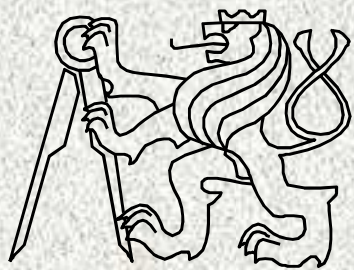


Databáze a JDBC API

úvod



A0B36PR2-Programování 2
Fakulta elektrotechnická
České vysoké učení technické

Databáze

Japonský zákon o autorsko-právní ochraně databází (1986):

- Databáze je soubor informací, jako jsou znaky, čísla, diagramy, jejichž **systematická struktura** umožňuje, aby tyto informace mohly být vyhledávány pomocí **počítače**.

Hromadné zpracování dat

- “klasické” zpracování založené na systému ovládání souborů OS (COBOL, PL1, Pascal)
- Autoři aplikací opakovaně řešili problémy:
 - s **persistencí (stálostí) dat**
 - s **redundancí a nekonzistencí dat**
 - s **obtížností přístupu k datům z ad hoc pohledu**
 - s **izolovaností dat různého původu**
 - s **ochranou dat před poškozením a zneužitím**
 - s **integritou datové základny s modelovanou realitou**
 - s **vyjádřením logiky dat, vztahů mezi záznamy**
 - s **koordinací přístupu více uživatelů k datům**

Příklad HZD – dílčí aplikační programy

- pro údržbu číselníku studijních oborů
- pro definici nového studijního předmětu
- pro definici nového studijního plánu
- pro zaregistrování nového studenta
- pro zápis daného předmětu danému studentovi referentem studijního oddělení na daný semestr
- program pro zaevidování prospěchu studenta z předmětu v daném semestru referentem.

Později budeme chtít přidat novou funkčnost – dělení studentů do oborů

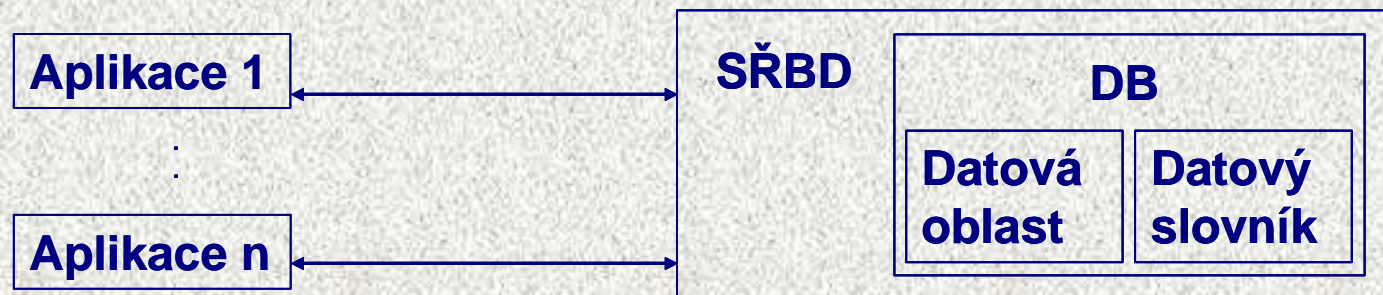
- nový permanentní soubor s informacemi typu “daný student má zájem o daný studijní obor“
- nové aplikační programy pro zaznamenání zájmu o daný obor daným studentem, pro vyhodnocení zájmu a pro rozdělení studentů do oborů.

