

## Introducción

Presentamos el álgebra relacional y las operaciones definidas para entender cómo se accede y se muestran los datos guardados, que constituyen la base para escribir las sentencias SQL para una consulta que un usuario puede hacer o para formar parte de una aplicación que necesita acceder a los datos en las tablas de un esquema de base de datos. Veremos que hay operaciones básicas y derivadas. El álgebra relacional forma parte del modelo relacional presentado por Edgar Frank Codd en 1970.

## 1. Álgebra relacional y cálculo relacional

El modelo considera la base de datos como una colección de relaciones. De manera simple, una relación representa una tabla, que no es más que un conjunto de filas, cada fila es un conjunto de campos y cada campo representa un valor que, interpretado dentro del atributo, describe el mundo real en un aspecto de cada fila. Esta fila también se puede denominar tupla y a cada columna también se le puede llamar atributo.

Para manipular la información, utilizamos un lenguaje relacional que son dos lenguajes formales: el álgebra relacional y el cálculo relacional. El álgebra relacional permite describir la forma de realizar una consulta, en cambio, el cálculo relacional sólo indica lo que se desea devolver, [describe los resultados a obtener con las operaciones del álgebra]. (Wikipedia, s.f., <https://bit.ly/3ogfgrZ>).

“El **álgebra relacional** es un conjunto de operaciones que describen paso a paso cómo computar una respuesta sobre las relaciones, tal y como éstas son definidas en el modelo relacional” (Wikipedia, s.f., <https://bit.ly/2VqbOim>). Además, el álgebra relacional describe el aspecto de la manipulación de datos. Estas operaciones se usan como una representación intermedia de una sentencia SELECT en SQL que consulta una base de datos y, debido a sus propiedades algebraicas, sirven para obtener una versión inicial de una consulta y ayudan a entender cómo se realizará esa consulta SELECT en SQL.

“El lenguaje más común para construir las consultas a bases de datos relacionales es SQL, *Structured Query Language* o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales” (Wikipedia, s.f., <https://bit.ly/33AjD9z>). Recurrimos a IBM, para citar una definición de la empresa IBM (2020) de SQL:

SQL es un lenguaje estandarizado que sirve para definir y manipular los datos de una base de datos relacional. De acuerdo con el modelo relacional de datos, la base de datos se crea como un conjunto de tablas, las relaciones se representan mediante valores en las tablas y los datos se recuperan especificando una tabla de resultados que puede derivarse de una o más tablas base. (IBM, 2020, <https://ibm.co/3mnIVix>).

## 2. Definiciones

### Grado

Traemos esta definición de grado para entender los requisitos de las operaciones, diciendo que grado es la cantidad de atributos de una relación.

### Unión compatible

Una unión es compatible entre dos relaciones si ellas poseen el mismo grado y sus dominios son los mismos de izquierda a derecha.

### Las operaciones del álgebra relacional:

#### Básicas

Cada operador del álgebra acepta una o dos relaciones y retorna una relación como resultado.  $\sigma$  (sigma minúscula) y  $\Pi$  (Pi) son operadores unarios o sea que se basan en una relación, el resto de los operadores son binarios porque se basan en dos relaciones. Las operaciones básicas del álgebra relacional son:

Selección ( $\sigma$ )

Permite seleccionar un subconjunto de tuplas de una relación ( $R$ ), todas aquellas que cumplan la(s) condición(es)  $P$ , esto es:

$$\sigma_P(R)$$

Ejemplo con una relación o tabla que se llame Alumnos y tenga un atributo llamado Apellido:

$$\sigma_{\text{Apellido=Gómez}}(\text{Alumnos})$$

Selecciona todas las tuplas que contengan Gómez como apellido en la relación Alumnos.  
Una condición puede ser una combinación booleana, donde se pueden usar operadores como:

$\wedge, \vee$ ,  
combinándolos con operadores  
 $<, >, \leq, \geq, =, \neq$   
(Wikipedia, s.f., <https://bit.ly/3qgNoGh>).

#### Proyección ( $\Pi$ )

Dentro de esta operación cabe mencionar a la operación asterisco ( $*$ ) que muestra todas las columnas, es una “proyección sin exclusión” ya que no excluye a ningún atributo de las relaciones seleccionadas.

