



TRANSACCIONALIDAD EN LAS OPERACIONES





¿QUÉ VAMOS A VER?

- Transaccionalidad
- TRANSACTION
- COMMIT
- ROLLBACK



¡VAMOS A COMENZAR!





TRANSACCIONALIDAD EN LAS OPERACIONES



TRANSACCIONALIDAD

- En palabras simples, es una unidad de trabajo que se realiza en una BBDD.
- En conjunto, las transacciones son una secuencia de trabajo que va a ser realizadas de manera manual o automáticas.
- La unidad de trabajo también se puede entender como un cambio realizado en la BBDD, por ende, la transacción en sí sería un conjunto de cambios.
- Algo tan simple como crear, modificar u eliminar un campo de una tabla, hace referencia a una transacción en la tabla en cuestión.



PROPIEDADES ACID



- **ATOMICITY:** La atomicidad de las tablas nos garantiza que todas las operaciones dentro de la unidad de trabajo se completen con éxito, de lo contrario, la transacción se aborta en el punto de falla y con ello, las operaciones anteriores se revierten para asegurar el estado optimo anterior.
- **COHERENCE:** La coherencia nos garantiza que la base de datos cambia correctamente de estado una vez que la transacción fue exitosa.
- **ISOLATION:** El aislamiento permite que las transacciones funcionen de manera independiente y transparente entre sí.
- **DURABILITY:** La durabilidad nos asegura que el efecto de una transacción que ya está comprometida va a persistir en caso de que exista una falla de sistema.

