CS 327E Class 3

Sept 11, 2020

Syntax of Join Queries

```
SELECT {c1}, {c2}, {c3}, ... {cn}
FROM {T1}
[INNER|OUTER] JOIN {T2} ON {T1.c1} = {T2.c3}
[WHERE {T1.c2} < {T2.c4}]
[ORDER BY {T1.c1}]
```
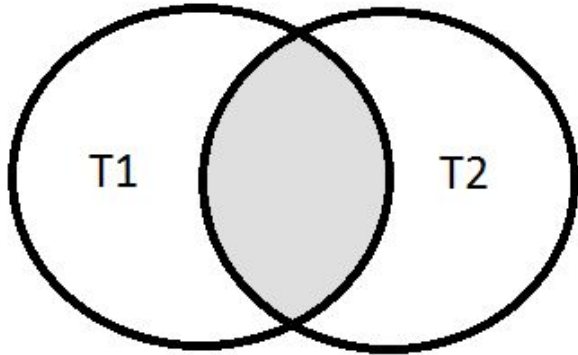
Inner Joins

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


Employee

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

Department

depid	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

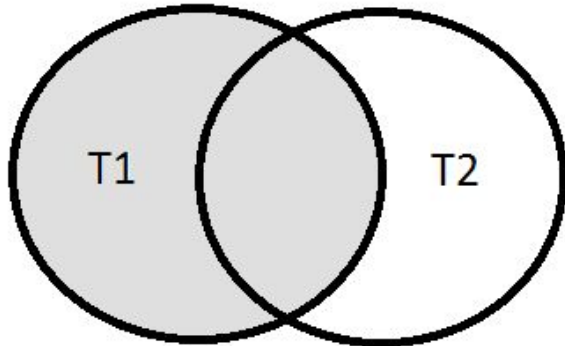| emp_name	dep_name
Mike	Sales
Dave	Product
Jim	Engineering
Sunil	Sales
Morgan	Engineering

