Let's say we're creating a database for a library, and we want to track information about books, authors, and book loans. Here's how you can create and relate these tables:

create database library\_db;

use library\_db;

**Table 1: Authors**

This table will store information about different authors.

**CREATE TABLE authors (**

**author\_id INT PRIMARY KEY,**

**first\_name VARCHAR(50),**

**last\_name VARCHAR(50),**

**birth\_date DATE**

**);**

**Table 2: Books**

This table will store information about different books, along with the author they are associated with.

**CREATE TABLE books (**

**book\_id INT PRIMARY KEY,**

**title VARCHAR(100),**

**publication\_year YEAR,**

**author\_id INT,**

**FOREIGN KEY (author\_id) REFERENCES authors(author\_id)**

**);**

**Table 3: Book Loans**

This table will store information about loans of books to library patrons. It will include information about the book, the borrower, and the loan date.

**CREATE TABLE book\_loans (**

**loan\_id INT PRIMARY KEY,**

**book\_id INT,**

**borrower\_name VARCHAR(100),**

**borrower\_email VARCHAR(255), -- New column for the borrower's email**

**loan\_date DATE,**

**RETURNED BOOLEAN DEFAULT 0,**

**FOREIGN KEY (book\_id) REFERENCES books(book\_id)**

**);**

**In these tables, we have established relationships as follows:**

* **The books table has a foreign key author\_id that references the author\_id in the authors table. This represents the relationship between authors and the books they have written.**
* **The book\_loans table has a foreign key book\_id that references the book\_id in the books table. This represents the relationship between books and the loans made for those books.**

