



دستور Having در SQL Server در سناریوهای مختلفی از جمله تجزیه و تحلیل فروش، بررسی عملکرد کارمندان، تحلیل بودجه و سایر موارد، قابل استفاده است. فرض کنید در شرایطی لازم باشد مشتریانی شناسایی شوند که تعداد سفارش‌هایشان از حد خاصی بیش‌تر است؛ در چنین شرایطی، می‌توانید با کمک Having، نتایج را براساس مقادیر تجمعی محاسبه‌شده فیلتر کنید. پیش‌تر **دستور Select** و **دستور آیدیت**، به همراه **دستور Insert** و **نحوه ایجاد جدول** و **ساخت View** را مورد بررسی قرار دادیم و در این مطلب، قصد داریم Having را به همراه مثال، مورد بررسی قرار دهیم. علاوه بر این، می‌توانید برای یادگیری مباحث پراهمیت حوزه SQL Server و کوئری‌نویسی، به **آموزش جامع SQL Server** نیز رجوع کنید.

دستور HAVING در SQL Server چیست؟

عبارت HAVING در SQL Server، یک جزء مهم از **زبان کوئری نویسی T-SQL** محسوب می‌شود و به واسطه آن، امکان فیلترکردن داده‌های تجمعی (Aggregated Data) فراهم می‌شود. برخلاف عبارت WHERE که قبل از هر Grouping یا Aggregation، فیلترسازی تک تک سطرها را انجام می‌دهد، در دستور HAVING، پس از فرآیند گروه‌بندی و تجمیع، به فیلترسازی گروه‌هایی از سطرها می‌پردازد. به همین دلیل، دستور HAVING به‌طور خاص برای اعمال شرط روی داده‌های تجمعی مفید است.

کاربردهای دستور HAVING در SQL Server

چند کاربرد دستور HAVING در SQL Server در ادامه بررسی می‌شوند:

- **فیلترسازی نتایج تجمعی:** عبارت HAVING در SQL Server، امکان فیلترکردن گروه‌هایی از سطرها را براساس توابع تجمعی مانند SUM، AVG، COUNT، MIN و MAX فراهم می‌کند. به عنوان مثال، شما می‌توانید با کمک دستور HAVING، گروه‌هایی با مجموع فروش بالاتر از یک مقدار آستانه مشخص را فیلتر کنید.
- **خلاصه‌سازی داده:** با ترکیب دستور HAVING در SQL Server به همراه عبارت GROUP BY، می‌توانید داده‌ها را خلاصه‌سازی کرده و گروه‌ها را براساس معیارهای مشخصی فیلتر کنید. این عمل به منظور تهیه گزارش‌ها و تحلیل ترندهای مربوط به دیتا مفید است.
- **شناسایی ناهنجاری‌ها:** عبارت HAVING می‌تواند به شناسایی ناهنجاری‌ها (Anomalies) یا داده‌های پرت (Outlier Data) در داده‌های تجمعی کمک کند. برای مثال، می‌توان از HAVING برای پیدا کردن گروه‌هایی با مقادیر بالا یا پائین غیرمعمول نسبت به سایر داده‌های مجموعه استفاده کنید.



تفاوت بین WHERE و HAVING

تفاوت عبارات WHERE و HAVING در SQL Server، زمان بندی و محدوده عملکرد آنها است. عبارت WHERE را که در [این مقاله](#) بررسی کرده ایم، داده های خام پیش از خلاصه سازی، فیلتر می شوند. این عبارت براساس شروط تعیین شده، تک تک ردیف ها را پیش از هر گروه بندی یا تجمیع، فیلترسازی و روی مجموعه داده اصلی عمل می کند. در حالی که عبارت HAVING، این گروه های ردیف ها را پس از فرآیند Grouping و Aggregation فیلتر می کند؛ به طوری که این عبارت پس از اعمال GROUP BY، روی مجموعه نتیجه اعمال شود. به عبارت دیگر، در حالی که دستور WHERE برای فیلتر کردن داده های خام قبل از خلاصه سازی به کار می رود، دستور HAVING در SQL Server برای فیلتر کردن داده های تجمعی یا گروه هایی از ردیف ها پس از خلاصه سازی یا گروه بندی آنها مورد استفاده قرار می گیرد. این تمایز، نظارت دقیق تر بر تحلیل داده ها را ممکن می کند.

