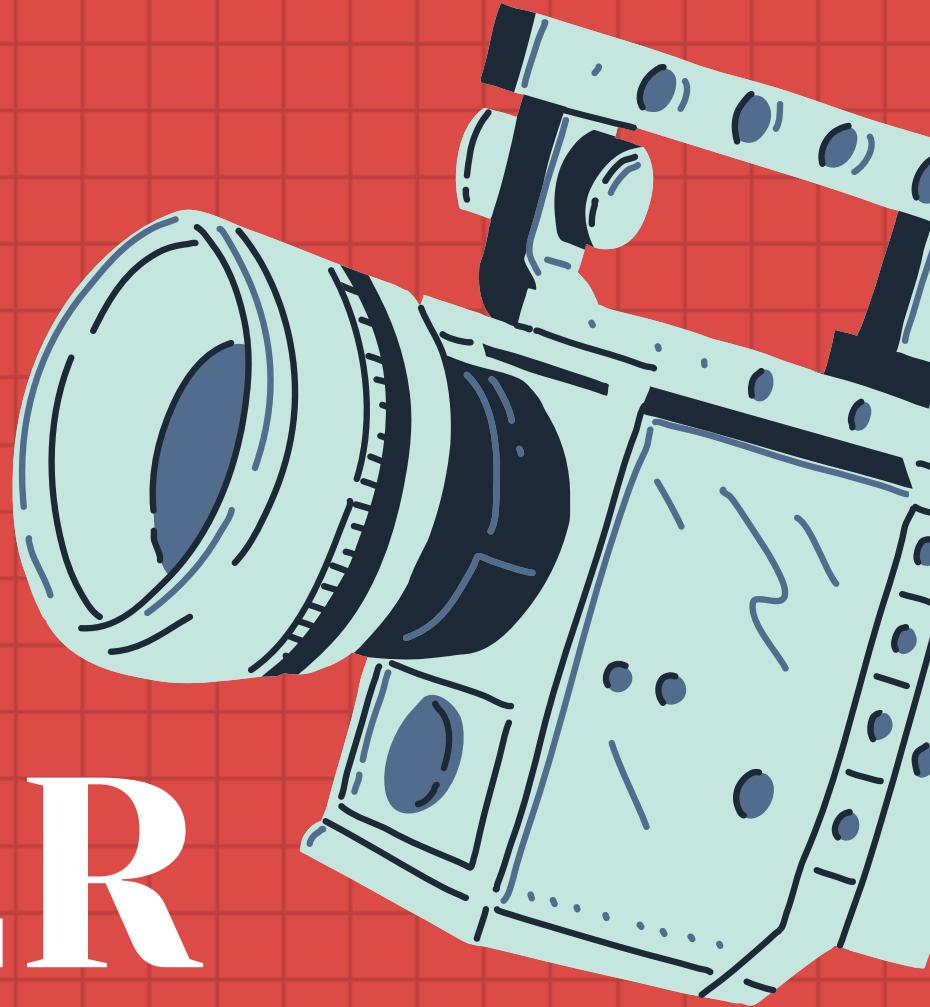
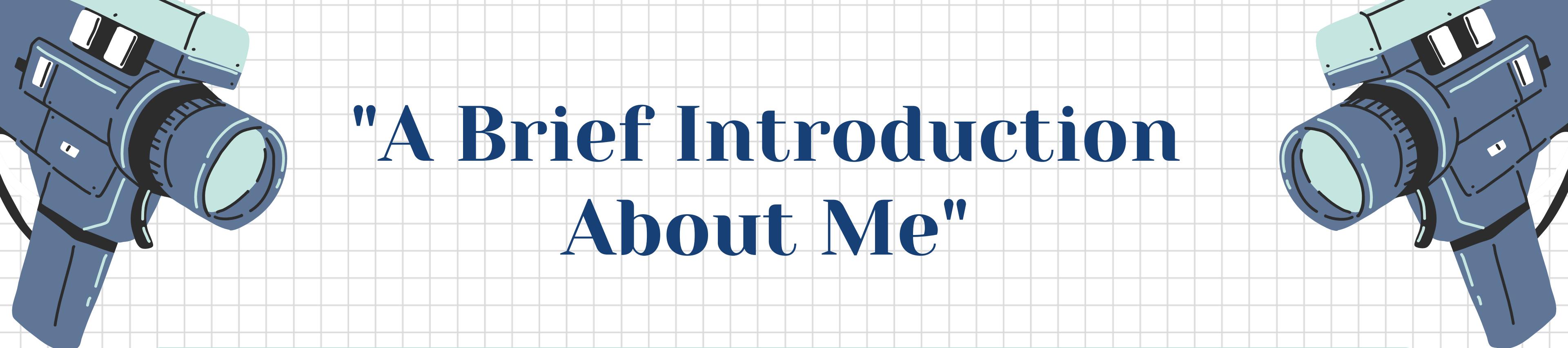


TFC
THE-FILM-CORNER

Online movie rental store





"A Brief Introduction About Me"

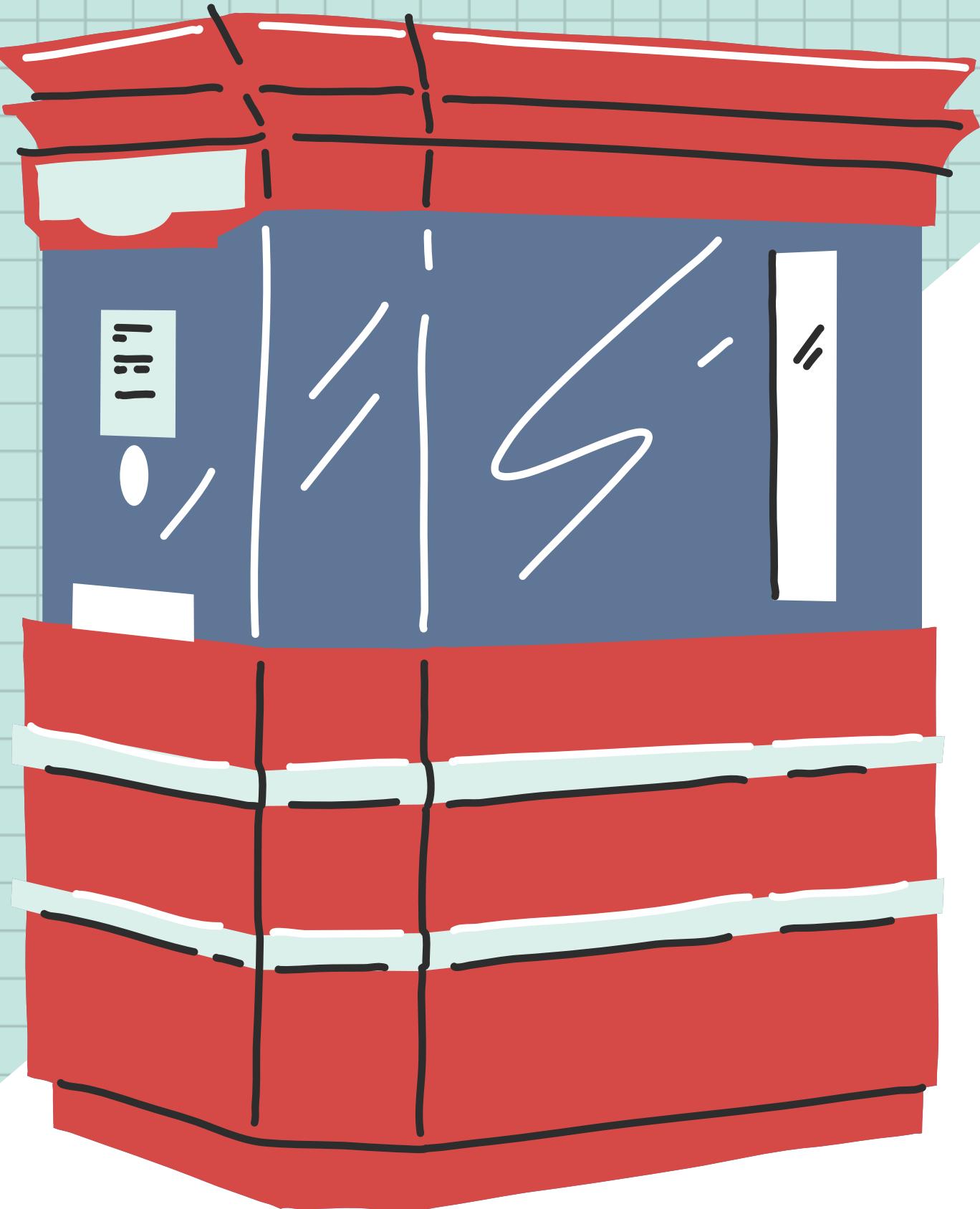
Hello, my name is Aditya Mishra, and I am currently working as a Retail Manager. In this project, I am taking on the role of an analyst recently hired by Source Media, which has acquired a business called The FILM CORNER. As a data analyst, my task is to help the company manage its database by writing queries and analyzing data, as this information is crucial to the company's success. I will work with various tables, generate insights, and assist the company in making informed decisions and solving problems to ensure the success of this new business.

