

EVOLUCIÓN DE LAS BASES DE DATOS

CLASE SEMANA 5 DEL 20 Y 21-05

MARÍA DOLORES COSTA

IMPORTANTE: LOS DATOS POR SÍ SOLO NO TIENEN SIGNIFICADO HASTA QUE SE LES DA UN CONTEXTO

La BD (Base de Datos) es

- ✓ un gran almacén de datos que se define y se crea una sola vez
- ✓ puede ser accedida por varios usuarios aún al mismo tiempo (*concepto de concurrencia*),
- ✓ posibilita el acceso a todos los departamentos que posean los permisos necesarios en una organización.

La BD no solo contiene los datos de dicha organización sino también almacena una **descripción de dichos datos**.

Esta descripción se denomina **Metadatos** y se almacena en un **Diccionario de Datos** y es lo que permite que exista **independencia de datos física y lógica**.



EVOLUCIÓN DE LAS BASES DE DATOS



EVOLUCIÓN DE LAS BASES DE DATOS

- Fichas en mobiliarios
- Organizadas por orden alfabético
- Procedimientos manuales
- Impedían la concurrencia, accesibilidad a la información
- En caso de siniestros, se perdía la información
- Aprox hasta 1960

Ficheros

Archivos
magnéticos

- Tarjetas perforadas
- Tabulador electromagnético

- Tabla relacional
- Tablas relacionales de objetos
- Programación orientada a objetos
- Servidores
- Desde 1970

Base de datos

MANIPULACIÓN DE BASE DE DATOS

Podemos reconocer al menos 4 niveles jerárquicos que desempeñan distintos roles en cuanto a la intervención de una BD



ADMINISTRADOR DE LA BASE DE DATOS

Se encarga de

1. La administración física y lógica de la DB
2. Establece el tipo de datos e índices a implementar
3. Toma decisiones en cuanto a la ubicación física de los datos
4. Es el encargado de la política de seguridad
5. Es el responsable del acceso concurrente de los datos

DISEÑADOR DE LA BASE DE DATOS

1. Realiza el diseño de la DB
2. Identifica los datos. Definir los tipos de datos, índices y estructuras de almacenamiento a implementar.
3. Genera las relaciones entre los datos
4. Advierte las restricciones sobre los datos y sobre sus relaciones

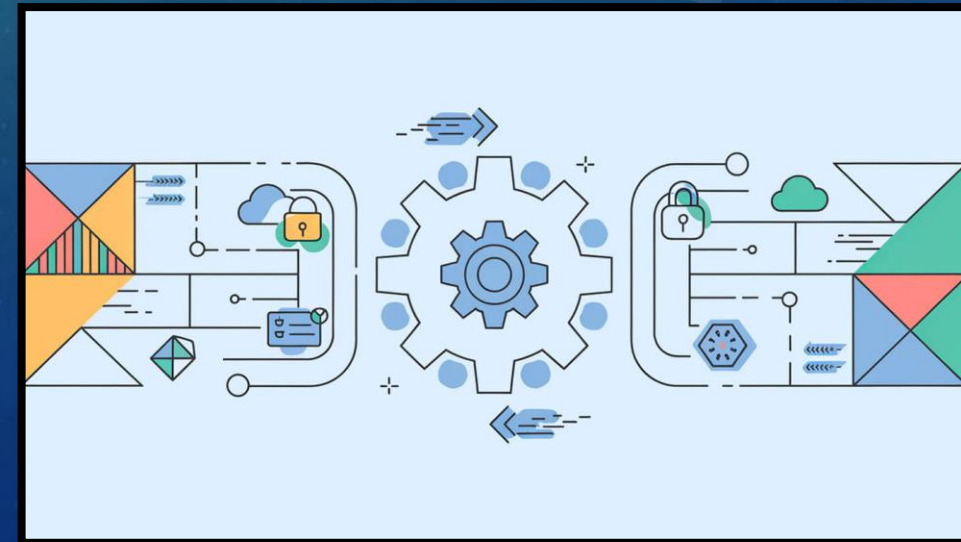
El **diseñador** debe tener un profundo conocimiento de los datos de la Organización, Empresa, Proyecto o Emprendimiento sobre el que está trabajando:

- Conocer las reglas del negocio
- Comportamiento de los datos
- Funcionamiento de la empresa



LOS PROGRAMADORES

1. Implementan las aplicaciones desarrolladas utilizando diversos lenguajes de programación
2. Los lenguajes son elegidos según criterio, experiencia del desarrollador y siempre en un todo de acuerdo al tipo de necesidad que plantea el negocio, la organización, el emprendimiento, etc.
3. La BD estará lista cuando se conecte al software.



