INFORMATION TECHNOLOGY ENGINEERING

CLASS : ITE M1

NAME : Thouerng sreymealea

Part 1: Basic Queries

1. Retrieve Basic Data

• Write a query to list the first 10 films, displaying the film_id, title, and release_year

```
#list the first 10 films
SELECT
film_id,title
 FROM sakila.film
 ORDER BY film_id
 LIMIT 10;
```

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

• Find the total number of customers in the database

```
LIMIT 10;


#Find the total number of customers
SELECT COUNT(*) AS total_of_customers
FROM sakila.customer;
```

total_of_customers
599

2. Filter Data

• List all movies in the film table released in the year 2006

```
#List all movies in the film table released in the year 2006
SELECT
film_id,title,release_year
FROM sakila.film
WHERE release_year='2006';
```

film_id	title	release_year
1	ACADEMY DINOSAUR	2006
2	ACE GOLDFINGER	2006
3	ADAPTATION HOLES	2006
4	AFFAIR PREJUDICE	2006
5	AFRICAN EGG	2006
6	AGENT TRUMAN	2006
7	AIRPLANE SIERRA	2006
8	AIRPORT POLLOCK	2006
9	ALABAMA DEVIL	2006
10	ALADDIN CALENDAR	2006
11	ALAMO VIDEOTAPE	2006
12	ALASKA PHANTOM	2006
13	ALI FOREVER	2006
14	ALICE FANTASIA	2006
15	ALIEN CENTER	2006
16	ALLEY EVOLUTION	2006
17	ALONE TRIP	2006
18	ALTER VICTORY	2006
19	AMADEUS HOLY	2006
20	AMELIE HELLFIGHTERS	2006
21	AMERICAN CIRCUS	2006

- Retrieve all customers whose last name starts with the letter “S”

```
#Retrieve all customers whose last name starts with the letter “S”
SELECT
customer_id,first_name,last_name
FROM sakila.customer
WHERE LEFT(last_name, 1) = 's';
```

customer_id	first_name	last_name
1	MARY	SMITH
34	REBECCA	SCOTT
51	ALICE	STEWART
52	JULIE	SANCHEZ
75	TAMMY	SANDERS
92	TINA	SIMMONS
105	DAWN	SULLIVAN
126	ELLEN	SIMPSON
127	ELAINE	STEVENS
144	CLARA	SHAW
158	VERONICA	STONE
163	CATHY	SPENCER
165	LORRAINE	STEPHENS
178	MARION	SNYDER
195	VANESSA	SIMS
204	ROSEMARY	SCHMIDT
228	ALLISON	STANLEY
254	MAXINE	SILVA
268	NINA	SOTO
279	DIANNE	SHELTON
283	FELICIA	SUTTON

3. Sort and Limit Results

- Find the top 5 longest movies (length), displaying title and length

```
#Find the top 5 longest movies (length), displaying title and length
SELECT title,length
FROM sakila.film
 ORDER BY length DESC
LIMIT 5;
```

title	length
GANGS PRIDE	185
HOME PITY	185
CHICAGO NORTH	185
CONTROL ANTHEM	185
DARN FORRESTER	185

• List the 5 most recent rentals, showing rental_id, rental_date, and customer_id

```
#List the 5 most recent rentals, showing rental_id, rental_date, and customer_id
SELECT rental_id,rental_date,customer_id
FROM sakila.rental
ORDER BY rental_date DESC
LIMIT 5;
```

rental_id	rental_date	customer_id
11739	2006-02-14 15:16:03	373
14616	2006-02-14 15:16:03	532
11676	2006-02-14 15:16:03	216
15966	2006-02-14 15:16:03	374
13486	2006-02-14 15:16:03	274

Part 2: Advanced Queries

4. Joins

• Write a query to find the names of actors who appeared in the movie “Academy Dinosaur”

```
#Write a query to find the names of actors who appeared in the movie “Academy Dinosaur"(Joins)
SELECT actor.first_name, actor.last_name
FROM sakila.actor
JOIN sakila.film_actor ON actor.actor_id = film_actor.actor_id
JOIN sakila.film ON film_actor.film_id = film.film_id
WHERE film.title = 'Academy Dinosaur';
```

first_name	last_name
PENELOPE	GUINESS
CHRISTIAN	GABLE
LUCILLE	TRACY
SANDRA	PECK
JOHNNY	CAGE
MENA	TEMPLE
WARREN	NOLTE
OPRAH	KILMER
ROCK	DUKAKIS
MARY	KEITEL

