

# Adatbázisok

Gyakorlat 01 - *SQL alapismeretek I.*  
(Egyszerű, egytáblás lekérdezések. *SELECT...FROM...WHERE*,  
*Kifejezések, függvények*)

# Gyakorlatok menete

A gyakorlat elején (5 perc)

- Bejelentkezés a Moodle-rendszerbe
- Opcionálisan teszt/zh feladatok megoldása
- Elméleti összefoglaló, gyakorló feladatok megoldása közösen (15-30 perc)
- Feladatmegoldás, problémás feladatok megbeszélése (60-70 perc)

# Részvétel, órai munka

## Részvétel

- A részvételt a jelenléti íven kell bejelölni minden egyes alkalommal
- A részvétel megtagadható, ha valaki 15 percnél többet késik, illetve az óra befejezése előtt több, mint 15 perccel önkényesen távozik (kivéve, ha az illető az órán az összes feladatot megoldja, és a megoldást bemutatja)
- Online részvétel – **előzetes egyeztetés alapján** – félévente egyszer lehetséges
- Az adott gyakorlat más csoportban való teljesítése félévente 2 alkalommal lehetséges, amennyiben van elég férőhely, és **az illető legkésőbb az óra elején ezt jelzi**

## Órai munka

- Elvárás, hogy a hallgató a tananyaggal foglalkozzon, és
- Legalább 5 feladat megoldását vagy a próbálkozást az óra végéig feltöltse

# Bónusz pontok

A féléves aktivitás alapján max. 10% bónusz szerezhető

- ☐ Órai munka (egyedi ötlet/megoldás, megoldott feladatok száma) – max. 5%
- ☐ Részvétel és aktivitás (Kahoot, Teams) előadáson – max. 5%

# Mivel fogunk dolgozni az első néhány gyakorlaton?

Azure Data Studio

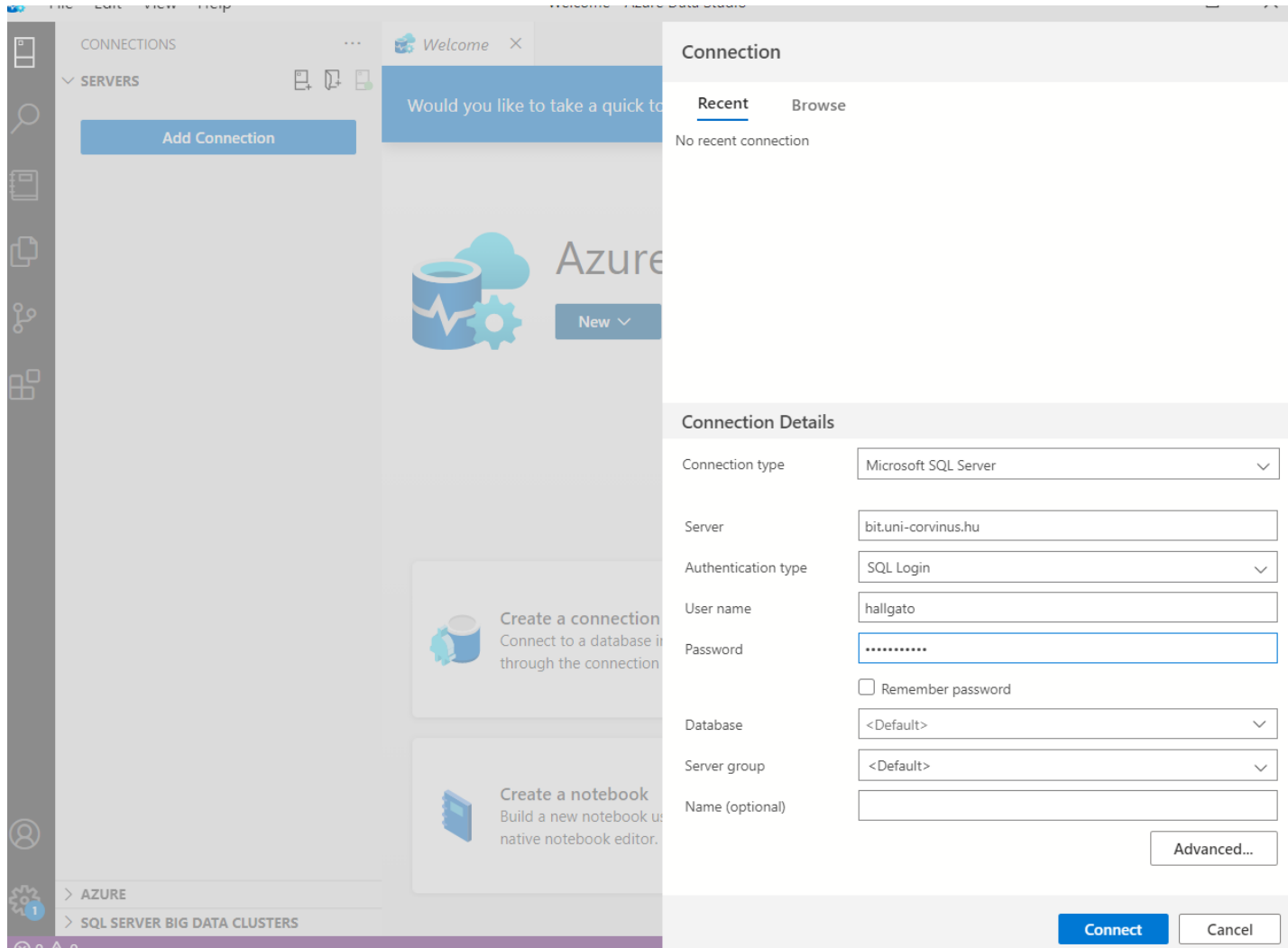
## Azure Data Studio – tantermi gépeken rajta van

