Name : Sagar Sidhu

Email : sagarsidhu33321@gmail.com

Assignment Name : sql basics

Drive Link : N/A

Github Link : N/A

1. Create a table called employees with the following structure

emp_id (integer, should not be NULL and should be a primary key)

emp_name (text, should not be NULL)

age (integer, should have a check constraint to ensure the age is at least 18)

email (text, should be unique for each employee)

salary (decimal, with a default value of 30,000).

Write the SQL query to create the above table with all constraints.

Answer:

```
CREATE TABLE employees (
    emp_id INTEGER PRIMARY KEY NOT NULL,
    emp_name TEXT NOT NULL,
    age INTEGER CHECK (age >= 18),
    email TEXT UNIQUE,
    salary DECIMAL DEFAULT 30000
);
```

2. Explain the purpose of constraints and how they help maintain data integrity in a database. Provide examples of common types of constraints

Constraints are rules enforced on columns in a database table to ensure the accuracy, consistency, and integrity of the data stored.

Purpose of Constraints

- Ensure **valid data entry**
- Prevent **invalid or duplicate data**
- Maintain **relationships between tables**
- Enforce **business rules**

Common Types of Constraints (with Examples)

Constraint Type	**Purpose**	**Example**
NOT NULL	Ensures a column cannot have a NULL value	emp_name TEXT NOT NULL
UNIQUE	Ensures all values in a column are different	email TEXT UNIQUE
PRIMARY KEY	Uniquely identifies each row and cannot be NULL	emp_id INTEGER PRIMARY KEY
FOREIGN KEY	Maintains referential integrity between tables	FOREIGN KEY (dept_id) REFERENCES departments(dept_id)
CHECK	Ensures values meet a specific condition	age INTEGER CHECK (age >= 18)
DEFAULT	Assigns a default value if none is provided	salary DECIMAL DEFAULT 30000

3.Why would you apply the NOT NULL constraint to a column? Can a primary key contain NULL values? Justify our answer.

- The NOT NULL constraint is used when we want to make sure that a column always has a value. It helps prevent missing or empty data in important fields. For example, if we want every employee to have a name, we use NOT NULL on the emp_name column. A primary key cannot contain NULL values because it must uniquely identify each row in a table. If the value is NULL, it means it’s unknown, and we can’t use it to identify a row. That’s why a primary key is always automatically NOT NULL.

4.Explain the steps and SQL commands used to add or remove constraints on an existing table. Provide an example for both adding and removing a constraint.

Steps to **Add a Constraint**

Use ALTER TABLE followed by ADD CONSTRAINT.

Example: Add a UNIQUE constraint to the email column

ALTER TABLE employees

ADD CONSTRAINT unique_email **UNIQUE** (email);

Steps to **Remove a Constraint**

Use ALTER TABLE followed by DROP CONSTRAINT.

Note: You must know the **name of the constraint** to remove it. If you didn't name it while creating, the database gives it a default name.

Example: Remove the unique_email constraint

ALTER TABLE employees

DROP CONSTRAINT unique_email;

Summary

Operation	**Command Example**
Add Constraint	ALTER TABLE table_name ADD CONSTRAINT constraint_name constraint_definition;

| Drop Constraint| ALTER TABLE table_name DROP CONSTRAINT constraint_name;

5. Explain the consequences of attempting to insert, update, or delete data in a way that violates constraints. Provide an example of an error message that might occur when violating a constraint.

Constraints are rules set on table columns to protect data integrity. If you try to **insert**, **update**, or **delete** data in a way that breaks these rules, the database will block the operation and return an error.

Consequences of Violating Constraints

- **INSERT**: Adding a row with missing or invalid data will fail.
- **UPDATE**: Changing data that breaks a constraint will be rejected.
- **DELETE**: Deleting a record that is linked by a foreign key can cause errors.

Example 1: Violating NOT NULL

INSERT INTO employees (emp_id, emp_name, age)

VALUES (101, **NULL**, 25);

Error message:

```
ERROR: null value in column "emp_name" violates not-null constraint
```

Example 2: Violating UNIQUE

```
INSERT INTO employees (emp_id, emp_name, age, email)
VALUES (102, 'Alice', 30, 'john@example.com');  -- email already exists
```

Error message:

```
ERROR: duplicate key value violates unique constraint "unique_email"
```

6. You created a products table without constraints as follows:

```
CREATE TABLE products (
product_id INT,
product_name VARCHAR(50),
price DECIMAL(10, 2));
```

Now, you realise that

The product_id should be a primary key

The price should have a default value of 50.00

Answer:

```
ALTER TABLE products
ADD CONSTRAINT pk_product_id PRIMARY KEY (product_id);


ALTER TABLE products
ALTER COLUMN price SET DEFAULT 50.00;
```

7. You have two tables:

Students Table

student_id	student_name	class_id
1	Alice	101

student_id	student_name	class_id
2	Bob	102
3	Charlie	101

Classes Table

class_id	class_name
101	Math
102	Science
103	History

Answer:

Write a query to fetch the student_name and class_name for each student using an INNER JOIN.

```
SELECT
  students.student_name,classes.class_name
FROM
  students INNER JOIN classes
ON
  students.class_id = classes.class_id;
```

8. Consider the following three tables:

Orders Table

order_id	order_date	customer_id
1	2024-01-01	101
2	2024-01-03	102

Customers Table

customer_id	customer_name
101	Alice
102	Bob

Products Table

product_id	product_name	order_id
1	Laptop	1
2	Phone	NULL

Write a query that shows all order_id, customer_name, and product_name, ensuring that all products are listed even if they are not associated with an order ,Hint: (use INNER JOIN and LEFT JOIN)

Answer:

```
SELECT
  products.order_id,
  customers.customer_name,
  products.product_name
FROM
  products
LEFT JOIN orders
  ON products.order_id = orders.order_id
LEFT JOIN customers
  ON orders.customer_id = customers.customer_id;
```

9.Given the following tables:

Sales Table

sale_id	product_id	amount
1	101	500
2	102	300
3	101	700

Products Table

product_id	product_name
101	Laptop
102	Phone

Write a query to find the total sales amount for each product using an INNER JOIN and the SUM() function

---

Answer:

```
SELECT
  p.product_name,
  SUM(s.amount) AS total_sales
FROM
  sales s
INNER JOIN
  products p
ON
  s.product_id = p.product_id
GROUP BY
  p.product_name;
```

10. You are given three tables:

Orders Table

order_id	order_date	customer_id
1	2024-01-02	1
2	2024-01-05	2

Customers Table

customer_id	customer_name
1	Alice
2	Bob

Order_Details Table

order_id	product_id	quantity
1	101	2
1	102	1
2	101	3

Write a query to display the order_id, customer_name, and the quantity of products ordered by each customer using an INNER JOIN between all three tables.

Answer:

```
SELECT
   o.order_id,c.customer_name,od.quantity
FROM orders o
INNER JOIN customers c
   ON o.customer_id = c.customer_id
INNER JOIN order_details od
   ON o.order_id = od.order_id;
```

SQL Commands

1.Identify the primary keys and foreign keys in maven movies db. Discuss the differences

Primary Keys

Primary keys uniquely identify each row in a table. They cannot be NULL or duplicated.

Table	**Primary Key**
actor	actor_id
address	address_id
category	category_id
city	city_id
country	country_id
customer	customer_id
film	film_id
inventory	inventory_id
language	language_id
payment	payment_id
rental	rental_id
staff	staff_id
store	store_id
film_actor	(film_id, actor_id)
film_category	(film_id, category_id)

Foreign Keys

Foreign keys link records between tables using referenced primary keys.

Table	Foreign Key(s)	References
address	city_id	city(city_id)
city	country_id	country(country_id)
customer	store_id, address_id	store, address
film	language_id, original_language_id	language
film_actor	film_id, actor_id	film, actor
film_category	film_id, category_id	film, category
inventory	film_id, store_id	film, store
payment	customer_id, staff_id, rental_id	customer, staff, rental
rental	inventory_id, customer_id, staff_id	inventory, customer, staff
staff	address_id, store_id	address, store
store	address_id, manager_staff_id	address, staff

Primary Key vs Foreign Key

Feature	Primary Key	Foreign Key
Purpose	Uniquely identifies each record	Connects one table to another
Uniqueness	Must be unique	Can have duplicates
NULL allowed	Not allowed	Allowed (depending on the relationship)

Feature	Primary Key	Foreign Key
Used in	The table it belongs to	Refers to another table

---

2. List all details of actors

```
%%sql
SELECT * FROM actor limit 10;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
10 rows affected.
```

actor_id	first_name	last_name	last_update
1	PENELOPE	GUINESS	2006-02-15 04:34:33
2	NICK	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
6	BETTE	NICHOLSON	2006-02-15 04:34:33
7	GRACE	MOSTEL	2006-02-15 04:34:33
8	MATTHEW	JOHANSSON	2006-02-15 04:34:33
9	JOE	SWANK	2006-02-15 04:34:33
10	CHRISTIAN	GABLE	2006-02-15 04:34:33

3.List all customer information from DB

```
%%sql
SELECT * FROM customer limit 10;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
10 rows affected.
```

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
1	1	MARY	SMITH	MARY.SMITH@sakilacustomer.org	5	1	2006-02-14 22:04:36	2006-02-15 04:57:20
2	1	PATRICIA	JOHNSON	PATRICIA.JOHNSON@sakilacustomer.org	6	1	2006-02-14 22:04:36	2006-02-15 04:57:20
3	1	LINDA	WILLIAMS	LINDA.WILLIAMS@sakilacustomer.org	7	1	2006-02-14 22:04:36	2006-02-15 04:57:20
4	2	BARBARA	JONES	BARBARA.JONES@sakilacustomer.org	8	1	2006-02-14 22:04:36	2006-02-15 04:57:20
5	1	ELIZABETH	BROWN	ELIZABETH.BROWN@sakilacustomer.org	9	1	2006-02-14 22:04:36	2006-02-15 04:57:20
6	2	JENNIFER	DAVIS	JENNIFER.DAVIS@sakilacustomer.org	10	1	2006-02-14 22:04:36	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
7	1	MARIA	MILLER	MARIA.MILLER@sakilacustomer.org	11	1	2006-02-14 22:04:36	2006-02-15 04:57:20
8	2	SUSAN	WILSON	SUSAN.WILSON@sakilacustomer.org	12	1	2006-02-14 22:04:36	2006-02-15 04:57:20
9	2	MARGARET	MOORE	MARGARET.MOORE@sakilacustomer.org	13	1	2006-02-14 22:04:36	2006-02-15 04:57:20
10	1	DOROTHY	TAYLOR	DOROTHY.TAYLOR@sakilacustomer.org	14	1	2006-02-14 22:04:36	2006-02-15 04:57:20

4.List different countries.

```
%%sql
SELECT DISTINCT country FROM country;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
109 rows affected.
```

country
Afghanistan
Algeria
American Samoa

country
Angola
Anguilla
Argentina
Armenia
Australia
Austria
Azerbaijan
Bahrain
Bangladesh
Belarus
Bolivia
Brazil
Brunei
Bulgaria
Cambodia
Cameroon
Canada

country

Chad

Chile

China

Colombia

Congo, The Democratic Republic of the

Czech Republic

Dominican Republic

Ecuador

Egypt

Estonia

Ethiopia

Faroe Islands

Finland

France

French Guiana

French Polynesia

Gambia

country
Germany
Greece
Greenland
Holy See (Vatican City State)
Hong Kong
Hungary
India
Indonesia
Iran
Iraq
Israel
Italy
Japan
Kazakstan
Kenya
Kuwait
Latvia

country
Liechtenstein
Lithuania
Madagascar
Malawi
Malaysia
Mexico
Moldova
Morocco
Mozambique
Myanmar
Nauru
Nepal
Netherlands
New Zealand
Nigeria
North Korea
Oman

country
Pakistan
Paraguay
Peru
Philippines
Poland
Puerto Rico
Romania
Runion
Russian Federation
Saint Vincent and the Grenadines
Saudi Arabia
Senegal
Slovakia
South Africa
South Korea
Spain
Sri Lanka

country
Sudan
Sweden
Switzerland
Taiwan
Tanzania
Thailand
Tonga
Tunisia
Turkey
Turkmenistan
Tuvalu
Ukraine
United Arab Emirates
United Kingdom
United States
Venezuela
Vietnam

country
Virgin Islands, U.S.
Yemen
Yugoslavia
Zambia

5.Display all active customers.

```
%%sql
SELECT * FROM customer
WHERE active = 1 limit 10;# use limit to avoid large output
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
10 rows affected.
```

