

ADMBD1302-608-2020 – Modelamiento de base de datos (1)–ⁱⁱⁱ

Felipe Alfonso González L.

Continuidad en Ingeniería en Informática, IACC Chile., 2020-2021

Instituto Superior de Artes y
Ciencias de la Comunicación,
IACC
Av. Salvador 1318, Metro Santa
Isabel, Providencia, Santiago.
Chile.

*f.alfonso@res-ear.ch – felipe.alfonso.glz@gmail.com – <https://twitter.com/felipealfonsog>
<https://glzengrg.com> - <https://freeshell.de/felipe> - <https://linkedin.com/in/felipealfonsog>*

Introducción al periodo

El uso de bases de datos como medios de almacenamiento de información ha tenido un incesante desarrollo durante las últimas dos décadas. Dada la relevancia de la información para la sociedad moderna, garantizar la seguridad y el buen manejo de la misma se ha convertido en un aspecto de vital interés para la sociedad moderna. En un principio, las bases de datos fueron utilizadas para almacenar información en formato digital de una manera muy conveniente, facilitando su recuperación y manipulación para generar reportes basados en los datos almacenados. Esta tendencia fue incrementándose a una velocidad increíble, lo cual hizo necesario desarrollar nuevos métodos y técnicas para manipular dichos datos. La teoría de base datos toma la información de las organizaciones y la almacena para que posteriormente los usuarios de la misma puedan recuperarla y procesarla de una manera por demás fácil e intuitiva.

1.

La importancia de manejar un DBMS, su arquitectura e instalación es fundamental para manejar y proteger información. Para esto, existen mecanismos, que posibilitaran el manejo de esta de una manera rápida y expedita. Es de gran importancia y que se imponen en estos procesos y operaciones son estandarizar las bases de datos como por ej. Un modelo relacional de BBDD. El DBMS ofrecerá a la compañía la posibilidad de almacenar y manejar con alto grado de eficiencia, abstracción, flexibilidad e incluso con facilidad, bases de datos. Existen componentes fundamentales para instalar y configurar un DBMS, aquellos componentes son:

- 1- Datos
- 2- Procedimientos
- 3- Hardware
- 4- Software
- 5- Red

- 6- Personas
- 7- Servidores de BB.DD.

Es también importante considerar que los componentes de un DBMS, son:

- 1- Hardware: dispositivos físicos, servidores o estaciones.
- 2- Software: Es si mismo el DBMS, y es importante a considerar el O.S. a operar, las aplicaciones de programas, y programas utilitarios.
- 3- Usuarios: Son usuarios de un DBMS, como por ej.; Administrador del sistema, Administrador de BB.DD., y diseñadores de BB.DD.
- 4- Analistas y Desarrolladores
- 5- Procedimientos: Son reglas y normas a controlar para mantener de manera integra los datos.
- 6- Datos: Son el componente primario para el desarrollo de información relevante.

Es muy importante considerar como fundamentales en toda esta operación, con profesionales adecuados, que tal como se mencionan en el punto 4 son los analistas y desarrolladores, que tienen altas habilidades para el desarrollo de software, y son fundamentales para la toma de decisiones.

El proceso de instalación de un DBMS es un tema complejo y delicado. Estos procesos deben considerar:

- 1- Lectura del manual de instalación.
- 2- Requisitos sobre el O.S.
- 3- Considerar especificaciones técnicas como por ej., tipo de procesador a usar, etc.
- 4- Chequeo de cantidad optima de memoria RAM y H.D.

Ahora bien, para proceder con la configuración de un DBMS, se debe considerar y proceder con lo Sgte.:

- I- Espacio de Almacenamiento:
 - a) Establecer cantidad de espacio físico idóneo
 - b) Establecer correcta configuración, para un máximo desempeño.

