



# ADMINISTRACIÓN DE BASE DE DATOS

## UNIDAD 1 EL DBA

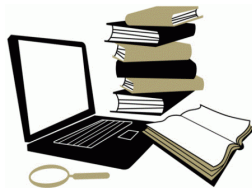
POR: MCES AMAURY CORIA RAMIREZ

Se dice que en la industria se usan los términos ADMINISTRACIÓN DE DATOS y ADMINISTRACION DE LA BASE DE DATOS.

En algunos casos se consideran sinónimos, para David Kroenke (Procesamiento de base de datos), ADMINISTRACION DE DATOS, se refiere a una función que se aplica a toda la empresa.

El término ADMINISTRACION DE LA BASE DE DATOS, se refiere a la función que es específica para una base de datos en particular, incluyendo las aplicaciones que la procesan.

- ◉ Las bases de datos varían en tamaño y alcance, desde una base de datos personal de un usuario único hasta una gran base de datos interorganizacional.
- ◉ Todas las bases de datos requieren de una administración, aunque las tareas varían en cuanto a su complejidad.
- ◉ Para aplicaciones multiusuario, la administración se vuelve más importante y difícil. Por lo tanto tiene reconocimiento formal.



- ◉ Mantener en funcionamiento una base de datos con docenas o centenas de usuarios requiere de mucho tiempo, así como conocimientos técnicos y técnicas diplomáticas que usualmente se manejan en una oficina de administración de base de datos.
- ◉ Al jefe de la oficina se le conoce como administrador de la base de datos o DBA.

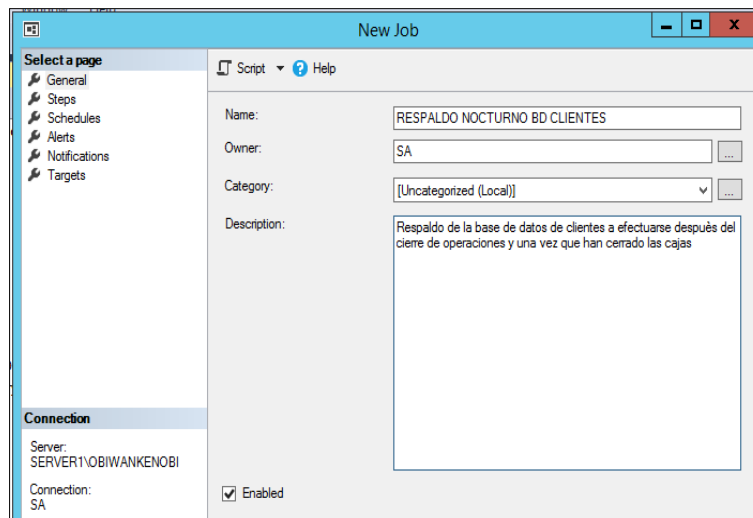
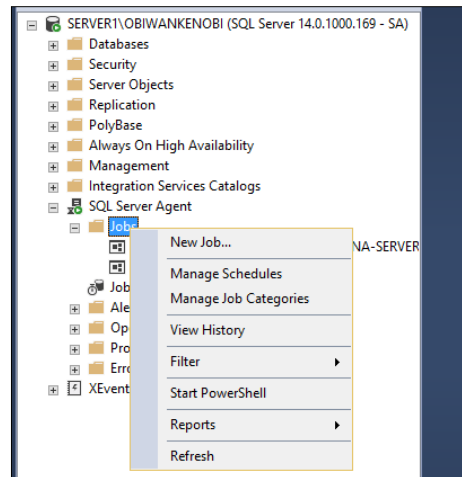


