

Day 6

- كل database Engine لديه طريقة فى الـ Creation والـ saving بتاع الـ data
- فى Microsoft الـ Database بتتخزن على شكلين :

1) Physical Representation :

- ودا عبارة عن الـ physical Database Files على الـ hard disk
- الـ Database بتتـ create على Two Files

1- file . ldf :

- ودا بيكون الـ log file اللي بيضم الـ transactions

2- file . mdf :

- ودا بيكون الـ Data file اللي بيضم الـ Data + meta data + relations

2) Logical Representation :

- دا عبارة عن الـ logical Files على الـ SQL
- بيتخزن الـ Creation فى الـ File group
- الـ Primary File Group دا عبارة عن pointer بيشار على الـ mdf file
- أى New table بيتخزن فى الـ Primary File group ومنه فى الـ mdf (بيتتم Automatic)

-
- Microsoft حبت تخلى الـ Data تتخزن على كذا file مش الـ mdf file بس (عشان لو حصل Bad sector على الـ mdf file الـ data كلها هتضيع) فقرروا يوزعوا الـ data على الـ mdf file ودا يعتبر الـ Master file وكذا Secondary file ودول بيبقى عبارة عن ndf file
 - طبعا فى المقابل الـ SQL عمل برضو كذا File group غير الـ primary file group اللي هو الـ default file group
 - كل file group بيشار على one or more physical file
-

* فكرة الـ file group فى تحسين الـ performance :

- دلوقتى لو عندى two tables بعمل بينهم joins كتير فبالتالى أخزنهم فى physical file واحد ولا على two different physical files ؟؟

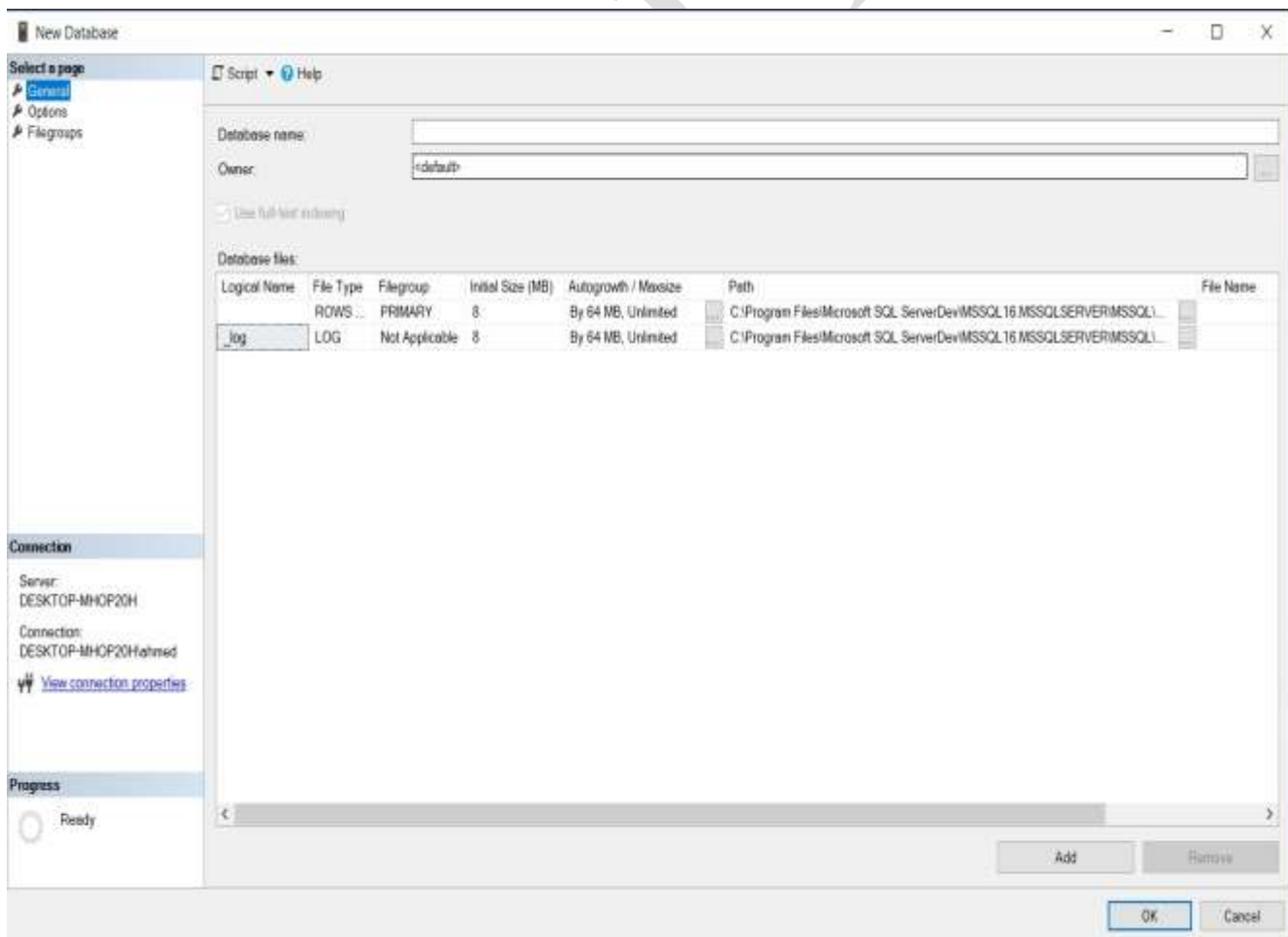
الاجابة : طبعا على two different physical files ودا لأن بيتم قراية الـ Data فى الـ hard disk بطريقة Sequential فبالتالى لو الـ data على two different physical files فبالتالى هقدر أقرأ الـ Data بطريقة Parallel وبالتالي ه يكون أسرع من إنهم يكونوا فى نفس الـ hard disk