- c) Determinar nivel de exactitud en el diseño de la BB.DD.
- 2- Definición de espacio, se establece espacio requerido para una tabla específica.
- 3- Creación del espacio. Se crean 3 tablas, reservando espacio para las Sgts.:
 - a) SYSCATSPACE: Almacena catalogo de tablas.
 - b) TEMPSPACE1: Almacena tablas temporales.
 - c) USERSPACE1: Almacena tablas, y respectivos índices creados por los usuarios.
 Estas tablas son generadas manera automática, aunque de cualquier modo puede ser creadas por la misma DBMS.
 El espacio inicial de una BB.DD. Estará definido por omisión o defecto.
- 4- Configuración de memoria: Es vital, gran cantidad de memoria para almacenar información, procesos y ejecutar operaciones. Para todo lo anterior se requiere una memoria cache o temporal con el objetivo de minimizar las operaciones de lectura y escritura desde y hacia un H.D.; Desde y hacia se debe definir un Buffer pool que elimina la subsecuente lectura desde el medio físico mientras se mantengan en el Buffer pool.
- 5- Memoria compartida, establece:
 - a) La asignación y liberación de estas áreas de memoria, que ocurren cuando es asignada a un área particular, en un evento particular, por ej.: la conexión a una BB.DD.
 - b) Durante estos eventos la DBMS asignan memoria de la siguiente manera:
 - Al inicio del DBMS.
 - Cuando una base de datos es conectada por primera vez.
 - Cuando una App. Se conecta a una BB.DD.
- 6- Instancias Múltiples, son un conjunto completo de Apps necesarias para el correcto funcionamiento de una DBMS y se pueden instalar múltiples instancias, como:
 - Soporte para Apps: instancias para diferentes organizaciones con diferentes BB.DD. o una única instancia, bajo la mirada de un solo SysAdmin.
 - Consolidación de los servidores, esta característica permite reducir el numero de servidores a utilizar ya que no es necesario un servidor para cada DBMS.
 - Soporte Técnico, Son actividades a nivel de software o hardware, por lo que es ideal un equipo técnico de experiencia o capacitado.

permite el almacenamiento de estructuras de datos, como:

- lógicas
- Y físicas.

Las estructuras lógicas permiten independencia en los datos definidos y en la forma física en como los datos son definidos. Garantizando integridad y desempeño del DBMS, considerando:

La extensión, que compone segmentos y se asignan bloques de espacio libres y continuos, y presentan estas ventajas:

- Se elimina intervención del Sysadmin.
- Se asegura que las Apps no dejen de funcionar.

Segmento, la mayoría de los DBMS utiliza la técnica - Mirror- y requieren ciertas características:

- Los segmentos deben almacenarse dentro del mismo sistema.
- Los segmentos copiados deben ser almacenados en un Host diferente.
- Espacio de tablas. Es un grupo de particiones en la BB.DD. Están compuestos por omisión en la base de datos y son:

- 1) Syscatspace.
- 2) Estructuras de datos que guarda el usuario.
- 3) El espacio de tablas temporales.

En las operaciones sobre una BB.DD. implican las siguientes consideraciones:

Espacios, Los DBMS han experimentado un avance para calcular el espacio de almacenamiento, estos trataran como arreglo lineal de paginas contiguas de BB.DD., logrando usos eficientes de espacio de almacenamiento. Estos espacios de datos serán recuperados para posterior procesamiento. El DBMS dispone un área compartida de gran capacidad de área de memoria (Shared Buffer Poll), organizado como frames, leídos desde el dispositivo HD y cargados en el área común de memoria compartida.

Espacios Privados, son BB.BB. denominadas -Formularios o paneles de control-, su función es:

- Guiar a usuarios a la reorganización de funciones específicas.
- Controlan el acceso a las tablas.

Espacios para objetos, su estructura de datos mantiene -Los llamados archivos-, manejados por el Sysadmin.

Una buena administración redundara en:

- Mejoras de funcionamiento.
- Reducción de tiempos de acceso.
- Almacenamiento persistente y robusto.

Deben considerarse los llamados -Tipos-,

- Clustered.
- Unique.
- Fulltext.

En las estructuras de datos se debe considerar importante el control de archivos en dispositivos de almacenamiento, y su gerencia es parte del Sysadmin junto al diseñador, esto

Bitácoras, cambios que quedan registrados de forma serial, así el DBMS, realiza operaciones como:

- Rollbacks.

- Ejecutar operaciones de control.
 - o Las bitácoras ocupan bastante espacio debido a transacciones denominadas -respaldos-.
 - o Los datos quedan grabados en la bitácora.
 - o Es necesario tener mecanismos de control llamados Checkpoints, que son definidos por el DBA.

Particiones, índices que pueden ser divididas en unidades pequeñas.

Partición de rango, son valores específicos:

- Campos claves
- Fechas específicas

Partición de lista, lista de valores discretos para ser utilizadas como claves.

Partición Hash, requiere claves Hash, distribuye filas de datos de manera uniforme.

Configurar los diferentes modos de operaciones de un DBMS dan a entender que se debe tomar en cuenta las siguientes operaciones:

- Operaciones esenciales de un DBMS, proporcionan de manera segura, rápida y confiable 3 aspectos importantes:
 - a) Procesos y actualizaciones
 - b) Recuperar datos y garantizar su correcto uso.
- El DBMS, debe considerar:
 - o El software requerido para manejo y control de la BB.DD.
 - o Datos representativos del modelo
 - o Procedimientos.
 - o El lenguaje de la BB.DD.
 - o Procesador de consultas.
 - o Gestión de tiempo de ejecución.
 - o Manejo de ejecución y autorización de usuarios y modificaciones.
 - o Ejecutar funciones para creación de tablas, respaldos y datos copiados.
 - o Modelo de BB.DD. garantiza almacenamiento y recuperación de datos.
 - o Generador de reportes, genera apoyo para la toma de decisiones:
 - Recuperación de Callback
 - Permanencia o Commit.

