

PREGUNTAS TEÓRICAS

BASES DE DATOS

Ismael Abu-Jadur García

1. ¿Qué es una base de datos?

Es un **conjunto de archivos relacionados** entre sí que representan el dominio de un problema del mundo real.

Estas **almacenar tanto los datos/hechos como la representación lógica** del dominio del problema, y que están sujetos a **unas restricciones existentes** en el dominio del problema, las cuales también son almacenadas en esos archivos

2. ¿Qué problemas evita normalizar en FN3 en las operaciones de Borrado?

Eliminación de información, ya que se **elimina la información de la dependencia entre dos atributos no primos** cuando se elimina la última tupla en la que está presente

3. ¿Qué problemas evita normalizar en FN2 en las operaciones de Inserción?

Redundancia de valores de **atributos dependientes**, ya que se repite su valor para un mismo valor del atributo clave. **Inconsistencia** de valores de **atributos dependientes**, ya que pueden tener distinto valor para un mismo valor del atributo clave.

4. ¿Por qué el modelo relacional considera la existencia de claves foráneas?

Para **representar las relaciones** que existen entre las **entidades del mundo real** manteniendo las características o propiedades de estas relaciones.

5. Un determinante funcional es:

Uno o un conjunto de atributos de una relación del cual depende funcionalmente de forma completa otro atributo de la misma relación

6. Definición de la Forma Normal de Boyce-Codd

Una relación R se encuentra en forma normal FNBC, si y sólo si, se **encuentra en FN1** y toda **clave candidata** es un **determinante funcional** de la relación R.

7.

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, e4),
(a2, b4, c5, d1, e5)

Indique las dependencias funcionales que no se aplican a R

1	A → C	5	D → E
2	C → A	6	E → B
3	E → A	7	C → B
4	B → D	8	B → A

2-7 → Son las que pueden ser aplicadas.

1-3-4-5-6-8 → No se aplican.

Para saber si se cumple o no, indicar la siguiente frase:

Para cada valor de A hay un único valor de C → APARTADO 1 //No se cumple//

Para cada valor de C hay un único valor de A → APARTADO 2 (No puede asegurarse, ya que no tenemos más valores de c1,c2,c3,c5 para saber si siempre le corresponde la misma a).

8. Características de una base de datos

Versatilidad: Pueden existir varias visiones de una misma información.

Desempeño: Las bases de datos deben asegurar un tiempo de respuesta adecuado.

Capacidad de acceso: Los usuarios de la base de datos reclaman a esta continuamente información sobre los datos.

Mínima redundancia: Se debe evitar la redundancia de datos.

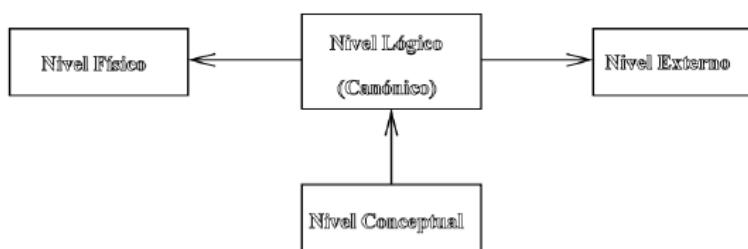
Simplicidad: Deben permitir la inclusión y/o modificación de nuevos ítems de datos sin complejidad.

Integridad: Los datos almacenados tienen que ser verídicos.

Seguridad: Protección de los datos contra su pérdida total o parcial.

Organización: Cuanto mejor sea la organización física, más rapidez de acceso.

9. Visión de los datos en una BD.



El conjunto de niveles garantiza la independencia de los datos.

Nivel Físico: Representa como se almacena la información en los dispositivos.

Nivel Conceptual: Representación del problema del mundo real.

Nivel Externo: Visión que tienen los usuarios finales de la BD.

Nivel Lógico: Es el nexo de unión entre niveles, independiente de las visiones externas e internas.

