

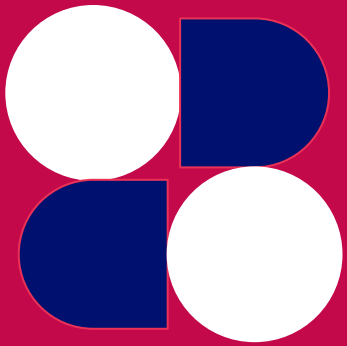


Programa PUCP DATA DRIVEN

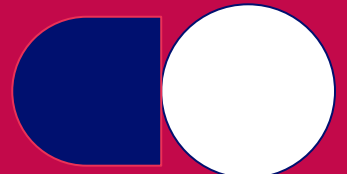
Formación tecnológica

Dirección Académica de Planeamiento y Evaluación
Dirección de Gestión del Talento Humano





SESIÓN 5: SQL





Al finalizar la sesión, lograrás:

- Implementar consultas SQL sobre un modelo relacional usando operadores adicionales



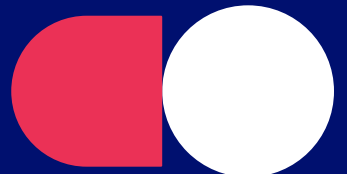
Contenido de la sesión:



- 1 Estructura Básica de Consulta SQL
- 2 Operaciones Básicas Adicionales
- 3 Valores Desconocidos
- 4 Taller



1. Estructura Básica de Consulta SQL





ESTRUCTURA BÁSICA SQL



Retomando La cláusula “SELECT”

- SQL permite duplicados tanto en las tablas como en los resultados de las consultas.
- Para forzar la eliminación de duplicados, se debe agregar la palabra clave **DISTINCT** después del **SELECT**.

Ejemplo: Encuentre los nombres de los departamentos de todos los instructores y elimine los duplicados

```
SELECT DISTINCT nombres_docente  
from model.[faci-inc_tb_docentes]
```

	ABC
	nombres docente
1	John
2	William
3	Bobby
4	Christopher
5	Nathan
6	Brian
7	Christopher
8	Sean
9	Peter
10	James
11	Richard
12	Andrew
13	Fred
14	John
15	James
16	Connor





ESTRUCTURA BÁSICA SQL



Retomando La cláusula “SELECT”

- La palabra reservada **ALL** especifica que los duplicados no deben eliminarse.

```
SELECT ALL nombres_docente  
from model.[faci-inc_tb_docentes]
```

- Por defecto las consultas son **ALL***

	ABC
	nombres_docente
1	John
2	William
3	Bobby
4	Christopher
5	Nathan
6	Brian
7	Christopher
8	Sean
9	Peter
10	James
11	Richard
12	Andrew
13	Fred
14	John
15	James
16	Connor





ESTRUCTURA BÁSICA SQL



Retomando La cláusula “SELECT”

- Un asterisco en la cláusula **SELECT** indica “**TODOS** los atributos”

SELECT *

from model.[faci-inc_tb_docentes]

- Un atributo puede ser un literal con cláusula from

SELECT 'A'

from model.[faci-inc_tb_docentes]

	ABC untitled1
1	A
2	A
3	A
4	A
5	A
6	A
7	A
8	A
9	A
10	A
11	A
12	A
13	A
14	A
15	A
16	A





Retomando La cláusula “SELECT”

- La cláusula **SELECT** puede contener expresiones aritméticas que involucran la operación, +, −, * y /, y que operan sobre constantes o atributos de tuplas.

- Ejemplo:

La consulta:

- ```
SELECT id_docente
 , nombres docente
 , CAST(dni AS FLOAT) / 10
 from [model].[faci-inc_tb_docentes]
```

Un pequeño cambio de  
tipo de dato para  
ilustrar el ejemplo





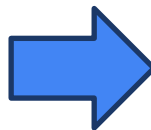
# ESTRUCTURA BÁSICA SQL



## Retomando La cláusula “SELECT”

- Devolverá una tabla que es igual a la tabla del `faci-inc_tb_docentes`, excepto que el valor del `dni` se divide por 10.

