DEV ACADEMY
TE KURA HANGARAU
O AOTEAROA

DB Relationships Review

and

Complex Joins

Agenda

- **Database Relationships**
  - Database Diagrams
  - One to One
  - One to Many
  - Many to Many

- **Complex Joins**
  - Joining more than 2 tables

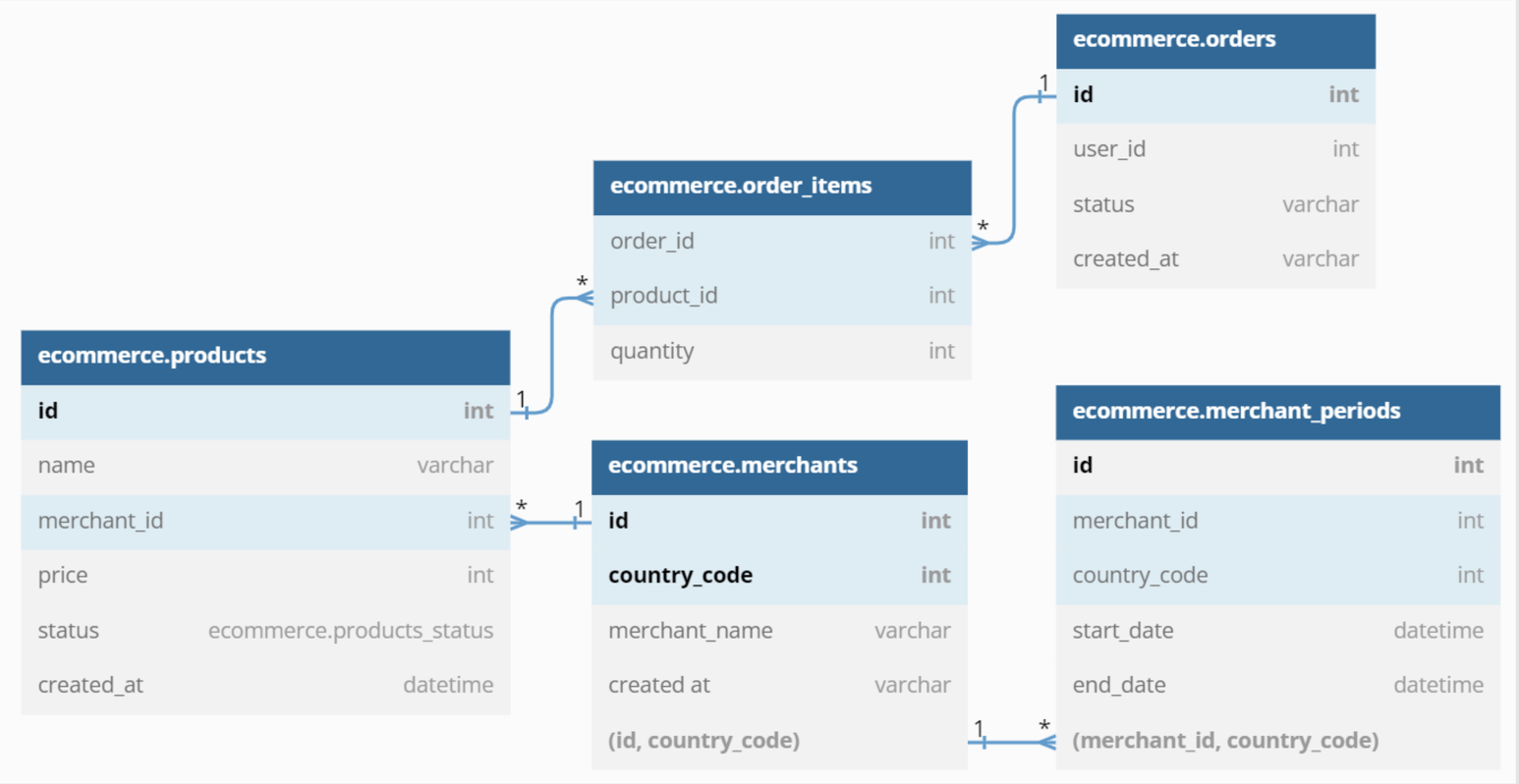
Database Relationships

Database Diagrams

- Entity Relationship Model (ERM)
  - Describes relationships within a specific domain
- Entity Relationship Diagram (ERD)
  - A visual representation of relationships within a specific domain
- Database Diagram
  - A visual representation of the relationships within a database
- The relationships can be between things like:
  - People
  - Objects
  - Places
  - Events

Database Relationships

Database Diagrams



Database Relationships

One to One

- One record in a table is associated with a maximum of one record in another table.
- Examples:
  - One employee to One company car
  - One person to One active passport
  - One account to One profile

Database Relationships

One to One

```
exports.up = function(knex) {
  return knex.schema.createTable('students', table => {
    table.increments('id')
    table.string('name')
    table.integer('age')
  })
};
```

Database Relationships

One to One

```
exports.up = function(knex) {
  return knex.schema.createTable('students', table => {
    table.increments('id')
    table.string('name')
    table.integer('age')
  })
};
```

students	
id	increments
name	string
age	int

Database Relationships

One to One

```
exports.up = function(knex) {
  return knex.schema.createTable('students', table => {
    table.increments('id')
    table.string('name')
    table.integer('age')
  })
};
```

```
exports.up = function(knex) {
  return knex.schema.createTable('studentDetails', table => {
    table.increments('id')
    table.integer('student_id')
    table.integer('height')
  })
};
```

students	
id	increments
name	string
age	int

Database Relationships

One to One

```
exports.up = function(knex) {
  return knex.schema.createTable('students', table => {
    table.increments('id')
    table.string('name')
    table.integer('age')
  })
};
```

```
exports.up = function(knex) {
  return knex.schema.createTable('studentDetails', table => {
    table.increments('id')
    table.integer('student_id')
    table.integer('height')
  })
};
```

