



APELLIDOS \_\_\_\_\_ NOMBRE \_\_\_\_\_

## Instrucciones

- El examen consta de dos partes. Esta parte dura media hora. La segunda parte durará dos horas y media.
- Escribe **en esta hoja** tu nombre y la respuesta al test.
- Puedes utilizar lápiz.

## PARTE 1: Test de cuestiones teórico-prácticas

1. (2 puntos) **SOBRE LA SIGUIENTE CUADRÍCULA**, escribe para cada una de las siguientes preguntas la **ÚNICA** afirmación correcta (a, b, c ó d). Una pregunta mal contestada NO puntúa negativo.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A														
puntos según nº aciertos →			0.0	0.2	0.4	0.6	0.8	1,0	1,2	1,4	1,6	1.8	2.0	2.0



APELLIDOS \_\_\_\_\_ NOMBRE \_\_\_\_\_

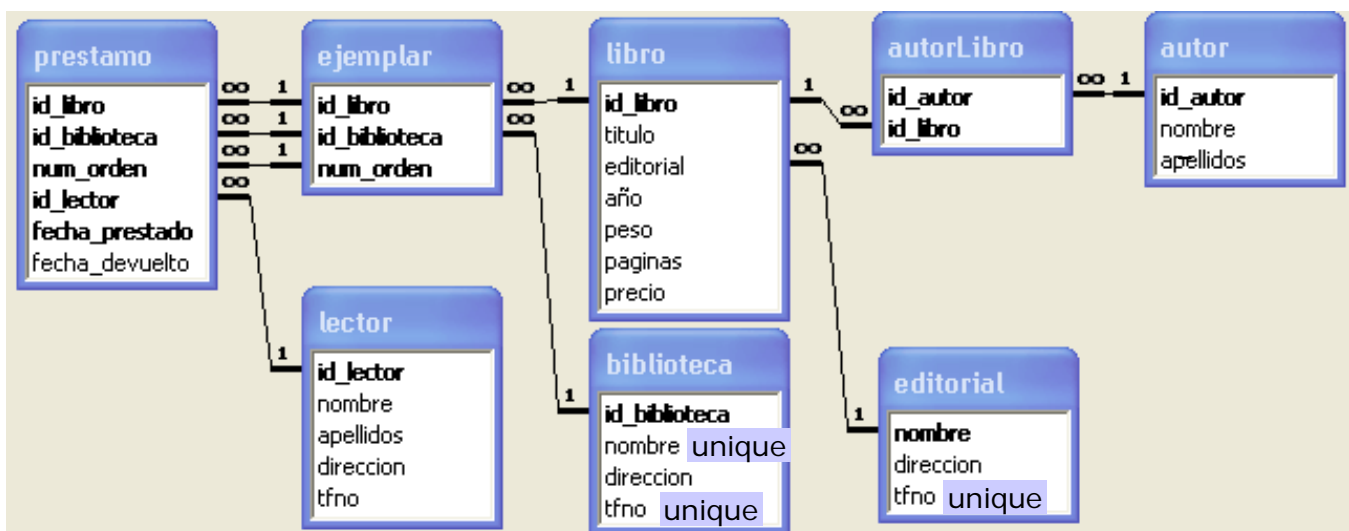
## Instrucciones

- Tiempo de esta parte: 2 horas y media.
- Escribe en esta hoja tu nombre y la respuesta a la pregunta 2. Escribe tu nombre **en todas las hojas** que uses.
- Puedes utilizar lápiz.
- Escribe las respuestas a los ejercicios **siguiendo el mismo orden** que en el enunciado.

## PARTE 2: Ejercicios

Una **red de pequeñas bibliotecas** desarrolla un sistema para la consulta de sus libros, de la localización de éstos y para la gestión de los préstamos. Los lectores pueden tomar prestados ejemplares de libro de cualquiera de las bibliotecas de la red.

**Esquema de la BD** (los atributos en **negrita** de cada tabla forman su clave primaria)



### autor

- Contiene un identificador numérico asignado a los autores, su nombre y apellidos. No se admite que haya nulos en ninguno.

