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 database tables to ensure the accuracy, validity, and consistency of the data. They define how data can be inserted, updated, or deleted, helping maintain data integrity by preventing invalid data from being stored in the database.

- Primary key
- Foreign key
- Not Null
- Unique
- Check
- Default

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

The **NOT NULL constraint** is applied to ensure essential fields are never left blank. A **primary key** inherently disallows NULL values because it must uniquely and reliably identify each row in the table.

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.

Violating constraints leads to errors that prevent the operation from completing, ensuring data integrity. Each type of constraint has specific rules and associated error messages. Understanding these rules helps design and manage databases effectively, avoiding unintended consequences.

ERROR: Column 'emp_name' cannot be null

ERROR: Duplicate entry '1' for key 'emp_id_UNIQUE'

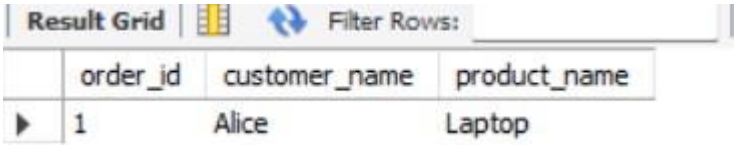
ERROR: Cannot add or update a child row: a foreign key constraint fails

8. Consider the following three tables:

-- 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

ans:

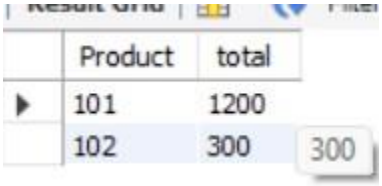


Result Grid | Filter Rows:

order_id	customer_name	product_name
1	Alice	Laptop

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

ans :



Product	total
101	1200
102	300

300

10 . 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.

ans:

order_id	customer_name	product_id	Quantity
2	Bob	101	2
1	Alice	101	2
2	Bob	102	1
1	Alice	102	1
2	Bob	101	3
1	Alice	101	3

SQL COMMANDS

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

Primary key:	Foreign key:
actor_id actor_award_id advisor_id category_id city_id country_id customer_id film_id inventory_id investor_id language_id payment_id rental_id staff_id store_id	Category_id Film_id

2. List all details of actors

actor_id	first_name	last_name	last_update
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
27	JULIA	MCQUEEN	2006-02-15 04:34:33

3. -List all customer information from DB.

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
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	2	LISA	ANDERSON	LISA.ANDERSON@sakilacustomer.org	15	1	2006-02-14 22:04:36	2006-02-15 04:57:20
12	1	NANCY	THOMAS	NANCY.THOMAS@sakilacustomer.org	16	1	2006-02-14 22:04:36	2006-02-15 04:57:20
13	2	KAREN	JACKSON	KAREN.JACKSON@sakilacustomer.org	17	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

4 -List different countries

Afghanistan
Algeria
American Samoa
Angola
Anguilla
Argentina
Armenia
Australia
Austria
Azerbaijan
Bahrain
Bangladesh
Belarus
Bolivia
Brazil
Brunei
Bulgaria
Cambodia
Cameroon
Canada
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
Germany
Greece
Greenland
Holy See (Vatican City State)
Hong Kong
Hungary
India
Indonesia
Iran
Iraq
Israel
Italy
Japan
Kazakstan
Kenya
Kuwait
Latvia
Liechtenstein
Lithuania
Madagascar
Malawi
Malaysia
Mexico
Moldova
Morocco
Mozambique
Myanmar
Nauru
Nepal
Netherlands
New Zealand
Nigeria
North Korea
Oman
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
Sudan
Sweden
Switzerland
Taiwan
Tanzania
Thailand
Tonga
Tunisia
Turkey
Turkmenistan
Tuvalu
Ukraine
United Arab Emirates
United Kingdom
United States
Venezuela
Vietnam
Virgin Islands, U.S.
Yemen
Yugoslavia
Zambia

5 -Display all active customers.

