

DAW/DAM. ACTIVIDAD EVALUABLE 3. DISEÑO DE PROGRAMAS EN MYSQL. PARTE 2

DAW/DAM. Bases de datos (BD)

AE3. DISEÑO DE PROGRAMAS EN MYSQL

Actividad evaluable 3. Parte 2

Abelardo Martínez y Pau Miñana

Curso 2023-2024

Aspectos a tener en cuenta

Importante

Si buscas las soluciones por Internet o preguntas al oráculo de ChatGPT, te estarás engañando a ti mismo. Ten en cuenta que **ChatGPT no es infalible ni todopoderoso.**

Es una gran herramienta para agilizar el trabajo una vez se domina una materia, pero usarlo como atajo en el momento de adquirir habilidades y conocimientos básicos perjudica gravemente tu aprendizaje. Si lo utilizas para obtener soluciones o asesoramiento respecto a las tuyas, revisa cuidadosamente las soluciones propuestas igualmente. Intenta resolver las actividades utilizando los recursos que hemos visto y la documentación extendida que encontrarás en el “Aula Virtual”.

A. Instrucciones y normativa

1. DESCRIPCIÓN

Se pide diseñar en **MySQL** el conjunto de PROGRAMAS (*SCRIPTS*) más adecuado para cada caso.

Entrega **únicamente** los programas que correspondan con la última cifra de tu DNI/NIE:

0-4 → MODELO A

5-9 → MODELO B

2. PLAZO DE ENTREGA Y PORCENTAJES

- **Porcentajes en la EVALUACIÓN:** 30% de la nota total es para las evaluables.
- **Porcentajes de la ACTIVIDAD:** 50% de las evaluables (hay dos por evaluación).
- **Plazo de entrega de AMBAS PARTES:** **23:59 del lunes, 25 de marzo de 2024**

3. CALIFICACIÓN

La entrega no es obligatoria ni hay nota mínima. Se **calificará de 0 a 10 según el apartado de calificación** proporcionado en este documento (VER DETALLE).

4. RECURSOS

Debes estudiar todos los materiales que te hemos proporcionado, prestando especial atención a las tareas no evaluables y a todo el material extra.

Utiliza la plantilla que te proporcionamos para facilitar la corrección de la actividad.

5. PLAGIO

Debes evitar que otros alumnos se copien tu trabajo y tener cuidado para prevenir esta situación.

Tarea INDIVIDUAL. En caso de sospecha de autoría será requerida una entrevista oral.

6. INSTRUCCIONES DE ENTREGA

La tarea se entregará en un único PDF que contendrá los ejercicios resueltos siguiendo la plantilla indicada.

NO SE ACEPTARÁN ENTREGAS EN OTRO FORMATO.

7. SOLUCIONES Y RESULTADOS

Recibirás la calificación desglosada por cada apartado, y el total, junto con cualquier

comentario que brinde sugerencias sobre cómo podrías haberlo hecho mejor.

B. Pasos a seguir

1. Descarga del Aula Virtual la plantilla y el *script* DDL asociado que se corresponde con tu modelo (A o B).
2. En los enunciados dispones del diagrama físico, aunque puedes generarlo con MySQL Workbench siguiendo los pasos de este vídeo: <https://www.youtube.com/watch?v=8M1eGDkWffk>
3. **Recuerda que deberás partir de esa BASE DE DATOS, independientemente del diagrama que presentaste en la anterior actividad evaluable.**
4. **Recomendaciones:**
 - Se recomienda realizar también las consultas y scripts correspondientes a la primera parte a modo de práctica, pues ambas actividades se complementan bastante.
 - Diseña primero los procedimientos, funciones y triggers y, en caso necesario, crea datos ficticios para las pruebas.
 - Usa la sintaxis MySQL.
 - Usa mayúsculas/minúsculas y tabulaciones/espacios para hacer el código lo más legible.
 - No uses variables de usuario dentro del programa. Usa variables locales si las necesitas.
 - Recuerda también usar los errores en los programas oportunos para evitar que se realicen operaciones no permitidas. Puedes usar esta sintaxis para los mensajes de error:

```
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT =  
'Operación no permitida\n*****\n====> Mensaje de error.\n*****\n';  
  
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT =  
'Parámetros inesperados\n*****\n====> Mensaje de error.\n*****\n';
```