### autorLibro

- Contiene los autores de cada libro. Un libro puede tener, cero (anónimo), uno o varios autores. Un autor puede serlo de varios libros. El identificador del autor es clave extranjera a la tabla autor y el del libro clave extranjera a la tabla libro.

### editorial

- El nombre y dirección son cadenas de caracteres. El teléfono debe ser único para cada editorial. Los tres campos deben ser obligatorios.

### libro

- Se refiere a una publicación y no a un ejemplar concreto. El identificador del libro es el ISBN, una cadena de 10 o 13 números.

- El título es una cadena de caracteres. La editorial es una clave extranjera a la tabla editorial e indica la editorial del libro. El año es el de publicación de la obra y el peso, número de páginas y precio son tres enteros positivos.

#### **biblioteca**

- Cada fila tiene información de una de las bibliotecas de la red. Nombre y teléfono son dos claves candidatas.

#### **ejemplar**

- Cada ejemplar de libro se identifica por el libro, la biblioteca a la que pertenece y un número de orden. Éste último permite distinguir los diferentes ejemplares de una misma obra de una biblioteca. Por ejemplo, la biblioteca 3 puede tener dos ejemplares de “el perfume”, cada uno tendrá un número de orden diferente, por ejemplo el uno y el dos.
- El identificador de libro es clave extranjera que referencia a la tabla libro y el identificador de biblioteca otra clave extranjera que referencia a la tabla biblioteca e indica a qué biblioteca pertenece el ejemplar.

#### **préstamo**

- Recoge los ejemplares de libro que se han prestado a los lectores. Un lector ha podido tomar en préstamo el mismo ejemplar en diferentes ocasiones. Por ello la clave primaria de la tabla incluye la fecha de préstamo.
- Hay que tener en cuenta que id\_libro, id\_biblioteca y num\_orden conforman una clave extranjera que referencia a la tabla ejemplar. Por tanto identifican el ejemplar de libro concreto del que se trata.
- Por otro lado id\_lector es una clave extranjera a la tabla lector. Es el lector que tomó el libro prestado.
- El atributo fecha\_prestado contiene la fecha en la que el lector tomó el libro en préstamo.
- Fecha\_devuelto **valdrá nulo si el lector tiene todavía el libro en préstamo**. Cuando devuelva el libro se asignará la fecha de devolución a este campo.

2. (0,5 puntos) Completa la siguiente **instrucción SQL** que crea la tabla **préstamo**. Añade los tipos y restricciones de integridad que faltan según la descripción inicial de las tablas.

```
CREATE TABLE prestamo (
    id_libro    VARCHAR(13)
    id_biblioteca
    num_orden
    id_lector
    fecha_prestado
    fecha_devuelto

);
```

3. (0,5 puntos) Escribe una **instrucción sql** que cree la tabla con el esquema: **Coste (lector, precioTotal)** con una fila para cada lector de la BD que tenga ejemplares en préstamo. Para cada lector se tendrá su identificador junto al precio total del material en préstamo.

