

Advanced Backend as a Service (BaaS)

Agenda

You will start building your Dianping mini-program

Consider reviews

Users can post reviews on restaurants. So relationship-wise:

- Each restaurant can be reviewed by many users, and
- Each user can review many restaurants

This is called a **many-to-many relationship**.

Many to Many Relationships

Excel++

Let's go further

wagon.xlsx

Rechercher dans la feuille

Accueil Mise en page Tableaux Graphiques SmartArt Formules

A1 fx

	A	B	C	D	E	F
1		first_name	last_name	social_security_number	age	
2		George	Abitbol	1 12 34 89 124 123	42	
3		Michel	Hazavanicus	1 94 91 12 123 492	25	
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						

patients

Mode Normal Prêt

wagon.xlsx

Rechercher dans la feuille

Accueil Mise en page Tableaux Graphiques SmartArt Formules

B5

	A	B	C	D	E	F
1		first_name	last_name	specialty		
2		Sigmund	Freud	Psychology		
3		Henri	Castafolte	Robotics		
4		John	Doe	Cardiag Surgery		
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						

patients doctors +

Mode Normal Prêt

Consultations?

- **One** doctor can have **many** patients
- **One** patient can see **many** doctors

wagon.xlsx

Rechercher dans la feuille

Accueil Mise en page Tableaux Graphiques SmartArt Formules

C1

	A	B	C	D	E	F	G
1	patient ?	doctor ?					
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							

patients doctors consultations

Mode Normal Prêt

wagon.xlsx

Rechercher dans la feuille

Accueil Mise en page Tableaux Graphiques SmartArt Formules

C1

	A	B	C	D	E	F	G
1	patient_id	doctor_id					
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							

patients doctors consultations

Mode Normal Prêt

Microsoft Excel window titled "wagon.xlsx" showing a table with patient data.

The table has columns labeled: id, first_name, last_name, social_security_number, and age.

The data rows are:

	A	B	C	D	E	F
	id	first_name	last_name	social_security_number	age	
1	1	George	Aabitbol	1 12 34 89 124 123	42	
2	2	Michel	Hazavanicus	1 94 91 12 123 492	25	
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						

The "patients" tab is selected at the bottom of the window.

wagon.xlsx

Rechercher dans la feuille

Accueil Mise en page Tableaux Graphiques SmartArt Formules

A4 fx

1	A	B	C	D	E	F
1	id	first_name	last_name	social_security_number	age	
2		1 George	Abitbol	1 12 34 89 124 123	42	
3		2 Michel	Hazavanicus	1 94 91 12 123 492	25	
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						

patients doctors consultations

Mode Normal Prêt

wagon.xlsx

Rechercher dans la feuille

Accueil Mise en page Tableaux Graphiques SmartArt Formules

A5

	A	B	C	D	E	F
1	id	first_name	last_name	specialty		
2		1 Sigmund	Freud	Psychology		
3		2 Henri	Castafolte	Robotics		
4		3 John	Doe	Cardiag Surgery		
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						

patients doctors consultations

Mode Normal Prêt

wagon.xlsx

Rechercher dans la feuille

Accueil Mise en page Tableaux Graphiques SmartArt Formules

C1

	A	B	C	D	E	F	G
1	patient_id	doctor_id					
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							

patients doctors consultations

Mode Normal Prêt

George Abitbol (**id = 1**) has seen Doctor John Doe (**id = 3**)

The screenshot shows a Microsoft Excel spreadsheet titled "wagon.xlsx". The interface includes a ribbon bar with tabs like Accueil, Mise en page, Tableaux, Graphiques, SmartArt, and Formules. Below the ribbon is a toolbar with various icons. The main area displays a table with three columns: A (patient_id), B (doctor_id), and C (consultation status). Row 1 contains the column headers. Row 2 contains the values 1 and 3 respectively. Row 3 is blank. Rows 4 through 14 are also blank. The status bar at the bottom indicates "Mode Normal" and "Prêt".

