

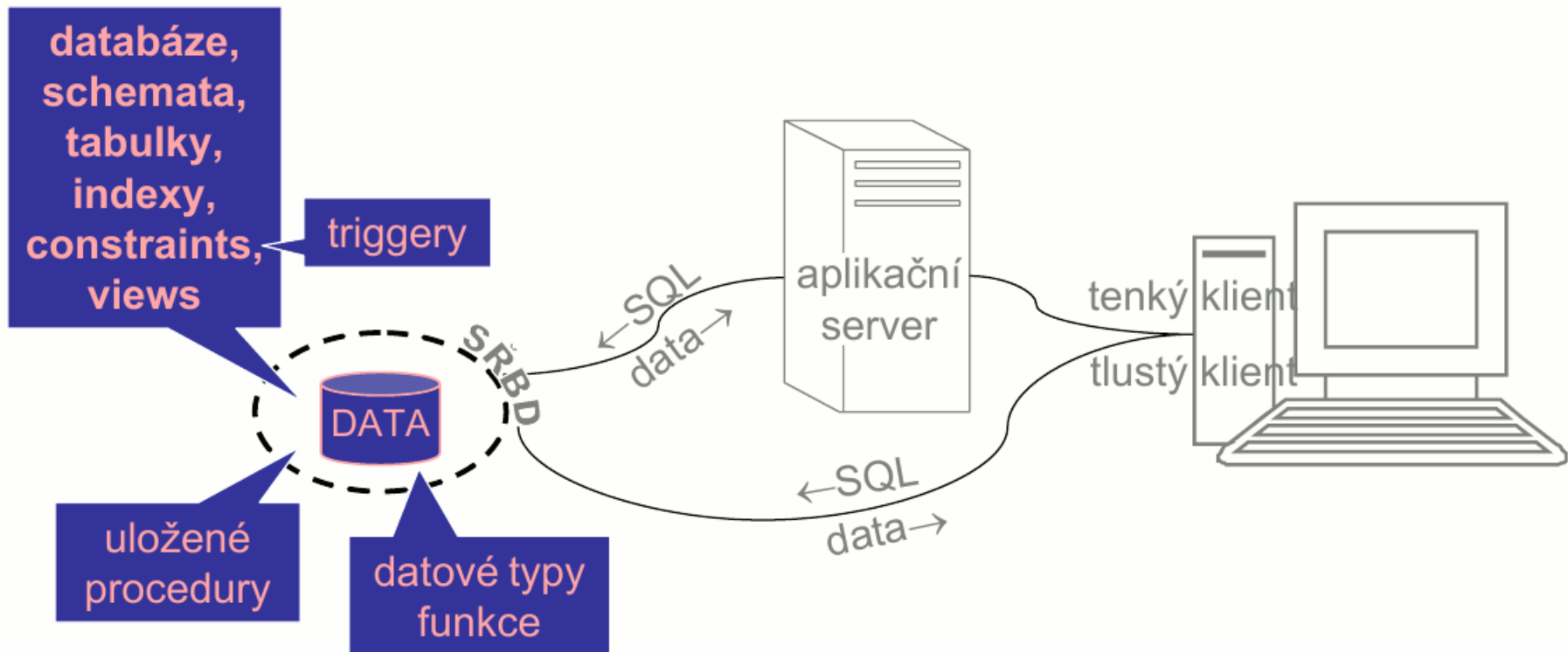
# Databáze I

## 5. přednáška

Helena Palovská  
palovska@vse.cz

SQL jazyk definice dat -  
- DDL (data definition language)

# Objekty SQL databáze



# Základní

**databáze,  
schemata,  
tabulky,  
indexy,  
constraints,  
views**



# Databáze/schéma

- Logické členění spravovaných databázových objektů na celky odpovídající oblastem zájmu či zodpovědnostem
- V různých SŘBD různá koncepce
  - (ne)překrývání schémat
  - vztah uživatelů a schémat
    - vlastnictví, přístup k objektům schématu

# Definice tabulky

- Určení sloupců a jejich datových typů
- Primární klíče a další constraints
- Způsob fyzické organizace a metody správy

