



Modalidad Abierta y a Distancia

Administración de Base de Datos

Guía didáctica



Facultad de Ingenierías y Arquitectura

Departamento de Ciencias de la Computación y Electrónica

Administración de Base de Datos

Guía didáctica

Carrera	PAO Nivel
▪ <i>Tecnologías de la Información</i>	V

Autor:

Encalada Encalada Ángel Eduardo



D R B D _ 3 0 2 0

Asesoría virtual
www.utpl.edu.ec

Universidad Técnica Particular de Loja

Administración de Base de Datos

Guía didáctica

Encalada Encalada Ángel Eduardo

Diagramación y diseño digital:

Ediloja Cía. Ltda.

Telefax: 593-7-2611418.

San Cayetano Alto s/n.

www.ediloja.com.ec

edilojacialtda@ediloja.com.ec

Loja-Ecuador

ISBN digital - 978-9942-39-767-6



**Reconocimiento-NoComercial-CompartirIgual
4.0 Internacional (CC BY-NC-SA 4.0)**

Usted acepta y acuerda estar obligado por los términos y condiciones de esta Licencia, por lo que, si existe el incumplimiento de algunas de estas condiciones, no se autoriza el uso de ningún contenido.

Los contenidos de este trabajo están sujetos a una licencia internacional Creative Commons **Reconocimiento-NoComercial-CompartirIgual 4.0 (CC BY-NC-SA 4.0)**. Usted es libre de **Compartir – copiar y redistribuir el material en cualquier medio o formato. Adaptar – remezclar, transformar y construir a partir del material citando la fuente, bajo los siguientes términos: Reconocimiento- debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios.** Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciatante. **No Comercial-no puede hacer uso del material con propósitos comerciales. Compartir igual-Si remezcla, transforma o crea a partir del material, debe distribuir su contribución bajo la misma licencia del original.** No puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia. <https://creativecommons.org/licenses/by-nc-sa/4.0/>

Índice

1. Datos de información.....	8
1.1. Presentación de la asignatura	8
1.2. Competencias genéricas de la UTPL	8
1.3. Competencias específicas de la carrera.....	8
1.4. Problemática que aborda la asignatura.....	9
2. Metodología de aprendizaje.....	9
3. Orientaciones didácticas por resultados de aprendizaje	10
Primer bimestre.....	10
Resultado de aprendizaje 1 y 2	10
Contenidos, recursos y actividades de aprendizaje	10
 Semana 1	10
 Unidad 1. Administración de una base de datos	11
1.1. Panorámica de la administración de un sistema de base de datos	11
1.2. Diseño físico	16
Actividades de aprendizaje recomendadas	20
 Semana 2	21
1.3. Implementación y carga	21
Actividades de aprendizaje recomendadas	24
 Resultado de aprendizaje 3 y 4	24
Contenidos, recursos y actividades de aprendizaje	24
 Semana 3	25
1.4. Seguridad de la base de datos	25
1.5. Gestión de la confidencialidad	28
Actividades de aprendizaje recomendadas	32
Autoevaluación 1.....	34
 Semana 4	38
1.6. Gestión de la disponibilidad y rendimiento.....	38

Actividades de aprendizaje recomendadas	45
Resultado de aprendizaje 5.....	45
Contenidos, recursos y actividades de aprendizaje	46
Semana 5	46
1.7. Gestión de la integridad y control de concurrencia.....	46
Actividades de aprendizaje recomendadas	51
Resultado de aprendizaje 6, 7 y 8.....	52
Contenidos, recursos y actividades de aprendizaje	52
Semana 6	52
1.8. Respaldo y recuperación.....	52
Actividades de aprendizaje recomendadas	55
Resultado de aprendizaje 9, 10 y 11.....	56
Contenidos, recursos y actividades de aprendizaje	56
Semana 7	56
1.9. Arquitecturas de acceso a bases de datos	56
Actividades de aprendizaje recomendadas	63
Autoevaluación 2.....	64
Semana 8	68
Actividades finales del bimestre.....	68
Segundo bimestre	69
Resultado de aprendizaje 12.....	69
Contenidos, recursos y actividades de aprendizaje	69
Semana 9	69
Unidad 2. Bases de datos distribuidas	69
2.1. Panorámica de las bases de datos distribuidas	69
Actividades de aprendizaje recomendadas	75

Resultado de aprendizaje 13 y 14	75
Contenidos, recursos y actividades de aprendizaje	76
Semana 10	76
2.2. Patrones de distribución de datos	76
Actividades de aprendizaje recomendadas	78
Resultado de aprendizaje 15 y 16	79
Contenidos, recursos y actividades de aprendizaje	79
Semana 11	80
2.3. Aplicaciones de las bases de datos distribuidas	80
Actividades de aprendizaje recomendadas	85
Autoevaluación 3.....	87
Resultado de aprendizaje 17.....	91
Contenidos, recursos y actividades de aprendizaje	91
Semana 12	91
Unidad 3. Conceptos avanzados de bases de datos	91
3.1. Soporte a las decisiones.....	91
Actividades de aprendizaje recomendadas	101
Semana 13	102
3.2. Procesamiento analítico en línea	102
Actividades de aprendizaje recomendadas	110
Semana 14	111
3.3. Bases de datos de propósito especial.....	111
3.4. Gestión del conocimiento.....	116
Actividades de aprendizaje recomendadas	117
Resultado de aprendizaje 18, 19 y 20.....	118
Contenidos, recursos y actividades de aprendizaje	119

Semana 15	119
3.5. Conectores y APIs	119
Actividades de aprendizaje recomendadas	122
Autoevaluación 4.....	123
Semana 16	126
Actividades finales del bimestre.....	126
4. Solucionario	127
5. Referencias bibliográficas	136



1. Datos de información

1.1. Presentación de la asignatura



1.2. Competencias genéricas de la UTPL

- Orientación a la innovación y a la investigación.
- Pensamiento crítico y reflexivo.
- Trabajo en equipo.

1.3. Competencias específicas de la carrera

- Construir modelos específicos de ciencias de la computación mediante esquemas matemáticos y estadísticos, para propiciar el uso y explotación eficiente de datos e información.
- Administrar los servicios de tecnologías de información de la organización utilizando buenas prácticas de la industria asegurando la continuidad operacional del negocio.

1.4. Problemática que aborda la asignatura

Desarrollar las capacidades que permitan, por un lado, realizar funciones de administración de bases de datos usando herramientas y procedimientos apropiados para garantizar la integridad y seguridad de los datos y, por otro lado, comprender y dimensionar el potencial uso que tienen las bases de datos en otros ámbitos de aplicación tal como el apoyo a la toma decisiones empresariales, la gestión del conocimiento y otros.



2. Metodología de aprendizaje

El estudio de la asignatura de Administración Bases de Datos lo desarrollaremos aplicando principalmente el método de aprendizaje basado en casos, el cual le permitirá aplicar sus experiencias de aprendizaje para la resolución de un problema real o hipotético dado. Esta estrategia busca que usted se convierta en generador de soluciones encaminadas a resolver casos reales o hipotéticos, facilitando el desarrollo de un pensamiento crítico, y acortando la distancia entre la teoría y la práctica. A través del aprendizaje autónomo podrá interiorizar el escenario del problema planteado, y reflexionar sobre posibles estrategias de solución; luego, con la guía del docente deberá seleccionar la solución más idónea, la que llevará a ejecución en el marco del aprendizaje práctico experimental mediante la aplicación de los métodos, técnicas y herramientas estudiadas.



3. Orientaciones didácticas por resultados de aprendizaje



Primer bimestre

Resultado de aprendizaje 1 y 2

- Distingue entre la administración de datos y administración de bases de datos.
- Realiza las funciones básicas de un administrador de datos, incluida la planificación de la base de datos, análisis, diseño, implementación, mantenimiento y protección.

Por medio de estos dos resultados de aprendizaje usted comprenderá cómo funciona el ciclo de desarrollo de la base de una base de datos, contrastando entre lo que es un administrador de datos y un administrador de base de datos que son los principales roles que intervienen en el proceso. Así mismo, los resultados le permitirán desarrollar la capacidad para construir el diseño físico de una base de datos, y para ejecutar la implementación con una carga inicial de datos.

Contenidos, recursos y actividades de aprendizaje



Semana 1

¡Estimado estudiante, bienvenido!

Durante el primer bimestre nos centraremos en estudiar el proceso y las principales actividades que se desarrollan en el marco de la administración de una base de datos.

Es importante aclarar que para llevar a la práctica la mayoría de los temas que aquí estudiaremos, lo haremos usando el SGBD¹ [MySQL](#), y la herramienta de administración [MySQL Workbench](#).

Unidad 1. Administración de una base de datos

1.1. Panorámica de la administración de un sistema de base de datos

Es muy valioso que antes de iniciar el estudio de esta asignatura, usted recuerde algunos de los conceptos más importantes relacionados con los sistemas de bases de datos.

Revise el siguiente recurso de aprendizaje que le servirá para recordar y reforzar su compresión acerca de la arquitectura, funciones, componentes, y tipos de usuarios de un SGBD:



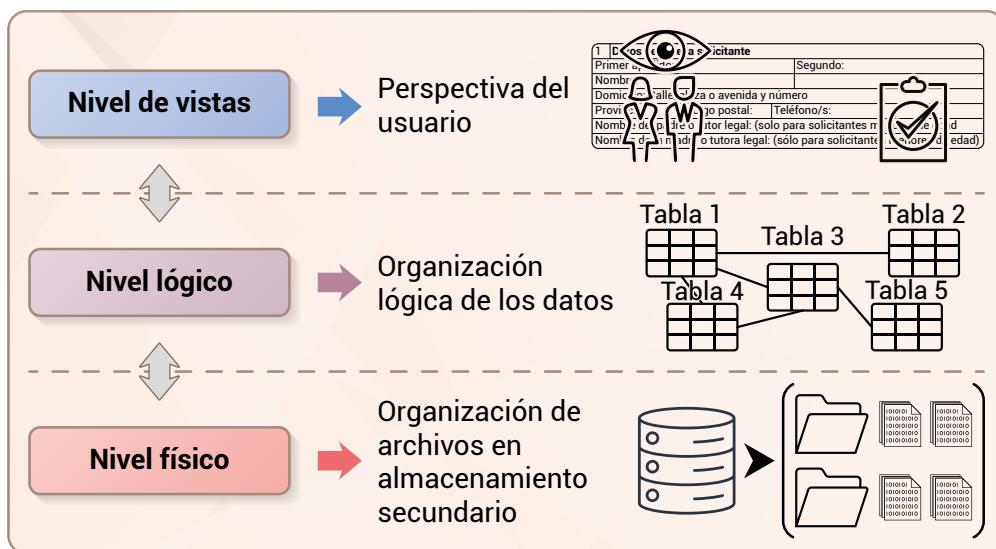
[Revisión de conceptos de bases de datos](#)

Texto básico (Hueso Ibáñez, 2016), apartados 1.2 a 1.5, pp. 10 -14.

Con base en lo explicado por Hueso Ibáñez (2016), es importante que no pierda de vista los tres niveles de abstracción de la arquitectura de los sistemas de base de datos (figura 1), son muy importantes de cara a comprender varios temas que abordaremos más adelante.

Figura 1

Niveles de abstracción de los datos



Nota. Encalada, A., 2023.

Ahora, entrando en materia, es necesario saber que cuando hablamos de administración de un sistema de base de datos nos referimos al proceso de gestionar y mantener una base de datos para garantizar de que esté disponible, protegida y optimizada para satisfacer las necesidades de una organización. Implica una serie de tareas clave, las cuales se exponen en el siguiente recurso:

Aspectos clave administración bases de datos

En resumen, la administración de una base de datos es un proceso complejo que requiere una planificación cuidadosa, una instalación y configuración adecuadas, el mantenimiento y la optimización continua de la base de datos, la implementación de medidas de seguridad sólidas y la capacidad de brindar soporte y resolver problemas de manera efectiva.

Es importante que la construcción de una base de datos se realice de forma ordenada, aplicando un proceso y una metodología adecuados, independientemente del tamaño de la base de datos. Se debe cumplir el proceso establecido a fin de lograr una implementación que garantice la integridad y la seguridad de los datos.

Ciclo de vida de una base de datos

El desarrollo de una base de datos pasa por un proceso que se conoce como ciclo de vida. Empieza por analizar las necesidades de datos (estudio inicial), continúa en la tarea de diseño (conceptual, lógico y físico), para su posterior implementación, pruebas, y puesta en operación; pasando al final a una fase de monitoreo y mantenimiento permanente, tal como lo muestra la figura 2.

Figura 2

Ciclo de vida de la base de datos (DBLC)



Nota. Adaptado de *Bases de Datos. Diseño, implementación y administración* (p.378), por Coronel, C., Morris, S. y Rob, P., 2011, Cengage Learning.

En el marco de este proceso, hay dos roles o funciones claramente identificadas, que es importante saber diferenciar: administración de datos, y administración de la base de datos.

Administrador de datos vs administrador de la base de datos

En los orígenes de las bases de datos, la gestión de los datos y de la base de datos eran tareas que las asumía una misma persona, cuya función se conoce como **Administrador de Datos**, quien se encargaba de la planeación, el diseño, la implementación y el mantenimiento de la base de datos. Más, actualmente, el gran crecimiento de los datos y la evolución de los SGBD cada vez más complejos y especializados, llevaron al surgimiento de un rol específico para quien se encarga de la gestión técnica de la base de datos, conocido como **Administrador de Base de Datos o DBA**.



Para entender mejor el rol Administrador de Datos, revise el siguiente recurso en el que se explica el concepto, alcance, coste, y beneficios de dicha función:

[Administración de datos](#)

Biblioteca Virtual UTPL - *Ebook Central* (Beynon-Davies, 2018), apartados 22.1 a 22.5, pp. 365 - 369.

Como puede ver, actualmente el administrador de datos es un rol de alto nivel dentro de las empresas, protagonista del análisis y diseño del sistema de bases de datos, y encargado de gestionar los recursos de datos dentro de la organización.



Revise el siguiente recurso, el cual le permitirá comprender mejor el alcance de la función de Administración de una Base de Datos, y las características fundamentales del administrador de la base de datos, conocido como DBA:

[Administración de la base de datos](#)

Biblioteca Virtual UTPL - *Ebook Central* (Beynon-Davies, 2018), apartados 23.1 a 23.3, pp. 375 - 377.

Como explica Beynon-Davies (2018), el DBA es un puesto netamente técnico, que toma el control del proceso de construcción de la base de datos a partir del diseño físico.

Resumiendo, digamos entonces que el DBA es el especialista técnico experto en el SGBD elegido y quien asume el control, custodia y realiza

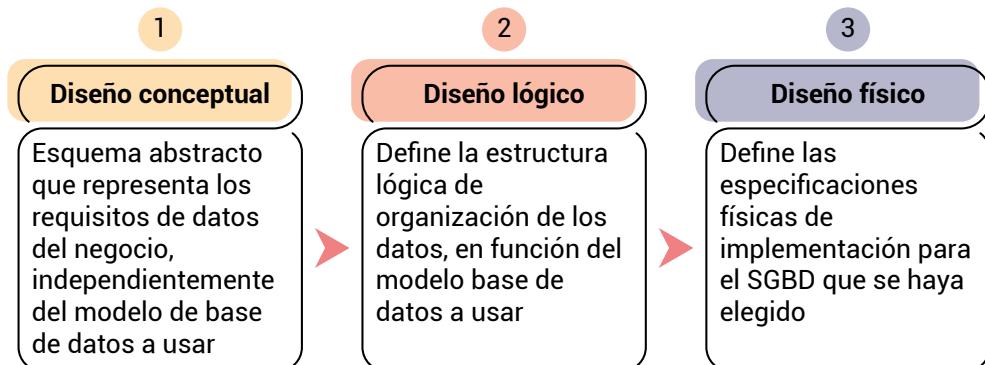
mantenimiento de la base de datos en operación, con facultades para implementar controles y medidas (técnicas y administrativas) que garanticen una operación segura y eficiente. En cambio, el administrador de los datos tiene una función más estratégica de planeación, diseño (a nivel conceptual) y aprovechamiento de los datos. El diseño lógico de la base de datos también lo suele asumir el administrador de datos, ya que no es imprescindible aún saber manejar el SGBD, aunque sí debe comprender los conceptos del modelo de base de datos a usar.

Proceso de diseño de una base de datos

El diseño de la base de datos determina en un alto grado el éxito o fracaso de la implementación final. Por ello es muy importante que usted recuerde con claridad el proceso que se debe seguir para diseñar una base de datos, tal que garantice la integridad y seguridad de los datos. La figura 3 nos resume las 3 etapas del diseño de una base de datos y el alcance de cada una.

Figura 3

Proceso de diseño de una base de datos



Nota. Encalada, A., 2023.

No olvide que el diseño conceptual (Modelo E-R) y diseño lógico (Modelo Relacional), ya lo estudió en la asignatura de Fundamentos de Base de Datos, por lo que es relevante que vuelva a revisar dichos conceptos. Es importante que recuerde que el modelo es la representación del diseño; por ejemplo, el Modelo E-R es una manera de analizar y representar el diseño conceptual (pueden existir otros modelos). Así mismo, el modelo relacional, permite representar el diseño lógico de una base de datos relacional; si usáramos otro tipo de base de datos (por ejemplo, NoSQL) deberíamos usar el modelo lógico acorde al tipo de base de datos elegido.

1.2. Diseño físico

Veremos ahora cómo se establecen las especificaciones físicas de implementación de la base de datos a partir del diseño conceptual y lógico, que usted aprendió en la asignatura de Fundamentos de Bases de Datos.

Mientras que el diseño conceptual se ocupa del “para qué” y el diseño lógico se ocupa del “qué”, el diseño físico se ocupa del “cómo”. Aquí es donde se adapta el diseño lógico a un SGBD concreto. Es decir, en el diseño físico se establecen todas las especificaciones técnicas para implementar la base de datos. Por lo que, en este punto es necesario haber elegido el SGBD que se va a emplear, y conocer a fondo las capacidades y prestaciones del motor.

Aquí es donde aparece el rol conocido como DBA (Administrador de la Base de Datos), que es aquel especialista en el SGBD elegido, y es quien se encarga de todo el proceso que sigue partiendo del diseño físico (implementación, carga, pruebas, puesta en operación, y mantenimiento), por lo tanto, el DBA es quien está en capacidad de aprovechar al máximo las prestaciones del motor, de cara a garantizar seguridad y eficiencia en la operación de la base de datos.



Revise el siguiente recurso que le permitirá comprender el proceso del diseño físico de una base de datos, aplicado a un caso de estudio:

[Bases de datos: Diseño Físico](#) (Encalada, 2016b)

Realmente el proceso de diseño físico puede tener distintos matices. Dependiendo del tipo de proyecto, de la magnitud y criticidad de los datos, de la carga esperada, y del nivel de disponibilidad requerido, pueden ajustarse los pasos, o limitarse el alcance de cada uno de ellos.

La figura 4 ilustra un proceso de diseño físico simplificado, que puede aplicar a la mayoría de los proyectos.

Figura 4*Proceso simplificado del diseño físico*

Nota. Tomado de *Bases de Datos. Guía Didáctica* (p.101), por A. Encalada y J. Morocho, 2022, EdiLoja.

Como vemos, el artefacto principal resultante del diseño físico es el script para la implementación de la base de datos, el cual está escrito en el lenguaje de la base de datos elegida (SQL en bases de datos relacionales) y contiene toda la secuencia de comandos necesarios para materializar la base datos, con base en las especificaciones lógicas y físicas que se hayan establecido. La gran mayoría de las especificaciones que se definen en cada paso del proceso, se van incorporando al *script*.

En la tabla 1, se describe el alcance de cada paso. De ellos, los dos primeros los abordaremos en este apartado, y los siguientes en semanas posteriores.

Tabla 1*Descripción del proceso del diseño físico*

Paso	Descripción
1. Traducir el modelo lógico a lenguaje del SGBD	Implica traducir a lenguaje de base de datos, todas aquellas especificaciones definidas a nivel del diseño lógico. En el caso de las bases de datos relacionales implica traducir el diseño de tablas, con todas sus restricciones a lenguaje de definición de datos de SQL.

Paso	Descripción
2. Definir la organización física de la base de datos.	Conlleva definir como se organizarán las estructuras lógicas (tablas en bases de datos relacionales) en el almacenamiento secundario (disco) del servidor, de tal manera que se garantice espacio suficiente y acceso eficiente a los datos.
3. Definir mecanismos y restricciones de seguridad.	Además de las restricciones de integridad que ya vienen recogidas en el diseño conceptual y lógico de la base de datos, es necesario establecer otros mecanismos que ayuden a garantizar la seguridad de los datos. Por ejemplo: gestión de autenticación y autorización, implementación de vistas, entre otros.
4. Establecer otras especificaciones de implementación.	Otras especificaciones de implementación que se pueden considerar son, por ejemplo: la implementación de índices para mejorar el rendimiento de las consultas, configuraciones y restricciones necesarias a nivel del servidor para garantizar la disponibilidad del servicio.

Nota. Tomado de *Bases de Datos. Guía Didáctica* (p.102), por A. Encalada y J. Morocho, 2022, EdiLoja.

Traducir el modelo lógico a lenguaje de bases de datos

Lo primero que se hace en el diseño físico es generar el *script* inicial, que contiene la secuencia de comandos que permitirán implementar las estructuras y especificaciones definidas en el diseño lógico.

Como sabemos, en las bases de datos relacionales, el lenguaje que se utiliza para la generación del *script* de creación de la base de datos se denomina DDL (Data Definition Language), que es parte de SQL. El cual es bastante similar entre los distintos SGBD relacionales. En la asignatura de Fundamentos de Bases de Datos usted ya se introdujo en el manejo de este lenguaje, pero es necesario profundizar.

Pase a revisar el siguiente recurso con el que podrá recordar y reforzar el manejo y uso de DDL para implementar bases de datos en MySQL:



[Diseño físico de bases de datos](#)

Biblioteca Virtual UTPL - eLibro (Hueso Ibáñez, 2015), capítulo 3, pp. 91-105

Más allá de que para generar el *script* nos ayudemos de alguna herramienta especializada, como MySQL Workbench, Oracle SQLDeveloper Datamodeler,

PowerDesigner DataArchitect, u otra, es importante saber escribir y usar DDL, ya que en muchos casos es necesario introducir cambios en el script.

Revise el siguiente recurso donde podrá conocer las consideraciones que se deben tener en cuenta al adaptar el modelo lógico de datos al SGBD.



[Bases de datos: Traducir modelo lógico a SGBD](#)

REA [VIDEO] (Encalada, 2016a)

Para identificar cuáles restricciones de integridad se pueden implementar mediante DDL, considere que usualmente la mayoría de los SGBD relacionales solo permiten implementar las siguientes restricciones:

- Integridad de entidades (constraint PRIMARY KEY).
- Integridad referencial (constraint FOREIGN KEY).
- Claves alternativas (constraint UNIQUE).
- Campos obligatorios (constraint NOT NULL).
- Validación de dominio (constraint CHECK).

El resto de las restricciones normalmente las deberá controlar con Triggers o en la aplicación de usuario final.

Definir la organización física de la base de datos

A nivel físico, cada gestor de base de datos maneja su propio esquema y estructuras de almacenamiento, incluso si se tratan de motores del mismo tipo. Por ejemplo, la forma que Oracle, MySQL, SQL Server, DB2, PostgreSQL almacenan los datos en disco, difieren completamente, aunque todos son SGBD relacionales.

En este punto del diseño físico se debe establecer la manera en que se distribuirán los datos en el disco del servidor, en función de las prestaciones y facilidades del motor elegido. El objetivo es facilitar al motor el proceso de lecturas y escrituras (operaciones I/O), es decir, que las transferencias de bloques de datos entre almacenamiento primario (memoria RAM) y almacenamiento secundario (disco) sean lo más rápidas posibles. Por ejemplo: tablas que se acceden de manera concurrente, se las puede almacenar en ubicaciones distintas.



Revise el siguiente recurso donde se encuentra una interesante explicación de las estructuras de almacenamiento físico de una base de datos Oracle. Le servirá para comprender mejor la gran diferencia que existe entre el nivel lógico y el nivel físico de una base de datos:

[Almacenamiento y diseño físico en Oracle](#)

Biblioteca Virtual UTPL - eLibro (Cuadra et al., 2014), apartado 5.1, pp. 411- 416.

Entonces, en cuanto a diseño físico, lo principal que se debe definir es:

- **El espacio requerido en almacenamiento secundario:** con base en una proyección de crecimiento del volumen de datos de tres a cinco años.
- **La distribución física de los datos:** con base en un análisis de las tablas críticas, según el nivel de transaccionalidad y concurrencia esperado.



Actividades de aprendizaje recomendadas

Le invito a desarrollar las siguientes actividades para fortalecer el aprendizaje de lo estudiado en la presente semana:

1. Analice la correlación que existe entre el ciclo de vida de desarrollo de software (SDLC) y el ciclo de vida de bases de datos (DBLC).

Hágalo contraponiendo cada fase de manera que establezca las dependencias de uno con otro. Por ejemplo, en una línea de tiempo, indique lo que se debe ejecutar de uno y otro proceso, graficando con líneas dirigidas las dependencias existentes.

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.

2. A partir del diseño lógico propuesto para una base de datos orientada a la gestión del alquiler de bicicletas en una ciudad, realice la traducción a lenguaje DDL para MySQL. En el script DDL incluya especificaciones que permitan que las tablas físicamente se almacenen en un directorio distinto al predeterminado por MySQL.

El diseño lógico propuesto lo tiene disponible en el siguiente enlace:
[Diseño lógico caso de estudio alquiler de bicicletas.](#)

Lo puede hacer apoyado en la herramienta de modelado que ofrece el software MySQL Workbench, con la cual puede generar el *script DDL*. Pero teniendo presente que deberá saber leer el *script* para introducir los cambios necesarios de manera que incluyan todas las restricciones del caso y las especificaciones del problema.



Semana 2

1.3. Implementación y carga

Veremos ahora cómo se produce la materialización de la base de datos una vez definidas las especificaciones de implementación en el marco del diseño físico. Y también entenderemos a qué se refiere la carga de la base de datos, que formalmente siempre se realiza luego de la implementación.

Concluido el diseño físico se procede a implementar la base de datos. Ello implica:

- Instalar y configurar el SGBD.
- Crear y configurar la base de datos (esquema, estructuras físicas, tablas, índices, etc.) con todas las especificaciones del diseño físico.
- Configurar cuentas de usuario y permisos.
- Realizar la carga inicial de los datos.

Las particularidades de esta fase son muy dependientes del SGBD que se vaya a utilizar. Cada motor –aunque sean del mismo tipo– maneja sus propios esquemas de instalación, configuración, y optimización. En nuestro caso usaremos como servidor de base de datos [MySQL](#), y como herramienta de administración [MySQL Workbench](#).

Instalación y configuración del SGBD

El primer paso es preparar el entorno operativo sobre el cual se va a desplegar el servicio de la base de datos. Ello conlleva la preparación del servidor, la instalación y configuración del sistema operativo, y la instalación del SGBD seleccionado para materializar nuestra base de datos.



Revise el siguiente recurso con el que podrá realizar el proceso de instalación y configuración de un servidor de base de datos MySQL.

[Instalación y configuración de un SGBD](#)

Texto básico (Hueso Ibáñez, 2016), apartados 2.1 a 2.4.1, pp. 19 - 33.

Como ya lo aprendió en la semana anterior, a nivel físico cada Sistema Gestor de Base de Datos (SGBD) tiene sus propias características, y ello se extiende entre la instalación y configuración. Es necesario estudiar cada motor específico para comprender las prestaciones que tiene, y con ello determinar la configuración más adecuada en función de las prestaciones del equipo, de la carga proyectada, y del nivel de disponibilidad requerido.

Creación y configuración de la base de datos

Una vez instalado el motor, estamos listos para implementar la base de datos. Ello implica materializar todas aquellas estructuras de datos, restricciones, y especificaciones definidas en el diseño. Básicamente conlleva dos pasos:

- a. Crear el esquema de base de datos.
- b. Ejecutar el *script DDL*.



Para comprender el proceso de creación de una base de datos, busque y revise en YouTube el video titulado “[Como crear y configurar Base Datos en MySQL utilizando el Workbench y la consola de Windows \(CMD\)](#)”.

Una vez generada la base de datos, lo que sigue es cargar y ejecutar el *script DDL* que se generó en el marco del diseño físico. Con ello tendremos creadas todas las tablas con las especificaciones lógicas y físicas definidas, y con las restricciones de integridad establecidas. Y si en el diseño físico se estableció una organización física diferente a la predeterminada, se tendrá que primero configurar dichos espacios en cuanto al sistema operativo.