C. Rúbrica. Criterios de calificación

ÍTEMS EVALUABLES	DETALLE ÍTEMS EVALUABLES	PUNTUACIÓN
Programa 1	El programa es correcto: hace lo que se pide.	2
Programa 2	Usa ALIAS cuando es necesario para facilitar y hacer más clara las consultas dentro de los programas. Hace un uso apropiado de las mayúsculas/minúsculas, las tabulaciones y los saltos de línea para hacer cada SCRIPT lo más legible posible. Usa mensajes de error.	2
Programa 3		2
Programa 4		2
Programa 5		2

D. Enunciados

Se pide diseñar en **MySQL** el conjunto de PROGRAMAS (*SCRIPTS*) más adecuado para cada caso.

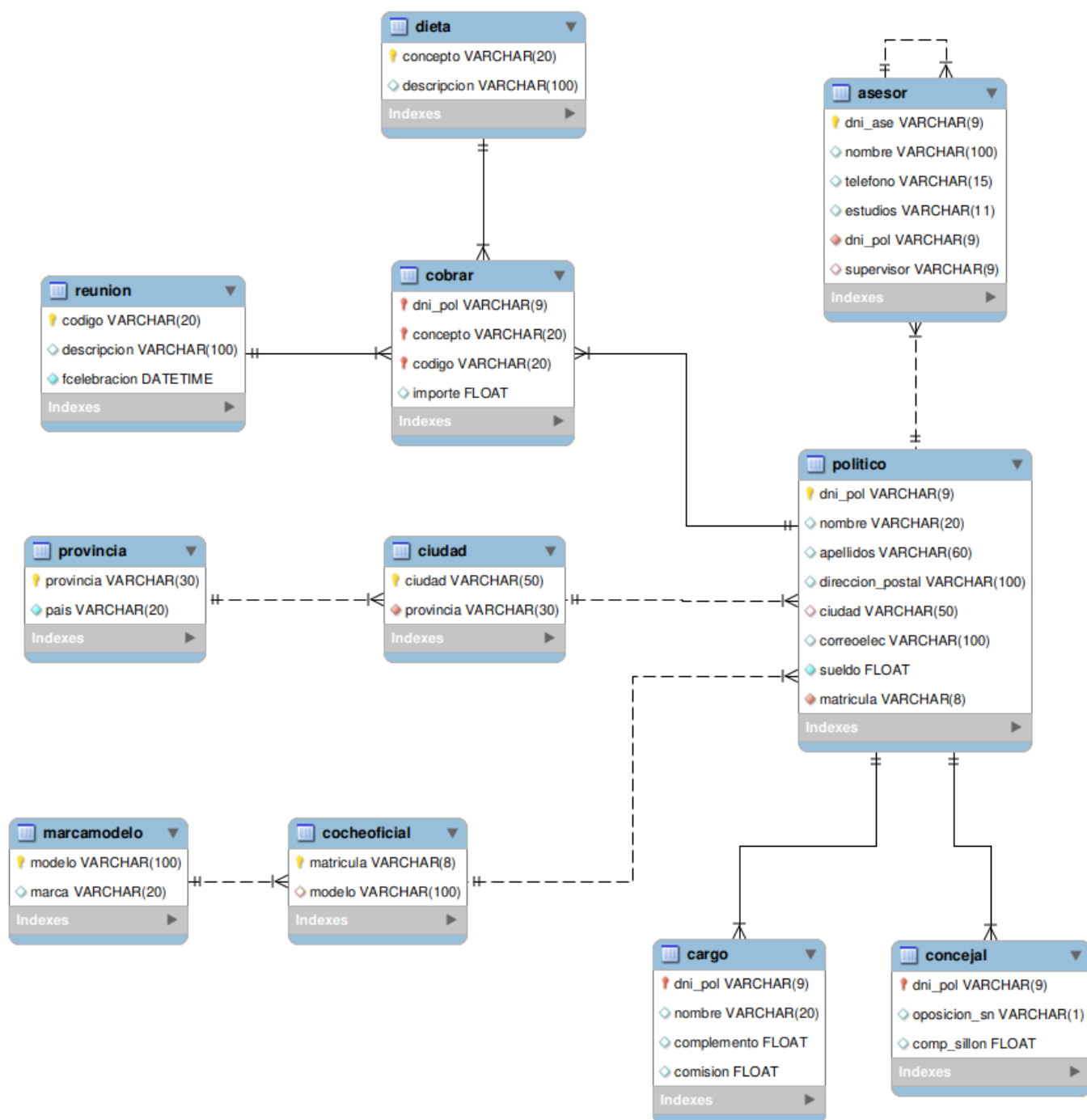
Entrega **únicamente** los programas que correspondan con la última cifra de tu DNI/NIE:

0-4 → MODELO A

5-9 → MODELO B

Modelo A

Ayuntamiento de Corrupoly (EVALUABLE)



A1. Programa 1

Crea un procedimiento almacenado para obtener el dni, nombre y apellidos de los políticos que han cobrado alguna dieta cuya concepto comienza por la letra pasada como parámetro (de entrada) y son del tipo pasado como parámetro (de entrada) -CA para Cargo y CO para Concejal- y devuelve el número de resultados en un tercer parámetro (de salida). Ordena los resultados por nombre y apellidos.

Se pide:

a) Crea el procedimiento **pListarPoliticodietas_porLetrayTipo**. Muestra un mensaje de error si no se recibe 'CA' o 'CO' como 2º parámetro.

A2. Programa 2

Crea una función para obtener, dado el concepto de dieta, el número de cobros asociados y una segunda función para obtener, también dado un concepto de dieta, el importe total de euros que se han cobrado. Usando las dos funciones, lista la dieta (concepto y descripción) que más cobros asociados tiene y otro listado para la dieta (concepto y descripción) que más dinero ha costado.

Se pide:

a) Crea la función **fNumCobrosDieta**

b) Crea la función **fImporteCobrosDieta**

A3. Programa 3

Crea el trigger necesario para asegurar que cuando insertamos una reunión, la fecha de celebración no sea anterior o igual a la fecha actual.

Se pide:

a) Crea el trigger **tComprobarFechaReunion**

A4. Programa 4

Al traducir del diseño conceptual al modelo relacional, existía una pérdida semántica en las especializaciones que no eran Parcial+Solapada (PS), quedándose como una restricción de integridad que dijimos que “ya resolveríamos más adelante”. Pues bien, mediante triggers podemos traducir cualquier especialización (TS, TD o PD) y superar así esa pérdida semántica, aunque no de manera sencilla. Solo las disyuntas tienen una solución con triggers más o menos sencilla.

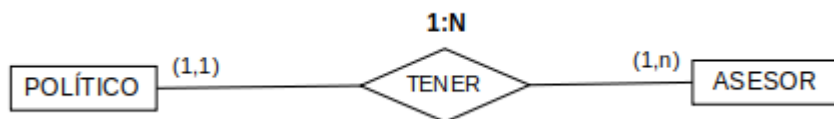
Aunque en el diagrama la especialización es total (un político puede ser cargo y concejal a la vez), **crea los triggers necesarios para asegurar la especialización DISYUNTA de POLÍTICO en Cargo y Concejal** durante las inserciones, ignorando las actualizaciones y los borrados por su complejidad.

Se pide:

- Crea el **trigger** necesario **para prevenir inserciones incorrectas en Cargo**
- Crea el **trigger** necesario **para prevenir inserciones incorrectas en Concejal**

A5. Programa 5

Crea los triggers necesarios para asegurar la **participación 1:N de la relación entre POLÍTICO y ASESOR**, ignorando las inserciones por motivos de simplicidad de gestión.