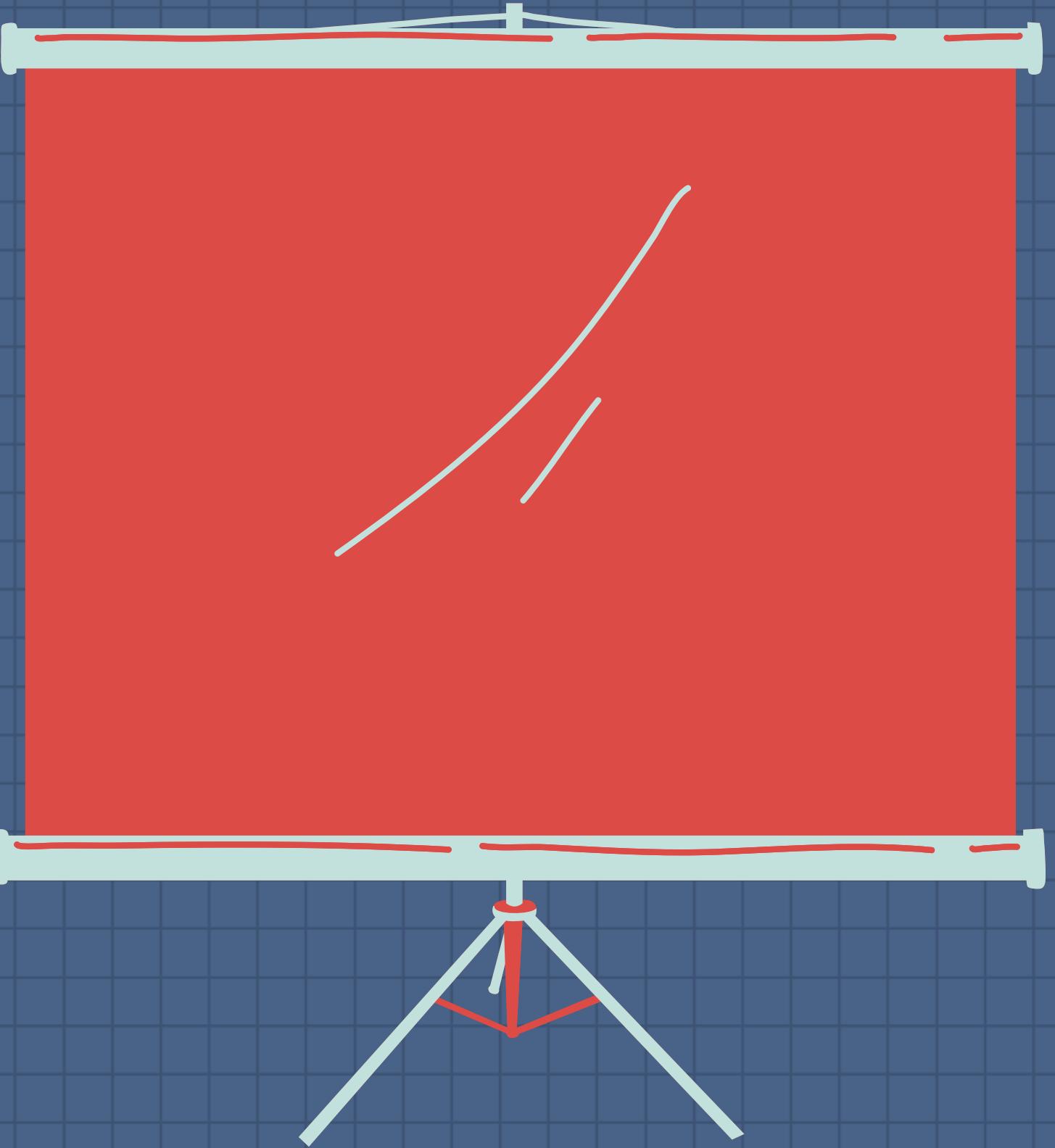
THE-FILM-CORNER



Welcome to "THE FILM CORNER," one of the best online movie rental stores. Store offers a vast collection of movies in all languages, genres, and types, including animated films, available to customers from various cities and countries.

Whether you're in the mood for a classic film or the latest blockbuster, our diverse selection has something for every movie enthusiast. Recently, "THE FILM CORNER" was acquired by Source Media, enhancing our ability to provide an even more comprehensive and seamless movie rental experience to our valued customers worldwide.

