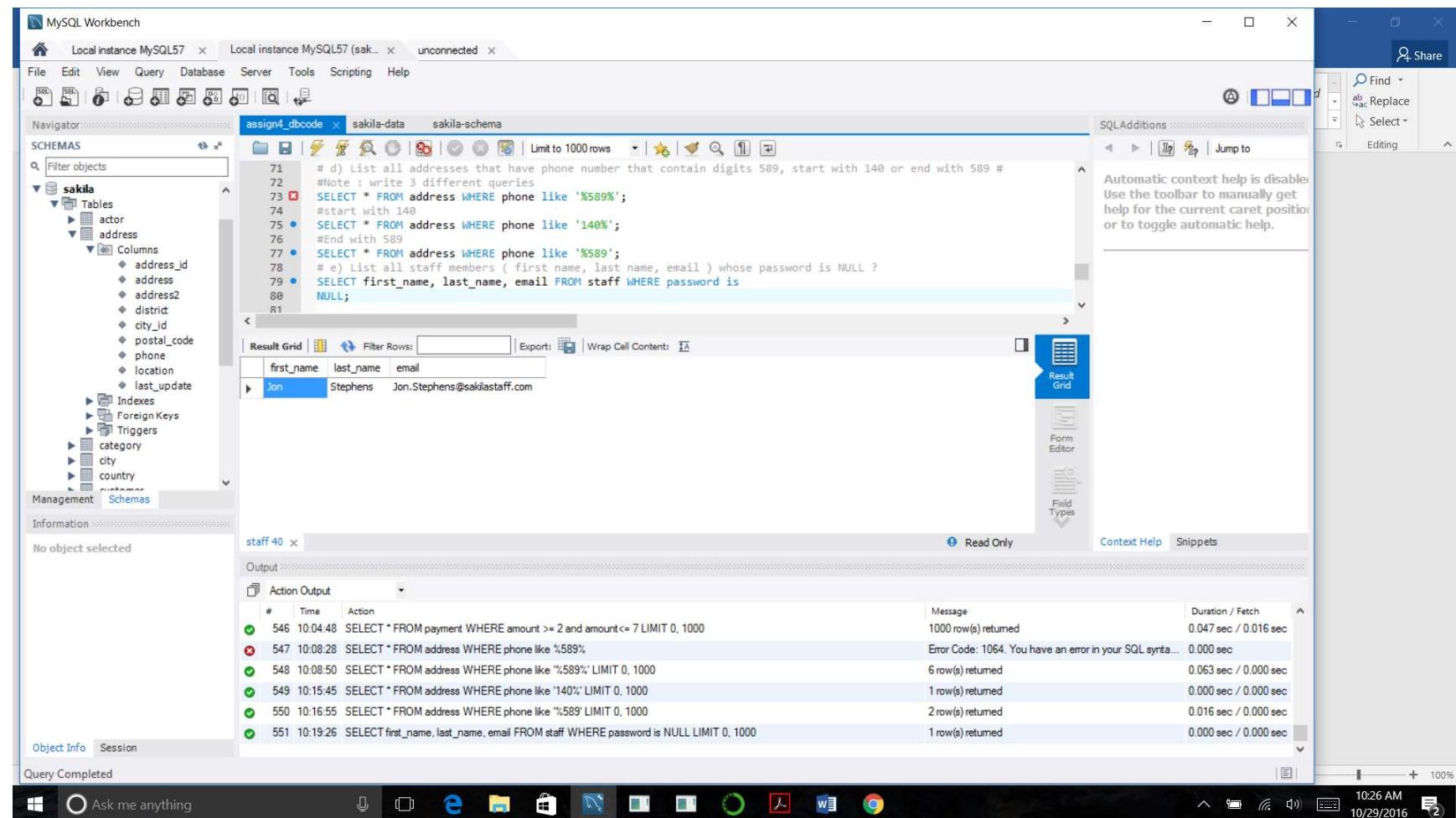
address_id	address	address2	district	city_id	postal_code	phone	location	last_update
4	1411Lillydale Drive	NULL	QLD	576	6172235589	BLOB	2014-09-25 22:30:09	
333	1860 Taguig Loop		West Java	119	59550	38158430589	BLOB	2014-09-25 22:31:32

- Action Output:** Shows the execution history of the queries:

#	Time	Action	Message	Duration / Fetch
545	09:59:36	SELECT * FROM payment WHERE amount BETWEEN 2 and 7 LIMIT 5	5 row(s) returned	0.016 sec / 0.000 sec
546	10:04:48	SELECT * FROM payment WHERE amount >= 2 and amount<= 7 LIMIT 0, 1000	1000 row(s) returned	0.047 sec / 0.016 sec
547	10:08:28	SELECT * FROM address WHERE phone like '%589%'	Error Code: 1064. You have an error in your SQL syntax...	0.000 sec
548	10:08:50	SELECT * FROM address WHERE phone like "%589%" LIMIT 0, 1000	6 row(s) returned	0.063 sec / 0.000 sec
549	10:15:45	SELECT * FROM address WHERE phone like '140%' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
550	10:16:55	SELECT * FROM address WHERE phone like "%589%" LIMIT 0, 1000	2 row(s) returned	0.016 sec / 0.000 sec
- System Bar:** Shows the Windows taskbar with various icons and the system clock.

