

DML → View From Multi Tables

Alter View V-Join (Sid, Sname, did, dname)
with encryption

45

Select s.t-id, s.t-name, d.dept-id, dept-name
From Student s join department d
on d.dept-id = s.dept-id;

1 Delete XXXXXXX

Insert - update VVVV

Insert into V-join

Values (21, 'nada', 700, 'cloud'); *error*

- table data Insert into اوصاف

لديكم يحضرن الـ **allows** الـ **Table** الامان مشاهدة قسر اعمل **Insert** في الـ **Table**.

Indexed View

Create view V-data
with schemabinding
as

Select ins-name, salary
From dbo.Instructor
Where dept-id = 10;

- Alter table instructor alter column ins-degree Varchar(50); vvvv
- Alter table instructor alter column ins-name Varchar(100); XXXXX

٢) من ٣ مهام لازم أكتب فيهم اسم الـ Schema ١) حذف احوال Table ٢) حذف لوكلات data Select ٣) إيجاده لا Indexed View

: Table Schema مع الـ اسم فيها اكتب مضطراً بقى الى 3 حالات الـ هي الـ 3

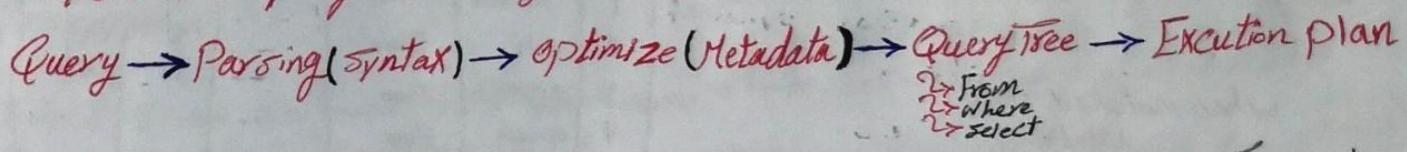
- 1 لما اجي اعمل Indexed View
 - 2 لما اجي اعمل Call Scalar function
 - 3 لما اجي اجيب Data من database تانينه

Day 9
5/4/2023

Stored Procedures

SQL: Day9
Stored Procedures
The Cycle of the query in the Engine

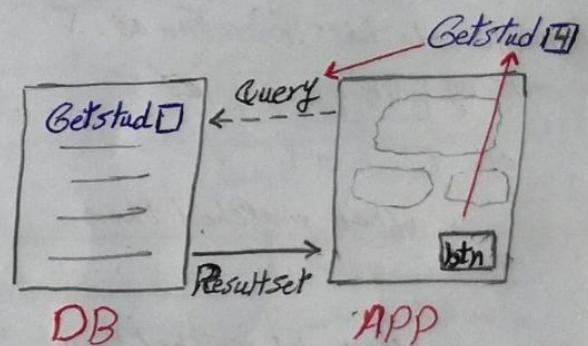
* The Cycle of the query in the Engine:-



- الـ user يكتب الـ query يقوم الـ engine بـ query يتحقق أول حاجة الـ engine مطابقاً لـ query.
- بعد ما يتم تدقيق الـ Syntax يدخل query في الـ optimize حيث يتم تفاصيل objects وأسماء الـ tables من الـ DB.
- بعد تدقيق query يتحقق query tree حيث يتم تفاصيل query tree مع جدول student الذي يحتوي على عمليات select.
- في آخر مرحلة وهي حالة التغيير يتم تغيير ترتيب العمليات الموجودة في الـ query.
- output query وطبع الـ query.

+ كل ما في query ما يكتب كل ما بعد الحرف جداً ما يعبر عن ماداً أثير على الـ Network Traffic

- Create procedure Getstud @Id int
as
Select *
From Student
Where Id = @Id ;



- Execute Getstud 4;

+ مميزات استخدام الـ stored procedure

- ترفع من الـ security للأقسام بالـ DB لـ query لما يجيء داخل Call بـ query على الـ DB.
- معلومات تخص الـ data وـ delete وـ insert وـ select وـ update وـ insert وـ delete.
- كما أنّ قدرة سببها كلّ نوع الـ queries يمكنها جواه الـ View وـ Function.
- عدد الـ Network Characters

- عند دليل الـ SP الـ stored procedure لـ query يقوم الـ engine بـ query ثم يقوم بـ execute الـ procedure.
- يدخل دليل الـ SP عند دليل procedure Call.
- تتم معالجة خطأ try catch error.
- ويمكن تعيين الـ SP من الـ APP.

+ مميزات استخدام الـ stored procedure:

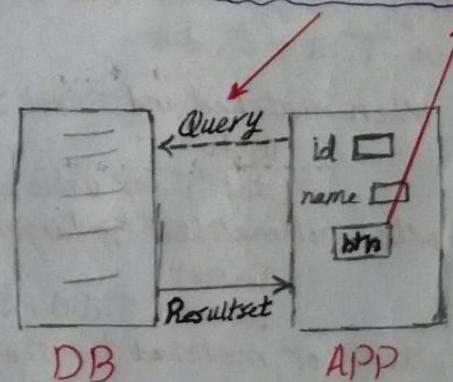
- هنا أنا أتعامل مع أسماء الـ tables وـ data بدلاً من الـ DB.
- أقسام بالـ DB.

ـ هنا عدد الـ Network Traffic أقل مما يحصل مع الـ characters.

ـ هنا لـ razem هو query يدخل الـ query cycle وهذا ينعكس على الـ performance.

ـ عند حدوث أيّ error هنا ينحصر في الـ APP أو الـ Website.

insert into student (id, name)
values (7, 'Ali)

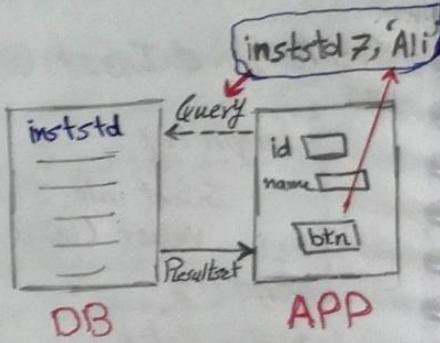


Create Proc inststd @id int, @name Varchar(20)
as

- If not exists (Select id From Student where id = @id)
 Insert into Student (id, name)
 Values (@id, @name)
else
 Select 'Duplicate ID';

- Executive inststd 7, 'Ali';

هنا أنا قدرت أمنع حدوث الـ IF exception error
دلي قررت أدخلها مبرمجاً في Stored procedures



في SQL من اختناقال business rules يعني طرق معينة في حساب البيانات ويعمل على ذلك APP developer كتب لها query أو زوايا أو أخرى لـ logic إنما يوجدون في SP كمثلك اختناقال logic بأخطاء وعمر طريقها أقل بـ 4-5 خطوط فقط كـ SP التي تبرهن حساب البيانات عن طريق إنشاء id بناءً على الموقف ببساطة وصيغة حساب بسهولة.

Types of Stored Procedures

1] Built-in SP

SP_bindrule
SP_unbindrule

SP_rename
SP_addtype

SP_helpconstraint
SP_helptext

2] User Defined SP

3] Trigger

2] User Defined Stored Procedure

Create proc Getst

as
 Select *
 From Student;

Executive Getst;

Create Proc Getstbyadd @add Varchar(20)

as
 Select st-id, st-name, st-address
 From Student
 Where st-address = @add

Executive Getstbyadd 'Alex';

[create Proc Instst @id int, @name varchar(20)]

as begin try

Insert into Student (St-id, St-Fname)
Values (@id, @name)

End try

begin catch

Select 'error'

End catch;

- Executive Instst 44, 'Ali';

هذا SQL قادر على إثارة خطأ
عند حدوث خطأ في حال
إدخال user مدخل يدخل Insert لـ ID
هو موجود بالفعل، سترسل
رسالة خطأ وتحذف message
الخطأ في طريقة catch

بعد SQL كان الخطأ حدوث
error

عند إدخال id موجود بالفعل
وسيتم إثارة خطأ error
تتحقق DB وما يدخل في Security دائمًا

DB

[create proc Sumdata @x int, @y int]

as

Select @x + @y;

-- Sumdata 4, 9 → Calling Parameter by Position.

Output
13

-- Sumdata @y=9, @x=4 → Calling Parameter by Name.

Output
13

[create Proc Sumdata @x int, @y int = 100]

as

Select @x + @y;

-- Sumdata 3 → SP. Parameter has Default Value.

Output
103

[Alter proc Sumdata @x int = 50, @y int = 100]

as

Select @x + @y;

-- Sumdata

Output
150

 → SP. Parameters have Default Values.

[create proc GetStbyage @age1 int, @age2 int]

as

Select St-id, St-Fname

From Student

Where St-age between @age1 and @age2;

هذا SQL يمكنه إثارة خطأ

الطلب إلى أنماط ما بين العبرتين

إذاً نناديه لهم لها.

هذا SQL قادر على إثارة خطأ

يتابع إدخاله و قادر على إثارة خطأ

الكتل SQL أو Table output

وذاك لأن SQL ليس له

[Insert into table4 (St-id, St-Fname)]

execute GetStbyage 23, 28;

الكتل SQL أو Table output

وذاك لأن SQL ليس له

declare @t table (x int, y varchar(20))

Insert into @t

execute GetStbyage 23, 28;

يتم تغيير إدخال Select

داخل Insert

وذاك لأن SQL ليس له

الكتل SQL أو Table output

وذاك لأن SQL ليس له

الكتل SQL أو Table output

```
create proc Getdata @id int,  
                    @X int output,  
                    @Y int output
```

table من output هى الـ **data**
وكلها قيم محددة بعدد العبر
إلى أننا مدحهار **SP**

25 select @ X = stage , @ Y = dept_id
From student
Where st-id = @ id ;

انواع امتیازات هستند
output Variables
Container ۵ مساحت
کمینه ۵ مساحت
JP output
الا output
منلا ۵ بسیر بیعتقد علی
الا output تباخته ۵ ای جو

```
declare @X int, @Y int  
execute Getdata 6, @X output, @Y output  
select @X, @Y;
```

لارزم وانت بتبعها (Variables) بالقيم
اكي هتخرج في الاتخاف output لارزم الـ Variables
 تكون الاوده زى كدا ←
 $= \text{stage}$

- input E_{gill} in one parameter (isotropic) \rightarrow
 - output E_{gill} in two parameter

Create Proc Getdata @X int output,
@Y int output

as
Select @X = st-age, @y = dept-id
From Student
Where st-id = @X ;

```
declare @x int = 6, @y int  
execute Getdata @x output, @y output  
select @x, @y;
```

Classification
② one Parameter

Input output parameter ϵ in line
(@X)

One parameter

output parameter gill
(@y)

- output 5 input 5 @x creating list

*Types of Parameters in Stored procedures :-

Types of Parameters in Stored Procedures :-

- ① Input parameter ② Output parameter ③ Input output parameter ④ Return parameter

-- Create proc getalldata @Col Varchar(20),
@tab Varchar(20)

as execute ('Select' + @col + ' From' + @tab);

--- getalldata 'ins-name', 'instructor';

هناك قدرت Dynamic Query در Function & view إلخ مساعدة خواص objects هناك بعض أمثلة ادوات security مثل الـ DB و MySQL و Oracle كـ MySQL و الـ APP Developer ليزا مثل الـ

3 Trigger

- actions of Listen → Server (كود مخفى) داخل الـ Server (Implicit Code) ← هو عبارة عن كود مخفى
- Parameter via Call (أعمالها تتم من خلال دالة) ← هو عبارة عن دالة Special type ← هو عبارة عن دالة

* Triggers on Table (Insert, update, delete)

* After * Instead of

Create trigger t1

on Student

After Insert

as

select 'Welcome to iTi',

Insert into Student (st-id, st-fname)

Values (779, 'Ali')

Create trigger t2

on Student

After (or) update

as

Select getdate()

update Student

Set st-age = 1;

Create trigger t3

on student

instead of delete

as

Select 'Not allowed for user - + username';

delete from Student where st-id = 779;

Create trigger t4

on department

instead of delete, update, insert

as

Select 'Not allowed';