Full_name
AARON SELBY
ADAM GOOCH
ADRIAN CLARY
AGNES BISHOP
ALAN KAHN
ALBERT CROUSE
ALBERTO HENNING
ALEX GRESHAM
ALEXANDER FENNELL
ALFRED CASILLAS
ALFREDO MCADAMS
ALICE STEWART
ALICIA MILLS
ALLAN CORNISH

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

rental_id
76
573
1185
1422
1476
1725
2308
2363
3284
4526
4611
5244
5326
6163
7273
7841
8033
8074
8116
8326
9571

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

	title
▶	ACADEMY DINOSAUR
	ADAPTATION HOLES
	AFRICAN EGG
	AIRPLANE SIERRA
	AIRPORT POLLOCK
	ALADDIN CALENDAR
	ALAMO VIDEOTAPE
	ALASKA PHANTOM
	ALICE FANTASIA
	ALLEY EVOLUTION
	ALTER VICTORY
	AMADEUS HOLY

8 - List the total number of films whose replacement cost is greater than $15 and less than $20.

	Total_number
▶	214

9 - Display the count of unique first names of actors

	Total_distint_names
▶	128

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

	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
	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’.

	first_name
▶	BARBARA
	BETTY
	BRENDA

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

	title
▶	ACE GOLDFINGER
	AFFAIR PREJUDICE
	AFRICAN EGG
	ALAMO VIDEOTAPE
	AMISTAD MIDSUMMER

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

ANGELA
AMY
ANNA
AMANDA
ANN
ALICE
ASHLEY
ANDREA
ANNE
ANNIE
ANITA
AMBER
APRIL
ALICIA
AUDREY
ANNETTE
ANA
ALMA
AGNES
ARLENE
ALLISON
ANTHONY
ANDREW
ARTHUR
ALBERT
ADAM
AARON
ALAN

ANTONIO
ALLEN
ALFRED
ALEXANDER
ALEX
ALVIN
ANGEL
ANDRE
ARNOLD
ADRIAN
ALLAN
ARMANDO
ALFREDO
ALBERTO
ANDY
AUSTIN

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

PATRICIA
LINDA
BARBARA
MARIA
LISA
SANDRA
DONNA
LAURA
JESSICA
CYNTHIA
ANGELA
MELISSA
BRENDA
ANNA
REBECCA
VIRGINIA
PAMELA
MARTHA
DEBRA
AMANDA
TERESA
GLORIA
CHRISTINA
THERESA
ANDREA
SARA

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

city
Abha
Acuña
Adana
Addis Abeba

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

first_name
JENNIFER
VIRGINIA
STEPHANIE
JANICE
NICOLE
DENISE
BONNIE
ANNIE
CONNIE
MONICA
JUANITA
ANITA

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

first_name
BRENDA
FRANCES
IRENE
CRYSTAL
TRACY
GRACE
ERIN
ERICA
BRITTANY
KRISTEN
KRISTIN
ARLENE

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

first_name
ALICE
ANNIE
ANITA
AMBER
APRIL
AGNES
AARON
ALLEN
ALVIN
ANGEL
ANDRE
ALLAN

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

first_name
ANTONIO
ARMANDO
ALFREDO
ALBERTO

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

title
ACADEMY DINOSAUR
AGENT TRUMAN
AIRPLANE SIERRA
ALABAMA DEVIL
ALASKA PHANTOM
ALI FOREVER
ALTER VICTORY
AMADEUS HOLY
ANTHEM LUKE
APOLLO TEEN
ARACHNOPHOBIA R...
ARGONAUTS TOWN

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

title

22 - Get the top 50 actors using limit operator.

Full_name
PENELOPE GUINESS
NICK WAHLBERG
ED CHASE
JENNIFER DAVIS
JOHNNY LOLLOBRIGIDA
BETTE NICHOLSON
GRACE MOSTEL
MATTHEW JOHANSSON
JOE SWANK
CHRISTIAN GABLE
ZERO CAGE
KARL BERRY

23 - Get the distinct film ids from inventory table.

film_id
1
2
3
4
5
6
7
8
9
10
11
12

Question 1:

-- Retrieve the total number of rentals made in the Sakila database.

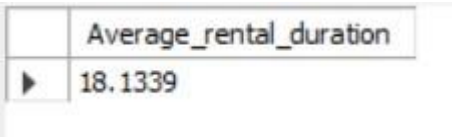
-- Hint: Use the COUNT() function.

Total_number
16044

Question 2:

-- Find the average rental duration (in days) of movies rented from the Sakila database.

-- Hint: Utilize the AVG() function.



Average_rental_duration
18.1339

Question 3:

-- Display the first name and last name of customers in uppercase.

-- Hint: Use the UPPER () function.

