SQL

by Pete Brumm

TO FOLLOW ALONG

- 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

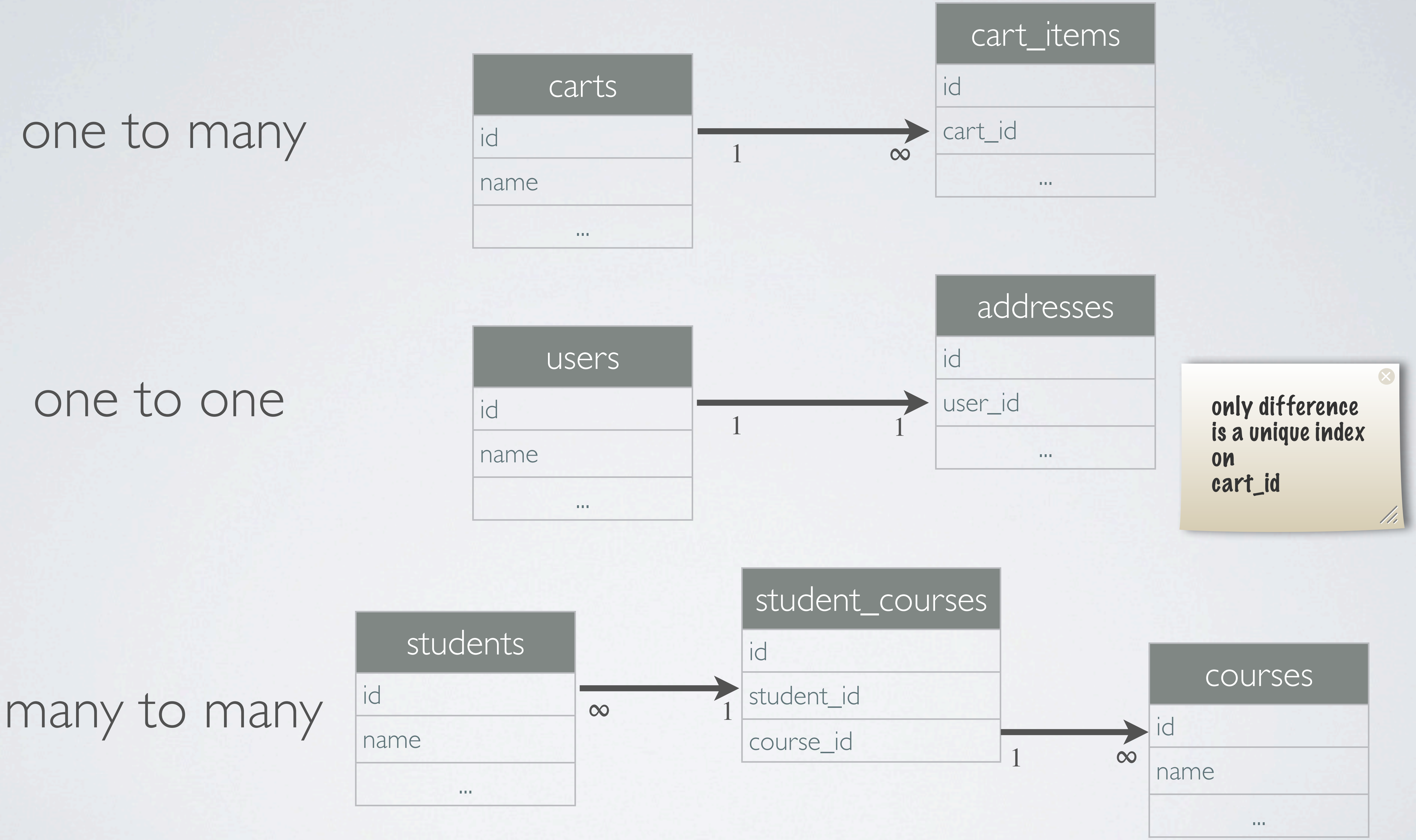
Open Source

- Postgres
- MySql
- Sqlite

COLUMN TYPES

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

DB RELATION TYPES



one to many

carts
id
name
...

1 ∞

cart_items
id
cart_id
...

one to one

users
id
name
...

1 1

addresses
id
user_id
...

only difference
is a unique index
on
cart_id

many to many

students
id
name
...

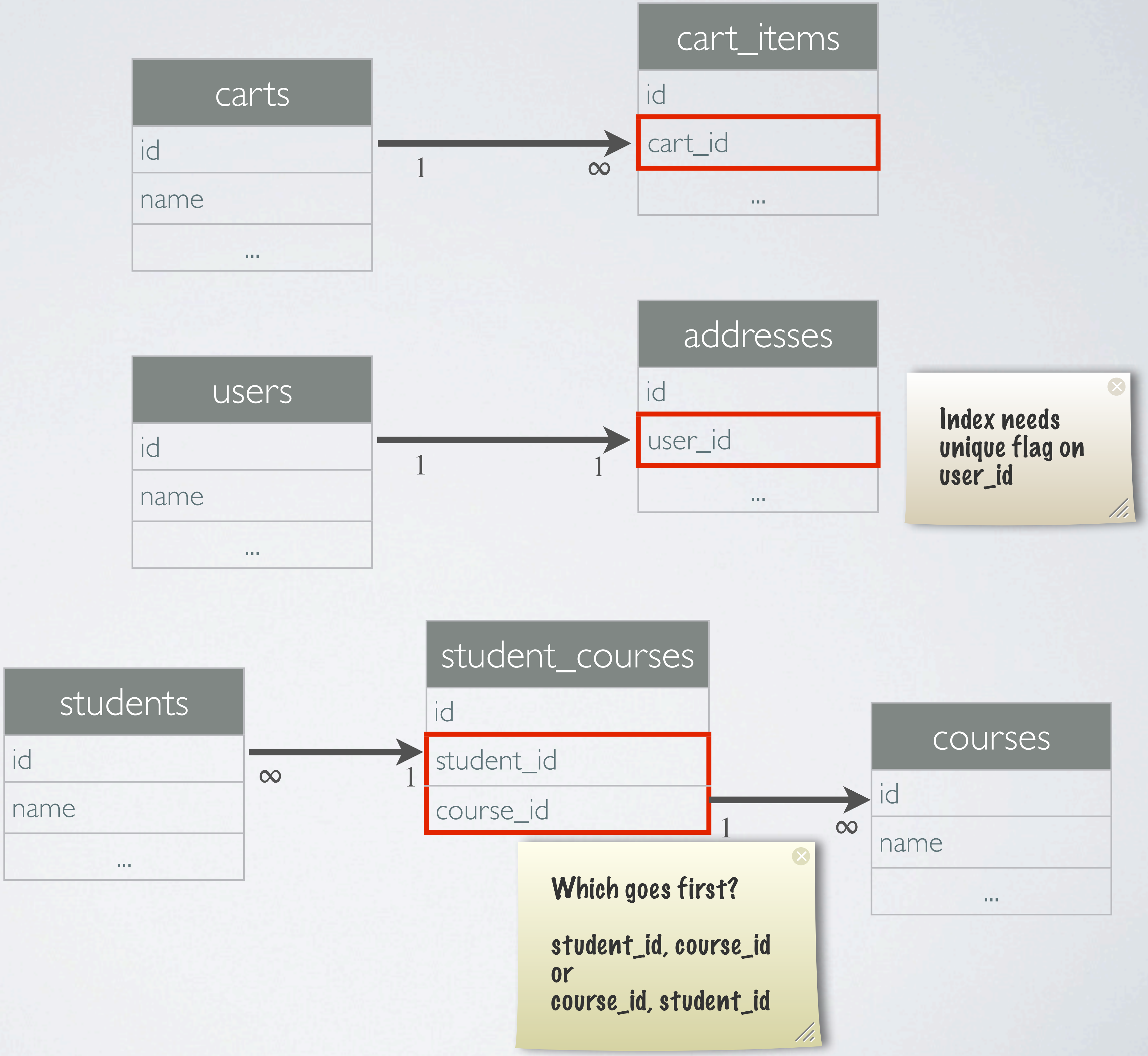
∞ 1

student_courses
id
student_id
course_id

1 ∞

courses
id
name
...

