

EJERCICIOS NORMALIZACIÓN

1. Define Base de Datos Relacional.

Una base de datos relacional es en la cual todas las tablas/relaciones están normalizadas.

2. ¿Qué es la Normalización?

La Normalización son una serie de reglas que, de cumplirse aseguran que el esquema diseñado tendrá un buen comportamiento en cuanto a redundancia, pérdida de información y representación de la misma.

3. Define la Primera Forma Normal.

Una tabla está en 1FN cuando todos sus datos son atómicos, es decir, cada celda contiene un único valor.

4. Define Segunda Forma Normal.

Una tabla está en 2FN si está en 1FN y cada atributo no clave tiene una dependencia funcional plena de la clave primaria, lo que significa que no depende de una parte de la clave primaria si esta es compuesta.

5. Define Tercera Forma Normal.

Una tabla está en 3FN si está en 2FN y además ningún atributo no clave dependen transitivamente de la clave primaria, es decir, no hay dependencias funcionales indirectas entre atributos no clave, es decir, que ningún atributo que no sea clave depende de otro que tampoco es clave.

6. Explica cuando una tabla NO está en Primera Forma Normal, los inconvenientes que puede tener y como arreglarías el modelado para que estuviera en Primera Forma Normal. Pon un ejemplo.

Cuando una tabla NO está en 1FN:

Inconvenientes: Puede tener listas o conjuntos en una celda, dificultando consultas y manipulación de datos.

Solución: Dividir esos valores en filas separadas.

Ejemplo:

Antes:

Cliente	Teléfonos
C1	123, 456

Después:

Cliente	Teléfonos
C1	123
C1	456

7. Explica cuando una tabla NO está en Segunda Forma Normal, los inconvenientes que puede tener y como arreglarías el modelado para que estuviera en Segunda Forma Normal. Pon un ejemplo.

Cuando una tabla NO está en 2FN:

Inconvenientes: Algunos datos dependen de solo una parte de la clave primaria compuesta, causando redundancia.

Solución: Dividir en varias tablas. Se crearía una nueva tabla con los atributos no primarios que dependen de una parte de la clave junto con esa parte de la clave.

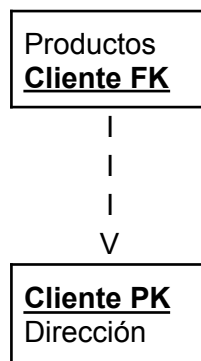
Ejemplo:

Antes:

<u>Pedido PK</u>
Productos
<u>Cliente PK</u>
Dirección

Después:

<u>Pedido PK</u>



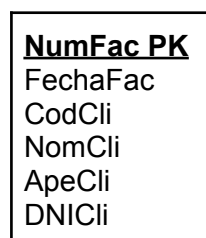
8. Explica cuando una tabla NO está en Tercera Forma Normal, los inconvenientes que puede tener y como arreglarías ese modelado para que estuviera en Tercera Forma Normal. Pon un ejemplo.

Cuando una tabla NO está en 3FN:

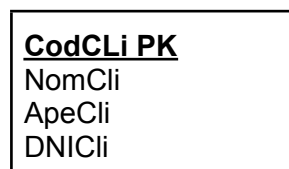
Inconvenientes: Hay dependencias transitivas, es decir, un atributo no clave depende de otro atributo no clave.

Solución: Separar los atributos relacionados en tablas adicionales para eliminar la dependencia transitiva. Se crea otra tabla donde están los atributos no clave que dependen de otro no clave, junto con el otro no clave del que dependen tmb.

Antes:



Después:



=

|
|
|
|
|
o

^

NumFac PK FechaFac CodCli FK
--

9. ¿En qué forma normal no se encuentra la siguiente relación? Aplica la regla correspondiente y arregla la relación dividién

dola en las relaciones necesarias. R: (NumMat, NomAlu, ApeAlu, DirAlu, CodCiu, NomCiu)

Problema: Existe una dependencia transitiva entre CodCiu y NomCiu que viola la 3FN.

Solución: Dividir la relación en:

R1: (NumMat, NomAlu, ApeAlu, DirAlu, CodCiu)

R2: (CodCiu, NomCiu)

10. ¿En qué forma normal no se encuentra la siguiente relación? Explica que debe cumplir una relación para estar en dicha forma normal. Aplica la regla correspondiente y arregla la relación dividiéndola en las relaciones necesarias. R: (NumMat, CodAsignatura, NomAsignatura, NomAlu, ApeAlu, DNIALu, NotaAluAsig)

Problema: Hay dependencias parciales y transitivas.

Solución: Dividir en:

R1: (NumMat, CodAsignatura, NotaAluAsig)

R2: (CodAsignatura, NomAsignatura)

R3: (DNIALu, NomAlu, ApeAlu)

Que entra en el examen: punto 1, terminología,

EJERCICIOS INTEGRIDAD

1. Define qué significa que una Base de Datos Relacional sea Integra ¿Cómo se puede conseguir?

Una base de datos relacional es íntegra cuando garantiza que los datos reflejan correctamente la realidad, evitando inconsistencias. Se consigue aplicando reglas de integridad que controlan la validez de los datos, como integridad de entidad, referencial y semántica, durante la creación y manipulación de las tablas. **(No tiene información errónea almacenada.) y que después de realizar cualquier operación siga siendo íntegra.**

2. Las Reglas de Integridad (RI) se pueden clasificar en Específicas y Generales. Defínelas.

Específicas: Aplican a una base de datos concreta y dependen del modelo de datos. Por ejemplo, "la cantidad no puede ser negativa". **Se llaman reglas de integridad semánticas.**

Generales: Son comunes a todas las bases de datos y controlan aspectos como claves primarias y claves ajenas (integridad referencial). **Entidad, clave y referencial.**

3. Define RI de la Entidad y de la Clave ¿Cómo se controla al crear la tabla?

RI de la Entidad: Garantiza que la clave primaria no puede tener valores nulos, asegurando que cada registro sea único y identificable.

RI de la Clave: Garantiza que los valores de la clave primaria sean únicos en la tabla.

¿Cómo se controla?: Al crear la tabla, se define una clave primaria, lo que asegura unicidad y no permite valores nulos.

4. Define RI Referencial ¿Cómo se controla al crear la tabla?

Definición: Garantiza que los valores de una clave ajena coincidan con los valores de la clave primaria en la tabla referida, o que sean nulos. Que se correspondan con la clave primaria a la que hacen referencia.

¿Cómo se controla? Al definir una clave foránea en la tabla, indicando cómo manejar cambios en la tabla referida.

5. ¿Qué son las RI Semánticas? ¿Cómo se controlan al crear la tabla?

Definición: Son restricciones que reflejan reglas del mundo real y la lógica del negocio, como "los precios deben ser mayores a cero".

Se puede controlar con los máximos y mínimos. Se controla añadiendo restricciones UNIQUE, CHECK, DEFAULT y/o con Triggers.