



**UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO**

**CENTRO UNIVERSITARIO UAEM ZUMPANGO**

**INGENIERÍA EN COMPUTACIÓN**

**“APUNTES DE FUNDAMENTO DE BASE DE DATOS”**

**APUNTES**

**POR**

**MTE-MI ROSA ERENDIRA REYES LUNA**

**20 DE OCTUBRE DE 2017**

## ÍNDICE

ÍNDICE.....	2
Presentación .....	4
Introducción .....	4
Propósito .....	5
Competencias Genéricas.....	5
Requisitos.....	5
Unidad de Competencia I.....	6
Identificar y establecer los elementos de un Sistema de Base de Datos.....	6
Fundamentos de Bases de Datos distribuidas .....	7
Sistemas de información y base de datos.....	8
Factores de Entorno de un Sistema de información.....	9
Introducción a la Base de Datos.....	10
Independencia de datos respecto a los procedimientos.....	11
Características para un SBD. ....	12
Sistema de archivos.....	13
Unidad de Competencia II .....	15
Generar modelos lógicos/conceptuales de base de datos.....	15
Modelos lógicos/conceptuales de las bases de datos .....	16
Arquitectura de las Bases de Datos.....	17
Modelos de las Bases de Datos .....	17
Modelado de datos .....	18
Aplicar las bases matemáticas para el acceso a una base de datos. ....	20
Unidad de Competencia III.....	20
Fundamentos del Algebra relacional .....	21
Un Panorama General del Algebra Relacional. ....	22
Una Sintaxis para el Álgebra Relacional. ....	23

Lenguaje SQL.....	27
Unidad de Competencia IV.....	32
Utilizar las reglas de normalización en el diseño de base de datos. ....	32
Normalización de Bases de Datos.....	33
Normalización de Base de Datos. ....	33
Grados de normalización. ....	34
Unidad de Competencia V.....	40
Analizar la concurrencia de las transacciones en un sistema de administración de base de datos .....	40
Transacciones en un Sistema de Bases de Datos .....	41
Transacciones.....	42
Candados.....	44
Tipos De Candados:.....	44
Ventajas y Desventajas .....	46
Ejemplo de una conexión de bases de datos en netbeans: .....	46
Conclusión .....	52
Recomendaciones .....	52
Índice de Tablas e Ilustraciones.....	54
Referencias.....	55

## **Presentación**

El documento presenta una compilación de notas retomadas de fuentes arbitradas con la finalidad de integrar a los estudiantes del Programa Educativo de Ingeniería en Computación al tratamiento de información, ya que ésta es el activo de mayor impacto en las empresas, instituciones educativas y oficinas de gobierno. Para alcanzar el objetivo trazado se requiere exponer los conceptos fundamentales de Bases de Datos, para dominar terminología y así dar pauta al modelado de datos.

Representar información requiere de habilidades cognitivas para identificar entidades y atributos que permitan definir objetos útiles para la solución de problemas a través de sistemas informáticos. Una vez que se logra abstraer los elementos que intervienen en una problemática se analizan los cambios que puede sufrir un dato a través de la manipulación con operaciones algebraicas relacionales.

La destreza en actualizar la información y consultarla, requiere de software que agilice la búsqueda del dato a través de valores únicos como claves primarias; por tanto, se introduce al estudiante en el uso del Sistema Gestor de Base de Datos utilizando el lenguaje SQL que está conformado por un lenguaje de definición de datos y un lenguaje de manipulación de datos.

Finalmente, se aborda el manejo de transacciones para resolver redundancia y colisiones al momento de almacenar datos. La secuencia de notas está diseñada como recurso de apoyo al estudiante de la unidad de aprendizaje Fundamentos de Bases de Datos ofertada en el Centro Universitario UAEM Zumpango.

## **Introducción**

La base de datos ha ido tomando un papel importante a lo largo de los años, debido a que su función más importante es proporcionar la materia prima necesaria a los sistemas de información.

Al irse desarrollando las Bases de Datos han permitido buscar diferentes maneras de almacenar la información y así mismo dándole espacio a modelos lógicos como es la Entidad Relación.

En la primera y segunda unidad se abordara la importancia que tienen las Bases de Datos, así como la representación lógica de la información a través de los modelos Entidad-Relación, Relacional, Objeto-Relacional.

A partir tercera unidad se explicarán aspectos sobre el manejo de las Bases de Datos en un sistema de información, usando bases matemáticas y algebraicas para el uso de lenguajes utilizados para el manejo de Bases de Datos.

En la unidad cuatro se muestra el proceso de normalización, que ha sido de gran ayuda en la eliminación de información redundante e innecesaria.

Finalmente, en la unidad cinco se cubren aspectos de las transacciones y posteriormente las formas sobre el manejo de concurrencia, principalmente con el uso de candados o bloqueos (Universidad Autónoma del Estado de México, 2010).

### **Propósito**

Conocer los aspectos y conceptos fundamentales de Bases de Datos, e ir identificando las ventajas del uso del sistema de datos, así como las funciones que hacen tener a los datos almacenados en las BD íntegros y seguros.

Diseñar e implementar un Sistema de Base de datos con ayuda de software de creación de BD como es SQL (Universidad Autónoma del Estado de México, 2010).

### **Competencias Genéricas**

Diseñar e implementar un Sistema de Base de datos con ayuda de software de creación de BD como es SQL (Universidad Autónoma del Estado de México, 2010).

### **Requisitos**

Para cursar satisfactoriamente la unidad de aprendizaje de Bases de datos el alumno deberá contar con conocimientos previos la materia de:

- ✓ Organización de archivos.
- ✓ Estructuras de datos.
- ✓ Programación Estructurada.

## **Unidad de Competencia I**

**Identificar y establecer los elementos de un Sistema de Base de Datos**



# Fundamentos de Bases de Datos

## Temario

- Sistemas de Información
- Datos
- Información
- Sistemas de archivos
- Formas de almacenamiento

## Objetivos

- Manejar adecuadamente los conceptos y términos del sistema de base de datos
- Profundidad de elección del sistema de información
- Identificar la manera en que funciona la distribución en determinados Sistemas Gestores de Bases de Datos.

## Glosario

- Proceso: denota la ejecución de alguna acción o grupo de acciones.
- Entidad es una persona, un grupo, un departamento o cualquier sistema que recibe o emite información.
- Archivos: Conjunto lógico de información o de datos que se designa con un nombre y se configura como una unidad autónoma completa para el sistema o el usuario.
- Registros: Representa un objeto único de datos implícitamente estructurados en una tabla.
- Campo: Es la mínima unidad de información a la que se puede acceder.
- Datos: Cifra, letra o palabra que se suministra a la computadora como entrada y la máquina almacena en un determinado formato.

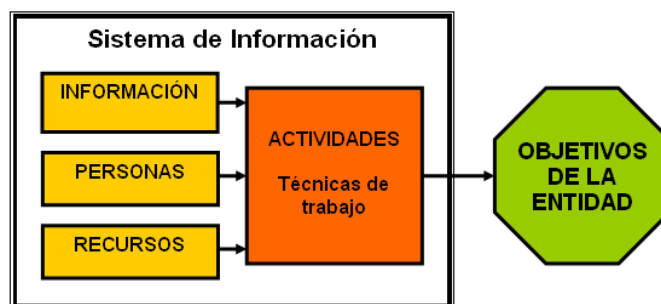
## *Sistemas de información y base de datos*

Un sistema de información (SI) se puede definir técnicamente como un conjunto de componentes relacionados que recolectan, procesan, almacenan y distribuyen información para Apoyar a la toma de decisiones.

Un sistema de información consta de un conjunto organizado de elementos los cuales son:

- ✓ Personas
- ✓ Datos
- ✓ Actividades o técnicas de trabajo
- ✓ Recursos materiales en general

Estos elementos interactúan entre sí para el procesamiento de datos y de la información para así poder distribuirla de una manera adecuada en una organización en función a sus objetivos.



*Ilustración 1 Elementos de un sistema de Información*

Las clasificaciones de los sistemas de información se basan según el entorno de aplicación, se mencionan los siguientes:

- ✓ **Sistemas transaccionales:** la cual sustituye los procedimientos manuales por a otros, basados en computadoras y trata con procesos de rutina bien estructurados, e incluye aplicaciones para el mantenimiento de registros.
- ✓ **Sistema de gestión estratégico:** Sistemas de información que ayudan a los administradores de nivel superior a abordar y resolver cuestiones estratégicas y tendencias a largo plazo tanto de la compañía como su entorno exterior.
- ✓ **Sistema al soporte de decisiones:** Sistema interactivo basado en computadora el cual aporta modelos y datos para resolver problemas no estructurados.



## ***Factores de Entorno de un Sistema de información***

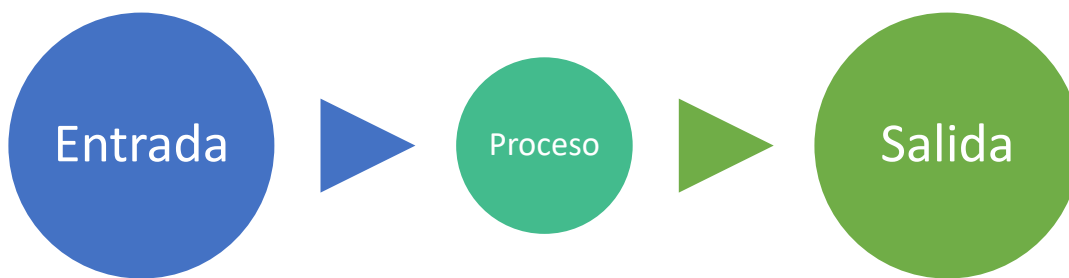
Un sistema de información contiene información sobre una organización y su entorno. Dentro de este entorno figuran clientes, proveedores, competidores, accionistas y agencias reguladoras que interactúan con la organización y sus sistemas de información (Senn, 1992).



*Ilustración 2 Factores que interactúan con los Sistemas de Información*

Existen 3 actividades las cuales producen la información que dicha organización necesita para tomar una decisión, controlar operaciones, analizar problemas y crear nuevos productos o servicios estas son las siguiente:

- ✓ **Entrada:** captura o recolecta datos en bruto tanto del interior de la organización como de su entorno externo.
- ✓ **Procesamiento:** convierte esa entrada de datos en una forma más significativa.
- ✓ **Salida:** transfiere la información procesada a la gente que la usará o a las actividades para las que se utilizará.



