Zaawansowane Techniki WWW
(HTML, CSS i JavaScript)

Dr inż. Marcin Zieliński

Środa 15:30 - 17:00 sala: A-1-04

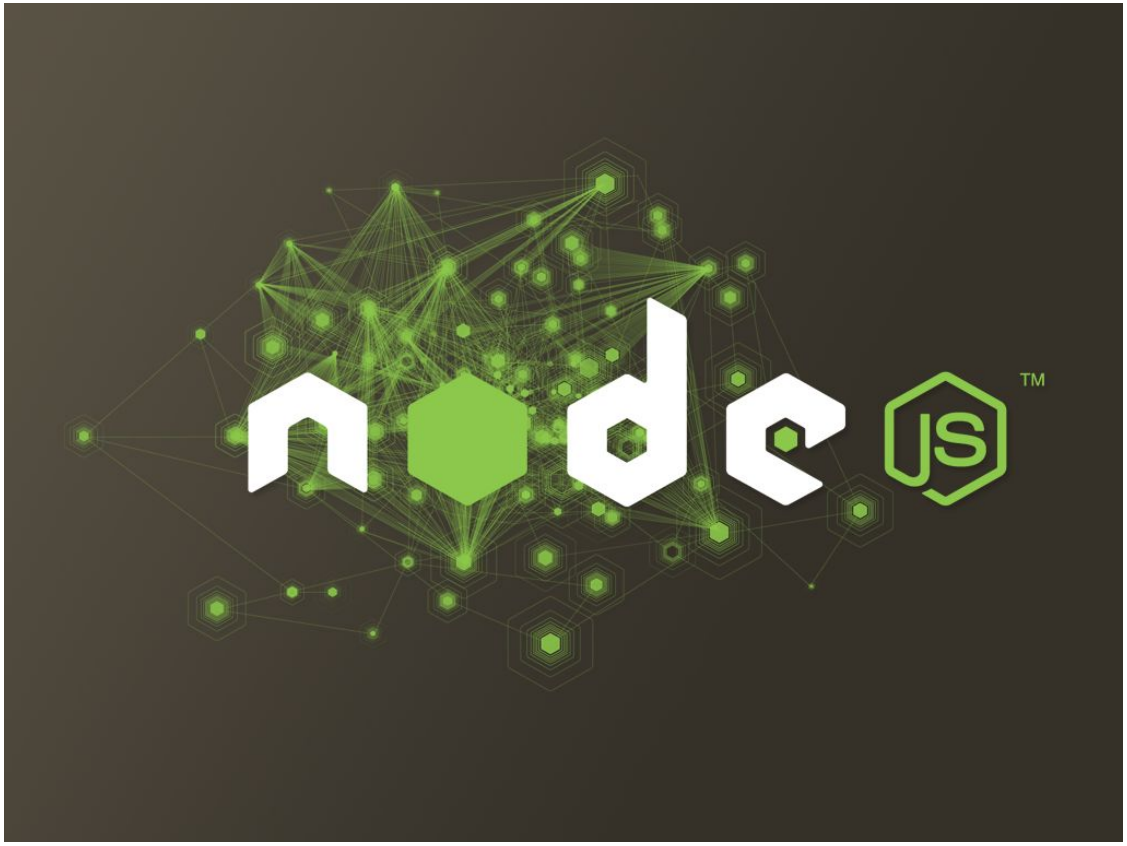
WYKŁAD 12

Wykład dla kierunku: **Informatyka Stosowana II rok**

Rok akademicki: **2015/2016 - semestr zimowy**

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Node.js

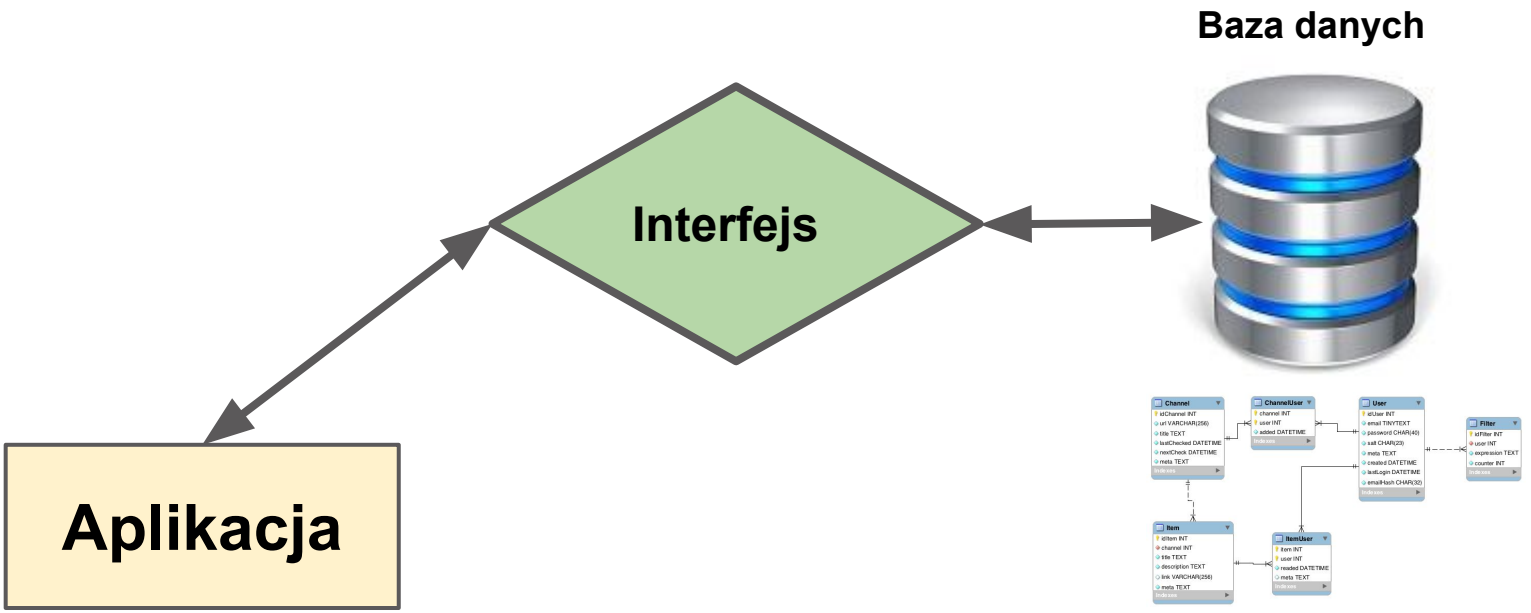


http://nodejs.org/

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

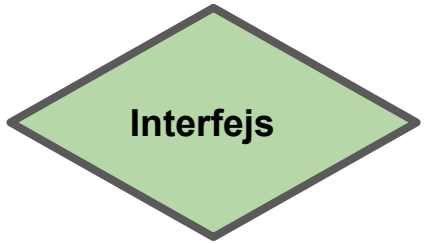
Bazy danych

W środowisku Node.js aby korzystać z baz danych konieczne jest użycie interfejsu umożliwiającego połączenie z serwerem baz danych i obsługujących komunikację pomiędzy aplikacją a serwerem baz danych.