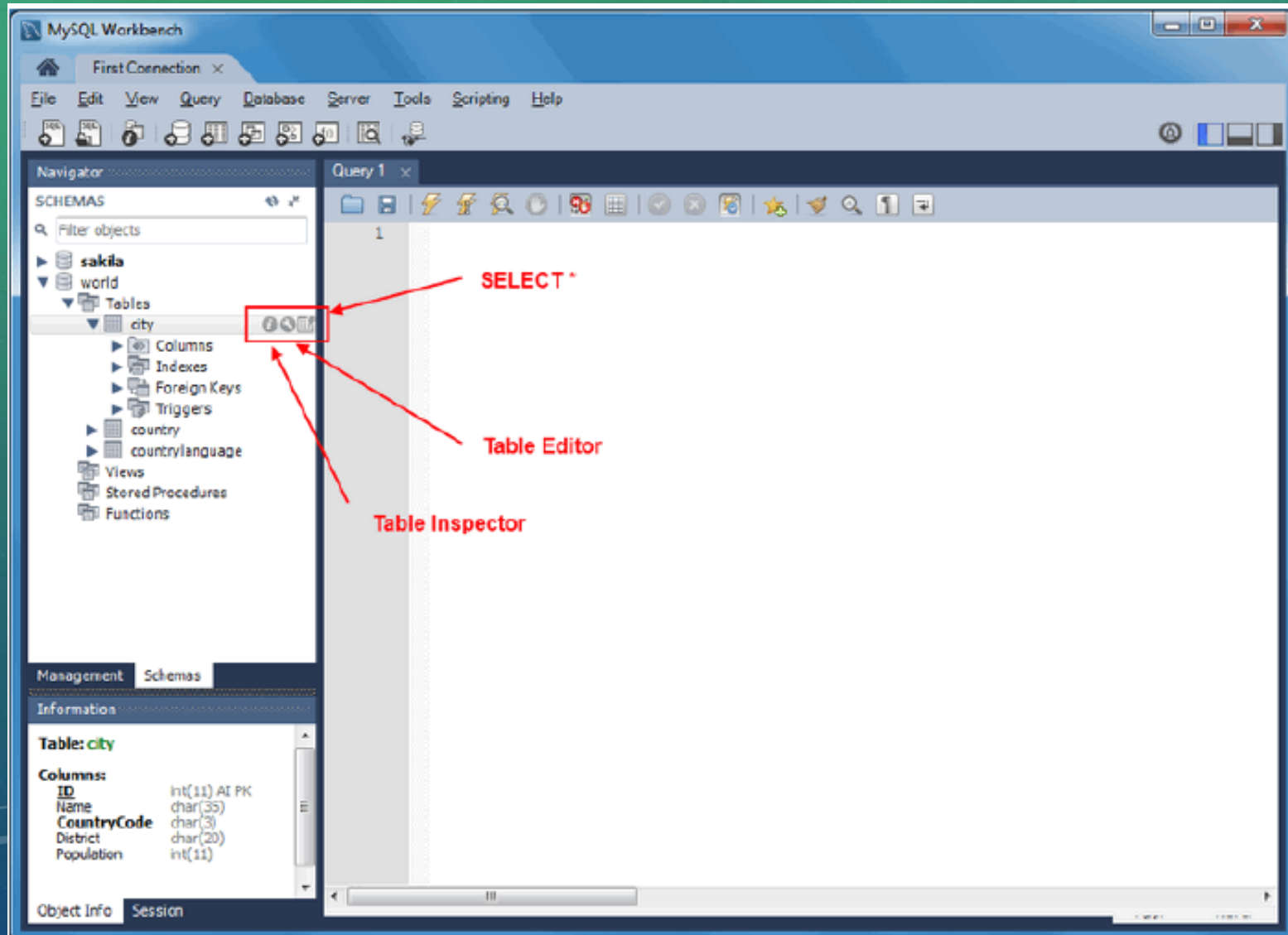
USUARIOS

En términos informáticos, los clientes son aquellos que tendrán acceso a la DB según su función, una vez que haya sido implementada.

Antes de la implementación, el paso previo es la prueba o testeo de funcionamiento, con carga de datos de muestra, para darle el visto bueno al funcionamiento.

Luego, se cargarán los datos previos, si los hubiera o se procederá directamente a la implementación si se trata de un proyecto que comienza desde cero con su desarrollo.

DATOS ADICIONALES DE LA INTERFAZ DE MYSQL



Tablas: Icono de tabla.

Vistas: Icono similar a una tabla, pero con un pequeño ojo o símbolo distintivo.

Procedimientos almacenados: Icono de un cilindro o un bloque con una flecha.

Funciones: Similar al icono de procedimiento, pero con un pequeño "f(x)".

