



Sistemi za analizu podataka

*On-line Analytical Processing
(OLAP)*

Sadržaj

- Osnovne karakteristike OLAP sistema
- Arhitektura OLAP sistema
- Dimenzione ROLAP strukture
- Osnovni OLAP upiti
- Agregacione funkcije
- Unakrsno tabeliranje
- Analitičke funkcije i upiti

Osnovne karakteristike OLAP sistema

- **On-Line Analytical Processing (OLAP)**
 - Neposredna analitička obrada podataka
 - **Multidimenziona analiza podataka**
 - "Rotiranje" i "detaljizacija" u prikazu podataka, do zahtevanog nivoa
 - Kreiranje i interaktivno istraživanje agregiranih podataka, nastalih od velikih količina operativnih podataka
 - Sprovođenje analiza trendova i izuzetaka
 - Izvršavanje analitičkih modela
 - modeli predviđanja
 - modeli regresione analize



Osnovne karakteristike OLAP sistema

- **On-Line Analytical Processing (OLAP)**
 - OLAP transakcije predstavljaju, pretežno, **ad hoc upite**, takve da
 - mogu da obuhvate i **nekoliko hiljada torki** operativnih podataka
 - zahtevaju **spajanje torki i primenu funkcija agregiracije podataka**, nad velikim skupovima podataka
 - poželjno je da se OLAP transakcije **izvode nad DW bazom** podataka
 - upotreba OLTP sistema u svrhe podrške OLAP-a, može rezultovati u **neprihvatljivo lošim performansama** realizacije upita



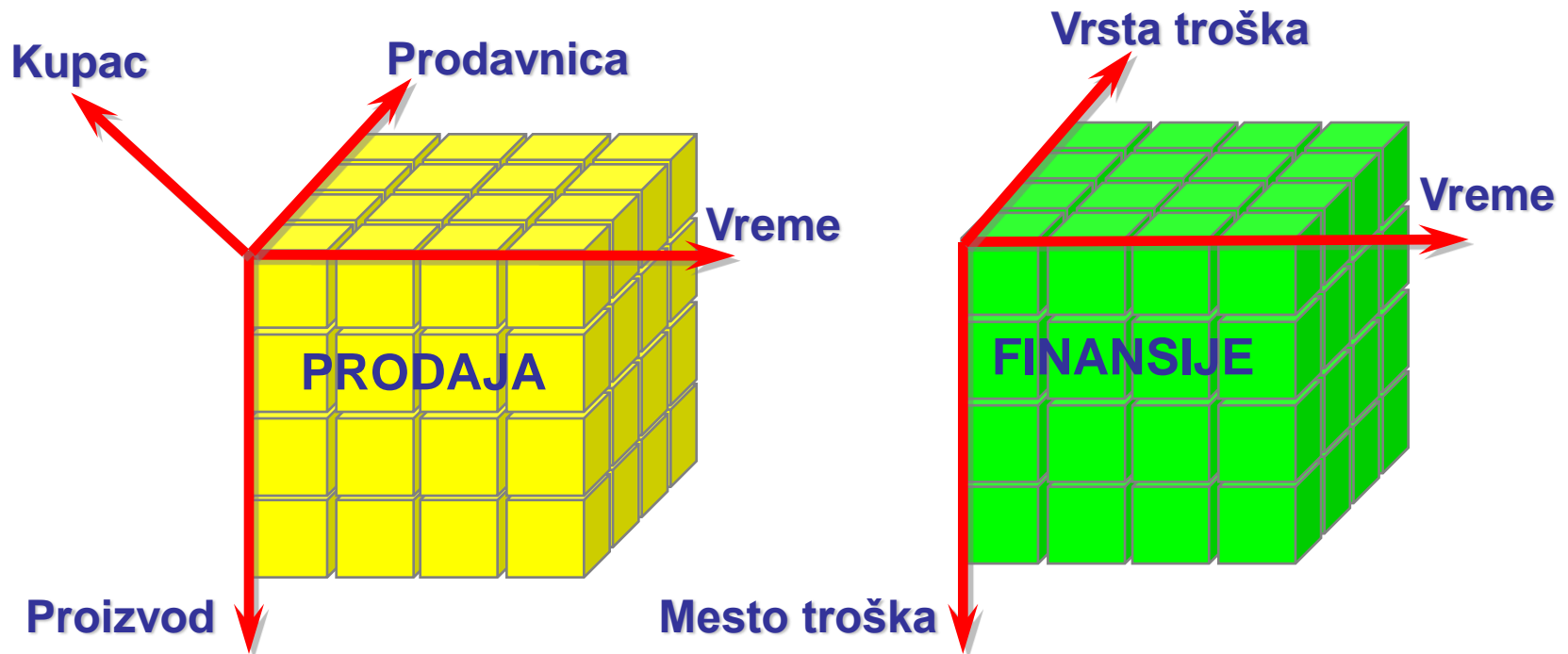
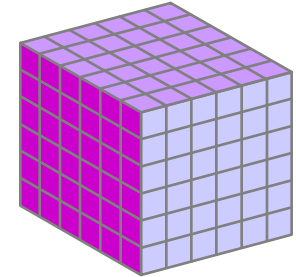
Osnovne karakteristike OLAP sistema

- **OLAP – osnovne karakteristike**
 - multidimenzione strukture podataka
 - primena multidimenzionih tehnika analize podataka
 - repozitorijum meta podataka
 - UI, prilagođen znanju, potrebama i mogućnostima korisnika – "intuitivno" korišćenje i fleksibilno izveštavanje
 - primena višeslojnih i distribuiranih K/S arhitektura

Osnovne karakteristike OLAP sistema

- OLAP – multidimenzionost**

- parametri modela – dimenzije
- podaci se nalaze u preseku dimenzija





Primer:
multidimenzionost OLAP struktura

Osnovne karakteristike OLAP sistema

Račun

<i>RačBR</i>	<i>KupačID</i>	<i>Datum</i>	<i>Ukupno</i>
2034	201	15. 12. 01.	3500
2035	202	15. 12. 01.	1800
2036	201	16. 12. 01.	2000
2037	202	16. 12. 01.	1800

Stavka

<i>RačBR</i>	<i>Proizvod</i>	<i>JedCena</i>	<i>Količina</i>
2034	miš	150	20
2034	flash	20	25
2035	miš	150	12
2036	flash	20	100
2037	flash	20	25
2037	miš	150	2



Primer:
multidimenzionost OLAP struktura

Osnovne karakteristike OLAP sistema

Kupac / Vreme

	15. 12. 01.	16. 12. 01.	Σ
201	3500	2000	5500
202	1800	800	2600
Σ	5300	2800	8100

Činjenice – mere
poslovnih
aktivnosti

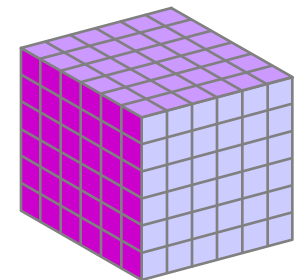
Kupac / Proizvod

	201	202	Σ
miš	20	14	34
flash	125	25	150
Σ	145	39	184

Agregirani
podaci

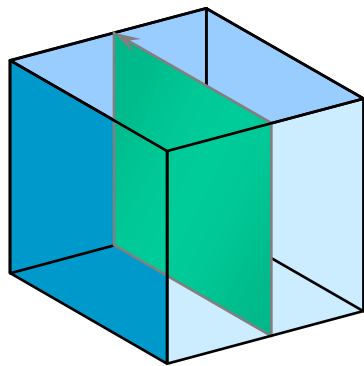
Osnovne karakteristike OLAP sistema

- **OLAP – multidimenzionost**
 - karakteristika svakog OLAP sistema
 - bez obzira na primenjenu arhitekturu i tehnološka rešenja
 - poželjne osobine
 - tehnička mogućnost kreiranja neograničenog broja dimenzija i nivoa agregacije podataka
 - tehnička mogućnost primene neograničenog broja analitičkih funkcija nad multidimenzionom strukturom
 - mogućnost dinamičkog upravljanja retkim matricama
 - fleksibilnost u oblikovanju različitih dimenzionih struktura ("generičnost")



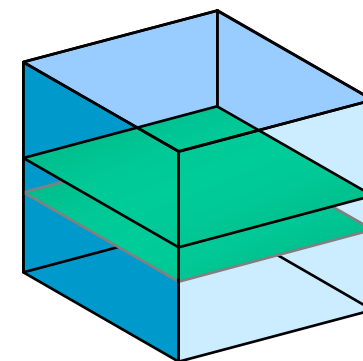
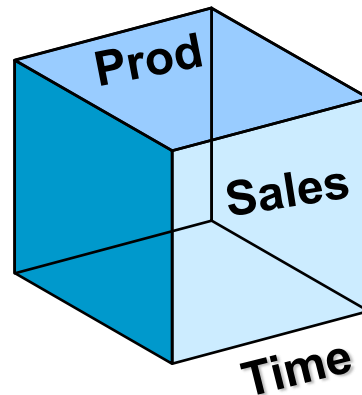
Osnovne karakteristike OLAP sistema

- OLAP – multidimenzionost**

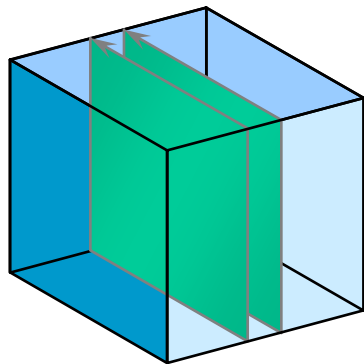


Product mgr. view

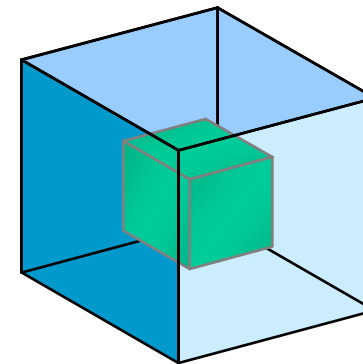
Market



Regional mgr. view



Financial mgr. view



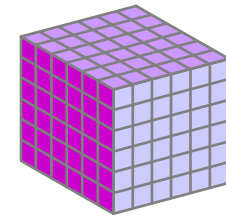
Ad hoc view

Osnovne karakteristike OLAP sistema

- **Opšti pristupi u realizaciji multidimenzionosti**

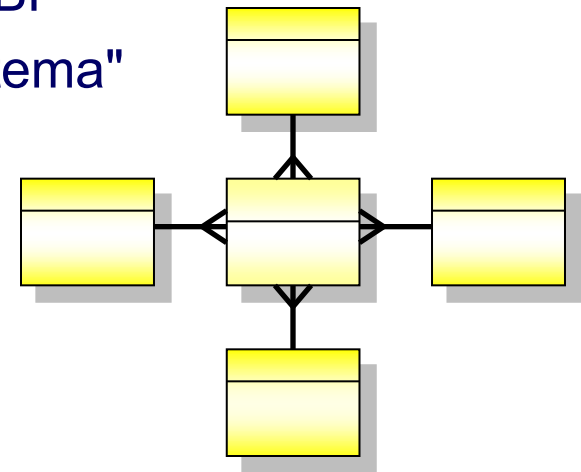
- **putem hiperkocke**

- specijalizovana multidimenzionalna struktura podataka



- **putem koncepta relacionog modela podataka**

- OLAP struktura = zvezdasta šema DW BP
- videti temu "Strukture šeme BP DW sistema"

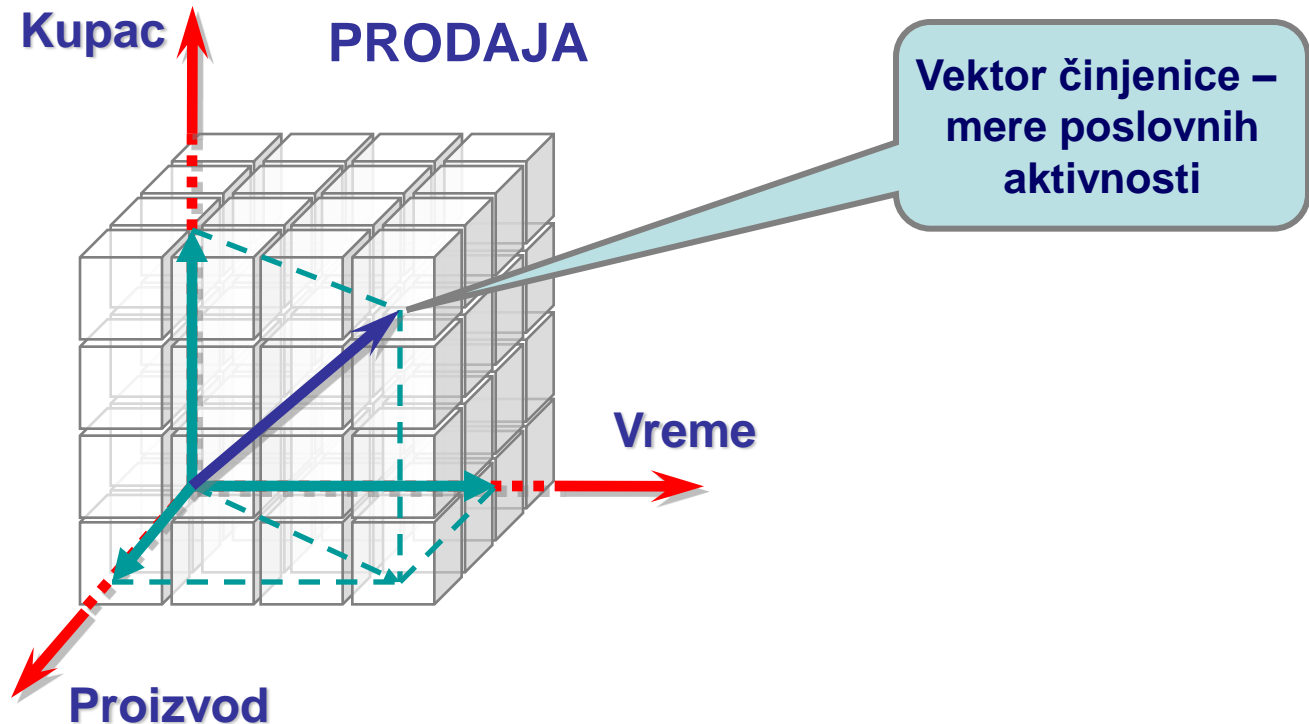


Osnovne karakteristike OLAP sistema

- **Hiperkocka (kocka podataka, Data Cube)**
 - hiperkocka – kocka u n-dimenzionalnom prostoru, $n \in \mathbb{N}$
 - svaka dimenzija podataka pridružuje se jednoj hiperivici hiperkocke
 - mere činjenica su zabeležene kao vrednosti u hiperprostoru
 - svaka tačka hiperprostora hiperkocke je opredeljena
 - jednom kombinacijom vrednosti svih dimenzija, tj.
 - koordinatama odgovarajućeg vektora u hiperprostoru
 - dimenzijama se pridružuju **unapred izračunati, agregirani ("sumarni") podaci**

Osnovne karakteristike OLAP sistema

- Hiperkocka (kocka podataka, Data Cube)



Osnovne karakteristike OLAP sistema

- **Hiperkocka (kocka podataka, Data Cube)**
 - zasnovana na tzv. hypercube organizaciji podataka
 - fizička struktura podataka
 - višedimenzionalni nizovi
 - "keširani" u operativnoj memoriji
 - fizički nezavisni od servera BP
 - tehnike izgradnje hiperkocke
 - nestandardne
 - zavisne od proizvođača OLAP sistema
 - po pravilu, generisanje hiperkocke traje dugo, ali se upiti brzo izvršavaju

