

فصل ششم: آشنایی با یک زبان رابطه ای

زبان SQL

- زبان SQL (*Structured Query Language*)
 - یکی از زبان ها برای پایگاه داده های مدل رابطه ای
 - نسخه های متفاوتی از آن تا به امروز ارائه شده است
 - اولین نسخه *sql-86* در سال ۱۹۷۴
 - آخرین نسخه 14 مربوط به سال ۲۰۱۷
 - هر زبان پایگاه داده ها از ۳ بخش زیر تشکیل شده است:
 - زبان تعریف داده (*DDL-Data Declaration Language*)
 - زبان تغییر داده (*DML-Data Manipulation Language*)
 - زبان کنترل داده (*DCL-Data Control Language*)

آشنایی با پرس و جو ها در SQL

- استفاده عمده از SQL در قالب پرس و جو (Query) است.
- بخش "زبان تغییر داده (DML)" با پرس و جوها و تغییر داده ها تعامل دارد.
- SQL اجازه می دهد یک پرس و جو (به طور اساسی یک سوال) از پایگاه داده طرح کنید و پایگاه داده داده ای را ارائه می کند که به پرس و جوی شما پاسخ می دهد.

آشنایی با پرس و جو ها در SQL

- مثال: در پایگاه داده ای برای ذخیره جزئیات فروشندگان، فروش خودرو، انواع خودرو فروخته شده و غیره،
 - هر فروشنده در هر ماه چه تعداد خودرو فروخته است؟
 - هر فروشنده برای شرکت چه مبلغی در آمد داشته است؟
- می توانید در قالب پرس و جو (Query) به داده های پاسخ این سوالات دست یابید.

آشنایی با پرس و جو ها در SQL

■ اجزای پرس و جو در SQL

■ دستور (*statement*): یک دستورالعمل یا فرمان است.

■ مثال: "داده هایی را بده"

■ جزء (*clause*): شرایطی را برای دستور تعیین می کند.

■ مثال: به جای عبارت بالا می گوئیم "تنها داده های **فروش** را بده"

■ شرط (*statement*): محدودیت ها را تعیین می کند.

■ مثال: "تنها داده های فروش **در ماه شهریور** را بده"

نحوه نوشتن دستورات SQL (Syntax)

- شناسه (*identifier*): با استفاده از یک پایگاه داده، جدول یا نام فیلد چیزی را به طور منحصر به فرد مشخص می کند.
- مثال: اگر پایگاه داده ای با نام "MyDB" ایجاد کردید شناسه آن " MyDB " است.
- مثال: در دستور زیر *SalesPeople* مشخص می کند داده ها از چه جدولی انتخاب شوند، در نتیجه شناسه است.

```
SELECT PersonFirstName  
FROM SalesPeople;
```

نحوه نوشتن دستورات SQL (Syntax)

■ مقدار (*literal*): یک مقدار واقعی را مشخص می کند.

■ مثال: عدد "۱۲۰"، نام "علی"، تاریخ "۱۳۹۷/۰۹/۰۱"

■ مثال: لیست فروشندگان با نام کوچک `Bob`

```
SELECT PersonFirstName, PersonLastName  
FROM SalesPeople  
WHERE PersonFirstName = 'Bob' ;
```

■ در این مثال `Bob` در عملگر مقایسه یک مقدار است.

نحوه نوشتن دستورات SQL (Syntax)

- کلمه کلیدی (*keyword*): کلمه ای که معنای خاصی برای سیستم مدیریت پایگاه داده دارد.
- تعداد این کلمات ۱۸۵ کلمه است.
- از مهمترین آنها می توان به موارد زیر اشاره کرد:

<i>ADD</i>	<i>ALTER</i>	<i>AND</i>	<i>ANY</i>	<i>ASC</i>
<i>BETWEEN</i>	<i>CREATE</i>	<i>DELETE</i>	<i>DESC</i>	<i>DISTINCT</i>
<i>DROP</i>	<i>GROUP</i>	<i>HAVING</i>	<i>INSERT</i>	<i>JOIN</i>
<i>NULL</i>	<i>ORDER</i>	<i>PRIMARY</i>	<i>SELECT</i>	<i>TABLE</i>
<i>UPDATE</i>	<i>WHERE</i>			

ساخت و حذف پایگاه داده

■ برای ایجاد پایگاه داده از دستور زیر استفاده می کنیم:

`CREATE DATABASE` نام پایگاه داده

■ حداکثر طول نام ۱۲۸ کاراکتر است.