La operación proyección (sin el  $*$ ) “permite mostrar algunos de los atributos de una relación, dando como resultado un subconjunto vertical” (Wikipedia, s.f., <https://bit.ly/3qgNoGh>), es decir los atributos elegidos de todas las tuplas de la relación, esto es:

$$\Pi_{A_1, A_2, \dots, A_n}(R)$$

Donde  
 $A_1, A_2, \dots, A_n$   
son atributos de la relación **R**.

A continuación, mostramos un ejemplo con la relación o tabla Alumnos (ya mencionada) y algunos de sus atributos:

$\Pi_{Apellido, Semestre, NumeroControl}(Alumnos)$

“Selecciona los atributos Apellido, Semestre y Número Control de la relación Alumnos, mostrados como un subconjunto de la relación Alumnos” (Wikipedia, s.f., <https://bit.ly/3qgNoGh>).

#### Producto cartesiano (x)

El producto cartesiano de dos relaciones se escribe como:

$$R \times S$$

Y entrega una nueva relación, cuyas tuplas corresponden a una combinación de todas las tuplas de R con cada una de las tuplas de S, y sus atributos corresponden a los de R seguidos por los de S.

Ejemplo: ahora con otra nueva relación o tabla, Maestros:

$$Alumnos \times Maestros$$

Muestra una nueva relación que contiene cada una de las tuplas de la relación Alumnos junto con las tuplas de la relación Maestros, mostrando primero los atributos de Alumnos seguidos por las tuplas de Maestros. (Wikipedia, s.f., <https://bit.ly/3qgNoGh>).

#### ≡ Accordion 1

Accordion 1 Content

#### ≡ Accordion 2

Accordion 2 Content

#### ≡ Accordion 3

Accordion 3 Content

#### Unión (U)

La operación

$$R \cup S$$

retorna el conjunto de tuplas que están en R, en S o en ambas. R y S deben ser uniones compatibles en el grado y los tipos de datos que la componen.

#### Diferencia (-)

La diferencia de dos relaciones, R y S, denotada por:

$$R - S$$

Entrega todas aquellas tuplas que están en R, pero no en S. (Wikipedia, s.f., <https://bit.ly/3qgNoGh>).

R y S deben ser uniones compatibles y tener la misma cantidad de columnas.

#### Operaciones no básicas

Entre los operadores no básicos tenemos:

#### Intersección (∩)

La intersección de dos relaciones se puede especificar en función de otros operadores básicos:

$$R \cap S = R - (R - S)$$

(Wikipedia, s.f., <https://bit.ly/3qgNoGh>).

O sea, traer las filas de R que no se diferencian de las filas de S, es decir aquellas que coinciden con las filas de S, un modo indirecto de definir que filas están en R y en S.

“La intersección, como en teoría de conjuntos, corresponde al conjunto de todas las tuplas que están en R y en S, siendo R y S uniones compatibles” (Wikipedia, s.f., <https://bit.ly/3qgNoGh>) (misma cantidad de atributos o grado). En SQL se debe recurrir a operaciones de combinación, que están fuera del alcance de esta materia, pero queremos dejar testimonio de que existe esta operación.

#### Combinación (Join, natural, inner o interna, outer o externa, etcétera)

Una combinación de dos relaciones es equivalente a:

$$R \bowtie_F S = \sigma_F(R \times S)$$

Esta operación es mucho más útil que el operador básico producto cartesiano, pues especifica una regla para la combinación de los atributos y no produce una gran cantidad de tuplas que no tienen nada en común.

#### División (/)

Supongamos que tenemos dos relaciones A (x, y) y B (y), donde el dominio del atributo “y” en A y en B es el mismo.