*Ilustración 3 Elementos de un Sistema de Información*

Estos sistemas también requieren de una retroalimentación que es la salida que se devuelve al personal de la organización para ayudar a evaluar o corregir la etapa de entrada.

### ***Introducción a la Base de Datos***

Para que se lleve a cabo el tratamiento de información es necesario contar con una organización, un sistema de información con múltiples accesos el cual debe de ser dinámica e incrementar Accesos Aleatorios, es necesario un Software /Hardware.

Los elementos para crear una base de datos son los siguientes:

1. Dato: es el hecho o suceso el cual debe ser cuantificable o cualificable el cual se refiere a los atributos.
2. Analista del sistema: Es el experto en el desarrollo del sistema.
3. Tipos de Almacenamiento: un sistema gestor de base de datos el cual permite la comunicación contando con el CPU, BD Y SO

En este caso una base de datos correctamente diseñada permite obtener acceso a información actualizada y precisa.

## *Independencia de datos respecto a los procedimientos*

La independencia de datos es una forma de gestión de bases de datos que mantiene los datos separados de otros programas que podrían usar esos datos. De forma parecida a una medida adicional de seguridad, esta independencia de datos se asegura de que la información no pueda ser modificada o reorganizada por ningún otro programa. Una de las funciones de la independencia de datos es restringir acceso a la estructura de almacenamiento de los datos por aplicaciones de usuario.

Se requiere lo que es una vista conceptual y una estructura de datos.

Vista conceptual: Como se perciben los datos del problema en el mundo real.

Estructura de datos: Es la forma en la que se almacena por tanto la independencia de la información con respecto a los procedimientos los cuales deben satisfacer a los niveles de abstracción para que esta sea efectiva las cuales son las siguientes.

1. **Independencia lógica:** tiene la capacidad de modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación, estos esquemas conceptuales se pueden modificar al emplear o reducir la base de datos.
2. **Independencia física:** Es la capacidad de modificar el esquema interno son tener que alterar el esquema conceptual, esto se refiere a la separación entre las aplicaciones y las estructuras físicas de almacenamiento.

La independencia física en general es la relación producto -cliente.

Base de Datos
Archivos
Registro
Campo
Sub campo
Datos

*Tabla 1 Independencia física*

#### Definiciones:

- ✓ Base de datos: conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.
- ✓ Archivos: Conjunto lógico de información o de datos que se designa con un nombre y se configura como una unidad autónoma completa para el sistema o el usuario.
- ✓ Registros: Representa un objeto único de datos implícitamente estructurados en una tabla.
- ✓ Campo: Es la mínima unidad de información a la que se puede acceder.
- ✓ Datos: Cifra, letra o palabra que se suministra a la computadora como entrada y la máquina almacena en un determinado formato.
- ✓ Almacenamiento de base de datos

Las bases de datos son creadas para almacenar grandes cantidades de datos de forma permanente por lo que deben ser actualizados y consultados constantemente.

#### Almacenamiento físico de bases de datos

La mayoría de las bases de datos se almacenan en las llamadas memorias secundarias, especialmente discos duros, aunque, en principio, pueden emplearse también discos ópticos, memorias flash, etc.

Las razones por las cuales las bases de datos se almacenan en memorias secundarias son:

- ✓ En general, las bases de datos son demasiado grandes para entrar en la memoria primaria.
- ✓ La memoria secundaria suele ser más barata que la memoria primaria (aunque esta última tiene mayor velocidad).
- ✓ La memoria secundaria es más útil para el almacenamiento de datos permanente, puesto que la memoria primaria es volátil.

#### ***Características para un SBD.***

- ✓ Versatilidad: para la representación de la información
  - Mundo real en donde se encuentra la visión parcial y la visión global.

- ✓ Desempeño: habla sobre el rendimiento y la capacidad
- ✓ Capacidad de Acceso: el cual se encarga de que el dato esté disponible.
- ✓ Simplicidad: son los elementos necesarios para describir la función de un sistema.
- ✓ Integridad: que haya datos incorrectos.
- ✓ Seguridad y Privacidad: salvaguardar mecanismos para que los datos no sufran daños, S.O, H.D, Y RAM.

#### Características principales

- ✓ Existe redundancia controlada e integridad referencia.
- ✓ Implementa mayor seguridad.
- ✓ Existen actualizaciones directas.
- ✓ No tienen independencia ni lógica ni física.
- ✓ Sus componentes son HW, SW, BD y Usuarios.

#### ***Sistema de archivos***

Son normas y procedimientos para poder almacenar la información todo sistema operativo debe de contar con uno. Al sistema de archivos que usan los sistemas operativos se les llama sistemas de archivos nativos y es la manera en que un SO guarda su información.

También nos encontramos con algo que se llama dar formato el cual es un dispositivo de almacenamiento el cual prepara el dispositivo para poder guardar la información.

El sistema de archivos es el manejo de:

- ✓ Archivos en texto plano
- ✓ Búsqueda de tres tipos: secuencial, aleatorio, indexado.
- ✓ No existen actualizaciones, se simulan dando de baja el registro y dando de alta el nuevo con la modificación.
- ✓ Tienen independencia lógica y física la mayoría de ellos.
- ✓ Cuenta con redundancia no controlada y errores de referencia
- ✓ Sus componentes son: archivos, registros y campos.



## **Unidad de Competencia II**

**Generar modelos lógicos/conceptuales de base de datos**



## Modelos lógicos/conceptuales de las bases de datos

### Temario

- Arquitectura de un Sistema de Base de Datos
- Tipos de modelos de Base de Datos
  - Físico
  - Lógico
  - De vistas
- Características de los modelos lógicos/conceptuales

### Objetivos

Que el alumno desarrolle de manera correcta la arquitectura de la base de datos, así como los diagramas, considerando de manera adecuada los esquemas a desarrollar.

### Glosario

- Entidades: es la representación de un objeto o concepto del mundo real que se describe en una base de datos.
- Relaciones: son asociaciones entre tablas que se crean utilizando sentencias de unión para recuperar datos.
- Cliente: es una aplicación informática o una computadora que consume un servicio remoto en otra computadora conocido como servidor, normalmente a través de una red de telecomunicaciones.
- Servidor: es una aplicación en ejecución (software) capaz de atender las peticiones de un cliente y devolverle una respuesta en concordancia. Los servidores se pueden ejecutar en cualquier tipo de computadora, incluso en computadoras dedicadas a las cuales se les conoce individualmente como «el servidor».
- Sistema monousuario: Es la unidad que es utilizada por una sola persona, y que trabaja con un solo sistema operativo que permite un único usuario.
- Sistema multiusuario: Se encarga de dar servicio a varios usuarios, donde en la entrada de información pelean por el espacio en memoria.



## ***Arquitectura de las Bases de Datos***

De acuerdo con Silberschatz, Korth & Sudarshan (2002), las Bases de Datos se ven influenciadas particularmente a la arquitectura de la computadora como puede ser la conexión en red el paralelismo y la distribución. Las diferentes arquitecturas del BD son dos:

### ✓ Centralizada:

Una computadora consiste en una o unas pocas unidades centrales de procesamiento y un número de dispositivos que se encuentran conectados a un bus común el cual les permite el acceso a la memoria compartida.

Esta computadora se encuentra a su vez en dos tipos de sistemas

- Monousuario: Es la unidad que es utilizada por una sola persona, y que trabaja con un solo sistema operativo que permite un único usuario.
- Multiusuario: Se encarga de dar servicio a varios usuarios, donde en la entrada de información pelean por el espacio en memoria.

### ✓ Cliente-Servidor:

Debido a que la tecnología ha dado un gran paso hoy en día, se ha suplantado el sistema Centralizado ya ahora se conoce como cliente-servidor, donde el servidor adopta el papel de Centralizado ya que proporciona al cliente el acceso de forma limitada.

## ***Modelos de las Bases de Datos***

Los modelos de Base de Datos se basan principalmente en la forma en la que se van a extraer los datos.

Para poder recopilar estos datos de forma eficiente y para que los usuarios puedan simplificar la interacción con el sistema, se desarrollan tres niveles que se presentan a continuación del orden más bajo hasta el que tiene mayor orden.

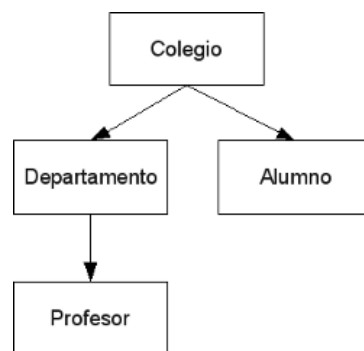
- Nivel físico: Describe detalladamente la forma de almacenar los datos, así como su estructura.
- Nivel lógico: En este nivel se describirán los datos involucrados dentro de la BD, así como la relación que tienen entre ellos, en este nivel se ven involucrados los administradores de la BD.

- Nivel de vistas: Se proporciona una parte de la BD a cada uno de los usuarios para así simplificar la interacción con la misma, debido a que no todos los usuarios necesitan de toda la información.

### ***Modelado de datos***

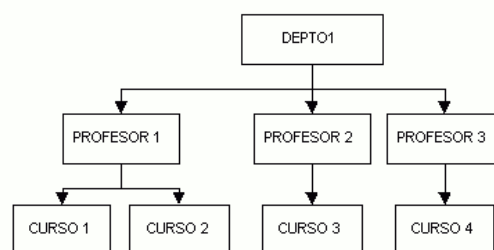
Según Silberschatz el modelo de datos se define como una colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia (Silberschatz, Korth, & Sudarshan, 2002).

- ✓ Modelo jerárquico: Su estructura es similar a la de un organigrama su estructura se conecta mediante puntadores que continúan la dirección del registro relacionado, pero surge un problema debido a que algunos datos se ajustan a una estructura estrictamente y este dato no puede ser modificado.



*Ilustración 4 Modelo de datos jerárquico*

- ✓ Modelo de red: Este modelo es una versión mejorada del modelo jerárquico debido a que presenta la misma estructura, pero adopta la relación de las redes y puede tomar una relación uno a muchos.



*Ilustración 5 Modelo de datos de red*

- Modelo entidad-relación: Este modelo toma como base el mundo real ya que cuenta con entidades y relaciones, las entidades son vistas como objetos o cosa y esta a su vez tiene objetos que son lo que lo describe, por ejemplo, una

case seria la entidad y sus objetos serían el color, las dimensiones, el número de puertas, etc.