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
1	1	MARY	SMITH	MARY.SMITH@sakilacustomer.org	5	1	2006-02-14 22:04:36	2006-02-15 04:57:20
2	1	PATRICIA	JOHNSON	PATRICIA.JOHNSON@sakilacustomer.org	6	1	2006-02-14 22:04:36	2006-02-15 04:57:20
3	1	LINDA	WILLIAMS	LINDA.WILLIAMS@sakilacustomer.org	7	1	2006-02-14 22:04:36	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
4	2	BARBARA	JONES	BARBARA.JONES@sakilacustomer.org	8	1	2006-02-14 22:04:36	2006-02-15 04:57:20
5	1	ELIZABETH	BROWN	ELIZABETH.BROWN@sakilacustomer.org	9	1	2006-02-14 22:04:36	2006-02-15 04:57:20
6	2	JENNIFER	DAVIS	JENNIFER.DAVIS@sakilacustomer.org	10	1	2006-02-14 22:04:36	2006-02-15 04:57:20
7	1	MARIA	MILLER	MARIA.MILLER@sakilacustomer.org	11	1	2006-02-14 22:04:36	2006-02-15 04:57:20
8	2	SUSAN	WILSON	SUSAN.WILSON@sakilacustomer.org	12	1	2006-02-14 22:04:36	2006-02-15 04:57:20
9	2	MARGARET	MOORE	MARGARET.MOORE@sakilacustomer.org	13	1	2006-02-14 22:04:36	2006-02-15 04:57:20
10	1	DOROTHY	TAYLOR	DOROTHY.TAYLOR@sakilacustomer.org	14	1	2006-02-14 22:04:36	2006-02-15 04:57:20

6.-List of all rental IDs for customer with ID 1.

```
%%sql
SELECT rental_id FROM rental
WHERE customer_id = 1;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
32 rows affected.
```

rental_id
76
573
1185
1422
1476
1725
2308
2363
3284
4526
4611
5244
5326

rental_id
6163
7273
7841
8033
8074
8116
8326
9571
10437
11299
11367
11824
12250
13068
13176
14762
14825

rental_id
15298
15315

7.Display all the films whose rental duration is greater than 5 .

```
%%sql
SELECT * FROM film
WHERE rental_duration > 5 limit 10;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
10 rows affected.
```

film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features	last_update
1	ACADEMY DINOSAUR	A Epic Drama of a Feminist And a Mad Scientist who must Battle a Teacher	2006	1	None	6	0.99	86	20.99	PG	{'Deleted Scenes', 'Behind the Scenes'}	2006-02-15 05:03:42

film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features	last_update
		in The Canadian Rockies										
3	ADAPTATION HOLES	A Astounding Reflection of a Lumberjack And a Car who must Sink a Lumberjack in A Baloon Factory	2006	1	None	7	2.99	50	18.99	NC-17	{'Trailers', 'Deleted Scenes'}	2006-02-15 05:03:42

film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features	last_update
5	AFRICAN EGG	A Fast-Paced Documentary of a Pastry Chef And a Dentist who must Pursue a Forensic Psychologist in The Gulf of Mexico	2006	1	None	6	2.99	130	22.99	G	{'Deleted Scenes'}	2006-02-15 05:03:42

film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features	last_update
7	AIRPLANE SIERRA	A Touching Saga of a Hunter And a Butler who must Discover a Butler in A Jet Boat	2006	1	None	6	4.99	62	28.99	PG-13	{'Trailers', 'Deleted Scenes'}	2006-02-15 05:03:42
8	AIRPORT POLLOCK	A Epic Tale of a Moose And a Girl who must Conf	2006	1	None	6	4.99	54	15.99	R	{'Trailers'}	2006-02-15 05:03:42

film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features	last_update
		ront a Monkey in Ancient India										
10	ALADDIN CALENDAR	A Action-Packed Tale of a Man And a Lumberjack who must Reach a Feminist in Ancient China	2006	1	None	6	4.99	63	24.99	NC-17	{'Trailers', 'Deleted Scenes'}	2006-02-15 05:03:42

film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features	last_update
11	ALAMO VIDEOTAPE	A Boring Epistle of a Butler And a Cat who must Fight a Pastry Chef in A MySQL Convention	2006	1	None	6	0.99	126	16.99	G	{'Commentaries', 'Behind the Scenes'}	2006-02-15 05:03:42
12	ALASKA PHANTOM	A Fanciful Saga of a Hunter And a	2006	1	None	6	0.99	136	22.99	PG	{'Commentaries', 'Deleted Scenes'}	2006-02-15 05:03:42

film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features	last_update
		Pastry Chef who must Vanquish a Boy in Australia										
14	ALICE FANTASIA	A Emotional Drama of a A Shark And a Database Administrator who must Van	2006	1	None	6	0.99	94	23.99	NC-17	{'Trailers', 'Deleted Scenes', 'Behind the Scenes'}	2006-02-15 05:03:42

film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features	last_update
		quish a Pioneer in Soviet Georgia										
16	ALLEY EVOLUTION	A Fast-Paced Drama of a Robot And a Composer who must Battle a Astronaut in New	2006	1	None	6	2.99	180	23.99	NC-17	{'Trailers', 'Commentaries'}	2006-02-15 05:03:42

film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features	last_update
		Orleans										

8.- List the total number of films whose replacement cost is greater than 20.

```
%%sql
SELECT COUNT(*) AS total_films
FROM film
WHERE replacement_cost > 15 AND replacement_cost < 20;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
1 rows affected.
```

total_films
214

9. Display the count of unique first names of actors.

```
%%sql
SELECT
   COUNT(DISTINCT first_name) AS unique_first_name_count
FROM
   actor;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
1 rows affected.
```

unique_first_name_count
128

10.Display the first 10 records from the customer table .

```
%%sql
SELECT * FROM customer
LIMIT 10;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
10 rows affected.
```

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
1	1	MARY	SMITH	MARY.SMITH@sakilacustomer.org	5	1	2006-02-14 22:04:36	2006-02-15 04:57:20
2	1	PATRICIA	JOHNSON	PATRICIA.JOHNSON@sakilacustomer.org	6	1	2006-02-14 22:04:36	2006-02-15 04:57:20
3	1	LINDA	WILLIAMS	LINDA.WILLIAMS@sakilacustomer.org	7	1	2006-02-14 22:04:36	2006-02-15 04:57:20
4	2	BARBARA	JONES	BARBARA.JONES@sakilacustomer.org	8	1	2006-02-14 22:04:36	2006-02-15 04:57:20
5	1	ELIZABETH	BROWN	ELIZABETH.BROWN@sakilacustomer.org	9	1	2006-02-14 22:04:36	2006-02-15 04:57:20
6	2	JENNIFER	DAVIS	JENNIFER.DAVIS@sakilacustomer.org	10	1	2006-02-14	2006-02-15

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
							22:04:36	04:57:20
7	1	MARIA	MILLER	MARIA.MILLER@sakilacustomer.org	11	1	2006-02-14 22:04:36	2006-02-15 04:57:20
8	2	SUSAN	WILSON	SUSAN.WILSON@sakilacustomer.org	12	1	2006-02-14 22:04:36	2006-02-15 04:57:20
9	2	MARGARET	MOORE	MARGARET.MOORE@sakilacustomer.org	13	1	2006-02-14 22:04:36	2006-02-15 04:57:20
10	1	DOROTHY	TAYLOR	DOROTHY.TAYLOR@sakilacustomer.org	14	1	2006-02-14 22:04:36	2006-02-15 04:57:20

11.Display the first 3 records from the customer table whose first name starts with ‘b’.

```
%%sql
SELECT * FROM customer
WHERE first_name LIKE 'B%'
LIMIT 3;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
3 rows affected.
```

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
4	2	BARBARA	JONES	BARBARA.JONES@sakilacustomer.org	8	1	2006-02-14 22:04:36	2006-02-15 04:57:20
14	2	BETTY	WHITE	BETTY.WHITE@sakilacustomer.org	18	1	2006-02-14 22:04:36	2006-02-15 04:57:20
31	2	BRENDA	WRIGHT	BRENDA.WRIGHT@sakilacustomer.org	35	1	2006-02-14 22:04:36	2006-02-15 04:57:20

12.Display the names of the first 5 movies which are rated as ‘G’.

```
%%sql
SELECT title
FROM film
WHERE rating = 'G'
LIMIT 5;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
5 rows affected.
```

title
ACE GOLDFINGER
AFFAIR PREJUDICE
AFRICAN EGG

title
ALAMO VIDEOTAPE
AMISTAD MIDSUMMER

13.Find all customers whose first name starts with "a".

```
%%sql
SELECT * FROM customer
WHERE first_name LIKE 'A%';
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
44 rows affected.
```

