آموزشکده فنی پسران آمل- علامه حسن زاده آملی

آزمایشگاه پایگاه داده ها

کاردانی پیوسته کامپیوتر

حسین اخوان اسکی

کارشناس ارشد نرم افزار

فهرست مطالب

عنوان **صفحه**

عنوان **صفحه**

فصل اول

معرفی پایگاه داده و نصب و راه‌اندازی SQL Server

پایگاه داده چیست؟

پایگاه داده (Database) مجموعه‌ای سازمان‌یافته از داده‌هاست که به‌صورت الکترونیکی ذخیره می‌شود و امکان ذخیره، جستجو، ویرایش و بازیابی سریع اطلاعات را فراهم می‌کند.

در دنیای واقعی تقریباً تمام سیستم‌ها از پایگاه داده استفاده می‌کنند، مانند:

سیستم ثبت نمرات دانشگاه

فروشگاه‌های اینترنتی

سامانه‌های بانکی

شبکه‌های اجتماعی

DBMS چیست؟

DBMS یا Database Management System نرم‌افزاری است که مدیریت پایگاه داده را بر عهده دارد.

وظایف DBMS:

ایجاد پایگاه داده

ذخیره و بازیابی اطلاعات

تأمین امنیت داده‌ها

مدیریت کاربران

نمونه DBMS ها:

MySQL

Oracle

PostgreSQL

Microsoft SQL Server

در این درس، تمرکز ما روی SQL Server است.

معرفی Microsoft SQL Server

SQL Server یک سیستم مدیریت پایگاه داده رابطه‌ای (RDBMS) است که توسط شرکت مایکروسافت توسعه داده شده است.

مزایای SQL Server:

محیط گرافیکی قدرتمند (SSMS)

امنیت بالا

پشتیبانی از T-SQL

مناسب برای پروژه‌های کوچک تا سازمانی

نسخه‌های SQL Server

نسخه‌های رایج:

Enterprise: مناسب سازمان‌های بزرگ

Standard: کاربرد تجاری

Developer: رایگان برای آموزش و توسعه

Express: رایگان با محدودیت حجم

دانلود و نصب SQL Server

پیش‌نیازها:

ویندوز ۱۰ یا ۱۱ (۶۴ بیت)

حداقل ۴ گیگابایت RAM

مراحل کلی نصب:

اجرای فایل نصب

انتخاب New Standalone Installation

انتخاب Feature Engine

تنظیم Instance (پیش‌فرض)

تنظیم Authentication

احراز هویت (Authentication)

دو روش اصلی:

Windows Authentication

SQL Server Authentication (ساخت کاربر sa)

نصب SQL Server Management Studio (SSMS)

SSMS محیط گرافیکی برای مدیریت SQL Server است.

امکانات SSMS:

اجرای Query

مدیریت دیتابیس‌ها

طراحی جداول

بکاپ و ریستور

آشنایی با محیط SSMS

بخش‌های اصلی:

Object Explorer

Query Window

Database Engine

Execute

Messages / Results

مثال عملی: اتصال به SQL Server

اجرای SSMS

انتخاب Server Type: Database Engine

Authentication: Windows Authentication

Connect

تمرین‌های پایان فصل اول

تمرین ۱:

تعریف پایگاه داده و DBMS را با مثال توضیح دهید.

تمرین ۲:

سه نمونه از نرم‌افزارهایی که از پایگاه داده استفاده می‌کنند نام ببرید.

تمرین ۳ (عملی):

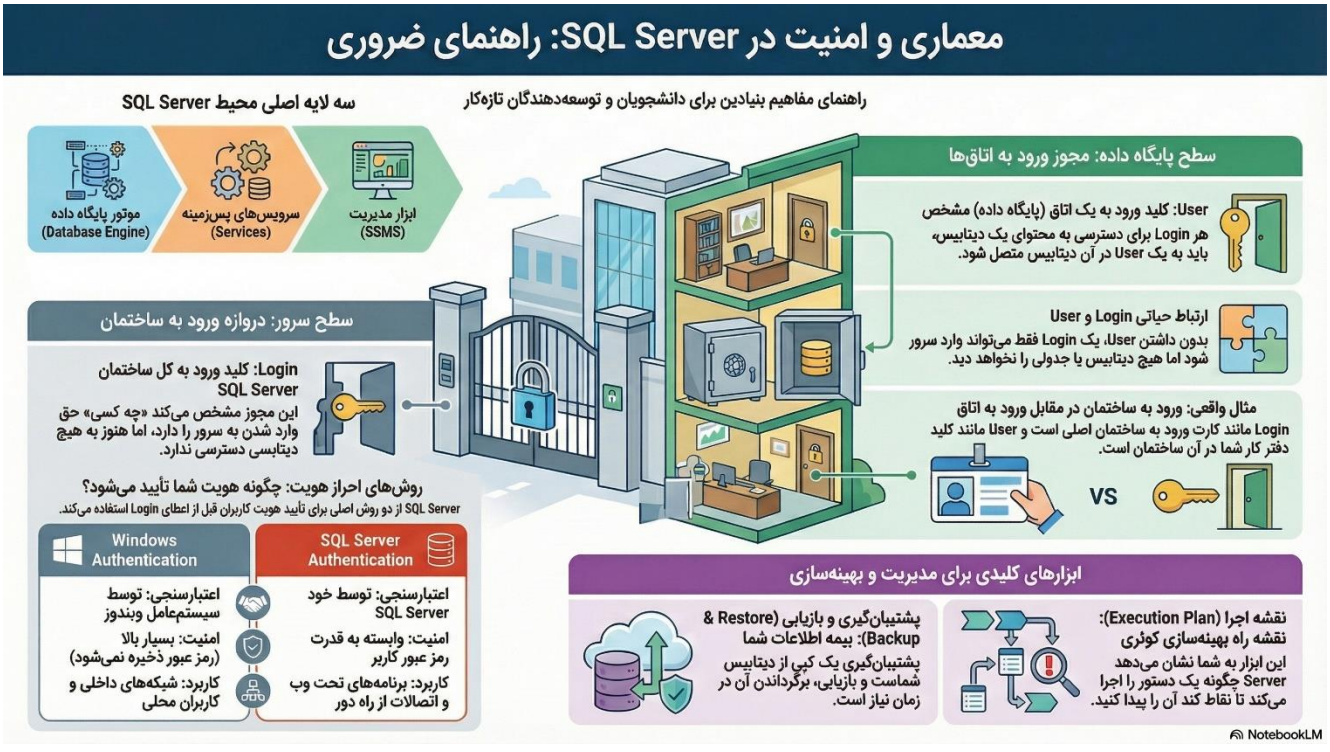
SQL Server و SSMS را روی سیستم خود نصب کرده و اسکرین‌شات از Object Explorer تهیه کنید.

تمرین ۴:

تفاوت Windows Authentication و SQL Authentication را بنویسید.

فصل دوم

آشنایی کامل و عمیق با محیط، ابزارها، احراز هویت و سرویس‌دهنده‌ها در Microsoft SQL Server



درک کلی محیط کاری SQL Server

وقتی SQL Server نصب می‌شود، در واقع **سه لایه اصلی** داریم:

۱. **موتور پایگاه داده (Database Engine)**

بخشی که داده‌ها را ذخیره می‌کند، Queryاجرا می‌کند و امنیت را کنترل می‌کند.

۲. **سرویس‌های SQL Server**

این سرویس‌ها در پس‌زمینه ویندوز اجرا می‌شوند و اگر متوقف شوند، SQL Serverعملاً از کار می‌افتد.

۳. **محیط مدیریت (SSMS)**

ابزاری که ما به‌عنوان کاربر از طریق آن با Database Engine ارتباط برقرار می‌کنیم.

درک این سه لایه باعث می‌شود دانشجو بداند مشکل از کجاست:

- نرم‌افزار؟
- سرویس؟
- یا دسترسی کاربر؟

راه‌اندازی و نقش سرویس‌دهنده‌ها (SQL Server Services)

SQL Serverمانند یک برنامه معمولی اجرا نمی‌شود؛
بلکه به‌صورت **Service**در ویندوز فعال است.

SQL Server Database Engine Service

این سرویس:

- فایل‌های دیتابیس را مدیریت می‌کند
- دستورات SQL را پردازش می‌کند
- کاربران را احراز هویت می‌کند

اگر این سرویس **Stop**شود:

- اتصال به SQL Server غیرممکن می‌شود
- تمام برنامه‌های متصل قطع می‌شوند

SQL Server Browser Service

این سرویس کمک می‌کند:

- سیستم‌های دیگرInstance ها را پیدا کنند
- اتصال شبکه‌ای ساده‌تر شود

در سیستم‌های تک‌کاربره ممکن است حیاتی نباشد، اما برای آموزش روشن بودن آن توصیه می‌شود.

احراز هویت — (Authentication) نگاه عمیق امنیتی

احراز هویت یعنی:

SQL Serverبررسی کند آیا این شخص اجازه ورود دارد یا نه

Windows Authentication (تشریح کامل)

در این روش:

- SQL Serverاز سیستم‌عامل سؤال می‌کند
- اگر ویندوز کاربر را تأیید کند، ورود انجام می‌شود

مزیت بزرگ:

- رمز عبور داخل SQL Server ذخیره نمی‌شود
- امنیت بسیار بالا

محدودیت:

- مناسب کاربرانی است که روی همان سیستم یا شبکه داخلی هستند

SQL Server Authentication(تشریح کامل)

در این روش:

- نام کاربری و رمز در خود SQL Server ذخیره می‌شود
- مستقل از ویندوز عمل می‌کند

کاربرد:

- اتصال از راه دور
- برنامه‌های تحت وب
- نرم‌افزارهای کلاینت–سرور

ریسک:

- اگر رمز ضعیف باشد، امکان نفوذ وجود دارد

Mixed Modeچرا مهم است؟

Mixed Modeیعنی:

- SQL Serverهم ویندوز را قبول دارد و هم کاربران داخلی خودش را.

در آموزش:

- دانشجو هر دو روش واقعی بازار کار را یاد می‌گیرد
- در پروژه‌های بعدی دچار محدودیت نمی‌شود

Login – مجوز ورود به سرور

Login اولین مرحله امنیت است.

Loginیعنی:

»چه کسی اجازه دارد وارد SQL Server شود«

ویژگی‌های:Login

- در سطح **Server**تعریف می‌شود
- هنوز به هیچ دیتابیسی وصل نیست

انواع:Login

- Loginویندوزی
- LoginداخلیSQL

اشتباه رایج:

دانشجو فکر می‌کند با ساخت Login می‌تواند جدول‌ها را ببیند؛

در حالی که **هنوز هیچ دسترسی‌ای به دیتابیس ندارد.**

User — مجوز کار داخل پایگاه داده

Userمرحله دوم امنیت است.