- Az SQL Server-hez való csatlakozást, adatbázisokból való lekérdezéseket és egyéb tevékenységeket lehetővé tevő alkalmazás
- Letölthető a következő linkről:

[Download and install Azure Data Studio - Azure Data Studio | Microsoft Learn](#)

- Windows, Linux és Mac alatt is működik
- Alternatívák:
  - SSMS (SQL Server Management Studio)
  - SQLCMD parancssori eszköz (Linux alatt is létezik hasonló)
  - Azure lekérdező felülete
  - Visual Studio

# Azure Data Studio– csatlakozás az SQL szerverhez



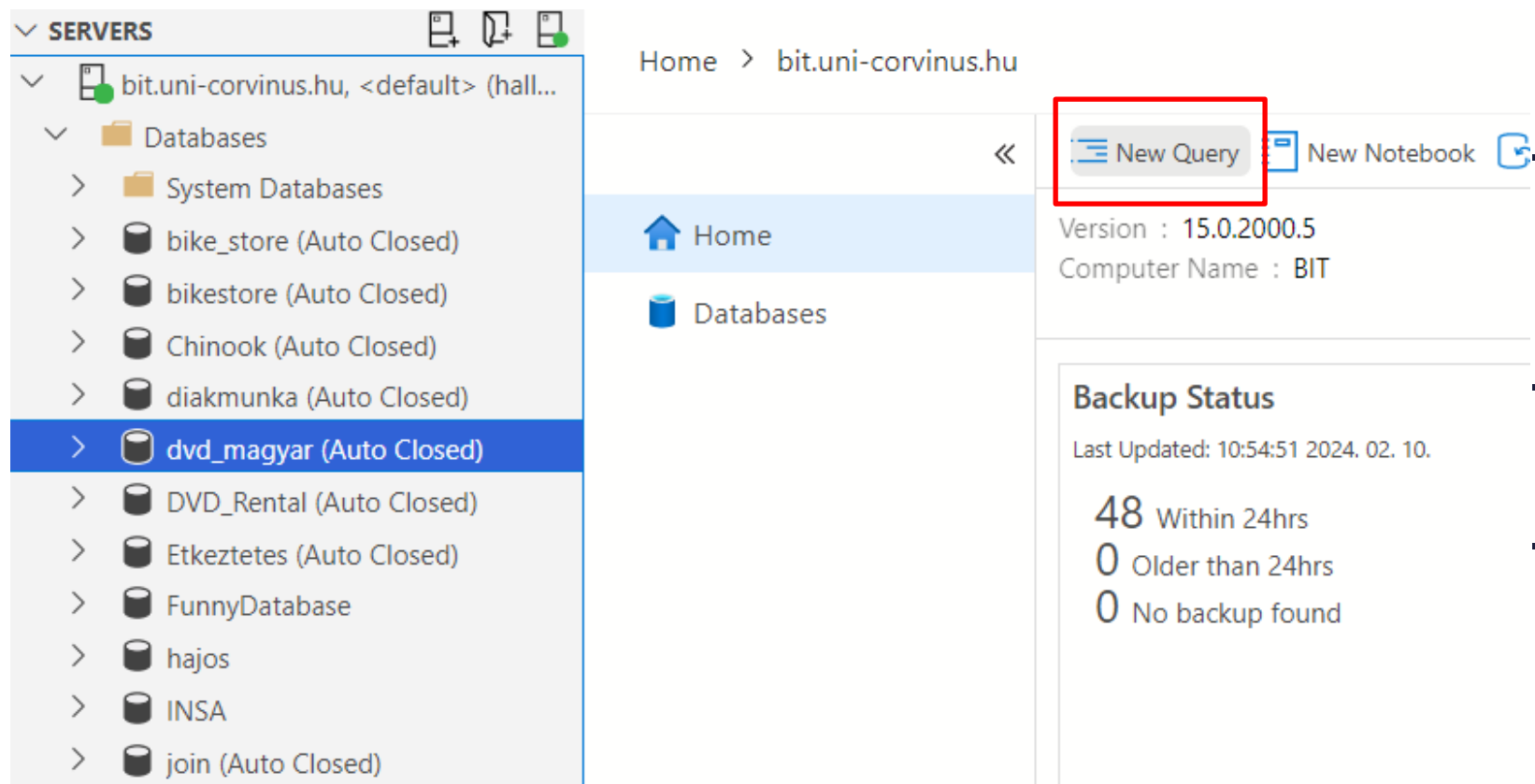
Add connection vagy Create a connection, majd