4. (0,5 puntos) Escribe la **instrucción sql** necesaria para modificar la tabla **Coste**, creada en la pregunta anterior, sumando al precioTotal de las filas existentes el de los **ejemplares de libro** devueltos **durante** el último año de más de 500 páginas.
5. (0,5 puntos) Escribe la **instrucción sql** necesaria para incluir en la tabla **Coste**, creada en una pregunta anterior, aquellos lectores que **no** tienen libros en préstamo pero sí tienen libros devueltos **durante el último año (últimos 12 meses)** de más de 500 páginas. Asígnales como precio total el de estos libros.
6. (0,5 puntos) Escribe una **consulta sql** que muestre el título del libro junto al nombre y apellido del lector que lo tomó en préstamo para todos aquellos préstamos realizados durante el año actual de la editorial Ra-Ma. No queremos que haya repeticiones en el resultado, pero sólo queremos que aparezca, por ejemplo, **distinct** si es necesario.
7. (0,5 puntos) Crea una **vista** llamada **PrecioPréstamos** que muestre por cada biblioteca (**id\_biblioteca**), el precio total de los ejemplares que tenga actualmente en préstamo.
8. (0,5 puntos) Escribe una **vista** llamada **SemiVacías** que obtenga el identificador de las bibliotecas tales que el precio total de los ejemplares actualmente en préstamo supera la mitad del precio total de los libros de la biblioteca (de todos, estén o no prestados). Ten en cuenta que puedes utilizar la vista anterior para la definición de esta nueva vista.
9. (1 punto) Escribe una **consulta sql** que obtenga los nombres de las bibliotecas que tienen (prestado o no) al menos un ejemplar de cada uno de los libros que tienen a Alejandro Dumas como autor o coautor.
10. (1 punto) Escribe una **consulta sql** que obtenga los datos de aquellos autores que **no** tienen en una misma biblioteca varias de sus obras (ejemplares de libros distintos).
11. (1 punto) Escribe una **consulta sql** que obtenga para cada libro *escrito por un solo autor* (hablamos de obras y no de ejemplares) el número de veces que se ha sacado en préstamo, de cuántas bibliotecas y por cuántos lectores diferentes. Hay que tener en cuenta que algunos libros no se habrán sacado nunca en préstamo, pero también tendrán que aparecer en la solución. De cada libro se incluirá su título y editorial junto al nombre y apellidos de su único autor.
12. (1 punto) Escribe una **consulta sql** que obtenga para cada libro titulado “cien años de soledad” con autor Gabriel García Márquez, el identificador del libro junto al nombre de cada biblioteca donde está y junto al número de ejemplares disponibles (**no prestados**) de ese libro en esa biblioteca. Si no hay ejemplares disponibles del libro en una biblioteca no nos interesa que salga en el resultado esa pareja libro-biblioteca.
13. (0,5 puntos) Escribe una **consulta en álgebra relacional** que muestre el nombre y apellidos de los lectores que nunca hayan tomado en préstamo obras de la biblioteca ‘El cubo’.

ALGUNAS FUNCIONES (puede que no hagan falta todas)

Nombre(argumentos)	Significado	Ejemplos
<b>sysdate</b>	Devuelve la fecha actual	<b>sysdate</b> → '16-jun-2008'
<b>to_char(&lt;fecha&gt;, &lt;formato&gt;)</b>	Convierte la fecha a cadena según el formato especificado	<b>to_char(sysdate,'YYYY')</b> → '2008' <b>to_char(sysdate,'YYYY/MM/DD')</b> → '2008/06/16'
<b>to_date(&lt;cadena&gt;,&lt;formato&gt;)</b>	Convierte la cadena dada según el formato especificado a fecha	<b>to_date('16-06-08','DD-MM-YY')</b> → [fecha de hoy]
<b>substr(&lt;cadena&gt;,&lt;inicio&gt;,&lt;avance&gt;)</b>	Extrae una subcadena de <cadena> de longitud <avance> empezando en el carácter <inicio>.	<b>substr('mi cadena', 5, 3)</b> → 'ade'

2.

```
CREATE TABLE prestamo (  
  id_libro      VARCHAR(13) not null,  
  id_biblioteca varchar(15) not null,  
  num_orden     int not null,  
  id_lector     varchar(15) not null,  
  fecha_prestado date not null,  
  fecha_devuelto date,  
  primary key   (id_libro, id_biblioteca, num_orden,  
                 id_lector, fecha_prestado),  
  foreign key   (id_libro, id_biblioteca, num_orden) references  
                 ejemplar(id_libro, id_biblioteca, num_orden),  
  foreing key   (id_lector) references lector(id_lector)  
);
```

3. (0,5 puntos) Escribe una instrucción sql que cree la tabla con el esquema: **Coste** (lector, precioTotal) con una fila para cada lector de la BD que tenga ejemplares en préstamo. Para cada lector se tendrá su identificador junto al precio total del material en préstamo.

**create table coste as**

```
select P.id_lector as lector, sum(L.precio) as precioTotal  
from   prestamo P join  
       libro L on P.id_libro=L.id_libro  
where  p.fecha_devuelto is null  
group by P.id_lector
```

4. (0,5 puntos) Escribe la instrucción sql necesaria para modificar la tabla **Coste**, creada en la pregunta anterior, sumando al precioTotal de las filas existentes el de los ejemplares de libro devueltos por el lector durante el último año de más de 500 páginas.