# K čemu jsou datové typy

- Validace vstupu
- Operace, funkce, metody
- Význam, užití
- Objem, čas
  - uložení
  - zpracování operací
  - vyhledávání, indexace

# Základní datové typy

**číslo**  
*množství, počet*

přesná – pevná desetinná čárka

měna

nepřesná – plovoucí desetinná čárka

**výpočty**

**text** dané maximální délky  
*název, jméno, kód*

**zobrazení, porovnání, slepení**  
manipulace po písmenech

**datum a čas**  
*kdy*

**co bylo/je/bude dříve, výpočet intervalu**  
Year, Month, Day

logická hodnota  
*ano/ne*

**logické výrazy**

typ vyjmenovaných hodnot  
*zařazení do nějaké kategorie*

**vyhledání, třídění**



# Datové typy v SQL

## Základní datové typy

- číslo
- textový údaj
- datum a čas
- ano/ne (logický)

## Nové datové typy

- bitové řetězce
- volný text
- obrázky, audio, multimedia
- prostorová data
- časové intervaly
- ...

+ „uživatelsky“ definované typy

# Primární klíč

- Slouží pro odkazování na řádek tabulky
  - Musí být unikátní (UNIQUE)
  - Musí existovat (tj. NOT NULL)
  - Neměl by se měnit
    - V některých SŘBD to ani nejde, pokud na řádek existuje odkaz.
      - tzn. neměl by vzniknout důvod ho měnit !
  - Významový/nevýznamový
    - event. automaticky generovaný

# Další constraints

- NOT NULL, DEFAULT
- UNIQUE
- CHECK()
- FOREIGN KEY

<http://krokodata.vse.cz/SQL/CreateTable>  
<http://krokodata.vse.cz/SQL/AlterTable>

# Referenční integrita

- SŘBD musí zajistit, aby odkazy **cizí klíč → primární klíč** nebyly do prázdna.
- Stát by se to mohlo
  - zadáním chybného „odkazu“ (ve slave řádku)
  - změnou či smazáním odkazovaného (master) řádku

# Pravidla pro mazání „master“ řádků

- Pravidlo je součástí definice FOREIGN KEY
- Může být:
  - nesmí se mazat, když existuje odkaz (RESTRICT)
  - když se smaže, všechny odkazy na něj se přepíše na prázdnou hodnotu (SET NULL)
  - se smazáním "master" se smažou k němu příslušné "slave" (CASCADE)

# Indexy

- Jsou pomocné databázové struktury sloužící k **urychlení vyhledávání záznamů**
  - vyhledávání podle daných příznaků
    - podle primárního klíče
    - podle hodnot v dané položce či skupině položek
- SŘBD je udržuje v aktuálním stavu

# Indexy

- Nejčastější způsoby organizace:
  - B-stromy
  - Bitové indexy
  - Hashované/clusterované indexy

