

در زبان برنامه‌نویسی انتخاب شده انقیاد نوع و مقدار چگونه و در چه زمانی انجام می‌شود؟ آیا تعاریف متغیرها ضمنی است یا صریح و یا هر دو نوع تعریف وجود دارد؟ با ذکر مثال توضیح داده شود.

انقیاد داده در SQL فرآیند مرتبط کردن یک نشانگر متغیر در یک دستور SQL با یک متغیر در برنامه است. انقیاد داده‌ها بسته به نوع دستور SQL و درایور مورد استفاده می‌تواند به صورت ایستا یا پویا انجام شود. اتصال داده‌های ایستا در زمان کامپایل انجام می‌شود، درحالی‌که اتصال داده‌های پویا در زمان اجرا انجام می‌شود. اتصال داده‌ها را می‌توان به صورت صریح یا ضمنی انجام داد، بسته به نحو دستور SQL و درایور مورد استفاده. در زبان SQL، انقیاد نوع و مقدار به صورت ضمنی وجود دارد، به این معنی که شما در تعریف ستون‌ها و جداول نوع داده‌ها را مشخص نمی‌کنید. بلکه، بر اساس مقادیری که در ستون‌ها ذخیره می‌شوند، نوع داده به صورت ضمنی تشخیص داده می‌شود.

به طور معمول، در زمان ایجاد یک جدول، شما فقط نام ستون‌ها و نوع داده‌هایی که در آن‌ها ذخیره می‌شوند (مانند عددی، رشته، تاریخ و غیره) را مشخص می‌کنید. برای مثال، در ایجاد جدول زیر:

```
CREATE TABLE Employees (  
    ID INT,  
    Name VARCHAR(50),  
    Age INT,  
    Salary DECIMAL(10, 2)  
);
```

در این مثال، برای ستون‌ها نوع داده‌های مشخص شده است. ستون "ID" به عنوان یک عدد صحیح (INT)، ستون "Name" به عنوان یک رشته با طول حداکثر ۵۰ کاراکتر (VARCHAR(50))، ستون "Age" به عنوان یک عدد صحیح (INT)، و ستون "Salary" به عنوان یک عدد اعشاری با ۲ رقم اعشار (DECIMAL(10,2)) تعریف شده‌اند.

به طور خلاصه، در زبان SQL، تعریف نوع داده به صورت صریح انجام نمی‌شود و نوع داده بر اساس مقادیر ورودی تشخیص داده می‌شود، که به صورت ضمنی است.

در زبان SQL، تعریف نوع داده به صورت صریح در مواردی مانند تعریف پارامترهای تابع‌ها، متغیرهای موقت (Temporary Variables) و دستورات دیگر انجام می‌شود. در این موارد، شما باید نوع داده‌ها را به صورت صریح مشخص کنید.

برای مثال، در تعریف یک پروسیجر (Stored Procedure) در SQL Server، می‌توانید نوع داده پارامترها را به صورت صریح مشخص کنید. در مثال زیر، یک پروسیجر به نام "GetEmployeeByID" تعریف شده است که با دریافت یک شناسه کارمند، اطلاعات کارمند متناظر را برمی‌گرداند:

```
CREATE PROCEDURE GetEmployeeByID
    @EmployeeID INT
AS
BEGIN
    SELECT * FROM Employees WHERE ID = @EmployeeID;
END;
```

در این مثال، پارامتر "EmployeeID" به عنوان یک عدد صحیح (INT) تعریف شده است.

به طور خلاصه، در زبان SQL، تعریف نوع داده به صورت صریح در برخی موارد از جمله تعریف پارامترها و متغیرهای موقت انجام می شود، در حالی که در تعریف ستون ها و جداول نوع داده به صورت ضمنی بر اساس مقادیر ورودی تشخیص داده می شود.