Userیعنی:

»این Login داخل کدام دیتابیس و با چه سطحی کار کند«

هر Login می‌تواند:

- در یک دیتابیس User داشته باشد
- در دیتابیس دیگر نداشته باشد

مثال کاملاً واقعی

- Login → ورود به ساختمان
- User → ورود به اتاق مشخص

بدون User:

- Loginوارد سرور می‌شود
- اما دیتابیس را نمی‌بیند

Object Explorer — نقشه کامل SQL Server

Object Explorerمثل:

»تابلوی کنترل کل سیستم پایگاه داده«

در این بخش می‌توانید:

- دیتابیس‌ها را ببینید
- کاربران را مدیریت کنید

- جدول‌ها را بررسی کنید
- وضعیت سرور را کنترل کنید

Object Explorer Detail – مدیریت حرفه‌ای

این بخش زمانی اهمیت پیدا می‌کند که:

- دیتابیس‌ها زیاد شوند
- جدول‌ها متعدد باشند

مزیت:

- نمایش لیستی و مرتب
- فیلتر و مرتب‌سازی ساده‌تر

در پروژه‌های واقعی بسیار کاربردی است.

Document Window – محل کار واقعی دانشجو

Document Windowجایی است که:

- Queryنوشته می‌شود
- Scriptذخیره می‌شود
- خروجی دیده می‌شود

Server & Databases – تفکیک بسیار مهم

سطح Server:

- Login
- تنظیمات امنیتی
- سرویس‌ها

سطح Database:

- Table

- View
- User
- Stored Procedure

Execute Script – اجرای دستورات

Executeیعنی:

SQL Serverدستور را دریافت و اجرا می‌کند

روش‌ها:

- دکمهExecute
- کلیدF5

نکته مهم:
اگر دیتابیس اشتباه انتخاب شود،
دستور درست اجرا می‌شود ولی نتیجه اشتباه خواهد بود.

Query Execution Plan – فهم عملکردSQL Server

Execution Planبه شما نشان می‌دهد:

- SQL Serverاز کجا شروع کرده
- چه مراحلی طی شده
- کدام بخش زمان‌بر بوده

این ابزار پایه:

- بهینه‌سازی
- افزایش سرعت
- کاهش مصرف منابع

Backup Database – امنیت اطلاعات

Backupیعنی:

بیمه کردن اطلاعات

حتی بهترین برنامه‌نویس هم بدون Backup شکست می‌خورد.

Full Backup:

- کل دیتابیس
- ساده
- آموزشی

Restore Database – بازگشت به زندگی

Restoreیعنی:

برگرداندن دیتابیس از فایلBackup

کاربرد:

- خرابی سیستم
- انتقال پروژه
- تمرین عملی

بدونRestore ، Backupارزشی ندارد.

SQL Profiler – دیدن پشت صحنه

SQL Profilerبه شما نشان می‌دهد:

- چهQuery هایی اجرا می‌شوند
- کدام کاربر
- در چه زمانی

ابزاری حرفه‌ای برای:

- عیب‌یابی
- تحلیل عملکرد
- آموزش پیشرفته

Query Analyzer – تحلیل و تست

Query Analyzerکمک می‌کند:

- خطاها را بفهمید
- عملکرد دستور را بررسی کنید
- قبل از اجرای نهایی تست بگیرید

در نسخه‌های جدید، این قابلیت داخل SSMS ادغام شده است.

تمرین‌های فصل دوم

۱. تفاوت Login و User را با مثال واقعی توضیح دهید.

۲. اگر Login وجود داشته باشد ولی User نباشد چه می‌شود؟

۳. چرا Execution Plan برای سرعت مهم است؟

٤. Full Backupچه مشکلی را حل می‌کند؟

٥. SQL Profilerچه زمانی استفاده می‌شود؟

فصل سوم

ساخت پایگاه داده و ساخت کاربر پایگاه داده با استفاده از SSMS و Script در Microsoft SQL Server

تا اینجا:

- SQL Serverنصب شده
- محیط SSMS شناخته شده
- مفاهیم Login و User درک شده

اما **هیچ پایگاه داده‌ای بدون ساخت Database و User معنا ندارد.**
در دنیای واقعی:

- برنامه‌نویس باید دیتابیس بسازد
- مدیر سیستم باید کاربر تعریف کند
- امنیت باید از ابتدا رعایت شود

پایگاه داده (Database) دقیقاً چیست؟

Databaseیک **کانتینر منطقی** است که همه اجزای زیر را در خود نگه می‌دارد:

- جدول‌ها(Table)
- کاربران(User)
- Viewها
- Stored Procedureها
- Functionها
- Indexها

نکته بسیار مهم:
در SQL Server ، **Loginخارج از Database** و **Userداخل Database** تعریف می‌شود.

روش‌های ساخت پایگاه داده در SQL Server

در SQL Server دو روش اصلی برای ساخت Database وجود دارد:

۱. **روش گرافیکی(SSMS)**

۲. **روش نوشتن Script باT-SQL**

ساخت پایگاه داده با استفاده از محیط گرافیکیSSMS

مرحله ۱: ورود بهObject Explorer

پس از اتصال به سرور، در Object Explorer مسیر زیر را دنبال می‌کنیم:

Server→ Databases

روی **Databases**راست‌کلیک می‌کنیم و گزینه **New Database**را انتخاب می‌کنیم.

مرحله ۲: تنظیم نام پایگاه داده

در پنجره باز شده:

- در قسمت **Database name**یک نام مناسب وارد می‌کنیم

مثال:

- StudentDB

📌**نکته آموزشی:**

- نام دیتابیس نباید فاصله داشته باشد
- بهتر است انگلیسی باشد
- بیانگر کاربرد آن باشد

مرحله ۳: بررسی تنظیمات (اختیاری ولی مهم)

در سمت چپ پنجره New Database گزینه‌هایی می‌بینید مثل:

- General
- Files

- Options

در سطح آموزشی:

- معمولاً تنظیمات پیش‌فرض کافی است
- در پروژه‌های بزرگ این بخش اهمیت زیادی دارد

با زدن دکمه **OK**، پایگاه داده ساخته می‌شود.

بررسی پایگاه داده ساخته‌شده

بعد از ساخت:

- Databaseجدید در Object Explorer ظاهر می‌شود
- اگر دیده نشد، روی Databases راست‌کلیک → Refresh

اکنون SQL Server آماده ذخیره اطلاعات است.

ساخت پایگاه داده با استفاده از Script (T-SQL)

روش دوم، روش حرفه‌ای‌تر و قابل کنترل‌تر است.

مفهومT-SQL

T-SQLزبان دستوری SQL Server است که به ما اجازه می‌دهد:

- پایگاه داده بسازیم
- کاربر تعریف کنیم
- جدول ایجاد کنیم

مثال: ساخت Database باScript

در Query Window دستور زیر را می‌نویسیم:

```
CREATE DATABASE StudentDB;
GO
```

سپس دکمه **Execute (F5)**را می‌زنیم.

اگر پیغام خطا نداشتیم، دیتابیس ساخته شده است.

📌مزیت:Script

- قابل ذخیره و استفاده مجدد
- قابل انتقال به سیستم دیگر
- مناسب پروژه‌های واقعی

تفاوت روش گرافیکی وScript

روش گرافیکی:

- ساده
- مناسب مبتدی‌ها
- کندتر در پروژه‌های بزرگ

روش:Script

- سریع
- حرفه‌ای
- قابل کنترل و مستندسازی

مفهوم امنیت در سطحDatabase

امنیت در SQL Server دو مرحله دارد:

۱. ورود به سرور(Login)

۲. کار در دیتابیس(User)

در این فصل تمرکز ما روی **User**است.

ساخت User برای پایگاه داده (روش گرافیکی)

مرحله ۱: رفتن به مسیر Users

Databases
└── StudentDB
  └── Security
    └── Users

روی **Users** راست‌کلیک → **New User**

مرحله ۲: تنظیمات User

در پنجره باز شده:

- User name: مثلاً student_user
- Login name: یکی از Loginهای موجود
- Default Schema: معمولاً dbo

📌 نکته مهم:
User بدون Login معنا ندارد.

مرحله ۳: تعیین سطح دسترسی (اختیاری در این مرحله)

در بخش Membership می‌توان:

- دسترسی خواندن
- نوشتن
- مدیریت

را مشخص کرد.

ساخت User با استفاده از Script (T-SQL)

ابتدا باید Login وجود داشته باشد.

فرض می‌کنیم Login از قبل ساخته شده است.

ساخت User با Script:

```
USE StudentDB;
GO
CREATE USER student_user FOR LOGIN student_login;
GO
```

📌این دستور:

- Userرا داخل دیتابیس می‌سازد
- آن را به Login متصل می‌کند

بررسی User ساخته‌شده

بعد از اجرای دستور:

- به مسیر Users بروید
- Userجدید را مشاهده کنید

اگر User دیده نمی‌شود:

- Refreshرا فراموش نکنید

Default Schema چیست؟

Schemaمثل:

یک پوشه برای اشیای پایگاه داده

اگر Default Schema درست تنظیم نشود:

- جدول‌ها در جای نامناسب ساخته می‌شوند

- مشکلات دسترسی به‌وجود می‌آید

در آموزش:

Default Schema = dbo

خطاهای رایج دانشجویان (بسیار مهم)

۱. ساخت Login ولی نساختن User

۲. اجرای Script در دیتابیس اشتباه

۳. فراموش کردن GO

٤. نداشتن دسترسی کافی

این خطاها کاملاً طبیعی هستند و با تمرین برطرف می‌شوند.

تمرین‌های فصل سوم

تمرین ۱

یک دیتابیس به نام SchoolDB با SSMS بسازید.

تمرین ۲

همان دیتابیس را با Script T-SQL بسازید.

تمرین ۳

برای دیتابیس ساخته‌شده یک User ایجاد کنید.

