

The background features a soft, abstract design. A large, light pink shape occupies the right side and bottom. A smaller, light blue shape is in the top left corner. A thin, wavy, golden-brown line curves across the left side of the image. The text 'MAVEN MOVIES' is centered horizontally across the middle of the image.

MAVEN MOVIES

List all details of actors.

```
select * from actor;
```

Result Grid

Filter Rows:

Edit:

	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
	7	GRACE	MOSTEL	2006-02-15 04:34:33
	8	MATTHEW	JOHANSSON	2006-02-15 04:34:33
	9	JOE	SWANK	2006-02-15 04:34:33
	10	CHRISTIAN	GABLE	2006-02-15 04:34:33


List all customer information from DB.

```
SELECT * FROM customer;
```

Result Grid									
Filter Rows:									
Edit: Export/Import: Wrap Cell Content:									
	customer_id	store_id	first_name	last_name	email	address_id	active	create_date	
▶	1	1	MARY	SMITH	MARY.SMITH@sakilacustomer.org	5	1	2006-02-14 22:04:36	2
	2	1	PATRICIA	JOHNSON	PAT PATRICIA.JOHNSON@sakilacustomer.org		1	2006-02-14 22:04:36	2
	3	1	LINDA	WILLIAMS	LINDA.WILLIAMS@sakilacustomer.org	7	1	2006-02-14 22:04:36	2
	4	2	BARBARA	JONES	BARBARA.JONES@sakilacustomer.org	8	1	2006-02-14 22:04:36	2
	5	1	ELIZABETH	BROWN	ELIZABETH.BROWN@sakilacustomer.org	9	1	2006-02-14 22:04:36	2
	6	2	JENNIFER	DAVIS	JENNIFER.DAVIS@sakilacustomer.org	10	1	2006-02-14 22:04:36	2
	7	1	MARIA	MILLER	MARIA.MILLER@sakilacustomer.org	11	1	2006-02-14 22:04:36	2
	8	2	SUSAN	WILSON	SUSAN.WILSON@sakilacustomer.org	12	1	2006-02-14 22:04:36	2
	9	2	MARGARET	MOORE	MARGARET.MOORE@sakilacustomer.org	13	1	2006-02-14 22:04:36	2







List different countries.

```
SELECT * FROM COUNTRY;
```

Result Grid  Filter Rows: <input type="text"/> Edit: <input type="text"/>			
	country_id	country	last_update
▶	1	Afghanistan	2006-02-15 04:44:00
	2	Algeria	2006-02-15 04:44:00
	3	American Samoa	2006-02-15 04:44:00
	4	Angola	2006-02-15 04:44:00
	5	Anguilla	2006-02-15 04:44:00
	6	Argentina	2006-02-15 04:44:00
	7	Armenia	2006-02-15 04:44:00
	8	Australia	2006-02-15 04:44:00
	9	Austria	2006-02-15 04:44:00
	10	Azerbaijan	2006-02-15 04:44:00

Display all active customers.

```
select * from customer
where active = 1;
```

Result Grid  Filter Rows: <input type="text"/> Edit:    Export/Import:   Wrap Cell Content							
	customer_id	store_id	first_name	last_name	email	address_id	active
▶	1	1	MARY	SMITH	MARY.SMITH@sakilacustomer.org	5	1
	2	1	PATRICIA	JOHNSON	PATRICIA.JOHNSON@sakilacustomer.org	6	1
	3	1	LINDA	WILLIAMS	LINDA.WILLIAMS@sakilacustomer.org	7	1
	4	2	BARBARA	JONES	BARBARA.JONES@sakilacustomer.org	8	1
	5	1	ELIZABETH	BROWN	ELIZABETH.BROWN@sakilacustomer.org	9	1
	6	2	JENNIFER	DAVIS	JENNIFER.DAVIS@sakilacustomer.org	10	1
	7	1	MARIA	MILLER	MARIA.MILLER@sakilacustomer.org	11	1
	8	2	SUSAN	WILSON	SUSAN.WILSON@sakilacustomer.org	12	1
	9	2	MARGARET	MOORE	MARGARET.MOORE@sakilacustomer.org	13	1

List of all rental IDs for customer with ID 1.

```
SELECT rental_id, customer_id  
FROM rental  
where customer_id = 1;
```

Result Grid | Filter Rows

	rental_id	customer_id
▶	76	1
	573	1
	1185	1
	1422	1
	1476	1
	1725	1
	2308	1
	2363	1
	3284	1
	4526	1



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

```
SELECT
    title, rental_duration
FROM
    film
WHERE
    rental_duration > 5;
```

Result Grid			Filter Rows:
	title	rental_duration	
▶	ACADEMY DINOSAUR	6	
	ADAPTATION HOLES	7	
	AFRICAN EGG	6	
	AIRPLANE SIERRA	6	
	AIRPORT POLLOCK	6	
	ALADDIN CALENDAR	6	
	ALAMO VIDEOTAPE	6	
	ALASKA PHANTOM	6	
	ALICE FANTASIA	6	
	ALLEY EVOLUTION	6	

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

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

Result Grid   Filter Rows: <input type="text"/> Export			
	title	replacement_cost	total_films
▶	ADAPTATION HOLES	18.99	214
	AGENT TRUMAN	17.99	214
	AIRPORT POLLOCK	15.99	214
	ALAMO VIDEOTAPE	16.99	214
	AMERICAN CIRCUS	17.99	214
	ANALYZE HOOSIERS	19.99	214
	ANGELS LIFE	15.99	214
	ANNIE IDENTITY	15.99	214
	ANTHEM LUKE	16.99	214
	APACHE DIVINE	16.99	214