Upper_case_Names
PENELOPE GUINESS
NICK WAHLBERG
ED CHASE
JENNIFER DAVIS
JOHNNY LOLLOBRIGIDA
BETTE NICHOLSON
GRACE MOSTEL
MATTHEW JOHANSSON
JOE SWANK
CHRISTIAN GABLE
ZERO CAGE
KARL BERRY

Question 4:

-- Extract the month from the rental date and display it alongside the rental ID.

-- Hint: Employ the MONTH() function.

Month	rental_id
5	1
5	2
5	3
5	4
5	5
5	6
5	7
5	8
5	9
5	10
5	11
5	12

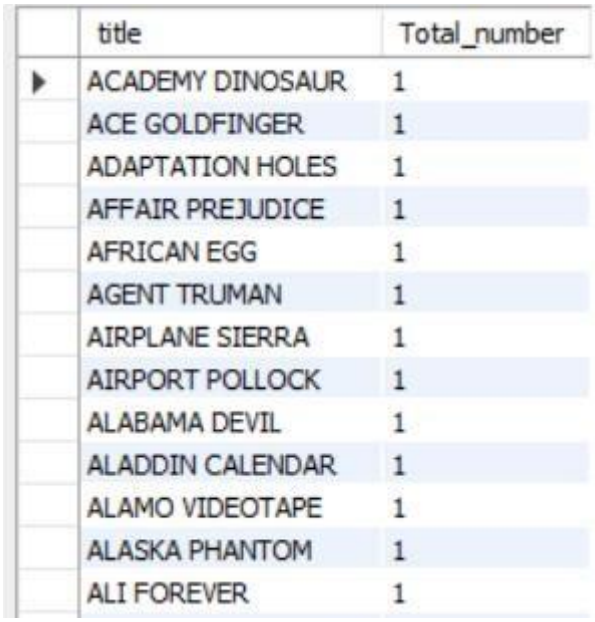
Question 5:

-- Retrieve the count of rentals for each customer (display customer ID and the count of rentals).

-- Hint: Use COUNT () in conjunction with GROUP BY.

customer_id	count_rental
1	32
2	27
3	26
4	22
5	38
6	28
7	33
8	24
9	23
10	25
11	24
12	28

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.

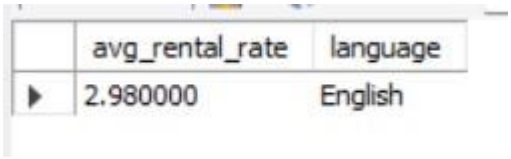


title	Total_number
ACADEMY DINOSAUR	1
ACE GOLDFINGER	1
ADAPTATION HOLES	1
AFFAIR PREJUDICE	1
AFRICAN EGG	1
AGENT TRUMAN	1
AIRPLANE SIERRA	1
AIRPORT POLLOCK	1
ALABAMA DEVIL	1
ALADDIN CALENDAR	1
ALAMO VIDEOTAPE	1
ALASKA PHANTOM	1
ALI FOREVER	1

-- Question 8:

-- Find the average rental rate of movies in each language.

-- Hint: JOIN film and language tables, then use AVG () and GROUP BY.



avg_rental_rate	language
2.980000	English

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.

Title	First_name	Last_name
ACADEMY DINOSAUR	JOEL	FRANCISCO
ACADEMY DINOSAUR	GABRIEL	HARDER
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
ACADEMY DINOSAUR	ROBERT	BAUGHMAN
ACADEMY DINOSAUR	HENRY	BILLINGSLEY
ACADEMY DINOSAUR	SERGIO	STANFIELD

-- 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.

Full_name
SISSY SOBIESKI
RICHARD PENN
EMILY DEE
CHRIS DEPP
GROUCHO WILLIAMS

-- 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

full_name	SUM(amount)
MARY SMITH	118.68
PATRICIA JOHNSON	128.73
LINDA WILLIAMS	135.74
BARBARA JONES	81.78
ELIZABETH BROWN	144.62
JENNIFER DAVIS	93.72
MARIA MILLER	151.67
SUSAN WILSON	92.76
MARGARET MOORE	89.77
DOROTHY TAYLOR	99.75
LISA ANDERSON	106.76
NANCY THOMAS	103.72
KAREN JACKSON	131.73
BETTY WHITE	117.72

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.