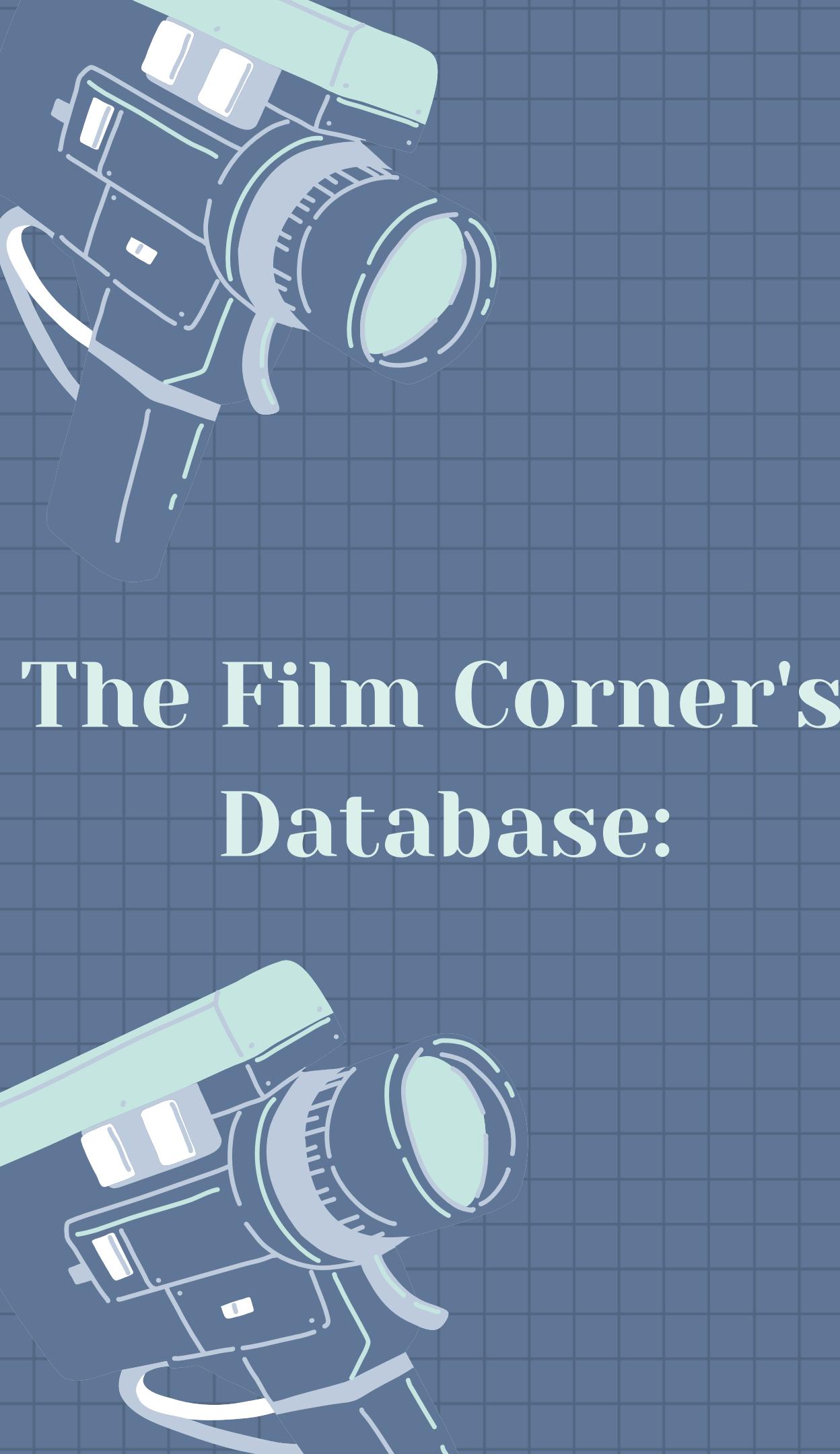




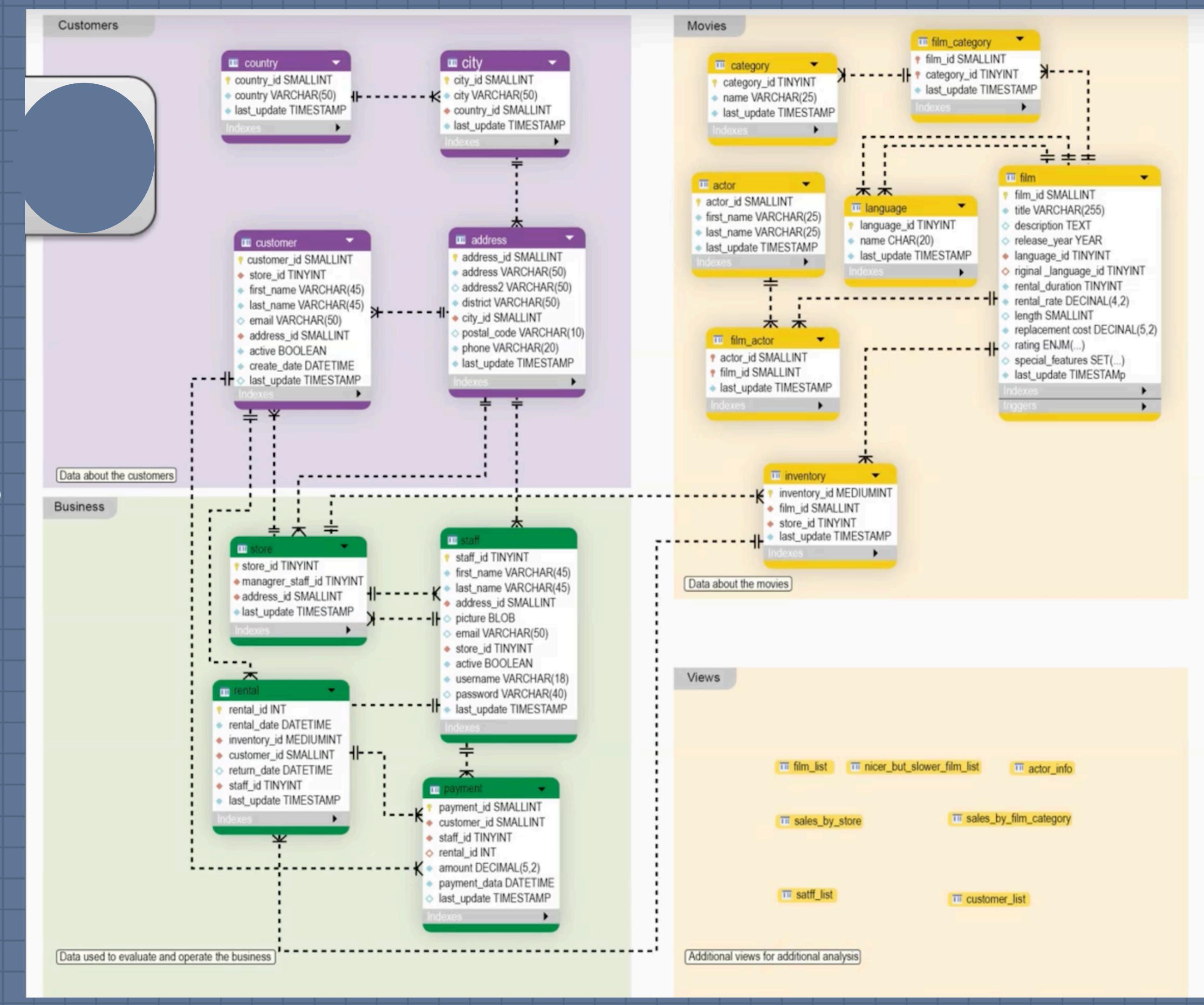
X X

Let's dive in !!

