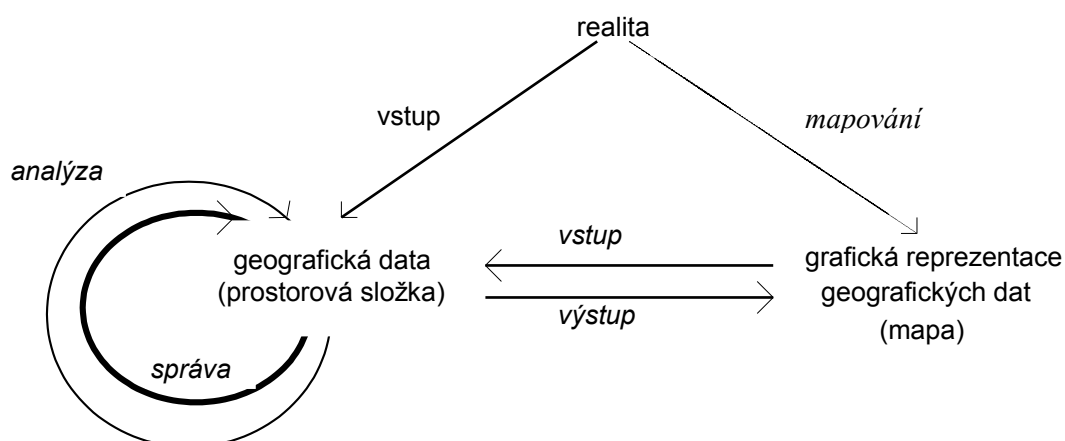


Lekce 6 - Správa prostorových dat

1. Cíle lekce.....	1
2. Organizace dat.....	1
2.1 Souborově orientovaný přístup.....	2
2.2 Databázově orientovaný přístup	2
3. Databáze v GIS.....	2
4. Možnosti databázové přístupu.....	6
4.1 Bezešvá data.....	6
4.2 Transakce, verzování a historie dat.....	8
5. Uchování rastrových dat	9
6. Hierarchické datové struktury.....	10
6.1 Indexování pixelů - příklad.....	10
6.2 Kvartérní strom.....	10
6.3 Výhody hierarchických datových struktur.....	11
6.4 Porovnání uložení dat v souborech, hierarchických strukturách a relačních databázích.....	11
7. Zkušební otázky.....	11

1. Cíle lekce

- Provést diskusi o správě (managementu) geografických dat jako jedné ze základních funkcí GIS:
 - vstup
 - **management (správa)**
 - analýza
 - výstup



- Uvést možnosti různého uchování prostorových dat v souborech, v hierarchických datových strukturách a v relačních databázích

2. Organizace dat

Systémy pro zpracování větších množství dat se objevily v šedesátých letech v oblastech evidence materiálu, kusovníky, příprava výroby. Síťový přístup a transakční zabezpečení je poprvé rutinně implementováno v systémech rezervace letenek.

Při zpracování velkého množství dat můžeme rozlišit dva přístupy

- souborově orientovaný přístup
- databázově orientovaný přístup

2.1 Souborově orientovaný přístup

Souborově orientovaný přístup byl uplatňován v počátcích zpracování dat, ale přetrvává (zejména v GIS dodnes). Je charakterizován těmito nevýhodami

- závislostí aplikace na fyzické struktuře dat
- nepružným přístupem k datům, pokud s nimi pracuje více uživatelů

Obě nevýhody může mít i špatně vytvořená aplikace, i když pracuje s daty uloženými v databázi.

Například:

- aplikace při aktualizaci dat zahájí první transakci, přečte data z databáze do paměti, ukončí první transakci, data zpracuje v paměti, zahájí druhou transakci, zapíše data zpět do databáze a ukončí druhou transakci - tato aplikace nefunguje korektně v multiuživatelském režimu
- aplikace zpracovává data v relační databázi, ale datový model není zveřejněn, takže data nejsou přístupná jiným aplikacím
- aplikace využívá pro svoji práci datový model, který je sice zveřejněn, ale je složitý (aplikace je například „přeparametrizována“ - zpracování dat je řízeno mnoha parametry, které jsou uloženy v databázi), takže žádná jiná aplikace s daty nemůže pracovat, protože její vytvoření by bylo příliš drahé

Takové aplikace mají samozřejmě i všechny níže uvedené nevýhody databázového přístupu.

