

-- Task : SQL Joins (Inner, Left, Right, Full)

-- Database: library\_management\_system

use library\_management\_system ;

-- 1 INNER JOIN: Readers who issued books

-- Shows only those readers who have actually issued a book

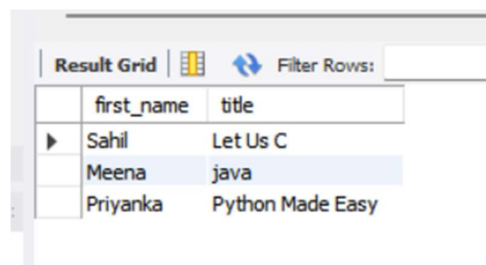
SELECT r.first\_name, b.title

FROM Readers r

INNER JOIN Issue i ON r.reader\_id = i.reader\_id

INNER JOIN Books b ON i.isbn = b.isbn;

Output:-



The screenshot shows a 'Result Grid' window with a 'Filter Rows' button. The grid contains three rows of data. The first row has columns 'first\_name' and 'title'. The second row shows 'Sahil' and 'Let Us C'. The third row shows 'Meena' and 'java'. The fourth row shows 'Priyanka' and 'Python Made Easy'.

first_name	title
Sahil	Let Us C
Meena	java
Priyanka	Python Made Easy

-- 2 LEFT JOIN: All readers (even if they never issued a book)

-- If a reader hasn't issued any book, the title will be NULL

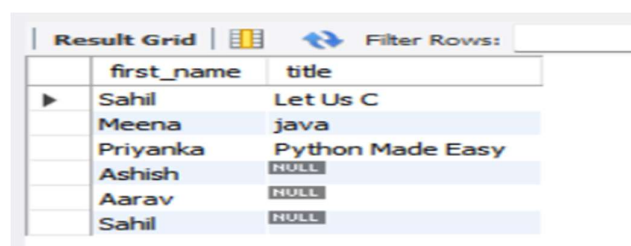
SELECT r.first\_name, b.title

FROM Readers r

LEFT JOIN Issue i ON r.reader\_id = i.reader\_id

LEFT JOIN Books b ON i.isbn = b.isbn;

Output:-



The screenshot shows a 'Result Grid' window with a 'Filter Rows' button. The grid contains six rows of data. The first row has columns 'first\_name' and 'title'. The second row shows 'Sahil' and 'Let Us C'. The third row shows 'Meena' and 'java'. The fourth row shows 'Priyanka' and 'Python Made Easy'. The fifth row shows 'Ashish' and 'NULL'. The sixth row shows 'Aarav' and 'NULL'. The seventh row shows 'Sahil' and 'NULL'.

first_name	title
Sahil	Let Us C
Meena	java
Priyanka	Python Made Easy
Ashish	NULL
Aarav	NULL
Sahil	NULL

-- 3 RIGHT JOIN: All issued books (even if reader info is missing)

-- Shows all issued records and their books

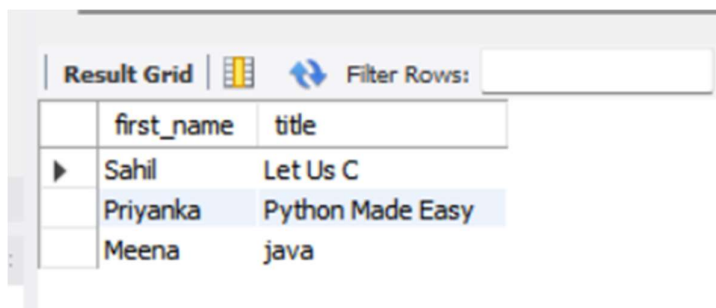
```
SELECT r.first_name, b.title
```

```
FROM Readers r
```

```
RIGHT JOIN Issue i ON r.reader_id = i.reader_id
```

```
RIGHT JOIN Books b ON i.isbn = b.isbn;
```

Output:-



The screenshot shows a 'Result Grid' window with a 'Filter Rows' input field. Below the header, there are three rows of data. The first row has a play button icon in the first column, 'Sahil' in the second column, and 'Let Us C' in the third column. The second row has 'Priyanka' in the second column and 'Python Made Easy' in the third column. The third row has 'Meena' in the second column and 'java' in the third column.

	first_name	title
▶	Sahil	Let Us C
	Priyanka	Python Made Easy
	Meena	java

-- 4 FULL OUTER JOIN: Combine LEFT & RIGHT joins

-- MySQL doesn't directly support FULL JOIN, so we use UNION

```
SELECT r.first_name, b.title
```

```
FROM Readers r
```

```
LEFT JOIN Issue i ON r.reader_id = i.reader_id
```

```
LEFT JOIN Books b ON i.isbn = b.isbn
```

```
UNION
```

```
SELECT r.first_name, b.title
```

```
FROM Readers r
```

```
RIGHT JOIN Issue i ON r.reader_id = i.reader_id
```

```
RIGHT JOIN Books b ON i.isbn = b.isbn;
```

Output:-

Result Grid			Filter Rows:
	first_name	title	
▶	Sahil	Let Us C	
	Meena	java	
	Priyanka	Python Made Easy	
	Ashish	NULL	
	Aarav	NULL	
	Sahil	NULL	

-- 5 CROSS JOIN: Every reader with every book

-- Creates a big list of all possible reader-book pairs

SELECT r.first\_name, b.title

FROM Readers r

CROSS JOIN Books b;

Output:-

Result Grid			Filter Rows:
	first_name	title	
▶	Sahil	java	
	Sahil	Python Made Easy	
	Sahil	Let Us C	
	Meena	java	
	Meena	Python Made Easy	
	Meena	Let Us C	
	Priyanka	java	
	Priyanka	Python Made Easy	
	Priyanka	Let Us C	
	Ashish	java	

Result Grid			Filter Rows:
	first_name	title	
	Priyanka	Let Us C	
	Ashish	java	
	Ashish	Python Made Easy	
	Ashish	Let Us C	
	Aarav	java	
	Aarav	Python Made Easy	
	Aarav	Let Us C	
	Sahil	java	
	Sahil	Python Made Easy	
	Sahil	Let Us C	