students	
id	increments
name	string
age	int

studentDetails	
id	increments
student_id	int
height	int

Database Relationships

One to One

```
exports.up = function(knex) {
  return knex.schema.createTable('students', table => {
    table.increments('id')
    table.string('name')
    table.integer('age')
  })
};
```

```
exports.up = function(knex) {
  return knex.schema.createTable('studentDetails', table => {
    table.increments('id')
    table.integer('student_id').references('students.id').unique()
    table.integer('height')
  })
};
```

students	
id	increments
name	string
age	int

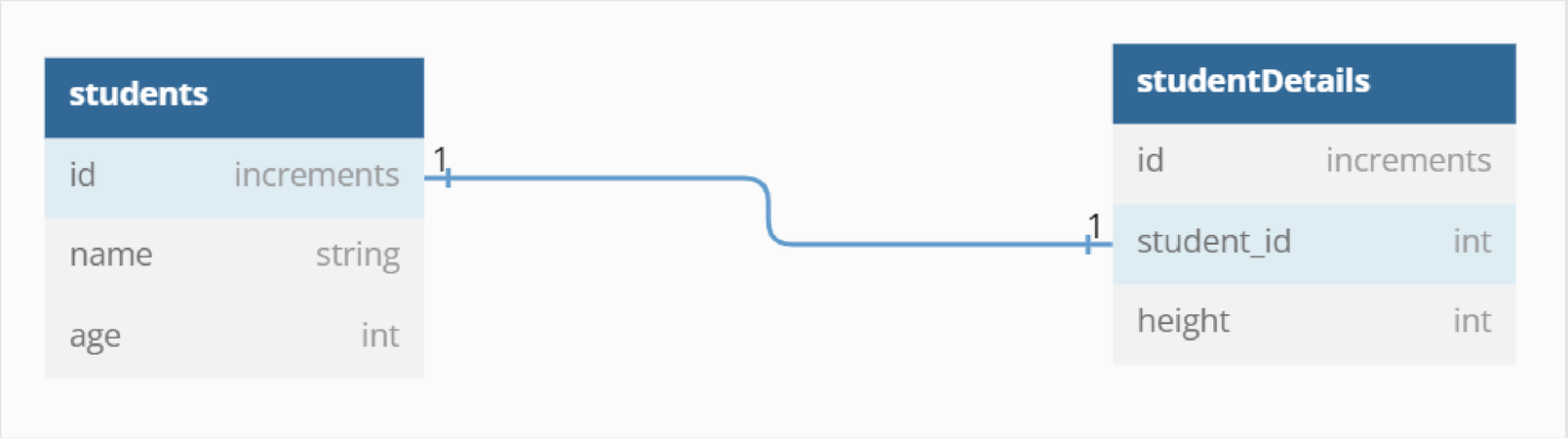
studentDetails	
id	increments
student_id	int
height	int

Database Relationships

One to One

```
exports.up = function(knex) {
  return knex.schema.createTable('students', table => {
    table.increments('id')
    table.string('name')
    table.integer('age')
  })
};
```

```
exports.up = function(knex) {
  return knex.schema.createTable('studentDetails', table => {
    table.increments('id')
    table.integer('student_id').references('students.id').unique()
    table.integer('height')
  })
};
```



Database Relationships

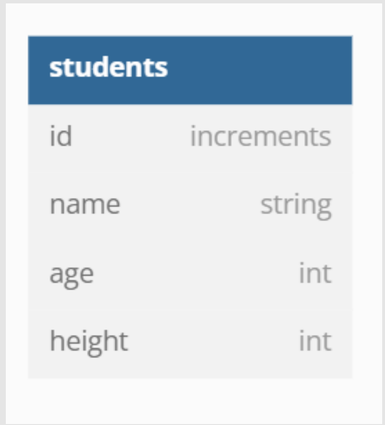
One to One

```
exports.up = function(knex) {
  return knex.schema.createTable('students', table => {
    table.increments('id')
    table.string('name')
    table.integer('age')
    table.integer('height')
  })
};
```

Database Relationships

One to One

```
exports.up = function(knex) {
  return knex.schema.createTable('students', table => {
    table.increments('id')
    table.string('name')
    table.integer('age')
    table.integer('height')
  })
};
```



Database Relationships

One to Many

- One record in a table can be associated with one or more records in another table.
- Examples:
  - One employee to Many sales
  - One person to Many cars
  - One person to Many pets

Database Relationships

One to Many

```
exports.up = function(knex) {
  return knex.schema.createTable('students', table => {
    table.increments('id')
    table.string('name')
  })
};
```

Database Relationships

One to Many

```
exports.up = function(knex) {
  return knex.schema.createTable('students', table => {
    table.increments('id')
    table.string('name')
  })
};
```

students	
id	increments
name	string

Database Relationships

One to Many

```
exports.up = function(knex) {
  return knex.schema.createTable('students', table => {
    table.increments('id')
    table.string('name')
  })
};
```

```
exports.up = function(knex) {
  return knex.schema.createTable('teachers', table => {
    table.increments('id')
    table.string('name')
    table.string('subject')
  })
};
```

students	
id	increments
name	string

Database Relationships

One to Many

```
exports.up = function(knex) {
  return knex.schema.createTable('students', table => {
    table.increments('id')
    table.string('name')
  })
};
```

```
exports.up = function(knex) {
  return knex.schema.createTable('teachers', table => {
    table.increments('id')
    table.string('name')
    table.string('subject')
  })
};
```