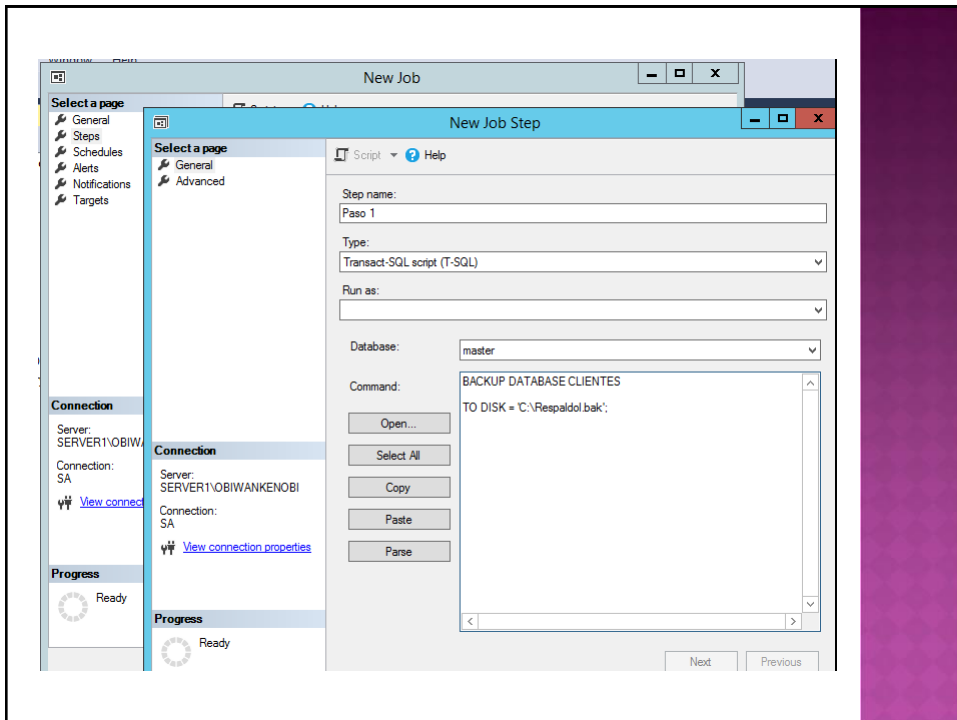
La responsabilidad del DBA es facilitar el desarrollo y el uso de la base de datos. Por lo general esto significa balancear las metas conflictivas de protección de la base de datos y maximizar su disponibilidad y beneficio para los usuarios. Es responsable del desarrollo, operación y mantenimiento de la base de datos y de sus aplicaciones.

## 1.- ADMINISTRACIÓN DE LA ESTRUCTURA DE LA BASE DE DATOS

- ◉ Participación en el diseño inicial de la base de datos y en su implementación lo cual implica:
  - Modelado a nivel lógico y físico
  - Análisis de espacio y crecimiento
  - Documentación (diccionario de datos, diagram E-R)
  - Configuración
  - Scripts
  - Pruebas en instancias de testing
  - Administración de cambios

## 2.- CONTROL DE PROCESOS REPETITIVOS





```
#!/bin/bash

USER="usuario"
PASSWORD="password123"
DATABASE="test"

FINAL_OUTPUT=eduardo/`date +%Y%m%d`_${DATABASE}.sql
mysqldump --user=$USER --password=$PASSWORD $DATABASE > $FINAL_OUTPUT
gzip $FINAL_OUTPUT
```

```
crontab -e
```

```
00 11 * * * /ruta/donde/se/encuentra/backup_database.sh
```

- Minutos (un rango de 0 al 59)
- Hora (En formato 24hrs)
- Días del mes (del 1 al ... depende del mes)
- Mes (1 al 12)
- Días de la semana (0 al 6, siendo el 0 domingo)
- Acción a ejecutar

### 3.- ADMINISTRACIÓN DE DERECHOS Y RESTRICCIONES

GRANT PUBLIC TO ALEXIS;

ALTER SERVER ROLE SECURITYADMIN ADD  
MEMBER ALEXIS;

GRANT SELECT ON CLIENTE TO ALEXIS;  
DENY DELETE ON CLIENTE TO ALEXIS;

SP\_HELPUSER ALEXIS;  
SP\_TABLE\_PRIVILEGES CLIENTE;

