

Jazyk SQL – vytvoření tabulky

Jiří Zacpal



KATEDRA INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

KMI/DATAB Databáze

Jak získat zápočet

- Za získání **15 bodů**
- Za co získat body?
 - Úkol na hodině = 1 bod
 - 2. písemné práce = 0 až 5 bodů
- Pokud se nepodaří získat příslušný počet bodů, dostanete dodatečný úkol

Konzultace



- v pracovně 5.071
 - Středa 12:00-13:00
- jindy po vzájemné domluvě
- email: jiri.zacpal@upol.cz
- MS Teams

PostgreSQL

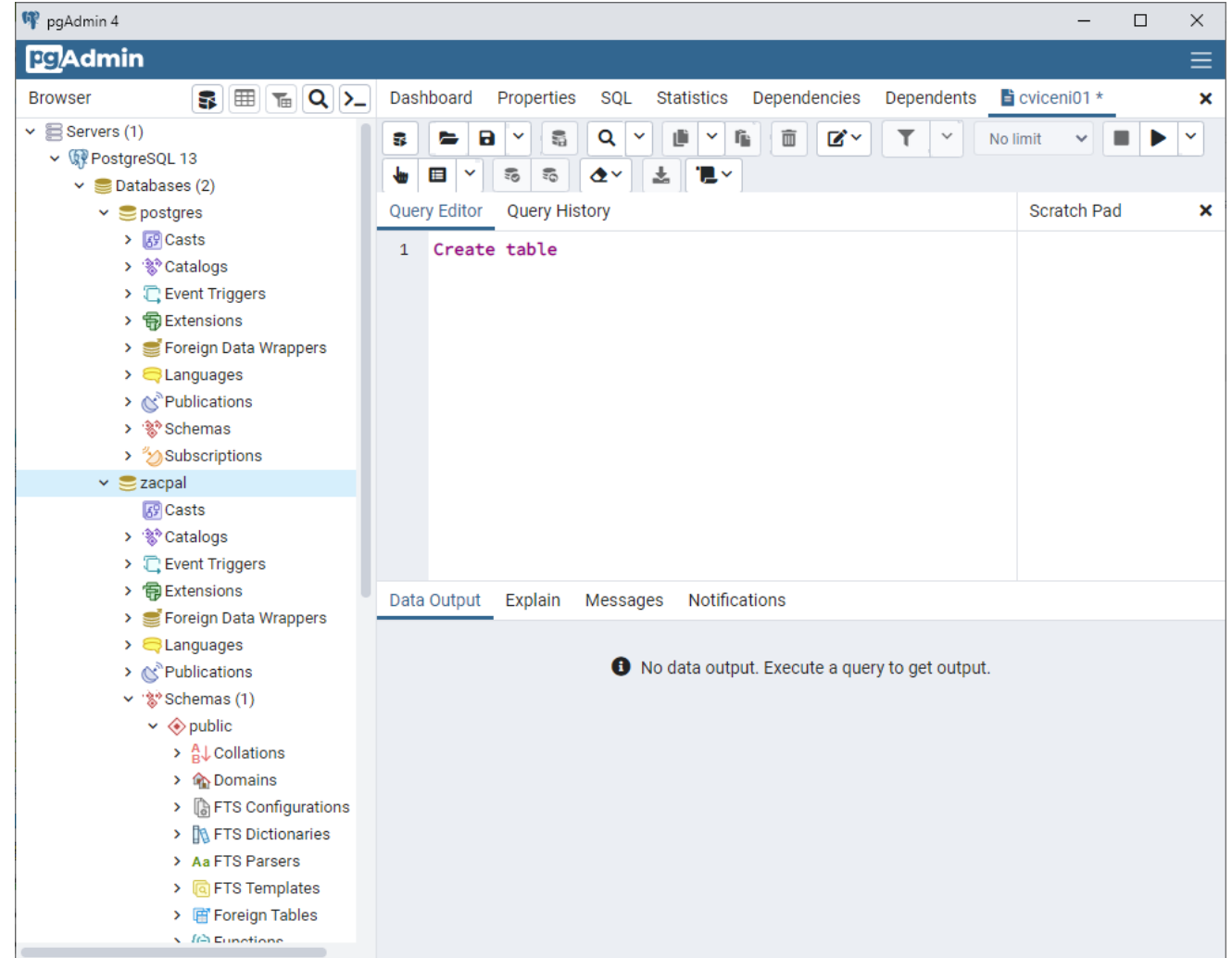


- Svobodný a otevřený objektově-relační databázový systém.
- Na jeho vývoji se podílí globální komunita vývojářů a firem, stejně jako v případě mnoha dalších otevřených programů, jelikož PostgreSQL není vlastněn jedinou firmou.
- Funkce PostgreSQL zahrnují:
 - databázové transakce s atomicitou, konzistencí, izolovaností a trvalostí (ACID),
 - automaticky aktualizovatelné pohledy, materializované pohledy,
 - trigger, cizí klíče a uložené procedury.
- Je navržen pro zpracování celé řady vytížení, od jednotlivých strojů po datové sklady nebo webové služby s mnoha souběžnými uživateli.
- PostgreSQL je primárně vyvíjen pro Linux, dále je dostupný pro macOS server, FreeBSD, OpenBSD a Microsoft Windows a je vydáván pod licencí MIT.

PostgreSQL - pgAdmin



- Nástroj pro správu databáze.
- Vytvoření databáze (Object – Create – Database).
- Vytvoření query (Pravé tlačítko na databázi – Query Tool).
- Přejmenování a uložení query.