Connection Details:

- Server name: bit.uni-corvinus.hu
- Authentication: SQL Login
- User name: hallgato
- Password: Password123

Otthonról való eléréshez corvinus VPN lehet szükséges

# Azure Data Studio – csatlakozás meglévő adatbázishoz



A SERVERS részben bontsuk ki a Databases fület!

A listában kattintsunk rá az adatbázis nevére!

Kattintsunk a New Query ikonra!

Az első gyakorlaton a dvd\_magyar adatbázishoz kell csatlakozni



# SQL (Structured Query Language)

BCE

- ☐ ANSI és ISO standard, de nagyon megengedő → eltérő nyelvjárások is megfelelnek a szabványnak
- ☐ Nem procedurális, hanem deklaratív nyelv
  - ☐ A végrehajtás módját az adatbáziskezelő rendszer részeként a lekérdezés optimalizáló határozza meg
- ☐ Strukturált lekérdező nyelv, de nem csak lekérdezésre alkalmas:
  - ☐ DDL: data definition language (CREATE, ALTER, DROP)
  - ☐ DML: data manipulation language (INSERT, UPDATE, DELETE)
  - ☐ **DQL: data query language (SELECT)**
  - ☐ TCL: transaction control language (COMMIT, ROLLBACK, SAVEPOINT)
  - ☐ DCL: data control language (GRANT, REVOKE)
  - ☐ További típusok: pl. procedurális nyelvi elemek, stb.

- ☐ Nem case-sensitive, kis és nagybetűk szabadon használhatóak
- ☐ Az utasítás bárhol (két szó között) több sorba törhető, indentálható
- ☐ Az objektumok neve nem tartalmazhat különleges karaktert, szóköz helyett alulvonást használunk (\_)
- ☐ Gyakoribb adattípusok kezelése
  - ☐ Szöveg konstans megadása egyszeres aposztrófok között lehetséges, pl: 'Adatbázisrendszerek'
  - ☐ Dátum konstans megadásához is aposztróf használata szükséges, pl: '2020.02.11'
  - ☐ Tört számokban tizedes pont van, pl: 15.33

# A SELECT utasítás felépítése

Az SQL lekérdező utasítása, alapformája a következő:

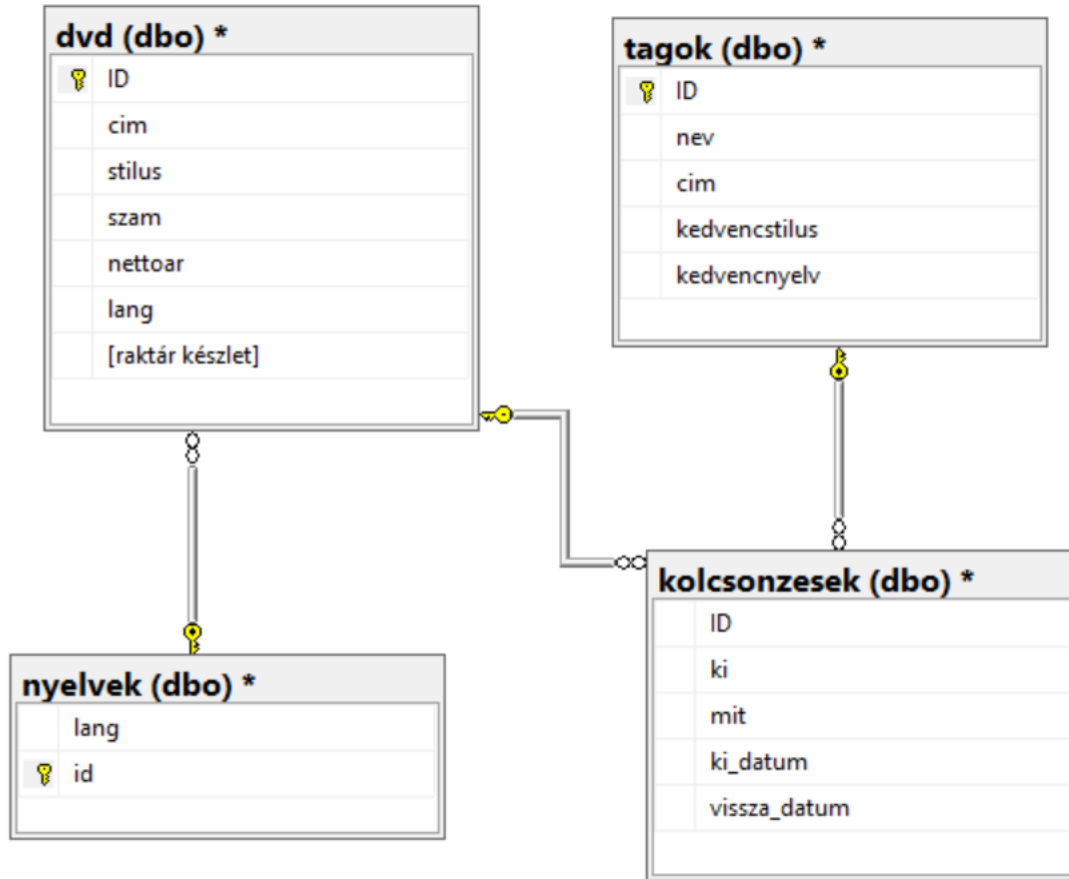
<b>SELECT...</b>	<b>// oszlopok kiválasztása</b>
<b>FROM ...</b>	<b>// táblák kiválasztása</b>
<b>WHERE ...</b>	<b>// szűrőfeltétel megadása a sorokra</b>
<b>GROUP BY ...</b>	<b>// csoportosítás</b>
<b>HAVING ...</b>	<b>// szűrőfeltétel a csoportokra</b>
<b>ORDER BY...</b>	<b>// sorbarendezés</b>

# A DVD adatbázis felépítése

BCE

Az adatbázis 4 táblából áll:

1. kölcsönzések
2. dvd
3. tagok
4. nyelvek



# SELECT ... FROM BCE

SELECT \*

FROM táblanév

-- a tábla összes oszlopát listázza

SELECT oszloplista

FROM táblanév

-- csak a felsorolt oszlopokat listázza

SELECT DISTINCT oszlopnév

FROM táblanév

-- az adott oszlop különböző értékeit listázza

# SELECT ... FROM példák

BCE

1. Listázzuk az adatbázisban lévő DVD-k minden adatát!

```
SELECT *  
FROM dvd
```

2. Listázzuk a DVD-k címeit és nettó árait!

```
SELECT cim, nettoar  
FROM dvd
```

3. Milyen stílusú DVD-k kölcsönözhetők?

```
SELECT DISTINCT stilus  
FROM dvd
```

# Származtatott oszlopok, oszlopnevek

A SELECT utáni oszloplistában származtatott oszlopok is szerepelhetnek:

- Oszlopokból álló kifejezések
- Függvények -- később

Az új vagy meglévő oszlopoknak saját nevet is adhatunk az AS kulcsszóval.  
Az oszlopnév aposztrófok vagy szögletes zárójelek közé kerül

Példák:

1. DVD-k címe és bruttó ára (nettoar\*1,27):

```
SELECT cim, nettoar*1.27 AS 'bruttoar'  
FROM dvd
```

2. A DVD-k címe és stílusa kötőjellel elválasztva:

```
SELECT cim+'-'+stilus AS [Cím]  
FROM dvd
```

# SELECT...FROM...WHERE

BCE

A WHERE kulcsszó után megadhatunk egy logikai kifejezést, mint szűrőfeltételt. A feltétel lehet egyszerű vagy állhat több részből is, logikai operátorokkal (pl. AND, OR) összekötve

Példa:

Listázzuk azon DVD-k adatait, amelyek stílusa történelem, és nettó áruk 3000 felett van!

```
SELECT *  
FROM dvd  
WHERE stilus= 'történelem' AND nettoar > 3000
```



# SELECT...FROM...WHERE (folyt)

BCE

A WHERE feltétel megfogalmazásában hasznos a következő, ún. predikátumfüggvények használata:

- BETWEEN – intervallumba esés leírására
- IN – halmazba tartozás leírására
- LIKE – helyettesítő karakterekkel való keresés leírására

Példák:

1. Listázzuk azon dvd-k adatait, amelyek nettó ára 3000 és 4000 között van!
2. Melyek azok a DVD-k, amelyek stílusa történelem, irodalom, vagy földrajz?
3. Melyek azok a dvd-k, amelyek címe K-betűvel kezdődik?

# SELECT...FROM...WHERE (folyt)

1. 

```
SELECT *  
FROM dvd  
WHERE nettoar BETWEEN 3000 AND 4000
```
2. 

```
SELECT *  
FROM dvd  
WHERE stilus IN ('történelem', 'irodalom', 'földrajz')
```
3. 

```
SELECT *  
FROM dvd  
WHERE cim LIKE 'K%'
```

# Kifejezések

Az egyszerű kifejezések konstansokat, változókat\*, oszlopneveket és függvényeket tartalmazhatnak, pl:

- 'Dr.' (szöveges konstans)
- Nettóbér (oszlopnév)
- YEAR('2010.01.01') (függvény, dátum konstans)

Az összetett kifejezések operátorokat is tartalmazhatnak, pl:

- 'Dr. ' + Vezetéknév + ' ' + Keresztnév (összefűzés)
- Nettóbér \* 1.27 (szorzás)

A kifejezések mindig egy értéket adnak vissza

\* Később tanuljuk

# Speciális kifejezés - CASE

CASE – többirányú elágazás megvalósítása, két formája van

CASE

WHEN feltétel<sub>1</sub> THEN kifejezés<sub>1</sub>  
WHEN feltétel<sub>2</sub> THEN kifejezés<sub>2</sub>

...

WHEN feltétel<sub>n</sub> THEN kifejezés<sub>n</sub>  
[ELSE kifejezés]

END

CASE kifejezés

WHEN érték<sub>1</sub> THEN kifejezés<sub>1</sub>  
WHEN érték<sub>2</sub> THEN kifejezés<sub>2</sub>

...

WHEN értékn THEN kifejezésn  
[ELSE kifejezés]

END

Pl:

CASE WHEN nettoar > 1500 THEN 'drága'  
WHEN nettoar < 500 THEN 'olcsó'  
ELSE 'reális'

END

Pl:

CASE kategória  
WHEN 'R' THEN 'Road'  
WHEN 'M' THEN 'Mountain'  
ELSE 'Other'

END

# Konstansok

Olyan szimbólumok, amelyek valamilyen adat értékét ábrázolják. Az ábrázolás módja az adat típusától függ. Gyakoribb konstansok és típusok\*:

Konstans	Típus	Példa
Szöveges konstans	varchar(x) (x: a szöveg max. hossza)	'Budapest'
Unicode szöveges konstans	nvarchar(x) (x: a szöveg max. hossza)	N'Budapest'
Egész konstans	int	25
Bit konstans	bit	1
Decimális konstans	decimal(x, y) (x: a számjegyek száma, y: a tizedesjegyek száma)	12.45
Dátum/Idő konstans	date, datetime, time	'2012.01.15' '2020.02.11 22:11:33' '06:12:10'