- برضو الـ file group بتفيد فى موضوع الـ backup ودا عن طريق أن فيه نوع من الـ backup اسمه الـ file group backup ودا بيكون ان أنا عدلت فى table و موجود فى الـ file group 2 فبالتالى أنا هعمل backup لـ file group 2 بس مش كل الـ files (وحتى لو هعمل backup على كل الـ files فهو أسرع برضو من إن تكون الـ data على one file لأن الـ backup ه يكون parallel مش sequential)

* SSMS (SQL Server Management Studio) :

- فى حالة عاوز أعمل الـ database على كذا file group وكذا physical files

هتظهر النافذة دى Database folder right click → New database →



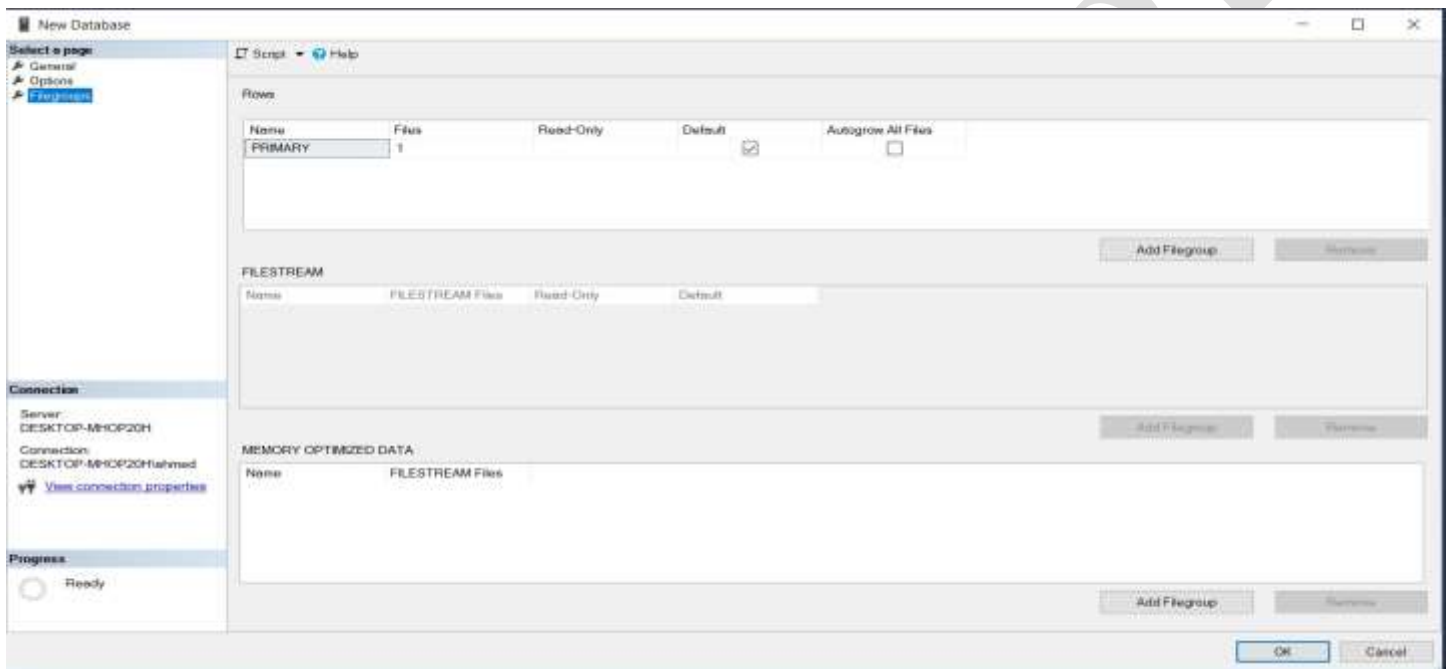
- نیجی فی الـ Database files هتلاقى الـ Two main files والـ properties بتاعتهم

