FINAL DRILLING: ARRIENDO DE DVDS

Javier Medina Muñoz

1. Construye las siguientes consultas:
    a. Aquellas usadas para insertar, modificar y eliminar un Customer, Staff y Actor.

Customer:

Insertar:

```
INSERT INTO customer(store_id,first_name, last_name, email, address_id, activebool, create_date, last_update, active) VALUES ( 1,'Jon','Snow','jsnow@gmail.com',5,TRUE,'2010-11-04', CURRENT_DATE, 1);
```

	customer_id [PK] integer	store_id smallint	first_name character varying (45)	last_name character varying (45)	email character varying (50)	address_id smallint	activebool boolean	create_date date	last_update timestamp without time zone	active integer
596	596	1	Enrique	Forsythe	enrique.forsythe@sakilacustomer.org	602	true	2006-02-14	2013-05-26 14:49:45.738	1
597	597	1	Freddie	Duggan	freddie.duggan@sakilacustomer.org	603	true	2006-02-14	2013-05-26 14:49:45.738	1
598	598	1	Wade	Delvalle	wade.delvalle@sakilacustomer.org	604	true	2006-02-14	2013-05-26 14:49:45.738	1
599	599	2	Austin	Cintron	austin.cintron@sakilacustomer.org	605	true	2006-02-14	2013-05-26 14:49:45.738	1
600	602	1	Jon	Snow	jsnow@gmail.com	5	true	2010-11-04	2024-10-16 00:00:00	1

Modificar:

```
UPDATE customer SET email = 'esto_es_una_prueba@gmail.com' WHERE customer_id = 20;
```

	customer_id [PK] integer	store_id smallint	first_name character varying (45)	last_name character varying (45)	email character varying (50)	address_id smallint	activebool boolean	create_date date	last_update timestamp without time zone	active integer
17	17	1	Donna	Thompson	donna.thompson@sakilacustomer.org	21	true	2006-02-14	2013-05-26 14:49:45.738	1
18	18	2	Carol	Garcia	carol.garcia@sakilacustomer.org	22	true	2006-02-14	2013-05-26 14:49:45.738	1
19	19	1	Ruth	Martinez	ruth.martinez@sakilacustomer.org	23	true	2006-02-14	2013-05-26 14:49:45.738	1
20	20	2	Sharon	Robinson	esto_es_una_prueba@gmail.com	24	true	2006-02-14	2024-10-16 16:20:45.414019	1
21	21	1	Michelle	Clark	michelle.clark@sakilacustomer.org	25	true	2006-02-14	2013-05-26 14:49:45.738	1
22	22	1	Laura	Rodriguez	laura.rodriguez@sakilacustomer.org	26	true	2006-02-14	2013-05-26 14:49:45.738	1

Eliminar:

```
DELETE FROM customer WHERE first_name = 'Jon' AND last_name = 'Snow';
```