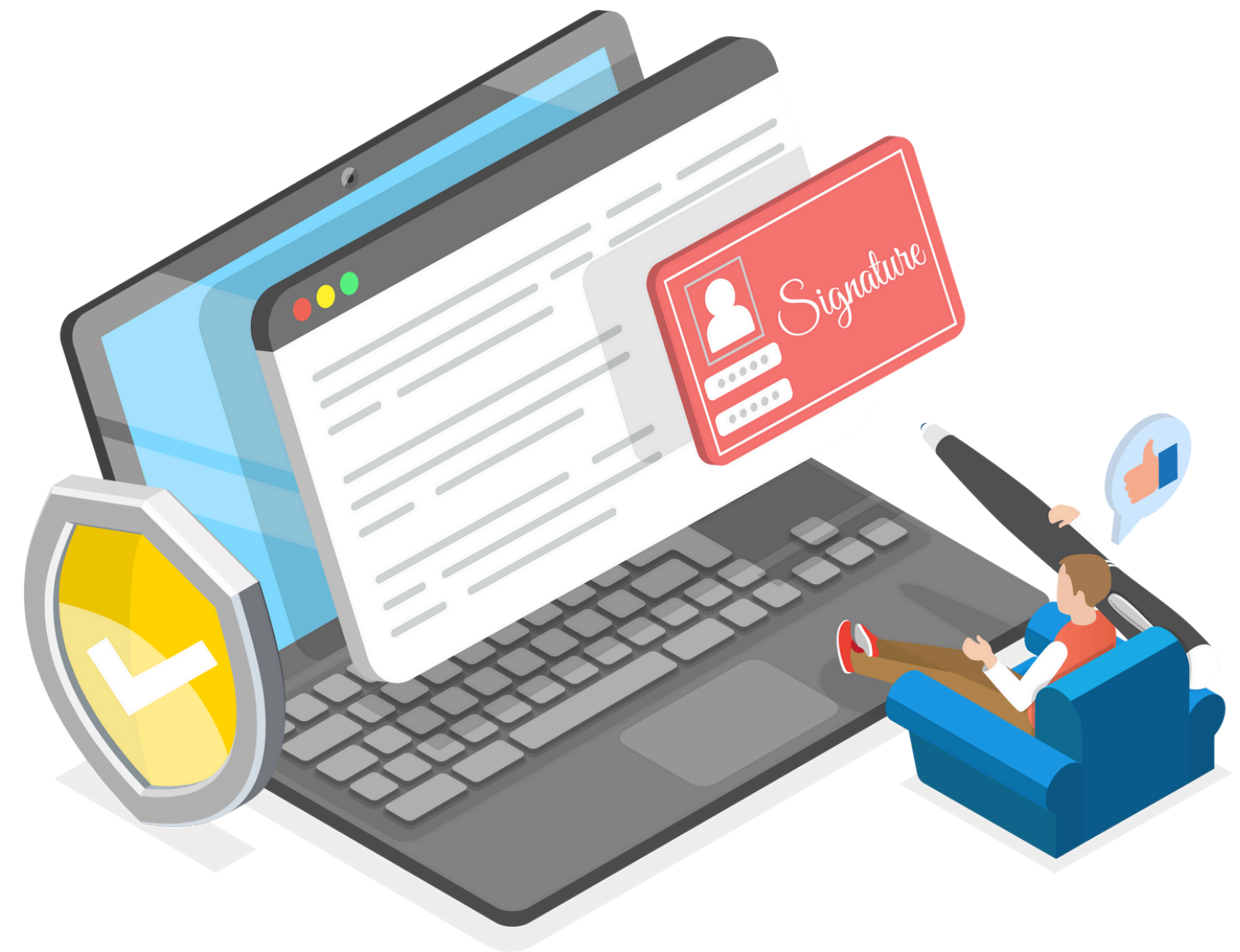


MyISAM



- **Características:**

- *Se establece por defecto cuando se crea una tabla, salvo que se especifique lo contrario.*
 - *Soporta transacciones.*
 - *Realiza bloqueos de registros.*
 - *Soporta un gran número de consultas SQL.*
- No es capaz de realizar un bloqueo de tabla, algo que podría ser un problema si tenemos un acceso simultáneo de usuarios que quieran cambiar demasiados registros en ese preciso momento.



