

## آشنایی با دستورات کار با پایگاه داده

### واحد کارسوم

پس از مطالعه این واحد کار از فراگیر انتظار می‌رود که:

- ۱ - با اصول ایجاد پایگاه داده در MySQL آشنا باشد.
- ۲ - بتواند یک جدول جدید در MySQL ایجاد کند.
- ۳ - بتواند یک جدول را در MySQL حذف کند.
- ۴ - ساختار جداول را اصلاح کند.
- ۵ - بتواند جدول‌ها را تغییر نام دهد.
- ۶ - بتواند یک رکورد و داده‌های جدید به جدول اضافه کند.
- ۷ - بتواند اطلاعات را از جداول حذف کند.
- ۸ - بتواند اطلاعات جدول‌های موجود را ویرایش و به‌روز رسانی کند.
- ۹ - بتواند اطلاعات جدول‌ها را انتخاب کند.

زمان (ساعت)	
عملی	نظری
۳۵	۱۷

## مقدمه

همان‌طور که می‌دانید سیستم‌های مدیریت پایگاه داده مختلفی وجود دارند که با توجه به نوع نیاز، اندازه سیستم، بودجه و ... می‌توان یکی از آن‌ها را برای پیاده‌سازی سیستم بانک اطلاعاتی محیط عملیاتی استفاده کرد، سیستم‌های مدیریت پایگاه داده معمولاً از یک زبان ساخت یافته استاندارد با نام SQL به‌طور مشترک استفاده می‌کنند.

## ۳-۱ آشنایی با SQL

- SQL مخفف عبارت Structured Query Language (زبان جستجویی ساخت یافته) است و به کاربر امکان اتصال و دسترسی به اطلاعات موجود در یک پایگاه داده را می‌دهد.
- زبان SQL قادر است تا برای یک پایگاه داده عمل جستجو و گزینش اطلاعات را انجام دهد و همچنین اطلاعات ذخیره شده در یک پایگاه داده را بازیابی، حذف، ذخیره، اضافه یا به روزرسانی کند.
- زبان SQL یک استاندارد بین‌المللی است.

### Table ۳-۱-۱

با مفهوم جدول در واحدکار دوم آشنا شده‌اید. جدول‌ها مهم‌ترین عناصر پایگاه داده هستند که برای ذخیره و نگهداری سازمان یافته اطلاعات مورد استفاده قرار می‌گیرند.

### Query ۳-۱-۲

از Query برای ایجاد یک پرسش و جستجو در جدول و استخراج اطلاعات مورد نظر استفاده می‌شود. نتایج حاصل از یک Query در یک جدول موقت که دارای فیلدها و سطرهای واجد شرایط جستجو است، نمایش داده می‌شود.

مثال: به عنوان مثال می‌خواهیم، مشخصات دانش‌آموزانی را که در ترم دوم سال تحصیلی ۸۵-۸۶ موفق به گذراندن درس پایگاه داده شده‌اند از جدول دانش‌آموزان استخراج کنیم. برای این منظور از یک Query استفاده می‌کنیم. عمده دستورات SQL برای ایجاد و مدیریت

Query استفاده می‌شوند. در مباحث بعدی، کار با یک Query را بررسی می‌کنیم.

## ۳-۱-۲ SQL Data Manipulation Language (DML) یا زبان دستکاری

### اطلاعات SQL

این بخش از دستورات SQL، شامل دستوراتی برای به روز کردن، ورود اطلاعات یا حذف رکوردها در محیط پایگاه داده است. عمده دستورات این بخش عبارتند از:

- Select**: برای استخراج اطلاعات از یک جدول بانک اطلاعاتی مورد استفاده قرار می‌گیرد.
- Update**: برای به روز کردن اطلاعات یک جدول بانک اطلاعاتی مورد استفاده قرار می‌گیرد.
- Delete**: برای حذف اطلاعات از یک جدول بانک اطلاعاتی مورد استفاده قرار می‌گیرد.
- Insert**: برای ورود اطلاعات به یک جدول بانک اطلاعاتی مورد استفاده قرار می‌گیرد.

## ۳-۱-۴ SQL Data Definition Language (DDL) یا زبان تعریف اطلاعات SQL

از این قسمت می‌توان برای ایجاد یا حذف جداول از بانک اطلاعاتی، تعریف اندیس‌ها، کلید واژه‌ها، ایجاد ارتباط و پیوند بین جداول یا ایجاد محدودیت‌های دسترسی استفاده کرد. عمده دستورات این بخش عبارتند از:

**Create Table**: برای ایجاد یک جدول جدید به کار می‌رود.

**Alter Table**: برای تغییر اطلاعات موجود در یک جدول به کار می‌رود.

**Drop Table**: برای حذف یک یا چند جدول از بانک اطلاعاتی به کار می‌رود.

**Create Index**: برای ایجاد اندیس یا کلید واژه در یک جدول به کار می‌رود.

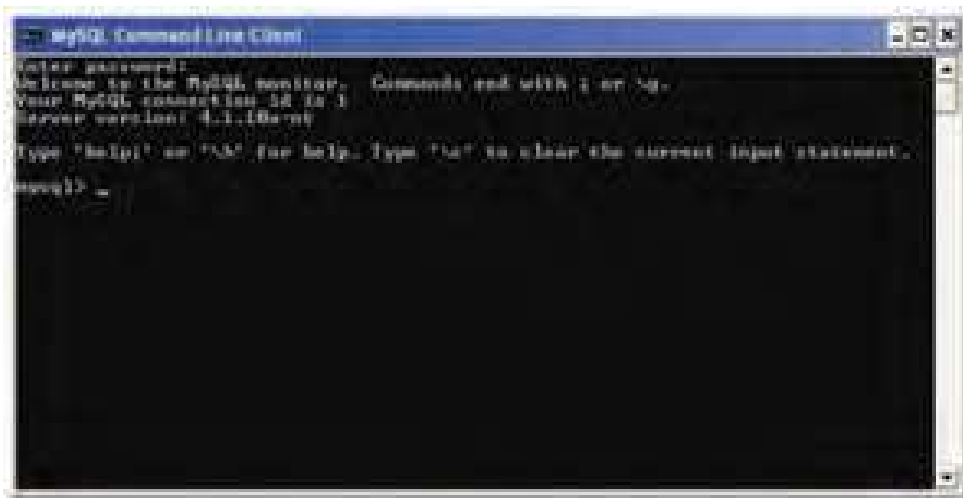
**Drop Index**: برای حذف اندیس یا کلید واژه در یک جدول به کار می‌رود.

**اندیس (Index)**: اندیس عبارت است از یک شماره که به هر یک از فیلدها در سطرهای یک جدول اختصاص داده می‌شود. اندیس‌ها از دید کاربر کاملاً مخفی هستند. استفاده از اندیس‌ها باعث می‌شود تا برنامه بتواند مقادیر سطرهای مختلف را بر حسب مقدار یک فیلد و بر حسب شماره اندیس آن‌ها از کم به زیاد یا برعکس مرتب کند و در عملیات جستجو باعث بالا رفتن سرعت جستجو می‌شود.

برای اجرای دستورات SQL و مشاهده نتیجه آن باید از یک سیستم مدیریت پایگاه داده استفاده کرد. سیستم مدیریت پایگاه داده‌ای که در این کتاب استفاده شده است، MySQL می‌باشد.

**نکته** در انتهای هر دستور SQL باید از علامت ; استفاده شود.

ابتدا از نصب MySQL روی سیستم خود مطمئن شوید، با نصب MySQL در واحدکار اول آشنا شده‌اید. سپس تمام دستورات را از طریق command line اجرا کنید. برای ورود به محیط command line از منوی Start گزینه All Programs سپس MySQL، MySQL Server 5.1 و در نهایت MySQL Command Line Client را انتخاب نمایید.



شکل ۳-۱

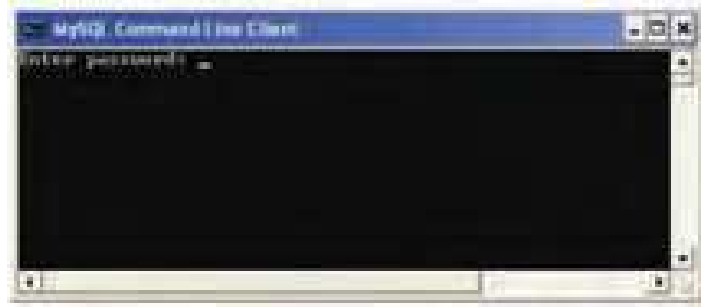
اگر رمز عبوری را هنگام نصب تعیین کرده‌اید، آن را وارد کنید، سپس کلید Enter را فشار دهید و اگر هیچ رمز ورودی تعیین نشده است، بدون وارد کردن هیچ کاراکتری، کلید Enter را فشار دهید.

## ۳-۲ دسترسی به MySQL از طریق خط فرمان

اگر قصد داشته باشید از طریق خط فرمان با محیط MySQL کار کنید، ابتدا باید MySQL را روی سیستم نصب کنید (با شیوه نصب MySQL در واحد کار اول آشنا شدید). از منوی Start و در قسمت All Programs گزینه MySQL، سپس MySQL Server و بعد از آن MySQL Command Line Client را انتخاب کنید.

در پنجره شکل ۳-۲ اگر کلمه رمزی را هنگام نصب MySQL تعیین کرده‌اید آن را وارد کنید، سپس کلید Enter را فشار دهید، در غیر این صورت بدون وارد کردن هیچ

کاراکتری کلید Enter را فشار دهید.



شکل ۲-۳

مقابل عبارت `MySQL>` علامت چشمک‌زن ظاهر می‌شود که به این معناست که به MySQL وارد شده‌اید و می‌توانید دستورات لازم را اجرا کنید. پس از ورود به محیط MySQL اگر بخواهید اعمال مختلف را روی یک بانک اطلاعاتی انجام دهید، ابتدا باید با استفاده از دستور `use` پایگاه داده مورد نظر را انتخاب کنید. ساختار این دستور به شکل زیر است:

نام پایگاه داده `use`;

مثال: `use db1;`

## ۳-۳ دستور `create`

دستور `Create` برای ایجاد یک جزء جدید در سیستم‌های پایگاه داده مورد استفاده قرار می‌گیرد.

