

Assignment3

CSC 4710 Database System

Hyunki Lee

Panther#: 002-34-4677

Part2

1. The number of tables: 23 tables.

select count(*) as number_of_tables

from information_schema.tables

where table_schema = 'sakila';

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including tables, views, stored procedures, and functions for the 'sakila' schema. The 'sakila' schema is selected, and its tables are listed. The 'Query' pane in the center contains the following SQL query:

```
1 select count(*) as number_of_tables
2 from information_schema.tables
3 where table_schema = 'sakila';
```

The 'Result Grid' pane at the bottom shows the execution results. It displays a single row with the value '23' under the column 'number_of_tables'. The 'Action Output' pane at the bottom right shows the execution details, including the time taken (01:36:14) and the message '1 row(s) returned'.

| # | Time | Action | Message | Duration / Fetch |
|---|----------|---|-------------------|-----------------------|
| 1 | 01:36:14 | select count(*) as number_of_tables from information_schema.tables where table_sch... | 1 row(s) returned | 0.000 sec / 0.000 sec |

2.

show columns

from customer;

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'sakila' database selected. The main query editor contains the SQL query: `show columns from customer;`. The 'Result Grid' at the bottom shows the output of the query, which is the structure of the 'customer' table. The table has 10 columns: customer_id, store_id, first_name, last_name, email, address_id, active, create_date, and last_update. The 'customer_id' column is the primary key and has an auto-increment attribute.

| Field | Type | Null | Key | Default | Extra |
|-------------|-------------------|------|-----|-------------------|---|
| customer_id | smallint unsigned | NO | PR1 | | auto_increment |
| store_id | tinyint unsigned | NO | MUL | | |
| first_name | varchar(45) | NO | | | |
| last_name | varchar(45) | NO | MUL | | |
| email | varchar(50) | YES | | | |
| address_id | smallint unsigned | NO | MUL | | |
| active | tinyint(1) | NO | | 1 | |
| create_date | datetime | NO | | | |
| last_update | timestamp | YES | | CURRENT_TIMESTAMP | DEFAULT_GENERATED on update CURRENT_TIMESTAMP |

Table: customer
Columns:
customer_id smallint UN AI PK
store_id tinyint UN
first_name varchar(4)
last_name varchar(4)
email varchar(5)
address_id smallint UN
active tinyint(1)
create_date datetime
last_update timestamp

Result 4 x
Output
Action Output
Time Action Message Duration / Fetch
1 01:46:57 show columns from customer 9 row(s) returned 0.015 sec / 0.000 sec

3.

select *

from actor

limit 5;

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: sakila-mv-schema sakila-mv-data

Query 1

```
1 select *
2 from actor
3 limit 5;
```

Result Grid

| actor_id | first_name | last_name | last_update |
|----------|------------|--------------|---------------------|
| 1 | PENELOPE | GUINNESS | 2006-02-15 04:34:33 |
| 2 | NECK | WAHLBERG | 2006-02-15 04:34:33 |
| 3 | ED | CHASE | 2006-02-15 04:34:33 |
| 4 | JENNIFER | DAVIS | 2006-02-15 04:34:33 |
| 5 | JOHNNY | LOLLOBRIGIDA | 2006-02-15 04:34:33 |

Table: customer

Columns:

- customer_id: smallint(4) UNSIGNED ZEROFILL PRIMARY KEY
- store_id: tinyint(4) UNSIGNED ZEROFILL
- first_name: varchar(45)
- last_name: varchar(45)
- email: varchar(45)
- address_id: smallint(4) UNSIGNED ZEROFILL
- active: tinyint(1) UNSIGNED ZEROFILL
- create_date: datetime
- last_update: timestamp

actor 9 x

Output

Action Output

| # | Time | Action | Message | Duration / Fetch |
|---|----------|-----------------------------|-------------------|-----------------------|
| 1 | 01:54:44 | select * from actor limit 5 | 5 row(s) returned | 0.000 sec / 0.000 sec |

4.