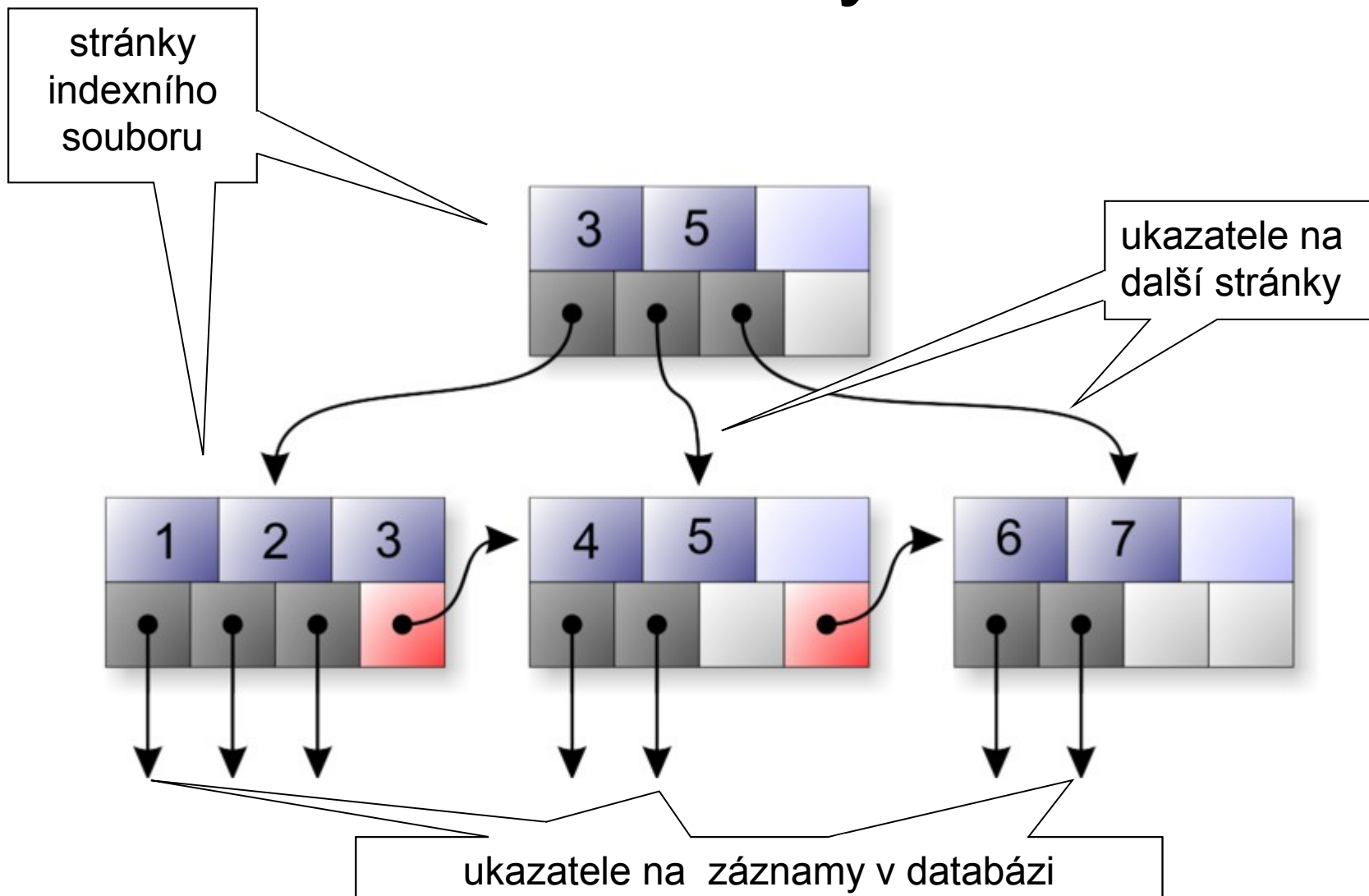
# B-stromy

- Jsou optimalizovány pro čtení stránek\* při vyhledávání
- Mají nejlepší dynamické vlastnosti
  - při reakcích na aktualizace v databázi

\* ) stránka je jednotkou čtení/zápisu z/do vnější paměti

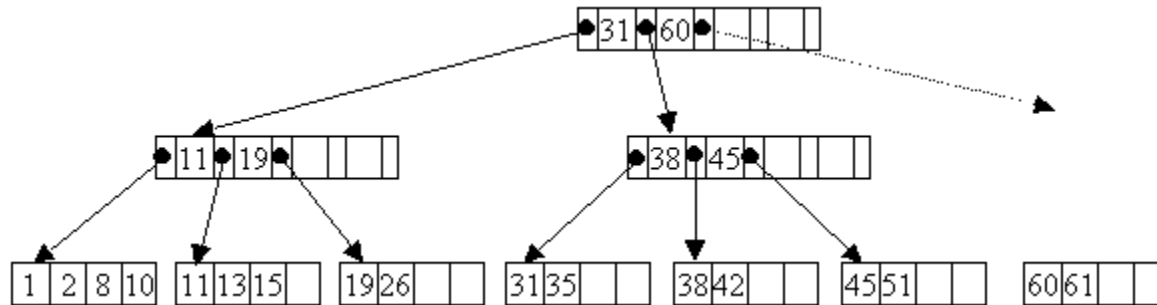


# B-stromy



# B-stromy

- Na 1 stránce mohou být stovky klíčů  
3 až 4 úrovně stromu stačí pro  $10^6$  až  $10^8$   
~ 3 až 4 čtení k nalezení záznamu