**Insertion of data for authors table**

**INSERT INTO authors (author\_id, first\_name, last\_name, birth\_date)**

**VALUES**

**(1, 'John', 'Doe', '1980-05-15'),**

**(2, 'Jane', 'Smith', '1975-10-22'),**

**(3, 'Michael', 'Johnson', '1992-03-08'),**

**(4, 'Emily', 'Williams', '1988-12-03'),**

**(5, 'Robert', 'Brown', '1971-07-19'),**

**(6, 'Olivia', 'Jones', '1995-09-12'),**

**(7, 'William', 'Davis', '1983-06-28'),**

**(8, 'Ava', 'Wilson', '2000-02-14'),**

**(9, 'James', 'Anderson', '1979-11-07'),**

**(10, 'Sophia', 'Martinez', '1990-04-30'),**

**(11, 'Liam', 'Rodriguez', '1986-08-09'),**

**(12, 'Isabella', 'Garcia', '1974-12-18'),**

**(13, 'Noah', 'Lopez', '1998-01-25'),**

**(14, 'Mia', 'Hernandez', '1989-07-02'),**

**(15, 'Benjamin', 'Smith', '1982-03-21'),**

**(16, 'Charlotte', 'Lee', '1996-10-11'),**

**(17, 'Ethan', 'Clark', '1977-09-05'),**

**(18, 'Amelia', 'Lewis', '1993-05-08'),**

**(19, 'Lucas', 'Hall', '1972-04-03'),**

**(20, 'Luna', 'Young', '1997-11-16'),**

**(21, 'Alexander', 'King', '1985-06-13'),**

**(22, 'Harper', 'Baker', '2002-08-27'),**

**(23, 'Daniel', 'Green', '1991-01-10'),**

**(24, 'Evelyn', 'Adams', '1978-09-23'),**

**(25, 'Jackson', 'Campbell', '1994-12-07');**

**Insertion of data for books table**

**INSERT INTO books (book\_id, title, publication\_year, author\_id)**

**VALUES**

**(1, 'The Book of Codes', 2010, 1),**

**(2, 'A Journey to the Stars', 2005, 2),**

**(3, 'Time Travel Chronicles', 2018, 3),**

**(4, 'Mysteries Unveiled', 2021, 4),**

**(5, 'The Enigma Code', 2008, 5),**

**(6, 'Beyond the Horizon', 2015, 6),**

**(7, 'Echoes of Eternity', 2012, 7),**

**(8, 'Lost and Found', 2019, 8),**

**(9, 'Secrets of the Past', 2006, 9),**

**(10, 'The Hidden Truth', 2014, 10),**

**(11, 'Into the Unknown', 2009, 11),**

**(12, 'Whispers in the Wind', 2016, 12),**

**(13, 'Infinite Possibilities', 2017, 13),**

**(14, 'The Silent Echo', 2004, 14),**

**(15, 'Beneath the Surface', 2011, 15),**

**(16, 'Shadows of the Past', 2020, 16),**

**(17, 'Eternal Flames', 2007, 17),**

**(18, 'The Forgotten Land', 2013, 18),**

**(19, 'Boundless Skies', 2003, 19),**

**(20, 'Parallel Realms', 2022, 20),**

**(21, 'Ripples in Time', 2018, 21),**

**(22, 'The Wandering Soul', 2001, 22),**

**(23, 'Fragments of Destiny', 2010, 23),**

**(24, 'Beyond the Veil', 2009, 24),**

**(25, 'The Lost Key', 2015, 25);**

**Insertion of data for book\_loans table**

**INSERT INTO book\_loans (loan\_id, book\_id, borrower\_name, borrower\_email, loan\_date, RETURNED)**

**VALUES**

**(1, 1, 'Alice Smith', 'alice.smith@gmail.com', '2023-08-01', 0),**

**(2, 2, 'Bob Johnson', 'bob.johnson@yahoo.com', '2023-07-15', 1),**

**(3, 3, 'Emma Davis', 'emma.davis@hotmail.com', '2023-08-10', 0),**

**(4, 4, 'Daniel Lee', 'daniel.lee@example.com', '2023-07-05', 1),**

**(5, 5, 'Olivia Brown', 'olivia.brown@gmail.com', '2023-08-18', 0),**

**(6, 6, 'William Garcia', 'william.garcia@yahoo.com', '2023-06-20', 0),**

**(7, 7, 'Ava Martinez', 'ava.martinez@example.com', '2023-07-28', 1),**

**(8, 8, 'James Wilson', 'james.wilson@hotmail.com', '2023-08-04', 0),**

**(9, 9, 'Sophia Anderson', 'sophia.anderson@gmail.com', '2023-08-02', 1),**

**(10, 10, 'Liam Taylor', 'liam.taylor@yahoo.com', '2023-07-10', 0),**

**(11, 11, 'Isabella Hernandez', 'isabella.hernandez@example.com', '2023-06-15', 0),**

**(12, 12, 'Noah Lewis', 'noah.lewis@hotmail.com', '2023-07-03', 1),**

**(13, 13, 'Mia Hall', 'mia.hall@gmail.com', '2023-08-06', 0),**

**(14, 14, 'Benjamin Young', 'benjamin.young@example.com', '2023-08-19', 1),**

**(15, 15, 'Charlotte King', 'charlotte.king@hotmail.com', '2023-07-25', 0),**

**(16, 16, 'Ethan Baker', 'ethan.baker@gmail.com', '2023-08-11', 0),**

**(17, 17, 'Amelia Green', 'amelia.green@example.com', '2023-06-12', 1),**

**(18, 18, 'Lucas Adams', 'lucas.adams@hotmail.com', '2023-07-22', 0),**

**(19, 19, 'Luna Campbell', 'luna.campbell@gmail.com', '2023-08-07', 1),**

**(20, 20, 'Alexander Moore', 'alexander.moore@yahoo.com', '2023-08-03', 0),**

**(21, 21, 'Harper Wright', 'harper.wright@hotmail.com', '2023-07-09', 0),**

**(22, 22, 'Daniel Jackson', 'daniel.jackson@example.com', '2023-06-14', 1),**

**(23, 23, 'Evelyn White', 'evelyn.white@gmail.com', '2023-08-05', 0),**

**(24, 24, 'Jackson Brown', 'jackson.brown@hotmail.com', '2023-08-14', 0),**

**(25, 25, 'Sophia Wilson', 'sophia.wilson@yahoo.com', '2023-08-20', 0);**

**Exercise 1: Retrieve the Full Name of Authors**

**Write a query to retrieve the full names (first name and last name concatenated) of all authors.**

**Answer:**

**SELECT CONCAT(first\_name, ' ', last\_name) AS full\_name FROM authors;**

**Explanation: In this exercise, the CONCAT function is used to combine the first\_name and last\_name columns into a single string representing the full name of each author.**

**Exercise 2: Count the Number of Characters in Book Titles**

**Write a query to calculate the length (number of characters) of each book's title.**

**Answer:**

**SELECT title, LENGTH(title) AS title\_length FROM books;**

**Explanation: The LENGTH function returns the number of characters in the given string. In this case, it's used to calculate the length of book titles.**

