**SQL JOINS**

In SQL, the various types of joins are used to combine data from two or more tables based on a related column between them. Let's explain each type of join using a real-time example of authors and books.

Consider two tables: Authors and Books.

Authors Table: Books Table:

|  |  |
| --- | --- |
| AuthorID | AuthorName |
| 1 | John Smith |
| 2 | Jane Doe |
| 3 | Michael Johnson |
| 4 | Sarah Wilson |

|  |  |  |
| --- | --- | --- |
| BookID | BookName | AuthorID |
| 1 | The Book of Wisdom | 1 |
| 2 | A Tale of Two Cities | 3 |
| 3 | Journey to the Stars | 2 |
| 4 | Programming 101 | 5 |

Now, let's explore each type of join and see the resultant tables.

**1. Inner Join:**

An inner join returns only the matching rows from both tables based on the specified condition.

SQL code:

SELECT Authors.AuthorID, Authors.AuthorName, Books.BookName

FROM Authors

INNER JOIN Books ON Authors.AuthorID = Books.AuthorID;

Resultant Table:

|  |  |  |
| --- | --- | --- |
| AuthorID | AuthorName | BookName |
| 1 | John Smith | The Book of Wisdom |
| 2 | Jane Doe | Journey to the Stars |
| 3 | Michael Johnson | A Tale of Two Cities |

Explanation: The inner join combines the Authors and Books tables based on the matching AuthorID column. It only returns the rows where there is a match in both tables. The book "Programming 101" is not included because there is no corresponding author with AuthorID = 5.

**2. Left Join:**

A left join returns all the rows from the left table and the matching rows from the right table. If there is no match, it returns NULL values for the columns of the right table.

SQL code:

SELECT Authors.AuthorID, Authors.AuthorName, Books.BookName

FROM Authors

LEFT JOIN Books ON Authors.AuthorID = Books.AuthorID;

Resultant Table:

|  |  |  |
| --- | --- | --- |
| AuthorID | AuthorName | BookName |
| 1 | John Smith | The Book of Wisdom |
| 2 | Jane Doe | Journey to the Stars |
| 3 | Michael Johnson | A Tale of Two Cities |
| 4 | Sarah Wilson | NULL |

Explanation: The left join returns all the rows from the Authors table and the matching rows from the Books table. Since there is no matching author for the book "Programming 101", the corresponding values in the BookName column are NULL.

**3. Right Join:**

A right join returns all the rows from the right table and the matching rows from the left table. If there is no match, it returns NULL values for the columns of the left table.

SQL code:

SELECT Authors.AuthorID, Authors.AuthorName, Books.BookName

FROM Authors

RIGHT JOIN Books ON Authors.AuthorID = Books.AuthorID;

Resultant Table:

|  |  |  |
| --- | --- | --- |
| AuthorID | AuthorName | BookName |
| 1 | John Smith | The Book of Wisdom |
| 3 | Michael Johnson | A Tale of Two Cities |
| 2 | Jane Doe | Journey to the Stars |
| NULL | NULL | Programming 101 |

Explanation: A right join returns all the rows from the Books table and the matching rows from the Authors table. The book “Programming 101” doesn’t have a matching author in the ‘Authors’ table, so it includes NULL values for the ‘AuthorID’ and ‘AuthorName’ columns.

**4. Full Join:**

A full join returns all the rows from both the left and right tables. It includes both the matching and non-matching rows. If there is no match, it returns NULL values for the columns of the table that doesn't have a corresponding match.

SQL code:

SELECT Authors.AuthorID, Authors.AuthorName, Books.BookName

FROM Authors

FULL JOIN Books ON Authors.AuthorID = Books.AuthorID;

Resultant Table:

|  |  |  |
| --- | --- | --- |
| AuthorID | AuthorName | BookName |
| 1 | John Smith | The Book of Wisdom |
| 2 | Jane Doe | Journey to the Stars |
| 3 | Michael Johnson | A Tale of Two Cities |
| 4 | Sarah Wilson | NULL |
| NULL | NULL | Programming 101 |

Explanation: The full join combines all the rows from both the Authors and Books tables, including the matching and non-matching rows. Since there is no corresponding author for the book "Programming 101" and no corresponding book for author ID 4, the corresponding values in the columns are NULL.

**5. Cross Join:**

A cross join returns the Cartesian product of both tables. It combines every row from the first table with every row from the second table.

SQL code:

SELECT Authors.AuthorID, Authors.AuthorName, Books.BookName

FROM Authors

CROSS JOIN Books;two tables: "Authors" and "Books".

Resultant Table:

|  |  |  |
| --- | --- | --- |
| AuthorID | AuthorName | BookName |
| 1 | John Smith | The Book of Wisdom |
| 1 | John Smith | A Tale of Two Cities |
| 1 | John Smith | Journey to the Stars |
| 1 | John Smith | Programming 101 |
| 2 | Jane Doe | The Book of Wisdom |
| 2 | Jane Doe | A Tale of Two Cities |
| 2 | Jane Doe | Journey to the Stars |
| 2 | Jane Doe | Programming 101 |
| 3 | Michael Johnson | The Book of Wisdom |
| 3 | Michael Johnson | A Tale of Two Cities |
| 3 | Michael Johnson | Journey to the Stars |
| 3 | Michael Johnson | Programming 101 |
| 4 | Sarah Wilson | The Book of Wisdom |
| 4 | Sarah Wilson | A Tale of Two Cities |
| 4 | Sarah Wilson | Journey to the Stars |
| 4 | Sarah Wilson | Programming 101 |

Explanation: The cross join combines every row from the Authors table with every row from the Books table, resulting in a combination of all possible author-book pairs.

That covers all the different types of joins in SQL with a real-time example using the authors and books scenario.