Display the count of unique first names of actors.

```
select count(distinct first_name) from actor;
```

	count(distinct first_name)
▶	128

```
select * from customer
limit 10;
```

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

[illegible]







Display the names of the first 5 movies which are rated as 'G'.

```
select title, rating
from film
where rating like 'G%'
limit 5;
```

Result Grid			Filter Rows:
	title	rating	
▶	ACE GOLDFINGER	G	
	AFFAIR PREJUDICE	G	
	AFRICAN EGG	G	
	ALAMO VIDEOTAPE	G	
	AMISTAD MIDSUMMER	G	

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

```
select * from customer
where first_name like 'a%';
```

Result Grid  Filter Rows: <input type="text"/> Edit:    Export/Import:   Wrap Cell Content: <input type="checkbox"/>							
	customer_id	store_id	first_name	last_name	email	address_id	active
▶	29	2	ANGELA	HERNANDEZ	ANGELA.HERNANDEZ@sakilacustomer.org	33	1
	32	1	AMY	LOPEZ	AMY.LOPEZ@sakilacustomer.org	36	1
	33	2	ANNA	HILL	ANNA.HILL@sakilacustomer.org	37	1
	40	2	AMANDA	CARTER	AMANDA.CARTER@sakilacustomer.org	44	1
	48	1	ANN	EVANS	ANN.EVANS@sakilacustomer.org	52	1
	51	1	ALICE	STEWART	ALICE.STEWART@sakilacustomer.org	55	1
	63	1	ASHLEY	RICHARDSON	ASHLEY.RICHARDSON@sakilacustomer.org	67	1
	81	1	ANDREA	HENDERSON	ANDREA.HENDERSON@sakilacustomer.org	85	1
	85	2	ANNE	POWELL	ANNE.POWELL@sakilacustomer.org	89	1
	97	2	ANNIE	RUSSELL	ANNIE.RUSSELL@sakilacustomer.org	101	1
	136	2	ANITA	MORALES	ANITA.MORALES@sakilacustomer.org	140	1
	139	1	AMBER	DIXON	AMBER.DIXON@sakilacustomer.org	143	1
	142	1	APRIL	BURNS	APRIL.BURNS@sakilacustomer.org	146	1
	152	1	ALICIA	MILLS	ALICIA.MILLS@sakilacustomer.org	156	1
	173	1	AUDREY	RAY	AUDREY.RAY@sakilacustomer.org	177	1

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

```
select * from customer  
where first_name like '%a';
```

customer_id	store_id	first_name	last_name
2	1	PATRICIA	JOHNSON
3	1	LINDA	WILLIAMS
4	2	BARBARA	JONES
7	1	MARIA	MILLER
11	2	LISA	ANDERSON
16	2	SANDRA	MARTIN
17	1	DONNA	THOMPSON
22	1	LAURA	RODRIGUEZ
26	2	JESSICA	HALL
28	1	CYNTHIA	YOUNG
29	2	ANGELA	HERNANDEZ

Display the list of first 4 cities which start and end with 'a'


```
select * from city
where city like 'a%a'
limit 4;
```

Result Grid					Filter Rows:		Edit:
	city_id	city	country_id	last_update			
▶	2	Abha	82	2006-02-15 04:45:25			
	4	Acua	60	2006-02-15 04:45:25			
	5	Adana	97	2006-02-15 04:45:25			
	6	Addis Abeba	31	2006-02-15 04:45:25			
✱	NULL	NULL	NULL	NULL			

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



```
SELECT customer_id, first_name, last_name
FROM customer
WHERE first_name LIKE '%NI%';
```

Result Grid  Filter Rows: <input data-bbox="2698 637 3048 727" type="text"/>			
	customer_id	first_name	last_name
▶	6	JENNIFER	DAVIS
	35	VIRGINIA	GREEN
	41	STEPHANIE	MITCHELL
	66	JANICE	WARD
	68	NICOLE	PETERSON
	74	DENISE	KELLY
	88	BONNIE	HUGHES
	97	ANNIE	RUSSELL
	106	CONNIE	WALLACE
	131	MONICA	HICKS
	135	JUANITA	MASON
	136	ANITA	MORALES

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

```
SELECT first_name  
FROM customer  
WHERE first_name LIKE '_R%';
```

	first_name
▶	BRENDA
	FRANCES
	IRENE
	CRYSTAL
	TRACY
	GRACE
	ERIN
	ERICA
	BRITTANY
	KRISTEN
	KRISTIN
	ARLENE
	IRMA
	PRISCILLA
	BRANDY
	TRACEY

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

```
SELECT *  
FROM customer  
WHERE first_name LIKE 'A%'  
AND LENGTH(first_name) >= 5;
```

Result Grid				
Filter Rows:				
	customer_id	store_id	first_name	last_name
▶	29	2	ANGELA	HERNANDEZ
	40	2	AMANDA	CARTER
	51	1	ALICE	STEWART
	63	1	ASHLEY	RICHARDSON
	81	1	ANDREA	HENDERSON
	97	2	ANNIE	RUSSELL
	136	2	ANITA	MORALES
	139	1	AMBER	DIXON
	142	1	APRIL	BURNS
	152	1	ALICIA	MILLS
	173	1	AUDREY	RAY
	175	1	ANNETTE	OLSON
	217	2	AGNES	BISHOP
	225	1	ARLENE	HARVEY



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

```
SELECT first_name FROM customer  
WHERE first_name LIKE 'A%O';
```