	customer_id [PK] integer	store_id smallint	first_name character varying (45)	last_name character varying (45)	email character varying (50)	address_id smallint	activebool boolean	create_date date	last_update timestamp without time zone	active integer
587	587	1	Sergio	Stamfield	sergio.stamfield@sakilacustomer.org	593	true	2006-02-14	2013-05-26 14:49:45.738	1
588	588	1	Marion	Ocampo	marion.ocampo@sakilacustomer.org	594	true	2006-02-14	2013-05-26 14:49:45.738	1
589	589	1	Tracy	Herrmann	tracy.herrmann@sakilacustomer.org	595	true	2006-02-14	2013-05-26 14:49:45.738	1
590	590	2	Seth	Hannon	seth.hannon@sakilacustomer.org	596	true	2006-02-14	2013-05-26 14:49:45.738	1
591	591	1	Kent	Arsenault	kent.arsenault@sakilacustomer.org	597	true	2006-02-14	2013-05-26 14:49:45.738	1
592	592	1	Terrance	Roush	terrance.roush@sakilacustomer.org	598	true	2006-02-14	2013-05-26 14:49:45.738	0
593	593	2	Rene	Mcalister	rene.mcalister@sakilacustomer.org	599	true	2006-02-14	2013-05-26 14:49:45.738	1
594	594	1	Eduardo	Hiatt	eduardo.hiatt@sakilacustomer.org	600	true	2006-02-14	2013-05-26 14:49:45.738	1
595	595	1	Terrence	Gunderson	terrence.gunderson@sakilacustomer.org	601	true	2006-02-14	2013-05-26 14:49:45.738	1
596	596	1	Enrique	Forsythe	enrique.forsythe@sakilacustomer.org	602	true	2006-02-14	2013-05-26 14:49:45.738	1
597	597	1	Freddie	Duggan	freddie.duggan@sakilacustomer.org	603	true	2006-02-14	2013-05-26 14:49:45.738	1
598	598	1	Wade	Delvalle	wade.delvalle@sakilacustomer.org	604	true	2006-02-14	2013-05-26 14:49:45.738	1
599	599	2	Austin	Cintron	austin.cintron@sakilacustomer.org	605	true	2006-02-14	2013-05-26 14:49:45.738	1

Staff:

Insertar:

```
INSERT INTO staff(first_name, last_name, address_id, email, store_id, active, username, password, last_update) VALUES ( 'Rob','Stark',504,'rstark@gmail.com',2,TRUE,'Rob','asdfghjklñlkj623482842392342', '2022-05-29');
```

	staff_id [PK] integer	first_name character varying (45)	last_name character varying (45)	address_id smallint	email character varying (50)	store_id smallint	active boolean	username character varying (16)	password character varying (40)	last_update timestamp without time zone	picture bytea
1	1	Mike	Hillyer	3	Mike.Hillyer@sakilastaff.com	1	true	Mike	8cb2237d0679ca88db6464eac60da96345513964	2006-05-16 16:13:11.79328	[binary dat...
2	2	Jon	Stephens	4	Jon.Stephens@sakilastaff.com	2	true	Jon	8cb2237d0679ca88db6464eac60da96345513964	2006-05-16 16:13:11.79328	[null]
3	3	Rob	Stark	504	rstark@gmail.com	2	true	Rob	asdfghjklñlkj623482842392342	2022-05-29 00:00:00	[null]

Modificar:

```
UPDATE staff SET first_name = 'Ana', username = 'Ana' WHERE first_name = 'Mike';
```

	staff_id [PK] integer	first_name character varying (45)	last_name character varying (45)	address_id smallint	email character varying (50)	store_id smallint	active boolean	username character varying (16)	password character varying (40)	last_update timestamp without time zone	picture bytea
1	1	Ana	Hillyer	3	Mike.Hillyer@sakilastaff.c...	1	true	Ana	8cb2237d0679ca88db6464ea...	2024-10-16 16:29:15.897315	[binary da...
2	2	Jon	Stephens	4	Jon.Stephens@sakilastaff...	2	true	Jon	8cb2237d0679ca88db6464ea...	2006-05-16 16:13:11.79328	[null]
3	3	Rob	Stark	504	rstark@gmail.com	2	true	Rob	asdfghjklñlkj6234828423923...	2022-05-29 00:00:00	[null]

Eliminar:

```
DELETE FROM staff WHERE staff_id = 3;
```

	staff_id [PK] integer	first_name character varying (45)	last_name character varying (45)	address_id smallint	email character varying (50)	store_id smallint	active boolean	username character varying (16)	password character varying (40)	last_update timestamp without time zone	picture bytea
1	1	Ana	Hillyer	3	Mike.Hillyer@sakilastaff.com	1	true	Ana	8cb2237d0679ca88db6464eac60da96345513964	2024-10-16 16:29:15.897315	[binary dat...
2	2	Jon	Stephens	4	Jon.Stephens@sakilastaff.com	2	true	Jon	8cb2237d0679ca88db6464eac60da96345513964	2006-05-16 16:13:11.79328	[null]

