Intro to Backend Development

Lecture 3 · Relational Databases

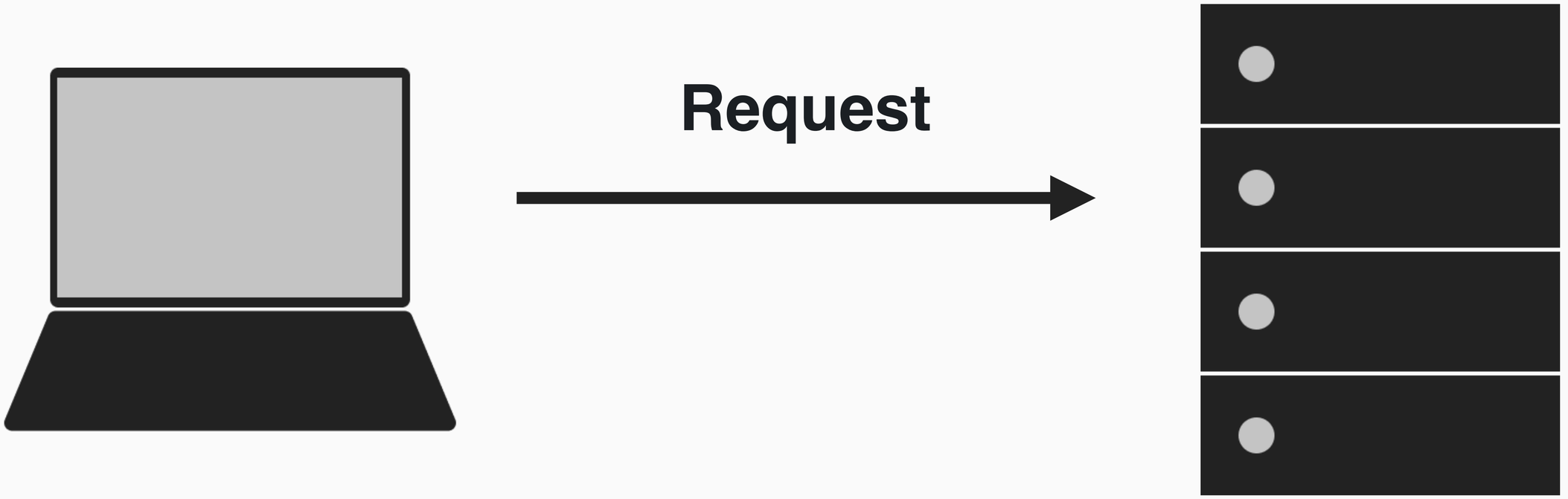
Kidus Zegeye
Tony Matchev

Announcements

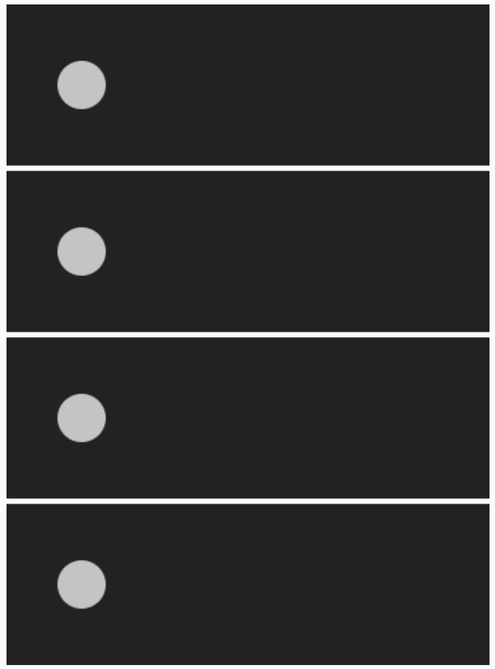
Announcements

