

SOFTVERSKO INŽENJERSTVO
ZA SISTEME BAZA PODATAKA
ODGOVORI NA PITANJA ZA
USMENI DEO ISPITA
J A N U A R 2 0 2 3

Git: <https://github.com/NikolaVetnic/seds>

01

DW sistemi - motivacija nastanka, koncepcija.

Motivacija nastanka DW sistema: ulogu igraju motivacioni faktori poslovanja, kao i uloga inf. sistema u podršci motivacionih faktora (kvalitetne informacije o poslovanju, podrška upravljanju). Podrška poslovanju se pruža upravljačkim informacionim sistemima - *MIS*, i kasnije sistemima za podršku odlučivanju - *DSS*.

Management Information Systems (MIS) uvedeni sa ciljem obezbeđivanja operativnih podataka o (svakodnevnom) poslovanju, pružaju mogućnost kreiranja izveštaja na zahtev ali su skromnih mogućnosti u smislu podrške analizi.

On-Line Transaction Processing (OLTP) sistemi pripadaju *MIS* sistemima - projektovani su da obezbede performanse pri transakcijama, rukuju kompleksno struktuiranim podacima, ali sa nedovoljno dobro iskazanom semantikom i visokom disperzijom istih, te su nepogodni za realizaciju zahtevnih upita. Što se karakteristika tiče, **1)** tipične su operacije ažuriranja, **2)** transakcije rukuju sa malo podataka, **3)** osnovna jedinica je slog, **4)** strukture su statičke. Mogućnosti analize *MIS* sistema se svode na analizu operativnih podataka, ili korišćenja izveštaja koji se generišu nad njima - reč je o primitivnom rešenju, odnosno obradi posebno izdvojenih podataka (izvodi i "eksplozija" izvoda, nedostatak opšte vremenske odrednice, neusaglašenost, i slično).

Decision Support Systems (DSS) su savremeno i bolje rešenje budući da obezbeđuju softversku podršku analizi podataka. Kao ulaz podržava istorijske operativne podatke (drastično različitih formata) kao ulaz, podržava matematičke modele analize poslovanja, a proizvodi informacije bitne za proces odlučivanja. Opšta struktura *DSS*-a podrazumeva **1)** komponente za upravljanje podacima (skladištenje, ekstrahovanje i filtriranje, alati za upite), **2)** za upravljanje modelima, i **3)** za prezentaciju podataka.

Koncepcija DW sistema: *DW* je posledica potrebe za skladištenjem i obradom velike količine podataka, i kao takav predstavlja osnovni element nove generacije *DSS*-ova. "*DW* je na temu orijentisana, integrisana, od vremena zavisna i nepromenljiva kolekcija podataka u službi podrške procesu donošenja odluka." (*Bill Inmon*)

DW sistemi su dakle orijentisani na teme (npr. prodaja, marketing, proizvodnja) - podaci se ne kategorišu po funkcionalnim celinama kao u *OLTP* sistemima, pri čemu jedna tema može biti zanimljiva višestrukim funkcijama, a takođe različite teme mogu deliti podatke.

DW podaci su integrisani, odnosno svi podaci o entitetu su standardizovani i čuvaju se na jednom mestu, pa tako *DW* predstavlja centralizovanu bazu.

DW podaci su vremenski zavisni, odnosno *DW* sadrži vreme, a ponekad i prostornu dimenziju, u "granuliranom" vidu. Podaci se mogu ticati kako prošlosti i sadašnjosti, tako i budućnosti (predikcije). Samo *DW* se periodično ažurira.

DW podaci su nepromenljivi, odnosno unosi se retko brišu ili modifikuju.

02

DW sistemi - tematske karakteristike, poređenje DW i OLTP.

Tematske karakteristike DW sistema: tematske karakteristike obuhvataju dimenzionost, granularnost i diskretizaciju vremena.

Dimenzionost podrazumeva iskazivanje podataka uskladištenih u *DW* spram različitih konteksta (dimenzija) koji ih opisuju, dok se sami podaci dele na **1)** činjenične (vrednosti parametara promenljive u vremenu) i **2)** dimenzione (konstantne dimenzione karakteristike).

Granularnost podrazumeva različite nivoe na kojima se vrši agregacija podataka pri ažuriranju *DW*-a (npr. da li se ukupna zarada sabira na nivou dana ili nedelje). Ključna odluka u ovom slučaju je nivo granularnosti koji mora odgovarati potrebama poslovanja.

Diskretizacija vremena podrazumeva podelu vremenske komponente poslovanja na diskretne jedinice sa kojima se povezuju relevantni unosi *DW* baze podataka. Za razliku od *OLTP* sistema podaci u *DW* bazama imaju daleko duži životni vek, nikad ne postaju relevantni te se zato i ne brišu.

	OLTP	DW
tipične operacije	operacije ažuriranje	operacije upita
kritične operacije	transakcije ažuriranja	transakcije upita
ažuriranje baze podataka	veliki broj DML operacija	punjenje i periodično osvežavanje
frekvencija upita	niska/srednja	visoka
kompleksnost upita	niska	visoka
količina podataka po transakciji	mala/srednja	velika
očekivano vreme odgovora	do sekunde	nekoliko sekundi do više sati

Poređenje *DW* i *OLTP* sistema - transakcione karakteristike.

	OLTP	DW
vremenska diskretizacija	dan - sekunda	dan - godina
aktuelnost podataka	do godine	više godina
obim baze podataka	MB - GB	GB - TB
povećanje obima baze podataka	linearno	polinomno / eksponencijalno
granularnost podataka	elementarni podaci	agregirani podaci
nivo agregacije	nizak	visok
šema baze podataka	normalizovana, kompleksna	denormalizovana, prostija

Poređenje *DW* i *OLTP* sistema - karakteristike podataka.