Result Grid	
	first_name
▶	ANTONIO
	ARMANDO
	ALFREDO
	ALBERTO

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

```
select title, rating
from film
where rating in ('pg', 'pg-13');
```

Result Grid   Filter Rows: <input type="text"/>		
	title	rating
▶	ACADEMY DINOSAUR	PG
	AGENT TRUMAN	PG
	AIRPLANE SIERRA	PG-13
	ALABAMA DEVIL	PG-13
	ALASKA PHANTOM	PG
	ALI FOREVER	PG
	ALTER VICTORY	PG-13
	AMADEUS HOLY	PG
	ANTHEM LUKE	PG-13
	APOLLO TEEN	PG-13
	ARACHNOPHOBIA R...	PG-13
	ARGONAUTS TOWN	PG-13
	ARIZONA BANG	PG
	ARSENIC INDEPEND...	PG
	ATTACKS HATE	PG-13
	ATTRACTION NEWTON	PG-13

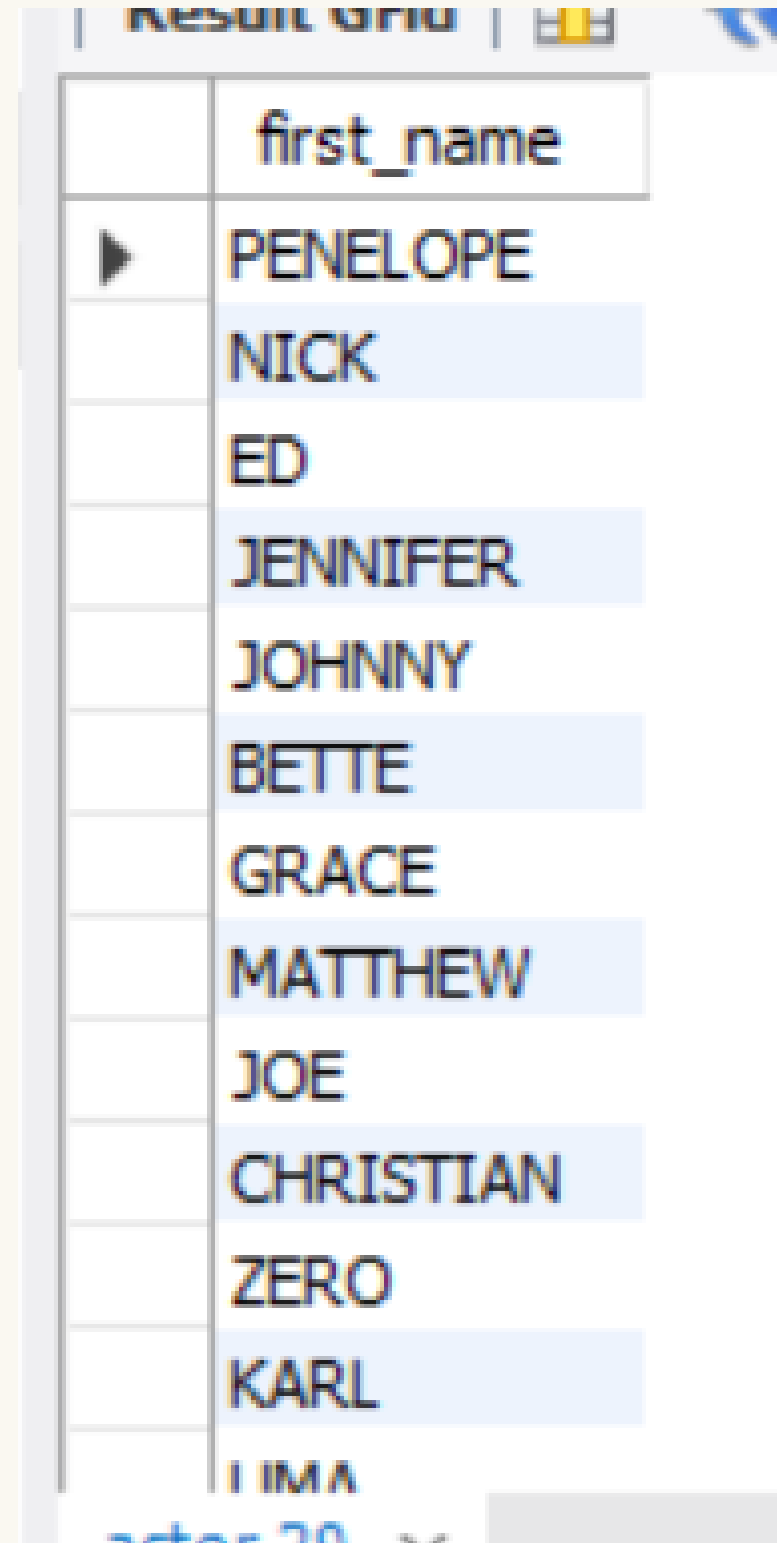
Get the films with length between 50 to 100 using between operator.

```
select title, length from film
where length between 50 and 100;
```

Result Grid			Filter Rows:
	title	length	
▶	ACADEMY DINOSAUR	86	
	ADAPTATION HOLES	50	
	AIRPLANE SIERRA	62	
	AIRPORT POLLOCK	54	
	ALADDIN CALENDAR	63	
	ALICE FANTASIA	94	
	ALONE TRIP	82	
	ALTER VICTORY	57	
	AMELIE HELLFIGHTERS	79	
	AMISTAD MIDSUMMER	85	
	ANACONDA CONFESS...	92	
	ANGELS LIFE	74	
	ANINTE IDENTTV	86	

