

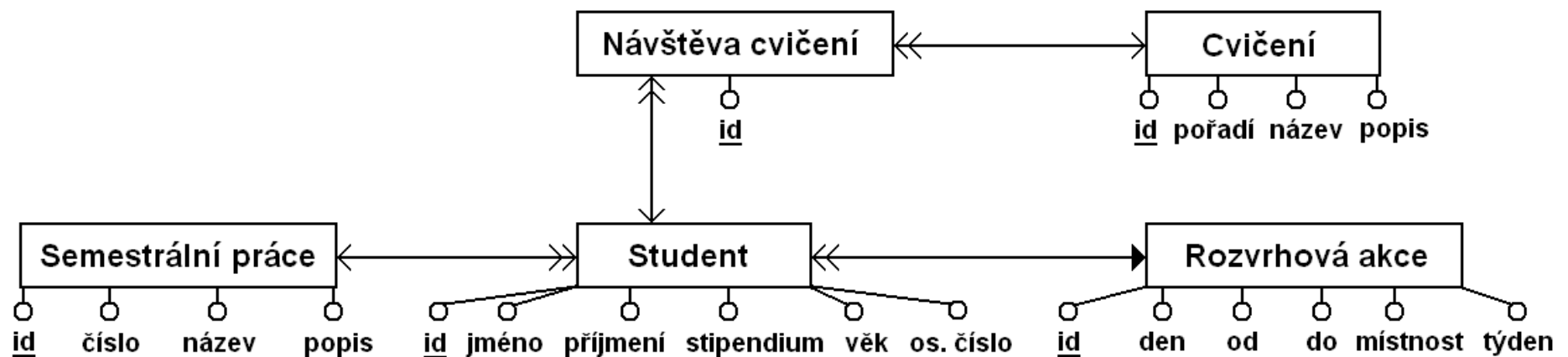
KIV/ZIS cvičení 6

Tomáš Potužák

Pokračování SQL

- Klauzule GROUP BY a dotazy nad více tabulkami
- Stáhnout soubor *studenti_dotazy_sql.mdb*

ERA Diagram po rozkladu vazby M:N:



Slučování záznamů do skupin (1)

- Chceme zjistit informace obsažené ve více záznamech najednou
- Klauzule GROUP BY se seznamem sloupců, podle kterých se mají záznamy spojit
- Nakonec lze přidat podmínku pro již sloučené záznamy – klauzule HAVING s podmínkou

Slučování záznamů do skupin (2)

- Příklad 1 – Vypište výše stipendií vyplácené studentům s počtem studentů, který danou výši stipendia dostává. Studenty bez stipendia neuvažujte.
 - `SELECT stipendium, COUNT(os_cislo) AS [Počet studentů] FROM student WHERE stipendium > 0 GROUP BY stipendium;`

Slučování záznamů do skupin (3)

- Příklad 2 – Zadání jako příklad 1, ale vypište jen stipendia, která berou alespoň 2 studenti
 - `SELECT stipendium, COUNT(os_cislo) AS [Počet studentů] FROM student WHERE stipendium > 0 GROUP BY stipendium HAVING COUNT(os_cislo) >= 2;`
- Příklad 3 – Vypište stáří studentů s počtem studentů u každého věku

Slučování záznamů do skupin (4)

- Příklad 4 – Zadání jako příklad 3, ale vypište jen věk, který mají alespoň 2 studenti
- Příklad 5 – Zadání stejné jako příklad 4, ale místo věku použijte rok narození (sloupec vypočtený jako (2018 - vek)).
Pozor – v klauzuli HAVING a GROUP BY se nedají použít nové názvy sloupců, celý výraz se tam musí zopakovat.

Spojení více tabulek (1)

- Chceme získat informaci, která je ve více tabulkách (Např. seznam studentů s názvy a čísla semestrálních prací)
- Tabulky, ze kterých chceme informaci získat, by spolu měli souviset (měla by mezi nimi existovat vazba → v obou tabulkách jsou související záznamy)
- Spojení se dělá přes rovnost primárního a odpovídajícího cizího klíče

Spojení více tabulek (2)

- Spojení je více typů podle toho, jak se přistupuje k záznamům, které nemají odpovídající záznam v druhé tabulce (např. student bez semestrální práce)
 - *Přirozené spojení (natural join)* – do výsledku dotazu zahrnuty jen záznamy vyskytující se v obou tabulkách (tj. cizí klíč je nastaven). Záznamy (řádky) z pravé i levé tabulky, které nemají odpovídající záznam v druhé tabulce nejsou do dotazu zahrnuty

Spojení více tabulek (3)

- *Levé spojení (left join)* – Do výsledku dotazu jsou zahrnuty i řádky z levé tabulky, které nemají odpovídající záznam v pravé tabulce
- *Pravé spojení (right join)* – Do výsledku dotazu jsou zahrnuty i řádky z pravé tabulky, které nemají odpovídající záznam v levé tabulce

Přirozené spojení (1)

- Příklad 6 – Vypište studenty (jméno, příjmení, os. číslo) společně s rozvrhovou akcí, kterou navštěvují (den, týden, od, do)
 - SELECT jmeno, prijmeni, os_cislo, den, tyden, od, do FROM student, rozvrhova_akce WHERE rozvrhova_akce_id = rozvrhova_akce.id;
- Příklad 7 – Jako příklad 6, ale vypište jen studenty starší 21 let chodící na pondělní rozvrhové akce