students	
id	increments
name	string

teachers	
id	increments
name	string
subject	string

Database Relationships

One to Many

```
exports.up = function(knex) {
  return knex.schema.createTable('students', table => {
    table.increments('id')
    table.string('name')
    table.integer('teacher_id')
  })
};
```

```
exports.up = function(knex) {
  return knex.schema.createTable('teachers', table => {
    table.increments('id')
    table.string('name')
    table.string('subject')
  })
};
```

students	
id	increments
name	string

teachers	
id	increments
name	string
subject	string

Database Relationships

One to Many

```
exports.up = function(knex) {
  return knex.schema.createTable('students', table => {
    table.increments('id')
    table.string('name')
    table.integer('teacher_id')
  })
};
```

```
exports.up = function(knex) {
  return knex.schema.createTable('teachers', table => {
    table.increments('id')
    table.string('name')
    table.string('subject')
  })
};
```

students	
id	increments
name	string
teacher_id	int

teachers	
id	increments
name	string
subject	string

Database Relationships

One to Many

```
exports.up = function(knex) {
  return knex.schema.createTable('students', table => {
    table.increments('id')
    table.string('name')
    table.integer('teacher_id').references('teachers.id')
  })
};
```

```
exports.up = function(knex) {
  return knex.schema.createTable('teachers', table => {
    table.increments('id')
    table.string('name')
    table.string('subject')
  })
};
```

students	
id	increments
name	string
teacher_id	int

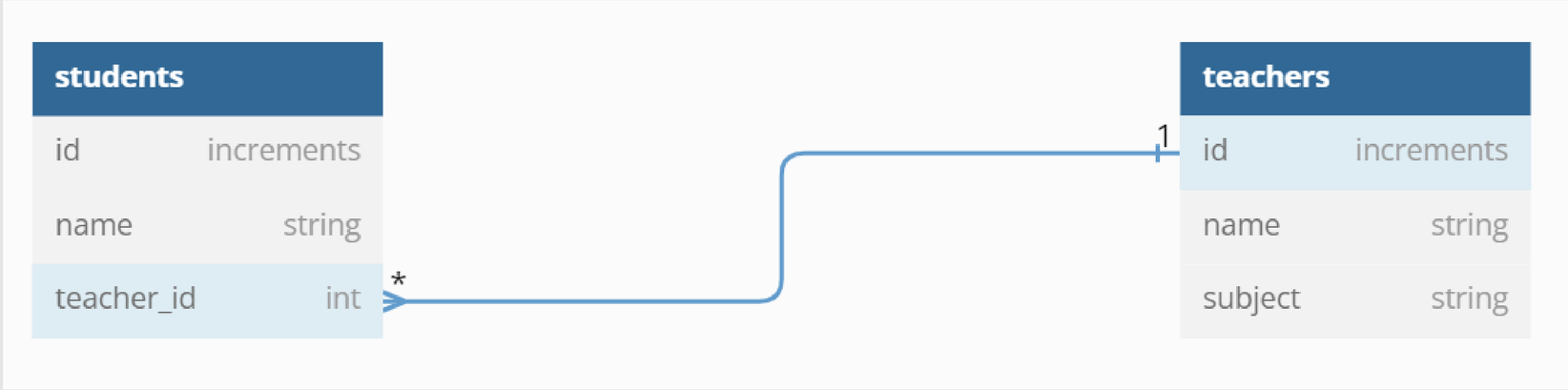
teachers	
id	increments
name	string
subject	string

Database Relationships

One to Many

```
exports.up = function(knex) {
  return knex.schema.createTable('students', table => {
    table.increments('id')
    table.string('name')
    table.integer('teacher_id').references('teachers.id')
  })
};
```

```
exports.up = function(knex) {
  return knex.schema.createTable('teachers', table => {
    table.increments('id')
    table.string('name')
    table.string('subject')
  })
};
```