Get the top 50 actors using limit operator.

```
select first_name from actor  
limit 50;
```



A screenshot of a database application window titled "Result Grid". It displays a table with one column, "first_name", and 10 rows of data. The rows are: PENELOPE, NICK, ED, JENNIFER, JOHNNY, BETTE, GRACE, MATTHEW, JOE, and CHRISTIAN. The rows are alternatingly highlighted with light blue and white backgrounds. Below the visible rows, the text "ZERO" and "KARL" are partially visible, indicating the list continues. At the bottom of the window, a status bar shows "actor 20" and a scroll bar.

	first_name
▶	PENELOPE
	NICK
	ED
	JENNIFER
	JOHNNY
	BETTE
	GRACE
	MATTHEW
	JOE
	CHRISTIAN
	ZERO
	KARL
	LIMA

Get the distinct film ids from
inventory table.

```
select distinct film_id from inventory;
```

Result Grid	
	film_id
▶	1
	2
	3
	4
	5
	6
	7
	8
	9
	10
	11
	12
	13
	15
	16

FUNCTIONS

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

```
select count(*) as total_rentals  
from rental;
```

Result Grid	
	total_rentals
▶	16044

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

```
SELECT  
    ROUND(AVG(rental_duration), 2) AS rental_duration  
FROM  
    film;
```

Result Grid	
	rental_duration
▶	4.99


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

```
SELECT UPPER(first_name) AS upper_first_name,  
       UPPER(last_name) AS upper_last_name  
FROM customer;
```

Result Grid			Filter Rows:
	upper_first_name	upper_last_name	
▶	MARY	SMITH	
	PATRICIA	JOHNSON	
	LINDA	WILLIAMS	
	BARBARA	JONES	
	ELIZABETH	BROWN	
	JENNIFER	DAVIS	
	MARIA	MILLER	
	SUSAN	WILSON	
	MARGARET	MOORE	
	DOROTHY	TAYLOR	
	LISA	ANDERSON	
	NANCY	THOMAS	
	KAREN	JACKSON	
	BETTY	WHITE	

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

```
SELECT rental_id,  
       MONTH(rental_date) AS rental_month  
FROM rental;
```



	rental_id	rental_month
▶	1	5
	2	5
	3	5
	4	5
	5	5
	6	5
	7	5
	8	5
	9	5
	10	5
	11	5

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

```
select customer_id, count(*) as rental_count  
from rental  
group by customer_id;
```

Result Grid			Filter Rows:	
	rental_id	rental_month		
▶	1	5		
	2	5		
	3	5		
	4	5		
	5	5		
	6	5		
	7	5		
	8	5		
	9	5		
	10	5		
	11	5		

Find the total revenue generated by each store.


```
select rental_id, sum(amount) from payment  
group by rental_id;
```

Result Grid | Filter Rows:

	rental_id	sum(amount)
▶	NULL	9.95
	1	2.99
	2	2.99
	3	3.99
	4	4.99
	5	6.99
	6	0.99
	7	1.99
	8	4.99
	9	4.99
	10	5.99
	11	8.99
	12	4.99

Determine the total number of rentals
for each category of movies.

```
SELECT c.name AS category_name,  
       COUNT(r.rental_id) AS total_rentals  
FROM rental r  
JOIN inventory i ON r.inventory_id = i.inventory_id  
JOIN film f ON i.film_id = f.film_id  
JOIN film_category fc ON f.film_id = fc.film_id  
JOIN category c ON fc.category_id = c.category_id  
GROUP BY c.name  
ORDER BY total_rentals DESC;
```



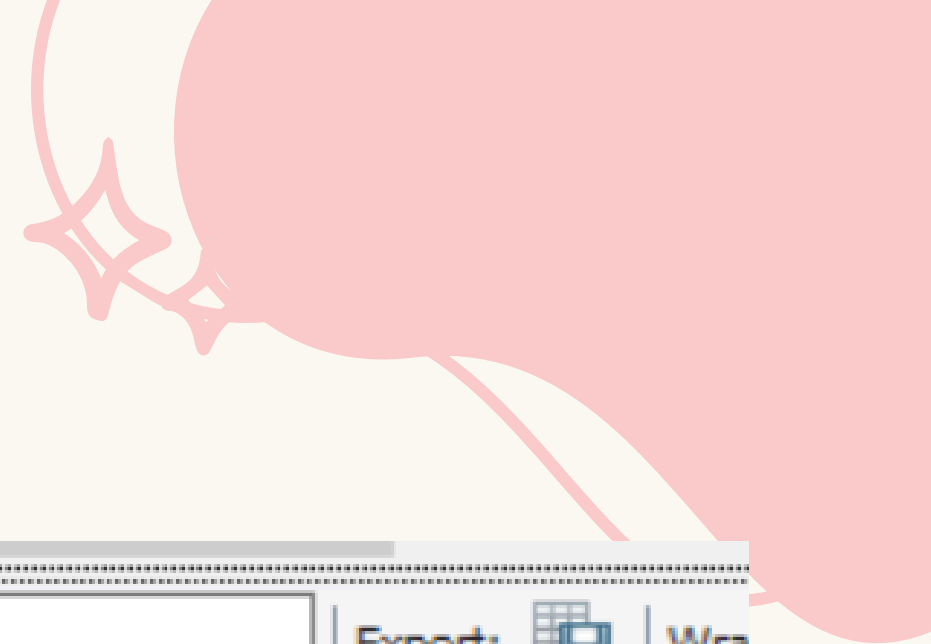
	category_name	total_rentals
	Animation	1166
	Action	1112
	Sci-Fi	1101
	Family	1096
	Drama	1060
	Documentary	1050
	Foreign	1033
	Games	969
	Children	945
	Comedy	941
	New	940
	Classics	939
	Horror	846
	Travel	837
	Music	830

Find the average rental rate of movies in each language.

