

Bases de Datos I

Autor: Camilo Cardona Patiño



Bases de Datos I / Camilo Cardona Patiño, / Bogotá D.C., Fundación
Universitaria del Área Andina. 2017

978-958-5460-28-7

Catalogación en la fuente Fundación Universitaria del Área Andina (Bogotá).

© 2017. FUNDACIÓN UNIVERSITARIA DEL ÁREA ANDINA
© 2017, PROGRAMA INGENIERIA DE SISTEMAS
© 2017, CAMILO CARDONA PATIÑO

Edición:

Fondo editorial Areandino

Fundación Universitaria del Área Andina

Calle 71 11-14, Bogotá D.C., Colombia

Tel.: (57-1) 7 42 19 64 ext. 1228

E-mail: publicaciones@areandina.edu.co

<http://www.areandina.edu.co>

Primera edición: noviembre de 2017

Corrección de estilo, diagramación y edición: Dirección Nacional de Operaciones virtuales

Diseño y compilación electrónica: Dirección Nacional de Investigación

Hecho en Colombia

Made in Colombia

Todos los derechos reservados. Queda prohibida la reproducción total o parcial de esta obra y su tratamiento o transmisión por cualquier medio o método sin autorización escrita de la Fundación Universitaria del Área Andina y sus autores.

Bases de Datos I

Autor: Camilo Cardona Patiño



Índice

UNIDAD 1 Introducción a los sistemas de información

Introducción	9
Metodología	10
Desarrollo temático	11
Introducción a los sistemas de información	11
Elementos de un sistema de información	11
Tareas que realiza un sistema de información	12
Sistemas de información	13
Tipos de sistemas de información	15

UNIDAD 1 Definición de base de datos

Introducción	19
Metodología	20
Desarrollo temático	21
Definición de base de datos	21
Sistemas de gestión de bases de datos	21
Arquitectura de sistemas de bases de datos	24



Índice

UNIDAD 2 Diseño de bases de datos

Introducción	29
Metodología	30
Desarrollo temático	31
Diseño de bases de datos	31
Etapas para el diseño de bases de datos	31
Modelos de datos	33
Tipos de modelos de bases de datos	34
Restricciones de integridad	36

UNIDAD 2 Normalización

Introducción	39
Metodología	40
Desarrollo temático	41
Normalización	41
Dependencia incoherente	41
Claves o llaves	43
Dependencias	44



Índice

UNIDAD 3 Álgebra relacional

Introducción	53
Metodología	54
Desarrollo temático	55
Álgebra relacional	55
Unión	56
Intersección	56
Diferencia	56
Producto cartesiano	57
Alumnos × Materias	58
Selección	59
Proyección	61
Combinación o Join	61

UNIDAD 3 Cálculo relacional

Introducción	64
Metodología	65
Desarrollo temático	66
Cálculo relacional	66
Tipos de cálculo relacional	67
Lógica de primer orden	67



Índice

UNIDAD 4 Lenguaje SQL

Introducción	71
Metodología	72
Desarrollo temático	73
Lenguaje SQL	73
DDL: Lenguaje de definición de datos	73
Comando Create	74
Comando Drop	74
Comando Alter	75
DML: Lenguaje de manipulación de datos	75
Comando select	76
Comando Update	76
Comando Insert	77
Comando Delete	77

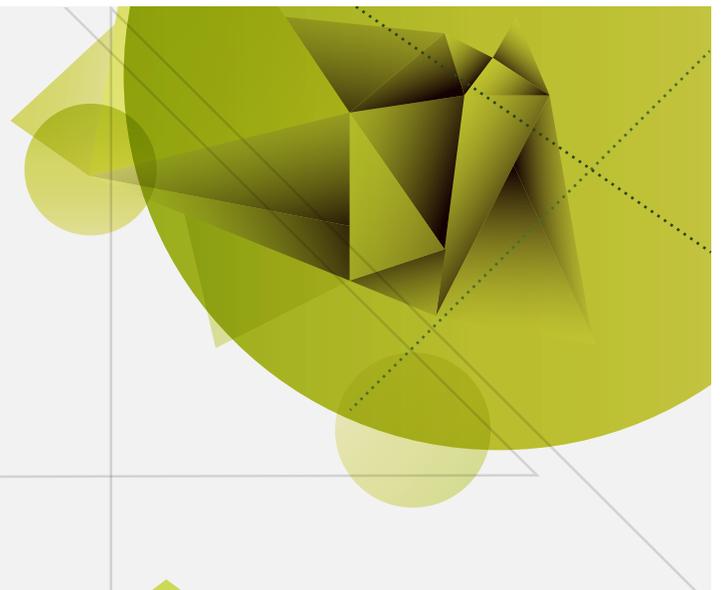
UNIDAD 4 Vistas

Introducción	79
Metodología	80
Desarrollo temático	81
Vistas	81
Creación de vistas	83
CRUD Operaciones con registros	84
Bibliografía	86

1

Unidad 1

Introducción a
los sistemas de
información



Bases de Datos I

Autor: Camilo Cardona Patiño

Introducción

Actualmente, las bases de datos juegan un papel fundamental en el desempeño y competitividad de las empresas, proveen la capacidad de analizar los aspectos relevantes al momento de tomar decisiones, es por ello, que vemos cada vez un mayor interés en mejorar los procesos y prácticas relacionados con el manejo de la información.

Los sistemas de información son los encargados de soportar, mantener, mejorar y aumentar los diferentes flujos de información generados al interior y al exterior de la organización, esta realidad ha obligado a que cada vez más empresas estén dispuestas a invertir grandes cantidades de dinero en tecnología, pues han visto los beneficios y ventajas que ofrecen.

Se invita al estudiante a leer completamente la guía de manera que logre alcanzar gran solidez en cuanto a fundamentación teórica, de igual forma, se le recomienda analizar los ejemplos planteados, siguiendo paso a paso el desarrollo a fin de lograr claridad y comprensión en los conceptos, estar atento de la realización de las actividades, participación en los foros y formular todas las dudas que pueda tener en relación al tema tratado.

Introducción a los sistemas de información

Un sistema de información son un conjunto de elementos que interactúan entre sí para la consecución de un objetivo, para el caso especial de sistemas de información, debemos aclarar que las actividades realizadas por el sistema están encaminadas a la recopilación, administración, almacenamiento y recuperación de información, es importante aclarar que tal sistema no debe ser necesariamente del orden computacional, ya que también es posible llevar un sistema de información en papel, sin embargo existen grandes y claras ventajas de emplear sistemas computarizados, especialmente para el manejo de grandes volúmenes de información.

Los sistemas de información herramientas de apoyo para quien las usa, así, por ejemplo, los sistemas permiten la toma de decisiones a nivel de gerencia, también lo usan otras áreas como la financiera, la de recurso humano, incluso la de sistemas.

Elementos de un sistema de información

Ahora, analicemos los diferentes elementos del Sistema de información que interactúan en conjunto:

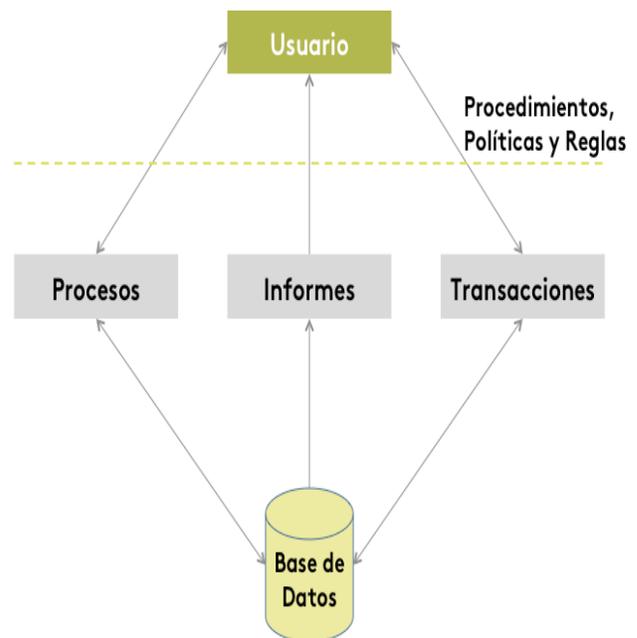


Figura 1. Elementos de un Sistema de Información
Fuente: Propia.

Usuario

Hace referencia a las personas que interactúan con el sistema de información, abarcando desde el operador que tiene la misión de recopilar, ordenar y cargar la información, hasta el director o ejecutivo que emplea tal información ordenada para la toma de decisiones.

Procedimientos, políticas y reglas

Son una serie de procedimientos que son definidos por la organización, el administra-

dor o el dueño de la información, los cuales buscan principalmente la protección, la integridad y el aseguramiento de la información, definiendo la manera en que el usuario puede interactuar con la información bajo una serie de controles y verificaciones.

Procesos

Se trata de los elementos que contienen un método o lógica que permite la lectura de información desde la base de datos, esta lógica permite recuperar, actualizar, modificar y crear información en los registros, estos procesos son creados y/o manipulados por el usuario.

Informes

Los informes son salidas ordenadas y controladas de información, resultado de la aplicación de filtros y criterios de búsqueda por parte del usuario o de procesos automáticos.

Transacciones

Pertenece a las opciones programadas que permiten la interacción entre el usuario y la interfaz para realizar modificaciones, agregaciones, consultas y borrado de valores o registros.

Base de datos

La base de datos es el almacén que contiene toda la información en forma ordenada y de fácil acceso y recuperación, tal información permite la toma de decisiones a nivel gerencial.

Tareas que realiza un sistema de información

En términos generales, se puede decir que existen cuatro diferentes actividades que realiza un Sistema de Información, las cua-

les están encaminadas a la generación, administración y recuperación de la información veamos:

- **Entrada de información:** o captura, es el proceso mediante el cual, los datos son ingresados al sistema.
- **Almacenamiento de información:** es la acumulación de los datos, esta puede hacerse en medios electrónicos como los computadores o físicos como registros en papel a fin de conservar la información.
- **Procesamiento de la información:** son los procesamientos de los datos que ordenan, cruzan y almacenan sistemáticamente la información, permitiendo su fácil recuperación en el futuro.
- **Salida de información:** se trata de los reportes e informes que se producen de la base de datos, estos resultados permiten observar tendencias, gustos o preferencias.



Figura 2. Tareas de un Sistema de Información
Fuente: Propia.

Actores involucrados en un sistema de información

Los sistemas de información fueron diseñados para contener información sobre una empresa y su entorno, por ello, encontra-

mos diferentes actores contenidos dentro de su estructura y funcionamiento como: accionistas, clientes, proveedores, entida-

des reguladoras los cuales se relacionan de una u otra manera con la empresa u organización.

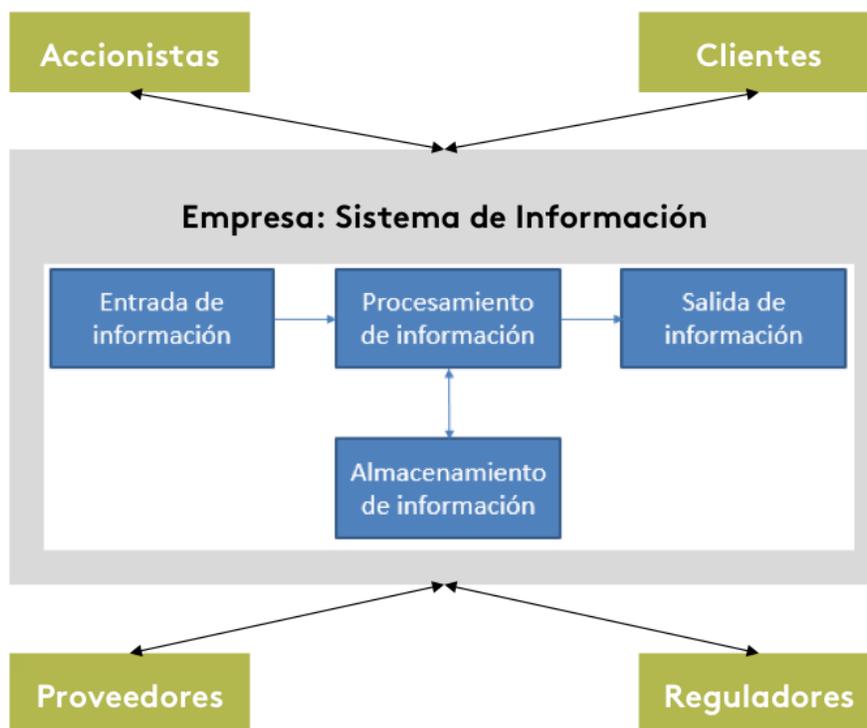


Figura 3. Actores
Fuente: Propia.

Sistemas de información

De acuerdo a su especialidad y aplicación, los sistemas de información reciben una clasificación, ya que no es posible que un solo sistema brinde toda la información que una organización puede necesitar.

- **Sistemas de información estratégicos**
Son los sistemas que asisten a los altos directivos a enfrentar y resolver aspectos estratégicos y tendencias a largo plazo, a nivel de organización o ambiente interno, pero también a nivel de ambiente externo. Busca ayudar en la armonización del comportamiento, la capacidad y las decisiones de corto y largo plazo.

- **Sistemas de información administrativos**

Se emplean en los procesos de control, administración, supervisión, toma de decisiones, permite determinar el estado de los procesos en las organizaciones. Por lo general, con este tipo de sistemas es posible la generación de informes habituales aunque también es posible obtener información instantánea de las operaciones.

Apoyan a las decisiones no rutinarias y tienden a enfocarse en decisiones menos estructuradas para las cuales los requisitos de información no siempre son claros.

■ **Sistemas de información del conocimiento**

Se encargan de acumular grandes cantidades de conocimiento, facilitando los procesos de aprendizaje y avance, en él se reflejan las experiencias aprendidas a lo largo del desempeño organizacional. Estos tipos de sistemas están entre las aplicaciones de crecimiento más rápidas en los negocios actuales.

■ **Sistemas de información operativos**

Ofrecen colaboración a los directivos para realizar búsqueda de las actividades y las transacciones básicas de la organización como ventas, ingresos, transacciones bancarias, nómina, solicitudes de crédito y flujo de materia prima.

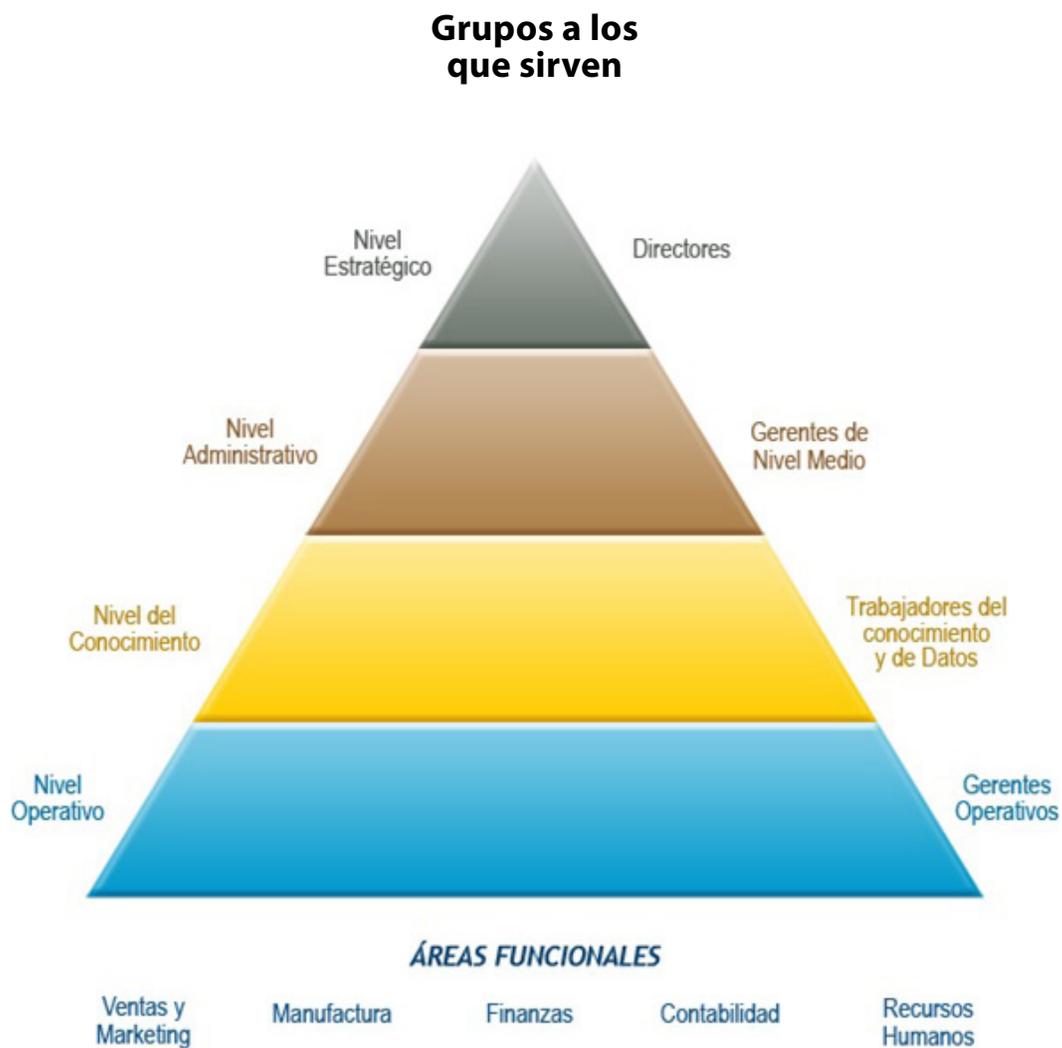


Imagen 1. Tipos de Sistemas de Información (2004)

Fuente: http://biblioteca.itson.mx/oa/dip_ago/introduccion_sistemas/img/piramide.jpg

Tipos de sistemas de información

Dentro de la tipología de los sistemas de información, podemos reconocer 7 principales tipos.