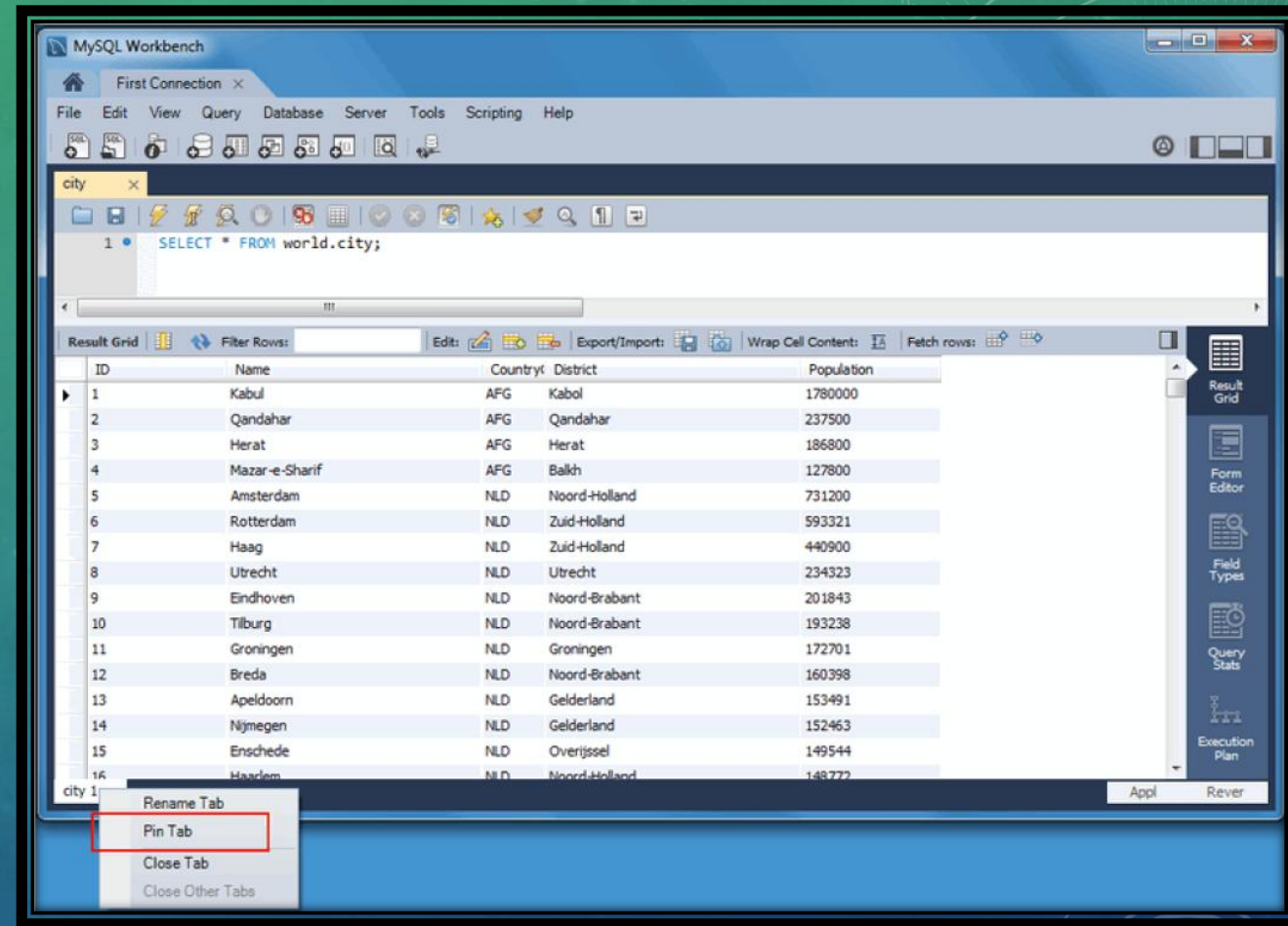
Triggers: Un rayo o un símbolo que indica acción.

PIN TAB

Permite anclar una pestaña abierta (por ejemplo, una consulta SQL o una tabla) para que no se cierre accidentalmente.

Cuando fijas una pestaña, aparecerá un pequeño ícono de pin en la pestaña.

Esta acción es útil cuando estás trabajando con varias consultas o tablas y quieres mantener algunas abiertas mientras navegas por otras.



Para fijar una pestaña click derecho sobre la pestaña y selecciona Pin Tab o Fijar Pestaña.

Para desanclarla, haz clic derecho nuevamente y selecciona Unpin Tab o Desfijar Pestaña.

PLAN DE EJECUCIÓN

Muestra cómo el motor de la base de datos va a ejecutar una consulta SQL.

Sirve para entender cómo se accede a los datos, qué índices se utilizan, el orden de las tablas y el costo de la consulta.

En MySQL, se utiliza el comando EXPLAIN para ver el plan de ejecución.

Visual Explain Type

MySQL 5.7+ Cost Information

View the JSON EXPLAIN source

Hover over to see additional information

Visual Explain Tab

```
1 SELECT CONCAT(customer.last_name, ', ', customer.first_name) AS customer,
2   address.phone, film.title
3 FROM rental INNER JOIN customer ON rental.customer_id = customer.customer_id
4 INNER JOIN address ON customer.address_id = address.address_id
5 INNER JOIN inventory ON rental.inventory_id = inventory.inventory_id
6 INNER JOIN film ON inventory.film_id = film.film_id
7 WHERE rental.return_date IS NULL
8 AND rental_date + INTERVAL film.rental_duration DAY < CURRENT_DATE() LIMIT 5;
```

