

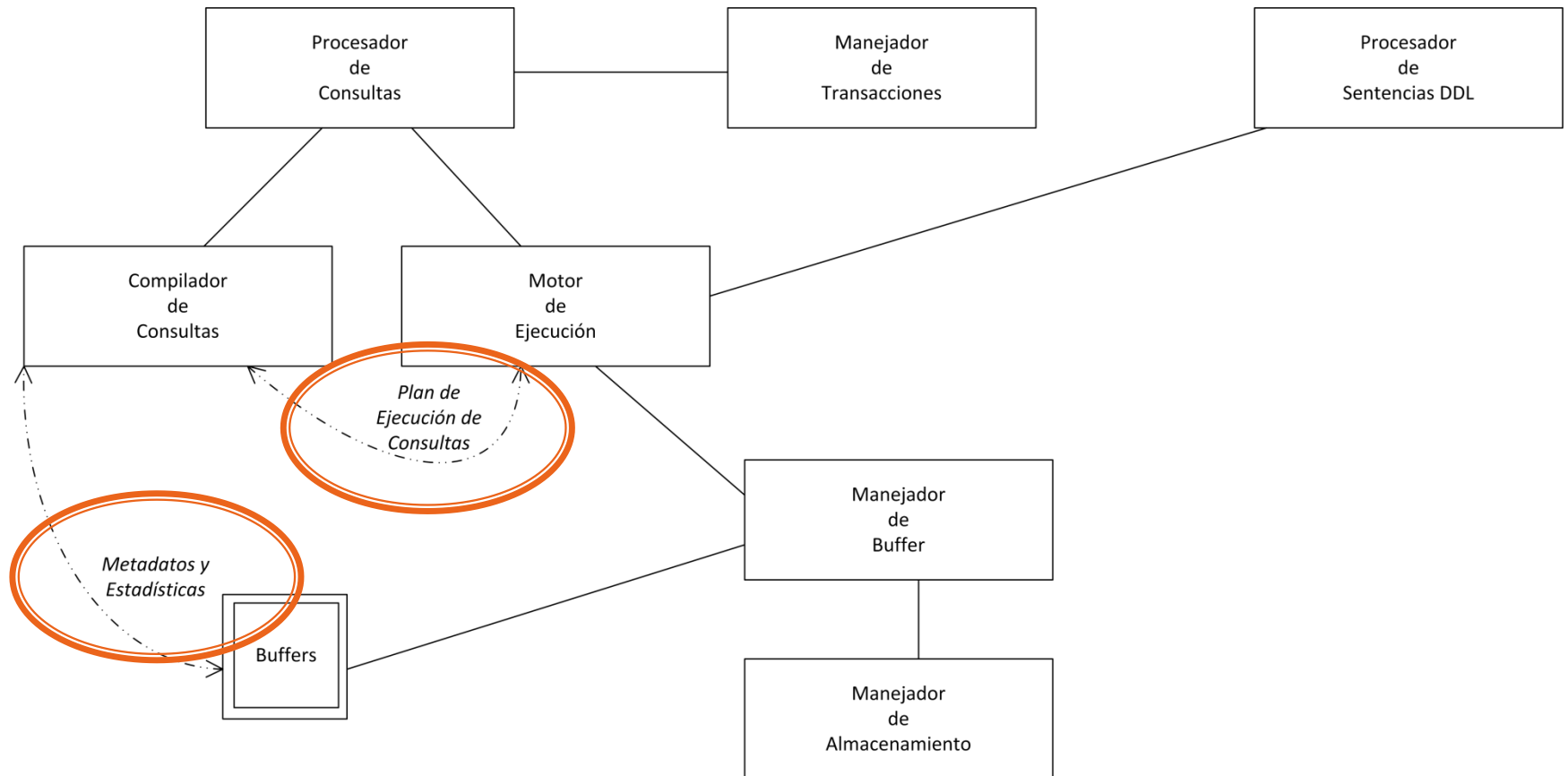
# Bases de Datos 1

Alejandra Lliteras

[alejandra.lliteras@lifa.info.unlp.edu.ar](mailto:alejandra.lliteras@lifa.info.unlp.edu.ar)

