**UNIDAD 3: PostgreSQL**

1. [UNIDAD 2: Introducción](https://learn.nextu.com/mod/page/view.php?id=10253&pid=P_WEB_DATABASE)

**MySQL**

Con la idea de ampliar las opciones de herramientas que puedes utilizar para implementar tu solución de base de datos relacional, en esta unidad trabajaremos con el sistema manejador de base de datos PostgreSQL. En esta unidad buscamos que puedas aplicar los conceptos de bases de datos, que ya has aprendido en el curso usando este manejador, con la ayuda de la línea de comando y la herramienta gráfica PgAdmin. La idea es que te vayas familiarizando con la utilización de los conceptos de bases de datos relacionales y su representación en PostgreSQL. Seguimos usando nuestro enfoque teórico-práctico, para que con la ayuda de talleres y ejercicios puedas explorar y practicar los comandos y sentencias explicadas en video y lecturas.

**Objetivos de aprendizaje**

1. Interpretar las diferentes opciones para definir usuarios y sus correspondientes permisos con la finalidad de restringir el acceso a tus datos a diferentes tipos de usuarios. Crear claves primarias y foráneas y sus restricciones a una solución de base de datos en MySQL.
2. Diferenciar los distintos tipos de funciones predefinidas en PostgreSQL con la finalidad de realizar operaciones más avanzadas sobre nuestra base de datos.
3. Crear vistas en PostgreSQL con la finalidad de visualizar el contenido de nuestras tablas más apropiadamente de acuerdo con el alcance de las consultas.
4. Aplicar restricciones y usar secuencias.

2. Lección 1: Conociendo PostgreSQL

[2.1. PSQL](https://learn.nextu.com/mod/lesson/view.php?id=10261&pid=P_WEB_DATABASE)

Acceder por consola con el comando:

Cliente – Servidor

Cliente: psql

Comando:

>psql -U postgres -W -h localhost

Argumentos

-W 🡪 es para que el sistema pida la clave

-h 🡪 localhost o IP

-d 🡪 Nombre de la base de datos para ingresar directamente en la conexión

Comandos postgres:

Postgres=# \l 🡪 Para listar las DBs

Postgres=# CREATE DATABASE practicando; 🡪 Para crear una DB con el nombre practicando

Postgres=# \c practicando; 🡪 Para cambiar a la base de datos practicando (todo en posgres tiene un usuario y una contraseña)

practicando=# \dt 🡪 Para listar tablas

practicando=# CREATE TABLE usuarios(

practicando(# id SERIAL, 🡪 Incrementar la clave y no nulo

practicando(# nombre VARCHAR NOT NULL); 🡪 Para crear tablas

practicando=# \d+ usuarios 🡪 Describe la tabla y para salir se presiona q

practicando=# INSERT INTO usuarios (nombre) VALUES (‘Luis’); 🡪 Para Insertar datos

practicando=# \q 🡪 Para salir del prompt de PostgreSQL

[2.2. Taller 1](https://learn.nextu.com/mod/lesson/view.php?id=10261&pageid=46661&pid=P_WEB_DATABASE)