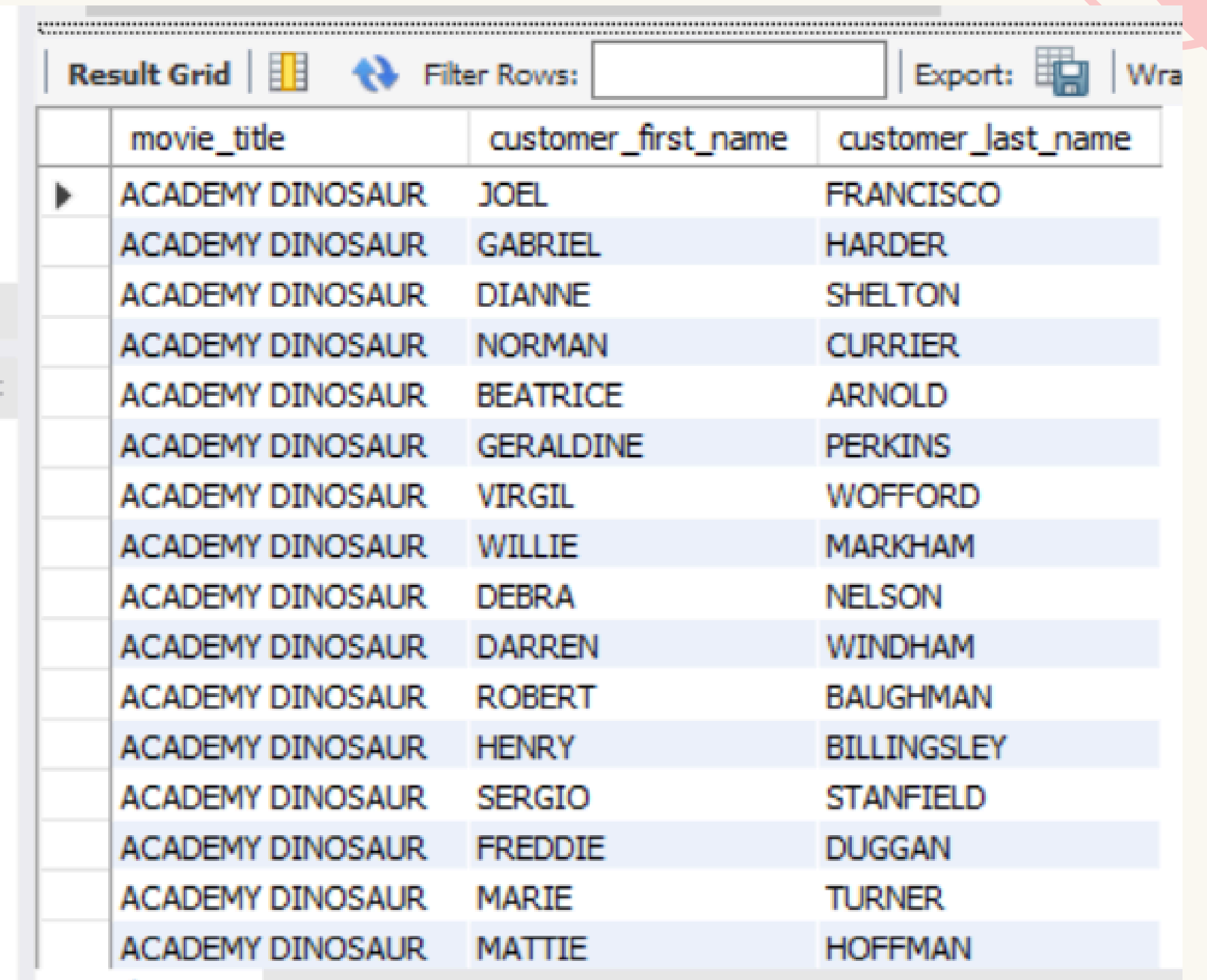
```
SELECT l.name AS language_name,  
       AVG(f.rental_rate) AS average_rental_rate  
FROM film f  
JOIN language l ON f.language_id = l.language_id  
GROUP BY l.name  
ORDER BY average_rental_rate DESC;
```

Result Grid			Filter Rows:
	language_name	average_rental_rate	
▶	English	2.980000	

Display the title of the movie, customer's first name, and last name who rented it.

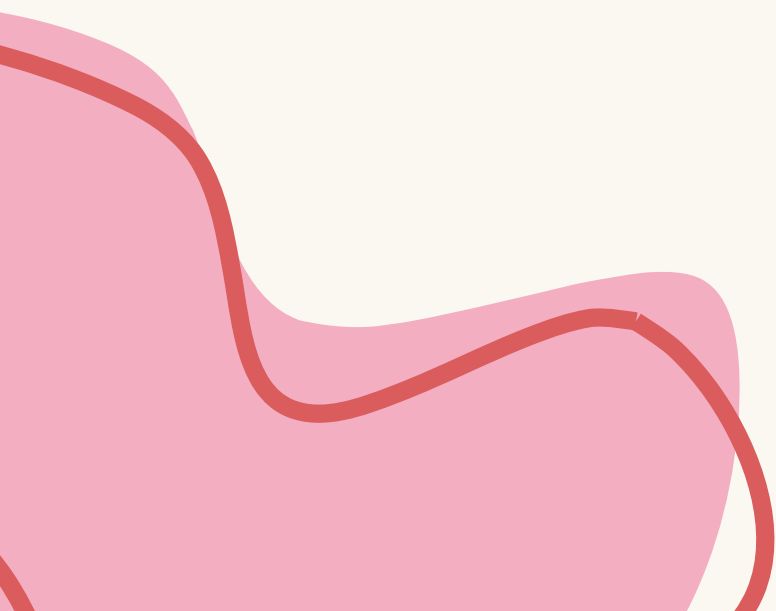


```
SELECT f.title AS movie_title,  
       c.first_name AS customer_first_name,  
       c.last_name AS customer_last_name  
FROM rental r  
JOIN inventory i ON r.inventory_id = i.inventory_id  
JOIN film f ON i.film_id = f.film_id  
JOIN customer c ON r.customer_id = c.customer_id;
```