En las Operaciones de una BB.DD., destacan, Baja y Alta:

- Baja: Es un borrado de un registro.
- Alta: Gestiona nuevos registros a la BB.DD.

Recovery, protección de los datos ante cualquier daño de su integridad y confiabilidad. De otro modo sucede lo siguiente:

- o Daño en la superficie magnética del disco
- o Errores en las aplicaciones.
- o Errores de usuarios.

Es importante también el manejo de Índices:

- o Un nuevo registro se realiza de manera secuencial.

- o Búsquedas y recuperación contraria a la inserción secuencial.
- o Técnica de Indexación, son procesos de búsqueda y recuperación.

A pesar de que ya existe una importancia bastante grande en el significado de las BB.DD., estos deben estar en condiciones que garanticen a un mayor nivel de seguridad y respectivo control de datos. Esto de hecho ha generado una total nueva disciplina, orientada a conceptos relacionados con Cyber-Seguridad.

Generalmente los datos se respaldan, con una copia de archivos, luego se almacenan incluso en zonas geográficas distintas como incluso otros países y así recuperarlos y seguir con las operaciones de la organización. Se debe tomar en cuenta la seguridad y métodos para proteger datos electrónicos, impresos e incluso manuales. Es fundamental mantener la información privada, confidencial y sensitiva protegida.

Existen varios métodos de monitoreo, de respaldo, recuperación, formas de replicas, métodos de back ups, comandos de respaldo y otros que podre abordar en la respuesta numero dos y tres.

Una forma confiable e integra de mantenimiento y respaldo de la información es, la Permisología, esta operación es fundamental y lograra un total y establecido tipo de acceso, autenticaciones, privilegios asociados a usuarios o grupos. Dentro de los privilegios existen un ROLE, que podrá empaquetar un grupo de funciones relacionadas con la BB.DD. El establecimiento de ROLES hace mas fácil el control de acceso a los datos, y, en consecuencia, una mejor gerencia por parte del SYSAdmin.

Hay operaciones de comandos que respaldan los datos. Me referiré en detalle sobre este tópico en la respuesta numero dos.

Implementar métodos de respaldo y recuperación es fundamental para salvaguardar la información. Con el tiempo se han generado variadas técnicas en esta área, como:

- Espejo (Mirroring), crear una copia completa en un momento específico del tiempo. No requiere de ningún proceso o algoritmo de comprensión de datos.
- Replica (Replication), a esto también se le denomina base de datos publica.
- Elementos y frecuencia de respaldo, debido a las amenazas en informática como hacking o desastres naturales es fundamental tomar en cuenta el medio de almacenamiento en su forma física, su vida útil, volúmenes, etc.
- Es ideal tener un sistema de rotación de los medios físicos para respaldo y definir la frecuencia bajo la cual se llevan a cabo los respaldos.
- Comandos de respaldos, son comandos en el sistema para la implementación de respaldos ligado al O.S. y dada la importancia que exista un error humano entre los comandos a utilizar hay programas utilitarios que pueden hacerse cargo de estas operaciones.

Las operaciones de monitoreo y control son hoy en día

fundamentales, debido al exponencial flujo de información que se experimenta día a día. Para esto debe existir operaciones de control y monitoreo sobre las BB.DD. que se han tornado muy importantes. Estas son:

- Monitoreo general, esto se refiere a los problemas generales en el desempeño de los sistemas. Es imperativo establecer buenas practicas y directrices para que se pueda enfrentar cualquier inconveniente de una manera rápida.
- Monitoreo del espacio en Disco, se deben establecer políticas de respaldo y recuperación, por lo que incrementara su espacio en Disco requerido. Es una tarea que el Sysadmin debe proyectar.
- Monitoreo de Logs, son archivos con registros de transacciones, operaciones y cambios realizados a la BB.DD. que pueden ser analizados.
- Monitoreo de memoria compartida, se debe compartir memoria RAM con otros programas, es vital aquí la gestión de la BB.DD. y la instalación y configuración del DBMS, para especificar cuanta memoria será reservada para el sistema.
- Monitoreo de BB.DD., se mantienen en observación constante las bases de datos, esto permite reportes para ser analizados. El proceso de actividades es conocido como DAM (Database Activity Monitoring).
- Monitoreo de Modos de operación, el monitoreo constante hace posible la generación de Logs de alerta, que conserva información relacionada a errores o situaciones irregulares, esto hace posible tomar las acciones correctivas necesarias para prevenir ocurrencias de algún tipo en el futuro.
- Monitoreo de espejos, tal como se menciona antes la técnica "Mirror" o espejo es muy efectiva. Permite garantizar integridad y confiabilidad de los datos.