INDEXES



one to many

carts: id, name, …

cart_items: id, cart_id, …

1 → ∞

one to one

users: id, name, …

addresses: id, user_id, …

1 → 1

Index needs unique flag on user_id

many to many

students: id, name, …

student_courses: id, student_id, course_id

courses: id, name, …

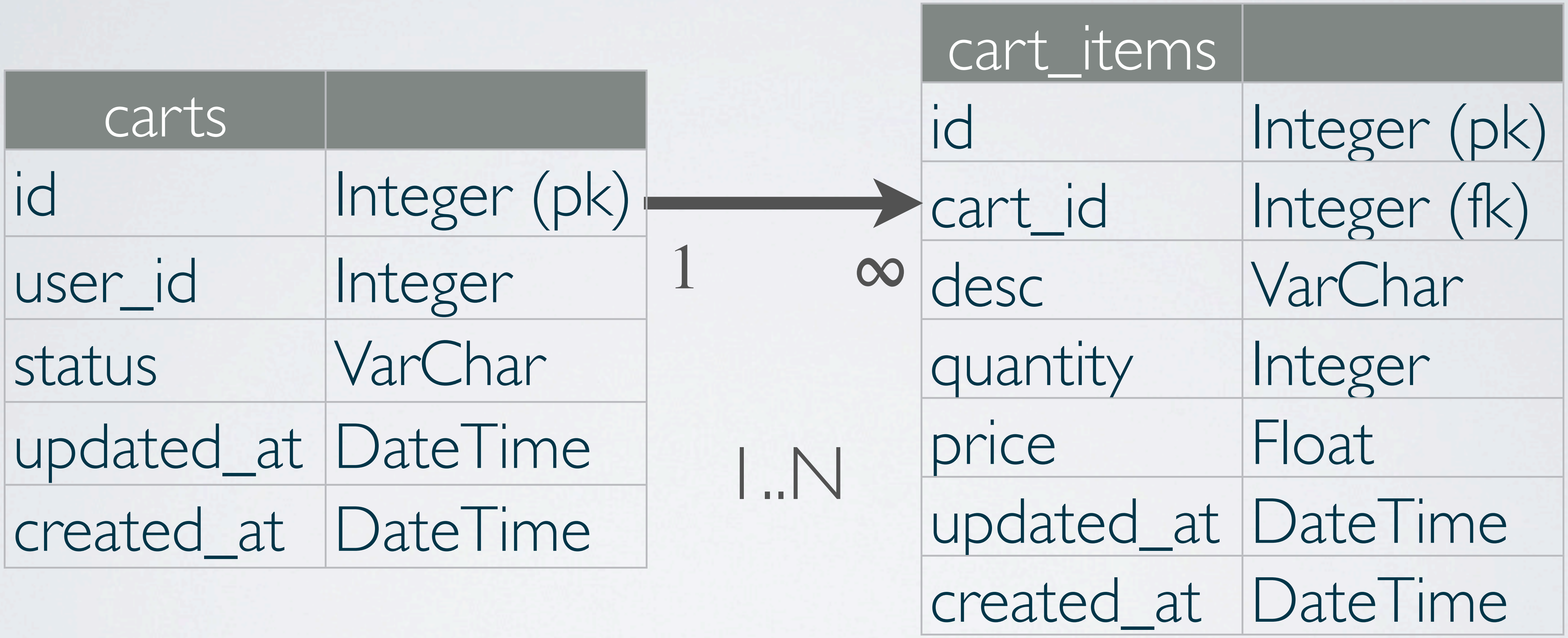
∞ → 1

1 → ∞

Which goes first?

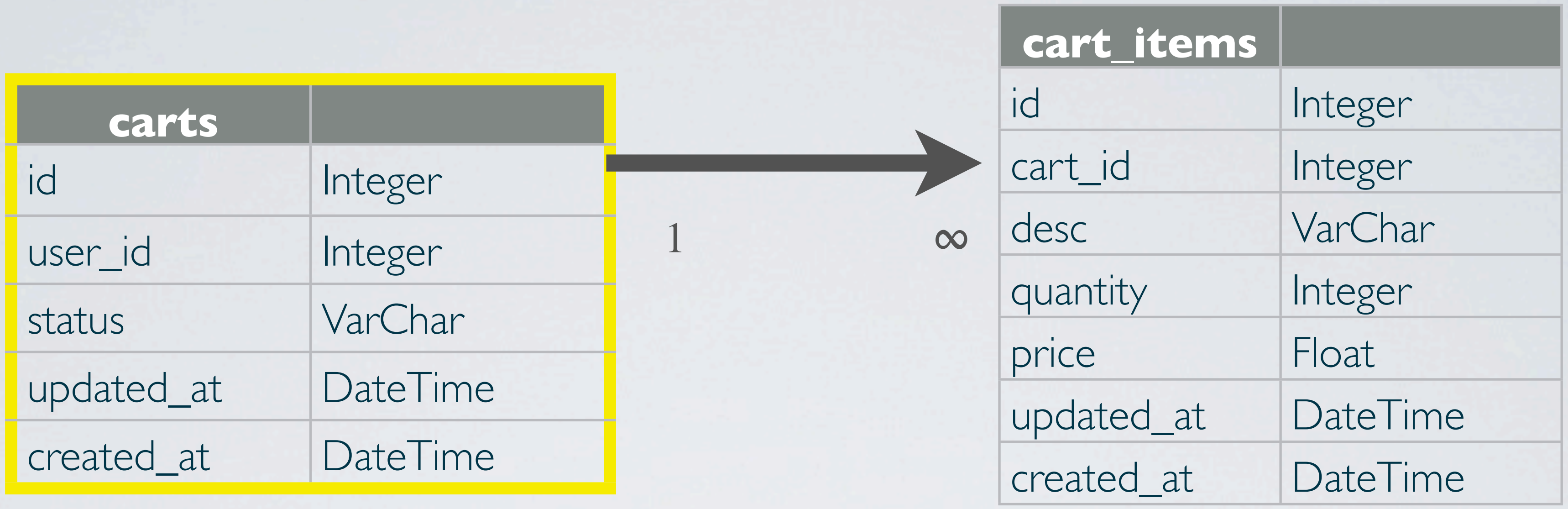
student_id, course_id
or
course_id, student_id

EXAMPLE



carts	
id	Integer (pk)
user_id	Integer
status	VarChar
updated_at	DateTime
created_at	DateTime

cart_items	
id	Integer (pk)
cart_id	Integer (fk)
desc	VarChar
quantity	Integer
price	Float
updated_at	DateTime
created_at	DateTime

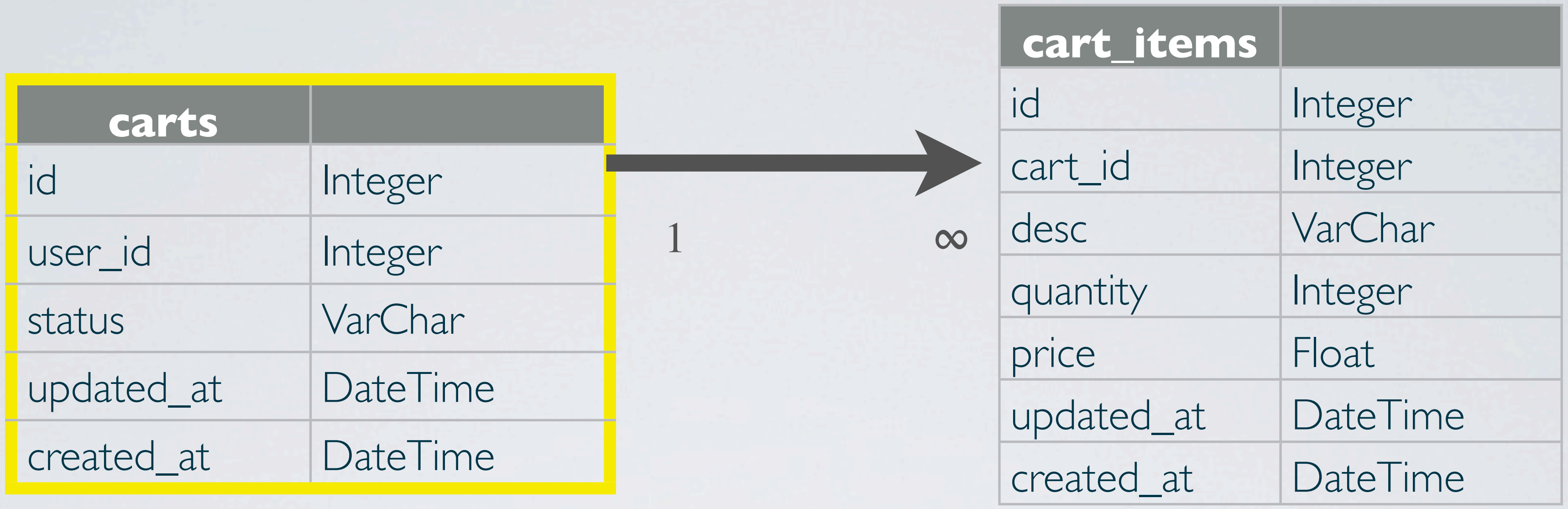


carts	
id	Integer
user_id	Integer
status	VarChar
updated_at	DateTime
created_at	DateTime

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

Lets add a cart

```
INSERT INTO carts
  (user_id, status, total, updated_at, created_at)
VALUES
  (1, 'open', 0, DATETIME('now'), DATETIME('now') )
```

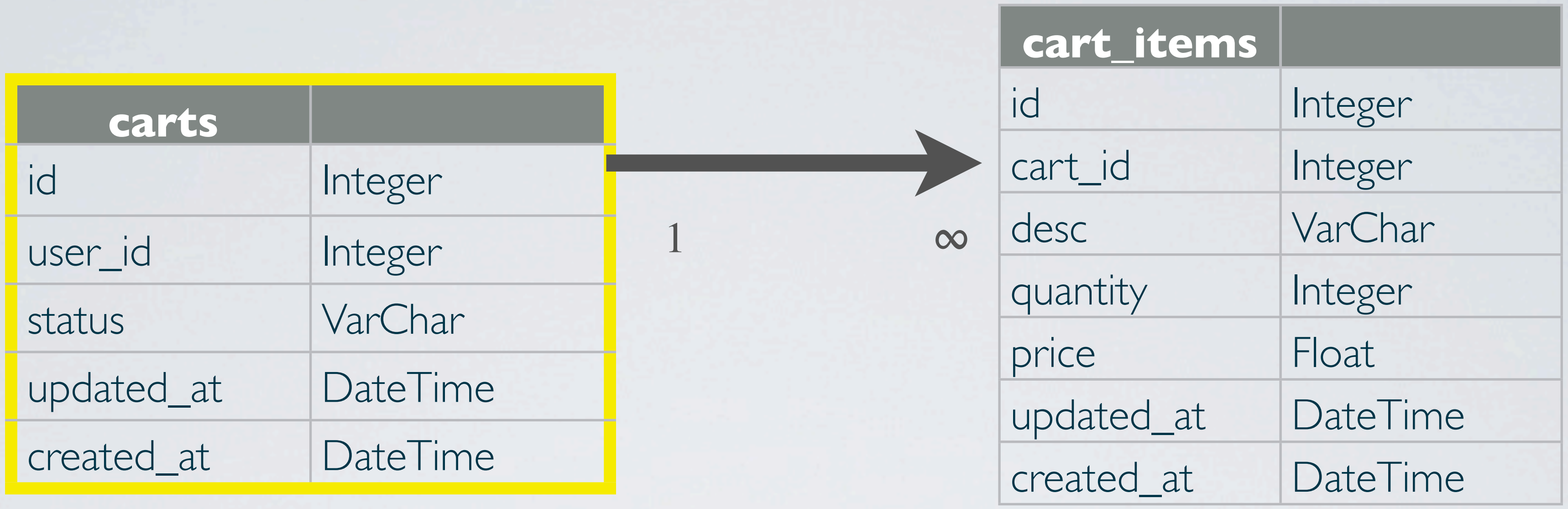


carts	
id	Integer
user_id	Integer
status	VarChar
updated_at	DateTime
created_at	DateTime