**Exercise 3: Convert Book Titles to Uppercase**

**Write a query to retrieve the book titles in uppercase.**

**Answer:**

**SELECT title, UPPER(title) AS uppercase\_title FROM books;**

**Explanation: The UPPER function converts a string to uppercase. Here, it's applied to book titles.**

**Exercise 4: Extract First Name from Borrower Names**

**Write a query to extract the first name from each borrower's name in the book\_loans table.**

**Answer:**

**SELECT borrower\_name, SUBSTRING\_INDEX(borrower\_name, ' ', 1) AS first\_name FROM book\_loans;**

**Explanation: The SUBSTRING\_INDEX function is used to extract the first name from the borrower\_name column. It extracts the part of the string before the first space.**

**Exercise 5: Replace Spaces with Underscores in Author Names**

**Write a query to replace spaces with underscores in author names.**

**Answer:**

**SELECT author\_id, first\_name, REPLACE(first\_name, ' ', '\_') AS modified\_first\_name FROM authors;**

**Explanation: The REPLACE function replaces occurrences of a substring (in this case, a space) with another substring (in this case, an underscore) within a given string.**

**Exercise 6: Extract Last Name from Author Names**

**Write a query to extract the last name from each author's full name.**

**Answer:**

**SELECT author\_id, CONCAT(first\_name, ' ', last\_name) AS full\_name, SUBSTRING\_INDEX(CONCAT(first\_name, ' ', last\_name), ' ', -1) AS last\_name FROM authors;**

**Explanation: The SUBSTRING\_INDEX function is used to extract the last part of the name after the last space, effectively extracting the last name.**

**Exercise 7: Display Borrower Names in Title Case**

**Write a query to display borrower names in title case (first letter of each word capitalized).**

**Answer:**

**SELECT borrower\_name, CONCAT( UPPER(SUBSTRING(borrower\_name, 1, 1)), LOWER(SUBSTRING(borrower\_name, 2)), IFNULL(CONCAT(' ', UPPER(SUBSTRING\_INDEX(SUBSTRING(borrower\_name, 2), ' ', 1))), '') ) AS title\_case\_name FROM book\_loans;**

**Explanation: This query uses various string functions to convert borrower names into title case. It capitalizes the first letter of each word while handling cases where names include spaces.**

**Exercise 8: Calculate the Average Title Length**

**Write a query to calculate the average length of book titles.**

**Answer:**

**SELECT AVG(LENGTH(title)) AS average\_title\_length FROM books;**

**Explanation: The AVG function calculates the average of the lengths of book titles.**

**Exercise 9: Extract Domains from Borrower Emails**

**Write a query to extract the domains from borrower email addresses.**

**Answer:**

**SELECT borrower\_name, email, SUBSTRING\_INDEX(email, '@', -1) AS domain FROM book\_loans;**

**Explanation: The SUBSTRING\_INDEX function extracts the part of the email address after the '@' symbol, effectively extracting the domain.**

**Exercise 10: Concatenate Book Title and Author's Last Name**

**Write a query to concatenate the book title and the last name of the book's author for all books.**

**Answer:**

**SELECT b.title, a.last\_name, CONCAT(b.title, ' by ', a.last\_name) AS title\_with\_author FROM books b JOIN authors a ON b.author\_id = a.author\_id;**

**Explanation: This query joins the books and authors tables and uses the CONCAT function to create a string that combines the book title and the author's last name.**

**Exercise 11: Count the Number of Borrowers with Each First Letter**

**Write a query to count the number of borrowers whose names start with each letter of the alphabet.**

**Answer:**

**SELECT UPPER(SUBSTRING(borrower\_name, 1, 1)) AS first\_letter, COUNT(\*) AS count FROM book\_loans GROUP BY first\_letter;**

**Explanation: This query uses the SUBSTRING and UPPER functions to extract the first letter of each borrower's name and then counts how many borrowers have names starting with each letter.**

**Exercise 12: Display Borrowers with Long Names**

**Write a query to display borrowers whose names have more than 10 characters.**

**Answer:**

**SELECT borrower\_name FROM book\_loans WHERE LENGTH(borrower\_name) > 10;**

**Explanation: The LENGTH function is used to determine the length of the borrower names. The WHERE clause filters the results to show only those with more than 10 characters.**