- PA2 is due **this Wednesday (10/25)**

Review


Request
Client
Server

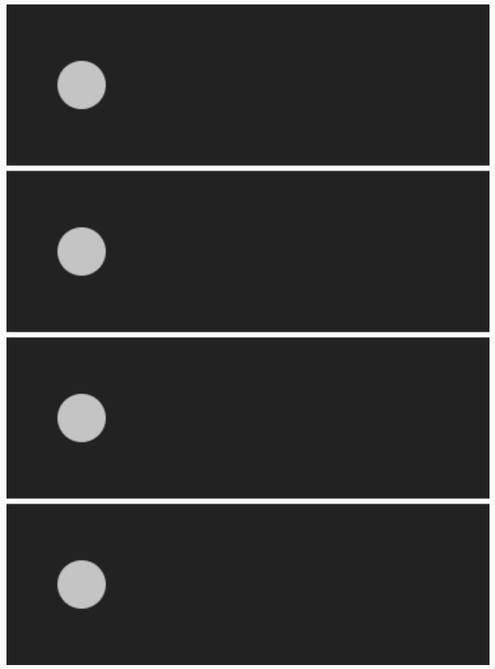
Request



Server

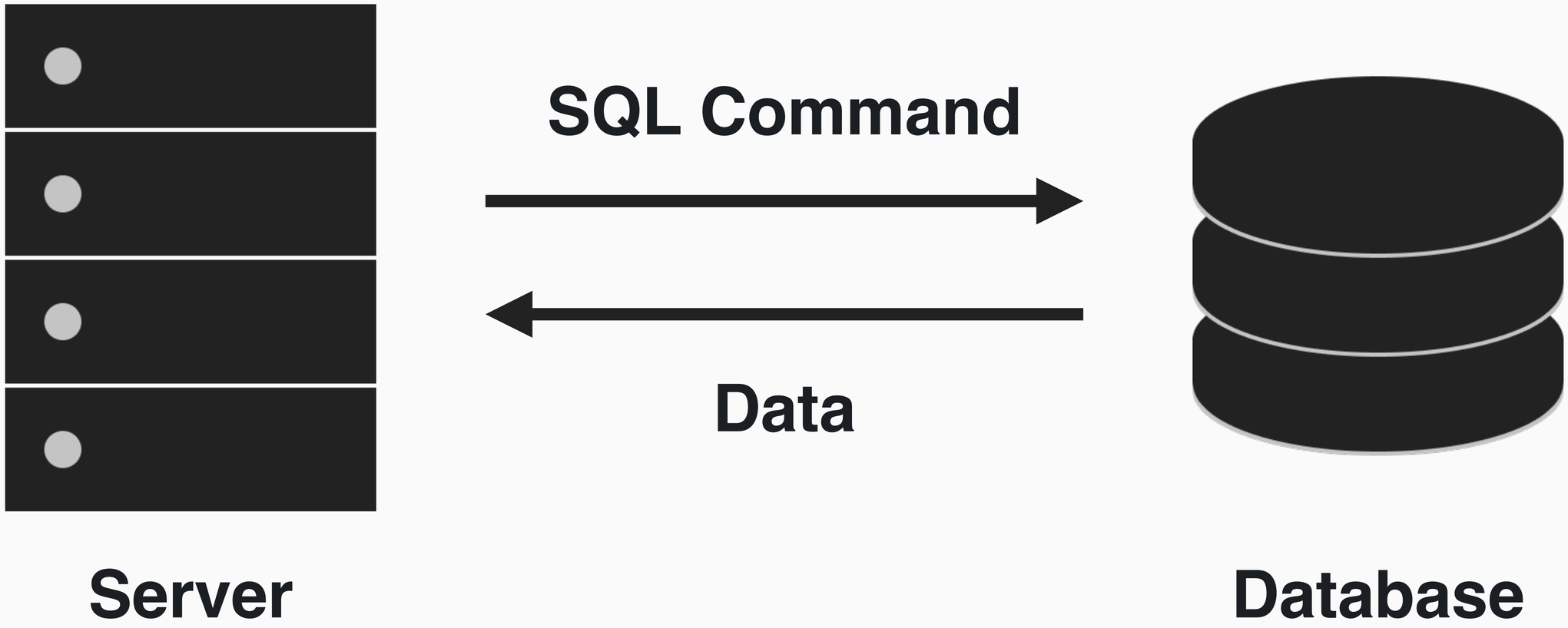
- Route triggered
- Data operation
- Return response

