NOBLE DESKTOP

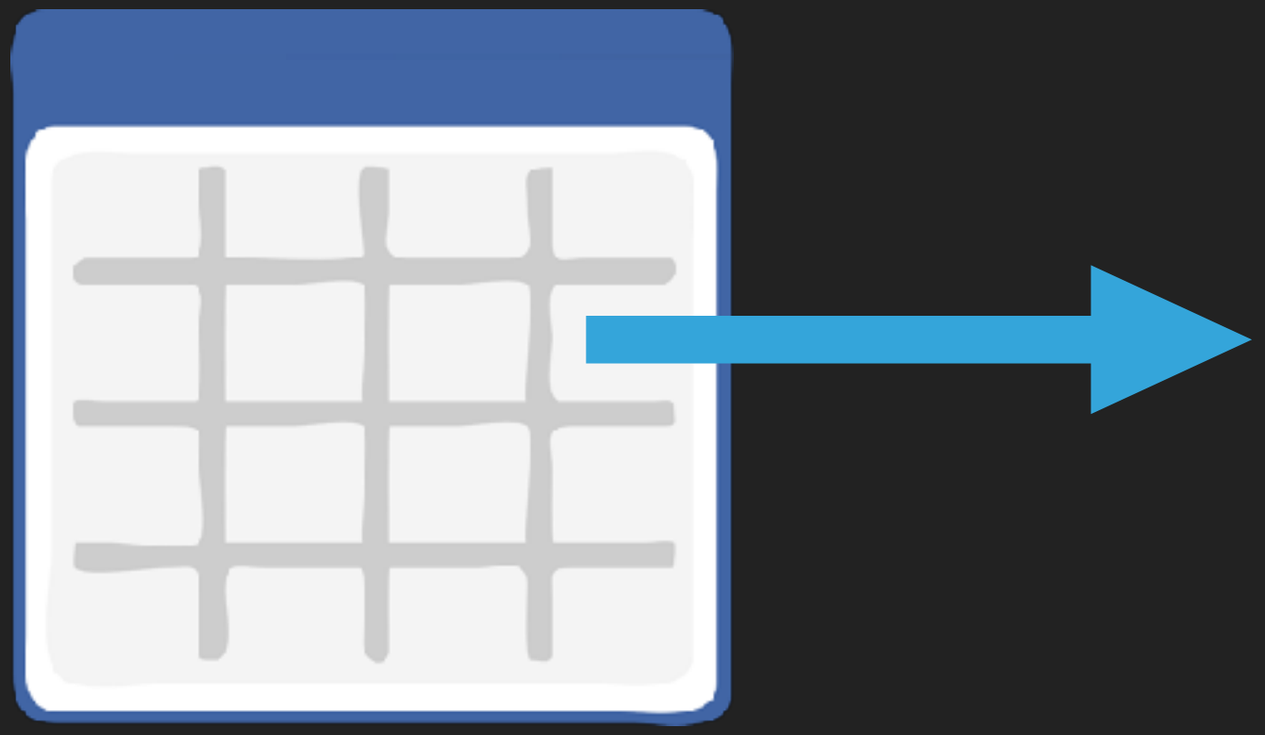
SQL BOOTCAMP

PRIMARY KEY

- Column that is unique per row
- Never changes, like a Social Security Number


LANNELS for HEROES
ockersflannels
ckersge

- Other columns can change
- We can still refer to the exact same row with primary key



FOREIGN KEY

- Reference to primary key in different table
- Like a “link” to that row

DBeaver 6.0.5 - products_pkey

Commit Rollback Auto PostgreSQL - rcarrington public@rcarrington

Database Navigator Projects

Enter a part of table name here

- PostgreSQL - rcarrington
  - anita
  - jared
  - postgres
  - **rcarrington**
    - Schemas
      - **public**
        - Tables
          - products
            - Columns
            - Constraints
              - products_pkey
            - Foreign Keys
            - Indexes
            - Dependencies
            - References
            - Partitions
            - Triggers
            - Rules
          - purchase_items
            - Columns
            - Constraints
            - Foreign Keys
              - purchase_items_product_id_fkey
              - purchase_items_purchase_id_fkey
            - Indexes
            - Dependencies
            - References
            - Partitions
            - Triggers

