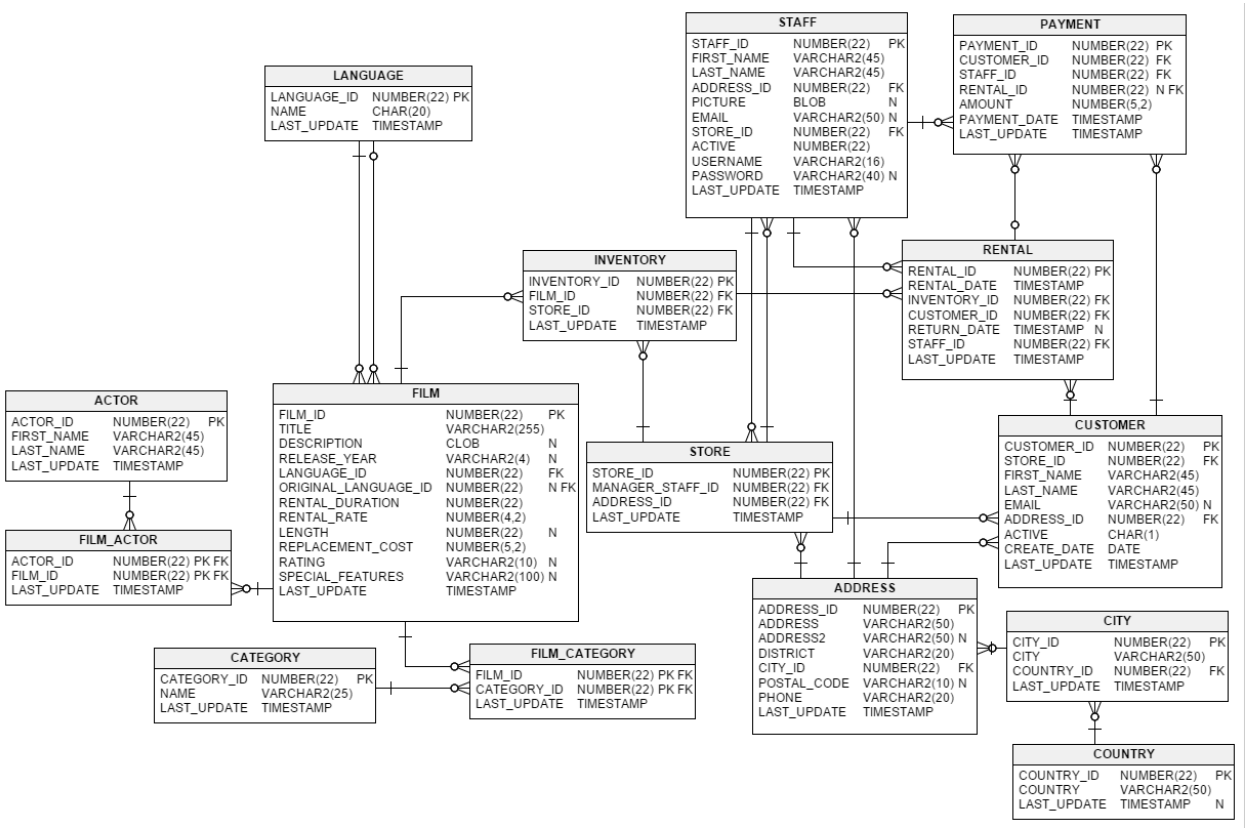
Introduction

The Sakila database is a nicely normalised schema modelling a DVD rental store, featuring things like films, actors, film-actor relationships, and a central inventory table that connects films, stores, and rentals.



Exercises

1. Display the first and last name of each actor in a single column in upper case letters in alphabetic order. Name the column Actor Name.

```
2 •  SELECT UPPER(CONCAT( first_name,'  ',last_name)) AS 'Actor Name' FROM actor
3    order by 'Actor Name';
4
```

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

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

2. Find all actors whose last name contain the letters GEN:

```
6 •  SELECT first_name,last_name FROM actor
7    WHERE last_name LIKE '%GEN%';
```

Result Grid | Filter Rows: | Export:

first_name	last_name
VIVIEN	BERGEN
JODIE	DEGENERES
GINA	DEGENERES
NICK	DEGENERES

3. Using IN, display the country_id and country columns of the following countries: Afghanistan, Bangladesh, and China:

```
11 •  SELECT country_id,country FROM country
12    WHERE country IN ('Afghanistan', 'Bangladesh', 'China');
```