### ۳-۳-۱ ایجاد یک پایگاه داده (`Create a DataBase`)

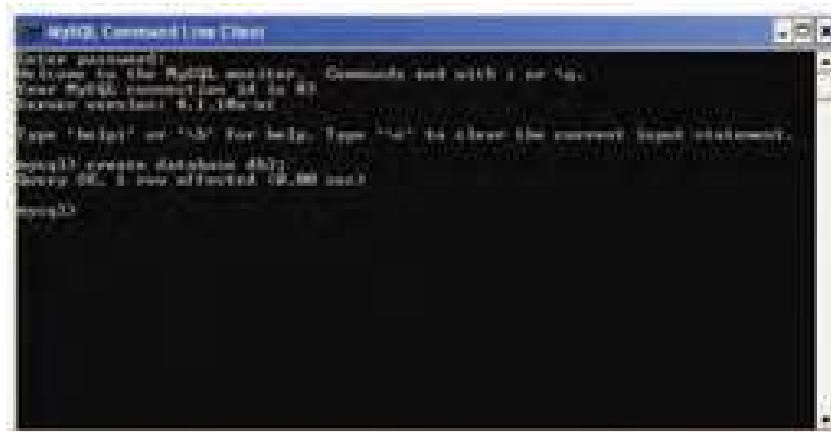
برای ایجاد یک پایگاه داده جدید از این دستور استفاده می‌شود:

نام پایگاه داده `create DataBase`;

مثال: `create DataBase UniverArea ;`

مثال: `create DataBase db3 ;`

برای اجرای دستور فوق در خط فرمان، دستور مورد نظر را وارد کنید، سپس کلید `Enter` را فشار دهید.




شکل ۳-۳

پیغام Query OK به این معناست که درخواست با موفقیت انجام پذیرفته است.

### ۲-۳-۳ ایجاد یک جدول (create Table)

برای ایجاد یک جدول ۳ عامل اصلی را باید در نظر گرفت:  
تعیین یک نام منحصر به فرد برای جدول  
تعیین تعداد ستون‌های (فیلدها) جدول و نام آن‌ها  
تعیین نوع داده‌ای ستون‌های جدول و اندازه آن‌ها در صورت نیاز  
شکل کلی دستور ایجاد جدول در پایگاه داده به صورت زیر است:

```
create Table Students  
(  
  StudentID Numeric (8) ,  
  Name Char (15) ,  
  Family Char (15) ,  
  ...  
)
```

**نکته**  نوع داده انتخابی برای ستون‌ها، باید یکی از انواع داده‌های مجاز در SQL باشد. برای فیلدهایی که می‌خواهید حداکثر طول مجاز را تعیین کنید، می‌توانید مقدار آن را در پرانتز جلوی نام نوع داده فیلد ذکر کنید.

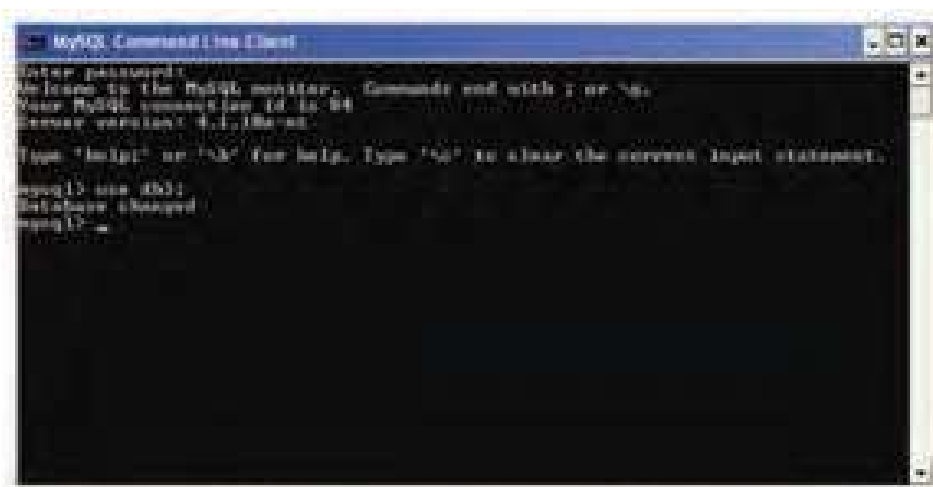
مثال: نوع داده‌ای از نوع کاراکتر با حداکثر طول ۱۵:

Name Char (15);

برای ایجاد جدول توسط خط فرمان، در صورتی که پایگاه داده را قبلاً ایجاد کرده باشید، ابتدا باید با استفاده از دستور

نام پایگاه داده use

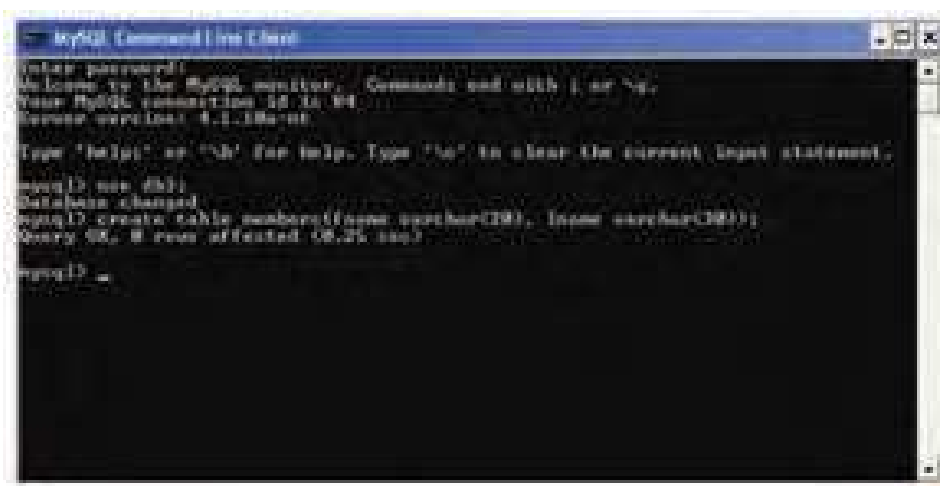
مشخص نمایید که می‌خواهید روی کدام پایگاه داده کار کنید.



شکل ۳-۴

حال برای ایجاد جدول باید دستور ایجاد جدول به همراه فیلدها و نوع هر کدام را به صورت زیر درج کنید:

(... و نوع متغیر دوم فیلد دوم و نوع متغیر اول فیلد اول) نام جدول create table



شکل ۳-۵

سایر دستورات SQL را نیز میتوانید از طریق خط فرمان اجرا کنید.

### ۳-۳-۳ ایجاد یک اندیس (create Index)

از این دستور برای ایجاد اندیس در ستون‌های جدول استفاده می‌شود. در یک جدول می‌توان برای یک یا چند ستون جدول، اندیس ایجاد کرد که این اندیس‌ها باعث بالا رفتن سرعت جستجو در رکوردهای جدول می‌شوند. اندیس یک شماره است که به هر سطر جدول اختصاص داده می‌شود و معمولاً از صفر شروع می‌شود. اندیس‌ها از دید کاربر مخفی هستند و هر اندیس یک نام منحصر به فرد دارد. شکل کلی ایجاد یک اندیس به صورت زیر است:

```
نام اندیس create Index  
(نام فیلد مورد نظر) نام جدول on  
مثال: create Index Rowup  
on Students (StudentID) ;
```

**نکته** آوردن واژه Unique قبل از واژه Index در دستور قبل، باعث ایجاد یک شماره منحصر به فرد برای هر رکورد در ستون جدول می‌شود، به این معنی که هیچ دو رکوردی دارای اندیس‌های یکسان نخواهند بود.

- اندیس‌ها به صورت پیش‌فرض صعودی (از کم به زیاد) هستند، برای ایجاد یک اندیس با ترتیب برعکس (از زیاد به کم) باید بعد از نام فیلد در پرانتز، عبارت DESC ذکر شود.
- به منظور ایجاد یک اندیس برای بیش از یک فیلد، باید نام فیلدهای مورد نظر را به ترتیب در پرانتز بعد از نام جدول وارد کرد.

**مثال:** برای ایجاد اندیس روی فیلدهای StudentID و Name و Family، در جدول ۳-۱، کد SQL را می‌توان به صورت زیر نوشت:

```
create Index Rowup  
on Students (StudentID , Name , Family) ;
```



جدول ۳-۱

Student Table					
StudentID	Name	Family	Major	Area	Grade
41252214	Ahmad	Rezaee	Computer	Tehran	۱۸
10724113	Ehsan	Amiri	Electronic	Karaj	۱۴
10254861	Zahra	Hoseini	Computer	Tehran	۱۷
27365187	Sahar	Ahmadi	Electronic	Bam	۱۶
35654415	Hesam	Razavi	Electronic	Tehran	۱۹



تمرین:

جدول ۳-۱ را در MySQL ایجاد کنید.

## ۳-۴ دستور select

دستور Select برای انتخاب و استخراج اطلاعات مورد نظر از یک یا چند جدول و سپس مشاهده نتایج در یک جدول موقت استفاده می‌شود. شکل کلی این دستور به شکل زیر است:

نام ستون یا ستون‌های مورد نظر select

نام جدول from

مثال: برای مشاهده ستون‌های Name و Family از جدول ۳-۱، دستور SQL به صورت

زیر نوشته می‌شود:

کد:

```
select Name,Family
```

```
from Student
```

خروجی

Name	Family
Ahamd	Rezaee
Ehsan	Amiri
Zahra	Hoseini
Sahar	Ahmadi

**نکته** برای انتخاب تمام ستون‌های جدول می‌توان از نشانگر \* در قسمت نام ستون استفاده کرد. در این صورت تمام ستون‌های جدول در خروجی نمایش داده خواهند شد:

```
select *  
from Student;
```

### تمرین:

۱ - یک دستور SQL بنویسید که فیلدهای نام، نام خانوادگی و نمره جدول ۱-۳ را نشان دهد.  
۲: جدول‌های ۲-۸، ۲-۹ و ۲-۱۰ (واحدکار دوم) را در محیط MySQL ایجاد کنید، سپس تمام فیلدهای این سه جدول را در خروجی نمایش دهید.

