



ESCALA PARA EL PROYECTO DJANGO-2 - SQL (/PROJECTS/DJANGO-2-SQL)

Debes evaluar a 1 estudiante en este equipo.



repositorio Git

git@vogsphere.42urduliz.com:vogsphere/intra-uuid-90857b0d-572d-4e07-



Introducción

Para el correcto desarrollo de esta evaluación, respete las siguientes reglas:

- Manténgase educado, amable, respetuoso y constructivo durante toda la conversación. Es una cuestión de confianza entre usted y la comunidad de 42.
- Resalte los posibles problemas que haya tenido con el trabajo presentado a la persona o grupo que está calificando y tómese el tiempo para hablar y debatir sobre ellos.
- Acepte que el tema del examen o las funciones requeridas pueden dar lugar a diferentes interpretaciones. Escuche la perspectiva de su compañero de discusión con una mente abierta (¿tiene razón o no?) y califíquela de la manera más justa posible. Los métodos de enseñanza de 42 solo tienen sentido si la evaluación entre pares se toma en serio.

Pautas

- Solo debes evaluar lo que encontrarás en el repositorio GiT del estudiante o grupo.
- Tómate el tiempo para verificar que el repositorio GiT coincida con el estudiante o grupo y el proyecto.
- Verifica dos veces que no se haya usado un alias malicioso para engañarte y hacerte calificar algo diferente del contenido oficial del repositorio.
- Si una secuencia de comandos que se supone que ayuda a evaluar el examen es proporcionada por cualquiera de las partes, la otra parte tendrá que comprobarlo estrictamente para evitar sorpresas desagradables.
- Si el estudiante que evalúa aún no ha realizado este proyecto, tendrá que leer el tema del examen en su totalidad antes de comenzar la evaluación.
- Usa las banderas disponibles en este sistema de calificación para señalar un proyecto vacío o no funcional, un fallo en la norma, trampas, etc. En ese caso, la evaluación se detiene y la calificación final es 0 (o -42 si es un problema de trampas). Sin embargo, si no se trata de un problema de trampas, estás invitado a seguir hablando sobre el trabajo que se ha hecho (o no hecho, de hecho) para identificar los problemas que llevan a este estancamiento y evitarlo la próxima vez.

Archivos adjuntos

tema.pdf (https://cdn.intra.42.fr/pdf/pdf/96068/en.subject.pdf)

d05.tar.gz (https://cdn.intra.42.fr/document/document/18976/d05.tar.gz)

Prefacio

descripción

Observando las reglas

- El repositorio contiene el trabajo del estudiante o grupo evaluado.
- El estudiante o grupo evaluado podrá explicar su trabajo en cualquier momento durante la evaluación.
- Durante toda la evaluación se observarán las instrucciones generales y específicas del día.
- Para este proyecto, debes clonar su repositorio Git en su estación.



Si



No

Entrenamiento de Python-Django - 2 - ORM

- Hard code the result of an exercise means this exercise get a 0. - You must test the exercises with Python 3. If an exercise doesn't work with Python3, even if it works with Python2, it's considered invalid. - Start recreating a brand new database: `psql -c "drop database formationdjango;"` `psql -c "create database formationdjango;"` OR (if it works on osx) `dropdb formationdjango createdb formationdjango` - If some there are some migration folders left in the app, delete them: `rm -rf */migrations` (from the project's root) - Every instruction requiring an URL consider the project's development server of the exercise is running thanks to the `./manage.py runserver` command. - You must run the tests following the indicated order.

Exercise 00 - Building a table in SQL

- Enter this command in the terminal:

`psql -U djangouser -d formationdjango -c "\d ex00_movies"`

The output must be: Did not find any relation named "ex00_movies".

Start the Django development server.

- Enter the url 127.0.0.1:8000/ex00/init in the brower:

You must get a page simply indicating "OK"

- Enter this command in the terminal again:

`psql -U djangouser -d formationdjango -c "\d ex00_movies"`

The output should be:

Column | Type | Modifiers

```
-----+-----+-----
episode_nb | integer | not null title
| character varying(64) | not null opening_crawl | text
| director | character varying(32) | not null
producer | character varying(128) | not null
release_date | date | not null Indexes:
"ex00_movies_pkey" PRIMARY KEY, btree (episode_nb)
"ex00_movies_title_key" UNIQUE CONSTRAINT, btree (title)
```

- Once again, enter the URL: 127.0.0.1:8000/ex00/init The page should still display "OK".



Yes



No

Exercise 01 - Build a table in the ORM

- Enter this command in the terminal:

`psql -U djangouser -d formationdjango -c "\d ex01_movies"`

The output should be: Did not find any relation named "ex01_movies".