X X



The Film Corner's Database:



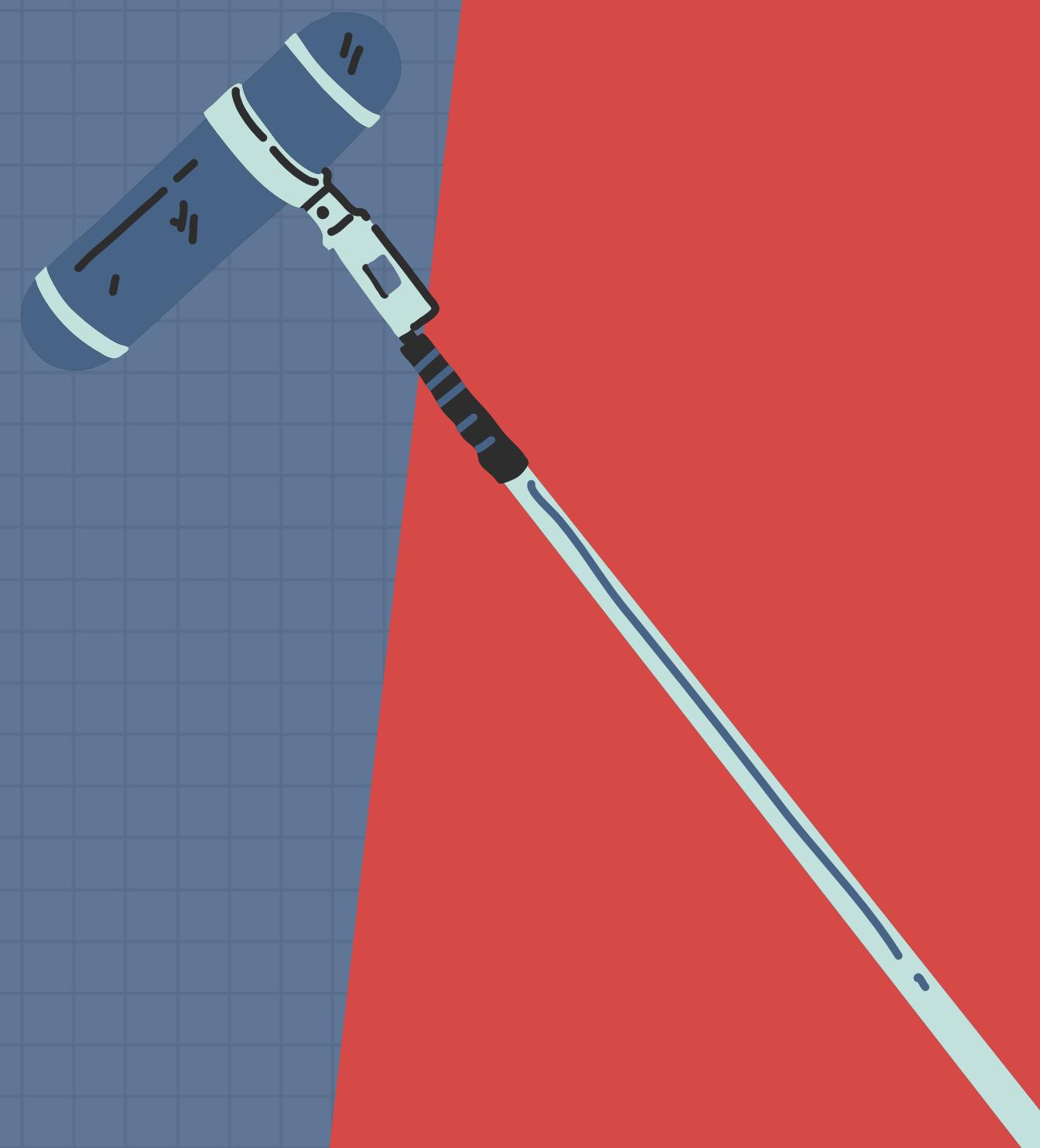
The marketing manager has requested a list of all customers of the rental shop, including each customer's first name, last name, and email address.

```
4 SELECT
5 first_name,
6 last_name,
7 email
8 FROM CUSTOMER
9
```

Data Output Messages Notifications

≡+

	first_name	last_name	email
1	MARY	SMITH	MARY.SMITH@sakilacustomer.org
2	PATRICIA	JOHNSON	PATRICIA.JOHNSON@sakilacustomer.org
3	LINDA	WILLIAMS	LINDA.WILLIAMS@sakilacustomer.org
4	BARBARA	JONES	BARBARA.JONES@sakilacustomer.org
5	ELIZABETH	BROWN	ELIZABETH.BROWN@sakilacustomer.org
6	JENNIFER	DAVIS	JENNIFER.DAVIS@sakilacustomer.org
7	MARIA	MILLER	MARIA.MILLER@sakilacustomer.org



The marketing manager now requests that you organize the customer list you previously provided by last name, starting from Z to A. Additionally, in the event that there are multiple entries with the same last name, please use the first name as a secondary criterion for ordering, also from Z to A.

```
4  
5  SELECT  
6    last_name,  
7    first_name,  
8    email  
9  FROM customer  
10 ORDER BY last_name DESC , first_name DESC
```

X

X

	last_name text	first_name text	email text
1	YOUNG	CYNTHIA	CYNTHIA.YOUNG@sakilacustomer.org
2	YEE	MARVIN	MARVIN.YEE@sakilacustomer.org
3	YANEZ	LUIS	LUIS.YANEZ@sakilacustomer.org
4	WYMAN	BRIAN	BRIAN.WYMAN@sakilacustomer.org
5	WRIGHT	BRENDA	BRENDA.WRIGHT@sakilacustomer.org
6	WREN	TYLER	TYLER.WREN@sakilacustomer.org
7	WOODS	FLORENCE	FLORENCE.WOODS@sakilacustomer.org