|      | ABC        | ABC             | 12 untitled3 |
|------|------------|-----------------|--------------|
|      | id_docente | nombres_docente |              |
| 477  | 685        | Aaron           | 25107624     |
| 1084 | 1044       | Aaron           | 41969313     |
| 1429 | 1472       | Aaron           | 51258842     |
| 571  | 171        | Abigail         | 79014971     |
| 809  | 918        | Abigail         | 96906069     |
| 877  | 584        | Abigail         | 43641678     |
| 888  | 596        | Abigail         | 83346804     |
| 1313 | 1544       | Abigail         | 30871714     |
| 71   | 150        | Adam            | 99352071     |
| 97   | 192        | Adam            | 96626878     |
| 105  | 207        | Adam            | 31242746     |
| 211  | 411        | Adam            | 53620454     |
| 332  | 860        | Adam            | 98889704     |
| 422  | 568        | Adam            | 91575032     |



|      | ABC        | ABC             | 12 untitled3 |
|------|------------|-----------------|--------------|
|      | id_docente | nombres_docente |              |
| 477  | 685        | Aaron           | 2510762.4    |
| 1084 | 1044       | Aaron           | 4196931.3    |
| 1429 | 1472       | Aaron           | 5125884.2    |
| 571  | 171        | Abigail         | 7901497.1    |
| 809  | 918        | Abigail         | 9690606.9    |
| 877  | 584        | Abigail         | 4364167.8    |
| 888  | 596        | Abigail         | 8334680.4    |
| 1313 | 1544       | Abigail         | 3087171.4    |
| 71   | 150        | Adam            | 9935207.1    |
| 97   | 192        | Adam            | 9662687.8    |
| 105  | 207        | Adam            | 3124274.6    |
| 211  | 411        | Adam            | 5362045.4    |
| 332  | 860        | Adam            | 9888970.4    |
| 422  | 568        | Adam            | 9157503.2    |





## Retomando La cláusula “WHERE”

- SQL permite el uso de los conectores lógicos AND, OR, y NOT para concatenar condiciones/filtros
- Los operandos de los conectores lógicos pueden ser expresiones que involucren a los operadores de comparación  
 $<$ ,  $\leq$ ,  $>$ ,  $\geq$ ,  $=$ , y  $\neq$
- **OJO:** Los operadores de igualdad o diferencia (  $=$  ,  $\neq$  ) aplican para todos los tipos de datos, pero los comparadores de mayor o menor (  $<$ ,  $\leq$ ,  $>$ ,  $\geq$  ) tienen más sentido al aplicarlos en tipos de datos numéricos, si se aplican en tipos de datos de texto tomará el orden alfabético
- Las comparaciones se pueden aplicar a los resultados de expresiones aritméticas directamente en WHERE





## Retomando La cláusula “WHERE”

- Ejemplo: Encontrar los nombres de todos los cursos ofrecidos en el ciclo ‘2024-2’ y que tengan un  $(\text{creditaje} * 2) > 8$

```
SELECT nombre_curso
from model.[faci-inc_tb_cursos]
WHERE ciclo = ‘2024-2’ and $(\text{creditaje} * 2) > 8$
```