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

Lets add a cart

```
INSERT INTO carts
  (user_id, status, total, updated_at, created_at)
VALUES
  (1, 'open', 0, DATETIME('now'), DATETIME('now') )
```



carts	
id	Integer
user_id	Integer
status	VarChar
updated_at	DateTime
created_at	DateTime

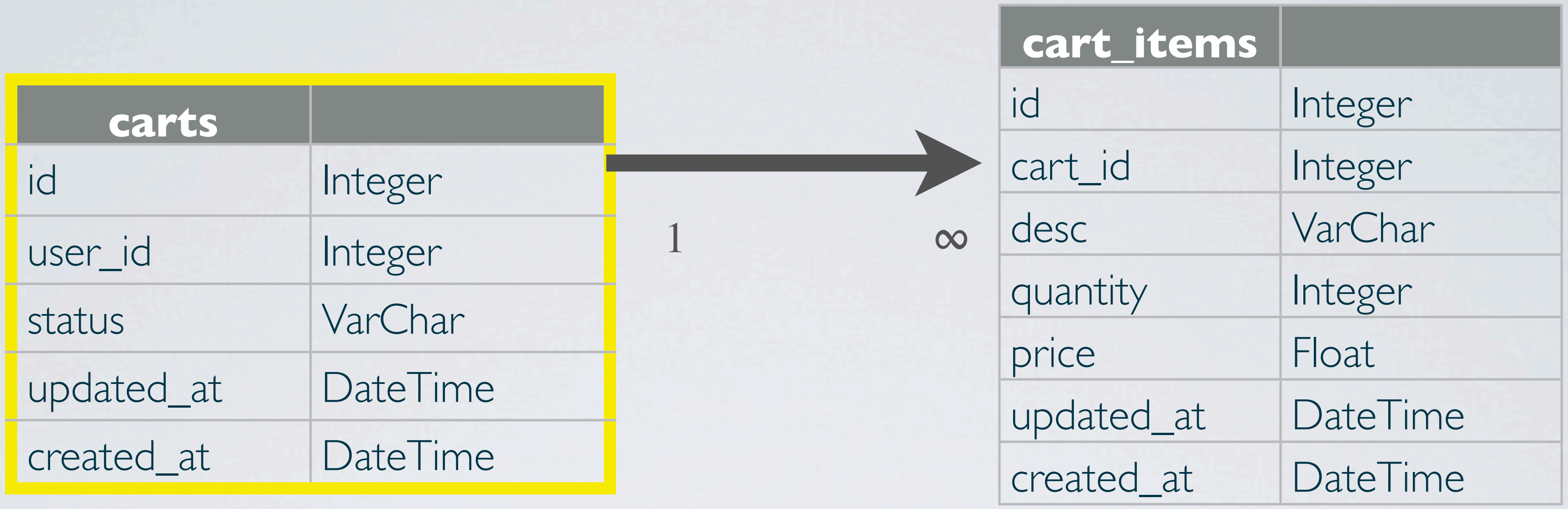
1

∞

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

Lets add a cart

```
INSERT INTO carts
  (user_id, status, total, updated_at, created_at)
VALUES
  (1, 'open', 0, DATETIME('now'), DATETIME('now') )
```



carts	
id	Integer
user_id	Integer
status	VarChar
updated_at	DateTime
created_at	DateTime

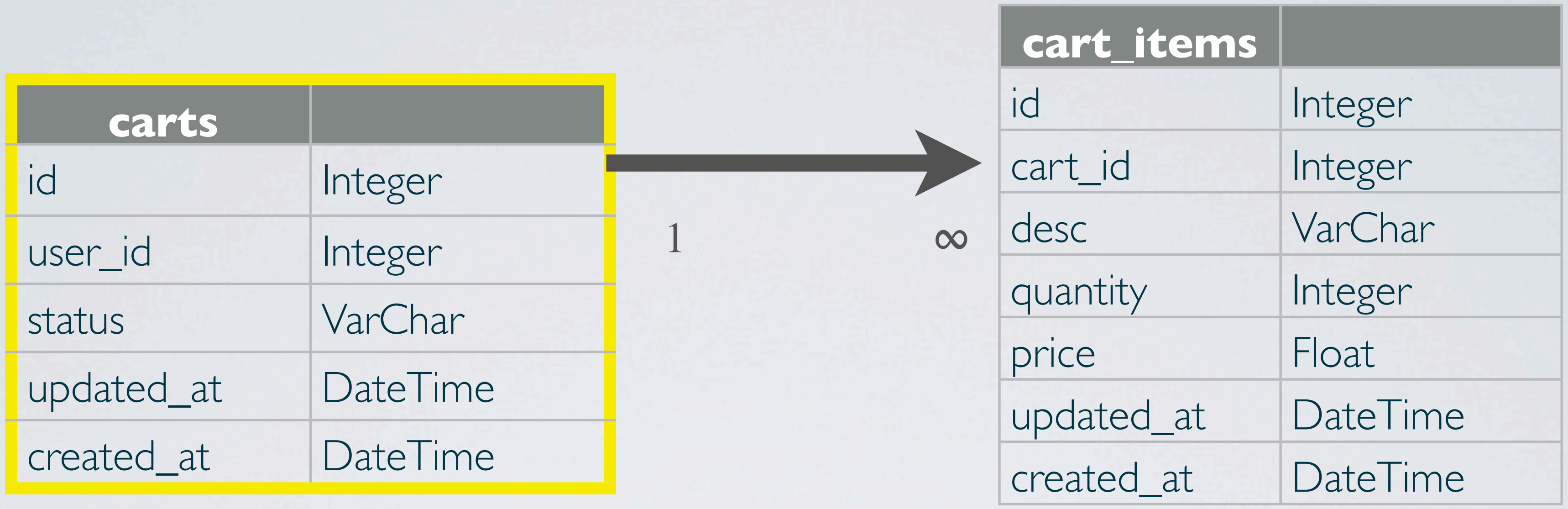
1

∞

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

Lets add a cart

```
INSERT INTO carts
  (user_id, status, total, updated_at, created_at)
VALUES
  (1, 'open', 0, DATETIME('now'), DATETIME('now') )
```

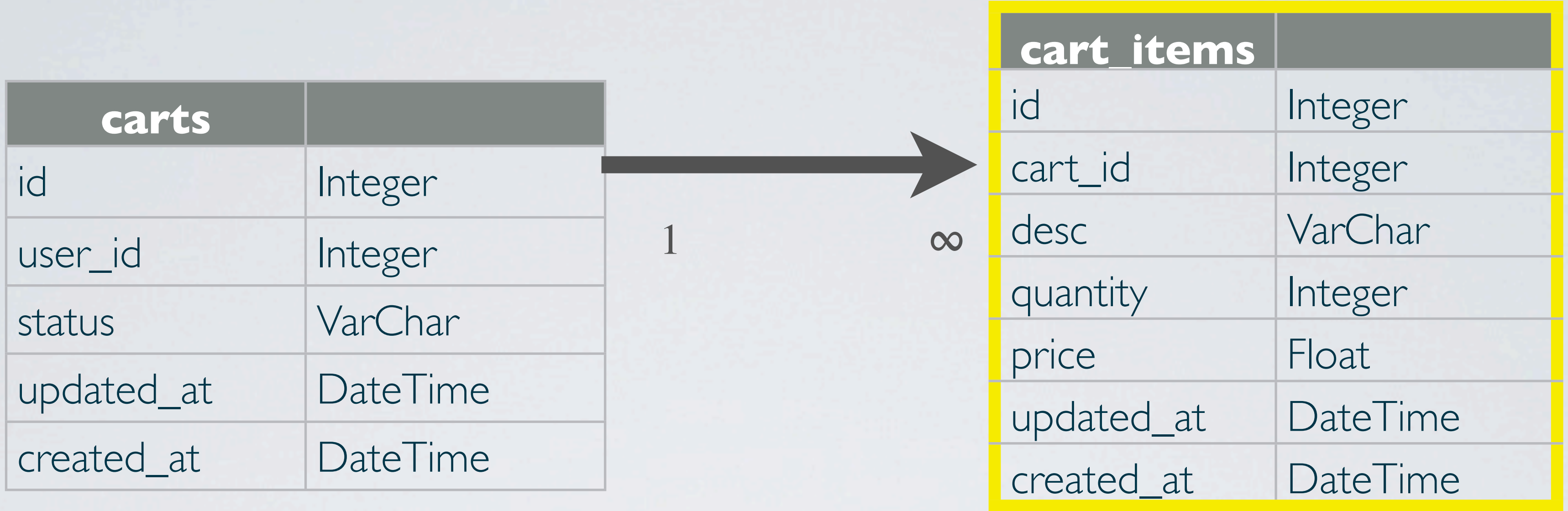


carts	
id	Integer
user_id	Integer
status	VarChar
updated_at	DateTime
created_at	DateTime

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

Lets add a cart

```
INSERT INTO carts
  (user_id, status, total, updated_at, created_at)
VALUES
  (1, 'open', 0, DATETIME('now'), DATETIME('now') )
```