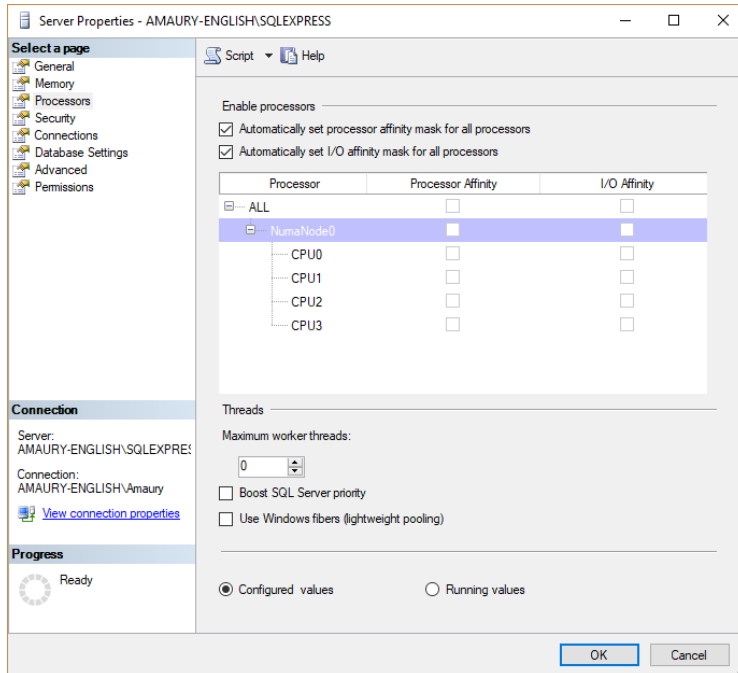
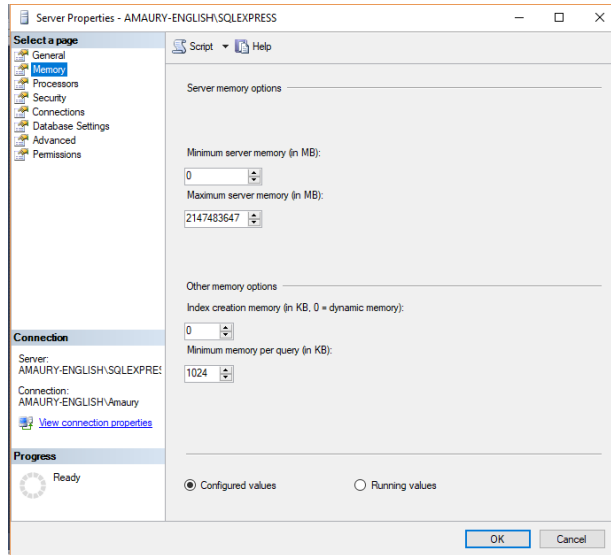
PROCESOS	DEPENDENCIAS			
	DIVISION FINANCIERA	SISTEMAS	CARTERA	CONTABILIDAD
Gestión de centros transaccionales	X	X	X	
Administración de sistemas		X		
Atención al cliente		X	X	
Conciliación de cuentas	X			X

