

BASES DE DATOS

Grado en Informática
Curso académico 2012/2013
(Convocatoria de febrero)

CALIFICACIÓN

Pregunta 1

Pregunta 2

Pregunta 3

Pregunta 4

Pregunta 5

Total

Examen Final (Teoría)

Indicaciones Generales:

- ✓ El examen tiene una duración de 150 minutos.
- ✓ Es obligatorio entregar este folio con las soluciones.

Pregunta 1 (3 puntos)

Proponga un modelo EE-R para un sistema informático que debe atender a los siguientes requisitos de información.

En una determinada granja se desea mantener la información correspondiente a la alimentación que se suministra a los animales que son explotados en la misma.

El control y seguimiento de la alimentación que se proporciona a cada uno de los animales de la granja tiene como objetivo el estudio y análisis de los resultados de la misma sobre la producción y beneficio que se obtiene de los animales. En este sentido, cada animal sigue una dieta alimenticia en base a una serie de criterios determinados por los veterinarios y gestores de la granja (necesidades de nutrientes dependiendo del tipo de animal, edad, etc., disponibilidad de los alimentos, coste, etc.).

La dieta seguida por cada uno de los animales de la granja puede ser variada a lo largo de su vida, siendo de interés para los gestores de la misma el conocimiento de esta información.

Cada dieta está compuesta por una serie de alimentos que son ingeridos por los animales en diferentes tomas a lo largo del día. En cada toma, cada uno de los animales ingiere unas cantidades determinadas en la dieta de uno o varios alimentos.

Los alimentos y las cantidades ingeridas de los mismos por cada uno de los animales en su dieta son determinados basándose en criterios veterinarios y económicos.

Se consideran además los siguientes supuestos semánticos en el problema:

SUPUESTO 1: Una dieta se establece para un animal y no para todos los animales de la misma especie. Si bien, una misma dieta puede ser seguida al mismo tiempo por varios animales de la granja.

SUPUESTO 2: Cada animal puede seguir varias dietas a la vez.

SUPUESTO 3: Se desea guardar información histórica sobre todas las dietas que han seguido los animales.

SUPUESTO 4: Cada uno de los animales tiene un código asociado que puede ir impreso bien en la oreja o bien en el lomo.

SUPUESTO 5: Se desea mantener información de los nutrientes sean o no necesitados por los animales de la granja.

SUPUESTO 7: Se entiende por una dieta al conjunto de alimentos que recibe el animal a lo largo del día, los cuales son ingeridos en diferentes tomas. En este sentido no se considerará, por simplicidad, regímenes alimenticios distribuidos en un tiempo superior a veinticuatro horas.

SUPUESTO 8: Las dietas están constituidas por un conjunto de alimentos o productos alimenticios que contienen los nutrientes que necesitan los animales en su alimentación.

Pregunta 2 (2 puntos)

Dada la siguiente extensión de R(A,B,C,D,E)

(a1,b2,c1,d3,e2)
(a2,b2,c3,d3,e4)
(a1,b3,c2,d1,e3)
(a2,b4,c5,d1,e5)

Indique las dependencias funcionales que no se aplican a R

- 1- A → C
2- C → A
3- E → A
4- B → D

- 5- D → E
6- E → B
7- C → B
8- B → A

Pregunta 3 (2 puntos)

