

SQL

II. Mierne Pokročilý



Ako Začneme?

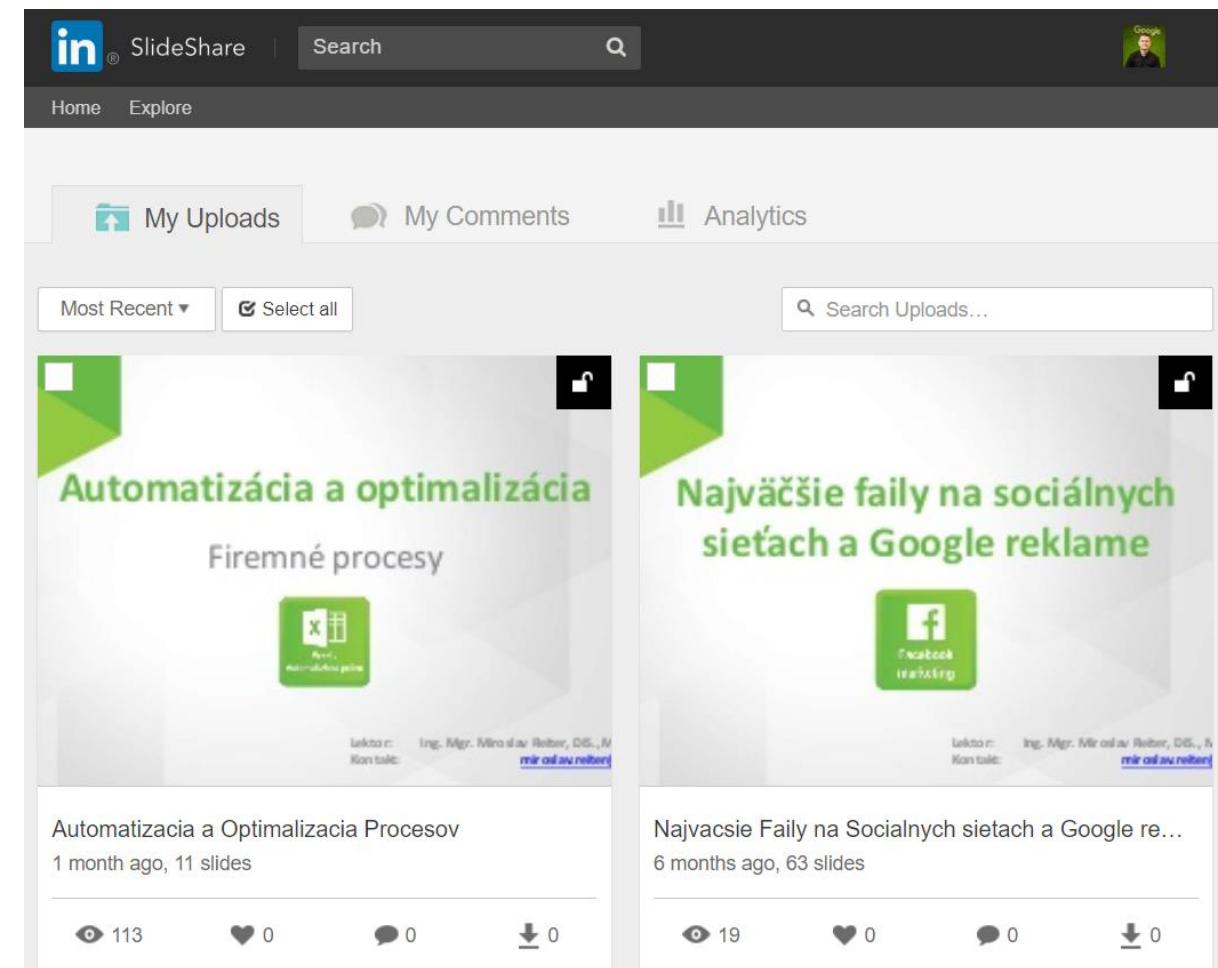
1. Pripojenie na DB/DBMS

- https://github.com/miroslav-reiter/Kurzy_SAV_Analytika_Python_R

2. Pridajte si ma na LinkedIn

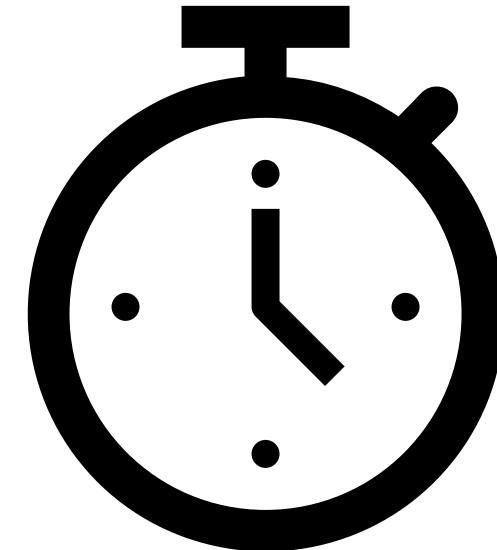
- www.linkedin.com/in/miroslav-reiter/

3. Prezentácia a materiály po prednáške



Úvodné Informácie

- Časový rozvrh (9:00-13:30)
 - Programujeme (50 min)
 - Prestávky (10 min)
 - Obedová prestávka
 - Mobilné telefóny a zariadenia
-
- Priprav si otázky a rovno sa pýtaj
 - Interaktívna forma



O mne - Miroslav Reiter

4

40000+
klientov a
1000+ firiem

IT Architekt
Programátor
Manažér

Microsoft
Google
ISTQB tréner

134
certifikácií

151 príručiek
a publikácií

13 škôl

62 projektov

Vlastná firma



MOTIVÁCIA

Študuje 5 odborov a absolvoval už 12 univerzít. Ako zvláda stres a manažovanie času?



Foto: Jakub Kovalík pre FMK UCM | Miroslav Reiter na prednáške Grow with Google na FMK UCM.

Nikola Kotláriková

19. júl 2022 · 8 min. čítania



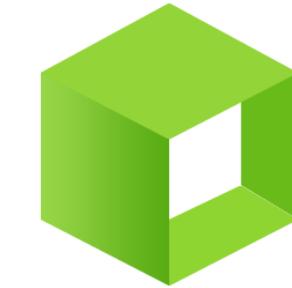
Miroslav Reiter

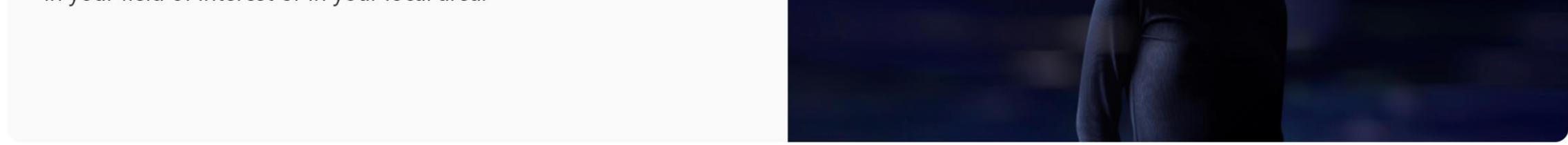


1. PhDr. VŠM (Podnikovný manažment)
2. Ing. STU FEI (Aplikovaná informatika)
3. Mgr. UK FM (Strategický manažment a marketing)
4. Mgr. VŠM (Manažérstvo kvality)
5. Mgr. VŠEMVŠ (Verejná správa)
6. Mgr. DTI (Učiteľstvo ekonomických predmetov)
7. DiS. AMOS (Cestovný ruch)
8. MBA LIGS (Executive management)
9. DBA Humanum (IT manažment)
10. MPA IES (Verejná správa a samospráva krajov)
11. MSc. Humanum (Bezpečnosť inf. systémov)
12. Ing. Paed. IGIP STU
13. Mgr. PEVŠ (Bezpečnosť informačných systémov)



DIGITÁLNA
UNIVERZITA





Filter

[People](#) [Events](#)

[Search by filter](#)

- > Program
- > Country/Region
- > State/Province
- > Language
- > Award Category
- > Technology Area

Search Profiles

[Search Profiles](#)

[Search](#)

[Most Valuable Professionals](#)

[Country/Region: Slovakia](#)

Most Valuable Professionals



[Miroslav Reiter](#)

IT Architect and Programmer Hard worker Lecturer and Certified Trainer

Slovakia

Most Valuable Professionals



[Peter Belko](#)

Content Developer, Technical Writer, Consultant, Trainer for M365 + Office apps

Slovakia

Most Valuable Professionals

[About](#) [Events](#) [Find an MVP](#)[Profile](#) [Events](#)

Headline

🌟 IT Architect and Programmer 📚 Hard worker 🎓
Lecturer and Certified Trainer

Biography

👉 I'm a hard worker, intellectual and joker. I love learning and teaching as well. My main objective is to teach people and improve their IT knowledge. To create truly practical knowledge necessary for life and present it in an interesting way. I don't like snake charmers and people who cannot...

[▼ Read more](#)

Miroslav Reiter

 Slovakia IT Academy s.r.o.

Most Valuable Professionals

High Impact Activities

Award Category

M365

Technology Area

Visio, Excel

Languages

English, Slovak

Social



This community leader has not added a high impact activity yet.

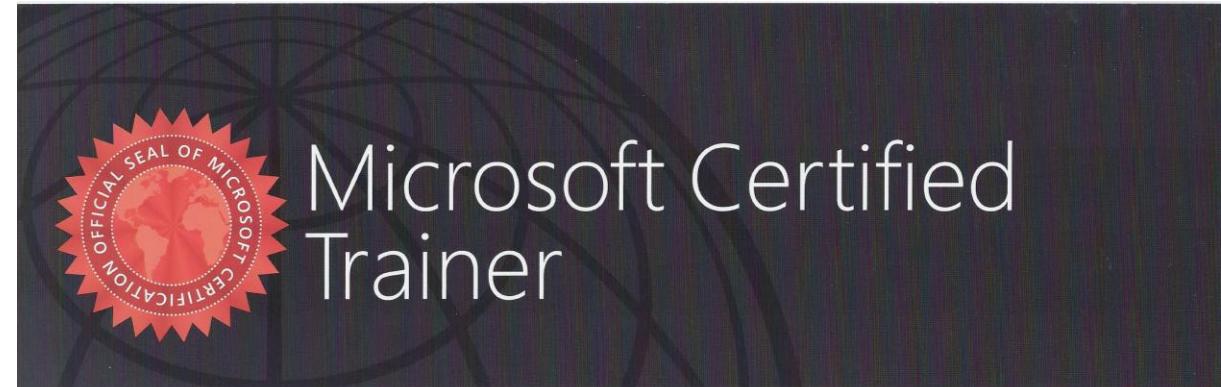
Miroslav Reiter

získava status

Google Certified Trainer

Automation

Google



Microsoft Certified
Trainer

MIROSLAV REITER

Has successfully completed the requirements to be recognized as a Microsoft Certified Trainer

N. S.
Satya Nadella
Chief Executive Officer

Microsoft
CERTIFIED
Trainer

Znalec

PhDr. Ing. Miroslav Reiter

Evidenčné číslo: 915864

Miesto výkonu činnosti

Tomášikova 50
83104 Bratislava
Slovenská republika
[Zobrazit na mape](#)

Kontaktné údaje

Mobil: +421 908 163 084
E-mail: znapec@it-academy.sk

Odbory a odvetvia

Odbor / Odvetvie	História	Stav
100000 - Elektrotechnika		
100400 - Riadiaca technika, výpočtová technika (hardware)	14.02.2023 - Zápis	AKTÍVNY
100800 - Nosiče zvukových a zvukovoobrazových záznamov	14.02.2023 - Zápis	AKTÍVNY
100900 - Počítačové programy (software)	14.02.2023 - Zápis	AKTÍVNY
101000 - Bezpečnosť a ochrana informačných systémov	14.02.2023 - Zápis	AKTÍVNY

Vyberte si online kurz

Naučte sa programovať, tvoriť webstránky a grafiku, manažovať alebo sa zamerajte na osobný rozvoj. Všetko jednoducho vďaka našim online kurzom z pohodlia domova.

Ročné Predplatné na všetky Online Kurzy

~~2290 €~~**490 €**

Prístup pre Vás do všetkých Aktuálnych aj Pripravovaných Online Kurzov

12 mesačná platnosť

Kúpiť teraz[Zistíť viac](#)

529 kurzov v ponuke



Zábavné online lekcie



Akreditované kurzy



12 rokov skúseností



Certifikovaní profesionálni lektori

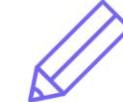
Odporučame Kurzy špeciálne pre vás



Online kurz SQL I.
Začiatočník
110,00€ 142,00€



Online kurz SAP I.
Začiatočník
224,00€ 292,50€



Online kurz Copywriting I.
Začiatočník
108,00€ 115,00€



Balík SAP Profesionál
998,80€ 998,80€



Online kurz Digitálny Marketing
146,00€ 199,00€



Online kurz Facebook Marketing I. Začiatočník
104,00€ 142,00€



Luigi, Mário
a Yoshi

Čo Robíte?

1. Študent/učiteľ'

2. Zamestnanec

3. Podnikateľ'

4. Nezamestnaný/materská

5. Dievča pre všetko



Vaše Ciele a Očakávania

15

1. Získat ďalšie znalosti jazyka SQL

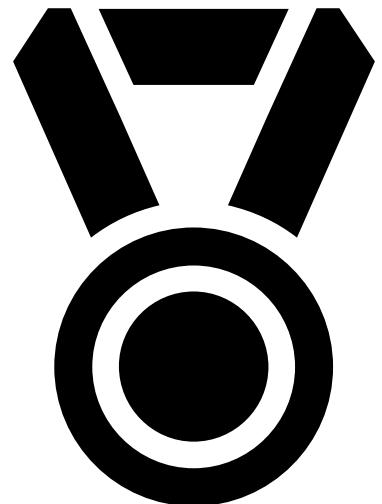
2. Doplniť si znalosti z Databáz

3. Základy analytického myslenia

4. Základy spracovania dát

5. Doplniť si znalosti z Data Science

Zábava je v zaručená v každom bode :-)



SQL a NoSQL



Tabuľka

Údaj (Data)	Názov Poľa (Field Name)			Pole (Field)
	Obchodník	Tovar	Cena	
Záznam 1 entita (Record)	Mrkvička	Hrebeň	453€	Odberateľ
	Hraško	Papuče	7523€	1 teplárenská a.s.
	Mrkvička	Papuče	35€	Alojz Zima
	Mrkvička	Náradie	643€	Montáže a.s.
	Jánošík	Hrebeň	345€	Ján Strapatý
	Jánošík	Čistiace pr.	1345€	Viktor Čistič
	Hraško	Čistiace pr.	245€	Čistenie a.s.
	Hraško	Aviváž	324€	Huhu s.r.o.
	Mrkvička	Deodorant	323€	Smrad s.r.o.
	Jánošík	Deodorant	2344€	Huhu s.r.o.
	Jánošík	Aviváž	323€	Huhu s.r.o.

SQL Jazyk – DML

- 1. SELECT** – výber dát
- 2. INSERT** – vloženie dát
- 3. UPDATE** – aktualizácia dát
- 4. DELETE** – zmazanie dát
- 5. UPSERT** – aktualizácia a súčasne vkladanie dát



C R U D

Dátové Dopytovanie

SQL (Structured Query Language)

1. E-commerce platformy
(MySQL a PostgreSQL)
2. Bankovníctvo a finančné služby
(Oracle Database a MS SQL Server)
3. Zdravotníctvo

NoSQL (Not Only SQL)

1. Sociálne siete
(Cassandra a MongoDB)
2. Real-time analýza a monitorovanie
(Redis)
3. Big Data a analytika
(HBase a Cassandra)

Vytvorenie Tab. a Vloženie Dát

SQL (Structured Query Language)

```
1. CREATE TABLE Customers (
2.     ID int NOT NULL AUTO_INCREMENT,
3.     FirstName varchar(255) NOT NULL,
4.     LastName varchar(255) NOT NULL,
5.     Email varchar(255),
6.     PRIMARY KEY (ID)
7. );
8. INSERT INTO Customers (FirstName, LastName,
Email)
9. VALUES ('Ján', 'Novák', 'jan.novak@email.sk'),
10.          ('Eva', 'Biela', 'eva.biela@email.sk');
```

NoSQL (Not Only SQL)

```
1. // Vytvorenie kolekcie a pridanie dokumentov
2. db.customers.insertMany([
3.     { firstName: "Ján", lastName: "Novák",
email: "jan.novak@email.sk" },
4.     { firstName: "Eva", lastName: "Biela",
email: "eva.biela@email.sk" }
5. ]);

6. // Aktualizácia dokumentu MongoDB
7. db.customers.updateOne(
8.     { lastName: "Novák" },
9.     { $set: { email: "novy.email@email.sk" } }
10. );

11. // Výber dokumentov
12. db.customers.find({ lastName: "Novák" });
```

Aktualizácia a Výber Dát

SQL (Structured Query Language)

1. **UPDATE** Customers
2. **SET** Email = 'novy.email@email.sk'
3. **WHERE** LastName = 'Novák';
4. **SELECT * FROM** Customers
5. **WHERE** LastName = 'Novák';

NoSQL (Not Only SQL)

1. # Uloženie dát REDIS
2. **SET** customer:1000 "{
 'firstName': 'Ján', 'lastName':
 'Novák', 'email':
 'jan.novak@email.sk' }"
3. # Získanie dát
4. **GET** customer:1000

Tréning a Precvičovanie

SQL (Structured Query Language)

1. <https://sqlzoo.net>
2. <https://sqlfiddle.com/>
3. <https://www.db-fiddle.com/>
4. <https://dbfiddle.uk/>
5. <https://www.hackerrank.com/domains/sql>

NoSQL (Not Only SQL)

1. <https://nosqlzoo.net/>
2. <https://learn.mongodb.com/>



DBeaver Community

Free Universal Database Tool

Star 33,943



Follow @dbeaver_news

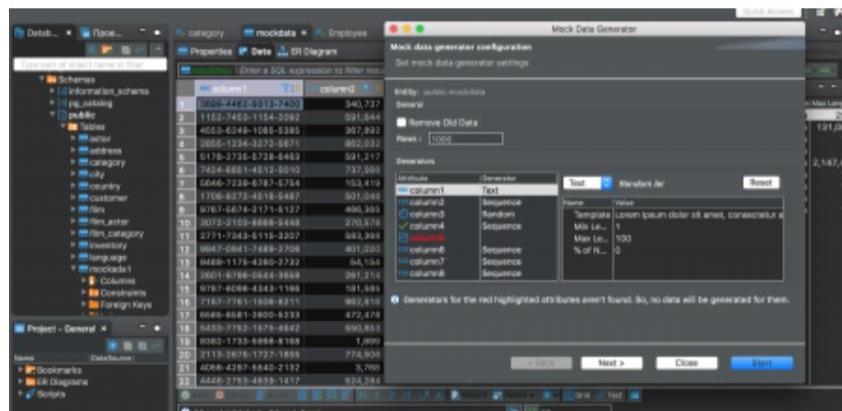
search here ...