Su diagrama consta de distintos componentes como son el rectángulo que representa conjunto de entidades, elipses que representan atributos, rombos que representan relaciones entre conjunto de entidades y Líneas que una a los componentes.

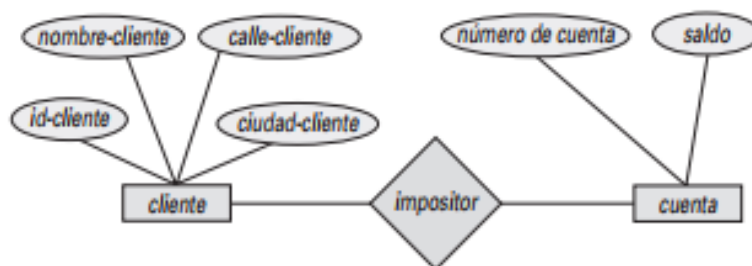


Ilustración 6 Modelo entidad-relación

- Modelo relacional: Este modelo está compuesto por tablas para representar los datos y la relación que existe entre ellos, cada tabla está compuesta por varias columnas y estas columnas tienen un nombre único.

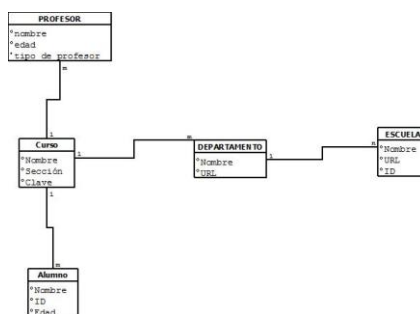


Ilustración 7 Modelo relacional

## **Unidad de Competencia III**

**Aplicar las bases matemáticas para el acceso a una base de datos.**



# Fundamentos del Algebra relacional

## Temario

- Un panorama general del Algebra Relacional.
- Selección
- Proyección
- Producto
- Unión
- Intersección
- Diferencia
- Reunión
- División
- Sintaxis Algebra Relacional
- Otras Operaciones Relacionales

## Objetivos

- Identificar claramente las ventajas y desventajas del uso de Base de Datos.
- Manejar adecuadamente los conceptos de Algebra Relacional y SQL
- Resolver problemas de manera analítica y lógica, Álgebra Relacional y/o Calculo Relacional.

## Glosario

- Base de Datos: Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite.
- Dato: Los datos son números, letras o símbolos que describen objetos, condiciones o situaciones.
- Atributos: Son las columnas de una relación y describen características particulares.
- Tuplas: Cada uno de los renglones en una relación que contiene valores para cada uno de los atributos

- **Algebra Relacional:** Es un conjunto de operaciones que describen paso a paso como computar una respuesta sobre las relaciones, tal y como estas son definidas en el modelo relación.
- **SQL (Structured Query Language):** Es un lenguaje de programación estándar e interactiva para la obtención de información desde una base de datos y para actualizarla.
- **Select:** Corresponde a la operación proyección del algebra relacional. Se usa para listar los atributos deseados del resultado de una consulta.
- **From:** Corresponde a la operación producto cartesiano del algebra relacional. Lista las relaciones que deben ser analizadas en la evaluación de la expresión
- **Where:** Corresponde al predicado que engloba a los atributos de las relaciones que aparecen en la cláusula from.

## **Introducción**

Adquirir los conocimientos necesarios relacionados a la manipulación del modelo relacional, es decir, lo que se denomina Álgebra Relacional y Cálculo Relacional, comenzando por explicar las operaciones básicas del modelo relacional: Seleccionar, Proyectar y Renombrar, seguidamente las operaciones de la teoría matemática de conjuntos: Unión, intersección, la diferencia y el producto cartesiano, además se definirán algunas operaciones adicionales de álgebra relacional. Por otra parte, se presentan los conceptos básicos del cálculo relacional, analizando la sintaxis de las consultas, empleando variables de tuplas y de dominio

### ***Un Panorama General del Algebra Relacional.***

El álgebra relacional es un lenguaje de consulta procedimental. Consta de un conjunto de operaciones que toman como entrada una o dos relaciones y producen como resultado una nueva relación. Las operaciones fundamentales del algebra relacional son selección, proyección, unión, diferencia de conjuntos, producto cartesiano y renombramiento. Además de las operaciones fundamentales hay otras operaciones, por ejemplo, intersección de conjuntos, reunión natural, división y asignación. Estas operaciones se definirán en términos de operaciones fundamentales (Ayala, 2015).

- ✓ **SELECCIÓN:** Extrae las tuplas especificadas de una relación dada (o sea, restringe la relación sólo a las tuplas que satisfagan una condición específica).
- ✓ **PROYECCIÓN:** Extrae los atributos especificados de una relación dada.
- ✓ **PRODUCTO:** A partir de dos relaciones especificadas, constituye una relación que contiene todas las combinaciones posibles de tuplas, una de cada una de las dos relaciones.
- ✓ **UNIÓN:** Construye una relación formada por todas las tuplas que aparecen en cualquiera de las dos relaciones especificadas.
- ✓ **INTERSECCIÓN:** Construye una relación formada por aquellas tuplas que aparezcan en las dos relaciones especificadas.
- ✓ **DIFERENCIA:** Construye una relación formada por todas las tuplas de la primera relación que no aparezcan en la segunda de las dos relaciones específicas.
- ✓ **REUNIÓN:** A partir de dos relaciones especificadas, construye una relación que contiene todas las posibles combinaciones de tuplas, una de cada una de las dos relaciones, tales que las dos tuplas participantes en una combinación dada satisfagan alguna condición específica.
- ✓ **DIVISIÓN:** Toma dos relaciones, una binaria y una unaria, y construye una relación formada por todos los valores de un atributo de la relación binaria que concuerdan (en el otro atributo) con todos los valores en la relación unaria.

**Nota:** Existe una analogía obvia entre la capacidad de anidar expresiones algebraicas en álgebra relacional y la capacidad de anidar expresiones aritméticas en aritmética ordinaria. En efecto, el hecho de que las relaciones estén cerradas en el álgebra es importante exactamente por las mismas razones por las que es importante el hecho de que los números estén cerrados en la aritmética ordinaria. (Ayala, 2015).

### *Una Sintaxis para el Álgebra Relacional.*

#### **Selección**

La operación SELECCIONAR ( $\sigma$ ) sirve para elegir un subconjunto de tuplas de una relación que satisfacen una condición de selección. Por ejemplo, para

seleccionar el subconjunto de tuplas EMPLEADO que trabajan en el departamento 4 o cuyo salario rebasa los \$3,000.00 podemos especificar las dos condiciones con la operación SELECCIONAR, y expresarlas de la siguiente manera:

$\sigma_{DNO=4 \text{ OR SALARIO} > \$3,000.00}$  (EMPLEADO)

En general, denotamos la operación SELECCIONAR con  $\sigma_{(\text{condición de selección})}$  (<Nombre de la relación >) donde:

- ✓ El símbolo  $\sigma$  (sigma) denota el operador de SELECCIONAR,
- ✓ Y la condición de selección es una expresión booleana especificada en términos de los atributos de la relación.

La relación que resulta de la operación SELECCIONAR tiene los mismos atributos que la relación especificada en <nombre de la relación>. La expresión booleana especificada en <condición de selección> se compone de una o más cláusulas de la forma, <nombre del atributo>, <operador de comparación>, <valor constante> o <nombre del atributo>, <operador comparación>, <nombre del atributo>; donde:

- ✓ Se asigna el nombre de un atributo en <nombre de la relación>,
- ✓ <operador de comparación > es normalmente uno de los operadores {=, <=, >=, <>}
- ✓ Se debe tomar un valor constante del dominio del atributo.
- ✓ Mientras las cláusulas pueden conectarse arbitrariamente con los operadores booleanos Y (AND), O (OR) y NO (NOT) para formar una condición de selección en general
- ✓ Si queremos seleccionar las tuplas de todos los empleados que trabajan en el departamento 4 y ganan más de \$2,500.00 o que trabajan en el departamento 5 y ganan más de \$3,000.00, podemos especificar la siguiente operación:
  - SELECCIONAR:  $\sigma_{(DNO=4 \text{ Y SALARIO} > 2500) \text{ O } (DNO=5 \text{ Y SALARIO} > 3000)}$  (EMPLEADO)

El resultado se muestra a continuación:



nombre	paterno	materno	matricula	nacimiento	direccion	sexo	Salario	supervisor	dno
francisco	lopez	lopez	45555	8/12/45	sn diego	m	4000	00007	5
jennifer	ortega	lozano	54321	6/20/31	toquila	f	4300	00007	4
ramses	sanchez	mendez	84444	9/15/52	sn miguel	m	3800	45555	5

Ilustración 8 Resultado de aplicar  $\sigma$  (DNO=4 Y SALARIO>2500)

El operador SELECCIONAR es unario; esto es, se aplica a una sola relación. Por añadidura, la operación de selección no puede abarcar más de una tupla. El grado de la relación resultante de una operación SELECCIONAR es el mismo que el de la relación original R a la que se aplicó la operación, porque tiene los mismos atributos de R. (Ayala, 2015).

## PROYECCIÓN ( $\Pi$ )

Si visualizamos una relación como una tabla, la operación SELECCIONAR elige algunas filas de la tabla y desecha otras. La operación PROYECCIÓN, en cambio, selecciona ciertas columnas de la tabla y desecha las demás. Por ejemplo, si queremos el apellido, en nombre y el salario de todos los empleados, podemos usar la siguiente operación:

$\Pi$ (PATERNO, NOMBRE, SALARIO (EMPLEADO))

paterno	nombre	Salario
bond	james	9000
lopez	francisco	4000
espana	josefa	2500
ortega	jennifer	4300
perez	juan	3000
sanchez	ramses	3800
zelaya	alicia	2500
hidalgo	aldama	2500

Ilustración 9 Resultado de aplicar  $\Pi$ (PATERNO, NOMBRE, SALARIO (EMPLEADO))

La forma general de la operación PROYECTAR es  $\Pi$ <LISTA DE ATRIBUTOS> (<nombre de la relación>) donde:

- ✓  $\Pi$  ( $\pi$ ) es el símbolo de la operación PROYECTAR
- ✓ y <lista de atributos> es una lista de atributos de la relación especificada por <nombre de la relación>. La relación así creada tiene sólo los atributos especificados en <lista de atributos> y en el mismo orden en que aparecen en

lista. Por ello, su grado es igual al número de atributos en <lista de atributos> Si la lista de atributos sólo contiene atributos no llave de una relación, es probable que aparezcan tuplas repetidas en el resultado.