Přirozené spojení (2)

- Místo plně kvalifikovaných názvů (viz předchozí dva příklady) lze použít *aliasy* → zkrácení zápisu
- Nezaměňovat s novým pojmenováním sloupce (klauzule AS)!
- Příklad 8 – Jako příklad 6 s aliasy
 - SELECT jmeno, prijmeni, os_cislo, den, tyden, od, do FROM student s, rozvrhova_akce r WHERE s.rozvrhova_akce_id = r.id;

Přirozené spojení (3)

- Příklad 9 – Vypište studenty (jméno, příjmení, os_číslo) společně s jejich semestrální prací (číslo, název)
- Příklad 10 – Vypište seznam semestrálních prací (číslo, název) s počtem studentů, kteří si tyto práce vybrali
- Příklad 11 – Vypište seznam studentů společně s počtem cvičení, které navštívili

Přirozené spojení (4)

- Příklad 12 – Vypište studenty (jméno, příjmení, os. číslo) společně s cvičeními, které navštívili (pořadí, název)
- Příklad 13 – Vytvořte dotaz, který vypíše všechny studenty, kteří byli alespoň na dvou cvičení s počtem návštěv cvičení
- Příklad 14 – Vypište studenty starší 20ti let s počtem návštěv cvičení, kteří byli alespoň na dvou cvičeních

Přirozené spojení (5)

- Jak se vyhodnocuje dotaz z příkladu 14

Tabulka student

| id | jmeno | prijmeni | os_cislo | vek | ... |
|----|-------|----------|----------|-----|-----|
| 1 | Tomáš | Marný | A01 | 18 | ... |
| 2 | Jana | Malá | A02 | 25 | ... |
| 3 | Lenka | Volná | A03 | 21 | ... |
| 4 | Petr | Nový | A04 | 22 | ... |

Tabulka navsteva_cviceni

| id | student_id | cviceni_id |
|----|------------|------------|
| 1 | 1 | 1 |
| 2 | 2 | 1 |
| 3 | 2 | 2 |
| 4 | 2 | 5 |
| 5 | 3 | 7 |
| 6 | 4 | 1 |
| 7 | 4 | 3 |

Přirozené spojení (6)

- Pro přirozené spojení existuje i alternativní zápis, který používá MS Access
- Příklad 15 – Vypište studenty (jméno, příjmení, věk) starší 20ti let společně s jejich semestrální prací (číslo, název)
 - `SELECT jmeno, prijmeni, vek, cislo, nazev
FROM student s INNER JOIN
semestralni_prace sp ON
s.semestralni_prace_id = sp.id WHERE vek >
20;`

Levé (pravé) spojení (1)

- Příklad 16 – Vypište úplný seznam semestrálních prací s počtem studentů, kteří si je vybrali
 - SELECT cislo, nazev,
COUNT(semestralni_prace_id) AS
Pocet_studentu FROM semestralni_prace sp
LEFT JOIN student s ON
s.semestralni_prace_id = sp.id GROUP BY
cislo, nazev;

Levé (pravé) spojení (2)

- Příklad 17 – Vypište seznam všech studentů s počtem návštěv cvičení, kteří byli maximálně na jednom cvičení. Použijte pravé spojení.
- Aby ve sloupci cizího klíče v jedné tabulce mohlo být NULL:
 - U cizího klíče musí být „Je nutno vyplnit – NE“
 - U vazby mezi tabulkami nesmí být zaškrtnuto „Zajistit referenční integritu“

DML – další dotazy

- Kromě příkazu SELECT jsou součástí DML příkazy pro manipulaci se záznamy v tabulce
 - Přidání záznamu – INSERT
 - Úprava záznamu – UPDATE
 - Smazání záznamu – DELETE
- V MS Access většinou nejsou potřeba, úpravy záznamů se provádí interaktivně v tabulkách/formulářích

Příkaz INSERT

- INSERT INTO *tabulka* VALUES (*hodn1*, *hodn2*, ...);
- INSERT INTO *tabulka* (*sloupec1*, *sloupec2*, ...) VALUES (*hodn1*, *hodn2*, ...)
- Příklad 18 – Vložte novou semestrální práci do tabulky semestrální práce
 - INSERT INTO *semestralni_prace* (*cislo*, *nazev*, *popis*) VALUES (9, 'Zaměstnanci', 'Databáze zaměstnanců firmy');

Příkaz UPDATE

- UPDATE *tabulka* SET sloupec1 = hodn1, sloupec2 = hodn2,... WHERE *podmínka*;
- Příklad 19 – Změňte název a popis semestrální práce číslo 9 na „Hotel“ a „Databáze hostů hotelu“
 - UPDATE semestralni_prace SET nazev = 'Hotel', popis = 'Databáze hostů hotelu' WHERE cislo = 9;

Příkaz DELETE

- DELETE FROM *tabulka* WHERE *podmínka*;
- Příklad 20 – Smažte semestrální práci číslo 9 z tabulky semestrální práce
 - DELETE FROM semestralni_prace WHERE cislo = 9;

Nastavení přístupových práv

- U databáze lze nastavit přístupové heslo
 - Musí být otevřena ve *výhradním režimu*
- Lze vytvářet nové uživatele, skupiny uživatelů
- Uživatele lze přidávat/odebírat ze skupin
- Uživatelů či skupinám lze přiřadit práva k jednotlivým objektům – tabulkám, dotazům, atd.