**واژه کلیدی DISTINCT:** چنانچه در ستون‌های مورد جستجو، موارد تکراری وجود داشته باشد در نتیجه خروجی نمایش داده خواهند شد. برای جلوگیری از چنین موردی و عدم نمایش موارد تکراری، پس از دستور Select عبارت DISTINCT نوشته می‌شود:

جدول ۲-۳

Orders Table	
Order Number	Company
1012	Dell
1014	Sega
1016	Dell
1018	Sony

مثال ۱:

کد:

```
select DISTINCT Company  
from Orders;
```

خروجی

Company
Dell
Sega
Sony

مثال ۲:

کد:

```
select Company
from Orders;
```

خروجی

Company
Dell
Sega
Dell
Sony

## ۵-۳ دستور where

دستور where برای افزودن شرط یا شرط‌هایی برای محدود کردن نتایج جستجو و استخراج نتایج دقیق‌تر استفاده می‌شود. این دستور باید پس از دستور select و تعیین ستون‌ها از جدول مورد نظر به کار رود. با استفاده از عملگرهای and، or و پرانتز می‌توان چندین شرط را با هم ترکیب کرد. برنامه فقط سطرهایی از جدول را که با شرط‌های ارائه شده سازگاری دارند نمایش خواهد داد. این دستور با عملگرهای زیر می‌تواند به کار رود:

جدول ۳-۳

عملگر	مفهوم	عملگر	مفهوم
=	برابر با، مساوی	=!	نا برابر با، نامساوی
<	کوچک‌تر یا کمتر	=<	کوچک‌تر مساوی
>	بزرگ‌تر یا بیشتر	=>	بزرگ‌تر مساوی
Between	بین دو مقدار	like	تعیین یک قالب برای جستجو

مثال: انتخاب نام و نام خانوادگی دانش‌آموزانی که شهر محل تولد آن‌ها تهران است از جدول ۳-۱ (Student) به این صورت انجام می‌شود:

کد:

```
select Name , Family
from Student
where Area = "Tehran ";
```

خروجی

Name	Family
Ahmad	Rezaee
Zahra	Hoseini
Hesam	Razavi

### ۱-۵-۳ اعمال بیش از یک شرط و ادغام شرط‌ها با عملگر and

مثال: مشخصات نام و نام خانوادگی دانش‌آموزانی که شهر آنها تهران بوده و نمره آنها بیش از پانزده است:  
کد:

```
select Name , Family  
from Student  
where Area = " Tehran"  
and Grade > 15;
```

خروجی

Name	Family
Ahmad	Rezaee
Hesam	Razavi

**نکته** در هنگام اعمال شرط جستجو، مقادیر رشته‌ای (متن) باید بین دو علامت " " قرار بگیرند، ولی در مورد موارد عددی این کار لازم نیست.

**تمرین:** 

در جدول S که آنرا در تمرین ۲ ایجاد کردید، اطلاعات رکوردهایی را نمایش دهید که در شهر تهران تولید شده‌اند.

### ۲-۵-۳ عبارت like

این عبارت برای تعیین یک قالب خاص برای جستجو به کار می‌رود:  
مثال: نام دانش‌آموزانی که شهر محل تولد آنها با حرف K شروع می‌شود:

```
select Name from Student  
where area like "K%";
```

مثال: نام دانش‌آموزانی که نام شهر محل تولد آنها به an ختم می‌شود:

```
select Name from Student  
where Area like "%an";
```

مثال: نام دانش‌آموزانی که شهر محل تولد آنها دارای حروف st است:

```
Select Name From Student
```

Where Area like "%st%";

مثال: نام دانش‌آموزانی که شهر محل تولد آن‌ها ۶ حرفی بوده و با حرف T شروع می‌شود:

select Name from Student

where Area like "T-----" ;

نتیجۀ گیری: علامت % به معنای هر چند کاراکتر یا حرف دلخواه در قالب ارائه شده و علامت - نمایانگر یک کاراکتر دلخواه است.

## ۳-۶ دستور insert

این دستور برای وارد کردن و ایجاد یک رکورد جدید در جدول استفاده می‌شود. شکل کلی این دستور به این شرح است:

insert into نام جدول

values (...و مقدار ۲ و مقدار ۱) ;

می‌توان تعیین کرد که مقادیر مورد نظر به ترتیب به کدام ستون‌های جدول وارد شوند. برای این منظور ابتدا نام ستون‌های مورد نظر را به ترتیب جلوی نام جدول در یک پرانتز وارد کرده و با کاما از هم جدا می‌کنیم. سپس مقادیر متناظر را به همان ترتیب پس از واژه values در پرانتز وارد می‌کنیم.

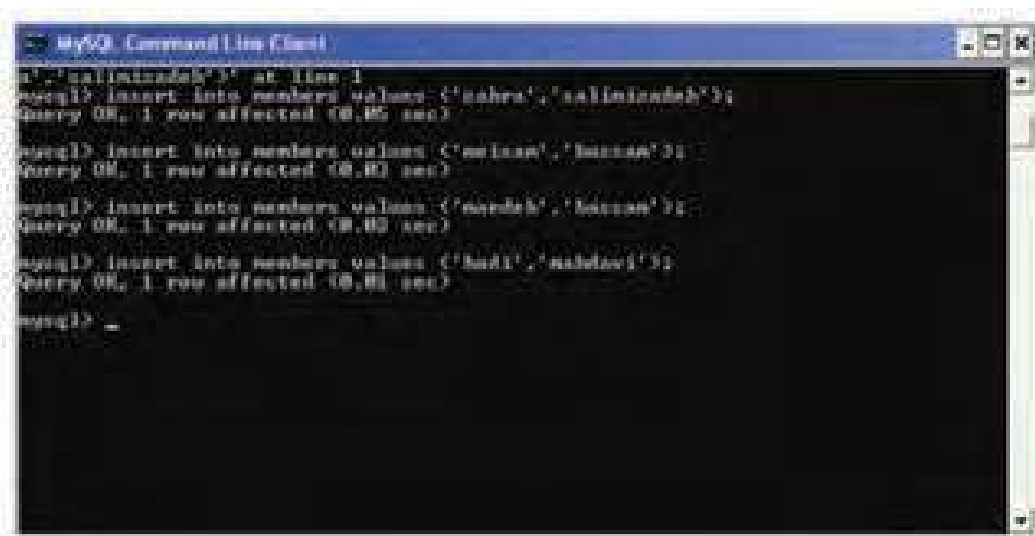
**نکته** چنانچه برای فیلد یا فیلدهایی مقداری در نظر گرفته نشود، جای آن‌ها در جدول خالی می‌ماند. فقط باید به ترتیب نام ستون‌ها و مقادیر دقت کرد.

در شکل ۳-۶ درج داده در جدول members را مشاهده می‌کنید. حال برای اینکه اطلاعات درج شده را مشاهده کنید، می‌توانید از دستور زیر استفاده نمایید:

SELECT

\* from members;

این دستور را نوشته، سپس کلید Enter را فشار دهید. مشاهده می‌کنید اطلاعاتی که قبلاً در جدول وارد شده بود مطابق شکل ۳-۷ ظاهر شده‌اند.



```
mysql> insert into members values ('zahra','sallimzadeh');
Query OK, 1 row affected (0.05 sec)

mysql> insert into members values ('nohooon','hassan');
Query OK, 1 row affected (0.02 sec)

mysql> insert into members values ('mehdi','hassan');
Query OK, 1 row affected (0.02 sec)

mysql> insert into members values ('hadi','sahbani');
Query OK, 1 row affected (0.01 sec)

mysql> _
```

شکل ۳-۶



```
mysql> insert into members values ('zahra','hassan');
Query OK, 1 row affected (0.02 sec)

mysql> insert into members values ('hadi','sahbani');
Query OK, 1 row affected (0.01 sec)

mysql> select * from members;
+-----+-----+
| Name | Name |
+-----+-----+
| zahra | hassan |
| mehdi | hassan |
| hadi | sahbani |
+-----+-----+
4 rows in set (0.01 sec)

mysql> _
```

شکل ۳-۷

**مثال:** مشخصات نام و نام خانوادگی را برای یک دانش‌آموز به نام Omid Safavi در جدول ۳-۱ وارد کنید.

شرح دستور وارد کردن نام و وضعیت جدید جدول:  
کد:

```
insert into Student ( Name , Family )
values ( "Omid" , "Safavi" );
```

## خروجی

Student Table					
StudentID	Name	Family	Major	Area	Grade
41252214	Ahmad	Rezaee	Computer	Tehran	18
10724113	Ehsan	Amiri	Electronic	Karaj	14
10254861	Zahra	Hoseini	Computer	Tehran	17
27365187	Sahar	Ahmadi	Electronic	Bam	16
35654415	Hesam	Razavi	Electronic	Tehran	19
	Omid	Safavi			

## ۳-۷ دستور update

این دستور برای تغییر اطلاعات موجود در یک فیلد جدول و جایگزینی آن با یک مقدار جدید به کار می‌رود.  
شکل کلی این دستور به صورت زیر است:

نام جدول update

مقدار جدید = نام فیلد set

مقدار قبلی = نام فیلد where ؛

مثال: نام دانش‌آموز احمد رضایی را به علی تغییر دهید:

```
update Student
```

```
set Name = "Ali"
```

```
where Name = "Ahmad" and Family = "Rezaee";
```

• همچنین می‌توان برای تعیین رکوردی که می‌خواهیم تغییر دهیم، از مقدار یک فیلد دیگر به غیر از فیلدی که می‌خواهیم تغییر کند استفاده کنیم. در این حالت از فیلد مذکور، برای آدرس دهی رکورد مورد نظر استفاده می‌شود.

مثال: نام دانش‌آموز با شماره دانش‌آموزی ۴۱۲۵۲۲۱۴ را به علی در جدول ۳-۱ تغییر دهید:

• اعمال دستور تغییر و وضعیت جدید جدول:

کد:

```
update Student set Name = "Ali "
```

```
where StudentID = "41252214 ";
```

### خروجی

Student Table					
StudentID	Name	Family	Major	Area	Grade
41252214	Ali	Rezaee	Computer	Tehran	18
10724113	Ehsan	Amiri	Electronic	Karaj	14
10254861	Zahra	Hoseini	Computer	Tehran	17
27365187	Sahar	Ahmadi	Electronic	Bam	16
35654415	Hesam	Razavi	Electronic	Tehran	19

### تغییر اطلاعات در بیش از یک فیلد در رکورد

در این حالت تمام مقادیر جدیدی که می‌خواهیم تغییر دهیم، جلوی واژه کلیدی Set به ترتیب نوشته و با کاما از هم جدا می‌کنیم. سپس به همان ترتیب، فیلدها با مقادیر قبلی را در جلوی واژه where می‌نویسیم. شکل کلی آن به صورت زیر است:

نام جدول update

مقدار = نام فیلد ۲ ، مقدار = نام فیلد ۱ set

؛ ... و مقدار قبلی = نام فیلد ۲ و مقدار قبلی = نام فیلد ۱ where

• همچنین می‌توان فقط از یک فیلد در قسمت شرط برای آدرس دهی رکورد مورد

نظر استفاده کرد.

### تمرین:

در جدول ۹-۲ از واحد کار دوم، مشخصات تولیدکننده P6 را تغییر دهید، طوری که وزن قطعه‌های تولید شده به ۲۵ و رنگ قطعه به زرد تبدیل شود.

