Exploring Join Types

Jason P. Browning
EXECUTIVE DIRECTOR, DIXIE STATE UNIVERSITY
@jason_from_ky

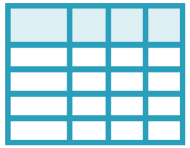
Overview

Combining records and data from multiple tables

Joining is what makes a database “relational”

Three types of joins

Data Keys



Keys are fields that describe relationships between tables

Primary key uniquely identifies each record in the table

Foreign key in a table refers to a primary key (data) in another table

Relationships

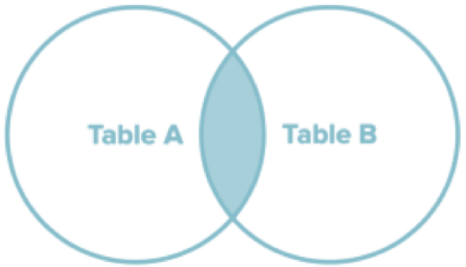
Customers

customer_id	first_name	last_name	city	state
121	John	Smith	Tuscaloosa	Alabama
122	Jane	Doe	Meridian	Mississippi
123	Bob	Clark	Santa Fe	New Mexico
124	Estella	Dodd	Tulsa	Oklahoma
125	Clair	Fletcher	Portland	Oregon

Orders

order_id	order_date	order_amt	shipped	customer_id
9001	10/19/2018	385.95	Yes	124
9002	10/22/2018	1922.40	Backorder	125
9003	10/14/2018	320.55	Yes	124
9004	10/15/2018	12.40	Yes	122
9005	10/02/2018	1350.22	Backorder	123

Inner Join



Returns all rows from two or more tables that meet the join condition

Joined fields must exist in both tables

```
SELECT customers.*,
       orders.*
  FROM customers
INNER JOIN orders
    ON customers.customer_id = orders.customer_id;
```

Inner Join

Specify source table for each field

INNER JOIN keyword specifies type of join

ON specifies fields to join

Inner Join Results

customer_id	first_name	last_name	city	state
121	John	Smith	Tuscaloosa	Alabama
122	Jane	Doe	Meridian	Mississippi
123	Bob	Clark	Santa Fe	New Mexico
124	Estella	Dodd	Tulsa	Oklahoma
125	Clair	Fletcher	Portland	Oregon

order_id	order_date	order_amt	shipped	customer_id
9001	10/19/2018	385.95	Yes	124
9002	10/22/2018	1922.40	Backorder	125
9003	10/14/2018	320.55	Yes	124
9004	10/15/2018	12.40	Yes	122
9005	10/02/2018	1350.22	Backorder	123

customer_id	first_name	last_name	city	state	order_id	order_date	order_amt	shipped	customer_id
122	Jane	Doe	Meridian	Mississippi	9004	10/15/2018	12.40	Yes	122
123	Bob	Clark	Santa Fe	New Mexico	9005	10/02/2018	1350.22	Backorder	123
124	Estela	Dodd	Tulsa	Oklahoma	9001	10/19/2018	385.95	Yes	124
124	Estella	Dodd	Tulsa	Oklahoma	9003	10/14/2018	320.55	Yes	124
125	Clair	Fletcher	Portland	Oregon	9002	10/22/2018	1922.40	Backorder	125

```
SELECT customers.first_name,
       customers.last_name,
       orders.order_date,
       orders.order_amount
  FROM customers
INNER JOIN orders
    ON customers.customer_id = orders.customer_id;
```

first_name	last_name	order_date	order_amt
Jane	Doe	10/15/2018	12.40
Bob	Clark	10/02/2018	1350.22
Estela	Dodd	10/19/2018	385.95
Estella	Dodd	10/14/2018	320.55
Clair	Fletcher	10/22/2018	1922.40

```
SELECT customers.first_name,
       customers.last_name,
       orders.order_date,
       orders.order_amount
  FROM customers
INNER JOIN orders
    ON customers.customer_id = orders.customer_id
 WHERE customers.last_name = 'Dodd'
```

first_name	last_name	order_date	order_amt
Estela	Dodd	10/19/2018	385.95
Estella	Dodd	10/14/2018	320.55

Alternative Inner Join Syntax

```
    FROM customers
INNER JOIN orders
    ON customers.customer_id = orders.customer_id
```

```
    FROM customers
    JOIN orders
      ON customers.customer_id = orders.customer_id
```

```
    FROM customers,
        orders
  WHERE customers.customer_id = orders.customer_id
```