	A	B	C	D	E	F	G
1	patient_id	doctor_id					
2	1	3					
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							

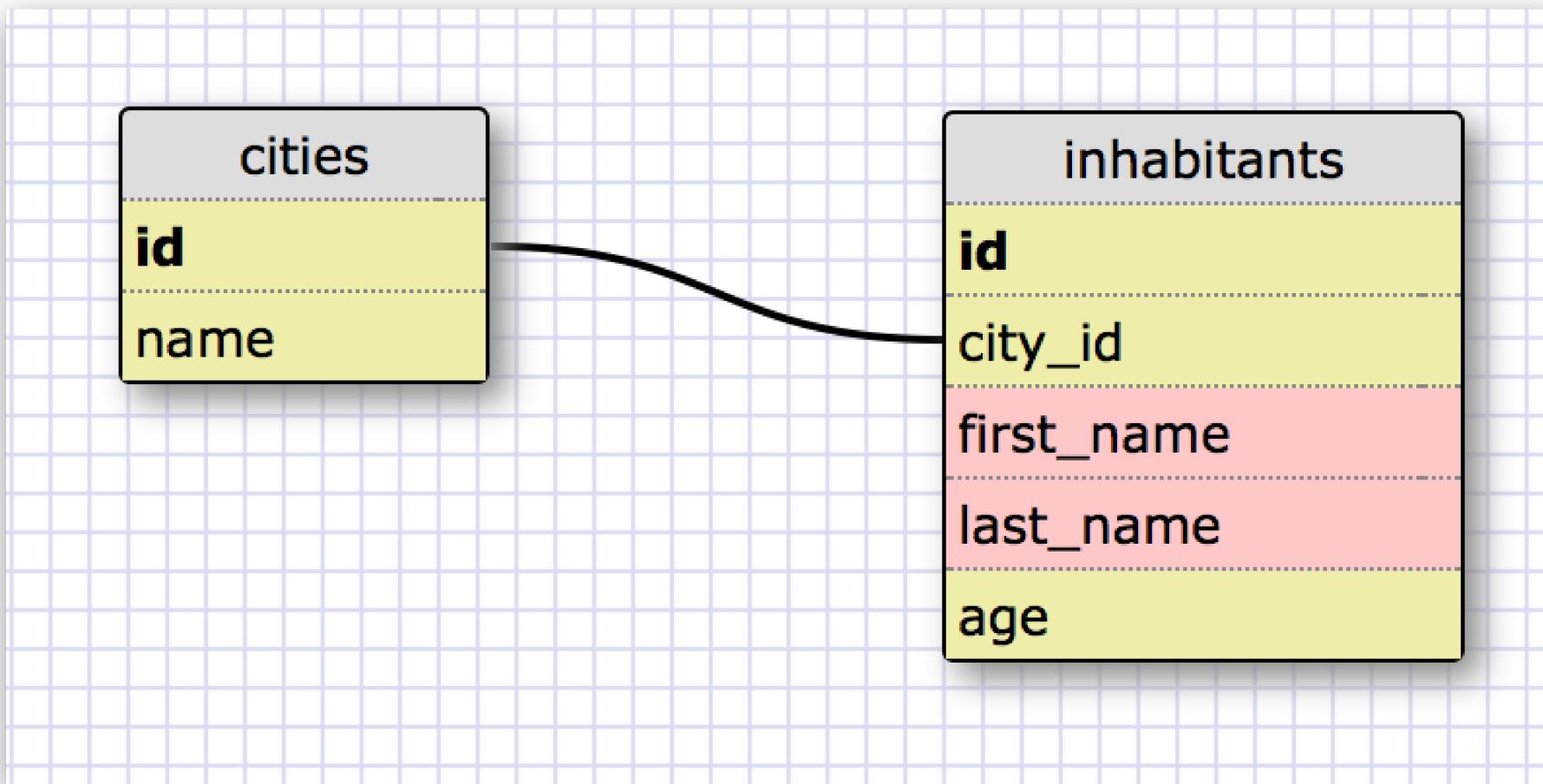
n:n relation (many to many)

A patient **has many** doctors and a doctor **has many** patients.

