

# Programación Python para Big Data - Tarea lección 4

Kevin Martínez García

8 de junio de 2022

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Creación de las tablas con Adminer</b>	<b>2</b>
<b>3. Ejercicio 1</b>	<b>4</b>
<b>4. Ejercicio 2</b>	<b>5</b>
<b>5. Ejercicio 3</b>	<b>5</b>
<b>6. Ejercicio 4</b>	<b>6</b>
<b>7. Ejercicio 5</b>	<b>6</b>
<b>8. Ejercicio 6</b>	<b>7</b>

## 1. Introducción

En la actividad relacionada a la lección 4 se nos solicitó hacer uso de *docker-compose* y *PostgreSQL* para crear y realizar diferentes operaciones sobre dos tablas en una base de datos. A continuación explicaremos cada uno de los ejercicios solicitados junto con capturas de pantalla para poder verificar los resultados.

## 2. Creación de las tablas con Adminer

Para empezar, se nos solicitó usar Adminer para crear una base de datos *actividad* que contuviese dos tablas, *notas* y *edición*. Comenzamos creando la base de datos actividad como aparece en la Figura 1 a continuación.

The screenshot shows the Adminer 4.8.1 web interface. At the top, there's a language dropdown set to 'Español' and a breadcrumb 'PostgreSQL » db » Crear Base de datos'. Below this, a large blue button says 'Crear Base de datos'. On the left, there's a 'DB:' dropdown menu. Below that are links for 'Comando SQL', 'Importar', and 'Exportar'. On the right, there's a text input field containing 'actividad' and a 'Guardar' button with a plus icon.

Figura 1: Creación de la base de datos actividad

Seguidamente, procedemos a generar sus dos respectivas tablas. En primer lugar, creamos la tabla edición con las siguientes columnas.

- *id\_edic*: variable de tipo *integer* y *auto incrementable* que funcionará como clave primaria de la tabla edición.
- *numero*: variable de tipo *varchar* que nos indica el nombre de la edición.

A continuación en la Figura 2 vemos la creación de edición haciendo uso de Adminer.

The screenshot shows the Adminer interface for creating a table named 'edicion'. At the top, there's a 'Nombre de la tabla:' field with 'edicion' and a 'Guardar' button. Below this is a table with columns: 'Nombre de columna', 'Tipo', 'Longitud', 'Opciones', 'NULL', 'AI?', and '+'. The first row has 'id\_edic' as the column name, 'integer' as the type, and the 'AI?' checkbox is checked. The second row has 'numero' as the column name, 'character vary' as the type, and the 'AI?' checkbox is unchecked. Below the table, there's an 'Incremento automático:' field, a checkbox for 'Valores predeterminados', and a 'Comentario' field. At the bottom are 'Guardar' and 'Eliminar' buttons.

Nombre de columna	Tipo	Longitud	Opciones	NULL	AI?	+
id_edic	integer			<input type="checkbox"/>	<input checked="" type="radio"/>	<input checked="" type="checkbox"/>
numero	character vary			<input type="checkbox"/>	<input type="radio"/>	<input checked="" type="checkbox"/>

Figura 2: Creación de la tabla edición

Seguidamente procedimos a crear la tabla notas con las siguientes columnas.

- *id\_notas*: variable de tipo *interger* y *auto incrementable* que funcionará como clave primaria de la tabla notas.

- *name*: variable de tipo *varchar* que nos indica el nombre de cada alumno.
- *edad*: variable de tipo *integer* que nos indica la edad de cada alumno.
- *notas*: variable de tipo *numeric* que nos indica la nota de cada alumno.
- *id\_edic*: variable de tipo *integer* que nos relaciona cada alumno con una edición.

En la Figura 3 podemos ver la creación de esta tabla con Adminer.

