

INVESTIGACION

Juan Pablo Montoya Bedoya

QUIPUX

JAVA-MYSQL

2024

Contenido

INICIO DEL TALLER	2
1. HAVING	2
2. TABLAS.....	3
3. VISTAS	3
4. DISPARADORES	3
5. SECUENCIAS.....	4
6. PROCEDIMIENTO	4
7. FUNCIÓN	4
8. INDEX.....	5
9. ATOMICIDAD.....	5

INICIO DEL TALLER

1. HAVING

En mi sistema de biblioteca, uso HAVING cuando quiero analizar grupos de datos. Por ejemplo, necesitaba saber qué autores tienen más de 3 libros prestados para organizar mejor el inventario. HAVING me permite hacer este filtro después de agrupar, algo que aprendí que WHERE no puede hacer.

```
SELECT autor, COUNT(*) AS libros_prestados
FROM prestamos
GROUP BY autor
HAVING COUNT(*) > 3;
```

2. TABLAS

Las tablas son la base de mi sistema. Cuando empecé, me di cuenta que necesitaba una estructura clara para guardar la información de los libros. Es como crear cajones organizados donde cada libro tiene su lugar específico con sus datos importantes.

```
CREATE TABLE libros (  
  id INT,  
  titulo VARCHAR(50),  
  autor VARCHAR(50)  
);
```

3. VISTAS

Las vistas facilitan mucho el trabajo. Por ejemplo, crear una vista que me muestra solo los libros disponibles para préstamo. Así no tengo que escribir la misma consulta cada vez que alguien viene a buscar un libro.

```
CREATE VIEW libros_prestables AS  
SELECT titulo, autor  
FROM libros  
WHERE estado = 'disponible';
```

4. DISPARADORES

Los disparadores son como mis asistentes automáticos. Cuando alguien devuelve un libro, necesito que varias cosas ocurran automáticamente. El disparador se encarga de todo sin que yo tenga que hacerlo manualmente.

```
CREATE TRIGGER libro_devuelto
AFTER UPDATE ON prestamos
BEGIN
    UPDATE libros SET disponible = true;
END;
```

5. SECUENCIAS

Las secuencias las uso para no tener que inventar números de registro. Cada vez que llega un libro nuevo, la secuencia me da automáticamente el siguiente número disponible.

```
CREATE SEQUENCE libro_id
START WITH 1;
```

6. PROCEDIMIENTO

Creé un procedimiento para el préstamo de libros porque es algo que hago muchas veces al día. En vez de escribir varias consultas, el procedimiento hace todo de una vez.

```
CREATE PROCEDURE prestar(
    libro_id INT,
    estudiante_id INT
)
BEGIN
    INSERT INTO prestamos VALUES(libro_id, estudiante_id);
END;
```

7. FUNCIÓN

Tengo una función que me ayuda a calcular fechas de devolución. Le doy la fecha del préstamo y me dice cuándo debe volver el libro, considerando que presto por una semana.

```
CREATE FUNCTION fecha_devolver(fecha DATE)
RETURNS DATE
BEGIN
    RETURN fecha + 7;
END;
```

8. INDEX

Los índices los uso para hacer búsquedas más rápidas. Como los estudiantes suelen buscar por título o autor, creé índices para estas columnas.

```
CREATE INDEX idx_busqueda
ON libros(titulo);
```

9. ATOMICIDAD

La atomicidad es crucial cuando presto un libro. Necesito asegurarme que todo el proceso se complete correctamente o que no se haga nada.

```
BEGIN TRANSACTION;
    UPDATE libros SET prestado = true;
    INSERT INTO prestamos VALUES (libro_id, alumno_id);
COMMIT;
```