1000 films.

```
select count(film_id)
```

```
from film
```

```
where release_year between 2000 and 2007;
```

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays the database structure, including the 'sakila' database and its tables. The 'film' table is selected, and its columns are listed: film_id, title, description, release_year, language_id, original_language_id, rental_duration, rental_rate, length, replacement_cost, rating, special_features, and last_update. The 'Query' pane in the center contains the following SQL query:

```
1 select count(film_id)
2 from film
3 where release_year between 2000 and 2007;
```

The 'Result Grid' pane on the right shows the query results:

| count(film_id) |
|----------------|
| 1000 |

At the bottom, the 'Action Output' pane shows the execution details:

| # | Time | Action | Message | Duration / Fetch |
|---|----------|--|-------------------|-----------------------|
| 1 | 02:04:16 | select count(film_id) from film where release_year between 2000 and 2007 | 1 row(s) returned | 0.000 sec / 0.000 sec |

5.

Total 1000 films

```
select count(film_id)
```

```
from film;
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'sakila' database selected. The 'Columns' tab for the 'film' table is active, showing columns like film_id, title, description, etc. The main query editor contains the following SQL query:

```
1 select count(film_id)
2 from film;
```

The 'Result Grid' at the bottom shows the execution results:

| | count(film_id) |
|---|----------------|
| 1 | 1000 |

The bottom status bar indicates that the query was executed successfully, returning 1 row(s) in 0.000 seconds.

6.

select title, length

from film

where length = (select max(length)

from film);

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays the 'sakila' database structure, including tables like 'actor', 'address', 'category', 'city', 'country', 'customer', 'film', 'film_actor', 'film_category', 'film_text', 'inventory', 'language', 'payment', 'rental', 'staff', 'store', 'actor_info', 'customer_list', and 'film_list'. The 'film' table is selected, showing its columns: film_id, title, description, release_year, language_id, original_lang, rental_duration, rental_rate, length, replacement_cost, rating, special_features, and last_update. The 'Columns' pane for 'film' is visible, showing the 'length' column with a definition of 'length smallint UNSIGNED'. The main query editor contains the following SQL query:

```
1 select title, length
2 from film
3 where length = (select max(length)
4                 from film);
```

The 'Result Grid' pane shows the results of the query, displaying 10 rows of film titles and their lengths (all 185).

| title | length |
|--------------------|--------|
| CHICAGO NORTH | 185 |
| CONTROL ANTHEM | 185 |
| DARIN FORRESTER | 185 |
| GANGS PRIDE | 185 |
| HOME PTY | 185 |
| MUSCLE BRIGHT | 185 |
| POND SEATTLE | 185 |
| SOLDIERS EVOLUTION | 185 |
| SWEET BROTHERHOOD | 185 |
| WORST BANGER | 185 |

The 'Output' pane at the bottom shows the execution log, including the query execution time and the number of rows returned.

| # | Time | Action | Message | Duration / Fetch |
|---|----------|--|----------------------|-----------------------|
| 1 | 02:05:47 | select count(film_id) from film | 1 row(s) returned | 0.000 sec / 0.000 sec |
| 2 | 02:08:03 | select * from film | 1000 row(s) returned | 0.000 sec / 0.000 sec |
| 3 | 02:12:38 | select title from film order by length desc | 1000 row(s) returned | 0.000 sec / 0.000 sec |
| 4 | 02:12:46 | select title, length from film order by length desc | 1000 row(s) returned | 0.000 sec / 0.000 sec |
| 5 | 02:14:44 | select title, length from film where length = (select max(length) from film) | 10 row(s) returned | 0.000 sec / 0.000 sec |

7.

select title, replacement_cost

from film

where replacement_cost = (select max(replacement_cost)

from film);

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays the 'sakila' database structure, including tables like 'actor', 'address', 'category', 'city', 'country', 'customer', 'film', 'inventory', 'language', 'payment', 'rental', 'staff', and 'store'. The 'film' table is selected, showing its 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.

The central 'Query' pane contains the following SQL query:

```
1 select title, replacement_cost
2 from film
3 where replacement_cost = (select max(replacement_cost)
4 from film);
5
```

The 'Result Grid' pane at the bottom displays the query results as a table with two columns: 'title' and 'replacement_cost'. The results show 53 rows of film titles and their replacement costs, all of which are 29.99.

| title | replacement_cost |
|--------------------------|------------------|
| ARABIA DOGMA | 29.99 |
| BALLROOM MOCKINGBIRD | 29.99 |
| BLINDNESS GUN | 29.99 |
| BONNIE HOLOCAUST | 29.99 |
| CHARLOTS CONSPIRACY | 29.99 |
| CLOCKWORK PARADISE | 29.99 |
| CLYDE THEORY | 29.99 |
| CRUELTY UNFORGIVEN | 29.99 |
| CLIPBOARD SINNERS | 29.99 |
| DESPERATE TRAINSPOTTING | 29.99 |
| DIRTY ACE | 29.99 |
| DOCTOR GRAIL | 29.99 |
| EARTH VISION | 29.99 |
| EVERYONE CRAFT | 29.99 |
| EXTRAORDINARY CONQUER... | 29.99 |
| FANTASIA PARK | 29.99 |
| FELD FROGMEN | 29.99 |
| FLATLINERS KILLER | 29.99 |

The 'Output' pane at the bottom shows the execution details: 1 row(s) returned, 0.000 sec / 0.000 sec.

8. 178 films for rating 'G'.