Aplikace s uvedenými nevýhodami se bohužel v GIS vyskytují velmi často.

2.2 Databázově orientovaný přístup

Databáze využívají pro přístup k datům systém řízení báze dat (SŘBD, Data Base Management System - DBMS)

Databázové modely:

- hierarchický model
- síťový model
- relační model
- objektově orientovaný model
- relační model s prvky objektového přístupu

Výhody databázového přístupu

- centralizované řízení
- efektivní sdílení dat
- datová nezávislost aplikací
- snadná implementace nových aplikací, cena vývoje aplikací
- přístup k datům bez programování
- řízená redundance dat
- uživatelské pohledy na data
- data jsou lépe dokumentována
- je přesně definován datový model a tím je posílena konzistence dat
- bezpečnost dat, zajištění přístupových práv

Nevýhody databázového přístupu

- cena uložení dat
- složitost
- centralizované riziko

3. Databáze v GIS

Neprostorová data jsou v současné době obvykle uchovávána v relačních databázích¹. Použití relačních databází pro uchovávání geografických dat a především jejich prostorové složky vede k některým komplikacím:

- geografická informace je komplexní, a proto je aktualizace geografických dat prováděna složitými transakcemi, které se týkají většího počtu záznamů v několika tabulkách

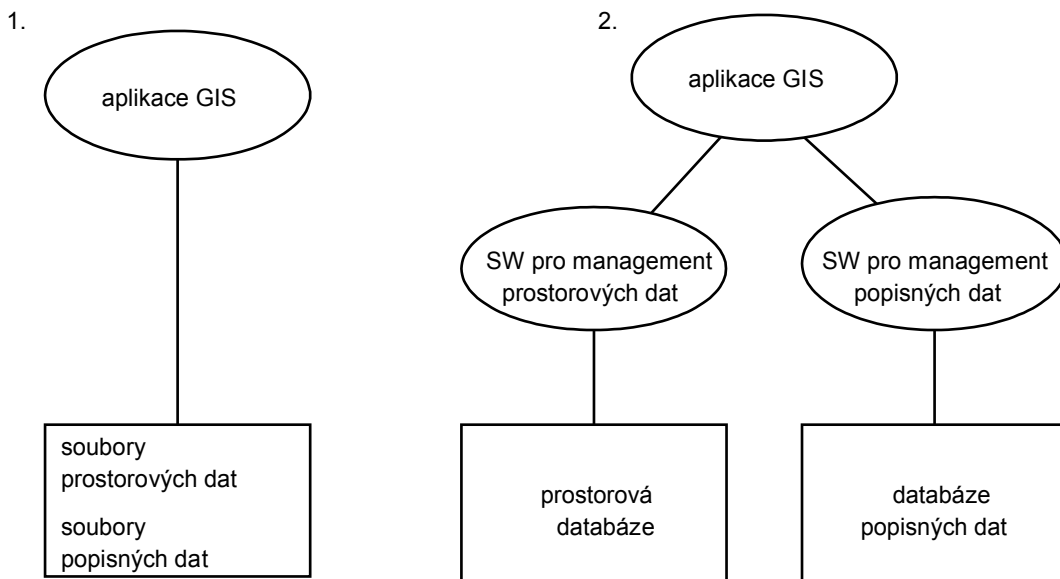
¹ Objektové databáze nejsou v současné době rutinně používány pro práci s větším množstvím dat. Důvodem je jednak jednoduchost relačních databází a vysoká důvěra projektantů i uživatelů v jejich robustnost, jednak skutečnost, že relační databáze jsou obohacovány objektově orientovanými prvky.