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;



The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema **sakila** with tables **actor**, **address**, and **category**.
- SQL Editor:** Displays the following SQL code:

```
71 # d) List all addresses that have phone number that contain digits 589, start with 140 or end with 589 #
72 #Note : write 3 different queries
73 SELECT * FROM address WHERE phone like '%589%';
74 #start with 140
75 SELECT * FROM address WHERE phone like '140%';
76 #End with 589
77 SELECT * FROM address WHERE phone like '%589';
78 # e) List all staff members ( first name, last name, email ) whose password is NULL ?
79 SELECT first_name, last_name, email FROM staff WHERE password is
80 NULL;
81
```
- Result Grid:** Shows the result of the final query:

first_name	last_name	email
Jon	Stephens	Jon.Stephens@sakilastaff.com
- Output:** Shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
546	10:04:48	SELECT * FROM payment WHERE amount >= 2 and amount<= 7 LIMIT 0, 1000	1000 row(s) returned	0.047 sec / 0.016 sec
547	10:08:28	SELECT * FROM address WHERE phone like '%589%'	Error Code: 1064. You have an error in your SQL syntax...	0.000 sec
548	10:08:50	SELECT * FROM address WHERE phone like "%589%" LIMIT 0, 1000	6 row(s) returned	0.063 sec / 0.000 sec
549	10:15:45	SELECT * FROM address WHERE phone like '140%' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
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

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;
```

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema for the **sakila** database, including tables like actor, address, category, city, country, customer, and film.
- SQL Editor:** Displays the SQL query:

```
74 #start with 140  
75 • SELECT * FROM address WHERE phone like '140%';  
76 #End with 589  
77 • SELECT * FROM address WHERE phone like '%589';  
78 # e) List all staff members ( first_name, last_name, email ) whose password is NULL ?  
79 • SELECT first_name, last_name, email FROM staff WHERE password is  
NULL;  
80 # f) Select all films that have title names like ZOO and rental duration greater than or equal to 4  
81 • SELECT title,rental_duration  
82 FROM film WHERE title like '%ZOO%' AND rental_duration >= 4;  
83  
84
```
- Result Grid:** Shows the results of the query:

title	rental_duration
MEMENTO ZOOLANDER	4
UNFORGIVEN ZOOLANDER	7
ZOOLANDER FICTION	5
- Table Browser:** Shows the structure of the **film** table, including columns like **film_id**, **title**, **description**, **release_year**, **language_id**, **original_language_id**, **rental_duration**, **rental_rate**, **length**, **replacement_cost**, **rating**, and **special_features**.
- Output Window:** Displays the execution log for the query:

Action	Time	Message	Duration / Fetch
548 10:08:50	SELECT * FROM address WHERE phone like "%589%" LIMIT 0, 1000	6 row(s) returned	0.063 sec / 0.000 sec
549 10:15:45	SELECT * FROM address WHERE phone like '140%' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
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

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;

The screenshot shows the MySQL Workbench interface with a query editor window open. The query being run is:

```
79 • SELECT first_name, last_name, email FROM staff WHERE password is
80 NULL;
81 # f) Select all films that have title names like ZOO and rental duration greater than or equal to 4
82 SELECT title, rental_duration
83 FROM film WHERE title like '%ZOO%' AND rental_duration >= 4;
84 # g) Display addresses as N/A when the address2 field is NULL # Note : use IF and CASE statements
85 SELECT
86 IF (address2 is null, 'N/A', address) AS address
87 FROM address LIMIT 10;
88
89 # h) What is the cost of renting the movie ACADEMY DINOSAUR for 2 weeks ? # Note : use of column alias
```

The results grid shows the output of the query:

address
N/A
N/A
N/A
N/A
1913 Hanoi Way
1121 Loja Avenue
692 Joliet Street
1566 Trend Manor

The status bar at the bottom right indicates the time is 10:32 AM and the date is 10/29/2016.

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: assign4_dbcode sakila-data sakila-schema

SCHEMAS: Filter objects

sakila Tables: actor address category city country customer film

Columns: film_id title description release_year language_id original_language rental_duration rental_rate length replacement_cost rating

Result Grid: address

Output: Action Output

SQLAdditions: Find Replace Select Jump to Editing

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid Form Editor Field Types

Context Help Snippets

Read Only

Object Info Session

Query Completed

10:34 AM 10/29/2016

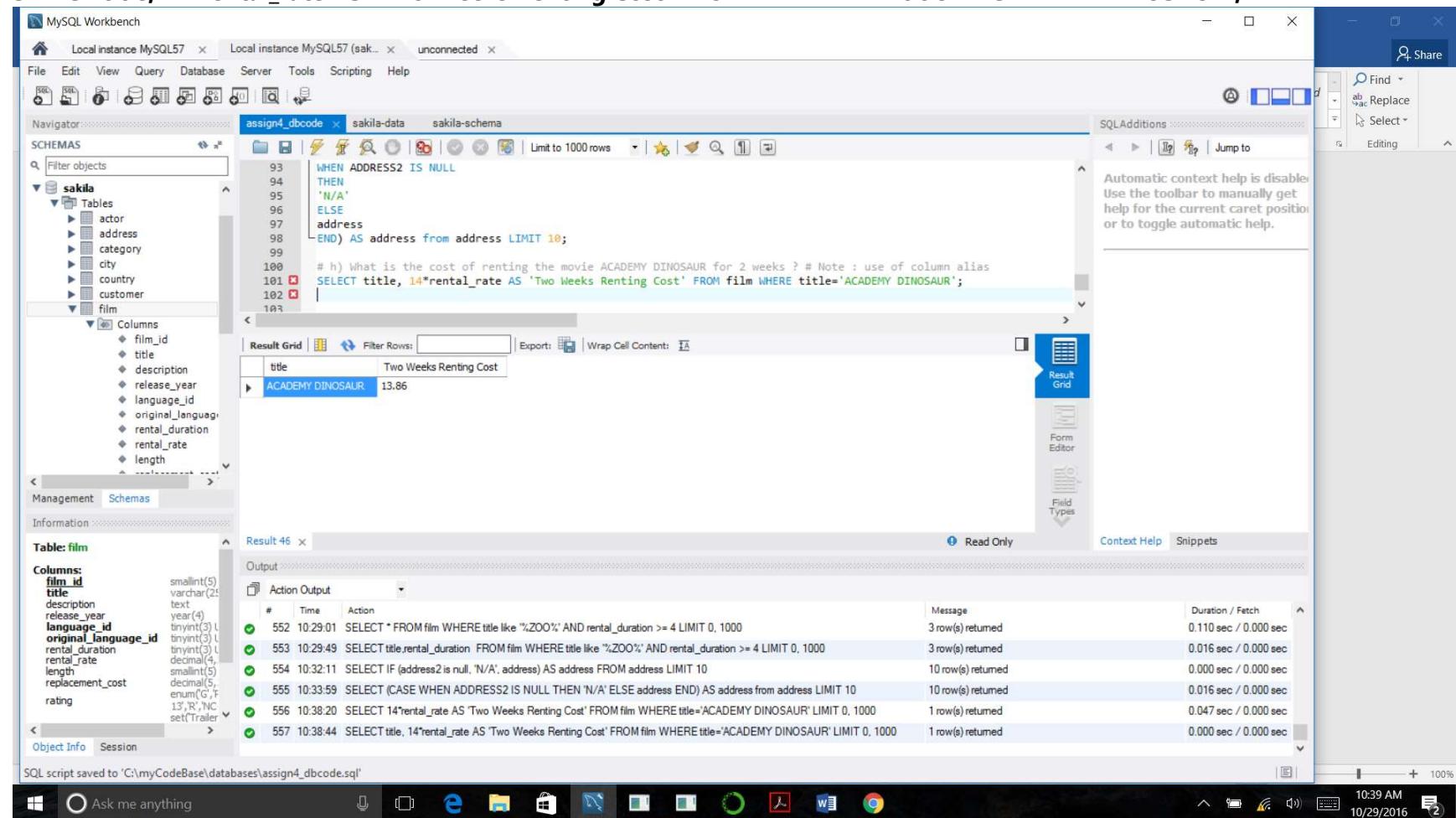
```
# Using CASE Statement
SELECT
(CASE
WHEN ADDRESS2 IS NULL
THEN
'N/A'
ELSE
address
END) AS address from address LIMIT 10;
```

address
N/A
N/A
N/A
N/A
1913 Hanoi Way
1121 Loja Avenue
692 Joliet Street
156A Trnval Manor

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

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';



The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Displays the SQL query:

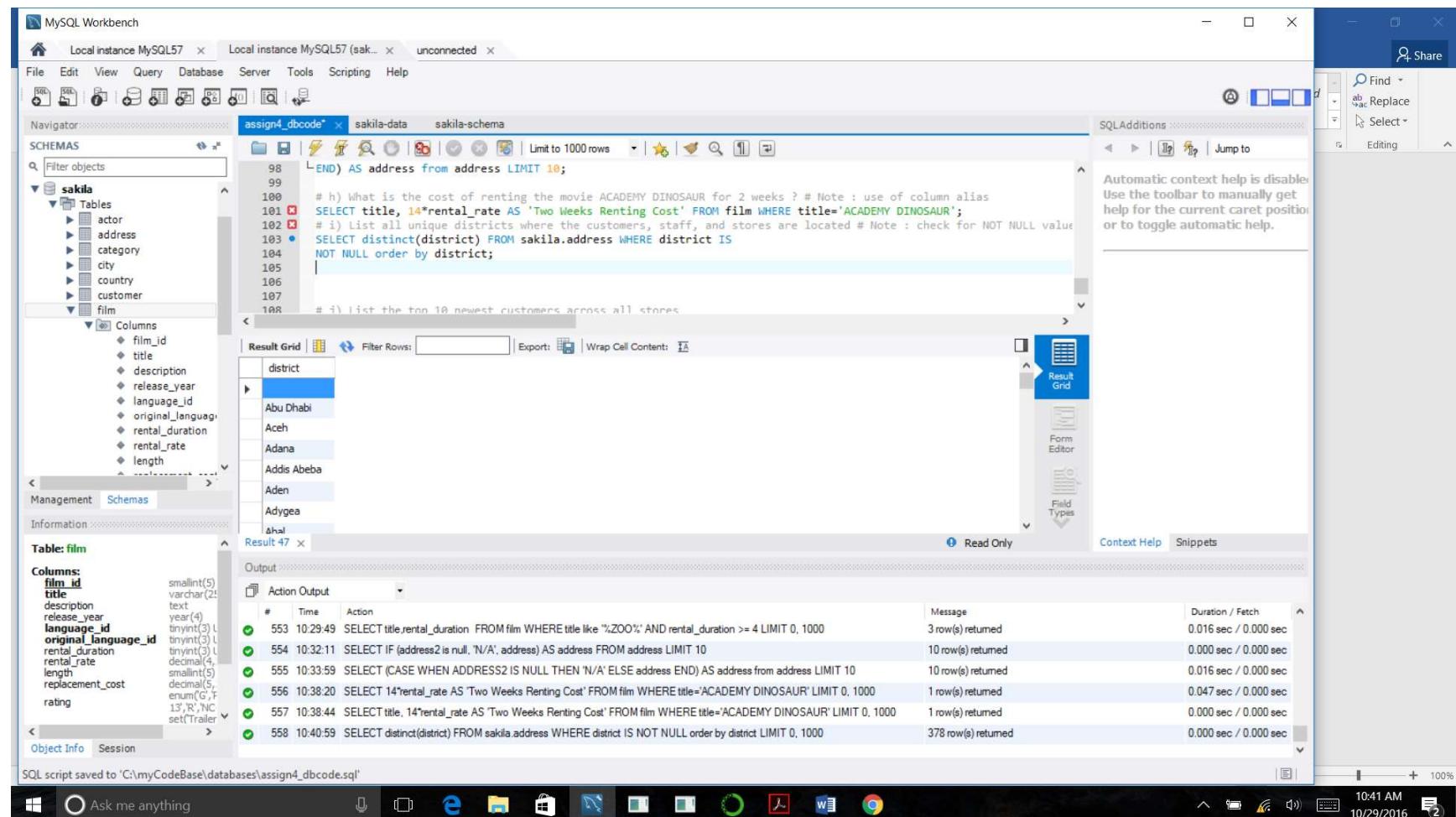
```