*<PostgreSQL - rcarrington> Script-9 products_pkey

Properties PostgreSQL - rcarrington rcarrington Schemas public

Name: products_pkey

Comment:

Owner:

Type: PR

Expression:

Object ID: 16

Constraint columns

Source

Attribute
123 id

1 items

EST en_US Sel: 0 | 0

DBeaver 6.0.5 - purchase_items_product_id_fkey

Commit Rollback Auto PostgreSQL - rcarrington public@rcarrington

Database Navigator Projects

Enter a part of table name here

- PostgreSQL - rcarrington
  - anita
  - jared
  - postgres
  - **rcarrington**
    - Schemas
      - **public**
        - standard public schema
        - products
        - purchase_items
          - Columns
          - Constraints
          - Foreign Keys
            - purchase_items_product_id_fkey
            - purchase_items_purchase_id_fkey
          - Indexes
          - Dependencies
          - References
          - Partitions
          - Triggers
          - Rules
        - purchases
        - users
    - Views
    - Materialized Views
    - Indexes
    - Functions
    - Sequences
    - Data types
    - Aggregate functions

*<PostgreSQL - rcarrington> Script-9 purchase_items_product_id_fkey

Properties

PostgreSQL - rcarrington rcarrington Schemas public Tables purchas

Name: purchase_items_product_id_fkey

Comment:

Owner: purchase_items

Type: FOREIGN KEY

Object ID: 16571

Deferrable Deferred

Foreign key columns

Source

Attribute	Referenced Column
123 product_id	id

1 items

EST en_US Sel: 0 | 0

QUESTION #1

- What e-mail address is associated with the latest purchase from Wyoming?

QUESTION #2

- The largest order from purchase_items was for what product?
- Use both price and quantity

QUESTION #3

- What is the name of the person who made the largest return?
- Use both price and quantity