Go

[Home](#) [About](#) [Download](#) [Documentation](#) [News](#) [Support](#) [DBeaver PRO](#) [CloudBeaver](#) [DBeaver Merch](#)

Universal Database Tool

DBeaver Community is a free cross-platform database tool for developers, database administrators, analysts, and everyone working with data. It supports all popular SQL databases like MySQL, MariaDB, PostgreSQL, SQLite, Apache Family, and more.



DBeaver Community

Open-source version

- Basic support for relational databases:
MySQL, SQL Server, PostgreSQL and [others](#)

DBeaver PRO

Commercial versions

- All DBeaver Community features
- Advanced security

[Privacy & Cookies Policy](#)



IT_Academy_VITA_U MR

100 GB, 10 domén

Prehľad

Faturačné nastavenia

Balík služieb

Správa domény

vita.sk



Prehľad

Faturačné nastavenia

DNS



Web



FTP účty

CMS



Emaily



Databázy



Zoznam



Informácie o používateľovi

Používateľia



Prihlasovacie údaje pre používateľa DB_MySQL databázy DB_MySQL

Nastaviť poznámku

Prihlasovacie údaje

Meno DB_MySQL

Login DB_MySQL

Heslo Heslo k účtu nie je možné zistíť dodatočne.

[Zabudnuté heslo?](#)

Odporučané pripojenie

(funguje všade)

Databázový server mysql57.r2.websupport.sk

Port 3311

Pripojenie cez socket

(funguje len na našich serveroch)

Socket /tmp/mysql57.sock

localhost

(funguje len na našich serveroch)

Nastavenia predvoleného databázového servera pre localhost.

Môžete to zmeniť v [mysql connect nastavení](#)

Pomoc a podpora

Správa databázy

PhpMyAdmin

- <https://dbadmin.r2.websupport.sk/>
- môžete si vytvoriť vlastnú "skratku" v sekcií Služby v rámci Web nastavení

Adminer

- <https://adminer.r2.websupport.sk/>

[**> Príklad použitia v PHP skriptoch s mysqli_connect**](#)[**> Príklad použitia v PHP skriptoch s PDO pripojením**](#)



IT_Academy_VITA_U MR

100 GB, 10 domén

Prehľad

Fakturačné nastavenia

Balík služieb

Správa domény

vita.sk



Prehľad

Fakturačné nastavenia



DNS



Web



FTP účty

CMS



Emaily



Databázy



Zoznam



Informácie o používateľovi

Používateľia



Prihlasovacie údaje pre používateľa DB_PostgreSQL databázy DB_PostgreSQL

Nastaviť poznámku

Prihlasovacie údaje

Meno DB_PostgreSQL

Login DB_PostgreSQL

Heslo Heslo k účtu nie je možné zistíť dodatočne.

[Zabudnuté heslo?](#)

Odporučané pripojenie

(funguje všade)

Databázový server postgresql.r2.websupport.sk

Port 5432

Pripojenie cez socket



(funguje len na našich serveroch)

Socket host=localhost

localhost

(funguje len na našich serveroch)

Nastavenia predvoleného databázového servera pre localhost.

Môžete to zmeniť v [mysqli connect nastavení](#)

Pomoc a podpora

Správa databázy

Adminer

- <https://adminer.r2.websupport.sk/>

[Príklad použitia v PHP skriptoch s pg_connect](#) [Príklad použitia v PHP skriptoch s PDO pripojením](#)

Pripojenie na DB

MySQL

- DB_MySQL - Ucime-sa-sql-123
- Databázový server (Host): mysql57.r2.websupport.sk
- Name (DB): DB_MySQL
- Login (User): DB_MySQL
- Heslo (Password): Ucime-sa-sql-123
- Port: 3311
- Socket: /tmp/mysql57.sock

PostgreSQL

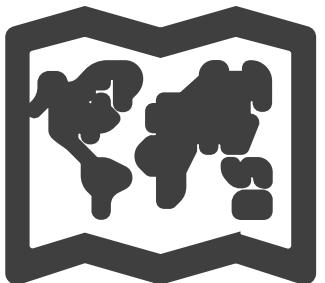
- DB_PostgreSQL - Ucime-sa-sql-123
- Databázový server: postgresql.r2.websupport.sk
- Meno: DB_PostgreSQL
- Login: DB_PostgreSQL
- Heslo (Password): Ucime-sa-sql-123
- Port: 5432
- Socket: host=localhost

Úlohy SQL a noSQL

1. Osvojenie si pojmov SQL a NoSQL
2. Prehľad medzi DBMS
3. Pochopenie funkcií SQL
4. Prihlásenie sa do DBMS



SQL Databázy a Schémy



Čo je Schéma?

- Organizácia DB objektov ako sú tabuľky, pohľady (views), procedúry a iné prvky do logicky zoskupených celkov
- Určuje štruktúru DB objektov a ich vztahy
- Slúži na definovanie, ako sú dátá uložené v DB a ako sa s nimi pracuje, vrátane informácií o dátových typoch, reláciách medzi tabuľkami a ďalších pravdlách



MySQL

Databáza

1. V MySQL, databáza je súbor štruktúr, ktoré obsahujú údaje.
2. Tieto štruktúry zahŕňajú tabuľky, pohľady, indexy, a podobne, ktoré sú uložené na disku.
3. MySQL umožňuje používateľom vytvoriť viac databáz, ktoré môžu byť izolované jedna od druhej, umožňujúc tak organizovať a oddeliť údaje pre rôzne aplikácie alebo účely.

Schéma

1. V MySQL, "schéma" je v podstate synonymom pre "databázu".
2. Pojem "schéma" sa môže tiež odkazovať na štruktúru databázy vrátane definícií tabuliek, dátových typov stĺpcov, relácií medzi tabuľkami (napríklad cudzie klúče) a iných prvkov databázy.
3. V iných SQL databázových systémoch môže mať schéma viac špecifický význam ako logický kontajner, ktorý môže obsahovať viaceré tabuľky a objekty.

MySQL DB a Schémy

- Rozdiel v MySQL je teda viac o **terminológii** ako o **funkčnosti**
- V praxi, keď hovoríme o vytvorení "databázy" alebo "schémy" v MySQL, hovoríme o **vytvorení súboru objektov**, ktoré môžu uchovávať údaje
- Výraz "**schéma**" môže byť použitý **na zdôraznenie štruktúrnej organizácie týchto objektov**, zatiaľ čo "**databáza**" sa zvyčajne používa na **označenie samotného kontajnera údajov**



PostgreSQL

Databáza

1. Databáza predstavuje samostatný databázový systém s vlastnou kolekcou dát a databázových objektov
2. Každá databáza je izolovaná od ostatných databáz
3. Používatelia, ktorí majú prístup k jednej databáze, nemusia mať prístup k inej, pokial' im to nie je explicitne povolené
4. Databázy v PostgreSQL sú vhodné na oddelenie údajov pre rôzne aplikácie alebo rôznych klientov

Schéma

1. Logický kontajner, ktorý môže uchovávať tabuľky, výhľady, indexy a ďalšie databázové objekty.
2. Jedna databáza môže obsahovať viacero schém, čo umožňuje organizovať a spravovať objekty efektívnejšie.
3. Schémy sú užitočné na oddelenie a organizáciu dát a funkcií v rámci jednej databázy.
4. Napríklad, rôzne schémy môžu byť použité pre rôzne oddelenia spoločnosti alebo pre rôzne úrovne prístupu.

PostgreSQL a Schémy

- Databáza je **vrcholná úroveň izolácie**; je to úplne **samostatná entita**, ktorá môže obsahovať **jedno alebo viac schém**
- Schéma je skôr ako **podskupina v rámci databázy**, ktorá pomáha **organizovať a riadiť objekty v rámci tejto databázy**



SQLite

Databáza

1. Celá databáza je uložená v jednom súbore na disku
2. Tento súbor obsahuje všetky dátu, definície tabuliek, indexy, a tak ďalej
3. Databáza v SQLite je teda kompaktný, samostatný súbor, ktorý možno ľahko prenášať medzi systémami alebo aplikáciami

Schéma

1. Označuje štruktúru databázy, t.j. všetky definície databázových objektov ako sú tabuľky, indexy, triggers, a podobne
2. Schéma je teda popis toho, ako sú dátu organizované a ako sa s nimi manipuluje v rámci databázového súboru
3. V SQLite neexistujú oddelené schémy v rámci jednej databázy, ako je to v niektorých iných RDBMS (napr. PostgreSQL); každá databáza má jednu schému

SQLite a Schémy

- Databáza je **fyzický súbor**, ktorý **uchováva údaje** a **informácie o štruktúre** týchto údajov
- Schéma je **popis štruktúry** týchto **údajov** (tabuľky, indexy, atď.) a je inherentnou súčasťou databázového súboru
- V praxi, keď sa hovorí o **schéme v SQLite**, myslí sa tým **obvykle súbor príkazov alebo definícií**, ktoré určujú, ako sú **tabuľky** a **iné objekty vytvorené** a ako sú **medzi sebou vzájomne prepojené**

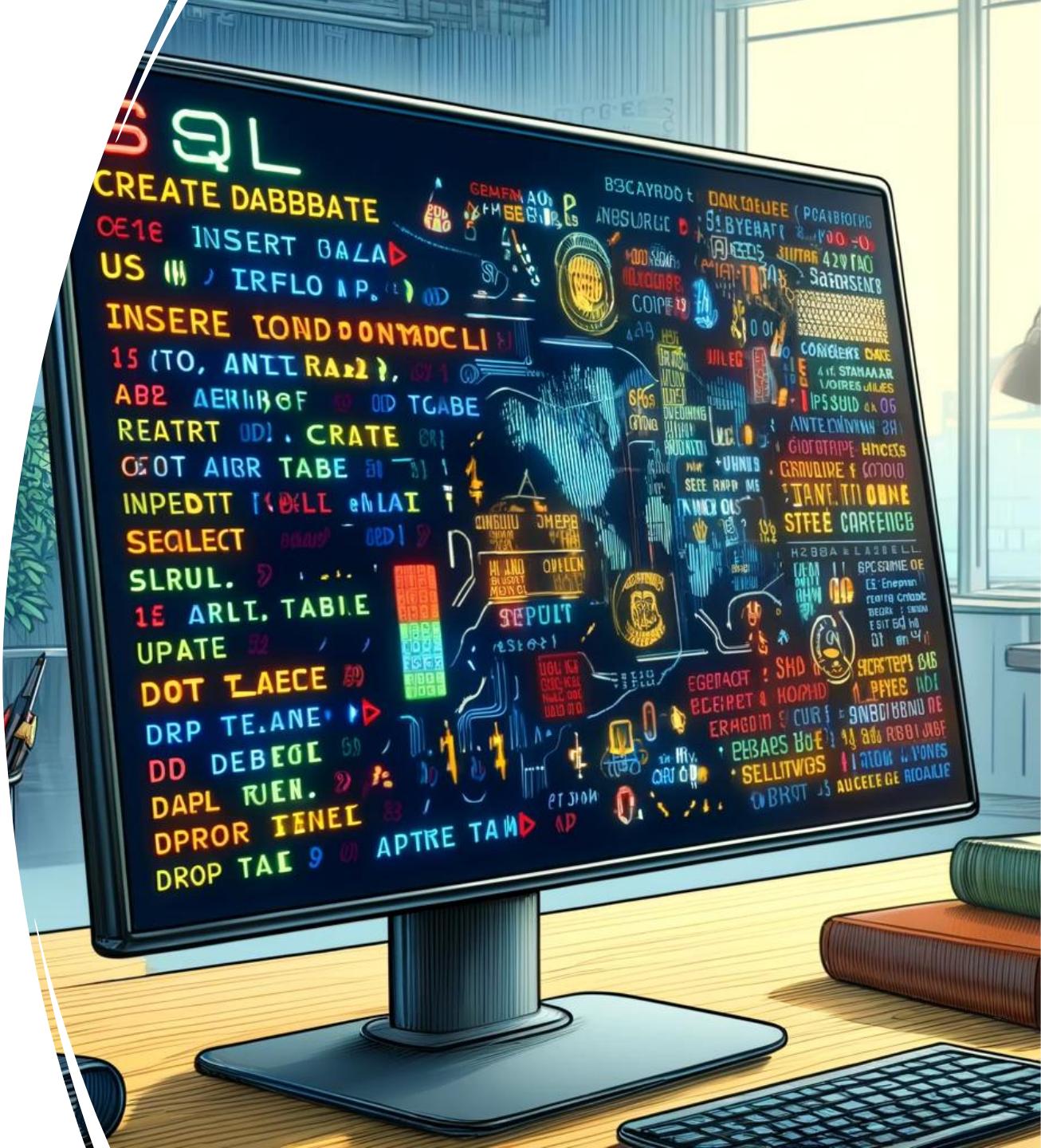


MySQL Schémy (Databázy)

1. **CREATE DATABASE** skola;
2. **USE** skola;
3. **DROP DATABASE** skola;
1. **CREATE TABLE** studenti (
2. id INT AUTO_INCREMENT
 PRIMARY KEY,
3. meno VARCHAR(100),
4. priezvisko VARCHAR(100),
5. vek INT
6.);
7. **INSERT INTO** studenti (meno,
 priezvisko, vek) **VALUES** ('Jana',
 'Nováková', 21);

Úlohy Schémy

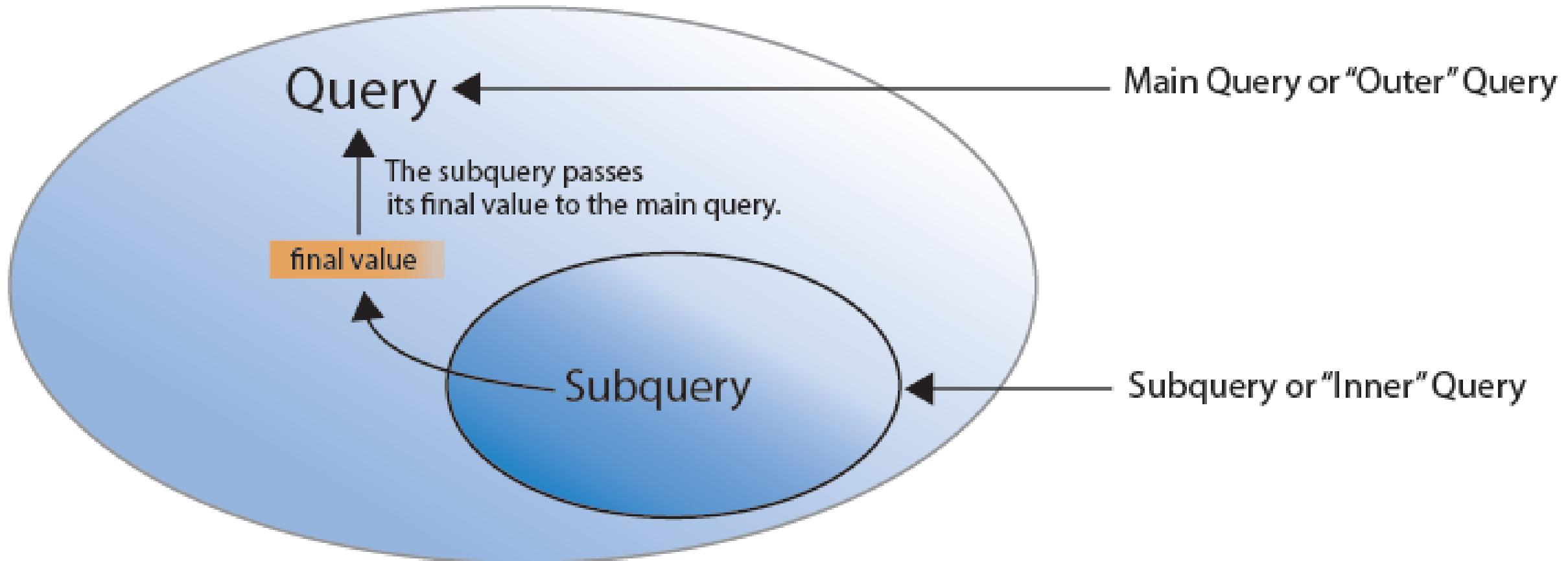
1. Osvojenie si pojmu schéma
2. Pochopenie schém v rôznych DBMS
3. Používanie schém
4. Zobrazenie štruktúry tabuľiek



SQL Poddopyty SubQueries



Vnorené Dopyty



Vnorené Dopyty

Výhody

1. Flexibilita v dopytoch
2. Možnosť zapuzdrenia logiky
3. Zlepšenie čitateľnosti dopytov, pokial sú subquery dobre napísané

Nevýhody

1. Môže spomaliť výkon dopytu, najmä pri korelovaných subquery
2. Môže byť zložité na správu a optimalizáciu v prípade zložitých databázových schém

```
SELECT
    Name, Country, Gender
FROM
    Database
WHERE
    Country IN
```

} Outer Query

```
SELECT
    Country
FROM
    Tourist1
WHERE
    Gender = 'Male'
```

} Sub Query
or
Inner Query

Vnorené Dopyty

```
select
    name,
    area
from
    countries
where country_id in (
    select
        country_id
    from
        countries
    where
        area > 5000000
)
order by
    area,
    name;
```

outer
query

subquery

Table: Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT *
FROM Customers
WHERE age = (
    SELECT MIN(age)
    FROM Customers
);
```

customer_id	first_name	last_name	age	country
2	Robert	Luna	22	USA
3	David	Robinson	22	UK

Table: Customers

customer_id	first_name
1	John
2	Robert
3	David
4	John
5	Betty

