ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Korszerű adatbázisok előadás 05

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Miről lesz szó?

Speciális adattípusok a T-SQL-ben

- Xml
- Json?
- Geometry
- Geography

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

XML – Extensible Markup Language

- Általános célú leíró nyelv
- A számítógépek számára is olvasható formátum
- Unicode karakterkészlet támogatása
- Platformfüggetlen
- Case-sensitive
- Fontosabb alkalmazások
  - **Különböző rendszerek, platformok közötti adatcsere**
  - Metadata adatok ábrázolása
  - Keresés a weben
  - Interaktív web oldalak

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

XML alapelemek

- Az adatokat < és > jelek közötti elemek jelölik (tag-ek): <elem>
- Az elemek neve, jelentésük és alkalmazásuk módja nem korlátozott, néhány formai megkötés kivételével
- Minden elemnek van egy záró párja: </ elem >, az adatokat a nyitó és záró párok közrefogják
- Az elemeknek lehetnek tulajdonságaik is, a tulajdonság értékei idézőjelek között vannak
- Az elemek egymásba is ágyazhatók, de nem lehet közöttük átfedés (jól formázottság)
- Az elemek helyzete rögzített (rendezettség)

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

XML – speciális karakterek

Karakter	Helyettesítő kód
&	&amp;
”	&quot;
<	&lt;
>	&gt;
’	&apos;

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

XML dokumentumok

- Első soruk mindig egy vezérlési utasítás (verzió + egyebek), pl: <?xml version= ”1.0” encoding= ” ISO-8859-15” ?>
- A dokumentum mindig fa-struktúrát követ – kötelező gyökérelem
- Azonos nevű, de különböző célú elemek az un. névterek (namespace) segítségével különböztethetők meg (előtagok): <prefix:elemnév> … </prefix:elemnév>
- Kommentek is elhelyezhetők: <!--komment -->
- Ha hiányzik a gyökér elem, akkor XML töredékről (fragment) beszélünk

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

XML Példa1

```
<Szallashely>
  <szallas.dbo.Szallashely>
    <SZALLAS_ID>1</SZALLAS_ID>
    <SZALLAS_NEV>Sába-Ház</SZALLAS_NEV>
    <HELY>Balaton-dél</HELY>
    <CSILLAGOK_SZAMA>0</CSILLAGOK_SZAMA>
    <TIPUS>vendégház</TIPUS>
    <ROGZITETTE>Béla</ROGZITETTE>
    <ROGZ_IDO>2016-02-28</ROGZ_IDO>
</szallas.dbo.Szallashely>
  <szallas.dbo.Szallashely>
    <SZALLAS_ID>2</SZALLAS_ID>
    <SZALLAS_NEV>Családi Ház</SZALLAS_NEV>
    <HELY>Balaton-dél</HELY>
    <CSILLAGOK_SZAMA>0</CSILLAGOK_SZAMA>
    <TIPUS>vendégház</TIPUS>
    <ROGZITETTE>Béla</ROGZITETTE>
    <ROGZ_IDO>2016-03-02</ROGZ_IDO>
</szallas.dbo.Szallashely>
 </Szallashely>
```

Elem-centrikus ábrázolás

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

XML – Példa2

```
<Szallashely>
  <szallas.dbo.Szallashely SZALLAS_ID="1" SZALLAS_NEV="Sába-Ház"  TIPUS="vendégház" />
  <szallas.dbo.Szallashely SZALLAS_ID="2" SZALLAS_NEV="Családi Ház„ TIPUS="vendégház" />
  <szallas.dbo.Szallashely SZALLAS_ID="3" SZALLAS_NEV="Fortuna Apartman" TIPUS="Apartman"/>
  <szallas.dbo.Szallashely SZALLAS_ID="4" SZALLAS_NEV="Fortuna panzió" TIPUS="panzió" />
  <szallas.dbo.Szallashely SZALLAS_ID="5" SZALLAS_NEV="Fortuna Panzió" TIPUS="panzió" />
  <szallas.dbo.Szallashely SZALLAS_ID="6" SZALLAS_NEV="Kentaur Hotel" TIPUS="Hotel"  />
  <szallas.dbo.Szallashely SZALLAS_ID="7" SZALLAS_NEV="Szieszta Apartmanház"TIPUS="Apartman" />
  <szallas.dbo.Szallashely SZALLAS_ID="8" SZALLAS_NEV="Hotel Három Hattyú" TIPUS="Hotel" />
  <szallas.dbo.Szallashely SZALLAS_ID="9" SZALLAS_NEV="Jáde panzió" TIPUS="panzió"/>
  <szallas.dbo.Szallashely SZALLAS_ID="21" SZALLAS_NEV="Müller Vendégház" TIPUS="vendégház" />
</Szallashely>
```

