IIC2413

AYUDANTÍA 6

PSQL y PHP

PSQL

¿Qué es?

- psql es una interfaz de terminal para PostgreSQL
- Permite escribir consultas de forma interactiva, enviarlas a PostgreSQL y ver los resultados.
- También se puede ingresar información desde un archivo o mediante argumentos de línea de comandos.



Postgresql

- ¿Qué es? Sistema de gestión de bases de datos relacionales (RDBMS) de código abierto.
- ¿Para qué sirve? Almacena, organiza y gestiona datos complejos de forma segura.
- Casos de uso: Ideal para aplicaciones web, análisis de datos y sistemas empresariales.
- Nuestro entorno de trabajo: Se encuentra instalado y ejecutándose en el servidor de Ubuntu (stonebraker).

Comandos Linux/Ubuntu

- ls (listar)
- mkdir (crear carpeta)
- touch (crear archivo)
- nano (editor integrado en la terminal)
- rm (eliminar archivo)
- rm -r (eliminar carpeta) / rm -rf
- cd (cambiar de directorio)
- pwd (muestra la ruta del directorio de trabajo actual)
- cat (muestra contenido del archivo)

ubuntu

Ubuntu es un sistema operativo de código abierto basado en Linux

Comandos psql

Por defecto para conectarse a postgres se utiliza el siguiente comando

```
psql
```

Si no funciona tratar con

```
psql postgres
```

Comandos psql

4. Para ejecutar un archivo SQL

```
psql -f archivo.sql
```

i. Desde la consola de psql

```
ayudantia=# \i consulta_1.sql
```

ii. Desde terminal de Ubuntu (local o servidor)

```
antosierra@AntitoNotebook:~/ayudantia$ psql ayudantia -f consulta_1.sql
```

Comandos psql

1. Para crear base de datos

```
CREATE DATABASE nombre_db;
```

2. Para conectarse a una base de datos dentro de postgres

```
\c nombre_db
```

3. Para ver todas las bases de datos de tu equipo

```
\l
```

4. Para ver las tablas de la base de datos actual

```
\dt
```

Comandos psql

5. Para ver los atributos que recibe y las IC de las tablas

```
\d nombre_tabla
```

6. Para salir de postgres

```
\q
```

7. Para crear usuarios

```
CREATE USER nombre_usuario WITH PASSWORD 'contraseña';
```

9. Para visualizar los usuarios de tu equipo

```
\du
```

PHP

¿Qué es php?

- PHP (Hypertext Preprocessor) es un lenguaje de programación del lado del servidor diseñado para el desarrollo web.
- Es utilizado para crear páginas dinámicas e interactivas.
- Se ejecuta en el servidor y genera HTML para ser enviado al navegador.
- Facebook, Yahoo, Wikipedia, Spotify, etc.

¿Cómo ejecutarlo?

- Tener instalado php en el dispositivo.
- Para ejecutar archivos se ingresa por consola “php” junto al nombre del archivo, ejemplo “php archivo.php”

Recordar siempre usar el tag
<?php
?>

```php
<?php
echo "hola mundo\n";
echo "Escribe tu nombre: ";
echo readline();
?>
```

```
php archivo.php
hola mundo
Escribe tu nombre: mi_nombre
mi_nombre%
```

Variables y tipos de datos

Declaración de variable:
$variable

- String ($nombre = "Juan";)
- Integer ($edad = 25;)
- Float ($precio = 10.5;)
- Boolean ($es_valido = true;)
- Array ($frutas = ["Manzana", "Banana", "Naranja"];)

```php
<?php
$nombre = "Juan";
$edad = 25;
$precio = 10.5;
$es_valido = true;
$frutas = ["Manzana", "Banana", "Naranja"];

echo $nombre, "\n";
echo $edad, "\n";
echo $precio, "\n";
echo $es_valido, "\n";
print_r($frutas);
?>
```

Terminal ⟶

```
php ayudantia_2.php
Juan
25
10.5
1
Array
(
    [0] => Manzana
    [1] => Banana
    [2] => Naranja
)
```

PHP

```php
<?php
$nombre = "Juan";
$edad = 25;
$precio = 10.5;
$es_valido = true;
$frutas = ["Manzana", "Banana", "Naranja"];

echo $nombre, "\n";
echo $edad, "\n";
echo $precio, "\n";
echo $es_valido, "\n";
print_r($frutas);
?>
```

Python

```python
nombre = "Juan"
edad = 25
precio = 10.5
es_valido = True
frutas = ["Manzana", "Banana", "Naranja"]

print("Esto es un string: " + nombre)
print(edad)
print(precio)
print(es_valido)
print(frutas)
```

Operadores lógicos

- Verdadero: “true”
- Falso: “false”
- Igual: “x == y”
- Distinto: “x != y”
- Menor que / Mayor que: “ x < y”
- Menor o igual que / Mayor o igual que: “x >= y”
- Negación: “!x”
- Conjunción(Y): “&&”
- Disjunción(O): “||”