Request

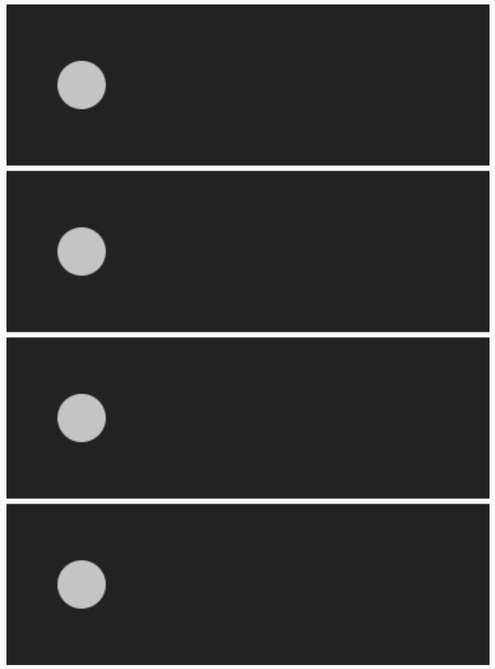


Server

- Route triggered
- Data operation
- Return response

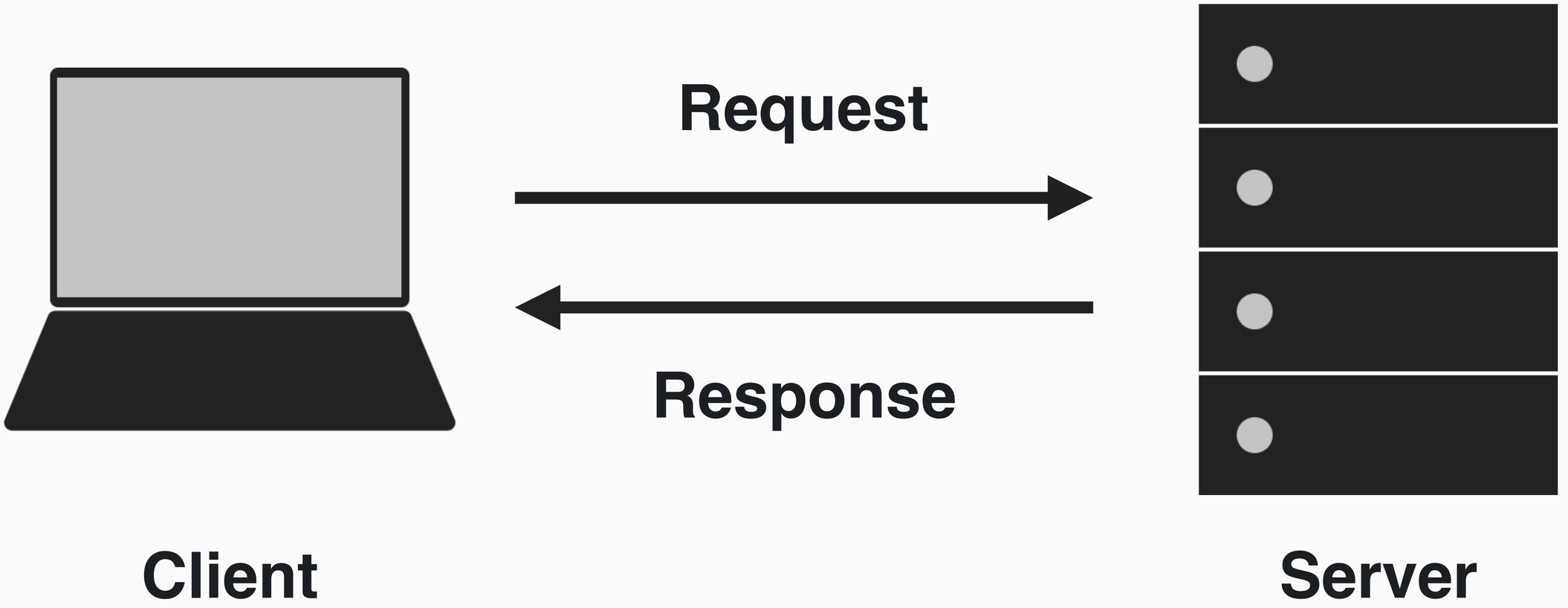