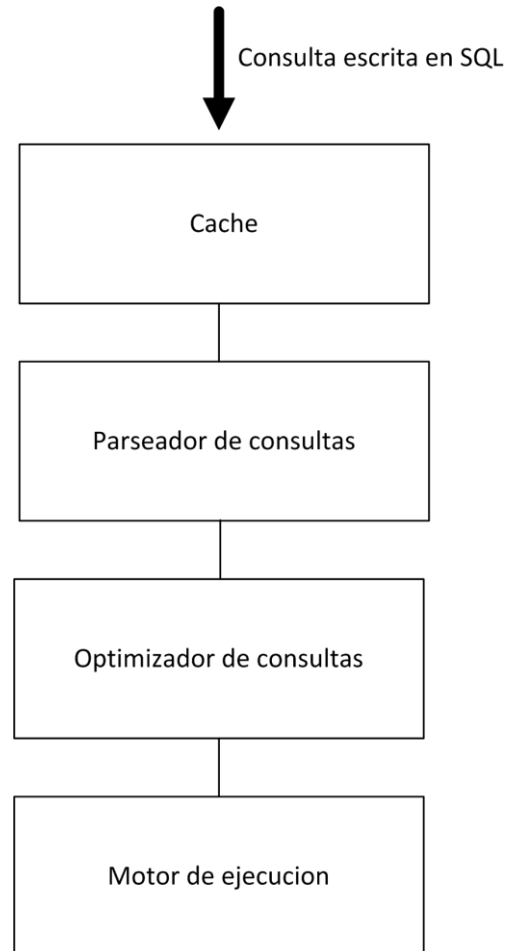


# Plan de Ejecución



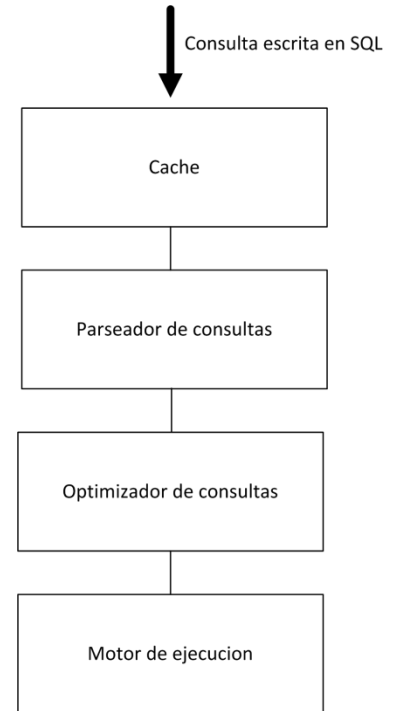
# MySQL

# Procesador de consultas



# Procesador de consultas

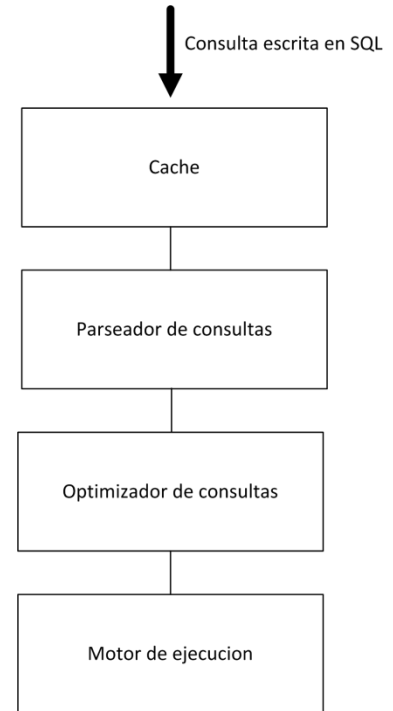
- ▶ Ante una consulta que llega al procesador
  - ▶ Si esta en la caché, entonces retorna los resultados de la misma
  - ▶ Si no esta en la caché
    - ▶ La parsea
    - ▶ Intenta optimizarla
    - ▶ La Ejecuta