Komparacija: načini realizacije multidimenzionalnosti

Osnovne karakteristike OLAP sistema

KARAKTERISTIKE	RELACIONI PRISTUP	PRISTUP ZASNOVAN NA HIPERKOCKI
ORGANIZACIJA PODATAKA	<i>Zvezdasta šema (standardno)</i>	<i>Hiperkocka (nestandardno)</i>
FLEKSIBILNOST	<i>Visoka (dozvoljava neograničen broj dimenzija)</i>	<i>Niska (dodavanje nove dimenzije traži reorganizaciju hiperkocke)</i>
OBIM PODATAKA	<i>Srednji do veliki</i>	<i>Mali do srednji</i>
PERFORMANSE S OBZIROM NA OBIM PODATAKA	<i>Dobar za mali obim Prihvatljiv za srednji i veliki obim</i>	<i>Odličan za mali i srednji obim Loš za veliki obim</i>

Sadržaj

- Osnovne karakteristike OLAP sistema
- Arhitektura OLAP sistema
- Dimenzione ROLAP strukture
- Osnovni OLAP upiti
- Agregacione funkcije
- Unakrsno tabeliranje
- Analitičke funkcije i upiti

Arhitektura OLAP sistema

- **Opšta OLAP arhitektura**
 - **OLAP baza podataka (OLAP server)**
 - DW baza podataka
 - OLTP baza podataka
 - sopstveni alati za izgradnju i eksploataciju struktura hiperkocke
 - **OLAP "front-end" komponenta (OLAP klijent)**
 - podrška multidimenzionalne analize i prikaza podataka

Arhitektura OLAP sistema

- **OLAP server - mogućnosti**
 - tradicionalni RDBMS
 - specijalizovani SQL server
 - ROLAP server
 - MOLAP server
 - HOLAP server

Arhitektura OLAP sistema

- **Tradicionalni RDBMS**
 - nisu primarno namenjeni da efikasno podrže OLAP upite i baze podataka obima GByte - TByte
 - ne poseduju analitičke funkcije
 - mogu biti, u određenoj meri, iskorišćeni u funkciji podrške OLAP sistema

Arhitektura OLAP sistema

- **Specijalizovani SQL server**
 - Cilj: bolja podrška OLAP upita nad relaciono orijentisanom multidimenzionom strukturom podataka
 - **proširenja SQL-a novim tipovima funkcija**
 - SQL operatori ROLLUP, CUBE, WINDOW
 - **unapređenje koncepta fizičke organizacije podataka**
 - novi tipovi indeksnih struktura i fizičko particioniranje
 - **unapređenje SQL optimizatora upita**
 - mogućnost zadavanja direktiva optimizatoru upita (OPTIMIZE)
 - paralelizacija izvršenja SELECT naredbe
 - unapređena logika korišćenja indeksnih struktura

Arhitektura OLAP sistema

- **ROLAP server (Relational OLAP Engine)**
 - međusloj između "pozadinske" relacione BP (DW ili OLTP) i OLAP klijenta
 - **transformiše i optimizuje SQL OLAP upite za izvršenje nad pozadinskom BP**
 - identifikuje poglede koje treba realizovati kao materijalizovane poglede
 - transformiše SQL upite tako da koriste materijalizovane poglede
 - transformiše SQL upite tako da efikasnije agregiraju tražene podatke
 - obezbeđuje transformaciju pogleda na strukturu zvezdaste šeme u pogled na strukturu hiperkocke
 - dimenzione ROLAP strukture

Arhitektura OLAP sistema

- **MOLAP server**
(Multidimensional OLAP Engine)
 - međusloj između "pozadinske" relacije BP (DW ili OLTP) i OLAP klijenta
 - **direktna podrška multidimenzionalnih struktura tipa hiperkocke**
 - upotreba višedimenzionalnih nizova u cilju implementacije hiperkocki
 - podaci u hiperkockama su "keširani" i nezavisni od pozadinske BP
 - OLAP upiti se direktno izvršavaju nad hiperkockom

Arhitektura OLAP sistema

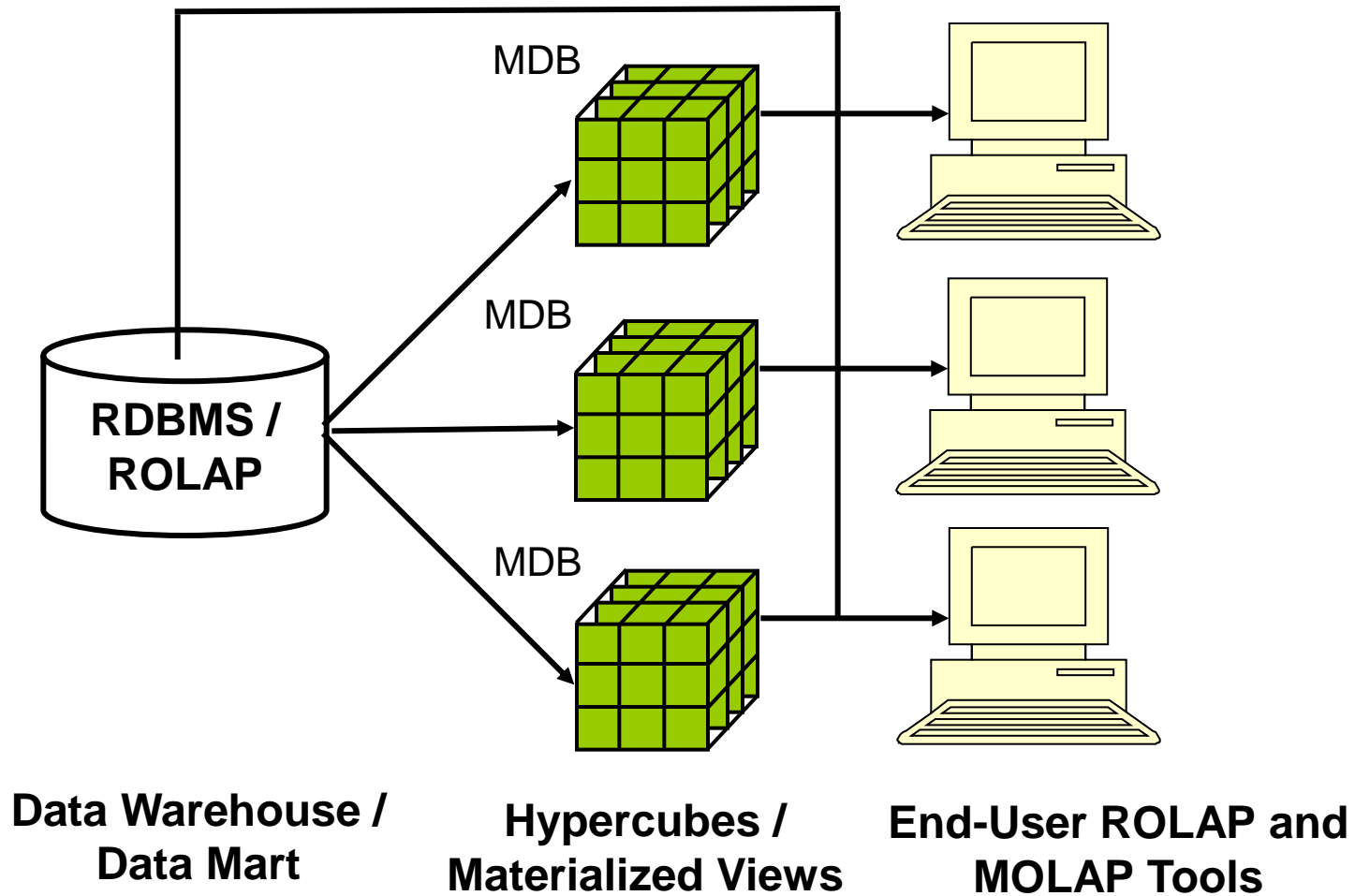
- **MOLAP server**
(Multidimensional OLAP Engine)
 - specijalizovani server za efikasnu podršku
 - realizacije upita koji sadrže skupovne (sumarne) funkcije i zahtevaju grupisanja podataka
 - izračunavanja složenih logičkih, statističkih i vremenski zavisnih funkcija
 - uobičajeno, MOLAP serveri sadrže "bogate" biblioteke specijalizovanih funkcija
 - po pravilu, izgradnja i inicijalizacija struktura tipa hiperkocke je vremenski zahtevna
 - efikasna realizacija (brzo izvršenje) OLAP upita nad pripremljenom hiperkockom

Arhitektura OLAP sistema

- **HOLAP server (Hybrid OLAP Engine)**
 - kombinuje dobra svojstva MOLAP i ROLAP servera
 - pozadinska BP: ROLAP server i DW BP
 - agregirani podaci: MOLAP server i hiperkočke
 - kompromis
 - ROLAP omogućava efikasnu podršku rada s velikim količinama podataka
 - MOLAP omogućava efikasne OLAP upite nad "izvodima" podataka manjeg obima, strukturiranim u obliku hiperkočke

Arhitektura OLAP sistema

- **HOLAP server (Hybrid OLAP Engine)**



Sadržaj

- Osnovne karakteristike OLAP sistema
- Arhitektura OLAP sistema
- Dimenzione ROLAP strukture
- Osnovni OLAP upiti
- Agregacione funkcije
- Unakrsno tabeliranje
- Analitičke funkcije i upiti

Dimenzione ROLAP strukture

- Osnovna pretpostavka
 - dimenzije se organizuju hijerarhijski
 - analize hijerarhijski organizovanih podataka
 - roll-up - kretanje prema vrhu hijerarhije
 - drill-down - kretanje prema dnu hijerarhije
 - najviši hijerarhijski nivo je tzv. "ALL level"
 - npr. ALL years, ALL countries, ALL employees, itd.
 - analitičar podataka je tipično odgovoran za primenu hijerarhijskih analiza podataka
- Hijerarhije dimenzija mogu se upotrebiti u optimizaciji upita
 - zasnovanoj na query rewrite tehnici
 - u kombinaciji s materijalizovanim pogledima

Dimenzione ROLAP strukture

- Tipično za analize podataka
 - primena drill-down analiza od izabranog nivoa hijerarhije
 - prvo se posmatraju podaci višeg nivoa agregacije pa tek onda nižeg
- Tipično za query rewrite tehniku
 - upotreba materijalizovanog pogleda
 - selekcija podataka najniže zahtevanog nivoa granularnosti
 - agregacija podataka na višim hijerarhijskim nivoima
 - suprotna logika pripreme podataka od one koju zahteva drill-down analiza

Dimenzione ROLAP strukture

- ROLAP serveri podržavaju specijalizovane koncepte za dimenzije i hijerarhije
 - posebne naredbe (sintaksa) za kreiranje dimenzija i hijerarhija
- Motivacija
 - da procesor upita može automatski primeniti query rewrite tehniku optimizacije upita, jer
 - eksplicitno prepoznaje kreirane hijerarhije
 - garantovano je očuvanje određenih ograničenja u hijerarhiji
 - u protivnom, takva optimizacija upita, ne bi mogla garantovati pouzdane rezultate

Dimenzione ROLAP strukture

- Naredbe za kreiranje hijerarhijskih struktura
 - kreiranje dimenzija, atributa i hijerarhijskih nivoa
 - deklarisanje veza između atributa na različitim hijerarhijskim nivoima
 - deklarisanje veza između atributa na istim hijerarhijskim nivoima
- Preduslov
 - već postoji kreirana dimenziona tabela u šemi DW BP
- Efekti
 - ROLAP server prepoznaje kreirane hijerarhije
 - obezbeđenje provere važenja ograničenja koja su ugrađena u deklaracije hijerarhija

Dimenzione ROLAP strukture

- Primer
 - 2NF dimenziona tabela u zvezdastoj šemi

```
CREATE TABLE Location  
(  
  ShopId Number(8)    PRIMARY KEY,  
  City Varchar(20)    NOT NULL,  
  District Varchar(20) NOT NULL,  
  Country Varchar(20) NOT NULL,  
  Mayor Varchar(20),  
  President Varchar(20)  
);
```

Dimenzione ROLAP strukture

- **Kreiranje dimenzije**

```
CREATE DIMENSION <dimension_name>
```

```
LEVEL <column_name> IS <table_name.column_name>...
```

- **Primer**

- kreiranje dimenzije nad tabelom Location

```
CREATE DIMENSION Location_DIM
```

```
LEVEL ShopId IS Location.ShopId
```

```
LEVEL City IS Location.City
```

```
LEVEL District IS Location.District
```

```
LEVEL Country IS Location.Country
```


Dimenzione ROLAP strukture

- **Klauzula HIERARCHY**
 - definiše veze tipa N : 1 između nivoa u naredbi CREATE DIMENSION
 - ukoliko je atribut na N strani NOT NULL, veza definiše istovremeno i odgovarajuću funkcionalnu zavisnost
- svaka CREATE DIMENSION naredba može sadržati više HIERARCHY klauzula
 - u slučaju višestrukih hijerarhija nad istom dimenzionom tabelom

Dimenzione ROLAP strukture

- **Klauzula HIERARCHY**

HIERARCHY <hierarchy_name> (
 <column_name1> **CHILD OF** <column_name2> ...)

- **Primer**

– deklarisanje hijerarhije nad dimenzijom Location_DIM

```
HIERARCHY location_rollup (  
    ShopId           CHILD OF  
    City             CHILD OF  
    District         CHILD OF  
    Country)
```

Dimenzione ROLAP strukture

- **Klauzula ATTRIBUTE ... DETERMINES**
 - definiše veze tipa N : 1 ili 1 : 1, odnosno funkcionalne zavisnosti, između LEVEL atributa i drugih atributa dimenzione tabele koji pripadaju istom hijerarhijskom nivou
 - ti atributi funkcionalno zavise od odgovarajućeg LEVEL atributa
 - upotrebljava se u naredbi CREATE DIMENSION, kao poslednja klauzula

Dimenzione ROLAP strukture

- **Klauzula ATTRIBUTE ... DETERMINES**

ATTRIBUTE <column_name1> **DETERMINES**

<column_name2>...

