**Procedimientos Almacenados**

**VISTAS**

**1. Muestra detalles del contenido con nombre de estado, tipo, país y cantidad de episodios.**

CREATE VIEW vista\_contenido\_detallado AS

SELECT c.c\_id, c.c\_titulo, p.p\_nombre AS pais, e.e\_nombre AS estado, c.c\_canepisodios

FROM contenido c

JOIN paises p ON c.c\_pais = p.p\_id

JOIN estado e ON c.c\_estado = e.e\_id;

SELECT \* FROM vista\_contenido\_detallado;

**2. Muestra qué contenidos tiene un usuario en su Biblioteca con título del contenido, país, estado de lectura y fecha.**

CREATE VIEW vista\_biblioteca\_usuario\_detalle AS

SELECT b.b\_id, u\_id, c.c\_titulo, p.p\_nombre AS pais, b.b\_estado, b.b\_fecha

FROM biblioteca b

JOIN contenido c ON b.c\_id = c.c\_id

JOIN paises p ON c.c\_pais = p.p\_id;

SELECT \* FROM vista\_biblioteca\_usuario\_detalle;

**3. Muestra reportes realizados por los usuarios con nombre del país, título, motivo y fecha.**

CREATE VIEW vista\_reportes\_completos AS

SELECT r.r\_id, c.c\_titulo, p.p\_nombre AS pais, r.motivo\_r, r.r\_fecha\_creacion

FROM reportes r

JOIN contenido c ON r.c\_id = c.c\_id

JOIN paises p ON c.c\_pais = p.p\_id;

SELECT \* FROM vista\_reportes\_completos;

**FUNCIONES**

**1. Cuenta cuántos contenidos hay según estado, tipo y país.**

DELIMITER //

CREATE FUNCTION fn\_total\_contenido\_estado(

v\_estado INT,

v\_tipo INT,

v\_pais INT

) RETURNS INT

BEGIN

DECLARE total INT;

SELECT COUNT(\*) INTO total

FROM contenido

WHERE c\_estado = v\_estado AND c\_tipo = v\_tipo AND c\_pais = v\_pais;

RETURN total;

END //

DELIMITER ;

SELECT fn\_total\_contenido\_estado(1, 1, 3) AS total\_contenido;

**2. Retorna el nombre de un estado en específico.**

DELIMITER //

CREATE FUNCTION fn\_nombre\_estado\_contenido(

v\_c\_id INT

) RETURNS VARCHAR(50)

BEGIN

DECLARE nombre\_estado VARCHAR(50);

SELECT e.e\_nombre INTO nombre\_estado

FROM contenido c

JOIN estado e ON c.c\_estado = e.e\_id

WHERE c.c\_id = v\_c\_id;

RETURN nombre\_estado;

END //

DELIMITER ;

SELECT fn\_nombre\_estado\_contenido(5) AS estado\_contenido;

**3. Verifica si un contenido tiene un género específico.**

DELIMITER //

CREATE FUNCTION fn\_tiene\_genero(

v\_c\_id INT,

v\_g\_id INT

) RETURNS BOOLEAN

BEGIN

DECLARE existe INT;

SELECT COUNT(\*) INTO existe

FROM contenido\_generos

WHERE c\_id = v\_c\_id AND g\_id = v\_g\_id;

RETURN existe > 0;

END //

DELIMITER ;

SELECT fn\_tiene\_genero(8, 6) AS tiene\_genero;

**PROCEDIMIENTOS**

**1. Filtra contenido por país, estado y tipo.**

DELIMITER //

CREATE PROCEDURE sp\_filtrar\_contenido\_avanzado(

IN v\_pais INT,

IN v\_estado INT,

IN v\_tipo INT

)

BEGIN

SELECT c.c\_id, c.c\_titulo, c.c\_canepisodios, p.p\_nombre, e.e\_nombre

FROM contenido c

JOIN paises p ON c.c\_pais = p.p\_id

JOIN estado e ON c.c\_estado = e.e\_id

WHERE c.c\_pais = v\_pais AND c.c\_estado = v\_estado AND c.c\_tipo = v\_tipo;

END //

DELIMITER ;

CALL sp\_filtrar\_contenido\_avanzado(3, 1, 1);

**2. Consulta reportes hechos por un usuario sobre contenidos de un país y de un tipo específico.**

DELIMITER //

CREATE PROCEDURE sp\_reportes\_por\_usuario(

IN v\_usuario INT,

IN v\_pais INT,

IN v\_tipo INT

)

BEGIN

SELECT r.r\_id, r.r\_descripcion, c.c\_titulo, p.p\_nombre

FROM reportes r

JOIN contenido c ON r.c\_id = c.c\_id

JOIN paises p ON c.c\_pais = p.p\_id

WHERE r.u\_id = v\_usuario AND c.c\_pais = v\_pais AND c.c\_tipo = v\_tipo;

END //

DELIMITER ;

CALL sp\_reportes\_por\_usuario(10, 2, 2);

**3. Muestra biblioteca por usuario, estado de la lectura y país del contenido.**

DELIMITER //

CREATE PROCEDURE sp\_biblioteca\_filtrada(

IN v\_usuario INT,

IN v\_estado ENUM('favorito','leyendo','finalizado'),

IN v\_pais INT

)

