

# طراحی پایگاه داده

■ گردآوری و تحلیل نیازهای داده ای

■ تفکر پیرامون دلیل ایجاد پایگاه داده

■ داده هایی که می خواهد ذخیره شوند

■ داده های پاسخی که می خواهند ارائه شوند.

■ مثال: باشگاه عضویت علاقه مندان فیلم. پایگاه داده اعضای باشگاه، باید:

■ اطلاعات اعضاء

■ جزئیات گردهمایی ها و شرکت کنندگان

# طراحی پایگاه داده

- گردآوری و تحلیل نیازهای داده ای
  - نوشتن لیستی از اطلاعاتی که می خواهید ذخیره کنید.
  - اطلاعات ارتباط با اعضاء
  - اطلاعات پرداخت حق عضویت
  - به طور خلاصه لیست زیر قابل تهیه است:
  - برای عضو: نام کامل، تاریخ تولد، آدرس، ایمیل، تاریخ عضویت
  - برای گردهمایی: تاریخ، مکان، شرکت کنندگان

# طراحی پایگاه داده

■ تجزیه منطقی داده ها

■ در نگاه اول به نظر می رسد تمام اطلاعات در یک جدول ذخیره شود.

| Name   | Date of Birth | Address                     | Email           | Date of Joining | Meeting Date | Location            | Did Member Attend? |
|--------|---------------|-----------------------------|-----------------|-----------------|--------------|---------------------|--------------------|
| Martin | Feb 27, 1972  | 1 The Avenue, NY            | martin@some.com | Jan 10, 2005    | Mar 30, 2005 | Lower West Side, NY | Y                  |
| Jane   | Dec 12, 1967  | 33 Some Road, Washington    | Jane@server.net | Jan 12, 2005    | Mar 30, 2005 | Lower West Side, NY | N                  |
| Kim    | May 22, 1980  | 19 The Road, New Townsville | kim@mail.com    | Jan 23, 2005    | Mar 30, 2005 | Lower West Side, NY | Y                  |

■ سوال: اگر گردهمایی ها بیش از یکی باشد!؟

# طراحی پایگاه داده

■ تجزیه منطقی داده ها

■ ساده ترین راه، افزودن رکورد جدید به ازای هر گردهمایی است.

| Name   | Date of Birth | Address                     | Email           | Date of Joining | Meeting Date   | Location             | Did Member Attend? |
|--------|---------------|-----------------------------|-----------------|-----------------|----------------|----------------------|--------------------|
| Martin | Feb 27, 1972  | 1 The Avenue, NY            | martin@some.com | Jan 10, 2005    | Mar 30, 2005   | Lower West Side, NY  | Y                  |
| Martin | Feb 27, 1972  | 1 The Avenue, NY            | martin@some.com | Jan 10, 2005    | April 28, 2005 | Lower North Side, NY | Y                  |
| Jane   | Dec 12, 1967  | 33 Some Road, Washington    | Jane@server.net | Jan 12, 2005    | Mar 30, 2005   | Lower West Side, NY  | N                  |
| Jane   | Dec 12, 1967  | 33 Some Road, Washington    | Jane@server.net | Jan 12, 2005    | April 28, 2005 | Upper North Side, NY | Y                  |
| Kim    | May 22, 1980  | 19 The Road, New Townsville | kim@mail.com    | Jan 23, 2005    | Mar 30, 2005   | Lower West Side, NY  | Y                  |
| Kim    | May 22, 1980  | 19 The Road, New Townsville | kim@mail.com    | Jan 23, 2005    | April 28, 2005 | Upper North Side, NY | Y                  |

# طراحی پایگاه داده

■ تجزیه منطقی داده ها

■ مشکلات!؟

■ تکرار داده ها

■ به روز رسانی داده ها

■ تعدد پاسخ برای استخراج اطلاعات اعضا

# طراحی پایگاه داده

■ تجزیه منطقی داده ها

■ راه حل دوم، اضافه کردن ستون ها به ازای هر گردهمایی

- Full name ■
- Date of birth ■
- Address ■
- Email address ■
- Date member joined club ■
- Meeting date 1 ■
- Location 1 ■
- Whether member attended meeting 1 ■
- Meeting date 2 ■
- Location 2 ■
- Whether member attended meeting 2 ■
- Meeting date 3 ■
- Location 3 ■
- Whether member attended meeting 3 ■