Nombre de la tabla:

Nombre de columna	Tipo	Longitud	Opciones	NULL	<input type="radio"/> AI?	+
<input type="text" value="id_notas"/>	<input type="text" value="integer"/>	<input type="text"/>		<input type="checkbox"/>	<input checked="" type="radio"/>	<input type="button" value="x"/>
<input type="text" value="name"/>	<input type="text" value="character vary"/>	<input type="text" value="80"/>		<input type="checkbox"/>	<input type="radio"/>	<input type="button" value="x"/>
<input type="text" value="edad"/>	<input type="text" value="integer"/>	<input type="text"/>		<input type="checkbox"/>	<input type="radio"/>	<input type="button" value="x"/>
<input type="text" value="notas"/>	<input type="text" value="numeric"/>	<input type="text"/>		<input type="checkbox"/>	<input type="radio"/>	<input type="button" value="x"/>
<input type="text" value="id_edic"/>	<input type="text" value="integer"/>	<input type="text"/>		<input type="checkbox"/>	<input type="radio"/>	<input type="button" value="x"/>

Incremento automático:  ☐ Valores predeterminados ☐

Comentario

Figura 3: Creación de la tabla notas

Una vez creadas las tablas quedaba establecer la relación entre ambas. Para ello agregamos una clave externa a la tabla notas para que su columna *id\_edic* señalase a la clave primaria de la tabla edición. Adjunto en la Figura 4 una captura de las tabla notas con la clave externa ya agregada.

Tabla: notas

[Visualizar contenido](#) [Mostrar estructura](#) [Modificar tabla](#) [Nuevo Registro](#)

Columna	Tipo	Comentario
<b>id_notas</b>	integer Incremento automático [nextval("notas_ID_notas_seq")]	
<b>name</b>	character varying(80)	
<b>edad</b>	integer	
<b>notas</b>	numeric	
<b>id_edic</b>	integer	

Índices

<b>PRIMARY</b>	<b>id_notas</b>
----------------	-----------------

[Modificar índices](#)

Claves externas

Origen	Destino	AL BORRAR	AL ACTUALIZAR	
<b>id_edic</b>	edicion(id_edic)	NO ACTION	NO ACTION	<a href="#">Modificar</a>

[Agregar clave externa](#)

Disparadores

[Agregar disparador](#)

Figura 4: Tablas con la clave externa ya agregada

Una vez creada la base de datos, sus respectivas tablas y la relación entre ellas, pudimos proceder a los diferentes ejercicios.

### 3. Ejercicio 1

En este primer ejercicio se nos solicitaba poblar las bases de datos con los datos proporcionados en el manual de la actividad. Para ello, se hizo uso de la librería `psycopg2` tal y como vimos en las sesiones de teoría (por extensión no adjuntaré el código en este manual, lo adjuntaré en la entrega). Teniendo en cuenta que la tabla `notas` tiene una clave foránea con la tabla `edición`, se debía poblar en primer lugar esta última tabla. El resultado de este proceso es el que aparece en la Figura 5 a continuación.

Mostrar: edicion

Visualizar contenido   Mostrar estructura   Modificar tabla   Nuevo Registro

Mostrar   Condición   Ordenar   Limite: 50   Longitud de texto: 100   Acción: Mostrar

SELECT \* FROM "edicion" LIMIT 50 (0.001 s) Modificar

<input type="checkbox"/> Modify	id_edic	numero
<input type="checkbox"/> modificar	1	Uno
<input type="checkbox"/> modificar	2	Dos
<input type="checkbox"/> modificar	3	Tres

Resultado completo: ☐ 3 registros   Modified:    Selected (0):      Exportar (3)

Importar

Figura 5: Contenido de la tabla edición

Una vez hecho esto, pudimos seguir un procedimiento similar y poblar la tabla `notas`. El resultado es el que aparece en la Figura 6.

Mostrar: notas

Visualizar contenido   Mostrar estructura   Modificar tabla   Nuevo Registro

Mostrar   Condición   Ordenar   Limite: 50   Longitud de texto: 100   Acción: Mostrar

SELECT \* FROM "notas" LIMIT 50 (0.000 s) Modificar

<input type="checkbox"/> Modify	id_notas	name	edad	notas	id_edic
<input type="checkbox"/> modificar	1	Isabel Maniega	30	5.6	1
<input type="checkbox"/> modificar	2	José Manuel Peña	30	7.8	1
<input type="checkbox"/> modificar	3	Pedro López	30	5.2	2
<input type="checkbox"/> modificar	4	Julia García	22	7.3	1
<input type="checkbox"/> modificar	5	Amparo Mayora	28	8.4	3
<input type="checkbox"/> modificar	6	Juan Martínez	30	6.8	3
<input type="checkbox"/> modificar	7	Fernando López	35	6.1	2
<input type="checkbox"/> modificar	8	María Castro	41	5.9	3

