**\*Predispozicia mit instalaci GIT**

**https://git-scm.com/**

## Přihlaš se

username

password

## Vytvoř repositář

* Zaskrtni si vytvoreni README.md

## Nahraj soubor/y na GIT

-vytvoř repozitář a zaklikni Add file

-můžeš protáhnout celou složku s podsložkami kódu

## Stažení (clone) repositáře na lokal

1. Najdi složku, kam chceš uložit SQL kódy (např. C:\SQL\_Kody)
2. Klikni do prázdného místa ve složce pravým tlačítkem myši
3. Vyber „Git Bash Here“

-> Otevře se černé okno s terminálem, kde můžeš hned psát:

git clone <https://github.com/tvuj-tym/nazev-repozitare.git>

napr git clone [https://github.com/EMGpowerbi/SQL-BI\_WMS\_STOCHOV](https://github.com/EMGpowerbi/SQL-BI_WMS_STOCHOV.git)

**Obsah obrázku text, snímek obrazovky, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.**

## Git PULL

* Posílá změny z GitHub na počítač
* + zajisti vždy aktualne zmeny nahrane na lokal

A Manualne

* + 1. **Otevři složku**, kde máš repozitář (např. C:\SQL\_Kody\sql-repozitar)
    2. Pravým tlačítkem klikni do prázdného místa ➜ **„Git Bash Here“**
    3. Do okna napiš:

git pull

* + 1. Zobrazí se hláška jako Already up to date. nebo se stáhnou nové soubory, v terminalu vidíš aj seznam zmen
    2. Teď máš 100% aktuální obsah, který můžeš ihned otevírat v SSMS

B Automaticka ikonka – pre auto pull

* 1. Vytvoření .bat souboru

@echo off

cd /d C:\SQL\_Kody\sql-repozitar

git pull

pause

* 1. Kam to uložit:
     1. Otevři Průzkumník
     2. Jdi do složky, kam si přeješ soubor uložit (např. C:\Users\TvojeJmeno\Desktop)
     3. Klikni pravým tlačítkem → Nový → Textový dokument
     4. Otevři ho a vlož výše uvedený kód
     5. Ulož jako:
     6. aktualizuj-sql.bat

## Git PUSH

* Posílá změny z počítače na GitHub

| **Krok** | **Příkaz** | **Co dělá** |
| --- | --- | --- |
| 1. Vybereš změny | git add . | Řekneš Gitu, které soubory chceš sledovat |
| 2. Potvrdíš změny | git commit -m "..." | Vytvoříš „snímek“ změn s popisem (commit) |
| 3. ODEŠLEŠ změny | git push | Nahraješ změny do GitHub repozitáře |

Pr. Uprav si subor, ulož, spusti bash a zapiš

git add .

git commit -m "Oprava výpočtu zásob"

git push

## Smaž objekt v repo

**1. Mazání souboru přímo na GitHubu (přes web)**

**🪜 Postup:**

1. Otevři **GitHub repozitář**
2. Najdi a klikni na soubor, který chceš smazat (např. procedury/sklad/usp\_PrijemSkladu.sql)

Obsah obrázku text, snímek obrazovky, Písmo, číslo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

**🧽 2. Mazání lokálně v počítači + push na GitHub**

Pokud máš repozitář **naklonovaný v počítači**:

**🪜 Postup:**

1. Otevři složku s repozitářem (např. C:\projekty\sql-kody)
2. Najdi soubor, který chceš smazat a **smaž ho klasicky v Průzkumníku**
3. Otevři **Git Bash** nebo PowerShell v té složce a spusť:

git add .

git commit -m "Smazání souboru xyz.sql"

git push

✅ Git zaznamená změnu a odstraní soubor i na GitHubu

## Proveď změnu v SQL kódu

📌 Předpoklad: máš repozitář stažený v počítači a otevřený v SSMS.

🪜 1. Otevři .sql soubor v SSMS

Najdi si soubor v lokální složce (např. C:\SQL\_Kody\sql-repozitar\s10WMS\usp\_Vypocet.sql)

Otevři ho v SQL Server Management Studiu

📝 2. Uprav kód

Proveď změnu jako obvykle (např. upravíš výpočet, přidáš podmínku)

Soubor ulož (Ctrl+S)

🔄 3. Zapiš změnu do Git (verzování)

Otevři Git Bash ve složce repozitáře a zadej:

git add .

git commit -m "Úprava procedury usp\_Vypocet.sql – změna výpočtu skladu"

git push

✅ Tím jsi změnu zapsal do historie a odeslala zpět na GitHub.