El operador división A / B retorna todos los distintos valores de x tales que para todo valor “y” en B existe una tupla

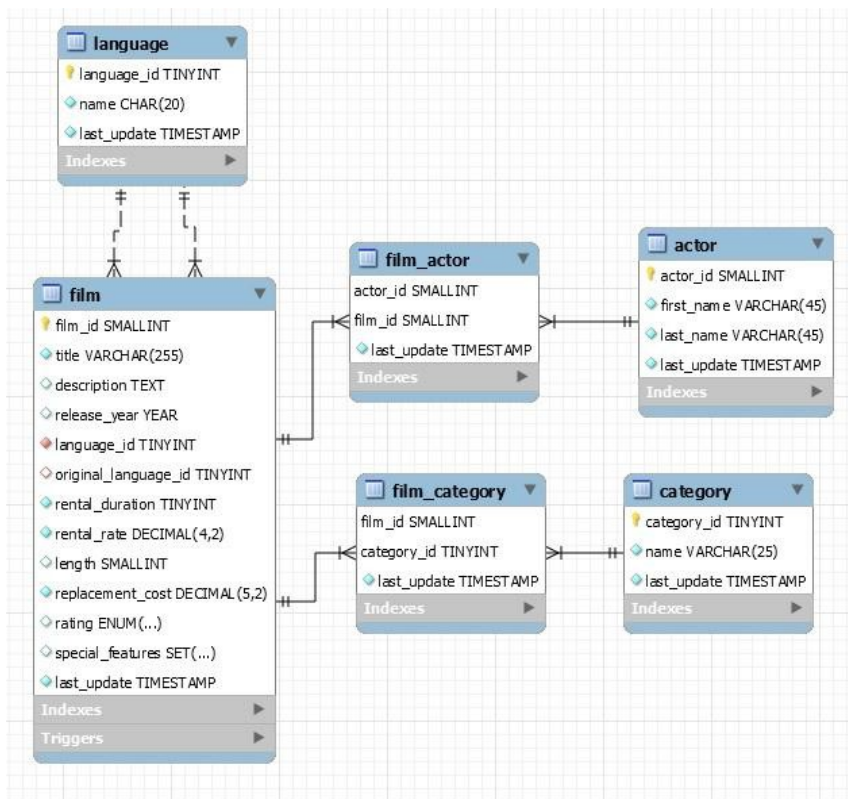
$$\langle x, y \rangle$$

(Wikipedia, s.f., <https://bit.ly/3qgNoGh>).

### 3. Ejemplos

Extraemos de algunas tablas del modelo Sakila de la base con la que trabajamos en la materia, a saber, *Actor*, *Film*, *Language* y *Category*; con sus columnas.

**Figura 1: Modelo de datos Sakila**



Fuente: elaboración propia.

**Figura 1. Modelo de datos a usar como ejemplo de operaciones de álgebra relacional.** Del modelo de datos Sakila, elegimos seis tablas, las mismas con las que hicimos un modelo con MySQL Workbench.

**Selección ( $\sigma$ ) aplicando una condición p sobre la relación R**

$$\sigma_P(R)$$

$\sigma_{\text{last\_name} = \text{'DUNST'}}(\text{ACTOR})$

Veamos cómo será en SQL con la sentencia SELECT:

SELECT \* FROM actor WHERE last\_name = 'DUNST';

**Figura 2: Resultado en SQL de la operación selección**

```
SELECT * FROM actor WHERE last_name = 'DUNST'
```

actor_id	first_name	last_name	last_update
106	GROUCHO	DUNST	2006-02-15 04:34:33

Fuente: elaboración propia

**Figura 2. La operación selección del álgebra relacional aplicada en SQL.** De la operación selección del álgebra relacional se deriva a la sentencia SQL SELECT con la cláusula WHERE y la condición a cumplir por las filas de la tabla para ser mostradas.

**Proyección ( $\Pi$ )**

Usamos la letra griega  $\Pi$  (Pi) para simbolizar la operación proyección que indica que se muestren solo algunas columnas de la relación o tabla, según la forma ya vista

$$\Pi_{A_1, A_2, \dots, A_n}(R)$$

- Veremos un ejemplo con la tabla *category*, las columnas *name* y *last\_update*:

$\Pi_{\text{name, last\_update}}(\text{CATEGORY})$

Su equivalente en el lenguaje SQL es la sentencia *SELECT name, last\_update FROM category*. Veamos en qué resulta en la siguiente figura.

