

# SQL

by Pete Brumm

# TO FOLLOW ALONG

- [github.com/pbrumm/presentation\\_sql](https://github.com/pbrumm/presentation_sql)
- Install “sqlite manager for firefox”
  - <http://bit.ly/sqlitemanager>

# DATABASES

## YOU MAY HAVE HEARD OF

### **Enterprise**

- Oracle
- Microsoft Sql Server
- Sybase
- IBM DB2

### **Open Source**

- Postgres
- MySql
- Sqlite

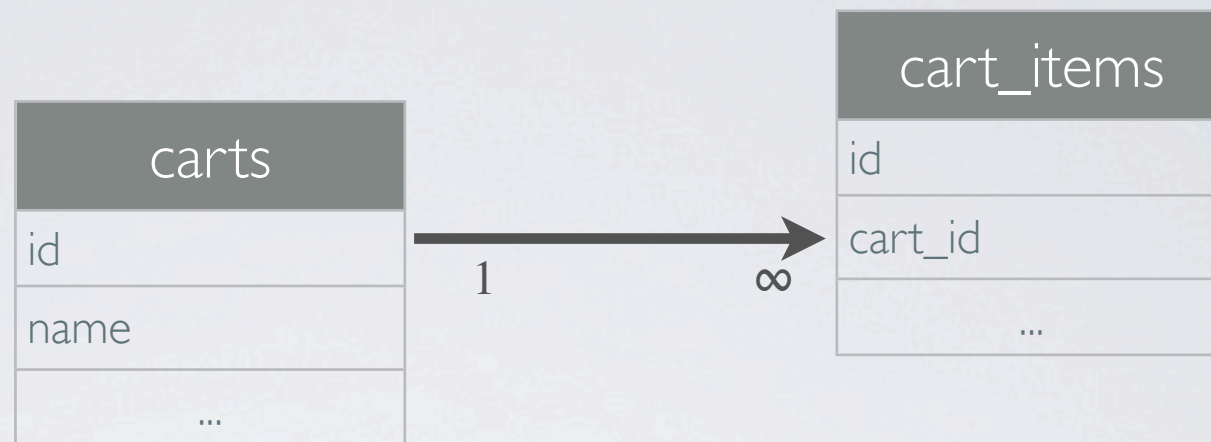


# COLUMN TYPES

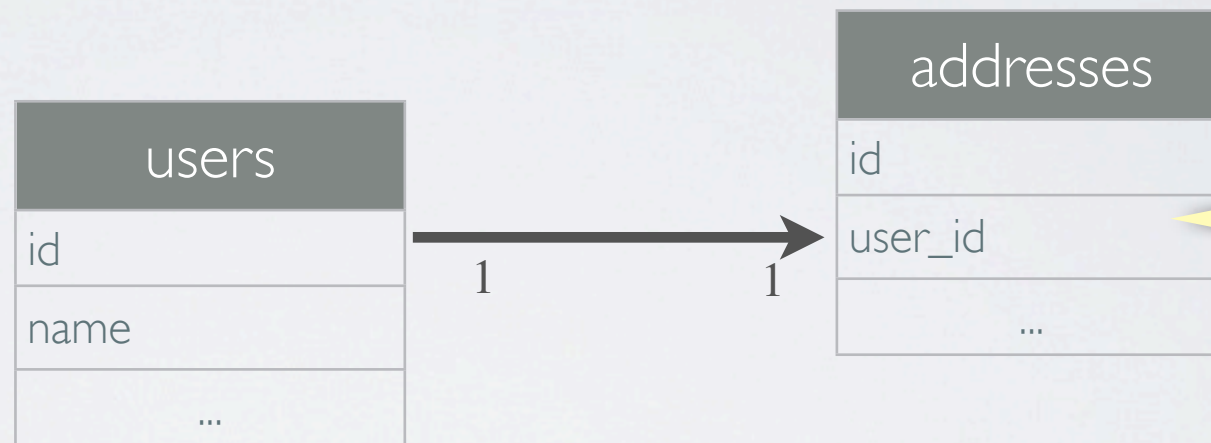
- Integer
- Float
- Double
- Varchar
- Bool
- Datetime
- Char