PHP

```php
<?php
$verdadero = true;
$falso = false;

$cond1 = 1 == 2;
$cond2 = "b" != "a";
$cond3 = 10 < 11;
$cond4 = 10 >= 11;
$cond5 = !true;
$cond6 = $verdadero && $falso;
$cond7 = $verdadero || $falso;
?>
```

Python

```python
verdadero = True
falso = False

cond1 = 1 == 2
cond2 = "b" != "a"
cond3 = 10 < 11
cond4 = 10 >= 11
cond5 = not True
cond6 = verdadero and falso
cond7 = verdadero or falso
```

Bucles

PHP

```php
<?php
$i = 0;
while ($i < 10) {
    echo $i;
    $i++;
}
?>
```

```php
<?php
for ($j = 0; $j < 10; $j++) {
    echo $j;
}
?>
```

Python

```python
i = 0
while i < 10:
    print(i)
    i += 1
```

```python
for j in range(10):
    print(j)
```

Listas

Más funciones de arrays: https://www.w3schools.com/php/php_arrays_functions.asp

- Se pueden instanciar listas de 2 formas:
  - $lista = array('a', 'b', 'c', 'd', 'e');
  - $lista = ['a', 'b', 'c', 'd', 'e'];
- Obtener elemento: $lista[indice]
- Modificar elemento lista: $lista[indice] = $elemento;
- Nuevo elemento:
  - $lista[] = $new_elem
  - array_push($lista, $new_elem)
- Largo: $largo = count($lista)
- Pop: $ultimo_elemento = array_pop($lista)
- Eliminar elemento: unset($lista[índice])
- Recorrer: foreach($lista as $elemento){iteración}

PHP

```php
<?php
//Definir una lista
$lista_con_array = array('a', 'b', 'c', 'd', 'e');
$lista_con_corchetes = ['a', 'b', 'c', 'd', 'e'];

//Obtener elemento
$elemento = $lista_con_array[0];

//Modificar elemento
$lista_con_array[0] = 'z';

//Nuevo elemento
$lista_con_array[] = 'final';
array_push($lista_con_array, "final2");
```

Python

```python
# Definir una lista
lista_con_array = ['a', 'b', 'c', 'd', 'e']

# Obtener elemento
elemento = lista_con_array[0]

# Modificar elemento
lista_con_array[0] = 'z'

# Nuevo elemento
lista_con_array.append('final')
lista_con_array.append('final2')
```

PHP

```php
//Largo lista
$largo = count($lista_con_array);

//Recorrer lista
foreach ($lista_con_array as $dato){
    echo $dato, "\n";}

// Método pop
$ultimo_elemento = array_pop($lista_con_array);
echo "Pop: ", $ultimo_elemento, "\n";

// Eliminar elemento en específico
unset($lista_con_array[2]);
```

Python

```python
# Largo lista
largo = len(lista_con_array)

# Recorrer lista
for dato in lista_con_array:
    print(dato)

# Método pop
ultimo_elemento = lista_con_array.pop()
print("Pop:", ultimo_elemento)

# Eliminar elemento en específico
lista_con_array.pop(2)
```

PHP

```php
// Listas de listas (matrices)
$matriz1 = array(
    array(1, 2, 3),
    array(4, 5, 6),
    array(7, 8, 9)
);
$matriz2 = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
];
```

Python

```python
# Listas de listas (matrices)
matriz1 = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]
matriz2 = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]
```

Strings

Más funciones de strings: https://www.w3schools.com/php/php_string_functions.asp

- Definir string: $string = 'Hola Mundo';
- Obtener carácter según índice: $string[índice]
- Concatenar strings: $string . ' :D', "\n" (puede ser con “.” o “,”)
- Largo string: $largo = strlen($string)
- Mayúsculas: strtoupper($string)
- Minúsculas: strtolower($string)
- Reemplazar: str_replace($fragmento_a_reemplazar, $texto_reemplazante, $string);
- Método split (convertir a lista): explode(",", $string);

PHP

```php
<?php
$string = 'Hola Mundo';
echo $string[6], "\n";
echo $string . ' :D', "\n";
echo strlen($string), "\n";
echo strpos($string, 'Mundo'), "\n";
echo strtoupper($string), "\n";
echo strtolower($string), "\n";
echo str_replace('Mundo', 'Planeta', $string), "\n";
$string = "Hola,Mundo";
$spliteado = explode(",", $string);
?>
```

Python

```python
string = 'Hola Mundo'
print(string[6])
print(string + ' :D')
print(len(string))
print(string.find('Mundo'))
print(string.upper())
print(string.lower())
print(string.replace('Mundo', 'Planeta'))
string = "Hola,Mundo"
spliteado = string.split(",")
```