Tomando sin duda en cuenta la BB.DD. que se entrego, para desarrollar las consultas en SQL. Desarrolle una BB.DD. nombrada como ARTICULOS con sus respectivas Tablas, 'Cod_Art', 'Descripcion' y 'Precio'.

¿Cómo se puede obtener de el/los artículos con el menor usuario?, a mi parecer es bastante simple obtener en este caso el resultado. Para lograr el objetivo de menor precio se debe utilizar la función MIN, la cual devolverá como resultado un valor mínimo de la columna deseada. Funciones MAX & MIN estas funciones retornan, respectivamente, el menor y mayor valor de los valores contenidos en la columna especificado.

Estas dos funciones son muy necesarias al momento de trabajar en la hoja de cálculo Excel o en una BB.DD. y es que las mismas tienen como función principal ayudar a encontrar valores determinados.

En el caso de la función MIN va a proporcionar el valor mínimo de una lista de valores, excluyendo los valores de textos y lógicos.

En este tipo de función se pueden considerar los siguientes aspectos que nombrare a continuación:

- Se toman en cuenta los valores lógicos y los textos de números escritos directamente en la lista de argumentos.
- Los argumentos pueden llegar a ser tanto nombres, matrices, número o referencias que contengan números.
- En el caso que los argumentos que no contengan ningún número entonces la función MIN devuelve 0.
- En el caso que el argumento se trata de una referencia o una matriz, se utilizará solamente los números que estén contenidos ya sea en la matriz o referencia. Lo que son las celdas vacías, los textos contenidos en las referencia o matriz y los valores lógicos se pasaran por alto.
- Si existen valores de error o texto que no se puedan traducir a números simplemente generarán un error.^{iv}

Mientras que la función MAX va a permitir poder obtener el valor máximo de un conjunto de valores ya sea una lista de números ubicados en una BB.DD. De esta manera, la función permite devolver el valor máximo de un conjunto de valores.

Como un argumento de la función se puede ingresar los números que se quieren evaluar o simplemente un rango o la referencia de una celda que contenga los valores numéricos. Ej.:

SELECT MIN(Salario) FROM Empleados;
Y retorna en algún caso hipotético: 150.000
Otro ejemplo utilizando la función MAX:
SELECT MAX(Salario) FROM Empleados;
Retornando en algún caso hipotético: 980.000

En este caso se requiere tratar de encontrar el articulo con el menor precio, por lo siguiente la columna donde se hará la consulta, considerando que realicé una columna con determinados ID para cada valor esta en la segunda fila 'Preciocol' siendo Precio la Tabla. La instrucción SQL sería:

SELECT MIN(Preciocol) FROM ARTICULOS.Precio;

Ahora bien, para responder a la pregunta numero dos, acerca de, ¿cómo generaría una consulta que contenga la descripción de cada artículo con su respectivo precio, y, ordenándolo con la misma forma descendente en base al precio del articulo? El ordenamiento del resultado de una consulta, obtenida mediante una clausula SELECT, se logra mediante el uso de la clausula ORDER la cual organizara los registros en base al contenido del campo especificado como parámetro para la misma. Serán ordenados de forma ascendente, sin embargo, puede ser revertido.

La siguiente sintaxis de esta cláusula es la siguiente:

SELECT <columna 1, columna 2, ... columna n> FROM <nombre de la tabla> ORDER BY <columna> ASC/DESC.

También se utilizan ciertos elementos, utilizando la clausula ORDER, tomando en cuenta, como dije anteriormente

desarrollé una estructura SQL con sus respectivos ID para cada columna, siendo en este caso 'Cod_Art', 'Descripcion' y 'Precio' las respectivas Tablas. El siguiente script SQL a aplicar, sería el siguiente:

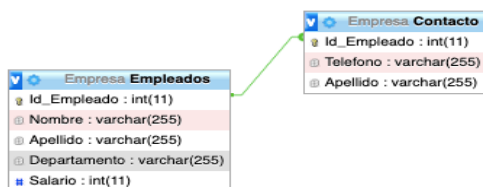
```
SELECT DISTINCT Descripcioncol, Preciocol
FROM ARTICULOS.Descripcion, ARTICULOS.Precio;
```

Esta consulta SQL dará como resultado valores del campo 'Preciocol' y a su lado su descripción. Como siguiente ejercicio final, la siguiente consulta SQL es para ordenar los elementos de la tabla de menor a mayor, mostrando su respectiva descripción de cada artículo junto al precio descendente, sería:

```
SELECT DISTINCT Descripcioncol, Preciocol
FROM ARTICULOS.Descripcion, ARTICULOS.Precio
ORDER BY Preciocol DESC;
```

La importancia del manejo de la información he hecho que la relevancia de las DBMS ampliase su gestión en bases de datos y tablas y abarcando la estructura lógica. SQL mantiene comandos que permiten manipular tablas en bases de datos tanto por usuarios como por administradores. Estos hacen posible un mejor control de los datos y parámetros que garantizan su confiabilidad e integridad. Los DBMS mejoraron, en su desarrollo, y pueden ejecutar consultas en bases de datos, pudiendo combinar, de varias maneras, datos contenidos en diferentes tablas para poder generar reportes facilitando la toma de decisiones gerenciales de manera rápida y oportuna.