- **Primer**

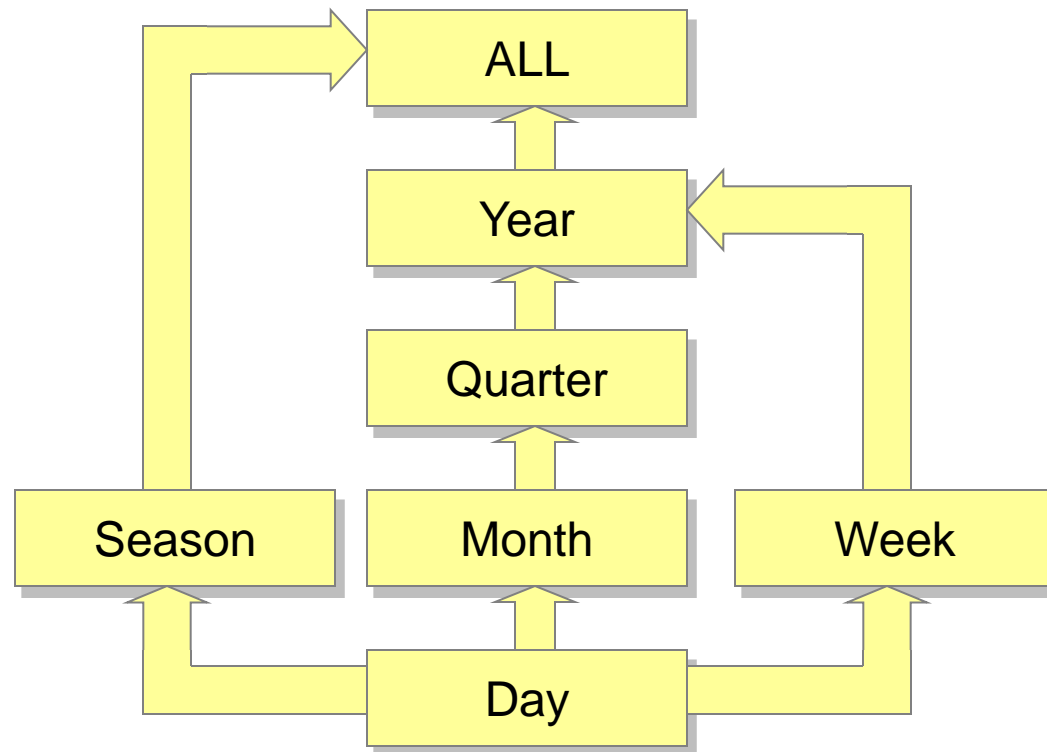
– deklarisanje veza između atributa nad dimenzijom
Location_DIM

ATTRIBUTE City **DETERMINES** Location.Mayor

ATTRIBUTE Country **DETERMINES** Location.President

Dimenzione ROLAP strukture

- Primer
 - višestruka vremenska hijerarhija



Dimenzione ROLAP strukture

- Primer
 - 2NF dimenziona tabela Time u zvezdastoj šemi

```
CREATE TABLE Time (  
    CurrentDate Date      PRIMARY KEY,  
    Month Number(2)      NOT NULL,  
    Quarter Number(1)    NOT NULL,  
    Year Number(4)       NOT NULL,  
    Season Number(1)     NOT NULL,  
    Week Number(2)       NOT NULL,  
    DayOfWeek Varchar (10),  
    MonthName Varchar (10))
```

Dimenzione ROLAP strukture

- Primer
 - kreiranje dimenzije i hijerarhije nad tabelom Time

CREATE DIMENSION Time_DIM

LEVEL CurrentDate **IS** Time.CurrentDate

LEVEL Month **IS** Time.Month

LEVEL Quarter **IS** Time.Quarter

LEVEL Year **IS** Time.Year

LEVEL Season **IS** Time.Season

LEVEL Week **IS** Time.Week

HIERARCHY weekly_rollup (

CurrentDate **CHILD OF**

Week **CHILD OF**

Year)

Dimenzione ROLAP strukture

- Primer

- kreiranje dimenzije i hijerarhije nad tabelom Time

...

HIERARCHY seasonal_rollup (
 CurrentDate **CHILD OF**
 Season)

HIERARCHY calendar_rollup (
 CurrentDate **CHILD OF**
 Month **CHILD OF**
 Quarter **CHILD OF**
 Year)

ATTRIBUTE CurrentDate **DETERMINES** Time.DayOfWeek

ATTRIBUTE Month **DETERMINES** Time.MonthName

Dimenzione ROLAP strukture

- Primer

- 3NF dimenzione tabele u šemi tipa pahuljice

```
CREATE TABLE Product (  
    ProductId Number(6) PRIMARY KEY,  
    ProductName Varchar(20),  
    TypeId Number(4) NOT NULL REFERENCES ProdType)
```

```
CREATE TABLE ProdType (  
    TypeId Number(4) PRIMARY KEY,  
    TypeDesc Varchar(30),  
    IndustryId Number(4) NOT NULL REFERENCES Industry)
```

```
CREATE TABLE Industry (  
    IndustryId Number(4) PRIMARY KEY,  
    IndustryName Varchar(40))
```

Dimenzione ROLAP strukture

- Primer
 - kreiranje dimenzije i hijerarhije nad više tabela

CREATE DIMENSION Product_DIM

LEVEL ProductId **IS** Product.ProductId

LEVEL Typeld **IS** ProdType.Typeld

LEVEL IndustryId **IS** Industry.IndustryId

HIERARCHY merchandise_rollup (

ProductId **CHILD OF**

Typeld **CHILD OF**

IndustryId

JOIN KEY Product.Typeld **REFERENCES** Typeld

JOIN KEY ProdType.IndustryId **REFERENCES**

IndustryId)

Dimenzione ROLAP strukture

- Primer

- kreiranje dimenzije i hijerarhije nad više tabela

...

ATTRIBUTE ProductId **DETERMINES** Product.ProductName

ATTRIBUTE TypeId **DETERMINES** ProdType.TypeDesc

ATTRIBUTE IndustryId **DETERMINES** Industry.IndustryName

Dimenzione ROLAP strukture

- Verodostojnost podataka
 - u kontekstu kreiranih dimenzija koje će koristiti optimizator upita
 - zavisi od zadovoljenja sledećih ograničenja
 - primarnih ključeva tabele činjenica i dimenzionih tabela
 - ograničenja referencijalnih integriteta
 - između tabele činjenica i tabela dimenzija
 - između samih tabela dimenzija u šemi tipa pahuljice
 - ograničenja zabrane nula vrednosti i
 - funkcionalnih zavisnosti između atributa u hijerarhiji

Dimenzione ROLAP strukture

- On-line provera ograničenja na nivou DW BP
 - po pravilu, onemogućava se
 - jer je vremenski vrlo zahtevna
 - obezbeđenje zadovoljenja ograničenja poverava se ETL procesu
 - ETL procedure treba da garantuju validnost podataka
 - u određenim situacijama, poželjna odložena, periodična provera svih ograničenja u DW BP
 - putem specijalnih batch programa koji se pokreću u periodu van radnog vremena sistema

Sadržaj

- Osnovne karakteristike OLAP sistema
- Arhitektura OLAP sistema
- Dimenzione ROLAP strukture
- Osnovni OLAP upiti
- Agregacione funkcije
- Unakrsno tabeliranje
- Analitičke funkcije i upiti

Osnovni OLAP upiti

- **OLAP klijent**

- podržava različite vrste upita – operacija nad OLAP serverom
 - grupisanja i agregacije podataka
 - agregacije i detaljizacije
 - isecanja
 - primenu različitih vrsta agregacionih funkcija
 - rangiranja
 - unakrsna tabeliranja

Osnovni OLAP upiti

- **OLAP upiti**
 - veliki broj OLAP upita može biti iskazan putem standardne SQL SELECT naredbe
 - osnovnih
 - grupisanja i agregacije podataka
 - agregacije i detaljizacije
 - isecanja
 - složenijih
 - rangiranja
 - unakrsna tabeliranja
 - najčešće, u upotrebi
 - SQL klauzule GROUP BY i HAVING i
 - SQL agregacione (skupovne) funkcije
 - postoje i specijalizovane SQL SELECT klauzule

Osnovni OLAP upiti

- **Izabrani osnovni OLAP upiti**
 - agregacije podataka sa skupovnim funkcijama
 - grupisanja i agregacije podataka
 - agregacije (roll-up) i detaljizacije (drill-down)
 - isecanja (slice i dice)

Primer:

vrste OLAP upita i operacija

Osnovni OLAP upiti

Račun

<i>RačBR</i>	<i>KupačID</i>	<i>Datum</i>	<i>Ukupno</i>
2034	201	15. 12. 01.	3500
2035	202	15. 12. 01.	1800
2036	201	16. 12. 01.	2000
2037	202	16. 12. 01.	1800

Stavka

<i>RačBR</i>	<i>Proizvod</i>	<i>JedCena</i>	<i>Količina</i>
2034	miš	150	20
2034	flash	20	25
2035	miš	150	12
2036	flash	20	100
2037	flash	20	25
2037	miš	150	2

Osnovni OLAP upiti

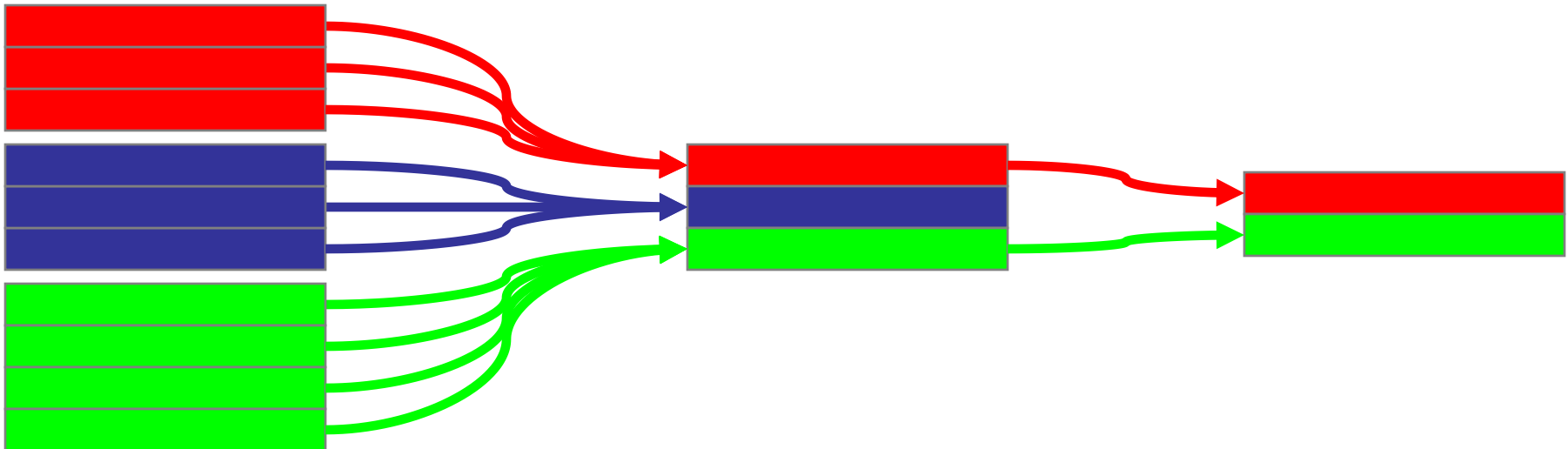
- **Agregacije podataka sa skupovnim funkcijama**
 - SQL SELECT podržava osnovne agregacione funkcije
 - COUNT, SUM, AVG, STDEV, MIN, MAX
 - Primeri

```
SELECT COUNT(*) AS NoOfItems  
FROM Stavka WHERE RacBR LIKE '203%'
```

```
SELECT COUNT(DISTINCT Proizvod) AS NoOfProiz  
FROM Stavka WHERE Proizvod LIKE 'f%';
```

Osnovni OLAP upiti

- **Grupisanja i agregacije podataka**
 - grupisanje po zadatom kriterijumu jednakosti podataka
 - agregiranje podataka po grupama
 - primena SQL klauzula GROUP BY i HAVING
 - primena SQL agregacionih (skupovnih) funkcija



Osnovni OLAP upiti

- **Grupisanja i agregacije podataka**

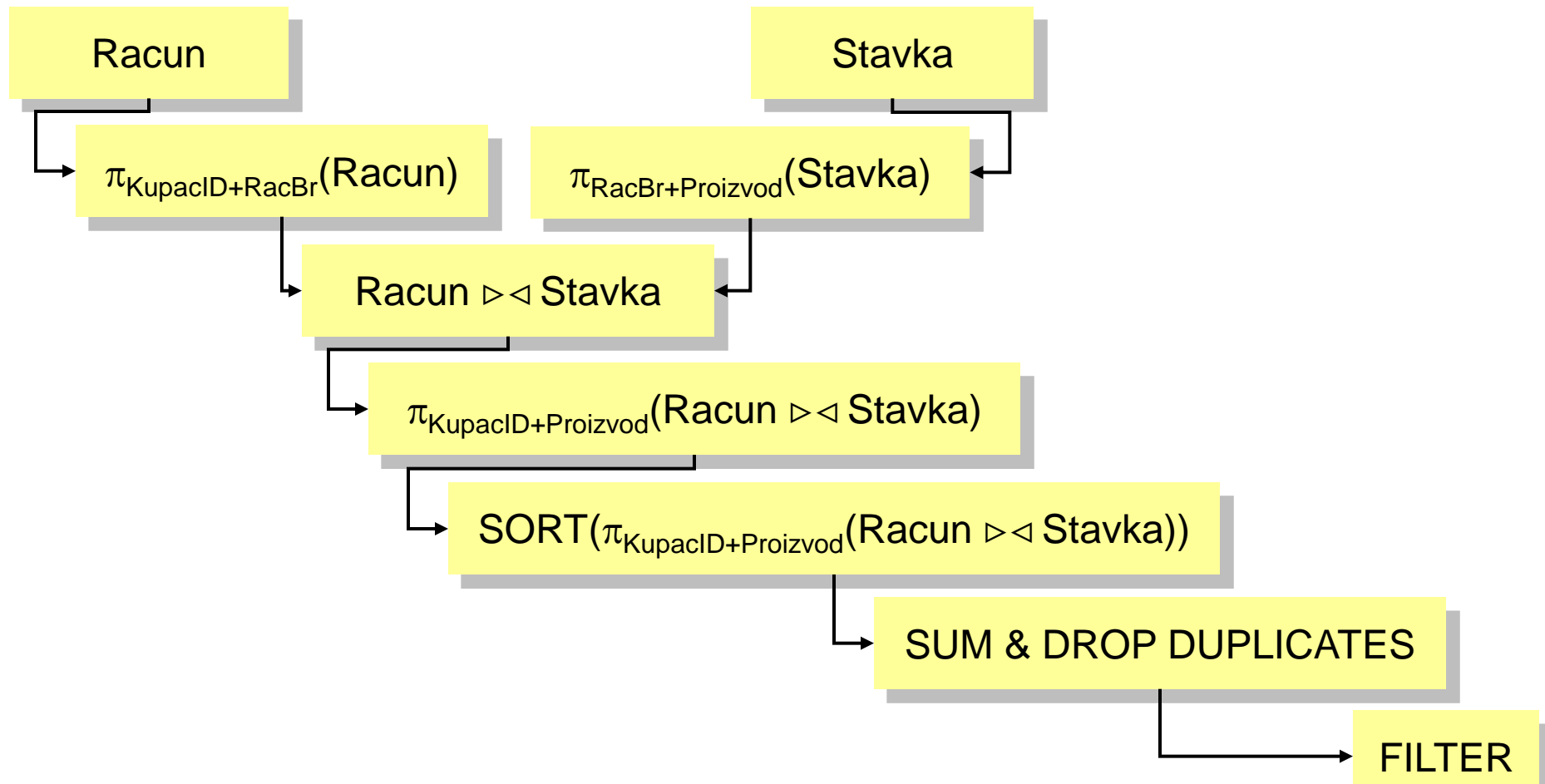
- Primer

- za svaku kombinaciju kupac-proizvod, prikazati ukupnu količinu prodatih artikala i ostvarenu cenu
- prikazati samo kombinacije, za koje je ukupna prodana količina veća ili jednaka 20

```
SELECT r.KupacID, s.Proizvod, SUM(Kolicina) UkKolic,  
                                             SUM(JedCena * Kolicina) UkCena  
FROM Racun r, Stavka s  
WHERE r.RacBr = s.RacBr  
GROUP BY r.KupacID, s.Proizvod  
HAVING SUM(Kolicina) >= 20
```

Osnovni OLAP upiti

- **Grupisanja i agregacije podataka**
 - optimizator upita formira stablo realizacije upita



Primer:

grupisanje i agregacija (SUM)

Osnovni OLAP upiti

Grupisanje i sortiranje

<i>KupacID</i>	<i>Proizvod</i>	<i>Kolicina</i>	<i>Cena</i>
201	flash	25	500
201	flash	100	2000
201	miš	20	3000
202	flash	25	500
202	miš	12	1800
202	miš	2	300

Primer:

grupisanje i agregacija (SUM) osnovni OLAP upiti

Grupisanje i sortiranje

<i>KupacID</i>	<i>Proizvod</i>	<i>Kolicina</i>	<i>Cena</i>
201	flash	25	500
201	flash	100	2000
201	miš	20	3000
202	flash	25	500

<i>KupacID</i>	<i>Proizvod</i>	<i>UkKolič</i>	<i>UkCena</i>	
202	201	flash	125	2500
202	201	miš	20	3000
	202	flash	25	500
	202	miš	14	2100

