
25GIIN – Diseño y Desarrollo de Sistemas de Información

Actividad 3 - Portafolio

Gagliardo Miguel Angel – **Grupo 6**

31 de Noviembre de 2024

Analisis Normalizacion del modelo en Tercera Forma Normal (3FN)

Primero, y en base al trabajo entregado en la UC2, he decidido si el modelo de datos proporcionado cumplía con los requisitos de la tercera forma normal (3FN). Para esto, he revisado cada forma normal (1FN, 2FN y 3FN) de manera general, aplicándola al diagrama antes mencionado en la UC2.

1. Primera Forma Normal (1FN)

Una tabla está en **1FN** si:

- Cada columna contiene valores atómicos (sin listas o conjuntos).
- No hay columnas repetidas o grupos de datos multivaluados.

Revisión:

En general, todas las tablas del modelo parecen tener atributos atómicos. Por ejemplo:

- Las tablas como **especie_animal**, **especie_vegetal**, **especie_mineral** tienen columnas con valores atómicos (sin listas o estructuras multivaluadas).
- Las columnas de tipo **enum** son atómicas, ya que los valores de los **enum** se almacenan internamente como números enteros, lo que garantiza la **atomicidad**.

Conclusión: El modelo está en **1FN**.

2. Segunda Forma Normal (2FN)

Una tabla está en **2FN** si:

- Está en **1FN**.
- Todos los atributos **no clave** dependen completamente de la PK (es decir, no hay dependencias parciales, donde un atributo depende solo de una parte de una clave compuesta).

Revisión:

Analizando cada tabla clave y sus dependencias:

- **personal**: La clave primaria es **dni**. Todos los atributos dependen completamente de **dni**, no hay dependencia parcial.
- **proyecto**: La clave primaria es **id_proyecto**, y las relaciones con **id_investigador** e **id_especie** (FKs) son correctas dado que no hay dependencia parcial.
- **alimentacion**: Las FKs **depredador** y **presa** son primarias.
- **especie**: Todos los atributos de la entidad **especie** (como **nombre_cientifico**, **nombre_vulgar**, y **numero_individuos**) dependen completamente de **id_especie** (PK), sin dependencias parciales.
- **encuesta**: La tabla ha sido modificada a partir de la UC2 para tener una PK **id_encuesta** y ahora tiene **tres** claves foráneas: **id_excursion (nullable)**, **id_albergue** y **dni_vis**. Todas independientes de la clave primaria, no hay dependencias parciales.

Conclusión: El modelo está en **2FN**, ya que todas las tablas cumplen con la condición de no tener dependencias parciales.

3. Tercera Forma Normal (3FN)

Una tabla está en **3FN** si:

- Está en **2FN**
- No hay dependencias transitivas, es decir, un atributo no clave no depende de otro atributo no clave.

Revisión:

Analizando las tablas:

- **personal**: Todos los atributos dependen de la PK **dni**. No hay atributos que dependan transitivamente de otros atributos no clave.
- **especie**: No hay atributos en **especie** que dependan de otros atributos que no sean la clave primaria **id_especie**.
- **proyecto**: No hay dependencias transitivas en este caso.
- **alimentacion**: No hay dependencias transitivas. La relación entre **depredador** y **presa** ya está modelada mediante FKs.
- **Encuesta**: No hay atributos que dependan de otros atributos que no sean PKs/FKs.

Conclusión: El modelo está en **3FN**. No hay dependencias transitivas en ninguna de las tablas.

Implementacion en SQL

Nota: Para este punto se ha incluido 1 archivo:

data.sql: Incluye la creacion de la base y las respectivas tablas, junto con la carga de datos correspondiente

A continuacion los modelos de consultas con comentarios sobre las mismas, que de igual manera son explicadas en el [siguiente gist de github](#). El formato se verá “extraño” dado que es extraido de un archivo en markdown, pero es bastante explicito.

25GIIN - UC3 - Consultas SQL

Dada un area del parque:

Areas del parque:

```
mysql> SELECT * FROM area;
+-----+-----+-----+-----+
| id_area | nombre                | extension | id_parque |
+-----+-----+-----+-----+
| 1       | Bosque de los alces   | 5000.00   | 1         |
| 2       | Lago Encantado        | 3000.50   | 1         |
| 3       | Montaa del guila      | 4000.75   | 1         |
| 4       | Selva Tropical        | 7000.10   | 1         |
| 5       | Pradera de los ciervos | 6000.30   | 1         |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

1. Quienes la vigilan?

