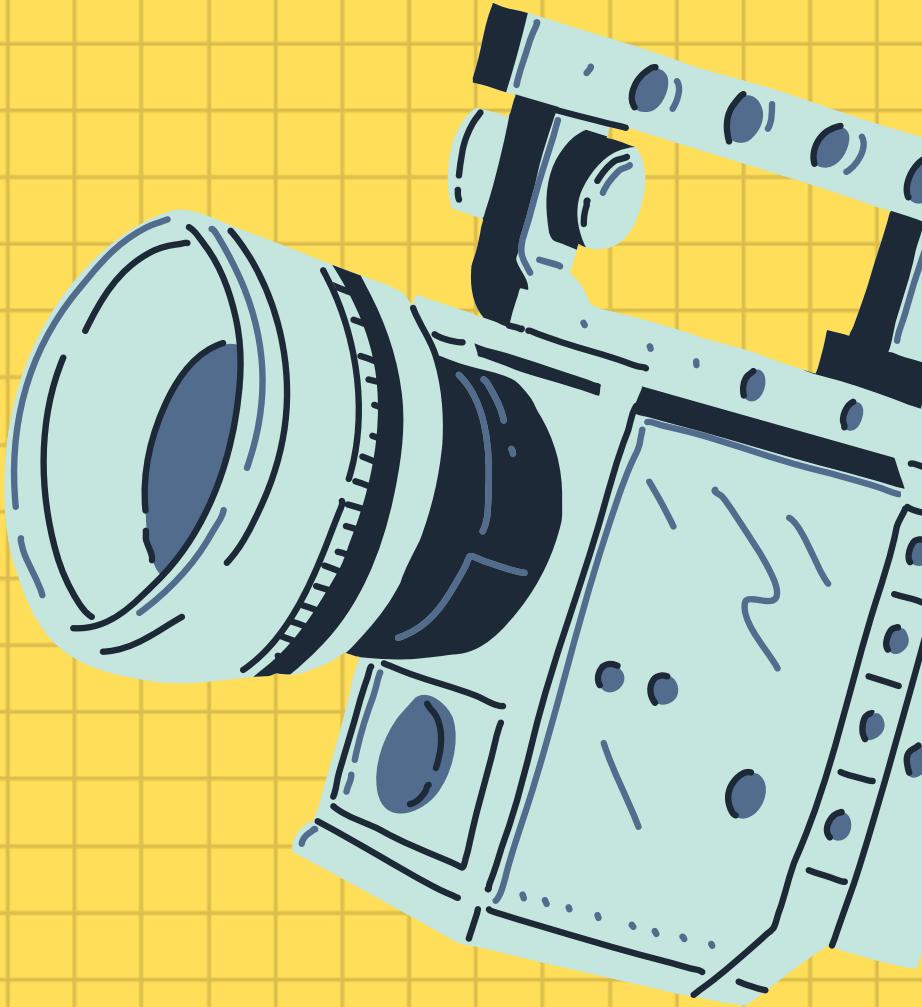
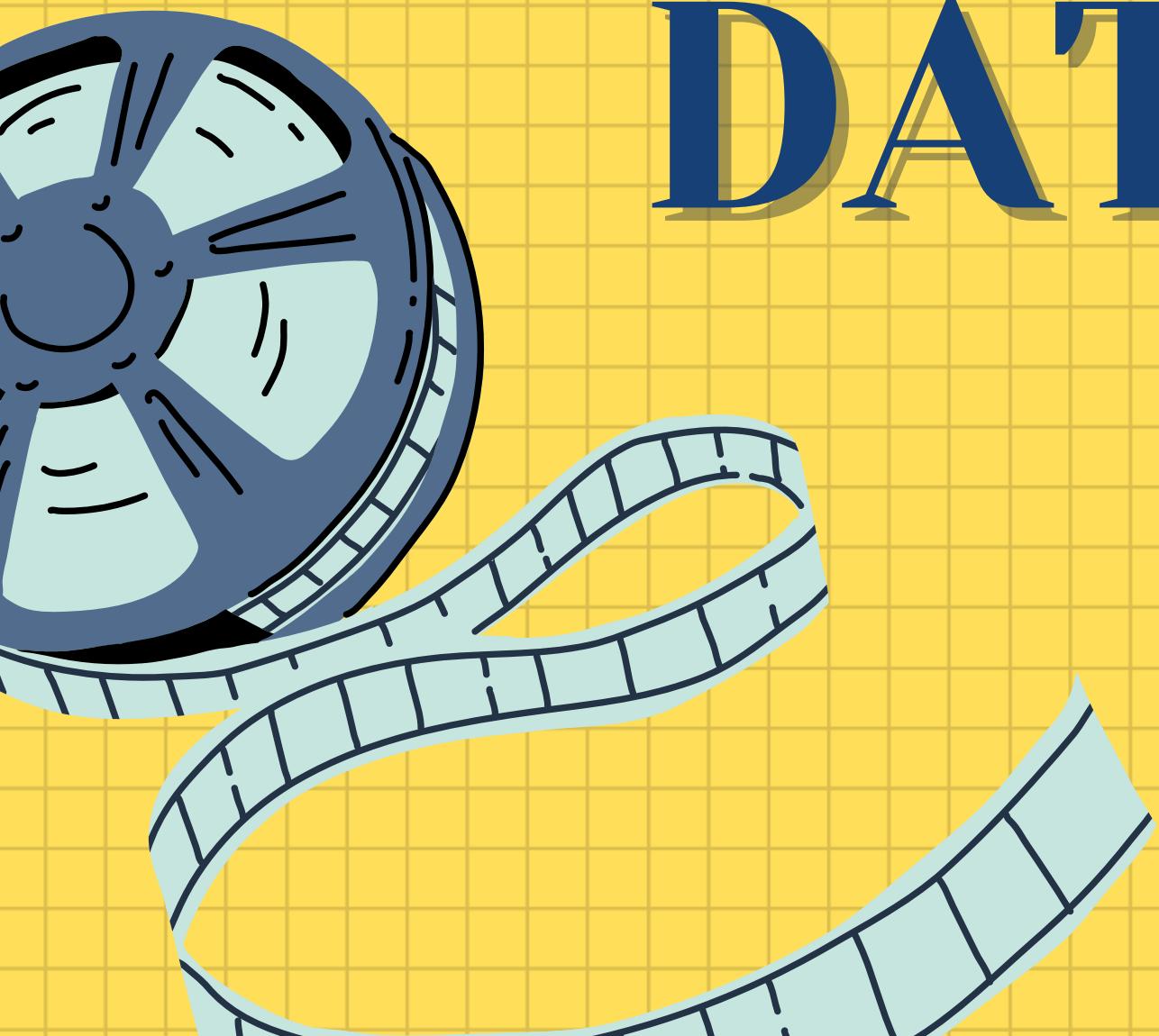
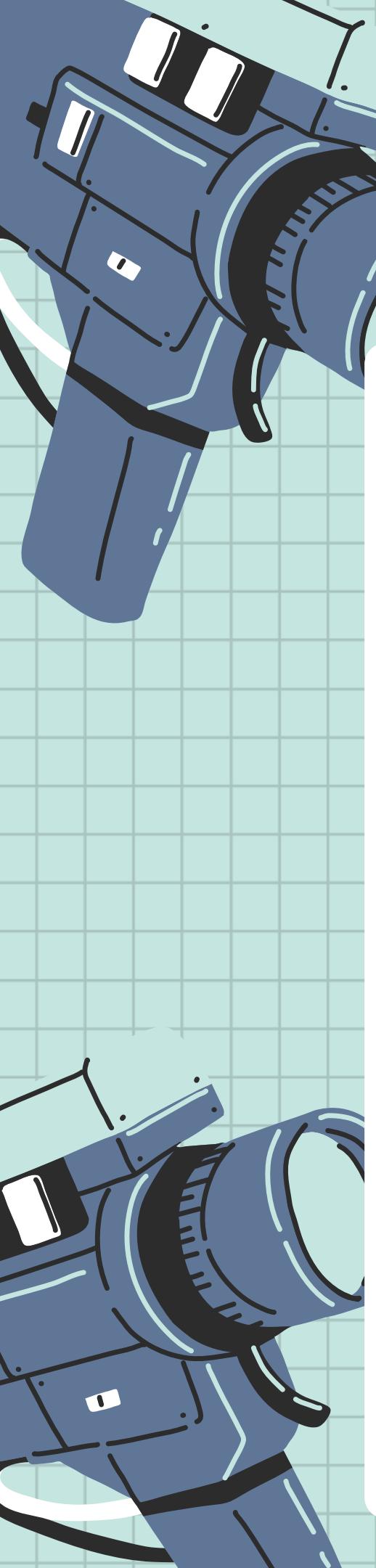


SAKILA MOVIE

DATA SET

MySQL - PROJECT





Learning Goals

1

To develop a foundational understanding and proficiency in constructing and executing basic MySQL queries using clauses **SELECT, FROM, DISTINCT, WHERE, AND, and OR.**

2

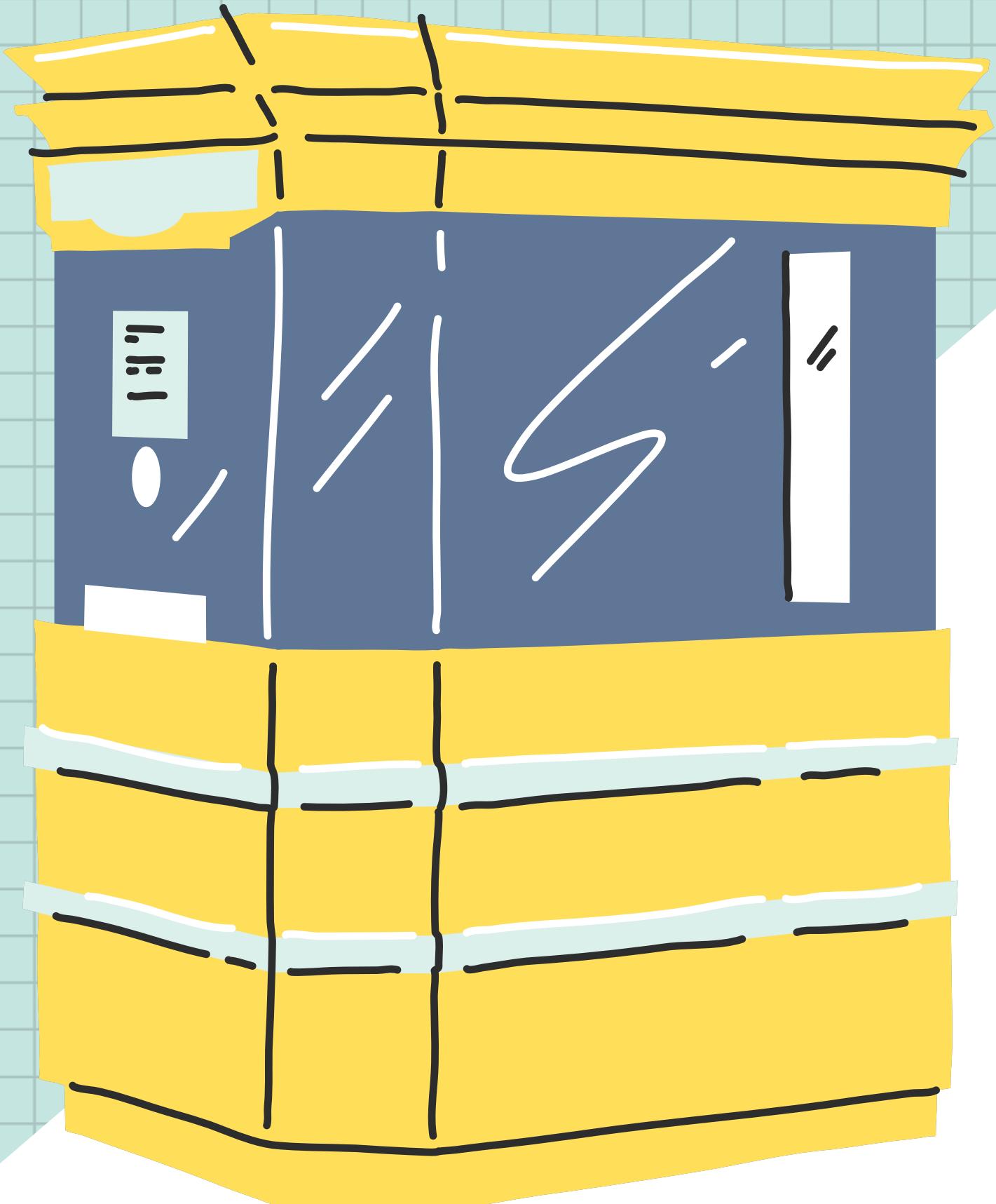
To gain proficiency in constructing and executing MySQL queries involving joins to combine data from multiple tables and understand relationships between tables.

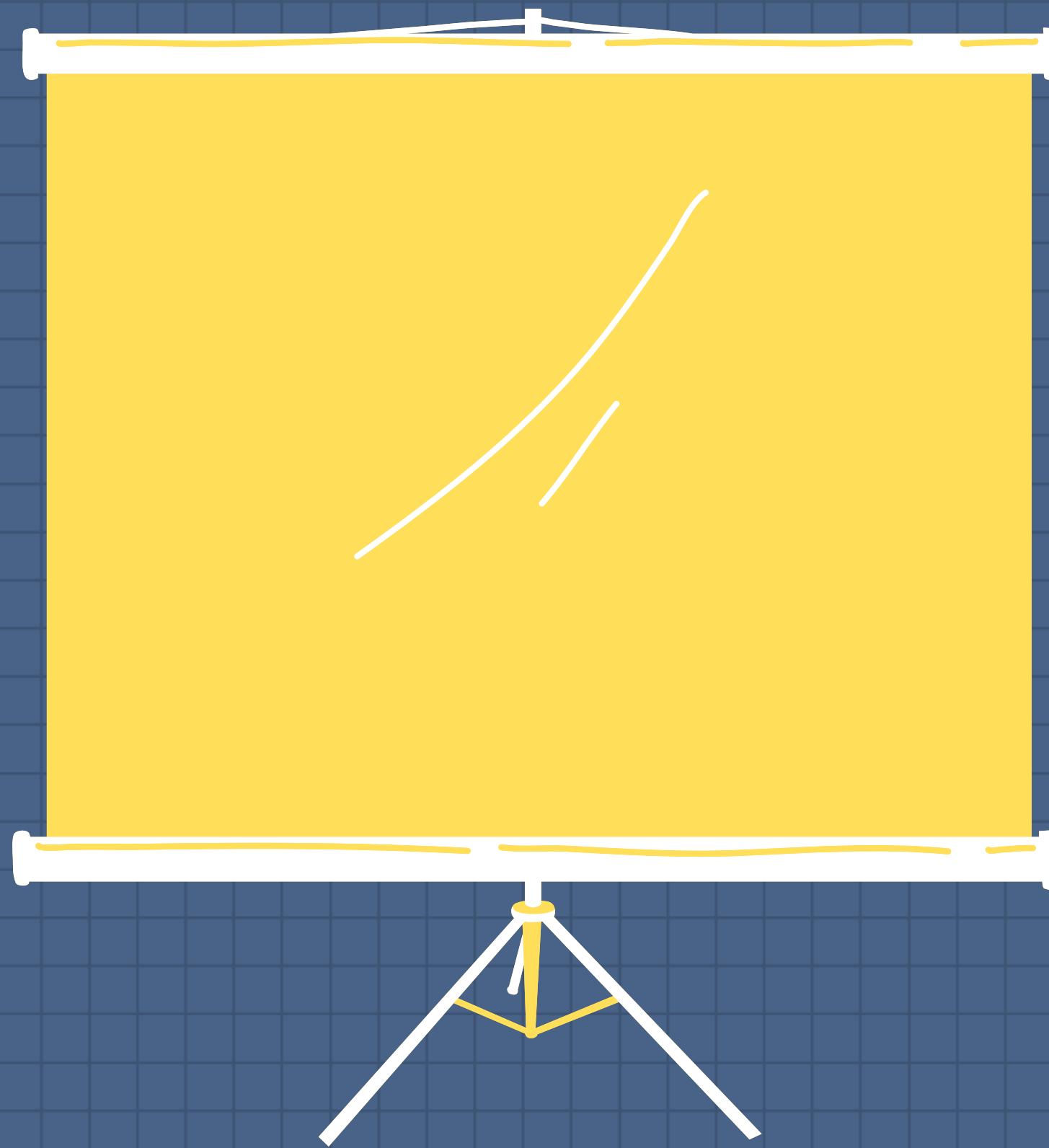
3

To develop advanced SQL skills by constructing and executing MySQL queries using subqueries, aggregate functions, and window functions to perform complex data retrieval and analysis.

Contents

- 1 SCHEMA DIAGRAM
- 2 BASIC QUERIES
- 3 JOINS, RELATIONSHIPS, SUB QUERIES, AGGREGATION
- 4 ADVANCED QUERIES

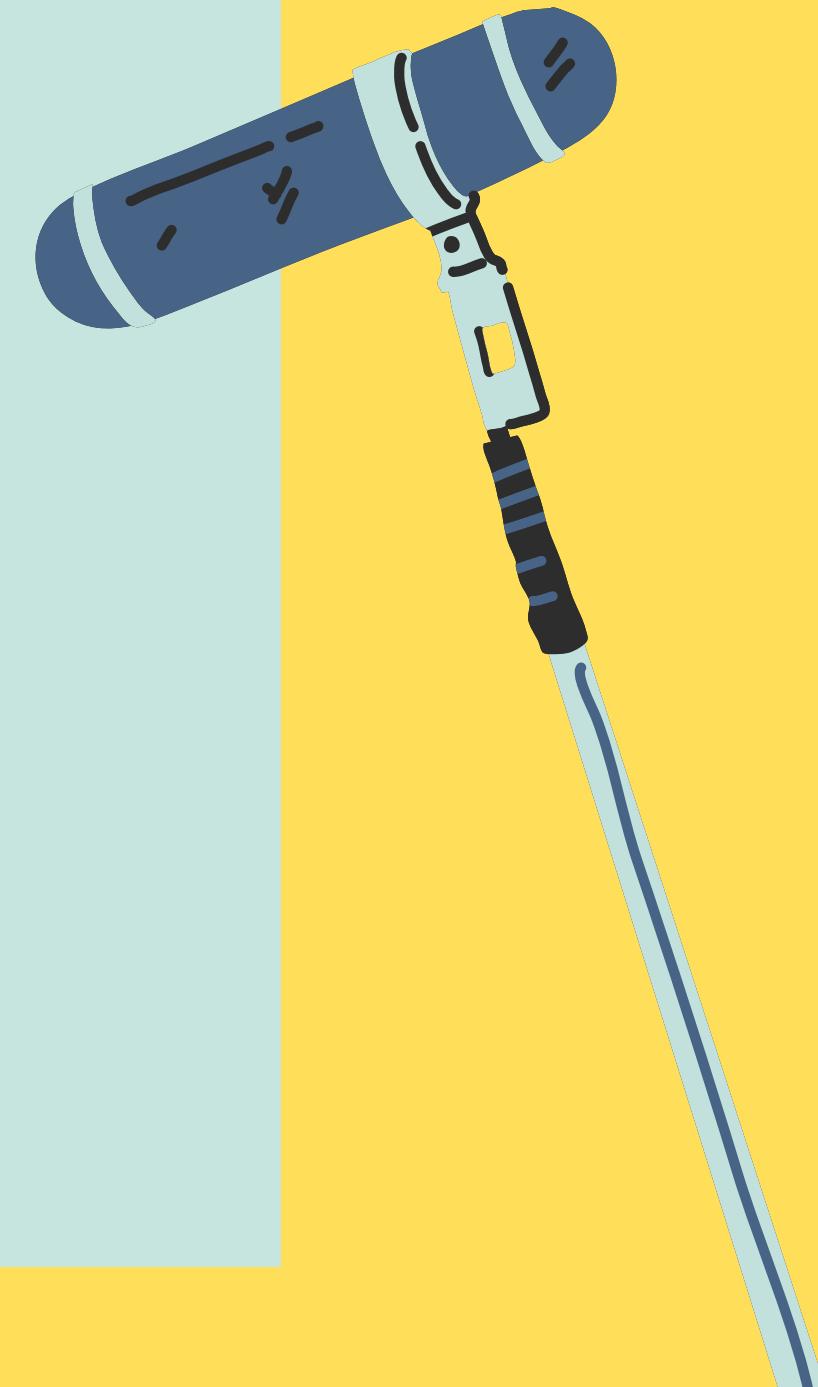
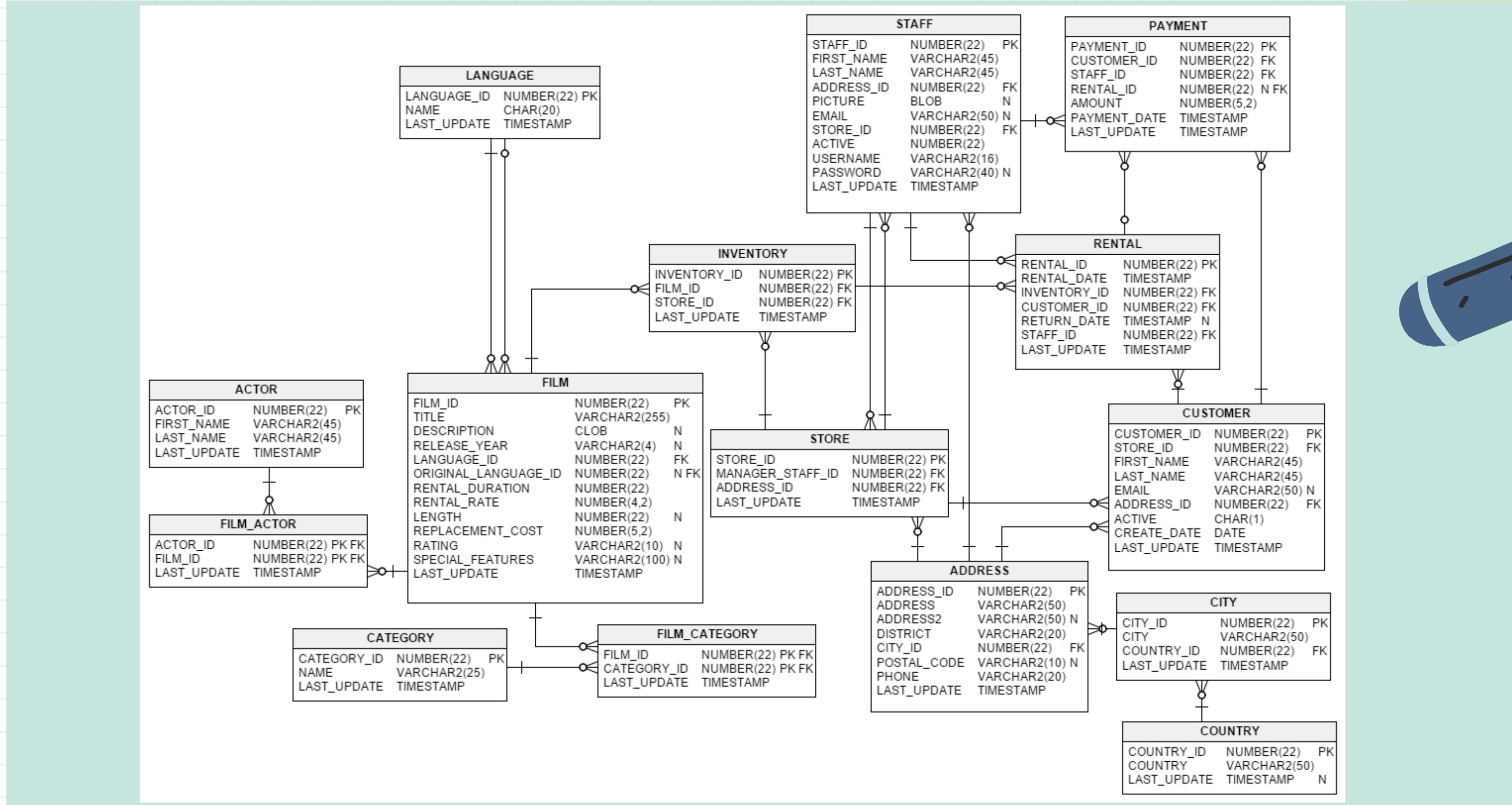


