




CONSULTAS AVANZADAS CON POSTGREE

ING. MOISÉS ÁLVAREZ HUAMÁN

SEMANA 8



CONSULTAS COMPLEJAS

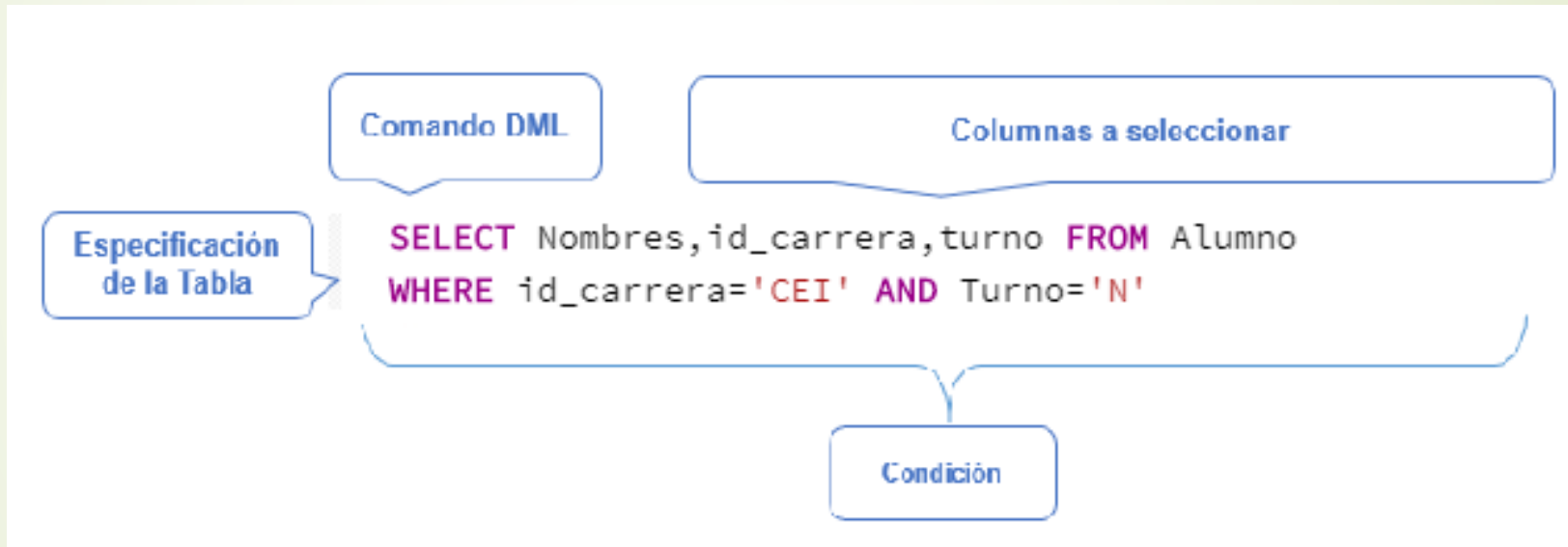
- “El álgebra relacional es un tipo de álgebra con una serie de operadores que trabajan sobre una o varias relaciones para obtener una relación resultado” (PostgreSQL).
 - Las operaciones más importantes disponibles en álgebra relacional son:
 - A continuación, se detalla el empleo de dichos operadores:
- 

CONSULTA DE SELECCIÓN

- ▶ Permiten visualizar determinados registros bajo condiciones que se establecen en la sentencia, extrayendo los resultados necesarios a consultar .
- ▶ EJEMPLO: Mostrar los alumnos que son de computación y del turno noche que tienen un préstamo.

	nombres character varying (40)	id_carrera character (3)	turno character (1)
1	KAREN	CEI	N
2	ROSA	CEI	N

Al ejecutar dicha sentencia, se podrá visualizar únicamente los datos que cumplen con la condición establecida que corresponde a los propietarios tipo empresa:



PROYECCIÓN

- “Esta operación aplicada a una relación que produce una nueva con solamente los atributos (columnas) especificados” (PostgreSQL).
- Para ver cómo funciona este concepto se realiza el siguiente ejercicio práctico. Primero visualizaremos la tabla de Alumno, obteniendo los siguientes resultados:
- `SELECT * FROM Alumno`

	id_alumno character (8)	id_carrera character (3)	apellidos character varying (40)	nombres character varying (40)	sexo character (1)	direccion character varying (40)	semestre character (3)	turno character (1)
1	20694075	CON	ALVAREZ CASAS	MIGUEKATTY JENIFER	F	JR. AREQUIPA 234	I	D
2	20694077	FAR	RAYGAL ORTIZ	CLAUDIA	F	JR. LEONCIO PRADO 230	II	D
3	20694078	AGR	CASAS CARHUANCHO	JULIO	M	JR. HUASCARIPA 2321	VI	D
4	20694079	MEC	RAFAEL SINCHE	JORGE LUIS	M	JR. HUANUCO 231	IV	D



Comando DML

Comando para especificar la Tabla

```
SELECT Id_alumno, Nombres, Apellidos, Id_carrera FROM Alumno
```

Especificación de las columnas a visualizar

Nombre de la Tabla

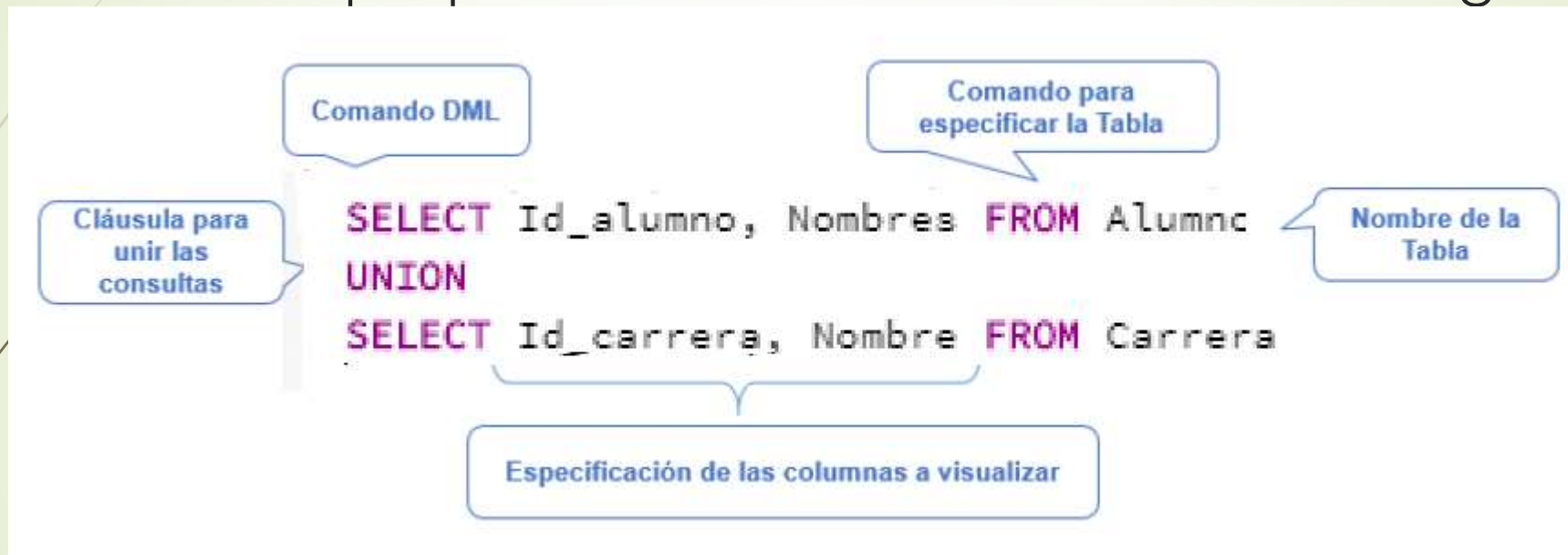
	id_alumno character (8)	nombres character varying (40)	apellidos character varying (40)	id_carrera character (3)
1	20694075	MIGUEKATTY JENIFER	ALVAREZ CASAS	CON
2	20694077	CLAUDIA	RAYGAL ORTIZ	FAR
3	20694078	JULIO	CASAS CARHUANCHO	AGR
4	20694079	JORGE LUIS	RAFAEL SINCHE	MEC
5	20694080	MAE	VERASTEGUI CASIMIRO	IND
6	20694081	KAREN	BALDEON MEZA	CEI