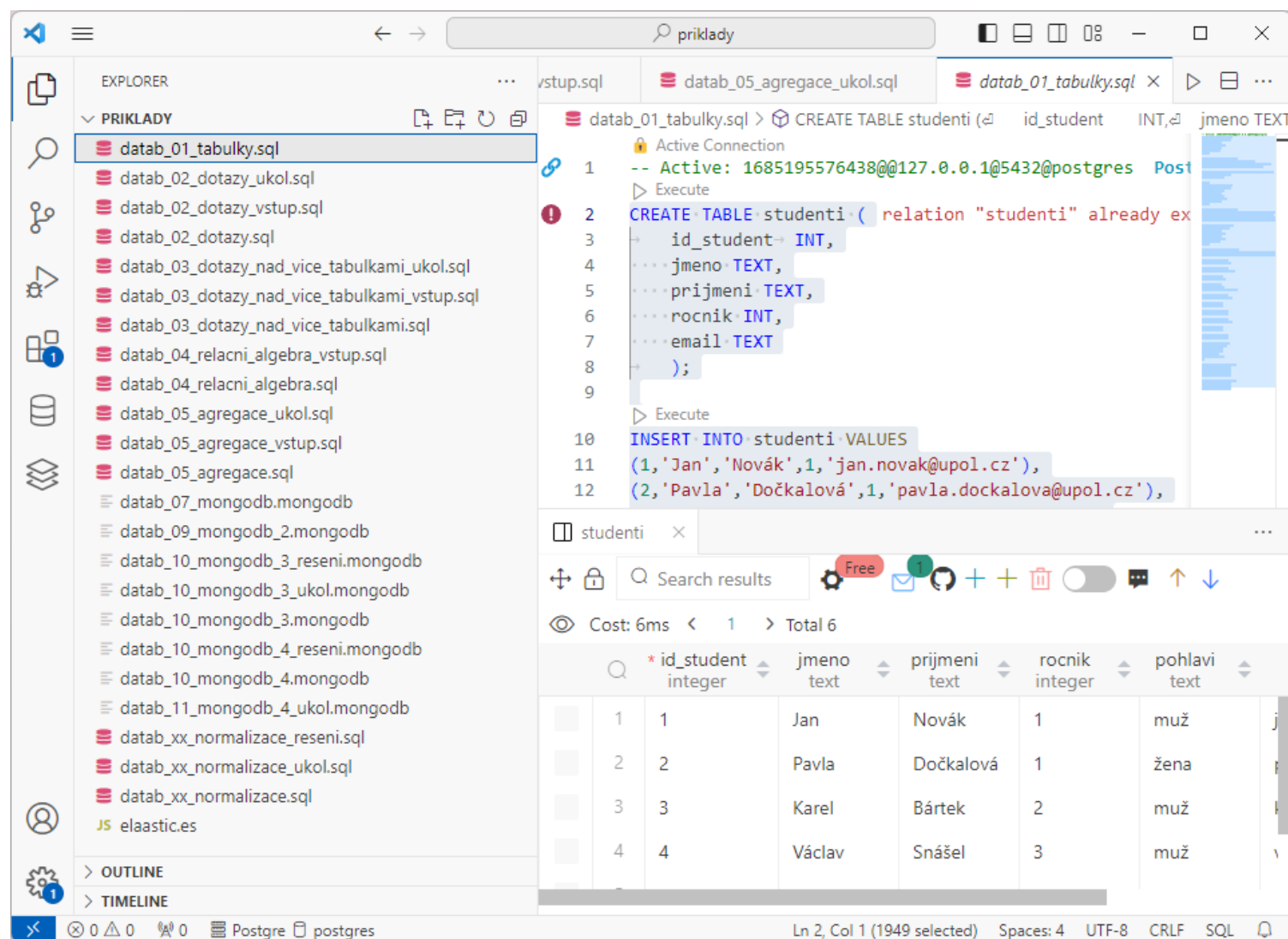


SQL online



- Na adrese: sqliteonline.com

ID	Name	Hint
1	SQL Online	for Data Science
2	Kirill N.	https://www.linkedin.com/in/sqlit...
3	Twitter	https://twitter.com/SqliteOnlineCom
4	Chart	LINE-SELECT name, cos(id), sin...
5	Short CODE	s* tableName => SELECT * FR...
6	SQLite 3.36.0	SQL OnLine on JavaScript
7	[RightClick] mouse	Opens many additional features
8	Left-Panel, Table	[RightClick] mouse "Context menu"
9	Tabs	mouse: [RightClick] , [MiddleClic...
10	SQL Editor	autocomplete: [Ctrl-Space] or [Al...
11	Size table	Fast scroll million rows
12	Share	Create public link D



The screenshot shows the VS Code interface with the following components:

- EXPLORER:** A list of SQL files under the 'PRIKLADY' folder, including 'datab_01_tabulky.sql', 'datab_02_dotazy_ukol.sql', 'datab_02_dotazy_vstup.sql', 'datab_02_dotazy.sql', 'datab_03_dotazy_nad_vice_tabulkami_ukol.sql', 'datab_03_dotazy_nad_vice_tabulkami_vstup.sql', 'datab_03_dotazy_nad_vice_tabulkami.sql', 'datab_04_relacni_algebra_vstup.sql', 'datab_04_relacni_algebra.sql', 'datab_05_agregace_ukol.sql', 'datab_05_agregace_vstup.sql', 'datab_05_agregace.sql', 'datab_07_mongodb.mongodb', 'datab_09_mongodb_2.mongodb', 'datab_10_mongodb_3_reseni.mongodb', 'datab_10_mongodb_3_ukol.mongodb', 