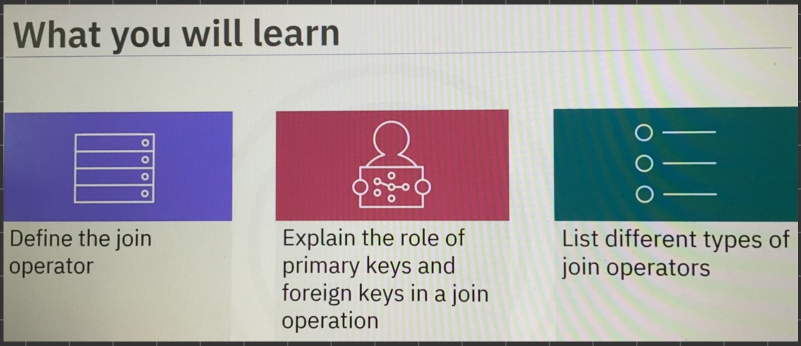
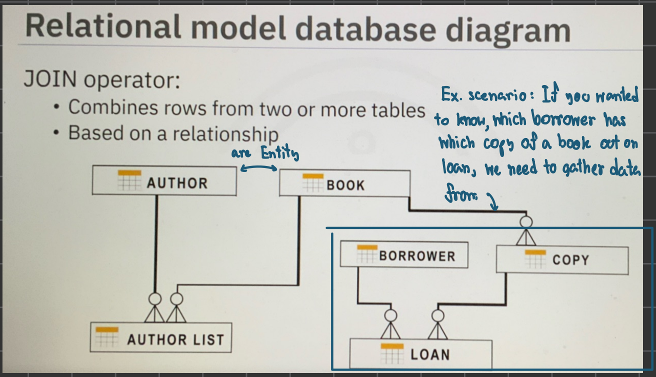
Join overview



What you will learn

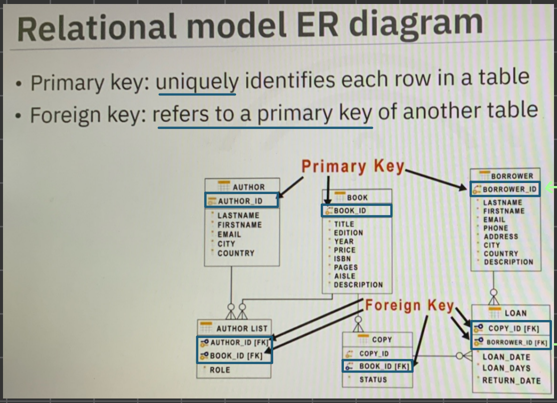
- Define the join operator
- Explain the role of primary keys and foreign keys in a join operation
- List different types of join operators



Relational model database diagram

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

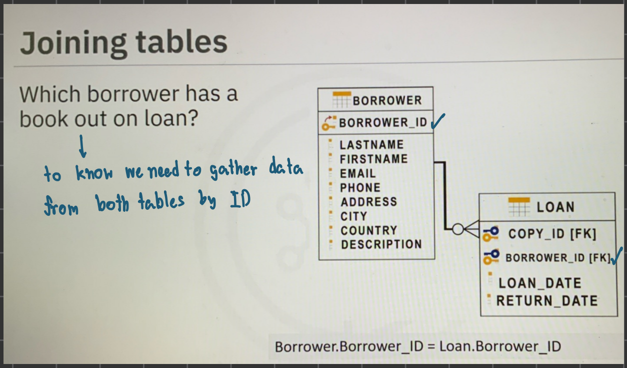
Ex. scenario: If you wanted to know, which borrower has which copy of a book out on loan, we need to gather data from



Relational model ER diagram

- Primary key: uniquely identifies each row in a table
- Foreign key: refers to a primary key of another table

in loan, att. is the foreign key, which refers to the Primary key in Borrower Table

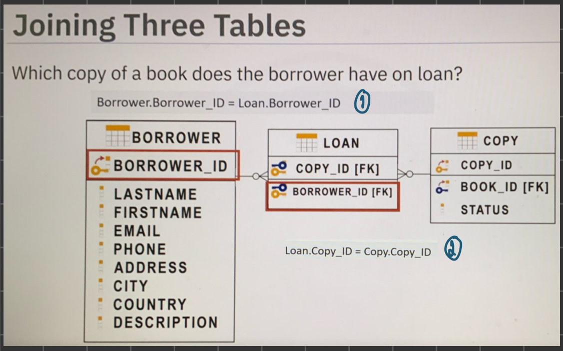


Joining tables

Which borrower has a book out on loan?