دنا هنا علامة الـ trigger

حيث كل مرة أروع دالة

جواه هي فقط بعدها الـ Insert ينزل

طبعاً موجود Insert فعلت الدالة

البيبة داخل الجدول ولا هيبيت لا user

رسالة 'Welcome to iTi'

لما باجي أعمل الـ trigger لـ Insert

لأنها بقى فعله تغييرها بعد actions إل بعده كذا باجي أعملها

update لـ trigger على كل actions

trigger student على كل actions

لـ update كل message

هذا هو بقى 2 بيزر دلـ المطلوب منك وتحقيقه

بيانات اليوم إلى بيـنـه فـلا

permission لـ trigger على كل actions

student table على كل actions

هـيـبـيـتـ الـ message

هـيـبـ

Create trigger t7
on Sales.Student
After update
as

Select 'hi';

Alter trigger t7
2 error

Alter trigger Sales.t7
2 error

Create trigger t8
on Sales.Student
After update

as

if update(name)
Select 'hi';

Trigger used inside Alter trigger
Trigger must be in the same Schema
Trigger must be in the same Schema
Schema must be the same as the table

alter table department disable trigger t4;
alter table department enable trigger t4;

Trigger enable-disable
Drop trigger and re-create if needed

Create trigger t5
on Course
After update
as

Select * From inserted
Select * From deleted;

update course
Set crs_name='html', Top_id=1
Where crs_id=900;

use update instead of
Insert and Select instead of
Deleted

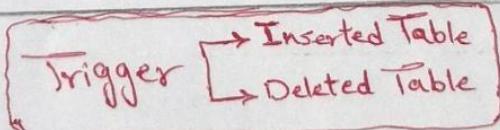
crs_id	crs_name	Top_id
900	html	1

crs_id	crs_name	Top_id
900	oracle	2

Trigger Schema [15]
Buat trigger pada object yang dibuat otomatis
Trigger pada object yang dibuat otomatis

Trigger used inside Alter trigger
Trigger must be in the same Schema
Trigger must be in the same Schema
Schema must be the same as the table

trigger in code [16]
Update trigger must be in the same Schema
Update trigger must be in the same Schema
Schema must be the same as the table



Trigger create || trigger fire ||
Two Tables will be created
Trigger will be created
data types || columns

data from Two Tables
Insert query ||
Delete query

and then
Delete query

Inserted Table ::

delete rows ||
Delete Table ::

update query
Updated Table ::

Inserted Table ::
Deleted Table ::

Original data ::
Deleted Table ::

output

Output

* output is runtime trigger

delete From Student

output getdate(), deleted.St-Fname

Where St-id = 21;

2023-4-6 13:24:21 nada

هنا نلاحظ هنا عامل trigger على جدول student
وكذلك باستخدام output قدرت استخدمها كـ runtime trigger
ويمكنك على مستوى ال Query بها يعني لبيانات ال output
أو delete يمكنه إرسال messages وبيانات.

update Student

set St-Fname = 'Ali'

output Susernamel, inserted.St-Stage into history

where St-id = 1;

Insert into Student (St-id, St-Fname)

output 'Welcome to ITI'

values (444, 'Ali');

هذا نلاحظ هنا output مع update

وهنا أذكر أخذ القيم التي أنا أرغب
وأدخلها في history table

insert into history
values (444, 'Ali')

هذا نلاحظ هنا insert مع output

insert into history
values (444, 'Ali')

XML

9/14/2023
XML: Day 1

SGML & HTML

XML for What?

XML is not for What?

How to Write Well Formed XML?

* HTML Issues ??

- human interference always.
- Fixed Tags.
- HTML Designed to Present not to store.

* SGML Issues ??

- So difficult.

SGML & HTML

* XML For What??

XML

- Is readable and understandable
- 100% Portable - Application Independent
- Extensible - Platform Independent
- Uses human, not computer language. - Language Independent
- We can convert the data to XML for easy manipulation.

* XML is not For ??

- XML is not database *but* we can send the data from the DB with this Format.
- XML is not network protocol HTTP, TCP *but* we can send data XML via protocol.
- XML doesn't need Compiler, as it is human code.
- XML is file based system.

* How to write well formed XML ??

1] XML declaration : to ensure you write XML Format.

<?XML Version="1.0"?>

2] You shouldn't write XML tag.

<XML> ---- </XML>

3] XML File has one Root/Parent ELEMENT.

<employee>

</employee>

4] Tag is Case Sensitive.

<name> ---- </Name> → error

<name> --- </name> → ✓✓

5] Attribute must enclosed by " ".

<employee SSN=1234> → error

<employee SSN="1234"> → ✓✓

6 One tag mustn't contain 2 attributes with same name.

<Employee (ssn)="1234" (ssn)="3567"> → error

7 Overlapping is not supported.

```
<employee>
  <name> Ammar </name>
<employee> → error
  <name>
```

* How to Validate XML ?

- Write the Requirements Via DTD File.
- ② Schema File.

* What is a DTD ?

هي مبارزة مع XML نكتب في XML File بخط Code .
لديه Syntax و هي تكتب في XML File .
ويجب أن يكون XML Based Rules .
لأن XMLvalidate .
ويجب أن يكون XML Standard .
الDTD يبيّن XML Elements بالترتيب .

* DTD Rules

```
<family Attribute> Content </family>
<family> Ahmed </family>
```

DTD

* Content [Ahmed] datatype → PCDATA [string]

* Element Occurrence → + 1 to infinity
* 0 to infinity
? 0 or 1

The Syntax

<!ELEMENT Elementname Datatype>

<!ELEMENT Family (* PCDATA)>

* Attribute datatype → - CDATA [String]

- NMTOKEN [int-string]

- NMToken [int and string]

- ENUMERATION (" " | " " | " ")

- ID ("A1")

(كتلتين وكتلة)

اختيارية متعددة
لرقم من الألف شرقي

The Syntax

<!ATTLIST Elementname [Attributename] Datatype * Required * Implied

<!ATTLIST Family Attribute CDATA * REQUIRED

XML

<Family>

<!-- Self Closed Element -->

<Family>

<Father> Ahmed </Father>

<Family>

<Father>

<name> Ali </name>

</Father>

</Family>

<Family>

<Father>

<name> Ali </name>

<age> 22 </age>

<Salary> 8200 </Salary>

</Father>

</Family>

<Family>

<Father>

<name> Ali </name>

<age> 22 </age>

</Father>

<Mother>

<name> Sara </name>

<age> 21 </age>

</Mother>

</Family>

<Family>

<Father> Ahmed </Father>

<Father> Ali </Father>

<Father> Ayman </Father>

</Family>

<Family>

<Father>

<daughter> Sara </daughter>

</Father>

<Father>

<daughter> Alyaa </daughter>

<son> Ali </son>

</Father>

</Family>

DTD

<!ELEMENT Family EMPTY >

<!ELEMENT Family (Father) >

<!ELEMENT Father (#PCDATA) >

<!ELEMENT Family (Father) >

<!ELEMENT Father (name) >

<!ELEMENT name (#PCDATA) >

مكتوب في الأспектين تكون الأخطاء ملحوظات
عند تقييم .
الخطأ

<!ELEMENT Family (Father) >

<!ELEMENT Father (name, age, salary) >

<!ELEMENT name (#PCDATA) >

<!ELEMENT age (#PCDATA) >

<!ELEMENT salary (#PCDATA) >

(name, salary, age) لونها برتقالي Father في
Validated في error circled
الخطأ

<!ELEMENT Family (father, mother) >

<!ELEMENT father (name, age) >

<!ELEMENT mother (name, age) >

<!ELEMENT name (#PCDATA) >

<!ELEMENT age (#PCDATA) >

هنا كل الفئتين elements التي هي مطلوبة
كما هو عليه في المثلث name, age
الخطأ

<!ELEMENT Family (father)*+>

<!ELEMENT Father (#PCDATA) >

? مكتوب في الأспектين father *+ و father #+
الخطأ

<!ELEMENT Family (father)+>

<!ELEMENT Father (daughter*, son*) >

<!ELEMENT daughter (#PCDATA) >

<!ELEMENT son (#PCDATA) >

هذا يتحقق بالـ father element (أي father يتحقق بالـ father element)
والـ daughter يتحقق بالـ daughter (أي daughter يتحقق بالـ daughter element)
الخطأ

- فـ father يتحقق بالـ (daughter | son) أي father يتحقق بالـ daughter or son
يتحقق بالـ father element (أي father يتحقق بالـ father element)
مع بعض صفات واحدة .

الخطأ

يتحقق مع بعض صفات واحد مفهوم موجود والـ father له .

ومن بين الاثنين مفهوم تتحقق حالياً كـ father .

يعني يتحقق كـ father يا مفهوم تتحقق كـ father .

* DTD Issues :-

- Issues :-

 - ① Limitation of Data types. (PCDATA, CDATA, NMTOKEN).
 - ② Complexity. (DTD → XML)
 - ③ Doesn't Follow XML Rules.
 - ④ Number of occurrence undefined.
 - ⑤ Mixed Content issue.
 - ⑥ Not W3C Standard.

* XML Schema Features :-

- ⑪ Follow W3C Standard.
 - ⑫ Follow XML Rules.
 - ⑬ Various of Data types (int, float, decimal, date, string, ...).
 - ⑭ Limit Number of Occurrence.
 - ⑮ Solve mixed Content.

-<13books>

<book>
 <isbn>123Abc </isbn>
 <author>J. K. Rowling </author>
 <name>Harry Potter </name>
 <year>2005 </year>

رغم اختلاف لا book للأول والا book الثاني
إلا أنه لا XMRules معه وتمام
الـ Validation DTD تمام معه \checkmark
ـ business rule فقط ولا ننسى في حال
ـ مطابقاً DTD.

```
</book>
<from> Cairo </from>
<to> Jeddah </to>
<ticket> 12377 </ticket>
<isbn> 123Abc </isbn>
</book>
```

لادخن لا
كتاب كتاب

<1Books>

<Books xmlns:bt="www.bookreserve.com" xmlns:br="www.bookread.com">

```
<br:book>
<br:isbn>123Abc </br:isbn>
<br:author>J.K. Rawling </br:author>
<br:name>Harry Potter </br:name>
<br:year>2005 </br:year>
```

—</bR:book>

<bt:book>
<bt:from> Cairo <bt:from>
<bt:to> Jeddah <bt:to>
<bt:ticket> 12377 </bt:tic

لارجست XML developer لا طریق namespace لا تمیز دارند
• BI developer لا ویرایش کنند

-</bt=book>

داخل الـ Schema لازم ينطوي على تكتب بالاسم بالكامل وذلك عن طريق كتابة (كx) في بداية إلمنت ELEMENT.

