

KIV/ZIS cvičení 5

Tomáš Potužák

Úvod do SQL (1)

- *SQL (Structured Query Language)* je standardizovaný strukturovaný dotazovací jazyk pro práci s databází
- Veškeré operace v databázi se dají provádět pomocí příkazů SQL
- V MS Access je možné psát dotazy přímo v SQL místo návrhového zobrazení či průvodce

Úvod do SQL (2)

- SQL se skládá z částí:
 - *DDL (Data Definition Language)* – Příkazy pro vytváření struktury databáze (vytvoření a úprava tabulek)
 - *DML (Data Manipulation Language)* – Příkazy pro získání (zobrazení) dat z databáze a jejich úpravy (přidání, změna, odebrání záznamu)

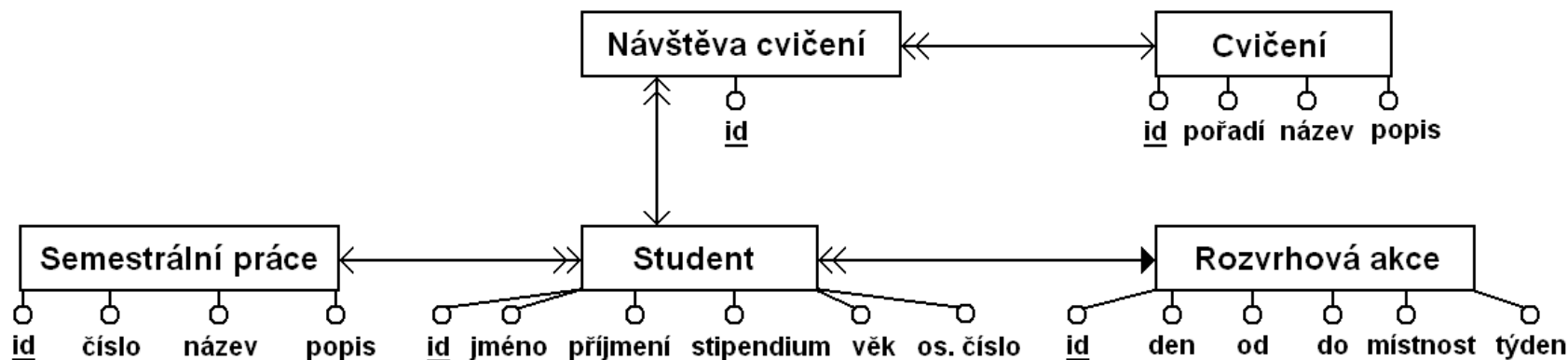
Úvod do SQL (3)

- Příkazy DDL se v MS Access příliš nepoužívají, pro tvorbu tabulek lze bez problémů použít návrhové zobrazení
- Místo DML příkazů lze využít formuláře nebo datové listy tabulek
- Některé složitější dotazy je nutno zapsat v SQL, někdo SQL upřednostňuje

DB pro procvičení SQL

- Budeme pracovat s DB ze cvičení 3 a 4
- Stáhnout soubor *db5.accdb*

ERA Diagram po rozkladu vazby M:N:



SQL v MS Access

- SQL konzole, do které lze zapisovat SQL dotazy se dá zobrazit v návrhovém zobrazení dotazů
- Pro každý dotaz vytvořený v návrhovém zobrazení lze zobrazit jeho SQL verzi
- V návrhovém zobrazení kliknout na lištu návrhového zobrazení pravým tlačítkem a z menu vybrat položku *Zobrazení SQL*

Příkaz SELECT

- Nejběžnější dotaz, získává specifikované informace z jedné či více tabulek databáze
- Syntaxe příkazu:

```
SELECT [DISTINCT|ALL] {*|[sloupec_vyraz [AS nove_jmeno] [...]}  
    FROM tabulka [alias] [...]  
[WHERE podminka]  
[GROUP BY seznam_sloupcu] [HAVING podminka]  
[ORDER BY seznam_sloupcu]
```

Výpis všech údajů v tabulce

- Příklad 1 – Napište SQL dotaz, který vypíše všechny studenty.
 - `SELECT * FROM student;`
- Příklad 2 – Napište SQL dotaz, který vypíše všechny semestrální práce
- Příklad 3 – Napište SQL dotaz, který vypíše všechny cvičení

Výběr sloupců z tabulky

- Příklad 4 – Napište dotaz, který vypíše seznam studentů (jméno, příjmení, os. číslo)
 - `SELECT jmeno, prijmeni, os_cislo FROM student;`
- Příklad 5 – Napište dotaz, který vypíše čísla a názvy semestrálních prací
- Příklad 6 – Napište dotaz, který vypíše pořadí a názvy cvičení

