(Acceso programático a Bases de Datos desde Python)

Laboratorio de bases de datos.
Grado en Ing. en Telemática, 2024-2025.
EIF.
Universidad Rey Juan Carlos.

---

1. Objetivos

- Realizar conexiones a bases de datos PostgreSQL desde Python.
- Implementar comandos DDL y DML en Python.

2. Acceso a Base de Datos *bpsimple* desde Python.

1. Elabora un script en Python "bpsimple.py"que gestione la conexión a la base de datos *bpsimple*. El Script deberá permitir la conexión a la base de datos mediante el uso de un archivo de configuración çonfig.ini", gestionar posibles errores de conexión y terminar cerrando la conexión.

   A partir del script creado para garantizar la conexión a la base de datos bpsimple, realiza las siguientes acciones:

2. Crea una nueva tabla llamada .employees"que almacenará información sobre los empleados de la empresa. La tabla debe tener los siguientes campos: employee_id (serial, clave primaria), name(cadena de caracteres de hasta 50 caracteres), last_name (cadena de caracteres de hasta 50 caracteres), start_date (date), salary(numeric(8,2)).

3. Introduce información para tres empleados, proporcionando valores para los campos name, last_name, start_date, y salary:

   - Empleado 1:
     - Nombre: William
     - Apellidos: Turner
     - Fecha de Incorporación: 2003-01-15
     - Salario: 40000.00
   - Empleado 2:
     - Nombre: Charlotte
     - Apellidos: Anderson
     - Fecha de Incorporación: 2003-02-01
     - Salario: 50000.00
   - Empleado 3:
     - Nombre: James
     - Apellidos: Parker
     - Fecha de Incorporación: 2003-03-10
     - Salario: 40000.00

4. A continuación, comprueba que se ha generado correctamente la tabla .employeesz que se han introducido correctamente todos los datos generando una consulta que devuelva toda la información de la tabla .employees".

5. Al añadir empleados a la base de datos, se desea registrar que empleado gestionó cada orden de pedido .ºrderinfo". Para ello, modifica la tabla .ºrderinfo.ªñadiendo una nuevo columna con el campo .employee_id"que será un entero que represente el identificador del empleado que gestionó la orden de pedido.

6. Inserta la información de los empleados encargados de gestionar cada pedido en la tabla .ºrderinfo". Los datos son los siguientes:

   - Orderinfo 1: Empleado 2.
   - Orderinfo 2: Empleado 3.
   - Orderinfo 3: Empleado 3.
   - Orderinfo 4: Empleado 1.
   - Orderinfo 5: Empleado 3.

7. A continuación, comprueba que se ha actualizado correctamente la tabla .ºrderinfo". Para ello, genera una consulta que recoga el identificador de orderinfo (.ºrderinfo_id"), el identificador del cliente que realizó el pedido (çustomer_id"), el nombre y apellido del cliente ("fname", "lname") y, por último, el identificador, el nombre y el apellido del empleado (.employee_id", "name", "last_name") que gestionó el pedido.

8. Crea una vista "Stock_Productos"que recoga la siguiente información: el identificador del item (ïtem_id"), la descripción del item ("description") y la cantidad de stock de ese item ("quantity"de la tabla stock) incluso de aquellos items de los que no hay stock.

9. Muestra la vista que se acaba de crear con todos los registros que posea actualmente.

10. Debido a un defecto de fábrica se deben retirar todos los productos del item_id 4 del stock. Por tanto, lleva a cabo la eliminación de este registro de la tabla stock. Tras borrar el registro correspondiente, comprueba que todo es correcto utilizando la vista creada en el apartado 8.

3. Acceso a Base de Datos *Universidad* desde Python.

1. Elabora un script en Python üniversidad.py"que gestione la conexión a la base de datos *universidad*. El Script deberá permitir la conexión a la base de datos mediante el uso de un archivo de configuración çonfig.ini", gestionar posibles errores de conexión y terminar cerrando la conexión.

   A partir del script creado para garantizar la conexión a la base de datos universidad, realiza las siguientes acciones:

2. Crea una nueva tabla llamada "publicacion"que almacenará información sobre las publicaciones de la universidad. La tabla debe tener los siguientes campos: id_publicación (serial, clave primaria), título(cadena de caracteres de hasta 50 caracteres), tipo_publicacion (cadena de caracteres de hasta 50 caracteres, en principio Revista o Congreso), Fecha_Publicación (date), Autores (integer) y coste (numeric(8,2)).

