



عنوان مقاله: دستور inner join در SQL Server

نویسنده مقاله: تیم فنی نیک آموز

تاریخ انتشار: ۱۳ خرداد ۱۴۰۳

منبع: <https://nikamooz.com/inner-join-command-in-sql-server/>

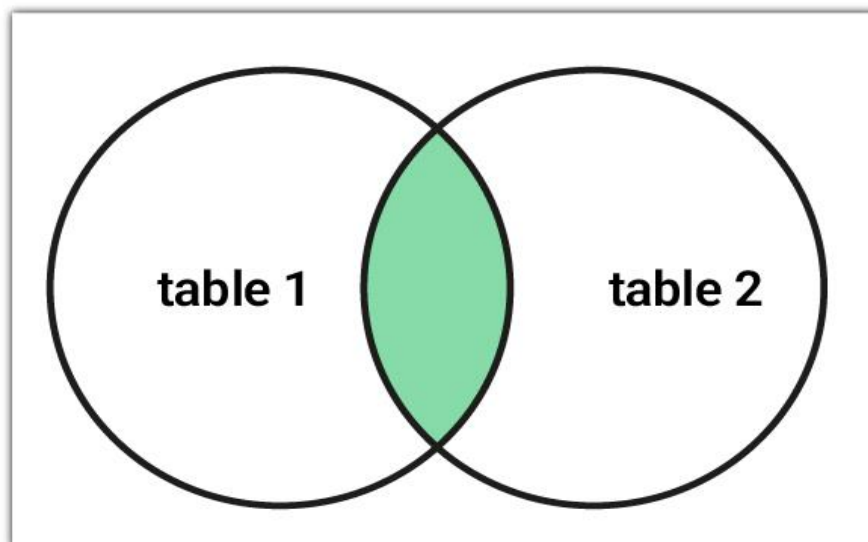
دستور Inner Join، قلب تپنده کوئری‌های پیچیده SQL Server است. این دستور به شما امکان می‌دهد تا الگوها و روندهای پنهان در داده‌های رابطه‌ای را کشف کرده و گزارش‌های دقیق و گویا تهیه کنید. در این مقاله، به بررسی جامع و کامل این دستور، نحوه استفاده و کاربرد آن خواهیم پرداخت. با خواندن این مطلب، به راحتی می‌توانید از دستور inner join در SQL Server استفاده کنید.

دستور JOIN و اهمیت آن در پایگاه داده

در دنیای پایگاه داده رابطه‌ای، برای بازیابی اطلاعات از چندین جدول مرتبط، از قابلیت Join استفاده می‌کنیم. Join به [SQL Server](#) می‌گوید چگونه از داده‌های یک جدول برای انتخاب سطرهای جدول دیگر استفاده کند. به زبان ساده، [دستورات join در SQL Server](#) به شما این امکان را می‌دهند تا داده‌های دو یا چند جدول را براساس ستون‌های مشترک، باهم ترکیب کنید.

دستور inner join در SQL Server

Inner Join روشی برای ترکیب کردن داده‌های دو یا چند جدول براساس یک ارتباط منطقی (معمولاً یک فیلد مشترک) است. این کار، یک نتیجه‌گیری موقت از ترکیب داده‌های دو جدول براساس شرط تعیین‌شده ایجاد می‌کند که فقط شامل سطرهایی است که در هر دو جدول، اطلاعات مطابق دارند.



فرض کنید در یک پایگاه داده، دو جدول مجزا با اطلاعات مرتبط به هم دارید. دستور inner join در SQL Server به شما این امکان را می‌دهد تا این جداول را براساس یک ستون مشترک، بهم وصل کرده و فقط داده‌های مرتبط را نمایش دهید.

- جدول اول به نام Customers شامل مشخصات مشتریان است (نام، نام خانوادگی، شماره تماس)

- جدول دوم به نام Orders شامل لیست سفارشات مشتریان است (شناسه سفارش، کد محصول، تاریخ سفارش). اکنون می‌خواهیم فهرستی از نام مشتریان به همراه سفارشات آن‌ها تهیه کنیم.

دستور inner join در SQL Server دو جدول را براساس یک ستون مشترک بهم وصل می‌کند.

SELECT

```
Customers.Name, Orders.OrderID
```

```
FROM Customers
```

```
INNER JOIN Orders
```

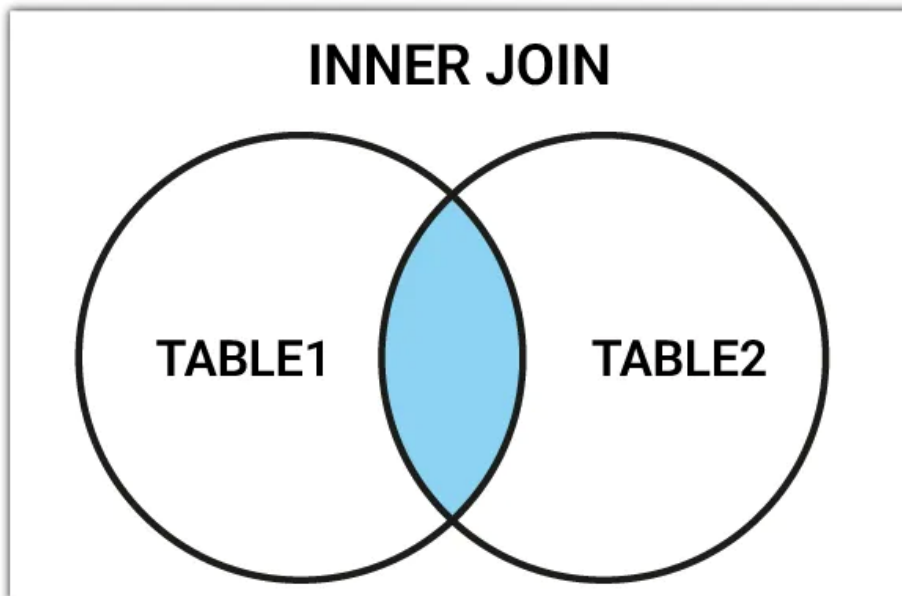
```
ON Customers.CustomerID = Orders.CustomerID;
```

