Uso de bases de datos

Práctica 1: El lenguaje SQL I

Queremos disponer de una base de datos para registrar información sobre una clínica veterinaria especializada en perros, que describiremos en términos de un conjunto de relaciones.

Las relaciones con las que trabajamos son las siguientes (claves primarias subrayadas, claves foráneas en *cursiva*) y los atributos no pueden tener valor *NULL* a menos que se diga lo contrario.

Éste es el conjunto de relaciones que describen la base de datos:

DOG (id_dog, name_dog, breed, birth, death, sex, color, fur, *id_owner*)

La relación *DOG* contiene los datos generales sobre los perros que aparecen en la BD. En concreto, para cada perro se guarda el identificador (*id_dog*) que es clave primaria, el nombre (*name*), la raza (*breed*), la fecha de nacimiento (*birth*), la fecha de defunción (*death*) que puede ser *NULL* en caso de que todavía esté vivo y en ningún caso puede ser anterior a *birth*, el sexo (*sex*), el color del pelo (*color*), el tipo de pelo (*fur*) y el identificador de su propietario (*id_owner*), que es clave foránea a *OWNER* con política de actualización en cascada.

El atributo *sex* puede tomar los valores '*M*' para macho (*male*) y '*F*' para hembra (*female*).

El atributo *fur* puede tomar los valores '*Short*', '*Medium*' y '*Long*' según si se trata de un perro de pelo corto, mediano o largo.

Existe la posibilidad de que pueda haber perros diferentes con el mismo nombre.

Aunque un perro pueda tener más de un propietario, en este caso solo se guarda uno como contacto de referencia.

OWNER (id_owner, name_owner, phone, birth, address)

La relación contiene información sobre los propietarios de los perros. En concreto, para cada propietario, se guarda el identificador (*id_owner*) que es clave primaria, el nombre del propietario (*name_owner*), el teléfono (*phone*), la fecha de nacimiento (*birth*) y la dirección (*address*).

Existe la posibilidad de que pueda haber propietarios diferentes con el mismo nombre.

DRUG (id_drug, name_drug, format, type)

Información sobre los medicamentos que se pueden recetar. Para cada medicamento se almacena su identificador (*id_drug*) que es clave primaria, el nombre (*name_drug*), el formato de la presentación (*format*) y su tipología (*type*).

Los nombres de los medicamentos no pueden repetirse.

VACCINE (id_vaccine, name_vaccine, periodicity)

Guarda información sobre las vacunas que hay que suministrar a los perros. Para cada vacuna se guarda el identificador (*id_vaccine*) que es clave primaria, el nombre (*name_vaccine*) y la periodicidad en meses con la que se tienen que suministrar (*periodicity*). Este atributo tendrá valor *NULL* si se trata de vacunas de una sola dosis (que será su valor por defecto).

Los nombres de las vacunas no pueden repetirse.

TEST (id_test, type_test)

Esta relación contiene las pruebas que se pueden hacer en la clínica canina. Se almacenan los valores (*id_test*) que es la clave primaria y el tipo de la prueba que se puede realizar (*type_test*).

VISIT (id_visit, *id_dog,* date, reason, id_veterinary, comments)

La relación *VISIT* contiene los datos de las visitas de los perros al veterinario. En concreto, para cada visita se guarda el identificador de la visita (*id_visit*) que es clave primaria, el identificador del perro (*id_dog*) que es clave foránea a *DOG* con política de actualización en cascada, la fecha de la visita (*date*), el motivo de la visita (*reason*), el número de colegiado del veterinario que visita el perro (*id_veterinary*) y los comentarios (*comments*), que pueden ser *NULL,* que el veterinario considere necesarios.

El atributo *reason* solo puede tomar los valores {*'vaccination'*, '*follow-up*', '*illness*'}.

PRESCRIPTION (*id_visit, id_drug,* dose*,* duration)

La relación *PRESCRIPTION* contiene la información de los medicamentos que se recetan a los perros en las visitas. En concreto, para cada prescripción, se guarda el identificador de la visita (*id_visit*), el identificador del medicamento recetado (*id_drug*), la dosis que tiene que tomar el perro (*dose*) y durante cuántos días se tiene que tomar el medicamento (*duration*).

Los atributos (*id_visit*, *id_drug*) son clave primaria, *id_visit* es clave foránea a *VISIT* con política de actualización en cascada e *id_drug* es clave foránea a *DRUG* con política de actualización en cascada.

El atributo *duration* debe ser mayor que 0.

