

Databases and SQL for Data Science with Python

Eng. Mohamed Soliman

Module 7

Advanced SQL for Data Engineer I

ACID Transactions

What is a transaction?

a **transaction** is any operation that is treated as a single unit of work, which either completes fully or does not complete at all, and leaves the storage system in a consistent state.



Transaction example

BankAccounts

AccountNumber	AccountName	Balance
B001	Rose	300
B002	James	13450
B003	Shoe Shop	124000
B004	Corner Shop	76000

ShoeShop

Product	Stock	Price
Boots	12	200
High heels	8	600
Brogues	10	150
Trainers	14	300

Transaction example

BankAccounts

AccountNumber	AccountName	Balance
B001	Rose	300
B002	James	13450
B003	Shoe Shop	124000
B004	Corner Shop	76000

ShoeShop

Product	Stock	Price
Boots	12	200
High heels	8	600
Brogues	10	150
Trainers	14	300

Transaction example

BankAccounts

AccountNumber	AccountName	Balance
B001	Rose	300
B002	James	13450
B003	Shoe Shop	124000
B004	Corner Shop	76000

ShoeShop

Product	Stock	Price
Boots	12	200
High heels	8	600
Brogues	10	150
Trainers	14	300

```
UPDATE BankAccounts  
SET Balance = Balance-200  
WHERE AccountName = 'Rose';
```

Transaction example

BankAccounts

AccountNumber	AccountName	Balance
B001	Rose	300
B002	James	13450
B003	Shoe Shop	124000
B004	Corner Shop	76000

ShoeShop

Product	Stock	Price
Boots	12	200
High heels	8	600
Brogues	10	150
Trainers	14	300

Transaction example

BankAccounts

AccountNumber	AccountName	Balance
B001	Rose	300
B002	James	13450
B003	Shoe Shop	124000
B004	Corner Shop	76000

ShoeShop

Product	Stock	Price
Boots	12	200
High heels	8	600
Brogues	10	150
Trainers	14	300

```
UPDATE BankAccounts  
SET Balance = Balance+200  
WHERE AccountName = 'Shoe Shop';
```

Transaction example

BankAccounts

AccountNumber	AccountName	Balance
B001	Rose	300
B002	James	13450
B003	Shoe Shop	124000
B004	Corner Shop	76000

ShoeShop

Product	Stock	Price
Boots	12	200
High heels	8	600
Brogues	10	150
Trainers	14	300

Transaction example

BankAccounts

AccountNumber	AccountName	Balance
B001	Rose	300
B002	James	13450
B003	Shoe Shop	124000
B004	Corner Shop	76000

ShoeShop

Product	Stock	Price
Boots	12	200
High heels	8	600
Brogues	10	150
Trainers	14	300

```
UPDATE ShoeShop
SET Stock = Stock-1
WHERE Product = 'Boots';
```

Transaction example

BankAccounts

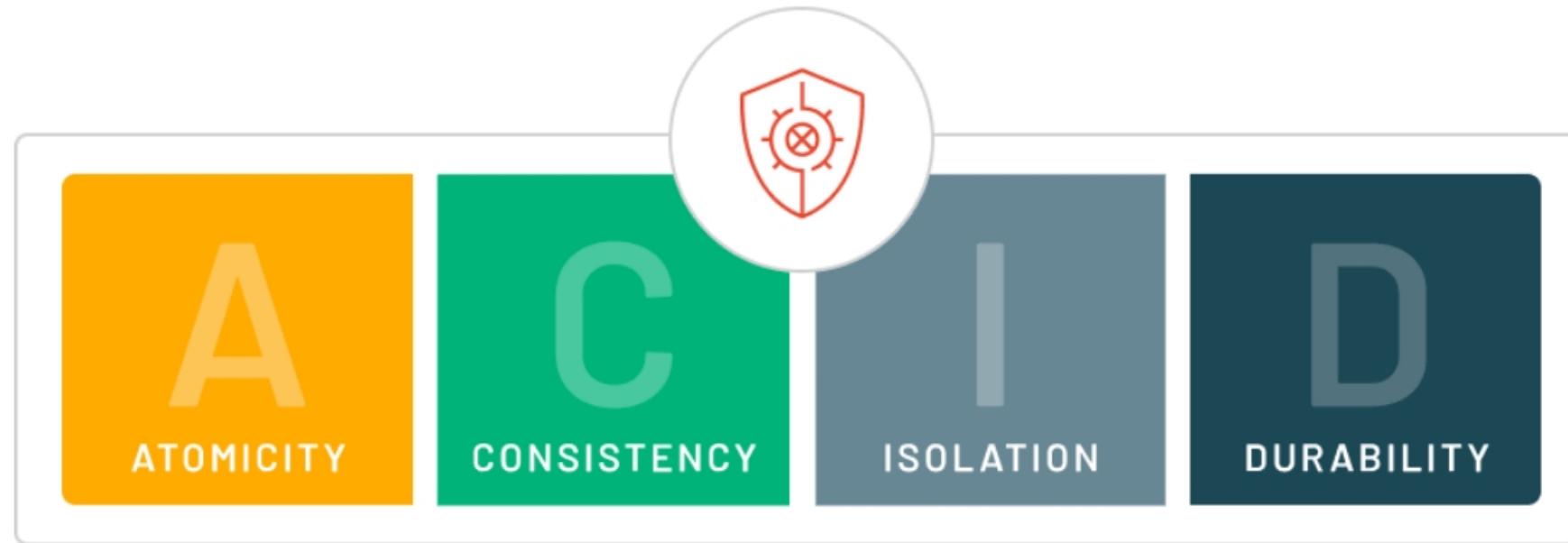
AccountNumber	AccountName	Balance
B001	Rose	300
B002	James	13450
B003	Shoe Shop	124000
B004	Corner Shop	76000

ShoeShop

Product	Stock	Price
Boots	12	200
High heels	8	600
Brogues	10	150
Trainers	14	300

If any of these UPDATE statements fail, the whole transaction must fail

What is an ACID transaction?



What is an ACID transaction?

Atomic

All changes must be performed successfully or not at all.

Consistent

Data must be in a consistent state before and after the transaction.

What is an ACID transaction?

Isolated

No other process can change the data while the transaction is running.

Durable

The changes made by the transaction must persist.

ACID commands

- **BEGIN**

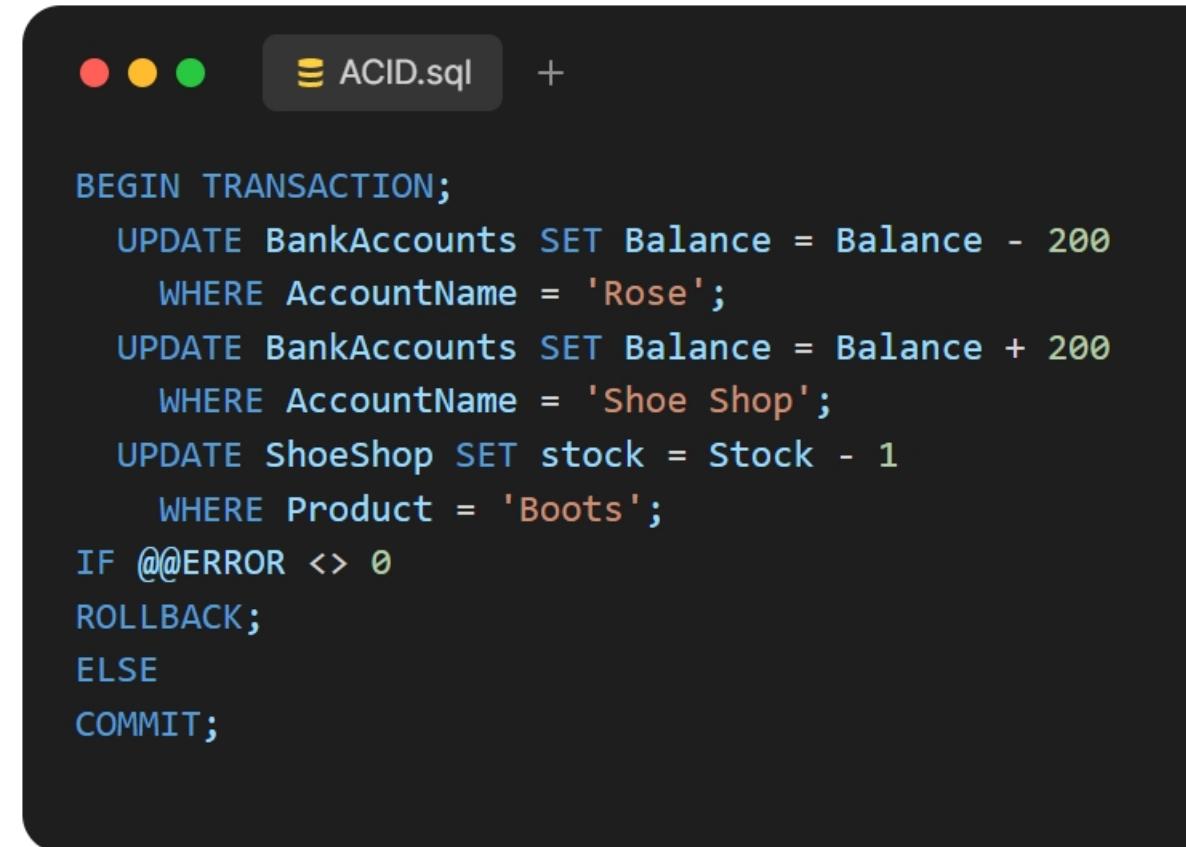
- Start the ACID transaction

- **COMMIT**

- All statements complete successfully
- Save the new database state

- **ROLLBACK**

- One or more statements fail



```
BEGIN TRANSACTION;
UPDATE BankAccounts SET Balance = Balance - 200
WHERE AccountName = 'Rose';
UPDATE BankAccounts SET Balance = Balance + 200
WHERE AccountName = 'Shoe Shop';
UPDATE ShoeShop SET stock = Stock - 1
WHERE Product = 'Boots';
IF @@ERROR <> 0
ROLLBACK;
ELSE
COMMIT;
```

ACID transaction Importance

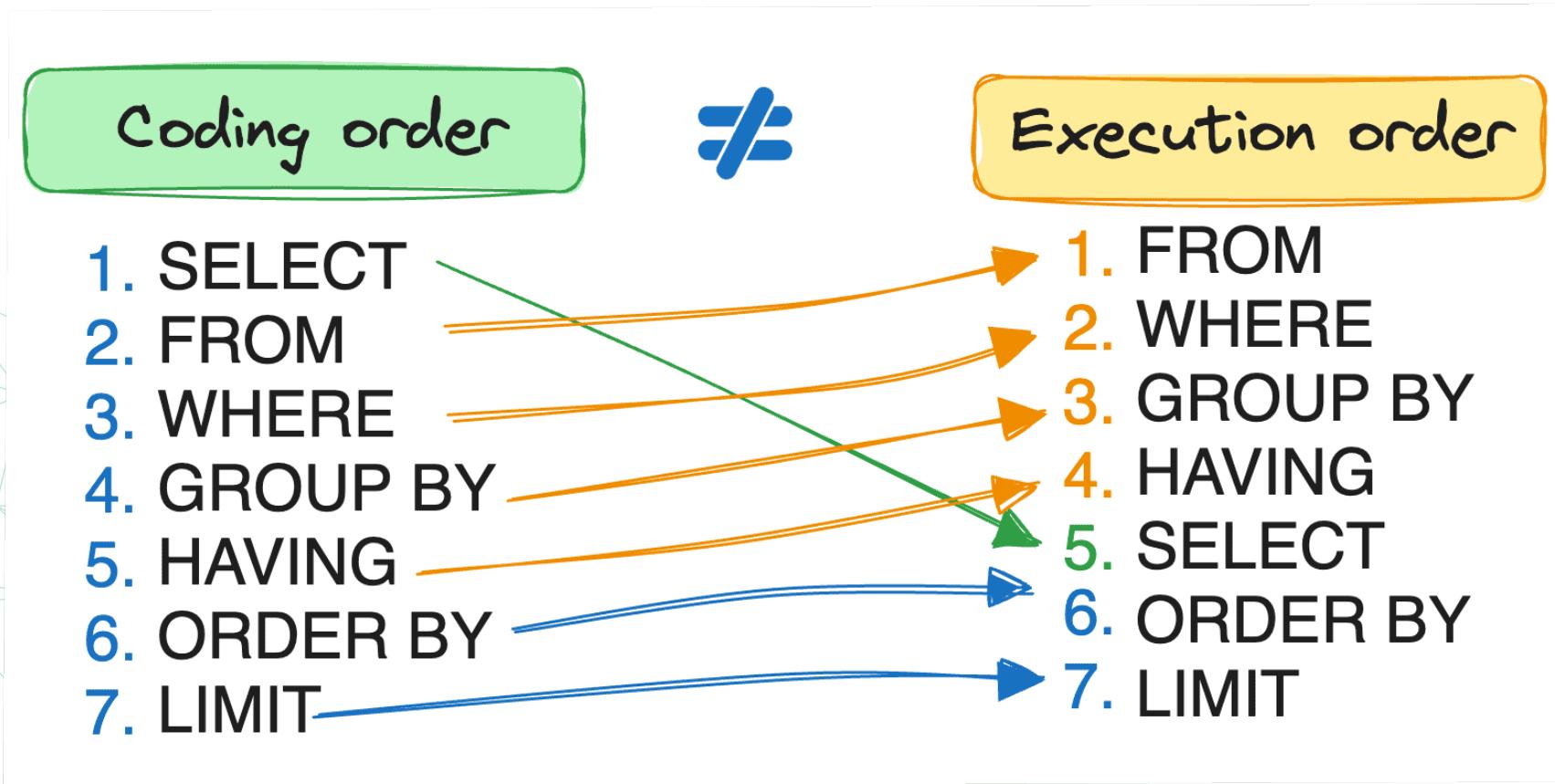
Why are ACID transactions a good thing to have?