نحوه استفاده از دستور HAVING در SQL Server

فرآیند فیلتر کردن نتایج حاصل از توابع تجمعی، مانند SUM و AVG، با استفاده از HAVING در SQL Server انجام می شود. در این بخش قصد داریم نحوه استفاده از دستور Having در SQL Server را به همراه مثال شرح دهیم.

- **شروع با عبارت SELECT:** در وهله اول، یک عبارت SELECT بنویسید که در آن از توابع تجمعی برای اعمال محاسبات روی داده های گروه بندی شده استفاده شود.
- **افزودن عبارت GROUP BY:** از عبارت GROUP BY به منظور تعیین ستون هایی استفاده کنید که براساس آنها می خواهید داده ها گروه بندی شوند.

- **استفاده از HAVING برای فیلترینگ گروه‌ها:** پس از عبارت GROUP BY، از دستور HAVING در SQL Server استفاده کنید. بدین شیوه، می‌توانید براساس شروط خاص، گروه‌های ایجادشده را فیلتر کنید. معمولاً این شروط، توابع تجمعی را شامل می‌شوند و می‌توان در آن‌ها از عملگرهای منطقی مانند AND، OR و NOT استفاده کرد.
- **تعیین شروط فیلترسازی در HAVING:** در دستور HAVING، کاندیشن‌ها یا همان شروط فیلترینگ مدنظر را ذکر کنید. شما باید مطمئن شوید که شروط روی نتایج تجمعی اعمال می‌شوند و نه تک‌به‌تک ردیف‌های جدول.
- **مرتب‌سازی نتایج:** در صورت نیاز، امکان استفاده از ORDER BY برای مرتب‌سازی مجموعه نتایج نهایی براساس یک ستون خاص وجود دارد.
- **اجرای کوئری:** پس از تکمیل مراحل فوق، می‌توانید کوئری خود را اجرا کنید. در چنین شرایطی، شما باید نتایج فیلترشده براساس معیارهای تعیین‌شده در HAVING را مشاهده کنید.

برای درک اقدامات فوق، به مثال زیر توجه کنید:

```
SELECT
    CategoryID,
    COUNT(ProductID) AS TotalProducts
FROM
    Products
GROUP BY
    CategoryID
HAVING
    COUNT(ProductID) > 5;
```

کوئری فوق برای یافتن دسته‌بندی‌های محصولات با حداقل ۵ محصول استفاده می‌شود. در این کوئری، Select کردن CategoryID از جدول Products انجام می‌شود. سپس با استفاده از تابع Count، تعداد محصولات هر Category شمرده شده و طبق CategoryID گروه‌بندی خواهند شد. درنهایت، با اعمال دستور Having در SQL Server به فیلترکردن نتایج گروه‌بندی‌شده می‌پردازیم. بدین شیوه، تنها Category هایی نمایش داده می‌شوند که مجموع تعداد محصولات آن‌ها، بزرگ‌تر از ۵ باشد.

برای درک بهتر، به خروجی حاصل از اجرای این کوئری توجه کنید:

The screenshot shows a SQL query window with the following text:

```

SELECT
    CategoryID,
    COUNT(ProductID) AS TotalProducts
FROM
    Products
GROUP BY
    CategoryID
HAVING
    COUNT(ProductID) > 5;
    
```

Below the query window, a results grid is displayed with the following data:

	CategoryID	TotalProducts
1	1	14
2	2	12
3	3	13
4	4	10
5	5	7
6	6	6
7	8	12

ترکیب HAVING با گروه بندی (GROUP BY)

ترکیب عبارات GROUP BY و HAVING در SQL Server رویکرد قدرتمندی برای گروه بندی داده ها در سطرهای خلاصه و سپس فیلتر کردن آن گروه ها براساس شرط است. **عبارت GROUP BY** برای گروه بندی ردیف های موجود در نتیجه یک کوئری براساس یک یا چند ستون استفاده می شود. به بیان دیگر، این عبارت رکوردها را براساس مقادیر موجود در ستون های مشخص شده، دسته بندی و خلاصه سازی می کند. معمولاً GROUP BY همراه با توابع تجمعی مانند SUM، AVG، COUNT، MIN و MAX به کار می رود. از سوی دیگر، دستور HAVING در SQL Server، مشابه یک فیلتر عمل می کند؛ برخلاف WHERE که در آن سطرها قبل از گروه بندی فیلتر می شوند، در HAVING، فیلتر پس از گروه بندی اعمال خواهد شد.

