

EXAMEN ExDAW_2C_BD**CUATRIMESTRE:SEGUNDO****Fecha: 14/05/2021**

<i>CICLO: CFGS DAW</i>		<i>CURSO:1º</i>	<i>CALIFICACIÓN:</i>
<i>MÓDULO: Base de Datos</i>		<i>Turno:</i>	
<i>Nombre: Jose Antonio</i>		<i>Apellidos: Mateo Lara</i>	

Instrucciones:

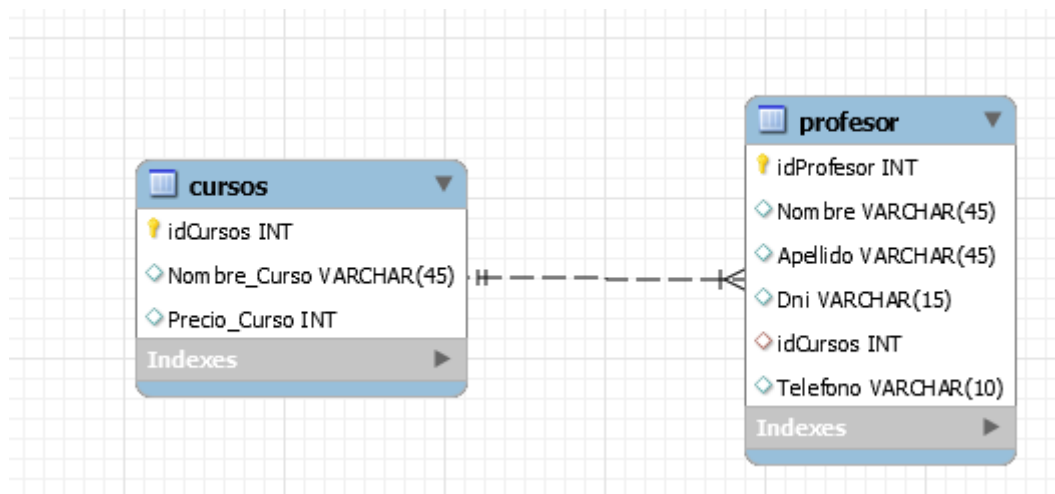
- *Esta prueba tiene como finalidad evaluar los aprendizajes del módulo Entornos de Desarrollo.*
- *Lee atentamente el enunciado de los ejercicios.*
- *Responde sobre el mismo documento.*

Si las instrucciones no se siguen como se especifican el examen no será evaluado (Es obligatorio que el formato del fichero subido a la plataforma sea PDF)

PARTE PRÁCTICA.

- El examen práctico tiene una puntuación máxima de 10 puntos.
- Para superar la parte práctica se requiere alcanzar un mínimo de 5 puntos.
- Una vez realizada la parte práctica, se subirá un fichero **comprimido**, en el apartado habilitado (final de la página web) del módulo correspondiente en la plataforma, que contendrá:
 - Un documento Word/Writer creado por el alumno, donde se realizarán las distintas actividades planteadas y se generará en PDF
- El nombre del fichero comprimido que se envía a la plataforma deberá tener el siguiente formato:
 - **Apellido1_Apellido2_Nombre_ExDAW_2C_BD**

Ejercicio 1. (3 puntos)



Indicar las sentencias SQL necesarias para realizar las siguientes acciones (leer todas antes de comenzar por la primera):

- a. Crear la base de datos Academia.

```
CREATE DATABASE ACADEMIA
```

- b. Crear las tablas Cursos y Profesor, identificando las claves primarias.