Module 7

Advanced SQL for Data Engineer I

JOIN Operations

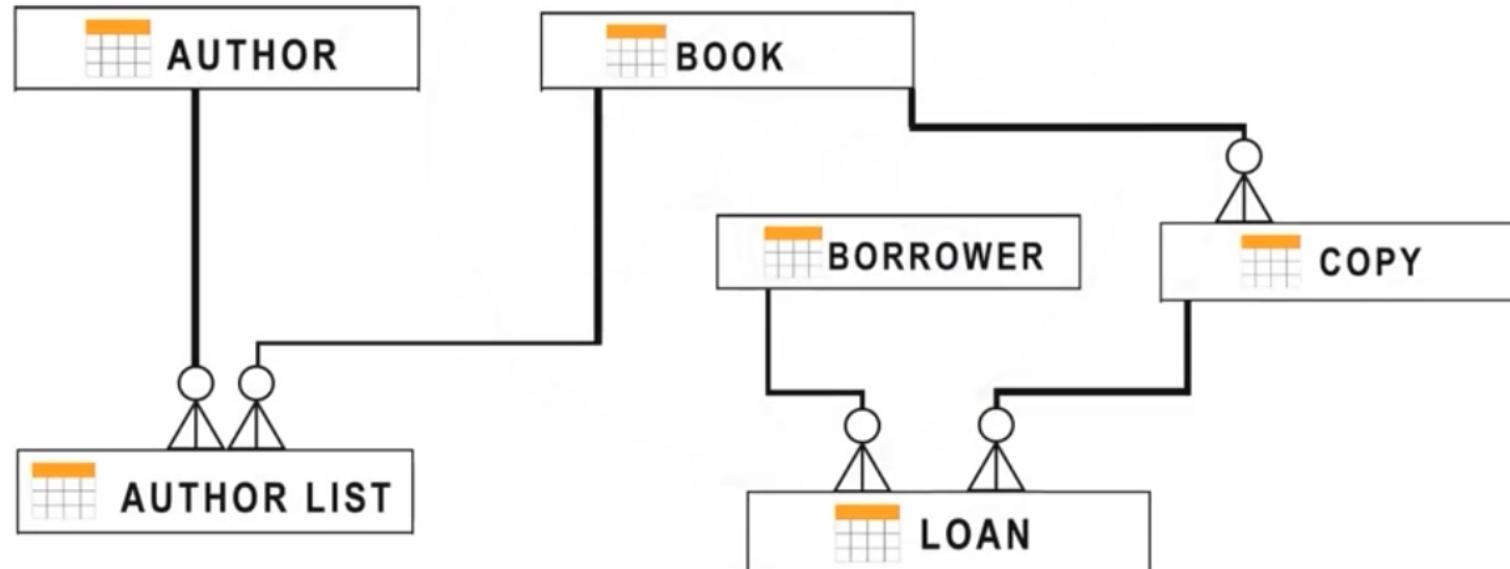
Coding and Execution Orders



Relational model database diagram

JOIN operator:

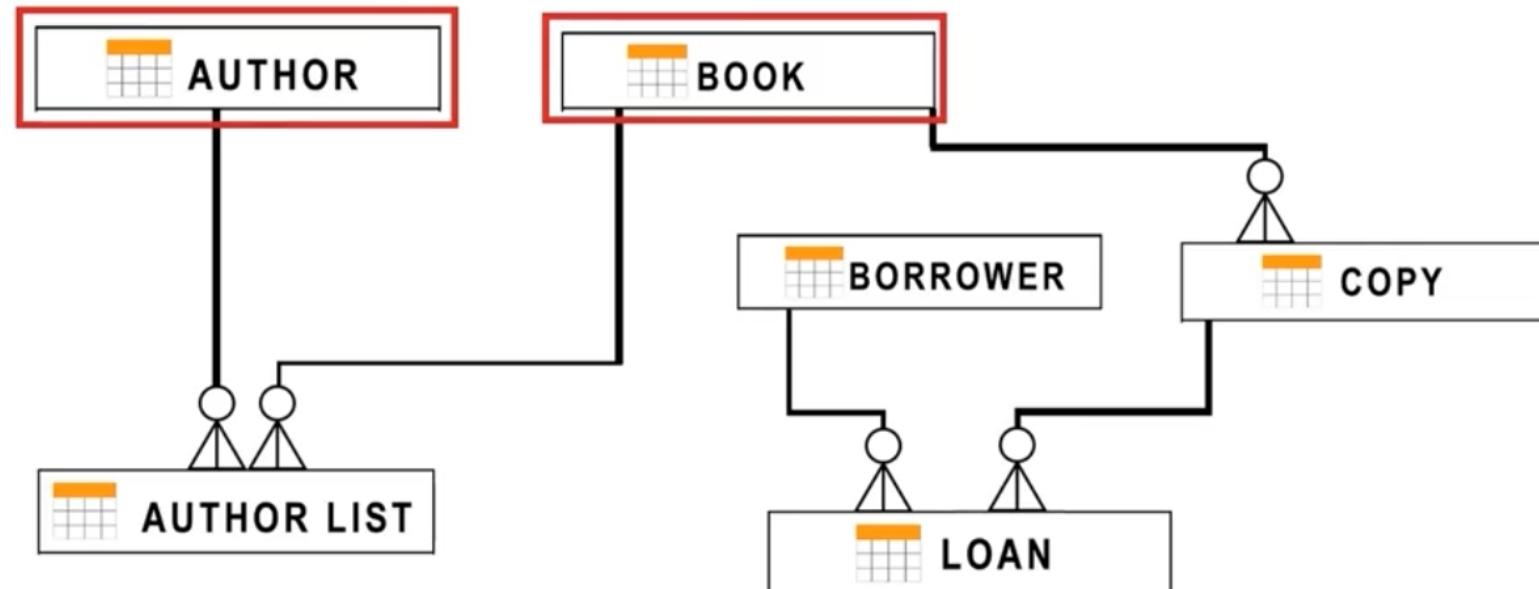
- Combines rows from two or more tables
- Based on a relationship



Relational model database diagram

JOIN operator:

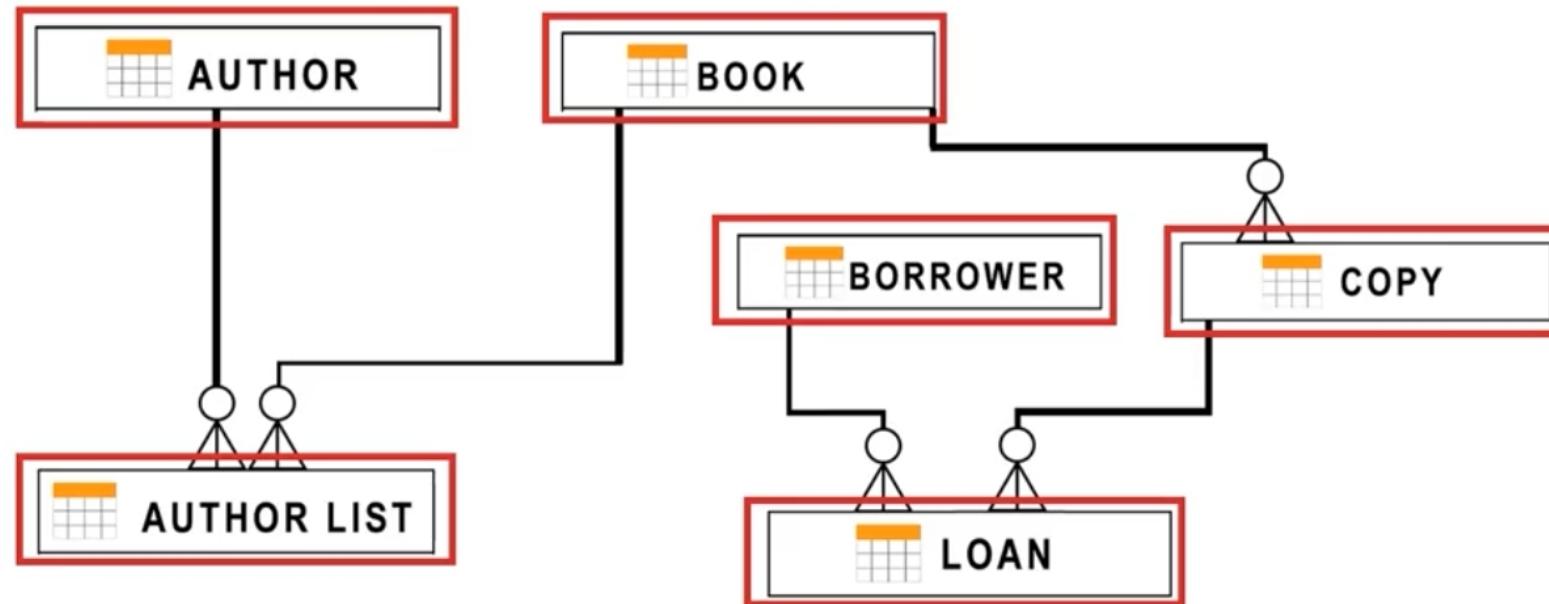
- Combines rows from two or more tables
- Based on a relationship



Relational model database diagram

JOIN operator:

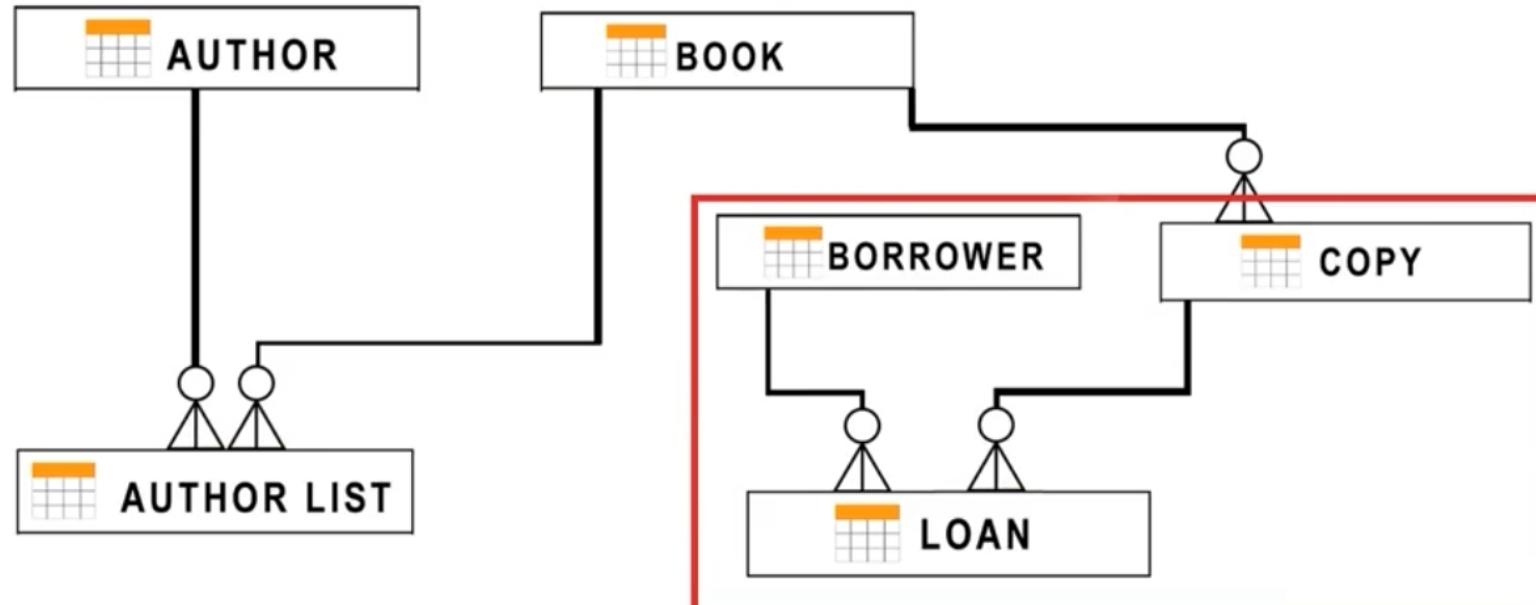
- Combines rows from two or more tables
- Based on a relationship



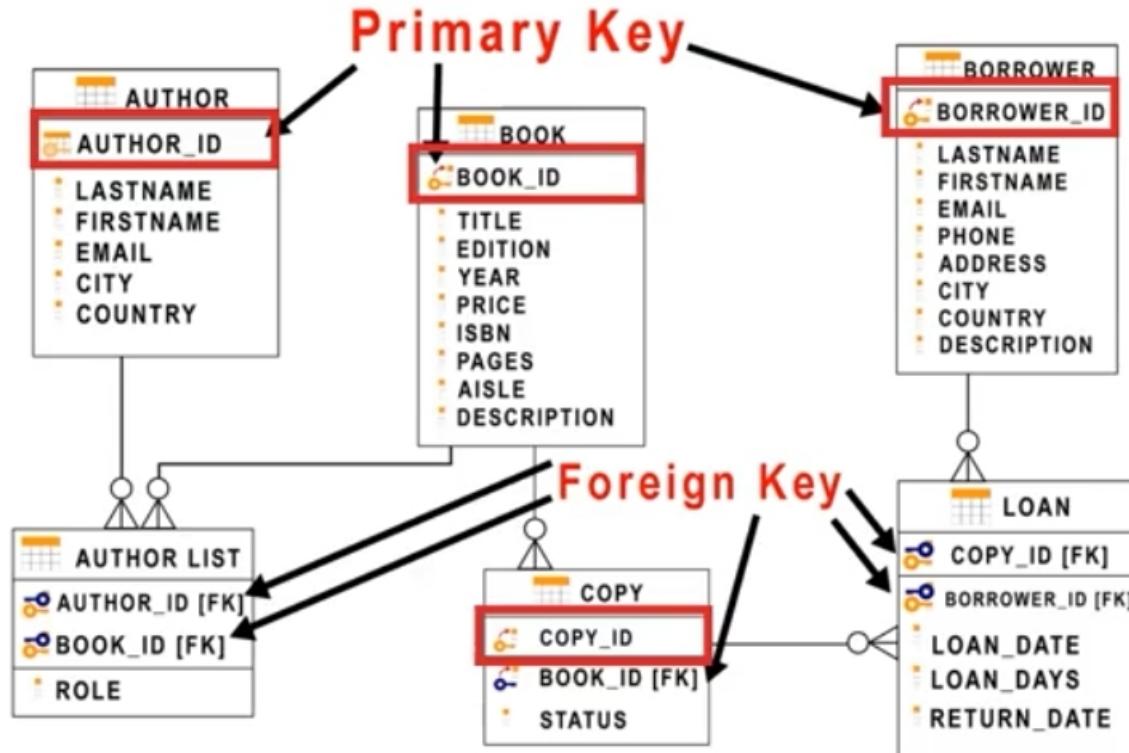
Relational model database diagram

JOIN operator:

