

SQL

Hecho por: Alejandro Escolano Torres

Profesora: Natalia Gavaldá

EJERCICIOS A REALIZAR

¿Qué porcentaje supone el coste de alquiler sobre el coste de reemplazar?

```
1 SELECT title,
2 rental_rate,
3 replacement_cost,
4 round(replacement_cost /
5 rental_rate,2) AS
6 new_column
7 FROM public.film
```

	title	rental_rate	replacement_cost	new_column
	character varying (255)	numeric (4,2)	numeric (5,2)	numeric
1	Chamber Italian	4.99	14.99	3.00
2	Grosse Wonderful	4.99	19.99	4.01
3	Airport Pollock	4.99	15.99	3.20
4	Bright Encounters	4.99	12.99	2.60
5	Academy Dinosaur	0.99	20.99	21.20
6	Ace Goldfinger	4.99	12.99	2.60
7	Adaptation Holes	2.99	18.99	6.35
8	Affair Prejudice	2.99	26.99	9.03
9	African Egg	2.99	22.99	7.69

¿Cuántas veces tienen que alquilar cada película para igualar o superar el coste de reemplazar la película?

```
1 SELECT title, rental_rate,replacement_cost,
2 ceil(replacement_cost / rental_rate) AS new_operation
3 FROM public.film
```

	title	rental_rate	replacement_cost	new_operation
	character varying (255)	numeric (4,2)	numeric (5,2)	numeric
1	Chamber Italian	4.99	14.99	4
2	Grosse Wonderful	4.99	19.99	5
3	Airport Pollock	4.99	15.99	4
4	Bright Encounters	4.99	12.99	3

¿Cuántas películas disponibles?

```
1 SELECT COUNT(DISTINCT film_id) AS peliculas_disponibles
2 FROM inventory;
3
```

	peliculas_disponibles
	bigint
1	958

¿Cual es el precio más caro? ¿Y el mas barato? ¿Cual es el precio medio de alquiler? ¿Qué variabilidad de precios tenemos?

```
1 SELECT
2 MAX(rental_rate) AS precio_mas_caro,
3 MIN(rental_rate) AS precio_mas_barato,
4 AVG(rental_rate) AS precio_medio,
5 STDEV(rental_rate) AS variabilidad_precio
6 FROM film;
```

	precio_mas_caro	precio_mas_barato	precio_medio	variabilidad_precio
	numeric	numeric	numeric	numeric
1	4.99	0.99	1.6483932126350043	0.991980000000000000

¿Cual es el coste más pequeño de reemplazo? ¿Cual es el más grande?

Query Editor

Query History

1

SELECT

2

MIN(replacement_cost) AS coste_mas_bajo,

3

MAX(replacement_cost) AS coste_mas_alto

4

FROM film;

5

Data Output

Explain

Messages

Notifications

coste_mas_bajo

numeric

coste_mas_alto

numeric

1

9.99

29.99

¿Cómo se llaman los actores que empiezan por la letra A?

1	SELECT first_name, last_name		
2	FROM actor		
3	WHERE first_name ILIKE 'A%';		
4			

Data Output		Explain	Messages	Notifications
first_name	last_name			
character varying (45)	character varying (45)			
1	Alec	Wayne		
2	Audrey	Olivier		
3	Anne	Cronyn		
..	..			

¿Cuales son las películas que podemos alquilar por más de 10€?

SELECT title, rental_rate	
FROM film	
WHERE rental_rate > 10;	

Output		Explain	Messages	Notifications
itle	rental_rate			
character varying (255)	numeric (4,2)			

¿Cuantas películas podemos alquilar entre 5 y 10 euros?

SELECT COUNT (*) AS cantidad	
FROM film	
WHERE rental_rate BETWEEN 5 AND 10;	

Output		Explain	Messages	Notifications
cantidad				
bigint				
0				

¿Cuantas películas podemos alquilar por menos de 5€ y con una duración menor a 100 minutos?

