

سنبدأ معكم رحلتنا في القسم العملي لمادة قواعد المعطيات 2 بعد أن أنهينا رحلتنا مع مادة قواعد المعطيات 1.

بداية بالنسبة لل *tool* التي سنستخدمها هذا الفصل فيصبح ب *SQL server* وذلك لأن هذا ال *engine* يعتبر ممتاز بالنسبة لل *Performance* من حيث التعامل مع *Data* ضخمة جدا.

حيث أن أحد أهداف المقرر هو تحسين آلية ال *optimizer* وتسريع تنفيذ ال *Queries*.

فإن المعالجة الصحيحة لل *Data* يفترض أن تكون في ال *Data base layer* وهذه هي الطريقة الأصح لل *processing*.

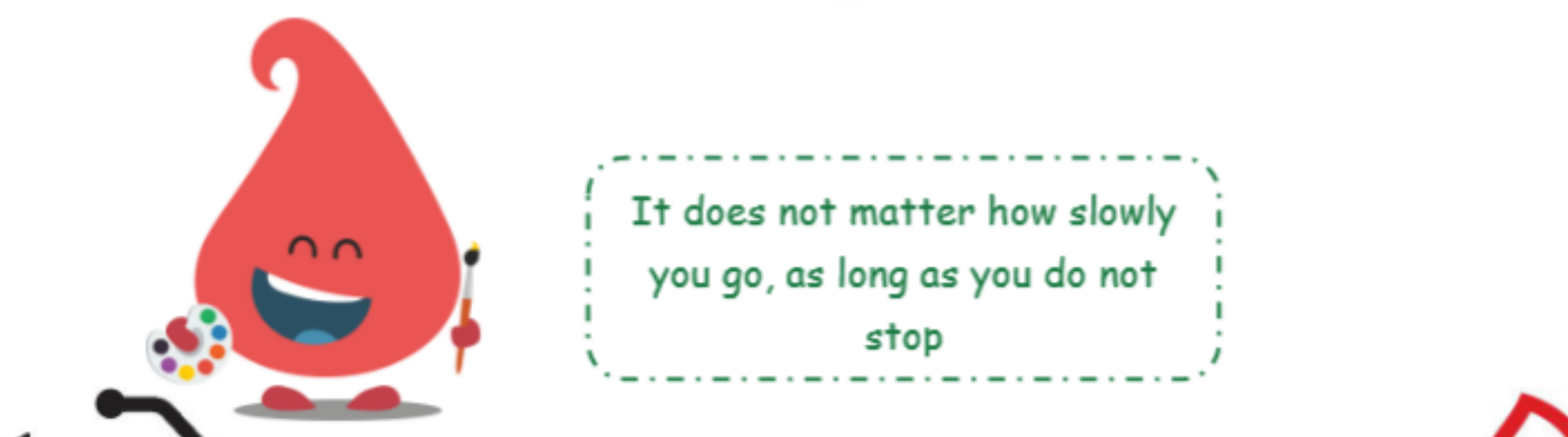
الآن بالنسبة ل *SQL server engine* يوجد هناك حالتين كما تعلمنا وهما:

1- *Window Authentication*

2- *SQL server Authentication* : وفي هذه الحالة يجب إدخال *username* و *password*.

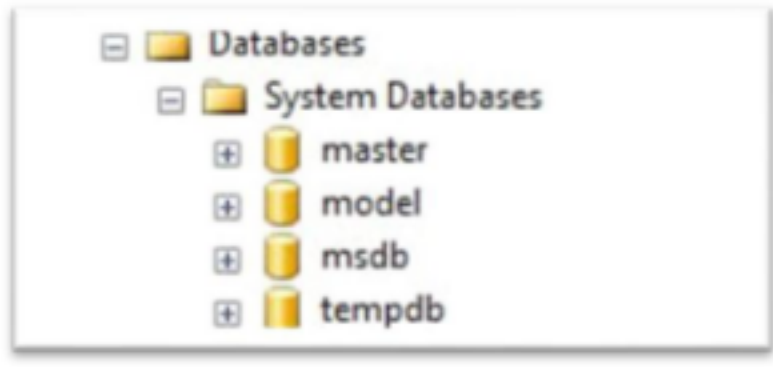
سنفكر الآن عن طريقة التخزين وتركيب الملفات لل *Data base* في *SQL server* وهذا ما نتحدثا عنه في محاضرة النظرية.