Cursos:

```
CREATE TABLE cursos(
idCursos INT(10) NOT NULL,
nombre_curso VARCHAR(45) NOT NULL,
precio_curso INT NOT NULL,
CONSTRAINT PRIMARY KEY(idCursos)
);
```

Profesor:

```
CREATE TABLE profesor(
idProfesor INT(10) NOT NULL,
nombre VARCHAR(20) NOT NULL,
apellido VARCHAR(20) NOT NULL,
Dni VARCHAR (15) NOT NULL,
telefono VARCHAR(10) NOT NULL,
idCursos INT(10) NOT NULL,
CONSTRAINT PRIMARY KEY(idProfesor)
)
```

- c. Modificar la tabla Profesor para añadir la foreign key del idCursos de la tabla Cursos.

```
ALTER TABLE profesor
  ADD COLUMN idCursos INT NOT NULL,
  ADD FOREIGN KEY (idCursos)
    REFERENCES cursos (idCursos)
```

- d. Insertar al menos dos registros por cada una de las tablas. El nombre de Uno de los cursos será 'Base de Datos' .

CURSOS:

```
INSERT INTO cursos VALUES(001,"Programacion Python",1400);
INSERT INTO cursos VALUES (002,"Base de Datos",750);
INSERT INTO cursos VALUES (003,"Big Data",5400);
```

PROFESOR:

```
INSERT INTO profesor VALUES(001,"Manuel","Gomez","78957246B",652782146,003);
INSERT INTO profesor VALUES(002,"Sara","Calvo","75841267A",658423815,001);
```

- e. Eliminar el campo telefono de la tabla Profesor.

```
ALTER TABLE profesor DROP COLUMN telefono
```

- f. Cambiar el nombre del Curso 'Base de Datos' por el de 'Oracle'

```
UPDATE cursos SET nombre_curso = "Oracle" WHERE idCursos=2
```

- g. Eliminar el Curso con nombre 'oracle'

```
DELETE FROM cursos WHERE nombre_curso="Oracle"
```

Ejercicio 2. (2 puntos)

Acceda a la dirección:

https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all

donde encontrará una base de datos de ejemplo de la web w3schools.

Ejecute las sentencias SQL necesarias para obtener los siguientes resultados (muestre una captura con la sentencia y los resultados):

- a) Muestre, las distintas ciudades de la tabla Customers que pertenecen al Country = 'Spain'. Ordene los resultados alfabéticamente por el nombre de la City. (Tablas Customers.

El resultado sería:

Result:	
Number of Records: 3	
City	
Barcelona	
Madrid	
Sevilla	

SELECT Distinct city from customers where country = "Spain" Order By city

SQL Statement:

```
SELECT Distinct city from customers where country = "Spain" Order By city
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 3

City
Barcelona
Madrid
Sevilla

b) Muestre los campos, ProductID, ProductName y Price de la tabla [Products] de aquellos productos cuyo ProductName comience por la letra 'C' y que tengan un precio > 23 €. Muestre el resultado ordenado por Price descendente. Incluya sus iniciales en los nombres de columnas.

(Tabla Products)

El resultado sería:

Result:		
Number of Records: 3		
ProductID_IR	ProductName_IR	price_IR
38	Côte de Blaye	263.5
18	Carnarvon Tigers	62.5
60	Camembert Pierrot	34

SELECT ProductID AS ProductID_JAML, ProductName AS ProductName_JAML, Price AS Price_JAML FROM Products WHERE ProductName LIKE "C%" AND Price>23 ORDER BY Price DESC

SQL Statement:

```
SELECT ProductID AS ProductID_JAML, ProductName AS ProductName_JAML, Price AS Price_JAML FROM Products WHERE ProductName LIKE "C%" AND Price>23 ORDER BY Price DESC
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