**update coste**

```
set  precioTotal = precioTotal +  
      (select sum(L.precio)  
       from   libro L join  
              prestamo P on L.id_libro=P.id_libro  
       where  coste.lector=P.id_lector and L.paginas>500 and  
              p.fecha_devuelto >= sysdate-365)
```

5. (0,5 puntos) Escribe la instrucción sql necesaria para incluir en la tabla **Coste**, creada en una pregunta anterior, aquellos lectores que no tienen libros en préstamo pero sí tienen libros devueltos durante el último año (últimos 12 meses) de más de 500 páginas. Asíñales como precio total el de estos libros.

**insert into coste**

```
select p.id_lector, sum(l.precio)  
from   libro l join  
       prestamo p on l.id_libro=p.id_libro  
where  p.fecha_devuelto >= sysdate-365 and l.paginas>500 and  
       p.id_lector not in (select id_lector from prestamo where fecha_devuelto is null)  
group by p.id_lector
```

6. (0,5 puntos) Escribe una **consulta sql** que muestre el título del libro junto al nombre y apellido del lector que lo tomó en préstamo para todos aquellos préstamos realizados durante el año actual de la editorial Ra-Ma. No queremos que haya repeticiones en el resultado, pero sólo queremos que aparezca distinct si es necesario.

```
select distinct L.titulo, LC.nombre, LC.apellido
from ( libro L join
      prestamo P on L.id_libro=P.id_libro) join
      lector LC on P.id_lector=LC.id_lector
where L.editorial="Ra-Ma" and fecha_prestado like '%' || to_char(sysdate,'yyyy')
```

-- Otra forma para la última condición year(fecha\_prestado)=year(sysdate)

7. (0,5 puntos) Crea una **vista** llamada **PrecioPréstamos** que muestre por cada biblioteca (id\_biblioteca), el precio total de los *ejemplares actualmente en préstamo*.

```
create view precioPréstamos as
select P.id_biblioteca, sum(L.precio) as precioTotal
from libro L join
      prestamo P on E.id_libro=P.id_libro
where P.fecha_devuelto is null
group by P.id_biblioteca
```

8. (0,5 puntos) Escribe una **vista** llamada **SemiVacías** que obtenga el identificador de las bibliotecas tales que el precio total de los *ejemplares actualmente en préstamo* supera la mitad del precio total de los libros de la biblioteca (de todos, estén o no prestados). Ten en cuenta que puedes utilizar la vista anterior para la definición de esta nueva vista.

```
create view SemiVacías as
select E.id_biblioteca
from ejemplar E join
      libro L on E.id_libro=L.id_libro
group by E.id_biblioteca
having sum(L.precio)/2 <
```

```
(select PP.precioTotal
from precioPréstamos PP
where PP.id_biblioteca= E.id_biblioteca)
```

Haciéndolo al revés parece más simple, pero mantenemos la correlación (subrayada):

```
create view SemiVacías as
select PP.precioTotal
from precioPréstamos PP
where PP.precioTotal >
```

```
select sum(L.precio)/2
from ejemplar E join
      libro L on E.id_libro=L.id_libro
where PP.id_biblioteca = E.id_biblioteca
```

9. (1 punto) Escribe una **consulta sql** que obtenga los nombres de las bibliotecas que tienen (prestado o no) al menos un ejemplar de cada uno de los libros que tienen a Alejandro Dumas como autor o coautor.

```
select B.nombre  
from biblioteca B  
where not exists
```

```
(select AL.id_libro  
 from autorLibro AL join  
      autor A on AL.id_autor=A.id_autor  
 where A.nombre="Alejandro" and A.apellidos="Dumas"  
 minus  
 select E.id_libro  
 from ejemplar E  
 where E.id_biblioteca=B.id_biblioteca)
```