Úvod

Databáze

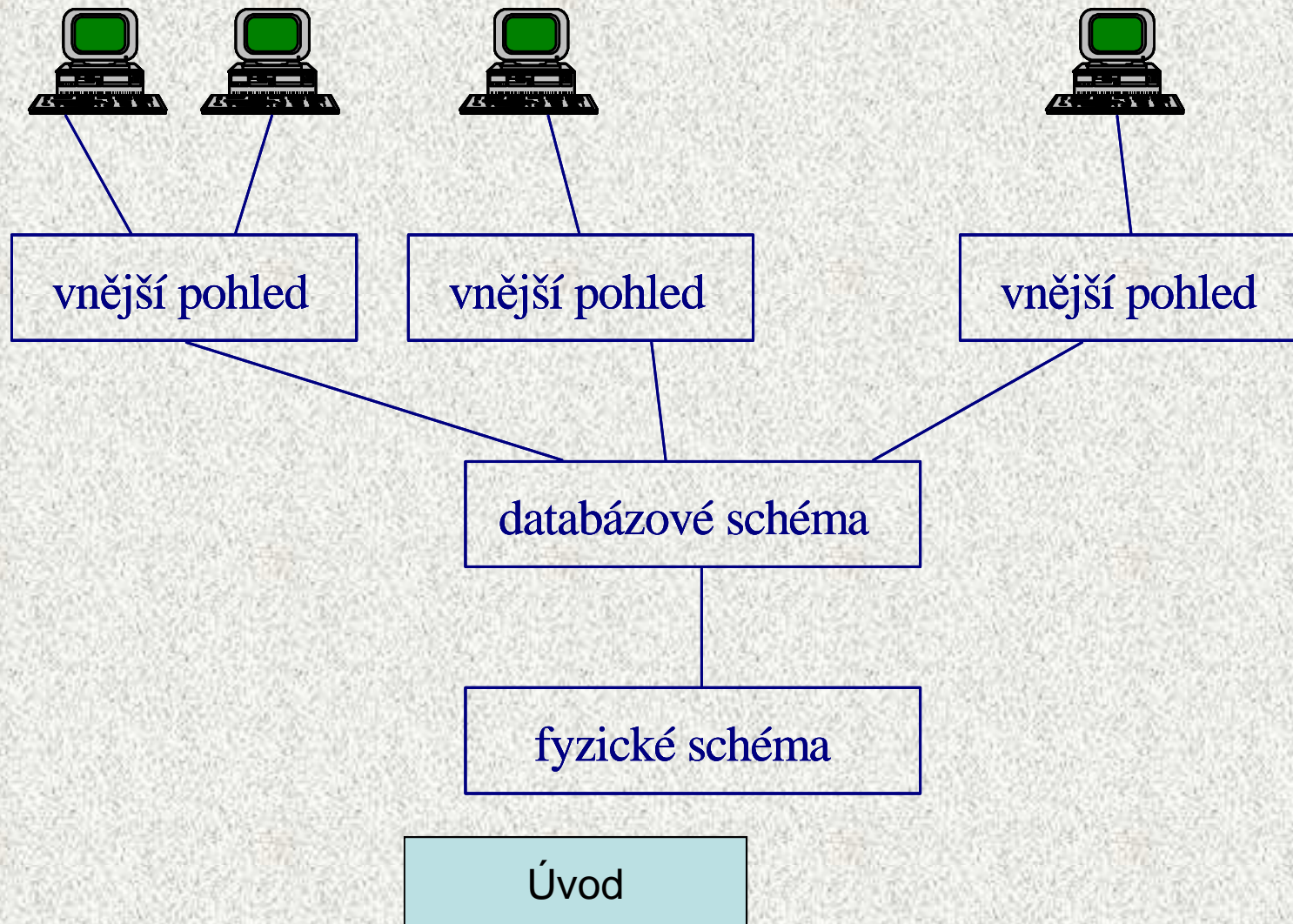
Databázová technologie se zabývá řízením velkého množství perzistentních, spolehlivých a sdílených dat

- Pro data nestačí vnitřní paměť
- Data přetrvávají od zpracování ke zpracování
- Data lze rekonstruovat po chybě
- Data jsou přístupná více uživatelům a to i současně



- DBS = SŘBD + DB
- Databáze zaznamenává čtyři typy informace:
schéma databáze, datové prvky, vztahy mezi nimi, integritní omezení
- Základní paradigma: existence dat v DB je nezávislá na aplikačních programech – programy nemají deklarace vnějších datových struktur

Pohled na data z různé úrovně abstrakce



Datové modely

Datový model je množina konceptů pro popis dat, popis vztahů mezi nimi, popis jejich sémantiky a popis omezení na kvalitu uložených dat.