Carga inicial de datos

En toda implementación de base de datos se requiere realizar una carga inicial, que corresponde a la inserción de los datos con los que la base de datos entrará en producción. Normalmente, en la mayoría de los casos las bases de datos al entrar en operación no empiezan 100 % vacías, siempre existirán tablas en las cuales se necesita cargar una información inicial, normalmente en tablas de catálogos y/o tablas de parámetros. Incluso existen escenarios más complejos en los cuales se necesita migrar la información de una base de datos anterior, cuando, por ejemplo, se está implementando un nuevo sistema que reemplaza a uno antiguo.

Muchos autores la consideran a la *carga inicial* como otra fase dentro del ciclo de desarrollo de una base de datos, precisamente, por la complejidad que puede llegar a tener.

El proceso para realizar la carga inicial es similar al siguiente:

1. **Identificar datos de partida:** implica establecer cuál es la información que se necesita tener precargada en la base de datos para cuando el sistema entre en producción.
2. **Seleccionar fuentes:** se debe identificar de dónde se va a obtener los datos que se requieren cargar. Las fuentes pueden ser diversas: otros sistemas, *internet*, hojas de cálculo, documentos digitales, documentos impresos, etc.
3. **Preparar los datos a cargar:** corresponde a la extracción de los datos desde las fuentes, y su preparación para la carga. Se lo puede hacer con el uso de herramientas especializadas, hojas de cálculo, o con programación.
4. **Cargar los datos:** con los datos preprocesados, se procede a importar los datos en las tablas de la base de datos, mediante procesos de carga masiva.

En YouTube, revise el video titulado “[Base de Datos #MySQL](#)

| [Carga Masiva de Datos MySQL](#)”, en el cual se explica cómo realizar cargas masivas de datos en MySQL, usando el comando `load`.



Es muy importante que todo este proceso de carga quede automatizado, de manera que se pueda volver a ejecutar. Tome en cuenta que luego de la fase de implementación y carga, corresponde realizar pruebas a la base de datos, y para ello se parte de la base de datos inicializada, incorporando nueva información no real, como parte dicho control de calidad. Por lo que, una vez superadas las pruebas, y antes de entrar en operación, se deberá volver a inicializar a la base de datos, lo que implica volver a ejecutar el proceso de carga inicial.



Actividades de aprendizaje recomendadas

Le invito a desarrollar las siguientes actividades para fortalecer el aprendizaje de lo estudiado en la presente semana:

1. Intente hacer su propia implementación de MySQL server, personalizando el juego de caracteres, y el número de sesiones concurrentes permitidas.

Para ello, si ya tiene instalada y corriendo una instancia del motor MySQL, y no desea desinstalar, puede descargar una versión anterior de [MySQL](#) y ejecutar la instalación en un puerto distinto, por ejemplo 3307.

2. Para el caso de estudio “[Alquiler de bicicletas](#)”, implemente al base de datos en MySQL y realice una carga inicial con los datos disponibles en el siguiente enlace: [DatosAlquilerBicicletas](#).

Para ello debe usar el script DDL construido previamente, el cual lo debe ejecutar como superusuario (root). Una vez creada la base de datos, deberá realizar la carga de datos a cada tabla a partir del archivo Excel compartido.

Resultado de aprendizaje 3 y 4

- Explica el concepto de seguridad de base de datos.
- Dado un protocolo de seguridad, asegura una base de datos.

Con estos resultados de aprendizaje comprenderá lo que conlleva la gestión de la seguridad de los datos y los objetivos que se buscan alcanzar.

Además, le permitirán desarrollar la capacidad para definir la estructura de un plan de seguridad, con base en el análisis y aplicación de medidas específicas que se pueden implementar el contexto de la gestión de la confidencialidad y de la disponibilidad de una base de datos.

Contenidos, recursos y actividades de aprendizaje



Semana 3

1.4. Seguridad de la base de datos

Una de las mayores preocupaciones al implementar una base de datos es la seguridad de los datos. Que no se produzcan accesos no autorizados, que no ocurran ataques que afecten a la integridad de los datos, que no haya robo de información, que no existan inconvenientes con la operación continua del servicio de base de datos, que no se filtre información confidencial, etc., son las preocupaciones comunes. Y tanto el administrador de datos como el DBA son los llamados a establecer todos los mecanismos preventivos y correctivos que permitan proteger el contenido, garantizar la integridad, y asegurar la operación de la base de datos.

La seguridad de los datos debe analizarse principalmente desde tres perspectivas: la base de datos, las aplicaciones, y la infraestructura.

Dependiendo de las capacidades del SGBD, muchos controles para la seguridad de la información se pueden implementar en el motor (autenticación, autorización, restricciones de integridad, etc.); pero no es el único punto. Las aplicaciones de usuario final que acceden a la base de datos también son un punto crítico por el cual pueden producirse accesos indebidos que vulneren la base de datos, por lo que es muy importante que entre los aplicativos existan los debidos controles de autenticación y de autorización. Y así mismo, a nivel físico (servidores y red transmisión de datos) se deben implementar controles y restricciones que coadyuven a disminuir la probabilidad de que ocurra una vulneración a los datos o una afectación a la operación del servicio.

Objetivos de la seguridad de los datos.

La seguridad de los datos conlleva 3 objetivos principales: garantizar la **confidencialidad** de la información, preservar la **integridad** de los datos, y asegurar la **disponibilidad** del servicio de base de datos. Conocida como Tríada CID.



Revise los siguientes recursos en línea donde encontrará una explicación sobre los 3 objetivos que abarca la seguridad de una base de datos: confidencialidad, integridad, y disponibilidad.

- [¿Qué es la tríada de la CIA?](#)
- [Confidencialidad, integridad y disponibilidad \(tríada CIA\)](#)

No lo olvide, la confidencialidad, integridad y disponibilidad (también conocida como CIA) son tres conceptos clave en la seguridad de bases de datos.

- **Confidencialidad:** se refiere a garantizar que los datos almacenados en una base de datos solo están disponibles para aquellos que tienen autorización para verlos. Esto puede lograrse a través de mecanismos de autenticación y autorización, así como también mediante el uso de cifrado para proteger la transmisión y el almacenamiento de datos confidenciales.
- **Integridad:** se refiere a la exactitud y consistencia de los datos en una base de datos. La integridad de los datos debe ser protegida contra cualquier modificación no autorizada, ya sea accidental o intencional. Esto se puede lograr mediante la implementación de controles de acceso, la realización de copias de seguridad y la aplicación de reglas de integridad de los datos.
- **Disponibilidad:** se refiere a la capacidad de una base de datos para estar accesible y disponible cuando es necesario. Esto incluye la disponibilidad de la base de datos en sí, así como también la disponibilidad de las aplicaciones y servicios que utilizan los datos almacenados en la base de datos. La disponibilidad debe ser protegida contra fallos de *hardware* y *software*, así como también contra ataques de denegación de servicio.

En resumen, la confidencialidad, integridad y disponibilidad son tres aspectos cruciales de la seguridad de las bases de datos que deben ser considerados y protegidos para garantizar la privacidad, la exactitud y la accesibilidad de los datos importantes.

Amenazas y contramedidas

Las amenazas, también llamadas vulnerabilidades, son justamente aquellos eventos que puedan poner en riesgo o impedir el cumplimiento de tres objetivos mencionados anteriormente. Por ejemplo:

- Fallos de *hardware*.
- Fallos de *software*.
- Fallos en mecanismos de control de acceso físico.
- Fallos de comunicaciones (red).
- Fallos de energía.
- Falta de capacitación de usuarios finales.
- Suplantación de identidad (*Phishing*).
- Denegación de servicios (DoS).
- Implantación de *Malware*, etc.

Las contramedidas o medidas de seguridad son las acciones, políticas o mecanismos que evitan, neutralizan, o minimizan el riesgo de cualquier amenaza. Algunas de las medidas más comúnmente usadas se muestran en la tabla 2.

Tabla 2
Medidas de seguridad comúnmente usadas

Objetivo	Mecanismo de seguridad
Confidencialidad	<ul style="list-style-type: none">▪ Control de acceso físico a instalaciones.▪ Control de acceso lógico (autenticación y autorización), tanto en base de datos, como en la aplicación.▪ Empleo de mecanismos de cifrado de los datos.▪ Uso de vistas de datos.▪ Uso de <i>firewalls</i>.
Integridad	<ul style="list-style-type: none">▪ Control de acceso físico y lógico.▪ Implementación de restricciones de integridad (en la base de datos y en la aplicación).▪ Validación de datos.▪ Gestión de transacciones y control de concurrencia.▪ Copias de seguridad (respaldos).

Objetivo	Mecanismo de seguridad
Disponibilidad	<ul style="list-style-type: none"> ▪ Redundancia en almacenamiento secundario (RAID). ▪ Redundancia eléctrica. ▪ Servidores de respaldo. ▪ Protocolos de recuperación ante desastres. ▪ Segmentación de redes. ▪ Copias de seguridad (respaldos)

Nota. Adaptado de *Bases de Datos. Guía Didáctica*, por A. Encalada y J. Morocho, 2022, p.107, EdiLoja.

Complementariamente, es muy importante se definan políticas y protocolos de seguridad que regulen el acceso lógico y físico a los servidores y servicios de la base de datos.

1.5. Gestión de la confidencialidad

En el marco de la seguridad de una base de datos, la gestión de la confidencialidad es fundamental, puesto que como se ha dicho es esencial para garantizar la privacidad y protección de la información almacenada en la base de datos. Los datos solo deben estar disponibles para aquellos usuarios que tienen autorización para verlos.

Hay muchos mecanismos que ayudan a gestionar y asegurar la confidencialidad (ver tabla 2) y que deben ser considerados al diseñar e implementar una base de datos. Nos enfocaremos en dos de ellos en el presente apartado:

- **Control de acceso lógico:** que corresponde a la gestión de acceso de usuarios (autenticación y autorización), en este caso centrándonos en el lado del SGBD.
- **Uso de vistas de datos:** que permite aprovechar la posibilidad que crear vistas de datos para evitar un acceso directo a las tablas y a todos los datos almacenados en estas.

Gestión de acceso de usuarios

Permite implementar el control lógico de acceso a la base de datos.

Corresponde al **control de autenticación y autorización**. La autenticación corresponde a la validación de credenciales (*login* y *password*) del usuario, mientras que la autorización valida los permisos del usuario.

Revise el siguiente recurso que le permitirá comprender y configurar el sistema de acceso a los recursos de un servidor MySQL, mediante la implementación de mecanismos de autenticación y autorización:



Gestión de cuentas de usuario y permisos

Texto básico (Hueso Ibáñez, 2016), apartados 3.1 a 3.2.2, pp. 59 - 74.

Como vemos, en el marco de la gestión de la seguridad de una base de datos es posible establecer restricciones de acceso a los distintos recursos del servicio de base de datos, dependiendo del perfil de usuario que requiera conectarse al motor. Y debe quedar claro que, una cosa es la autenticación que valida las credenciales mediante usuario y contraseña, y otra cosa son las autorizaciones que tiene el usuario para acceder a los datos y recursos del servidor.

La gestión del acceso de usuarios implica:

- Gestión de usuarios.
- Gestión de privilegios.
- Gestión de roles.
- Gestión de perfiles.

La diferencia entre roles y perfiles depende de cada motor de base de datos. Normalmente, un rol corresponde a una agrupación de privilegios, mientras que un perfil corresponde a una configuración específica de cuotas de recursos y otras restricciones de acceso a la base de datos.

Es importante entonces que en el marco diseño físico e implementación de la base de datos, se incorporen e implementen especificaciones de autenticación y autorización que ayuden a asegurar la confidencialidad, y también la integridad de los datos.



Para aprender a crear usuarios y asignar permisos en MySQL, observe en YouTube el video “[Gestión de cuentas de usuario y permisos en MySQL](#)”.

No olvide que las cuentas de superusuario (root en MySQL) solo están reservadas para uso del DBA. Jamás se debe acceder con ese tipo de cuentas desde las aplicaciones de usuario final.

Vistas

Las vistas son mecanismos de seguridad muy útiles en la gestión de bases de datos, ya que permiten limitar el acceso a ciertos datos por parte de los usuarios y reducir la exposición de información confidencial.

Una vista es una tabla virtual que se crea a partir de una o varias tablas en la base de datos, y muestra solo los datos seleccionados por la consulta subyacente. Las vistas pueden definirse con diferentes niveles de permisos de acceso, lo que permite controlar qué datos pueden ver y manipular los usuarios.

A continuación, se presentan algunas formas en las que las vistas pueden mejorar la seguridad en una base de datos:

1. **Limitar el acceso a datos delicados:** las vistas pueden ser utilizadas para ocultar datos confidenciales de los usuarios que no tienen permiso para verlos. Por ejemplo, si una tabla contiene información personal delicada, se puede crear una vista que muestre solo los campos necesarios para los usuarios autorizados.
2. **Simplificar el acceso a los datos:** las vistas pueden ser usadas para reducir la cantidad de tablas y campos que los usuarios necesitan consultar para obtener la información que necesitan. Esto puede reducir la cantidad de errores que se cometen durante las consultas y aumentar la seguridad al evitar la exposición accidental de datos.
3. **Mejorar el rendimiento de las consultas:** las vistas pueden ser empleadas para optimizar el rendimiento de las consultas, ya que las consultas complejas pueden ser definidas en una vista y los usuarios solo necesitan consultar la vista en lugar de las tablas subyacentes.
4. **Controlar el acceso a los datos:** las vistas pueden ser utilizadas para controlar el acceso a los datos mediante la definición de permisos de acceso específicos para las vistas. Los usuarios que no tienen permiso para acceder a una vista no podrán ver los datos que contiene.

- 5. Evitar el acceso directo a las tablas:** las vistas se convierten en una barrera adicional ante un posible intento de acceder directamente a las tablas de la base de datos. El dar permiso para acceder a una vista no conlleva permisos de acceso directo a las tablas.

La creación de vistas en bases de datos no reviste complicación, se trata simplemente de crear un objeto que contiene la especificación de una consulta SQL.

Revise el siguiente recurso con el cual aprenderá a generar y gestionar vistas en MySQL.



Vistas

Texto básico (Hueso Ibáñez, 2016), apartado 4.5, pp. 124-126.

Como ha constatado, se trata simplemente de definir un objeto basado en un comando SELECT. La figura 5, muestra un ejemplo de creación de una vista. En este caso se está creando una vista de nombre `v_peliculas_es` que permitirá proyectar una porción de la información de películas en las que filtran solamente ciertas columnas y filas; solamente las columnas que constan en la cláusula SELECT, y solamente las filas correspondientes a películas en español.

Figura 5

Ejemplo comando DDL de creación de vistas

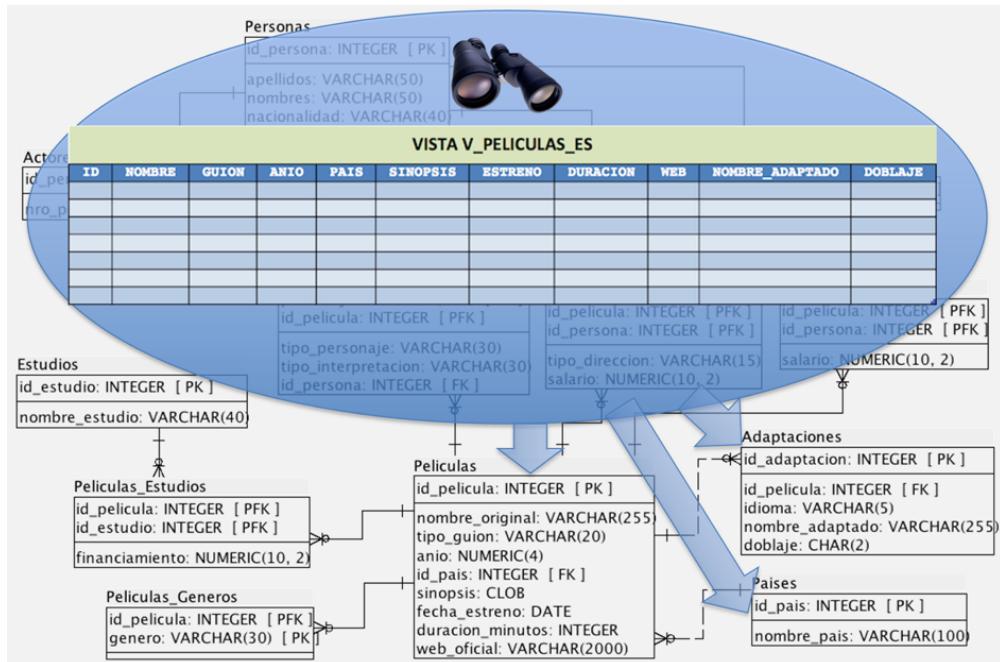
```
CREATE VIEW V_PELICULAS_ES
  (ID,NOMBRE,GUION,ANIO,PAIS,SINOPSIS,ESTRENO,DURACION,WEB,NOMBRE_ADAPTADO,
  DOBLAJE)
  FROM
  SELECT  P.ID_PELICULA,
          P.NOMBRE_ORIGINAL,
          P.TIPO_GUION,
          P.ANIO,
          X.NOMBRE_PAIS,
          P.SINOPSIS,
          P.FECHA_ESTRENO,
          P.DURACION_MINUTOS,
          P.WEB_OFICIAL,
          A.NOMBRE_ADAPTADO,
          A.DOBLAJE
  FROM    PELICULAS P
          LEFT OUTER JOIN ADAPTACIONES A ON P.ID_PELICULA = A.ID_PELICULA
          LEFT OUTER JOIN PAISES X ON P.ID_PAIS = X.ID_PAIS
  WHERE   A.IDIOMA = "ES";
```

Nota. Encalada, A., 2023.

Con ello, el usuario al que se le dé acceso a *v_peliculas_es* solo podrá ver esa porción de los datos cada vez que consulte la vista (figura 6).

Figura 6

Una vista desde la perspectiva del usuario



Nota. Encalada, A., 2023.



Actividades de aprendizaje recomendadas

Le invito a desarrollar las siguientes actividades para fortalecer el aprendizaje de lo estudiado en la presente semana:

1. Considere el caso de un sistema de información para un hospital, con el que se gestiona entre otros, las historias clínicas de los pacientes. Analice cuáles son las principales amenazas a las que podría estar expuesta la base de datos del hospital, y las medidas para contrarrestar esos riesgos.

Para su análisis, considere las vulnerabilidades a las que podría estar expuesto tanto la aplicación de usuario final como el servicio de la

base de datos. Considere también la criticidad de la información que se maneja y almacena en el hospital.

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.

2. Para su entorno MySQL, investigue como actualizar la contraseña de la cuenta del usuario root y restringir el acceso con esa cuenta solo a ciertas IPs.

Investigue como hacerlo tanto desde la herramienta gráfica, como desde la consola MySQL.

3. Para la base de datos del caso de estudio “[Alquiler de bicicletas](#)”, cuya implementación se realizó en semanas previas, defina un nuevo usuario (use mismo nombre de usuario de su cuenta UTPL), y asígnele permisos solo lectura para las tablas usuarios, bicicletas, y rastreadores ; y de lectura-escritura para el resto de las tablas.

Una vez asignados los permisos, verifíquelos accediendo con la nueva cuenta, e intentando acceder y modificar los datos de todas las tablas. Solo se debería permitir actualizar la tabla alquileres.

4. Es momento de efectuar una autoevaluación de sus aprendizajes para medir sus conocimientos respecto a lo estudiado en la primera parte de la unidad 1 (semanas 1 a 3), en torno a la administración de un SGBD, diseño físico, implementación y carga, y gestión de la seguridad de una base de datos. Responda al siguiente cuestionario.



Autoevaluación 1

Seleccione la opción correcta.

1. Suponga siguientes tablas:

```
FACTURA (nro_factura, cliente, telefono, direccion,  
fecha, subtotal, iva, total)

ITEM_FACTURA (iditem, nro_factura, cantidad, producto,  
costo_unit, costo_total)
```

¿Cuál de las siguientes restricciones no es posible implementar en una base de datos relacional a través de DDL, y, por lo tanto, requiere que se controle en las aplicaciones y/o con el uso de triggers?

- a. El subtotal, IVA, total de factura deben ser mayores a cero.
 - b. Una factura debe tener obligadamente al menos un ítem.
 - c. Cada registro en ítem_factura debe estar obligadamente asociado a una factura.
 - d. La llave primaria de factura es nro_factura.
2. La organización física de la base de datos se refiere a:
- a. La ubicación del servidor de base de datos.
 - b. La manera en que está organizada la información en tablas.
 - c. Los protocolos que se aplicarán para acceder físicamente al servidor de base de datos.
 - d. La forma en que se almacenarán los datos en el almacenamiento secundario o disco.
3. ¿Cuál de las siguientes amenazas a la seguridad de la base de datos, se podría contrarrestar efectuando respaldos sistemáticos y periódicos?
- a. Datos perdidos por fallos en disco.
 - b. Robo de contraseñas.
 - c. Información confidencial accesible por usuarios no autorizados.
 - d. Inyección SQL.

4. La información que se agrega durante la carga inicial de la base de datos corresponde a:
 - a. Datos que se requieren para realizar pruebas de carga y estrés.
 - b. Datos que agregan los usuarios cuando comienza la operación de la base de datos.
 - c. Datos iniciales con los que la base de datos entrará en producción.
 - d. La carga inicial se refiere creación de la base de datos, no implica carga de información.
5. Quien se encarga de levantar los requerimientos y construir el modelo conceptual de datos cumple el rol de:
 - a. Administrador de datos.
 - b. Administrador de base de datos o DBA.
 - c. Analista de base de datos.
6. En el ciclo de vida de una base de datos (DBLC), luego del diseño viene la Implementación y carga. ¿A qué nos referimos con carga?
 - a. Al ingreso de datos por parte del usuario final.
 - b. A las pruebas de sobrecarga que se deben hacer en toda base de datos.
 - c. A llenar las tablas con datos aleatorios para efectuar pruebas luego.
 - d. A cargar los datos iniciales y/o migrados desde sistemas anteriores.
7. Aquellas restricciones de integridad que no se puedan implementar en la base de datos mediante lenguaje de definición de datos (DDL) o mediante *triggers*, las debe controlar:
 - a. El DBA.
 - b. El usuario.
 - c. La aplicación.
 - d. No se requieren controlar.

8. Suponga que usted es un DBA a cargo de la administración de la base de datos de un hospital. ¿Qué medidas de seguridad serían más apropiadas recomendar en este caso ante la amenaza de "Entrada ilegal por parte de un hacker"?
 - a. Acceso a la base de datos solo desde servidor de aplicaciones / Tecnología RAID / Cifrado de datos / Implementación de un *firewall*.
 - b. Acceso a la base de datos solo desde servidor de aplicaciones / Cifrado de datos / Tecnología RAID.
 - c. Acceso a la base de datos solo desde servidor de aplicaciones / Uso de Vistas / Cifrado de datos / Implementación de un *firewall*.
 - d. Tecnología RAID / Uso de Vistas / Implementación de un *firewall* / Cifrado de datos.
9. El componente de seguridad que busca asegurar la validez y completitud de los datos almacenados en la base de datos se llama:
 - a. Seguridad de acceso.
 - b. Integridad.
 - c. Disponibilidad.
 - d. Confidencialidad.
10. Hablando de seguridad de los datos, la confidencialidad de la información se refiere a la necesidad de:
 - a. Mantener en secreto las claves de acceso a la base de datos.
 - b. Restringir el acceso a los datos solo a usuarios autorizados.
 - c. Impedir el acceso a los datos desde fuera de la organización.

11. ¿Cuáles dos de las siguientes afirmaciones son correctas respecto a las vistas de base de datos?
- a. Ayudan a limitar el acceso a datos confidenciales.
 - b. Se trata de otra estructura de almacenamiento que se usa en lugar de las tablas.
 - c. Se refiere a la forma en la que le aparecen los datos al usuario final en las aplicaciones.
 - d. Reducen la complejidad de las consultas y por ende los errores que se cometan desde las aplicaciones.
12. En el marco de control de acceso lógico hablamos de autenticación y autorización. La autorización corresponde a la validación de credenciales, mientras que la autenticación corresponde a la validación de permisos.
- a. Verdadero.
 - b. Falso.

[Ir al solucionario](#)



1.6. Gestión de la disponibilidad y rendimiento

Como se explicó en la semana anterior, uno de los aspectos más importantes que se deben garantizar al implementar una base de datos es la **disponibilidad**, que significa que el servicio de la base de datos debe operar dentro de parámetros de **continuidad y rendimiento** aceptables.

La garantía de disponibilidad de la base de datos es una responsabilidad que recae principalmente en el administrador de la base de datos (DBAs), con el apoyo y respaldo el equipo de TI y la organización en sí.

Gestión de la continuidad

Para garantizar la continuidad del servicio de la base de datos, es necesario seguir buenas prácticas de administración y gestión de la base de datos. A continuación, algunas estrategias y técnicas ayudan a garantizar la continuidad del servicio:

1. **Implementar redundancia:** la implementación de redundancia es clave para garantizar la continuidad del servicio. Por ejemplo, se pueden utilizar servidores espejo o réplicas de la base de datos para asegurarse de que si un servidor falla, el otro pueda continuar funcionando.
2. **Realizar copias de seguridad regularmente:** la realización de copias de seguridad regularmente es esencial para garantizar la disponibilidad de la base de datos. Las copias de seguridad permiten restaurar los datos en caso de que se produzca un fallo en el sistema.
3. **Monitorizar el rendimiento del sistema:** la monitorización del rendimiento del sistema puede ayudar a detectar problemas antes de que se conviertan en fallos críticos. Se pueden usar herramientas de monitorización para supervisar el uso de recursos y la actividad de la base de datos.
4. **Realizar pruebas de recuperación de desastres:** las pruebas de recuperación de desastres pueden ayudar a garantizar que el sistema

de base de datos pueda recuperarse en caso de un fallo crítico. Se deben hacer pruebas regularmente para asegurarse de que el sistema de recuperación de desastres funcione correctamente.

5. **Implementar políticas de seguridad adecuadas:** La implementación de políticas de seguridad adecuadas puede ayudar a garantizar la continuidad al proteger la base de datos de accesos no autorizados y asegurar la integridad de la información.

Estas son algunas estrategias que se pueden implementar para garantizar la continuidad del servicio de base de datos, siendo la redundancia una de las más importantes, sobre todo en implementaciones con requisitos de alta disponibilidad.

Gestión del rendimiento

En lo que respecta al **rendimiento** es fundamental que una vez que se haya concluido con la implementación y carga de la base de datos, se someta a esta a una serie de pruebas para evaluar si cumple con los parámetros esperados.

A raíz de las pruebas se podría necesitar realizar ajustes a la implementación de la base de datos, que es lo que se conoce como afinación del desempeño. Si el diseño de la base de datos fue bien hecho, los ajustes que se deban hacer serán mínimos. En cambio, si se hizo un mal diseño los ajustes podrían implicar incluso regresar al diseño conceptual, algo nada deseable. Por ello en este punto es relevante destacar la importancia que tiene el realizar un buen diseño, eso es crucial para garantizar el éxito de la implementación.

Ante un problema de rendimiento, se deben identificar las causas en un orden determinado, tomando como base los niveles de abstracción de los datos.

En el siguiente recurso encontrará una explicación que le permitirá comprender cómo es el proceso de afinación de un base de datos, en función del orden y prioridad en que se deben buscar las causas que originan el problema:

[Proceso de afinación de una base de datos \(Encalada, 2022d\)](#)

En bases de datos relacionales, los problemas de rendimiento en su mayor parte se detectan al realizar consultas SQL (comando SELECT), dado que es la operación más compleja y que representa una mayor carga de trabajo para el sistema gestor de la base de datos. Por ello muchas de las técnicas y mecanismos orientados a mejorar el rendimiento, tienen que ver justamente con la optimización de consultas, e incluyen entre otros, el manejo de índices, la introducción de redundancia (desnormalización), y las buenas prácticas SQL.

Índices

Uno de los mecanismos que ayudan a mejorar el rendimiento de las operaciones de consulta en una base de datos, es la implementación de índices. Un índice es una estructura ordenada asociada a un determinado atributo, que facilita la búsqueda y ordenación en ese atributo. Evita que cuando el motor necesite filtrar u ordenar por un campo indexado tenga que hacer un barrido secuencial de todas las filas de una tabla, optimizando con ello los tiempos de respuesta.

Para comprender mejor los índices y su implementación en MySQL le invito a revisar el siguiente recurso:



[Índices](#)

Texto básico (Hueso Ibáñez, 2016), apartado 5.1, pp. 134 - 141.

Aunque existan múltiples tipos de índices, el propósito final es facilitar al motor las tareas de búsqueda y ordenamiento de los datos. Y son necesarios, ya que, por defecto, una tabla no es una estructura ordenada, los registros no se guardan en un orden específico, se van agregando siempre al final en el orden de llegada.