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 Join

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

Employee

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

Department

depid	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
```
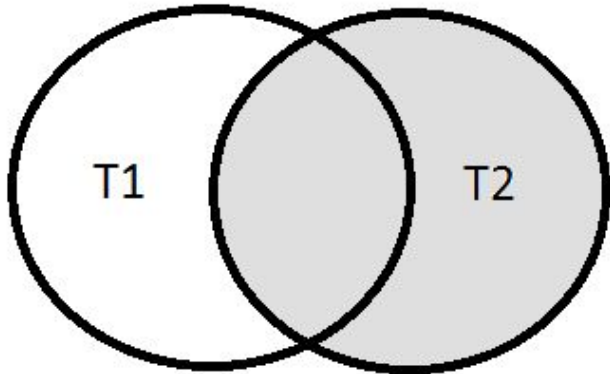
Employee

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

Department

depid	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 Join

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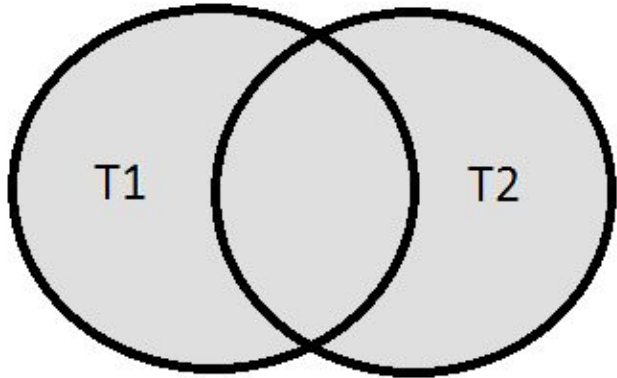
Employee

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

Department

depid	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	

Why Postgres?

- "The world's most advanced open source database"
- Relational model
- ANSI SQL compliant
- Flexible extension mechanism
- Code base used by research and commercial projects
- Moderately easy to use
- Used for OLTP + OLAP workloads
- Performs on small - medium size data (< TB)
- Performs on small - medium QPS (< 50K)
- Scaling can be complex, involving multiple efforts

Set up Postgres

https://github.com/cs327e-fall2020/snippets/wiki/Postgres-Setup-Guide

Recall: Relational Data Model Design Principles

P1. A table models one Entity Type and an Entity Type is modeled by one table.

P2. The set of fields of a table represent the attribute types of an entity.

P3. Each field is assigned a primitive type that best fits its domain of values.

P4. Each table has a Primary Key (PK) which is made up of one or more fields that uniquely represent each entity.

P5. A child table has a Foreign Key (FK) that references its parent's PK.

P6. A *M:N* relationship is modeled by one junction table.

Recall: Common Transforms

- `CREATE TABLE T2 AS SELECT a, b, c FROM T1`

- `SELECT a, b, c FROM T1`
  `UNION [DISTINCT]`
  `SELECT x AS a, y AS b, z AS c FROM T2`

- `SELECT a, b, c, 'some string' AS s FROM T1`
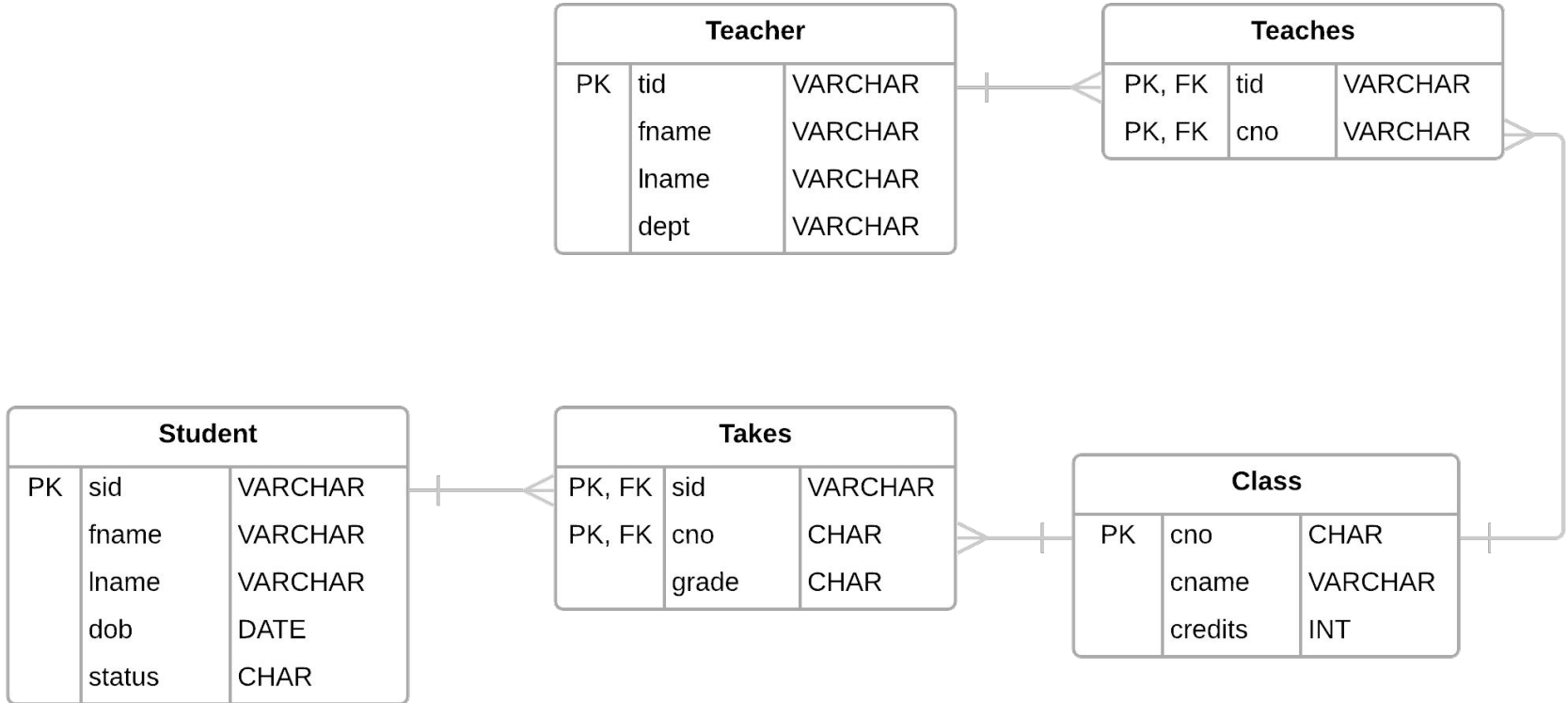 **`UNION ALL`**
  `SELECT d, e, f, 'some string' AS s FROM T2`

Continue Data Modeling Exercise

Classes	
tid	VARCHAR
instructor	VARCHAR
dept	VARCHAR
cno	VARCHAR
cname	VARCHAR
credits	INT

Current_Students	
sid	VARCHAR
fname	VARCHAR
lname	VARCHAR
dob	VARCHAR
cno	VARCHAR
cname	VARCHAR
credits	INT
grade	VARCHAR

New_Students		
PK	sid	VARCHAR
	fname	VARCHAR
	lname	VARCHAR
	dob	DATE

Lucidchart Demo

Teacher

PK	tid	VARCHAR
	fname	VARCHAR
	lname	VARCHAR
	dept	VARCHAR

Teaches

PK, FK	tid	VARCHAR
PK, FK	cno	VARCHAR

Student

PK	sid	VARCHAR
	fname	VARCHAR
	lname	VARCHAR
	dob	DATE
	status	CHAR

Takes

PK, FK	sid	VARCHAR
PK, FK	cno	CHAR
	grade	CHAR

Class

PK	cno	CHAR
	cname	VARCHAR
	credits	INT

Practice Problem

Who are the students who take CS329E with Prof. Mitra? Return their sid, first name, last name and grades sorted by sid.

Student(<u>sid</u>, fname, lname, dob, status)

Class(<u>cno</u>, cname, credits)

Teacher(<u>tid</u>, instructor, dept)

Takes(<u>sid</u>, <u>cno</u>, grade)

Teaches(<u>tid</u>, <u>cno</u>)

Project 2

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