- Listy sekvenčně propojeny  
⇒ uspořádání záznamů podle ind. klíče

# Bitové indexy

- Vhodné pro malou doménu (množinu hodnot) vyhledávacího klíče
- Pro každou hodnotu vytvořen bitový vektor
  - pozicím odpovídají jednotlivé záznamy
- Možná i vhodná komprimace vektorů
  - rozbalování „od začátku“

# Hash indexy

- Hash funkce vypočítá adresu stránky uložení záznamu
  - Nutné značné rezervy pro ukládání záznamů
- Nejrychlejší „index“
- Můžeme mít nejvýše jeden hash index (clustrovaný index)  
protože je to fyzické uložení záznamu

# Pohledy

- Odvozené tabulky definované jako výsledek určitého dotazu
  - uživatelský „pohled“
    - přístupová práva
    - legacy pohledy
    - usnadnění formulace požadavků
  - materializovaný pohled
    - pro zefektivnění zpracování požadavků

# Pohledy "with check option"

- Aktualizovatelné pohledy –  
slouží pro zápis či mazání dat  
– aktualizace nesmí způsobit  
nekonzistenci „pohledu“

SQL jazyk manipulace dat -  
- DML (data manipulation language)

Pro dotazy slouží SELECT...

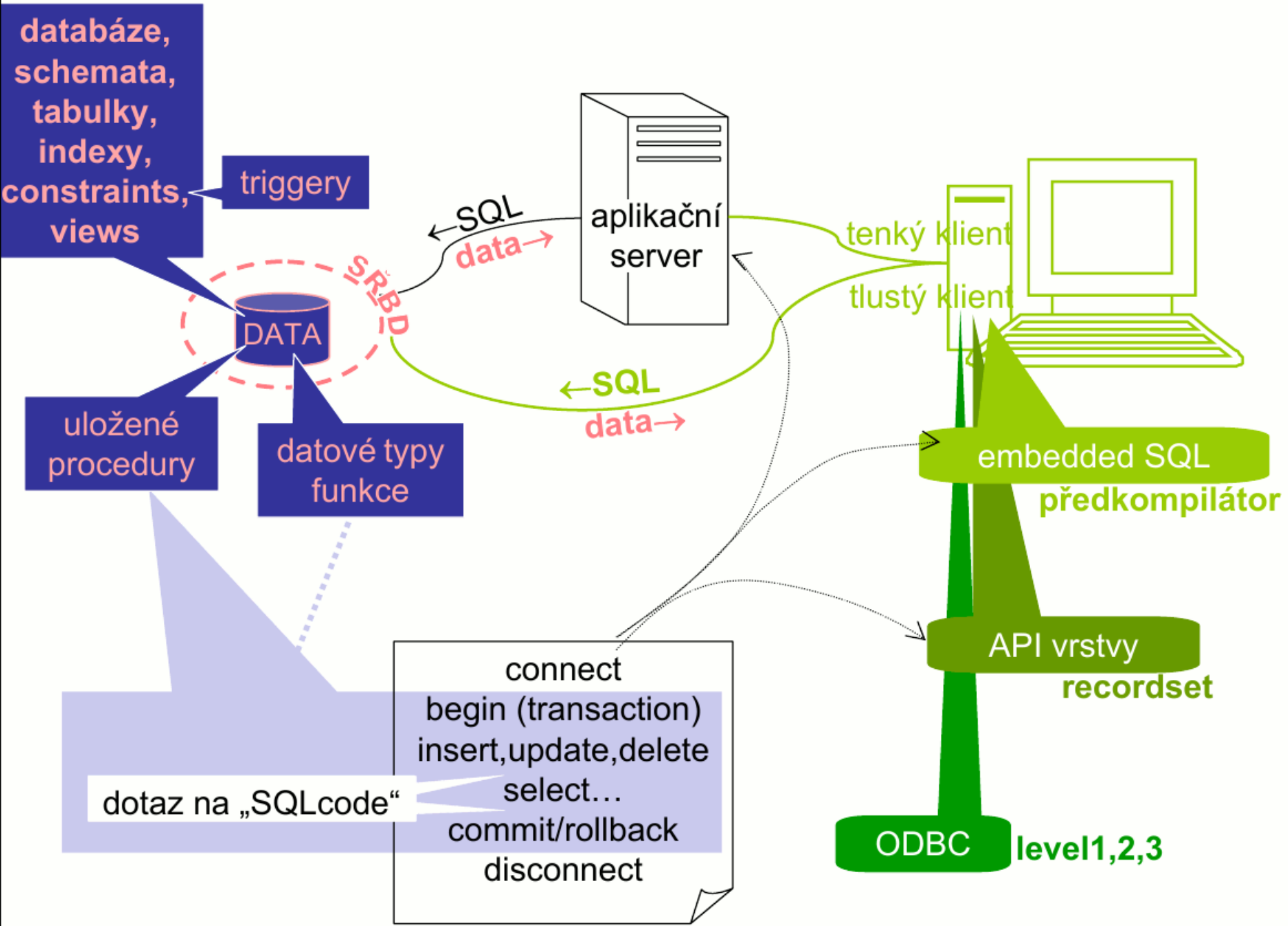


# Pro aktualizaci dat

- **INSERT**  
vložení nového řádku
- **UPDATE**  
změna dat v existujícím řádku
- **DELETE**  
smazání řádku

# Manipulace dat v aplikaci

- Aktualizační příkaz nemusí být proveden
  - nedostatečná oprávnění
  - porušil nějaké omezení, a tím způsobil chybu
  - jiná chyba
    - nedostatek prostředků na provedení
    - uváznutí
    - ...
- Jak to v aplikaci zjistí?
  - návratové kódy příkazu



# Logická jednotka zpracování dat

- Buď se mají udělat všechny z určitého seznamu operací, nebo žádná z nich.
- K tomu si musí SŘBD vést žurnál operací, a umět se vrátit zpět k danému bodu.
- Zpracování dělíme na tyto logické jednotky – **transakce**.

# Transakce

- Jednotka zpracování dat – ACID
  - Atomická: má se provést vše nebo nic. Pokud nelze dokončit, má se navrátit stav zpracovávaných dat před začátek transakce