تمرین ۴

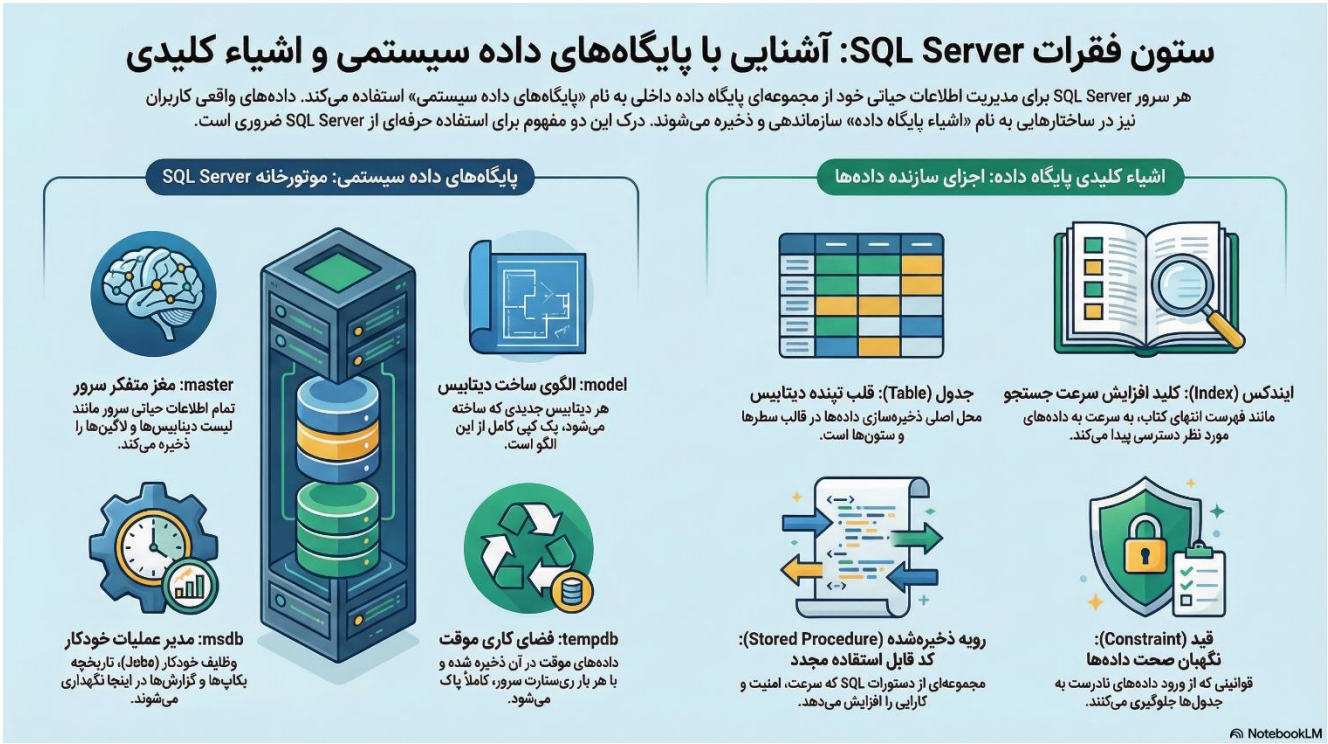
تفاوت Login و User را دوباره با مثال توضیح دهید.

تمرین ۵ (تحلیلی)

اگر User ساخته شود ولی Login حذف شود چه اتفاقی می‌افتد؟

فصل چهارم

معرفی پایگاه‌های داده سیستمی و آشنایی عمیق با اشیای پایگاه داده در Microsoft SQL Server



مقدمه

تا اینجا دانشجو:

- SQL Serverرا نصب کرده
- با محیط SSMS کار کرده
- پایگاه داده و کاربر ساخته

اما هنوز یک سؤال اساسی باقی مانده است:

»خود SQL Server اطلاعاتش را کجا و چطور نگه می‌دارد؟«

پاسخ این سؤال در **پایگاه‌های داده سیستمی** و **اشیای پایگاه داده** نهفته است.
فصل چهارم، ستون فقرات درک حرفه‌ای SQL Server است.

پایگاه داده سیستمی چیست؟

پایگاه داده سیستمی (System Database) دیتابیسی است که:

- توسط خود SQL Server ساخته می‌شود
- اطلاعات حیاتی سرور را نگه می‌دارد
- حذف یا دستکاری نادرست آن می‌تواند کل SQL Server را از کار بیندازد

معرفی پایگاه داده master (مهم‌ترین دیتابیس)

masterچیست؟

پایگاه داده **master**مغز SQL Server است.
تمام اطلاعات حیاتی سرور در این دیتابیس ذخیره می‌شود.

masterچه اطلاعاتی را نگه می‌دارد؟

- لیست تمام پایگاه‌های داده موجود
- اطلاعاتLogin ها
- تنظیماتInstance

- مسیر فایل‌های دیتابیس‌ها

📌اگر master آسیب ببیند:

- SQL Serverبالا نمی‌آید
- هیچ دیتابیسی شناخته نمی‌شود

به همین دلیل:

Backupگرفتن از master بسیار مهم است.

پایگاه داده msdb (دیتابیس مدیریت عملیات)

msdbچیست؟

پایگاه داده **msdb**برای مدیریت عملیات خودکار SQL Server استفاده می‌شود.

وظایف اصلی:msdb

- نگهداریJob های SQL Server Agent
- ذخیره اطلاعات Backup وRestore
- نگهداری گزارش‌ها و تاریخچه عملیات
- زمان‌بندی وظایف خودکار

📌مثال واقعی:
اگر SQL Server هر شب به‌صورت خودکار Backup می‌گیرد،
اطلاعات این کار در **msdb**ذخیره می‌شود.

پایگاه داده model (الگو یا قالب)

modelچیست؟

پایگاه داده **model**یک قالب (Template) است.

هر زمان که: یک دیتابیس جدید ساخته می‌شود

SQL Server: از model کپی می‌گیرد

کاربرد آموزشی:model

- اگر در model جدول یا تنظیم خاصی باشد، همه دیتابیس‌های جدید آن را خواهند داشت

پایگاه داده tempdb (موقتی اما حیاتی)

tempdbچیست؟

tempdbپایگاه داده‌ای است برای:

- داده‌های موقت
- عملیات موقتیSQL Server

چه چیزهایی در tempdb ذخیره می‌شود؟

- جدول‌های موقت
- نتایج میانیQuery ها
- عملیات Sort وJoin
- داده‌هایSession ها

📌ویژگی بسیار مهم:
با هر بار Restart شدنSQL Server ، **tempdbکاملاً پاک می‌شود.**

مقایسه پایگاه‌های داده سیستمی

نام دیتابیس	نقش اصلی	حساسیت
master	اطلاعات حیاتی سرور	بسیار زیاد
msdb	عملیات وJob ها	زیاد
model	الگوی دیتابیس جدید	متوسط
tempdb	داده‌های موقت	زیاد

اشیای پایگاه داده (Database Objects) چیستند؟

اشیای پایگاه داده اجزایی هستند که:

- داده‌ها را نگه می‌دارند
- روی داده‌ها عملیات انجام می‌دهند
- منطق برنامه را پیاده‌سازی می‌کنند

بدون این اشیا، دیتابیس فقط یک پوسته خالی است.

Table (جدول) — مهم‌ترین شیء پایگاه داده

Tableجایی است که:

- داده‌ها واقعاً ذخیره می‌شوند

ساختار Table:

- ستون‌ها(Column)
- ردیف‌ها(Row)

📌تمام مفاهیم بعدی به Table وابسته هستند.

View (نمای مجازی)

Viewیک جدول واقعی نیست.

بلکه:

نتیجه یک Query ذخیره‌شده است

کاربرد View:

- ساده‌سازیQuery های پیچیده
- افزایش امنیت (مخفی کردن ستون‌ها)
- استفاده مجدد از منطقQuery

Index (ایندکس) — افزایش سرعت

Indexشبیه: فهرست آخر کتاب

بدون Index : جستجو کند می‌شود

باIndex : سرعت Query به‌شدت افزایش می‌یابد

Index 📌روی Table ساخته می‌شود ولی داده مستقل ندارد.

Function (تابع)

Functionمجموعه‌ای از دستورات است که:

- مقدار مشخصی را برمی‌گرداند

انواع:

- Scalar Function(یک مقدار)
- Table-Valued Function(جدول)

کاربرد:

- محاسبات
- استفاده مجدد از منطق

Stored Procedure (رویه ذخیره‌شده)

Stored Procedureمجموعه‌ای از دستورات SQL است که:

- یک‌بار نوشته می‌شود
- بارها اجرا می‌شود

مزایا:

- سرعت بالاتر
- امنیت بیشتر
- کاهش خطا

Trigger (ماشه)

Triggerبه‌صورت خودکار اجرا می‌شود:

- بعد ازInsert
- بعد ازUpdate
- بعد ازDelete

کاربرد:

- کنترل صحت داده
- ثبت لاگ
- جلوگیری از عملیات غیرمجاز

Constraint (قیدها)

Constraintبرای:

حفظ صحت داده‌ها

انواع رایج:

- Primary Key
- Foreign Key
- Unique
- Check
- Not Null

بدون:Constraint

- دیتابیس قابل اعتماد نیست.

Schema (طرحواره)

Schemaمثل: یک پوشه برای اشیای دیتابیس

مزایا:

- نظم‌دهی ، مدیریت بهتر، کنترل دسترسی

پیش‌فرض: dbo

Synonym (نام مستعار)

Synonym اجازه می‌دهد: برای یک شیء، نام مستعار تعریف کنیم

کاربرد:

- ساده‌سازی نام‌ها
- اتصال به دیتابیس‌های دیگر

Diagram (نمودار پایگاه داده)

Diagramنمایش گرافیکی:

- جدول‌ها
- ارتباط‌ها

تمرین‌های فصل چهارم

۱. نقش هر یک از دیتابیس‌هایmaster ، msdb، modelو tempdb را توضیح دهید.

۲. چرا tempdb بعد از Restart پاک می‌شود؟

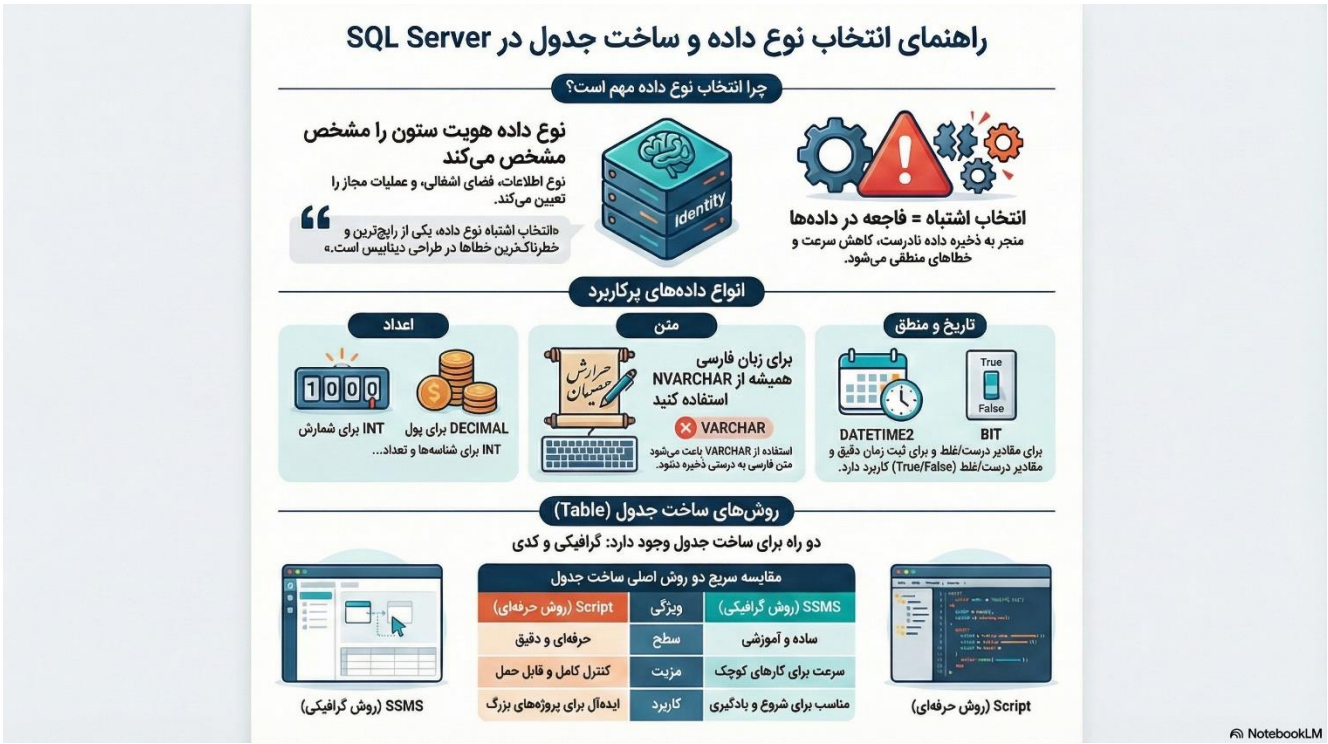
۳. تفاوت Table و View چیست؟

٤. Stored Procedureچه مزیتی نسبت به Query معمولی دارد؟

٥. Constraintچرا برای صحت داده مهم است؟

فصل پنجم

معرفی انواع داده‌ها و ساخت جداول پایگاه داده با SSMS و Script در Microsoft SQL Server