- Každý model používá koncepty odpovídající úrovni abstrakce
 - Fyzický model: úroveň datových bloků
 - Souborový model: položka, záznam, soubor
 - Databázový model: atribut, tabulka, reference
 - Konceptuální model: model využívající koncepty blízké lidskému myšlení
 - objekty a vztahy mezi objekty

Konceptuální modely

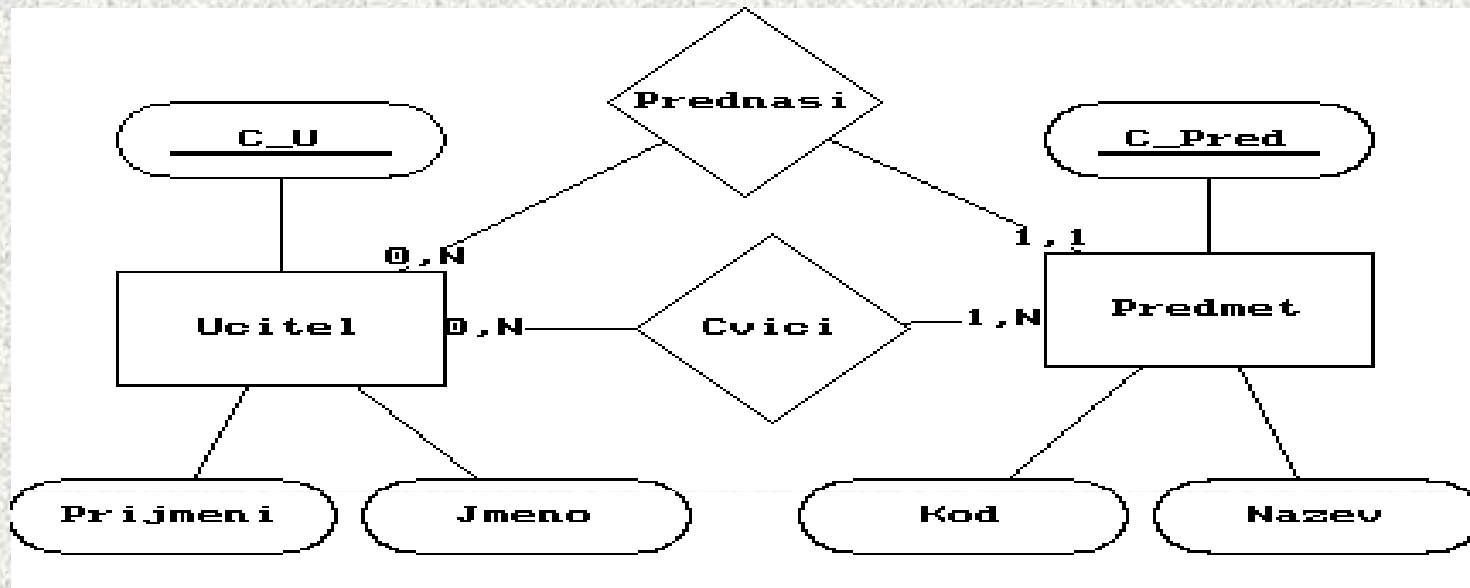
- společný „jazyk“ pro analytiky, projektanty, programátory a uživatele
- integrace uživatelských pohledů
- vstup pro návrh implementace
- použité konstrukty jsou nezávislé na pozdější implementaci

E-R Konceptuální model

- Projektant na základě podrobného seznámení s modelovanou realitou:
 - identifikuje typy entit
 - identifikuje typy vztahů, do kterých entity identifikovaných typů mohou vstupovat
 - přiřadí jednotlivým typům entit a vztahů atributy popisující blíže vlastnosti
 - formuluje různá integritní omezení, vyjadřující s větší či menší přesností soulad schématu s modelovanou realitou

Úvod

E-R diagram na úrovni typů



UČITEL má identifikátor (klíčový atribut) Č_U a neklíčové atributy PŘÍJMENÍ a JMÉNO. Podobně PŘEDMĚT.

