

فهرستی از کلمات کلیدی زبان برنامه نویسی انتخاب شده نوشته شود و شرحی در مورد کاربرد آنها داده شود.

• SELECT

از این عبارت برای مشخص کردن ستون هایی که می خواهید داده های آن ها را از جدول مد نظر خود بازیابی و استخراج کنید را استفاده می شود.
در این مثال ستون های FirstName و LastName انتخاب شده اند.

```
SELECT FirstName, LastName
```

• FROM

از این عبارت برای مشخص کردن جدول یا جدول هایی که میخواهید توسط دستور SELECT از آنها داده هایی را بازیابی و استخراج کنید استفاده می شود.
در این مثال نام جدولی که میخواهیم داده های ستون های FirstName و LastName
آن را مشاهده کنیم، تعیین کرده ایم.

```
FROM Employees
```

• WHERE

از این عبارت برای فیلتر کردن، برقراری شرط ها و محدودیت ها روی فیلدهایی که میخواهید از جدول مد نظر خود بازیابی کنید استفاده می شود. یعنی تنها داده هایی نمایش داده می شوند که شرط برای آن ها برقرار بوده است.
در این مثال برای داده های FirstName و LastName که قرار است از جدول Employees استخراج و بازیابی شوند این شرط در نظر گرفته است که حتما Department آن ها معادل Sales باشد.

```
WHERE Department = 'Sales';
```

• INSERT

از این عبارت برای برای افزودن یک رکورد جدید به صورت سطری به جدول استفاده می شود. یعنی هنگام استفاده از این دستور نام جدول، ستون ها، و مقادیری که میخواهید برای آن ها تعریف کنید را مشخص می کنید.

در این مثال ابتدا نام جدول که معادل با Customers است نوشته شده، سپس نام ستون هایی که میخواهیم داده ی آنها را تعیین و به جدول اضافه کنیم را می نویسیم سپس از عبارت VALUES استفاده میکنیم و در نهایت مقادیر مد نظر خود را برای هر یکی از ستون ها یادداشت میکنیم.

```
INSERT INTO Customers (FirstName, LastName, Email, Phone)
VALUES ('John', 'Doe', 'johndoe@example.com', '123-456-7890');
```

• UPDATE

از این عبارت برای به روزرسانی و اصلاح رکوردهای یک جدول استفاده می شود. یعنی هنگام استفاده از این دستور نام جدول، ستون ها، و مقادیر جدیدی که میخواهید برای آن ها تعریف کنید را مشخص می کنید.

در این مثال قیمت محصول با شماره ایدی ۱۲۳ به ۱۹.۹۹ در جدول Products تغییر کرده است.

```
UPDATE Products
SET Price = 19.99
WHERE ProductID = 123;
```

• DELETE

از این عبارت برای حذف کردن یک یا چند رکورد از یک جدول استفاده می شود.
در این مثال سطری که شماره سفارش آن معادل ۴۵۶ بوده است از جدول Orders حذف شده است.

```
DELETE FROM Orders  
WHERE OrderID = 456;
```

• CREATE

از این عبارت برای ایجاد اشیا در پایگاه داده استفاده می شود. برای مثال برای ساخت و ایجاد یک جدول، تابع، procedure، trigger، و...
در این مثال یک جدول جدید به نام Products ساخته شده است. هنگام ساخت یک جدول جدید باید نام و نوع ستون های آن مشخص شده باشد.

```
CREATE TABLE Products (  
    ProductID INT PRIMARY KEY,  
    ProductName VARCHAR(255) NOT NULL,  
    Price DECIMAL(10, 2) NOT NULL,  
);
```

• ALTER

از این عبارت برای اصلاح و ایجاد تغییرات در یک جدول از پایگاه داده استفاده می شود.
نمونه ای از این تغییرات و اصلاحات می تواند افزودن، حذف کردن و... یک سطر از جدول باشد.

در این مثال یک ستون جدید به نام Discount به جدول Products اضافه شده است.
هنگام ایجاد یک ستون جدید باید نوع آن مشخص باشد که در این مثال نوع ستون Decimal، Discount می باشد طول آن معادل ۵ و دارای دو رقم اعشار می باشد.

```
ALTER TABLE Products  
ADD Discount DECIMAL(5, 2);
```

• DROP

از این عبارت برای حذف اشیا پایگاه داده مانند جدول، ویوها و.. استفاده می شود. بازگرداندن این عملیات غیر ممکن است. با حذف یک جدول تمامی داده های مربوط به آن از پایگاه داده حذف خواهند شد.