به زبان ساده، این دستور تک تک سطرهای جدول Customers را بررسی می‌کند و در جدول Orders به دنبال سطر می‌گردد که مقدار ستون CustomerID آن با مقدار ستون CustomerID در حال بررسی در جدول Customers مطابقت داشته باشد. اگر مطابقت پیدا شود، آن سطر از هر دو جدول برداشته شده و در خروجی نهایی قرار می‌گیرد.

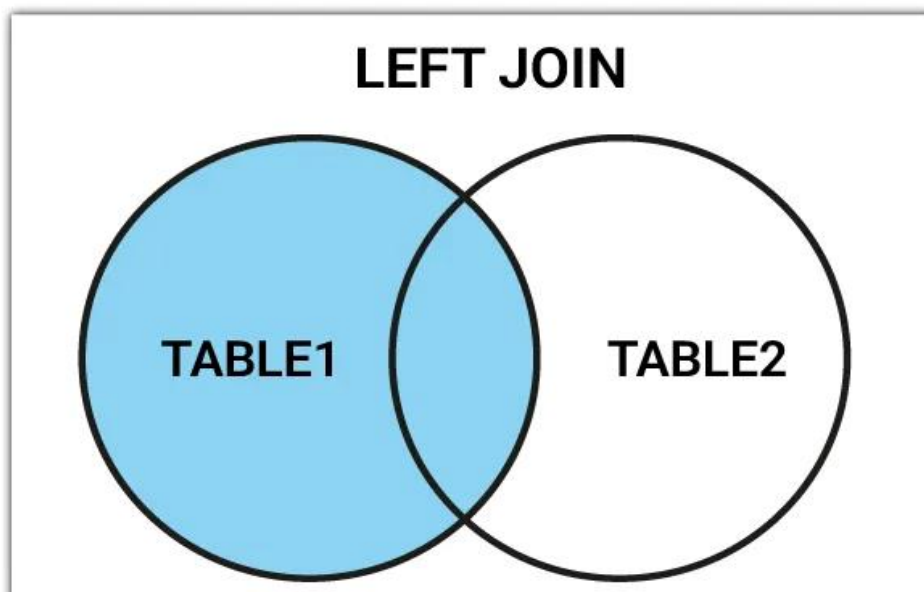
تفاوت دستور inner join با انواع join

Inner Join یکی از انواع مختلف Join در SQL Server است. انواع مختلفی از Join وجود دارد که با استفاده از دستورات زیر قابل استفاده هستند:

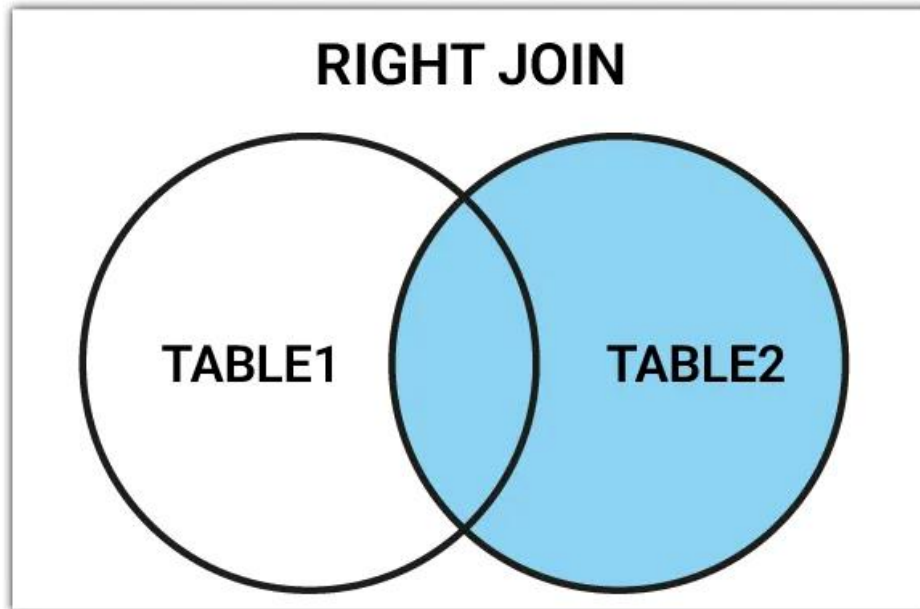
- **INNER JOIN**: نتایج این دستور فقط شامل سطرهایی است که در هر دو جدول دارای مقادیر مطابقت در ستون مشترک باشند.