The screenshot shows a database interface with a 'Result Grid' tab. It displays the results of the SQL query, showing 15 rows of data. The columns are 'movie_title', 'customer_first_name', and 'customer_last_name'. All movie titles are 'ACADEMY DINOSAUR'. The customers listed are Joel Francisco, Gabriel Harder, Dianne Shelton, Norman Currier, Beatrice Arnold, Geraldine Perkins, Virgil Wofford, Willie Markham, Debra Nelson, Darren Windham, Robert Baughman, Henry Billingsley, Sergio Stanfield, Freddie Duggan, Marie Turner, and Mattie Hoffman.

	movie_title	customer_first_name	customer_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
	ACADEMY DINOSAUR	FREDDIE	DUGGAN
	ACADEMY DINOSAUR	MARIE	TURNER
	ACADEMY DINOSAUR	MATTIE	HOFFMAN






Retrieve the customer names along with the total amount they've spent on rentals.

```
SELECT c.first_name,
       c.last_name,
       SUM(p.amount) AS total_spent
FROM customer c
JOIN payment p ON c.customer_id = p.customer_id
GROUP BY c.customer_id, c.first_name, c.last_name
ORDER BY total_spent DESC;
```

Result Grid			
Filter Rows:			
	first_name	last_name	total_spent
▶	KARL	SEAL	221.55
	ELEANOR	HUNT	216.54
	CLARA	SHAW	195.58
	RHONDA	KENNEDY	194.61
	MARION	SNYDER	194.61
	TOMMY	COLLAZO	186.62
	WESLEY	BULL	177.60
	TIM	CARY	175.61
	MARCIA	DEAN	175.58
	ANA	BRADLEY	174.66
	JUNE	CARROLL	173.63
	LENA	JENSEN	170.67
	DIANE	COLLINS	169.65
	ARNOLD	HAVENS	167.67
	CURTIS	IRBY	167.62

List the titles of movies rented by each customer in a particular city (e.g., 'London').

```
SELECT c.first_name AS customer_first_name,  
       c.last_name AS customer_last_name,  
       f.title AS movie_title,  
       ci.city AS city_name  
FROM customer c  
JOIN address a ON c.address_id = a.address_id  
JOIN city ci ON a.city_id = ci.city_id  
JOIN rental r ON c.customer_id = r.customer_id  
JOIN inventory i ON r.inventory_id = i.inventory_id  
JOIN film f ON i.film_id = f.film_id  
WHERE ci.city = 'London'  
GROUP BY c.customer_id, f.title  
ORDER BY c.last_name, c.first_name;
```

Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 				
	customer_first_name	customer_last_name	movie_title	city_name
▶	MATTIE	HOFFMAN	CONQUERER NUTS	London
	MATTIE	HOFFMAN	WRATH MILE	London
	MATTIE	HOFFMAN	COLDBLOODED DARLING	London
	MATTIE	HOFFMAN	DARKNESS WAR	London
	MATTIE	HOFFMAN	WATERSHIP FRONTIER	London
	MATTIE	HOFFMAN	JAWBREAKER BROOKLYN	London
	MATTIE	HOFFMAN	TIGHTS DAWN	London
	MATTIE	HOFFMAN	CHICKEN HELLFIGHTERS	London
	MATTIE	HOFFMAN	DOOM DANCING	London
	MATTIE	HOFFMAN	PITY BOUND	London
	MATTIE	HOFFMAN	EMPIRE MALKOVICH	London
	MATTIE	HOFFMAN	ACADEMY DINOSAUR	London
	MATTIE	HOFFMAN	FACTORY DRAGON	London
	MATTIE	HOFFMAN	VELVET TERMINATOR	London
	MATTIE	HOFFMAN	TRACY CIDER	London
	MATTIE	HOFFMAN	FROGMEN BREAKING	London

Display the top 5 rented movies along with the number of times they've been rented.

```
SELECT f.title AS movie_title,  
       COUNT(r.rental_id) AS rental_count  
FROM rental r  
JOIN inventory i ON r.inventory_id = i.inventory_id  
JOIN film f ON i.film_id = f.film_id  
GROUP BY f.title  
ORDER BY rental_count DESC  
LIMIT 5;
```

Result Grid			Filter Rows:
	movie_title	rental_count	
▶	BUCKET BROTHERHOOD	34	
	ROCKETEER MOTHER	33	
	FORWARD TEMPLE	32	
	GRIT CLOCKWORK	32	
	JUGGLER HARDLY	32	

Determine the customers who have rented movies from both stores (store ID 1 and store ID 2).

```
SELECT c.customer_id,  
       c.first_name,  
       c.last_name  
FROM rental r  
JOIN inventory i ON r.inventory_id = i.inventory_id  
JOIN customer c ON r.customer_id = c.customer_id  
WHERE i.store_id IN (1, 2)  
GROUP BY c.customer_id, c.first_name, c.last_name  
HAVING COUNT(DISTINCT i.store_id) = 2;
```