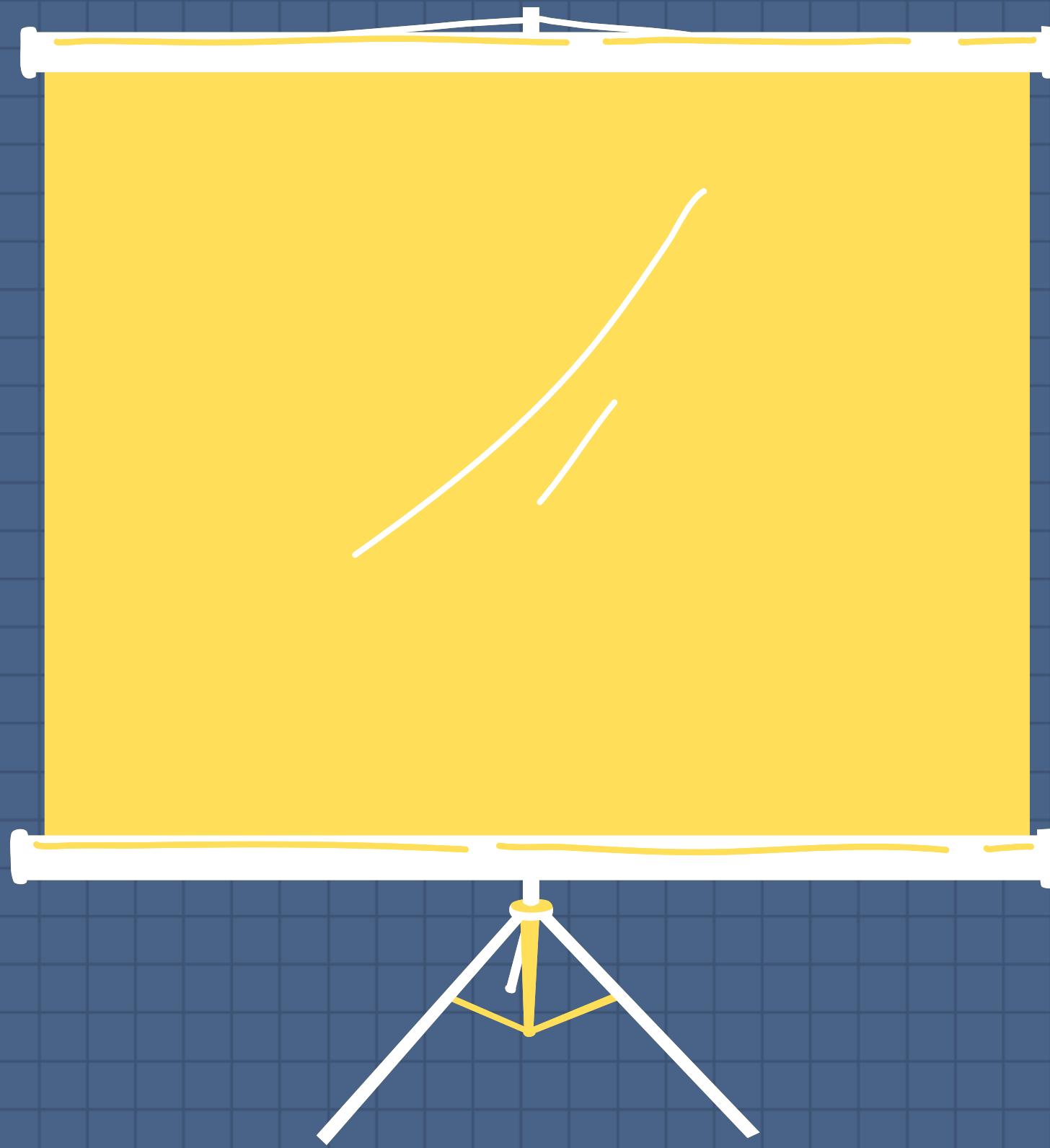


SCHEMA DIAGRAM



Represents Relationship Between all Tables





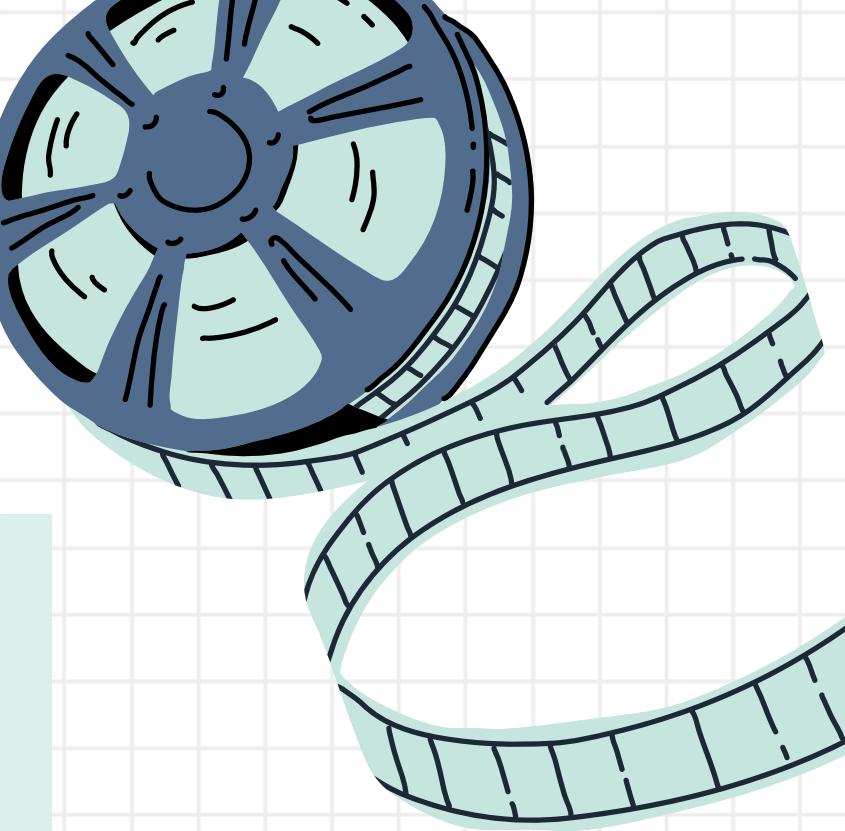
BASIC QUERIES



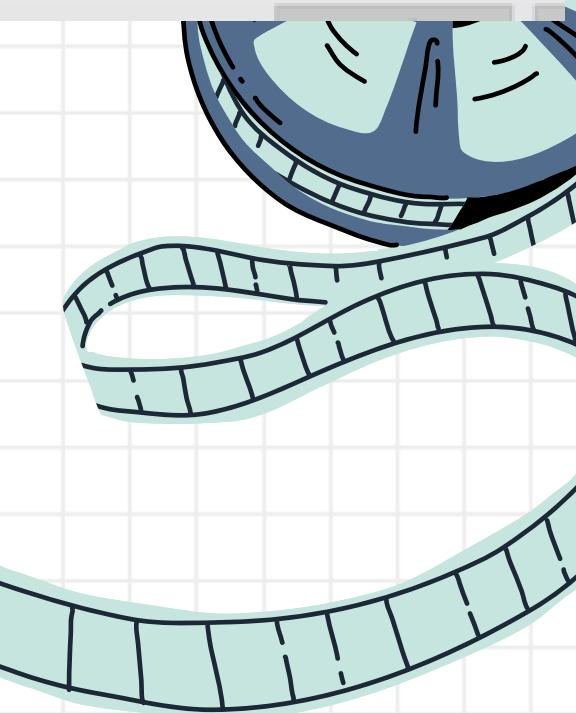
1. Retrieve all columns from the film table.

```
1 •      SELECT  
2          *  
3      FROM  
4      films  
5
```

RESULT



	film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost
▶	1	ACADEMY DINOSAUR	A Epic Drama of a Feminist And a Mad Scientist ...	2006	1	NULL	6	0.99	86	20.99
	2	ACE GOLDFINGER	A Astounding Epistle of a Database Administrat...	2006	1	NULL	3	4.99	48	12.99
	3	ADAPTATION HOLES	A Astounding Reflection of a Lumberjack And a ...	2006	1	NULL	7	2.99	50	18.99
	4	AFFAIR PREJUDICE	A Fanciful Documentary of a Frisbee And a Lum...	2006	1	NULL	5	2.99	117	26.99
	5	AFRICAN EGG	A Fast-Paced Documentary of a Pastry Chef An...	2006	1	NULL	6	2.99	130	22.99



2. Select distinct categories from the category table.

1 ●

SELECT DISTINCT

2

(NAME)

3

FROM

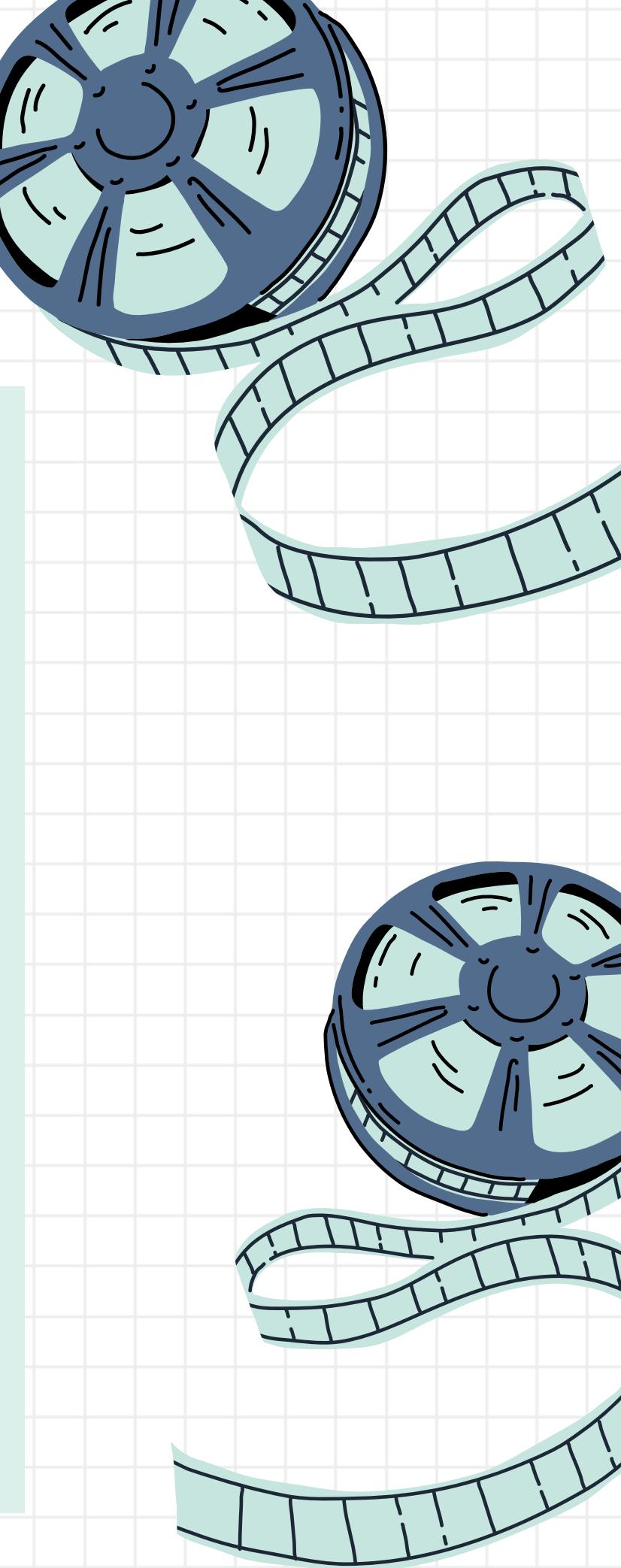
4

category;

X
X

RESULT

	NAME
▶	Action
	Animation
	Children
	Classics
	Comedy
	Documentary

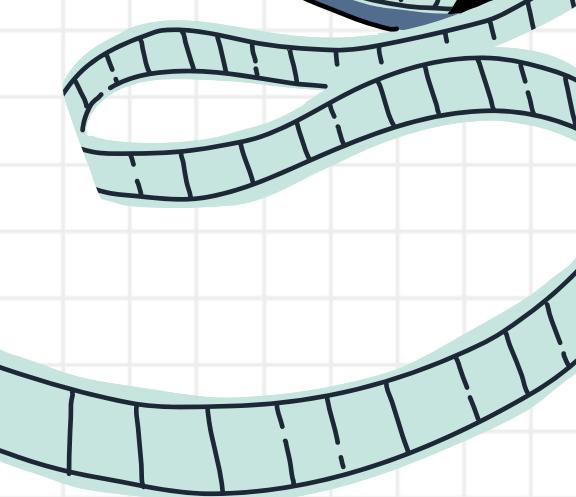
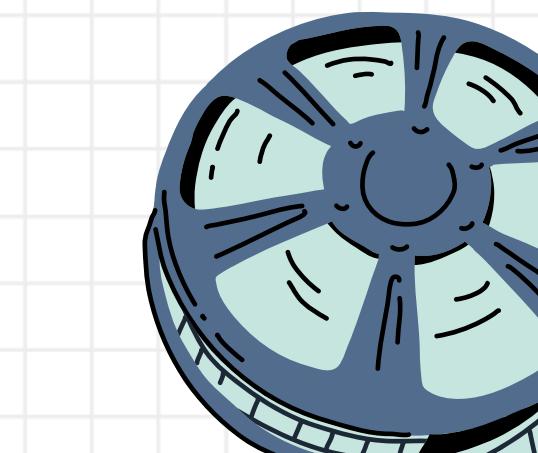
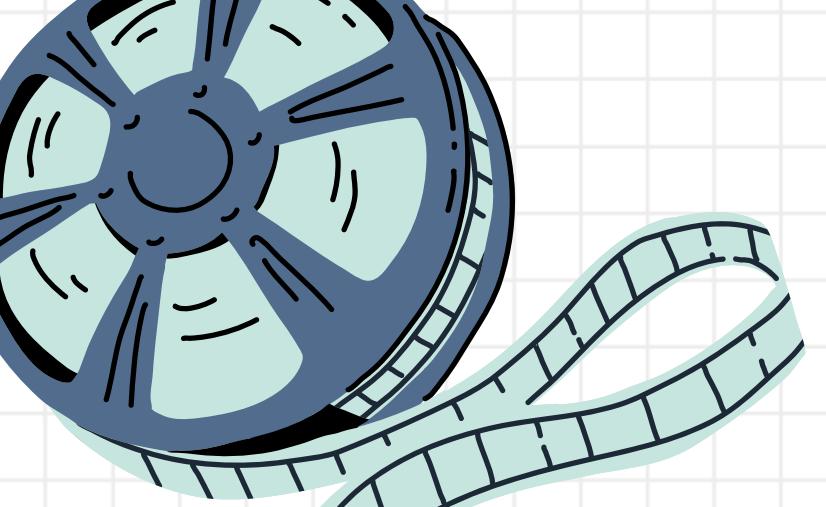


3.Get the first 10 rows from the actor table.

```
1 •      SELECT
2
3      FROM
4          actor
5      LIMIT 10;
```

RESULT

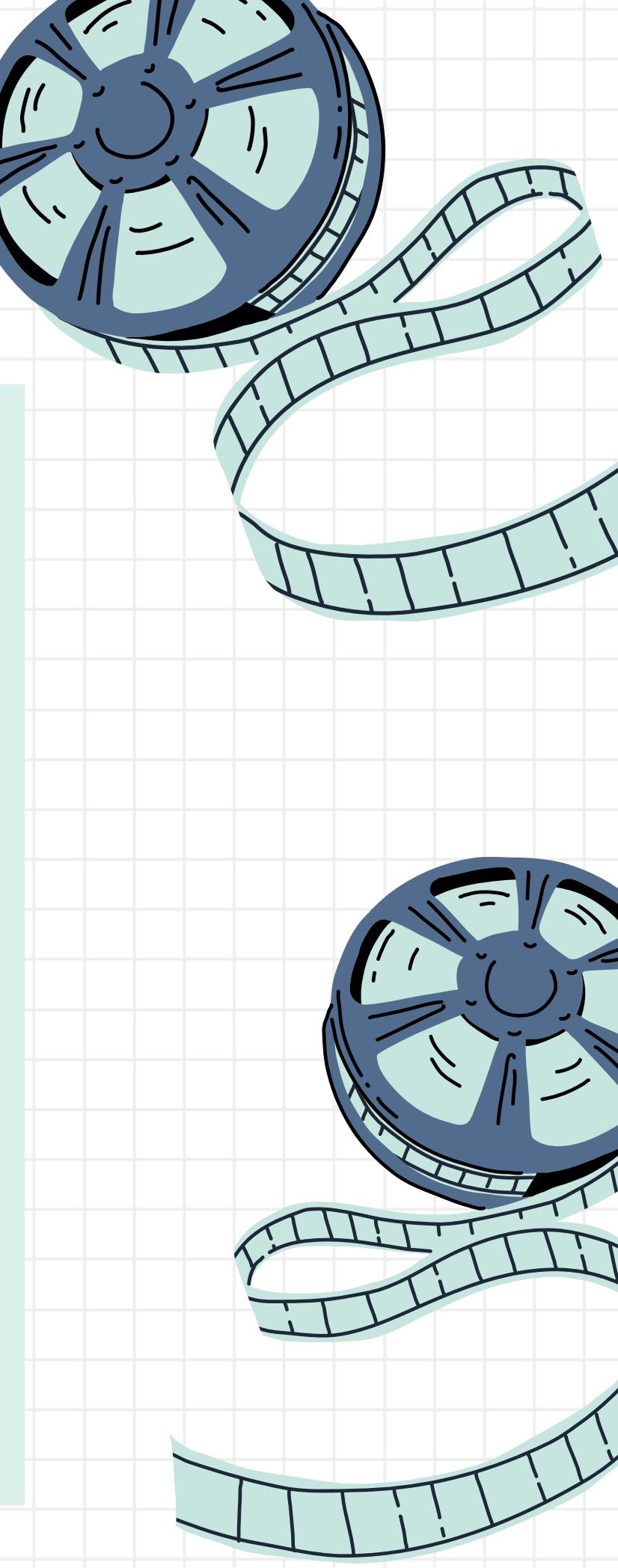
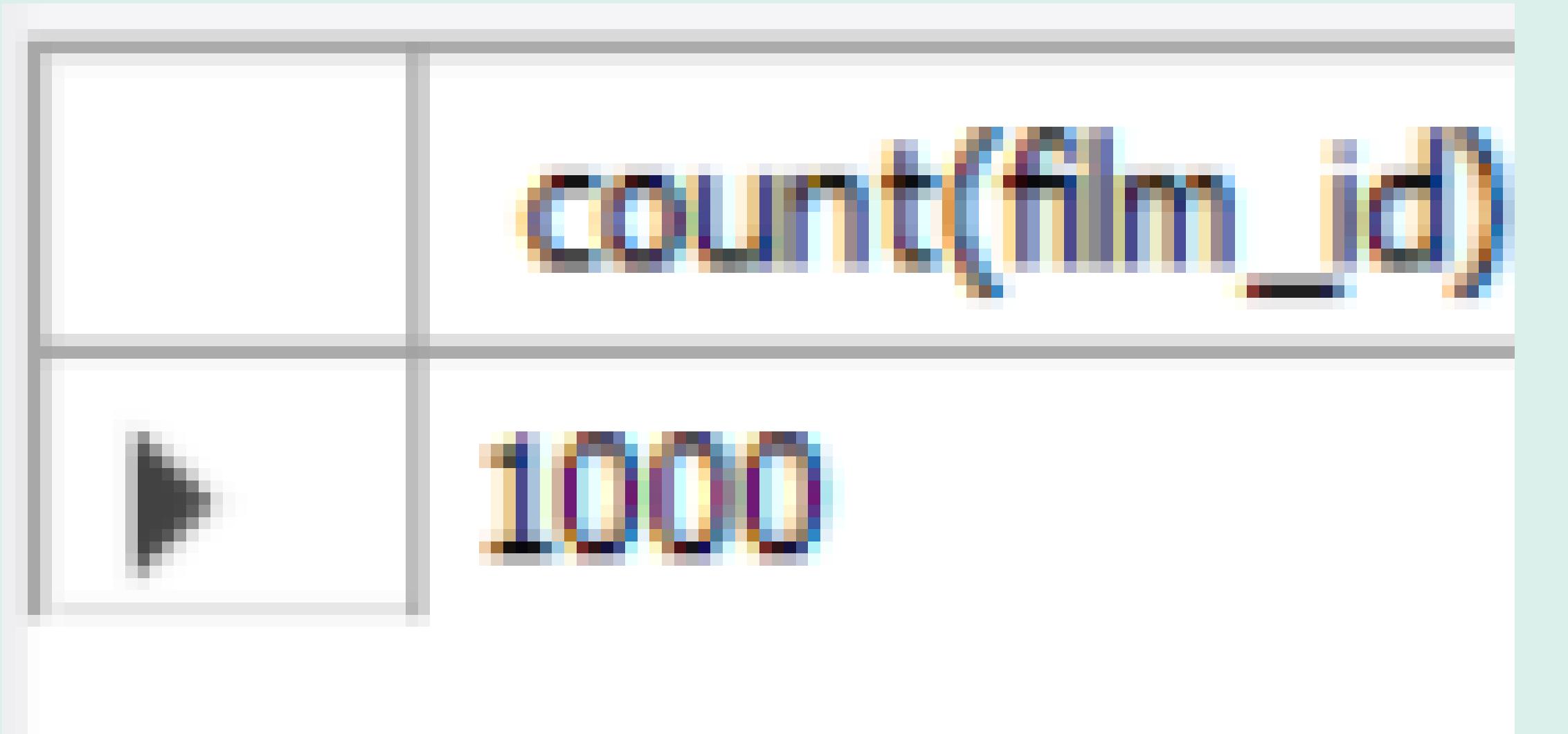
	actor_id	first_name	last_name	last_update
▶	1	PENELOPE	GUINNESS	2006-02-15 04:34:33
	2	NICK	WAHLBERG	2006-02-15 04:34:33
	3	ED	CHASE	2006-02-15 04:34:33
	4	JENNIFER	DAVIS	2006-02-15 04:34:33
	5	JOHNNY	LOLLOBRIGIDA	2006-02-15 04:34:33
	6	BETTE	NICHOLSON	2006-02-15 04:34:33



