



Universidad del  
**Rosario**

Educación Continua  
y Consultoría

# Diplomado en Ciencia de Datos



#URSolucionesInnovadoras

#URConsultoría



Universidad del  
**Rosario**

Educación Continua  
y Consultoría

**BASES DE DATOS 2**

# **Consultas en SQL**



Universidad del  
**Rosario**

Educación Continua  
y Consultoría

## Preliminares

# El entorno de trabajo

## CREDENCIALES DE CONEXIÓN

**Server:** diplomadodatascience.ckg8nba7iwez.us-west-2.rds.amazonaws.com

**Port:** 5432

**User:** postgres

**Pass:** MEeLaN2z





Universidad del  
**Rosario**

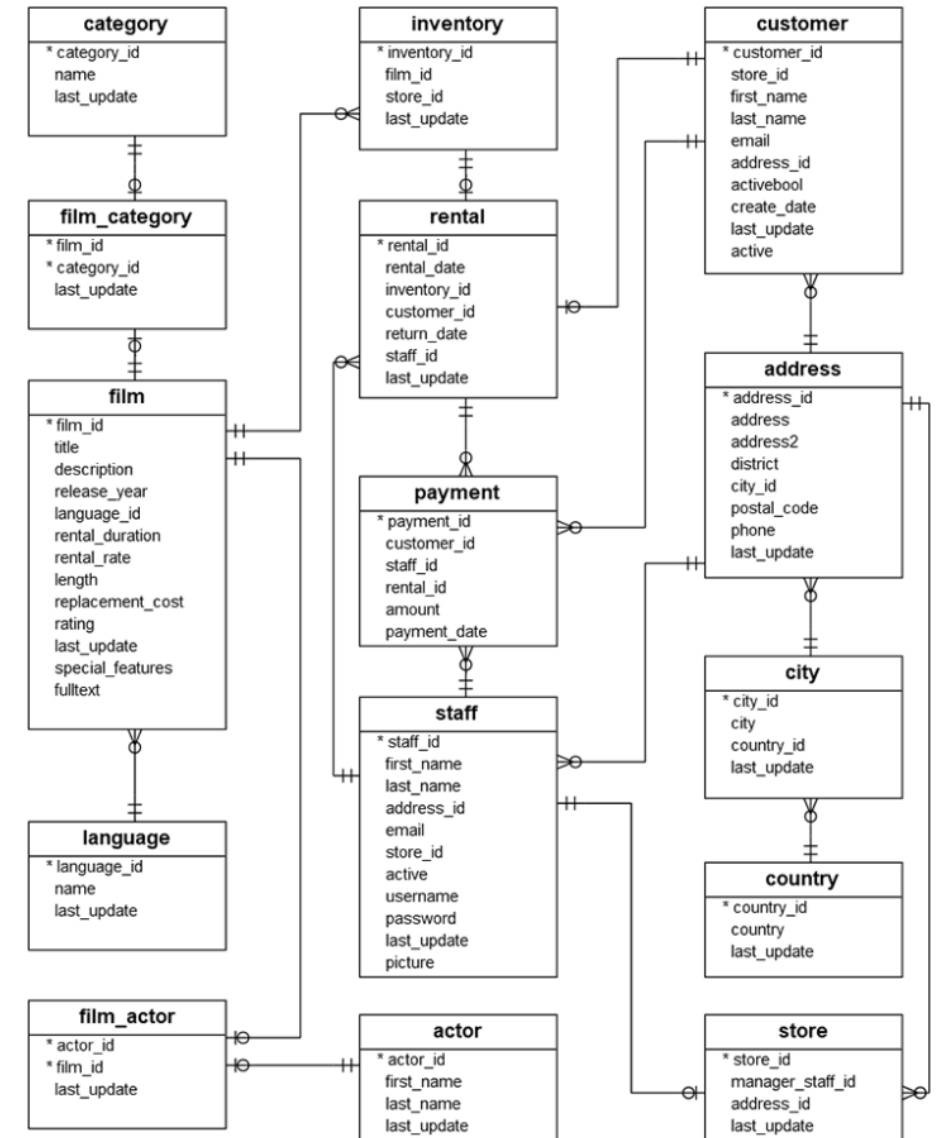
Educación Continua  
y Consultoría

## Preliminares

# Sobre el modelo a consultar

<https://www.postgresqltutorial.com/postgresql-getting-started/postgresql-sample-database/>

DVD Rental ER Model





---

# Contenido

- Cláusulas SQL
- Recuperar datos con SELECT
- Recuperar las filas con condiciones WHERE
- Ordenamiento de resultados usando ORDER BY
- Funciones de agregación de SQL
- Agrupamiento de datos utilizando GROUP BY
- Subconsultas
- Operaciones en tablas múltiples
- realizar operaciones de ajuste (UNION, INTERSECT, EXCEPT).





---

# Contenido

- **Cláusulas SQL**
- Recuperar datos con SELECT
- Recuperar las filas con condiciones WHERE
- Ordenamiento de resultados usando ORDER BY
- Funciones de agregación de SQL
- Agrupamiento de datos utilizando GROUP BY
- Subconsultas
- Operaciones en tablas múltiples
- Operaciones de ajuste (UNION, INTERSECT, EXCEPT).