```
SELECT customer_id, first_name
FROM Customers
WHERE customer_id IN (
    SELECT customer_id
    FROM Orders
);
```

customer_id	first_name
1	John
2	Robert
4	John
3	David

Table: Orders

order_id	amount	customer_id
1	200	4
2	500	10
3	300	3
4	800	1
5	150	2

Poh'l'ady (SQL Views)

@@



Čo je Pohľad (View)?

- **Virtuálna (derivovaná) tabuľka**
- Vytvára sa na základe SQL dopytu
- **Neukladá fyzicky dátu**
- Poskytuje spôsob, ako **vidieť** dátu z jednej
alebo viacerých tabuľiek zjednotené do
jedného výstupu
- **Nie všetky pohľady sú upraviteľné**



Typy Pohľadov

Na čítanie (Read-Only Views)

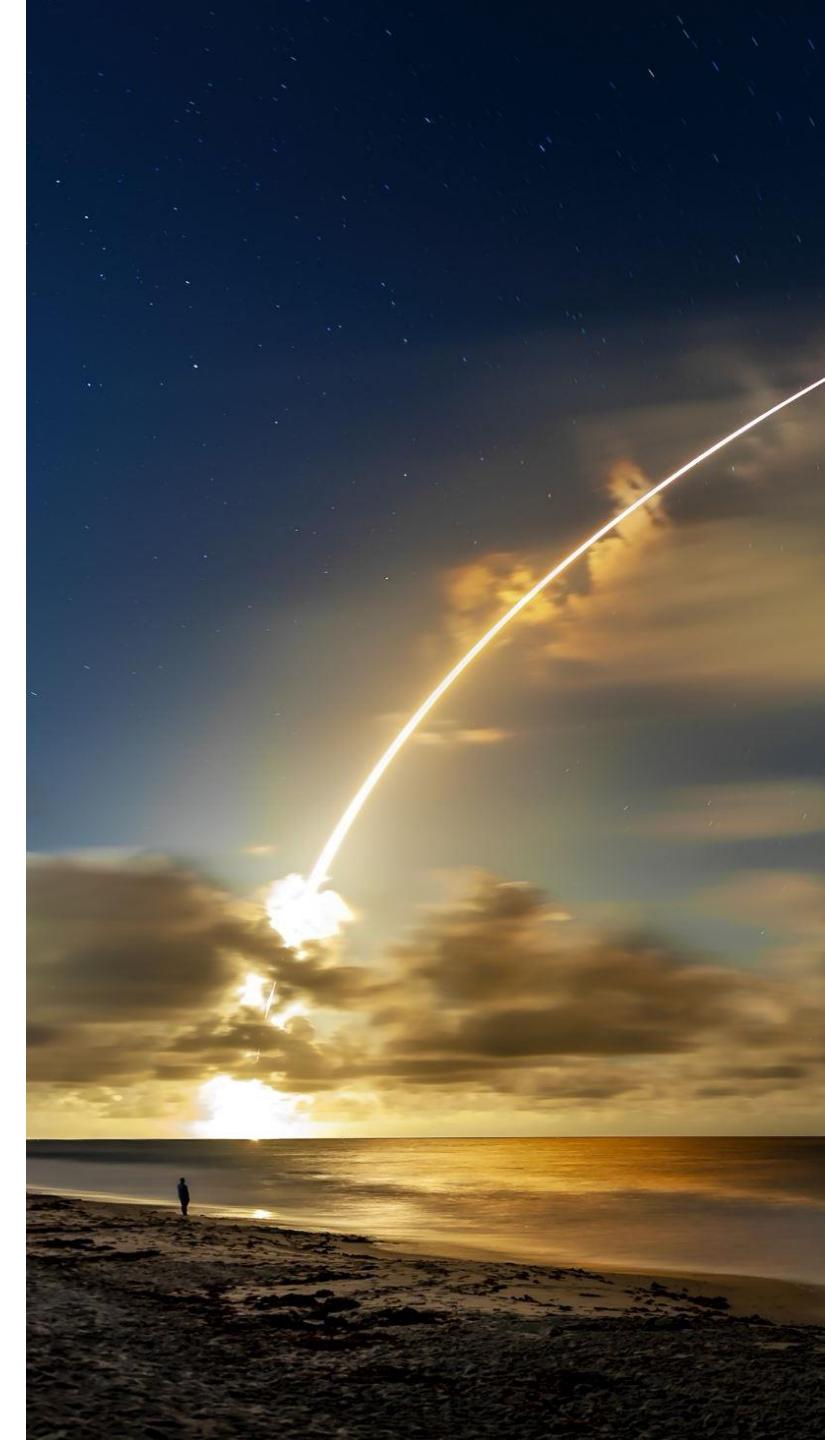
- Tieto pohľady sú **bežnejšie** a sú **obmedzené len na čítanie**
- Nie je možné cez ne vkladať, meniť alebo mazať dátu
- Ak pohľad obsahuje:
 1. Agregácie (SUM, COUNT, AVG, atď.)
 2. Rôzne tabuľky spojené operáciou JOIN
 3. Zoskupenie dát pomocou GROUP BY
 4. Poddopyty v SELECT klauzule
 5. Funkcie, ktoré vrátia rôzne výsledky pri každom volaní, ako RANDOM() alebo CURRENT_TIME()

Updatable Views

- **Niekteré** pohľady **umožňujú vkladanie**, úpravy a mazanie dát
- Tieto pohľady musia splňať určité podmienky, ako napríklad:
 1. Musia obsahovať stĺpce, ktoré jednoznačne identifikujú riadky
 2. Nesmú obsahovať žiadne z vyššie uvedených operácií, ktoré by obmedzovali možnosť zápisu (napríklad agregácie alebo JOINy)

Výhody Pohľadov

- 1. Bezpečnosť:** Pohľady môžu obmedziť prístup k určitým stĺpcom alebo riadkom v tabuľke, čo pomáha chrániť citlivé dátá
- 2. Zjednodušenie zložitých dopytov:** Namiesto toho, aby používatelia museli opakovane písat zložité SQL dopyty, môžu používať pohľady, ktoré tieto dopyty už obsahujú
- 3. Údržba:** Ak dôjde k zmenám v DB schémach, môže byť jednoduchšie aktualizovať pohľady než zmeniť všetky aplikácie, ktoré závisia na týchto dátach



Pohľady (Views)

Vytvorenie

1. **CREATE VIEW** zamestnanci_vyber **AS**
2. **SELECT** meno, priezvisko,
oddelenie
3. **FROM** zamestnanci
4. **WHERE** aktivny = 'ano';

Používanie a Výber

1. **SELECT * FROM** zamestnanci_vyber;

Pohľady (Views)

Aktualizácia

1. **ALTER VIEW** zamestnanci_vyber **AS**
2. **SELECT** meno, priezvisko,
oddelenie, pozicia
3. **FROM** zamestnanci
4. **WHERE** aktivny = 'ano';

Mazanie

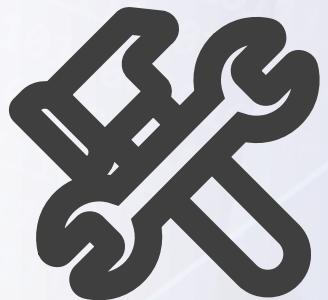
1. **DROP VIEW** zamestnanci_vyber;

Úlohy Pohľady

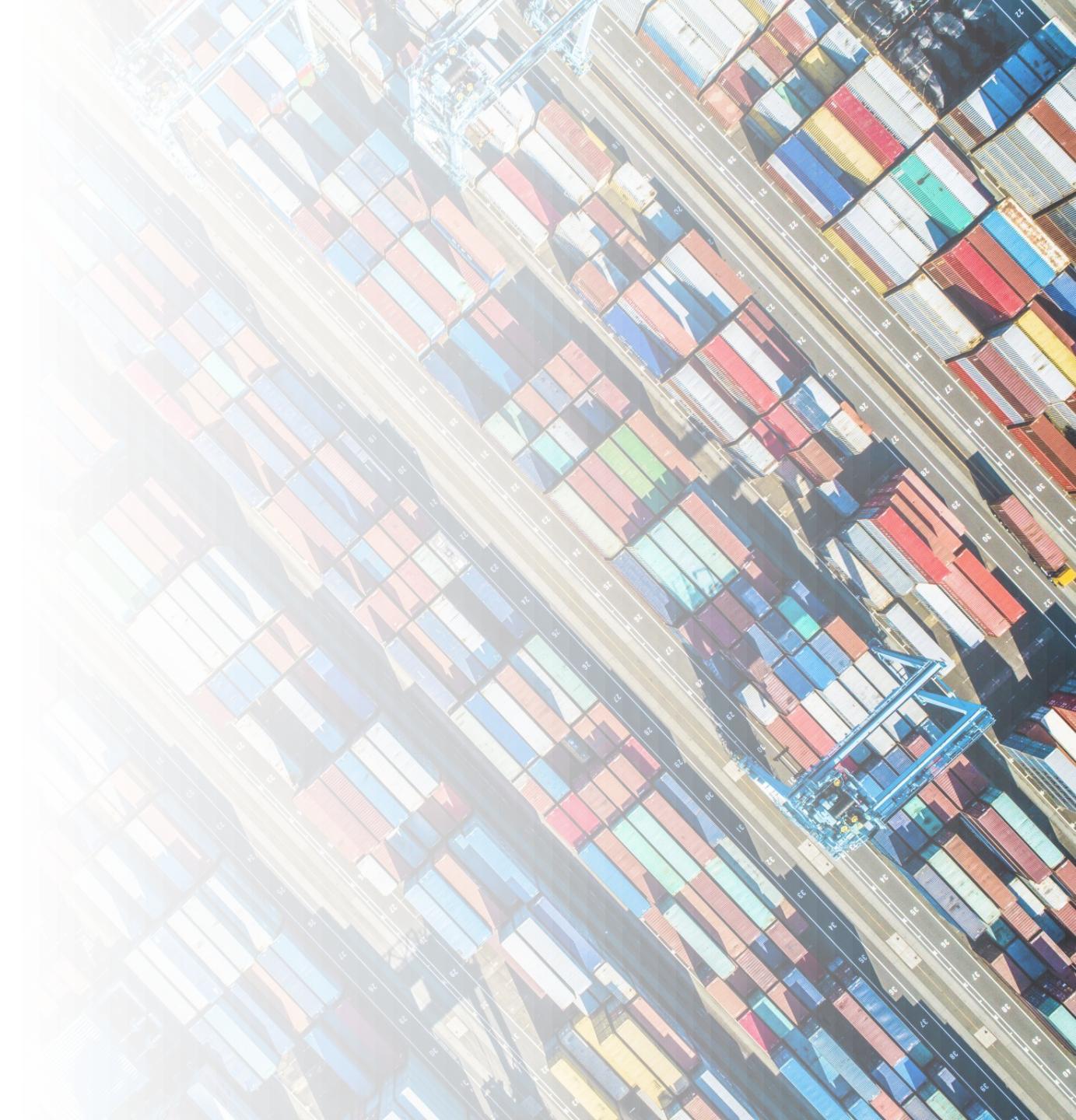
1. Osvojenie si pojmu pohľad (view)
2. Používanie pohľadov
3. Vytvorenie a zmazanie pohľadov



Spracovanie Dát v SQL



Importy csv



SQL Jazyk – DML

- 1. SELECT** – výber dát
- 2. INSERT** – vloženie dát
- 3. UPDATE** – aktualizácia dát
- 4. DELETE** – zmazanie dát
- 5. UPSERT** – aktualizácia a súčasne vkladanie dát



C R U D

INSERT – Vkladanie Dát

- Tvar s/bez určenia stĺpcov
 - **INSERT INTO** tabuľka **VALUES** (1, ‘Hodnota’)
- Tvar s vymenovanými stĺpcami
 - **INSERT INTO** tabuľka (id, value)
VALUES (1, ‘Hodnota’)
- Vkladanie údajov z inej tabuľky
 - **INSERT INTO** tabuľka_1 (id, value)
SELECT id, value **FROM** tabuľka_2

UPDATE - Aktualizácia dát

- **UPDATE** tabuľka **SET**
value=‘Hodnota’
- Nezabudnúť where (filter)

DELETE - Odstránenie Dát

- **DELETE** tabuľka **WHERE** podmienka
- Nezabudnúť where (filter)

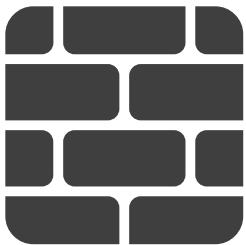
MERGE

- Aktualizovať, vkladať alebo odstraňovať riadky tabuľky na základe zdrojových údajov
- MERGE – spojenie INSERT & UPDATE (štandard SQL 2003)
- **MERGE INTO** cieľová_tabuľka **AS** cieľ
USING (**SELECT** ...) **AS** zdroj
(stĺpce)
ON (porovnávacie kritérium/predikát)
WHEN MATCHED THEN
 UPDATE SET stĺpec =
 zdroj.stĺpec
WHEN NOT MATCHED THEN
 INSERT (stĺpce) **VALUES** ...
 [**WHEN predikát THEN** ...]

REPLACE - Nahradenie Dát

- **REPLACE**(source, old_text, new_text);
- **REPLACE INTO** ciel' (stĺpce) **VALUES** (...)
- **REPLACE INTO** ciel' **SET** stĺpec = ...
- **REPLACE INTO** ciel' (stĺpce) **SELECT** ...
- Aj odstránenie text t.j. nahradenie ničím „“

Materiálizované Pohľady



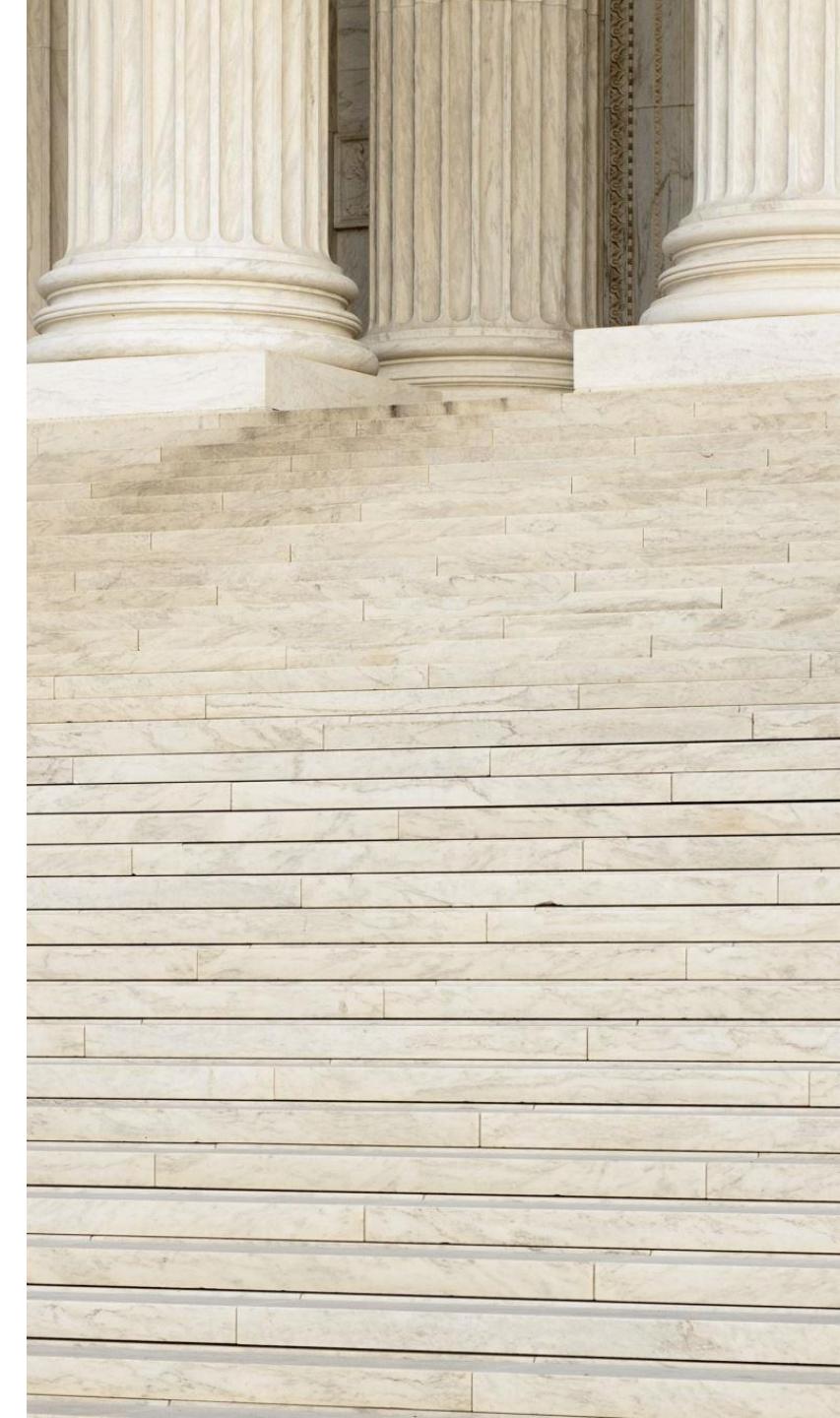
Materializované Pohľady

- Špeciálny typ pohľadu v SQL databázach
- **Fyzicky ukladá výsledky DB dopytu na disk**
- Na rozdiel od štandardného pohľadu, ktorý výsledky len dynamicky generuje pri každom vykonaní
- Toto uloženie na disk umožňuje materiálovým pohľadom poskytovať **rýchlejší prístup k dátam**, pretože **dáta už sú vypočítané a pripravené na použitie**



Použitie Mater. Pohľadov

1. Operácie sú **výpočtovo náročné** a **dopyty zahŕňajú veľké objemy dát alebo zložité výpočty** (napríklad **veľké agregácie** alebo **spojenia**)
2. Dáta sa **nezmenia príliš často**, čo znamená, že **náklady na obnovu materiálového pohľadu** nie sú príliš vysoké v porovnaní s **výhodami rýchleho čítania dát**
3. Je potrebná **rýchla odpoveď** na **dopyty**, ako je to často v **analytických** a **reportovacích aplikáciách**



Vytvorenia Mater. Pohľadu v SQL

1. **CREATE MATERIALIZED VIEW** sales_summary **AS**
2. **SELECT** product_id, **SUM(sales)** **AS** total_sales
3. **FROM** sales
4. **GROUP BY** product_id;

Zobrazenie Aktuálnych Zásob

1. **CREATE MATERIALIZED VIEW** stock_status **AS**
2. **SELECT** product_id, product_name,
SUM(quantity) **AS** total_quantity
3. **FROM** inventory
4. **GROUP BY** product_id, product_name;

Analýzu Návštěvnosti Webstránok

1. **CREATE MATERIALIZED VIEW** daily_web_traffic
AS
2. **SELECT** date, COUNT(visitor_id) **AS**
visitor_count
3. **FROM** web_traffic
4. **WHERE** date >= CURRENT_DATE - INTERVAL '30
days'
5. **GROUP BY** date;

Finančné Sumáre

1. CREATE MATERIALIZED VIEW