10. Definición de Granularidad y ligadura, que tienen en común.

Granularidad: Es el nivel de detalle con el que se define la representación externa de un sistema derivada de la representación lógica.

A mayor granularidad, mayor independencia de los datos.

A mayor granularidad, mayor complejidad del software.

Ligadura: Es el momento en que se transforman los esquemas externos usados en una aplicación en términos de esquema interno. Tenemos dos tipos:

Ligadura Lógica: Vinculación entre el nivel externo y lógico.

Ligadura Física: Vinculación entre el nivel lógico y físico.

El proceso de vinculación es recomendable realizarlo en la fase de ejecución:

-Compilación > Enganche > ejecución > acceso a BD.



11. Define que es un SGBD y sus componentes.

Un SGBD es un conjunto de programas que permiten al usuario de una base de datos realizar un conjunto de tareas, entre las cuales tenemos:

- Definición de datos -Manipulación de datos.
- Mantener privacidad y seguridad de los datos. -Mantener integridad de la BD.

Los componentes son 7:

- 1) **DDL:** Lenguaje de definición de datos, permite la representación lógica de los datos, estos compilados son almacenados en el diccionario de datos.
- 2) **DSDL:** Lenguaje de manipulación de almacenamiento de datos: Define los datos correspondientes al dominio del problema a los dos niveles de abstracción, denominado Esquema de la BD.
- 3) **DML:** Lenguaje de manipulación de datos, permite realizar las funciones de gestión de datos: Manipular los datos, definir el nivel externo y definir las vistas que tendrán los usuarios del esquema de una BD.
- 4) **Diccionario de datos:** Es una metabase de datos que contiene información acerca de los datos que pueden ser almacenados en la BD como son:
 - a. Esquema Lógico y Esquema Físico.
 - b. Subesquemas y restricciones
 - c. Información de integridad.
- 5) **Monitor de BD:** Es un componente software que garantiza la correcta, segura, integra y eficiente acceso y almacenaje de los datos. Se encarga de ser la interfaz entre el programa de aplicación y los datos.
- 6) **Administrador de Base de datos:** Componente humano que tendrá asignadas las tareas de:
 - a. Definir el esquema lógico de la BD.
 - b. Definir el esquema físico de la BD.
 - c. Definir esquemas o visiones externas.
 - d. Control de Privacidad de los datos.
 - e. Especificación de los procedimientos para mantener la seguridad de los datos.
- 7) **Usuarios de la BD:**
 - a. **Usuario Terminal:** Usuarios que a través de programas de aplicación interactúan con la BD.
 - b. **Usuario técnico:** Desarrollan programas que usarán los U. terminales.
 - c. **Usuario Especializado:** usan el SGBD como herramienta de desarrollo.
 - d. **Usuario Crítico:** Usuario del staff de una BD.

12. ¿Qué es la abstracción? Técnicas usadas.

Es una técnica que aísla un elemento de su contexto o del resto de elementos formando una clase u objeto.

Generalización: Un conjunto de objetos puede ser visto como un objeto más general.

Especialización: Proceso inverso a generalización.

Instanciación: Proceso por el cual se crea un objeto.

Clasificación: Proceso inverso a instanciación

Agregación: Considera un objeto según los elementos que lo forman.

Refinamiento: Inverso a la agregación.

13. Diferencias entre los tipos de entidades del modelo E-R.

Fuerte: Su existencia no depende de la existencia de otra entidad.

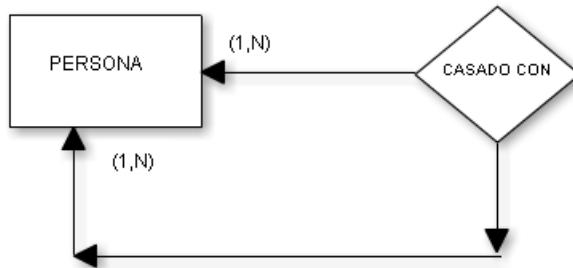
Débil: Su existencia depende de otra entidad fuerte:

Identificación: No puede ser identificada a no ser que se identifique mediante una entidad fuerte.