**مثال:** در جدول ۱-۳ نام، نام خانوادگی و رشته تحصیلی دانش‌آموز به شماره

دانش‌آموزی ۴۱۲۵۲۲۱۴ را به ترتیب به Ali، Saeedi و Electronic تغییر دهید:

• اعمال دستور تغییر و وضعیت جدید جدول:

کد:

```
update Student
```

```
set Name = "Ali " , Family = " Saeedi " , Major =" Electronic "
```

```
where StudentID = 41252214;
```



### خروجی

Student Table					
StudentID	Name	Family	Major	Area	Grade
41252214	Ali	Saeedi	Electronic	Tehran	18
10724113	Ehsan	Amiri	Electronic	Karaj	14
10254861	Zahra	Hoseini	Computer	Tehran	17
27365187	Sahar	Ahmadi	Electronic	Bam	16
35654415	Hesam	Razavi	Electronic	Tehran	19

### ۳-۸ دستور delete

از این دستور برای حذف اطلاعات یک رکورد در جدول استفاده می‌شود.  
 شکل کلی این دستور به شکل زیر است:

نام جدول delete from

مقدار فیلد از رکورد مورد نظر = نام فیلد where

مثال: در جدول ۳-۱، رکورد اطلاعات دانش‌آموز به شماره دانش‌آموزی ۱۰۷۲۴۱۱۳

را حذف کنید.

اعمال دستور حذف و وضعیت جدید جدول:

کد:

delete from Student

where StudentID = "10724113";

### خروجی

Student Table					
StudentID	Name	Family	Major	Area	Grade
41252214	Ahmad	Rezaee	Computer	Tehran	18
10254861	Zahra	Hoseini	Computer	Tehran	17
27365187	Sahar	Ahmadi	Electronic	Bam	16
35654415	Hesam	Razavi	Electronic	Tehran	19

### حذف تمام رکوردهای یک جدول

این امکان وجود دارد که بدون حذف یک جدول، تمام رکوردهای درون آن را پاک کرد. این به منزله خالی کردن جدول از اطلاعات است، یعنی ساختار، اندیس‌ها و ویژگی‌های جدول دست نخورده باقی مانده و فقط مقادیر درون آن از بین می‌رود.

شکل کلی این دستور به صورت زیر است:

delete from جدول ;

یا

delete \* from جدول ;

مثال: حذف تمام رکوردهای جدول Student

delete from Student;

## ۳-۹ دستور order by

اطلاعات خروجی در دستور Select به طور کلی بی نظم است و مقادیر خروجی در ستون‌های جدول بر حسب مقدار هیچ ستونی مرتب نیستند. این دستور برای مرتب کردن اطلاعات جدول بر اساس مقادیر یک یا چند ستون بر حسب شاخص‌هایی مثل ترتیب حروف الفبا، بزرگ‌تر یا کوچک‌تر بودن اعداد و ... استفاده می‌شود. شکل کلی این دستور به صورت زیر است:

نام جدول مورد نظر from نام فیلدهای مورد نظر select

نام فیلد مورد نظر order by ;

فیلدهای معرفی شده در دستور select از جدول مذکور نمایش داده شده و اطلاعات بر حسب ستون مقابل دستور order by مرتب می‌شوند.

**عبارت ASC:** به کار بردن این دستور پس از دستور order by باعث مرتب شدن اطلاعات درون جدول از کم به زیاد می‌شود که البته انتخاب پیش فرض SQL است و نیازی به درج آن نیست.

**عبارت DESC:** به کار بردن این واژه پس از دستور order by باعث مرتب شدن اطلاعات جدول از زیاد به کم در اعداد و در حروف از Z تا A می‌شود.

مثال: از جدول Student، مشخصات نام و نام خانوادگی دانش‌آموزان رشته Electronic را بر حسب حروف الفبا نمایش دهید.  
کد:

```
select Name , Family from Student
```

```
where Major ="Electronic"
```

```
order by Family ;
```

خروجی

Name	Family
Sahar	Ahmadi
Ehsan	Amiri
Hesam	Razavi

**نکته** همچنین می‌توان یک جدول را بر حسب بیش از یک فیلد مرتب کرد، برای این منظور پس از دستور Order By نام تمام فیلدهای مورد نظر را نوشته و آنها را با کاما از هم جدا می‌کنیم. اولویت مرتب‌سازی با فیلدهایی است که نام آنها اول ذکر شده باشد.

**مثال:** جدول ۱-۳ را بر حسب فیلد نام خانوادگی (بر حسب حروف از A تا Z) و بر حسب نمره (از کم به زیاد) مرتب کنید:

**نکته** در جدول خروجی مثال فوق به ارتباط مقادیر ستون‌های Family و Grade توجه کنید. با وجود این که نمره درسی دانش‌آموز زهرا حسینی از احمد رضایی کمتر است، ولی به دلیل این که اولویت مرتب‌سازی با نام خانوادگی است، نام حسینی بالاتر از رضایی قرار گرفته است.

کد:

```
select Name , Family , Grade From Student
order by Family ASC , Grade DESC ;
```

خروجی

Name	Family	Grade
Sahar	Ahmadi	16
Ehsan	Amiri	14
Zahra	Hoseini	17
Hesam	Razavi	19
Ahmad	Rezaee	18

**نکته** می‌توان در دستور order by به جای ذکر نام ستون‌ها از یک عدد به جای آن استفاده کرد. عدد استفاده شده، شماره قرارگیری ستون مورد نظر در دستور Select است. برای مثال اولین ستون دستور Select عدد ۱ و دومین ستون عدد ۲ و ... را خواهند داشت. بازنویسی دستور بالا با استفاده از این روش به صورت زیر خواهد بود:

کد:

```
select Name , Family , Grade from Student
order by 2 ASC , 3DESC;
```

## ۱۰-۳ دستور and و or

از and و or برای ترکیب شرطها در دستور where استفاده می شود. گاهی اوقات اطلاعات مورد نیاز ما در دستور select دارای چندین شرط مختلف است. به طور مثال ما مشخصات دانش آموزان ترم دوم رشته Computer که معدل آنها بیش از ۱۵ است را می خواهیم. در این حالت باید هر یک از شرطهای مسئله را به تنهایی تعریف کرده و سپس آنها را با هم ترکیب کنیم. برنامه به ارزیابی هر یک از شرطها پرداخته، سپس نتایج حاصله را با هم ترکیب کرده و خروجی را تعیین می کند. عملگر and برای اجرای دستور نیاز دارد تا تمام شرطهای تعیین شده برای آن، درست باشد. در حالی که عملگر or فقط نیاز دارد که حداقل یکی از شرطهای آن درست باشد. در جداول زیر چگونگی برقراری نتیجه نهایی در هنگام استفاده از این دو عملگر را مشاهده می کنیم:

and		
شرط ۱	شرط ۲	نتیجه
درست	درست	درست
درست	غلط	غلط
غلط	درست	غلط
غلط	غلط	غلط

or		
شرط ۱	شرط ۲	نتیجه
درست	درست	درست
درست	غلط	درست
غلط	درست	درست
غلط	غلط	غلط

چگونگی حالت های فوق را در مثال های زیر بررسی می کنیم:  
مثال ۱: مشخصات دانش آموزانی را از جدول ۱-۳ (Student) ارائه دهید که نمره آنها بیش از ۱۵ و رشته تحصیلی آنها Electronic باشد:  
کد:

```
select * from Student  
where Major ='Electronic' and Grade > 15 ;
```

## خروجی


StudentID	Name	Family	Major	Area	Grade
27365187	Sahar	Ahmadi	Electronic	Bam	16
35654415	Hesam	Razavi	Electronic	Tehran	19

مثال ۲: مشخصات دانش آموزانی را از جدول Student ارائه دهید که نمره آن‌ها بیش از ۱۷ بوده یا رشته تحصیلی آن‌ها Computer باشد:  
 کد:

```
select * from Student
where Major = 'Computer' OR Grade > 17 ;
```

## خروجی

StudentID	Name	Family	Major	Area	Grade
41252214	Ahmad	Rezaee	Computer	Tehran	18
35654415	Hesam	Razavi	Electronic	Tehran	19

**نکته**  همچنین می‌توان با استفاده از عملگر پرانتز چندین شرط مختلف را به صورت دسته‌های مجزا تعریف کرده و سپس همه آن‌ها را با هم ترکیب کرد. در این حالت شرط‌های هر گروه ابتدا داخل پرانتز با هم ترکیب شده، سپس نتیجه حاصله به عنوان نتیجه نهایی آن گروه شرط به دستور Where ارسال می‌شود و در نهایت نتایج تمام گروه‌ها با هم ترکیب می‌شوند.

مثال: از جدول Student مشخصات دانش آموزانی را ارائه دهید که نمره آن‌ها ۱۸ و رشته تحصیلی آن‌ها Computer بوده یا نمره آن‌ها ۱۶ و رشته تحصیلی آن‌ها Electronic باشد:  
 کد:

```
select * from Student
where ( ( Grade = 18 and Major = 'Computer' ) or ( Garde = 16 and Major = 'Electronic' ) ) " ;
```

### خروجی

StudentID	Name	Family	Major	Area	Grade
41252214	Ahmad	Rezaee	Computer	Tehran	18
27365187	Sahar	Ahmadi	Electronic	Bam	16

## ۱۱-۳ دستور in

این دستور برای مشاهده اطلاعات رکوردهایی از جدول به کار می‌رود که شما مقدار دقیق حداقل یکی از فیلدهای آن را می‌دانید.

شکل کلی این دستور به صورت زیر است:

نام جدول from نام فیلدهایی که می‌خواهید نمایش داده شود select  
 where ( ... و مقدار ۲ و مقدار ۱ ) in نام فیلد

مثال: از جدول Student مشخصات دانش‌آموزانی را ارائه دهید که نام خانوادگی آنها

احمدی یا رضایی است:

کد:

```
select * from Student
```

```
where Family in ( "Ahmadi" , "Rezaee" );
```

### خروجی

Student Table					
StudentID	Name	Family	Major	Area	Grade
41252214	Ahmad	Rezaee	Computer	Tehran	18
27365187	Sahar	Ahmadi	Electronic	Bam	16

## ۱۲-۳ دستور between ... and

این دستور برای انتخاب اطلاعات در یک محدوده خاص، در بین دو مقدار تعیین شده استفاده می‌شود. این مقادیر می‌تواند از نوع عددی، متن یا تاریخ باشد. (باید نوع مدنظر با نوع داده‌ای فیلدهای استفاده شده در دستور یکسان باشد).