```
monthly_financial_summary AS
```

```
2. SELECT MONTH(transaction_date) AS month,  
SUM(sales_amount) AS total_sales, SUM(cost)  
AS total_cost
```

```
3. FROM transactions
```

```
4. GROUP BY MONTH(transaction_date);
```

Sledovanie Výkonu Zamestnancov

1. **CREATE MATERIALIZED VIEW**
employee_performance **AS**
2. **SELECT** employee_id, COUNT(task_id) **AS**
tasks_completed, SUM(hours_logged) **AS** total_hours
3. **FROM** task_logs
4. **GROUP BY** employee_id;

Segmentácia Zákazníkov

```
1. CREATE MATERIALIZED VIEW customer_segments AS
2. SELECT customer_id, SUM(total_purchase) AS
total_spent, CASE
3. WHEN SUM(total_purchase) > 1000 THEN 'High
Value'
4. WHEN SUM(total_purchase) BETWEEN 500 AND 1000
THEN 'Medium Value'
5. ELSE 'Low Value'
6. END AS segment
7. FROM purchases
8. GROUP BY customer_id;
```

Úlohy Mater. Pohľady

1. Osvojenie si pojmu materializovaný pohľad (materialized view)
2. Používanie materializovaných pohľadov
3. Osvojenie si benefitov materializovaných pohľadov



SQL Spojenia (Joins)



Primárny a Cudzí klúč

Primary Key



ID	Name
1	Spencer
2	Steve
3	Mark

Employee

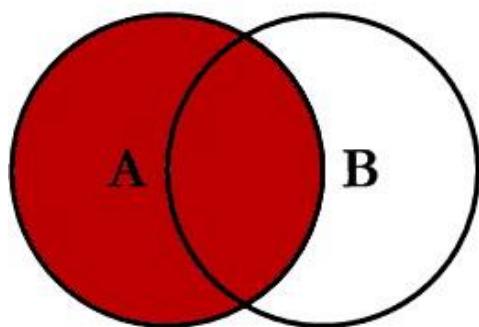
Foreign Key



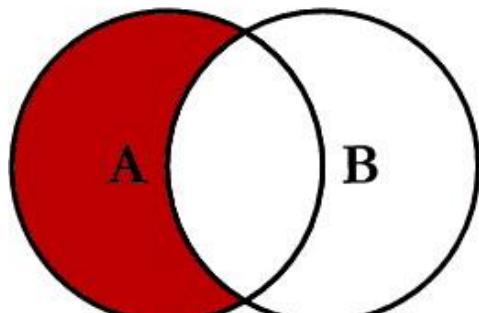
ID	EmployeeID	Equipment
1	1	Desk Comp
2	2	Laptop
3	1	Laptop
4	3	Mobile

Employee Equipment

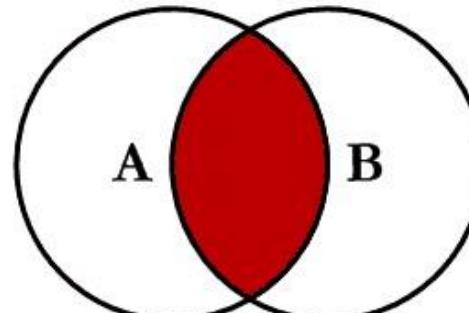
SQL JOINS



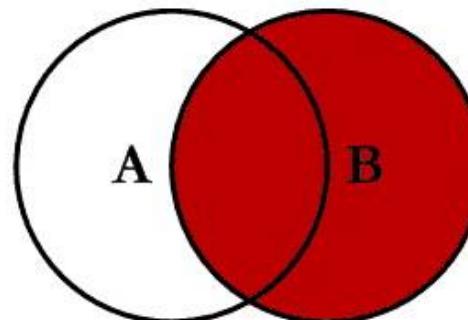
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



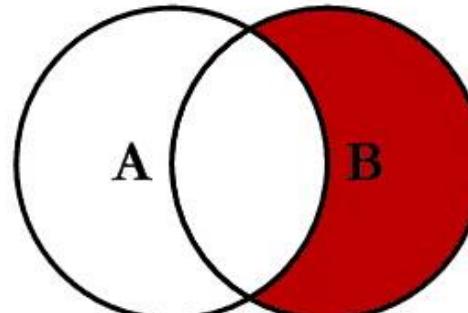
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



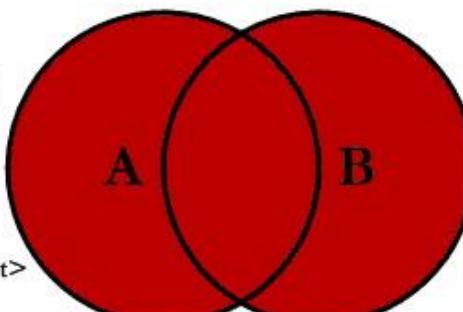
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



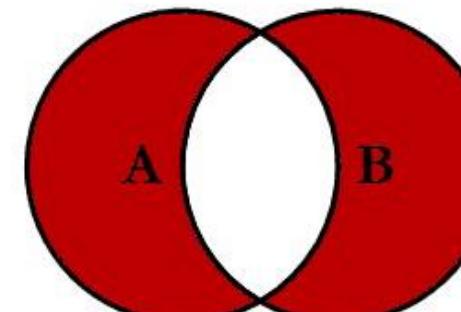
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



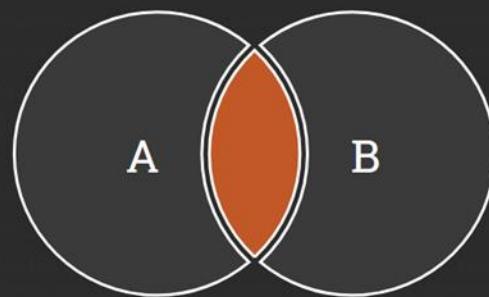
```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```



SQL Joins Visualizer

SQL Joins Visualizer help to you build SQL JOIN between two tables by using of Venn diagrams.

Working offline and as mobile app.



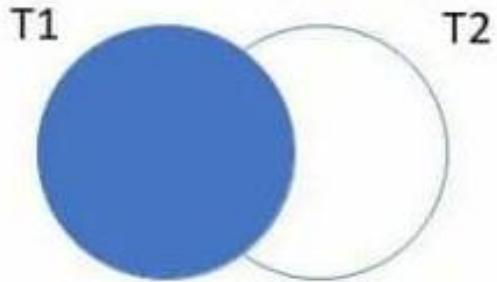
Inner join produces only the set of records that match
in both Table A and Table B

```
SELECT * FROM TableA A  
INNER JOIN TableB B ON  
A.key = B.key
```

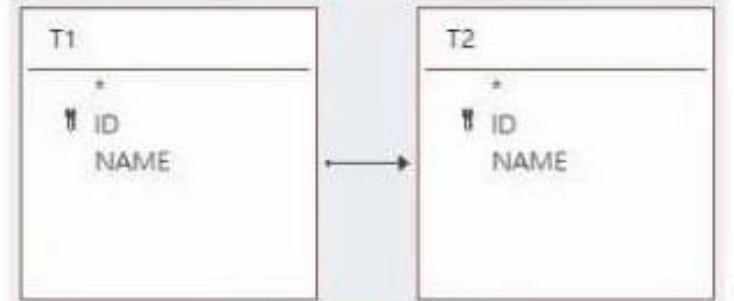
Copy SQL

T1

ID	NAME
1	AAA
2	BBB
3	CCC
T2	
1	CCC
2	DDD
3	EEE

**LEFT JOIN**

T1.ID	T1.NAME	T2.ID	T2.NAME
1	AAA	NULL	NULL
2	BBB	NULL	NULL
3	CCC	1	CCC



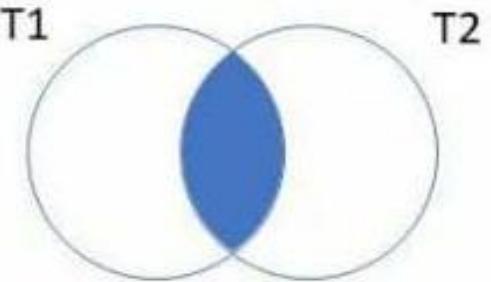
SELECT * FROM T1

LEFT JOIN T2

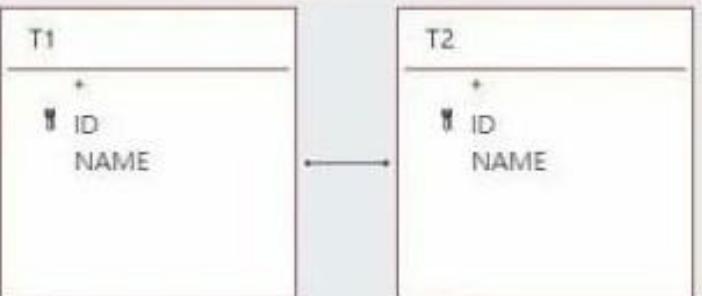
ON T1.NAME = T2.NAME

T1

ID	NAME
1	AAA
2	BBB
3	CCC
T2	
1	CCC
2	DDD
3	EEE

**INNER JOIN**

T1.ID	T1.NAME	T2.ID	T2.NAME
3	CCC	1	CCC



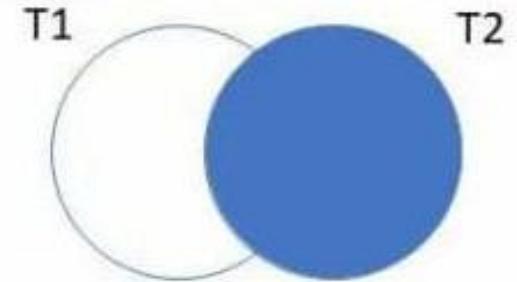
SELECT * FROM T1

INNER JOIN T2

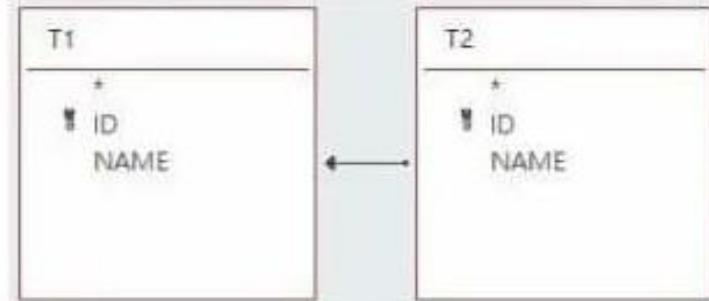
ON T1.NAME = T2.NAME

T1

ID	NAME
1	AAA
2	BBB
3	CCC
T2	
1	CCC
2	DDD
3	EEE

**RIGHT JOIN**

T1.ID	T1.NAME	T2.ID	T2.NAME
3	CCC	1	CCC
NULL	NULL	2	DDD
NULL	NULL	3	EEE



SELECT * FROM T1

RIGHT JOIN T2

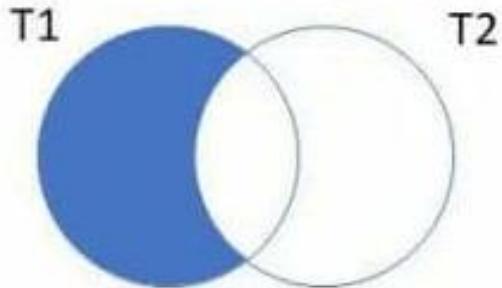
ON T1.NAME = T2.NAME

T1

ID	NAME
1	AAA
2	BBB
3	CCC

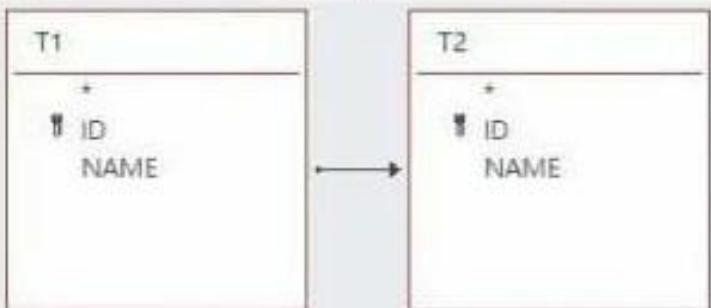
T2

ID	NAME
1	CCC
2	DDD
3	EEE



LEFT JOIN (EXCLUDE T2)

T1.ID	T1.NAME	T2.ID	T2.NAME
1	AAA	NULL	NULL
2	BBB	NULL	NULL



SELECT * FROM T1

LEFT JOIN T2

ON T1.NAME = T2.NAME

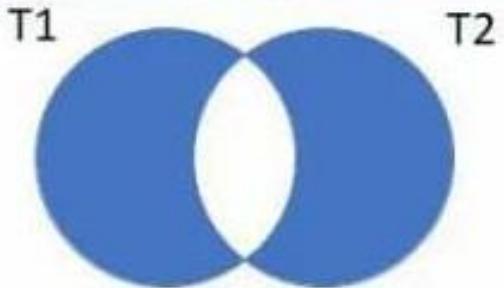
WHERE T2.ID IS NULL

T1

ID	NAME
1	AAA
2	BBB
3	CCC

T2

ID	NAME
1	CCC
2	DDD
3	EEE



UNION ALL (EXCLUDE INNER)

T1.ID	T1.NAME	T2.ID	T2.NAME
1	AAA	NULL	NULL
2	BBB	NULL	NULL
NULL	NULL	2	DDD
NULL	NULL	3	EEE

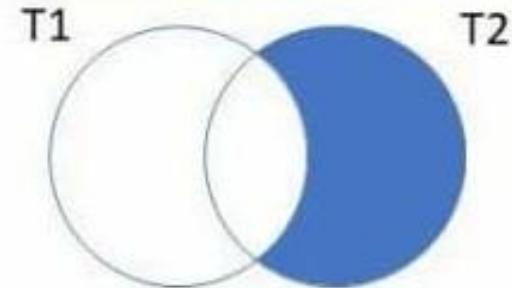
SELECT * FROM T1
LEFT JOIN T2 ON T1.NAME = T2.NAME
WHERE T2.ID IS Null
UNION ALL
SELECT * FROM T1
RIGHT JOIN T2 ON T1.NAME = T2.NAME
WHERE T1.ID IS NULL

T1

ID	NAME
1	AAA
2	BBB
3	CCC

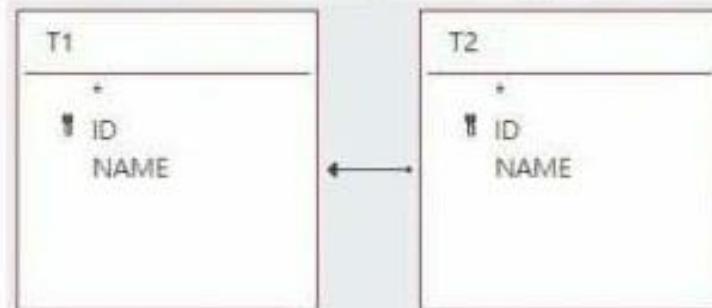
T2

ID	NAME
1	CCC
2	DDD
3	EEE



RIGHT JOIN (EXCLUDE T1)

T1.ID	T1.NAME	T2.ID	T2.NAME
NULL	NULL	2	DDD
NULL	NULL	3	EEE



SELECT * FROM T1

RIGHT JOIN T2

ON T1.NAME = T2.NAME

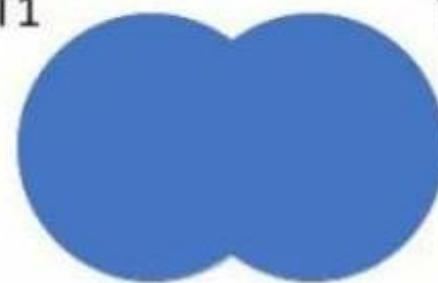
WHERE T1.ID IS NULL

T1	
ID	NAME
1	AAA
2	BBB
3	CCC

T2	
ID	NAME
1	CCC
2	DDD
3	EEE

T1

T2

**FULL OUTER JOIN**

T1.ID	T1.NAME	T2.ID	T2.NAME
1	AAA	NULL	NULL
2	BBB	NULL	NULL
3	CCC	1	CCC
NULL	NULL	2	DDD
NULL	NULL	3	EEE

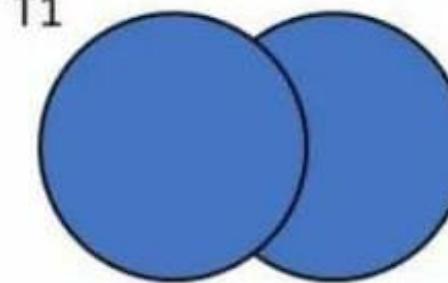
```
SELECT * FROM T1
FULL OUTER JOIN T2
ON T1.NAME=T2.NAME
```

T1	
ID	NAME
1	AAA
2	BBB
3	CCC

T2	
ID	NAME
1	CCC
2	DDD
3	EEE

T1

T2

**UNION ALL**

T1.ID	T1.NAME	T2.ID	T2.NAME
1	AAA	NULL	NULL
2	BBB	NULL	NULL
3	CCC	1	CCC
NULL	NULL	2	DDD
NULL	NULL	3	EEE

```
SELECT * FROM T1
LEFT JOIN T2 ON T1.NAME = T2.NAME
UNION ALL
SELECT * FROM T1
RIGHT JOIN T2 ON T1.NAME = T2.NAME
WHERE T1.ID IS Null
```

SELECT water FROM coconut
WHERE water IS NOT NULL



SELECT * FROM coconut
INNER JOIN mouth ON
coconut.water=mouth.water
WHERE mouth.water IS NOT NULL

Spojenie Tabuliek - Join

1.