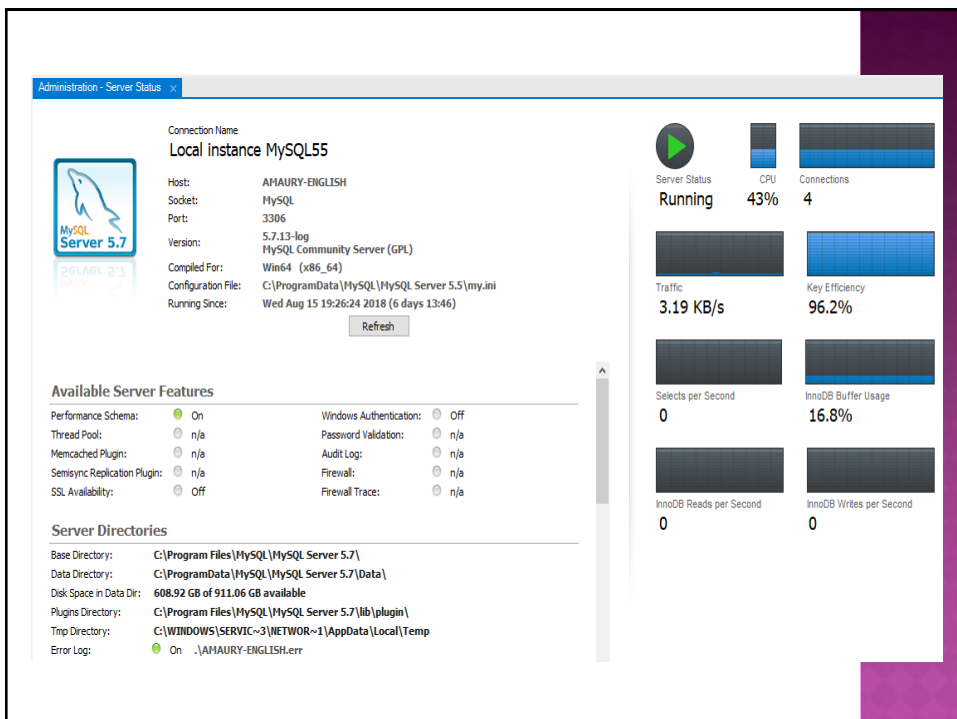
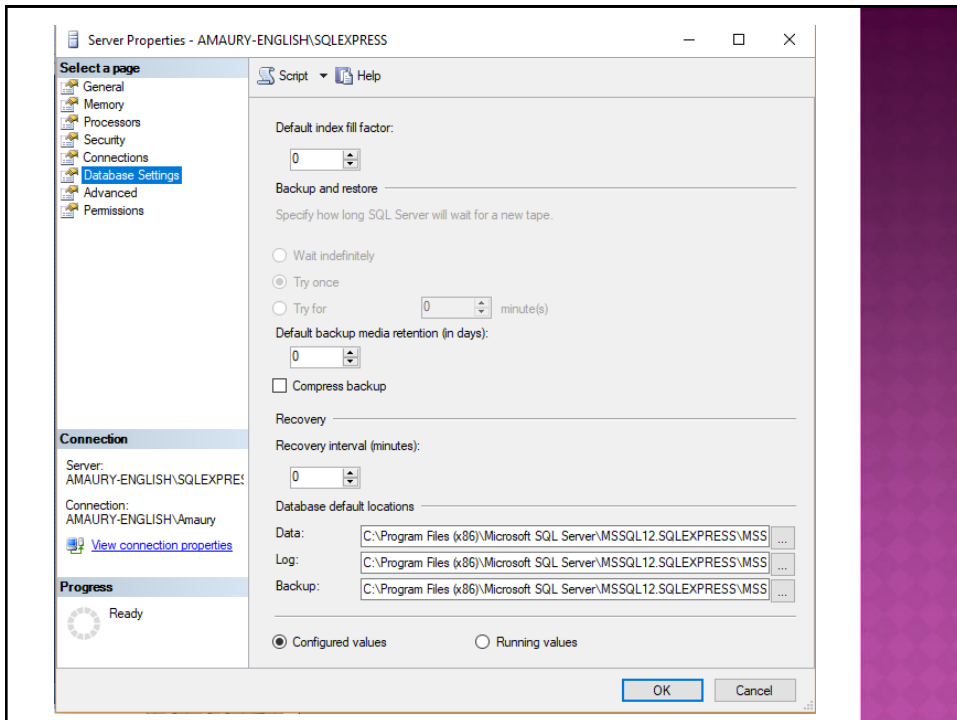
**S=SELECT, I=INSERT, U=UPDATE, D=DELETE**

#### CONTABILIDAD

	CLIENTE	COCHE	EMPLEADO	INVENTARIO	PROVEEDOR	PUESTO	VENTA
JEFE	S	S	S	S	S	S	S
ASESOR	S	S	S	SIUD	SIUD	S	SIUD

## 4.- RESPONSABILIDADES DEL PROCESAMIENTO Y MEMORIA







**SYSBENCH** [common-options] --test =name [test-options] command

Donde:

Command = prepare/run/cleanup/help

```
MYSQL> sysbench --test=cpu --cpu-max-prime=2000 run
```

```
MYSQL> sysbench --num-threads = 64 -test=memory run
```

## 5.- DESARROLLO DE LA SEGURIDAD DE LA BASE DE DATOS



## 6.- ACCESO A LA BASE DE DATOS Y SEGURIDAD A NIVEL DE APLICACIÓN



## 7.- ADMINISTRACIÓN DEL SGBD (DBMS)

- ◉ Licenciamiento.
- ◉ Documentación del proceso de instalación, desinstalación y actualización.
- ◉ Verificación de parches y actualizaciones.
- ◉ Documentación de rutas de instalación, servidor(es), direcciones IP, segmento de red, asegurar el acceso a la instancia de producción e instancias de pruebas.
- ◉ Permisos de acceso a servidores de producción y pruebas.

