

Hernandez Ramirez Miguel Angel

Bases de Datos Grupo: 1

08/05/2024

EJERCICIO DE CLASE

1) Separar el nombre del alumno y nombre del asesor (si aplica)

ALUMNO

```
UPDATE ALUMNO
SET
    nombre_Alumno = SPLIT_PART(nombre_Alumno, ' ', 1),
    ap_Paterno = SPLIT_PART(nombre_Alumno, ' ', 2);
```

```
ALTER TABLE ALUMNO
ADD ap_Paterno VARCHAR(25);
```

Output Messages Notifications

id_alumno	nombre_alumno	id_asesor	ap_paterno
integer	character varying (25)	character varying (25)	character varying (25)
1	Mauricio	as-1	Barrientos
2	Mario	as-2	Tabura
3	Luz	as-1	Rueda
4	Jorge	null	Santillan
5	Gabriel	as-3	Gaytan

ASESOR

```
UPDATE ASESOR
SET
    ap_Paterno = SPLIT_PART(nombre_Asesor, ' ', 2),
    nombre_Asesor = SPLIT_PART(nombre_Asesor, ' ', 1);
```

```
ALTER TABLE ASESOR
ADD ap_Paterno VARCHAR(25);
```

Output Messages Notifications

id_asesor	nombre_asesor	ap_paterno
character varying (25)	character varying (25)	character varying (25)
as-1	Jorge	Campos
as-2	Laura	Sandoval
as-3	Adolfo	Millan
as-4	Fernando	Arreola

2) Alumno U Asesor (sólo en el atributo nombre)

```
SELECT nombre_Alumno AS nombre
FROM ALUMNO

UNION

SELECT nombre_Asesor AS nombre
FROM ASESOR;
```

Output Messages Notifications

nombre
character varying (25)
Jorge
Mario
Adolfo
Gabriel
Laura
Fernando
Luz
Mauricio

3) Alumno INTERSECT Asesor (sólo en el atributo nombre)

```
SELECT nombre_Alumno AS nombre
FROM ALUMNO

INTERSECT

SELECT nombre_Asesor AS nombre
FROM ASESOR;
```

a Output Messages Notifications

nombre
character varying (25)
Jorge

4) Alumno - Asesor y Asesor – Alumno

```
SELECT nombre_Alumno AS nombre
FROM ALUMNO
EXCEPT
SELECT nombre_Asesor AS nombre
FROM ASESOR;
```

a Output Messages Notifications

nombre
character varying (25)
Gabriel
Luz
Mauricio
Mario

```
SELECT nombre_Asesor AS nombre
FROM ASESOR
EXCEPT
SELECT nombre_Alumno AS nombre
FROM ALUMNO;
```

a Output Messages Notifications

nombre
character varying (25)
Adolfo
Laura
Fernando

5) ALUMNO NATURAL JOIN ASESOR

```
SELECT *
FROM ALUMNO
NATURAL JOIN ASESOR;
```

a Output Messages Notifications

id_asesor	ap_paterno	id_alumno	nombre_alumno	nombre_asesor
character varying (25)	character varying (25)	integer	character varying (25)	character varying (25)

5.1) ALUMNO CROSS JOIN ASESOR

```
SELECT *
FROM ALUMNO
CROSS JOIN ASESOR;
```

Output Messages Notifications

id_alumno	nombre_alumno	id_asesor	ap_paterno	id_asesor	nombre_asesor	ap_paterno
integer	character varying (25)	character varying (25)	character varying (25)	character varying (25)	character varying (25)	character varying (25)
1	Mauricio	as-1	Barrientos	as-2	Laura	Sandoval
2	Mario	as-2	Tabura	as-2	Laura	Sandoval
3	Luz	as-1	Rueda	as-2	Laura	Sandoval
4	Jorge	null	Santillan	as-2	Laura	Sandoval
5	Gabriel	as-3	Gaytan	as-2	Laura	Sandoval
1	Mauricio	as-1	Barrientos	as-3	Adolfo	Millan
2	Mario	as-2	Tabura	as-3	Adolfo	Millan
3	Luz	as-1	Rueda	as-3	Adolfo	Millan
4	Jorge	null	Santillan	as-3	Adolfo	Millan
5	Gabriel	as-3	Gaytan	as-3	Adolfo	Millan
1	Mauricio	as-1	Barrientos	as-4	Fernando	Arreola
2	Mario	as-2	Tabura	as-4	Fernando	Arreola
3	Luz	as-1	Rueda	as-4	Fernando	Arreola
4	Jorge	null	Santillan	as-4	Fernando	Arreola
5	Gabriel	as-3	Gaytan	as-4	Fernando	Arreola

- 6) Modificar la tabla Alumno, agregue los siguientes atributos: carrera varchar(40) y edad smallint

```
ALTER TABLE ALUMNO
ADD carrera VARCHAR(40),
ADD edad SMALLINT;
```

```
SELECT * FROM ALUMNO
```