وذلك عن طريق كتابة (Ex) في بابيّه

<1Books>

* XML Schema types:-

1) Simple type

<family> abc </family>

2) Complex type

* <family1> (Self Close)

* <family name="abc"> dssd </family> (Element with content and Attribute)

* <family> abc (Mixed Content)

<name> sss </name>

</Family>

1) Simple type

XML: <family> Ali </family>

Schema: <x5:element name="family" type="x5:string" />

2) Complex type

XML:

<family>
 <name>Ali </name>
</family>

Schema

<x5:element name="family">
 <x5:complexType>
 <x5:sequence>
 <x5:element name="name" type="x5:string" />
 </x5:sequence>
 </x5:complexType>
</x5:element>

XML

<family>
 <father>
 <name> Ali </name>
 </father>
</family>

Schema

<x5:element name="family">
 <x5:complexType>
 <x5:sequence>
 <x5:element name="father" type="x5:complexType" />
 </x5:sequence>
 </x5:complexType>
</x5:element>

(XML) <Family>

<Father>

<name> Ali </name>

<age> 22 </age>

<salary> 1200 </salary>

</Father>

</Family>

(Schema)

<xsd:element name="Family">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="father">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="name" type="xsd:string"/>

<xsd:element name="age" type="xsd:integer"/>

<xsd:element name="salary" type="xsd:integer"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

(XML)

<Family>

<father>

<name> Ahmed </name>

</father>

<father>

<name> Ali </name>

</father>

<father>

<name> Osama </name>

</father>

</Family>

(Schema)

<xsd:element name="Family">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="father" minOccurs="0" maxOccurs="unbounded">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="name" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:sequence>

<xsd:complexType>

</xsd:element>

XML

```

<family>Ahmed
  <father>Osama
    <ssn>123</ssn>
  </father>
</family>
  
```

Complex type "النوع المركب" معلومات متعددة
Simple type "النوع البسيط" معلومات بسيطة

Schema

```

<xsd:element name="family">
  <xsd:complexType mixed="true">
    <xsd:sequence>
      <xsd:element name="father">
        <xsd:complexType mixed="true">
          <xsd:sequence>
            <xsd:element name="ssn" type="xsd:integer"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
  
```

XML

```

<family>
  <father>Osama
    <ssn>123</ssn>
  </father>
</family>
  
```

<"father"> امر تكراري <xsd:sequence>

```

<student>Osama
  <ssn>123</ssn>
</student>
  
```

student list <students> (list)

<"student"> امر تكراري <xsd:sequence>

```

<student>Osama
  <ssn>123</ssn>
</student>
  
```

XML

```

<family loc="Cairo">
  <father age="22" name="Ahmed">
    <son ssn="123">
      <name>Ali</name>
    </son>
  </father>
</family>

```

Schema

```

<xsd:element name="family">
  <xsd:complexType mixed="true">
    <xsd:sequence>
      <xsd:element name="father">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="son">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="name" type="xsd:string"/>
                </xsd:sequence>
                <xsd:attribute name="ssn" type="xsd:integer"/>
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="age" type="xsd:integer"/>
    <xsd:attribute name="name" type="xsd:integer"/>
  </xsd:complexType>
</xsd:element>

```

<xsd:attribute name="Loc" type="xsd:string" use="required"/>

XML

```

<family loc="Cairo">ALALI</family>

```

Schema

```

=> <family loc="Cairo">ALALI</family>

```

```

<xsd:element name="family">
  <xsd:complexType mixed="true">
    <xsd:attribute name="Loc" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>

```

XML

```

<family /> self close

```

Schema

```

<xsd:element name="family">
  <xsd:complexType/>
</xsd:element>

```

XML <Family Loc="Cairo" > ABC

<grandfather SSN="10327" > ALALI

<father> Ahmed

<mother job="Eng" Sport="Football" > Sara </mother>
<daughter age="22" Sport="Football" > Alyaa </daughter>
<daughter age="22" Sport="Football" > Eman </daughter>
<son age="22" Sport="Football" > Ahmed </son>

</father>

<father> mohammed

<mother job="Eng" > Sara </mother>
<daughter age="22" Sport="Football" > Alyaa </daughter>
<son age="22" Sport="Football" > Samy </son>
<son age="22" Sport="Football" > Ahmed </son>

</father>

<uncle> Ali </uncle>

<uncle> Osama </uncle>

<uncle> Ziad </uncle>

</grandfather>

</family>

Schema

<xsd:element name="Family">
<xsd:complexType mixed="true">

<xsd:sequence>

<xsd:element name="grandfather">

<xsd:complexType mixed="true">

<xsd:sequence>

<xsd:element name="father" minOccurs="1" maxOccurs="unbounded">

<xsd:complexType mixed="true">

<xsd:sequence>

<xsd:element name="mother">

<xsd:complexType mixed="true">

<xsd:attribute name="job" type="xsd:string"/>

<xsd:attribute name="Sport" type="xsd:string"/>

</xsd:complexType>

</xsd:element>

<xsd:element name="daughter" minOccurs="0" maxOccurs="unbounded">

<xsd:complexType mixed="true">

<xsd:attribute name="age" type="xsd:integer" use="optional"/>

<xsd:attribute name="Sport" type="xsd:string"/>

</xsd:complexType>

</xsd:element>

<xsd:element name="Son" minOccurs="0" maxOccurs="5">

<xsd:complexType mixed="true">

<xsd:attribute name="age" type="xsd:integer"/>

<xsd:attribute name="Sport" type="xsd:string"/>

</xsd:complexType>

</xsd:element>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="uncle" type="xsd:string" minOccurs="1" maxOccurs="10"/>

<xsd:sequence>

<xsd:attribute name="SSN" type="xsd:integer"/>

</xsd:complexType>

</xs:element>

```
<xs:element>
</xs:sequence>
<xs:Attribute name="loc" type="xs:string"/>
</xs:ComplexType>
</xs:element>
```

* Create Special Data Type

<xs:schema>

```
<xs:simpleType name="Special DT">
<xs:restriction base="xs:string">
<xs:enumeration value="football"/>
<xs:enumeration value="Volleyball"/>
<xs:enumeration value="tennis"/>
</xs:restriction>
</xs:simpleType>
```

بيانات مختبرة، لها قيمة

Data Validation

Schema يحد قيم

of restriction

```
<xs:simpleType name="Special DT2">
<xs:restriction base="xs:integer">
<xs:minInclusive value="18"/>
<xs:maxInclusive value="60"/>
</xs:restriction>
</xs:simpleType>
```

</xs:schema>

```
<xs:schema>
<xs:element name="Book">
<xs:complexType>
<xs:sequence>
<xs:element name="Title" type="xs:string"/>
<xs:element name="Author" type="xs:string"/>
<xs:element name="Year" type="xs:int"/>
<xs:element name="Price" type="xs:decimal"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

</xs:schema>

XPATH - XSL

* CSS issues with XML presentation:-

① All the data will be displayed everytime.

- We want to hide some information (Such displaying).

② CSS built for html.

- We cannot transform any data with another value such $\text{Salary} = \text{Salary} * 10$.

XSL → extensible stylesheet language.

XSLT → extensible stylesheet language transformation.

XSL

- Transform into html

XSLT

- Transform into html, pdf, doc, -----

XPath

XPath → new technique of How catch any element from XML.

```

<Family>
  <Father SSN = "123">
    <name>Ali</name>
    <age>22</age>
    <salary>5200</salary>
  </Father>
</Family>

```

$<\text{Family}>$ ⇒ root element node.

$<\text{Father}>, <\text{name}>$ ⇒ element node.

$\text{SSN} = "..."$ ⇒ attribute node.

Ali, 22, 5200 ⇒ atomic values.

Parent: $\swarrow \searrow$ $<\text{name}> \Rightarrow <\text{Father}>$ $<\text{Father}> \Rightarrow <\text{Family}>$

Child:

$\swarrow \searrow$ $<\text{Family}> \Rightarrow <\text{Father}>$ $<\text{Father}> \Rightarrow <\text{name}>, <\text{age}>, <\text{salary}>$

Ancestor:

$\nwarrow \nearrow$ $<\text{age}> \Rightarrow <\text{Father}>, <\text{Family}>$

Descendent:

$\nearrow \nwarrow$ $<\text{Family}> \Rightarrow <\text{Father}>, <\text{name}>, <\text{age}>, <\text{salary}>$

Siblings:

$\nearrow \nwarrow$ $<\text{name}> \Rightarrow <\text{age}>, <\text{salary}>$

```

<employees>
  <emp name="abc">
    <ssn> 222 </ssn>
    <name> Ali </name>
    <password> P@$$word </password>
    <hiredate> 2000 </hiredate>
    <salary> 1200 </salary>
    <comment> hello </comment>
  </emp>
  <emp name="xyz">
    <ssn> 333 </ssn>
    <name> Amr </name>
    <password> P@@@35W </password>
    <hiredate> 2000 </hiredate>
    <salary> 1300 </salary>
    <comment> hello </comment>
  </emp>
  <emp name="dsda">
    ——
    ——
    ——
    ——
    ——
  </emp>
  <emp name="cdec">
    ——
    ——
    ——
    ——
  </emp>
  <comment> All employees </comment>
</employees>

```

XPath

/ → Catching all XML File, Stands out of employees root node.

/Employees → Stands out of emp node element.

/Employees/emp/ssn → <ssn> 222 </ssn>, <ssn> 333 </ssn>, ---

/Employees/emp/ssn/text() → 222, 333, ---

/Employees/emp/@name → "abc", "xyz", "dsda", "cdec"

and /Employees/emp/comment → /Employees/comment → "hello", "hello", "hello", "hello", "All employees"

? = //Comment → retrieve all Comment node from any place.

/Employees/emp[1] → Catch the first emp>

/Employees/emp[1]/name → Catch the <name> From the First<emp>.

| Employees | emp[last()] → Catch the last <emp>
| Employees | emp[last()]/@name → Catch the attribute From the last <emp>
| Employees | emp[last()]/name → Catch the <name> From the last <emp>
| Employees | emp[last() - 1] → Catch the prelast <emp>
| Employees | emp[Position() > 2] → Catch the 3rd - last <emp>
| Employees | emp[@name] → Catch all <emp> that have @name
| Employees | emp[not(@name)] → Catch all <emp> that don't have @name
--- / . / --- → Go back one step to the parent

XSL

```
<XSL:stylesheet>
  <XSL:template match="/">
    <html>
      <head>
        <title> Students </title>
      </head>
      <body>
        <div style="background-color: red; font-size: 18px"> Ahmed </div>
        <div style="background-color: green"> Osama </div>
        <div style="background-color: yellow"> Rana </div>
      </body>
    </html>
  </XSL:template>
</XSL:stylesheet>
```

```
<XSL:stylesheet>
  <XSL:template match="/">
    <html>
      <head>
        <title> Lab2 </title>
      </head>
      <body>
        <div style="background-color: yellow">
          <XSL:value-of select="CATALOG/CD/TITLE"></XSL:value-of>
        </div>
        <div style="background-color: gray">
          <XSL:value-of select="CATALOG/CD/ARTIST"></XSL:value-of>
        </div>
      </body>
    </html>
  </XSL:template>
</XSL:stylesheet>
```

<XSL:stylesheet>

```
<XSL:template match="/">
<html>
  <head>
    <title>Lab3 </title>
  </head>
  <body border="5">
    <table>
      <tr>
        <th>Title </th> <th>Artist </th>
      </tr>
      <xsl:for-each select="CATALOG/CD">
        <tr>
          <td><xsl:value-of select="TITLE"></xsl:value-of> </td>
          <td><xsl:value-of select="ARTIST"></xsl:value-of> </td>
        </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</XSL:template>
</XSL:stylesheet>
```

<XSL:template match="/">

```
<html>
  <head>
    <title>Lab4 </title>
  </head>
  <body border="3">
    <table>
      <tr>
        <th>Title </th> <th>Artist </th> <th>Price </th>
      </tr>
      <xsl:for-each select="CATALOG/CD">
        <xsl:if test="PRICE > 9">
          <tr>
            <td>
              <xsl:value-of select="TITLE"></xsl:value-of>
            </td>
            <td>
              <xsl:value-of select="ARTIST"></xsl:value-of>
            </td>
            <td>
              <xsl:value-of select="PRICE"></xsl:value-of>
            </td>
          </tr>
        </xsl:if>
      </xsl:for-each>
    </table>
  </body>
</html>
</XSL:template>
```