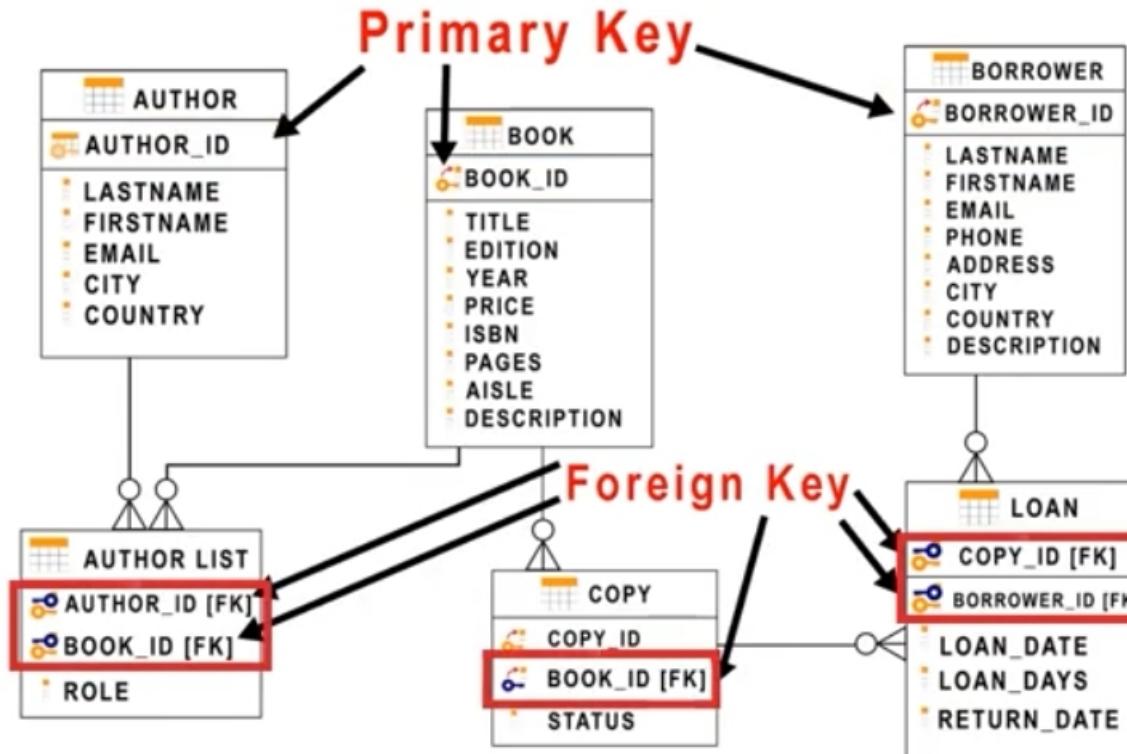
- Combines rows from two or more tables
- Based on a relationship



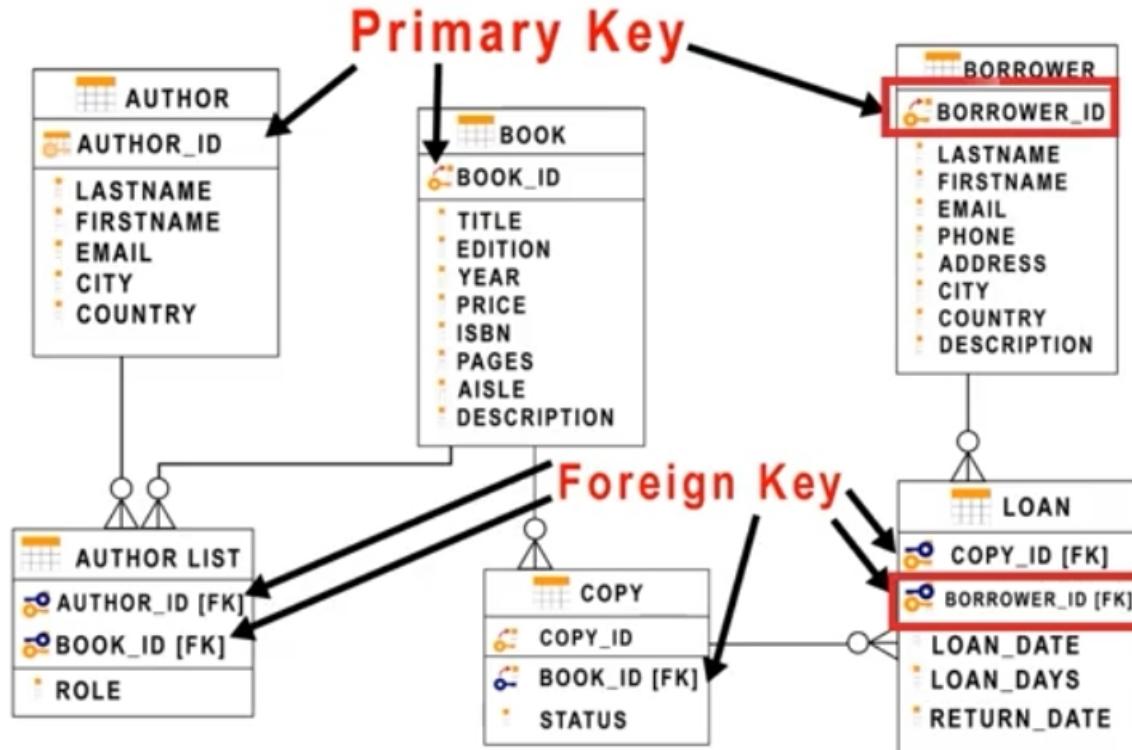
Relational model ER diagram



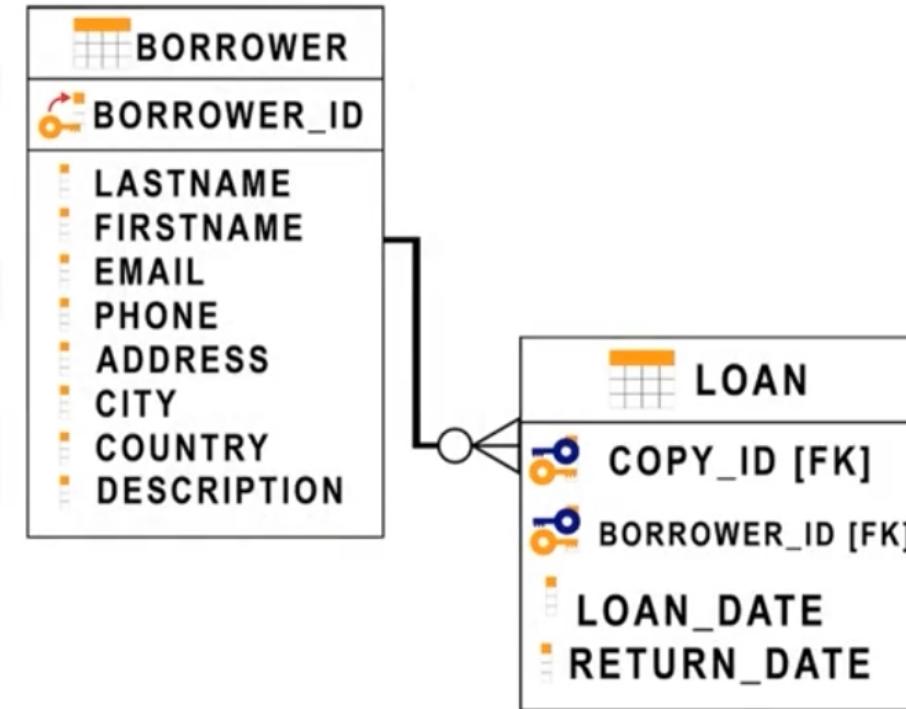
Relational model ER diagram



Relational model ER diagram

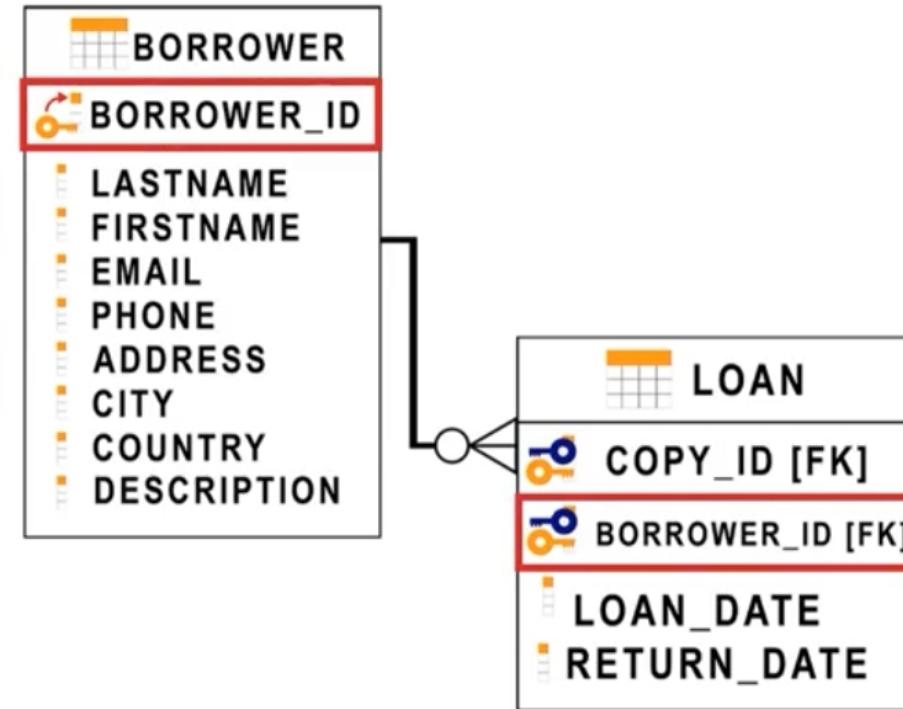


Joining tables



Borrower.Borrower_ID = Loan.Borrower_ID

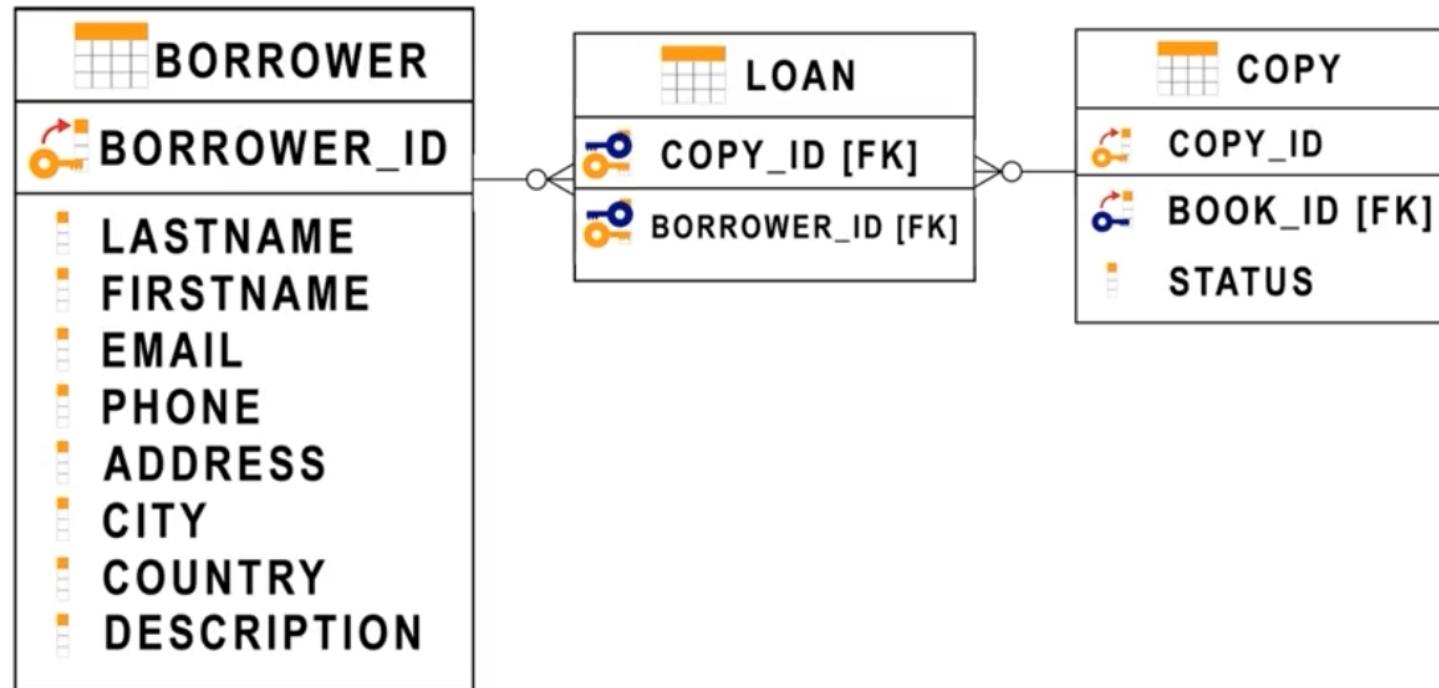
Joining tables



Borrower.Borrower_ID = Loan.Borrower_ID

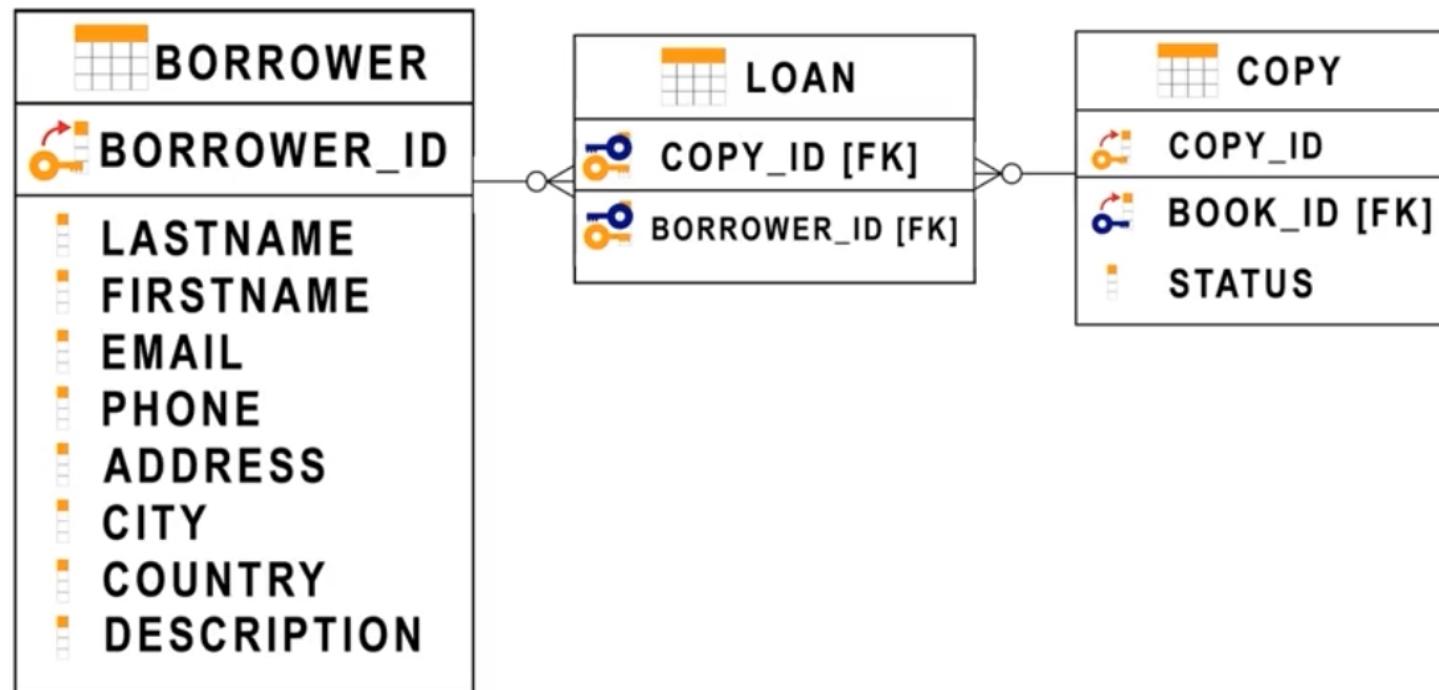
Joining Three Tables

Which copy of a book does the borrower have on loan?