# DB RELATION TYPES

one to many

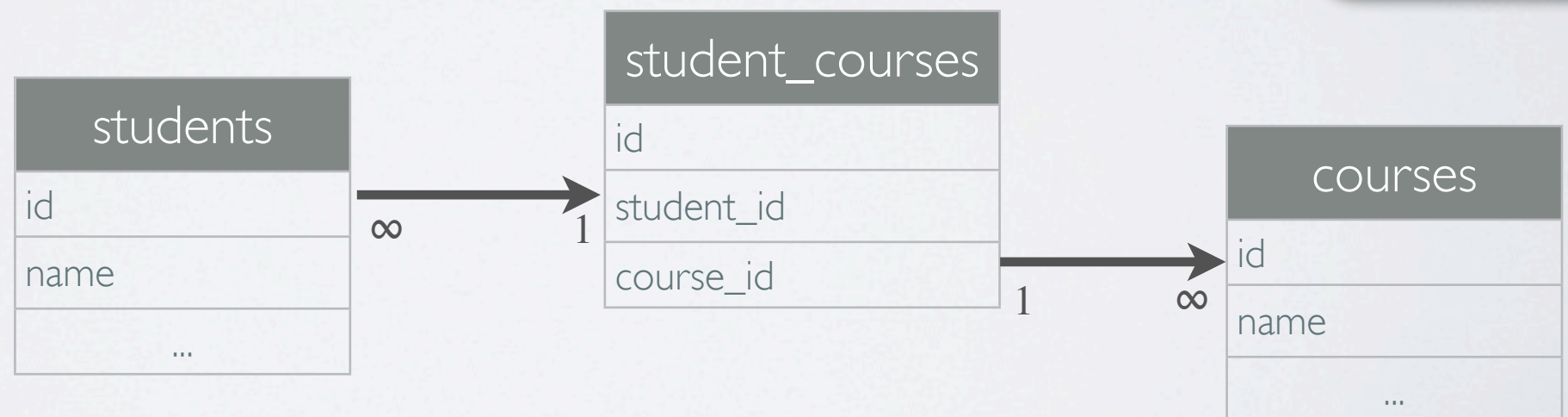


one to one



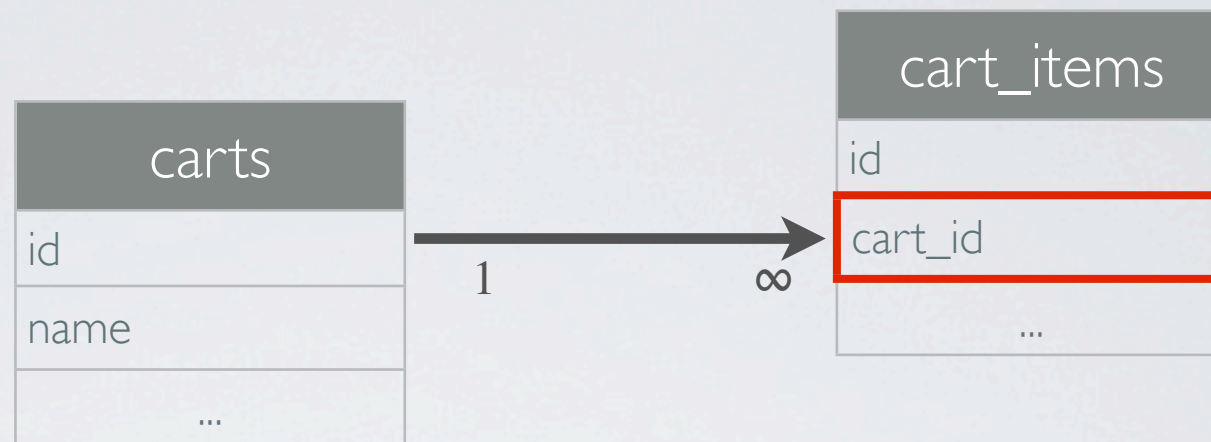
only difference  
is a unique  
index on  
cart\_id