\* A típusok közül csak a leggyakrabban használtak szerepelnek

# Operátorok

Az operátorok egy vagy több kifejezésen értelmezett szimbólumok („műveletek”)

A gyakoribb operátor típusok és operátorok:

Típus	Operátorok	Megjegyzés
Aritmetikai operátorok	+, -, *, /, %	?: az egész osztás maradéka
Logikai operátorok	NOT, AND, OR	
Összehasonlító operátorok	<, >, =, <>, >=, <=	
Szöveg operátorok	+, %, _	+ : szövegek összefűzése % : helyettesítő operátor (egy vagy több karakter vagy üres) _ : helyettesítő operátor (egy karakter)

Lehetnek beépített függvények vagy a felhasználó által létrehozott függvények.

A paramétereken végzett műveletek eredményét adják vissza egy vagy több érték formájában.

Függvény típusok:

- Beépített függvények
  - Aggregáló függvények (pl: SUM(), MIN(), MAX())
  - Analitikus függvények (pl: LAG(), LEAD(), FIRST\_VALUE())
  - Rangsor függvények (pl: RANK(), NTILE())
  - Rowset függvények (pl: OPENROWSET())
  - **Skalár függvények\*** (pl: YEAR(), LEFT(), ROUND())
  - Egyéb függvények (pl: ISNULL())
- Felhasználói függvények

\* Egyelőre főleg skalár függvényekkel foglalkozunk

# Fontosabb matematikai függvények

Függvény	Kötelező paraméterek	Funkció
POWER(x, y)	x: a hatványalap y: a kitevő (mindkettő numerikus kifejezés)	Hatványozás adott kitevőre
SQRT(x)	x: numerikus kifejezés	A szám négyzetgyökének számítása
ROUND(x, y)	x: a kerekítendő valós szám y: a kerekítés pontossága (mindkettő numerikus kifejezés)	Adott pontosságú kerekítés
ABS()	x: numerikus kifejezés	Az adott szám abszolútértékét adja meg

## PÉLDÁK

- a) `SELECT POWER(3, 5)`
- b) `SELECT SQRT(2020)`
- c) `SELECT ABS(-210.3)`

- d) `SELECT ROUND(32.332, 1)`
- e) `SELECT ROUND(322, -2)`
- f) `SELECT ROUND(232.2, 0)`



# Fontosabb dátum/idő függvények

Függvény	Kötelező paraméterek	Funkció
GETDATE()	--	Az aktuális rendszeridőt adja vissza
DAY(d) MONTH(d) YEAR(d)	d: dátum kifejezés	Az adott dátumhoz tartozó nap, hónap sorszámát, illetve az évszámot adja vissza
DATEADD(x, y, z)	x: a hozzáadandó dátumegység (day, week, month, year stb.) y: a hozzáadandó egységek száma z: dátum kifejezés	Adott dátumhoz ad hozzá adott számú napot, hetet, hónapot stb.
DATEDIFF(x, y, z)	x: dátumegység neve y: start dátum, z: vég dátum	A két dátum között lévő dátumegységek számát adja meg

## PÉLDÁK

a) `SELECT GETDATE()`

b) `SELECT MONTH(GETDATE())`

c) `SELECT DATEADD(year, -1, GETDATE())`

d) `SELECT DATEDIFF(day, '2019.01.01', GETDATE())`

# Fontosabb szöveg függvények

Függvény	Kötelező paraméterek	Funkció
LEN(x)	x: szöveg (string)	A szöveg hosszát adja meg karakterekben
LEFT(x, y) RIGHT(x, y)	x: szöveg (string) y: egész kifejezés	Az adott szövegből balról, illetve jobbról levág adott számú karaktert
LOWER(x) UPPER(x)	x: szöveg	A szöveget csupa kis-, illetve nagybetűssé alakítja
CHARINDEX(x, y)	x: a keresett szöveg y: a szöveg, amelyben keresünk	A keresett szöveg első előfordulásának pozícióját adja meg. Ha a keresett szöveg nem található, 0 lesz a visszaadott érték.

