

[Página Principal](#) / [Mis cursos](#) / [Base de Datos](#) / [UNIDAD 3](#) / [Lenguaje SQL](#)

Lenguaje SQL

Propiedades de las transacciones

Las transacciones tienen que cumplir **cuatro** propiedades:

- **Atomicidad:** se realizan todas las operaciones involucradas con éxito o no se realiza ninguna.
- **Consistencia:** la base de datos pasa de un estado consistente a otro.
- **Aislamiento:** cada transacción es aislada, es decir que mientras está ocurriendo ninguna otra transacción puede acceder a los datos que están siendo modificados por la misma.
- **Durabilidad:** los cambios ocurridos en la base de datos por causa de la transacción son persistentes, es decir que no se pueden perder por causa de una falla.



Por este motivo se dice que las transacciones tienen propiedades **ACID** (por sus siglas en inglés).

En el lenguaje SQL, para marcar el comienzo de una transacción, se utiliza la sentencia START TRANSACTION y para marcar el fin de la transacción se utiliza COMMIT.

Si en lugar de cerrar la operación deseamos cancelar los cambios tenemos que ejecutar la sentencia ROLLBACK.



Volviendo al ejemplo de la transferencia de Juan a Pedro, se haría de esta manera:



SQL

```
START TRANSACTION

UPDATE movimientos
  SET monto = monto - 550
  WHERE CBU = (SELECT CBU from cuentas WHERE ID=23)

UPDATE movimientos
  SET monto = monto + 550
  WHERE CBU = (SELECT CBU FROM cuentas WHERE ID=44)

COMMIT
```

El **DBMS** implementa mecanismos que refuerzan el comportamiento de las transacciones y aseguran que ante una falla, los estados de todas las tablas involucradas vuelvan a ser los mismos que antes de que comenzara la transacción. Estos son los llamados mecanismos de recuperación, ellos utilizan puntos de control o marcas de tiempo que les permiten poder volver a esos estados anteriores al comienzo de la transacción. Veremos estos conceptos en la unidad 4.

Muchas veces es útil trabajar con variables cuando usamos transacciones. Los nombres de las variables en lenguaje SQL se antepone con el símbolo **@** y a continuación una sucesión de caracteres, por ejemplo **@nom21**. Se deben declarar previamente a su utilización indicando su tipo.

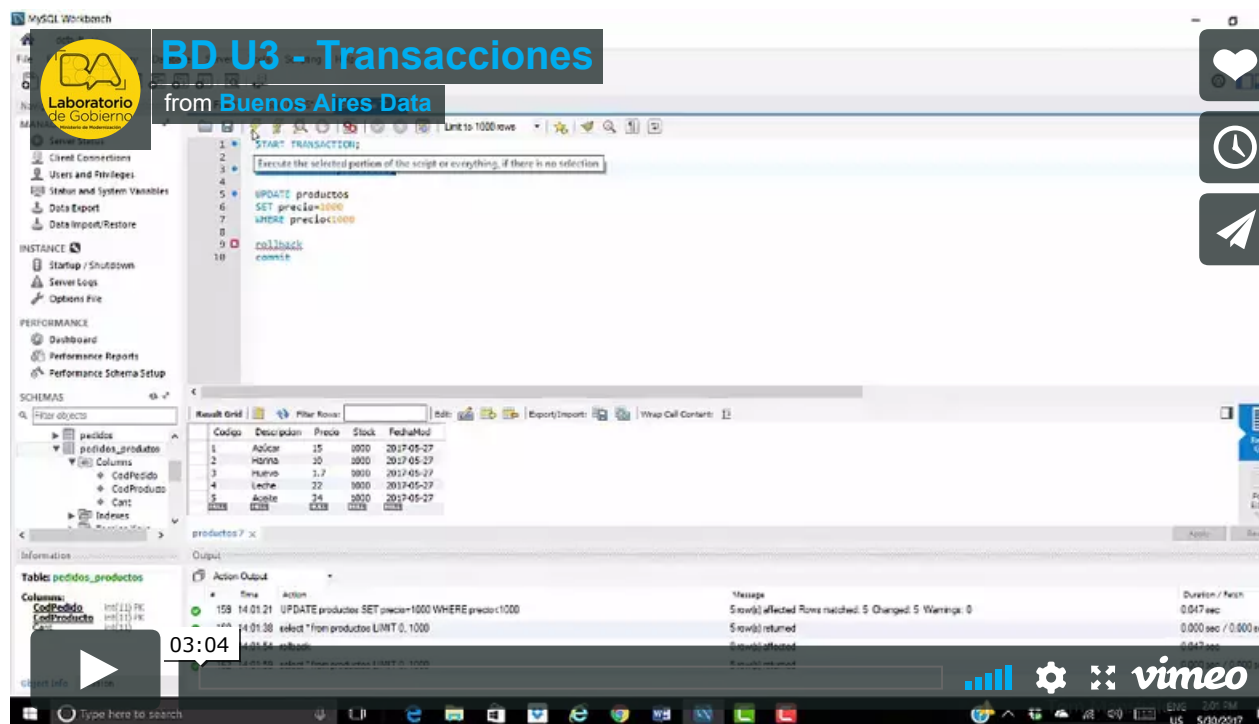
Vemos a continuación un ejemplo muy común del uso de una variable:

```
</> SQL

set @maxmonto = (SELECT max(monto) from movimientos)
SELECT @maxmonto
```

Con esta serie de comandos se ha creado la variable **@maxmonto**, y se le ha asignado el valor del máximo monto de la tabla **movimientos**, para luego mostrarlo en la siguiente sentencia.

Te presentamos el siguiente video titulado "Transacciones" donde se muestran algunos ejemplos muy interesantes de cómo se trabaja con transacciones en el entorno MySQL Workbench.



The screenshot displays a MySQL Workbench interface with a video player overlay. The video title is "BD U3 - Transacciones" from "Buenos Aires Data". The video content shows a SQL script being executed in the "SQL Editor" tab, which includes a transaction start, an update on the "productos" table, and a commit. The "Result Grid" shows the state of the "productos" table before and after the update. The "Output" tab shows the execution results of the SQL commands.

Codigo	Descripcion	Precio	Stock	FechaMod
1	Aplicar	15	1000	2017-05-27
2	Mama	30	2000	2017-05-27
3	Mucho	1.7	3000	2017-05-27
4	Leche	22	3000	2017-05-27
5	Aviole	24	3000	2017-05-27

The video player shows a play button and a timestamp of 03:04. The video player interface includes a progress bar and a volume control icon.

[< Anterior](#)[Siguiete >](#)