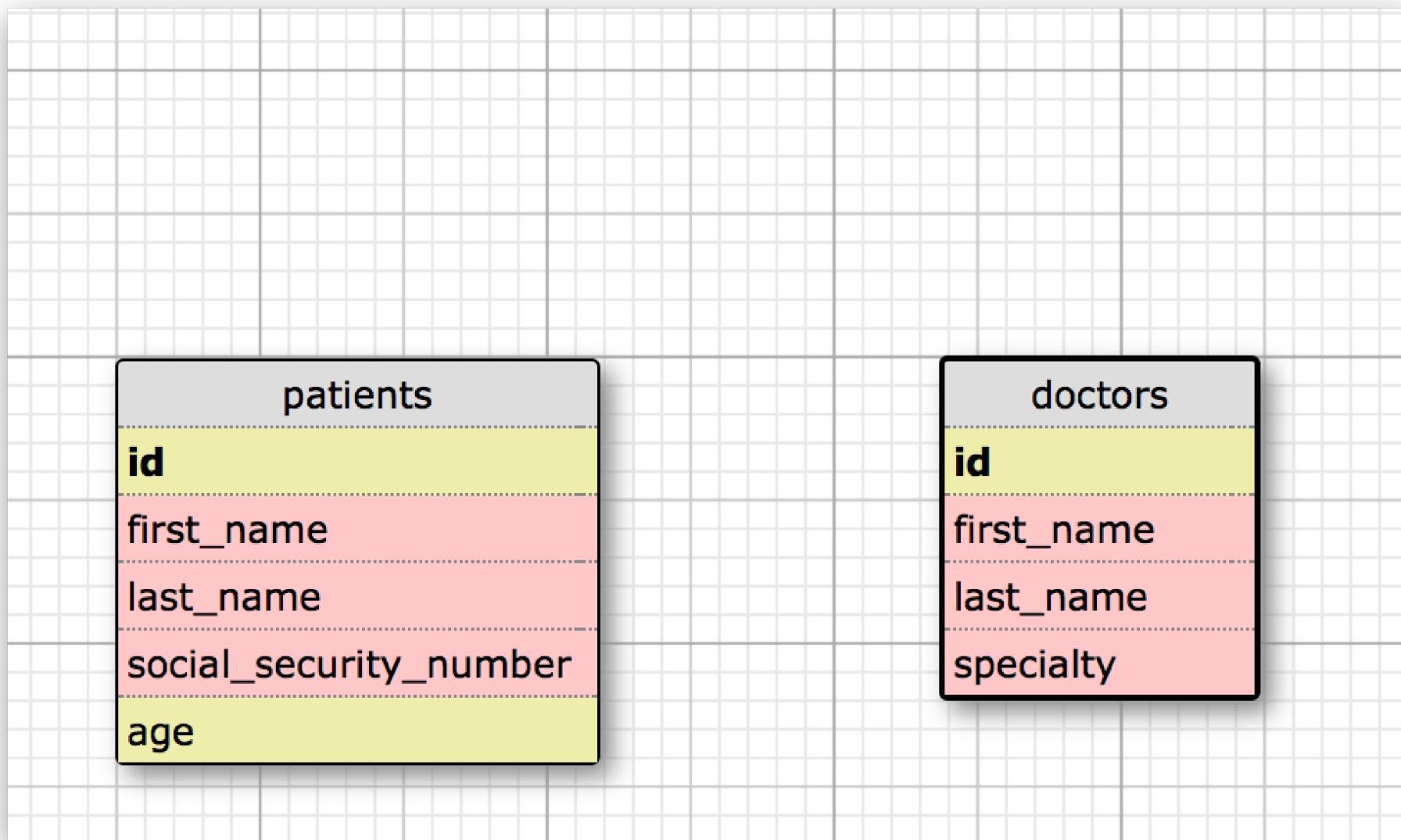
You can download this example: [consultations.xlsx](#)