Odstranění nadbytečných řádků

- Pro odstranění duplicitních řádků použijeme klauzuli DISTINCT
- Příklad 7 – Napište dotaz, který vypíše seznam dnů, ve kterých se učí KIV/ZIS
 - SELECT DISTINCT den FROM rozvrhova_akce;
- Příklad 8 – Napište dotaz, který vypíše seznam počátečních časů cvičení KIV/ZIS

Výběr s odvozeným sloupcem

- Nemusíme vybírat pouze sloupce tabulky, lze vytvořit nové výpočtem z původních
- Příklad 9 – Vypište studenty společně s rokem jejich narození
 - `SELECT jmeno, prijmeni, os_cislo, 2010 - vek AS rok_narozeni FROM student;`
- Příklad 10 – Vypište studenty s jejich stipendii po zdanění daní 10 %

Výběr s podmínkou (1)

- Použije se klauzule WHERE s podmínkou
- Do výsledku dotazu se zahrnou pouze záznamy splňující podmínku
- Příklad 11 – Vypište všechny studenty starší 21 let
 - `SELECT jmeno, prijmeni, os_cislo FROM student WHERE vek >= 21;`

Výběr s podmínkou (2)

- V podmínice můžou být:
 - Standardní operátory – >, >=, =, <=, <, <>, AND, OR, NOT
 - Interval – BETWEEN cislo1 AND cislo2
 - Členství ve výčtu – IN ('Jana', 'Petr')
 - Hledání podobných řetězců – LIKE 'Ja*' (% , řetězec začínající „Ja“ libovolné délky), LIKE 'Jan?' (_, řetězec začínající „Jan“ délky 4)
 - Testování nulové hodnoty – IS NULL

Výběr s podmínkou (3)

- Příklad 12 – Vypište všechny studenty mezi 21 a 23 lety
- Příklad 13 – Vypište všechny studenty starší 22 let, kteří neberou stipendium
- Příklad 14 – Vypište všechny studenty, jejichž jméno začíná na „A“
- Příklad 15 – Vypište všechny studenty kteří se jmenují Tomáš, Jana nebo Petr

Řazení (1)

- K řazení slouží klauzule ORDER BY se seznamem sloupců, podle kterých se řadí
- Na konec lze přidat klauzule ASC (vzestupné řazení – implicitní) nebo DESC (sestupné řazení)
- Příklad 16 – Vypište všechny studenty starší 20ti let seřazené podle věku
 - `SELECT * FROM student WHERE vek >= 20 ORDER BY vek;`

Řazení (2)

- Příklad 17 – Vypište studenty seřazené podle příjmení, jména a os. čísla
- Příklad 18 – Vypište jméno, příjmení, os. číslo a stipendium studentů seřazených podle velikosti stipendia sestupně
- Příklad 19 – Vypište jméno, příjmení a os. číslo a věk studentů seřazených podle věku sestupně

Agregační funkce (1)

- COUNT() – vrací počet nenulových (Pouze různých od NULL!) hodnot (záznamů) ve specifikovaném sloupci
- SUM() – vrací součet hodnot ve specifikovaném sloupci
- MIN() – vrací minimum
- MAX() – vrací maximum
- AVG() – vrací aritmetický průměr

Agregační funkce (2)

- Příklad 20 – Vypište počet studentů
 - `SELECT COUNT(*) FROM student;`
- Příklad 21 – Vypište průměrné, maximální a minimální stipendium ze studentů, kteří nějaké pobírají.
 - `SELECT AVG(stipendium) AS Prumerne, MIN(stipendium) AS Minimalni, MAX(stipendium) AS Maximalni FROM student WHERE stipendium > 0;`

Agregační funkce (3)

- Příklad 22 – Vypište čas prvního cvičení předmětu ZIS v pondělí v sudý týden
- Příklad 23 – Vypište nejnižší, nejvyšší a průměrný věk studentů
- Příklad 24 – Vypište počet témat semestrálních prací
- Příklad 25 – Vypište sumu, kolik musí škola zaplatit studentům na stipendiích

Vnořený příkaz SELECT (1)

- V některých případech potřebujeme zjistit netriviální informaci, k dokončení jednoduchého příkazu SELECT
- Použijeme vnořený příkaz SELECT, který tuto informaci zjistí
- Vnořený SELECT se celý píše do vnějšího (hlavního) SELECT a uzavírá se pro větší přehlednost do závorek

Vnořený příkaz SELECT (2)

- Příklad 26 – Vypište jméno a příjmení nejmladšího studenta
 - SELECT jmeno, prijmeni FROM student WHERE vek = (SELECT min(vek) FROM student)
 - Nelze napsat jen „WHERE vek = min(vek)“
 - SELECT TOP 1 jmeno, prijmeni FROM student ORDER BY vek
 - Alternativa bez vnořeného dotazu – funguje v MS Access, jinde se používá LIMIT

Vnořený příkaz SELECT (3)

- Příklad 27 – Vypište jméno a příjmení studenta s nejvyšším stipendiem