many to many

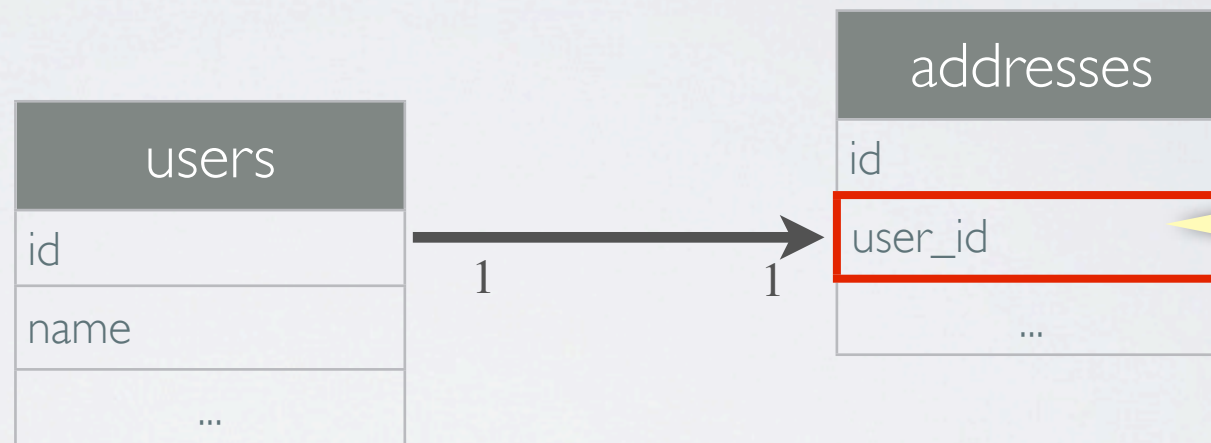


# INDEXES

one to many

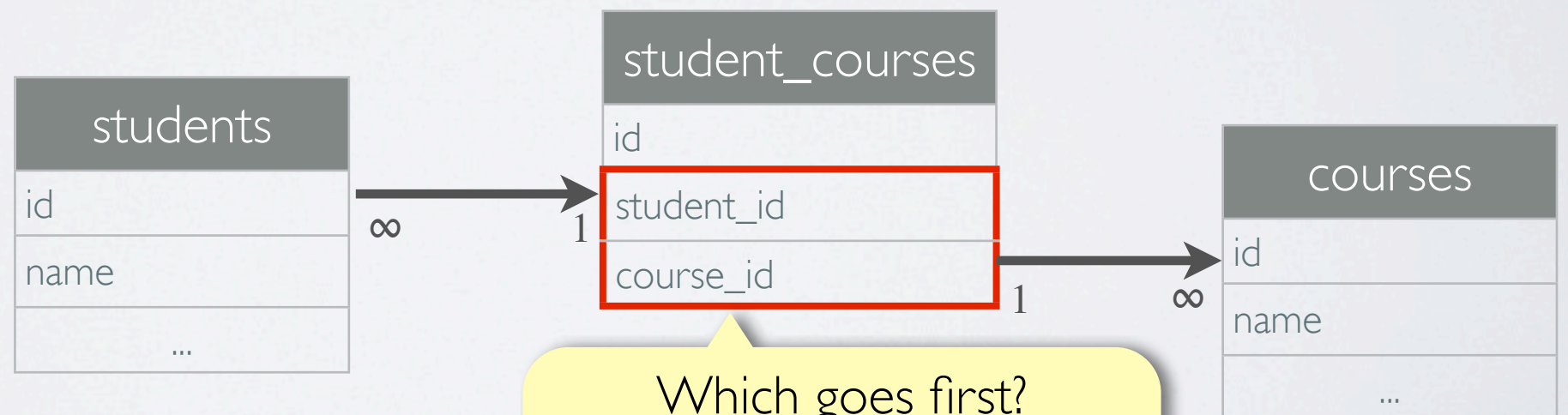


one to one



Index needs unique flag on user\_id

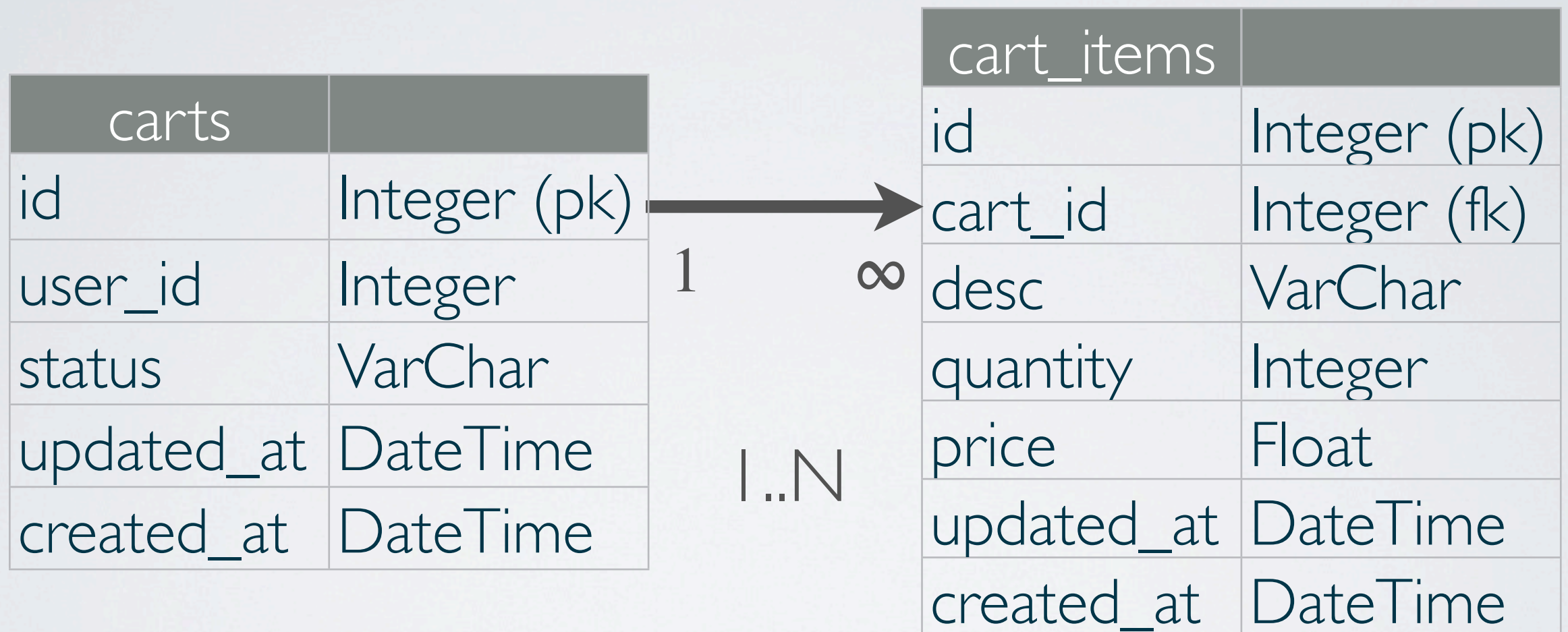
many to many



Which goes first?  
student\_id, course\_id  
or course\_id, student\_id



# EXAMPLE





Lets add a cart

**INSERT INTO** carts

(user\_id, status, total, updated\_at, created\_at)

**VALUES**

(1, 'open', 0, DATETIME('now'), DATETIME('now')) )





Lets add a cart

**INSERT INTO carts**

(user\_id, status, total, updated\_at, created\_at)

**VALUES**

(1, 'open', 0, DATETIME('now'), DATETIME('now')) )



Lets add a cart

**INSERT INTO** carts

(user\_id, status, total, updated\_at, created\_at)

**VALUES**

(1, 'open', 0, DATETIME('now'), DATETIME('now')) )





Lets add a cart

**INSERT INTO** carts

(user\_id, status, total, updated\_at, created\_at)

**VALUES**

(1, 'open', 0, DATETIME('now'), DATETIME('now')) )





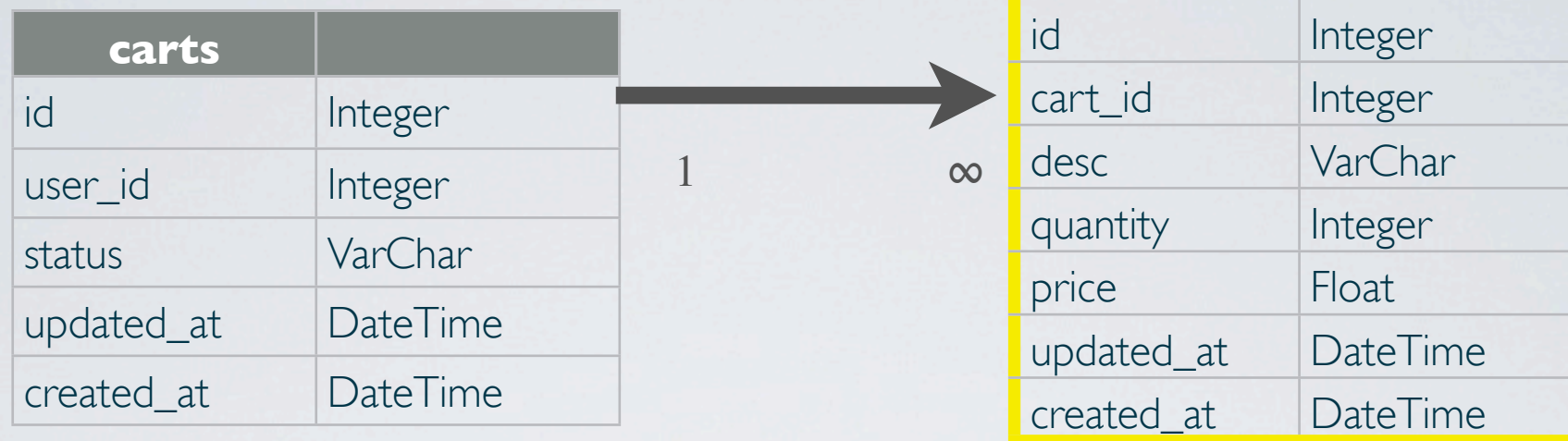
Lets add a cart

**INSERT INTO** carts

(user\_id, status, total, updated\_at, created\_at)

