

CREACIÓN BASES DE DATOS Y CONSULTAS SQL

Presentado por:

CARLOS DAVID ESCAMILLA

Presentado a:

ING. HELVER ALEXANDER LIMAS SUAREZ

Asignatura:

BASES DE DATOS

TECNOLOGÍA EN PROGRAMACIÓN DE SISTEMAS INFORMÁTICOS

FACULTAD DE ESTUDIOS A DISTANCIA

UNIVERSIDAD PEDAGÓGICA Y TECNOLÓGICA DE COLOMBIA

TUNJA

2025

SCRIPTS PARA LA CREACIÓN DE LAS 7 TABLAS.

1. TABLA AUTOR

```
CREATE TABLE TbAuthor (  
    idautor INTEGER PRIMARY KEY ASC,  
    nomautor TEXT(100) NOT NULL,  
    paisautor TEXT(40) NOT NULL,  
    fecnacautor NUMERIC NOT NULL,  
    emailautor TEXT(100),  
    idiomaautor TEXT(100) NOT NULL  
);
```

2. TABLA EDITORIAL

```
CREATE TABLE TbEditorial (  
    id_editorial INTEGER PRIMARY KEY ASC AUTOINCREMENT,  
    nom_editorial TEXT (500) NOT NULL,  
    dir_editorial TEXT (500) NOT NULL,  
    tel_editorial TEXT (500),  
    ema_editorial TEXT (500),  
    web_editorial TEXT (500));
```

3. TABLA LIBRO

```
CREATE TABLE Tblibro (  
    isbn INTEGER PRIMARY KEY,  
    título TEXT NOT NULL,  
    formato TEXT NOT NULL,  
    No_páginas INTEGER NOT NULL,  
    año INTEGER NOT NULL,  
    votos INTEGER NOT NULL,  
    editorial INTEGER NOT NULL,  
    FOREIGN KEY (editorial) REFERENCES TbEditorial (id_editorial)
```

ON DELETE CASCADE

);

4. TABLA TRADUCTOR-LIBRO

CREATE TABLE tb_traductor_libro(

isbn_libro INTEGER REFERENCES tb_libro (isbn) ON DELETE CASCADE ON UPDATE CASCADE ,

id_traductor INTEGER REFERENCES tb_traductor (id) ON DELETE CASCADE ON UPDATE CASCADE,

rol TEXT(100),

fecha_entrega INTEGER DEFAULT DATE,

nota TEXT(500),

PRIMARY KEY (

isbn_libro ASC,

id_traductor ASC)

);

5. TABLA CALIFICACIÓN

CREATE TABLE TbCalificacion (

id_calificacion INTEGER PRIMARY KEY AUTOINCREMENT,

valor_calificacion REAL,

com_calificacion TEXT(100),

fecha_calificacion TEXT DEFAULT (DATE('now')),

isbn INTEGER,

FOREIGN KEY (isbn) REFERENCES TbLibro (isbn)

ON DELETE CASCADE

ON UPDATE CASCADE

);

6. TABLA TRADUCTOR

CREATE TABLE TbTraductor (

id_traductor INTEGER PRIMARY KEY ASC AUTOINCREMENT,

nomape_traductor TEXT (100) NOT NULL,

idorigen_traductor TEXT (100) NOT NULL,

```
idiestino_traductor TEXT (100) NOT NULL,  
email_traductor TEXT (100));
```

7. TABLA AUTOR-LIBRO

```
CREATE TABLE tb_autor_libro(  
isbn_libro INTEGER REFERENCES tb_libro (isbn) ON DELETE CASCADE ON UPDATE CASCADE ,  
id_autor INTEGER REFERENCES tb_autor (id) ON DELETE CASCADE ON UPDATE CASCADE,  
categoria TEXT(30),  
sinopsis TEXT(5000),  
PRIMARY KEY (  
isbn_libro ASC,  
id_autor ASC)  
);
```

CONSULTAS SQL

Agregación por grupos con GROUP BY

- **Encuentra la calificación promedio de cada libro**

```
SELECT
    c.isbn,
    l.título,
    AVG(c.valor_calificacion) AS calificacion_promedio
FROM
    TbCalificacion c
JOIN
    TbLibro l ON c.isbn = l.isbn
GROUP BY
    c.isbn, l.título;
```

- **une títulos**

```
SELECT
    l.título,
    AVG(c.valor_calificacion) AS promedio_calificacion
FROM
    TbLibro l
JOIN
    TbCalificacion c ON l.isbn = c.isbn
GROUP BY
    l.título;
```

- **Selecciona libros con una calificación de 4.0 o superior**

```
SELECT
```

```

I.título,
AVG(c.valor_calificacion) AS promedio_calificacion
FROM
    TbLibro l
JOIN
    TbCalificacion c ON l.isbn = c.isbn
GROUP BY
    l.título
HAVING
    AVG(c.valor_calificacion) >= 4.0;