There is no data on where the 2032 games will be held. The LEFT JOIN will include a row for 2032 even though it has no corresponding city.

```
SELECT games.yr, city.country  
FROM games JOIN city
```

[Submit SQL](#)

[Restore default](#)

Result:	
yr	country
2004	Australia
2008	Australia
2012	Australia
2032	Australia
2004	Greece
2008	Greece
2012	Greece
2032	Greece
2004	China
2008	China
2012	China
2032	China
2004	UK
2008	UK
2012	UK
2032	UK



Kartézsky súčin

Spojenie Tabuliek - Join

The SELECT .. JOIN statement

Sometimes you need to access two or more tables to get the data required.

games		city	
yr	city	name	country
1896	Athens	Sydney	Australia
1948	London	Athens	Greece
2004	Athens	Beijing	China
2008	Beijing	London	UK
2012	London		

1.

You can use a JOIN to get results from two or more related tables. In this example each row of the table `games` is related to a row of the table `city`. If you want to find the country where the games took place you must JOIN the games table to the city table on the common field that is `games.city` and `city.name`

```
SELECT games.yr, city.country
FROM games JOIN city
ON (games.city = city.name)
```

Result:	
yr	country
1896	Greece
1948	UK
2004	Greece
2008	China
2012	UK

**A SQL query walks up to two tables in a restaurant and asks:
“Mind if I join you?”**

Dopyty zahrňujúce viac ako 1 reláciu (tabuľku)

KARTEZIÁNSKY SÚČIN TABULIEK (RELÁCIÍ) R X S

Tabuľka R

A	B
1	2
3	4

Tabuľka S

C	D
5	6
7	8

Príkaz: SELECT * FROM R;

Príkaz: SELECT * FROM S;

Tabuľka R x S

A	B	C	D
1	2	5	6
1	2	7	8
3	4	5	6
3	4	7	8

Príkaz : SELECT * FROM R,S;

alebo : SELECT * FROM R JOIN S;

Príklady spájania rôznych tabuliek z databázy Restaurants:

SELECT * FROM students JOIN restaurants;

Spojenie tabuliek so selekciami študentov s „rozpočtom“ väčším alebo rovným ako štyridsať euro:

```
SELECT *
FROM students JOIN restaurants
WHERE budget>=40;
```

Spájanie tabuľiek cez korešpondujúce stĺpce – natural join

Tabuľka student_meals			Tabuľka students		
id	student_name	meal_name	name	budget	age
1	Peter	kava	Peter	40	21
2	Lucia	jahody	Lucia	30	20
3	Peter	steak			

Tabuľka student_meals JOIN students ON student_name=name					
id	student_name	meal_name	budget	age	
1	Peter	kava	40	21	
2	Lucia	jahody	30	20	
3	Peter	steak	40	21	

Dotaz spájajúci tabuľky z databázy Restaurants:

```
SELECT * FROM students  
JOIN student_meals ON students.name =student_meals.student_name
```

Rozdiel medzi klauzulou “join ... on” a klauzulou “join ... where”

Z hľadiska výstupu sú nasledovné príkazy identické, t.j. kartézky súčin tabuľiek s podmienkou WHERE je ekvivaletný príkazu JOIN s podmienkou za klauzulou ON. Nasledujúce dva dotazy dávajú identický výsledok:

```
SELECT *
FROM students JOIN student_meals
WHERE students.name= student_meals.student_name
```

```
SELECT *
FROM students JOIN student_meals
ON students.name= student_meals.student_name
```

Najdôležitejší rozdiel medzi použitím klauzúl je, že v druhom prípade je možné modifikovať operáciu JOIN klauzulami LEFT a RIGHT.

Príklad: Vypíšte meno študenta, jeho rozpočet a obľúbené jedlá. S využitím modifikátorov LEFT a RIGHT zobrazte aj študentov, ktorí neobľubujú žiadne jedlo.

```
SELECT *
FROM student_meals RIGHT JOIN students
ON students.name= student_meals.student_name
```

Vnorený príkaz join

MODIFIKÁCIE PRÍKAZU JOIN V MYSQL 5.0	
T1 JOIN T2	kartézsky súčin tabuľiek T1 a T2
T1 JOIN T2 WHERE podmienka	kartézsky súčin tabuľiek T1 a T2 s následne aplikovanou selekciami WHERE
T1 JOIN T2 ON podmienka	identický výsledok ako v predošлом prípade
T1 JOIN T2 ON podmienka rovnosti	špeciálny príklad predošlého príkazu. Tabuľky T1 a T2 sa „spoja“ na základe prvkov v podmienke rovnosti za predpokladu, že sa nachádzajú v obidvoch tabuľkách. Nazýva sa aj EQUIJOIN a je podobný tzv. NATURAL JOIN (ked' atribúty majú v oboch tabuľkách identické meno)
T1 LEFT JOIN T2 ON podmienka rovnosti	v prípade neexistencie zodpovedajúceho prvku v T2 v podmienke doplní hodnoty prvkov z T2 hodnotou NULL
T1 RIGHT JOIN T2 ON podmienka rovnosti	v prípade neexistencie zodpovedajúceho prvku v T1 v podmienke doplní hodnoty prvkov T1 hodnotou NULL

Vnútorné spojenie - Inner join

1.

There is no data on where the 2032 games will be held. The LEFT JOIN will include a row for 2032 even though it has no corresponding city.

```
SELECT games.yr, city.country
FROM games inner JOIN city
ON (games.city = city.name)
```

Submit SQL

Restore default

Result:	
yr	country
2004	Greece
2008	China
2012	UK

SELECT water FROM coconut
WHERE water IS NOT NULL



Page: krumony

```
SELECT * FROM coconut
INNER JOIN mouth ON
coconut.water=mouth.water
WHERE mouth.water IS NOT NULL
```

Žiadny NULL

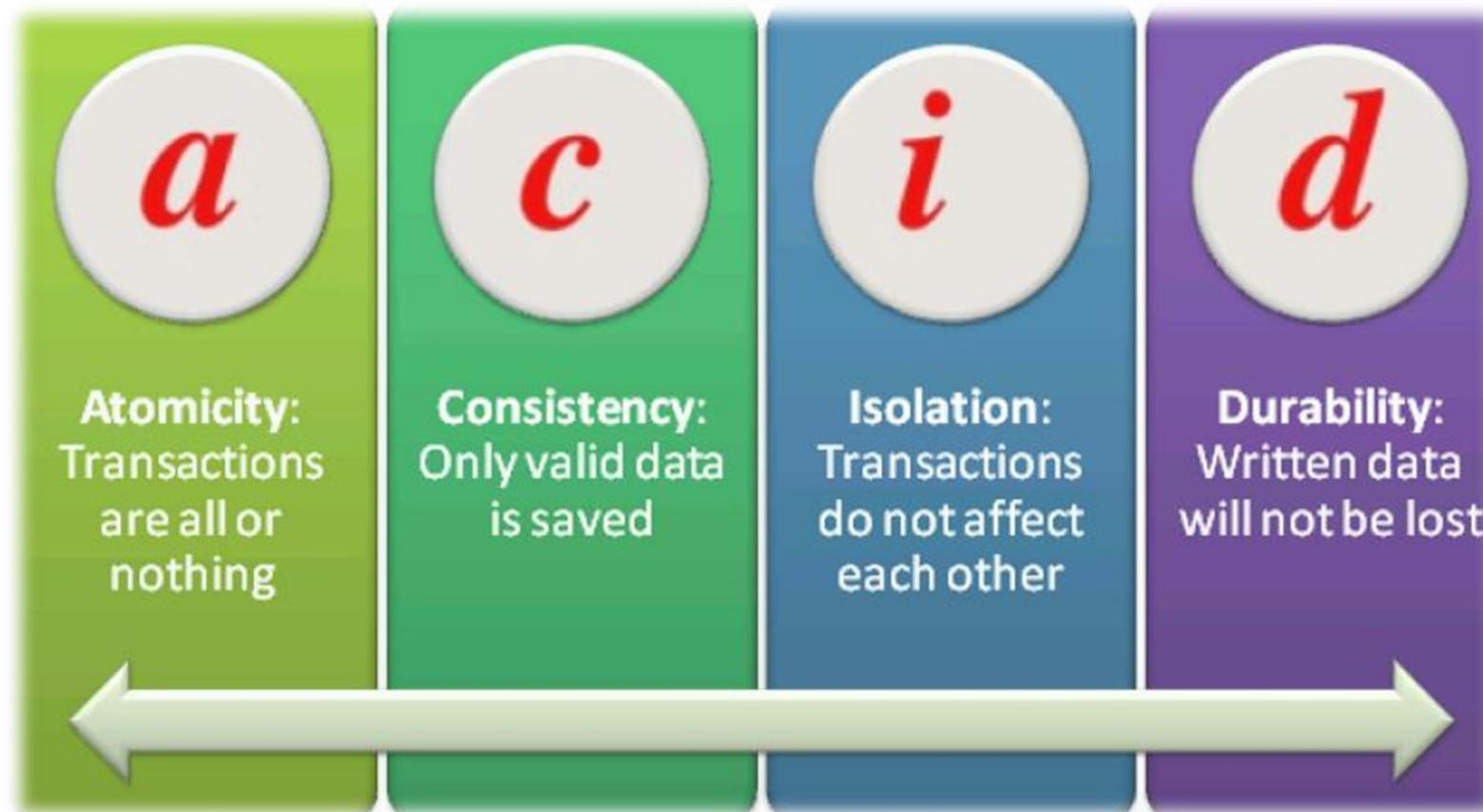
SQL Transakcie



SQL Jazyk – transakcie

- Niektoré DB podporujú transakcie
- Uzatvárajú skupiny DML príkazov do atomických celkov
- Atomický celok sa aplikuje alebo zahodí –
COMMIT, ROLLBACK
- Možnosť uloženia medzivýsledku – **SAVEPOINT**

Databáza - ACID



Transakcie

- Začiatok transakcie
 - **START TRANSACTION**
- Ukončenie transakcie
 - **COMMIT**
 - **ROLLBACK TO SAVEPOINT** názov
- Uloženie stavu
 - **SAVEPOINT** názov
- Toto platí pre PostgreSQL, MySQL.
MS SQL Server používa podobnú syntax.

Uzamykanie tabulek

PostgreSQL

- **LOCK TABLE** tabuľka **IN** režim **MODE**
- Režimy uzamykania:
 1. ACCESS SHARE
 2. ROW SHARE
 3. ROW EXCLUSIVE
 4. SHARE UPDATE EXCLUSIVE
 5. SHARE
 6. SHARE ROW EXCLUSIVE
 7. EXCLUSIVE
 8. ACCESS EXCLUSIVE

MySQL

- **LOCK TABLES** tabuľka **READ|WRITE**, ...
- **UNLOCK TABLES**

Zámky

- **Shared** – Operácie, ktoré neupravujú obsah
- **Update** – Len jedna operácia môže v danom momente upravovať obsah
 - Používa sa ako ochrana proti „Deadlocku“
- **Exclusive** – podobne ako update, ale navyše zabráni čítanie obsahu
- **Intent** – indikuje budúci zámer operácie nad údajmi (nižšia priorita než klasické zámky)
 - Intent shared – čítanie obsahu (nie všetkého)
 - Intent exclusive – modifikácia obsahu (nie všetkého)
 - Shared with intent exclusive – čítanie obsahu a modifikácia jeho časti

Indexy SQL



Čo je index?

- Špeciálne vyhľadávacie tabuľky
- Na urýchlenie vyhľadávania údajov
- Ukazovateľ údajov v tabuľke
- Pomáha zrýchliť **SELECT** dopyty a **klauzuly WHERE**
- Spomaľuje však **vstup dát** s príkazmi **UPDATE** a **INSERT**
- Môžu byť vytvorené alebo zrušené bez efektu na dátu

Predstav si index v knihe

Indexy v skratke

SLOW QUERY - BY PINALDAVE



(C) SQLAuthority.com

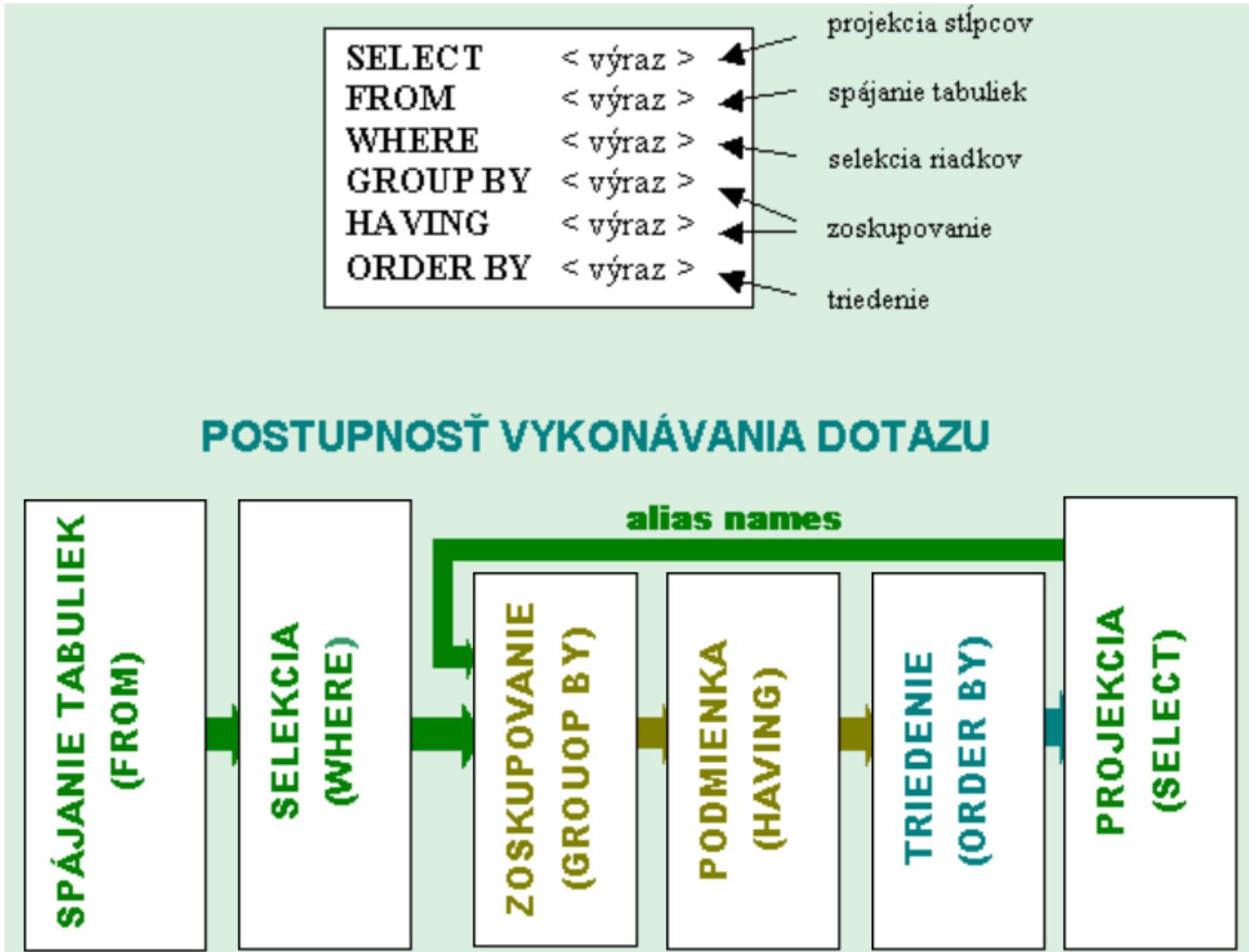


Je OK, ak vykonanie môjho dopytu trvá
2,5 hodiny?

Ako dosiahnut'
najrýchlejší dopyt?

*"The fastest query is
the one you never
make"*

Štruktúra SQL dopytu



Vytvorenie Indexu

- CREATE INDEX umožňuje pomenovať index, špecifikovať **tabuľku** a ktorý **stĺpec** alebo **stĺpce** na indexovanie a označiť, či je index **vzostupne** alebo **zostupne**
- Indexy **môžu byť** tiež **jedinečné**, podobne ako **obmedzenie UNIQUE**
- Zabraňuje **duplicitným záznamom** v **stĺpci** alebo **kombinácií stĺpcov**, na ktorých je index
- Syntax pre vytvorenie:
 - **CREATE INDEX index_meno ON tabulka_meno;**
- Syntax pre vymazanie:
 - **DROP INDEX index_meno;**

Aké typy indexov poznáš?

Typy indexov

B-tree

Hash

GiST

SP-GiST

GIN

- **Jedno-stĺpcové indexy**
 - Index s jedným stĺpcom je index, ktorý je vytvorený **iba na základe jedného stĺpca** tabuľky.

- Základná syntax:

```
CREATE INDEX index_meno  
ON tabulka_meno  
(nazov_stlpca);
```

Predvolene B-tree

Typy indexov

Viacriadkové/členné indexy

- Je definovaný na viac ako jednom stĺpci tabuľky.
- Základná syntax:

```
CREATE INDEX index_meno ON  
tabulka_meno (stĺpec1_meno,  
stĺpec2_meno);
```

Jedinečné indexy

- Nielen pre výkon, ale aj pre integritu údajov.
- Neumožňuje vloženie duplicitných hodnôt do tabuľky.

- Základná syntax:

```
CREATE UNIQUE INDEX index_meno  
ON tabulka_meno (meno_stĺpca);
```

Typy indexov

Čiastkové indexy

- Vytvorený nad **podmnožinou tabuľky**
- Podmnožina je definovaná podmieneným výrazom.
- Index obsahuje položky **iba pre tie riadky** tabuľky, ktoré **uspokojujú predikát**.
- Základná syntax:

CREATE INDEX index_name

ON table_name (**podmienený_výraz**);

Implicitné indexy

- Sú indexy, ktoré **databázový server automaticky** vytvorí pri **vytvorení objektu**.
- Indexy sa **automaticky** vytvárajú pre **obmedzenia primárneho klúča** a **jedinečné obmedzenia**.