## 8.- MANTENIMIENTO DEL REPOSITORIO

- ◉ Definición inicial de rutas de archivos primarios, secundarios, tablespaces, logs, unidades de disco.
- ◉ Monitoreo constante del espacio y crecimientos y definir un porcentaje cuando se deba levantar una alerta.
- ◉ Estadísticos mensuales.
- ◉ Documentar procesos de attach y detach y de migración.

## 9.- MONITOREO

- ◉ SEGURIDAD
- ◉ QUERYS EN PROCESO
- ◉ INDEXACION

\* ver queries sql server

## 10.- MANTENIMIENTO

### -- SHRINK LOG

USE PRUEBA;

DBCC SHRINKFILE (HOTEL\_LOG, 1);

### --DBCC SHRINKDATABASE;

DBCC SHRINKDATABASE (DB, 10);

--(10% de espacio disponible)

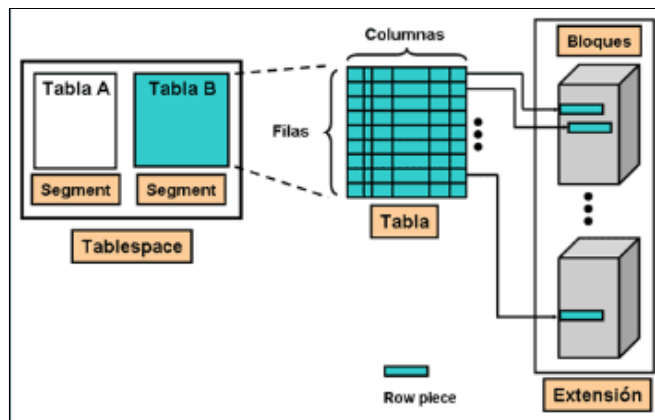
DBCC SHRINKDATABASE (DB, TRUNCATEONLY);

--(toma el último tamaño tomado o el que tenga como predefinido la base de datos)

### --Checar integridad de las páginas de datos

DBCC CHECKTABLE

('HumanResourcesDB.Employee');



```
ALTER INDEX IDX_Employee  
ON HumanResources.Employee  
REORGANIZE;
```

```
ALTER INDEX ALL ON  
HumanResources.Employee  
REORGANIZE;
```

```
ALTER INDEX PK_Employee ON  
HumanResources.Employee  
REBUILD;
```

```
DBCC DBREINDEX ('HumanResources.Employee',  
PK_Employee,80);
```

--80% de espacio utilizado en las páginas de los índices)

## ◉ Limpieza de logs binarios Mysql

Si tienes activados los logs binarios, estos a la larga ocuparán varios Mb de nuestro disco y generalmente no tiene sentido mantenerlos.

Para “purgar” los logs binarios podemos hacerlo de una forma pasiva (modificando la configuración del servidor) o bien, de manera activa, ejecutando la instrucción PURGE MASTER LOGS desde el prompt de MySQL.

La forma pasiva, consiste en que MySQL limpie por su cuenta los logs binarios. Para hacer ésto, basta agregar al archivo de configuración my.cnf: `expire_logs_days = 7`

De forma activa, desde un prompt de MySQL puedes ejecutar, por ejemplo:

```
PURGE MASTER LOGS TO 'bin-log.000012';
PURGE MASTER LOGS BEFORE '2008-07-23 00:00:00'
```

Evita eliminar los logs binarios borrando directamente el archivo desde el sistema operativo (`rm -f binlog.*`).

Presta especial atención al purgar los logs, pues si la base de datos está siendo replicada y los logs a “purgar” aun no han sido recogidos por el SLAVE, el mismo puede perderse de datos.

## ◉ Optimización de tablas Mysql

Las tablas, al fin y al cabo, son archivos (dependiendo del motor de almacenamiento, uno o más archivos, un archivo único, etc.).

Como archivo, al cabo de varias operaciones de manipulación de datos (INSERT, UPDATE, DELETE) se puede generar fragmentación, espacio inutil, etc.

Por tal razón, desde el prompt de MySQL puedes ejecutar la instrucción `OPTIMIZE TABLE nombre_de_tabla` para que MySQL haga lo suyo con los archivos.

### ◉ Vaciamiento de caché de consultas

La caché de consultas almacena los resultados de una consulta, así el servidor responde más rápido si algún otro cliente los requiere.