Alias

Temporary label in a query

```
SELECT c.first_name,
       c.last_name,
       o.order_date,
       o.order_amount
  FROM customers AS c
INNER JOIN orders AS o
    ON c.customer_id = o.customer_id
 WHERE c.last_name = 'Dodd'
```

```
FROM customers c
INNER JOIN orders o
```

first_name	last_name	order_date	order_amt
Estela	Dodd	10/19/2018	385.95
Estella	Dodd	10/14/2018	320.55

Aliasing Conventions

First letter	Shortcut	ABC sequential
FROM customers c INNER JOIN orders o	FROM customers cust INNER JOIN orders ord	FROM customers a INNER JOIN orders b

Specifying the Join Fields

The ON clause is used to specify fields to join on or "match"

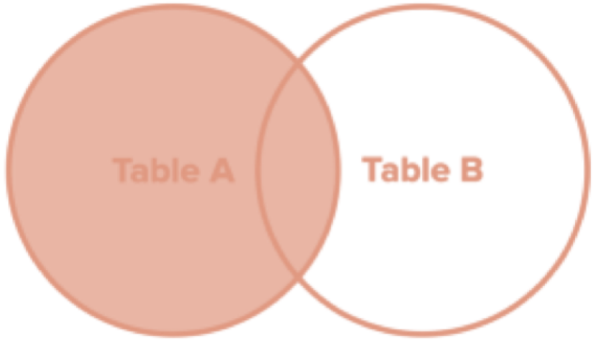
```
ON c.customer_id = o.customer_id
```

If both fields share an identical name, the keyword USING can be used

```
USING (customer_id)
```

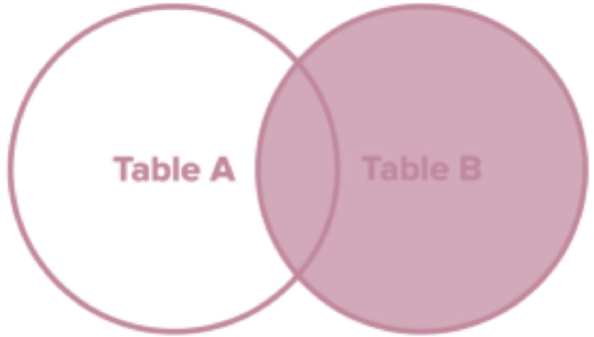
Consistency will make your code more easily readable

Outer Joins



Left Join

Returns all records from the left table along with any matching records from the right table



Right Join

Returns all records from the right table along with any matching records from the left table

```
SELECT c.first_name,
       c.last_name,
       o.order_date,
       o.order_amount
  FROM customers c
LEFT OUTER JOIN orders o
    ON c.customer_id = o.customer_id
```

Left Outer Join

LEFT OUTER JOIN keyword specifies type of join

ON specifies fields to join

Left Join Results

customer_id	first_name	last_name	city	state
121	John	Smith	Tuscaloosa	Alabama
122	Jane	Doe	Meridian	Mississippi
123	Bob	Clark	Santa Fe	New Mexico
124	Estella	Dodd	Tulsa	Oklahoma
125	Clair	Fletcher	Portland	Oregon

order_id	order_date	order_amt	shipped	customer_id
9001	10/19/2018	385.95	Yes	124
9002	10/22/2018	1922.40	Backorder	125
9003	10/14/2018	320.55	Yes	124
9004	10/15/2018	12.40	Yes	122
9005	10/02/2018	1350.22	Backorder	123

first_name	last_name	order_date	order_amt
John	Smith	[null]	[null]
Jane	Doe	10/15/2018	12.40
Bob	Clark	10/02/2018	1350.22
Estela	Dodd	10/19/2018	385.95
Estella	Dodd	10/14/2018	320.55
Clair	Fletcher	10/22/2018	1922.40

```
SELECT c.first_name,
       c.last_name,
       o.order_date,
       o.order_amount
  FROM customers c
RIGHT OUTER JOIN orders o
    ON c.customer_id = o.customer_id
```

