**دستور Select**

دستور SELECT برای انتخاب استفاده می‌شود و یکی از ساده‌ترین دستورهای SQL است. یادگیری آن بسیار ضروری است، چرا که تقریباً در همه دستورهای دیگر استفاده می‌شود. بهتر است کلمات کلیدی SQL را به صورت حروف بزرگ بنویسید تا خواندن و درک دستورها آسان‌تر شود. همان‌طور که از نام این دستور مشخص است، از آن برای انتخاب داده‌ها از یک پایگاه داده استفاده می‌شود. مثال ساده‌ی استفاده این دستور مانند زیر است:

|  |
| --- |
| SELECT \* FROM table |

این دستور دو بخش دارد. بخش نخست (SELECT \*) مشخص می‌کند که چه ستون‌هایی را می‌خواهیم انتخاب کنیم. علامت ستاره به این معنی است که می‌خواهیم همه ستون‌های جدول را انتخاب کنیم. بخش دوم (FROM table) به موتور پایگاه داده اعلام می‌کند که می‌خواهیم داده‌ها را از کجا دریافت کنیم. در استفاده‌های عملی باید به‌جای کلمه «table»، نام جدولی را بنویسیم که قصد داریم داده‌ها را از آن دریافت کنیم.  
این دستور به نام دستور «انتخاب ستاره» مشهور است. استفاده از علامت ستاره روش خوبی برای درک محتویات کلی یک جدول است؛ اما استفاده از آن در برنامه‌های واقعی چندان توصیه نمی‌شود. زمانی که از دستور انتخاب ستاره استفاده می‌کنیم، ارائه داده‌هایی که درخواست شده است بر عهده موتور پایگاه داده قرار می‌دهیم. در این شرایط هیچ کنترلی بر روی ترتیب داده‌های بازگشتی وجود ندارد، بنابراین اگر ستون جدیدی به جدول اضافه شود، احتمالاً متوجه می‌شویم که متغیرهایی که در زبان برنامه‌نویسی تعریف کرده‌ایم، دیگر داده‌های صحیحی را نمایش نمی‌دهند. می‌توان به طور صریح مشخص کرد که چه ستون‌هایی از جدول باید بازگردانده شوند، مانند دستور زیر:

|  |
| --- |
| SELECT age, name FROM people |

اگر بخواهیم داده دیگری را انتخاب کنیم که در هیچ کدام از جدول‌های دیگر ذخیره نشده است، می‌توانیم به صورت زیر عمل کنیم:

|  |  |
| --- | --- |
| SELECT age, '1234' FROM people |  |

هر رشته که درون (‘ ‘) نوشته می‌شود، به‌جای نام ستون از جداول مربوط پایگاه داده بازگردانده می‌شود

**دستور Alias نام مستعار**

برای نام مستعار می‌توان از هر نامی استفاده کرد؛ اما ترجیح داده می‌شود که از حروف الفبا استفاده شود. پیش از نام ستون، این نام مستعار به صورت پیشوند استفاده می‌شود. نام مستعار (Alias) بلافاصله پس از آنکه اعلان شود به جدول انتساب می‌یابد. به‌جای اینکه از نام طولانی جدول استفاده کنیم، می‌توانیم یک نام ساده و آسان‌تر برای آن انتخاب کنیم تا راحت‌تر به خاطر بسپاریم، اما مزیت این دستور چیست؟ در مواردی که لازم است دستور انتخاب برای بیش از یک جدول استفاده شود، ممکن است به راحتی نام جداول و ردیف‌ها باعث سردرگمی شود و مشخص نشود که کدام ستون متعلق به کدام جدول است. در حالتی که دو جدول نام‌های ستون یکسانی داشته باشند، ممکن است که کوئری پایگاه‌داده بدون ارجاع صریح به نام یا نام مستعار جدول دچار اشکال شود.