93 WHEN ADDRESS2 IS NULL
94 THEN
95 'N/A'
96 ELSE
97 address
98 END) AS address from address LIMIT 10;
99
100 # h) What is the cost of renting the movie ACADEMY DINOSAUR for 2 weeks ? # Note : use of column alias
101 SELECT title, 14*rental_rate AS 'Two Weeks Renting Cost' FROM film WHERE title='ACADEMY DINOSAUR';
102
103

```
- Result Grid:** Shows the result of the query:

title	Two Weeks Renting Cost
ACADEMY DINOSAUR	13.86
- Information Panel (Table: film):** Shows the columns and their data types for the film table.
- Output Panel:** Displays the execution log with 7 rows of data, each showing a query ID, time, action, message, and duration.
- System Bar:** Shows the Windows taskbar with various icons and the system clock indicating 10:39 AM on 10/29/2016.

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;



The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** The main pane displays the following SQL code:

```

# 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
    
```
- Result Grid:** The results of the third query are shown in a table:

district
Abu Dhabi
Aceh
Adana
Addis Abeba
Aden
Adygea
Ahal
- Output Window:** The bottom pane shows the execution log with 58 entries, including the execution of the third query:

```

553 10:29:49 SELECT title,rental_duration FROM film WHERE title like "%ZOO%" AND rental_duration >= 4 LIMIT 0, 1000
554 10:32:11 SELECT IF(address2 is null, 'N/A', address) AS address FROM address LIMIT 10
555 10:33:59 SELECT (CASE WHEN ADDRESS2 IS NULL THEN 'N/A' ELSE address END) AS address from address LIMIT 10
556 10:38:20 SELECT 14*rental_rate AS 'Two Weeks Renting Cost' FROM film WHERE title='ACADEMY DINOSAUR' LIMIT 0, 1000
557 10:38:44 SELECT title, 14*rental_rate AS 'Two Weeks Renting Cost' FROM film WHERE title='ACADEMY DINOSAUR' LIMIT 0, 1000
558 10:40:59 SELECT distinct(district) FROM sakila address WHERE district IS NOT NULL order by district LIMIT 0, 1000
    
```
- Information:** The left sidebar shows the database schema for the `sakila` database, including tables like `actor`, `address`, `category`, etc.
- System:** The taskbar at the bottom shows the date and time as 10:41 AM on 10/29/2016.

j) List the top 10 newest customers across all stores

SELECT * FROM customer ORDER BY create_date DESC LIMIT 10;

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema for the sakila database, including tables like film, customer, and address.
- SQL Editor:** Contains the SQL query: `SELECT * FROM customer ORDER BY create_date DESC LIMIT 10;`.
- Result Grid:** Displays the results of the query, showing 10 rows of customer data:

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
275	2	CAROLE	BARNETT	CAROLE.BARNETT@sakilacustomer.org	280	1	2006-02-14 22:04:37	2006-02-15 04:57:
281	2	LEONA	OBIEN	LEONA.OBIEN@sakilacustomer.org	286	1	2006-02-14 22:04:37	2006-02-15 04:57:
278	2	BILLIE	HORTON	BILLIE.HORTON@sakilacustomer.org	283	1	2006-02-14 22:04:37	2006-02-15 04:57:
276	1	BRANDY	GRAVES	BRANDY.GRAVES@sakilacustomer.org	281	1	2006-02-14 22:04:37	2006-02-15 04:57:
599	2	AUSTIN	CINTRON	AUSTIN.CINTRON@sakilacustomer.org	605	1	2006-02-14 22:04:37	2006-02-15 04:57:
280	2	TRACEY	BARRETT	TRACEY.BARRETT@sakilacustomer.org	285	1	2006-02-14 22:04:37	2006-02-15 04:57:
274	1	NAOMI	JENNINGS	NAOMI.JENNINGS@sakilacustomer.org	279	1	2006-02-14 22:04:37	2006-02-15 04:57:
279	2	DIANNE	SHELTON	DIANNE.SHELTON@sakilacustomer.org	284	1	2006-02-14 22:04:37	2006-02-15 04:57:
277	2	OLGA	JIMENEZ	OLGA.JIMENEZ@sakilacustomer.org	282	1	2006-02-14 22:04:37	2006-02-15 04:57:
272	2	DOTCILLA	LOME	DOTCILLA.LOME@sakilacustomer.org	278	1	2006-02-14 22:04:37	2006-02-15 04:57:

- Output:** Shows the execution log with two entries:

 - 554 10:32:11 SELECT IF (address2 is null, 'N/A', address) AS address FROM address LIMIT 10
 - 555 10:33:59 SET FCT (CASE WHEN ADDRESS2 IS NULL THEN 'N/A' ELSE address END) AS address FROM address LIMIT 10

The status bar at the bottom right indicates the time is 10:45 AM and the date is 10/29/2016.

QUESTION 3

a) Show total number of movies