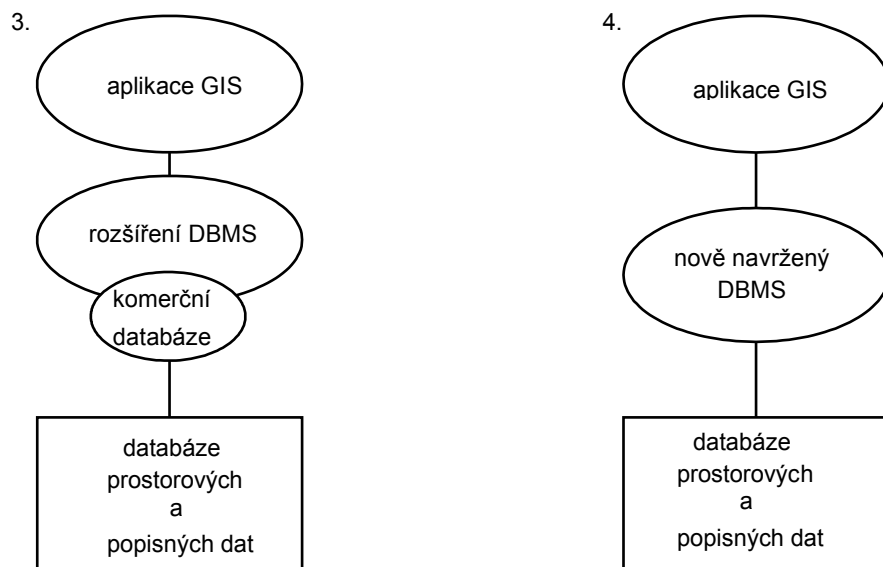
- prostorová data mají proměnnou délku záznamů v závislosti na počtu souřadnic, je problém je efektivně uložit v normalizovaném tvaru v čistě relačním prostředí
- manipulace s prostorovými daty vyžaduje pracovat s relacemi jako sousednost, incidence, překryv, apod., tato práce není podporována v relačních databázích
- relačním databázím chybí grafické prostředí nezbytné pro práci s prostorovými daty
- prostorové vztahy mezi geografickými daty vyžadují složitý systém zamykání vět při transakcích
- pro uložení geografických dat je požadováno několik datových typů, například
 - obrazy
 - slova
 - souřadnice
 - komplexní objekty
- v GIS má význam pořadí záznamů uložených v databázi - např. pořadí bodů v lomené čáře (to je v relační databázi nevýznamné)
- některá integritní pravidla jsou složitá (hrany se nemohou křížit jinde než v uzlovém bodě)

Lze rozpoznat čtyři strategie v organizaci prostorových dat pomocí databází.

1. Systém obsahuje speciální SW pro správu prostorových i popisných dat, tedy nevyužívá databázi. Tuto strategii používají stolní a úzce specializované systémy. Produkty této kategorie mají své opodstatnění zejména tam, kde úlohy vyžadují velmi speciální datové struktury z hlediska optimálního přístupu a využití běžných databázových produktů by proces neúnosně zpomalilo.
2. V systému je implementován komerční DBMS pro uchování popisných dat, a vyvinut speciální SW pro správu prostorových dat. Kategorie prostředků, které používají tuto strategii, je dnes nejrozšířenější kategorií. Prostředky spojují propracované technologie relačních databází a speciální datové struktury reprezentující prostorovou složku dat. Někdy zahrnují výkonné grafické editory CAD systémů. Spojení mezi atributovou a prostorovou složkou dat je zabezpečeno klíčem, který je obsažen jak v databázové tabulce RDBMS, tak v příslušném elementu speciálního souboru prostorových dat. Příklad: Arc GIS, v minulosti MGE.
3. V systému je využita (obvykle relační) databáze pro uchování popisných i prostorových dat, pro práci s prostorovou složkou geografických dat je systém (DBMS) rozšířen o speciální SW. Vazby mezi jednotlivými entitami systému jsou realizovány relačním přístupem. Příklad: System 9, GEOVIEW, LIDS/SDS, hodně firem o tento přístup své systémy rozšířilo (například Arc GIS a uložení dat v Oracle SDO).
4. Budoucnost: systém je založen na nově vyvinuté databázi, která umí zpracovávat integrovaně obě složky prostorových dat. V současné době realizují někteří producenti databází kroky v tomto směru. Příkladem může být Spatial Data Option relační databáze ORACLE (od verze 7.3). (aktuální stav SDO je pospán například v http://download.oracle.com/docs/cd/B19306_01/appdev.102/b14255/sdo_intro.htm#i877656)

Popsané strategie jsou znázorněny na následujících obrázcích.