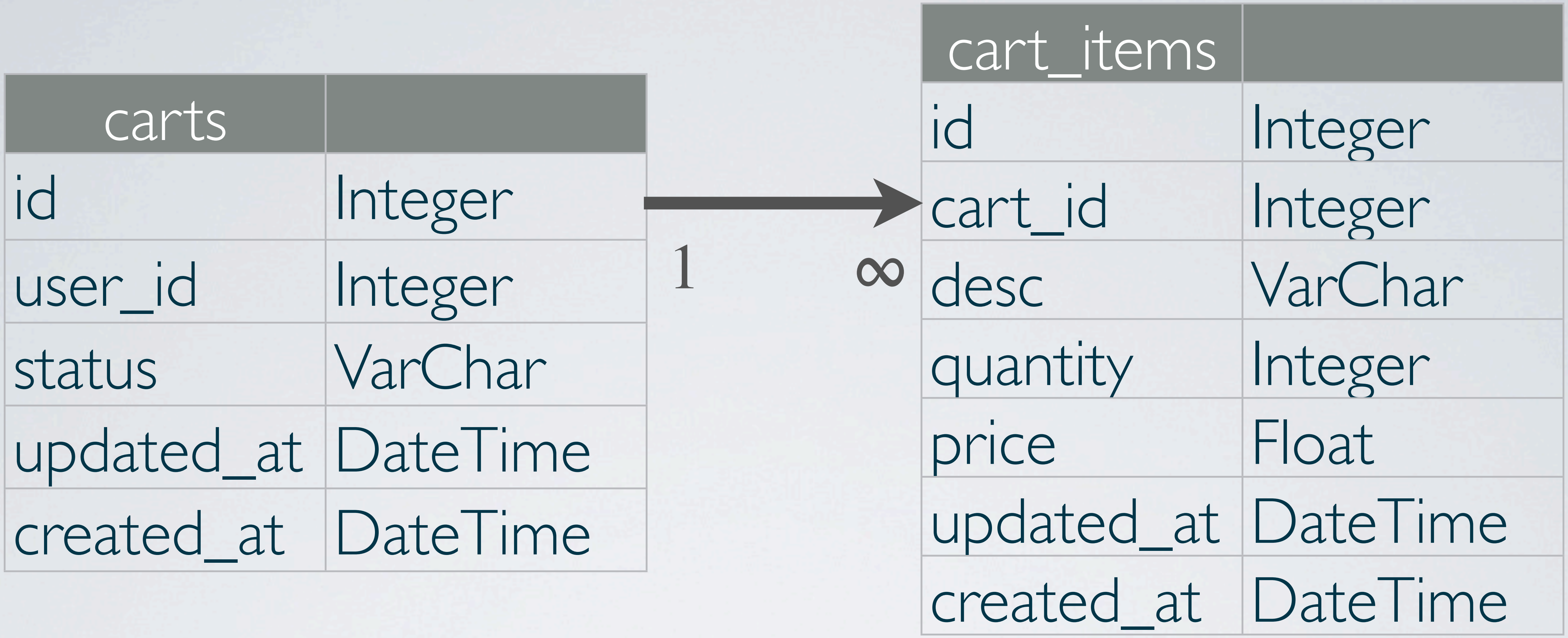


carts	
id	Integer
user_id	Integer
status	VarChar
updated_at	DateTime
created_at	DateTime

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

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') )
```

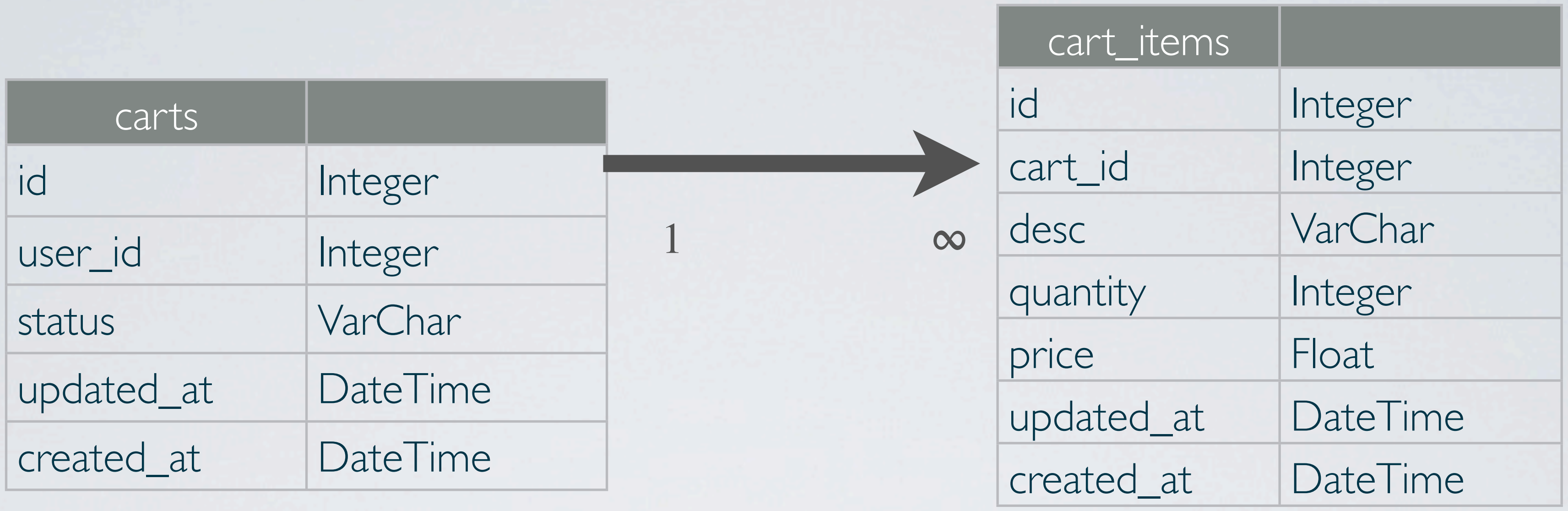


carts	
id	Integer
user_id	Integer
status	VarChar
updated_at	DateTime
created_at	DateTime

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"



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

Lets query some data

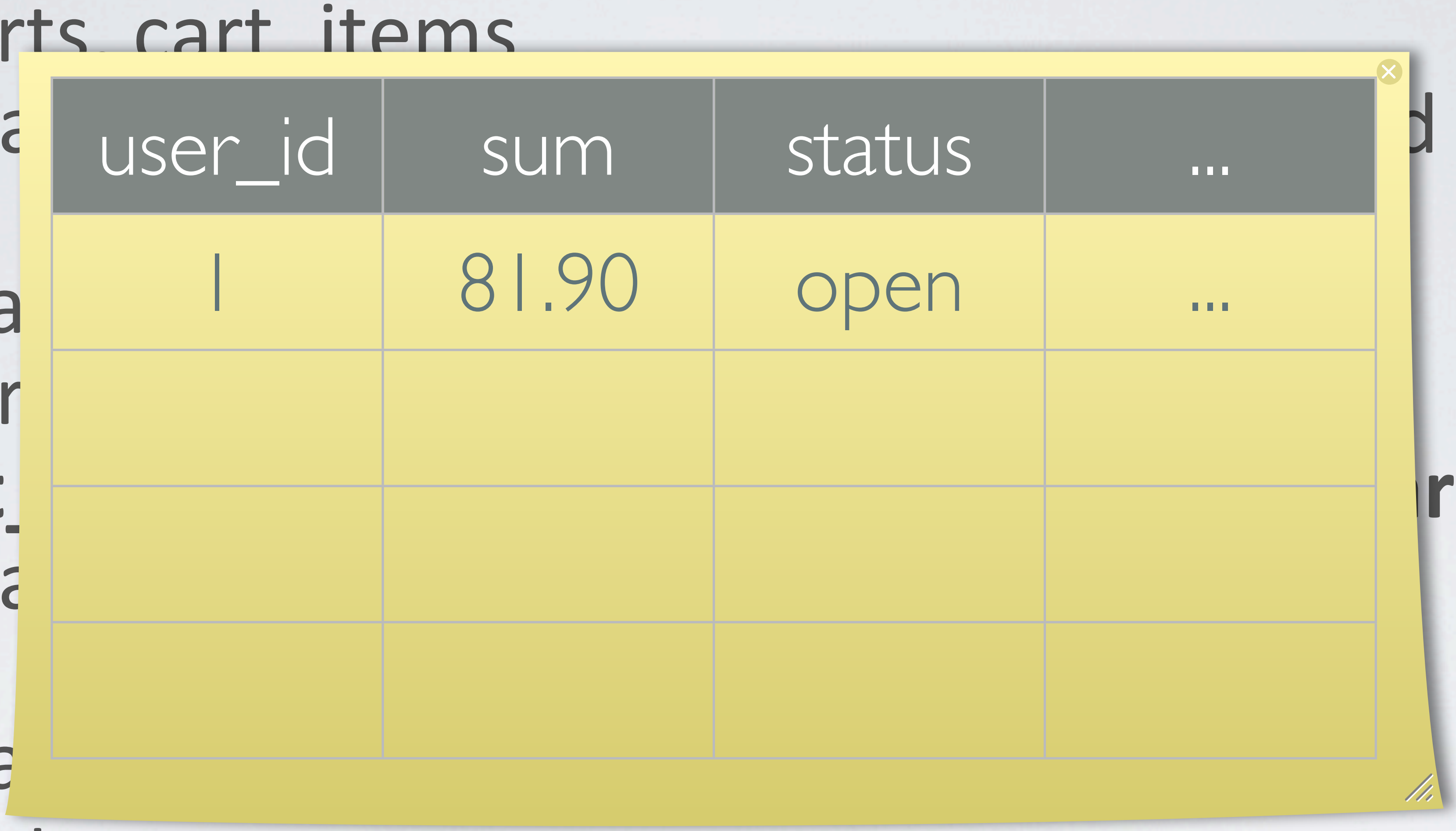
- Get all cart_items for a cart
  - **SELECT * FROM cart_items WHERE cart_id = 1**
- Remove an cart_item from a cart
  - **DELETE FROM cart_items WHERE id = 2**
- Find all carts that have a status of "open"
  - **SELECT * FROM carts WHERE status = 'open'**

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
```