## Recuperar datos con SELECT

# Cláusulas SQL

Cláusula SQL	Función	Obligatorio
SELECT	Muestra una lista de los campos que contienen datos de interés.	Sí
FROM	Muestra las tablas que contienen los campos de la cláusula SELECT.	Sí
WHERE	Especifica los criterios de campo que cada registro debe cumplir para poder ser incluido en los resultados.	No
ORDER BY	Especifica la forma de ordenar los resultados.	No
GROUP BY	En una instrucción SQL que contiene funciones de agregado, muestra los campos que no se resumen en la cláusula SELECT.	Solo si están estos campos
HAVING	En una instrucción SQL que contiene funciones de agregado, especifica las condiciones que se aplican a los campos que se resumen en la instrucción SELECT.	No







Universidad del  
**Rosario**



Educación Continua  
y Consultoría

# Hojas de trucos

**Sources**

- W3Schools.com
- DataQuest.io

## SQL CHEATSHEET



@AbzAaron

### Commands / Clauses

**SELECT** Select data from database  
**FROM** Specify table we're pulling from  
**WHERE** Filter query to match a condition  
**AS** Rename column or table with alias  
**JOIN** Combine rows from 2 or more tables  
**AND** Combine query conditions. All must be met  
**OR** Combine query conditions. One must be met  
**LIMIT** Limit rows returned. See also **FETCH & TOP**  
**IN** Specify multiple values when using **WHERE**  
**CASE** Return value on a specified condition  
**IS NULL** Return only rows with a NULL value  
**LIKE** Search for patterns in column  
**COMMIT** Write transaction to database  
**ROLLBACK** Undo a transaction block

**ALTER TABLE** Add/Remove columns from table  
**UPDATE** Update table data  
**CREATE** Create TABLE, DATABASE, INDEX or VIEW  
**DELETE** Delete rows from table  
**INSERT** Add single row to table  
**DROP** Delete TABLE, DATABASE, or INDEX

**GROUP BY** Group data into logical sets  
**ORDER BY** Set order of result. Use **DESC** to reverse order  
**HAVING** Same as **WHERE** but filters groups  
**COUNT** Count number of rows  
**SUM** Return sum of column  
**AVG** Return average of column  
**MIN** Return min value of column  
**MAX** Return max value of column

### Joins

**a INNER JOIN b**

**a LEFT JOIN b**

**a RIGHT JOIN b**

**a FULL OUTER JOIN b**

### Examples

Select all columns with filter applied

```
SELECT * FROM tbl
WHERE col > 5;
```

Select first 10 rows for two columns

```
SELECT col1, col2
FROM tbl LIMIT 10;
```

Select all columns with multiple filters

```
SELECT * FROM tbl
WHERE col1 > 5 OR col2 < 2;
```

Select all rows from col1 & col2 ordering by col1

```
SELECT col1, col2
FROM tbl ORDER BY 1;
```

Return count of rows in table

```
SELECT COUNT(*)
FROM tbl;
```

Return sum of col1

```
SELECT SUM(col1)
FROM tbl;
```

Return max value for col1

```
SELECT MAX(col1)
FROM tbl;
```

Compute summary stats by grouping col2

```
SELECT AVG(col1) FROM tbl
GROUP BY col2;
```

Combine data from 2 tables using left join

```
SELECT * FROM tbl1 AS t1 LEFT JOIN
tbl2 AS t2 ON t2.col1 = t1.col1;
```

Aggregate and filter result

```
SELECT col1,
COUNT(*) AS total
FROM tbl
GROUP BY col1
HAVING COUNT(*) > 10;
```

Implementation of CASE statement

```
SELECT col1,
CASE
  WHEN col1 > 10 THEN 'more than 10'
  WHEN col1 < 10 THEN 'less than 10'
  ELSE '10'
END AS NewColumnName
FROM tbl;
```

### Data Definition Language

CREATE	ALTER
CREATE DATABASE MyDatabase;	ALTER TABLE MyTable DROP COLUMN col5;
CREATE TABLE MyTable ( id int, name varchar(10));	ALTER TABLE MyTable ADD col5 int;
CREATE INDEX IndexName ON TableName(col1);	<b>DROP</b> DROP DATABASE MyDatabase; DROP TABLE MyTable;

### Data Manipulation Language

UPDATE	INSERT
UPDATE MyTable SET col1 = 50 WHERE col2 = 'something';	INSERT INTO MyTable (col1, col2) VALUES ('value1', 'value2');
<b>DELETE</b> DELETE FROM MyTable WHERE col1 = 'something';	<b>SELECT</b> SELECT col1, col2 FROM MyTable;

### Order Of Execution

- 1 FROM
- 2 WHERE
- 3 GROUP BY
- 4 HAVING
- 5 SELECT
- 6 ORDER BY
- 7 LIMIT



<https://www.instagram.com/p/CdFxG9MLai4/>





---

# Contenido

- Cláusulas SQL
- **Recuperar datos con SELECT**
- Recuperar las filas con condiciones WHERE
- Ordenamiento de resultados usando ORDER BY
- Funciones de agregación de SQL
- Agrupamiento de datos utilizando GROUP BY
- Subconsultas
- Operaciones en tablas múltiples
- Operaciones de ajuste (UNION, INTERSECT, EXCEPT).