La cache de consultas también es susceptible a fragmentación, en especial cuando se ejecutan SELECTs con resultados de diferente tamaño.

Por lo tanto, no está de más, que frecuentemente se vacíe esta cache de consultas.

Para hacerlo, basta ejecutar la instrucción `FLUSH QUERY CACHE`.

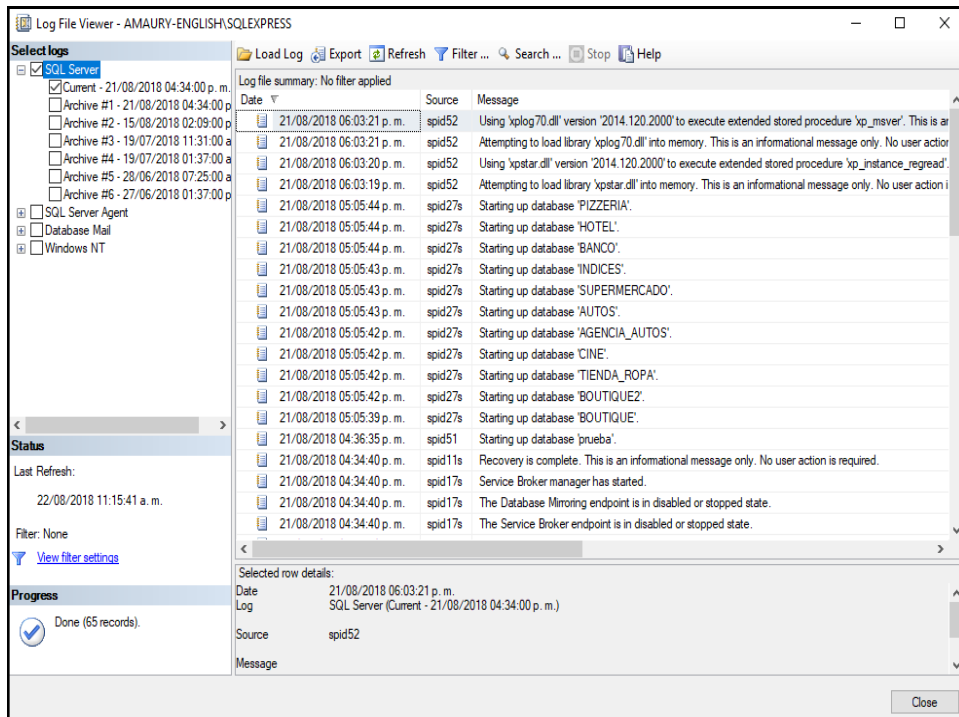
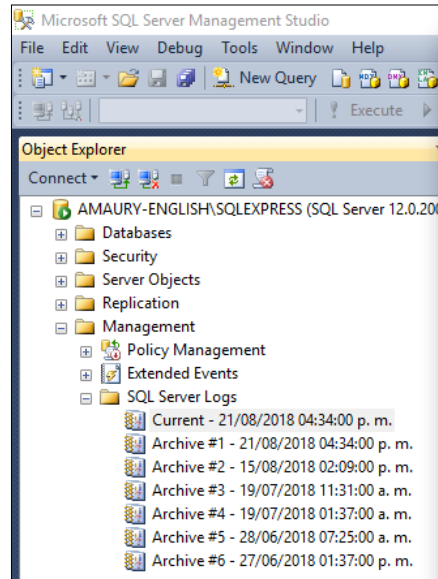
### ◉ Rotación de logs binarios Mysql

MySQL crea un nuevo archivo de log binario cada vez que es reiniciado o cada vez que se llegue a un tamaño máximo que se puede especificar en el `my.cnf`, por ejemplo: `max_binlog_size=4M`.

A veces, puedes querer rotar los logs por tiempo en vez de por tamaño.

Para rotar los tamaños por tiempo, puedes crear un script a ejecutar desde el cron que ejecute simplemente un `FLUSH LOGS`.