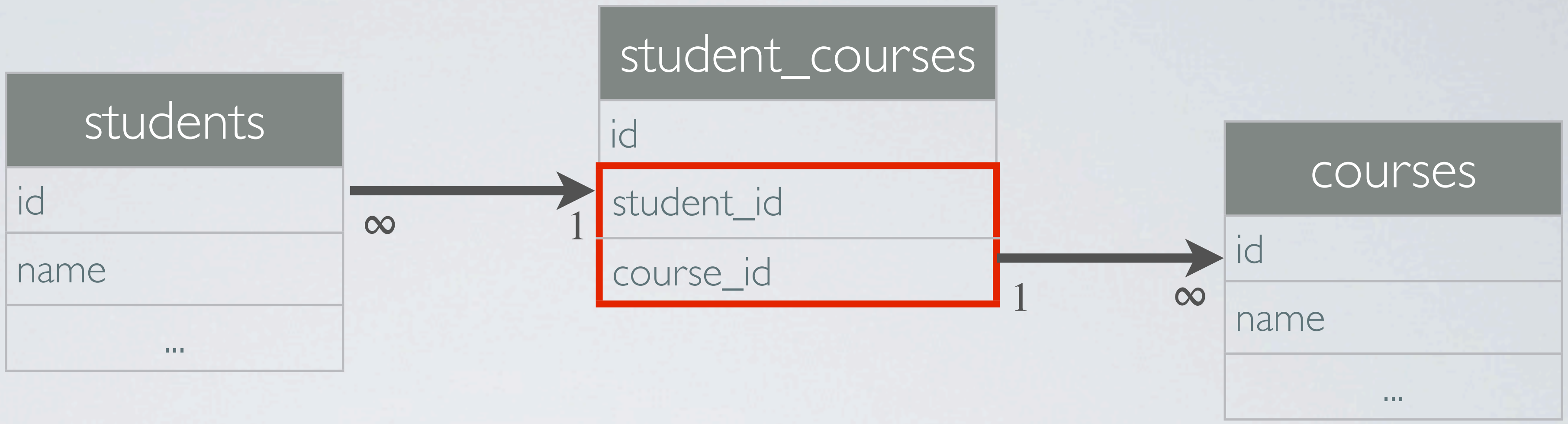
get cart with total price IMPLICIT JOIN, JOIN, INNER JOIN

```
SELECT carts.*, SUM(cart_items.price)
FROM carts, cart_items
WHERE ca                                    d = 1

SELECT ca
FROM car
JOIN cart_                                  rt_id
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	...

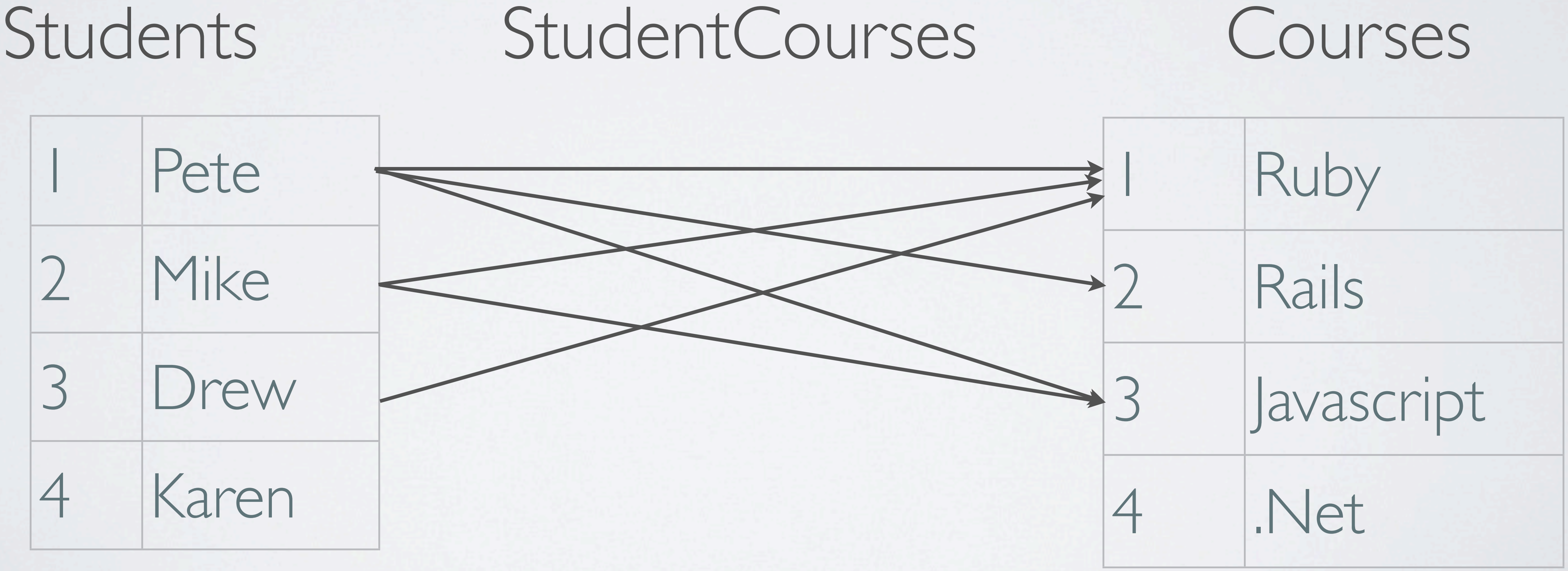
many to many



students
id
name
...

student_courses
id
student_id
course_id

courses
id
name
...



Students

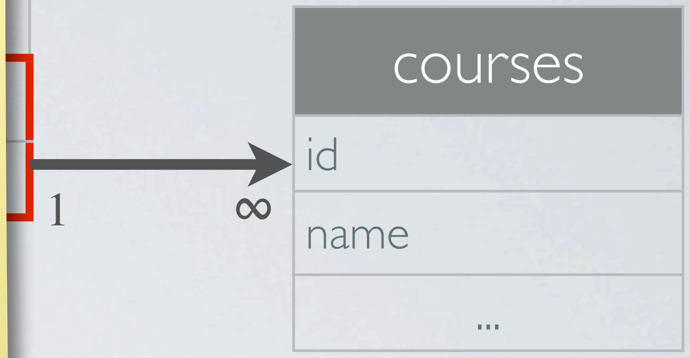
1	Pete
2	Mike
3	Drew
4	Karen

StudentCourses

Courses

1	Ruby
2	Rails
3	Javascript
4	.Net

many to many

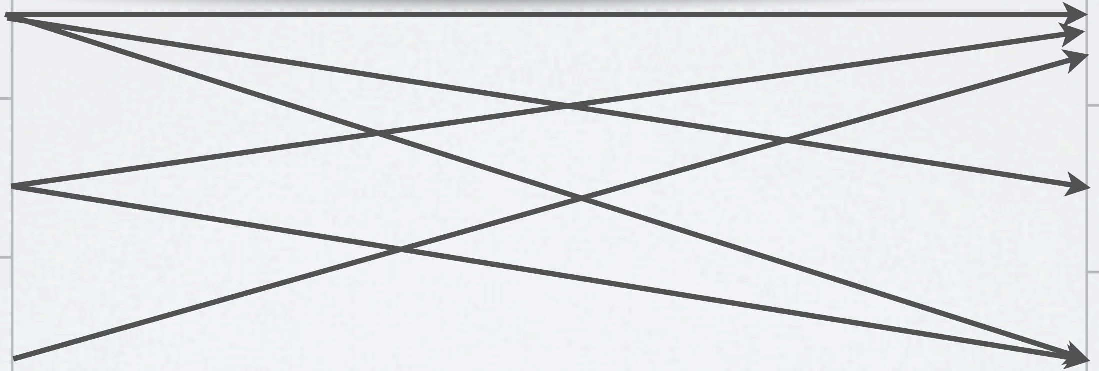


student_courses