VACCINATION (*id_visit, id_vaccine*)

La relación *VACCINATION* contiene la información de las vacunas que se han administrado a los perros en las visitas. A veces, en una misma visita, se puede inyectar más de una vacuna. Se almacenan los valores (*id_visit, id_vaccine*) que son clave primaria, donde *id_visit* es clave foránea de *VISIT* con política de actualización en cascada e *id_vaccine* es clave foránea de *VACCINE* con política de actualización en cascada.

DOG_TEST (*id_test, id_visit*)

La relación *DOG_TEST* contiene la información de las pruebas efectuadas a los perros en las visitas. Se almacenan los valores (*id_test, id_visit*) que son clave primaria, donde *id_test* es clave foránea de

TEST con política de actualización en cascada e *id_visit* es clave foránea de *VISIT* con política de actualización en cascada.

ACLARACIONES

Se proporciona el fichero `create_db.sql` con las sentencias SQL necesarias para crear la base de datos y el fichero `inserts_db.sql` con las sentencias de inserción de datos que hay que ejecutar para responder a las preguntas de la práctica, una vez resuelto el ejercicio 1.

Recordad que para poder trabajar sobre tablas que forman parte de un esquema de base de datos concreto, se debe utilizar el nombre del esquema como prefijo, o hay que tener actualizada la variable `search_path`. Para vuestra comodidad, al principio de cada sesión indicad:

```
SET search_path TO ”nombre_bd_que_utilicéis”;
```

Asímismo, acordaos de eliminar de la solución que entreguéis aquellas sentencias auxiliares que utilicéis para vuestras pruebas, como *DROP*, inserciones de prueba, etc. que puedan alterar las salidas de los resultados esperados.

Nota importante: El SQL implementado en PostgreSQL puede aceptar diferentes variantes de sintaxis, que además pueden variar según la versión que instaléis, y que pueden ser o no SQL estándar. Evitad (a menos que se indique lo contrario) utilizar sentencias de este tipo, y concentraros en las que se explican en los módulos didácticos. Esto es especialmente relevante en el caso del módulo 4 (evaluado en esta primera parte de la práctica), donde se explica SQL estándar. Si utilizáis sentencias SQL estándar, vuestro código funcionará en cualquier SGBD.

Pregunta 1 (30 % puntuación)

Enunciado

En el fichero `create_db.sql` disponéis de las sentencias SQL necesarias para crear las tablas *OWNER, DRUG, VACCINE, TEST, VISIT, PRESCRIPTION, VACCINATION* y *DOG_TEST*. Este fichero está incompleto y no se ajusta necesariamente a la descripción dada en el enunciado.

Después de un último análisis de la base de datos, se ha llegado a la conclusión de que se necesita introducir las siguientes mejoras:

1. ¿Qué sentencias son necesarias para **crear** la tabla *DOG* según la definición dada? (10%)
2. También se pide dar las sentencias SQL de **alteración** que permitan hacer algunos cambios en la base de datos:
    - 2.1. En la tabla *PRESCRIPTION* queremos hacer las siguientes modificaciones (5%):
        a. Añadir la restricción de clave primaria (*id_visit*, *id_drug*).
        b. El atributo *dose* sólo puede tomar los valores {*1*, *2*, *3*} para indicar el número de dosis diarias que ha de tomar el perro.
    - 2.2. En la tabla *OWNER* queremos añadir las siguientes modificaciones (5%):
        a. El atributo *address* puede ser nulo.
        b. La columna *birth* no es necesaria y se ha de suprimir.
    - 2.3. En la tabla *VACCINE* queremos hacer las siguientes modificaciones (5%):
        a. La columna *periodicity* si no es *null*, ha de ser más grande que cero.
        b. Añadir el atributo *pharma* para almacenar el nombre del laboratorio que ha hecho la vacuna. El valor por defecto debe ser nulo.
    - 2.4. En la tabla *VISIT* queremos añadir las siguientes modificaciones (5%):
        a. La columna *DATE* no puede tomar valor nulo.
        b. Añadir la restricción de clave foránea de *id_dog* como clave foránea de *DOG* con política de actualización en cascada en caso de *update*.

En el fichero `inserts_db.sql` encontraréis las sentencias de inserción que debéis realizar sobre las tablas **una vez modificadas**. Tened en cuenta que el número de columnas que deben incluir las tablas es el que viene dado en el fichero `inserts_db.sql`, por lo cual no se aceptarán como soluciones válidas aquellas prácticas que contemplen otro tipo de columna no incluida en este fichero.

Nota importante: Para aplicar los cambios requeridos **NO** debéis utilizar los dominios explicados en el módulo teórico.