آیا در زبان SQL متغیرهای ایستا، پویا در پشته، پویا در هیپ به طور صریح، پویا در هیپ به طور ضمنی وجود دارند؟ با مثال توضیح داده شود. برای توصیف هر یک از موارد یک قطعه کد نوشته شود. همچنین توضیح داده شود که هر یک این متغیرها در این زبان چگونه پیاده سازی شده اند. آیا می توانید سرعت تخصیص این متغیرها را در این زبان مقایسه کنید؟

در زبان SQL، متغیرهای محلی وجود دارند که می توانند یک مقدار داده ای از یک نوع خاص را نگه دارند. متغیرهای محلی با استفاده از دستور DECLARE ایجاد می شوند و با استفاده از دستور SET یا SELECT مقداردهی می شوند.

متغیرهای محلی در SQL می توانند از نوع های داده ای مختلفی مانند int، varchar، date، xml و ... باشند. متغیرهای محلی در SQL به صورت ایستا، پویا در پشته، پویا در هیپ به طور صریح یا پویا در هیپ به طور ضمنی تعریف می شوند. در ادامه به توضیح این موارد با مثال می پردازیم.

- متغیرهای ایستا: این متغیرها در زمان کامپایل تعریف و مقداردهی می شوند و طول عمر آنها تا پایان بچ یا رویه ای که در آن تعریف شده اند است. این متغیرها در حافظه پشته قرار می گیرند و سرعت تخصیص و دسترسی به آنها بالاست. برای مثال، در کد زیر یک متغیر ایستا به نام @MyCounter با نوع int تعریف و مقداردهی شده است:

```
-- Declare and initialize a static variable
DECLARE @MyCounter INT = 0;
-- Print the value of the variable
PRINT @MyCounter;
```

- متغیرهای پویا در پشته: این متغیرها در زمان اجرا تعریف و مقداردهی می شوند و طول عمر آنها تا پایان بچ یا رویه ای که در آن تعریف شده اند است. این متغیرها نیز در حافظه پشته قرار می گیرند و سرعت تخصیص و دسترسی به آنها بالاست. برای مثال، در کد زیر یک متغیر پویا در پشته به نام @MyName با نوع varchar تعریف و مقداردهی شده است:

```
-- Declare a dynamic variable in stack
DECLARE @MyName VARCHAR(50);
-- Assign a value to the variable at run time
SET @MyName = 'Bing';
-- Print the value of the variable
PRINT @MyName;
```

• متغیرهای پویا در هیپ به‌طور صریح: این متغیرها در زمان اجرا تعریف و مقداردهی می‌شوند و طول عمر آنها تا پایان بچ یا رویه‌ای که در آن تعریف شده‌اند است. این متغیرها در حافظه هیپ قرار می‌گیرند و سرعت تخصیص و دسترسی به آنها کمتر از متغیرهای پشته است. برای مثال، در کد زیر یک متغیر پویا در هیپ به‌طور صریح به نام @MyDate با نوع date تعریف و مقداردهی شده است:

```
-- Declare an explicit dynamic variable in heap
DECLARE @MyDate DATE;
-- Assign a value to the variable at run time
SELECT @MyDate = GETDATE();
-- Print the value of the variable
PRINT @MyDate;
```

• متغیرهای پویا در هیپ به‌طور ضمنی: این متغیرها در زمان اجرا تعریف و مقداردهی می‌شوند و طول عمر آنها تا پایان دستوری که در آن تعریف شده‌اند است. این متغیرها نیز در حافظه هیپ قرار می‌گیرند و سرعت تخصیص و دسترسی به آنها کمتر از متغیرهای پشته است. برای مثال، در کد زیر یک متغیر پویا در هیپ به‌طور ضمنی به نام @MyNumber با نوع int تعریف و مقداردهی شده است:

```
-- Declare an implicit dynamic variable in heap
SELECT @MyNumber = 10;
-- Print the value of the variable
PRINT @MyNumber;
```

به‌طور کلی، سرعت تخصیص متغیرها در SQL بستگی به نوع متغیر و نحوه استفاده از آنها دارد. متغیرهای ایستا به دلیل تخصیص یک‌باره سریع‌تر از متغیرهای پویا تخصیص داده می‌شوند.

منابع:

- <https://Poe.com>
- <https://Bing.com>
- <https://stackoverflow.com>
- Data Binding (ODBC): [1]
- Using Parameters (ODBC): [2]