• Retrieve a list of customers along with the titles of movies they rented. Include customer_id, first_name, last_name, and title.

```
# Retrieve a list of customers along with the titles of movies they rented. Include customer_id, first_name, last_name, and title.
SELECT
    customer.customer_id,
    customer.first_name,
    customer.last_name,
    film.title
FROM sakila.customer
JOIN sakila.rental ON customer.customer_id = rental.customer_id
JOIN sakila.inventory ON rental.inventory_id = inventory.inventory_id
JOIN sakila.film ON inventory.film_id = film.film_id
ORDER BY customer.customer_id, film.title
LIMIT 0, 1000;

 #Find the total revenue generated by all rentals.
```

customer_id	first_name	last_name	title
1	MARY	SMITH	ADAPTATION HOLES
1	MARY	SMITH	AMISTAD MIDSUMMER
1	MARY	SMITH	ATTACKS HATE
1	MARY	SMITH	BIKINI BORROWERS
1	MARY	SMITH	CLOSER BANG
1	MARY	SMITH	CONFIDENTIAL INTERVIEW
1	MARY	SMITH	DALMATIONS SWEDEN
1	MARY	SMITH	DETECTIVE VISION
1	MARY	SMITH	DOORS PRESIDENT
1	MARY	SMITH	EXPECATIONS NATURAL

5. Aggregations

• Find the total revenue generated by all rentals

```
 #Find the total revenue generated by all rentals.
SELECT SUM(amount) AS total_revenue
FROM sakila.payment;
```

total_revenue
67406.56

• Retrieve the number of films in each category, sorted by the category name

```
#Retrieve the number of films in each category, sorted by the category name.
SELECT
    category.name AS category_name,
    COUNT(film.film_id) AS film_count
FROM sakila.category
JOIN sakila.film_category ON category.category_id = film_category.category_id
JOIN sakila.film ON film_category.film_id = film.film_id
GROUP BY category.name
ORDER BY category.name;
```

category_name	film_count
Action	64
Animation	66
Children	60
Classics	57
Comedy	58
Documentary	68
Drama	62
Family	69
Foreign	73
Games	61
Horror	56
Music	51
New	63
Sci-Fi	61
Sports	74

6. Subqueries

- List the names of customers who have rented more than 30 movies

```
#List the names of customers who have rented more than 30 movies
SELECT first_name,last_name
FROM sakila.customer
WHERE customer_id IN(
SELECT customer_id
FROM sakila.rental
GROUP BY customer_id
HAVING COUNT(rental_id) >30
);
```

first_name	last_name
MARY	SMITH
ELIZABETH	BROWN
MARIA	MILLER
HELEN	HARRIS
MICHELLE	CLARK
JESSICA	HALL
SHIRLEY	ALLEN
CYNTHIA	YOUNG
ANGELA	HERNANDEZ
MELISSA	KING
VIRGINIA	GREEN
MARTHA	GONZALEZ
CATHERINE	CAMPBELL
DIANE	COLLINS

- Find all films that have never been rented

```
# Find all films that have never been rented
SELECT film_id, title
FROM sakila.film
WHERE film_id NOT IN (
    SELECT film_id
    FROM sakila.inventory
    WHERE inventory_id IN (
        SELECT inventory_id
        FROM sakila.rental
    )
);
```

film_id	title
14	ALICE FANTASIA
33	APOLLO TEEN
36	ARGONAUTS TOWN
38	ARK RIDGEMONT
41	ARSENIC INDEPENDENCE
87	BOONDOCK BALLROOM
108	BUTCH PANTHER
128	CATCH AMISTAD
144	CHINATOWN GLADIATOR
148	CHOCOLATE DUCK
171	COMMANDMENTS EXPRESS
192	CROSSING DIVORCE

7.Case Statements

```
#7. Case Statements
SELECT title,length,
CASE
WHEN length < 60 THEN 'short'
WHEN length BETWEEN 60 AND 120 THEN 'Medium'
WHEN length > 120 THEN 'Long'
END AS movie_legth_category
FROM sakila.film;
```

title	length	movie_legth_category
ACADEMY DINOSAUR	86	Medium
ACE GOLDFINGER	48	short
ADAPTATION HOLES	50	short
AFFAIR PREJUDICE	117	Medium
AFRICAN EGG	130	Long

Part 3: Database Management 8.