Psql \d tabulka, \di

Cvičenia – Indexy

1. Skontroluj si implicitné indexy v konzole
2. Vyrobi klasický index
3. Vyrobi viacstlpový index
4. Vyrobi čiastkový index

CREATE UNIQUE INDEX

nazov_indexu

ON tabulka (stlpec1, stlpec2)

WHERE stlpec **IS NULL**

```
C:\Users\Miroslav>psql -U postgres
Password for user postgres:
psql (9.6.5)
WARNING: console code page (852) differs from windows code page (1250)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

postgres=# \di
                                         List of relations
 Schema |        Name        | Type | owner | Table
-----+----------------+-----+-----+-----+
 public | dummy_pkey    | index | postgres | dummy
 public | svet_pkey     | index | postgres | svet
 public | zamestnanci_pkey | index | postgres | zamestnanci
(3 rows)
```

Indexovanie

Čo indexuj

- Väčšinu cudzích kľúčov
- Bežné WHERE kritériá
- Bežné agregované stĺpce

✓ Nauč sa používať špeciálne typy indexov: výrazové, čiastočné...

Čo neindexuj

- Nemali by sa používať v stĺpcoch, ktoré obsahujú veľký počet hodnôt NULL.
- Malé tabuľky, stĺpce s nízkou kardinalitou
- Tabuľky s častými a veľkými aktualizáciami alebo vkladným operácií

✓ Pozor na veľký počet indexov...

Typy Indexov

B-Tree

- Pre väčšinu typov údajov a dopytov

GIN

- Pre JSONB/range/arrays

GiST

- Pre fulltextové vyhľadávanie a geopriestorové dátové typy

SP-GiST

- Pre väčšie súbory údajov s prirodzeným, ale nerovnomerným zoskupením

BRIN

- Pre skutočne veľké množiny údajov, ktoré sa postupne zoradia

Hash

- Pre operácie ekvivalencie/rovnosti

Nevyužívané indexy v PostgreSQL

```
Administrator: C:\Windows\System32\cmd.exe - psql -e -U postgres
AND PI.indisunique IS FALSE;
   indexname      | tablename
-----+-----
film_fulltext_idx | film
idx_actor_last_name | actor
idx_fk_address_id | customer
idx_fk_city_id | address
idx_fk_country_id | city
idx_fk_customer_id | payment
idx_fk_film_id | film_actor
idx_fk_inventory_id | rental
idx_fk_language_id | film
idx_fk_rental_id | payment
idx_fk_staff_id | payment
idx_fk_store_id | customer
idx_last_name | customer
idx_store_id_film_id | inventory
idx_title | film
(15 rows)

Time: 17,348 ms
dvd_rental=#
```

Duplicitné indexy v PostgreSQL

```
C:\> Administrator: C:\Windows\System32\cmd.exe - psql -e -E -U postgres
SELECT
    indrelid::regclass AS TableName
    ,array_agg(indexrelid::regclass) AS Indexes
FROM pg_index
GROUP BY
    indrelid
    ,indkey
HAVING COUNT(*) > 1;
    tablename | indexes
-----+-----
(0 rows)
```

Time: 2,637 ms
dvd_rental=#

Chceme vidieť všetky indexy

```
SELECT i.relname AS indname,  
       i.relnowner AS indowner,  
       idx.indrelid::regclass,  
       am.amname AS indam,  
       idx.indkey,  
       ARRAY(  
           SELECT pg_get_indexdef(idx.indexrelid, k + 1, true)  
           FROM generate_subscripts(idx.indkey, 1) AS k  
           ORDER BY k  
       ) AS indkey_names,  
       idx.indexprs IS NOT NULL AS indexprs,  
       idx.indpred IS NOT NULL AS indpred  
  FROM pg_index AS idx  
  JOIN pg_class AS i  
  ON i.oid = idx.indexrelid  
  JOIN pg_am AS am  
  ON i.relam = am.oid;
```

```
SELECT  
       U.username      AS user_name,  
       ns.nspname      AS schema_name,  
       idx.indrelid :: REGCLASS AS table_name,  
       i.relname        AS index_name,  
       idx.indisunique  AS is_unique,  
       idx.indisprimary AS is_primary,  
       am.amname        AS index_type,  
       idx.indkey,  
       ARRAY(  
           SELECT pg_get_indexdef(idx.indexrelid, k + 1, TRUE)  
           FROM generate_subscripts(idx.indkey, 1) AS k  
           ORDER BY k  
       ) AS index_keys,  
       (idx.indexprs IS NOT NULL) OR (idx.indkey::int[] @> array[0]) AS is_functional,  
       idx.indpred IS NOT NULL AS is_partial  
  FROM pg_index AS idx  
  JOIN pg_class AS i  
  ON i.oid = idx.indexrelid  
  JOIN pg_am AS am  
  ON i.relam = am.oid  
  JOIN pg_namespace AS NS ON i.relnamespace = NS.OID  
  JOIN pg_user AS U ON i.relnowner = U.userysid  
 WHERE NOT nspname LIKE 'pg%'; -- Excluding system tables
```

Questions a test time!



The 3-Minute Test

Expert, competent, or just a greenhorn?
Test your SQL performance skills in just three minutes. >

Table of Contents

- Preface
- Anatomy of an Index
- The Where Clause
- Performance and Scalability
- The Join Operation
- Clustering Data
- Sorting and Grouping
- Partial Results
- Modifying Data
- Execution Plans
- Myth Directory
- Example Schema
- Glossary

= SQL Indexing and Tuning e-Book

A site explaining SQL indexing to developers—no crap about administration.

SQL indexing is the most effective tuning method—yet it is often neglected during development. *Use The Index, Luke* explains SQL indexing from grounds up and doesn't stop at ORM tools like Hibernate.

Use The Index, Luke is the free web-edition of SQL Performance Explained. If you like this site, consider getting the book. Also have a look at the shop for other cool stuff that supports this site.

SQL Indexing in MySQL, Oracle, SQL Server, etc.

Use The Index, Luke presents indexing in a vendor agnostic fashion. Product specific notes are provided like here:

DB2 MySQL Oracle PostgreSQL SQL Server

Use The Index, Luke covers SQL indexing for PostgreSQL. Tests were conducted with PostgreSQL 9.0 through 9.6.

Have more questions about SQL indexing or tuning? No problem—have a look at my training and tuning services at <http://winand.at>.



English
by Markus Winand.

Email address

Subscribe blog

Tweet

@MarkusWinand

@SQLPerfTip

vote

Vote



DO NOT USE
OFFSET FOR
PAGINATION

Learn
why

Príkaz VACUUM

- Používa sa na **obnovenie ukladacieho priestoru odstránením zastaralých údajov alebo n-tíc z databázy**
- **Zakaždým**, keď vykonávate **aktualizáciu v tabuľke**, sa **pôvodný záznam uchováva** v databáze.
- Vacuum **odstráni tieto staré záznamy** t.j. n-tice a zníži veľkosť databázy
- Môžete vakuovať **iba tie tabuľky**, v ktorých **máte oprávnenia** VACUUM.
- V rámci **transakcie nemôžete** spustiť príkaz VACUUM.

```
VACUUM [FULL] [FREEZE] [VERBOSE] [table_name];
```

OR

```
VACUUM [FULL] [FREEZE] [VERBOSE]
ANALYZE table_name [ (col1, col2, ... col_n) ];
```

Najdôležitejší príkaz v PostgreSQL

Odporúčania na VACUUM

- VACUUM spúšťať orientačne raz denne
- VACUUM FULL raz mesačne
- To "orientačne" znamená, že približne dôjde k zmene 10 % riadkov tabuľky
- Príkaz VACUUM FULL môžete urýchliť odstránením indexov
- Po dokončení príkazu indexu znova vytvoríte

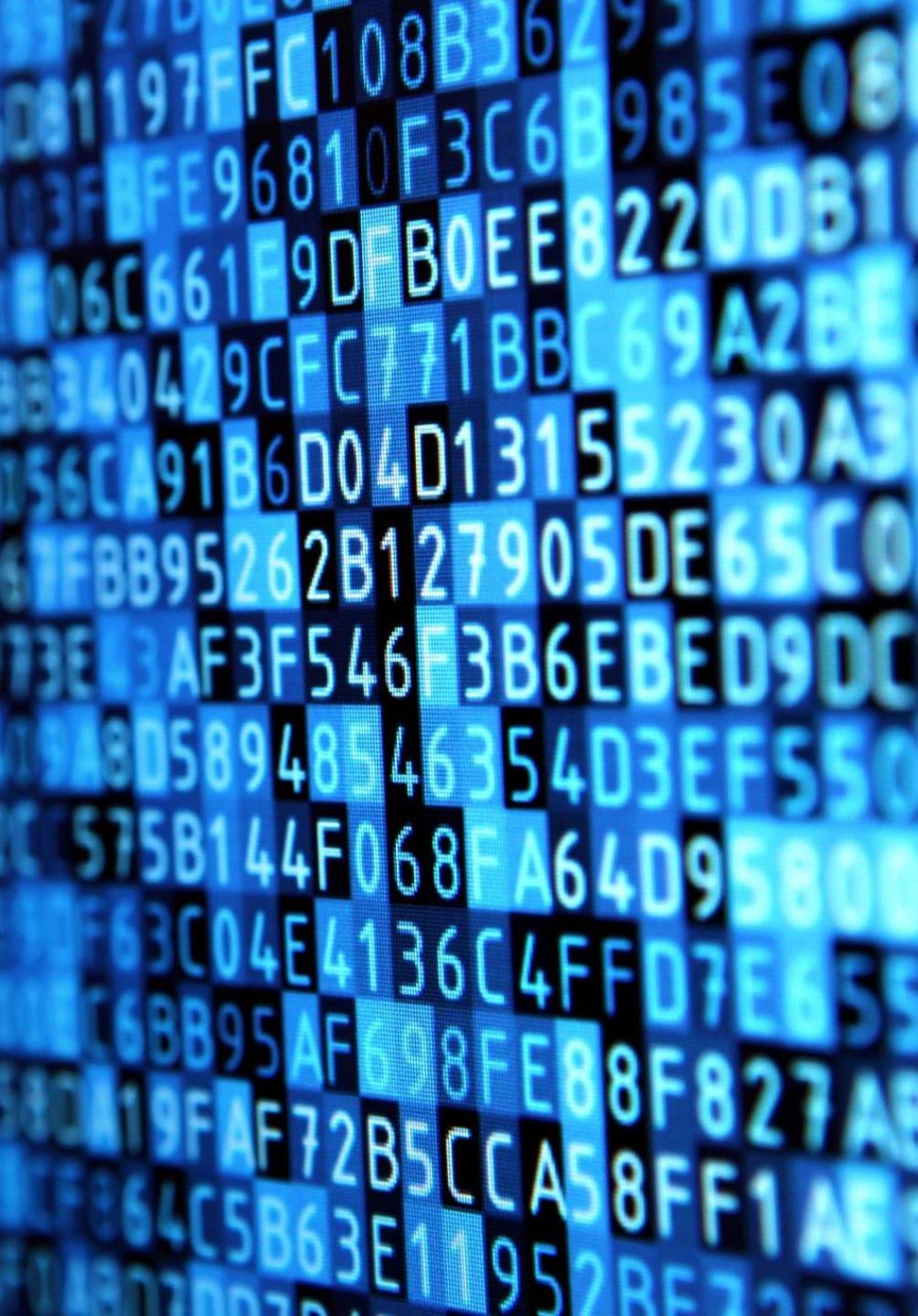
Od 8.4 Autovacuum

Nástroje na SQL a Databázy



Nástroje na Správu Databáz

- 1.**Dbeaver**
- 2.**DataGrip**
- 3.**phpMyAdmin**
- 4.**MySQL Workbench**
- 5.**PgAdmin**
- 6.**Microsoft SQL Server Management Studio (SSMS)**
- 7.**Oracle SQL Developer**
- 8.**SQLite Browser**



Čo si Má Nainštalovať?

110

- DBeaver
- Jetbrains Datalore
- Jetbrains DataGrip
- DbVisualizer
- MySQL Server + Workbench
- WAMP | XAMPP | LAMP
- phpMyAdmin





DBeaver Community

Free Universal Database Tool

Star 30,662



Follow @dbeaver_news

search here ...

Go

[Home](#) [About](#) [Download](#) [Sources](#) [Documentation](#) [News](#) [Support](#) [DBeaver PRO](#) [CloudBeaver](#)

Universal Database Tool

Free multi-platform database tool for developers, database administrators, analysts and all people who need to work with databases. Supports all popular databases: MySQL, PostgreSQL, SQLite, Oracle, DB2, SQL Server, Sybase, MS Access, Teradata, Firebird, Apache Hive, Phoenix, Presto, etc.

[Download](#)

Address	T	City	T	State	T	Country	T	Name	Level	Type	Table Name
Ge 7 Block 0		Brasilia	SP	Brazil	75			retail_cus...	0	INTEGER	Customer
#210 111 St NW		Edmonton	AB	Canada	76			first...	1	CHAR	Customer
793 W Pender Street		Vancouver	BC	Canada	78			lastNa...	2	CHAR	Customer
1800 Amphitheatre Parkway		Mountain View	CA	USA	94			co...	3	CHAR	Customer
1 Microsoft Way		Redmond	WA	USA	95			addres...	4	CHAR	Customer
827 Broadway		New York	NY	USA	96			city...	5	CHAR	Customer
1 Infinite Loop		Cupertino	CA	USA	98			state...	6	CHAR	Customer
541 Del Medio Avenue		Mountain View	CA	USA	94			country...	7	CHAR	Customer
801 N Alh Street		Reno	NV	USA	89			post...	8	CHAR	Customer
120 S Orange Ave		Orlando	FL	USA	32			phone...	9	CHAR	Customer
Teufelsstrasse 8		Berlin		Germany	10			fax...	10	CHAR	Customer
88 Salem Street		Boston	MA	USA	21			email...	11	CHAR	Customer
182 E Superior Street		Chicago	IL	USA	80			supp...	12	CHAR	Customer
319 N. Frances Street		Madison	WI	USA	53						
2211 W Berry Street		Fort Worth	TX	USA	78						
1033 N Park Ave		Tucson	AZ	USA	85						
302 S 700 E		Salt Lake City	UT	USA	84						
788 Dundas Street West		Toronto	ON	Canada	M1						
230 Elgin Street		Ottawa	ON	Canada	K2						
194A Chain Lake Drive		Halifax	NS	Canada	B3						
889 Osborne Street		Winnipeg	MB	Canada	R3						
5112 48 Street		Yellowknife	NT	Canada	X1						



DBeaver 23.2.1

File Edit Navigate Search SQL Editor Database Window Help



Enter a part of object name here
> DB_MySQL - mysql57.r2.websupport.sk:3311
> DB_PostgreSQL - postgresql.r2.websupport.
> DBeaver Sample Database (SQLite)
> Kurz_SQL - mysql57.r2.websupport.sk:3311

Connect to a database

Connection Settings

PostgreSQL connection settings

Main PostgreSQL Driver properties SSH SSL

+ Network configurations...

Server

Connect by: Host URL

URL: jdbc:postgresql://surus.db.elephantsql.com:5432/ovuevjvl

Host: surus.db.elephantsql.com Port: 5432

Database: ovuevjvl

Authentication

Authentication: Database Native

Username: ovuevjvl

Password: XXXXXXXXXXXXXX Save password

Advanced

Session role: Local Client: PostgreSQL Binaries

[You can use variables in connection parameters.](#) [Connection details \(name, type, ...\)](#)

Driver name: PostgreSQL

Driver Settings Driver license

Test Connection ... < Back Next > Finish Cancel

Connection test

Connected (1304 ms)

Server: PostgreSQL 11.18 (Ubuntu 11.18-1.pgdg20.04+1) PostgreSQL 11.18 (Ubuntu 11.18-1.pgdg20.04+1) on x86_64-pc-linux-gnu, compiled by gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0, 64-bit

Driver: PostgreSQL JDBC Driver 42.5.2

OK Details >



The world's most popular open source database



Contact MySQL | Login | Register

MySQL™

MySQL.com

Downloads

Documentation

Developer Zone

Products

Cloud

Services

Partners

Customers

Why MySQL?

News & Events

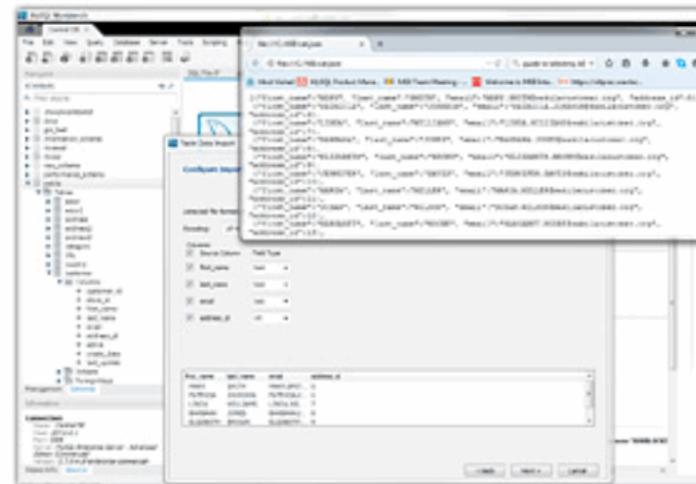
How to Buy

- MySQL HeatWave
- ❖ MySQL Enterprise Edition
 - Datasheet (PDF)
 - Technical Specification
 - MySQL Database
 - MySQL Document Store
 - Oracle Enterprise Manager
 - Enterprise Monitor
 - Enterprise Backup

MySQL Workbench

Enhanced Data Migration

[Download Now »](#)



MySQL Workbench is a unified visual tool for database architects, developers, and DBAs.

MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more

MySQL Workbench is available on Windows, Linux and Mac OS X.

https://www.mysql.com/products/enterprise/high_availability.html





Many databases, one tool

[Download](#)[Free 30-day trial](#)

Why DataGrip

Enjoy working with databases



Dashboard

Welcome, [itacademysk](#)**CPU Usage:** 0% used – 0.00s of 100s. Resets in 1 hour, 56 minutes [More Info](#)**File storage:** 0% full – 100.0 KB of your 512.0 MB quota [More Info](#)[Upgrade Account](#)

Recent Consoles

[+ 5 -](#)*You have no recent consoles.*

New console:

[\\$ Bash](#)[>>> Python ▾](#)[More...](#)

Version

[2.7](#)[3.7](#)[3.8](#)

Recent Files

[+ 5 -](#)

/home/itacademysk/.bashrc
/home/itacademysk/.gitconfig
/home/itacademysk/.profile
/home/itacademysk/.pythonstartup.py
/home/itacademysk/.vimrc

[+ Open another file](#)[Browse files](#)

Recent Notebooks

[+ 5 -](#)

Your account does not support Jupyter Notebooks. [Upgrade your account](#) to get access!