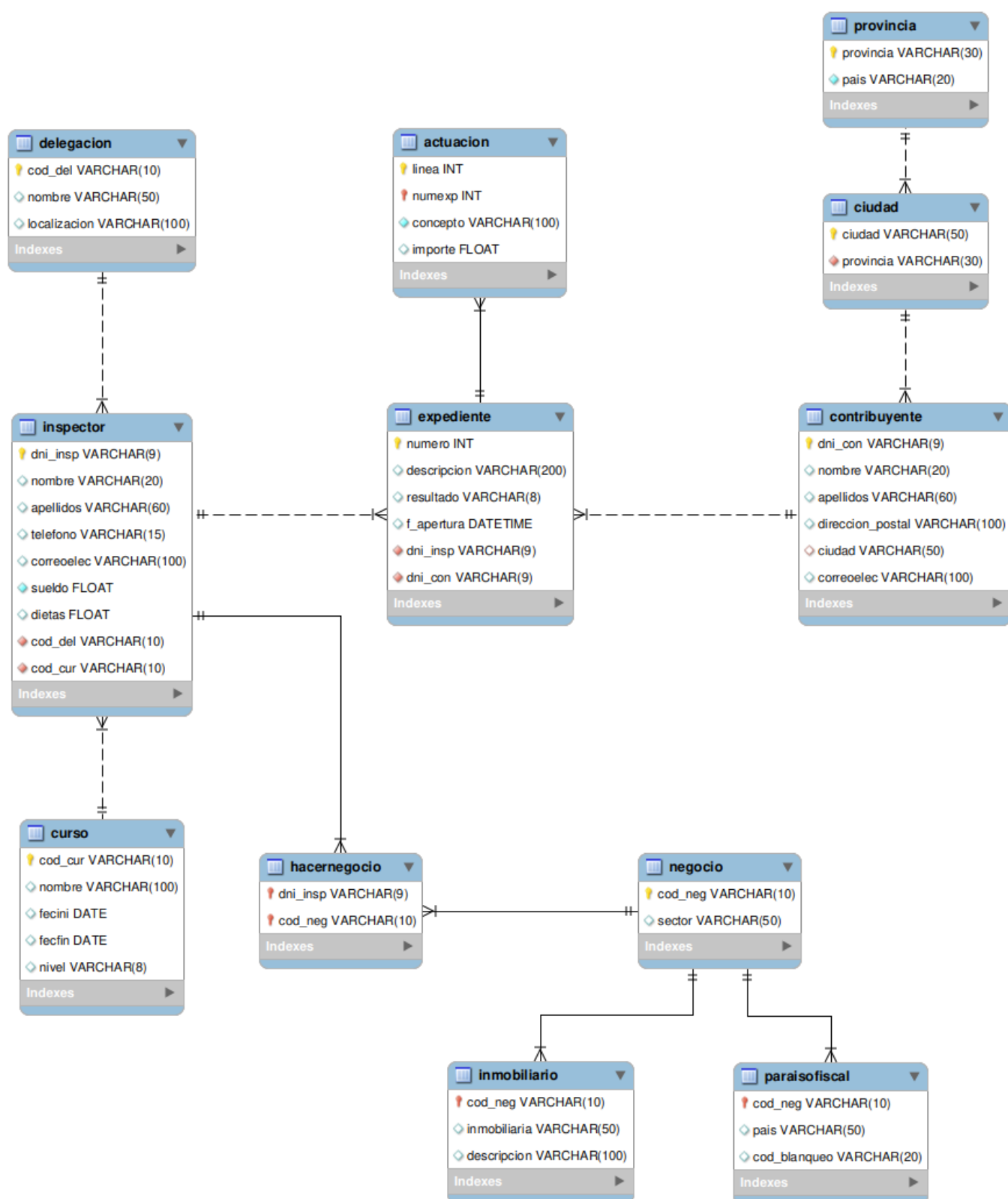


Se pide:

- Crea el **trigger** necesario **para prevenir borrados de ASESOR** que rompan la participación (1,n) en la relación de POLÍTICO con ASESOR.
- Crea el **trigger** necesario **para prevenir actualizaciones del campo "dni_pol" de la tabla ASESOR** que rompan la participación (1,n) en la relación de POLÍTICO con ASESOR.