ملاحظة:



1 قواعد معطيات 2 | م. عبد البديع مراد

يوجد 4 *databases* يجب عدم التعديل عليهم لا عتبارهم *dictionary* الخاص ب *sql server* وهم:



- master
- model
- msdb
- tempdb

لذا ينصح ببناء ال *database* الخاصة بنا

وفي *SQL server* فإنه عند إنشاء أي *Data base* محتما سيتم بناء *files* 2 منها وهما *mdf*, *Ldf*.

في ال *mdf* يتم تخزين كل ال *dictionary* الخاصة بال *data base* فيها أي البيانات التي ترفقة لجميع ال tables ار ما يعرف بال *meta data* وفي ال *ndf* عادة يخزن فيها ال *Data* التي ستكون موجودة ضمن هذه ال tables.

أما *ldf* هي *transaction file* يتم فيه تخزين التعليقات *select, insert, update, delete dml data (modification language)* أي أنه لا يقوم بتخزين ال *data* أو *result of query* بل فقط بتخزين التعليمة التي تم اجراءها ويتم فيه تخزين التعليقات *sequentially* والهدف من ذلك بالطبع هو ال *recovery* أو عملية استرجاع ال *data*.

نفرض لدينا *database* وكمنا ب *insert* لأكثر من *row* على *tables* هذه ال *database* إذا لم يكن لدينا *ndf* سيتم تخزين هذه ال *data* في ال *mdf*

تذكرة:

- أنواع التعليقات *queries*:
- 1. *ddl data definition language (create, alter, drop,)*
- 2. *dml data modification language (select, insert, update, delete)*



سؤال:

نحن نعلم أن مبدأ التخزين هو على ال *files* في ال *hard* إذا قمنا بعملية *insert* ل *row* فهل سيتم تخزينه في ال *mdf* مباشرة؟

أولاً، فإن هذه التعليمة تخزن في ال *log* بصيغة معينة يفهمها ال *Engine* لنفرض أننا قمنا بمد ذلك مباشرة بعمل *select* لهذه ال *data* التي تم إدخالها، فهل سيقيم يد ال data لم لا؟

نعم، سيتم جلب ال *data* ولكن من ال *memory*.

ملاحظة: سرعة الأداء (*performance*) تكون ممتازة عندما تكون عملية القراءة من ال *memory*.

2 قواعد معطيات 2 | م. عبد البديع مراد

الآن لو قمنا بعمل *shut down* لل *engine* فهل سيتم حفظ هذه ال *data* ونقلها إلى *mdf*؟

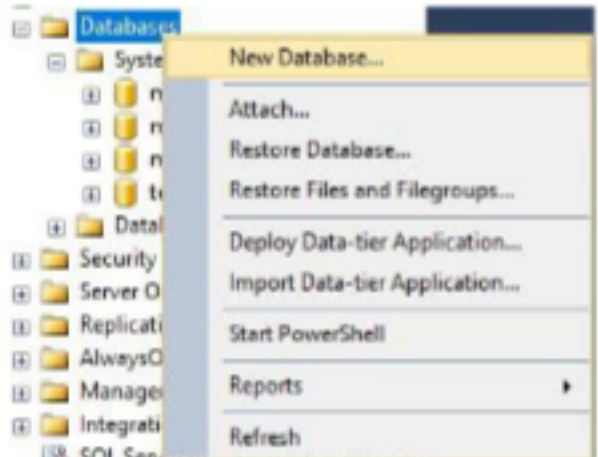
ان في *SQL server* يوجد *2 processors* :

- DB writer** : مسؤول عن تفرغ ال *memory* ونقل ال *data* الموجود فيها لل *hard*.
- DB Reader** : مسؤول عن قراءة ال *data* ووضعها في ال *memory*.