Descargar e instalar PostgreSQL

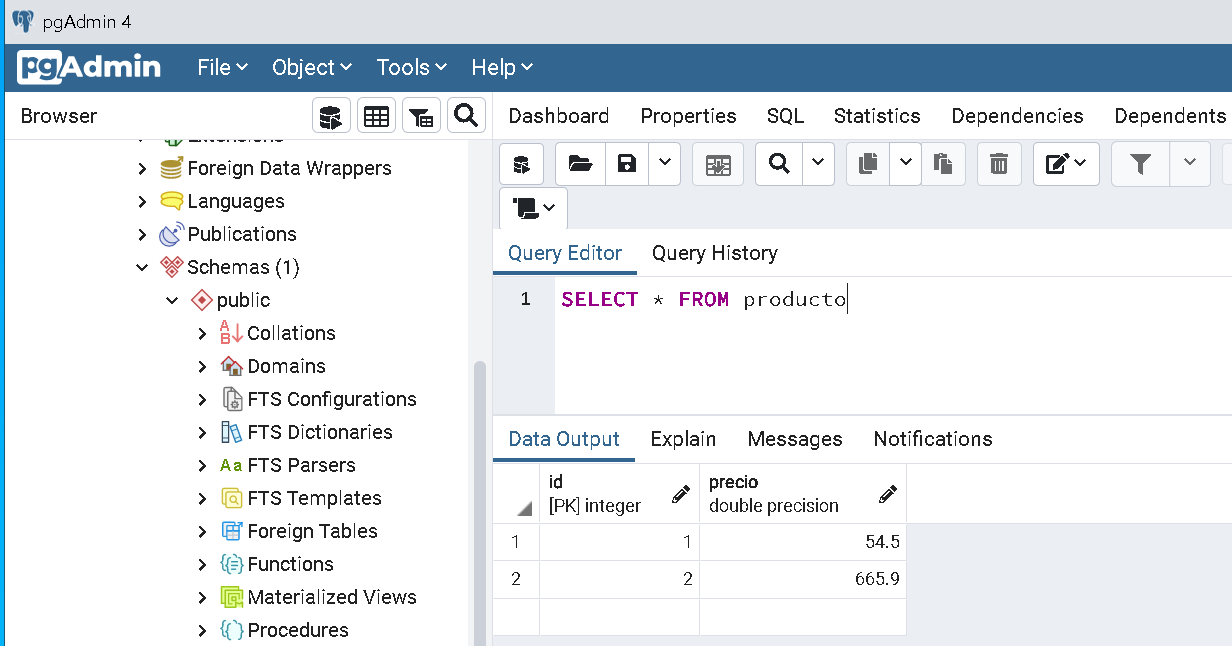
<https://www.postgresql.org/download/>

<https://www.pgadmin.org/>

2.3. Taller 2

2.4. PGAdmin

2.5. Taller 3



2.6. ¿Sabías qué?

**Tipos de Datos en PostgreSQL**

¡Bienvenido!

En esta lectura te proporcionamos un resumen de los tipos de datos más utilizados en PostgreSQL, categorizados en tres grupos:



**Strings**

| **Tipo** | **Descripción** |
| --- | --- |
| **CHAR(largo)** | Secuencia de caracteres (string) de tamaño fijo |
| **VARCHAR(largo)** | Secuencia de caracteres (string) de tamaño variable |
| **TEXT** | String largos |
|  |  |



**Numéricos**

| **Tipo** | **Descripción** |
| --- | --- |
| **SMALLINT** | Entero desde -32768 a 32767 (usado sin signo para booleanos) |
| **INTEGER** | Entero desde -2147483648 a 2147483647 |
| **BIGINT** | Entero desde -9223372036854775808 a 9223372036854775807 |
| **FLOAT** | Decimal Pequeño (8 bytes) |
| **REAL** | Decimal Grande (8 bytes) |
| **NUMERIC(n,s)** | Decimal con n dígitos normales y sin dígitos en la parte decimal |



**Otros**

| **Tipo** | **Descripción** |
| --- | --- |
| **DATE** | Fecha |
| **TIMESTAMP** | Fecha y hora HH:MM:SS |
| **TIME** | Hora |
| **SERIAL** | Entero autoincremental |
| **BOOLEAN** | True o false (verdadero o falso) |
| **CIRCLE** | Círculos |
| **POINT** | Punto en el plano |
| **MONEY** | Dinero |
| **CIDR** | Dirección IP |

Esperamos que esta guía sea útil a la hora de escoger tus datos en PostgreSQL.

3. Lección 2: Triggers y Stored Procedures

[3.1. Triggers](https://learn.nextu.com/mod/lesson/view.php?id=10262&pid=P_WEB_DATABASE)

Click derecho en “Trigger Functions”.

1. Primero se define el nombre del trigger
2. Luego se define el lenguaje para crear el trigger (opciones: internal, c o plpgsql)
3. Sí se selecciona plpgsql, se procede a ingresar el código(code), el cual debe ser un SQL (siempre se debe retornar un valor), por ejemplo:

BEGIN

INSERT INTO cambios\_de\_nombre (descripción, fecha) VALUES

(CONCAT(‘Cambio de nombre de: <’, OLD.nombre, ‘> a <’, NEW.nombre,’>’), NOW());

RETURN NEW;

END

1. Luego se debe ir a la tabla donde se desea que se dispare el trigger, se expande la tabla y se busca 🡪 Triggers, botón derecho y Trigger, se ingresa el nombre del trigger y luego en Definition se define cual es el Trigger que se desea disparar, por ejemplo:

Public.registrar\_cambio\_de\_nombre

1. Por último en la sección de Eventos se define cuando se quiere disparar el Trigger: BEFORE, AFTER o INSERT; UPDATE; DELETE or TRUNCATE, y se determina cual columna va a afectar (nombre)

[3.2. Ejercicio Práctico 1](https://learn.nextu.com/mod/lesson/view.php?id=10262&pageid=46671&pid=P_WEB_DATABASE)

3.3. Stored Procedures

Postgres no tiene el objeto SP definidos con ese nombre.

Lo que nos ofrece son las funciones, se pueden invocar cuando sea requerido

3.4. Ejercicio Práctico 2

3.5. ¿Sabías qué?

**Cápsula de conocimiento**

**Triggers y Funciones de PostgreSQL**

Aquí un resumen de algunos aspectos interesantes sobre triggers y funciones.