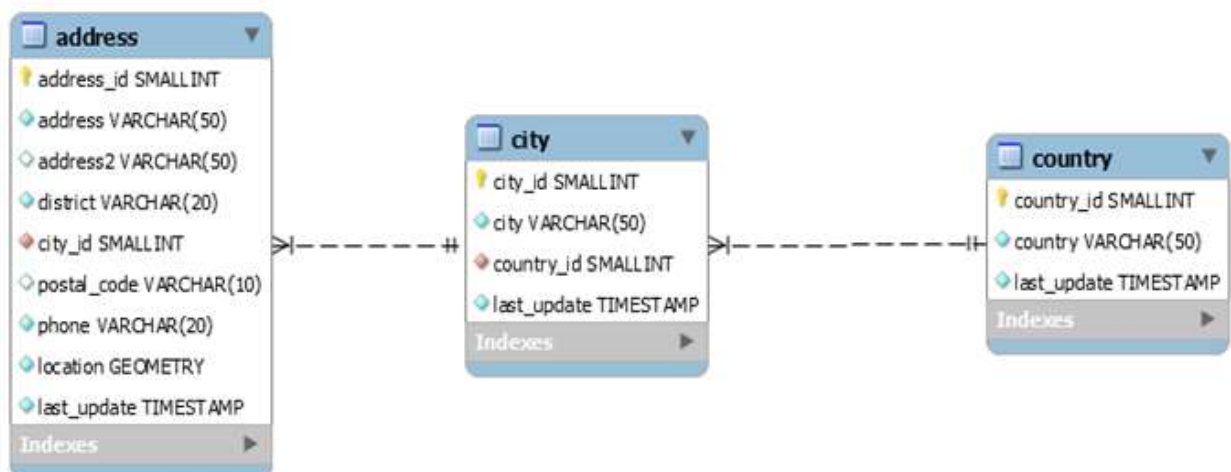
Number of Records: 3

ProductID_JAML	ProductName_JAML	Price_JAML
38	Côte de Blaye	263.5
18	Carnarvon Tigers	62.5
60	Camembert Pierrot	34

Ejercicio 3. (2 puntos)

Se utilizará la base de datos "Examen_Final".

Se utilizarán las tablas address, Country y City relacionadas 1 a M por el campo city_id y Country_id.



Crear un procedimiento almacenado, llamado Pais(), que reciba como parámetros una dirección (address) y devuelva un parámetro de salida con el Nombre del país al que pertenece esa dirección.

Ejemplo de ejecución:

```
SET @Address = 'La Serna';
Call pais(@Address, @pais);
SELECT @pais;
```

El resultado sería:

Result Grid	
	@pais
▶	Spain

```
CREATE PROCEDURE pais (address VARCHAR(50) , country VARCHAR(50))
SELECT country.country
FROM country, city, address
WHERE address.city_id = city.city_id AND city.country_id=country.country_id AND
address.address=@Address
```

Ejercicio 4. (3 puntos)

Realizar las siguientes consultas de bases de datos, mostrando la captura con la sentencia y resultados obtenidos.

Se utilizará la base de datos "Examen_Final". Se puede consultar el modelo de la base de datos en el anexo al final de este documento. Cuidado con las mayúsculas y minúsculas del nombre de la tabla si da error de ejecución, en los enunciados se indica el nombre en mayúsculas, pero podría variar, no importa.

- a. Obtener los títulos de las películas, Nombre y Apellidos en los que participo el actor con nombre 'PENELOPE', ordenado por título de película.
(Tablas: FILM, ACTOR, FILM_ACTOR)

```
SELECT DISTINCT film.title, actor.first_name, actor.last_name FROM film, actor, film_actor WHERE actor.first_name="PENELOPE" ORDER BY film.title
```

✓ Mostrando filas 0 - 24 (total de 4000. La consulta tardó 27.9570 segundos.) [título: ACADEMY DINOSAUR... - AIRPLANE SIERRA...]