SQL Command
Data
Server
Database

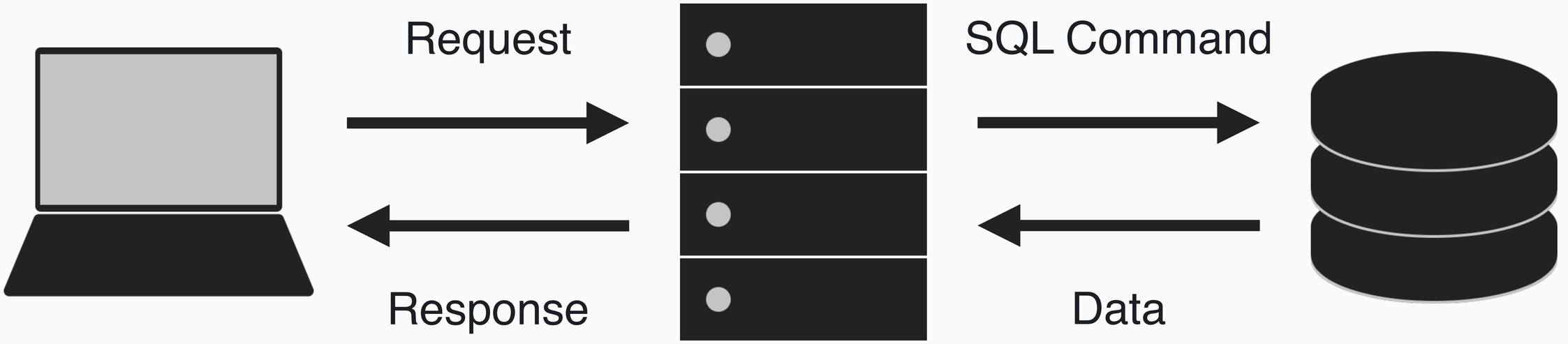
Request

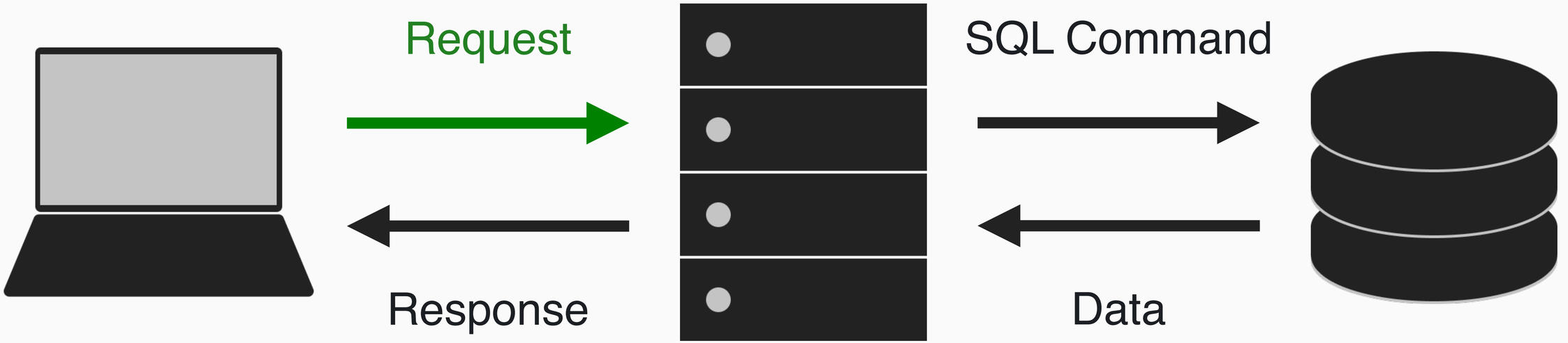