🚀 4. Zapiš na server přes SSMS

Pokud chceš změnu hned aplikovat do databáze, klikni v SSMS na F5 (Execute)

## Notifikace

Vypnout e-maily pro GitHub Actions -zatím

1. Jdi na GitHub.com a přihlas se
2. Klikni na svůj avatar (vpravo nahoře) → Settings
3. Vlevo klikni na Notifications
4. Najdi sekci "GitHub Actions"
5. Uprav volbu:
   * Odškrtni ✅ "Email" nebo
   * Vyber jen "Only on failure", pokud chceš dostávat jen chyby (ne každé spuštění)

## Branche

**🌿 Co je to „branch“ v Gitu?**

**Branch = větev vývoje.**  
Představ si to jako **kopii kódu**, ve které si můžeš něco upravit nebo vyzkoušet, aniž bys ovlivnila originál.

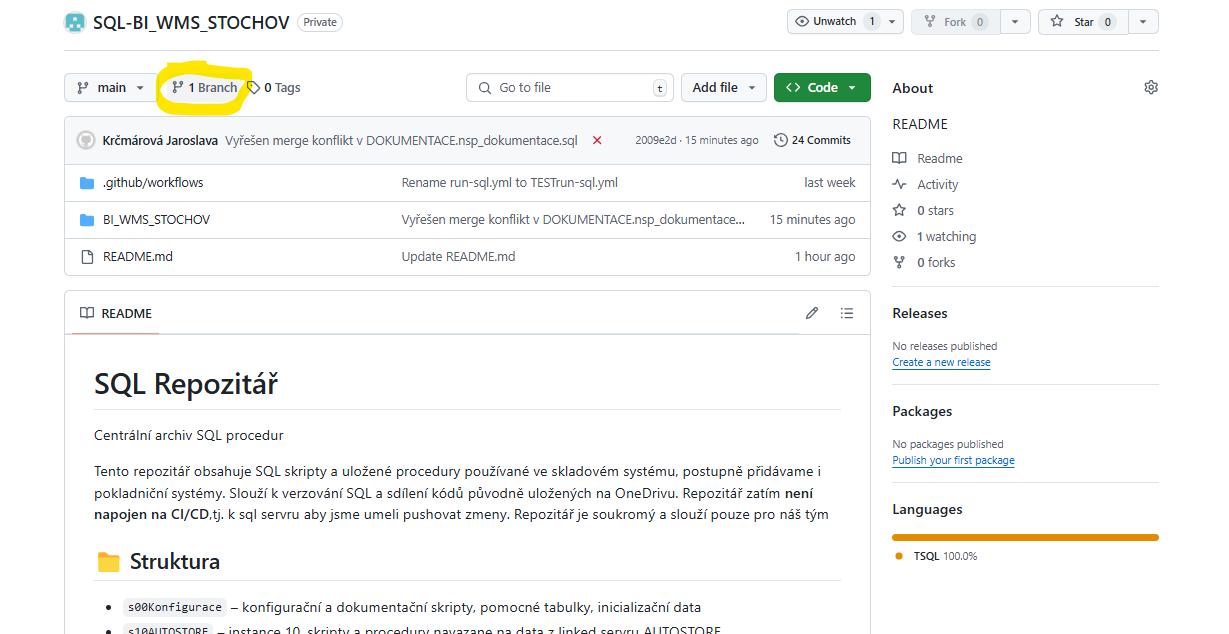
**🧠 Proč větve vůbec používat?**

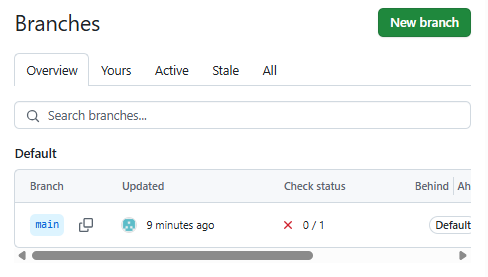
✅ Aby jsme mohli :

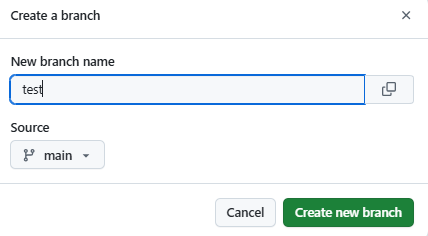
* dělat změny **nezávisle na ostatních**
* něco otestovat, přepsat nebo připravit bez rizika
* a **až máš hotovo**, změnu **sloučíš (merge)** zpět do hlavní větve

**🧵 Typické větve v reálném projektu:**

| **Větev** | **K čemu slouží** |
| --- | --- |
| main nebo master | Hlavní produkční kód (schválený a ověřený) |
| develop | Pracovní verze pro vývoj |
| feature/... | Nové funkce, procedury, změny (např. feature/novy\_vypocet) |
|  |  |







* Dlasi krok -změna v sql
* Commit
* Při obrazeni repo Compare & pull request
* 