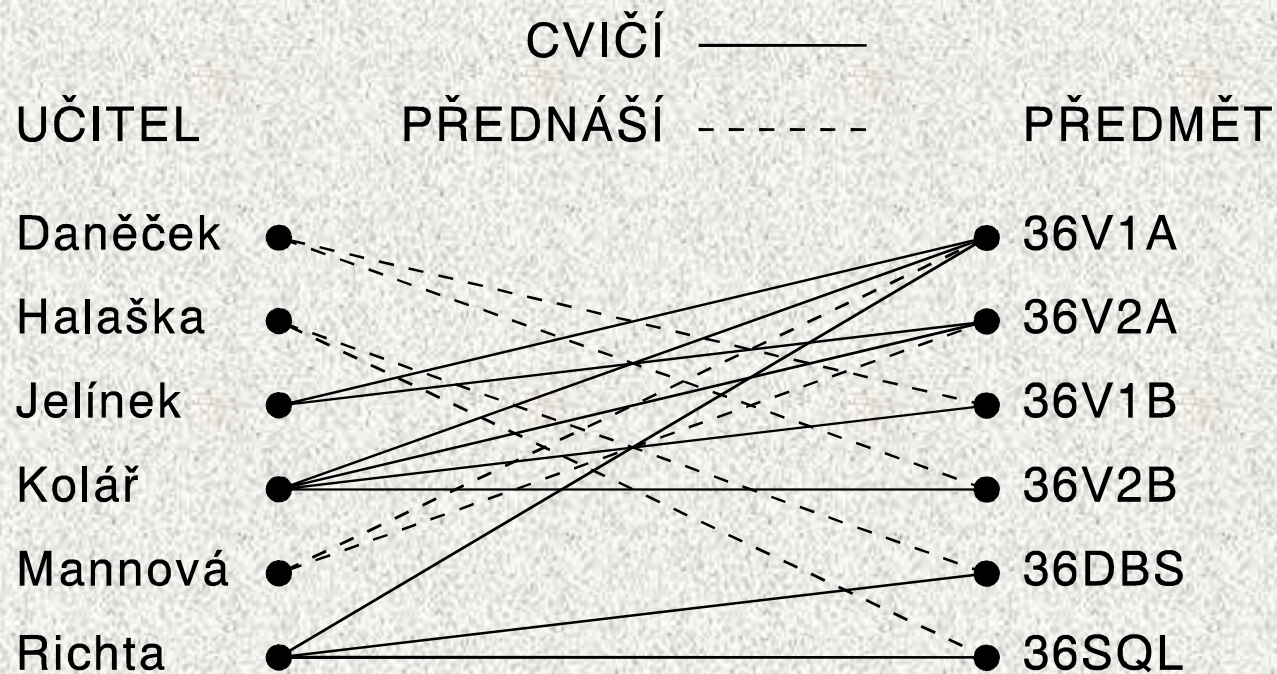
Daný UČITEL vstupuje do vztahu CVIČÍ minimálně nulkrát, maximálně n-krát (může být cvičícím mnoha předmětů, ale nemusí být ani jednoho).

Daný PŘEDMĚT vstupuje do vztahu CVIČÍ minimálně jednou, maximálně n-krát (musí mít alespoň jednoho cvičícího, může jich mít mnoho)

Daný PŘEDMĚT vstupuje do vztahu PŘEDNÁŠÍ právě jednou (musí mít právě jednoho přednášejícího)

Úvod

E-R diagram na úrovni výskytů



Úvod

Relační model

Schéma relace R lze zapsat jako

$R(A_1: D_1, \dots, A_n: D_n)$

R ... jméno relace

A_i ... jméno atributu

D_i ... doména - množina přípustných hodnot pro atribut A_i

Příklad:

STUDENT(C_ST: CHAR(8),
PŘÍJMENÍ: CHAR(15),
ROČNÍK: {1..6},
SKUPINA: NUMBER(2))

Komerční terminologie:

relace tabulka
atribut sloupec
n-tice řádek

Relační instance - komerčně se nazývá tabulka

Mějme relaci $R(A:\{1,2\}, B:\{1,2,3\}) \dots D_A=\{1, 2\}, D_B =\{1, 2, 3\}$

Možné relace dle tohoto schématu:

$R1^* = ((1,1), (1,2), (1,3), (2,1), (1,2), (2,3), (3,1), (3,2), (3,3))$

$R2^* = ((1,1), (1,2), (2,1), (3,3))$

$R3^* = ((3,1), (1,2), (2,3))$

$R1^* = D_A \times D_B \quad Ri^* \subseteq D_A \times D_B$

| R3* | |
|-----|---|
| A | B |
| 3 | 1 |
| 1 | 2 |
| 2 | 3 |

| R2* | |
|-----|---|
| A | B |
| 1 | 1 |
| 1 | 2 |
| 2 | 1 |
| 3 | 3 |

| R1* | |
|-----|---|
| A | B |
| 1 | 1 |
| 1 | 2 |
| 1 | 3 |
| 2 | 1 |
| 2 | 2 |
| 2 | 3 |
| 3 | 1 |
| 3 | 2 |
| 3 | 3 |