## Recuperar datos con SELECT

# Recuperar una tabla

```
SELECT
    category_id,
    name,
    last_update
FROM
    category;
```

Comodín \*



```
SELECT
    *
FROM
    category;
```





Universidad del  
**Rosario**

Educación Continua  
y Consultoría

---

Recuperar datos con SELECT

# Comentarios

```
-- asdasdasda  
  
/* asdasdasda  
   asdasdasda  
*/
```





Recuperar datos con **SELECT**

# Filtrar columnas



Se dejan únicamente los nombres de las columnas deseadas

```
SELECT
    name,
    category_id,
FROM
    category;
```





## Recuperar datos con SELECT

# Eliminar tuplas duplicadas

- Se usa la palabra DISTINCT

```
SELECT  
    first_name  
FROM  
    actor;
```

Entrega todos los  
resultados, así tengan  
tuplas repetidas

```
SELECT  
    DISTINCT first_name  
FROM  
    actor;
```

Elimina las tuplas repetidas





Recuperar datos con SELECT

# Columnas calculadas/derivadas

- Operaciones matemáticas como resultado en columna
- La columna toma el nombre de la operación (por defecto)
- Se puede poner un *alias* al nombre de la columna



```
SELECT  
    title,  
    replacement_cost + rental_rate  
FROM film;
```

```
SELECT  
    title,  
    replacement_cost + rental_rate AS price  
FROM film;
```



## Recuperar datos con SELECT

# Operadores matemáticos

Operator	Description	Example	Result
+	addition	2 + 3	5
-	subtraction	2 - 3	-1
*	multiplication	2 * 3	6
/	division (integer division truncates the result)	4 / 2	2
%	modulo (remainder)	5 % 4	1
^	exponentiation (associates left to right)	2.0 ^ 3.0	8
/	square root	/ 25.0	5
/	cube root	/ 27.0	3

<https://www.postgresql.org/docs/9.5/functions-math.html>





Recuperar datos con **SELECT**

# Funciones matemáticas



Function	Return Type	Description	Example	Result
<code>abs(x)</code>	(same as input)	absolute value	<code>abs(-17.4)</code>	17.4
<code>cbrt(dp)</code>	dp	cube root	<code>cbrt(27.0)</code>	3
<code>ceil(dp or numeric)</code>	(same as input)	nearest integer greater than or equal to argument	<code>ceil(-42.8)</code>	-42
<code>ceiling(dp or numeric)</code>	(same as input)	nearest integer greater than or equal to argument (same as ceil)	<code>ceiling(-95.3)</code>	-95
<code>degrees(dp)</code>	dp	radians to degrees	<code>degrees(0.5)</code>	28.6478897565412

<https://www.postgresql.org/docs/9.5/functions-math.html>



---

# Contenido

- Cláusulas SQL
- Recuperar datos con SELECT
- **Recuperar las filas con condiciones WHERE**
- Ordenamiento de resultados usando ORDER BY
- Funciones de agregación de SQL
- Agrupamiento de datos utilizando GROUP BY
- Subconsultas
- Operaciones en tablas múltiples
- Operaciones de ajuste (UNION, INTERSECT, EXCEPT).





## Recuperar datos con SELECT

# La cláusula WHERE

```
SELECT *  
FROM film  
WHERE length >184;
```

Condiciones/Predicados básicos:

- **Comparación:** Compara el valor de una expresión con el valor de otra expresión
- Prueba de **rango:** si el valor de una expresión cae dentro de un determinado rango de valores
- Establecer **pertenencia:** Probar si el valor de una expresión es igual a uno de un conjunto de valores
- Coincidencia de **patrón:** Comprueba si una cadena coincide con un patrón específico.
- **Null:** Comprueba si una columna tiene un valor nulo (desconocido).





Universidad del  
**Rosario**

Educación Continua  
y Consultoría

Recuperar datos con SELECT

# La cláusula WHERE



```
SELECT *  
FROM film  
WHERE length >184;
```





## Recuperar datos con SELECT

# Condiciones básicas

- **Comparación:** Compara el valor de una expresión con el valor de otra expresión
- Prueba de **rango:** si el valor de una expresión cae dentro de un determinado rango de valores
- Establecer **pertenencia:** Probar si el valor de una expresión es igual a uno de un conjunto de valores
- Coincidencia de **patrón:** Comprueba si una cadena coincide con un patrón específico.
- **Null:** Comprueba si una columna tiene un valor nulo (desconocido).



Recuperar datos con SELECT

# Operadores de comparación

COMPARISON OPERATORS	
SYMBOL	MEANING
=	Equal to
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
<> or !=	Not equal to

**QUERY:** ¿Qué películas duran menos de una hora?



## Recuperar datos con SELECT

# Predicados más complejos

Se pueden generar predicados más complejos utilizando los operadores lógicos **AND**, **OR** y **NOT**, con paréntesis (si es necesario o se desea)

Reglas para evaluar una expresión condicional

- Una expresión se evalúa de izquierda a derecha
- Las subexpresiones entre paréntesis se evalúan primero
- Los NOT se evalúan antes de los AND y los OR
- Los AND se evalúan antes que los OR

Se recomienda el uso de paréntesis, para eliminar cualquier posible ambigüedad.