**Exercise 13: Replace 'Chronicles' in Book Titles**

**Write a query to replace the word 'Chronicles' in book titles with 'Saga'.**

**Answer:**

**SELECT title, REPLACE(title, 'Chronicles', 'Saga') AS updated\_title FROM books;**

**Explanation: The REPLACE function is used to replace occurrences of 'Chronicles' with 'Saga' in book titles.**

**Exercise 14: Extract Borrower Names without Last Names**

**Write a query to extract the first names of borrowers whose names do not include a last name.**

**Answer:**

**SELECT borrower\_name, IF(INSTR(borrower\_name, ' ') = 0, borrower\_name, SUBSTRING\_INDEX(borrower\_name, ' ', 1)) AS first\_name FROM book\_loans;**

**Explanation: The INSTR function is used to find the position of the first space in the borrower's name. If there is no space, the entire name is considered a first name.**

**Exercise 15: Combine Author Names into a List**

**Write a query to combine the first names of authors into a comma-separated list.**

**Answer:**

**SELECT GROUP\_CONCAT(first\_name SEPARATOR ', ') AS author\_list FROM authors;**

**Explanation: The GROUP\_CONCAT function combines values from multiple rows into a single string, separated by a specified separator (in this case, a comma and a space).**

**Exercise 16: Display Author Names in Reverse Order**

**Write a query to display author names in reverse order (last name followed by first name).**

**Answer:**

**SELECT author\_id, CONCAT(last\_name, ', ', first\_name) AS reverse\_name FROM authors;**

**Explanation: The CONCAT function is used to concatenate the last name, comma, space, and first name to reverse the order of author names.**

**Exercise 17: Remove Duplicates from Borrower Names**

**Write a query to retrieve a list of unique borrower names, removing any duplicates.**

**Answer:**

**SELECT DISTINCT borrower\_name FROM book\_loans;**

**Explanation: The DISTINCT keyword is used to retrieve unique values from the borrower\_name column.**

**Exercise 18: Display Books Published in the 21st Century**

**Write a query to display book titles that were published in the 21st century (year 2001 and later).**

**Answer:**

**SELECT title FROM books WHERE publication\_year >= 2001;**

**Explanation: The WHERE clause filters the results to include only books with a publication year of 2001 or later.**

**Exercise 19: Convert Author Last Names to Uppercase**

**Write a query to retrieve author last names in uppercase.**

**Answer:**

**SELECT author\_id, UPPER(last\_name) AS uppercase\_last\_name FROM authors;**

**Explanation: The UPPER function converts the last\_name column values to uppercase.**

**Exercise 20: Extract Book Titles without Leading Articles**

**Write a query to extract book titles without leading articles ('The', 'A', 'An').**

**Answer:**

**select \* from books**

**where substring\_index(title, ' ', 1) not in('The', 'A', 'An');**

**SELECT title, TRIM(SUBSTRING(title, IF(title LIKE 'The %', 5, IF(title LIKE 'A %', 3, IF(title LIKE 'An %', 4, 1)))) AS extracted\_title FROM books;**

**Explanation: The SUBSTRING function extracts the title starting from the position after the leading article if present. The TRIM function is used to remove any leading spaces.**

**Exercise 21: Display Borrower Names with a Specific Pattern**

**Write a query to display borrower names that start with 'A' and end with 's'.**

**Answer:**

**SELECT borrower\_name FROM book\_loans WHERE borrower\_name LIKE 'A%s';**

**Explanation: The LIKE keyword is used with the pattern 'A%s' to match borrower names that start with 'A' and end with 's'.**

**Exercise 22: Concatenate Author First Names and Last Names**

**Write a query to concatenate the first name and last name of authors, separated by a space.**

**Answer:**

**SELECT author\_id, CONCAT(first\_name, ' ', last\_name) AS full\_name FROM authors;**

**Explanation: The CONCAT function is used to concatenate the first\_name, space, and last\_name columns.**

**Exercise 23: Count Borrowers with the Same First Names**

**Write a query to count the number of borrowers who have the same first name.**

**Answer:**

**SELECT first\_name, COUNT(\*) AS borrower\_count FROM ( SELECT SUBSTRING\_INDEX(borrower\_name, ' ', 1) AS first\_name FROM book\_loans ) AS borrower\_first\_names GROUP BY first\_name HAVING borrower\_count > 1;**