JOINS

JOINS

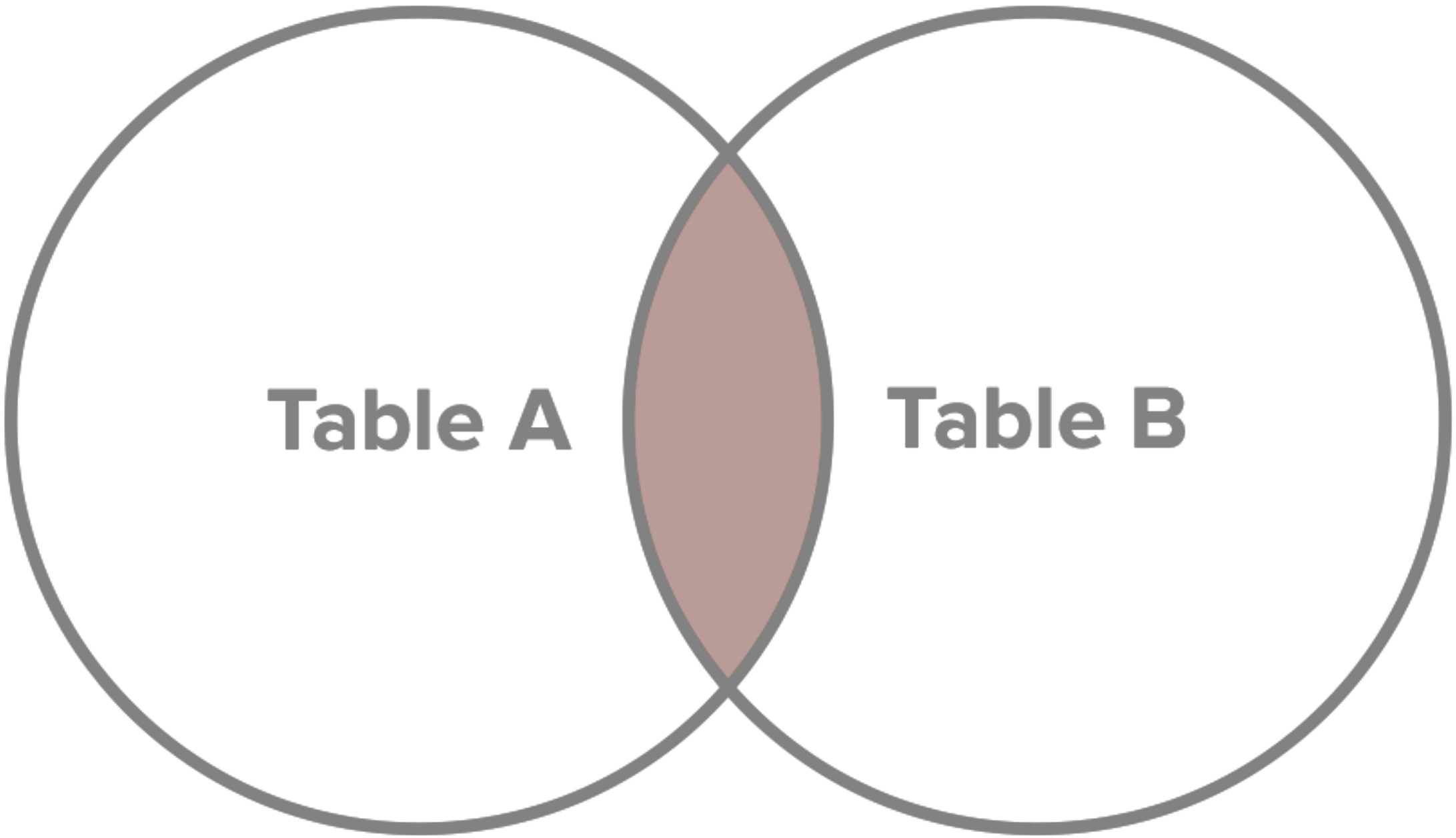
- Combine data across tables
- Relies on primary/foreign keys

- Rows are matched via primary/foreign keys and glued end-to-end

INNER JOIN

- If any primary/foreign key is missing, row is ignored
- Results may be smaller than either table being joined

Inner Join



INNER JOIN

```
SELECT name, email
FROM purchases JOIN users
ON purchases.user_id = users.id
```

ALIASES

- Nickname for a table or column in query
- Usually only a letter or two, to make query shorter

ALIASES

```
SELECT *
FROM products as pr JOIN purchases as pu
ON pr.id = pu.product_id
```