1. SPT o Sistemas para el Procesamiento de Transacciones

Este sistema cambia el procedimiento de llevar registros manuales y los reemplaza con los medios electrónicos para la gestión de la información. Opera mediante rutinas y algoritmos claros y repetibles.

Está basado en la computadora y la relación de los trabajos rutinarios, es el más importante y el más utilizado dentro de la empresa, pues reduce el tiempo de las operaciones o actividades rutinarias de la empresa.

Características de SPT:

- Generalmente son de los primeros sistemas de información con los que la empresa se atreve a incursionar en el área de sistemas informáticos para la gestión de la información.
- Ofrecen apoyo a las áreas operativas.
- Con el tiempo, se evidencian ahorros en mano de obra y mejoras en productividad.
- Se caracteriza por realizar cálculos y operaciones simples.

2. SSD o Sistemas de Soporte para la Decisión

Es un sistema de interacción fundamentado en el procesamiento computarizado, ofrece un enfoque hacia la toma de decisiones, empleando datos y modelos para ofrecer respuestas a problemas no estructurados.

Este tipo de sistemas no pretende eliminar el criterio, experiencia e intuición humana, pero con su ayuda, las personas cuentan con más herramientas y mayores argumentos para fundamentar las diferentes opciones que se tienen.

3. SSDG o Sistemas de Soporte para la Decisiones en Grupo

A diferencia del SSD, este enfoque involucra a diferentes niveles y personas, las cuales pueden compartir opiniones, argumentos y conocimientos, para afinar desde varios ángulos la justificación de una decisión importante. Permite compartir documentos y contenido multimedia, ofrece herramientas para la gestión de citas y reuniones.

Características de SSDG:

- Flexibilidad.
- Uso fácil e intuitivo.
- Ofrece apoyo a la toma de decisiones.
- Cuenta con multi-disciplinabilidad.
- Gestión automática de registros.

Elementos de SSDG:

- Moderador de diálogo.
- Modelos de coordinación
- Base de datos

Alternativas de SSDG:

- Conferencias virtuales.
- Conferencias presenciales.
- Redes de decisiones.

4. STC o Sistemas de Trabajo con Conocimientos

Son el tipo de sistemas empleados en las estaciones de trabajo, de manera que cumplan con el objetivo de reco-

pilar, canalizar y enrutar la mayor parte de la información que la organización obtiene de su entorno, acá podemos observar flujos de datos provenientes de los clientes y los colaboradores principalmente.

Cuando se observa el conocimiento como activo intangible es difícil de administrar, incluso apenas se está comprendiendo cómo se puede administrar; el conocimiento es parte fundamental

para las organizaciones y su forma de hacer negocios y tener ventajas competitivas. Dichas herramientas nos ayudan a identificar o clasificar nuestros activos de conocimiento y a llevar los procesos de su gestión de una manera más efectiva, todas las habilidades y los conocimientos deben de ser identificados y valorados, accesibles desde cualquier sitio, deben ser capturados o almacenados, para que a su vez se puedan desarrollar y mejorar.

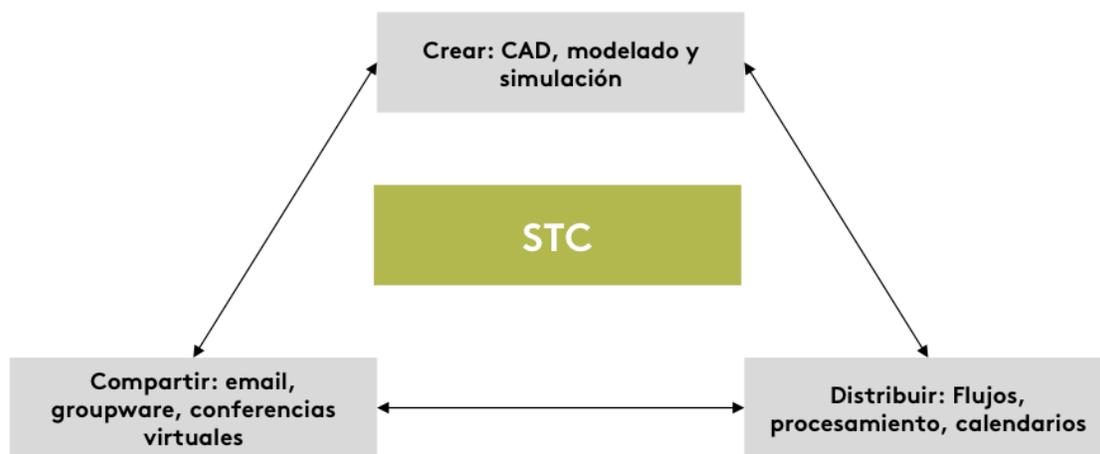


Figura 5. Estructura STC
Fuente: Propia.

5. SAO o Sistemas de Automatización de Oficinas

Brinda comunicación y coordinación entre las actividades desarrolladas por un grupo de personas o áreas funcionales, su objetivo es incrementar la productividad en la manipulación de documentos.

Características de SAO:

- Conservan el KNOW HOW de la organización.

- Permite conectar y coordinar las actividades.
- Es aplicable a todos los niveles de conocimiento.

Actividades de SAO:

- Administración de documentos.
- Programación de las actividades de las personas y grupos.
- Comunicación con personas y grupos.

- Administración de los datos.
- Administración de proyectos.

6. SIA o Sistemas de Información para la Administración

Proveen información general a los administradores, también apoyan la toma de decisiones, incluyen, procedimientos, Software, bases de datos y dispositivos de ayuda.

Es posible obtener informes del comportamiento en pro de mejorar la efectividad de los procesos.

Características de SIA:

- Entregar información general en forma periódica, programada y constante.
- Suministrar una forma de comunicación transversal a todas las áreas.
- Soportar el proceso de planeación.
- Apoyo en la toma de decisiones.

7. SIE o Sistemas de Información Estratégicos

Son sistemas diseñados para apoyar a la alta gerencia, abarca situaciones y comportamientos globales, permite establecer tendencias y realizar proyecciones tanto del entorno interno como del externo.

Características de SIE:

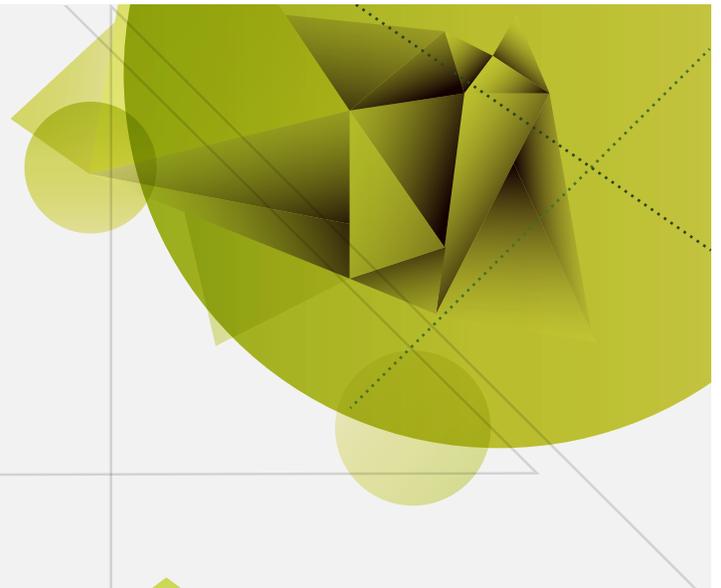
- Apoyan el proceso de toma de decisiones.
- Esta enfocados en proveer información necesaria para el cumplimiento de metas estratégicas.
- Son de vital importancia para el desempeño del negocio.
- Son la fuente de grandes cambios y reformas al interior de la organización.



1

Unidad 1

Definición de base
de datos



Bases de Datos I

Autor: Camilo Cardona Patiño

Introducción

En la presente cartilla veremos la definición de bases de datos, los sistemas de gestión de bases de datos, entendiendo la manera como permiten la creación, administración y modificación de un banco de datos, veremos también la arquitectura de bases de datos y sus características.

Aprenderemos las estructuras básicas que existen para modelar bases de datos, las funciones que deben cumplir para garantizar la integridad, la seguridad y la consistencia de la información.

Se invita al estudiante a leer completamente la guía de manera que logre alcanzar gran solidez en cuanto a la definición de Bases de datos, sus características, así como a la arquitectura de sistemas de bases de datos presentados, se le recomienda revisar los contenidos adicionales a fin de lograr claridad y comprensión, estar atento de la realización de las actividades, participación en los foros y formular todas las dudas que pueda tener en relación al tema tratado.

Definición de base de datos

Una base de datos o banco de datos son una serie de datos organizados que mantienen algún tipo de relación entre sí, la información es acumulada y aprovechada por las organizaciones.

Algunas características:

- Mínima redundancia.
- Los datos poseen independencia lógica y física.
- Capacidades de respaldo y recuperación de la información.
- Soporte a múltiples usuarios simultáneamente.
- Administración por medio de lenguajes de programación específicos.
- Posibilidad de realizar consultas complejas.
- Aseguramiento de la integridad de los datos.
- Control de acceso.

Sistemas de gestión de bases de datos

Un sistema de gestión de bases de datos o SGBD es un tipo de aplicación y Software

encargado de ofrecer una interfaz de comunicación entre el usuario, las aplicaciones y la base de datos, este software permite la implementación de políticas de seguridad, controles de acceso, mejora de las consultas, construcción, manipulación de datos, en general, toda la creación y accesos a la base de datos, generalmente está compuesto por un lenguaje de programación dividido en tres componentes:

DDL: Lenguaje de definición de datos.

DML: Lenguaje de manipulación de datos.

DCL: Lenguaje de control de datos.

Veamos más detalles:

DDL Lenguaje de definición de datos (Data Definition Language)

Permite definir las estructuras que almacenarán los datos y los procedimientos o funciones para realizar las consultas.

Al principio fue usado el referente a Codayl, donde se establecía el esquema de la base de datos usando descripciones de los registros y campos del modelo. Posteriormente se utilizó como parte del SQL y en la actualidad se hace referencia a cualquier lenguaje formal usado para describir datos o estructuras.

DML Lenguaje de manipulación de datos

(Data Manipulation Language):

Por medio de este lenguaje los usuarios pueden realizar consultas o manipular datos que se encuentran ya organizados en una forma estructurada.

En la actualidad el DML más utilizado es el SQL (Structured Query Language), usado para manipular datos de base de datos relacional. Entre otros se encuentran: IMS/DL1, CODASYL.

Los DML se pueden clasificar en dos grupos:

Lenguajes de consulta procedimentales: El usuario da instrucciones para que se desarrollen procedimientos u operaciones sobre la base de datos.

Lenguajes de consulta no procedimentales: El usuario realiza una descripción de un procedimiento específico.

Ejemplos de comandos DML son:

SELECT: utilizado para recuperación de datos en lenguajes como SQL, indicando la información a recuperar mediante cuatro partes básicas: la cláusula Select seguida de la descripción de lo que se quiere buscar, los nombres de las columnas. La cláusula From seguida de la especificación de la tabla o tablas de las que se realizan la consulta. Ambas cláusulas son obligatorias.

Las otras dos cláusulas son opcionales, la cláusula Where como criterio de selección y la cláusula Orderby como criterio de ordenación.

INSERT: utilizada para la inserción de nuevos datos en una tabla, agregando nuevos registros a la tabla especificada. También es utilizado para la inserción de varias filas a la vez indicando una consulta que genere varios registros con valores a insertar.

DELETE: mediante esta sentencia podemos eliminar filas en una tabla. Se pueden borrar una fila o varias filas. En el caso que se quiera borrar todas las filas la tabla queda vacía pero la definición de la tabla no desaparece.

UPDATE: se utiliza para modificar valores de una o más columnas en las filas seleccionadas de una tabla específica. En la sentencia se debe detallar la tabla destino a modificar.

DCL Lenguaje de control de datos

(Data Control Language)

Permite al administrador controlar el acceso a los datos contenidos en la Base de Datos.

Ejemplo de comandos DCL, tenemos:

REVOKE: elimina permisos concedidos por el comando GRANT.

GRANT: asigna permisos a usuarios y roles para tareas específicas.

COMPONENTES

Entre los componentes que encontramos en un SMDB (Sistema Manejador de Base de Datos) tenemos:

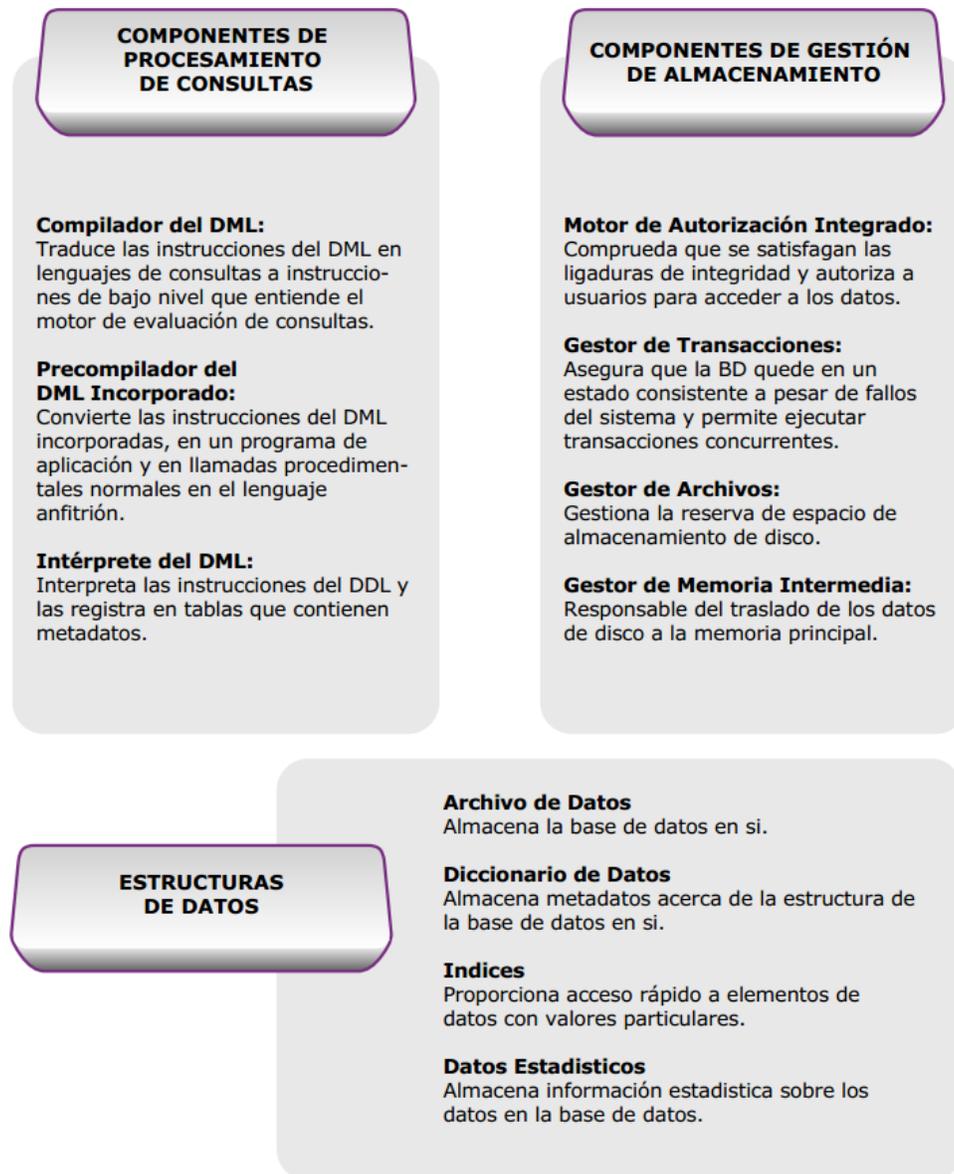


Imagen 1. Lenguajes

Fuente: <http://www.byspel.com/wp-content/uploads/2015/01/Lenguaje-de-control-de-Datos.png>

Los sistemas de gestión de bases de datos permiten administrar la información en forma ordenada, fácil y transparente, estos se realizan por medio de rutinas automáticas.