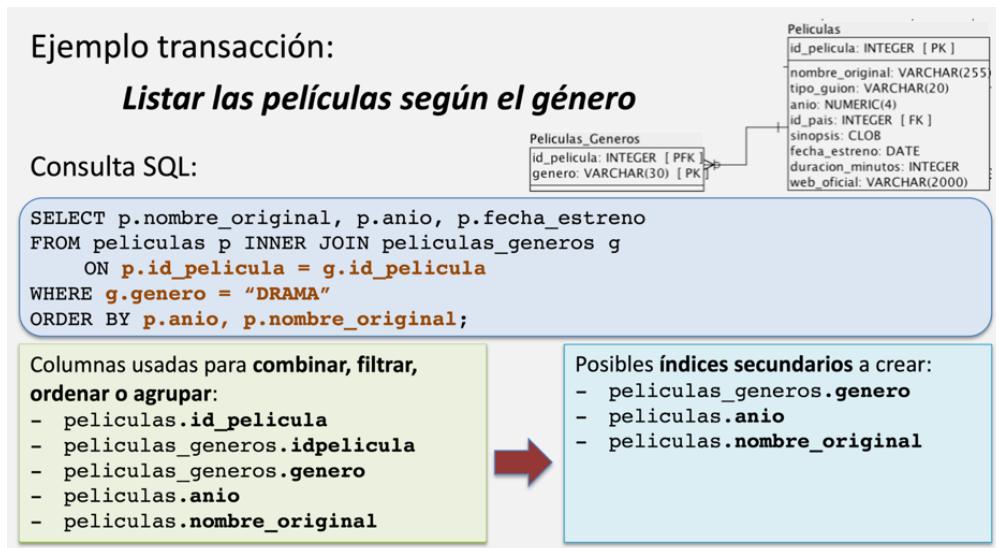
Los índices son útiles sobre todo en tablas que tienen una gran cantidad de registros. Y se los identifica con base en:

- Atributos que son más usados en operaciones de consulta para **filtrar, combinar, ordenar o agrupar** información.
- Llaves primarias (PK) siempre tienen un índice asociado (el SGBD lo crea automáticamente).
- Para llaves foráneas (FK) se recomienda siempre crear un índice (algunos SGBD también lo hacen de forma automática).

En la figura 7, se muestra un ejemplo de cómo se identifican las columnas a ser indexadas. Se toman como base consultas que se realizarían con frecuencia y que acceden a las tablas de mayor volumen, y a partir de ello se identifican las columnas que son utilizadas para combinar, filtrar, ordenar, y agrupar. De esas columnas se asume que las llaves primarias y llaves foráneas siempre estarán indexadas, y las restantes son las que se tendrían que indexar manualmente.

Figura 7

Ejemplo de identificación de claves candidatas a indexar



Nota. Tomado de *Índices* (p. 5), por Encalada, A., 2022.

Para conocer los escenarios en los que se debe considerar la creación de índices, revise el siguiente recurso, el cual los resume:



Creación de índices

Biblioteca Virtual UTPL - Ebook Central (Beynon-Davies, 2018), apartado 20.6, p. 344.

Es importante entonces saber elegir las columnas a indexar –aparte de llaves primarias y foráneas– considerando el hecho de que los índices ayudan a mejorar la velocidad de las consultas, pero también ralentizan las actualizaciones. En bases de datos transaccionales se debe reducir el uso

de índices al mínimo necesario, sobre todo en columnas de tablas que se actualizan con mucha frecuencia.

Desnormalización

La desnormalización es la acción de introducir redundancia a una base de datos, es el proceso contrario a la normalización. El objetivo es reducir los cálculos y/o la cantidad de tablas que requieren acceder en una consulta SQL. La inclusión de atributos derivados, la unificación de tablas, la duplicación de campos en distintas tablas, son ejemplos de introducción de redundancia.

Revise el siguiente recurso, que le permitirá conocer las principales técnicas de desnormalización:

[Desnormalización](#)

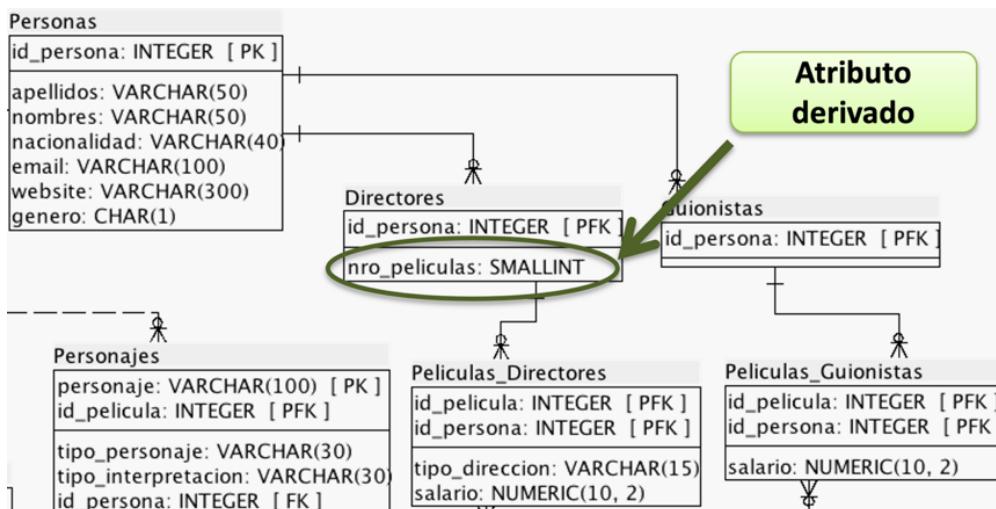


Biblioteca Virtual UTPL - *Ebook Central* (Beynon-Davies, 2018), apartado 20.7, pp. 345 - 346.

De las técnicas mencionadas, quizá una de las más usadas es la técnica de *datos derivados* o también llamada *inclusión de atributos derivados*. En la figura 8, se muestra un ejemplo de aplicación de esta técnica. En ese caso *nro_peliculas* es un campo que se considera como derivado o calculado, ya que si no existiera se podría calcular a partir del resto de la información de la base de datos, por lo tanto, es un campo redundante; pero que puede ser muy útil para acelerar la ejecución de ciertas consultas.

Figura 8

Ejemplo de inclusión de atributos derivados



Nota. Adaptado de *Técnicas de desnormalización* (p. 9), por Encalada, A., 2017.

En la figura 9, se muestra cómo sería la consulta sin atributos derivados y con la inclusión de atributos derivados. La diferencia es clara, en este caso se reduce el número de tablas a consultar, y se evita la necesidad del agrupamiento y agregación. Naturalmente, la segunda consulta representará menos carga para el SGBD y se ejecutará mucho más rápido.

Figura 9

Contraste de consultas con la inclusión de atributos derivados

Sin atributos derivados:

```
SELECT p.apellidos, p.nombres,
       COUNT(distinct f.id_pelicula) nro_peliculas
  FROM personas p INNER JOIN directores d
    ON p.id_persona = d.id_persona
   INNER JOIN peliculas_directores f
    ON d.id_persona = f.id_persona
 GROUP BY p.apellidos, p.nombres;
```

Con atributos derivados:

```
SELECT p.apellidos, p.nombres, d.nro_peliculas
  FROM personas p INNER JOIN directores d
    ON p.id_persona = d.id_persona;
```

Nota. Adaptado de *Técnicas de desnormalización* (p. 9), por Encalada, A., 2017.

Sin embargo, la desnormalización debe usarse con mucho cuidado, sobre todo en bases de datos transaccionales, ya que, en contraposición, puede conllevar riesgos de inconsistencia en los datos. En el ejemplo del atributo derivado, será necesario garantizar que ese campo siempre se encuentre sincronizado con la información de la cual se deriva, lo que implica además que las actualizaciones serán más demoradas.

Buenas prácticas SQL

Cuando el SGBD recibe una consulta SQL, esta pasa por un proceso de análisis, simplificación y optimización. Es decir, el motor antes de ejecutar la consulta trata de optimizarla de manera que se utilice el camino más corto para llegar al resultado.

Sin embargo, eso no significa que no debamos tener cuidado al momento de escribir una consulta. Uno de los mecanismos quizá más efectivos para lograr el mejor rendimiento del sistema de base de datos, es aplicar buenas prácticas al momento de escribir sentencias SQL.

 Revise el siguiente documento el cual contiene las recomendaciones más relevantes para tener en cuenta en la formulación de consultas SQL.

[Buenas prácticas SQL \(Encalada, 2017b\)](#)

Algunas de las recomendaciones más relevantes se resumen a continuación:

- Evitar el SELECT *. Traer únicamente los datos necesarios.
- Evitar productos cartesianos.
- Combinar en la cláusula FROM (JOIN)
- Ordenar solo si es necesario.
- Cualificar los nombres de columna en consultas multitabla.
- En subconsultas, si es posible, usar IN en lugar de NOT IN.
- Usar EXISTS en lugar de IN.
- Evitar DISTINCT si no es necesario.
- Paginar.
- Usar LIKE al final de las condiciones.



Actividades de aprendizaje recomendadas

Le invito a desarrollar las siguientes actividades para fortalecer el aprendizaje de lo estudiado en la presente semana:

1. Asuma que usted es el DBA en una institución bancaria; que acciones y mecanismos implementaría para asegurar la disponibilidad de la base de datos. Un banco exige una disponibilidad 24x7x365.

Considere sobre todo las acciones preventivas, teniendo presente el alcance 24x7x365 que significa 24 horas al día, 7 días a la semana, 365 días al año, es decir, disponibilidad máxima. Ello implica que, si el servidor falla, la reactivación debe ser casi inmediata.

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.

2. Para la base de datos del caso de estudio “[Alquiler de bicicletas](#)”, cuya implementación se realizó en semanas previas, implemente índices para tres columnas que no sean ni PK ni FK. Debe justificar el por qué de las columnas seleccionadas.

Para la selección de las columnas debe considerar aquellas que usted anticipa serán usadas con mucha frecuencia para operaciones de filtrado, ordenamiento, o agrupamiento.

3. Proponga un ejemplo donde sería útil aplicar la técnica de desnormalización “Duplicación de los datos en relaciones muchos-a-muchos”.

Para ello considere un esquema de tablas que implementen una relación M:N y una consulta SQL que acceda a dichas tablas.

Contraste la estructura de la consulta antes de desnormalizar, y luego de hacerlo. La consulta debería ser mucho más simple en el 2.º caso.

Resultado de aprendizaje 5

- Explica los diferentes protocolos de control de concurrencia.

Por medio de este resultado de aprendizaje conocerá como se realiza la gestión de la integridad de la base de datos, y especialmente como se

realiza el control de la concurrencia. Comprenderá lo que es una transacción en bases de datos, y el por qué es tan importante para garantizar la integridad de los datos. Y entenderá cómo el SGBD hace la gestión de transacciones, y los mecanismos que usa para asegurar la consistencia cuando muchas transacciones se ejecutan simultáneamente.

Contenidos, recursos y actividades de aprendizaje



Semana 5

1.7. Gestión de la integridad y control de concurrencia

La integridad de una base de datos se refiere a la exactitud, consistencia y fiabilidad de los datos almacenados en ella. La integridad de los datos es esencial para garantizar la confiabilidad de la información y para evitar errores y conflictos.

Hay varios mecanismos que pueden utilizarse para garantizar la integridad de una base de datos:

1. **Restricciones de integridad:** son reglas que se definen en la base de datos para garantizar que los datos cumplan con ciertos requisitos y se mantengan coherentes. Por ejemplo, puede haber una restricción que exige que todos los registros de una tabla tengan un valor único en una columna determinada.
2. **Transacciones:** una transacción es un grupo de operaciones que se realizan juntas como una sola operación. Si una transacción no se completa correctamente, se deshacen todos los cambios efectuados durante la transacción para garantizar la integridad de la base de datos.
3. **Validación de datos:** se pueden utilizar validaciones para garantizar que los datos introducidos en la base de datos cumplan con ciertos criterios, como un formato de fecha o un valor numérico dentro de un rango determinado.

4. **Copias de seguridad y recuperación:** es importante hacer copias de seguridad periódicas de la base de datos y tener un plan de recuperación en caso de fallos en el sistema. Esto ayuda a garantizar la integridad de los datos y a recuperarlos en caso de pérdida.

En resumen, la integridad de una base de datos es esencial para garantizar la calidad y la confiabilidad de la información almacenada. Es fundamental implementar medidas efectivas para proteger la integridad de la base de datos y asegurarse de que los datos estén siempre correctos y coherentes.

Gestión de transacciones y control de concurrencia

Uno de los conceptos fundamentales en bases de datos es la transacción, la cual permite agrupar varias operaciones en una sola unidad de ejecución; lo que ayuda a garantizar en gran medida la integridad de los datos. Todo SGBD debe asegurar la unicidad de una transacción, y además debe permitir que varias transacciones se puedan ejecutar de forma concurrente, dado que actualmente muchos usuarios acceden al mismo tiempo a la base de datos a través de las aplicaciones.

Es claro que la gestión de transacciones y el control de concurrencia no lo realiza el DBA, ni el programador, ni el usuario; sino el SGBD. Pero es necesario que usted conozca cómo se efectúa este proceso y su importancia.

En el siguiente recurso encontrará una explicación que le ayudará a comprender lo que es una transacción, sus propiedades, y los problemas que se generan a raíz de la concurrencia:



[Gestión de transacciones: Introducción](#)

Biblioteca Virtual UTPL - *Ebook Central* (Beynon-Davies, 2018), apartados 29.1 a 29.3, pp. 444 - 449.

Entonces, es claro que sin el manejo de transacciones sería muy difícil asegurar la integridad de la base de datos, sobre todo cuando se producen fallos en medio del registro de una serie de operaciones relacionadas. Y sin el control de concurrencia, igualmente se correría el riesgo de que se produzcan actualizaciones perdidas, datos no registrados o recuperaciones inconsistentes.

Métodos de bloqueo

El DBMS está obligado a garantizar las propiedades ACID de las transacciones, sea que se ejecuten de forma secuencial o de forma concurrente. Y la principal herramienta para asegurar dichas propiedades son los conocidos como mecanismos de bloqueo.

Revise el siguiente recurso de aprendizaje que le permitirá conocer cuáles son y cómo funcionan los principales mecanismos de bloqueo en el marco del control de concurrencia:



Gestión de transacciones: control de concurrencia

Biblioteca Virtual UTPL - Ebook Central (Beynon-Davies, 2018), apartado 29.4, pp. 449 - 455.

Como ha podido constatar, la gestión de transacciones y el control de concurrencia tiene como propósito garantizar la **seriabilidad** de la ejecución de transacciones, esto quiere decir que el DBMS debe planificar la ejecución concurrente de transacciones tal que el resultado final sea el mismo que se daría si las transacciones se ejecutaran de manera serial o secuencial (una a la vez). La figura 10, muestra un ejemplo de planificación serializable – también llamada secuenciable- para el caso de transacciones que realizan transferencias bancarias de la cuenta A a la cuenta B.

Figura 10

Ejemplo planificación serial vs planificación secuenciable equivalente

Planificación secuencial		Planificación secuenciable	
T ₁	T ₂	T ₁	T ₂
<code>read(A);</code>		<code>read(A);</code>	
<code>A := A-50;</code>		<code>A := A-50;</code>	
<code>write(A);</code>		<code>write(A);</code>	
<code>A = 1000</code>	<code>read(B);</code>	<code>A = 1000</code>	<code>A = 855</code>
<code>B = 2000</code>	<code>B := B+50;</code>	<code>B = 2000</code>	<code>B = 2145</code>
<code>write(B).</code>		<code>write(B).</code>	
<code>read(A);</code>		<code>read(B);</code>	
<code>temp := A*0.1;</code>		<code>B := B+50;</code>	
<code>A := A-temp;</code>		<code>write(B).</code>	
<code>write(A);</code>		<code>read(B);</code>	
<code>read(B);</code>		<code>B := B+temp;</code>	
<code>B := B+temp;</code>		<code>write(B).</code>	
<code>write(B).</code>			<i>estado consistente</i>

Nota. Encalada, A., 2023.

Definición de transacciones en SQL

En SQL estándar se especifica que una transacción SQL comienza automáticamente con la primera sentencia SQL ejecutada por un usuario o un programa. La transacción continúa con las sentencias subsiguientes hasta que finaliza a consecuencia de cualquiera de los siguientes escenarios:

- Una sentencia COMMIT finaliza la transacción con éxito, haciendo que los cambios a la base de datos sean permanentes. Una nueva transacción comienza inmediatamente después de la sentencia COMMIT.
- Una sentencia ROLLBACK aborta la transacción deshaciendo las modificaciones que haya efectuado a la base de datos. Una nueva transacción comienza inmediatamente después de la sentencia ROLLBACK.
- La terminación de un programa con éxito (para SQL programado) también finaliza la transacción correctamente, igual que si hubiera ejecutado una sentencia COMMIT.
- La terminación anormal del programa (para SQL programado), también aborta la transacción, del mismo modo que si hubiera ejecutado una sentencia ROLLBACK.

Revise el siguiente recurso con el que aprenderá sobre la manera en que se implementan transacciones en MySQL:

[Transacciones](#)



Biblioteca Virtual UTPL - eLibro (Hueso Ibáñez, 2015), apartado 5.5, pp. 152 - 155.

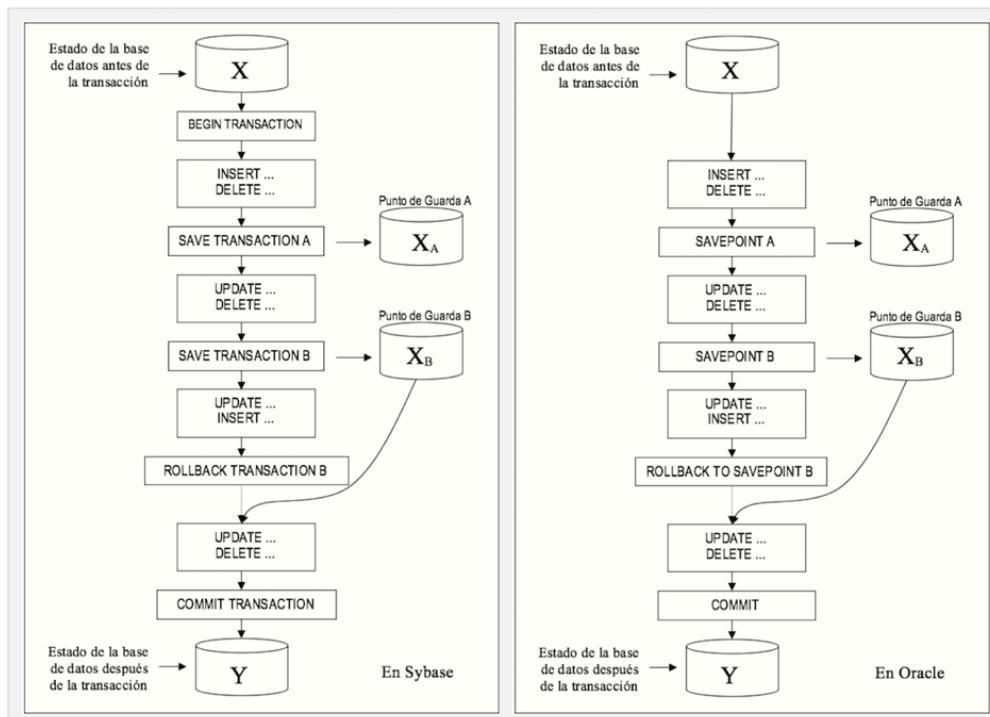
Para complementar lo dicho por (Hueso Ibáñez, 2015), algunos SGBD adaptan el modelo de transacciones de SQL estándar para proporcionar capacidades adicionales. La figura 11, nos muestra un ejemplo de transacciones de los SGBD Sybase y Oracle.

En el caso de Sybase se incluyen cuatro sentencias de procesamiento de transacciones:

- La sentencia BEGIN TRANSACTION señala el comienzo explícito de la transacción.
- La sentencia COMMIT TRANSACTION señala el final con éxito de una transacción.
- La sentencia SAVE TRANSACTION establece un punto de guarda a mitad de una transacción. Sybase guarda el estado de la base de datos en un punto actual de la transacción y le asigna al estado guardado un nombre de punto de guarda, especificado en la sentencia.
- La sentencia ROLLBACK TRANSACTION tiene dos papeles. Si se designa un punto de guarda en la sentencia ROLLBACK Sybase deshace los cambios de la base de datos efectuados desde el punto de guarda. Si no hay ningún punto de guarda designado la sentencia ROLLBACK deshace los cambios desde la sentencia BEGIN TRANSACTION.

Figura 11

El modelo de transacción de Sybase vs Oracle



Nota. Encalada, A., 2023.

El mecanismo de punto de guarda de Sybase es usado por muchos SGBD y es especialmente útil en transacciones complejas que contienen muchas sentencias. Oracle de hecho maneja conceptos parecidos, la principal diferencia es que Oracle se adapta más al estándar establecido (la transacción inicia automáticamente con la primera sentencia SQL).



Actividades de aprendizaje recomendadas

Le invito a desarrollar las siguientes actividades para fortalecer el aprendizaje de lo estudiado en la presente semana:

1. Para la base de datos MySQL del caso de estudio “[Alquiler de bicicletas](#)”, cuya implementación se realizó en semanas previas, realice una prueba del uso y aplicación de los comandos MySQL para el manejo de transacciones.

Para ello, pruebe hacer varias modificaciones (INSERT, UPDATE, DELETE) aleatorias sobre distintas tablas, enmarcadas en una transacción. Y utilice los comandos para confirmar (COMMIT) o deshacer (ROLLBACK) las transacciones, verificando que efectivamente, los cambios se registran permanentemente (COMMIT), o se deshacen (ROLLBAK).

2. Investigue acerca de los **Triggers** en bases de datos relacionales, y su utilidad para ayudar a garantizar la integridad de la base de datos. Plantee un caso específico donde sería útil implementarlos.

Para ello primero, estudie lo que es un trigger y cómo funciona su ejecución. Luego revise ejemplos de aplicación en diferentes contextos, y con base en ello plante su propio caso de aplicación. Los trigger pueden ser un gran complemento para implementar restricciones de integridad que no se pueden implementar con DDL.

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.

Resultado de aprendizaje 6, 7 y 8

- Explica el concepto de copia de seguridad y recuperación.
- Dado un protocolo de copia de seguridad, realiza una copia de seguridad de la base de datos.
- Recupera una base de datos.

Estos resultados de aprendizaje están orientados a que usted comprenda y sepa describir las características y tipos de copias de seguridad de bases de datos. Le permitirán analizar y diseñar la planificación que se debe efectuar para que el proceso de respaldo sea efectivo. Y le proveerán capacidad para respaldar una base de datos y para restaurarla posteriormente.

Contenidos, recursos y actividades de aprendizaje



Semana 6

1.8. Respaldo y recuperación

Una de las medidas de seguridad que se debe implementar en toda base de datos, es la obtención periódica de respaldos, también llamados copias de seguridad, es decir, obtener en almacenamiento terciario (cintas, discos externos, etc.) una copia de los datos almacenados en la base de datos, a efectos de garantizar la no pérdida de información en caso de que se produzca un fallo en el servidor, o en caso de que se produzca un desastre o se pierdan los datos por algún otro motivo.

En cualquier plan de contingencia orientado a garantizar la seguridad de los datos, la medida que nunca puede faltar es la obtención periódica de respaldos.

La tarea de “Respaldos y recuperación” es una las funciones que debe cumplir el DBA, y aquí también es importante aclarar que, aunque el respaldo se realice siguiendo todas las recomendaciones técnicas, siempre es necesario que periódicamente se realice un ensayo de recuperación

de la base de datos, a efectos que validar tanto los respaldos como el procedimiento de recuperación. Es la mejor manera de estar preparados y saber qué hacer ante una caída de la base de datos.

La obtención de respaldos debe realizarse siguiendo un protocolo que garantice la efectividad y buen resguardo de cada copia realizada.

Política de respaldo y recuperación de bases de datos

La política de respaldo y recuperación de bases de datos es un conjunto de reglas y procedimientos que se establecen para garantizar la disponibilidad de la información en una base de datos en caso de desastres naturales, fallos de *hardware* o errores humanos. Esta política es esencial para cualquier organización que dependa de los datos almacenados en sus sistemas para su operación diaria.

La política de respaldo y recuperación de bases de datos normalmente incluye los siguientes elementos:

1. Frecuencia de respaldo: se establece la frecuencia con la que se efectuarán los respaldos de la base de datos.
2. Método de respaldo: se determina el método de respaldo a utilizar, que puede ser una copia de seguridad completa, incremental o diferencial.
3. Ubicación de almacenamiento: se especifica el lugar donde se almacenarán los respaldos de la base de datos.
4. Verificación de respaldo: se establece un proceso para verificar que los respaldos se ejecutaron correctamente y que la información puede ser recuperada.
5. Plan de recuperación ante desastres: se define un plan detallado para la recuperación de la base de datos en caso de un desastre, incluyendo la identificación de los recursos necesarios y el tiempo estimado para la recuperación.

La política de respaldo y recuperación de bases de datos es un componente crítico de la gestión de la información y es esencial para garantizar la continuidad del negocio. Es importante actualizar y revisar periódicamente la política para asegurarse de que sigue siendo efectiva y adecuada para las necesidades actuales de la organización.

¿Cómo se obtiene y restaura un respaldo?

La copia de seguridad y recuperación de una base de datos depende del sistema de gestión de bases de datos (DBMS) que se esté utilizando. Aquí hay algunos pasos generales:

1. Planificar el respaldo: determinar qué información es relevante, cuánto tiempo se necesita para recuperarla y qué frecuencia se deben realizar copias de seguridad.
2. Realizar una copia de seguridad: se puede hacer una copia de la base de datos completa o solo de partes específicas, como tablas o transacciones.
3. Almacenamiento de copias de seguridad: las copias de seguridad deben almacenarse en un lugar seguro y fácil de recuperar, como un dispositivo externo o una nube.
4. Pruebas de recuperación: es fundamental probar periódicamente la recuperación de la copia de seguridad para asegurarse de que funciona correctamente en caso de una falla.
5. Recuperación de datos: en caso de una falla, se puede recuperar la base de datos usando la copia de seguridad más reciente.

Estos son los pasos generales, pero los detalles pueden variar según el DBMS utilizado.

Revise el siguiente recurso que le permitirá aprender cómo se efectúa el respaldo y recuperación en bases de datos MySQL.

Preste especial atención a los tipos de respaldos que existen y a las recomendaciones que se deben seguir para asegurar que los respaldos sean efectivos:



[Copias de seguridad de bases de datos](#)

Biblioteca Virtual UTPL - eLibro (Hueso Ibáñez, 2015), capítulo 9, pp. 243 - 258.

En sistemas con requerimientos de alta disponibilidad se necesita trabajar con motores que permitan realizar respaldos en caliente, es decir, que

las copias de seguridad se puedan obtener mientras el servicio está en operación, sin que ello conlleve riesgo de inconsistencia en los respaldos.

Otras recomendaciones para tener en cuenta son las siguientes:

- Los respaldos deben estar debidamente etiquetados.
- Debe haber múltiples réplicas del mismo respaldo almacenados en lugares diferentes.
- El acceso físico a los respaldos debe estar protegido.
- En muchos casos se contratan servicios de empresas aseguradoras.



Actividades de aprendizaje recomendadas

Le invito a desarrollar las siguientes actividades para fortalecer el aprendizaje de lo estudiado en la presente semana:

1. Para la base de datos MySQL del caso de estudio “[Alquiler de bicicletas](#)”, cuya implementación se efectuó en semanas previas, ejercite la obtención de respaldos y la restauración de los datos.

Para ello primero cree un segundo esquema de base de datos, que usará para restaurar el respaldo. Luego obtenga un respaldo lógico de la base de datos; y a continuación restaure el respaldo en el segundo esquema. Verifique que los datos se restauraron completamente. Hágalo primero de manera asistida utilizando la herramienta gráfica, y también después empleando comandos en consola MySQL. Repita la práctica realizando respaldos y restauraciones parciales.

2. Investigue y analice ejemplos de planes de recuperación ante desastres.

A partir de los ejemplos que hay seleccionados, identifique: la estructura del documento, los actores que intervienen y suscriben el documento, la composición del documento, los mecanismos preventivos y correctivos aplicados, y el protocolo de actuación establecido para cuando ocurra el fallo o desastre.

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.

Resultado de aprendizaje 9, 10 y 11

- Describe la arquitectura de base de datos cliente-servidor.
- Describe la arquitectura de base de datos de n-tier.
- Describe el concepto de servicios web y el papel de SOAP.

Estos resultados de aprendizaje favorecerán su comprensión acerca de las diferentes arquitecturas de acceso a bases de datos, las cuales con el paso de los años han ido evolucionando, desde un esquema de acceso centralizado hasta un esquema cliente-servidor multicapa. Podrá constatar que el esquema de conexión al SGBD es un factor que también incide en el rendimiento y seguridad de un sistema de información. Así mismo, descubrirá otros mecanismos de acceso a bases de datos a través de servicios web.

