

EJERCICIOS DE NIVELACIÓN SQL

Indicaciones

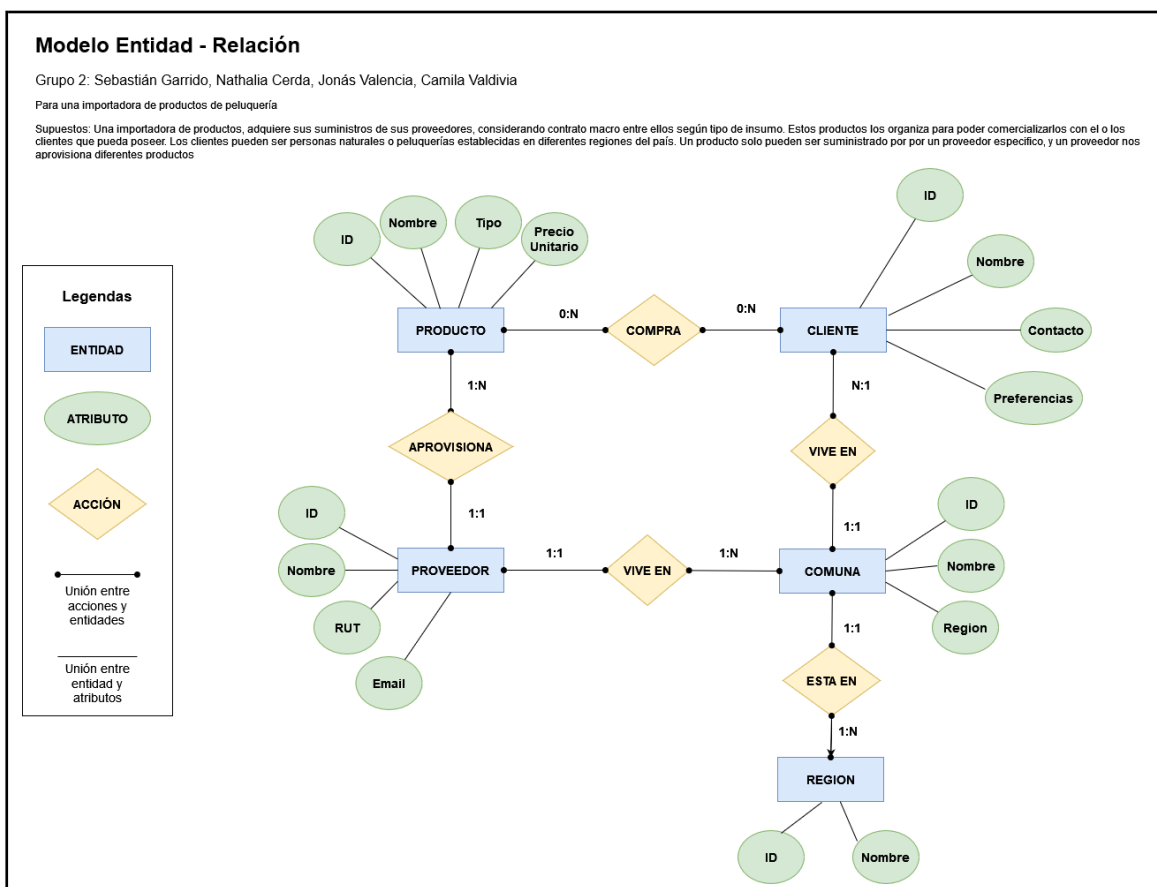
- Modo de entrega: Individual, en aula de MIND HUB, Entrega clase 8.1
- Plazo: Hasta las 9:00 del 12 de abril
- Adjuntos: Este documento, más archivos de diagrama & schema.sql

Grupo N° 2:

Integrantes:

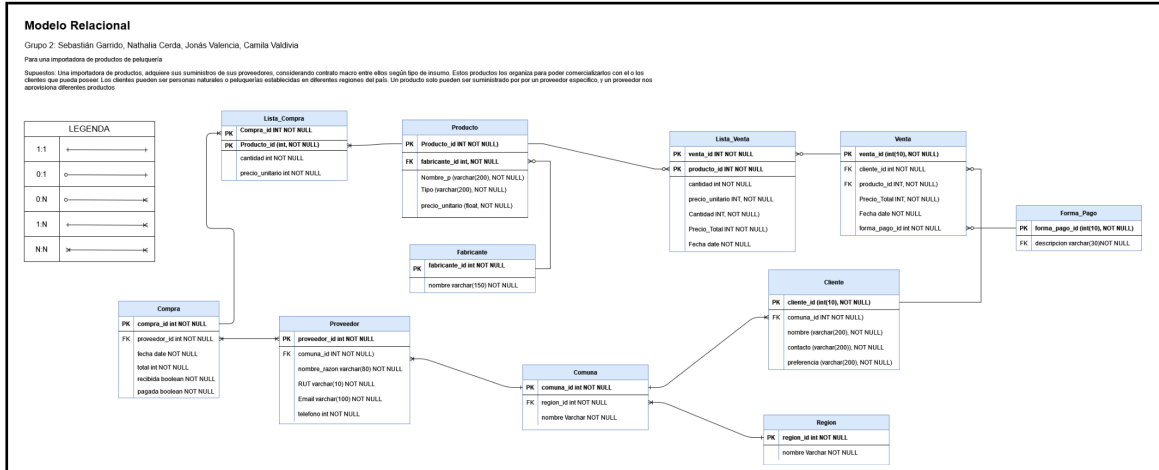
- Nathalia Cerda
- Sebastián Garrido
- Camila Valdivia
- Jonás Valencia