- Ask the student to create the migration for the application ex01 only.
- Ask them to apply the migration.
- Enter the command in the terminal once again:

`psql -U djangouser -d formationdjango -c "\d ex01_movies"`

The output must be:

Column	Type	Modifiers
-----+-----+----- title		
character	varying(64)	not null episode_nb
integer	not null opening_crawl	text
director	character varying(32)	not null
producer	character varying(128)	not null
release_date	date	not null Indexes:
"ex01_movies_pkey"	PRIMARY KEY, btree (episode_nb)	
"ex01_movies_title_key"	UNIQUE CONSTRAINT, btree (title)	
"ex01_movies_title_e9be860e_like"	btree (title	
varchar_pattern		

☒ Yes

☐ No

Exercise 02 - Inserting SQL data

- test the urls: 127.0.0.1:8000/ex02/display and 127.0.0.1:8000/ex02/populate "No data available" or error messages indicating that the ex02_movies table doesn't exist must be displayed.
- test the url: 127.0.0.1:8000/ex02/init "OK" must be displayed.
- test the url: 127.0.0.1:8000/ex02/display "No data available" must be displayed.
- test the url: 127.0.0.1:8000/ex02/populate for each film, "OK" must be displayed.
- test the url: 127.0.0.1:8000/ex02/display All the data described in the subject must be displayed. The order doesn't matter. The opening_crawl column must appear and display "None" or nothing at all.

☒ Yes

☐ No

Exercise 03 - Inserting data in the ORM

- Operate the migration of the application ex03 only.
- test the url: 127.0.0.1:8000/ex03/display "No data available" must be displayed.
- test the url: 127.0.0.1:8000/ex03/populate for each film, "OK" must be displayed.
- test the url: 127.0.0.1:8000/ex03/display All the data described in the subject must be displayed. The order doesn't matter. The opening_crawl column must appear and display "None" or nothing at all.

☒ Yes

☐ No

Exercise 04 - Deleting data in SQL

- test the urls: 127.0.0.1:8000/ex04/display, 127.0.0.1:8000/ex04/populate and 127.0.0.1:8000/ex04/remove . "No data available" or error messages indicating that the table ex04_movies doesn't exist must be displayed.
- test the url: 127.0.0.1:8000/ex04/init "OK" must be displayed.
- test the urls: 127.0.0.1:8000/ex04/display and 127.0.0.1:8000/ex04/remove "No data available" must be displayed.
- test the url: 127.0.0.1:8000/ex04/populate for each film, "OK" must be displayed.
- test the url: 127.0.0.1:8000/ex04/display All the data described in the subject must be displayed. The order doesn't matter. The opening_crawl column must appear and display "None" or nothing at all.
- test the url: 127.0.0.1:8000/ex04/remove A page must display a form with a drop-down list of film titles matching the ones in the ex04_movies table. Select a film and validate. The form must redisplay and the film you've just selected must not appear in the list anymore.

- test the url: 127.0.0.1:8000/ex04/display All the data described in the subject must be displayed, minus the film you've just deleted.
- test the url: 127.0.0.1:8000/ex04/populate For each film, an error must be displayed, except for the film you've just deleted that must make an "OK" appear.

✓ Yes

✗ No

Exercise 05 - Deleting data in the ORM

- Operate the migration of the application ex05 only.
- test the urls: 127.0.0.1:8000/ex05/display and 127.0.0.1:8000/ex05/remove "No data available" must be displayed.
- test the url: 127.0.0.1:8000/ex05/populate For each film, "OK" must be displayed.
- test the url: 127.0.0.1:8000/ex05/display All the data described in the subject must be displayed. The order doesn't matter. The opening_crawl column must appear and display "None" or nothing at all.
- test the url: 127.0.0.1:8000/ex05/remove A page must display a form with a drop-down list of film titles matching the ones in the Movie model of this application. Select a film and validate. The form must redisplay and the film you've just selected must not appear in the list anymore.
- test the url: 127.0.0.1:8000/ex05/display All the data described in the subject must be displayed, minus the film you've just deleted.
- test the url: 127.0.0.1:8000/ex05/populate For each film, an error must be displayed, except for the film you've just deleted that must make an "OK" appear.

✓ Yes

✗ No

Exercise 06 - Updating data in SQL

- test the urls: 127.0.0.1:8000/ex06/display, 127.0.0.1:8000/ex06/populate and 127.0.0.1:8000/ex06/update. "No data available" or error messages indicating that the table ex06_movies doesn't exist must be displayed.
- test the url: 127.0.0.1:8000/ex06/init "OK" must be displayed.
- test the urls: 127.0.0.1:8000/ex06/display 127.0.0.1:8000/ex06/update "No data available" must be displayed.
- test the url: 127.0.0.1:8000/ex06/populate For each film, "OK" must be displayed.
- test the url: 127.0.0.1:8000/ex06/display All the data described in the subject must be displayed, including the fields created and updated that must be set at a similar date. The order doesn't matter. The opening_crawl column must appear and display "None" or nothing at all.
- test the url: 127.0.0.1:8000/ex06/update A page must display a form with a drop-down list of film titles matching the ones in the ex06_movies table. Select a film, enter a text and validate. Then go to this address: 127.0.0.1:8000/ex06/display. The field "crawling_text" of the selected film must now include the text typed in the form. The field "updated" must be updated with the current time whereas the "created" fields mustn't have changed.