to know we need to gather data from both tables by ID

Borrower.Borrower_ID = Loan.Borrower_ID



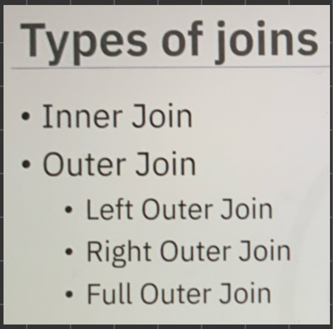
Joining Three Tables

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

Borrower.Borrower_ID = Loan.Borrower_ID 1

Loan.Copy_ID = Copy.Copy_ID 2

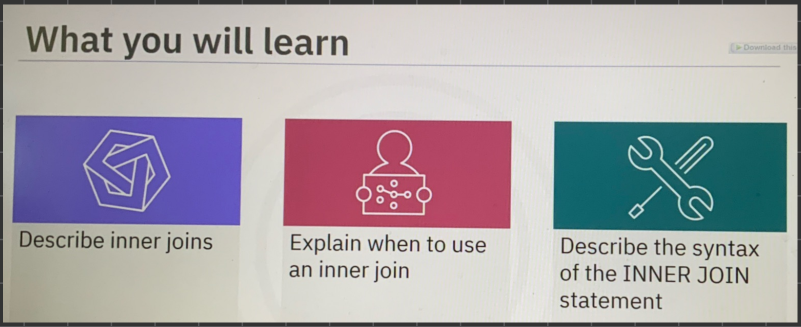
- joinning three tables, first we have to join two tables, then join another



Types of joins

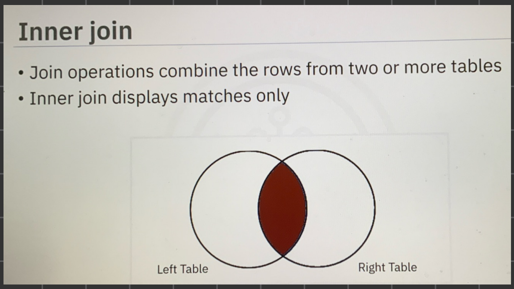
- Inner Join
- Outer Join
  - Left Outer Join
  - Right Outer Join
  - Full Outer Join

Inner Join



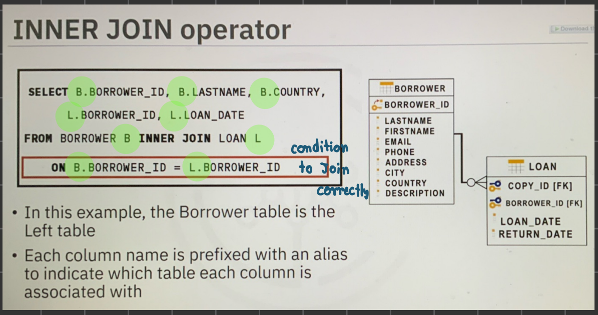
What you will learn

- Describe inner joins
- Explain when to use an inner join
- Describe the syntax of the INNER JOIN statement



Inner join

- Join operations combine the rows from two or more tables
- Inner join displays matches only

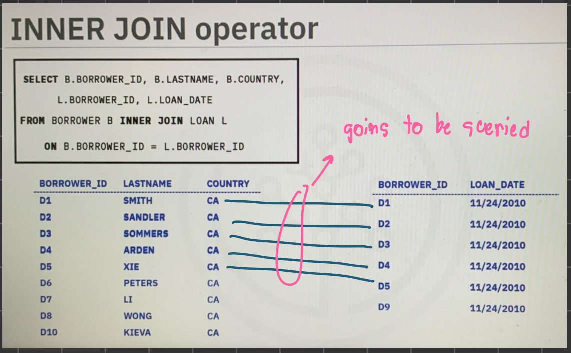


INNER JOIN operator

```
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
```

condition to Join correctly

- 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

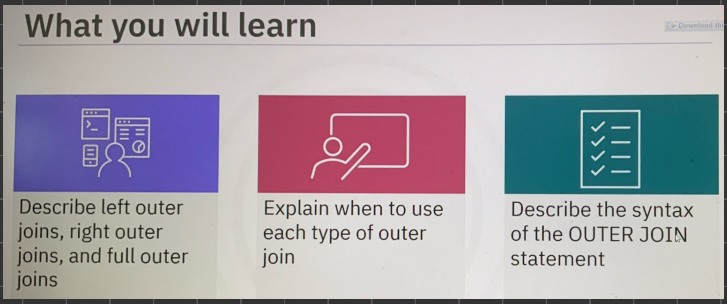
```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,
    L.BORROWER_ID, L.LOAN_DATE
FROM BORROWER B INNER JOIN LOAN L
    ON B.BORROWER_ID = L.BORROWER_ID
```