RELACE ... chceme-li přeložit jako **vztah**, tedy vztah mezi atributy „jedné tabulky“

Příklad relačních dat

STUDENT:

| C_ST | JMÉNO | ROČNÍK | SKUPINA |
|------|-----------|--------|---------|
| ST1 | Novák | 1 | 25 |
| ST2 | Nováková | 1 | 32 |
| ST3 | Nováček | 1 | 65 |
| ST4 | Nováčková | 1 | 12 |
| ST5 | Nový | 1 | 5 |
| ST6 | Nová | 1 | 1 |
| ST7 | Novák | 1 | 25 |

PŘEDMĚT:

| C_PRED | NÁZEV | KREDITY | KLASIF |
|--------|---------------|---------|--------|
| 36V1A | Výpočetní ... | 4 | Z |
| 36V1B | Výpočetní ... | 4 | Z |
| 36V2A | Výpočetní ... | 4 | Z,ZK |
| 36V2B | Výpočetní ... | 4 | Z,ZK |
| 01MA1 | Matema ... | 4 | Z,ZK |
| 02FY1 | Fyzika ... | 4 | KLZ |
| 36DBS | Databázo ... | 4 | Z,ZK |

ZÁPIS:

| C_SEM | C_Z_ST | C_Z_PRED | C_PAR | ZAPOČET | ZNÁMKA |
|-------|--------|----------|-------|---------|--------|
| 962 | ST1 | 36V2A | 1 | | |
| 962 | ST2 | 36V2A | 1 | | |
| 962 | ST3 | 36V2A | 1 | | |
| 962 | ST4 | 36V2A | 1 | | |
| 962 | ST5 | 36V2A | 2 | | |
| 962 | ST6 | 36V2A | 2 | | |
| 962 | ST7 | 36V2A | 2 | | |
| 962 | | | | | |
| 962 | | ... | | | |
| 962 | | | | | |
| 962 | ST4 | 36V2B | 1 | | |
| 962 | ST5 | 01MA1 | 1 | | |
| 962 | ST6 | 02FY1 | 1 | | |
| 962 | ST7 | 36DBS | 1 | | |

Příklad dotazu: seznam předmětů studenta ST4 v semestru 962:

```
SELECT NAZEV FROM ZAPIS JOIN PŘEDMĚT
WHERE (C_Z_PRED=C_PRED AND C_Z_ST= ST4 AND C_SEM=962)
```

Množina operací relačního modelu

- založ novou relaci (tabulku)
- vyber n-tice (řádky) zadaných vlastností ze zadané relace (tabulky) a vytvoř z nich novou relaci (operace **selekce**)
- vytvoř novou relaci (tabulku) z existující relace (tabulky) **projekcí** na zadané atributy
- vytvoř novou relaci (tabulku) ze zadaných relací (tabulek) jako podmnožinu jejich kartézského součinu (operace **spojení**)
- přidej nové n-tice (řádky) do dané relace (tabulky) (**INSERT**)
- vymaž n-tice (řádky) zadaných vlastností ze zadané relace (tabulky) (**DELETE**)
- ve vybraných n-ticích (řádcích) zadané relace (tabulky) změň hodnoty zadaných atributů (**UPDATE**)
- vytvoř novou relaci (tabulku) ze zadaných relací (tabulek) pomocí množinových operací sjednocení, průnik, rozdíl (**UNION, INTERSECTION, EXCEPT**)
V souladu s tím, že relace je množinou n-tic u těchto operací chápeme tabulku jako množinu řádků

Jazyk pro definici dat - DDL

Příklad:

```
Create table STUDENT
```

```
(c_stud      Number      Primary key;  
jméno       VarChar    Not Null;  
rok         Number     Check rok in (1,2,3,4,5,6);  
skupina     Number     Not Null;  
adresa      Varchar);
```

```
Create index on STUDENT (rok, skupina)
```

Výsledek DDL příkazů se ukládá do datového slovníku databáze

Indexem rozumíme pomocnou tabulku, která obsahuje dva sloupce:

- Setříděný klíč původní tabulky, relace
- Odkaz na klíč v původní tabulce, relaci

Jazyk pro manipulaci s daty - DML

Jazyky pro manipulaci s daty se člení do dvou skupin:

–**Procedurální jazyky**, které od programátora požadují, aby v aplikačním programu zadával, se kterými daty má být manipulováno a jak má manipulace proběhnout, aby bylo dosaženo požadovaného výsledku.

