

Sea el siguiente esquema relacional que almacena información relativa a un concurso realizado entre alumnos de la asignatura de Bases de datos para medir su pericia en la resolución de consultas SQL. El esquema relacional es el siguiente:

CONSULTA(cod_con: entero, enunciado: char(60), dificultad: char(5))

CP: {cod_con}

VNN: {enunciado, dificultad}

TABLA(nom_tab: char(15), grado: entero, cardinalidad: entero)

CP: {nom_tab}

VNN: {grado, cardinalidad}

ALUMNO(dni: entero, nombre: char(40), edad: entero, nota_bda: real)

CP: {dni}

VNN: {nombre, edad}

NECESITA(cod_con: entero, nom_tab: char(15))

CP: {cod_con, nom_tab}

CAj: {cod_con} → CONSULTA

CAj: {nom_tab} → TABLA

RESUELVE(cod_con: entero, dni: entero, tiempo: real, correcta: char(2))

CP: {cod_con, dni}

CAj: {cod_con} → CONSULTA

CAj: {dni} → ALUMNO

VNN: {tiempo, correcta}

Donde las relaciones tienen el siguiente significado:

- **Consulta:**
 - *cod_con*: código de la consulta
 - *enunciado*: texto de la consulta
 - *dificultad*: ['Baja', 'Media', 'Alta']
- **Tabla:**
 - *nom_tab*: nombre de la tabla
 - *grado*: cuántos atributos tiene
 - *cardinalidad*: cuántas tuplas tiene
- **Alumno:**
 - *dni*: D.N.I. del alumno
 - *nombre*: cómo se llama
 - *edad*: cuántos años tiene
 - *nota_bda*: obtenida en el último examen de BDASI
- **Necesita**: la consulta de código *cod_con* necesita para su resolución la tabla de nombre *nom_tab*.
- **Resuelve**: el alumno de D.N.I. *dni* ha resuelto la consulta de código *cod_con* en *tiempo* segundos. La solución dada es buena si *correcta* toma el valor 'Sí', en caso contrario tomará el valor 'No'.

Resolver las siguientes cuestiones:

- 1) Mediante el DDL de SQL podríamos definir el esquema lógico de la Base de datos anterior. Para cada una de las restricciones de integridad siguientes indica en qué casos no se podrían expresar dentro de la instrucción "Create table" y en cuales sí que sería posible. En este último caso indica cómo se expresaría la restricción en SQL. (0'4 puntos)

Responde caso por caso.

- a) Toda consulta necesita al menos una tabla.

No se puede representar en las instrucciones del DDL en ORACLE. Habría que definirla en lenguaje natural, en SQL estándar se podría incluir un aserto.

- b) El valor del atributo *correcta* en *Resuelve* sólo puede ser "Sí" o "No".

Añadiendo en la instrucción de creación de la tabla RESUELVE la cláusula: CHECK (VALUE IN ('Sí', 'No'))

- c) El valor del atributo *cardinalidad* en la relación *Tabla* no puede ser negativa.

Añadiendo en la instrucción de creación de la tabla TABLA la cláusula: CHECK (VALUE >=0)

- d) En la relación *Necesita* la restricción: CAj: {cod_con}→CONSULTA.

REFERENCES Consulta (cod_con).

- 2) Escriba en SQL las instrucciones que permitirían resolver las siguientes consultas:

- a) Obtener el código y el enunciado de cada consulta resuelta por algún alumno y que necesite más de dos tablas. (0'6 puntos)

```
SELECT C.cod_con,C.enunciado
FROM Consulta C
WHERE C.cod_con IN (SELECT cod_con FROM Resuelve) AND
      2<(SELECT COUNT(*)
        FROM Necesita N WHERE C.cod_con=N.cod_con);
```

- b) Obtener el nombre de la tabla que es necesitada en un mayor número de consultas (puede haber más de una). (0'6 puntos)

```
SELECT N.nomtabla
FROM Necesita N
GROUP BY N.nom_tab
HAVING COUNT(*) = (SELECT MAX(COUNT(*))
                  FROM Necesita N1
                  GROUP BY N1.nom_tab)
```

- c) Obtener de todos los alumnos que hay en la base de datos de los que se desconozca la nota en BDASI, el dni, el nombre indicando el número total de consultas que han resuelto y la cantidad total de segundos que han utilizado en su resolución. (0'8 puntos)

```
SELECT A.dni, A.nombre, COUNT(R.cod_con), SUM(R.tiempo)
FROM Alumno A LEFT JOIN Resuelve R ON R.dni=A.dni
WHERE A.nota IS NULL
GROUP BY A.dni, A.nombre;
```

- d) Obtener el dni y el nombre de los alumnos que han resuelto correctamente todas las consultas de dificultad 'Alta' si es que hay alguna. (0'8 puntos)

Solución 1

```
SELECT A.dni, A.nombre
FROM Alumno A
WHERE NOT EXISTS
    (SELECT *
     FROM Consulta C
     WHERE C.dificultad ='Alta' AND
           NOT EXISTS (SELECT *
                      FROM Resuelve R
                      WHERE R.dni=A.dni AND
                            R.cod_con=C.cod_con AND
                            R.correcta ='Sí')) AND
    EXISTS (SELECT *
            FROM Consulta C1
            WHERE C1.dificultad ='Alta');
```

Solución 2

```
SELECT A.dni, A.nombre
FROM Alumno A
WHERE (SELECT COUNT(*)
FROM Consulta C WHERE C.dificultad ='Alta')
=
(SELECT COUNT(*)
FROM Resuelve R, Consulta C2
WHERE R.dni=A.dni AND R.cod_con=C2.cod_con AND
C2.dificultad='Alta' AND R.correcta ='Sí')) AND
EXISTS (SELECT *
FROM Consulta C1
WHERE C1.dificultad ='Alta');
```

- e) Obtener el código y el enunciado de las consultas que han sido resueltas (correcta o incorrectamente) por algún alumno menor de 20 años, indicando para cada consulta cuántas tablas necesita. Sólo interesan aquellas consultas que necesiten más de 2 tablas. (0'8 puntos)

```
SELECT C.cod_con, C.enunciado, COUNT(N.nom_tab)
```

```
FROM Consulta C, Necesita N
WHERE C.cod_con=N.cod_con
GROUP BY C.cod_con, C.enunciado
HAVING C.cod_con IN (SELECT R.cod_con
                     FROM Resuelve R
                     WHERE R.dni IN (SELECT dni
                                     FROM Alumno A
                                     WHERE A.edad<20))
AND COUNT(N.nom_tab)>2;
```