goins to be sceried

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

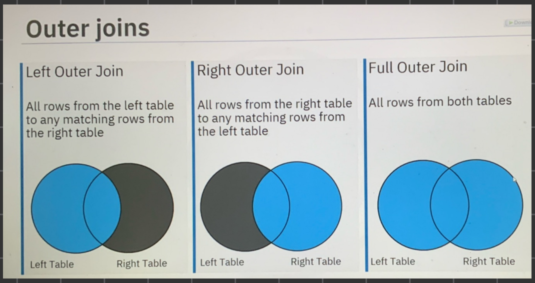
BORROWER_ID	LOAN_DATE
D1	11/24/2010
D2	11/24/2010
D3	11/24/2010
D4	11/24/2010
D5	11/24/2010
D9	11/24/2010

Outer Join



What you will learn

- Describe left outer joins, right outer joins, and full outer joins
- Explain when to use each type of outer join
- Describe the syntax of the OUTER JOIN statement



Outer joins

Left Outer Join

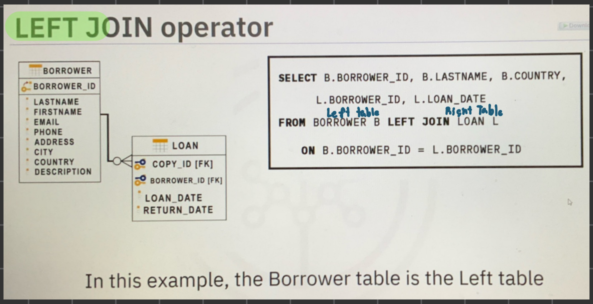
All rows from the left table to any matching rows from the right table

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

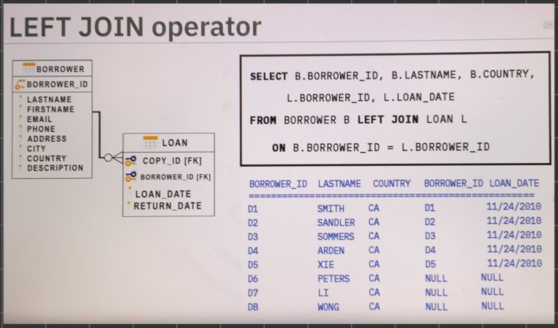


LEFT JOIN operator

```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,
     L.BORROWER_ID, L.LOAN_DATE
       Left table            Right Table
FROM BORROWER B LEFT JOIN LOAN L

   ON B.BORROWER_ID = L.BORROWER_ID
```

In this example, the Borrower table is the Left table

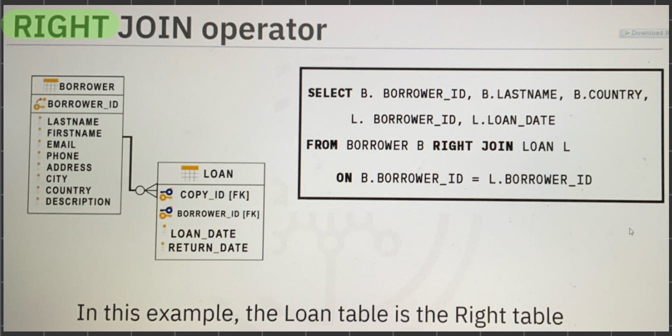


LEFT JOIN operator

```
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
```

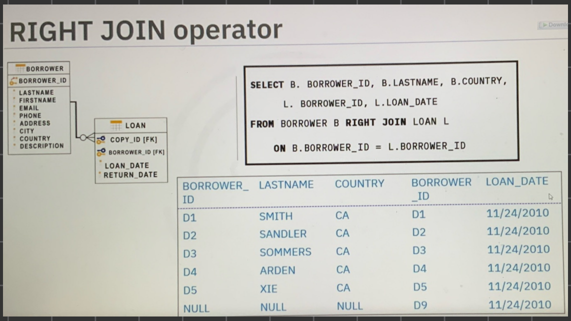


RIGHT JOIN operator

```
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

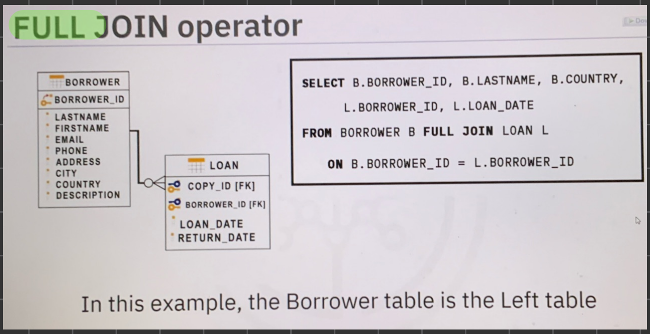


RIGHT JOIN operator