```

<XSL:template match="/">
  <html>
    <head>
      <title>Lab5</title>
    </head>
    <body>
      <table border="3">
        <tr>
          <th>Title</th><th>Artist</th><th>Price</th>
        </tr>
        <XSL:for-each select="CATALOG/CD">
          <XSL:if test="PRICE > 9">
            <tr background-color="red">
              <td><XSL:value-of select="TITLE"/></td>
              <td><XSL:value-of select="ARTIST"/></td>
              <td><XSL:value-of select="PRICE"/></td>
            </tr>
          </XSL:if>
          <XSL:if test="PRICE < 9">
            <tr background-color="green">
              <td><XSL:value-of select="TITLE"/></td>
              <td><XSL:value-of select="ARTIST"/></td>
              <td><XSL:value-of select="PRICE"/></td>
            </tr>
          </XSL:if>
        </XSL:for-each>
      </table>
    </body>
  </html>
</XSL:template>

```

(نهايات مطبى بإنعام
 و XSL:output لـ output XML
 و XSL:output لـ output HTML
 و XSL:output لـ output لـ الأرض)
 > ← أكبر من
 < ← صغير من

<XSL:template match="/">
 <html>
 <head>
 <title/>
 </head>
 <body>
 <table border="1">
 <tr>
 <th>Title</th><th>Artist</th><th>Price</th>
 </th>
 <XSL:for-each select="CATALOG/CD">
 <!-- XSL:Sort Select="PRICE" dataType="number" Order="ascending" --></sort-->
 <XSL:Sort Select="TITLE" dataType="Text" order="ascending"></sort>
 <XSL:choose>
 <XSL:when test="PRICE > 8">
 <tr bgcolor="red">
 <td><XSL:value-of select="TITLE"/></td>
 <td><XSL:value-of select="ARTIST"/></td>
 <td><XSL:value-of select="PRICE"/></td>
 </tr>
 <XSL:when test="PRICE <= 8">
 <tr bgcolor="green">
 <td><XSL:value-of select="TITLE"/></td>
 <td><XSL:value-of select="ARTIST"/></td>
 <td><XSL:value-of select="PRICE"/></td>
 </tr>
 <XSL:when test=">">
 <tr bgcolor="blue">
 <td><XSL:value-of select="TITLE"/></td>
 <td><XSL:value-of select="ARTIST"/></td>
 <td><XSL:value-of select="PRICE"/></td>
 </tr>
 </XSL:otherwise>
 </XSL:choose>
 </XSL:for-each>
 </table>
 </body>
 </html>
 <XSL:template>

20/5

XML in SQL

For XML Table to XML

- For XML raw → For normal Select Statement.
 - For XML auto → For Join Statement.
 - For XML Path → output (elements + attributes)
 - For XML explicit

XML to Table

Open XML

15 steps

- ① declare XML Variable
 - ② declare int Variable
 - ③ Execute ~~-----~~
 - ④ Select *
From OpenXML(---
With(---
 - ⑤ Execute
Op-XML-remove document

For XML

For XML row

```
Select *  
From Student  
For XML Raw ('Student')
```

Select *
From Student
For XML Raw ('Student'), elements, Root ("")

Select *
From Student
For XML raw ('Student'), elements xsinil, Root ('Students')
null values (مفردة) لlive elements لـ xsinil تم استخدامه (نحو)

For XML auto

Select topic . Top-Id , Top-name , Crs-Id , Crs-name
From Topic , Course
Where Topic . Top-Id = Course . Top-Id
For XML auto , Elements ;

لارجع XML auto تيافر (س) الجدول Default

```
<Topic>
  <Top-id>1 </Top-id>
  <Top-name> Programming </Top-name>
  <Course>
    <Crs-id>200 </Crs-id>
    <Crs-name>C </Crs-name>
  </Course>
  <Course>
    <Crs-id>300 </Crs-id>
    <Crs-name>OOP </Crs-id>
  </Course>
</Topic>
```

For XML Path

Select St-id "@StudentID",
 St-Fname "StudentName/Firstname",
 St-Lname "StudentName/Lastname",
 St-Address "Address"
 From Student
 For XML Path ('Student');

```
<Student StudentID="1">
  <StudentName>
    <Firstname> Ali </Firstname>
    <Last name> Magdy </Lastname>
  </StudentName>
  <Address> Cairo </Address>
</Student>
```

Steps :-

1) declare @docs XML = '<Students>

Variable (أول خطوة بعده XML) 1
 XML معرفة بـ XML مع مساعدة XML →

Open XML