**Triggers**

1. Los triggers se implementan en dos partes:
   * La función que hace el cambio.
   * La asociación del trigger a la tabla.
2. El código asociado a la función debe estar entre **BEGIN** y **END**.
3. El objeto **NEW** representa al nuevo registro y **OLD** al registro antes de los cambios.
4. La función del trigger suele devolver el objeto **NEW** si lo modifica, y en caso contrario devuelve **NULL.**
5. Los triggers que se ejecutan antes (**BEFORE**) de una condición, pueden modificar los campos asignando **NEW.[campo] = [valor]**.
6. Los triggers que se ejecutan después (**AFTER**) de una condición, modifican los campos ejecutando un SQL como **UPDATE**, **INSERT**, etc.

**Funciones**

1. En PostgreSQL no existen Stored Procedures, en lugar de eso se usan las funciones de una forma general.
2. El código asociado a la función debe estar entre **BEGIN** y **END**.
3. Una función se puede definir con valor de retorno void, para lo cual no debe hacer **RETURN** al final.
4. Aunque la gente suele usar el lenguaje de PostgreSQL (plpgsql) para definir las funciones, estas pueden estar escritas en otros lenguajes como C, Python, Perl, Ruby, habilitando esos otros lenguajes.

Esperamos que esta información sea útil cuando utilices las funciones y triggers en PostgreSQL.

4. Lección 3: Funciones Predefinidas

[4.1. Funciones Matemáticas](https://learn.nextu.com/mod/lesson/view.php?id=10263&pid=P_WEB_DATABASE)

PGAdmin

Constantes:

Pi()

Sqrt() 🡪 Raíz cuadrada

Round() 🡪 Para redondear

4.2. Taller 1

4.3. Funciones de String y Fecha

CONCAT()

TRIM()

LOWER()

UCASE()

PostgreSQL a diferencia de MySQL si hace diferencia entre mayúsculas y minúsculas cuando se ejecuta un LIKE. Para no diferencias entre mayúsculas y minúsculas se usa la función ILIKE

Otra solución es utilizar el operador ~ para expresiones regulares, pero es sensible a mayúsculas y minúsculas, para no hacer sensible a mayúsculas y minúsculas se utiliza ~\*

Para hacer operaciones con fechas

Fecha + 20 🡪 En este caso se obtendrá la fecha 20 días después

EXTRACT(month FROM fecha) AS mes

Función para calcular la edad: AGE, requiere de una fecha

4.4. Taller 2

4.5. ¿Sabías qué?

**Cápsula de conocimiento**

**Funciones Predefinidas de PostgreSQL**

PostgreSQL proporciona una amplia variedad de funciones predefinidas, que permiten manejar los datos de nuestra base de datos. En esta lectura encontrarás un conjunto de las funciones que son consideradas como las más usadas. Esta guía busca ayudar a familiarizarte con las funciones en PostgreSQL.



**Operadores**

| **Función** | **Descripción** |
| --- | --- |
| **+** | SUMA |
| **-** | RESTA |
| **/** | DIVISIÓN |
| **\*** | MULTIPLICACIÓN |



**Strings**

| **Función** | **Descripción** |
| --- | --- |
| **CONCAT ||** | Concatenar |
| **REPLACE** | Reemplazar |
| **LOWER** | Cambiar a minúsculas |
| **UCASE** | Cambiar a mayúsculas |
| **TRIM** | Quitar espacios al comienzo y final |
| **SUBSTR** | Substring |



**Matemáticas**

| **Función** | **Descripción** |
| --- | --- |
| **EXP** | Función exponencial |
| **FLOOR** | Piso de un número |
| **POWER** | Potencia |
| **ROUND** | Redondeo |
| **SQRT** | Raíz cuadrada |



**Otras**

| **Función** | **Descripción** |
| --- | --- |
| **AGE** | Edad |
| **NOW** | Fecha y hora actual |
| **AREA** | Área |
| **Center** | Centro de un objeto |

Recuerda que existen más funciones en PostgreSQL y te invitamos a indagar sobre ellas en el sitio oficial de PostgreSQL. Por el momento, esperamos que encuentres beneficiosa esta lectura en tu proceso de aprendizaje de las bases de datos relacionales.

4.6. Taller 3

5. Lección 4: Subqueries y Otros Objetos

[5.1. Vistas](https://learn.nextu.com/mod/lesson/view.php?id=10264&pid=P_WEB_DATABASE)

Las vistas devuelven resultados parciales de datos.

Las vistas consisten en el resultado de un query pudiéndose guardar ese resultado en una tabla llamada vista y la cual se puede realizar consultas.

