

Databáze - cvičení

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

2026/2027

Zadání

► Mějme následující zadání informačního systému:

V informačním systému (IS) budeme evidovat **studenty (Student)**.

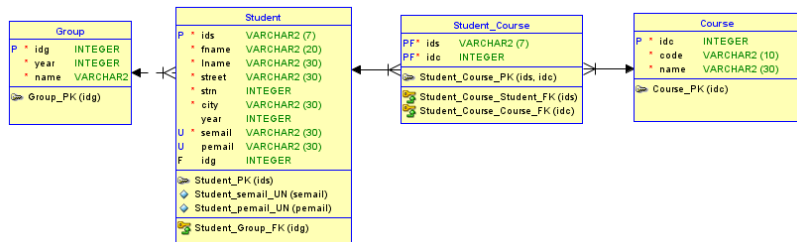
Pro každého studenta budeme evidovat jeho PortalID, jméno, příjmení, ulici, číslo popisné, město, aktuální ročník studia, školní email a osobní email. Všechny atributy kromě aktuálního ročníku studia a osobního emailu jsou povinné.

V IS budeme evidovat **studijní skupiny (Group)** s povinnými atributy ročník a název. Student může být nejvýše v jedné studijní skupině.

V IS budeme evidovat **studijní předměty (Course)** s povinnými atributy idc, kód a název předmětu. Student může studovat více předmětů, předmět studuje více studentů¹.

¹Nebudeme nyní uvažovat opakované zápisy předmětu studenta v různých letech.

Relační datový model (ERD)



Úkoly I

1. Do tabulky Student přidejte atribut datum narození, birthday, který bude povinný. Použijte implicitní hodnotu 1.1.1800 pomocí konstrukce default příkazu jazyka pro definici dat `alter table`.
2. Pomocí příkazu jazyka pro modifikaci dat `update` nastavte vybranému uživateli smysluplné datum narození.
3. Pomocí příkazu jazyka pro modifikaci dat `delete` smažte vybraného uživatele. Proč některé záznamy nelze smazat?
4. Z relačního datového modelu (ERD) nastudujte vhodné pořadí rušení tabulek příkazy jazyka pro definici dat `drop table` a zrušte všechny tabulky databáze studentů.

Úkoly II – Vytvoření databáze

1. Stáhněte skript `cv-4-create.sql` a spuštěním vytvořte tabulky databáze studentů.
2. Stáhněte skript `cv-4-init.sql` a spuštěním inicializujte databázi – vložte záznamy do tabulek databáze studentů.
3. Příkazy jazyka pro dotazování dat `select – from` prozkoumejte obsah tabulek.

Úkoly III - dotazování databáze

Pro následující zadání napište dotazy v SQL:

1. **Dotaz:** Vrať studenty, kteří jsou ve druhém ročníku.
2. **Dotaz:** Vrať studenty, kteří jsou v prvním ročníku.
3. **Dotaz:** Vrať studenty, kteří bydlí v Praze nebo Prostějově.
Napište pomocí `or` i `in`.
4. **Dotaz:** Vrať studenty, kteří nemají nastaven soukromý email.
5. **Dotaz:** Vrať id studentů, kteří mají nastaven soukromý email.
Použijte `is not null` i `not ... is null`.
6. **Dotaz:** Vrať studenty, kteří nebydlí v Praze nebo Prostějově.
Napište pomocí `not in` i logických spojek `or` nebo `and`.

Úkoly IV – dotazování

Pro následující zadání napište dotazy v SQL:

1. **Dotaz:** Vrať studenty, kteří nemají studijní skupinu s id 1 nebo 2.
2. **Dotaz:** Stejný dotaz jako v bodě 1, ale chceme vrátit i studenty, kteří nemají žádnou skupinu.
3. **Dotaz:** Vrať studenty, kteří se narodili v roce 2007. Napište pomocí `or` nebo `and` i `between – and`.
4. **Dotaz:** Vrať studenty, kteří se nenarodili v roce 2007. Napište pomocí `or` nebo `and` i `between – and`.
5. **Dotaz:** Vrať skupiny, které v názvu obsahují řetězec 'R2'.

Úkoly V – dotazování

Pro následující zadání napište dotazy v SQL:

1. **Dotaz:** Vrať studenty (id, jméno a příjmení), jejich studijní skupiny (název) i předměty (kód předmětu) takové, že studenti se narodili v roce 2007 a předměty jsou garantované Katedrou informatiky, tedy `Course.code` začíná řetězcem 'KMI'.
Uvažujte pouze studenty, kteří mají nějakou studijní skupinu.
2. **Dotaz:** Vrať id a název skupin, jejichž studenti studují nějaký předmět z Katedry matematické analýzy, tj. kód předmětu začíná řetězcem 'KMA'.
3. **Dotaz:** Vrať kódy předmětů z Katedry informatiky, které mají zapsané studenti z Prahy (kromě studentů z ostatních měst).
4. **Dotaz:** Vrať kódy předmětů z Katedry informatiky, které nemají zapsané žádní studenti z Prahy. Pokuste se vysvětlit, proč není možné řešit jednoduchou změnou podmínky v selekci na `s.city <> 'Praha'`.