```
<Student StudentID="1">
  <StudentName>
    <Firstname> Ali </Firstname>
    <Last name> Magdy </Lastname>
  </StudentName>
  <Address> Cairo </Address>
</Student>
```

For XML Path
 Open XML (XML to Table)

2) declare @hdocs int

Variable (ثانية خطوة بعده XML) 2
 doc int من نوع handle

3) Execute SP_XML-preparedocument @hdocs output, @docs

4) Select *

From OPENXML (@hdocs, '/Student')
 with (StudentID int '@StudentID',
 Address varchar(10) 'Address',
 StudentFirst varchar(10) 'StudentName/Firstname',
 StudentLast varchar(10) 'StudentName/Lastname'
),

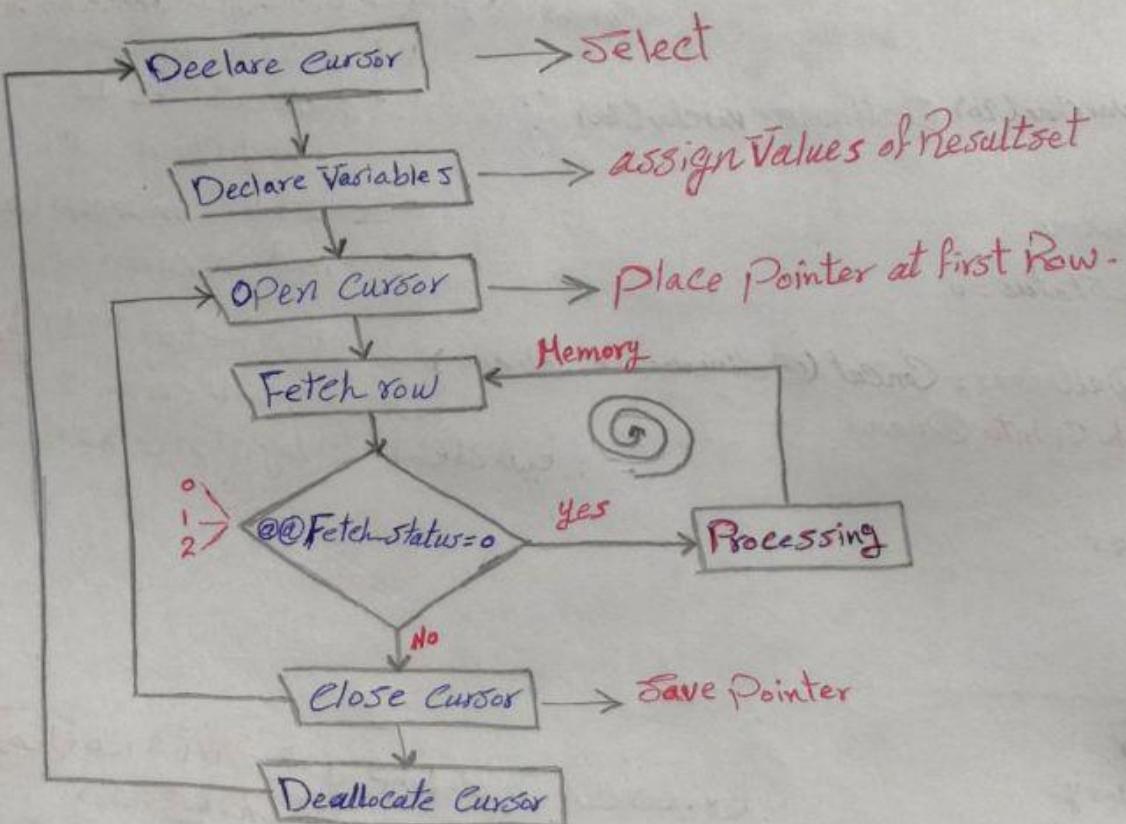
5) Execute SP_XML-removedocument @hdocs

data (بيانات الـ XML) 3
 الموجودة في XML، يدخل memory Table في Table
 memory Table في Insert (إدخال)
 Select * into Tablename
 From openxml (---
 ---)

Day 10
16/4/2023

CURSOR

→ How to deal with the Resultset row by row (Loop For Rows)



declare C1 cursor

For select stid, st.fname
From student
where st.address = 'cairs'

For read only --- update

declare @id int @name Varchar(20)

Open C1

Fetch C1 into @id, @name

While @@FETCH_STATUS = 0

begin

select @id, @name

fetch C1 into @id, @name

End

Close C1

Deallocate C1

output

1 - 1 -

1 - 1 -

1 - 1 -

Separate Row num 1

Separate Row num 2

Separate Row num 3

declare C1 Cursor
For Select st-fname
From student
Where st-fname is not null
For read only

Select @allnames
Close C,
deallocate C,

declare C1 Cursor
For Select Salary
From Instructor

For update
declare @Sal int
open C1
Fetch C1 into @Sal
while @@Fetch_Status = 0
begin
 IF @Sal >= 300
 update Instructor
 set Salary =
 Where Current
 else
 update Instructor
 set Salary =
 Where Current
 Fetch C1 into @Sal
End

Close C_1
deallocate C_1

declare C1 Cursor
For Select St-Fname
From Student
For read only

هذا ينادي على اسسوء الترتيب في البرمجة
يعني كل من Cursor ذات
Table Ahmed و Amr

declare @name varchar(20), @Counter int=0, @Flag int=0
Open C1
Fetch C1 into @name
While @@Fetch_Status=0
begin
-- IF @name = 'Ahmed'
Set @Flag=1
-- IF @name = 'Amr'
begin
IF @Flag=1
begin
Set @Counter+=1
Set @Flag=0
End
End
Fetch C1 into @name
End
Select @Counter
Close C1
Deallocate C1

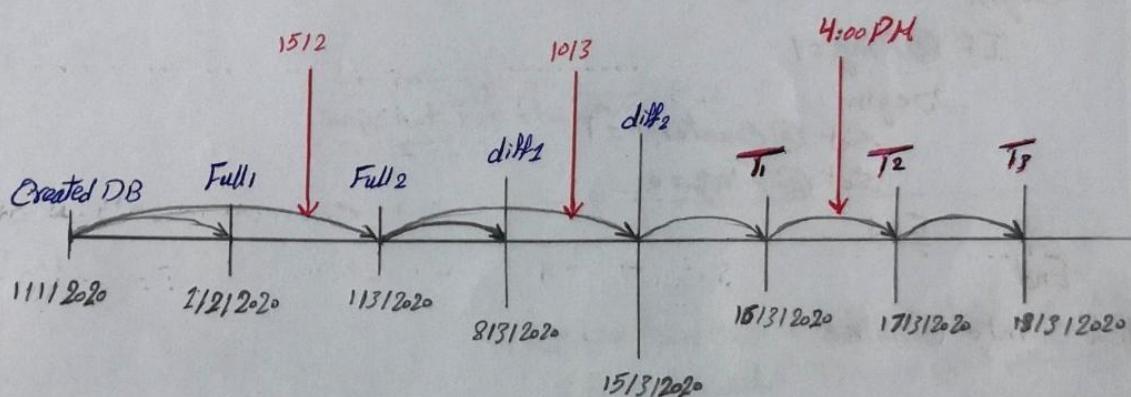
Backup - Restore

Types of backups:-

- 1) Full backup → From Date of created DB (to) Date of backup day.
- 2) Differential backup → From Last Full Backup date (to) Date of backup day.
- 3) Transaction log backup → From Last Backup regardless of the Backup type.
- 4) Filegroup backup

يعني يكون ال Backup المطلوب Differential Backup + Full Backup
يعني يكونعبارة عن الملفات metadata و Data

Lock File Jlna Backup ي يكون Transaction log Backup
يعني Lock File Jlna للرجوع Queries Jlna Backup يعني



Created DB Jlna backup يملي على Full1 , Full2

- Full Backup Full2 Jlna backup يملي على diff2 , diff2

- Full Backup Full2 Jlna backup يملي على diff2 , diff2

* فـ Full و Full2 Jlna backup يملي على diff2 و Full2 Jlna backup يملي على diff2

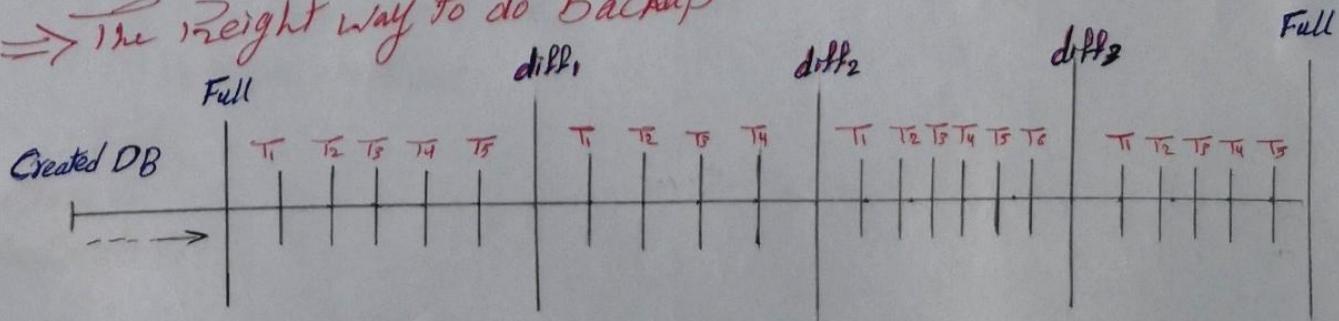
يعني أقدر أعمل اد اسماً يام 15/13 و 8/13 و 1/3 و 1/2 و 1/1

10/13 و 15/12

* قدر أعمل لـ DB لـ Date and Time Queries Jlna backup يملي على diff2

4:00 PM Jlna backup day

⇒ The Right way to do backup



* كل عشر ساعة تتم Full backup كل أسبوع تتم diff backup كل يوم تتم diff backup

* Three ways to do Backup :-

- 1) Backup like Tasks تم اختيار Tasks لا DB و اختار Right click وذلك من طريق From Wizard
 - 2) Backup Query طريقة تابع و أكتب النوع الذي يطلب في ال Backup Query
 - 3) Backup database ITI to disk = 'D:\iti-db.bak' => Full backup
- Backup database ITI to disk = 'D:\iti-db.bak' => diff backup With differential**

مهمة Job New Job Right click SQL Server agent وذلك من طريق SQL Server agent name name لا داعي وطبعاً بختار انفصاله master db اكتب الكود المراد تنفيذه Run query Schedule Job دلالة Job دلالة كل قراراتك و كل مو. لوهاب بختار Start Job Right click Task لاختار

Identity Insert

Select *
From Student

StId	Stname	Stage
1	-	-
2	-	-
3	-	-
10	-	-
11	-	-
12	-	-

Identity هو StId يعني مفترض أنه
في بياناتي موجود حساب بين الـ 3 و 10

Set Identity_Insert Student ON;

Insert into Student (StId, Stname, Stage)
Values (4, -, -), (5, -, -), (6, -, -), (7, -, -),
(9, -, -), (10, -, -)

Set Identity_Insert Student OFF;

عن طريق الامر Identity Insert على الجدول
وأدخلت الورق ON وذلك يرجع إلى بدل
داخل الـ Identity Column Insert
الـ Insert تفعيل بعدها العرض OFF.

* How to reset identity

1) using Truncate statement

2) use insert-identity (ON) then (OFF)

3) use dbcc checkident (Table-name, Reseed, [num])

dbcc checkident (Student, Reseed, 3);

Identity Reset creation
Insert but there is no new value

* To know the Identity of the Table.

Result Set
Select @@identity

Identity current
Select IDENT_CURRENT ('Student')

The Three Ways to Create

Backup

Identity Insert (ON / OFF)

How to Reset Identity

Bulk Insert

Types of insert statement:-

- 1) Simple insert.
- 2) Insert constructor.
- 3) Insert based on select.
- 4) Insert based on execute.
- 5) Bulk Insert.

لوبغارة data \| \data \| Insert جس نه موجودة في Hard Disk \| file

Bulk Insert Student
From 'd:\mydata.txt'
with(Fieldterminator = ',')

Snapshot

* Snapshot is:-

- readonly Database.
- used for reporting only.
- Capture DB in the time of taking snapshot.