Obr. 27 - Strategie v organizaci prostorových dat

Představu o tom, jak jsou ukládána prostorová data v produktech prvních dvou kategorií, si můžeme utvořit z popisu datových struktur uvedených v kapitole 2. Zde uvedme jako příklad prostředku kategorie 3 popis jednoduchého serveru prostorových dat, pracujícího v prostředí RDBMS.

Tabulka `server`, ve které budou uchovávána prostorová data:

```
create table server
(
  intern_id  number(10),
  id1       char(25),
  id2       char(25),
  status    char(10),
  typ_objektu char(10),
  objekt    long raw);
);
create unique index server_intern on server(intern_id);
```

Pomocná tabulka `server_index`, která zabezpečí přístup grid metodou:

```
create table server_index
(
  intern_id number(10),
```

```

        grid_id    number(10)
    );
create index server_index_idx1 on server_index(intern_id);
create index server_index_idx2 on server_index(grid_id);

```

Pomocné tabulky pro rozsahový výběr:

```

create table grid_selection
(
    grid_id    number(10)
);
create table ids
(
    intern_id  number(10)
);

```

Vkládání gridového indexu do serveru

```

/*
imin    - minimum x v gridu
imax    - maximum x v gridu
jmin    - minimum y v gridu
jmax    - maximum y v gridu
DBlink  - interní identifikátor v serveru
*/

for(i=imin;i<=imax;++i)
    for(j=jmin;j<=jmax;++j)
    {
        MakeInd(&i, &j, &DBsq);
        EXEC SQL INSERT INTO SERVER_INDEX (INTERN_ID, GRID_ID)
            VALUES (:DBlink, :DBsq);
    }

```

Rozsahový výběr ze serveru:

```

/*
imin    - minimum x v rozsahovem vyberu
imax    - maximum x v rozsahovem vyberu
jmin    - minimum y v rozsahovem vyberu
jmax    - maximum y v rozsahovem vyberu
DBlink  - interni identifikator v serveru
*/

/* inicializace pomocnych tabulek */

EXEC SQL DELETE FROM GRID_SELECTION;
EXEC SQL DELETE FROM IDS;

```

```

for(i=imin;i<=imax;++i)
  for(j=jmin;j<=jmax;++j)
  {
    MakeInd(&ii,&jj,&DBsq);
    EXEC SQL INSERT INTO GRID_SELECTION (GRID_ID) VALUES
      (:DBsq);
  }

/* v tabulce GRID_SELECTION jsou vsechny elementarni ctverce */

EXEC SQL INSERT INTO IDS (INTERN_ID)
  SELECT INTERN_ID FROM SERVER_INDEX WHERE GRID_ID IN
  (SELECT GRID_ID FROM GRID_SELECTION);

/* v tabulce IDS jsou vsechny interni identifikatory
  rozsahoveho vyberu */

EXEC SQL DECLARE curs02 CURSOR FOR
  SELECT OBJEKT FROM SERVER
  WHERE INTERN_ID IN (SELECT INTERN_ID FROM IDS);

EXEC SQL OPEN curs02;
while fetch(...)
{
  .
  akce s vybranymi objekty
  .
}

