

Class 3 Postgres

Elements of Databases

Sept 10, 2021

Join Queries

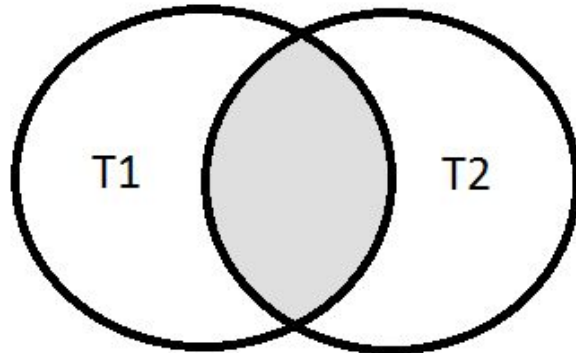
- Queries that use a JOIN operation
- Several flavors of joins
- Pervasive in relational database workloads
- Many optimizations to run efficiently

- Inner joins
- Natural joins
- Outer joins
- Right joins
- Left joins
- Full joins
- Self joins

Inner joins (and Natural joins)

```
SELECT *  
FROM T1  
[INNER] JOIN T2  
ON T1.c1 = T2.c1;
```

```
SELECT a.c1, b.c1  
FROM T1 a  
[INNER] JOIN T1 b  
USING c1;
```



Employee

<u>empid</u>	emp_name	emp_dep
2	Mike	1
23	Dave	2
3	Sarah	
5	Jim	4
6	Sunil	1
37	Morgan	4

Department

<u>depid</u>	dep_name
1	Sales
2	Product
3	Research
4	Engineering
5	HR

```
SELECT emp_name, dep_name  
FROM Employee JOIN Department  
ON emp_dep = depid
```

Result Table

emp_name	dep_name
Mike	Sales
Dave	Product
Jim	Engineering
Sunil	Sales
Morgan	Engineering

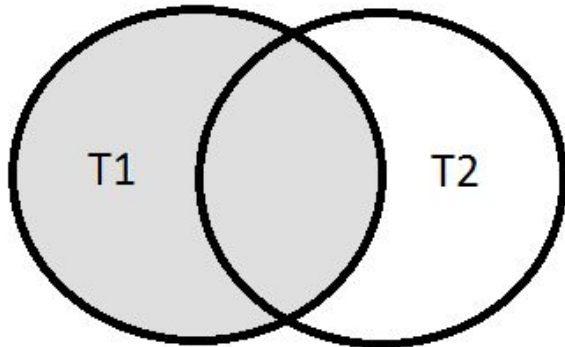
Inner Joins

```
SELECT *  
FROM T1  
[INNER] JOIN T2 ON T1.c1 = T2.c1  
[INNER] JOIN T3 ON T2.c2 = T3.c2;
```

```
SELECT *  
FROM T1  
[INNER] JOIN T2 ON T1.c1 = T2.c1 AND T1.c2 = T2.c2  
[INNER] JOIN T3 ON T2.c2 = T3.c2;
```

Left Outer Joins

```
SELECT *  
FROM T1  
LEFT [OUTER] JOIN T2  
ON T1.c1 = T2.c1;
```



Employee

<u>empid</u>	emp_name	emp_dep
2	Mike	1
23	Dave	2
3	Sarah	
5	Jim	4
6	Sunil	1
37	Morgan	4

Department

<u>depid</u>	dep_name
1	Sales
2	Product
3	Research
4	Engineering
5	HR

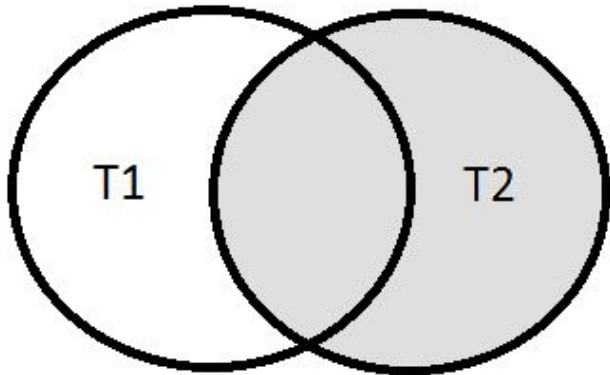
```
SELECT emp_name, dep_name  
FROM Employee LEFT JOIN Department ON emp_dep = depid  
ORDER BY emp_name
```

Result Table

emp_name	dep_name
Dave	Product
Jim	Engineering
Mike	Sales
Morgan	Engineering
Sarah	
Sunil	Sales

Right Outer Joins

```
SELECT *  
FROM T1  
RIGHT [OUTER] JOIN T2  
ON T1.c1 = T2.c1;
```