تصور کنید باید مشخص شود که کدام تأمین کنندگان در لیست محصولات شما، بیش از ۳ محصول متفاوت ارائه کرده اند. برای دستیابی به این موضوع، می توان از کوئری زیر با استفاده از عبارات GROUP BY و HAVING استفاده کرد:

```

SELECT
    SupplierID,
    COUNT(ProductID) AS NumberOfProducts
FROM
    Products
GROUP BY
    SupplierID
HAVING
    COUNT(ProductID) > 3;
    
```

```

SELECT
    SupplierID,
    COUNT(ProductID) AS NumberOfProducts
FROM
    Products
GROUP BY
    SupplierID
HAVING
    COUNT(ProductID) > 3;
    
```

	SupplierID	NumberOfProducts
1	1	5
2	2	4
3	7	5
4	8	4
5	12	5

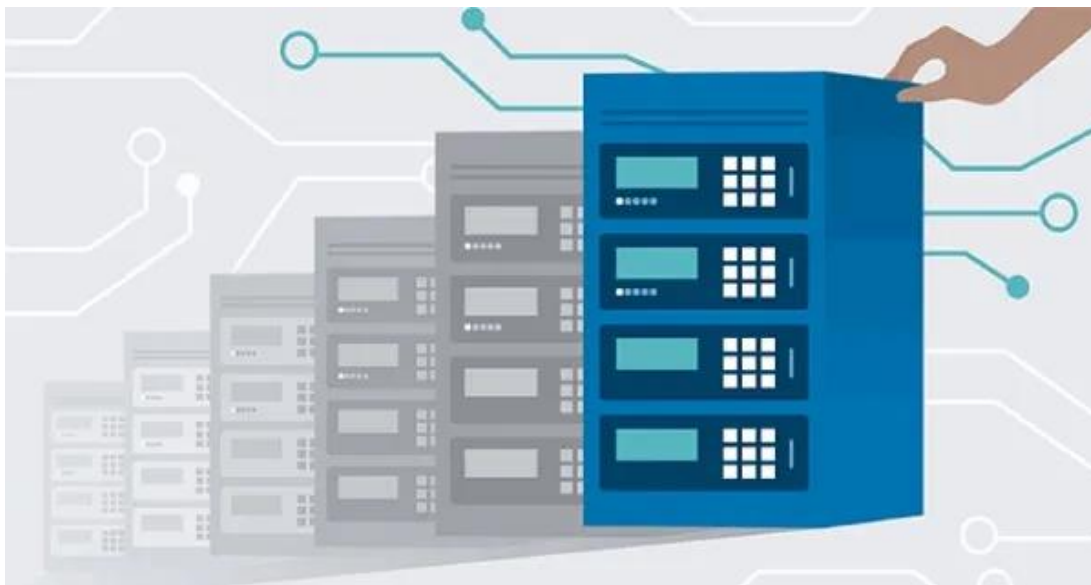
نکات و ترفندهای حرفه ای برای بهینه سازی کوئری های HAVING

اگر شما فردی هستید که می‌خواهید در **مسیر شغلی DBA** و موفقیت در آن قرار بگیرید، بهینه‌سازی کوئری‌ها، به‌ویژه آن‌هایی که دارای Having هستند، به منظور بهبود کارایی حائز اهمیت است. در این بخش، به بررسی برخی از تکنیک‌ها و ترفندهای مناسب برای بهبود کارایی این کوئری‌ها می‌پردازیم.

- **استفاده از ایندکس گذاری (Indexing) مؤثر:** اطمینان حاصل کنید که ستون‌های استفاده‌شده در GROUP BY ایندکس‌گذاری شده هستند. این کار می‌تواند با کاهش زمان لازم برای مرتب‌سازی و گروه‌بندی داده‌ها، فرآیند Grouping را به‌طور قابل‌توجهی سرعت ببخشد. علاوه‌براین، **ایندکس گذاری** برای ستون‌هایی که در عبارات SELECT ، GROUP BY و دستور HAVING در SQL Server استفاده می‌شوند، می‌تواند مفید باشد. با این حال، قبل از انجام این کار، لازم است مزایا و معایب را به‌طور کامل بررسی کنید.
- **فیلترینگ اولیه با WHERE:** در صورت امکان، قبل از Grouping، با به‌کارگیری عبارت WHERE، سطرها را فیلتر کنید. کاهش تعداد آن‌ها در مراحل اولیه فرآیند، می‌تواند به کاهش کارهای مربوط به مراحل گروه‌بندی و HAVING شده منجر شود و در نتیجه، عملکرد را بهبود بخشد.
- **بهینه‌سازی توابع تجمعی:** برخی از توابع تجمعی می‌توانند پرهزینه‌تر به حساب بیایند. به‌عنوان مثال، معمولاً COUNT(*) سریع‌تر از COUNT(column) است؛ زیرا برای مقادیر NULL بررسی انجام نمی‌دهد. از ساده‌ترین تابع تجمعی که نیازمندی‌های شما را برآورده می‌کند، استفاده کنید.
- **به حداقل رساندن ستون‌ها در GROUP BY:** هرچه ستون‌های بیشتری را در یک GROUP BY قرار دهید، Grouping پیچیده‌تر می‌شود. بنابراین، فقط ستون‌های ضروری را برای گروه‌بندی در نظر بگیرید تا محاسبات، ساده‌تر و سریع‌تر شوند.