Estructura lógica: La explicación lógica y modelo relacional de las correspondientes tablas en la base de datos empresa, conectada con la tabla Contacto, dentro de la base de datos:



Una tabla creada en SQL requiere que cada tabla creada tenga un nombre asignado, así como los campos que contendrá. Se le debe asignar un nombre y, necesita definir el tipo de dato. De tal manera:

```
CREATE TABLE `Empresa`.`Empleados` ( `Id_Empleado`
INT(11) NULL , `Nombre` VARCHAR(255) NOT NULL ,
`Apellido` VARCHAR(255) NOT NULL , `Departamento`
VARCHAR(255) NOT NULL , `Salario` INT(11) NOT
NULL , PRIMARY KEY ( `Id_Empleado` )) ENGINE =
InnoDB;
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Id_Empleado	int(11)			No	None			Change Drop More
2	Nombre	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	Apellido	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
4	Departamento	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
5	Salario	int(11)			No	None			Change Drop More

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	Id_Empleado	0	A	No	

Para definir la auto incrementación en fila Id_Empleado, se ejecuta el siguiente script:

```
ALTER TABLE `Empleados` CHANGE `Id_Empleado`
`Id_Empleado` INT(11) NOT NULL AUTO_INCREMENT;
```

Al hablar de Índice, y la creación de esta con base en varias columnas, con la finalidad de posibilitar la búsqueda de registros mas expedita, se utiliza:

```
ALTER TABLE `Empresa`.`Empleados` ADD INDEX
`Clave_empleado` ( `Id_Empleado` , `Nombre` , `Apellido` ,
`Departamento` , `Salario` );
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Id_Empleado	int(11)			No	None			Change Drop More
2	Nombre	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	Apellido	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
4	Departamento	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
5	Salario	int(11)			No	None			Change Drop More

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	Id_Empleado	0	A	No	
Edit Drop	Clave_empleado	BTREE	No	No	Id_Empleado Nombre Apellido Departamento Salario	0 0 0 0 0	A A A A A	No No No No No	








































Una forma de mantenimiento en SQL para eliminar registros de una BB.DD., y para borrar tabla o un índice. La sintaxis es:

```
DELETE FROM `Empleados` WHERE `Id_Empleado` =
"11";
```

Options		Id_Empleado	Nombre	Apellido	Departamento	Salario
<input type="checkbox"/>						
<input type="checkbox"/>	Edit Copy Delete	1	Luis	Saldiver	Contabilidad	340000
<input type="checkbox"/>	Edit Copy Delete	2	Laura	Perez	IT	480000
<input type="checkbox"/>	Edit Copy Delete	3	Estela	Mardonez	Mercadeo	360000
<input type="checkbox"/>	Edit Copy Delete	4	Maria	Lara	Ventas	980000
<input type="checkbox"/>	Edit Copy Delete	5	Jose	Gutierrez	Finanzas	150000
<input type="checkbox"/>	Edit Copy Delete	6	Pedro	Zerpa	Contabilidad	400000
<input type="checkbox"/>	Edit Copy Delete	7	Esteban	Saldiver	IT	289000
<input type="checkbox"/>	Edit Copy Delete	8	Luisa	Perez	Mercadeo	545000
<input type="checkbox"/>	Edit Copy Delete	9	Jaime	Mardonez	Ventas	600000
<input type="checkbox"/>	Edit Copy Delete	10	Loreto	Lara	Finanzas	388000

<input type="checkbox"/>	 Edit  Copy  Delete	8	Luisa	Perez	Mercadeo	545000
<input type="checkbox"/>	 Edit  Copy  Delete	9	Donald	Mardonez	IT	600000
<input type="checkbox"/>	 Edit  Copy  Delete	10	Loreto	Lara	Finanzas	388000

