

SQL SERVER

درس : روش پژوهش و ارائه

استاد : زردتشیتان

پژوهشگر : نیما آدینه وند



فهرست

- ✓ مقدمه
- ✓ تاریخچه SQL
- ✓ SQL و مزایا آن
- ✓ انواع پایگاه داده
- ✓ انواع کلیدها
- ✓ DATA TYPE IN SQL
- ✓ توابع ستونی
- ✓ دستورات SQL
- ✓ چندجدولی ها
- ✓ منابع



مقدمه

SQL به خودی خود، یک سیستم مدیریت بانک اطلاعاتی با یک محصول مجزا نیست، بلکه جزء لاینفک یک سیستم مدیریت بانک اطلاعاتی است، به طوری که با قرار دادن دستورات SQL در برنامه‌های کاربردی می‌توان قابلیت برنامه‌ها را افزایش داد.



تاریخچه SQL

- زبان SQL در اوایل دهه 1970 توسط دونالد دی. چمبرلین و ریچارد اف. بویس طراحی شد
- این دونفر در شرکت IBM و در آزمایشگاه تحقیقاتی سانخوزه کار می‌کردند.
- SQL ابتدا با نام SEQUEL ساخته شد.



SQL و مزایای آن

SQL وسیله‌ای است برای سازماندهی، مدیریت و بازیابی داده‌هایی که توسط بانک اطلاعاتی در کامپیوتر ذخیره شده‌اند. SQL یک زبان کامپیوتری است که برای کار با یک بانک اطلاعات از آن استفاده می‌شود. SQL در حقیقت با یک بانک اطلاعاتی موسوم به بانک اطلاعاتی رابطه‌ای کار می‌کند.



انواع پایگاه داده

- (1) پایگاه داده رابطه‌ای (Relational Database)
- (2) پایگاه داده غیررابطه‌ای (NoSQL)
- (3) پایگاه داده شی‌گرا (Object-Oriented Database)
- (4) پایگاه داده توزیع شده (Distributed Database)
- (5) پایگاه داده ابری (Cloud Database)
- (6) پایگاه داده زمان محور یا سری زمانی (Time-series Database)
- (7) پایگاه داده مبتنی بر حافظه (In-Memory Database)



انواع کلیدها

1. Primary Key (کلید اصلی)
2. Foreign Key (کلید خارجی)
3. Candidate Key (کلید نامزد)
4. Unique Key (کلید یکتا)



DATA-TYPE IN SQL

DATA TYPE	Numeric Types
INT	عدد صحیح
SMALLINT	عدد صحیح کوچک
BIGINT	عدد صحیح بزرگ
FLOAT / DOUBLE	اعداد اعشاری با دقت شناور
DECIMAL	اعداد اعشاری با دقت ثابت (مناسب قیمت)

DATA TYPE	STRING TYPES
CHAR(n)	رشته با طول ثابت
VARCHAR(n)	رشته با طول متغیر
TEXT	متن های طولانی

BOOLEAN	TRUE/FALSE
BLOB	ذخیره فایل/عکس، و داده های خام

DATA TYPE	DATE & TIME TYPES
DATE	تاریخ
TIME	زمان
DATETIME	تاریخ + زمان
TIMESTAMP	تاریخ + زمان با تغییر خودکار سیستم

```
CREATE TABLE Users (  
    id INT PRIMARY KEY,  
    name VARCHAR(50),  
    age INT,  
    salary DECIMAL(10,2),  
    created_at DATETIME  
);
```





توابع ستونی

```
SELECT  
    AVG(Quota + Sales) AS Avg_Quota_Sales  
FROM SALESREPS;
```

	Avg_Quota_Sales
1	3020

در SQL می‌توانید با استفاده از یک سری از توابع ستونی، داده‌ها را به یک شکل خلاصه شده از بانک اطلاعاتی استخراج نمایید. به عنوان مثال، پرس‌وجو زیر میانگین دو ستون از جدول SALESREPS را با استفاده از تابع ستونی `AVG()` محاسبه می‌کند.



محاسبه جمع یک ستون

تابع ستونی SUM() جمع مقادیر داده‌های یک ستون را محاسبه می‌کند. داده‌های این ستون باید از نوع عددی باشند.

```
SELECT  
    SUM(Sales) AS SUM_Sales  
FROM SALESREPS;
```

	SUM_Sales
1	6000



محاسبه حداقل و حداکثر مقادیر

توابع ستونی `MIN()` و `MAX()` به ترتیب، کوچکترین و بزرگترین مقادیر یک ستون را بدست می آورند. داده های ستون می توانند عددی، رشته ای یا اطلاعات مربوط به تاریخ و زمان باشند.

```
SELECT  
    MIN(Sales) AS MIN_Sales  
FROM SALESREPS;
```

```
SELECT  
    MAX(Sales) AS MAX_Sales  
FROM SALESREPS;
```

	MIN_Sales
1	2000

	MAX_Sales
1	4000



دستورات SQL

```
CREATE TABLE users ( id INT PRIMARY KEY, name VARCHAR(50) );
```

برای ساختن جدول، دیتابیس یا اشیای دیگر استفاده می‌شود.

```
INSERT INTO users (name) VALUES (22, 'nima');
```

برای اضافه کردن رکورد جدید به جدول.

```
UPDATE users SET age = 23 WHERE id = 1;
```

برای به روز رسانی مقدارهای موجود.

```
DELETE FROM users WHERE id = 1;
```

برای حذف رکورد از جدول.

