

طراحی پایگاه داده پیشرفته

■ نرمال سازی:

■ شامل تعدادی دستورالعمل است که به شما در ایجاد یک ساختار پایگاه داده خوب کمک می کند.

■ این دستورالعمل ها به تعدادی شکل نرمال (Normal Form) تقسیم می شوند که به طور کلی ۵ شکل نرمال وجود دارد.

■ **1NF: جدول زمانی نرمال خواهد بود که شرایط زیر را داشته باشد:**

■ نوع داده ها و اطلاعات در هر ستون یکسان باشد.

■ محتویات هر ستون یک مقدار مشخص باشد اتمیک باشد.

■ داده های تکراری وجود نداشته باشند.

■ هر سطر از جدول منحصر به فرد باشد یعنی بتوان کلید اصلی تعریف نمود.

طراحی پایگاه داده پیشرفته

■ **2NF:** یک جدول در فرم دوم نرمال است اگر شرایط زیر را داشته باشد:

■ در فرم اول نرمال باشد.

■ کلید ستون‌های غیر کلید وابستگی تابعی کامل به کلید ستون‌های کلید داشته باشد. (به عبارت دیگر هر ستون باید به یک کلید اصلی وابسته باشد)

■ **3NF:** یک رابطه در فرم سوم نرمال است اگر شرایط زیر را داشته باشد:

■ در فرم اول و دوم نرمال باشد

■ فاقد وابستگی‌های با واسطه باشد. (به عبارت دیگر هر ستون باید به طور مستقیم به ستون کلیدی وابسته باشد)

طراحی پایگاه داده پیشرفته

■ مثال: آیا رابطه زیر 1NF است؟ چرا؟

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

طراحی پایگاه داده پیشرفته

■ مثال: آیا رابطه زیر 2NF است؟ چرا؟

Field Name	Data Type	Notes
FilmId	integer	Primary key
FilmName	varchar(100)	
ActorId	integer	Primary key
ActorName	varchar(200)	
DateOfBirth	date	

طراحی پایگاه داده پیشرفته

■ مثال: آیا رابطه زیر 3NF است؟ چرا؟

Field Name	Data Type
MemberId	integer
FirstName	nvarchar (50)
LastName	nvarchar (50)
DateOfBirth	date
Street	varchar (100)
City	varchar (75)
State	varchar (75)
ZipCode	varchar (75)
Email	varchar (200)
DateOfJoining	date

طراحی پایگاه داده پیشرفته

- اطمینان از صحت داده ها با محدودیت ها (Constraint)
- برای اینکه در ورود داده به ستون ها محدودیت هایی قائل شویم تا صحت، سازگاری و عدم تکرار در آنها حفظ شود، از تعدادی دستور استفاده می کنیم.

- NOT NULL
- UNIQUE
- CHECK
- PRIMARY KEY
- FOREIGN KEY

طراحی پایگاه داده پیشرفته

■ اطمینان از صحت داده ها با محدودیت ها (Constraint)

■ جلوگیری از ورود داده NULL:

■ در هنگام ایجاد جدول:

```
CREATE TABLE MyTable  
(  
Column1 int NOT NULL,  
Column2 varchar(20) ,  
Column3 varchar(12) NOT NULL  
)
```

■ پس از ایجاد جدول:

```
ALTER TABLE MyTable  
ALTER COLUMN Column2 varchar(20) NOT NULL;
```

طراحی پایگاه داده پیشرفته

■ اطمینان از صحت داده ها با محدودیت ها (Constraint)

■ جلوگیری از ورود داده تکراری:

■ در هنگام ایجاد جدول:

```
CREATE TABLE MyUniqueTable  
(  
Column1 int,  
Column2 varchar(20) UNIQUE,  
Column3 varchar(12) UNIQUE  
);
```

■ اعمال محدودیت و نامگذاری آن:

```
ALTER TABLE YetAnotherTable  
ADD CONSTRAINT MyUniqueConstraint UNIQUE(Column2,  
Column3);
```


طراحی پایگاه داده پیشرفته

■ اطمینان از صحت داده ها با محدودیت ها (Constraint)

■ بررسی شرطی خاص در هنگام ورود داده:

■ در هنگام ایجاد جدول:

```
CREATE TABLE NamesAges  
(  
Name varchar(50) ,  
Age int CHECK (Age >= 0)  
);
```

■ اعمال محدودیت و نامگذاری آن:

```
ALTER TABLE Employee  
ADD CONSTRAINT HourlyLess CHECK (AvgMonthlyWage >  
HourlyRate);
```

طراحی پایگاه داده پیشرفته

■ اطمینان از صحت داده ها با محدودیت ها (Constraint)

■ ایجاد کلید اصلی:

■ در هنگام ایجاد جدول:

```
CREATE TABLE HolidayBookings  
(  
  CustomerId int PRIMARY KEY,  
  BookingId int,  
  Destination varchar(50)  
);
```

■ اعمال محدودیت و نامگذاری آن:

```
ALTER TABLE MoreHolidayBookings  
ADD CONSTRAINT more_holiday_pk PRIMARY KEY  
(CustomerId, BookingId);
```

طراحی پایگاه داده پیشرفته

■ اطمینان از صحت داده ها با محدودیت ها (Constraint)

■ ایجاد کلید خارجی:

■ در هنگام ایجاد جدول:

```
CREATE TABLE Attendance
(
  LocationId integer,
  MeetingDate date,
  MemberAttended char(1),
  MemberId integer,
  CONSTRAINT SomeTable_fk1
  FOREIGN KEY (LocationId)
  REFERENCES Location(LocationId)
);
```