# Procesador de consultas

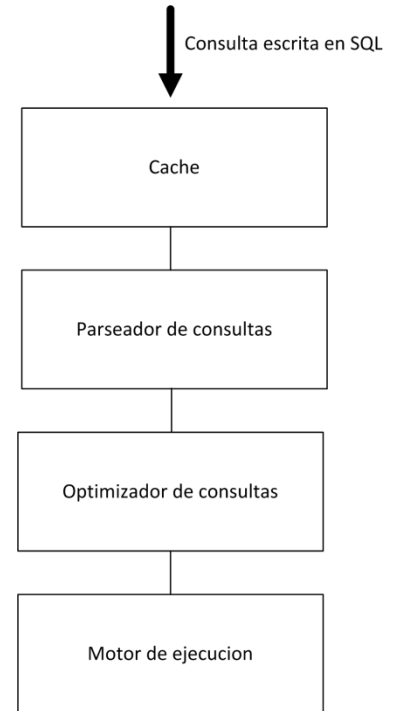
## ► Cache

- Se habilita o deshabilita su uso
- En caso de estar habilitado su uso, las operaciones de insert, delete y update, borran la cache



# Procesador de consultas

- ▶ Optimizador
  - ▶ Usa el plan de ejecución
  - ▶ Contempla las estadísticas de las tablas involucradas



# Optimización de consultas





# Optimización de consultas

- ▶ Con el uso de las aplicaciones que emplean bases de datos relacionales:
  - La cantidad de tuplas en las tablas crecen
  - La cantidad de usuarios que acceden a la misma se incrementa


Esto puede conllevar a la degradación en la performance de las consultas

# Análisis de consultas

- ▶ Análisis (temporal) de que consultas demoran mas de un cierto tiempo
  - Analizar el log de consultas lentas, para ello
    - Detener el servicio de MySQL 5.7
    - Acceder al archivo my.ini (por ejemplo, en ProgramData/MySQL)
      - Verificar que no este con permisos de sólo lectura
      - Activar el registro de log
      - Setear el tiempo máximo de tolerancia
      - Incluir sentencia para que registre consultas sin índices
      - Guardar cambios en el archivo
    - Iniciar el servicio de MySQL 5.7