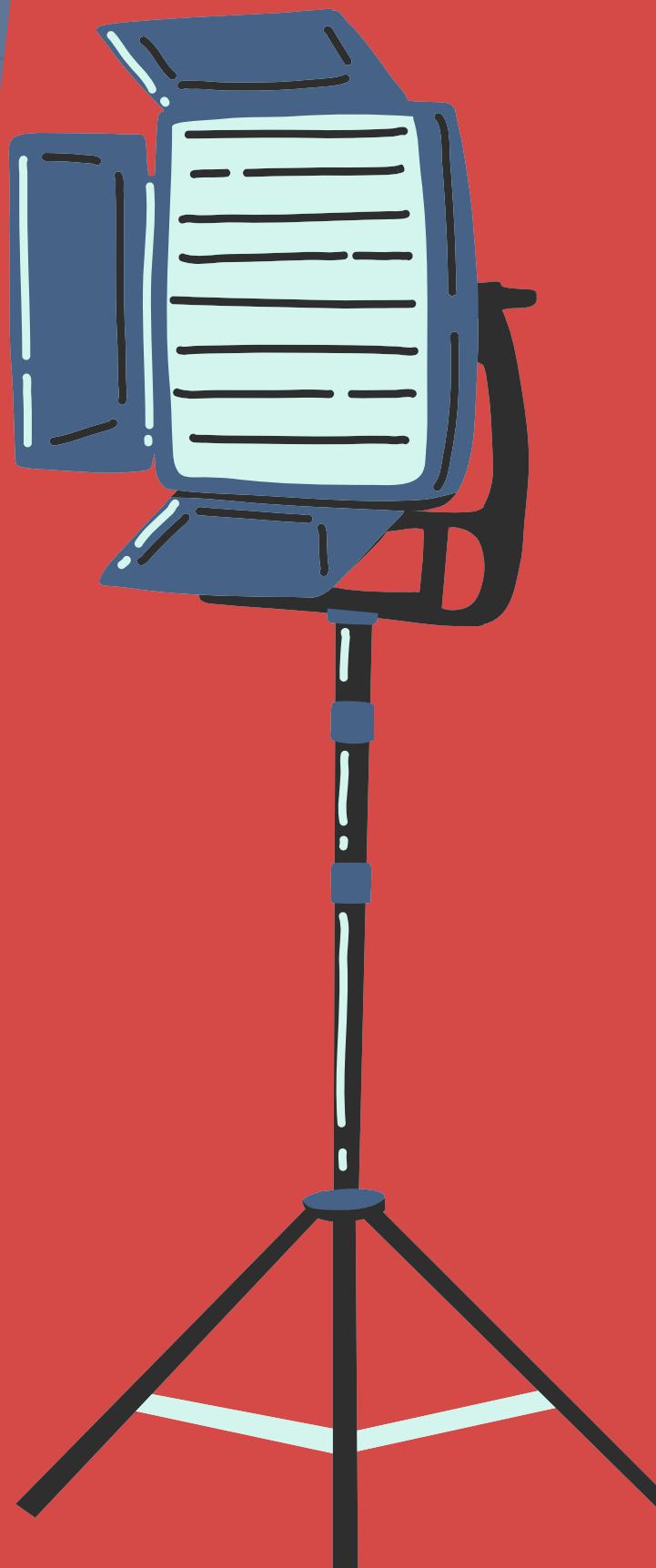
A member of the marketing team is inquiring about the various prices that have been paid in the business in the past. To simplify their task, you can also arrange those prices from highest to lowest.

```
3  
4 SELECT DISTINCT  
5 amount  
6 FROM payment  
7 ORDER BY amount DESC  
8
```

Data Output Messages Notifications

amount
numeric (5,2) 

1	11.99
2	10.99
3	9.99
4	9.98
5	8.99
6	8.97
7	7.00



How many films does the company have ?

```
select count(*) from film
```



Data Output		Messages	Notifications
count bigint			

1	1000
---	------

What is the latest Rental date ?

```
2
3 SELECT
4 rental_date
5 FROM rental
6 ORDER BY rental_date DESC
7 LIMIT 1
```

Data Output		Messages	Notifications
rental_date timestamp with time zone			

1	2020-02-14 16:16:03+01
---	------------------------

The marketing team is organizing an event during which they intend to provide a gift voucher to the customer identified by customer ID 100. Initially, they need to determine the total number of payments made by this customer, as they plan to issue a \$1 voucher per payment made.

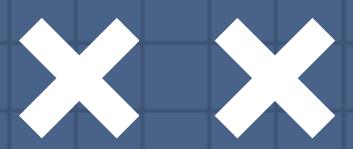
```
4  
5  SELECT  
6    count(*)  
7  FROM payment  
8 WHERE customer_id = 100
```

Data Output		
	count	bigint
1	24	

From the support team, you get a request to find out the last name of the customer with the first name Erica.

```
3  SELECT  
4    last_name  
5  FROM customer  
6 WHERE First_name = 'ERICA'
```

Data Output		
	last_name	text
1	MATTHEWS	



You receive a request from the inventory manager inquiring about the number of rentals that have not yet been returned.

Secondly, The sales manager requests a list of all payment IDs where the amount is \$2 or less.

Additionally, they require a list that includes each payment ID along with its corresponding amount.

```
4  
5  SELECT COUNT(*)  
6  FROM rental  
7  WHERE return_date is null;  
8  
9  
10 SELECT  
11 payment_id,  
12 amount  
13 FROM payment  
14 WHERE amount <= 2  
15  
16
```

Data Output Messages Notifications

	payment_id	amount
1	16050	1.99
2	16051	0.99
3	16053	0.99
4	16056	1.99
5	16059	0.99
6	16062	0.99
7	16064	0.99
8	16077	0.99
9	16081	1.99



The support manager inquires about a list detailing payments made by customers 322, 346, and 354.

Specifically, they seek amounts for these customers that are either less than \$2 or greater than \$10.

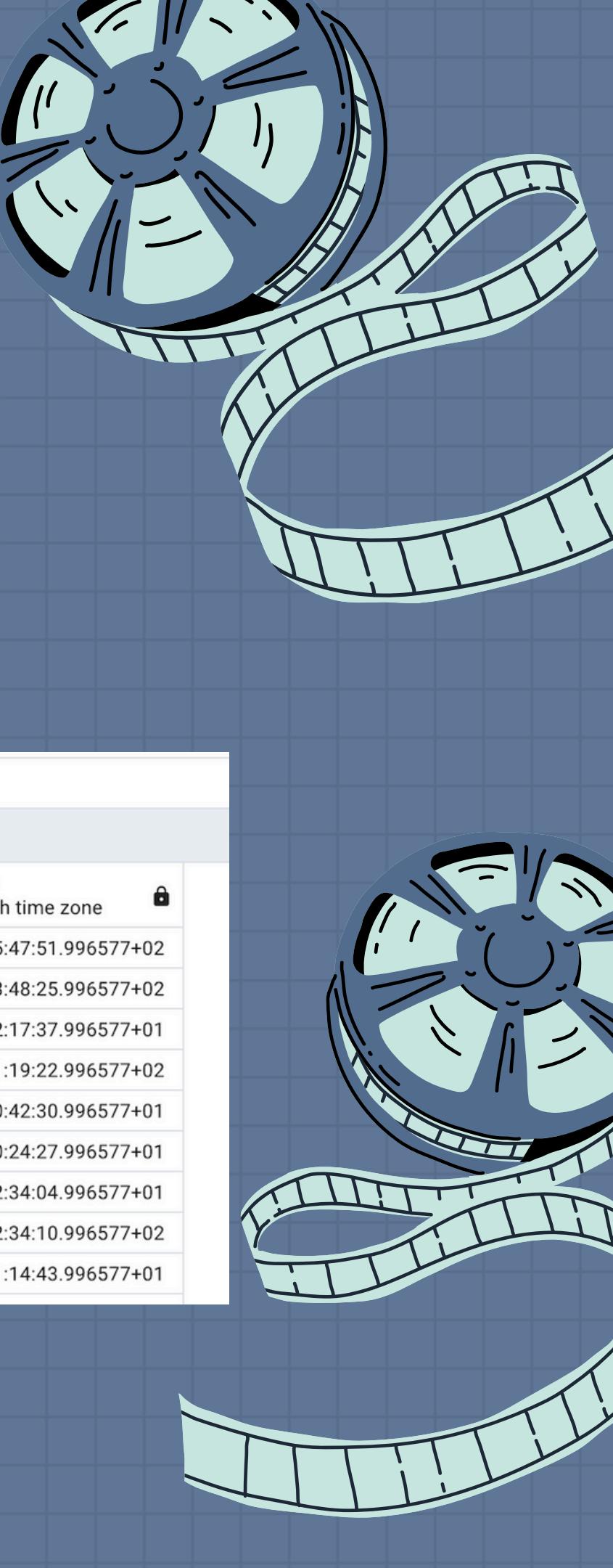
Additionally, the list should be ordered by customer number in ascending order, and then order it by amount in descending order.

```
4  
5  SELECT  
6    *  
7  FROM payment  
8 WHERE customer_id in (322, 346, 354)  
9 AND (amount < 2 OR amount > 10)  
10 ORDER BY customer_id, amount DESC
```