4.Find the total number of films in the film table.

```
1   SELECT  
2   COUNT(film_id)  
3   FROM  
4   film
```

RESULT

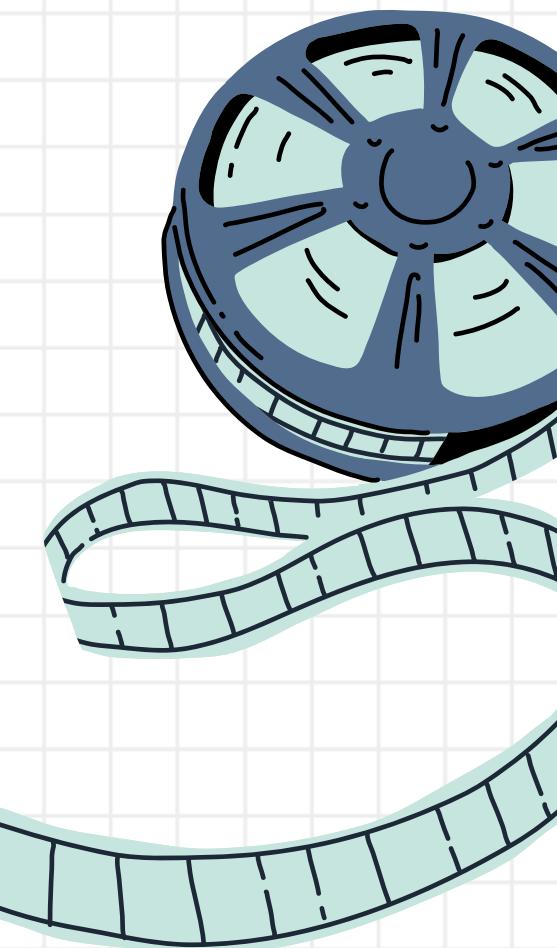
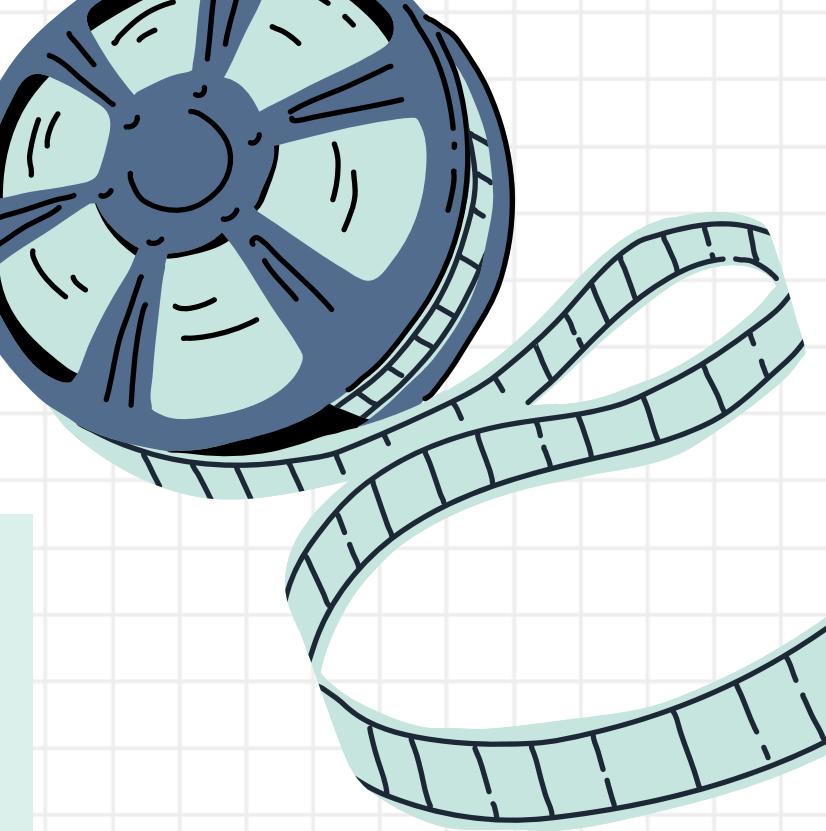


5.List all films released after the year 2005.

```
1 *   SELECT  
2           TITLE, RELEASE_YEAR  
3 FROM  
4     FILM  
5 WHERE  
6   RELEASE_YEAR > 2005;
```

RESULT

	TITLE	RELEASE_YEAR
▶	ACADEMY DINOSAUR	2006
	ACE GOLDFINGER	2006
	ADAPTATION HOLES	2006
	AFFAIR PREJUDICE	2006
	AFRICAN EGG	2006
	AGENT TRUMAN	2006

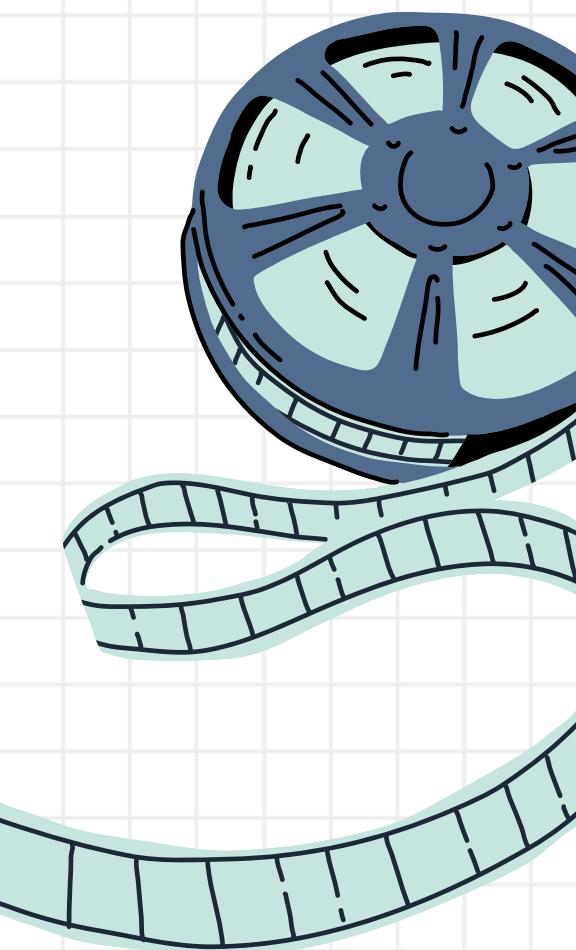
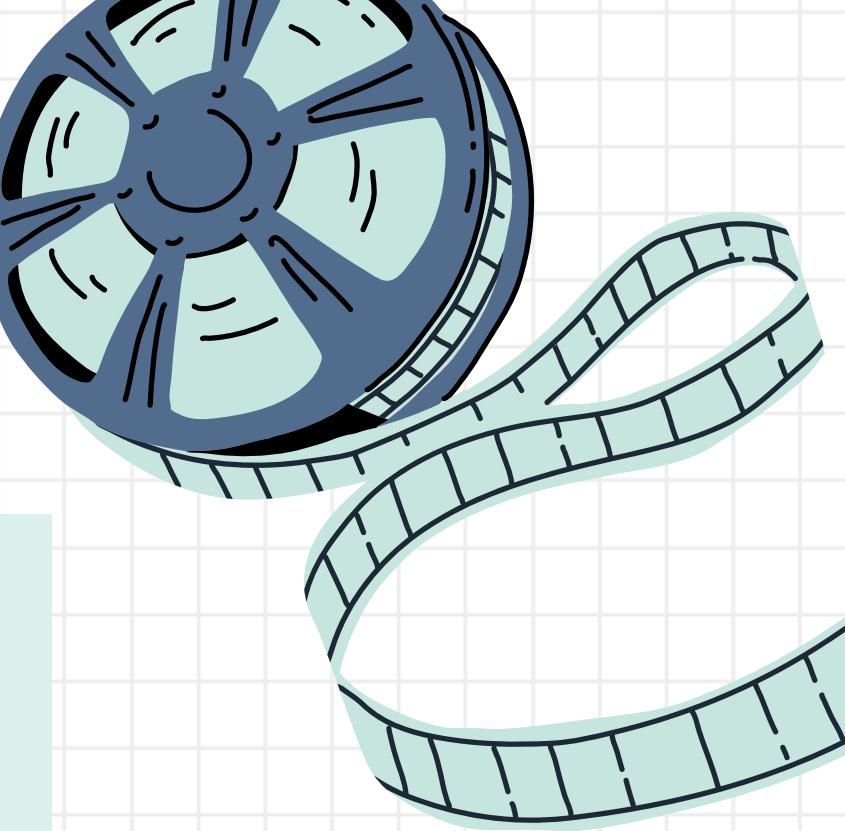


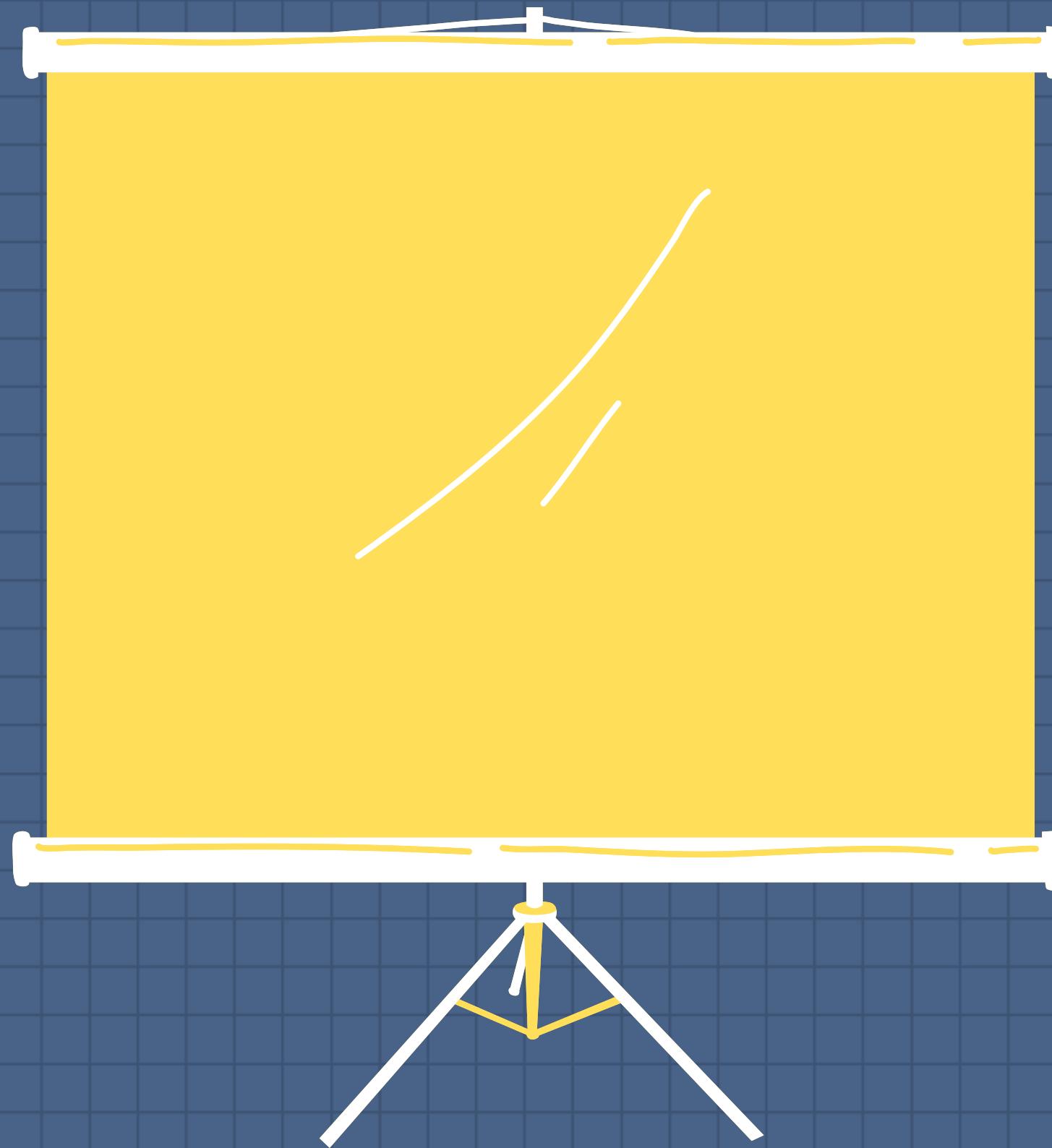
6.Retrieve the titles and descriptions of films where the title contains "war."

```
1 •      SELECT  
2          TITLE, DESCRIPTION  
3      FROM  
4          FILM  
5      WHERE  
6          TITLE LIKE '%WAR';
```

RESULT

TITLE	DESCRIPTION
DARKNESS WAR	A Touching Documentary of a Husband And a H...
ROCKY WAR	A Fast-Paced Display of a Squirrel And a Explor...





QUERIES USING JOINS AND RELATIONS

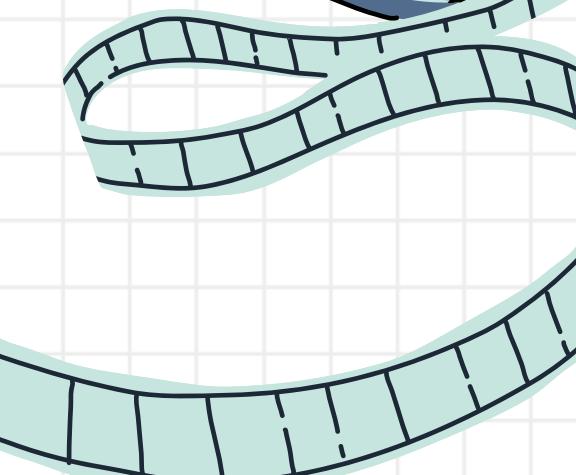
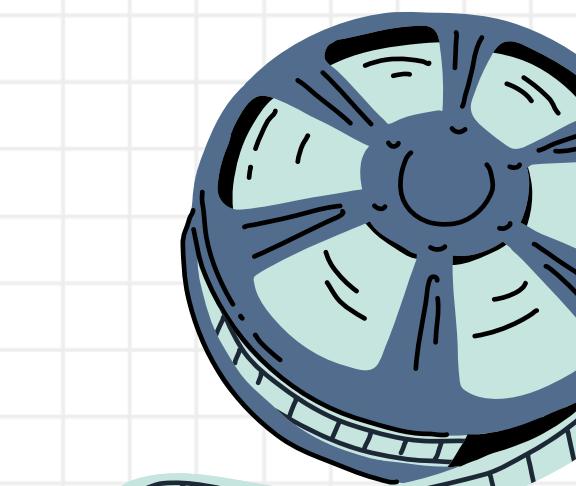
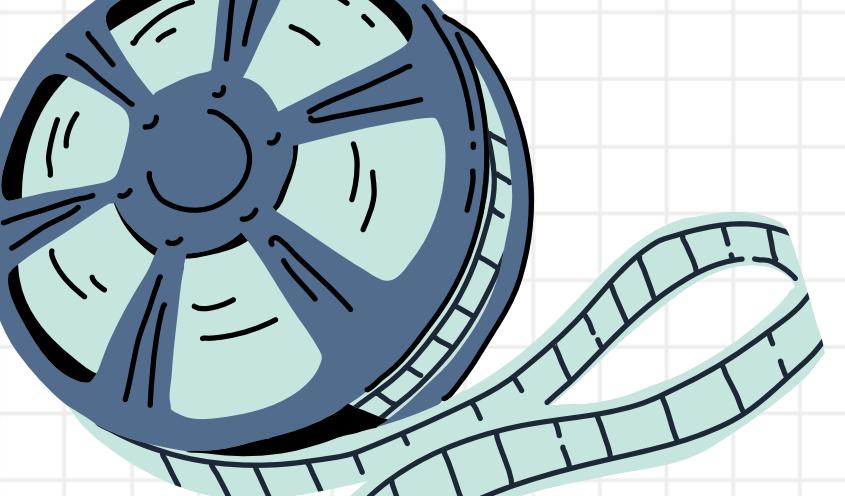


7.Find the first and last name of the actor who played the most roles.

```
1  SELECT
2      ACTOR.FIRST_NAME, ACTOR.LAST_NAME, K.NO_OF_FILMS
3  FROM
4      ACTOR
5      JOIN
6      (SELECT
7          ACTOR_ID, COUNT(FILM_ID) AS NO_OF_FILMS
8      FROM
9          FILM_ACTOR
10     GROUP BY ACTOR_ID) AS K ON ACTOR.ACTOR_ID = K.ACTOR_ID
11     ORDER BY NO_OF_FILMS DESC
12     LIMIT 1;
```

RESULT

	FIRST_NAME	LAST_NAME	NO_OF_FILMS
▶	GINA	DEGENERES	42

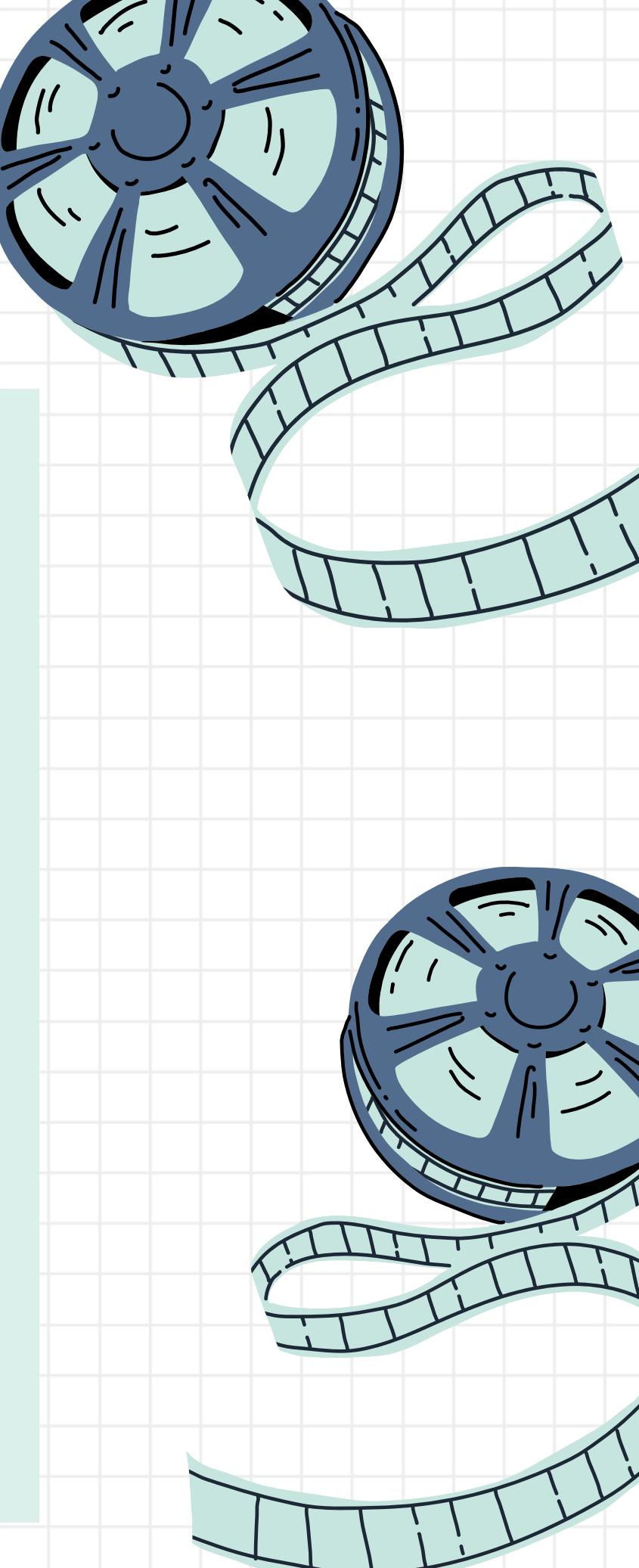


8. Get the names and email addresses of customers who have rented more than 30 films.

```
1 •   SELECT
2       customer.FIRST_NAME, customer.EMAIL, K.NUM_FILMS
3   FROM
4       customer
5       JOIN
6       (SELECT
7           CUSTOMER_ID, COUNT(inventory_id) AS NUM_FILMS
8   FROM
9       rental
10      GROUP BY CUSTOMER_ID
11      HAVING NUM_FILMS > 30) AS K ON CUSTOMER.CUSTOMER_ID = K.CUSTOMER_ID
12      ORDER BY K.NUM_FILMS DESC;
```