Attribútum-centrikus ábrázolás

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

XML dokumentum séma

- Az XML-dokumentum típusát határozza meg (elemek, tulajdonságok, felépítés)
- Adatcsere esetén felhasználható pl. a típus ellenőrzésére
- Több séma leíró nyelv is létezik, pl. XSD (XML Schema Definition)

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="brightstar">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="magnitude" type="xs:decimal"/>
      <xs:element name="distance" type="xs:integer"/>
    </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Relációs adatok XML-lé alakítása

FOR XML záradék – a lekérdezés eredménysorait XML-formátumra hozza

Formája:

SELECT …

FROM …

…

FOR XML AUTO | RAW | PATH

Példa:

```
SELECT [SZALLAS_ID]
      ,[SZALLAS_NEV]
      ,[TIPUS]
      ,[ROGZITETTE]
      ,[ROGZ_IDO]
      ,[CIM]
FROM [Szallashely]
FOR XML AUTO | RAW | PATH
```

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

FOR XML AUTO

Az elemek elrendezése automatikus, nem testreszabható.

Tulajdonságai:

- Több tábla esetén az elemek egymásba ágyazódnak
- A beágyazásnál fontos a SELECT-beli oszlopok sorrendje
- Az ELEMENTS kulcsszó segítségével elem-centrikus lesz az ábrázolás
- Az ORDER BY nélkül az elemek sorrendje nem előrejelezhető

Példa:

```
SELECT   f.FOGLALAS_PK,
         v.NEV,
         f.METTOL,
         f.FELNOTT_SZAM
FROM Foglalas f JOIN Vendeg v
   ON f.UGYFEL_FK = v.USERNEV
ORDER BY f.FOGLALAS_PK,v.NEV
FOR XML AUTO, ELEMENTS
```

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

FOR XML RAW

Az adatokat egy táblázathoz hasonló módon rendezi el

Tulajdonságai:

- Minden sorból egy row nevű elem lesz, amelynek attribútumai az oszlopok
- Az ELEMENTS kulcsszó itt is használható

Példa:

```
SELECT [SZALLAS_ID]
      ,[SZALLAS_NEV]
      ,[TIPUS]
      ,[ROGZITETTE]
      ,[ROGZ_IDO]
      ,[CIM]
FROM [Szallashely]
FOR XML RAW, ELEMENTS
```

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

FOR XML PATH

Lehetővé teszi az XML-formátum testreszabását

Tulajdonságai:

- Az elemek és attribútumok leírása XPath kifejezésekkel történik
- Alapesetben minden oszlopból egy elem lesz
- Attribútum-centrikus elrendezéshez a SELECT-beli oszlop alias elé @ karakter kell

Példa:

```
SELECT f.FOGLALAS_PK as [@azon],
       v.NEV,
       f.METTOL,
       f.FELNOTT_SZAM
FROM Foglalas f JOIN Vendeg v
       ON f.UGYFEL_FK = v.USERNEV
ORDER BY f.FOGLALAS_PK, v.NEV
FOR XML PATH('Foglalas'),
ROOT('Foglalások')
```

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

XML séma lekérdezése

FOR XML … , XMLSCHEMA

Formája:

SELECT …

FROM …

…

FOR XML AUTO | RAW | PATH

XMLSCHEMA(’ sémanév’)