Relational Database

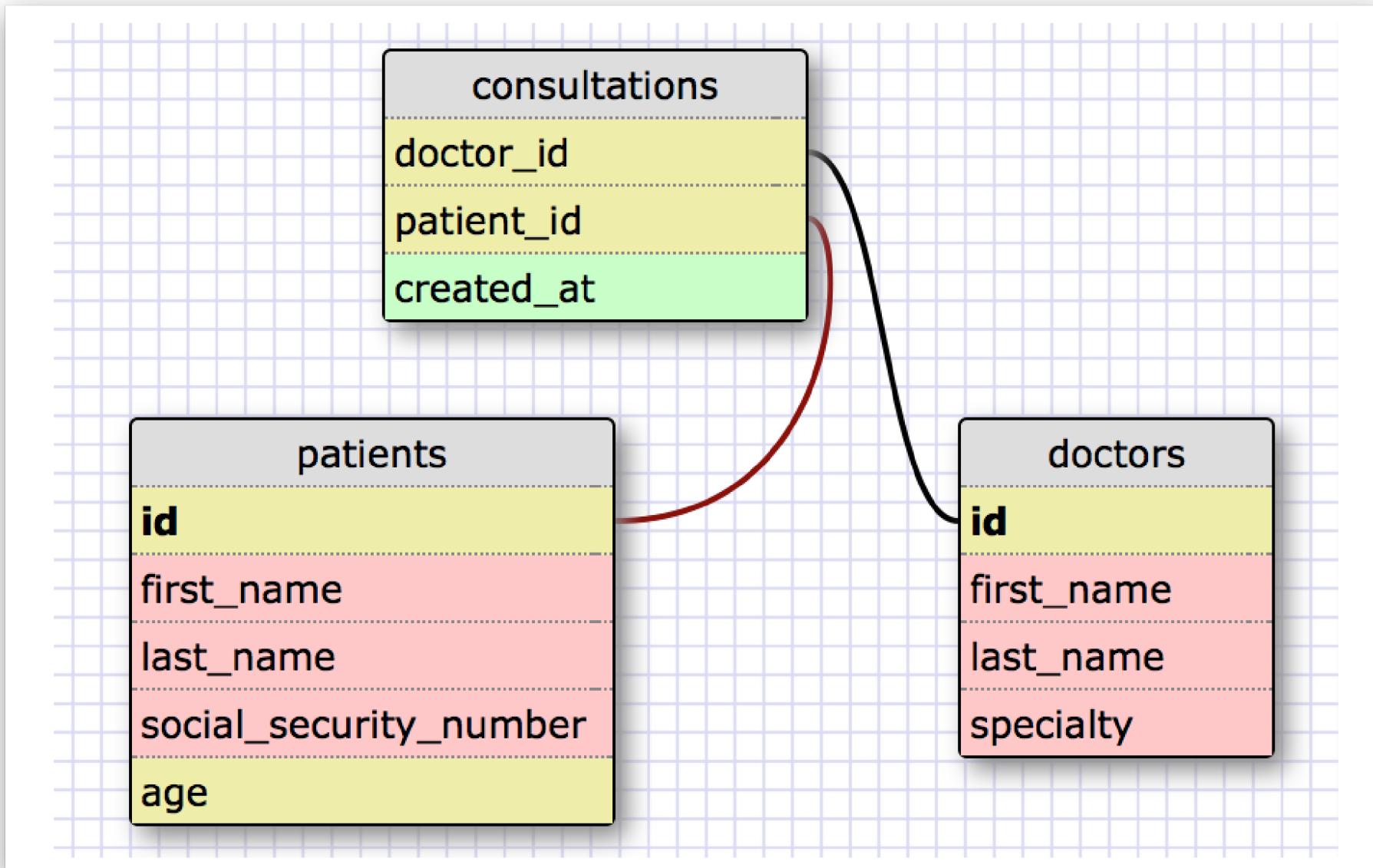
1:n



n:n ?



n:n



Vocabulary

- A schema is composed of **tables**.
- Each table has a set of **columns**.
- When inserting data in a table, you create a **record** in a new **row**.