**Explanation: This query first extracts the first names of borrowers using the SUBSTRING\_INDEX function. Then, it groups the results by first name and uses the HAVING clause to show only those with more than one borrower.**

**Exercise 24: Display Book Titles with 'Code' in the Title**

**Write a query to display book titles that contain the word 'Code' anywhere in the title.**

**Answer:**

**SELECT title FROM books WHERE title LIKE '%Code%';**

**Explanation: The % symbol in the LIKE pattern acts as a wildcard, matching any characters before or after 'Code'.**

**Exercise 25: Display Borrower Names without Leading and Trailing Spaces**

**Write a query to display borrower names without any leading or trailing spaces.**

**Answer:**

**SELECT borrower\_name, TRIM(borrower\_name) AS trimmed\_name FROM book\_loans;**

**Explanation: The TRIM function is used to remove any leading or trailing spaces from the borrower\_name column values.**

**Exercise 26: Display Authors with Names of a Specific Length**

**Write a query to display authors whose first names have exactly 4 characters.**

**Answer:**

**SELECT author\_id, first\_name**

**FROM authors**

**WHERE LENGTH(first\_name) = 4;**

**Exercise 27: Extract Borrower Email Domains**

**Write a query to extract the domains from borrower email addresses in the book\_loans table.**

**Answer:**

**SELECT borrower\_name, email,**

**SUBSTRING\_INDEX(email, '@', -1) AS email\_domain**

**FROM book\_loans;**

**Exercise 28: Display Titles without Leading or Trailing Spaces**

**Write a query to display book titles without any leading or trailing spaces.**

**Answer:**

**SELECT title,**

**TRIM(title) AS trimmed\_title**

**FROM books;**

**Exercise 29: Count Borrower Names with a Specific Substring**

**Write a query to count the number of borrowers whose names contain the substring 'son'.**

**Answer:**

**SELECT COUNT(\*) AS count**

**FROM book\_loans**

**WHERE borrower\_name LIKE '%son%';**

**Exercise 30: Display Authors with Names in Title Case**

**Write a query to display author names in title case (first letter of each word capitalized).**

**Answer:**

**SELECT author\_id,**

**CONCAT(**

**UPPER(SUBSTRING(first\_name, 1, 1)),**

**LOWER(SUBSTRING(first\_name, 2)),**

**' ',**

**UPPER(SUBSTRING(last\_name, 1, 1)),**

**LOWER(SUBSTRING(last\_name, 2))**

**) AS title\_case\_name**

**FROM authors;**

**Exercise 31: Concatenate Borrower Names with Loan IDs**

**Write a query to concatenate the borrower names with their corresponding loan IDs.**

**Answer:**

**SELECT loan\_id, borrower\_name,**

**CONCAT(borrower\_name, ' (Loan ID: ', loan\_id, ')') AS borrower\_with\_id**

**FROM book\_loans;**

**Exercise 32: Calculate Average Borrower Name Length**

**Write a query to calculate the average length of borrower names.**

**Answer:**

**SELECT AVG(LENGTH(borrower\_name)) AS average\_name\_length**

**FROM book\_loans;**

**Exercise 33: Display Titles with More Than One Word**

**Write a query to display book titles that consist of more than one word.**

**Answer:**

**SELECT title**

**FROM books**

**WHERE title LIKE '% %';**

**Exercise 34: Display Borrowers with Names Starting with 'J'**

**Write a query to display borrower names that start with the letter 'J'.**

**Answer:**

**SELECT borrower\_name**

**FROM book\_loans**

**WHERE borrower\_name LIKE 'J%';**

**Exercise 35: Concatenate Author Names with Birth Years**

**Write a query to concatenate author names with their birth years in parentheses.**

**Answer:**

**SELECT author\_id, first\_name, last\_name, birth\_date,**

**CONCAT(first\_name, ' ', last\_name, ' (', YEAR(birth\_date), ')') AS full\_name\_with\_birth\_year**

**FROM authors;**

**Exercise 36: Display Titles with 'Mysteries' in the Title**

**Write a query to display book titles that contain the word 'Mysteries' anywhere in the title.