Result Grid | Filter Rows: | Edit: | Export/Impo

country_id	country
1	Afghanistan
12	Bangladesh
23	China
NULL	NULL

4. List the last names of actors, as well as how many actors have that last name.

```
15 •  SELECT last_name , COUNT(*) AS NUM_ACTORS
16    FROM actor
17    group by last_name
18    order by last_name;
```

Result Grid | Filter Rows: | Export: | W

last_name	NUM_ACTORS
AKROYD	3
ALLEN	3
ASTAIRE	1
BACALL	1
BAILEY	2
BALE	1
BALL	1
BARRYMORE	1
BASINGER	1
BENING	2
BERGEN	1
BERGMAN	1

5. List last names of actors and the number of actors who have that last name, but only for names that are shared by at least two actors

```
21 •  SELECT last_name , COUNT(*) AS NUM_ACTORS
22    FROM actor
23    group by last_name
24    HAVING NUM_ACTORS <3
25    order by last_name;
```

Result Grid | Filter Rows: | Export:

last_name	NUM_ACTORS
ASTAIRE	1
BACALL	1
BAILEY	2
BALE	1
BALL	1
BARRYMORE	1
BASINGER	1
BENING	2

6. The actor HARPO WILLIAMS was accidentally entered in the actor table as GROUCHO WILLIAMS. Write a query to fix the record.

```
28 •  update actor
29    set first_name= 'HARPO'
30    WHERE (first_name='GROUCHO' and last_name='WILLIAMS');
31 •  SELECT * FROM actor where first_name LIKE 'HARPO';
```

Result Grid | Filter Rows: | Edit: | Export/I

actor_id	first_name	last_name	last_update
172	HARPO	WILLIAMS	2024-06-19 14:26:08
NULL	NULL	NULL	NULL

7. Use JOIN to display the first and last names, as well as the address, of each staff member. Use the tables staff and address:

```
34 •  SELECT
35        S.first_name,
36        S.last_name ,
37        a.address
38    FROM
39        STAFF S
40    JOIN address a on S.address_id=a.address_id;
```

Result Grid | Filter Rows: | Export: | Wr

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

8. List each film and the number of actors who are listed for that film. Use tables film_actor and film. Use inner join.

```
43 •  select f.title as Film_title,
44    COUNT(fa.actor_id) AS NUM_ACTORS
45    from film f
46    inner join film_actor fa on f.film_id= fa.film_id
47    group by f.film_id, f.title
48    order by f.title;
49
```

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

Film_title	NUM_ACTORS
AMISTAD MIDSUMMER	4
ANACONDA CONFES...	5
ANALYZE HOOSIERS	5
ANGELS LIFE	9
ANNIE IDENTITY	3
ANONYMOUS HUMAN	9
ANTHEM LUKE	2
ANTITRUST TOMAT...	7

9. How many copies of the film Hunchback Impossible exist in the inventory system?

```
54 •  SELECT COUNT(i.inventory_id) AS copies_count
55    FROM film f
56    INNER JOIN inventory i ON f.film_id = i.film_id
57    WHERE f.title = 'Hunchback Impossible';
```

Result Grid | Filter Rows: | Export: | Wrap Ce

copies_count
6

10. Using the tables payment and customer and the JOIN command, list the total paid by each customer. List the customers alphabetically by last name

```
61 •  select c.last_name ,sum(p.amount) as total_paid
62    from customer c
63    inner join payment p on p.customer_id= c.customer_id
64    group by c.customer_id,c.first_name,c.last_name
65    order by c.last_name;
66
```

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

last_name	total_paid
ABNEY	97.79
ADAM	133.72
ADAMS	92.73
ALEXANDER	105.73
ALLARD	160.68
ALLEN	126.69

11. The music of Queen and Kris Kristofferson have seen an unlikely resurgence. As an unintended consequence, films starting with the letters `K` and `Q` have also soared in popularity. Use subqueries to display the titles of movies starting with the letters `K` and `Q` whose language is English.

```
71 •  select title from film
72    where (title like 'K%' OR title like 'Q%') and
73    language_id= (SELECT language_id from language l
74    where name='English');
```

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

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

12. Use subqueries to display all actors who appear in the film `Alone Trip`.

```
82 •  SELECT CONCAT( first_name,'   ',last_name) as ACTORS
83    FROM actor
84    WHERE actor_id IN (
85        SELECT actor_id
86        FROM film_actor
87        WHERE film_id IN (
88            SELECT film_id
89            FROM film f
90            WHERE title = 'Alone Trip'));
91
```