Sumiranje

Primer:
grupisanje i agregacija (SUM) osnovni OLAP upiti

Grupisanje i sortiranje

<i>KupacID</i>	<i>Proizvod</i>	<i>Kolicina</i>	<i>Cena</i>
201	flash	25	500
201	flash	100	2000
201	miš	20	3000
202	flash	25	500

<i>KupacID</i>	<i>Proizvod</i>	<i>UkKolič</i>	<i>UkCena</i>
201	flash	125	2500
201	miš	20	3000
202	flash	25	500

Sumiranje

<i>KupacID</i>	<i>Proizvod</i>	<i>UkKolič</i>	<i>UkCena</i>
201	flash	125	2500
201	miš	20	3000
202	flash	25	500

Filtriranje grupa

Osnovni OLAP upiti

- **Agregacije i detaljizacije podataka**
 - temelje se na dimenzijama i njihovim hijerarhijama
 - **Agregacija – Roll-Up**
 - redukovanje dimenzija hiperkocke odbacivanjem dimenzije i agregacijom podataka
 - "odbacuje" se dimenzija najnižeg nivoa
 - vrši se sumiranje podataka po odbačenoj dimenziji
 - operacija se može tranzitivno primeniti, do odbacivanja poslednje preostale dimenzije

Osnovni OLAP upiti

- **Agregacije i detaljizacije podataka**
 - temelje se na dimenzijama i njihovim hijerarhijama
 - **Detaljizacija – Drill-Down**
 - "simetrična" operacija za Roll Up
 - vraća se "odbačena" dimenzija nižeg nivoa
 - obezbeđuje se detaljni prikaz podataka po vraćenoj dimenziji
 - "stavka", na sledećem nivou, postaje "zaglavlje" za stavke višeg nivoa detaljnosti
 - operacija se može tranzitivno primeniti, do povratka poslednje preostale dimenzije

Osnovni OLAP upiti

- **Tehnike Roll-Up-a / Drill-Down-a**
 - **Dimenzioni Roll-Up (alternativno Drill-Down)**
 - odbacivanjem cele dimenzije u zvezdastoj šemi
 - sprovodi se isključivo nad tabelom činjenica
 - odbacuje se strani ključ dimenzije i po njemu se vrši agregacija
 - **Hijerarhijski Roll-Up (alternativno Drill-Down)**
 - izborom višeg hijerarhijskog nivoa dimenzije
 - sprovodi se nad tabelom činjenica i izabranom tabelom dimenzije
 - vrši se agregacija po izabranom nivou hijerarhije i atributu dimenzije
 - agregacija iznad najvišeg nivoa hijerarhije (ALL) svodi hijerarhijski roll-up na dimenzioni roll-up
 - **Kombinovani Roll-Up (alternativno Drill-Down)**
 - kombinacija prethodna dva postupka

Osnovni OLAP upiti

- **Dimenzioni Roll-Up**

- Primer

- Posmatra se materijalizovani pogled – tabela činjenica

```
CREATE MATERIALIZED VIEW KupacProiz AS
```

```
SELECT r.KupacID, s.Proizvod, SUM(Kolicina) UkKolic,
```

```
                SUM(JedCena * Kolicina) UkCena
```

```
FROM Racun r, Stavka s
```

```
WHERE r.RacBr = s.RacBr
```

```
GROUP BY r.KupacID, s.Proizvod
```

```
HAVING SUM(Kolicina) >= 20
```

<i>KupacID</i>	<i>Proizvod</i>	<i>UkKolič</i>	<i>UkCena</i>
<i>201</i>	<i>flash</i>	<i>125</i>	<i>2500</i>
<i>201</i>	<i>miš</i>	<i>20</i>	<i>3000</i>
<i>202</i>	<i>flash</i>	<i>25</i>	<i>500</i>

Osnovni OLAP upiti

- **Dimenzioni Roll-Up**

- Primer

- odbacivanje dimenzije Proizvod
- upit
 - prikazati kupce sa ostvarenim ukupnim cenama po svim kupljenim proizvodima

```
SELECT KupacID, SUM(UkCena) UkCenaKup
FROM KupacProiz
GROUP BY KupacID
```

<i>KupacID</i>	<i>UkCenaKup</i>
<i>201</i>	<i>5500</i>
<i>202</i>	<i>500</i>

Osnovni OLAP upiti

- **Dimenzioni Roll-Up**

- Primer

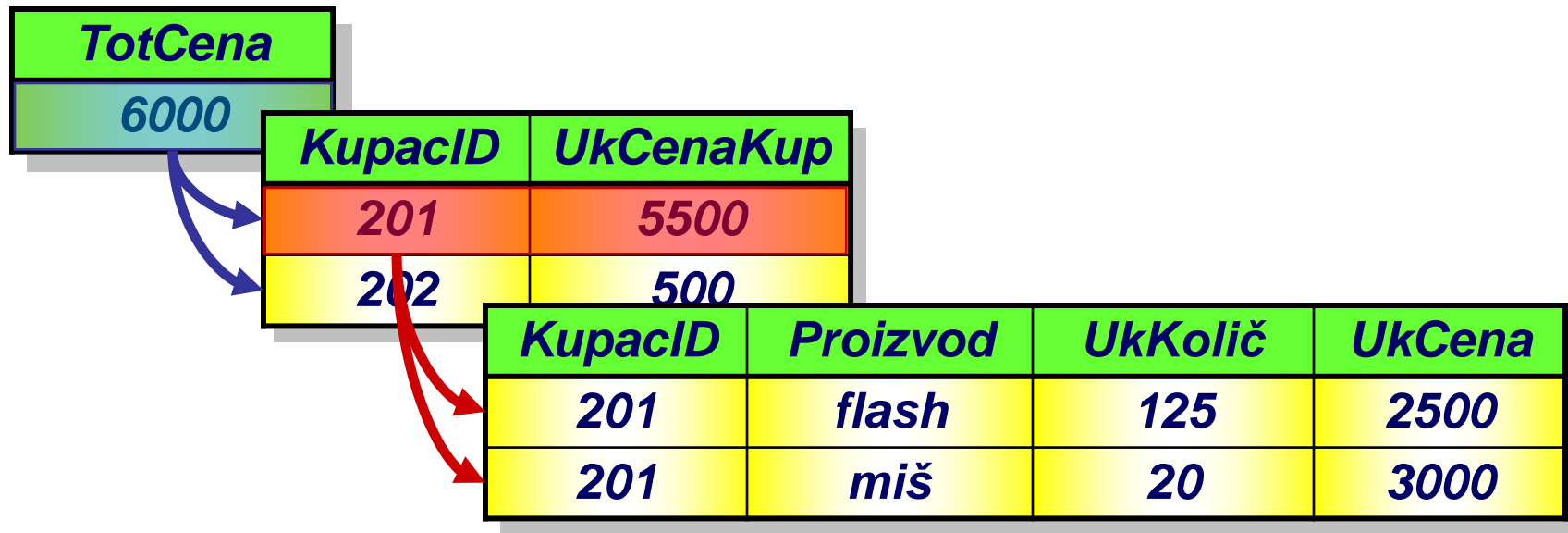
- odbacivanje dimenzije Kupac
- upit
 - prikazati ukupni ostvareni novčani promet po svim kupcima i proizvodima

```
SELECT SUM(UkCena) TotCena  
FROM KupacProiz
```

TotCena
6000

Osnovni OLAP upiti

- **Detaljizacija – Drill-Down**
 - Primer



Osnovni OLAP upiti

- **Hijerarhijski Roll-Up**
 - Primer
 - posmatra se zvezdasta šema
 - tabela činjenica
 - Sales, sa merom Total
 - dimenzione tabele
 - Location(ShopId, City)
 - Product(ProdId, ProdName, Type)
 - Time(TimeId, Week)
 - hijerarhije dimenzija
 - ShopId→City (Location hierarchy)
 - ProdId→Type (Product hierarchy)
 - TimeId→Week (Time hierarchy)

Osnovni OLAP upiti

- **Hijerarhijski Roll-Up**

- Primer

- roll-up odbacivanjem ShopId i agregacijom po danu u dimenziji Location

```
SELECT ProdId, City, TimeId, SUM(Total) AS City_Total  
FROM Sales s, Location l  
WHERE s.ShopId = l.ShopId  
GROUP BY ProdId, City, TimeId
```

- roll-up odbacivanjem ProdId i agregacijom po tipu u dimenziji Product

```
SELECT Type, ShopId, TimeId, SUM(Total) AS Type_Total  
FROM Sales s, Product p  
WHERE s.ProdId = p.ProdId  
GROUP BY Type, ShopId, TimeId
```

Osnovni OLAP upiti

- **Hijerarhijski Roll-Up**

- Primer

- roll-up odbacivanjem ShopId i ProdId i agregacijom po danu i tipu u dimenzijama Location i Product

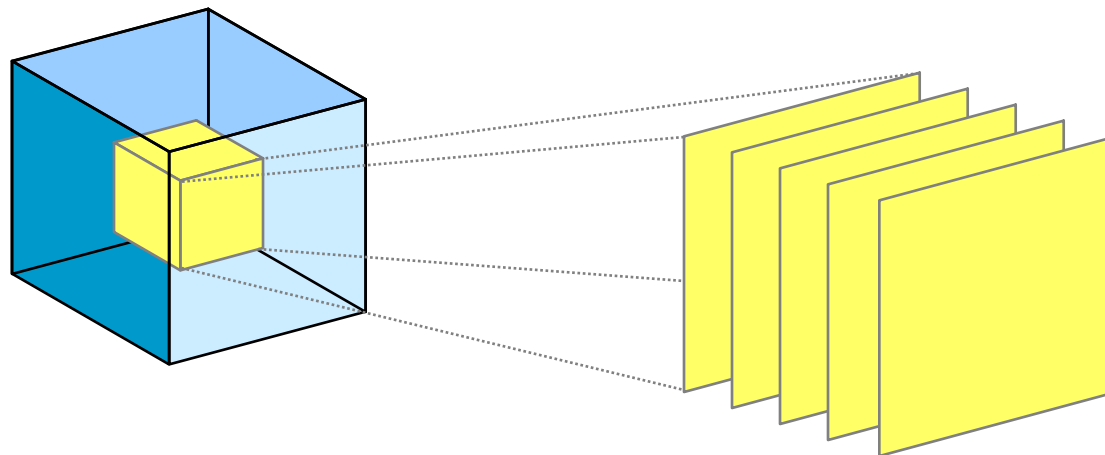
```
SELECT Type, City, TimeId, SUM(Total) AS Type_City_Tot
FROM Sales s, Product p, Location l
WHERE s.ShopId = l.ShopId AND s.ProdId = p.ProdId
GROUP BY Type, City, TimeId
```

- roll-up odbacivanjem nivoa u sve tri dimenzije

```
SELECT Type, City, Week, SUM(Total) AS Type_City_Week_Tot
FROM Sales s, Product p, Location l, Time t
WHERE s.ShopId = l.ShopId AND s.ProdId = p.ProdId
      AND s.TimeId = t.TimeId
GROUP BY Type, City, Week
```

Osnovni OLAP upiti

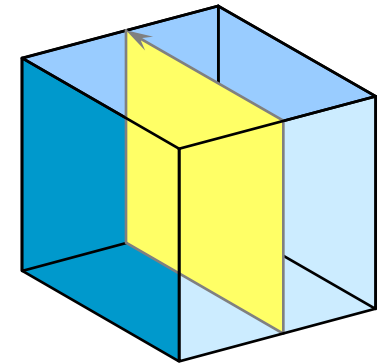
- **Isecanja (Slice & Dice)**
 - obezbeđuju redukovanje (odbacivanje) dimenzija hiperkočke
 - fiksiranjem vrednosti izabranih dimenzija
 - projekcijom vektora činjenica na preostale dimenzije



Osnovni OLAP upiti

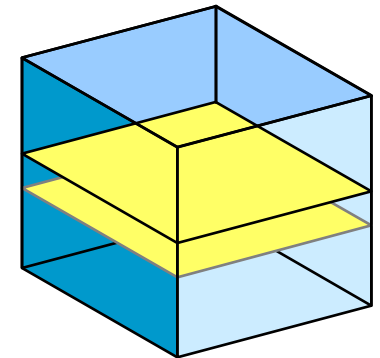
- **Slice**

- za svaku odbačenu dimenziju fiksira se konstantan parametar (uslov jednakosti u WHERE klauzuli)



- **Dice**

- za barem jednu odbačenu dimenziju fiksira se interval mogućih vrednosti
 - za ostale odbačene dimenzije može se fiksirati interval vrednosti ili jedna vrednost



Osnovni OLAP upiti

- **Slice i Dice**

- Primeri

- tabela činjenica: Sales, sa merom Total
- dimenzione tabele: Location(ShopId, City), Product(ProdId, ProdName, Type), Time(TimeId, Week)

```
SELECT ProdId, TimeId, Total  
FROM Sales  
WHERE ShopId = 101
```

```
SELECT ProdId, TimeId, Total  
FROM Sales  
WHERE ShopId BETWEEN 101 AND 200
```

```
SELECT TimeId, Total  
FROM Sales  
WHERE ShopId = 101 AND ProdId = 100101
```

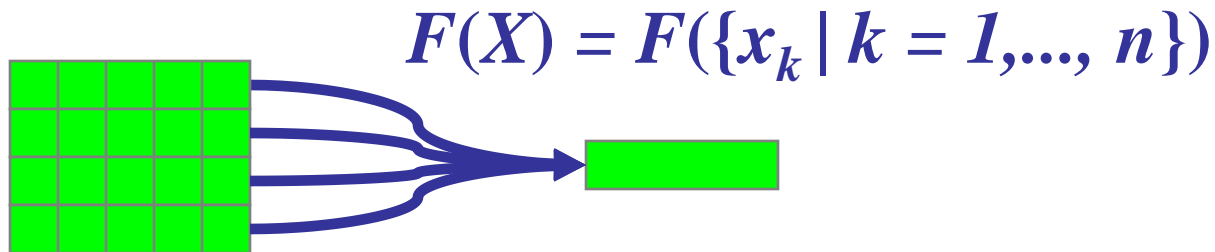
Sadržaj

- Osnovne karakteristike OLAP sistema
- Arhitektura OLAP sistema
- Dimenzione ROLAP strukture
- Osnovni OLAP upiti
- Agregacione funkcije
- Unakrsno tabeliranje
- Analitičke funkcije i upiti

Agregacione funkcije

- **Logika specifikacije**

- argument: skup vrednosti
- rezultat: agregirana vrednost



$$X = [x_k]_{k=1, \dots, n} = [x_{ij}]_{i=1, \dots, m, j=1, \dots, n_i}$$

Agregacione funkcije

- Upotreba u OLAP upitima
 - vrlo intenzivna i nezaobilazna
- Izračunavanje agregacionih funkcija
 - resursno zahtevno
 - zahteva uređivanje (sortiranje) podataka
- Ideja za optimizaciju izračunavanja
 - upotreba agregiranih vrednosti finije granularnosti u izračunavanju agregiranih vrednosti grublje granularnosti
 - posebno interesantna u slučaju roll-up i pivoting operacija
 - nažalost, nije uvek primenljiva
 - zavisi od prirode primenjenih agregacionih funkcija

Agregacione funkcije

- Ideja za optimizaciju izračunavanja
 - posmatra se dvodimenzionalni niz (matrica) vrednosti

$$X = [x_{ij}]_{i=1,\dots,m, j=1,\dots,n_i}$$

- nad kojim treba izračunati agregacionu funkciju

$$F(X) = F(\{x_k \mid k=1,\dots,n\}) = F(\{x_{ij} \mid i=1,\dots,m, j=1,\dots,n_i\})$$

