1. Consultas básicas

1. Cree un Jupyter notebook, el cual servirá como documento de trabajo y entrega. Incluya todas las respuestas a las consultas solicitadas en el notebook.

2. Conéctese al mismo servidor usado en las diapositivas, pero en este caso a la base de datos **taller_uni**.

3. Liste todas las tablas en el esquema público.

4. Liste el nombre de todas las unidades académicas.

5. Liste los nombres de todas las unidades académicas a las que están asociados los profesores. Elimine duplicados con `distinct` antes del atributo de interés (nombre de la unidad).

6. Para cada tabla liste 5 registros usando `limit` al final de la consulta.

7. Realice una consulta que retorne un listado de los instructores con su nombre, ID y el doble de su salario.

8. Realice una consulta para obtener todos los cursos ofrecidos por la unidad 'IIND'.

9. Liste todos los nombres de los instructores con el edificio donde queda ubicada su respectiva unidad. Use un cross join.

10. Liste todos los instructores con sus clases. Use un cross join.

11. Liste todos los instructores con sus clases. Use un natural join.

12. Liste todos los IDs de los instructores con los códigos de los cursos que dictan. Use un natural join.

    Alternativamente se puede usar `join using` para hacer un join que use solo algunos atributos comunes, no todos. Por ejemplo, para listar todos los IDs de los instructores con los códigos de los cursos que dictan podemos ejecutar el siguiente comando

    ```
    select inst_ID, curso_cod
    from (instructor natural join dicta)
    join curso using (curso_cod);
    ```

    La cláusula `as` se puede utilizar para renombrar los atributos en la relación que se obtiene como resultado. Por ejemplo, en el siguiente comando cambiamos el nombre del primer atributo seleccionado

    ```
    select nombre as nombre_inst, curso_cod
    from instructor, dicta
    where instructor.inst_ID = dicta.inst_ID;
    ```

    La cláusula `as` también se puede utilizar para renombrar las relaciones consideradas. Por ejemplo, el comando anterior lo podríamos escribir como

```
select nombre, curso_cod
from instructor as I, dicta as D
where I.inst_ID = D.inst_ID;
```

Esto puede ser útil cuando se requiere comparar una relación consigo misma. Por ejemplo, si se quiere determinar los cursos que tienen al menos tantos créditos como al menos un curso de IIND, ejecutaríamos el comando

```
select distinct A.nombre
from curso as A, curso as B
where A.creditos > B.creditos  and B.nombre_unid = 'IIND';
```

Estos nombres para las relaciones se conocen como alias.

13. Realice una consulta en la que compare una relación consigo misma (diferente a curso).

2. Más consultas

1. Liste todos los cursos con una 'a' en su nombre.
2. Liste todos los instructores cuya unidad empieza con 'I' de ingeniería.
   a) Para estos instructores liste los códigos de todos los cursos que dictaron.
   b) Para estos instructores liste los códigos y nombres de todos los cursos que dictaron, así como el año y semestre.
3. Determine el número promedio de créditos en los cursos ofrecidos por 'IIND'.
4. Liste el número promedio créditos en los cursos ofrecidos por cada unidad.
   a) Ordene sus resultados por el nombre de la unidad.
   b) Filtre sus resultados para incluir solamente las unidades que ofrezcan curso con un promedio de máximo 4 créditos.
5. Liste el salario promedio de los profesores de cada unidad académica, así como el salario mínimo y máximo. Su relación debe tener 4 atributos con nombres 'nombre_unid', 'prom_salario', 'min_salario', 'max_salario', en ese orden.

3. Otra base de datos: world

1. Conéctese al mismo endpoint de las secciones anteriores, pero a la base de datos world.
2. Esta base tiene 3 tablas: city, country y city_language. Con esta información realicemos algunas consultas. Revisemos por ejemplos las primeras 5 filas de la tabla city

```
query = "select␣*␣from␣city␣limit␣5"
cursor.execute(query)
result = cursor.fetchall()
result
```

Note que el resultado se almacena en result al ejecutar fetchall(). Una forma alternativa de imprimir cada fila/registro resultado de la consulta es la siguiente

```
for row in result:
    print(row)
```

Si en algún momento se genera un error, la conexión queda bloqueada con una transacción que no terminó. Para descartar la transacción actual y volver a intentar ejecute un commit sobre la conexión.

```
engine.commit()
```

3. Cuente ahora el número de registros en cada tabla.
4. Usando la tabla country, determine los continentes que aparecen en la tabla.
5. Liste los nombres y códigos de los países en el continente Europe.
6. Determine los 5 países con mayor población de Europa.
7. Determine la población promedio de los países de Europa.
8. Liste los países que tienen una población superior al promedio del continente. Para cada país incluya nombre y población.
9. Determine los tipos de gobierno existentes en Europa.
10. Determine los países europeos que tienen un tipo de gobierno entre los tres más comunes en el continente. Para cada país incluya nombre y población.
11. Liste cada país con su capital, población total, población de la capital y fracción de población que vive en la capital. Reporte el top 20 de los países con mayor y menor fracción de población en la capital. Por motivos de legibilidad, se espera que la fracción se muestre como un número con 4 posiciones decimales. Hint: considere el uso de CAST AS y ROUND.
12. Genere una lista de los países que hablan inglés como idioma oficial. Para cada país cuyo idioma oficial es el inglés, liste la fracción de la población que habla inglés y ordene de acuerdo de menor a mayor.
13. Determine los 5 idiomas más hablados por número de países y por población total que lo habla.