Se crea un nombre de vista y una definición (la consulta de los valores que se quieren que retorne)

PostgreSQL tiene dos tipos de vistas:

* + - Vistas: son vistas y se calculan en el momento
    - Vistas Materializadas: crea unos resultados a partir de una consulta. Sirven para datos que no cambien frecuentemente. Para aparezcan datos se debe ejecutar el comando refresh. Click derecho en la vista materializada y seleccionar refresh view with data

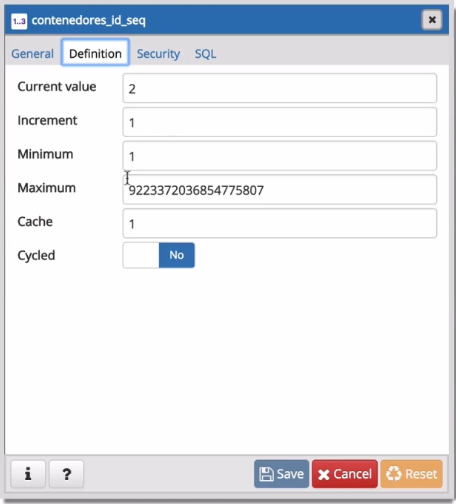
Vistas son útiles para reportes o dashboards.

5.2. Ejercicio

5.3. Secuencias y Restricciones

Con los campos seriales postgres crea unos objetos tipo secuencia.

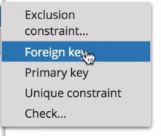
Para ver propiedades se realiza click derecho properties, donde se puede observar:



Y los valores pueden ser cambiados.

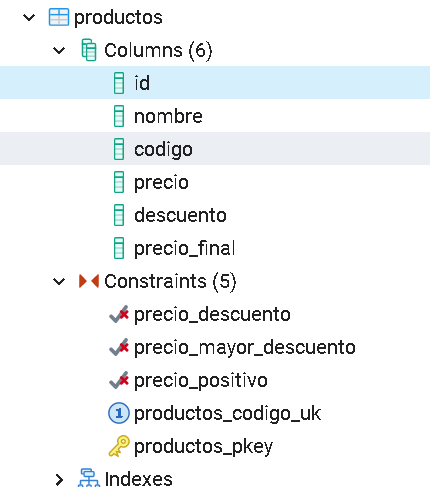
Además postgre permite chequeos de restricciones mucho más completos que otros manejadores de base de datos.

En el objeto tabla se puede observar los contraints y los tipos son:

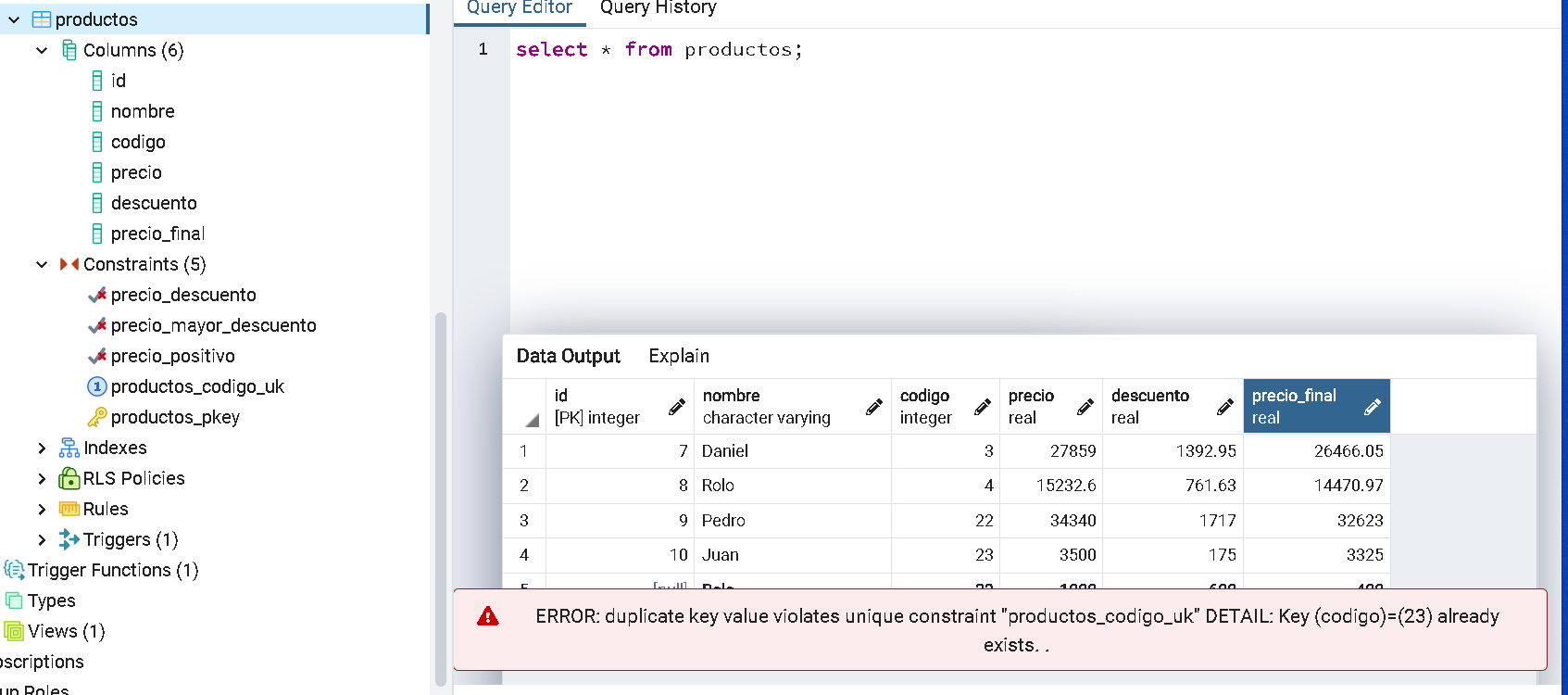


5.4. Taller

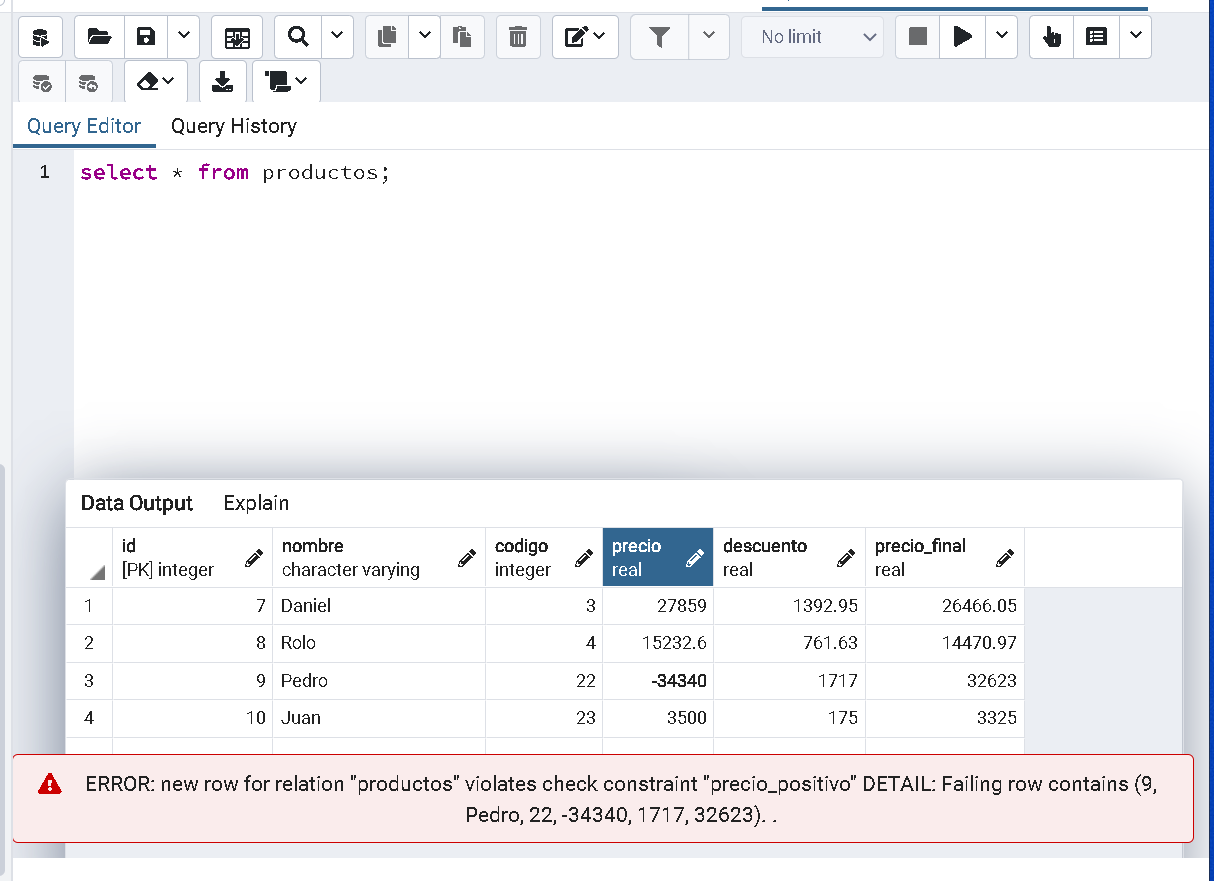
Punto 1.