```
SELECT COUNT(*) AS TOTAL_NO_OF_MOVIES FROM film;
```

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `sakila` with tables `country`, `customer`, and `film`. The `film` table is expanded, showing columns: `film_id`, `title`, `description`, `release_year`, `language_id`, `original_language_id`, `rental_duration`, `rental_rate`, `length`, `replacement_cost`, `rating`, `special_features`, and `last_update`.
- SQL Editor:** Contains a multi-line SQL script. Line 110 contains the query: `SELECT COUNT(*) AS TOTAL_NO_OF_MOVIES FROM film;`. This line is highlighted in blue, indicating it is the current selection.
- Result Grid:** Displays the result of the query: `TOTAL_NO_OF_MOVIES` with value `1000`.
- Output Window:** Shows the execution log:
 - Action Output:
 - 556 10:38:20 SELECT 14*rental_rate AS 'Two Weeks Renting Cost' FROM film WHERE title='ACADEMY DINOSAUR' LIMIT 0, 1000
 - 557 10:38:44 SELECT COUNT(*) AS TOTAL_NO_OF_MOVIES FROM film WHERE title='ACADEMY DINOSAUR' LIMIT 0, 1000
 - Message: 1 row(s) returned
 - Duration / Fetch: 0.047 sec / 0.000 sec
 - Message: 1 row(s) returned
 - Duration / Fetch: 0.000 sec / 0.000 sec
- Right Panel:** Includes sections for "SQLAdditions", "Automatic context help is disabled", "Export PDF", "Create PDF", "Edit PDF", "Adobe Acrobat Pro", and "Learn more".
- System Bar:** Shows the date and time: 10:49 AM, 10/29/2016.

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;

The screenshot shows the MySQL Workbench interface with a query editor window open. The query editor contains the following code:

```
103 • SELECT distinct(district) FROM sakila.address WHERE district IS
104 NOT NULL order by district;
# j) List the top 10 newest customers across all stores
106 • SELECT * FROM customer ORDER BY create_date DESC LIMIT 10;
107
#####
108 ##### QUESTION 3 #####
109 # a) Show total number of movies
110 • SELECT COUNT(*) AS TOTAL_NO_OF_MOVIES FROM film;
111 # b) What is the minimum payment received and max payment received across all transactions ?
112 • SELECT min(amount) AS 'MINIMUM_PAYMENT', max(amount) AS 'MAXIMUM_PAYMENT' from payment;
113
114
115 # c) Number of customers that rented movies between Feb-2005 and May-2005 ( based on payment date ).
116 # d) List all movies where replacement_cost is greater than 15$ or rental_duration is between 6 and 10 days
```

The results grid shows the following data:

MINIMUM_PAYMENT	MAXIMUM_PAYMENT
0.00	11.99

The status bar at the bottom right indicates the time is 10:51 AM and the date is 10/29/2016.

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';`

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
105 # j) List the top 10 newest customers across all stores
106 • SELECT * FROM customer ORDER BY create_date DESC LIMIT 10;
107
108 ##### QUESTION 3 #####
109 # a) Show total number of movies
110 • SELECT COUNT(*) AS TOTAL_NO_OF_MOVIES FROM film;
111 # b) What is the minimum payment received and max payment received across all transactions ?
112 • SELECT min(amount) AS 'MINIMUM_PAYMENT', max(amount) AS 'MAXIMUM_PAYMENT' from payment;
113
114 # c) Number of customers that rented movies between Feb-2005 and May-2005 ( based on payment date ).
115 • SELECT count(*) AS NO_OF_CUSTOMERS FROM payment WHERE payment_date BETWEEN '20050201' AND '20050531';
116
117
118 # d) List all movies where replacement_cost is greater than 15$ or rental_duration is between 6 and 10 days
```

The results grid shows a single row with the value 994 for the column 'NO_OF_CUSTOMERS'.

On the right side of the interface, there is a sidebar titled 'SQLAdditions' which includes options like 'Share', 'Find', 'Export PDF', 'Create PDF', and 'Edit PDF'. Below that is an advertisement for 'Adobe Acrobat Pro'.

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';
```

The screenshot shows the MySQL Workbench interface. The SQL editor pane contains the following code:

```
108 ##### QUESTION 3 #####
109 # a) Show total number of movies
110 SELECT COUNT(*) AS TOTAL_NO_OF_MOVIES FROM film;
111 # b) What is the minimum payment received and max payment received across all transactions ?
112 SELECT min(amount) AS 'MINIMUM_PAYMENT', max(amount) AS 'MAXIMUM_PAYMENT' from payment;
113
114 # c) Number of customers that rented movies between Feb-2005 and May-2005 ( based on payment date ).
115 SELECT count(*) AS NO_OF_CUSTOMERS FROM payment WHERE payment_date BETWEEN '20050201' AND '20050531';
116
117 # d) List all movies where replacement_cost is greater than 15$ or rental_duration is between 6 and 10 days
118 SELECT title,replacement_cost,rental_duration FROM film where replacement_cost > 15 OR
rental_duration between '6' AND '10';
```

The results grid displays the following data:

title	replacement_cost	rental_duration
ACADEMY DINOSAUR	20.99	6
ADAPTATION HOLES	18.99	7
AFFAIR PREJUDICE	26.99	5
AFRICAN EGG	22.99	6
AGENT TRUMAN	17.99	3
AIRPLANE SIERRA	28.99	6
AIRPORT POLLOCK	15.99	6
ALABAMA DEVIL	21.99	3
ALADDIN CALENDAR	24.99	6
ALAMO VIDEOTAPE	16.99	6

The status bar at the bottom right shows the time as 11:03 AM and the date as 10/29/2016.

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';

The screenshot shows the MySQL Workbench interface. The main window displays a query editor with the following SQL code:

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

The result grid shows a single row with the value 'NULL'.

On the right side of the interface, there is a vertical panel titled "SQL Additions" which contains a list of database objects and their sizes. The list includes:

Object	Size
D...	1,785 KB
D...	257 KB
D...	207 KB
D...	287 KB
D...	759 KB
D...	247 KB
W...	16 KB
	476 KB
D...	5,175 KB
D...	3,780 KB
D...	1,419 KB
D...	14,910 KB
D...	44 KB
D...	27 KB
W...	15 KB
	97 KB
W...	29 KB
pp...	539 KB
Co...	44,153 KB
	527 KB
	397 KB
	361 KB

The bottom of the screen shows the Windows taskbar with various application icons and the system clock indicating 11:30 AM on 10/29/2016.

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

SELECT AVG(replacement_cost) FROM film;

The screenshot shows the MySQL Workbench interface with the following details:

- SQL Editor:** Displays the following SQL code:

```
115 SELECT count(*) AS NO_OF_CUSTOMERS FROM payment WHERE payment_date BETWEEN '20050201' AND '20050531';
116
117 # d) List all movies where replacement_cost is greater than 15$ or rental_duration is between 6 and 10 days
118 SELECT title,replacement_cost,rental_duration FROM film WHERE replacement_cost > 15 OR
119 rental_duration BETWEEN '6' AND '10';
120
121 # e) What is the total amount spent by customers for movies in the year 2005 ?
122 SELECT sum(rental_rate) FROM film WHERE release_year='2005';
123
124 # f) What is the average replacement cost across all movies ?
125 SELECT AVG(replacement_cost) FROM film;
```
- Result Grid:** Shows the result of the last query:

AVG(replacement_cost)
19.98400
- Object Info:** Shows information about the **payment** table, including columns and their types.
- Session:** Shows the history of queries run in the session, including the execution of the last query.
- System Tray:** Shows the date and time as 11:33 AM, 10/29/2016.

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

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

The screenshot shows the MySQL Workbench interface with the following details:

- SQL Editor:** Displays the following code:

```
116
117 # d) List all movies where replacement_cost is greater than 15$ or rental_duration is between 6 and 10 days
118 SELECT title,replacement_cost,rental_duration FROM film WHERE replacement_cost > 15 OR
119 rental_duration BETWEEN '6' AND '10';
120
121 # e) What is the total amount spent by customers for movies in the year 2005 ?
122 SELECT sum(rental_rate) FROM film WHERE release_year='2005';
123
124 # f) What is the average replacement cost across all movies ?
125 SELECT AVG(replacement_cost) FROM film;
126
127 # g) What is the standard deviation of rental rate across all movies ? Assignment 4
128 SELECT STD(rental_rate) as 'Standard Deviation' from film;
129
```
- Result Grid:** Shows the output of the last query:

Standard Deviation
1.6455698101265719
- Output Window:** Shows the execution log:

Action	Time	Message	Duration / Fetch
SELECT rental_rate FROM film WHERE release_year='2005' LIMIT 0, 1000	570 11:28:14	0 row(s) returned	0.015 sec / 0.000 sec
SUM(amount) FROM payment WHERE release_year='2005' LIMIT 0, 1000	571 11:29:38	Error Code: 1054 Unknown column 'release_year' in '...	0.000 sec

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

SELECT AVG(rental_duration) FROM film;

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema for 'assign4_dbcode'. The 'payment' table is selected, displaying its columns: payment_id, customer_id, staff_id, rental_id, amount, payment_date, and last_update.
- SQL Editor:** Contains the following code:

```
122 # f) What is the average replacement cost across all movies ?
123
124 # g) What is the standard deviation of rental rate across all movies ? Assignment 4
125 • SELECT AVG(replacement_cost) FROM film;
126
127 # h) What is the midrange of the rental duration for all movies
128 • SELECT STD(rental_rate) as 'Standard Deviation' from film;
129
130 # a) Customers sorted by first Name and last name in ascending order.
131 • SELECT AVG(rental_duration) FROM film;
132
133
134 ##### QUESTION 4 #####
135 # a) Customers sorted by first Name and last name in ascending order.
```
- Result Grid:** Displays the result of the query: **AVG(rental_duration)** = **4.9850**.
- Output:** Shows the log of actions taken:
 - Action: SELECT sum(amount) FROM payment WHERE release_year='2005' LIMIT 0, 1000. Message: Error Code: 1054. Unknown column 'release_year' in ... Duration / Fetch: 0.000 sec.
 - Action: SELECT sum(amount) FROM payment WHERE release_year='2005' LIMIT 0, 1000. Message: Error Code: 1054. Unknown column 'release_year' in ... Duration / Fetch: 0.000 sec.
- System Bar:** Shows the Windows taskbar with various application icons and the date/time: 11:36 AM 10/29/2016.

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;

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with the `customer` table selected. The `customer` table has columns: `customer_id`, `store_id`, `first_name`, `last_name`, `email`, `address_id`, `active`, `create_date`, and `last_update`.
- SQL Editor:** Contains the following SQL code:

```
127 # g) What is the standard deviation of rental rate across all movies ? Assignment 4
128 • SELECT STD(rental_rate) as 'Standard Deviation' from film;
129
130 # h) What is the midrange of the rental duration for all movies
131 • SELECT AVG(rental_duration) FROM film;
132
133 #####
134 ##### QUESTION 4 #####
135 # a) Customers sorted by first Name and last name in ascending order.
136 • SELECT first_name, last_name FROM customer order by first_name ASC, last_name ASC;
```
- Result Grid:** Displays the results of the query:

first_name	last_name
AARON	SELBY
ADAM	GOOCH
ADRIAN	CLARY
AGNES	BISHOP
ALAN	KAHN
ALBERT	CROUSE
ALBERTO	HENNING
ALEX	GRESHAM
ALEXANDER	FENNELL
ALFRED	CASILLAS
- Output:** Shows the execution log with three entries:| # | Time | Action | Message | Duration / Fetch |
| --- | --- | --- | --- | --- |
| 573 | 11:30:17 | SELECT sum(rental_rate) FROM film WHERE release_year=2005 LIMIT 0, 1000 | 1 row(s) returned | 0.000 sec / 0.000 sec |
| 574 | 11:33:50 | SELECT AVG(replacement_cost) FROM film LIMIT 0, 1000 | 1 row(s) returned | 0.000 sec / 0.000 sec |
| 575 | 11:34:20 | SELECT STD(rental_rate) as 'Standard Deviation' from film LIMIT 0, 1000 | 1 row(s) returned | 0.015 sec / 0.000 sec |

b) Group distinct addresses by district.

SELECT distinct address FROM sakila.address group by district;

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Standard database management tools like New, Open, Save, Print, etc.
- Navigator:** Shows the schema 'assign4_dbcode' with the 'sakila' database selected. Under 'Tables', 'address' is expanded, showing columns: address_id, address, address2, district, city_id, postal_code, phone, location, last_update.
- SQL Editor:** Contains a numbered list of SQL queries:
 - 129
 - 130 # h) What is the midrange of the rental duration for all movies
 - 131 • SELECT AVG(rental_duration) FROM film;
 - 132 ✘
 - 133
 - 134 ##### QUESTION 4 #####
 - 135 # a) Customers sorted by first Name and last name in ascending order.
 - 136 ✘ SELECT first_name, last_name FROM customer order by first_name ASC, last_name ASC;
 - 137 # b) Group distinct addresses by district.
 - 138 ✘ SELECT distinct address FROM sakila.address group by district;
 - 139 # c) Count of movies that are either G/NC-17/PG-13/PG/R grouped by rating.
 - 140
- Result Grid:** Displays the results of the last query, showing a list of addresses:
 - 18 Duisburg Boulevard
 - 669 Firozabad Loop
 - 1078 Stara Zagora Drive
 - 842 Salzburg Lane
 - 614 Pak Kret Street
 - 751 Lima Loop
 - 1157 Nyeri Loop
 - 387 Mwene-Ditu Drive
 - 775 ostka Drive
 - 1759 Niznekamsk Avenue
- Output:** Shows the execution log:

#	Time	Action	Message	Duration / Fetch
576	11:36:16	SELECT AVG(rental_duration) FROM film LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
577	13:53:53	SELECT first_name, last_name FROM customers order by first_name ASC, last_name ASC LIMIT 0, 1000	Error Code: 1146. Table 'sakila customers' doesn't exist	0.015 sec
578	13:54:46	SELECT first_name, last_name FROM customers order by first_name ASC, last_name ASC LIMIT 0, 1000	100 rows selected, 100 total rows	0.015 sec / 0.000 sec
- System Bar:** Shows the Windows taskbar with various pinned icons (File Explorer, Edge, Task View, etc.) and the system clock (2:09 PM) and date (10/29/2016).

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');

The screenshot shows the MySQL Workbench interface. The SQL editor pane contains the following code:

```
131 • SELECT AVG(rental_duration) FROM film;
132 ✘
133
134 ##### QUESTION 4 #####
135 # a) Customers sorted by first Name and last name in ascending order.
136 ✘ SELECT first_name, last_name FROM customer ORDER BY first_name ASC, last_name ASC;
137 # b) Group distinct addresses by district.
138 ✘ SELECT DISTINCT address FROM sakila.address GROUP BY district;
139 # c) Count of movies that are either G/NC-17/PG-13/PG/R grouped by rating.
140 ✘ SELECT COUNT(*) FROM film WHERE rating IN ('G','NC-17','PG-13','PG','R');
141
142 # d) Number of addresses in each district.
```

The results grid shows the output of the last query:

count(*)
1000

The status bar at the bottom right indicates the current time is 2:15 PM on 10/29/2016.

d) Number of addresses in each district.

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

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema **sakila** with the **address** table selected. The **Columns** section lists: address_id, address, address2, district, city_id, postal_code, phone, location, last_update.
- SQL Editor:** Displays a script with numbered comments and queries. Line 142 contains the query: `SELECT district, count(address) FROM address GROUP BY district;`.
- Result Grid:** Shows the output of the query in a tabular format. The columns are **district** and **count(address)**. The data is as follows:

district	count(address)
Abu Dhabi	3
Aceh	2
Adana	1
Addis Abeba	2
Aden	1
Adygea	1
Ahal	1
al-Daqahliya	1
al-Manama	1

- Action Output:** Shows the execution log with three entries corresponding to the queries in the SQL editor.
- System Bar:** Includes the Windows taskbar with icons for File Explorer, Edge, and other applications, along with the date and time (10/29/2016, 2:17 PM).

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;

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Includes icons for New Connection, Open Connection, Save, Print, Copy, Paste, Find, and others.
- Navigator:** Shows the schema 'sakila' with tables: actor, address, category, city, country, customer, and film. The 'film' table is expanded to show columns: film_id, title, description, release_year, language_id, original_language, rental_duration, rental_rate, and length.
- SQL Editor:** The query `SELECT title,rental_rate FROM film WHERE rental_rate > '1' order by title DESC;` is highlighted in red, indicating an error or warning.
- Result Grid:** Displays the results of the query, showing 75 rows of movie titles and their rental rates. The first few rows are:

title	rental_rate
ZORRO ARK	4.99
ZOOLANDER FICTION	2.99
YENTL IDAHO	4.99
WYOMING STORM	4.99
WRONG BEHAVIOR	2.99
WORST BANGER	2.99
WORKING MICROCOSMOS	4.99
WORKER TARZAN	2.99
WORDS HUNTER	2.99
WONKA SEA	2.99
- Output:** Shows the execution log with three entries, all successful (green checkmarks). The last entry is:

Action	Message	Duration / Fetch
584 14:12:46 SELECT COUNT(*) FROM film WHERE rating in ('G','NC-17','PG-13','PG','R') LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
585 14:14:18 SELECT * FROM film WHERE rating in ('G','NC-17','PG-13','PG','R') LIMIT 0, 1000	1000 row(s) returned	0.047 sec / 0.000 sec
586 14:15:12 SELECT COUNT(*) FROM film WHERE rating in ('G','NC-17','PG-13','PG','R') LIMIT 0, 1000	1 row(s) returned	0.016 sec / 0.000 sec
- System Bar:** Shows the Windows taskbar with various pinned icons like File Explorer, Edge, and others. The system tray shows the date and time as 2:21 PM, 10/29/2016.

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;`

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema structure for the `assign4_dbcode` database, including tables like address, category, city, country, customer, and film.
- SQL Editor:** Displays the following SQL code:

```
139 # c) Count of movies that are either G/NC-17/PG-13/PG/R grouped by rating.
140 SELECT count(*) FROM film WHERE rating in ('G','NC-17','PG-13','PG','R');
141 # d) Number of addresses in each district.
142 SELECT district, count(address) FROM address GROUP BY district;
143
144 # e) Find the movies where rental rate is greater than 1$ and order result set by descending order.
145 SELECT title,rental_rate FROM film WHERE rental_rate > '1' order by title DESC;
146
147 # f) Top 2 movies that are rated R with the highest replacement cost ?
148 SELECT title,rating,replacement_cost FROM film WHERE rating = 'R' order by replacement_cost DESC LIMIT 2;
```
- Result Grid:** Shows the results of the last query:

title	rating	replacement_cost
CHARIOTS CONSPIRACY	R	29.99
CUPBOARD SINNERS	R	29.99
- Output Window:** Shows the execution log:

Action	Time	Message	Duration / Fetch
SELECT count(*) FROM film WHERE rating in ('G','NC-17','PG-13','PG','R')	14:15:13	1 row(s) returned	0.016 sec / 0.000 sec
SELECT district, count(address) FROM address GROUP BY district LIMIT 0, 1000	14:16:36	378 row(s) returned	0.016 sec / 0.000 sec
SELECT title,rating,replacement_cost FROM film WHERE rating = 'R' order by replacement_cost DESC LIMIT 2	14:17:30	ER0 invalid command	0.017 sec / 0.000 sec
- System Bar:** Shows the Windows taskbar with various application icons and the system clock indicating 2:29 PM on 10/29/2016.

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;
```

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with the `film` table selected. The `rental_rate` column is expanded, showing its data types and constraints.
- SQL Editor:** Displays two queries:
 - # 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;
- Result Grid:** Shows the result of the second query:

rental_rate	COUNT(*)
0.99	341
- Output:** Shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
588	14:20:29	SELECT * FROM film WHERE rental_rate > '1' order by title DESC LIMIT 0, 1000	659 row(s) returned	0.047 sec / 0.000 sec
589	14:20:56	SELECT title,rental_rate FROM film WHERE rental_rate > '1' order by title DESC LIMIT 0, 1000	659 row(s) returned	0.031 sec / 0.000 sec
590	14:26:48	SELECT title,rating,replacement_cost FROM film WHERE rating = 'R' order by replacement_cost DESC LIMIT 0, 1000	195 row(s) returned	0.000 sec / 0.000 sec
591	14:27:21	SELECT title,rating,replacement_cost FROM film WHERE rating = 'R' order by replacement_cost DESC LIMIT 2	2 row(s) returned	0.000 sec / 0.000 sec
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

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 '%commentaries%' ORDER BY length DESC LIMIT 2;
```

MySQL Workbench

Local instance MySQL57 (sakila)

File Edit View Query Database Server Tools Scripting Help

Navigator: assign4_dbcode x sakila-data sakila-schema

SCHEMAS: address, category, city, country, customer, film

Columns: film_id, title, description, release_year, language_id, original_language, rental_duration, rental_rate, length, replacement_cost, rating, special_features

Find: # g) Find the most frequently occurring (mode) rental rate across products.

150
151
152 x SELECT rental_rate, COUNT(*) FROM film GROUP BY rental_rate ORDER BY COUNT(rental_rate) DESC LIMIT 1;

153
154
155 x
156 x
157
158 # h) Find the top 2 movies with movie length greater than 50mins and which has commentaries as a special features.

SELECT title,length,special_features FROM film WHERE length > 50 AND special_features LIKE '%commentaries%' ORDER BY length DESC LIMIT 2

Result Grid: title, length, special_features

title	length	special_features
CONTROL ANTHEM	185	Commentaries
HOME PITY	185	Trailers,Commentaries,Behind the Scenes

SQLAdditions: Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Management Schemas

Information: Table: address

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 leng...	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 '%commentaries%' ORDER BYI...	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

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;

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema structure for the 'assign4_dbcode' database, including tables like address, category, city, country, customer, and film.
- SQL Editor:** Contains the following SQL code:

```
155 
156 SELECT title,length,special_features FROM film WHERE length > 50 AND special_features LIKE '%Kommentaries%' ORDER BY length DESC LIMIT 2;
157 
158 # i) List the years with more than 2 movies released.
159 
160 SELECT release_year, title FROM film;
161 
162 SELECT distinct(release_year), count(*) FROM film GROUP BY release_year HAVING COUNT(*) > 2;
163
```
- Result Grid:** Displays the result of the last query:

release_year	count(*)
2006	1000
- Output:** Shows the execution log with the following entries:

Action	Time	Message	Duration / Fetch
SELECT distinct(release_year), count(*) FROM film GROUP BY release_year HAVING COUNT(*) > 2 LIMIT 0, 1000	599 14:53:45	1 row(s) returned	0.016 sec / 0.000 sec
SELECT distinct(release_year), count(*) FROM film GROUP BY release_year HAVING COUNT(*) > 2 LIMIT 0, 1000	600 14:55:00	1 row(s) returned	0.016 sec / 0.000 sec
SELECT release_year, title FROM film GROUP BY release_year LIMIT 0, 1000	601 14:55:25	1 row(s) returned	0.000 sec / 0.000 sec
SELECT release_year, title FROM film LIMIT 0, 1000	602 14:55:47	1000 row(s) returned	0.016 sec / 0.000 sec
SHOW DATABASES	603 14:56:25	8 row(s) returned	0.015 sec / 0.000 sec
USE sakila #shows the list of all tables in the database SHOW TABLES	604 14:56:25	Error Code: 1064. You have an error in your SQL syntax...	0.000 sec
SELECT distinct(release_year), count(*) FROM film GROUP BY release_year HAVING COUNT(*) > 2 LIMIT 0, 1000	605 14:56:32	1 row(s) returned	0.000 sec / 0.000 sec
- System Bar:** Shows the Windows taskbar with various application icons and the system clock indicating 2:56 PM on 10/29/2016.