طراحی پایگاه داده پیشرفته

■ اطمینان از صحت داده ها با محدودیت ها (Constraint)

■ ایجاد کلید خارجی:

■ پس از ایجاد جدول:

```
ALTER TABLE name_of_table_to_add_foreign_key
ADD CONSTRAINT name_of_foreign_key
FOREIGN KEY
(name_of_column_that_is_foreign_key_column)
REFERENCES
name_of_table_that_is_referenced(name_of_column_being_
referenced)
```

طراحی پایگاه داده پیشرفته

■ اطمینان از صحت داده ها با محدودیت ها (Constraint)

■ تسريع در نتايج با تعريف ايندكس

■ با تعريف ايندكس، موتور جستجوي پایگاه داده می تواند بازیابی را

سريع تر نمايد.

```
CREATE INDEX <index_name>  
ON <table_name> (<column_names>)
```

مثال:

```
CREATE INDEX member_name_index  
ON MemberDetails (FirstName, LastName);
```

نکته: با تعريف ايندكس، به صورت پيش فرض نتايج به طور صعودی مرتب می شوند.

طراحی پایگاه داده پیشرفته

■ اطمینان از صحت داده ها با محدودیت ها (Constraint)

■ تسریع در نتایج با تعریف ایندکس

■ می توان شاخص را به صورت یکتا نیز تعریف کرد و ترتیب صعودی و

نزولی را نیز تعیین نمود.

```
CREATE UNIQUE INDEX member_name_idx  
ON MemberDetails (LastName DESC, FirstName);
```

طراحی پایگاه داده پیشرفته

■ اصلاح طراحی پایگاه داده باشگاه فیلم

■ اصلاح جدول شرکت کنندگان در گردهمایی ها

■ ستونی برای اعلام تمایل عضو به شرکت نکردن یا شرکت کردن زائد است.

می توان تنها اطلاعات افرادی را که شرکت می کنند در این جدول وارد کرد

و ستون تمایل به شرکت را حذف نمود.

```
CREATE TABLE Attendance  
(  
    MeetingDate date NOT NULL,  
    LocationId integer NOT NULL,  
    MemberId integer NOT NULL  
);
```

طراحی پایگاه داده پیشرفته

■ اصلاح طراحی پایگاه داده باشگاه فیلم

■ اصلاح جدول اطلاعات اعضاء

```
CREATE TABLE MemberDetails
(
  MemberId integer NOT NULL PRIMARY KEY,
  FirstName vargraphic(50) NOT NULL,
  LastName vargraphic(50) NOT NULL,
  DateOfBirth date,
  Street varchar(100),
  City varchar(75),
  State varchar(75),
  ZipCode varchar(12),
  Email varchar(200),
  DateOfJoining date
);
```


طراحی پایگاه داده پیشرفته

■ اصلاح طراحی پایگاه داده باشگاه فیلم

■ اصلاح جدول فیلم ها

```
CREATE TABLE Films
(
    FilmId integer NOT NULL PRIMARY KEY,
    FilmName varchar(100) NOT NULL,
    YearReleased integer,
    PlotSummary varchar(2000),
    AvailableOnDVD char(1),
    Rating integer,
    CategoryId integer
);
```

طراحی پایگاه داده پیشرفته

■ اصلاح طراحی پایگاه داده باشگاه فیلم

■ اصلاح جدول طبقه بندی فیلم ها

```
CREATE TABLE Category
(
  CategoryId integer NOT NULL PRIMARY KEY,
  Category varchar(100)
);
```

■ اصلاح جدول طبقه بندی مورد علاقه

```
CREATE TABLE FavCategory
(
  CategoryId integer NOT NULL,
  MemberId integer NOT NULL,
  CONSTRAINT favcategory_pk PRIMARY KEY (CategoryId,
  MemberId)
);
```

طراحی پایگاه داده پیشرفته

■ اصلاح طراحی پایگاه داده باشگاه فیلم

■ اصلاح شرکت کنندگان در گردهمایی ها

```
CREATE TABLE Attendance
(
    MeetingDate date NOT NULL,
    LocationId integer NOT NULL,
    MemberId integer NOT NULL,
    CONSTRAINT attendance_pk PRIMARY KEY (LocationId,
    MemberId)
);
```

طراحی پایگاه داده پیشرفته

■ اصلاح طراحی پایگاه داده باشگاه فیلم

■ اصلاح جدول فیلم ها

```
CREATE TABLE Films
(
    FilmId integer NOT NULL,
    FilmName varchar(100) NOT NULL,
    YearReleased integer,
    PlotSummary varchar(2000),
    AvailableOnDVD char(1),
    Rating integer,
    CategoryId integer
);
```

■ ساخت ایندکس برای جدول اعضاء

```
CREATE UNIQUE INDEX member_name_idx
ON MemberDetails (LastName DESC, FirstName);
```

طراحی پایگاه داده پیشرفته

■ اصلاح طراحی پایگاه داده باشگاه فیلم

■ ساخت کلیدهای خارجی

```
ALTER TABLE Attendance
ADD CONSTRAINT attend_loc_fk
FOREIGN KEY (LocationId)
REFERENCES Location(LocationId);
```

```
ALTER TABLE Attendance
ADD CONSTRAINT attend_memdet_fk
FOREIGN KEY (MemberId)
REFERENCES MemberDetails(MemberId);
```

```
ALTER TABLE Films
ADD CONSTRAINT films_cat_fk
FOREIGN KEY (CategoryId)
REFERENCES Category(CategoryId);
```