RESULT

	FIRST_NAME	EMAIL	NUM_FILMS
▶	ELEANOR	ELEANOR.HUNT@sakilacustomer.org	46
	KARL	KARL.SEAL@sakilacustomer.org	45
	CLARA	CLARA.SHAW@sakilacustomer.org	42
	MARCIA	MARCIA.DEAN@sakilacustomer.org	42
	TAMMY	TAMMY.SANDERS@sakilacustomer.org	41
	SUE	SUE.PETERS@sakilacustomer.org	40
	WESLEY	WESLEY.BULL@sakilacustomer.org	40

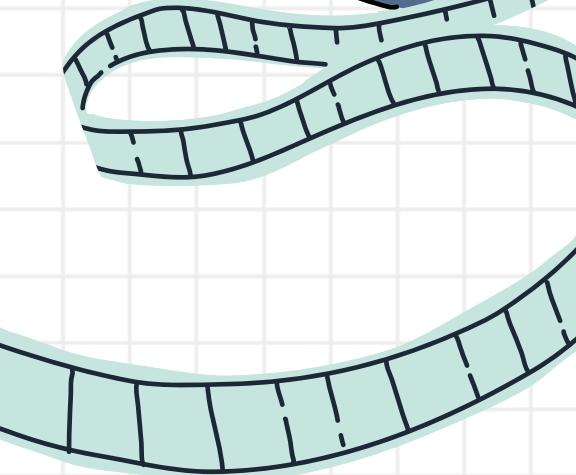
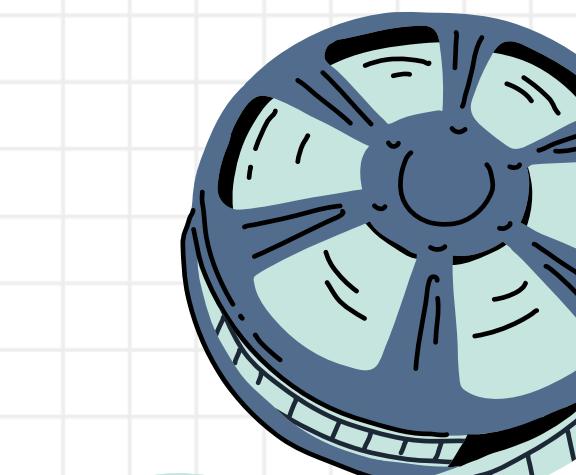
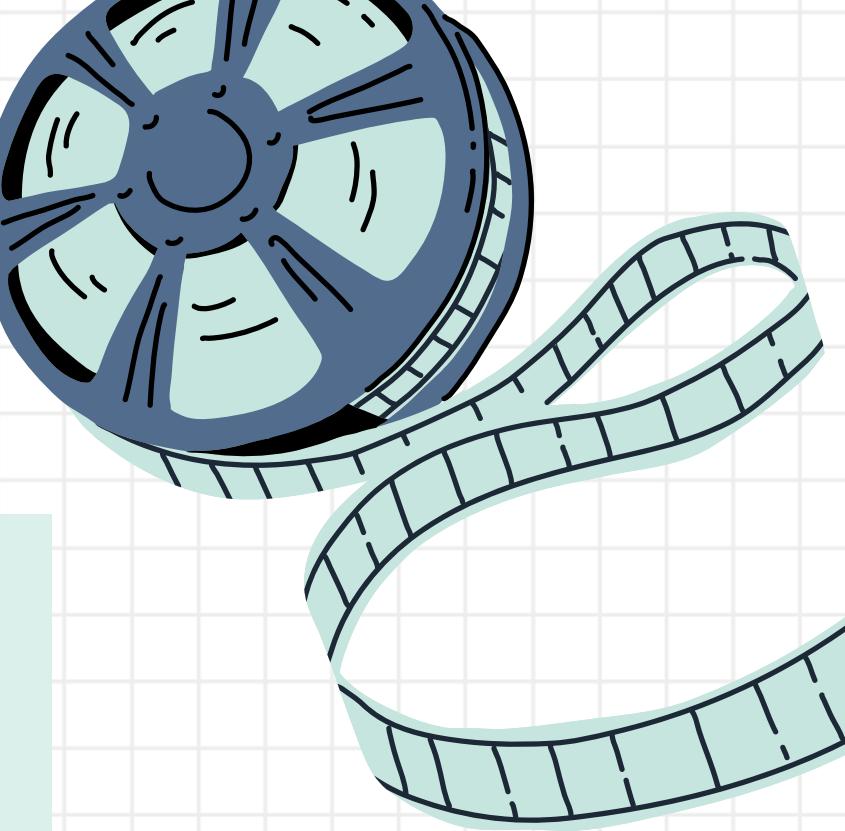


9. List the films and their categories.

```
1 •   SELECT  
2       film.title, k.name  
3   FROM  
4       film  
5       JOIN  
6   (SELECT  
7       film_category.film_id, category.name  
8   FROM  
9       film_category  
10      JOIN category ON film_category.category_id = category.category_id  
11      ORDER BY film_category.film_id) AS k ON film.film_id = k.film_id;
```

RESULT

	title	name
▶	AMADEUS HOLY	Action
	AMERICAN CIRCUS	Action
	ANTITRUST TOMATOES	Action
	ARK RIDGEMONT	Action
	BAREFOOT MANCHURIAN	Action
	BERETS AGENT	Action



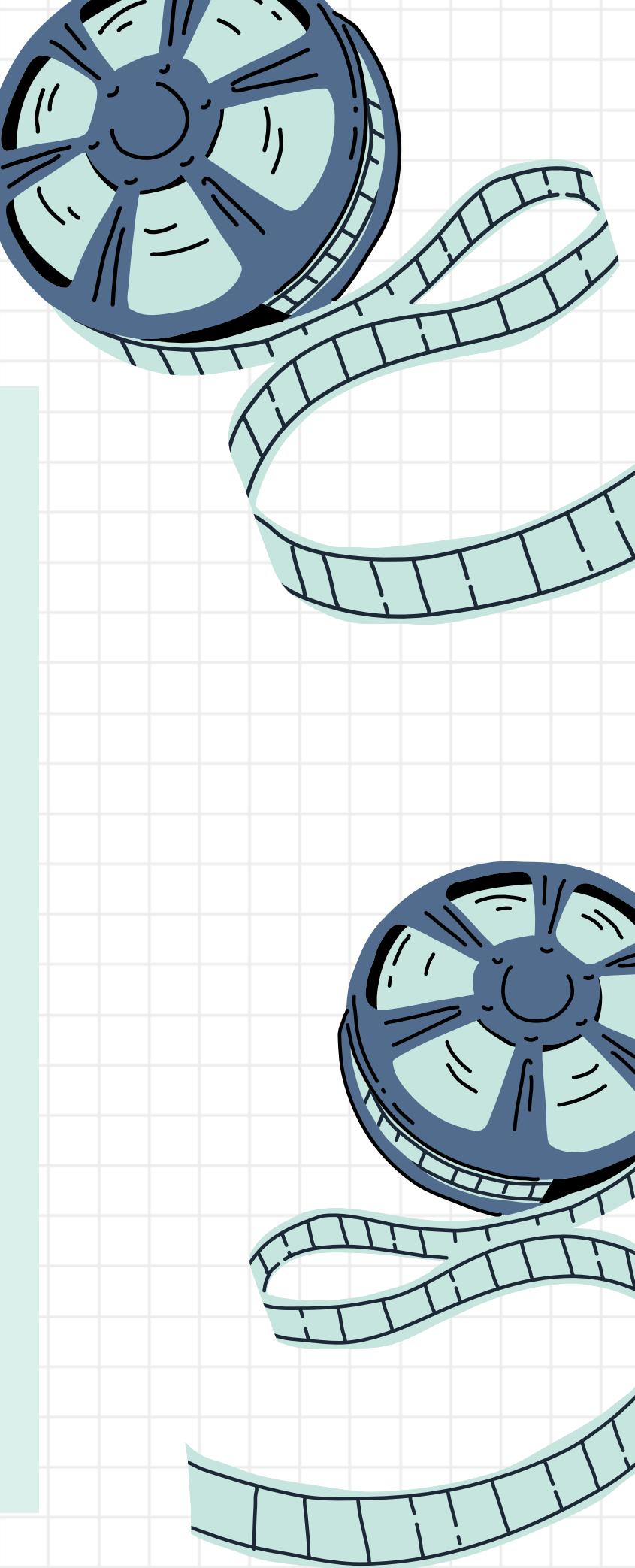
10. Show the top 5 customers with the highest total rental payments.

```
1 •  SELECT
2      customer.first_name, customer.last_name, k.total_payment
3  FROM
4      customer
5      JOIN
6      (SELECT
7          customer_id, SUM(amount) AS total_payment
8      FROM
9          payment
10     GROUP BY customer_id
11     ORDER BY total_payment DESC
12     LIMIT 5) AS k ON customer.customer_id = k.customer_id;
```

X
X

RESULT

	first_name	last_name	total_payment
▶	KARL	SEAL	221.55
	ELEANOR	HUNT	216.54
	CLARA	SHAW	216.54
	MARION	SNYDER	194.61
	RHONDA	KENNEDY	194.61



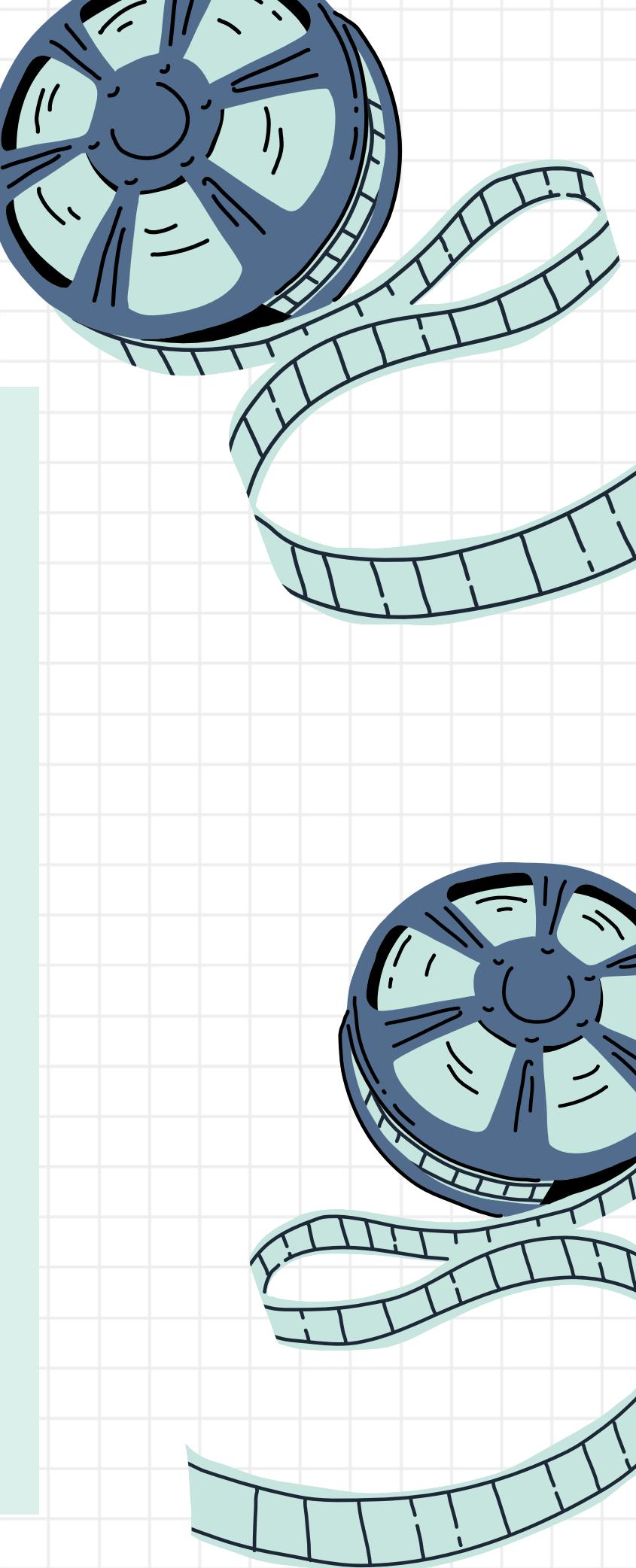
11. Retrieve the names of actors who have appeared in the same film together.

```
1 •  SELECT
2      m.title,
3      m.first_name,
4      m.last_name,
5      ROW_NUMBER() OVER(PARTITION BY m.title ORDER BY m.first_name) AS row_num
6  FROM
7      (SELECT
8          film.title,
9          k.first_name,
10         k.last_name
11     FROM
12         film
13     JOIN
14         (SELECT
15             film_actor.film_id,
16             actor.first_name,
17             actor.last_name
18           FROM
19             film_actor
20           JOIN
21             actor
22           ON
23             film_actor.actor_id = actor.actor_id) AS k
24     ON
25         film.film_id = k.film_id) AS m;
```

X
X

RESULT

	title	first_name	last_name	row_num
▶	ACADEMY DINOSAUR	CHRISTIAN	GABLE	1
	ACADEMY DINOSAUR	JOHNNY	CAGE	2
	ACADEMY DINOSAUR	LUCILLE	TRACY	3
	ACADEMY DINOSAUR	MARY	KEITEL	4
	ACADEMY DINOSAUR	MENA	TEMPLE	5
	ACADEMY DINOSAUR	OPRAH	KILMER	6
	ACADEMY DINOSAUR	PENELOPE	GUINNESS	7
	ACADEMY DINOSAUR	ROCK	DUKAKIS	8
	ACADEMY DINOSAUR	SANDRA	PECK	9
	ACADEMY DINOSAUR	WARREN	NOLTE	10
▶	ACE GOLDFINGER	BOB	FAWCETT	1
	ACE GOLDFINGER	CHRIS	DEPP	2
	ACE GOLDFINGER	MINNIE	ZELLWEGER	3
	ACE GOLDFINGER	SEAN	GUINNESS	4



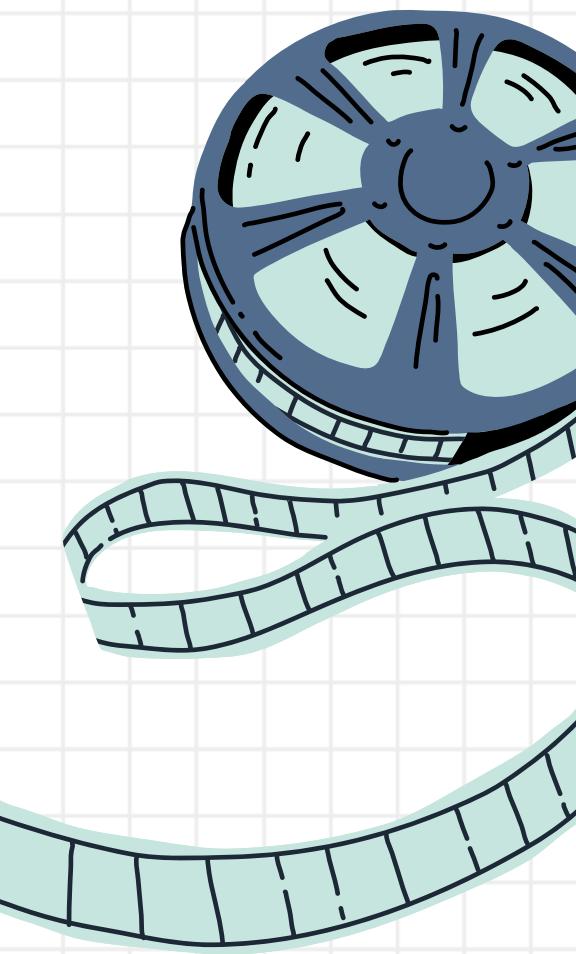
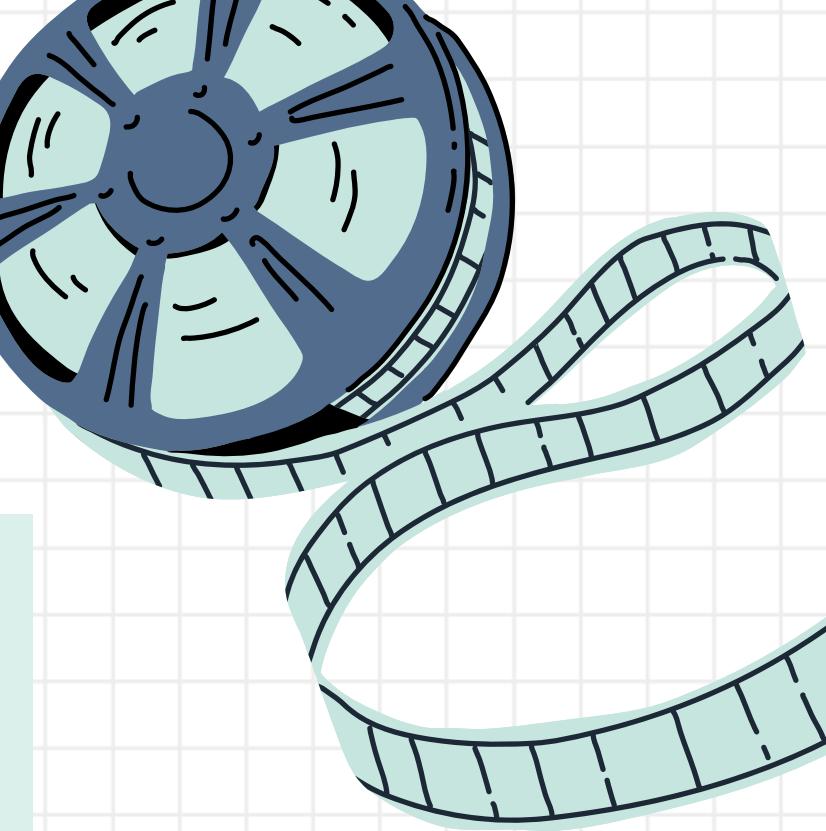
12. Find customers who have not rented any films.

```
1 • SELECT  
2     customer.customer_id,  
3     customer.first_name,  
4     customer.last_name,  
5     rental.inventory_id  
6 FROM  
7     customer  
8     LEFT JOIN  
9     rental ON customer.customer_id = rental.customer_id  
10    GROUP BY customer.first_name , customer.last_name  
11    HAVING inventory_id IS NULL;
```

X
X

RESULT

	customer_id	first_name	last_name	inventory_id



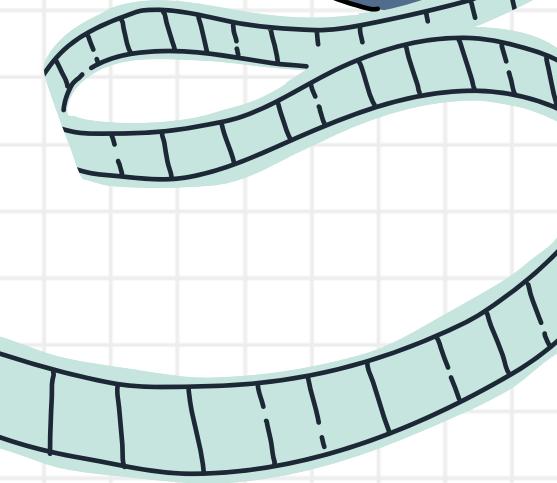
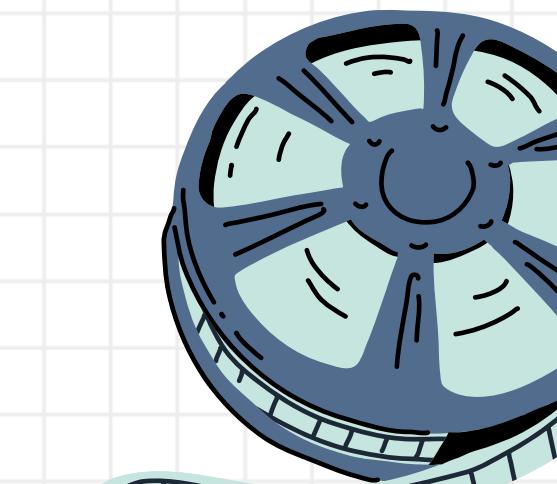
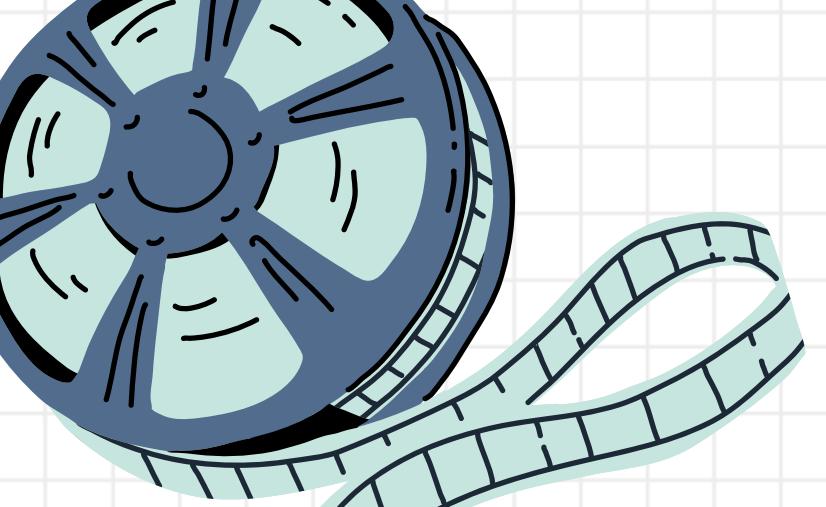
13. Show the staff members and the stores they manage.

```
1 •   SELECT  
2       st.first_name, st.last_name, ad.address  
3   FROM  
4       staff st  
5   JOIN  
6       address ad ON st.address_id = ad.address_id;
```