**VALUES**

(1, 'open', 0, DATETIME('now'), DATETIME('now')) )



Lets add a cart items

**INSERT INTO** cart\_items

(cart\_id, desc, quantity, price, updated\_at, created\_at)

**VALUES**

(1, 'book 1', 1, 35.95, DATETIME('now'), DATETIME('now')) )

**INSERT INTO** cart\_items

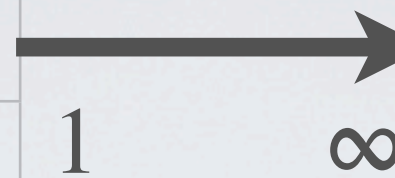
(cart\_id, desc, quantity, price, updated\_at, created\_at)

**VALUES**

(1, 'book 2', 2, 45.95, DATETIME('now'), DATETIME('now')) )



carts	
id	Integer
user_id	Integer
status	VarChar
updated_at	DateTime
created_at	DateTime

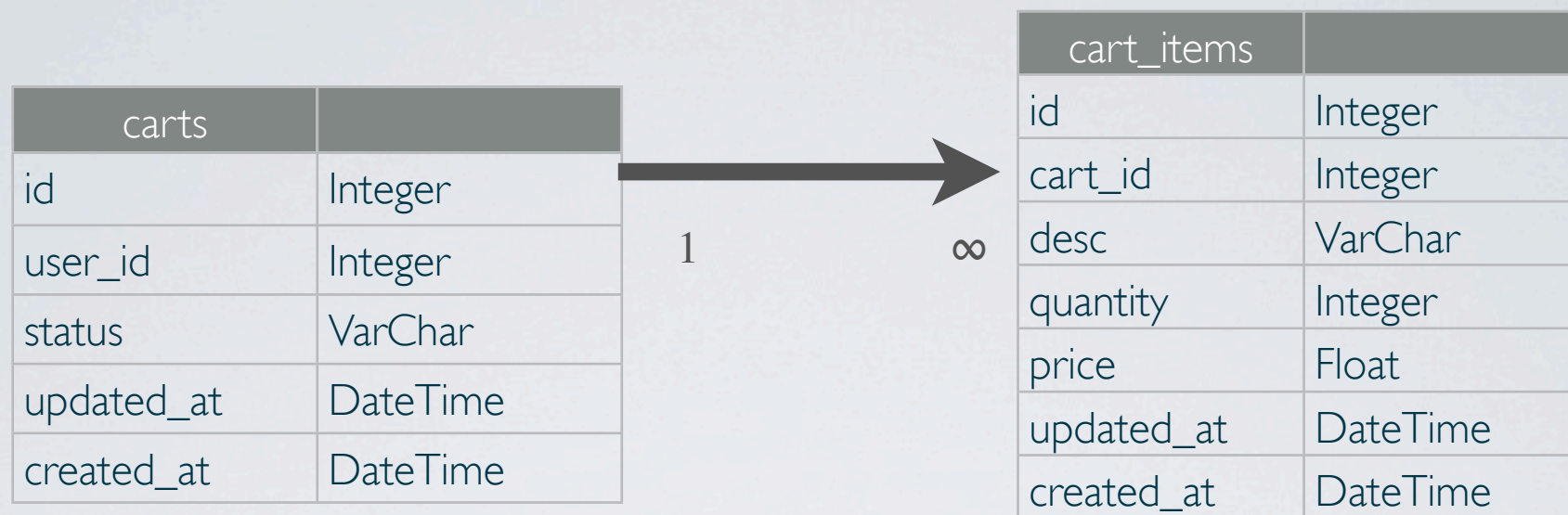


cart_items	
id	Integer
cart_id	Integer
desc	VarChar
quantity	Integer
price	Float
updated_at	DateTime
created_at	DateTime

What types of questions can be answered?

- Get all cart\_items for a cart
- Remove an cart\_item from a cart
- Find all carts that have a status of “open”





Lets query some data

- Get all cart\_items for a cart
  - **SELECT \* FROM cart\_items WHERE cart\_id = 1**
- Remove an cart\_item from a cart
  - **DELETE FROM cart\_items WHERE id = 2**
- Find all carts that have a status of “open”
  - **SELECT \* FROM carts WHERE status = 'open'**

get cart with total price IMPLICIT JOIN, JOIN, INNER JOIN

```
SELECT carts.*, SUM(cart_items.price)
FROM carts, cart_items
WHERE carts.id = cart_items.cart_id AND carts.id = 1
```

```
SELECT carts.*, SUM(cart_items.price)
FROM carts
JOIN cart_items WHERE carts.id = cart_items.cart_id
WHERE carts.id = 1
```

```
SELECT carts.*, SUM(cart_items.price)
FROM carts
INNER JOIN cart_items WHERE carts.id = cart_items.cart_id
WHERE carts.id = 1
```



get cart with total price IMPLICIT JOIN, JOIN, INNER JOIN

SELECT carts.\*, SUM(cart\_items.price)