La operación PROYECTAR elimina implícitamente cualquier tupla repetida, así que el resultado de la operación PROYECTAR es un conjunto de tuplas y por tanto una relación válida. Por ejemplo, consideremos la siguiente operación (Ayala, 2015):

$\Pi_{\text{SEXO, SALARIO}}(\text{EMPLEADO})$

sexo	salario
M	9000
M	4000
F	2500
F	4300
M	3000
M	3800
F	2500
M	2500

Ilustración 10 Resultado de aplicar  $\Pi_{\text{SEXO, SALARIO}}(\text{EMPLEADO})$

### Secuencia de operaciones y cambio de nombre de los atributos.

En general, es posible que deseemos aplicar varias operaciones del álgebra relacional una tras otra. Para ello, podemos escribir las operaciones en una sola expresión del álgebra relacional anidándolas, o bien, podemos aplicar una operación a la vez y crear relaciones de resultados intermedios. En el segundo caso, tendremos que nombrar las relaciones que contienen los resultados intermedios. Por ejemplo, si queremos obtener el nombre de pila, el apellido y el salario de todos los empleados que trabajan en el departamento no. 5, debemos aplicar una operación SELECCIONAR y una operación PROYECTAR. Podemos escribir una sola expresión del álgebra relacional, a saber:

$\Pi_{\text{PATERNO, NOMBRE, SALARIO}}(\sigma_{\text{DNO}=5}(\text{EMPLEADO}))$ .

Como alternativa, podemos mostrar explícitamente la secuencia de operaciones, dando un nombre a cada una de las relaciones intermedias, como sigue:

$\text{EMPLS\_DEP5} \leftarrow \sigma_{\text{DNO}=5}(\text{EMPLEADO})$  RESULTADO  $\leftarrow \Pi_{\text{NOMBRE, PATERNO, SALARIO}}(\text{EMPL\_DEP5})$  (Ayala, 2015).

A menudo es más sencillo descomponer una secuencia compleja de operaciones especificando relaciones de resultados intermedios que escribiendo una sola expresión del álgebra relacional. También podemos usar esta técnica para cambiar el nombre de los atributos de las relaciones intermedias y de la resultante. Esto puede ser útil cuando se trata de operaciones más complejas. Si queremos cambiar de nombre los atributos de una relación resultante al aplicar una operación de álgebra relacional, bastará con que incluyamos una lista con los nuevos nombres de atributos entre paréntesis, como el siguiente ejemplo:  $TEMP \leftarrow \sigma_{DNO=5}(EMPLEADO)$

$$R_{(NOMPILA, APPATERNO, SALARIO)} \leftarrow \Pi_{NOMBRE, PATERNO, SALARIO}(TEMP)$$

Si no se cambian los nombres, los atributos de la relación resultante de una operación SELECCIONAR serán los mismos que los de la relación original y estarán en el mismo orden. En el caso de una operación PROYECTAR sin cambio de nombres, la relación resultante tendrá los mismos nombres de atributos que aparecen en la lista de proyección, y en el mismo orden (Ayala, 2015).

### *Lenguaje SQL*

- ✓ Una vista en MySQL se define utilizando la orden CREATE VIEW, para definir una vista se le debe dar un nombre y se debe construir una consulta que genere dicha vista, la forma de CREATE VIEW es la siguiente:

Ejemplo:

```
CREATE VIEW Todos-Los-Clientes
AS (SELECT Nombre-Sucursal-Nombre-Cliente
FROM Impositor, Cuenta
WHERE Impositor. Número, Cuenta= Cuenta. Número-Cuenta)
```

- ✓ Un borrado se expresa de igual modo que una consulta. Se pueden borrar tuplas completas, un borrado en SQL se expresa:

```
DELETE FROM R
WHERE P
```

Ejemplo:

```
DELETE FROM Cuenta
WHERE Nombre-Sucursal
```

- ✓ Para insertar datos en una relación se especifica la tupla que se desea insertar, los atributos de las tuplas que se inserten deberán pertenecer al mismo dominio.

Ejemplo:

```
INSERT INTO Cuenta
VALUES ('C-9732', 'navacerrada', 1200)
```

- ✓ En determinadas situaciones puede ser deseable cambiar un valor dentro de una tupla sin cambiar todos los valores de la misma, para este tipo de situaciones se utiliza la instrucción update.

Ejemplo:

```
UPDATE Cuenta
SET Saldo = Saldo * 1.05
```

- ✓ Una transacción consiste en una secuencia de instrucciones de consulta y actualizaciones, una de las siguientes instrucciones SQL debe finalizar la transacción.
  - Commit work: Compromete la transacción actual, hace que los cambios realizados sean permanentes en la base de datos. Después de que se comprometa la transacción se inicia una nueva transacción automática.
  - Rollback work: Causa el retroceso de la transacción actual, es decir, deshace todas las actualizaciones realizadas por las instrucciones SQL de la transacción.

Las insuficiencias en el estándar SQL ANSI/ISO y las diferencias entre los dialectos SQL son suficientemente significativas para que una aplicación deba ser siempre modificada cuando se pasa de una base de datos SQL a otra. Estas diferencias incluyen:

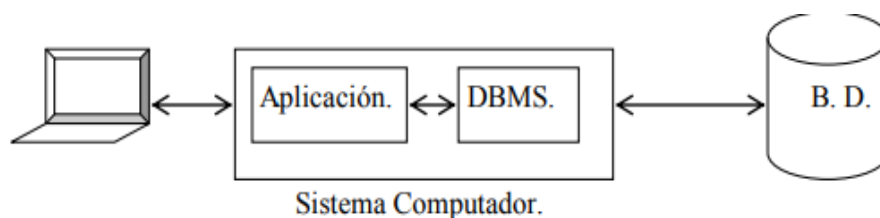
- Código de error. El estándar no especifica los códigos de error a devolver cuando SQL detecta un error.
- Tipos de datos. El estándar define un conjunto mínimo de tipos de datos, pero omite algunos de los más populares y útiles, como las fechas y horas.
- Tablas de sistema. El estándar no dice nada sobre las tablas de sistema que proporcionan información referente a la estructura de la propia BD's. Cada vendedor tiene su propia estructura, e incluso

las cuatro implementaciones de SQL de IBM se diferencian una de la otra

- SQL interactivo. El estándar especifica el SQL programado utilizado por un programa de aplicación, y no el SQL interactivo.
- Interfaz de programa. El estándar utiliza una técnica abstracta para utilizar SQL desde dentro de un programa de aplicación escrito en COBOL, C, FORTRAN, etc., ningún producto SQL comercial utiliza esta técnica, y hay variaciones considerables en las interfaces reales de programas utilizados.
- SQL dinámico. El estándar no incluye las características para desarrollar herramientas como escritores de informes.

La creciente popularidad de la conexión de computadoras por red durante los últimos años ha tenido un fuerte impacto en la gestión de bases de datos y ha dado a SQL una nueva prominencia. Conforme las redes pasan a ser más comunes, las aplicaciones que han corrido tradicionalmente en una minicomputadora o maxi computadora tradicional se están transfiriendo a redes de área local con estaciones de trabajo de sobremesa y servidores. En estas redes SQL juega un papel crucial como vínculo entre una aplicación que corre en una estación de trabajo y el DBMS que gestiona los datos compartidos en el servidor (Ayala, 2015).

**Arquitectura centralizada:** La arquitectura de base de datos tradicional utilizada por DB2 y las BD's sobre minis como Oracle e Ingres se muestran en la siguiente figura.



*Ilustración 11 Gestión de la base de datos en arquitectura centralizada*

En esta arquitectura el DBMS y los datos físicos residen ambos en un sistema maxi o minicomputador central, junto con el programa de aplicación esta arquitectura el DBMS y los datos físicos residen ambos en un sistema maxi o minicomputador

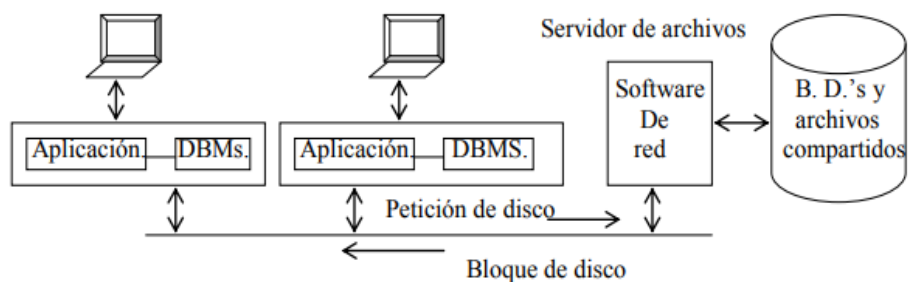
central, junto con el programa de aplicación. Como el sistema es compartido por muchos usuarios, cada usuario experimenta una degradación del rendimiento cuando el sistema tiene una carga fuerte. (Ayala, 2015).

### Arquitectura de servidor de archivos.

En esta arquitectura, una aplicación que corre en un computador personal puede acceder de forma transparente a datos localizados en un servidor de archivos, que almacena los archivos compartidos. Cuando una aplicación en PC solicita datos de un archivo compartido, el software de red recupera automáticamente el bloque solicitado del archivo en el servidor. En este caso, cada PC ejecuta su propia copia del software DBMS.

Esta arquitectura puede producir un fuerte tráfico de red y un bajo rendimiento para ciertas consultas. (Ayala, 2015).