DB Schema Composer

db.lewagon.com

Core User Journey

Notice how service products - like Eleme, Tmall, Airbnb - have the same basic types of functionality

Take **Dianping** for example!



User Login

As a user, I can sign up and log in

Restaurant Index

As a user, I can view all restaurants

The screenshot shows a mobile application interface for a restaurant search. At the top, there is a header with signal strength, time (7:39 AM), battery level (100%), and a back arrow. Below the header is a search bar with the placeholder "输入商户名、地点、找优惠". Underneath the search bar are four filter buttons: "附近", "美食", "智能排序", and "筛选". The main content area displays three restaurant entries, each with a thumbnail image, name, rating, price per person, location, distance, and some descriptive text or buttons.

- 舞渔日料美人胶原锅(太平洋店)** 排 买
★★★★★ 9834条 ¥137/人
日本料理 徐家汇 2.9km
徐汇区日本菜热门榜第2名 可停车
2019上海必吃餐厅
团 1元 火炙生牛肉寿司1份
- 金锚传菜(田尚坊商业中心店)** 订 外
★★★★★ 1496条 ¥134/人
上海本帮菜 漕河泾/田林 329m
有宝宝椅 有WIFI 有包厢 有大桌
“鸡肉鲜香滑嫩,回味无穷”
买 每满100减12元,每满100减10元
团 298元 4人套餐
- 头牌农家菜(徐汇店)** 外
★★★★★ 540条 ¥96/人
农家菜 漕河泾/田林 478m
有包厢 有景观位 可停车 有宝宝椅
“有梅干菜烧凤爪等独家菜”

Restaurant Page

As a user, I can view one restaurant



Review Index

As a user, I can read and write reviews for a restaurant

中国联通 7:40 AM 100%
舞渔日料美人胶原锅

洛洛智久 Lv8
10月3日
打分 ★★★★★

来上海旅行，好吃的日料店值得去攻略！
来上海旅行，可以吃到好吃的日料，无需飞霓虹，直接在国内就能吃到啦！
食材新鲜，品质上乘，服务周到，秉承一贯日系的理念！
推荐的三家日料店，有2家是必吃榜的店～
可以说这三家日料店，值得一去再去的，适...
推荐：握寿司·炙烤鹅肝 美人胶原锅（小份）

浏览4138 赞99+ 评论4

YUMMY~麦麦 Lv6
9月29日
打分 ★★★★★

三刷舞渔...托点评霸王餐的福...没有等太久...
#停车#太平洋百货停车比较复杂...要停在 大厦的停车场...我是活生生的绕场两周才找到停车位...而且停车场与商场不通...要走上...又
有减负 ~