Contenidos, recursos y actividades de aprendizaje



Semana 7

1.9. Arquitecturas de acceso a bases de datos

La evolución de los SGBD se ha dado en muchos ámbitos: en los esquemas de almacenamiento, en los esquemas de acceso, en los esquemas de control de concurrencia, en los esquemas de recuperación, etc. Y uno de los ámbitos en los que ha evolucionado de manera significativa, es en el que tiene que ver con los esquemas de conexión, es decir en la manera que en que un usuario o sistema accede a la base de datos, e interactúa con esta a través del SGBD. A partir de ello han ido surgiendo distintas arquitecturas de conexión y acceso:

- Arquitectura centralizada.
- Arquitectura cliente / servidor.
- Arquitectura de n-capas.
- Arquitectura basada en servicios web.

Arquitectura centralizada

Inicialmente, las bases de datos corrían en grandes servidores denominados mainframes, donde todos los procesos y aplicaciones que permitían acceder a los datos corrían en ese computador central, ningún proceso relacionado con la extracción, procesamiento y/o visualización de los datos se realizaba fuera del servidor central. Las personas que accedían a consultar o registrar datos en la base de datos, lo hacían a través de terminales “tontos” cuya única función era desplegar datos en pantalla y capturar datos desde teclado.

Pase a revisar el siguiente recurso que le permitirá comprender la composición y características de la arquitectura centralizada.



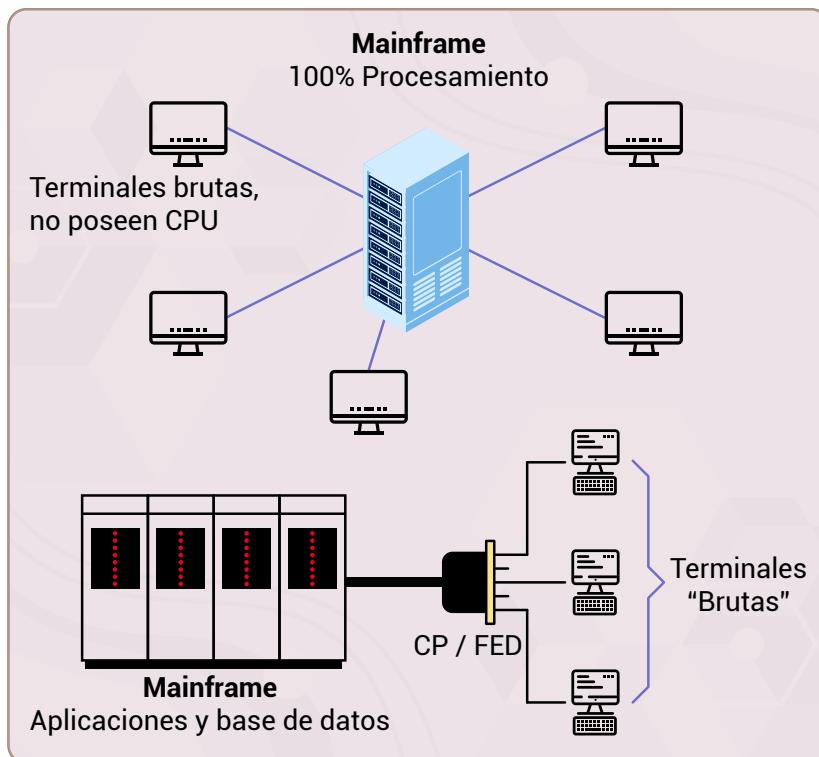
[Arquitecturas de bases de datos](#)

REA (Encalada, 2021), apartado 1, pp. 1 - 2.

Como ha podido apreciar, en la arquitectura centraliza todos los procesos, tanto de la base de datos, como de las aplicaciones de usuario final, corren en el servidor. Las terminales de los usuarios solo sirven para visualizar la interfaz. La figura 12, ilustra lo dicho.

Figura 12

Arquitectura centralizada



Nota. Adaptado de *Arquitectura Cliente Servidor* (p.3), por Campo, J., 2008.

Arquitectura cliente-servidor

Con el aparecimiento de las minicomputadoras y microcomputadoras (ordenadores personales) que tenían capacidad de procesamiento propio, y con la evolución de las redes informáticas, surgió la posibilidad de que no todos los procesos relacionados con el acceso a una base de datos se ejecuten en el **servidor**, sino que ciertas tareas se puedan realizar directamente en el equipo **cliente**, desde el que el usuario accede a la base de datos. En este escenario sucede entonces que el procesamiento de los datos ocurre en dos puntos/momentos:

- En el **servidor**: que es donde está la base datos y el SGBD, el cual a petición del cliente registra /extrae datos en/desde la base.
- En el **cliente**: se procesan y visualizan los datos a través de aplicaciones de usuario final.

El siguiente recurso explica con mayor detalle esta arquitectura.
Ponga atención a los ejemplos que allí se plantean.



Arquitecturas de bases de datos

REA (Encalada, 2021), apartado 2, pp. 3 - 6.

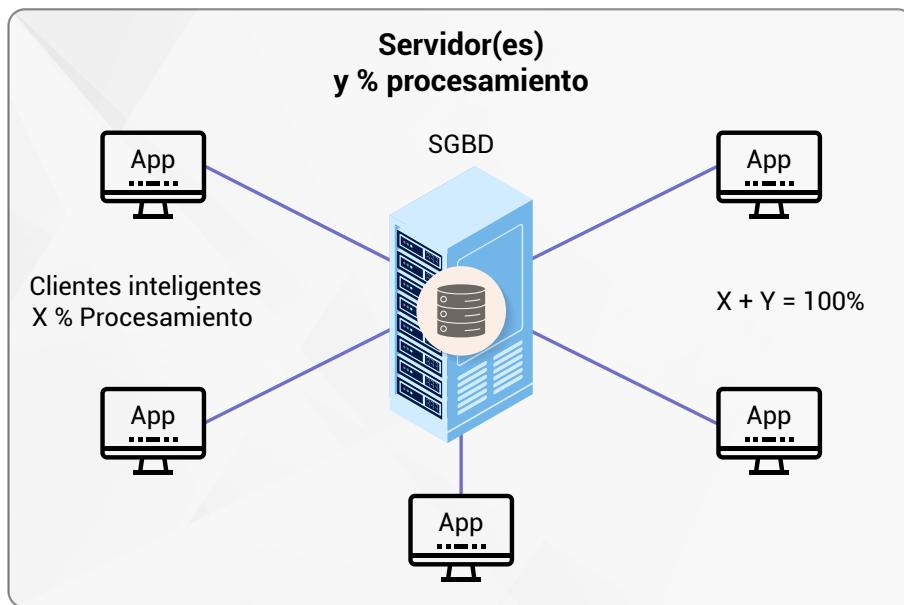
Vale aclarar que esta arquitectura surgió antes de la expansión de la *internet*, por lo que, este tipo de implementaciones las realizaban empresas que disponían de la infraestructura necesaria: servidores, y ordenadores o estaciones de trabajo, interconectados por una red informática.

Y para ser más precisos, a esta arquitectura se la conoce también como arquitectura cliente/servidor básica o arquitectura de 2 capas, en alusión a los puntos entre los que se distribuye el procesamiento de los datos (cliente y servidor).

A esta arquitectura también se puede explicar analizando los niveles de distribución de datos y procesos. Como explica Coronel et al. (2011), y como lo puede observar en la figura 13, en la arquitectura cliente/servidor básica, los datos y los procesos de la base de datos están en un solo sitio, mientras que los procesos de las aplicaciones de usuario final se ejecutan en múltiples sitios.

Figura 13

Arquitectura cliente-servidor



Nota. Adaptado de *Arquitectura Cliente Servidor* (p.10), por Campo, J., 2008.

Arquitectura de n-capas

Tome en cuenta que, siempre que ocurra que una parte de los procesos relacionados con el tratamiento de los datos se ejecutan fuera del servidor de base de datos, estaremos hablando de una arquitectura cliente/servidor, ya que en esos casos siempre vamos a tener equipos principales que proveen servicios —conocidos como servidores— y equipos cliente que se conectan a los servidores para hacer uso de esos servicios. Y por supuesto, cuando hablamos de sistemas de bases de datos, el servidor y servicio que nunca podrá estar ausente es de la base de datos, aquel que a través del SGBD provee acceso a los datos.

Entonces, cuando usted escuche arquitectura de tres o más capas, se está haciendo referencia a una arquitectura cliente/servidor, donde existe un cliente y varios servidores y/o servicios encadenados. Y donde normalmente el servicio de la base de datos se encuentra al final de la cadena de servidores.



Revise el siguiente recurso para que tenga más claridad de lo explicado. Note las distintas vistas, y terminología que se usa para nombrar a las diferentes capas de esta arquitectura:

Arquitecturas de bases de datos

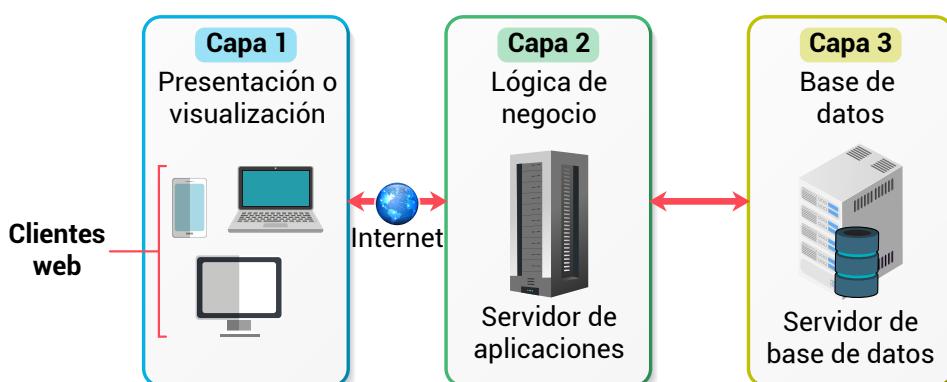
REA (Encalada, 2021), apartado 3, pp. 6 - 8.

Como habrá notado, en la arquitectura de tres capas entra en juego la web, quiere decir que esta arquitectura apareció con el surgimiento de *internet*, en el que la interfaz de usuario de la aplicación se despliega a través de un cliente web. Por ello se dice que esta arquitectura es para aplicaciones web.

Como vimos en la arquitectura de 2 capas, el procesamiento (lógica de negocio) y la visualización (interfaz de usuario) de los datos lo realiza la aplicación que corre en el lado del cliente. En cambio, en aplicaciones web con arquitectura de 3 capas, la lógica del negocio se implementa en un servidor llamado servidor de aplicaciones, y en el cliente solo se ejecuta una aplicación que actúa como interfaz de usuario y que se conecta a través de protocolos web al servidor de aplicaciones; la lógica del negocio que se ejecuta en dicho cliente es mínima. Y el servidor de aplicaciones, es el que a su vez establece comunicación con el servidor de base de datos que como dijimos siempre ocupa la última capa. Observe la figura 14.

Figura 14

Arquitectura de 3 capas



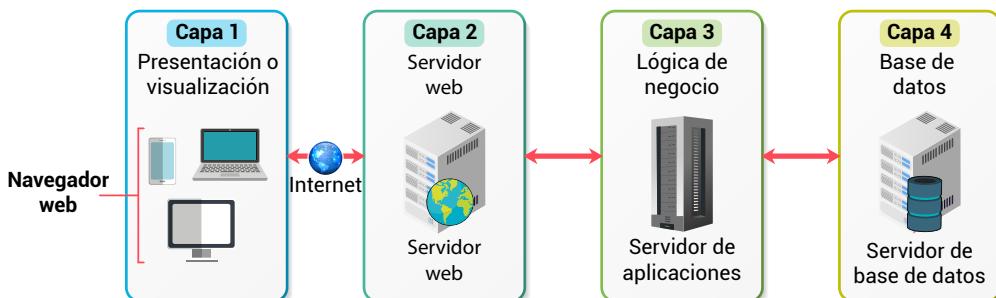
Nota. Tomado de *Administración de Bases de Datos. Guía Didáctica* (p.53), por Encalada, A., 2020, EdiLoja.

En cambio, las arquitecturas de n-capas o multicapa son las que tienen más de 3 capas, es decir, aquellas en las que como siempre existe una capa de

cliente, y al menos 3 capas de servidores o servicios, y donde igualmente la base de datos está en la capa final. Un ejemplo de esta arquitectura es el que se muestra en la figura 15, y se da igualmente en aplicaciones web cuando el cliente es un navegador web, por lo que también se requiere la implementación de un servidor http.

Figura 15

Arquitectura multicapa



Nota. Tomado de *Administración de Bases de Datos. Guía Didáctica* (p. 54), por Encalada, A., 2020, EdiLoja.

Arquitectura basada en servicios web

Cuando hablamos de servicios web nos referimos a aplicaciones que ofrecen funcionalidades a través de internet mediante el uso de protocolos estándar de la web, como HTTP y XML. Los servicios web se comunican entre sí y con aplicaciones cliente mediante mensajes que se intercambian a través de un canal de comunicación seguro.

La arquitectura de bases de datos basada en servicios es una forma de diseño de bases de datos que se enfoca en la creación de servicios de datos que se pueden reutilizar en diferentes aplicaciones. En lugar de diseñar una única base de datos que contenga toda la información necesaria para una aplicación específica, se crean servicios web que proporcionan acceso a datos específicos y se exponen a través de interfaces API. Estos servicios se pueden utilizar en múltiples aplicaciones y sistemas, lo que permite una mayor flexibilidad y escalabilidad. Dichos servicios en muchos casos se implementan usando SOAP (Simple Object Access Protocol).

SOAP es un protocolo de mensajería que permite a los servicios web intercambiar información en un formato estándar. Este protocolo define cómo deben estructurarse los mensajes para que puedan ser interpretados correctamente por los servicios web que los envían y reciben. SOAP es una

de las opciones disponibles para implementar servicios *web*, y se emplea comúnmente en aplicaciones empresariales que requieren una gestión segura y confiable de la información.

SOAP es un protocolo de mensajería que permite la interacción entre aplicaciones en diferentes lenguajes de programación y plataformas. En el contexto de las bases de datos, SOAP puede ser empleado para acceder a los servicios *web* que ofrecen funcionalidades relacionadas con la gestión de bases de datos, como la consulta y actualización de datos.



Actividades de aprendizaje recomendadas

Le invito a desarrollar las siguientes actividades para fortalecer el aprendizaje de lo estudiado en la presente semana:

1. Para el caso de la arquitectura de 3 capas, analice ¿qué implicación tiene esa capa intermedia (capa de lógica de negocio) respecto a la seguridad de los datos?

Para su análisis, tome en consideración que la capa de lógica de negocios es la que se comunica directamente con la base de datos.

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.

2. Esquematice un ejemplo de aplicación *web* con acceso a datos.

Consulte algún ejemplo de aplicación *web* con acceso a datos, con el que pueda identificar y describir con claridad y concreción su arquitectura. Puede buscar en repositorios de artículos o de tesis, trabajos realizados sobre desarrollo de aplicaciones *web*, donde sus autores expliquen la arquitectura de desarrollo de la aplicación implementada (capas, componentes y tecnologías utilizadas). Y con base en ello dibuje un esquema gráfico que ilustre el esquema de capas y tecnologías empleadas.

3. Es momento de efectuar una autoevaluación de sus aprendizajes para medir sus conocimientos respecto a lo estudiado en la segunda parte de la unidad 1 (semanas 4 a 7), en torno a la gestión de la integridad y concurrencia, respaldo y recuperación, y arquitecturas de acceso a bases de datos. Responda al siguiente cuestionario.



Autoevaluación 2

Seleccione la opción correcta.

1. ¿Cuál de las siguientes afirmaciones es cierta respecto al respaldo y recuperación de una base de datos?
 - a. Se deben realizar simulaciones de recuperación de la base de datos de forma periódica.
 - b. Los respaldos se deben sacar cuando existe presunción de fallo en el servidor de base de datos.
 - c. La obtención de los respaldos de los datos es una tarea que le compete a cada responsable de las aplicaciones que acceden a la base de datos.
2. En el procesamiento de transacciones, la propiedad del aislamiento podría verse comprometida sobre todo cuando el SGBD ejecuta a la vez.
 - a. Solo una transacción.
 - b. Varias transacciones que acceden a distintos elementos de datos.
 - c. Varias transacciones que acceden a un mismo elemento de datos.
3. En toda empresa, dentro de los planes de contingencia que se elaboran como parte de la administración de desastres, debe incluirse siempre:
 - a. Implementar un *datacenter* de respaldo en otro lugar distante.
 - b. Obtener respaldos periódicamente y realizar simulacros de recuperación.
 - c. Por cada base de datos tener dos servidores replicados, de manera que, si falla uno, entra a operar el otro.

4. Una transacción siempre debe:
 - a. Acceder a todos los registros de una tabla.
 - b. Realizar modificaciones en la base de datos.
 - c. Llevar a la base de datos de un estado consistente a otro.
5. En la ejecución concurrente de transacciones, una actualización perdida se produce cuando:
 - a. Una transacción T1 actualiza un dato X y luego otra transacción T2 sobrescribe ese valor sin considerar el valor actualizado por T1.
 - b. Una transacción T2 lee un dato que fue actualizado previamente por T1, y T1 retrocede (*rollback*) luego de que T2 finalizó.
 - c. Una transacción lee una serie de datos, mientras otra transacción los está actualizando.
6. En las arquitecturas cliente/servidor básicas, el cliente es una máquina que proporciona:
 - a. Capacidades de interfaz de usuario y procesamiento local.
 - b. Servicios que pueden ser consumidos por otros equipos.
 - c. Conectividad entre el navegador web y el servidor de aplicaciones.
7. ¿Cuál de las siguientes afirmaciones es correcta respecto a sistema de arquitectura centralizada?
 - a. Una parte del procesamiento de datos se realiza en los terminales de usuario final.
 - b. Suelen requerir contar con una computadora central de gran tamaño.
 - c. Se utilizan en la implementación de aplicaciones web.

8. En las arquitecturas de 3 o 4 capas para aplicaciones *web*, la mayor parte de la lógica de negocio se implementa en el:
 - a. Cliente.
 - b. Servidor *web*.
 - c. Servidor de aplicaciones.
 - d. Servidor de base de datos.
9. Una arquitectura multicapa no es cliente/servidor.
 - a. Verdadero.
 - b. Falso.
10. El principal responsable de asegurar la disponibilidad de una base de datos es:
 - a. El administrador de datos.
 - b. El administrador del *datacenter*.
 - c. El DBA.
 - d. El fabricante del SGBD.
 - e. El administrador de las aplicaciones.
11. En bases de datos los índices son importantes porque:
 - a. Ayudan a acelerar las búsquedas y ordenamiento en consultas SQL.
 - b. Ayudan a que la base de datos ocupe menos espacio en disco.
 - c. Evitan pérdida de información.
 - d. Ayudan que las operaciones de actualización (INSERT, UPDATE, DELETE) se ejecuten con mayor rapidez.
12. La desnormalización es una de las técnicas disponibles para optimizar el rendimiento de la base de datos, sin embargo, representa un riesgo, ya que puede provocar:
 - a. Consultas cada vez más lentas.
 - b. Desconexión frecuente a la base de datos.
 - c. Pérdida de la consistencia de los datos.
 - d. Problemas de durabilidad (los cambios a veces no se registran en la base de datos).

13. Una buena práctica SQL es:
- a. Ordenar siempre los resultados.
 - b. Evitar el uso innecesario del SELECT *.
 - c. Evitar usar alias para los nombres de columnas y/o tablas.
 - d. Las comparaciones tipo LIKE deben ir al inicio dentro de la cláusula WHERE.
14. Para garantizar la seriabilidad de las transacciones concurrentes se emplean métodos de bloqueo, uno de ellos es el “bloqueo exclusivo” el cual consiste en que si una transacción tiene un bloqueo exclusivo sobre un elemento de datos.
- a. Solamente podrá leer el elemento.
 - b. Podrá leer y actualizar el elemento.
 - c. No podrá acceder al elemento de datos.
15. En MySQL es posible realizar respaldos en caliente.
- a. Verdadero.
 - b. Falso.

[Ir al solucionario](#)



Semana 8



Actividades finales del bimestre

Hemos llegado al final del primer bimestre. En esta última semana sus esfuerzos deben estar enfocados en prepararse para rendir la primera evaluación bimestral. Le recomendamos hacerlo de manera ordenada y sistemática con base en las siguientes recomendaciones:

1. Repase los contenidos estudiados durante la unidad 1, en el orden que marca la ruta de aprendizaje. En su repaso incluya aquellos recursos y explicaciones de refuerzo que recibió en el marco de las tutorías semanales.
2. Complemente el repaso de los contenidos estudiados con la revisión de las actividades de aprendizaje desarrolladas (prácticas, foros, *chat*, autoevaluaciones, y cuestionarios).
3. Refuerce el repaso de aquellos apartados que usted considera le han resultado más difícil de asimilar. Tome nota de aquellas cuestiones en las que necesite una mayor retroalimentación.
4. Haga uso de los espacios de tutorías para plantear sus inquietudes a su tutor y recibir la retroalimentación necesaria.

El examen bimestral está estructurado con preguntas que tienen como objetivo evaluar su capacidad de comprensión, aplicación, y análisis en torno a los temas estudiados.

Esperamos que su experiencia de aprendizaje durante este primer bimestre haya sido positiva y enriquecedora.

¡Muchos éxitos en su evaluación!



Segundo bimestre

Resultado de aprendizaje 12

- Distingue entre las bases de datos homogéneos, heterogéneos y distribuidos federados.

Este resultado de aprendizaje le permitirá comprender lo que son las bases de datos distribuidas, y en ese marco tendrá la capacidad de identificar y caracterizar las diferentes clases que existen de este tipo de base de datos. Para ello, discutirá sobre los principales conceptos y principios que definen a las bases de datos distribuidas, y examinará la tipificación que existe en función de criterios relacionados con la heterogeneidad, la distribución, y la autonomía.

Contenidos, recursos y actividades de aprendizaje



Semana 9

¡Estimado estudiante, bienvenido a la segunda parte de nuestra asignatura!

Durante el segundo bimestre nos centraremos en estudiar las bases de datos distribuidas, y varios conceptos, usos, y aplicaciones avanzadas que actualmente tienen las bases de datos.

Unidad 2. Bases de datos distribuidas

2.1. Panorámica de las bases de datos distribuidas

Hasta ahora hemos venido asumiendo que el contenido de una base datos físicamente se almacena en un solo sitio (en el servidor que ejecuta el SGBD). Es lo que se conoce como base de datos centralizada.

No debe confundir base de datos centralizada con arquitectura de acceso centralizada. Recuerde que la arquitectura se refiere a todo el esquema de acceso, procesamiento y visualización de los datos. En cambio, cuando

hablamos de base de datos centralizada nos referimos específicamente a que los archivos de datos de la base residen en un solo servidor.

En contraposición a las bases de datos centralizadas, existen las bases de datos distribuidas, en las que, como es lógico suponer, los datos no están en un solo sitio.

Una base de datos distribuida es un tipo especial de implementación de base de datos, en la cual, aunque se conservan las mismas estructuras lógicas de almacenamiento de una base de datos centralizada, los datos físicamente no residen en un solo lugar; están dispersos en una serie de nodos que integran el sistema distribuido. Cuadra et al. (2014), la define como **“una colección de datos integrados lógicamente en una base de datos, pero físicamente pueden ser procesados y almacenados en varios nodos distribuidos sobre una red de ordenadores”**. (p.265).

Revise el siguiente recurso en el que se explica más en detalle lo que define y caracteriza a un sistema de bases de datos distribuidas. Ponga especial atención a los pro y contras de este tipo de bases de datos, y a los componentes de la arquitectura de un SGBD distribuido:



[Conceptos de bases de datos distribuidas](#)

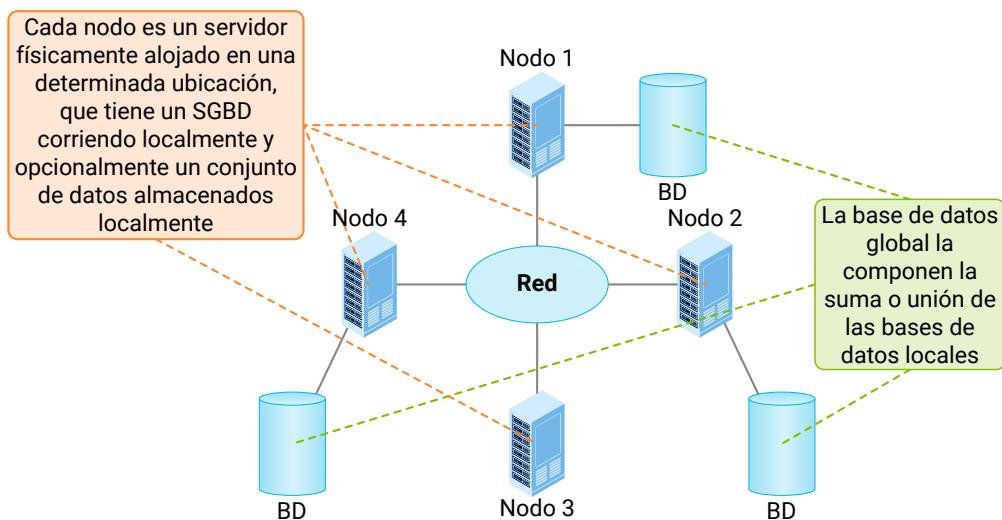
Texto básico (Hueso Ibáñez, 2016), apartado 6.1, pp. 192 - 195.

Entonces, un sistema de base de datos distribuida corresponde a una implementación del servicio de base de datos en el que intervienen al menos dos servidores (nodos), en los cuales se distribuyen tanto el procesamiento y cómo el almacenamiento de los datos. Y en este caso, al hablar de procesamiento, no se refiere a las aplicaciones de usuario final, sino al motor de base de datos. Es decir, en cada servidor corre una instancia del SGBD.

La figura 16, ilustra la topología de un SGBD distribuido, donde cada nodo es un servidor físicamente alojado en una determinada ubicación, con una instancia del SGBD corriendo localmente, y –opcionalmente– con un subconjunto de los datos en su almacenamiento secundario. Y todos los nodos estarían comunicados e integrados entre sí, pudiendo con ello acceder a los datos almacenados localmente y también a los datos almacenados en los otros servidores.

Figura 16

Topología de un sistema de base de datos distribuida



Nota. Encalada, A., 2023.

Elmasri y Navathe (2016), plantean las condiciones mínimas que debe cumplir una base de datos distribuida para ser considerada como tal:

- Conexión de nodos de bases de datos a través de una red informática. Hay múltiples ordenadores, llamados sitios o nodos. Estos sitios deben estar conectados por una red subyacente para transmitir datos y comandos entre los sitios.
- Interrelación lógica de las bases de datos conectadas. Es esencial que la información en los distintos nodos de la base de datos esté lógicamente relacionada.
- Posible ausencia de homogeneidad entre los nodos conectados. No es necesario que todos los nodos sean idénticos en términos de datos, hardware y software (p. 842).

Transparencia en un sistema de base de datos distribuida



Hay un principio fundamental de las bases de datos distribuidas que es el siguiente:

Desde el punto de vista del usuario, un sistema distribuido deberá ser idéntico a un sistema no distribuido.

Esto quiere decir que los usuarios que acceden a la base de datos distribuida deberán comportarse exactamente como si fuera una base de datos centralizada. El SGBD distribuido debe soportar las mismas funciones, el mismo lenguaje, garantizar las propiedades ACID de las transacciones, etc. Y en cuanto a los datos, el usuario no debe necesitar conocer donde están almacenados o como están distribuidos. Dicho de otro modo, en términos de SQL, la lógica de las operaciones de manipulación de datos SELECT, INSERT, UPDATE y DELETE no deberá sufrir cambios.

Para comprender mejor este tema, revise el siguiente recurso en el que se amplía la explicación del principio de transparencia que caracteriza a las bases de datos distribuidas:



[Fragmentación y transparencia](#)

Biblioteca Virtual UTPL - Ebook Central (Beynon-Davies, 2018), apartados 37.3 y 37.4, pp. 541 - 543.

Lo que Beynon-Davies (2018), define como *ubicación transparente*, *fragmentación transparente*, y *replicación transparente*, muchos autores lo unifican y llaman simplemente *transparencia de la distribución*. En la tabla 3, se describen los diferentes niveles de transparencia.

Tabla 3

Niveles de transparencia

Nivel de transparencia	Descripción
Transparencia de la distribución.	El acceso a los datos debe ser independiente de localización, de la fragmentación, y de la replicación. Es decir, para el usuario debe ser transparente la distribución física de los datos.
Transparencia de la transacción.	En un sistema distribuido, el procesamiento de transacciones y el control de concurrencia debe garantizar la consistencia de los datos y debe ser transparente, independiente de si se trata de una transacción local, o global. Local cuando solo se acceden los datos almacenados en el nodo accedido, y global cuando la transacción requiere acceder a datos que están en nodos remotos.
Transparencia de fallas.	En sistema de base de datos distribuida, si falla un nodo, el resto del sistema debe continuar operativo. Deberá continuar el procesamiento de transacciones, excepto aquellas que requieran acceso a los datos almacenados en el nodo caído. Mayor disponibilidad del sistema.