Data Output Messages Notifications

A screenshot of a database application interface showing a table of payment data. The table has columns: payment_id, customer_id, staff_id, rental_id, amount, and payment_date. The data shows various payments made by customers 322, 346, and 354, with amounts ranging from 0.99 to 1.99. The table is sorted by customer_id in ascending order and amount in descending order.

	payment_id integer	customer_id smallint	staff_id smallint	rental_id integer	amount numeric (5,2)	payment_date timestamp with time zone
1	25784	322	2	3627	1.99	2020-04-06 05:47:51.996577+02
2	25794	322	1	9252	1.99	2020-04-30 13:48:25.996577+02
3	16167	322	2	166	0.99	2020-01-26 02:17:37.996577+01
4	25788	322	2	6673	0.99	2020-04-12 11:19:22.996577+02
5	20023	322	2	11456	0.99	2020-03-02 20:42:30.996577+01
6	20027	322	1	15450	0.99	2020-03-23 00:24:27.996577+01
7	17420	322	1	2554	0.99	2020-02-19 02:34:04.996577+01
8	25783	322	1	3478	0.99	2020-04-05 22:34:10.996577+02
9	20248	346	2	10521	0.99	2020-03-01 11:14:43.996577+01



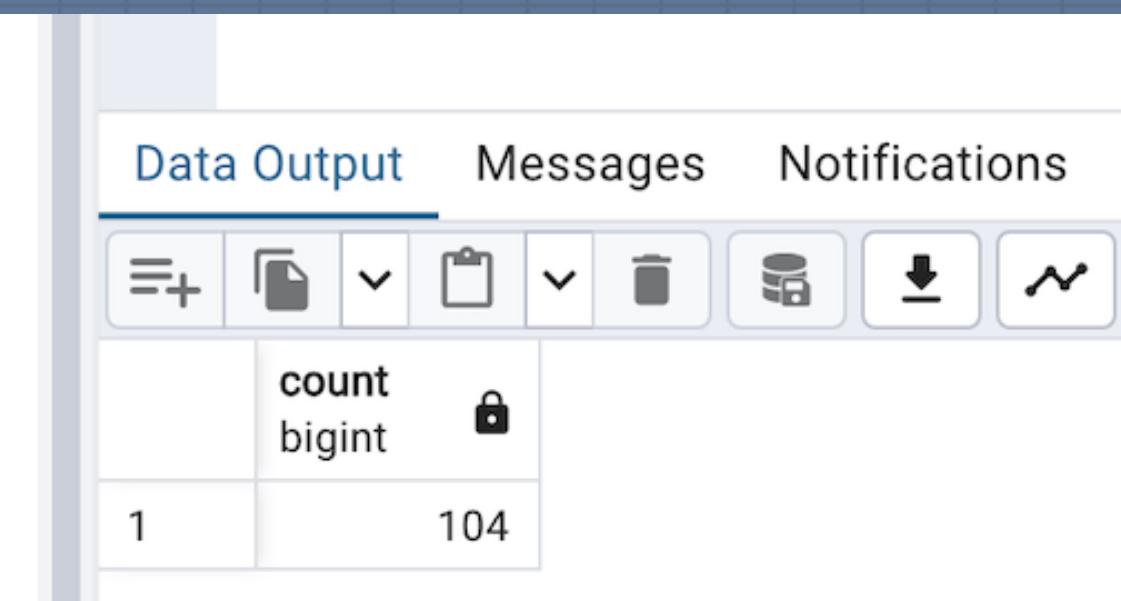
The manager has identified faulty payments and we now need to determine the number of affected transactions.

Specifically, we need to find out how many payments were made on January 26 and 27, 2020,

with amounts ranging from \$1.99 to \$3.99.

```
4  
5 SELECT  
6 COUNT(*)  
7 FROM payment  
8 WHERE (payment_date BETWEEN '2020-01-26' AND '2020-01-28')  
9 AND (amount BETWEEN 1.99 AND 3.99)
```

Data Output Messages Notifications



	count bigint
1	104

The support manager has approached you again, informing you of six complaints from customers regarding their payments. You now need to investigate these payments for the customers with the following IDs: 12, 25, 67, 93, 124, and 234. Specifically, you need to look into payments of these customers with amounts of \$4.99, \$7.99, and \$9.99, all occurring in January 2020.

```
5  
6 SELECT  
7 *  
8 FROM payment  
9 WHERE (customer_id IN (12,25,67,93,124,234))  
10 AND amount IN (4.99, 7.99, 9.99)  
11 AND payment_date BETWEEN '2020-01-01' AND '2020-02-01'  
12
```

Data Output							
	payment_id	customer_id	staff_id	rental_id	amount	payment_date	
1	16699	12	1	988	4.99	2020-01-30 22:36:29.996577+01	
2	16700	12	1	1084	4.99	2020-01-31 10:38:43.996577+01	
3	16736	25	1	90	7.99	2020-01-25 13:59:51.996577+01	
4	16822	67	2	331	9.99	2020-01-27 01:50:52.996577+01	
5	16870	93	1	1025	4.99	2020-01-31 03:10:03.996577+01	
6	16929	124	2	1039	4.99	2020-01-31 05:00:55.996577+01	
7	17116	234	2	1125	4.99	2020-01-31 16:52:10.996577+01	

The inventory manager requires assistance once more; this time, we need to determine the number of movies that have the word 'documentary' in their description.

```
3  
4 SELECT  
5 COUNT(*)  
6 FROM film  
7 WHERE description LIKE '%Documentary'
```

Data Output		Messages	Notifications
	count bigint		🔒
1	101		



You need to determine the number of customers in the database whose first name is three letters long and whose last name ends with either an X or a Y.

```
4 SELECT  
5 COUNT(*)  
6 FROM customer  
7 WHERE first_name LIKE '___'  
8 AND (last_name LIKE '%X'  
9 OR last_name LIKE '%Y')
```