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

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

13. You want to run an email marketing campaign in Canada, for which you will need the names and email addresses of all Canadian customers. Use joins to retrieve this information.

```
103 •   select first_name as 'Canadian Customers' ,Email  from customer c
104     join address a on c.address_id=a.address_id
105     join city ci on a.city_id = ci.city_id
106     join country co on ci.country_id = co.country_id
107     where country = 'Canada';
```

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

Canadian Customers	Email
DERRICK	DERRICK.BOURQUE@sakilacustomer.org
DARRELL	DARRELL.POWER@sakilacustomer.org
LORETTA	LORETTA.CARPENTER@sakilacustomer.org
CURTIS	CURTIS.IRBY@sakilacustomer.org
TROY	TROY.QUIGLEY@sakilacustomer.org

14. Sales have been lagging among young families, and you wish to target all family movies for a promotion. Identify all movies categorized as famiy films.

```
112
113 •   select title as 'Family Movies' from film where film_id in
114     (SELECT film_id FROM film_category where category_id =
115     (SELECT category_id  FROM category c where c.name ='Family'));
116
```

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

Family Movies
AFRICAN EGG
APACHE DIVINE
ATLANTIS CAUSE
BAKED CLEOPATRA
BANG KWAI
BEDAZZLED MARRIED
BILKO ANONYMOUS
BLANKET BEVERLY
BLOOD ARGONAUTS

15. Create a Stored procedure to get the count of films in the input category (IN category_name, OUT count)

```
118     DELIMITER $$
119 •   CREATE PROCEDURE Get_Count_Of_Films (
120         IN category_name varchar(45),
121         OUT film_count INT
122     )
123     BEGIN
124         SELECT COUNT(*)
125         INTO film_count
126         FROM film f
127         join film_category fc on f.film_id = fc.film_id
128         join category c on fc.category_id = c.category_id
129         WHERE c.name = category_name;
130     END$$
131     DELIMITER ;
132
133 •   CALL Get_Count_Of_Films('Family',@film_count);
134 •   select @film_count;
135
```

16. Display the most frequently rented movies in descending order.

```
138 •  SELECT f.title, COUNT(f.title) as rentals from film f
139    JOIN
140        (SELECT r.rental_id, i.film_id FROM rental r
141        JOIN
142        inventory i ON i.inventory_id = r.inventory_id) a
143        ON a.film_id = f.film_id
144        GROUP BY f.title
145        ORDER BY rentals DESC;
146
```

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

title	rentals
BUCKET BROTHERHOOD	34
ROCKETEER MOTHER	33
FORWARD TEMPLE	32
GRIT CLOCKWORK	32
JUGGLER HARDLY	32
RIDGEMONT SUBMARINE	32
SCALAWAG DUCK	32
APACHE DIVINE	31
GOODFELLAS SALUTE	31

17. Write a query to display for each store its store ID, city, and country.

```
149 •  SELECT s.store_id, c.city, co.country
150    FROM store s
151    JOIN address a ON s.address_id = a.address_id
152    JOIN city c ON a.city_id = c.city_id
153    JOIN country co ON c.country_id = co.country_id;
154
```

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

store_id	city	country
1	Lethbridge	Canada
2	Woodridge	Australia

18. List the genres and its gross revenue.

```
158 •   select c.name as Genre,
159     sum(p.amount) as Gross_revenue
160     from payment p
161     join rental r on p.rental_id = r.rental_id
162     join inventory i on r.inventory_id = i.inventory_id
163     join film_category fc on i.film_id = fc.film_id
164     join category c on fc.category_id = c.category_id
165     group by c.name
166     order by Gross_revenue Desc;
```

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

Genre	Gross_revenue
Sports	5314.21
Sci-Fi	4756.98
Animation	4656.30
Drama	4587.39
Comedy	4383.58
Action	4375.85
New	4351.62
Games	4281.33
Foreign	4270.67

19. Create a View for the above query(18)

```
169 •   create view Genres_Grossrevenue
170     as
171     select c.name as Genre,
172     sum(p.amount) as Gross_revenue
173     from payment p
174     join rental r on p.rental_id = r.rental_id
175     join inventory i on r.inventory_id = i.inventory_id
176     join film_category fc on i.film_id = fc.film_id
177     join category c on fc.category_id = c.category_id
178     group by c.name
179     order by Gross_revenue Desc;
```

20. Select top 5 genres in gross revenue view.

```
181 •   select * from genres_grossrevenue
182     limit 5;
183
184
```

Result Grid | Filter Rows: | Expor

Genre	Gross_revenue
Sports	5314.21
Sci-Fi	4756.98
Animation	4656.30
Drama	4587.39
Comedy	4383.58

genres_grossrevenue6 ×