# ESTRUCTURA BÁSICA SQL

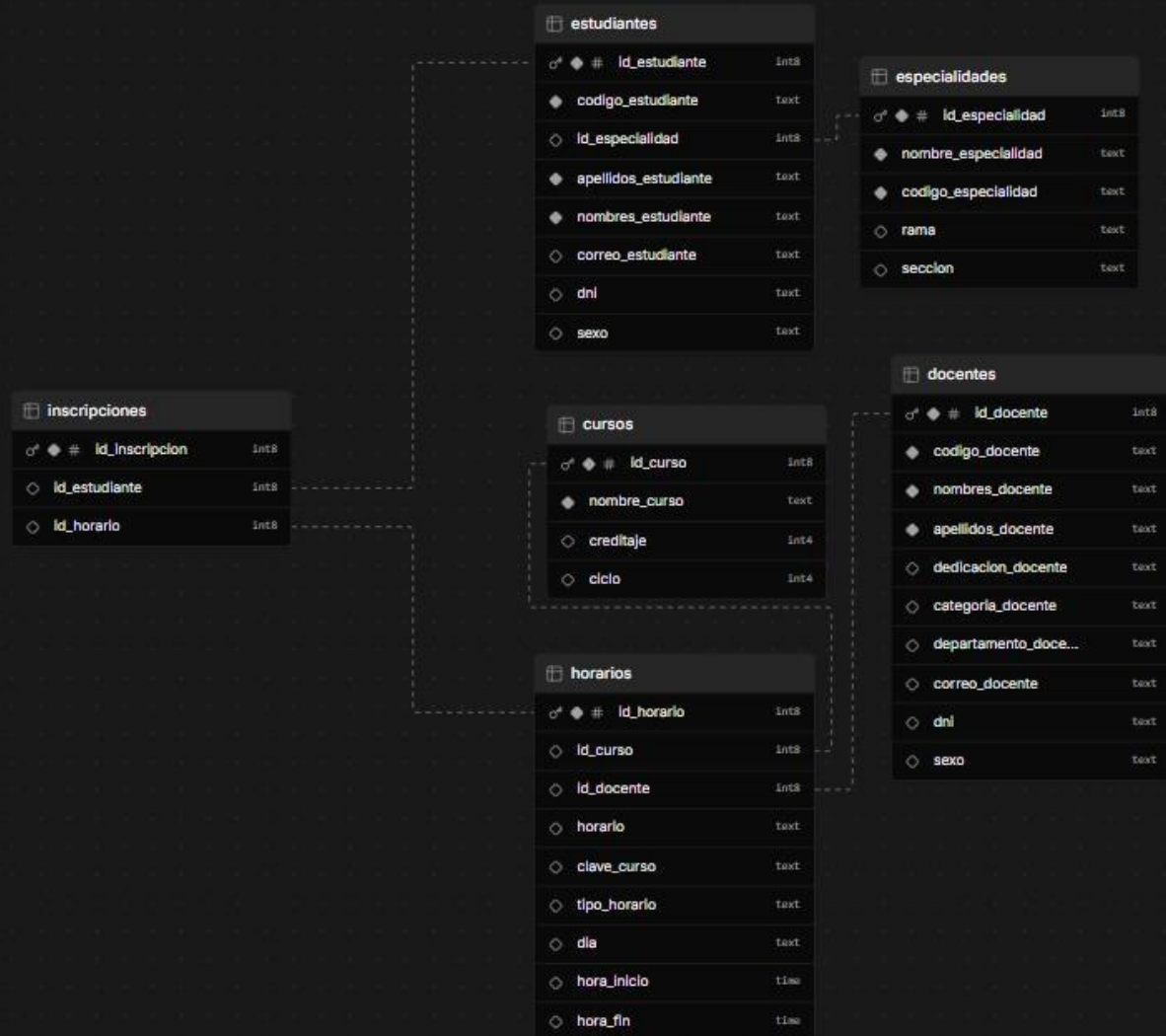


## Retomando La cláusula “WHERE”

```
SELECT nombre_curso
from model.[faci-inc_tb_cursos]
WHERE ciclo = '2024-2' and (creditaje*2) > 8
```

|    | ABC nombre_curso                      | ABC creditaje ↑ | ABC ciclo |
|----|---------------------------------------|-----------------|-----------|
| 1  | TEMAS AVANZADOS EN REDES Y TELEMÁTICA | 5               | 2024-2    |
| 2  | GESTIÓN DE LA INNOVACIÓN              | 5               | 2024-2    |
| 3  | MOTORES DE COMBUSTIÓN INTERNA         | 5               | 2024-2    |
| 4  | CREATIVIDAD E INNOVACIÓN              | 5               | 2024-2    |
| 5  | SEMINARIO                             | 5               | 2024-2    |
| 6  | PROYECTO DE TESIS 2                   | 5               | 2024-2    |
| 7  | INGENIERÍA DE RECURSOS HÍDRICOS       | 5               | 2024-2    |
| 8  | TEMAS ESPECIALES DE CONSTRUCCIÓN      | 5               | 2024-2    |
| 9  | PROYECTO DE INGENIERÍA INDUSTRIAL 2   | 5               | 2024-2    |
| 10 | PROYECTO DE INGENIERÍA INDUSTRIAL 2   | 5               | 2024-2    |
| 11 | INGENIERÍA DE TRÁFICO                 | 5               | 2024-2    |





## docentes

|                                                                                                                                                                       |                      |      |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|------|
|   # | id_docente           | int8 |
|                                                                                      | codigo_docente       | text |
|                                                                                      | nombres_docente      | text |
|                                                                                      | apellidos_docente    | text |
|                                                                                      | dedicacion_docente   | text |
|                                                                                      | categoria_docente    | text |
|                                                                                      | departamento_doce... | text |
|                                                                                      | correo_docente       | text |
|                                                                                      | dni                  | text |
|                                                                                      | sexo                 | text |