# my.ini

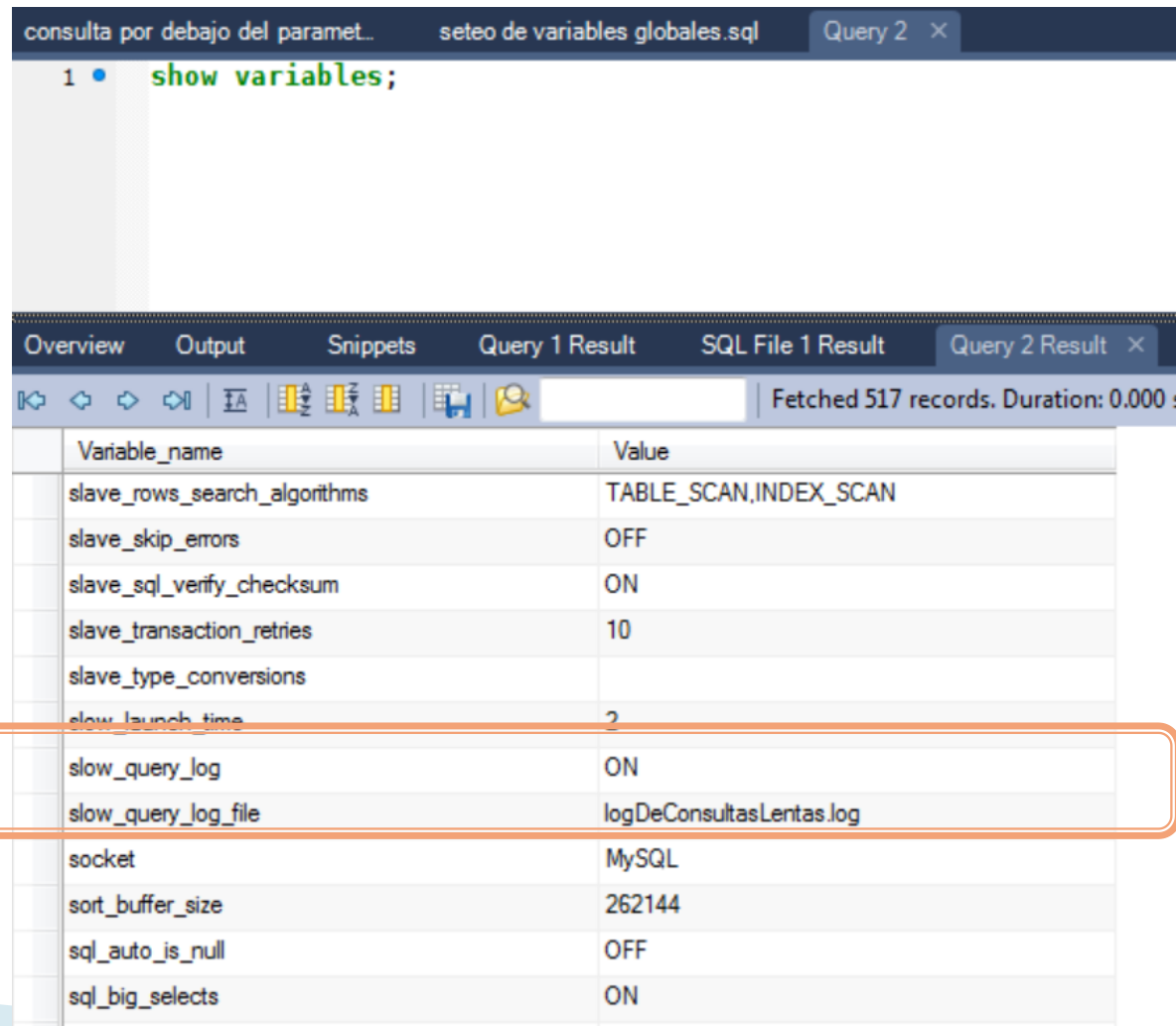
```
...  
# SERVER SECTION  
...  
[mysqld]  
...  
# General and Slow logging.  
log-output=FILE  
general-log=0  
general_log_file="miLogDeError.log"  
slow-query-log=1 # 1=true, 0=false  
slow_query_log_file="logDeConsultasLentas.log"  
long_query_time=0.05 #consultas que tarden mas de 0.05  
                        milisegundos  
log-queries-not-using-indexes  
...
```



# Análisis de consultas

- ▶ Corroborar el valor de las variables previamente seteadas.

- show variables;



consulta por debajo del paramet... seteo de variables globales.sql Query 2 x

```
1 • show variables;
```

Overview Output Snippets Query 1 Result SQL File 1 Result Query 2 Result x

Fetches 517 records. Duration: 0.000

Variable_name	Value
slave_rows_search_algorithms	TABLE_SCAN,INDEX_SCAN
slave_skip_errors	OFF
slave_sql_verify_checksum	ON
slave_transaction_retries	10
slave_type_conversions	
slow_launch_time	2
slow_query_log	ON
slow_query_log_file	logDeConsultasLentas.log
socket	MySQL
sort_buffer_size	262144
sql_auto_is_null	OFF
sql_big_selects	ON