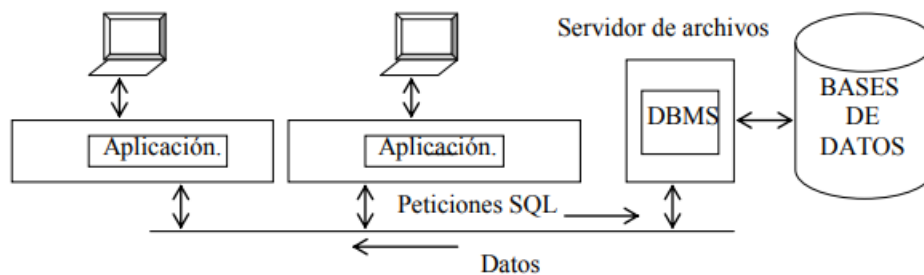
### Arquitectura cliente/servidor



*Ilustración 12 Gestión de base de datos en una arquitectura servidora de archivos*

Las computadoras personales están combinadas en una red de área local junto con el servidor de base de datos que almacena los BD's compartidos. Las funciones del DBMS están divididas en dos partes. Las "frontales" (front-end) de base de datos, tales como herramientas de consulta interactiva, escritores de informes y programas de aplicación, se ejecutaban en PC. La máquina de soporte (back-end) de la BD que almacena y gestiona los datos se ejecuta en el servidor. SQL se ha convertido en el lenguaje de BD estándar para comunicación entre las herramientas frontales y la máquina de soporte de esta arquitectura. La arquitectura cliente/servidor reduce el tráfico de red y divide la carga de trabajo de la BD's las funciones intensivas tales como visualización de datos se concentran en la PC. Las funciones intensivas en

procesos de datos como el proceso de consultas se realizan en el servidor. (Ayala, 2015).



*Ilustración 13 Gestión de bases de datos en una arquitectura cliente/servidor*

## Resumen

- ✓ SQL está basado en el álgebra relacional formal
- ✓ SQL incluye varias constructoras de lenguaje para consultas sobre la base de datos
- ✓ Todas las operaciones de algebra relacional pueden ser usadas en SQL
- ✓ Las relaciones de vistas se pueden definir como relaciones que contienen el resultado de consultas.
- ✓ Las vistas son útiles para no mostrar información innecesaria
- ✓ SQL incluye constructoras para actualizar borrar o insertar información
- ✓ Una transacción consiste en una secuencia de operaciones que deben ser atómicas.
- ✓ Las modificaciones sobre la base de datos pueden conducir a la generación de valores nulos en las tuplas (Ayala, 2015).

## **Unidad de Competencia IV**

**Utilizar las reglas de normalización en el diseño de base de datos.**





# Normalización de Bases de Datos

## Temario

- Definición de normalización.
- Grados de normalización.

## Objetivos

- Identificar claramente del uso de la normalización.
- Reconocer los pasos de normalización.

## Glosario

- Redundancia: hace referencia al almacenamiento de los mismos datos varias veces en diferentes lugares, la redundancia de datos puede provocar problemas como:
  - Incremento del trabajo.
  - Desperdicio de espacio de almacenamiento.
  - Consistencia de datos.
- Integridad: El proceso de normalización de bases de datos consiste en aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional.

Las bases de datos relacionales se normalizan para:

- Evitar redundancias de los datos.
- Evitar problemas de actualización.
- Proteger la integridad de los datos.

## *Normalización de Base de Datos.*

La normalización también hace las cosas fáciles de entender. La normalización es el proceso mediante el cual se transforma datos complejos a un conjunto de estructuras de datos más pequeñas, que además de ser más simples y estables, son más fáciles de entender. También se puede entender la normalización como una serie de reglas que sirven para ayudar a los diseñadores de bases de datos a desarrollar un esquema

que minimice los problemas de lógica. Las guías que normalización provee crean el marco de referencia para simplificar una estructura de datos compleja.

Regla	Descripción
Primera Forma Normal (1FN)	Incluye la eliminación de todos los grupos repetidos.
Segunda Forma Normal (2FN)	Asegura que todas las columnas que no son llave sean completamente dependientes de la llave primaria (PK).
Tercera Forma Normal (3FN)	Elimina cualquier dependencia transitiva. Una dependencia transitiva es aquella en la cual las columnas que no son llave son dependientes de otras columnas que tampoco son llave.

*Ilustración 14 Descripción de las Reglas de Normalización*

Otra ventaja de la normalización de base de datos es el consumo de espacio. Una base de datos normalizada ocupa menos espacio en disco duro que una no normalizada. Hay menos repetición de datos, lo que tiene como consecuencia menor espacio de memoria. El proceso de normalización tiene un nombre y una serie de reglas para cada fase. Esto puede parecer un poco confuso al principio, pero poco a poco se va entendiendo el proceso, así como las razones para hacerlo de esta manera. (Escuela de Educacion Tecnica No. 2, 2003)

### ***Grados de normalización.***

#### **Primera Forma Normal.**

La regla de la primera forma normal establece que las columnas repetidas deben eliminarse y colocarse en tablas separadas. Poner la base de datos en la primera forma normal resuelve el problema de los encabezados de columna múltiples. La normalización ayuda a clarificar la base de datos y a organizar en partes más pequeñas y más fáciles de entender. En lugar de tener que entender una tabla gigantesca y monolítica que tiene muchos diferentes aspectos, solo tenemos que entender los objetos pequeños y más tangibles (Escuela de Educacion Tecnica No. 2, 2003).

### **Segunda Forma Normal.**

La regla de la segunda forma normal establece que todas las dependencias parciales se deben eliminar y separar dentro de sus propias tablas. Una dependencia parcial es el término que describe aquellos datos que no dependen de la llave primaria de la tabla para identificarlos (Escuela de Educacion Tecnica No. 2, 2003).

### **Tercera Forma Normal.**

Una tabla está normalizada en esta forma si todas las columnas que no son llaves son funcionalmente dependientes por completo de la llave primaria y no hay dependencias transitivas. Cuando las tablas están en su Tercera Forma Normal se previenen errores de lógica cuando se insertan o borran registros. Cada columna en una tabla está identificada de manera única por la llave primaria, y no debe haber datos repetidos (Escuela de Educacion Tecnica No. 2, 2003).

### **Límite de la normalización.**

La siguiente decisión es **¿Qué tan lejos llevar la normalización?** La normalización es una ciencia subjetiva. Determinar las necesidades de simplificación depende de nosotros. Si nuestra base de datos va proveer información a un solo usuario para un propósito simple y existente pocas posibilidades de expansión, normalizar los datos hasta la 3FN quizá sea algo exagerado. Las reglas de normalización existen como guías para crear tablas que sean fáciles de manejar, así como flexibles y eficientes. A veces puede ocurrir que normalizar los datos hasta el nivel más alto no tenga sentido.

Existen seis niveles más de normalización que no se han discutido aquí. Ellos son forma normal la 4FN, 5FN. Estas formas de normalización pueden llevar las cosas más allá de lo que necesitamos. Estas existen para hacer una base de datos realmente relacional. Tienen que ver principalmente con dependencias múltiples y claves relacionales (Escuela de Educacion Tecnica No. 2, 2003).

### Ejemplo de Normalización:

En el primer nivel de la Normalización se tiene que tomar los siguientes pasos:

- 1: Eliminar los grupos repetitivos de las tablas individuales
- 2: Crear una tabla por separado por cada grupo de datos relacionados
- 3: Identificar cada grupo de datos relacionados con cada clave primaria

Por ejemplo, en la siguiente tabla de Usuarios se podría decir que está en el caso 0 de la Normalización ya que no se están tomando en cuenta los pasos anteriores, ya que en este caso se está rompiendo la primera regla puesto que se repiten los campos de sitios web1 y sitios web2, otro problema podría ser la llave primaria no se toma en cuenta la regla tres.

Para todo esto lo iremos resolviendo de la siguiente manera para cada campo tipo contador un auto incremento para cada registro en otro caso que pasa como en este caso si se tienen 2 usuarios con el mismo nombre y queremos diferenciarlos.

Usuarios					
ID_Usuarios	Nombre	Empresa	Direccion Empresa	Sitio Web1	Sitios Web2
1	Pedro Lago	Grupo Lala	Paseo Rancho Viejo	www.lala.com	www.grupola.com
1	Pedro Lago	Grupo Lala	Paseo Rancho Viejo	www.lala.com	www.grupola.com
2	Perla Mondragón	Grupo Bimbo	Naucalpan de Juárez	www.bimbo.com	www.grupobi.com
2	Perla Mondragón	Grupo Bimbo	Naucalpan de Juárez	www.bimbo.com	www.grupobi.com
3	Carlos Espinoza	Grupo Sabritas	Cuautitlán Izcalli	www.sabritas.com	www.gruposa.com

*Tabla 2 Relación Usuarios*

Ahora en este otro caso podemos decir que la tabla está en el primer caso de la normalización ya que se ha solucionado el problema de la limitante del campo de sitios web.

Sin embargo, se tienen otros problemas porque cada vez que se introduce un valor nuevo del registro de la tabla usuarios se tiene que duplicar el nombre de la tabla empresa y del usuario con esto no solo la base crecerá sino que con esto no solo es muy sencillo de corromperse por lo que se puede escribir mal alguno de los datos redundantes

Así que se aplica el segundo nivel de la normalización y tiene los siguientes pasos:

- 1: Crear tablas separadas para aquellos grupos de datos que se aplicaran para varios datos
- 2: Relacionar estas tablas mediante una clave externa

Usuarios			
ID_Usuarios	Nombre	Empresa	Dirección Empresa
1	Pedro Lago	Grupo Lala	Paseo Rancho Viejo
2	Perla Mondragón	Grupo Bimbo	Naucalpan de Juárez
3	Carlos Espinoza	Grupo Sabritas	Cuautitlán Izcalli

*Tabla 3 Relación Usuarios*

SitiosWeb		
ID_Sitios web	ID_Usuarios	Url
1	1	www.lala.com
2	1	www.grupola.com
3	2	www.bimbo.com
4	2	www.grupobi.com
5	3	www.sabritas.com
6	3	www.gruposa.com

*Tabla 4 Relación SitiosWeb*

En la clave primaria en la tabla de usuarios está relacionada ahora con la clave externa de la tabla sitios web

No importa que la clave externa difiera del nombre de la clave primaria siempre y cuando este correcta.

Esto sin duda mejora el modelado de datos; sin embargo, que ocurre si en dado caso se quiere añadir a otro empleado de la empresa grupo lala o 50 más de grupo bimbo ahora tendríamos el nombre de la empresa y su dirección multiplicándose otra situación que puede inducir o introducir errores de nuestros datos que tenemos actuales.

En dado caso se tiene que aplicar el nivel 3 de la Normalización que nos indica:

1: Eliminar aquellos campos que no dependan de la clave, por ejemplo, los datos que tenemos aquí, el nombre de la empresa y su dirección no tiene nada que ver con su campo Id\_Usuarios sin embargo tiene que ver con su llave primaria.