Proponga un modelo conceptual para la siguiente implementación de una base de datos relacional.

```
/* Se crean las tablas del esquema propuesto */
CREATE TABLE ZonaUrbana (
    nombre_zona VARCHAR2(20) NOT NULL,
    od_zona LONG,
    CONSTRAINT pk_zon
    PRIMARY KEY (nombre_zona),
    CONSTRAINT ck_zon
    CHECK (nombre_zona = UPPER(nombre_zona)) );
```

```
CREATE TABLE Vivienda (
    calle VARCHAR2(40) NOT NULL,
    numero NUMBER(3) NOT NULL,
    tipo_vivienda VARCHAR2(1),
    codigo_postal NUMBER(5),
    metros NUMBER(5),
    od_vivienda LONG,
    nombre_zona VARCHAR2(20) NOT NULL,
    CONSTRAINT pk_viv
    PRIMARY KEY (calle, numero),
    CONSTRAINT ck_num
    CHECK (numero > 0),
    CONSTRAINT ck_tv
    CHECK (tipo_vivienda IN ('C', 'B')),
    CONSTRAINT fk_viv_zon
    FOREIGN KEY (nombre_zona)
    REFERENCES ZonaUrbana(nombre_zona));
```

```
CREATE TABLE Persona (
    dni NUMBER(8) NOT NULL,
    nombre_persona VARCHAR2(20) NOT NULL,
    apellidos_persona VARCHAR2(40) NOT NULL,
    od_persona LONG,
    dni_c NUMBER(8) NOT NULL,
    calle VARCHAR2(30) NOT NULL,
    numero NUMBER(3) NOT NULL,
    CONSTRAINT pk_per
    PRIMARY KEY (dni),
    CONSTRAINT fk_per_viv
    FOREIGN KEY (calle, numero)
    REFERENCES Vivienda(calle, numero),
    CONSTRAINT fk_per_per
    FOREIGN KEY (dni_c)
    REFERENCES Persona(dni));
```

```
CREATE TABLE BloqueCasas (
    calle VARCHAR2(30) NOT NULL,
    numero NUMBER(3) NOT NULL,
    metros_b NUMBER(4),
    od bloque LONG,
    CONSTRAINT pk_blo
    PRIMARY KEY (calle, numero),
    CONSTRAINT fk_blo_viv
    FOREIGN KEY (calle, numero)
    REFERENCES Vivienda(calle, numero),
    ON DELETE CASCADE,
    CONSTRAINT ck_mb
    CHECK (metros_b > 0));
```

```
CREATE TABLE CasaParticular (
    calle VARCHAR2(30) NOT NULL,
    numero NUMBER(3) NOT NULL,
    metros_c NUMBER(4),
    od_casa LONG,
    dni_cp NUMBER(8) NOT NULL,
    CONSTRAINT pk_cas
    PRIMARY KEY (calle, numero),
    CONSTRAINT fk_cas_per
    FOREIGN KEY (dni_cp)
    REFERENCES Persona(dni),
    CONSTRAINT fk_cas_viv
    FOREIGN KEY (calle, numero)
    REFERENCES Vivienda(calle, numero)
    ON DELETE CASCADE,
    CONSTRAINT ck_mt
    CHECK (metros_c > 0));
```

```
CREATE TABLE Piso (
    calle VARCHAR2(30) NOT NULL,
    numero NUMBER(3) NOT NULL,
    escalera VARCHAR2(1) NOT NULL,
    planta NUMBER(2) NOT NULL,
    puerta VARCHAR2(2) NOT NULL,
    metros_p NUMBER(3),
    od_piso LONG,
    dni_p NUMBER(8) NOT NULL,
    CONSTRAINT pk_pis
    PRIMARY KEY (calle, numero, escalera, planta, puerta),
    CONSTRAINT ck_mep
    CHECK (metros_p > 0),
    CONSTRAINT fk_pis_blo
    FOREIGN KEY (calle, numero)
    REFERENCES BloqueCasas(calle, numero)
    ON DELETE CASCADE,
    CONSTRAINT fk_pis_per
    FOREIGN KEY (dni_p)
    REFERENCES Persona(dni));
```

```
CREATE TABLE HabitaPiso (
    dni NUMBER(8) NOT NULL,
    calle VARCHAR2(30) NOT NULL,
    numero NUMBER(3) NOT NULL,
    escalera VARCHAR2(1) NOT NULL,
    planta NUMBER(2) NOT NULL,
    puerta VARCHAR2(2) NOT NULL,
    CONSTRAINT pk_hap
    PRIMARY KEY (dni),
    CONSTRAINT fk_hap_cas
    FOREIGN KEY (calle, numero, escalera, planta, puerta)
    REFERENCES Piso(calle, numero, escalera, planta, puerta)
    ON DELETE CASCADE,
    CONSTRAINT fk_hbp_per
    FOREIGN KEY (dni)
    REFERENCES Persona(dni));
```

Pregunta 4 (1.5 puntos)

Explique todas las variantes posibles de transformación del modelo conceptual al modelo relacional para una relación que tiene una cardinalidad máxima 1:N

Pregunta 5 (1.5 puntos)

Defina los conceptos de granularidad y ligadura, explique la relación que existe entre ellos.