UNIÓN

- ▶ “La unión de R y S es el conjunto de elementos que existen en R, ó en S, ó en las dos. Un elemento que existe tanto en R como en S aparece solamente una vez en la unión” (PostgreSQL).
- ▶ Mediante esta operación se puede realizar consultas de diversas tablas, presentando un solo conjunto de resultados que existen en las tablas establecidas. Hay que recordar que cada consulta con la cláusula UNION debe tener el mismo número de columnas y ser del mismo tipo de dato de las columnas establecidas secuencialmente.

Para visualizar el código y los nombres de la tabla Alumno y Carrera se realiza la siguiente sentencia, en donde especificamos las columnas a seleccionar en la consulta y la tabla a la que pertenece en donde se detalla lo siguiente:



	id_alumno	nombres
	character	character varying (40)
1	20694079	JORGE LUIS
2	20694082	CAROLINA
3	20694081	KAREN
4	ENF	ENFERMERIA
5	MEC	MECANICA DE PRODUCCI...



INTERSECCIÓN

- Esta operación une distintas consultas con la misma cantidad de parámetros en la Selección y retorna solo los registros duplicados. “La intersección de R y S es el conjunto de elementos que existen en R y en S.”
(PostgreSQL) La cláusula para esta operación en SQL es INTERSECT.
- Para visualizar el código de las carreras que tienen alumnos registrados en la tabla Alumno, emplearemos la intersección.

Esta operación une distintas consultas con la misma cantidad de parámetros en la Selección y retorna solo los registros duplicados. “La intersección de R y S es el conjunto de elementos que existen en R y en S.” (PostgreSQL) La cláusula para esta operación en SQL es INTERSECT.

Para visualizar el código de los alumnos de la institución que pertenecen a una carrera, emplearemos la intersección.

Cláusula de inserción

```
SELECT Id_carrera FROM Alumno  
INTERSECT
```

```
SELECT Id_carrera FROM Carrera
```

Consulta 1

Consulta 2

	id_carrera character (3)
1	CON
2	MEC
3	FAR
4	IND
5	CEI



DIFERENCIA

- “La diferencia de R y S es el conjunto de elementos que existen en R pero no en S. $R-S$ es diferente a $S-R$, $S-R$ sería el conjunto de elementos que existen en S pero no en R.” (PostgreSQL)
- Mediante este operador se visualizan los datos de una consulta entre tablas en donde el resultado a presentar es de la primera consulta menos los valores de las siguientes.
- La cláusula en SQL es EXCEPT.

Para indicar el funcionamiento de esta cláusula, reemplazaremos a la consulta anterior la cláusula INTERSECT por EXCEPT, el cual nos mostrara el siguiente resultado:

```
150
151 SELECT Id_carrera FROM Carrera
152 EXCEPT
153 SELECT Id_carrera FROM Alumno|
```

	Data Output	Explain	Messages	Notifications
	id_carrera character (3)			
1	SEC			



COMBINACIÓN

- Esta acción se realiza por medio de la cláusula JOIN la cual “combinar dos o más relaciones según una condición para obtener tuplas compuestas por atributos de las dos relaciones combinadas” (PostgreSQL).

PRODUCTO CARTESIANO

- ▶ Este término hace referencia a las combinaciones cruzadas en la cual “se emplea en lenguaje SQL el término de CROSS JOIN ó separando las relaciones usadas en el producto con comas, en el FROM de la sentencia SQL. para realizar dicha acción” (PostgreSQL).