'datab_10_mongodb_3.mongodb', 'datab_10_mongodb_4_reseni.mongodb', 'datab_10_mongodb_4.mongodb', 'datab_11_mongodb_4_ukol.mongodb', 'datab_xx_normalizace_reseni.sql', 'datab_xx_normalizace_ukol.sql', 'datab_xx_normalizace.sql', and 'js elastic.es'.
- Code Editor:** The 'datab_01_tabulky.sql' file is open, showing SQL queries. The first query is a 'CREATE TABLE' statement for 'studenti' with columns 'id_student' (INT), 'jmeno' (TEXT), 'prijmeni' (TEXT), 'rocnik' (INT), and 'email' (TEXT). The second query is an 'INSERT INTO' statement with values for the 'studenti' table.
- Results:** The 'studenti' table is displayed with 6 rows. The columns are 'id_student' (integer), 'jmeno' (text), 'prijmeni' (text), 'rocnik' (integer), and 'pohlavi' (text).

	* id_student	jmeno	prijmeni	rocnik	pohlavi
	integer	text	text	integer	text
1	1	Jan	Novák	1	muž
2	2	Pavla	Dočkalová	1	žena
3	3	Karel	Bártek	2	muž
4	4	Václav	Snášel	3	muž

Vytvoření tabulky

Příklad

- Chceme vytvořit databázi, do které budeme ukládat tato:
 - Data o **studentech**, kteří ve škole studují.
 - Data o **předmětech**, které se zde vyučují.
 - Data o **vyučujících**, kteří ve škole vyučují.
 - Data o tom, který **student** studuje jaký **předmět**.
 - Data o tom, který **vyučující** vyučuje který **předmět**.