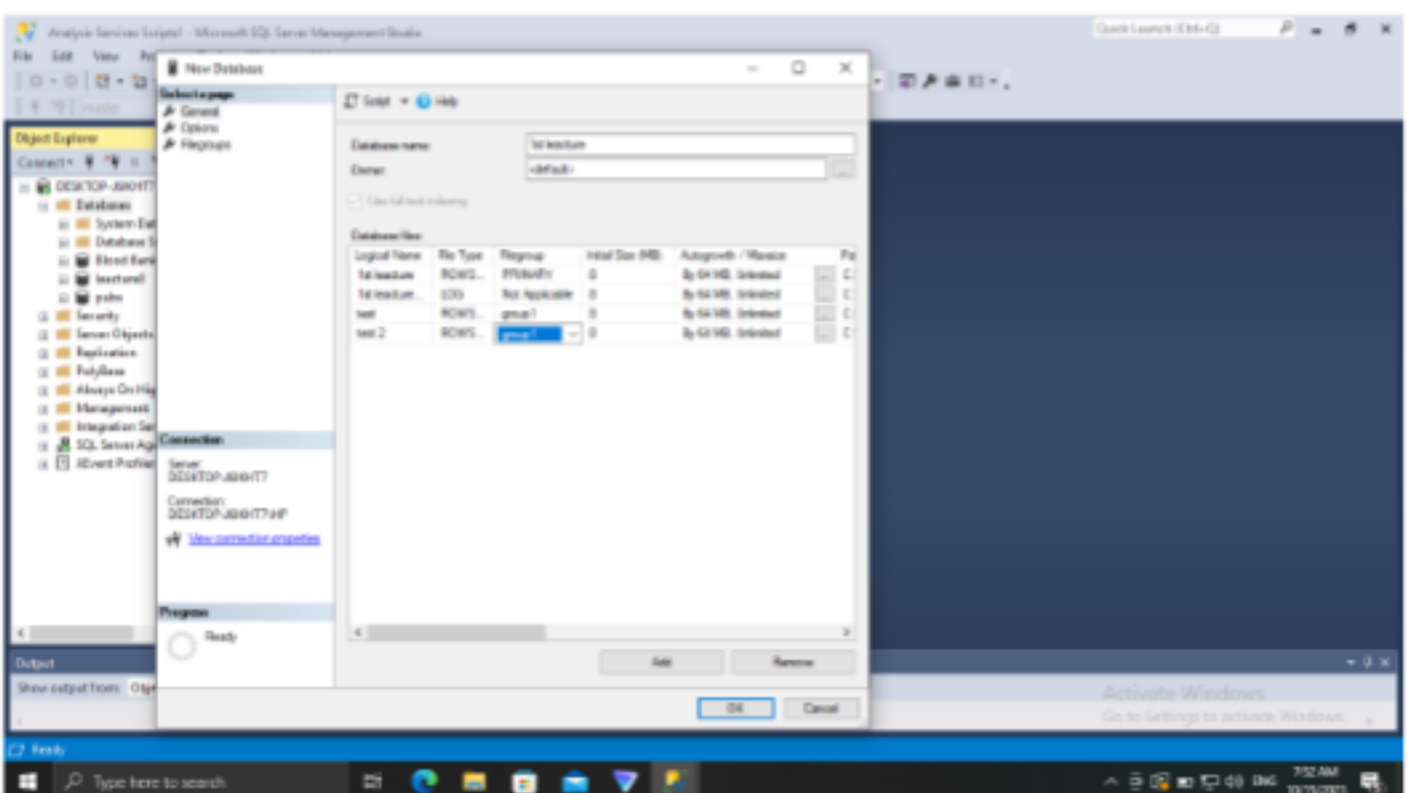
ففي حالتنا السابقة سيقيم بتشغيل *DB writer* وفحص فيما إذا كان هناك *data* موجودة في ال *memory* ولم يتم كتابتها على ال *database* فإلزم بكتابتها على ال *data base*.

ولكن في حال لم يتم بعمل *shut down* بشكل نظامي وعلى فرض حدث قطع مفاجئ للتيار الكهربائي، في هذه الحالة نحن أصبحنا بحاجة إلى *Recovery* وأي أن هناك *transaction* تم كتابتها على ال *log* ولكن أقرأها فملأها لم يتمكن على *mdf* و *ndf* فيهم ال *engine* أنه الآن في حالة *recovery* أي أنه يقوم بتحميل التعليقات الموجودة على ال *log* ويقوم بتنفيذها على الملفات الخاصة بال *data base*.