FROM cart

WHERE ca

SELECT ca

FROM cart

**JOIN** cart\_

WHERE ca

SELECT ca

FROM carts

**INNER JOIN** cart\_items WHERE carts.id = cart\_items.cart\_id

WHERE carts.id = 1

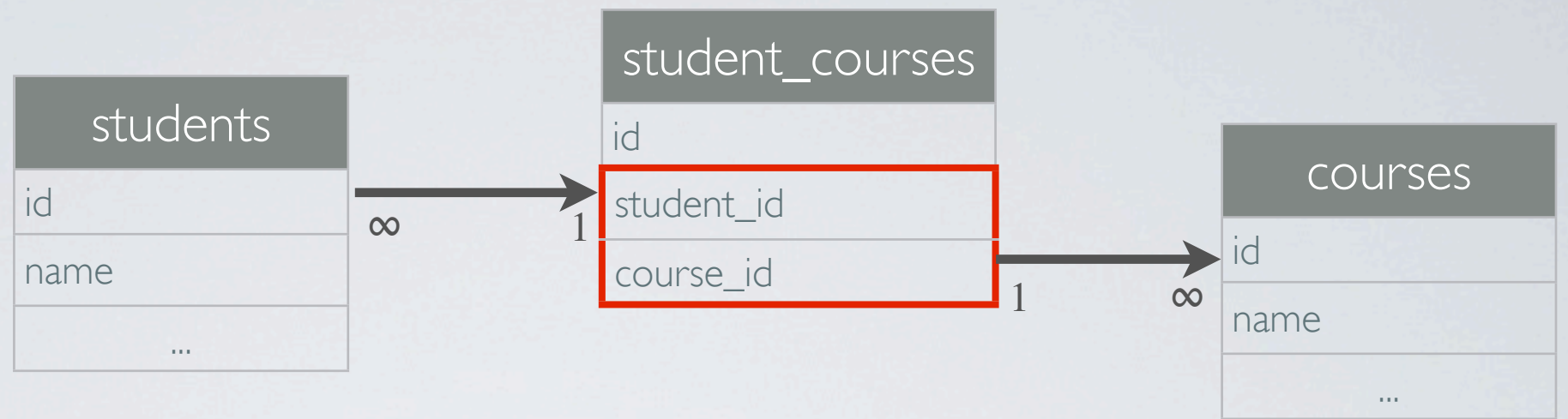
user_id	sum	status	...
1	81.90	open	...

= 1

\_id



many to many



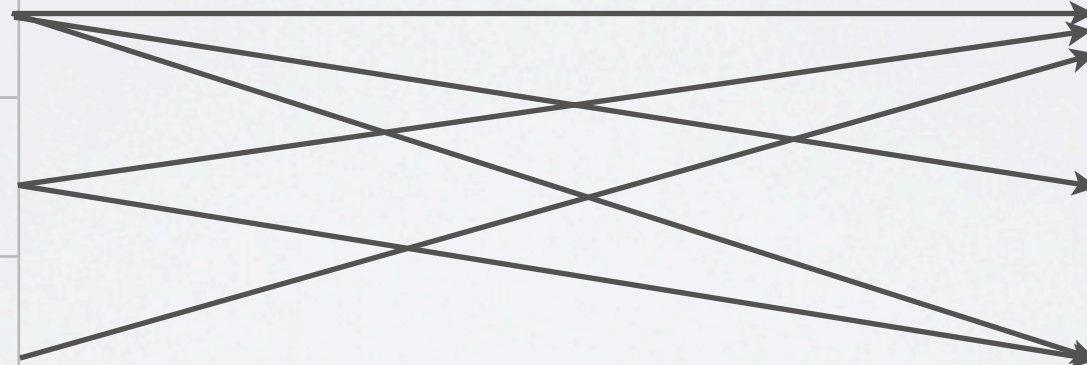
Students

1	Pete
2	Mike
3	Drew
4	Karen

StudentCourses

Courses

1	Ruby
2	Rails
3	Javascript
4	.Net



many to many

id
name

## student\_courses

student_id	course_id
1	1
1	2
1	3
2	1
2	3
3	1

courses
id
name
...