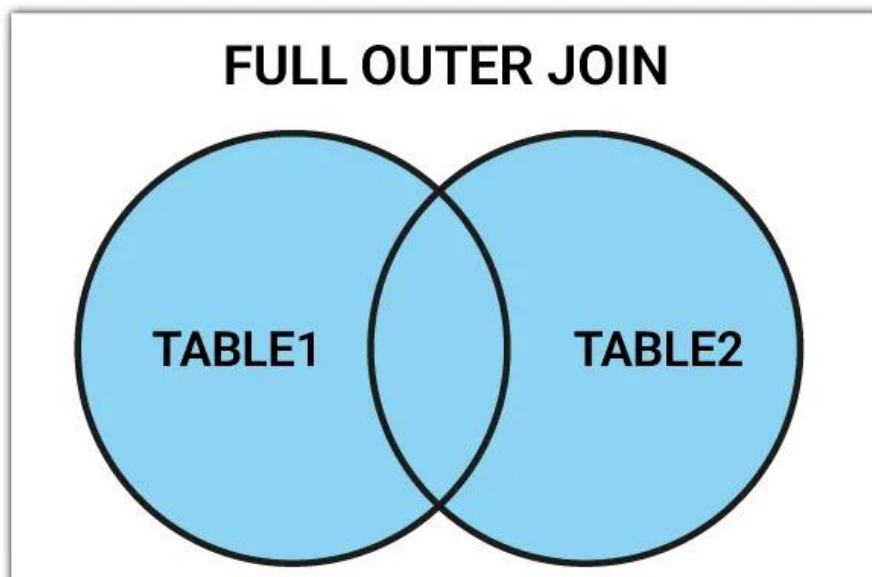
- **LEFT JOIN:** این دستور تمام سطرهای جدول سمت چپ (Left) را به همراه سطرهای مطابق در جدول سمت راست (Right) ارائه می‌دهد.



- **RIGHT JOIN:** برعکس Left Join، این دستور تمام سطرهای جدول سمت راست را همراه با سطرهای مطابقت یافته در جدول سمت چپ برمی‌گرداند.



- **FULL JOIN**: این دستور تمام سطرها را برمی‌گرداند، چه در جدول چپ و چه در جدول راست مطابقت وجود داشته باشد یا خیر.

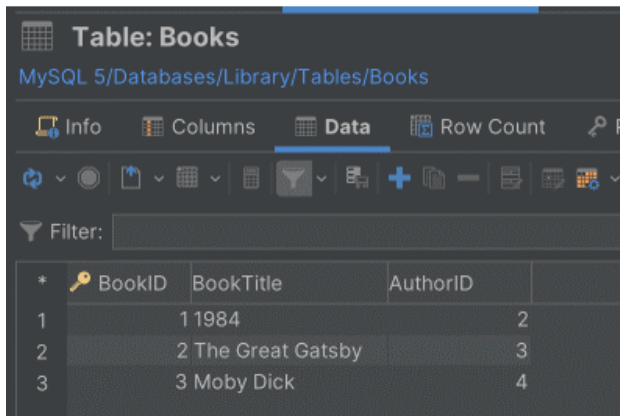


- **CROSS JOIN**: این دستور یک ترکیب جفت به جفت از تمام سطرهاى دو جدول را برمی‌گرداند.

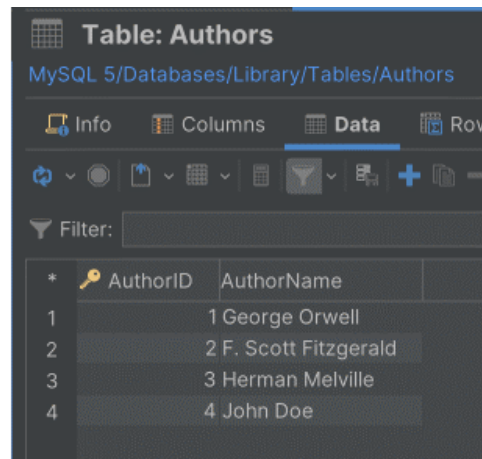
نحوه استفاده از دستور inner join در SQL Server

این بخش را با یک مثال کاربردی پیش می‌بریم تا به‌طور دقیق، نحوه استفاده از دستور inner join در SQL Server را متوجه شوید. برای استفاده از این دستور، بعد از اینکه مطمئن شدید پیش‌نیازهای لازم از جمله محیط دیتابیس SQL Server، و دانش اولیه را دارید، به‌شکل زیر اقدام کنید:

فرض کنید دو جدول با نام‌های Authors (نویسندگان) و Books (کتاب‌ها) داریم. جدول Authors شامل اطلاعات مربوط به نویسندگان و جدول Books شامل جزئیات کتاب‌ها، از جمله نویسنده آن‌ها است



| BookID | BookTitle | AuthorID |
|--------|------------------|----------|
| 1 | 1984 | 2 |
| 2 | The Great Gatsby | 3 |
| 3 | Moby Dick | 4 |



| AuthorID | AuthorName |
|----------|---------------------|
| 1 | George Orwell |
| 2 | F. Scott Fitzgerald |
| 3 | Herman Melville |
| 4 | John Doe |

می‌خواهیم فهرستی از کتاب‌ها به‌همراه نویسندگان را نمایش دهیم.