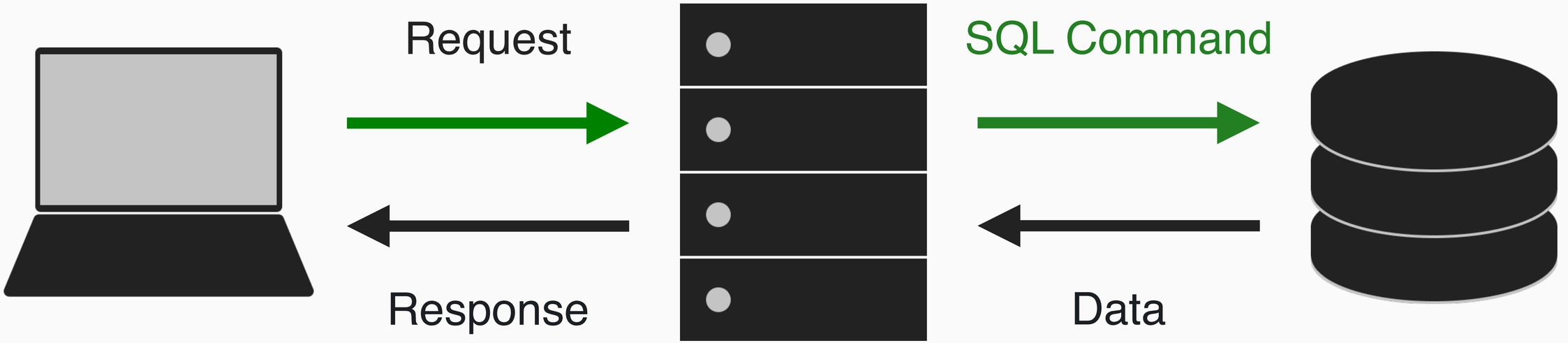
Server

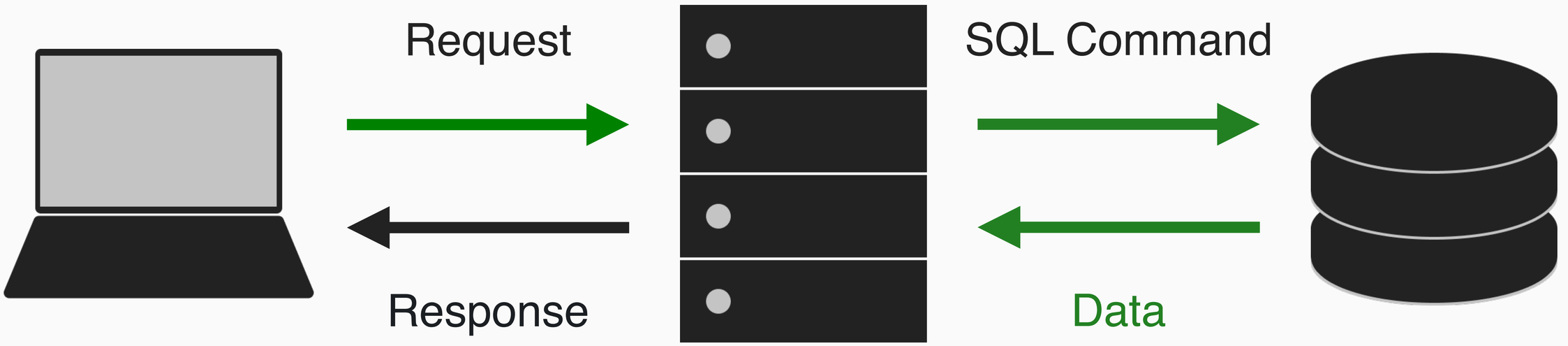
Route triggered
Data operation
Return response