Bazy danych

Od tego momentu mamy pełną możliwość wysyłania kwerend do serwera SQL w celu pobrania odpowiednich danych. Zapytanie formułujemy w postaci obiektu string:




```
var zapytanie = 'select * from users';
```

Następnie zapytanie musi zostać wysłane do serwera SQL, za pomocą metody .query(), która przyjmuje dwa argumenty: pierwszy zapytanie, drugi to wywołanie zwrotne w którym będziemy odbierać dane:

```
connection.query(zapytanie, function(err, rows) {
            res.render('participants',{dane: rows});
        });
```

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Bazy danych

Tablica “rows” zawiera tyle wierszy ile zostało zwróconych w wyniku zapytania SQL do bazy. Tablica ta może zostać w całości przekazana do widoku:


```
connection.query(zapytanie, function(err, rows) {
            res.render('participants',{dane: rows});
        });
```

W widoku możemy iterować po przekazanej tablicy i wyświetlać odpowiednia pola korzystając z notacji obiektowej:

```
#shortDescription
      ol
        each item in dane
          li
            b #{item.name} #{item.sname}
             | , #{item.affiliation}, #{item.country}
```

Bazy danych

Baza: na serwerze lokalnym (127.0.0.1)
Nazwa bazy: Personel
Tabela w bazie mySQL: dane

id	imie	nazwisko
1	Maciej	Adamczyk
2	Jan	Kowalski
3	Anna	Nowak

```
var connection = mysql.createConnection({
        host     : 'localhost',
        user     : 'root',
        password : 'test',
        database : 'Personel'
        });
connection.connect();
```

```
var zapytanie = 'select * from dane;
```

```
connection.query(zapytanie, function(err, rows) {
          console.log(rows[0].id + rows[0].imie + rows[0].nazwisko);
          console.log(rows[1].id + rows[1].imie + rows[1].nazwisko);
          console.log(rows[2].id + rows[2].imie + rows[2].nazwisko);
        });
```

UNIWERSYTET
JAGIELLOŃSKI
W KRAKOWIE

Bazy danych i ORM

Większość nowoczesnych aplikacji internetowych jest pisanych w językach obiektowych (Java, JavaScript, PHP, etc), gdzie logika biznesowa i model danych jest reprezentowany przez obiekty.

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Bazy danych i ORM

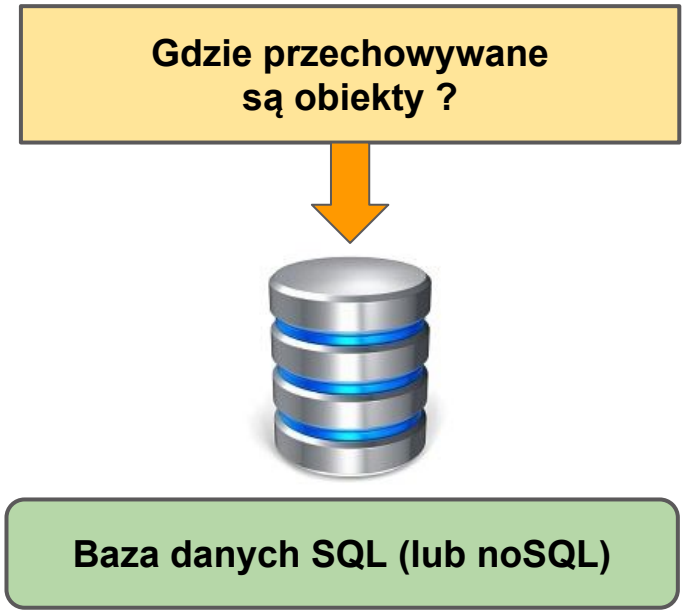
Większość nowoczesnych aplikacji internetowych jest pisanych w językach obiektowych (Java, JavaScript, PHP, etc), gdzie logika biznesowa i model danych jest reprezentowany przez obiekty.

Gdzie przechowywane są obiekty ?

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Bazy danych i ORM

Większość nowoczesnych aplikacji internetowych jest pisanych w językach obiektowych (Java, JavaScript, PHP, etc), gdzie logika biznesowa i model danych jest reprezentowany przez obiekty.



UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Bazy danych i ORM

Większość nowoczesnych aplikacji internetowych jest pisanych w językach obiektowych (Java, JavaScript, PHP, etc), gdzie logika biznesowa i model danych jest reprezentowany przez obiekty.

Gdzie przechowywane są obiekty ?

Baza danych SQL (lub noSQL)

Jak w bazie danych reprezentowane są dane ?

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

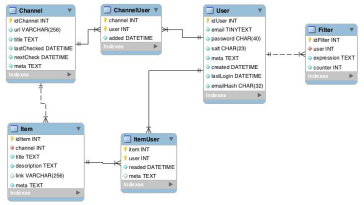
Bazy danych i ORM

Większość nowoczesnych aplikacji internetowych jest pisanych w językach obiektowych (Java, JavaScript, PHP, etc), gdzie logika biznesowa i model danych jest reprezentowany przez obiekty.

Gdzie przechowywane są obiekty ?

Baza danych SQL (lub noSQL)

Jak w bazie danych reprezentowane są dane ?



Dane przechowywane są w postaci relacji (lub dokumentów) połączonych zależnościami.

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

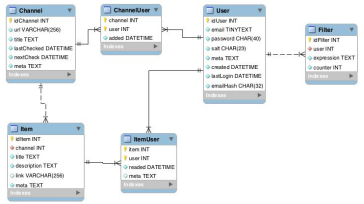
Bazy danych i ORM

Większość nowoczesnych aplikacji internetowych jest pisanych w językach obiektowych (Java, JavaScript, PHP, etc), gdzie logika biznesowa i model danych jest reprezentowany przez obiekty.

Gdzie przechowywane są obiekty ?

Baza danych SQL (lub noSQL)

Jak w bazie danych reprezentowane są dane ?



Dane przechowywane są w postaci relacji (lub dokumentów) połączonych zależnościami.

Zatem w bazie danych nie przechowujemy obiektów które dokładnie odpowiadają strukturze i modelowi danych aplikacji!!!

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Bazy danych i ORM

Pojawił się pomysł przechowywania danych w tzw. “obiektowych bazach danych”, czyli strukturach o takiej samej formie jak w aplikacji bez konieczności wykonywania dodatkowych operacji zmieniających format obiektowy na relacyjny i odwrotnie.

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Bazy danych i ORM

Pojawił się pomysł przechowywania danych w tzw. “obiektowych bazach danych”, czyli strukturach o takiej samej formie jak w aplikacji bez konieczności wykonywania dodatkowych operacji zmieniających format obiektowy na relacyjny i odwrotnie.

Pomysł ten jednak nie został wprowadzony w życie !

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Bazy danych i ORM

Pojawił się pomysł przechowywania danych w tzw. “obiektowych bazach danych”, czyli strukturach o takiej samej formie jak w aplikacji bez konieczności wykonywania dodatkowych operacji zmieniających format obiektowy na relacyjny i odwrotnie.