Funciones de los SDBD:

- Permitir el acceso externo a los bancos de información en forma rápida y segura.
- Responder a las peticiones de los usuarios.
- Llevar un registro de control de accesos.
- Administrar el uso y rendimiento de la base de datos.
- Interacción con el manejador de archivos. Esto a través de las sentencias en DML al comando del sistema de archivos. Así el Manejador de base de datos es el responsable del verdadero almacenamiento de los datos. Respaldo y recuperación.

Características de un SGBD:

- Diccionarios de datos: se establece para garantizar el entendimiento de las definiciones, términos y variables que se manejan en la base de datos.
- Seguridad: debe contar con mecanismo de protección de lectura y escritura, siempre apegados al plan o parametrización definida por el administrador.
- Independencia: la independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- Redundancia mínima: se asegura que la información no este repetida o únicamente lo necesario para el proceso.
- Consistencia: cuando hay información repetida, es necesario garantizar que en

caso de actualizaciones, se actualice en todos los campos en donde existe.

- Integridad: se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.
- Abstracción: los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos.
- Respaldo y recuperación: debe incluir además, programar o rutinas que permitan la realización de copias de seguridad y la restauración de las mismas en caso de algún siniestro.
- Control de concurrencia: generalmente varias personas acceden simultáneamente a la misma base de datos o registro, Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.

Para más detalles, visitar: <http://manejadores-de-bases-de-datos.wikispaces.com/>

Arquitectura de sistemas de bases de datos

La arquitectura de un sistema de bases de datos depende caso completamente del SGBD empleado, en ella, analizamos aspectos como distribución, conectividad en red y paralelismo.

Distribución

Dependiendo de la naturaleza del negocio, la información puede estar en el lugar donde es generada y luego unificada, o en el centro de cómputo central destinado a con-

solidar la información, lo importante es que ésta sea accesible a quien la necesite en el momento requerido.

Conectividad en red

Para que una base de datos sea realmente útil, debe permitir acceso desde puntos re-

motos o alejados de forma que el conocimiento sea útil para los usuarios, estas ventajas son posibles gracias al internet y las redes locales, las bases de datos pueden ser accedidas por empleados, clientes, proveedores o entidades de control.

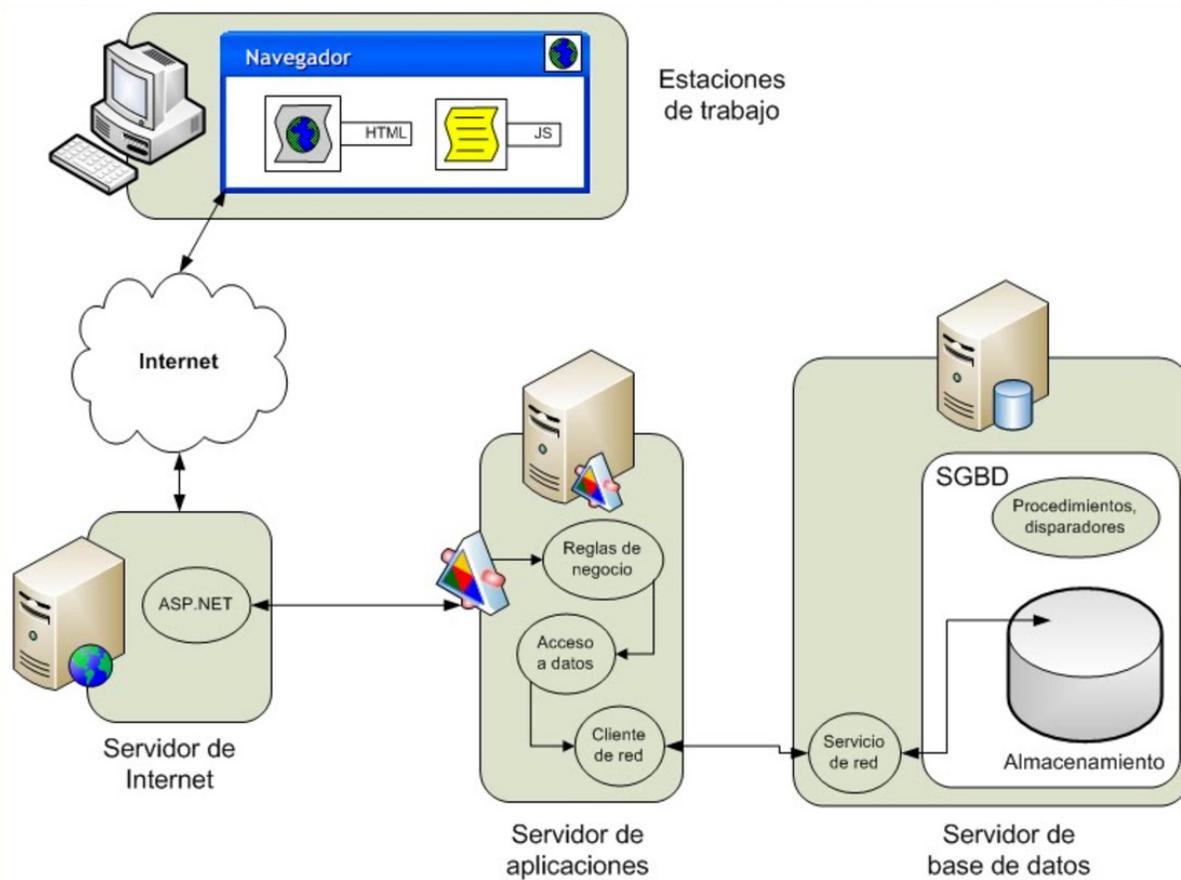


Imagen 2: Conectividad en Red

Fuente: https://s15w007.files.wordpress.com/2013/03/aplicacion_en_n_capas.jpg

Paralelismo

Una base de datos ofrece servicios a cientos o miles de usuarios simultáneamente, por consiguientes, sus capacidades de procesa-

miento deben ser acordes a la demanda y uso exigidos. Se le conoce como la capacidad de ejecutar una cantidad N de transacciones por segundo.

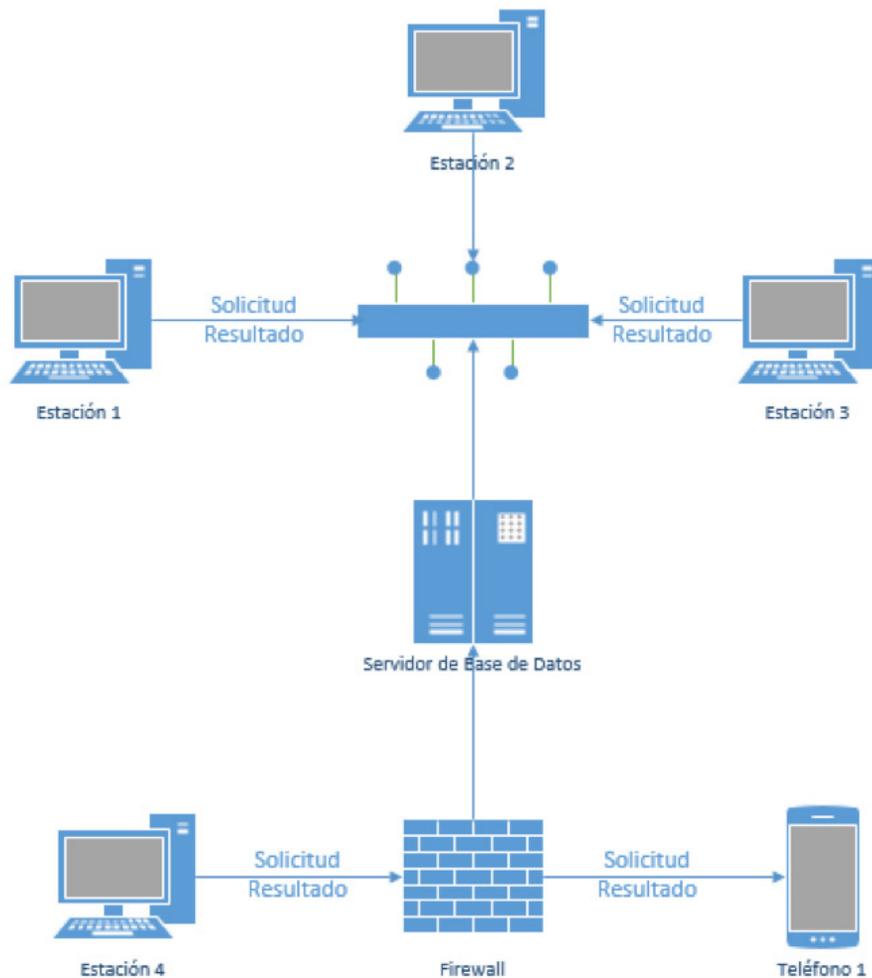


Imagen 3: Paralelismo
Fuente: Propia.

Niveles de abstracción

Basado en la función y visión que tiene el usuario de la base de datos, es posible encontrar 3 niveles de interacción diferentes:

- Nivel físico
El nivel más bajo de abstracción describe como se almacenan realmente los datos. En el nivel físico se describen en detalle las estructuras de datos complejas de bajo nivel.
- Nivel conceptual
Que es el siguiente nivel más alto de abstracción, se describe cuáles son los datos reales que están almacenados en la base de datos y qué relaciones existen entre los datos.
- Nivel lógico
El siguiente nivel más alto de abstracción describe que datos se almacenan en la base de datos y que relaciones existen entre esos datos. La base de datos completa se describe así en términos de un número pequeño de estructuras relativamente simples en el nivel físico, los usuarios del nivel lógico no necesitan preocuparse de esta complejidad. Los administradores de base de datos, que deben decidir la información que se mantiene en la base de datos, usan el nivel lógico de abstracción.

Independencia lógica y física de los datos

El concepto de independencia de datos lo podemos definir como la capacidad para modificar el esquema en un nivel del sistema sin tener que modificar el esquema del

nivel inmediato superior. Se pueden definir dos tipos de independencia de datos:

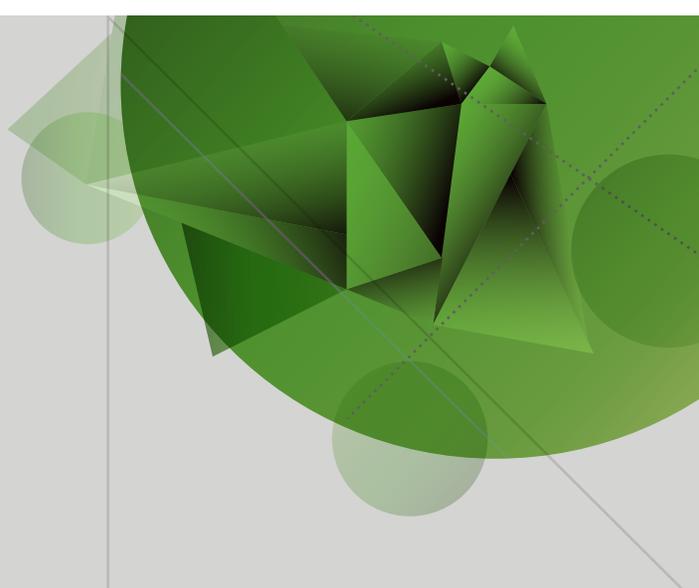
- La independencia lógica
Es la capacidad de modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación. Se puede modificar el esquema conceptual para ampliar la base de datos o para reducirla. Si, por ejemplo, se reduce la base de datos eliminando una entidad, los esquemas externos que no se refieran a ella no deberán verse afectados.
- La independencia física
Es la capacidad de modificar el esquema interno sin tener que alterar el esquema conceptual (o los externos). Por ejemplo, puede ser necesario reorganizar ciertos ficheros físicos con el fin de mejorar el rendimiento de las operaciones de consulta o de actualización de datos. Dado que la independencia física se refiere sólo a la separación entre las aplicaciones y las estructuras físicas de almacenamiento, es más fácil de conseguir que la independencia lógica.



2

Unidad 2

Diseño de bases de
datos



Bases de Datos I

Autor: Camilo Cardona Patiño

Introducción

Aprender los fundamentos generales de bases de datos es de elemental importancia para poder avanzar en el diseño de sistemas integrales y el almacenamiento de la información, el estudiante conocerá las diferentes etapas del diseño de bases de datos y algunas técnicas que facilitan el modelamiento de los datos.

Se expondrán los diferentes tipos de modelos de datos, su función, implementación y el papel que juegan dentro de la teoría de las bases de datos, se verá también el mecanismo adecuado para garantizar la consistencia de la información.

Se invita al estudiante a revisar todo el material proporcionado para trabajar durante la semana, procurando revisar, analizar y estudiar todos los ejemplos planteados, se le invita insistentemente a exponer todas las dudas que pueda presentar de manera que tenga claridad en todos los conceptos trabajados.

Diseño de bases de datos

Debido a su complejidad, diseñar bases de datos no es tarea fácil, existen parámetros como: los requisitos del sistema de información, la interoperabilidad y la complejidad misma de la información que hacen del diseño de bases de datos, una tarea nada sencilla.

Por tal motivo, es recomendable dividir el proceso del desarrollo de bases de datos en etapas, más fáciles de desarrollar, controlar y gerenciar. De cada etapa se obtiene el insumo necesario para la siguiente de manera que el avance es progresivo, ofreciendo cierta tolerancia a cambios y simplificando el proceso global.

El resultado final del diseño de bases de datos, entrega una clara estructura de datos con la que un sistema de información determinado puede cumplir una serie de funciones y características previamente determinadas.

Etapas para el diseño de bases de datos

Etapas 1. Análisis de requerimientos

La forma correcta de entender los requerimientos del cliente es por medio de la comunicación, el cliente y/o los usuarios

expresan sus necesidades y expectativas respecto a los beneficios que esperan obtener del sistema de información.

Se deben documentar e identificar todos los requisitos actuales como la información inicial, los procesos y las salidas.

En esta etapa se obtiene el insumo inicial para dar comienzo al desarrollo de la base de datos. Tales insumos son:

- Identificar el ámbito de la base de datos.
- Establecer los mecanismos para la recolección de la información.
- Definir las vistas de usuario.
- Establecer los requerimientos de procesamiento.
- Construcción del diccionario de datos.

Etapas 2: Diseño conceptual

Esta etapa del diseño es independiente de la infraestructura y la tecnología a emplear, ya que analiza únicamente la estructura de la información.

Se deben detallar ampliamente la estructura y significado de los datos, a esta altura del diseño, se obtiene una representación abstracta de la realidad, se pueden evidenciar las relaciones entre los diferentes componentes de la información así como sus cualidades propias.

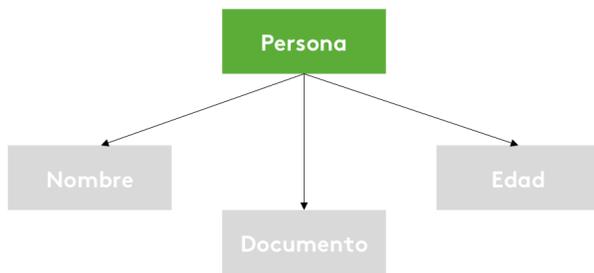


Figura 1. Diseño Conceptual "Persona"
Fuente: Propia.