3. Introduce información para cinco publicaciones, proporcionando valores para los campos título, tipo_publicación, Fecha_publicacion, autores y coste:

   - Publicación 1:
     - Título: Avances en Medicina Preventiva
     - Tipo de Publicación: Revista
     - Fecha de Publicación: 2023-01-15
     - Autores: 3
     - Coste: 1500.00 €
   - Publicación 2:

- Título: Desarrollos en Inteligencia Artificial
- Tipo de Publicación: Congreso
- Fecha de Publicación: 2023-02-01
- Autores: 5
- Coste: 500.00 €

- Publicación 3:
  - Título: Estudios Sociológicos
  - Tipo de Publicación: Revista
  - Fecha de Publicación: 2023-03-10
  - Autores: 4
  - Coste: 1800.00 €
- Publicación 4:
  - Título: Innovaciones en Ingeniería de Software
  - Tipo de Publicación: Congreso
  - Fecha de Publicación: 2023-04-05
  - Autores: 5
  - Coste: 420.00 €
- Publicación 5:
  - Título: Avances en Neurociencia
  - Tipo de Publicación: Revista
  - Fecha de Publicación: 2023-04-01
  - Autores: 5
  - Coste: 2000.00 €

4. A continuación, comprueba que se ha generado correctamente la tabla "publicacionz que se han introducido correctamente todos los datos generando una consulta que devuelva toda la información de la tabla "publicacion".

5. Dado que el campo Autores puede ser poco preciso, genera una moddificación en la tabla "publicacion"que convierta el nombre del campo .ªutores.en "num_autores".

6. Se quiere actualizar la base de datos para poder tener una relación entre los profesores y las publicaciones. Cómo esta relación es de cardinalidad N:N, es decir, una publicación puede estar hecha por varios profesores y, cada profesor puede tener varias publicaciones, es necesario crear una nueva tabla que relaciona Publicacion-Profesor. Por todo ello, crea, a través de Python, una nueva tabla llamada "Publicaciones_Profesores"que tenga como campos Id_publicacion(INTEGER y clave primaria) y un idprofesor(INTEGER y clave Primaria). No es necesario especificar que son claves foráneas, simplemente crear los dos campos y, en la sentencia de `CREATE TABLE...` añadir al final la siguiente instrucción `PRIMARY KEY (id_publicacion, idprofesor)`.

7. Realiza la inserción de datos en la tabla considerando que las publicaciones tienen los siguientes autores, recuerda que solo necesitas insertar el id de la publicación y el id del profesor:

   - **ID_Publicacion 1 (Avances en Medicina Preventiva):**
     - Profesores Asociados:
       - Luis Martinez (IDProfesor: 2003)
       - Juan Lopez (IDProfesor: 2001)
       - Maria Garcia (IDProfesor: 2002)
   - **ID_Publicacion 2 (Desarrollos en Inteligencia Artificial):**
     - Profesores Asociados:
       - Ana Rodriguez (IDProfesor: 2004)

        - Miguel Perez (IDProfesor: 2007)
    - **ID_Publicacion 3 (Estudios Sociológicos):**
        - Profesores Asociados:
            - Maria Garcia (IDProfesor: 2002)
            - Carlos Fernandez (IDProfesor: 2005)
            - Isabel Sanchez (IDProfesor: 2008)

8. Para comprobar que todo es correcto, realiza una consulta a través de python que muestre el nombre del departamento, el nombre y apellido del profesor junto con el título de la publicación y su fecha de publicación.

9. Se desea actualizar la tabla de estudiantes para eliminar la columna .edad"de todos los estudiantes.

10. Una vez eliminada la columna edad, generar una nueva columna en la tabla .estudiantesçon el nombre "Fecha_Nacimiento". Creada la columna actualiza a todos los estudiantes con las siguientes fechas de nacimiento:

    - **Matrícula 1001:** 1998-03-12
    - **Matrícula 1002:** 2000-07-22
    - **Matrícula 1003:** 2005-02-10
    - **Matrícula 1004:** 2003-11-05
    - **Matrícula 1005:** 1998-09-30
    - **Matrícula 1006:** 2002-01-18
    - **Matrícula 1007:** 2002-04-07
    - **Matrícula 1008:** 2001-06-14
    - **Matrícula 1009:** 2004-08-23
    - **Matrícula 1010:** 2003-10-11

11. Para comprobar que la actualziación de esta tabla ha sido satisfactoria, genera una consulta que devuelva la matricula, nombre, apellido y edad (pero calculada a partir de la fecha de nacimiento), de cada estudiante. (Muestra la edad en años).