- حتى الآن انت معاك الـ mdf file والـ ldf file

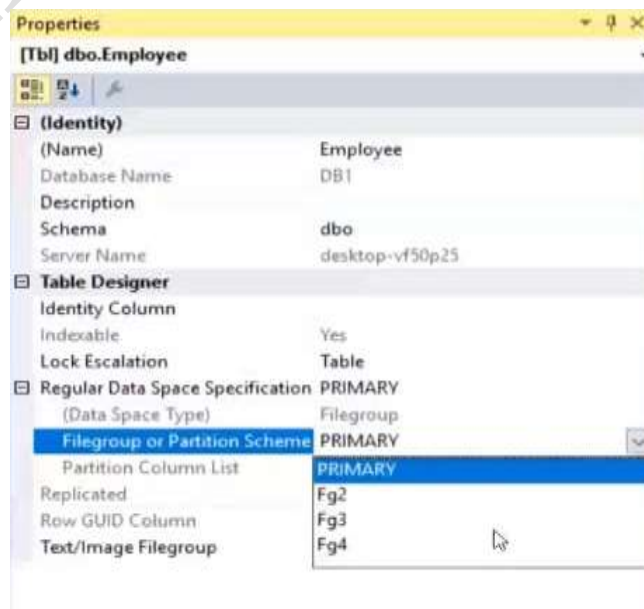
- عاوز أضيف files من الـ Add هيضيف files وطبعاً هتكون الـ ndf file (ROWS File type)

- أحدد الـ properties بتاعتهم (الـ name والـ initial size والـ max والـ auto growth والـ path " طبعاً لو على السيرفير هيبقى فيه كذا "hard disk)

- الـ File group property دى هتلاقى ان كل الـ files واخدين الـ primary file group وعشان أقدر أغير دا لازم أضيف File groups أنا من عندى ودا عن طريق :



- بعد ما أضيف الـ File groups هتلاقى الـ drop box فى الـ file group property
- بحدد من الـ table properties ان الـ Table يتخزن تبع أنهُو file group فى الـ table design



* ال column properties :

- ال identity : ودى بتحدد إذا كان ال column دا إنت عاوزه identity ولا لا . لو أختارت Yes هيطهر Two properties وهما