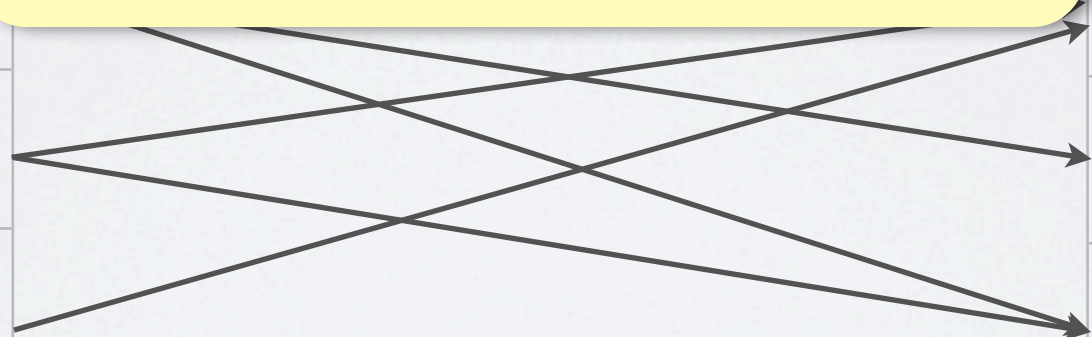
∞

## Students

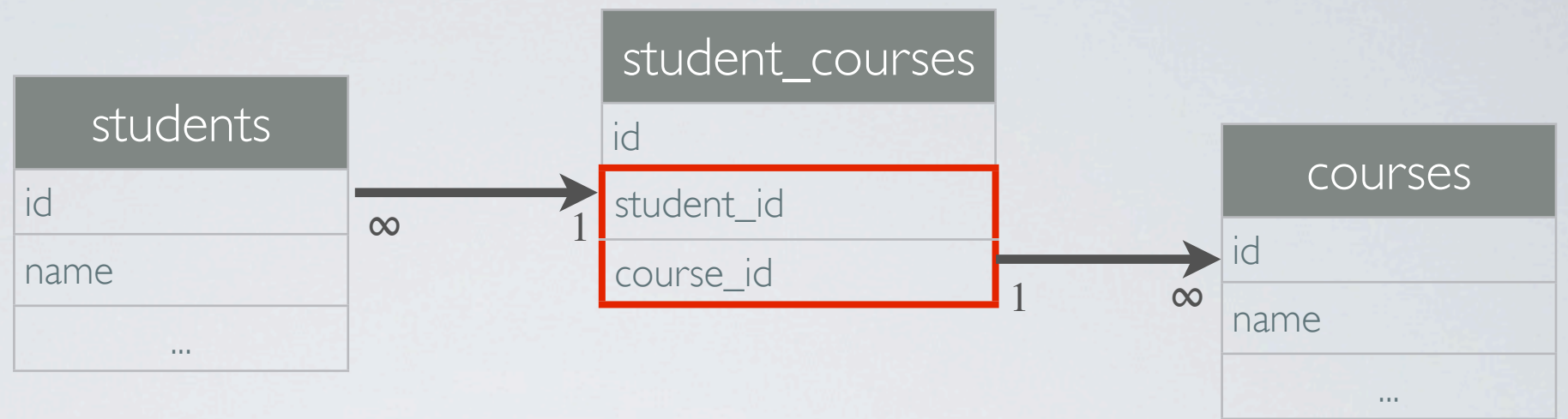
1	Pete
2	Mike
3	Drew
4	Karen

## Courses

1	Ruby
2	Rails
3	Javascript
4	.Net



many to many

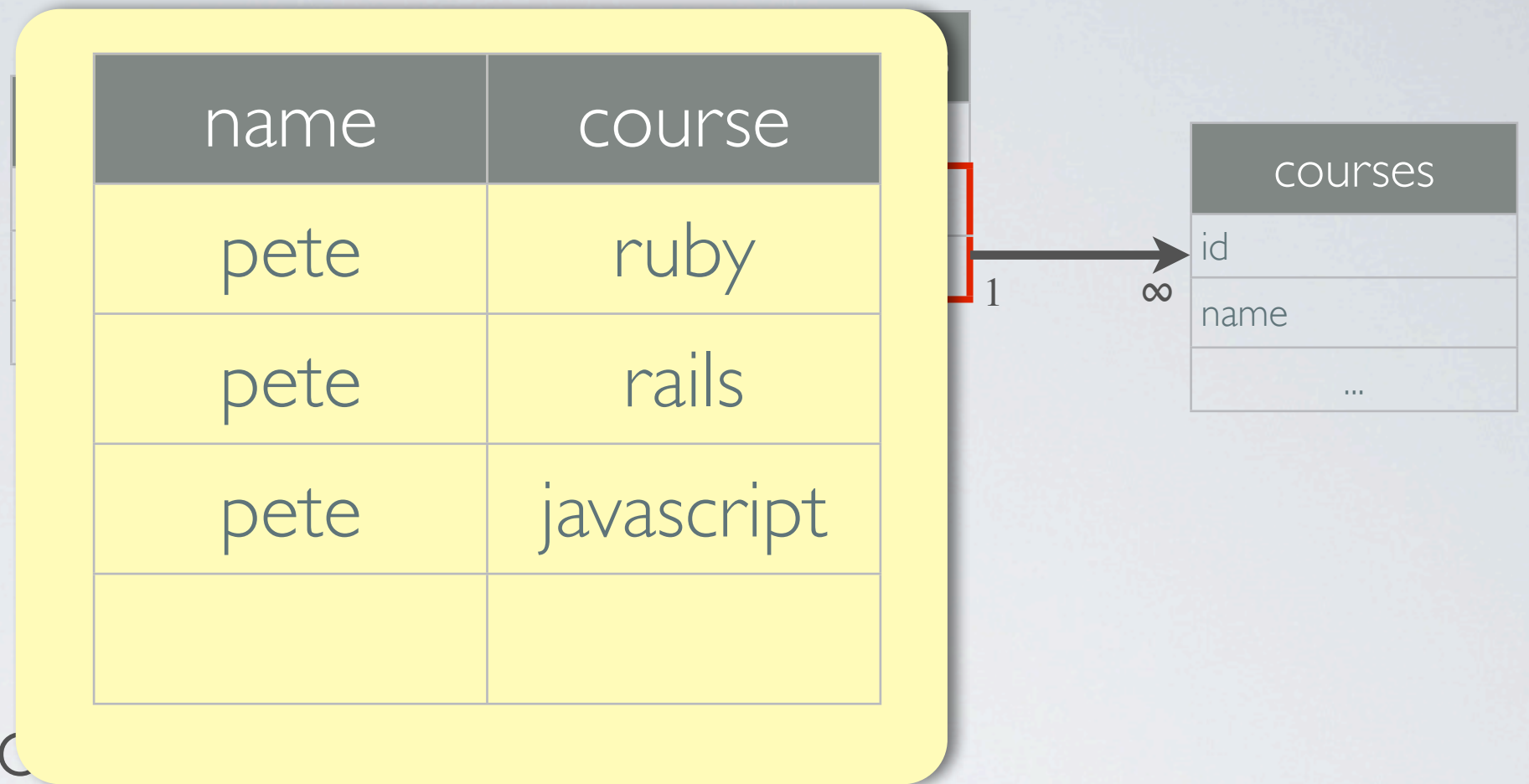


Get Pete's Courses

```
SELECT students.name, courses.name
FROM students
JOIN student_courses
  ON students.id = student_courses.student_id
JOIN courses
  ON student_courses.course_id = courses.id
WHERE students.id = 1
```



