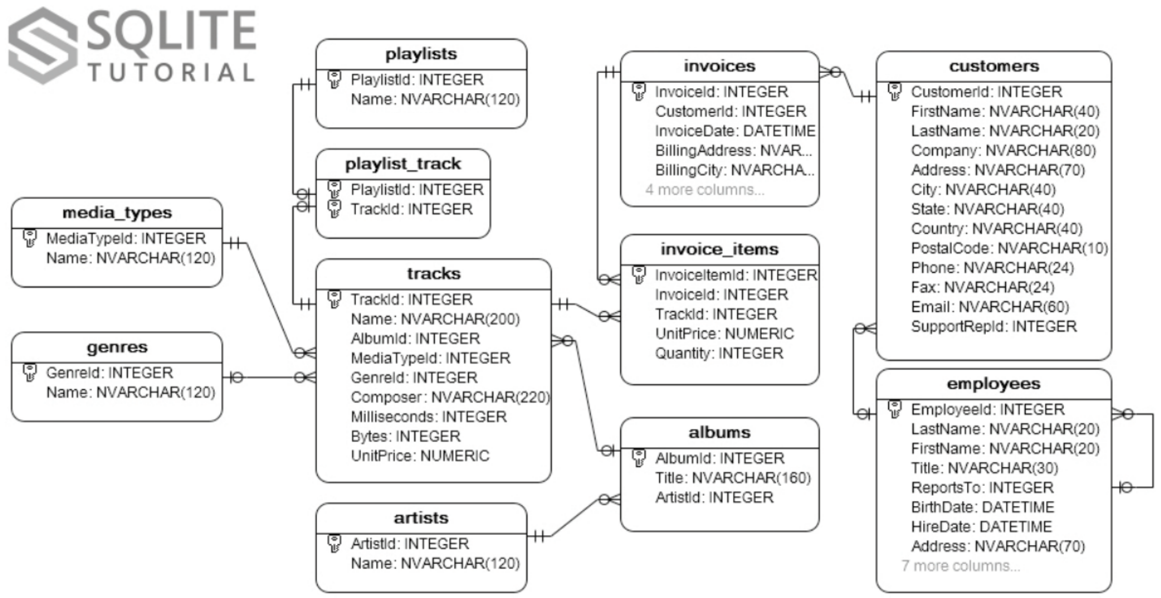
SQL Joins

Justin Post

Relational Databases

- Often want to combine data from multiple tables to summarize/model



Create Our Own Database and Do Joins!

```
import sqlite3
import pandas as pd
con = sqlite3.connect(':memory:')
cursor = con.cursor()
cursor.execute("CREATE TABLE IF NOT EXISTS dept (name TEXT, rank TEXT);")
```

```
## <sqlite3.Cursor object at 0x0000022F83CABE30>
```

```
cursor.execute(
  """
    INSERT INTO
      dept (name, rank)
    VALUES
      ("Justin", "Associate"),
      ("Jung-Ying", "Full"),
      ("Arnab", "Associate"),
      ("Spencer", "Full");
    """)
```

```
## <sqlite3.Cursor object at 0x0000022F83CABE30>
```

```
pd.read_sql("SELECT * FROM dept", con)
```

```
##           name       rank
## 0       Justin  Associate
## 1    Jung-Ying       Full
## 2        Arnab  Associate
## 3      Spencer       Full
```

Create Our Own Database and Do Joins!

```
cursor = con.cursor()
cursor.execute("CREATE TABLE IF NOT EXISTS seminar (name TEXT, topic TEXT);")
```

```
## <sqlite3.Cursor object at 0x0000022F83CDC260>
```

```
cursor.execute(
  """
    INSERT INTO
      seminar (name, topic)
    VALUES
      ("Jung-Ying", "Genetics"),
      ("Jonathan", "Design"),
      ("Arnab", "ML"),
      ("Dennis", "Non-parametrics");
    """)
```

```
## <sqlite3.Cursor object at 0x0000022F83CDC260>
```