ALIASES

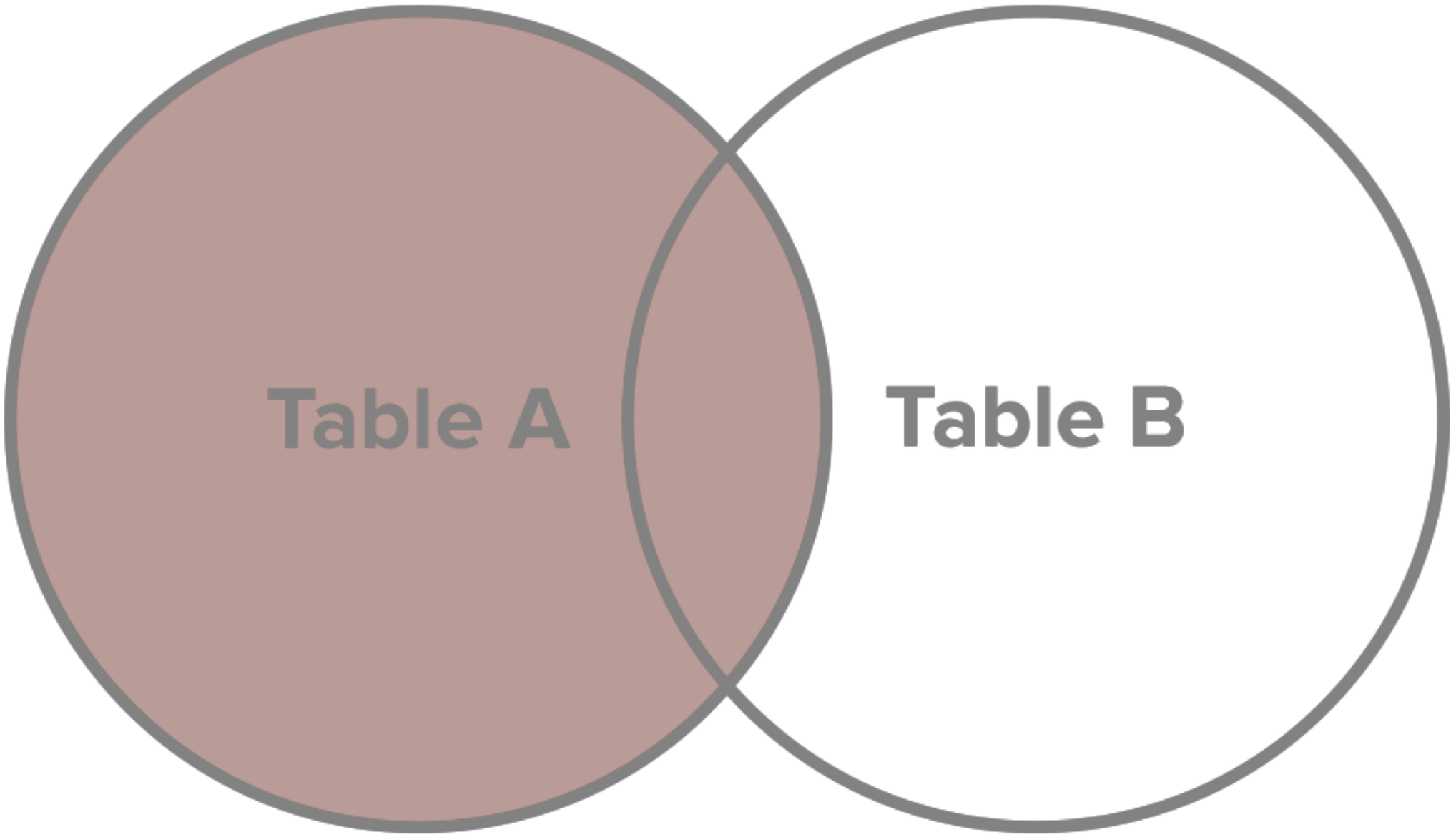
```
SELECT * FROM
JOIN products as pro,
purchases as pur;
```

