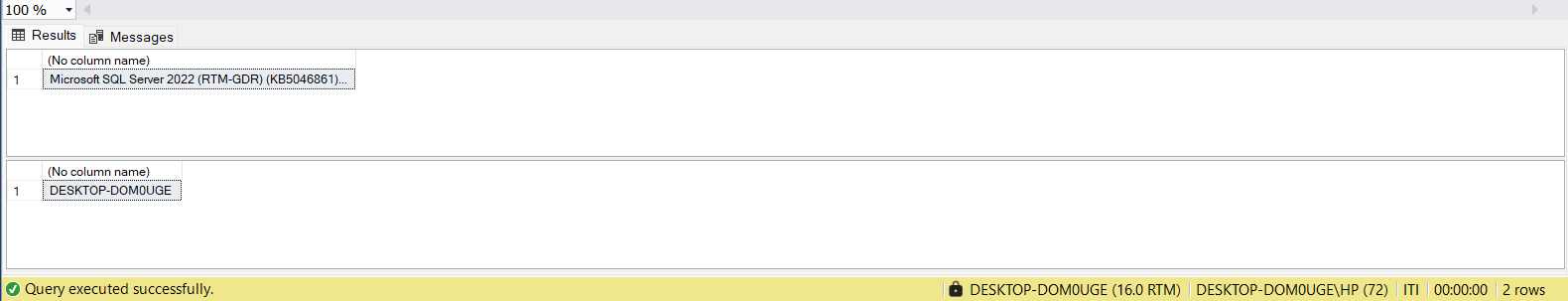
Display results of the following two statements and explain what is the meaning of @@AnyExpression

select @@VERSION

select @@SERVERNAME



1. اول حاجة الversion بيعرض الإصدار اللي شغال عليه دلوقتي على الجهاز انه مثلا Microsoft sql server 2022
2. تاني حاجة الserver اللي مشبك عليه دلوقتي من الجهاز نفسه يعني الجهاز نفسه شغال ع أي سيرفر حاليا

**Report**

1. ايه هو الClass Digram

الـ **Class Diagram** هو نوع من أنواع مخططات UML (Unified Modeling Language) وبيستخدم في **تصميم البرامج (خاصة في البرمجة الكائنية - OOP)**.

**🎯 تعريف بسيط:**

**Class Diagram** هو رسم بيبيّن الكلاسات (classes) في النظام، وبيوضّح العلاقات بينها، والخصائص (attributes) والوظائف (methods) اللي جوا كل كلاس.

**💡 بيتكوّن من إيه؟**

كل **Class** بيتعرض كـ "صندوق" فيه 3 أقسام:

scss

CopyEdit

┌────────────────────┐

│ اسم الكلاس (Class Name) │

├────────────────────┤

│ الخصائص (Attributes) │

├────────────────────┤

│ الدوال (Methods/Operations)│

└────────────────────┘

**🧱 مثال بسيط:**

**Class: Student**

pgsql

CopyEdit

┌───────────────┐

│ Student │

├───────────────┤

│ - id: int │

│ - name: string│

│ - age: int │

├───────────────┤

│ + register() │

│ + login() │

└───────────────┘

* - يعني الخاصية **private**
* + يعني الوظيفة **public**

**🔗 أنواع العلاقات بين الكلاسات:**

| **العلاقة** | **معناها** | **مثال** |
| --- | --- | --- |
| **Association** | علاقة عامة | طالب يدرس في جامعة |
| **Aggregation** | جزء من - لكن مش مملوك | قسم يحتوي على طلاب |
| **Composition** | جزء من - ومملوك | جسم الإنسان يحتوي على قلب |
| **Inheritance** (وراثة) | كلاس يرث من كلاس تاني | Admin يرث من User |

**📌 بيستخدم إمتى؟**

* وانت بتصمم نظام كبير قبل ما تكتب كود
* لو بتعمل تحليل أو توثيق لنظام مبرمج بـ OOP (زي Java, C#, Python OOP)
* في المشاريع الجامعية والتقارير الفنية

**أدوات ترسم بيها Class Diagram**

* draw.io
* StarUML
* Lucidchart
* Visual Paradigm

1. **ايه هو الbuild in stored presedure**

**يعني إيه Built-in Stored Procedure في SQL Server؟**

**Built-in Stored Procedures** هي إجراءات مخزّنة (stored procedures) جاهزة ومعمولة مسبقًا من قبل **SQL Server** نفسه، وموجودة جوه النظام.

يعني بدل ما تكتب كود من الصفر، تقدر تستخدم واحدة من دول علشان تنفّذ أو تستعلم أو تتحكم في حاجة في الداتا بيز.