Joining Three Tables

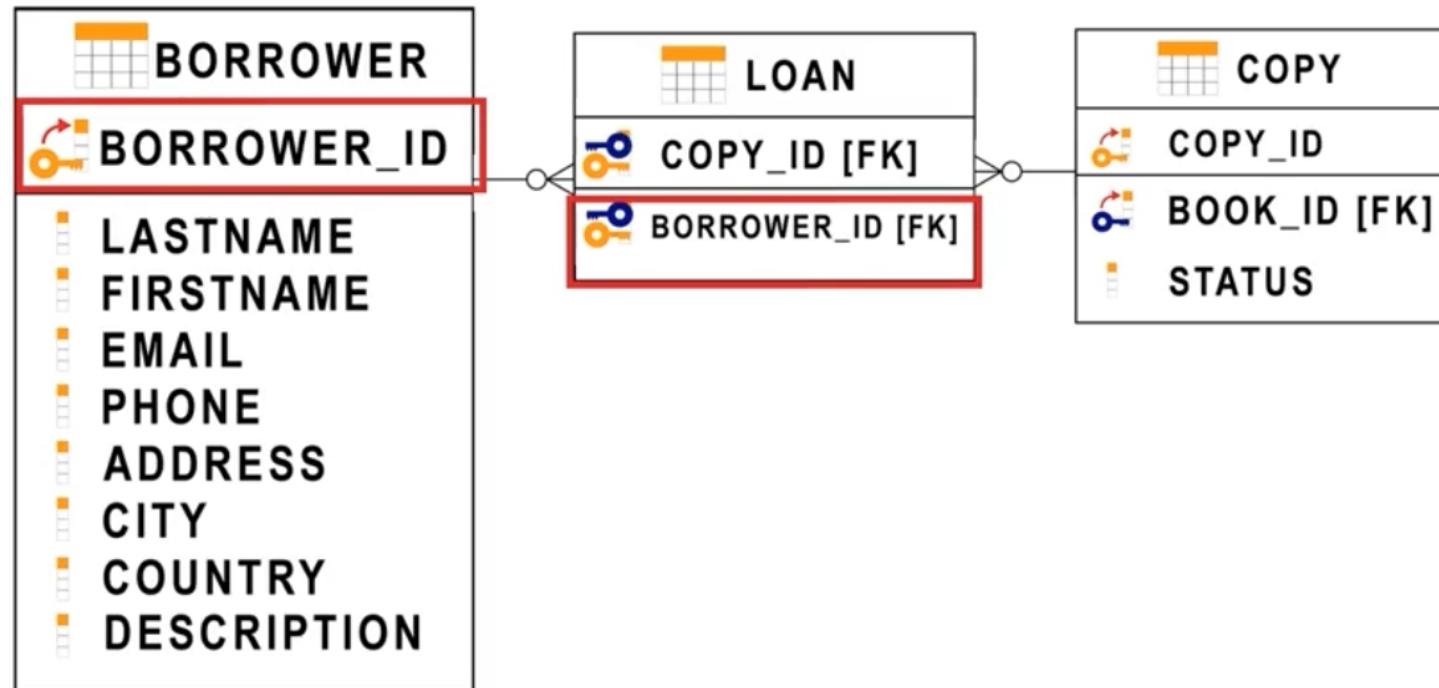
Which copy of a book does the borrower have on loan?



Joining Three Tables

Which copy of a book does the borrower have on loan?

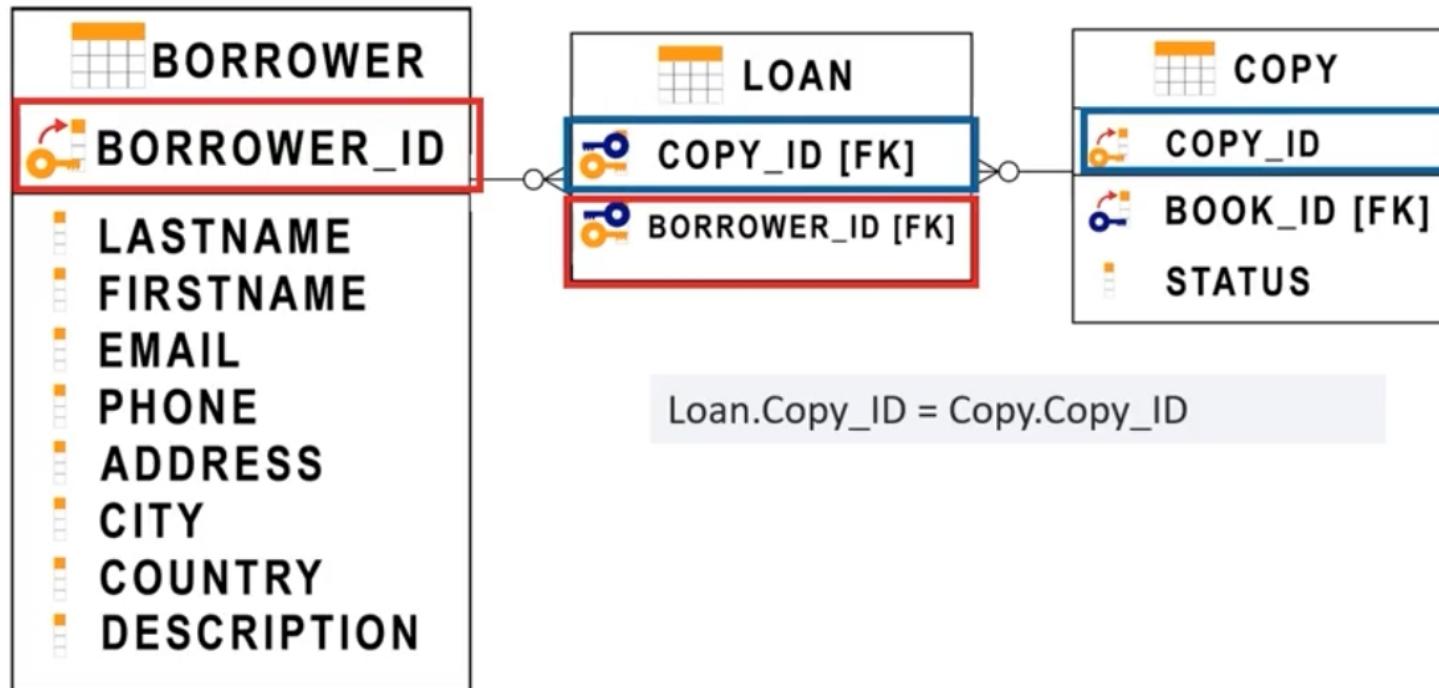
Borrower.Borrower_ID = Loan.Borrower_ID



Joining Three Tables

Which copy of a book does the borrower have on loan?