```

4. Možnosti databázové přístupu

4.1 Bezešvá data

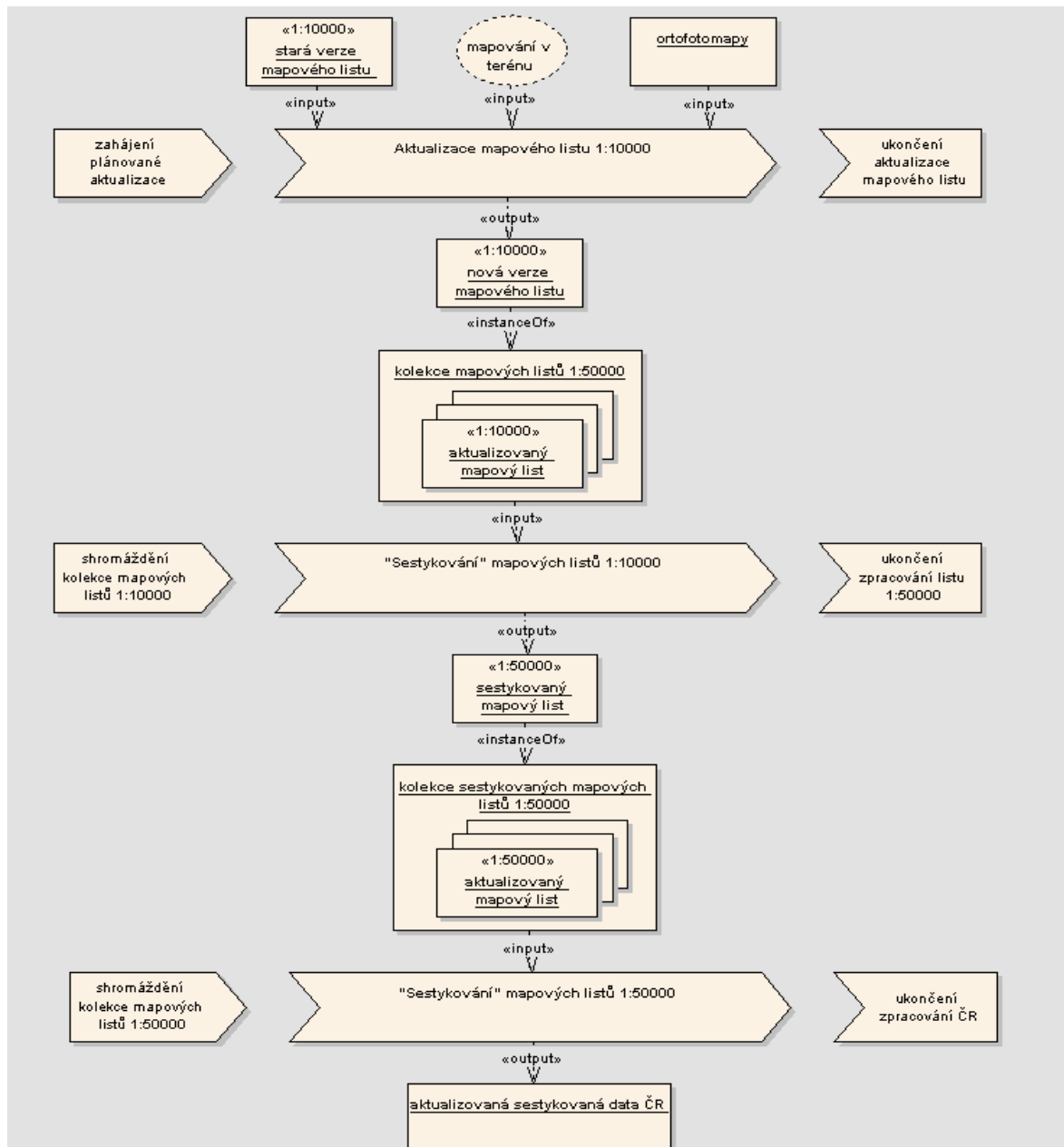
Před příchodem počítačové techniky a GIS byla prostorová data za větší území rozdělena do mapových listů a po těchto mapových listech aktualizována. Na rámech mapových listů vznikaly tzv. švy – objekty byly děleny na rámech mapových listů. I při nejpřísnějších organizačních opatřeních se stávalo, že docházelo k nesouladu na stycích mapových listů: objekt, který se vyskytoval na jednom mapovém listu, nepokračoval, nebo pokračoval chybně na listu sousedním. Proto byly hledány možnosti ukládání dat za celé území vcelku, beze švů, tedy tzv. bezešvých dat. I když bezešvá data lze udržovat i v souborech, reálný smysl má bezešvost až v okamžiku, kdy mohla být prostorová data ukládána do databáze.