title	full_name
ACADEMY DINOSAUR	CECIL VINES
ACADEMY DINOSAUR	MATTIE HOFFMAN
ACE GOLDFINGER	MATTIE HOFFMAN
ADAPTATION HOLES	MATTIE HOFFMAN
AFFAIR PREJUDICE	CECIL VINES
AFFAIR PREJUDICE	MATTIE HOFFMAN
AFRICAN EGG	MATTIE HOFFMAN
AGENT TRUMAN	CECIL VINES
AGENT TRUMAN	MATTIE HOFFMAN
AIRPLANE SIERRA	CECIL VINES
AIRPLANE SIERRA	MATTIE HOFFMAN
AIRPORT POLLOCK	MATTIE HOFFMAN
ALABAMA DEVIL	CECIL VINES
ALABAMA DEVIL	MATTIE HOFFMAN
ALADDIN CALENDAR	CECIL VINES
ALADDIN CALENDAR	MATTIE HOFFMAN

-- 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.

title	Number_of_times
ACADEMY DINOSAUR	23
ACE GOLDFINGER	7
ADAPTATION HOLES	12
AFFAIR PREJUDICE	23
AFRICAN EGG	12

-- 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.

first_name
MARIE
JANET
FRANCES
ANN
DIANE
ALICE
JULIE
HEATHER
TERESA
GLORIA
JEAN
CHERYL
MILDRED
JOAN
ASHLEY
KELLY

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

	Ranks	first_name
▶	1	CHARLOTTE
	2	TOMMY
	3	MANUEL
	4	ANDREW
	5	DELORES
	6	NELSON
	7	CASSANDRA
	8	MINNIE
	9	ELLEN
	10	DANNY
	11	APRIL
	12	DEANNA
	13	RAYMOND
	14	THEODORE
	15	RONALD

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

	cumulative_revenue
▶	2980.00

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

	film_length	avg_rental_duration
▶	46	5.1275
	47	4.8165
	48	5.1774
	49	4.9130
	50	4.6016
	51	5.1589
	52	5.0351
	53	4.9322
	54	5.0805
	55	4.2619
	56	4.7143
	57	5.0254
	58	5.2051
	59	5.2203
	60	5.0339

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

each_category	title	category_id
1	AMADEUS HOLY	1
2	AMERICAN CIRCUS	1
3	ANTITRUST TOMATOES	1
1	ALTER VICTORY	2
2	ANACONDA CONFESSIONS	2
3	ARGONAUTS TOWN	2
1	BACKLASH UNDEFEATED	3
2	BEAR GRACELAND	3
3	BENEATH RUSH	3
1	ALICE FANTASIA	4
2	ARIZONA BANG	4
3	BEAST HUNCHBACK	4
1	AIRPLANE SIERRA	5
2	ANTHEM LUKE	5
3	BRINGING HYSTERICAL	5

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

-- across all customers.

rental_counts	Total	avg(rental_rate)
16044	47211.56	2.942630

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

month_year	total_revenue
2005-05	4823.44
2005-06	9629.89
2005-07	28368.91
2005-08	24070.14
2006-02	514.18

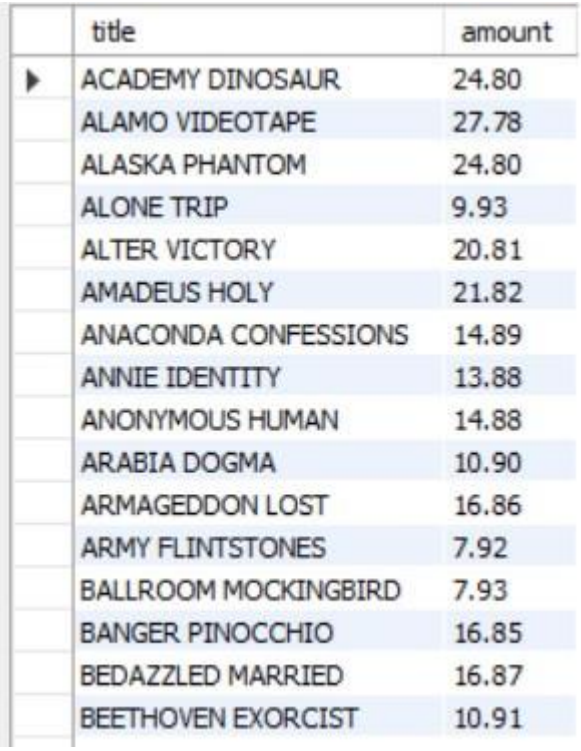
7. Identify the customers whose total spending on rentals falls within the top 20% of all custommer;

first_name	amount
MARY	118.68
PATRICIA	128.73
LINDA	135.74
BARBARA	81.78
ELIZABETH	144.62
JENNIFER	93.72
MARIA	151.67
SUSAN	92.76
MARGARET	89.77
DOROTHY	99.75
LISA	106.76
NANCY	103.72
KAREN	131.73
BETTY	117.72
HELEN	134.68
SANDRA	118.72