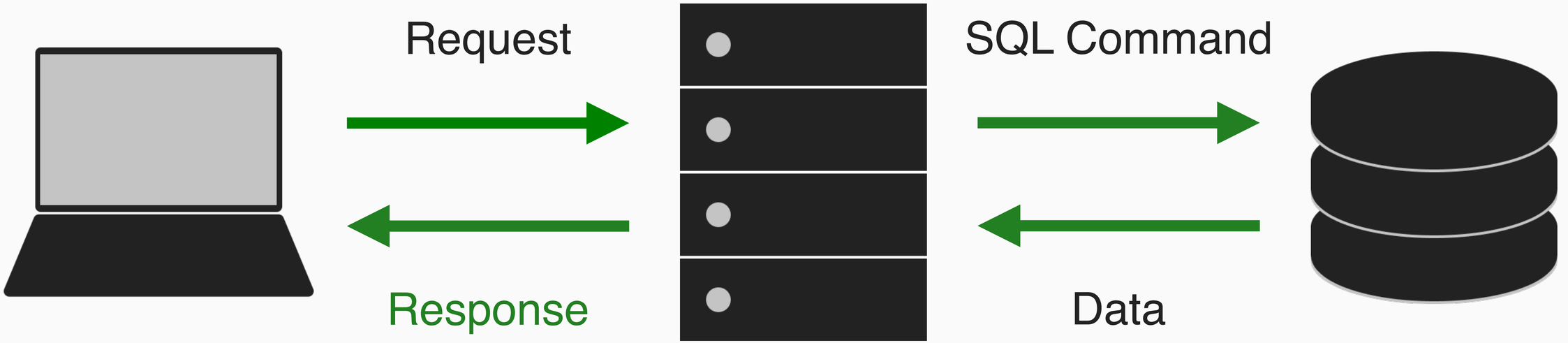

Request
Response
Client
Server


Request
Response
SQL Command
Data
Client
Server
Database


Request
SQL Command
Response
Data
Client
Server
Database


Request
Response
SQL Command
Data
Client
Server
Database


Request
Response
SQL Command
Data
Client
Server
Database


Request
Response
SQL Command
Data
Client
Server
Database

Single Table Database

- Tables structure information into rows and columns
- Columns represent a field with a defined type
- Rows represent a collection of column values
- Query and manipulate data with SQL

Course Overview

1. Routes
2. Databases
3. Relational Databases
4. Abstractions
5. Containerization & DevOps
6. Deployment & Services
7. Hack Challenge & Authentication
8. Images & OAuth

Course Overview

1. Routes
2. Databases
3. Relational Databases
4. Abstractions
5. Containerization & DevOps
6. Deployment & Services
7. Hack Challenge & Authentication
8. Images & OAuth

Questions?

Relational Databases

Overview

- Allows relationships between data
- Which comments belong to which post?
- Which users are friends?
- Which transactions relate to which senders and receivers?

id	name
1	“Jessica”
2	“Shungo”

User Table

id	text	user_id
1	“Shohei Ohtani is the coolest person ever ⚾”	2
2	“I love TikTok!”	1

Post Table

id	name
1	“Jessica”
2	“Shungo”

User Table

id	text	user_id
1	“Shohei Ohtani is the coolest person ever ⚾”	2
2	“I love TikTok!”	1

Post Table

id	name
1	“Jessica”
2	“Shungo”

User Table

id	text	user_id
1	“Shohei Ohtani is the coolest person ever ⚾”	2
2	“I love TikTok!”	1

Post Table

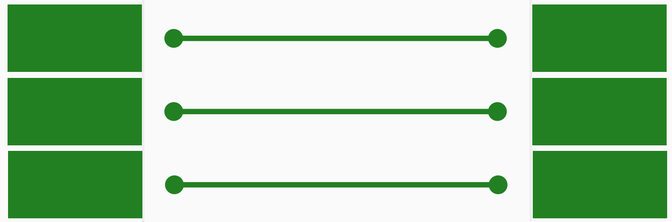
Foreign Keys

- Column in one table that uniquely identifies a specific row
- Also called a “secondary key”
- user_id field in Post Table refers to the id belonging to a user row
- Foreign Key column = relationship

Relationship Types

1. One-to-One
2. One-to-Many
3. Many-to-Many

One-to-One



- Row *x* in table A refers to **one** row *y* in table B
- Row *y* in table B refers to **one** row *x* in table A
- Example: students and netid's
  - Each student has only **one** netid
  - Each netid belongs to only **one** student

User Table

id	name
10	“Shungo”
20	“Jessica”

NetID Table

id	netid	user_id
30	“jzs27”	20
40	“sn685”	10

User Table

id	name
10	“Shungo”
20	“Jessica”

NetID Table

id	netid	user_id
30	“jzs27”	20
40	“sn685”	10

User Table

id	name
10	“Shungo”
20	“Jessica”