مقدمه

در فصل‌های قبل:

- پایگاه داده ساختیم
- کاربر تعریف کردیم
- با اشیای دیتابیس آشنا شدیم

اما پایگاه داده بدون **جدول (Table)** و جدول بدون **نوع داده (Data Type)** هیچ ارزشی ندارد.

در این فصل، دانشجو یاد می‌گیرد:

- هر نوع داده دقیقاً چه کاربردی دارد
- چه زمانی از چه نوع داده‌ای استفاده کند
- چگونه جدول‌ها را اصولی طراحی و ایجاد کند
- تفاوت ساخت جدول با SSMS و Script چیست

📌انتخاب اشتباه نوع داده، یکی از **رایج‌ترین و خطرناک‌ترین خطاها** در طراحی دیتابیس است.

نوع داده (Data Type) چیست؟

نوع داده مشخص می‌کند:

- چه نوع اطلاعاتی در یک ستون ذخیره شود
- چه مقدار فضا اشغال شود
- چه عملیات‌هایی روی داده مجاز باشد

مثلاً:

- نام ⟶ متن
- سن ⟶ عدد
- تاریخ تولد ⟶ تاریخ
- نمره ⟶ عدد اعشاری

اگر نوع داده اشتباه انتخاب شود:

- داده نادرست ذخیره می‌شود
- سرعت کاهش می‌یابد
- خطاهای منطقی ایجاد می‌شود

دسته‌بندی کلی انواع داده در SQL Server

در SQL Server ، انواع داده‌ها به چند گروه اصلی تقسیم می‌شوند:

۱. داده‌های عددی(Numeric)
۲. داده‌های متنی(Character / String)
۳. داده‌های تاریخ و زمان(Date & Time)
٤. داده‌های منطقی و خاص
٥. داده‌های دودویی(Binary)
٦. داده‌های ویژه(Special Types)

در ادامه هر گروه را **کاملاً تشریحی** بررسی می‌کنیم.

انواع داده عددی(Numeric Data Types)

INT: پرکاربردترین نوع داده عددی.

ویژگی‌ها:

- ذخیره اعداد صحیح
- بدون اعشار
- مناسب شناسه‌ها، سن، تعداد

مثال:

```
Age INT
```

BIGINT : مشابه INT اما با ظرفیت بسیار بیشتر.

کاربرد:

- شمارنده‌های بسیار بزرگ
- سیستم‌های پرترافیک

SMALLINT وTINYINT : برای اعداد کوچک‌تر استفاده می‌شوند و فضای کمتری اشغال می‌کنند.

📌نکته طراحی:
در پروژه‌های واقعی، انتخاب نوع عددی مناسب باعث **کاهش حجم دیتابیس** می‌شود.

DECIMAL / NUMERIC : برای اعداد اعشاری دقیق استفاده می‌شود.

مثال:

```
Score DECIMAL(5,2)
```

یعنی:

- کل ارقام: ۵
- اعشار: ۲

مناسب:

- نمره، مبلغ، قیمت

انواع داده متنی(String Data Types)

CHAR : طول ثابت دارد.

مثال:

```
Gender CHAR(1)
```

حتی اگر مقدار کوتاه باشد، فضا کامل اشغال می‌شود.

VARCHAR: طول متغیر دارد.

مثال:

LastName VARCHAR(50)

📌پرکاربردترین نوع داده متنی.

NVARCHAR(**بسیار مهم**) : برای ذخیره **زبان فارسی** استفاده می‌شود.

مثال:

FirstName NVARCHAR(50)

اگر از VARCHAR استفاده کنید:

- متن فارسی به‌درستی ذخیره نمی‌شود

TEXT / NTEXT(**منسوخ**) : در نسخه‌های جدید توصیه نمی‌شود.

به‌جای آن:

NVARCHAR(MAX)

انواع داده تاریخ و زمان

DATE

فقط تاریخ (سال، ماه، روز)

TIME

فقط زمان (ساعت، دقیقه، ثانیه)

DATETIME

تاریخ + زمان

مثال:

RegisterDate DATETIME

📌پرکاربرد در:

- ثبت‌نام، لاگ سیستم، گزارش‌ها

DATETIME2: نسخه دقیق‌تر و بهینه‌ترDATETIME

در طراحی جدید **توصیه می‌شود.**

داده‌های منطقی و خاص

BIT: برای ذخیره مقادیر:

- ۰ (False)
- ۱ (True)

مثال:

IsActive BIT

UNIQUEIDENTIFIER: برای ذخیرهGUID

مناسب سیستم‌های توزیع‌شده.

داده‌های دودویی(Binary)

برای ذخیره:

- تصویر
- فایل
- داده رمزنگاری‌شده

نوع رایج:

VARBINARY(MAX)

📌در پروژه‌های بزرگ، ذخیره فایل در دیتابیس توصیه نمی‌شود مگر ضرورت.

اصول انتخاب نوع داده (نکات طراحی حرفه‌ای)

۱. کوچک‌ترین نوع داده مناسب را انتخاب کنید

۲. برای فارسی حتماًNVARCHAR

۳. برای پول از DECIMAL استفاده کنید

٤. از DATETIME2 به‌جای DATETIME استفاده کنید

٥. از NULL و NOT NULL آگاهانه استفاده کنید

ساخت جدول (Table) چیست؟

Tableمحلی است که:

- داده‌ها به‌صورت ردیف و ستون ذخیره می‌شوند
- ستون‌ها نوع داده دارند
- ردیف‌ها داده واقعی هستند

بدون:Table

پایگاه داده فقط یک پوسته خالی است.

ساخت جدول با استفاده از SSMS (روش گرافیکی)

مرحله ۱

مسیر:

Databases

└── StudentDB

└── Tables

Right Click → New → Table

مرحله ۲: تعریف ستون‌ها

برای هر ستون:

- Name
- Data Type
- Allow Nulls

مثال:

- StudentID → INT
- FirstName → NVARCHAR(50)
- Age → INT

مرحله ۳: ذخیره جدول

نام جدول را وارد می‌کنیم:

Students

ساخت جدول با استفاده از Script (T-SQL)

روش حرفه‌ای و دقیق‌تر.

مثال کامل:

```
CREATE TABLE Students
(
   StudentID INT NOT NULL,
   FirstName NVARCHAR(50),
   LastName NVARCHAR(50),
   Age INT,
   RegisterDate DATETIME2
);
```

تفاوت ساخت جدول با SSMS وScript

SSMS:

- ساده
- آموزشی
- کند در پروژه بزرگ

Script:

- حرفه‌ای
- قابل ذخیره
- قابل انتقال
- کنترل کامل

خطاهای رایج دانشجویان

۱. استفاده از VARCHAR برای فارسی

۲. انتخاب INT برای مقادیر اعشاری

۳. نداشتن NOT NULL برای ستون‌های مهم

٤. نام‌گذاری نامناسب ستون‌ها

تمرین‌های فصل پنجم

تمرین ۱

جدولی برای اطلاعات دانشجو طراحی کنید.

تمرین ۲

برای هر ستون، دلیل انتخاب نوع داده را بنویسید.

تمرین ۳ (عملی)

همان جدول را:

- یک‌بار باSSMS و یک‌بار باScript بسازید.

تمرین ۴

تفاوت VARCHAR و NVARCHAR را توضیح دهید.

فصل ششم

آشنایی با انواع کلیدها و روابط بین جداول و ایجاد ارتباط بین جداول با Primary Keyو Foreign Key در Microsoft SQL Server

مقدمه (چرا کلیدها و روابط حیاتی هستند؟)

فرض کنید:

- یک جدول دانشجو داریم
- یک جدول درس داریم
- یک جدول انتخاب واحد داریم

اگر بین این جدول‌ها **ارتباط منطقی** وجود نداشته باشد:

- داده‌ها تکراری می‌شوند
- خطاهای زیاد ایجاد می‌شود
- گزارش‌گیری غیرممکن می‌شود

📌 **کلیدها (Keys)** و **روابط (Relationships)** باعث می‌شوند:

- داده‌ها منظم باشند
- یکپارچگی اطلاعات حفظ شود
- پایگاه داده هوشمند عمل کند

کلید (Key) چیست؟

کلید، یک ستون (یا مجموعه‌ای از ستون‌ها) است که:

- هر رکورد را به‌صورت یکتا شناسایی می‌کند
- ارتباط بین جدول‌ها را ممکن می‌سازد

به زبان ساده:

کلید = شناسنامه هر سطر جدول

انواع کلیدها در SQL Server

در SQL Server چند نوع کلید مهم داریم:

۱. Primary Key

۲. Foreign Key

۳. Candidate Key

٤. Composite Key

٥. Unique Key (به‌صورت Constraint)

در این فصل تمرکز اصلی روی **Primary Key** و **Foreign Key** است.

کلید اصلی (Primary Key) چیست؟

Primary Key ستونی است که:

- هر مقدار آن **منحصر به‌فرد** است
- مقدار NULL ندارد
- هر رکورد را یکتا مشخص می‌کند

📌هر جدول:

- فقط **یک Primary Key** دارد
- اما می‌تواند چند ستون در آن مشارکت داشته باشند

مثال ساده Primary Key

جدول:Students

StudentID	FirstName	LastName
۱	علی	احمدی
۲	سارا	رضایی

در این جدول: StudentID بهترین گزینه برای Primary Key است.