**Figura 3: Resultado en SQL de la operación proyección en la tabla category**

```
SELECT name, last_update
FROM category
```

name	last_update
Action	2006-02-15 04:46:27
Animation	2006-02-15 04:46:27
Children	2006-02-15 04:46:27
Classics	2006-02-15 04:46:27
Comedy	2006-02-15 04:46:27
Documentary	2006-02-15 04:46:27
Drama	2006-02-15 04:46:27
Family	2006-02-15 04:46:27
Foreign	2006-02-15 04:46:27
Games	2006-02-15 04:46:27
Horror	2006-02-15 04:46:27
Music	2006-02-15 04:46:27
New	2006-02-15 04:46:27
Sci-Fi	2006-02-15 04:46:27
Sports	2006-02-15 04:46:27
Travel	2006-02-15 04:46:27

Fuente: elaboración propia.

**Figura 3. La operación proyección del álgebra relacional aplicada en SQL.** De la operación proyección del álgebra relacional se deriva la sentencia SQL SELECT, con la lista de las columnas a mostrar después de la cláusula SELECT. La proyección trae las columnas elegidas de todas las filas, aunque pueden combinarse las operaciones proyección y selección para traer algunas columnas de algunas filas.

$\Pi$  name, last\_update ( $\sigma$  name < 'Classic' (CATEGORY))

**Figura 4: Resultado en SQL de las operaciones proyección y selección combinadas en SQL**

```
SELECT name, last_update
FROM category
WHERE name < 'Classic'
```

name	last_update
Action	2006-02-15 04:46:27
Animation	2006-02-15 04:46:27
Children	2006-02-15 04:46:27

Fuente: elaboración propia.

**Figura 4. Las operaciones proyección y selección del álgebra relacional aplicada en SQL.** De la combinación de operaciones vemos que devuelven las columnas elegidas de las filas que cumplen con la condición que filtra la columna *name*, tienen valores alfabéticamente anteriores a Classic.

**Producto cartesiano (x)**

El producto cartesiano de dos relaciones se escribe como:

$R \times S$

En las relaciones o tablas del modelo Sakila, *category* y actor, el producto cartesiano se escribiría *category* x actor y en la sentencia SQL:

**Figura 5: Resultado de la operación Producto Cartesiano en SQL**

```
SELECT * FROM actor, category
```

actor_id	first_name	last_name	last_update	category_id	name	last_update
1	PENELOPE	GUINESS	2006-02-15 04:34:33	1	Action	2006-02-15 04:46:27
1	PENELOPE	GUINESS	2006-02-15 04:34:33	2	Animation	2006-02-15 04:46:27
1	PENELOPE	GUINESS	2006-02-15 04:34:33	3	Children	2006-02-15 04:46:27
1	PENELOPE	GUINESS	2006-02-15 04:34:33	4	Classics	2006-02-15 04:46:27
1	PENELOPE	GUINESS	2006-02-15 04:34:33	5	Comedy	2006-02-15 04:46:27
1	PENELOPE	GUINESS	2006-02-15 04:34:33	6	Documentary	2006-02-15 04:46:27
1	PENELOPE	GUINESS	2006-02-15 04:34:33	7	Drama	2006-02-15 04:46:27
1	PENELOPE	GUINESS	2006-02-15 04:34:33	8	Family	2006-02-15 04:46:27
1	PENELOPE	GUINESS	2006-02-15 04:34:33	9	Foreign	2006-02-15 04:46:27
1	PENELOPE	GUINESS	2006-02-15 04:34:33	10	Games	2006-02-15 04:46:27
1	PENELOPE	GUINESS	2006-02-15 04:34:33	11	Horror	2006-02-15 04:46:27
1	PENELOPE	GUINESS	2006-02-15 04:34:33	12	Music	2006-02-15 04:46:27
1	PENELOPE	GUINESS	2006-02-15 04:34:33	13	New	2006-02-15 04:46:27
1	PENELOPE	GUINESS	2006-02-15 04:34:33	14	Sci-Fi	2006-02-15 04:46:27
1	PENELOPE	GUINESS	2006-02-15 04:34:33	15	Sports	2006-02-15 04:46:27
1	PENELOPE	GUINESS	2006-02-15 04:34:33	16	Travel	2006-02-15 04:46:27
2	NICK	WAHLBERG	2006-02-15 04:34:33	1	Action	2006-02-15 04:46:27
2	NICK	WAHLBERG	2006-02-15 04:34:33	2	Animation	2006-02-15 04:46:27
2	NICK	WAHLBERG	2006-02-15 04:34:33	3	Children	2006-02-15 04:46:27

Fuente: elaboración propia.