گام اول: راه اندازی دیتاست

در این بخش، دو جدول ساده به نام‌های Authors و Books ایجاد می‌کنیم:

جدول Authors

این جدول برای ذخیره اطلاعات مربوط به نویسندگان کتاب‌ها استفاده می‌شود. ساختار جدول به‌صورت زیر است:

| ستون | نوع داده | توضیحات |
|------------|--------------|---|
| AuthorID | INT | شناسه منحصر به فرد هر نویسنده (کلید اصلی) |
| AuthorName | VARCHAR(۲۵۵) | نام و نام خانوادگی نویسنده |

۱. دستور ایجاد جدول

```
CREATE TABLE Authors (
  AuthorID INT PRIMARY KEY,
  AuthorName VARCHAR(255));
```

۲. درج اطلاعات اولیه

برای اینکه جدول Authors خالی نباشد، چند نمونه رکورد به آن اضافه می‌کنیم:

```
INSERT INTO Authors (AuthorID, AuthorName)
VALUES
(۱, 'George Orwell'),
(۲, 'F. Scott Fitzgerald'),
(۳, 'Herman Melville'),
(۴, 'John Doe');
```

در این دستورات، ۱، ۲، ۳ و ۴ شناسه‌های منحصر به فرد هر نویسنده و George Orwell ، F. Scott Fitzgerald ، Herman Melville و John Doe نام و نام خانوادگی آن‌ها هستند.

جدول Books

این جدول، اطلاعاتی در مورد کتاب‌ها و نویسنده‌های آن‌ها را در خود جای می‌دهد. ساختار جدول به شکل زیر است:

| ستون | نوع داده | توضیحات |
|-----------|--------------|--|
| BookID | INT | شناسه منحصر به فرد برای هر کتاب (کلید اصلی جدول) |
| BookTitle | VARCHAR(255) | عنوان کتاب |
| AuthorID | INT | شناسه منحصر به فرد نویسنده کتاب |

ستون AuthorID در جدول کتاب‌ها، به ستون AuthorID در جدول نویسندگان اشاره می‌کند (کلید خارجی). این ارتباط به ما امکان می‌دهد تا با استفاده از AuthorID ، اطلاعات مربوط به آن را از جدول نویسندگان بازیابی کنیم.

۱. دستور ایجاد جدول

```
CREATE TABLE Books (
  BookID INT PRIMARY KEY,
  BookTitle VARCHAR(255),
  AuthorID INT,
  FOREIGN KEY (AuthorID) REFERENCES Authors (AuthorID)
);
```

۲. پر کردن جدول با اطلاعات نمونه

```
INSERT INTO Books (BookID, BookTitle, AuthorID)
VALUES
(۱, '۱۹۸۴', ۲),
(۲, 'The Great Gatsby', ۳),
(۳, 'Moby Dick', 4);
```

این اسکرپت، سه کتاب نمونه را به جدول Books اضافه می‌کند:

- کتاب ۱۹۸۴ با شناسه ۱ و نویسنده با شناسه ۲
- کتاب گتسبی بزرگ با شناسه ۲ و نویسنده با شناسه ۳
- کتاب موبی دیک با شناسه ۳ و نویسنده با شناسه ۴
- با داشتن این جدول و اطلاعات، حالا می‌توانیم کتاب‌ها را از طریق AuthorID مشترک به نویسندگان آن‌ها با دستور Inner Join که در قسمت بعدی خواهیم گفت، را وصل کنیم.

گام دوم: استفاده از دستور INNER JOIN در SQL Server

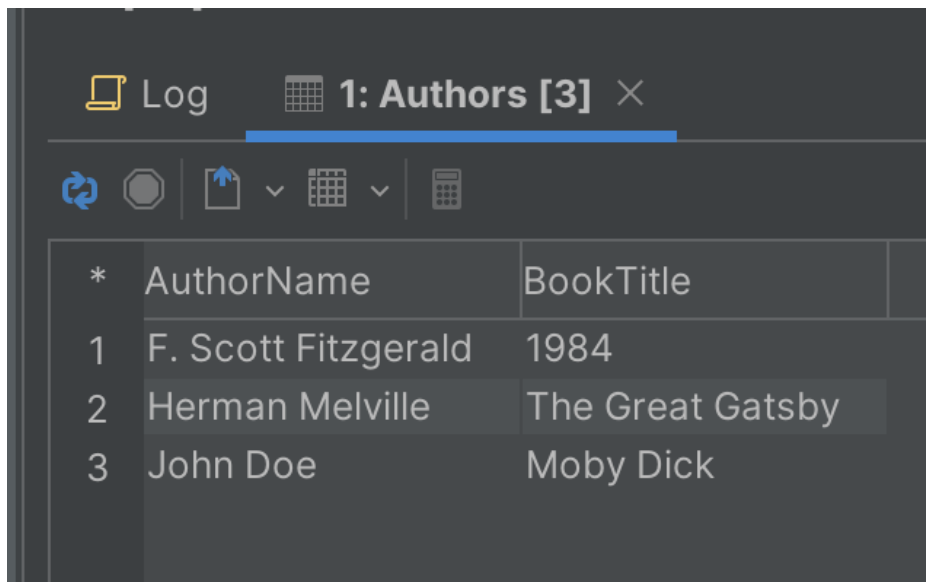
حالا اطلاعات دو جدول را با دستور INNER JOIN ادغام کرده و جدولی با ستون نام نویسنده و کتاب‌های آن‌ها را خروجی می‌گیریم.

```
SELECT
  Authors.AuthorName,
  Books.BookTitle
FROM
  Authors
INNER JOIN
  Books
ON
  Authors.AuthorID = Books.AuthorID;
```

- **SELECT**: این بخش مشخص می‌کند کدام ستون‌ها را می‌خواهیم از جداول بازیابی کنیم. در اینجا، ستون AuthorName از جدول Authors و ستون BookTitle از جدول Books را انتخاب کردیم.
- **FROM**: این بخش مشخص می‌کند اطلاعات از کدام جداول باید خوانده شود. در اینجا، از هر دو جدول Authors و Books استفاده می‌کنیم.
- **INNER JOIN**: این بخش، نوع اتصال بین دو جدول را مشخص می‌کند. در اینجا از INNER JOIN استفاده کردیم. این دستور ردیف‌هایی را برمی‌گرداند که در هر دو جدول وجود داشته باشند. در مثال ما، یعنی کتاب‌هایی که نویسنده آن‌ها در جدول نویسندگان ثبت شده است.