Data Output		Messages	Notifications
	count bigint		🔒
1	3		



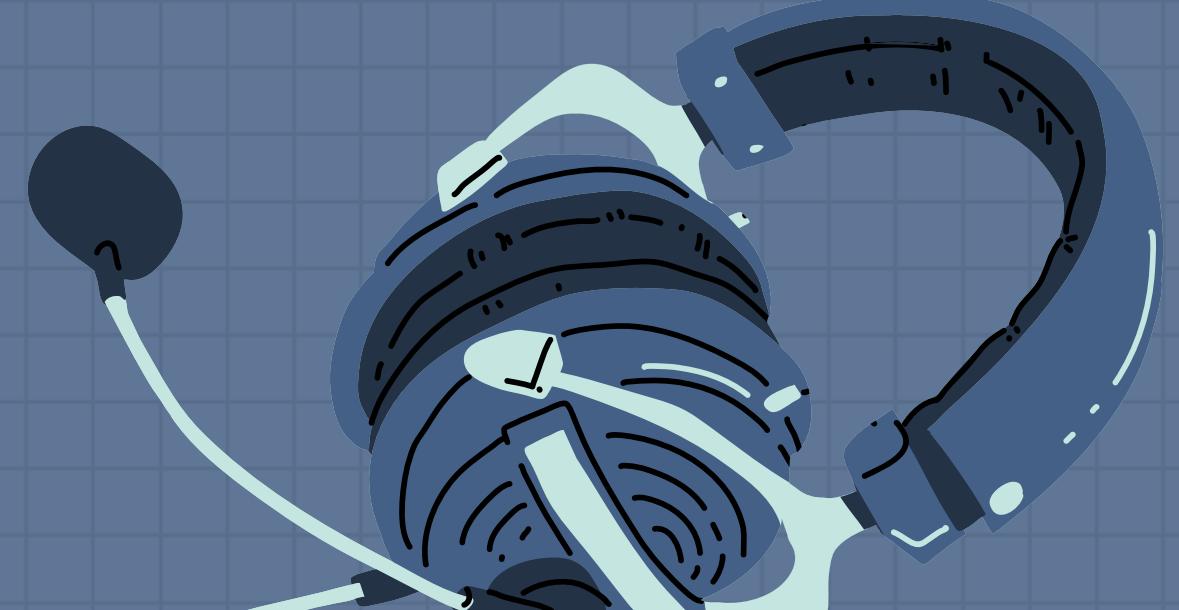
The manager now wants to gather more information about the films in THE FILM CORNER store.

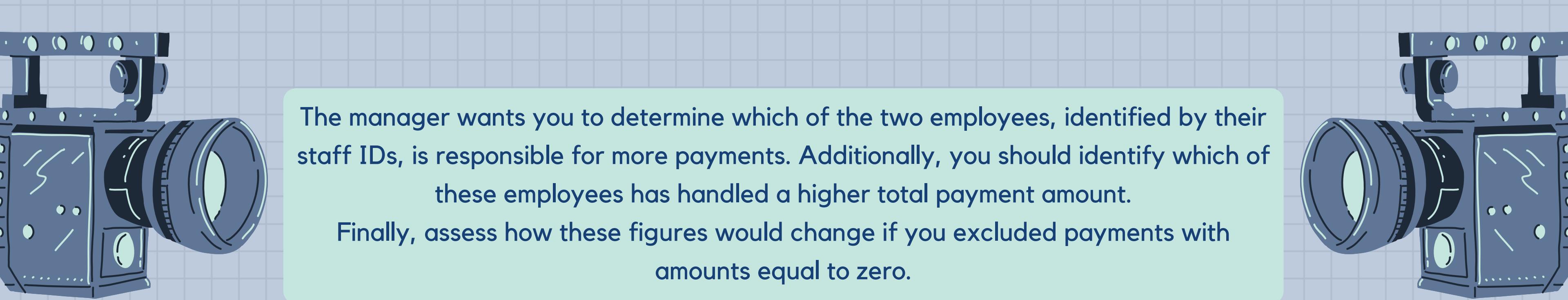
You are requested to write a query to determine the replacement costs of these films. Specifically, you should find the minimum, maximum, and average (rounded to two decimal places) replacement costs, as well as the total sum.

```
SELECT  
MIN(replacement_cost) AS Minimum ,  
MAX(replacement_cost) AS Maximum ,  
ROUND(AVG(replacement_cost), 2 ) AS Average,  
SUM(replacement_cost) AS Total  
FROM film
```

Data Output Messages Notifications

	minimum numeric	maximum numeric	average numeric	total numeric
1	9.99	29.99	19.98	19984.00





The manager wants you to determine which of the two employees, identified by their staff IDs, is responsible for more payments. Additionally, you should identify which of these employees has handled a higher total payment amount.

Finally, assess how these figures would change if you excluded payments with amounts equal to zero.

```
5  SELECT
6    staff_id,
7    SUM(amount),
8    COUNT(*)
9  FROM payment
10 WHERE amount != 0
11 GROUP BY staff_id
12 ORDER BY 2 DESC
13
```

Data Output Messages Notifications

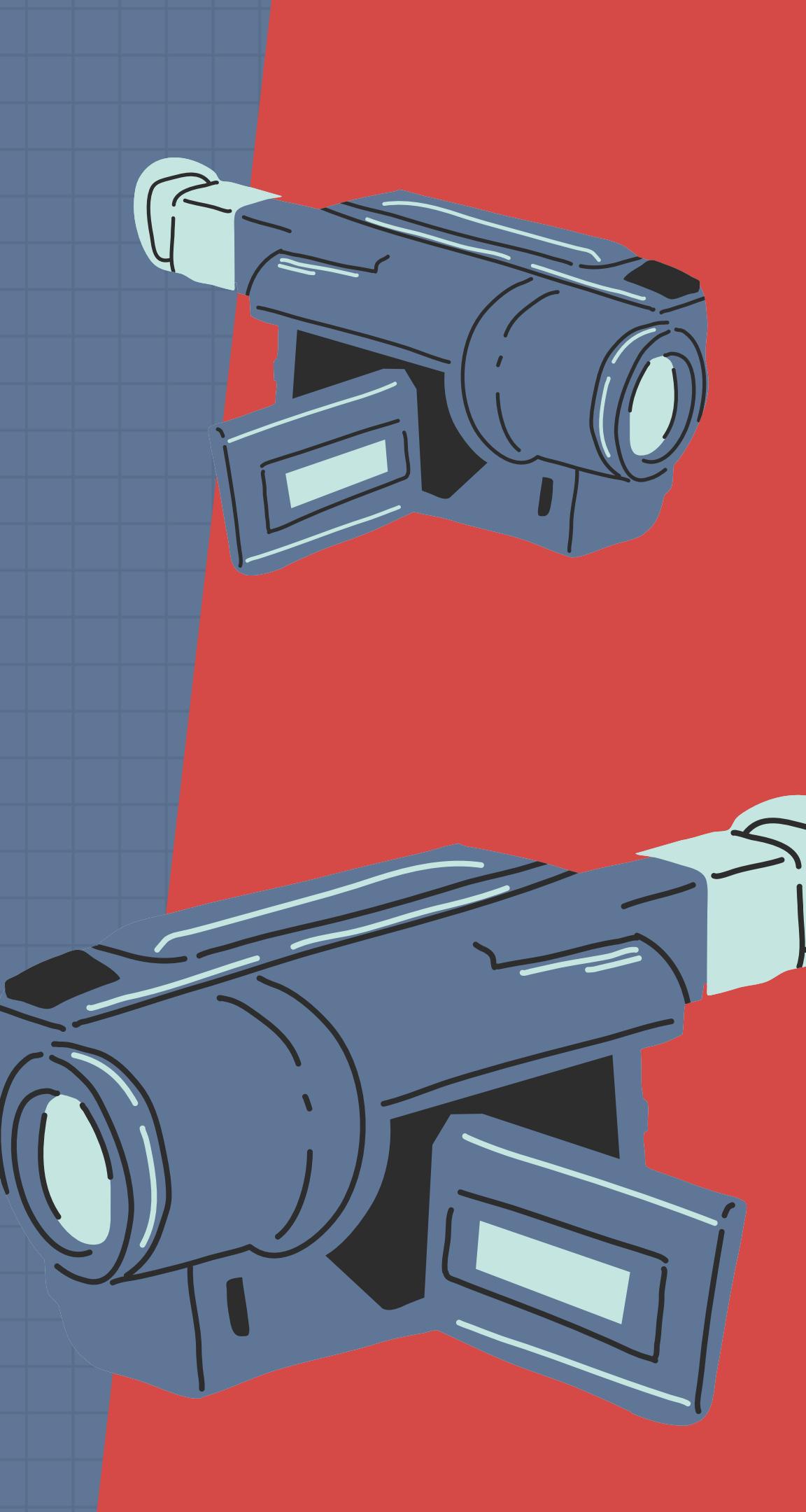
	staff_id	sum	count
1	2	33927.04	7983
2	1	33489.47	8042