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
29	2	ANGELA	HERNANDEZ	ANGELA.HERNANDEZ@sakilacustomer.org	33	1	2006-02-14 22:04:36	2006-02-15 04:57:20
32	1	AMY	LOPEZ	AMY.LOPEZ@sakilacustomer.org	36	1	2006-02-14 22:04:36	2006-02-15 04:57:20
33	2	ANNA	HILL	ANNA.HILL@sakilacustomer.org	37	1	2006-02-14 22:04:36	2006-02-15 04:57:20
40	2	AMANDA	CARTER	AMANDA.CARTER@sakilacustomer.org	44	1	2006-02-14 22:04:36	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
48	1	ANN	EVANS	ANN.EVANS@sakilacustomer.org	52	1	2006-02-14 22:04:36	2006-02-15 04:57:20
51	1	ALICE	STEWART	ALICE.STEWART@sakilacustomer.org	55	1	2006-02-14 22:04:36	2006-02-15 04:57:20
63	1	ASHLEY	RICHARDSON	ASHLEY.RICHARDSON@sakilacustomer.org	67	1	2006-02-14 22:04:36	2006-02-15 04:57:20
81	1	ANDREA	HENDERSON	ANDREA.HENDERSON@sakilacustomer.org	85	1	2006-02-14 22:04:36	2006-02-15 04:57:20
85	2	ANNE	POWELL	ANNE.POWELL@sakilacustomer.org	89	1	2006-02-14 22:04:36	2006-02-15 04:57:20
97	2	ANNIE	RUSSELL	ANNIE.RUSSELL@sakilacustomer.org	101	1	2006-02-14 22:04:36	2006-02-15 04:57:20
136	2	ANITA	MORALES	ANITA.MORALES@sakilacustomer.org	140	1	2006-02-14 22:04:36	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
139	1	AMBER	DIXON	AMBER.DIXON@sakilacustomer.org	143	1	2006-02-14 22:04:36	2006-02-15 04:57:20
142	1	APRIL	BURNS	APRIL.BURNS@sakilacustomer.org	146	1	2006-02-14 22:04:36	2006-02-15 04:57:20
152	1	ALICIA	MILLS	ALICIA.MILLS@sakilacustomer.org	156	1	2006-02-14 22:04:36	2006-02-15 04:57:20
173	1	AUDREY	RAY	AUDREY.RAY@sakilacustomer.org	177	1	2006-02-14 22:04:36	2006-02-15 04:57:20
175	1	ANNETTE	OLSON	ANNETTE.OLSON@sakilacustomer.org	179	1	2006-02-14 22:04:36	2006-02-15 04:57:20
181	2	ANA	BRADLEY	ANA.BRADLEY@sakilacustomer.org	185	1	2006-02-14 22:04:36	2006-02-15 04:57:20
196	1	ALMA	AUSTIN	ALMA.AUSTIN@sakilacustomer.org	200	1	2006-02-14 22:04:36	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
217	2	AGNES	BISHOP	AGNES.BISHOP@sakilacustomer.org	221	1	2006-02-14 22:04:36	2006-02-15 04:57:20
225	1	ARLENE	HARVEY	ARLENE.HARVEY@sakilacustomer.org	229	1	2006-02-14 22:04:36	2006-02-15 04:57:20
228	2	ALLISON	STANLEY	ALLISON.STANLEY@sakilacustomer.org	232	1	2006-02-14 22:04:36	2006-02-15 04:57:20
320	2	ANTHONY	SCHWAB	ANTHONY.SCHWAB@sakilacustomer.org	325	1	2006-02-14 22:04:37	2006-02-15 04:57:20
333	2	ANDREW	PURDY	ANDREW.PURDY@sakilacustomer.org	338	1	2006-02-14 22:04:37	2006-02-15 04:57:20
346	1	ARTHUR	SIMPKINS	ARTHUR.SIMPKINS@sakilacustomer.org	351	1	2006-02-14 22:04:37	2006-02-15 04:57:20
352	1	ALBERT	CROUSE	ALBERT.CROUSE@sakilacustomer.org	357	1	2006-02-14 22:04:37	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
367	1	ADAM	GOOCH	ADAM.GOOCH@sakilacustomer.org	372	1	2006-02-14 22:04:37	2006-02-15 04:57:20
375	2	AARON	SELBY	AARON.SELBY@sakilacustomer.org	380	1	2006-02-14 22:04:37	2006-02-15 04:57:20
389	1	ALAN	KAHN	ALAN.KAHN@sakilacustomer.org	394	1	2006-02-14 22:04:37	2006-02-15 04:57:20
398	1	ANTONIO	MEEK	ANTONIO.MEEK@sakilacustomer.org	403	1	2006-02-14 22:04:37	2006-02-15 04:57:20
412	2	ALLEN	BUTTERFIELD	ALLEN.BUTTERFIELD@sakilacustomer.org	417	1	2006-02-14 22:04:37	2006-02-15 04:57:20
423	2	ALFRED	CASILLAS	ALFRED.CASILLAS@sakilacustomer.org	428	1	2006-02-14 22:04:37	2006-02-15 04:57:20
439	2	ALEXANDER	FENNELL	ALEXANDER.FENNELL@sakilacustomer.org	444	1	2006-02-14 22:04:37	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
454	2	ALEX	GRESHAM	ALEX.GRESHAM@sakilacustomer.org	459	1	2006-02-14 22:04:37	2006-02-15 04:57:20
467	2	ALVIN	DELOACH	ALVIN.DELOACH@sakilacustomer.org	472	1	2006-02-14 22:04:37	2006-02-15 04:57:20
503	1	ANGEL	BARCLAY	ANGEL.BARCLAY@sakilacustomer.org	508	1	2006-02-14 22:04:37	2006-02-15 04:57:20
515	1	ANDRE	RAPP	ANDRE.RAPP@sakilacustomer.org	521	1	2006-02-14 22:04:37	2006-02-15 04:57:20
522	2	ARNOLD	HAVENS	ARNOLD.HAVENS@sakilacustomer.org	528	1	2006-02-14 22:04:37	2006-02-15 04:57:20
525	2	ADRIAN	CLARY	ADRIAN.CLARY@sakilacustomer.org	531	1	2006-02-14 22:04:37	2006-02-15 04:57:20
548	1	ALLAN	CORNISH	ALLAN.CORNISH@sakilacustomer.org	554	1	2006-02-14 22:04:37	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
556	2	ARMANDO	GRUBER	ARMANDO.GRUBER@sakilacustomer.org	562	1	2006-02-14 22:04:37	2006-02-15 04:57:20
567	2	ALFREDO	MCADAMS	ALFREDO.MCADAMS@sakilacustomer.org	573	1	2006-02-14 22:04:37	2006-02-15 04:57:20
568	2	ALBERTO	HENNING	ALBERTO.HENNING@sakilacustomer.org	574	1	2006-02-14 22:04:37	2006-02-15 04:57:20
582	2	ANDY	VANHORN	ANDY.VANHORN@sakilacustomer.org	588	1	2006-02-14 22:04:37	2006-02-15 04:57:20
599	2	AUSTIN	CINTRON	AUSTIN.CINTRON@sakilacustomer.org	605	1	2006-02-14 22:04:37	2006-02-15 04:57:20

14.- Find all customers whose first name ends with "a"

```
%%sql

SELECT * FROM customer

WHERE first_name LIKE '%a' limit 20;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies

20 rows affected.
```

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
2	1	PATRICIA	JOHNSON	PATRICIA.JOHNSON@sakilacustomer.org	6	1	2006-02-14 22:04:36	2006-02-15 04:57:20
3	1	LINDA	WILLIAMS	LINDA.WILLIAMS@sakilacustomer.org	7	1	2006-02-14 22:04:36	2006-02-15 04:57:20
4	2	BARBARA	JONES	BARBARA.JONES@sakilacustomer.org	8	1	2006-02-14 22:04:36	2006-02-15 04:57:20
7	1	MARIA	MILLER	MARIA.MILLER@sakilacustomer.org	11	1	2006-02-14 22:04:36	2006-02-15 04:57:20
11	2	LISA	ANDERSON	LISA.ANDERSON@sakilacustomer.org	15	1	2006-02-14 22:04:36	2006-02-15 04:57:20
16	2	SANDRA	MARTIN	SANDRA.MARTIN@sakilacustomer.org	20	0	2006-02-14 22:04:36	2006-02-15 04:57:20
17	1	DONNA	THOMPSON	DONNA.THOMPSON@sakilacustomer.org	21	1	2006-02-14 22:04:36	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
22	1	LAURA	RODRIGUEZ	LAURA.RODRIGUEZ@sakilacustomer.org	26	1	2006-02-14 22:04:36	2006-02-15 04:57:20
26	2	JESSICA	HALL	JESSICA.HALL@sakilacustomer.org	30	1	2006-02-14 22:04:36	2006-02-15 04:57:20
28	1	CYNTHIA	YOUNG	CYNTHIA.YOUNG@sakilacustomer.org	32	1	2006-02-14 22:04:36	2006-02-15 04:57:20
29	2	ANGELA	HERNANDEZ	ANGELA.HERNANDEZ@sakilacustomer.org	33	1	2006-02-14 22:04:36	2006-02-15 04:57:20
30	1	MELISSA	KING	MELISSA.KING@sakilacustomer.org	34	1	2006-02-14 22:04:36	2006-02-15 04:57:20
31	2	BRENDA	WRIGHT	BRENDA.WRIGHT@sakilacustomer.org	35	1	2006-02-14 22:04:36	2006-02-15 04:57:20
33	2	ANNA	HILL	ANNA.HILL@sakilacustomer.org	37	1	2006-02-14 22:04:36	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
34	2	REBECCA	SCOTT	REBECCA.SCOTT@sakilacustomer.org	38	1	2006-02-14 22:04:36	2006-02-15 04:57:20
35	2	VIRGINIA	GREEN	VIRGINIA.GREEN@sakilacustomer.org	39	1	2006-02-14 22:04:36	2006-02-15 04:57:20
37	1	PAMELA	BAKER	PAMELA.BAKER@sakilacustomer.org	41	1	2006-02-14 22:04:36	2006-02-15 04:57:20
38	1	MARTHA	GONZALEZ	MARTHA.GONZALEZ@sakilacustomer.org	42	1	2006-02-14 22:04:36	2006-02-15 04:57:20
39	1	DEBRA	NELSON	DEBRA.NELSON@sakilacustomer.org	43	1	2006-02-14 22:04:36	2006-02-15 04:57:20
40	2	AMANDA	CARTER	AMANDA.CARTER@sakilacustomer.org	44	1	2006-02-14 22:04:36	2006-02-15 04:57:20

15.Display the list of first 4 cities which start and end with ‘a’

```
%%sql

SELECT city FROM city

WHERE city LIKE 'a%a'
```

```
LIMIT 4;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
4 rows affected.
```

city
Abha
Acua
Adana
Addis Abeba

16.Find all customers whose first name have "NI" in any position.

```
%%sql
SELECT * FROM customer
WHERE first_name LIKE '%NI%';
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
29 rows affected.
```

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
6	2	JENNIFER	DAVIS	JENNIFER.DAVIS@sakilacustomer.org	10	1	2006-02-14 22:04:36	2006-02-15 04:57:20
35	2	VIRGINIA	GREEN	VIRGINIA.GREEN@sakilacustomer.org	39	1	2006-02-14 22:04:36	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
41	1	STEPHANIE	MITCHELL	STEPHANIE.MITCHELL@sakilacustomer.org	45	1	2006-02-14 22:04:36	2006-02-15 04:57:20
66	2	JANICE	WARD	JANICE.WARD@sakilacustomer.org	70	1	2006-02-14 22:04:36	2006-02-15 04:57:20
68	1	NICOLE	PETERSON	NICOLE.PETERSON@sakilacustomer.org	72	1	2006-02-14 22:04:36	2006-02-15 04:57:20
74	1	DENISE	KELLY	DENISE.KELLY@sakilacustomer.org	78	1	2006-02-14 22:04:36	2006-02-15 04:57:20
88	2	BONNIE	HUGHES	BONNIE.HUGHES@sakilacustomer.org	92	1	2006-02-14 22:04:36	2006-02-15 04:57:20
97	2	ANNIE	RUSSELL	ANNIE.RUSSELL@sakilacustomer.org	101	1	2006-02-14 22:04:36	2006-02-15 04:57:20
106	1	CONNIE	WALLACE	CONNIE.WALLACE@sakilacustomer.org	110	1	2006-02-14 22:04:36	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
131	2	MONICA	HICKS	MONICA.HICKS@sakilacustomer.org	135	1	2006-02-14 22:04:36	2006-02-15 04:57:20
135	2	JUANITA	MASON	JUANITA.MASON@sakilacustomer.org	139	1	2006-02-14 22:04:36	2006-02-15 04:57:20
136	2	ANITA	MORALES	ANITA.MORALES@sakilacustomer.org	140	1	2006-02-14 22:04:36	2006-02-15 04:57:20
150	2	DANIELLE	DANIELS	DANIELLE.DANIELS@sakilacustomer.org	154	1	2006-02-14 22:04:36	2006-02-15 04:57:20
158	1	VERONICA	STONE	VERONICA.STONE@sakilacustomer.org	162	1	2006-02-14 22:04:36	2006-02-15 04:57:20
172	1	BERNICE	WILLIS	BERNICE.WILLIS@sakilacustomer.org	176	1	2006-02-14 22:04:36	2006-02-15 04:57:20
188	1	MELANIE	ARMSTRONG	MELANIE.ARMSTRONG@sakilacustomer.org	192	1	2006-02-14 22:04:36	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
239	2	MINNIE	ROMERO	MINNIE.ROMERO@sakilacustomer.org	243	1	2006-02-14 22:04:36	2006-02-15 04:57:20
265	2	JENNIE	TERRY	JENNIE.TERRY@sakilacustomer.org	270	1	2006-02-14 22:04:36	2006-02-15 04:57:20
268	1	NINA	SOTO	NINA.SOTO@sakilacustomer.org	273	1	2006-02-14 22:04:36	2006-02-15 04:57:20
284	1	SONIA	GREGORY	SONIA.GREGORY@sakilacustomer.org	289	1	2006-02-14 22:04:37	2006-02-15 04:57:20
291	1	TONI	HOLT	TONI.HOLT@sakilacustomer.org	296	1	2006-02-14 22:04:37	2006-02-15 04:57:20
310	2	DANIEL	CABRAL	DANIEL.CABRAL@sakilacustomer.org	315	1	2006-02-14 22:04:37	2006-02-15 04:57:20
338	1	DENNIS	GILMAN	DENNIS.GILMAN@sakilacustomer.org	343	1	2006-02-14 22:04:37	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
362	1	NICHOLAS	BARFIELD	NICHOLAS.BARFIELD@sakilacustomer.org	367	1	2006-02-14 22:04:37	2006-02-15 04:57:20
398	1	ANTONIO	MEEK	ANTONIO.MEEK@sakilacustomer.org	403	1	2006-02-14 22:04:37	2006-02-15 04:57:20
456	2	RONNIE	RICKETTS	RONNIE.RICKETTS@sakilacustomer.org	461	1	2006-02-14 22:04:37	2006-02-15 04:57:20
504	1	NATHANIEL	ADAM	NATHANIEL.ADAM@sakilacustomer.org	509	1	2006-02-14 22:04:37	2006-02-15 04:57:20
542	2	LONNIE	TIRADO	LONNIE.TIRADO@sakilacustomer.org	548	1	2006-02-14 22:04:37	2006-02-15 04:57:20
571	2	JOHNNIE	CHISHOLM	JOHNNIE.CHISHOLM@sakilacustomer.org	577	1	2006-02-14 22:04:37	2006-02-15 04:57:20

