

Databáze - cvičení

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

2026/2027

Úkoly I – zadání a konceptuální model

1. 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.

2. Napište lineární zápis entitního typu Course a vyznačte klíče. Napište vazbu reprezentující zápis studenta na předmět.

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

Úkoly II – konceptuální a relační datový model

1. Do konceptuálního modelu v ERD z minulého cvičení přidejte entitní typ Course, definujte datové typy atributů, povinné atributy a identifikátor.
2. Do ERD přidejte vazbu mezi Student a Course. Nastavte správně název vazby, povinnost členství entitních typů ve vazbě a kardinalitu vazby.
3. V nástroji Oracle Modeler převed'te konceptuální model v ERD na relační datový model v ERD. Dle postupu z přednášky nastavte šablonu názvů cizích klíčů pouze na název odkazovaného atributu, nikoli na název odkazované tabulky a odkazovaného atributu.
4. Popište (vysvětlete) vytvořený relační datový model a soustřeďte se přitom na implementaci vazby mezi entitními typy Student a Course.

Úkoly III – implementace vazby v SQL

1. Ve VS Code vytvořte soubor `db-cv-3.sql`, do kterého zapisujete všechny příkazy pro následující úkoly. Každý příkaz spustíte označením a stiskem Shift + Enter.
2. Napište příkaz jazyka pro definici dat (DDL) **create table** pro vytvoření tabulky `Course`, definujte primární klíč a povinné atributy.
3. Jedním příkazem jazyka pro manipulaci s daty (DML) **insert** vložte do tabulky `Course` několik záznamů.
4. Napište příkaz **create table** pro vytvoření **vazební tabulky** implementující vazbu $M : N$ mezi tabulkami `Student` a `Course`, definujte cizí klíče, primární klíč a povinné atributy.
5. Jedním příkazem **insert** vložte do vazební tabulky několik záznamů tak, aby jeden student nestudoval žádný předmět a ostatní studenti studovali více než dva předměty.

Úkoly IV – dotazování tabulek ve vazbách I

1. Příkazem SQL `select` s konstrukcí `join` vypište studenty a `idc` předmětů, které mají studenti zapsané. Vysvětlete počet záznamů ve výsledku.
2. Do předchozího řešení doplňte operaci `join` tak, abyste vypsalí studenty a jejich předměty (tedy kompletní záznamy jejich předmětů). Vysvětlete počet záznamů ve výsledku.
3. Napište SQL dotaz, který vypíše předměty a jejich studenty. V čem se výsledek liší od výsledku předchozího dotazu?
4. Napište stejný dotaz jako v bodu 2, ale vypište pouze `idp`, jméno a příjmení studentů a názvy předmětů.
5. Napište stejný dotaz jako v bodu 4, ale jméno a příjmení studenta vypište jako nový atribut `student_name`, který bude obsahovat: jméno studenta, ' ' a příjmení studenta.

Úkoly V – dotazování tabulek ve vazbách II

1. Napište SQL dotaz pro zadání: vrať studenty, jejich studijní skupiny, a jejich předměty.
2. Pro studenta, který studuje nejméně jeden předmět a má zadanou studijní skupinu, napište příkaz update, který mu nastaví studijní skupinu na null.
3. Spusťte znovu dotaz z bodu 1 a vysvětlete, proč neobsahuje záznam tohoto studenta, přestože ten má zapsané nějaké předměty.
4. Pokuste se posunout operaci Student ⋈ Group v dotazu z bodu 1 z první na poslední místo v příkazu select. Proč je to možné? Je výsledek stejný?