Usuarios		
ID_Usuarios	Nombre	ID_Empresa
1	Pedro Lago	1
2	Perla Mondragón	2
3	Carlos Espinoza	3

*Tabla 5 Relación Usuarios con clave externa ID\_Empresa*

Empresas		
ID_Empresa	Nombre	Dirección
1	Grupo Lala	Paseo Rancho Viejo
2	Grupo Bimbo	Naucalpan de Juárez
3	Grupo Sabritas	Cuautitlán Izcalli

*Tabla 6 Relación Empresas*

SitiosWeb		
ID_Sitios web	ID_Usuarios	Url
1	1	www.lala.com
2	1	www.grupola.com
3	2	www.bimbo.com
4	2	www.grupobi.com
5	3	www.sabritas.com
6	3	www.gruposa.com

*Tabla 7 Relación SitiosWeb con claves externas*

## Resumen

La normalización es una técnica que se utiliza para crear relaciones lógicas apropiadas entre tablas de una base de datos. Ayuda a prevenir errores lógicos en la manipulación de datos. La normalización facilita también agregar nuevas columnas sin romper el esquema actual ni las relaciones.

Existen varios niveles de normalización: Primera Forma Normal, Segunda Forma Normal, Tercera Forma Normal, Forma Normal Boyce-Codd, Cuarta Forma Normal, Quinta Forma Normal o Forma Normal de Proyección-Unión, Forma Normal de Proyección-Unión Fuerte, Forma Normal de Proyección-Unión Extra Fuerte y Forma Normal de Clave de Dominio. Cada nuevo nivel o forma nos acerca más a hacer una base de datos verdaderamente relacional.

Se retomaron las primeras tres formas. Éstas proveen suficiente nivel de normalización para cumplir con las necesidades de la mayoría de las bases de datos. Normalizar demasiado puede conducir a tener una base de datos ineficiente y hacer a su esquema demasiado complejo para trabajar. Un balance apropiado de sentido común y práctico puede ayudarnos a decidir cuándo normalizar.

## **Unidad de Competencia V**

**Analizar la concurrencia de las transacciones en un sistema de administración de base de datos**





## Transacciones en un Sistema de Bases de Datos

### Temario

- Transacciones
- Candados
- Ventajas y Desventajas de los candados
- Concurrencia de las transacciones

### Objetivos

- Utilización del lenguaje SQL para manejo de transacciones.
- Administrar las transacciones.
- Identificar y resolver los problemas generados por el uso de candados.

### Glosario

- Base de Datos: Es un almacén que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente.
- Bloqueo: Un bloqueo o ban es una restricción (total, parcial, permanente o temporal) que se impone a un usuario dentro de un sistema informático. Lo habitual es que el usuario sea bloqueado (o baneado) cuando viola las normas de uso de un servicio determinado.
- Serialización: Servidores de Aplicaciones: Es un mecanismo para transporte de objetos de datos en diferentes contextos de los sistemas informáticos. Su característica es el procesamiento de dichos datos uno a la vez.
- Sistema Manejador de Base de Datos: También conocido como Sistema Gestor de Base de Datos, son aplicativos que tienen como objetivo facilitar la construcción y manipulación de bases de datos sirviendo como interfaz entre éstas, los usuarios y los distintos tipos de aplicaciones que las utilizan. Sus siglas pueden ser SMBD o SGBD.

## **Introducción**

Los administradores de base de datos tienen varias responsabilidades en los procesos de control del rendimiento para lo cual uno de los elementos básicos a detectar es el control de concurrencia de distintos usuarios. (Silberschatz, 2002)

El control de transacciones concurrentes en una base de datos brinda un eficiente desempeño del Sistema de Base de Datos, puesto que permite controlar la ejecución de transacciones que operan en paralelo, ingresando a información compartida y, por lo tanto, interfiriendo potencialmente unas con otras.

SQL es un lenguaje de programación de propósito especial diseñado para manejar datos en una base de datos relacional, y es utilizado por un gran número de aplicaciones y organizaciones. Contiene una variedad de características y herramientas que se pueden utilizar para desarrollar y administrar bases de datos y soluciones de todo tipo basadas en ellas.

## ***Transacciones***

Una transacción es una unidad de la ejecución de un programa que accede y posiblemente actualiza varios elementos de datos. Una transacción se inicia por la ejecución de un programa de usuario escrito en un lenguaje de manipulación de datos de alto nivel o lenguaje de programación, y está delimitado por instrucciones de la forma inicio de transacción y fin de transacción. (Elmasri, 2007)

Es necesario que las transacciones tengan las propiedades ACID: atomicidad, consistencia, aislamiento y durabilidad.

La atomicidad asegura que bien o todos o ninguno de los efectos se reflejen; un fallo no puede dejar a la base de datos en un estado en el cual una transacción se haya ejecutado parcialmente.

La consistencia asegura que, si la base de datos es consistente inicialmente, la ejecución de la transacción (debido a la misma) deja la base de datos en un estado consistente.

El aislamiento asegura que, en la ejecución concurrente de transacciones, están aisladas entre sí de tal manera que cada una tiene la impresión de que ninguna otra transacción se ejecuta concurrentemente con ella.

La durabilidad asegura que, una vez de la transacción se ha comprometido, las actualizaciones hechas por la transacción no se pierden incluso si hay un fallo del sistema.

La ejecución concurrente de transacciones mejora la productividad y la utilización del sistema, y también reduce el tiempo de espera de las transacciones. Cuando varias transacciones se ejecutan concurrentemente en la base de datos, puede que deje de conservarse la consistencia de los datos (Coronel & Morris, 2011)

- ✓ Una transacción debe estar en uno de los estados siguientes:

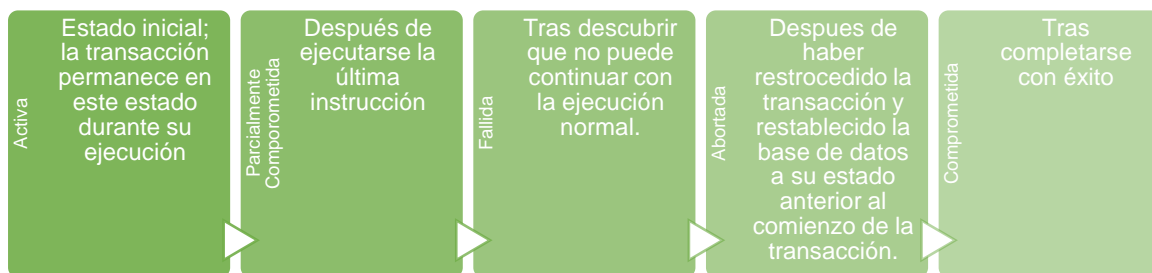
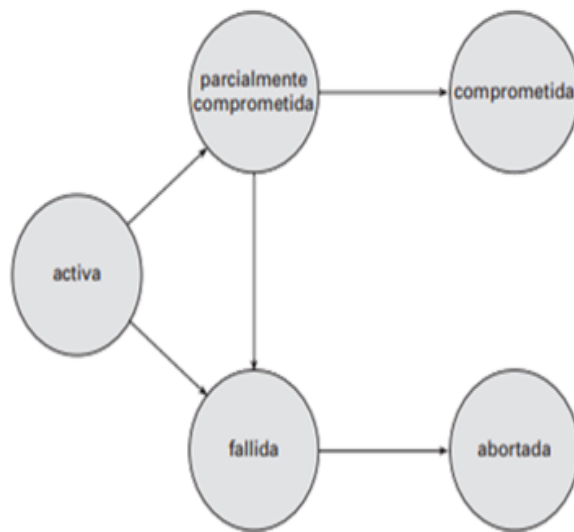


Ilustración 15 Descripción de los estados de las transacciones



*Ilustración 16 Proceso de una transacción reflejada en su estado*

## ***Candados***

Variable asociada a un elemento de información de la base de datos y describe el estado de ese elemento respecto a las posibles operaciones que se pueden aplicar a él. En general, hay un candado por cada elemento de información en la base de datos. Son usados como una forma de transacciones concurrentes. El gestor de bloqueos (subsistema del SGBD) gestiona y controla el acceso a los candados. (Abelló & Rollón, 2006)

### ***Tipos De Candados:***

1. Binarios
2. De modo múltiple

### **Técnica de Bloqueo: Candados Binarios**

Puede tener dos estados o valores: bloqueado y desbloqueado. A cada elemento X de la base de datos se asocia un candado distinto. Si el valor del candado sobre x es 1, ninguna operación de la base de datos que solicite el elemento podrá tener acceso a él. Si el valor de candado sobre X es 0, se podrá tener acceso al elemento cuando se solicite.

Una transacción solicita acceso a un elemento X emitiendo una operación `bloquear_elemento(X)`.

Si  $Candado(X)=1$ , la transacción tiene que esperar; si no, la transacción asigna 1 a  $Candado(X)$  (bloquea el elemento) y tiene que esperar; si no, la transacción obtiene el acceso. Cuando la transacción termina de usar el elemento, emite una operación `desbloquear_elemento(X)`, que asigna un 0 a  $Candado(X)$  (desbloquea elemento) para que otras transacciones puedan tener acceso a X. Un candado binario impone una exclusión mutua sobre el elemento de información.

Cuando se utiliza el bloqueo binario, toda transacción debe obedecer las siguientes reglas:

- 1.- Una transacción T debe emitir una operación `bloquear_elemento(X)` antes de que se realice cualquier operación `leer_elemento(X)` o `escribr_elemento(X)` en T.
- 2.- Una transacción T debe emitir la operación `desbloquear_elemento(X)` después de haber completado todas las operaciones `leer_elemento(X)` y `escribr_elemento(X)` en T.
- 3.- Una transacción T no emitirá una operación `bloquear_elemento(X)` si ya posee el bloqueo del elemento X.
- 4.- Una transacción T no emitirá una operación `desbloquear_elemento(X)` a menos que ya posea el bloqueo del elemento X.

### **Técnica De Bloqueo: Candados Múltiples**

Permite que varias transacciones tengan acceso al mismo elemento X si lo hace exclusivamente para leerlo. Sin embargo, si una transacción va a escribir un elemento X deberá poseer acceso exclusivo a él. En este esquema hay tres operaciones de bloqueo: `bloquear_lectura(X)`, `bloquear_escritura(X)` y `desbloquear(X)`. Ahora un  $Candado(X)$ , tiene tres posibles estados: bloqueo para leer, bloqueo para escribir o desbloqueado.