**Figura 5. La operación producto cartesiano en SQL.** Destacamos que produce tantas filas como la multiplicación de la cantidad de filas de la primera tabla por la cantidad de filas de la segunda tabla (y de las subsiguientes tablas que se puedan agregar), generando una gran cantidad de filas aunque no tengan ninguna relación entre ellas. En este caso, no se usó la proyección por lo que se sumaron todas las columnas de las dos tablas. Por cuestión de espacio, no se muestran las 3200 filas que produjo el producto cartesiano (200 filas de *actor* y 16 de *category*).

**Unión (U)**

Como vimos antes, la operación unión de relaciones o tablas se expresa

## R U S

En el modelo Sakila, podemos unir las tablas *language* y *actor*, como veremos, con una operación previa, con la expresión:

### LANGUAGE U ACTOR

Que como tienen columnas diferentes y algunas del mismo tipo, podremos agregarle una proyección y, en el ejemplo, podremos agregar un literal que nos ayude a entender el resultado, quedando la expresión:

```
II name,'(Language)', last_update (LANGUAGE)) U (II last_name,'(actor)' last_update (ACTOR))
```

Vemos el resultado en la siguiente figura:

**Figura 6: Resultado de la operación Unión en SQL**

The screenshot shows a SQL query editor with the following query:

```
1 SELECT last_name, '(actor)', last_update
2 from actor
3 union
4 select name, '(language)', last_update
5 from language
6 order by 1
```

Below the query, the result grid is displayed, showing the results of the UNION operation. The columns are last\_name, (actor), and last\_update. The results are ordered by the first column (last\_name).

last_name	(actor)	last_update
DUNST	(actor)	2006-02-15 04:34:33
English	(language)	2006-02-15 05:02:19
FAWCETT	(actor)	2006-02-15 04:34:33
French	(language)	2006-02-15 05:02:19
GABLE	(actor)	2006-02-15 04:34:33
GARLAND	(actor)	2006-02-15 04:34:33
German	(language)	2006-02-15 05:02:19
GIBSON	(actor)	2006-02-15 04:34:33

Fuente: elaboración propia.

**Figura 6. Resultado de la operación unión en SQL.** Destacamos que la operación unión es conmutativa, es decir no importa el orden de las relaciones, pero si importa que las columnas proyectadas tengan el mismo tipo de datos y que coincidan en el grado o cantidad de columnas. En la figura se mostraron algunas filas donde se distinguen, por el literal usado, a qué tabla pertenece cada fila. Notar también que, una vez realizada la unión, puede marcarse el orden de aparición de las filas. En este caso, el 1 indica que se ordene ascendentemente por la primera fila.

**Combinación (*join*, natural, *inner* o interna, *outer* o externa, etcétera)**

Una combinación de dos relaciones es equivalente a:

$$R \bowtie_F S = \sigma_F(R \times S)$$

Esta operación es una de las más utilizadas en las aplicaciones de bases de datos relacionales, ya que permite combinar las columnas distribuidas en distintas tablas, luego de haber sido normalizadas (veremos este proceso en los siguientes módulos).