Vytvoření struktury tabulky

- základní syntaxe:

```
CREATE TABLE jmeno (  
    sloupec_1    typ [omezení],  
    ...  
    sloupec_n    typ [omezení]  
);
```

Příklad

- Vytvoříme tabulku pro uložení dat o studentech:

```
CREATE TABLE studenti (  
    id_student INT,  
    jmeno TEXT,  
    prijmeni TEXT,  
    rocnik INT,  
    email TEXT  
);
```

Vkládání záznamů do tabulky

- obecná syntaxe příkazu **INSERT**:

INSERT INTO *jmeno* **VALUES**(*udaj1*, *udaj 2*, ..., *udaj n*)

- *udaj 1*, ..., *udaj n* odpovídají jednotlivým atributům tabulky

- vkládání jen některých atributů

INSERT INTO *jmeno*(*atr1*, ..., *atr n*) **VALUES**(*udaj1*, ..., *udaj n*)

- *udaj 1*, ..., *udaj n* odpovídají atributům *atr1*, ..., *atr n* tabulky
- hodnoty ostatních atributů budou NULL

Příklad

- Vložíme do tabulky studenti několik řádků:

```
INSERT INTO studenti VALUES
```

```
(1, 'Jan', 'Novák', 1, 'jan.novak@upol.cz'),  
(2, 'Pavla', 'Dočkalová', 1, 'pavla.dockalova@upol.cz'),  
(3, 'Karel', 'Bártek', 2, 'karel.bartek@upol.cz');
```

- Můžeme také vložit jen některé sloupce:

```
INSERT INTO studenti(id_student, jmeno, prijmeni) VALUES
```

```
(4, 'Václav', 'Snášel'),  
(5, 'Iva', 'Nová');
```

Příklad



- Výsledná tabulka vypadá takto:

	Q	id_student integer	jmeno text	prijmeni text	rocnik integer	email text
<input type="checkbox"/>	1	1	Jan	Novák	1	jan.novak@upol.cz
<input type="checkbox"/>	2	2	Pavla	Dočkalová	1	pavla.dockalova@upol.cz
<input type="checkbox"/>	3	3	Karel	Bártek	2	karel.bartek@upol.cz
<input type="checkbox"/>	4	4	Václav	Snášel	(NULL)	(NULL)
<input type="checkbox"/>	5	5	Iva	Nová	(NULL)	(NULL)

Úkol



- Vytvořte tabulku predmety, které bude obsahovat id_predmetu, zkratku, nazev, semestr, vyucujici a kredity.
- Do tabulky doplňte všechny předměty, které máte zapsány.

Řešení



```
CREATE TABLE predmety (  
    id_predmet INT,  
    zkratka TEXT,  
    nazev TEXT,  
    semestr TEXT,  
    vyucujici TEXT,  
    kredity INT  
);
```


Příklad

- Je tabulka vhodně navržena?
- Co vyučující?

INSERT INTO predmety **VALUES**

(1, 'KMI/DATA', 'Databáze', 'ZS', 'Mgr. Jiří Zacpal,
Ph.D.', 5),

(2, 'KMI/WEBA', 'Webové aplikace', 'ZS', 'Jiří Zacpal', 3),

(3, 'KMI/SOFT', 'Softwarové inženýrství', 'ZS', 'Zacpal
Jiří', 6),

(4, 'KMI/ZPP1', 'Základy programování pro IT 1', 'ZS', 'J.
Zacpal', 4);

Příklad

- Vhodnější bude vytvořit pro vyučujícího samostatnou tabulku:

```
CREATE TABLE vyucujici (  
    id_vyucujici INT,  
    jmeno TEXT,  
    prijmeni TEXT,  
    titul TEXT,  
    kancelar TEXT,  
    email TEXT  
);
```

Příklad

- Vložíme do ní vyučující:

```
INSERT INTO vyucujici VALUES
```

```
(1, 'Jiří', 'Zacpal', 'Mgr.,  
Ph.D.', '5.071', 'jiri.zacpal@upol.cz'),  
(2, 'Arnošt', 'Večerka', 'RNDr.', '5.071', 'arnost.vecerka@upol  
.cz');
```

- Musíme upravit tabulku predmety. Jak?