**🛠 أمثلة مشهورة على Built-in Stored Procedures:**

| **الإجراء** | **وظيفته** |
| --- | --- |
| sp\_help | يعرض معلومات عن جدول أو فيو أو كائن معين |
| sp\_columns | يعرض معلومات الأعمدة (columns) في جدول معين |
| sp\_helptext | يعرض كود الإجراء المخزن أو الفانكشن |
| sp\_rename | يغيّر اسم جدول أو عمود |
| sp\_databases | يعرض أسماء قواعد البيانات |
| sp\_tables | يعرض كل الجداول في قاعدة البيانات |
| sp\_who | يعرض معلومات عن المستخدمين المتصلين |
| sp\_lock | يعرض معلومات عن الـ locks |
| sp\_depends | يعرض الكائنات اللي بتعتمد على كائن معين |

**📌 مثال عملي:**

**1. sp\_help**

sql

CopyEdit

EXEC sp\_help Employees;

يعرض كل المعلومات عن جدول Employees.

**2. sp\_rename**

sql

CopyEdit

EXEC sp\_rename 'Employees.Address', 'EmpAddress', 'COLUMN';

بيغيّر اسم العمود Address في جدول Employees إلى EmpAddress.

**3. sp\_helptext**

لو عندك stored procedure اسمها GetEmployees:

sql

CopyEdit

EXEC sp\_helptext 'GetEmployees';

يعرض كود الـ procedure دي.

**💡 ليه نستخدم الـ Built-in procedures؟**

* توفر وقت
* تريحك من كتابة استعلامات معقدة
* بتسهل إدارة قواعد البيانات

**⚠️ ملحوظة:**

كل الـ built-in stored procedures بيبدأوا بـ sp\_

1. أيه هو الSQL Injection

**يعني إيه SQL Injection؟**

**SQL Injection** هي **ثغرة (vulnerability)** بتحصل لما المبرمج يكتب كود SQL ويدخل فيه بيانات من المستخدم **من غير ما يحميها**، فـ المخترق يقدر يحقن (inject) كود SQL جوه البيانات.

يعني بدل ما المستخدم يدخل اسم، يدخل كود SQL يكسر القاعدة أو يسرّب بيانات.

**🧠 مثال واقعي:**

تخيل إنك عامل صفحة تسجيل دخول، والكود ده هو اللي بيشتغل ورا الكواليس:

sql

CopyEdit

SELECT \* FROM Users WHERE username = 'userInput' AND password = 'passInput';

وإنت بتنفّذ الكود ده كده (في C# أو PHP أو غيره):

sql

CopyEdit

string query = "SELECT \* FROM Users WHERE username = '" + username + "' AND password = '" + password + "'";

لو الهاكر كتب بدل الاسم:

sql

CopyEdit

' OR '1'='1

يبقى الكود النهائي بقى كده:

sql

CopyEdit

SELECT \* FROM Users WHERE username = '' OR '1'='1' AND password = ''

✅ والنتيجة؟: هيعدّي الدخول حتى لو مش عارف الباسورد 😱

**🎯 أخطر ما في SQL Injection:**

* يقدر يعدّي دخول من غير كلمة سر
* يقدر يمسح جداول
* يقدر ياخد نسخة من كل الداتا (سرقة بيانات)
* يقدر يوقف النظام بالكامل

**🛡️ إزاي نحمي نفسنا؟**

1. ✅ **استخدم parameterized queries**
   * زي SqlCommand مع @parameters في C# أو PreparedStatement في Java
2. ✅ **استخدم ORM** زي Entity Framework أو Sequelize
3. ✅ **فلتر البيانات المدخلة من المستخدم** (Sanitization)
4. ✅ **اقلل صلاحيات حساب الـ SQL اللي بيتصل بالداتا بيز**

**✅ مثال حماية باستخدام C#:**

csharp

CopyEdit

SqlCommand cmd = new SqlCommand("SELECT \* FROM Users WHERE username = @username AND password = @password", connection);

cmd.Parameters.AddWithValue("@username", usernameInput.Text);

cmd.Parameters.AddWithValue("@password", passwordInput.Text);

كده حتى لو حد كتب كود خبيث، هيتعامل كأنه **نص عادي** مش كود.

**📌 الخلاصة:**

| **💥 SQL Injection** | **👮‍♂️ الحل** |
| --- | --- |
| ثغرة بسبب دمج البيانات مع الكود مباشرة | استخدم parameterized queries |
| بتسمح للمهاجم يعدّل أو يسرّب أو يمسح بيانات | فلترة البيانات وتقليل الصلاحيات |