Las reglas en los bloqueos múltiples son (Marques, 2011):

- ✓ La transacción T debe ejecutar la operación bloquear\_lectura o bloquear\_escritura antes de que realice cualquier operación leer\_elemento. Luego debe ejecutar desbloquear\_elemento.
- ✓ La transacción T debe ejecutar la operación bloquear\_escritura antes de que realice cualquier operación escribir\_elemento. Luego debe ejecutar desbloquear\_elemento.
- ✓ No se ejecutan bloqueos sobre un elemento si ya tiene un bloqueo sobre este mismo. Si se requiere se pueden permitir excepciones.

### ***Ventajas y Desventajas***

La gran finalidad de los bloqueos es que impiden los conflictos de actualización. Se presentan cuando usuarios requieren la lectura o modificación de datos que en ese instante están siendo modificados por otros usuarios. Entre las características básicas de los bloqueos tenemos:

- ✓ Permiten la serialización de las transacciones, haciendo que un usuario a la vez pueda modificar un dato.
- ✓ Son necesarios en las transacciones simultáneas ya que permiten que usuarios accedan y actualicen datos en mismos intervalos de tiempo.

### ***Ejemplo de una conexión de bases de datos en netbeans:***

Para la elaboración de una conexión de una interfaz java a una base de datos es necesario seguir algunos pasos para realizarlo de la manera correcta y que no se presenten errores. En este ejemplo que se llevara a cabo, la base de datos está elaborada para el registro de productos de una papelería.

La interfaz cuenta con un menú donde dichos productos se pueden dar de alta, dar de baja y buscar.

Los requerimientos para este ejemplo:

- ✓ Es necesario contar con Wampserver instalado en el ordenador.

1. Se crea la interfaz que se adecue a la información a guardar, para esto utilizamos el IDE NetBeans.
2. Se inserta un JFrame el cual servirá para el acceso a la interfaz.

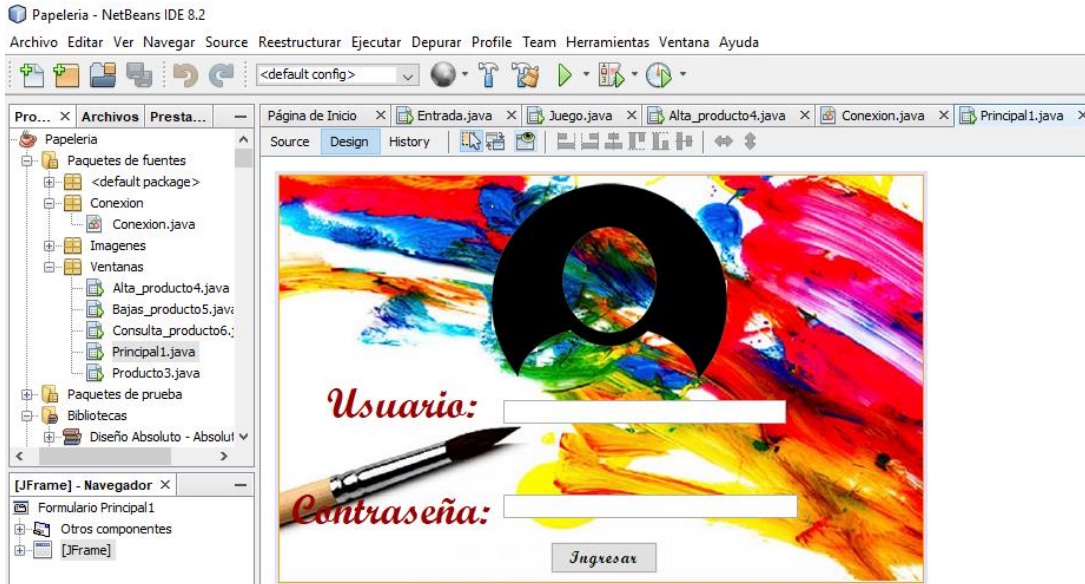


Ilustración 17 Interfaz de logueo en Netbeans

3. Se crea un menú donde se muestren las acciones que se realizaran en nuestra base de datos.

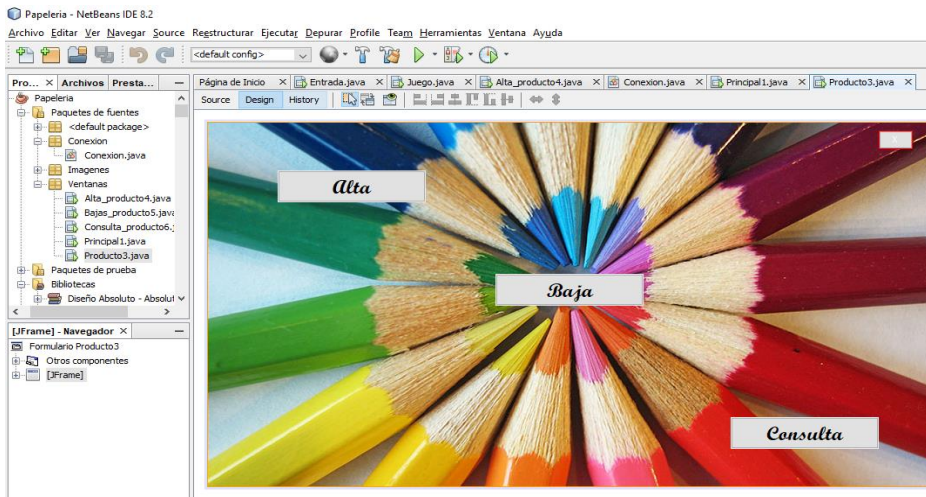


Ilustración 18 Interfaz de menú

4. Se agrega acción al botón alta, el cual abrirá el formulario para el registro de productos.

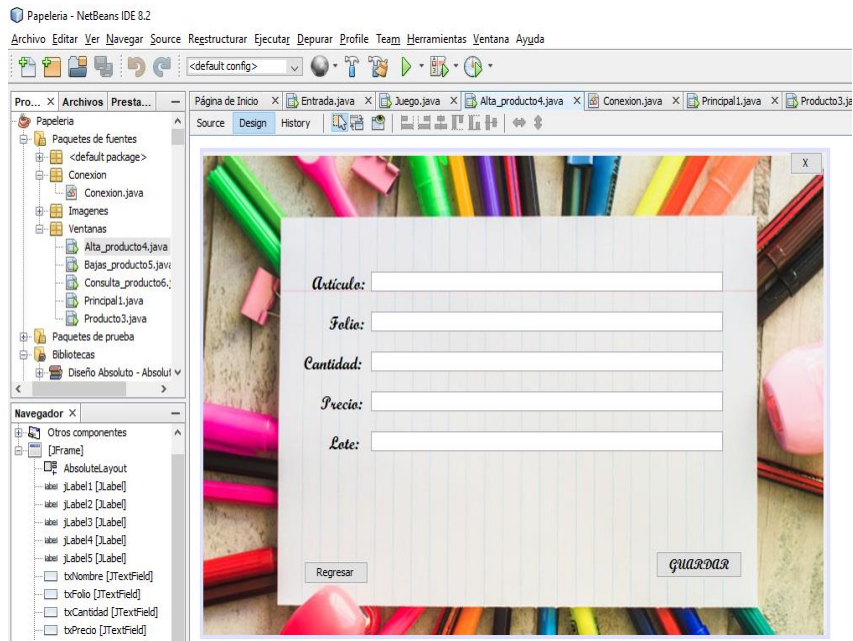


Ilustración 19 Interfaz registro de productos

- Realizamos la acción del botón guardar, creando el código para que el contenido en los campos de texto se guarde en la base de datos.

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    Conexion cc = new Conexion();
    Connection cn = cc.conectar();
    if (cn != null) {
        JOptionPane.showMessageDialog(null, "Conexion Exitosa");
    } else {
        JOptionPane.showMessageDialog(null, "Error al conectar BD");
    }

    String art, fol, cant, prec, lote;
    String sql = "";
    art = txNombre.getText();
    fol = txFolio.getText();
    cant = txCantidad.getText();
    prec = txPrecio.getText();
    lote = txLote.getText();

    sql = "INSERT INTO producto (articulo, ID_producto, cantidad, precio, lote) VALUES (?, ?, ?, ?, ?)";
    try {
        PreparedStatement pst = cn.prepareStatement(sql);
        pst.setString(1, art);
        pst.setString(2, fol);
        pst.setString(3, cant);
        pst.setString(4, prec);
        pst.setString(5, lote);
        int n = pst.executeUpdate();
        if (n > 0) {
            JOptionPane.showMessageDialog(null, "Registro guardado");
        }
    } catch (SQLException ex) {
        Logger.getLogger(Alta_producto4.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

Ilustración 20 Codificación para insertar registro en la base de Datos



- Se crea el formulario de bajas el cual, para eliminar el producto primero se tendrá que buscar con el nombre, por lo tanto, se crea la acción del botón “buscar”.

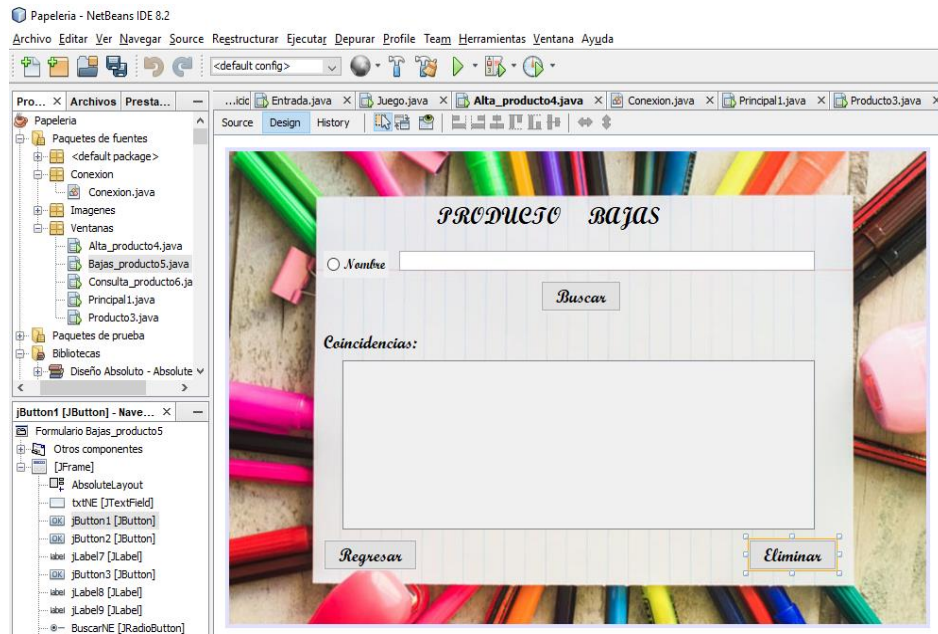


Ilustración 21 Interfaz baja de registro

- Se agregan las acciones en el botón eliminar:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    DefaultTableModel model = (DefaultTableModel) TableU.getModel();
    int fila = TableU.getSelectedRow();
    String valor = TableU.getValueAt(fila, 0).toString();
    if(fila>=0)
    {
        try {
            PreparedStatement pst = cn.prepareStatement(sql);
            PreparedStatement pst = cn.prepareStatement("DELETE FROM producto WHERE Nombre='"+valor+"'");
            pst.executeUpdate();
            JOptionPane.showMessageDialog(null,"dato eliminado");
        } catch (SQLException ex) {
            Logger.getLogger(Bajas_producto5.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    // TODO add your handling code here:
}
}
```

Ilustración 22 Código de eliminación de registro

- Se crea una clase de nombre conexión, esta nos permitirá conectar la interfaz a la base de datos, es este caso nuestra base de datos es llamada “papelería”