写点评 收藏 分享 推荐到好物圈

We will cover more advanced user actions later on

Dianping Data Schema

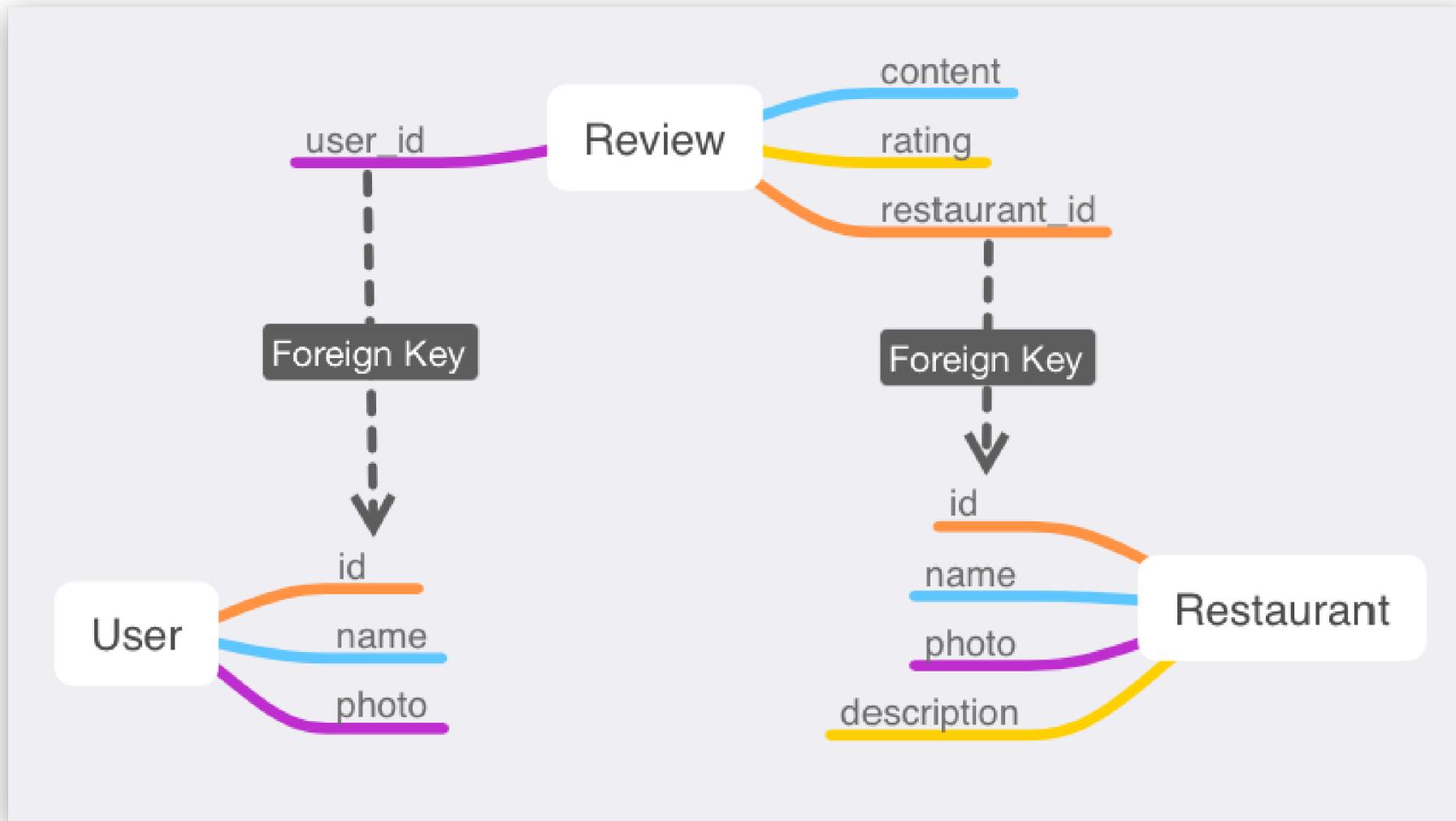
Review relationship is **many-to-many**



Therefore, review's relationship can be its own (joint) table



Schema



Live Code 1: Set up a unified front- and back-end 🚀

Your turn!

EXERCISE 1: SET UP YOUR FRONT AND BACKEND 💪

Live Code 2: All set? Let's build our database 

We only want users who have signed up and logged in to execute certain functions

Create table

Table name: restaurants

Data sheet notes: There will be less bugs if your comments are well written.

ACL ?

Set data table entry permission (new)

Public Logged-in user User group Private
需要先[创建临时用户](#)

Set the default read and write permissions for the row (query, modify, delete)

Public Read Write
需要先[创建临时用户](#)

Logged-in user Read Write

Creator Read Write

User group Read Write

[Cancel](#) [Create](#)

Your turn!

EXERCISE 2: CREATE YOUR DATABASE 💪

User login

Remember that the BaaS SDK allows user login, we can now store the user and allow sign up and log in

How does the **login** work?