در این مثال جدول ObsoleteTable حذف شده است.

```
DROP TABLE ObsoleteTable;
```

• JOIN

از این عبارت برای برقراری ارتباط بین دو یا چند جدول استفاده می شود. برقراری این عبارت می تواند مشابه با ضرب کارتیزین و یا با استفاده از یک ستون مشترک بین هر دو جدولی انجام شود.

در این مثال دو جدول Orders و Customers بر اساس ستون مشترکشان یعنی CustomerID با یکدیگر JOIN شده اند و ستون های Orders.OrderID و Customers.FirstName از بین تمامی ستون های آنها انتخاب و مقادیر آنها نمایش داده شده است.

```
SELECT Orders.OrderID, Customers.FirstName,  
Customers.LastName, Orders.OrderDate, Orders.TotalAmount  
FROM Orders  
INNER JOIN Customers ON Orders.CustomerID =  
Customers.CustomerID;
```

• GROUP BY

از این عبارت برای خلاصه نشان دادن نتایج خروجی استفاده می شود. از این عبارت زمانی استفاده می شود که گروهی از سطر ها دارای مقدار داده ی یکسانی می باشند. هنگام استفاده از این دستور می توان از توابعی مانند sum, count و... استفاده کرد. در این مثال مجموع کل مقادیر ستون Revenue از جدول Sales توسط تابع Sum محاسبه شده و مطابق با Category ها گروه بندی شده اند. یعنی خروجی این قطعه کد شامل مجموع در آمد کتگوری های مختلف می باشد.

```
SELECT Category, SUM(Revenue) AS TotalRevenue
FROM Sales
GROUP BY Category;
```

• HAVING

این عبارت مانند عبارت WHERE می باشد و برای فیلتر کردن و برقراری شرط ها و محدودیت ها روی داده های استخراج شده استفاده می شود. تنها تفاوت این است که عبارت HAVING همواره همراه با عبارت GROUP BY ظاهر می شود. در این مثال مجموع کل مقادیر ستون TotalSales از جدول Sales توسط تابع Sum محاسبه شده و مطابق با Category و Month گروه بندی شده اند. یعنی خروجی این قطعه کد شامل کل فروش بر اساس کتگوری ها و ماه های مختلف می باشد با این تفاوت که خروجی تنها شامل نتایجی می باشد که مجموع کل فروش آن ها از ۱۰۰۰۰ بیشتر است.

```
SELECT Category, Month, SUM(TotalSales) AS
MonthlyTotalSales
FROM Sales
GROUP BY Category, Month
HAVING SUM(TotalSales) > 10000;
```

• ORDER BY

از این عبارت برای مرتب کردن نتایج به دست آمده به صورت صعودی، نزولی یا ترکیبی از هر دوی آن ها روی یک ستون استفاده می شود.

در این مثال ستون های ProductID, ProductName و Price از جدول Products انتخاب شده و خروجی بر اساس قیمت کالا ها به صورت نزولی مرتب شده است.

```
SELECT ProductID, ProductName, Price
FROM Products
ORDER BY Price DESC;
```

• DISTINCT

از این عبارت برای عدم نمایش سطر های تکراری از یک ستون استفاده می شود.
در این مثال مقادیر ستون City از جدول Customers انتخاب و بازیابی شده اند استفاده از عبارت DISTINCT باعث می شود که اگر مشتری ها از شهر های مشترک بودند نام هر شهر فقط یک بار نمایش داده شود یعنی در خروجی نتیجه تکراری وجود نداشته باشد.

```
SELECT DISTINCT City
FROM Customers;
```

• AS

از این عبارت برای ساخت نام مستعار برای ستون و جدول ها استفاده می شود. استفاده از این دستور باعث افزایش خوانایی و نتایج بهتر در هنگام جست و جو خواهد شد.
در این مثال زمانی که خروجی به کاربر نمایش داده می شود نام ستونی که مقدار فیلد SUM(Revenue) را معادل با TotalRevenue نمایش می دهد.

```
SELECT Category, SUM(Revenue) AS TotalRevenue
FROM Sales
GROUP BY Category;
```