Actor:

Insertar:

```
INSERT INTO actor(first_name, last_name, last_update) VALUES ('Robert', 'Downey', '2023-10-17');
```

	actor_id [PK] integer	first_name character varying (45)	last_name character varying (45)	last_update timestamp without time zone
196	196	Bela	Walken	2013-05-26 14:47:57.62
197	197	Reese	West	2013-05-26 14:47:57.62
198	198	Mary	Keitel	2013-05-26 14:47:57.62
199	199	Julia	Fawcett	2013-05-26 14:47:57.62
200	200	Thora	Temple	2013-05-26 14:47:57.62
201	201	Robert	Downey	2023-10-17 00:00:00

Modificar:

UPDATE actor SET first_name = 'Javier', last_name = 'Medina' WHERE actor_id = 100;

	actor_id [PK] integer	first_name character varying (45)	last_name character varying (45)	last_update timestamp without time zone
97	97	Meg	Hawke	2013-05-26 14:47:57.62
98	98	Chris	Bridges	2013-05-26 14:47:57.62
99	99	Jim	Mostel	2013-05-26 14:47:57.62
100	100	Javier	Medina	2024-10-16 16:35:34.771232
101	101	Susan	Davis	2013-05-26 14:47:57.62
102	102	Walter	Torn	2013-05-26 14:47:57.62
103	103	Matthew	Leigh	2013-05-26 14:47:57.62
104	104	Penelope	Cronyn	2013-05-26 14:47:57.62

Eliminar:

DELETE FROM actor WHERE first_name = 'Robert' AND last_name = 'Downey';

	actor_id [PK] integer	first_name character varying (45)	last_name character varying (45)	last_update timestamp without time zone
188	188	Rock	Dukakis	2013-05-26 14:47:57.62
189	189	Cuba	Birch	2013-05-26 14:47:57.62
190	190	Audrey	Bailey	2013-05-26 14:47:57.62
191	191	Gregory	Gooding	2013-05-26 14:47:57.62
192	192	John	Suvari	2013-05-26 14:47:57.62
193	193	Burt	Temple	2013-05-26 14:47:57.62
194	194	Meryl	Allen	2013-05-26 14:47:57.62
195	195	Jayne	Silverstone	2013-05-26 14:47:57.62
196	196	Bela	Walken	2013-05-26 14:47:57.62
197	197	Reese	West	2013-05-26 14:47:57.62
198	198	Mary	Keitel	2013-05-26 14:47:57.62
199	199	Julia	Fawcett	2013-05-26 14:47:57.62
200	200	Javier	Medina	2024-10-16 16:34:29.781148

b. Listar todas las “rental” con los datos del “customer” dado un año y mes.

```
SELECT r.rental_id, r.rental_date, r.return_date, c.first_name, c.last_name, c.email
FROM rental r
JOIN customer c ON r.customer_id = c.customer_id
WHERE EXTRACT(YEAR FROM r.rental_date) = 2005
AND EXTRACT(MONTH FROM r.rental_date) = 05;
```