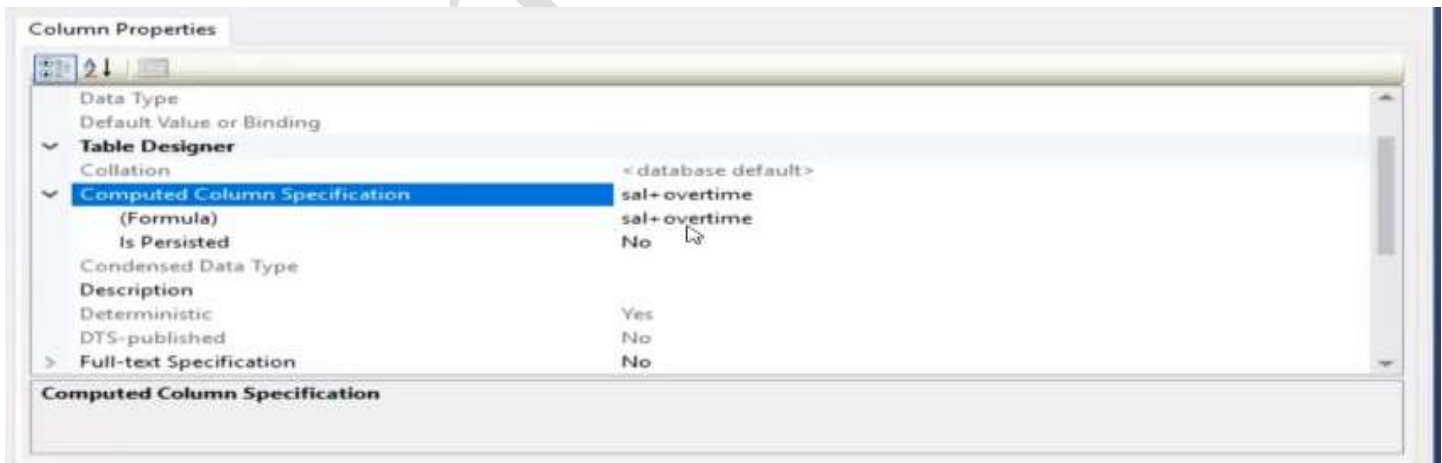
(1) ال Identity seed : ودى ال value اللى هيبدا منها ال Identity

(2) ال identity increment : ودى مقدار الزيادة



- لو عندى Driven attribute (كنت مش بعمله فى ال mapping) ، فى ال SQL server بحط ال driven attribute فى ال Table عادى ولكن باجى فى ال column property الخاصة بيه

- عندى مثلا ال Netsal عبارة عن Sal + over



فهاجى فى ال computed column specification وأحط فى ال formula المعادلة اللى بتحسب ال driven attribute

- خاصية ال is persisted بتحدد إذا كان ال driven column هيت save على ال hard disk ولا لا

(نحتاج أعمله save لو أنا بستخدم الـ column فى حسابات تانية)

الـ Is Sparse property :

- دى الـ property بتسمح إنك تحجز أو متحجز space for Null values

- بستخدمها مع الـ columns اللى متوقع قيم الـ Null فيها كتير

- الـ default إن الـ null يتحجزله space

Is Sparse → yes → prevent space for NULL

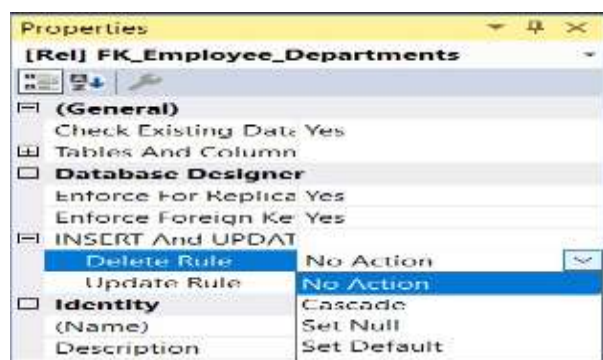
Is Sparse → No (default) → allow Space for NULL

* الـ Relationship Properties :

- عندى Two Properties محتاج أعرفهم وهما الـ Delete Rule و الـ Update Rule

- قبل كدا كان عندى قاعدة بتقول (إن مينفعش أعمل Delete \ Update لـ parent وهو عنده child)

- أقدر أكسر القاعدة دى عن طريق الـ Two properties



- لكل property منهم 4 إختيارات وهم :

1) No Action :

- دى بتمنع إنك تـ Delete \ Update لـ Parent ليه Childs (القاعدة الأساسية)

2) Cascade :

- اللى بيحصل على الـ Parent من Delete \ Update يحصل على كل الـ Childs ليه

3) Set Null :

- فى حالة إنك عملت Delete \ Update على Parent الـ Childs

الـ Update (لوعملت Update لـ PK هتتحول الـ Fk فى الـ Child لـ NULL)

ال Delete (لو عملت Delete لـ Parent هيتم وضع NULL فى ال Child FK)

4) Set Default :