Resultado completo: ☐ 8 registros   Modified:    Selected (0):      Exportar (8)

Importar

Figura 6: Contenido de la tabla notas

## 4. Ejercicio 2

Para este segundo ejercicio, se nos solicitaba actualizar la columna notas para dos alumnos en concreto. Para ello se hizo uso de la sentencia `UPDATE` de `sql` sobre los alumnos Pedro López y María Castro. En la Figura 7 podemos ver como ambas notas han quedado actualizadas con la puntuación solicitada.

**Mostrar: notas**

[Visualizar contenido](#) [Mostrar estructura](#) [Modificar tabla](#) [Nuevo Registro](#)

Limite

Longitud de texto

Acción

`SELECT * FROM "notas" LIMIT 50 (0.000 s)` [Modificar](#)

<input type="checkbox"/> Modify	id_notas	name	edad	notas	id_edic
<input type="checkbox"/> <a href="#">modificar</a>	1	Isabel Maniega	30	5.6	<a href="#">1</a>
<input type="checkbox"/> <a href="#">modificar</a>	2	José Manuel Peña	30	7.8	<a href="#">1</a>
<input type="checkbox"/> <a href="#">modificar</a>	4	Julia García	22	7.3	<a href="#">1</a>
<input type="checkbox"/> <a href="#">modificar</a>	5	Amparo Mayora	28	8.4	<a href="#">3</a>
<input type="checkbox"/> <a href="#">modificar</a>	6	Juan Martinez	30	6.8	<a href="#">3</a>
<input type="checkbox"/> <a href="#">modificar</a>	7	Fernando López	35	6.1	<a href="#">2</a>
<input type="checkbox"/> <a href="#">modificar</a>	3	Pedro López	30	6.4	<a href="#">2</a>
<input type="checkbox"/> <a href="#">modificar</a>	8	María Castro	41	5.2	<a href="#">3</a>

Resultado completo

☐ 8 registros

Modify

Selected (0)

Exportar (8)

[Importar](#)

Figura 7: Tabla notas actualizada

## 5. Ejercicio 3

En esta tercera actividad hicimos una lectura de todos los datos de la tabla notas. Para ello, usamos un `cursor` para ejecutar la instrucción `SELECT * FROM notas` y un bucle `for` para mostrar por la pantalla del terminal todas las filas recuperadas. El resultado es el que puede verse en la Figura 8.

```
(base) kevin@kevin-ubuntu:~/Documentos/MasterPython/9.-ProgramacionBigData/Leccion4$ python3 script.py
(1, 'Isabel Maniega', 30, Decimal('5.6'), 1)
(2, 'José Manuel Peña', 30, Decimal('7.8'), 1)
(4, 'Julia García', 22, Decimal('7.3'), 1)
(5, 'Amparo Mayora', 28, Decimal('8.4'), 3)
(6, 'Juan Martínez', 30, Decimal('6.8'), 3)
(7, 'Fernando López', 35, Decimal('6.1'), 2)
(3, 'Pedro López', 30, Decimal('6.4'), 2)
(8, 'María Castro', 41, Decimal('5.2'), 3)
```

Figura 8: Lectura de la tabla notas

## 6. Ejercicio 4

En la cuarta actividad se pedía hacer una lectura de aquellas filas cuya columna notas tuviese un valor entre 5 y 6.5. Para ello, hicimos uso de la sentencia **BETWEEN** para recuperar las notas contenidas en ese intervalo. El resultado por terminal es el que puede verse en la Figura 9.

```
(base) kevin@kevin-ubuntu:~/Documentos/MasterPython/9.-ProgramacionBigData/Leccion4$ python3 script.py
(1, 'Isabel Maniega', 30, Decimal('5.6'), 1)
(7, 'Fernando López', 35, Decimal('6.1'), 2)
(3, 'Pedro López', 30, Decimal('6.4'), 2)
(8, 'María Castro', 41, Decimal('5.2'), 3)
```

Figura 9: Alumnos con notas entre 5 y 6.5

Esta misma instrucción podía ejecutarse directamente desde Adminer para obtener el resultado que aparece en la Figura 10.