<pre>SELECT COUNT(*) AS cantidad FROM film WHERE rental_rate < 5 AND length < 100;</pre>	
Output	Explain Messages Notifications
cantidad	
378	

¿Qué precio de alquiler tienen las siguientes películas? Giant Troopers, Gilbert Pelican, Gilmore Boiled

<pre>SELECT title, rental_rate FROM film WHERE title IN ('Giant Troopers', 'Gilbert Pelican', 'Gilmore Boiled');</pre>	
Output	Explain Messages Notifications
title	rental_rate
character varying (255)	numeric (4,2)
Giant Troopers	2.99
Gilbert Pelican	0.99
Gilmore Boiled	0.99

¿Qué rating tiene la película “Ali Forever”? ¿Cuánta es su duración?

```
SELECT title, rating, length
FROM film
WHERE title = 'Ali Forever';
```

Output	Explain	Messages	Notifications
title	rating	length	
character varying (255)	mpaa_rating	smallint	
Ali Forever	PG	150	

¿Nos falta por informar algún precio de alquiler en nuestra base de datos?

```
SELECT COUNT(*) AS peliculas_sin_precio
FROM film
WHERE rental_rate IS NULL;
```

Output	Explain	Messages	Notifications
peliculas_sin_precio			
0			

Obten por ‘rating’:

- El número de películas
- El precio medio de alquiler
- El mínimo precio de alquiler
- El máximo precio de alquiler
- La duración media de las películas
- El año de la película más antigua

- El año de la película más nueva

```
SELECT
rating,
COUNT(*) AS num_peliculas,
AVG(rental_rate) AS precio_medio,
MIN(rental_rate) AS precio_min,
MAX(rental_rate) AS precio_max,
AVG(length) AS duracion_medio,
MIN(release_year) AS año_mas_antiguo,
MAX(release_year) AS año_mas_reciente
FROM film
GROUP BY rating
ORDER BY rating;
```

rating	num_peliculas	precio_medio	precio_min	precio_max	duracion_medio	año_mas_antiguo	año_mas_reciente
G	178	1.888076404943820	0.99	4.99	111.050561797752890	2006	2006
PG	194	1.0518556701030928	0.99	4.99	112.0051546291752577	2006	2006
PG-13	223	1.034843049273543	0.99	4.99	120.4439461883408072	2006	2006
R	195	1.9387179487179487	0.99	4.99	118.6615384615384615	2006	2006
NC-17	210	1.9709523809523810	0.99	4.99	113.2285714285714286	2006	2006

Obten por ‘rating’:

El número de películas y quédate únicamente con aquellos

rating que tengan más de 200 películas

El precio medio de alquiler y quédate únicamente con aquellos

rating que tenga un precio medio superior a 3

La duración media de las películas y quédate con aquellos

rating que tengan una duración media mayor a 115 minutos

```
SELECT rating, COUNT(*) AS num_peliculas
FROM film
GROUP BY rating
HAVING COUNT(*) > 200;
```

rating	num_peliculas
G	178
PG-13	223
NC-17	210

```
SELECT rating, AVG(rental_rate) AS precio_medio
FROM film
GROUP BY rating
HAVING AVG(rental_rate) > 3;
```

rating	precio_medio
PG-13	1.034843049273543
G	1.0518556701030928

```
1 SELECT rating, AVG(length) AS duracion_medio
2 FROM film
3 GROUP BY rating
4 HAVING AVG(length) > 100;
5
```

rating	duracion_medio
PG-13	120.4439461883408072
NC-17	113.2285714285714286
G	111.050561797752890
PG	112.0051546291752577
R	118.6615384615384615

Obtén las direcciones de aquellos clientes de nuestro videoclub

Ahora, añade las ciudades de las que son nuestros clientes

Añade también el país

```

1 SELECT
2   c.customer_id,
3   c.first_name,
4   c.last_name,
5   a.address,
6   ci.city,
7   co.country
8 FROM customer c
9 JOIN address a ON c.address_id = a.address_id
10 JOIN city ci ON a.city_id = ci.city_id
11 JOIN country co ON ci.country_id = co.country_id;

```

customer_id	first_name	last_name	address	city	country
1	Jared	Ely	1003 Qinhuangdao Street	Purwakarta	Indonesia
2	Mary	Smith	1913 Hanoi Way	Sasebo	Japan
3	Patricia	Johnson	1121 Loja Avenue	San Bernardino	United States

Obtén solamente las películas que tienen un actor que tenga un apellido que empiece por la letra “C”

```

1 SELECT DISTINCT f.title
2 FROM film f
3 JOIN film_actor fa ON f.film_id = fa.film_id
4 JOIN actor a ON fa.actor_id = a.actor_id
5 WHERE a.last_name ILIKE 'C%';
6

```

title
Graceland Dynamite
Opus Ice
Wonderful Drop

Responde a las siguientes preguntas:

- ¿Cuántos actores tiene cada película?
- ¿Cuáles son las películas que tienen más de 2 actores?
- ¿Cual es la película que tiene más actores?

```

1 SELECT
2   f.title,
3   COUNT(fa.actor_id) AS numero_actores
4 FROM film f
5 JOIN film_actor fa ON f.film_id = fa.film_id
6 GROUP BY f.title
7 ORDER BY numero_actores DESC
8 LIMIT 1;
9