–**Neprocedurální jazyky**, které od uživatele požadují zadat se kterými daty má být manipulováno a co se má s určenými daty provést. Otázku jak manipulace proběhne se ponechává na databázovém procesoru. Typický reprezentant – **jazyk SQL**

SQL Structured Query Language

Standardizovaný [dotazovací jazyk](#) používaný pro práci s daty v relačních databázích

SQL příkazy se dělí na čtyři základní skupiny:

- Příkazy pro manipulaci s daty ([SELECT](#), [INSERT](#), [UPDATE](#), [DELETE](#), ...)
- Příkazy pro definici dat ([CREATE](#), [ALTER](#), [DROP](#), ...)
- Příkazy pro řízení přístupových práv ([GRANT](#), [REVOKE](#))
- Příkazy pro řízení [transakcí](#) ([START TRANSACTION](#), [COMMIT](#), [ROLLBACK](#))
- Ostatní nebo speciální příkazy

Příklad neprocedurálního jazyka DML - SQL

- vyhledání a výběr n-tic(řádků) zadaných vlastností
- vkládání nových n-tic(řádků) do relační tabulky
- úprava existujících n-tic(řádků) v relační tabulce
- vymazání n-tic(řádků) z relační tabulky

Příklad:

- vložíme nový řádek do tabulky studentů
`INSERT INTO STUDENT(C_ST,JMENO,ROCNIK,SKUPINA)
VALUES('ST4', 'Nováčková', 1, 12);`
- Na řádcích, kde je chybné příjmení 'Muller' opravíme příjmení na 'Müller'
`UPDATE STUDENT
SET JMENO='Müller' WHERE JMENO='Muller';`
- zrušíme všechny studenty 1. ročníku
`DELETE FROM STUDENT WHERE ROCNIK=1;`

Dotaz – “koření práce s databází”

- *D1: Vyber z tabulky STUDENT jediný řádek, který se týká uživatele-studenta, který s aplikací právě pracuje (je přihlášen pomocí svého uživatelského jména a hesla).*
- *D2: Vyber z tabulek ZÁPIS a STUDENT záznamy o všech studentech, kteří si aktuální předmět zapsali.*
- *D3: Vyber z tabulek ZÁPIS a STUDENT záznamy o všech studentech, kteří chodí na cvičení z aktuálního předmětu 'ke mně', tedy jsou zapsáni do cvičebních paralelek, které uživatel-učitel učí.*
- *D4: Vyber z tabulky TERMÍNY řádky, které se týkají neuzavřených zkušebních termínů předmětů, které má přihlášený uživatel-student zapsané a ještě neabsolvované.*

Zadání dotazu technikou QUERY BY EXAMPLE

D5: Vyber z tabulky TERMÍNY řádky, které se týkají zkušebních termínů pouze předmětu '36V2A', u nichž zkouší učitel, jehož jméno je podobné vzoru 'Man%'.

| 54201 FEL | | | Přihlášení studenta na zkoušku | | | Strana 1/2 18.06.98 | | |
|-----------|-------|-------|--------------------------------|--------------------|-----|---------------------|------|-------|
| Datum | Čas | Místo | Kód | Název | Vol | Zkoušející | Pozn | Uzáv. |
| 18.06. | 09:00 | K1B | 36V2A | Výpočetní technika | 4 | Mannová Bož | | 16.06 |
| 22.06. | 09:00 | K1B | 36V2A | Výpočetní technika | 9 | Mannová Bož | | 19.06 |
| 23.06. | 09:00 | K1B | 36V2A | Výpočetní technika | 22 | Mannová Bož | | 19.06 |
| 24.06. | 09:00 | K1B | 36V2A | Výpočetní technika | 27 | Mannová Bož | | 22.06 |
| 25.06. | 09:00 | K1B | 36V2A | Výpočetní technika | 13 | Mannová Bož | | 23.06 |
| 29.06. | 09:00 | K1B | 36V2A | Výpočetní technika | 9 | Mannová Bož | | 26.06 |
| 30.06. | 09:00 | K1B | 36V2A | Výpočetní technika | 13 | Mannová Bož | | 26.06 |