- Koraci ugnježdavanja agregacionih funkcija

- (A) m puta primeniti ugnježdenu agregacionu funkciju nad vrstama matrice X

$$(\forall i \in \{1,\dots,m\})(G(X_i) = G(\{x_{ij} \mid j=1,\dots,n_i\}))$$

- (B) jedanput primeniti globalnu agregacionu funkciju nad dobijenim rezultatima ugnježdene agregacione funkcije iz (A)

$$F(X) = H(\{G(X_i) \mid i=1,\dots,m\})$$

Agregacione funkcije

- **Vrste agregacionih funkcija**

- **distributivne**

- mogu da se izračunavaju postupno, rekurzivnom primenom nad podskupovima podataka

- **algebarske**

- mogu da se izračunavaju postupno, primenom drugih agregacionih funkcija nad podskupovima podataka

- **holističke**

- ne mogu da se izračunavaju postupno, nad podskupovima podataka
 - izračunavaju se uvek "ispočetka", nad celim skupom podataka

Agregacione funkcije

- **Vrste agregacionih funkcija**

- **distributivne**

$$F(X) = H(\{F(\{x_{ij} | j=1, \dots, n_i\}) | i=1, \dots, m\})$$

- **algebarske**

$$F(X) = H(\{G^p(\{x_{ij} | j=1, \dots, n_i\}) | i=1, \dots, m\})$$

- G^p je vektorska funkcija reda p , gde je $p \in \mathbb{N}$ konstanta za F
 - vraća uvek, kao rezultat, p -torku vrednosti
 - predstavlja kombinaciju p (distributivnih) agregacionih funkcija
 - H kombinuje distributivnu primenu funkcija iz G^p i algebarskih operacija

- **holističke**

$$F(X) = F(\{x_{ij} | i=1, \dots, m, j=1, \dots, n_i\})$$

Agregacione funkcije

- **Vrste agregacionih funkcija**
 - **distributivne**
 - **COUNT(), SUM(), MAX(), MIN()**
 - $SUM(\{x_{ij}\}) = SUM(\{SUM(\{x_j\})\}_i)$
 - $COUNT(\{x_{ij}\}) = SUM(\{COUNT(\{x_j\})\}_i)$
 - $MAX(\{x_{ij}\}) = MAX(\{MAX(\{x_j\})\}_i)$
 - **algebarske**
 - **AVG(), STDDEV(), MaxN(), MinN(), Center_of_Mass()**
 - $AVG(\{x_{ij}\}) \neq AVG(\{AVG(\{x_j\})\}_i)$
 - $AVG(\{x_{ij}\}) = SUM(\{SUM(\{x_j\})\}_i) / SUM(\{COUNT(\{x_j\})\}_i)$
 - » $p = 2, G^p = (SUM, COUNT), H = SUM(\{x_j\}) / SUM(\{y_j\})$
 - **holističke**
 - **Median(), MostFrequent() (ili Model()), Rank(), TopN**

Agregacione funkcije

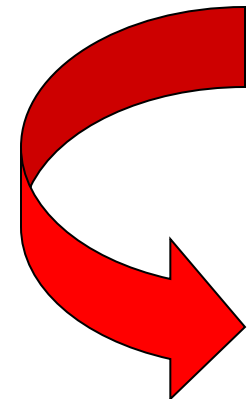
- **Distributivne funkcije**

- Primer **SUM**

```
CREATE MATERIALIZED VIEW KupacProiz_MV AS
  SELECT r.KupacID, s.Proizvod, SUM(Kolicina) UkKolic
  FROM Racun r, Stavka s
  WHERE r.RacBr = s.RacBr
  GROUP BY r.KupacID, s.Proizvod
```

```
SELECT r.KupacID, SUM(Kolicina) KupUkKolic
FROM Racun r, Stavka s
WHERE r.RacBr = s.RacBr
GROUP BY r.KupacID
```

```
SELECT KupacID, SUM(UkKolic) KupUkKolic
FROM KupacProiz_MV
GROUP BY KupacID
```



Agregacione funkcije

- **Algebarske funkcije**

- Primer **AVG**

```
CREATE MATERIALIZED VIEW KupacProiz_MV AS  
  SELECT r.KupacID, s.Proizvod, AVG(Kolicina) SrKol,  
          SUM(Kolicina) UkKolic, COUNT(*) UkStav  
FROM Racun r, Stavka s  
WHERE r.RacBr = s.RacBr  
GROUP BY r.KupacID, s.Proizvod
```

```
SELECT r.KupacID, AVG(Kolicina) KupSrKolic  
FROM Racun r, Stavka s  
WHERE r.RacBr = s.RacBr  
GROUP BY r.KupacID
```

```
SELECT KupacID, SUM(UkKolic) / SUM(UkStav) KupSrKolic  
FROM KupacProiz_MV  
GROUP BY KupacID
```

Agregacione funkcije

- **Holističke funkcije**

- Primer **Median**

- $\text{Median}(\{x_i \mid i = 1, \dots, n\})$
- vrednost srednjeg člana uređenog niza vrednosti
 - kada je n neparan
- srednja vrednost srednja dva člana uređenog niza vrednosti
 - kada je n paran

- tabela činjenica

- $\text{GradProiz_Prodaja}(\{\text{Grad, Proizvod, Količina}\})$
- grupisanje vrednosti niza po gradovima i proizvodima unutar gradova
- $\text{Median}(\{količina_{ij} \mid i=1, \dots, m, j=1, \dots, n_i\})$
 - $i=1, \dots, m$ - indeks različitih gradova
 - $j=1, \dots, n_i$ - indeks različitih proizvoda za i -ti grad

Agregacione funkcije

- **Holističke funkcije**

- Primer **Median**

- Median(3, 5, 6, 7, 8, 9, 10, 11, 12) = 8
 - SORT(Količina)

- drugi način - grupisanje

- $m = 3$
 - (NS, KG, BG)
- $n_1 = 3$
 - (NS, (Cipele, Jeans, Košulja))
- $n_2 = 2$
 - (KG, (Cipele, Jeans))
- $n_3 = 4$
 - (BG, (Cipele, Jeans, Košulja, Tašna))

GradProiz_Prodaja

Grad	Proizvod	Količ.
NS	Cipele	10
NS	Jeans	11
NS	Košulja	12
KG	Cipele	5
KG	Jeans	6
BG	Cipele	7
BG	Jeans	8
BG	Košulja	9
BG	Tašna	3

Agregacione funkcije

- **Holističke funkcije**

- Primer **Median**

- Materijalizacija pogleda
 - GradMedian_MV

GradMedian_MV

<i>Grad</i>	<i>ListaKoličina</i>	<i>GrMed</i>
<i>NS</i>	<i>10, 11, 12</i>	<i>11.0</i>
<i>KG</i>	<i>5, 6</i>	<i>5.5</i>
<i>BG</i>	<i>3, 7, 8, 9</i>	<i>7.5</i>

- $\text{Median}(3, 5, 6, 7, 8, 9, 10, 11, 12) = 8$
 - ne može se izračunati uz pomoć kolone GrMed
 - može se izračunati uz pomoć kolone ListaKoličina – opet isti niz
 - » `MERGE(ListaKoličina)`

GradProiz_Prodaja

<i>Grad</i>	<i>Proizvod</i>	<i>Količ.</i>
<i>NS</i>	<i>Cipele</i>	<i>10</i>
<i>NS</i>	<i>Jeans</i>	<i>11</i>
<i>NS</i>	<i>Košulja</i>	<i>12</i>
<i>KG</i>	<i>Cipele</i>	<i>5</i>
<i>KG</i>	<i>Jeans</i>	<i>6</i>
<i>BG</i>	<i>Cipele</i>	<i>7</i>
<i>BG</i>	<i>Jeans</i>	<i>8</i>
<i>BG</i>	<i>Košulja</i>	<i>9</i>
<i>BG</i>	<i>Tašna</i>	<i>3</i>

Agregacione funkcije

- **Holističke funkcije**

- Primer **TopN / BottomN**

- funkcija selekcije N najbolje ili najlošije rangiranih torki
 - po zatom kriterijumu
 - meri koja se može iskoristiti u ORDER BY klauzuli i
 - po zadatim dimenzijama

- Upit - primer

- selektovati N najbolje / najlošije prodavanih proizvoda u zatom gradu
 - selektovati N najbolje / najlošije prodavanih proizvoda za svaki grad

Agregacione funkcije

- **Holističke funkcije**

- Primer **TopN / BottomN**

- Upiti tipa N najbolje / najlošije rangiranih mogu se realizovati na više načina

- korišćenjem specijalizovanih funkcija tipa TOP, LIMIT, ROWNUM, u kombinaciji s klauzulom ORDER BY (po potrebi)
 - korišćenjem analitičkih funkcija RANK() i DENSE_RANK()
 - obrađene u posebnom odeljku prezentacije (analitičke funkcije i upiti)
 - korišćenjem povezanih ugnježdenih upita, tipa
 - "selektovati torke za koje broj torki sa boljim / gorim vrednostima po zatom kriterijumu je manji od N"

Agregacione funkcije

- **Holističke funkcije**

- Primer **TopN / BottomN**

- <n> - maksimalan broj prvoselektovanih torke

- **Microsoft SQL Server**

- **SELECT TOP** <n> column FROM table

- **PostgreSQL and MySQL**

- **SELECT** column FROM table **LIMIT** <n> [**OFFSET** <m>]

- <m> - od koje torke počinje selektovanje

- **Oracle**

- **SELECT** column FROM table **WHERE ROWNUM** <= <n>

Agregacione funkcije

- **Holističke funkcije**

- generalno, mogu biti resursno vrlo zahtevne
 - pogotovo u situacijama primene u unakrsnom tabeliranju (operator CUBE)
 - kada treba napraviti niz od 2^n GROUP BY operacija, za n kolona u GROUP BY klauzuli
- primer **TopN** / **BottomN** i optimizacija upita
 - nema smisla selektovati i prenositi u OM ceo skup torki, ukoliko se zahteva samo prvih N
 - različiti SUBP podržavaju direktivu optimizatoru upita, kojom se saopštava da se upit ograničava samo na prvih N torki

Aggregacione funkcije

- **Holističke funkcije**

- primer **TopN / BottomN** i optimizacija upita

- TopN direktiva optimizatoru upita

- IBM DB2

```
SELECT ...
```

```
...
```

```
ORDER BY ...
```

```
OPTIMIZE FOR <n> ROWS
```

- Oracle

- OPTIMIZER_MODE = FIRST_ROWS_N

- » inicijalizacioni parametar

- ALTER SESSION SET OPTIMIZER_MODE=FIRST_ROWS_N

- » parametar sesije

- SELECT /*+ FIRST_ROWS(N) */ ...

- » optimizaciona direktiva

Sadržaj

- Osnovne karakteristike OLAP sistema
- Arhitektura OLAP sistema
- Dimenzione ROLAP strukture
- Osnovni OLAP upiti
- Agregacione funkcije
- Unakrsno tabeliranje
- Analitičke funkcije i upiti

Unakrsno tabeliranje

- **Unakrsno tabeliranje**
 - **Pivoting (Cross-Tabulation)**
 - grupisanje i agregiranje podataka po izabranim dimenzijama
 - u n-dimenzionalnoj hiperkocki, bira se k ($k < n$) dimenzija
 - vrši se grupisanje i agregiranje ("sumiranje") podataka i formira se k-dimenzionalna matrica
 - vrši se agregiranje podataka po svakoj dimenziji, ili svakoj kombinaciji dimenzija matrice
 - npr, biraju se 2 dimenzije i formira dvodimenzionalna matrica

Unakrsno tabeliranje

- **Pivoting**

- Primer

- Posmatra se tabela činjenica

KupacProizDat(KupacID, Proizvod, Datum, Total)

- pivoting po dimenzijama (**KupacID, Proizvod**)

<i>Kup / Pr</i>	<i>flash</i>	<i>miš</i>	Σ
<i>201</i>	<i>2500</i>	<i>3000</i>	<i>5500</i>
<i>202</i>	<i>500</i>	<i>2100</i>	<i>2600</i>
Σ	<i>3000</i>	<i>5100</i>	<i>8100</i>

Unakrsno tabeliranje

- **Pivoting**
 - Primer

Relaciona (tabelarna)
reprezentacija

<i>KupačID</i>	<i>Proizvod</i>	<i>UkCena</i>
201	flash	2500
201	miš	3000
201		5500
202	flash	500
202	miš	2100
202		2600
	flash	3000
	miš	5100
		8100

Unakrsno tabeliranje

- Pivoting**

– Primer

Location

<i>CityId</i>	<i>City</i>
1	NS
2	KG
3	BG

SalesPerson

<i>PerId</i>	<i>Name</i>
1	Jovan
2	Suzana
3	Janko
4	Suzana
5	Ana

Sales

<i>CityId</i>	<i>PerId</i>	<i>TimId</i>	<i>Amnt</i>
1	1	1	230
1	1	2	300
1	1	1	310
1	2	7	50
2	3	1	550
2	3	5	100
3	4	6	880
3	5	1	60
3	5	2	60
3	5	4	140

Time

<i>TimId</i>	<i>Day</i>
1	Mon
2	Tue
3	Wed
4	Thu
5	Fri
6	Sat
7	Sun

Unakrsno tabeliranje

- **Pivoting**

- Primer

- po dimenzijama (City, Day)

	<i>Mon</i>	<i>Tue</i>	<i>Wed</i>	<i>Thu</i>	<i>Fri</i>	<i>Sat</i>	<i>Sun</i>	SubTotal
<i>BG</i>	60	60	0	140	0	880	0	1140
<i>KG</i>	550	0	0	0	100	0	0	650
<i>NS</i>	540	300	0	0	0	0	50	890
SubTotal	1150	360	0	140	100	880	50	2680

Unakrsno tabeliranje

- **Pivoting**

- Primer – više SQL naredbi

```
SELECT City, Day, SUM(Amnt) AS Sales
FROM Sales s, Location l, Time t
WHERE s.CityId = l.CityId
      AND s.TimeId = t.TimeId
GROUP BY City, Day
```

<i>City</i>	<i>Day</i>	<i>Sales</i>
<i>BG</i>	<i>Mon</i>	<i>60</i>
<i>BG</i>	<i>Tue</i>	<i>60</i>
<i>BG</i>	<i>Thu</i>	<i>140</i>
<i>BG</i>	<i>Sat</i>	<i>880</i>
<i>KG</i>	<i>Mon</i>	<i>550</i>
<i>KG</i>	<i>Fri</i>	<i>100</i>
<i>NS</i>	<i>Mon</i>	<i>540</i>
<i>NS</i>	<i>Tue</i>	<i>300</i>
<i>NS</i>	<i>Sun</i>	<i>50</i>

Unakrsno tabeliranje

- Pivoting**

- Primer – više SQL naredbi

```
SELECT Day, SUM(Amnt) AS Total
FROM Sales s, Time t
WHERE s.TimId = t.TimId
GROUP BY Day
```

<i>Day</i>	<i>Total</i>
<i>Mon</i>	1150
<i>Tue</i>	360
<i>Thu</i>	140
<i>Fri</i>	100
<i>Sat</i>	880
<i>Sun</i>	50

```
SELECT City, SUM(Amnt) AS Total
FROM Sales s, Location l
WHERE s.CityId = l.CityId
GROUP BY City
```

