A6. Acceso simultáneo de los datos: Políticas de bloqueo.

Índice.

A6. Acceso simultáneo de los datos: Políticas de bloqueo.

1. Acceso concurrente a los datos.

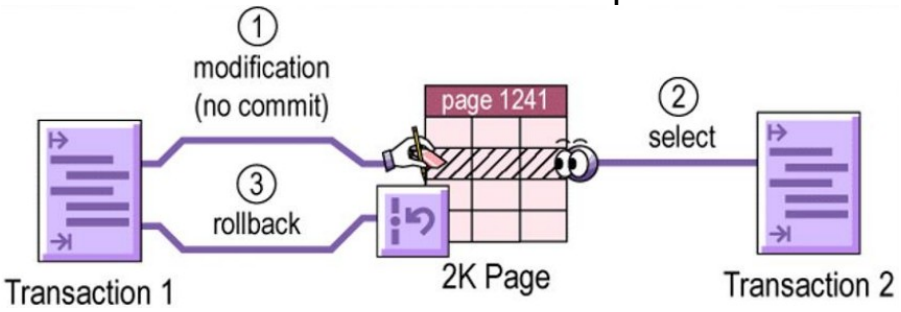
El acceso concurrente a los datos se produce cuando dos transacciones distintas intentan acceder de forma concurrente a unos mismos datos.

A6. Acceso simultáneo de los datos: Políticas de bloqueo.

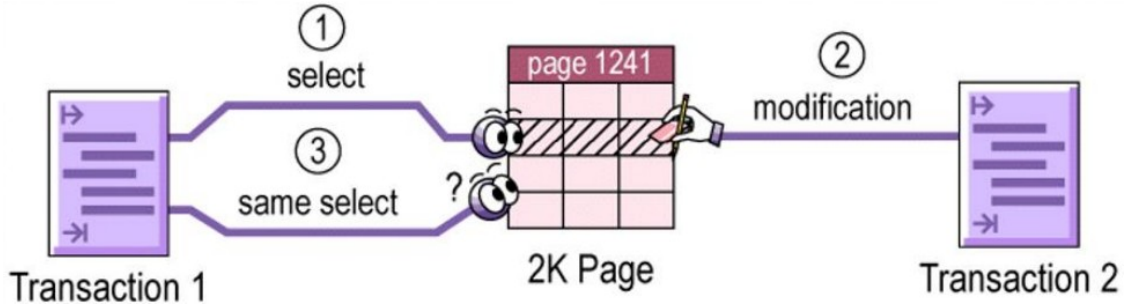
1. Acceso concurrente a los datos.

Los problemas asociados a la concurrencia de datos son los siguientes:

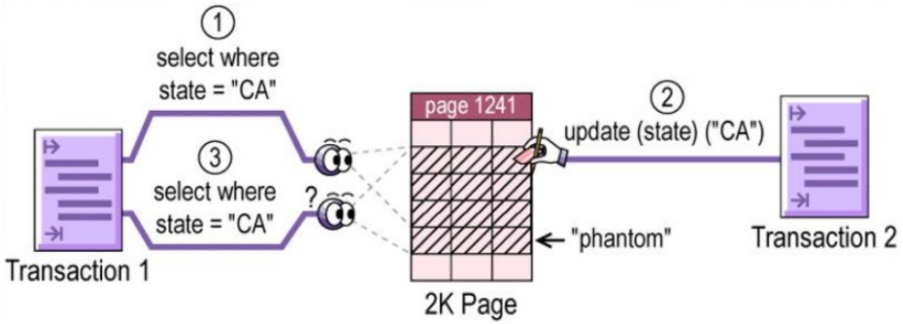
- **Lectura sucia** → si una transacción lee datos que aún no han sido confirmados.



- **Lectura no repetible** → si una transacción lee una misma fila varias veces y obtiene valores distintos.



- **Lectura fantasma** → si dos transacciones iguales devuelven filas distintas.



2. Niveles de aislamiento.

El **nivel de aislamiento de una transacción** (Transaction Isolation Level) es el grado en que se aísla una transacción ante las modificaciones en los recursos (o datos) realizados por otras transacciones.

Nivel de aislamiento	Tipos de violaciones		
	Lectura sucia	Lectura no repetible	Fantasmas
READ UNCOMMITTED	Sí	Sí	Sí
READ COMMITTED	No	Sí	Sí
REPEATABLE READ	No	No	Sí
SERIALIZABLE	No	No	No
SNAPSHOT	No	No	No

2. Niveles de aislamiento.

El **nivel de aislamiento de una transacción** es una característica de vital importancia en el desarrollo de aplicaciones de Bases de Datos, porque afecta a los tipos y a la duración de los bloqueos que se producen en la Base de Datos, lo que repercute en el rendimiento y en el tiempo de respuesta a las Consultas y Transacciones.