Nota: En el siguiente enlace encontraréis información sobre el orden de ejecución de las expresiones.

https://www.postgresql.org/docs/current/static/sql-expressions.html#SYNTAX-EXPRESS-EVAL

Criterios de evaluación

- *Las preguntas tienen un peso que se especifica, según el número de apartados y la dificultad.*
- *Las preguntas no contestadas no penalizan.*
- *Las sentencias SQL que no se puedan ejecutar (den error de sintaxis) no serán evaluadas.*
- *Las sentencias de creación de tablas que propongáis que den error al ejecutar el fichero inserts_db.sql que os proporcionamos no serán evaluadas (en definitiva, el número de columnas que han de incluir las tablas es el que viene dado en el fichero inserts_db.sql).*
- *No se evaluarán propuestas de solución que utilicen dominios (CREATE DOMAIN).*

Solución

1. Creación de la tabla *DOG*:

```
CREATE TABLE DOG (
     id_dog SMALLINT NOT NULL,
     name_dog VARCHAR(255) NOT NULL,
     breed VARCHAR(255) NOT NULL,
     birth DATE NOT NULL,
     death DATE,
     sex VARCHAR(255) NOT NULL,
     color VARCHAR(255) NOT NULL,
     fur VARCHAR (255) NOT NULL,
     id_owner SMALLINT NOT NULL,
     CONSTRAINT PK_DOG PRIMARY KEY(id_dog),
     CONSTRAINT   FK_OWNER_DOG   FOREIGN   KEY   (id_owner)   REFERENCES
OWNER(id_owner) ON UPDATE CASCADE,
     CONSTRAINT sex CHECK (sex IN ('M','F')),
     CONSTRAINT fur CHECK (fur IN ('Short', 'Medium', 'Long')),
     CONSTRAINT CHECK_DEATH CHECK (death IS NULL OR birth < death));
```

2. Alteración de las tablas:

2.1. Tabla *PRESCRIPTION*:

```
ALTER TABLE PRESCRIPTION
ADD CONSTRAINT PK_PRESCRIPTION PRIMARY KEY(id_visit, id_drug),
ADD CONSTRAINT dose CHECK (dose IN (1, 2, 3));
```

2.2. Tabla *OWNER*:

```
ALTER TABLE OWNER
ALTER address DROP NOT NULL,
DROP COLUMN birth;
```

2.3. Tabla *VACCINE*:

```
ALTER TABLE VACCINE
ADD CONSTRAINT PERIODICITY_NO_ZERO CHECK (periodicity IS NULL OR periodicity > 0),
ADD COLUMN pharma VARCHAR(255) DEFAULT NULL;
```

2.4. Tabla *VISIT*:

```
ALTER TABLE VISIT
ALTER date SET NOT NULL,
ADD CONSTRAINT FK_DOG_VISIT FOREIGN KEY(id_dog) REFERENCES DOG(id_dog) ON UPDATE CASCADE;
```

Pregunta 2 (45 % puntuación)

Enunciado

1. Diseñad una consulta que devuelva los perros de raza *Boxer*, que estén vacunados contra la rabia: '*Rabies*' en la base de datos. En concreto, se pide el nombre del perro, el nombre del propietario, el teléfono del propietario y el número de vacunas de la rabia que se le han administrado en total. El resultado de la consulta se quiere ordenado de mayor a menor por el número de vacunas y, en caso de empate, por orden alfabético del nombre del perro.

2. Diseñad una consulta que devuelva los propietarios de perros con al menos un macho y una hembra de la misma raza, que tanto el macho como la hembra estén vivos y que nunca se hayan visitado en el veterinario por enfermedad (*illness*). En concreto, se pide el nombre del propietario, el teléfono y la raza o razas de perros. El resultado de la consulta se quiere ordenado alfabéticamente por raza.

3. Diseñad una vista (*blood_chemistry_panel*) que obtenga los 3 perros vivos con más pruebas de tipo *Blood Chemistry Panel* realizadas, que no tengan visitas al veterinario entre el 1 de enero del año 2020 y el 31 de diciembre del año 2022. En concreto, queremos el nombre del perro, la raza, la fecha de nacimiento, el sexo y el número de pruebas del tipo indicado. El resultado de la consulta se quiere ordenado de mayor a menor por el número de tests, y en caso de empate, alfabéticamente por el nombre del perro.