17- Find all customers whose first name have "r" in the second position .

```
%%sql

SELECT * FROM customer

WHERE first_name LIKE '_r%';
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
45 rows affected.
```

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
31	2	BRENDA	WRIGHT	BRENDA.WRIGHT@sakilacustomer.org	35	1	2006-02-14 22:04:36	2006-02-15 04:57:20
47	1	FRANCES	PARKER	FRANCES.PARKER@sakilacustomer.org	51	1	2006-02-14 22:04:36	2006-02-15 04:57:20
76	2	IRENE	PRICE	IRENE.PRICE@sakilacustomer.org	80	1	2006-02-14 22:04:36	2006-02-15 04:57:20
102	1	CRYSTAL	FORD	CRYSTAL.FORD@sakilacustomer.org	106	1	2006-02-14 22:04:36	2006-02-15 04:57:20
108	1	TRACY	COLE	TRACY.COLE@sakilacustomer.org	112	1	2006-02-14 22:04:36	2006-02-15 04:57:20
114	2	GRACE	ELLIS	GRACE.ELLIS@sakilacustomer.org	118	1	2006-02-14 22:04:36	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
160	2	ERIN	DUNN	ERIN.DUNN@sakilacustomer.org	164	1	2006-02-14 22:04:36	2006-02-15 04:57:20
169	2	ERICA	MATTHEWS	ERICA.MATTHEWS@sakilacustomer.org	173	0	2006-02-14 22:04:36	2006-02-15 04:57:20
187	2	BRITTANY	RILEY	BRITTANY.RILEY@sakilacustomer.org	191	1	2006-02-14 22:04:36	2006-02-15 04:57:20
194	2	KRISTEN	CHAVEZ	KRISTEN.CHAVEZ@sakilacustomer.org	198	1	2006-02-14 22:04:36	2006-02-15 04:57:20
214	1	KRISTIN	JOHNSTON	KRISTIN.JOHNSTON@sakilacustomer.org	218	1	2006-02-14 22:04:36	2006-02-15 04:57:20
225	1	ARLENE	HARVEY	ARLENE.HARVEY@sakilacustomer.org	229	1	2006-02-14 22:04:36	2006-02-15 04:57:20
255	2	IRMA	PEARSON	IRMA.PEARSON@sakilacustomer.org	260	1	2006-02-14 22:04:36	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
273	2	PRISCILLA	LOWE	PRISCILLA.LOWE@sakilacustomer.org	278	1	2006-02-14 22:04:37	2006-02-15 04:57:20
276	1	BRANDY	GRAVES	BRANDY.GRAVES@sakilacustomer.org	281	1	2006-02-14 22:04:37	2006-02-15 04:57:20
280	2	TRACEY	BARRETT	TRACEY.BARRETT@sakilacustomer.org	285	1	2006-02-14 22:04:37	2006-02-15 04:57:20
290	1	KRISTINA	CHAMBERS	KRISTINA.CHAMBERS@sakilacustomer.org	295	1	2006-02-14 22:04:37	2006-02-15 04:57:20
298	1	ERIKA	PENA	ERIKA.PENA@sakilacustomer.org	303	1	2006-02-14 22:04:37	2006-02-15 04:57:20
318	1	BRIAN	WYMAN	BRIAN.WYMAN@sakilacustomer.org	323	1	2006-02-14 22:04:37	2006-02-15 04:57:20
329	2	FRANK	WAGGONER	FRANK.WAGGONER@sakilacustomer.org	334	1	2006-02-14 22:04:37	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
331	1	ERIC	ROBERT	ERIC.ROBERT@sakilacustomer.org	336	1	2006-02-14 22:04:37	2006-02-15 04:57:20
335	1	GREGORY	MAULDIN	GREGORY.MAULDIN@sakilacustomer.org	340	1	2006-02-14 22:04:37	2006-02-15 04:57:20
346	1	ARTHUR	SIMPKINS	ARTHUR.SIMPKINS@sakilacustomer.org	351	1	2006-02-14 22:04:37	2006-02-15 04:57:20
365	2	BRUCE	SCHWARZ	BRUCE.SCHWARZ@sakilacustomer.org	370	1	2006-02-14 22:04:37	2006-02-15 04:57:20
366	1	BRANDON	HUEY	BRANDON.HUEY@sakilacustomer.org	371	1	2006-02-14 22:04:37	2006-02-15 04:57:20
369	2	FRED	WHEAT	FRED.WHEAT@sakilacustomer.org	374	1	2006-02-14 22:04:37	2006-02-15 04:57:20
384	2	ERNEST	STEPP	ERNEST.STEPP@sakilacustomer.org	389	1	2006-02-14 22:04:37	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
388	2	CRAIG	MORRELL	CRAIG.MORRELL@sakilacustomer.org	393	1	2006-02-14 22:04:37	2006-02-15 04:57:20
400	2	BRYAN	HARDISON	BRYAN.HARDISON@sakilacustomer.org	405	1	2006-02-14 22:04:37	2006-02-15 04:57:20
417	1	TRAVIS	ESTEP	TRAVIS.ESTEP@sakilacustomer.org	422	1	2006-02-14 22:04:37	2006-02-15 04:57:20
425	2	FRANCIS	SIKES	FRANCIS.SIKES@sakilacustomer.org	430	1	2006-02-14 22:04:37	2006-02-15 04:57:20
426	1	BRADLEY	MOTLEY	BRADLEY.MOTLEY@sakilacustomer.org	431	1	2006-02-14 22:04:37	2006-02-15 04:57:20
429	2	FREDERICK	ISBELL	FREDERICK.ISBELL@sakilacustomer.org	434	1	2006-02-14 22:04:37	2006-02-15 04:57:20
436	1	TROY	QUIGLEY	TROY.QUIGLEY@sakilacustomer.org	441	1	2006-02-14 22:04:37	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
443	2	FRANCISCO	SKIDMORE	FRANCISCO.SKIDMORE@sakilacustomer.org	448	1	2006-02-14 22:04:37	2006-02-15 04:57:20
472	1	GREG	ROBINS	GREG.ROBINS@sakilacustomer.org	477	1	2006-02-14 22:04:37	2006-02-15 04:57:20
493	1	BRENT	HARKINS	BRENT.HARKINS@sakilacustomer.org	498	1	2006-02-14 22:04:37	2006-02-15 04:57:20
502	1	BRETT	CORNWELL	BRETT.CORNWELL@sakilacustomer.org	507	1	2006-02-14 22:04:37	2006-02-15 04:57:20
514	2	FRANKLIN	TROUTMAN	FRANKLIN.TROUTMAN@sakilacustomer.org	520	1	2006-02-14 22:04:37	2006-02-15 04:57:20
517	2	BRAD	MCCURDY	BRAD.MCCURDY@sakilacustomer.org	523	1	2006-02-14 22:04:37	2006-02-15 04:57:20
522	2	ARNOLD	HAVENS	ARNOLD.HAVENS@sakilacustomer.org	528	1	2006-02-14 22:04:37	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
529	2	ERIK	GUILLEN	ERIK.GUILLEN@sakilacustomer.org	535	1	2006-02-14 22:04:37	2006-02-15 04:57:20
556	2	ARMANDO	GRUBER	ARMANDO.GRUBER@sakilacustomer.org	562	1	2006-02-14 22:04:37	2006-02-15 04:57:20
589	1	TRACY	HERRMANN	TRACY.HERRMANN@sakilacustomer.org	595	1	2006-02-14 22:04:37	2006-02-15 04:57:20
597	1	FREDDIE	DUGGAN	FREDDIE.DUGGAN@sakilacustomer.org	603	1	2006-02-14 22:04:37	2006-02-15 04:57:20

18 - Find all customers whose first name starts with "a" and are at least 5 characters in length.

```
%%sql

SELECT * FROM customer

WHERE first_name LIKE 'a%' AND LENGTH(first_name) >= 5;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies

34 rows affected.
```