```

title	numero_actores
Lambs Cincinatti	15

```

1 SELECT
2   f.title,
3   COUNT(fa.actor_id) AS numero_actores
4 FROM film f
5 JOIN film_actor fa ON f.film_id = fa.film_id
6 GROUP BY f.title
7 HAVING COUNT(fa.actor_id) > 2
8 ORDER BY numero_actores DESC;
9

```

title	numero_actores
Lambs Cincinatti	15
Dracula Crystal	13
Boondock Ballroom	13

```

1 SELECT
2     f.title,
3     COUNT(fa.actor_id) AS numero_actores
4 FROM film f
5 JOIN film_actor fa ON f.film_id = fa.film_id
6 GROUP BY f.title
7 ORDER BY numero_actores DESC;
8

```

	title	numero_actores	
	character varying (255)	bigint	
1	Lambs Cincinatti	15	
2	Random Go	13	
3	Crazy Home	13	
4	Boondock Ballroom	13	

Desde el videoclub estamos empezando a guardar las opiniones de los clientes referentes a una película. Esta información queremos guardarla en base de datos para que podamos analizarla:

- film_id
- customer_id
- review_date
- review_description

Llama a esta tabla “reviews” seguido de la inicial de tu nombre y tu apellido. Por ejemplo, reviews_ng sería el nombre que le daría.

```

1 CREATE TABLE reviews_jc (
2     review_id SERIAL PRIMARY KEY,
3     film_id INT NOT NULL REFERENCES film(film_id),
4     customer_id INT NOT NULL REFERENCES customer(customer_id),
5     review_date DATE NOT NULL,
6     review_description TEXT
7 );

```

Data Output Explain **Messages** Notifications

CREATE TABLE

Query returned successfully in 1 secs 103 msec.

Cambia la opinión del cliente 7, que ha hecho de la película 4. Nos ha comunicado que su opinión era errónea.

Su opinión es la siguiente: “La película es bastante divertida y para todo los públicos”

```

1 UPDATE reviews_jc
2 SET review_description = 'La película es bastante divertida y para todo los públicos'
3 WHERE customer_id = 7 AND film_id = 4;
4

```

Data Output Explain **Messages** Notifications

UPDATE 0

Query returned successfully in 195 msec.

Este cliente nos pide que borremos sus opiniones de nuestra base de datos, de manera que no queden registradas en la tabla

```
1 DELETE FROM reviews_jc
2 WHERE customer_id = 7;
3
```

Data Output	Explain	Messages	Notifications
DELETE 0			
Query returned successfully in 93 msec.			

Añade una nueva columna a la tabla de reviews que te parezca interesante. Por ejemplo, el número de estrellas que le darías a una película. Puedes llamarle “review_stars” y con datatype int2

Renombra una de las columnas de la tabla de reviews. Por ejemplo, renombra la columna “review_description” a “review_opinion”

```
Query Editor Query History
1 ALTER TABLE reviews_jc
2 ADD COLUMN review_stars INT2;
3
```

Data Output	Explain	Messages	Notifications
ALTER TABLE			
Query returned successfully in 184 msec.			

```
1 ALTER TABLE reviews_jc
2 RENAME COLUMN review_description TO review_opinion;
3
```

Data Output	Explain	Messages	Notifications
ALTER TABLE			
Query returned successfully in 212 msec.			

Debido a que ya no tenemos información de las reseñas que dejan los clientes en nuestra tabla, procedemos a borrarla.

```
1 DROP TABLE IF EXISTS reviews_jc;
2
```

Data Output	Explain	Messages	Notifications
DROP TABLE			
Query returned successfully in 511 msec.			

Crea una vista que se llame como tu nombre y apellidos e introduce todos los campos que te resulten interesantes de las películas y sus actores.

Query Editor Query History

```

1 CREATE VIEW juan_cruz AS
2 SELECT
3     f.film_id,
4     f.title,
5     f.release_year,
6     f.rating,
7     f.length,
8     f.rental_rate,
9     a.first_name || ' ' || a.last_name AS actor_nombre
10 FROM film f
11 JOIN film_actor fa ON f.film_id = fa.film_id

```

Data Output Explain Messages Notifications

CREATE VIEW

Query returned successfully in 540 msec.