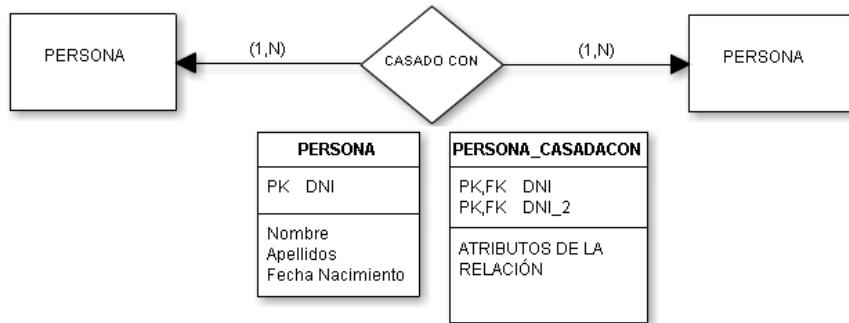
Existencia: Esta entidad puede ser identificada sin necesidad de la fuerte, pero su existencia depende de ella.

Una debilidad de identificación implica una debilidad de existencia, pero no al contrario.

14. Transformación de Conceptual a relacional



Pueden ser tratadas como relaciones unarias y aplicar las RTCAR:



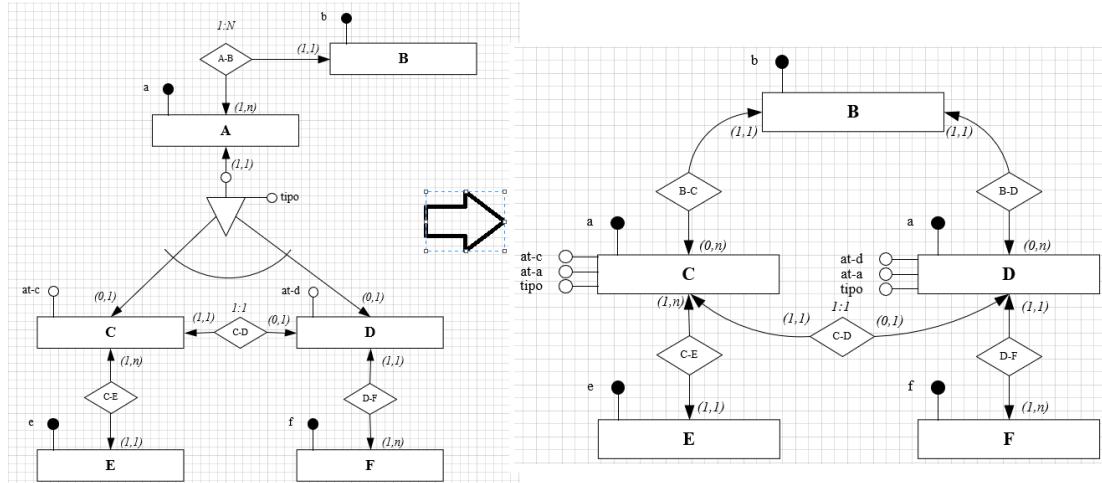
15. Tipos de Interrelaciones en E-R

PRTECAR 1: Eliminar atributos múltiples generando una entidad débil 1:N o N:N.

PRTECAR 2: Eliminar atributos compuestos.

