

Nombre: _Rodríguez López, Alejandro_____ UO _281827___ PL: _03___

INSTRUCCIONES

La entrega de las soluciones será en formato PDF con nombre de fichero:

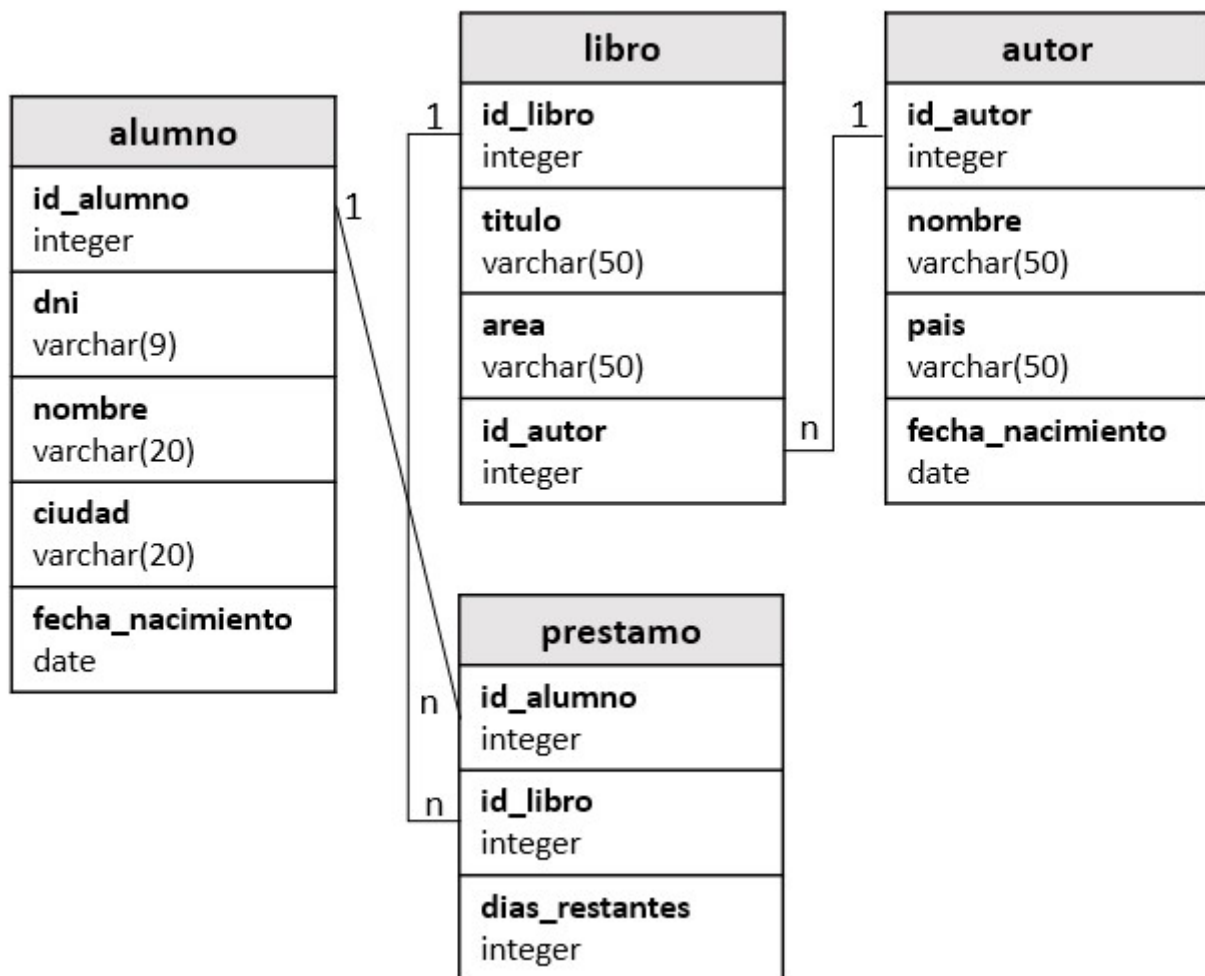
Ex1_UOXXXXXX_Nombre_Apellido1.pdf

No incluir signos de puntuación en el nombre del fichero tales como acentos o diéresis.

Indica en el apartado RESPUESTA tu orden SQL para cada ejercicio. Se proporciona la salida esperada como resultado. **Importante:** Obtener la misma salida no es indicativo de obtener la máxima puntuación del ejercicio.

BD: biblioexamen (Descarga y ejecuta los scripts disponibles en Campus Virtual). Los datos almacenados en la BD **no** contienen signos de puntuación tales como acentos.

MODELO E-R de la BD biblioexam



Ejercicio 1 (1.5 puntos)

(0.75 puntos) Crea una nueva tabla, llamada grado, que indicará los datos del grado que está realizando un alumno de la base de datos. La información y restricciones se indican a continuación:

TABLA	COLUMNA	TIPO	RESTRICCIONES
grado	id_grado	integer	PRIMARY KEY
	nombre	varchar(50)	UNIQUE NOT NULL
	creditos	integer	NOT NULL
	fecha_creacion	date	NOT NULL

RESPUESTA:

```
CREATE TABLE grado (  
    id_grado INTEGER PRIMARY KEY,  
    nombre VARCHAR(50) UNIQUE NOT NULL,  
    creditos INTEGER NOT NULL,  
    fecha_creacion DATE NOT NULL  
);
```

(0.25 puntos) Inserta en la nueva tabla dos tuplas, tal que obtengas la siguiente salida:

```
biblioexamen=# SELECT * FROM grado;  
 id_grado |          nombre          | creditos | fecha_creacion  
-----+-----+-----+-----  
      1 | Grado en Ingenieria Informatica |      240 | 1985-02-15  
      2 | Grado en Ingenieria Industrial   |      240 | 1988-08-20
```

RESPUESTA:

```
INSERT INTO grado VALUES (1, 'Grado en Ingenieria Informatica', 240, '1985/02/15');  
INSERT INTO grado VALUES (2, 'Grado en Ingenieria Industrial', 240, '1988/08/20');
```

(0.25 puntos) Modifica la tabla alumno para añadir una nueva columna, llamada id_grado, que será de tipo INTEGER y cuya **única restricción** es FOREIGN KEY en referencia al id_grado de la nueva tabla.