```
select count(film_id)
```

```
from film
```

```
where rating='G';
```

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including tables like 'actor', 'address', 'category', 'city', 'country', 'customer', 'film', 'film_actor', 'film_category', 'film_text', 'inventory', 'language', 'payment', 'rental', 'staff', 'store', and 'staff_list'. The 'film' table is selected, showing its 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. The 'Indexes' pane shows the 'PRIMARY' index on 'film_id'. The 'Views' pane shows several views like 'actor_info', 'customer_list', 'film_list', 'nicer_but_slower_film_list', 'sales_by_film_category', 'sales_by_store', 'staff_list', and 'film_in_stock'. The 'Administration' pane shows the 'payment' table with its columns: payment_id (smallint, UN AI PK), customer_id (smallint, UN), staff_id (tinyint, UN), rental_id (int), amount (decimal(5,2)), payment_date (datetime), and last_update (timestamp).

The main query editor shows the following SQL query:

```
1 select count(film_id)
2 from film
3 where rating='G';
4
```

The 'Result Grid' pane shows the query results:

| count(film_id) |
|----------------|
| 178 |

The 'Output' pane shows the execution log:

| # | Time | Action | Message | Duration / Fetch |
|---|----------|---|---------------------|-----------------------|
| 1 | 02:22:07 | select title, replacement_cost from film where replacement_cost = (select max(replac... | 53 row(s) returned | 0.000 sec / 0.000 sec |
| 2 | 02:26:17 | select title, rating from film where rating='G' | 178 row(s) returned | 0.000 sec / 0.000 sec |
| 3 | 02:26:39 | select count(film_id) from film where rating='G' | 1 row(s) returned | 0.000 sec / 0.000 sec |

9.

English

Italian

Japanese

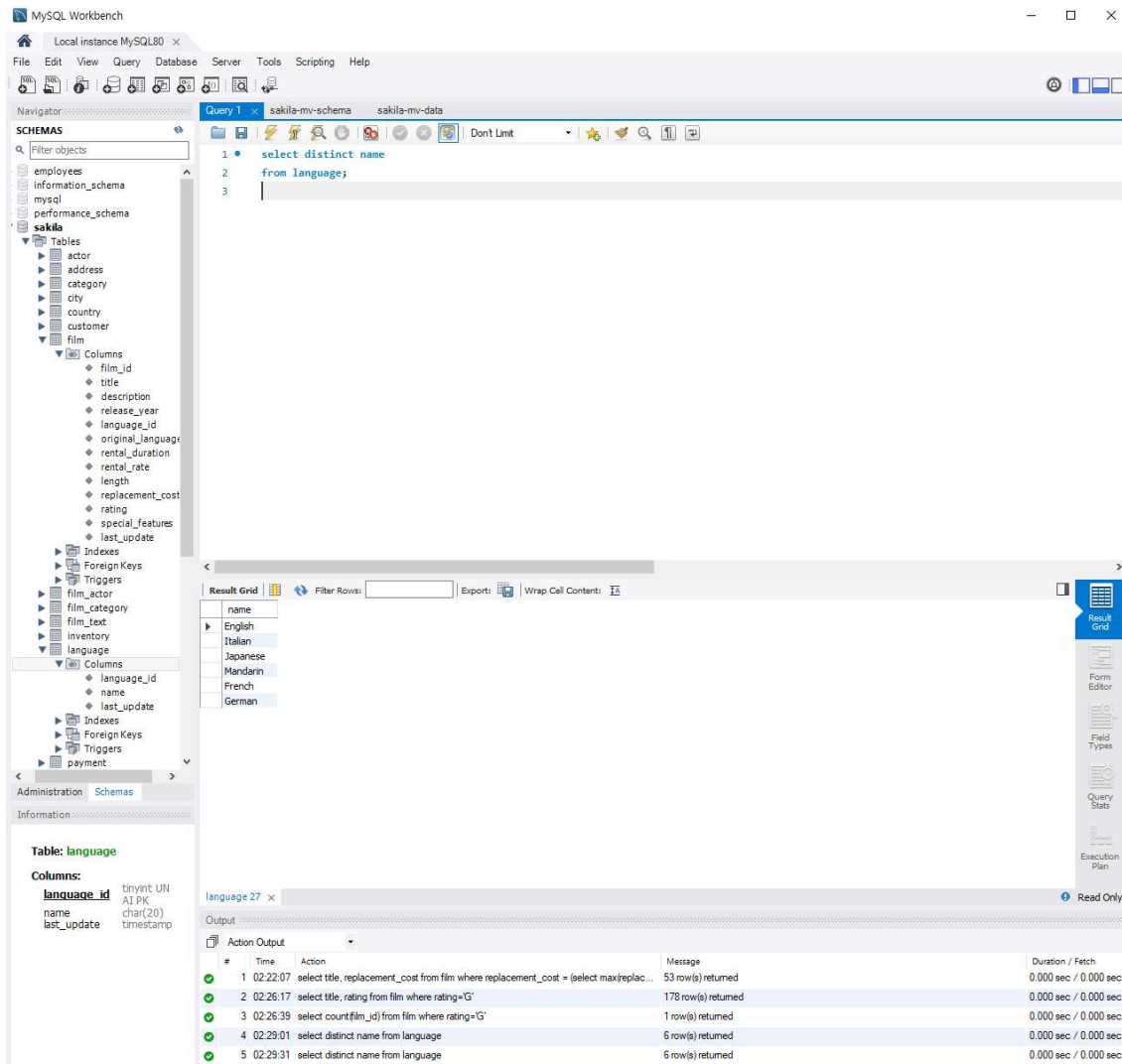
Mandarin

French

German

select distinct name

from language;



MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator sakila-mv-schema sakila-mv-data

Query 1