- فى حالة إنك عملت Update Parent PK ال Child Fk هياخد قيمة ال Default (لازم قيمة ال Default تكون واحدة من قيم ال Parent Pk لأنه Foreign Key)
- فى حالة إنك عملت Delete Parent ال Child Fk هياخد قيمة ال Default

- فى حالة ال Weak Entity بختار ال Delete \ Update عبارة عن Cascade

* Schema :

- قبل ما تظهر ال Schema كان فيه 3 مشاكل وهم
- (1) مفيش Two Objects فى نفس ال database بنفس الاسم
- (2) لو فيه 200 Table وأنا محتاج أستخدام 5 منهم بشكل معين فمش هقدر أعمل logical Grouping
- (3) ال SQL User Permissions بحيث لو فيه User ليه Permission على 3 table وهما نفس ال permission
- Microsoft قدرت تحل المشاكل دى بإستخدام ال Schema

1) dbo.Student (default Schema)

Hr.Student (new Schema) Both are different

3) User → Schema Permissions

* Create Schema Schema_name

- معنى ذلك هتيجى فى ال

Database Folder → Security Folder → Schema Folder → Schema_name

* إضافة Table أو أى Object جوا الـ Hr Schema

- **Alter Schema** Schema_name **transfer** Object_name

- الـ schema مجرد تسمية لـ group of objects

- مش بتأثر على relationships أو غيرها

لو عاوز أحدد الـ schema فى الـ creation هكتب اسم الـ schema مع إسم الـ object

Create Table Schema_name.Table_name

(
)

* 10 steps For User\ Schema :

(1) أخلى الـ authentication بتاعت الـ server عبارة عن Mixed ودا عن طريق :

Server Right click → Properties → Security tap → mixed

(2) أعمل Restart لـ Server عشان يطبق التعديل :

Server right click → Restart

(3) أعمل New Login :

Server Security Folder → Logins Folder Right Click → New Login

(4) أخلى الـ login عبارة عن DB User

DB Folder → Security Folder → Users Folder Right click → New User

(5) نعمل الـ Schema :

Create Schema Schema_name

(6) نضيف الـ objects :

Alter Schema Schema_name **transfer** Object_name

(7) نـ Connect بالـ Admin على الـ server :

Schema Folder → Schema name Double Click → Permissions Tap

- (8) نحدد الـ permissions للـ user على الـ Schema :
من خلال Grant \ Deny \ with Grant في الـ panel
(الـ Default يكون Deny عشان الـ user لما بيتتم إنشاؤه بيبقى بلا أى Permission)
- (9) أعمل Disconnect وأدخل بالـ User
- (10) أقف على الـ DB وأعمل new Query وأتأكد إن إسم الـ User
اللى ظاهر فى الـ New Query tap

* Synonym :

- هو عبارة عن pointer بيكون ليه إسم
- بيشار على Object معين
- أقدر أستخدمه فى حالة إن مستخدم Table فى كذا Function وعدلت إسم الـ table فبالتالى هضطر ألفت على كل الـ Functions وأعدل ، بس فى الـ synonym أقدر أعدل الإسم فى الـ Synonym بس وبالتالي هيسمع فى مكان عادى بالإسم الجديد (لأن إسم الـ Synonym مازال نفسه لكن الـ Table هو اللى غير اسمه وأنا مستخدم الـ synonym Name)

Create Synonym Synonym_name **for** Schema_name.Table_name

Alter Synonym Synonym_name **for** Schema_New_name.Table_New_name

Drop Table Table_Name

Delete From Table Table_Name

Truncate Table Table_Name

- الـ Drop : بتمسح الـ data والـ structure
- الـ Delete : بتمسح الـ data
- الـ Truncate : بتمسح الـ data

Delete Vs Truncate :

Delete	Truncate
- تقدر تمسح جزء من الـ Rows لأن بيجي معاها Where	- مش بيجي معاها Where
- أبطأ (عشان دايماً بتتسجل فى الـ log file وتتنفذ)	- أسرع (عشان مش دايماً بتتسجل فى الـ log file)
- أقدر أعمل roll back	- مقدرش أعمل roll back
- مش بتعمل Reset for Identity بعد ما تمسح	- بتعمل reset for Identity بعد ما تمسح

