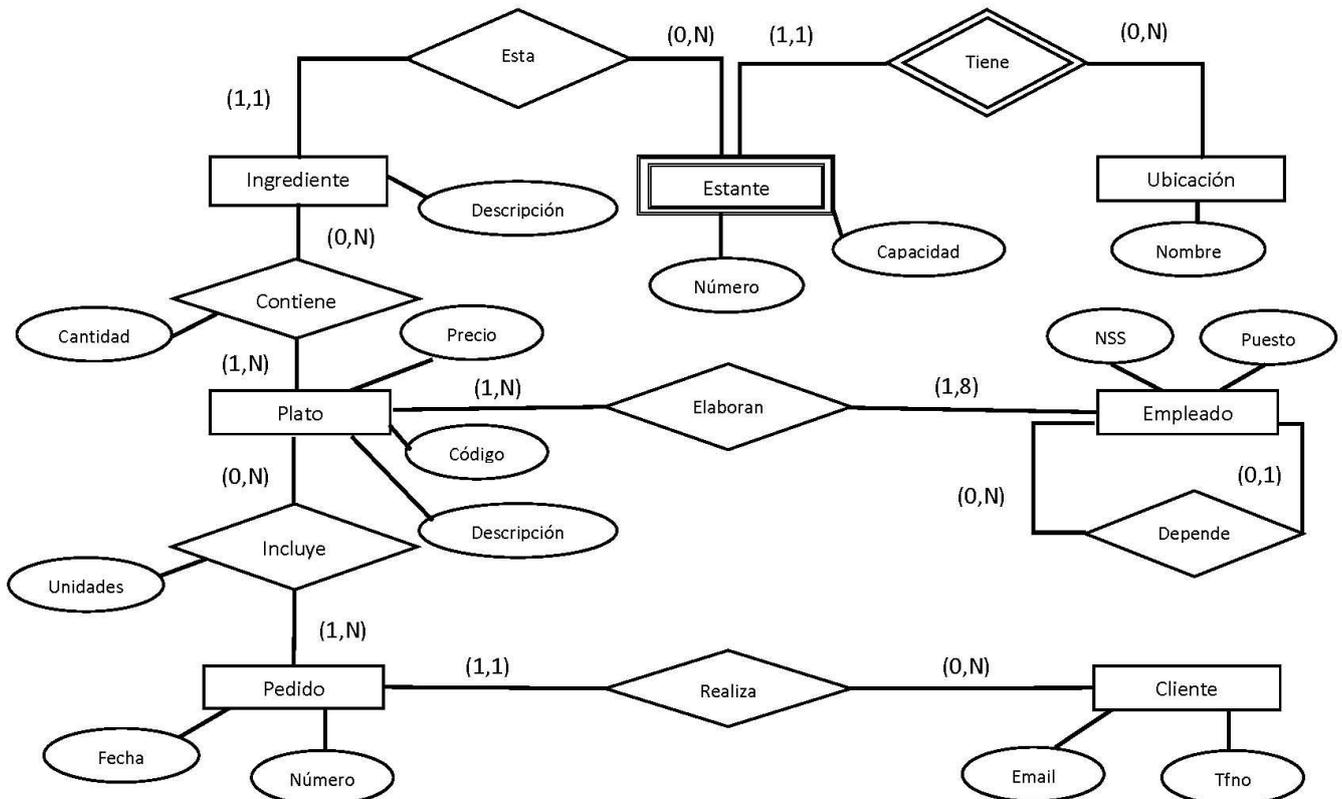


Examen de Bases de datos Grado de Ingeniería en Informática Septiembre 2014

- 1) (3 puntos) Necesitamos una base de datos para gestionar una empresa de preparación de comidas. En la empresa trabajan empleados de los que se necesita saber su nombre, número de la SS, puesto que ocupan (cocinero, pinche, limpiador...) y al mando de que empleado están. La empresa elabora diferentes platos, cada uno de los cuales tiene un código, una descripción (cocido madrileño, torrijas, conejo a la cazadora...) y un precio. Para la elaboración de cada plato se necesita saber los ingredientes necesarios (identificables mediante su nombre), así como la cantidad de cada uno de ellos y dónde encontrarlos en las cocinas. Los ingredientes se encuentran dispuestos en diferentes ubicaciones que se identifican mediante un nombre (cámara frigorífica A, cámara frigorífica B, despensa principal, congelador...) y están divididas en estantes numerados de los que se debe conocer su capacidad. Naturalmente, dos ubicaciones distintas pueden tener estantes con la misma numeración. Además, necesitamos saber que empleados son capaces de preparar cada plato. Todos los empleados deben ser capaces de participar en la elaboración de al menos un plato y no exceden de ocho.

También registraremos a los clientes de los que se sabrá su número de teléfono, email, así como los pedidos que han realizado, los cuales se identifican mediante un número. Cada pedido contendrá la fecha en la que se realizó, los platos encargados y la cantidad de cada uno de ellos.

- (1,5 puntos) Se pide diseñar el esquema conceptual con el modelo entidad-relación especificando las restricciones de participación mínimo-máximo.



- (1 puntos) Traducir dicho modelo a modelo relacional e incluir las reglas de Integridad referencial

Tipos de entidades:

Empleados(NSS, Puesto, Jefe*) Se ha incluido la relación Dependencia

Clientes(Email, Tfno)

Pedidos(Número, Fecha, email) Se ha incluido la relación Realiza

Platos(Código, Descripción, Precio)

Ingredientes(Descripción, Nombre, Numero) Se ha incluido la relación Esta
Ubicaciones(Nombre)
Estantes(Nombre,Número,Capacidad) Se ha incluido la relación Tiene

Tipos de relaciones:

Incluye(Número, Código, Unidades)

Elaboran(NSS, Código)

Contiene(Código,Descripción,Cantidad)

- (0,5 puntos) Indicar las restricciones no reflejadas en el modelo relacional

Las restricciones no reflejadas en el modelo relacional son:

-Un empleado elabora un máximo de 8 platos y como mínimo uno

Restricciones de participación total en un N a N

- Un plato contiene al menos un ingrediente

- Un pedido incluye al menos un plato

- Un plato es elaborado por al menos un empleado.