	rental_id integer	rental_date timestamp without time zone	return_date timestamp without time zone	first_name character varying (45)	last_name character varying (45)	email character varying (50)
1	2	2005-05-24 22:54:33	2005-05-28 19:40:33	Tommy	Collazo	tommy.collazo@sakilacustomer.org
2	3	2005-05-24 23:03:39	2005-06-01 22:12:39	Manuel	Murrell	manuel.murrell@sakilacustomer.org
3	4	2005-05-24 23:04:41	2005-06-03 01:43:41	Andrew	Purdy	andrew.purdy@sakilacustomer.org
4	5	2005-05-24 23:05:21	2005-06-02 04:33:21	Delores	Hansen	delores.hansen@sakilacustomer.org
5	6	2005-05-24 23:08:07	2005-05-27 01:32:07	Nelson	Christenson	nelson.christenson@sakilacustomer.org
6	7	2005-05-24 23:11:53	2005-05-29 20:34:53	Cassandra	Walters	cassandra.walters@sakilacustomer.org
7	8	2005-05-24 23:31:46	2005-05-27 23:33:46	Minnie	Romero	minnie.romero@sakilacustomer.org
8	9	2005-05-25 00:00:40	2005-05-28 00:22:40	Ellen	Simpson	ellen.simpson@sakilacustomer.org
9	10	2005-05-25 00:02:21	2005-05-31 22:44:21	Danny	Isom	danny.isom@sakilacustomer.org
10	11	2005-05-25 00:09:02	2005-06-02 20:56:02	April	Burns	april.burns@sakilacustomer.org
11	12	2005-05-25 00:19:27	2005-05-30 05:44:27	Deanna	Byrd	deanna.byrd@sakilacustomer.org
12	13	2005-05-25 00:22:55	2005-05-30 04:28:55	Raymond	Mcwhorter	raymond.mcwhorter@sakilacustomer.org
13	14	2005-05-25 00:31:15	2005-05-26 02:56:15	Theodore	Culp	theodore.culp@sakilacustomer.org
14	15	2005-05-25 00:39:22	2005-06-03 03:30:22	Ronald	Weiner	ronald.weiner@sakilacustomer.org
15	16	2005-05-25 00:43:11	2005-05-26 04:42:11	Steven	Curley	steven.curley@sakilacustomer.org
16	17	2005-05-25 01:06:36	2005-05-27 00:43:36	Isaac	Oglesby	isaac.oglesby@sakilacustomer.org
17	18	2005-05-25 01:10:47	2005-05-31 06:35:47	Ruth	Martinez	ruth.martinez@sakilacustomer.org
18	19	2005-05-25 01:17:24	2005-05-31 06:00:24	Ronnie	Ricketts	ronnie.ricketts@sakilacustomer.org

Para datos completes revisar “rental.csv”

c. Listar Número, Fecha (payment_date) y Total (amount) de todas las “payment”.

```
SELECT payment_id, payment_date, amount
FROM payment;
```

	payment_id [PK] integer	payment_date timestamp without time zone	amount numeric (5,2)
1	17503	2007-02-15 22:25:46.996577	7.99
2	17504	2007-02-16 17:23:14.996577	1.99
3	17505	2007-02-16 22:41:45.996577	7.99
4	17506	2007-02-19 19:39:56.996577	2.99
5	17507	2007-02-20 17:31:48.996577	7.99
6	17508	2007-02-21 12:33:49.996577	5.99
7	17509	2007-02-17 23:58:17.996577	5.99
8	17510	2007-02-20 02:11:44.996577	5.99
9	17511	2007-02-20 13:57:39.996577	2.99
10	17512	2007-02-16 00:10:50.996577	4.99
11	17513	2007-02-16 01:15:33.996577	6.99
12	17514	2007-02-17 01:26:00.996577	0.99
13	17515	2007-02-17 04:32:51.996577	0.99
14	17516	2007-02-18 18:26:38.996577	6.99
15	17517	2007-02-20 07:03:29.996577	8.99
16	17518	2007-02-21 14:42:28.996577	0.99
17	17519	2007-02-15 10:54:44.996577	3.99
18	17520	2007-02-15 19:36:27.996577	4.99

Para datos completes revisar “payment.csv”

d. Listar todas las “film” del año 2006 que contengan un (rental_rate) mayor a 4.0.

```
SELECT *
FROM film
WHERE release_year = 2006
AND rental_rate > 4.0;
```