Query cost: 8813.61

query_block #1

2.00K rows

8813.61

nested loop

2.00K rows

6413.61

nested loop

2.00K rows

4013.61

nested loop

2.00K rows

1613.61

nested loop

2.00K rows

213.0

1000 rows

Full Table Scan

film

1400.61

2 rows

Non-Unique Key Lookup

inventory
idx_fk_film_id

2400.0

1 row

Non-Unique Key Lookup

rental
idx_fk_inventory_id

2400.0

1 row

Unique Key Lookup

customer
PRIMARY

2400.0

1 row

Unique Key Lookup

address
PRIMARY

Result 1 x

Read Only

Creamos la tabla Clientes

```
CREATE TABLE clientes (  
id INT PRIMARY KEY,  
nombre VARCHAR(50),  
ciudad VARCHAR(50),  
edad INT,  
INDEX indice_ciudad (ciudad)  
);
```

Insertamos algunos datos

```
INSERT INTO clientes (id, nombre, ciudad, edad) VALUES  
(1, 'Ana', 'Apolinario Saravia', 30),  
(2, 'Luis', 'Anta', 25),  
(3, 'María', 'Tartagal', 35),  
(4, 'Pedro', 'Secantlas', 28),  
(5, 'Lucía', 'Apolinario Saravia', 22);
```

Ahora, queremos ver cómo MySQL ejecutará la siguiente consulta:

```
SELECT nombre, ciudad FROM clientes WHERE ciudad = 'Apolinario Saravia';
```

El resultado podría verse así

Id	select_type	table	type	possible_Keys	key	key_len	ref
----	-------------	-------	------	---------------	-----	---------	-----

PLAN DE EJECUCIÓN - EJEMPLO

- Consulta: `SELECT * FROM rental WHERE rental_date = '2005-05-24';`
- Para ver plan de ejecución: `EXPLAIN SELECT * FROM rental WHERE rental_date = '2005-05-24';`

Columna	Descripción
id	Identificador de la consulta. Indica el orden de ejecución
Select_type	Tipo de consulta: Simple, Primary, Subquery, Derived, etc.
tabla	Tabla a la que se accede en cada paso
type	Tipo de unión: All, index, range, ref, etc. Cuando más específico mejor
Possible_keys	Índices que podrían utilizarse para la consulta
key	Índice utilizado en la consulta
Key_len	Longitud del índice usado
ref	Columnas o constantes utilizadas en el índice
row	Número estimado de filas que se recorrerán
Extra	Información adicional, por ejemplo Using index, Using where

FINALIDAD DEL PLAN DE EJECUCIÓN

1. Optimización de Consultas:

- ✓ Identificar consultas ineficientes y proponer mejoras.
- ✓ Verificar si los índices se están utilizando correctamente o si faltan.

2. Diagnóstico de Problemas de Rendimiento:

Detectar consultas que recorren demasiadas filas (type = ALL), indicando que no se están usando índices.

3. Verificar el Uso de JOINS:

Asegurar que las tablas se estén uniendo en el orden más eficiente.

4. Identificación de **bottlenecks** (Botellas de Rendimiento):

Verificar operaciones costosas, como ordenamientos (Using filesort) o operaciones temporales (Using temporary).

¿POR QUÉ SURGEN LAS BOTTLENECKS?

1. Consultas Innecesariamente Complejas:

Consultas con JOINS excesivos o mal estructurados.

Consultas que no utilizan índices correctamente.

2. Falta de Índices o Índices Ineficientes:

Si una consulta busca datos sin un índice adecuado, MySQL debe recorrer todas las filas (FULL TABLE SCAN), lo que es mucho más lento.

3. Tablas Temporales y Ordenación (Using temporary, Using filesort):

Cuando MySQL crea tablas temporales o realiza ordenaciones complejas, puede consumir memoria y disco.

4. Bloqueos y Esperas (Locks): Si varias consultas intentan acceder a la misma tabla o fila, se pueden generar bloqueos que retrasan otras operaciones.