```
1 select distinct name
2 from language;
3
```

Result Grid

| name |
|----------|
| English |
| Italian |
| Japanese |
| Mandarin |
| French |
| German |

Table: language

Columns:

- language_id tinyint UNSIGNED AUTO INCREMENTAL PRIMARY KEY
- name char(20)
- last_update timestamp

Output

| # | Time | Action | Message | Duration / Fetch |
|---|----------|---|---------------------|-----------------------|
| 1 | 02:22:07 | select title, replacement_cost from film where replacement_cost = (select max(replac... | 53 row(s) returned | 0.000 sec / 0.000 sec |
| 2 | 02:26:17 | select title, rating from film where rating='G' | 178 row(s) returned | 0.000 sec / 0.000 sec |
| 3 | 02:26:39 | select count(film_id) from film where rating='G' | 1 row(s) returned | 0.000 sec / 0.000 sec |
| 4 | 02:29:01 | select distinct name from language | 6 row(s) returned | 0.000 sec / 0.000 sec |
| 5 | 02:29:31 | select distinct name from language | 6 row(s) returned | 0.000 sec / 0.000 sec |

10. 56 films for 'Horror'.

```
select count(fc.film_id)
```

```
from film_category fc, category c
```

```
where c.name='Horror' and fc.category_id = c.category_id;
```

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays the 'sakila' database structure, including tables like 'actor', 'address', 'category', 'city', 'country', 'customer', and 'film'. The 'film_category' table is selected, showing its columns: 'film_id', 'category_id', and 'last_update'. The main query editor displays the following SQL query:

```
1 select count(fc.film_id)
2 from film_category fc, category c
3 where c.name='Horror' and fc.category_id = c.category_id;
4
```

The 'Result Grid' pane shows the query result:

| count(fc.film_id) |
|-------------------|
| 56 |

The 'Table: film_category' pane shows the table's columns and their data types:

| Columns: | |
|-------------|----------------|
| film_id | smallint UN PK |
| category_id | tinyint UN PK |
| last_update | timestamp |

The 'Action Output' pane shows the execution details:

| # | Time | Action | Message | Duration / Fetch |
|---|----------|---|-------------------|-----------------------|
| 1 | 02:41:20 | select count(fc.film_id) from film_category fc, category c where c.name='Horror' and f... | 1 row(s) returned | 0.000 sec / 0.000 sec |

11. Yes

select *

from customer

where last_name='HIATT';

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'sakila' database selected. The 'customer' table is highlighted under the 'Tables' section. The main query editor contains the following SQL code:

```
1 select *
2 from customer
3 where last_name='HIATT';
4
```

The 'Result Grid' at the bottom shows the execution results for the query. The first row is the header, and the second row is the data for the customer with last_name 'HIATT'.

| customer_id | store_id | first_name | last_name | email | address_id | active | create_date | last_update |
|-------------|----------|------------|-----------|----------------------------------|------------|--------|---------------------|---------------------|
| 594 | 1 | EDUARDO | HIATT | EDUARDO.HIATT@sakilacustomer.org | 600 | 1 | 2006-02-14 22:04:37 | 2006-02-15 04:57:20 |

Below the result grid, the 'Table: customer' structure is shown:

Table: customer

Columns:

- customer_id: smallint(5) UNSIGNED ZEROFILL, PRIMARY KEY
- store_id: tinyint(4) UNSIGNED ZEROFILL
- first_name: varchar(45)
- last_name: varchar(45)
- email: varchar(50)
- address_id: smallint(5) UNSIGNED ZEROFILL
- active: tinyint(1) UNSIGNED ZEROFILL
- create_date: datetime
- last_update: timestamp

The 'Output' pane at the bottom shows the execution log with 6 rows of messages, including the successful execution of the query and the retrieval of 1 row.

12.

select f.film_id, f.title, a.first_name, a.last_name

from actor a, film f, film_actor fa

where a.last_name='SWANK' and a.first_name='JOE' and a.actor_id = fa.actor_id and
fa.film_id = f.film_id;

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays the 'sakila' database structure, including tables like 'actor', 'film', and 'film_actor'. The 'film_actor' table is selected, showing its columns: 'actor_id', 'film_id', and 'last_update'. The main query editor displays the following SQL query:

```
1 select f.film_id, f.title, a.first_name, a.last_name
2 from actor a, film f, film_actor fa
3 where a.last_name='SWANK' and a.first_name='JOE' and a.actor_id = fa.actor_id and fa.film_id = f.film_id;
4
```

The 'Result Grid' at the bottom shows the query results, which are 25 rows of data. The columns are 'film_id', 'title', 'first_name', and 'last_name'. The results are as follows:

| film_id | title | first_name | last_name |
|---------|---------------------|------------|-----------|
| 30 | ANYTHING SAVANNAH | JOE | SWANK |
| 74 | BIRCH ANTI-TRUST | JOE | SWANK |
| 147 | CHOCOLATE HARRY | JOE | SWANK |
| 148 | CHOCOLATE DUCK | JOE | SWANK |
| 191 | CROOKED FROGMEN | JOE | SWANK |
| 200 | CURTAIN VIDEOTAPE | JOE | SWANK |
| 204 | DALMATIANS SWEDEN | JOE | SWANK |
| 434 | HORROR REIGN | JOE | SWANK |
| 510 | LAWLESS VISION | JOE | SWANK |
| 514 | LEBOWSKI SOLDIERS | JOE | SWANK |
| 552 | MAJESTIC FLOATS | JOE | SWANK |
| 650 | PACIFIC AMISTAD | JOE | SWANK |
| 671 | PERDITION FARGO | JOE | SWANK |
| 697 | PRIMARY GLASS | JOE | SWANK |
| 722 | REEP SALUTE | JOE | SWANK |
| 752 | RUNNER MADIGAN | JOE | SWANK |
| 811 | SMILE EARRING | JOE | SWANK |
| 815 | SNATCHERS MONTEZ... | JOE | SWANK |

The 'Output' pane at the bottom shows the execution details: '1 02:50:49 select f.film_id, f.title, a.first_name, a.last_name from actor a, film f, film_actor fa wher... 25 row(s) returned'. The duration is 0.000 sec / 0.000 sec.

13. 28 films for 'Ralph Cruz'

```
select count(f.film_id)
```

```
from actor a, film f, film_actor fa
```

```
where a.last_name='CRUZ' and a.first_name='RALPH' and a.actor_id = fa.actor_id and  
fa.film_id = f.film_id;
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'sakila' database schema with tables like actor, film, and film_actor. The central query editor contains the following SQL query:

```
1 select count(f.film_id)
2 from actor a, film f, film_actor fa
3 where a.last_name='CRUZ' and a.first_name='RALPH' and a.actor_id = fa.actor_id and fa.film_id = f.film_id;
4
```

