





SQL File 5* SQL File 6* SQL File 8* SQL File 9* SQL File 7*

 Limit to 1000 rows

1 • `SELECT first_name, last_name FROM sakila.actor;`

Result Grid   Filter Rows: Export:  Wrap Cell C

	first_name	last_name
	PENELOPE	GUINNESS
	NICK	WAHLBERG
	ED	CHASE
	JENNIFER	DAVIS
	JOHNNY	LOLLOBRIGIDA
	BETTE	NICHOLSON
	GRACE	MOSTEL
	MATTHEW	JOHANSSON
	JOE	SWANK
	CHRISTIAN	GABLE
	ZERO	CAGE
	KARL	BERRY
	UMA	WOOD
	VIVIEN	BERGEN
	CUBA	OLIVIER
	FRED	COSTNER
	HELEN	VOIGHT
	DAN	TORN
	BOB	FAWCETT
	LUCILLE	TRACY
	KIRSTEN	PALTROW
	ELVIS	MARX
	SANDRA	KILMER

actor 2 ×

SQL File 5* SQL File 6* SQL File 8* SQL File 9* SQL File 7* IceCreamSchema* su

Limit to 1000 rows

```
1
2 • SELECT CONCAT(first_name, " ", last_name, " ") AS Actor_name
3 FROM actor;
```

Result Grid

Actor_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
UMA WOOD
VIVIEN BERGEN
CUBA OLIVIER
FRED COSTNER
HELEN VOIGHT
DAN TORN

Result 5 x

SQL File 5* SQL File 6* SQL File 8* SQL File 9* SQL File 7* IceCreamSchema* surfers x Unsolved

Limit to 1000 rows

```
1 • SELECT actor_id, first_name, last_name FROM sakila.actor WHERE first_name LIKE "Joe";
```

Result Grid

actor_id	first_name	last_name
9	JOE	SWANK
NULL	NULL	NULL

SQL File 5* SQL File 6* SQL File 8* SQL File 9* SQL File 7* IceCreamSchema*

Limit to 1000 rows

```
1 • SELECT * FROM sakila.actor WHERE last_name LIKE "%GEN%";
```

Result Grid Filter Rows: Edit: Export/Import:

actor_id	first_name	last_name	last_update
14	VIVIEN	BERGEN	2006-02-15 04:34:33
41	JODIE	DEGENERES	2006-02-15 04:34:33
107	GINA	DEGENERES	2006-02-15 04:34:33
166	NICK	DEGENERES	2006-02-15 04:34:33
NULL	NULL	NULL	NULL

SQL File 5* SQL File 6* SQL File 8* SQL File 9* SQL File 7* IceCreamS

Limit to 1000 rows

```
1 • SELECT * FROM sakila.actor WHERE last_name LIKE "%LI%"
2 ORDER BY last_name;
```

Result Grid Filter Rows: Edit: Export/Import:

actor_id	first_name	last_name	last_update
86	GREG	CHAPLIN	2006-02-15 04:34:33
82	WOODY	JOLIE	2006-02-15 04:34:33
15	CUBA	OLIVIER	2006-02-15 04:34:33
34	AUDREY	OLIVIER	2006-02-15 04:34:33
72	SEAN	WILLIAMS	2006-02-15 04:34:33
137	MORGAN	WILLIAMS	2006-02-15 04:34:33
172	GROUCHO	WILLIAMS	2006-02-15 04:34:33
83	BEN	WILLIS	2006-02-15 04:34:33
96	GENE	WILLIS	2006-02-15 04:34:33
164	HUMPHREY	WILLIS	2006-02-15 04:34:33
NULL	NULL	NULL	NULL

SQL File 5* SQL File 6* SQL File 8* SQL File 9* SQL File 7* IceCreamSchema* surfers x Unsolved SQL File 10*

Limit to 1000 rows

```
1 • SELECT country_id, country FROM sakila.country WHERE country IN ("Afghanistan", "Bangladesh", "China");
```

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

country_id	country
1	Afghanistan
12	Bangladesh
23	China
NULL	NULL

Navigator SCHEMAS Filter objects

- Tables
 - actor
 - Columns
 - actor_id
 - first_name
 - middle_name
 - last_name
 - last_update
 - Indexes