Database Relationships

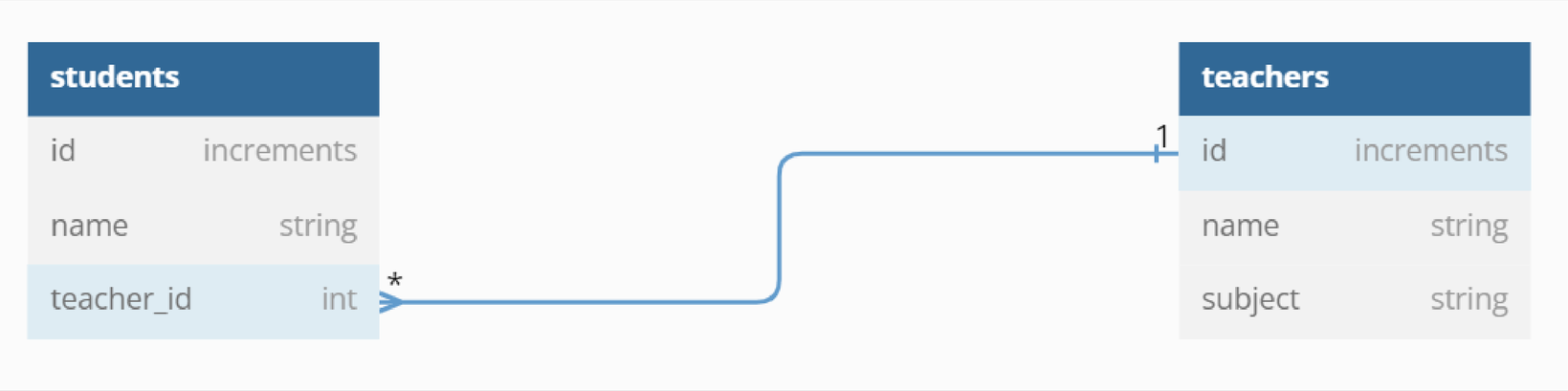
One to Many

students

id	name	teacher_id

teachers

id	name	subject



Database Relationships

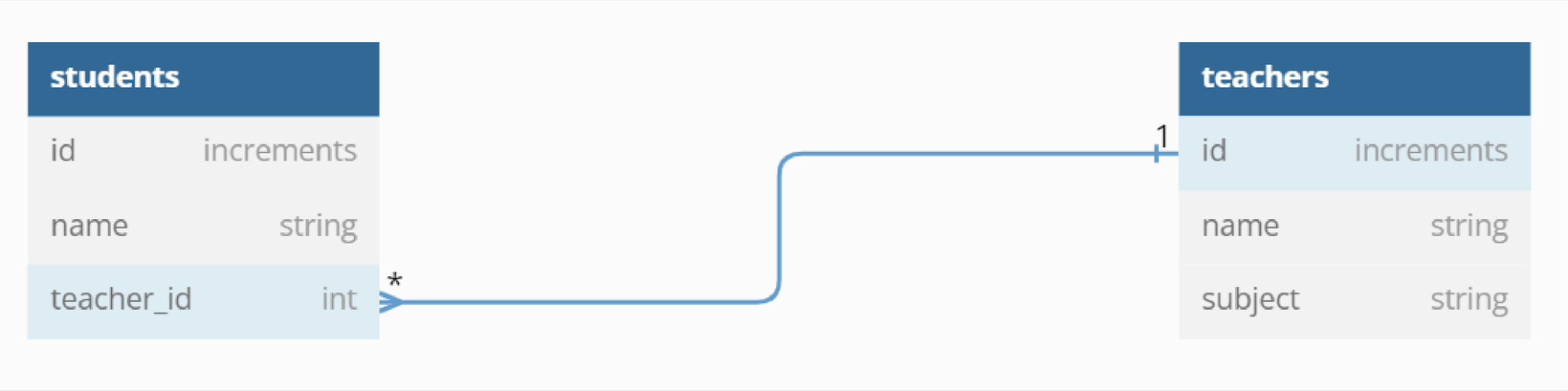
One to Many

students

id	name	teacher_id
1	JV	
2	Sarah	

teachers

id	name	subject



Database Relationships

One to Many

students

id	name	teacher_id
1	JV	1
2	Sarah	1

teachers

id	name	subject
1	Jatin	React

Joins

One to Many

students

id	name	teacher_id
1	JV	1
2	Sarah	1

teachers

id	name	subject
1	Jatin	React

```
db('students')
  .join('teachers', 'students.teacher_id', 'teachers.id')
  .where('teachers.name', 'Jatin')
  .select('students.name')
```

Joins

One to Many

students

id	name	teacher_id
1	JV	1
2	Sarah	1

teachers

id	name	subject
1	Jatin	React
1	Jatin	React

```
db('students')
  .join('teachers', 'students.teacher_id', 'teachers.id')
  .where('teachers.name', 'Jatin')
  .select('students.name')
```

Joins

One to Many

students			teachers		
id	name	teacher_id	id	name	subject
1	JV	1	1	Jatin	React
2	Sarah	1	1	Jatin	React

```
[
  { name: JV },
  { name: Sarah }
]
```

Database Relationships

One to Many

students

id	name	teacher_id
1	JV	1
2	Sarah	1

teachers

id	name	subject
1	Jatin	React

Database Relationships

One to Many

students

id	name	teacher_id
1	JV	1
2	Sarah	1

teachers

id	name	subject
1	Jatin	React
2	Lani	Databases

Database Relationships

Many to Many

- Multiple records in a table can be associated with multiple records in another table.
- Examples:
  - Many students to Many teachers
  - Many authors to Many books
  - Many people to Many properties
- To model a many-to-many relationship, we need to create a new table to connect the other two.
- This new table can be called a relationship table, joining table, intermediate table, linking table or junction table.

Database Relationships

Many to Many

Database Relationships

Many to Many

Database Relationships

Many to Many

```
exports.up = function(knex) {
  return knex.schema.createTable('students', table => {
    table.increments('id')
    table.string('name')
  })
};
```

```
exports.up = function(knex) {
  return knex.schema.createTable('teachers', table => {
    table.increments('id')
    table.string('name')
  })
};
```

students	
id	increments
name	string

teachers	
id	increments
name	string

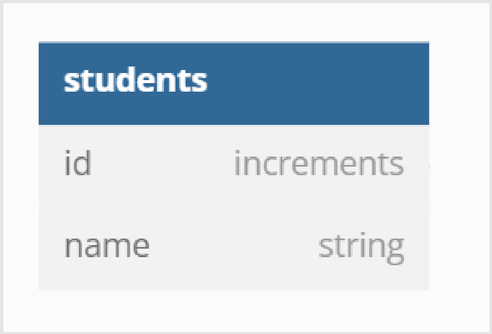
Database Relationships

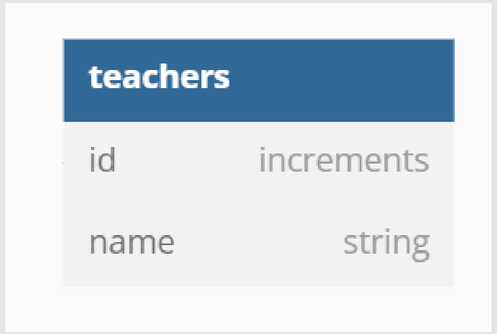
Many to Many

```
exports.up = function(knex) {
  return knex.schema.createTable('students', table => {
    table.increments('id')
    table.string('name')
  })
};
```

```
exports.up = function(knex) {
  return knex.schema.createTable('students_teachers', table => {
    table.integer('student_id')
    table.integer('teacher_id')
  })
};
```

```
exports.up = function(knex) {
  return knex.schema.createTable('teachers', table => {
    table.increments('id')
    table.string('name')
  })
};
```