Right Outer Join

RIGHT OUTER JOIN keyword specifies type of join

ON specifies fields to join

Right Join Results

customer_id	first_name	last_name	city	state
121	John	Smith	Tuscaloosa	Alabama
122	Jane	Doe	Meridian	Mississippi
123	Bob	Clark	Santa Fe	New Mexico
124	Estella	Dodd	Tulsa	Oklahoma
125	Clair	Fletcher	Portland	Oregon

order_id	order_date	order_amt	shipped	customer_id
9001	10/19/2018	385.95	Yes	124
9002	10/22/2018	1922.40	Backorder	125
9003	10/14/2018	320.55	Yes	124
9004	10/15/2018	12.40	Yes	122
9005	10/02/2018	1350.22	Backorder	123
9006	10/24/2018	920.40	Yes	

first_name	last_name	order_date	order_amt
[null]	[null]	10/24/2018	920.40
Jane	Doe	10/15/2018	12.40
Bob	Clark	10/02/2018	1350.22
Estela	Dodd	10/19/2018	385.95
Estella	Dodd	10/14/2018	320.55
Clair	Fletcher	10/22/2018	1922.40

Outer Join Syntax

LEFT OUTER JOIN

LEFT JOIN

```
SELECT c.first_name,
    c.last_name,
    o.order_date,
    o.order_amount
FROM customers c
RIGHT JOIN orders o
    ON c.customer_id = o.customer_id
```

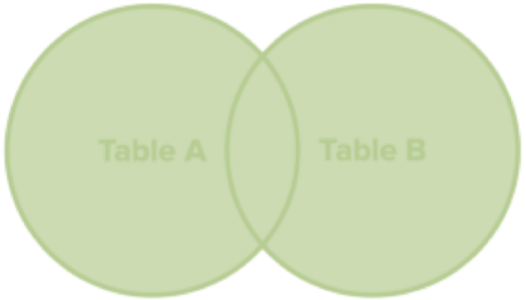
```
FROM orders o
LEFT JOIN customers c
```

Left vs. Right Joins

Left outer joins are far more prevalent in practice

Can be easier to read and interpret

Full Join



Returns all rows from two or more tables regardless of whether the join condition is met

If no match, the missing side will contain null

```
SELECT c.first_name,
       c.last_name,
       o.order_date,
       o.order_amount
  FROM customers c
FULL OUTER JOIN orders o
    ON c.customer_id = o.customer_id
```

Full Outer Join

FULL OUTER JOIN keyword specifies type of join

ON specifies fields to join

Full Join Results

customer_id	first_name	last_name	city	state
121	John	Smith	Tuscaloosa	Alabama
122	Jane	Doe	Meridian	Mississippi
123	Bob	Clark	Santa Fe	New Mexico
124	Estella	Dodd	Tulsa	Oklahoma
125	Clair	Fletcher	Portland	Oregon

order_id	order_date	order_amt	shipped	customer_id
9001	10/19/2018	385.95	Yes	124
9002	10/22/2018	1922.40	Backorder	125
9003	10/14/2018	320.55	Yes	124
9004	10/15/2018	12.40	Yes	122
9005	10/02/2018	1350.22	Backorder	123
9006	10/24/2018	920.40	Yes	

first_name	last_name	order_date	order_amt
[null]	[null]	10/24/2018	920.40
Jane	Doe	10/15/2018	12.40
Bob	Clark	10/02/2018	1350.22
Estela	Dodd	10/19/2018	385.95
Estella	Dodd	10/14/2018	320.55
Clair	Fletcher	10/22/2018	1922.40
John	Smith	[null]	[null]

Lookup tables

Database tables that contain data that specify the values for given codes

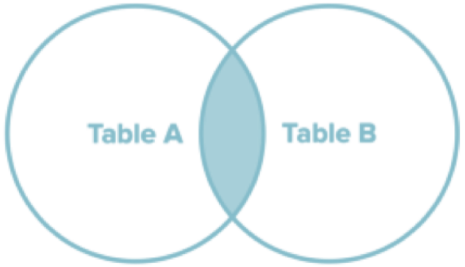
Demo

Implementing joins

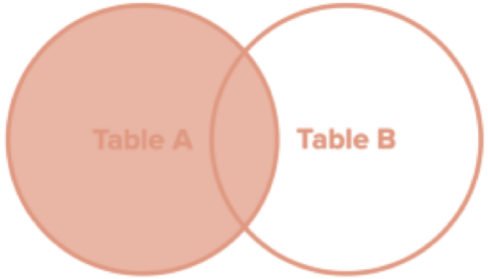
Employing lookup tables

Summary

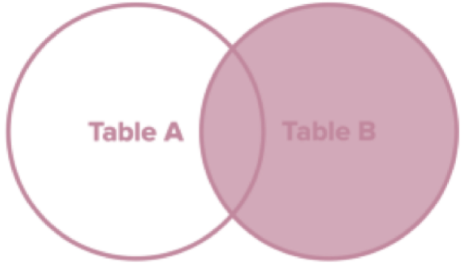
Inner join



Left outer join



Right outer join



Full outer join