

## Sesión 9: Monitorización avanzada en MySQL

### 1.1. ¿Qué significa monitorizar MySQL?

Monitorizar MySQL es **medir y observar** cómo se comporta el servidor para:

- Detectar **consultas lentas**.
- Saber si el cuello de botella está en **CPU, disco, RAM o bloqueos**.
- Ver qué tablas y consultas consumen más recursos.
- Tomar decisiones de **optimización** (índices, diseño de tablas, configuración).

En MySQL tenemos tres “piezas clave” para una monitorización avanzada:

1. **EXPLAIN / EXPLAIN ANALYZE**: cómo se ejecuta una consulta.
2. **performance\_schema**: métricas muy detalladas a bajo nivel.
3. **sys schema**: vistas “bonitas” que resumen performance\_schema.

Y además:

- **slow\_query\_log**: registro de consultas que tardan más de X segundos.

### 1.2. EXPLAIN y EXPLAIN ANALYZE

EXPLAIN muestra el **plan de ejecución** de una consulta: qué índices usa, en qué orden se leen las tablas, si hace escaneos completos, etc.

Ejemplo:

```
EXPLAIN SELECT * FROM clientes WHERE email = 'a@b.com';
```

Campos importantes:

- **type**: tipo de acceso (ALL, index, ref, const...).
- **possible\_keys**: índices que podrían usarse.
- **key**: índice que realmente usa.
- **Rows**: filas estimadas a leer.

### 1.3. slow\_query\_log

Es un log especial donde MySQL guarda **consultas lentas**.

Parámetros clave:

- **slow\_query\_log:** activar/desactivar.
- **long\_query\_time:** umbral en segundos para considerar “lenta”.
- **log\_output:** dónde se guardan (archivo o tabla).

Ejemplo típico:

```
SET GLOBAL slow_query_log = 1;  
SET GLOBAL long_query_time = 0.5; -- consultas > 0,5 s se logean  
SET GLOBAL log_output = 'TABLE'; -- se guardan en mysql.slow_log
```

Con esto luego puedes hacer:

```
SELECT * FROM mysql.slow_log ORDER BY start_time DESC;
```

### 1.4. performance\_schema

Es una **base de datos interna** que MySQL usa para registrar:

- Tiempos de ejecución de consultas.
- Esperas (waits) en disco, locks, red, etc.
- Estadísticas por tabla, por índice, por usuario...

No se usa directamente para el día a día porque es **muy técnica**, pero es la base de todo.

### 1.5. sys schema

Es un esquema que se apoya en performance\_schema y te da **vistas amigables**:

- **sys.statement\_analysis:** ranking de consultas por tiempo total, número de ejecuciones, etc.
- **sys.schema\_table\_statistics:** tablas con más lecturas/escrituras.
- **sys.user\_summary:** resumen por usuario.
- **sys.waits\_by\_host\_by\_event\_type:** esperas más importantes.

Es algo así como un **“mini Statspack / AWR” para MySQL**, ya montado.