	film_id [PK] integer	title character varying (255)	description text	release_year integer	language_id smallint	rental_duration smallint	rental_rate numeric (4,2)	length smallint	replacement_cos numeric (5,2)
1	133	Chamber Italian	A Fateful Reflection of a Moose And a Husband who must Overcome a Monkey in Nigeria	2006	1	7	4.99	117	1
2	384	Grosse Wonderful	A Epic Drama of a Cat And a Explorer who must Redeem a Moose in Australia	2006	1	5	4.99	49	1
3	8	Airport Pollock	A Epic Tale of a Moose And a Girl who must Confront a Monkey in Ancient India	2006	1	6	4.99	54	1
4	98	Bright Encounters	A Fateful Yarn of a Lumberjack And a Feminist who must Conquer a Student in A Jet Boat	2006	1	4	4.99	73	1
5	2	Ace Goldfinger	A Astounding Epistle of a Database Administrator And a Explorer who must Find a Car in Ancient China	2006	1	3	4.99	48	1
6	7	Airplane Sierra	A Touching Saga of a Hunter And a Butler who must Discover a Butler in A Jet Boat	2006	1	6	4.99	62	2
7	10	Aladdin Calendar	A Action-Packed Tale of a Man And a Lumberjack who must Reach a Feminist in Ancient China	2006	1	6	4.99	63	2
8	13	Ali Forever	A Action-Packed Drama of a Dentist And a Crocodile who must Battle a Feminist in The Canadian Rockies	2006	1	4	4.99	150	2
9	20	Amelie Hellfighters	A Boring Drama of a Woman And a Squirrel who must Conquer a Student in A Baloon	2006	1	4	4.99	79	2
10	21	American Circus	A Insightful Drama of a Girl And a Astronaut who must Face a Database Administrator in A Shark Tank	2006	1	3	4.99	129	1
11	28	Anthem Luke	A Touching Panorama of a Waitress And a Woman who must Outrace a Dog in An Abandoned Amusement Park	2006	1	5	4.99	91	1
12	31	Apache Divine	A Awe-Inspiring Reflection of a Pastry Chef And a Teacher who must Overcome a Sumo Wrestler in A U-Boat	2006	1	5	4.99	92	1
13	32	Apocalypse Flamingos	A Astounding Story of a Dog And a Squirrel who must Defeat a Woman in An Abandoned Amusement Park	2006	1	6	4.99	119	1
14	265	Dying Maker	A Intrepid Tale of a Boat And a Monkey who must Kill a Cat in California	2006	1	5	4.99	168	2
15	44	Attacks Hate	A Fast-Paced Panorama of a Technical Writer And a Mad Scientist who must Find a Feminist in An Abandoned Mine Shaft	2006	1	5	4.99	113	2
16	45	Attraction Newton	A Astounding Panorama of a Composer And a Frisbee who must Reach a Husband in Ancient Japan	2006	1	5	4.99	83	1

Para datos completes revisar “film.csv”

2. Realiza un Diccionario de datos que contenga el nombre de las tablas y columnas, si éstas pueden ser nulas, y su tipo de dato correspondiente

```
SELECT
  table_name AS "Tabla",
  column_name AS "Columna",
  data_type AS "Tipo de Dato",
  is_nullable AS "¿Puede ser nulo?"
FROM
  information_schema.columns
WHERE
  table_schema = 'public'
AND table_name IN('actor','store','address','category','city','country','customer',
'film_actor','film_category','inventory','language','rental','staff','payment','film')
ORDER BY table_name ASC;
```

	Tabla name	Columna name	Tipo de Dato character varying	¿Puede ser nulo? character varying (3)
9	address	address	character varying	NO
10	address	city_id	smallint	NO
11	address	address_id	integer	NO
12	address	phone	character varying	NO
13	category	name	character varying	NO
14	category	last_update	timestamp without time zone	NO
15	category	category_id	integer	NO
16	city	city_id	integer	NO
17	city	city	character varying	NO
18	city	last_update	timestamp without time zone	NO
19	city	country_id	smallint	NO
20	country	last_update	timestamp without time zone	NO
21	country	country	character varying	NO
22	country	country_id	integer	NO
23	customer	last_name	character varying	NO
24	customer	customer_id	integer	NO
25	customer	store_id	smallint	NO
26	customer	address_id	smallint	NO

Para datos completes revisar “diccionario.csv”