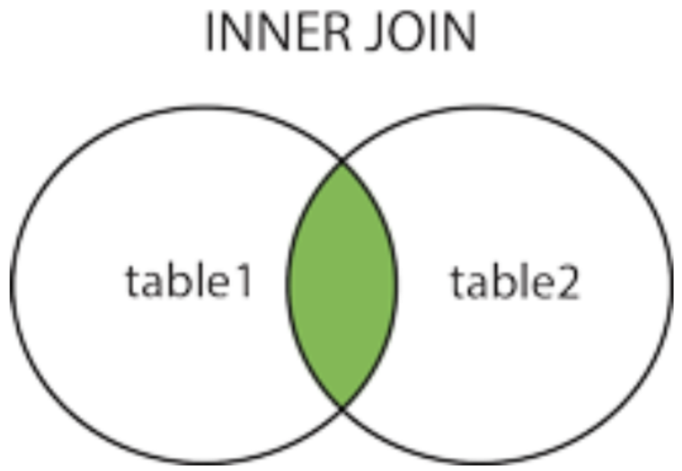
```
pd.read_sql("SELECT * FROM seminar", con)
```

```
##          name            topic
## 0   Jung-Ying         Genetics
## 1    Jonathan           Design
## 2       Arnab               ML
## 3      Dennis  Non-parametrics
```

Joins

Combining two (or more) tables in `SQL` is called doing a **join**

- Many types of joins: `left_join()`, `right_join()`, `inner_join()`, `full_join()` are most common
- Inner Join: Returns records with matching keys in both tables



Inner Join: Returns records with matching keys

Dept

```
##          name       rank
## 0      Justin  Associate
## 1   Jung-Ying       Full
## 2       Arnab  Associate
## 3     Spencer       Full
```

seminar

```
##          name             topic
## 0   Jung-Ying          Genetics
## 1    Jonathan            Design
## 2       Arnab                ML
## 3      Dennis   Non-parametrics
```

```
inner = """
  SELECT d.name, d.rank, s.topic FROM dept as d
  INNER JOIN seminar as s ON s.name = d.name
  """
pd.read_sql(inner, con)
```

```
##          name       rank     topic
## 0   Jung-Ying       Full  Genetics
## 1       Arnab  Associate        ML
```

Inner Join: Returns records with matching keys

Dept

```
##         name       rank
## 0     Justin  Associate
## 1  Jung-Ying       Full
## 2      Arnab  Associate
## 3    Spencer       Full
```

seminar

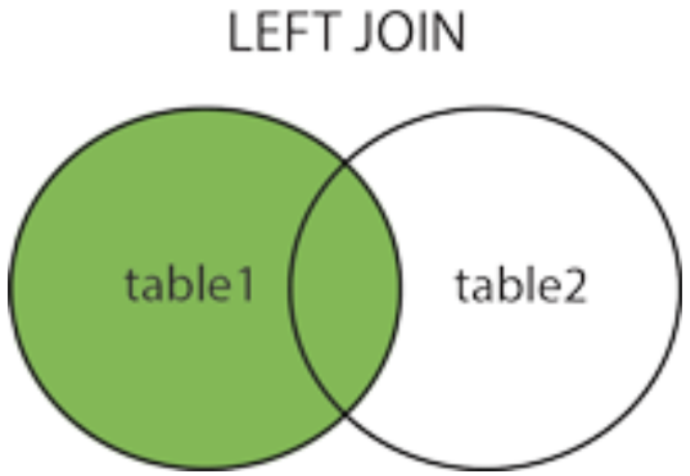
```
##         name              topic
## 0  Jung-Ying           Genetics
## 1   Jonathan             Design
## 2      Arnab                 ML
## 3     Dennis    Non-parametrics
```

```
pd.merge(
  left = pd.read_sql("SELECT * FROM dept", con),
  right = pd.read_sql("SELECT * FROM seminar", con),
  how = "inner",
  on = "name")
```

```
##         name       rank     topic
## 0  Jung-Ying       Full  Genetics
## 1      Arnab  Associate        ML
```