10. (1 punto) Escribe una **consulta sql** que obtenga los datos de aquellos autores que **no** tienen en una misma biblioteca varias de sus obras (ejemplares de libros distintos).

```
select A.*  
from autor A  
where not exists
```

```
(select E.id_biblioteca  
 from autorLibro AL join  
      ejemplar E on AL.id_libro=E.id_libro  
 where AL.id_autor=A.id_autor  
 group by E.id_biblioteca  
 having count(distinct E.id_libro) > 1)
```

11. (1 punto) Escribe una **consulta sql** que obtenga para cada libro *escrito por un solo autor* (hablamos de obras y no de ejemplares) el número de veces que se ha sacado en préstamo, de cuántas bibliotecas y por cuántos lectores diferentes. Hay que tener en cuenta que algunos libros no se habrán sacado nunca en préstamo, pero también tendrán que aparecer en la solución. De cada libro se incluirá su título y editorial junto al nombre y apellidos de su único autor.

```
select L.titulo, Leditorial, A.nombre, A.apellidos,  
      count(P.id_libro), count(distinct P.id_biblioteca), count(distinct P.id_lector)  
from ((libro L join  
      autorLibro AL on L.id_libro=AL.id_libro) join  
      autor A on AL.id_autor=A.id_autor) left outer join  
      prestamo P on L.id_libro=p.id_libro  
group by L.id_libro, L.titulo, Leditorial, A.nombre, A.apellidos  
having count(distinct A.id_autor)=1
```

```
select L.titulo, Leditorial, A.nombre, A.apellidos,  
      count(P.id_libro), count(distinct P.id_biblioteca), count(distinct P.id_lector)  
from ((libro L join  
      autorLibro AL on L.id_libro=AL.id_libro) join  
      autor A on AL.id_autor=A.id_autor) left outer join  
      prestamo P on L.id_libro=p.id_libro  
where L.id_libro IN (select AL.id_libro  
                    from autorLibro AL
```

```

group by AL.id_libro join
having count( AL.id_autor)=1)
group by L.id_libro, L.titulo, L.editorial, A.nombre, A.apellidos

```

12. (1 punto) Escribe una **consulta sql** que obtenga para cada libro titulado “cien años de soledad” con autor Gabriel García Márquez, el identificador del libro junto al nombre de cada biblioteca donde está y junto al número de ejemplares disponibles (**no prestados**) de ese libro en esa biblioteca. Si no hay ejemplares disponibles del libro en una biblioteca no nos interesa que salga en el resultado esa pareja libro-biblioteca.

```

select AL.id_libro, B.nombre, count(*)
from (((autor A join
autorLibro AL on A.id_autor=A.id_autor) join
libro L on AL.id_libro=L.id_libro) join
ejemplar E on L.id_libro=E.id_libro) join
biblioteca B on E.id_biblioteca=B.id_biblioteca
where L.titulo='cien años de soledad' and A.nombre='Gabriel' and
A.apellidos='García Márquez' and
not exists

```

```

(select *
from prestamo P
where P.fecha_devuelto is null and
P.id_libro=E.id_libro and
P.id_biblioteca=E.id_biblioteca and
P.num_orden=E.num_orden)

```

```

group by AL.id_libro, B.nombre

```

13. (0,5 puntos) Escribe una **consulta en álgebra relacional** que muestre el nombre y apellidos de los lectores que nunca hayan tomado en préstamo obras de la biblioteca ‘El cubo’.

```

B_elCubo (id_bib) ← πid_biblioteca ( σnombre='El cubo'(biblioteca) )
Pr_elCubo ← πid_lector ( préstamo |X|id_biblioteca=id_bib B_elCubo )
Todos_Lect ← πid_lector (Lector)
Lect_no_elCubo (id_lect)← Todos_Lect – Pr_elCubo
Resultado ← πnombre, apellidos (Lect_no_elCubo |X|id_lect=id_lector Lector)

```

No se pide la solución SQL. No obstante, esta es:

```

select L.nombre, L.apellidos
from lector L
where L.id_lector not in

```

```

(select P.id_lector
from prestamo P join
biblioteca B on P.id_biblioteca=B.id_biblioteca
where B.nombre='El cubo')

```