Nivel de transparencia	Descripción
Transparencia del desempeño.	Los tiempos de respuesta en sistema distribuido deben ser iguales o mejores a los de un sistema no distribuido. En promedio los tiempos deben mejorar, ya que la contención en el acceso a mismo nodo disminuye. Se balancea la carga.
Transparencia de la heterogeneidad.	El sistema de base de datos distribuida debe operar independientemente del <i>hardware</i> y <i>software</i> que se esté utilizado en cada nodo.

Nota. Encalada, A., 2023.

Tipos de sistemas de bases de datos distribuidas

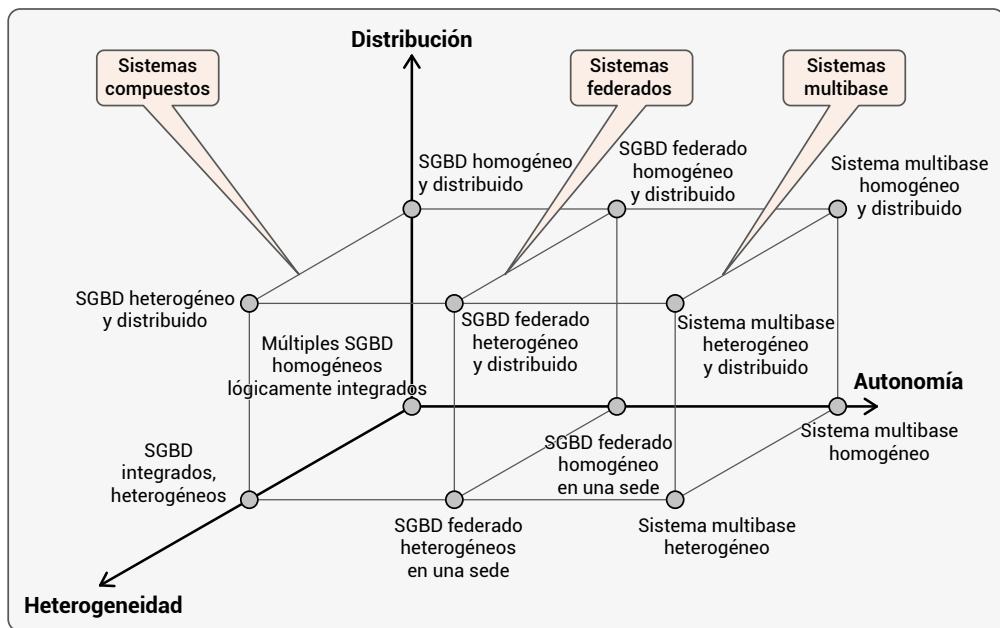
Como lo explica Hueso Ibáñez (2016), la arquitectura de un sistema de base de datos distribuida puede variar en función de 3 dimensiones:

- **Distribución:** si los datos reposan en una sola ubicación o en distintos sitios.
- **Heterogeneidad:** si los nodos tienen distintas configuraciones entre *hardware* y/o *software*.
- **Autonomía:** el grado de autonomía local que tiene cada nodo. Hay 3 niveles:
 - **Estrechamente integrados:** hay dependencia de un nodo central que balancea y distribuye la carga de trabajo.
 - **Semiautónomos:** cada nodo tiene autonomía local, pero a su vez está integrado al resto de nodos del sistema.
 - **Totalmente autónomos:** cada nodo es totalmente independiente y aislado.

Y en función del grado de distribución, heterogeneidad y autonomía podemos tener múltiples tipos de SGBD distribuidos. En la figura 17, Cuadra et al. (2014), ilustra las tres dimensiones, y las clases de SGBD distribuidos según el grado de cada dimensión.

Figura 17

Clases de SGBD distribuidos



Nota. Tomado de *Desarrollo de Bases de Datos. Casos Prácticos desde el Análisis a la Implementación* (p.101), por Cuadra, D., et al., 2014, Ra-Ma.

De ellos, hay tres tipos que merecen ser destacados: los sistemas homogéneos, heterogéneos, y federados.

Con el estudio del siguiente recurso de aprendizaje comprenderá el alcance de los tres tipos principales de bases de datos distribuidas:



Tipos de sistemas de bases de datos distribuidas

Biblioteca Virtual UTPL - Ebook Central (Beynon-Davies, 2018), apartado 37.7, pp.545 - 547.

Para complementar lo explicado por Beynon-Davies (2018), cuando hablamos de sistemas homogéneos normalmente se trata nodos que tiene la misma configuración a nivel de SGBD, sistema operativo, e incluso hardware. En cambio, en sistemas heterogéneos la variación se da sobre todo entre sistema operativo y/o SGBD. En los sistemas federados se entiende que hablamos de nodos que pueden ser homogéneos o heterogéneos, pero que tienen la característica de ser semiautónomos.



Actividades de aprendizaje recomendadas

Le invito a desarrollar las siguientes actividades para fortalecer el aprendizaje de lo estudiado en la presente semana:

1. Investigue acerca de las 12 reglas que Christopher J. Date propuso en el marco de las bases de datos distribuidas, y que las debe cumplir todo SGBD distribuido. Relacione dichas reglas con los niveles de transparencia aquí explicados.

Tome en consideración que dichas reglas fueron propuestas con base en la topología original de los sistemas de bases de datos distribuidas, que corresponde a un sistema federado donde los nodos son semiautónomos.

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.

2. Uno de los niveles de transparencia en bases de datos distribuidas corresponde a la *Transparencia de la transacción*, que implica que el SGBD distribuido debe garantizar la ejecución consistente de transacciones, independientemente de si la transacción es local o global (distribuida). Investigue y analice cuál es la diferencia del procesamiento de transacciones que se realiza en un sistema centralizado vs. en un sistema distribuido.

Considere que en una transacción distribuida intervienen varios nodos, y a cada uno le corresponde registrar y confirmar los cambios que se efectúen en el nodo que administran. La transacción global solo podrá confirmarse si todos los nodos han confirmado los cambios locales.

Resultado de aprendizaje 13 y 14

- Explica el concepto de repetición en lo que respecta a las bases de datos distribuidas.
- Distingue entre la reflexión horizontal y vertical en lo que respecta a las bases de datos distribuidas.

Por medio de estos resultados de aprendizaje conocerá los patrones de distribución de datos que se aplican al diseñar una base de datos distribuida, comprenderá cómo se reparten los datos en los diferentes

nodos de un sistema distribuido, y con base en ello, sabrá diferenciar entre lo que es replicación, y lo que es fragmentación —horizontal o vertical— de los datos.

Contenidos, recursos y actividades de aprendizaje



Semana 10

2.2. Patrones de distribución de datos

Con lo aprendido hasta ahora, sabemos que un sistema de base de datos distribuida se estructura de un conjunto de nodos ubicados en distintos lugares, donde cada nodo ejecuta un SGBD local y almacena una porción de los datos de la base de datos. Y donde, además, si se trata de un sistema distribuido integrado (compuesto o federado) existirá un SGBD global encargado de orquestar la operación de todo el conjunto de nodos.

A partir de lo anterior, corresponde ahora preguntarnos: ¿cómo se distribuyen los datos?, ¿qué criterios se deben considerar para determinar los datos que se deben almacenar en cada nodo?, ¿pueden repetirse los datos almacenados en distintos nodos?

Fragmentación y replicación

Las preguntas anteriores se responden estudiando los dos esquemas de distribución que pueden aplicarse en este tipo de sistemas: fragmentación y replicación. La fragmentación divide una tabla en varias partes y cada parte se almacena en un nodo del sistema distribuido. La replicación, en cambio, guarda una copia íntegra de la tabla en varios nodos del sistema distribuido. Pudiéndose aplicar ambos esquemas de manera combinada.

Pase a revisar el siguiente recurso con el que usted comprenderá cómo se aplican ambos patrones de distribución (fragmentación y replicación) en el contexto de una base de datos distribuida.



Patrones de distribución

REA (Encalada, 2020a)

Sobre la conveniencia de fragmentar una tabla, hay algunas razones para hacerlo que conviene destacar y que las debe tener presente:

- Es útil porque las aplicaciones de bases de datos distribuidas normalmente funcionan con vistas, por lo tanto, se puede emplear distintas relaciones en distintos nodos para formar la unidad distribuida.
- Permite obtener mayor eficiencia porque los datos se almacenen cerca del nodo donde se utilizan con mayor frecuencia.
- Permite aumentar el grado de concurrencia porque la fragmentación de las relaciones permite que una transacción pueda dividirse en subconsultas que operan sobre estos fragmentos.
- Proporciona más seguridad porque los datos no requeridos por un nodo local no se almacenen en él, por tanto, no están disponibles para personas no autorizadas. (Cuadra et al., 2014, p. 274).

Así mismo, tome en consideración las siguientes tres normas que determinan la calidad de la fragmentación de una relación:

- **Completitud.** Todos los datos de una relación fragmentada han de encontrarse en, al menos, un fragmento.
- **Disyunción.** Los fragmentos deberán ser disjuntos, es decir, los datos que aparecen en un fragmento no deben aparecer en otro, excepto las claves primarias donde sí pueden aparecer más de una vez como en el caso de la fragmentación vertical.
- **Reconstrucción.** Siempre se ha de poder reconstruir la base de datos global a partir de los fragmentos. (Cuadra et al., 2014, p. 274).

Diseño de una base de datos distribuida

La **fragmentación** y la **replicación** constituyen los patrones base que se usan en el diseño físico de una base de datos distribuida, sin embargo, se deben tener en cuenta algunos factores en dicho proceso.



En el siguiente recurso encontrará una explicación del proceso a seguir y los aspectos a considerar en el diseño de una base de datos distribuida. Ponga especial atención al caso práctico:

Diseño de bases de datos distribuidas

Biblioteca Virtual UTPL - Ebook Central (Beynon-Davies, 2018), apartados 37.9 y 37.10, pp. 548 - 549.

Como bien explica Beynon-Davies (2018) el diseño conceptual y lógico de una base de datos distribuida es similar al de una base de datos centralizada, porque como ya sabemos de cara al usuario final el esquema de datos es global y la distribución de los datos en distintos nodos, es transparente. Lo que varían son las especificaciones físicas de distribución de los datos almacenados en cada tabla, con base en los patrones vistos anteriormente.



Actividades de aprendizaje recomendadas

Le invito a desarrollar las siguientes actividades para fortalecer el aprendizaje de lo estudiado en la presente semana:

1. Suponga las tablas de una base de datos de una empresa de servicios:

clientes (cedula, apellidos, nombres, telefono, direccion, sucursal, tipo_cliente, id_nacionalidad).

sucursales (sucursal, nombre, provincia, ciudad, dirección, telefono).

tipos_cliente (tipo_cliente, descripcion, descuento).

nacionalidades (id_nacionalidad, nombre).

vehiculos (id_vehiculo, placa, chasis, marca, modelo, año, color, sucursal).

Se decide implementar un sistema SGBD distribuido y se establece que se hará una distribución geográfica de los datos, basados en la provincia. Es decir, se tiene un nodo en cada provincia donde opere la empresa. Con base en lo anterior, analice y explique: en este caso ¿se debería aplicar fragmentación, replicación, ambas?, ¿Cuáles tablas se deberían fragmentar y cuáles replicar?

Para el análisis parte identificando aquella tabla en la que se pueda aplicar fragmentación horizontal primaria, y partir de ello identifique aquellas tablas relacionadas en las que se pueda aplicar fragmentación derivada. Y para aquellas tablas transversales lo recomendable será replicar.

2. Ensaye un esquema gráfico de cómo se vería la topología de la base de datos distribuida analizada en la actividad anterior, asumiendo que la empresa de servicios opera las provincias de Pichincha, Guayas, y Loja. En Pichincha con dos 3 oficinas, en Guayas con 2 oficinas, y en Loja con 1 oficina.

Para ello, sería recomendable que dibuje el esquema sobre el mapa del Ecuador, haciendo constar nodos en la ubicación correspondiente, e ilustrando lo que contendría almacenado cada nodo (fragmentos o réplicas de cada tabla), la comunicación entre los nodos, y el acceso desde las terminales de trabajo de cada oficina.

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.

Resultado de aprendizaje 15 y 16

- Analiza la conveniencia de una recomendación para replicar (o no replicar) una base de datos distribuida en una situación dada.
- Analiza la conveniencia de una recomendación de particionamiento de una base de datos distribuida en una situación dada.

Con estos resultados de aprendizaje usted podrá examinar las aplicaciones que tienen las bases de datos distribuidas y los diferentes esquemas de distribución. Podrá conocer distintos escenarios en los cuales actualmente se aplican los conceptos y principios que rigen a este tipo de bases de datos. Y mediante el análisis de escenarios específicos tendrá la capacidad de establecer cuando es conveniente replicar o particionar una base de datos.

Contenidos, recursos y actividades de aprendizaje



2.3. Aplicaciones de las bases de datos distribuidas

Actualmente, las bases de datos distribuidas se utilizan ampliamente en una variedad de aplicaciones y escenarios, desde empresas hasta instituciones gubernamentales, universidades, institutos de investigación, y otras organizaciones. Pero antes de analizar los posibles usos de esta arquitectura de bases de datos, es importante primero que usted conozca los pros y contras que tiene una implementación de este tipo.

Ventajas y desventajas de las bases de datos distribuidas

Como en toda solución informática, los sistemas de bases de datos distribuidas tienen sus pros y contras que usted debe tener presente a efectos de determinar el escenario adecuado en el que conviene implementar este tipo de bases de datos.

La principal ventaja de este tipo de implementaciones radica en la mejora de la disponibilidad, ya que la carga del servicio se distribuye entre varios servidores y con ello mejoran los tiempos de respuesta, y también disminuye la probabilidad de caídas del sistema ante fallos de *hardware*. Pero existen otras ventajas que se detallan en la tabla 4.

Tabla 4

Razones que justifican la implementación de una base de datos distribuida

Ventaja	Descripción
Refleja la estructura organizativa	Generalmente, las organizaciones están distribuidas de forma natural en diversas ubicaciones y en cada ubicación utilizan su base de datos, por lo que reflejan la estructura de la compañía, y es factible que desde las sucursales se hagan consultas locales a los datos, pero también desde las oficinas centrales accedan a información global de todas las ubicaciones.
Mejora la compartición de los datos y la autonomía local	Los usuarios de una ubicación pueden ser capaces de acceder a los datos que residen en otras ubicaciones. Aun así, se diferencia entre un administrador global que es responsable del sistema completo y el administrador local que administra el SGBD local.

Ventaja	Descripción
Mayor disponibilidad	En un sistema centralizado una falla en el servidor de base de datos hace que se detengan las operaciones del SGBD. Pero en un sistema distribuido una falla de un nodo no paraliza todo el sistema, sino que seguirá funcionando y las tareas asignadas al nodo fallido son redirigidas hacia otro nodo.
Mayor fiabilidad	Debido a que los datos se pueden replicar podrán estar disponibles en más de un nodo por lo que un fallo en un nodo o en el enlace de comunicaciones no hace que los datos dejen de estar disponibles.
Mayores prestaciones	Debido a que los datos están localizados junto al nodo de mayor demanda los tiempos de acceso a estos datos son más rápidos que en una base de datos centralizada. Además, ya que cada nodo gestiona una parte de los datos existe menor contienda por los recursos (procesador, E/S de disco) que en una base de datos centralizada.
Economía	En este punto se considera que el mantener una red de pequeñas computadoras que iguala en procesamiento a un <i>mainframe</i> , por ejemplo, resulta mucho más económico y fácil de agregar más equipos a la red. Así mismo la distribución de los datos resulta mucho más económica ejecutando un procesamiento local en lugar de los altos costos que representa la transmisión de datos.
Crecimiento modular	Ya que es mucho más fácil añadir nodos a la red sin afectar el funcionamiento del sistema distribuido y expandir los recursos para lograr mayores capacidades de procesamiento y almacenamiento, a diferencia de una base de datos centralizada en donde se debe repotenciar o reemplazar un servidor con los consiguientes costos que representa.
Integración	Es una gran ventaja de los SGBD distribuidos, pues muchas organizaciones tienen que continuar con sus antiguas aplicaciones y para que estas puedan trabajar con las nuevas y más modernas deben integrarse y de esta forma están haciendo un procesamiento distribuido.
Capacidad de competir	Muchas organizaciones han tenido que reorganizar sus operaciones y utilizar tecnología de bases de datos distribuidas para seguir siendo competitivas.

Nota. Adaptado de *Administración de Base de Datos. Guía didáctica*, por J. Morocho et al., 2020, pp.88-90, EdiLoja.

Revise el siguiente recurso que complementa la explicación de las ventajas de las bases de datos distribuidas. Ponga atención al ejemplo que plantea el autor:



[Ventajas de los datos distribuidos](#)

Biblioteca Virtual UTPL - *Ebook Central* (Beynon-Davies, 2018), apartados 37.6, pp. 544 - 545.

En contraposición, el principal inconveniente de los sistemas distribuidos de bases de datos es la complejidad añadida que es necesaria para garantizar la coordinación apropiada entre los nodos o emplazamientos. Esta creciente complejidad tiene varias facetas que se describen en la tabla 5.

Tabla 5

Inconvenientes de las bases de datos distribuidas

Desventaja	Descripción
Complejidad.	Al ofrecer mayor disponibilidad, un nivel aceptable de rendimiento, mayor fiabilidad y la replicación de datos, todo esto de forma transparente al usuario, provoca que la complejidad del SGBD distribuido aumente.
Costo.	El costo de operación y mantenimiento de un sistema distribuido es superior a un centralizado, ya que se debe considerar los costos de mantener una red de comunicaciones y los equipos que ello involucra, además en cada nodo se requiere de personal para administrar los datos locales.
Seguridad.	En un sistema centralizado existe de cierta forma un solo punto de vulnerabilidad, pero en un entorno distribuido se requiere seguridad en todos los nodos en donde se repliquen los datos.
Control de integridad más complicado.	Por la propia naturaleza de la distribución de datos y la replicación en algunos nodos, el control de integridad se vuelve más complicado que si se tiene un sistema centralizado.
Carencia de estándares.	No se visualiza estándares que gobiernen por ejemplo las comunicaciones y acceso a datos en un entorno distribuido, sobre todo cuando se tratan de sistemas heterogéneos en los que se usan diferentes SGBD. Tampoco se cuenta con herramientas o metodologías que ayuden a convertir un SGBD centralizado en un SGBD distribuido.
Falta de experiencia.	Todavía no se ha alcanzado el mismo nivel de experiencia que se ha logrado con los SGBD centralizados lo que puede convertirse en un obstáculo para la adopción de este tipo de tecnología.
Diseño de la base de datos más complejo.	Si ya en algunos casos el diseño de una base de datos centralizada es complicado, una base de datos distribuida debe contemplar la fragmentación de los datos, la asignación de fragmentos en los nodos y las cuestiones de replicación de los datos.

Nota. Adaptado de *Administración de Base de Datos. Guía didáctica*, por J. Morocho et al., 2020, pp.90-91, EdiLoja.

Usos actuales de las bases de datos distribuidas

Las bases de datos distribuidas tienen una variedad de usos en la actualidad entre los que destacan:

- **Aplicaciones empresariales:** las empresas suelen utilizar bases de datos distribuidas para integrar datos de diferentes departamentos y localizaciones geográficas.
- **E-commerce:** las plataformas de comercio electrónico a menudo emplean bases de datos distribuidas para manejar grandes cantidades de transacciones y datos de clientes.
- **Sistemas de gestión de contenidos:** los sistemas de gestión de contenidos utilizan bases de datos distribuidas para asegurar la disponibilidad y la escalabilidad del contenido.
- **Sistemas de mensajería:** las plataformas de mensajería emplean bases de datos distribuidas para manejar la información de los usuarios y sus mensajes en tiempo real.
- **Sistemas de juegos en línea:** los juegos en línea suelen usar bases de datos distribuidas para manejar grandes cantidades de información de juegos y usuarios, así como para asegurar un rendimiento y una disponibilidad óptima.
- **Sistemas de análisis de datos:** Las plataformas de análisis de datos utilizan bases de datos distribuidas para procesar grandes volúmenes de datos y acelerar los tiempos de procesamiento.
- **Sistemas de gestión de recursos humanos:** las empresas emplean bases de datos distribuidas para manejar la información de los empleados, incluyendo los registros de pago, la información de seguros y otros detalles relacionados con los empleados.
- **Investigación científica:** los científicos usan bases de datos distribuidas para almacenar y compartir grandes cantidades de datos experimentales y de investigación, lo que les permite colaborar en tiempo real y realizar análisis en paralelo.

Y también se utilizan para implementar soluciones de alta disponibilidad, empleando mecanismos como la replicación y el balanceo de carga. La replicación completa es muy útil para implementar servidores de respaldo que puedan entrar en operación inmediatamente ante un fallo del servidor principal.

Replicación de una base de datos

Una de las medidas de seguridad recomendadas para garantizar la disponibilidad del servicio de base de datos, es mantener un servidor de respaldo con una réplica de la base de datos, actualizada en línea. Con ello, si el servidor primario falla, se podrá dar continuidad al servicio de manera inmediata con el servidor de respaldo.

Dependiendo del SGBD que estemos usando, existen múltiples herramientas y mecanismos para mantener una réplica de la base datos en tiempo real. En nuestro caso analizaremos como hacerlo utilizando MySQL.

Revise el siguiente recurso con el que podrá comprender cómo funciona un esquema de replicación en MySQL.

[Replicación en MySQL](#)



Texto básico (Hueso Ibáñez, 2016), apartados 6.2.1 y 6.2.3, pp. 196-201.

Como habrá podido comprender, la replicación de bases de datos MySQL es un proceso que permite crear copias de una base de datos en uno o varios servidores secundarios a partir de un servidor principal, con el fin de mejorar la disponibilidad, escalabilidad y redundancia de los datos.

El proceso de replicación implica que los cambios realizados en la base de datos principal se envían y se aplican automáticamente en los servidores secundarios. De esta manera, los usuarios pueden acceder a los datos en cualquier servidor, lo que mejora el rendimiento y la disponibilidad del sistema.

Complementariamente, lo invito a revisar el siguiente video donde se muestra una explicación de [cómo ensayar una replicación en un mismo equipo](#), se requiere tener instaladas dos instancias de MySQL en puertos diferentes.



A continuación, tiene un resumen de los pasos que se siguen en MySQL.

1. **Configurar el servidor principal:** en el servidor principal se debe habilitar la replicación y crear un usuario específico para la replicación.

2. **Configurar el servidor secundario:** en el servidor secundario se debe configurar la conexión con el servidor principal y la base de datos que se va a replicar.
3. **Iniciar la replicación:** una vez configurados los servidores principales y secundarios, se puede iniciar la replicación para que los cambios realizados en el servidor principal se envíen y se apliquen automáticamente en los servidores secundarios.

Si la configuración no se ejecuta correctamente puede generar problemas de consistencia y conflictos. Por ello, es recomendable seguir las recomendaciones y buenas prácticas de MySQL en la configuración de la replicación.



Actividades de aprendizaje recomendadas

Le invito a desarrollar las siguientes actividades para fortalecer el aprendizaje de lo estudiado en la presente semana:

1. Amplíe la investigación de las ventajas y desventajas de los sistemas de bases de datos distribuidas, y con base en ello identifique en qué otros escenarios serían propicio el uso de este tipo de implementaciones.

Tome en consideración que cuando hablamos de aplicaciones de bases de datos distribuidas no nos referimos exclusivamente a su empleo bajo el esquema y topología tradicional (sistema federado), sino también usos que se derivan de conceptos, principios y tecnologías desarrollados en torno a las bases de datos distribuidas.

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.

2. Para la base de datos MySQL del caso de estudio “[Alquiler de bicicletas](#)”, cuya implementación se realizó en el primer bimestre, aplique replicación para configurar una base de datos espejo que se actualice en tiempo real.

En este caso la base de datos que ya tiene configurada actuará como maestro. Y deberá (si no lo ha hecho) instalar una nueva instancia de MySQL en un puerto distinto (por ejemplo 3307), la cual actuaría

como esclavo. A continuación, lo invito a revisar el siguiente video:
[Replicación MySQL maestro-esclavo en la misma PC.](#)

3. Es momento de realizar una autoevaluación de sus aprendizajes para medir sus conocimientos en torno a lo estudiado en la unidad 2: bases de datos distribuidas. Responda al siguiente cuestionario.



Autoevaluación 3

Seleccione la opción correcta.

1. Suponga un escenario en el que tenemos un servidor de base de datos, y varias estaciones de trabajo. En las estaciones de trabajo se ejecutan aplicaciones que piden datos al servidor, y que luego los procesan localmente para generar informes gerenciales. ¿A qué tipo de escenario corresponde?
 - a. Procesamiento distribuido.
 - b. Base de datos distribuida.
 - c. Procesamiento distribuido y base de datos distribuida.
 - d. Base de datos y procesamiento centralizados.
2. Aquellos sistemas de bases de datos distribuidas, en los que los SGBD locales son autónomos, pero que tienen capacidad de actuar conjuntamente cuando se requieren acceder a nodos remotos y ejecutar transacciones globales, se denominan:
 - a. Sistemas compuestos.
 - b. Sistemas federados.
 - c. Sistemas independientes.
 - d. Sistemas heterogéneos.
3. En un sistema de base de datos distribuida, cuando decimos que los usuarios no deben ser conscientes de en cuál servidor están almacenados los datos estamos hablando de:
 - a. Transparencia de la distribución.
 - b. Transparencia de la transacción.
 - c. Transparencia de la heterogeneidad.

4. En una base de datos distribuida, cuando un usuario consulta los datos de una tabla fragmentada, debería especificar en la cláusula from de la sentencia select:
 - a. Simplemente, el nombre de la tabla, tal como lo haría en una base de datos no distribuida.
 - b. El nombre de la tabla y el nombre del fragmento donde están las tuplas que busca.
 - c. El nombre de la tabla, el nombre del fragmento donde están las tuplas que busca, y el servidor donde reside ese fragmento.
5. ¿Cuál de las siguientes corresponde actualmente a una desventaja de los sistemas de gestión de bases de datos distribuidos?
 - a. Datos dispersos: al estar los datos distribuidos en varios servidores es muy proclive a que se pierda la integridad de la base de datos.
 - b. Disponibilidad disminuida: el fallo de uno de los nodos conlleva una reducción de la disponibilidad del sistema.
 - c. Ausencia de normas: no hay protocolos estándar de comunicación entre bases de datos en sistemas heterogéneos.
6. En sistemas de bases de datos distribuidas, la replicación consiste en:
 - a. Implementar dos o más nodos del sistema distribuido con la misma configuración hardware y software.
 - b. Mantener una copia íntegra de una o más tablas en múltiples bases de datos que forman un sistema de base de datos distribuidos.
 - c. Crear vistas en cada uno de los nodos del sistema distribuido, las mismas que acceden a una base de datos principal.

7. En un esquema de base de datos distribuida, aquellas tablas que corresponden a catálogos, tal como países, estados_civiles, tipos_cliente, etc., se las debería.
- Fragmentar horizontalmente.
 - Fragmentar verticalmente.
 - Replicar.
8. ¿Cuál de las siguientes afirmaciones respecto a bases de datos distribuidas es cierta?
- Para la distribución de los datos se puede aplicar fragmentación, replicación o ambos.
 - Si se cae uno de los nodos, se cae todo el sistema.
 - Una consulta solo puede acceder a los datos de una localidad.
9. Suponga las siguientes tablas:
- Bodegas (id_bodega, administrador, ciudad, dirección, teléfono)
- Artículos (id_artículo, nombre, peso, precio, cantidad, id_bodega)
- En un entorno de base de datos distribuida se decide que la distribución (fragmentación) geográfica de los datos se hará de acuerdo con la ciudad. En este caso, ¿qué tipo de fragmentación es la que se deberá aplicar a la tabla artículos?
- Fragmentación horizontal.
 - Fragmentación vertical.
 - Fragmentación horizontal derivada.
 - Fragmentación mixta.
10. La implementación de servidores de respaldo que contienen una copia completa de la base de datos, actualizada en tiempo real, constituye una de las aplicaciones o usos actuales de bases de datos distribuidas.
- Verdadero.
 - Falso.

11. La replicación en MySQL:
- Requiere al menos dos servidores MySQL.
 - Requiere al menos un maestro y un esclavo.
 - Sirve para hacer copias de seguridad.
 - Permite optimizar las consultas.
12. Relacione las normas que determinan la calidad de la fragmentación, con su descripción más apropiada:
1. Completitud.
 2. Disyunción.
 3. Reconstrucción.
 - A. Se debe poder consultar y obtener la tabla global.
 - B. No puede haber tuplas que no tengan un fragmento asignado.
 - C. No debería ocurrir que una misma tupla esté almacenada en dos fragmentos.
 - a. 1A, 2B, 3C.
 - b. 1C, 2A, 3B.
 - c. 1B, 2C, 3A.
 - d. 1A, 2C, 3B.