The 'Result Grid' at the bottom shows the query result:

| count(f.film_id) |
|------------------|
| 28 |

The bottom right pane shows the 'Action Output' for the query execution:

| # | Time | Action | Message | Duration / Fetch |
|---|----------|---|--------------------|-----------------------|
| 1 | 02:50:49 | select f.film_id, f.title, a.first_name, a.last_name from actor a, film f, film_actor fa where... | 25 row(s) returned | 0.000 sec / 0.000 sec |
| 2 | 02:52:16 | select f.film_id, f.title, a.first_name, a.last_name from actor a, film f, film_actor fa where... | 28 row(s) returned | 0.000 sec / 0.000 sec |
| 3 | 02:52:35 | select count(f.film_id) from actor a, film f, film_actor fa where a.last_name='CRUZ' an... | 1 row(s) returned | 0.000 sec / 0.000 sec |

14. 6 records in inventory

```
select count(i.film_id)
```

```
from inventory i, film f
```

```
where f.title = 'AGENT TRUMAN' and f.film_id = i.film_id;
```

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays the database structure, including tables like 'actor', 'address', 'category', 'city', 'country', 'customer', 'film', 'film_actor', 'film_category', 'film_text', 'inventory', 'language', 'payment', 'rental', 'staff', 'store', and 'actor_info'. The 'film' table is expanded, showing columns like 'film_id', 'title', 'description', 'release_year', 'language_id', 'original_language_id', 'rental_duration', 'rental_rate', 'length', 'replacement_cost', 'rating', 'special_features', and 'last_update'. The 'inventory' table is also expanded, showing columns like 'inventory_id', 'film_id', 'store_id', and 'last_update'.

The main query editor shows the following SQL query:

```
1 select count(i.film_id)
2 from inventory i, film f
3 where f.title = 'AGENT TRUMAN' and f.film_id = i.film_id;
4
5
```

The 'Result Grid' shows the query results:

| count(i.film_id) |
|------------------|
| 6 |

The 'Action Output' pane shows the execution details of the query:

| # | Time | Action | Message | Duration / Fetch |
|---|----------|---|----------------------|-----------------------|
| 1 | 02:50:49 | select f.film_id, f.title, a.first_name, a.last_name from actor a, film f, film_actor fa where... | 25 row(s) returned | 0.000 sec / 0.000 sec |
| 2 | 02:52:16 | select f.film_id, f.title, a.first_name, a.last_name from actor a, film f, film_actor fa where... | 28 row(s) returned | 0.000 sec / 0.000 sec |
| 3 | 02:52:35 | select count(f.film_id) from actor a, film f, film_actor fa where a.last_name='CRUZ' an... | 1 row(s) returned | 0.000 sec / 0.000 sec |
| 4 | 02:56:43 | select * from inventory i | 4581 row(s) returned | 0.000 sec / 0.000 sec |
| 5 | 02:58:44 | select f.* from inventory i, film f where f.title = 'AGENT TRUMAN' | 4581 row(s) returned | 0.000 sec / 0.000 sec |
| 6 | 02:59:11 | select f.* from film f where f.title = 'AGENT TRUMAN' | 1 row(s) returned | 0.000 sec / 0.000 sec |
| 7 | 02:59:47 | select f.* from inventory i, film f where f.title = 'AGENT TRUMAN' and f.film_id = i.film... | 6 row(s) returned | 0.000 sec / 0.000 sec |
| 8 | 02:59:53 | select i.* from inventory i, film f where f.title = 'AGENT TRUMAN' and f.film_id = i.film_id | 6 row(s) returned | 0.000 sec / 0.000 sec |
| 9 | 03:00:23 | select count(i.film_id) from inventory i, film f where f.title = 'AGENT TRUMAN' and f.fil... | 1 row(s) returned | 0.000 sec / 0.000 sec |

15.