# طراحی پایگاه داده

- تجزیه منطقی داده ها
- بررسی راه حل دوم
- بخشی از تکرار داده ها حذف شده اما ساختار انعطاف پذیر نیست.
- برای ۱۰ گردهمایی نیاز به ۳۰ ستون داریم.
- هر بار گردهمایی اضافه شود، ساختار جدول تغییر می یابد.
- راه حل اصلی تجزیه داده به بخش های منطقی است.
- در این مثال اطلاعات پیرامون دو بخش مجزا است: اعضا و گردهمایی

# طراحی پایگاه داده

■ تجزیه منطقی داده ها

■ جدول اعضاء

| Name   | Date of Birth | Address                        | Email           | Date of Joining |
|--------|---------------|--------------------------------|-----------------|-----------------|
| Martin | Feb 27, 1972  | 1 The Avenue, NY               | martin@some.com | Jan 10, 2005    |
| Jane   | Dec 12, 1967  | 33 Some Road,<br>Washington    | Jane@server.net | Jan 12, 2005    |
| Kim    | May 22, 1980  | 19 The Road,<br>New Townsville | kim@mail.com    | Jan 23, 2005    |



# طراحی پایگاه داده

■ تجزیه منطقی داده ها

■ جدول گردهمایی ها

| Name   | Meeting Date   | Location             | Did Member Attend? |
|--------|----------------|----------------------|--------------------|
| Martin | Mar 30, 2005   | Lower West Side, NY  | Y                  |
| Martin | April 28, 2005 | Lower North Side, NY | Y                  |
| Jane   | Mar 30, 2005   | Lower West Side, NY  | N                  |
| Jane   | April 28, 2005 | Upper North Side, NY | Y                  |
| Kim    | Mar 30, 2005   | Lower West Side, NY  | Y                  |
| Kim    | April 28, 2005 | Upper North Side, NY | Y                  |

# طراحی پایگاه داده

■ انتخاب نوع داده صحیح

■ مثال: شماره تلفن؟!

■ عوامل موثر در انتخاب نوع داده:

■ کاربرد: محاسبات ریاضی، ساعت، متن

■ اندازه

■ ذخیره صحیح اطلاعات

■ کاراکترهای غیرانگلیسی

# طراحی پایگاه داده

■ انتخاب نوع داده صحیح

■ جدول اطلاعات اعضاء

- *Name* varchar (75)
- *DateOfBirth* date
- *Address* varchar (200)
- *Email* varchar (200)
- *DateOfJoining* date

# طراحی پایگاه داده

■ انتخاب نوع داده صحیح

■ جدول شرکت کنندگان در گردهمایی

- **Name** varchar (75)
- **MeetingDate** date
- **Location** varchar (200)
- **MemberAttended** char (1)

# طراحی پایگاه داده

■ با توضیحات عنوان شده درباره کلید اصلی و ویژگی های آن ، برای شناسایی منحصر به فرد اعضا از کلید اصلی *MemberId* در دو جدول استفاده می کنیم یعنی ساختار نهایی جدول اطلاعات اعضاء به صورت زیر است

- *MemberId* integer
- *Name* varchar (75)
- *DateOfBirth* date
- *Address* varchar (200)
- *Email* varchar (200)
- *DateOfJoining* date

## طراحی پایگاه داده

■ با دقت در ساختار جدول شرکت کنندگان در گردهمایی می توان در یافت که ترکیب *MeetingDate* و *MemberId* می تواند کلید اصلی باشد. برای ارتباط این جدول با جدول اطلاعات اعضاء از کلید خارجی استفاده می کنیم.

- *Name* varchar (75)
- *MeetingDate* date
- *Location* varchar (200)
- *MemberAttended* char (1)
- *MemberId* integer

# طراحی پایگاه داده

## ■ دستور ساخت جدول و تعریف کلیدها در MySQL

```
CREATE TABLE نام_جدول  
(  
    نوع_داده نام_ستون_یا_صفت_خاصه  
    .  
    .  
    .  
    PRIMARY KEY (نام_ستون_یا_ستون_های_کلید_اصلی) ,  
    FOREIGN KEY (نام_ستون_یا_ستون_های_کلید_خارجی) REFERENCES  
    (نام_ستون_یا_ستون_های_کلید_اصلی) نام_جدول_رجوع_شونده  
) ;
```