```
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

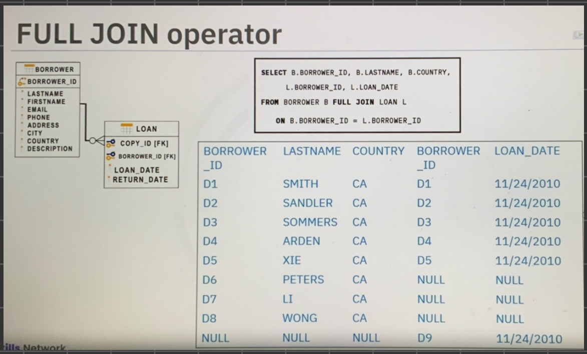


FULL JOIN operator

```
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
```

In this example, the Borrower table is the Left table



FULL JOIN operator

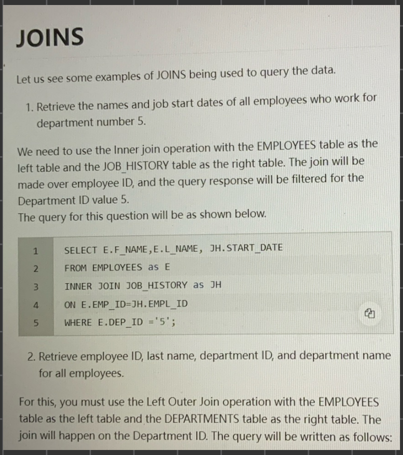
```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,
     L.BORROWER_ID, L.LOAN_DATE
FROM BORROWER B FULL JOIN LOAN L

   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

ills Network

LAB



JOINS

Let us see some examples of JOINS being used to query the data.

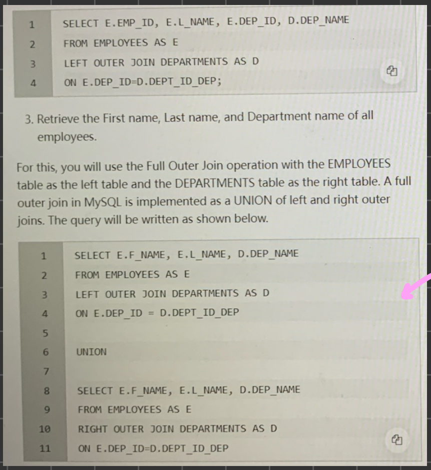
1. Retrieve the names and job start dates of all employees who work for department number 5.

We need to use the Inner join operation with the EMPLOYEES table as the left table and the JOB_HISTORY table as the right table. The join will be made over employee ID, and the query response will be filtered for the Department ID value 5.
The query for this question will be as shown below.

```
SELECT E.F_NAME,E.L_NAME, JH.START_DATE
FROM EMPLOYEES as E
INNER JOIN JOB_HISTORY as JH
ON E.EMP_ID=JH.EMPL_ID
WHERE E.DEP_ID ='5';
```

2. Retrieve employee ID, last name, department ID, and department name for all employees.

For this, you must use the Left Outer Join operation with the EMPLOYEES table as the left table and the DEPARTMENTS table as the right table. The join will happen on the Department ID. The query will be written as follows:



```
SELECT E.EMP_ID, E.L_NAME, E.DEP_ID, D.DEP_NAME
FROM EMPLOYEES AS E
LEFT OUTER JOIN DEPARTMENTS AS D
ON E.DEP_ID=D.DEPT_ID_DEP;
```

3. Retrieve the First name, Last name, and Department name of all employees.

For this, you will use the Full Outer Join operation with the EMPLOYEES table as the left table and the DEPARTMENTS table as the right table. A full outer join in MySQL is implemented as a UNION of left and right outer joins. The query will be written as shown below.

```
SELECT E.F_NAME, E.L_NAME, D.DEP_NAME
FROM EMPLOYEES AS E
LEFT OUTER JOIN DEPARTMENTS AS D
ON E.DEP_ID = D.DEPT_ID_DEP

UNION

SELECT E.F_NAME, E.L_NAME, D.DEP_NAME
FROM EMPLOYEES AS E
RIGHT OUTER JOIN DEPARTMENTS AS D
ON E.DEP_ID=D.DEPT_ID_DEP
```

besides Full join,
we can use Union
to join both left and right