```

Subconsultas:

- **Encuentra todos los libros publicados por MacLehose Press, con ID predefinido**
SELECT título
FROM Tblibro
WHERE editorial = 4;
- **Encuentra todos los libros publicados por MacLehose Press, con una consulta previa**
SELECT l.título
FROM Tblibro l
JOIN TbEditorial e ON l.editorial = e.id_editorial
WHERE e.nom_editorial = 'MacLehose Press';
- **Encuentra todas las calificaciones de 'Whale'**
SELECT c.*
FROM TbCalificacion c
JOIN Tblibro l ON c.isbn = l.isbn
WHERE l.título = 'Whale';
- **Encuentra la calificación promedio de "Whale"**
SELECT AVG(c.valor_calificacion) AS promedio
FROM TbCalificacion c
JOIN Tblibro l ON c.isbn = l.isbn
WHERE l.título = 'Whale';
- **Encuentra al autor de "Pyre "**
SELECT a.nomautor
FROM TbAuthor a
JOIN tb_autor_libro al ON a.idautor = al.id_autor

```
JOIN Tblibro l ON l.isbn = al.isbn_libro
WHERE l.título = 'Pyre';
```

- **Encuentra todos los libros de Fernanda Melchor, usando IN**

```
SELECT l.título
FROM Tblibro l
WHERE l.isbn IN (
    SELECT al.isbn_libro
    FROM tb_autor_libro al
    JOIN TbAuthor a ON a.idautor = al.id_autor
    WHERE a.nomautor = 'Fernanda Melchor'
);
```

- **Usa IN para buscar varios autores**

```
SELECT l.título, a.nomautor
FROM Tblibro l
JOIN tb_autor_libro al ON l.isbn = al.isbn_libro
JOIN TbAuthor a ON al.id_autor = a.idautor
WHERE a.nomautor IN ('Fernanda Melchor', 'Gabriela Cabezón Cámara', 'Juan Gabriel Vásquez');
```

Operaciones con conjuntos:

UNION

- **Selecciona todos los autores, etiquetándolos como autores**
SELECT
 nomautor AS nombre,
 emailautor AS email,
 'Autor' AS rol
FROM TbAuthor;
- **Selecciona todos los traductores, etiquetándolos como traductores**
SELECT
 nomape_traductor AS nombre,
 email_traductor AS email,
 'Traductor' AS rol
FROM TbTraductor;
- **Combina autores y traductores en un conjunto de resultados**
SELECT

```
nomautor AS nombre,
```

```
emailautor AS email,
```

```
'Autor' AS rol  
FROM TbAuthor
```

UNION

```
SELECT  
    nomape_traductor AS nombre,  
    email_traductor AS email,  
    'Traductor' AS rol  
FROM TbTraductor;
```

INTERSECT (Suponga que los nombres son únicos)

- Encuentra autores y traductores

```
SELECT nomautor AS nombre  
FROM TbAuthor
```

INTERSECT

```
SELECT nomape_traductor AS nombre  
FROM TbTraductor;
```

- **Encuentra libros traducidos por Sophie Hughes**

```
SELECT l.título  
FROM Tblibro l  
JOIN tb_traductor_libro tl ON l.isbn = tl.isbn_libro  
JOIN TbTraductor t ON tl.id_traductor = t.id_traductor  
WHERE t.nomape_traductor = 'Sophie Hughes';
```

- **Encuentra libros traducidos por Margaret Jull Costa**

```
SELECT l.título  
FROM Tblibro l  
JOIN tb_traductor_libro tl ON l.isbn = tl.isbn_libro  
JOIN TbTraductor t ON tl.id_traductor = t.id_traductor  
WHERE t.nomape_traductor = 'Margaret Jull Costa';
```

- **Encuentra la intersección de libros**

```
SELECT l.título
```



```
FROM Tblibro l
JOIN tb_traductor_libro tl ON l.isbn = tl.isbn_libro
JOIN TbTraductor t ON tl.id_traductor = t.id_traductor
WHERE t.nomape_traductor = 'Sophie Hughes'
```

INTERSECT

```
SELECT l.título
FROM Tblibro l
JOIN tb_traductor_libro tl ON l.isbn = tl.isbn_libro
JOIN TbTraductor t ON tl.id_traductor = t.id_traductor
WHERE t.nomape_traductor = 'Margaret Jull Costa';
EXCEPT (Suponga que los nombres son únicos)
```

- **Encuentra traductores que no son autores**

```
SELECT nomape_traductor AS nombre
FROM TbTraductor
EXCEPT
SELECT nomautor AS nombre
FROM TbAuthor;
```