- I can take more than one snapshot for the same DB in different times.
- Used only with NTF5 format only.
- Copy on write Concept.

ITI - database

ID	Name	—	age
1	Ahmed	—	22
2	Eman	—	21
3	Omar	—	25
4	Sally	—	22

Snapshot DB (18/11/2020)

ID	Name	—	age
1	Ahmed	—	22
2	Eman	—	21
3	Omar	—	25
4	Sally	—	22

ادل DB ایت Snapshot فاصیہ
ادل داتا الوجوہة خالد DB وقیہ
صافت ال Snapshot، همینکو as readonly
و بنجھے 90% هن بتلوں DB نہیں کھانہ.
هن کئی نکا لفظ لا DB الوجوہة
عندکو و تھے ملکا دینا ایسا دینا
لے کر پیدا کرنا، restore

ITI - Database

ID	Name	—	age
1	Ahmed	—	22
4	Sally	—	28
9	Khaled	—	21

id=1
Copy on write

ID	Name	—	age
2	Eman	—	21
3	Omar	—	25
4	Sally	—	22

ادل age وغیرت لا Eman, omar | delete Calek Lisit
Copy ایل Snapshot ایل ایل ایل Sally ایل
ادل ایل ایل ایل ایل ایل ایل ایل ایل
Pointer file friend Ahmed ری ماہم و زیاد
Server ایل ایل ایل ایل ایل ایل ایل ایل
ادل ایل ایل ایل ایل ایل ایل ایل ایل
و سفر ایل ایل ایل ایل ایل ایل ایل ایل

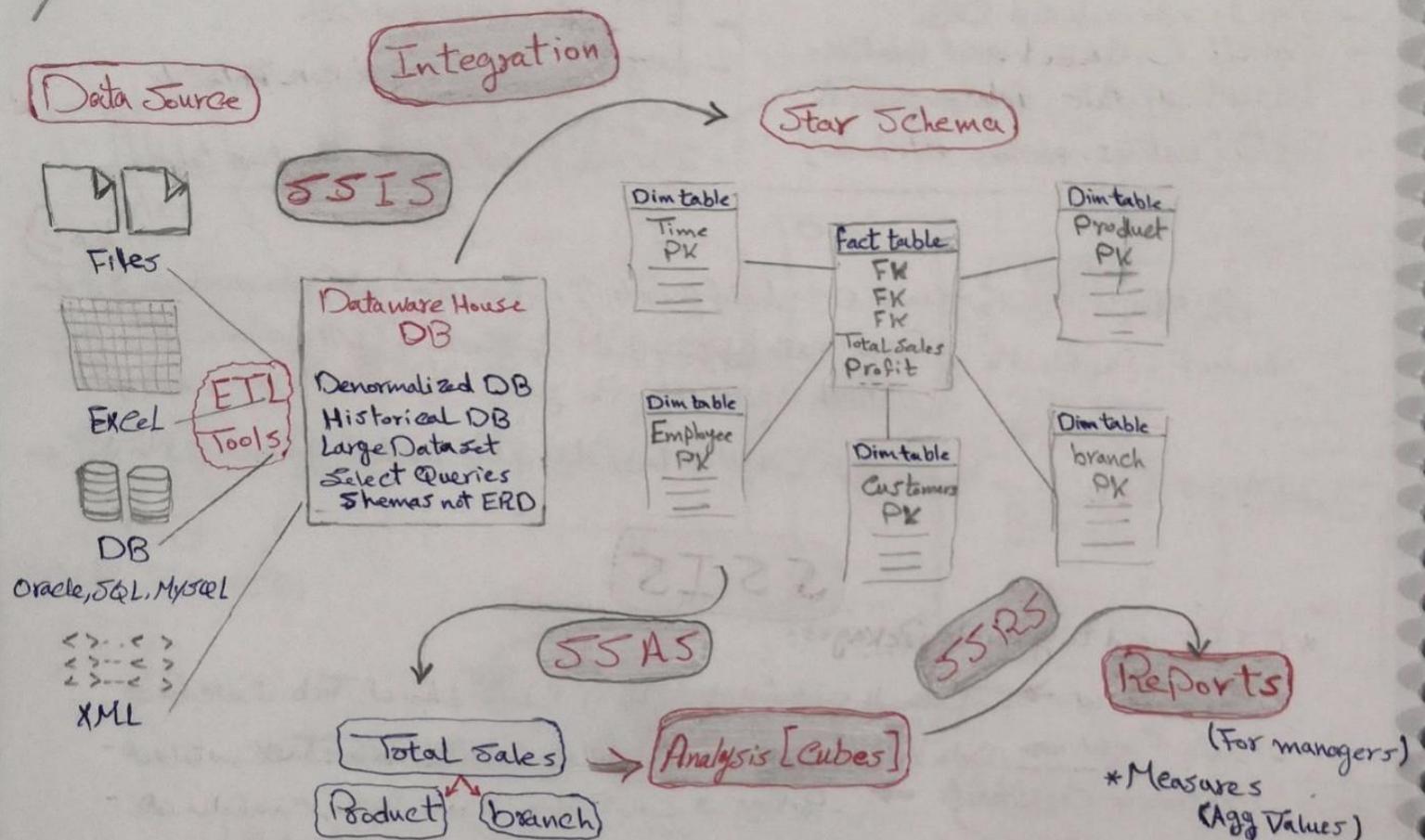
Create Database itisnap

on
(
name = 'ITI', → DB ایل mdf ایل ایل ایل
filename = 'd:\itisnap.ss' → ایل ایل ایل ایل
) → File Create جس ایل ایل ایل ایل
as Snapshot of ITI; → HardDisk\itisnap ایل ایل ایل

* Restore database From Snapshot :-

restore database ITI → DataBase
From database-Snapshot = 'itisnap';
2> SnapshotName

ITI DB ایل restore ایل ایل ایل ایل
.itisnap ایل ایل ایل ایل ایل ایل ایل ایل



- Fact Table
- Fact Table measures (new managers) → measures needed by managers
- Fact Table dimensions (Product, branch) → dimensions needed by managers
- Fact Table level of granularity (Total Sales, Product, branch) → level of granularity needed by managers
- Fact Table insert, update, delete (new managers) → insert, update, delete needed by managers
- Fact Table select (new managers) → select needed by managers
- Fact Table denormalized (new managers) → denormalized needed by managers
- Fact Table warehouse (new managers) → warehouse needed by managers
- Fact Table cube (new managers) → cube needed by managers
- Fact Table report (new managers) → report needed by managers
- Fact Table datamining (new managers) → datamining needed by managers
- Fact Table Discover hidden Relationships (new managers) → discover hidden relationships needed by managers
- Fact Table Classify data (new managers) → classify data needed by managers
- Fact Table Predict the future data (new managers) → predict the future data needed by managers

الهدف من BI هو بناء Reports و Websites و APPs لدعم managers

BI Leads to:-

- * Fact-based decision making.
- * Single point of trust →

فما هي العوامل التي تؤدي إلى BI
 - مصدرون للبيانات (Data Sources) مثل Tables و Views
 - دليل البيانات (Datatype) مثل Integers و Dates و Booleans
 - إدخالات (Data Input) مثل Queries و Subqueries
 - خوارزميات (Data Processing) مثل Transformation و Transformation Rules
 - خوارزميات (Data Mining) مثل Datamining algorithms
 - عوامل التعلم الآلي (Machine Learning) مثل Predictive modeling
 - عوامل التحليل (Analytics) مثل Data Analysis و Data Visualization
 - عوامل التقارير (Reporting) مثل Reports و Dashboards
 - عوامل التحكم (Control) مثل Security و Access Control
 - عوامل التخزين (Storage) مثل Data Warehouses و Data Lakes

Operation DB

OLTP

- Data of run business.
- ERD: normalized DB.
- Small Dataset and volatile.
- Insert, update, delete, select.
- ERD [entities, relations, attributes]

Datawarehouse

OLAP

- Historical data 3-15 years
- ETL: denormalized DB.
- Large Dataset and non-volatile
- only Select mostly
- Schema [fact tables, dimension tables]

فقط من الـ Datawarehouse لو أنت أخذت معيin level of granularity يعني متى تنزل لـ قلص يعني
 Per Week و Per Month بـ Reports (أيام) و (أيام) Per Quarter بـ DWH (أيام) و (أيام) و
 . Cell Level بـ Per Year بـ Reports (أيام) و (أيام) و
 . DB تكون خريطة من الـ DWH level of granularity ما الـ DB ← العادي.

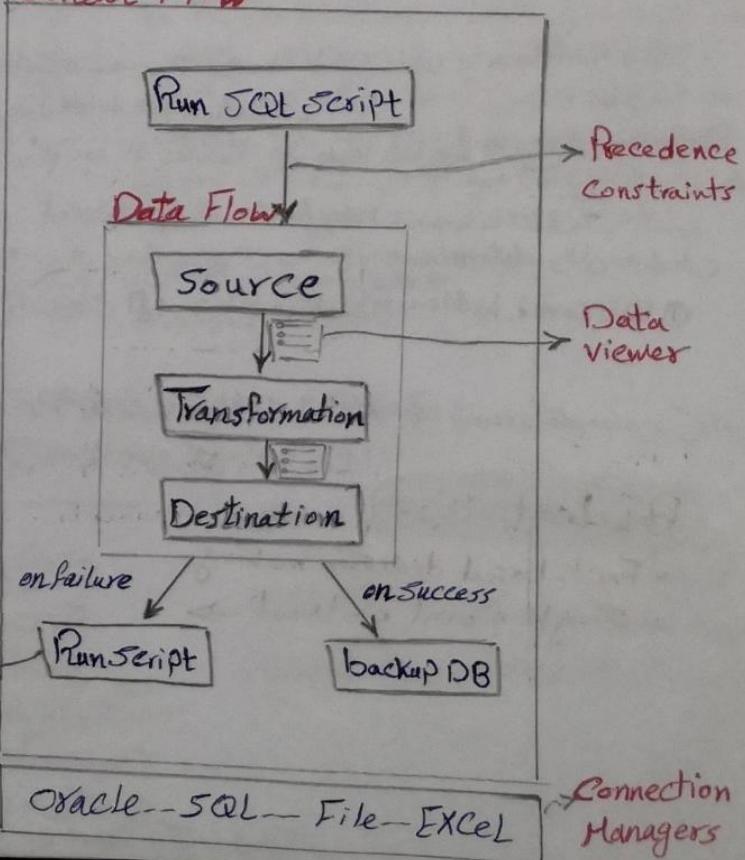
SSIS

* SSIS used to generate Packages.

- Control Flow → Data Flow Tasks (التي تجري على Tab)
- Data Flow → Data Flow Transformation (التي تجري على Control Flow Layer: Task)
- Precedence Constraints → Package (أي Pipeline)
- Connection Managers → data (أي Connection)
- Variables → هي Package هي في Table و هي store في Container
- Data Viewers → هي Package هي في Data Flow

لارزم لهايدين.
 Package
 تجعل
 ترسها على الورق
 المثلث قبل ما تفتحها.

Control Flow



Operation DB (OLTP) VS Data Warehouse (OLAP)

SSIS

SSIS Packages

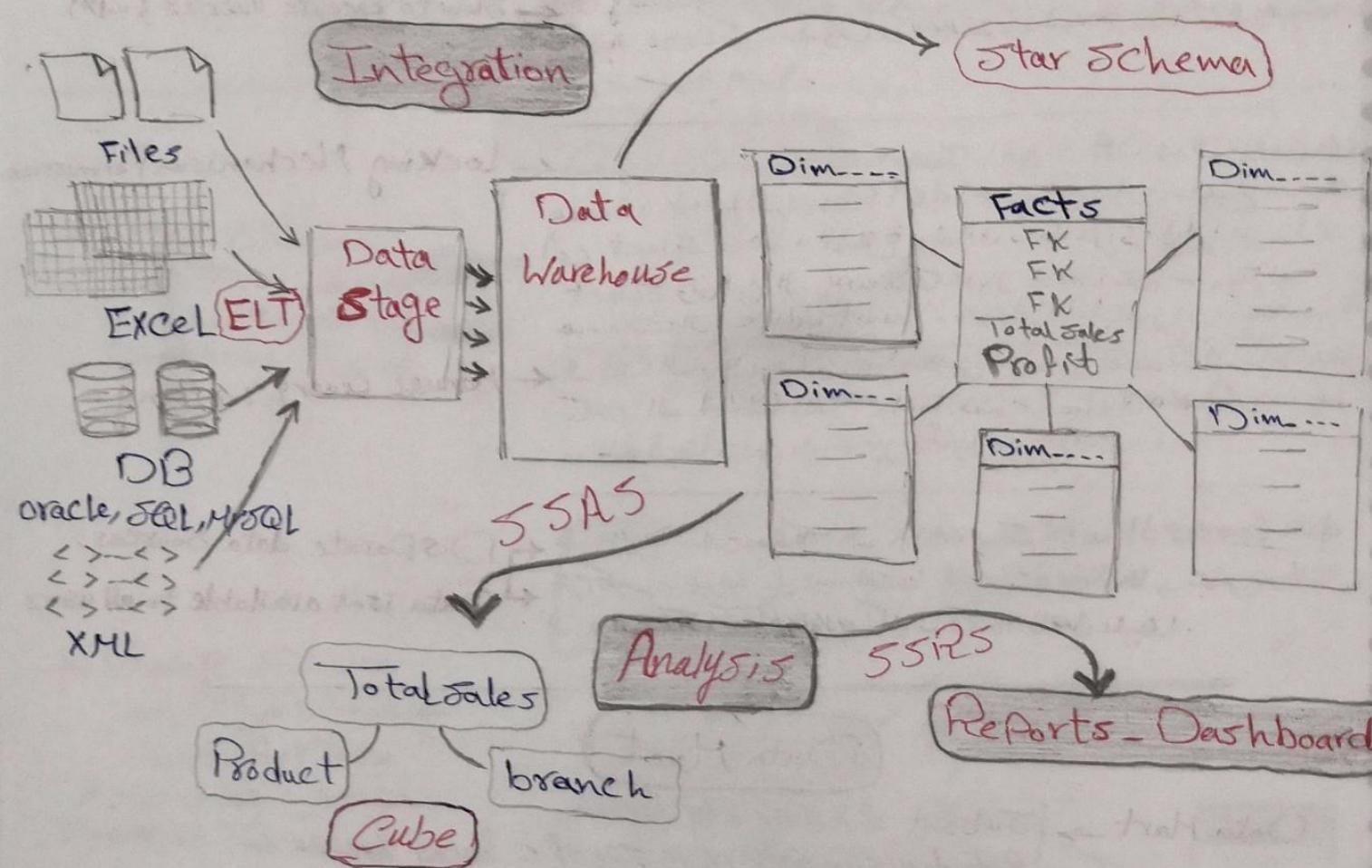
OK

Connection
Managers

BI
Day 2
28/4/2023

BI: Day 2

BI



Data-Warehouse is ---

- Select, Insert, Update, Delete, Performance, etc. (مقدمة في الـ DWH)
- Insert-update-delete فقط مفهوم
- Subject-oriented (الـ DWH يبيّن تفاصيل في الـ Subject والـ measure)
- Integrated (الـ DWH ي統合 different sources into one integrated Data)
- Time-varying (الـ DWH يحتوي على historical data over time)
- Non-Volatile (الـ DWH هو عبارة عن داتا لا يخسرها)
- Levels of granularity (الـ DWH هي قواعد بيانات متعددة من حيث الحجم)

Why we use OLAP not OLTP?

because OLTP is ...

ال DB to Cube سريع execution time 1000 queries per second
Performance is better than OLAP because it is faster and more efficient.
ال OLAP is slower than OLTP because it is slower and more complex.