```

~/
package Conexion;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

/**
 *
 * @author Jesus
 */
public class Conexion {
    Connection conectar = null;
    public Connection conectar () {
        try {

            Class.forName("org.gjt.mm.mysql.Driver");
            conectar = DriverManager.getConnection("jdbc:mysql://localhost/nombre de la base de datos","usuario","contraseña");//
        } catch (ClassNotFoundException | SQLException e) {
            {
                System.out.println(" ERROR AL CONECTAR"+e.getMessage());
            }
            return conectar;
        }
    }
}

```

*Ilustración 23 Codificación de la Conexión de Base de Datos en Java*

## 9. Creación de la base de datos en MySQL.

```

1  CREATE DATABASE papeleria;
2  use papeleria;
3
4  CREATE TABLE producto(
5
6  articulo varchar(50),
7  ID_producto int AUTO_INCREMENT,
8  cantidad int,
9  precio float,
10  lote varchar(10),
11  PRIMARY KEY(ID_producto)
12  );
13
14

```

*Ilustración 24 Query para creación de base de datos y tabla*

10. Para verificar que nuestra base de datos funciona, ejecutamos la aplicación y llenamos los campos de la ventana alta y se guardan.

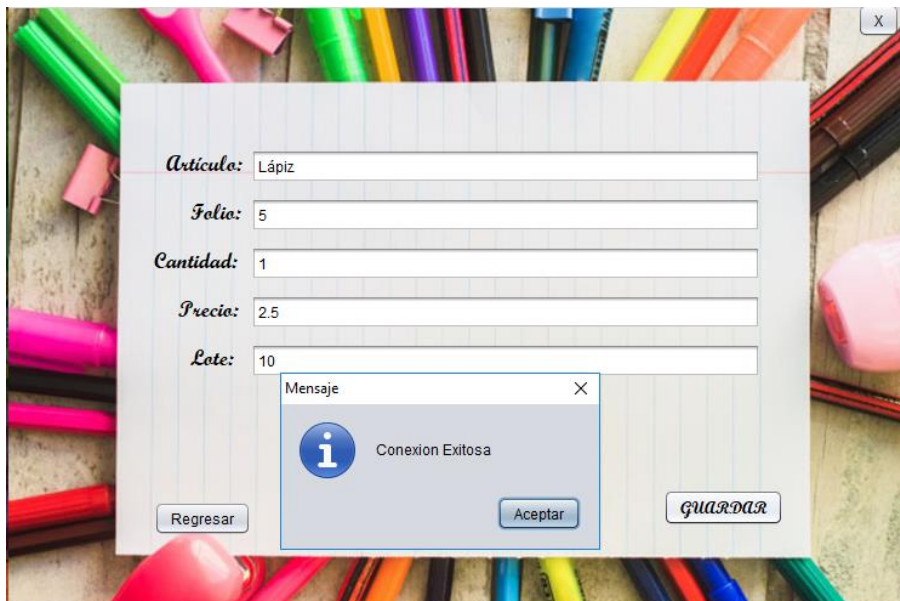


Ilustración 25 Prueba de Conexión con la Base de Datos

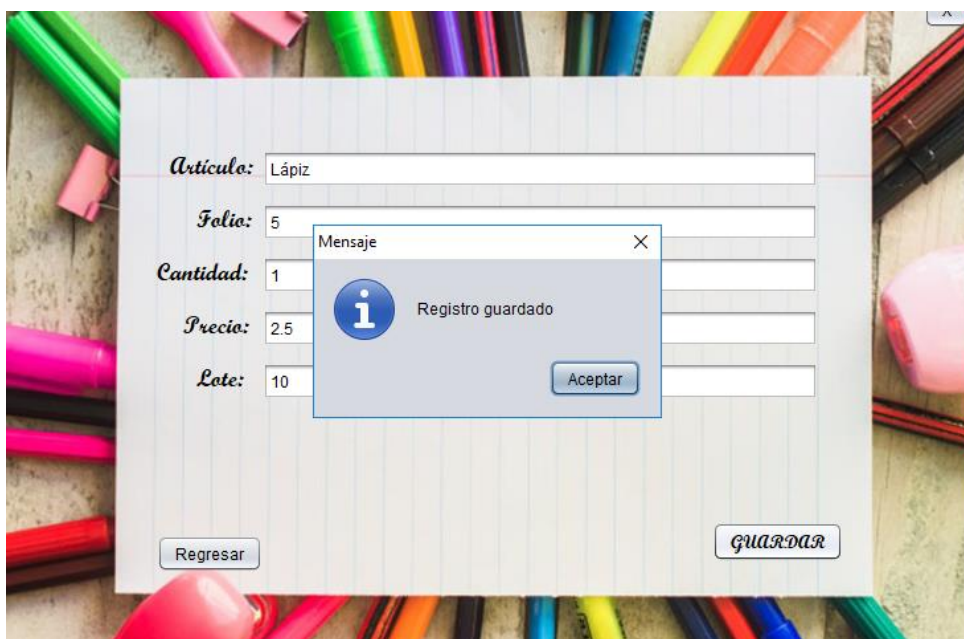


Ilustración 26 Alta de Registro

## 11. Para concluir verificaremos la base de datos en phpMyAdmin.



*Ilustración 27 Consulta de registro en la Base de Datos*

## Conclusión

A través del desarrollo de las notas se enuncian los elementos claves para que el estudiante conozca las ventajas de las bases de datos que se superponen a los sistemas de archivos del pasado, así como la evolución del modelo relacional que implementa SQL, mostrando la excelente herramienta que es en la administración, seguridad y fiabilidad de los datos.

Por lo tanto, se sabe que conocer el proceso interno, la estructura e implementación de base de datos se muestra la importancia que realizan en el mundo laboral, y como cada ente que las utiliza es dependiente de ellas.

## Recomendaciones

Principalmente vemos la necesidad de conocer cada día más el entorno de las bases de datos. Aprender de manera didáctica y autodidáctica con mayor dedicación. Conocer las especificaciones que nos presenta cuando

estructuramos las tablas de cada base de datos, realizando nuestro trabajo más práctico y sencillo.



## Índice de Tablas e Ilustraciones

Ilustración 1 Elementos de un sistema de Información .....	8
Ilustración 2 Factores que interactúan con los Sistemas de Información .....	9
Ilustración 3 Elementos de un Sistema de Información.....	10
Ilustración 4 Modelo de datos jerárquico.....	18
Ilustración 5 Modelo de datos de red .....	18
Ilustración 6 Modelo entidad-relación .....	19
Ilustración 7 Modelo relacional .....	19
Ilustración 8 Resultado de aplicar $\sigma$ (DNO=4 Y SALARIO>2500) .....	25
Ilustración 9 Resultado de aplicar $\Pi$ PATERNO, NOMBRE, SALARIO (EMPLEADO).....	25
Ilustración 10 Resultado de aplicar $\Pi$ SEXO, SALARIO (EMPLEADO).....	26
Ilustración 11 Gestión de la bases de datos en arquitectura centralizada .....	29
Ilustración 12 Gestión de base de datos en una arquitectura servidora de archivos	30
Ilustración 13 Gestión de bases de datos en una arquitectura cliente/servidor.....	31
Ilustración 14 Descripción de las Reglas de Normalización .....	34
Ilustración 15 Descripción de los estados de las transacciones .....	43
Ilustración 16 Proceso de una transacción reflejada en su estado .....	44
Ilustración 17 Interfaz de logueo en Netbeans.....	47
Ilustración 18 Interfaz de menú .....	47
Ilustración 19 Interfaz registro de productos .....	48
Ilustración 20 Codificación para insertar registro en la base de Datos .....	48
Ilustración 21 Interfaz baja de registro .....	49
Ilustración 22 Código de eliminación de registro .....	49
Ilustración 23 Codificación de la Conexión de Base de Datos en Java .....	50
Ilustración 24 Query para creación de base de datos y tabla .....	50
Ilustración 25 Prueba de Conexión con la Base de Datos .....	51
Ilustración 26 Alta de Registro .....	51
Ilustración 27 Consulta de registro en la Base de Datos.....	52

## Referencias

- Abelló, A., & Rollón, E. (2006). *Diseño y Administracion de Bases de Datos* . Barcelona: Edicions UPC .
- Ayala, J. (2015). *Apuntes de Fundamentos de Bases de Datos*. Obtenido de <http://ri.uaemex.mx/bitstream/handle/20.500.11799/33944/secme-19274.pdf?sequence>
- Coronel, C., & Morris, S. (2011). *Bases de Datos* . Mexico, DF: CENGAGE Learning.
- Elmasri, R. A. (2007). *Fundamentos de Sistemas de Bases de Datos*. Madrid: Redes Informaticas.
- Escuela de Educacion Tecnica No. 2. (2003). *Normalizacion de Base de Datos*. Buenos Aires.
- Marques, M. (2011). *Bases de Datos* . Castelló de la Plana.
- Senn, J. (1992). *Sistemas de Información*. México: Iberoamericana.
- Silberschatz, A., Korth, H., & Sudarshan, S. (2002). *Fundamentos de Bases de Datos*. Aravaca(Madrid): McGRAW-HILL/INTERAMERICANA DE ESPAÑA.
- Universidad Autónoma del Estado de México. (2010). *Programa de Estudios por Competencias Fundamentos de Bases de Datos*.