X
X

RESULT

	first_name	last_name	address
▶	Mike	Hillyer	23 Workhaven Lane
	Jon	Stephens	1411 Lillydale Drive

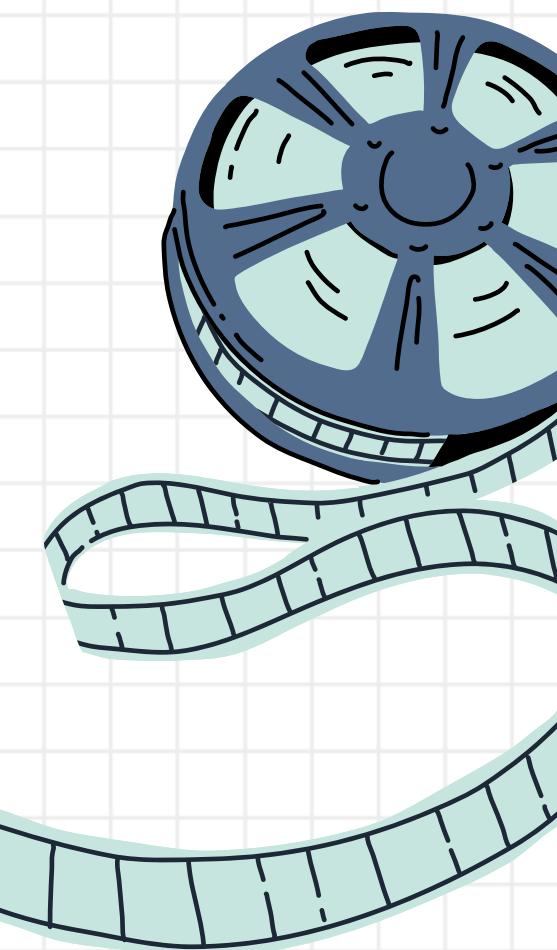
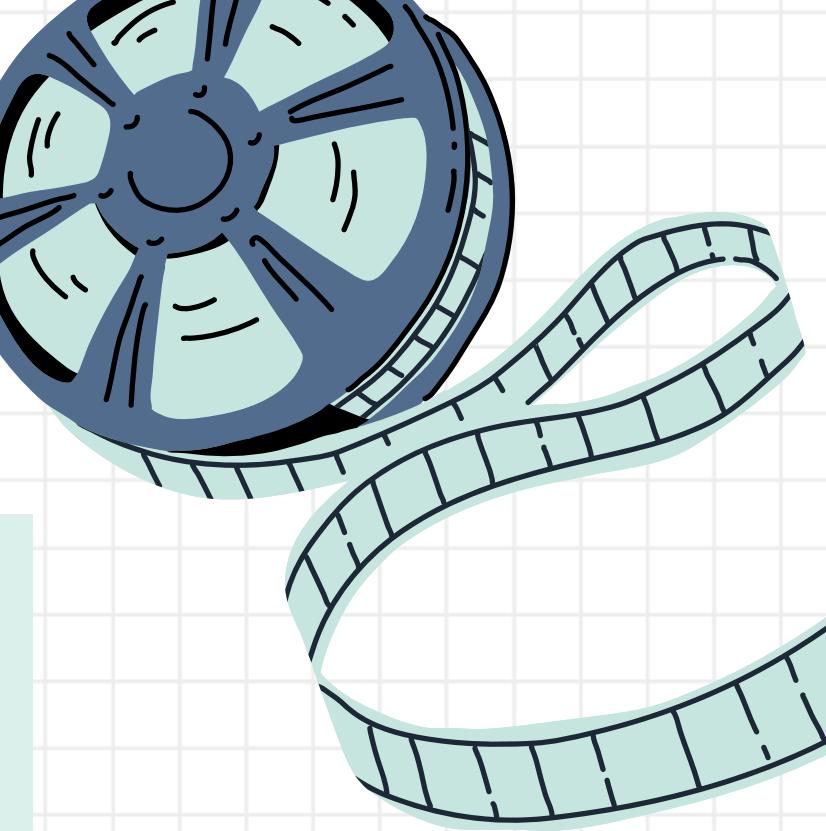


14. List all films and their respective languages.

```
1   SELECT
2       f.title, l.name
3   FROM
4       film f
5   JOIN
6       language l ON f.language_id = l.language_id;
```

RESULT

	title	name
▶	ACADEMY DINOSAUR	English
	ACE GOLDFINGER	English
	ADAPTATION HOLES	English
	AFFAIR PREJUDICE	English
	AFRICAN EGG	English
	AGENT TRUMAN	English
	AIRPLANE SIERRA	English
	AIRPORT POLLOCK	English

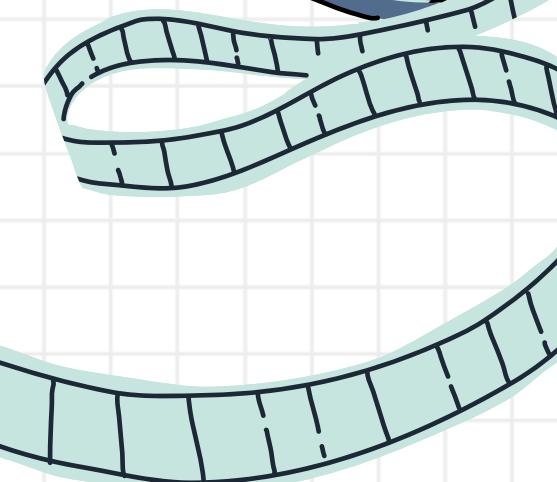
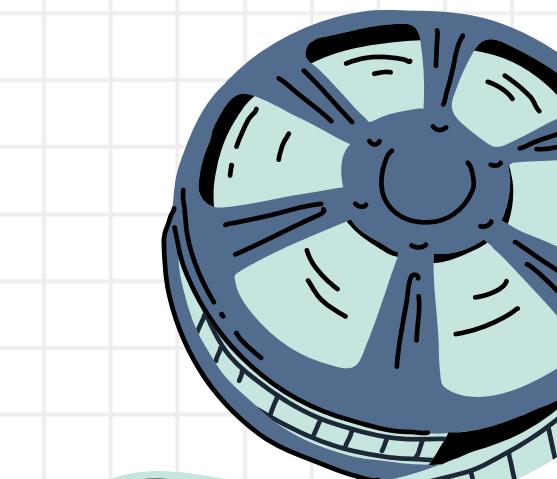
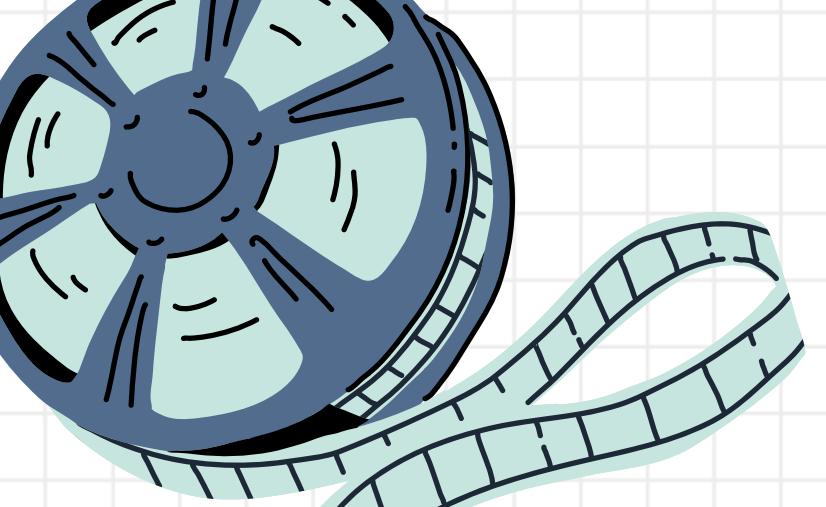


15. Find the total revenue for each category.

```
1  SELECT
2      x.name, SUM(p.amount) AS revenue
3  FROM
4      payment p
5      JOIN
6          (SELECT
7              a.name, r.rental_id
8          FROM
9              rental r
10         JOIN (SELECT
11             k.name, i.inventory_id
12         FROM
13             inventory i
14         JOIN (SELECT
15             c.name, fc.film_id
16         FROM
17             category c
18         JOIN film_category fc ON c.category_id = fc.category_id) AS k ON i.film_id = k.film_id) AS a ON r.inventory_id = a.inventory_id) AS x ON p.rental_id = x.rental_id
19  GROUP BY x.name;
```

RESULT

	name	revenue
▶	Action	4375.85
	Animation	4656.30
	Children	3655.55
	Classics	3639.59
	Comedy	4383.58
	Documentary	4217.52
	Drama	4587.39
	Family	4226.07
	Foreign	4270.67
	Games	4281.33
	Horror	3722.54
	Music	3417.72
	New	4351.62
	Sci-Fi	4756.98
	Sports	5314.21
	Travel	3549.64



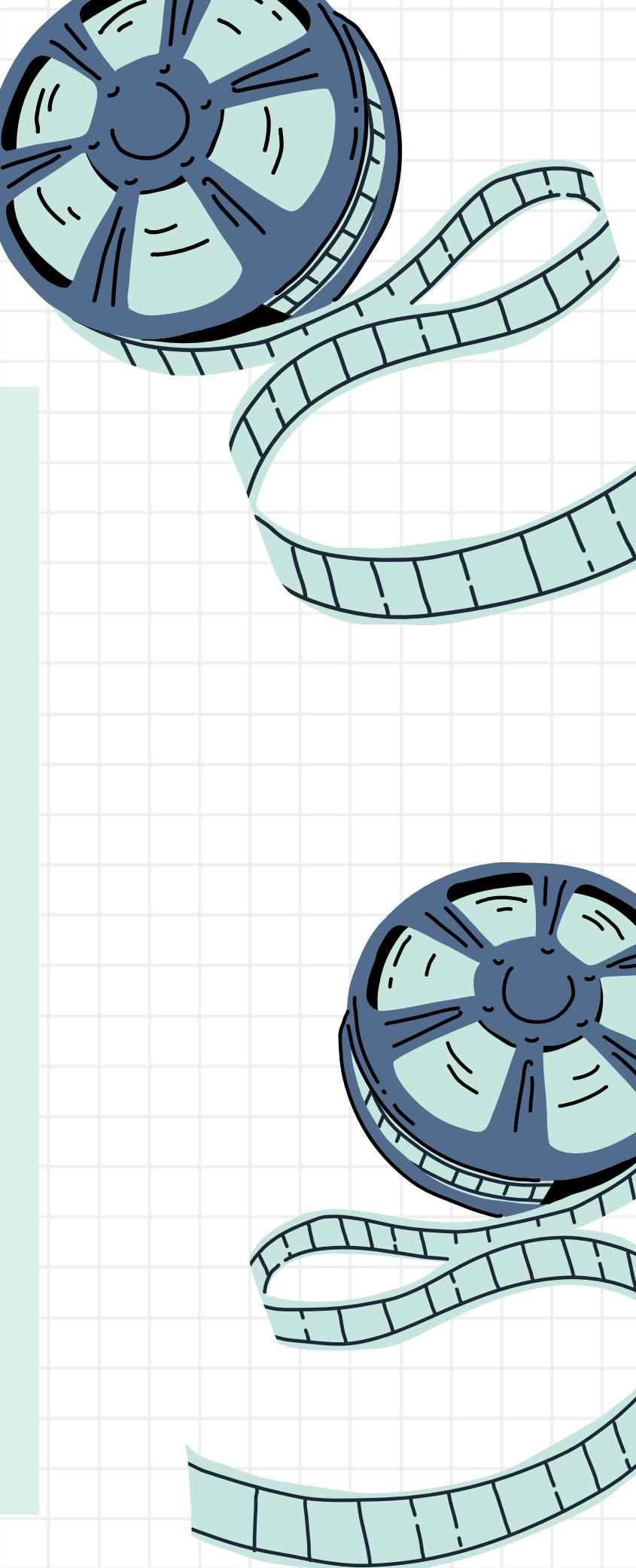
16. Show the last rental date for each customer.

```
1 •   SELECT
2       c.first_name, c.last_name, r.last_date
3   FROM
4       customer c
5       JOIN
6       (SELECT
7           customer_id, MAX(rental_date) AS last_date
8       FROM
9           rental
10      GROUP BY customer_id) AS r ON c.customer_id = r.customer_id;
```

X
X

RESULT

	first_name	last_name	last_date
▶	MARY	SMITH	2005-08-22 20:03:46
	PATRICIA	JOHNSON	2005-08-23 17:39:35
	LINDA	WILLIAMS	2005-08-23 07:10:14
	BARBARA	JONES	2005-08-23 07:43:00
	ELIZABETH	BROWN	2006-02-14 15:16:03
	JENNIFER	DAVIS	2005-08-23 06:41:32
	MARIA	MILLER	2005-08-21 04:49:48
	SUSAN	WILSON	2005-08-23 14:31:19
	MARGARET	MOORE	2006-02-14 15:16:03
	DOROTHY	TAYLOR	2005-08-22 21:59:29
	LISA	ANDERSON	2006-02-14 15:16:03
	NANCY	THOMAS	2005-08-23 20:28:44
	KAREN	JACKSON	2005-08-22 20:51:24
	BETTY	WHITE	2006-02-14 15:16:03
	HELEN	HARRIS	2006-02-14 15:16:03

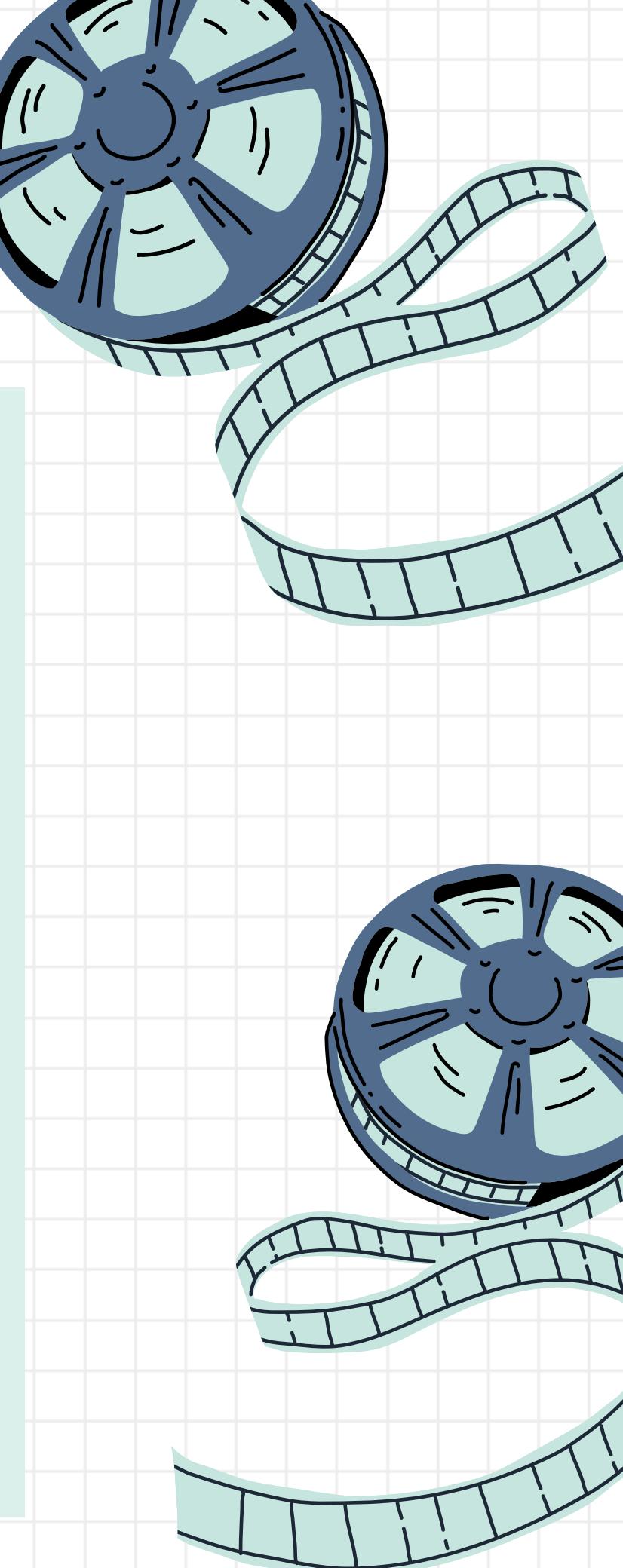


17. Find the films with a rental rate greater than the average rental rate.

```
1  SELECT  
2      title, rental_rate  
3  FROM  
4      film  
5  WHERE  
6      rental_rate > (SELECT  
7          AVG(rental_rate)  
8      FROM  
9          film)  
10     ORDER BY rental_rate DESC;
```

RESULT

	title	rental_rate
	ACE GOLDFINGER	4.99
	AIRPLANE SIERRA	4.99
	AIRPORT POLLOCK	4.99
▶	ALADDIN CALENDAR	4.99
	ALI FOREVER	4.99
	AMELIE HELLFIGHTERS	4.99
	AMERICAN CIRCUS	4.99
	ANTHEM LUKE	4.99

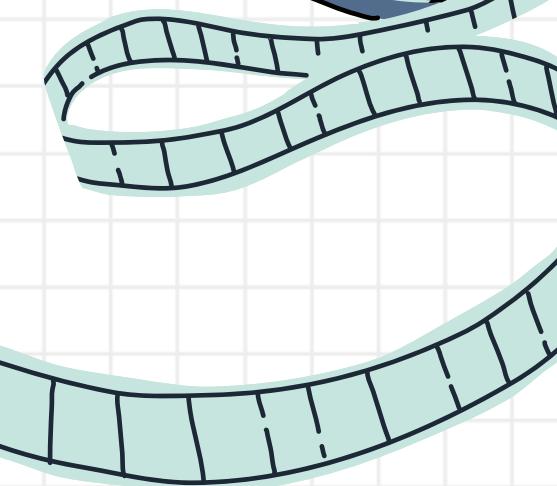
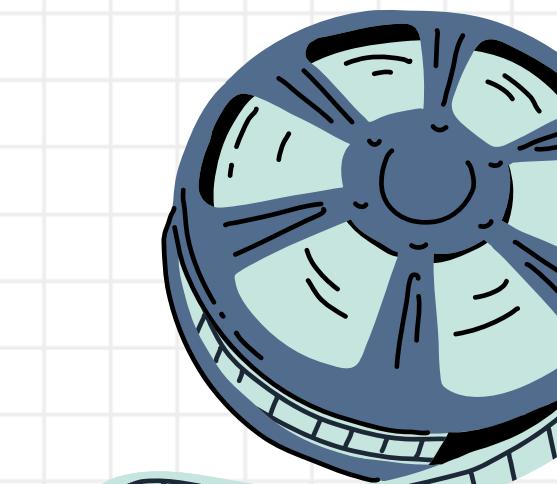
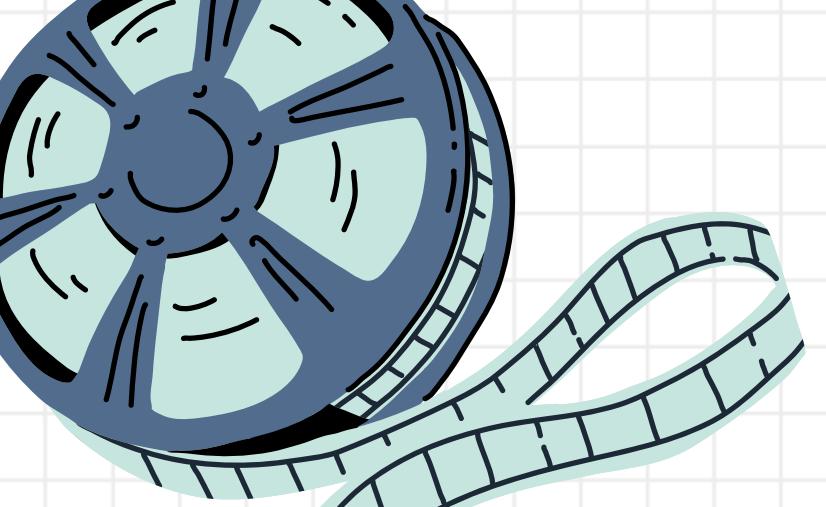


18. List customers who have rented films from the "Music" category.

```
1  SELECT DISTINCT
2      (c.first_name), c.last_name
3  FROM
4      customer c
5          JOIN
6      (SELECT
7          r.customer_id
8      FROM
9          rental r
10     JOIN (SELECT
11         i.inventory_id
12     FROM
13         inventory i
14     JOIN (SELECT
15         fc.film_id
16     FROM
17         film_category fc
18     JOIN (SELECT
19         c.name, c.category_id
20     FROM
21         category c
22     JOIN film_category fc ON c.category_id = fc.category_id
23     WHERE
24         c.name LIKE 'music') v ON fc.category_id = v.category_id) e ON i.film_id = e.film_id) d ON r.inventory_id = d.inventory_id) h ON c.customer_id = h.customer_id;
```