El resultado de esta etapa e insumo para la siguiente es algún tipo de modelo de datos de alto nivel, generalmente se emplea el modelo entidad relación, que permite visualizar gráficamente las relaciones, dependencias y asociaciones entre datos.

Etapa 3. Diseño lógico

En esta etapa del diseño, se debe definir la tecnología a emplear, igualmente, el modelo de datos obtenido previamente se debe adaptar la tecnología.

El objetivo es convertir el esquema abstracto en un modelo sistemático que pueda ser desarrollado sobre un sistema de gestión de bases de datos determinado. Existen en la actualidad herramientas CASE que se encargan de convertir el modelo conceptual en una serie de comandos tipo LDD, facilitando el proceso de diseño lógico.

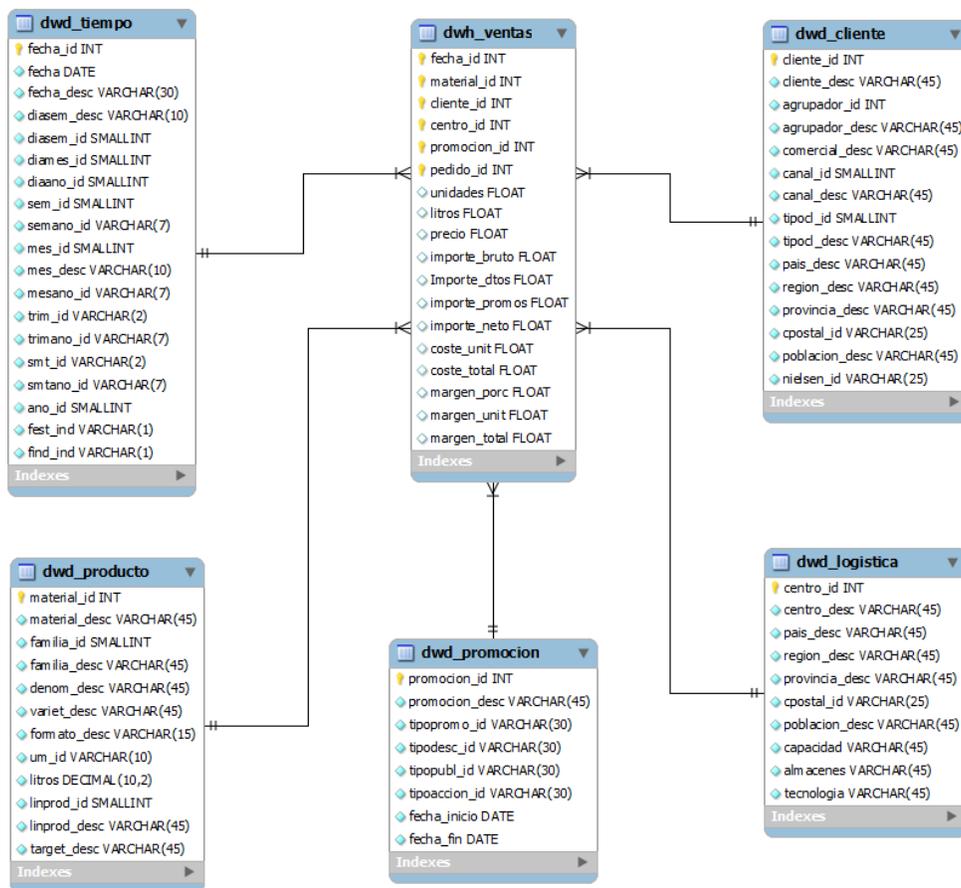


Imagen 1. Diseño lógico

Fuente: <https://churriwifi.files.wordpress.com/2009/12/disenio-fisico-prototipo-er3.png>

Etapa 4. Diseño físico

El diseño físico es el proceso de implementación física del modelo de datos lógico en un sistema de gestión de bases de datos.

Incluye las estructuras de archivos determinadas en las que almacenar las tablas, o relaciones, de la base de datos, y en asegurar que se puede acceder a las relaciones de forma rápida, eficiente y segura.

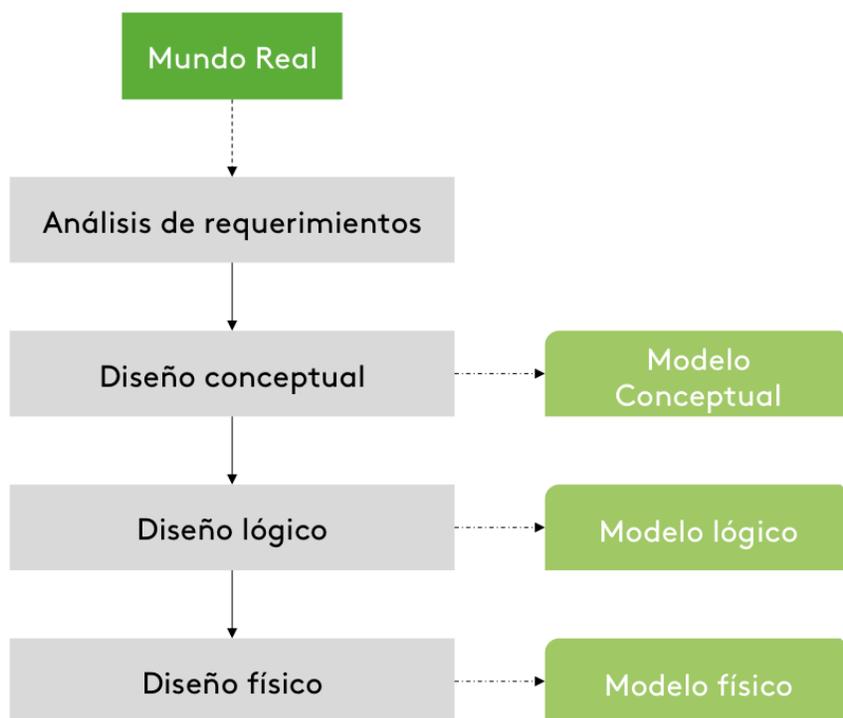


Figura 2. Diseño de bases de datos
Fuente: Propia.

Modelos de datos

El modelo de datos es una herramienta de abstracción que permite documentar y organizar la información en pro de ofrecer una interfaz de comunicación entre los encargados de soporte técnico y los usuarios del sistema. Su enfoque está en la planeación del desarrollo de aplicaciones para decidir cómo almacenar los datos y acceder a la información.

Apoyados en un lenguaje de programación, los modelos de datos proveen la estructura

de la información homogénea que permite la comunicación y el intercambio de información entre sistemas.

Los modelos de datos son de gran importancia en la creación de sistemas de información, de ellos es posible la administración y empleo útil y eficiente de enormes volúmenes de información. Los modelos de datos aportan una estructura para el acceso y el almacenamiento, lo que permite escalar las aplicaciones que hacen uso de la información.

Para garantizar la eficiencia y desempeño adecuados de la base de datos, es necesario realizar inversiones importantes en el desarrollo, operación y mantenimiento de los sistemas de información, en ocasiones se encuentran en el mercado empresas con deficientes sistemas de información completamente aislados de los demás sistemas,

esta solución no ofrece mayores capacidades de crecimiento o escalabilidad a la empresa, por lo que están condenados a desaparecer.

Un modelo de datos debe tener una serie de características, las cuales conforman un conjunto de conceptos matemáticos que ayudan a manipular el objeto, veamos:

Característica	Definición	Ejemplo
Estática	Se refiere a las propiedades propias de un objeto o entidad y que únicamente sirven para definirlo.	Atributos
Dinámica	Se trata de los resultados obtenidos de la combinación u operaciones realizadas sobre las propiedades de entidades u objetos	Consultas
De integridad	Operaciones que garantizan que los cambios se ejecuten en todas las entidades y propiedades involucradas, que guardan alguna relación entre sí.	Transacciones

Tabla 1. Características de aplicaciones
Fuente: Propia.

Tipos de modelos de bases de datos

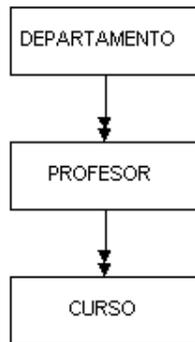
Modelo Jerárquico

En el modelo jerárquico, la información se dispone en una estructura similar a un árbol invertido, de manera que cada elemen-

to puede ramificar en otro, cada hijo puede tener un único padre, sin embargo un padre puede tener múltiples hijos.

La estructura presenta algunas ineficiencias ya que no provee mecanismos para el manejo de redundancia de datos, ni para el manejo de rutas muy extensas o incompletas.

Estructura lógica



Ejemplo de base de datos

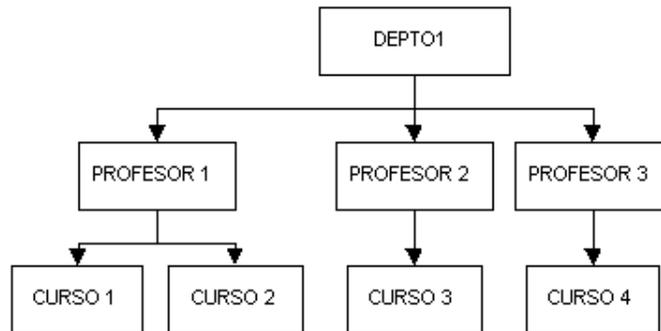


Imagen 2. Modelo jerárquico
Fuente: <http://elies.rediris.es/elies9/fig4-8.gif>

Modelo de Red

Estructura la información utilizando dos construcciones fundamentales, llamadas récords (registros) y sets (conjuntos); los primeros contienen campos y los últimos definen relaciones entre los registros de uno a

muchos (ej.: un cliente, muchos productos comprados). Una de las diferencias con el modelo jerárquico, es que amplía las posibilidades de asociación entre padre e hijos, ahora los hijos pueden tener más de un padre, pueden existir relaciones N:N.

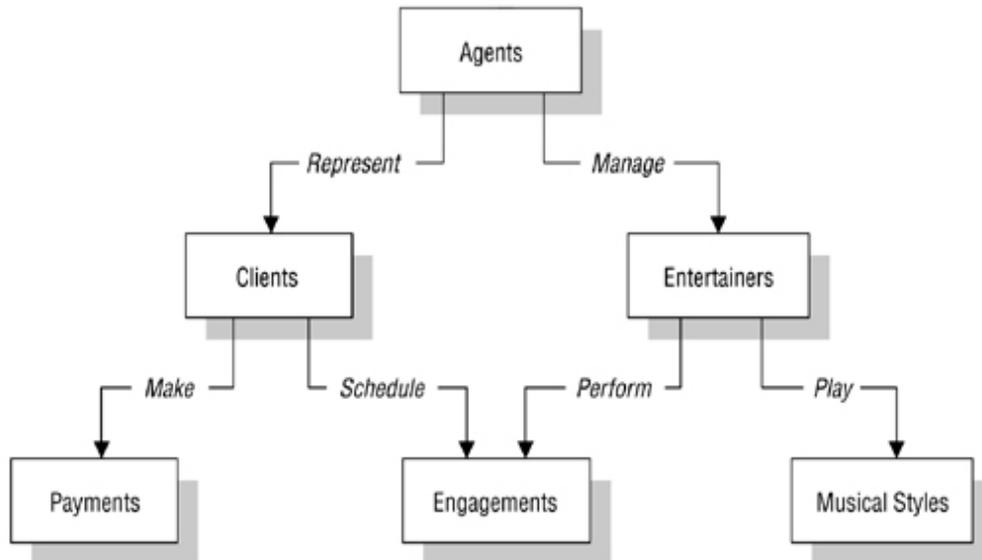


Imagen 3. Modelo en Red
Fuente: <http://www.creativasfera.com/wp-content/uploads/2012/02/datos-modelo-red.jpg>

Modelo Relacional

El modelo relacional está desarrollado con base en las matemáticas y el álgebra rela-

cional, aplica conceptos de lógica y teoría de conjuntos para proyectar relaciones y dependencias entre las tablas y los objetos.

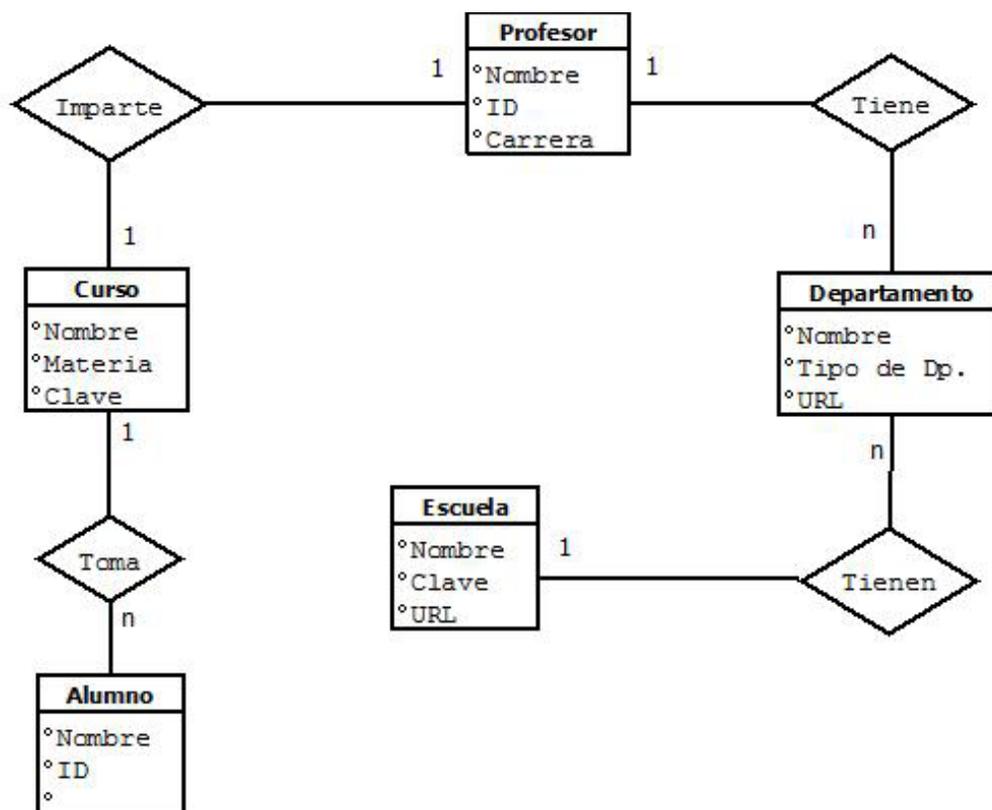


Imagen 4. Modelo relacional

Fuente: <https://arimuesc.files.wordpress.com/2013/02/profesor.jpeg>

Restricciones de integridad

Las restricciones de integridad garantizan la consistencia semántica de los datos, también son un medio que certifica que los cambios realizados por usuarios, no provoquen la pérdida de consistencia de los datos.

Así como en el mundo real tenemos restricciones físicas, por ejemplo nada puede viajar más rápido que la velocidad de la luz, en

el mundo virtual, también es necesario poner limitaciones a ciertos comportamientos no deseados.

Elementos de una restricción de integridad:

- Operación de actualización, que puede ser inserción, actualización o borrado, a partir de ella, se ejecuta una secuencia para comprobar el cumplimiento o no de la restricción.

- **Cumplimiento**, la condición debe cumplirse o no, generalmente es una operación de tipo booleano que arroja como resultado un verdadero o un falso.
- **Acción**, es la función que debe ser ejecutada tomando como base el resultado de la comparación anterior, dependiendo de tal resultado se ejecutaran unas u otras instrucciones.

Referencial: las claves ajenas de una tabla hija se tienen que corresponder con la clave primaria de la tabla padre con la que se relaciona. Por ejemplo, en la tabla familiares de los empleados necesitaremos el documento de identidad del empleado, que es la clave ajena de la tabla.

Tipos de integridad

TIPOS DE INTEGRIDAD	TIPOS DE RESTRICCIÓN
DOMINIO	DEFAULT CHECK REFERENTIAL
ENTIDAD	PRIMARY KEY UNIQUE
REFERENCIAL	FOREYGH KEY CHECK

Imagen 5. Integridad

Fuente: <https://deaheesy22.files.wordpress.com/2010/03/imagen19.png>

De dominio: se emplea para restringir el rango de valores que pueden ser asignados a una variable determinada, por ejemplo: NumeroCelular: Únicamente puede tener 10 dígitos numéricos, no puede tener ni más, ni menos, ni tampoco valores alfabéticos.