Output Messages Notifications

id_alumno	nombre_alumno	id_asesor	ap_paterno	carrera	edad
integer	character varying (25)	character varying (25)	character varying (25)	character varying (40)	smallint
1	Mauricio	as-1	Barrientos	[null]	[null]
2	Mario	as-2	Tabura	[null]	[null]
3	Luz	as-1	Rueda	[null]	[null]
4	Jorge	null	Santillan	[null]	[null]
5	Gabriel	as-3	Gaytan	[null]	[null]

- 7) Insertar los siguientes 5 registros:
 id_Alumno,nombre_Alumno, ap_paterno, carrera,edad,id_asesor
 6,Isaac,Lemus,Petrolera,30,as-1
 7,Gabriela,Suarezs,Industrial,24,as-3
 8,Pablo,Gonzalez,Computacion,23,as-2
 9,David,Rivera,Industrial,25,as-1
 10,Dayana,Plata,Computacion,24,as-4

	id_alumno	nombre_alumno	id_asesor	ap_paterno	carrera	edad
	integer	character varying (25)	character varying (25)	character varying (25)	character varying (40)	smallint
1	6	Isaac	as-1	Lemus	Petrolera	30
2	8	Pablo	as-2	Gonzalez	Computacion	23
3	9	David	as-1	Rivera	Industrial	25
4	10	Dayana	as-4	Plata	Computacion	24
5	1	Mauricio	as-1	Barrientos	Petrolera	27
6	2	Mario	as-2	Tabura	Telecomunicaciones	24
7	3	Luz	as-1	Rueda	Computacion	27
8	4	Jorge	null	Santillan	Industrial	25
9	5	Gabriel	as-3	Gaytan	Computacion	19
10	7	Gabriela	as-3	Suarezs	Industrial	24

8) Actualizar los 5 registros iniciales para asignar valores en los atributos agregados

1,Petrolera,27

2,Telecomunicaciones,24

3,Computacion,27

4,Industrial,25

5,Computacion,19

```
UPDATE ALUMNO
SET carrera = 'Petrolera', edad = 27
WHERE id_Alumno = 1;

UPDATE ALUMNO
SET carrera = 'Telecomunicaciones', edad = 24
WHERE id_Alumno = 2;

UPDATE ALUMNO
SET carrera = 'Computacion', edad = 27
WHERE id_Alumno = 3;

UPDATE ALUMNO
SET carrera = 'Industrial', edad = 25
WHERE id_Alumno = 4;

UPDATE ALUMNO
SET carrera = 'Computacion', edad = 19
WHERE id_Alumno = 5;
```

9) Validar que los datos coincidan con los esperado

	id_alumno integer	nombre_alumno character varying (25)	id_asesor character varying (25)	ap_paterno character varying (25)	carrera character varying (40)	edad smallint
1	1	Mauricio	as-1	Barrientos	Petrolera	27
2	2	Mario	as-2	Tabura	Telecomunicaciones	24
3	3	Luz	as-1	Rueda	Computacion	27
4	4	Jorge	null	Santillan	Industrial	25
5	5	Gabriel	as-3	Gaytan	Computacion	19
6	6	Isaac	as-1	Lemus	Petrolera	30
7	7	Gabriela	as-3	Suarezs	Industrial	24
8	8	Pablo	as-2	Gonzalez	Computacion	23
9	9	David	as-1	Rivera	Industrial	25
10	10	Dayana	as-4	Plata	Computacion	24

10) Nombre completo del alumno de mayor edad

```
SELECT nombre_Alumno || ' ' || ap_Paterno AS nombre_completo
FROM ALUMNO
ORDER BY edad DESC
LIMIT 1;
```

Output	Messages	Notifications
<div> <div>nombre_completo</div> <div>text</div> <div>Isaac Lemus</div> </div>		

11) Nombre completo del alumno de menor edad

```
SELECT nombre_Alumno || ' ' || ap_Paterno AS nombre_completo
FROM ALUMNO
ORDER BY edad ASC
LIMIT 1;
```

a Output Messages Notifications	
<div> <div>nombre_completo</div> <div>text</div> </div>	
Gabriel Gaytan	

12) Cantidad de alumnos por carrera

```
SELECT carrera, COUNT(*) AS cantidad_alumnos
FROM ALUMNO
GROUP BY carrera;
```

a Output Messages Notifications	
<div> <div>carrera</div> <div>character varying (40)</div> </div>	
<div> <div>cantidad_alumnos</div> <div>bigint</div> </div>	
Industrial	3
Telecomunicaciones	1
Computacion	4
Petrolera	2

13) Nombre de la carrera que tiene a la persona más joven. Agrupar los datos.

```
SELECT carrera
FROM ALUMNO
GROUP BY carrera
HAVING MIN(edad) = (
    SELECT MIN(edad)
    FROM ALUMNO
);
```

a Output Messages Noti	
<div> <div>carrera</div> <div>character varying (40)</div> </div>	
Computacion	