InnoDB



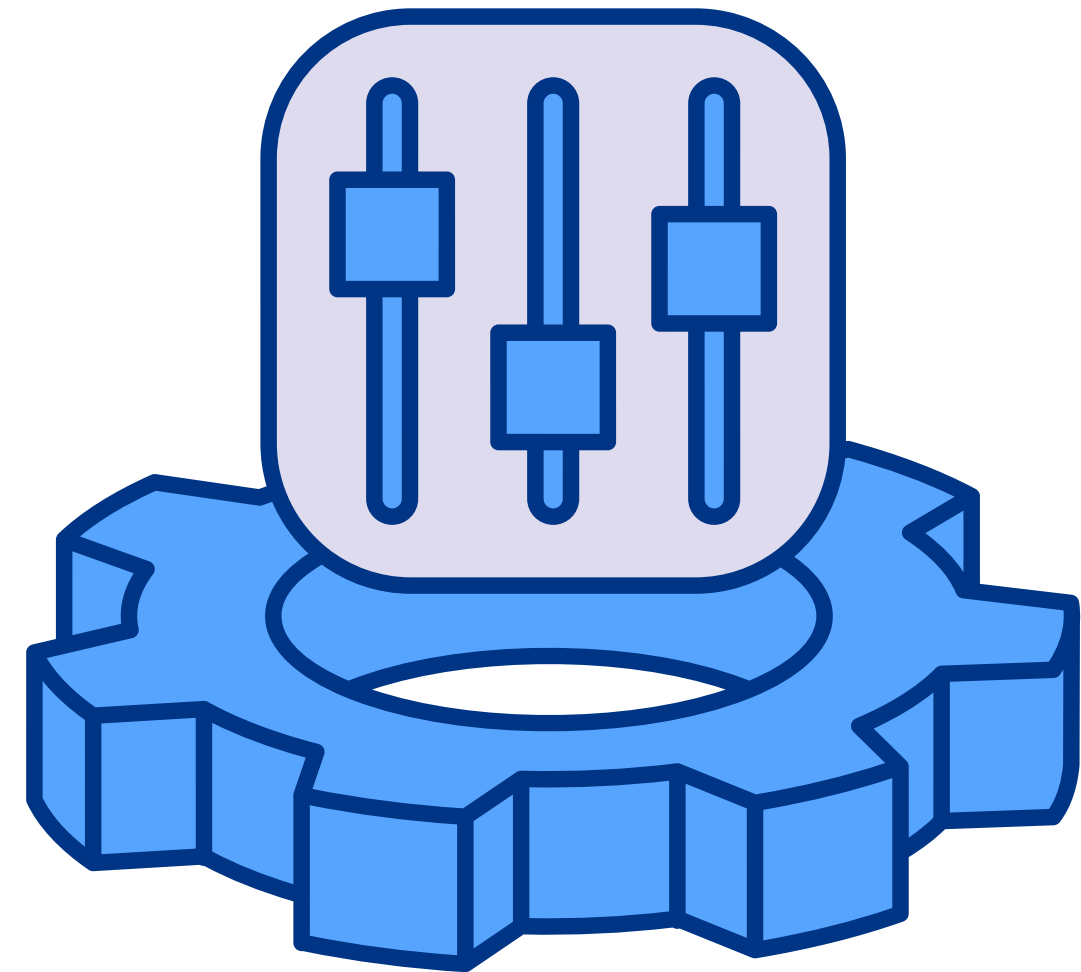
- **Características:**

- *Tiene bloqueo de registros.*
 - *Es capaz de soportar transacciones e integridad de datos, es decir, previene que ingresen datos no deseados.*
 - *Aplica las características propias de **ACID**, para garantizar la integridad de las tablas.*
- Una desventaja que se podría nombrar es que exige un buen diseño en el ecosistema de sus tablas, por ende, eso significa que la creación o almacenamiento de la información va a ser “más” lento.



¿QUÉ TIPO DE TABLA USAR?

- Si solo se va a utilizar un gestor de mantenimiento, se podría seleccionar **MyISAM**.
- Si necesitamos velocidad y un mínimo consumo de los recursos del servidor: **MyISAM**.
- Si tendremos varios gestores de mantenimiento: **InnoDB**.
- Si vamos a priorizar el diseño relacional de nuestra base de datos: **InnoDB**.
- A partir de la **versión 4.0** MySQL da soporte por defecto a tablas de tipo **InnoDB**.





CONFIRMACIÓN DE UNA TRANSACCIÓN

- Los comandos **DML** *INSERT*, *UPDATE* y *DELETE* son los encargados de mantener un control transaccional.
- Una transacción se puede iniciar utilizando **START**. De esa manera va a persistir hasta encontrarse con el comando **COMMIT** o **ROLLBACK**.
- En caso de existir una falla en el sistema o la BBDD, el comando de **ROLLBACK** se ejecuta automáticamente.
- **COMMIT** también hace el trabajo de ser aquel comando que se utiliza para guardar los cambios invocados por la transacción.





VUELTA ATRÁS DE UNA TRANSACCIÓN

- **ROLLBACK** es el comando que se utiliza para deshacer transacciones que aún no se han cargado a la BBDD.
- Este comando solo va a deshacer los cambios que se intentaron realizar luego del último **COMMIT** utilizado.





AUTOCOMMIT

- Por defecto, las BBDD traen el modo *autocommit* activado, es decir, que, si manualmente no le especificamos que nos encontremos dentro de una transacción, todo comando DML que apliquemos será guardado sin oportunidad de *rollback*.
- Al ser una variable de sesión, estamos facultados para configurar el *autocommit* cada vez que vamos a utilizar transacciones.



AUTOCOMMIT



- **START TRANSACTION;**
- **SET *autocommit* = 0;**
- **INSERT INTO** *profesores* (nombre, apellido, escuela, fecha_contratacion, sueldo) **VALUES** ('Alfonsina', 'Araya', 'Escuela D-25', '2021-01-01', 600000);
- **INSERT INTO** *profesores* (nombre, apellido, escuela, fecha_contratacion, sueldo) **VALUES** ('Bernardo', 'Bustos', 'Escuela Z-001', '2021-02-02', 850000);
- **SELECT * FROM** *profesores*;
- **ROLLBACK;**
- **SELECT * FROM** *profesores*;
- **SET *autocommit* = 1;**





GRACIAS POR
LA ATENCIÓN