Borrower.Borrower_ID = Loan.Borrower_ID



Loan.Copy_ID = Copy.Copy_ID

Types of joins

Table A

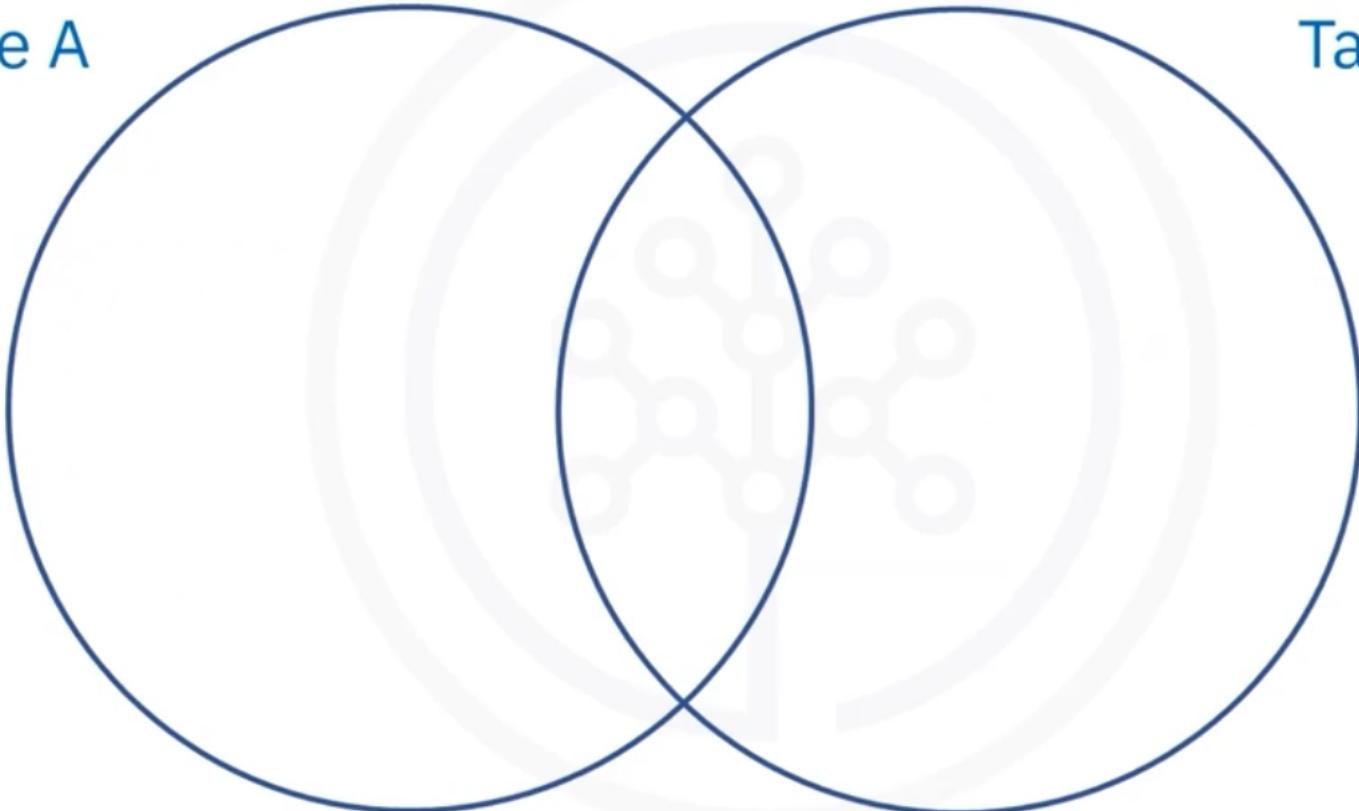


Table B

Types of joins

Table A

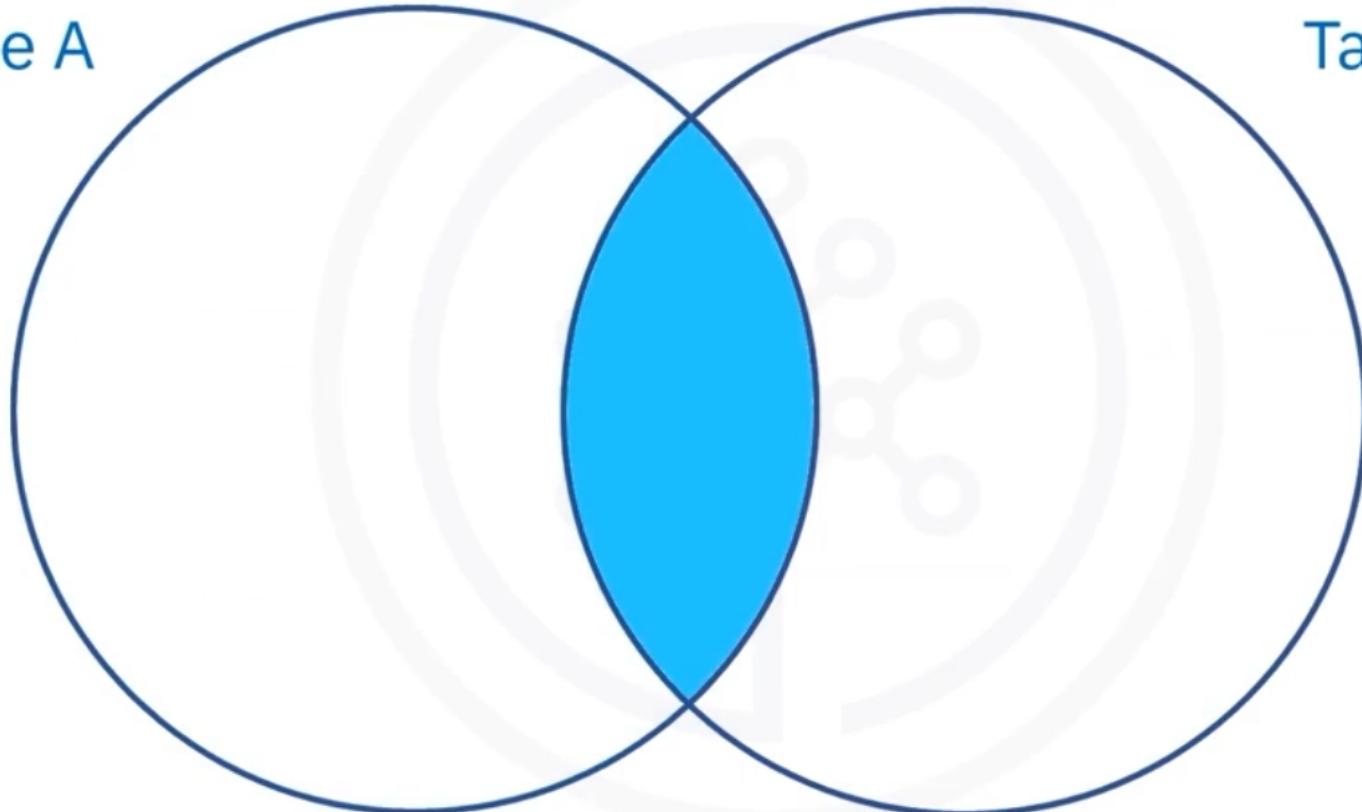


Table B

Types of joins

Table A

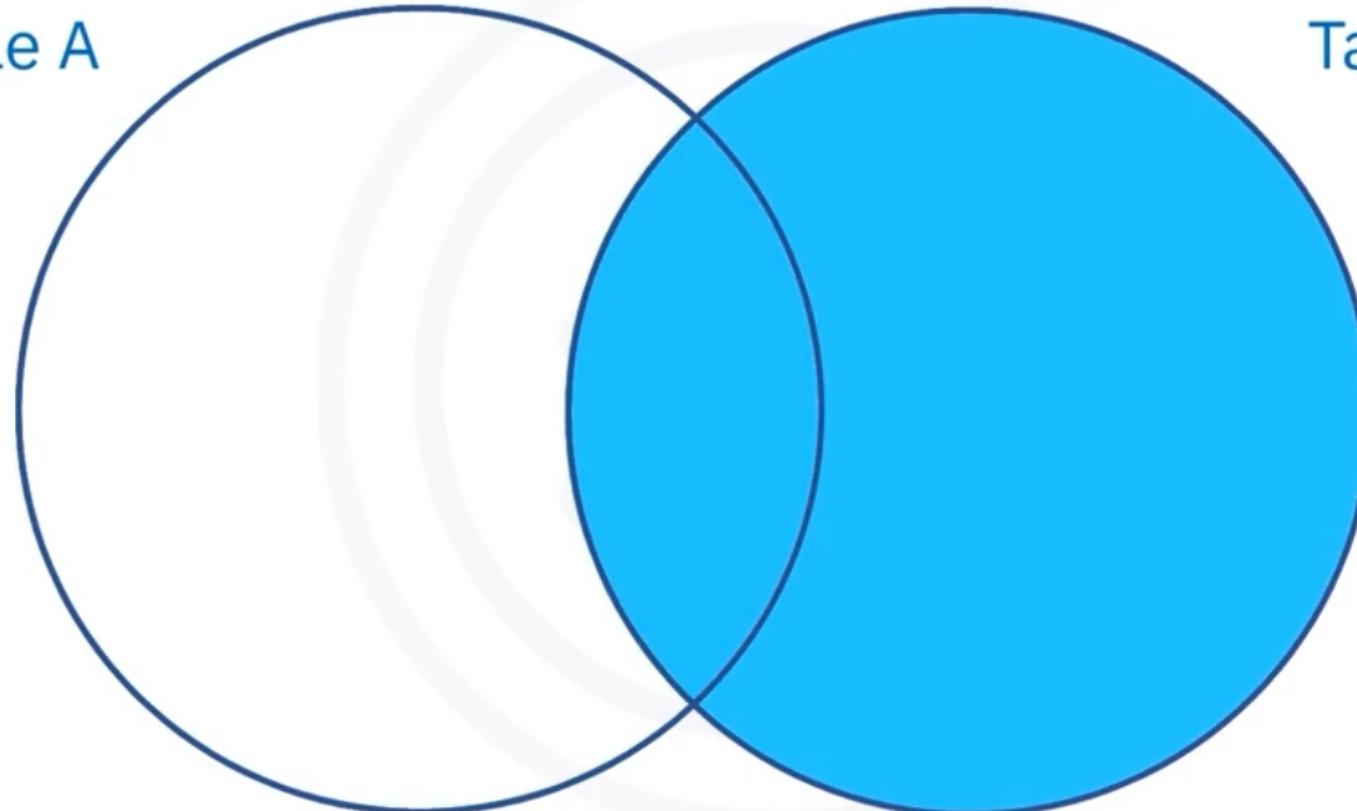


Table B

Types of joins

Table A

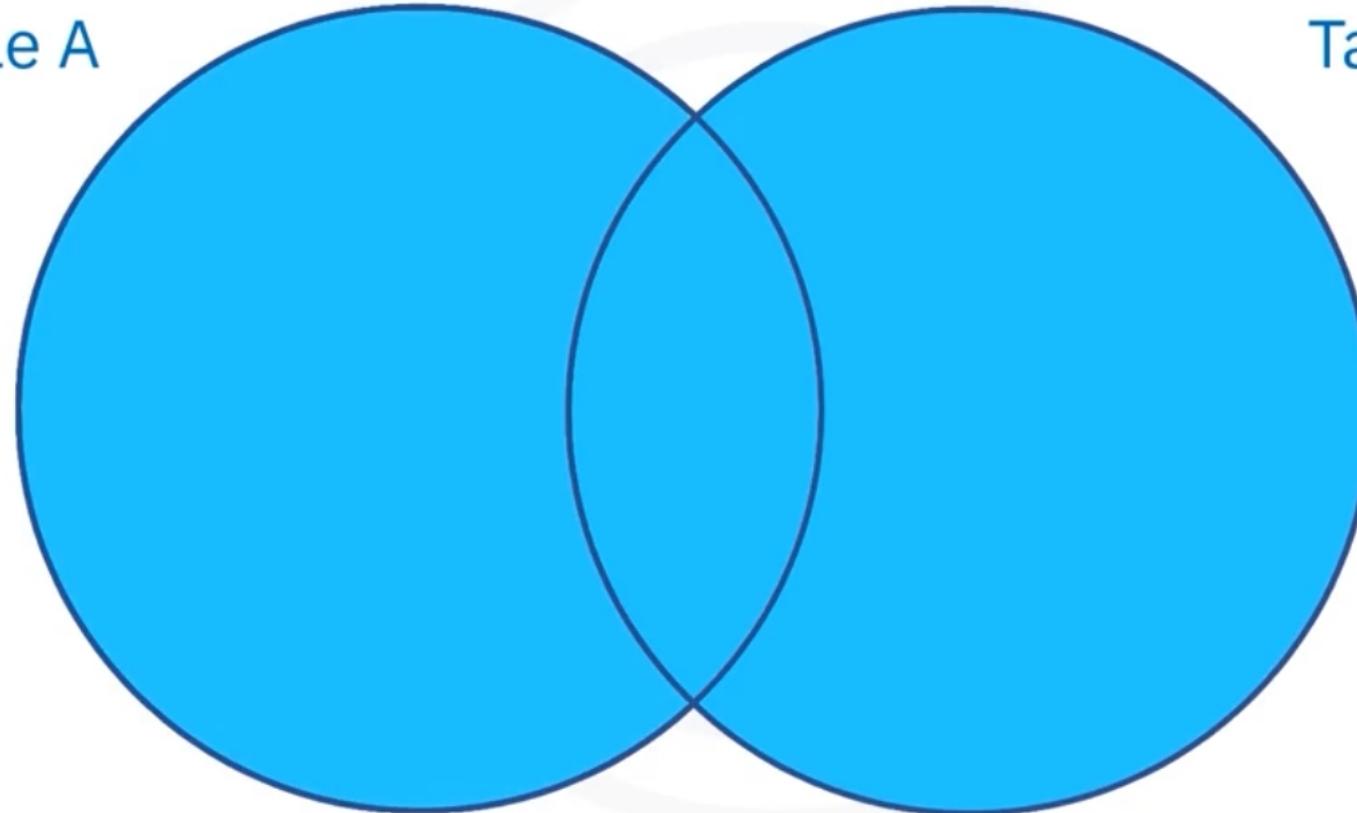


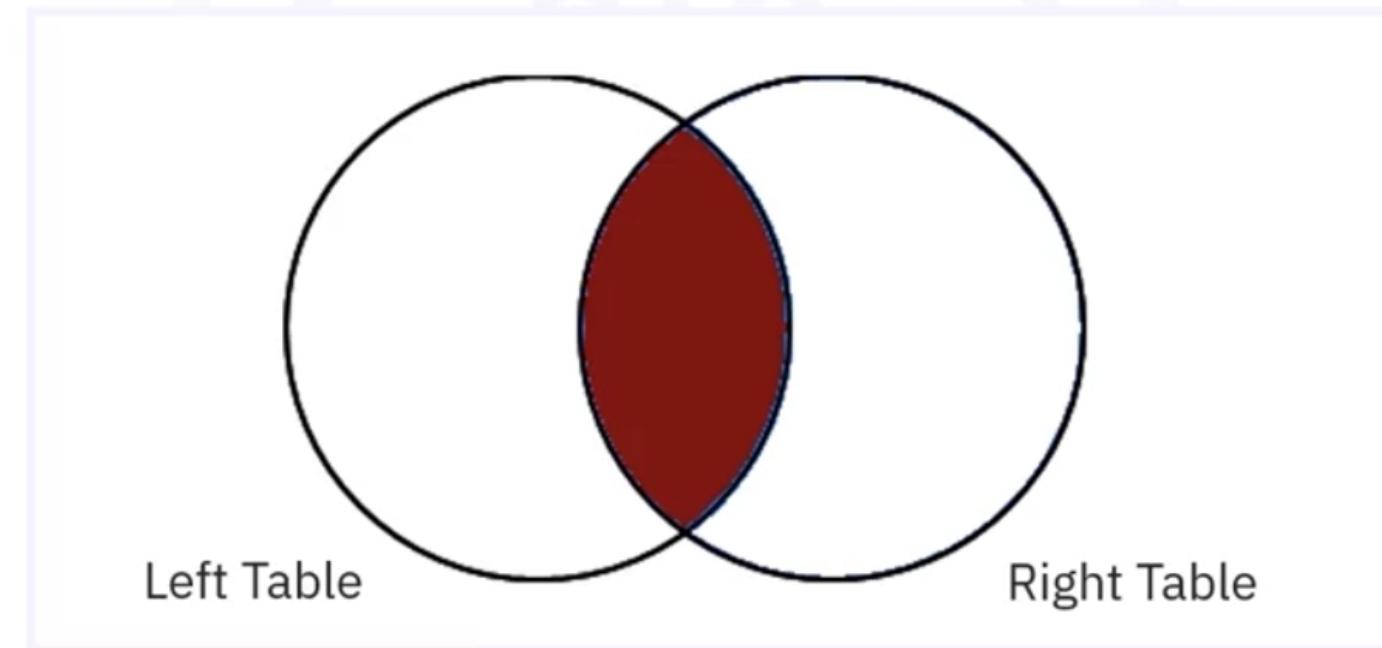
Table B

Types of joins

- Inner Join
- Outer Join
 - Left Outer Join
 - Right Outer Join
 - Full Outer Join

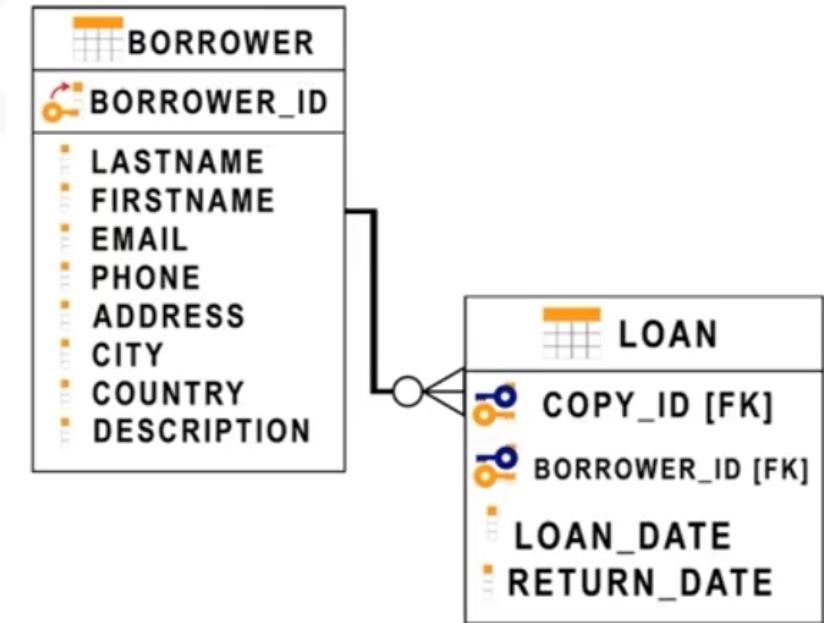
Inner Join

- Inner join displays matches only



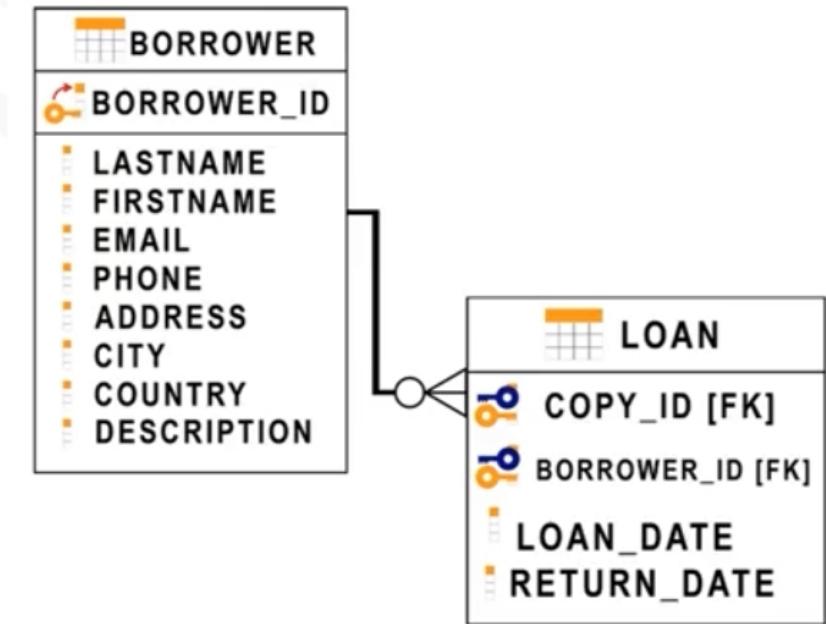
INNER JOIN operator

```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,  
       L.BORROWER_ID, L.LOAN_DATE  
FROM BORROWER B INNER JOIN LOAN L  
ON B.BORROWER_ID = L.BORROWER_ID
```