NetID Table

id	netid	user_id
30	“jzs27”	20
40	“sn685”	10

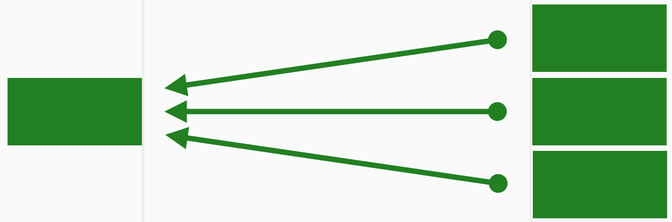
```
CREATE TABLE user (
  id   INTEGER PRIMARY KEY AUTOINCREMENT,
  name TEXT NOT NULL
);

CREATE TABLE netid (
  id      INTEGER PRIMARY KEY AUTOINCREMENT,
  netid   TEXT NOT NULL,
  user_id INTEGER UNIQUE,
  FOREIGN KEY user_id REFERENCES user(id)
);
```

```
INSERT INTO user (name)
     VALUES (“Jessica”);
```

```
INSERT INTO netid (netid, user_id)
     VALUES (“jzs27”, 1);
```

One-to-Many



- Row *x* in table A refers to **many** rows *y1, y2, …* in table B
- Row *y* in table B refers to **one** row *x* in table A
- Example: baseball teams and players
  - Each team can have **many** players
  - Each player belongs to only **one** team

Album Table

id	team_name
10	“IGOR”
20	“4:44”

Song Table

id	name	team_id
30	“EARFQUAKE”	10
40	“I THINK”	10
50	“4:44”	20
60	“Moonlight”	20

Album Table

id	team_name
10	“IGOR”
20	“4:44”

Song Table

id	name	team_id
30	“EARFQUAKE”	10
40	“I THINK”	10
50	“4:44”	20
60	“Moonlight”	20

Album Table

id	team_name
10	“IGOR”
20	“4:44”

Song Table

id	name	team_id
30	“EARFQUAKE”	10
40	“I THINK”	10
50	“4:44”	20
60	“Moonlight”	20

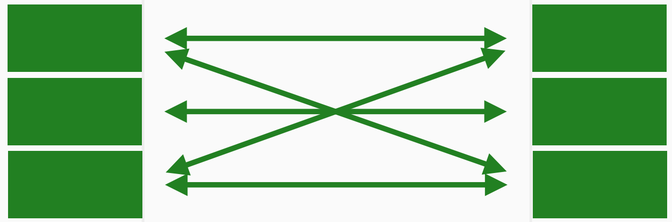
```
CREATE TABLE album (
  id   INTEGER PRIMARY KEY AUTOINCREMENT,
  name TEXT NOT NULL
);

CREATE TABLE song (
  id        INTEGER PRIMARY KEY AUTOINCREMENT,
  name      TEXT NOT NULL,
  album_id  INTEGER NOT NULL,
  FOREIGN KEY album_id REFERENCES album(id)
);
```

```
INSERT INTO song (name, album_id)
     VALUES (“EARFQUAKE”, 10);


SELECT *
  FROM song
 WHERE album_id = 10;
```

Many-to-Many



- Row *x* in table A refers to **many** rows *y1, y2, …* in table B
- Row *y* in table B refers to **many** rows *x1, x2, …* in table A
- Example: students and courses
  - Each student can be in **many** courses
  - Each course can have **many** students

Implementation

- Three potential choices, two are not ideal:
  1. Multiple columns to store more foreign keys
  2. Single column to store a list of id’s
  3. Join / association table

Student Table

id	netid	course1	course2	course3
1	“sn685”	1110	1998	2110
2	“jzs27”	1110	2110	2800

Student Table

id	netid	course1	course2	course3
1	“sn685”	1110	1998	2110
2	“jzs27”	1110	2110	2800

What happens if a student wants more than 3 courses?

Student Table

id	netid	course1	course2	course3		
1	“sn685”	1110	1998	2110		
2	“jzs27”	1110	2110	2800		
3	“gg387”	1110	1998	2110	2800	3110

Student Table

id	netid	course1	course2	course3	course4	course5
1	“sn685”	1110	1998	2110	*null*	*null*
2	“jzs27”	1110	2110	2800	*null*	*null*
3	“gg387”	1110	1998	2110	2800	3110

Student Table

id	netid	course1	course2	course3	course4	course5
1	“sn685”	1110	1998	2110	*null*	*null*
2	“jzs27”	1110	2110	2800	*null*	*null*
3	“gg387”	1110	1998	2110	2800	3110
4	“rm834”	4810	*null*	*null*	*null*	*null*