1. User submits the username and password to **register** for an account and automatically **logs in** using the same credentials
2. If the user already has an account (with matching user name), will try to **log in** using those same credentials

Live Code 3: Sign-up and Login



Sign Up

Sign-up form

```
<!-- user.wxml -->
<form>
  <view>Username</view>
  <input name="username" placeholder="Enter Your Name"></input>
  <view>Password</view>
  <input name="password" type="password" placeholder="Enter Your Password"></input>
  <button formType="submit">Sign Up</button>
</form>
```

User Registration

Get form data

```
<!-- user.wxml -->
<form bindsubmit="onRegister">
    <!-- ... -->
</form>
```

```
// user.js
Page({
    //...
    onRegister: function(event) {
        // console.log(event)
        let username = event.detail.value.username
        let password = event.detail.value.password
    }
    //...
})
```

Store user to the database

```
// user.js
Page({
  onRegister: function(event) {
    // get user credentials from sign-up form
    wx.BaaS.auth.register({
      username: username,
      password: password
    }).then(dosomething).catch(function(err) {
      wx.showModal({
        title: 'Sign up failed',
        content: err.message
      })
    })
  }
})
```

User in the BaaS

Guide

Dev

- Data
- Content
- File
- User**
- Pay
- Engine
- Statistics
- Settings

Operations

- SMS
- Admin Panel

User list Group management

User source: Full platform WeChat nickname: User nickname User gender: User gender

Query Clear Advanced Search Expand

	id	Nickname	Gender	用户所在国家	用户所在省份	Operation
<input type="checkbox"/>	98688479807380		Other			Set groups Delete

1 datum < 1 > 20 / page Goto

User Profile Page

Once user has signed up or logged in, navigate to his/her profile page

Store user's data **locally** in `user.js` by getting the request data

```
// user.js
Page({
  data: {
    currentUser: null
  },
  onRegister: function(event) {
    //...
    let page = this

    wx.BaaS.auth.register({
      username: username,
      password: password
    }).then(page.getRequestData).catch(function(err) {
      // HError
    })
  },
  //...
})
```

Bind data to display user's profile

```
<!-- user.wxml -->
<view wx:if="{{currentUser}}">
  <text>{{currentUser._username}}</text>
</view>

<view wx:else>
  <!-- sign up -->
</view>
```

*Note: the code placement of user profile and sign-up form. Both blocks of code are written in `user.wxml`. Therefore, we need to set specific **conditions** for different actions*

Save and refresh!

Did you notice a problem? Users now have to register again. If users try to register with the same username, there's an error! Terrible UX 😬

Log In

Users who already have an account should be able to log in with the same credentials

To get the user currently logged in:

```
// user.js in onLoad function
wx.BaaS.auth.getCurrentUser().then(dosomething).catch(function(err) {
  wx.showModal({
    title: 'Error',
    content: err.message
  })
})
```

Login, similar to sign-up

```
wx.BaaS.auth.login({
  username: username,
  password: password
}).then(dosomething).catch(function(err) {
  wx.showModal({
    title: 'Sign in failed',
    content: err.message
  })
})
```

Now users can view both sign-up and login forms! Poor UX 😭

We need a data property to determine and change the state of the user - register or login

user.js

```
Page({
  data: {
    state: 'register', // or login
    //...
  },
  //...
})
```

user.wxml

```
<!-- profile page -->
<view wx:else>
  <view wx:if="{{state == 'register'}}">
    <!-- sign up -->
  </view>

  <view wx:if="{{state == 'login'}}">
    <!-- log in -->
  </view>
</view>
```

user.wxml

```
<!-- in signup -->
<view bindtap="changeState">Already have an account? Log in</view>

<!-- in login -->
<view bindtap="changeState">Need an account? Sign up</view>
```

user.js

```
changeState: function () {
  if (this.data.state == 'register') {
    this.setData({
      state: 'login'
    })
  } else {
    this.setData({
      state: 'register'
    })
  }
},
```

Log Out

Users can only log out after they've logged in. Simple enough!

user.wxml

```
<view wx:if="{{currentUser}}>
  <text>{{currentUser._username}}</text>
  <button bindtap="onLogout">Log Out</button>
</view>
```

user.js

```
onLogout: function() {
  wx.BaaS.auth.logout()
  this.setData({
    currentUser: null
  })
},
```

Once user logs out, return to sign-up or login form

Your turn!

EXERCISE 3: USER SIGN-UP & LOGIN 💪

Happy 2nd Day of BaaS 😊