INNER JOIN operator

```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,  
       L.BORROWER_ID, L.LOAN_DATE  
FROM BORROWER B INNER JOIN LOAN L  
      ON B.BORROWER_ID = L.BORROWER_ID
```

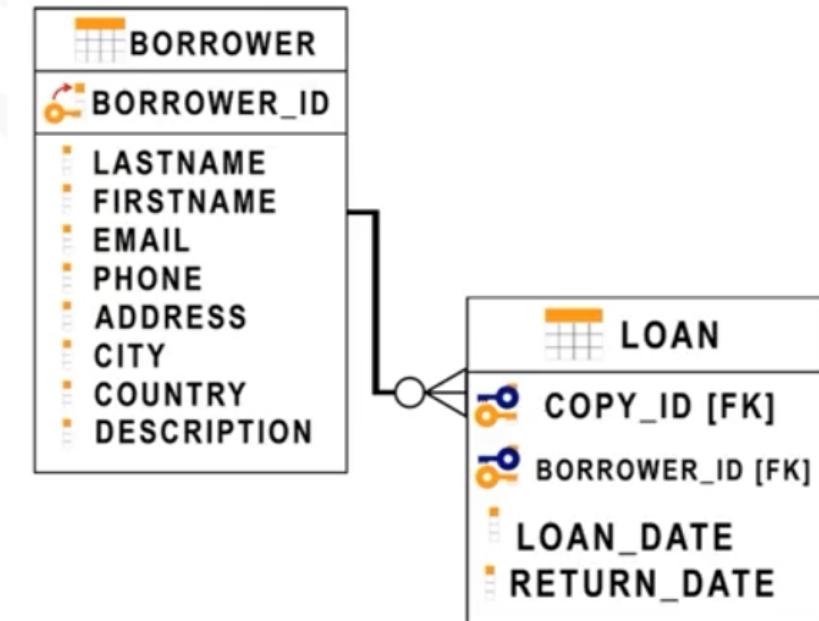




INNER JOIN operator

```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,  
       L.BORROWER_ID, L.LOAN_DATE  
FROM BORROWER B INNER JOIN LOAN L  
      ON B.BORROWER_ID = L.BORROWER_ID
```

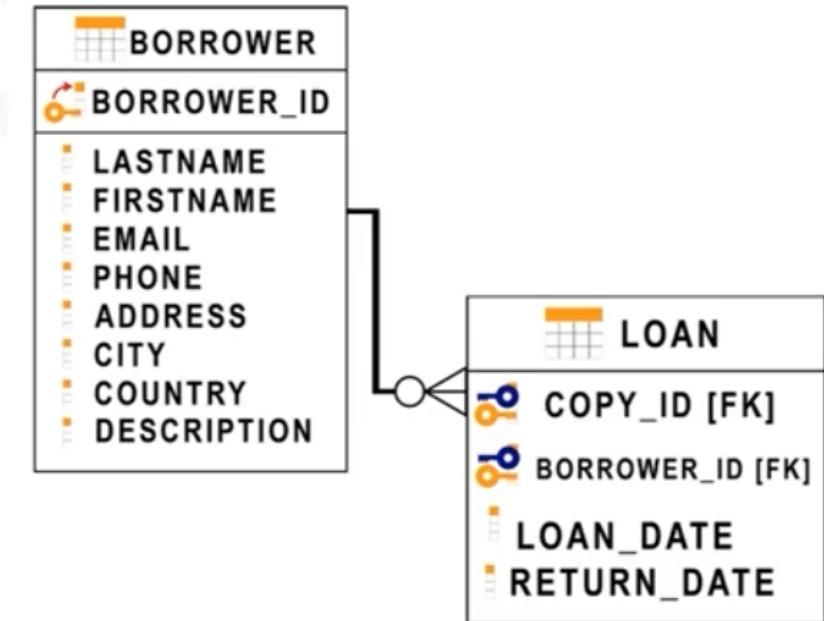
- In this example, the Borrower table is the Left table



INNER JOIN operator

```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,  
       L.BORROWER_ID, L.LOAN_DATE  
FROM BORROWER B INNER JOIN LOAN L  
ON B.BORROWER_ID = L.BORROWER_ID
```

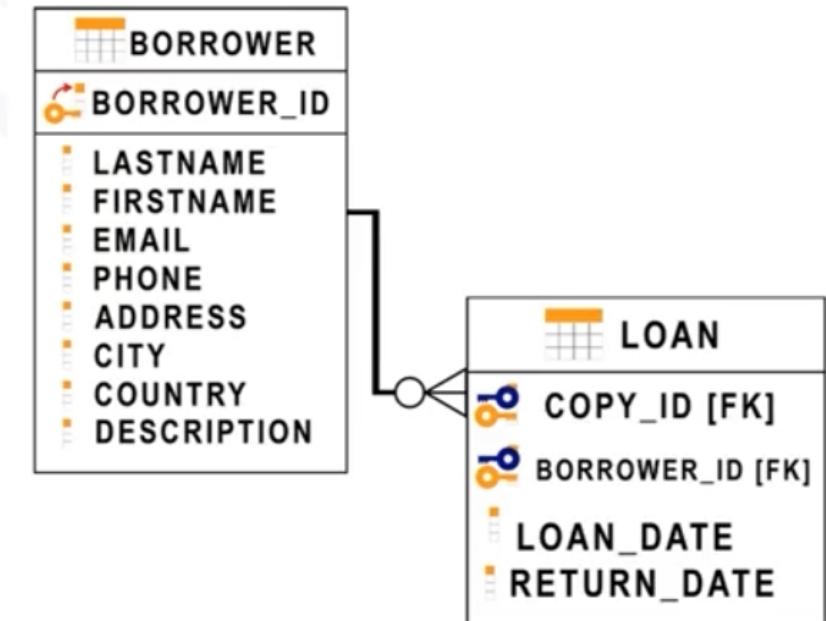
- In this example, the Borrower table is the Left table



INNER JOIN operator

```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,  
       L.BORROWER_ID, L.LOAN_DATE  
FROM BORROWER B INNER JOIN LOAN L  
ON B.BORROWER_ID = L.BORROWER_ID
```

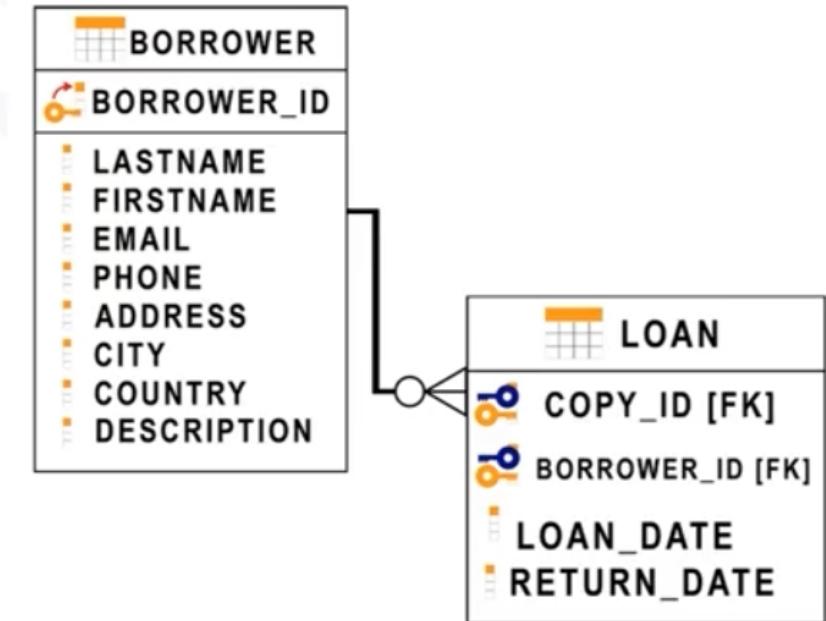
- In this example, the Borrower table is the Left table



INNER JOIN operator

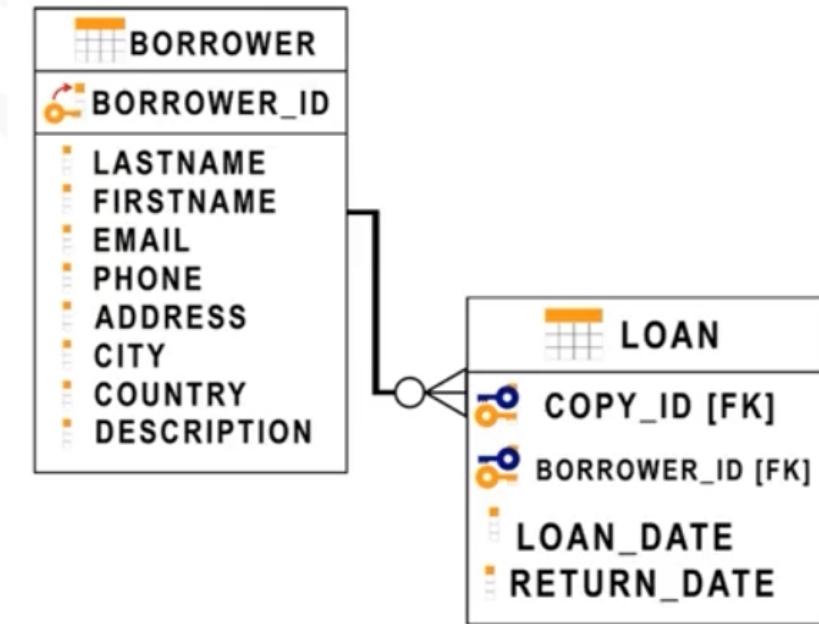
```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,  
       L.BORROWER_ID, L.LOAN_DATE  
FROM BORROWER B INNER JOIN LOAN L  
ON B.BORROWER_ID = L.BORROWER_ID
```

- In this example, the Borrower table is the Left table



INNER JOIN operator

```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,  
       L.BORROWER_ID, L.LOAN_DATE  
FROM BORROWER B INNER JOIN LOAN L  
ON B.BORROWER_ID = L.BORROWER_ID
```



- In this example, the Borrower table is the Left table
- Each column name is prefixed with an alias to indicate which table each column is associated with

INNER JOIN operator

```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,  
       L.BORROWER_ID, L.LOAN_DATE  
FROM BORROWER B INNER JOIN LOAN L  
ON B.BORROWER_ID = L.BORROWER_ID
```

BORROWER_ID	LASTNAME	COUNTRY
D1	SMITH	CA
D2	SANDLER	CA
D3	SOMMERS	CA
D4	ARDEN	CA
D5	XIE	CA
D6	PETERS	CA
D7	LI	CA
D8	WONG	CA
D10	KIEVA	CA

BORROWER_ID	LOAN_DATE
D1	11/24/2010
D2	11/24/2010
D3	11/24/2010
D4	11/24/2010
D5	11/24/2010
D9	11/24/2010

INNER JOIN operator

```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,  
       L.BORROWER_ID, L.LOAN_DATE  
FROM BORROWER B INNER JOIN LOAN L  
ON B.BORROWER_ID = L.BORROWER_ID
```

BORROWER_ID	LASTNAME	COUNTRY
D1	SMITH	CA
D2	SANDLER	CA
D3	SOMMERS	CA
D4	ARDEN	CA
D5	XIE	CA
D6	PETERS	CA
D7	LI	CA
D8	WONG	CA
D10	KIEVA	CA

BORROWER_ID	LOAN_DATE
D1	11/24/2010
D2	11/24/2010
D3	11/24/2010
D4	11/24/2010
D5	11/24/2010
D9	11/24/2010

INNER JOIN operator

```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,  
       L.BORROWER_ID, L.LOAN_DATE  
FROM BORROWER B INNER JOIN LOAN L  
ON B.BORROWER_ID = L.BORROWER_ID
```

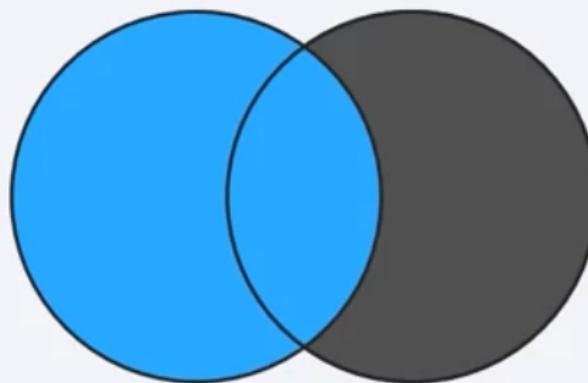
BORROWER_ID	LASTNAME	COUNTRY
D1	SMITH	CA
D2	SANDLER	CA
D3	SOMMERS	CA
D4	ARDEN	CA
D5	XIE	CA
D6	PETERS	CA
D7	LI	CA
D8	WONG	CA
D9	KIEVA	CA

BORROWER_ID	LOAN_DATE
D1	11/24/2010
D2	11/24/2010
D3	11/24/2010
D4	11/24/2010
D5	11/24/2010
D9	11/24/2010

Outer Joins

Left Outer Join

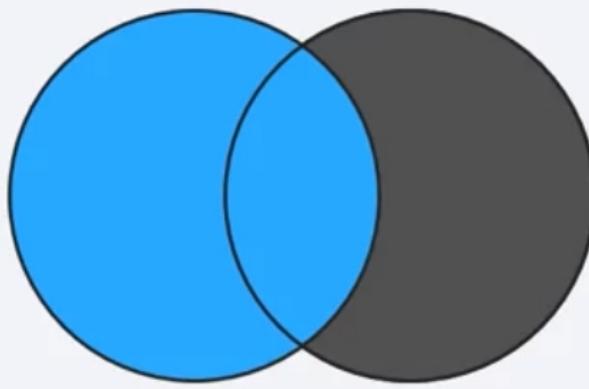
All rows from the left table
to any matching rows from
the right table