**QUERY:** ¿Qué películas duran entre una y dos horas?



```
SELECT *  
FROM film  
WHERE  
    length > 184 AND  
    release_year = 2006 AND  
    rating <> 'R';
```





Universidad del  
**Rosario**

Educación Continua  
y Consultoría

Recuperar datos con SELECT

# Rangos con **BETWEEN/NOT BETWEEN**

Incluye los valores extremos

```
SELECT *  
FROM film  
WHERE length BETWEEN 60 AND 120;
```





Recuperar datos con SELECT

# Comparación de un valor en una lista de valores con **IN/NOT IN**

```
SELECT *  
FROM film  
WHERE rental_duration IN (4, 7);
```

QUERY: Liste todas las películas cuyo rating es R, G o PG-13





Recuperar datos con **SELECT**

# Patrones de coincidencia

SQL tiene dos símbolos especiales para buscar por coincidencia de patrón:

- **%** Representa una secuencia de cero o más caracteres (comodin)
- **\_** Representa un solo character
- Se escapa con **#**



## Recuperar datos con SELECT

# Patrones de coincidencia

## Ejemplo

Que comiencen por Flor

```
SELECT Name
FROM city
WHERE CountryCode = 'COL' AND
      Name LIKE 'Flor%';
```

Que comiencen por Flor y tenga cinco caracteres después

```
SELECT Name
FROM city
WHERE CountryCode = 'COL' AND
      Name LIKE 'Flor_____';
```

Que terminen en ra

```
SELECT Name
FROM city
WHERE CountryCode = 'COL' AND
      Name LIKE '%ra';
```

Que tenga cinco caracteres después y termine en ra

```
SELECT Name
FROM city
WHERE CountryCode = 'COL' AND
      Name LIKE '_____ra';
```



Recuperar datos con **SELECT**

# Patrones de coincidencia



Algunas preguntas

- ¿Cuántas películas inician con la palabra “**Heaven**”?
- ¿Cuántos nombres diferentes de actores o actrices comienzan con las letras “**Ca**”?
- ¿Qué ciudades tienen en su nombre las letras “**de**” o “**del**”



---

Recuperar datos con SELECT

# Valores nulos

Un valor nulo (**NULL**) es un valor desconocido, por lo que no se puede comparar con un string con = o <>

Para comparar con nulo se usan las palabras clave

**IS NULL/IS NOT NULL**





---

# Contenido

- Cláusulas SQL
- Recuperar datos con SELECT
- Recuperar las filas con condiciones WHERE
- **Ordenamiento de resultados usando ORDER BY**
- Funciones de agregación de SQL
- Agrupamiento de datos utilizando GROUP BY
- Subconsultas
- Operaciones en tablas múltiples
- Operaciones de ajuste (UNION, INTERSECT, EXCEPT).







Recuperar datos con **SELECT**

# Ordenamiento de resultados

**ORDER BY** puede ordenar el resultado de la consulta en orden ascendente (**ASC**) o descendente (**DESC**)



```
SELECT *  
FROM country  
ORDER BY Surface_area DESC;
```

```
SELECT *  
FROM country  
ORDER BY  
continent ASC,  
region ASC,  
Surface_area DESC;
```



---

# Contenido

- Cláusulas SQL
- Recuperar datos con SELECT
- Recuperar las filas con condiciones WHERE
- Ordenamiento de resultados usando ORDER BY
- **Funciones de agregación de SQL**
- Agrupamiento de datos utilizando GROUP BY
- Subconsultas
- Operaciones en tablas múltiples
- Operaciones de ajuste (UNION, INTERSECT, EXCEPT).





---

Recuperar datos con **SELECT**

# Agregación/Resumen



El estándar ISO define cinco funciones

- **COUNT**: devuelve el número de valores en una columna especificada
- **SUM**: devuelve la suma de los valores en una columna específica
- **AVG**: devuelve el promedio de los valores en una columna específica
- **MIN**: devuelve el valor más pequeño en una columna específica
- **MAX**: devuelve el mayor valor en una columna específica



Recuperar datos con **SELECT**

# Agregación/Resumen

## Características

- Operan en una sola columna de la tabla
- Retornan un solo valor
- **COUNT**, **MIN**, **MAX** funciona con campos numéricos y no numéricos
- Las funciones eliminan los nulos antes de operar (menos **COUNT(\*)**)
- **COUNT(\*)** es especial, cuenta todas las columnas de la tabla
- Se pueden eliminar duplicados antes de contar usando **DISTINCT** ( no funciona para **MIN** ni **MAX**)
- Se pueden usar solo en la lista del **SELECT** o en la cláusula **HAVING**





Recuperar datos con SELECT

# Agregación/Resumen Ejemplo



¿Cuántas películas hay de rating PG-13?

```
SELECT COUNT(*)  
FROM film  
WHERE rating = 'PG-13';
```

¿Cuál es la duración media de un film?