Při správě bezešvých dat je vhodné, pokud se najdou nějaké přirozené oblasti – části území, podle kterých lze data rozdělit – příkladem může být digitální katastrální mapa a katastrální území. Pokud takové oblasti neexistují (příkladem může být topografická mapa), může bezešvost vést ke vzniku velkých objektů – například ploch, jejichž hranice jsou tvořeny až několika desítkami tisíc vrcholů. Tato situace klade zvýšené nároky na SW – musí si poradit stejně dobře s několika desítkami tisíc malých objektů (například s plochami o několika vrcholech) i s několika velkými objekty (například s již zmíněnými plochami o několika desítkách tisíc vrcholů).

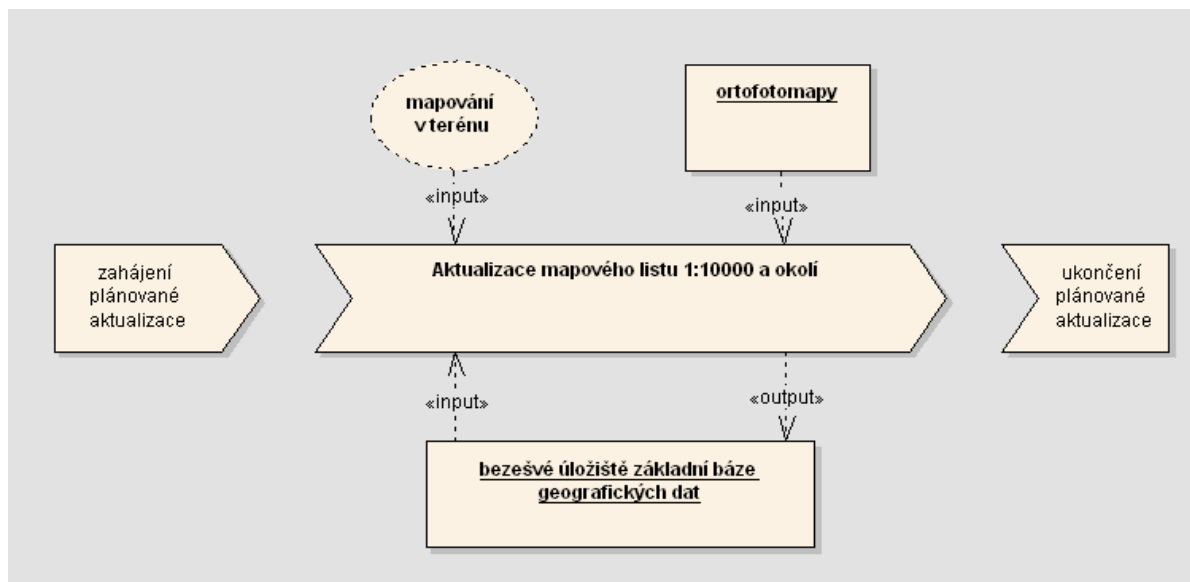
Bezešvé uložení dat vede ke změnám procesů s prostorovými daty, příkladem může být dřívější a stávající stav ZABAGED.

- Dříve - tři stavy dat:
 aktuální s nevyřešenými styky
 méně aktuální se styky vyřešeným v rámci listu 1:50 000
 neaktuální s vyřešenými styky v rámci celého území ČR
- Nyní – jeden centrálně uložený bezešvý korektní stav dat, což vedlo ke zvýšení produktivity výroby na téměř 1 500 mapových listů ročně

Dřívější stav



Současný stav



4.2 Transakce, verzování a historie dat

Dlouhé transakce

Na úrovni databáze je také řešen jeden z charakteristických rysů systému – podpora práce v režimu dlouhých optimistických transakcí a uchovávání historie. Při potvrzení transakce uživatel musí vyřešit všechny konflikty, tedy situace, kdy byl daný objekt od okamžiku vzniku transakce do okamžiku jejího potvrzení změněn jiným uživatelem. Při vyhodnocení konfliktu v rámci dané transakce se zohledňují tři stavy dat:

- stav aktuálních dat v okamžiku, kdy data vstupovala do dlouhé transakce
- stav aktuálních dat v okamžiku, kdy končí ukončení dlouhé transakce
- stav dat změněných danou dlouhou transakcí

Optimistická varianta řešení dlouhých transakcí je velmi výhodná v tom, že není nutno data pro účely aktualizace zamykat a omezovat tak práci ostatních uživatelů.