## PÉLDÁK

- a) `SELECT LEN('Budapest')`
- b) `SELECT LEFT('Kiss Béla',4)`
- c) `SELECT LOWER('Nagy Laci')`

- d) `SELECT UPPER('Nagy Laci')`
- e) `SELECT CHARINDEX('al', ' Dalma')`

# Konverziós függvények

Függvény	Kötelező paraméterek	Funkció
CAST(x AS y)	x: kifejezés y: típusnév	Az x kifejezést y típusúra alakítja
CONVERT(x, y)	x: típusnév y: kifejezés	Az y kifejezést x típusúra alakítja

A CAST szabványos – ezért preferált- , a CONVERT pedig SQL implementáció függő  
A CONVERT függvény rendelkezik egy opcionális paraméterrel, amellyel  
formátumot is lehet beállítani – nem törzsanyag.

## PÉLDÁK

- a) SELECT CAST(GETDATE() AS DATE)
- b) SELECT CAST('14.55' AS DECIMAL(5,2))
- c) SELECT CONVERT(INT, 5.32)
- d) SELECT CONVERT(VARCHAR(10), GETDATE(), 111)

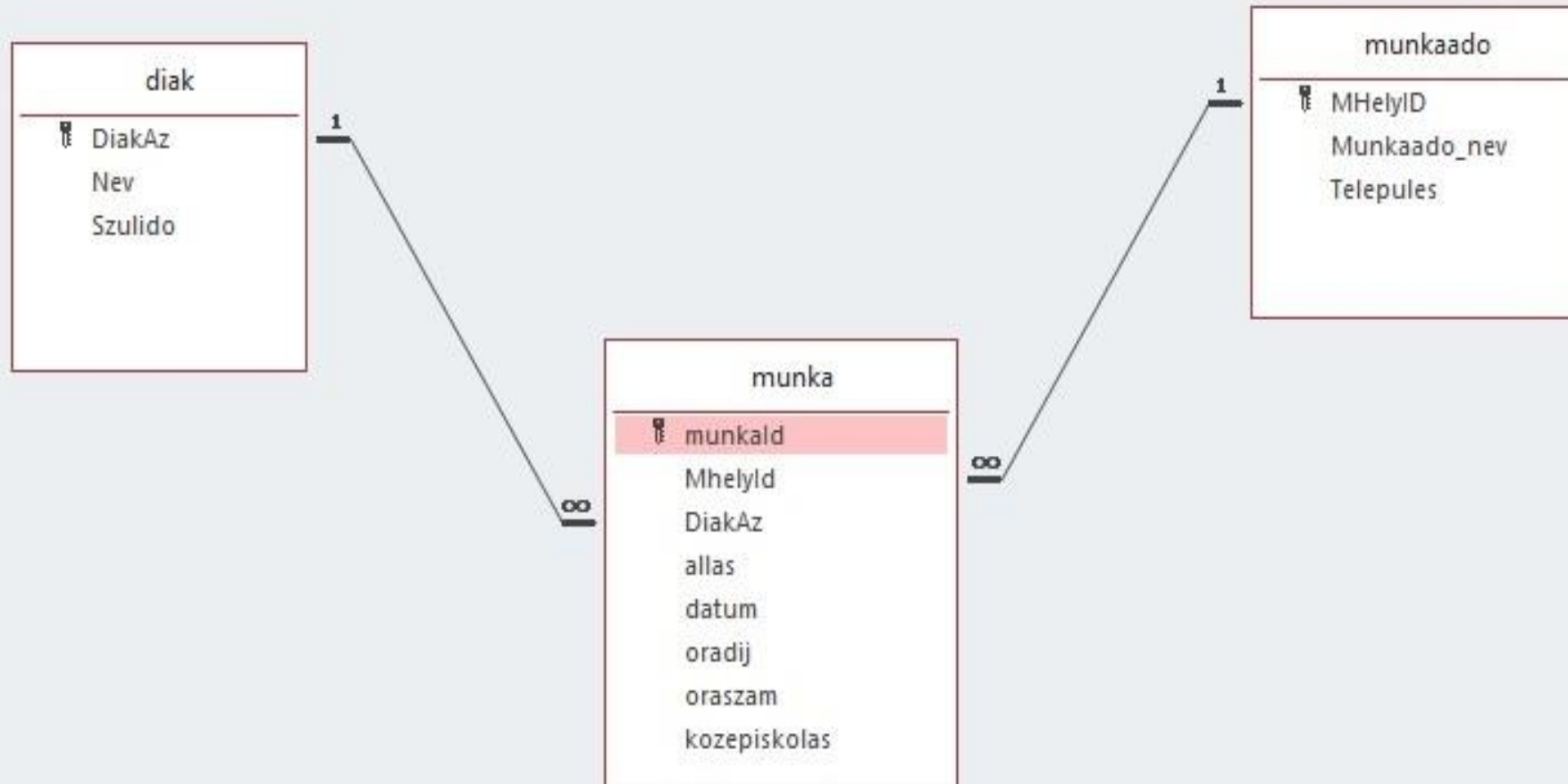
# Egyéb függvények

Függvény	Kötelező paraméterek	Funkció
ISNULL(x,y)	x: kifejezés y: érték	Ha x értéke NULL, akkor y-nal helyettesíti
IIF(x,y,z)	x: feltétel y: igaz érték z: hamis érték	Ha az x feltétel igaz, akkor y, különben z a visszaadott érték

## PÉLDÁK

- a) `SELECT ISNULL(nettóár, 0)`
- b) `SELECT ISNULL(vezetéknév, '')`
- c) `SELECT IIF(LEN(mobilszam)=9, 'OK', 'Nem OK')`

# A gyakorlaton használt adatbázis - Diákmunka



# Lekérdezés minták I. – Diákmunka adatbázis

1. Listázzuk a júniusi (6. havi) munkák minden adatát!

```
SELECT *  
FROM munka  
WHERE MONTH(datum)=6
```

2. Listázzuk a ruhatáros munkák adatait, tüntessük fel a naponta kereshető összeget is egy új oszlopban!  
Az oszlop neve legyen 'Napi kereset'

```
SELECT *, oradij * oraszam AS 'Napi kereset' FROM munka  