[Ir al solucionario](#)

Resultado de aprendizaje 17

- Demuestra una comprensión de los sistemas de procesamiento y almacenamiento de datos analíticos en línea.

Este resultado de aprendizaje favorecerá su comprensión y capacidad de uso de soluciones orientadas al procesamiento de datos con características particulares en cuanto a tipo y volumen, y que tienen un uso y aprovechamiento particular en distintos ámbitos de la gestión empresarial, toma de decisiones, gestión del conocimiento, y otros usos o propósitos no convencionales. Le permitirá comprender la importancia que actualmente tienen las tecnologías orientadas al procesamiento y almacenamiento de datos analíticos en línea (OLAP) que se utilizan para procesar grandes cantidades de datos y permitir análisis complejos en tiempo real.

Contenidos, recursos y actividades de aprendizaje



Semana 12

Unidad 3. Conceptos avanzados de bases de datos

3.1. Soporte a las decisiones

Actualmente, los datos se han convertido en un activo muy importante para las organizaciones, cuyo adecuado aprovechamiento puede traer muchos beneficios al manejo operativo, táctico y estratégico del negocio.

Inicialmente, las empresas se ocupan de aprovechar las tecnologías de la información para automatizar y optimizar sus procesos operativos, mediante la implementación de sistemas transaccionales (TPS¹). Pero luego buscarán mediante sistemas de información de gestión (MIS²), aprovechar los datos que se acumulan en los sistemas transaccionales, para obtener reportes que les permitan a los mandos operativos evaluar el desempeño, y controlar y supervisar el negocio. Y luego requerirán aplicar procesos

1 Transaction Processing System

2 Management Information Systems

avanzados de análisis y exploración a los datos para mediante sistemas de apoyo a las decisiones (DSS³) obtener informes que sirvan de apoyo a las decisiones administrativas que enfrentan los mandos medios, y mediante sistemas de apoyo a ejecutivos (ESS⁴) generar informes de alto nivel para la alta gerencia.

Nos enfocaremos principalmente en los sistemas de apoyo a las decisiones (DSS). Cuya definición según Rollano (2017) es la siguiente:

Los DSS son sistemas que combinan modelos analíticos con datos operativos y consultas interactivas para el apoyo al análisis de los mandos medios que se enfrentan a situaciones de toma de decisiones. De ahí que un DSS está dirigido a apoyar el análisis de problemas semiestructurados y no estructurados. Los DSS hacen hincapié en el cambio, la flexibilidad y una rápida respuesta. Con un DSS realizamos un menor esfuerzo para vincular a los usuarios con los flujos de información estructurada dando un mayor énfasis en modelos, supuestos, consultas ad hoc, y gráficos. (p. 47)

Tanto para los DSS como para los ESS, se requiere que los datos a explorar y analizar se encuentren consolidados e integrados. Ello implica que primero se deberá unificar y resumir el contenido de las bases de datos transaccionales en un único repositorio, llamado almacén de datos.

Almacenes de datos

Un almacén de datos (data warehouse) es una base de datos que se utiliza para almacenar grandes cantidades de datos de múltiples fuentes. La idea es que los datos se almacenen de manera centralizada, estructurada, e integrada, para que los usuarios puedan acceder a ellos de manera eficiente y efectuar consultas y análisis complejos.



Revise el siguiente recurso en el cual le permitirá conocer los conceptos, pros, contras, componentes, tipos, y también el proceso de construcción de los almacenes de datos.

[Almacenes de datos](#)

3 Decision Support System

4 Executive Support System

Como bien lo expone Beynon-Davies (2018), uno de los conceptos más extendidos es el propuesto por Bill Inmon (padre de la Data Warehouse) quién define a un almacén de datos como “una colección de datos integrados, temáticos, historiados y no volátiles, organizados para apoyar el proceso de toma de decisiones”.

- **Integrado**, porque integra fuentes de datos heterogéneas, en una sola estructura consistente.
- **Temático**, porque los datos se organizan por temas (hechos) a analizar.
- **Historiado**, (o variable en el tiempo), porque los hechos normalmente tienen asociada una dimensión temporal, que indica cuando ocurrieron.
- **No volátil**, porque la información permanece, no se elimina, no se actualiza en tiempo real, es de solo lectura.

Otro concepto interesante de almacenes de datos es el que proponen Trujillo et al. (2013) quienes los definen como:

Sistemas que almacenan datos históricos para ser utilizados por los DSS y ESS para toma de decisiones estratégicas. Dichos sistemas son eminentemente de consulta y están enfocados a extraer conocimiento de los datos históricos almacenados. Por ello, un almacén de datos no es una base de datos en el sentido tradicional, donde cualquier aplicación de usuario final puede realizar inserciones, actualizaciones y borrados sobre la base de datos. (p. 6)

Aquí es muy importante que usted tenga claro la diferencia entre una **base de datos transaccional** y un **almacén de datos**. Por un lado, la base de datos transaccional al ser el repositorio de datos de los TPS (Sistemas Transaccionales) donde los datos se actualizan permanentemente y necesita un afinamiento orientado a optimizar el rendimiento de las operaciones de actualización (insert, update y delete, en SQL), por lo tanto, se requiere un diseño altamente normalizado, e indexado lo menos posible.

Por otro lado, el almacén de datos siendo la fuente de datos de los DSS donde los datos cambian con menos frecuencia y su uso está centrado en la generación de informes para los directivos, requiere, en cambio, estar optimizado para que las operaciones de consulta (select en SQL) sean muy rápidas, por lo tanto, es permisible en este caso introducir ciertos niveles de redundancia y agregar tantos índices como hagan falta. En la tabla 6 puede ver una lista detallada de las características que diferencian a un almacén de datos de una base de datos transaccional.

Tabla 6

Comparación entre una base de datos transaccional y un almacén de datos

Base de datos Operacional	Almacén de Datos (Data Warehouse)
Datos Operacionales.	Datos Consolidados.
Actual (Importancia del dato actual).	Actual + histórico (importancia del dato histórico).
Detallada (Datos en general desagregados).	Detallada + resumida (datos en distintos niveles de agregación).
Predomina la actualización.	Predomina la consulta.
Consultas simples y estandarizadas.	Consultas complejas y dinámicas.
Diseño normalizado.	Diseño desnormalizado.
Dinámica.	Estática.
Cientos de usuarios (por ejemplo, los usuarios de una organización).	Decenas de usuarios (por ejemplo, los gerentes, los analistas).
MB - GB de datos	GB - TB de datos.
OLTP: OnLine Transaction Processing (Procesamiento de transacciones en línea).	OLAP: OnLine Analytical Processing (Procesamiento analítico en línea).

Nota. Encalada, A., 2023.

Tenga muy presente el factor integrador, que es un muy importante. Una base de datos transaccional normalmente se focaliza en una parte del negocio (ventas, contabilidad, recursos humanos, etc.). Un almacén de datos, en cambio, integra toda la información del negocio.

Otra forma de contrastar estos dos enfoques es según el tipo de procesamiento que se realiza sobre los datos, desde esa perspectiva existen dos tipos de sistemas:

- **Procesamiento de Transacciones en Línea (OLTP):** es la tecnología asociada al conjunto TPS + base de datos transaccional, donde el procesamiento se centra en el registro de transacciones, por ello el

SGBD debe contar con un gestor de transacciones que garantice las propiedades ACID.

- **Procesamiento Analítico en Línea (OLAP):** es la principal tecnología asociada al conjunto DSS + almacén de datos, donde el procesamiento está orientado a la exploración y análisis de los datos. Y, tanto la aplicación como el SGBD subyacente deben garantizar una extracción rápida de los datos.

Cuando una empresa se constituye lo primero que implementará son sistemas transaccionales, para automatizar la operación del negocio. Cuando los datos recopilados a través de los TPS crezcan a un volumen importante, probablemente será cuando se plantee la necesidad de implementar un almacén de datos que le permita integrar todos los datos y sacar un mejor provecho a la información acumulada. Hasta tanto, los reportes gerenciales se obtendrán directamente de las bases de datos transaccionales.

Recuerde, OLTP corresponde al procesamiento de se realiza a través de los sistemas transaccionales usando bases de datos transaccionales. Mientras que OLAP corresponde al procesamiento que se hace a través de los sistemas de soporte a las decisiones utilizando almacenes de datos.

¿Cuándo implementar un almacén de datos?

- Cuando tienen tasas de consulta y explotación de datos muy alta, que están afectando al rendimiento de los sistemas transaccionales.
- Cuando se incrementan los reportes para los niveles gerenciales.
- Cuando se desea consolidar información de distintas fuentes.
- Cuando se desea aprovechar la información acumulada para análisis de tendencias.
- Cuando existen preguntas que no se resuelven con una simple consulta de datos, y se requiere aplicar técnicas avanzadas de análisis.
- Cuando se necesita integrar la información, con el objetivo de analizarla, para poder tomar mejores decisiones.

Inteligencia de negocios

En general un sistema de inteligencia tiene como objetivo brindar la información y el conocimiento necesario a sus usuarios, de manera que les ayude a intuir nuevas respuestas a nuevas circunstancias. En términos de negocio la inteligencia busca justamente eso, convertirse en una herramienta de desarrollo de la intuición gerencial, en alusión a lo que Bill Gates afirma sobre la toma de decisiones empresariales “a menudo te tienes que guiar por la intuición”.

Revise los contenidos fundamentales planteándose como objetivos: definir con precisión el concepto de inteligencia de negocios, identificar sus beneficios, y reconocer las técnicas y herramientas comúnmente usadas para este tipo de soluciones.

Hasta ahora se ha hablado de los almacenes de datos asumiéndolos como aquella base de datos que permite reunir la información de múltiples fuentes en un solo repositorio, que luego se lo utiliza estrictamente para realizar consultas y generar informes que sirvan de apoyo a la toma de decisiones.

Con el tiempo, esta solución técnica que buscaba integrar toda la información de la empresa y alivianar la carga de los sistemas transaccionales fue evolucionando y adquirió cada vez mayor importancia por el gran beneficio estratégico que representa para las empresas el aprovechar sus datos históricos para generar información que les permita por ejemplo conocer mejor a sus clientes, entender mejor el mercado, optimizar costos, focalizar mejor su publicidad, etc., es decir ser más competitivos. Al punto que actualmente la información ha pasado a ser un activo y un recurso estratégico para las organizaciones.

Y esa necesidad latente de aprovechar al máximo los datos, ha llevado a desarrollar nuevas soluciones cada vez más avanzadas que permiten explotar de forma muy eficiente la información que se guarda en un almacén de datos. Actualmente, los directivos de una empresa ya cuentan con herramientas que les permite por sí mismos extraer y visualizar la información que necesitan.

A todo ese conjunto de datos, estrategias y herramientas que permiten integrar, organizar, explorar, generar, analizar y visualizar información útil

para mejorar el negocio, es lo que se conoce como **Inteligencia de Negocios** (**BI**⁵), también conocida como **inteligencia empresarial**.

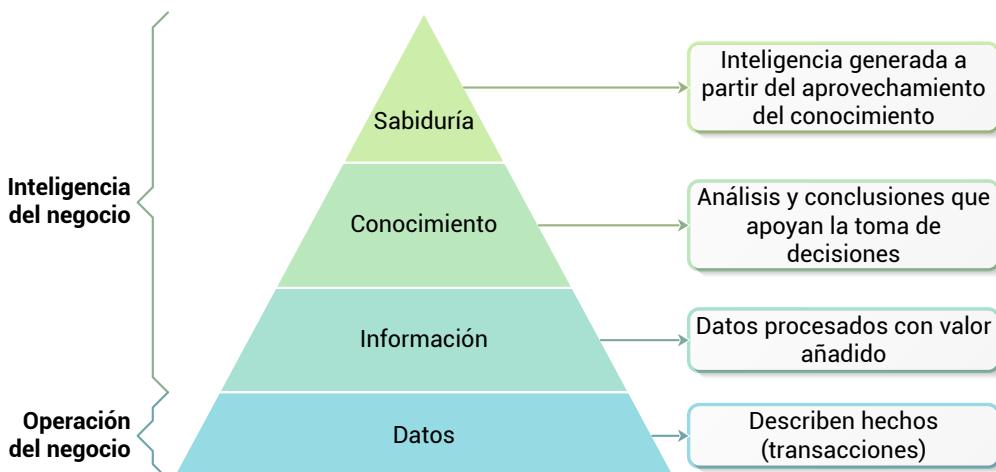
Aquí algunas otras definiciones de BI de autores reconocidos:

- “Es convertir la información de la empresa en un arma estratégica” (Teddy Dale).
- “Se refiere al proceso de convertir datos en conocimiento y conocimiento en acciones para crear la ventaja competitiva del negocio” (*Data Warehouse Institute*).
- “Conjunto de estrategias, acciones y herramientas enfocadas a la administración y creación de conocimiento mediante el análisis de datos existentes en una organización o empresa” (Ahumada-Tello et al., 2012).
- “Es el proceso de analizar y transformar los datos operacionales de una organización en un repositorio accesible con información de mucho valor, y una adecuada distribución de la información en la manera más conveniente a las personas correctas, en el momento preciso y en forma oportuna para tomar mejores decisiones” (Pratte, 2005).

Como puede observar, el término “**conocimiento**” es recurrente, y ese es realmente el objetivo final de la inteligencia de negocios, utilizar la experiencia histórica de la empresa —registrada en los datos acumulados en sus bases de datos transaccionales— para aprender de ella y generar nuevo conocimiento que pueda ser aprovechado para crecer y mejorar. A ese **proceso de transformar los datos en conocimiento** se lo suele conocer como pirámide del conocimiento (figura 18).

5 Del inglés *Business Intelligence*.

Figura 18
Pirámide del conocimiento



Nota. Encalada, A., 2023.

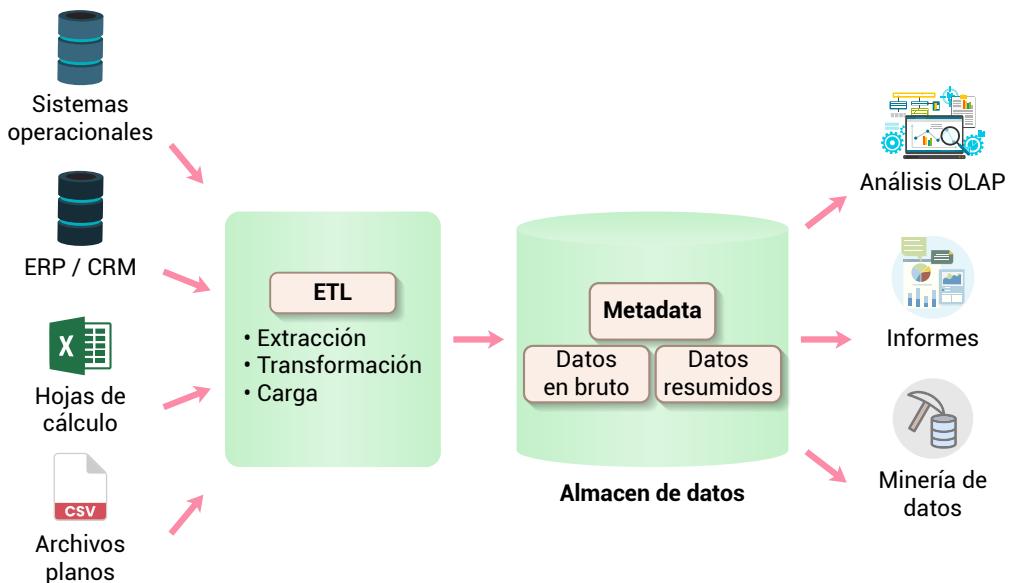
Arquitectura de una solución de inteligencia de negocios

La figura 19, muestra la arquitectura típica de una solución de inteligencia de negocios. Se parte de un conjunto datos operacionales de origen, cuyas fuentes pueden ser internas (bases de datos transaccionales, archivos planos, hojas electrónicas, etc.) o externas (indicadores bursátiles, información gubernamental, etc.). Pueden ser múltiples fuentes heterogéneas, a partir de las cuales se realiza un proceso de obtención, agrupación, limpieza, depuración e integración de datos, de manera que al almacén pase solo lo necesario; proceso conocido como Extracción, Transformación y Carga (ETL⁶). Como resultado del ETL, los datos limpios se guardan en el almacén de datos en esquemas multidimensionales y a partir de allí se extraen, analizan y visualizan a través de las técnicas y herramientas de inteligencia de negocios.

6 Extract, Transform and Load

Figura 19

Arquitectura de una solución de inteligencia de negocios



Nota. Tomado de *Administración de Base de Datos. Guía didáctica* (p. 97), por Encalada, A., 2020, EdiLoja.

Técnicas y herramientas de inteligencia de negocios

En inteligencia de negocios la exploración y análisis de los datos se puede hacer principalmente mediante:

- **Consultas simples**, utilizando por ejemplo el SQL estándar en bases de datos relacionales.
- **Análisis OLAP**, utilizando operaciones OLAP (que corresponden a características extendidas de SQL), o con el uso de herramientas OLAP. Permite el análisis de las métricas del negocio, con datos resumidos almacenados y pre-procesados a nivel de cubos (se los estudia más adelante).
- **Minería de datos**, aplicando técnicas estadísticas y de inteligencia artificial (redes neuronales, redes bayesianas, etc.) para descubrir patrones ocultos en los datos.

En cuanto a soluciones tecnológicas para inteligencia de negocios, en el mercado existen muchas herramientas especializadas en la exploración, análisis y visualización de datos. Comúnmente conocidas como *plataformas*

BI, soluciones de inteligencia de negocios o soluciones de inteligencia empresarial. La tabla 7 **muestra algunas de las herramientas más conocidas**.

Tabla 7
Herramientas de inteligencia de negocios

Herramientas comerciales	Herramientas OpenSource
<ul style="list-style-type: none">▪ IBM Cognos Analytics.▪ SAP Business Objects.▪ Yellowfin.▪ Sisense.▪ Oracle BI.▪ Microstrategy.▪ Power BI.▪ Qlikview.	<ul style="list-style-type: none">▪ Pentaho.▪ SpagoBI.▪ OpenI.▪ RapidMiner.▪ LogiReport.▪ JasperReports.

Nota. Encalada, A., 2023.

La mayoría de estas son consideradas herramientas OLAP, es decir, realizan la exploración y análisis de los datos utilizando principalmente operaciones OLAP, complementadas con análisis estadísticos. Sin embargo, actualmente se avanza notablemente en el desarrollo e implementación de herramientas de **minería de datos** para extraer información no obvia en los datos; *RapidMiner* es un ejemplo de ese tipo de herramientas.

Minería de datos

Se ha mencionado antes a la minería de datos como uno de los métodos para explorar los datos con el objetivo de descubrir patrones no evidentes.

También conocida como *extracción de conocimiento en bases de datos*, la minería de datos es “la extracción de información interesante (no-trivial, implícita, previamente desconocida y potencialmente útil), relaciones y/o patrones a partir de los datos en grandes bases de datos.” (Trujillo et al., 2013, p. 82).

Así como la minería tradicional es la acción de excavar en la tierra para encontrar y extraer minerales, la minería de datos lo que hace es excavar en los datos buscando información o conocimiento oculto que le pueda servir a una organización para orientar mejor sus decisiones y sus acciones. La idea es que, aplicando diferentes tipos de algoritmos sobre un conjunto de datos, se puedan descubrir estructuras o patrones útiles. Por ejemplo, es común

escuchar que detrás de las operaciones financieras de ciertas personas u organizaciones, podrían ocultarse delitos financieros como el lavado de dinero, en este caso la minería de datos puede ayudar en mucho a encontrar y contrastar patrones que alerten sobre estos hechos. Si una persona que normalmente se maneja en un cierto rango de actividad financiera, y de pronto esa actividad se altera considerablemente (tanto en montos como en frecuencia), podría significar una alerta que se requiere investigar.

El poder predecir un hecho antes de que ocurra, es información de alto valor estratégico para cualquier organización. Por ejemplo, para una empresa comercial el poder predecir el volumen de ventas en función de distintas estrategias publicitarias, sería información invaluable. Y así lograr obtener ese tipo de conocimiento es a donde apunta la minería de datos.

La minería de datos en sí es un campo de estudio muy amplio que está en pleno desarrollo, y que actualmente se basa en el uso de técnicas estadísticas y de inteligencia artificial. Y para complementar este tema, considere las siguientes características que definen a la minería de datos según Trujillo et al. (2013):

- Descubre hechos y relaciones de datos.
- Se necesita poca intervención humana.
- Encuentra patrones.
- Determina y establece reglas.
- Almacena y reutiliza reglas.
- Presenta información a los usuarios.
- Puede llevar muchas horas.
- El usuario final debe ser capaz de analizar resultados. (p. 83).



Actividades de aprendizaje recomendadas

Le invito a desarrollar las siguientes actividades para fortalecer el aprendizaje de lo estudiado en la presente semana:

1. Investigue más en detalle otros tipos de sistemas de información, principalmente: Sistemas de procesamiento de transacciones (TPS), Sistemas de Información de Gestión (MIS), Sistemas de Información Ejecutiva (EIS). Establezca las principales características y propósitos

que los definen y que los diferencian respecto a Sistemas de apoyo a la toma de decisiones (DSS).

En su análisis no pierda de vista el objetivo de cada tipo de sistema, las características que lo definen, los usuarios a los que está dirigido, y el tipo de base de datos que se usa como repositorio de datos.

2. Analice la relación que existe entre la inteligencia de negocios y big data. ¿Son conceptos no vinculados?, ¿Se complementan?, ¿Lo uno abarca a lo otro?

Para ello, investigue primero lo que es big data, y los usos y aplicaciones que tiene actualmente. Luego, contraste ambos conceptos en función de los objetivos, beneficios, y tecnologías que aportan. Tenga presente que big data no es únicamente el almacenamiento y gestión de grandes volúmenes de datos, es todo un ecosistema orientado al procesamiento, análisis, y visualización de esos conjuntos de datos masivos.

3. Plantee ejemplos en lo que ha sido útil la minería de datos.

A través de *internet*, busque casos de éxito de la aplicación de minería de datos para generar conocimiento útil y estratégico, y que no hubiese sido posible obtener sin la aplicación de técnicas y algoritmos de minería de datos.

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.



Semana 13

3.2. Procesamiento analítico en línea

La necesidad de un soporte a la toma de decisiones provocó que surjan nuevas herramientas de consulta más avanzadas denominadas Procesamiento Analítico en Línea (OLAP). Las aplicaciones OLAP tienen utilidad en diversas operaciones de una organización, por ejemplo, el cálculo de presupuestos, análisis de ventas, predicción de stock de inventarios, segmentación de mercados y/o clientes, entre otras.

El procesamiento analítico en línea (OLAP) hace referencia a los sistemas de soporte a las decisiones con los que realizan análisis multidimensionales sobre datos corporativos (almacenes de datos), y que según Rollano (2017).

Soportan amigablemente los análisis de cualquier usuario, así como las posibilidades de navegación, seleccionando la información a obtener, permitiendo el análisis de datos segmentados y que permiten ir reduciendo el conjunto de datos reportados. Este tipo de selecciones se refleja en la visualización de la estructura multidimensional, mediante unos campos de selección que nos permitan elegir el nivel de agregación (jerarquía) de la dimensión, y/o la elección de un dato en concreto, pudiendo con ello realizar, entre otras, las acciones de rotar, bajar atributos, navegar, expandir o colapsar los datos mostrados. (p.117).

Por lo tanto, le permiten al directivo de la empresa evaluar el comportamiento de un indicador o métrica desde distintas perspectivas llamadas dimensiones.

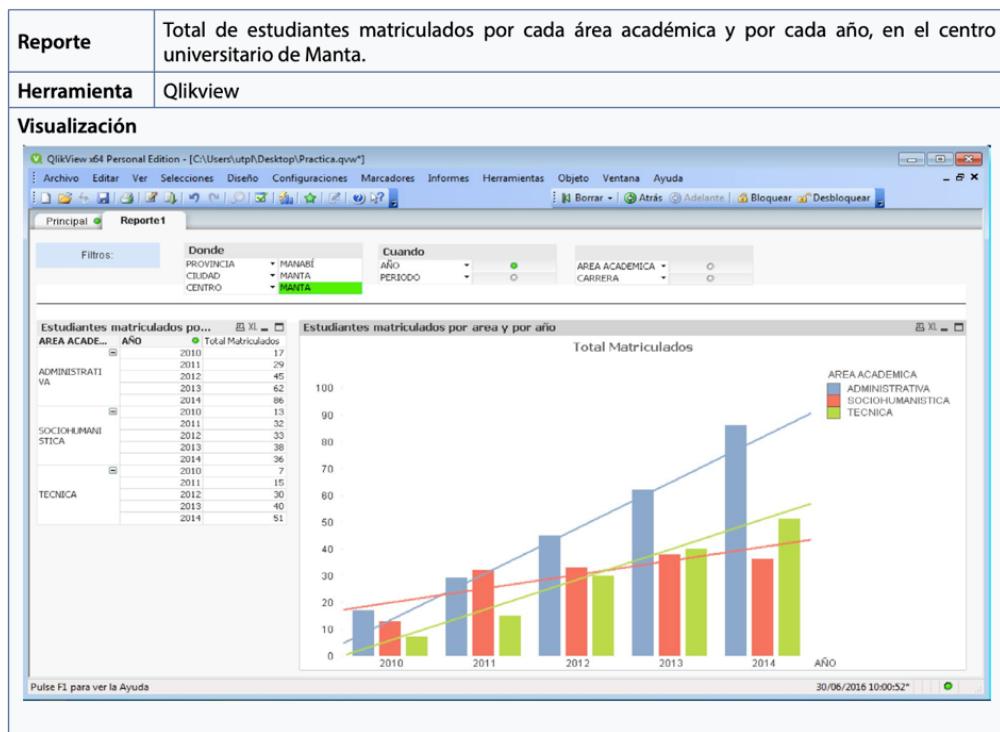
Le invito a revisar el siguiente recurso que le permitirá ampliar la comprensión del concepto OLAP, descubrir las reglas y características de las herramientas OLAP, y conocer los diferentes tipos de servidores OLAP:



[Procesamiento analítico en línea](#)

Biblioteca Virtual UTPL - Ebook Central (Beynon-Davies, 2018), apartados 41.1 a 41.6, pp. 596 - 600.

Como lo ha podido ver existen una serie de 12 reglas que se espera sean observadas por las herramientas OLAP, que nacen básicamente del hecho de que estas herramientas buscan estar al servicio de los mandos medios y altos de la empresa, de manera que el propio directivo sea capaz de personalizar e incluso construir sus propios informes, sin necesidad de una asistencia especializada. En la figura 20, se muestra un ejemplo de cómo sería la interfaz de este tipo de herramientas, en ella, el propio usuario tiene la capacidad por ejemplo de personalizar el tipo de visualización, las dimensiones y parámetros de cada informe.

Figura 20*Ejemplo de herramienta OLAP (datos simulados)*

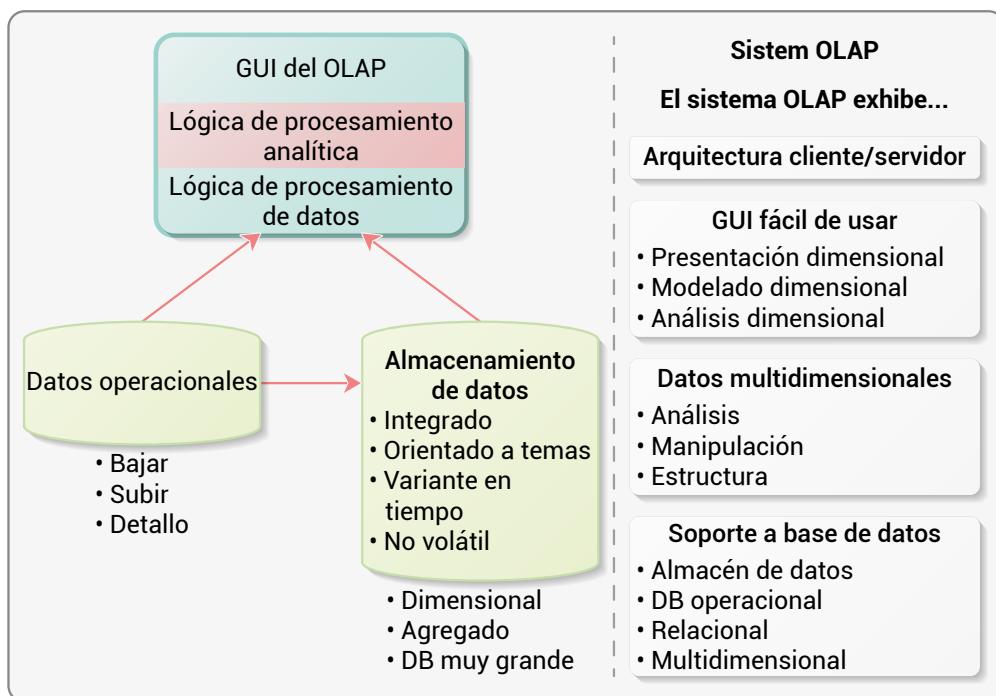
Nota. Encalada, A., 2023.