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

Students

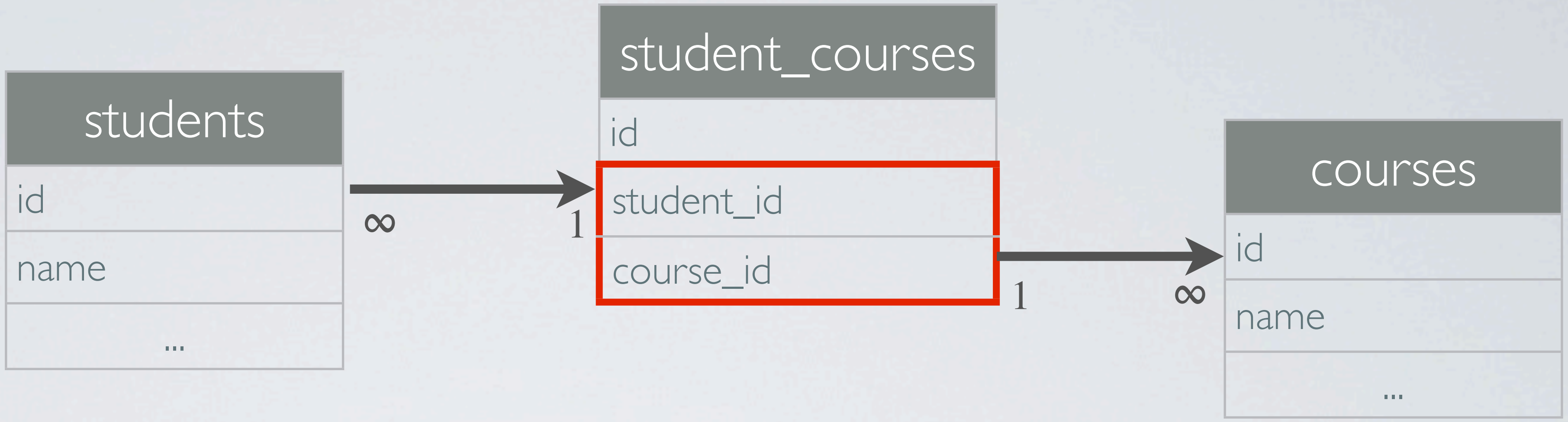
1	Pete
2	Mike
3	Drew
4	Karen



Courses

1	Ruby
2	Rails
3	Javascript
4	.Net

many to many



students
id
name
...

∞

1

student_courses
id
student_id
course_id

1

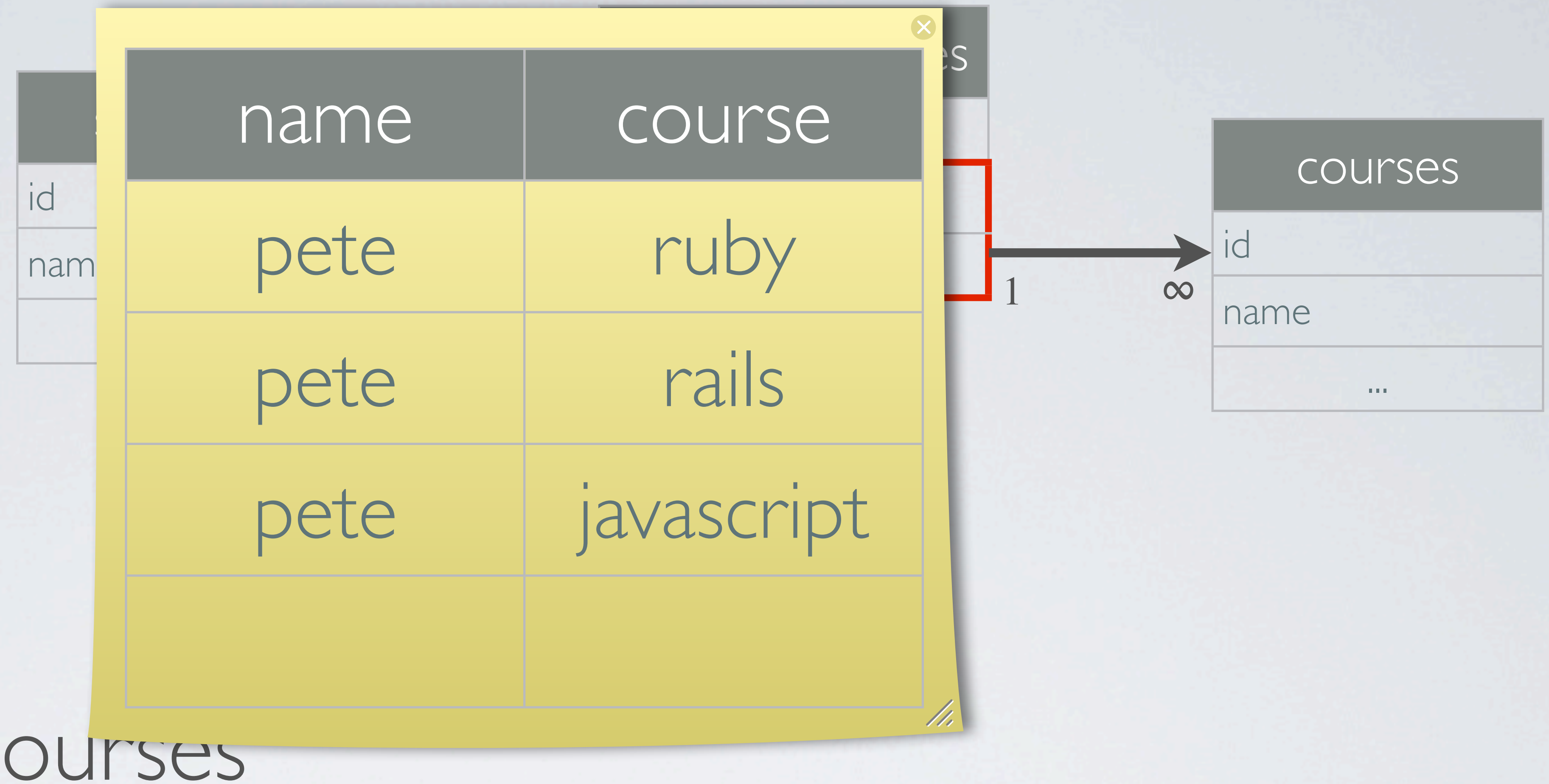
∞

courses
id
name
...

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

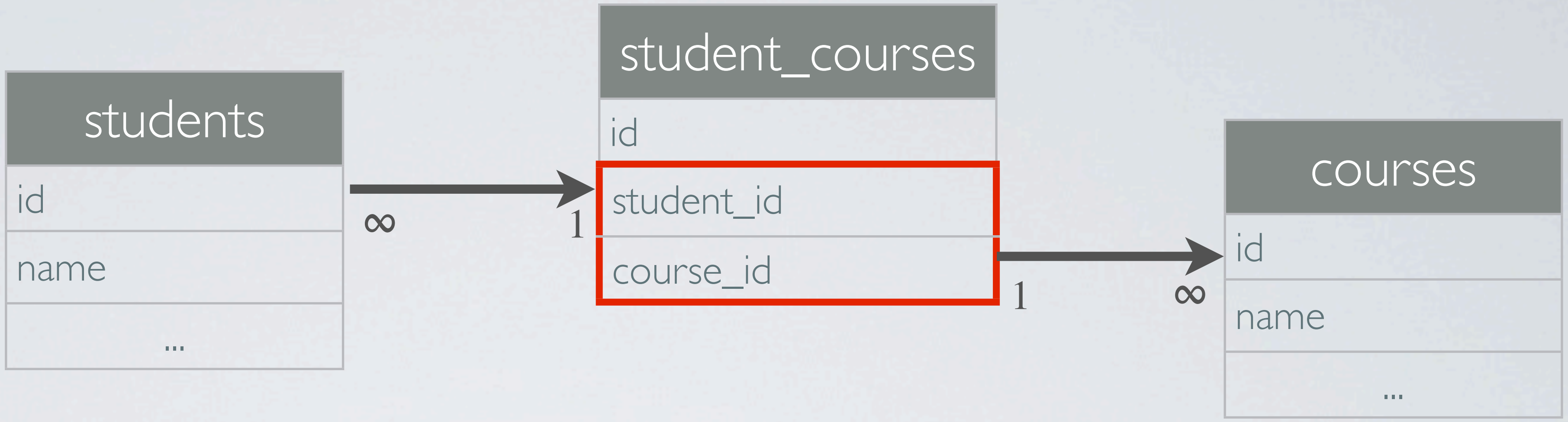


name	course
pete	ruby
pete	rails
pete	javascript

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



students
id
name
...

∞ 1

student_courses
id
student_id
course_id

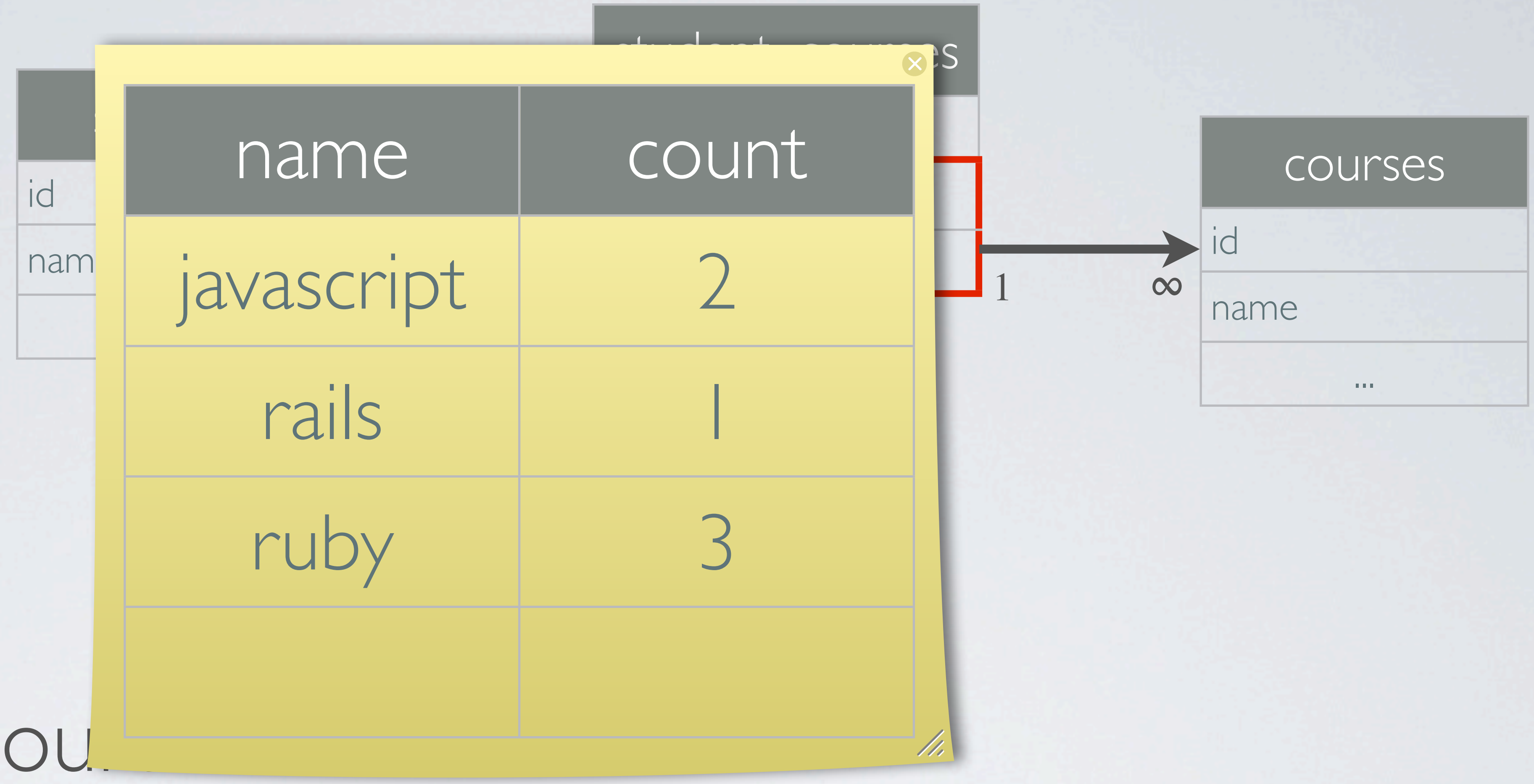
1 ∞

courses
id
name
...

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
```