Példa:

```
SELECT [SZALLAS_ID]
      ,[SZALLAS_NEV]
      ,[TIPUS]
      ,[ROGZITETTE]
      ,[ROGZ_IDO]
      ,[CIM]
FROM [Szallashely]
FOR XML AUTO, XMLSCHEMA('')
```

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

XML típusú változó deklarálása

DECLARE @változónév xml (xml típus)

- Típusos xml esetén a típus is megadható (zárójelben)
- A táblák xml oszlopai is hasonlóan deklarálhatók,
  pl: CREATE TABLE t
  (oszlop1 típus1,
  xmloszlop xml,
  …)
- Az xml változók és oszlopok XQuery-vel lekérdezhetők

Példa:

```
declare @x xml
set @x = cast(N'
<Foglalasok>
  <Foglalas FOGLALAS_PK="558"
METTOL="2016-04-06" FELNOTT_SZAM="2">
    <Vendeg NEV="Fő Nándor" />
  </Foglalas>
  <Foglalas FOGLALAS_PK="559"
METTOL="2016-04-06" FELNOTT_SZAM="2">
    <Vendeg NEV="Kelemen Áron" />
  </Foglalas>
</Foglalasok>' as xml)
```

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

XML típus - Megjegyzések

- Az xml típus nem támogatja a következő tábla-, illetve oszlopkényszereket: PRIMARY KEY, FOREIGN KEY, UNIQUE, COLLATE
- Meggondolandó az xml típusú oszlop külön táblába helyezése, amennyiben:
  - Indexelni szeretnénk az xml mező alapján, és az elsődleges kulcs nem a clustered index kulcsa
  - A táblának nincs elsődleges kulcsa
  - A táblának nincs clustered index kulcsa
  - Nem akarjuk, hogy a tábla scan lassú legyen az xml oszlop miatt

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Mikor érdemes xml típust használni?

- Nem strukturált vagy félig strukturált adatok esetén
- Platform független modell esetén
- Ha az adatok szerkezete nem ismert, vagy várhatóan gyakran változik
- Hierarchikus adatok esetén
- Ha azt szeretnénk, hogy jól formázott (well-formed), validált adatokkal dolgozzunk
- SOAP, ADO.NET vagy OLE DB-n keresztüli adatelérés esetén

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

XML adatok tárolási lehetőségei

- XML adattípus – Az adatok belső tárolása az xml-nek megfelelő módon történik (elemek, attribútumok, hierearchiák stb.)
- Leképezés az XML és a relációs tárolás között – Annotált séma esetén (AXSD) az XML egy vagy több tábla oszlopainak feleltethető meg. Ez a séma nem lehet rekurzív.
- Varchar(MAX) és Nvarchar(MAX) – Az adatoknak egy másolata tárolódik. Ez a tárolási mód többnyire speciális célú alkalmazások esetén hasznos.

Ha az xml adatokkal semmilyen műveletet nem végzünk, csak tároljuk őket, akkor megfontolandó az adatok szöveges tárolása (varchar(MAX) vagy nvarchar(MAX))

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

XML adattípus metódusai

- query(XQuery) –
  XML elemek, attribútumok lekérdezése
- value(XQuery, SQLType) –
  XML változó vagy oszlop skalár érték lekérdezése
- modify(XML_DML) –
  XML változó vagy oszlop értékek módosítása
- nodes(XQuery) as Table(Column)–
  Logikai másolatot készít XML csomópontokból
- exist(Xquery)—
  1-et ad vissza, ha az XQuery eredménye nem üres