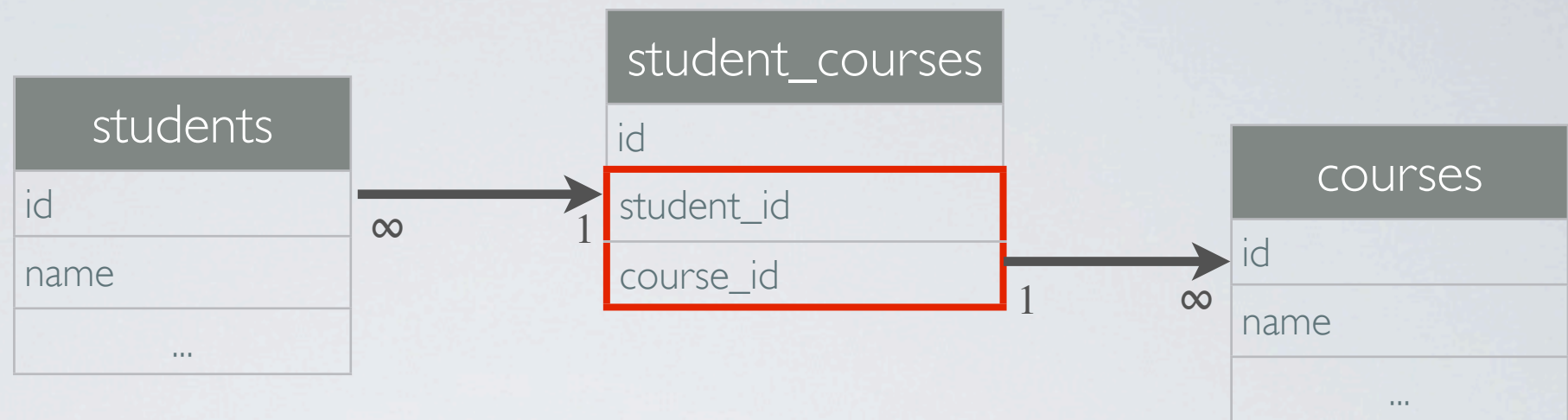
many to many



Get Pete's Co

```
SELECT students.name, courses.name
FROM students
JOIN student_courses
  ON students.id = student_courses.student_id
JOIN courses
  ON student_courses.course_id = courses.id
WHERE students.id = 1
```

many to many

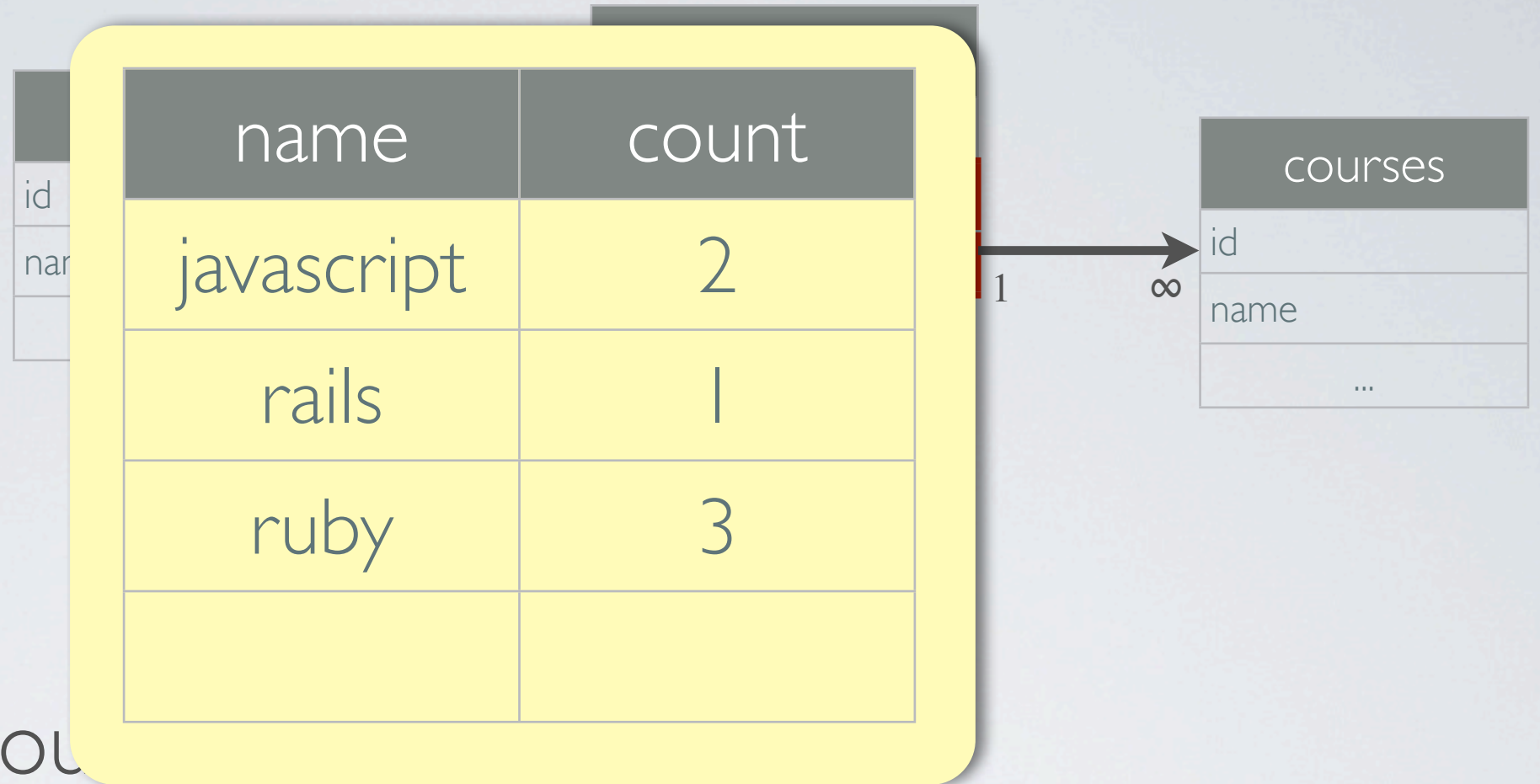


Get Pete's Courses

```
SELECT students.name, courses.name
FROM students
JOIN student_courses
  ON students.id = student_courses.student_id
JOIN courses
  ON student_courses.course_id = courses.id
```