Outer Joins

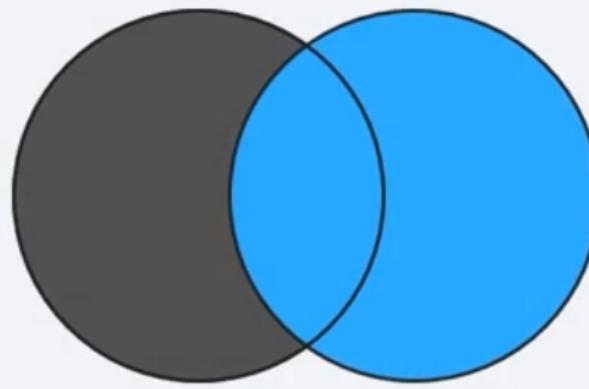
Left Outer Join

All rows from the left table to any matching rows from the right table



Right Outer Join

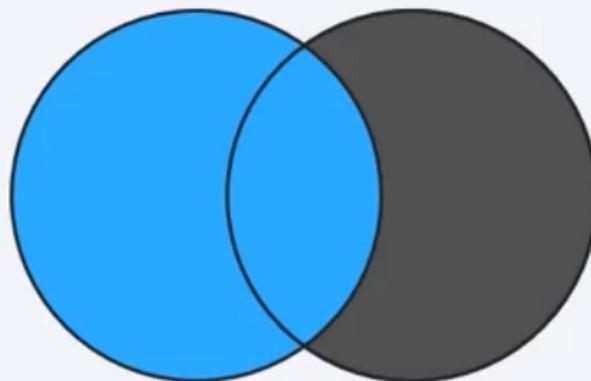
All rows from the right table to any matching rows from the left table



Outer Joins

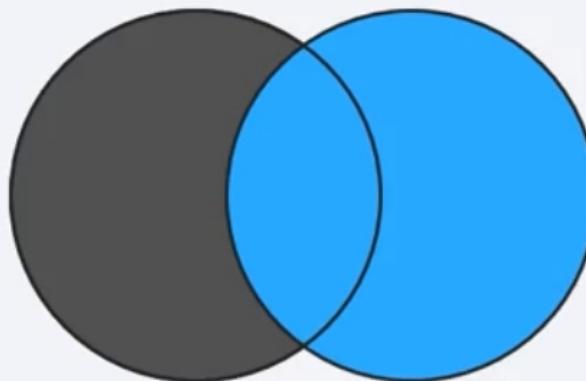
Left Outer Join

All rows from the left table to any matching rows from the right table



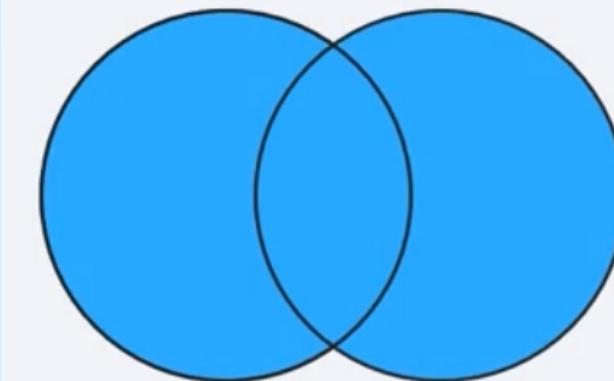
Right Outer Join

All rows from the right table to any matching rows from the left table

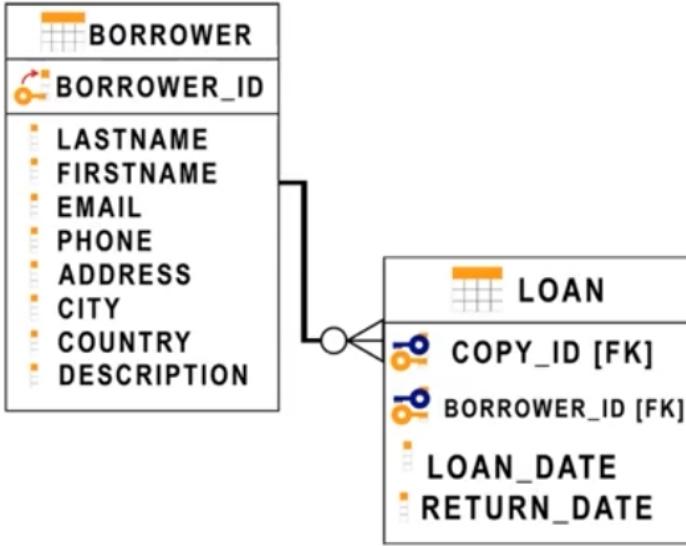


Full Outer Join

All rows from both tables



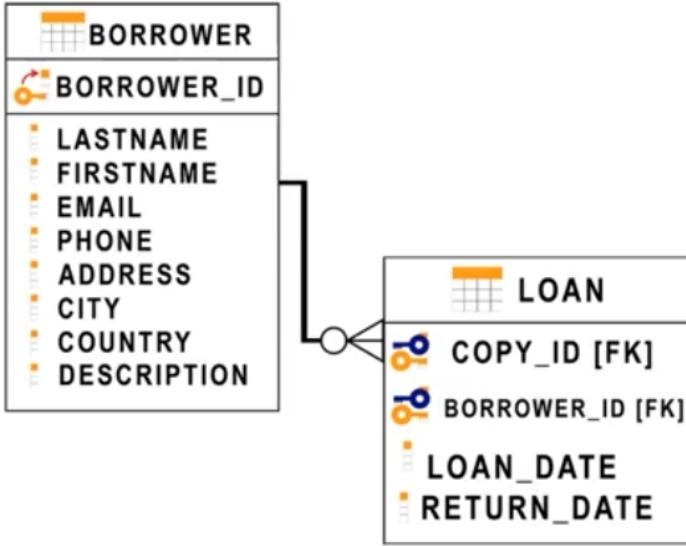
LEFT JOIN operator



```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,  
      L.BORROWER_ID, L.LOAN_DATE  
FROM BORROWER B LEFT JOIN LOAN L  
ON B.BORROWER_ID = L.BORROWER_ID
```

In this example, the Borrower table is the Left table

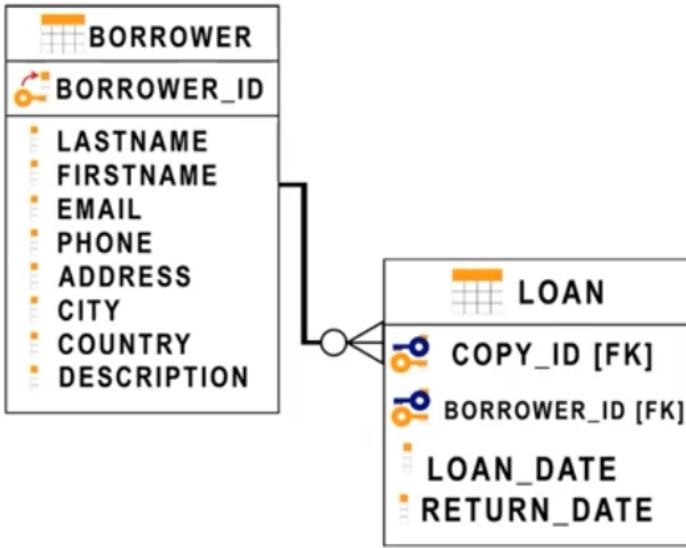
LEFT JOIN operator



```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,  
       L.BORROWER_ID, L.LOAN_DATE  
FROM BORROWER B LEFT JOIN LOAN L  
ON B.BORROWER_ID = L.BORROWER_ID
```

In this example, the Borrower table is the Left table

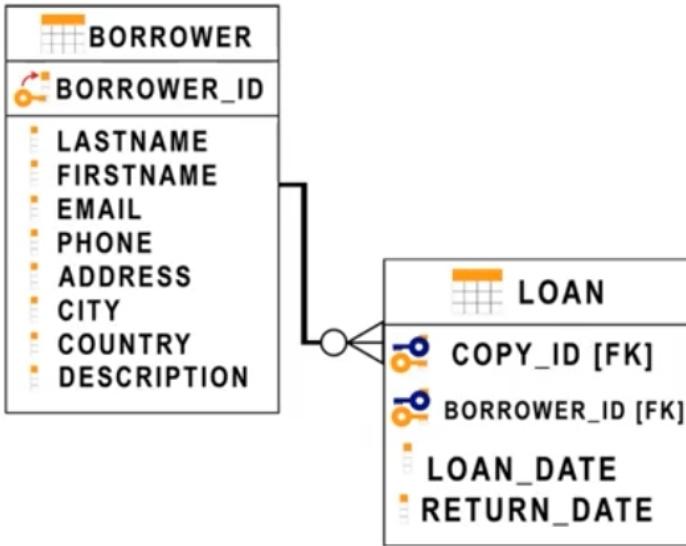
LEFT JOIN operator



```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,  
L.BORROWER_ID, L.LOAN_DATE  
FROM BORROWER B LEFT JOIN LOAN L  
ON B.BORROWER_ID = L.BORROWER_ID
```

In this example, the Borrower table is the Left table

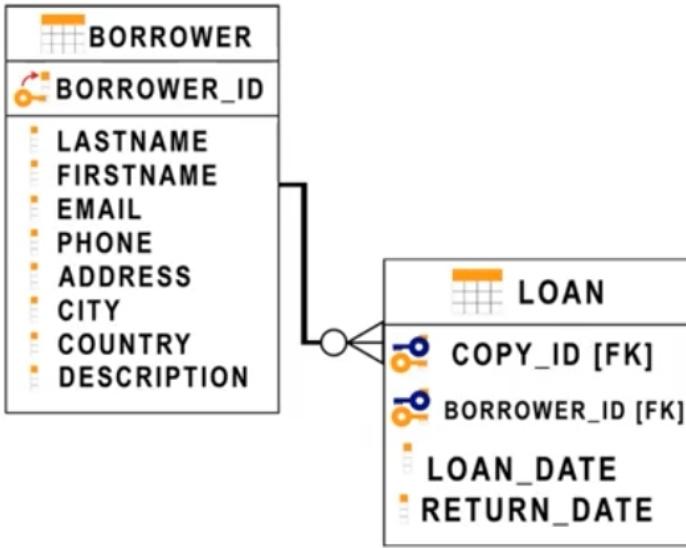
LEFT JOIN operator



```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,  
       L.BORROWER_ID, L.LOAN_DATE  
FROM BORROWER B LEFT JOIN LOAN L  
ON B.BORROWER_ID = L.BORROWER_ID
```

In this example, the Borrower table is the Left table

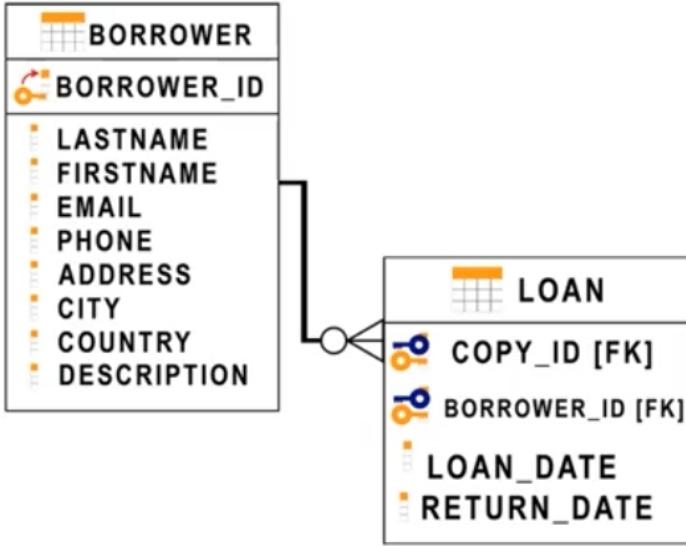
LEFT JOIN operator



```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,  
       L.BORROWER_ID, L.LOAN_DATE  
FROM BORROWER B LEFT JOIN LOAN L  
ON B.BORROWER_ID = L.BORROWER_ID
```

In this example, the Borrower table is the Left table

LEFT JOIN operator

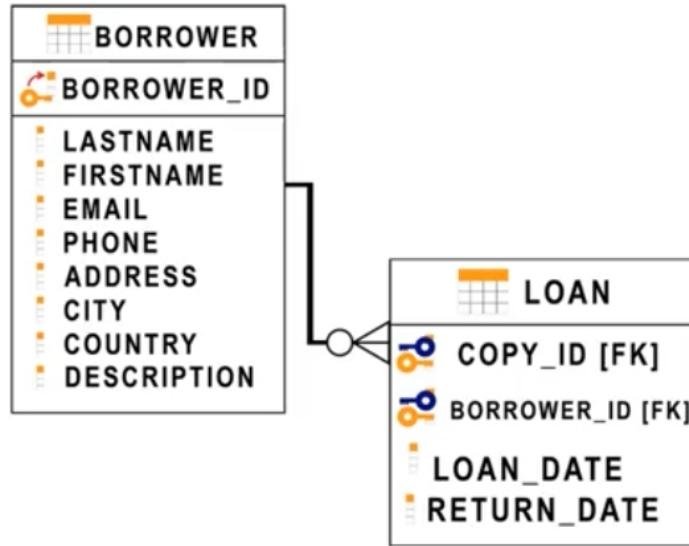


```

SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,
       L.BORROWER_ID, L.LOAN_DATE
FROM BORROWER B LEFT JOIN LOAN L
ON B.BORROWER_ID = L.BORROWER_ID
    
```

BORROWER_ID	LASTNAME	COUNTRY	BORROWER_ID	LOAN_DATE
D1	SMITH	CA	D1	11/24/2010
D2	SANDLER	CA	D2	11/24/2010
D3	SOMMERS	CA	D3	11/24/2010
D4	ARDEN	CA	D4	11/24/2010
D5	XIE	CA	D5	11/24/2010
D6	PETERS	CA	NULL	NULL
D7	LI	CA	NULL	NULL
D8	WONG	CA	NULL	NULL

LEFT JOIN operator



```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,
       L.BORROWER_ID, L.LOAN_DATE
FROM BORROWER B LEFT JOIN LOAN L
ON B.BORROWER_ID = L.BORROWER_ID
```

BORROWER_ID	LASTNAME	COUNTRY	BORROWER_ID	LOAN_DATE
D1	SMITH	CA	D1	11/24/2010
D2	SANDLER	CA	D2	11/24/2010
D3	SOMMERS	CA	D3	11/24/2010
D4	ARDEN	CA	D4	11/24/2010
D5	XIE	CA	D5	11/24/2010
D6	PETERS	CA	NULL	NULL
D7	LI	CA	NULL	NULL
D8	WONG	CA	NULL	NULL