Result Grid			
Filter Rows:			
	customer_id	first_name	last_name
▶	1	MARY	SMITH
	2	PATRICIA	JOHNSON
	3	LINDA	WILLIAMS
	4	BARBARA	JONES
	5	ELIZABETH	BROWN
	6	JENNIFER	DAVIS
	7	MARIA	MILLER
	8	SUSAN	WILSON
	9	MARGARET	MOORE
	10	DOROTHY	TAYLOR
	11	LISA	ANDERSON
	12	NANCY	THOMAS
	13	KAREN	JACKSON
	14	BETTY	WHITE
	15	HELEN	HARRIS
	16	SANDRA	MARTIN





WINDOWS FUNCTION






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

```
SELECT
  c.customer_id,
  c.first_name,
  c.last_name,
  SUM(p.amount) AS total_spent,
  RANK() OVER (ORDER BY SUM(p.amount) DESC) AS customer_rank
FROM
  customer c
JOIN
  rental r ON c.customer_id = r.customer_id
JOIN
  payment p ON r.rental_id = p.rental_id
GROUP BY
  c.customer_id, c.first_name, c.last_name
ORDER BY
  total_spent DESC;
```

Result Grid  Filter Rows: <input type="text"/> Export:  Wrap					
	customer_id	first_name	last_name	total_spent	customer_rank
▶	526	KARL	SEAL	221.55	1
	148	ELEANOR	HUNT	216.54	2
	144	CLARA	SHAW	195.58	3
	137	RHONDA	KENNEDY	194.61	4
	178	MARION	SNYDER	194.61	4
	459	TOMMY	COLLAZO	186.62	6
	469	WESLEY	BULL	177.60	7
	468	TIM	CARY	175.61	8
	236	MARCIA	DEAN	175.58	9
	181	ANA	BRADLEY	174.66	10
	176	JUNE	CARROLL	173.63	11
	50	DIANE	COLLINS	169.65	12
	259	LENA	JENSEN	168.68	13
	522	ARNOLD	HAVENS	167.67	14
	410	CURTIS	IRBY	167.62	15
	403	MIKE	WAY	166.65	16

Calculate the cumulative revenue generated by each film over time.

```
SELECT
  f.title AS movie_title,
  p.payment_date,
  p.amount,
  SUM(p.amount) OVER (
    PARTITION BY f.film_id
    ORDER BY p.payment_date
    ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
  ) AS cumulative_revenue
FROM
  payment p
JOIN
  rental r ON p.rental_id = r.rental_id
JOIN
  inventory i ON r.inventory_id = i.inventory_id
JOIN
  film f ON i.film_id = f.film_id
ORDER BY
  f.title,
  p.payment_date;
```

Result Grid   Filter Rows: <input type="text"/> Export:  Wrap Cell Cont				
	movie_title	payment_date	amount	cumulative_revenue
▶	ACADEMY DINOSAUR	2005-05-27 07:03:28	0.99	0.99
	ACADEMY DINOSAUR	2005-05-30 20:21:07	1.99	2.98
	ACADEMY DINOSAUR	2005-06-15 2005-06-15 02:57:51		3.97
	ACADEMY DINOSAUR	2005-06-17 20:24:00	0.99	4.96
	ACADEMY DINOSAUR	2005-06-21 00:30:26	1.99	6.95
	ACADEMY DINOSAUR	2005-07-07 10:41:31	0.99	7.94
	ACADEMY DINOSAUR	2005-07-07 20:59:06	0.99	8.93
	ACADEMY DINOSAUR	2005-07-08 19:03:15	0.99	9.92
	ACADEMY DINOSAUR	2005-07-10 13:07:31	0.99	10.91
	ACADEMY DINOSAUR	2005-07-27 07:51:11	0.99	11.90
	ACADEMY DINOSAUR	2005-07-29 09:41:38	1.99	13.89
	ACADEMY DINOSAUR	2005-07-30 22:02:34	1.99	15.88
	ACADEMY DINOSAUR	2005-07-31 21:36:07	0.99	16.87
	ACADEMY DINOSAUR	2005-07-31 22:08:29	0.99	17.86
	ACADEMY DINOSAUR	2005-08-02 00:47:19	0.99	18.85
	ACADEMY DINOSAUR	2005-08-02 20:13:10	3.99	22.84

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

```
SELECT
  f.title AS movie_title,
  f.length AS film_length,
  AVG(f.rental_duration) OVER (PARTITION BY f.length) AS avg_rental_duration_for_length
FROM
  film f
ORDER BY
  f.length, f.title;
```

Result Grid				Filter Rows:	Export:	Wrap C
	movie_title	film_length	avg_rental_duration_for_length			
▶	ALIEN CENTER	46	5.2000			
	IRON MOON	46	5.2000			
	KWAI HOMEWARD	46	5.2000			
	LABYRINTH LEAGUE	46	5.2000			
	RIDGEMONT SUBMARINE	46	4.7143			
	DIVORCE SHINING	47	4.7143			
	DOWNHILL ENOUGH	47	4.7143			
	HALLOWEEN NUTS	47	4.7143			
	HANOVER GALAXY	47	4.7143			
	HAWK CHILL	47	4.7143			
	SHANGHAI TYCOON	47	4.7143			
	SUSPECTS QUILLS	47	4.7143			
	ACE GOLDFINGER	48	5.2727			
	HEAVEN FREEDOM	48	5.2727			
	MIDSUMMER GROUNDHOG	48	5.2727			
	NOTTING SPEAKEASY	48	5.2727			