¿Cuáles films duran más que el promedio de duración de los films?

¿Cuántos films duran más que el promedio de duración de los films?



Recuperar datos con SELECT

# Conteos con DISTINCT

Cantidad de nombres diferentes

```
SELECT COUNT(DISTINCT first_name)
FROM actor;
```

Algunos descriptivos de los films

```
SELECT
COUNT(film_id) AS num_films,
SUM(length) AS total_lenght
FROM film
WHERE rating = 'R';
```





---

# Contenido

- Cláusulas SQL
- Recuperar datos con SELECT
- Recuperar las filas con condiciones WHERE
- Ordenamiento de resultados usando ORDER BY
- Funciones de agregación de SQL
- **Agrupamiento de datos utilizando GROUP BY**
- Subconsultas
- Operaciones en tablas múltiples
- Operaciones de ajuste (UNION, INTERSECT, EXCEPT).







## Recuperar datos con SELECT

# Agrupaciones en resúmenes

Los resultados anteriores son filas de totales

Podemos hacer subtotales, agrupar (**GROUP BY**) por alguna de las columnas seleccionadas en el **SELECT**

### Columnas de agrupación

Toda columna del **SELECT** debe estar acá, a menos que sea agregación

Tienen que tener valor único por grupo  
Para que funcionen, en el SELECT solo pueden haber nombres de columnas, agregaciones, o constantes

```
SELECT  
rating,  
COUNT(film_id) AS num_films,  
SUM(length) AS total_lenght  
FROM film  
GROUP BY rating;
```



Recuperar datos con SELECT

# Restringir aparición de grupos con HAVING

Se filtra por los resultados de las agregaciones

```
SELECT
rating,
COUNT(film_id) AS num_films,
SUM(length) AS total_lenght
FROM film
GROUP BY rating
HAVING COUNT(film_id) > 200;
```





---

# Contenido

- Cláusulas SQL
- Recuperar datos con SELECT
- Recuperar las filas con condiciones WHERE
- Ordenamiento de resultados usando ORDER BY
- Funciones de agregación de SQL
- Agrupamiento de datos utilizando GROUP BY
- **Subconsultas**
- Operaciones en tablas múltiples
- Operaciones de ajuste (UNION, INTERSECT, EXCEPT).





Recuperar datos con SELECT

# Subconsultas

Es una sentencia **SELECT** embebida dentro de otra sentencia **SELECT**. El resultado de la consulta **interna** determina el contenido del resultado final de la consulta

```
SELECT * FROM category;
```

```
SELECT category_id FROM category WHERE name = 'Music'
```

```
SELECT * FROM film_category  
WHERE category_id = (SELECT category_id FROM category WHERE name = 'Music')
```



Recuperar datos con SELECT

# Indentación



```
SELECT * FROM film_category  
WHERE category_id = (SELECT category_id FROM category WHERE name = 'Music')
```

VS.

```
SELECT *  
FROM film_category  
WHERE category_id = (  
    SELECT category_id  
    FROM category  
    WHERE name = 'Music')
```



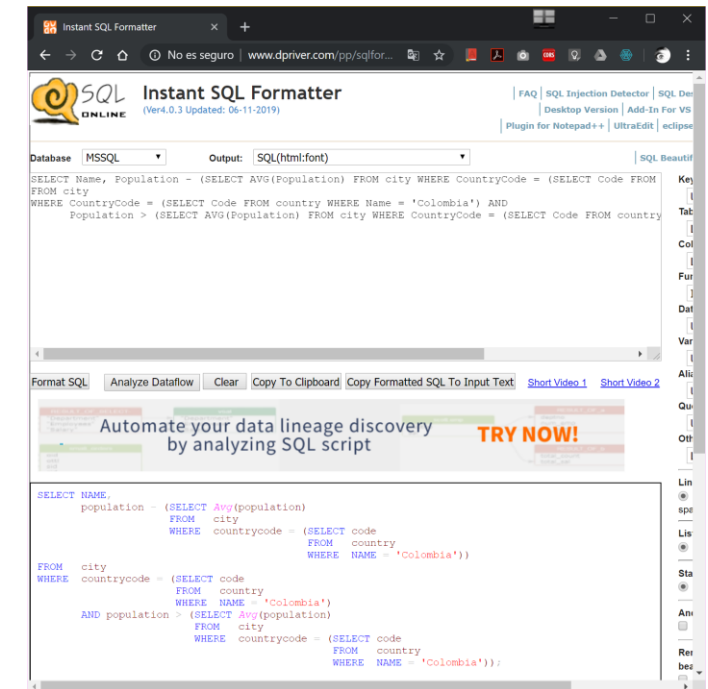
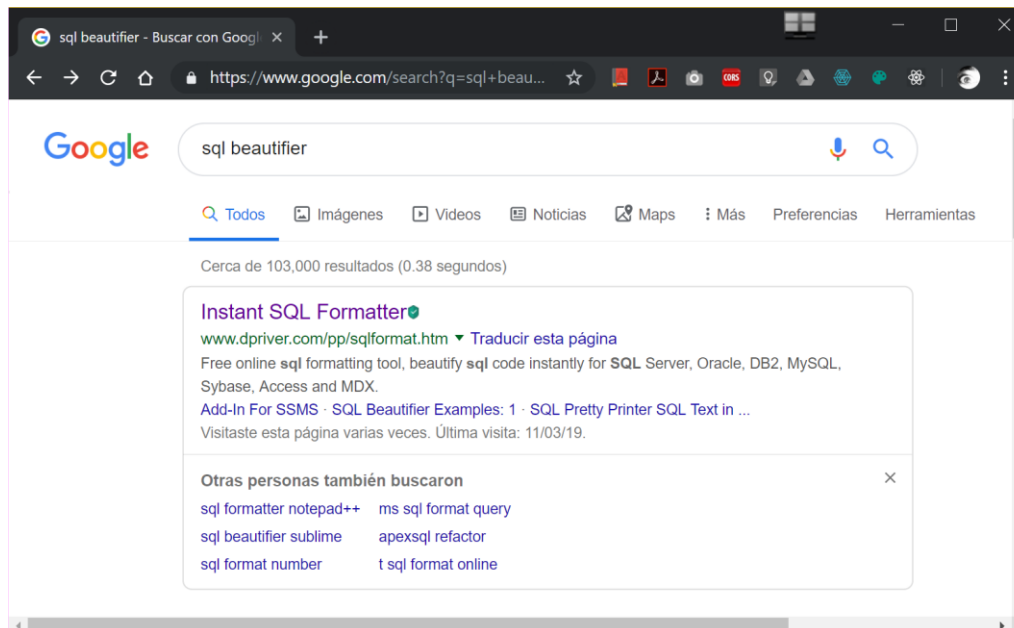
Universidad del  
**Rosario**