5. Limitaciones de Hardware: Falta de memoria, CPU saturado o discos lentos.

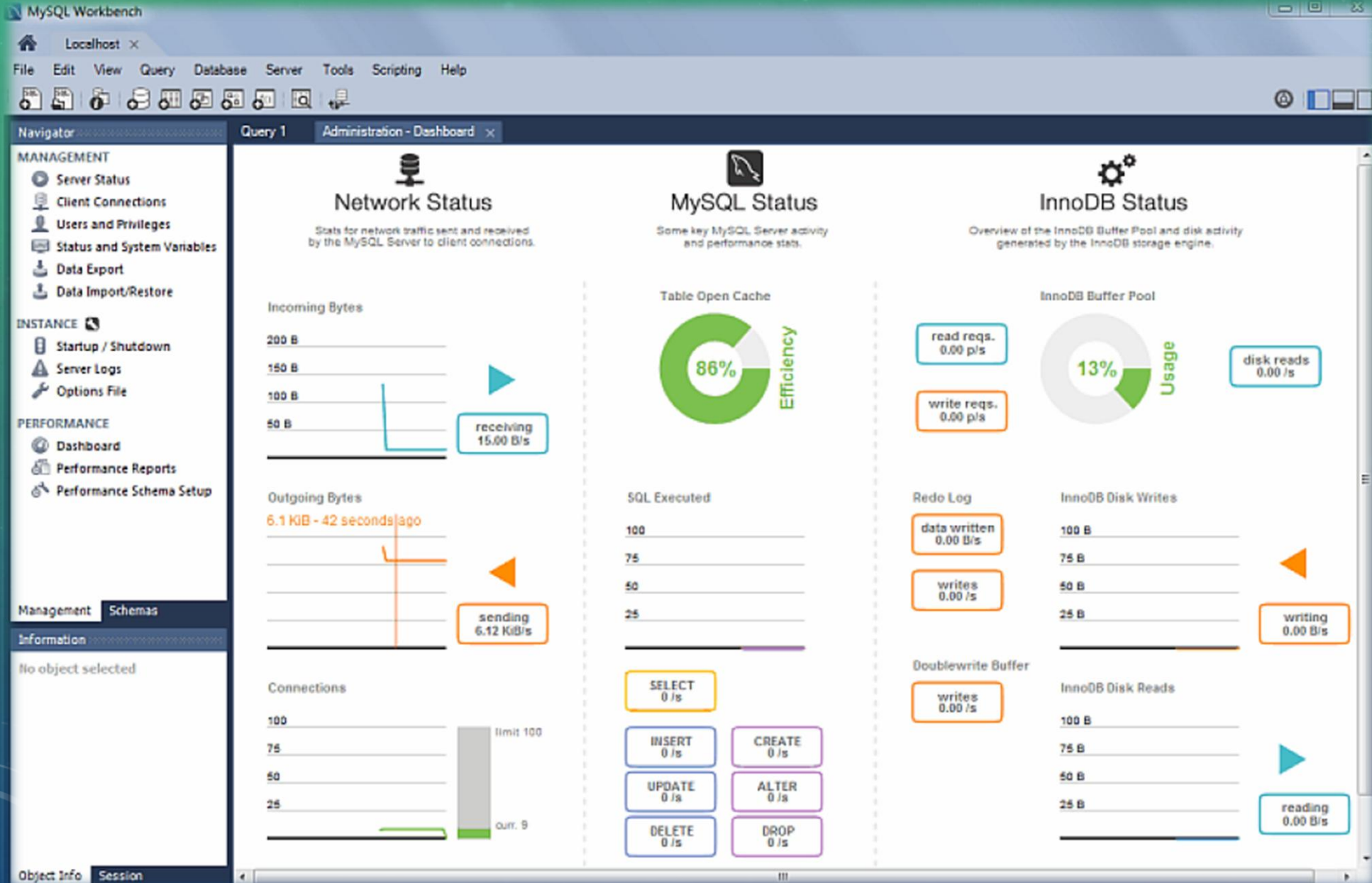
SECCIÓN DE RENDIMIENTO EN EL NAVEGADOR

Incluye:

- Dashboard (Panel de Control)
- Performance Reports (Informes de Rendimiento)
- Performance Schema Setup (Configuración del Esquema de Rendimiento)

Proporciona una representación gráfica de estadísticas clave del estado del servidor MySQL y ofrece una vista general de los subsistemas del servidor MySQL.

PERFORMANCE DASHBOARD



PERFORMANCE REPORT: TOP I/O BY BYTES

MySQL Workbench

Localhost x

File Edit View Query Database Server Tools Scripting Help

Navigator Query 1 Administration - Performance Re...

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Management Schemas

Information

No object selected

Object Info Session

Localhost Performance Reports

Report

- I/O
 - Top I/O by File by Bytes
 - Top I/O by File by Latency
 - Top I/O by Wait by Bytes
 - Top I/O by Wait by Latency
 - Top I/O by Thread by Latency
- Top
 - Top 10 Worst Latency
 - Top 10 Most Active Tables
- Problematic Statements
 - Statement Analysis
 - Runtimes in 95th Percentile
 - Using Temp Tables
 - With Sorting
 - Full Table Scans
 - Errors or Warnings
- Schema
 - Schema Object Overview
 - Schema Index Statistics
 - Schema Table Statistics
 - Schema Table Statistics (with InnoDB...)
 - Tables with Full Table Scans
 - Unused Indexes
- wait
 - Waits by Latency
 - Waits by User by Latency
 - Wait Classes by Latency
 - Wait Classes by Average Latency
- InnoDB
 - InnoDB Buffer Stats by Schema
 - InnoDB Buffer Stats by Table