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

XML adattípus metódusai - Példák

```
declare @x xml
set @x = cast(N'
<Foglalasok>
  <Foglalas FOGLALAS_PK="558"
METTOL="2016-04-06"
FELNOTT_SZAM="2">
    <Vendeg NEV="Fő Nándor" />
  </Foglalas>
  <Foglalas FOGLALAS_PK="559"
METTOL="2016-04-06"
FELNOTT_SZAM="2">
    <Vendeg NEV="Kelemen Áron" />
  </Foglalas>
</Foglalasok>' as xml)
```

```
SELECT
  @x.query('/Foglalasok/Foglalas/Vendeg’),
  @x.value('(/Foglalasok/Foglalas/@FOGLALAS_PK)[1]',
'int’),
   @x.exist('/Foglalasok/Foglalas[(@METTOL cast as
xs:date?) eq xs:date("2016-04-06")]')
```

```
SELECT fogl.value('@FOGLALAS_PK', 'int’) as id
FROM @x.nodes('/Foglalasok/Foglalas’)
        as t(fogl)
```

```
declare @value nvarchar(50) = 'Kiss Béla'
set @x.modify('replace value of
(/Foglalasok/Foglalas/Vendeg/@NEV)[1] with
sql:variable("@value")')
```

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

XQuery

Standard nyelv az XML dokumentumokon való navigálásra

Tulajdonságai:

- Az SQL Server nem minden XQuery lehetőséget támogat
- Case sensitive
- Az XQuery által visszaadott eredmény tartalmazhat elemi és komplex értékeket is
- Tartalmaz kétirányú elágazási lehetőséget (if – then –else)

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

XQuery függvények

Függvény kategóriák

- Numerikus, pl. round()
- Logikai, not(), true(), false()
- Szöveg, pl. concat()
- Aggregáló, pl. count(), sum()
- Csomóponti, pl. local-name()
- Adat eléresi, data(), string()
- SQL Server kiterjesztés, sql:column(), sql:variable()

Példa:

```
DECLARE @x AS XML;
SET @x=N'
<root>
 <a>1<c>3</c><d>4</d></a>
 <b>2</b>
</root>';
SELECT
 @x.query('*') AS Complete_Sequence,
 @x.query('data(*)') AS Complete_Data,
 @x.query('data(root/a/c)') AS
Element_c_Data;
```

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

XQuery navigáció

Teljes elérési út: Node-name/child::element-name[@attribute-name=value]

A navigáció iránya (Axis) mellett szűrhetők a csomópontok (Node test), feltételek adhatók meg az attribútumokra (Predicates)

Nav. Irány	Leírás
child::	Az aktuális csomópont gyermekét adja vissza (alapértelmezett irány)
descendant::	Az aktuális csomópont összes leszármazottját adja vissza
self::	Az aktuális csomópont
@	Attribútum
parent::	A szülő csomópontot adja vissza

Összehasonlító operátorok	Érték
=	eq
<	lt
>	gt
!=	ne

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

XPath

Az XQuery-hez hasonló, XML-navigálást lehetővé tevő nyelv

Tulajdonságai:

- Az Xquery-nél régebbi, korlátozottabb funkcionalitás
- Egy XPath kifejezés eredménye lehet csomópontok halmaza, logikai érték, szöveg vagy szám
- Az elérési út / jelekkel elválasztott lépésekből áll
- Kétféle elérési út: abszolút és relatív
- Minden lépés 3 részből áll: irány::csomópont típus [feltétel]

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

FLWOR

FOR-LET-WHERE-ORDER BY-RETURN – for each jellegű XQuery ciklus

```
DECLARE @x AS XML;
SET @x = N'
<CustomersOrders>
 <Customer custid="1">
 <companyname>Customer NRZBB</companyname>
 <Order orderid="10692">
 <orderdate>2007-10-03T00:00:00</orderdate>
 </Order>
 <Order orderid="10702">
 <orderdate>2007-10-13T00:00:00</orderdate>
 </Order>
 <Order orderid="10952">
 <orderdate>2008-03-16T00:00:00</orderdate>
 </Order>
 </Customer>
</CustomersOrders>';
```

Példa:

```
SELECT @x.query('for $i in
CustomersOrders/Customer/Order
 let $j := $i/orderdate
 where $i/@orderid < 10900
 order by ($j)[1]
 return
 <Order-orderid-element>
 <orderid>{data($i/@orderid)}</orderid>
 {$j}
 </Order-orderid-element>')
 AS [Filtered, sorted and reformatted
orders with let clause]
```