RESULT

	first_name	last_name
▶	TED	BREAUX
	MELVIN	ELLINGTON
	JAMIE	RICE
	BERNARD	COLBY
	KELLY	TORRES
	ELIZABETH	BROWN
	BYRON	BOX
	JOHNNY	TURPIN
	CHRISTINA	RAMIREZ
	DIANE	COLLINS
	CLIFFORD	BOWENS
	WENDY	HARRISON
	MARIAN	MENDOZA



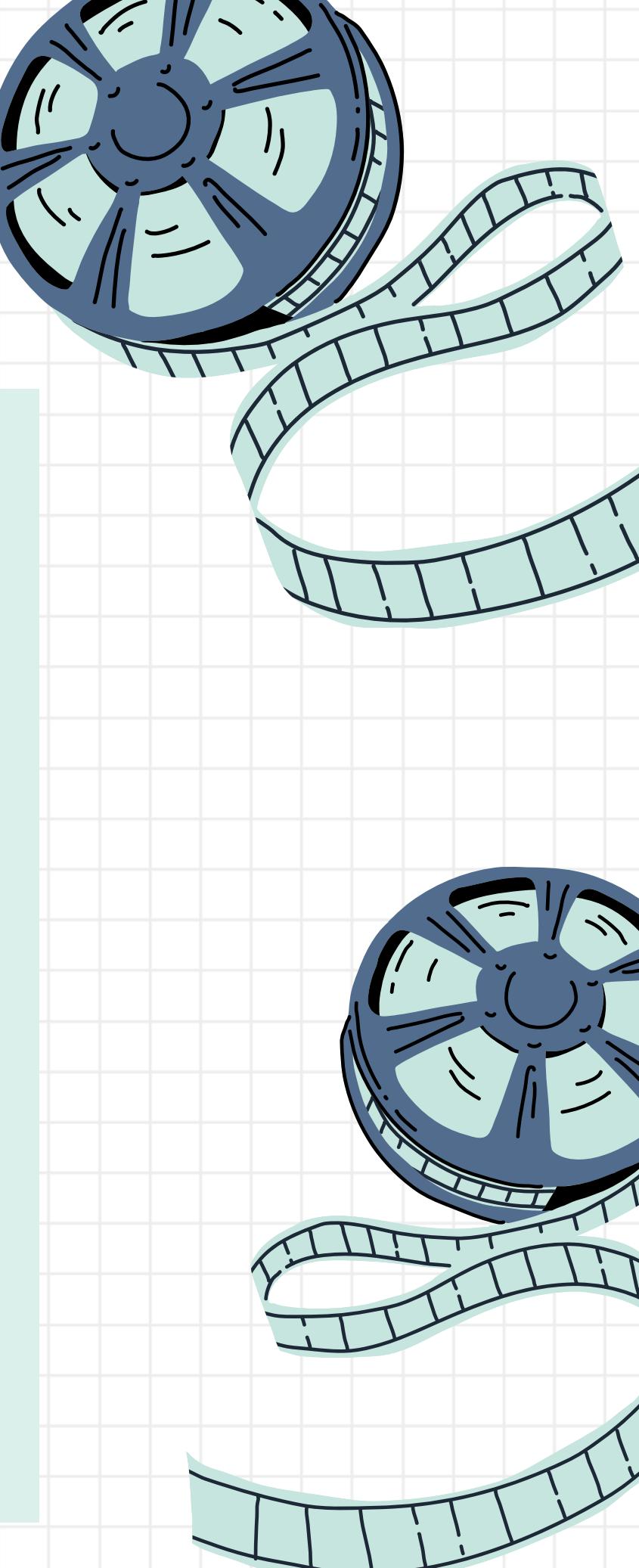
19. Find the actors who have appeared in the most films.

```
1  SELECT
2      a.first_name, a.last_name, f.num_films
3  FROM
4      actor a
5          JOIN
6      (SELECT
7          actor_id, COUNT(film_id) AS num_films
8      FROM
9          film_actor
10     GROUP BY actor_id) f ON a.actor_id = f.actor_id
11     ORDER BY f.num_films DESC;
```

X
X

RESULT

	first_name	last_name	num_films
▶	GINA	DEGENERES	42
	WALTER	TORN	41
	MARY	KEITEL	40
	MATTHEW	CARREY	39
	SANDRA	KILMER	37
	SCARLETT	DAMON	36
	UMA	WOOD	35
	VAL	BOLGER	35
	HENRY	BERRY	35

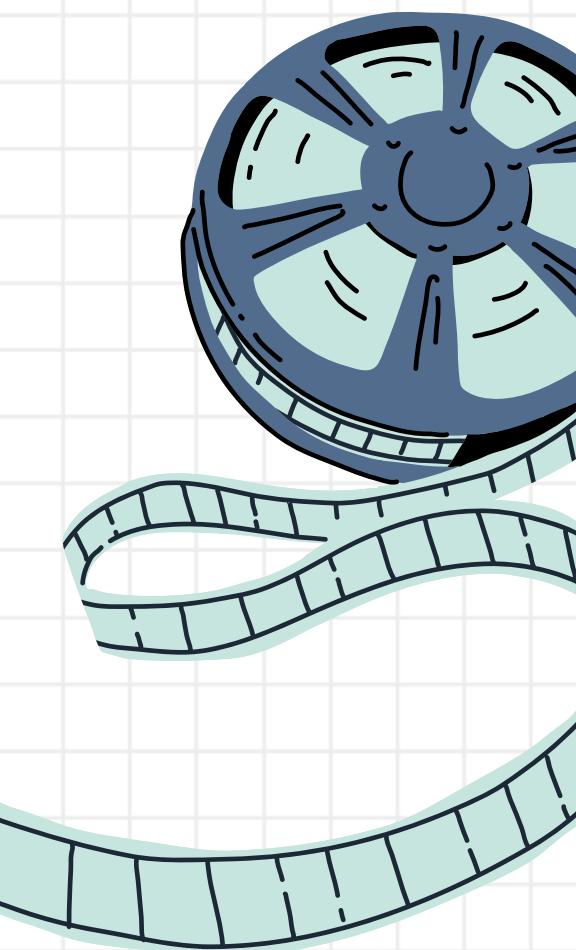
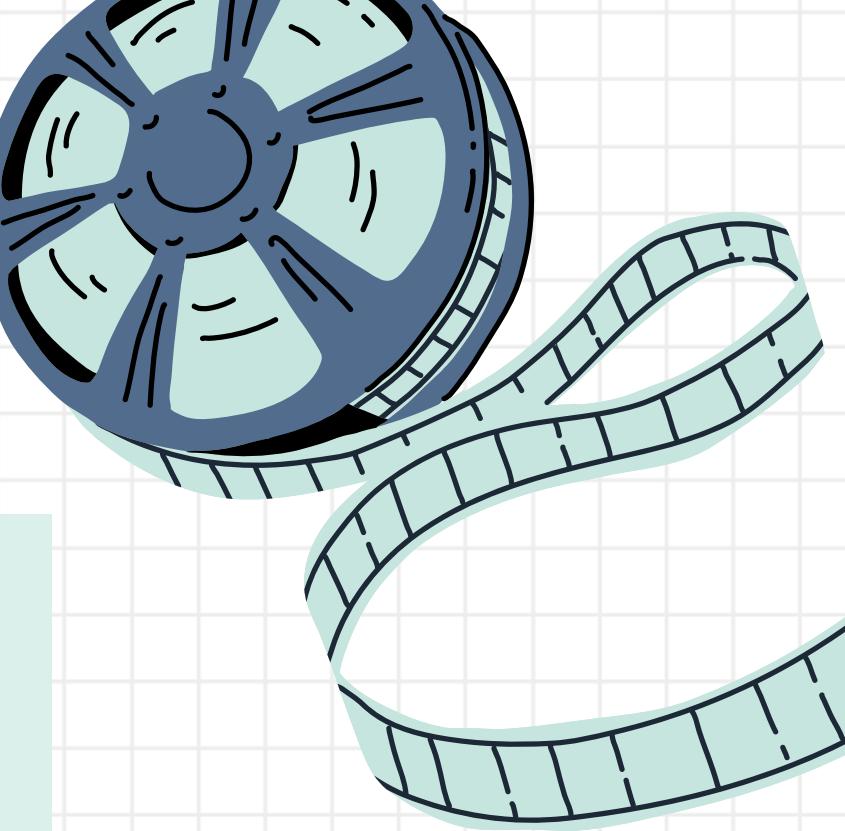


20. Retrieve the number of films in each category.

```
1 •   SELECT
2       c.name, f.no_films
3   FROM
4       category c
5       JOIN
6       (SELECT
7           category_id, COUNT(film_id) AS no_films
8   FROM
9       film_category
10      GROUP BY category_id) f ON c.category_id = f.category_id;
```

RESULT

	name	no_films
▶	Action	64
	Animation	66
	Children	60
	Classics	57
	Comedy	58
	Documentary	68
	Drama	62
	Family	69
	Foreign	73



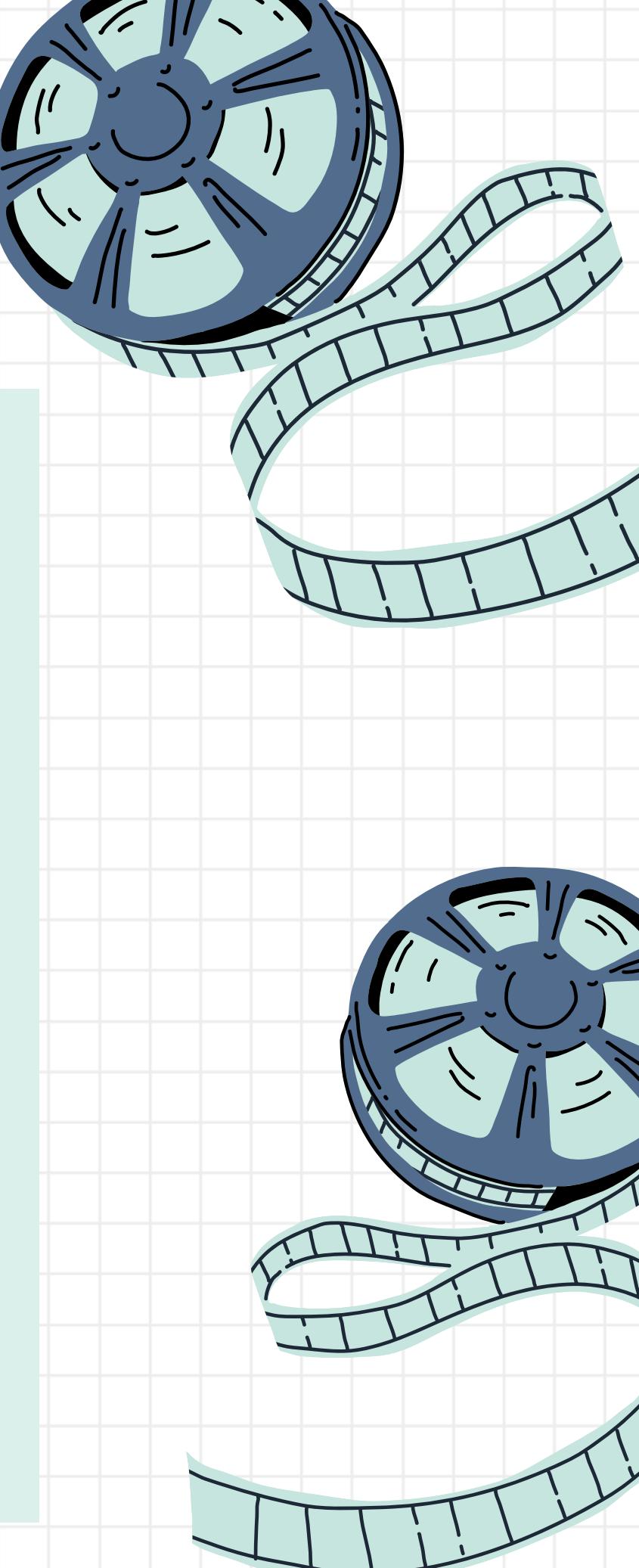
21. Show the customer with the highest total rental duration.

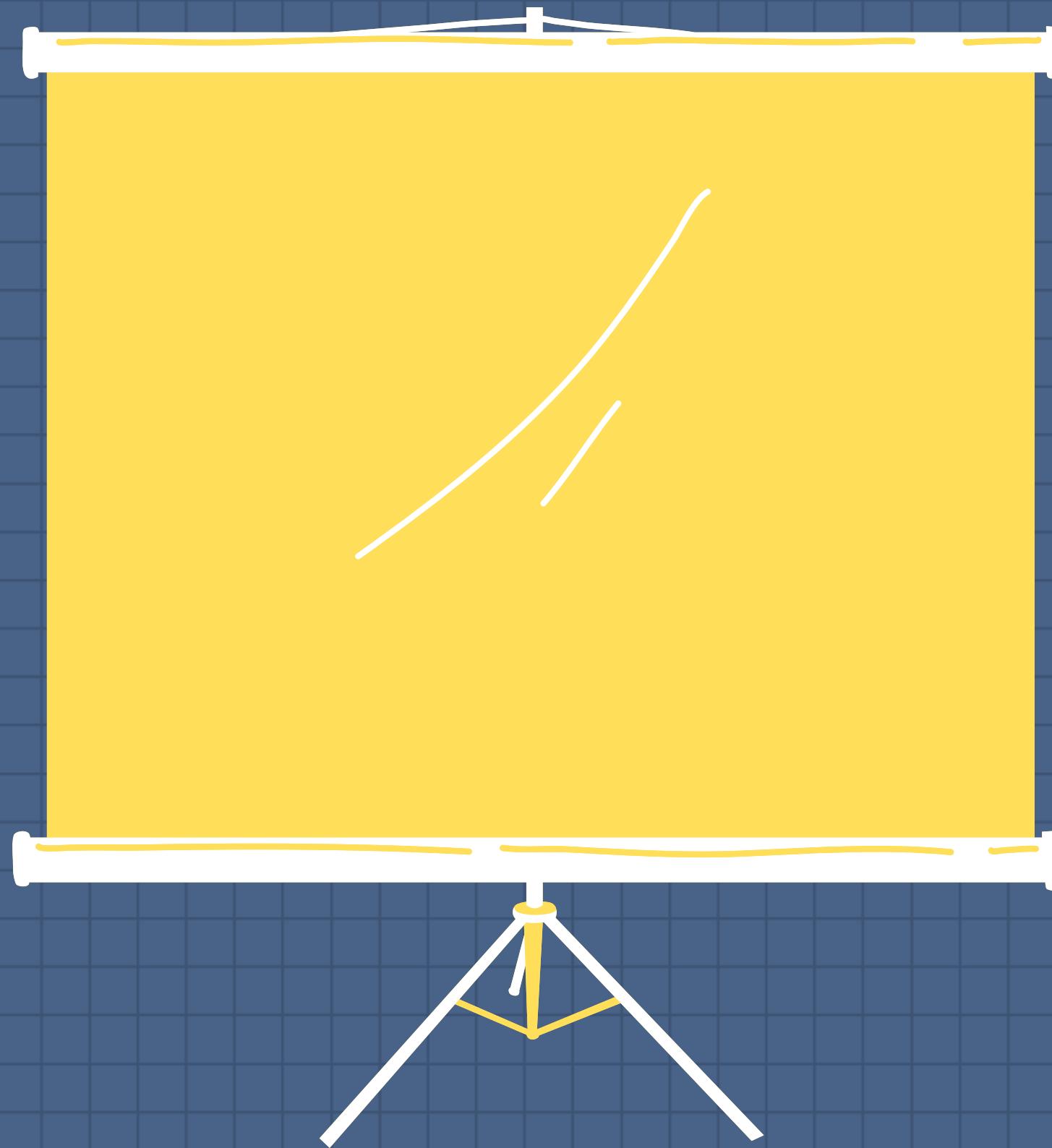
```
1 •   SELECT
2       c.first_name, c.last_name, SUM(r.duration) AS duration
3   FROM
4       customer c
5       JOIN
6           (SELECT
7               customer_id, DATEDIFF(return_date, rental_date) AS duration
8           FROM
9               rental) r ON c.customer_id = r.customer_id
10      GROUP BY c.first_name
11      ORDER BY duration DESC;
```

X
X

RESULT

	first_name	last_name	duration
▶	MARION	SNYDER	349
	LESLIE	GORDON	296
	TRACY	COLE	274
	JAMIE	RICE	274
	WILLIE	HOWELL	265
	KARL	SEAL	263
	JESSIE	BANKS	253
	TERRY	CARLSON	243
	ELEANOR	HUNT	241





ADVANCED QUERIES



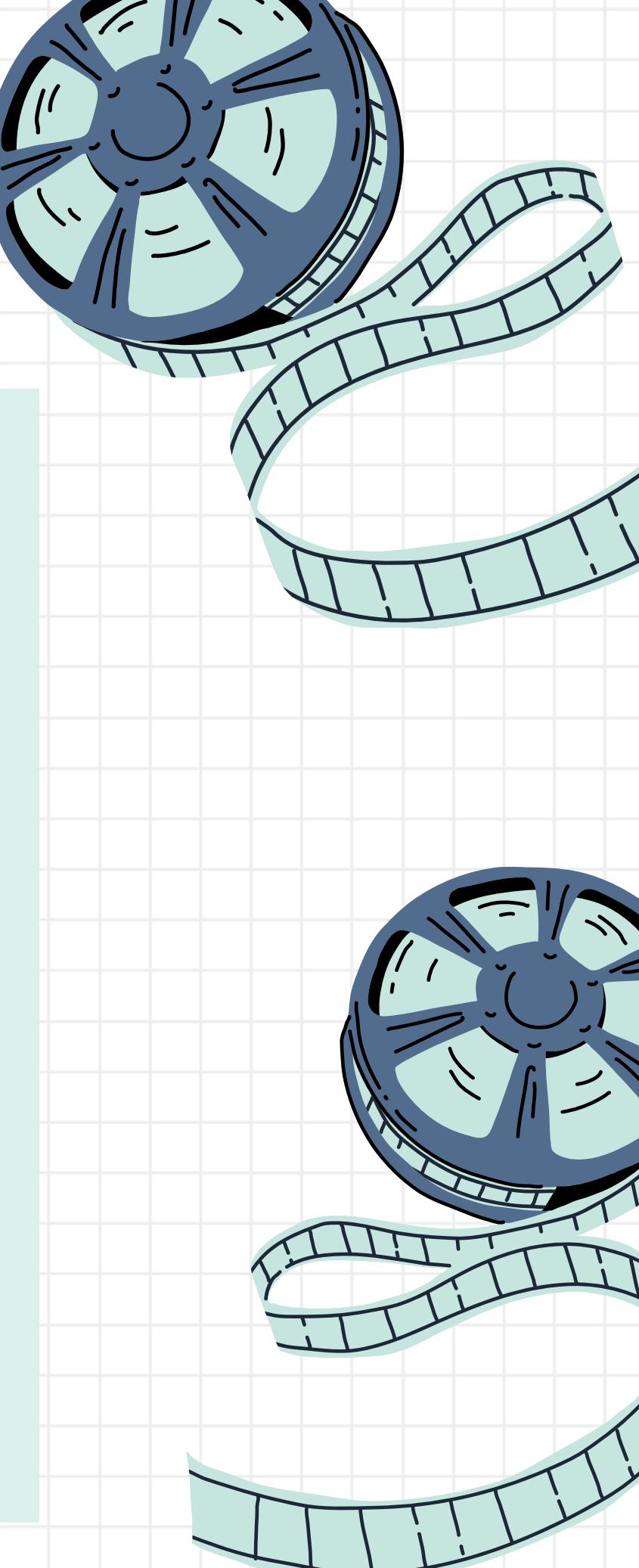
22. Calculate the average rental duration for each film.

```
1  SELECT
2      f.title, ROUND(AVG(y.duration), 2) duration
3  FROM
4      film f
5          JOIN
6      (SELECT
7          i.film_id, r.duration
8      FROM
9          inventory i
10     JOIN (SELECT
11         inventory_id, DATEDIFF(return_date, rental_date) AS duration
12     FROM
13         rental) r ON i.inventory_id = r.inventory_id) y ON f.film_id = y.film_id
14  GROUP BY f.title
15  ORDER BY duration DESC;
```

X
X

RESULT

	title	duration
▶	AFRICAN EGG	7.09
	FLIGHT LIES	7.08
	HARDLY ROBBERS	7.00
	IMPACT ALADDIN	7.00
	WAGON JAWS	6.90
	MOTHER OLEANDER	6.86
	MADRE GABLES	6.79
	REDS POCUS	6.78
	SILVERADO GOLDFINGER	6.73