Nota: En los siguientes enlaces encontraréis información sobre:

- funciones agregadas: https://www.postgresql.org/docs/8.2/functions-aggregate.html
- la cláusula *LIMIT*: https://www.postgresql.org/docs/8.1/queries-limit.html
- funciones de fecha y hora: https://www.postgresql.org/docs/8.1/functions-datetime.html

Criterios de evaluación

- *Todas las preguntas tienen el mismo peso.*
- *Las preguntas no contestadas no penalizan.*
- *Las sentencias SQL que no se puedan ejecutar (den error de sintaxis) no serán evaluadas.*
- *Se valorará positivamente el uso de sentencias SQL estándar (al margen de otros elementos indicados en el enunciado).*
- *Para obtener la máxima nota, la propuesta de solución de cada pregunta debe incluir el resultado (captura de pantalla o similar).*
- *Para obtener la máxima nota en cada pregunta, la propuesta de solución se tiene que ajustar estrictamente a lo que se pide en el enunciado (por ejemplo, tiene que incorporar todas las columnas que se esperan en el resultado, se deben hacer las ordenaciones pedidas, …)*
- *Igualmente, para obtener la máxima nota, la presentación de los resultados debe ser precisa y amigable (nombres de las columnas, redondeo de decimales, …)*
- *Para obtener la máxima nota, la propuesta de solución no debe presentar operaciones o cláusulas innecesarias.*

Solución

1.
```
SELECT d.name_dog, o.name_owner, o.phone, COUNT(d.id_dog) AS
number_of_doses
FROM DOG d JOIN owner o ON (d.id_owner = o.id_owner)
           JOIN visit vi ON (d.id_dog = vi.id_dog)
           JOIN vaccination va ON (vi.id_visit = va.id_visit)
           JOIN vaccine vn ON (va.id_vaccine = vn.id_vaccine)
WHERE d.breed = 'Boxer' AND
      vi.reason = 'vaccination' AND
     vn.name_vaccine = 'Rabies'
GROUP BY d.id_dog, d.name_dog, o.name_owner, o.phone
ORDER BY number_of_doses DESC, d.name_dog;


name_dog |  name_owner  |   phone   | number_of_doses
---------+--------------+-----------+----------------
Chloe    | Sergio Ortiz | 655729481 |        5
(1 row)
```

La condición *vi.reason = 'vaccination'*, podría no ser necesaria ya que, las vacunas se ponen en visitas de vacunación. Por tanto, si se comprueba mediante *triggers* al modificar los datos, la condición no sería necesaria.

2.
```
SELECT o.name_owner, o.phone, breed
FROM DOG d LEFT JOIN visit v ON v.id_dog = d.id_dog
JOIN OWNER o ON d.id_owner = o.id_owner
WHERE (reason IS NULL OR reason <> 'illness') AND d.death IS NULL
GROUP BY o.id_owner, o.name_owner, o.phone, breed
HAVING COUNT(DISTINCT sex) > 1
ORDER BY breed;
```

```
  name_owner   |   phone   |    breed
---------------+-----------+----------------
Javier Diaz    | 634927361 | French Bulldog
Javier Diaz    | 634927361 | Great Dane
Laura Gonzalez | 656734829 | Pekingese
(3 rows)
```

3.
```
CREATE OR REPLACE VIEW blood_chemistry_panel AS (
SELECT d.name_dog, d.breed, d.birth, d.sex, COUNT(dt.id_test) AS
blood_panels
FROM DOG d JOIN VISIT v ON d.id_dog = v.id_dog
          JOIN DOG_TEST dt ON v.id_visit = dt.id_visit
          JOIN TEST t ON dt.id_test = t.id_test
WHERE d.death is NULL AND
       t.type_test = 'Blood Chemistry Panel' AND
       d.id_dog NOT IN
          (SELECT v2.id_dog
           FROM VISIT V2
           WHERE v2.date BETWEEN '2020-01-01' AND '2022-12-31')
GROUP BY d.id_dog, d.name_dog, d.breed, d.birth, d.sex
ORDER BY blood_panels DESC, d.name_dog
LIMIT 3);

SELECT * FROM blood_chemistry_panel;
```

```
name_dog |  breed    |   birth    | sex | blood_panels
---------+-----------+------------+-----+-------------
Daisy    | Chihuahua | 2017-09-28 |  F  |      2
Bear     | Bulldog   | 2021-01-17 |  M  |      1
Duke     | Bulldog   | 2017-06-14 |  M  |      1
(3 rows)
```

Pregunta 3 (25 % puntuación)

Enunciado

Se ha sabido que a causa de un efecto adverso al administrar *Doxycyline* a los perros de raza 'Bulldog', que hayan sido vacunados más de una vez de *Leptospirosis* entre los años 2015 y 2022, se les ha de reducir la dosis de *Doxyciline* a 1 en todas las prescripciones a partir del año 2023. Obviamente, los perros deben estar vivos.

Proponed una **única sentencia SQL** para corregir las filas incorrectas (en el caso de que utilicéis más de una, el ejercicio se considerará incorrecto). Por otro lado, mostrad el conjunto de las filas que se actualizan.