Options						
<input type="checkbox"/>	Edit	Copy	Delete			
<input type="checkbox"/>	Edit	Copy	Delete	1	Luis	Saldiver Contabilidad 340000
<input type="checkbox"/>	Edit	Copy	Delete	2	Laura	Perez IT 480000
<input type="checkbox"/>	Edit	Copy	Delete	3	Estela	Mardonez Mercadeo 360000
<input type="checkbox"/>	Edit	Copy	Delete	4	Maria	Lara Ventas 980000
<input type="checkbox"/>	Edit	Copy	Delete	5	Jose	Gutierrez Finanzas 150000
<input type="checkbox"/>	Edit	Copy	Delete	6	Pedro	Zerpa Contabilidad 400000
<input type="checkbox"/>	Edit	Copy	Delete	7	Esteban	Saldiver IT 289000
<input type="checkbox"/>	Edit	Copy	Delete	8	Luisa	Perez Mercadeo 545000
<input type="checkbox"/>	Edit	Copy	Delete	9	Jaime	Mardonez Ventas 600000
<input type="checkbox"/>	Edit	Copy	Delete	10	Loreto	Lara Finanzas 388000

+ Options					Nombre
<input type="checkbox"/>					
<input type="checkbox"/>				Luis	
<input type="checkbox"/>				Laura	
<input type="checkbox"/>				Estela	
<input type="checkbox"/>				Maria	
<input type="checkbox"/>				Jose	
<input type="checkbox"/>				Pedro	
<input type="checkbox"/>				Esteban	
<input type="checkbox"/>				Luisa	
<input type="checkbox"/>				Jaime	
<input type="checkbox"/>				Loreto	
<input type="checkbox"/>				Estefani	
<input type="checkbox"/>				Angel	

```
SELECT Id_Empleado,SUM(Salario) FROM Empleados
GROUP BY Id_Empleado HAVING SUM(Salario)>300000
```

Showing rows 0 - 8 (9 total, Query took 0.0064 seconds.)

```
SELECT Id_Empleado,SUM(Salario) FROM Empleados GROUP BY Id_Empleado HAVING SUM(Salario)>360000
```

Show all | Number of rows: 25 | Filter rows: Search this table

Options

	Id_Empleado	SUM(Salario)
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	340000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	480000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	360000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	980000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	400000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	545000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	9	600000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	10	388000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	12	780000

Check all | With selected: Edit Copy Delete Export

Intersecciones:

Tomando en cuenta que se creó la base de datos 'Contacto', la sintaxis es:

```
CREATE TABLE `Contacto`.`3` ( `Id_Empleado` INT(11)
NULL AUTO_INCREMENT , `Telefono`
VARCHAR(255) NOT NULL , `Apellido`
VARCHAR(255) NOT NULL , PRIMARY KEY
(`Id_Empleado`)) ENGINE = InnoDB;
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Id_Empleado	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	Telefono	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	Apellido	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More

Check all | With selected: Browse Change Drop Primary Unique Index Fulltext Add to c

Print | Propose table structure | Track table | Move columns | Normalize

Add 1 column(s) after Apellido Go

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	Id_Empleado	0	A	No	

Create an index on 1 column(s) Go

Partitions

Inserto datos:

```
INSERT INTO `3` (`Id_Empleado`, `Telefono`, `Apellido`)
VALUES (NULL, '955787689', 'Saldiver'), (NULL,
'964356545', 'Perez'), (NULL, '987445323', 'Mardonez'),
(NULL, '934558879', 'Lara')
```

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Options

	Id_Empleado	Telefono	Apellido
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	955787689	Saldiver
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	964356545	Perez
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	987445323	Mardonez
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	934558879	Lara

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

'Union' permite combinar los registros obtenidos mediante la ejecución de dos o mas cláusulas, que deben contemplar

la misma cantidad de columnas.

```
SELECT Id_Empleado, Nombre FROM Empleados UNION
SELECT Id_Empleado, Telefono FROM Contacto;
```

Showing rows 0 - 15 (16 total, Query took 0.0045 seconds.)

```
SELECT Id_Empleado, Nombre FROM Empleados UNION SELECT Id_Empleado, Telefono FROM Contacto
```

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Options

Id_Empleado	Nombre
1	Luis
2	Luana
3	Estela
4	Maria
5	Jose
6	Pedro
7	Esteban
8	Luisa
9	Jaime
10	Loreto
11	Estefani
12	Angel
1	955787689
2	964356545
3	987445323
4	934558879

SQL mantiene un conjunto de comandos que logran manejar bases de datos de una forma fácil e intuitiva. La cláusula SELECT siempre tiene una gran versatilidad, Permite la realización de cualquier tipo de consulta sobre los datos. Estas características han hecho de SQL el lenguaje preferido para el manejo y gestión de los sistemas de bases de datos, lo cual ha generado que se desarrollen diferentes versiones para ser usadas en entornos más específicos.

Originalmente MySQL era un software propietario, 5 años después de 1995 se volvió OpenSource con soporte para Linux-Unix.

MySQL no estuvo orientado a una gran escala, su uso era mas bien practico. Fué tan bien escrito, que se convirtió en un DBMS preferido.

Su instalación difiere un poco dependiendo de ciertos aspectos, como:

- Establecer compatibilidad para MySQL, se requiere verificar si soporta y esta respaldado por la empresa Oracle Corp.
- Decidir cual distribución instalar, binario, o código abierto (Permite acceso para desarrollo), verificar descarga bajo Oracle Corp., Verificar bajo que O.S. se instalará.
- MySQL Community Server, descargable de manera libre, para pequeños o medianos usuarios, soportada bajo licencia GPL, así se tiene acceso a esta, como código abierto, soportada para colaboradores y Developers.

Bajo la licencia GPL, existen limitaciones comparado a Enterprise Edition que tiene funciones adicionales para el ámbito empresarial.

Se puede crear una tabla con un respectivo ID, por eje.:
 CREATE TABLE Empleados (Nombre varchar(50),
 Apellido varchar(50), ID int AUTOINCREMENT
 PRIMARY KEY (ID));

Para ingresar 10 registros en la tabla anterior considerando que en la columna Apellido cinco de ellos deben comenzar con la letra D.:

```
INSERT INTO Empleados (Nombre, Apellido, ID)
VALUES ('Dohn', 'Lemon', ''), ('Daul', 'Martney', ''),
('Detephen', 'Ding', ''), ('Daily', 'Moon', ''), ('Dinitry',
'Larry', ''), ('Stephen', 'King', ''), ('Larry', 'Olson', ''),
('Peter', 'Woom', ''), ('Tonny', 'Waka', ''), ('David',
'Ben', '') SELECT Apellidos FROM Empleados WHERE
```


Apellido like = "D%");

MySQL dispone de un optimizador de consultas y este utiliza distribución de claves almacenadas y otros factores para decidir el orden en que se deben unir las tablas y que índice se debe usar para una tabla específica.

Las estructuras de datos almacenan información entre objetos y relaciones entre ellas. Para ello utiliza el modelo de entidad relación (ER).

Un modelo -ER- se construye de manera gráfica mediante símbolos como:

- Rectángulos.
- Líneas.

Estas representan entidades en:

- Un proceso.
- Bajo características.
- Relaciones existentes.

Las relaciones tienen elementos, como, objetos, atributos, relaciones, tablas.

Tablas:

Las tablas son estructuras donde se almacenan los datos, sin estas, sería imposible el concepto de datos relacionales.

Para esto se deben considerar aspectos como:

- Cantidad.
- Número de columnas.
- Tipo de dato.

Para crear una tabla llamada Contacto_Empleados, la que debe contener las columnas

Identificación del empleado, número telefónico y correo electrónico, su instrucción es la Sgte.:

```
CREATE TABLE IF NOT EXISTS Contacto_Empleados
(identificación int AUTOINCREMENT
PRIMARY KEY (Id_Empleado), Numero_Tel varchar(50),
Email varchar(250)
);
```

Creación de usuarios:

Se pueden generar usuarios de diferentes maneras, pero requieren de un usuario Root o Superusuario bajo conexión al servidor, y bajo la tabla de usuarios en MySQL.

Permisología:

Al crear usuarios y asignarles permisos, se deben tomar en cuenta que tipo de recursos requeridos tendrá. Se pueden asignar permisos particulares a un usuario o con acceso completo.

Perfiles:

Estos van a repercutir al ejecutar diferentes funciones, se deben establecer permisos adecuados para no poner en riesgo la seguridad de los datos.

Mantenimiento:

MySQL cuenta con varias herramientas para mantener las tablas, y así:

- Analizar
- Optimizar
- Verificar
- Reparar Tablas

Dependiendo de el número de transacciones, sufrirán un

proceso de fragmentación y existe para ello un determinado comando:

OPTIMIZE nombre de la tabla;

También el comando para verificar integridad de las tablas, cuya sintaxis es:

CHECK TABLE nombre de la tabla;

Para crear un usuario que pueda acceder a ambas tablas, pero únicamente para ingresar y modificar registros, bajo un particular requerimiento, la instrucción es la Sgte.:

Crear usuario:

```
CREATE USER usuario_supervisor IDENTIFIED BY
PASSWORD 'fafarafa65';
```

Otorgar permisos:

```
GRANT INSERT, UPDATE ON Empleados,
Contacto_Empleados TO 'usuario_supervisor';
```

Otras instrucciones bajo requerimientos específicos:

Crear una vista de la base de datos Empleados que contenga el Nombre y Apellido de todos aquellos empleados que comiencen con la letra D.:

```
CREATE VIEW Temporal_Empleados AS SELECT Nombre,
Apellido FROM
Empleados WHERE Nombre LIKE 'D%';
```

Borrar la tabla Contacto_Empleados:

```
DROP TABLE Contacto_Empleados;
```

MySQL en el tiempo, ha generado una comunidad muy grande en Internet entre Developers, para el desarrollo de aplicaciones con esta tecnología.

Al realizar gestión de seguridad del sistema mediante MySQL, existen factores que deben ser tomados en cuenta al momento de construir un sistema robusto en seguridad. Es fundamental definir los permisos de usuarios, y tomar medidas a nivel físico en servidores de datos.

Es importante mantener políticas de seguridad apropiadas, que en sí son un conjunto de normas y directrices, que establecen como los datos pueden ser accedidos. Implica restricciones y el tipo de operación que se pueden llegar a ejecutar sobre las mismas. Es importante considerar:

- Passwords o claves de acceso son el primer frente de protección contra vulnerabilidades. Este proceso de creación de passwords implica:
 - Longitud mínima, debe contener un mínimo de 10 caracteres combinando letras y números.
 - Composición, como regla no debe tener datos personales o públicos.
 - Almacenamiento de passwords, tomando en cuenta los pasos anteriores se debe memorizar, nunca almacenar passwords.

- Control histórico, obliga a usuarios a no repetir passwords.
- Compartir Passwords, usuarios deben ser instruidos para que se comprenda la implicancia de la responsabilidad que recae en el usuario final.
- Transferencia electrónica, los passwords no deben ser transmitidos bajo ninguna circunstancia bajo internet.

Protección contra hackers, se debe combinar con el uso de buenas practicas de manera generalizada y se recomienda:

- Mantener BB.DD. detrás de un Firewall en hardware y software.
- Al usar Unix, nunca trabajar bajo usuario Root.
- Aplicar archivos de configuración para que mysql actúe de manera automática
- No almacenar variables en el entorno mysql_pwd
- Restringir acceso a tabla mysql.user
- Actualizar MySQL a su ultima versión ya que siempre incluyen mejoras.

Respaldo de datos, la utilidad de copias de seguridad es representada por su regularidad o frecuencia. MySQL tiene herramientas como MySQL Enterprise Backup.

Hacer un respaldo completo es necesario, pero no siempre conveniente, ya que generan grandes archivos y consumen tiempos de procesamiento, y no son efectivos, ya que todos los respaldos son siempre diferentes respecto a su copia anterior. Las copias de seguridad completa e incrementable son mas pequeñas y tardan menos tiempo. Para hacer estos cambios incrementales en el registro binario, el servidor debe iniciarse con la opción --log-bin para habilitar ese registro. Es posible indicarle a MySQL el cierre de registro binario actual usando la opción mysqladmin flush-logs.

MySQL tiene un comando simple para recuperar datos, la sintaxis es la siguiente:

```
Mysqldump -u "usuario" -p "contraseña"
```

```
<nombre_de_la_base_de_datos> <
```

```
nombre_del_respaldo.sql>;
```

Ejemplo:

```
mysqldumpo -u supervisor -p 1536c* finanzas
```

financiero.sql;

Para la recuperación de datos es fundamental un ambiente optimizado en la BB.DD. Esto implica maximizar la eficiencia y velocidad. También en los métodos de acceso a los servidores, datos y la recuperación a través de técnicas de diseño, análisis estadístico y de monitoreo.

Para lograr la máxima optimización se deben tomar en cuenta:

- Optimización del tamaño de los datos.
- Optimización del tipo de datos.
- Creación de índices.

Una auditoria a las bases de datos, significa monitorear acciones ejecutadas por los usuarios no configurados correctamente, así se detectan errores.

Para ello es fundamental establecer los privilegios a los usuarios. Así de tal manera, al crear privilegios a usuarios se podrán tener en consideración que estos solo pueden ejecutar ciertas operaciones, así se garantiza el excesivo uso de privilegios.

Es necesario para la seguridad, el monitorear y establecer acciones ejecutadas por los usuarios y que se puedan rastrear el acceso a los datos y verificar acciones sospechosas.

MySQL cuenta con el comando GRANT para definir parámetros para una optima auditoria de una BB.DD.

Parámetros como:

- GRANT ALL PRIVILEGES ON Empleados.* TO Juan; (Privilegios para tablas empleados a usuario Juan)
- Ej.2:
GRANT ALL PRIVILEGES *.* TO Juan; (Todos los privilegios sobre todas las bases de datos a Juan)

La encriptación es fundamental para la auditoria de datos, estos son datos convertidos a datos binarios, que resultan muy difíciles de leer.

Se requieren de estas auditorías para garantizar la integridad de los datos y afirmar que los cambios realizados son correctos y legales.

ⁱ IACC (2019). Síntesis de los documentos de estudio, Administración de Base de datos.

ⁱⁱ Síntesis de cierto material obtenido del sitio web <https://bit.ly/3aLxDS4>

ⁱⁱⁱ Todos los derechos reservados para Felipe Alfonso González López, Estudiante IP IACC Continuidad en Ingeniería en Informática – 2020 – 2021.