ویژگی‌های مهمPrimary Key

Primary Key:

- تکراری نیست
- NULLنیست
- سرعت جستجو را بالا می‌برد
- اساس ارتباط با جدول‌های دیگر است

📌بدونPrimary Key : جدول «استاندارد» محسوب نمی‌شود.

ایجاد Primary Key هنگام ساخت جدول(Script)

```
CREATE TABLE Students
(
  StudentID INT NOT NULL,
  FirstName NVARCHAR(50),
  LastName NVARCHAR(50),
  CONSTRAINT PK_Students PRIMARY KEY (StudentID)
);
```

در اینجا:

- PK_Studentsنام کلید است
- StudentIDکلید اصلی جدول است

۷-۶ایجاد Primary Key با SSMS (روش گرافیکی)

مراحل:

۱. روی Table راست‌کلیک-->Design

۲. ستون مورد نظر را انتخاب کنید

۳. Right Click → Set Primary Key

٤. ذخیره جدول

📌ستون کلید اصلی با علامت 🔑 مشخص می‌شود.

کلید خارجی (Foreign Key) چیست؟

Foreign Keyستونی است که:

- به Primary Key جدول دیگر اشاره می‌کند
- ارتباط بین دو جدول را برقرار می‌کند

به زبان ساده:

Foreign Key= پل ارتباطی بین جدول‌ها

مثال مفهومی Foreign Key

- جدول Students → StudentID (Primary Key)
- جدول Enrollments → StudentID (Foreign Key)

یعنی:

هر انتخاب واحد مربوط به یک دانشجو است.

ساختار منطقی ارتباط بین جدول‌ها

جدول فرزند(Child)	جدول والد(Parent)
Enrollments	Students
Enrollments	Courses

📌جدول فرزند دارای Foreign Key است.

ایجاد جدول‌ها و ارتباط با Primary Key و Foreign Key (Script)

جدول Students

```
CREATE TABLE Students
(
  StudentID INT NOT NULL,
  FirstName NVARCHAR(50),
  CONSTRAINT PK_Students PRIMARY KEY (StudentID)
);
```

جدول Courses

```
CREATE TABLE Courses
(
  CourseID INT NOT NULL,
  CourseName NVARCHAR(50),
  CONSTRAINT PK_Courses PRIMARY KEY (CourseID)
);
```

جدول Enrollments (ایجاد Foreign Key)

```
CREATE TABLE Enrollments
(
  EnrollmentID INT NOT NULL,
  StudentID INT NOT NULL,
  CourseID INT NOT NULL,
  CONSTRAINT PK_Enrollments PRIMARY KEY (EnrollmentID),
  CONSTRAINT FK_Enrollments_Students FOREIGN KEY (StudentID)
    REFERENCES Students(StudentID),
  CONSTRAINT FK_Enrollments_Courses FOREIGN KEY (CourseID)
    REFERENCES Courses(CourseID)
);
```

📌اینجا:

- StudentIDو CourseID کلید خارجی هستند
- ارتباط بین جدول‌ها برقرار شده است

ایجاد Foreign Key با SSMS (روش گرافیکی)

مراحل:

۱. Designجدول فرزند

۲. Right Click → Relationships

۳. Add

٤. مشخص کردن:

- Primary Key Table
- Foreign Key Column

٥. ذخیره جدول

انواع روابط بین جداول

۱– رابطه یک به یک(One to One)

مثال:

- Person
- PersonDetail

کم‌کاربرد

۲– رابطه یک به چند (One to Many) (رایج‌ترین)

مثال:

- Students → Enrollments

یک دانشجو، چند انتخاب واحد.

۳– رابطه چند به چند(Many to Many)

با جدول واسط انجام می‌شود.

مثال:

- Students ↔ Courses
- جدول واسطEnrollments :

مزایای استفاده از کلیدها و روابط

- جلوگیری از داده‌های تکراری
- حفظ یکپارچگی اطلاعات(Data Integrity)
- افزایش دقت گزارش‌ها
- طراحی استاندارد دیتابیس

تمرین‌های فصل ششم

تمرین ۱

سه جدول:

- Students
- Courses
- Enrollments

طراحی کنید.

تمرین ۲: Primary Keyهر جدول را مشخص کنید و دلیل بیاورید.

تمرین ۳ : Foreign Keyها را به‌صورت Script ایجاد کنید.

تمرین ۴ : روابط بین جدول‌ها را رسم کنید.(ER Diagram)

فصل هفتم

آشنایی با انواع دستورات SQL شاملDQL ، DML، DDL، DCLو TCLدر Microsoft SQL Server

۱-۷مقدمه فصل هفتم SQL) چگونه با پایگاه داده صحبت می‌کند؟(

SQLزبانی است که از طریق آن:

- داده‌ها را **می‌خوانیم**
- داده‌ها را **اضافه، ویرایش و حذف** می‌کنیم
- ساختار دیتابیس را **ایجاد یا تغییر** می‌دهیم
- سطح دسترسی کاربران را **کنترل** می‌کنیم
- عملیات‌ها را **مدیریت و ایمن** می‌کنیم

برای نظم و درک بهتر، دستورات SQL به **۵ گروه اصلی** تقسیم می‌شوند:

کاربرد	نام کامل	گروه
پرس‌وجوی داده	Data Query Language	DQL
تغییر داده	Data Manipulation Language	DML
تعریف ساختار	Data Definition Language	DDL
کنترل دسترسی	Data Control Language	DCL
کنترل تراکنش	Transaction Control Language	TCL

۷-۲ DQL (Data Query Language)

DQLچیست؟

دستورات DQL برای:

- **خواندن و مشاهده داده‌ها**
- بدون تغییر در اطلاعات دیتابیس

📌مهم‌ترین دستور:DQL

SELECT

دستور) SELECT پایه‌ای‌ترین دستور(SQL

نمایش تمام اطلاعات یک جدول:

```
SELECT * FROM Students;
```

نمایش ستون‌های خاص:

```
SELECT FirstName, LastName FROM Students;
```

شرط‌گذاری باWHERE

```
SELECT * FROM Students
WHERE Age > 20;
```

مرتب‌سازی با ORDER BY

```
SELECT * FROM Students
ORDER BY LastName ASC;
```

فیلتر داده‌ها با AND و OR

```
SELECT * FROM Students
WHERE Age > 18 AND IsActive = 1;
```

📌 نکته مهم:
DQL فقط داده را **می‌خواند** و هیچ تغییری ایجاد نمی‌کند.

۳-۷ DML (Data Manipulation Language)

DML چیست؟

دستورات DML برای:

- **افزودن**
- **ویرایش**
- **حذف** داده‌ها

روی **محتوای جدول** کار می‌کنند نه ساختار آن.

دستور INSERT (افزودن داده)

```
INSERT INTO Students (StudentID, FirstName, Age)
```

', 20);علیVALUES (1, N'

📌برای متن فارسی:

'Nمتن فارسی'

دستور) UPDATE ویرایش داده(

UPDATE Students

SET Age = 21

WHERE StudentID = 1;

⛔هشدار:

بدون WHERE تمام رکوردها تغییر می‌کنند!

دستور) DELETE حذف داده(

DELETE FROM Students

WHERE StudentID = 1;

تفاوت DELETE وTRUNCATE

DELETE	TRUNCATE
قابل برگشت (در تراکنش)	غیرقابل برگشت
شرط دارد	بدون شرط
کندتر	سریع‌تر

۷-۴ DDL (Data Definition Language)

DDLچیست؟

دستورات DDL برای:

- ایجاد
- تغییر
- حذف ساختار دیتابیس

یعنی:

اسکلت پایگاه داده

دستور CREATE

ایجاد جدول:

```
CREATE TABLE Courses
(
CourseID INT PRIMARY KEY,
CourseName NVARCHAR(50)
);
```

دستور ALTER

تغییر ساختار جدول:

اضافه کردن ستون:

```
ALTER TABLE Students
ADD Email NVARCHAR(100);
```

دستور DROP

حذف کامل شیء:

```
DROP TABLE Courses;
```

⛔غیرقابل بازگشت

دستور TRUNCATE

حذف تمام داده‌ها بدون حذف جدول:

```
TRUNCATE TABLE Students;
```

۷-۵ DCL (Data Control Language)

DCL چیست؟

برای:

- مدیریت دسترسی کاربران
- تعیین مجوزها

در دیتابیس‌های چندکاربره بسیار حیاتی است.

دستور GRANT

دادن مجوز:

```
GRANT SELECT ON Students TO User1;
```

دستور REVOKE

لغو مجوز:

```
REVOKE SELECT ON Students FROM User1;
```

📌با DCL می‌توان مشخص کرد:

- چه کسی فقط بخواند

- چه کسی ویرایش کند
- چه کسی مدیر باشد

۷-۶ TCL (Transaction Control Language)

TCLچیست؟

برای کنترل **تراکنش‌ها** استفاده می‌شود.

تراکنش:

مجموعه‌ای از عملیات که باید یا **همه اجرا شوند** یا **هیچ‌کدام اجرا نشوند**

دستورBEGIN TRANSACTION

```
BEGIN TRANSACTION;
```

دستورCOMMIT

ثبت تغییرات:

```
COMMIT;
```

دستورROLLBACK

بازگشت تغییرات:

```
ROLLBACK;
```

مثال کامل تراکنش

```
BEGIN TRANSACTION;
```

```
UPDATE Students SET Age = 22 WHERE StudentID = 1;
UPDATE Students SET Age = 23 WHERE StudentID = 2;

ROLLBACK;
```

📌هیچ تغییری ذخیره نمی‌شود.

۷-۷مقایسه کامل گروه دستوراتSQL

گروه	روی چه چیزی اثر دارد؟	قابل بازگشت
DQL	داده	❌
DML	داده	✅
DDL	ساختار	❌
DCL	دسترسی	❌
TCL	تراکنش	✅

۸-۷خطاهای رایج دانشجویان

۱. اجرای UPDATE بدونWHERE

۲. حذف جدول به‌جای داده

۳. استفاده نکردن ازTransaction

٤. ناآشنایی با GRANT وREVOKE

۹-۷تمرین‌های پایان فصل هفتم

تمرین ۱

۵ دستور SELECT با شرط‌های مختلف بنویسید.

تمرین ۲

یک رکورد درج، سپس ویرایش و در نهایت حذف کنید.

تمرین ۳

یک جدول ایجاد و سپس یک ستون به آن اضافه کنید.

تمرین ۴

سناریویی بنویسید که با ROLLBACK قابل برگشت باشد.

فصل هشتم

آشنایی با ساختار کلی دستور SELECT و بخش‌های مختلف آن در Microsoft SQL Server

---

۱-۸مقدمه فصل هشتم) چرا SELECT مهم‌ترین دستور SQL است؟(

تقریباً **۹۰٪ کار با پایگاه داده** شامل:

- استخراج اطلاعات
- گزارش‌گیری
- تحلیل داده

همه‌ی این‌ها با دستور **SELECT**انجام می‌شود.