Top I/O by File by Bytes

Show the top global IO consumers by bytes usage by file

File	Re...	Total Read	Avg Read	Writes	Total Written	Avg Writ...	Total	Write %
@datadir/mysql/backup_history...	144044	559.35 MB	3.98 KB	704	490.52 KB	713 bytes	559.83 MB	0.09
@datadir/mysql/backup_historyf...	3332	29.77 MB	9.15 KB	0	0 bytes	0 bytes	29.77 MB	0.00
@datadir/mysql/backup_progres...	3332	11.76 MB	3.61 KB	0	0 bytes	0 bytes	11.76 MB	0.00
@datadir/mysql/backup_history...	476	16.27 KB	35 bytes	950	32.47 KB	35 bytes	48.74 KB	66.62
@datadir/mysql/backup_progres...	476	16.27 KB	35 bytes	951	32.50 KB	35 bytes	48.77 KB	66.64
@datadir/mysql/proc.MYD	324	163.53 KB	517 bytes	52	76.30 KB	1.47 KB	239.82 KB	31.81
@datadir/performance_schema/s...	28	2.83 KB	104 bytes	0	0 bytes	0 bytes	2.83 KB	0.00
@datadir/performance_schema/f...	28	15.24 KB	557 bytes	0	0 bytes	0 bytes	15.24 KB	0.00
@datadir/mysql/proc.frm	28	12.27 KB	449 bytes	0	0 bytes	0 bytes	12.27 KB	0.00
@datadir/performance_schema/s...	28	3.72 KB	136 bytes	0	0 bytes	0 bytes	3.72 KB	0.00
@datadir/mysql/user.frm	28	12.59 KB	460 bytes	0	0 bytes	0 bytes	12.59 KB	0.00
@datadir/performance_schema/t...	21	10.53 KB	513 bytes	0	0 bytes	0 bytes	10.53 KB	0.00
@datadir/performance_schema/t...	21	9.70 KB	473 bytes	0	0 bytes	0 bytes	9.70 KB	0.00
@datadir/performance_schema/f...	21	2.58 KB	126 bytes	0	0 bytes	0 bytes	2.58 KB	0.00
@datadir/performance_schema/f...	21	3.57 KB	174 bytes	0	0 bytes	0 bytes	3.57 KB	0.00
@datadir/performance_schema/f...	21	3.66 KB	179 bytes	0	0 bytes	0 bytes	3.66 KB	0.00
@datadir/mysql/procs_priv.frm	21	5.07 KB	247 bytes	0	0 bytes	0 bytes	5.07 KB	0.00
@datadir/performance_schema/f...	21	7.46 KB	364 bytes	0	0 bytes	0 bytes	7.46 KB	0.00
@datadir/mysql/proxies_priv.frm	21	4.36 KB	213 bytes	0	0 bytes	0 bytes	4.36 KB	0.00
@datadir/performance_schema/f...	21	7.49 KB	365 bytes	0	0 bytes	0 bytes	7.49 KB	0.00
@datadir/performance_schema/f...	21	13.44 KB	655 bytes	0	0 bytes	0 bytes	13.44 KB	0.00
@datadir/performance_schema/f...	21	12.33 KB	601 bytes	0	0 bytes	0 bytes	12.33 KB	0.00
@datadir/performance_schema/f...	21	12.33 KB	601 bytes	0	0 bytes	0 bytes	12.33 KB	0.00
@datadir/performance_schema/s...	21	2.41 KB	118 bytes	0	0 bytes	0 bytes	2.41 KB	0.00
@datadir/performance_schema/t...	21	7.21 KB	352 bytes	0	0 bytes	0 bytes	7.21 KB	0.00
@datadir/performance_schema/s...	21	3.13 KB	153 bytes	0	0 bytes	0 bytes	3.13 KB	0.00
@datadir/performance_schema/f...	21	3.43 KB	167 bytes	0	0 bytes	0 bytes	3.43 KB	0.00

Export... Copy Selected Copy Query Refresh

PERSONAL SCHEMA SETUP: INTRODUCCIÓN

Se pueden activar, desactivar o ajustar los instrumentos que MySQL utiliza para recopilar datos de rendimiento.

1. Configuración Básica (Easy Setup):

- Permite activar rápidamente el Performance Schema con ajustes predeterminados.
- Es ideal para obtener información general sin sobrecargar el sistema.

2. Configuración Avanzada (Show Advanced):

- Permite ajustar instrumentos específicos, como consultas lentas (slow query), bloqueos (locks), o accesos a tablas (table access).
- Puedes activar o desactivar componentes específicos para reducir la carga de monitoreo.

MySQL Workbench

Localhost x

File Edit View Query Database Server Tools Scripting Help

Navigator Query 1 Administration - Performance Sc...