SQL File 5* SQL File 6* SQL File 8* SQL File 9* SQL File 7* IceC

```
1 • ALTER TABLE actor
2 ADD COLUMN middle_name VARCHAR(30) AFTER first_name;
3
```

Management Schemas Output

File Edit View Query Database Server Tools Scripting Help

Navigator SCHEMAS Filter objects

- Tables
 - actor
 - Columns
 - actor_id
 - first_name
 - middle_name
 - last_name
 - last_update
 - Indexes
 - Foreign Keys

Management Schemas

Information

Column: last_name

Definition: last_name tinyblob

SQL File 5* SQL File 6* SQL File 8* SQL File 9* SQL File 7* IceCreamSchema* surfers x

```
1 • ALTER TABLE actor
2 MODIFY last_name BLOB(45);
3
4
```

Output

Action Output

#	Time	Action	Message
---	------	--------	---------

Navigator

SCHEMAS

Filter objects

- Tables
 - actor
 - Columns
 - actor_id
 - first_name
 - last_name
 - last_update
 - Indexes
 - Foreign Keys
 - Triggers

Management Schemas

SQL File 5* SQL File 6* SQL File 8* SQL File 9* SQL File 7*

Limit to 1000 rows

```
1 ALTER TABLE actor
2 DROP middle_name;
3
4
```

SQL File 5* SQL File 6* SQL File 8* SQL File 9* SQL File 7* IceCreamSchema* surfers x Unsolved SQL File 10*

Limit to 1000 rows

```
1 ALTER TABLE actor
2 MODIFY last_name VARCHAR(45);
3 SELECT last_name, COUNT(*) FROM actor GROUP BY last_name;
```

Result Grid Filter Rows: Export: Wrap Cell Content:

last_name	COUNT(*)
AKROYD	3
ALLEN	3
ASTAIRE	1
BACALL	1
BAILEY	2
BALE	1
BALL	1
BARRYMORE	1
BASINGER	1
BENING	2
BERGEN	1
BERGMAN	1
BERRY	3
BIRCH	1
BLOOM	1
BOLGER	2
BRIDGES	1
BRODY	2

SQL File 5* SQL File 6* SQL File 8* SQL File 9* SQL File 7* IceCreamSchema* surfers x Unsolved*

Limit to 1000 rows

```
1 SELECT last_name, COUNT(*) count FROM actor GROUP BY last_name Having COUNT(*) > 2;
2
3
```

Result Grid Filter Rows: Export: Wrap Cell Content:

last_name	count
AKROYD	3
ALLEN	3
BERRY	3
DAVIS	3
DEGENERES	3
GARLAND	3
GUINNESS	3
HARRIS	3
HOFFMAN	3
HOPKINS	3
JOHANSSON	3

SQL File 6* SQL File 8* SQL File 9* SQL File 7* IceCreamSchema* surfers x Un

Limit to 1000 rows

```
1 UPDATE actor
2 SET first_name = 'Harpo'
3 WHERE first_name = 'Groucho' AND last_name = 'Williams';
4
```

SQL File 6* SQL File 8* SQL File 9* SQL File 7* IceCreamSchema* surfers x Unsolved* SQL File 10*

Limit to 1000 rows

```
1 UPDATE actor
2 SET first_name = 'Groucho'
3 WHERE first_name = 'Harpo' AND last_name = 'Williams';
4
```

1 • DESCRIBE sakila.address;

2

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

Field	Type	Null	Key	Default	Extra
address id	smallint(5) unsigned	NO	PRI	NULL	auto increment
address	varchar(50)	NO		NULL	
address2	varchar(50)	YES		NULL	
district	varchar(20)	NO		NULL	
city id	smallint(5) unsigned	NO	MUL	NULL	
postal code	varchar(10)	YES		NULL	
phone	varchar(20)	NO		NULL	
location	geometry	NO	MUL	NULL	
last update	timestamp	NO		CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP

1 • SELECT first_name, last_name FROM staff LEFT JOIN address

2 ON staff.address_id = address.address_id;

3

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

first_name	last_name
Mike	Hillver
Jon	Stephens

```
1 • SELECT amount FROM payment LEFT JOIN staff
2     ON payment.staff_id = staff.staff_id;
3
4
```

