EXAMEN FINAL DE PRÁCTICAS DE BASES DE DATOS

18 de junio de 2003

Nombre:

**LEA ATENTAMENTE LAS PREGUNTAS Y RESUELVA CADA
EJERCICIO COMO SE INDIQUE (EN CASO CONTRARIO EL
EJERCICIO NO PUNTUARÁ)
CADA EJERCICIO VALE UN PUNTO**

Dadas las siguientes tablas:

Libro (signatura, autor, titulo, editor, clase, prestado)

Usuario (carnet, nombre, dirección, fecha_ingreso)

Clase (clave, tiempo_de_prestamo)

Prestamo (signatura, carnet, fecha_inicio)

Con las siguientes claves ajenas:

Libro.clase → Clase

Prestamo.signatura → Libro

Prestamo.carnet → Usuario

Y las siguientes restricciones:

Prestado debe tomar por defecto el valor 1

fecha_ingreso es un campo obligatorio

Realizar en SQL los siguientes ejercicios:

1. Crear las tablas tal y como se haría en SQL y, una vez creadas:
Agregar a la tabla Usuario el campo sexo de un solo carácter
Agregar a la tabla Libro el campo índice de tipo entero

```
CREATE TABLE clase(  
Clave char PRIMARY KEY,  
Tiempo_de_prestamo NUMBER);
```

```
CREATE TABLE libro(  
signatura varchar2(10) PRIMARY KEY,  
autor varchar2(50),  
titulo varchar2(50),  
editor varchar2(30),  
prestado NUMBER DEFAULT 1,  
clase char references clase);
```

```
CREATE TABLE Usuario(  
Carnet varchar(10) PRIMARY KEY,  
Nombre varchar2(50),  
Dirección varchar2(50),  
fecha_ingreso DATE NOT NULL  
);
```

```
CREATE TABLE Prestamo (  
Signatura varchar2(10),  
Carnet varchar(10),  
fecha_inicio date,  
PRIMARY KEY(signatura, carnet),  
FOREIGN KEY (signatura) REFERENCES libro,  
FOREIGN KEY (carnet) REFERENCES usuario);
```

```
alter table usuario add sexo CHAR;  
alter table Libro add indice INTEGER;
```

2. Dar un ejemplo de inserción en cada una de las tablas. Modificar la clase de los préstamos de tipo 3 de forma que la duración total sea de 180 días en vez de 365. Borrar aquellos usuarios cuyo número de carnet sea mayor de 3

Tabla Libro

```
INSERT INTO libro VALUES ('INF01', 'Pepe', 'Los problemas', 'MG', 1,1);
```

```
INSERT INTO libro VALUES('INF02', 'Paco', 'Psicología', 'LL', 1,1);
```

```
INSERT INTO libro VALUES ('FIS01', 'Pepe', 'Fundamentos físicos', 'RO-MO', 2,1);
INSERT INTO libro VALUES('MAT02','Lola', 'Cálculo', 'MM', 3,1);
INSERT INTO libro VALUES ('MAT05', 'Susi', 'Matemáticas discretas', 'MM', 2,1);
```

Tabla Usuario

```
INSERT INTO usuario VALUES(1, 'Loli', 'Alarcos', '1/1/1989', 'm');
INSERT INTO usuario VALUES(2, 'Carmen', 'Alarcos', '7/8/1999', 'm');
INSERT INTO usuario VALUES(3, 'Alejandro', 'Postas', '16/2/2001', 'v');
INSERT INTO usuario VALUES(4, 'Elena', 'Calatrava', '30/4/2001', 'm');
INSERT INTO usuario VALUES(5, 'Miguel', 'Encinas', '23/5/1986', 'v');
```

Tabla Clase

```
INSERT INTO clase VALUES(1, 15);
INSERT INTO clase VALUES(2, 30);
INSERT INTO clase VALUES(3, 365);
```

Tabla prestamo

```
INSERT INTO prestamo VALUES('INF01', 1, '1/6/2002');
INSERT INTO prestamo VALUES('INF02', 1, '15/6/2002');
INSERT INTO prestamo VALUES('FIS01', 2, '7/5/2002');
INSERT INTO prestamo VALUES('MAT02', 3, '21/8/2001');
INSERT INTO prestamo VALUES('MAT05', 3, '21/6/2002');
```

UPDATE clase

```
SET tiempo_de_prestamo=180
```

```
Where clave=3;
```

```
DELETE FROM usuario WHERE carnet>3;
```

3. Seleccionar, alfabéticamente, el título de todos los libros del área de informática (signatura INFXX) dónde XX actúan como comodines, es decir, podrán tomar dos valores cualesquiera (por ejemplo, INF00, INF01, INF99, ...)

```
SELECT titulo
```

```
FROM libro
WHERE signatura LIKE 'INF%'
ORDER BY titulo;
```

Salida:

Los problemas
Psicología

4. Seleccionar la fecha de inicio de los préstamos que tienen los usuarios que viven en la calle Alarcos

```
SELECT fecha_inicio
FROM prestamo, usuario
WHERE prestamo.carnet=usuario.carnet
AND usuario.dirección='Alarcos';
```

5. Seleccionar la fecha de inicio de los préstamos de los libros de la editorial MM que tienen los usuarios que ingresaron después del año 2000

```
SELECT fecha_inicio
FROM prestamo, libro, usuario
WHERE prestamo.carnet=usuario.carnet
AND prestamo.signatura=libro.signatura
AND editor='MM'
AND fecha_ingreso>'31/12/2000';
```

Salida:

21/08/01
21/06/02

6. Devolver, para cada editor, la cantidad de libros prestados que tiene

```
SELECT count(libro.signatura), editor
FROM libro, prestamo
WHERE prestamo.signatura=libro.signatura
GROUP BY editor;
```

Salida:

COUNT(LIBRO.SIGNATURA) EDITOR

1 LL

1 MG

2 MM

7. Seleccionar el nombre de aquellos usuarios cuyo número de carnet es menor que diez veces la clase de cualquier libro que tenga prestado

Select nombre

From usuario

Where carnet < ANY (select clase*10 from libro, prestamo where prestamo.signatura=libro.signatura and prestamo.carnet=usuario.carnet);

8. Seleccionar la fecha de ingreso de aquellos usuarios tales que todos los libros que tienen prestados tienen una signatura menor que su número de carnet

Select fecha_ingreso

From usuario where carnet > ALL (select signatura from prestamo where prestamo.carnet=usuario.carnet);

9. Crear una vista (libro_persona) que contenga los datos de los libros (signatura y autor) y de las personas (carnet, nombre y sexo) que tienen prestados los libros

Insertar, si es posible, una fila de valores en la vista

```
CREATE VIEW libro_persona
```

```
AS SELECT libro.signatura, libro.autor, usuario.carnet, usuario.nombre, sexo
```

```
FROM libro, usuario, prestamo
```

```
WHERE libro.signatura=prestamo.signatura AND usuario.carnet=prestamo.carnet;
```

No se puede insertar porque no está incluido el campo fecha_ingreso de usuario que está forzado a ser NOT NULL.

10. Realizar un programa PL/SQL que, mediante un bucle, muestre los libros que son de informática y tienen un valor par en el campo indice.