<u>empid</u>	emp_name	emp_dep
2	Mike	1
23	Dave	2
3	Sarah	
5	Jim	4
6	Sunil	1
37	Morgan	4

<u>depid</u>	dep_name
1	Sales
2	Product
3	Research
4	Engineering
5	HR

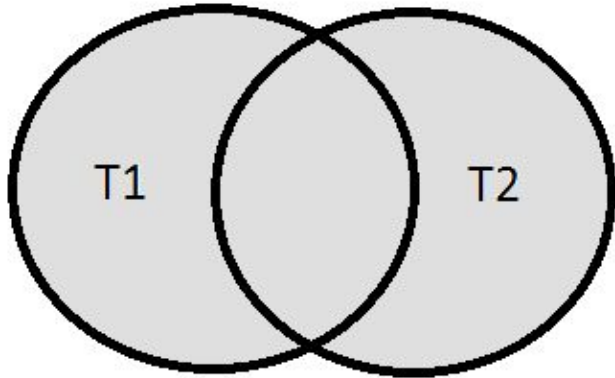
```
SELECT emp_name, dep_name  
FROM Employee RIGHT JOIN Department ON emp_dep = depid  
ORDER BY dep_name, emp_name
```

Result Table

emp_name	dep_name
Jim	Engineering
Morgan	Engineering
	HR
Dave	Product
	Research
Mike	Sales
Sunil	Sales

Full Outer Joins

```
SELECT *  
FROM T1  
FULL [OUTER] JOIN T2  
ON T1.c1 = T2.c1;
```



<u>empid</u>	emp_name	emp_dep
2	Mike	1
23	Dave	2
3	Sarah	
5	Jim	4
6	Sunil	1
37	Morgan	4

<u>depid</u>	dep_name
1	Sales
2	Product
3	Research
4	Engineering
5	HR

```
SELECT emp_name, dep_name  
FROM Employee FULL JOIN Department ON emp_dep = depid  
ORDER BY dep_name, emp_name
```

Result Table

emp_name	dep_name
Jim	Engineering
Morgan	Engineering
	HR
Dave	Product
	Research
Mike	Sales
Sunil	Sales
Sarah	

Self Joins

```
SELECT a.c1, b.c1
FROM T1 a
[INNER] JOIN T1 b
ON a.c1 = b.c1;
```

Employee

empid	emp_name	emp_dep	dob
2	Mike	1	1990-01-31
23	Dave	2	1983-04-01
3	Sarah		2001-08-02
5	Jim	4	1960-06-13
6	Sunil	1	1979-10-19
37	Morgan	4	1990-01-31

Employee

empid	emp_name	emp_dep	dob
2	Mike	1	1990-01-31
23	Dave	2	1983-04-01
3	Sarah		2001-08-02
5	Jim	4	1960-06-13
6	Sunil	1	1979-10-19
37	Morgan	4	1990-01-31

```
SELECT e1.emp_name, e2.emp_name, e2.dob
FROM Employee e1 JOIN Employee e2
ON e1.empid != e2.empid AND e1.dob = e2.dob
```

Result Table

e1.emp_name	e2.emp_name	dob
Mike	Morgan	1990-01-31
Morgan	Mike	1990-01-31

Postgres Overview:

- “The world’s most advanced open source database”
- Implements relational model
- ANSI SQL compliant
- Flexible extension mechanism
- Code base used by research and commercial projects
- Moderately easy to use
- Used for OLTP + (small) OLAP workloads
- Performs on small - medium size data (< TB)
- Performs on small - medium QPS (< 50K)
- Scale reads with read replicas
- Scale writes with application-level sharding

Instapoll on your Postgres setup

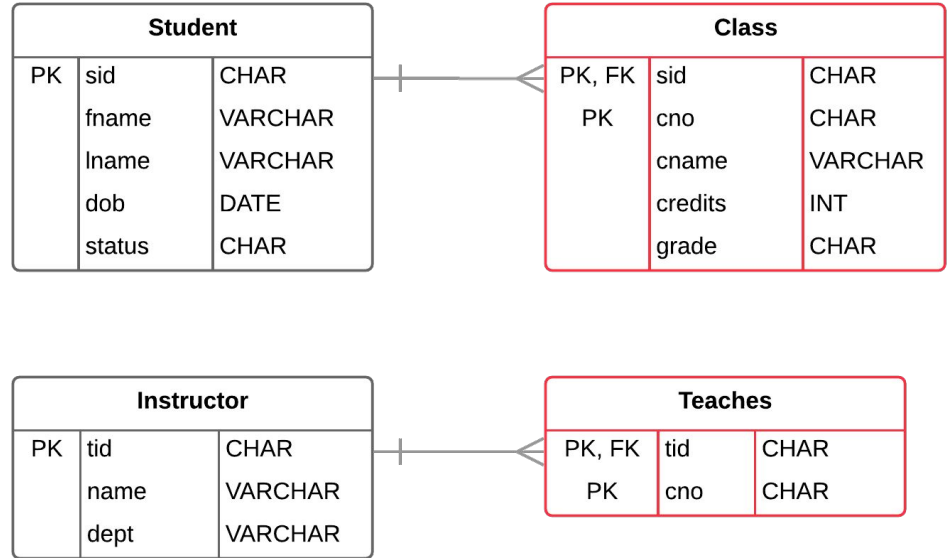
<https://github.com/cs327e-fall2021/snippets/wiki/Postgres-Setup-Guide>