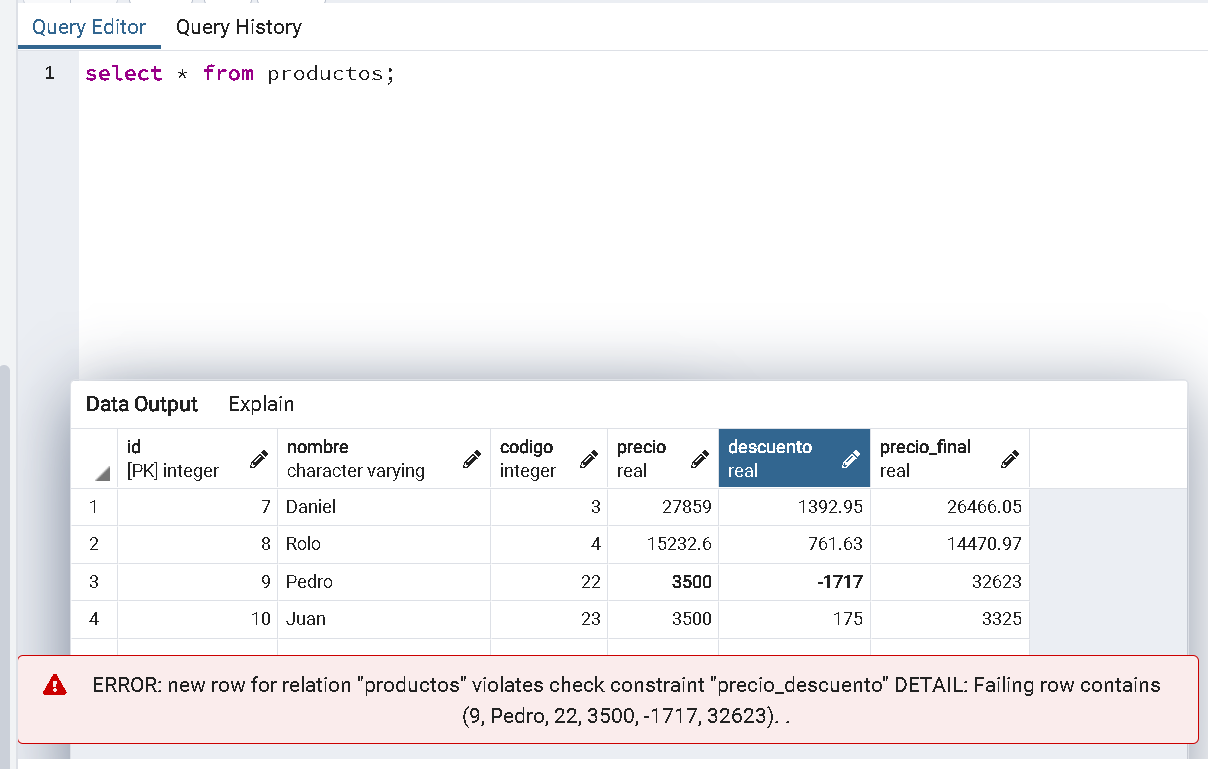
Punto 2.



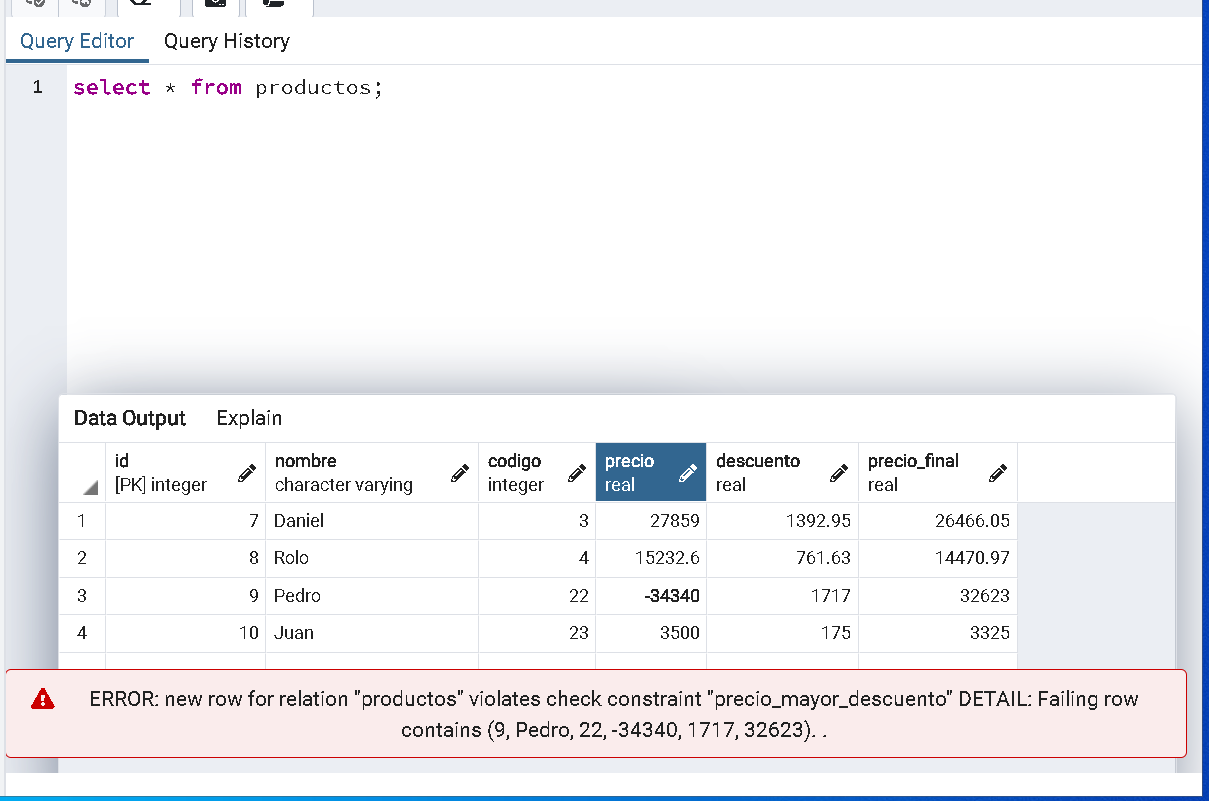
Punto 3.