Para visualizar el funcionamiento de esta cláusula visualizaremos el contenido de dos tablas a emplear.


```
155 SELECT * FROM Alumno
```

```
156
```

[Data Output](#) [Explain](#) [Messages](#) [Notifications](#)

	id_alumno character (8)	id_carrera character (3)	apellidos character varying (40)	nombres character varying (40)	sexo character (1)	direccion character varying (40)
1	20694075	CON	ALVAREZ CASAS	MIGUEKATTY JENIFER	F	JR. AREQUIPA 234
2	20694077	FAR	RAYGAL ORTIZ	CLAUDIA	F	JR. LEONCIO PRADO 230
3	20694078	AGR	CASAS CARHUANCHO	JULIO	M	JR. HUASCARIPA 2321
4	20694079	MEC	RAFAEL SINCHE	JORGE LUIS	M	JR. HUANUCO 231

```
155 SELECT * FROM Carrera
```

```
156
```

[Data Output](#) [Explain](#) [Messages](#) [Notifications](#)

	id_carrera character (3)	nombre character varying (40)
1	AGR	AGROPECUARIA
2	CEI	COMPUTACION E INFRO...
3	CON	CONTABILIDAD
4	ENF	ENFERMERIA
5	FAR	FARMACIA

Con la siguiente sentencia se multiplican los valores de las tablas en donde se visualizar el siguiente resultado:



	id_alumno character (8)	id_carrera character (3)	apellidos character varying (40)	nombres character varying (40)	sexo character (1)	direccion character varying (40)
1	20694075	CON	ALVAREZ CASAS	MIGUEKATTY JENIFER	F	JR. AREQUIPA 234
2	20694077	FAR	RAYGAL ORTIZ	CLAUDIA	F	JR. LEONCIO PRADO 230
3	20694078	AGR	CASAS CARHUANCHO	JULIO	M	JR. HUASCARIPA 2321



COMBINACIONES INTERNAS

- ▶ “La sentencia INNER JOIN es la sentencia JOIN por defecto, y consiste en combinar cada fila de una tabla con cada fila de la otra tabla, seleccionando aquellas filas que cumplan una determinada condición” (SQL INNER JOIN).
- ▶ La cláusula INNER JOIN interviene entre dos relaciones, para lo cual el resultado que se obtiene después de aplicar al producto cartesiano.

Como apreciamos en el ejercicio anterior la tabla de Alumno consta de 10 columnas y la de Carrera consta de 8 columnas, las cuales al agruparlas con la sentencia INNER JOIN se obtendrá 18 columnas con los valores correspondientes a la relación fijada, mediante la siguiente sentencia:

Comando DML

Cláusula para combinación interna

```
SELECT * FROM Alumno  
INNER JOIN Carrera ON Alumno.id_carrera=Carrera.id_carrera
```

Columnas de relación de las tablas

	id_alumno character (8)	id_carrera character (3)	apellidos character varying (40)	nombres character varying (40)	sexo character (1)	direccion character varying (40)
1	20694075	CON	ALVAREZ CASAS	MIGUEKATTY JENIFER	F	JR. AREQUIPA 234
2	20694077	FAR	RAYGAL ORTIZ	CLAUDIA	F	JR. LEONCIO PRADO 230
3	20694078	AGR	CASAS CARHUANCHO	JULIO	M	JR. HUASCARIPA 2321
4	20694079	MEC	RAFAEL SINCHE	JORGE LUIS	M	JR. HUANUCO 231

COMBINACIONES EXTERNAS

- Retorna todas las tuplas de la combinación que tengan un atributo común, más todas las tuplas de la relación de la izquierda que no tengan un equivalente en la relación de la derecha. “El resultado es NULL en el lado derecho cuando no hay coincidencia. En algunas bases de datos LEFT JOIN se denomina LEFT OUTER JOIN” (w3schools.com).

The diagram shows a SQL query with three callout boxes explaining its parts:

- Comando DML**: Points to the entire query.
- Cláusula para combinación interna**: Points to the `LEFT OUTER JOIN` clause.
- Columnas de relación de las tablas**: Points to the `ON Alumno.id_carrera=Carrera.id_carrera` condition.

```
SELECT * FROM Alumno
LEFT OUTER JOIN Carrera ON Alumno.id_carrera=Carrera.id_carrera
```


RIGHT OUTER JOIN

- Con esta cláusula se retorna todas las tuplas de la combinación que tengan un atributo común, más todas las tuplas de la relación de la derecha que no tengan un equivalente en la relación de la izquierda.