<i>City</i>	<i>Total</i>
<i>BG</i>	1140
<i>KG</i>	650
<i>NS</i>	890

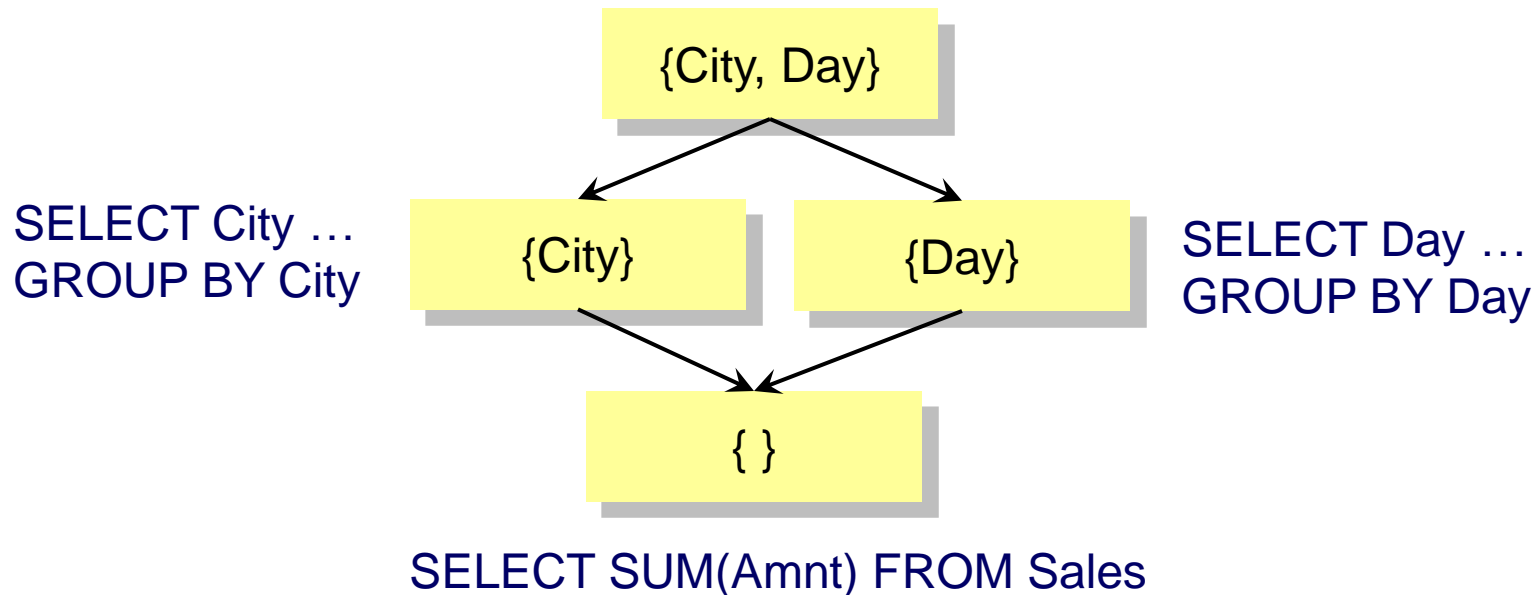
```
SELECT SUM(Amnt)
FROM Sales
```

<i>SUM(Amnt)</i>
2680

Unakrsno tabeliranje

- **Mreža kocke (Cube Lattice)**
 - graf kombinacija dimenzija po kojima se vrši agregiranje podataka

```
SELECT City, Day, SUM(Amnt)... GROUP BY City, Day
```



Unakrsno tabeliranje

- **Pivoting**

- Primer – jedna SQL naredba i relaciona prezentacija

```
(SELECT City, Day, SUM(Amnt)
FROM Sales NATURAL JOIN Location NATURAL JOIN Time
GROUP BY City, Day)
```

```
UNION ALL
```

```
(SELECT City, NULL, SUM(Amnt)
FROM Sales NATURAL JOIN Location
GROUP BY City)
```

```
UNION ALL
```

```
(SELECT NULL, Day, SUM(Amnt)
FROM Sales NATURAL JOIN Time
GROUP BY Day)
```

```
UNION ALL
```

```
(SELECT NULL, NULL, SUM(Amnt) FROM Sales)
```

Unakrsno tabeliranje

- **Efikasnost upita**

- pivoting na n dimenzija zahteva 2^n SELECT naredbi, tipa roll-up operacija
 - bilo da su nezavisne, ili povezane sa UNION ALL
- optimizator upita bi svaku takvu SELECT naredbu tretirao posebno
 - ne bi koristio međurezultate za optimizaciju SELECT naredbi, iako je to moguće
 - svaka podređena SELECT naredba u mreži kocke mogla bi biti realizovana korišćenjem rezultata njenih direktno nadređenih SELECT naredbi
- motivacija za uvođenje novih operatora u GROUP BY klauzulu
 - ROLLUP, CUBE, GROUPING SETS

Unakrsno tabeliranje

- **Efikasnost upita**

- Primer

- neka je V1 materijalizovan rezultat SELECT naredbe

```
SELECT City, Day, SUM(Amnt) AS Sale
FROM Sales NATURAL JOIN Location NATURAL JOIN Time
GROUP BY City, Day
```

- vrši se query rewriting naredbe

```
SELECT City, NULL, SUM(Amnt)
FROM Sales NATURAL JOIN Location
GROUP BY City
```

- u oblik

```
SELECT City, NULL, SUM(Sale)
FROM V1
GROUP BY City
```

Unakrsno tabeliranje

- **Operator GROUP BY CUBE**

- ekvivalent za niz od 2^n SELECT ... GROUP BY naredbi

- za svaku moguću kombinaciju od n zadatih dimenzija, koja se pojavljuje kao čvor u mreži kocke

- Primer

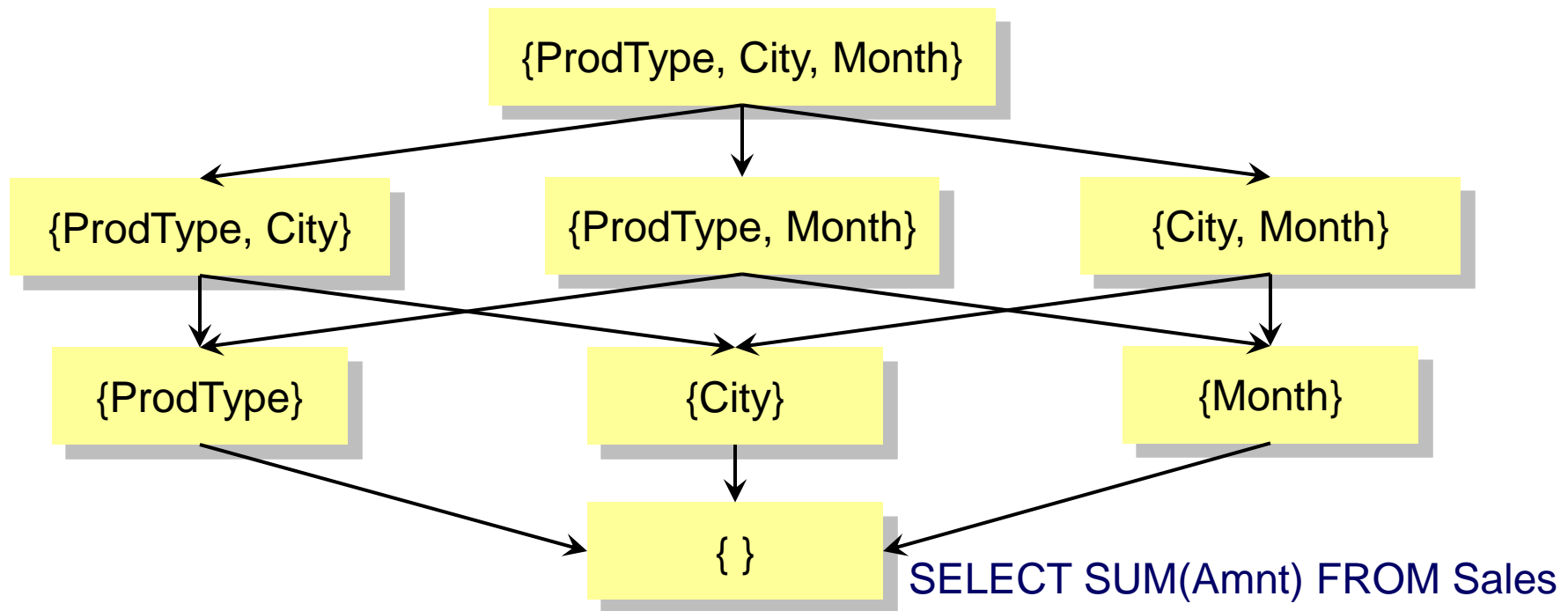
```
SELECT City, Month, SUM(Amnt) AS Sale
FROM Sales NATURAL JOIN Location NATURAL JOIN Time
GROUP BY CUBE(City, Month)
```

- zamena za 4 SELECT ... GROUP BY naredbe
 - {City, Month}, {City}, {Month}, {}

Unakrsno tabeliranje

- Operator GROUP BY CUBE**

```
SELECT ProdType, City, Month, SUM(Amnt)
FROM Sales NATURAL JOIN Product NATURAL JOIN Location
      NATURAL JOIN Time
GROUP BY CUBE(ProdType, City, Month)
```



Unakrsno tabeliranje

- **Operator GROUP BY ROLLUP**

- ekvivalent za niz od n SELECT ... GROUP BY naredbi

- za svaki levi podniz od n zadatih dimenzija – jedan put od minimuma do maksimuma u mreži kocke

- Primer

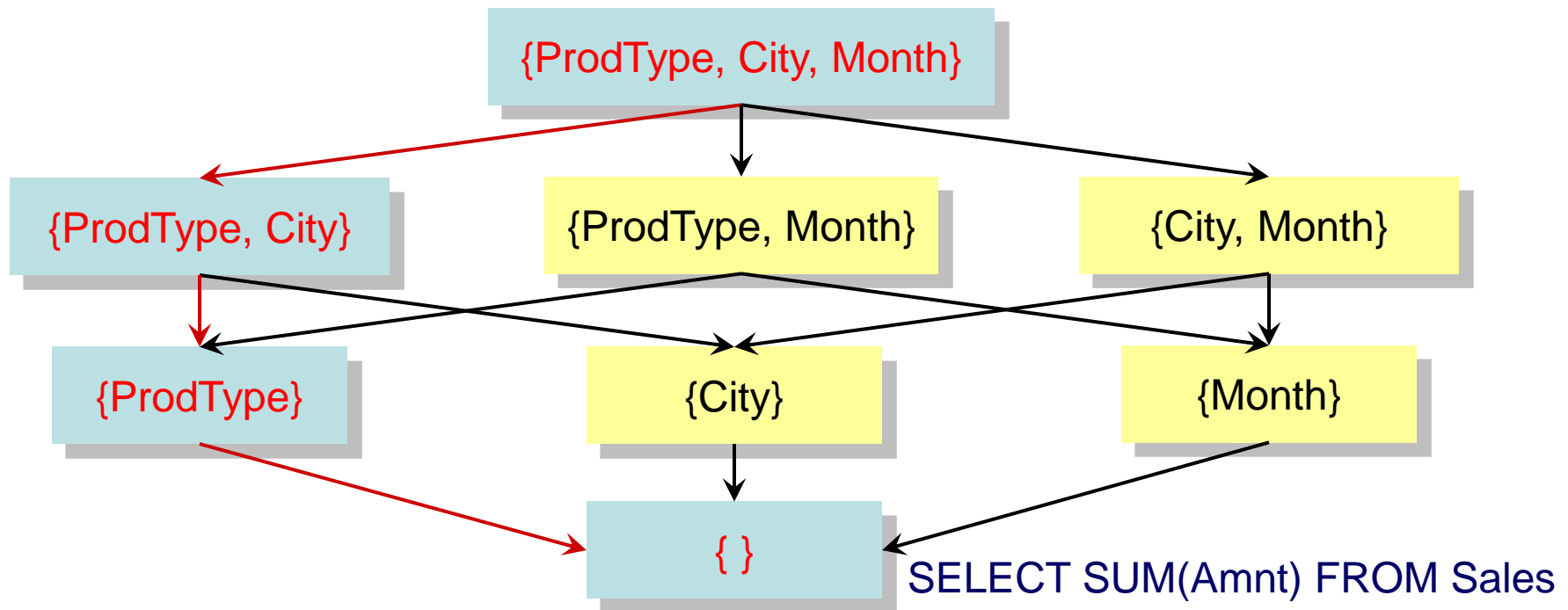
```
SELECT City, Day, SUM(Amnt) AS Sale
FROM Sales NATURAL JOIN Location NATURAL JOIN Time
GROUP BY ROLLUP(City, Day)
```

- zamena za 3 SELECT ... GROUP BY naredbe
 - {City, Day}, {City}, {}

Unakrsno tabeliranje

- Operator GROUP BY ROLLUP**

```
SELECT ProdType, City, Month, SUM(Amnt)
FROM Sales NATURAL JOIN Product NATURAL JOIN Location
      NATURAL JOIN Time
GROUP BY ROLLUP(ProdType, City, Month)
```



Unakrsno tabeliranje

- **Operator GROUP BY GROUPING SETS**

- ekvivalent za eksplicitnu specifikaciju željenih SELECT ... GROUP BY naredbi

- Primer

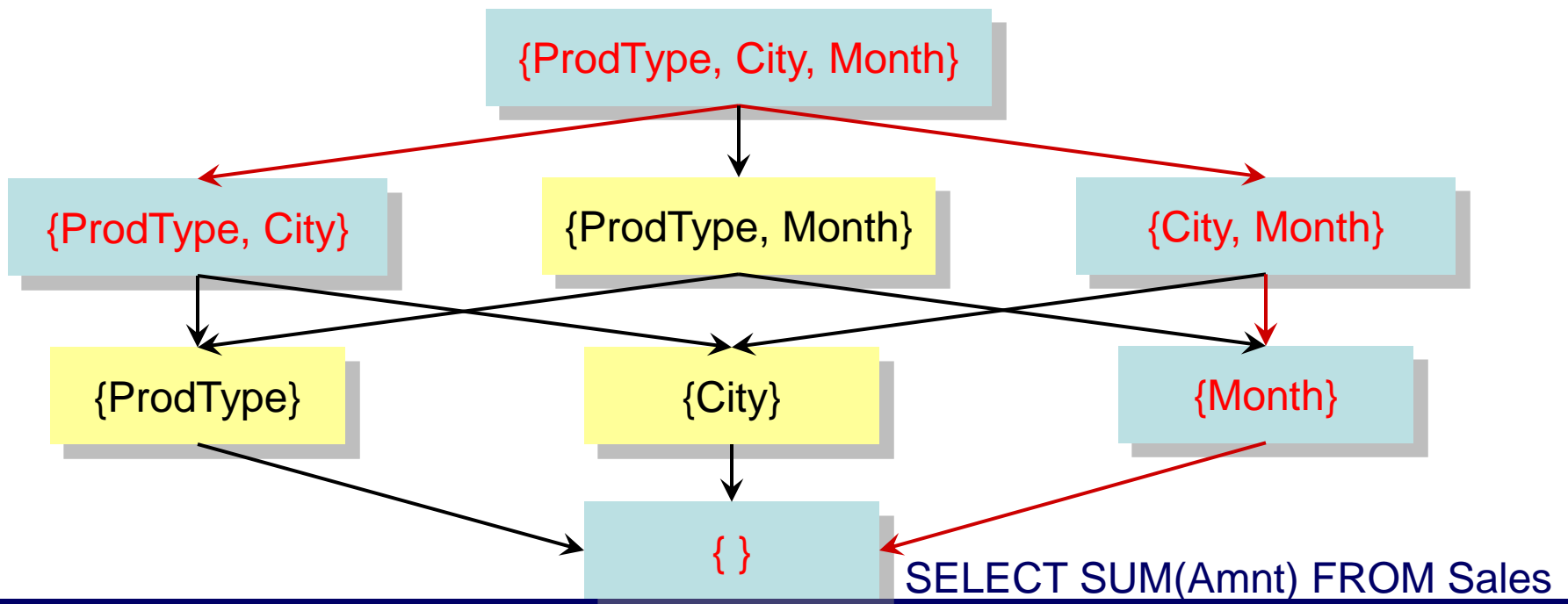
```
SELECT City, Day, SUM(Amnt) AS Sale
FROM Sales NATURAL JOIN Location NATURAL JOIN Time
GROUP BY GROUPING SETS((City, Day), City)
```