|  |
| --- |
| SELECT age, '1234' , '1234' As c1 FROM people |

**دستورWhere**

دستور SELECT برای بازیابی داده‌ها بسیار عالی است؛ اما اگر بخواهیم نتایج را کمی بیشتر فیلتر کنیم، چکار باید بکنیم؟ اگر بخواهیم افرادی را که تنها چشمان آبی دارند بازیابی کنیم، باید چه‌کار کنیم؟ دستور انتخاب افرادی که در ماه فروردین متولد شده‌اند و به عنوان مکانیک کار می‌کنند چطور است؟ این جایی است که می‌توان از دستور WHERE استفاده کرد. این دستور امکان اعمال شروطی برای دستور SELECT را فراهم می‌کند و می‌توان آن را به سادگی به انتهای آن دستور اضافه کرد.

|  |  |
| --- | --- |
|  | SELECT age, name FROM people WHERE age > 10 |

**عملگر AND**

کوئری بالا محدود به افرادی شده است که بیشتر از ده سال سن دارند. می‌توان با استفاده از عملگر AND چند شرط را با هم ترکیب کرد

|  |
| --- |
| SELECT age, name FROM people WHERE age > 10AND age < 20 |

دستور AND دقیقاً به همان ترتیبی که در زبان معمولی استفاده می‌کنیم، عمل می‌کند. این دستور یک شرط دیگر را به عبارت ما اضافه می‌کند. در مثال فوق، داده‌های بازیابی شده می‌توانند هر رکوردی باشند، به شرط این‌که سن فرد بین 1۰ تا 2۰ باشد. از آنجایی که هیچ نتیجه‌ای مطابق شرط‌های دستور وجود نداشته است، هیچ داده‌ای نیز بازیابی نشده است.

**عملگر OR**

دستور دیگری که می‌توان همراه با دستور where استفاده کرد، عملگر OR است. مانند مثال زیر:

|  |  |
| --- | --- |
| SELECT age, name FROM people WHERE age > 10 OR name = 'Joe' |  |

این کوئری رکوردهایی را برمی‌گرداند که در آن‌ها سن فرد بالاتر از 10 باشد یا نام برابر با «Joe» باشد. در مثال فوق ما تنها از یک علامت مساوی استفاده کردیم. در بسیاری از زبان‌های برنامه‌نویسی برای بررسی هم ارزی، از دو علامت مساوی (==) استفاده می‌شود. این شرایط برای اغلب موتورهای پایگاه داده لازم نیست (البته در محیط‌های مختلف این وضعیت فرق میکند و بهتر است ابتدا این نکته بررسی شود).

**دستور Orderترتیب**

دستور Order برای مرتب‌سازی نتایج بازیابی شده استفاده می‌شود. استفاده از این دستور نیز آسان است، کافی است آن را به انتهای دستور خود اضافه کنید:

|  |
| --- |
| SELECT name, age FROM people ORDER BY age DESC |

در این دستور باید نام ستون و نوع ترتیب را که ASC برای ترتیب صعودی و DESC برای ترتیب نزولی است مشخص کرد. همچنین میتوان بر اساس چند ستون، مرتب‌سازی را انجام داد. مانند مثال زیر:

|  |  |
| --- | --- |
| SELECT name, age FROM people ORDER BY name ASC, age DESC |  |

عبارت ORDER BY احتمالاً وقتی با دستورهای دیگر ترکیب می‌شود، مفید خواهد بود. در همه کوئری‌ها، داده‌ها با یک ترتیب منطقی یا عددی بازیابی نمی‌شوند. این دستور، آن وضعیت را تغییر می‌دهد.

**نکته**

**تفاوت (\*)Count و (Count(column**

در (\*)Count ، همه Row‌ ها و مقادیرشان مورد جستجو قرار می‌گیرد، اما در (Count(column فقط مقادیر غیر Null ستون مورد نظر،مورد جستجو قرار می‌گیرد.

**با اجرای Script زیر خروجی عدد 90.34 می‌باشد**

|  |
| --- |
| DECLARE @Value1 DECIMAL(5,2) = 80.22  DECLARE @Value2 DECIMAL(5,2) =90.34  SELECT (0.5 \* ((@Value1 + @Value2) + ABS(@Value1 - @Value2))) AS MaxColumn |

اشکال در این روش این است که، اگر مقدار یکی از اعداد Null باشد،ماکزیمم بین دو عدد Null نمایش داده خواهد شد.