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
29	2	ANGELA	HERNANDEZ	ANGELA.HERNANDEZ@sakilacustomer.org	33	1	2006-02-14 22:04:36	2006-02-15 04:57:20
40	2	AMANDA	CARTER	AMANDA.CARTER@sakilacustomer.org	44	1	2006-02-14 22:04:36	2006-02-15 04:57:20
51	1	ALICE	STEWART	ALICE.STEWART@sakilacustomer.org	55	1	2006-02-14 22:04:36	2006-02-15 04:57:20
63	1	ASHLEY	RICHARDSON	ASHLEY.RICHARDSON@sakilacustomer.org	67	1	2006-02-14 22:04:36	2006-02-15 04:57:20
81	1	ANDREA	HENDERSON	ANDREA.HENDERSON@sakilacustomer.org	85	1	2006-02-14 22:04:36	2006-02-15 04:57:20
97	2	ANNIE	RUSSELL	ANNIE.RUSSELL@sakilacustomer.org	101	1	2006-02-14 22:04:36	2006-02-15 04:57:20
136	2	ANITA	MORALES	ANITA.MORALES@sakilacustomer.org	140	1	2006-02-14 22:04:36	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
139	1	AMBER	DIXON	AMBER.DIXON@sakilacustomer.org	143	1	2006-02-14 22:04:36	2006-02-15 04:57:20
142	1	APRIL	BURNS	APRIL.BURNS@sakilacustomer.org	146	1	2006-02-14 22:04:36	2006-02-15 04:57:20
152	1	ALICIA	MILLS	ALICIA.MILLS@sakilacustomer.org	156	1	2006-02-14 22:04:36	2006-02-15 04:57:20
173	1	AUDREY	RAY	AUDREY.RAY@sakilacustomer.org	177	1	2006-02-14 22:04:36	2006-02-15 04:57:20
175	1	ANNETTE	OLSON	ANNETTE.OLSON@sakilacustomer.org	179	1	2006-02-14 22:04:36	2006-02-15 04:57:20
217	2	AGNES	BISHOP	AGNES.BISHOP@sakilacustomer.org	221	1	2006-02-14 22:04:36	2006-02-15 04:57:20
225	1	ARLENE	HARVEY	ARLENE.HARVEY@sakilacustomer.org	229	1	2006-02-14 22:04:36	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
228	2	ALLISON	STANLEY	ALLISON.STANLEY@sakilacustomer.org	232	1	2006-02-14 22:04:36	2006-02-15 04:57:20
320	2	ANTHONY	SCHWAB	ANTHONY.SCHWAB@sakilacustomer.org	325	1	2006-02-14 22:04:37	2006-02-15 04:57:20
333	2	ANDREW	PURDY	ANDREW.PURDY@sakilacustomer.org	338	1	2006-02-14 22:04:37	2006-02-15 04:57:20
346	1	ARTHUR	SIMPKINS	ARTHUR.SIMPKINS@sakilacustomer.org	351	1	2006-02-14 22:04:37	2006-02-15 04:57:20
352	1	ALBERT	CROUSE	ALBERT.CROUSE@sakilacustomer.org	357	1	2006-02-14 22:04:37	2006-02-15 04:57:20
375	2	AARON	SELBY	AARON.SELBY@sakilacustomer.org	380	1	2006-02-14 22:04:37	2006-02-15 04:57:20
398	1	ANTONIO	MEEK	ANTONIO.MEEK@sakilacustomer.org	403	1	2006-02-14 22:04:37	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
412	2	ALLEN	BUTTERFIELD	ALLEN.BUTTERFIELD@sakilacustomer.org	417	1	2006-02-14 22:04:37	2006-02-15 04:57:20
423	2	ALFRED	CASILLAS	ALFRED.CASILLAS@sakilacustomer.org	428	1	2006-02-14 22:04:37	2006-02-15 04:57:20
439	2	ALEXANDER	FENNELL	ALEXANDER.FENNELL@sakilacustomer.org	444	1	2006-02-14 22:04:37	2006-02-15 04:57:20
467	2	ALVIN	DELOACH	ALVIN.DELOACH@sakilacustomer.org	472	1	2006-02-14 22:04:37	2006-02-15 04:57:20
503	1	ANGEL	BARCLAY	ANGEL.BARCLAY@sakilacustomer.org	508	1	2006-02-14 22:04:37	2006-02-15 04:57:20
515	1	ANDRE	RAPP	ANDRE.RAPP@sakilacustomer.org	521	1	2006-02-14 22:04:37	2006-02-15 04:57:20
522	2	ARNOLD	HAVENS	ARNOLD.HAVENS@sakilacustomer.org	528	1	2006-02-14 22:04:37	2006-02-15 04:57:20

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
525	2	ADRIAN	CLARY	ADRIAN.CLARY@sakilacustomer.org	531	1	2006-02-14 22:04:37	2006-02-15 04:57:20
548	1	ALLAN	CORNISH	ALLAN.CORNISH@sakilacustomer.org	554	1	2006-02-14 22:04:37	2006-02-15 04:57:20
556	2	ARMANDO	GRUBER	ARMANDO.GRUBER@sakilacustomer.org	562	1	2006-02-14 22:04:37	2006-02-15 04:57:20
567	2	ALFREDO	MCADAMS	ALFREDO.MCADAMS@sakilacustomer.org	573	1	2006-02-14 22:04:37	2006-02-15 04:57:20
568	2	ALBERTO	HENNING	ALBERTO.HENNING@sakilacustomer.org	574	1	2006-02-14 22:04:37	2006-02-15 04:57:20
599	2	AUSTIN	CINTRON	AUSTIN.CINTRON@sakilacustomer.org	605	1	2006-02-14 22:04:37	2006-02-15 04:57:20

19- Find all customers whose first name starts with "a" and ends with "o".

```
%%sql

SELECT * FROM customer

WHERE first_name LIKE 'a%o';
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
4 rows affected.
```

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
398	1	ANTONIO	MEEK	ANTONIO.MEEK@sakilacustomer.org	403	1	2006-02-14 22:04:37	2006-02-15 04:57:20
556	2	ARMANDO	GRUBER	ARMANDO.GRUBER@sakilacustomer.org	562	1	2006-02-14 22:04:37	2006-02-15 04:57:20
567	2	ALFREDO	MCADAMS	ALFREDO.MCADAMS@sakilacustomer.org	573	1	2006-02-14 22:04:37	2006-02-15 04:57:20
568	2	ALBERTO	HENNING	ALBERTO.HENNING@sakilacustomer.org	574	1	2006-02-14 22:04:37	2006-02-15 04:57:20

20 - Get the films with pg and pg-13 rating using IN operator.

```
%%sql
SELECT * FROM film
WHERE rating IN ('PG', 'PG-13') LIMIT 5; # use LIMIT to avoid large output
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
5 rows affected.
```

film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features	last_update
1	ACADEMY DINOSAUR	A Epic Drama of a Feminist And a Mad Scientist who must Battle a Teacher in The Canadian Rockies	2006	1	None	6	0.99	86	20.99	PG	{'Behind the Scenes', 'Deleted Scenes'}	2006-02-15 05:03:42
6	AGENT TRUMAN	A Intrepid Panorama of a Robot And	2006	1	None	3	2.99	169	17.99	PG	{'Deleted Scenes'}	2006-02-15 05:03:42

film_id	**title**	**description**	**release_year**	**language_id**	**original_language_id**	**rental_duration**	**rental_rate**	**length**	**replacement_cost**	**rating**	**special_features**	**last_update**
		a Boy who must Escape a Sumo Wrestler in Ancient China										
7	AIRPLANE SIERRA	A Touching Saga of a Hunter And a Butler who must Discover a Butler in	2006	1	None	6	4.99	62	28.99	PG-13	{'Trailers', 'Deleted Scenes'}	2006-02-15 05:03:42

film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features	last_update
		A Jet Boat										
9	ALABAMA DEVIL	A Thoughtful Panorama of a Database Administrator And a Mad Scientist who must Outgun a Mad Scientist in A Jet Boat	2006	1	None	3	2.99	114	21.99	PG-13	{'Trailers', 'Deleted Scenes'}	2006-02-15 05:03:42

film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features	last_update
12	ALASKA PHANTOM	A Fanciful Saga of a Hunter And a Pastry Chef who must Vanquish a Boy in Australia	2006	1	None	6	0.99	136	22.99	PG	{'Commentaries', 'Deleted Scenes'}	2006-02-15 05:03:42

21 - Get the films with length between 50 to 100 using between operator.

```
%%sql

SELECT * FROM film

WHERE length BETWEEN 50 AND 100 limit 5;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies

5 rows affected.
```

film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features	last_update
1	ACADEMY DINOSAUR	A Epic Drama of a Feminist And a Mad Scientist who must Battle a Teacher in The Canadian Rockies	2006	1	None	6	0.99	86	20.99	PG	{'Behind the Scenes', 'Deleted Scenes'}	2006-02-15 05:03:42
3	ADAPTATION	A Astoundin	2006	1	None	7	2.99	50	18.99	NC-	{'Trailers', 'Deleted	2006-02-15

film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features	last_update
	HOLES	g Reflection of a Lumberjack And a Car who must Sink a Lumberjack in A Baloon Factory								17	Scenes'}	05:03:42
7	AIRPLANE SIERRA	A Touching Saga of a	2006	1	None	6	4.99	62	28.99	PG-13	{'Trailers', 'Deleted Scenes'}	2006-02-15 05:03:42

film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features	last_update
		Hunter And a Butler who must Discover a Butler in A Jet Boat										
8	AIRPORT POLLOCK	A Epic Tale of a Moose And a Girl who must Confron	2006	1	None	6	4.99	54	15.99	R	{'Trailers'}	2006-02-15 05:03:42

film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features	last_update
		t a Monkey in Ancient India										
10	ALADDIN CALENDAR	A Action-Packed Tale of a Man And a Lumberjack who must Reach a Feminist in Anc	2006	1	None	6	4.99	63	24.99	NC-17	{'Trailers', 'Deleted Scenes'}	2006-02-15 05:03:42

film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features	last_update
		ient China										

22 - Get the top 50 actors using limit operator.

```
%%sql
SELECT * FROM actor
LIMIT 50;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
50 rows affected.
```

actor_id	first_name	last_name	last_update
1	PENELOPE	GUINESS	2006-02-15 04:34:33
2	NICK	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
6	BETTE	NICHOLSON	2006-02-15 04:34:33
7	GRACE	MOSTEL	2006-02-15 04:34:33
8	MATTHEW	JOHANSSON	2006-02-15 04:34:33
9	JOE	SWANK	2006-02-15 04:34:33

actor_id	first_name	last_name	last_update
10	CHRISTIAN	GABLE	2006-02-15 04:34:33
11	ZERO	CAGE	2006-02-15 04:34:33
12	KARL	BERRY	2006-02-15 04:34:33
13	UMA	WOOD	2006-02-15 04:34:33
14	VIVIEN	BERGEN	2006-02-15 04:34:33
15	CUBA	OLIVIER	2006-02-15 04:34:33
16	FRED	COSTNER	2006-02-15 04:34:33
17	HELEN	VOIGHT	2006-02-15 04:34:33
18	DAN	TORN	2006-02-15 04:34:33
19	BOB	FAWCETT	2006-02-15 04:34:33
20	LUCILLE	TRACY	2006-02-15 04:34:33
21	KIRSTEN	PALTROW	2006-02-15 04:34:33
22	ELVIS	MARX	2006-02-15 04:34:33
23	SANDRA	KILMER	2006-02-15 04:34:33
24	CAMERON	STREEP	2006-02-15 04:34:33
25	KEVIN	BLOOM	2006-02-15 04:34:33
26	RIP	CRAWFORD	2006-02-15 04:34:33

actor_id	**first_name**	**last_name**	**last_update**
27	JULIA	MCQUEEN	2006-02-15 04:34:33
28	WOODY	HOFFMAN	2006-02-15 04:34:33
29	ALEC	WAYNE	2006-02-15 04:34:33
30	SANDRA	PECK	2006-02-15 04:34:33
31	SISSY	SOBIESKI	2006-02-15 04:34:33
32	TIM	HACKMAN	2006-02-15 04:34:33
33	MILLA	PECK	2006-02-15 04:34:33
34	AUDREY	OLIVIER	2006-02-15 04:34:33
35	JUDY	DEAN	2006-02-15 04:34:33
36	BURT	DUKAKIS	2006-02-15 04:34:33
37	VAL	BOLGER	2006-02-15 04:34:33
38	TOM	MCKELLEN	2006-02-15 04:34:33
39	GOLDIE	BRODY	2006-02-15 04:34:33
40	JOHNNY	CAGE	2006-02-15 04:34:33
41	JODIE	DEGENERES	2006-02-15 04:34:33
42	TOM	MIRANDA	2006-02-15 04:34:33
43	KIRK	JOVOVICH	2006-02-15 04:34:33

actor_id	first_name	last_name	last_update
44	NICK	STALLONE	2006-02-15 04:34:33
45	REESE	KILMER	2006-02-15 04:34:33
46	PARKER	GOLDBERG	2006-02-15 04:34:33
47	JULIA	BARRYMORE	2006-02-15 04:34:33
48	FRANCES	DAY-LEWIS	2006-02-15 04:34:33
49	ANNE	CRONYN	2006-02-15 04:34:33
50	NATALIE	HOPKINS	2006-02-15 04:34:33

23 - Get the distinct film ids from inventory table

```
%%sql

SELECT count(DISTINCT film_id) FROM inventory;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies

1 rows affected.
```

count(DISTINCT film_id)
958

Functions

Basic Aggregate Functions:

Question 1:Retrieve the total number of rentals made in the Sakila database.Hint: Use the COUNT() function

```
%%sql

SELECT COUNT(*) AS total_rentals

FROM rental;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
```

```
1 rows affected.
```

total_rentals
16044

Question 2:Find the average rental duration (in days) of movies rented from the Sakila database.Hint: Utilize the AVG() function.

```
%%sql
SELECT AVG(rental_duration) AS average_rental_duration
FROM film;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
1 rows affected.
```

average_rental_duration
4.9850

String Functions:

Question 3:Display the first name and last name of customers in uppercase.Hint: Use the UPPER () function.

```
%%sql
SELECT
   UPPER(first_name) AS first_name_upper,
   UPPER(last_name) AS last_name_upper
FROM customer limit 20;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
20 rows affected.
```

first_name_upper	last_name_upper
MARY	SMITH
PATRICIA	JOHNSON

first_name_upper	last_name_upper
LINDA	WILLIAMS
BARBARA	JONES
ELIZABETH	BROWN
JENNIFER	DAVIS
MARIA	MILLER
SUSAN	WILSON
MARGARET	MOORE
DOROTHY	TAYLOR
LISA	ANDERSON
NANCY	THOMAS
KAREN	JACKSON
BETTY	WHITE
HELEN	HARRIS
SANDRA	MARTIN
DONNA	THOMPSON
CAROL	GARCIA
RUTH	MARTINEZ

first_name_upper	last_name_upper
SHARON	ROBINSON

Question 4:Extract the month from the rental date and display it alongside the rental ID.Hint: Employ the MONTH() function

```
%%sql

SELECT

  rental_id,

  MONTH(rental_date) AS rental_month

FROM

  rental limit 10; # use limit to avoid large output
```

 * mysql+mysqlconnector://root:***@localhost/mavenmovies

10 rows affected.

rental_id	rental_month
1	5
2	5
3	5
4	5
5	5
6	5
7	5
8	5
9	5

rental_id	rental_month
10	5

GROUP BY:

Question 5:Retrieve the count of rentals for each customer (display customer ID and the count of rentals).Hint: Use COUNT () in conjunction with GROUP BY

```
%%sql
SELECT
    customer_id,
    COUNT(*) AS rental_count
FROM
    rental
GROUP BY
    customer_id limit 10; #use limit to avoid large output
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
10 rows affected.
```

customer_id	rental_count
1	32
2	27
3	26
4	22
5	38
6	28
7	33

customer_id	rental_count
8	24
9	23
10	25

.Question 6:Find the total revenue generated by each store.Hint: Combine SUM() and GROUP BY

```
%%sql

SELECT

   c.store_id,

   SUM(p.amount) AS total_revenue

FROM

   payment p

JOIN

   customer c ON p.customer_id = c.customer_id

GROUP BY

   c.store_id;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies

2 rows affected.
```

store_id	total_revenue
1	37001.52
2	30414.99

Question 7:Determine the total number of rentals for each category of movies.Hint: JOIN film_category, film, and rental tables, then use cOUNT () and GROUP BY

```
%%sql

SELECT
```

```
    c.store_id,
    SUM(p.amount) AS total_revenue
FROM
    payment p
JOIN
    customer c ON p.customer_id = c.customer_id
GROUP BY
    c.store_id;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
2 rows affected.

 store_id   total_revenue

 1          37001.52

 2          30414.99
```

Question 8:Find the average rental rate of movies in each language.Hint: JOIN film and language tables, then use AVG () and GROUP BY

```
%%sql
SELECT
    l.name AS language,
    AVG(f.rental_rate) AS average_rental_rate
FROM
    film f
JOIN
    language l ON f.language_id = l.language_id
GROUP BY
    l.name
ORDER BY
    average_rental_rate ;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
1 rows affected.
```

language	average_rental_rate
English	2.980000

Joins

Questions 9 -Display the title of the movie, customer s first name, and last name who rented it.Hint: Use JOIN between the film, inventory, rental, and customer tables.

```
%%sql
SELECT
   f.title,
   c.first_name,
   c.last_name
FROM
   rental r
JOIN
   inventory i ON r.inventory_id = i.inventory_id
JOIN
   film f ON i.film_id = f.film_id
JOIN
   customer c ON r.customer_id = c.customer_id limit 10;  #use limit to avoid large output
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
10 rows affected.
```

title	first_name	last_name
ACADEMY DINOSAUR	JOEL	FRANCISCO
ACADEMY DINOSAUR	GABRIEL	HARDER

title	first_name	last_name
ACADEMY DINOSAUR	DIANNE	SHELTON
ACADEMY DINOSAUR	NORMAN	CURRIER
ACADEMY DINOSAUR	BEATRICE	ARNOLD
ACADEMY DINOSAUR	GERALDINE	PERKINS
ACADEMY DINOSAUR	VIRGIL	WOFFORD
ACADEMY DINOSAUR	WILLIE	MARKHAM
ACADEMY DINOSAUR	DEBRA	NELSON
ACADEMY DINOSAUR	DARREN	WINDHAM

.Question 10:Retrieve the names of all actors who have appeared in the film "Gone with the Wind."Hint: Use JOIN between the film actor, film, and actor tables.

```
%%sql

SELECT

   a.first_name,

   a.last_name

FROM

   film f

JOIN

   film_actor fa ON f.film_id = fa.film_id

JOIN

   actor a ON fa.actor_id = a.actor_id

WHERE

   f.title = 'Gone with the Wind';
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
```

```
0 rows affected.

 first_name   last_name
```

Question 11:Retrieve the customer names along with the total amount they've spent on rentals.Hint: JOIN customer, payment, and rental tables, then use SUM() and GROUP BY.

```
%%sql

SELECT

   c.customer_id,

   c.first_name,

   c.last_name,

   SUM(p.amount) AS total_amount_spent

FROM

   customer c

JOIN

   payment p ON c.customer_id = p.customer_id

GROUP BY

   c.customer_id, c.first_name, c.last_name

ORDER BY

   total_amount_spent DESC limit 10;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies

10 rows affected.

 customer_id   first_name   last_name   total_amount_spent

 526           KARL         SEAL        221.55

 148           ELEANOR      HUNT        216.54

 144           CLARA        SHAW        195.58

 137           RHONDA       KENNEDY     194.61
```

customer_id	first_name	last_name	total_amount_spent
178	MARION	SNYDER	194.61
459	TOMMY	COLLAZO	186.62
469	WESLEY	BULL	177.60
468	TIM	CARY	175.61
236	MARCIA	DEAN	175.58
181	ANA	BRADLEY	174.66

Question 12:List the titles of movies rented by each customer in a particular city (e.g., 'London').Hint: JOIN customer, address, city, rental, inventory, and film tables, then use GROUP BY.

```
%%sql
SELECT
   c.first_name,
   c.last_name,
   ci.city,
   f.title
FROM
   customer c
JOIN
   address a ON c.address_id = a.address_id
JOIN
   city ci ON a.city_id = ci.city_id
JOIN
   rental r ON c.customer_id = r.customer_id
JOIN
```

```
   inventory i ON r.inventory_id = i.inventory_id
JOIN
   film f ON i.film_id = f.film_id
WHERE
   ci.city = 'London'
ORDER BY
   c.last_name, f.title;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
48 rows affected.

first_name  last_name  city    title

MATTIE      HOFFMAN    London  ACADEMY DINOSAUR

MATTIE      HOFFMAN    London  CHICKEN HELLFIGHTERS

MATTIE      HOFFMAN    London  COLDBLOODED DARLING

MATTIE      HOFFMAN    London  CONQUERER NUTS

MATTIE      HOFFMAN    London  DARKNESS WAR

MATTIE      HOFFMAN    London  DOOM DANCING

MATTIE      HOFFMAN    London  DRIFTER COMMANDMENTS

MATTIE      HOFFMAN    London  EMPIRE MALKOVICH

MATTIE      HOFFMAN    London  FACTORY DRAGON

MATTIE      HOFFMAN    London  FLATLINERS KILLER

MATTIE      HOFFMAN    London  FROGMEN BREAKING
```

first_name	last_name	city	title
MATTIE	HOFFMAN	London	JAWBREAKER BROOKLYN
MATTIE	HOFFMAN	London	JUMPING WRATH
MATTIE	HOFFMAN	London	PITY BOUND
MATTIE	HOFFMAN	London	PRINCESS GIANT
MATTIE	HOFFMAN	London	TIGHTS DAWN
MATTIE	HOFFMAN	London	TRACY CIDER
MATTIE	HOFFMAN	London	VELVET TERMINATOR
MATTIE	HOFFMAN	London	VIDEOTAPE ARSENIC
MATTIE	HOFFMAN	London	WATERSHIP FRONTIER
MATTIE	HOFFMAN	London	WOMEN DORADO
MATTIE	HOFFMAN	London	WRATH MILE
CECIL	VINES	London	AMADEUS HOLY
CECIL	VINES	London	ARABIA DOGMA
CECIL	VINES	London	BACKLASH UNDEFEATED
CECIL	VINES	London	BLOOD ARGONAUTS
CECIL	VINES	London	CAT CONEHEADS
CECIL	VINES	London	CAT CONEHEADS

first_name	last_name	city	title
CECIL	VINES	London	CHARIOTS CONSPIRACY
CECIL	VINES	London	COMA HEAD
CECIL	VINES	London	DAISY MENAGERIE
CECIL	VINES	London	DARKNESS WAR
CECIL	VINES	London	DISTURBING SCARFACE
CECIL	VINES	London	EDGE KISSING
CECIL	VINES	London	GRIT CLOCKWORK
CECIL	VINES	London	INVASION CYCLONE
CECIL	VINES	London	JEEPERS WEDDING
CECIL	VINES	London	KANE EXORCIST
CECIL	VINES	London	LABYRINTH LEAGUE
CECIL	VINES	London	MOONSHINE CABIN
CECIL	VINES	London	NONE SPIKING
CECIL	VINES	London	PURPLE MOVIE
CECIL	VINES	London	RAGE GAMES
CECIL	VINES	London	RANGE MOONWALKER
CECIL	VINES	London	SHOOTIST SUPERFLY

first_name	last_name	city	title
CECIL	VINES	London	TELEMARK HEARTBREAKERS
CECIL	VINES	London	TOMATOES HELLFIGHTERS
CECIL	VINES	London	TRIP NEWTON

Advanced Joins and GROUP BY:

Question 13:Display the top 5 rented movies along with the number of times they've been rented.Hint: JOIN film, inventory, and rental tables, then use COUNT () and GROUP BY, and limit the results

```
%%sql

SELECT

   f.title AS movie_title,

   COUNT(r.rental_id) AS times_rented

FROM

   film f

JOIN

   inventory i ON f.film_id = i.film_id

JOIN

   rental r ON i.inventory_id = r.inventory_id

GROUP BY

   f.title

ORDER BY

   times_rented DESC

LIMIT 5;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies

5 rows affected.
```

movie_title	times_rented
BUCKET BROTHERHOOD	34
ROCKETEER MOTHER	33
FORWARD TEMPLE	32
GRIT CLOCKWORK	32
JUGGLER HARDLY	32

Question 14:Determine the customers who have rented movies from both stores (store ID 1 and store ID 2).Hint: Use JOINS with rental, inventory, and customer tables and consider COUNT() and GROUP BY.

```
%%sql
SELECT
    r.customer_id,
    c.first_name,
    c.last_name,
    COUNT(DISTINCT i.store_id) AS store_count
FROM
    rental r
JOIN
    inventory i ON r.inventory_id = i.inventory_id
JOIN
    customer c ON r.customer_id = c.customer_id
GROUP BY
    r.customer_id, c.first_name, c.last_name
HAVING
    COUNT(DISTINCT i.store_id) = 2 LIMIT 10; #use limit to avoid large output
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
```