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Management Schemas

Information

No object selected

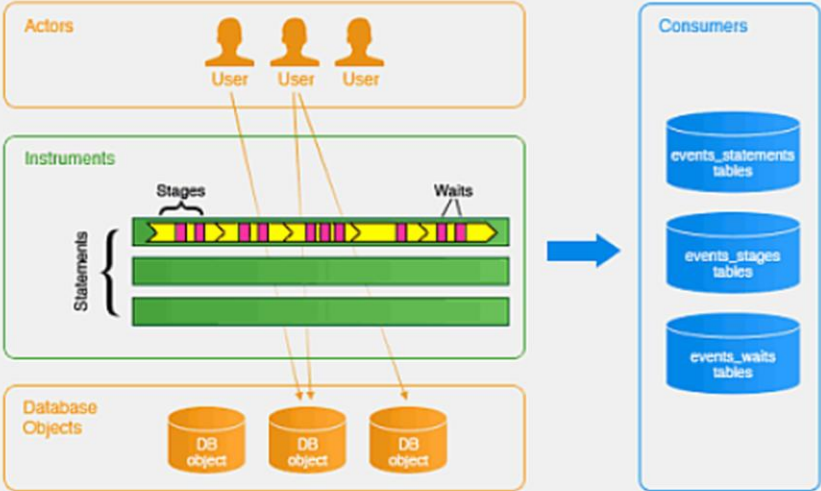
Object Info Session

Performance Schema - Setup

Easy Setup Introduction Instruments Consumers Actors & Objects Threads Options

Performance Schema Basics

The performance schema collects data from various aspects of MySQL performance and gives very detailed information about what exactly is happening inside your MySQL database server. For each statement executed, the PS instruments will gather various statistics and timing information in different levels of granularity and from different subsystems, from network to disk storage, and keep them in the `performance_schema.events_*` tables.



Actors

User User User

Instruments

Statements Stages Waits

Database Objects

DB object DB object DB object

Consumers

events_statements tables

events_stages tables

events_waits tables

Configuring Performance Schema

To control the trade-off between data collected and overhead, the performance schema gives you a few fine grained configuration options.

You can configure what, when and how much will be instrumented by the Performance Schema by tweaking three option categories:

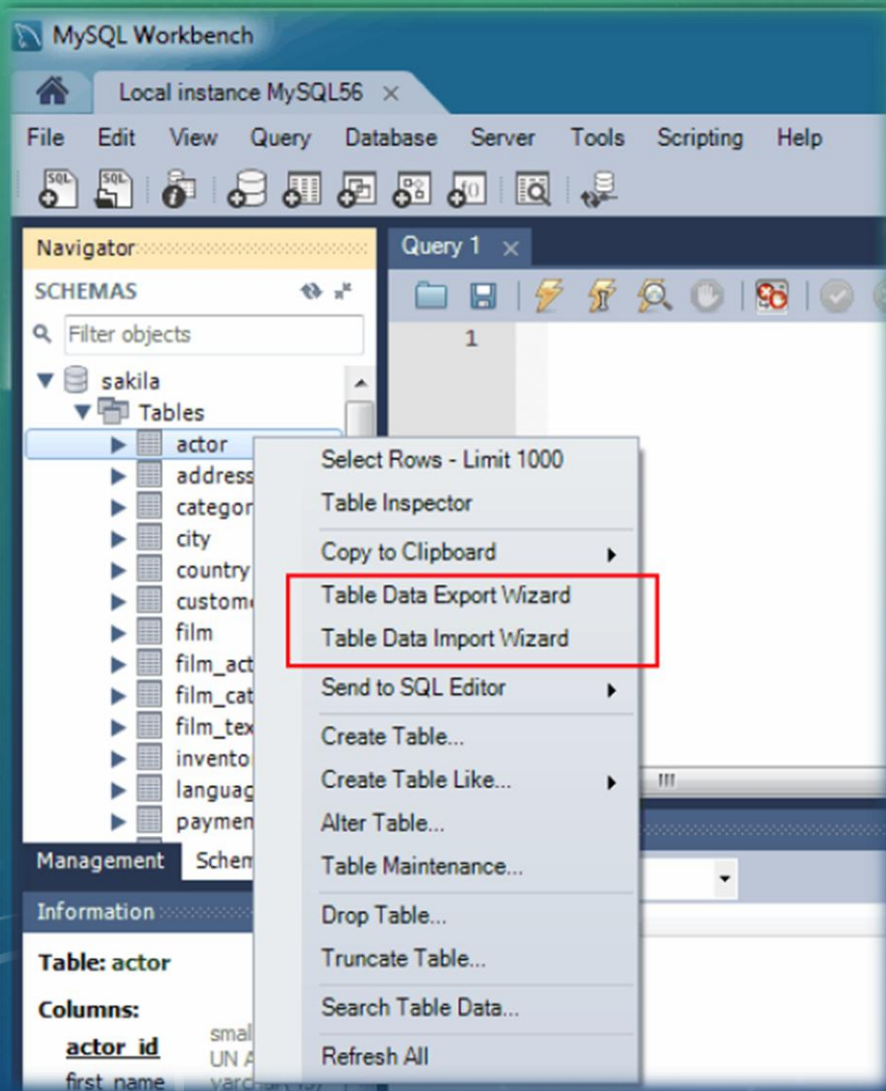
- * Actors - filters the users, hosts and DB objects to collect data for the performance_schema. This was introduced in MySQL 5.6.
- * Instruments - allow fine-tuning of what kind of stats are gathered for whatever is being monitored
- * Consumers - toggle which of the performance_schema.event_* tables should be filled

You can also use the simplified configuration interface for one-click setup of the performance schema for some common use cases.