Joins

- Left Join: Returns all records from the 'left' table and any matching records from the 'right' table



Left Join: Return left table and matching right records

Dept

```
##         name       rank
## 0     Justin  Associate
## 1   Jung-Ying       Full
## 2      Arnab  Associate
## 3    Spencer       Full
```

seminar

```
##         name            topic
## 0  Jung-Ying         Genetics
## 1   Jonathan           Design
## 2      Arnab               ML
## 3     Dennis  Non-parametrics
```

```
left = """
  SELECT d.name, d.rank, s.topic FROM dept as d
  LEFT JOIN seminar as s ON s.name = d.name
  """
pd.read_sql(left, con)
```

```
##         name       rank     topic
## 0     Justin  Associate      None
## 1   Jung-Ying       Full  Genetics
## 2      Arnab  Associate        ML
## 3    Spencer       Full      None
```

Left Join: Return left table and matching right records

Dept

```
##         name       rank
## 0     Justin  Associate
## 1  Jung-Ying       Full
## 2      Arnab  Associate
## 3    Spencer       Full
```

seminar

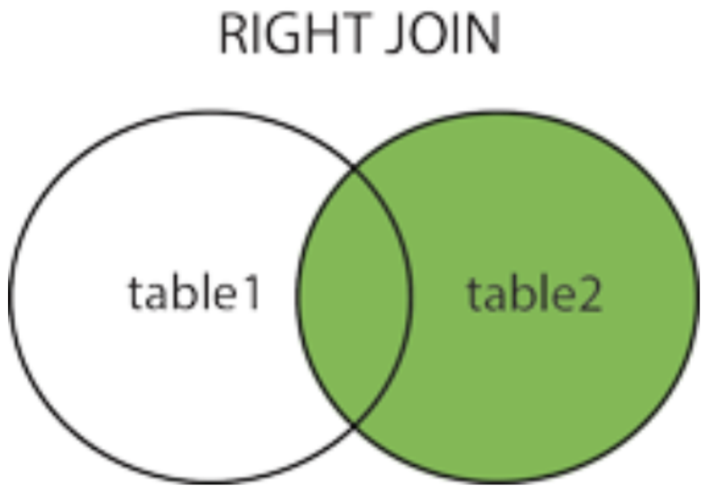
```
##         name            topic
## 0  Jung-Ying         Genetics
## 1   Jonathan           Design
## 2      Arnab               ML
## 3     Dennis  Non-parametrics
```

```
pd.merge(
  left = pd.read_sql("SELECT * FROM dept", con),
  right = pd.read_sql("SELECT * FROM seminar", con),
  how = "left",
  on = "name")
```

```
##         name       rank     topic
## 0     Justin  Associate       NaN
## 1  Jung-Ying       Full  Genetics
## 2      Arnab  Associate        ML
## 3    Spencer       Full       NaN
```

Joins

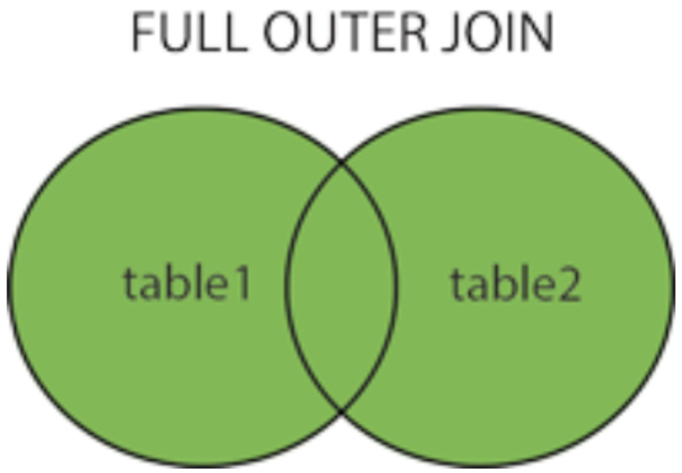
- Right Join: Returns all records from the 'right' table and any matching records from the 'left' table



- Not supported in `sqlite`'s `SQL`! Just do a left join and switch the tables.