La elección del modo de aislamiento tiene una mayor importancia cuando mayor sea la necesidad de concurrencia de la Base de Datos.

SET TRANSACTION ISOLATION LEVEL [READ UNCOMMITTED | READ COMMITTED | REPEATABLE READ | SERIALIZABLE | SNAPSHOT]

2. Niveles de aislamiento.

Los distintos niveles de aislamiento son:

- **Read Uncommited** → No hay ningún tipo de bloqueo → lectura sucia, no repetible y fantasma.

```
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED ;
```

- **Read Commited** → los datos leídos se pueden modificar por otra transacción → lectura no repetible y fantasma.

```
SET TRANSACTION ISOLATION LEVEL READ COMMITTED ;
```

- **Repeatable Read** → ningún registro leído por un SELECT se modificará por otra transacción → lectura fantasma.

```
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ ;
```

- **Serializable** → todas las transacciones se ejecutarán de forma secuencial, sin paralelismo.

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE ;
```

- **Snapshot** → no se verán las modificaciones de datos efectuadas por otras transacciones.

```
SET TRANSACTION ISOLATION LEVEL  SNAPSHOT;
```

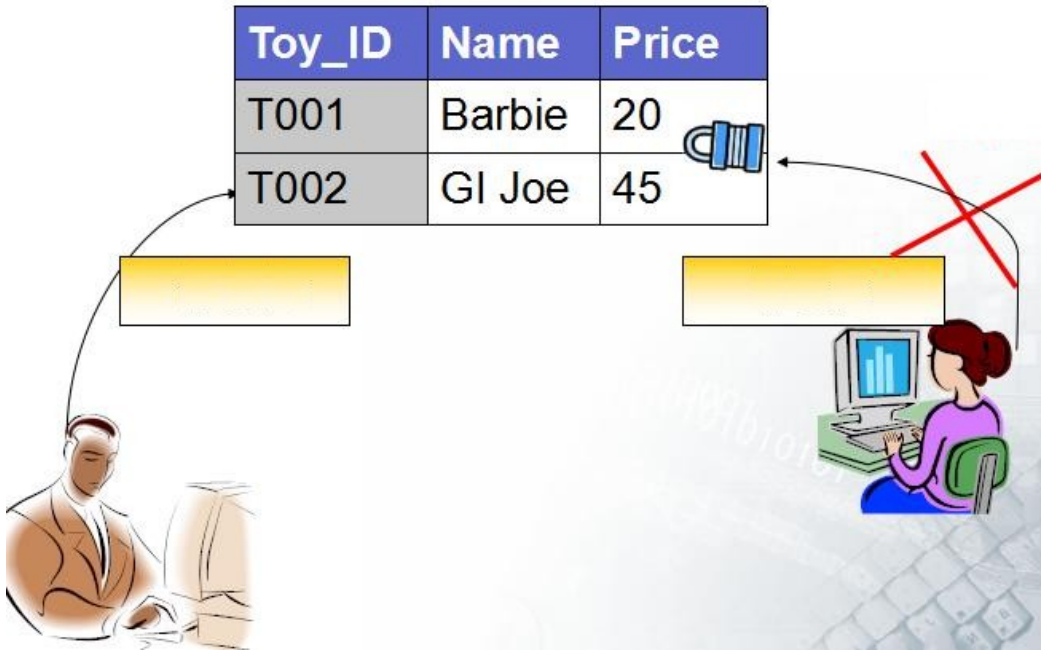
El nivel de aislamiento que utiliza InnoDB por defecto es **Repeatable Read**.

3. Políticas de bloqueo.

Si una transacción accede a los datos, lo realiza de forma exclusiva, de tal forma que otra transacción NO pueda acceder a los mismos datos que están siendo utilizados por otra transacción hasta que haya terminado.

El bloqueo de datos se puede realizar a distintos niveles:

- A nivel de Base de Datos.
- A nivel de Tabla.
- A nivel de Fila.
- A nivel de Columna.

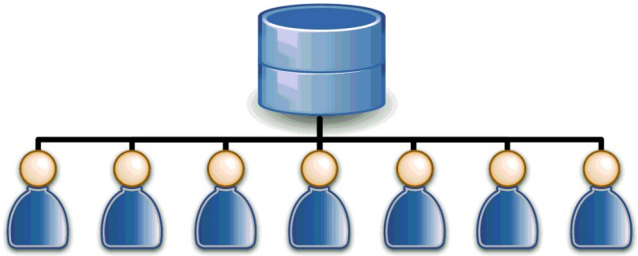


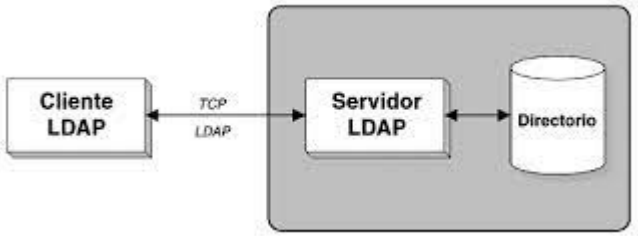
Cabe destacar que InnoDB SÓLO realiza el bloqueo a nivel de FILA.

4. Acceso a la información.

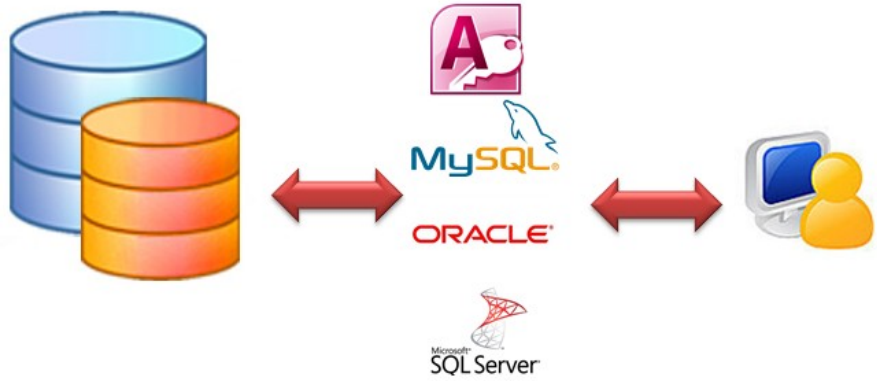
A la hora de realizar la Administración de Seguridad en el acceso a la información de una Base de Datos, suele haber dos tipos de seguridad:

- **Integrada en el Sistema Operativo** → suele contar con un gestor de Base de Datos con los usuarios de un Sistema de Dominio o con un servicio de directorio (LDAP) para proporcionar el acceso a determinados recursos.





- **Proporcionada por el SGBD (nativa)** → el propio software servidor proporciona mecanismos a través de los cuales se autoriza a un usuario la utilización de los distintos elementos de la Base de Datos.



5. Cuentas de Usuario.

Las cuentas de usuario sirven para asignar permisos sobre los distintos objetos de la Base de Datos con unos determinados privilegios.

Las operaciones asociadas son:

- **Creación.**

  CREATE USER <nombre> IDENTIFIED BY <password>

- **Eliminación.**

  DROP USER <usuario> [CASCADE]

  CASCADE borra todos los objetos del esquema

- **Modificación.**

  UPDATE mysql.user
  SET host = <dirección_IP> WHERE user = "YO";
  FLUSH PRIVILEGES;

- **Renombrado de usuario.**

  RENAME USER usuario@localhost TO nuevousuario@localhost;

- **Cambio de password.**

  SET PASSWORD FOR usuario@localhost = PASSWORD('abc123.');

6. Privilegios.

Un usuario de la Base de Datos puede manipular los objetos de la Base de Datos, pero también se le puede denegar dichos permisos o una parte de los mismos.

Los comandos relacionados son:

- **Grant** → otorga permisos a los diferentes niveles (host, base de datos, tabla, columna).