Database Relationships

Many to Many

```
exports.up = function(knex) {
  return knex.schema.createTable('students', table => {
    table.increments('id')
    table.string('name')
  })
};
```

```
exports.up = function(knex) {
  return knex.schema.createTable('students_teachers', table => {
    table.integer('student_id')
    table.integer('teacher_id')
  })
};
```

```
exports.up = function(knex) {
  return knex.schema.createTable('teachers', table => {
    table.increments('id')
    table.string('name')
  })
};
```

students	
id	increments
name	string

students_teachers	
student_id	int
teacher_id	int

teachers	
id	increments
name	string

Database Relationships

Many to Many

```
exports.up = function(knex) {
  return knex.schema.createTable('students', table => {
    table.increments('id')
    table.string('name')
  })
};
```

```
exports.up = function(knex) {
  return knex.schema.createTable('students_teachers', table => {
    table.integer('student_id').references('students.id')
    table.integer('teacher_id').references('teachers.id')
  })
};
```

```
exports.up = function(knex) {
  return knex.schema.createTable('teachers', table => {
    table.increments('id')
    table.string('name')
  })
};
```

students	
id	increments
name	string

students_teachers	
student_id	int
teacher_id	int

teachers	
id	increments
name	string

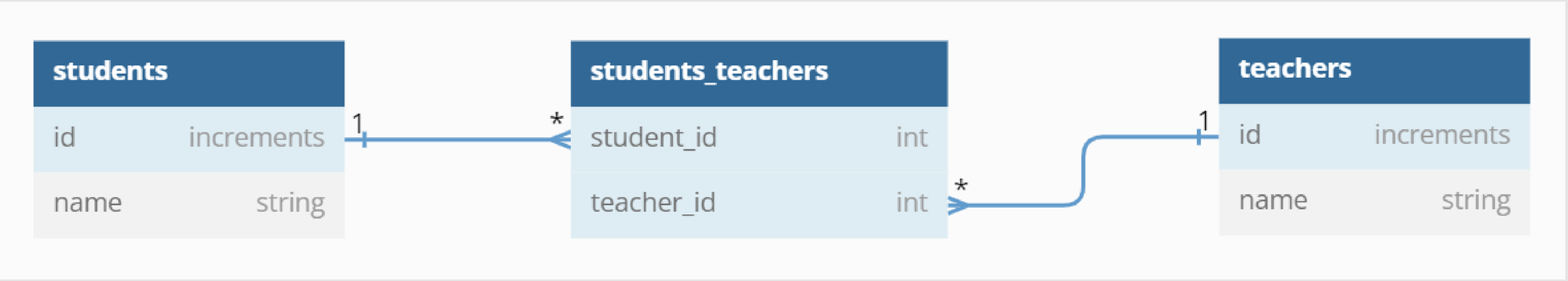
Database Relationships

Many to Many

```
exports.up = function(knex) {
  return knex.schema.createTable('students', table => {
    table.increments('id')
    table.string('name')
  })
};
```

```
exports.up = function(knex) {
  return knex.schema.createTable('students_teachers', table => {
    table.integer('student_id').references('students.id')
    table.integer('teacher_id').references('teachers.id')
  })
};
```

```
exports.up = function(knex) {
  return knex.schema.createTable('teachers', table => {
    table.increments('id')
    table.string('name')
  })
};
```