Mike Hillyer

Jon Stephens

```
select s.first_name, s.last_name
```

```
from staff s;
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree with the 'sakila' database selected. The 'Tables' list under 'sakila' includes 'actor', 'address', 'category', 'city', 'country', 'customer', 'film', 'film_actor', 'film_category', 'film_text', 'inventory', 'language', 'payment', 'rental', and 'staff'. The 'Columns' list for 'staff' includes 'staff_id', 'first_name', 'last_name', 'address_id', 'picture', 'email', 'store_id', 'active', 'username', 'password', and 'last_update'. The 'Views' list includes 'actor_info', 'customer_list', 'film_list', 'nicer_but_slower_film_list', and 'sales_by_film_category'.

The central query editor shows the following SQL query:

```
1 select s.first_name, s.last_name
2 from staff s;
3
4
5
```

The 'Result Grid' at the bottom displays the results of the query:

| first_name | last_name |
|------------|-----------|
| Mike | Hillyer |
| Jon | Stephens |

The 'Output' pane at the bottom shows the execution details:

| # | Time | Action | Message | Duration / Fetch |
|---|----------|---|-------------------|-----------------------|
| 1 | 03:03:47 | select s.first_name, s.last_name from staff s | 2 row(s) returned | 0.000 sec / 0.000 sec |

16. The longest rental duration is 7

select f.title, f.rental_duration

from film f

where f.rental_duration=(select max(rental_duration)

from film);

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays a tree view of databases, with 'sakila' selected. Below it, the 'Tables' pane shows the 'film' table selected, and its columns are listed. The main query editor displays the following SQL query:

```
1 select f.title, f.rental_duration
2 from film f
3 where f.rental_duration=(select max(rental_duration)
4 from film);
5
```

The 'Result Grid' pane shows the results of the query, listing the title and rental_duration for each film. The results are as follows:

| title | rental_duration |
|---------------------|-----------------|
| ADAPTATION HOLES | 7 |
| ANONYMOUS HUMAN | 7 |
| ARGONAUTS TOWN | 7 |
| BKINI BORROWERS | 7 |
| BLACKOUT PRIVATE | 7 |
| BLANKET BEVERLY | 7 |
| BOILED DARES | 7 |
| BOONDOCK BALLROOM | 7 |
| BORN SPINAL | 7 |
| BORROWERS BEDAZZLED | 7 |
| BOWFINGER GABLES | 7 |
| BRAVEHEART HUMAN | 7 |
| BRIDE INTRIGUE | 7 |
| BRINGING HYSTERICAL | 7 |
| BROOKLYN DESERT | 7 |
| BUCKET BROTHERHOOD | 7 |
| CANYON STOCK | 7 |
| CARRIE BUNCH | 7 |

The 'Output' pane at the bottom shows the execution of the query, indicating that 191 rows were returned in 0.000 seconds.

17.

LINDA WILLIAMS

select c.first_name, c.last_name, c.address_id, a.address

from customer c, address a

where a.address = '692 Joliet Street' and a.address_id = c.address_id;

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator sakila-mv-schema sakila-mv-data

Query 1

```
1 select c.first_name, c.last_name, c.address_id, a.address
2 from customer c, address a
3 where a.address = '692 Joliet Street' and a.address_id = c.address_id;
4
5
```

Result Grid

| first_name | last_name | address_id | address |
|------------|-----------|------------|-------------------|
| LINDA | WILLIAMS | 7 | 692 Joliet Street |

Table: store

Columns:

- store_id tinyint UN AI
- manager_staff_id PK tinyint
- address_id smallint UN
- last_update timestamp

Result 69

Output

| # | Time | Action | Message | Duration / Fetch |
|---|----------|---|--|-----------------------|
| 1 | 03:36:41 | select f.title, f.rental_duration from film f where f.rental_duration=(select max(rental_d... | 191 row(s) returned | 0.000 sec / 0.000 sec |
| 2 | 03:38:33 | select c.first_name, c.last_name, c.address_id from customer c | 599 row(s) returned | 0.000 sec / 0.000 sec |
| 3 | 03:39:12 | select c.first_name, c.last_name, c.address_id, a.address, a.address2 from customer ... | Error Code: 1064. You have an error in your SQL syntax; check the manual that cor... | 0.000 sec |
| 4 | 03:39:27 | select c.first_name, c.last_name, c.address_id, a.address, a.address2 from customer... | 361197 row(s) returned | 0.000 sec / 0.094 sec |
| 5 | 03:39:44 | select c.first_name, c.last_name, c.address_id, a.address from customer c, address a | 361197 row(s) returned | 0.000 sec / 0.079 sec |
| 6 | 03:41:57 | select c.first_name, c.last_name, c.address_id, a.address from customer c, address ... | 1 row(s) returned | 0.000 sec / 0.000 sec |
| 7 | 03:44:08 | select c.first_name, c.last_name, c.address_id, a.address from customer c, address ... | 1 row(s) returned | 0.000 sec / 0.000 sec |

18.

select c.first_name, c.last_name, p.amount

from customer c, payment p

where p.amount = (select max(amount) from payment) and p.customer_id = c.customer_id;

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'sakila' database selected. The main query editor contains the following SQL query:

```
1 select c.first_name, c.last_name, p.amount
2 from customer c, payment p
3 where p.amount = (select max(amount) from payment) and p.customer_id = c.customer_id;
4
5
6
```

The 'Result Grid' at the bottom shows the results of the query, displaying columns 'first_name', 'last_name', and 'amount'. The results are as follows:

| first_name | last_name | amount |
|------------|------------|--------|
| KAREN | JACKSON | 11.99 |
| VICTORIA | GIBSON | 11.99 |
| VANESSA | SMITH | 11.99 |
| ALMA | AUSTIN | 11.99 |
| ROSEMARY | SCHMIDT | 11.99 |
| TANYA | GILBERT | 11.99 |
| RICHARD | MCCREARY | 11.99 |
| NICHOLAS | BARFIELD | 11.99 |
| KENT | ARSENVAULT | 11.99 |
| TERRANCE | ROUSH | 11.99 |

The bottom status bar indicates that the query was executed successfully, returning 10 rows in 0.000 seconds.

```
where p.amount = (select min(amount) from payment) and p.customer_id =
c.customer_id;
```

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The main window is titled 'Local instance MySQL80'. The Navigator pane on the left shows the database structure, including schemas, tables, columns, and indexes. The Query Editor in the center contains the following SQL query:

```

1 • select c.first_name, c.last_name, p.amount
2   from customer c, payment p
3  where p.amount = (select min(amount) from payment) and p.customer_id = c.customer_id;
4

```

The Results pane on the right shows the output of the query. It includes a 'Result Grid' with 24 rows and 3 columns: first_name, last_name, and amount. The data is as follows:

| first_name | last_name | amount |
|------------|-----------|--------|
| HELEN | HARRIS | 0.00 |
| CAROLYN | PEREZ | 0.00 |
| CHRISTINE | ROBERTS | 0.00 |
| HEATHER | MORRIS | 0.00 |
| MILDRED | BAILEY | 0.00 |
| TAMMY | SANDERS | 0.00 |
| TAMMY | SANDERS | 0.00 |
| FLORENCE | WOODS | 0.00 |
| GAIL | KNIGHT | 0.00 |
| CATHY | SPENCER | 0.00 |
| ANNETTE | OLSON | 0.00 |
| LUCY | WHEELER | 0.00 |
| NATALIE | MEYER | 0.00 |
| ALLISON | STANLEY | 0.00 |
| MARGIE | WADE | 0.00 |
| CASSANDRA | WALTERS | 0.00 |
| SONIA | GREGORY | 0.00 |
| JUSTIN | NGO | 0.00 |

The bottom status bar shows the execution time: 0.000 sec / 0.000 sec. The output pane shows the message: 'select c.first_name, c.last_name, p.amount from customer c, payment p where p.amount = (select min(amount) from payment) and p.customer_id = c.customer_id;'. The output also shows the number of rows returned: 24 row(s) returned.

20.

CUBA OLIVIER

28 movies

select distinct a.*, count(a.actor_id) as number_of_movies

from actor a, film_actor fa

where fa.actor_id = (select max(total) from (select count(*) as total from film_actor
group by film_id) as result) and

fa.actor_id = a.actor_id;

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays the database structure, including tables like 'actor', 'film', and 'film_actor'. The main query editor contains the following SQL code:

```
1 select distinct a.*, count(a.actor_id) as number_of_movies
2 from actor a, film_actor fa
3 where fa.actor_id = (select max(total) from (select count(*) as total from film_actor
4 group by film_id) as result) and
5 fa.actor_id = a.actor_id;
6
```

Below the query editor, the 'Result Grid' shows the output of the query:

| actor_id | first_name | last_name | last_update | number_of_movies |
|----------|------------|-----------|---------------------|------------------|
| 15 | CUBA | OLIVIER | 2006-02-15 04:34:33 | 28 |

At the bottom, the 'Output' pane shows the execution message: '1 04:29:59 select distinct a.*, count(a.actor_id) as number_of_movies from actor a, film_actor fa where fa.actor_id = (select ... 1 row(s) returned'.