• **ON**: این بخش شرط اتصال را مشخص می‌کند. در اینجا، شرط برقراری ارتباط این است که مقدار ستون AuthorID در جدول Authors باید با مقدار ستون AuthorID در جدول Books برابر باشد.

با اجرای این دستور، جدولی حاوی دو ستون نمایش داده می‌شود. ستون اول شامل نام نویسندگان (AuthorName) و ستون دوم شامل عناوین کتاب‌های آن‌ها (BookTitle) است.



| * | AuthorName | BookTitle |
|---|---------------------|------------------|
| 1 | F. Scott Fitzgerald | 1984 |
| 2 | Herman Melville | The Great Gatsby |
| 3 | John Doe | Moby Dick |

مثال های استفاده از دستور inner join در SQL Server

برای درک بهتر در این بخش، مثال دیگری از دستور inner join در SQL Server را باهم بررسی می‌کنیم. فرض کنید دو جدول به نام‌های «محصولات» و «دسته‌بندی‌ها» دارید. جدول «محصولات» شامل اطلاعاتی مثل شناسه محصول، نام محصول و شناسه مربوط به دسته‌بندی آن محصول است. جدول «دسته‌بندی‌ها» هم شامل شناسه دسته‌بندی و توضیحاتی درباره آن دسته‌بندی می‌شود. حالا می‌خواهیم متوجه شویم نام هر محصول به کدام دسته‌بندی تعلق دارد. اینجاست که دستور inner join به کمکمان می‌آید.

در مثالی که گفتیم، جداول «محصولات» و «دسته‌بندی‌ها» به شکل زیر هستند:

جدول محصولات

| شناسه محصول (ProductID) | نام محصول (ProductName) | شناسه دسته‌بندی (CategoryID) |
|-------------------------|-------------------------|------------------------------|
| 1 | چای | 1 |
| 2 | کافه | 1 |
| 3 | شربت انیسون | 2 |

جدول دسته‌بندی‌ها

| شناسه دسته‌بندی (CategoryID) | نام دسته‌بندی (CategoryName) | توضیحات |
|------------------------------|------------------------------|---|
| 1 | نوشیدنی‌ها | نوشابه، قهوه، چای، آبجو و... |
| 2 | چاشنی‌ها | سس‌های شیرین و تند، چاشنی‌ها، کره‌های مالیدنی و ادویه‌جات |
| 3 | شیرینی‌جات | دسرها، آب‌نبات‌ها و نان‌های شیرین |

برای اینکه بفهمیم هر محصول به کدام دسته‌بندی تعلق دارد، می‌توانیم از دستور زیر استفاده کنیم:

```
SELECT ProductID, ProductName, CategoryName
FROM Products
INNER JOIN Categories ON Products.CategoryID = Categories.CategoryID;
```

با اجرای این دستور، خروجی به صورت زیر خواهد بود:

| شناسه محصول (ProductID) | نام محصول (ProductName) | نام دسته‌بندی (CategoryName) |
|-------------------------|-------------------------|------------------------------|
| 1 | چای | نوشیدنی‌ها |
| 2 | کافه | نوشیدنی‌ها |
| 3 | شربت انیسون | چاشنی‌ها |

همانطور که مشاهده می‌کنید، دستور inner join در SQL Server فقط محصولاتی را برمی‌گرداند که دارای یک دسته‌بندی مرتبط در جدول «دسته‌بندی‌ها» باشند. به عبارت دیگر، اگر محصولی وجود داشته باشد که شناسه دسته‌بندی آن در جدول «دسته‌بندی‌ها» موجود نباشد، آن محصول در نتیجه نمایش داده نخواهد شد.

کاربرد دستور inner join در SQL Server

دستور inner join در SQL Server یکی از ابزارهای قدرتمند در پایگاه‌های داده است که به برقراری ارتباط بین جداول مختلف کمک می‌کند. اما این ارتباط چه فایده‌ای دارد؟ اکنون به چند نمونه از کاربردهای آن در دنیای واقعی نگاهی خواهیم داشت.

۱- فروشگاه‌های آنلاین

تصور کنید در حال جستجو میان محصولات یک فروشگاه اینترنتی هستید. به لطف دستور inner join در SQL Server محصولات را می‌بینید که هم با جستجوی شما مطابقت دارند و هم در انبار موجود هستند. این کار باعث صرفه‌جویی در وقت شما می‌شود و دیگر نیازی نیست نگران دیدن محصولات بی‌فایده‌ای باشید که قابل خرید نیستند.

۲- بانکداری و امور مالی

بانک‌ها جداول جداگانه‌ای برای اطلاعات حسابداران و تراکنش‌های آن‌ها دارند. با استفاده از Inner Join آن‌ها می‌توانند این جداول را باهم ترکیب کنند تا به شما امکان دهند سابقه تراکنش‌های یک حساب خاص را به صورت یکجا مشاهده کنید. دیگر لازم نیست برای دیدن جزئیات تراکنش‌های خود، تک تک آن‌ها را جستجو کنید.