## EJERCICIOS

- Encuentre los nombres y apellidos de todos los estudiantes que hagan parte de la especialidad 8 y que sean de sexo 'M'
- Encuentre todos los horarios, id de curso, tipo de horario, día de clase, hora de inicio, y hora de fin para el curso con clave 'IND275', que ha sido dictado por el profesor con id 1092 o por el profesor con id 100, y su día de clase NO es el 'Lunes'





## EJERCICIOS

- Encuentre los nombres y apellidos de todos los estudiantes que hagan parte de la especialidad 8 y que sean de sexo 'M'

```
SELECT nombres_estudiante
 , apellidos_estudiante
FROM model.[faci-inc_tb_estudiantes]
WHERE id_especialidad = 8 AND sexo = 'M';
```

|   | ABC<br>nombres_estudiante | ABC<br>apellidos_estudiante |
|---|---------------------------|-----------------------------|
| 1 | Joseph                    | Morris                      |
| 2 | Darrell                   | Hamilton                    |
| 3 | Rickey                    | Pittman                     |
| 4 | Walter                    | Tucker                      |
| 5 | Peter                     | Oconnor                     |
| 6 | Andrew                    | Taylor                      |
| 7 | David                     | Reyes                       |
| 8 | William                   | Watkins                     |





# ESTRUCTURA BÁSICA SQL



## EJERCICIOS

- Encuentre todos los horarios, id de curso, tipo de horario, día de clase, hora de inicio, y hora de fin para el curso con clave 'IND275', que ha sido dictado por el profesor con id 1092 o por el profesor con id 100, y su día de clase NO es el 'Lunes'

**SELECT** horario

, id\_curso  
, tipo\_horario  
, dia  
, hora\_inicio  
, hora\_fin  
, id\_docente

**FROM** model.[faci-inc\_tb\_horarios]

**WHERE** clave\_curso = 'IND275' AND

(id\_docente = 1092 OR id\_docente = 100) AND

dia <> 'Lunes';

|   | ABC<br>horario | ABC<br>id_curso | ABC<br>tipo_horario | ABC dia   | ABC<br>hora_inicio | ABC<br>hora_fin |
|---|----------------|-----------------|---------------------|-----------|--------------------|-----------------|
| 1 | 0731           | 179             | CLA                 | Jueves    | 15:00:00           | 16:30:00        |
| 2 | 0733           | 179             | CLA                 | Miércoles | 19:00:00           | 20:30:00        |
| 3 | 0731           | 179             | EXA                 | Viernes   | 12:00:00           | 14:15:00        |
| 4 | 0736           | 179             | EXA                 | Jueves    | 18:00:00           | 21:00:00        |
| 5 | 0736           | 179             | PRA                 | Miércoles | 10:00:00           | 11:30:00        |





## La cláusula “AS”

- SQL permite renombrar tablas y atributos usando la cláusula **AS** (solo en la respuesta, no significa que el MR cambie):

Sintaxis:

old-name **AS** new-name





# ESTRUCTURA BÁSICA SQL



## La cláusula “AS”

- Ejemplos:

Encuentre los nombres de todos los estudiantes cuyo genero será 'M'.

```
SELECT nombres_estudiante AS NOMBRE
FROM model.[faci-inc_tb_estudiantes]
WHERE sexo = 'M';
```

|   | ABC NOMBRE |
|---|------------|
| 1 | Joseph     |
| 2 | Darrell    |
| 3 | Rickey     |
| 4 | Walter     |

Encuentre los nombres de todos los estudiantes cuyo genero será 'M'.

```
SELECT T.nombres_estudiante
FROM model.[faci-inc_tb_estudiantes] AS T
WHERE sexo = 'M';
```

|   | ABC nombres_estudiante |
|---|------------------------|
| 1 | Joseph                 |
| 2 | Darrell                |
| 3 | Rickey                 |
| 4 | Walter                 |

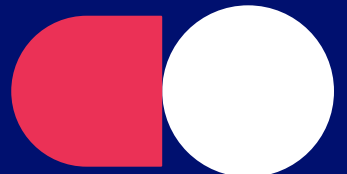
- La cláusula **AS** es opcional y puede omitirse

`model.[faci-inc_tb_estudiantes] AS T`  $\equiv$  `model.[faci-inc_tb_estudiantes] T`  
`nombres_estudiante AS NOMBRE`  $\equiv$  `nombres_estudiante NOMBRE`





## **2. Operaciones Básicas Adicionales**





# OPERACIONES BASICAS ADICIONALES



## Operadores de *Strings*

- SQL incluye un operador de coincidencia de cadenas para comparaciones en cadenas de caracteres.
- El operador **LIKE** usa patrones que se describen usando dos caracteres especiales:
  - Porcentaje ( % ). El % busca coincidencias con cualquier subcadena.
  - Guión bajo ( \_ ). El \_ busca cualquier caracter.