Pomysł ten jednak nie został wprowadzony w życie !

Powstało natomiast rozwiązanie alternatywne nazywane:
Mapowaniem Obiektowo-Relacyjnym (ORM) [*ang. Object-Relational Mapping*]

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Bazy danych i ORM

Pojawił się pomysł przechowywania danych w tzw. “obiektowych bazach danych”, czyli strukturach o takiej samej formie jak w aplikacji bez konieczności wykonywania dodatkowych operacji zmieniających format obiektowy na relacyjny i odwrotnie.

Pomysł ten jednak nie został wprowadzony w życie !

Powstało natomiast rozwiązanie alternatywne nazywane:
Mapowaniem Obiektowo-Relacyjnym (ORM) [*ang. Object-Relational Mapping*]

Do czego służy ORM ?

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Bazy danych i ORM

Pojawił się pomysł przechowywania danych w tzw. “obiektowych bazach danych”, czyli strukturach o takiej samej formie jak w aplikacji bez konieczności wykonywania dodatkowych operacji zmieniających format obiektowy na relacyjny i odwrotnie.

Pomysł ten jednak nie został wprowadzony w życie !

Powstało natomiast rozwiązanie alternatywne nazywane:
Mapowaniem Obiektowo-Relacyjnym (ORM) [*ang. Object-Relational Mapping*]

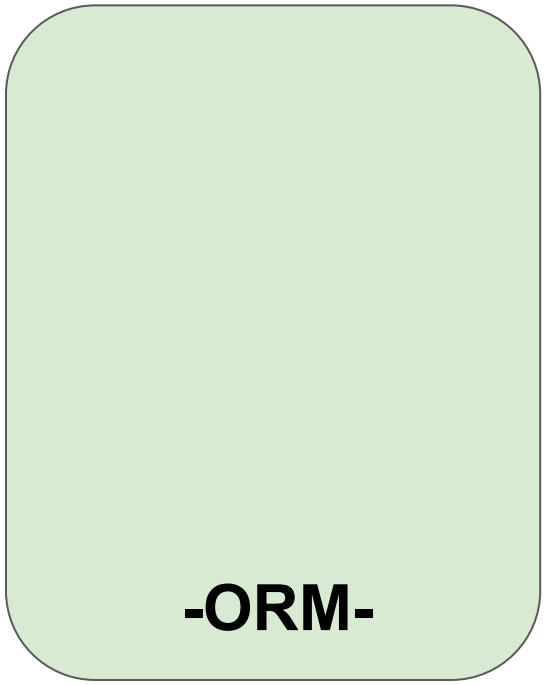
Do czego służy ORM ?

Implementuje warstwę pośredniczącą w zagadnieniu komunikacji baza danych - aplikacja, która umożliwia zamianę danych z postaci tabalarycznej (relacji) na reprezentację obiektową.

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

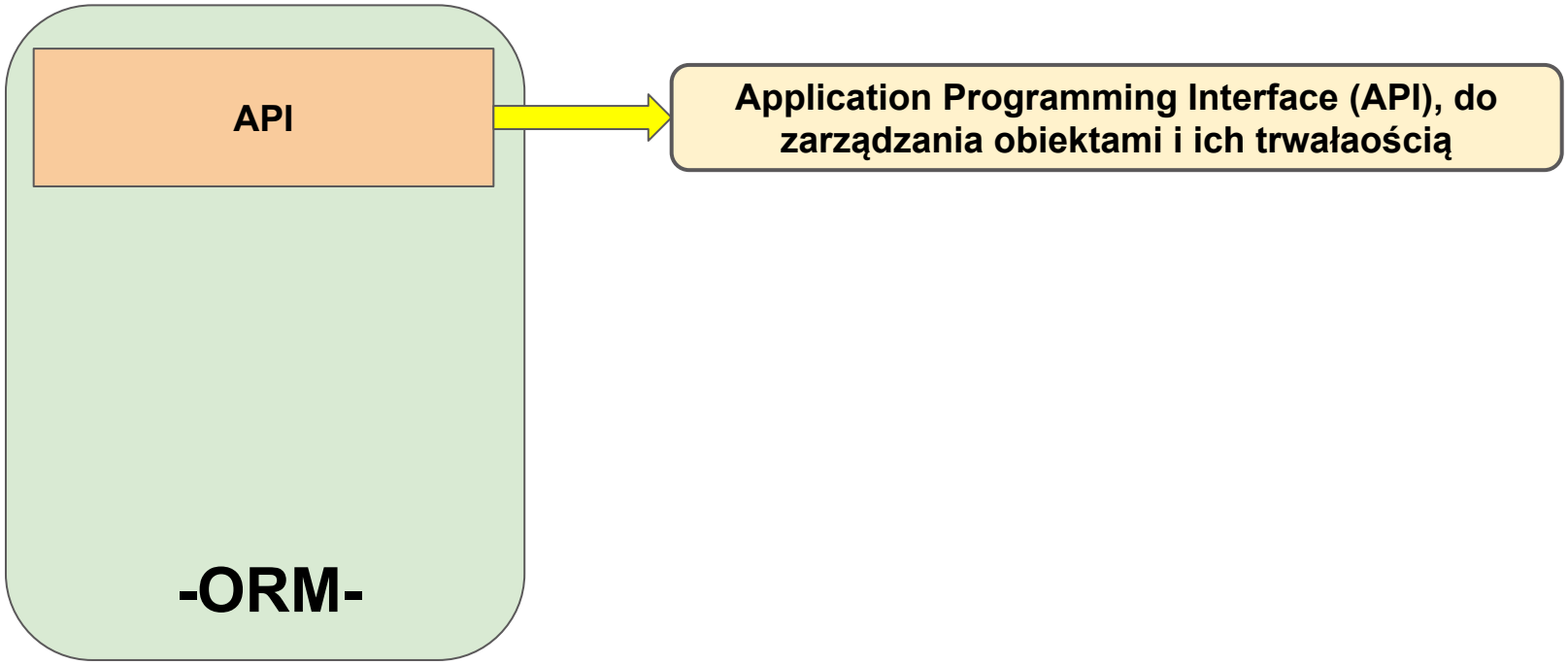
Bazy danych i ORM

Technologia ORM charakteryzuje się trzema elementami:



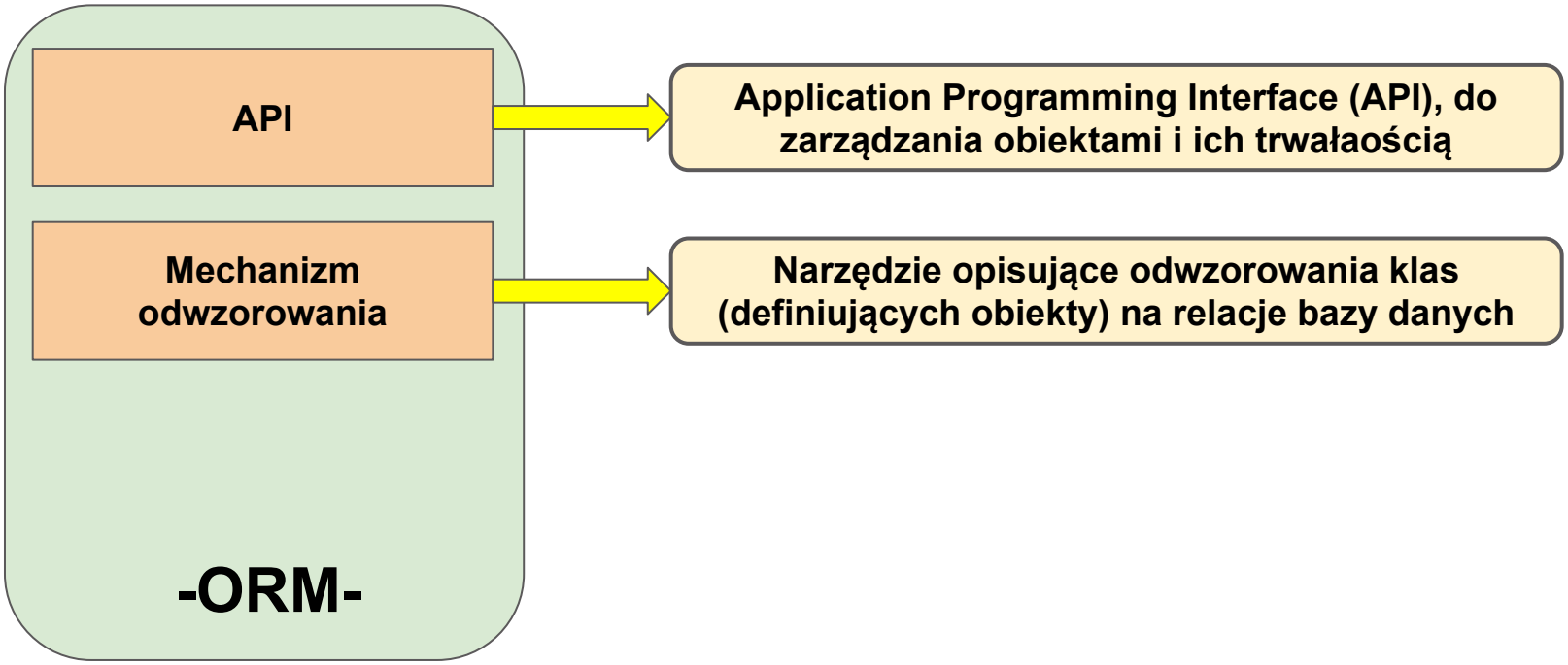
Bazy danych i ORM

Technologia ORM charakteryzuje się trzema elementami:



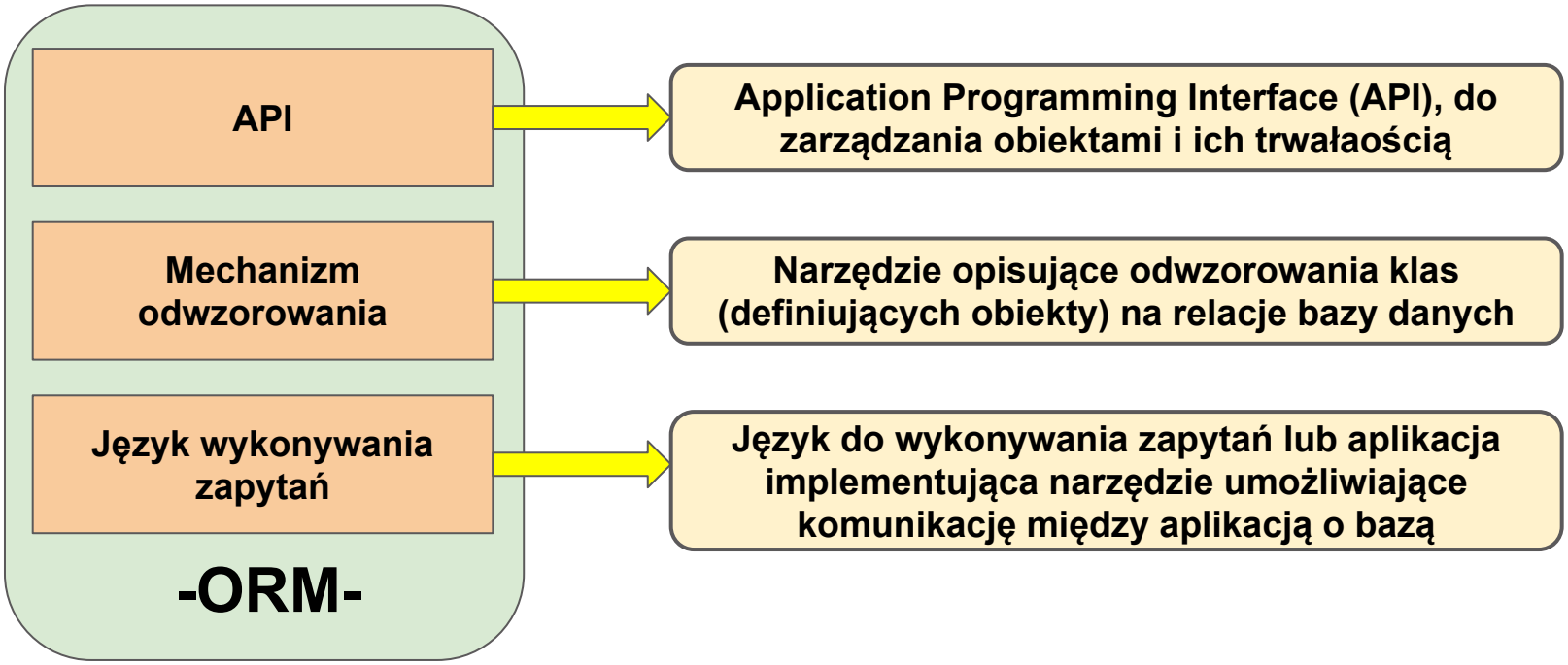
Bazy danych i ORM

Technologia ORM charakteryzuje się trzema elementami:



Bazy danych i ORM

Technologia ORM charakteryzuje się trzema elementami:



Bazy danych i ORM

Przykład - podejście klasyczne:

```
function zmienHaslo(_login, _haslo){
    var zapytanie = 'UPDATE users SET password'+_haslo+'WEHERE
login='_login;
    connection.query(zapytanie, function(err,rows){
        .....
    });
}
```

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Bazy danych i ORM

Przykład - podejście klasyczne:

```
function zmienHaslo(_login, _haslo){
    var zapytanie = 'UPDATE users SET password'+_haslo+'WEHERE
login='_login;
    connection.query(zapytanie, function(err,rows){
        .....
    });
}
```

Przykład - podejście obiektowe:

```
Uzytkownik u;
function zmienHaslo(_login, _haslo){
    u.SetHaslo(_login,_haslo);
     .....
}
```