Punto 4.



Punto 5.



5.5. ¿Sabías qué?

**Cápsula de conocimiento**

**Buenas Prácticas para las Vistas**

Las vistas pueden resultar de utilidad en ciertas ocasiones. Aquí te mostramos los escenarios donde puedes usar las mismas:



**Seguridad**: Muchas veces quieres que la gente pueda ver ciertos datos, pero que no tenga acceso a tus tablas, por lo cual puedes crear una vista que tenga lo que te interesa compartir.



**Simplicidad**: Las vistas pueden contener datos que vengan de distintas tablas abstrayendo al usuario de la fuente. Por lo tanto, pueden simplificar la manera de acceder y unir datos mostrando lo necesario.



**Organización**: Las vistas pueden organizar nuestros datos desde distintos puntos de vista, por ejemplo, la vista de producto para el cliente no es igual a la vista de un producto para el fabricante. Entonces podemos organizar y crear distintos puntos de vista de los mismos datos con fines distintos.



**Capa de Servicio**: Las vistas proporcionan información a terceros que aún si las estructuras fuentes cambian, se puede alterar la vista para que devuelva lo mismo, de manera de ser transparente para el que la usa.

Es importante resaltar que MySQL soporta Vistas (VIEWS) pero no Vistas materializadas (MATERIALIZED VIEWS).

[6. UNIDAD 3: Prueba](https://learn.nextu.com/mod/quiz/view.php?id=10266&pid=P_WEB_DATABASE)

|  |  |
| --- | --- |
| **Comenzado en** | Thursday, 6 de May de 2021, 18:42 |
| **Estado** | Terminados |
| **Finalizado en** | Thursday, 6 de May de 2021, 18:49 |
| **Tiempo empleado** | 6 mins 53 segundos |
| **Puntos** | 10/10 |
| **Calificación** | **100** de un total de 100 |

Top of Form

### Pregunta 1

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  es la base de datos por defecto a la cual nos conectamos en psql.

Seleccione una:

a. una base de datos con el mismo nombre del usuario de la conexión

b. Template0

c. Template1

d. Practicando

#### Retroalimentación

**Cuando no especificamos una base de datos usando la opción -d , pqsl nos lleva a la base de datos por defecto la cual es la del usuario**

La respuesta correcta es: una base de datos con el mismo nombre del usuario de la conexión

### Pregunta 2

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

¿Cuáles son los pasos para crear una función que implemente un Stored Procedure en Postgres?

Seleccione una o más de una:

a. Elaboramos un query que implemente el código que realiza las acciones que deseamos que se ejecuten y lo salvamos para usarlo luego.

b. Creamos una función desde el menú de Funciones (Functions) y le damos un nombre

c. Definimos la función creando rellenando en la parte del código el conjunto de sentencias escritas en el lenguaje seleccionado.

d. Indicamos cuando deseamos que se active la función.

#### Retroalimentación

**La creación del query, la creación de la función y la definición de la función son los pasos recomendados para crear una función que implemente un Stored Procedure en PostgreSQL.**

Las respuestas correctas son: Elaboramos un query que implemente el código que realiza las acciones que deseamos que se ejecuten y lo salvamos para usarlo luego., Creamos una función desde el menú de Funciones (Functions) y le damos un nombre, Definimos la función creando rellenando en la parte del código el conjunto de sentencias escritas en el lenguaje seleccionado.

### Pregunta 3

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

PostgreSQL usa \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ para crear tablas en una base de datos.

Seleccione una:

a. Sus propias sentencias

b. Sentencias SQL

c. Sentencias NoSQL

d. Sentencias psql

#### Retroalimentación

**PostgreSQL es un manejador de base de datos relacionales SQL por lo tanto usa el mismo conjunto de sentencias de SQL para crear tablas.**

La respuesta correcta es: Sentencias SQL

### Pregunta 4

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ es una tabla que incluye parte de las columnas de una tabla, donde los resultados de una consultan no se almacenan sino por el contrario se calculan al momento.

Seleccione una:

a. Vistas

b. Vistas materializadas

c. Tabla parcial

d. Query

#### Retroalimentación

Postgres tiene dos tipos de vistas: las vistas normales y las materializadas. En las vistas normales los resultados de la consultas no se almacenan.

