

Facultad de Informática. Ingeniería en Informática / del Software / de Computadores.
Examen de Bases de datos. Curso 2016-2017. 5 de septiembre de 2017

NIF/NIE

Apellidos y nombre

Las soluciones deben entregarse en las hojas que se os han entregado y dentro del espacio designado para ellas. Podéis utilizar papel de sucio, pero las respuestas definitivas deben indicarse en las hojas de los enunciados. NO se puede usar lápiz ni boli rojo.

Ejercicio 1 (3 puntos). Una cadena de cines requiere implementar una base de datos relacional que cumpla los siguientes requisitos:

- La cadena posee cines en diferentes distritos del país (como máximo una por distrito). De cada cine almacenaremos el distrito y la dirección.
- Cada cine dispone de varias salas de proyección, como mínimo 2 y como máximo 6. Cada una de ellas viene identificada por un número único en cada cine pero en diferentes cines puede haber salas con el mismo número. De cada una de las salas debe conocerse su aforo.
- Para gestionar los pases de las diferentes películas en cartelera, se deben almacenar en la base de datos las películas que se proyectan en las salas. De las películas se necesita saber el título, que las identifica de forma única, el año de estreno, la categoría a la que pertenecen (comedia, drama o musical). Cada sala puede tener varios pases.
- Cada pase se identifica por la hora de inicio. Todas las películas se proyectan en, al menos, un pase de una sala de un cine de la cadena. También, nos interesa saber si la proyección de una película en un pase de una sala se proyecta en versión original o doblada.

- a. (1.5 puntos) Se pide diseñar el esquema conceptual con el modelo entidad-relación especificando las restricciones de participación mínimo-máximo y cardinalidad.

- b. (1 puntos) Traducir dicho esquema a relacional e incluir las restricciones de integridad referencial.

- c. (0,5 puntos) Indicar las restricciones no reflejadas en el esquema relacional.

Ejercicio 2 (6 puntos). Dadas las siguientes relaciones:

Asignatura (Codigo, Nombre, NumCreditos)

Alumno (DNI, Nombre, Edad, NumCredAprobados)

Matricula (DNIAlumno, CodAsignatura, Nota)

a. (3 puntos) Escribe las siguientes consultas en lenguaje SQL:

- (0,5 puntos) Mostrar el DNI de los alumnos (sin repeticiones) de más de 30 años matriculados en una asignatura de 6 créditos.

- (0,5 puntos) Mostrar el nombre de las asignaturas de 12 créditos con más de 300 alumnos.

- (0,5 puntos) Mostrar para todos los alumnos el nombre y el número de asignaturas en las que está matriculado. Si algún alumno no está matriculado en ninguna asignatura se debe de mostrar un 0.

- (0,75 puntos) Mostrar el DNI de los alumnos que solo tienen asignaturas matriculadas de 6 créditos.

- (0,75 puntos) Mostrar el nombre del alumno de mayor edad matriculado en la asignatura de código 123.

- b. (1,5 puntos) Crea un procedimiento almacenado que reciba por argumento el DNI de un alumno y muestre todas las asignaturas en las que está matriculado el alumno (nombre, número de créditos y nota), el número total de créditos matriculados y la nota media del expediente de dicho alumno. Para calcular la nota media se suman la $\text{nota} \times \text{numCreditosAsignatura}$ de todas las asignaturas aprobadas ($\text{nota} \geq 5$) y se divide por el número de créditos en los que está matriculado el alumno. Si el alumno no está matriculado en ninguna asignatura se mostrará el siguiente mensaje 'El alumno xxx no se ha matriculado de ninguna asignatura'.

- c. (1,5 puntos) Escribe un disparador que mantenga el número de créditos aprobados de la tabla Alumno actualizado. Es decir, cada vez que se modifique una nota de la tabla Matricula se comprobará si la nota es igual o superior a 5 y si lo es se actualizarán los créditos aprobados del alumno.

Ejercicio 3 (1 punto). Dada la tabla Matricula(DNI, Asignatura, Creditos) vacía considera la ejecución de la siguiente lista de instrucciones SQLDeveloper asumiendo que **autocommit=off**:

```
savepoint paso_uno;
INSERT INTO MATRICULA VALUES ('123456789X', 'BBDD', 6);

-- paso 1 --
savepoint paso_dos;
update MATRICULA
  set Creditos= Creditos+ 1
where DNI= '123456789X' and asignatura='BBDD';

-- paso 2 --
rollback to savepoint paso_dos;

-- paso 3 --
update MATRICULA
  set Creditos= Creditos+ 2
where DNI= '123456789X' and asignatura='BBDD';

-- paso 4 --
rollback;

-- paso 5 --
INSERT INTO MATRICULA VALUES ('123456789X', 'BBDD', 12);
update MATRICULA
  set Creditos= Creditos+ 3
where DNI= '123456789X' and asignatura='BBDD';

-- paso 6 --
savepoint paso_tres;
commit;

-- paso 7 --
create table Expediente(DNI varchar(20) PRIMARY KEY, MatriculaTotal number(5));
Insert into Expediente values('00000000P', 45);
rollback to savepoint paso_tres;

-- paso 8 --
select * from Expediente where DNI = '00000000P';
rollback;

-- paso 9 --
```

- Describe qué valor tiene `Creditos` para la fila `'123456789X'` exactamente en el momento indicado por cada comentario `-- paso N --` (aunque todavía no sea un valor definitivo).
- Indica con qué instrucción empieza cada una de las transacciones de la secuencia.
- ¿Se produce algún error? Si lo hay, indica en qué instrucción y por qué.
- ¿Qué tablas quedan al final de la ejecución?