Nota importante: cuando probéis la implementación de esta actualización, tened en cuenta que siempre debéis partir de la misma base de datos, de otro modo podéis encontrar incoherencias en vuestros análisis.

Criterios de evaluación

- *Las sentencias SQL que no se puedan ejecutar (den error de sintaxis) no serán evaluadas.*
- *Para obtener la máxima nota en cada pregunta, la propuesta de solución se tiene que ajustar estrictamente a lo que se pide en el enunciado (hay que actualizar los datos estrictamente necesarios) y con una única sentencia de UPDATE.*
- *Para obtener la máxima nota, la solución debe ser eficiente (por ejemplo, se valorará negativamente hacer más joins de las necesarias).*
- *Se debe argumentar la respuesta proporcionada. Para obtener la máxima puntuación se debe mostrar la relación de filas que se actualizan.*
- *Igualmente, para obtener la máxima nota, la presentación de los resultados debe ser precisa y amigable (nombres de las columnas, redondeo de decimales, …).*
- *Para obtener la máxima nota, la propuesta de solución no debe presentar operaciones o cláusulas innecesarias.*

Solución

Filas afectadas antes de hacer la actualización:

```
SELECT d.id_dog, d.name_dog, v.date, p.dose, dr.name_drug
FROM PRESCRIPTION p NATURAL JOIN VISIT v
                    NATURAL JOIN DOG d
                    NATURAL JOIN DRUG dr
WHERE dr.name_drug = 'Doxycycline' AND
   d.breed = 'Bulldog' AND
   d.death IS NULL AND
   EXTRACT (YEAR FROM v.date) > 2022 AND
   d.id_dog IN (
      SELECT vi.id_dog
      FROM VISIT vi NATURAL JOIN vaccination
                    NATURAL JOIN vaccine vn
      WHERE vn.name_vaccine = 'Leptospirosis' AND
            EXTRACT (YEAR FROM vi.date) BETWEEN 2015 AND 2022
      GROUP BY vi.id_dog
      HAVING COUNT(*) > 1);
```

```
id_dog | name_dog |    date    |dose | name_drug
-------+----------+------------+-----+------------
     2 | Rocky    | 2023-01-22 |   3 | Doxycycline
    23 | Duke     | 2023-01-22 |   2 | Doxycycline
(2 rows)
```

Actualización:

```
-- Option 1
UPDATE PRESCRIPTION
SET dose = 1
WHERE id_drug IN
  (SELECT d.id_drug
   FROM DRUG d
   WHERE d.name_drug = 'Doxycycline') AND
         id_visit IN
           (SELECT v.id_visit
            FROM VISIT v JOIN DOG d ON v.id_dog = d.id_dog
            WHERE d.death is NULL AND
                  EXTRACT (YEAR FROM v.date) > 2022 AND
                  breed = 'Bulldog' AND
                  v.id_dog IN
                     (SELECT id_dog
                      FROM VISIT vi NATURAL JOIN VACCINATION va
                                    NATURAL JOIN VACCINE vn
                      WHERE vn.name_vaccine = 'Leptospirosis' AND
                        EXTRACT (YEAR FROM vi.date) BETWEEN 2015 AND 2022
                      GROUP BY id_dog
                      HAVING COUNT(*) > 1));

-- Option 2
UPDATE prescription
SET dose = 1
WHERE (id_visit, id_drug) IN
   (SELECT p.id_visit, p.id_drug
    FROM dog AS d NATURAL JOIN visit AS v
                  NATURAL JOIN prescription AS p
                  NATURAL JOIN drug AS dr
    WHERE dr.name_drug = 'Doxycycline' AND
          EXTRACT (YEAR FROM v.date) >= 2023 AND
          d.breed = 'Bulldog' AND
          d.death IS NULL AND
          d.id_dog IN
            (SELECT v.id_dog
             FROM visit AS v NATURAL JOIN vaccination AS vacn
                             NATURAL JOIN vaccine AS vac
             WHERE v.reason = 'vaccination' AND
                   EXTRACT (YEAR FROM v.date) BETWEEN 2015 AND 2022 AND
                   vac.name_vaccine = 'Leptospirosis'
             GROUP BY v.id_dog
             HAVING COUNT (*) > 1));
```

Filas afectadas después de hacer la actualización:

```
id_dog | name_dog |    date    | dose |  name_drug
-------+----------+------------+------+------------
     2 | Rocky    | 2023-01-22 |         1| Doxycycline
    23 | Duke     | 2023-01-22 |         1| Doxycycline
(2 rows)
```