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

amount
2.99
0.99
5.99
0.99
9.99
4.99
4.99
0.99
3.99
5.99
5.99
4.99
4.99
7.99
2.99

```
1 -- 6c. List each film and the number of actors who are listed for that film.
2 • SELECT title, COUNT(actor_id)
3     FROM
4         film_actor
5     INNER JOIN
6         film
7     ON
8         film.film_id = film_actor.film_id
9     GROUP BY title;
```

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

title	COUNT(actor_id)
ACADEMY DINOSAUR	10
ACE GOLDFINGER	4
ADAPTATION HOLES	5
AFFAIR PREJUDICE	5
AFRICAN EGG	5
AGENT TRUMAN	7
AIRPLANE SIERRA	5

```
1 • SELECT COUNT(inventory.film_id) as x
2     FROM inventory
3     WHERE inventory.film_id IN
4     (
5         SELECT film_id FROM film
6         WHERE title LIKE "Hunchback Impossible");
```

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

x
6


```
1 -- 6e. Using the tables `payment` and `customer` and the `JOIN` command, list the total p
2 SELECT
3     customer.last_name, customer.first_name, SUM(payment.amount)
4 FROM
5     customer
6 INNER JOIN
7     payment
8 ON
9     customer.customer_id = payment.customer_id
10 GROUP BY
11     last_name;
```

last_name	first_name	SUM(payment.amount)
ABNEY	RAFAEL	97.79
ADAM	NATHANIEL	133.72
ADAMS	KATHLEEN	92.73
ALEXANDER	DIANA	105.73
ALLARD	GORDON	160.68
ALLEN	SHIRLEY	126.69
ALVAREZ	CHARLENE	114.73
ANDERSON	LISA	106.76

```
1 -- 7a. The music of Queen and Kris Kristofferson have seen an unlikely resur
2 • SELECT title FROM film WHERE title LIKE "K%" OR title LIKE "Q%";
3
```

title
KANE EXORCIST
KARATE MOON
KENTUCKIAN GIANT
KICK SAVANNAH
KILL BROTHERHOOD
KILLER INNOCENT
KING EVOLUTION
KISS GLORY
KISSING DOLLS
KNOCK WARLOCK
KRAMER CHOCOLATE
KWAI HOMEWARD
QUEEN LUKE
QUEST MUSSOLINI
QUILLS BULL

```
1 -- 7b. Use subqueries to display all actors who appear in the film 'Alone Trip'.
2 SELECT
3     first_name, last_name
4 FROM
5     actor
6 WHERE
7     actor_id
8 IN
9     (SELECT actor_id FROM film_actor WHERE film_id IN (SELECT film_id FROM film WHERE title LIKE "Alone Trip"));
10
```

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

first_name	last_name
ED	CHASE
KARL	BERRY
UMA	WOOD
WOODY	JOLIE
SPENCER	DEPP
CHRIS	DEPP
LAURENCE	BULLOCK
RENEE	BALL

```
1 -- 7c. You want to run an email marketing campaign in Canada, for which you will need the names and email addresses
2 SELECT
3     first_name, last_name, email
4 FROM
5     customer
6 WHERE
7     address_id
8 IN
9     (SELECT address_id from address WHERE city_id IN
10      (SELECT city_id FROM city WHERE country_id IN
11       (SELECT country_id FROM country WHERE country LIKE "Canada"
12        )
13       )
14      );
15
```

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

first_name	last_name	email
DERRICK	BOUROUE	DERRICK.BOUROUE@sakilacustomer.org
DARRELL	POWER	DARRELL.POWER@sakilacustomer.org
LORETTA	CARPENTER	LORETTA.CARPENTER@sakilacustomer.org
CURTIS	IRBY	CURTIS.IRBY@sakilacustomer.org
TROY	OUIGLEY	TROY.OUIGLEY@sakilacustomer.org

```
1 -- 7d. Sales have been lagging among young families, and you wish to target all family movies for a promotion.
2 SELECT
3     title
4 FROM
5     film
6 WHERE
7     film_id
8 IN
9     (SELECT film_id FROM film_category WHERE category_id IN
10      (SELECT category_id from category WHERE name LIKE "Family"
11      )
12     );
```