RESPUESTA:

```
ALTER TABLE alumno ADD COLUMN id_grado INTEGER REFERENCES grado(id_grado);
```

(0.25) Asigna a los alumnos con id_alumno 1 y 2 su grado con id_grado 1 y 2, respectivamente. No es necesario asignar un id_grado al resto de alumnos.

RESPUESTA:

```
UPDATE alumno SET id_grado=1 WHERE id_alumno=1;  
UPDATE alumno SET id_grado=2 WHERE id_alumno=2;
```

Ejercicio 2 (1.5 puntos)

Muestra, para los alumnos con id inferior a 5, su id, nombre, id de libro que tenga en préstamo, y el correspondiente título y autor de tales libros en préstamo.

id_alumno	nombre	id_libro	titulo	nombre
1	Alberto Vega	1	El silencio	Juan Rufol
2	Luisa Perez	3	El hobbit	Willian Golding
2	Luisa Perez	14	Vuelta al mundo	Leticia Lopez Juarez
2	Luisa Perez	15	Entre las tinieblas	Leticia Lopez Juarez
3	Antonio Martinez	2	El esclavo	Juan Rufol
3	Antonio Martinez	7	Ingenieria de Software	Mario Benedetti
4	Jose Garcia	10	Tutorial avanzado de C++	Ana Moreno
4	Jose Garcia	11	Tutorial SQL	Ana Moreno

RESPUESTA:

```
SELECT alumno.id_alumno, alumno.nombre, libro.id_libro, libro.titulo, autor.nombre
FROM alumno
JOIN prestamo USING(id_alumno)
JOIN libro USING(id_libro)
JOIN autor USING (id_autor)
WHERE alumno.id_alumno < 5;
```

Ejercicio 3 (2 puntos)

Muestra los id de libros, su título y área de aquellos libros cuyo título empiece por la letra E y su área sea Novela. **Es obligatorio el uso de algún operador de conjuntos (UNION, INTERSECT, EXCEPT, ...).**

id_libro	titulo	area
1	El silencio	Novela
9	Entregame tu corazon	Novela

RESPUESTA:

```
SELECT libro.id_libro, libro.titulo, libro.area
FROM libro
WHERE UPPER(libro.titulo) LIKE 'E%'
INTERSECT
SELECT libro.id_libro, libro.titulo, libro.area
FROM libro
WHERE UPPER(libro.area) = 'NOVELA';
```

Ejercicio 4 (2 puntos)

Diseña una consulta que muestre el id de alumno y la cantidad de libros en préstamo que tiene cada alumno. Sólo debe formar parte del resultado el TOP 7 de alumnos con más libros en préstamo.

id_alumno	cantidad_prestamos
5	3
6	3
2	3
11	3
9	3
8	2
3	2

RESPUESTA:

```
SELECT alumno.id_alumno, COUNT(prestamo.id_libro)
FROM alumno
JOIN prestamo USING (id_alumno)
GROUP BY (alumno.id_alumno)
ORDER BY COUNT(prestamo.id_libro) DESC
LIMIT (7);
```

Ejercicio 5 (1.5 puntos)

Obtén, mediante el uso de una subconsulta los títulos de los libros en préstamo cuyos días restantes del préstamo de algún ejemplar sean mayor o igual a la media de los días restantes de todos los préstamos, sin repetir en el resultado el título de un mismo libro para el que hay más de un ejemplar que satisface la condición del ejercicio.

```
          titulo
-----
Tutorial SQL
El silencio
El hobbit
Como la vida misma
Entre las tinieblas
Tutorial avanzado de C++
La isla perdida
Vuelta al mundo
```

RESPUESTA:

```
SELECT DISTINCT libro.titulo
FROM libro
JOIN prestamo USING (id_libro)
WHERE prestamo.dias_restantes >= (
    SELECT AVG(prestamo.dias_restantes)
    FROM prestamo
);
```

Ejercicio 6 (1.5 puntos)

Mostrar el id de alumno y su nombre de aquellos alumnos que tengan en préstamo todos los libros cuyo área del libro es Internet.

id_alumno	nombre
4	Jose Garcia
10	Antonio Guerrero

RESPUESTA:

```
SELECT alumno.id_alumno, alumno.nombre
FROM alumno
WHERE alumno.id_alumno IN (
    SELECT alumno.id_alumno
    FROM alumno
    JOIN prestamo USING (id_alumno)
    JOIN libro USING (id_libro)
    WHERE UPPER(libro.area) = 'INTERNET'
    GROUP BY (alumno.id_alumno)
    HAVING COUNT(prestamo.id_libro) = (
        SELECT COUNT(libro.id_libro)
        FROM libro
        WHERE UPPER(libro.area)='INTERNET'
    )
);
```