# OPERACIONES BASICAS ADICIONALES



## Operadores de *Strings*

- Ejemplo: Encuentre los nombres de todos los docentes cuyo nombre incluya la subcadena “an”.

```
SELECT nombres_docente
FROM model.[faci-inc_tb_docentes]
WHERE nombres_docente LIKE '%an%'
```

|   | ABC nombre |
|---|------------|
| 1 | Nathan     |
| 2 | Brian      |
| 3 | Sean       |
| 4 | Andrew     |
| 5 | Brian      |
| 6 | Brandon    |
| 7 | Nathaniel  |
| 8 | Anthony    |





# OPERACIONES BASICAS ADICIONALES



## Operadores de *Strings*

- Ejemplos de coincidencia de patrones:
  - 'intro%' coincide con cualquier cadena que comience por "intro".
  - '%comp%' coincide con cualquier cadena que contenga "comp" como subcadena en cualquier posición.
  - '\_\_\_' coincide con cualquier cadena de exactamente tres caracteres.
  - '\_\_\_ %' coincide con cualquier cadena de al menos tres caracteres.





# OPERACIONES BASICAS ADICIONALES



## Operadores de *Strings*

- Ejemplo: Encuentre los nombres de todos los docentes cuya longitud sean exactamente 4 letras, no permita repetidos.

```
SELECT DISTINCT nombres_docente
FROM model.[faci-inc_tb_docentes]
WHERE nombres_docente LIKE '____'
```

Acá hay ' \_ ' x4

|   | ABC nombres_docente |
|---|---------------------|
| 1 | Adam                |
| 2 | Alan                |
| 3 | Alex                |
| 4 | Anna                |
| 5 | Anne                |
| 6 | Beth                |
| 7 | Brad                |
| 8 | Carl                |
| 9 | Chad                |





# OPERACIONES BASICAS ADICIONALES



## Operadores de *Strings*

- SQL admite una variedad de operaciones de cadena, como por ejemplo:
  - Concatenación (usando “||”)
  - Convertir mayúsculas y minúsculas [y viceversa] (usando UPPER() o LOWER() respectivamente)
  - Encontrar la longitud de la cadena (usando LEN() )
  - Extraer subcadenas
  - Etc.





# OPERACIONES BASICAS ADICIONALES



## Operadores de *Strings*

- Ejemplo: Encuentre los nombres y apellidos de todos los docentes cuyo apellido sea de por lo menos 5 letras. Para la respuesta informe la cantidad de letras que tiene el apellido en una columna llamada “LONGITUD” y concatene los nombres y apellidos en una sola columna de respuesta titulada “NOMBRE\_COMPLETO” de tal manera que el nombre esté en minúsculas sostenidas y el apellido esté en mayúsculas sostenidas. No permita repetidos en “NOMBRE\_COMPLETO”

```
SELECT DISTINCT (LOWER(nombres_docente) ||
 UPPER(apellidos_docente)) AS
 NOMBRE_COMPLETO
 , LEN(apellidos_docente) AS 'LONGITUD'
FROM model.[faci-inc_tb_docentes]
WHERE apellidos_docente LIKE '_____%';
```





# OPERACIONES BASICAS ADICIONALES



## Operadores de *Strings*

```
SELECT DISTINCT (LOWER(nombres_docente) || UPPER(apellidos_docente)) AS NOMBRE_COMPLETO
 , LEN(apellidos_docente) AS 'LONGITUD'
FROM model.[faci-inc_tb_docentes]
WHERE apellidos_docente LIKE '_____%';
```

|   | ABC             |              |
|---|-----------------|--------------|
|   | NOMBRE_COMPLETO | 12L LONGITUD |
| 1 | aaronCERVANTES  | 9            |
| 2 | aaronMORALES    | 7            |
| 3 | aaronSTEWART    | 7            |
| 4 | abigailGILBERT  | 7            |
| 5 | abigailHAMILTON | 8            |
| 6 | abigailROBERTS  | 7            |
| 7 | abigailVASQUEZ  | 7            |
| 8 | abigailYOUNG    | 5            |
| 9 | adamBEASLEY     | 7            |





