

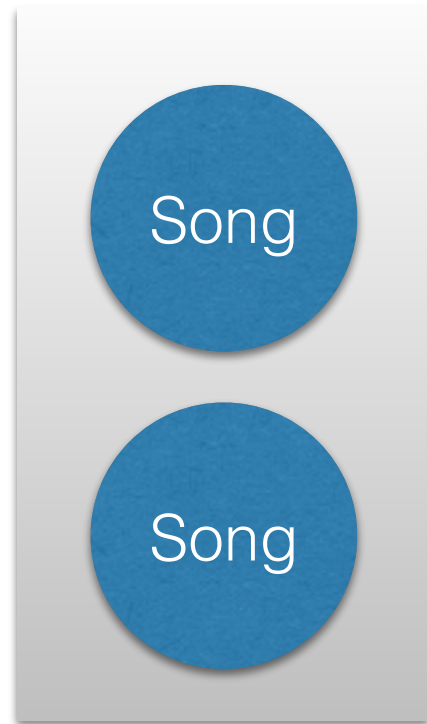
ORM

Object Relational Mapping



What is an ORM?

OOP



Relational DB



Object Relational Mapping

Examples

Create a song record in **songs**

```
$song = new Song();  
$song->title = 'Sound of winter';  
$song->artist_id = 25;  
$song->price = 1.29;  
$song->save();
```



Table: **songs**

id
title
artist_id
price
created_at

ORM Advantages

Development speed - You don't have to write low-level data access code.

Think about your database in an object oriented fashion

Don't have to write SQL, but still learn SQL!

Examples

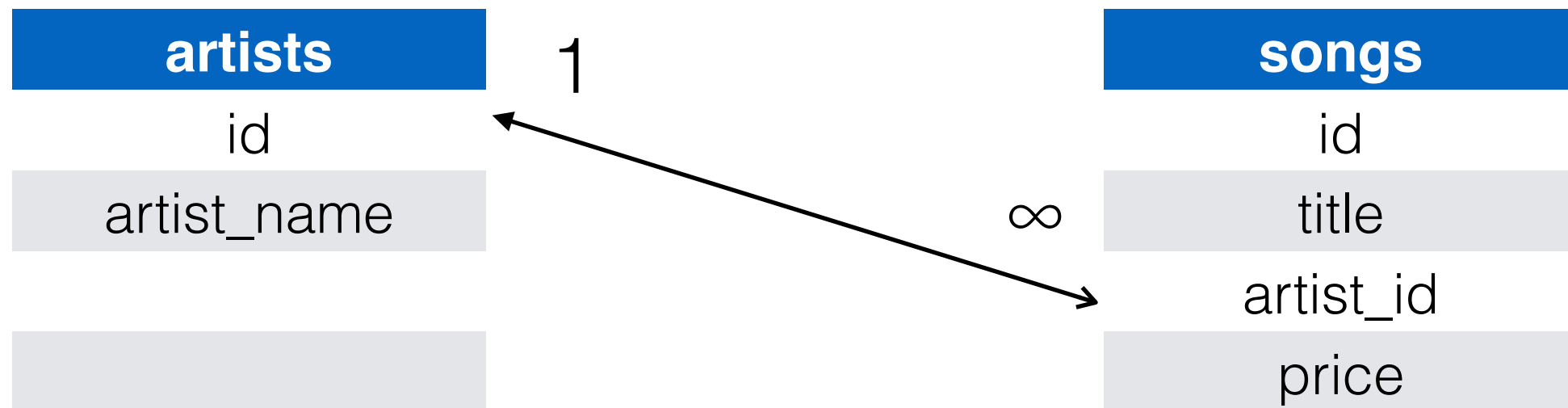
Updating song record 5 in **songs**

```
$song = Song::find(5);  
$song->title = 'Sound of winter';  
$song->price = 0.99;  
$song->save();
```