شکل کلی این دستور به شکل زیر است:

نام جدول from نام فیلدهای مورد نظر برای نمایش select  
 نام فیلد مورد نظر where  
 مقدار ۱ and مقدار ۲ between ;

**نکته** در مورد داده‌های متنی، ترتیب قرارگیری حروف الفبا مد نظر برنامه است.

**مثال:** از جدول Student مشخصات دانش‌آموزانی را ارائه دهید که نام خانوادگی آنها بین دو مقدار احمدی و حسینی باشد. در ضمن خروجی بر حسب نام خانوادگی به ترتیب حروف الفبا مرتب باشد:

کد:

```
select * from Student
where Family between 'Ahmadi' and 'Hoseini'
order by family;
```

خروجی

StudentID	Name	Family	Major	Area	Grade
27365187	Sahar	Ahmadi	Electronic	Bam	16
10724113	Ehsan	Amiri	Electronic	Karaj	14
10254861	Zahra	Hoseini	Computer	Tehran	17

**نکته** قرار گرفتن رکوردهای ۲ مقدار تعیین شده در خروجی دستور select، به نوع برنامه پایگاه داده و مدیریت آن بستگی دارد. در برخی از پایگاه‌های داده رکوردهای هر دو مقدار ابتدا و انتها نمایش داده می‌شود و در برخی هم فقط یکی از آنها نمایش داده می‌شود. برای نمایش اطلاعات خارج از محدوده تعیین شده از یک عملگر not قبل از دستور between ... and استفاده می‌شود.

**مثال:** از جدول ۱-۳ مشخصات دانش‌آموزانی را ارائه دهید که نمره آنها خارج از محدوده ۱۲ تا ۱۵ باشد. در ضمن مقادیر خروجی بر حسب نام خانوادگی به ترتیب حروف الفبا مرتب باشد:

کد:

```
select * from Student
where Grade not between 12 and 15
order by Family;
```

خروجی

StudentID	Name	Family	Major	Area	Grade
27365187	Sahar	Ahmadi	Electronic	Bam	16
10254861	Zahra	Hoseini	Computer	Tehran	17
35654415	Hesam	Razavi	Electronic	Tehran	19
41252214	Ahmad	Rezaee	Computer	Tehran	18

### ۱۳-۳ ویژگی alias

همانطور که در تمام مثال‌های ذکر شده از دستور select مشاهده کردید، نام یا عنوانی که در بالای هر ستون در خروجی نمایش داده می‌شود، همان نام فیلد مربوط به داده در جدول اصلی پایگاه داده است. گاهی اوقات تمایل بر این است که این نام را تغییر دهیم یا در مواردی که در بخش توابع SQL خواهیم دید، یک ستون جدید ایجاد می‌کنیم که باید نامی برای آن تعیین شود. از ویژگی alias برای در نظر گرفتن یک نام مستعار و موقت برای قرار گرفتن در بخش عنوان هر ستون در خروجی دستور select استفاده می‌شود. شکل کلی این دستور به صورت زیر است:

نام جدول from (alias) عنوان as نام فیلد select

مثال: از جدول Student نام و نام خانوادگی دانش‌آموزان رشته Electronic را بر حسب حروف الفبا در ۲ ستون تحت عنوان‌های "نام" و "نام خانوادگی" نمایش دهید:  
کد:

```
select Name as نام , Family as نام خانوادگی from Student
where Major = 'Electronic'
order by Family ;
```

خروجی

نام	نام خانوادگی
Sahar	Ahmadi
Ehsan	Amiri
Hesam	Razavi





**نکته** می‌توان روی مقادیر ستون‌هایی که داده عددی دارند، عملیات ریاضی (مثل ضرب، تقسیم و ...) انجام داده و سپس نتایج محاسبات را در یک ستون جدید با عنوان دلخواه تعیین شده توسط ویژگی alias در خروجی دستور select نمایش داد. در این حالت ستون یا ستون‌ها به جدول خروجی اضافه می‌شود.

در این ارتباط مثالی را ذکر می‌کنیم:  
 مثال: فرض کنید جدولی داریم به نام Books که در آن مشخصات چند کتاب مختلف قرار دارد. در این جدول قیمت کتاب‌ها به دلار است و ما می‌خواهیم قیمت کتاب‌ها را در یک ستون جدید به تومان مشاهده کنیم پس به این روش عمل می‌کنیم:

جدول ۳-۴

Books Table			
Price	Publisher	Name	no
8	Microsoft	Learning Access	1
6	DeveloperStudio	Java Script Samples	2
7	DeveloperStudio	HTML Reference	3

کد

select no , Name , Publisher , Price , Price \* 840 as قیمت به تومان from Books ;

خروجی

Books Table				
no	Name	Publisher	Price \$	قیمت به تومان
1	Learning Access	Microsoft	8	6720
2	Java Script Samples	DeveloperStudio	6	5040
3	HTML Reference	DeveloperStudio	7	5880

## ۱۴-۳ پیوند جدول‌ها

تا این قسمت تمام مثال‌ها و مسئله‌هایی که در SQL به آن‌ها پاسخ دادیم، مسئله‌هایی بودند که در آن‌ها اطلاعات ما فقط از یک جدول استخراج می‌شد. اما در برنامه‌نویسی واقعی پایگاه داده‌ها، ما مجبور هستیم که اطلاعات خود را از بیش از یک جدول استخراج کنیم.

در این حالت ابتدا باید جدول‌هایی که می‌خواهیم اطلاعات را از آن‌ها استخراج کنیم، با هم پیوند دهیم. هدف از ایجاد این ارتباط تلفیق اطلاعات در جدول‌ها و چاپ اطلاعات مورد نظر در خروجی است.

برای پیوند دادن جدول‌ها به عنوان یادآوری این نکات را در نظر داشته باشید:  
**کلید اصلی:** فیلد کلید اصلی در یک جدول، فیلدی است که شرایط زیر را داشته باشد:  
مقدار آن برای هر نمونه رکورد (سطر) منحصر به فرد و غیرتکراری باشد. به عبارت دیگر هیچ ۲ رکوردی در یک جدول در این فیلد مقدار یکسان نداشته باشند. کلید اصلی وجه تمایز ۲ نمونه رکورد مختلف در یک جدول است.  
طول مقادیر آن در حد امکان کوتاه باشد.

**نکته**  یک جدول می‌تواند بیش از یک کلید اصلی داشته باشد.

**مثال:** فیلد شماره دانش‌آموزی در جدول Student، کلید اصلی است. هیچ دو دانش‌آموزی نمی‌توانند دارای شماره دانش‌آموزی یکسان باشند.

**کلید خارجی:** کلید خارجی، فیلدی است که در یک جدول کلید اصلی و در جدول دیگر به تنهایی کلید اصلی نباشد. از کلید خارجی برای ارتباط یک به چند ۲ جدول با هم استفاده می‌شود.

### شرط ارتباط دو جدول

برای ارتباط بین جدول‌ها باید شرط‌های زیر برقرار باشد. باید قبل از طراحی پایگاه داده و جدول‌های آن موارد زیر را جهت ارتباط جدول‌های مورد نظر رعایت کرد:

- وجود فیلد مشترک دقیقاً از یک نوع و یک اندازه.
- فیلد مشترک در یکی از جدول‌ها کلید اصلی و در جدول دیگر کلید خارجی باشد.

**نکته**  لزومی ندارد مقادیر موجود در جدول‌های مبدأ و مقصد یکسان باشد.

### معرفی دو جدول دیگر

از این به بعد ما در مثال‌های خود از ۲ جدول دیگر به غیر از جدول Student، به

نام‌های Courses (درس‌ها) و Selection (انتخاب واحد) به شرح زیر استفاده می‌کنیم:

جدول ۳-۵

Courses Table			
CoType	Credit	CoTitle	CourseID
نوع درس	تعداد واحد	عنوان درس	(کلید اصلی) کد درس

جدول ۳-۶

Selection Table				
Grade	Year	Term	CourseID	StudentID
نمره	سال تحصیلی	ترم تحصیلی	کد درس (کلید اصلی خارجی)	شماره دانش آموزی (کلید اصلی خارجی)

**نکته** • در تمام مثال‌های قبلی، ما در دستور select فقط نام ستون‌ها را به تنهایی ذکر می‌کردیم، زیرا در آن زمان، اطلاعات ما فقط از یک جدول استخراج می‌شد. اما در هنگام پیوند دو جدول و استفاده از چند جدول در دستور select باید نام ستون را به همراه نام جدول مربوط به آن ذکر کرد. این کار ۲ دلیل اصلی دارد:

- باعث تمایز ستون‌های مشترک در جدول‌ها از یکدیگر می‌شود و مشخص می‌کند که هر ستون مربوط به کدام جدول است.
- باعث خوانایی و دقت بیشتر برنامه می‌شود.

شکل کلی این دستور به صورت زیر است:

نام ستون . نام جدول

مثال: انتخاب ستون StudentID از جدول Student:

Student.StudentID

### ۱-۱۴-۳ مثال‌های پیوند جدول‌ها

در این قسمت با ارائه چندین مثال، انواع حالت‌های مختلف پیوند جدول‌ها را بررسی می‌کنیم. از داده‌های موجود در جداول زیر برای مثال‌ها استفاده می‌کنیم:

**توجه:** جدول انتخاب واحد نشان‌دهنده این است که هر دانش‌آموز چه واحدهای درسی را در چه ترم و سالی و با چه نمره‌ای گذرانده است.