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

category_name	rental_count	running_total
Sports	1179	1179
Animation	1166	2345
Action	1112	3457
Sci-Fi	1101	4558
Family	1096	5654
Drama	1060	6714
Documentary	1050	7764
Foreign	1033	8797
Games	969	9766
Children	945	10711
Comedy	941	11652
New	940	12592
Classics	939	13531
Horror	846	14377
Travel	837	15214
Music	830	16044

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



title	amount
ACADEMY DINOSAUR	24.80
ALAMO VIDEOTAPE	27.78
ALASKA PHANTOM	24.80
ALONE TRIP	9.93
ALTER VICTORY	20.81
AMADEUS HOLY	21.82
ANACONDA CONFESSIONS	14.89
ANNIE IDENTITY	13.88
ANONYMOUS HUMAN	14.88
ARABIA DOGMA	10.90
ARMAGEDDON LOST	16.86
ARMY FLINTSTONES	7.92
BALLROOM MOCKINGBIRD	7.93
BANGER PINOCCHIO	16.85
BEDAZZLED MARRIED	16.87
BEETHOVEN EXORCIST	10.91

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

months	amount
7	28368.91
8	24070.14
6	9629.89
5	4823.44
2	514.18

Normalisation and CTE

1. First Normal Form (1NF):

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

Ans: each table cell contain a single value

No repeating values in a group

No repeating groups

Each record(row) is unique

violating First Normal Form (1NF) could be the film table if it contains a repeating group or multivalued fields.

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.

Ans: Should be INF

No partial Dependency

Occur when there is composite Key

Steps to Normalize the Table to Achieve 2NF

To remove the partial dependency, split the table into two or more tables where each non-key attribute is fully dependent on the entire primary key.

Step 1: Split the film_actor Table

Original film_actor Table:

film_id actor_id last_update

Split it into two tables:

A table to store the relationship between films and actors.

A table to store the last_update attribute.

Step 2: Create New Tables

film_actor Table (for relationships):

film_id actor_id

film_actor_update Table (for last_update):

film_id last_update

Step 3: Populate the New Tables

Remove last_update from the original table and move it to film_actor_update, ensuring the relationship between film_id and last_update remains intact.

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.

ANS:

Step 1: Identify Dependencies

Direct Dependencies:

payment_id → customer_id, staff_id, rental_id, amount, payment_date, customer_name

Transitive Dependency:

payment_id → customer_id → customer_name

Step 2: Decompose the Table

To remove the transitive dependency, create a new table for customer information and move customer_name to this table.

Step 3: Create New Tables

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.

Normalization Process:

Let's take the **`rental`** table from the Sakila database and guide it through the normalization process.

Step 1: Analyze the Initial Unnormalized Form (UNF)

An **unnormalized form (UNF)** table may have repeating groups or multivalued attributes. Assume the `rental` table in its UNF form looks like this:

rental_id	rental_date	inventory_id	customer_id	return_date	staff_id	film_title	category
1	2024-01-01 10:00:00	101	201	2024-01-03 15:00:00	1	"Avengers"	Action
2	2024-01-01 12:00:00	102	202	2024-01-03 17:00:00	2	"Finding Nemo"	Animation
3	2024-01-02 09:00:00	103	203	2024-01-04 11:00:00	1	"Avengers"	Action

Violations in UNF:

1. **Repeating Group**: Columns like `film_title` and `category` are attributes of the `inventory_id`, not the `rental_id`.
2. **Multivalued Dependencies**: Data about the films (e.g., `film_title` and `category`) is repeated for every rental.

Step 2: Convert to First Normal Form (1NF)

To convert the table to **1NF**:

1. Ensure atomicity (no multivalued attributes or repeating groups).
2. Break down repeating groups into separate rows or tables.