Modelo Entidad Relación:





Modelo Relacional



<https://app.diagrams.net/#G1835BfBncUJ61VdIMyg-DYukwohtnno8v>

Código Schema .SQL

```
/*
DROP TABLE CLIENTE;
DROP TABLE COMPRA;
DROP TABLE COMUNA;
DROP TABLE FABRICANTE;
DROP TABLE FORMA_PAGO;
DROP TABLE PRODUCTO;
DROP TABLE PROVEEDOR;
DROP TABLE REGION;
DROP TABLE VENTA;
*/
CREATE TABLE fabricante (
    fabricante_id integer PRIMARY KEY AUTOINCREMENT NOT NULL,
    nombre VARCHAR(100) NOT NULL
);

CREATE TABLE forma_pago (
    forma_pago_id integer PRIMARY KEY AUTOINCREMENT NOT NULL,
    descripcion VARCHAR(100) NOT NULL
);

CREATE TABLE region (
    region_id integer PRIMARY KEY AUTOINCREMENT NOT NULL,
    nombre VARCHAR(100) NOT NULL
);
```

```
CREATE TABLE comuna (  
  comuna_id integer PRIMARY KEY AUTOINCREMENT NOT NULL,  
  nombre VARCHAR(100) NOT NULL  
);  
  
CREATE TABLE cliente (  
  cliente_id integer PRIMARY KEY AUTOINCREMENT NOT NULL,  
  comuna_id integer NOT NULL,  
  nombre VARCHAR(200) NOT NULL,  
  contacto VARCHAR(200) NOT NULL,  
  preferencia VARCHAR(200) NOT NULL,  
  CONSTRAINT fk_cliente_comuna FOREIGN KEY (comuna_id)  
REFERENCES comuna (comuna_id) ON DELETE NO ACTION ON UPDATE  
CASCADE  
);  
  
CREATE TABLE proveedor (  
  proveedor_id integer PRIMARY KEY AUTOINCREMENT NOT NULL,  
  comuna_id integer NOT NULL,  
  nombre_razon VARCHAR(200) NOT NULL,  
  rut VARCHAR(10) NOT NULL,  
  email VARCHAR(100) NOT NULL,  
  telefono integer NOT NULL,  
  CONSTRAINT fk_proveedor_comuna FOREIGN KEY (comuna_id)  
REFERENCES comuna (comuna_id) ON DELETE NO ACTION ON UPDATE  
CASCADE  
);  
  
CREATE TABLE producto (  
  producto_id integer PRIMARY KEY AUTOINCREMENT NOT NULL,  
  fabricante_id integer NOT NULL,  
  nombre VARCHAR(200) NOT NULL,  
  CONSTRAINT fk_producto_fabricante FOREIGN KEY (fabricante_id)  
REFERENCES fabricante (fabricante_id) ON DELETE NO ACTION ON  
UPDATE CASCADE  
);  
  
CREATE TABLE compra (  
  compra_id integer PRIMARY KEY AUTOINCREMENT NOT NULL,  
  proveedor_id integer NOT NULL,  
  fecha date NOT NULL,  
  total integer NOT NULL,  
  CONSTRAINT fk_compra_proveedor FOREIGN KEY (proveedor_id)  
REFERENCES proveedor (proveedor_id) ON DELETE NO ACTION ON  
UPDATE CASCADE
```

```
);  
  
CREATE TABLE venta (  
    venta_id integer PRIMARY KEY AUTOINCREMENT NOT NULL,  
    cliente_id integer NOT NULL,  
    forma_pago_id integer NOT NULL,  
    fecha date NOT NULL,  
    total integer NOT NULL,  
    CONSTRAINT fk_venta_cliente FOREIGN KEY (cliente_id)  
REFERENCES cliente (cliente_id) ON DELETE NO ACTION ON UPDATE  
CASCADE  
    CONSTRAINT fk_venta_forma_pago FOREIGN KEY (forma_pago_id)  
REFERENCES forma_pago (forma_pago_id) ON DELETE NO ACTION ON  
UPDATE CASCADE  
);
```

Guía 8.1:

Con la base de datos del aula (sakila.db), responda a las siguientes consultas en SQL. Usaremos <https://sqliteonline.com/>

1. De la tabla film traer solo las columnas title, release_year, special_features y rating.

```
SELECT title, release_year, special_features, rating FROM film
```