من الأفضل أن يكون حجم ال *data* ثابت نحدد وأن يكون لدينا *administrator* يراقب تغير حجم ال *data* في حال كنا بحاجة إلى *Hard memories* جديدة أو *raid* (سنأخذ شرح مفصل عنه في محاضرة النظرية)

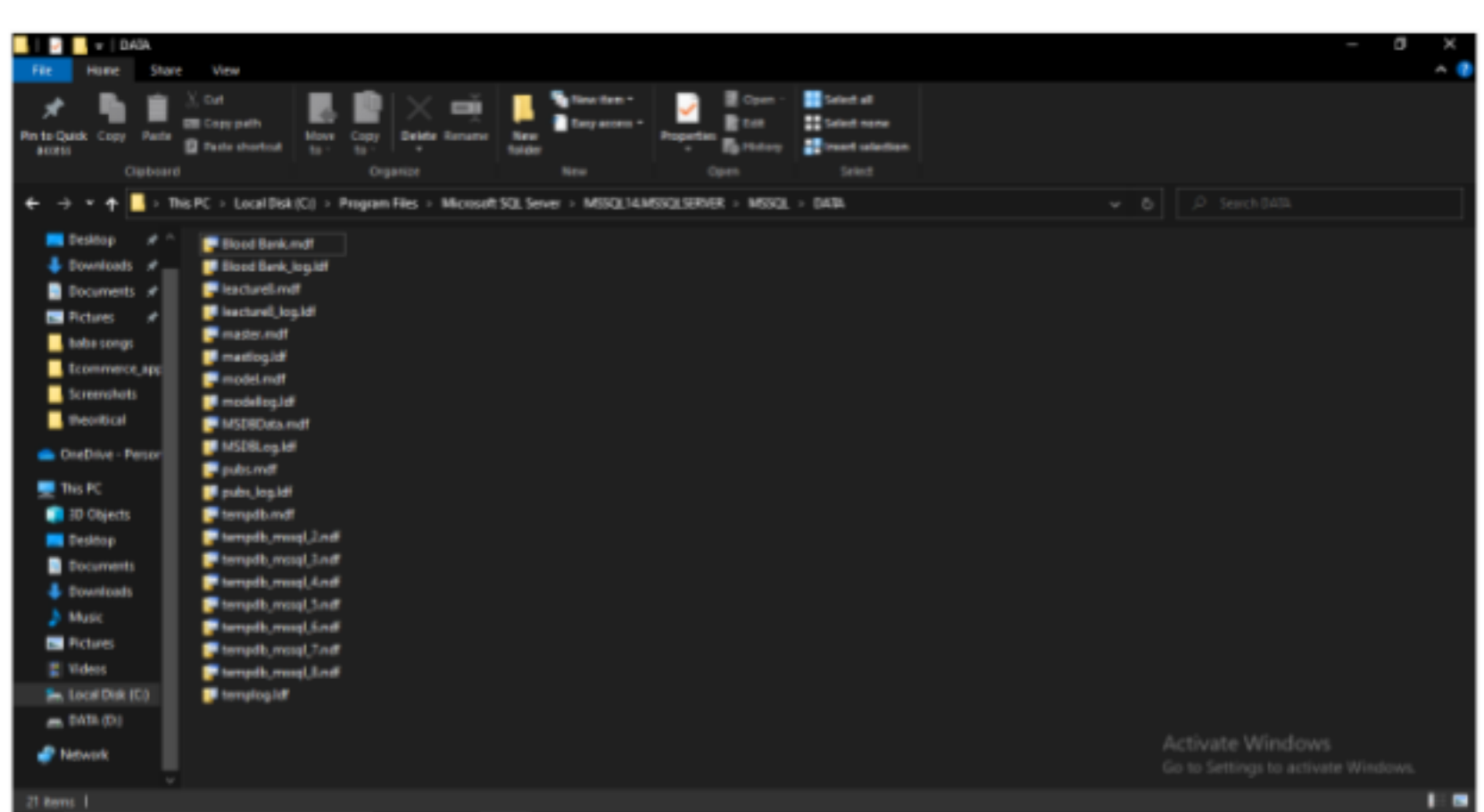


والآن نذهب للتطبيق العملي على *SQL server* فلر قمنا بعمل *New Data base* نلاحظ ما يلي:



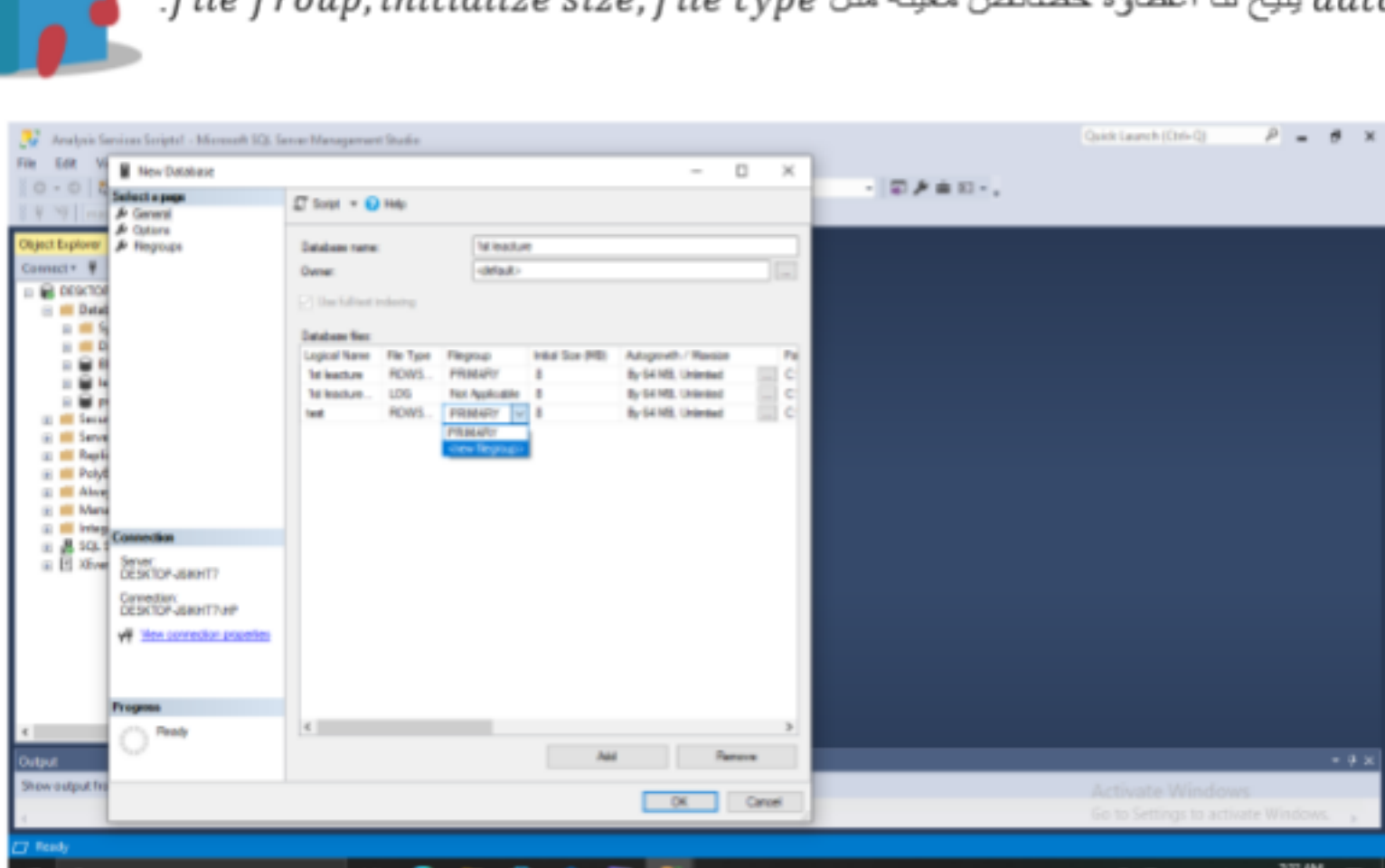
3 قواعد معطيات 2 | م. عبد البديع مراد

لو ذهبنا إلى هذا المسار الخاص بال *data base* التي انشأناها نشاهد ما يلي:

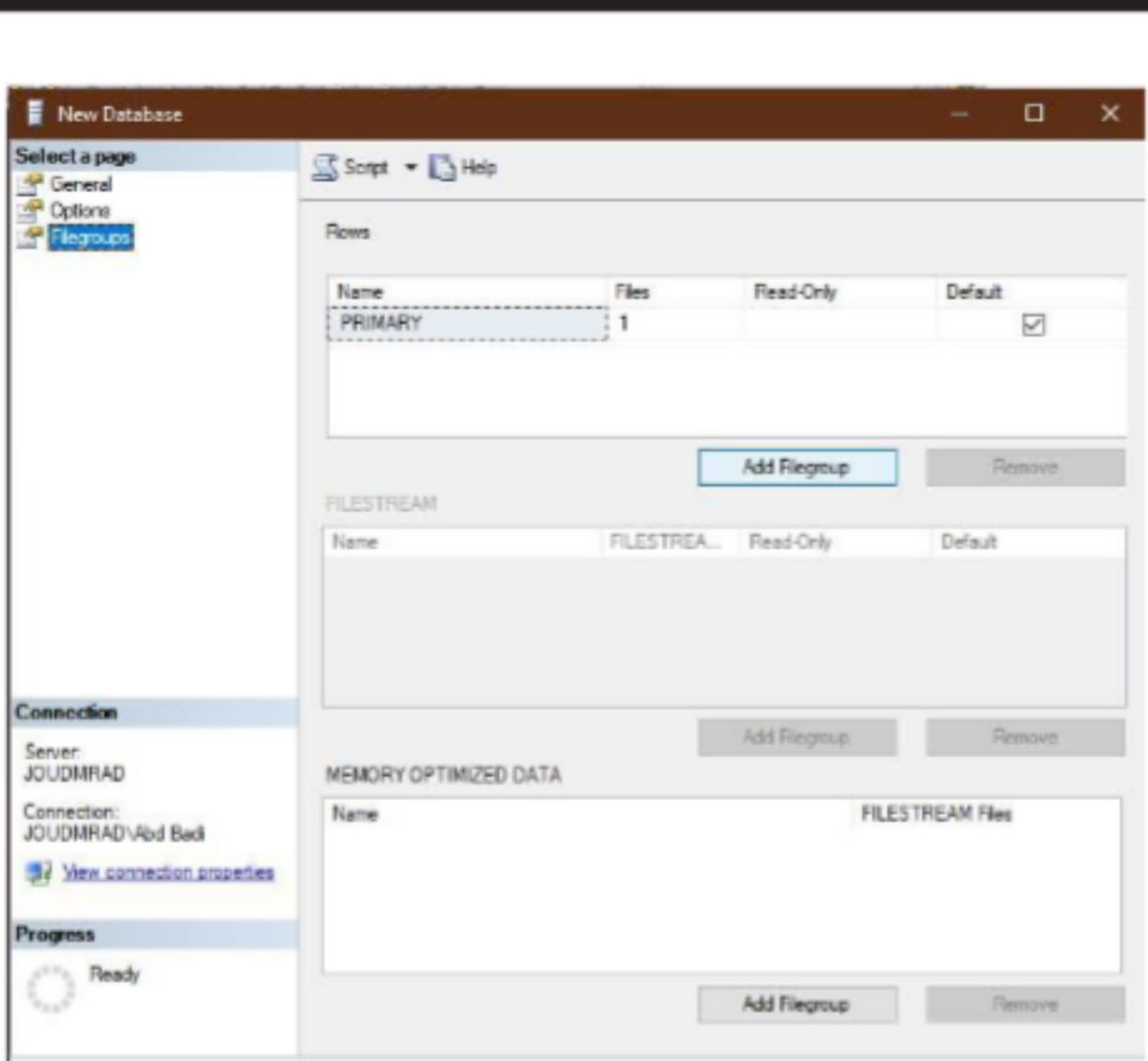


ونلاحظ وجود ملفات بالحقبة *mdf, ldf, ndf* لهذه ال *data base* وكل ال *data bases* الأخرى الموجودة في هذا المسار.

الآن نمود للنافذة الخاصة بإنشاء ال *data base* ونضغط على *add* فيتم إنشاء *file* جديد على *data base* نتيج لنا اعطاه خصائص معينة مثل *file type, initialize size, file froup*.