Obtén haciendo una subconsulta en la cláusula WHERE, todas aquellas películas que están en el idioma de inglés

```

1 SELECT *
2 FROM film
3 WHERE language_id = (
4     SELECT language_id
5     FROM language
6     WHERE name = 'English'
7 );
8

```

Data Output Explain Messages Notifications

film_id	title	description	release_year	language_id	rental_duration	rental_rate
133	Chamber Italian	A Fateful Reflec...	2006	1	7	
384	Grosse Wonderful	A Epic Drama of...	2006	1	5	
8	Airport Pollock	A Epic Tale of a ...	2006	1	6	
98	Bright Encounters	A Fateful Yarn o...	2006	1	4	
1	Academy Dinosaur	A Epic Drama of...	2006	1	6	
2	Ace Goldfinger	A Astounding E...	2006	1	3	

Subconsultas

- Obtén haciendo una subconsulta en la cláusula WHERE, todos aquellos clientes que viven en una dirección que empieza por A

- Obtén haciendo una subconsulta en la cláusula WHERE, aquellos clientes que han se han gastado más de 190€

```

1 SELECT *
2 FROM customer
3 WHERE customer_id IN (
4     SELECT customer_id
5     FROM payment
6     GROUP BY customer_id
7     HAVING SUM(amount) > 190
8 );
9

```

Data Output Explain Messages Notifications

customer_id	store_id	first_name	last_name	email	address_id
137	2	Rhonda	Kennedy	rhonda.kennedy@sakilacust...	
148	1	Eleanor	Hunt	eleanor.hunt@sakilacustom...	
178	2	Marion	Snyder	marion.snyder@sakilacusto...	
526	2	Karl	Seal	karl.seal@sakilacustomer.org	

```

1 SELECT *
2 FROM customer
3 WHERE address_id IN (
4     SELECT address_id
5     FROM address
6     WHERE address ILIKE 'A%'
7 );
8

```

Data Output Explain Messages Notifications

customer_id	store_id	first_name	last_name

Obtén aquellos film_id que tienen más de un replacement_cost de 2€ y obtén el total de film_id.

```

1 SELECT COUNT(DISTINCT film_id) AS total
2 FROM film
3 WHERE replacement_cost = 2
4 GROUP BY film_id
5 HAVING COUNT(*) > 1;
6

```

Data Output	Explain	Messages	Notifications
total bigint			

Obtén haciendo una subconsulta en la cláusula WITH

- La suma de los amount que de los clientes que han pagado más de 190€
- El número de clientes que han pagado más de 190€

```

1 WITH clientes_gastadores AS (
2     SELECT customer_id
3     FROM payment
4     GROUP BY customer_id
5     HAVING SUM(amount) > 190
6 )
7 SELECT COUNT(*) AS num_clientes
8 FROM clientes_gastadores;
9

```

Data Output	Explain	Messages	Notifications
num_clientes bigint			
1	4		

```

1 WITH clientes_gastadores AS (
2     SELECT customer_id
3     FROM payment
4     GROUP BY customer_id
5     HAVING SUM(amount) > 190
6 )
7 SELECT SUM(amount) AS total_gastado
8 FROM payment
9 WHERE customer_id IN (SELECT customer_id FROM clientes_gastadores);
10

```

Data Output	Explain	Messages	Notifications
total_gastado numeric			
1	806.36		

Obtén haciendo una subconsulta en la cláusula WITH

- El número de veces que un cliente ha alquilado una película.
- El número de veces que un cliente ha alquilado una película en el año 2005 y en el 2006

```

1 WITH alquileres_por_año AS (
2   SELECT customer_id, EXTRACT(YEAR FROM rental_date) AS año, COUNT(*) AS total
3   FROM rental
4   WHERE EXTRACT(YEAR FROM rental_date) IN (2005, 2006)
5   GROUP BY customer_id, EXTRACT(YEAR FROM rental_date)
6 )
7 SELECT * FROM alquileres_por_año;
8

```

Data Output	Explain	Messages	Notifications
	customer_id smallint	año double precision	total bigint
1	3	2005	26
2	573	2005	29
3	422	2006	1

```

1 WITH alquileres AS (
2   SELECT customer_id, film_id, COUNT(*) AS veces
3   FROM rental r
4   JOIN inventory i ON r.inventory_id = i.inventory_id
5   GROUP BY customer_id, film_id
6 )
7 SELECT * FROM alquileres;
8

```

Data Output	Explain	Messages	Notifications
	customer_id smallint	film_id smallint	veces bigint
1	596	103	1
2	176	121	1
3	459	724	1