Úprava struktury tabulky

- základní syntaxe:

- přidání sloupce:

```
ALTER TABLE jmeno  
    ADD COLUMN sloupec    typ [omezení];
```

- smazání sloupce:

```
ALTER TABLE jmeno  
    DROP COLUMN sloupec;
```

Příklad

- Nejprve odstraníme sloupec vyucujici z tabulky predmety:

```
ALTER TABLE predmety
```

```
    DROP COLUMN vyucujici;
```

- Následně sloupec se stejným názvem přidáme:

```
ALTER TABLE predmety
```

```
    ADD COLUMN vyucujici INT;
```

Příklad



- Tabulka predmety teď vypadá takto:

id_predmet integer	zkratka text	nazev text	semestr text	kredity integer	vyucujici integer
1	KMI/DATA	Databáze	ZS	5	(NULL)
2	KMI/WEBA	Webové aplikace	ZS	3	(NULL)
3	KMI/SOFT	Softwarové inženýrství	ZS	6	(NULL)
4	KMI/ZPP1	Základy programování pro	ZS	4	(NULL)

- Musíme aktualizovat její řádky.

Aktualizace záznamů

- obecná syntaxe příkazu **UPDATE**:

UPDATE jmeno

SET atribut **1**= udaj1,... atribut n=udaj n
[WHERE podmínka]

- aktualizace jednoho záznamu:

UPDATE jmeno

SET atribut **1**= udaj1,... atribut n=udaj n
WHERE unikatni_atribut=hodnota

Příklad

- U všech předmětů nastavíme vyučujícího Jiřího Zacpala:

```
UPDATE predmety
```

```
SET vyucujici = 1;
```

- U předmětu KMI/SOFT nastavíme jako vyučujícího Arnošta Večerku:

```
UPDATE predmety
```

```
SET vyucujici = 1;
```


Příklad



- Poslední věc, kterou musíme v databázi nastavit, je umožnit zaznamenat skutečnost, že daný student studuje zvolený předmět.
- Do tabulky studenti přidáme sloupec predmet:

```
ALTER TABLE studenti  
    ADD COLUMN predmet INT;
```

- Všem studentům nastavíme, že studují předmět KMI/DATAB:

```
UPDATE studenti  
SET predmet = 1;
```

- Co ale když student studuje více předmětů?
- Toto není dobré řešení.
- Sloupec předmět opět smažeme:

```
ALTER TABLE studenti  
    DROP COLUMN predmet;
```

Příklad

- Zkusíme tedy do tabulky predmety přidat sloupec student.

```
ALTER TABLE predmety  
ADD COLUMN student INT;
```

- Všem předmětům nastavíme, že je studuje Jan Novák:

```
UPDATE predmety  
SET student = 1;
```

- Co ale když předmět studuje více studentů?
- Toto není dobré řešení.
- Sloupec student opět smažeme:

```
ALTER TABLE predmety  
DROP COLUMN student;
```

Příklad



- Správný řešením je vytvořit novou tabulku, kterou nazveme např. zapsani:

```
CREATE TABLE zapsani(  
    student INT,  
    predmet INT,  
    rok TEXT,  
    uspel BOOLEAN  
);
```

Příklad

- To tabulky zaznameneáme, že jeden student studuje více předmětů:

```
INSERT INTO zapsani VALUES
```

```
(1,1,1, '23/24', FALSE),
```

```
(2,1,2, '23/24', FALSE),
```

```
(3,1,3, '23/24', FALSE),
```

```
(4,1,4, '23/24', FALSE);
```

- A také to, že jeden předmět studuje více studentů:

```
INSERT INTO zapsani VALUES
```

```
(5,2,2, '23/24', FALSE),
```

```
(6,3,2, '23/24', FALSE),
```

```
(7,4,2, '23/24', FALSE);
```

Příklad



- Tabulka zapsani vypadá takto:

id_zapsani integer	student integer	predmet integer	rok text	uspel boolean
1	1	1	23/24	false
2	1	2	23/24	false
3	1	3	23/24	false
4	1	4	23/24	false
5	2	2	23/24	false
6	3	2	23/24	false
7	4	2	23/24	false

Bodovaný úkol



- Vytvořte tabulky pro uložení dat o vaší knihovně. Budeme do ní ukládat toto:
 - Data o knihách z knihovny (název, autor, rok vydání, počet stran, vydavatel, žánr).
 - Data o autorech (jméno a příjmení, národnost, informaci zda žije či ne).
- Do takto vytvořené databáze uložte data alespoň o
 - 5 knihách
 - 3 spisovatelích.