Arquitectura OLAP

Para hablar de la arquitectura OLAP, en primer lugar, usted debe tener claro que estos sistemas están diseñados para trabajar con datos de un almacén de datos y/o directamente con datos operacionales. La figura 21, muestra el esquema de una arquitectura OLAP aplicada a un entorno cliente-servidor, donde se evidencia lo antes mencionado.

Figura 21

Arquitectura cliente/servidor de un OLAP



Nota. Tomado de *Bases de Datos. Diseño, implementación y administración* (p. 534), por Coronel, C., et al., 2011, Cengage Learning.

Como puede ver, OLAP es un ambiente para análisis de datos avanzados que toma como insumo el contenido del almacén de datos y proporciona soporte para la toma de decisiones. La configuración de un OLAP generalmente es cliente-servidor. En la parte cliente se ejecuta la Interfaz Gráfica de Usuario (GUI) de OLAP mientras que en el servidor se ejecuta la analítica de procesamiento y la lógica de OLAP para el procesamiento de datos. En la mayoría de las implementaciones el almacén de datos y el sistema OLAP son ambientes complementarios.

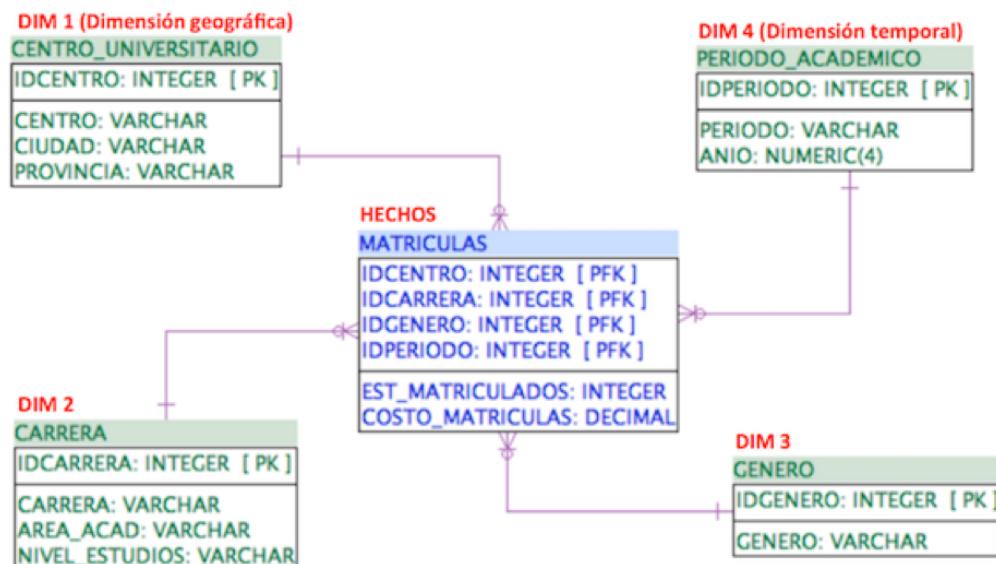
OLAP con SQL

Como ya habrá comprendido con base en lo que estudió en los apartados anteriores, un servidor OLAP a nivel de almacén de datos, utiliza estructuras multidimensionales para guardar los datos y sus relaciones. Y para implementar dichas estructuras, se puede usar bases de datos relacionales, y SQL como lenguaje de consulta y exploración de datos.

En un almacén de datos como ya se ha dicho, la representación de la información se emplea el análisis multidimensional, donde para un tema en específico (hecho), se analiza una serie de métricas (indicadores) desde distintas perspectivas (dimensiones). Si la implementación se la hace sobre un SGBD relacional (SGBDR), ello comúnmente se traduce en un esquema tipo **estrella**, donde en el centro hay una tabla llamada tabla de hechos (FACT) y varias tablas satélites en torno a la tabla de hechos llamadas tablas de dimensiones. A este esquema estrella se lo conoce también como **cubo OLAP**.

La figura 22, muestra un ejemplo de cubo OLAP para analizar los hechos referentes al tema “matrículas” en el contexto de una institución educativa. En este caso Matrículas corresponde a la tabla de hechos, y Centro_Universitario, Periodo_Académico, Carrera, y Género son las tablas dimensionales. Implica que en este caso el cubo OLAP nos permitirá analizar sobre información de *matrículas*, las métricas: *número de estudiantes matriculados* y *costo de matrículas*, desde las perspectivas o dimensiones: *centro, ciudad, provincia, periodo, año, carrera, área académica, nivel de estudios, y género*.

Figura 22
Ejemplo cubo OLAP “Matrículas”



Nota. Tomado de *Base de Datos Avanzada. Guía Didáctica* (p. 104). Por Encalada, A., 2017, EdiLoja.

Debe saber también que este tipo de diseño multidimensional suele ser **desnormalizado**, pues en este caso no importa que exista redundancia, lo importante es que la extracción de los datos sea mucho más eficiente. En la figura 22, por ejemplo, la tabla dimensional Centro_Universitario no llega a 3FN⁷, pues existen dependencias transitivas, lo que provoca que existan muchas repeticiones de nombres de ciudad y de nombres de provincia; desde el punto de vista de una base de datos transaccional eso no sería deseable, pero en un almacén de datos es muy útil y necesario, ya que ello ayuda a mejorar el rendimiento de las operaciones de consulta.

Entonces, de un cubo OLAP, son muchos los informes que un directivo podría generar. Por ejemplo, a partir del cubo OLAP de la figura 22, se podrían obtener informes, tales como:

- Total de estudiantes matriculados por cada área académica y por cada año, en el centro universitario de Riobamba.
- Costo promedio de matrículas por carrera y por año.
- Relación hombres vs mujeres matriculados por área académica y por nivel de estudios.

Y para obtener dichos informes se podrían usar herramientas especializadas -como Tableau, Qlik, PowerBI, etc-, o si es una base de datos relacional se puede usar directamente el comando SELECT del lenguaje SQL, usando la capacidad que este tiene para generar informes estadísticos mediante el uso de las cláusulas GROUP BY, HAVING, y extensiones SQL para OLAP.

SQL usted estudió en la asignatura de Fundamentos de Bases de Datos, sin embargo, aquí se le facilitan algunos recursos para revisar y profundizar el manejo de este lenguaje de cara a la generación de reportes estadísticos o informes gerenciales.

Revise el siguiente recurso con el cual podrá recordar y reforzar con base en ejemplos, como se hace uso de SQL para obtener reportes estadísticos, mediante la agrupación de datos y el manejo funciones de agrupamiento, usando MySQL:

[Uso de SQL para la obtención de reportes estadísticos \(Encalada, 2022c\)](#)

Como ha evidenciado, en SQL los agrupamientos permiten generar informes de resumen, donde las filas de los datos fuente –especificados en la cláusula FROM- se agrupan en uno o varios conjuntos, y por cada conjunto de filas se aplica un cálculo estadístico (contar, sumar, promediar, etc.) mediante las denominadas funciones de agregación o agrupamiento.

Las cláusulas GROUP BY y HAVING completan la estructura de la sentencia SELECT. En la figura 23, se puede observar la estructura completa de dichas cláusulas y el orden en el que se deben especificar.

Figura 23

Cláusulas de la sentencia SELECT

```
SELECT      columna(s), función de agregación  
FROM        tabla o combinación de tablas  
[WHERE       condición de selección de filas]  
[GROUP BY    expresión de agrupamiento]  
[HAVING     condición de selección de grupos]  
[ORDER BY   expresión de ordenamiento];
```

Nota. Encalada, A., 2023.

Recuerde que la condición que se especifica en la cláusula WHERE se aplica para filtrar los datos antes de agrupar, mientras que la condición que se especifica en la cláusula HAVING se aplica luego de agrupar, para restringir el resultado solo a los grupos que cumplan dicho predicado. El ordenamiento se realiza siempre al final, una vez generados y filtrados los grupos.

Para complementar lo estudiado, en la figura 24, se muestra un ejemplo de consulta SQL de agrupamiento con todas sus cláusulas. Específicamente, dicha consulta permite obtener la estadística de los supervisores junto al salario mínimo de los empleados que cada supervisor tiene a su cargo, y únicamente en aquellos casos donde el salario mínimo es mayor a 1500 dólares.

Figura 24

Ejemplo consulta SQL con agrupamiento

```
SELECT supervisor, MIN(salario)
FROM empleados
WHERE supervisor IS NOT NULL
GROUP BY supervisor
HAVING MIN(salario) > 1500
ORDER BY MIN(salario) DESC;
```

Nota. Encalada, A., 2023.

Aprovechemos la consulta mostrada en la figura 24, para resumir el proceso que realiza el motor al ejecutarla:

1. Primero obtiene los datos base, recuperando en este caso las filas de la tabla empleados (cláusula FROM).
2. De los datos base filtra aquellas filas que correspondan a empleados que tienen supervisor (cláusula WHERE).
3. Sobre las filas resultantes ordena y agrupa con base en el supervisor. Es decir, arma un grupo de filas por cada valor distinto de supervisor (cláusula GROUP BY).
4. Por cada grupo determina el salario más bajo (función de agregación).
5. Luego, selecciona únicamente aquellos grupos cuyo salario mínimo es superior a 1500 (cláusula HAVING).
6. Ordena los grupos resultantes según el salario mínimo de mayor a menor (cláusula ORDER BY).
7. Finalmente, como resultado de la consulta, por cada grupo proyecta el valor del supervisor y del salario mínimo asociado (cláusula SELECT).

Es muy importante comprender este proceso para poder especificar y hacer un uso correcto de las distintas cláusulas que conforman la operación SELECT de SQL.



Actividades de aprendizaje recomendadas

Le invito a desarrollar las siguientes actividades para fortalecer el aprendizaje de lo estudiado en la presente semana:

1. Sobre la base de datos MySQL del caso de estudio “[Alquiler de bicicletas](#)”, cuya implementación se realizó en el primer bimestre, escriba las sentencias SQL que permitan obtener los siguientes reportes estadísticos:
 - a. Total de alquileres realizados por cada usuario en el año 2022.
 - b. Promedio de alquileres diarios por cada tipo de bicicleta.
 - c. Porcentaje de alquileres que se realizan en fin de semana, y de lunes a viernes.
 - d. Promedio de número de alquileres que se hacen en franjas horarias de 2 horas: de 08h00 a 10h00, de 10h00 a 12h00, de 12h00 a 14h00, de 14h00 a 16h00, de 16h00 a 18h00, y de 18h00 a 20h00. El objetivo es determinar las horas pico.
 - e. Obtener los 5 usuarios que más uso hacen del servicio de alquiler de bicicletas.

Para ello primero, agregue información a las tablas, principalmente a la tabla alquileres, que sea suficiente para probar y validar las consultas. Analice cada reporte y en función de ello determine la información que debería agregar. Luego, con base en lo aprendido escriba las sentencias SQL para resolver los ejercicios propuestos. Considere los ordenamientos que sean necesarios de acuerdo con el contexto y propósito de cada reporte.

2. Para cada una de las 5 estadísticas generadas en la actividad anterior, construya una visualización de datos que sea apropiada.

Para la visualización puede utilizar Excel, o cualquier otra herramienta que usted conozca. Puede usar distintos tipos de gráficos como barras, líneas, gráficos circulares, gráficos de áreas, o cualquier otro tipo de visualización que usted considera apropiada para cada reporte.

El objetivo es que ensaye una aproximación a la visualización de datos, que es el mecanismo con el cual normalmente se debe presentar la información a un directivo en el contexto de una solución de inteligencia de negocios.



Semana 14

3.3. Bases de datos de propósito especial

Típicamente, las bases de datos se han usado para almacenar información de tipo transaccional en el marco de la gestión operativa de un negocio. Gestión de ventas, gestión bancaria, gestión del proceso de producción, gestión de inventarios, gestión contable, gestión académica, etc., son algunos de los ejemplos de los usos más comunes. A esas bases de datos se las conoce como: bases de datos transaccionales, o bases de datos operativas.

Pero hay otros usos de las bases de datos, que van más allá de la gestión operativa del negocio, que cumplen propósitos más especializados e incluso más estratégicos, y a ello nos enfocaremos en este punto.

La evolución de las tecnologías de información y comunicación (*TIC*) ha provocado que se necesiten nuevos enfoques para el manejo de la información. Hoy en día, se tienen una vasta cantidad de bases de transaccionales en las organizaciones y empresas y además se cuenta con información que crece exponencialmente y que provienen de redes de sensores, redes sociales, aplicaciones y servicios ofrecidos mediante *internet*, datos de corte social/científico, entre otros.

Cuando se piensa en bases de datos inmediatamente se visualizan las bases de datos transaccionales tradicionales como bases de datos de facturación, inventario, gestión financiera, etc. Sin embargo, también se requiere consultar en tipos de datos como: texto libre, imágenes, video y audio, documentos, etc. y esto ha originado la generación de bases de datos con propósitos específicos. Por ejemplo: se cuenta con una base de datos de *podcasts* de programas de radio, charlas, tutoriales o música un caso puede ser la base de datos de *podcast* de la Corporación de Radio y Televisión Española (RTVE).

En función de los diferentes propósitos que se da a la información se han seleccionado algunos tipos de bases de propósito especial:

- Base de datos textuales.
- Bases de datos multimedia.
- Bases de datos temporales.
- Bases de datos espaciales.

Antes de revisar en detalle algunas de las bases de datos de propósito especial, analice la clasificación ampliada de estas que se muestra en el siguiente recurso:

[Clasificación bases de datos propósito especial](#)

Considere que una base de datos en particular no necesariamente debe encasillarse en una sola de las categorías descritas. Por ejemplo, la base de datos de un sistema de registros médicos puede ser considerada como transaccional (gestiona atenciones médicas), temporal (la fecha de atención, y la cronología sintomática y evolutiva son de alta relevancia) y textual (almacena el contenido de prescripciones y exámenes médicos).

También, tome en cuenta que esta categorización –basada en el tipo y propósito de la información que almacena la base de datos– no se tiene que asociar a un tipo particular de SGBD. Por ejemplo, para una base de datos textual podría usarse un SGBD relacional o uno orientado a objetos, para una base de datos espacial se podría usar un motor relacional, para una base de datos multimedia se podría cambio usar, en cambio, un SGBD NoSQL.

Nos corresponde ahora revisar más en detalle cuatro de las bases de datos de propósito especial más reconocidas y usadas actualmente. Nos referimos a las bases de datos textuales, multimedia, temporales, y espaciales.

Bases de datos de texto

Una base de datos textual realiza la manipulación de datos textuales. Fuentes de texto o documentos son básicamente textos completos de algunos artículos, libros o revistas. Estas fuentes son típicamente indexadas mediante la identificación de palabras clave que aparecen en el texto y su frecuencia relativa con la que aparecen en estos documentos.

Para el proceso de indexación, las palabras de relleno o palabras comunes denominadas en inglés **stopwords** deben eliminarse del proceso. Estas palabras pueden ser muchas palabras cuando se está intentando indexar una colección de documentos, estas técnicas han sido desarrolladas para reducir el número de palabras clave y dejar aquellas que son más relevantes para una colección de documentos.

Estos conceptos están asociados a técnicas avanzadas de recuperación de información, que no son parte de esta asignatura. Sin embargo, es necesario analizar algunas bases de datos de este tipo para que se familiarice con su funcionamiento.

Bases de datos multimedia

Las bases de datos multimedia proveen funciones que permiten a los usuarios almacenar y consultar diferentes tipos de información multimedia, que incluye imágenes (fotos o dibujos), *videoclips* (películas, videos), *clips* de audio, (canciones, mensajes de voz, conferencias, entre otros) y documentos (libros o artículos).

Para las bases de datos multimedia entre los tipos de consultas se incluyen aquellas que requieren la localización de fuentes multimedia que contienen ciertos objetos de interés. Se citan los siguientes ejemplos:

- Es posible que desee ubicar todos los *videoclips* en una base de datos de video que incluya cierta persona, e.g. Papa Francisco.
- También, es posible que desee recuperar *videoclips* basados en ciertas actividades incluidas en ellos, como *videoclips* en los que un determinado jugador o equipo anota un gol de fútbol.

Las consultas anteriores se basan en recuperación basada en el contenido, porque la fuente multimedia se recupera en función de que contiene ciertos objetos o actividades. Por esta razón, una base de datos multimedia debe usar algún modelo para organizar e indexar las fuentes multimedia en función de sus contenidos.

La identificación de los contenidos de las fuentes multimedia es una tarea difícil y que requiere mucho tiempo. Hay dos enfoques principales que se pueden emplear:

1. El primer enfoque se basa en el análisis automático de las fuentes multimedia para identificar ciertas características matemáticas de sus contenidos. Este enfoque utiliza diferentes técnicas según el tipo de fuente multimedia (imagen, vídeo, audio o texto).
2. El segundo enfoque depende de la identificación manual de los objetos y actividades de interés en cada fuente multimedia y del uso de esta información para indexar las fuentes. Este enfoque se puede aplicar a todas las fuentes multimedia, pero requiere una fase de pre-proceso manual, donde un debe escanear cada fuente multimedia para identificar y catalogar los objetos y actividades que contiene, de modo que puedan usarse para indexar las fuentes.

Bases de datos de espaciales

Las bases de datos espaciales incorporan funcionalidades que brindan soporte para bases de datos que realizan un seguimiento de los objetos en un espacio multidimensional. Por ejemplo, las bases de datos cartográficas que almacenan mapas incluyen descripciones espaciales bidimensionales de sus objetos, desde países y estados hasta ríos, ciudades, carreteras, mares, etc.

Los sistemas que manejan datos geográficos y aplicaciones relacionadas se conocen como Sistemas de Información Geográfica (GIS) y se usan en áreas tales como aplicaciones medioambientales, sistemas de transporte, sistemas de respuesta de emergencia y gestión de batalla. Otras bases de datos, como las bases de datos meteorológicos para información meteorológica, son tridimensionales, ya que las temperaturas y otra información meteorológica están relacionadas con puntos espaciales tridimensionales.

En general, una base de datos espacial almacena objetos que tienen características espaciales que los describen y que tienen relaciones espaciales entre ellos. Las relaciones espaciales entre los objetos son importantes, y a menudo se necesitan cuando se consulta la base de datos. Aunque una base de datos espacial puede referirse a un espacio n -dimensional para cualquier n , se trabaja en dos dimensiones como la ilustración.

Una base de datos espacial está optimizada para almacenar y consultar datos relacionados con objetos en el espacio, incluidos puntos, líneas y

polígonos. Las imágenes de satélite son un ejemplo destacado de datos espaciales. Las consultas planteadas en estos datos espaciales, donde los predicados para la selección tratan con parámetros espaciales, se denominan consultas espaciales. Por ejemplo, ‘¿Cuáles son los nombres de todas las bibliotecas dentro de las cinco millas del edificio de la Facultad de Informática en Georgia Tech?’ Es una consulta espacial.

Mientras que las bases de datos típicas procesan datos numéricos y de caracteres, se necesita agregar funcionalidad adicional para que las bases de datos procesen los tipos de datos espaciales.

Una consulta como: ‘Enumerar todos los clientes ubicados dentro de las veinte millas de la sede de la compañía’ requerirá el procesamiento de tipos de datos espaciales normalmente fuera del alcance del álgebra relacional estándar y puede implicar consultar una base de datos geográfica externa que mapee la sede de la compañía y cada cliente a un mapa en 2D basado en su dirección. Efectivamente, cada cliente estará asociado a una posición de <latitud, longitud>. No se puede utilizar un índice tradicional basado en los códigos postales de los clientes u otros atributos no espaciales para procesar esta consulta, ya que los índices tradicionales no son capaces de ordenar datos de coordenadas multidimensionales. Por lo tanto, existe una necesidad especial de bases de datos adaptadas para el manejo de datos y consultas espaciales.

Bases de datos científicas

Las bases de datos bibliográficas son recopilaciones de publicaciones de contenido científico-técnico, como artículos de revistas, libros, tesis, congresos, etc., de contenido temático, que tienen como objetivo reunir toda la producción bibliográfica posible sobre un área de conocimiento.

Las bases de datos documentales o bibliográficas contienen información sobre documentos, es decir, referencias a documentos, pero también muchas de ellas contienen los textos completos de esos documentos, generalmente en formato *pdf*. Por ejemplo, Medline, es una base de datos que recopila todo lo que se publica a nivel mundial sobre Medicina.

Si un investigador necesita saber qué se ha publicado sobre un tema de su interés puede localizar esta información buscando en la base de datos que exista sobre su área de investigación o en alguna multidisciplinar.

Las bases de datos contienen información relevante, actualizada, precisa, contrastada y de calidad. Para todas las áreas científicas existe alguna base de datos específica o al menos alguna multidisciplinaria.

Las bases de datos bibliográficas facilitan mucho la tarea de estar informado sobre todo lo que se ha publicado sobre un campo de investigación.

Búsquedas típicas en una base de datos:

- Conocer sobre las publicaciones de un tema determinado.
- Estar al día sobre lo nuevo que va apareciendo sobre un campo de investigación.
- Saber qué ha publicado un autor determinado.
- Identificar instituciones en dónde se está investigando sobre un tema.
- Conocer las revistas que publican más sobre una determinada área de estudio.

3.4. Gestión del conocimiento

Hasta ahora nuestro estudio se ha centrado en la gestión de información estructurada que se almacena en bases de datos tradicionales, las mismas que contienen datos referentes a hechos relacionados con la gestión operativa, administrativa y de dirección de una organización en particular. Información que es de interés o uso exclusivo de la organización que custodia esos datos, y mucha de ella incluso es confidencial.

Pero si ampliamos nuestra visión nos daremos cuenta de que esa información transaccional es realmente una porción muy pequeña respecto a toda la información que existe en una empresa, mucha de ella no visible, y que corresponde al conocimiento adquirido y generado por el conjunto de personas que hacen la empresa, a lo largo de su existencia. Mucho de ese conocimiento se encuentra formalizado y documentado en recursos como manuales, guías, normativas, reportes, y otros tipos de publicaciones internas, pero también mucho de ese conocimiento es tácito y experiencial, y reside en la mente cada miembro de la institución.

La gestión de conocimiento apunta justamente a establecer mecanismos que permitan capturar e indexar todo ese conocimiento explícito o tácito de manera que pueda ser conservado, accedido y transmitido de forma

efectiva. Según Martínez et al. (2010), “la Gestión del Conocimiento consiste en poner a disposición del conjunto de miembros de una institución, de un modo ordenado, práctico y eficaz, además de los conocimientos explicitados, la totalidad de los conocimientos particulares, esto es, tácitos, de cada uno de los miembros de dicha institución que puedan ser útiles para el más inteligente y mejor funcionamiento de la misma, y el máximo desarrollo y crecimiento de dicha institución” (p. 64).



Revise el siguiente recurso de aprendizaje en el que se expone con mayor detalle el alcance y proceso asociados a la gestión del conocimiento dentro de las organizaciones. Preste atención al ciclo de la GdC, y a las etapas del proceso:

[Gestión del conocimiento](#) (Encalada, 2022b)

No lo olvide, la gestión del conocimiento conlleva tres etapas:

1. **Adquisición:** de fuentes internas o externas, físicas o digitales, y acerca de conocimiento tácito o explícito. El conocimiento tácito previamente debe ser materializado.
2. **Representación:** los objetos de conocimiento deben ser indexados y representados en un mapa de conocimiento.
3. **Búsqueda y recuperación:** el conocimiento representado y almacenado en una base de datos debe ser fácilmente accesible mediante una plataforma de búsqueda.



Actividades de aprendizaje recomendadas

Le invito a desarrollar las siguientes actividades para fortalecer el aprendizaje de lo estudiado en la presente semana:

1. Investigue en *internet* otros posibles usos de las bases de datos y profundice los ya identificados. También, identifique ejemplos de aplicación para cada tipo de base de datos de propósito especial.

Para los ejemplos, no necesita identificar el nombre de la base de datos en sí, es suficiente con identificar la plataforma en la que asumimos subyace una base de datos de propósito especial.

2. Analice el caso Netflix y con base en el servicio que provee y la información que maneja, identifique que tipos de bases de datos (según su propósito) subyacen en este servicio.

En su análisis, considere también las bases de datos transaccionales. Tome en consideración que dicha plataforma no solo lleva la gestión de los filmes, sino también maneja de cuentas, pagos, algoritmos de recomendación, etc.

3. Investigue cuáles son las formas y técnicas que existen para indexar y representar el conocimiento. Realice una breve descripción de cada uno.

Como referencia, algunos de los mecanismos conocidos son, por ejemplo: ontologías, tripletas objeto-atributo-valor, redes semánticas, árboles de decisión, entre otros.

4. Analice que relación existe entre las librerías digitales (también llamadas Bibliotecas digitales) y la gestión de conocimiento.

Considere la utilidad que estas pueden tener en la etapa de Búsqueda y Recuperación. Tome como base alguna biblioteca digital, como por ejemplo [PubMed](#).

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.

Resultado de aprendizaje 18, 19 y 20

- Explica el papel de ODBC, JDBC y XML en la implementación de una arquitectura de base de datos de n-tier.
- Utiliza XML en la implementación de una arquitectura de base de datos de n-tier.
- Conecta un número de diferentes bases de datos y/u otras aplicaciones usando SOAP.

Por medio de estos tres resultados de aprendizaje podrá conocer los distintos conectores y mecanismos que comúnmente se emplean para

acceder a las bases de datos desde las aplicaciones, enfocándose principalmente en las arquitecturas de n-capas. Además, le permitirán descubrir como utilizar XML en implementación de servicios que faciliten la conexión de la base de datos con sistemas externos.

Contenidos, recursos y actividades de aprendizaje



Semana 15

3.5. Conectores y APIs

El mecanismo por el cual las aplicaciones acceden a las bases de datos, es a través de los denominados conectores. Un conector de base de datos es un software que permite a una aplicación o programa acceder y comunicarse con una base de datos. Esencialmente, el conector actúa como una API, es decir como una interfaz entre la aplicación y la base de datos, permitiendo que los datos se transfieran de un lugar a otro.

Los conectores de base de datos se utilizan para realizar tareas como la recuperación de datos, la inserción de nuevos datos, la actualización de datos existentes y la eliminación de datos de la base de datos. También pueden usarse para ejecutar consultas y comandos específicos de la base de datos.

Cada base de datos puede tener su propio conector específico, ya que cada base de datos puede utilizar un lenguaje de consulta diferente y tener una estructura de datos única. Los conectores de base de datos más populares incluyen JDBC para bases de datos Java, ODBC para bases de datos Microsoft y ADO.NET para bases de datos de Microsoft .NET.

Papel de los conectores en la implementación de arquitecturas n-tier

Al final de la unidad 1, se explicó lo que es una arquitectura de acceso de base de datos de n-capas. Los conectores como ODBC, JDBC y ADO.NET tienen un papel importante en la implementación de ese tipo de arquitecturas:

1. **ODBC (Open Database Connectivity)**: es una API (Interfaz de Programación de Aplicaciones) que permite a las aplicaciones acceder a cualquier base de datos relacional que cumpla con el estándar ODBC. ODBC se encarga de comunicarse con la base de datos y devolver los resultados de las consultas realizadas por la aplicación. En una arquitectura n-tier, ODBC se utiliza para establecer la conexión entre la capa de presentación (*front-end*) y la capa de acceso a datos (*back-end*).
2. **JDBC (Java Database Connectivity)**: es una API similar a ODBC, pero diseñada específicamente para aplicaciones Java. JDBC permite a las aplicaciones Java acceder a cualquier base de datos relacional a través de una conexión establecida entre la aplicación y la base de datos. En una arquitectura n-tier, JDBC se usa para establecer la conexión entre la capa de presentación en Java y la capa de acceso a datos.
3. **ADO.NET (ActiveX Data Objects .NET)**: es una tecnología desarrollada por Microsoft para acceder a bases de datos desde aplicaciones .NET. ADO.NET permite a las aplicaciones .NET acceder a cualquier base de datos relacional a través de una conexión establecida entre la aplicación y la base de datos. En una arquitectura n-tier, ADO.NET se utiliza para establecer la conexión entre la capa de presentación en .NET y la capa de acceso a datos.

Resumiendo, los conectores como ODBC, JDBC y ADO.NET son tecnologías que permiten a las aplicaciones acceder a bases de datos relacionales de manera eficiente y segura. En una arquitectura n-tier, estas tecnologías se emplean para establecer la conexión entre la capa de lógica del negocio y la capa de acceso a datos. De esta manera, las aplicaciones pueden enviar consultas y recibir resultados de la base de datos de manera rápida y eficiente, independientemente de la plataforma en la que se esté desarrollando la aplicación.