1. Connect to your Postgres instance from psql
2. At the psql prompt, run command: `\dn`
3. How many schemas are listed in the output?

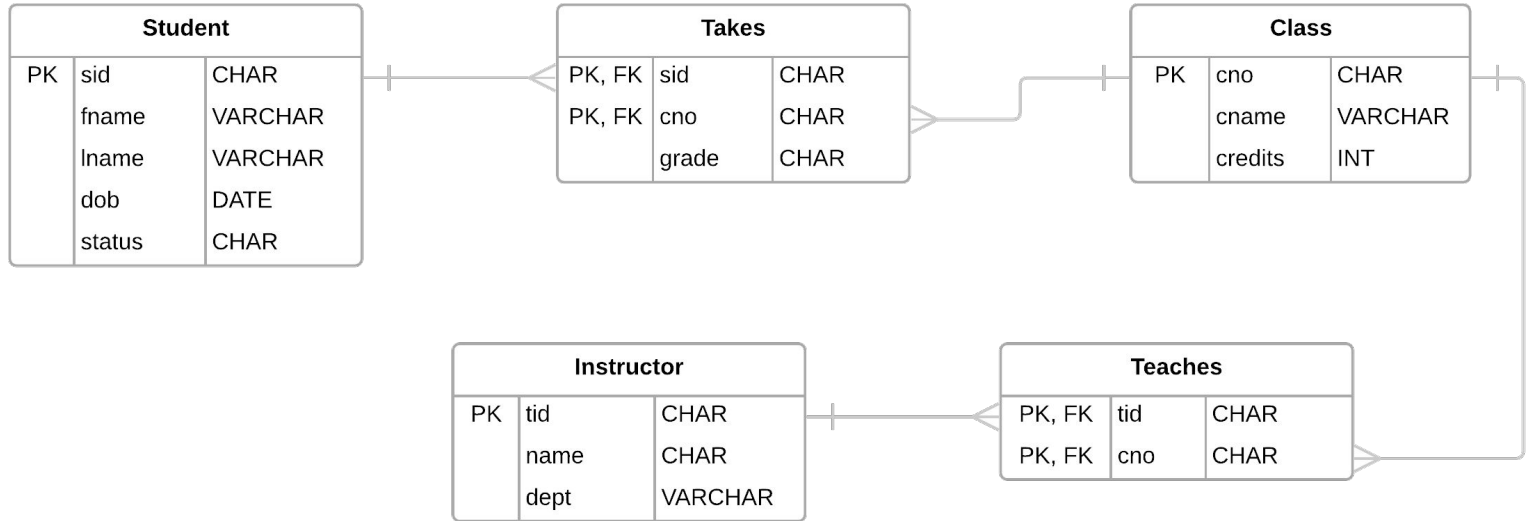
College Data Model v1

Domain Requirements:

1. A Student can take zero or more Classes.
2. A Class can have zero or more Students in it.
3. An Instructor can teach zero or more Classes.
4. A Class can be taught by zero or more Instructors.



College Data Model v2



Domain Requirements:

1. A Student can take **zero or more** Classes.
2. A Class can have **zero or more** Students in it.
3. An Instructor can teach **zero or more** Classes.
4. A Class can be taught by **zero or more** Instructors.

Postgres Code Lab:

- Clone [snippets](#) repo
- Open [postgres notebook](#)
- Create schema
- Create college tables
- Populate college tables
- Remodel tables
- Add primary keys
- Add foreign key
- Test foreign key

Practice Problem

Who are the students who take CS329E with Prof. Mitra?

For each student, return their sid, first and last names, and grade sorted by their sid.

Student(sid, fname, lname, dob, status)

Class(cno, cname, credits)

Teacher(tid, name, dept)

Takes(sid, cno, grade)

Teaches(tid, cno)

Project 2

<http://www.cs.utexas.edu/~scohen/projects/Project2.pdf>