# Análisis de consultas

## ► Leer el archivo de log creado

- logConsultasLentas.log (por ejemplo, en ProgramData/MySQL)

```
1 C:\Program Files (x86)\MySQL\MySQL Server 5.7\bin\mysqld.exe, Version: 5.7.12-log (MySQL Community Server (GPL)). started with:
2 TCP Port: 1024, Named Pipe: (null)
3 Time Id Command Argument
4 # Time: 2016-11-08T23:08:33.319558Z
5 # User@Host: root[root] @ localhost [127.0.0.1] Id: 4
6 # Query_time: 0.062400 Lock_time: 0.031200 Rows_sent: 1000 Rows_examined: 253
7 SET timestamp=1478646513;
8 select *
9 from sakila.actor join sakila.address join sakila.customer join sakila.payment join sakila.film join sakila.film_actor
10 where sakila.actor.first name like "%a%"
11 LIMIT 0, 1000;
12 C:\Program Files (x86)\MySQL\MySQL Server 5.7\bin\mysqld.exe, Version: 5.7.12-log (MySQL Community Server (GPL)). started with:
13 TCP Port: 1024, Named Pipe: (null)
14 Time Id Command Argument
15 # Time: 2016-11-09T10:34:26.442422Z
16 # User@Host: root[root] @ localhost [127.0.0.1] Id: 4
17 # Query_time: 0.078000 Lock_time: 0.000000 Rows_sent: 4 Rows_examined: 4
18 SET timestamp=1478687666;
19 SELECT 'table' AS 'OBJECT_TYPE', TABLE_CATALOG as 'CATALOG', TABLE_SCHEMA as 'SCHEMA', TABLE_NAME as 'NAME' FROM information_schema.
20 # Time: 2016-11-09T10:34:26.442422Z
21 # User@Host: root[root] @ localhost [127.0.0.1] Id: 4
22 # Query_time: 0.000000 Lock_time: 0.000000 Rows_sent: 0 Rows_examined: 0
23 SET timestamp=1478687666;
24 SELECT 'view' AS 'OBJECT_TYPE', TABLE_CATALOG as 'CATALOG', TABLE_SCHEMA as 'SCHEMA', TABLE_NAME as 'NAME' FROM information_schema.v
25 # Time: 2016-11-09T10:34:26.504822Z
26 # User@Host: root[root] @ localhost [127.0.0.1] Id: 4
27 # Query_time: 0.062400 Lock_time: 0.062400 Rows_sent: 7 Rows_examined: 60
28 SET timestamp=1478687666;
29 SELECT ROUTINE_TYPE AS 'OBJECT_TYPE', ROUTINE_CATALOG as 'CATALOG', ROUTINE_SCHEMA as 'SCHEMA', ROUTINE_NAME as 'NAME' FROM informat
30 |
```

# Análisis de consultas

## ► Leer el archivo de log creado

- logConsultasLentas.log (por ejemplo, en ProgramData/MySQL)

```
4 # Time: 2016-11-08T23:08:33.319558Z
5 # User@Host: root[root] @ localhost [127.0.0.1] Id: 4
6 # Query_time: 0.062400 Lock_time: 0.031200 Rows_sent: 1000 Rows_examined: 253
7 SET timestamp=1478646513;
8 select *
9 from sakila.actor join sakila.address join sakila.customer join sakila.payment
►10 |join sakila.film join sakila.film_actor
11 where sakila.actor.first_name like "%a%"
12 LIMIT 0, 1000;
```

# Análisis de consultas

- ▶ Leer el archivo de log creado
  - logConsultasLentas.log (por ejemplo, en ProgramData/MySQL)

```
4 # Time: 2016-11-08T23:08:33.319558Z
5 # User@Host: root[root] @ localhost [127.0.0.1] Id: 4
6 # Query_time: 0.062400 Lock_time: 0.031200 Rows_sent: 1000 Rows_examined: 253
7 SET timestamp=1478646513;
8 select *
9 from sakila.actor join sakila.address join sakila.customer join sakila.payment
▶10 |join sakila.film join sakila.film_actor
11 where sakila.actor.first_name like "%a%"
12 LIMIT 0, 1000;
```