"We need to determine which of the two employees achieved the highest sales amount in a single day. Additionally, we need to find out which of these employees recorded the most sales transactions in a single day. Note that payments with amounts equal to zero should not be included in this analysis."

```
5  SELECT
6    DATE(payment_date),
7    staff_id,
8    SUM(amount),
9    COUNT(*)
10   FROM payment
11  WHERE amount != 0
12  GROUP BY staff_id , DATE(payment_date)
13  ORDER BY SUM(amount) DESC
```

Data Output Messages Notifications

	date date	staff_id smallint	sum numeric	count bigint
1	2020-04-30	2	2866.42	658
2	2020-04-30	1	2736.75	625
3	2020-03-21	2	1505.52	348
4	2020-03-20	2	1455.76	324
5	2020-03-01	1	1433.58	342



The manager has identified April 28, 29, and 30, 2020, as days with exceptionally high revenue and wants to focus exclusively on these dates. The task is to calculate the average payment amount, grouping by customer and date. Additionally, restrict the results to customers and dates with more than one payment each, and sort the results in descending order of the average amount."

```
SELECT
customer_id,
DATE(payment_date),
ROUND(AVG(amount), 2) AS Avg_amount,
COUNT(*)
FROM payment
WHERE payment_date BETWEEN '2020-04-28' AND '2020-04-30 11:59'
GROUP BY customer_id, DATE(payment_date)
HAVING COUNT(*) > 1
ORDER BY ROUND(AVG(amount), 2) DESC
```

Data Output Messages Notifications

	customer_id smallint	date date	avg_amount numeric	count bigint
1	459	2020-04-29	10.49	2
2	510	2020-04-28	9.49	2
3	443	2020-04-28	9.49	2
4	245	2020-04-29	8.99	2
5	181	2020-04-29	7.99	2
6	573	2020-04-30	7.99	2

Project Summary

In this project, we tackled various queries to meet the needs of different departments within the company. Here's are some tasks that we tackled during the projects :

Customer List, Sorted Customers, Payment Prices

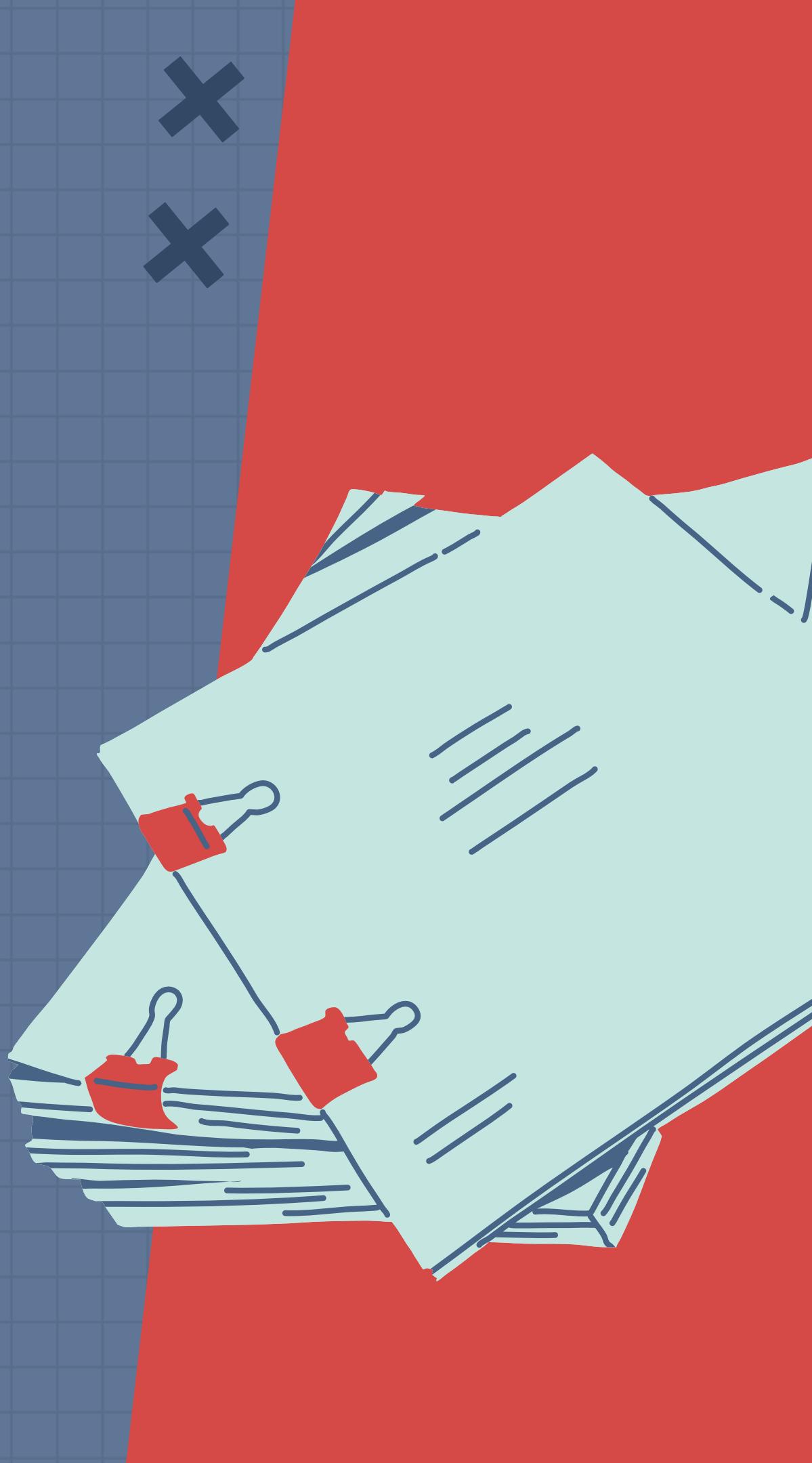
Film Count, Latest Rental Date, Found the most recent rental date.

Unreturned Rentals & Low Payments, Specific Customer Payments, Faulty Transactions, Complaint Analysis, Customer Name Criteria

Film Costs, Employee Payments, Employee Sales, High Revenue Days

We utilized basic SQL queries, including 'SELECT', 'ORDER BY', 'LIMIT', 'DISTINCT', and filtering functions like 'WHERE' and 'HAVING'. Aggregated functions such as 'SUM', 'AVG', 'MAX', and 'MIN', along with 'GROUP BY', were employed to generate the required insights and facilitate effective data management and analysis.

This approach ensured that we provided accurate and actionable data to support the company's various operational needs.





Thank you..