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

XML Index

Az xml típusú oszlopokhoz létrehozható index típus

Tulajdonságai:

- Minden tag, érték és útvonal indexelődik
- Kétféle típus
  - Primary XML Index: az első index az xml oszlopon, előfeltétele az elsődleges kulcs alapján képzett clustered index
  - Secondary XML Index: a második és további indexek, három típusa: PATH, VALUE és PROPERTY másodlagos indexek

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

OPENXML() függvény

Az XML formátumot táblává alakítja

Formája:

```
OPENXML( idoc int [ in] ,
rowpattern nvarchar [ in ] , [
flags byte [ in ] ] )
[ WITH ( SchemaDeclaration |
TableName ) ]
```

Paraméterek:

- Idoc: a dokumentum belső azonosítója
- Rowpattern: Xpath kifejezés (a sorokká alakítandó csomópontok leírása)
- Flags: az XML és a tábla közötti megfelel-tetés módját adja meg

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

OPENXML() függvény (folyt.)

Megjegyzések:

- A belső azonosító az **sp_xml_preparedocument** tárolt eljárás meghívása után jön létre
- A flags paraméter opcionális, lehetséges értékei:
  - 0 – attribútum-centrikus megjelenítés (alapértelmezett)
  - 1– attribútum-centrikus megjelenítés + XML_ATTRIBUTES
  - 2—elem-centrikus megjelenítés
  - 8—elem-centrikus megjelenítés + XML_ATTRIBUTES
- Ha már nincs szükség a belső ábrázolásra, akkor az a s**ys.sp_xml_removedocument** tárolt eljárás meghívásával szüntethető meg

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

OPENXML() függvény – Példa

```
DECLARE @idoc INT, @doc
NVARCHAR(1000)
SET @doc =N'
<Foglalasok>
  <Foglalas FOGLALAS_PK="558"
METTOL="2016-04-06"
FELNOTT_SZAM="2">
    <Vendeg NEV="Fő Nándor" />
  </Foglalas>
  <Foglalas FOGLALAS_PK="559"
METTOL="2016-04-06"
FELNOTT_SZAM="2">
    <Vendeg NEV="Kelemen Áron"
/>
  </Foglalas>
</Foglalasok>'
```

```
EXEC sp_xml_preparedocument
@idoc OUTPUT, @doc
```

```
SELECT * FROM OPENXML (@idoc,
'/Foglalasok/Foglalas/Vendeg',2)
WITH (FOGLALAS_PK int '../@FOGLALAS_PK',
      METTOL date '../@METTOL',
      FELNOTT_SZAM int '../@FELNOTT_SZAM',
      NEV nvarchar(50) '@NEV')
```

```
EXEC sys.sp_xml_removedocument
@DocHandle
```

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

JSON – Javascript Object Notation

A Javascript nyelvből kialakult, szöveg alapú, szabványos adatcsere formátum

Jellemzői:

- Az XML alternatívája
- Elsősorban strukturált adatok tárolására, továbbítására szolgál
- Alapelemei a kulcs: érték párok
- A formátumot a JSON-séma írja le
- A T-SQL-ben nincs explicit JSON típus

```
[{
  "FOGLALAS_PK":558,
  "METTOL":"2016-04-06",
  "FELNOTT_SZAM":2,
  "Vendeg":[{"NEV":"Fő Nándor"}]
  },{
  "FOGLALAS_PK":559,
  "METTOL":"2016-04-06",
  "FELNOTT_SZAM":2,
  "Vendeg":[{"NEV":"Kelemen Áron"}]
}]
```

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Relációs adatok JSON formátumra alakítása

FOR JSON záradék – a lekérdezés eredménysorait JSON-formátumra hozza

Formája:

SELECT …

FROM …

…

FOR JSON AUTO | PATH

Az ELEMENTS itt nem használható, a ROOT viszont igen (lásd. FOR XML)