8. Insert Data

.Insert customer

```
#insert data to table
INSERT INTO sakila.customer (first_name, last_name, email, address_id, activ
VALUES ('John', 'Doe', 'john.doe@example.com', 1, 1, NOW(), 1);
SELECT *
FROM sakila.customer
WHERE first_name = 'John' AND last_name = 'Doe';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_upda
600	1	John	Doe	john.doe@example.com	1	1	2024-12-13 13:23:23	2024-12-1

.Insert film

```
INSERT INTO sakila.film (title, description, release_year, language_id, rent
VALUES ('Epic Adventure', 'A thrilling tale of courage and survival.', 2024,
SELECT *
FROM sakila.film
WHERE title = 'Epic Adventure';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

film_id	title	description	release_year	language_id	original_language_id	rental_duration
1001	Epic Adventure	A thrilling tale of courage and survival.	2024	1	NULL	3
NULL	NULL	NULL	NULL	NULL	NULL	NULL

9. Update data

```
#9. UPDATE DATA
#Update the email of a specific customer in the customer table
UPDATE sakila.customer
SET email = 'newjohn.doe@example.com'
WHERE first_name = 'John' AND last_name = 'Doe';
SELECT customer_id, first_name, last_name, email
FROM sakila.customer
WHERE first_name = 'John' AND last_name = 'Doe';

#9. Change the rental rate for all movies in the film table to 4.99 where the length is greater th
SET SQL_SAFE_UPDATES = 0;
UPDATE sakila.film
SET rental_rate = 4.99
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

customer_id	first_name	last_name	email
607	John	Doe	newjohn.doe@example.com

```
#9. Change the rental rate for all movies in the film table to 4.99 where the length is greater th
SET SQL_SAFE_UPDATES = 0;
UPDATE sakila.film
SET rental_rate = 4.99
WHERE length > 120;
SET SQL_SAFE_UPDATES = 1;
SET SQL_SAFE_UPDATES = 0;

#10. Delete Data
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
6	10:43:22	SELECT customer_id, first_name, last_name, email F...	2 row(s) returned	0.000 sec / 0.000 sec
7	10:44:05	SET SQL_SAFE_UPDATES = 0	0 row(s) affected	0.000 sec
8	10:44:05	UPDATE sakila.film SET rental_rate = 4.99 WHERE ...	0 row(s) affected Rows matched: 457 Changed: 0 ...	0.032 sec
9	10:44:05	SET SQL_SAFE_UPDATES = 1	0 row(s) affected	0.000 sec
10	10:44:05	SET SQL_SAFE_UPDATES = 0	0 row(s) affected	0.000 sec

10. Delete Data

```
#10. Delete Data
# Delete a customer from the customer table who hasn't rented any movies.
DELETE FROM sakila.customer
WHERE customer_id NOT IN (
    SELECT DISTINCT customer_id
    FROM sakila.rental
);

# Remove all movies in the film table that were released before 2000.
DELETE FROM sakila.film
WHERE release_year < 2000;
SELECT film_id, title, release_year
FROM sakila.film;
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	10:49:29	DELETE FROM sakila.customer WHERE customer_id ...	0 row(s) affected	0.000 sec

```
# Remove all movies in the film table that were released before 2000.
DELETE FROM sakila.film
WHERE release_year < 2000;
SELECT film_id, title, release_year
FROM sakila.film;

# Create a Report
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch row

film_id	title	release_year
1	ACADEMY DINOSAUR	2006
2	ACE GOLDFINGER	2006
3	ADAPTATION HOLES	2006
4	AFFAIR PREJUDICE	2006
5	AFRICAN EGG	2006
6	AGENT TRUMAN	2006
7	AIRPLANE SIERRA	2006

film 10

Result Grid

Form Editor

Apply Revert

11. Create a Report

```sql
# Create a Report
#11. showing the top 10 customers who rented the most movies
SELECT c.customer_id, c.first_name, c.last_name, COUNT(r.rental_id) AS movies_rented
FROM sakila.customer c
JOIN sakila.rental r ON c.customer_id = r.customer_id
GROUP BY c.customer_id, c.first_name, c.last_name
ORDER BY movies_rented DESC
LIMIT 10;

#12. Revenue Analysis(Calculate the total revenue generated per category. Display the category nam
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

customer_id	first_name	last_name	movies_rented
148	ELEANOR	HUNT	46
526	KARL	SEAL	45
144	CLARA	SHAW	42
236	MARCIA	DEAN	42
75	TAMMY	SANDERS	41

Result 13 | Read Only

Output

12. Revenue Analysis