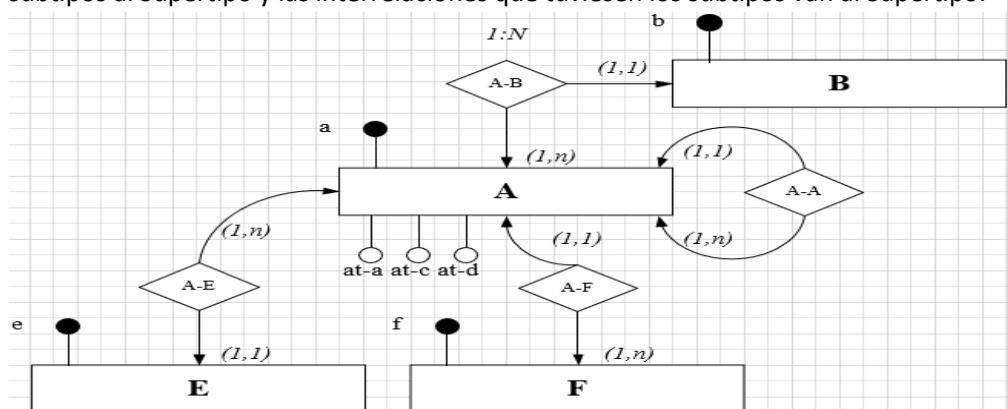
PRTECAR 3: Eliminar el Supertipo de entidad (Jerárquica), se pasarán todos los atributos del Supertipo a cada uno de los subtipos y cada uno de los tipos de interrelación del Supertipo se mantienen para cada subtipo

(NO APTA PARA PARCIALES)



↗ B (b, ...)
 ↗ C (a, at-a, at-c, tipo, ..., e(not null), **b(not null)**)
 ↗ D (a, at-a, at-d, tipo, ..., **b(not null)**, a')
 ↗ E (e, ...)
 ↗ F (f, ..., a(not null))

PRTECAR 4: Eliminar subtipos de entidad, se transferirán todos los atributos de los subtipos al Supertipo y las interrelaciones que tuviesen los subtipos van al Supertipo.



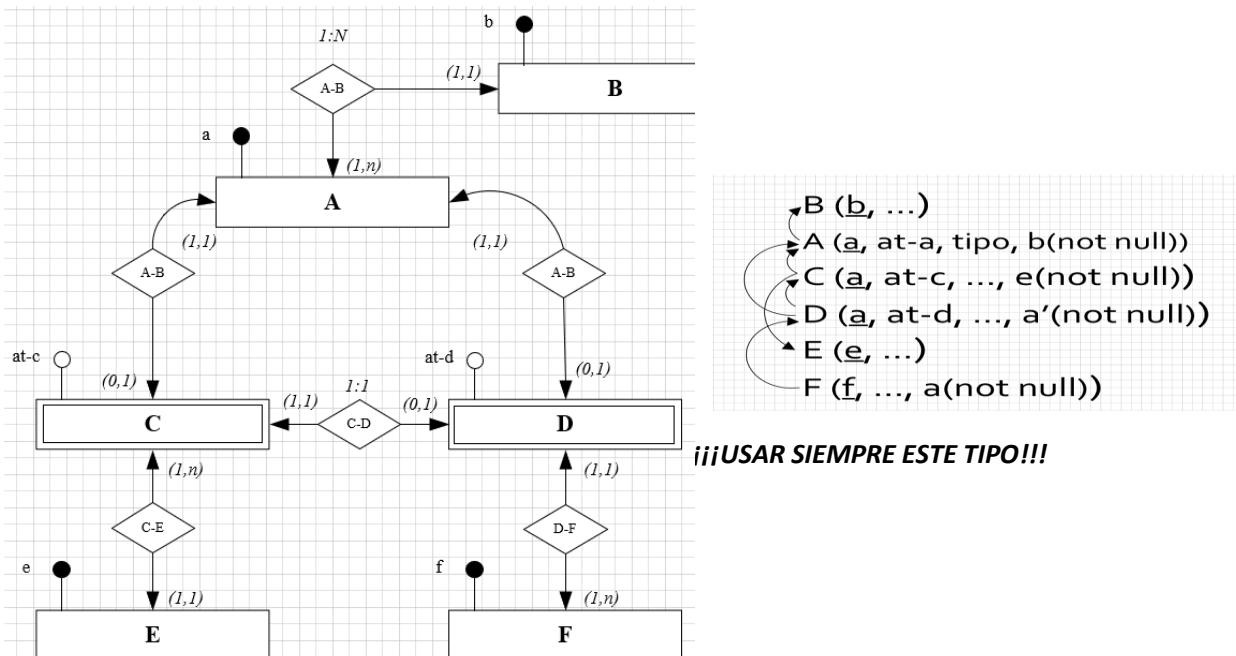
Se puede aplicar a totales/parciales, pero varía entre exclusivas e inclusivas.