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

```
SELECT
  c.customer_id,
  CONCAT(c.first_name, ' ', c.last_name) AS customer_name,
  COUNT(r.rental_id) AS total_rentals,
  AVG(COUNT(r.rental_id)) OVER() AS avg_rentals_across_all_customers,
  COUNT(r.rental_id) - AVG(COUNT(r.rental_id)) OVER () AS rental_difference
FROM
  customer c
JOIN
  rental r ON c.customer_id = r.customer_id
GROUP BY
  c.customer_id, c.first_name, c.last_name
ORDER BY
  rental_difference DESC;
```

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:					
	customer_id	customer_name	total_rentals	avg_rentals_across_all_customers	rental_difference
▶	148	ELEANOR HUNT	46	26.7846	19.2154
	526	KARL SEAL	45	26.7846	18.2154
	144	CLARA SHAW	42	26.7846	15.2154
	236	MARCIA DEAN	42	26.7846	15.2154
	75	TAMMY SANDERS	41	26.7846	14.2154
	197	SUE PETERS	40	26.7846	13.2154
	469	WESLEY BULL	40	26.7846	13.2154
	137	RHONDA KENNEDY	39	26.7846	12.2154
	178	MARION SNYDER	39	26.7846	12.2154
	468	TIM CARY	39	26.7846	12.2154
	5	ELIZABETH BROWN	38	26.7846	11.2154

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

```
SELECT
    DATE_FORMAT(payment_date, '%Y-%m') AS month,
    SUM(amount) AS total_revenue
FROM
    payment
GROUP BY
    DATE_FORMAT(payment_date, '%Y-%m')
ORDER BY
    month;
```

Result Grid |   Filter Rows:

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

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

```
WITH CategoryRentalCounts AS (  
  SELECT  
    c.name AS category_name,  
    f.title AS film_title,  
    COUNT(r.rental_id) AS rental_count  
  FROM category c  
  JOIN film_category fc ON c.category_id = fc.category_id  
  JOIN film f ON fc.film_id = f.film_id  
  JOIN inventory i ON f.film_id = i.film_id  
  JOIN rental r ON i.inventory_id = r.inventory_id  
  GROUP BY  
    c.name, f.title  
)  
SELECT  
  category_name,  
  film_title,  
  rental_count,  
  SUM(rental_count) OVER (PARTITION BY category_name ORDER BY rental_count DESC) AS running_total  
FROM CategoryRentalCounts  
ORDER BY category_name, running_total;
```

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:				
	category_name	film_title	rental_count	running_total
▶	Action	RUGRATS SHAKESPEARE	30	60
	Action	SUSPECTS QUILLS	30	60
	Action	STORY SIDE	28	144
	Action	HANDICAP BOONDOCK	28	144
	Action	TRIP NEWTON	28	144
	Action	PRIMARY GLASS	27	171
	Action	FANTASY TROOPERS	26	223
	Action	STAGECOACH ARMAGEDDON	26	223
	Action	CLUELESS BUCKET	25	273
	Action	HILLS NEIGHBORS	25	273
	Action	MOCKINGBIRD HOLLYWOOD	24	345
	Action	CELEBRITY HORN	24	345
	Action	DRIFTER COMMANDMENTS	24	345
	Action	FOOL MOCKINGBIRD	23	437
	Action	WOMEN DORADO	23	437
	Action	TRUMAN CRAZY	23	437



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

```
WITH CategoryRentalCounts AS (  
    SELECT c.name AS category_name, f.title AS film_title,  
           COUNT(r.rental_id) AS rental_count  
    FROM category c  
    JOIN film_category fc ON c.category_id = fc.category_id  
    JOIN film f ON fc.film_id = f.film_id  
    JOIN inventory i ON f.film_id = i.film_id  
    JOIN rental r ON i.inventory_id = r.inventory_id  
    GROUP BY c.name, f.title  
)  
  
CategoryAverageRental AS (  
    SELECT category_name, AVG(rental_count) AS avg_rental_count  
    FROM CategoryRentalCounts  
    GROUP BY category_name  
)  
  
SELECT crc.category_name, crc.film_title, crc.rental_count  
FROM CategoryRentalCounts crc  
JOIN CategoryAverageRental car ON crc.category_name = car.category_name  
WHERE crc.rental_count < car.avg_rental_count  
ORDER BY crc.category_name, crc.rental_count;
```

	category_name	film_title	rental_count
▶	Action	GOSFORD DONNIE	8
	Action	PARK CITIZEN	8
	Action	MONTEZUMA COMMAND	9
	Action	CASUALTIES ENCINO	9
	Action	ANTITRUST TOMATOES	10
	Action	GRAIL FRANKENSTEIN	10
	Action	DRAGON SQUAD	11
	Action	DARKO DORADO	11
	Action	LORD ARIZONA	11
	Action	MAGNOLIA FORRESTER	11
	Action	CROW GREASE	12
	Action	SPEAKEASY DATE	12
	Action	SIDE ARK	12
	Action	LAWRENCE LOVE	13
	Action	WEREWOLF LOLA	14
	Action	DANCES NONE	14

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

```
SELECT
    DATE_FORMAT(payment_date, '%Y-%m') AS month,
    SUM(amount) AS total_revenue
FROM
    payment
GROUP BY
    month
ORDER BY
    total_revenue DESC
LIMIT 5;
```

Result Grid |   Filter Rows:

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