**

**Answer:**

**SELECT title**

**FROM books**

**WHERE title LIKE '%Mysteries%';**

**Exercise 37: Remove Whitespace from Borrower Names**

**Write a query to remove all whitespace characters from borrower names.**

**Answer:**

**SELECT borrower\_name,**

**REPLACE(borrower\_name, ' ', '') AS name\_without\_spaces**

**FROM book\_loans;**

**Exercise 38: Calculate the Total Length of Borrower Names**

**Write a query to calculate the total length of all borrower names combined.**

**Answer:**

**SELECT SUM(LENGTH(borrower\_name)) AS total\_name\_length**

**FROM book\_loans;**

**Exercise 39: Display Titles with 'Journey' or 'Quest' in the Title**

**Write a query to display book titles that contain either 'Journey' or 'Quest' in the title.**

**Answer:**

**SELECT title**

**FROM books**

**WHERE title LIKE '%Journey%' OR title LIKE '%Quest%';**

**Exercise 40: Display Authors with More Than One Word in Name**

**Write a query to display authors whose names consist of more than one word.**

**Answer:**

**SELECT author\_id, first\_name, last\_name**

**FROM authors**

**WHERE CONCAT(first\_name, ' ', last\_name) LIKE '% %';**

**Exercise 41: Capitalize Borrower Names**

**Write a query to capitalize the first letter of each word in borrower names.**

**Answer:**

**SELECT borrower\_name,**

**CONCAT(**

**UPPER(SUBSTRING\_INDEX(borrower\_name, ' ', 1)),**

**' ',**

**UPPER(SUBSTRING\_INDEX(SUBSTRING(borrower\_name, LENGTH(SUBSTRING\_INDEX(borrower\_name, ' ')) + 2), ' ', -1))**

**) AS capitalized\_name**

**FROM book\_loans;**

**Exercise 42: Display Titles with Longest Length**

**Write a query to display the book title(s) with the longest length.**

**Answer:**

**SELECT title**

**FROM books**

**ORDER BY LENGTH(title) DESC**

**LIMIT 1;**

**Exercise 43: Replace 'Journey' with 'Adventure' in Book Titles**

**Write a query to replace occurrences of 'Journey' with 'Adventure' in book titles.**

**Answer:**

**SELECT title,**

**REPLACE(title, 'Journey', 'Adventure') AS updated\_title**

**FROM books;**

**Exercise 44: Display Borrowers with Names Ending in 's'**

**Write a query to display borrower names that end with the letter 's'.**

**Answer:**

**SELECT borrower\_name**

**FROM book\_loans**

**WHERE borrower\_name LIKE '%s';**

**Exercise 45: Combine First and Last Names of Borrowers**

**Write a query to combine the first name and last name of borrowers, separated by a space.**

**Answer:**

**SELECT borrower\_name,**

**CONCAT(SUBSTRING\_INDEX(borrower\_name, ' ', 1), ' ', SUBSTRING\_INDEX(borrower\_name, ' ', -1)) AS full\_name**

**FROM book\_loans;**

**Exercise 46: Display Book Titles without 'The'**

**Write a query to display book titles without the word 'The' at the beginning.**

**Answer:**

**SELECT title,**

**TRIM(REPLACE(title, 'The ', '')) AS updated\_title**

**FROM books**

**WHERE title LIKE 'The %';**

**Exercise 47: Count Digits in Borrower Names**

**Write a query to count the total number of digits in all borrower names combined.**

**Answer:**

**SELECT SUM(LENGTH(borrower\_name) - LENGTH(REPLACE(borrower\_name, 0, ''))) AS total\_digits**

**FROM book\_loans;**

**Exercise 48: Display Book Titles in Reverse Order**

**Write a query to display book titles in reverse order (last character first).**

**Answer:**

**SELECT title,**

**REVERSE(title) AS reversed\_title**

**FROM books;**

**Exercise 49: Remove Special Characters from Borrower Names**

**Write a query to remove all special characters from borrower names.**

**Answer:**

**SELECT borrower\_name,**

**REGEXP\_REPLACE(borrower\_name, '[^a-zA-Z ]', '') AS name\_without\_special\_chars**

**FROM book\_loans;**

**Exercise 50: Display Titles with 'Time' or 'Chronicles' in the Title**

**Write a query to display book titles that contain either 'Time' or 'Chronicles' in the title.**

**Answer:**

**SELECT title**

**FROM books**

**WHERE title LIKE '%Time%' OR title LIKE '%Chronicles%';**