All Web apps

You don't have any web apps.[Open Web tab](#)

[MySQL](#)[Postgres](#)

Initialize MySQL

Let's get started! The first thing to do is to initialize a MySQL server:

Enter a new password in the form below, and note it down: you'll need it to access the databases once you've created them. You will only need to do this once.

New password:

A password input field with a small circular icon containing a question mark in the top right corner.

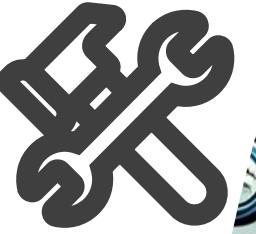
Confirm password:

A password input field with a small circular icon containing a question mark in the top right corner.

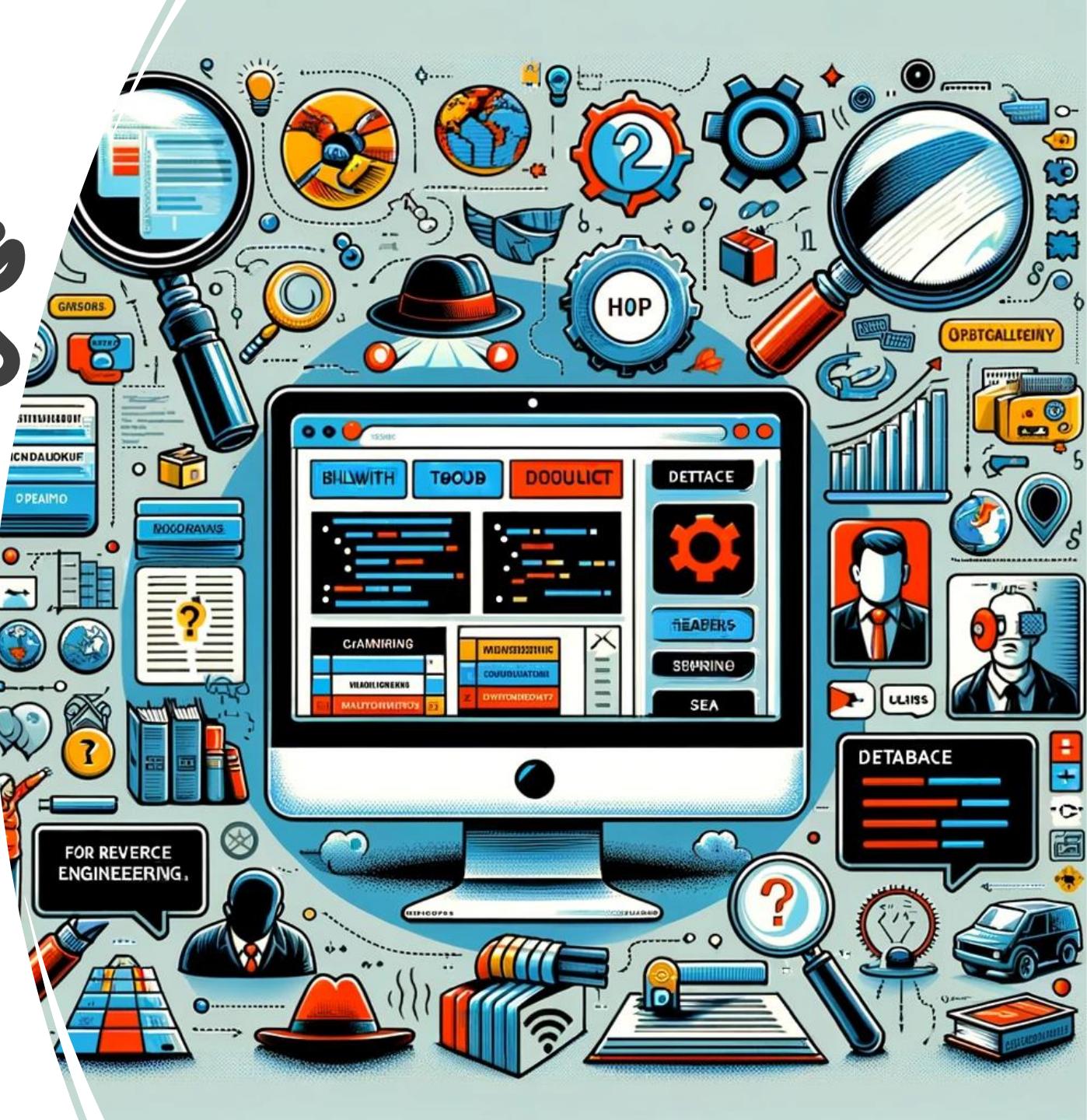
Initialize MySQL

This should be different to your main PythonAnywhere password, because it is likely to appear in plain text in any web applications you write.

Úlohy Nástroje



1. Zorientovanie v DBMS nástrojoch
2. Inštalácia a použitie DBeaver
3. Vyskúšanie phpmyadmin
4. Používanie Jetbrains Datalore



Dôležité Zdroje

Slovensko a Česko

- Albatrosmedia
- Kopp
- Grada
- Wolters Kluwer
- BEN
- Veda

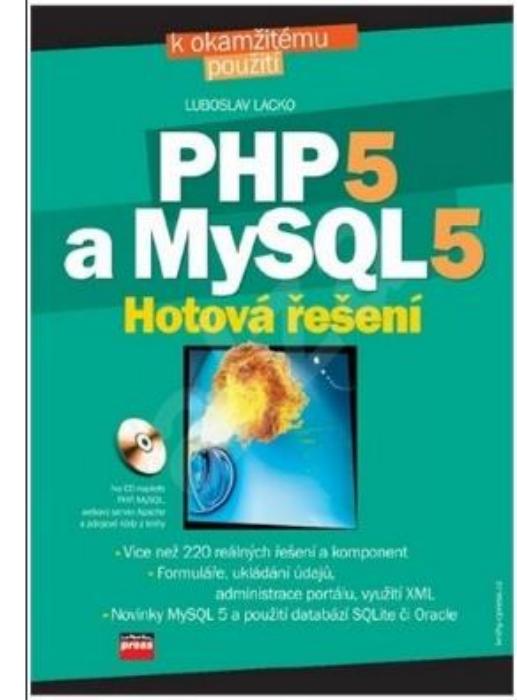
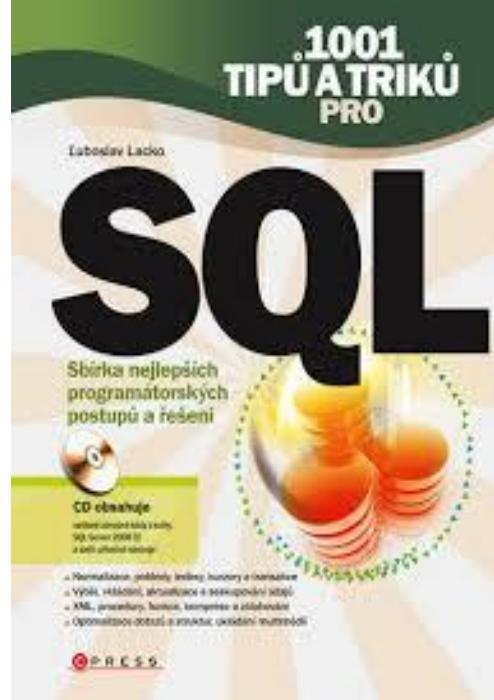
Zahraničie

- O'Reilly
- Manning
- Packt
- Apress
- Wiley
- No Starch Press

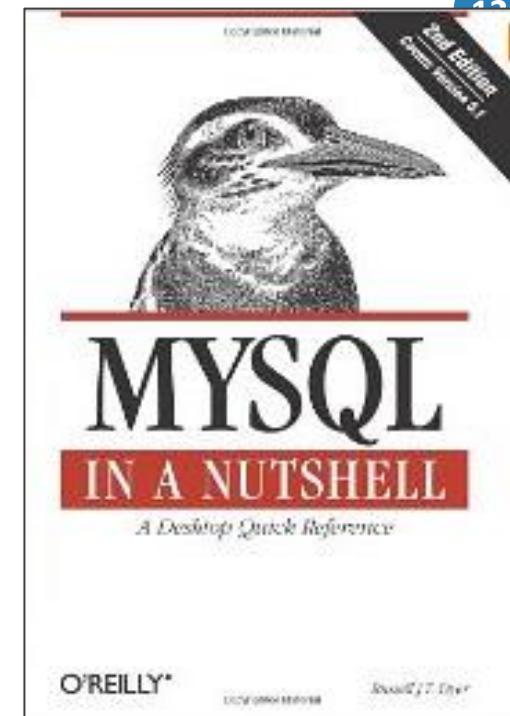
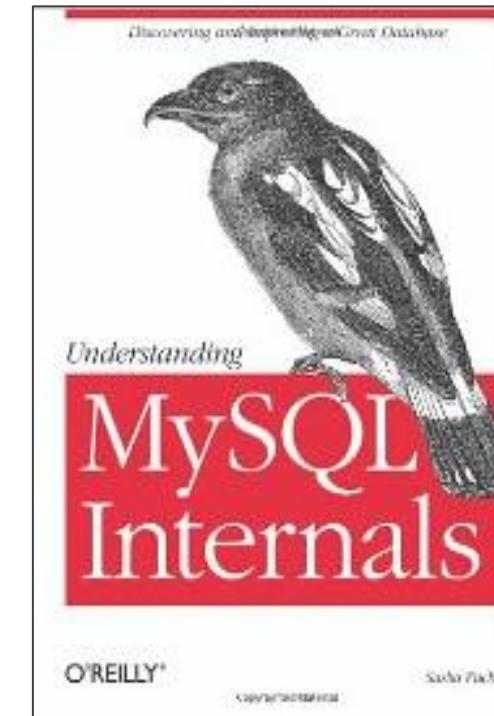
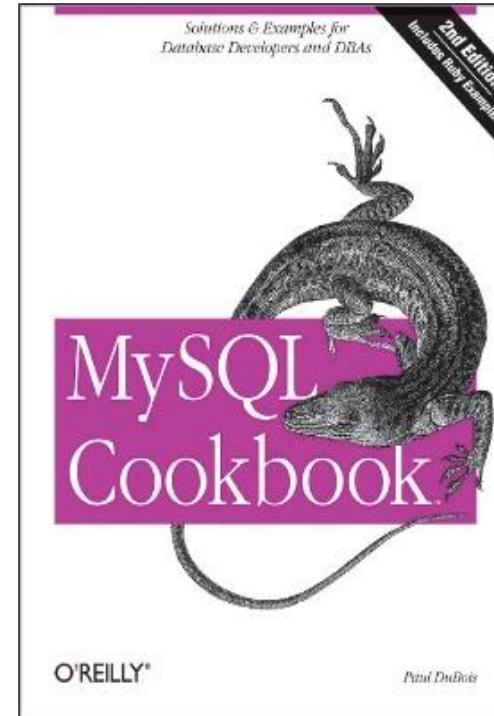
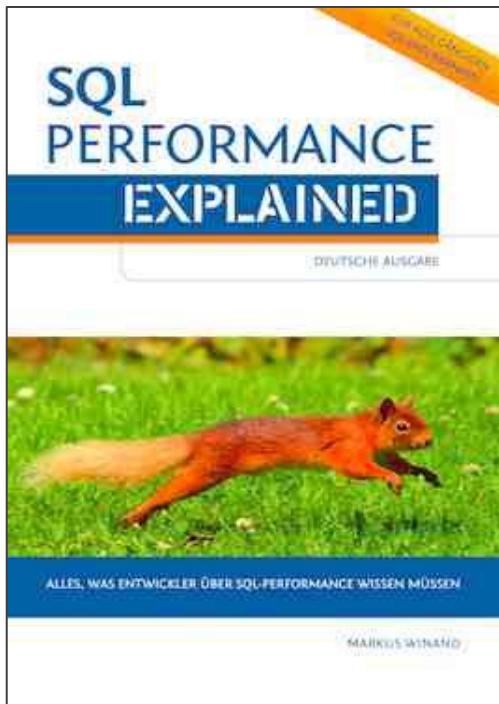
YouTube Tutoriály

- [IT Academy](#)

IT Academy



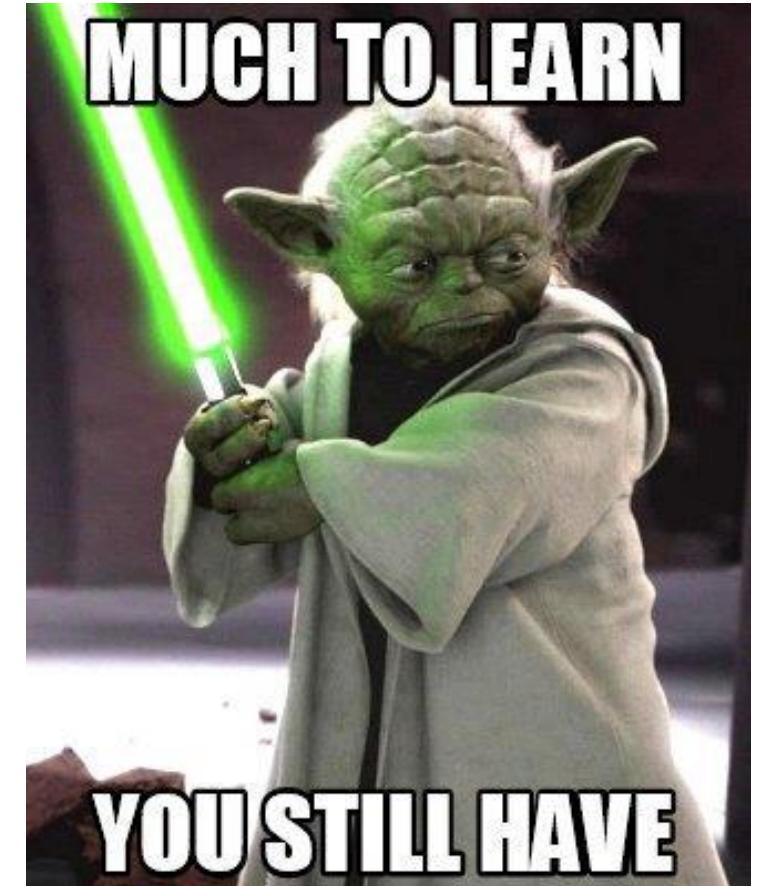
Čo sa oplatí/neoplatí prečítať SK/CZ?



Čo sa oplatí/neoplatí prečítať EN?

Čo Odporúčam si Pozriet'?

1. www.slideshare.net/osscube/indexing-the-mysql-index-key-to-performance-tuning
2. www.cs.duke.edu/csl/docs/mysql-refman/optimization.html
3. <https://cloud.google.com/spanner/docs/sql-best-practices>
4. www.use-the-index-luke.com/
5. <https://www.metabase.com/learn/sql-questions/sql-best-practices>
6. <https://www.sqlstyle.guide/>
7. <https://about.gitlab.com/handbook/business-technology/data-team/platform/sql-style-guide/>
8. https://docs.telemetry.mozilla.org/concepts/sql_style.html



SQL Style Guide

by [Simon Holywell](#) · [@Treffynnon](#)

 [Tweet](#)

- [Overview](#)
- [General](#)
 - [Do](#)
 - [Avoid](#)
- [Naming conventions](#)
 - [General](#)
 - [Tables](#)
 - [Columns](#)
 - [Aliasing or correlations](#)
 - [Stored procedures](#)
 - [Uniform suffixes](#)
- [Query syntax](#)
 - [Reserved words](#)
 - [White space](#)
 - [Indentation](#)
 - [Preferred formalisms](#)
- [Create syntax](#)
 - [Choosing data types](#)
 - [Specifying default values](#)
 - [Constraints and keys](#)
 - [Designs to avoid](#)
- [Appendix](#)
 - [Reserved keyword reference](#)
 - [Column data types](#)