يسهل عمل على DB داخل OLAP Insert, update, delete
Locking Mechanism Permission.
يسهل حفظ Rows Lock لبيانات المعمليات ويعين حفظ
أي Select من هذه الفترات وذلك يعتمد على تفصيل إجراءات
Select وربطها بال Query DWH تكون سريعة جداً
حيث أن Select, Insert, Update - Delete إدخالات زادت بعدها
ال DB العالمية تختلف اصنافها من كتابة أو Queries
عکس ال DWH بها Tools تقوم ببناء ال Queries وتحتها
وذلك أيفاً تجعل على توفير الموارد الوقت .
← Manual Query writing.

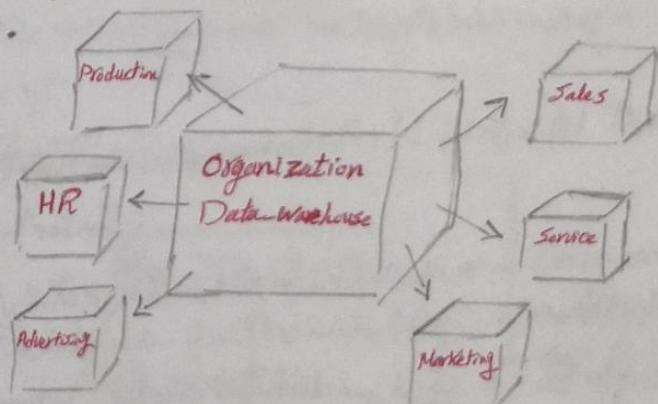
data sources separation ال DWH منفصل عن data sources.
ال DWH موجود في different locations على المدارد متفرق ومتعدد
Data isn't available to all users. ← كل ال users لا يحصل عليها .

Data Mart

Data Mart → Subset of a data warehouse.

- Aids decision making in specific focus area.
- Has a special focus.

ال Data Mart هوsubset من data warehouse لما أجمع المطلوب
Report أو فعل Cube بيكورة أسرع بكثير
من فعل ال DWH أو Cube لكنه محدود
وذلك لصغر حجم ال Data Mart .



Types of Data Marts:-

Dependent Data Mart

DWH يتم تجميع كل data داخل ال DWH
ثم يتم تقطيع ال DWH إلى مجموعة من
Reports ثم يتم استخراج ال Data marts
وال Data marts هي Cubes .
وذلك لصغر حجم ال Data Marts .
← مهام الشركات مختلفة بهذه الطريقة .

Independent Data Mart

يتم تجميع ال data من各部门
DWH مختلطة ثم يتم تجميع كل ال DWHs
الصغيرة في نموذج DWH كبير تكون من
Data Marts و تكون ال DWHs الصغير
ال DWH .

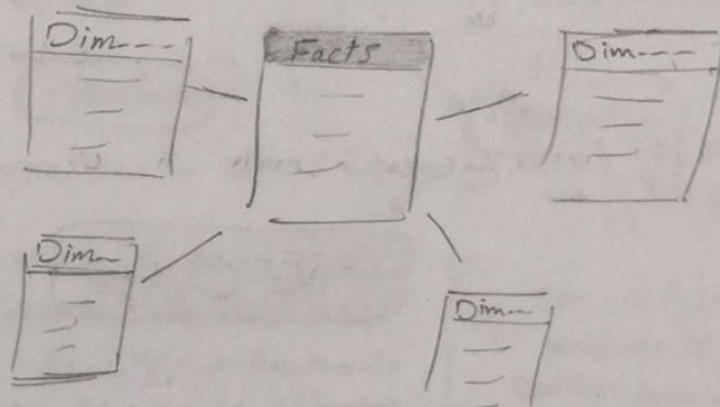
- هو عبارة عنsubset من DWH أو Data Mart
- Measures مجموعات قيم مترابطة يمكن حسابها
- Cube هو طريقة تحويل DWH أو Data Mart إلى OLAP

The Schemas

→ Star Schema

Fact Table هو جدول يحتوي على قيم

Dimensions Tables يحيط به

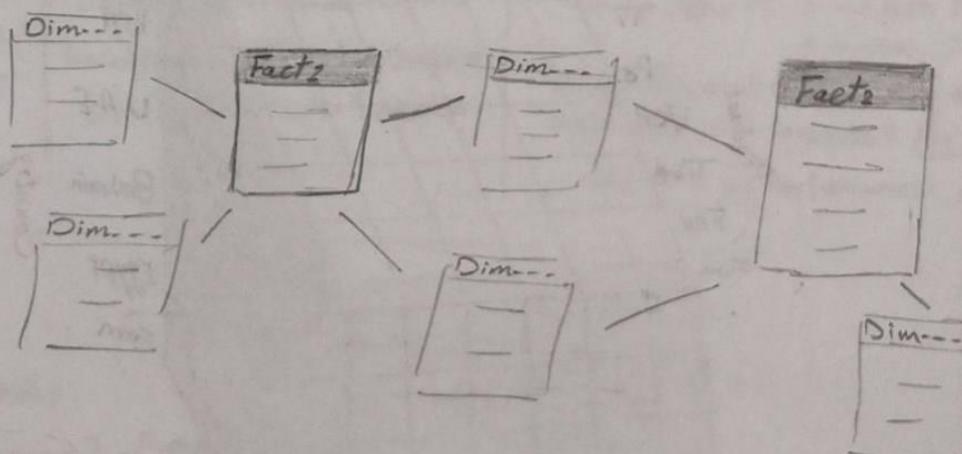


→ Galaxy Schema

هو عبارة عن نجمة

Fact Table يحيطونها

Dimensions Tables شكل نجم مع بعضه البعض



→ Snowflake Schema

Star Schema هو عبارة عن

Dimensions tables

other Dimensions و أبعاد

normalization كما

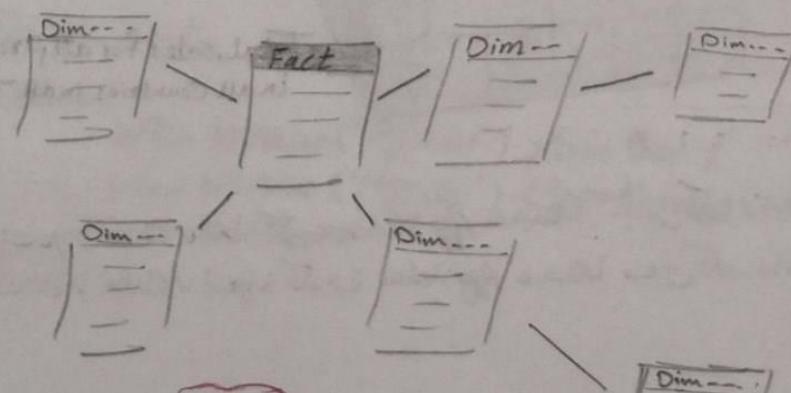
Cube داشتاييکه

و ليس Star Schema

كما يحتمل تحويل

normalization

Dimensions!



Star schema هي

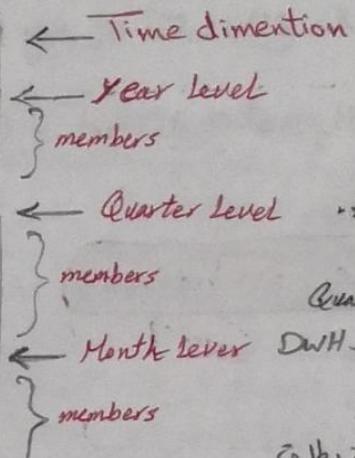
وليس Galaxy Schema

متعدد ال Dimensions

Schema

Level of Granularity

Time Dim	
• Year	1999 2000 2001
•• Quarter	Q1 Q2 Q3 Q4
••• Month	Jan Feb Mar etc



الدالة هو Level of granularity (الدقة) لـ Datas cube enter حاصل على data

month الـ level of granularity (الدقة) لـ DWH columns يمثل month levels

Quarter, Year الـ DB (الدقة) DWH columns يمثل month levels

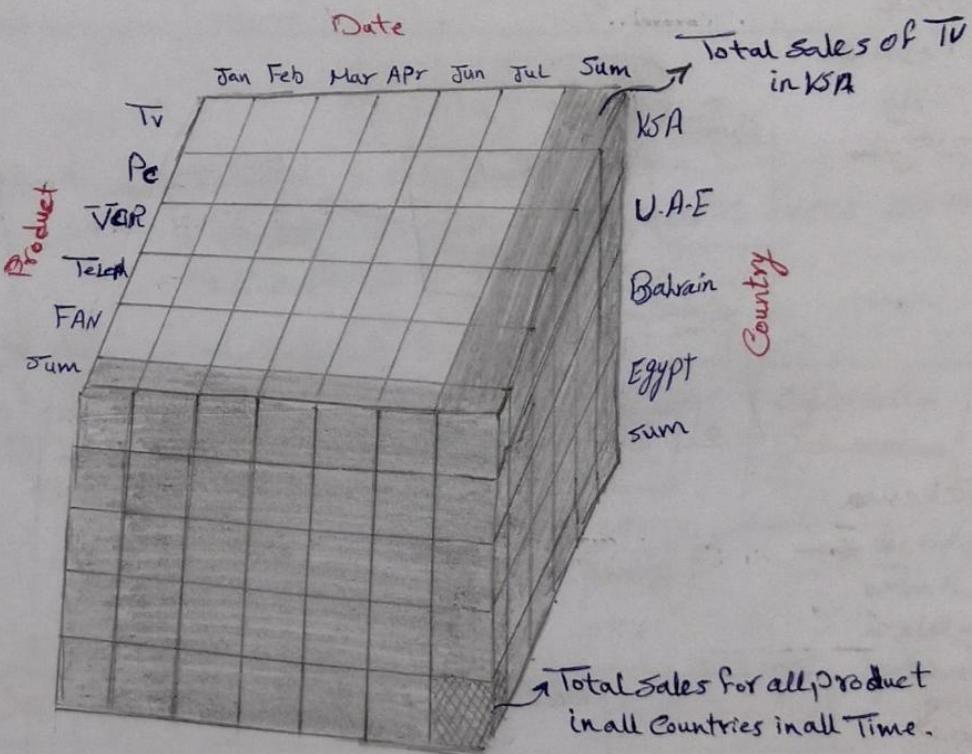
يطلق على هذه levels

في الـ DB العادي (1999-2001) و (Jan-Q1) member cells DWH (data cells)

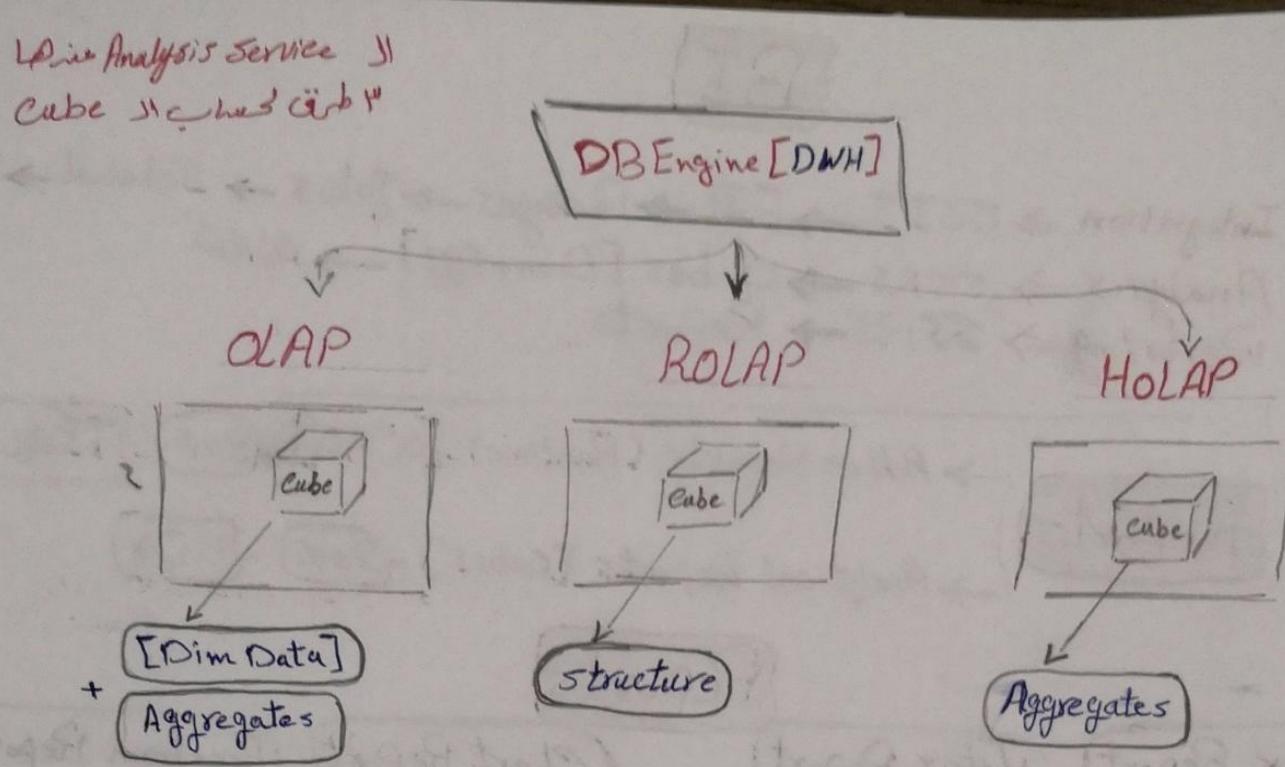
→ Hierarchy (الدقة) ←

(Year, Month) ← . مع greater levels ← هو عبارة عن hierarchy
 (Country, City) ←

The Cube



- Less details ← Less Agg Values ← التحليل عالي الـ Cube ← Rollup →
- More details ← More Agg Values ← التحليل منخفض الـ Cube ← drilldown →



* هو الطريقة الـ default data in cube يأخذ data copy من cube وينسب كل الـ Aggregates الى Combination ويتم حفظهم داخل الـ cube والـ performance يدور على جداً رغم الحجم الضخم لـ cube يفضل اما بين الـ low of granularity او بيس جيداً وهذا فنون زعيم كبيبة .

يتم مثل ال Structure Cube
measures فـ Σ متربياً عدد ال measures
الـ dimensions حجم ال cube وعند عمل
الـ Reports ال connect cube يحصل
على ال DWH ويحسب المـ measures
والـ Performance كـ KPIs.

- * يعمد الى measures ماتر الى Cube
- الـ Aggregates والـ DWH بـ data فـcube تكون الى Reports بيان
- الـ DWH لا data ويعصب measures الى
- * يعمد الى reports المبنية على data او كثرة الى Aggregates
- والـ Performance ينبع من data

Name : My sales

Value Expression: [Measures]. [Qty]

Goal Expression: 1000

Status Indicator: Traffic Light

Status Expression:- Page

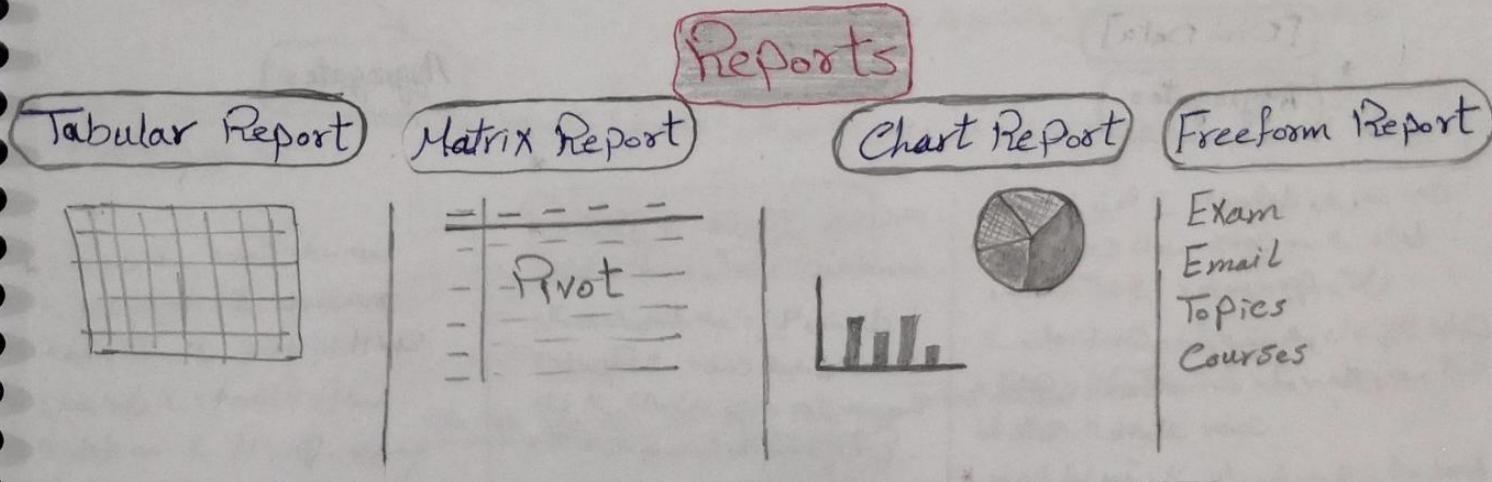
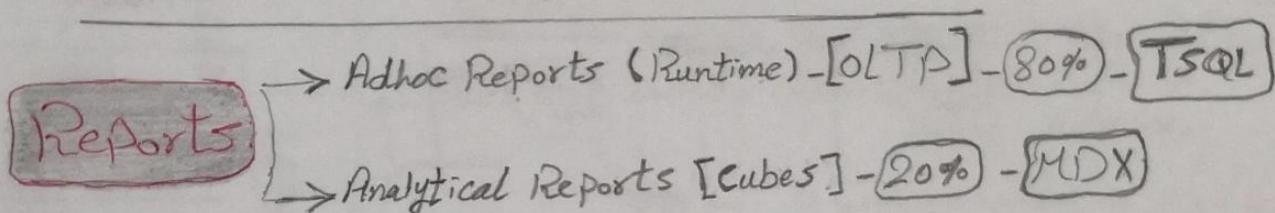
```
When KPIValue ("MySales") > 1000 then 1  
When KPIValue ("MySales") < 1000 then -1  
else 0
```

-End

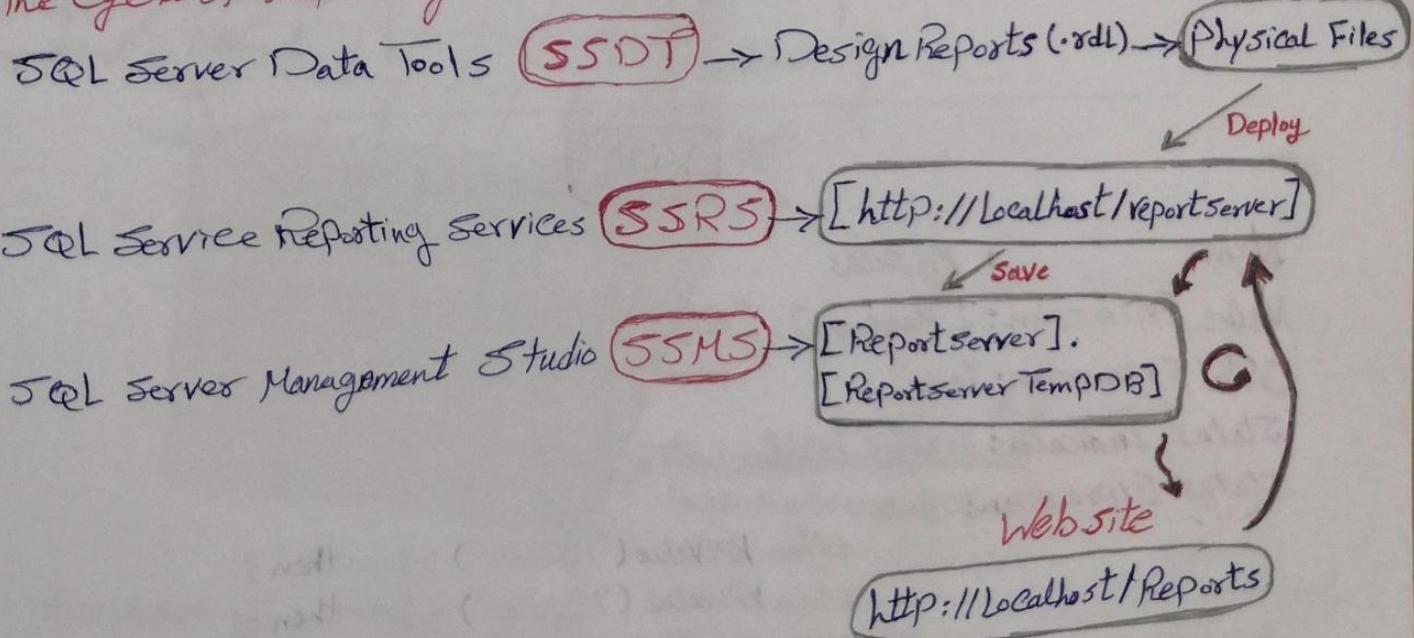
BI
Date
26/4/2023

BI

Integration \Rightarrow SSIS \Rightarrow ETL \Rightarrow Packages \Rightarrow Jobs \Rightarrow Schedule \Rightarrow DWH
Analysis \Rightarrow SSAS \Rightarrow Cubes [Dim+Agg] \Rightarrow OLAP
Reporting \Rightarrow SSRS \Rightarrow Reports



* The Cycle of Reporting :-



Top SQL Server Interview Questions for 2023

1. What is the Windows Authentication Mode in SQL Server?

This mode connects the server via a Windows account. The server uses the username and password for authentication. In this mode, SQL server authentication is disabled.

2. Give an example of a function in an SQL server that returns the first non-null expression from more than one column in arguments.

Select COALESCE(sid, sname, marks) from the student;

3. Explain the one-to-many relationship in the SQL Server database.

When a single column value in one table has a minimum of one dependent column value in some other table, a one-to-many relationship exists.

4. What is the significance of CHECK in SQL Server?

CHECK constraint limits the values that can be placed inside a table's column. This maintains integrity. The constraint is used column-wise to give specific values to that column. Example: CONSTRAINT CHK_Student CHECK (age<20)

5. How to find the 3rd highest marks from the Student table?