  - Udržuje databázi v konzistentním stavu. Před dokončením transakce mohou být data „špatně“ – neodrážejí fakta tak, jak jsou – ale po potvrzení jsou „správně“.

# Transakce

- Jednotlivé transakce jsou navzájem izolovány: „nevědí“ o nepotvrzených stavech dat měněných jinou transakcí. (V SQL lze zmírnit nastavením transaction isolation level)
- Změny provedené potvrzenou transakcí jsou trvalé
- ACID zkratka:  
Atomicity, Consistency, Isolation, Durability

# Řízení transakcí v SQL

- Určení začátku transakce
  - BEGIN, event. BEGIN TRANSACTION
  - režim může být nastaven tak, že jednotlivé příkazy jsou automaticky považovány za samostatnou transakci (obvykle při interaktivním zadávání příkazů)
  - po ukončení předchozí transakce někdy automaticky začíná následující

# Řízení transakcí v SQL

- Ukončení transakce
  - COMMIT – požadavek na trvalé uložení změn
    - pokud se transakce dostala do stavu chyby (návratový kód některého příkazu sdělil chybu), pak COMMIT není proveden, je nutný ROLLBACK
  - ROLLBACK – požadavek vrátit změny před začátek transakce
    - pokud nastala chyba (viz výše)
    - pokud stav dat nevyhovuje



# Transakce a zámky

- Automaticky se zamykají pro zápis záznamy měněné transakcí
  - lze explicitně zamknout i příkazem
- Zámky pro čtení závisí na nastavení TIL
- Zámky se uvolňují s ukončením transakce

# Uváznutí

- Uváznutí zámků mezi jednotlivými transakcemi
- SŘBD obvykle mají proces detekce deadlocků, násilně ukončí některou z transakcí dle vlastních pravidel

# Deadlock – příklad

1. transakce:

```
update ucty  
  set stav=stav-100  
  where cislouctu=501;
```

```
update ucty  
  set stav=stav+100  
  where cislouctu=1230;
```

2. transakce:

```
update ucty set  
stav=stav+200 where  
cislouctu=1230;  
update ucty  
set stav=stav-200  
where cislouctu=501;
```

Konec