Overview

TOP Klávesové Skratky

TOP Klávesové Skratky

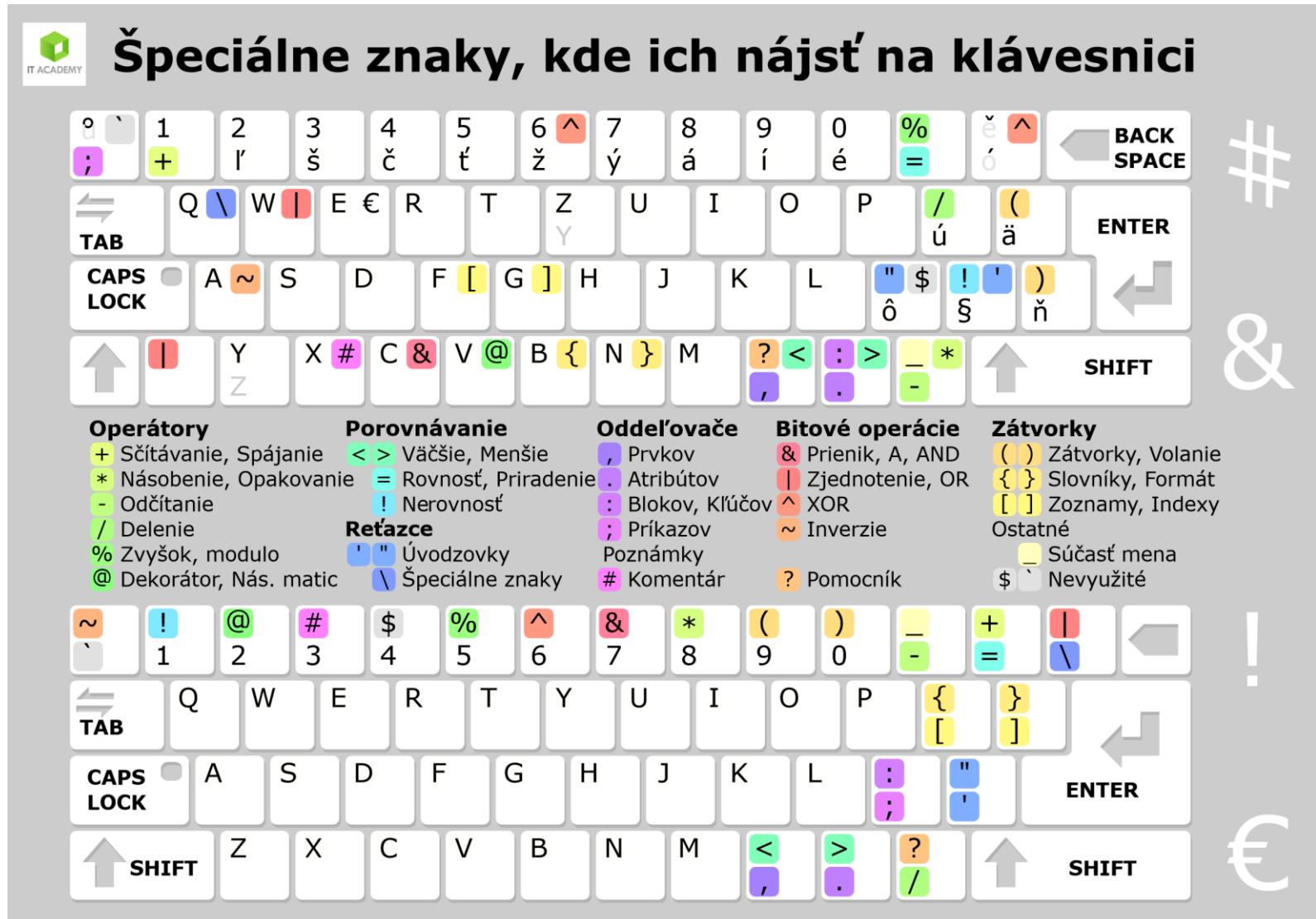
Práca s IDE

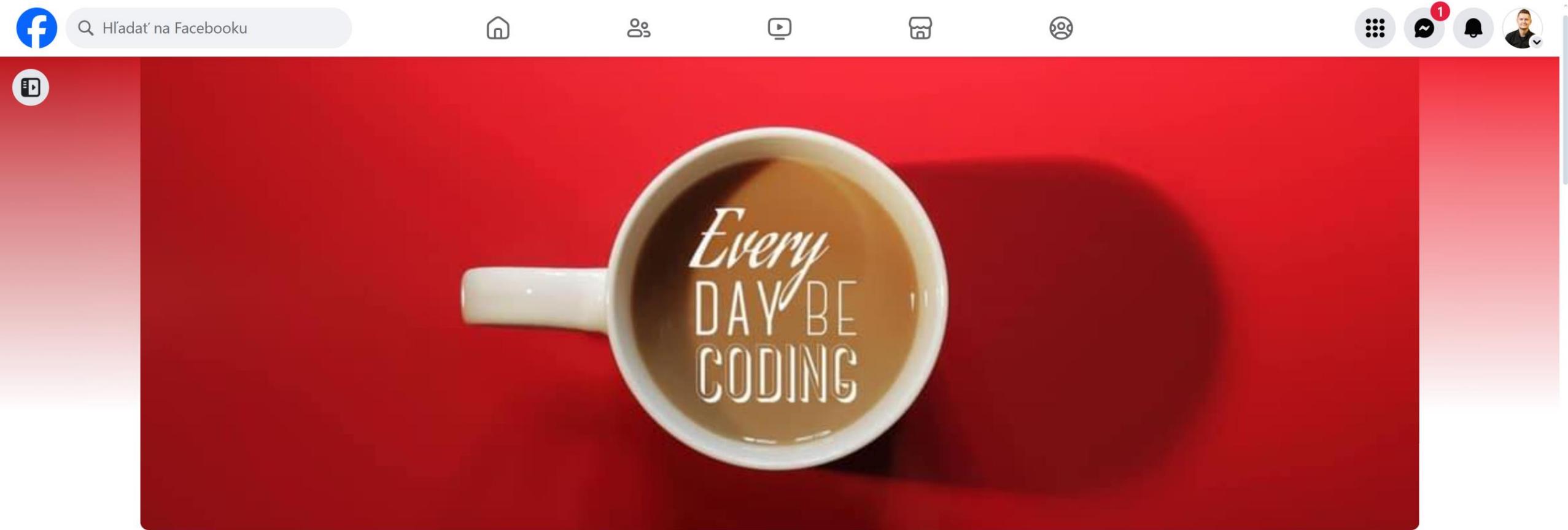
- Ctrl + D Delete zmaž riadok
- **Ctrl + Space** Asistent kódu
- **Ctrl + /** Komentáre
- Ctrl + A Označ všetko
- **Alt + /** Dokonči slovo
- Ctrl + F Hľadanie a náhrady
- Ctrl + Shift + F Kompakt režim
- Ctrl + Shift + S Ulož všetko

Práca s browserom

- Ctrl + T Vytvor nový tab
- Ctrl + W Zatvor aktuálny tab
- Ctrl + Shift + W Zatvor všetky taby
- **Ctrl + Shift + T** Otvor posledný tab
- Ctrl + Shift + J/F12 Web console
- **F11** Fullscreen

Efektívne Používanie Klávesnice





Vývojári

Verejná skupina · 7,4 tis. členov

+ Pozvať

↗ Zdieľať

👤 Člen ▾

▼

Diskusia

Ludia

Podujatia

Médiá

Súbory



...



Napíšte niečo...



Reel



Fotka/video



Anketa

Informácie

Skupina softvérových vývojárov. Táto skupina by mala byť miestom, kde sa môžu českoslovení vývojári vzájomne spoznať, vyžiadať si konštrukívnu... [Zobrazit viac](#)

👤 Verejná

Členov skupiny a ich príspevky bude vidieť ktokoľvek.





Hľadať na Facebooku



```
52 </div>
53 </body>
54 <script type="text/javascript">
55 <!--
56 var currentImage = "bigImage1";
57 var pages = Math.ceil.photos.length / 9);
58 updatePages();
59 updateAllImages();
60 // document.getElementById('bigImage0').src = 'images/wieksza' + photos[page] + '1';
61 // document.getElementById('bigImage0').style.display = '';
62 changePhotoDescription('1');
63
64
65
66
67
68
69
70
71
72
73
```

Programátori

Verejná skupina · 11,1 tis. členov

+ Pozvat

Člen ▾



Diskusia

Vybrané

Ludia

Podujatia

Média

Súbory



Napíšte niečo...



Reel



Fotka/video



Anketa

Informácie

Táto skupina slúži na dohadzovanie si kšeftíkov a pre hľadačov programátorov / vývojárov.

Verejná

Členov skupiny a ich príspevky bude vidieť ktokoľvek.

Viditeľná

Príspevky sú viditeľní iba členom skupiny.



I am programmer



I have Life



I have
stackoverflow



IT ACADEMY

Home

PUBLIC

Questions

Tags

Users

Companies

COLLECTIVES

Explore Collectives

LABS

Discussions

TEAMS

Create free Team

Looking for your Teams?

Tags

A tag is a keyword or label that categorizes your question with other, similar questions. Using the right tags makes it easier for others to find and answer your question.

[Show all tag synonyms](#)

sql

Popular

Name

New

sql

665934 questions

17 asked today, 398 this week

mysql

661556 questions

21 asked today, 273 this week

sql-server

332327 questions

8 asked today, 208 this week

postgresql

172541 questions

18 asked today, 267 this week

sqlite

94129 questions

6 asked today, 59 this week

t-sql

72751 questions

16 asked this week, 91 this month

sql-server-2008

53826 questions

124 asked this year

plsql

28607 questions

19 asked this week, 68 this month

apache-spark-sql

26556 questions

16 asked this week, 84 this month

mysqli

24951 questions

7 asked this week, 26 this month

sqlalchemy

23049 questions

5 asked today, 38 this week

sql-server-2005

18388 questions

18 asked this year



SQL #4956

před 12 měsíci

to tak máte knihu jako vstupní parametr, ale je to úplně k ničemu

```
ALTER PROCEDURE [dbo].[usp_GetDocuments] (
    @DocumentNumber varchar(30) = NULL,
    @ContactID nvarchar(30) = NULL,
    @Book nvarchar(30) = NULL,
    ...
BEGIN
    ...
    IF @Book IS NOT NULL
    BEGIN
        SELECT ...
        WHERE d.Kniha = '31.Kniha'
    END
END
```

0 COMMENTS PŘIDAT KOMENTÁŘ

SQL #4869

před 2 lety

Ako Skončíme?

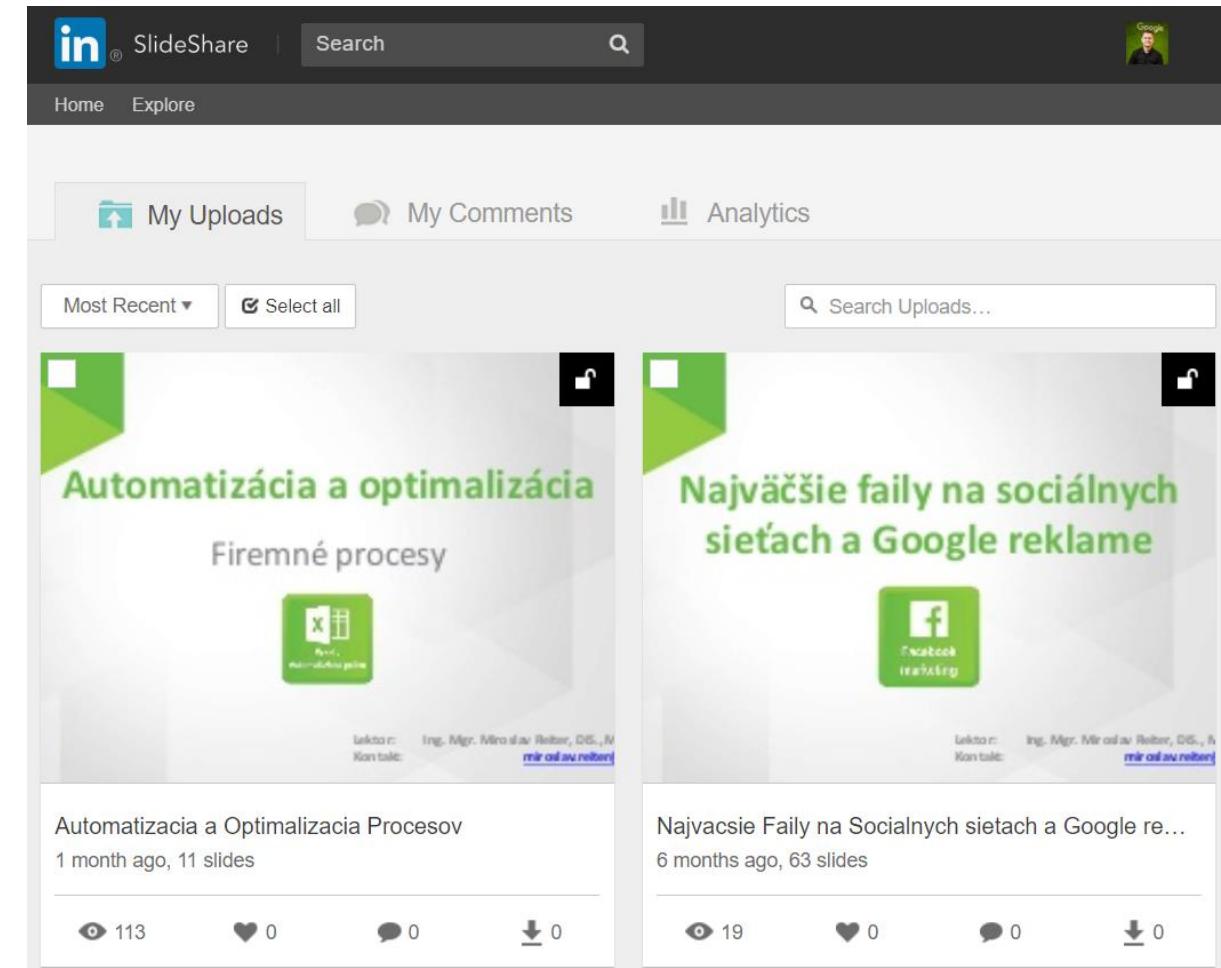
1. Stiahnite si Súbory

- https://github.com/miroslav-reiter/Kurzy_SAV_Analytika_Python_R

2. Pridajte si ma na LinkedIn

- www.linkedin.com/in/miroslav-reiter/

3. Prezentácia a materiály po prednáške



Vyber si online kurz

Nauč sa programovať, tvoriť webstránky a grafiku, manažovať alebo sa zameraj na osobný rozvoj. Všetko jednoducho vďaka našim online kurzom z pohodlia tvojho domova.

Ročné
predplatné na
všetky online
kurzy

~~2299.99€~~

399.99€

Priístup pre Teba do všetkých aktuálnych aj pripravovaných online kurzov

12 mesačná platnosť

🛒 Kúpiť teraz

Zadarmo

1. Kurzy SAV

2. YouTube kanál IT Academy

<https://www.youtube.com/c/IT-AcademySK>

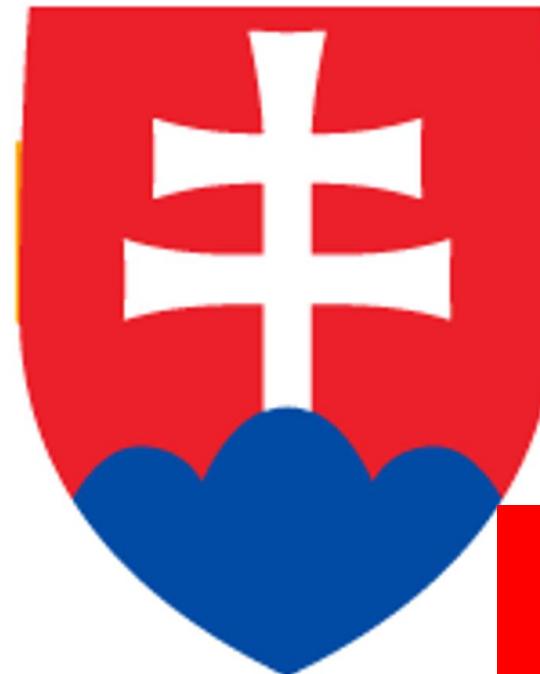
Platené

Moje kurzy na www.vita.sk

Online Kurzy Zdarma



Úrady práce,
sociálnych
vecí a rodiny



info@vita.sk

Ako sa s nami Spojit'?



ADRESA: IT Academy, s. r. o.

Budova KOLOSEO prízemie
Tomášikova 50/A
831 04 Bratislava



WEB: www.it-academy.sk



E-MAIL: info@it-academy.sk



TELEFÓN: +421 917 095 406



Ako Vieme Pomôcť?

#Copywriting

#Školenia

#Zamestnanci

#Pomáhame

#Rast

#Projekty

#Certifikácie

#Kurzy

#Tréningy

#Vzdelávanie

#PPC Kampane

#Elearning

#Mentoring

#Konzultácie

#Online

#Programovanie

#Vývoj

#Marketing

#Reklama

#Prenájom Techniky

- Domov
- Shorts
- Odbory
- Moje
- História

Ak chcete k videám pridať označenie páči sa mi, komentovať alebo sa prihlásiť na odber, musíte sa prihlásiť.

Prihlásiť sa

Preskúmať

- Trendy
- Hudba
- Filmy
- Hry
- Šport

Prehliadat kanály

Viac zo služby YouTube

- YouTube Premium
- YouTube Music
- YouTube Kids

Naštartuj sa s nami v IT!

IT ACADEMY PREZENČNÉ KURZY www.it-academy.sk

VITA ONLINE KURZY www.vita.sk

Daj Odber

IT Academy



Odoberať

Domov Videá Shorts Naživo Zoznamy Komunita



Videá ► Prehrať všetko



Online kurz Data Science a Jazyk R - Jemný Úvod do...

7 zhliadnutí • pred 1 hodinou



Ako programuje Google v Python? Toto je docstring Dokumentačny retazec Toto je nasa dokumentacia ku zdrojovemu kodu... Autor: Larry Page Verzia: 1.0

8:38

Online kurz Python - Ako programuje Google v Python?

51 zhliadnutí • pred 1 dňom

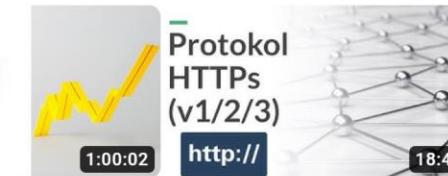


Typy Grafov

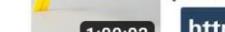


Online Kurz Microsoft Excel Grafy a Typy Grafov

61 zhliadnutí • pred 2 dňami



Protokol HTTPs (v1/2/3)



Online Kurz Počítačové Siete - Protokol HTTP(s), Stavové...

166 zhliadnutí • pred 3 dňami



Názory Odporúčania Hodnotenia



Online Kurz Efektívna Komunikácia - Názory a...

60 zhliadnutí • pred 3 dňami



MS Outlook

1. Duplicítne Emaily
2. Doplnky
3. Tipy a Triky

AKREDITOVANÝ KURZ

Microsoft Visual Basic

30:24

Online kurz Microsoft Outlook - Ako na...

142 zhliadnutí • pred 5 dňami

Oblíbené videá ► Prehrať všetko



SQL I. Začiatčník a. Databáza, Tabuľky, Select, Limit, Where, In, Between, And, Or, Komentár 11-2019

IT ACADEMY

1:51:23



Microsoft Excel - Kontingenčné Tabuľky a. Hárky, Duplicity, Zoradenie, Polia, Funkcie, Agregovanie, Stĺpce, Riadky, Zbalenie 2019

IT ACADEMY

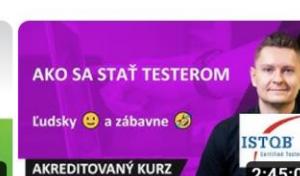
1:51:58



Siete I. Základy Siete a. Konzola, Ping, Ipconfig, Netstat, Sysinternals, TcpView, Traceroute 2018

IT ACADEMY

1:40:33



AKO SA STAŤ TESTEROM Udusky 😊 a zábavne 😊

ISTQB International Tester

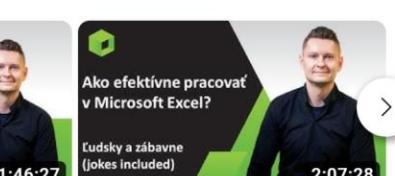
2:45:09



Microsoft Access I. Začiatčník a. Backstage, Databázy, Tabuľky, Porovnanie, Súťaž 03-2019

IT ACADEMY

1:46:27



Ako efektívne pracovať v Microsoft Excel? Udusky a zábavne (jokes included)

2:07:28

Dajte odber na IT Academy



[www.YOUTUBE.COM/C/IT-ACADEMYSK](https://www.youtube.com/c/IT-ACADEMYSK)