# طراحی پایگاه داده

■ دستور ساخت جدول و تعریف کلیدها در MySQL

■ جدول اطلاعات اعضاء

```
CREATE TABLE MemberDetails
(
    MemberId integer,
    FirstName nvarchar(50),
    LastName nvarchar(50),
    DateOfBirth date,
    Street varchar(100),
    City varchar(75),
    State varchar(75),
    ZipCode varchar(12),
    Email varchar(200),
    DateOfJoining date,
    PRIMARY KEY (MemberId)
);
```



# طراحی پایگاه داده

■ دستور ساخت جدول و تعریف کلیدها در MySQL

■ جدول شرکت کنندگان در گردهمایی

```
CREATE TABLE Attendance
(
    MeetingDate date,
    Location varchar(200),
    MemberAttended char(1),
    MemberId integer,
    PRIMARY KEY (MeetingDate, MemberId),
    FOREIGN KEY (MemberId) REFERENCES
    MemberDetails (MemberId)
);
```

# طراحی پایگاه داده

■ برای تکمیل پایگاه داده ی مثال، سه جدول اطلاعات فیلم، نوع (گروه یا ژانر) فیلم مورد علاقه اعضا و جدول توصیف طبقه فیلم ها نیز اضافه می کنیم

## ■ جدول فیلم ها

- |                                     |                    |
|-------------------------------------|--------------------|
| ■ <i>FilmId</i> integer             | شناسه فیلم         |
| ■ <i>FilmName</i> varchar (100)     | نام فیلم           |
| ■ <i>YearReleased</i> integer       | سال انتشار         |
| ■ <i>PlotSummary</i> varchar (2000) | خلاصه داستان       |
| ■ <i>AvailableOnDVD</i> char (1)    | انتشار نسخه فیزیکی |
| ■ <i>Rating</i> integer             | امتیاز             |
| ■ <i>CategoryId</i> integer         | شناسه طبقه         |

# طراحی پایگاه داده

## ■ جدول طبقه بندی فیلم

- *CategoryId* integer
- *Category* varchar (100)

شناسه طبقه

طبقه (ژانر)

## ■ جدول گروه فیلم مورد علاقه اعضا

- *CategoryId* integer
- *MemberId* integer

شناسه طبقه

شناسه عضو

# طراحی پایگاه داده

■ دستور ساخت جدول و تعریف کلیدها در MySQL

■ جدول طبقه بندی فیلم

```
CREATE TABLE Category
(
    CategoryId integer,
    Category varchar(100),
    PRIMARY KEY (CategoryId)
);
```

■ جدول گروه فیلم مورد علاقه اعضا

```
CREATE TABLE FavCategory
(
    CategoryId integer,
    MemberId integer,
    PRIMARY KEY (CategoryId, MemberId),
    FOREIGN KEY (CategoryId) REFERENCES Category (CategoryId),
    FOREIGN KEY (MemberId) REFERENCES MemberDetails(MemberId)
);
```

# طراحی پایگاه داده

■ دستور ساخت جدول و تعریف کلیدها در MySQL

■ جدول فیلم ها

```
CREATE TABLE Films
(
    FilmId integer,
    FilmName varchar(100),
    YearReleased integer,
    PlotSummary varchar(2000),
    AvailableOnDVD char(1),
    Rating integer,
    CategoryId integer,
    PRIMARY KEY (FilmId),
    FOREIGN KEY (CategoryId) REFERENCES Category
    (CategoryId)
);
```

# طراحی پایگاه داده

## ■ تمرین:

- گردهمایی ها در چندین مکان مختلف برگزار می شوند، که باعث افزونگی در جدول شرکت کنندگان می شود. چه تغییری می توان در ساختار جداول پایگاه داده ایجاد کرد؟
- دستورات SQL مورد نیاز در تمرین قبل را بنویسید و در عین حال جزئیات آدرس Location را به *Street*، *City* و *State* تقسیم کنید.