Současnost

Pro práci se stavy dat v čase a pro správu dlouhých transakcí lze použít prostředky databází, v databázi Oracle je to modul Oracle Workspace Manager. Modul ORACLE Workspace Manager uživatelům nabízí správu více verzí téhož záznamu v tabulce. Modul poskytuje:

- možnost registrace databázové tabulky pod systém správy verzí – dále odkazované jako verzované tabulky,
- vytváření hierarchicky organizovaných pracovních prostorů (workspace) pro práci s více verzemi téhož databázového záznamu současně; v každém pracovním prostoru se může vyskytovat

odlišná verze daného záznamu což je vlastnost, která se s výhodou uplatní při implementaci dlouhých transakcí,

- volitelně zamykání modifikovaných záznamů zamezujících jejich další změně v rámci jiných pracovních prostorů,
- funkce podporující řešení konfliktů vzniklých odlišnou změnou téhož záznamu v rámci různých pracovních prostorů,
- možnost vytvoření úložiště historických dat pro každou verzovanou tabulku spolu s vytvořením změnového pohledu (ukazuje, jak se data měnila v čase),
- možnost zobrazit data verzované tabulky tak jak vypadala v daném časovém bodě v minulosti.

Aktuální stav dat je udržován v základním pracovním prostoru (workspace) nazvaném LIVE.

Budoucnost

Jeden z požadavků může být umožnit práci s budoucím stavem prostorových dat, tedy umožnit zpracovat informace o navržených, ale ještě nerealizovaných změnách (projekty, geometrické plány, rozpracovaná změnová řízení). Tento požadavek je zajištěn pomocí mechanismu dlouhých transakcí resp. pomocí funkcí modulu Oracle Workspace Manager.

Navržený systém podporuje práci s různých založenými a pojmenovanými dlouhými transakcemi. Každá dlouhá transakce má definován tzv. pracovní prostor, který obsahuje platná data a data, která se aktualizovala v rámci této dlouhé transakce. Provedené změny (budoucí stav) včetně odvozených dat jsou přístupné v tomto pracovním prostoru resp. při přihlášení k dané dlouhé transakci. Při práci v rámci dané dlouhé transakce lze používat všechny běžné operace nad daty včetně kontrol a rutin na odvozování topologických dat. Určitá verze dat resp. určitá verze budoucího stavu dat je tedy přístupná přesměrováním zobrazovacího aparátu do pracovního prostoru určité dlouhé transakce.

Každá dlouhá transakce (a její pracovní prostor) obsahuje určitou verzi dat, přičemž aktuální data jsou v základním (LIVE) pracovním prostoru.

Minulost

Také pro uchovávání a zobrazování historie dat je využit modul Oracle Workspace Manager, který umožňuje verzovat data v tabulkách a pomocí pohledů zpřístupňovat obraz dat ve zvoleném časovém bodě nebo vytvářet změnová data za zvolený časový interval.

5. Uchování rastrových dat

- v matici řádek po řádku, každý pixel v jedné paměťové pozici, pokud je více vrstev, mohou být uchovávány v samostatně nebo v jedné paměťové pozici několik pixelů
- Freemanův řetězový kód pro uchování liniových prvků v rastroch (sousedí pixelu jsou očíslováni 0-7, do řetězu se zapisuje číslo dalšího sousedního pixelu v linii)
- komprese (run encoding) - místo pixelů jsou v paměťových pozicích dvojice (hodnota pixelu, počet pixelů) ve směru skenování

6. Hierarchické datové struktury

6.1 Indexování pixelů - příklad

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
1	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
2	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
3	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
4	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
5	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
6	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
7	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
8	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
9	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
10	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
11	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
12	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
13	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
14	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
15	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o