# Análisis de consultas

- ▶ Algunos tips sobre el archivo de log creado
  - Detectar aquellas consultas que mas se repiten en el log y que peores tiempos arrojan
  - Si una consulta aparece solo una vez, puede deberse a que la lentitud de su ejecución se deba a un factor externo (por ejemplo, problemas de red)
  - Tener en cuenta que el log es un archivo de texto, puede ser recomendable limitar su activación a actividades de monitoreo puntuales



# Análisis de consultas

- ▶ Una vez detectada la o las consultas críticas
  - Trabajar sobre cada una de ellas
    - Analizar su plan de ejecución
      - EXPLAIN

```
1 | EXPLAIN SELECT * FROM categoriesG
```

```
***** 1. row *****
      id: 1
    select_type: SIMPLE
        table: categories
         type: ALL
possible_keys: NULL
         key: NULL
        key_len: NULL
         ref: NULL
         rows: 4
       Extra:
1 row in set (0.00 sec)
```

- **Información obtenida a partir del comando “explain”**

- **Id:** es un identificador secuencial para cada uno de los select del query.
- **Select\_type:** indica el tipo de select
  - **Simple:** se trata de un simple select.
  - **Primary:** es el select más externo.
  - **Derived:** el select es parte de un subquery en el from.
  - **Subquery:** es el primer select en un subquery.
  - **Dependent subquery:** el select es un subquery que depende de uno más externo.
  - **Uncacheable subquery:** el subquery no se almacena en la caché.
  - **Union:** el select es la segunda parte de un union.
  - **Dependent union:** es la segunda parte de un union que depende de un query más externo.
  - **Union result:** el select es el resultado de un union.

- **Información obtenida a partir del comando “explain”**

- Table: es la tabla a la que se hace referencia.
- Type: indica como une las tablas MySQL
  - System: la tabla tiene 0 o 1 sola fila.
  - Const: la tabla tiene una sola fila que coincide y además está indizada.
  - Eq\_ref: se utilizan todas las partes de un índice.
  - Ref: todas las filas que coinciden con un campo del índice de otra tabla son leídas
  - index\_merge: el join utiliza una serie de índices
  - Range: un índice se utiliza para encontrar filas dentro de un rango.
  - All: todas las filas son leídas.

- **Información obtenida a partir del comando “explain”**

- Possible\_keys: muestra las claves que podrían ser utilizadas para encontrar los resultados.
- Key: indica que clave es utilizada por Mysql.
- REF: muestra que columnas son utilizadas en las comparaciones con los campos de KEY.
- Rows: muestra la cantidad de filas examinadas.
- Extra: muestra posible información útil por ejemplo (using filesort).
- Explain extended:
  - Show warnings;

---

```
1 | EXPLAIN SELECT * FROM categoriesG
```

---

```
***** 1. row *****
      id: 1
    select_type: SIMPLE
      table: categories
        type: ALL
possible_keys: NULL
          key: NULL
      key_len: NULL
         ref: NULL
        rows: 4
      Extra:
1 row in set (0.00 sec)
```

---

- ▶ Que decide el planificador?
  - Qué índices usar?
    - Recordemos que los tipos de índices son
      - B-Tree
      - Hash
      - R-Tree
      - Textuales
  - En que orden realizar los joins