Funciones

PHP

```php
<?php
function contar_caracteres($texto) {
    // Código de la función
    $largo = strlen($texto);
    return $largo;
}
$texto = "Hola mundo!";
$largo = contar_caracteres($texto);
echo $largo;
?>
```

Python

```python
def contar_caracteres(texto):
    # Código de la función
    largo = len(texto)
    return largo

texto = "Hola mundo!"
largo = contar_caracteres(texto)
print(largo)
```

Funciones filtro

filter_var($valor, $filtro,$opciones)

- Extensión incluida en PHP para validar y limpiar datos externos.
- Dos tipos:
  - Validar (FILTER_VALIDATE_*): devuelve el valor si es válido, o false si no.
  - Sanitizar (FILTER_SANITIZE_*): limpia el valor eliminando caracteres no deseados.

Hay varias funciones que pueden ser útiles para la limpieza de datos: https://www.w3schools.com/Php/php_ref_filter.asp

```php
<?php
$correo = "  ANA(@)Ejemplo.com  ";

// primero se limpia
$correo = strtolower(trim($correo));
$correo = filter_var($correo, FILTER_SANITIZE_EMAIL);

// despues se valida
if (filter_var($correo, FILTER_VALIDATE_EMAIL)) {
    echo "Correo válido: $correo";
} else {
    echo "Correo inválido: $correo";
}
?>
```

Archivos

- Abrir archivos
  - Lectura: $archivo = fopen("archivo.txt", "r");
  - Escritura (de cero): $archivo = fopen("archivo.txt", "w");
  - Agregar líneas: $archivo = fopen("archivo.txt", "a");
- Cerrar archivos: fclose($archivo);
- Escribir en archivos: fwrite($archivo, $string);
- Obtener líneas: fgets($archivo);
- Verificar final: feof($archivo);
- Aclaración: “fgets()” genera las líneas del archivo, al llamarla va a entregar la primera línea, la segunda, y así. La función “feof()” entrega un booleano dependiendo si se llegó al final del archivo (no puede generar más líneas con “fgets”)

PHP

```php
<?php
$archivo1 = fopen("archivo1.txt", "r"); //Abrir archivo en modo lectura

while (!feof($archivo1)) {      //feof es final del archivo
    $linea = fgets($archivo1);
    echo $linea;
}
fclose($archivo1); //Cerrar archivo
```

Python

```python
archivo1 = open("archivo1.txt", "r")  # Abrir archivo en modo lectura

for linea in archivo1:
    print(linea)

archivo1.close()  # Cerrar archivo
```

PHP

```php
$archivo2 = fopen("archivo2.txt", "w"); //Abrir archivo en modo escritura
fwrite($archivo2, 'sobreescribí!');
fclose($archivo2); //Cerrar archivo
```

```php
$archivo3 = fopen("archivo3.txt", "a"); //Abrir archivo en modo escritura al final del archivo
fwrite($archivo3, "\nme agregué!");
fclose($archivo3); //Cerrar archivo
```

Python

```python
archivo2 = open("archivo2.txt", "w")  # Abrir archivo en modo escritura
archivo2.write('sobreescribí!')
archivo2.close()  # Cerrar archivo
```

```python
archivo3 = open("archivo3.txt", "a")  # Abrir archivo en modo escritura al final del archivo
archivo3.write("\nme agregué!")
archivo3.close()  # Cerrar archivo
```

EJERCICIOS

EJERCICIOS

Contexto

Te entregan un archivo datos.csv con datos de personas. Los datos vienen sucios: hay correos mal escritos, edades que no son números y espacios o mayúsculas de más. Tu programa debe separar las filas buenas de las malas y dejar registro de decisiones tomadas.

```
id,nombre,correo,edad
1,Ana,ana@mail.com,25
2,Luis,luis@,30
3,Pedro,pedro@mail.com,abc
4,María,  MARIA@MAIL..COM ,22
```

EJERCICIOS

Pasos

1. Recorrer archivo de datos
2. Para cada fila:
    a. Limpiar el correo
    b. Validar el correo
    c. Validar la edad
    d. Identificar acción a tomar (¿eliminamos, ajustamos, corregimos, o ninguna?)
    e. Guardar en ‘datosok.csv’ si los datos son válidos
    f. Guardar en ‘datoslog.csv’ los logs de decisiones

RECOMENDACIONES ENTREGA

- Entrega hasta 16 de octubre 20:00 hrs
- Cuidado con usar extensiones de php que no estén habilitadas en el servidor (como ‘mbstring’ y 'mb_strlen'). Si las usan y no están, no se instalan simplemente se asume que la entrega no corre.
- Prueben que la entrega funciona en el servidor.
- Lean el enunciado con tiempo y dejen sus dudas en issues (hasta 9 de octubre).