Educación Continua  
y Consultoría

Recuperar datos con SELECT

# Indentación

Herramientas code formatter o code beautifier





## Recuperar datos con SELECT

# Tipos de subconsultas

- **Escalares:** Devuelven una columna con un solo valor

```
SELECT SUM(length)  
FROM film;
```

- **De fila:** Devuelve muchas columnas pero una sola fila

```
SELECT DISTINCT rating  
FROM film;
```

- **De tabla:** Múltiples filas y columnas en la salida

```
SELECT title, length FROM film  
WHERE title LIKE '%Cat%';
```







Recuperar datos con SELECT

## Uso de **ANY/SOME**

Por lo menos uno que cumpla la condición en el grupo

```
SELECT * FROM film
WHERE length > SOME (
    SELECT length FROM film WHERE rating = 'R'
);
```





Recuperar datos con SELECT

## Uso de **ALL**

Todos deben cumplir la condición en el grupo

```
SELECT * FROM film
WHERE length >= ALL (
    SELECT length FROM film WHERE rating = 'R'
);
```





---

# Contenido

- Cláusulas SQL
- Recuperar datos con SELECT
- Recuperar las filas con condiciones WHERE
- Ordenamiento de resultados usando ORDER BY
- Funciones de agregación de SQL
- Agrupamiento de datos utilizando GROUP BY
- Subconsultas
- **Operaciones en tablas múltiples**
- Operaciones de ajuste (UNION, INTERSECT, EXCEPT).

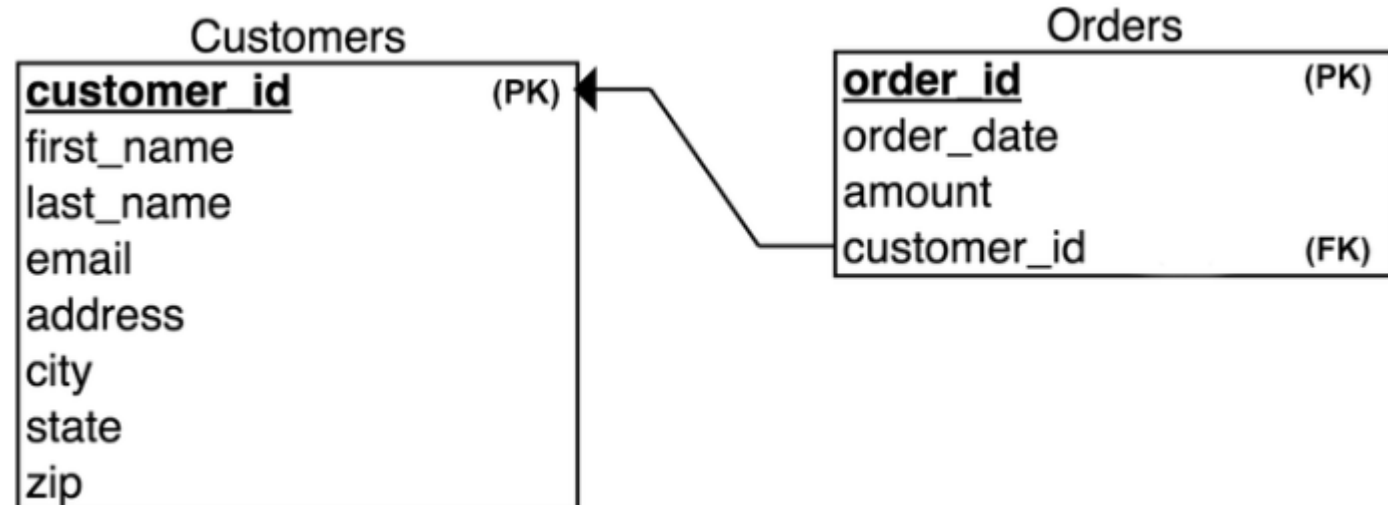
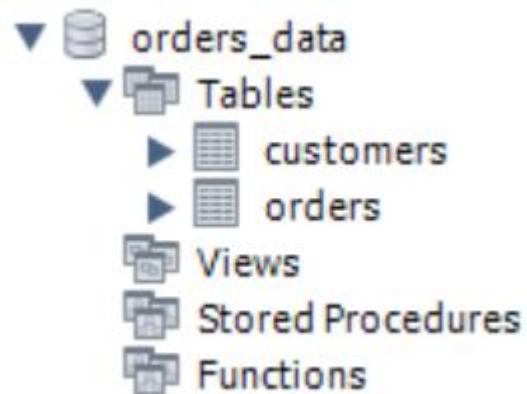