۳- مراکز درمانی

استفاده از دستور inner join در SQL Server در مراکز درمانی نیز کاربرد دارد. بیمارستان‌ها می‌توانند از Inner Join برای join کردن پرونده‌های بیماران به برنامه زمان‌بندی وقت ملاقات استفاده کنند؛ در نتیجه، پزشکان در زمان ویزیت به راحتی به سوابق پزشکی بیمار دسترسی خواهند داشت. این کار باعث می‌شود روند درمان با سرعت و دقت بیشتری انجام شود.

۴- شبکه های اجتماعی

شبکه‌های اجتماعی مانند فیسبوک یا توییتر از Inner Join برای اتصال پروفایل کاربران به پست‌ها یا توئیتهای آنها استفاده می‌کنند. این کار تضمین می‌کند محتوای درست برای کاربر درست نمایش داده شود؛ در نتیجه، دیگر نگران دیدن پست‌های تکراری یا نامرتب با علایق خود نخواهید بود.

۵- زنجیره تأمین و لجستیک

شرکت‌ها می‌توانند جزئیات حمل‌ونقل را با اطلاعات محصول ترکیب کنند تا به آنها در ردیابی محصولات در سراسر زنجیره تأمین کمک کند. این کار باعث می‌شود شرکت‌ها از موجودی انبار خود، مسیر حرکت کالاها و زمان رسیدن آنها به مقصد به خوبی مطلع باشند.

۳ نمونه واقعی استفاده از دستور Inner Join در SQL Server

برای درک بهتر اهمیت دستور inner join در SQL Server بیایید نگاهی به چند نمونه واقعی از شرکت‌های بزرگ بیاندازیم.

۱. آمازون

این غول تجارت الکترونیک برای مطابقت لیست محصولات با جزئیات فروشنده از Inner Join استفاده می‌کند. به این ترتیب، زمانی که محصولی در آمازون جستجو می‌کنید، علاوه بر اطلاعات محصول، جزئیاتی مانند فروشنده، امتیاز آنها و سایر موارد مهم نیز برایتان ظاهر می‌شود.

۲. جی‌پی مورگان

یکی از پیشگامان حوزه مالی جهان، از دستور inner join در SQL Server برای اتصال پروفایل مشتریان با سبد سرمایه‌گذاری آنها استفاده می‌کند. این ترکیب داده‌ها، تصویری جامع از وضعیت سرمایه‌گذاری مشتری را در اختیار بانک قرار می‌دهد.

۳. لینکدین

جالب است بدانید این شبکه اجتماعی حرفه‌ای، از عملیات Inner Join برای مطابقت پروفایل‌های کاربران با آگهی‌های شغلی استفاده می‌کند. این کار باعث می‌شود براساس مهارت‌ها، تجربیات و شبکه ارتباطی کاربر، پیشنهادات شغلی مرتبط به او نمایش داده شود.



خطاهای متداول دستور inner join در SQL Server

در این بخش، با خطاهای متداول دستور inner join در SQL Server آشنا خواهید شد. سپس راهکار رفع آن را نیز در بخش بعدی بررسی خواهیم کرد.

۱- حذف ناخواسته سطرها (گم شدن اطلاعات)

مشکل رایج با دستور inner join در SQL Server این بوده که ممکن است باعث حذف ناخواسته سطرها شود. این دستور تنها سطرهایی را برمی‌گرداند که در هر دو جدول، براساس شرط Join، مطابق باشند. اگر برای یک سطر در یک جدول، هیچ سطر مطابقی در جدول دیگر پیدا نشود، آن سطر به‌طور کلی از نتایج حذف می‌شود.

مثالی برای درک بهتر

فرض کنید دو جدول سفارشات و مشتریان دارید. جدول سفارشات شامل شناسه سفارش و شناسه مشتری است و جدول مشتریان شامل شناسه مشتری و نام مشتری است. می‌خواهید لیستی از نام مشتریان به‌همراه سفارشات آن‌ها تهیه کنید. با استفاده از دستور inner join در SQL Server براساس شناسه مشتری، تنها سفارشات مشتریانی نمایش داده می‌شود که در جدول مشتریان حضور دارند. در صورتی‌که مشتریانی در جدول سفارشات باشند که هنوز در جدول مشتریان ثبت نشدند، سفارشات آن‌ها در نتایج نهایی وجود نخواهند داشت.

۲- پیچیدگی بیش از حد کوئری

اگر تعداد زیادی از جداول را با دستور inner join در SQL Server به هم Join کنید، ممکن است کوئری شما بسیار پیچیده شود. این پیچیدگی باعث کندی در اجرای دستور و سخت‌تر خواندن کد می‌شود. سعی کنید با کمترین تعداد Join ممکن به نتیجه دلخواه برسید. گاهی اوقات، با کمی تغییر در ساختار جداول یا کوئری می‌توانید از تعدد Join ها کم کنید.

مثالی برای درک بهتر

فرض کنید برای نمایش اطلاعات یک محصول، به ۵ جدول مختلف نیاز دارید که با هم مرتبط هستند. با نوشتن یک Inner Join برای هر ارتباط، کوئری نهایی بسیار طولانی و خواندن آن سخت می‌شود. در چنین مواقعی بهتر است استراتژی کارآمدتری برای دسترسی به اطلاعات موردنظر در نظر بگیرید.