Comando SQL

SELECT \* FROM notas WHERE notas BETWEEN 5.0 AND 6.5

id_notas	name	edad	notas	id_edic
1	Isabel Maniega	30	5.6	1
7	Fernando López	35	6.1	2
3	Pedro López	30	6.4	2
8	María Castro	41	5.2	3

4 registros (0.001 s) [Modificar](#), [Explain](#), [Exportar](#)

SELECT \* FROM notas WHERE notas BETWEEN 5.0 AND 6.5

Ejecutar Limit rows:  ☐ Parar en caso de error ☐ Mostrar solamente errores

Figura 10: Alumnos con notas entre 5 y 6.5 desde Adminer

## 7. Ejercicio 5

En este ejercicio se nos solicitó recuperar la información de aquellos alumnos pertenecientes a la edición “Dos”. Esta actividad supone entender la diferencia entre la clave primaria *id\_edic* de la tabla edición y la columna *número*. En este caso la clave 2 coincide con la edición número “Dos”, sin embargo, podría no ser el caso siempre. Por tanto, lo que esto supone es que no podemos quedarnos con aquellos alumnos cuya columna *id\_edic* valga 2, sino que debemos relacionar ambas tablas. Los resultados del proceso por terminal son los que aparecen en la Figura 11.

```
(base) kevin@kevin-ubuntu:~/Documentos/MasterPython/9.-ProgramacionBigData/Leccion4$ python3 script.py
(7, 'Fernando López', 35, Decimal('6.1'), 2)
(3, 'Pedro López', 30, Decimal('6.4'), 2)
```

Figura 11: Alumnos de la edición Dos

Esta misma instrucción podía ejecutarse directamente desde Adminer para obtener el resultado que aparece en la Figura 12.

Comando SQL

```
SELECT *
FROM notas
WHERE id_edic IN (SELECT id_edic
FROM edicion
WHERE numero = 'Dos')
```

id_notas	name	edad	notas	id_edic
7	Fernando López	35	6.1	2
3	Pedro López	30	6.4	2

2 registros (0.002 s) [Modificar](#) [Explain](#) [Exportar](#)

```
SELECT *
FROM notas
WHERE id_edic IN (SELECT id_edic
FROM edicion
WHERE numero = 'Dos')
```

Ejecutar

Limit rows:

☐ Parar en caso de error

☐ Mostrar solamente errores

Figura 12: Alumnos de la edición Dos desde Adminer

## 8. Ejercicio 6

Para esta última actividad realizamos un borrado de los datos del alumno Pedro López haciendo uso de la sentencia DELETE. Los resultados son lo que pueden verse en la Figura 13 dónde vemos que en la tabla notas, ya no aparece este mismo alumno.

Mostrar: notas

Visualizar contenido

Mostrar estructura

Modificar tabla

Nuevo Registro

Mostrar

Condición

Ordenar

Limite

50

Longitud de texto

100

Acción

Mostrar

```
SELECT * FROM "notas" LIMIT 50 (0.000 s) Modificar
```

<input type="checkbox"/> Modify	id_notas	name	edad	notas	id_edic
<input type="checkbox"/> modificar	1	Isabel Maniega	30	5.6	1
<input type="checkbox"/> modificar	2	José Manuel Peña	30	7.8	1
<input type="checkbox"/> modificar	4	Julia García	22	7.3	1
<input type="checkbox"/> modificar	5	Amparo Mayora	28	8.4	3
<input type="checkbox"/> modificar	6	Juan Martínez	30	6.8	3
<input type="checkbox"/> modificar	7	Fernando López	35	6.1	2
<input type="checkbox"/> modificar	8	Maria Castro	41	5.2	3

Resultado completo

☐ 7 registros

Modify

Guardar

Selected (0)

Modificar

Clonar

Eliminar

Exportar (7)

[Importar](#)

Figura 13: Borrado del alumno Pedro López