■ برای حذف پایگاه داده از دستور زیر استفاده می کنیم:

`DROP DATABASE` نام پایگاه داده

مقایسه انواع داده در RDBMS ها

<i>ANSI SQL</i>	<i>MS Access</i>	<i>SQL Server 2000</i>	<i>IBM DB2</i>	<i>MySQL</i>	<i>Oracle 10</i>
<i>Character</i>	<i>char</i>	<i>char</i>	<i>char</i>	<i>char</i>	<i>char</i>
<i>Character varying</i>	<i>varchar</i>	<i>varchar</i>	<i>varchar</i>	<i>varchar</i>	<i>varchar</i>
<i>National character</i>	<i>char</i>	<i>nchar</i>	<i>graphic</i>	<i>char</i>	<i>nchar</i>
<i>National character varying</i>	<i>varchar</i>	<i>nvarchar</i>	<i>vargraphic</i>	<i>varchar</i>	<i>nvarchar</i>
<i>Integer</i>	<i>number(long integer)</i>	<i>int</i>	<i>int</i>	<i>int</i>	<i>int</i>
<i>Smallint</i>	<i>number(integer)</i>	<i>smallint</i>	<i>smallint</i>	<i>smallint</i>	<i>smallint</i>
<i>Real</i>	<i>Number(real)</i>	<i>real</i>	<i>real</i>	<i>real</i>	<i>real</i>
<i>Decimal</i>	<i>Number(decimal)</i>	<i>decimal</i>	<i>decimal</i>	<i>decimal</i>	<i>decimal</i>
<i>Date</i>	<i>date</i>	<i>datetime</i>	<i>date</i>	<i>date</i>	<i>date</i>
<i>Time</i>	<i>time</i>	<i>datetime</i>	<i>time</i>	<i>time</i>	<i>date</i>

بررسی انواع داده

Char/nchar ■

- داده های متنی را ذخیره می کند.
- باید مشخص گردد که چه تعداد کاراکتر (حداکثر ظرفیت ۲۵۶) در آن ذخیره می کنید در غیر اینصورت یک کاراکتر ظرفیت دارد.
- در صورت استفاده به تعداد کمتر از ظرفیت تعیین شده با فضای خالی در سمت راست پر خواهد شد.
- مثال: (8) char فضا برای ذخیره ۸ کاراکتر را تخصیص می دهد و تقریبا ۸ بایت فضا اشغال می کند.

بررسی انواع داده

varchar/nvarchar ■

- مشابه کاراکتر، داده های متنی را ذخیره می کند با این تفاوت که حداکثر به میزان ظرفیت تعریف شده خواهد بود.
- مشخص می کنید حداکثر چه تعداد کاراکتر (حداکثر ۸۰۰۰ کاراکتر) در آن ذخیره کنید در غیر اینصورت یک کاراکتر ظرفیت دارد.
- در صورت استفاده به تعداد کمتر از ظرفیت تعیین شده با فضای خالی در سمت راست پر خواهد شد.
- مثال: `varchar (50)` به این معناست که داده در این ستون حداکثر ۵۰ کاراکتر طول دارد.

بررسی انواع داده

Text/text ■

- داده های متنی که بیش از ۸۰۰۰ کاراکتر هستند
- باید استفاده نشود زیرا کنار گذاشته شده و حذف خواهد شد.
- به جای آن از `varchar (max)` استفاده شود.

image ■

- مشابه نوع داده قبلی با این تفاوت که برای داده های باینری است.
- مشابه مورد قبل باید استفاده نشود زیرا کنار گذاشته شده و حذف خواهد شد.
- به جای آن از `varbinary (max)` استفاده شود.

بررسی انواع داده

int ■

■ یا ***integer*** برای نگهداری مقادیر عددی صحیح

■ بازه اعداد بین $-2,147,483,648$ تا $2,147,483,647$

bigint ■

■ مشابه نوع داده قبلی با این تفاوت که برای اعداد بزرگتر است.