# OPERACIONES BASICAS ADICIONALES



## Ordenamiento de tuplas resultantes

- Para el ordenamiento se usa el modificador **ORDER BY** seleccionando la columna por la cual se desea ordenar
- **ORDER BY** se ubica después del WHERE (si lo hay)
- Ejemplo: Ordenar en orden alfabético los nombres de todos los docentes

```
SELECT nombres_docente
FROM model.[faci-inc_tb_docentes]
ORDER BY nombres_docente
```

|   | ABC nombres_docente |
|---|---------------------|
| 1 | Aaron               |
| 2 | Aaron               |
| 3 | Aaron               |
| 4 | Abigail             |
| 5 | Abigail             |
| 6 | Abigail             |





# OPERACIONES BASICAS ADICIONALES



## Ordenamiento de tuplas resultantes

- Podemos especificar **DESC** para orden descendente o **ASC** para orden ascendente, para cada atributo (el orden ascendente es el predeterminado)
- Ejemplo: Ordenar en orden descendente los nombres de todos los docentes

```
SELECT nombres_docente
FROM model.[faci-inc_tb_docentes]
ORDER BY nombres_docente DESC
```

|   | ABC nombres_docente |
|---|---------------------|
| 1 | Zachary             |
| 2 | Zachary             |
| 3 | Zachary             |
| 4 | Zachary             |
| 5 | Zachary             |
| 6 | Zachary             |
| 7 | Yvonne              |
| 8 | Yolanda             |
| 9 | Yolanda             |





# OPERACIONES BASICAS ADICIONALES



## Ordenamiento de tuplas resultantes

- Se puede ordenar en varios atributos. El orden es jerárquico, es decir, se orden primero por el primer atributo, después por el segundo, etc.
- Ejemplo: Ordenar en orden descendente los nombres de todos los docentes y ascendentemente sus apellidos.

```
SELECT nombres_docente
 , apellidos_docente
FROM model.[faci-inc_tb_docentes]
ORDER BY nombres_docente DESC, apellidos_docente
```

|   | ABC<br>nombres_docente | ABC apellidos_docente |
|---|------------------------|-----------------------|
| 1 | Zachary                | Banks                 |
| 2 | Zachary                | George                |
| 3 | Zachary                | Glenn                 |
| 4 | Zachary                | James                 |
| 5 | Zachary                | Mccoy                 |
| 6 | Zachary                | Perry                 |
| 7 | Yvonne                 | Evans                 |
| 8 | Yolanda                | Edwards               |
| 9 | Yolanda                | Li                    |



# OPERACIONES BASICAS ADICIONALES



## Predicados especiales de la cláusula “WHERE”

- SQL incluye un operador de comparación **BETWEEN**
- **BETWEEN** especifica el limite inferior y superior del filtro (conteniendo el límite). Tiene sentido para consultas numéricas
- Ejemplo: Encuentre los nombres de los cursos que tienen entre 2 y 5 créditos

```
SELECT nombre_curso
FROM model.[faci-inc_tb_cursos]
WHERE creditaje BETWEEN 2 AND 5
```

|   | ABC nombre_curso                      |
|---|---------------------------------------|
| 1 | TEMAS AVANZADOS EN REDES Y TELEMÁTICA |
| 2 | GESTIÓN DE LA INNOVACIÓN              |
| 3 | GESTIÓN DE LA INNOVACIÓN              |
| 4 | GESTIÓN DE LA INNOVACIÓN              |
| 5 | OLEOHIDRÁULICA Y NEUMÁTICA            |
| 6 | ESTRUCTURAS DE ACERO                  |



# OPERACIONES BASICAS ADICIONALES



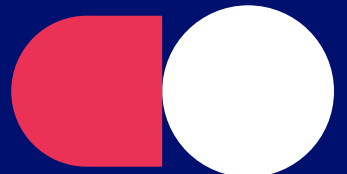
## Predicados especiales de la cláusula “WHERE”

- Comparación de tuplas (Tendrá sentido más adelante)