EXERCISES

LEFT (OUTER) JOIN

- Missing primary/foreign key is row is filled with nulls

Left Join



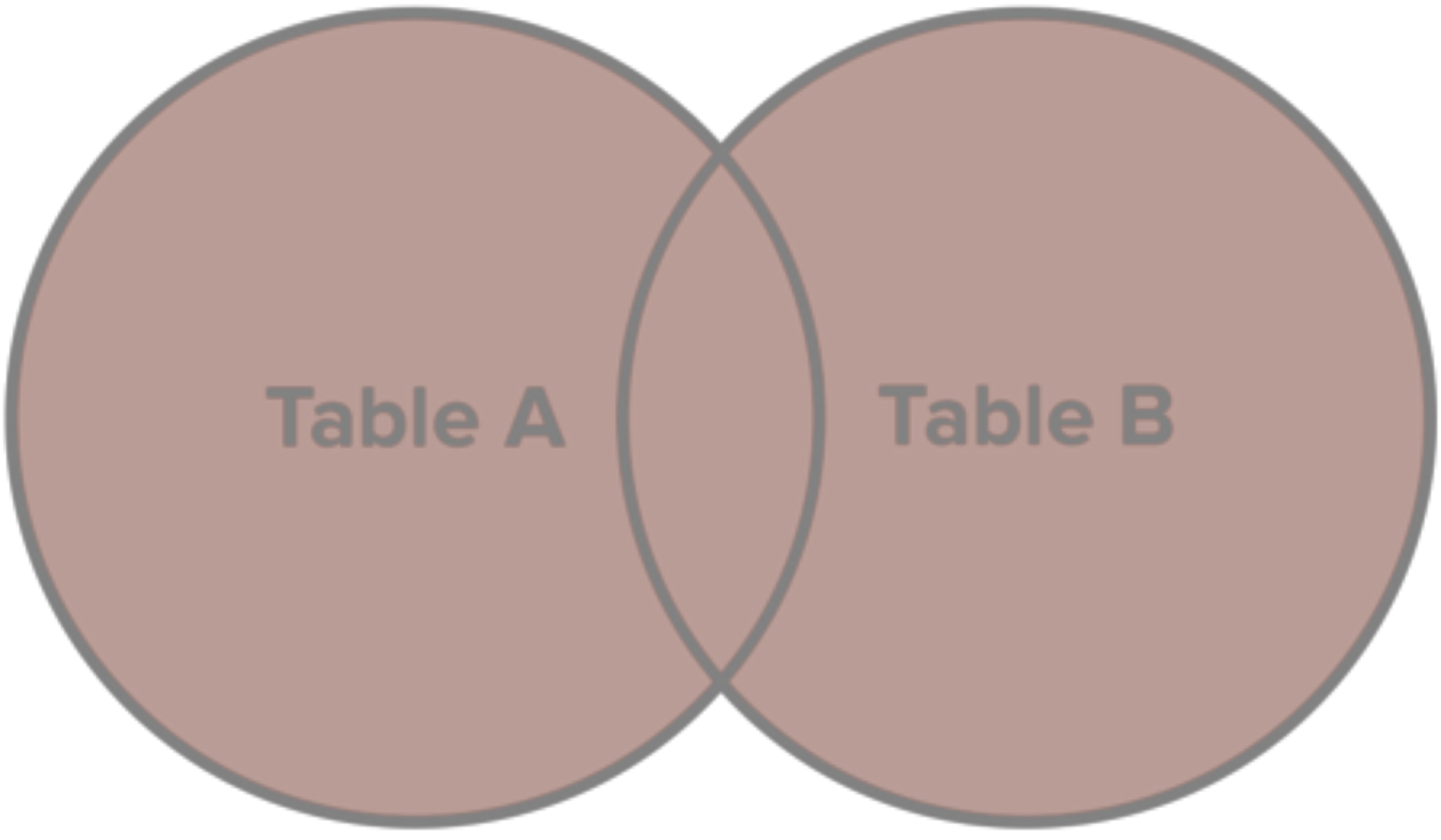
LEFT (OUTER) JOIN

```
SELECT name, email
FROM purchases LEFT JOIN users
ON purchases.user_id = users.id
```

FULL (OUTER) JOIN

- Missing primary/foreign key is row is filled with nulls
- Results will be size of largest table being joined