↗ B (b, ...)
 ↗ A (a, at-a, tipo, at-c, at-d, b, a'(null), e(null) ...)
 ↗ E (e, ...)
 ↗ F (f, ..., a(not null))

PRTECAR 5: Eliminar jerarquía, transformándola en interrelación, manteniéndose los tipos de interrelación en los que intervienen tanto subtipos como Supertipo.

Si es exclusiva participan con cardinalidad mínima 0.

Si es inclusiva participan con cardinalidad 0 o 1.



RTECAR 1: Todas las entidades son transformadas en tablas.

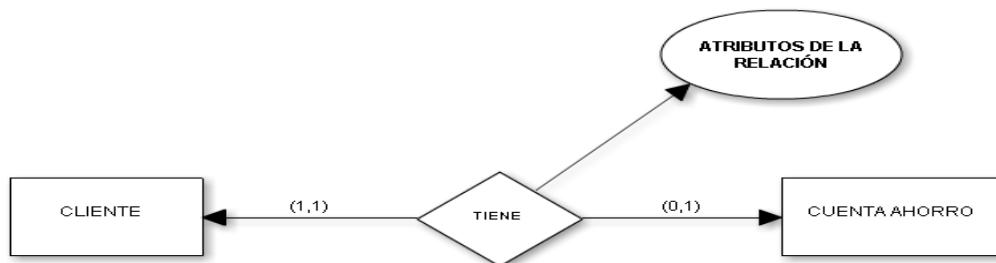
RTECAR 2.1 (1,1):

-Si tienen el mismo ID, todo a una tabla.

-Distinto ID, cada tabla tendrá su ID y el ID de la otra tabla como Foránea, para insertar datos se mete una transacción, y los atributos de la Relación van de las dos tablas.

RTECAR 2.2 (1,1) / (0,1) o (0,1)/(1,1):

El ID de la entidad que tiene cardinalidad máxima (1,1), pasa a la entidad que tiene cardinalidad mínima de manera foránea y alterna, no tomando valor nulo.



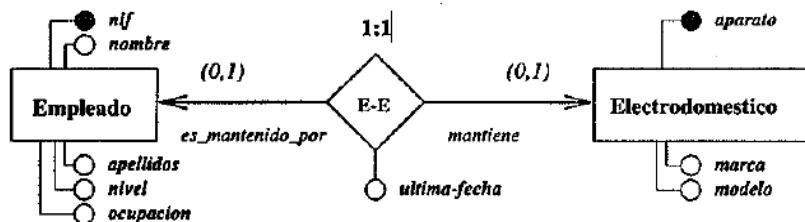
CLIENTE
PK DNI
Nombre Apellidos Fecha Nacimiento e-mail

CLIENTE CUENTA
PK,FK DNI PK,FK numCuenta ATRIBUTOS DE LA RELACIÓN

CUENTA
PK numCuenta Saldo Interés Fecha Apertura

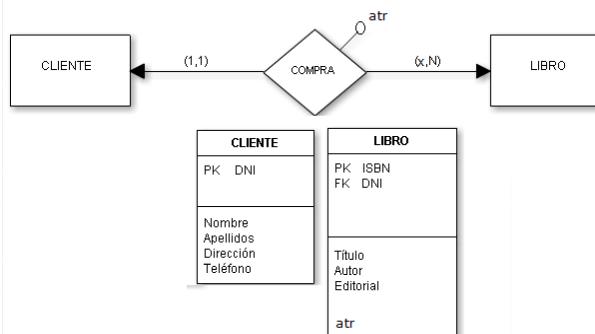


RTECAR 2.3 (0,1)/(0,1): Se genera una nueva tabla para la relación cuya ID será una de las dos PK, siendo la otra clave alterna y foránea.