BEGIN

SELECT b.b\_id, c.c\_titulo, b.b\_estado, p.p\_nombre

FROM biblioteca b

JOIN contenido c ON b.c\_id = c.c\_id

JOIN paises p ON c.c\_pais = p.p\_id

WHERE b.u\_id = v\_usuario AND b.b\_estado = v\_estado AND c.c\_pais = v\_pais;

END //

DELIMITER ;

CALL sp\_biblioteca\_filtrada(7, 'leyendo', 1);

**4. Busca contenido según palabras en el título, género y estado.**

DELIMITER //

CREATE PROCEDURE sp\_buscar\_contenido\_titulo\_genero\_estado(

IN v\_palabra VARCHAR(100),

IN v\_genero\_id INT,

IN v\_estado\_id INT

)

BEGIN

SELECT DISTINCT c.c\_id, c.c\_titulo, c.c\_descripcion, e.e\_nombre, g.g\_nombre

FROM contenido c

JOIN estado e ON c.c\_estado = e.e\_id

JOIN contenido\_generos cg ON c.c\_id = cg.c\_id

JOIN generos g ON cg.g\_id = g.g\_id

WHERE c.c\_titulo LIKE CONCAT('%', v\_palabra, '%')

AND cg.g\_id = v\_genero\_id

AND c.c\_estado = v\_estado\_id;

END //

DELIMITER ;

CALL sp\_buscar\_contenido\_titulo\_genero\_estado('slayer', 6, 1);

**5. Filtra mangas por mínimo de páginas, número de capítulo y país del contenido.**

DELIMITER //

CREATE PROCEDURE sp\_listar\_manga\_por\_paginas\_y\_capitulo(

IN v\_min\_paginas INT,

IN v\_capitulo INT,

IN v\_pais INT

)

BEGIN

SELECT m.m\_id, c.c\_titulo, m.m\_capitulo, m.m\_numeropag, p.p\_nombre

FROM manga m

JOIN contenido c ON m.c\_id = c.c\_id

JOIN paises p ON c.c\_pais = p.p\_id

WHERE m.m\_numeropag >= v\_min\_paginas

AND m.m\_capitulo = v\_capitulo

AND c.c\_pais = v\_pais;

END //

DELIMITER ;

CALL sp\_listar\_manga\_por\_paginas\_y\_capitulo(20, 3, 3);

**6. Muestra los contenidos más agregados en bibliotecas por estado, país y tipo.**

DELIMITER //

CREATE PROCEDURE sp\_contenido\_popular\_por\_biblioteca(

IN v\_estado ENUM('favorito','leyendo','finalizado'),

IN v\_pais INT,

IN v\_tipo INT

)

BEGIN

SELECT c.c\_id, c.c\_titulo, COUNT(b.b\_id) AS veces\_guardado, p.p\_nombre

FROM biblioteca b

JOIN contenido c ON b.c\_id = c.c\_id

JOIN paises p ON c.c\_pais = p.p\_id

WHERE b.b\_estado = v\_estado

AND c.c\_pais = v\_pais

AND c.c\_tipo = v\_tipo

GROUP BY c.c\_id, c.c\_titulo, p.p\_nombre

ORDER BY veces\_guardado DESC;

END //

DELIMITER ;

CALL sp\_contenido\_popular\_por\_biblioteca('favorito', 1, 1);

**7. Busca reportes según rango de fechas, motivo y país del contenido.**

DELIMITER //

CREATE PROCEDURE sp\_reporte\_contenido\_por\_fecha\_motivo(

IN v\_fecha\_ini DATE,

IN v\_fecha\_fin DATE,

IN v\_motivo INT,

IN v\_pais INT

)

BEGIN

SELECT r.r\_id, c.c\_titulo, r.r\_descripcion, r.r\_fecha\_creacion, p.p\_nombre

FROM reportes r

JOIN contenido c ON r.c\_id = c.c\_id

JOIN paises p ON c.c\_pais = p.p\_id

WHERE r.r\_fecha\_creacion BETWEEN v\_fecha\_ini AND v\_fecha\_fin

AND r.motivo\_r = v\_motivo

AND c.c\_pais = v\_pais;

END //

DELIMITER ;

CALL sp\_reporte\_contenido\_por\_fecha\_motivo('2025-06-01', '2025-06-30', 1, 3);

**8. Filtra contenido según estado, tipo y número mínimo de capítulos.**

DELIMITER //

CREATE PROCEDURE sp\_contenido\_estado\_y\_capitulos(

IN v\_estado INT,

IN v\_tipo INT,

IN v\_min\_capitulos INT

)

BEGIN

SELECT c.c\_id, c.c\_titulo, c.c\_canepisodios, e.e\_nombre, t.tipo\_nombre

FROM contenido c

JOIN estado e ON c.c\_estado = e.e\_id

JOIN (

SELECT 1 AS tipo\_id, 'Anime' AS tipo\_nombre

UNION ALL

SELECT 2, 'Manga'

UNION ALL

SELECT 3, 'Novela'

) AS t ON c.c\_tipo = t.tipo\_id

WHERE c.c\_estado = v\_estado

AND c.c\_tipo = v\_tipo

AND c.c\_canepisodios >= v\_min\_capitulos;

END //

DELIMITER ;

CALL sp\_contenido\_estado\_y\_capitulos(2, 2, 100);