```
SELECT TOP 3 marks FROM (SELECT DISTINCT TOP 3 marks FROM student ORDER BY marks DESC) a ORDER BY marks
```

6. What is a trigger?

When a table event occurs, such as INSERT, DELETE, or UPDATE, triggers allow executing an SQL code batch. Triggers are managed by DBMS and can also execute stored procedures. [2] For example, when a record is inserted in a database table, a trigger can be set.

7. When can records be deleted from a view in SQL Server?

Records can be deleted in a 'simple' view as it contains data from one table only.

8. List down some of the features of MS SQL Server.

- It provides an easy and straightforward Syntax.
- MS SQL uses [transact SQL](#).
- Query optimization is not supported.
- The transaction process does not allow rollbacks
- Clustering is not supported
- Statements are executed serially.

9. Which command can be used to get the version of SQL Server? [3]

To get the version of SQL Server, use:

```
Select SERVERPROPERTY('productversion')
```

10. In SQL Server, what is a user defined function?

A user defined function allows users to write their logic as per need. The advantage is that it is not limited to pre-defined functions and writing functions, simply complex SQL code. The return type is a table or a scalar value.

```
Example: Create function sample(@sid int)
```

```
returns table
```

```
as
```

```
return select * from s where Id = @sid
```

11. Explain types of replication in SQL Server.

There are three types of replication as follows:

Transactional replication- It is a process of data distribution from publisher to subscriber.

Transactional replication can be used when data is changed frequently.

Merge replication- It groups the data to a single centralized database from various sources.

Merge replication is used in cases where central and branch databases need to update information simultaneously.

Snapshot replication- This replication is the best way to replicate data that changes infrequently, and it is easiest to maintain. Example: Snapshot replication can be used for lists that are updated once per day and needs to be distributed from main server to branch servers.

12. Define referential integrity.

Every foreign key value must have a corresponding primary key value. The maintenance of this consistency between foreign and primary keys is known as referential integrity.

13. What are TCL Commands? and List down the TCL Commands available on SQL Server?

TCL or Transactional Control Language commands are used to manage different transactions taking place in a database. The three TCL commands are as follows:

1. Rollback- This is used to restore the database to the last committed state
2. Save Tran- This saves the transaction, and the transaction can be rolled back to this point.
3. Commit- Saves the transaction permanently in the database

14. Write a SQL Server Query to get the letter 'e' in the name 'Jenna' from the student table.

Select CHARINDEX('e',NAME,0) from student where name='Jenna'

15. As a SQL developer, how will you ensure that SQL server-based applications and databases perform well?

The volume of data, type of information stored, and data to be accessed must be checked. When a system is being upgraded, the present data should be analyzed, and the methods of accessing data should be checked to help understand problem design. Keeping the information about data is necessary when using a new system.

16. When should Server-based cursors be used?

When you require to work on one record at any instance of time, instead of taking all the data from the table as bulk. Cursors' performance is affected when large volumes of data are present.

17. Tell us about the working of the FLOOR function.

FLOOR function rounds the given non-integer value to the previous least integer—for example, FLOOR(5.6) returns 5

18. What do you know about scheduled tasks in SQL Server?

Scheduled jobs or tasks automate processes that can be run at a prescribed time at a regular interval. By scheduling tasks, human intervention is reduced, and tasks can be carried out at any time in the order that the user wants.

19. Mention a query that returns the list of triggers in a database.

```
Select * from sys.objects where type='tr'
```

20. Differentiate between rollback and commit.

When COMMIT is executed, all statements between BEGIN and COMMIT become persistent to the database. Whereas, when ROLLBACK is executed, all statements between ROLLBACK and BEGIN are reverted to the state.

21. Explain how to create a table in SQL.

The following query is used to create a SQL table:

```
Create table name_of_table( column1 datatype, column2 datatype )
```

For example:

```
create table Student
```

```
(
```

```
    Name varchar(20),
```

```
    DOB date,
```

```
    Marks nvarchar(5),
```

```
    Subject varchar(20) )
```

22. What is the function of a foreign key in a database?

A foreign key is used to define a relationship between the parent and child table connected by columns. The foreign key is a constraint that ensures that the values of the child table appear in the parent table. The foreign key of one table is the primary key of the other, and a table can have several foreign keys. For example:

```
student {ID, Name, Age, Contact, Gender, Add}
```

```
teacher{Teach_ID, Name, ID}
```

Here, ID is the foreign key for the teacher table.

23. What is the importance of views in a database?

There are scenarios where we need to look for a view to getting the solution, such as:

1. Aggregating data for performance
2. Customizing the schema and data for a set of users
3. Controlling access to columns and rows of data

24. Tell us the steps to hide SQL Server Instances.

To hide the SQL Server Instances, we need to make changes in SQL Server Configuration Manager, and to launch it, the following steps are needed:

1. Select instance of SQL server
2. Select properties after right-clicking
3. Set Hide Instances to Yes and click on APPLY
4. Post changes, restart the instance of SQL Server

25. Explain the DBCC command and its use.

Database Consistency Checker (DBCC) checks the consistency of the database; It helps in reviewing and monitoring the maintenance of database, tables, and operation validation. For example:

- DBCC CHECKALLOC checks all pages in the database to ensure they are correctly allocated.
- DBCC CHECKDB makes sure that indexes are correctly linked in the tables of the database.
- DBCC CHECKFILEGROUP checks all file groups for damage.

26. Describe the SIGN function.

The SIGN function is used to specify a number as positive, zero, or negative. It returns the following: SIGN (number)

Returns – 1 if number <0, +1 if number>0 and 0 if number=0

27. Define alternate key.

When a table has more than one candidate key (i.e., candidate for primary keys), one becomes the primary key, and the rest are the alternate keys.

28. Define Join. What are the different types of joins?

Joins are used in SQL queries to describe how different tables are related. They also allow users to select data from one table depending on the data of the other table. The different types of joins are:

1. INNER Joins
2. OUTER Joins- LEFT OUTER, RIGHT OUTER, FULL OUTER
3. CROSS Joins

29. Tell about the use of UPDATE STATISTICS.

UPDATE STATISTICS is used to update information about the distribution of the key values for one or more statistic groups/collections in the indexed view or specified table.

30. Define Full backup.

The most common type of backup in SQL server is the complete backup of the database. It also includes part of the transaction logs for recovery.

31. In SQL, what is meant by the identity column?

In SQL, an identity column generates numeric values automatically. These columns need not be indexed, and we can define the start and increment value of the identity column.

32. Explain the UNIQUE KEY constraint.

The UNIQUE constraint maintains the uniqueness of records in the set of columns to ensure there are no duplicate values. This constraint enforces entity integrity.

33. Define the process of de-normalization.

The process of de-normalization adds redundant data to a database in order to enhance the performance. This technique moved from higher to lower normal forms of the database. This speeds up the database access.

34. Show how table type constraint can be applied to a table.

Alter Table Name_of_the_Constraint

Alter Table Constraint_1

35. Differentiate between derived persistent attribute and derived attribute.

A derived attribute is obtained from values of other existing columns as its values do not exist on their own. A derived attribute that can be stored is a derived persistent attribute.