۳- محدودیت در جداول با ستون های حجیم

در دیتابیس، امکان Join کردن مستقیم جداول براساس ستون‌های حجیم که برای نگهداری متن‌های طولانی یا داده‌های باینری استفاده می‌شوند (مثل text, ntext و image) وجود ندارد. اما روش‌هایی غیرمستقیمی برای انجام این کار وجود دارند که در بخش بعدی بررسی خواهیم کرد.

راهکار Join کردن جداول با ستون حجیم

برای رفع این مشکل، می‌توانید از دو روش استفاده کنید:

• روش اول: استفاده از تابع SUBSTRING

برای اینکه دو جدول را براساس قسمت خاصی از ستون‌های حجیم آن‌ها بهم وصل کنیم، از تابع SUBSTRING استفاده می‌کنیم. این تابع بخشی از یک رشته را برمی‌گرداند. مثلاً دستور زیر، دو جدول t1 و t2 را براساس ۲۰ کاراکتر اول ستون textcolumn آن‌ها با یکدیگر Join می‌کند:

```
SELECT * FROM t1
JOIN t2
ON SUBSTRING(t1.textcolumn, 1, 20) = SUBSTRING(t2.textcolumn, 1, 20);
```

این روش در دستور inner join در SQL Server فقط براساس بخش ابتدایی ستون‌ها مقایسه انجام می‌دهد و برای کل محتوا مناسب نیست.

روش دوم: استفاده از تابع DATALENGTH

اگر می‌خواهیم دو جدول را براساس کل حجم (تعداد کاراکتر) ستون‌های حجیم آن‌ها با یکدیگر مقایسه کنیم، از تابع DATALENGTH استفاده می‌کنیم. این تابع، تعداد کل کاراکترهای یک رشته را برمی‌گرداند. مثلاً دستور زیر دو جدول p1 و p2 را براساس اینکه طول ستون pr_info آن‌ها باهم برابر باشند، Join می‌کند:

```
SELECT * FROM p1
JOIN p2
ON DATALENGTH(p1.pr_info) = DATALENGTH(p2.pr_info);
```

این روش تنها حجم کل ستون‌ها را مقایسه کرده و محتوای آن‌ها را بررسی نمی‌کند.

مثالی برای درک بهتر

فرض کنید دو جدول کتاب با عناوین books و book_descriptions داریم که به ترتیب شامل اطلاعات پایه کتاب‌ها و توضیحات مختصری از آن‌ها هستند. ساختار این جداول به صورت زیر است:

• جدول books

| ستون | نوع داده | توضیحات |
|---------|--------------|----------------------------|
| book_id | int | شناسه منحصر به فرد هر کتاب |
| title | varchar(255) | عنوان کتاب |
| author | varchar(255) | نویسنده کتاب |

• جدول book_descriptions

| ستون | نوع داده | توضیحات |
|----------------|----------|-------------------------------|
| description_id | int | شناسه منحصر به فرد هر توضیحات |
| book_id | int | شناسه کتاب مرتبط با توضیحات |
| description | text | توضیحات مختصر کتاب |

با توجه به اینکه ستون description از نوع text است، نمی‌توانیم به طور مستقیم دو جدول را براساس آن Join کنیم. از طرفی، می‌خواهیم لیستی از عناوین کتاب‌ها به همراه توضیحات مختصر آن‌ها را نمایش دهیم.

از آنجا که توضیحات مختصر در ابتدای ستون description قرار دارند، از روش اول (استفاده از تابع SUBSTRING) برای Join کردن دو جدول براساس ۲۰ کاراکتر اول ستون description استفاده می‌کنیم. دستور SQL Server برای انجام این کار به صورت زیر است:

```
SELECT b.title, bd.description
FROM books AS b
JOIN book_descriptions AS bd
ON SUBSTRING(bd.description, 1, 20) = SUBSTRING(b.description, 1, 20);
```

در این دستور، b نام مستعار برای جدول books و bd نام مستعار برای جدول book_descriptions است.

بهینه سازی دستور inner join در SQL Server

در این بخش، به نکاتی می‌پردازیم که به شما در نوشتن بهینه کوئری‌های دستور inner join در SQL Server و بهبود عملکرد آن‌ها کمک می‌کند.

محدود کردن نتایج

از شرط [Where](#) می‌توانید برای فیلترکردن داده‌های غیرضروری استفاده کنید. به عبارت دیگر، فقط سطر (Rows) و ستون‌هایی (Columns) را که واقعاً نیاز دارید، از پایگاه داده بیرون بکشید.

برای مثال، فرض کنید جدولی از اطلاعات فروش دارید که شامل ستون‌هایی مانند شناسه محصول، نام محصول، قیمت، تاریخ فروش و موارد دیگر است. اگر می‌خواهید لیستی از محصولات را ببینید که در ماه گذشته فروخته شدند، از شرط Where برای محدود کردن نتایج براساس تاریخ فروش استفاده کنید. با این کار، پایگاه داده مجبور نیست کل جدول را بررسی کند و فقط سطرهایی را برمی‌گرداند که با شرط Where مطابقت دارند.

استفاده از Indexes

Indexes (شاخص‌ها) ساختارهایی هستند که به پایگاه داده کمک می‌کنند تا داده‌ها را سریع‌تر پیدا کند. مطمئن شوید ستون‌هایی که در حال Join کردن آن‌ها هستید، ایندکس شدند، به‌خصوص اگر این ستون‌ها به جداول بزرگ تعلق دارند.