F1=Náp F4=Kon F7,8=Dotazy Enter=Přihl/Odhl Esc-F7=Pozn PgDn=Zkoušky

Transakce

Uživatel v roli UČITEL změní záznamy o zápisech studentů na zvolený předmět tím, že doplní informaci o příslušné klasifikaci. Může měnit více záznamů najednou. Po zapsání zápočtu několika studentům má dvě možnosti:

- **Odvolat** dosud provedenou práci. To se provádí explicitní funkcí ROLLBACK.
- **Potvrdit** dosud provedenou práci. To se provádí explicitní nebo implicitní funkcí COMMIT (potvrzení).

Objemu práce mezi dvěma potvrzeními se říká **transakce**.

Je to kvantum práce, které se buď provede celé, nebo se z něj neuplatní nic.

Příčiny pro odvolání:

- uživatel si „to“ rozmyslem a účinky dosud provedené práce **explicitně** odvolává
- n-tý krok transakce nebylo možné provést z důvodu logiky aplikace a z toho důvodu je třeba účinky předchozích kroků **explicitně** odvolat
- Při provádění n-tého kroku transakce došlo k přerušení(chybě) a z toho důvodu se **implicitně** odvolají účinky předchozích kroků

Manažer databáze

Manažer databáze je základní modul databázového procesoru, který je zodpovědný za tyto úlohy:

- Interakce s aplikačními programy
- Kontrola kvality dat - prosazování stanovených integritních omezení
- Kontrola přístupových práv k datům
- Průběžné zaznamenávání změn dat do žurnálu a zotavení dat po nehodě
- Koordinace přístupu k datům současně pracujících aplikací/uživatelů

Uživatelé databázového systému

- Správce databáze
- Příležitostný uživatel
- Aplikační programátoři
- Naivní uživatelé

Databázové systémy v počítačové síti

architektura klient – server:

- dvouvrstvá architektura
 - **databázový server,**
 - **mnoho (tlustých) klientů**
- třívrstvá architektura
 - **databázový server**
 - **aplikační server, dnes zahrnuje Web server**
 - **mnoho (tenkých) klientů**

Pozn.:

tlustý klient – běžně aplikační program na samostatném PC

tenký klient – typicky Web browser nebo ANSI terminál

Distribuovaná databáze:

- několik uzlů sítě, na každém jeden fragment databáze, každý fragment obsahuje jiná (svá) data – náročné na spolehlivost a trvalost spojení mezi uzly
- několik uzlů sítě s vlastními (lokálními) databázemi. Každá lokální databáze obsahuje „vlastní“ data a repliky dat z ostatních uzlů. Repliky se pravidelně osvěžují. Možnost provozu lokálních databází bez spojení s ostatními uzly.

Úvod

Dotazy

SELECT jaké sloupce chci získat

FROM zdroj, kde se mají data hledat

WHERE jaká kritéria musí výsledek splňovat

ORDER BY podle kterých sloupců se má výsledek řadit

Základní syntaxe příkazu SELECT:

SELECT výčet_atributů_oddělený_čárkou nebo znak * (všechny atributy),
FROM jméno_tabulky případně FROM tabulka_1 NATURAL JOIN
tabulka_2,

WHERE filtr relační operátory <, >, <=, >=, =, <> - operandy jsou atributy (jména sloupců) nebo konstanty,

- konstanty - čísla bez apostrofů, řetězce v apostrofech 'řetezec',
- jednotlivé filtry můžeme kombinovat logickými spojkami AND, OR a NOT (a použít závorky pro prioritu),
- operátor **LIKE** - možnost použít "velmi omezený regulární výraz"
 - % ... skupina znaků (i prázdná),
 - _ ... právě jeden znak
 - příklad: prijmeni like 'Nov%'

ORDER BY seznam_sloupců - zajišťuje řazení.