De entidad: debe existir un campo en la tabla denominado clave primaria, el cual debe tener un valor único que jamás se repita, tampoco puede ser nulo, por ejemplo: el documento de identidad.

2

Unidad 2

Normalización



Bases de Datos I

Autor: Camilo Cardona Patiño

Introducción

En la presente cartilla se tratará el tema de normalización de bases de datos, su desarrollo se basa en la implementación de una serie de reglas y técnicas que pretenden disminuir los posibles errores asociados a la duplicidad, las inconsistencias y las redundancias de los datos.

El proceso de normalización puede no ser simple en algunas ocasiones, sin embargo es un tema de gran importancia para el desarrollo de bases de datos ya que provee ventajas significativas en relación a la administración y rendimiento del sistema.

Se invita al estudiante a leer completamente la guía de manera que entienda los contenidos y conceptos desarrollados, así mismo, se le recomienda insistentemente analizar los ejemplos planteados, siguiendo paso a paso el desarrollo a fin de lograr claridad y comprensión en el procedimiento efectuado, estar atento de la realización de las actividades, participación en los foros y formular todas las dudas que pueda tener en relación al tema tratado.

Normalización

Es un proceso mediante el cual se organizan y estructuran los valores de una base de datos, para tal efecto, es necesaria la creación de tablas y de relaciones entre tales tablas, basado en reglas de diseño que buscan asegurar los datos, permitir flexibilidad al eliminar la redundancia y eliminar dependencias incoherentes.

Cuando existe redundancia en la información, se incurre en un mayor uso de espacio en disco –desperdicio en la capacidad de almacenamiento–, igualmente se complica la labor de administración y mantenimiento de la misma base de datos, ya que exige igualmente mayor capacidad de cómputo como memoria RAM y procesador, también implica que al cambiar un valor en alguna tabla, los datos redundantes deben ser reemplazados igualmente en todas las tablas en donde existan, de lo contrario se presentarían inconsistencias, por ejemplo, resulta mucho más fácil realizar un cambio en un cliente si tenemos una tabla exclusiva para el manejo de información relacionada con clientes y otra tabla relacionada con sus pedidos que si en lugar de ello tuviéramos en una misma tabla, los datos cliente y estos, fueran duplicados para cada pedido.

Dependencia incoherente

Una dependencia incoherente ocurre cuando se mezclan datos que no guardan ningún tipo de relación entre sí, esto complica la búsqueda de información y disminuye la capacidad de procesamiento pues se deben arrastrar además de los datos necesarios, otra serie de datos que no aportan ningún valor adicional, por ejemplo, si existe una tabla que almacena la información básica de un cliente, esta debe ser independiente del inventario.

Se han definido una serie de reglas que al ser aplicadas, ofrecen un estado de normalización en la base de datos, cada uno de los estados se conoce como forma normal, las reglas se van aplicando en forma progresiva y a medida que se implementan, se avanza en la depuración de la estructura de la base de datos.

Objetivo de la normalización

Cuando se recurre a la normalización de bases de datos, se pretende evitar incurrir en errores que pueden salir costosos en términos de capacidad de procesamiento, almacenamiento y consistencia de datos.

- Disminuir al máximo la redundancia en los datos.
- Prevenir problemas y complicaciones con la actualización de información en las tablas.

- Asegurar la integridad de los datos.

Normalmente, cuando se trabaja con el modelo relacional, una tabla, es también denominada una relación, sin embargo, para recibir esta calificación, la tabla debe cumplir una serie de condiciones:

- Cada columna debe tener un nombre único que la diferencie de las demás.
- No debe permitir dos o más filas idénticas, es decir no pueden haber duplicados.
- Las columnas manejan un tipo de datos que debe ser respetado.
- Algunos términos y equivalencias para la normalización.

Término	Equivalencia
Atributo	Columna o campo
Base de datos	Banco de Datos
Clave	Llave o código de identificación
Clave Ajena	También denominada "Foránea", es la llave proveniente de otra tabla
Clave Alternativa	Clave secundaria
Clave Candidata	Superclave mínima
Clave Primaria	Clave candidata seleccionada
Dependencia Multivaluada	Dependencia multivalor
Relación	Tabla o archivo
Registro	Fila, renglón o tupla
SGBDR	Sistema Gestor de Bases de Datos Relacionales
Tupla	Registro, fila o renglón
1FN	Primera Forma Normal
2FN	Segunda Forma Normal
3FN	Tercera Forma Normal
BCNF	Forma normal de Boyce-Codd
4FN	Cuarta Forma Normal
5FN	Quinta Forma Normal

Tabla 1. Equivalencias
Fuente: Propia.

Representación gráfica de los componentes más importantes en una tabla

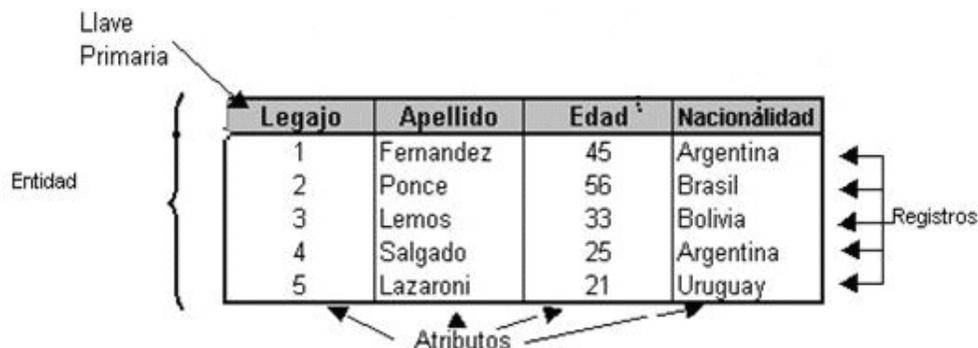


Imagen 1. Tabla

Fuente: http://www.cyta.com.ar/elearn/bd/images/m2_p1_f1.gif

Claves o llaves

Las llaves son instrumentos para la identificación de tuplas a nivel de tablas dentro de una base de datos, en la teoría y en la práctica se encuentran diferentes tipos de llaves, sin embargo de una u otra forma todas cumplen la misma función:

- Clave primaria
- Claves candidatas
- Clave foránea
- Clave alternativa
- Clave compuesta

Clave primaria

Generalmente por cuestiones de practicidad y presentación, la clave primaria se representa en la primera columna de la tabla, esta identifica unívocamente a cada una de las filas o tuplas contenidas en la tabla, en ocasiones, la llave primaria se configura para ser una magnitud numérica y autoincrementada, de manera que podemos estar seguros que nunca se va a repetir.

Claves candidatas

Las claves candidatas aparecen, cuando en una tabla, tenemos más de una columna que cumple con la condición para ser llave primaria, en este caso, podemos emplear como llave primaria una u otra, dependiendo de la conveniencia.

Clave foránea

Es la clave primaria de otra tabla, adicionalmente existe alguna tipo de relación entre ambas tablas.

Clave alternativa

Se trata de una llave candidata que no ha sido empleada como llave principal, sin embargo, debido a sus características, podría ser empleada como llave principal en cualquier momento.

Clave compuesta

Es una llave que está compuesta por dos o más columnas dentro de la tabla.

Dependencias

Ocurren cuando el valor en una tabla está subordinado al comportamiento de otro valor en otra tabla.

Dependencia funcional

En bases de datos, una dependencia funcional se manifiesta como una interconexión entre dos o más atributos, consideremos los siguientes planteamientos:

Una dependencia funcional puede ser vista como una relación de muchos a uno partiendo de un conjunto de elementos y su asociación con otros conjuntos (otra tabla).

Matemáticamente se explica así:

Sea R una relación.

Sean M y N subconjuntos de atributos de R .

Entonces si dos tuplas coinciden en el valor de M , también ambas también coincidirán en el valor de N .

Propiedades

Sean A , B , C y D subconjuntos arbitrarios del conjunto que tiene la relación R , escribir AB significa la unión de A y B , entonces tenemos:

Reglas de inferencia de Armstrong

1. Reflexividad: Si B es un subconjunto de A , entonces $A \rightarrow B$
2. Aumentación: Si $A \rightarrow B$, entonces $AC \rightarrow BC$
3. Transitividad: Si $A \rightarrow B$ y $B \rightarrow C$, entonces $A \rightarrow C$
4. Autodeterminación: $A \rightarrow A$

5. Descomposición: Si $A \rightarrow BC$, entonces $A \rightarrow B$ y $A \rightarrow C$

6. Unión: Si $A \rightarrow B$ y $A \rightarrow C$, entonces $A \rightarrow BC$

7. Composición: Si $A \rightarrow B$ y $C \rightarrow D$, entonces $AC \rightarrow BD$

Teorema de unificación general, u : Unión y $-$: Diferencia

Si $A \rightarrow B$ y $A \rightarrow C$, entonces $A \cup (C - B) \rightarrow BD$

Grados de normalización

Como se mencionó anteriormente, existen unas reglas de normalización para las bases de datos, si se cumple con cada regla, podemos decir que la base de datos tiene "forma normal #", ejemplo tercera forma normal, si se cumple con la tercera normalización. Aunque son posibles otros niveles de normalización, la tercera forma normal se considera el máximo nivel necesario para la mayor parte de las aplicaciones.

FORMAS NORMALES

Las formas normales son aplicadas a las tablas de una base de datos. Decir que una base de datos está en la forma normal N es decir que todas sus tablas están en la forma normal N



Imagen 2. Formas normales
Fuente: <http://image.slidesharecdn.com/segundaformanormal-130903212933-95/segundaforma-normal-4-638.jpg?cb=1378243851>

Primera forma normal (1FN)

Para que una tabla pueda ser considerada en su primera forma normal, debe cumplir con:

- La llave principal es única.
- La llave principal no contiene valores nulos.
- Atomicidad en todos sus atributos, si los elementos del dominio son simples e indivisibles.
- El número de columnas es invariable.
- Independencia a de orden para filas y columnas.

Segunda forma normal (2FN)

Para que una tabla pueda ser considerada en su segunda forma normal, ya debe estar en 1FN y además debe cumplir con:

- Todas las dependencias parciales se deben eliminar. "las dependencias parciales hacen referencia a datos que no dependen de la llave primaria para ser identificados en la tabla".

Tercera forma normal (3FN)

Para que una tabla pueda ser considerada en su tercera forma normal, ya debe estar en 2FN y además debe cumplir con:

- No hay dependencias transitivas.
- Todas las columnas son funcionalmente dependientes de la llave primaria.

Forma normal de Boyce-Codd (FNBC)

Una tabla está en FNBC sí y sólo si las únicas dependencias funcionales elementales son aquellas en las que la clave primaria determinan un atributo.

Cuarta forma normal (4FN)

Para que una tabla pueda ser considerada en su cuarta forma normal, ya debe estar en FNBC y además debe cumplir con:

- No existen dependencias multivaluadas.
- Se generan todas las relaciones externas con otras tablas.

Quinta forma normal (5FN)

Para que una tabla pueda ser considerada en su quinta forma normal, ya debe estar en 4FN y además debe cumplir con:

- Todas las relaciones de dependencia join o de reunión, se encuentran definidas por llaves candidatas.

Reglas de CODD

En su obra, Codd notó que existían bases de datos identificadas como racionales sin serlo realmente, por este motivo, publicó una serie de 12 reglas que al ser cumplidas, garantizan que efectivamente el sistema es racional, aunque no es necesario cumplir las doce, debido a la gran complejidad de algunas, un sistema se considera más racional en la medida que más reglas de Codd cumpla.

Regla 1: Regla de la Información

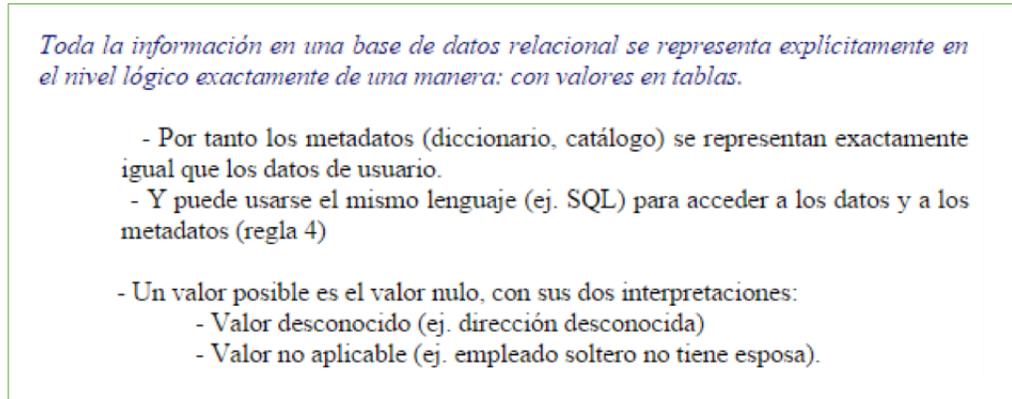


Imagen 3. Regla 1

Fuente: <http://www.galeon.com/nevifi/Archivos/Codd.pdf>

Regla 2: Regla del Acceso Garantizado

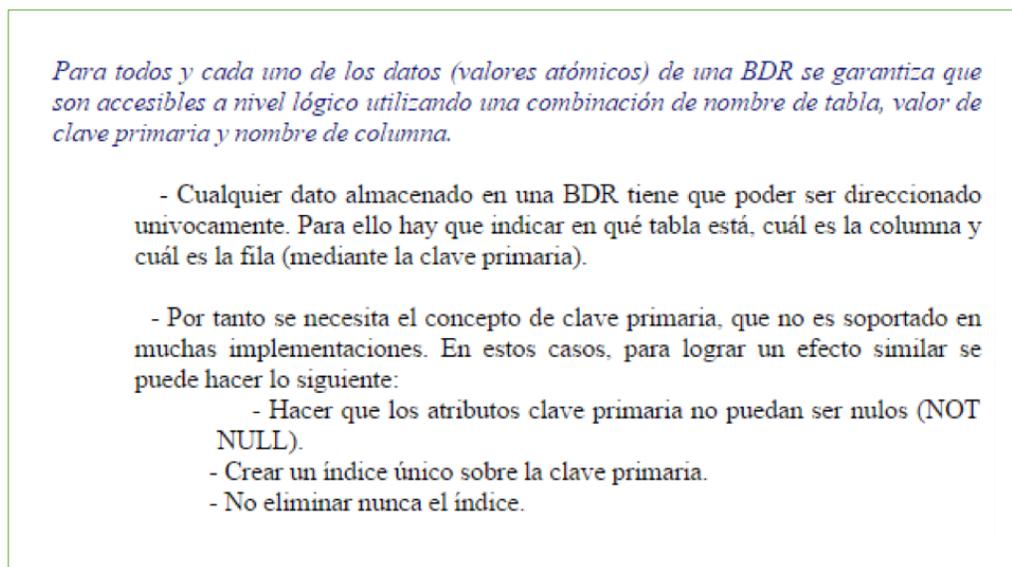


Imagen 4. Regla 2

Fuente: <http://www.galeon.com/nevifi/Archivos/Codd.pdf>

Regla 3: Tratamiento Sistemático de Valores Nulos

Los valores nulos (que son distintos de la cadena vacía, blancos, 0, ...) se soportan en los SGBD totalmente relacionales para representar información desconocida o no aplicable de manera sistemática, independientemente del tipo de datos.

- Se reconoce la necesidad de la existencia de valores nulos, para un tratamiento sistemático de los mismos.
- Hay problemas para soportar los valores nulos en las operaciones relacionales, especialmente en las operaciones lógicas.

Lógica trivaluada. Es una posible solución. Existen tres (no dos) valores de verdad: Verdadero, Falso y Desconocido (null). Se crean tablas de verdad para las operaciones lógicas:

- null Y null = null
- Verdadero Y null = null
- Falso Y null = Falso
- Verdadero O null = Verdadero
- etc.

Un inconveniente es que de cara al usuario el manejo de los lenguajes relacionales se complica pues es más difícil de entender.