Revised Table in 1NF:

rental_id	rental_date	inventory_id	customer_id	return_date	staff_id
1	2024-01-01 10:00:00	101	201	2024-01-03 15:00:00	1
2	2024-01-01 12:00:00	102	202	2024-01-03 17:00:00	2
3	2024-01-02 09:00:00	103	203	2024-01-04 11:00:00	1

Create a separate table for film information (`film` table):

inventory_id	film_title	category
101	Avengers	Action
102	Finding Nemo	Animation
103	Avengers	Action

Step 3: Convert to Second Normal Form (2NF)

To achieve **2NF**:

1. Ensure the table is already in **1NF**.
2. Eliminate **partial dependencies**, where non-key attributes depend on only part of a composite primary key.

Key Analysis:

- **`rental` table primary key**: `rental_id` (single-column key).
- All columns in the `rental` table depend fully on `rental_id`, so it is in 2NF.
- **`film` table primary key**: `inventory_id`.
  - `film_title` and `category` fully depend on `inventory_id`, so it is also in 2NF.

Normalized Tables (1NF to 2NF)

1. **`rental` Table**:

rental_id	rental_date	inventory_id	customer_id	return_date	staff_id
1	2024-01-01 10:00:00	101	201	2024-01-03 15:00:00	1
2	2024-01-01 12:00:00	102	202	2024-01-03 17:00:00	2
3	2024-01-02 09:00:00	103	203	2024-01-04 11:00:00	1

2. **`film` Table**:

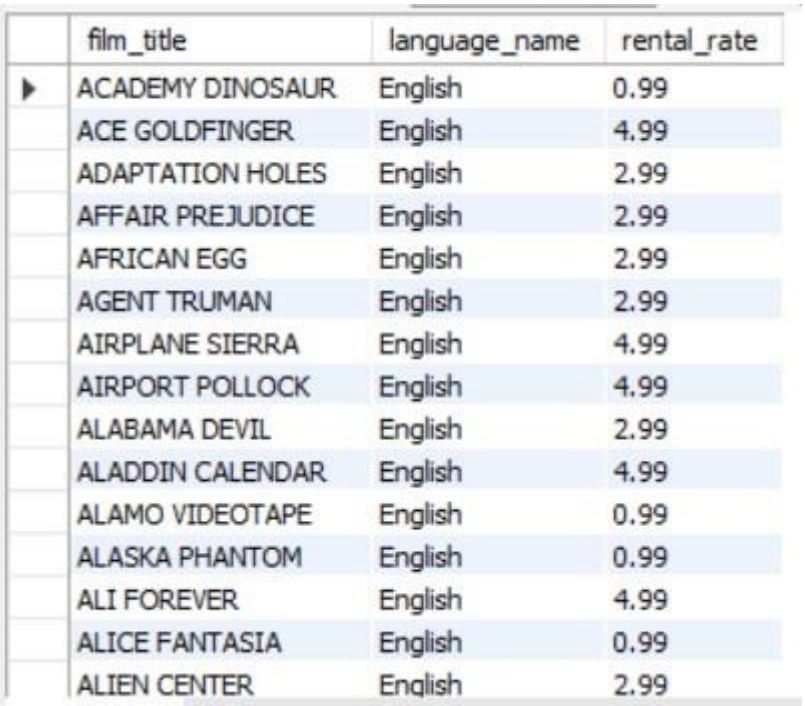
inventory_id	film_title	category
101	Avengers	Action
102	Finding Nemo	Animation
103	Avengers	Action

---

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.



film_title	language_name	rental_rate
ACADEMY DINOSAUR	English	0.99
ACE GOLDFINGER	English	4.99
ADAPTATION HOLES	English	2.99
AFFAIR PREJUDICE	English	2.99
AFRICAN EGG	English	2.99
AGENT TRUMAN	English	2.99
AIRPLANE SIERRA	English	4.99
AIRPORT POLLOCK	English	4.99
ALABAMA DEVIL	English	2.99
ALADDIN CALENDAR	English	4.99
ALAMO VIDEOTAPE	English	0.99
ALASKA PHANTOM	English	0.99
ALI FOREVER	English	4.99
ALICE FANTASIA	English	0.99
ALIEN CENTER	English	2.99

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.