Clear Event Tables Reset to Factory Defaults Revert Apply

PERFORMANCE SCHEMA SETUP

EXPORTAR DATOS DE UNA TABLA EN MYSQL WORKBENCH



1. Abrir el Explorador de Objetos:
En el panel izquierdo, ubica la base de datos y expande las tablas.

2. Clic Derecho sobre la Tabla:
Haz clic derecho sobre la tabla que quieres exportar.
Selecciona Table Data Export Wizard.

Select output file location

Table Data Export allows you to easily export data into csv, json datafiles.

File Path: D:\MySQLBackups\sakila-actor.csv

Browse...

☒ csv ☐ json

Options:

Line Separator

LF

Enclose Strings in

"

Field Separator

;

☒ Export to local machine

If checked rows will be exported on the location that started Workbench.

If not checked, rows will be exported on the server.

If server and computer that started Workbench are different machines, import of that file can be done manual way only.

< Back

Next >

Cancel

3. Configurar la Exportación:

Selecciona el formato de exportación (CSV, JSON, SQL, etc.).

Especifica la ubicación del archivo de exportación.

Puedes seleccionar qué columnas exportar y establecer filtros.

4. Opciones Avanzadas (opcional):

Puedes ajustar configuraciones como el delimitador de campos (ej., coma, punto y coma).¿

5. Iniciar la Exportación:

Haz clic en Start Export (Iniciar Exportación).

Revisa el log para asegurarte de que no hubo errores.

Select data for export

Select source table for export.

sakila.actor

Select columns you'd like to export

Export	Column name
<input checked="" type="checkbox"/>	actor_id
<input checked="" type="checkbox"/>	first_name
<input checked="" type="checkbox"/>	last_name
<input checked="" type="checkbox"/>	last_update

☒ Select / Deselect all entries

Row Offset:

Count:

Advanced >>

< Back

Next >

Cancel

Export Data

The following tasks will now be performed. Please monitor the execution.

- ☒ Prepare Export
- ☒ Export data to file

Finished performing tasks.
Click [Next >] to continue.

Message Log

- Data export
- Data export
- Data export
- Data export
- Data export
- Data export
- Data export
- Data export
- Data export
- Export finished
Export data to file finished
Finished performing tasks.

Hide Logs

< Back

Next >

Cancel

IMPORTAR DATOS A UNA TABLA EN MYSQL WORKBENCH

1. Abrir el Explorador de Objetos:

En el panel izquierdo, ubica la tabla donde quieres importar los datos.

2. Clic Derecho sobre la Tabla:

Haz clic derecho sobre la tabla.

Selecciona Table Data Import Wizard.

3. Seleccionar el Archivo a Importar:

Busca y selecciona el archivo (CSV, JSON, etc.) que contiene los datos a importar.

Select File to Import

Table Data Import allows you to easily import csv, json datafiles.
You can also create destination table on the fly.

File Path: D:\MySQLBackups\sakila-actor.csv

Browse...

< Back

Next >

Cancel

Select Destination

Select destination table and additional options.

- ☐ Use existing table: sakila.actor
- ☒ Create new table: sakila . sakila-actor-other
- ☐ Drop table if exists

< Back

Next >

Cancel

4. Configurar la Importación:

Define cómo se deben mapear las columnas del archivo con las columnas de la tabla.

Especifica el delimitador (ej., coma, tabulación).

Opcionalmente, puedes previsualizar los datos antes de importarlos.

5. Iniciar la Importación:

Haz clic en Start Import (Iniciar Importación).

Verifica el log para detectar errores o inconsistencias.

Configure Import Settings

Detected file format: csv

Options:

Line Separator

LF

Enclose Strings in

"

Field Separator

;

Encoding: utf-8

Columns:

☒ Source Column Field Type☒ actor_id int☒ first_name text☒ last_name text☒ last_update text

actor_id	first_name	last_name	last_update
1	PENELOPE	GUINNESS	2006-02-15 ...
2	NICK	WAHLBERG	2006-02-15 ...
3	ED	CHASE	2006-02-15 ...
4	JENNIFER	DAVIS	2006-02-15 ...
5	JOHNNY	LOLLOBRIG...	2006-02-15 ...

< Back

Next >

Cancel

Import Data

The following tasks will now be performed. Please monitor the execution.

- ☒ Prepare Import
- ☒ Import data file

Finished performing tasks.
Click [Next >] to continue.

Message Log

- Data import
- Data Import
- Data import
- Data import
- Data Import
- Data import
- Data import
- Data import
- Data import
- Data import
- Data import
- Data import
- Data Import
- Data import
- Data import
- Data import
- Data import
- Data import
- Data import
- Data import
- Data import
- Import finished
Import data file finished
Finished performing tasks.

Hide Logs

< Back

Next >

Cancel

IMPORTANTE

- Respalda tu base de datos antes de realizar importaciones o exportaciones.
- Revisa los formatos de las columnas para evitar errores (ej., fechas, decimales).
- En caso de problemas, revisa el log que se genera al final del asistente.