زبان SQL دارای دستورات متنوع و نسبتاً زیادی می‌باشد.

به این دستورات QUERY نیز گفته می‌شود. در این مقاله و قسمتهای بعدی آن تعدادی از مهمترین QUERY های زبان SQL مورد بررسی قرار می‌گیرند.

دستورهای SELECT و FROM

```
select name, family from student;
```

	name	family
1	sina	mousvi
2	nima	adineh

هدف نهایی از دادن انبوه اطلاعات به کامپیوتر، جستجو و یافتن اطلاعات مفید می‌باشد. به این عمل یعنی جستجوی اطلاعات در بانک اطلاعاتی Query نیز گفته می‌شود. اکثر دستورات زبان SQL نیز در همین راستا مورد استفاده قرار می‌گیرند. در این بین مهمترین و پرکاربردترین دستور را می‌توان دستور SELECT قلمداد کرد.

این دستور جهت انتخاب یک یا چند فیلد از یک یا چند جدول مختلف مورد استفاده قرار می‌گیرد، فیلدهای انتخاب شده پس از اجرای query روی صفحه نمایش داده خواهند شد.

بعنوان مثال اگر در بانک اطلاعاتی فرضی خودمان بخواهیم اسم و فامیل تمام دانشجویان را مشاهده کنیم باید برنامه‌ای به شکل روبرو بنویسیم :



```
select name, family from student;  
  
select * from student;
```

همانطور که از مثال بالا نیز مشخص است جلوی دستور **SELECT** نام فیلدهایی نوشته می‌شوند که قرار است نمایش داده شوند و بعنوان جداکننده نیز باید از کاراکتر کاما استفاده کرد. در صورتی که بخواهیم تمام فیلدهای یک جدول را ببینیم می‌توانیم بجای نوشتن اسم تمام فیلدها فقط از یک کاراکتر ستاره استفاده کنیم. کاراکتر ستاره به معنی تمام فیلدهای یک جدول می‌باشد.

مثلا دو دستور زیر باهم معادل‌اند:



دستور WHERE

دستور SELECT همان طوری که گفته شد جهت انتخاب و نمایش تعدادی از فیلدهای جدول مورد استفاده قرار میگیرد. در انی حالت تمام رکوردها نمایش داده خواهند شد، ولی در بیشتر موارد هدف از نوشتن Query نمایش رکوردهایی است که دارای شرایط ویژه ای می باشند مثلا نمایش مشخصات اساتیدی که بیش از چهل سال سن دارند.

دستور WHERE برای گذاشتن یک یا چند شرط به دستور SELECT تعداد رکوردهای خروجی (پاسخ) معمولا محدودتر می شود. بعنوان مثال اگر بخواهیم مشخصات اساتیدی که بیش از چهل سال سن دارند را ببینیم باید برنامه ای بصورت زیر بنویسیم :

```
SELECT *  
FROM teachers  
WHERE age > 40;
```



```
SELECT *  
FROM students  
WHERE age < 15 AND name = 'ALI' ;
```

```
SELECT *  
FROM students  
WHERE name NOT IN ('ali', 'reza') ;
```

```
SELECT *  
FROM students  
WHERE family LIKE '% zadeh' ;
```

```
SELECT name, age  
FROM teachers  
WHERE name LIKE 'Pe%' ;
```



دستور ORDERBY

دستور ORDERBY جهت SORT کردن رکوردهای نمایش داده شده مورد استفاده قرار میگیرد. با این دستور می توان مشخص کرد که رکوردهایی که قرار است نمایش داده شوند برحسب کدام فیلد باید مرتب شوند. به عنوان مثال برای مشاهده کردن مشخصات اساتیدی که سن آنها بیشتر از 29 سال می باشد و در ضمن لیست خروجی بترتیب اسم فامیل نیز مرتب شده باشد باید Query زیر را اجرا کرد :

```
SELECT *  
FROM teachers  
ORDERBY family;
```

```
SELECT *  
FROM teachers  
WHERE age > 30  
ORDERBY family, name
```



QUERY چندجدولی ها

تمام Query هایی که تاکنون مشاهده کردید، Query های تک جدولی بودند بدین معنی که در هر Query فقط اطلاعات یک جدول مورد جستجو قرار می گرفت.

در SQL امکان نوشتن Query های چندجدولی نیز وجود دارد. در این حالت اطلاعات چند جدول بطور همزمان مورد جستجو قرار می گیرد و حتی امکان مقایسه فیلدهایی از یک جدول با فیلدهایی از جدول دیگر نیز وجود دارد.



```
SELECT *  
FROM students, teachers  
WHERE students.family = teachers.family
```

```
SELECT teachers.family, teachers.name  
FROM students, teachers  
WHERE teachers.age < students.age
```



نکته

زبان SQL برخلاف زبانهای مانند C یا C++ یک زبان Case Sensitive نیست بدین معنی که به بزرگ یا کوچک بودن حروف حساس نمی باشد. بنابراین به راحتی می توان دستورات و کلمات کلیدی آنرا در هر برنامه ای با حروف کوچک یا بزرگ تایپ کرد.



منابع

- <https://www.w3schools.com>
- <https://bugeto.net>
- <https://blog.faradars.org>
- <https://nikamooz.com/sql-server-commands>

<https://www.atrilya.com/Blog/ArticleDetails/%D8%A2%D8%B4%D9%86%D8%A7%DB%8C%DB%8C-%D8%A8%D8%A7-sql-server-management-studio-ssms>



سپاس از همراهی شما دوستان عزیز