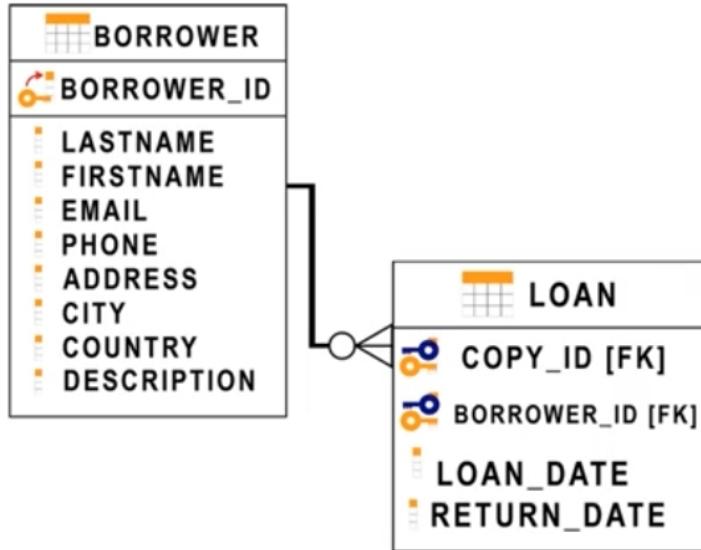
LEFT JOIN operator

- Example:

- Retrieving all employees with their department information
- Our query should return all employees, even if they are not assigned to a department.

```
LEFT_Join.sql +  
  
SELECT e.EmployeeID, e.EmployeeName, d.DepartmentName  
FROM Employees e  
LEFT OUTER JOIN Departments d  
ON e.DepartmentID = d.DepartmentID;  
  
/* `DepartmentName` will be NULL  
for employees without a department. */
```

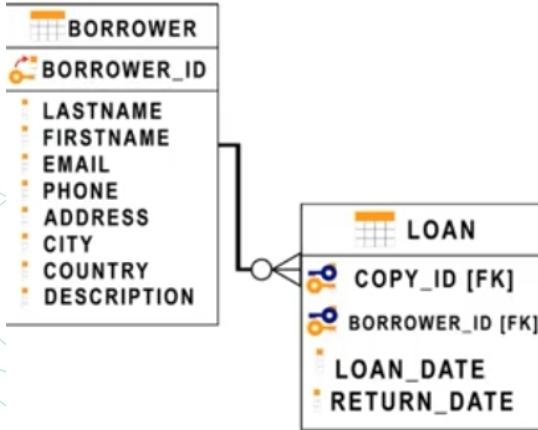
RIGHT JOIN operator



```
SELECT B. BORROWER_ID, B.LASTNAME, B.COUNTRY,  
L. BORROWER_ID, L.LOAN_DATE  
FROM BORROWER B RIGHT JOIN LOAN L  
ON B.BORROWER_ID = L.BORROWER_ID
```

In this example, the Loan table is the Right table

RIGHT JOIN operator

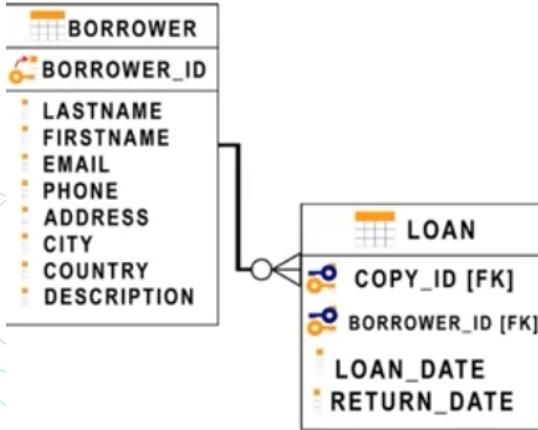


```

SELECT B. BORROWER_ID, B.LASTNAME, B.COUNTRY,
       L. BORROWER_ID, L.LOAN_DATE
FROM BORROWER B RIGHT JOIN LOAN L
ON B.BORROWER_ID = L.BORROWER_ID
  
```

BORROWER_ID	LASTNAME	COUNTRY	BORROWER_ID	LOAN_DATE
D1	SMITH	CA	D1	11/24/2010
D2	SANDLER	CA	D2	11/24/2010
D3	SOMMERS	CA	D3	11/24/2010
D4	ARDEN	CA	D4	11/24/2010
D5	XIE	CA	D5	11/24/2010
NULL	NULL	NULL	D9	11/24/2010

RIGHT JOIN operator



```

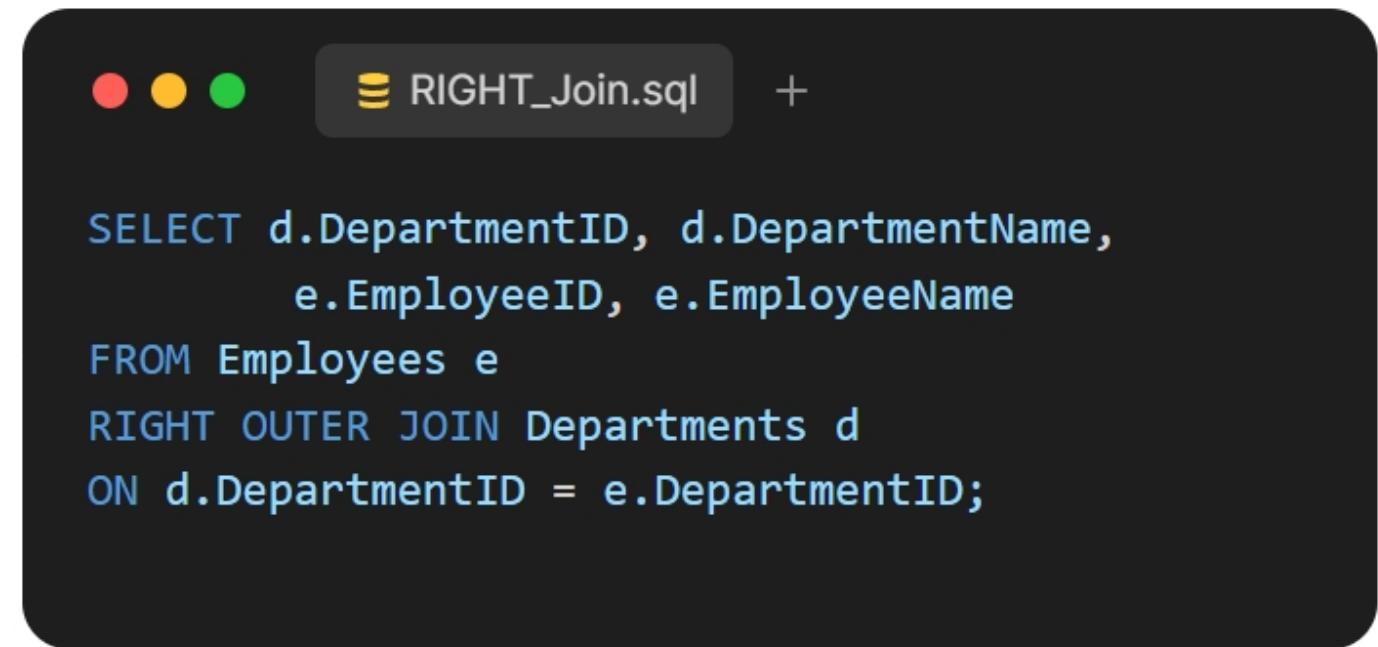
SELECT B. BORROWER_ID, B.LASTNAME, B.COUNTRY,
       L. BORROWER_ID, L.LOAN_DATE
FROM BORROWER B RIGHT JOIN LOAN L
ON B.BORROWER_ID = L.BORROWER_ID
    
```

BORROWER_ID	LASTNAME	COUNTRY	BORROWER_ID	LOAN_DATE
D1	SMITH	CA	D1	11/24/2010
D2	SANDLER	CA	D2	11/24/2010
D3	SOMMERS	CA	D3	11/24/2010
D4	ARDEN	CA	D4	11/24/2010
D5	XIE	CA	D5	11/24/2010
NULL	NULL	NULL	D9	11/24/2010

RIGHT JOIN operator

- Example:

- Retrieving all departments with their employee information
- Our query will return all departments, even if they don't have any employees assigned.



RIGHT_Join.sql

```
SELECT d.DepartmentID, d.DepartmentName,
       e.EmployeeID, e.EmployeeName
  FROM Employees e
RIGHT OUTER JOIN Departments d
    ON d.DepartmentID = e.DepartmentID;
```

FULL JOIN operator

```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,  
       L.BORROWER_ID, L.LOAN_DATE  
FROM BORROWER B FULL JOIN LOAN L  
ON B.BORROWER_ID = L.BORROWER_ID
```

BORROWER_ID	LASTNAME	COUNTRY	BORROWER_ID	LOAN_DATE
D1	SMITH	CA	D1	11/24/2010
D2	SANDLER	CA	D2	11/24/2010
D3	SOMMERS	CA	D3	11/24/2010
D4	ARDEN	CA	D4	11/24/2010
D5	XIE	CA	D5	11/24/2010
D6	PETERS	CA	NULL	NULL
D7	LI	CA	NULL	NULL
D8	WONG	CA	NULL	NULL
NULL	NULL	NULL	D9	11/24/2010

FULL JOIN operator

```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,  
L.BORROWER_ID, L.LOAN_DATE  
FROM BORROWER B FULL JOIN LOAN L  
ON B.BORROWER_ID = L.BORROWER_ID
```

BORROWER_ID	LASTNAME	COUNTRY	BORROWER_ID	LOAN_DATE
D1	SMITH	CA	D1	11/24/2010
D2	SANDLER	CA	D2	11/24/2010
D3	SOMMERS	CA	D3	11/24/2010
D4	ARDEN	CA	D4	11/24/2010
D5	XIE	CA	D5	11/24/2010
D6	PETERS	CA	NULL	NULL
D7	LI	CA	NULL	NULL
D8	WONG	CA	NULL	NULL
NULL	NULL	NULL	D9	11/24/2010

FULL JOIN operator

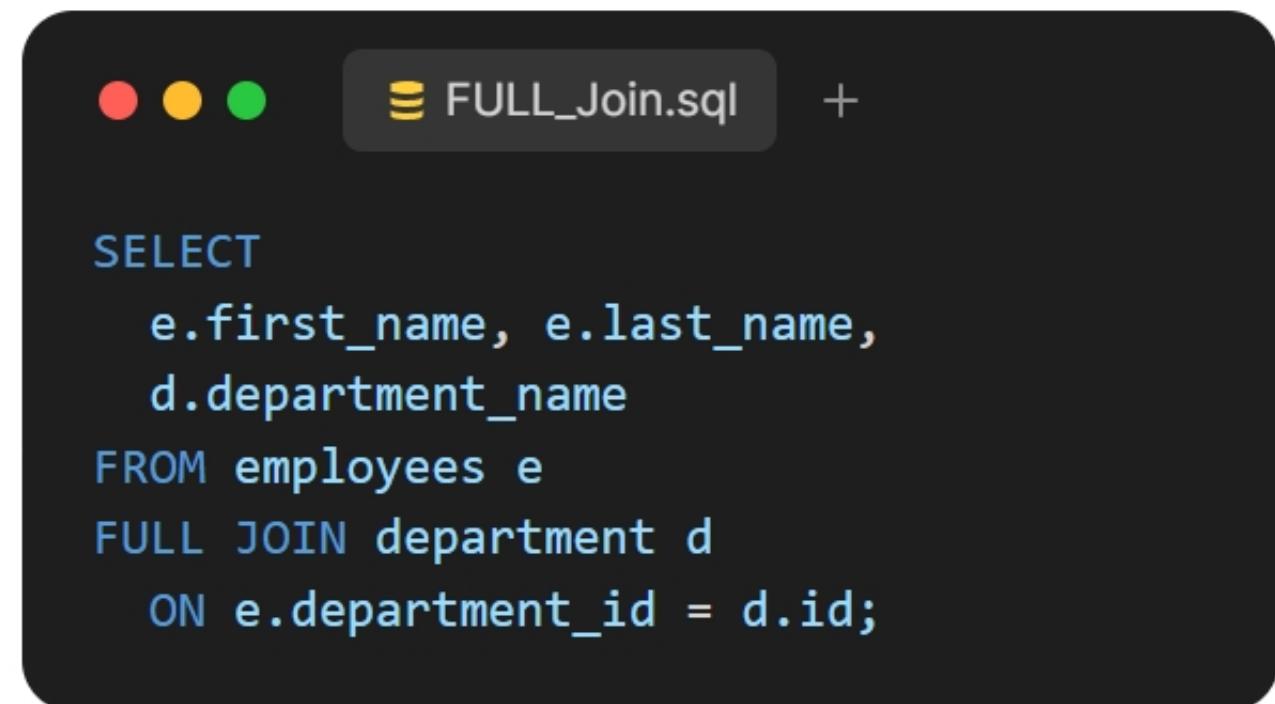
```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,  
       L.BORROWER_ID, L.LOAN_DATE  
FROM BORROWER B FULL JOIN LOAN L  
ON B.BORROWER_ID = L.BORROWER_ID
```

BORROWER_ID	LASTNAME	COUNTRY	BORROWER_ID	LOAN_DATE
D1	SMITH	CA	D1	11/24/2010
D2	SANDLER	CA	D2	11/24/2010
D3	SOMMERS	CA	D3	11/24/2010
D4	ARDEN	CA	D4	11/24/2010
D5	XIE	CA	D5	11/24/2010
D6	PETERS	CA	NULL	NULL
D7	LI	CA	NULL	NULL
D8	WONG	CA	NULL	NULL
NULL	NULL	NULL	D9	11/24/2010

FULL JOIN operator

- Example:

- Show All the Company's Employees and Departments
- We need to show all employees in a company, even if they don't have an assigned department.



The screenshot shows a code editor window with a dark theme. At the top, there are three colored circular icons (red, yellow, green) followed by the file name "FULL_Join.sql" and a "+" sign. The SQL query is displayed below:

```
SELECT e.first_name, e.last_name,
       d.department_name
  FROM employees e
 FULL JOIN department d
    ON e.department_id = d.id;
```

Module 7

Advanced SQL for Data Engineer I

CASE ..

CASE Syntax

```
CASE
    WHEN condition1 THEN result1
    WHEN condition2 THEN result2
    WHEN conditionN THEN resultN
    ELSE result
END;
```

SQL CASE Example

OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
2	10248	42	10
3	10248	72	5
4	10249	14	9
5	10249	51	40

SQL CASE Example

```
SELECT OrderID, Quantity,  
CASE  
    WHEN Quantity > 30 THEN 'The quantity is greater than 30'  
    WHEN Quantity = 30 THEN 'The quantity is 30'  
    ELSE 'The quantity is under 30'  
END AS QuantityText  
FROM OrderDetails;
```

SQL CASE Example

Result:

OrderID	Quantity	QuantityText
10248	12	The quantity is under 30
10248	10	The quantity is under 30
10248	5	The quantity is under 30
10249	9	The quantity is under 30
10249	40	The quantity is greater than 30

Q & A

Thanks