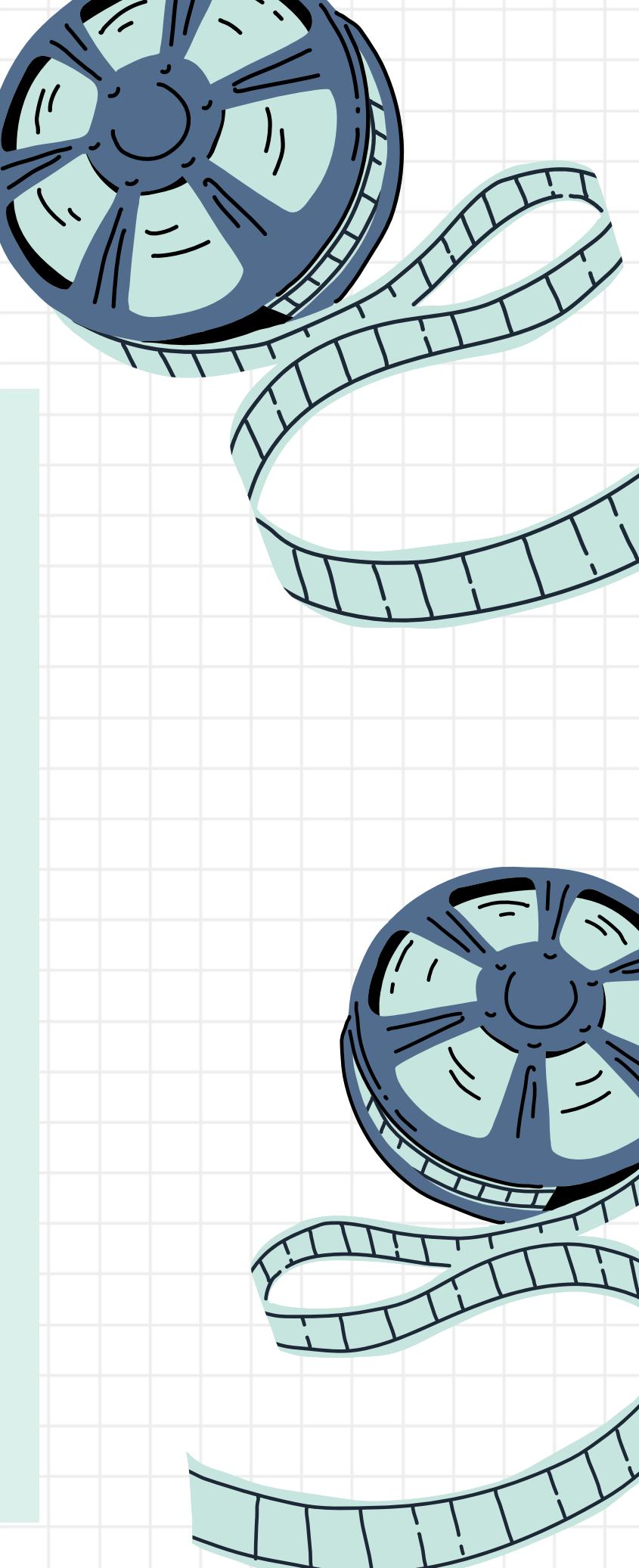


23. Find the five longest films.

```
1  SELECT
2      title, length
3  FROM
4      film
5  ORDER BY length DESC
6  LIMIT 5;
```

RESULT

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

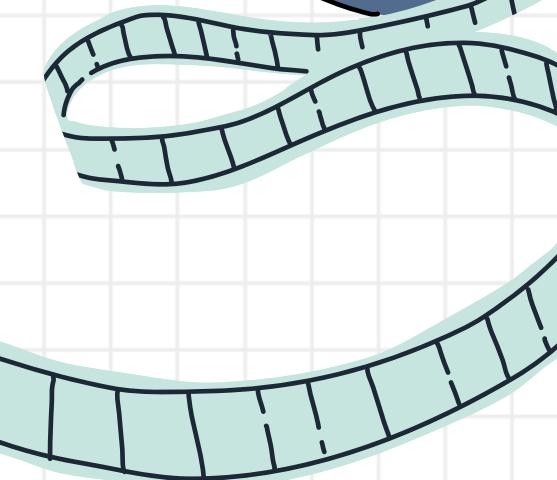
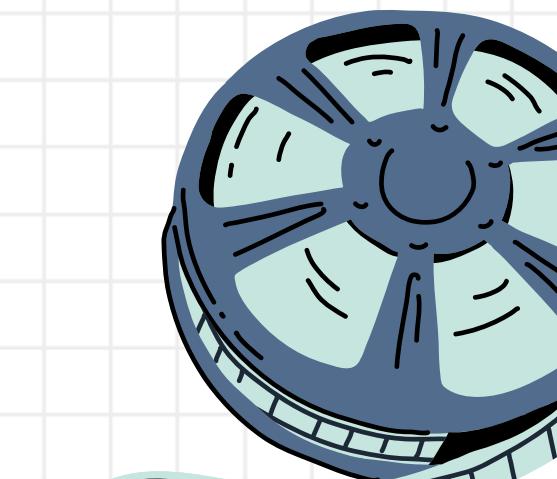
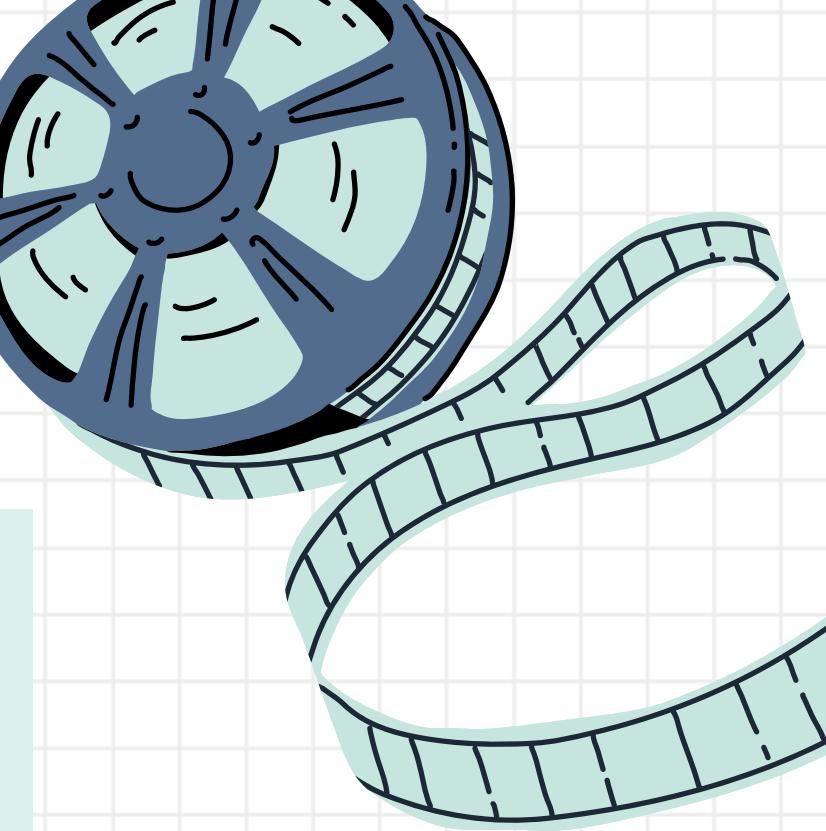


24. Show the customers who have rented the same film more than once.

```
1 •  SELECT
2      x.first_name, x.last_name, f.title, x.count
3  FROM
4      film f
5          RIGHT JOIN
6      (SELECT
7          y.first_name,
8          y.last_name,
9          i.film_id,
10         COUNT(i.film_id) AS count
11     FROM
12         inventory i
13     RIGHT JOIN (SELECT
14         c.first_name, c.last_name, r.inventory_id
15     FROM
16         rental r
17     LEFT JOIN customer c ON r.customer_id = c.customer_id) y ON y.inventory_id = i.inventory_id
18     GROUP BY y.first_name , y.last_name , i.film_id
19     HAVING count > 2) x ON x.film_id = f.film_id;
```

RESULT

	first_name	last_name	title	count
	EDWARD	BAUGH	FLATLINERS KILLER	3
	RANDY	GAITHER	DETECTIVE VISION	3
	GEORGE	LINTON	CADDYSHACK JEDI	3
▶	THELMA	MURRAY	DISCIPLE MOTHER	3

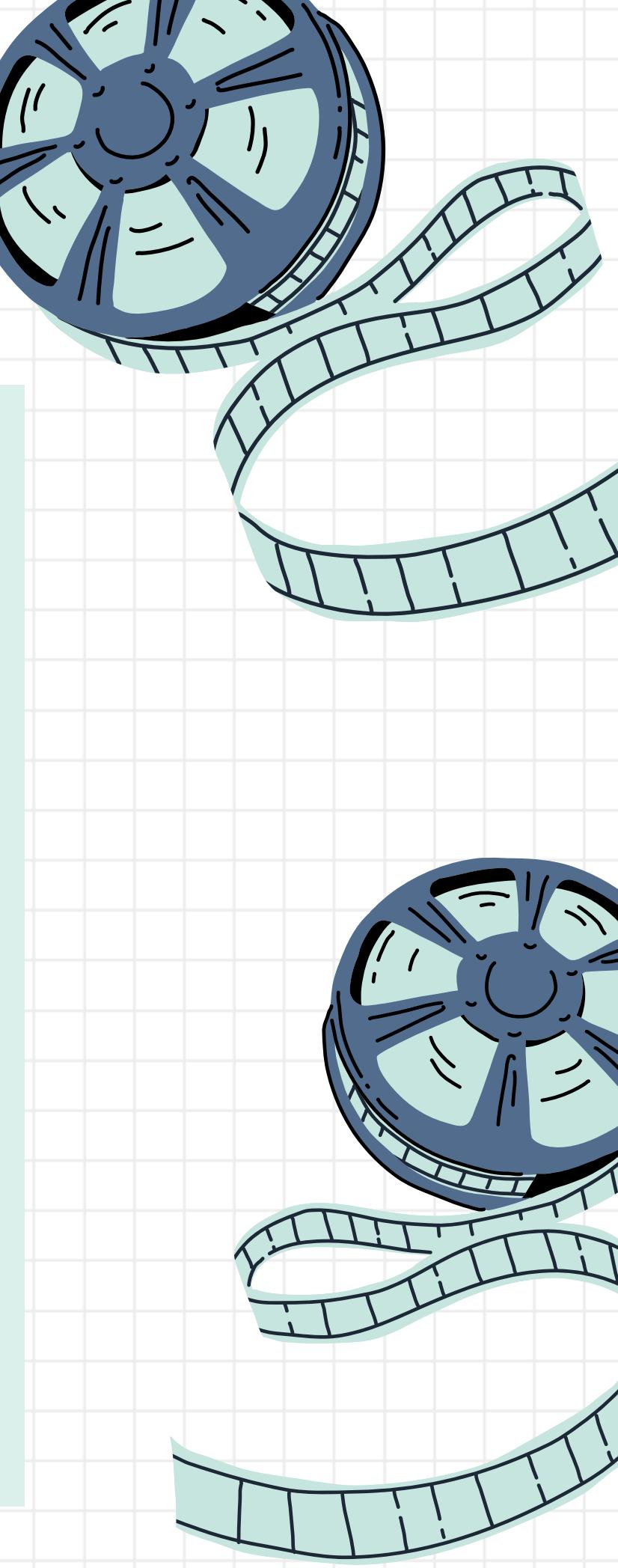
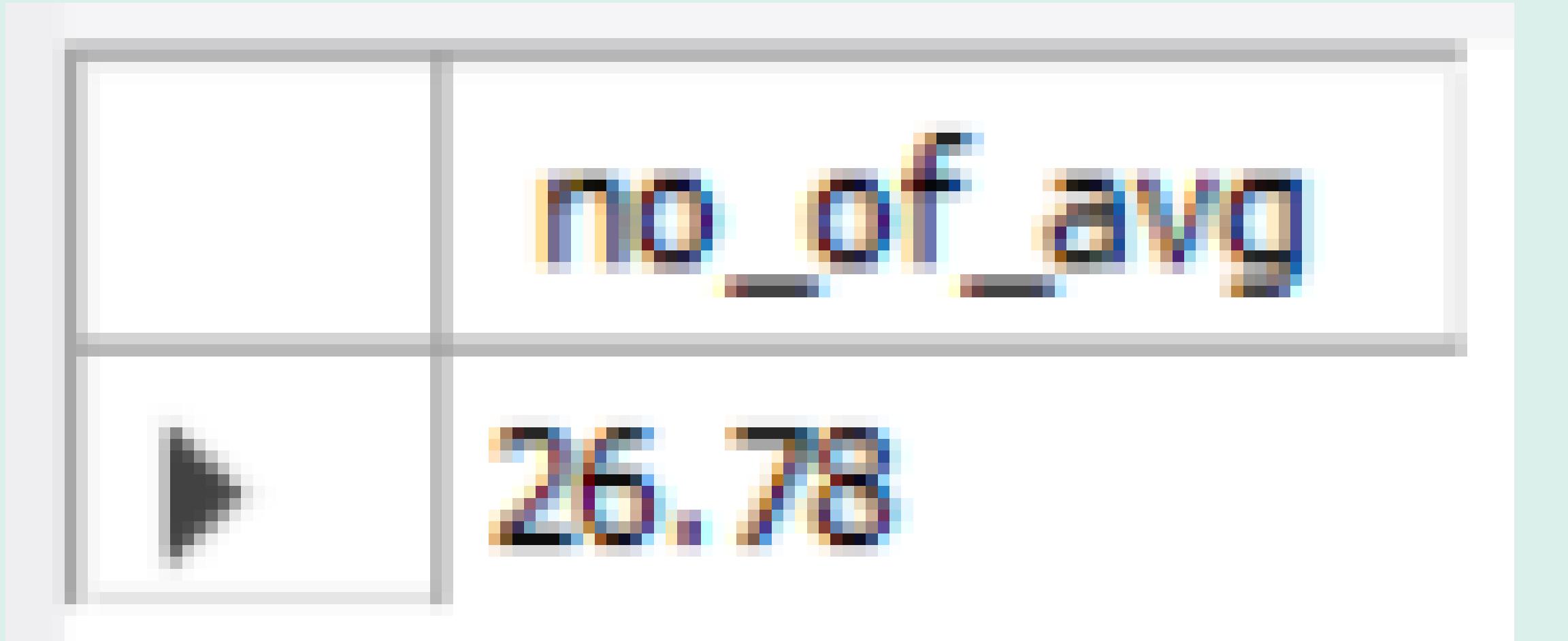


25. Calculate the average number of rentals per customer.

```
1 •   SELECT  
2       round(COUNT(*) / COUNT(DISTINCT customer_id),2) AS no_of_avg  
3   FROM  
4       rental;
```

X
X

RESULT

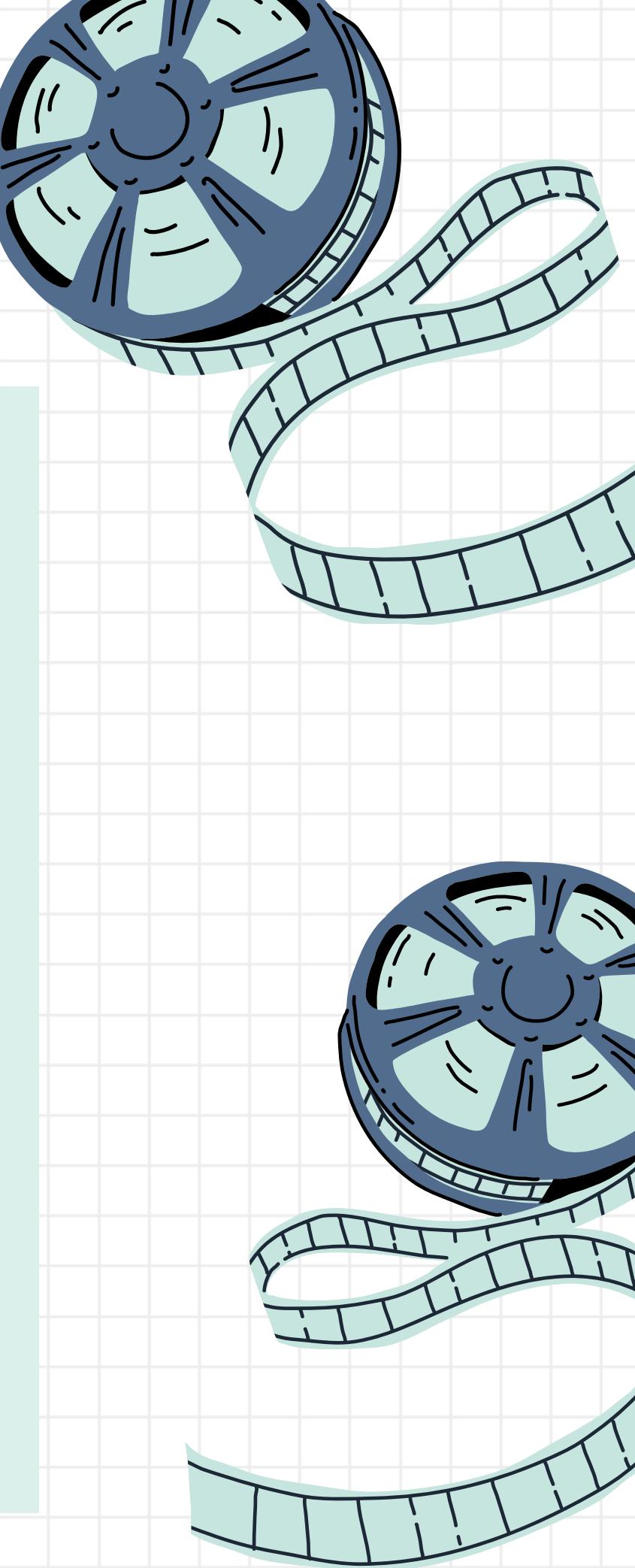


26. List the films that have not been rented.

```
1  SELECT
2      f.title
3  FROM
4      film f
5          LEFT JOIN
6      (SELECT
7          i.film_id, r.rental_id
8      FROM
9          inventory i
10     RIGHT JOIN rental r ON i.inventory_id = r.inventory_id) y ON f.film_id = y.film_id
11 WHERE
12     y.rental_id IS NULL;
```

RESULT

	title
▶	ALICE FANTASIA
	Apollo Teen
	ARGONAUTS TOWN
	ARK RIDGEMONT
	ARSENIC INDEPENDENCE
	BOONDOCK BALLROOM
	BUTCH PANTHER
	CATCH AMISTAD
	CHINATOWN GLADIATOR
	CHOCOLATE DUCK



27. Find the customers who have rented films from all categories.

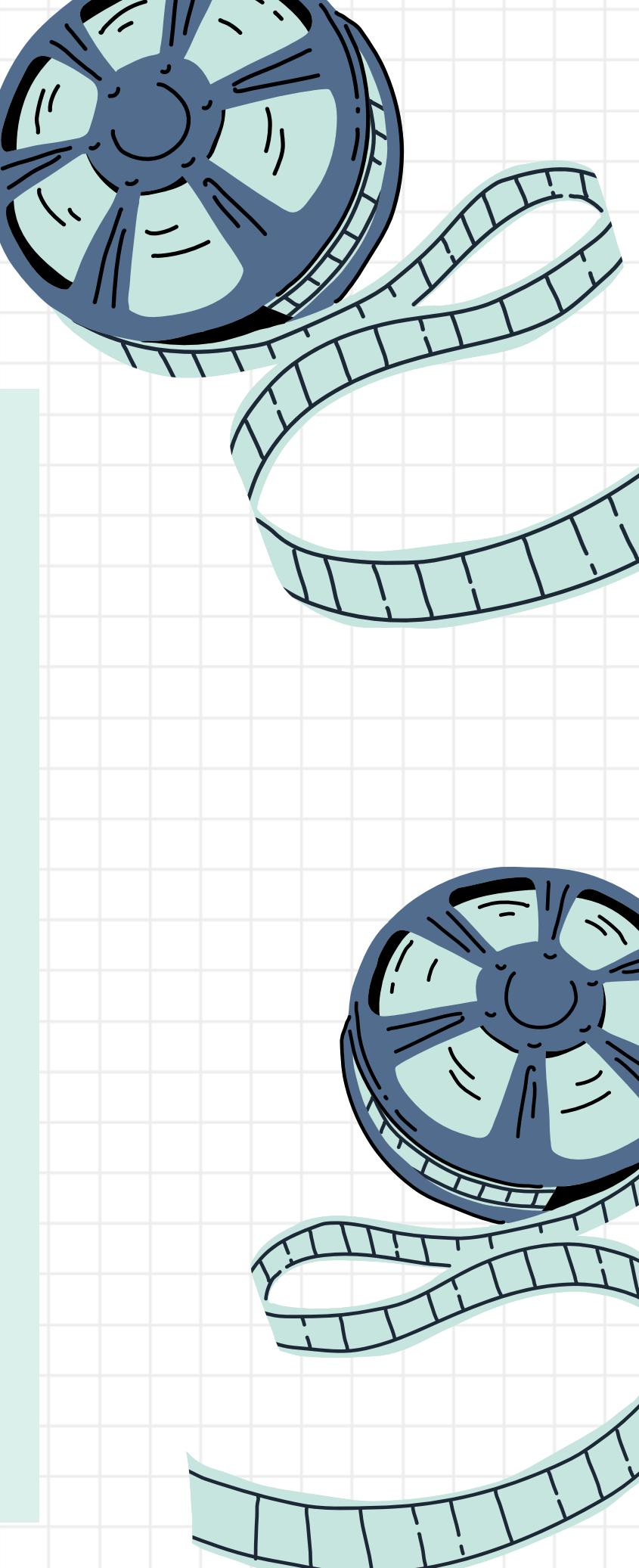
```
1 • SELECT
2     c.first_name,
3     c.last_name
4 FROM
5     customer c
6 • RIGHT JOIN (
7     SELECT
8         t.customer_id
9     FROM (
10        SELECT
11            y.customer_id,
12            y.category_id,
13            ROW_NUMBER() OVER (PARTITION BY y.customer_id ORDER BY y.category_id) AS row_num
14    FROM (
15        SELECT
16            DISTINCT fc.category_id,
17            x.customer_id
18        FROM
19            film_category fc
20        RIGHT JOIN (
```

```
21
22     r.customer_id,
23     i.film_id
24 FROM
25     rental r
26 LEFT JOIN
27     inventory i
28 ON
29     r.inventory_id = i.inventory_id
30 ) x
31 ON
32     fc.film_id = x.film_id
33 ORDER BY
34     x.customer_id,
35     fc.category_id
36 ) y
37 ) t
38 WHERE
39     t.row_num = 16
40 ) f
41 ON
42     c.customer_id = f.customer_id;
```