Modelo B

**Ministerio de Hacienda de Corrupoly
(EVALUABLE)**



B1. Programa 1

Crea una función para obtener, dado el número de expediente, el número total de actuaciones asociadas y una segunda función para obtener, también dado un número de expediente, el importe total de euros que se han recaudado. Usando las dos funciones, lista el expediente (número, descripción y resultado) que más actuaciones asociadas tiene y otro listado para el expediente (número, descripción y resultado) que más dinero ha recaudado.

Se pide:

- a) Crea la función **fNumActuacionesExpediente**
- b) Crea la función **fImporteTotalExpediente**

B2. Programa 2

Crea el trigger necesario para asegurar que cuando insertamos un expediente, la fecha de apertura no sea anterior o igual a la fecha actual.

Se pide:

- a) Crea el trigger **tComprobarFechaApertura**

B3. Programa 3

Al traducir del diseño conceptual al modelo relacional, existía una pérdida semántica en las especializaciones que no eran Parcial+Solapada (PS), quedándose como una restricción de integridad que dijimos que “ya resolveríamos más adelante”. Pues bien, mediante triggers podemos traducir cualquier especialización (TS, TD o PD) y superar así esa pérdida semántica, aunque no de manera sencilla. Solo las disyuntas tienen una solución con triggers más o menos sencilla.

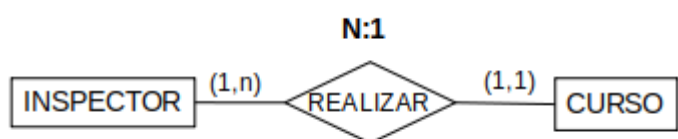
Crea los **triggers** necesarios **para asegurar la especialización DISYUNTA de NEGOCIO en Inmobiliario y Paraíso Fiscal** durante las inserciones, ignorando las actualizaciones y los borrados por su complejidad.

Se pide:

- Crea el **trigger** necesario **para prevenir inserciones incorrectas en Inmobiliario**
- Crea el **trigger** necesario **para prevenir inserciones incorrectas en Paraíso Fiscal**

B4. Programa 4

Crea los triggers necesarios para asegurar la **participación 1:N de la relación entre INSPECTOR y CURSO**, ignorando las inserciones por motivos de simplicidad de gestión.



Se pide:

- Crea el **trigger** necesario **para prevenir borrados de INSPECTOR** que rompan la participación (1,n) en la relación de INSPECTOR con CURSO.
- Crea el **trigger** necesario **para prevenir actualizaciones del campo "cod_cur" de la tabla INSPECTOR** que rompan la participación (1,n) en la relación de INSPECTOR con CURSO.

B5. Programa 5

Crea los triggers necesarios para asegurar que los números de línea de una misma actuación son consecutivos cuando se inserta una nueva línea de actuación de un expediente. Para hacerlo más sencillo, ignora los borrados, gestiona las inserciones y prohíbe las actualizaciones del campo "línea" en esa tabla.

Se pide:

a) Crea el trigger `tAntesActualizarLineasActuacion`

- Muestra un mensaje de error cuando se intenta cambiar el campo "línea"

b) Crea el trigger `tAntesInsertarLineasActuacion`

- El campo "línea" debe ser >0 y siempre consecutivo para una misma ACTUACIÓN, de manera que, si por ejemplo insertamos la línea 6 de la actuación del expediente 111, debe existir antes la 5. Si no existe esa línea 5, debe cancelarse la operación con un mensaje de error.



Obra publicada con [Licencia Creative Commons Reconocimiento Compartir igual 4.0](https://creativecommons.org/licenses/by-sa/4.0/)