Conectores en MySQL

En MySQL, los conectores y APIs son herramientas esenciales para interactuar con la base de datos y realizar operaciones como consultas, inserciones, actualizaciones y eliminaciones de datos.



Revise el siguiente recurso que le permitirá ampliar su comprensión acerca de los principales conectores, APIs, y drivers que maneja MySQL:

[Conectores y APIs de MySQL](#)

Texto básico (Hueso Ibáñez, 2016), apéndice A, pp. 222-225

Otro conector soportado es MySQL Connector/Python, que es un conector que permite la conexión de bases de datos MySQL con aplicaciones Python. En resumen, MySQL ofrece una amplia variedad de conectores y APIs para conectarse y trabajar con bases de datos MySQL desde diferentes lenguajes de programación y plataformas.

Papel de XML en la implementación de arquitecturas n-tier

XML (eXtensible Markup Language) es un lenguaje de marcado que permite la representación y el intercambio de datos en un formato estructurado y portátil. En una arquitectura de base de datos de n-tier, XML puede ser utilizado de varias maneras:

1. **Intercambio de datos:** XML puede ser utilizado para enviar y recibir datos entre las diferentes capas de la arquitectura. Por ejemplo, la capa de aplicación puede enviar una solicitud a la capa de datos en formato XML, y la capa de datos puede devolver los datos en formato XML a la capa de aplicación.
2. **Definición de datos:** XML también puede ser usado para definir la estructura y el formato de los datos almacenados en la base de datos. Esto permite una mayor flexibilidad y portabilidad de los datos, ya que pueden ser leídos y escritos por cualquier sistema compatible con XML.
3. **Integración con otras tecnologías:** XML es una tecnología ampliamente empleada en la industria, lo que significa que puede ser fácilmente integrado con otras tecnologías, como servicios web o aplicaciones de software, lo que hace que sea una opción ideal para implementar una arquitectura de base de datos n-tier.
4. **Descripción de servicios:** XML también puede ser empleado para describir los servicios que se ofrecen en la capa de aplicación, lo que

permite a los clientes acceder a los servicios de manera más clara y fácil.

En resumen, XML es una herramienta valiosa para la implementación de una arquitectura de base de datos n-tier, ya que permite una mayor flexibilidad, portabilidad e integración con otras tecnologías.



Actividades de aprendizaje recomendadas

Le invito a desarrollar las siguientes actividades para fortalecer el aprendizaje de lo estudiado en la presente semana:

1. Implemente una conexión desde PHP hacia MySQL usando [AppServ](#) o [XAMPP](#). Hágalo utilizando la base de datos MySQL del caso de estudio “[Alquiler de bicicletas](#)”, cuya implementación se realizó en el primer bimestre.

Para hacerlo puede apoyarse en el siguiente recurso [Conectar Base de datos y listar contenidos de tabla con PHP](#). El objetivo es que logre ensayar una conexión a la base de datos, y enviar distintas operaciones SQL de consulta (SELECT) y de actualización (INSERT, UPDATE, DELETE).

2. Investigue ¿cómo usar SOAP para conectar aplicaciones con bases de datos?

Para ello, identifique los pasos que se deben seguir para construir, desplegar, y usar un servicio compatible con SOAP. Sería recomendable, busque un ejemplo de real para comprender mejor dicho proceso.

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.

3. Es momento de efectuar una autoevaluación de sus aprendizajes para medir sus conocimientos en torno a lo estudiado en la unidad 3: Conceptos avanzados de bases de datos. Responda al siguiente cuestionario.



Autoevaluación 4

Seleccione la opción correcta.

1. Cuando decimos que un almacén de datos tiene la característica de ser historiado o variante en el tiempo, significa que:
 - a. Los datos se organizan por hechos.
 - b. Es una base de datos temporal.
 - c. Registra datos sobre hechos históricos.
2. La exploración de los datos históricos en búsqueda de patrones ocultos se logra mediante:
 - a. Consultas SQL.
 - b. Operaciones OLAP.
 - c. Minería de datos.
3. OLTP se refiere a los tipos de sistemas en los cuales la operación principal es:
 - a. La consulta de datos.
 - b. La actualización de datos.
 - c. La replicación de datos.
4. Las herramientas OLAP proveen interfaces de usuario que son:
 - a. Dinámicas, pero solo manejables por programadores o analistas de datos.
 - b. Dinámicas y fáciles de usar por usuarios de dirección del negocio.
 - c. Estáticas y para poder ser usadas por los directivos del negocio.
5. Un almacén de datos puede permitir integrar datos provenientes de hasta:
 - a. Una única fuente de datos.
 - b. Múltiples fuentes de datos homogéneas.
 - c. Múltiples fuentes de datos heterogéneas.

6. Al conjunto de procesos, estrategias y herramientas enfocadas a la administración y creación de conocimiento mediante el análisis de datos en una organización o empresa se denomina:
- Inteligencia de negocios.
 - Inteligencia artificial.
 - Minería de datos.
 - Inteligencia colectiva.
7. Las bases de datos de propósito especial corresponden a otros modelos de organización lógica y almacenamiento físico de los datos, como ocurre con las bases de datos relacionales, de red, jerárquicas, NoSQL, etc.
- Verdadero.
 - Falso.
8. Detrás del servicio que presta la red social Instagram existe una base de datos que primordialmente es
- Transaccional.
 - Textual.
 - Multimedia.
 - Temporal.
9. ¿Cuál de las siguientes sí debe ser considerada como una base de datos espacial?
- Bases de datos que contienen fotos de lugares turísticos.
 - Bases de datos que almacenan imágenes de mapas geográficos.
 - Bases de datos que permiten encontrar las direcciones físicas de organizaciones y empresas.
 - Bases de datos que permiten calcular la mejor ruta para llegar de un lugar a otro.
10. Una base de datos científica normalmente contiene:
- La información de un documento científico.
 - El documento científico.
 - La información y el documento científico.

11. La gestión del conocimiento tiene como finalidad:
- Capturar e indexar todo el conocimiento que subyace en una organización.
 - Administrar de mejor manera los recursos bibliográficos y científicos que posee una empresa.
 - Manejar adecuadamente la propiedad intelectual de las obras y patentes que se generan en una institución.
12. ¿Cuáles dos de las siguientes afirmaciones son correctas respecto a los conectores ODBC, JDBC y ADO.NET?
- Son protocolos para implementación de servicios web.
 - Permiten a las aplicaciones acceder a bases de datos relacionales.
 - Permiten describir los servicios que se despliegan en la capa de aplicación.
 - En arquitecturas n-capas se usan para conectar la capa de aplicación con la capa de datos.
13. XML es un formato de intercambio de datos.
- Verdadero.
 - Falso.
14. En el marco del procesamiento analítico en línea, las vistas multidimensionales facilitan:
- Analizar un indicador desde una sola perspectiva.
 - Analizar un indicador desde distintas perspectivas.
 - Analizar distintos indicadores para una sola dimensión.
15. Las herramientas OLAP están destinadas para ser usadas por:
- Los clientes de la empresa.
 - Los mandos medios y altos de la empresa.
 - El personal operativo de la empresa.

[Ir al solucionario](#)



Actividades finales del bimestre

Estamos finalizando el estudio de nuestra asignatura. Esperamos haber despertado y motivado en usted un mayor interés por esta subárea del conocimiento dentro de las tecnologías de la información. Debe saber que existe un extenso campo de estudio y especialización en torno a la gestión y aprovechamiento de datos, con múltiples opciones laborales. Lo animamos a continuar explorando este campo.

En esta última semana su dedicación debe estar enfocada sobre todo en preparar la segunda evaluación bimestral. Para hacerlo de la mejor manera le sugerimos lo siguiente:

1. Repase lo contenidos estudiados en las unidades 2 y 3, de acuerdo con la secuencia que marca la ruta de aprendizaje. En su repaso incluya aquellos recursos y explicaciones de refuerzo que recibió en el marco de las tutorías semanales.
2. Igualmente, es muy importante que revise las actividades de aprendizaje desarrolladas durante el bimestre (prácticas, foros, *chat*, autoevaluaciones, y cuestionarios).
3. Refuerce el repaso de aquellos temas en los que usted considera que no ha logrado alcanzar los resultados esperados. Tome nota de aquellas cuestiones en las que necesite una mayor retroalimentación.
4. Haga uso de los espacios de tutorías para plantear sus inquietudes a su tutor y recibir la retroalimentación necesaria.

Al igual que en el primer bimestre, las preguntas del examen tienen como principal propósito evaluar su capacidad de comprensión, aplicación, y análisis en torno a los temas estudiados.

¡Le deseamos el mayor de los éxitos!



4. Solucionario

Autoevaluación 1		
Pregunta	Respuesta	Retroalimentación
1	b	En bases de datos relacionales, en una relación 1:N no es posible controlar la cardinalidad mínima del extremo N mediante lenguaje DDL. En este caso primero se deberá registrar la cabecera de la factura y luego los ítems, por lo tanto, ocurrirá al menos un instante en el tiempo en el que exista la factura sin ítems. No es posible insertar en diferentes tablas al mismo tiempo.
2	d	La organización física se centra en definir las especificaciones orientadas al uso del espacio físico del servidor para el almacenamiento persistente de los datos. Tanto en cuanto a la proyección del espacio total requerido, como a la manera en que se distribuirán los archivos de datos en disco, de manera que el motor pueda ejecutar de manera eficiente las operaciones de lectura/escritura.
3	a	Las copias de seguridad es uno de los mecanismos de seguridad más importantes y de obligatorio cumplimiento. Garantizan que, en última instancia, ante un fallo físico en el servidor, los datos no se pierdan.
4	c	El proceso de implementación de la base de datos conlleva preparar y dejar listo el servicio de la base de datos para entrar en producción. Parte de ello es agregar los datos de partida. Toda base de datos normalmente parte con algunos datos ya registrados (catálogos, parámetros), e incluso en muchos casos, con datos migrados desde sistemas anteriores
5	a	Recuerde que el DBA se encarga de implementar y manejar el SGBD. Quien se encarga de planear el contenido de la base de datos es el Administrador de datos.
6	d	Una vez creada la estructura lógica y física de la base de datos (implementación), siempre hay datos que se deben cargar. Si es un nuevo sistema serán los datos de inicialización (catálogos, parámetros, etc.). Si es un sistema que reemplaza a otro, además será necesario migrar los datos existentes en la base de datos anterior.
7	c	Toda restricción de integridad se debe controlar, y se debe controlar de forma automatizada. Si el SGBD no puede hacerlo, entonces lo deberá hacer la aplicación

Autoevaluación 1		
Pregunta	Respuesta	Retroalimentación
8	c	El implementar redundancia a nivel de discos (RAID), no evitaría el ataque de un hacker. En cambio, evitar el acceso directo al servidor de bases de datos, evitar el acceso directo a las tablas, cifrar los datos y colocar una barrera delante del servidor si reduce las posibilidades de que aquello suceda.
9	b	La disponibilidad se refiere a que la base de datos sea accesible permanentemente. La confidencialidad busca evitar que se acceda a información no autorizada. El control de acceso es una medida de seguridad. Mientras que la integridad es la que en efecto busca la consistencia de los datos.
10	b	Podría confundir el tema de las claves de acceso, pero de hecho aquello está contenido en la restricción de acceso a usuarios autorizados. La confidencialidad busca evitar que se acceda a información no autorizada, y el control de acceso justamente previene aquello. Restringir el acceso desde internet no es un algo práctico en la actualidad.
11	a, d	Las vistas de base de datos permiten ocultar información sensible, excluyendo dicha información en la especificación de la vista. También ayudan a que el acceso de los datos desde las aplicaciones sea más eficiente, pues, la aplicación ya no necesitaría enviar una consulta compleja, simplemente consulta la vista, y la complejidad yace en la consulta que subyace en la vista. Las vistas de base de datos no reemplazan a las tablas. Tampoco se confundir con el nivel de vistas estudiado en el marco de los conceptos base.
12	b	<p>La validación de acceso lógico a una base de datos se realiza en dos pasos, y en orden:</p> <ol style="list-style-type: none"> 1. Autenticación: se comprueba que las credenciales del usuario (usuario y password) sean correctas. 2. Autorización: se validan los permisos que tiene asignado el usuario, y en función de ello se le permite acceder solo a los objetos y operaciones que tiene autorizado.

[Ir a la
autoevaluación](#)

Autoevaluación 2		
Pregunta	Respuesta	Retroalimentación
1	a	Las copias de seguridad son una tarea vital de la cuál es responsable el DBA, y se debe obtener de manera periódica y permanente. Y para garantizar su efectividad, periódicamente se deben realizar simulación de recuperación de los datos a partir de los respaldos, para verificar la validez de la copia, y para afinar el proceso de recuperación.
2	c	El aislamiento busca evitar que el accionar de una transacción, afecte a otra con la que se ejecuta de manera concurrente. Obviamente, ese conflicto solo puede ocurrir cuando acceden al mismo elemento de dato.
3	b	Datacenters de respaldo y/o servidores réplicas está sujeto a la capacidad de inversión de la empresa, a la criticidad de los datos y a la disponibilidad requerida. Lo que nunca puede faltar es el respaldar la base de datos de forma permanente.
4	c	No toda transacción actualiza datos de la base de datos, pueden existir transacciones que solo consulten. Tampoco es obligatorio que acceda a todos los registros de una tabla. Lo que no puede ocurrir es que una transacción se ejecute parcialmente, ya que ello generaría inconsistencia en los datos.
5	a	Si una transacción T2 actualiza un dato con base en su valor actual (ejemplo, aumenta un precio en 10%), es necesario que lea el valor más reciente. Si otra transacción T1 actualiza el dato previamente, pero T2 no toma en cuenta esa actualización, entonces el cálculo realizado será inconsistente. T2 sobrescribirá el valor sin haber considerado el valor más actual. En ese caso la actualización que hizo T1 al ser ignorada se considera como perdida.
6	a	El cliente es el equipo desde el que el usuario final accede a los datos (computador personal, estación de trabajo, dispositivo móvil, etc.). Por lo tanto, debe proveer una interfaz de usuario, y además procesamiento propio que permita descentralizar ciertos procesos.
7	b	En la arquitectura centralizada todas las operaciones inherentes al procesamiento de los datos se realizan en el servidor central. Los terminales solo despliegan resultados en pantalla y capturan datos desde teclado; por lo que requerían grandes computadores, para soportar toda la carga en el servidor central.

Autoevaluación 2		
Pregunta	Respuesta	Retroalimentación
8	c	En el servidor de aplicaciones es donde se implementan todos los procesos que controlan la lógica del negocio. Cierta parte se podría implementar también en el cliente (por ejemplo, con Javascript) y/o en la base de datos (con procedimientos almacenados), pero la mayor parte está en las aplicaciones.
9	b	Toda arquitectura de 3 o más capas es también Cliente/Servidor en la que seguirá habiendo una capa cliente (la primera capa), y lo que aumentan son las capas a nivel de servidores.
10	c	Decíamos que en general la seguridad de los datos es un tema que atañe al administrador de datos y al DBA. Pero en el caso de la disponibilidad es algo propio de la implementación física de la base de datos, que implica por ejemplo manejar redundancia de hardware o configurar adecuadamente el SGBD: por lo tanto, es de responsabilidad directa del DBA.
11	a	Los índices son estructuras ordenadas asociadas a una columna o conjunto de columnas. Al ser estructuras ordenadas permiten la implementación de algoritmos de búsqueda rápida que aceleran la localización y acceso de los registros de una tabla.
12	c	El problema de la desnормalización es que, al introducir redundancia en los datos, obliga a que se implementen controles para mantener sincronizados esos elementos redundantes. En bases de datos estáticas que solo son de consulta, eso no representa ningún problema, pero en bases de datos transaccionales que están en constante actualización, puede derivar en inconsistencias, lo cual no es deseable.
13	b	Muchos de los problemas de rendimiento en la ejecución de consultas SQL, se dan porque el usuario (programador) trae más datos de los que necesita procesar la aplicación, sea a nivel de filas o a nivel de columnas. Ello conlleva mayor carga de procesamiento para el motor, y mayor contención en el canal de comunicación. Se debe extraer únicamente los datos que se necesitan procesar.
14	b	“Bloqueo exclusivo” es como comúnmente se lo llama al bloqueo de escritura. La transacción que lo tenga podrá leer o escribir el elemento bloqueado, y ninguna otra transacción podrá accederlo -ni siquiera para lectura- mientras dure el bloqueo.

Autoevaluación 2

Pregunta	Respuesta	Retroalimentación
15	a	En MySQL si es posible realizar copias de seguridad físicas, sin necesidad de detener el servicio. Se pueden usar herramientas nativas o comerciales para el efecto.

[Ir a la
autoevaluación](#)

Autoevaluación 3		
Pregunta	Respuesta	Retroalimentación
1	a	En este caso la base de datos reside en un solo lugar, por lo tanto, es una base de datos centralizada. Por el contrario, el procesamiento no se realiza solo en el servidor, gran parte lo hacen los clientes, por lo tanto hay un procesamiento distribuido.
2	b	El enunciado propuesto define a los sistemas distribuidos semiautónomos, mejor conocidos como federados, en los que cada nodo local tiene la capacidad de resolver las peticiones de los usuarios de ese nodo, sin necesidad de acceder a otros nodos, salvo cuando la operación de datos requiera acceder a datos que están en otros servidores, en cuyo caso tienen capacidad de comunicarse y actuar juntamente con otros nodos del sistema. Pero en la mayoría de los casos no lo requieren ya que todos los datos propios de ese sitio deben estar almacenados en la base de datos local.
3	a	Decimos que para un usuario el sistema de base de datos distribuido debe verse igual que un sistema centralizado. Ello significa que, al realizar una consulta, el usuario no necesita especificar la ubicación de los datos, es decir, para el usuario la distribución de los datos le es indiferente.
4	a	Desde la visión del usuario, una base de datos distribuida debe verse igual que una base de datos centralizada. Eso incluye al lenguaje de manipulación de datos de SQL, La estructura y sintaxis de los comandos DML (SELECT, INSERT, UPDATE, DELETE) no varía.
5	c	Los datos distribuidos no implican afectación a la integridad, a menos que se haga un mal diseño. El fallo de un nodo, en efecto implica que ciertos usuarios no podrán acceder a la base de datos, pero el sistema no cae, en cambio en un sistema centralizado si falla el servidor todos los usuarios se verán afectados. La ausencia de protocolos estandarizados para la integración de SGBD de distintos fabricantes, es en efecto un problema en la actualidad.
6	b	En bases de datos distribuidas, la replicación es considerada un patrón de distribución de los datos, evaluable y aplicable a cada tabla del esquema de datos. Si decidimos que conviene replicar cierta relación, entonces se creará una copia exacta de la tabla (en términos de estructura y datos) en cada uno de los nodos.
7	c	Normalmente las tablas que almacenan catálogos son estáticas, su contenido casi no cambia, son tablas pequeñas, y su toda su información es requerida en todos los nodos del sistema. Por ello lo conveniente es que exista una copia íntegra de dichas tablas en cada nodo.

Autoevaluación 3		
Pregunta	Respuesta	Retroalimentación
8	a	Respecto al fallo de un nodo, recuerde que hay un principio de transparencia llamado transparencia de fallos que indica que indica que si falla un nodo el sistema debe seguir operando. En lo referente al acceso a los datos, recuerde para el usuario el sistema distribuido debe verse igual que un sistema centralizado, por lo que no puede haber restricción de acceso a datos por cuestiones de distribución. En lo que respecta a la distribución en efecto se puede aplicar fragmentación a ciertas tablas y replicación a otras.
9	c	ARTICULOS no tiene el atributo CIUDAD, sin embargo, los artículos si están asociados a una ciudad a través de la bodega a la que pertenecen. Implica que la tabla ARTICULOS se debe fragmentar con base en la fragmentación de BODEGAS, que es lo que se conoce como fragmentación derivada.
10	a	Uno de los legados que han dejado las bases de datos distribuidas es la capacidad de realizar una replicación parcial a total entre dos bases de datos. Esta capacidad es muy usada actualmente para implementar soluciones de alta disponibilidad, que permitan disponer de infraestructura de respaldo que pueda entrar en operación inmediatamente luego de la ocurrencia de un fallo.
11	b	Aunque normalmente la replicación se realiza para disponer de servidores de respaldo, y realizar balanceo de carga, lo que implica tener al menos dos servidores, técnicamente no es indispensable. Es posible replicar entre dos instancias del SGBD que corren en un mismo equipo, actuando siempre una de ellas como maestro. Las copias de seguridad se obtienen en medios terciarios. Y la replicación no determina en si una mejora en las consultas.
12	c	La completitud se refiere a que todas las tuplas de una relación deben estar almacenadas en algún fragmento. La disyunción establece que cada tupla se almacena en un solo fragmento, no puede haber redundancia. La reconstrucción implica que se debe poder volver a reunir todos los fragmentos para obtener la tabla completa.

Ir a la
autoevaluación

Autoevaluación 4		
Pregunta	Respuesta	Retroalimentación
1	b	Al hablar de historiado nos referimos a que los datos tienen asociada una dimensión temporal. Es decir que tienen información de cuando ocurrieron los hechos. Lo que define a una base de datos temporal.
2	c	La minería de datos en efecto permite aplicar técnicas avanzadas de exploración que pueden derivar en la identificación de información no visible en los datos, que puede ser útil para la alta gerencia.
3	b	OLTP corresponde al procesamiento que realizan los sistemas transaccionales, que tienen como objetivo principal la captura de datos relacionados con las operaciones diarias de la empresa. Por lo tanto, las operaciones más frecuentes son la agregación y modificación de los datos.
4	b	Las herramientas OLAP buscan estar al servicio de los mandos medios y altos de la empresa, de manera que el propio directivo sea capaz de personalizar sus informes, sin una asistencia especializada.
5	c	Uno de los propósitos de los almacenes de datos es justamente integrar en un solo repositorio datos de las distintas fuentes que dispongan la empresa, sin restricción de fabricante, tipo de base de datos o formato, siempre que sean datos estructurados.
6	a	Es la inteligencia de negocios la que busca proveer a los líderes de la empresa la información y el conocimiento necesario que ayuden a tomar las mejores decisiones a nivel administrativo y estratégico.
7	b	Las bases de datos de propósito especial hacen referencia al tipo de información que almacenan y al uso que se le da a esa información. No implican estructuras de almacenamiento particulares. Al contrario, estas bases de datos se implementan sobre modelos existentes (relacional, NoSQL).
8	c	Una misma base de datos según su propósito podría encajar en distintas categorías. En el caso de la información que maneja la red Instagram podría considerarse que detrás hay una base de datos transaccional ya que realiza una gestión de usuarios y publicaciones, también sería correcto decir que hay una base de datos textual, ya que seguramente los comentarios estarán indexados para búsquedas; pero como sabemos, el propósito principal de la red es compartir imágenes por lo que primordialmente sería una red multimedia.

Autoevaluación 4		
Pregunta	Respuesta	Retroalimentación
9	d	En las bases de datos espaciales se debe almacenar la geometría espacial de los elementos, no basta con una imagen o descripción del elemento. Es esa información espacial la que, en el caso de la cartografía, permite establecer rutas para llegar de un lugar a otro.
10	c	En la mayoría de los casos las bases de datos científicas contienen la metadata de los artículos, el resumen y el documento publicado (no necesariamente con su contenido indexado).
11	a	La gestión del conocimiento busca capturar el conocimiento tácito y explícito que tiene una empresa. Y que en el caso del explícito puede reposar en muchos medios digitales o impresos. En el caso del conocimiento tácito, debe ser materializado previamente.
12	b, d	ODBC, JDBC y ADO.NET son APIs que facilitan la conexión y comunicación con bases de datos relacionales, desde aplicaciones cliente servidor o aplicaciones web. En el caso de las aplicaciones web, estos conectar permiten vincular la capa de aplicación o lógica de negocio con la capa de datos.
13	a	En efecto XML es un lenguaje de marcado que tiene muchas aplicaciones, una de ellas el uso que se le da como formato para la representación e intercambio de datos a través de servicios web.
14	b	El análisis multidimensional le permite al directivo de la empresa evaluar el comportamiento de un indicador o métrica desde distintas perspectivas llamadas dimensiones.
15	b	Las herramientas OLAP están orientadas a generar y visualizar información útil para la toma de decisiones, por lo tanto, los beneficiarios de dicha información son las personas que tiene bajo su responsabilidad esa tarea (directores, gerentes, etc.)

**Ir a la
autoevaluación**



5. Referencias bibliográficas

- Ahumada-Tello, E., Zárate, R., Plascencia, I., & Perusquia-Velasco, J. (2012). Modelo de competitividad basado en el conocimiento: el caso de las pymes del sector de tecnologías de información en Baja California. *Revista International Administración & Finanzas*, pp. 5-13.
- Beynon-Davies, P. (2018). *Sistemas de Bases de Datos* (1.a ed.). Editorial Reverté. <https://ebookcentral.proquest.com/lib/utpl/detail.action?docID=6795626>.
- Coronel, C., Morris, S., & Rob, P. (2011). *Bases de Datos. Diseño, implementación y administración*. (9a. ed.). Cengage Learning.
- Cuadra, D., Castro, E., Iglesias, A. M., Martínez, P., Calle, F. J., de Pablo, C., Al-Jumaily, H., Moreno, L., Rivero, J., Martínez, J. L., Segura, I., & García, S. (2014). *Desarrollo de Bases de Datos. Casos Prácticos desde el Análisis a la Implementación* (2a. ed.). Ra-Ma.
- Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of Database Systems* (7a. ed.). Pearson.
- Encalada, A. E. (2016a). Bases de datos: Traducir modelo lógico a SGBD [Video]. En YouTube. <https://www.youtube.com/watch?v=0EENWOAOF8o>.
- Encalada, A. E. (2016b). *Bases de datos: Diseño Físico*. YouTube. <https://youtu.be/gUteZ60UBhc>.
- Encalada, A. E. (2017a). *Base de Datos Avanzada. Guía Didáctica* (1a. ed.). UTPL.
- Encalada, A. E. (2017b, septiembre). *Buenas prácticas SQL*. Opus Cum Data. <https://rebrand.ly/bpsql3728>.
- Encalada, A. E. (2017c, septiembre). *Técnicas de desnormalización*. Opus Cum Data. <https://rebrand.ly/tecdes8291>.

- Encalada, A. E. (2020a). *Patrones de distribución*. Opus Cum Data.
[https://opuscumdata.com/wp-content/uploads/2023/02/
PatronesDeDistribucion_v1.pdf](https://opuscumdata.com/wp-content/uploads/2023/02/PatronesDeDistribucion_v1.pdf).
- Encalada, A. E. (2020b). *Administración de Base de Datos. Guía didáctica*. EdiLoja.
- Encalada, A. E. (2021, septiembre). *Arquitecturas de bases de datos*. Opus Cum Data. <https://rebrand.ly/9svrpv6>.
- Encalada, A. E. (2022a, junio). Índices. Opus Cum Data. [https://
opuscumdata.com/wp-content/uploads/2022/08/Indices_v1.pdf](https://opuscumdata.com/wp-content/uploads/2022/08/Indices_v1.pdf)
- Encalada, A. E. (2022b). *Gestión del conocimiento*. YouTube. [https://rebrand.
ly/ohwvctc](https://rebrand.ly/ohwvctc).
- Encalada, A. E. (2022c, agosto 20). *Uso de SQL para la obtención de reportes estadísticos*. YouTube. <https://youtu.be/PuDgQ7n-DF8>.
- Encalada, A. E. (2022d, agosto 28). *Proceso de afinación de una base de datos*. YouTube. https://youtu.be/_AtZYELKMdc.
- Hueso Ibáñez, L. (2015). *Gestión de bases de datos* (2.a ed.). RA-MA Editorial. <https://elibro.net/es/ereader/bibliotecaupl/62491>.
- Hueso Ibáñez, L. (2016). *Administración de Sistemas Gestores de Bases de Datos* (2.a ed.). Ediciones de la U. [https://elibro.net/es/lc/
bibliotecaupl/titulos/106455](https://elibro.net/es/lc/bibliotecaupl/titulos/106455).
- Martínez, M., Pazos, J., & Segarra, S. (2010). *Gestión de la información y del conocimiento* (1a. ed.). Centro de estudios financieros.
- Morocho, J. C., Encalada, A. E., & Romero, A. (2020). *Administración de Base de Datos. Guía didáctica* (1a. ed.). EdiLoja.
- Pratte, D. (2005). Business Intelligence tools are key to building profits. Acesso em, 25.
- Rollano, R. (2017). *Inteligencia de Negocios y Toma de Decisiones* (3a. ed.). CreateSpace Independent Publishing Platform.

Trujillo, J. C., Mazón, N., & Pardillo, J. (2013). *Diseño y explotación de almacenes de datos. Conceptos básicos de modelado multidimensional* (1a. ed.). Editorial Club Universitario.