جدول ۳-۷

Courses Table			
CoType	Credit	CoTitle	CourseID
عملی	3	پایگاه داده	1011
عملی	3	سخت‌افزار	1012
نظری	2	زبان تخصصی	1013

جدول ۳-۸

Selection Table				
Grade	Year	Term	CourseID	StudentID
16	88 - 89	2	1011	41252214
14	88 - 89	2	1011	10724113
17	88 - 89	1	1012	41252214
11	88 - 89	1	1012	10724113
13	88 - 89	2	1013	10254861
8	87 - 88	2	1011	10254861
19	87 - 88	1	1012	27365187
16	87 - 88	1	1013	27365187
9	87 - 88	2	1011	35654415
17	87 - 88	2	1013	35654415

شکل کلی پیوند ۲ جدول برای استخراج اطلاعات به صورت زیر است:

نام رکورد یا رکوردهای مورد نظر برای نمایش  
 select  
 نام جدول‌ها  
 from  
 برابر قرار دادن فیلدهای مشترک ۲ جدول  
 where  
 ; بقیه شرط‌های مورد نظر  
 and

در این حالت ابتدا در دستور select نام ستون‌هایی را که می‌خواهیم از ۲ جدول نمایش دهیم، تعیین می‌کنیم. سپس نام ۲ جدول را در مقابل دستور from نوشته و در اولین شرط دستور where، نام فیلد مشترک را از هر ۲ جدول نوشته و آن‌ها را برابر هم قرار می‌دهیم. این شرط، شرط برقراری پیوند و تلفیق اطلاعات ۲ جدول است. در ادامه هم می‌توان شرط‌های دیگری را برای استخراج اطلاعات تعیین کرد. در مثال‌های بعدی این مسئله را بررسی می‌کنیم:

**مثال:** نام و نام خانوادگی و سال ورود دانش آموزانی را ارائه دهید که در ترم ۱ سال تحصیلی ۸۸-۸۹، درسی را با کد ۱۰۱۲ انتخاب کرده‌اند:  
**کد:**

```
select Student.Name , Student.Family , Selection.Term , Selection.Year
from Student , Selection
where Student.StudentID = Selection.StudentID
and CourseID = 1012 and Term = 1 and Year ='88 - 89'
order by Student.Family;
```

خروجی

Name	Family	Term	Year
Ehsan	Amiri	1	88 - 89
Ahmad	Rezaee	1	88 - 89

**مثال:** نام دروسی را ارائه دهید که دانش آموز با شماره دانش آموزی ۱۰۲۵۴۸۶۱ انتخاب کرده است:  
**کد:**

```
select Courses.CourseID , Courses.Co Title
from Courses , selection
where Courses.CourseID = Selection.CourseID
and Selection.StudentID = 10254861 ;
```

خروجی

CourseID	CourseTitle
1011	پایگاه داده
1013	زبان تخصصی

**مثال:** نام و نام خانوادگی دانش آموزانی را ارائه دهید که درس با کد ۱۰۱۳ در سال تحصیلی ۸۷-۸۸ را با نمره بالاتر از ۱۵ گذرانده‌اند:  
**کد:**

```
select Student.Name , Student.Family
from Student , Selection
where Student.Studentid = Selection.Studentid
and Selection.Courseid = '1013' and Year = '87 - 88' and Grade > 15 ;
```

خروجی

Name	Family
Sahar	Ahamdi
Hesam	Razavi

## ۲-۱۴-۳ پیوند بیش از دو جدول به یکدیگر

گاهی اوقات لازم است که اطلاعات مورد نیاز ما از ۳ جدول یا بیشتر استخراج شود. در این حالت باید تمام جدول‌ها را به هم پیوند دهیم، به این صورت که معمولاً از یک جدول سوم برای پیوند ۲ جدول دیگر استفاده می‌شود و دو به دو جدول‌هایی که با هم فیلد مشترک دارند را با ذکر شرط پیوند در دستور where به هم پیوند می‌دهیم. سپس بقیه شروط دلخواه را نیز ذکر می‌کنیم. شکل کلی این حالت به این صورت است:

```
نام ستون‌های مورد نظر از جدول‌ها select  
نام تمام جدول‌ها from  
برابر قرار دادن فیلد مشترک جدول‌های ۱ و ۲ where  
برابر قرار دادن فیلدهای مشترک جدول‌های ۲ و ۳ and  
and ... ;
```

**مثال:** نام و نام خانوادگی دانش‌آموزانی را ارائه دهید که حداقل یک درس از نوع نظری را انتخاب کرده باشند:  
**کد:**

```
select Students.Name , Students.Family , Courses.CoTitle , Courses.CoType  
from Students , Courses , Selections  
where Student.StudentID = Selection.StudentID  
and Courses.CourseID = Selection.CourseID ;  
and Courses.CoType = 'نظری' ;
```

### خروجی

Name	Family	CoTitle	CoType
Zahra	Hoseini	زبان تخصصی	نظری
Sahar	Ahamadi	زبان تخصصی	نظری
Hesam	Razavi	زبان تخصصی	نظری

با دقت در اطلاعات جدول‌های اصلی متوجه درست بودن نتایج خروجی خواهید شد.

## ۱۵-۳ دستور join

از دستور join برای پیوند دو یا چند جدول در پایگاه داده‌ها استفاده می‌شود. قبل از بیان دستور Join به جدول‌های زیر دقت کنید:

a1	b1	b1	c1
a2	b1	b2	c2
a3	b2	b3	c3
a		b	

می‌خواهیم عمل پیوند دو جدول a و b را طوری انجام دهیم که یک جدول با سه ستون ایجاد شود، در این صورت جدول حاصل به صورت زیر خواهد بود:

a1	b1	c1
a2	b1	c1
a3	b2	c2

برای ایجاد جدول سوم، سطرهای جدول اول با جدول دوم سنجیده می‌شوند، به عنوان مثال در سطر اول، b1 و a1 با هم متناظرند، و در جدول دوم عنصر b1 با عنصر c1 متناظر است. در نتیجه عنصر c1 به عنوان عنصر متناظر b1 و a1 در جدول سوم است. (c1، b1 و a1).

در سطر سوم از جدول اول نیز b2 و a3 با همدیگر متناظرند، در جدول دوم، عنصر b2 با عنصر c2 متناظر است. در نتیجه عنصر c2 به عنوان عنصر متناظر b2 و a3 در جدول سوم است. (c2، b2 و a3).

عمل join (الحاق) رابطه‌ای را برمی‌گرداند که شامل تمام ترکیب‌های حاصل از دو رابطه اولیه است. برای پیوند جدول‌ها با استفاده از دستور join، از روش کلی زیر استفاده می‌شود:

نام ستون‌های موردنظر برای نمایش از هر دو جدول select

نام جدول اول from

نام جدول دوم inner join

برابر قرار دادن فیلدهای مشترک دو جدول on

سایر شرط‌ها where

مثال: می‌خواهیم در جدول‌های Selection و Student، نام و نام خانوادگی، ترم و سال

تحصیلی دانش‌آموزانی را استخراج کنیم که در ترم ۱ سال تحصیلی ۸۸-۸۹، درسی با کد ۱۰۱۲ را انتخاب کرده‌اند:  
کد:

```
select Student.Name , Student.Family , Selection.Term , Selection.Year
from Student
inner join Selection
on Student.StudentID = Selection.StudentID
where Selection.CourseID = 1012 and Term = 1 and Year = ' 88 - 89 ' ;
```

خروجی

Name	Family	Term	Year
Ehsan	Amiri	1	88 - 89
Ahmad	Rezaee	1	88 - 89

مثال: نام و شماره درس‌هایی را ارائه دهید که دانش‌آموز با شماره دانش‌آموزی ۱۰۲۵۴۸۶۱ انتخاب کرده است:  
کد:

```
select Courses.CourseID , Courses.Co Title
from Courses
inner join Selection
on Courses.CourseID = Selection.CourseID
where Selection.StudentID = 10254861 ;
```

خروجی

CourseID	CourseTitle
1011	پایگاه داده
1013	زبان تخصصی

## ۱۶-۳ Select های تودرتو

می‌توان درون یک دستور select، دستور دیگری را تعریف کرد. در این حالت ابتدا دستور Select درونی اجرا شده و نتایج مورد نظر خود را که سطرهایی از جدول مربوط به آن هستند استخراج کرده و آن را به select بیرونی انتقال می‌دهد. سپس select



بیرونی نیز اجرا شده و نتایج استخراجی خود را با استفاده از فیلد مشترک بین ۲ جدول با سطرهای select درونی پیوند زده و سطرهای مطابق با شرطهای تعیین شده را نمایش می دهد.

 **نکته** به تعداد دستورات select تودرتو، سطح آن می گویند.

به علت پیچیدگی selectهای ۳ سطحی، ۴ سطحی و ... در این واحد کار فقط دستور select دو سطحی را بررسی می کنیم.

### ۱-۱۶-۳ selectهای ۲ سطحی

در این حالت یک دستور select درون دستور select دیگری تعریف می شود. نتایج خروجی فقط می تواند شامل سطرهایی از جدول Select بیرونی باشد. شکل کلی select دو سطحی به صورت زیر است:

نام ستونهای مورد نظر برای نمایش از جدول بیرونی select

نام جدول ۱ from

نام فیلد مشترک دو جدول (Select) in نام فیلد مشترک دو جدول where

نام جدول ۲ From

; ( ... شرطها Where

مثالهای قسمت های قبل پیوند جدولها را با استفاده از Selectهای تودرتو باز نویسی می کنیم

مثال: نام و نام خانوادگی دانش آموزانی را ارائه دهید که در ترم ۱ سال تحصیلی ۸۹-

۸۸، درس با کد ۱۰۱۲ را انتخاب کرده اند:

کد:

```
select Name , Family
```

```
from Student
```

```
where StudentID in ( select StudentID
```

```
from Selection
```

```
where CourseID = 1012 and Term = 1 and Year = ' 88 - 89 ' );
```

خروجی

Name	Family
Ehsan	Amiri
Ahmad	Rezaee

**نکته** تفاوت این حالت با مثال‌های قبلی در این است که ستون‌های نمایش داده شده فقط از یک جدول (جدول select بیرونی) نمایش داده می‌شوند.

**مثال:** شماره و نام درس‌هایی را ارائه دهید که دانش‌آموز با شماره دانش‌آموزی ۱۰۲۵۴۸۶۱ انتخاب کرده است:  
کد:

```
select CourseID , Co Title
from Courses
where CourseID in ( Select CourseID
                    from Selection
                    where StudentID = 10254861 ) ;
```

خروجی

CourseID	CourseTitle
1011	پایگاه داده
1013	زبان تخصصی

## ۱۷-۳ دستور union

از دستور Union برای ترکیب و ادغام دو یا چند ستون مختلف از ۲ یا چند جدول استفاده می‌شود. دستور union نتیجه را در یک ستون مشترک نمایش می‌دهد. در این دستور، نوع داده ستون‌های انتخاب شده برای ادغام باید یکسان باشند. دستور union در هنگام ترکیب فیلدها، در صورت برخورد با مقادیر تکراری، آن‌ها را حذف کرده و از هر مقدار یک نمونه را نمایش می‌دهد. برای مشاهده تمام مقادیر، حتی مقادیر تکراری باید از دستور union all استفاده کرد.