Exercises: Basic Crud

Relationships

One-to-Many Relationships



An artist **has many** songs

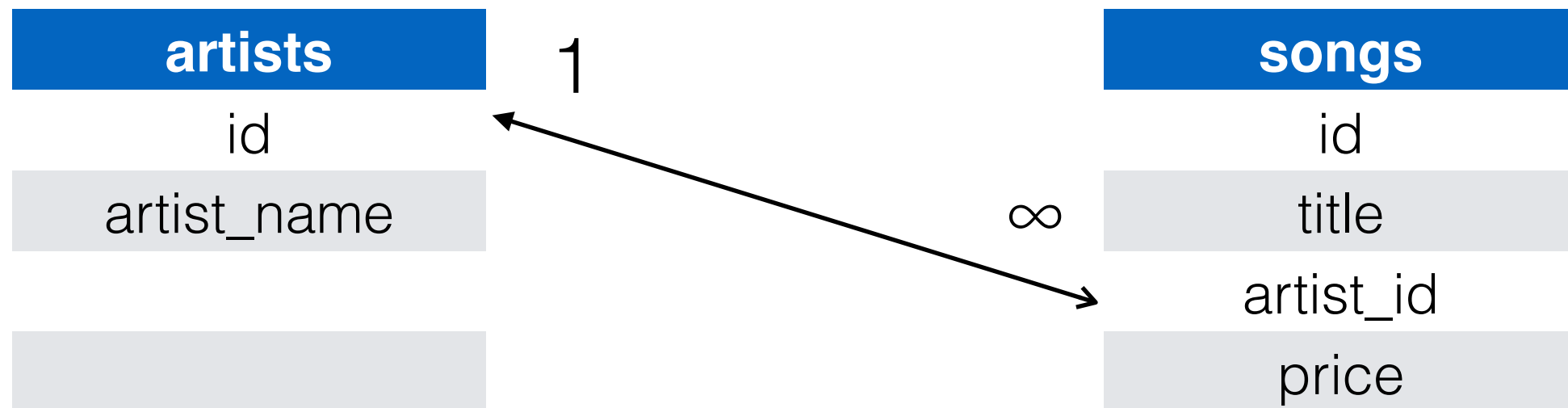
hasMany()

```
class Artist extends Model {  
  public function songs()  
  {  
    return $this->hasMany( 'App\Models\Song' );  
  }  
}
```

```
$artist = Artist::find(5);  
$songsByArtist = $artist->songs()->get();
```

```
SELECT * FROM artists WHERE artists.id = 5 LIMIT 1  
SELECT * FROM songs WHERE songs.artist_id = 5
```

One-to-Many Relationships



A song **belongs to** an artist

belongsTo()

```
$song = Song::all();
```

What about accessing the artist that each song belongs to?

```
class Song extends Model {  
    public function artist()  
    {  
        return $this->belongsTo( 'App\Models\Artist' );  
    }  
}
```

```
songs.artist_id = artists.id
```

```
$songs = Song::take(5)->get();
```

```
foreach ($songs as $song) {  
    // Artist model  
    var_dump($song->artist);  
}
```

Populating an object with data **on demand** (lazy loading)

N + 1 queries

6 queries

(1) SELECT * FROM songs LIMIT 5

(5) SELECT * FROM artist WHERE id = ?

Death by a Thousand Queries

- $N + 1$ problem might execute too many queries on a given page, which could be slow
- Lazy-loading is good if you have a small number of queries. Sometimes a few small queries are faster than one large, complex query

Eager Loading

- Load up models with related models. Also called hydration.
- Ex: Load up each **Song** model with the corresponding Artist model



Eager Loading

```
$songs = Song::with('artist', 'genre')  
    ->take(5)  
    ->get();
```

```
foreach ($songs as $song) {  
    $song->artist; // Artist model  
    $song->genre;  // Genre model  
}
```

Exercises: Relationships

Disadvantages

- Less flexibility depending on the ORM
- Performance i.e. $N + 1$ problem if you're not careful
- Developer productivity when learning an ORM

Popular ORMs

- Laravel: Eloquent
- Symfony: Doctrine, Propel
- Rails: ActiveRecord
- Java: Hibernate

Summary

- ORMs are great for rapid application development
- ORMs abstract away SQL so you can work with your database in an OOP fashion
- Typically you need to understand SQL to use an ORM