Name_	Total_revenue
0	67406.56

-- 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.

film_id	film_title	rental_duration	rank_
3	ADAPTATION HOLES	7	1
27	ANONYMOUS HUMAN	7	1
36	ARGONAUTS TOWN	7	1
70	BIKINI BORROWERS	7	1
78	BLACKOUT PRIVATE	7	1
80	BLANKET BEVERLY	7	1
84	BOILED DARES	7	1
87	BOONDOCK BALLROOM	7	1
88	BORN SPINAL	7	1
89	BORROWERS BEDAZZLED	7	1
92	BOWFINGER GABLES	7	1
94	BRAVEHEART HUMAN	7	1
97	BRIDE INTRIGUE	7	1
99	BRINGING HYSTERICAL	7	1
100	BROOKLYN DESERT	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

customer_id	first_name	last_name	email	rental_count
148	ELEANOR	HUNT	ELEANOR.HUNT@sakilacustomer.org	46
526	KARL	SEAL	KARL.SEAL@sakilacustomer.org	45
236	MARCIA	DEAN	MARCIA.DEAN@sakilacustomer.org	42
144	CLARA	SHAW	CLARA.SHAW@sakilacustomer.org	42
75	TAMMY	SANDERS	TAMMY.SANDERS@sakilacustomer.org	41
469	WESLEY	BULL	WESLEY.BULL@sakilacustomer.org	40
197	SUE	PETERS	SUE.PETERS@sakilacustomer.org	40
468	TIM	CARY	TIM.CARY@sakilacustomer.org	39
137	RHONDA	KENNEDY	RHONDA.KENNEDY@sakilacustomer.org	39
178	MARION	SNYDER	MARION.SNYDER@sakilacustomer.org	39
295	DAISY	BATES	DAISY.BATES@sakilacustomer.org	38
5	ELIZABETH	BROWN	ELIZABETH.BROWN@sakilacustomer.org	38
459	TOMMY	COLLAZO	TOMMY.COLLAZO@sakilacustomer.org	38
410	CURTIS	IRBY	CURTIS.IRBY@sakilacustomer.org	38
176	JUNE	CARROLL	JUNE.CARROLL@sakilacustomer.org	37

-- 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

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

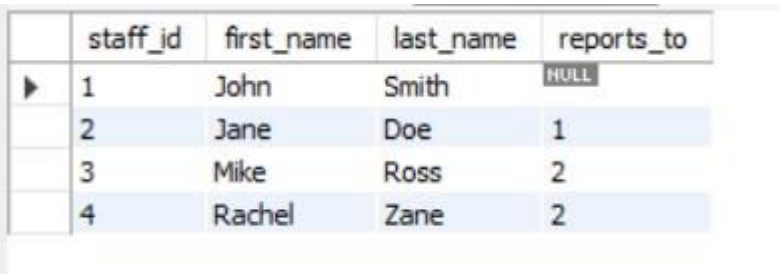
-- together, using the film_actor table.

actor1_id	actor1_first_name	actor1_last_name	actor2_id	actor2_first_name	actor2_last_name	film_title
40	JOHNNY	CAGE	188	ROCK	DUKAKIS	ACADEMY DINOSAUR
40	JOHNNY	CAGE	198	MARY	KEITEL	ACADEMY DINOSAUR
40	JOHNNY	CAGE	162	OPRAH	KILMER	ACADEMY DINOSAUR
40	JOHNNY	CAGE	108	WARREN	NOLTE	ACADEMY DINOSAUR
40	JOHNNY	CAGE	53	MENA	TEMPLE	ACADEMY DINOSAUR
188	ROCK	DUKAKIS	198	MARY	KEITEL	ACADEMY DINOSAUR
10	CHRISTIAN	GABLE	40	JOHNNY	CAGE	ACADEMY DINOSAUR
10	CHRISTIAN	GABLE	188	ROCK	DUKAKIS	ACADEMY DINOSAUR
10	CHRISTIAN	GABLE	198	MARY	KEITEL	ACADEMY DINOSAUR
10	CHRISTIAN	GABLE	162	OPRAH	KILMER	ACADEMY DINOSAUR
10	CHRISTIAN	GABLE	108	WARREN	NOLTE	ACADEMY DINOSAUR
10	CHRISTIAN	GABLE	30	SANDRA	PECK	ACADEMY DINOSAUR
10	CHRISTIAN	GABLE	53	MENA	TEMPLE	ACADEMY DINOSAUR
10	CHRISTIAN	GABLE	20	LUCILLE	TRACY	ACADEMY DINOSAUR
1	PENELOPE	GUINESS	40	JOHNNY	CAGE	ACADEMY DINOSAUR

-- 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



staff_id	first_name	last_name	reports_to
1	John	Smith	NULL
2	Jane	Doe	1
3	Mike	Ross	2
4	Rachel	Zane	2