Database Relationships

Many to Many

students

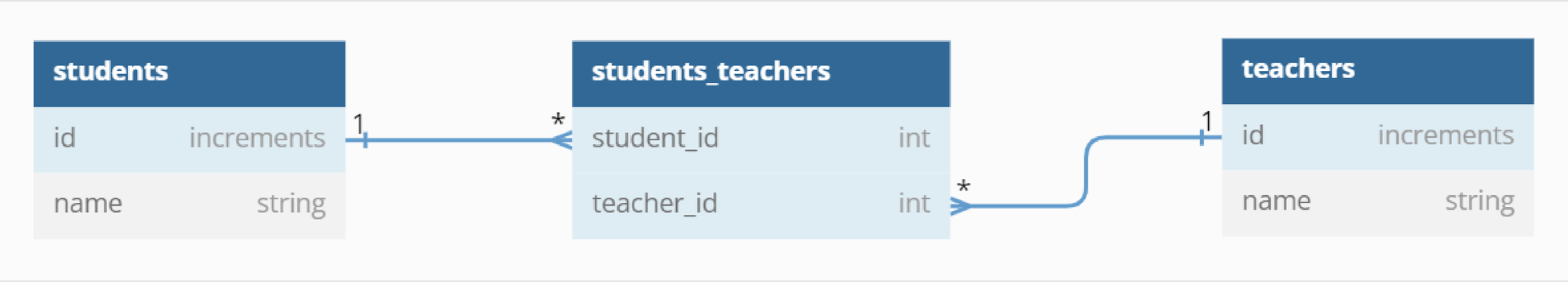
id	name

students_teachers

student_id	teacher_id

teachers

id	name



Database Relationships

Many to Many

students

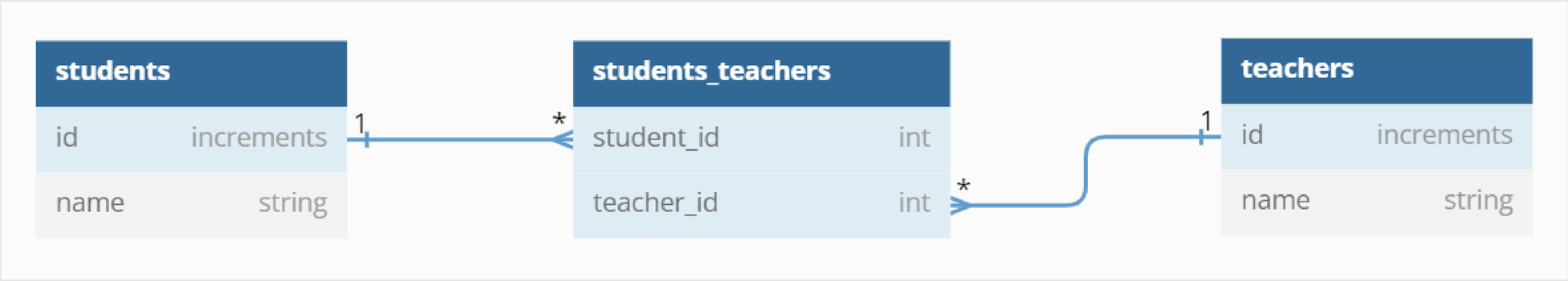
id	name
1	JV
2	Sarah

students_teachers

student_id	teacher_id

teachers

id	name



Database Relationships

Many to Many

students

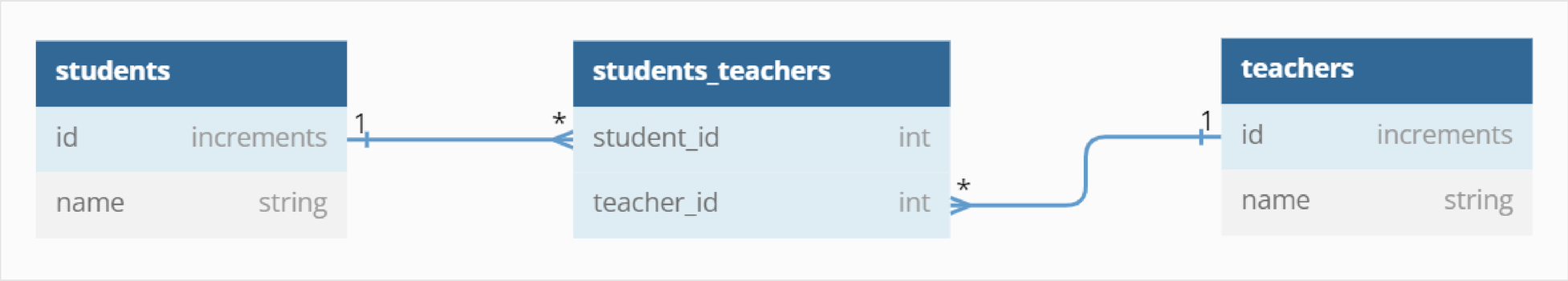
id	name
1	JV
2	Sarah

students_teachers

student_id	teacher_id

teachers

id	name
1	Jatin



Database Relationships

Many to Many

students

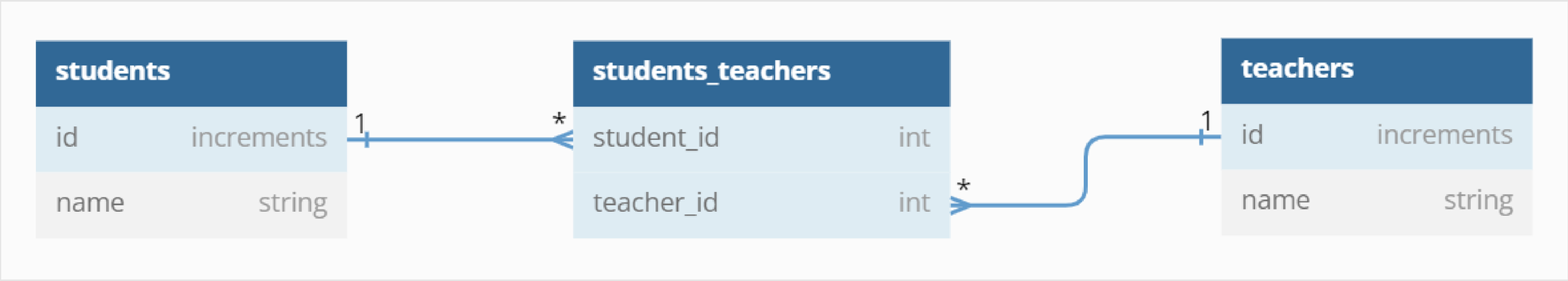
id	name
1	JV
2	Sarah

students_teachers

student_id	teacher_id
1	1

teachers

id	name
1	Jatin



Database Relationships

Many to Many

students

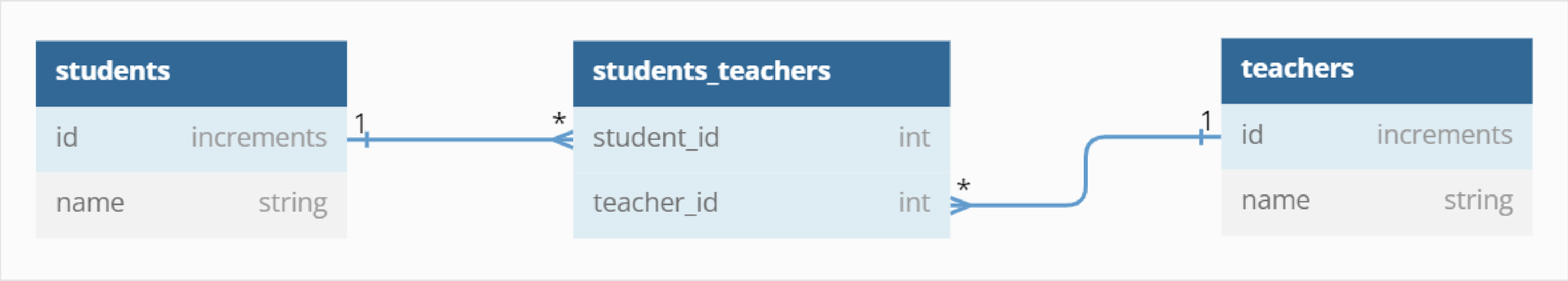
id	name
1	JV
2	Sarah

students_teachers

student_id	teacher_id
1	1
2	1

teachers

id	name
1	Jatin



Database Relationships

Many to Many

students

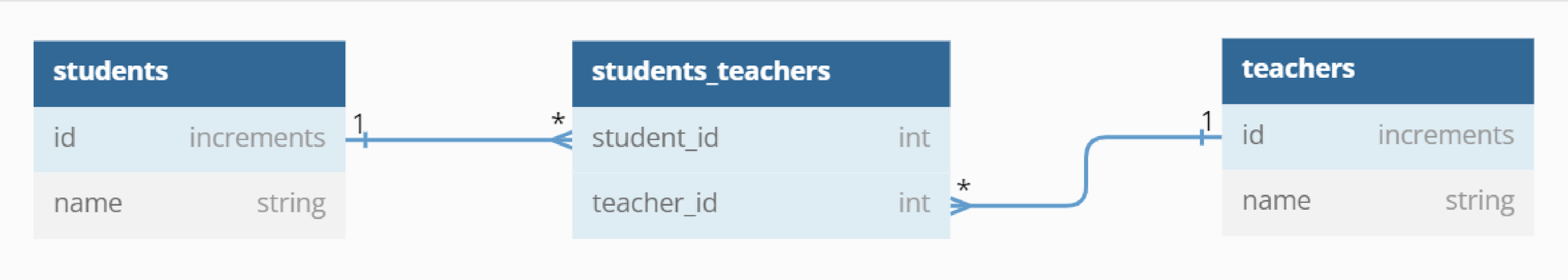
id	name
1	JV
2	Sarah

students_teachers

student_id	teacher_id
1	1
2	1

teachers

id	name
1	Jatin
2	Lani



Complex Joins

Many to Many

students

id	name
1	JV
2	Sarah

students_teachers

student_id	teacher_id
1	1
2	1
1	2
2	2

teachers

id	name
1	Jatin
2	Lani

Complex Joins

Many to Many

students

id	name
1	JV
2	Sarah

students_teachers

student_id	teacher_id
1	1
2	1
1	2
2	2

teachers

id	name
1	Jatin
2	Lani

```
db('students')
```

Complex Joins

Many to Many