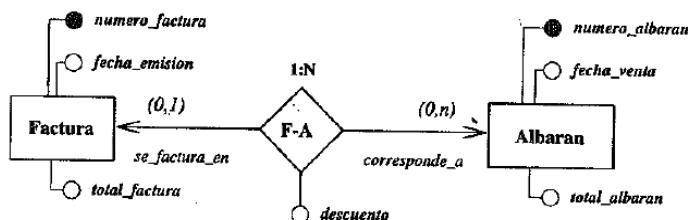


Empleado (nif, nombre, apellidos, nivel, ocupacion)
Electrodomestico (aparato, marca, modelo)
Empl_Elect (nif, aparato, ultima_fecha)

RTECAR 3.1 (x,n) / (1,1): El id de la entidad que tiene cardinalidad mínima (1,1) pasará a la máxima (x,n) como foránea y no podrá tomar valores nulos, los atributos de la relación pasarán a la tabla que participa con (x,n).



RTECAR 3.2 (0,1) / (0,n): Se crea una nueva entidad para representar la relación, cuya PK será el de la entidad con cardinalidad N, ambas claves serán foráneas y el ID de la entidad mínima nunca será nula.



Factura (numero_factura, fecha_emision, total_factura)
Albaran (numero_albaran, fecha_venta, total_albaran)
Fact_Alba (numero_albaran, numero_factura, descuento)

RTECAR 4: (x,n) / (y,n): Se crea una nueva entidad cuya PK será el agregado de las dos claves de cada tabla, siendo a su vez foráneas, incluyendo sus atributos.

16. Definir los tipos de integridad relacional.

Integridad de Clave: Ningún atributo que forme parte de la clave candidata podrá tomar valores nulos para ninguna tupla.

Integridad de referencia: Determina que todos los valores que toma una clave foránea deben ser valores nulos o valores que existen en la clave primaria que referencia. (FK).

Integridad de dominio: Son las restricciones impuestas por el propio problema.

17. Definiciones:

Clave candidata: Pueden ser claves primarias o alternas.

Clave alterna: Es aquella clave que no ha sido seleccionada como clave primaria, pero que puede identificar de forma única a una fila dentro de una tabla.

Dependencia funcional: Es una conexión entre uno o más atributos, que permite dado un valor X identificar sin ambigüedad a otro valor Y, tal que $X \rightarrow Y$, diciendo que, Y es funcionalmente dependiente de X, es decir el valor de un DNI (X) determina el valor del nombre, apellidos... (Y).

18. Algebra Relacional

Unión: Es la unión de R1 y R2, en una relación R3 cuyos componentes serán los de R1 y R2 sin repetición. **Select * from R1 UNION select * from R2.**

Diferencia: Es la resta de dos relaciones R1 y R2 en una relación R3 cuyos componentes serán los valores que estén en R1 y no estén en R2.

Select * From R1 Minus Select * from R2.

Selección: Se selecciona un conjunto de atributos de la relación dada

Select * from R where A=1

Proyección: Se crea un sub-esquema cuya extensión son las tuplas de R1.

Select R1.A from R1

Product: Se realiza el producto de dos relaciones R1 y R2. **Select * from R1, R2**

Intersección: Es la intersección de dos relaciones en una R3, cuya extensión serán las tuplas comunes a R1 y R2. **Select * from R1 intersect Select * from R2**

Join: Es la reunión de dos relaciones R1 y R2 en un R3, formada por las tuplas que resultan del producto de R1xR2 que satisfacen la condición.

select * from R1, R2 where R1.A=R2.C;

División: Se realiza la división en una R3 igual a la diferencia del esquema de R1 menos el de R2 y extensión todas las tuplas de R1 sin repetir que estén en R2.