Példa:

```
SELECT [SZALLAS_ID]
      ,[SZALLAS_NEV]
      ,[TIPUS]
      ,[ROGZITETTE]
      ,[ROGZ_IDO]
      ,[CIM]
FROM [Szallashely]
FOR JSON AUTO | PATH
```

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

json_query, json_value, json_modify, isjson

JSON formátumú adatokon értelmezett műveletek

- json_query(kifejezés, útvonal) – lekérdezés json formátumból
- json_value(kifejezés, útvonal) – érték lekérdezése json formátumból
- json_modify(kif, útv, új érték) – érték módosítása
- isjson(kifejezés) – json formátum ellenőrzése

Példák:

```
DECLARE @js nvarchar(MAX)
SET @js =
N'[{"FOGLALAS_PK":558,"METTOL":"2016-
04-06", "FELNOTT_SZAM":2,
"Vendeg":[{"NEV":"Fő Nándor"}]}]'
SELECT ISJSON(@js)
SELECT JSON_QUERY(@js, '$[0]')
SELECT JSON_VALUE(@js, '$[0].METTOL’)
SELECT @js = json_modify(@js,
'$[0].FOGLALAS_PK',500)
```

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

OPENJSON() függvény

A JSON formátumot táblává alakítja:
OPENJSON(kifejezés, útvonal) [WITH (oszlopdeklarációk)]

```
DECLARE @js nvarchar(MAX)
SET @js = N'[
  {
    "Order": {
      "Number":"SO43659",
      "Date":"2011-05-31T00:00:00"
    },
    "AccountNumber":"AW29825",
    "Item": {
      "Price":2024.9940,
      "Quantity":1
    }
  }]'
```

Példa:

```
SELECT * FROM OPENJSON ( @js )
WITH (
    Number VARCHAR(200) '$.Order.Number',
    Date DATETIME '$.Order.Date',
    Customer VARCHAR(200) '$.AccountNumber',
    Quantity INT '$.Item.Quantity',
    [Order] NVARCHAR(MAX) AS JSON
)
```

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Geography

Ellipszoid koordinátarendszerben lévő adatokat tárol
(pl. GPS szélességi és hosszúsági koordináták)

- Eredetileg .NET CLR adattípus
- OGC (Open Geospatial Consortium) szabvány
- Műveletek (sok van!) pl:
  - STArea() – területet számol
  - STDifference() – különbséget számol
  - STContains() – tartalmazást ellenőriz

Példák:

```
DECLARE @g geography
DECLARE @h geography
DECLARE @i geography
SET @g = geography::STGeomFromText('POLYGON((-
122.358 47.653, -122.348 47.649, -122.348
47.658, -122.358 47.658, -122.358 47.653))',
4326);
SET @h = geography::STGeomFromText('LINESTRING(-
122.360 47.656, -122.343 47.656)', 4326);
SET @i = geography::STGeomFromText('POINT(-
122.34900 47.65100)', 4326);
SELECT @g.STArea()
SELECT @g.STDifference(@h).ToString()
SELECT @h.STDistance(@i)
```