* إزاي أقدر أمسح Column data من غير ما أمسح الـ structure ؟

- الإجابة : هـ update الـ column بـ NULL

- الـ Database عبارة عن مجموعة من الـ Tables محاطة بـ Constraints

- مفهوم الـ Database Integrity بيوضح مدى تكامل وترابط الـ Data

- لازم الـ Rules تكون محققة الـ Business

- الـ DB Integrity بتتقسم لـ 3 Types of Integrity وهم :

1) Domain Integrity :

- ودا بيكون على الـ Column Level

- بيتأكد من توافق الـ Range of values

2) Entity Integrity :

- ودا بيكون على الـ Row Level

- بيحقق الـ Uniqueness of Rows

3) Referential Integrity :

- بيكون على الـ Relationships level

- بيحقق توافق الـ Relationships وإنها سليمة وصحيحة

- تحقيق الـ 3 Integrities سيكون على Two areas

1) DB Constraints

2) DB Object

* DB Constraints :

1) Domain Integrity :

- الهدف إن أحقق الـ Range of values في الـ Column
- دا سيكون عن طريق :

1) الـ Datatype : يعرف منه نوعية الـ data و الـ range of value المسموح

2) الـ Default Value : ممكن أقدر أحدد نوعية الـ data من خلال الـ Default Value

3) Allow NULL or Not Null

4) Check Constraints : ودى عبارة عن الـ Column Conditions في الـ Table Creation

2) Entity Integrity :

- عشان أضمن الـ Uniqueness في الـ Rows
- سيكون عن طريق :

1) PK Constraints :

إن أعمل Column عبارة عن Primary Key لـ Table وبالتالي الـ Rows هتبقى Unique
وكمـان Not NULL

2) Unique Constraints :

إن أعمل الـ Column عبارة عن Unique Column وبالتالي الـ Rows هتبقى Unique
- بيسمح بالـ Null Value بس مرة واحدة بس (Unique NULL)

3) Referential Integrity :

- بتأكد من سلامة الـ Relationships

- سيكون عن طريق :

(1) الـ FK Constraints :

إن أعمل Foreign key للـ Relationship فى الـ child Table وأتأكد إن

(مسموح بالـ NULL | مفيش Uniqueness فى الـ FK value | يكون بيـ Match PK Values)

* DB Objects :

1) Domain Integrity :

- سيكون عن طريق :

Rules (1

Triggers (2

2) Entity Integrity :

- سيكون عن طريق :

Index (1

Triggers (2

3) Referential Integrity :

- سيكون عن طريق :

(1) الـ Triggers

* DB Constraints 5 Types :

- 1) Check Constraint
 - 2) PK Constraints
 - 3) Unique Constraints
 - 4) FK Constraints
 - 5) Custom Constraint (logical code via SP , Triggers)
-

- طريقة كتابة الـ Constraints فى الـ Creation
(باجى فى آخر الـ Table واكتب الـ Syntax)

Constraint Const_name Const_Keyword

For Example :

Constraint C1 Primary Key (Col_Name)

- لو عاوز أعمل Composite Key هحط إسم الـ Columns مش Column واحد

Constraint C2 Unique (Col_Name)

Constraint C3 Unique (Col_Name)

Constraint C4 Unique (Col_Name1 , Col_Name2) **XXXXXXXXXXXXXX**

- مينفعش تعمل unique Constraints واحدة لـ multiple Columns ودا لأن كدا هيتكرر عادى فى الـ Column الواحد ولكن الأثنين مع بعض مش هيتكروا (نفس فكرة الـ composite Key)

Constraint C5 Check (Column \ Condition)

Constraint C5 Check (Salary > 1000)

- هيت Check على الـ Salary Values إنها تكون أكبر من 1000

Constraint C6 Foreign Key (Column_Name) **references** Table(Column_Name)

On delete Set NULL on Update Set Cascade

- فى الـ constraint دا هيعمل الـ Foreign key و يبقى عارف بيشاور على الـ PK فى أنهو Table و كمان عمل الـ delete \ update Rules وبالتالي لوجيت أعمل diagram هلاقيه أتعمل من غير أوصل الـ FK بـ PK و كمان هلاقى الـ Rules بتاعت الـ Update والـ Delete

* Constraints After Creation :

- لو عاوز أعمل Constraints على Table موجود فعلاً (مش وأنا بـ create) هستخدم الـ syntax