**Pasos para cumplir con el requerimiento “mostrar los nombres de los lenguajes de los films”**

Primero, realizaremos una **combinación** entre las tablas *language* y *film* (pues necesitamos saber cómo se llama el lenguaje correspondiente al *language\_id* que está en *Film*). La combinación realizará primero un producto cartesiano, es decir, para cada tupla de *lenguaje* (todas sus filas) hará una mezcla con cada una tupla de *film* y luego seleccionará aquellas tuplas en que *language.id* sea igual a *film.language\_id*, esto es:

**resultado**  $\bowtie_{language\_id (language) = language\_id (film)}$

La tabla de este nuevo resultado en las tablas Sakila sería:

**Figura 7: Resultado de la operación combinación (INNER JOIN) en SQL**

The screenshot shows a SQL query editor with the following query:

```
SELECT film.title, language.name
FROM language JOIN film
ON (film.language_id = language.language_id)
```

Below the query, the result grid is displayed, showing the results of the INNER JOIN operation. The columns are title and name. The results are ordered by title.

title	name
ACADEMY DINOSAUR	English
ACE GOLDFINGER	English
ADAPTATION HOLES	English
AFFAIR PREJUDICE	English
AFRICAN EGG	English
AGENT TRUMAN	English
AIRPLANE SIERRA	English
AIRPORT POLLOCK	English
ALABAMA DEVIL	English
ALADDIN CALENDAR	English
ALAMO VIDEOTAPE	English
ALASKA PHANTOM	English
ALI FOREVER	English
ALICE FANTASIA	English
ALIEN CENTER	English
ALLEY EVOLUTION	English
ALONE TRIP	English
AI TFR VICTORY	Finnish

Fuente: elaboración propia.

**Figura 7. Resultado de la operación combinación o *join* en SQL.** La operación combinación o *join* (en este caso es INNER JOIN), devuelve solamente las filas que cumplen con una condición: coincide el id del lenguaje en *film* y en *language*.

Cerramos el estudio del álgebra relacional, que nos ha abierto el camino a entender las múltiples posibilidades de trabajar con las tablas de un modelo de datos del que las empresas disponen para registrar sus operaciones y poder conocer sus aspectos más importantes para su funcionamiento, basado en una aplicación que utilice bases de datos relacionales. Existen más operaciones, pero hemos destacado las más utilizadas en las aplicaciones reales que como profesional podrás analizar, entender, diseñar y utilizar en la construcción de *software*.

