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

2025SCRIPTS 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),

idiomaaoutor 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,

idiorigen\_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

l.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;