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Bazy danych i ORM

Dla środowiska uruchomieniowego Node.js istnieje wiele aplikacji które wspierają model z zaimplementowną obsługą baz danych w podejściu ORM. Jednym z najpopularniejszych a zarazem łatwych w uzyciu jest pakiet:

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Bazy danych i ORM

Dla środowiska uruchomieniowego Node.js istnieje wiele aplikacji które wspierają model z zaimplementowną obsługą baz danych w podejściu ORM. Jednym z najpopularniejszych a zarazem łatwych w uzyciu jest pakiet:

Sequelize

www.sequelizejs.com

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Bazy danych i ORM

Dla środowiska uruchomieniowego Node.js istnieje wiele aplikacji które wspierają model z zaimplementowną obsługą baz danych w podejściu ORM. Jednym z najpopularniejszych a zarazem łatwych w uzyciu jest pakiet:

Sequelize

www.sequelizejs.com

Sequelize oferuje wsparcie dla najpopularniejszych relacyjnych baz danych:

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Bazy danych i ORM

Sequelize

www.sequelizejs.com

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Bazy danych i ORM

Sequelize

www.sequelizejs.com

Instalacja w środowisku Node.js przez repozytorium NPM:

```
~/node> npm install --save sequelize
```

Bazy danych i ORM

Sequelize

www.sequelizejs.com

Instalacja w środowisku Node.js przez repozytorium NPM:

```
~/node> npm install --save sequelize
```

oraz jeden z dodatkowych pakietów bazodanowych:

```
$ npm install --save pg pg-hstore //PostgreSQL
$ npm install --save mysql // MySQL
$ npm install --save sqlite3 // SQLite
$ npm install --save tedious // MSSQL
```

Bazy danych i ORM

Sequelize

www.sequelizejs.com

Instalacja w środowisku Node.js przez repozytorium NPM:

```
~/node> npm install --save sequelize
```

oraz jeden z dodatkowych pakietów bazodanowych:

```
$ npm install --save pg pg-hstore //PostgreSQL
$ npm install --save mysql // MySQL
$ npm install --save sqlite3 // SQLite
$ npm install --save tedious // MSSQL
```

w aplikacji:

```
var Sequelize = require('sequelize');
```

Bazy danych i ORM

Sequelize

Inicjalizacja połączenia z bazą danych dla Sequeliza:

```
var Sequelize = require('sequelize');
var sequelize = new Sequelize('database',    //wymagana
                              'uzytkownik', //wymagany
                              'haslo',
                              {'host','port'}
                              );
```

Bazy danych i ORM

Sequelize

Inicjalizacja połączenia z bazą danych dla Sequeliza:

```
var Sequelize = require('sequelize');
var sequelize = new Sequelize('database',    //wymagana
                              'uzytkownik', //wymagany
                              'haslo',
                              {'host','port'}
                              );
```

dość często (w przypadku lokalnej bazy danych) wymaga tylko:

```
var Sequelize = require('sequelize');
var sequelize = new Sequelize('database',    //wymagana
                              'uzytkownik', //wymagany
                              );
```

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Bazy danych i ORM

Sequelize

Co dalej ?

Bazy danych i ORM

Sequelize

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Co dalej ?

Musimy zdefiniować MODEL, czyli reprezentację bazy danych i ich własności

Bazy danych i ORM

Sequelize

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Co dalej ?

Musimy zdefiniować MODEL, czyli reprezentację bazy danych i ich własności

<u>Spróbujemy rozwiązać następujący problem i przedstawić go w postaci bazy danych:</u>

Chcemy utworzyć prosty system zarządzania projektami i zadaniami w tych projektach, tak aby do projektów można było przypisać zadania do zrealizowania.

Bazy danych i ORM

Sequelize

Co dalej ?

Musimy zdefiniować MODEL, czyli reprezentację bazy danych i ich własności

Spróbujemy rozwiązać następujący problem i przedstawić go w postaci bazy danych:

Chcemy utworzyć prosty system zarządzania projektami i zadaniami w tych projektach, tak aby do projektów można było przypisać zadania do zrealizowania.

Projekt
- tytul (STRING) - opis (TEXT) - utworzony (DATA)

Zadanie
- tytul (STRING) - status (BOOLEAN)

Bazy danych i ORM

Sequelize

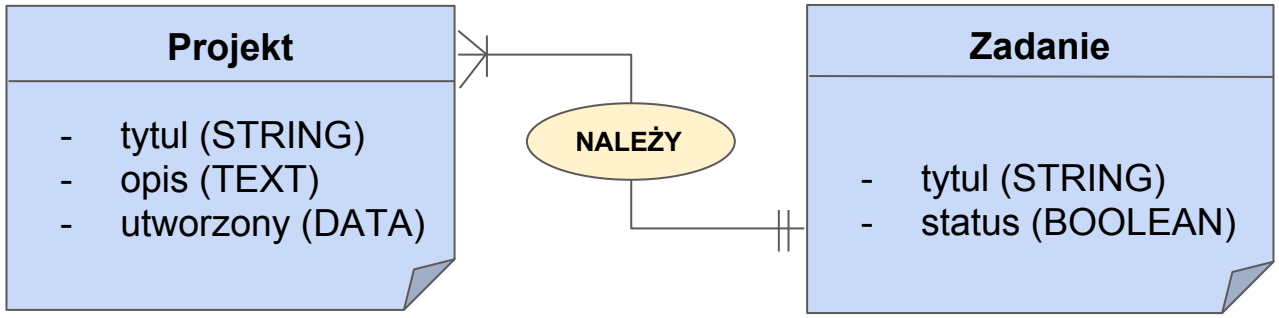
Projekt
- tytul (STRING) - opis (TEXT) - utworzony (DATA)

Zadanie
- tytul (STRING) - status (BOOLEAN)

Jakie zależności łączą te relacje ?

Bazy danych i ORM

Sequelize



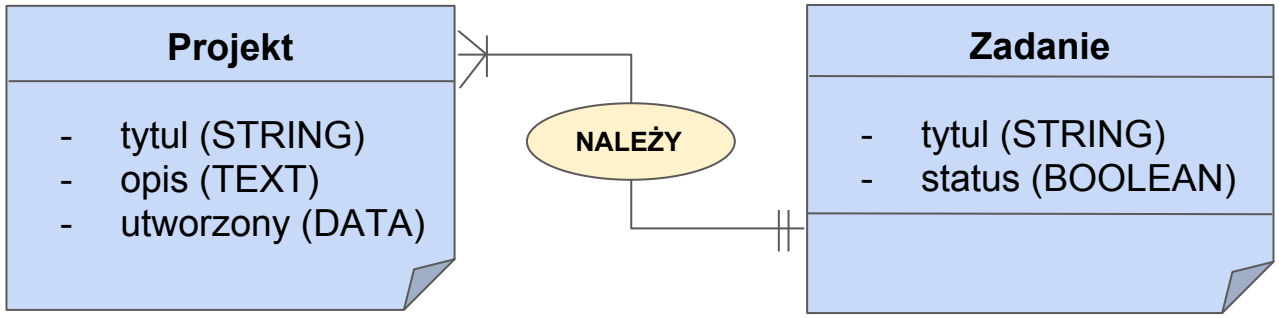
Jakie zależności łączą te relacje ?