- zamena za 2 SELECT ... GROUP BY naredbe
 - {City, Day}, {City}

Unakrsno tabeliranje

- Operator GROUP BY GROUPING SETS**

```
SELECT ProdType, City, Month, SUM(Amnt)
FROM Sales NATURAL JOIN Product NATURAL JOIN Location
      NATURAL JOIN Time
GROUP BY GROUPING SETS((ProdType, City, Month),
      (City, Month), (ProdType, City), (Month), ())
```



Unakrsno tabeliranje

- **Višestruki GROUP BY GROUPING SETS**

- mogućnost višestrukog navođenja operatora GROUPING SETS

- u cilju iskazivanja dekartovog proizvoda klauzula tipa GROUP BY

- Primer

```
SELECT City, Day, SUM(Amnt) AS Sale
FROM Sales NATURAL JOIN Location NATURAL JOIN Time
GROUP BY GROUPING SETS(City, Region),
          GROUPING SETS(Day, Month)
```

- zamena za $2 \times 2 = 4$ SELECT ... GROUP BY naredbe
 - {City, Day}, {City, Month}, {Region, Day}, {Region, Month}

Unakrsno tabeliranje

- Operatori CUBE, ROLLUP, GROUPING SETS
 - prednosti upotrebe
 - efikasnije izvršavanje i optimizacija pivoting upita
 - korišćenje međurezultata, saglasno poziciji upita u mreži kocke
 - izračunavanje agregiranih podataka grublje granularnosti korišćenjem agregiranih podataka finije granularnosti
 - korišćenje dobrih osobina distributivnih i algebarskih agregacionih funkcija
 - izbegavanje primene operatora UNION ALL
 - kompaktniji zapis upita, korišćenjem jedne, jednostavnije SELECT naredbe

Unakrsno tabeliranje

- Pregled osobina različitih vrsta OLAP upita

OLAP operacija	Upotreba atributa hijerarhija	Broj upotr. dimenzija	Upotreba agregacionih funkcija	Upotreba σ	Mogu biti iskazane putem SQL naredbe
Roll-up	Y	Isti ili manji	Y	N	SELECT... AGG GROUP BY
Drill-down	Y	Isti ili veći	Moguća	N	SELECT... AGG GROUP BY
Slice	N	Manji	N	Y	SELECT... WHERE cond
Dice	N	Može biti manji	N	Y	SELECT... WHERE cond
Pivot / Cross tab	Y	Manji	Y (višestruka)	N	Multiple roll-up / CUBE, ROLLUP

Sadržaj

- Osnovne karakteristike OLAP sistema
- Arhitektura OLAP sistema
- Dimenzione ROLAP strukture
- Osnovni OLAP upiti
- Agregacione funkcije
- Unakrsno tabeliranje
- Analitičke funkcije i upiti

Analitičke funkcije i upiti

- **Upiti sa analitičkim funkcijama i WINDOW klauzulom u izrazima**
 - dinamičko grupisanje podataka u odnosu na tekuću torku sa primenom agregacionih funkcija
- **Analitičke funkcije**
 - agregacione (skupovne) funkcije
 - opšte (SUM, AVG,...) i specijalizovane (RANK, LAG,...)
 - upotreba dominantno u sintaksnom obliku izraza s WINDOW klauzulom
 - primenjuju se nad grupama torke, ali se rezultat iskazuje uz pojedinačne torke unutar grupa

Analitičke funkcije i upiti

- **Primeri upita**

- **U1:** prikazati, za svaki dan, srednju vrednost prodane količine proizvoda u sledećih n dana, u odnosu na tekući dan
 - za svaki dan, mora se računati srednja vrednost za n susednih dana
- **U2:** prikazati prvih n proizvoda, rangiranih po kumulativnoj prodatoj količini za svaki mesec u protekloj godini
- **U3:** rangirati sve proizvode po ukupnoj prodatoj količini u protekloj godini i, za svaki proizvod, prikazati razliku ukupne prodane količine u odnosu na proizvod, rangiran ispod datog

Analitičke funkcije i upiti

- **Motivacija**

- upiti navedenog tipa mogu se iskazati i putem "klasičnih" SELECT naredbi (ANSI SQL:1992), ali
 - korišćenjem spajanja i kompleksnih podupita
 - često, SELECT naredba za takve upite je vrlo kompleksna i za pisanje i za izvršavanje
- mogu se znatno jednostavnije iskazati putem analitičkih funkcija s WINDOW klauzulom (ANSI SQL:1999)

Analitičke funkcije i upiti

- **WINDOW klauzula u analitičkoj funkciji**
 - specificira dinamički uređeni niz torki (window) u okolini svake tekuće torke
 - koja je predmet procesiranja SELECT naredbe
 - dozvoljava primenu agregacionih funkcija na nizu torki tipa window
 - sa ciljem pridruživanja rezultata funkcije tekućoj torci
 - Primer
 - svaka tekuća torka sadrži dnevnu prodatu količinu
 - primenom klauzule WINDOW, može se izračunati ukupna prodata količina za prethodna 3 dana ($n=3$) i pridružiti, kao rezultat, svakoj tekućoj torci
 - na taj način, dobija se trodnevna ukupna prodata količina
 - dinamički agregirana vrednost, u odnosu na tekuću torku


Analitičke funkcije i upiti

- Primer

- rezultat upita je dinamička vrednost SUM(Sales)

- izračunata uvek za prethodna tri dana, s obzirom na uređenje po TimeId

TimeId	DayOfWeek	Sales	SUM(Sales)
1	Monday	55	00
2	Tuesday	66	55
3	Wednesday	33	121
4	Thursday	55	154
5	Friday	77	154
6	Saturday	22	165
7	Sunday	11	154



Analitičke funkcije i upiti

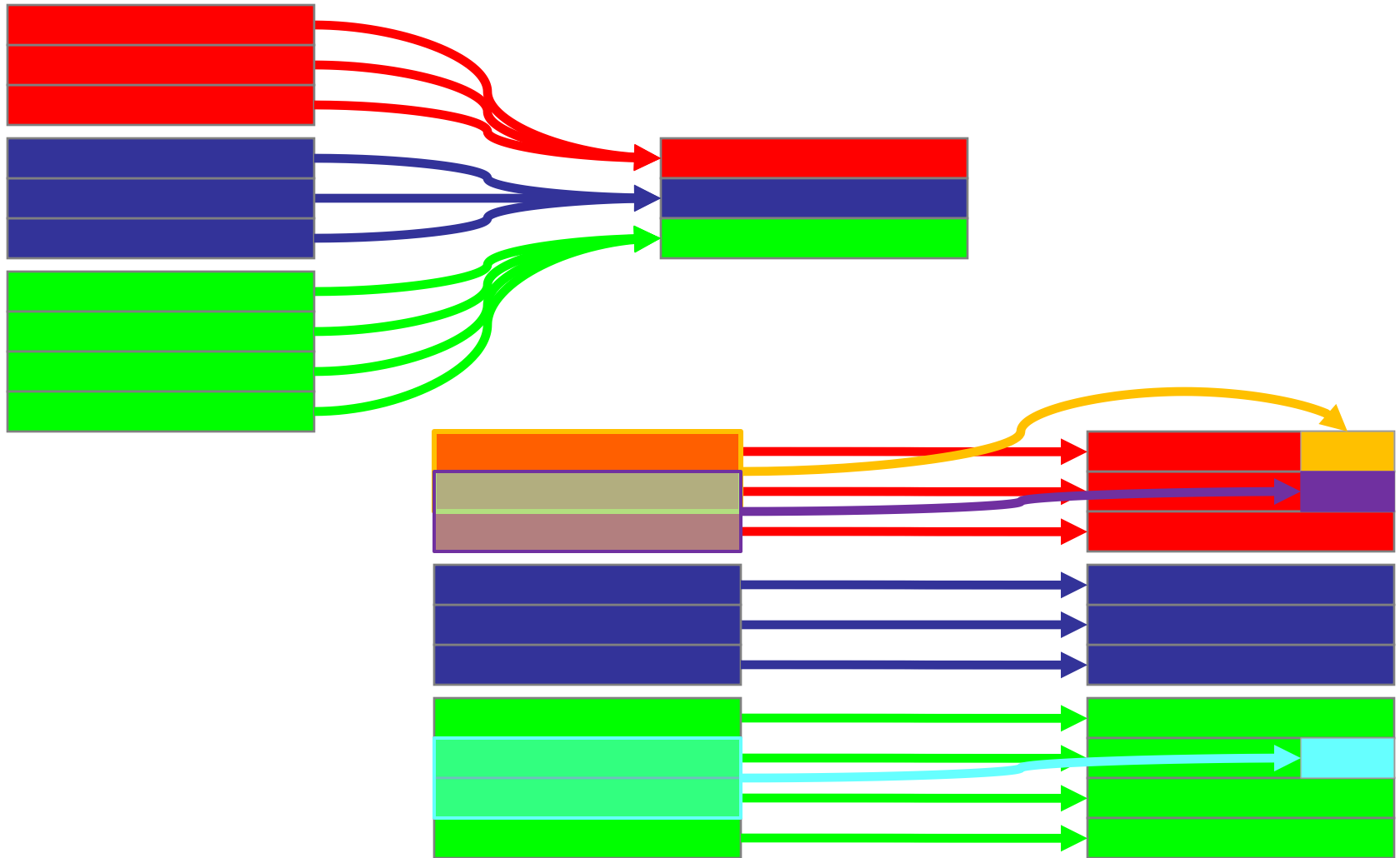
- **Poređenje WINDOW / GROUP BY**
 - I GROUP BY i analitička funkcija sa WINDOW klauzulom
 - uvek specificiraju particiju skupa torki
 - grupe su nepresečne
 - sprovodi se uređivanje generisanih grupa torki po zadatom kriterijumu (tj. listi izraza u GROUP BY ili PARTITION BY klauzuli)
 - ORDER BY klauzula – uređenje rezultujućeg skupa torki
 - GROUP BY
 - reprezentant cele grupe u rezultatu upita je tačno jedna toraka
 - agregacione funkcije primenjuju se samo nad celim grupama
 - jedna vrednost agregacione funkcije za jednu rezultujuću toraku

Analitičke funkcije i upiti

- **Poređenje WINDOW / GROUP BY**
 - analitička funkcija sa WINDOW klauzulom
 - u rezultat upita ulaze sve selektovane torke – kao kod "običnih" upita bez GROUP BY klauzule
 - po jedna vrednost agregacione funkcije pridružuje se svakoj (tekućoj) torci upita
 - agregacione funkcije primenjuju se nad izabranim torkama u grupama
 - definiše se eksplicitno opseg torki u grupi (WINDOW) na koji se primenjuje agregaciona funkcija u kontekstu tekuće torke
 - opsezi torki u grupi, za različitu tekuću torku, mogu se preklapati

Analitičke funkcije i upiti

- GROUP BY vs. WINDOW**



Analitičke funkcije i upiti

- **Sintaksa upita sa analitičkim funkcijama**

```
SELECT <attribute_list_1>,  
    <aggregate_function> OVER W AS <moving_window_name>  
FROM <table_list>  
WHERE <join_and_select_conditions>  
WINDOW W AS ([PARTITION BY <attribute_list_2>]  
    [ORDER BY <attribute_list_3>]  
    [<window_frame_declaration>])
```

- alternativno (Oracle)

```
SELECT <attribute_list_1>, <aggregate_function> OVER  
    ( [PARTITION BY part_clause] [ORDER BY ord_clause]  
    [<window_frame_declaration>] ) AS <moving_window_name>  
FROM <table_list>  
WHERE <join_and_select_conditions>
```

Analitičke funkcije i upiti

- **Sintaksa upita sa analitičkim funkcijama**

- Pravila

- attribute_list_2 je podlista od attribute_list_1
- attribute_list_3 je podlista od attribute_list_1
- attribute_list_2 i attribute_list_3 su nepresečne
- window_frame_declaration
 - specificira karakteristike window-a
- analitička funkcija izvršava se nakon svih slektovanja torki, spajanja, podupita i primene GROUP BY i HAVING klauzule

Analitičke funkcije i upiti

- **Način izvršavanja naredbe**

- SELECT ... FROM ... WHERE izvršava se uobičajeno

- kontekst tekuće torke t u SELECT je uvek poznat
 - WINDOW se može generisati u kontekstu tekuće torke t i generisane grupe, prema zadatoj PARTITION BY klauzuli

- Koraci

- particioniranje cele tabele na grupe vrši se saglasno PARTITION BY klauzuli
 - svaka grupa sadrži torke sa istim vrednostima za attribute_list_2
 - uređivanje torki u svakoj grupi particije vrši se saglasno ORDER BY klauzuli, tj. vrednostima za attribute_list_3
 - Window Framing - definisanje granica window-a
 - vrši se u odnosu na tekuću torku rezultata upita
 - uz pomoć klauzule <window_frame_declaration>

Analitičke funkcije i upiti

- **Način izvršavanja naredbe**
 - Kolona `moving_window_name`, za tekuću torku, dobija vrednost primenom agregacione funkcije na odgovarajući window
 - agregaciona funkcija primenjuje se na sve torke window-a
- **Napomene**
 - glavni upit može sadržati sopstvenu `ORDER BY` klauzulu
 - `ORDER BY` klauzula u specifikaciji za `WINDOW` nema uticaja na uređivanje torke glavnog upita
 - `<window_frame_declaration>` ima bitan uticaj na dobijene rezultate upita
 - jer definiše granice window-a u svakoj grupi particije tabele

Analitičke funkcije i upiti

- **Specifikacija granica za WINDOW**

– `<window_frame_declaration> ::=`
`[ROWS | RANGE]`
`{`
`{UNBOUNDED PRECEDING | <value_expr> PRECEDING}`
`| BETWEEN {CURRENT ROW |`
`UNBOUNDED PRECEDING |`
`<value_expr> {PRECEDING | FOLLOWING}}`
`AND {CURRENT ROW |`
`UNBOUNDED FOLLOWING |`
`<value_expr> {FOLLOWING | PRECEDING}}`
`}`

Analitičke funkcije i upiti

- **Specifikacija granica za WINDOW**

- **ROWS**

- granice za window opredeljuju se direktno
 - eksplicitnim navođenjem rednih brojeva torki (nakon primenjene ORDER BY klauzule), u odnosu na tekuću torku
 - npr. "obuhvati prozorom od n prethodnih do m narednih torki u odnosu na tekuću" ($\langle \text{value_expr} \rangle : m \geq 0, n \geq 0$)

- **RANGE**

- granice za window opredeljuju se posredno
 - redni brojevi torki izračunavaju se na osnovu vrednosti izraza $\langle \text{value_expr} \rangle$ (logical offset), koji se nasleđuju od izraza u ORDER BY klauzuli
 - dozvoljena je numerička vrednost izraza u slučaju izraza bilo kog tipa u ORDER BY, ili intervalna vrednost datuma samo u slučaju izraza datumskog tipa u ORDER BY

Analitičke funkcije i upiti

- **Specifikacija granica za WINDOW**
 - **UNBOUNDED PRECEDING**
 - početna granica za window je prva toraka u grupi
 - **UNBOUNDED FOLLOWING**
 - krajnja granica za window je poslednja toraka grupe
 - **CURRENT ROW**
 - označava tekuću toraku
 - može biti upotrebljena i kao početna i kao krajnja granica za window

Analitičke funkcije i upiti

- **Specifikacija granica za WINDOW**

- Sintaksna alternativa I:

