

Temario Fundamentos de Bases de Datos

Ismael Sallami Moreno

`ism350zsallami@correo.ugr.es`

`https://ismael-sallami.github.io/`

`https://elblogdeismael.github.io/`

Universidad de Granada

Licencia

Este trabajo está licenciado bajo una [Licencia Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/). <https://creativecommons.org/licenses/by-nc-nd/4.0/>

Usted es libre de:

- Compartir — copiar y redistribuir el material en cualquier medio o formato.

Bajo los siguientes términos:

- **Reconocimiento** — Debe otorgar el crédito adecuado, proporcionar un enlace a la licencia e indicar si se han realizado cambios. Puede hacerlo de cualquier manera razonable, pero no de una manera que sugiera que tiene el apoyo del licenciante o lo recibe por el uso que hace.
- **NoComercial** — No puede utilizar el material para fines comerciales.
- **SinObraDerivada** — Si remezcla, transforma o crea a partir del material, no puede distribuir el material modificado.



Índice general

1. Tema 1	5
2. Tema 2: Arquitectura de un SGBD	7
2.1. Lenguajes de una BD	7
3. Tema 3: Modelos de datos	9
3.1. Definición	9
3.2. Modelo Relacional	10
3.3. El modelo de datos relacional. ED. Integridad	12
3.3.1. La estructura de datos relacional	12
3.3.2. Restricciones o reglas de integridad	12
3.4. Otros modelos de datos	16
3.4.1. Modelo jerárquico	16

Capítulo 1

Tema 1

Capítulo 2

Tema 2: Arquitectura de un SGBD

2.1. Lenguajes de una BD

Recomendación ANSI/SPARC. Se propone un lenguaje de base de datos. En este se debe de definir, controlar y manipular los datos. Se denomina sub-lenguaje de la base de datos al que esta implementado por el propio SGBD. Tenemos distintas partes:

- DDL: definición de estructura de datos.
- DML: modificación, borrado y consulta de datos, además, se permite consultar los esquemas definidos de la Base de Datos.
- DCL: gestiona los requisitos de acceso a los datos y otro tipo de tareas de administración(creación de usuarios,...).

Este grupo propone que haya cada uno de los anteriores en cada uno de los niveles de la arquitectura. Esto no tiene mucho sentido. Esto se debe a que no tiene sentido tener alguno de los anteriores en determinados niveles. Si meto el DDL en el nivel interno, estoy dependiendo del nivel interno de la máquina.

En la realidad ha surgido la idea de que todo sea un estándar, pero cada fabricante lleva sus propias estrategias. Hay diferentes versiones de SQL y los SGBD han ido proporcionando soporte.

Acto seguido, nos centramos en el desarrollo de aplicaciones. Son de propósito general, ya que usan lenguajes como C++, Java,... Se usan diversas herramientas como es Oracle APEX, Sysbase PowerBuilder,... Con esto se proporciona un procesamiento avanzado de datos y una gestión eficaz de la interfaz del usuario.

Debemos de establecer un mecanismo que nos permite establecer una comunicación entre el lenguaje anfitrión y el de datos. Esto se conoce como *acoplamiento*. Distinguimos dos categorías.

- Débilmente acoplados. Lenguajes de propósito general y en este caso el programador puede distinguir entre sentencias del lenguaje anfitrión y las dispuestas por la propia BD.
- Fuertemente acoplados. Lenguajes y herramientas de propósito específico. Se parte del DSL como elemento central y se le van incorporando características para facilitar el desarrollo de aplicaciones.

Hay varias alternativas para implementar el acoplamiento débil:

- **Usar APIs de acceso a la BD.** Acceder a la BD desde el código fuente del lenguaje anfitrión.
- **DSL inmerso en el código fuente del lenguaje anfitrión.** Se escribe código híbrido.

Pasa lo mismo con el acoplamiento fuerte:

Las propuestas son ya propietarias, o bien la ejecución de Java sobre una máquina virtual que esta en el propio SGBD.

También han aparecido numerosos entornos de desarrollo que son específicos para las aplicaciones de gestión.

Capítulo 3

Tema 3: Modelos de datos

3.1. Definición

Previamente, debemos de llevar a cabo un análisis de los datos y de esta manera obtenemos el esquema conceptual y lógico de la BD. En este tema entre la siguiente fase de *Diseño*, dando como resultado el modelo lógico. Este se trata de uno ya implementado, por eso se dice que es un modelo implementable.