اگر دانشجو:

- SELECTرا عمیق بفهمد
- اجزای آن را درست ترکیب کند

می‌تواند:

- گزارش‌های واقعی سازمانی بسازد
- داده‌ها را تحلیل کند
- در پروژه‌های عملی موفق باشد

۸-۲ساختار کلی دستور SELECT

ساختار استاندارد SELECT به‌صورت زیر است:

SELECTستون‌ها

FROMجدول

WHEREشرط

GROUP BYستون

HAVINGشرط گروه

ORDER BYستون

📌نکته بسیار مهم:
ترتیب نوشتن بخش‌ها **اجباری** است، اما اجرای منطقی آن در SQL متفاوت انجام می‌شود.

۸-۳بخش) SELECT انتخاب ستون‌ها(

وظیفه SELECT چیست؟

SELECTمشخص می‌کند:

- چه ستون‌هایی
- از چه داده‌هایی

نمایش داده شوند.

انتخاب همه ستون‌ها

SELECT * FROM Students;

⛔در پروژه‌های واقعی توصیه نمی‌شود (کاهش کارایی).

انتخاب ستون‌های خاص

SELECT FirstName, LastName FROM Students;

📌سرعت بیشتر و خوانایی بالاتر.

استفاده از نام مستعار(Alias)

SELECT FirstName AS نام, LastName AS نام_خانوادگی

FROM Students;

۴-۸بخش) WHERE فیلتر کردن داده‌ها(

WHEREچه کاری انجام می‌دهد؟

WHEREداده‌ها را **فیلتر** می‌کند.

یعنی:

فقط رکوردهایی که شرط را دارند نمایش داده می‌شوند.

مثال سادهWHERE

SELECT * FROM Students

WHERE Age > 20;

عملگرهای شرطی مهم

عملگر	معنی
=	مساوی
<>	نامساوی
>	بزرگ‌تر
<	کوچک‌تر
BETWEEN	بین دو مقدار
LIKE	الگو
IN	داخل مجموعه

مثال LIKE

```
SELECT * FROM Students
WHERE LastName LIKE N'ا%';
```

(نام خانوادگی‌هایی که با «ا» شروع می‌شوند)

ترکیب شرط‌ها

```
SELECT * FROM Students
WHERE Age > 18 AND IsActive = 1;
```

۵-۸بخش) ORDER BY مرتب‌سازی(

ORDER BY چیست؟

برای مرتب‌سازی نتایج خروجی استفاده می‌شود.

مرتب‌سازی صعودی (پیش‌فرض)

```
SELECT * FROM Students
ORDER BY LastName;
```

مرتب‌سازی نزولی

```
SELECT * FROM Students
ORDER BY Age DESC;
```

مرتب‌سازی چندستونه

```
SELECT * FROM Students
ORDER BY LastName ASC, FirstName ASC;
```

۶-۸بخش) GROUP BY گروه‌بندی داده‌ها(

GROUP BY چیست؟

برای **گروه‌بندی داده‌ها** و استفاده از **توابع تجمیعی** به‌کار می‌رود.

توابع تجمیعی:

- COUNT
- SUM
- AVG
- MIN
- MAX

مثال GROUP BY

تعداد دانشجویان هر شهر:

```
SELECT City, COUNT(*) AS Total
FROM Students
GROUP BY City;
```

📌هر ستونی که در SELECT است و تجمیعی نیست، باید در GROUP BY بیاید.

۷-۸توابع تجمیعی پرکاربرد

COUNT

COUNT(*)

تعداد رکوردها

SUM

SUM(Score)

جمع مقادیر عددی

AVG

AVG(Age)

میانگین

MIN / MAX

کوچک‌ترین و بزرگ‌ترین مقدار

۸-۸بخش) HAVING فیلتر گروه‌ها(

تفاوت WHERE وHAVING

HAVING	WHERE
فیلتر گروه	فیلتر رکورد
بعد ازGROUP BY	قبل ازGROUP BY

مثالHAVING

نمایش شهرهایی با بیش از ۵ دانشجو:

```
SELECT City, COUNT(*) AS Total
FROM Students
GROUP BY City
HAVING COUNT(*) > 5;
```

۹-۸ترتیب منطقی اجرای SELECT درSQL

اگرچه ما به این ترتیب می‌نویسیم:

```
SELECT
FROM
WHERE
GROUP BY
HAVING
ORDER BY
```

اما SQL به این ترتیب اجرا می‌کند:

۱. FROM
۲. WHERE
۳. GROUP BY
٤. HAVING

۵. SELECT

۶. ORDER BY

📌دانستن این ترتیب برای نوشتن Query حرفه‌ای ضروری است.

۸-۱۰مثال جامع ترکیبی (خیلی مهم)

```
SELECT City, AVG(Age) AS AvgAge
FROM Students
WHERE IsActive = 1
GROUP BY City
HAVING AVG(Age) > 20
ORDER BY AvgAge DESC;
```

این Query:

- فقط دانشجویان فعال
- میانگین سن هر شهر
- شهرهایی با میانگین سن بالای ۲۰
- مرتب‌شده نزولی

۸-۱۱خطاهای رایج دانشجویان

۱. استفاده از ستون غیرگروه‌بندی‌شده در SELECT

۲. اشتباه گرفتن WHERE و HAVING

۳. ORDER BY قبل از GROUP BY

۴. استفاده زیاد از * SELECT

۸-۱۲تمرین‌های پایان فصل هشتم

تمرین ۱

دانشجویان بالای ۲۱ سال را نمایش دهید.

تمرین ۲

تعداد دانشجویان هر شهر را محاسبه کنید.

تمرین ۳

شهرهایی که بیش از ۳ دانشجو دارند را نمایش دهید.

تمرین ۴ (پروژه‌ای)

گزارشی تهیه کنید که:

- دانشجویان فعال
- میانگین سن
- گروه‌بندی بر اساس شهر
- مرتب‌سازی نزولی

فصل نهم

درج، حذف و به‌روزرسانی اطلاعات در جداول با استفاده از SSMS وScript SQL

(Insert, Update, Delete)در Microsoft SQL Server

۱-۹مقدمه فصل نهم (کار واقعی با داده‌ها)

تا اینجا:

- ساختار پایگاه داده را ایجاد کرده‌ایم
- جدول ساخته‌ایم
- کلید و ارتباط تعریف کرده‌ایم
- داده‌ها را با SELECT خوانده‌ایم

در این فصل وارد **مرحله عملی واقعی** می‌شویم:

- ثبت اطلاعات جدید
- اصلاح اطلاعات اشتباه
- حذف اطلاعات غیرضروری

📌این فصل پایه‌ی:

- فرم‌های ثبت‌نام
- سیستم‌های اداری
- نرم‌افزارهای تحت وب و دسکتاپ

است.

۹-۲آشنایی با مفهومDML

دستورات **Insert**، **Update**و **Delete** جزو گروه **DML (Data Manipulation Language)**هستند.

ویژگی‌های مهم:DML

- روی **داده‌ها** کار می‌کنند نه ساختار
- قابل استفاده در **Transaction**هستند
- در صورت اشتباه می‌توان آن‌ها را Rollback کرد

۹-۳دستور) INSERT درج اطلاعات(

INSERTچیست؟

INSERTبرای:

- افزودن رکورد جدید
- ثبت اطلاعات تازه

در جدول استفاده می‌شود.

ساختار کلیINSERT

INSERT INTO TableName (Column1, Column2, ...)

VALUES (Value1, Value2, ...);

مثال سادهINSERT

INSERT INTO Students (StudentID, FirstName, Age)

', 20);علیVALUES (1, N'

📌نکته مهم:

- ترتیب ستون‌ها باید با مقادیر یکی باشد
- برای متن فارسی حتماً از **N**استفاده شود

INSERTبدون ذکر نام ستون‌ها

INSERT INTO Students

', 22);سارا VALUES (2, N'

⛔توصیه نمی‌شود

)وابسته به ترتیب ستون‌هاست(

درج چند رکورد هم‌زمان

```
INSERT INTO Students (StudentID, FirstName, Age)
VALUES
(3, N'محمد', 21),
(4, N'نگار', 19);
```

۹-۴ درج اطلاعات با SSMS (روش گرافیکی)

مراحل:

۱. روی جدول راست‌کلیک
۲. Edit Top 200 Rows
۳. وارد کردن داده در سلول‌ها
۴. خروج از جدول ← ذخیره خودکار

📌 مناسب آموزش و تست
⛔ نامناسب پروژه‌های حرفه‌ای

۹-۵ خطاهای رایج در INSERT

۱. عدم تطابق نوع داده
۲. تکرار Primary Key
۳. فراموش کردن N برای فارسی
۴. درج NULL در ستون NOT NULL

۹-۶ دستور UPDATE (به‌روزرسانی اطلاعات)

UPDATE چیست؟

برای:

- اصلاح داده‌های موجود
- تصحیح اطلاعات اشتباه

استفاده می‌شود.

ساختار کلی UPDATE

UPDATE TableName

SET Column = Value

WHERE Condition;

مثال UPDATE با شرط

UPDATE Students

SET Age = 21

WHERE StudentID = 1;

UPDATE چند ستون

UPDATE Students

SET FirstName = N'علی اکبر',

Age = 22

WHERE StudentID = 1;

⛔ **خطر بسیار مهم UPDATE بدون WHERE**

UPDATE Students

SET Age = 18;

📌 تمام رکوردها تغییر می‌کنند!

۷-۹به‌روزرسانی با استفاده از SSMS

روش:

- Edit Top 200 Rows
- تغییر مقدار سلول
- خروج از جدول

📌ساده و بصری

⛔بدون کنترل دقیق

۸-۹دستور) DELETE حذف اطلاعات(

DELETEچیست؟

برای:

- حذف رکوردها
- پاک‌سازی داده‌های اضافی

استفاده می‌شود.

ساختارDELETE

```
DELETE FROM TableName
WHERE Condition;
```

مثالDELETE

```
DELETE FROM Students
WHERE StudentID = 4;
```

حذف گروهی داده‌ها

```
DELETE FROM Students
WHERE Age < 18;
```

⛔ خطر DELETE بدون WHERE

```
DELETE FROM Students;
```

📌 کل جدول پاک می‌شود!

۹-۹تفاوت DELETE و) TRUNCATE یادآوری(

TRUNCATE	DELETE
بدون شرط	شرط‌پذیر
غیرقابلRollback	قابلRollback
سریع‌تر	کندتر
لاگ ندارد	لاگ دارد

۱۰-۹استفاده از Transaction باInsert / Update / Delete

چرا Transaction مهم است؟

برای جلوگیری از:

- حذف اشتباه
- ویرایش ناخواسته
- خطای انسانی

مثالTransaction

```
BEGIN TRANSACTION;

UPDATE Students
SET Age = 25
WHERE StudentID = 2;

ROLLBACK;
```

📌هیچ تغییری ذخیره نمی‌شود.