Recuperar datos con **SELECT**

# Consultas en varias tablas

**JOIN:** Instrucción que combina datos de dos conjuntos de datos en una sola tabla





## Recuperar datos con SELECT

# Tablas de ejemplo

x		y	
1	x1	1	y1
2	x2	2	y2
3	x3	4	y3

```
CREATE TABLE tabla1 (  
    row_id INT PRIMARY KEY,  
    row_value VARCHAR(50)  
);
```

```
INSERT INTO tabla1 (row_id,row_value)  
VALUES (1,'x1'),(2,'x2'),(3,'x3');
```

```
CREATE TABLE tabla2 (  
    row_id INT PRIMARY KEY,  
    row_value VARCHAR(50)  
);
```

```
INSERT INTO tabla2 (row_id,row_value)  
VALUES (1,'y1'),(2,'y2'),(4,'y3');
```

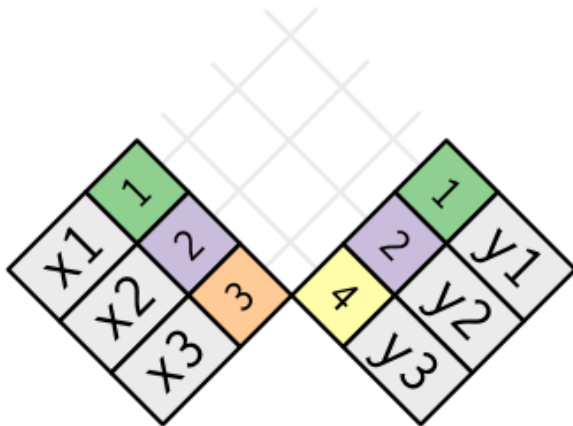




Recuperar datos con SELECT

# Consultas en varias tablas

Producto cartesiano: Combina todas las filas de una tabla con las filas de otra tabla (arma todos los posibles pares)



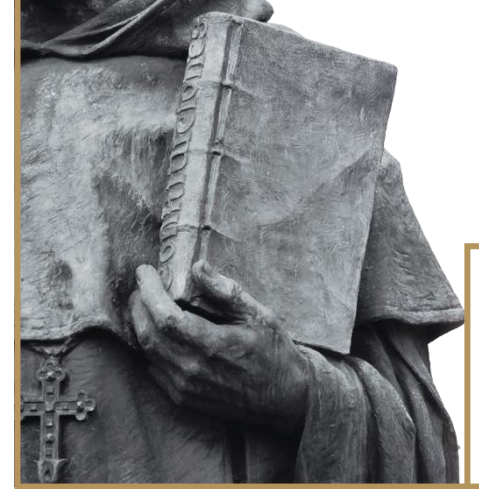
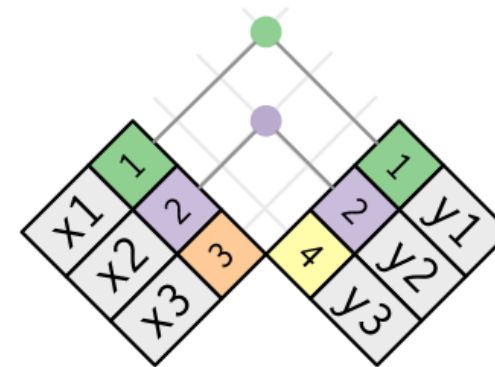
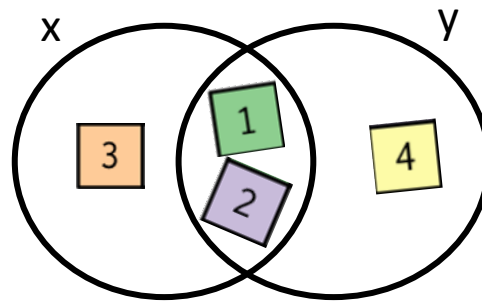
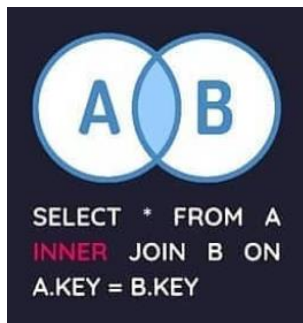
```
SELECT x.*, y.*  
FROM tabla1 x, tabla2 y;
```