■ بین $-9,223,372,036,854,775,808$ تا $9,223,372,036,854,775,807$

بررسی انواع داده

smallint ■

■ نگهداری مقادیر عددی صحیح کوچک

■ بازه اعداد بین ۳۲,۷۶۸ - تا ۳۲,۷۶۷

tinyint ■

■ نگهداری مقادیر عددی صحیح بسیار کوچک

■ بین ۰ تا ۲۵۵

■ مثال: ذخیره مقدار عددی متناظر با استان های کشور

بررسی انواع داده

decimal/numeric ■

■ هر دو برای نگهداری اعداد اعشاری با دقت و بازه یکسان داده ها

■ بازه اعداد بین $-10^{38} + 1$ تا $10^{38} - 1$

float ■

■ نگهداری اعداد اعشاری ممیز شناور

■ بازه اعداد بین $-1.79E+308$ تا $1.79E+308$

■ نکته: اعداد ۱۰۰٪ دقت ندارند. (به دلیل ذخیره سازی فیزیکی)

بررسی انواع داده

real ■

■ مشابه float برای نگهداری اعداد اعشاری بازه اعداد بین

$$10^{38} - 1 \text{ تا } -10^{38} + 1$$

money ■

■ نگهداری مقادیر اعشاری تا ۴ رقم اعشار

■ نماد واحد پول را ذخیره نمی کند.

■ بازه اعداد بین $-922,337,203,685,477.5808$ تا

بررسی انواع داده

smallmoney ■

■ مشابه *money* اما با بازه کوچکتر

■ بازه اعداد بین $-214,748.3648$ تا $214,748.3647$

date ■

■ نگهداری تاریخ

■ قالب *YYYY-MM-DD*

■ بازه بین *January 1, AD* تا *December 31, 9999*

بررسی انواع داده

datetime ■

■ مشابه *date* به همراه ساعت

■ بازه بین *January 1, 1753* تا *December 31, 9999*

■ اگر زمان مشخص نشود *۱۲:۰۰:۰۰* ذخیره شود.

datetime2 ■

■ تفاوت با قبلی دقت بالاتر (ذخیره *[hh:mm:ss[.nnnnnnnn]*)

■ بازه بین *January 1, AD* تا *December 31, 9999*

بررسی انواع داده

smalldatetime ■

■ مشابه *datetime* با بازه کوچکتر

■ بازه بین *January 1, 1900* تا *June 6, 2079*

datetimeoffset ■

■ تفاوت با *UTC* را اضافه می کند

■ ***YYYY-MM-DD hh:mm:ss[.nnnnnnnn] [+|-]hh:mm***

■ مثال: ***2014-03-24 13:00:00 +05:00***

بررسی انواع داده

time ■

■ نگهداری ساعت (۲۴ ساعته)

■ قالب *hh:mm:ss[.nnnnnnnn]*

hierarchyid ■

■ در *SQL Server 2008* به بعد قابل استفاده است

■ پیاده سازی داده سلسله مراتبی مثل ساختار شغلی کارکنان یک

شرکت

بررسی انواع داده

geometry ■

■ ذخیره اطلاعات جغرافیایی دو بعدی به روش هایی مانند نقطه

geography ■

■ ذخیره طول و عرض جغرافیایی

rowversion ■

■ یک عدد دودویی تولید شده توسط *SQL* را نگهداری می کند که

برای هر ردیف در پایگاه داده یکتاست.

بررسی انواع داده

uniqueidentifier ■

ذخیره شناسه یکتا سراسری (GUID) ■

binary ■

ذخیره اعداد دودویی ■

varbinary ■

مشابه قبل با این تفاوت که اندازه ستون فیزیکی به ازای هر ردیف ■

با توجه به مقدار ذخیره شده متفاوت است.

بررسی انواع داده

bit ■

■ ذخیره ۰ یا ۱ (معمولا برای صحیح و غلط)

xml ■

■ مخصوص ذخیره XML با قابلیت اجرای پرس و جو های خاص

■ از SQL Server 2005 به بعد اضافه شده است

ایجاد جدول

■ نحوه ساخت جدول

```
CREATE TABLE نام_جدول  
(  
نوع_داده_ستون نام_ستون  
)
```

■ مثال:

```
CREATE TABLE Train_Times  
(  
    start_location varchar(75),  
    destination varchar(75),  
    departs time,  
    arrives time  
) ;
```

تغییر جدول

■ افزودن ستون

```
ALTER TABLE نام_جدول  
ADD نوع_داده_ستون نام_فیلد
```

■ حذف ستون

```
ALTER TABLE نام_جدول  
DROP COLUMN نام_فیلد
```

تغییر جدول

- مثال: افزودن ستون *runs_at_weekend* با نوع داده *char(1)* به جدول *Train_Times*

```
ALTER TABLE Train_Times  
  ADD runs_at_weekend char(1);
```

- مثال: حذف ستون *runs_at_weekend* از جدول *Train_Times*

```
ALTER TABLE Train_Times  
  DROP COLUMN runs_at_weekend;
```

حذف جدول

■ نحوه حذف جدول

```
DROP TABLE Train_Times
```

■ مثال: حذف جدول *Train_Times*

```
DROP TABLE Train_Times
```