En cuanto al proceso de transformación, distinguimos:

1. Mundo real
2. Datos operativos
3. Esquema conceptual

En esta etapa se introduce lo que conocemos como *tablas*. En la última fase llevamos a cabo la implementación de esa tabla mediante un lenguaje.

Definición: Mecanismo formal para representar y manipular la información con la que voy a trabajar. Debe de constar de datos, operaciones y reglas de integridad.

La necesidad de usar el modelo de datos son:

- Se usan lenguajes de definición de datos.
- Es de muy bajo nivel.
- Se necesitan niveles superior para su abstracción.

El objetivo que se tiene es establecer que representen los datos y que los describan de una forma entendible y manipulable. Según ANSI distinguimos tres niveles:

- Externo
- Conceptual
- Interno

En cuanto a la clasificación:

- Basados en registros.
- Basados en objetos.
- Físico.

Los dos primeros son de nivel externo y conceptual, mientras que el último es de nivel interno.

3.2. Modelo Relacional

La información se organiza en tablas:

1. La estructura de almacenamiento son las tablas.
2. Integridad.
3. Consulta y Manipulación.

Esta tabla es a nivel lógico, en cuanto al nivel físico depende de varias estructuras, como son las listas enlazadas,...

Elementos a conocer:

- Atributo.
- Dominio: rango de valores que puede tomar (enteros, letras,...)
- Relación: Se conoce como cualquier subconjunto del producto cartesiano $D_1 \times D_2 \times \dots D_n$. En este caso a relación se refiere a la tabla. Se denota como $R(A_1..A_n)$. De esta manera conseguimos todos los valores que pueden tomar y de aquí sacamos los datos (solo a nivel teórico).
- Tupla: filas de la tabla.
- Cardinalidad de una relación: número de tuplas.
- Esquema de una relación R: Atributos de la relación junto a su dominio.
- Grado de una relación: Número de atributos. Aunque se supone que es invariable, esto depende ya que se puede modificar.
- Instancia de una relación: datos que tengo en un determinado momento.
- Esquema de la BD: colección de esquemas de relaciones junto con las restricciones de integridad.
- Instancia o estado de una BD: colección de instancias de relaciones que verifican las condiciones de integridad.
- BD relaciones: instancia de la BD junto con su esquema.

Condición de Normalización

- Todos los valores de los atributos de una relación son atómicos¹.
- Valor atómico es un valor no estructurado.
- Cuando una relación cumple la condición de normalización se dice que está en Primera Forma Normal.

¹No se pueden dividir más

Consecuencias

- No hay valores tipo conjunto.
- No hay valores tipo registro.
- No hay valores tipo tablas.

Problema

Todas las representaciones son extensivas, no se puede representar información del tipo “el valor del atributo asignaturas de un alumno es: FBD, ALG, LD”.

Consecuencias de la definición

- No hay tuplas duplicadas:
 - Por la definición conjuntista de relación.
- No hay orden en las filas ni en los atributos:
 - Al no estar ordenados ni los atributos ni las filas (conjuntos) el acceso es por nombre de atributo y valor.
- Varias instancias representan la misma relación.

A	B	C	D	E
a1	b1	c1	d1	e1
a1	b2	c2	d2	e1
a2	b1	c3	d3	e1
a2	b1	c4	d3	e1
a3	b2	c5	d1	e1

A	B	C	D	E
a2	b1	c4	d3	e1
a2	b1	c3	d3	e1
a1	b2	c2	d2	e1
a3	b2	c5	d1	e1
a1	b1	c1	d1	e1

En cuanto a las representaciones podemos distinguir:

- Representación física: Archivos, registros y campos.
- Representación intuitiva: Tabla, filas y columnas.
- Modelo matemático: Relación, tuplas y atributos.

3.3. El modelo de datos relacional. ED. Integridad

3.3.1. La estructura de datos relacional

Notación a utilizar

- Relación: R, S, T....
- Atributos: A, B, ...
- Esquema de relación: R[A1, A2, ..., An]
- Instancia de relación R: r
- Tuplas de una instancia: x1, x2, ..., \in r
- Valor de un atributo Ai en una tupla xj: xj[Ai] ó Aij

Valores nulos

- Algunas veces no se conoce el valor de un atributo para una determinada tupla. En esos casos a ese atributo de esa tupla se le asigna un valor nulo (NULL).
- Un valor nulo puede ser:
 - Un valor desconocido.
 - Un atributo no aplicable.
- En cualquier caso, ese valor es un valor más de todos los dominios de la base de datos.