## 11.- AUDITORÍA DE LOS LOGS





```
C:\Program Files\MySQL\MySQL Server 5.7\bin>mysql -h localhost -u root -p
Enter password: *****
```

```
mysql> set global general_log=ON
-> ;
Query OK, 0 rows affected (0.18 sec)
```

```
mysql> describe mysql.general_log;
```

Field	Type	Null	Key	Default
event_time	timestamp(6)	NO		CURRENT_TIMESTAMP(6)
user_host	mediumtext	NO		NULL
thread_id	bigint(21) unsigned	NO		NULL
server_id	int(10) unsigned	NO		NULL
command_type	varchar(64)	NO		NULL
argument	mediumblob	NO		NULL

```
mysql> select * from mysql.general_log order by event_time desc limit 5;
Empty set (0.02 sec)
```

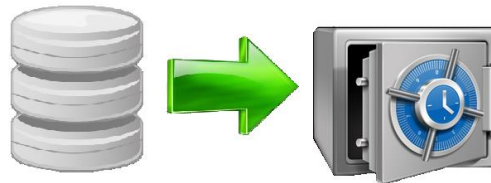
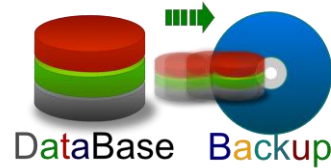
```
mysql> select * from mysql.general_log where user_host like '%192.168.%' order by event_time;
Empty set (0.00 sec)
```

```
mysql> set global general_log=OFF;
Query OK, 0 rows affected (0.15 sec)
```

```
mysql> truncate table mysql.general_log;
Query OK, 0 rows affected (0.70 sec)
```

## 12.- RESPALDO Y RESTAURACIÓN

- ◉ Frecuencia
- ◉ Horario
- ◉ Rotación
- ◉ Bitácora
- ◉ Tareas automatizadas
- ◉ Caja fuerte
- ◉ Sitio alternativo



```
BACKUP DATABASE HOTEL TO DISK = 'c:\HOTEL_ddmmaa_hhmm.bak';
```

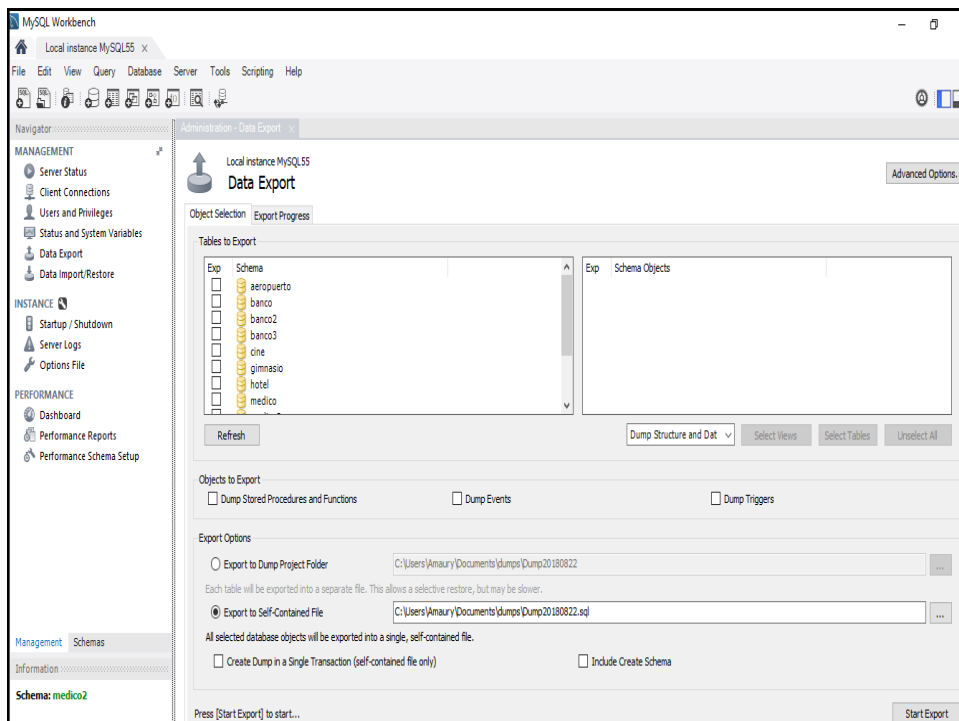
```
BACKUP DATABASE PRUEBA  
TO DISK = 'C:\RespaldoDiferencial_ddmmaa.bak'  
WITH DIFFERENTIAL;
```