ثبت نهایی تغییرات

```
COMMIT;
```

۱۱-۹ارتباط Insert وForeign Key

📌اگر Foreign Key وجود داشته باشد:

- نمی‌توان مقدار نامعتبر درج کرد

مثال:

```
INSERT INTO Enrollments (EnrollmentID, StudentID)
VALUES (1, 999);
```

⛔اگر StudentID = 999 وجود نداشته باشد ⟶ خطا

۱۲-۹خطاهای رایج دانشجویان درDML

۱. INSERTبدون توجه به کلید خارجی

۲. UPDATEبدونWHERE

۳. DELETEبدون بک‌آپ

٤. استفاده نکردن ازTransaction

۱۳-۹تمرین‌های پایان فصل نهم

تمرین ۱

۳ دانشجو جدید درج کنید.

تمرین ۲

سن یکی از دانشجویان را اصلاح کنید.

تمرین ۳

دانشجویان زیر ۱۸ سال را حذف کنید.

تمرین ۴ (مهم)

تمام مراحل بالا را داخل Transaction انجام دهید و سپس ROLLBACK کنید.

فصل دهم

آشنایی با توابع محاسباتی (Aggregate Functions) در Microsoft SQL Server

شاملMIN :، MAX، COUNT، SUM، AVG

۱-۱۰مقدمه فصل دهم (چرا توابع تجمیعی مهم هستند؟)

در دنیای واقعی، معمولاً نمی‌خواهیم:

- فقط داده خام ببینیم

بلکه می‌خواهیم بدانیم:

- چند نفر داریم؟
- کمترین و بیشترین مقدار چیست؟
- مجموع چقدر است؟
- میانگین چگونه است؟

📌این دقیقاً کاری است که **توابع محاسباتی (Aggregate Functions)** انجام می‌دهند.

کاربرد این توابع:

- گزارش‌گیری مدیریتی
- تحلیل نمرات، حقوق، فروش
- داشبوردها
- سیستم‌های آماری

۲-۱۰تابع محاسباتی (Aggregate Function) چیست؟

تابع محاسباتی:

- روی **چندین رکورد** کار می‌کند
- یک **خروجی واحد** تولید می‌کند

برخلاف توابع معمولی که:

- روی یک رکورد اجرا می‌شوند

لیست توابع محاسباتی اصلی

تابع	کاربرد
COUNT	شمارش
SUM	جمع

تابع	کاربرد
AVG	میانگین
MIN	کمترین مقدار
MAX	بیشترین مقدار

۳-۱۰نکات مهم قبل از شروع

۱. توابع Aggregate معمولاً همراه با **GROUP BY** استفاده می‌شوند

۲. مقدار NULL در محاسبات نادیده گرفته می‌شود

۳. نمی‌توان ستون عادی را کنار تابع Aggregate بدون GROUP BY آورد

۴-۱۰تابع) COUNT شمارش رکوردها(

COUNTچه کاری انجام می‌دهد؟

COUNTتعداد رکوردها را می‌شمارد.

انواعCOUNT

1. COUNT(*)

شمارش تمام رکوردها) حتیNULL)

```
SELECT COUNT(*) AS TotalStudents
FROM Students;
```

---

2. COUNT(ColumnName)

شمارش رکوردهایی که مقدار NULL ندارند

```
SELECT COUNT(Age) AS CountAge
FROM Students;
```

📌اگر Age برابر NULL باشد، شمرده نمی‌شود.

COUNTبا شرط

```
SELECT COUNT(*)
FROM Students
WHERE IsActive = 1;
```

۵–۱۰تابع) SUM محاسبه مجموع(

SUMچیست؟

جمع مقادیر عددی یک ستون.

مثال سادهSUM

```
SELECT SUM(Score) AS TotalScore
FROM Students;
```

📌فقط روی ستون‌های عددی قابل استفاده است.

SUMبا شرط

```
SELECT SUM(Salary)
FROM Employees
WHERE Department = N'فروش';
```

SUMهمراهGROUP BY

جمع حقوق هر دپارتمان:

```
SELECT Department, SUM(Salary) AS TotalSalary
FROM Employees
GROUP BY Department;
```

۶-۱۰تابع) AVG محاسبه میانگین(

AVGچیست؟

محاسبه میانگین مقادیر عددی.

مثالAVG

```
SELECT AVG(Age) AS AverageAge
FROM Students;
```

AVGبا شرط

```
SELECT AVG(Score)
FROM Students
WHERE CourseID = 1;
```

GROUP BYباAVG

میانگین نمره هر درس:

```
SELECT CourseID, AVG(Score) AS AvgScore
FROM Enrollments
GROUP BY CourseID;
```

۷-۱۰تابع) MIN کمترین مقدار(

MINچه کاری می‌کند؟

کمترین مقدار یک ستون را برمی‌گرداند.

مثال MIN عددی

```
SELECT MIN(Age) AS MinAge
FROM Students;
```

MINمتنی

```
SELECT MIN(LastName)
FROM Students;
```

📌بر اساس ترتیب الفبایی.

MINتاریخ

```
SELECT MIN(RegisterDate)
FROM Students;
```

قدیمی‌ترین تاریخ ثبت‌نام.

۸-۱۰تابع) MAX بیشترین مقدار(

MAXچیست؟

بیشترین مقدار یک ستون.

مثال MAX عددی

```
SELECT MAX(Score) AS MaxScore
FROM Students;
```

MAXتاریخ

```
SELECT MAX(RegisterDate)
FROM Students;
```

جدیدترین ثبت‌نام.

۹-۱۰استفاده هم‌زمان از چند تابعAggregate

```
SELECT
COUNT(*) AS Total,
MIN(Age) AS MinAge,
MAX(Age) AS MaxAge,
AVG(Age) AS AvgAge
FROM Students;
```

📌بسیار پرکاربرد در گزارش‌های مدیریتی.

۱۰-۱۰ترکیب Aggregate با) GROUP BY خیلی مهم(

مثال جامع

```
SELECT City,
COUNT(*) AS TotalStudents,
AVG(Age) AS AvgAge
FROM Students
GROUP BY City;
```

این Query:

- دانشجویان را بر اساس شهر گروه‌بندی می‌کند
- تعداد و میانگین سن هر شهر را محاسبه می‌کند

۱۱-۱۰استفاده از HAVING با توابع محاسباتی

چراHAVING ؟

چون WHERE روی نتیجه Aggregate کار نمی‌کند.

مثالHAVING

نمایش شهرهایی که بیش از ۵ دانشجو دارند:

```
SELECT City, COUNT(*) AS Total
FROM Students
GROUP BY City
HAVING COUNT(*) > 5;
```

۱۲-۱۰تفاوت WHERE و) HAVING یادآوری مهم(

HAVING	WHERE
بعد ازGROUP BY	قبل ازGROUP BY
روی گروه	روی رکورد
باAggregate	بدونAggregate

۱۳-۱۰خطاهای رایج دانشجویان

۱. استفاده از ستون غیرگروه‌بندی‌شده کنارAggregate

۲. استفاده از WHERE به‌جایHAVING

۳. انتظار محاسبه رویNULL

٤. استفاده SUM روی ستون متنی

۱۴-۱۰تمرین‌های پایان فصل دهم

تمرین ۱

تعداد کل دانشجویان را محاسبه کنید.

تمرین ۲

کمترین و بیشترین سن دانشجویان را نمایش دهید.

تمرین ۳

میانگین نمره هر درس را محاسبه کنید.

تمرین ۴

شهرهایی که میانگین سن آن‌ها بالاتر از ۲۲ است را نمایش دهید.

تمرین ۵ (تحلیلی)

گزارشی بسازید که شامل:

- تعداد
- میانگین
- کمترین

- بیشترین

نمره هر درس باشد.

فصل یازدهم

ایجاد View و استفاده از آن درQuery ها در Microsoft SQL Server

---

۱-۱۱مقدمه فصل یازدهم) چرا View اهمیت دارد؟(

در پایگاه داده‌های واقعی:

- Queryها طولانی می‌شوند
- JOINها زیاد می‌شوند
- کاربران نباید به همه جدول‌ها دسترسی مستقیم داشته باشند

اینجاست که **View**وارد می‌شود.

View 📌به ما کمک می‌کند:

- Queryهای پیچیده را ساده کنیم
- امنیت داده‌ها را بالا ببریم
- گزارش‌گیری را سریع‌تر و تمیزتر انجام دهیم
- ساختار دیتابیس را از کاربر پنهان کنیم

۲-۱۱چیست؟ View

Viewیک **جدول مجازی** است که:

- داده‌ها را ذخیره نمی‌کند
- نتیجه یک Query را نمایش می‌دهد
- همیشه داده‌های به‌روز را نشان می‌دهد

به زبان ساده:

Viewذخیره‌شده = Query

تفاوت View باTable

View	Table
داده ذخیره نمی‌کند	داده ذخیره می‌کند
فضای دیسک نمی‌گیرد	فضای دیسک می‌گیرد
وابسته به جدول	داده مستقل دارد
محدود	قابل Insert مستقیم

۳-۱۱چرا از View استفاده می‌کنیم؟

مهم‌ترین کاربردهای:View

۱. ساده‌سازیQuery های پیچیده

۲. افزایش امنیت داده‌ها

۳. محدود کردن ستون‌ها یا رکوردها

٤. استفاده مجدد ازQuery

٥. گزارش‌گیری استاندارد

📌در سازمان‌ها:

کاربر فقط با View کار می‌کند، نه جدول اصلی.

۴-۱۱ساختار کلی ایجادView

CREATE VIEW ViewName

AS

SELECT ...

FROM ...

WHERE ...

View 📌نام دارد و یک Query درون خود نگه می‌دارد.

۵-۱۱مثال ساده ساخت View

ساخت View برای نمایش اطلاعات دانشجویان

CREATE VIEW vw_Students

AS

SELECT StudentID, FirstName, LastName

FROM Students;

اکنون می‌توانیم بنویسیم:

SELECT * FROM vw_Students;

📌دقیقاً مثل جدول رفتار می‌کند.

۶-۱۱استفاده از View در Query ها

WHERE روی View

SELECT * FROM vw_Students

WHERE StudentID > 10;

ORDER BY روی View

SELECT * FROM vw_Students

ORDER BY LastName;

View 📌 کاملاً در SELECT قابل استفاده است.

۷-۱۱ایجاد View با شرط View) فیلترشده(

مثال: فقط دانشجویان فعال

```
CREATE VIEW vw_ActiveStudents
AS
SELECT *
FROM Students
WHERE IsActive = 1;
```

کاربرد:

- نمایش فقط داده‌های مجاز
- حذف نیاز به WHERE تکراری

View ۸-۱۱و) JOIN کاربردی‌ترین حالت(

ساخت View چندجدولی

```
CREATE VIEW vw_StudentCourses
AS
SELECT
S.StudentID,
S.FirstName,
C.CourseName
FROM Students S
INNER JOIN Enrollments E ON S.StudentID = E.StudentID
```

INNER JOIN Courses C ON E.CourseID = C.CourseID;

اکنون گزارش بسیار ساده می‌شود:

SELECT * FROM vw_StudentCourses;

📌بدون JOIN در Query نهایی!