```
GRANT <privilegio>
        ON [tabla | * | *.* | basedatos.* | basedatos.tabla]
        TO <usuario> [IDENTIFIED BY [PASSWORD] <password>]
        WITH [GRANT OPTION |
                MAX_QUERIES_PER_HOUR <numero> |
                MAX_UPDATES_PER_HOUR <numero> |
                MAX_CONNECTIONS_PER_HOUR <numero> |
                MAX_USER_CONNECTIONS <numero>]

<privilegio>
     ALL | ALTER | CREATE | CREATE USER | CREATE VIEW | DELETE | DROP | EXECUTE |
     FILE | INDEX | INSERT | LOCK TABLES | PROCESS | RELOAD | SELECT | SHOW DATABASES |
     SHOW VIEW | SHUTDOWN | UPDATE | USAGE | GRANT OPTION
```

- **Revoke** → quita permisos a un usuario sobre un objeto.

```
REVOKE <privilegio>
          ON [tabla | * | *.* | basedatos.* | basedatos.tabla]
          FROM  <usuario> [IDENTIFIED BY [PASSWORD] <password>]

<privilegio>
     ALL | ALTER | CREATE | CREATE USER | CREATE VIEW | DELETE | DROP | EXECUTE |
     FILE | INDEX | INSERT | LOCK TABLES | PROCESS | RELOAD | SELECT | SHOW DATABASES |
     SHOW VIEW | SHUTDOWN | UPDATE | USAGE | GRANT OPTION
```