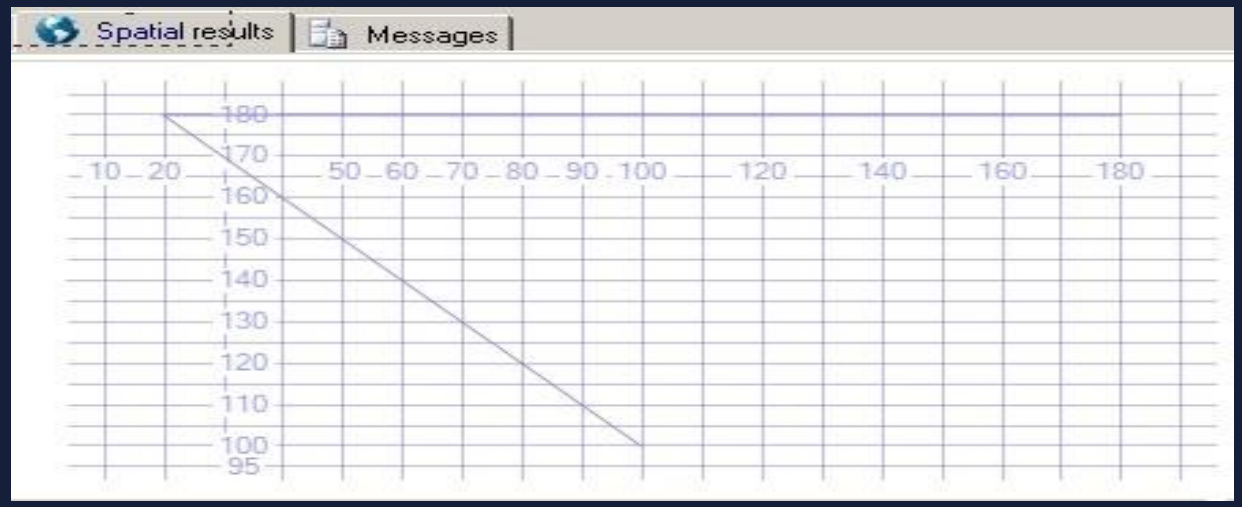
Geometry

Térbeli objektumok létrehozására, kezelésére alkalmas adattípus, amely euklédeszi koordinátarendszert használ.

- Eredetileg .NET CLR adattípus
- OGC (Open Geospatial Consortium) szabvány
- Műveletek (sok van!) pl:
  - STArea() – területet számol
  - STDifference() – különbséget számol
  - STContains() – tartalmazást ellenőriz

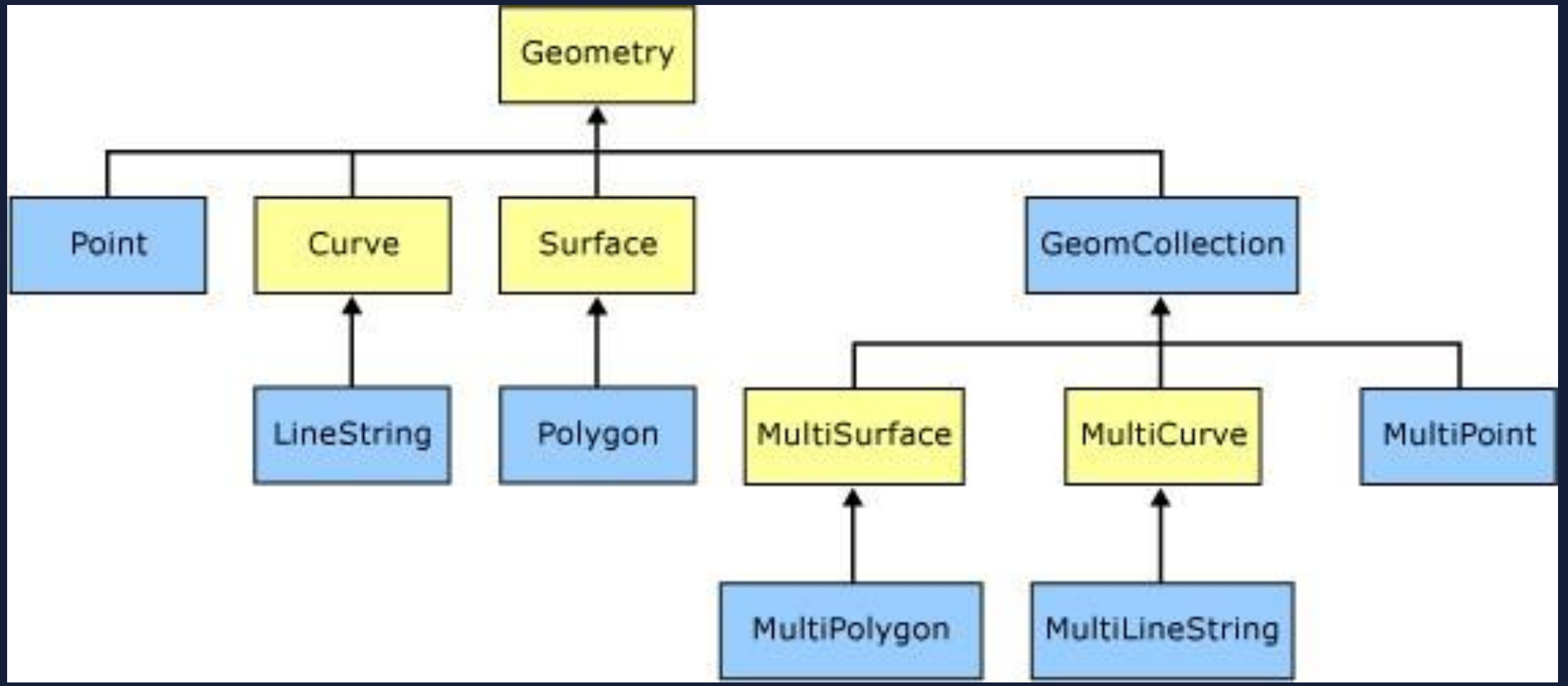
Példa:

```
DECLARE @g geometry;
SET @g =
geometry::STGeomFromText('LINESTRING (100
100, 20 180, 180 180)', 0);
SELECT @g;
```