View ۹-۱۱و توابعAggregate

مثال: تعداد دانشجویان هر درس

CREATE VIEW vw_CourseStats

AS

SELECT CourseID, COUNT(*) AS TotalStudents

FROM Enrollments

GROUP BY CourseID;

استفاده:

SELECT * FROM vw_CourseStats;

۱۰-۱۱ویرایش(ALTER VIEW) View

اگر بخواهیم View را تغییر دهیم:

ALTER VIEW vw_Students

AS

SELECT StudentID, FirstName

FROM Students;

📌 View بازنویسی می‌شود.

۱۱-۱۱حذف(DROP VIEW) View

```
DROP VIEW vw_Students;
```

⛔ فقط View حذف می‌شود، نه جدول اصلی.

۱۲-۱۱ایجاد View با) SSMS روش گرافیکی(

مراحل:

۱. Database → Views
۲. Right Click → New View
۳. انتخاب جدول‌ها
٤. نوشتنQuery
٥. Saveو نام‌گذاریView

📌مناسب آموزش
📌در پروژه حرفه‌ای Script ترجیح دارد.

۱۳-۱۱محدودیت‌هایView

۱. Viewداده ذخیره نمی‌کند
۲. برخیView ها قابل Insert / Update نیستند
۳. ORDER BYداخل View مجاز نیست) مگر با(TOP
٤. Viewبه جدول وابسته است

Update / Insert ۱۴-۱۱روی) View نکته پیشرفته(

اگر:View

- فقط از یک جدول
- بدونJOIN

- بدونAggregate

باشد، می‌توان:

INSERT INTO vw_Students

'اکبری');علی'VALUES (10, N' ,N'

📌اما درView های پیچیده معمولاً مجاز نیست.

View ۱۵-۱۱و امنیت (بسیار مهم)

سناریوی واقعی

- جدول Students شامل اطلاعات حساس است
- کاربر فقط باید نام را ببیند

راه‌حل:

CREATE VIEW vw_PublicStudents

AS

SELECT FirstName, LastName

FROM Students;

📌دسترسی فقط به View داده می‌شود، نه جدول.

۱۶-۱۱تفاوت View و) Stored Procedure مفهومی(

	View	Stored Procedure
عملیات پیچیده	فقطSELECT	
برنامه	جدول مجازی	
پیشرفته	ساده	

۱۷-۱۱خطاهای رایج دانشجویان

۱. تصور ذخیره شدن داده درView

۲. استفاده از ORDER BY داخلView

۳. حذف جدول و انتظار باقی ماندنView

٤. استفاده بی‌رویه ازView های تودرتو

۱۸-۱۱تمرین‌های پایان فصل یازدهم

تمرین ۱

Viewبرای نمایش نام و سن دانشجویان بسازید.

تمرین ۲

Viewبرای نمایش دانشجویان فعال ایجاد کنید.

تمرین ۳

Viewشامل JOIN بین Students و Courses بسازید.

تمرین ۴ (پروژه‌ای)

گزارشی بسازید که:

- نام درس
- تعداد دانشجویان

را از طریق View نمایش دهد.

فصل دوازدهم

ایجاد Stored Procedure و آشنایی با بخش‌های مختلف آن، ارسال پارامتر و فراخوانی روال‌ها در Microsoft SQL Server

---

۱-۱۲مقدمه فصل دوازدهم) چرا Stored Procedure بسیار مهم است؟(

در پروژه‌های واقعی:

- Queryها طولانی و پیچیده می‌شوند
- دستورات تکراری هستند
- امنیت اهمیت بالایی دارد
- سرعت اجرا مهم است

📌راه‌حل حرفه‌ای:SQL Server

Stored Procedure

با Stored Procedure می‌توان:

- منطق برنامه را داخل دیتابیس نگه داشت
- امنیت را افزایش داد
- کد را مرتب و قابل استفاده مجدد کرد
- سرعت اجرایQuery ها را بالا برد

۲-۱۲چیست؟ Stored Procedure

) Stored Procedureروال ذخیره‌شده:(

- مجموعه‌ای از دستورات SQL است
- با یک نام ذخیره می‌شود
- روی سرور اجرا می‌شود
- قابل فراخوانی مجدد است

به زبان ساده:

= Stored Procedureبرنامه کوچک داخل دیتابیس

تفاوت Stored Procedure و Query معمولی

Stored Procedure	Queryمعمولی
یک‌بار نوشته می‌شود	هر بار نوشته می‌شود
امنیت بیشتر	امنیت کمتر
سریع‌تر	کندتر
قابل مدیریت	قابل مدیریت نیست

۳-۱۲چرا از Stored Procedure استفاده می‌کنیم؟

مهم‌ترین مزایا:

۱. افزایش امنیت

۲. افزایش سرعت

۳. جلوگیری از تکرار کد

٤. ساده‌سازی برنامه‌نویسی

٥. مدیریت بهتر پروژه

📌در اکثر نرم‌افزارها:

ارتباط برنامه فقط با Stored Procedure است.

۴-۱۲ساختار کلیStored Procedure

CREATE PROCEDURE ProcedureName

```
AS
BEGIN
-- SQL Statements
END
```

۱۲-۵مثال ساده) Stored Procedure بدون پارامتر(

ایجادProcedure

```
CREATE PROCEDURE sp_GetAllStudents
AS
BEGIN
SELECT * FROM Students;
END
```

فراخوانیProcedure

```
EXEC sp_GetAllStudents;
```

یا

```
EXECUTE sp_GetAllStudents;
```

📌نتیجه:

- تمام اطلاعات دانشجویان نمایش داده می‌شود.

۱۲-۶اجزای اصلیStored Procedure

یک Stored Procedure معمولاً شامل:

۱. نامProcedure

۲. پارامترها (اختیاری)

۳. بدنه(BEGIN / END)

٤. دستوراتSQL

٥. دستورات کنترلیIF)، WHILEو(...

۷-۱۲ارسال پارامتر بهStored Procedure

چرا پارامتر مهم است؟

پارامتر باعث می‌شود:

- Procedureانعطاف‌پذیر شود
- یک Procedure چند کاربرد داشته باشد

ساخت Procedure با پارامتر ورودی

```
CREATE PROCEDURE sp_GetStudentById
@StudentID INT
AS
BEGIN
SELECT * FROM Students
WHERE StudentID = @StudentID;
END
```

فراخوانی Procedure با پارامتر

```
EXEC sp_GetStudentById 1;
```

یا

```
EXEC sp_GetStudentById @StudentID = 1;
```

۸-۱۲استفاده از چند پارامتر

```
CREATE PROCEDURE sp_GetStudentsByAge
@MinAge INT,
@MaxAge INT
AS
BEGIN
SELECT *
FROM Students
WHERE Age BETWEEN @MinAge AND @MaxAge;
END
```

فراخوانی:

```
EXEC sp_GetStudentsByAge 18, 25;
```

۹-۱۲پارامتر خروجی(OUTPUT Parameter)

پارامتر خروجی چیست؟

پارامتری که:

- مقدار از Procedure برمی‌گرداند

مثال پارامتر خروجی

```
CREATE PROCEDURE sp_GetStudentCount
@Total INT OUTPUT
AS
BEGIN
SELECT @Total = COUNT(*)
FROM Students;
```

END

فراخوانی و دریافت مقدار خروجی

DECLARE @Result INT;

EXEC sp_GetStudentCount @Total = @Result OUTPUT;

SELECT @Result AS TotalStudents;

Stored Procedure ۱۰-۱۲و دستوراتDML

مثال Insert باProcedure

CREATE PROCEDURE sp_InsertStudent

@StudentID INT,

@FirstName NVARCHAR(50),

@Age INT

AS

BEGIN

INSERT INTO Students (StudentID, FirstName, Age)

VALUES (@StudentID, @FirstName, @Age);

END

فراخوانی:

', 21;رضاEXEC sp_InsertStudent 10, N'

Stored Procedure ۱۱-۱۲وUPDATE

CREATE PROCEDURE sp_UpdateStudentAge

```
@StudentID INT,
@NewAge INT
AS
BEGIN
UPDATE Students
SET Age = @NewAge
WHERE StudentID = @StudentID;
END
```

۱۲-۱۲ Stored Procedure و DELETE

```
CREATE PROCEDURE sp_DeleteStudent
@StudentID INT
AS
BEGIN
DELETE FROM Students
WHERE StudentID = @StudentID;
END
```

📌امنیت بالا چون کاربر مستقیم DELETE نمی‌زند.

۱۲-۱۳ استفاده از شرط‌ها در Stored Procedure

```
CREATE PROCEDURE sp_CheckStudentAge
@Age INT
AS
BEGIN
IF @Age >= 18
```

```
    SELECT N'دانشجو مجاز است;Result AS '

ELSE

    SELECT N'دانشجو مجاز نیست;Result AS '

END
```

۱۲-۱۴ویرایش Stored Procedure

```
ALTER PROCEDURE sp_GetAllStudents

AS

BEGIN

SELECT StudentID, FirstName

FROM Students;

END
```

📌 Procedure بازنویسی می‌شود.

۱۲-۱۵حذف Stored Procedure

```
DROP PROCEDURE sp_GetAllStudents;
```

⛔ فقط Procedure حذف می‌شود، داده‌ها باقی می‌مانند.

۱۲-۱۶ایجاد Stored Procedure باSSMS

مراحل:

۱. Database → Programmability

۲. Stored Procedures

۳. Right Click → New → Stored Procedure

٤. نوشتن کد

۵. Save

📌مناسب آموزش

📌در پروژه حرفه‌ای Script توصیه می‌شود.

۱۲-۱۷ Stored Procedure و امنیت (بسیار مهم)

مزیت امنیتی:

- کاربر دسترسی مستقیم به جدول ندارد
- فقط EXEC Procedure مجاز است

```
GRANT EXECUTE ON sp_GetAllStudents TO User1;
```

📌یکی از دلایل اصلی استفاده در سازمان‌ها.

۱۲-۱۸خطاهای رایج دانشجویان

۱. فراموش کردنEXEC

۲. عدم تطابق نوع پارامتر

۳. عدم استفاده ازOUTPUT

٤. استفاده زیاد ازProcedure های تکراری

۱۲-۱۹تمرین‌های پایان فصل دوازدهم

تمرین ۱

Procedureبرای نمایش همه دانشجویان ایجاد کنید.

تمرین ۲

Procedureبا پارامتر سن بنویسید.

تمرین ۳

Procedureبرای Insert دانشجو ایجاد کنید.

تمرین ۴ (پروژه‌ای)

Procedureبنویسید که:

- نام درس را بگیرد
- تعداد دانشجویان آن درس را برگرداند(OUTPUT)