2. De la tabla actor traer solo el first_name y el last_name.

```
SELECT first_name, last_name FROM actor
```

3. De la tabla customer traer las columnas customer_id, first_name, email.

```
SELECT customer_id, first_name, email FROM customer
```

4. ¿Qué consulta tengo que ejecutar para traer todas las columnas de la tabla staff?

```
SELECT first_name, last_name, address_id, picture, email,  
store_id, active, username, password, last_update FROM staff
```

5. Ver toda la tabla film_list completa.

```
SELECT * FROM film_list
```

6. Filtrar los registros de la tabla payment que hayan pagado exactamente \$0.99.

```
SELECT * FROM payment WHERE amount=0.99
```

7. Idem a 6) pero traer solo las columnas customer_id, rental_id y amount para verificar que estén bien filtrados.

```
SELECT customer_id, rental_id, amount FROM payment WHERE amount =  
0.99
```

8. De la tabla payment traer todos los alquileres del cliente cuyo número de cliente sea 386 y el monto pagado sea mayor a 2.

```
SELECT customer_id, rental_id, amount FROM payment WHERE  
customer_id = 386 AND amount > 2
```

9. ¿Cuáles son todas las películas que pertenece al género Horror? Usar la tabla film_list.

```
SELECT * FROM film_list WHERE category='Horror'
```

10. ¿Qué películas pertenecen al género Foreign y duran más de 2 horas? Mostrar los títulos, la duración y el rating.

```
SELECT title as Título, length as duracion, rating FROM film_list  
WHERE category='Foreign' and length>120
```

11. Seleccionar de la tabla actor aquellos registros cuyo apellido sea Kilmer, Bloom o Wayne.

```
SELECT * FROM actor WHERE last_name in ('KILMER', 'BLOOM', 'WAYNE')
```

12. ¿Qué películas de la tabla film aceptan un período de alquiler entre 4 y 6 días? Mostrar solo las columnas título, duración del alquiler y duración de la película.

```
SELECT title as Título, rental_duration as "Duracion  
Alquiler", length as Duracion from film where rental_duration  
BETWEEN 4 and 6
```

13. Un cliente de la tienda 2 efectuó un pago por un alquiler pero el personal se olvidó de registrarlo y ahora no recuerda bien el nombre, solo sabe que el nombre empezaba con "CH". Listar los posibles clientes para ver si el empleado recuerda el nombre. Ordenarlos por nombre. Recordar que el símbolo de porcentaje % se utiliza como comodín.

```
SELECT * FROM customer WHERE first_name LIKE 'CH%' order by  
first_name
```

14. ¿Cuáles son los distintos géneros de películas disponibles en el videoclub?

```
SELECT name FROM category
```

15. Tabla film. ¿Qué películas cuestan menos de 4 dólares, duran entre 80 y 110 minutos y su rating es "PG-13"?

```
SELECT * FROM film where rental_rate < 4 and rating = 'PG-13'  
and length between 80 and 110
```

16. ¿Cuáles son los distintos valores posibles para el rating de las películas?

```
SELECT DISTINCT rating FROM film
```

17. De la tabla customer_list filtrar los registros de los clientes que sean de Argentina o de Bolivia. Ordenar por país y por nombre.

```
SELECT * FROM customer_list WHERE country = "Argentina" or  
country = "Bolivia" order by country, name
```

18. ¿Cuántos clientes inactivos hay? Ver tabla customer.

```
SELECT count(active) FROM customer WHERE active= 0
```

Respuesta= 15

19. ¿Cuál es el máximo y el mínimo monto pagos en el video club?, ¿y el promedio?

```
SELECT MAX(amount), MIN(amount), AVG(amount) FROM payment
```

Respuesta: MAX: 11.99, MIN: 0, AVG: 4.200667

20. ¿Cuántas películas aptas para todo público hay?, ¿cuál es la duración promedio? Redondear el promedio.

```
select count(title) as "Cantidad de películas para todo público"  
from film where rating = 'G' -----Respuesta: 178
```

```
select round(avg(length)) as "Duración promedio de películas  
para todo público (min)" from film where rating = 'G'  
-----Respuesta: 111
```

21. ¿Cuántos clientes hay de Argentina, Brazil, Bolivia o Chile? ¿Hay algún otro país de latinoamérica?

```
select country as País, count(name) as "Clientes por País" from  
customer_list where country in  
( 'Argentina', 'Bolivia', 'Brazil', 'Chile' ) group by country  
Respuesta: Argentina: 13, Bolivia: 2, Brazil: 28, Chile: 3
```

-- Se filtró los países para leer y extraer, para identificar los de latinoamérica,
-- usando la siguiente consulta:

```
SELECT DISTINCT country FROM country
```

-- Los países latinoamericanos encontrados son : 'Venezuela', 'Puerto Rico', 'Perú',
-- 'Paraguay', 'México', 'Colombia', 'Ecuador', 'Dominican Republic'