10 rows affected.

customer_id	first_name	last_name	store_count
1	MARY	SMITH	2
2	PATRICIA	JOHNSON	2
3	LINDA	WILLIAMS	2
4	BARBARA	JONES	2
5	ELIZABETH	BROWN	2
6	JENNIFER	DAVIS	2
7	MARIA	MILLER	2
8	SUSAN	WILSON	2
9	MARGARET	MOORE	2
10	DOROTHY	TAYLOR	2

Windows Function:

1. Rank the customers based on the total amount they've spent on rentals.

```
%%sql

SELECT

   customer_id,

   CONCAT(first_name, ' ', last_name) AS customer_name,

   SUM(amount) AS total_spent,

   RANK() OVER (ORDER BY SUM(amount) DESC) AS spending_rank

FROM

   payment

JOIN
```

```
    customer USING (customer_id)
GROUP BY
    customer_id, first_name, last_name LIMIT 10; #use limit to avoid large output
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
10 rows affected.

 customer_id   customer_name      total_spent   spending_rank

 526           KARL SEAL          221.55        1

 148           ELEANOR HUNT       216.54        2

 144           CLARA SHAW         195.58        3

 137           RHONDA KENNEDY     194.61        4

 178           MARION SNYDER      194.61        4

 459           TOMMY COLLAZO      186.62        6

 469           WESLEY BULL        177.60        7

 468           TIM CARY           175.61        8

 236           MARCIA DEAN        175.58        9

 181           ANA BRADLEY        174.66        10
```

2. Calculate the cumulative revenue generated by each film over time.

```
%%sql
SELECT
    f.film_id,
    f.title,
    r.rental_date,
   SUM(p.amount) OVER (PARTITION BY f.film_id ORDER BY r.rental_date) AS
cumulative_revenue
FROM
   payment p
JOIN rental r ON p.rental_id = r.rental_id
JOIN inventory i ON r.inventory_id = i.inventory_id
JOIN film f ON i.film_id = f.film_id LIMIT 10; #use limit to avoid large output
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
10 rows affected.

 film_id  title              rental_date          cumulative_revenue

 1        ACADEMY DINOSAUR   2005-05-27 07:03:28  0.99
 1        ACADEMY DINOSAUR   2005-05-30 20:21:07  2.98
 1        ACADEMY DINOSAUR   2005-06-15 02:57:51  3.97
 1        ACADEMY DINOSAUR   2005-06-17 20:24:00  4.96
 1        ACADEMY DINOSAUR   2005-06-21 00:30:26  6.95
 1        ACADEMY DINOSAUR   2005-07-07 10:41:31  7.94
 1        ACADEMY DINOSAUR   2005-07-07 20:59:06  8.93
 1        ACADEMY DINOSAUR   2005-07-08 19:03:15  9.92
 1        ACADEMY DINOSAUR   2005-07-10 13:07:31  10.91
 1        ACADEMY DINOSAUR   2005-07-27 07:51:11  11.90
```

3. Determine the average rental duration for each film, considering films with similar lengths.

```
%%sql
```

```
SELECT
   FLOOR(length / 10) * 10 AS length_group,
   AVG(rental_duration) AS avg_rental_duration
FROM
   film
GROUP BY
   length_group
ORDER BY
   length_group;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
15 rows affected.

 length_group   avg_rental_duration

 40             5.1071

 50             4.8088

 60             5.0571

 70             4.6753

 80             4.9091

 90             4.9138

 100            5.0000

 110            4.9221

 120            5.1081

 130            5.0541
```

length_group	avg_rental_duration
140	4.8235
150	5.2059
160	5.0517
170	4.9103
180	5.5217

4. Identify the top 3 films in each category based on their rental counts.

```
%%sql
SELECT
    category_name, title, rental_count FROM (
    SELECT
        c.name AS category_name,
        f.title,
        COUNT(r.rental_id) AS rental_count,
        RANK() OVER (PARTITION BY c.name ORDER BY COUNT(r.rental_id) DESC) AS rnk
    FROM
        rental r
    JOIN inventory i ON r.inventory_id = i.inventory_id
    JOIN film f ON i.film_id = f.film_id
    JOIN film_category fc ON f.film_id = fc.film_id
    JOIN category c ON fc.category_id = c.category_id
    GROUP BY
        c.name, f.title
) AS ranked
WHERE rnk <= 3 ;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
56 rows affected.
```

category_name	title	rental_count
Action	RUGRATS SHAKESPEARE	30
Action	SUSPECTS QUILLS	30
Action	HANDICAP BOONDOCK	28
Action	STORY SIDE	28
Action	TRIP NEWTON	28
Animation	JUGGLER HARDLY	32
Animation	DOGMA FAMILY	30
Animation	STORM HAPPINESS	29
Children	ROBBERS JOON	31
Children	IDOLS SNATCHERS	30
Children	SWEETHEARTS SUSPECTS	29
Classics	TIMBERLAND SKY	31
Classics	FROST HEAD	30
Classics	GILMORE BOILED	28
Classics	VOYAGE LEGALLY	28
Comedy	ZORRO ARK	31

category_name	title	rental_count
Comedy	CAT CONEHEADS	30
Comedy	CLOSER BANG	28
Documentary	WIFE TURN	31
Documentary	VIRGINIAN PLUTO	29
Documentary	EXPENDABLE STALLION	28
Drama	HOBBIT ALIEN	31
Drama	WITCHES PANIC	30
Drama	HARRY IDAHO	30
Family	NETWORK PEAK	31
Family	RUSH GOODFELLAS	31
Family	APACHE DIVINE	31
Foreign	ROCKETEER MOTHER	33
Foreign	SHOCK CABIN	30
Foreign	MOON BUNCH	29
Games	GRIT CLOCKWORK	32
Games	FORWARD TEMPLE	32
Games	MASSACRE USUAL	30

category_name	title	rental_count
Horror	PULP BEVERLY	30
Horror	FAMILY SWEET	29
Horror	SWARM GOLD	27
Music	SCALAWAG DUCK	32
Music	GREATEST NORTH	29
Music	CONFIDENTIAL INTERVIEW	29
Music	BOOGIE AMELIE	29
Music	DEER VIRGINIAN	29
Music	ENEMY ODDS	29
New	RIDGEMONT SUBMARINE	32
New	BUTTERFLY CHOCOLAT	30
New	FATAL HAUNTED	28
Sci-Fi	GOODFELLAS SALUTE	31
Sci-Fi	GRAFFITI LOVE	30
Sci-Fi	ENGLISH BULWORTH	30
Sci-Fi	MARRIED GO	30
Sports	TALENTED HOMICIDE	29

category_name	title	rental_count
Sports	GLEAMING JAWBREAKER	29
Sports	ROSES TREASURE	28
Sports	SATURDAY LAMBS	28
Travel	BUCKET BROTHERHOOD	34
Travel	MUSCLE BRIGHT	30
Travel	HORROR REIGN	27

5. Calculate the difference in rental counts between each customer's total rentals and the average rentalsacross all customers.

```
%%sql
WITH customer_rentals AS (
   SELECT customer_id, COUNT(*) AS rental_count
   FROM rental
   GROUP BY customer_id
), avg_rentals AS (
   SELECT AVG(rental_count) AS avg_count FROM customer_rentals
)
SELECT
   cr.customer_id,
   cr.rental_count,
   ar.avg_count,
   cr.rental_count - ar.avg_count AS difference_from_avg
FROM
   customer_rentals cr, avg_rentals ar limit 10; #use limit to avoid large output
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
```

```
10 rows affected.

customer_id   rental_count   avg_count   difference_from_avg

1             32             26.7846     5.2154

2             27             26.7846     0.2154

3             26             26.7846     -0.7846

4             22             26.7846     -4.7846

5             38             26.7846     11.2154

6             28             26.7846     1.2154

7             33             26.7846     6.2154

8             24             26.7846     -2.7846

9             23             26.7846     -3.7846

10            25             26.7846     -1.7846
```

6. Find the monthly revenue trend for the entire rental store over time.

```
%%sql

SELECT

   DATE_FORMAT(payment_date, '%Y-%m') AS month,

   SUM(amount) AS monthly_revenue

FROM

   payment

GROUP BY

   month

ORDER BY

   month;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
5 rows affected.
  month     monthly_revenue

  2005-05   4824.43

  2005-06   9631.88

  2005-07   28373.89

  2005-08   24072.13

  2006-02   514.18
```

7. Identify the customers whose total spending on rentals falls within the top 20% of all customers.

```
%%sql
WITH total_spending AS (
   SELECT
      customer_id,
      SUM(amount) AS total_amount
   FROM payment
   GROUP BY customer_id
), spending_ranked AS (
   SELECT *,
        NTILE(5) OVER (ORDER BY total_amount DESC) AS quintile
   FROM total_spending
)
SELECT *
FROM spending_ranked
WHERE quintile = 1 limit 10; #use limit to avoid large output
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
10 rows affected.

 customer_id   total_amount   quintile

 526           221.55         1

 148           216.54         1

 144           195.58         1

 137           194.61         1

 178           194.61         1

 459           186.62         1

 469           177.60         1

 468           175.61         1

 236           175.58         1

 181           174.66         1
```

8. Calculate the running total of rentals per category, ordered by rental count.

```
%%sql

SELECT

   c.name AS category,

   f.title,

   COUNT(r.rental_id) AS rental_count,

   SUM(COUNT(r.rental_id)) OVER (PARTITION BY c.name ORDER BY COUNT(r.rental_id)
DESC) AS running_total

FROM

   rental r
```

```
JOIN inventory i ON r.inventory_id = i.inventory_id

JOIN film f ON i.film_id = f.film_id

JOIN film_category fc ON f.film_id = fc.film_id

JOIN category c ON fc.category_id = c.category_id

GROUP BY c.name, f.title limit 10;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
10 rows affected.

category  title                  rental_count  running_total

Action    RUGRATS SHAKESPEARE    30            60

Action    SUSPECTS QUILLS        30            60

Action    HANDICAP BOONDOCK      28            144

Action    STORY SIDE             28            144

Action    TRIP NEWTON            28            144

Action    PRIMARY GLASS          27            171

Action    STAGECOACH ARMAGEDDON  26            223

Action    FANTASY TROOPERS       26            223

Action    CLUELESS BUCKET        25            273

Action    HILLS NEIGHBORS        25            273
```

9. Find the films that have been rented less than the average rental count for their respective categories.

```
%%sql

WITH film_rentals AS (

   SELECT
```

```
        fc.category_id,
        f.film_id,
        f.title,
        COUNT(r.rental_id) AS rental_count
    FROM rental r
    JOIN inventory i ON r.inventory_id = i.inventory_id
    JOIN film f ON i.film_id = f.film_id
    JOIN film_category fc ON f.film_id = fc.film_id
    GROUP BY fc.category_id, f.film_id, f.title
), category_avg AS (
    SELECT category_id, AVG(rental_count) AS avg_count
    FROM film_rentals
    GROUP BY category_id
)
SELECT
    fr.title,
    fr.rental_count,
    ca.avg_count,
    c.name AS category
FROM film_rentals fr
JOIN category_avg ca ON fr.category_id = ca.category_id
JOIN category c ON c.category_id = fr.category_id
WHERE fr.rental_count < ca.avg_count limit 20;
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
20 rows affected.
```

title	rental_count	avg_count	category
GOSFORD DONNIE	8	18.2295	Action
PARK CITIZEN	8	18.2295	Action
CASUALTIES ENCINO	9	18.2295	Action
MONTEZUMA COMMAND	9	18.2295	Action
ANTITRUST TOMATOES	10	18.2295	Action
GRAIL FRANKENSTEIN	10	18.2295	Action
DARKO DORADO	11	18.2295	Action
DRAGON SQUAD	11	18.2295	Action
LORD ARIZONA	11	18.2295	Action
MAGNOLIA FORRESTER	11	18.2295	Action
CROW GREASE	12	18.2295	Action
SIDE ARK	12	18.2295	Action
SPEAKEASY DATE	12	18.2295	Action
LAWRENCE LOVE	13	18.2295	Action
DANCES NONE	14	18.2295	Action
GLASS DYING	14	18.2295	Action
SHRUNK DIVINE	14	18.2295	Action

title	rental_count	avg_count	category
WEREWOLF LOLA	14	18.2295	Action
DEVIL DESIRE	15	18.2295	Action
ENTRAPMENT SATISFACTION	15	18.2295	Action

10. Identify the top 5 months with the highest revenue and display the revenue generated in each .

```
%%sql

SELECT

   DATE_FORMAT(payment_date, '%Y-%m') AS month,

   SUM(amount) AS revenue

FROM payment

GROUP BY month

ORDER BY revenue DESC

LIMIT 5;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies

5 rows affected.
```

month	revenue
2005-07	28373.89
2005-08	24072.13
2005-06	9631.88
2005-05	4824.43
2006-02	514.18

Normalisation & CTE

1. First Normal Form (1NF):

a. Identify a table in the Sakila database that violates 1NF. Explain how youwould normalize it to achieve 1NF.