14) Nombre de la carrera que tiene a la persona más grande. Agrupar los datos.

```
SELECT carrera
FROM ALUMNO
GROUP BY carrera
HAVING MAX(edad) = (
    SELECT MAX(edad)
    FROM ALUMNO
);
```

ta Output Messages Not	
<div> <div>carrera</div> <div>character varying (40)</div> </div>	
Petrolera	

15) Nombre de la carrera que tiene a la persona más joven. Usar subconsultas.

```
SELECT carrera
FROM ALUMNO
WHERE edad = (
    SELECT MIN(edad)
    FROM ALUMNO
)
LIMIT 1;
```

Output Messages Noti

carrera
Computacion

16) Nombre de la carrera que tiene a la persona más grande. Usar subconsultas

```
SELECT carrera
FROM ALUMNO
WHERE edad = (
    SELECT MAX(edad)
    FROM ALUMNO
)
LIMIT 1;
```

Output Messages No

carrera
Petrolera

17) Promedio de edad por carrera

```
SELECT carrera, AVG(edad) AS promedio_edad
FROM ALUMNO
GROUP BY carrera;
```

Output Messages Notifications

carrera	promedio_edad
Industrial	24.666666666666667
Telecomunicaciones	24.000000000000000
Computacion	23.250000000000000
Petrolera	28.500000000000000

18) borrar al asesor Adolfo Millan

```
DELETE FROM ASESOR
WHERE nombre_Asesor = 'Adolfo' AND ap_paterno = 'Millan';
```

```
SELECT * FROM ASESOR
```

Output Messages Notifications		
id_asesor	nombre_asesor	ap_paterno
character varying (25)	character varying (25)	character varying (25)
as-1	Jorge	Campos
as-2	Laura	Sandoval
as-4	Fernando	Arreola

19) Actualizar el id del asesor Fernando Arreola, asignar "as-5"

```
SELECT * FROM ASESOR
```

```
UPDATE ASESOR
SET id_Asesor = 'as-5'
WHERE nombre_Asesor = 'Fernando' AND ap_paterno = 'Arreola';
```

Output Messages Notifications		
id_asesor	nombre_asesor	ap_paterno
character varying (25)	character varying (25)	character varying (25)
as-1	Jorge	Campos
as-2	Laura	Sandoval
as-5	Fernando	Arreola

20) inicie sesión en dos terminales, conectados en ambas a la base de datos donde almaceno los datos del presente ejercicio

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Contraseña para usuario postgres:

psql (16.2)
ADVERTENCIA: El código de página de la consola (437) difiere del código de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Vea la página de referencia de psql «Notes for Windows users» para obtener más detalles.
Digite «help» para obtener ayuda.
```

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Contraseña para usuario postgres:

psql (16.2)
ADVERTENCIA: El código de página de la consola (437) difiere del código de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Vea la página de referencia de psql «Notes for Windows users» para obtener más detalles.
Digite «help» para obtener ayuda.
```

a. en ambas sesiones, inicie una transacción

```
postgres=# BEGIN;
BEGIN
```

```
postgres=# BEGIN;
BEGIN
```

b. en la ventana A, actualice el apellido del alumno con id = 7 por "Suarez"

```
postgres=# UPDATE ALUMNO
postgres=# SET ap_paterno = 'Suarez
postgres=# WHERE id_Alumno = 7;
UPDATE 1
```

- c. en la ventana B, seleccione toda la información del alumno con id = 7. ¿Qué observa?

```
postgres=# SELECT *
postgres=# FROM ALUMNO
postgres=# WHERE id_Alumno = 7;
 id_alumno | nombre_alumno | id_asesor | ap_paterno | carrera | edad
-----+-----+-----+-----+-----+-----
          7 | Gabriela      | as-3      | Suarezs    | Industrial | 24
(1 fila)
```

El apellido no cambio, aunque lo hicimos en la ventana A

- d. en la ventaba A, ingrese commit

```
postgres=# COMMIT;
COMMIT
```

- e. en la ventana B, seleccione toda la información del alumno con id = 7, posteriormente rollback. ¿Qué observa?

```
postgres=# SELECT *
postgres=# FROM ALUMNO
postgres=# WHERE id_Alumno = 7;
 id_alumno | nombre_alumno | id_asesor | ap_paterno | carrera | edad
-----+-----+-----+-----+-----+-----
          7 | Gabriela      | as-3      | Suarez     | Industrial | 24
(1 fila)

postgres=# ROLLBACK;
ROLLBACK
```

```
postgres=# ROLLBACK;
ROLLBACK
postgres=# SELECT *
postgres=# FROM ALUMNO
postgres=# WHERE id_Alumno = 7;
 id_alumno | nombre_alumno | id_asesor | ap_paterno | carrera | edad
-----+-----+-----+-----+-----+-----
          7 | Gabriela      | as-3      | Suarez     | Industrial | 24
(1 fila)
```

- f. justifique los resultados obtenidos

El commit aplicado evita que el rollback elimine los cambios que se hicieron en el apellido del alumno con ID=7, por lo que se vuelve un cambio definitivo que no puede deshacer el ROLLBACK.