WHERE allas='ruhatáros'
```

3. Listázzuk a munka tábla első 6 oszlopát! Ha a diáknak nincs azonosítója, akkor helyette 0-t írjunk!  
A dátumból csak az év, hónap és nap jelenjen meg! Az óradíjat 100 forintra kerekítve jelenítsük meg!

```
SELECT munkaid, Mhelyld, allas,  
ISNULL(DiakAz,0) AS 'Diakazon',  
CAST(datum as date) AS 'Dátum',  
ROUND(oradij, -2) AS 'Óradíj'  
FROM munka
```

# Lekérdezés minták II. – Diákmunka adatbázis

4. Listázzuk a munka tábla oszlopait! Új oszlopként jelenítsük meg a munkák érvényességi dátumát (dátum + 30 nap) és egy minősítő oszlopot 'Fizetés' néven. Ez utóbbi 450 Ft-os óradíj felett legyen 'Jó', alatta 'Nem túl jó'

```
SELECT *,  
    DATEADD(day, 30, datum) AS 'Érvényes',  
    CASE WHEN oradij > 450 THEN 'Jó' ELSE 'Nem túl jó' END AS 'Fizetés' FROM munka
```

5. 2003.12.31-n kérdezzük le a munka tábla tartalmát. Egy új oszlopban jelenítsük meg, hogy hány hónap telt el a dátum oszlop értékéhez képest. Ahol ez meghaladja a 6 hónapot, ott egy másik új oszlopban írjuk ki, hogy 'Archiválandó'.

Az új oszlopok neve legyen 'Eltelt hónapok száma', illetve 'Megjegyzés'

```
SELECT *,  
    DATEDIFF(month, datum, '2003.12.31') AS 'Eltelt hónapok száma',  
    IIF(DATEDIFF(month, datum, '2003.12.31') > 6, 'Archiválandó', '')  
    AS 'Megjegyzés'  
FROM [proba].[dbo].[munka]
```

6. Listázzuk a diák tábla oszlopait! Két új oszlopban jelenítsük meg a vezetéknévet és a keresztnévet is!

```
SELECT *,  
    LEFT(Nev, CHARINDEX(' ', Nev) - 1) AS 'Vezetéknév',  
    RIGHT(Nev, LEN(Nev) - CHARINDEX(' ', Nev)) AS 'Keresztnév'  
FROM diak
```



**Köszönöm  
a figyelmet!**