19. Formas Normales

1 forma Normal: Esta forma **elimina los valores repetidos** en una BD, una tabla estará en 1FN si cumple:

- Todos los **atributos son atómicos**
- Tiene **una PK única.**
- La **PK no puede tomar valores nulos.**

2 forma Normal: Una relación esta en 2FN si esta previamente en 1FN y si los **atributos que no forman parte de ninguna clave dependen de forma completa de la PK**, es decir no existen dependencias parciales (Todos los atributos que no son clave principal deben depender únicamente de la PK).

Si la clave principal es simple, estará en 2FN, en caso de que sea una clave agregada, se estudiará la dependencia funcional con el resto de atributos.

Ejemplo: R(**dni+fecha**, nombre, ciudad, código_postal)

Si todos los atributos dependen de **dni+fecha** esta en 2FN, en caso de que **nombre**, dependa únicamente de DNI, deberá externalizarse a una nueva tabla.

3 forma Normal: La tabla se encuentra en 3FN si ya esta en 2FN y si **no existe ninguna dependencia funcional transitiva** entre los atributos no clave.

Ejemplo: R(**id**, pueblo, código_postal)

Tenemos una dependencia de **ID → código_postal** y a su vez también tenemos que **ID → pueblo**, sin embargo, **código_postal → pueblo**.

Por tanto, se externalizará a una nueva tabla: **R2(Código_postal, pueblo)**

Forma Normal Boyce-Codd (FNBC): La tabla estará en FNBC y cada determinante (atributo que determina completamente a otro), es clave candidata

20. Integridad Funcional

Asertos: Son reglas que deben cumplirse si o si, puede afectar a una tabla o a muchas.

```
CREATE ASSERTION <Nombre Aserto>
  CHECK (<Acción>);
```

Trigger: Permiten **autogestionar la BD**, si un producto aumenta su precio, que otro campo de la tabla modifique su valor también.

```
CREATE OR REPLACE TRIGGER <nombre trigger>
  ON <Nombre tabla>
  AFTER/BEFORE INSERT/UPDATE/DELETE
  AS
  BEGIN
    SQL A EJECUTAR
  END
```

21. Propiedades dependencias funcionales

Reflexiva: Dados los atributos "a" y "b" $\in R$, para los que se cumple que " $b \subseteq a$ " entonces, en la relación R, está presente una dependencia funcional de forma que " $a \rightarrow b$ ".

- **Aumento:** Dados los atributos a y b de una relación R en la que está presente la dependencia funcional " $a \rightarrow b$ " entonces también estará presente la dependencia funcional $(a+c) \rightarrow (b+c)$, siendo "c" cualquier otro atributo que forme parte de la intención de la relación R.

- **Transitiva:** Dados los atributos "a", "b" y "c" de una relación R en la que están presentes las dependencias funcionales " $a \rightarrow b$ " y " $b \rightarrow c$ ", entonces también estará presente la dependencia funcional " $a \rightarrow c$ "

A estas tres reglas descritas se les denomina Axiomas de Armstrong y de ellos deducimos otro conjunto de propiedades:

- **Unión:** Dados los atributos "a", "b" y "c" de una relación R en la que están presentes las dependencias funcionales

- " $a \rightarrow b$ " y " $a \rightarrow c$ ", entonces también estará presente la dependencia funcional " $a \rightarrow (b+c)$ "

- **Pseudo-transitiva:** Dados los atributos "a", "b", "c" y "d" de una relación R en la que están presentes las dependencias funcionales " $a \rightarrow b$ " y " $(b+c) \rightarrow d$ ", entonces también estará presente la dependencia funcional " $(a+c) \rightarrow d$ "

- **Descomposición:** Dados los atributos a, b y c de una relación R en la que está presente la dependencia funcional " $a \rightarrow b$ " y se cumple que " $c \subseteq b$ " entonces también estará presente la dependencia funcional " $a \rightarrow c$ "

$$a \rightarrow c$$