شکل کلی استفاده از این دستور به صورت زیر است:

( نام فیلد ۱ انتخابی ) SQL Statement 1

union


( نام فیلد ۲ انتخابی ) SQL Statement 1 ;

مثال: در جدول ۱-۳، شماره دانش آموزی دانش آموزانی را نمایش دهید که یا دانش آموز رشته Electronic بوده و درس سخت افزار با کد ۱۰۱۲ را انتخاب کرده اند یا در درس زبان تخصصی با کد ۱۰۱۳ ثبت نام کرده اند:

```
select StudentID from Student
where Major = 'Electronic' and CourseID = 1012
union
select StudentID from Selection ;
where CourseID = 1013 ;
```

خروجی

StudentID
10254861
10724113
27365187
35654415

**نکته**  می توان در دستور union بیش از یک ستون مجزا را تعریف کرد. فقط باید تعداد ستون ها و ترتیب نوع داده ای ستون های تعریف شده یکسان باشند. برنامه مقادیر ستون ها را به ترتیب با هم پیوند می دهد.

مثال: شماره دانش آموزی، نام و نام خانوادگی دانش آموزانی را نمایش دهید که در رشته Electronic تحصیل کرده یا حداقل یک درس از نوع نظری را انتخاب کرده باشند:

```
select StudentID , Name , Family
from Student
where Major = 'Electronic'
union
select StudentID , Name , Family
from Student , Selection , Courses
where Student.StudentID = Selection.StudentID
and Selection.CourseID = Courses.CourseID
and CoType = 'نظری' ;
```

### خروجی

StudentID	Name	Family
10254861	Zahra	Hoseini
107241113	Ehsan	Amiri
27365187	Sahar	Ahmadi
35654415	Hesam	Razavi

## ۱۸-۳ دستور drop

از این دستور برای حذف یک جزء در محیط پایگاه داده استفاده می‌شود.

### ۱-۱۸-۳ حذف اندیس (drop index)

از این دستور برای حذف یک اندیس موجود در یک فیلد جدول استفاده می‌شود. شکل کلی این دستور به شرح زیر است:

نام جدول on نام اندیس drop index

مثال: drop index Rowup on Student ;

### ۲-۱۸-۳ حذف جدول یا پایگاه داده (drop DataBase یا drop table)

از این دستور برای حذف یک جدول در پایگاه داده یا یک پایگاه داده به صورت کامل استفاده می‌شود.

دستور حذف جدول:

نام جدول drop table ;


دستور حذف پایگاه داده:

مثال drop table Student ;

نام پایگاه داده drop DataBase ;

مثال: اگر پایگاه داده‌ای به نام School داشته باشیم، دستور حذف آن به شکل زیر است:

drop DataBase School ;

**نکته**  برای حذف تمام اطلاعات موجود در یک جدول بدون حذف ستون‌های

آن از دستور زیر استفاده می‌شود:

نام جدول truncate table ;

مثال truncate table School ;

این دستور باعث خالی شدن جدول از اطلاعات می شود، بدون این که خود جدول و ساختارش تغییری کند.

## ۱۹-۳ دستور alter

از دستور alter برای اضافه کردن یا حذف کامل یک ستون از یک جدول موجود در پایگاه داده استفاده می شود.

### ۱۹-۳-۱ اضافه کردن یک ستون

برای اضافه کردن یک ستون (فیلد) جدید به یک جدول موجود از دستور کلی زیر استفاده می شود:

نام جدول alter table

; نوع داده ستون نام ستون add

مثال: اضافه کردن یک ستون جدید به نام Age به جدول Student:

#### فیلدهای جدول Student قبل از اجرای دستور

Student Table					
StudentID	Name	Family	Major	Area	Grade

کد:

alter table Student

add Age Int ;

#### فیلدهای جدول Student پس از اجرای دستور

Student Table						
StudentID	Name	Family	Major	Area	Grade	Age

### ۱۹-۳-۲ حذف ستون

برای حذف یک ستون از یک جدول موجود به روش زیر عمل می شود:

نام جدول alter table

; نام ستون drop column

مثال: حذف فیلد Age از جدول Students:

جدول Student قبل از اجرای دستور

Student Table						
StudentID	Name	Family	Major	Area	Grade	Age

کد:

```
alter table Student
```

```
drop column Age ;
```

جدول Student پس از اجرای دستور

Student Table					
StudentID	Name	Family	Major	Area	Grade

## ۲۰-۳ توابع درون ساخته SQL

زبان SQL دارای یکسری توابع درون ساخته (آماده) برای محاسبات روی داده‌های جداول و ارائه خروجی مناسب برای اهداف خاص است. در هنگام استفاده از توابع SQL باید به نکات زیر دقت کرد: باید نوع داده هر ستون (فیلد) جدول متناسب با عملکرد تابع مورد استفاده باشد. برای مثال توابعی که اعمال ریاضی انجام می‌دهند، باید به فیلدهایی از نوع عددی اعمال شوند. می‌توان نتایج عملکرد هر تابع را با استفاده از ویژگی alias که قبلاً در مورد آن صحبت کردیم، به صورت یک ستون جدید با عنوان دلخواه نمایش داد. برای مثال می‌توان با استفاده از تابع avg، میانگین نمرات هر دانش‌آموز را محاسبه کرده و آن را در یک ستون جدید به نام average نمایش داد. این مورد را در مثال‌های ذکر شده بررسی می‌کنیم. برنامه، نتایج خروجی هر تابع را به صورت یک ستون جدید نمایش می‌دهد. شکل کلی استفاده از توابع درون ساخته به صورت زیر است:

نام جدول from (نام فیلد) \*function select

; نام تابع = function \*

مثال: محاسبه و نمایش معدل دانش‌آموزان با استفاده از تابع Avg و نمایش آن تحت

ستون جدیدی به نام دلخواه:

average :

select avg ( Grade ) as Average from Student ;

**نکته** در هنگام استفاده از توابع SQL، چنانچه شرطها برای دستور select تعریف شده باشد، تابع فقط مقادیر فیلدهایی از هر ستون را در محاسبه دخالت می دهد که با شروط ارائه شده سازگاری داشته باشند.

برای مثال وقتی می گوئیم که معدل دانش آموزان رشته Computer را در درس پایگاه داده محاسبه کنید، برنامه فقط میانگین نمرات درس پایگاه داده دانش آموزان را محاسبه می کند و به نمرات سایر دروس کاری ندارد.

**نکته** چنانچه توسط ویژگی alias، عنوانی برای ستون جدید ایجاد شده تعیین نکنید، خود برنامه به صورت خودکار یک نام پیش فرض به آن اختصاص می دهد.

### ۱-۲۰-۳ تابع avg

تابع avg برای محاسبه میانگین اعداد موجود در یک فیلد به کار می رود. این تابع اعداد را با هم جمع کرده و آنها را به طور اتوماتیک بر تعدادشان تقسیم می کند و نتیجه را به عنوان خروجی برمی گرداند.

**نکته** این تابع باید با فیلدهایی که داده آنها از نوع عددی است، اعمال شود.

**مثال:** معدل نمرات دانش آموزان را از جدول Student محاسبه کرده و در ستونی به نام معدل نمایش دهید:

کد:

select avg ( Students.Grade ) as معدل from Student

خروجی

معدل
16.8

**مثال:** معدل نمرات دانش آموزان رشته Computer را در درس پایگاه داده محاسبه کرده و در ستونی به نام معدل نمایش دهید:

کد:

select avg ( Selection.Grade ) as معدل

```
from Students,Selection,Courses
where Students.StudentID = Selection.StudentID
and Selection.CourseID = Courses.CourseID
and Major = 'Computer'
and CoTitle = ' پایگاه داده ' ;
```

خروجی

معدل
11.5

تابع sum ۲-۲۰-۳ تابع

تابع sum، مجموع مقادیر اعداد در یک فیلد را محاسبه کرده و به عنوان خروجی برمی گرداند.

**نکته** این تابع باید با فیلدهایی که داده آنها از نوع عددی است، به کار رود.

**مثال:** از جدول انتخاب واحد (Selection) مجموع نمرات دانش آموزان را در درس پایگاه داده با کد ۱۰۱۱ محاسبه کنید:  
کد:

```
select Sum ( Selection.Grade ) as مجموع نمرات پایگاه داده
from Selection,Courses
and Selection.CourseID = Courses.CourseID
and Selection.CourseID = 1011 ;
```

خروجی

مجموع نمرات پایگاه داده
47

**مثال:** از جدول انتخاب واحد، مجموع نمرات دانش آموز حسام رضوی را محاسبه کنید:  
کد:

```
select Sum ( Selection.Grade ) as مجموع نمرات حسام رضوی
from Student,Selection
where Student.StudentID = Selection.StudentID
```



and Name = ' Hesam '

and Family = ' Razavi ' ;

خروجی

مجموع نمرات حسام رضوی

26

### ۳-۲۰-۳ تابع min

این تابع، کمترین مقدار موجود در بین مقادیر فیلدهای یک ستون را برمی گرداند.

**نکته** این تابع با ستون‌هایی که داده آن‌ها از نوع عددی یا متنی باشد، می‌تواند به کار رود. در فیلدها با مقادیر عددی تابع کوچک‌ترین عدد و در فیلدها با مقادیر متنی، تابع کلمه‌ای را که به ترتیب حروف الفبا از آ تا ی در فارسی و A تا Z در انگلیسی در بالاترین رده باشد برمی گرداند.

**مثال:** کوچک‌ترین نام خانوادگی از نظر حروف الفبایی را از جدول Student نمایش

دهید:

کد:

select min (Family) as Family from Student ;

خروجی

Family

Ahmadi

**مثال:** کمترین نمره دانش‌آموزان را از جدول Selection نمایش دهید:

کد:

select min ( Grade ) as کمترین نمره from Selection ;

خروجی

کمترین نمره

8

### ۴-۲۰-۳ تابع max

این تابع، بیشترین مقدار موجود در بین مقادیر فیلدهای یک ستون را برمی گرداند.

**نکته** این تابع با ستون‌هایی که داده آن‌ها از نوع عددی یا متنی باشد، می‌تواند به کار رود. در فیلدها با مقادیر عددی تابع بزرگ‌ترین عدد و در فیلدها با مقادیر متنی، تابع کلمه‌ای را که به ترتیب حروف الفبا از آ تا ی در فارسی و A تا Z در انگلیسی در آخرین رده باشد برمی گرداند.

مثال: آخرین نام خانوادگی دانش آموزان را از جدول Student نمایش دهید:

کد:

```
select max (Family) as Family from Student;
```

خروجی

Family
--------

Rezaee
--------

مثال: بیشترین نمره دانش آموزان را از جدول Selection نمایش دهید:

کد:

```
select max (Grade) as بیشترین نمره from Selection ;
```

خروجی

بیشترین نمره
--------------

19
----

### ۵-۲۰-۳ تابع first

این تابع مقدار اولین رکورد را در یک فیلد برمی گرداند. ترتیب در اینجا، همان ترتیب قرارگیری رکوردها در جدول است.