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Geometry és Geography objektumok



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Uniqueidentifier

16 bájtos GUID (Globally Unique Identifier)

Tulajdonságai:

- Értéket kaphat
  - NEWID függvénnyel
  - NEWSEQUENTIALID függvénnyel
  - Karaktersorozat konvertálásával
- Csak összehasonlító műveletei vannak

Példák:

```
DECLARE @myid uniqueidentifier
= NEWID()
SELECT CAST(@myid  AS
char(255)) AS id
```

```
DECLARE @ID NVARCHAR(max) =
N'0E984725-C51C-4BF4-9960-
E1C80E27ABA0wrong';
SELECT @ID, CAST(@ID AS
uniqueidentifier) AS
TruncatedValue;
```

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Hierarchyid

Változó hosszúságú adattípus, amely a hierarchiában lévő pozíciót tárolja

Tulajdonságai:

- Összehasonlításkor először a szint számít
- Az ilyen típusú oszlop automatikusan nem jelenít meg egy fa struktúrát
- Maximális méret 892 B
- Átkonvertálható string vagy varbinary típusokra

Példa:

```
CREATE TABLE #SimpleDemo
( Level hierarchyid NOT NULL, Location
nvarchar(30) NOT NULL, LocationType
nvarchar(9) NULL );
```

```
INSERT xSimpleDemo VALUES
 ('/1/', 'Europe', 'Continent'), ('/2/', 'South America',
'Continent'), ('/1/1/', 'France', 'Country'), ('/1/1/1/',
'Paris', 'City')
```

```
SELECT CAST(Level AS nvarchar(100)) AS
[Converted Level], *
FROM #SimpleDemo ORDER BY Level
```

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Sql_variant

Különböző típusú adatokat képes pl. egy oszlopon belül tárolni

Tulajdonságok

- Mérete max. 8000 B lehet
- Műveletek végzése előtt át kell konvertálni a megfelelő típusra
- Lehet default értéke, lehet NULL
- Szerepelhet kulcsok oszlopai között
- Nem minden SQL típust támogat
- ODBC kapcsolattal nem kompatibilis

Példa:

```
CREATE TABLE #tableA(colA sql_variant, colB
INT)

INSERT INTO #tableA
values ( CAST(46279.1 as decimal(8,2)), 1689)

SELECT
SQL_VARIANT_PROPERTY(colA,'BaseType') AS
'Base Type’
FROM #tableA
WHERE colB = 1689
```

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

SQL Server Import and Export

Lehetővé teszi adatok átvitelét különböző típusú adatforrások között

Támogatott adatforrások

- Ismertebb vállalati adatbázis rendszerek (MS SQL, Oracle, IBM…)
- Szöveges fájlok
- Access, Excel,
- Azure adatforrások
- Open source adatbázisok (MySQL, Postgre SQL)
- Egyéb rendszerek