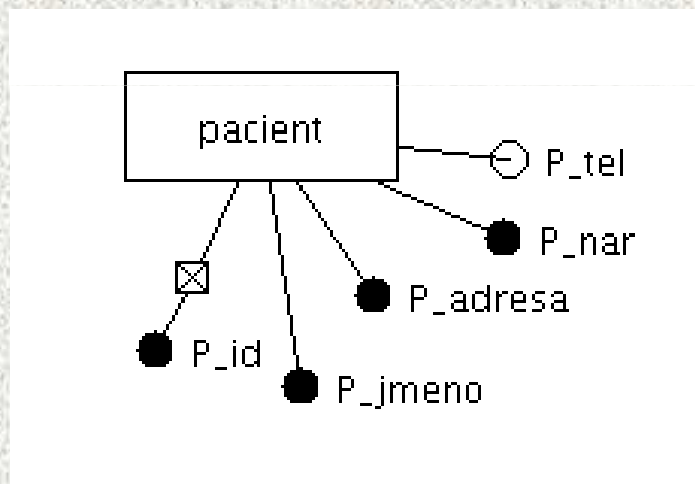
SELECT - poznámky

- Pokud se ve výstupu objevují stejné n-tice a my tento případ chceme vyloučit, uvede se bezprostředně za klíčové slovo SELECT slovo DISTINCT.
- SQL není case sensitive.
- Každý SQL příkaz je ukončen znakem ";"
- Při troše dobré vůle můžete zápis SQL příkazů "číst" jako požadavek v "přirozeném jazyku"

Příklad

- Jména a adresy pacientů, jejichž adresa začíná Picadili a nebo je abecedně dále, seřadit podle data narození:

```
select P_jmeno, P_adresa from pacient where P_adresa >= 'Picadili'  
order by P_nar
```

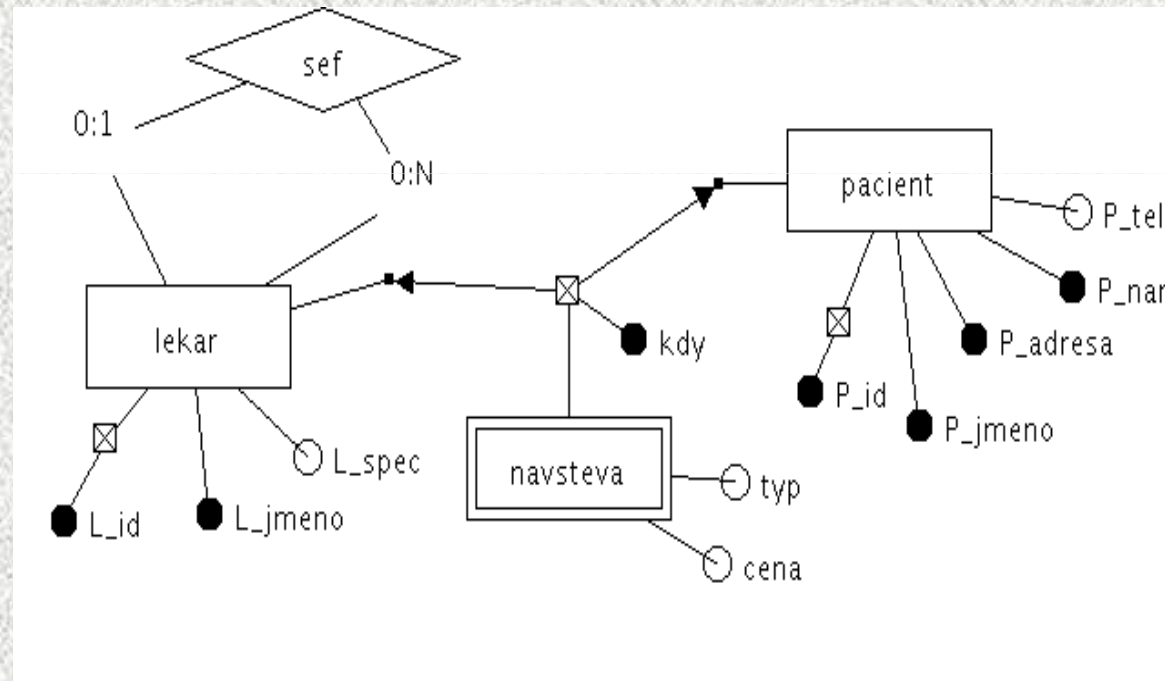


Příklad 2

- Čísla lékařů, které navštívila pacientka Chatrná.

```
select id_l from pacient join navsteva using (p_id)
```

```
where p_jmeno = 'Chatrná';
```



Úvod