RESULT

	first_name	last_name
▶	CYNTHIA	YOUNG
	CAROLYN	PEREZ
	TAMMY	SANDERS
	CINDY	FISHER
	CLARA	SHAW
	SALLY	PIERCE
	ROSEMARY	SCHMIDT
	LEAH	CURTIS
	CAROLE	BARNETT



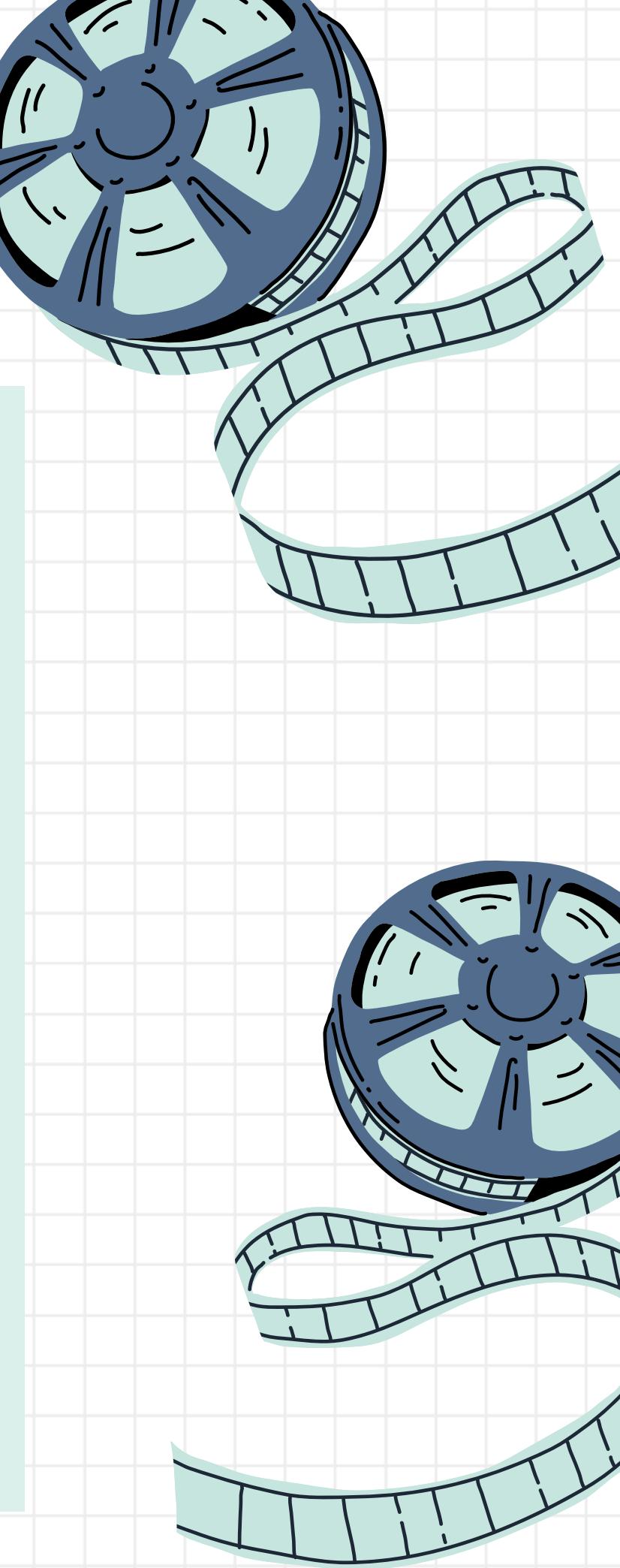
28. Show the films that were rented more than once on the same day.

```
1 •  SELECT DISTINCT
2      (i.film_id), r.rental_date, COUNT(i.film_id) AS count
3  FROM
4      rental r
5      LEFT JOIN
6      inventory i ON r.inventory_id = i.inventory_id
7  GROUP BY i.film_id
8  HAVING count > 1
9  ORDER BY count desc;
```

X
X

RESULT

	film_id	rental_date	count
▶	103	2005-05-25 12:30:15	34
	738	2005-05-25 21:46:54	33
	382	2005-05-25 05:12:29	32
	730	2005-05-25 06:04:08	32
	489	2005-05-25 10:52:13	32
	767	2005-05-26 16:19:46	32
	331	2005-05-27 15:51:30	32
	31	2005-05-25 01:59:46	31
	735	2005-05-25 17:30:42	31
	621	2005-05-25 18:28:09	31
	973	2005-05-25 19:27:51	31
	418	2005-05-27 06:43:59	31
	753	2005-05-27 07:37:02	31



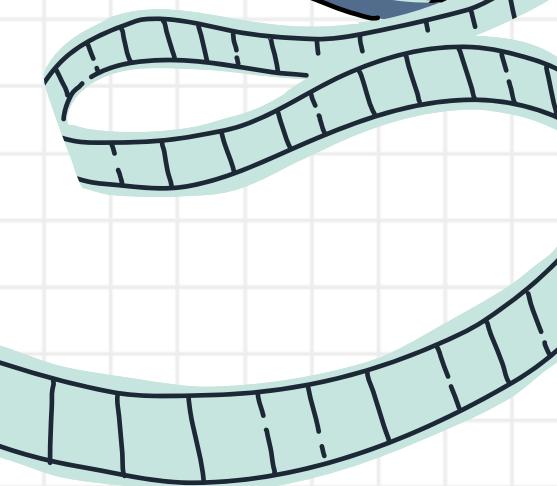
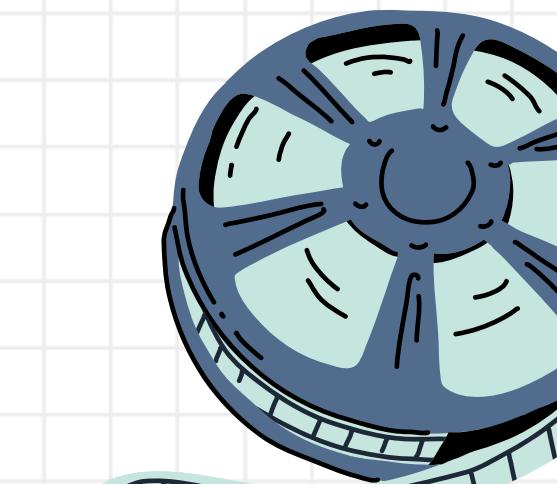
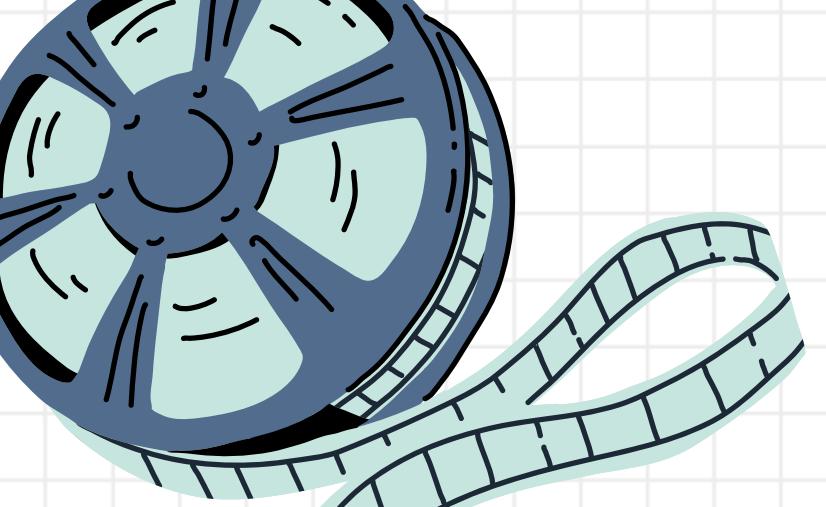
29. Calculate the total revenue for each staff member.

```
1 •   SELECT
2       s.first_name, s.last_name, p.revenue
3   FROM
4       staff s
5           JOIN
6       (SELECT
7           staff_id, SUM(amount) AS revenue
8   FROM
9       payment
10      GROUP BY staff_id
11      ORDER BY revenue) p ON s.staff_id = p.staff_id;
```

X
X

RESULT

	first_name	last_name	revenue
▶	Mike	Hillyer	33482.50
	Jon	Stephens	33924.06

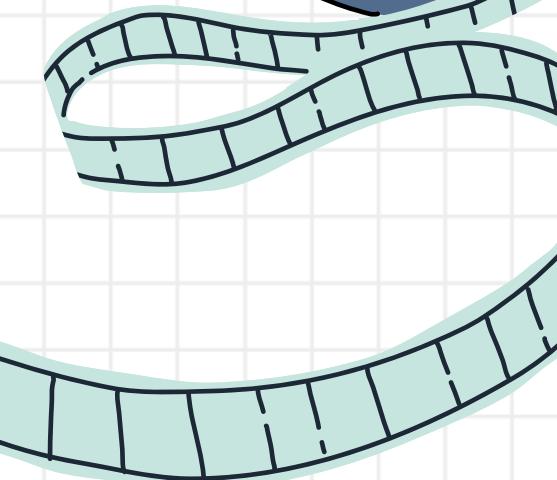
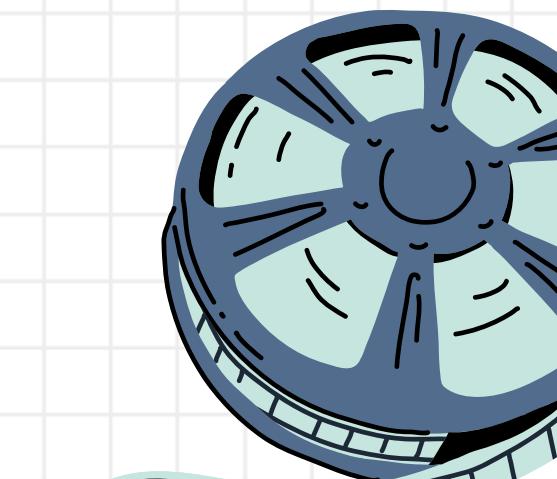
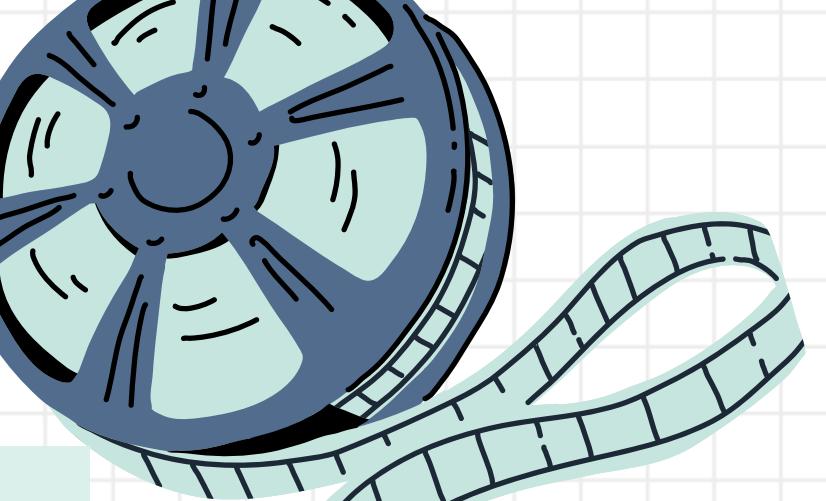


30. List the top 10 customers with the highest average rental durations.

```
1  SELECT
2      c.first_name, c.last_name, r.duration
3  FROM
4      customer c
5          RIGHT JOIN
6      (SELECT DISTINCT
7          (customer_id),
8              ROUND(AVG(DATEDIFF(return_date, rental_date)), 2) AS duration
9  FROM
10         rental
11     GROUP BY customer_id
12     ORDER BY duration DESC
13     LIMIT 10) r ON c.customer_id = r.customer_id;
```

RESULT

	first_name	last_name	duration
▶	KENNETH	GOODEN	6.44
	BRITTANY	RILEY	6.39
	KEVIN	SCHULER	6.36
	NEIL	RENNER	6.35
	MATHEW	BOLIN	6.32
	JESUS	MCCARTNEY	6.29
	MIGUEL	BETANCOURT	6.26
	CALVIN	MARTEL	6.26
	TROY	QUIGLEY	6.23
	KATHRYN	COLEMAN	6.12

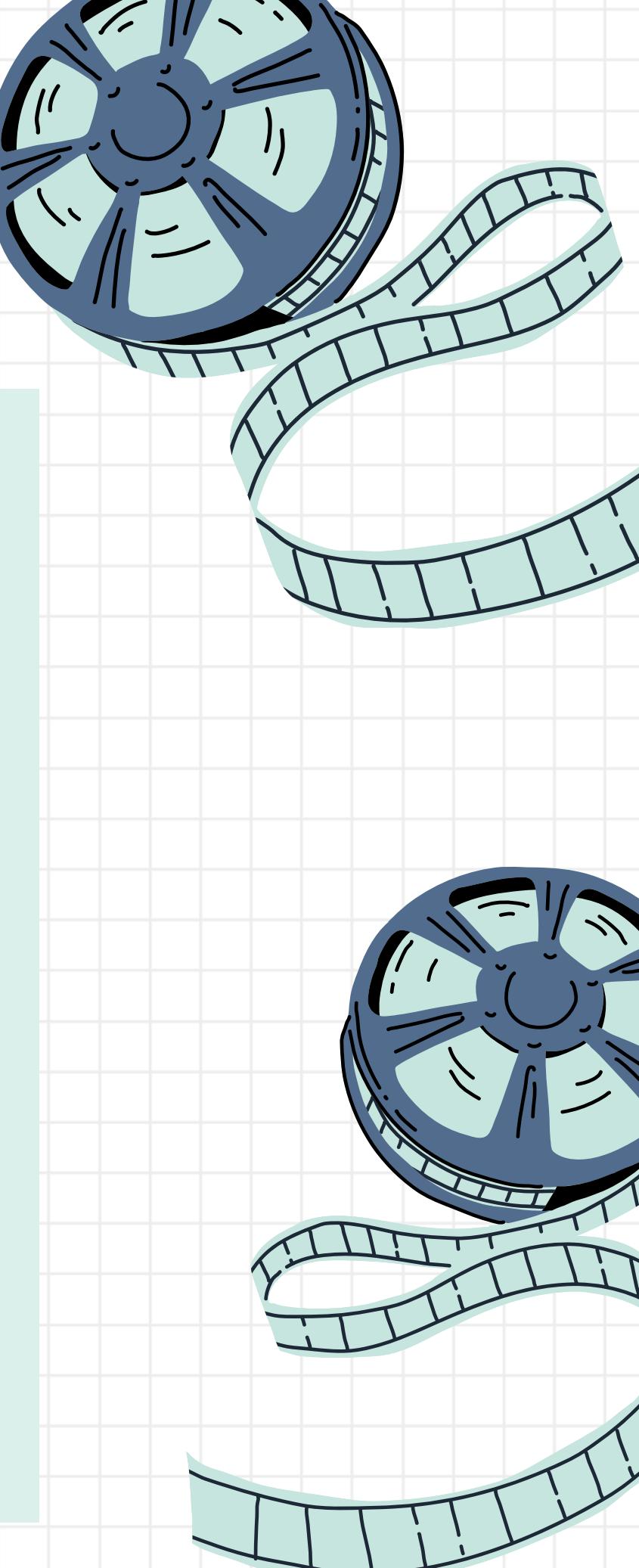


31. Find the customers who have rented the most films in each category.

```
1  SELECT
2      u.category_name,
3      c.first_name,
4      c.last_name
5  FROM
6      customer c
7  RIGHT JOIN
8      (
9          SELECT
10             cat.name AS category_name,
11             x.customer_id
12         FROM
13             category cat
14         RIGHT JOIN
15             (
16                 SELECT
17                     y.category_id,
18                     y.customer_id,
19                     MAX(y.count) AS count
20                 FROM
21                     (
22                         SELECT
23                             fc.category_id,
24                             m.customer_id,
25                             COUNT(m.customer_id) AS count
26                         FROM
27                             film_category fc
28             ) m
29             ON m.category_id = x.category_id
30             (
31                 SELECT
32                     r.customer_id,
33                     r.rental_id,
34                     i.film_id
35                 FROM
36                     rental r
37                 LEFT JOIN
38                     inventory i ON r.inventory_id = i.inventory_id
39             ) m
40             ON m.film_id = fc.film_id
41             GROUP BY
42                 fc.category_id,
43                 m.customer_id
44             )
45             GROUP BY
46                 y.category_id,
47                 y.customer_id
48             )
49             ) x
50             ON cat.category_id = x.category_id
51             ) u
52             ON c.customer_id = u.customer_id;
```

RESULT

	category_name	first_name	last_name
▶	Action	ROBERT	BAUGHMAN
	Animation	MINNIE	ROMERO
	Children	MANUEL	MURRELL
	Classics	CAROL	GARCIA
	Comedy	NELSON	CHRISTENSON
	Documentary	RONALD	WEINER
	Drama	DANNY	ISOM
	Family	CHARLOTTE	HUNTER



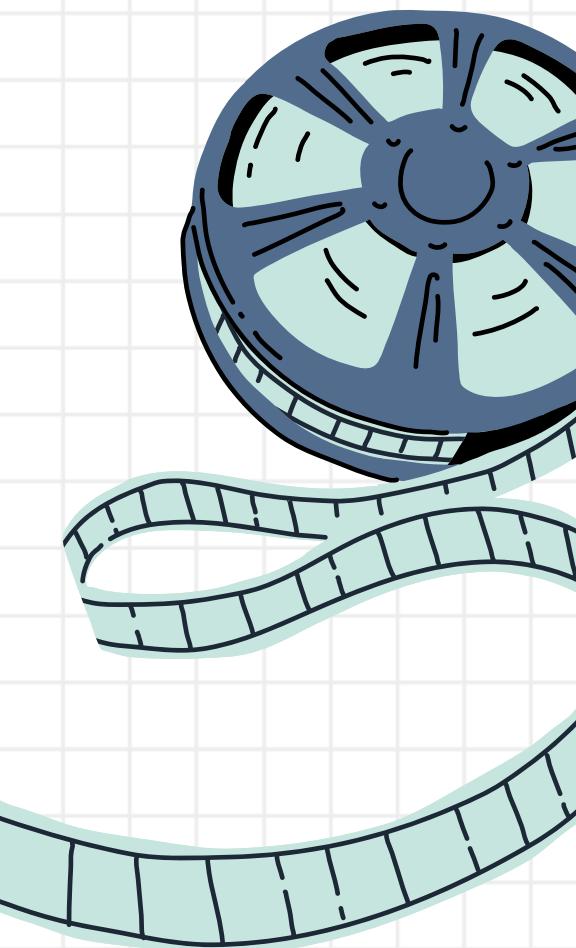
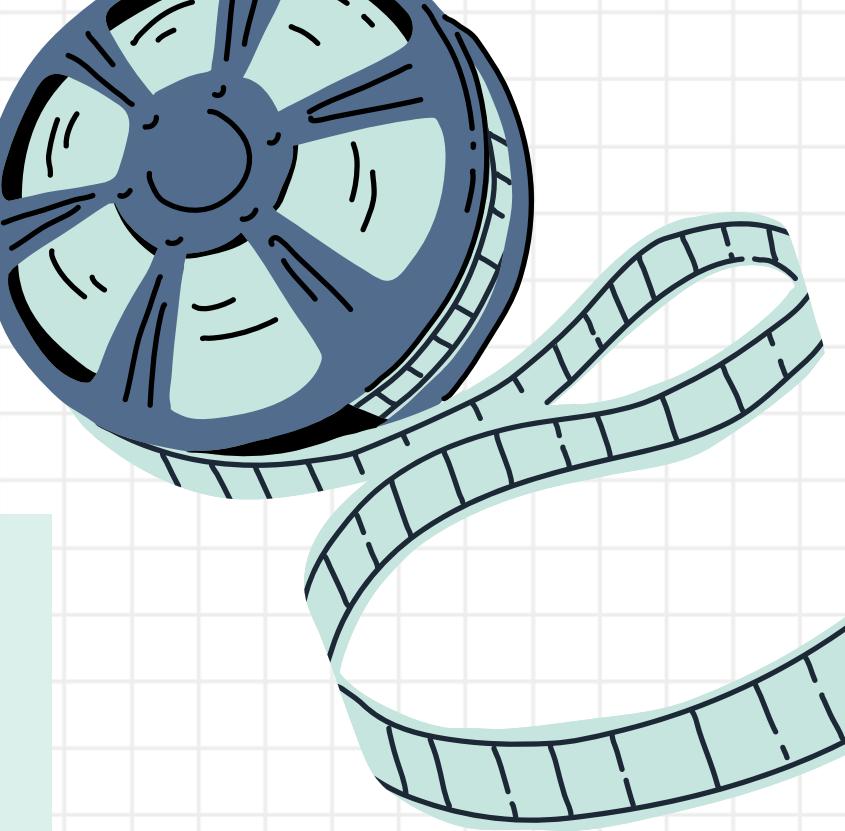
32. Retrieve the films that have been rented in the last 7 days.

```
1 •   SELECT
2       f.title
3   FROM
4       film f
5           RIGHT JOIN
6   (SELECT
7       i.film_id, x.ren_date
8   FROM
9       inventory i
10      RIGHT JOIN (SELECT
11          inventory_id, DATE(rental_date) AS ren_date
12      FROM
13          rental
14      WHERE
15          DATE(rental_date) >= DATE_SUB((SELECT
16              MAX(rental_date)
17          FROM
18              rental), INTERVAL 7 DAY)
19      GROUP BY inventory_id
20      ORDER BY ren_date DESC) x ON i.inventory_id = x.inventory_id) y ON f.film_id = y.film_id;
```

X
X

RESULT

	title
▶	ACE GOLDFINGER
	AFFAIR PREJUDICE
	AFRICAN EGG
	ALI FOREVER
	ALONE TRIP
	AMADEUS HOLY
	AMERICAN CIRCUS
	AMISTAD MIDSUMMER



Thank you ...