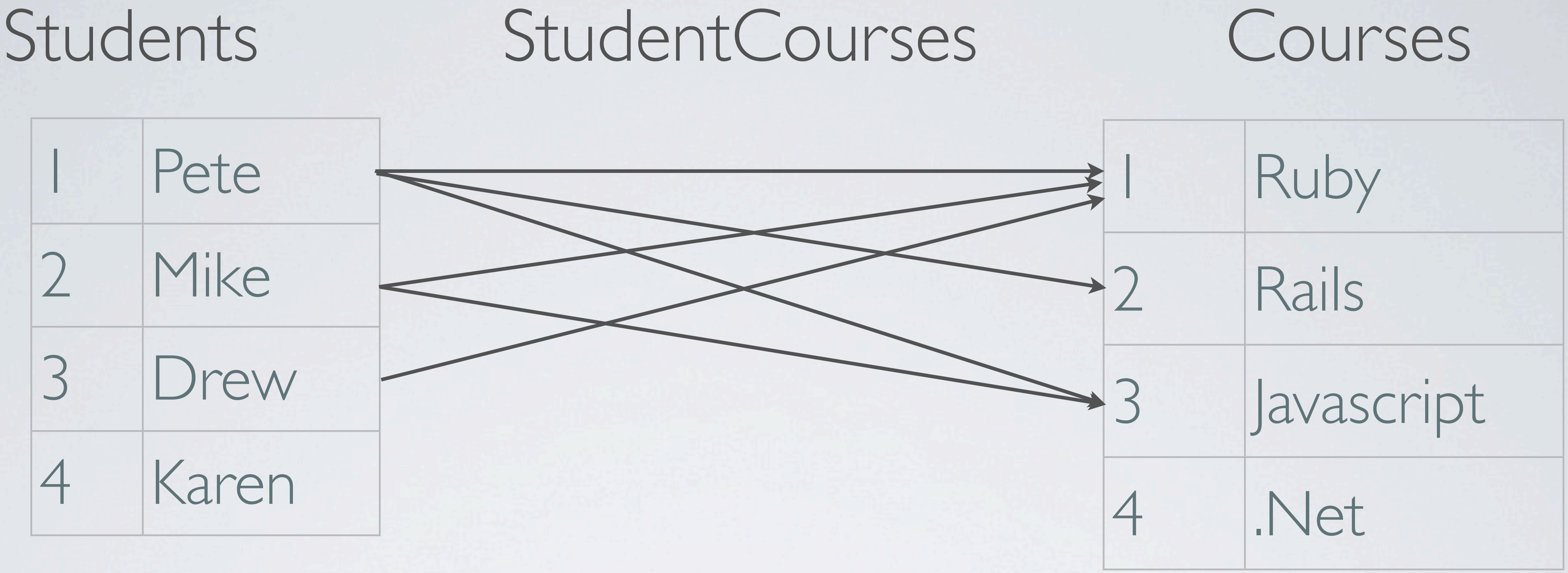
many to many



name	count
javascript	2
rails	1
ruby	3

Get Pete's Cou

```
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

1	Pete
2	Mike
3	Drew
4	Karen

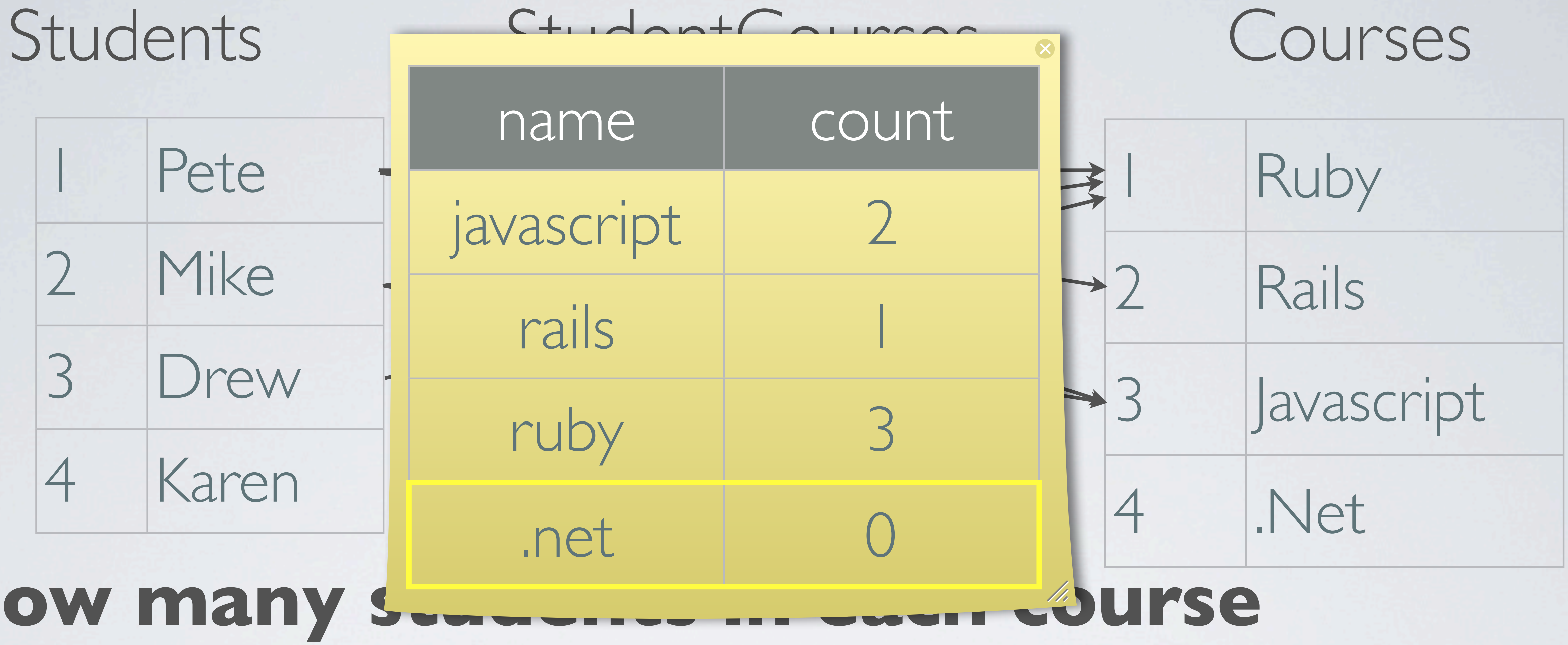
StudentCourses

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

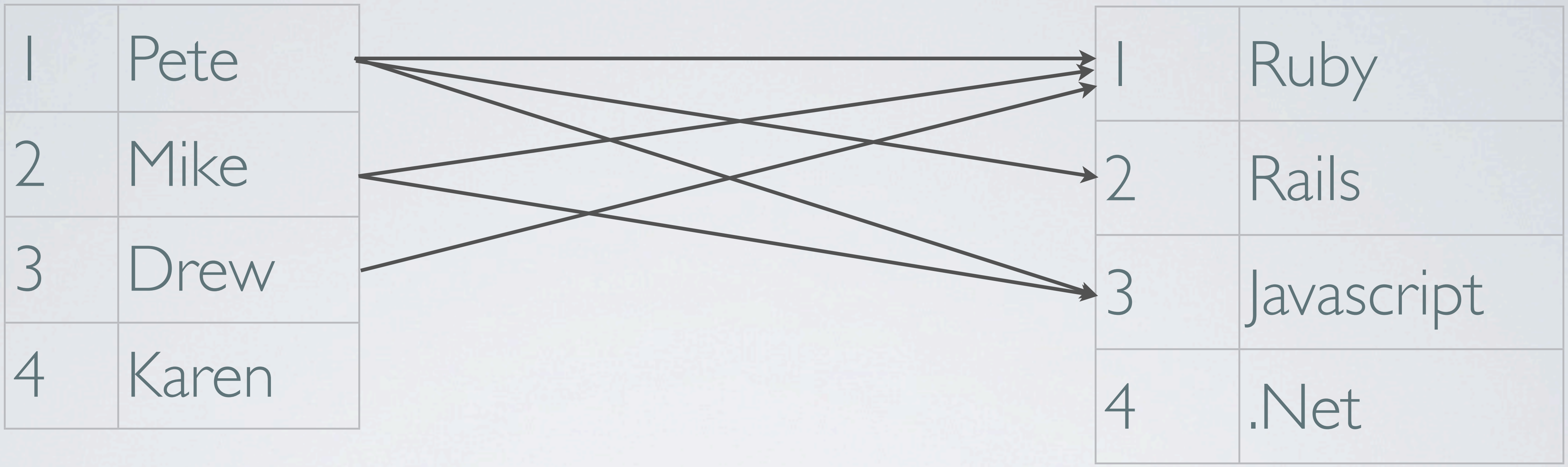
1	Ruby
2	Rails
3	Javascript
4	.Net

name	count
javascript	2
rails	1
ruby	3
.net	0

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 StudentCourses Courses



1	Pete
2	Mike
3	Drew
4	Karen

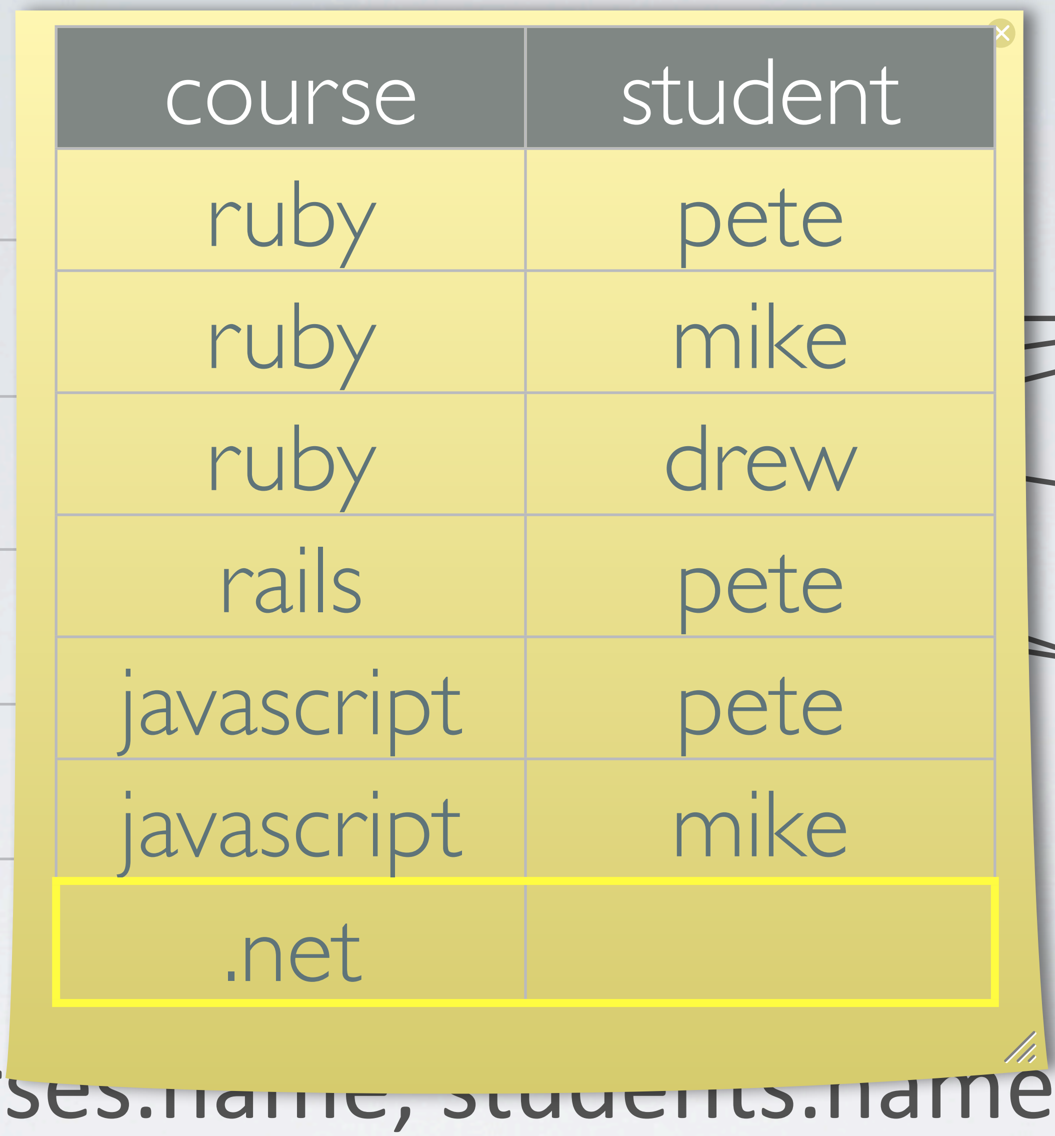
1	Ruby
2	Rails
3	Javascript