```
mysql> SELECT
->     a.nombre AS area,
->     p.dni,
->     p.nombre,
->     p.direccion,
->     p.telefono_dom,
->     p.telefono_movil
-> FROM
->     personal_vigilancia pv
-> JOIN
->     personal p ON pv.dni = p.dni
-> JOIN
->     area a ON pv.id_area = a.id_area
-> WHERE
->     a.nombre = "Bosque de los alces";
+-----+-----+-----+-----+
-----+-----+
| area                | dni       | nombre    | direccion                |
| telefono_dom | telefono_movil |
+-----+-----+-----+-----+
-----+-----+
| Bosque de los alces | 12345678 | Juan Perez | Calle Ficticia 123 |
555123456 | 555987654 |
+-----+-----+-----+-----+
-----+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT
->     p.dni,
->     p.nombre,
->     p.direccion,
->     p.telefono_dom,
->     p.telefono_movil
-> FROM
->     personal_vigilancia pv
-> JOIN
->     personal p ON pv.dni = p.dni
-> WHERE
->     pv.id_area = 2;
```

```
+-----+-----+-----+-----+-----+
-----+
| dni      | nombre      | direccion              | telefono_dom |
telefono_movil |
+-----+-----+-----+-----+-----+
-----+
| 23456789 | Ana Lopez   | Avenida Siempre Viva 456 | 555234567   |
555876543 |
+-----+-----+-----+-----+-----+
-----+
1 row in set (0.00 sec)
```

2. Quienes la conservan?

```
mysql> SELECT
->     p.dni,
->     p.nombre,
->     p.direccion,
->     p.telefono_dom,
->     p.telefono_movil
-> FROM
->     personal_conservacion pv
-> JOIN
->     personal p ON pv.dni = p.dni
-> WHERE
->     pv.id_area = 3;
```

```
+-----+-----+-----+-----+-----+
-----+
| dni      | nombre      | direccion              | telefono_dom |
telefono_movil |
+-----+-----+-----+-----+-----+
-----+
| 34567890 | Carlos Garca | Calle Los lamos 789 | 555345678   |
555765432 |
+-----+-----+-----+-----+-----+
-----+
1 row in set (0.00 sec)
```

3. Cuantas especies distintas residen en ella?

```
mysql> SELECT
->     a.nombre AS area_nombre,
->     COUNT(DISTINCT e.id_especie) AS num_especies_distintas
-> FROM
->     area a
-> JOIN
->     especie e ON e.id_area = a.id_area
-> WHERE
->     a.nombre = "Bosque de los alces"
-> GROUP BY
->     a.nombre;
+-----+-----+
| area_nombre          | num_especies_distintas |
+-----+-----+
| Bosque de los alces |                17     |
+-----+-----+
1 row in set (0.00 sec)
```

Dado un investigador:

1. ¿Tienes proyectos relacionados con especies minerales?

Ejemplo para el investigador Juan Perez con id 12345678

```
mysql> SELECT
->     p.nombre AS investigador_nombre,
->     pr.id_proyecto,
->     pr.presupuesto,
->     em.tipo AS especie_mineral
-> FROM
->     personal p
-> JOIN
->     proyecto pr ON p.dni = pr.id_investigador
-> JOIN
->     especie_mineral em ON pr.id_especie = em.id_especie
-> JOIN
->     especie e ON em.id_especie = e.id_especie
-> WHERE
->     p.dni = 12345678;
+-----+-----+-----+-----+
| investigador_nombre | id_proyecto | presupuesto | especie_mineral |
+-----+-----+-----+-----+
| Juan Perez         | 1           | 50000       | roca             |
| Juan Perez         | 6           | 35000       | roca             |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```


Para obtener todos los investigadores envueltos en proyectos que incluyen especies minerales:

```
mysql> SELECT
->     p.nombre AS investigador_nombre,
->     pr.id_proyecto,
->     pr.presupuesto,
->     em.tipo AS especie_mineral
-> FROM
->     personal p
-> JOIN
->     proyecto pr ON p.dni = pr.id_investigador
-> JOIN
->     especie_mineral em ON pr.id_especie = em.id_especie
-> JOIN
->     especie e ON em.id_especie = e.id_especie;
+-----+-----+-----+-----+
| investigador_nombre | id_proyecto | presupuesto | especie_mineral |
+-----+-----+-----+-----+
| Juan Perez         | 1           | 50000       | roca             |
| Juan Perez         | 6           | 35000       | roca             |
| Ana Lopez          | 7           | 33000       | cristal          |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

2. ¿En cuántos proyectos participa y cuánto dinero presupuestan dichos proyectos?

Consulta para el mismo investigador con DNI 12345678:

```
mysql> SELECT
->     p.nombre AS investigador_nombre,
->     COUNT(pr.id_proyecto) AS numero_de_proyectos,
->     SUM(pr.presupuesto) AS presupuesto_total
-> FROM
->     personal p
-> JOIN
->     proyecto pr ON p.dni = pr.id_investigador
-> WHERE
->     p.dni = 12345678
-> GROUP BY
->     p.dni;
+-----+-----+-----+
| investigador_nombre | numero_de_proyectos | presupuesto_total |
+-----+-----+-----+
| Juan Perez         | 3                   | 100000           |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Dado un albergue

1. Número de encuestas con calificación inferior a 3.

```
mysql> SELECT
->     COUNT(*) AS encuestas_bajas_calificacion
-> FROM
->     encuesta
-> WHERE
->     calificacion < 3;
+-----+
| encuestas_bajas_calificacion |
+-----+
|                               4 |
+-----+
1 row in set (0.00 sec)
```

1. Nombre de los albergues que han obtenido, por lo menos una vez, calificación de 1. (Recuerda que las encuestas van de 1-5)

```
mysql> SELECT DISTINCT a.nombre
-> FROM albergue a
-> JOIN encuesta e ON a.id_albergue = e.id_albergue
-> WHERE e.calificacion = '1';
+-----+
| nombre |
+-----+
| Abadia |
+-----+
1 row in set (0.00 sec)
```

Dada una especie:

1. ¿Quiénes son sus depredadores?

Sabemos que el salmon (ID 19) es depredado por varias especies:

```
mysql> SELECT ea.id_especie, e.nombre_vulgar FROM especie e JOIN
especie_animal ea ON e.id_especie = ea.id_especie;
+-----+-----+
| id_especie | nombre_vulgar |
+-----+-----+
|          11 | Len           |
|          12 | Lobo          |
|          13 | Oso pardo     |
|          14 | Gorila        |
|          15 | Elefante asitico |
|          16 | Ciervo rojo   |
```

```
|          17 | Cocodrilo de agua salada |
|          18 | Hipoptamo                |
|          19 | Guepardo                 |
|          20 | Zorro rojo               |
|          21 | Salmon Rojo              |
+-----+
11 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM alimentacion;
+-----+
| depredador | presa |
+-----+
|          11 |     16 |
|          11 |     19 |
|          13 |     21 |
|          19 |     21 |
+-----+
4 rows in set (0.00 sec)
```

Por tanto:

```
mysql> SELECT e.nombre_vulgar AS depredador FROM especie_animal ea JOIN
alimentacion a ON ea.id_especie = a.depredador
JOIN especie e ON e.id_especie = ea.id_especie WHERE a.presa = 21;
+-----+
| depredador |
+-----+
| Oso pardo  |
| Guepardo   |
+-----+
2 rows in set (0.00 sec)
```

1. ¿Cuántos hay en una determinada área?

Areas:

```
mysql> SELECT * FROM area;
+-----+-----+-----+-----+
| id_area | nombre                | extension | id_parque |
+-----+-----+-----+-----+
|        1 | Bosque de los alces   | 5000.00  |          1 |
|        2 | Lago Encantado        | 3000.50  |          1 |
|        3 | Piedra del guila      | 4000.75  |          1 |
|        4 | Selva Tropical        | 7000.10  |          1 |
|        5 | Pradera de los ciervos | 6000.30  |          1 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Ejemplo para ID 3, Piedra del aguila:

```
mysql> SELECT SUM(e.numero_individuos) AS total_individuos FROM especie e
JOIN area a ON e.id_area = a.id_area WHERE e.id_especie = 3 AND a.id_area =
3;
+-----+
| total_individuos |
+-----+
|                450 |
+-----+
1 row in set (0.00 sec)
```

Si tu modelo no es capaz de responder algunas de estas preguntas, explica por qué y qué cambios harías para que sí pudiera satisfacerlas.

El unico cambio que he tenido que hacer, es que la tabla encuestas dependia segun la definicion de la UC2 solo de excursion, asi que tuve que adaptarla para que se puedan insertar FKs (NULLABLEs) de `id_excursion` e `id_albergue`