`SELECT DISTINCT film.title, actor.first_name, actor.last_name FROM film, actor, film_actor WHERE actor.first_name="PENELOPE" ORDER BY film.title`

1 > >> Número de filas: 25 Filtrar filas:

+ Opciones

title	first_name	last_name
ACADEMY DINOSAUR	PENELOPE	PINKETT
ACADEMY DINOSAUR	PENELOPE	GUINNESS
ACADEMY DINOSAUR	PENELOPE	MONROE
ACADEMY DINOSAUR	PENELOPE	CROWYN
ACE GOLDFINGER	PENELOPE	PINKETT
ACE GOLDFINGER	PENELOPE	GUINNESS
ACE GOLDFINGER	PENELOPE	MONROE
ACE GOLDFINGER	PENELOPE	CROWYN
ADAPTATION HOLES	PENELOPE	GUINNESS
ADAPTATION HOLES	PENELOPE	MONROE
ADAPTATION HOLES	PENELOPE	CROWYN
ADAPTATION HOLES	PENELOPE	PINKETT
AFFAIR PREJUDICE	PENELOPE	PINKETT
AFFAIR PREJUDICE	PENELOPE	GUINNESS
AFFAIR PREJUDICE	PENELOPE	MONROE
AFFAIR PREJUDICE	PENELOPE	CROWYN
AFRICAN EGG	PENELOPE	CROWYN
AFRICAN EGG	PENELOPE	PINKETT
AFRICAN EGG	PENELOPE	GUINNESS
AFRICAN EGG	PENELOPE	MONROE
AGENT TRUMAN	PENELOPE	PINKETT
AGENT TRUMAN	PENELOPE	GUINNESS
AGENT TRUMAN	PENELOPE	MONROE
AGENT TRUMAN	PENELOPE	CROWYN
AIRPLANE SIERRA	PENELOPE	CROWYN

- b. Obtener los títulos de las películas que pertenecen a la categoría 'Comedy', ordenado por título de película.

(Tablas: FILM, FILM_CATEGORY, CATEGORY)

```
SELECT film.title FROM film, film_category, category WHERE category.category_id = film_category.category_id AND film_category.film_id = film.film_id AND category.name="Comedy" ORDER BY film.title
```

Mostrando filas 0 - 24 (total de 50, La consulta tardó 0.0153 segundos.) [título: AIRPLANE SIERRA... - HEDWIG ALTER...]

```
SELECT film.title FROM film, film_category, category WHERE category.category_id = film_category.category_id AND film_category.film_id = film.film_id AND category.name="Comedy" ORDER BY film.title
```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla

+ Opciones

Editar	Copiar	Borrar	título
			AIRPLANE SIERRA
			ANTHEM LUKE
			BRINGING HYSTERICAL
			CAPER MOTIONS
			CAT CONEHEADS
			CLOSER BANG
			CONNECTION MICROCOSMOS
			CONTROL ANTHEM
			CRAZY HOME
			DANDY PITTSBURGH
			DOOM DANCING
			DOWNHILL ENOUGH
			DYING MAJESTY
			ELEMENT FREDDY
			FERRY'S MOTHER
			FIREBALL SYRACUSE
			FLINTSTONES HAPPINESS
			FRANKENSTEIN STRANGER
			FREEDOM CLEOPATRA
			GOLD RIVER
			GROUNDHOG UNCUT
			GUMFIGHT MOON

- c. Obtener el nombre de todas las películas grabadas en italiano ordenadas por title, (Tablas: FILM, LANGUAGE)

```
SELECT film.title FROM film, language WHERE language.name="Italian" AND film.language_id=language.language_id ORDER BY film.title
```

Mostrando filas 0 - 2 (total de 3, La consulta tardó 0.0058 segundos.) [título: AFFAIR PREJUDICE... - ANNIE IDENTITY...]

```
SELECT film.title FROM film, language WHERE language.name="Italian" AND film.language_id=language.language_id ORDER BY film.title
```

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla

+ Opciones

Editar	Copiar	Borrar	título
			AFFAIR PREJUDICE
			AIRPORT POLLOCK
			ANNIE IDENTITY

- d. Obtener el nombre de los lenguajes y el total de películas realizadas, mostrando solo los lenguajes de los que se han realizado menos de 20 películas(Tabla: FILM y LENGUAJE)

SIN TIEMPO

Anexo. modelo de la base de datos Examen.