2) (2,5 puntos) Dada las relaciones:

Pacientes(Historial, SS, Nombre, Domicilio, Teléfono)

Ingresos(ID, Historial, CódMédico, Fecha, Diagnóstico)

Médicos(CódMédico, Nombre, Especialidad)

Escribir las siguientes consultas:

- (0,5 punto) AR: Mostrar los nombres de médicos que han atendido pacientes que hayan sido ingresados.

• $\Pi_{Nombre}(Medicos \bowtie Ingresos)$

- (0,5 punto) AR: Mostrar el diagnóstico y el nombre del médico que atendió a los pacientes con nombre Pedro García

$\Pi_{Diagnostico,Nombre}(Medicos \bowtie (Ingresos \bowtie (\Pi_{Historial}(\sigma_{Nombre='Pedro\ Garcia'}(Pacientes))))))$

- (0,5 punto) SQL: Mostrar el número de médicos de la especialidad Pediatría.

```
Select count(*)  
from Medicos  
where Especialidad='Pediatría'
```

- (0,5 puntos) SQL: Mostrar los numero de historia de los pacientes que han sido ingresados en alguna ocasión y nunca han sido atendidos por un médico de 'Pediatría'

```
Select Historial  
from Ingresos I  
  where not exists (select codMedico  
                   from Ingresos natural join Médicos  
                   where historia = I.historia  
                   and especialidad='Pediatría')
```

- (0,5 puntos) SQL: Mostrar los códigos, nombres y el número de ingresos asistidos de los médicos que hayan atendido el mayor número de ingresos.



```
Select CodMedico, Nombre, count(*)
from Ingresos natural join Médicos
group by CodMedico, Nombre
having count(*) >= ALL (select count(*)
                        from Ingresos
                        group by CodMedico)
```

3) (2,5 puntos) Dado el esquema:

Empleados(DNI, Nombre, codDept, Salario)
RegistroCambios (IdCambio, Usuario, Fecha, salAnterior, salActual)

- (1 punto) Implementar un trigger que registre en la tabla RegistroCambios cualquier modificación que se produzca en el salario de un empleado, indicando el usuario, la fecha en la que se realizó y el salario anterior y posterior a la modificación. El identificador se obtendrá de una secuencia denominada SEQCambios.

```
CREATE OR REPLACE TRIGGER RegCambios
AFTER UPDATE OF SALARIO ON EMPLEADOS
FOR EACH ROW
BEGIN
INSERT INTO RegistroCambios VALUES
(SEQCambios.NEXTVAL,USER(),SYSDATE(),:old.salario, :new.salario)
END;
```

- (1,5 punto) Escribir un procedimiento almacenado que liste para cada departamento el nombre y salario de cada empleado cuyo salario sea inferior a la media de su departamento. Incluir el total de dichos salarios por departamento

```
create or replace procedure SalrioMedio is

TB constant varchar2(1):=CHR(9);
lastDept EMP.CODDEPT%TYPE:='';
total number(7,2):=0;

CURSOR Datos IS
SELECT Nombre, codDept, salario
FROM Empleados E1
where salario < (select AVG(salario) from Empleados E2 where
E1.codDept=E2.codDept)
order by codDept;

BEGIN
dbms_output.put_line('Codigo Dept' ||TB|| rpad('Nombre',50,' ') ||TB||
'Salario');
dbms_output.put_line(rpad('-',80,'-'));

FOR rDatos IN Datos LOOP
if lastDept<>rDatos.CodDept and Total >0 then
dbms_output.put_line('Total Depto:' ||lastDept ||round(Total,2));
Total:= rDatos.Salario;
lastDept:=rDatos.CodDept;
else
Total:= Total + rDatos.Salario;
end if;
dbms_output.put_line(rDatos.CodDept ||TB|| rpad(rDatos.nombre,50,' ') ||TB||
round(rDatos.Salario,2));
END LOOP;
END;
```

- 5) (1 punto) Describir la funcionalidad de los puntos de control (SAVEPOINT)
Es posible declarar puntos intermedios en una transacción que permiten guardar el trabajo realizado hasta el mismo y a partir de los cuales es posible continuar.

Un punto de control se establece mediante la instrucción SAVEPOINT <nombre>

Para retroceder una transacción hasta un punto de control se utiliza la sentencia ROLLBACK TO SAVEPOINT <nombre>

Cuando una transacción es retrocedida hasta un punto de control:

- Solo deshace las sentencias realizadas tras el punto de control.
- Preserva el savepoint correspondiente y los savepoints anteriores a este.
- Libera los bloqueos obtenidos tras el savepoint y preserva los obtenidos antes del mismo.
- Una transacción que se retrocede a un punto de control permanece activa.

- 6) (1 punto) Dada la relación $R(A,B,C,D,E)$ y el conjunto de dependencias funcionales $F = \{AB \rightarrow C, C \rightarrow D, E \rightarrow BC, E \rightarrow BD\}$, se pide determinar las posibles claves primarias.

No puede haber claves candidatas de un único atributo porque los atributos A y E no se encuentran en ningún consecuente. Al menos, estos dos atributos deben formar parte de cualquier superclave.

Examinamos su cierre

$$\{A,E\}^+ = \{A,B,C,D,E\}$$

Es clave superclave y la única clave candidata de dos atributos, por lo que es también clave primaria.