# ورود اطلاعات به پایگاه داده

## ■ دستور ورود داده ها در MySQL

INSERT INTO (نام ستون یا ستون ها) نام\_جدول VALUES (مقادیر) ;

## ■ مثال: ورود شناسه طبقه "۱" و نام طبقه "ترسناک"

```
INSERT INTO Category (CategoryId, Category) VALUES (1, 'Thriller');
```

```
INSERT INTO Category (CategoryId, Category) VALUES (2, 'Romance');
```

```
INSERT INTO Category (CategoryId, Category) VALUES (3, 'Horror');
```

```
INSERT INTO Category (CategoryId, Category) VALUES (4, 'War');
```

```
INSERT INTO Category (CategoryId, Category) VALUES (5, 'Sci-fi');
```

# ورود اطلاعات به پایگاه داده

■ سوال: خروجی دستورات زیر چیست؟

۱) `INSERT INTO Category (Category, CategoryId) VALUES ('Historical', 6);`

۲) `INSERT INTO Category VALUES (6, 'Historical');`

■ نکته:

■ اگر نام ستون ها عنوان شود، *RDBMS* بر اساس نوع داده مقادیر را نسبت می دهد

■ اگر نام ستون ها آورده نشود، *RDBMS* بر اساس ترتیب هنگام تعریف، مقادیر را نسبت می دهد



# ورود اطلاعات به پایگاه داده

■ مزیت حذف نام ستون ها در هنگام ورود اطلاعات

■ تایپ کمتر اطلاعات و صرفه جویی در زمان

■ عیب حذف نام ستون ها در هنگام ورود اطلاعات

■ متکی به حافظه بودن و عدم امکان تطبیق مقادیر با ستون مرتبط

■ مثال:

```
INSERT INTO MemberDetails
VALUES
( 1, 'Katie', 'Smith', '1977-01-09', 'Main Road',
'Townsville', 'Stateside', '123456',
'katie@mail.com', '2004-02-23');
```

# ورود اطلاعات به پایگاه داده

```
INSERT INTO MemberDetails
(
    MemberId,
    FirstName,
    LastName,
    DateOfBirth,
    Street,
    City,
    State,
    ZipCode,
    Email,
    DateOfJoining
)
VALUES
(2, 'Bob', 'Robson', '1987-01-09', 'Little Street',
'Big City', 'Small State', '34565', 'rob@mail.com',
'2004-03-13');
```

# ورود اطلاعات به پایگاه داده

■ تمرین ۱: اطلاعات سه عضو جدید زیر را در پایگاه داده وارد کنید.

Member ID: 7

First Name: John

Last Name: Jackson

Date of Birth: May 27, 1974

Street: Long Lane

City: Orange Town

State: New State

Zip Code: 88992

Email: jjackson@mailme.net

Date of Joining:

November 21, 2005

Member ID: 8

First Name: Jack

Last Name: Johnson

Date of Birth: June 9, 1945

Street: Main Street

City: Big City

State: Mega State

Zip Code: 34566

Email: jjohnson@me.com

Date of Joining:

June 2, 2005

Member ID: 9

First Name: Seymour

Last Name: Botts

Date of Birth: October 21,  
1956

Street: Long Lane

City: Windy Village

State: Golden State

Zip Code: 65422

Email:

Seymour@botts.org

Date of Joining:

July 17, 2005

■ تمرین ۲: با توجه به تغییرات جدید کشوری، تمام ساکنان Long Lane در Orange Town اکنون در Big City زندگی می کنند. پایگاه داده را به روز رسانی کنید.

# به روز رسانی اطلاعات به پایگاه داده

■ دستور به روز رسانی داده ها در MySQL

```
UPDATE نام_جدول  
SET مقدار ۲ = نام_ستون ۲ , مقدار ۱ = نام_ستون ۱  
WHERE شرط_یا_شرایط ;
```

■ مثال: دو شهر به نام های *Small State* و *Stateside* با هم ادغام و به نام *Mega State* تغییر نام دهید.

```
UPDATE MemberDetails  
SET State = 'Mega State'  
WHERE  
State = 'Small State'  
OR  
State = 'Stateside' ;
```

نکته: به روز رسانی ممکن است بیش از یک سطر را تغییر دهد.

# حذف اطلاعات به پایگاه داده

■ دستور حذف داده ها در MySQL

```
DELETE FROM نام_جدول  
WHERE شرط_یا_شرایط;
```

■ مثال:

```
DELETE FROM MemberDetails  
WHERE MemberId = 3;
```

**نکته:** حذف داده از یک جدول ممکن است نیازمند حذف در سایر جداول نیز باشد.