3.3.2. Restricciones o reglas de integridad

- Son condiciones para preservar la semántica de una base de datos.
- Específicas del problema:
 - $0 \leq \text{edad} \leq 100$
 - $\text{créditos} > 0$
 - $\text{carácter} \in \{\text{'troncal'}, \text{'obligatoria'}, \text{'optativa'}, \dots\}$
- Propias del papel de los atributos en el esquema:
 - $\text{imparte.NRP} \in \text{profesor.NRP}$ (un profesor inexistente no puede impartir una asignatura)
 - $\text{cod_asig} \neq \text{nulo}$ (siempre debe conocerse el código de una asignatura)

Superclaves y claves (candidatas y primarias)

- Superclave: Cualquier conjunto de atributos que identifica unívocamente a cada tupla de una relación.
- Clave de una relación: superclave minimal.
- Por ejemplo, en la relación Asignaturas:
 - El conjunto de atributos {Cod_Asig, Nombre} identifica unívocamente cada tupla.
 - Sin embargo, no es minimal y no puede considerarse como una clave.
 - Cod_Asig por sí sola, es una clave.
- En una relación dada puede que más de un conjunto de atributos puedan ser elegidos como clave. Estos conjuntos de atributos se llaman claves candidatas.
- Cuando hay más de una clave candidata, hay que seleccionar una como principal. Esta clave recibe el nombre de clave primaria de la tabla.
- Criterio de selección: Tamaño, significado, **capacidad para recordarla**², fusión con otras tablas, etc.

Clave candidata y primaria (definición formal)

- Sea $R[A_1, A_2, \dots, A_n]$, $CC \subseteq \{A_1, A_2, \dots, A_n\}$ se denomina clave candidata si y solo si:
 - **Unicidad:** $\forall r$ instancia de R y $\forall t_1, t_2 \in r$ con $t_1 \neq t_2 \Rightarrow t_1[CC] \neq t_2[CC]$ ³
 - **Minimalidad:** No existe $CC' \subsetneq CC$ que verifique la unicidad.
- Una clave candidata es un atributo o conjunto de atributos que identifican a cada tupla en la relación y que, además, no existe un subconjunto de ellos que también identifiquen a cada tupla de la relación.
- Una clave primaria es una clave candidata elegida por el diseñador.
- Si CC verifica la unicidad pero no la minimalidad se denomina superclave.
- Completamos la notación para describir una relación, subrayando los atributos que forman su clave primaria etiquetándolos con CP. Si existen otras claves candidatas también se subrayan etiquetando el subrayado con CC:
 - Socio (#Socio_{CP}, DNI_{CC}, Nombre, Dirección)⁴

²Ejemplo de ello, es cuando vas a secretaría a preguntar por algo, donde te preguntan por tu DNI, en vez de N° de expediente.

³No puede coincidir dos tuplas distintas.

⁴En este caso estamos denotando como se referencia en la práctica.

Claves externas

- Conjunto de atributos en una relación que es una clave en otra (o incluso en la misma) relación.
- Podemos ver una clave externa como un conjunto de atributos de una relación cuyos valores en las tuplas deben coincidir con los valores de la clave primaria de las tuplas de la otra relación.

Claves externas

- Consideramos una relación R (relación que referencia) y un subconjunto de atributos de su esquema CE (clave externa); y una relación S (relación referenciada) cuya clave primaria CP coincide con CE.
- Si CE es una clave externa de R que referencia a CP en S, entonces: $\forall t \in r, \exists t' \in s$ tal que $t[CE] = t'[CP]$, donde r y s son las instancias de R y S en la base de datos.
- Eventualmente, R y S pueden ser la misma relación.
- Ejemplo: Imparte (NRP, Cod_Asig) referencia a Profesor (NRP).