Student Table

id	netid	courses
1	“sn685”	1110, 1998
2	“jzs27”	1110

Course Table

id	title	students
1110	“CS1110”	1, 2
1998	“CS1998”	1

Student Table

id	netid	courses
1	“sn685”	1110, 1998
2	“jzs27”	1110

Course Table

id	title	students
1110	“CS1110”	1, 2
1998	“CS1998”	1

Student Table

id	netid	courses
1	“sn685”	1110, 1998
2	“jzs27”	1110

Course Table

id	title	students
1110	“CS1110”	1, 2
1998	“CS1998”	1

Student Table

id	netid	courses
1	“sn685”	1110, 1998
2	“jzs27”	1110
3	“gg387”	1110, 1998

Course Table

id	title	students
1110	“CS1110”	1, 2
1998	“CS1998”	1

Student Table

id	netid	courses
1	“sn685”	1110, 1998
2	“jzs27”	1110
3	“gg387”	1110, 1998

Course Table

id	title	students
1110	“CS1110”	1, 2
1998	“CS1998”	1

Student Table

id	netid	courses
1	“sn685”	1110, 1998
2	“jzs27”	1110
3	“gg387”	1110, 1998

Course Table

id	title	students
1110	“CS1110”	1, 2, 3
1998	“CS1998”	1, 3

Student Table

id	netid	courses
1	“sn685”	1110, 1998
2	“jzs27”	1110

Course Table

id	title	students
1110	“CS1110"	1, 2
1998	“CS1998”	1

Student Table

id	netid	courses
1	“sn685”	1110, 1998
2	“jzs27”	1110

Course Table

id	title	students
1110	“CS1110”	1, 2
1998	“CS1998”	1

Student Table

id	netid
1	“sn685”
2	“jzs27”

Join / Association Table

student_id	course_id
1	1110
1	1998
2	1110

Course Table

id	title
1110	“CS1110”
1998	“CS1998”

Student Table

id	netid
1	“sn685”
2	“jzs27”

Join / Association Table

student_id	course_id
1	1110
1	1998
2	1110

Course Table

id	title
1110	“CS1110”
1998	“CS1998”

Student Table

id	netid
1	“sn685”
2	“jzs27”

Join / Association Table

student_id	course_id
1	1110
1	1998
2	1110

Course Table

id	title
1110	“CS1110”
1998	“CS1998”

Student Table

id	netid
1	“sn685”
2	“jzs27”

Join / Association Table

student_id	course_id
1	1110
1	1998
2	1110

Course Table

id	title
1110	“CS1110”
1998	“CS1998”

Student Table

id	netid
1	“sn685”
2	“jzs27”

Join / Association Table

student_id	course_id
1	1110
1	1998
2	1110

Course Table

id	title
1110	“CS1110”
1998	“CS1998”

Student Table

id	netid
1	“sn685”
2	“jzs27”

Join / Association Table

student_id	course_id
1	1110
1	1998
2	1110

Course Table

id	title
1110	“CS1110”
1998	“CS1998”

Student Table

id	netid
1	“sn685”
2	“jzs27”

Join / Association Table

student_id	course_id
1	1110
1	1998
2	1110

Course Table

id	title
1110	“CS1110”
1998	“CS1998”

Student Table

id	netid
1	“sn685”
2	“jzs27”

Join / Association Table

student_id	course_id
1	1110
1	1998
2	1110

Course Table

id	title
1110	“CS1110”
1998	“CS1998”

Student Table

id	netid
1	“sn685”
2	“jzs27”

Join / Association Table

student_id	course_id
1	1110
1	1998
2	1110
2	1998

Course Table

id	title
1110	“CS1110”
1998	“CS1998”

```
CREATE TABLE student (
   id    INTEGER PRIMARY KEY AUTOINCREMENT,
   netid TEXT NOT NULL
);

CREATE TABLE course (
   id    INTEGER PRIMARY KEY AUTOINCREMENT,
   title TEXT NOT NULL
);
```

```
CREATE TABLE join (
   id          INTEGER PRIMARY KEY AUTOINCREMENT,
   course_id   INTEGER NOT NULL,
   student_id INTEGER NOT NULL,
   FOREIGN KEY course_id REFERENCES course(id)
   FOREIGN KEY student_id REFERENCES student(id)
);
```

Demo