- **اجتناب از عبارات پیچیده در HAVING:** عبارات و توابع پیچیده در دستور HAVING در SQL Server ممکن است سرعت کوئری را بکاهند. در صورت امکان، عبارات را ساده سازی کنید یا آن ها را در یک Subquery محاسبه کنید.
- **استفاده از Approximation Functions:** در سناریوهایی که اعداد دقیق حیاتی نیستند، استفاده از توابع تقریبی می تواند به بهبود قابل توجه در عملکرد منجر شود.
- **بازنویسی و بهینه سازی کوئری ها:** گاهی اوقات، بازنویسی کوئری ها می تواند عملکرد را بهبود دهد. به عنوان مثال، تقسیم یک کوئری پیچیده به کوئری های ساده تر و کوچک تر، می تواند مفید واقع شود.
- **مانیتورینگ و تحلیل عملکرد کوئری:** با استفاده از ابزارهای پروفایلینگ ارائه شده توسط سیستم مدیریت پایگاه داده، به طور منظم، کوئری ها را تجزیه و تحلیل کنید تا درک کنید گلوگاهها (Bottlenecks) در کجا رخ می دهند. Execution Plan ها را بررسی کنید تا متوجه شوید کوئری ها چگونه پردازش می شوند و عملیات ناکارآمدی را شناسایی کنید.
- **سخت افزار و پیکربندی:** اطمینان پیدا کنید که پیکربندی پایگاه داده برای حجم کاری، بهینه سازی شده است. پارامترهایی مانند تخصیص حافظه، اندازه Buffer Pool و تنظیمات اجرای موازی (Parallel Execution) می توانند تأثیر بسزایی بر عملکرد داشته باشند. استفاده از منابع سخت افزاری مناسب نیز می تواند به بهبود عملکرد کوئری ها کمک کند.

در مجموع، با اجرای این استراتژی ها، می توانید عملکرد کوئری هایی که از دستور HAVING در SQL Server استفاده می کنند، بهینه سازی کنید و کارایی کلی عملیات دیتابیس را در سازمان بهبود دهید.



رفع خطاهای رایج در استفاده از HAVING

در ادامه، به برخی از اشتباهات رایج در استفاده از دستور HAVING در SQL Server اشاره خواهد شد:

- **فراموش کردن عبارت GROUP BY:** یکی از اشتباهات رایج، فراموش کردن عبارت GROUP BY قبل از HAVING است. از آنجایی که HAVING روی داده‌های گروه‌بندی شده اعمال می‌شود، در صورت عدم وجود GROUP BY، خطا رخ خواهد داد.
- **استفاده از Column Alias:** ارجاع به Column Alias هایی در HAVING که در مرحله اجرای کوئری قابل دسترسی نیستند، می‌تواند به خطا منجر شود. مطمئن شوید که Column Alias ها در یک عبارت SELECT قبلی تعریف شده‌اند یا به جای آن‌ها از نام اصلی ستون استفاده کنید.
- **استفاده مستقیم از توابع تجمعی:** استفاده مستقیم از توابع تجمعی در HAVING و بدون گروه‌بندی اولیه داده‌ها، می‌تواند نتایج غیرمنتظره یا خطا به همراه داشته باشد.

جمع بندی: دلایل استفاده از دستور HAVING در SQL Server

به واسطه دستور HAVING در SQL Server، فیلترینگ و تجزیه و تحلیل پیشرفته داده‌ها انجام می‌شود. شما می‌توانید با درک عملکرد، کاربردها و روش استفاده از HAVING، از کوئری‌ها به بهترین نحو و برای استخراج بینش‌های ارزشمند از داده‌ها بهره‌مند شوید.