### **3. Valores Desconocidos**





## Valores “NULL”

- Es posible que las tuplas tengan un valor nulo, denotado por **NULL**, para algunos de sus atributos.
- **NULL** significa un valor desconocido o que no existe un valor.
- El resultado de cualquier expresión aritmética que implique **NULL** es **NULL**
- Ejemplo:  $5 + \text{NULL}$  resulta en NULL





## Valores “NULL”

- El predicado **IS NULL** se puede utilizar como filtro para comprobar si hay valores nulos.
- Ejemplo: Encuentre toda la información de los docentes cuyo departamento es nulo.

```
SELECT *
FROM model.[faci-inc_tb_docentes]
WHERE departamento_docente IS NULL
```

- El predicado **IS NOT NULL** se usa para identificar los valores no nulos





## EJERCICIOS



- Encuentre los nombres y apellidos de todos los estudiantes que hagan parte de la especialidad 5 o 6 o 7 u 8 o 9 (Use between), que sean de sexo 'F', y que su nombre empiece por "Ju"
- Encuentre todos los id de curso, id de docente, clave de curso, día de clase, hora de inicio y hora de fin para los cursos cuya clave inicie en 'IND', que hayan sido dictado por profesores con un id menor a 20, y que su día de clase NO sea el 'Viernes'. Para el resultado muestre la hora de inicio y la hora de fin en una sola columna con los datos separados por " – " y titulada "INI – FIN"





## EJERCICIOS



- Encuentre los nombres y apellidos de todos los estudiantes que hagan parte de la especialidad 5 o 6 o 7 u 8 o 9 (Use between), que sean de sexo 'F', y que su nombre empiece por "Ju"

```
SELECT nombres_estudiante
 , apellidos_estudiante
FROM model.[faci-inc_tb_estudiantes]
WHERE (id_especialidad BETWEEN 5 AND 9) AND
 sexo = 'F' AND
 nombres_estudiante LIKE 'Ju%';
```

|   | ABC<br>nombres_estudiante | ABC apellidos_estudiante |
|---|---------------------------|--------------------------|
| 1 | Judy                      | Lee                      |
| 2 | Julie                     | Butler                   |
| 3 | Julie                     | Beck                     |
| 4 | Julie                     | Marquez                  |
| 5 | Judy                      | Jones                    |
| 6 | Judy                      | Smith                    |
| 7 | Julie                     | George                   |
| 8 | Julie                     | Garza                    |
| 9 | Julie                     | Griffin                  |



## EJERCICIOS



- Encuentre todos los id de curso, id de docente, clave de curso, día de clase, hora de inicio y hora de fin para los cursos cuya clave inicie en 'IND', que hayan sido dictado por profesores con un id menor a 20, y que su día de clase NO sea el 'Viernes'. Para el resultado muestre la hora de inicio y la hora de fin en una sola columna con los datos separados por “ – “ y titulada “INI – FIN”





# EJERCICIOS



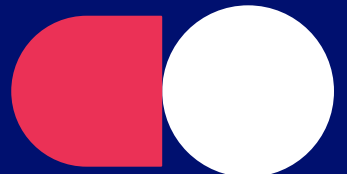
```
SELECT id_curso
 , id_docente
 , clave_curso
 , dia
 , (hora_inicio || ' - ' || hora_fin) AS 'INI - FIN'
FROM model.[faci-inc_tb_horarios]
WHERE clave_curso LIKE 'IND%' AND
 id_docente < 100 AND
 dia <> 'Lunes'
```

|   | ABC<br>id_curso | ABC<br>id_docente | ABC<br>clave_curso | ABC<br>dia | ABC INI - FIN       |
|---|-----------------|-------------------|--------------------|------------|---------------------|
| 1 | 239             |                   | IND231             | Jueves     | 18:00:00 - 19:30:00 |





## 4. Taller





# TALLER



- Documento “Taller\_IntroSQL.pdf”
- NO calificable
- Preferiblemente de manera individual para retar su conocimiento
- Desarrollo esperado durante la monitoria
- Al final de la monitoria se comparten las respuestas a todos los asistentes

**OJO:** Todos los caminos llegan a Roma





## BIBLIOGRAFÍA



Database system concepts / Abraham Silberschatz, Henry F. Korth and S. Sudarshan.  
Seventh Edition. 2019. Cap 2



**Gracias**