- {**UNBOUNDED PRECEDING** | <value_expr> **PRECEDING**}
- početna granica: eksplicitno se zadaje
- krajnja granica: podrazumevana, CURRENT ROW

- Sintaksna alternativa II:

- **BETWEEN... AND...**
- omogućava eksplicitno specificiranje, redom, početne i krajnje granice za window

Analitičke funkcije i upiti

- **Specifikacija granica za WINDOW**

- <value_expr>

- direktno ili posredno zadavanje početne (**PRECEDING**) ili krajnje (**FOLLOWING**) granice putem izraza

- može obuhvatati

- konstante
- kolone
- neanalitičke funkcije

- Primeri

- direktno specificiranje granice – upotreba konstante
 - ROWS n PRECEDING
- posredno specificiranje granice i upotreba INTERVAL
 - RANGE INTERVAL n {DAY | MONTH | YEAR} PRECEDING

Analitičke funkcije i upiti

- **Specifikacija granica za WINDOW**

- Pravila

- ROWS i RANGE ne mogu se istovremeno pojaviti u istoj WINDOW klauzuli
- granice prozora definišu se u odnosu na tekuću torku, ali ne moraju obuhvatati tekuću torku
 - ROWS 6 PRECEDING AND 3 PRECEDING
 - ROWS 1 FOLLOWING AND 5 FOLLOWING
- vrednost donje granice ne sme prevazići vrednost gornje granice
- granice prozora ne mogu preći preko granica grupe
 - ukoliko izračunati <value_expr> prevazilazi preko granice grupe, automatski se svodi na
 - UNBOUNDED PRECEDING, za donju granicu i
 - UNBOUNDED FOLLOWING, za gornju granicu

Analitičke funkcije i upiti

- Primer – radnici i ukupno radnika u istom depart.

```
SELECT EmpID, HireDate, DeptID,  
       COUNT(EmpID) OVER W AS EmpCount
```

```
FROM Employees
```

```
WINDOW W AS (PARTITION BY DeptID)
```

```
ORDER BY DeptID
```

- prozor obuhvata celu grupu, za svaki DeptID

<i>EmpID</i>	<i>HireDate</i>	<i>DeptID</i>	<i>EmpCount</i>
7300	2006-07-01	100	2
7301	2005-07-05	100	2
7400	1998-07-13	110	3
7422	1991-07-01	110	3
7473	1979-07-21	110	3
7800	2008-07-29	120	1

Analitičke funkcije i upiti

- Primer – radnici i ukupno radnika u svim depart.

```
SELECT EmpID, HireDate, DeptID,  
       COUNT(EmpID) OVER W AS EmpCntTot  
FROM Employees  
WINDOW W AS ()  
ORDER BY DeptID
```

- ceo upit je jedna grupa i prozor obuhvata celu grupu

<i>EmpID</i>	<i>HireDate</i>	<i>DeptID</i>	<i>EmpCntTot</i>
7300	2006-07-01	100	6
7301	2005-07-05	100	6
7400	1998-07-13	110	6
7422	1991-07-01	110	6
7473	1979-07-21	110	6
7800	2008-07-29	120	6

Analitičke funkcije i upiti

- Primer – radnici i broj radnika sa zaradom manjom za 500 i sa zaradom većom za 500

```
SELECT EmpID, Salary, DeptID,  
       COUNT(*) OVER (  
           PARTITION BY DeptID ORDER BY Salary  
           RANGE BETWEEN UNBOUNDED PRECEDING AND  
                               Salary – 500 PRECEDING  
       ) AS EmpCntLT500  
  
       COUNT(*) OVER (  
           PARTITION BY DeptID ORDER BY Salary  
           RANGE BETWEEN Salary + 500 FOLLOWING AND  
                               UNBOUNDED FOLLOWING  
       ) AS EmpCntGT500  
  
FROM Employees  
ORDER BY DeptID, Salary
```

Analitičke funkcije i upiti

- Primer – radnici i broj radnika sa zaradom manjom za 500 i sa zaradom većom za 500

```
SELECT EmpID, Salary, DeptID,
       COUNT(*) OVER (
         PARTITION BY DeptID ORDER BY Salary ...
```

<i>EmpID</i>	<i>Salary</i>	<i>DeptID</i>	<i>EmpCntLT500</i>	<i>EmpCntGT500</i>
7300	3000,00	100	0	1
7301	4000,00	100	1	0
7400	3500,00	110	0	1
7422	3800,00	110	0	0
7473	4200,00	110	1	0
7800	6000,00	120	0	2
7801	6600,00	120	1	1
7805	7800,00	120	2	0

Analitičke funkcije i upiti

- Primer – kumulirajuća suma po AcctNo

```
SELECT AcctNo, Date, Amnt, SUM(Amnt) OVER W AS Balance
FROM Ledger
WINDOW W AS (PARTITION BY AcctNo
              ORDER BY Date
              ROWS UNBOUNDED PRECEDING)
ORDER BY AcctNo, Date
```

<i>AcctNo</i>	<i>Date</i>	<i>Amnt</i>	<i>Balance</i>
73829	2007-07-01	113.5	113.5
73829	2007-07-05	-52.06	61.44
73829	2007-07-13	36.25	97.69
82930	2007-07-01	10.56	10.56
82930	2007-07-21	32.55	43.11
82930	2007-07-29	-5.02	38.09

Analitičke funkcije i upiti

- Primer – pomerajuća srednja vrednost

```
SELECT AcctNo, Date, Amnt, AVG(Amnt) OVER W AS Moving7Days
FROM Ledger
WINDOW W AS (PARTITION BY AcctNo
              ORDER BY Date
              RANGE INTERVAL 7 DAY PRECEDING)
ORDER BY AcctNo, Date
```

Analitičke funkcije i upiti

- Primer – pomerajuća srednja vrednost

```
SELECT AcctNo, Date, Amnt, AVG(Amnt) OVER W AS Moving7Days
FROM Ledger ...
```

<i>AcctNo</i>	<i>Date</i>	<i>Amnt</i>	<i>Moving7Days</i>
73829	2007-07-03	113.5	113.5
73829	2007-07-09	-52.01	30.75
73829	2007-07-13	36.25	-7.88
73829	2007-07-14	10.56	-1.73
73829	2007-07-19	32.55	26.45
82930	2007-07-01	100.25	100.25
82930	2007-07-10	10.01	10.01
82930	2007-07-25	11.02	11.02
82930	2007-07-26	100.56	55.79
82930	2007-07-30	-5.02	35.52

Analitičke funkcije i upiti

- Primer – centrirana agregirana vrednost

```
SELECT AcctNo, Date, Amnt, SUM(Amnt) OVER W AS Balance
FROM Ledger
WINDOW W AS
(PARTITION BY AcctNo
ORDER BY Date
RANGE BETWEEN INTERVAL 1 MONTH PRECEDING
AND INTERVAL 1 MONTH FOLLOWING)
ORDER BY AcctNo, Date
```

- Napomena

- granice se određuju samo u odnosu na mesec, a ne i na dan u mesecu

Analitičke funkcije i upiti

- Primer – centrirana agregirana vrednost

```
SELECT AcctNo, Date, Amnt, SUM(Amnt) OVER W AS Balance
FROM Ledger
WINDOW W AS
(PARTITION BY AcctNo
ORDER BY Date
RANGE BETWEEN INTERVAL 1 MONTH PRECEDING
AND INTERVAL 1 MONTH FOLLOWING) ...
```

<i>AcctNo</i>	<i>Date</i>	<i>Amnt</i>	<i>Balance</i>
73829	2007-07-01	113.50	61.49
73829	2007-08-05	-52.01	97.74
73829	2007-09-13	36.25	-15.76
82930	2007-07-01	10.56	43.11
82930	2007-08-21	32.55	43.11
82930	2007-10-29	-5.02	-5.02

Analitičke funkcije i upiti

- **Specijalizovane analitičke funkcije**

- ROW_NUMBER()
- RANK()
- DENSE_RANK()
- LEAD(sql_expr, offset, default)
- LAG(sql_expr, offset, default)
- FIRST_VALUE(sql_expr)
- LAST_VALUE(sql_expr)

- pripadaju klasi holističkih funkcija

Analitičke funkcije i upiti

- **ROW_NUMBER()**

- dodela rednog broja tekućoj torci u grupi

```
SELECT EmpID, HireDate, DeptID,  
        ROW_NUMBER() OVER W AS RowNo  
FROM Employees  
WINDOW W AS ( PARTITION BY DeptID  
              ORDER BY HireDate NULLS LAST)  
ORDER BY DeptID, RowNo
```

<i>EmpID</i>	<i>HireDate</i>	<i>DeptID</i>	<i>RowNo</i>
7300	2004-07-01	100	1
7301	2005-07-05	100	2
7400	1979-07-13	110	1
7422	1991-07-01	110	2
7473	1998-07-21	110	3
7800	2008-07-29	120	1

Analitičke funkcije i upiti

- **RANK()** i **DENSE_RANK()**
 - funkcije rangiranja torki po poziciji unutar grupe, saglasno zadatom kriterijumu rangiranja (sortiranja)
 - vraćaju redni broj pozicije torke unutar grupe
 - ukoliko n ($n > 1$) torki ima istu poziciju po rangiranju u grupi, svakoj od n torki dodeljuje se ista pozicija r i
 - RANK prvoj sledećoj torki dodeljuje poziciju $r + n$
 - tzv. rangiranje sa preskočenim pozicijama (gap ranking)
 - DENSE_RANK prvoj sledećoj torki dodeljuje poziciju $r + 1$
 - tzv. gusto rangiranje
 - za rangiranje uvek je neophodno izvršiti analizu svih analitičkih vrednosti \Rightarrow holistička funkcija

Analitičke funkcije i upiti

- **RANK() i DENSE_RANK()**

```
SELECT EmpID, Salary, DeptID,  
        RANK() OVER W AS GapRank,  
        DENSE_RANK() OVER W AS DenseRank  
FROM Employees  
WINDOW W AS ( PARTITION BY DeptID  
              ORDER BY Salary DESC NULLS FIRST)  
ORDER BY DeptID, Salary DESC
```

- rangiranje radnika po platama unutar svakog departmana, u opadajućem redosledu

Analitičke funkcije i upiti

- RANK() i DENSE_RANK()**

```
SELECT EmpID, Salary, DeptID,  
       RANK() OVER W AS GapRank,  
       DENSE_RANK() OVER W AS DenseRank  
FROM Employees ...
```

<i>EmpID</i>	<i>Salary</i>	<i>DeptID</i>	<i>GapRank</i>	<i>DenseRank</i>
7300	4000,00	100	1	1
7301	3000,00	100	2	2
7400	4200,00	110	1	1
7422	4200,00	110	1	1
7473	3800,00	110	3	2
7800	7800,00	120	1	1
7801	7800,00	120	1	1
7805	6600,00	120	3	2

Analitičke funkcije i upiti

- **RANK() i DENSE_RANK()**

- Alternativna sintaksa – skupovni oblik (Oracle)

```
[DENSE_]RANK(expr1, ..., exprn)  
                WITHIN GROUP (ORDER BY oexpr1, ..., oexprn)
```

- Upit

- selektovati poziciju radnika sa salary=1000 AND bonus=500 u tabeli Employees

```
SELECT RANK(1000, 500)  
                WITHIN GROUP (ORDER BY salary, bonus)  
FROM Employees
```

Analitičke funkcije i upiti

- **[LEAD | LAG](sql_expr, offset, default)**
 - izračunavanje izraza
 - nad zadatom narednom (LEAD) ili
 - nad zadatom prethodnom (LAG) torkom i
 - pridruživanje rezultata tekućoj torci
 - <sql_expr> - izraz čiju vrednost treba izračunati
 - <offset>
 - relativni nenegativni redni broj torke, u odnosu na tekuću
 - 0 – označava tekuću torku
 - LEAD – relativni broj naredne torke u odnosu na tekuću
 - LAG – relativni broj prethodne torke u odnosu na tekuću
 - <default> - zamenska vrednost za <sql_expr>
 - ukoliko <offset> "ispadne" izvan opsega prozora

Analitičke funkcije i upiti

- **[LEAD | LAG](sql_expr, offset, default)**

```
SELECT EmpID, Salary, DeptID,  
        LEAD(Salary, 1, 0) OVER W AS NextSal,  
        LAG(Salary, 1, 0) OVER W AS PrevSal  
FROM Employees  
WINDOW W AS ( PARTITION BY DeptID  
              ORDER BY Salary DESC NULLS LAST)  
ORDER BY DeptID, Salary DESC
```

- evidencija radnika po platama, u opadajućem redosledu unutar departmana
- sa prikazanom platom prvog narednog i prvog prethodnog radnika u istom departmanu

Analitičke funkcije i upiti

- [LEAD | LAG](sql_expr, offset, default)**

SELECT EmpID, Salary, DeptID,

LEAD(Salary, 1, 0) OVER W AS NextSal,

LAG(Salary, 1, 0) OVER W AS PrevSal

FROM Employees ...

<i>EmpID</i>	<i>Salary</i>	<i>DeptID</i>	<i>NextSal</i>	<i>PrevSal</i>
7300	4000,00	100	3000,00	0,00
7301	3000,00	100	0,00	4000,00
7400	4200,00	110	4000,00	0,00
7422	4000,00	110	3800,00	4200,00
7473	3800,00	110	0,00	4000,00
7800	7800,00	120	7800,00	0,00
7801	7800,00	120	6600,00	7800,00
7805	6600,00	120	0,00	7800,00

Analitičke funkcije i upiti

- **[FIRST | LAST]_VALUE(sql_expr)**
 - izračunavanje izraza
 - nad prvom torkom prozora (FIRST) ili
 - nad poslednjom torkom prozora (LAST)
 - poštuje se redosled torke, definisan sa ORDER BY klauzulom analitičke funkcije
 - pridruživanje rezultata tekućoj torci
 - <sql_expr> - izraz čiju vrednost treba izračunati

Analitičke funkcije i upiti

- **[FIRST | LAST]_VALUE(sql_expr)**

```
SELECT EmpID, Salary, DeptID,  
       Salary - FIRST_VALUE(Salary) OVER W AS GapMinSal,  
       LAST_VALUE(Salary) – Salary OVER W AS GapMaxSal  
FROM Employees  
WINDOW W AS ( PARTITION BY DeptID  
              ORDER BY Salary ASC NULLS FIRST)  
ORDER BY DeptID, Salary ASC
```

- evidencija radnika po platama, u rastućem redosledu unutar departmana
- sa prikazanom razlikom u odnosu na najmanju i najveću platu u okviru istog departmana

Analitičke funkcije i upiti

- **[FIRST | LAST]_VALUE(sql_expr)**

```
SELECT EmpID, Salary, DeptID,
```

```
Salary - FIRST_VALUE(Salary) OVER W AS GapMinSal,  
LAST_VALUE(Salary) – Salary OVER W AS GapMaxSal
```

```
FROM Employees ...
```

<i>EmpID</i>	<i>Salary</i>	<i>DeptID</i>	<i>GapMinSal</i>	<i>GapMaxSal</i>
7301	3000,00	100	0,00	1000,00
7300	4000,00	100	1000,00	0,00
7473	3800,00	110	0,00	400,00
7422	4000,00	110	200,00	200,00
7400	4200,00	110	400,00	0,00
7805	6600,00	120	0,00	1200,00
7801	7800,00	120	1200,00	0,00
7800	7800,00	120	1200,00	0,00

Sadržaj

- Osnovne karakteristike OLAP sistema
- Arhitektura OLAP sistema
- Dimenzione ROLAP strukture
- Osnovni OLAP upiti
- Agregacione funkcije
- Unakrsno tabeliranje
- Analitičke funkcije i upiti

Pitanja i komentari





Kraj prezentacije

Sistemi za analizu podataka

*On-line Analytical Processing
(OLAP)*