Comando DML

Cláusula para combinación interna

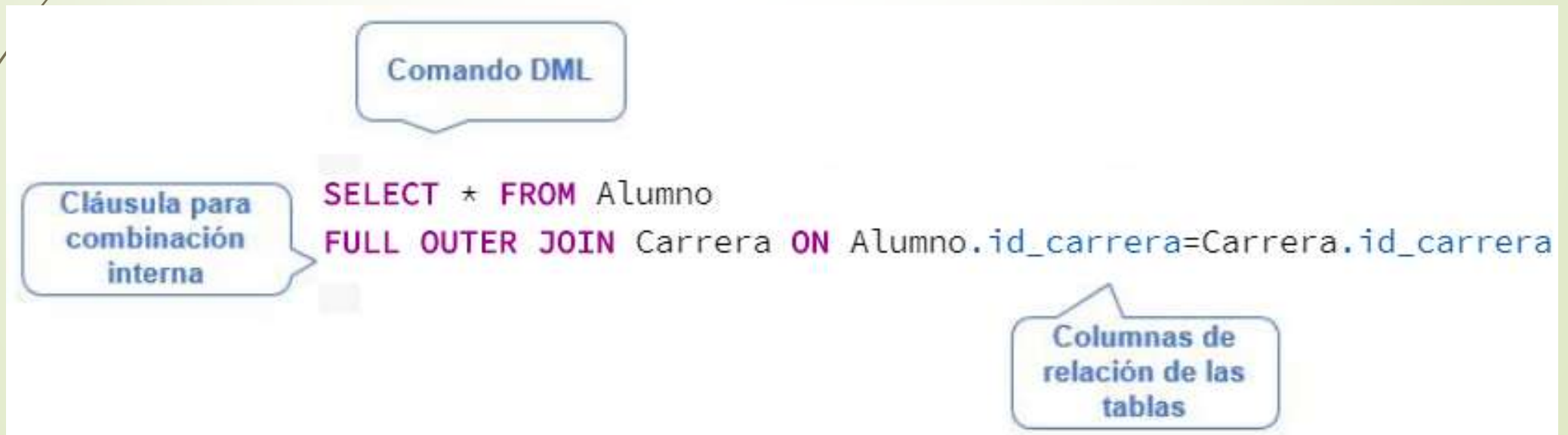
```
SELECT * FROM Alumno  
RIGHT OUTER JOIN Carrera ON Alumno.id_carrera=Carrera.id_carrera
```

Columnas de relación de las tablas

	id_alumno character (8)	id_carrera character (3)	apellidos character varying (40)	nombres character varying (40)	sexo character (1)	direccion character varying (40)	semestre character (3)	turno charac
1	20694075	CON	ALVAREZ CASAS	MIGUEKATTY JENIFER	F	JR. AREQUIPA 234	I	D
2	20694077	FAR	RAYGAL ORTIZ	CLAUDIA	F	JR. LEONCIO PRADO 230	II	D
3	20694078	AGR	CASAS CARHUANCHO	JULIO	M	JR. HUASCARIPA 2321	VI	D
4	20694079	MFC	RAFAEL SINCHE	JORGE LUIS	M	JR. HUANUCO 231	IV	D

FULL OUTER JOIN

- “El operador de combinación externa completa, FULL OUTER JOIN, que incluye todas las filas de ambas tablas, con independencia de que la otra tabla tenga o no un valor coincidente.” (Microsoft)
Mediante esta cláusula se retorna todas las tuplas de la combinación que tengan un atributo común y no común en ambas relaciones de izquierda y derecha



SELECT * FROM Alumno
FULL OUTER JOIN Carrera ON Alumno.id_alumno=Carrera.id_carrera

```
170 SELECT * FROM Alumno
171 FULL OUTER JOIN Prestamo ON Alumno.id_alumno=Prestamo.id_libro
172
```

[Data Output](#) [Explain](#) [Messages](#) [Notifications](#)

	sexo	direccion	semestre	turno	num_prestamo	id_alumno	id_libro	fecha_hora
	character (1)	character varying (40)	character (3)	character (1)	integer	character (8)	character (4)	date
	[null]	[null]	[null]	[null]	3	20694074	101	2019-10-03
	[null]	[null]	[null]	[null]	4	20694074	500	2019-10-03
L	F	AV. MARISCAL CASTILLA ...	VI	N	[null]	[null]	[null]	[null]
	M	JR. HUASCARIPA 2321	VI	D	[null]	[null]	[null]	[null]