Imagen 5. Regla 3

Fuente: <http://www.galeon.com/nevifi/Archivos/Codd.pdf>

Regla 4: Catálogo Dinámico en Línea Basado en el Modelo Relacional

La descripción de la base de datos se representa a nivel lógico de la misma manera que los datos normales, de modo que los usuarios autorizados pueden aplicar el mismo lenguaje relacional a su consulta, igual que lo aplican a los datos normales.

- Es una consecuencia de la regla 1 que se destaca por su importancia. Los metadatos se almacenan usando el modelo relacional, con todas las consecuencias.

Imagen 6. Regla 4

Fuente: <http://www.galeon.com/nevifi/Archivos/Codd.pdf>

Regla 5: Regla del Sublenguaje de Datos Completo

Un sistema relacional debe soportar varios lenguajes y varios modos de uso de terminal (ej: rellenar formularios, etc.). Sin embargo, debe existir al menos un lenguaje cuyas sentencias sean expresables, mediante una sintaxis bien definida, como cadenas de caracteres y que sea completo, soportando:

- *Definición de datos*
- *Definición de vistas*
- *Manipulación de datos (interactiva y por programa)*
- *Limitantes de integridad*
 - *Limitantes de transacción (iniciar, realizar, deshacer) (Begin, commit, rollback).*

- *Además de poder tener interfaces más amigables para hacer consultas, etc. siempre debe de haber una manera de hacerlo todo de manera textual, que es tanto como decir que pueda ser incorporada en un programa tradicional.*
- *Un lenguaje que cumple esto en gran medida es SQL.*

Imagen 7. Regla 5

Fuente: <http://www.galeon.com/nevifi/Archivos/Codd.pdf>

Regla 6: Regla de Actualización de Vistas

Todas las vistas que son teóricamente actualizables se pueden actualizar por el sistema.

- *El problema es determinar cuáles son las vistas teóricamente actualizables, ya que no está muy claro.*
- *Cada sistema puede hacer unas suposiciones particulares sobre las vistas que son actualizables.*

Imagen 8. Regla 6

Fuente: <http://www.galeon.com/nevifi/Archivos/Codd.pdf>

Regla 7: Inserción, Actualización y Borrado de Alto Nivel

La capacidad de manejar una relación base o derivada como un solo operando se aplica no sólo a la recuperación de los datos (consultas), si no también a la inserción, actualización y borrado de datos.

- Esto es, el lenguaje de manejo de datos también debe ser de alto nivel (de conjuntos). Algunas bases de datos inicialmente sólo podían modificar las tuplas de la base de datos de una en una (un registro de cada vez).

Imagen 9. Regla 7

Fuente: <http://www.galeon.com/nevifi/Archivos/Codd.pdf>

Regla 8: Independencia Física de Datos

Los programas de aplicación y actividades del terminal permanecen inalterados a nivel lógico cuandoquiera que se realicen cambios en las representaciones de almacenamiento o métodos de acceso.

- El modelo relacional es un modelo lógico de datos, y oculta las características de su representación física.

Imagen 10. Regla 8

Fuente: <http://www.galeon.com/nevifi/Archivos/Codd.pdf>

Regla 9: Independencia Lógica de Datos

Los programas de aplicación y actividades del terminal permanecen inalterados a nivel lógico cuandoquiera que se realicen cambios a las tablas base que preserven la información.

- Cuando se modifica el esquema lógico preservando información (no valdría p.ej. eliminar un atributo) no es necesario modificar nada en niveles superiores.
- Ejemplos de cambios que preservan la información:
 - Añadir un atributo a una tabla base.
 - Sustituir dos tablas base por la unión de las mismas. Usando vistas de la unión puedo recrear las tablas anteriores...

Imagen 11. Regla 9

Fuente: <http://www.galeon.com/nevifi/Archivos/Codd.pdf>

Regla 10: Independencia de Integridad

Los limitantes de integridad específicos para una determinada base de datos relacional deben poder ser definidos en el sublenguaje de datos relacional, y almacenables en el catálogo, no en los programas de aplicación.

- El objetivo de las bases de datos no es sólo almacenar los datos, si no también sus relaciones y evitar que estas (limitantes) se codifiquen en los programas. Por tanto en una BDR se deben poder definir limitantes de integridad.
 - Cada vez se van ampliando más los tipos de limitantes de integridad que se pueden utilizar en los SGBDR, aunque hasta hace poco eran muy escasos.
- Como parte de los limitantes inherentes al modelo relacional (forman parte de su definición) están:
 - Una BDR tiene **integridad de entidad**. Es decir, toda tabla debe tener una clave primaria.
 - Una BDR tiene **integridad referencial**. Es decir, toda clave externa no nula debe existir en la relación donde es primaria.

Imagen 12. Regla 10

Fuente: <http://www.galeon.com/nevifi/Archivos/Codd.pdf>

Regla 11: Independencia de Distribución

Una BDR tiene independencia de distribución.

- Las mismas órdenes y programas se ejecutan igual en una BD centralizada que en una distribuida.
- Las BDR son fácilmente distribuibles:
 - Se parten las tablas en fragmentos que se distribuyen.
 - Cuando se necesitan las tablas completas se recombinan usando operaciones relacionales con los fragmentos.
 - Sin embargo se complica más la gestión interna de la integridad, etc.
- Esta regla es responsable de tres tipos de **transparencia de distribución**:
 - **Transparencia de localización**. El usuario tiene la impresión de que trabaja con una BD local. (aspecto de la regla de independencia física)
 - **Transparencia de fragmentación**. El usuario no se da cuenta de que la relación con que trabaja está fragmentada. (aspecto de la regla de independencia lógica de datos).
 - **Transparencia de replicación**. El usuario no se da cuenta de que pueden existir copias (réplicas) de una misma relación en diferentes lugares.

Imagen 13. Regla 11

Fuente: <http://www.galeon.com/nevifi/Archivos/Codd.pdf>

Regla 12: Regla de la no Subversión

Si un sistema relacional tiene un lenguaje de bajo nivel (un registro de cada vez), ese bajo nivel no puede ser usado para saltarse (subvertir) las reglas de integridad y los limitantes expresados en los lenguajes relacionales de más alto nivel (una relación (conjunto de registros) de cada vez).

- Algunos problemas no se pueden solucionar directamente con el lenguaje de alto nivel.
- Normalmente se usa SQL inmerso en un lenguaje anfitrión para solucionar estos problemas. Se utiliza el concepto de cursor para tratar individualmente las tuplas de una relación. En cualquier caso no debe ser posible saltarse los limitantes de integridad impuestos al tratar las tuplas a ese nivel.

Imagen 14. Regla 12

Fuente: <http://www.galeon.com/nevifi/Archivos/Codd.pdf>

3

Unidad 3

Álgebra relacional



Bases de Datos I

Autor: Camilo Cardona Patiño

Introducción

El Álgebra relacional implica cálculos y operaciones lógicas aplicadas a la información en una base de datos de manera que es posible obtener un enorme provecho de su explotación, con las diferentes operaciones, es posible realizar búsquedas, consultas, filtros y cruces de información en forma rápida, eficiente y confiable.

El análisis y estudio relacional sienta las bases fundamentales para el entendimiento, comprensión y dominio del lenguaje SQL, que a su vez permite la construcción y tratamiento de consultas efectuadas sobre el sistema de gestión de bases de datos.

Se invita al estudiante a leer completamente la guía de manera que logre obtener una sólida y consistente fundamentación teórica, de igual forma, se le recomienda analizar los ejemplos planteados, siguiendo paso a paso el desarrollo a fin de lograr claridad y comprensión en el procedimiento efectuado, estar atento de la realización de las actividades, participación en los foros y formular todas las dudas que pueda tener en relación al tema.

Álgebra relacional

Debemos entender el álgebra relacional como un lenguaje de consulta basado en procedimientos o “procedimental”, incluye una serie de operaciones que se encargan de transformar una o más entradas, denominadas relación, para generar una nueva a partir de los insumos.

El álgebra relacional es empleada en la construcción y generación de consultas sobre una base de datos, mediante las operaciones definidas por el álgebra relacional es posible generar nuevas relaciones basados en la información almacenada, los pasos para generar dicha consulta son descritos por medio de un procedimiento, es por tal razón que reciben el nombre de consultas procedimentales, son un tipo de Lenguaje de Manipulación de Datos.

El álgebra relacional tiene las siguientes operaciones:

- Unión
- Intersección
- Diferencia
- Producto cartesiano
- Selección
- Proyección
- Combinación

Para entender mejor la función y el comportamiento de tales operaciones, veamos a continuación algunos ejemplos:

Nombre	Documento	Dirección	Genero
Alberto Orozco	123456	Calle 1 N 6-54	M
Andrea Caro	987654	Calle 2 N 8-56	F

Tabla 1. R
Fuente: Propia.

Nombre	Documento	Dirección	Genero
Martha Plazas	741852	Calle 18 N 26-04	F
Alberto Orozco	123456	Calle 1 N 6-54	M

Tabla 2. S
Fuente: Propia.

Unión

$R \cup S$, la unión de R y S es el conjunto de elementos que están en R o S o ambos. Un elemento solo aparece una sola vez.

Hallar el resultado de la unión de las tablas 1 y 2:

Nombre	Documento	Dirección	Genero
Alberto Orozco	123456	Calle 1 N 6-54	M
Andrea Caro	987654	Calle 2 N 8-56	F
Martha Plazas	741852	Calle 18 N 26-04	F

En este caso vemos que el registro perteneciente a "Alberto Orozco" no se duplica, únicamente se agrega una vez.

Intersección

$R \cap S$, el conjunto de elementos que aparecen en ambos R y S

Hallar el resultado de la intersección de las tablas 1 y 2:

Nombre	Documento	Dirección	Genero
Alberto Orozco	123456	Calle 1 N 6-54	M

Al calcular la intersección de las tablas 1 y 2, podemos notar que el único punto de cruce entre ambas es el registro perteneciente a "Alberto Orozco".

Diferencia

$R - S$, la diferencia de R y S, el conjunto de elementos que están en R pero no en S. Es importante resaltar que $R - S$ es diferente a $S - R$.

Hallar el resultado de la diferencia de las tablas 1 y 2:

Nombre	Documento	Dirección	Genero
Andrea Caro	987654	Calle 2 N 8-56	F

Hallar el resultado de la diferencia de las tablas 2 y 1:

Nombre	Documento	Dirección	Genero
Martha Plazas	741852	Calle 18 N 26-04	F

Producto cartesiano

El tratamiento de esta operación es igual al definido por la teoría de conjuntos como producto cruz, esta operación da como resultado una nueva tabla con todos los pares ordenados posibles entre ambas relaciones o tablas. Esta operación define una concatenación de todas las tuplas de ambas tablas, se expresa así:

$R \times S$

La forma correcta de expresar el producto cartesiano es $R \times S$, dando como origen un resultado que incluye todas las combinaciones posibles entre las tuplas de R y las tuplas de S.

Cabe mencionar que por notación que: $R \times S \neq S \times R$

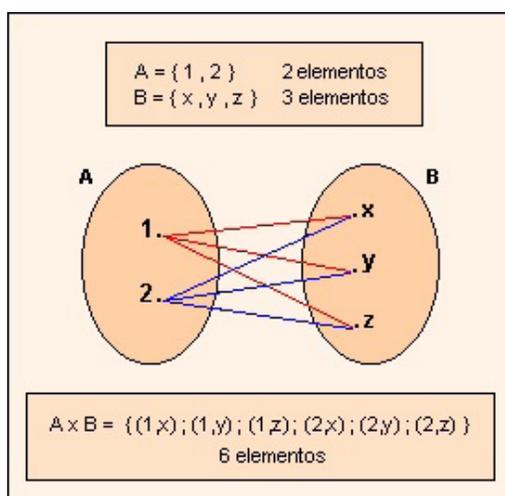


Imagen 1. Producto Cartesiano

Fuente: https://eserpade01b.wikispaces.com/file/view/prod_2.gif/270041828/238x179/prod_2.gif

Hallar el resultado del producto cartesiano de las siguientes tablas:

ID_Alumno	NombreAlumno	CodigoAlumno
001	Juan Barrera	123
002	María Cárdenas	456
003	Oscar Peña	789

Tabla 3. Alumnos
Fuente: Propia.

ID_Materia	NombreMateria
478	Bases de Datos
459	Programación

Tabla 4. Materias
Fuente: Propia.

Alumnos×Materias

ID_Alumno	NombreAlumno	CodigoAlumno	ID_Materia	NombreMateria
001	Juan Barrera	123	478	Bases de Datos
001	Juan Barrera	123	459	Programación
002	María Cárdenas	456	478	Bases de Datos
002	María Cárdenas	456	459	Programación
003	Oscar Peña	789	478	Bases de Datos
003	Oscar Peña	789	459	Programación

Ahora vamos a calcular el producto cartesiano de Materias × Alumnos

ID_Materia	NombreMateria	ID_Alumno	NombreAlumno	CodigoAlumno
478	Bases de Datos	001	Juan Barrera	123
478	Bases de Datos	002	María Cárdenas	456
478	Bases de Datos	003	Oscar Peña	789
459	Programación	001	Juan Barrera	123
459	Programación	002	María Cárdenas	456
459	Programación	003	Oscar Peña	789

Es evidente que el resultado de ambas operaciones es diferente.

Selección

Según el curso de Bases de datos de la Universidad Oberta de Cataluña “podemos ver la selección como una operación que sirve para elegir algunas tuplas de una relación y eliminar el resto. Más concretamente, la selección es una operación que, a partir de una relación, obtiene una nueva relación formada por todas las tuplas de relación de partida que cumplen una condición de selección especificada.

La selección es una operación unaria. Siendo C una condición de selección, la selección de T con la condición de C se indica como $T(C)$ ”.

Los atributos con valor “Null” por defecto no cumplen con ninguna condición.

Para construir operaciones de selección debemos emplear operadores lógicos, los cuales pueden ser:

= : símbolo de igual qué.

≠ ó !=: diferente a.

≥: mayor que o igual a.

≤: menor que o igual a.

>: mayor que.

<: menor que.

Adicionalmente, para incrementar el universo de posibilidades y combinaciones, podemos incluir los operadores:

AND: (\wedge) Es verdadero si y solo si, dos condiciones C1 y C2, se cumplen.

OR: (\vee) Es verdadero si, de dos condición C1 y C2, alguna o ambas se cumplen.

NOT: (\neg) Es verdadero si una condición C1 no se cumple, o es falsa si C1 se cumple.

Ejemplo:

Basado en la siguiente tabla de empleados, determinar:

- a. El número de empleados con más de 8 meses de trabajo en la entidad.
- b. El número de trabajadores que además de tener más de 8 meses de trabajo en la entidad, son menores de 30 años.

ID_Empleado	NombreEmpleado	EdadEmpleado	MesesTrabajo
12	Juan	25	10
23	Carla	33	7
34	Alvaro	29	12
45	Karina	40	15
56	Marina	32	8
67	Ernesto	28	5

Tabla 5. Empleados
Fuente: Propia.

El número de empleados con más de 8 meses de trabajo en la entidad.

La operación de selección sería:

σ (MesesTrabajo >8)

Resultado:

ID_Empleado	NombreEmpleado	EdadEmpleado	MesesTrabajo
12	Juan	25	10
34	Alvaro	29	12
45	Karina	40	15

El número de trabajadores que además de tener más de 8 meses de trabajo en la entidad, son menores de 30 años.

σ (MesesTrabajo >8 \wedge EdadEmpleado <30)

Resultado:

ID_Empleado	NombreEmpleado	EdadEmpleado	MesesTrabajo
12	Juan	25	10
34	Alvaro	29	12

Proyección

Según el curso de Bases de datos de la Univesidad Oberta de Cataluña “Podemos considerar la proyección como una operación que sirve para elegir algunos atributos de una relación y eliminar el resto, más concretamente, la proyección de una operación, que a partir de una relación, obtiene una nueva relación formada por todas las (sub)tuplas de la relación de partida que resultan de eliminar unos atributos especificados.

La proyección es una operación unaria. Siendo $\{A_i, A_j, \dots, A_k\}$ se indica como $T\{A_i, A_j, \dots, A_k\}$ ”

Ejemplo:

De la tabla 5. Empleados, obtener la proyección:

$T(\text{NombreEmpleado}, \text{EdadEmpleado})\text{Empleados}$

Resultado:

NombreEmpleado	EdadEmpleado
Juan	25
Carla	33
Alvaro	29
Karina	40
Marina	32
Ernesto	28

Combinación o Join

Según el curso de Bases de datos de la Univesidad Oberta de Cataluña “la combinación es una operación que, a partir de dos relaciones, obtiene una nueva relación formada por todas las tuplas que resultan de concatenar tuplas de la primera relación con tuplas de la segunda, y que cumplen una condición de combinación específica.