Każde zadanie należy do jakiegoś projektu

Dla jednego projektu może być zdefiniowanych wiele zadań

Bazy danych i ORM

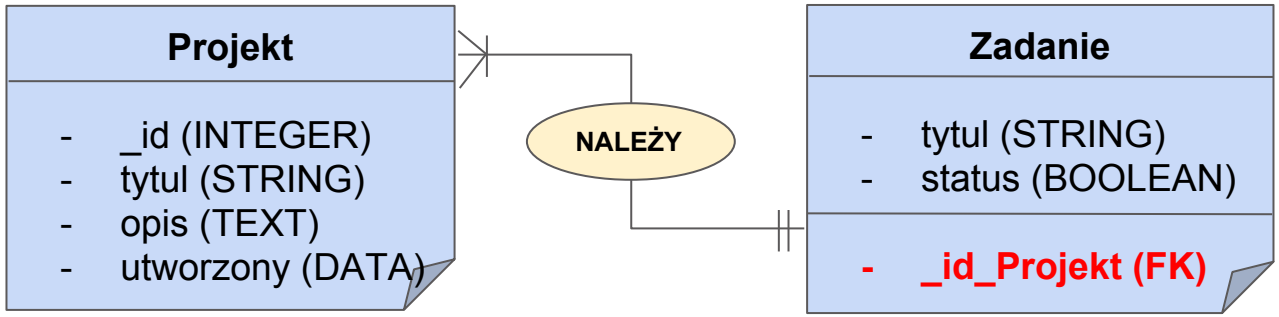
Sequelize



Fizycznie relacja ta oznacza że:

Bazy danych i ORM

Sequelize



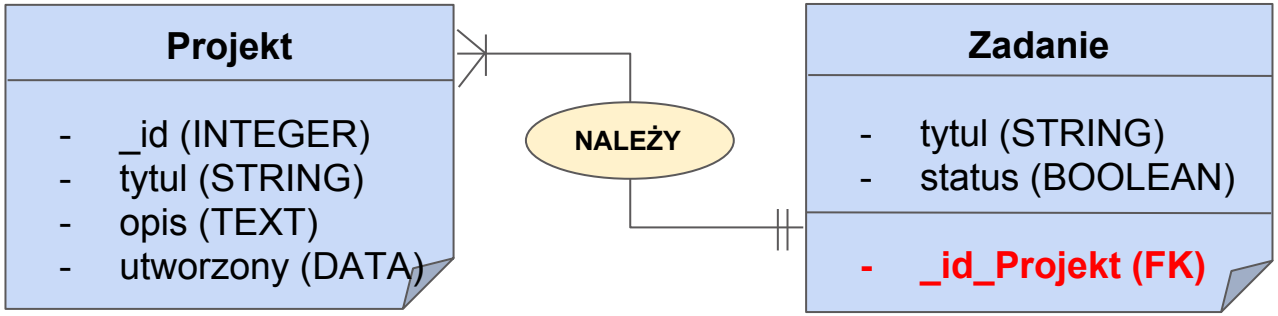
Fizycznie relacja ta oznacza że:

każdy rekord “Zadania” musi posiadać pole jednoznacznie wskazujące projekt do którego należy

projekt posiada wiele zadań i musi zapewniać dostęp do wszystkich zadań projektu.

Bazy danych i ORM

Sequelize



Fizycznie relacja ta oznacza że:

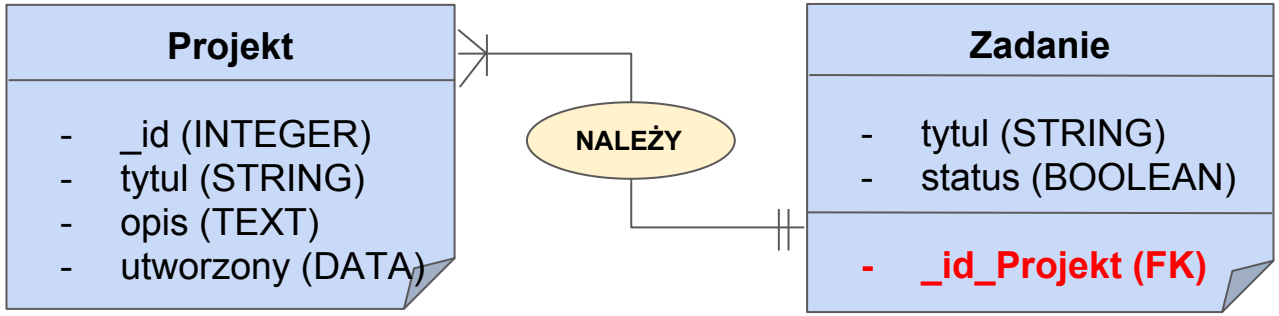
każdy rekord “Zadania” musi posiadać pole jednoznacznie wskazujące projekt do którego należy

projekt posiada wiele zadań i musi zapewniać dostęp do wszystkich zadań projektu.

REALIZACJA ZA POMOCĄ NARZĘDZIA SEQUELIZE:

Bazy danych i ORM

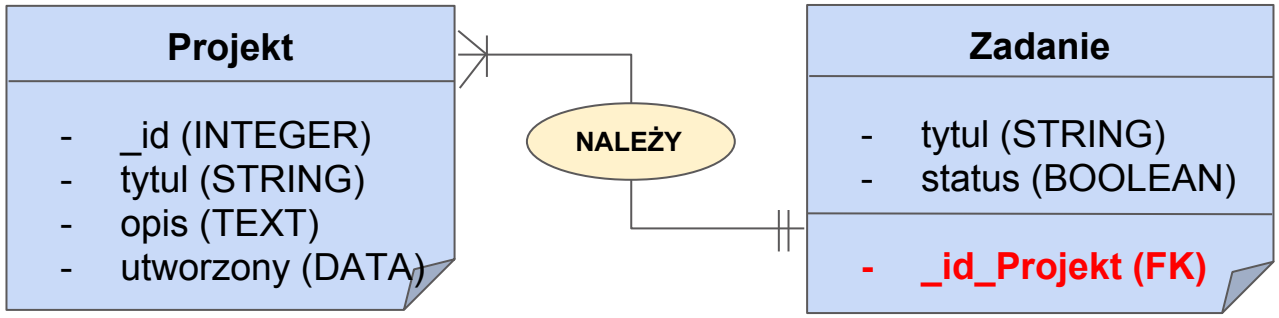
Sequelize



Na początek musimy zdefiniować MODEL który będzie reprezentował bazę danych:

Bazy danych i ORM

Sequelize



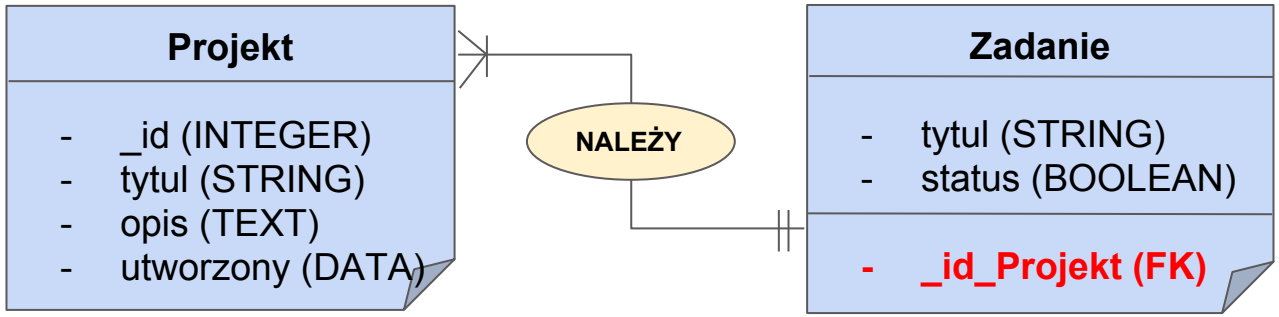
Na początek musimy zdefiniować MODEL który będzie reprezentował bazę danych:

```
var Projekt = sequelize.define('Projekt',{
    tytul: Sequelize.STRING,    // VARCHAR(255)
    opis: Sequelize.TEXT,
    utworzony: Sequelize.DATE   //DATETIME
});
```

```
sequelize.define( nazwa_modelu, {obiekt_wlasnosci} )
```

Bazy danych i ORM

Sequelize



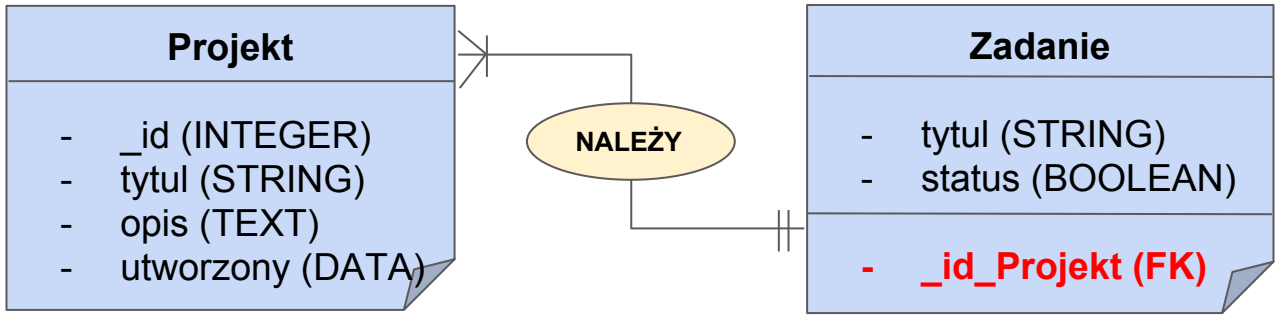
Na początek musimy zdefiniować MODEL który będzie reprezentował bazę danych:

```
var Zadanie = sequelize.define('Zadanie',{
    tytul: Sequelize.STRING,    // VARCHAR(255)
    status: Sequelize.BOLEAN    //TINYINT(1)
});
```

```
sequelize.define( nazwa_modelu, {obiekt_wlasnosci} )
```

Bazy danych i ORM

Sequelize

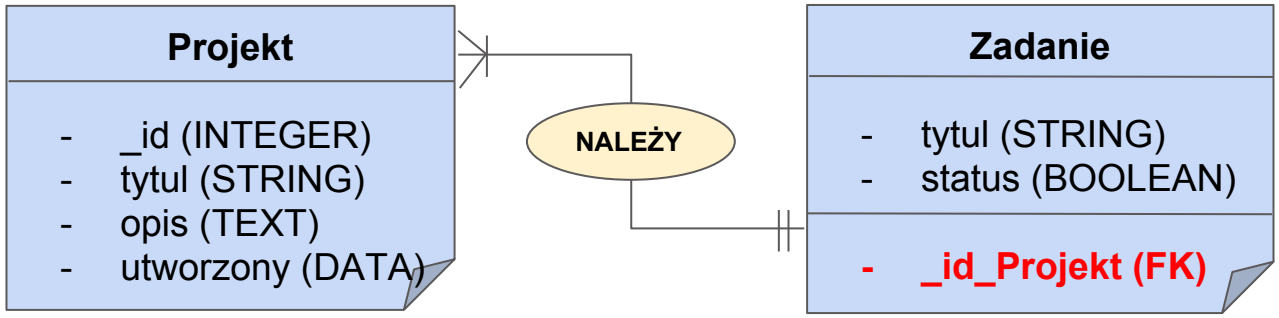


Zdefiniowanie zależności pomiędzy relacjami:

```
Zadanie.belongsTo( Projekt );
Projekt.hasMany( Zadanie );
```

Bazy danych i ORM

Sequelize



Zdefiniowanie zależności pomiędzy relacjami:

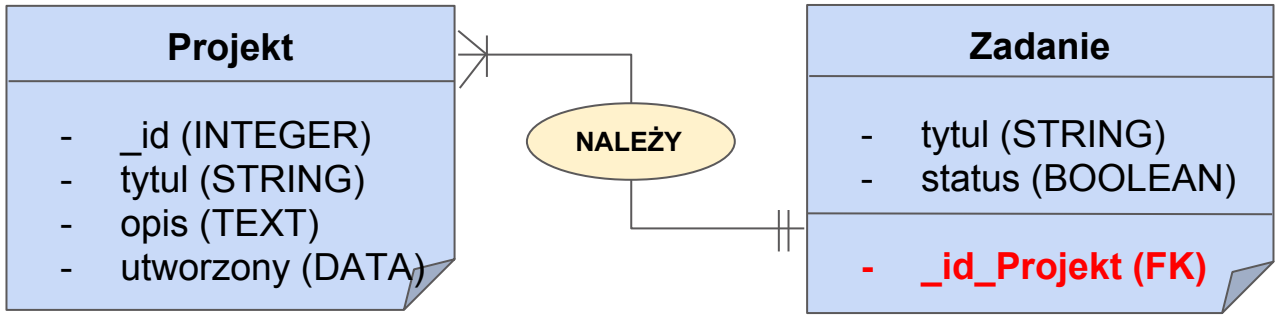
```
Zadanie.belongsTo( Projekt );
Projekt.hasMany( Zadanie );
```

Sequelize automatycznie definiuje powiązanie poiędzy relacjami ustawiając odpowiednie kolumny, klucze głowne i indeksy:

- **`.belongsTo()` powoduje, że każdy obiekt "Zadanie" posiada pole jednoznacznie wskazujące projekt do którego należy "_id_Projekt".**
- **`.hasMany()` powoduje, że jeśli wywołamy metodę wyszukującą spowoduje zwrócenie informacji o wszystkich zadaniach danego projektu.**

Bazy danych i ORM

Sequelize



Ostatnim krokiem utworzenia ORM dla bazy danych jest synchronizacja, która kontroluje czy model obiektowy zdefiniowany w aplikacji odpowiada strukturze bazy danych:

```
sequelize.sync();
```

W trakcie budowy aplikacji często zmienia się struktura tabel, daltego synchronizacja zapewnia że za każdym razem uruchamiana aplikacja będzie dopowiadała strukturze bazy danych.