Recuperar datos con SELECT

# Consultas en varias tablas

**JOIN** simple: Combina todas las filas de una tabla con las filas de otra tabla, y deja los resultados donde las llaves de ambas tablas corresponden



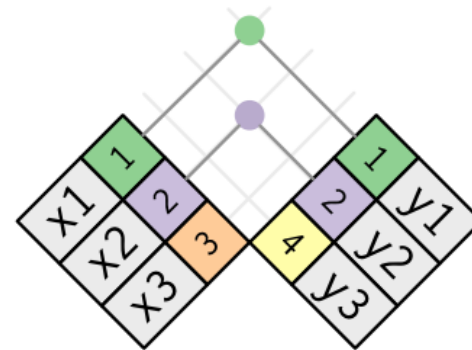




## Recuperar datos con SELECT

# Inner JOIN o JOIN simple

x		y	
1	x1	1	y1
2	x2	2	y2
3	x3	4	y3



key	val_x	val_y
1	x1	y1
2	x2	y2

```
SELECT x.*, y.*  
FROM tabla1 x, tabla2 y  
WHERE x.row_id = y.row_id;
```

=

```
SELECT x.*, y.*  
FROM tabla1 x JOIN tabla2 y  
ON x.row_id = y.row_id;
```

QUERY: ¿Cuántos resultados tendrá la union simple de **country** y **city**?





## Recuperar datos con SELECT

# OUTER Join

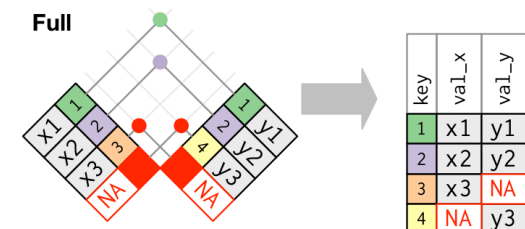
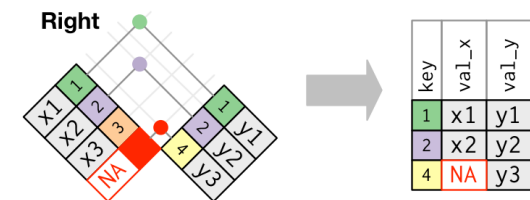
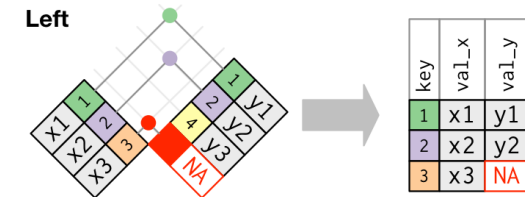
Es como un  
JOIN simple,  
pero retiene las  
filas que no  
satisfacen la  
condición.

Tiene tres tipos

```
SELECT x.*, y.*  
FROM tabla1 x  
LEFT OUTER JOIN tabla2 y  
ON x.row_id = y.row_id;
```

```
SELECT x.*, y.*  
FROM tabla1 x  
RIGHT OUTER JOIN tabla2 y  
ON x.row_id = y.row_id;
```

```
SELECT x.*, y.*  
FROM tabla1 x  
FULL OUTER JOIN tabla2 y  
ON x.row_id = y.row_id;
```





---

# Contenido

- Cláusulas SQL
- Recuperar datos con SELECT
- Recuperar las filas con condiciones WHERE
- Ordenamiento de resultados usando ORDER BY
- Funciones de agregación de SQL
- Agrupamiento de datos utilizando GROUP BY
- Subconsultas
- Operaciones en tablas múltiples
- Operaciones de ajuste (**UNION, INTERSECT, EXCEPT**).





Recuperar datos con SELECT

# Combinación de tablas

- UNION
- INTERSECT
- EXCEPT

Las tablas a unir deben tener el mismo número de columnas



```
SELECT length FROM film  
WHERE rating = 'R'  
UNION  
SELECT length FROM film  
WHERE rating = 'PG-13';
```

```
SELECT length FROM film  
WHERE rating = 'R'  
UNION ALL  
SELECT length FROM film  
WHERE rating = 'PG-13';
```

UNION ALL preserva tuplas duplicadas



Recuperar datos con SELECT

# Subconsultas con EXISTS

Las palabras **EXISTS** y **NOT EXISTS** están diseñadas solo para subconsultas

Retornan **TRUE** si la subconsulta tiene resultados o **FALSE** si es vacía

```
SELECT * FROM language lan
WHERE EXISTS(
    SELECT *
    FROM film
    WHERE language_id = lan.language_id
);
```

Unidad de Educación Continua y Consultoría  
construimos país desde

**#URSolucionesInnovadoras**  
**#URConsultoría**



@RosarioContinua



/EduContinuaURosario



@RosarioContinua