La respuesta correcta es: Vistas

### Pregunta 5

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ es el tipo de restricción de Postgres más adecuado para garantizar que el valor de un campo es único.

Seleccione una:

a. Restricción de único.

b. Restricción de chequeo.

c. Restricción de clave primaria .

d. Restricción de exclusión.

#### Retroalimentación

**Postgres tiene cuatro tipos de restricción pero la más adecuada para garantizar que el valor de un campo es único es la restricción que lleva este nombre.**

La respuesta correcta es: Restricción de único.

### Pregunta 6

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

Haga coincidir los nombres de las funciones en la columna de la izquierda con su respectiva  definición en la columna de la derecha.

|  |  |
| --- | --- |
| plpsql | Lenguaje en PostgreSQL - Respuesta 1 |

|  |  |
| --- | --- |
|  | |
| PGADmin | | Una herramienta gráfica para interactuar con usuarios - Respuesta 2 | | | |
| psql | Un cliente basadi en línea de commandos de PostgreSQL - Respuesta 3 | | |
| Trigger function | | | Es la forma de implementar triggers en PostgreSQL - Respuesta 4 | |

#### Retroalimentación

La respuesta correcta es: plpsql → lenguaje en PostgreSQL, PGADmin → una herramienta gráfica para interactuar con bases de datos PostgreSQL, psql → un cliente basado en línea de comandos de PostgreSQL, Trigger function → es la forma de implementar triggers en PostgreSQL

### Pregunta 7

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

Supongamos que se desea  estimar el peso promedio de los niños que asisten a una escuela. La información está en la base de datos llamada **escuela**, en la tabla **ninos**, que incluye entre otros campos el peso y la altura de estos niños. Pepe es el nuevo administrador de la base de datos y se está familiarizando con la herramienta PGAdmin. El es el encargado de obtener estos resultados, Ayude a Pepe a realizar los pasos en el orden correcto .Supongamos existe una conexión llamada servidor\_de\_la\_escuela.

Seleccione una:

a. Conectarse al servidor\_de\_la\_escuela con credenciales adecuadas; seleccionar  base de datos escuela; seleccionar la tabla ninos; revisar la tabla ninos; ir  a Query Tools para realizar la consulta que incorpore la operación matemática y ejecutar

b. Conectarse al servidor\_de\_la\_escuela con credenciales adecuadas; seleccionar  base de datos escuela; revisar la tabla ninos y ejecutar.

c. Conectarse al servidor\_escuela con credenciales adecuadas; seleccionar  base de datos escuela; seleccionar la tabla ninos; revisar la tabla ninos; realizar consulta que incorpore la operación matemática y ejecutar

d. Seleccionar  base de datos escuela; seleccionar la tabla ninos; revisar la tabla ninos; ir  a Query Tools para realizar la consulta que incorpore la operación matemática y ejecutar

#### Retroalimentación

**Los pasos para cumplir el requerimiento son los siguientes: Conectarse al servidor\_de\_la\_escuela con credenciales adecuadas; seleccionar  base de datos escuela; seleccionar la tabla ninos; revisar la tabla ninos; ir  a Query Tools para realizar la consulta que incorpore la operación matemática y ejecutar**

La respuesta correcta es: Conectarse al servidor\_de\_la\_escuela con credenciales adecuadas; seleccionar  base de datos escuela; seleccionar la tabla ninos; revisar la tabla ninos; ir  a Query Tools para realizar la consulta que incorpore la operación matemática y ejecutar

### Pregunta 8

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ nos permite realizar cualquier Query sobre nuestra base de datos.

Seleccione una:

a. Query Tools

b. Table/Properties (Tabla/Propiedades)

c. Schemas (Esquemas)

d. View/Edit data (Ver/Editar) data

#### Retroalimentación

**Desde el menú Tool, podemos acceder Query Tool para ingresar a la interfaz que permite realizar cualquier tipo de consultas o queries a la base de datos en las que nos encontramos.**

La respuesta correcta es: Query Tools

### Pregunta 9

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

Un trigger se implementa en PostgreSQL como \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Seleccione una:

a. Una función

b. Un procedimiento.

c. Una tabla

d. Un procedimiento en otro lenguaje.

#### Retroalimentación

En postgres los triggers se implementan usando funciones.

La respuesta correcta es: Una función

### Pregunta 10

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ es el primero paso al ingresar a la herramienta PGAdmin.

Seleccione una:

a. Conectarnos al servidor de la base de datos

b. Crear una base de datos

c. Monitorear la base de datos postgres

d. Crear una tabla

#### Retroalimentación

Al ingresar a la herramienta PGAdmin debemos conectarnos al servidor de la base de datos, de lo contrario no podremos realizar ninguna operación sobre nuestras bases de datos.

La respuesta correcta es: Conectarnos al servidor de la base de datos