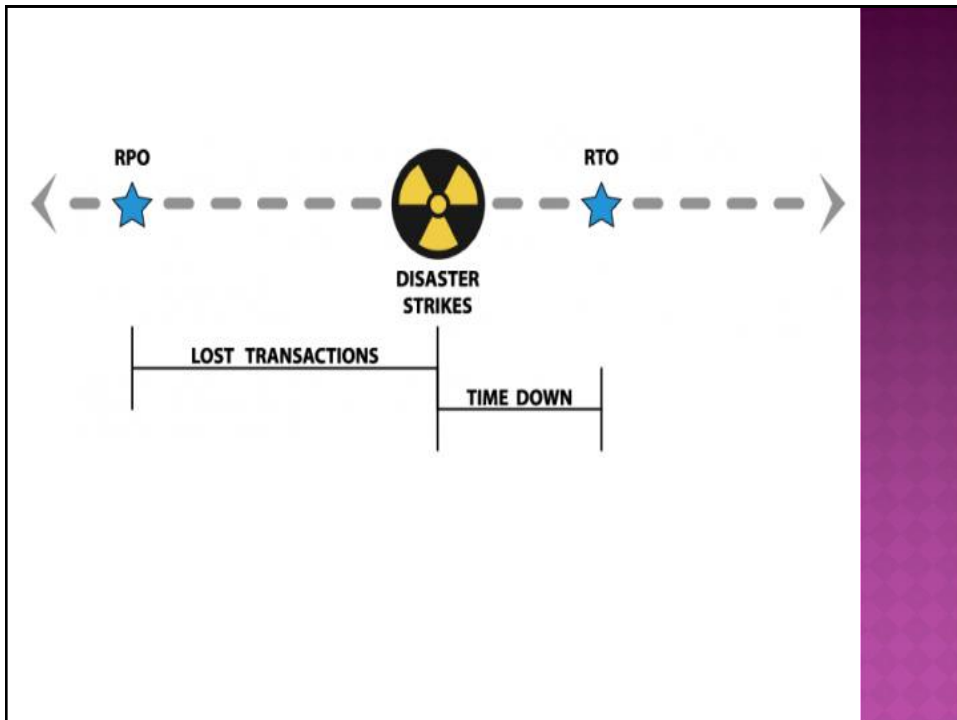
```
BACKUP LOG HOTEL TO HOTEL_LOG_ddmmaa_hhmm.bak;
```

```
USE master;  
RESTORE DATABASE prueba  
FROM DISK = 'C:\HOTEL_ddmmaa_hhmm.bak'  
WITH FILE = 1,  
MOVE 'HOTEL' TO 'C:\prueba.mdf',  
MOVE 'HOTEL_log' TO 'C:\prueba.ldf',  
NOUNLOAD, REPLACE;
```

```
RESTORE LOG HOTEL FROM DISK = 'C:\HOTEL_LOG_ddmmaa_hhmm.bak'  
WITH RECOVERY;
```

**mysqldump [opciones] nombre\_base\_datos**





## 13.- CONCURRENCIA

- ◉ Manejo de transacciones
- ◉ Llaves primarias
- ◉ Índices
- ◉ Propiedades ACID:
  - Atomic
  - Consistent
  - Isolated
  - Durable

## 14.- BLOQUEOS

- ◉ Exclusivo, compartido, deadlock.

USUARIO A	USUARIO B
Aplicar lock al papel	Aplicar lock a los lápices
Tomar el papel	Tomar los lápices
Aplicar lock a los lápices	Aplicar lock al papel

Aplicar lock al papel para el usuario A
Aplicar lock a los lápices para el usuario B
Procesar la solicitud del usuario A; escribir el registro de papel
Procesar la solicitud del usuario B; escribir el registro de lápices
Poner al usuario A en espera
Poner al usuario B en espera
¡Bloqueo!

## 15.- PROBLEMAS DE PÉRDIDA DE INFORMACIÓN

- ◉ Correcciones manuales o automatizadas
- ◉ Correcciones masivas

## 16.- MIGRACIONES

- ◉ De tabla a tabla
- ◉ De una base de datos a otra base de datos
- ◉ A una base de datos de producción
- ◉ A una base de datos de consultas
- ◉ A una base de datos de pruebas
- ◉ Base de datos distribuída
- ◉ Implicaciones en llaves primarias y autoincrementables