Alter Table Table_Name Add Constraint Cons_Name Cons_Keyword

Alter Table Table_Name Drop Constraint Cons_Name

(مع مراعاة إن الـ constraint بيطبق على الـ Old \ New Data يعني لو عندى data موجودة ومش بتحقق الـ constraint فى الحالة دى الـ constraint مش هيتـ Create)

لازم أعدل الـ data Manually أو أستخدم الـ rule (هيتـ معرفتها Later)

* Rule :

- لو أنا عاوز أعمل :

Constraint بس تتطبق على الـ New data بس

Constraint بس يكون shared مابين كذا Table

Data type عليه Condition و default value

أقدر أعمل دول بإستخدام الـ Rule

- الـ Rule هى عبارة عن Check Constraints بس الفرق إن الـ constraint بتبقى على table

Level أما الـ Rule بتكون على الـ Schema Level

- مكان الـ rule بيكون فى programmability Folder بلاقى الـ Rule Folder

- الـ Syntax بتاع الـ Rule :

Create Rule Rule_Name As Rule

Create Rule r1 As @X > 1000

- كذا أنا عملت rule بإستخدام variable إسمه X (أى حاجة بعد @ بتعتبر Variable) وبقول إن الـ X لازم يبقى أكبر من 1000

- أقدر أستخدم الـ rule بعد كذا بإن أربطها بأى column أنا عاوزه (بيعمل replace لـ X)

الـ Syntax المستخدم فى الحالة دى

Sp_bindRule Rule_Name , 'Table_Name.Column_Name'

Sp_bindRule r1 , 'Instructor.Salary'

Sp_bindRule r1 , 'Emp.overtime'

كدا أنا بـ bind الـ Rule بـ column (بحيث إنه ملوش دعوة بالـ Old Data المهم لو فيه Data دخلت فى الـ Column من بعد لحظه الـ bind rule هيبدا يعترض عليها لو هيا مش بـ Rule الـ Fit)
 - لو عاوز أعمل drop الـ rule لازم أعمل unbind الأول وبعد كدا الـ Drop

Sp_unbindRule Rule_Name , 'Table_Name.Column_Name'

Drop Rule Rule_Name

Sp_unbindRule r1 , 'Instructor.Salary'

Sp_unbindRule r1 , 'Emp.overtime'

Drop Rule r1

Constraints Vs Rule :

Rule	Constraints
Database Level -	Table Level -
Shared -	Not Shared -
Apply on New Data -	Apply on Old + New Data -
one Rule for Each Column -	Multiple Constraints on one Column -

Default :

- نفس الفكرة مع الـ rule هتيجى مع الـ default

- برضو مكانها بيكون على الـ Database level

- برضو فى الـ programmability Folder

- بيتم إنشاؤها بـ Create وأعملها bind

- لو عاوز أحذفها فبيكون Unbind وبعدين Drop

Create Default Default_Name **As** Default_Value

Sp_bindefault Default_Name , 'Table_Name.Column_Name'

Sp_unbindefault Default_Name , 'Table_Name.Column_Name'

Drop Default Default_Name

- فائدة الـ rule والـ Default هيا إن أعمل User Defined Data type

- هيثم عمل الـ rule و الـ default و هستخدم Stored Procedure فى إن Create user DT وبعد كذا أربط الـ rule و الـ default بالـ User DT
- وعشان كذا الـ rule بتبقى مع one column عشان بتستخدم فى الـ User DT

Create Rule Rule_Name **As** Rule_Constraints

Create Default Default_Name **As** Default_Value

Sp_addtype User_DT_Name , SQL_DT

Sp_bindrule Rule_Name, User_DT_Name

Sp_bindefault Default_Name, User_DT_Name

Create Rule r1 **As** @X > 1000

Create Default d1 **As** 5000

Sp_addtype ComplexDT , 'int'

Sp_bindrule r1 , ComplexDT

Sp_bindefault d1 , ComplexDT

- وبكدا أقدار أستخدم الـ Datatype فى الـ Table Creation

- الـ Output هيكون Datatype من النوع Int ولكن عليه Rule إنه يكون أكبر من 1000 والـ default value تكون بـ 5000

- لو فيه Column عليه Constraints و Rule بيتم تطبيق الـ constraint الأول ثم الـ Rule

Constraints → Rule