La combinación es una operación binaria. Siendo T y S dos relaciones cuyos esquemas no tienen ningún nombre de atributo común, y siendo B una condición de combinación, la combinación de T y S según la condición B se indica $T[B]S$.”

Ejemplo:

Hallar el resultado de la combinación de la tabla Profesional y Programas:

Profesional [Programa] Programas

ID_Profesional	NombreProfesional	Programa
123	Carlos	98
456	Alejandro	78
789	Luisa	45
147	Diana	32
258	Jorge	98

Tabla 6. Profesional
Fuente: Propia.

ID_Programa	NombreProyecto
98	Familias
65	Jovenes
32	Ancianos

Tabla 7. Programas
Fuente: Propia.

Obtenemos:

ID_Profesional	NombreProfesional	Programa	Programa	Programa
123	Carlos	98	98	Familias
147	Diana	32	32	Ancianos
258	Jorge	98	98	Familias

El resultado anterior muestra las tuplas de la tabla profesionales que tiene un programa asignado, tal programa está registrado en la tabla programas.

3

Unidad 3

Cálculo relacional



Bases de Datos I

Autor: Camilo Cardona Patiño

Introducción

Continuando con la línea de análisis y tratamiento matemático de bases de datos, estudiaremos a continuación los conceptos generales del cálculo relacional, sus componentes, y tipología, veremos también ejemplos clásicos para la formulación de consultas y operaciones en un sistema de gestión de la información.

Se invita al estudiante a leer completamente la guía de manera que logre obtener una sólida y consistente fundamentación teórica, de igual forma, se le recomienda analizar los ejemplos planteados, siguiendo paso a paso el desarrollo a fin de lograr claridad y comprensión en el procedimiento efectuado, estar atento de la realización de las actividades, participación en los foros y formular todas las dudas que pueda tener en relación al tema.

Cálculo relacional

La orientación del cálculo relacional es completamente diferente al del álgebra relacional, sin embargo, los dos cumplen la misma función, lo que implica que empleando uno u otro para construir consultas sobre bases de datos, se obtendrán los mismos resultados, todas las respuestas obtenidas mediante el empleo del cálculo relacional se definen por:

- Resultados
- Sentencia de cualificación
- Cuantificadores

Los resultados son los valores que coinciden con las condiciones buscadas y las sentencias de cualificación se refiere a las condiciones que se esperan sean cumplidas por los resultados.

Los cuantificadores son un concepto relativamente nuevo, este componente realiza una estimación previa del número aproximado de registros que serán afectados por alguna operación solicitada, su novedad radica en que la estimación se hace antes de ejecutar la tarea, estos cuantificadores se clasifican en:

Universales

Los cuantificadores Universales son usados para ofrecer la funcionalidad equivalente a la división del álgebra relacional, se emplean para revelar cuando a un grupo de tuplas similares se les aplica una condición particular.

Existenciales

Son aquellos que tratan de averiguar el número de registros que devolvería un tipo de consulta. Por ejemplo: saber el número de clientes de Bogotá que han comprado el producto 2016. Si el número de registros que satisfacen esta relación es mayor que cero, podemos generar la consulta para lanzar posteriormente un informe, en caso contrario se puede enviar un mensaje al usuario para que sepa que no hay ningún cliente con estas características.

A modo de comparación, el álgebra relacional realiza paso por paso cada una de las operaciones para obtener los resultados, generando en el proceso una serie de tablas y valores intermedios que sirven para obtener el resultado final, mientras que el cálculo relacional realiza toda la operación en un solo paso, es decir, se trata de un lenguaje declarativo.

Tipos de cálculo relacional

De acuerdo al tipo de estructura que se está procesando, el cálculo relacional puede ser tipificado como:

- Cálculo relacional orientado a tuplas.
- Cálculo relacional orientado a dominios.

Cálculo relacional orientado a tuplas

Para este tipo de operaciones, se incluyen las tuplas de una o más tablas, para este procesamiento, se utilizan los nombres de las tablas y las etiquetas como variables de tupla.

Cálculo relacional orientado a dominios

Se trata igual que el orientado a tuplas, pero en lugar de incluir tuplas, se incluyen dominios que alcanzan a más de 2 relaciones, recordemos que el dominio es el conjunto de todos los posibles datos que puede tomar una variable.

Lógica de primer orden

La base fundamental del cálculo relacional se denomina lógica de primer orden, esta lógica es un sistema diseñado para el manejo de predicados o afirmaciones realizadas sobre uno o más objetos con el objetivo de expresar relaciones entre los mismos. Por ejemplo, dentro de la lógica de primer orden podemos expresar el siguiente enunciado:

**Alberto es hijo de Mario
Por lo tanto, podemos concluir que
Mario es padre de Alberto**

Símbolos empleados

Símbolos de puntuación: paréntesis “(, ”)” y coma “,”.

Símbolos de variables: se representan mediante letras minúsculas del final del alfabeto: r, s, t, u, v, w, x, y, z.

Símbolos de constantes: se representan mediante letras minúsculas del principio del alfabeto: a, b, c, d, e.

Símbolos de funciones: también se representan con letras minúsculas, pero del centro del alfabeto: f, g, h, i...

Símbolos de predicados: se representan mediante letras mayúsculas.

Operadores lógicos: \neg (negación), \rightarrow (implicación), \wedge (conjunción) y \vee (disyunción).

Cuantificadores: \forall (cuantificador universal) y \exists (cuantificador existencial).

La jerarquía en la precedencia de los operadores y cuantificadores es (de mayor a menor):

1. \neg, \forall, \exists
2. \vee
3. \wedge
4. \rightarrow

Dicha precedencia se puede alterar utilizando los paréntesis.

Los operadores \rightarrow y \wedge y el cuantificador \forall se introducen en el lenguaje por conveniencia, ya que en realidad no son necesarios pues:

$$F \rightarrow G \equiv (\neg F) \vee G$$

$$\forall x (F) \equiv \neg \exists x (\neg F)$$

$$F \wedge G \equiv \neg (\neg F \vee \neg G)$$

Operaciones en cálculo relacional

Declaración de la consulta:

$\{t \mid P(t)\}$

Se lee:

El conjunto de todas las tuplas t , tal que el predicado P , es verdadero para t .

Ejemplos

Dadas las relaciones r y s :

- la **unión** se expresa

$$\{ t \mid r(t) \vee s(t) \}$$

– es decir, el conjunto de tuplas t tales que t está en r ó en s

Imagen 1. Unión

Fuente: http://www.dsi.fceia.unr.edu.ar/downloads/base_de_datos/CalculoRelacional.pdf

Veamos ahora un ejemplo de cálculo relacional para una operación de diferencia:

Dadas las relaciones r y s :

- la **diferencia** se expresa

$$r - s = \{ t \mid r(t) \wedge \neg s(t) \}$$

– es decir, el conj. de tuplas t tales que t está en r y **no** en s

Imagen 2. Diferencia

Fuente: http://www.dsi.fceia.unr.edu.ar/downloads/base_de_datos/CalculoRelacional.pdf

Veamos ahora un ejemplo de cálculo relacional para una operación de proyección:

Dadas las relaciones r y s :

- la **proyección** $\pi_{i_1, \dots, i_k}(r)$ se expresa

$$\{ t(k) \mid \exists u (r(u) \wedge t[1]=u[i_1] \wedge \dots \wedge t[k]=u[i_k]) \}$$

– donde $t(k)$ significa tuplas de grado k

Imagen 3. Proyección

Fuente: http://www.dsi.fceia.unr.edu.ar/downloads/base_de_datos/CalculoRelacional.pdf

Consultas con cálculo relacional:

Dada la relación "Estudiantes":

ID_Estudiante	NombreEstudiante	Edad_Estudiante	Ciudad_Estudiante
123	Carlos	25	Bogotá
456	Andrea	20	Bogotá
789	Carlos	24	Medellín
147	Jose	21	Cali

Tabla 1. Estudiantes
Fuente: Propia.

Seleccionar tuplas de estudiantes llamados Carlos:

Estudiante: NombreEstudiante = "Carlos"

ID_Estudiante	NombreEstudiante	Edad_Estudiante	Ciudad_Estudiante
123	Carlos	25	Bogotá
789	Carlos	24	Medellín

Ejemplo:

Seleccionar estudiantes que viven en Bogotá y tienen más de 23 años:

Estudiantes: Ciudad_Estudiante='Bogotá' AND Edad_Estudiante>23

ID_Estudiante	NombreEstudiante	Edad_Estudiante	Ciudad_Estudiante
123	Carlos	25	Bogotá

Seleccionar ID_Estudiante y NombreEstudiante de los estudiantes de Bogotá:

Estudiantes.ID_Estudiante, Estudiantes.NombreEstudiante: Ciudad_Estudiante = 'Bogotá'

ID_Estudiante	NombreEstudiante
123	Carlos
456	Andrea

4

Unidad 4

Lenguaje SQL



Bases de Datos I

Autor: Camilo Cardona Patiño

Introducción

Para el tratamiento de bases de datos, es de gran importancia tener amplio dominio y conocimiento sobre los lenguajes de definición y de manipulación de datos, ya que es por medio de tales lenguajes que nos es posible crear, cambiar y eliminar la información contenida dentro del sistema de gestión de bases de datos.

Este tipo de lenguajes son fáciles de interpretar e implementar, sin embargo es de mucha importancia saber exactamente la operación que se está realizando a fin de evitar pérdidas o inconsistencias en los datos.

Se invita al estudiante a leer completamente la guía de manera que logre alcanzar gran solidez en cuanto al tema de lenguajes de definición y manipulación de datos, de igual forma, se le recomienda analizar los contenidos planteados, siguiendo paso a paso el desarrollo a fin de lograr claridad y comprensión en el procedimiento efectuado, estar atento de la realización de las actividades, participación en los foros y formular todas las dudas que pueda tener en relación al tema tratado.

Lenguaje SQL

El SQL o lenguaje de consulta estructurado, es un lenguaje diseñado para la administración de bases de datos, este lenguaje es empleado por una gran variedad de motores de bases de datos debido a que se trata de un lenguaje normalizado, el cual ofrece un amplio y completo margen de operaciones posibles sobre los datos y su estructura.

Esta herramienta es empleada para gestionar, ordenar, actualiza y recuperar los datos almacenado en la base de datos.

SQL se conforma por operadores, clausulas, comandos y funciones, los que ejecutados en diferentes combinaciones, permiten la creación, actualización y eliminación de registros o atributos en una base de datos. Dentro de tales comandos, encontramos:

- DDL: Lenguaje de definición de datos
- DML: Lenguaje de manipulación de datos

DDL: Lenguaje de definición de datos

Provee las capacidades necesarias para definir las estructuras de almacenamiento, procedimientos y funciones sobre la base de datos, lo que significa que es posible crear la base de datos propiamente dicha, también la generación de tablas y vistas, igualmente provee la funcionalidad de modificar o eliminar cualquier objeto, instancia, relación o incluso la misma base de datos.

En general, por medio de este lenguaje podemos construir, mantener, modificar o eliminar la base de datos.

Comando	Función
Create	Permite la creación de campos, índices y tablas.
Drop	Permite la eliminación de tablas e índices.
Alter	Permite modificar la definición de los datos, o la adición de campos a una tabla.

Tabla 1. Comandos DDL
Fuente: Propia.

Comando Create

Este comando permite la creación de objetos dentro de la base de datos, dentro de algunos de los objetos que pueden ser creados tenemos: Vistas, Índices, Tablas, disparadores (Triggers), procedimientos y funciones.

Este tipo de sentencias son particularmente útiles para la gestión interna de un sistema de bases de datos relacionales, veamos algunos ejemplos:

Objeto a crear	Sintaxis del comando
Base de datos	<pre>CREATE DATABASE [IF NOT EXISTS] <i>nombre_BD</i>;</pre> <p>Ejemplo: CREATE DATABASE IF NOT EXISTS alumnos;</p>
Tabla	<pre>CREATE [TEMPORARY] TABLE [IF NOT EXISTS] <i>nombre_tabla</i>;</pre> <p>Ejemplo: CREATE TABLE IF NOT EXISTS nom_alumno;</p>
Índice	<pre>CREATE [UNIQUE FULLTEXT SPATIAL] INDEX <i>nombre_index</i> [USING <i>tipo_index</i>] ON <i>nombre_tabla</i> (<i>nom_col_index</i>,...);</pre> <p>Ejemplo: CREATE INDEX parte_nombre ON nom_alumno (nombre(10));</p>

Imagen 1. Create

Fuente: <http://image.slidesharecdn.com/comandosddl-y-dml-130407123153-phpapp02/95/comandos-ddl-y-dml-3-638.jpg?cb=1365337953>

Comando Drop

Este comando permite la eliminación de objetos dentro de la base de datos, dentro de algunos de los objetos que pueden ser eliminados tenemos: Vistas, Índices, Tablas, disparadores (Triggers), procedimientos y funciones.

Este tipo de sentencias son particularmente útiles para la gestión interna de un sistema de bases de datos relacionales, veamos algunos ejemplos:

Objeto a eliminar	Sintaxis del comando
Base de datos	<pre>DROP {DATABASE SCHEMA} [IF EXISTS] <i>db_name</i>;</pre> <p>Ejemplo: DROP DATABASE IF EXISTS alumnos;</p>
Índice	<pre>DROP INDEX <i>index_name</i> ON <i>tbl_name</i>;</pre> <p>Ejemplo: DROP INDEX nombre_2 ON alumnos;</p>
Tabla	<pre>DROP [TEMPORARY] TABLE [IF EXISTS] <i>tbl_name</i> [, <i>tbl_name</i>];</pre> <p>Ejemplo: DROP TABLE IF EXISTS usuarios;</p>

Imagen 2. Drop

Fuente: <http://image.slidesharecdn.com/comandosddl-y-dml-130407123153-phpapp02/95/comandos-ddl-y-dml-4-638.jpg?cb=1365337953>

Comando Alter

Con este tipo de sentencias, es posible cambiar la estructura de almacenamiento de la información, se pueden agregar o eliminar campos en una tabla, se pueden adicionar y quitar los índices de una tabla, etc. Veamos algunos ejemplos:

Objeto a modificar	Sintaxis del comando
Base de datos	<pre>ALTER {DATABASE SCHEMA} [db_name] alter_specification [, alter_specification];</pre> <p>Ejemplo: ALTER DATABASE DISABLE INDEXES;</p>
Tabla	<pre>ALTER [IGNORE] TABLE tbl_name alter_specification [, alter_specification];</pre> <p>Ejemplo: ALTER TABLE Persona CHANGE nombre nombres varchar(30);</p>

Imagen 3. Alter

Fuente: <http://image.slidesharecdn.com/comandosddl-y-dml-130407123153-phpapp02/95/comandos-ddl-y-dml-3-638.jpg?cb=1365337953>

DML: Lenguaje de manipulación de datos

Provee las capacidades necesarias para la administración de los datos almacenados dentro de las tablas, es decir cambiar, actualizar, recuperar, agregar y eliminar la información contenida por la base de datos.

Los comandos tipo DML son:

Comando	Función
Select	Este comando permite la recuperación de un grupo de datos definidos, los cuales pueden ser mostrados o empleados en una operación posterior.
Update	El comando se utiliza para realizar cambios o actualizaciones sobre los registros de la base de datos.
Insert	Este comando es empleado para agregar nuevas registros en las tablas.
Delete	Se emplea para eliminar registros en las tablas.

Tabla 2. Comandos DML

Fuente: Propia.