Dominio activo

- Subconjunto de valores del dominio de un atributo de una relación que está presente en la instancia de la relación⁵.
- Dada una relación R, su instancia r y un atributo A de R, el dominio activo de A es el siguiente conjunto:

$$\{a \mid \exists t \in r \text{ tal que } t[A] = a\}$$

- Podemos extender el concepto de dominio activo a un conjunto de atributos.
- Dada una relación R, su instancia r y un subconjunto de atributos CA de R, el dominio activo de CA es el siguiente conjunto:

$$\{(a_1, a_2, \dots, a_n) \mid \exists t \in r \text{ tal que } t[CA] = (a_1, a_2, \dots, a_n)\}$$

- Las claves externas establecen relaciones de inclusión entre los dominios activos de la clave externa y la clave referenciada.
- Puede haber más de una clave externa en una relación.
- Puede haber una clave externa a la clave primaria de la propia relación⁶.
- El resultado de que haya CE con una misma relación es el hecho de la *fusión de tablas*.

⁵Es decir, en un momento dado.

⁶No puede haber dos claves en la misma sección que se llamen igual, por lo que si en una relación nos encontramos con dos DNI, debemos de denotar uno de manera distinta, o bien usar una notación distinta.

Conceptos generales

- Condiciones de integridad:
 - Normas que mantienen la corrección semántica de una base de datos.
- Nos centramos en la integridad genérica: depende del papel que juegue un atributo en el diseño de la tabla:
 - Son metarreglas: generan reglas de integridad aplicadas a una base de datos concreta.
 - Existen la integridad de entidad y la integridad referencial.

Integridad de entidad

- No se debe permitir que una entidad sea representada en la base de datos si no se tiene una **información completa** de los atributos que son clave primaria de la entidad.
- Un atributo que forma parte de la clave primaria de una tupla en una relación, **no puede tener un valor nulo**⁷.

Integridad referencial

- Una base de datos en la que **todos los valores no nulos de una clave externa referencian valores reales de la clave referenciada** en la otra relación, cumple la regla de integridad referencial. Si esto se cumple, estamos cumpliendo con la regla de *integridad referencial*.
- Si una relación incluye una clave externa conectada a una clave primaria, el valor de la clave externa debe ser, o bien igual a un valor ya existente en el dominio activo de la clave primaria, o bien completamente nulo (si la semántica lo permite).
- La integridad referencial mantiene las conexiones en las bases de datos relacionales.

Restricciones del SGBD

EL SGBD debe encargarse de mantener las siguientes restricciones:

- La unicidad de la clave primaria y de las claves candidatas:
 - Frente a operaciones de inserción y actualización, el SGBD debe rechazar los valores introducidos que sean iguales a los presentes en la BD para los atributos que el diseñador ha definido como clave primaria y como claves candidatas.
- La integridad de identidad:

⁷Esto define la integridad de la tabla.

- Frente a operaciones de inserción y actualización, el SGBD debe rechazar las modificaciones que asignen un valor NULO a algún atributo de la clave primaria.
- La integridad referencial:
 - En inserciones:
 - Rechazar la tupla insertada si el valor de la clave externa no concuerda en la relación referenciada para alguna tupla en el valor su clave primaria, es decir, si no es nulo o bien no existe.
 - Si el valor para la clave externa es NULO y el diseño no lo permite, habrá que *rechazar* también esa inserción.
 - En borrados:
 - Si se borra la clave primaria en la relación referenciada, el diseñador puede establecer en el SGBD una de las siguientes alternativas de actuación:
 - ◊ Rechazar el borrado de la tupla.
 - ◊ Eliminar en cadena todas las tuplas que la referencian.
 - ◊ Poner valor nulo en la clave externa de todas esas tuplas.
 - En actualizaciones:
 - Si se actualiza la clave externa, actuar con el nuevo valor actualizado como en el caso de inserción.
 - Si se actualiza la clave primaria de la relación referenciada y el antiguo valor estaba referenciado en alguna relación, actuar como en el caso de borrado.

3.4. Otros modelos de datos

3.4.1. Modelo jerárquico

Fue el primero en implementarse físicamente:

- Nivel externo: aplicaciones Cobol o lenguaje del sistema.
- No había interactividad:
 - Carecía de un lenguaje de consulta.
- Estructura de datos básica:
 - Árbol:
 - Registro maestro: raíz.
 - Registros secundarios: dependen de los anteriores.
 - La BD es una colección de instancias de árboles.
 - Esta estructura plasma de forma muy directa:
 - Relaciones muchos a uno.

- Relaciones uno a uno.
- Para relaciones muchos a muchos:
 - Hay que duplicar toda la información sobre las entidades involucradas.

Bibliografía

- [1] Ismael Sallami Moreno, **Estudiante del Doble Grado en Ingeniería Informática + ADE**, Universidad de Granada, 2025.