Joins

- Outer Join: Returns all records when there is a match from the 'left' or 'right' table



- Not supported in `sqlite`'s `SQL`! Have to do some work!

Outer Join: Return all matches from both tables

Dept

```
##         name       rank
## 0     Justin  Associate
## 1  Jung-Ying       Full
## 2      Arnab  Associate
## 3    Spencer       Full
```

seminar

```
##         name            topic
## 0  Jung-Ying         Genetics
## 1   Jonathan           Design
## 2      Arnab               ML
## 3     Dennis  Non-parametrics
```

```
outer = """
  SELECT d.name, d.rank, s.topic FROM dept as d
      LEFT JOIN seminar as s ON s.name = d.name
  UNION
  SELECT s.name, d.rank, s.topic FROM seminar as s
      LEFT JOIN dept as d ON s.name = d.name
  """
pd.read_sql(outer, con)
```

```
##         name       rank            topic
## 0      Arnab  Associate               ML
## 1     Dennis       None  Non-parametrics
## 2   Jonathan       None           Design
## 3  Jung-Ying       Full         Genetics
## 4     Justin  Associate             None
## 5    Spencer       Full             None
```

Outer Join: Return all matches from both tables

Dept

```
##          name       rank
## 0      Justin  Associate
## 1   Jung-Ying       Full
## 2       Arnab  Associate
## 3     Spencer       Full
```

seminar

```
##          name            topic
## 0   Jung-Ying         Genetics
## 1    Jonathan           Design
## 2       Arnab               ML
## 3      Dennis  Non-parametrics
```

```
pd.merge(
  left = pd.read_sql("SELECT * FROM dept", con),
  right = pd.read_sql("SELECT * FROM seminar", con),
  how = "outer",
  on = "name")
```

```
##          name       rank            topic
## 0      Justin  Associate              NaN
## 1   Jung-Ying       Full         Genetics
## 2       Arnab  Associate               ML
## 3     Spencer       Full              NaN
## 4    Jonathan        NaN           Design
## 5      Dennis        NaN  Non-parametrics
```

Cross Join

Other `sqlite` supported join is the cross join

- Returns every combination of rows from the left table with the right table

```
cross = """
  SELECT * FROM dept
      CROSS JOIN seminar
  """
pd.read_sql(cross, con)
```

```
##           name       rank       name           topic
## 0       Justin  Associate  Jung-Ying        Genetics
## 1       Justin  Associate   Jonathan          Design
## 2       Justin  Associate      Arnab              ML
## 3       Justin  Associate     Dennis  Non-parametrics
## 4    Jung-Ying       Full  Jung-Ying        Genetics
## 5    Jung-Ying       Full   Jonathan          Design
## 6    Jung-Ying       Full      Arnab              ML
## 7    Jung-Ying       Full     Dennis  Non-parametrics
## 8        Arnab  Associate  Jung-Ying        Genetics
## 9        Arnab  Associate   Jonathan          Design
## 10       Arnab  Associate      Arnab              ML
## 11       Arnab  Associate     Dennis  Non-parametrics
## 12     Spencer       Full  Jung-Ying        Genetics
## 13     Spencer       Full   Jonathan          Design
## 14     Spencer       Full      Arnab              ML
```

Other Joins

Lots of other joins out there!

- See here for examples of how to implement them in sqlite!
  - The right sidebar has more than the standard joins.
- Also ways to do if then else type logic, intersections, etc.
- Can do basic summaries using `SQL` as well (including grouping), but we'll just use `python` for that!

To JupyterLab!

- Let's look at the chinook database and more involved joins!

Recap

- Joins allows us to combine two (or more) tables into one
- inner, left, and cross are all supported by `sqlite`
- `pandas` allows for left, right, outer, inner, and cross via the `pd.merge()` function
- Can write `SQL` code and use `pd.read_sql()`