Comando select

El comando select se emplea para recuperar tuplas de una base de datos, esta selección se puede realizar sobre todas o algunas las columnas, veamos algunos ejemplos de su aplicación:

Objeto a Seleccionar	Sintaxis del comando
Base de datos	
Tabla	<pre>SELECT [ALL DISTINCT UNIQUE] FROM nombre-de-tabla ORDER BY campo1 [ASC DESC], campo2 [ASC DESC], ...; Ejemplo: SELECT * FROM empleados ORDER BY oficio ASC, apellido DESC;</pre>
Índice	<pre>SELECT [ALL DISTINCT UNIQUE] lista-de-selecciones FROM nombre- de-tabla WHERE condición ORDER BY campo1 [ASC DESC], campo2 [ASC DESC], ...; Ejemplo: SELECT * FROM empleados WERE salario>1500 AND dept_no=20;</pre>

Imagen 4. Select

Fuente: <http://image.slidesharecdn.com/comandosddl-y-dml-130407123153-phpapp02/95/comandos-ddl-y-dml-5-638.jpg?cb=1365337953>

Comando Update

La principal función del comando Update es la de cambiar los atributos en los diferentes registros de la tabla, se puede emplear en combinación con el comando Select, veamos algunos ejemplos de su aplicación:

Objeto a Seleccionar	Sintaxis del comando
Base de datos	
Tabla	<pre>UPDATE nombre_tabla SET lista_asignaciones [WHERE Expresión_condicional]; UPDATE Tienda SET Ventas = 500 WHERE Nombre_tenda = "Los Angeles" AND Date = "08-Jan-1999";-</pre>
Índice	

Imagen 5. Update

Fuente: <http://image.slidesharecdn.com/comandosddl-y-dml-130407123153-phpapp02/95/comandos-ddl-y-dml-5-638.jpg?cb=1365337953>

Comando Insert

Es el tipo de sentencia empleada para la agregación de nuevos registros en la tabla, este comando genera nuevas tuplas, veamos algunos ejemplos de su aplicación:

Objeto a Seleccionar	Sintaxis del comando
Base de datos	-
Tabla	<pre>INSERT INTO "nombre_tabla" VALUES ("columna1", "columna2", ...); INSERT INTO Tienda (Nombre_tienda, venta, Fecha) VALUES ('Los Angeles', 900, '10-Jan-2013');</pre>
Índice	

Imagen 6. Insert

Fuente: <http://image.slidesharecdn.com/comandosddl-y-dml-6-638.jpg?cb=1365337953>

Comando Delete

Por último, pero igualmente importante el comando Delete sirve para eliminar registros de la tabla, veamos algunos ejemplos de su aplicación:

Objeto a Seleccionar	Sintaxis del comando
Base de datos	-
Tabla	<pre>DELETE FROM "nombre_tabla" WHERE [condición]; DELETE FROM Tienda WHERE Productos = "Cigarros"</pre>
Índice	

Imagen 7. Delete

Fuente: <http://image.slidesharecdn.com/comandosddl-y-dml-6-638.jpg?cb=1365337953>

4

Unidad 4

Vistas



Bases de Datos I

Autor: Camilo Cardona Patiño

Introducción

Las vistas ofrecen una serie interesante de nuevas posibilidades para la administración de los sistemas de información de bases de datos, con ellas, es posible aumentar el grado de control sobre la información, aumentar el grado de restricción a ciertos usuarios y mejorar el rendimiento del motor de bases de datos.

Se invita al estudiante a leer completamente la guía de manera que logre obtener una sólida y consistente fundamentación teórica, de igual forma, se le recomienda analizar los ejemplos planteados, siguiendo paso a paso el desarrollo a fin de lograr claridad y comprensión en el procedimiento efectuado, estar atento de la realización de las actividades, participación en los foros y formular todas las dudas que pueda tener en relación al tema.

Vistas

Las vistas son funciones esquemáticas que ofrecen una alternativa a la visualización de las tablas, cuando realizamos una consulta, podemos tener partes de las tablas, incluso partes de varias tablas, con las vistas podemos generar una salida visual que contiene la combinación de los resultados de la consulta, así:

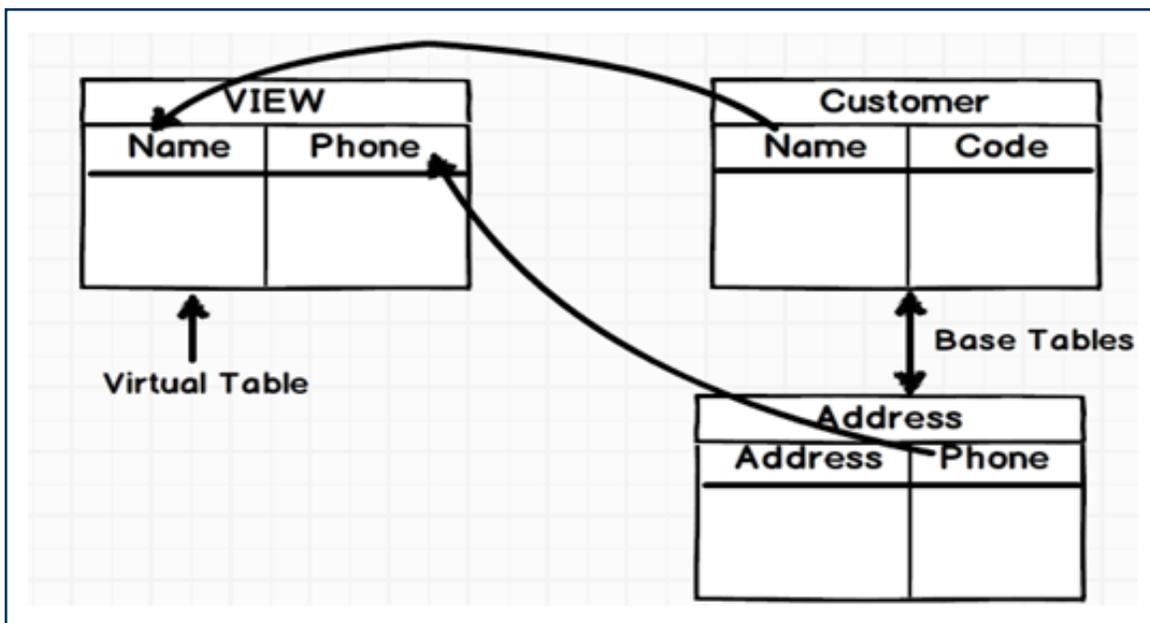


Figura 1. Vistas

Fuente: <http://www.dotnetinterviewquestions.in/contentpics/view....png>

En resumen, la vista guarda una consulta, la cual podemos emplear en operaciones posteriores, gracias a la propiedad de renombramiento, es posible asignarle un nombre a una consulta para ser almacenada temporalmente como una vista.

Funciones de las vistas:

Gracias a su versatilidad, con las vistas podemos:

- Ocultar información.
- Mejorar el rendimiento.
- Simplificar la administración de permisos de usuario.

Función	Características
Ocultar información	Con las vistas es posible permitir la visibilidad de algunos datos mientras que otros se mantienen ocultos y no se incluyen en la vista, la interacción entre el usuario y la vista es equivalente a la del usuario con la tabla, a diferencia que en la vista, es posible controlar la visibilidad de algunos atributos.
Mejorar el rendimiento	Ya que es posible almacenar la vista, resulta particularmente útil y conveniente para realizar actividades repetitivas que ayudan a alivianar la carga sobre el motor de base de datos con vistas previamente generadas en lugar de consultas repetitivas y complejas.
Simplificar la administración de permisos de usuario	Es posible la administración de permisos de usuario únicamente por medio de vistas, es decir a través de consultas previamente diseñadas y controladas, evitando fugas de información, errores y daños a la base de datos, ya que no es necesario concederle permisos directos a ciertos campos u operaciones.

Tabla 1
Fuente: propia.

Creación de vistas

Estructura general para la creación de vistas:

CREATE VIEW "NOMBRE_VISTA" AS "Instrucción SQL";

Estructura general para la eliminación de vistas:

DROP VIEW "NOMBRE_VISTA";

Ejemplos:

Podemos utilizar también una vista para aplicar uniones a dos tablas. En este caso, los usuarios sólo ven una vista en vez de dos tablas, y la instrucción SQL que los usuarios necesitan emitir se vuelve mucho más simple. Digamos que tenemos las siguientes dos tablas:

Tabla *Store_Information*

Store_Name	Sales	Txn_Date
Los Angeles	1500	05-Jan-1999
San Diego	250	07-Jan-1999
Los Angeles	300	08-Jan-1999
Boston	700	08-Jan-1999

Tabla *Geography*

Region_Name	Store_Name
East	Boston
East	New York
West	Los Angeles
West	San Diego

y deseamos construir una vista que tenga ventas organizadas según la región. Colocaríamos la siguiente instrucción SQL:

```
CREATE VIEW V_REGION_SALES
AS SELECT A1.Region_Name REGION, SUM(A2.Sales) SALES
FROM Geography A1, Store_Information A2
WHERE A1.Store_Name = A2.Store_Name
GROUP BY A1.Region_Name;
```

Esto nos brinda una vista, *V_REGION_SALES*, que se ha definido para las ventas de los negocios según los registros de la región. Si deseamos saber el contenido de esta vista, ingresamos,

```
SELECT * FROM V_REGION_SALES;
```

Resultado:

```
REGION SALES
East      700
West     2050
```

Imagen 2. Vistas

Fuente: <http://www.1keydata.com/es/sql/sql-create-view.php>

CRUD Operaciones con registros

Son una serie de funciones de almacenamiento persistente empleadas para el manejo de registros en una base de datos, su nombre proviene de las funciones básicas que relaciona:

- Create
- Read
- Update
- Delete

Create: son una serie de comandos que permiten la creación de tablas, campos, registros e índices en una base de datos.

Ejemplo:

```
CREATE DATABASE Universidad;
```

```
CREATE TABLE Alumno;
```

En los ejemplos anteriores vemos los comandos básicos para la creación de una base de datos y de una tabla respectivamente.

Read: el comando empleado para la lectura en una base de datos es el comando SELECT, el cual permite la recuperación de un grupo de valores almacenados.

Ejemplo:

```
SELECT * FROM Alumno;
```

El comando anterior recupera toda la información contenida en la tabla de nombre Alumno

Update: es el comando empleado para la actualización de un valor en una tabla, este comando siempre debe ir acompañado del comando SET y WHERE, ya que de lo contrario se corre el riesgo de actualizar toda la tabla.

Ejemplo:

```
UPDATE Alumno
```

```
SET NombreAlumno = Pedro
```

```
WHERE ID = 85
```

El comando sirve para reemplazar el nombre del alumno con "PEDRO" para el ID que corresponde a 85.

Delete: es el comando utilizado para eliminar campos o registros.

Ejemplo:

DELETE FROM Alumno

WHERE ApellidoAlumno: Gómez

El anterior comando elimina de la tabla alumno todos los alumnos "Registros" que tienen apellido Gómez.

Son una serie de "pasos rápidos" que podemos implementar para el tratamiento de los registros.

Un buen CRUD debe tener:

- Poder crear fácilmente pantallas de listado de entidades, con filtrado, ordenación, paginado y búsqueda; que dé la opción de edición directa sobre el listado y que permita mostrar/editar información de entidades relacionadas
- Poder crear fácilmente pantallas de edición de entidades, que tengan widgets adecuados para los tipos de datos que necesitamos, y que permita editar directamente entidades relacionadas, dando un interfaz adecuado a las claves foráneas y relaciones de todo tipo de cardinalidad y complejidad
- Poder navegar ágilmente entre entidades relacionadas
- Un sistema de permisos completo que permita restringir con granularidad por grupos de usuario hasta los niveles de columna y fila (i.e. qué registros puede editar cada usuario, y de cada uno, qué atributos)
- Un sistema de workflows que permita manejar diversos estados para cada entidad, aplicar permisos sobre las transiciones y disparar eventos sobre los cambios de estado
- Un sistema de auditoría de cambios que permita ver los históricos de modificaciones y si es factible, restaurar versiones anteriores
- y sobre todo, que todo lo anterior sea extensible, ya que siempre, siempre, necesitaremos más de lo que está previsto

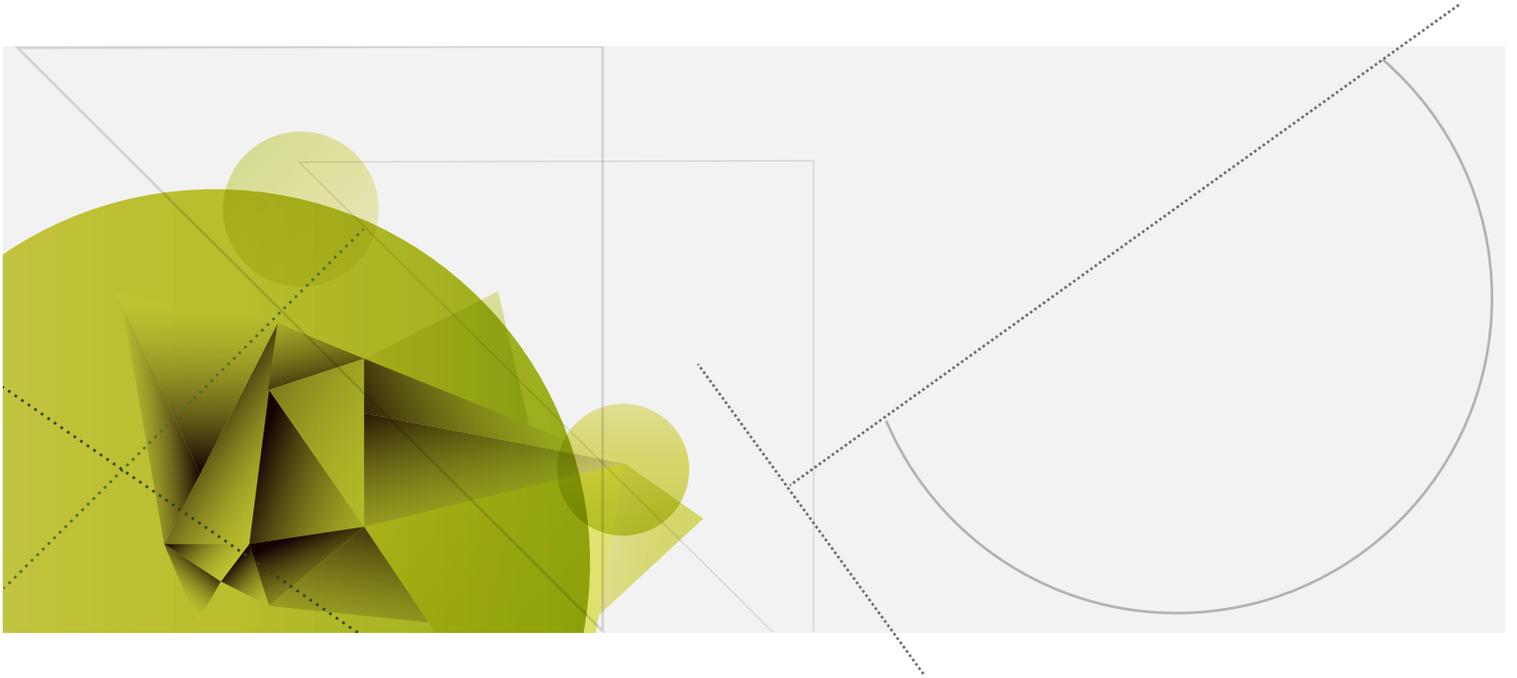
Imagen 3. CRUD

Fuente: <http://www.elprofesionaldelainformacion.com/contenidos/2007/noviembre/12.pdf>

Bibliografía

- **Date, C.** (2001). *Introducción a los Sistemas de Bases de Datos. Séptima Edición.* Prentice Hall.
- **Date, C. & Darwen, H.** (1997). *Guide to the SQL Standard.* Fourth Edition. Addison-Wesley.
- **De Miguel, A. & Piattini, M.** (1993). *Concepción y Diseño de Bases de Datos: del Modelo E/R al Modelo Relacional.*
- **Fernández, C.** (s.f.). *El Modelo relacional de datos: de los fundamentos a los modelos deductivos.* Editorial Díaz de Santos.
- **Laudon, K. & Laudon, J.** (2004). *Sistemas de Información Gerencial.* Octava Edición. México: Editorial Pearson – Prentice Hall.
- **Silberschatz, A., Kort, H. & Sudarshan, S.** (s.f.). *Fundamentos de Bases de Datos.* Editorial McGraw-Hill. 5ª Edición.
- **Ullman, J.** (s.f.). *Principles of Database Systems.* Computer Science Press.

Esta obra se terminó de editar en el mes de octubre
Tipografía Myriad Pro 12 puntos
Bogotá D.C.,-Colombia.



AREANDINA
Fundación Universitaria del Área Andina

MIEMBRO DE LA RED
ILUMNO