MS-SQL

Učební pomůcka od **ITnetwork.cz**

## Práce s tabulkou

**Datové typy**

celá čísla: tinyint, smallint, int, bigint; texty: nvarchar(), nchar(); ostaní: date, time, datetime, blob;

**Upřesnění datového typu:** NULL / NOT NULL, IDENTITY, UNIQUE, PRIMARY KEY, DEFAULT;

|  |  |
| --- | --- |
| **Vytvoření tabulky**  CREATE TABLE [Clanky](  [Id] INT IDENTITY,  [AutorId] INT,  [Popis] NVARCHAR(155),  [Url] NVARCHAR(155),  [KlicovaSlova] NVARCHAR(155),  [Titulek] NVARCHAR(155),  [Obsah] NVARCHAR(MAX),  [Publikovano] DATETIME,  PRIMARY KEY ([Id])  ); | **Úprava návrhu tabulky**  ALTER TABLE [Clanky]  ADD [Palce] INT;  ALTER TABLE [Clanky]  ALTER COLUMN [Palce] BIGINT;  ALTER TABLE [Clanky]  DROP COLUMN [Palce]; |

**Odstranění tabulky:** DROP TABLE [Clanky];

## Práce s daty

|  |  |
| --- | --- |
| **Vložení dat**  INSERT INTO [Uzivatele] (  [Jmeno],  [Prijmeni],  [DatumNarozeni],  [PocetClanku])  VALUES ('Jan', 'Novák', '1984-11-03', 17),  ('Tomáš', 'Marný', '1989-02-01', 6),  ('Josef', 'Nový', '1972-12-20', 9); | **Aktualizace záznamu**  UPDATE [Uzivatele]  SET [Prijmeni] = 'Dolejší',  [PocetClanku] = [PocetClanku] + 1  WHERE [Id] = 1;  **Mazání dat**  DELETE FROM [Uzivatele] WHERE [Id] = 2; |

## Dotazování

SELECT [Prijmeni], [PocetClanku] FROM [Uzivatele] WHERE [Jmeno] = 'Jan';

**syntaxe podmínky**: WHERE [*sloupec*] *operátor* '*hodnota*';

operátory: =, >, <, >=, <=, !=; \* = vše, % = libovolný počet znaků; \_ = jeden libovolný znak;

další operátory: AND, OR, LIKE, IN(), BETWEEN, LTRIM(), RTRIM();

SELECT \* FROM [Uzivatele] WHERE [DatumNarozeni] >= '1960-1-1' AND [PocetClanku] > 5;

SELECT [Prezdivka], [Email] FROM [Uzivatele] WHERE [Email] LIKE '%@gmail.com';

řazení: ORDER BY, DESC;

SELECT [Jmeno], [Prijmeni], [PocetClanku]

FROM [Uzivatele] ORDER BY [PocetClanku] DESC, [Prijmeni];

agregační funkce: COUNT(), AVG(), SUM(), MIN(), MAX(), TOP;

seskupování: GROUP BY, AS;

SELECT [Jmeno], COUNT(\*) FROM [Uzivatele] GROUP BY [Jmeno];

třídění skupin záznamů: HAVING

SELECT vybrane\_sloupce FROM nazev\_tabulky

WHERE vyhledavaci\_podminka GROUP BY seskupeni\_podle\_sloupce HAVING podminka\_seskupeni;

## Práce s více tabulkami

vazby mezi tabulkami: 1:N (one to many); M:N (many to many);

základní příkazy: JOIN, ON, INNER / LEFT / RIGHT JOIN;

|  |  |
| --- | --- |
| SELECT [Titulek], [Prezdivka]  FROM [Clanky]  JOIN [Uzivatele]  ON [Clanky].[AutorId] = [Uzivatele].[Id]  ORDER BY [Prezdivka]; | SELECT [Titulek], [Prezdivka]  FROM [Clanky]  RIGHT JOIN [Uzivatele]  ON [Clanky].[AutorId] = [Uzivatele].[Id]  ORDER BY [Prezdivka]; |

**alias - AS**

SELECT [U].[Prezdivka], [K].[Obsah], [C].[Titulek]

FROM [Komentare] AS [K]

INNER JOIN [Uzivatele] AS [U] ON [U].[Id] = [K].[UzivatelId]

INNER JOIN [Clanky] AS [C] ON [C].[Id] = [K].[ClanekId]

ORDER BY [K].[Datum];

**Vazební tabulka** (propojení dvou tabulek, sama nenese žádná data)

|  |  |
| --- | --- |
| **Vytvoření**  CREATE TABLE [ClanekSekce] (  [Id] INT IDENTITY,  [ClanekId] INT,  [SekceId] INT,  PRIMARY KEY ([Id])  ); | **Naplnění daty**  INSERT INTO [ClanekSekce] ([ClanekId], [SekceId]) VALUES  (1, 1),  (2, 1),  (2, 2),  (3, 2),  (4, 2); |

## Poddotazy

|  |  |
| --- | --- |
| **Korelovaný poddotaz** (souvisí s vnějším)  SELECT TOP 1  [U].[Prezdivka],  (  SELECT  COUNT(\*)  FROM [Clanky] AS [C]  WHERE [C].[AutorId] = [U].Id  ) AS [Pocet]  FROM [Uzivatele] AS [U]  ORDER BY [pocet] DESC; | **Nekorelovaný poddotaz**  SELECT  [C].Id,  [C].Titulek  FROM [Clanky] AS [C]  WHERE (  SELECT  COUNT(\*)  FROM [ClanekSekce]  WHERE ([ClanekSekce].[ClanekId] = [C].[Id])) = 0; |

víceřádkové operátory: IN, ALL, ANY, EXISTS;

## Transakce

ACID = nedělitelnost, validita, izolace, trvanlivost

BEGIN TRANSACTION

UPDATE [Ucty] --odečtení peněz

SET [Zustatek] = [Zustatek] – 100 WHERE [CisloUctu] = 123456789;

UPDATE [Ucty] --přičtení peněz

SET [Zustatek] = [Zustatek] + 100 WHERE [CisloUctu] = 987654321;

COMMIT TRANSACTION

zrušení transakce: ROLLBACK TRANSACTION

## Propojení tabulek

Referenci vytváříme z podřízené tabulky k nadřízené

CONSTRAINT nazev\_vztahu

FOREIGN KEY nazev\_sloupce -- sloupec podřízené tabulky

REFERENCES nadrizena\_tabulka(nazev\_sloupce)

ON UPDATE ... -- co se děje při aktualizaci záznamu

ON DELETE ... -- co se děje při smazaní záznamu

základní příkazy: FOREIGN KEY, ON UPDATE / DELETE + hodnoty: SET NULL (nastavení musí umožňovat NULL), SET DEFAULT, NO ACTION (výchozí; nelze odstranit nebo aktualizovat rodičovský záznam), CASCADE (smažou se/aktualizují se i propojené záznamy);

...CONSTRAINT [KomentareClanek] FOREIGN KEY ([ClanekId])

REFERENCES [dbo].[Clanky] ([Id])

ON DELETE CASCADE ON UPDATE CASCADE;

## Odstranění propojení

ALTER TABLE [Komentare]

DROP CONSTRAINT [KomentareClanek];

## Pozor na SQL injection!

Zranitelný kód: DELETE FROM [Uzivatele] WHERE [Prijmeni] = '" + prijmeni + "';

Ochrana: DELETE FROM [Uzivatele] WHERE [Prijmeni] = @prijmeni;