ایندکس‌ها شبیه فهرست کتاب عمل می‌کنند. با داشتن فهرست، برای پیدا کردن مطلب موردنظر در کتاب نیازی نیست کل کتاب را ورق بزنید؛ بلکه به راحتی می‌توانید به بخش مرتبط مراجعه کنید. به همین ترتیب، ایندکس‌ها به پایگاه داده کمک می‌کنند تا بدون نیاز به بررسی تک‌تک سطرهای جدول، داده‌های موردنظر را به‌طور مستقیم پیدا کند.

ترتیب Join کردن جداول

وقتی چندین جدول را باهم Join می‌کنید، دیتابیس سعی می‌کند با مرتب‌کردن مجدد Join ها، بهینه‌ترین روش را پیدا کند. به‌طور کلی، Join کردن جداول کوچک به جداول بزرگتر، معمولاً از روش برعکس، یعنی Join کردن جداول بزرگ به کوچک، کارایی بیشتری دارد.

فرض کنید سه جدول فروش (sales)، محصول (products) و شعبه (branches) دارید. اگر بخواهید لیستی از کل فروش هر محصول در هر شعبه را به دست بیاورید، بهتر است ابتدا دو جدول فروش و محصول را براساس شناسه محصول (product_id) جوین کنید. سپس خروجی آن را با جدول شعبه براساس شناسه شعبه (branch_id) به هم جوین دهید.

بررسی اطلاعات قبل از Join

قبل از اینکه دو جدول را با دستور inner join در SQL Server بهم وصل کنید، بهتر است استعلام جداگانه اجرا کنید تا مطمئن شوید این وسط داده‌های تکراری یا نامرتب وجود ندارد. این کار باعث می‌شود موقع وصل کردن جداول، بفهمید ممکن است چه اطلاعاتی جا بیافتد. فرض کنید می‌خواهید مشتریان و سفارشات آن‌ها را باهم Join کنید. بهتر است قبل انجام دستور استعلام جداگانه اجرا کنید تا ببینید آیا سفارش‌هایی وجود دارد که برای مشتری خاصی ثبت نشده باشد.

دقت در ارتباط بین جداول

در استفاده از دستور inner join در SQL Server، مهم‌ترین نکته، درک ارتباط بین جداولی است که می‌خواهید Join کنید. باید مطمئن شوید ستون‌هایی که برای Join کردن انتخاب کردید، در هر دو جدول هم‌معنی هستند و داده‌های مشابه دارند. برای مثال، شما نمی‌توانید جدول «کتاب‌ها» را براساس ستون «نویسنده» با جدول «مشتریان» براساس ستون «نویسنده» Join کنید؛ چون مفهوم «نویسنده» در این دو جدول باهم فرق دارد.

قرار دادن نام جدول قبل از ستون

برای اینکه خواندن و فهم کدهای SQL Server برای خودتان و دیگران راحت‌تر شود، بهتر است همیشه نام جدول را قبل از نام ستون ذکر کنید.

برای مثال، هر دو جدول Products (محصولات) و Categories (دسته‌بندی‌ها) ممکن است دارای یک ستون با نام CategoryID باشند. در صورتی که در دستور **SELECT** خود فقط از نام ستون‌ها استفاده کنید و نام جدول را ذکر نکنید، با فرض اینکه ستون موردنظر در هر دو جدول وجود ندارد، کد شما اجرا می‌شود. درحالی‌که اگر بخواهید از ستون CategoryID استفاده کنید، حتماً باید نام جدول را هم ذکر نمایید. در غیراینصورت، با خطا مواجه خواهید شد.

برای درک بهتر، فرض کنید شناسه محصول (ProductID)، نام محصول (ProductName) و نام دسته‌بندی (CategoryName) را از جدول محصولات به‌همراه نام دسته‌بندی مربوطه دریافت می‌کنید.

• روش اشتباه (احتمال بروز خطا)

```
SELECT ProductID, ProductName, CategoryID
FROM Products
INNER JOIN Categories ON CategoryID = CategoryID;
```

• روش صحیح

```
SELECT Products.ProductID, Products.ProductName, Categories.CategoryName
FROM Products
INNER JOIN Categories ON Products.CategoryID = Categories.CategoryID;
```

همانطور که می‌بینید، در روش صحیح، با ذکر نام جدول قبل از نام ستون، مشخص کردیم CategoryID موردنظر متعلق به کدام جدول است. این کار باعث خوانایی بهتر کد و جلوگیری از بروز خطا می‌شود.

جمع بندی: دستور inner join در SQL Server

در دنیای پهناور SQL Server، دستور inner join حکم ستون اصلی را دارد. این دستور به شما امکان می‌دهد تا جداول مجزا را به هم وصل و ارتباطات پنهان میان آن‌ها را کشف کنید. با Inner Join در SQL Server، می‌توانید به راحتی اطلاعات را از چندین منبع استخراج کرده و آن‌ها را در قالبی واحد و منسجم به نمایش بگذارید. با استفاده از این دستور، می‌توانید:

- ارتباطات بین جداول را مشخص کنید.
- گزارش‌های جامع‌تر و پرمحتواتر تهیه کنید.
- تحلیل‌های عمیق‌تر و دقیق‌تر انجام دهید.
- اطلاعات مرتبط را در یک نمای واحد جمع‌آوری کنید.

اگر در این مسیر، نیاز به راهنمایی بیشتر دارید، سؤالاتان را می‌توانید در انتهای همین مقاله بنویسید تا راهنمایی‌تان کنیم.