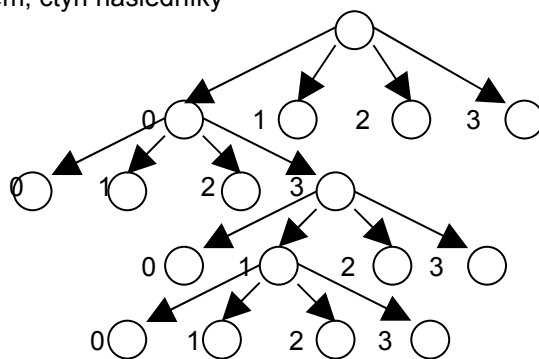
- čtverec 16x16 pixelů rozdělíme na čtyři podčtverce, očísujeme je 0, 1, 2, 3:

0	1
2	3

- čtverce 1,2 a 3 jsou homogenní, čtverec 0 rozdělíme na další čtyři podčtverce, očísujeme je 00, 01, 02 a 03
- čtverce 00, 01 a 02 jsou homogenní, čtverec 03 dále rozdělíme na 030, 031, 032, 033
- pouze čtverec 031 je nehomogenní, opět jej rozdělíme na čtverce 0310, 0311, 0312 a 0313
- rekurzivně jsme rozdělili prostor až jsme dosáhli jedné ze dvou podmínek:
 - všechny čtverce jsou homogenní
 - je dosaženo rozlišovací úrovně rastru
- podle Mortona jsou data zakódována takto: 16o 16o 16o 4o 1o 1X 1o 1o 4o 4o 64o 64o 64o

6.2 Kwartérní strom

Rozdělení území vždy do čtyř čtverců lze popsat kvartérním stromem - stromem, v němž má každý uzel, který není listem, čtyři následníky



Varianty kvartérních stromů:

- oktalové stromy pro 3D, krychle je rozdělena do osmi podkrychlí
- modelování povrchu Země oktahedronem, jehož povrch se skládá z osmi trojúhelníků, každý trojúhelník je dělen na čtyři další trojúhelníky

6.3 Výhody hierarchických datových struktur

- obě souřadnice lze jednoduše adresovat (2D lokalizace je určena jedním číslem)
- každý čtvereční metr Země má konzistentní 21 cifernou adresu
- z délky adresy lze automaticky odvodit rozlišení

6.4 Porovnání uložení dat v souborech, hierarchických strukturách a relačních databázích

Struktura	Výhody	Nevýhody
Soubor	<ul style="list-style-type: none">- rychlý přístup k datům pokud mají klíč- jednoduchá struktura a jednoduché programování	<ul style="list-style-type: none">- aplikace je závislá na datech- změna datového vyžaduje přeprogramování- pomalý přístup k datům pokud nemají klíč
Hierarchie	<ul style="list-style-type: none">- přidávání a rušení vět je snadné- rychlý přístup k datům na vyšších úrovních hierarchie	<ul style="list-style-type: none">- přístup k datům pouze pomocí cesty složené z pointrů- každá asociace vyžaduje zopakování dat v jiných větvích- vysoké nároky na paměť počítače
Relace	<ul style="list-style-type: none">- snadný přístup i pro laiky- flexibilita k neočekávaným požadavkům- snadná modifikace a přidávání nových dat, záznamů a vazeb- lze změnit fyzické uložení dat bez ztráty vazeb	<ul style="list-style-type: none">- sekvenční přístup je pomalý- metoda uložení dat způsobuje vzrůst času při zpracování dat- vzhledem ke flexibilitě vazeb lze dělat logické chyby

7. Zkušební otázky

1. Jaké strategie uchování prostorových a popisných dat. Porovnej rozdíly mezi nimi.
2. Jaké jsou výhody a nevýhody uchování prostorových dat v databázi?
3. Modifikuj koncepci kvartérního stromu tak, aby sloužil pro uchování dat o výškách terénu. Jaké jsou výhody a nevýhody tohoto přístupu proti jednoduchému rastru?

Připomínky a dotazy k obsahu lekce posílej, prosím, na adresu:
Rudolf Richter, richter@fi.muni.cz