4 قواعد معطيات 2 | م. عبد البديع مراد

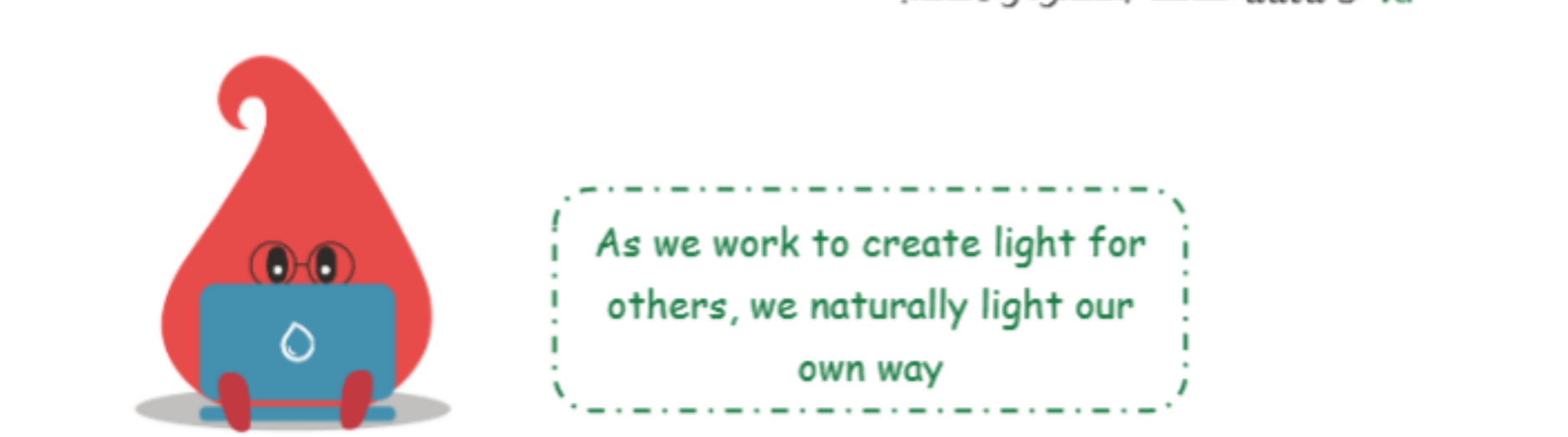


ولكن من هو ال *file group*؟ هو ببساطة لتنظيم العمل في ال *data base* عند التعامل مع عدة ملفات وذلك عبر تخزين كل مجموعة ملفات ضمن *file group* معين.

مثال:

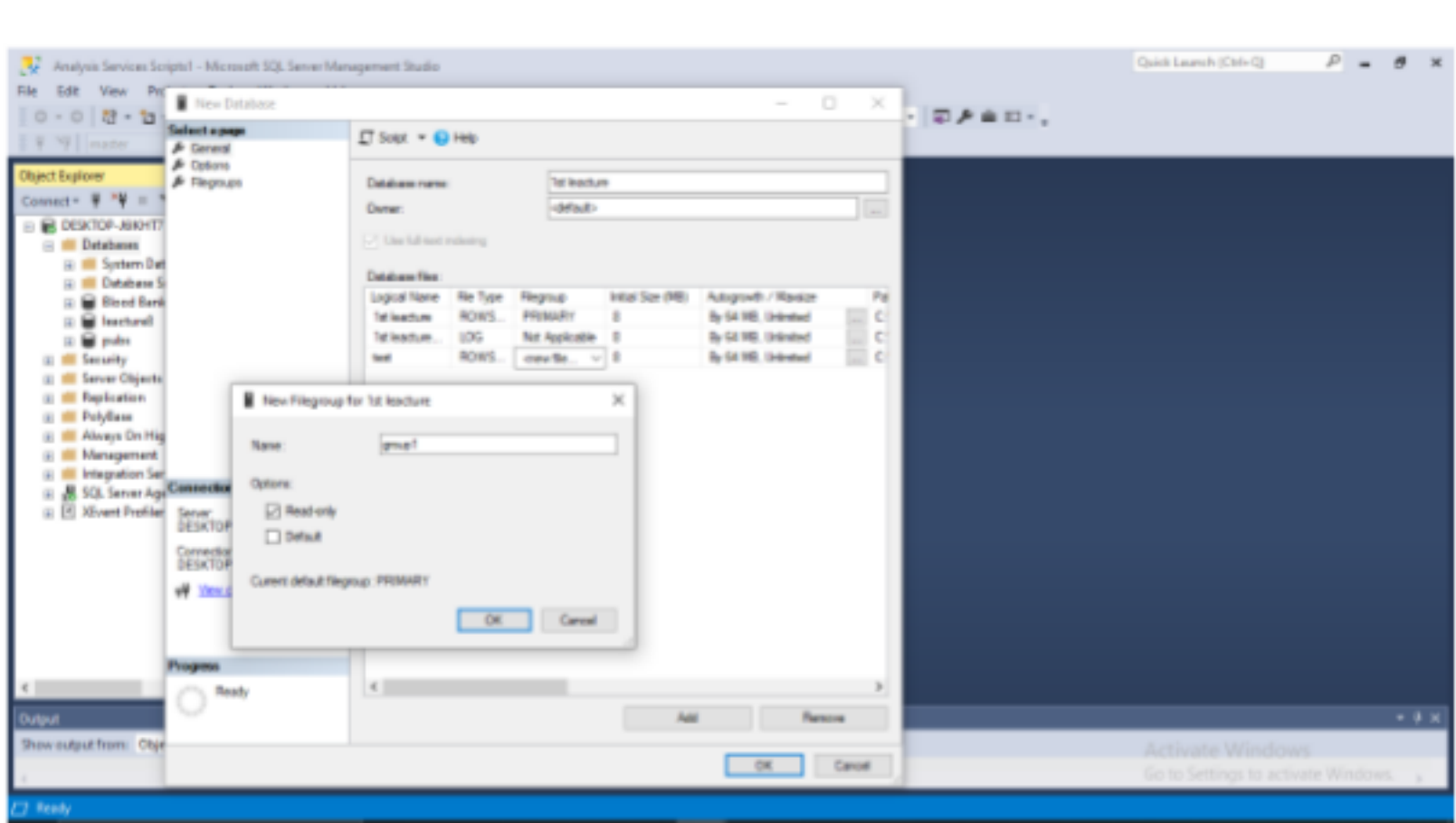
لو أردنا القيام بعملية أتمتة في كلية الهندسة المعلوماتية فلنذا:

- a. مديرية شؤون الطلاب: لها *data* خاصة بها ولها *tables* خاصة بها.
- b. مديرية الامتحانات: لها *data* خاصة بها ولها *tables* خاصة بها.
- c. ال *data* الخاصة بالديوان.
- d. ال *data* الخاصة بالذكورة والأساتذة.



5 قواعد معطيات 2 | م. عبد البديع مراد

وفي هذه الحالة لتنظيم العمل يتم وضع كل مديرية وال *data* الخاصة بهما ضمن *file group* مع ال عمل ان هذه المديرية والاقسام يمكنهم التواصل فيما بينهم وما إلى ذلك وإنشاء *file group* وإضافة ملف عليه:



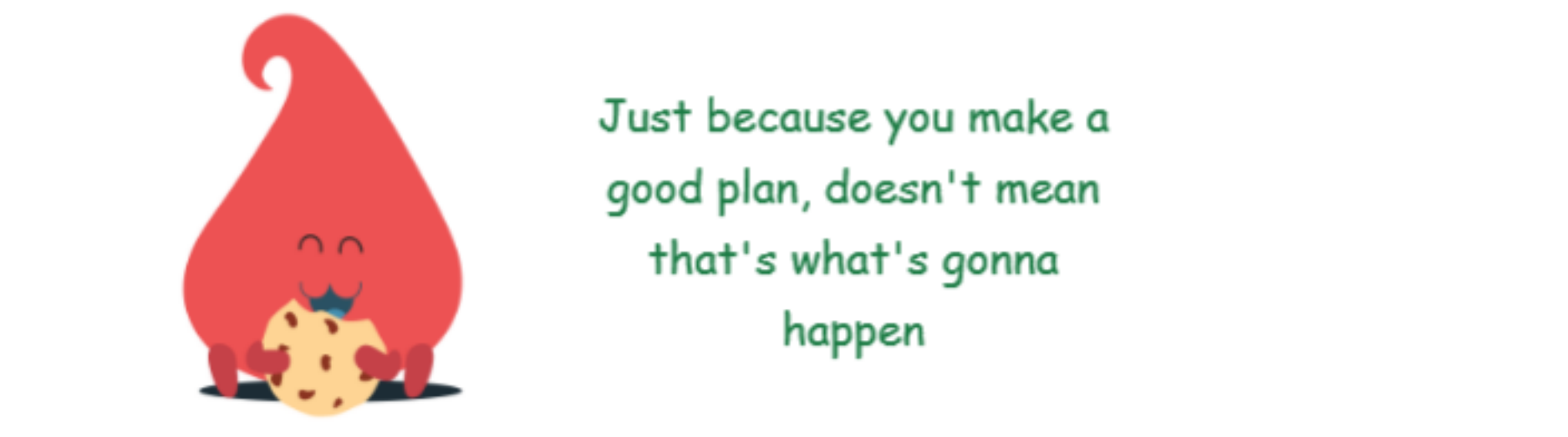
ملاحظة:

أيهما أفضل؟؟؟

ملف حجمه 1 تيرا نقوم بالكتابة عليه والقراءة منه أم عدة ملفات حجمها 1 تيرا؟

بالتأكيد عدة ملفات حجمها 1 تيرا هي الأفضل من ناحية الملفات

THE END



6 قواعد معطيات 2 | م. عبد البديع مراد