4	.Net

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

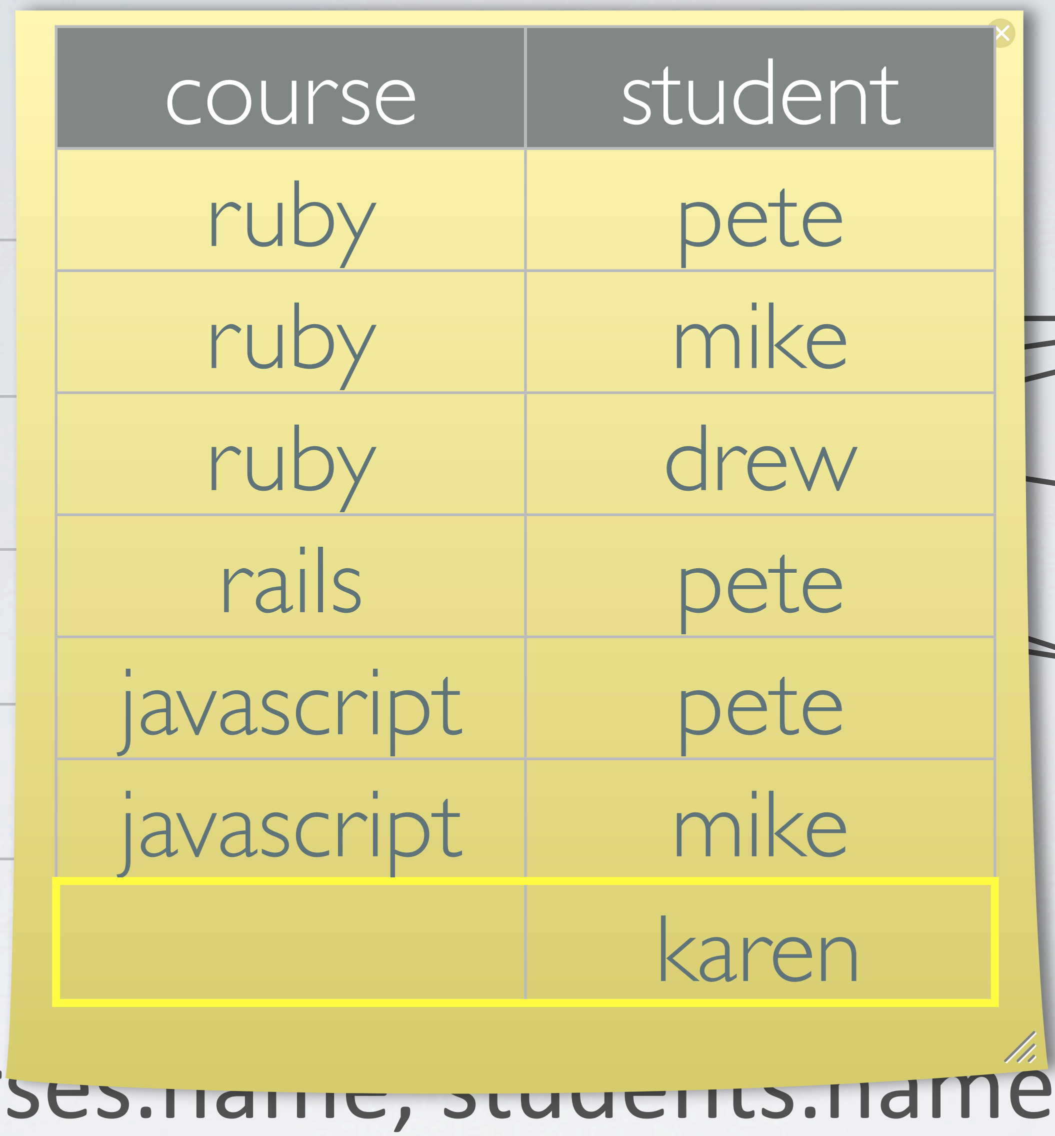
1	Ruby
2	Rails
3	Javascript
4	.Net

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

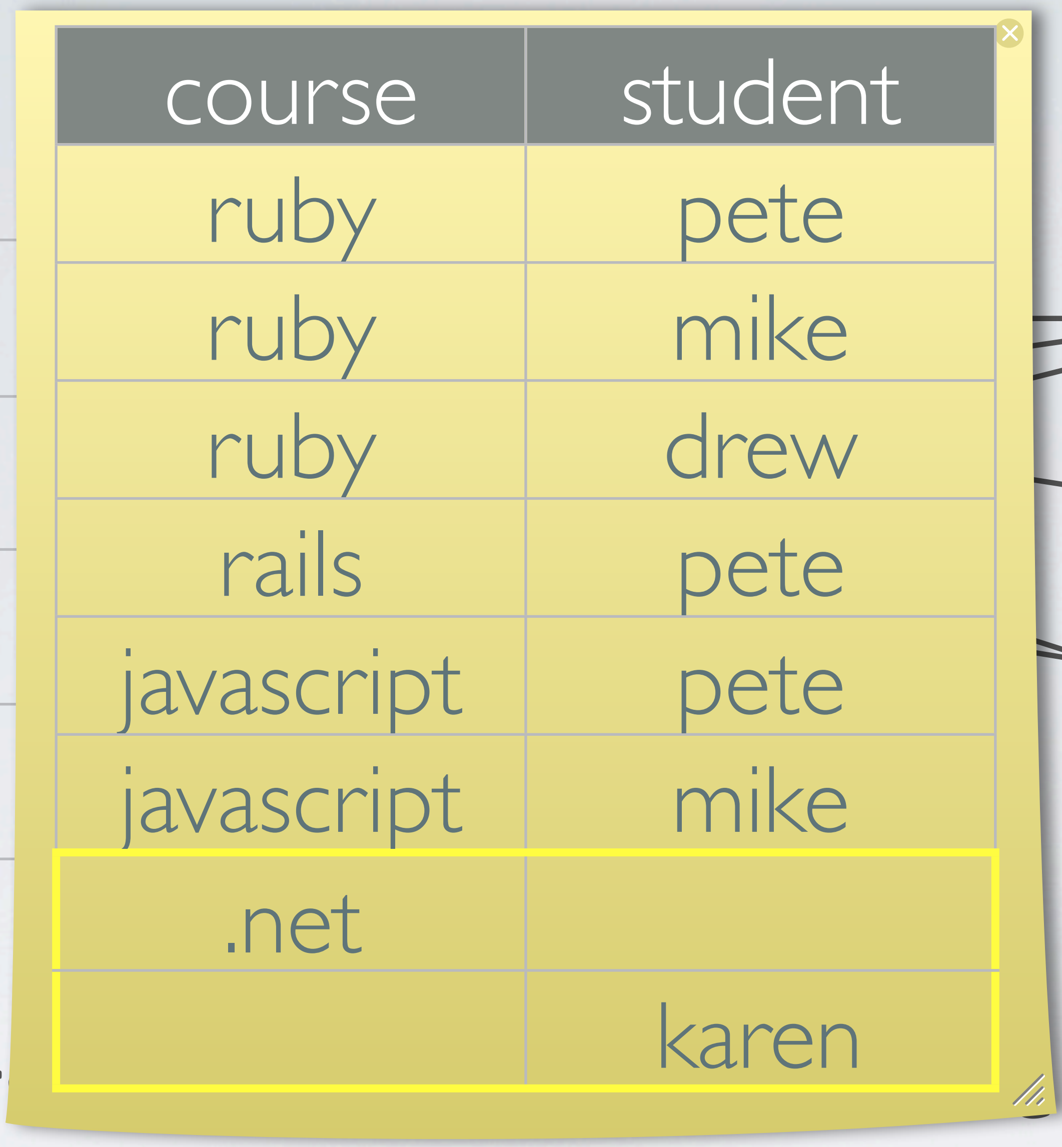
1	Ruby
2	Rails
3	Javascript
4	.Net

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



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

Courses

1	Ruby
2	Rails
3	Javascript
4	.Net

FULL JOIN

```
SELECT cour
 FROM courses
   FULL JOIN student_courses
    ON student_courses.course_id = courses.id
   FULL JOIN students
    ON student_courses.student_id = students.id
```