many to many



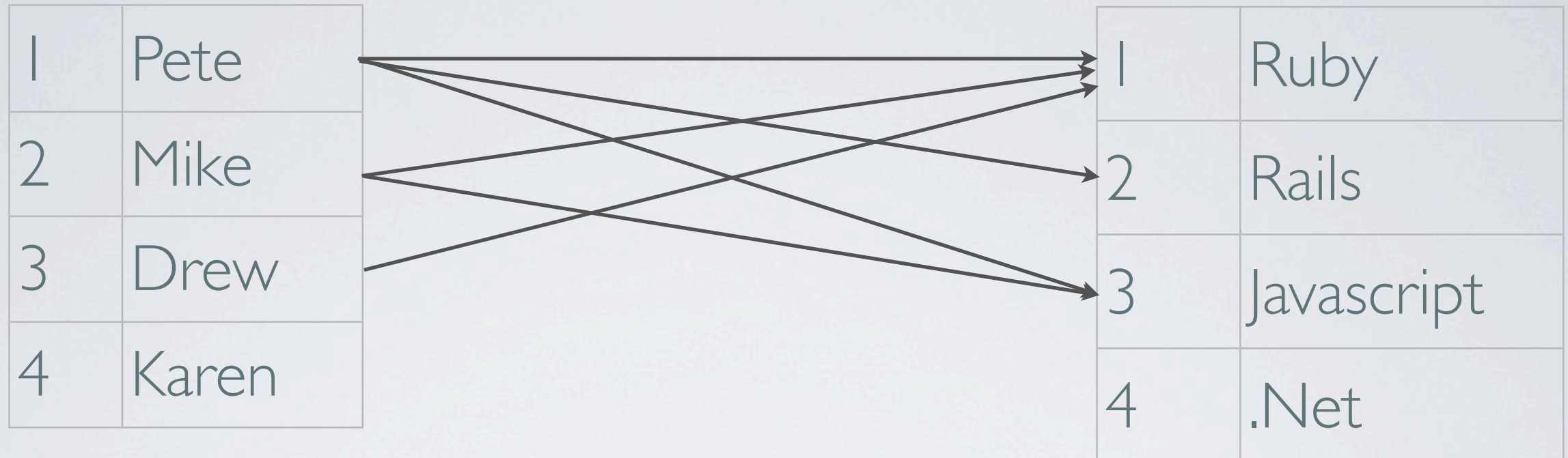
Get Pete's Courses

```
SELECT students.name, courses.name
FROM students
JOIN student_courses
  ON students.id = student_courses.student_id
JOIN courses
  ON student_courses.course_id = courses.id
```

Students

StudentCourses

Courses



## How many students in each course

```
SELECT courses.name, count(students.name)
FROM courses
LEFT JOIN student_courses
ON student_courses.course_id = courses.id
LEFT JOIN students
ON student_courses.student_id = students.id
GROUP BY courses.name
```



Students

1	Pete
2	Mike
3	Drew
4	Karen

name	count
javascript	2
rails	1
ruby	3
.net	0

Courses

1	Ruby
2	Rails
3	Javascript
4	.Net

## How many students in each course

```
SELECT courses.name, count(students.name)
FROM courses
LEFT JOIN student_courses
  ON student_courses.course_id = courses.id
LEFT JOIN students
  ON student_courses.student_id = students.id
GROUP BY courses.name
```

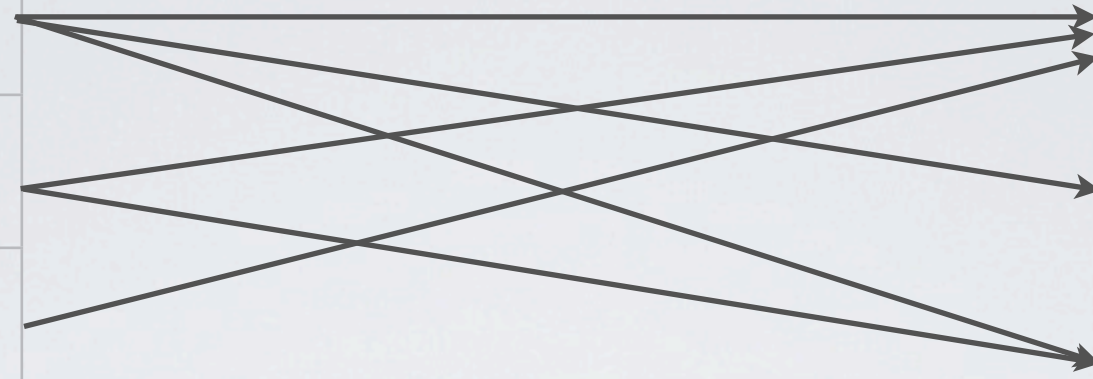
Students

1	Pete
2	Mike
3	Drew
4	Karen

StudentCourses

Courses

1	Ruby
2	Rails
3	Javascript
4	.Net



## LEFT JOIN

```
SELECT courses.name, students.name  
FROM courses  
LEFT JOIN student_courses  
ON student_courses.course_id = courses.id  
LEFT JOIN students  
ON student_courses.student_id = students.id
```



## Students

1	Pete
2	Mike
3	Drew
4	Karen

course

student

ruby

pete

ruby

mike

ruby

drew

rails

pete

javascript

pete

javascript

mike

.net

## Courses

1	Ruby
2	Rails
3	Javascript
4	.Net

## LEFT JOIN

```
SELECT courses.name, students.name
```

```
FROM courses
```

```
LEFT JOIN student_courses
```

```
ON student_courses.course_id = courses.id
```

```
LEFT JOIN students
```

```
ON student_courses.student_id = students.id
```

## Students

1	Pete
2	Mike
3	Drew
4	Karen

course	student
ruby	pete
ruby	mike
ruby	drew
rails	pete
javascript	pete
javascript	mike
	karen

## Courses

1	Ruby
2	Rails
3	Javascript
4	.Net

## RIGHT JOIN

```
SELECT courses.name, students.name  
FROM courses
```

```
RIGHT JOIN student_courses
```

```
ON student_courses.course_id = courses.id
```

```
RIGHT JOIN students
```

```
ON student_courses.student_id = students.id
```



## Students

1	Pete
2	Mike
3	Drew
4	Karen

## Courses

1	Ruby
2	Rails
3	Javascript
4	.Net

course	student
ruby	pete
ruby	mike
ruby	drew
rails	pete
javascript	pete
javascript	mike
.net	
	karen

## FULL JOIN

SELECT course  
FROM course

FULL JOIN student\_courses

ON student\_courses.course\_id = courses.id

FULL JOIN students

ON student\_courses.student\_id = students.id