Jak teraz odczytać dane z bazy danych ?

Bazy danych i ORM

Sequelize

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Jak teraz odczytać dane z bazy danych ?

```
router.get('/',function(req,res,next){
    Projekty.findAll()
        .success(function(projects){
            res.render('projekty',{projekty:projekt});
        })
        .error(next);
});
```

Bazy danych i ORM

Sequelize

Jak teraz odczytać dane z bazy danych ?

```
router.get('/',function(req,res,next){
    Projekty.findAll()
        .success(function(projects){
            res.render('projekty',{projekty:projekt});
        })
        .error(next);
});
```

`.findAll()` - zwraca wszystkie rekordy z relacji Projekty

Bazy danych i ORM

Sequelize

Jak teraz odczytać dane z bazy danych ?

```
router.get('/',function(req,res,next){
    Projekty.findAll()
        .success(function(projects){
            res.render('projekty',{projekty:projekt});
        })
        .error(next);
});
```

`.findAll()` - zwraca wszystkie rekordy z relacji Projekty

`.sucess()` - obsługuje zdarzenie w przypadku prawidłowej odpowiedzi z bazy danych

Bazy danych i ORM

Sequelize

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Jak teraz odczytać dane z bazy danych ?

```
router.get('/',function(req,res,next){
    Projekty.findAll()
        .success(function(projects){
            res.render('projekty',{projekty:projekt});
        })
        .error(next);
});
```

`.findAll()` - zwraca wszystkie rekordy z relacji Projekty

`.sucess()` - obsługuje zdarzenie w przypadku prawidłowej odpowiedzi z bazy danych

`.error()` - obsługuje zdarzenie w przypadku błędu

Bazy danych i ORM

Sequelize

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Jak odczytać wszystkie zadania dla danego projektu ?

```
router.get('/projekt/:id/tasks',function(req,res,next){
    Projekty.find(Number(req.params.id))
        .success(function(projekt){
            projekt.getZadania()
               .success(function(zadania){
                res.render('zadania',{projekt:projekt,
                                      zadania: zadania});
                })
        })
        .error(next);
});
```

`.getZadania()` - **metoda automatyczna powstała w trakcie tworzenia modelu**

Bazy danych i ORM

Sequelize

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Inne przydatne metody:

Bazy danych i ORM

Sequelize

Inne przydatne metody:

`.findById( [id] )` - **metoda poszukuje rekordu o konkretnym identyfikatorze**

Bazy danych i ORM

Sequelize

Inne przydatne metody:

`.findById( [id] )` - **metoda poszukuje rekordu o konkretnym identyfikatorze**

`.findOne( {where: { [warunki]} } )` - **metoda poszukuje rekordu spełniającego konkretne warunki podane jako obiekt**

Bazy danych i ORM

Sequelize

Inne przydatne metody:

`.findById( [id] )` - **metoda poszukuje rekordu o konkretnym identyfikatorze**

`.findOne( {where: { [warunki]} } )` - **metoda poszukuje rekordu spełniającego konkretne warunki podane jako obiekt**

```
Projekty.findOne({
   where: {
       tytul: 'projekt1'
       }
   })
   .then()
```

Bazy danych i ORM

Sequelize

Inne przydatne metody:

`.findById( [id] )` - metoda poszukuje rekordu o konkretnym identyfikatorze

`.findOne( {where: { [warunki]} } )` - metoda poszukuje rekordu spełniającego konkretne warunki podane jako obiekt

```
Projekty.findOne({
   where: {
       tytul: 'projekt1'
       }
   })
   .then()
```

```
SELECT *
FROM `Projekty`
WHERE (
  `Projekty`.`tytul` = 'Projekt1')
LIMIT 1;
```

Bazy danych i ORM

Sequelize

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Inne przydatne metody:

`.findById( [id] )` - metoda poszukuje rekordu o konkretnym identyfikatorze

`.findOne( {where: { [warunki]} } )` - metoda poszukuje rekordu spełniającego konkretne warunki podane jako obiekt

```
Projekty.findOne({
  where: {
    tytul: 'Projekt1',
    id: {
      $or: [
        [1,2,3],
        { $gt: 10 }
      ]
    }
  }
})
```

```
SELECT *
FROM `Projekty`
WHERE (
  `Projekty`.`tytul` = 'Projekt1'
  AND (`Projekty`.`id` IN (1,2,3) OR
`Projekty`.`id` > 10)
)
LIMIT 1;
```

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Bazy danych i ORM

Sequelize

Jak coś dodać do bazy danych INSERT

Bazy danych i ORM

Sequelize

Jak coś dodać do bazy danych INSERT

`.build( ).save( )` **- metoda “tworząca rekord” i zapisująca go w bazie danych**

Bazy danych i ORM

Sequelize

Jak coś dodać do bazy danych INSERT

`.build( ).save( )` - **metoda “tworząca rekord” i zapisująca go w bazie danych**

```
Zadanie.build({ tytul: 'foo', status: 0 })
   .save()
   .then(function(next) {
         //
    }).catch(function(error) {
   })
```

Bazy danych i ORM

Sequelize

Jak coś dodać do bazy danych INSERT

`.build( ).save( )` - **metoda "tworząca rekord" i zapisująca go w bazie danych**

```
Zadanie.build({ tytul: 'foo', status: 0 })
   .save()
   .then(function(next) {
         //
    }).catch(function(error) {
   })
```

```
INSERT INTO `Zadanie`
(`title`, `status`)
VALUES
(`foo`, 0)
```

Bazy danych i ORM

Sequelize

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Jak coś dodać do bazy danych INSERT

`.build( ).save( )` - metoda “tworząca rekord” i zapisująca go w bazie danych

```
Zadanie.build({ tytul: 'foo', status: 0 })
   .save()
   .then(function(next) {
         //
    }).catch(function(error) {
   })
```

```
INSERT INTO `Zadanie`
(`title`, `status`)
VALUES
(`foo`, 0)
```

Samo stworzenie zapytania metodą “build” jest nie wystarczające!

Stworzony rekord musi zostać zapisany do bazy danych metodą “save”

Bazy danych i ORM

Sequelize

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Usuwanie rekordów z bazy danych DELETE

Bazy danych i ORM

Sequelize

Usuwanie rekordów z bazy danych DELETE

`.destroy()` **- usuwa rekord z bazy danych spełniający określone warunki**

Bazy danych i ORM

Sequelize

Usuwanie rekordów z bazy danych DELETE

`.destroy()` - **usuwa rekord z bazy danych spełniający określone warunki**

```
Projekt.findById(4)
    .success(function(proj){
        proj.destroy()
            .success( )
            .error( )
        })
    .error();
```

```
DELETE FROM `Projekt`
WHERE `id` = 4
```

UNIWERSYTET
JAGIELLOŃSKI
W KRAKOWIE

KONIEC WYKŁADU 12