✓ Yes

✗ No

Exercise 07 - Updating data in the ORM

- Operate the migration of the application ex05 only.

- test the urls: 127.0.0.1:8000/ex07/display and 127.0.0.1:8000/ex07/update "No data available" must be displayed.
- test the url: 127.0.0.1:8000/ex07/populate For each film, "OK" must be displayed.
- test the url: 127.0.0.1:8000/ex07/display All the data described in the subject must be displayed, included the created and updated fields which must be set at a similar date. The order doesn't matter. The opening_crawl column must appear and display "None" or nothing at all.
- test the url: 127.0.0.1:8000/ex07/update A page must display a form with a drop-down list of film titles matching the ones in the ex07 of this application as well as a text field. Select a film, type some text and validate. Then go to this address: 127.0.0.1:8000/ex07/display. The field "crawling_text" of the selected film must now include the text typed in the form. The field "updated" must be updated with the current time whereas the "created" fields mustn't have changed.

✓ Yes

✗ No

Exercise 08 - Foreign key in SQL

- test the urls: 127.0.0.1:8000/ex08/display and 127.0.0.1:8000/ex08/populate "No data available" or error messages indicating the tables ex08_movies and ex08_people don't exist must be displayed.
- test the url: 127.0.0.1:8000/ex08/init "OK" must be displayed for each table.
- Let's check the ex08_people table. In the terminal, type the command: `psql -U djangouser -d formationdjango -c "\d ex08_people"` Check the output is indeed the description of a table with the fields and specifications described in the subject.
- Do the same with the ex08_planets table with the command: `psql -U djangouser -d formationdjango -c "\d ex08_planets"`
- test the urls: 127.0.0.1:8000/ex08/display. "No data available" must be displayed.
- test the url: 127.0.0.1:8000/ex08/populate For each file, "OK" must be displayed.
- test the url: 127.0.0.1:8000/ex08/display the result must be: Saesee Tiin - Iktotch - arid, rocky, windy Tion Medon
 - Utapau - temperate, arid, windy

✓ Yes

✗ No

Exercise 09 - Foreign key in the ORM

- Operate the migration of the application ex09.
- Let's check the People model. In the terminal, enter the command: `psql -U djangouser -d formationdjango -c "\d ex09_people"` Check that the output indeed is the description of a table with the fields and specifications described in the subject.
- Haga lo mismo con el modelo Planets con el comando: `psql -U djangouser -dformationdjango -c "\d ex09_planets"`
- Pruebe la URL: 127.0.0.1:8000/ex09/display. Debería aparecer el siguiente mensaje: "No hay datos disponibles, utilice la siguiente línea de comandos:"
 seguido de una línea de comandos. Ejecútelo en la terminal desde la raíz del proyecto. Si todo salió bien, debería aparecer el mensaje "Se instalaron 147 objetos de 1 dispositivo" .
- prueba la URL: 127.0.0.1:8000/ex09/display La pantalla debería ser: Saesee Tiin - Iktotch - árido, rocoso, ventoso Tion Medon
 - Utapau - templado, árido, ventoso



Si



No

Ejercicio 10 - Muchos a muchos en el ORM

- Operar la migración de la aplicación ex09.
- Instale el accesorio ex10_initial_fixture.json.
- Pruebe la URL: 127.0.0.1:8000/ex10/: Se debe mostrar un formulario. Compruebe que los campos coincidan con los descritos en el asunto.
- Introduzca los siguientes datos y valide el formulario (el formato de la fecha puede haber cambiado. Adáptelo si es necesario).
Fecha mínima de lanzamiento de películas: 1980-01-01
Fecha máxima de lanzamiento de películas: 1989-12-31 Diámetro del planeta mayor que: 1
Género del personaje: n/a

El resultado debe ser (no necesariamente en este orden):

Return of the Jedi	C-3PO	n/a	Tatooine
10465 El retorno del Jedi R2-D2	n/a	Naboo	
12120 El Imperio Contraataca C-3PO	n/a	Tatooine	
10465 El Imperio Contraataca R2-D2	n/a	Naboo	
12120			

- Rellene el formulario de nuevo. Esta vez, introduzca dos fechas idénticas (y cualquier otro dato en los demás campos). La página debe mostrar: "Nada que corresponda a su investigación".



Si



No

Calificaciones

No olvides revisar la bandera correspondiente a la defensa

✓ De acuerdo

★ Proyecto destacado

Trabajo vacío

📄 Trabajo incompleto

🗨️ Engañar

👤 Chocar

⚠️ Situación preocupante

🚫 Función prohibida

Conclusión

Deja un comentario en esta evaluación (2048 caracteres máximo)

Finalizar la evaluación