

# MySQL



# Repaso

2

```
SELECT id, title, release_date  
FROM movies  
WHERE release_date > '2001-01-01'  
ORDER BY title DESC  
LIMIT 5;
```

# Repaso

3

```
SELECT movies.title, movies.release_date, genres.name  
FROM movies  
    INNER JOIN genres ON genres.id = movies.genre_id  
WHERE movies.release_date > '2000-01-01';
```

# Repaso

4

```
SELECT actors.first_name, actors.last_name
FROM actors
    INNER JOIN actor_movie ON actors.id = actor_movie.actor_id
    INNER JOIN movies ON movies.id = actor_movie.movie_id
WHERE
    movies.title LIKE '%Harry Potter%';
```

# Repaso

5

Obtener los actores de las películas que digan “Harry Potter”

nombre	apellido
▶ Daniel	Radcliffe
Emma	Watson
Rupert	Grint
Daniel	Radcliffe
Emma	Watson
Rupert	Grint
Daniel	Radcliffe
Emma	Watson
Rupert	Grint
Helena	Bonham Carter



# IMAGINEN

6



# ACID

7

**A:** Atomicidad.

**C:** Consistencia.

**I:** Isolated (Aislamiento).

**D:** Durabilidad.

Un SGBD es **transaccional** cuando respeta el ACID.

# Transacciones - Sintaxis

8

**Start transaction;**

Consulta1

Consulta2...

**Commit;**

**Rollback;**



**GRACIAS**

9

**¿Preguntas?**



# Transacciones

Una vez realizada una conexión a través de PDO, es necesario comprender cómo PDO gestiona las transacciones antes de comenzar a realizar consultas.

Una transacción ayuda a cumplir con:

- **A**tomicidad.
- **C**onsistencia.
- **A**islamiento. (*Isolation*)
- **D**urabilidad.

(**ACID** por sus siglas en inglés).



## Transacciones

Garantiza que cualquier trabajo llevado a cabo en una transacción, incluso si se hace por etapas, sea aplicado a la base de datos de forma segura, y **sin interferencia de otras conexiones**, cuando sea consignado.

Mejora la eficiencia de las actualizaciones.





# Transacciones

Una transacción se implementa en 2 pasos:

1. `beginTransaction`
  2. `commit` o `rollback`
- 

# beginTransaction

```
<?php
    $db = new PDO($host, $db_user, $db_pass);

    // desactiva el modo auto-commit
    $db->beginTransaction();

    // código a ejecutar en la base de datos
    $stmt = $db->exec(" DELETE FROM movies
    WHERE title = 'Avatar' ");

    $db->commit();
```



## exec

Ejecuta una sentencia SQL y devuelve el número de filas afectadas por la sentencia.

```
$stmt = $db->exec(" DELETE FROM movies WHERE  
title = 'Avatar' ");
```



# commit

```
<?php
    $db = new PDO($host, $db_user, $db_pass);

    // desactiva el modo auto-commit
    $db->beginTransaction();

    // código a ejecutar en la base de datos
    $stmt = $db->exec(" DELETE FROM movies WHERE title
    = 'Avatar' ");

    $db->commit();

    // Desde aquí ya se habilita el auto-commit
    nuevamente
```

# rollBack

```
<?php
    $db = new PDO($host, $db_user, $db_pass);

    // desactiva el modo auto-commit
    $db->beginTransaction();

    // código a ejecutar en la base de datos
    $stmt = $db->exec(" DELETE FROM movies
    WHERE title = 'Avatar' ");

    // hace roll back
    $db->rollBack();
```

?>

# Con manejo de errores

```
<?php
$db->beginTransaction();

try {
    $db->exec(" DELETE FROM movies WHERE title = 'Avatar'
");
    $db->commit();
}
catch( PDOException $Exception ) {
    // Houston tenemos un problema
    $db->rollBack();
    echo $Exception->getMessage();
}
```

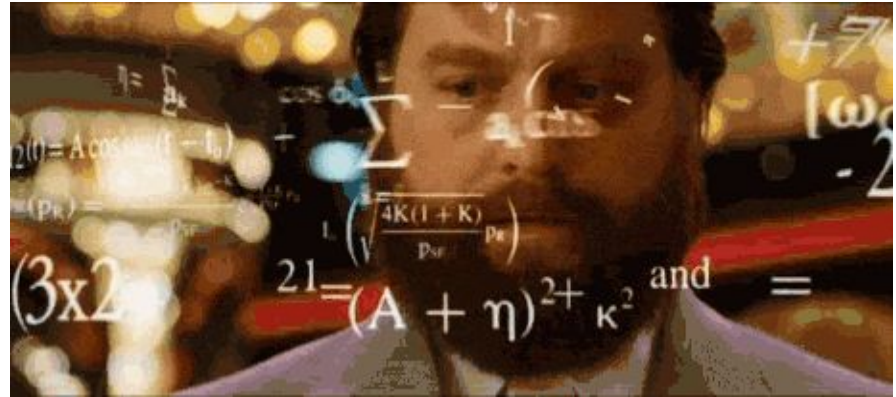
**¡A practicar!**



```
<?php  
    echo "¡Hora de practicar!";  
?>
```



# ¡Gracias!



## ¿Preguntas?