title
AFRICAN EGG
APACHE DIVINE
ATLANTIS CAUSE
BAKED CLEOPATRA
BANG KWAI
BEDAZZLED MARRIED
BILKO ANONYMOUS
BLANKET BEVERLY
BLOOD ARGONAUTS
BLUES INSTINCT
BRAVEHEART HUMAN
CHASING FIGHT
CHITSUM BEHAVIOR

```
1 -- 7e. Display the most frequently rented movies in descending order.
2 SELECT
3     title, COUNT(title)
4 AS
5     film_count
6 FROM
7     film, inventory, rental
8 WHERE
9     film.film_id = inventory.film_id AND inventory.inventory_id = rental.inventory_id
10 GROUP BY
11     1
12 ORDER BY
13     film_count
14 DESC;
```

title	film_count
BUCKET BROTHERHOOD	34
ROCKETEER MOTHER	33
RIDGEMONT SUBMARINE	32
JUGGLER HARDLY	32
FORWARD TEMPLE	32
SCALAWAG DUCK	32
GRIT CLOCKWORK	32
RUSH GOODFELLAS	31
APACHE DIVINE	31
TIMBERLAND SKY	31
ZORRO ARK	31
ROBBERS JOON	31

Limit to 1000 rows

```

1 -- 7f. Write a query to display how much business, in dollars, each store brought in.
2 • SELECT COUNT(amount) FROM payment GROUP BY staff_ID;
3

```

Result Grid

COUNT(amount)
8057
7992

Limit to 1000 rows

```

1 -- 7g. Write a query to display for each store its store ID, city, and country.
2 • SELECT
3     store_id, city, country
4 FROM
5     store, address, city, country
6 WHERE
7     store.address_id = address.address_id
8 AND
9     address.city_id = city.city_id
10 AND
11    city.country_id = country.country_id;

```

Result Grid

store_id	city	country
1	Lethbridge	Canada
2	Woodridge	Australia

Limit to 1000 rows

```

1 -- 7h. List the top five genres in gross revenue in descending order. (**Hint**: you may need to u
2 • SELECT
3     category.name, sum(payment.amount)
4 AS
5     revenue
6 FROM
7     payment, rental, inventory, film_category, category
8 WHERE
9     payment.rental_id = rental.rental_id AND rental.inventory_id = inventory.inventory_id AND inve
10 GROUP BY
11     1
12 ORDER BY
13     revenue
14 DESC
15 limit 5;

```

Result Grid

name	revenue
Sports	5314.21
Sci-Fi	4756.98
Animation	4656.30
Drama	4587.39
Comedy	4383.58

Navigator

SCHEMAS

Filter objects

- last_update
 - Indexes
 - Foreign Keys
 - Triggers
- staff
- store
- Views
 - actor_info
 - customer_list
 - film_list
 - nicer_but_slower_film_list
 - sales_by_film_category
 - sales_by_store
 - staff_list
 - top_5_genres
 - name
 - revenue

SQL File 6* SQL File 8* SQL File 9* surfers x Unsolved* SQL File 10* SQL File 11*

Limit to 1000 rows

```
1 -- 8a. In your new role as an executive, you would like to have a view that displays the top 5 genres by revenue.  
2 CREATE VIEW top_5_genres AS  
3 (  
4 SELECT  
5     category.name, sum(payment.amount)  
6 AS  
7     revenue  
8 FROM  
9     payment, rental, inventory, film_category, category  
10 WHERE  
11     payment.rental_id = rental.rental_id AND rental.inventory_id = film_category.inventory_id  
12 GROUP BY  
13     1  
14 ORDER BY  
15     revenue  
16 DESC  
17     limit 5);
```

Limit to 1000 rows

```
1 -- 8b. How would you display the view that you created in 8a?  
2 SELECT * FROM top_5_genres;  
3
```

Result Grid

Filter Rows: Export: Wrap Cell Content: IA

	name	revenue
	Sports	5314.21
	Sci-Fi	4756.98
	Animation	4656.30
	Drama	4587.39
	Comedy	4383.58