### ۶-۲۰-۳ تابع last

این تابع مقدار آخرین رکورد را در یک فیلد برمی گرداند. ترتیب در اینجا، همان ترتیب قرارگیری رکوردها در جدول است.

### ۷-۲۰-۳ تابع count

این تابع تعداد سطرهای موجود در یک فیلد را شمرده و مقدار آن را برمی گرداند. به این تابع، تابع سطر شمار هم می گویند.

مثال: این تابع فیلدهایی را که مقدار آن‌ها خالی یا تکراری باشد، نمی شمارد. برای شمارش تمام فیلدها (حتی تکراری) باید قبل از نام ستون در دستور count عبارت distinct ذکر شود.

مثال: تعداد درس‌هایی که دانش آموز با شماره دانش آموزی ۳۵۶۵۴۴۱۵ قبول شده را

ارائه دهید:

کد:

```
select count (CourseID) as تعداد دروس
```

from Selection

where StudentID = 35654415

and Grade > 10 ;

خروجی

تعداد دروس

1

مثال: تعداد دانش‌آموزانی که درس پایگاه داده را انتخاب کرده‌اند اعلام کنید:

کد:

select count (distinct Selection.CourseID) as تعداد دانش‌آموزان

from Selection , Courses

where Selection.CourseID = Courses.CourseID

and CoTitle = ' پایگاه داده ' ;

خروجی

تعداد دانش‌آموزان

4

### ۸-۲۰-۳ دستور group by

در هنگام استفاده از برخی توابع درون ساخته SQL که عمل محاسبه (مثل مجموع و میانگین) را روی داده‌ها انجام می‌دهند، این مشکل وجود دارد که این توابع قادر به جدا کردن و متمایز کردن اطلاعات موجود در دو ستون نسبت به هم نیستند و نتایج محاسبات را به صورت کلی برای همه آن‌ها در نظر می‌گیرند.

از دستور group by برای دسته‌بندی یک ستون بر حسب مقادیر مشابه فیلدهای یک ستون دیگر استفاده می‌شود.

شکل کلی استفاده از این دستور به صورت زیر است:

select .... و نام فیلد ۲ و نام فیلد ۱

from نام جدول

group by نام فیلد ;

مثال ۱: فرض کنید جدولی به صورت زیر برای نمرات ۲ دانش‌آموز داریم، می‌خواهیم

مجموع نمرات هر یک از این دو دانش‌آموز را نمایش دهیم:

جدول ۳-۹

Class Table	
Grade	Name
18	Ali Ahmadi
17	Reza Hoseini
20	Ali Ahmadi
13	Reza Hoseini

کد:

نمرات مجموع as `sum (Grade)`

from Class ;

در صورت عدم استفاده از دستور `group by` تابع `Sum` خروجی زیر را خواهیم داشت:

خروجی

مجموع نمرات
68

• مسلماً خروجی فوق مورد نظر نیست. می توان با استفاده از دستور `group by`، مجموع نمرات را برای دانش آموزان بر حسب ستون `Name` به صورت جدا محاسبه کرد و به این صورت نمایش داد:

کد:

مجموع نمرات as `sum (Grade)` , `Name`

from Class

group by Name ;

خروجی

Name	مجموع نمرات
Ali Ahmadi	38
Reza Hoseini	30

مثال ۲: از جدول `Student` و `Selection` مجموع نمرات هر دانش آموز را به همراه نام و نام خانوادگی که بر حسب نام خانوادگی مرتب شده باشد نمایش دهید:

کد:

مجموع نمرات as `sum (Selection.Grade)` , `Name` , `Family`

```
from Student , Selection
where Student.StudentID = Selection.StudentID
group by Name , Family
order by Family;
```

خروجی

Name	Family	مجموع نمرات
Sahar	Ahmadi	35
Ehsan	Amiri	25
Zahra	Hoseini	21
Hesam	Razavi	26
Ahmad	Rezaee	33

### ۹-۲۰-۳ دستور having

از این دستور برای افزودن شرط به توابع درون ساخته SQL استفاده می شود، زیرا از دستور where نمی توان برای کار با مقادیر خروجی توابع درون ساخته SQL استفاده کرد. به عبارت دیگر having برای اعمال شرط به ستون‌ها اعمال می شود و همان کاری را می کند که where در رکوردها انجام می دهد. با مثال‌های بعدی استفاده از دستور having را بررسی می کنیم:

 **نکته** دستور having معمولاً با group by می آید.

**مثال:** از جدول Selection و Student مجموع نمرات و نام و نام خانوادگی دانش‌آموزانی را که مجموع نمرات آن‌ها بیش از ۲۵ است نمایش دهید:  
 کد:

```
select Name , Family , sum (Selection.Grade) as مجموع نمرات
from Student , Selection
where Student.StudentID = Selection.StudentID
group by Name , Family
having Sum (Selection.Grade) > 25
order by Family;
```

خروجی

Name	Family	مجموع نمرات
Sahar	Ahmadi	۳۵
Ahmad	Rezaee	۳۳

**مثال:** از جدول Selection و Courses نام دروسی را ارائه دهید که ۴ بار توسط دانش آموزان انتخاب شده باشد:  
**کد:**

```
select CoTitle , Count (Selection.CourseID) as تعداد انتخاب
from Selection , Courses
where Selection.CourseID = Courses.CourseID
group by CoTitle
having Count (Selection.Grade) = 4 ;
```

خروجی

CoTitle	تعداد انتخاب
پایگاه داده	4


**۱۰-۲۰-۳ دستور select into**

از دستور select into در موارد زیر استفاده می شود:

- ایجاد یک Backup (نسخه پشتیبان) از یک جدول
- ایجاد یک آرشیو از رکوردهای یک جدول
- قرار دادن برخی از رکوردها یا فیلدهای مورد نظر از یک جدول در یک جدول جدید
- ایجاد یک نسخه پشتیبان از کل یک پایگاه داده در یک پایگاه داده جدید

شکل کلی استفاده از این دستور به صورت زیر است:

```
select [نام یک پایگاه داده جدید in] نام جدول جدید into نام فیلدهای مورد نظر
from نام جدول مبدأ ;
```

**نکته**  ذکر نام یک پایگاه داده جدید در دستور select into اختیاری است. چنانچه نامی در این قسمت ذکر نشود، برنامه نسخه پشتیبان را در همان پایگاه داده موجود ایجاد می کند و چنانچه نامی ذکر شود، برنامه نسخه پشتیبان را در پایگاه داده ذکر شده ایجاد می کند.

**مثال:** ایجاد یک Backup از جدول Student در یک جدول جدید:


```
select * into Student_Backup
from Student ;
```

**مثال:** ایجاد یک Backup از جدول Student در یک جدول دیگر در یک پایگاه داده جدید به نام Backup.mdb:

```
select * into Student_Backup in «Backup.mdb»
```

```
from Student ;
```

توجه: در مثال‌های زیر از اطلاعات جداول Selection، Courses، و Student که در صفحات قبل معرفی کردیم، استفاده می‌کنیم.

**نکته**  همچنین می‌توان فقط یک یا چند ستون یا رکورد را از یک جدول در یک جدول جدید وارد کرد. برای این کار باید نام آن فیلدها را پس از دستور select تعیین کرده و برای رکوردهای مورد نظر نیز شروط مورد نظر تعریف شود.

**مثال:** فیلدهای Name و Family را از جدول Student در یک جدول جدید به نام Names وارد کنید:  
 کد:

```
select Name , Family into Names
```

```
from Student ;
```

بعد از اجرای دستور فوق جدول جدیدی به نام Names با مشخصات بعد ایجاد خواهد شد.

خروجی

Name	Family
Ahmad	Rezaee
Ehsan	Amiri
Zahra	Hoseini
Sahar	Ahmadi
Hesam	Razavi

**مثال:** از جدول Student، مشخصات دانش‌آموزانی که نمره آن‌ها بیش از ۱۷ است را در جدول جدیدی به نام Highscores وارد کنید:  
 کد:

```
select * into Highscores
```

```
from Student
```

```
where Grade > 17 ;
```

بعد از اجرای دستور فوق جدول جدیدی به نام Highscores Table با مشخصات زیر ایجاد خواهد شد.

خروجی

Highscores Table					
StudentID	Name	Family	Major	Area	Grade
41252214	Ahmad	Rezaee	Computer	Tehran	18
35654415	Hesam	Razavi	Electronic	Tehran	19

مثال: مشخصات نام، نام خانوادگی، نام درس و نمره دانش آموزانی که در جدول انتخاب واحد، نمره آن‌ها بیش از ۱۵ بوده را در یک جدول جدید به نام Highgrades وارد کنید:  
 کد:

```
select Name , Family , CoTitle , Selection.Grade Into Highgrades
from Student , Selection , Courses
where Student.StudentID = Selection.StudentID
and Selection.CourseID = Courses.CourseID
and Grade > 15 ;
```

بعد از اجرای دستور فوق جدول جدیدی به نام Highgrades Table با مشخصات زیر ایجاد خواهد شد.

خروجی

Highgrades Table			
Name	Family	CoTitle	Grade
Ahmad	Rezaee	پایگاه داده	16
Sahar	Ahmadi	مباحث ویژه	19
Ahmad	Rezaee	مباحث ویژه	17
Sahar	Ahmadi	زبان تخصصی	16
Hesam	Razavi	زبان تخصصی	17



## واژه‌نامه

Average	میانگین
Count	تعداد
First	اولین
Group	گروه
Having	داشتن
Last	آخرین
Maximum	حداکثر
Minimum	حداقل
Sum	حاصل جمع

## خلاصه مطالب

- زبان SQL یک زبان ساخت یافته برای اتصال و ارتباط و کار با بانک‌های اطلاعاتی است.
- این زبانه با ارائه دستورات مختلف و کاربردی در تمام سیستم‌های مدیریت بانک‌های اطلاعاتی قابل دسترسی و استفاده است.
- برای اجرای دستورات SQL و مشاهده عملکرد آن‌ها، باید از یک سیستم مدیریت پایگاه داده‌ها استفاده شود.
- دستور create برای ایجاد یک جزء جدید در سیستم‌های پایگاه داده استفاده می‌شود.
- دستور select برای انتخاب و استخراج اطلاعات موردنظر از یک یا چند جدول و مشاهده نتایج استفاده می‌شود.
- دستور insert برای وارد کردن و ایجاد یک رکورد جدید در جدول استفاده می‌شود.
- دستور update برای تغییر اطلاعات موجود در یک فیلد جدول استفاده می‌شود.
- دستور delete برای حذف اطلاعات یک رکورد در جدول استفاده می‌شود.