students

id	name
1	JV
2	Sarah

students_teachers

student_id	teacher_id
1	1
2	1
1	2
2	2

teachers

id	name
1	Jatin
2	Lani

```
db('students')
  .join('students_teachers', 'students.id', 'students_teachers.student_id')
```

Complex Joins

Many to Many

students

id	name
1	JV
2	Sarah

students_teachers

student_id	teacher_id
1	1
2	1
1	2
2	2

teachers

id	name
1	Jatin
2	Lani

```
db('students')
  .join('students_teachers', 'students.id', 'students_teachers.student_id')
  .join('teachers', 'students_teachers.teacher_id', 'teachers.id')
```

Complex Joins

Many to Many

students		students_teachers		teachers	
id	name	student_id	teacher_id	id	name
1	JV	1	1	1	Jatin
1	JV	1	2	2	Lani
2	Sarah	2	1	1	Jatin
2	Sarah	2	2	2	Lani

```
db('students')
  .join('students_teachers', 'students.id', 'students_teachers.student_id')
  .join('teachers', 'students_teachers.teacher_id', 'teachers.id')
```

Complex Joins

Many to Many

students

id	name
1	JV
2	Sarah

students_teachers

student_id	teacher_id
1	2
2	2

teachers

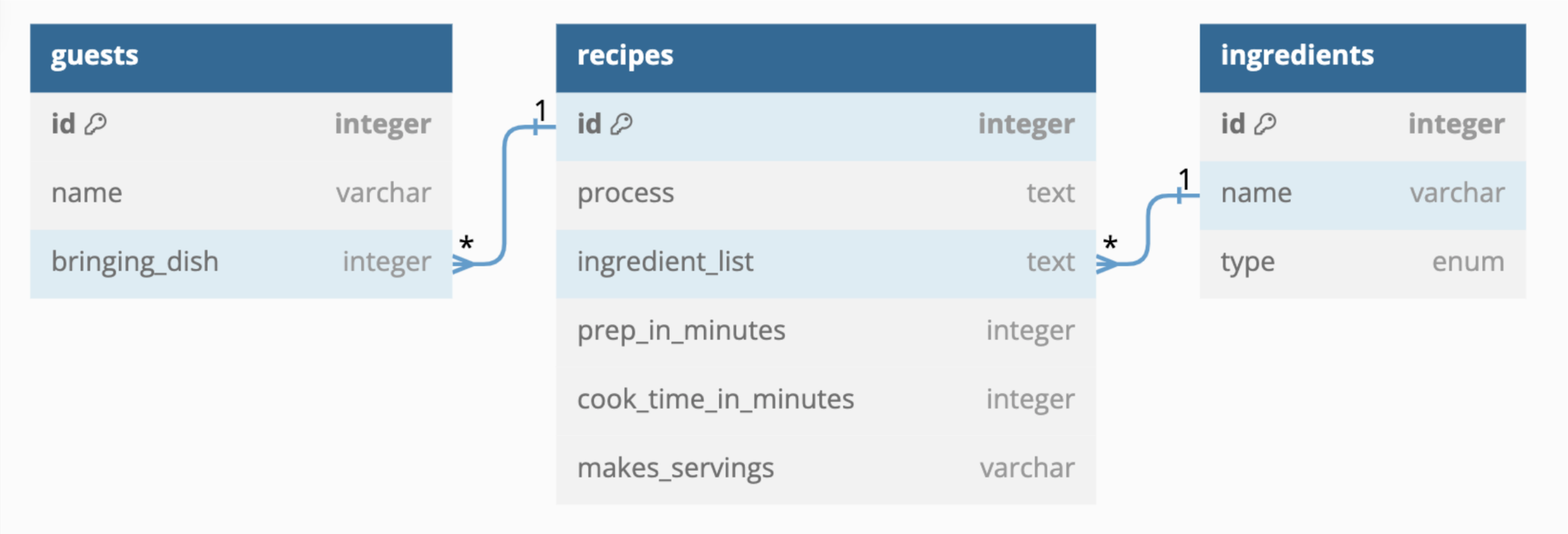
id	name
2	Lani
2	Lani

```
db('students')
  .join('students_teachers', 'students.id', 'students_teachers.student_id')
  .join('teachers', 'students_teachers.teacher_id', 'teachers.id')
  .where('teachers.name', 'Lani')
  .select('students.name as studentName', 'teachers.name as teacherName')
```

```
[
  { studentName: JV, teacherName: Lani },
  { studentName: Sarah, teacherName: Lani }
]
```