Full Join



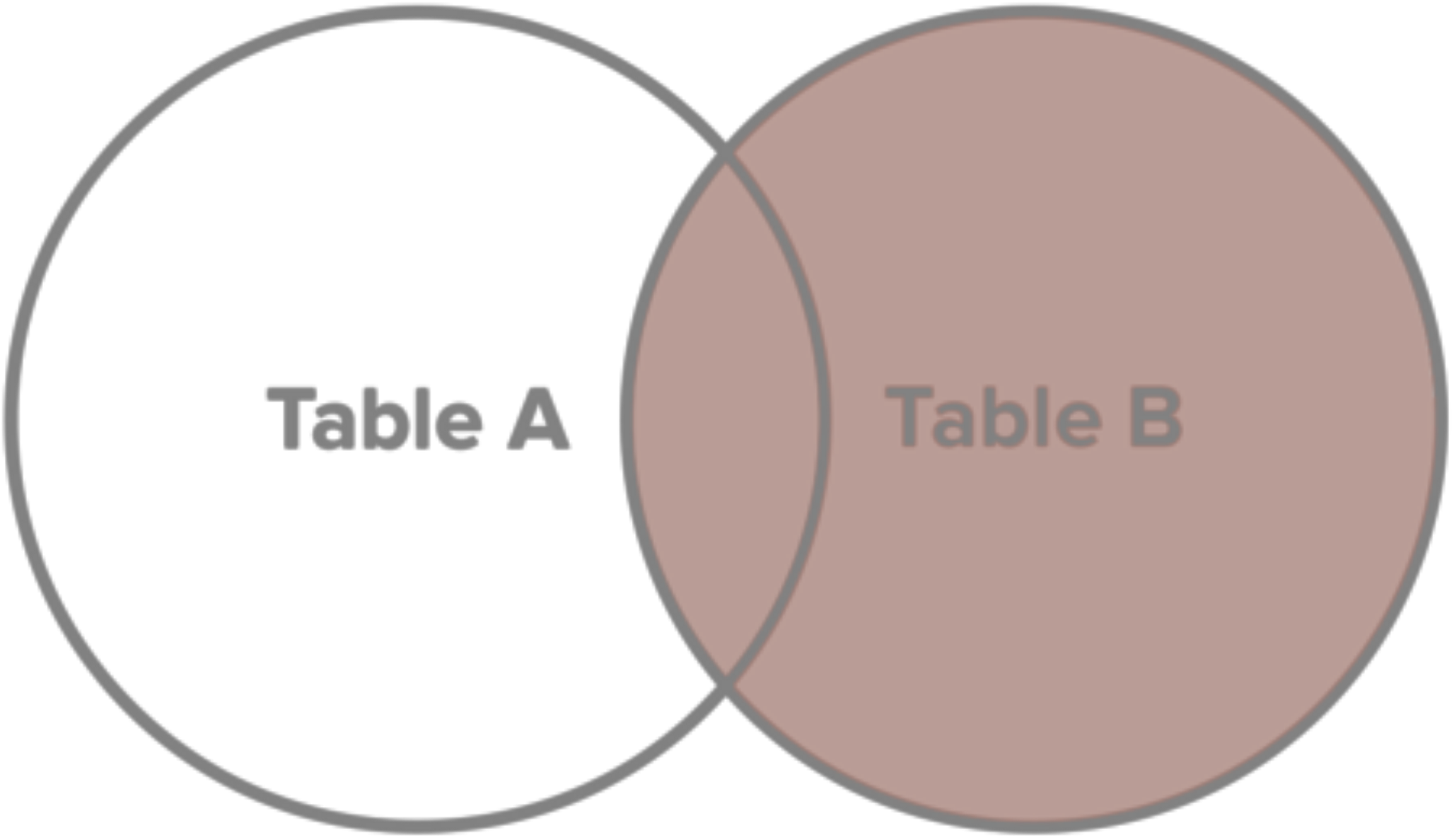
(FULL) OUTER JOIN

```
SELECT name, email
FROM purchases OUTER JOIN users
ON purchases.user_id = users.id
```

RIGHT (OUTER) JOIN

- Missing primary/foreign key is row is filled with nulls

Right Join



RIGHT (OUTER) JOIN

```
SELECT name, email
FROM purchases RIGHT JOIN users
ON purchases.user_id = users.id
```

EXERCISES