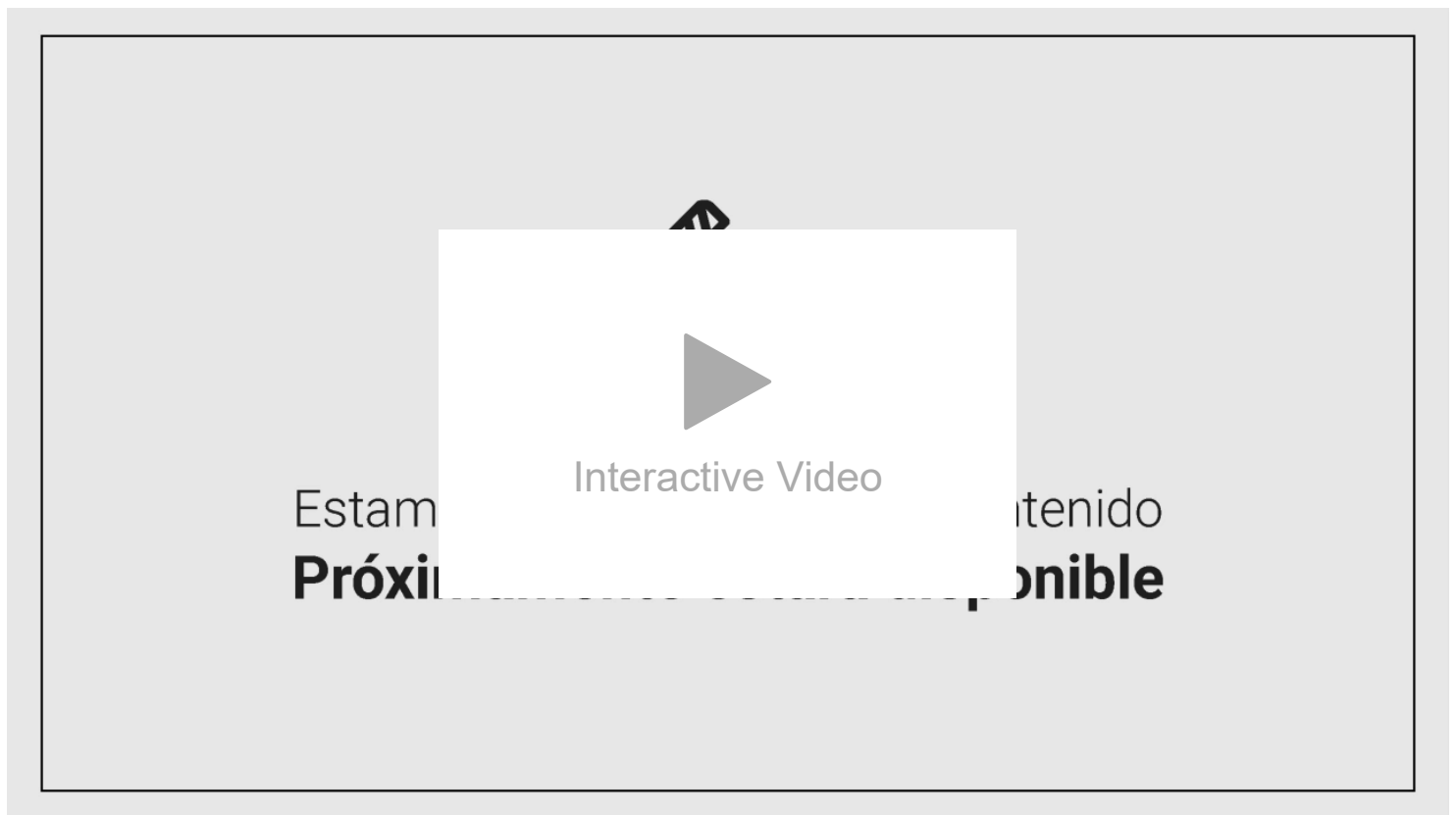
## Referencias

**IBM.** (2020). Lenguaje de consulta estructurada (SQL). Recuperado de <https://www.ibm.com/support/knowledgecenter/es/SS6NHC/com.ibm.swg.im.dashdb.sql.ref.doc/doc/c0004100.html>

**Wikipedia.** (s.f.). Álgebra relacional. Recuperado de [https://es.wikipedia.org/wiki/%C3%81lgebra\\_relacional](https://es.wikipedia.org/wiki/%C3%81lgebra_relacional)

**Wikipedia.** (s.f.). Base de datos relacional. Recuperado de [https://es.wikipedia.org/wiki/Base\\_de\\_datos\\_relacional](https://es.wikipedia.org/wiki/Base_de_datos_relacional)

## Video Conceptual



## Revisión del módulo

Hasta acá aprendimos

≡ [Motores de base de datos relacionales](#)

En esta lectura conocimos definiciones de una base de datos relacional, junto con las características de varios motores de bases de datos (Oracle, MySQL y SQLite) con las instrucciones para descargar el *software* de los mismos, con algunas instrucciones básicas pero suficientes para ponerse en marcha con alguna de ellas. En lecturas posteriores adoptamos MySQL para realizar las prácticas.

≡ [Componentes del motor de base de datos MySQL](#)

Se define a MySQL como la base para realizar las prácticas y se adopta MySQL Workbench para armar un modelo del esquema Sakila que viene con la instalación del motor para profundizar el conocimiento de un motor comercial muy difundido y la herramienta para ejecutar consultas básicas. Esta lectura permitirá poner en marcha todo lo necesario para trabajar en el resto de la materia.

≡ [Componentes del modelo relacional en MySQL](#)

Una vez operativa la base de datos y una herramienta para interactuar con ella, revisamos todos los componentes del modelo relacional en la misma. Se utiliza un diagrama del esquema Sakila, aislando seis tablas, para entender mejor las definiciones de relación, atributos y relaciones con tablas, columnas y claves foráneas. Avanzamos con el concepto de claves primarias.

≡ [Álgebra Relacional, definición y operaciones](#)

Desde la definición de Álgebra Relacional, describimos las operaciones simples y complejas, buscando su correlato en el lenguaje SQL, para ver cómo se implementan en la base de datos elegida y abordando las primeras consultas que aplican las operaciones: proyección, selección, unión, producto cartesiano y combinación.