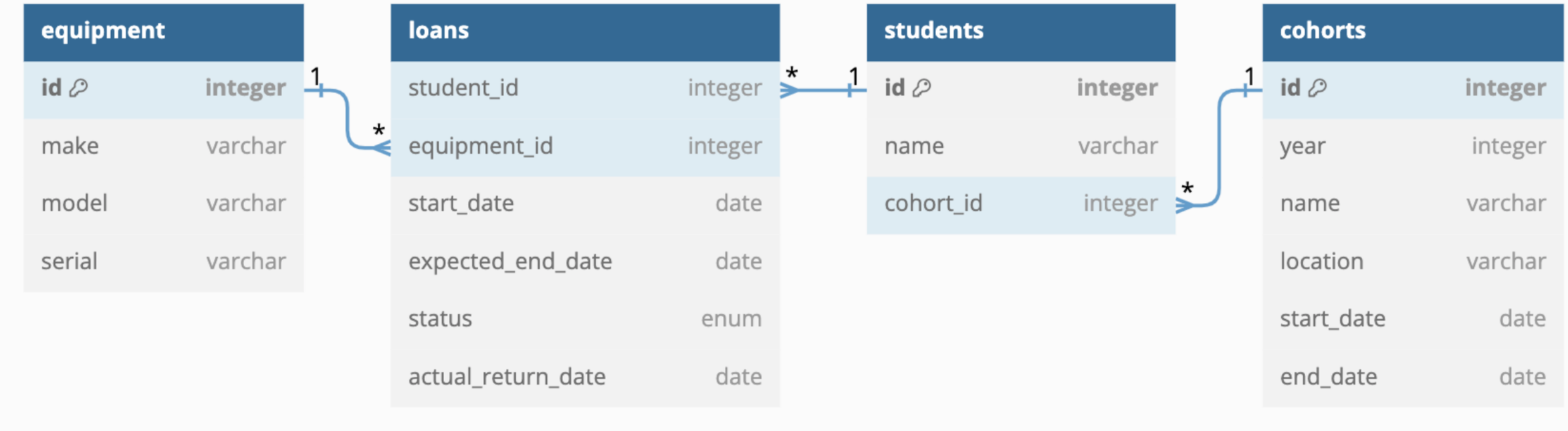
Complex Joins

NOT many to many



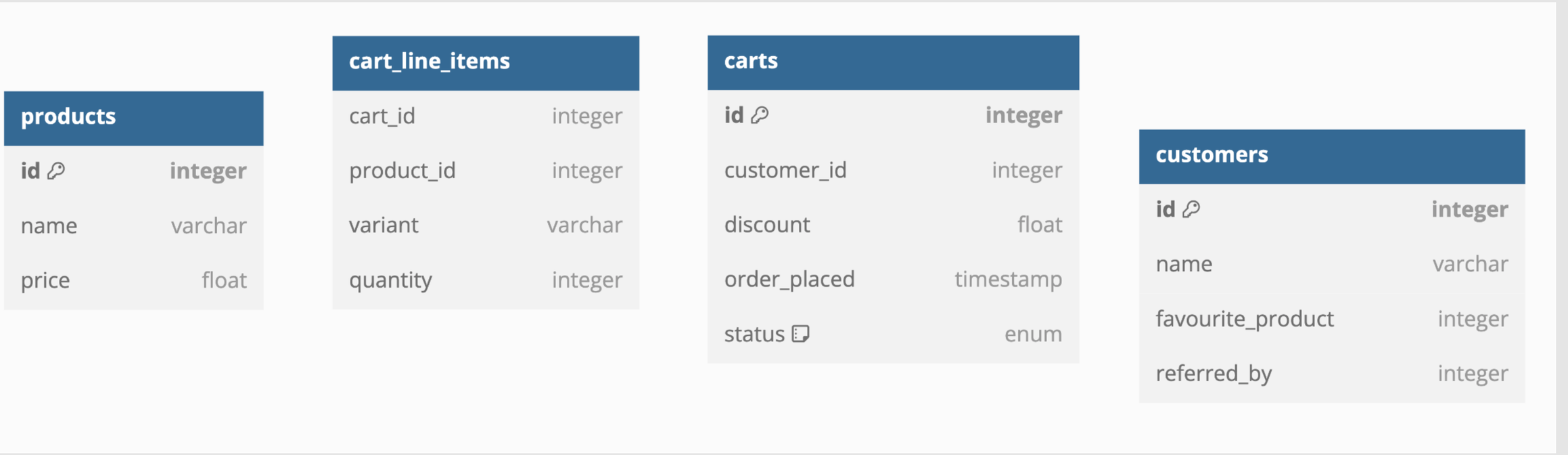
Complex Joins

not only many to many



Complex Joins

not only many to many



Challenge

dreamfest (+ reference images and feedback)