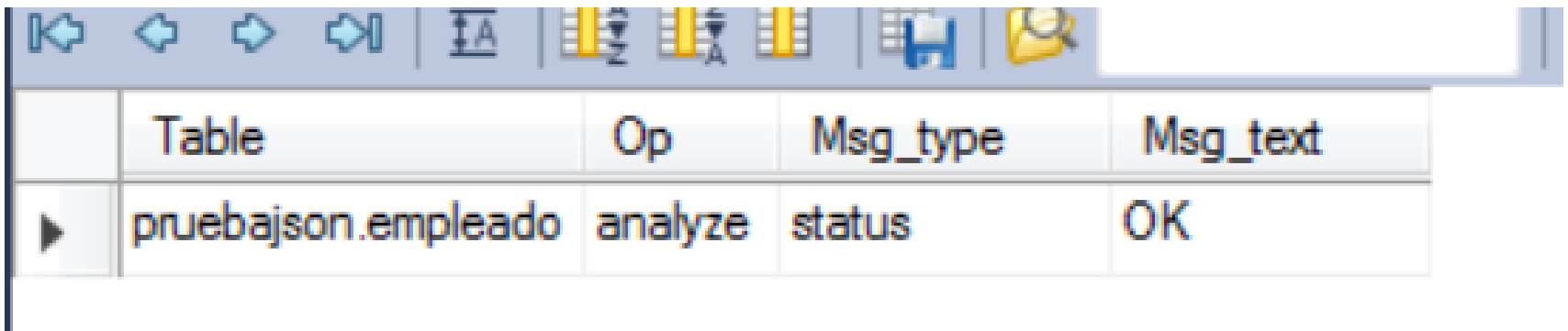
## ► Que decide el planificador?

### ◦ Qué índices usar?

- Para elegir el índice que considera mejor de todos los posibles de una tabla, consulta las estadísticas
- Tener en cuenta que las mismas deben ser actualizadas con cierta periodicidad para evitar la degradación en la performance de las consultas
- ANALYZE TABLE
- OPTIMIZE TABLE

## ► ANALYZE TABLE

ANALYZE TABLE  
PRUEBAJSON.EMPLEADO



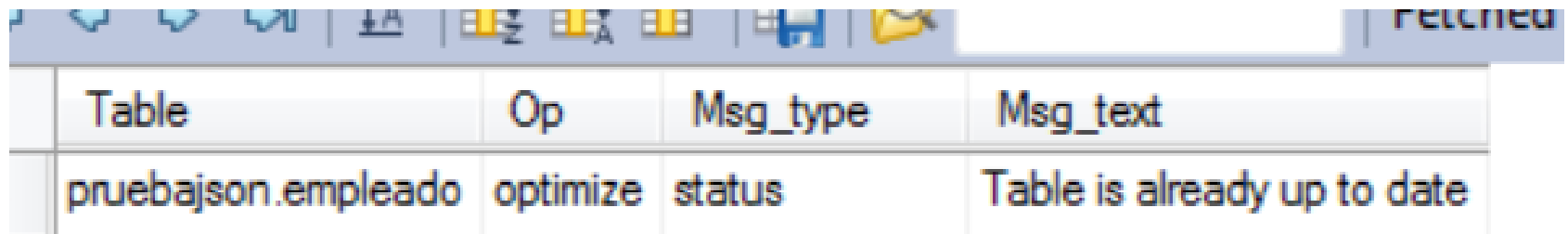
The screenshot shows a database management tool interface. At the top, there is a toolbar with various icons for navigation and editing. Below the toolbar is a table with the following data:

	Table	Op	Msg_type	Msg_text
►	pruebasjson.empleado	analyze	status	OK



## ► OPTIMIZE TABLE

OPTIMIZE TABLE  
PRUEBAJSON.EMPLEADO



The screenshot shows a database management tool interface. At the top, there is a toolbar with various icons for database operations. Below the toolbar, a table displays the results of an SQL command. The table has four columns: 'Table', 'Op', 'Msg\_type', and 'Msg\_text'. The first row of data shows the table 'pruebasjson.empleado' with the operation 'optimize', a status message, and the text 'Table is already up to date'.

Table	Op	Msg_type	Msg_text
pruebasjson.empleado	optimize	status	Table is already up to date

## ▶ IMPORTANTE

- ▶ Al crear un stored procedure y ejecutarlo, este ya queda compilado, por ello, las estadísticas que se actualicen luego de esto no serán contempladas en sus ejecuciones posteriores
  - En este caso, se debe recompilar el SP para asegurar que tome nuevas estadísticas y se mantenga optimizado.