```sql
#12. Revenue Analysis(Calculate the total revenue generated per category. Display the category nam
SELECT c.name AS category_name, SUM(p.amount) AS total_revenue
FROM sakila.category c
JOIN sakila.film_category fc ON c.category_id = fc.category_id
JOIN sakila.film f ON fc.film_id = f.film_id
JOIN sakila.inventory i ON f.film_id = i.film_id
JOIN sakila.rental r ON i.inventory_id = r.inventory_id
JOIN sakila.payment p ON r.rental_id = p.rental_id
GROUP BY c.category_id, c.name;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category_name	total_revenue
Action	4375.85
Animation	4656.30
Children	3655.55
Classics	3639.59
Comedy	4383.58

Result 14 | Read Only

13. Rental Trends

```
#13. Rental Trends( Identify the busiest rental month. Display the month and the total number of
SELECT MONTH(r.rental_date) AS rental_month, COUNT(r.rental_id) AS total_rentals
FROM sakila.rental r
GROUP BY rental_month
ORDER BY total_rentals DESC
LIMIT 1;

#Create Views
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

rental_month	total_rentals
7	6709

Result Grid

14. Create Views

```
#14. Create a view category_revenue
CREATE VIEW category_revenue AS
SELECT c.name AS category_name, SUM(p.amount) AS total_revenue
FROM sakila.category c
JOIN sakila.film_category fc ON c.category_id = fc.category_id
JOIN sakila.film f ON fc.film_id = f.film_id
JOIN sakila.inventory i ON f.film_id = i.film_id
JOIN sakila.rental r ON i.inventory_id = r.inventory_id
JOIN sakila.payment p ON r.rental_id = p.rental_id
GROUP BY c.category_id, c.name;
DROP VIEW category_revenue;

#15. Create an index on the title column of the film table
CREATE INDEX idx_film_title ON sakila.film(title);
DROP INDEX idx_film_title ON sakila.film;
#Add an index to the rental_date column of the rental table
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	11:05:51	CREATE VIEW category_revenue AS SELECT c.name...	0 row(s) affected	0.016 sec

- Views
  - actor_info
  - customer_list
  - customer_rentals
  - film_list
  - nicer_but_slower_
  - sales_by_film_cat
  - sales_by_store
  - staff_list

15. Create Indexes

```
#15. Create an index on the title column of the film table
CREATE INDEX idx_film_title ON sakila.film(title);
DROP INDEX idx_film_title ON sakila.film;
#Add an index to the rental_date column of the rental table
CREATE INDEX idx_rental_date ON sakila.rental(rental_date);

#16. Stored Procedure (Write a stored procedure get_customer_rentals)
DELIMITER //
CREATE PROCEDURE get_customer_rentals(IN cust_id INT)
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	11:07:33	CREATE INDEX idx_film_title ON sakila.film(title)	0 row(s) affected, 1 warning(s): 1831 Duplicate index `id...	0.062 sec

```
#Add an index to the rental_date column of the rental table
CREATE INDEX idx_rental_date ON sakila.rental(rental_date);
DROP INDEX idx_rental_date ON sakila.rental;

#16. Stored Procedure (Write a stored procedure get_customer_rentals)
DELIMITER //
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	11:09:56	CREATE INDEX idx_rental_date ON sakila.rental(rental...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.156 sec

16. Stored Procedure

```
#16. Stored Procedure (Write a stored procedure get_customer_rentals)
DELIMITER //
CREATE PROCEDURE get_customer_rentals(IN cust_id INT)
BEGIN
    SELECT f.title
    FROM sakila.film f
    JOIN sakila.inventory i ON f.film_id = i.film_id
    JOIN sakila.rental r ON i.inventory_id = r.inventory_id
    WHERE r.customer_id = cust_id;
END //

DELIMITER ;
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	11:13:34	CREATE PROCEDURE get_customer_rentals(IN cust_i...	0 row(s) affected	0.015 sec