- Example: Hypothetical table violating 1NF: 'customer_phones' with multiple phone numbers in a single column
- Fix: Split into separate rows or move phone numbers into a new table with customer_id foreign key

2. Second Normal Form (2NF):

a. Choose a table in Sakila and describe how you would determine whether it is in 2NF. If it violates 2NF, explain the steps to normalize it.

- Example table: 'film_actor' with composite key (film_id, actor_id)
- If it includes actor_name, it's not in 2NF. Fix: Move actor_name to 'actor' table, relate via actor_id.

3. Third Normal Form (3NF):

a. Identify a table in Sakila that violates 3NF. Describe the transitive dependencies present and outline the steps to normalize the table to 3NF.

- Example table: 'customer' has address_id, and address contains city_id, and city_id links to city
- If customer also contains city_name (transitive dependency), move city details to 'city' table and normalize

4. Normalization Process:

a. Take a specific table in Sakila and guide through the process of normalizing it from the initial unnormalized form up to at least 2NF.

- Example: Start with unnormalized rental_customer table with repeated fields
- Step 1: Split repeating groups into rows (1NF)
- Step 2: Ensure each non-key attribute depends on whole key (2NF)
- Step 3: Remove transitive dependencies like storing city_name with address_id (3NF)

5. CTE Basics:

a. Write a query using a CTE to retrieve the distinct list of actor names and the number of films they have acted in from the actor and film_actor tables.

%%sql

```
WITH actor_films AS (
   SELECT actor_id, COUNT(film_id) AS film_count
   FROM film_actor
   GROUP BY actor_id
)
SELECT
   a.first_name, a.last_name, af.film_count
FROM actor a
JOIN actor_films af ON a.actor_id = af.actor_id limit 20;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
20 rows affected.

 first_name   last_name       film_count

 PENELOPE     GUINESS         19

 NICK         WAHLBERG        25

 ED           CHASE           22

 JENNIFER     DAVIS           22

 JOHNNY       LOLLOBRIGIDA    29

 BETTE        NICHOLSON       20

 GRACE        MOSTEL          30

 MATTHEW      JOHANSSON       20

 JOE          SWANK           25

 CHRISTIAN    GABLE           22
```

first_name	last_name	film_count
ZERO	CAGE	25
KARL	BERRY	31
UMA	WOOD	35
VIVIEN	BERGEN	30
CUBA	OLIVIER	28
FRED	COSTNER	27
HELEN	VOIGHT	32
DAN	TORN	22
BOB	FAWCETT	25
LUCILLE	TRACY	30

6. CTE with Joins:

a. Create a CTE that combines information from the film and language tables to display the film title,language name, and rental rate.

```
%%sql

WITH film_info AS (

   SELECT f.film_id, f.title, l.name AS language, f.rental_rate

   FROM film f

   JOIN language l ON f.language_id = l.language_id

)

SELECT * FROM film_info limit 10; #use limit to avoid large output
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies

10 rows affected.
```

film_id	title	language	rental_rate
1	ACADEMY DINOSAUR	English	0.99
2	ACE GOLDFINGER	English	4.99
3	ADAPTATION HOLES	English	2.99
4	AFFAIR PREJUDICE	English	2.99
5	AFRICAN EGG	English	2.99
6	AGENT TRUMAN	English	2.99
7	AIRPLANE SIERRA	English	4.99
8	AIRPORT POLLOCK	English	4.99
9	ALABAMA DEVIL	English	2.99
10	ALADDIN CALENDAR	English	4.99

7.CTE for Aggregation:

a. Write a query using a CTE to find the total revenue generated by each customer (sum of payments) from the customer and payment tables.

```
%%sql
WITH customer_revenue AS (
   SELECT customer_id, SUM(amount) AS total_revenue
   FROM payment
   GROUP BY customer_id
)
SELECT c.customer_id, c.first_name, c.last_name, cr.total_revenue
FROM customer c
```

```
JOIN customer_revenue cr ON c.customer_id = cr.customer_id limit 10; #use limit to avoid large output
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
10 rows affected.
```

customer_id	first_name	last_name	total_revenue
1	MARY	SMITH	118.68
2	PATRICIA	JOHNSON	128.73
3	LINDA	WILLIAMS	135.74
4	BARBARA	JONES	81.78
5	ELIZABETH	BROWN	144.62
6	JENNIFER	DAVIS	93.72
7	MARIA	MILLER	151.67
8	SUSAN	WILSON	92.76
9	MARGARET	MOORE	89.77
10	DOROTHY	TAYLOR	99.75

8.CTE with Window Functions:

a. Utilize a CTE with a window function to rank films based on their rental duration from the film table.

```
%%sql
WITH ranked_films AS (
   SELECT title, rental_duration,
          RANK() OVER (ORDER BY rental_duration DESC) AS duration_rank
   FROM film
)
SELECT * FROM ranked_films limit 10; #use limit to avoid large output
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
10 rows affected.

 title                  rental_duration   duration_rank

 ADAPTATION HOLES       7                 1

 ANONYMOUS HUMAN        7                 1

 ARGONAUTS TOWN         7                 1

 BIKINI BORROWERS       7                 1

 BLACKOUT PRIVATE       7                 1

 BLANKET BEVERLY        7                 1

 BOILED DARES           7                 1

 BOONDOCK BALLROOM      7                 1

 BORN SPINAL            7                 1

 BORROWERS BEDAZZLED    7                 1
```

9.CTE and Filtering:

a. Create a CTE to list customers who have made more than two rentals, and then join this CTE with the customer table to retrieve additional customer details.

```
%%sql
WITH frequent_customers AS (
    SELECT customer_id, COUNT(*) AS rental_count
    FROM rental
    GROUP BY customer_id
    HAVING COUNT(*) > 2
)
SELECT c.*
FROM customer c
JOIN frequent_customers fc ON c.customer_id = fc.customer_id limit 10; # use limit to avoid large output
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
10 rows affected.
```

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
1	1	MARY	SMITH	MARY.SMITH@sakilacustomer.org	5	1	2006-02-14 22:04:36	2006-02-15 04:57:20
2	1	PATRICIA	JOHNSON	PATRICIA.JOHNSON@sakilacustomer.org	6	1	2006-02-14 22:04:36	2006-02-15 04:57:20
3	1	LINDA	WILLIAMS	LINDA.WILLIAMS@sakilacustomer.org	7	1	2006-02-14 22:04:36	2006-02-15 04:57:20
4	2	BARBARA	JONES	BARBARA.JONES@sakilacustomer.org	8	1	2006-02-14 22:04:36	2006-02-15 04:57:20
5	1	ELIZABETH	BROWN	ELIZABETH.BROWN@sakilacustomer.org	9	1	2006-02-14	2006-02-15

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
							22:04:36	04:57:20
6	2	JENNIFER	DAVIS	JENNIFER.DAVIS@sakilacustomer.org	10	1	2006-02-14 22:04:36	2006-02-15 04:57:20
7	1	MARIA	MILLER	MARIA.MILLER@sakilacustomer.org	11	1	2006-02-14 22:04:36	2006-02-15 04:57:20
8	2	SUSAN	WILSON	SUSAN.WILSON@sakilacustomer.org	12	1	2006-02-14 22:04:36	2006-02-15 04:57:20
9	2	MARGARET	MOORE	MARGARET.MOORE@sakilacustomer.org	13	1	2006-02-14 22:04:36	2006-02-15 04:57:20
10	1	DOROTHY	TAYLOR	DOROTHY.TAYLOR@sakilacustomer.org	14	1	2006-02-14 22:04:36	2006-02-15 04:57:20

.10.CTE for Date Calculations:

a. Write a query using a CTE to find the total number of rentals made each month, considering the rental_date from the rental table

```
%%sql

WITH monthly_rentals AS (

  SELECT DATE_FORMAT(rental_date, '%Y-%m') AS rental_month,
        COUNT(*) AS total_rentals
    FROM rental
    GROUP BY rental_month
)
SELECT * FROM monthly_rentals limit 10;
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
5 rows affected.
```

rental_month	total_rentals
2005-05	1156
2005-06	2311
2005-07	6709
2005-08	5686
2006-02	182

11. CTE and Self-Join:

a. Create a CTE to generate a report showing pairs of actors who have appeared in the same film together, using the film_actor table.

```
%%sql
WITH actor_pairs AS (
    SELECT fa1.film_id, fa1.actor_id AS actor1, fa2.actor_id AS actor2
    FROM film_actor fa1
    JOIN film_actor fa2 ON fa1.film_id = fa2.film_id AND fa1.actor_id < fa2.actor_id
)
SELECT ap.film_id, a1.first_name AS actor1_first, a1.last_name AS actor1_last,
       a2.first_name AS actor2_first, a2.last_name AS actor2_last
FROM actor_pairs ap
```

```
JOIN actor a1 ON ap.actor1 = a1.actor_id

JOIN actor a2 ON ap.actor2 = a2.actor_id

limit 10; #use limit to avoid large output
```

```
 * mysql+mysqlconnector://root:***@localhost/mavenmovies
10 rows affected.

film_id  actor1_first  actor1_last  actor2_first  actor2_last

1        PENELOPE      GUINESS      CHRISTIAN     GABLE

1        PENELOPE      GUINESS      LUCILLE       TRACY

1        PENELOPE      GUINESS      SANDRA        PECK

1        PENELOPE      GUINESS      JOHNNY        CAGE

1        PENELOPE      GUINESS      MENA          TEMPLE

1        PENELOPE      GUINESS      WARREN        NOLTE

1        PENELOPE      GUINESS      OPRAH         KILMER

1        PENELOPE      GUINESS      ROCK          DUKAKIS

1        PENELOPE      GUINESS      MARY          KEITEL

23       PENELOPE      GUINESS      JENNIFER      DAVIS
```

12. CTE for Recursive Search:

a. Implement a recursive CTE to find all employees in the staff table who report to a specific manager, considering the reports_to column

```
%%sql

WITH RECURSIVE staff_hierarchy AS (

   SELECT staff_id, first_name, last_name, reports_to

   FROM staff
```

```
    WHERE staff_id = 1  -- Replace with manager's ID

    UNION ALL

    SELECT s.staff_id, s.first_name, s.last_name, s.reports_to
    FROM staff s
    JOIN staff_hierarchy sh ON s.reports_to = sh.staff_id
)
SELECT * FROM staff_hierarchy;
```