	OLTP	DW
izvori podataka	operativno poslovanje	OLTP, interni i eksterni izvori
organizacija podataka	prema funkcijama	prema temama
podrška poslovnih procesa	operativno poslovanje	analiza i odlučivanje
forme za prikaz podataka	statičke, retko promenljive	kontekstno zavisne, promenljive
intenzitet korišćenja BP u vremenu	uniforman	neuniforman, mogući udarni termini

Poređenje *DW* i *OLTP* sistema - karakteristike poslovanja.

03

DW sistemi - arhitektura, vrste.

Arhitektura *DW* sistema: višeslojna je i obuhvata: **1)** servere, **2)** međupodručja *DW* sistema (mesto operativne pripreme za upis u *DW* sistem), **3)** *DW* aplikativne servere (upiti, analiza, prezentacija), **4)** *DW* klijente (UI za pristup *DW*-u), **5)** repozitorijum metapodataka.

Vrste *DW* sistema - prema opsegu pokrivenosti tema: razlikujemo *Enterprise Data Warehouse* (*EDW*) koji pokriva celo poslovanje, i *Data Mart* (*DM*) koji pokriva obično samo jednu temu.

EDW predstavlja korporativni *DW* i pokriva celokupno poslovanje. Razvija se inkrementalno, predstavlja jedan izvor podataka za celokupni menadžment, sinhronizuje se sa svih izvora i može biti osnova za pojedinačne *DM* sisteme.

DM predstavlja tematski *DW* i pokriva jednu temu poslovanja. Može se predstaviti kao pilot ozbiljnijeg *DW* sistema, a realizuje se kao (od drugih sistema) nezavisni ili zavisni *DM*.

Virtuelna *DW* arhitektura: ne postoji posebna *DW* baza; obezbeđen je direktan pristup OLTP sistemu uz korišćenje klijentskih alata za upite i analizu podataka; privremeno i kratkotrajno rešenje. Prednosti: malo ulaganje u IT opremu i ljudstvo, jednostavnost uvođenja i korišćenja. Mane: ne postoje istorijski ni agregirani podaci, kao ni centralizovani meta podaci niti procedure za čišćenje i integraciju.

Upakovana *DW* arhitektura: "upakovani" softverski proizvod koji obezbeđuje **1**) izgradnju *DW* iz različitih izvora, **2**) pristup *DW* jednostavnim alatima za upite i analizu, **3**) izgradnju lokalnog MDR - *Managed Detection & Response*. Prednosti: eliminiše direktnu komunikaciju sa OLTP bazom. Mane: eksplozija nezavisnih i neintegrisanih *DM*-ova, odsustvo zajedničke osnove meta podataka na nivou organizacije i tema poslovanja, prljavi podaci iz različitih izvora.

Arhitektura izvedenog *DW*: upakovana *DW* arhitektura sa jendim ECTL soft. paketom i DSA međupodručjem koje omogućava **1**) ekstrakciju, pročišćavanje, transformaciju i punjenje, **2**) agregiranje i sumiranje, **3**) kreiranje i održavanje centralnog MDR, **4**) administratorske f-je *DW* sistema, **5**) interfejs prema alatu za modelovanje *DW* baze. Prednosti: jedinstveni ECTL softver, centralizovan MDR. Mane: slaba integrisanost CMDR i lokalnih MDR, nezavisni i neintegrisani *DM*, podrška organizacionih potreba na invou jedne teme ili tipa radnog mesta.

Arhitektura povezanog *DW*: izvedeni *DM* sa softverom za razmenu meta podataka - usaglašenost CMDR i lokalnih MDR, CMDR je centralno mesto znanja u *DW* sistemu.

Arhitektura sa centralnim *DW*: jedinstven i integrisan *DW* sistem; čuva analitičke podatke, predstavlja izvor usaglašenih podataka cele organizacije; koriste se za upite i izveštaje nad analitičkim (atomičnim) podacima; odvojen je od OLTP baze i predstavlja osnovu za formiranje i osvežavanje baza povezanih *DM*-ova.

Arhitektura korporativnih *DW*: više izvora, "off-the-shell" ECTL paket, CMDR, meta data exchange, central data warehouse, povezani *DM*-ovi, centralno upravljanje, alati za upite i analizu.

04

DW sistemi - razvoj (strategija, razvoj inkrementa, pilot projekat).

Razvoj *DW* sistema: pri razvoju *DW* sistema primenjuje se opšti model razvoja softvera zasnovan na životnom ciklusu kojeg čine dve grupe faza: **1**) strategija, analize i projektovanja, i **2**) programiranje, uvođenje u upotrebu, korišćenje i održavanje. Prepoznaju se takođe dva metoda: **1**) iterativni (spiralni), koji karakteriše *top-down* pristup od opštog ka detaljno kroz nekoliko iteracija, i **2**) inkrementalni *bottom-up* pristup, gde se softver razvija *DM* po *DM* i postupno integriše i testira - često je u praksi reč o kombinaciji dva metoda.

Strategija: pri razvoju *DW* sistema strategija obezbeđuje *top-down* analizu organizacije i zahteva za *DW*, poštovanje potreba i zahteva korisnika, koncepciju dugoročnog razvoja korporativnog *DW*-a (u smislu da *EDW* neće degradirati u virtualni, upakovani ili *stovepipe DW*). Postavljeni zadaci uključuju **1**) identifikaciju i analizu ciljeva, faktora uspeha, indikatora ostvarenja, ograničenja i problema u poslovanju, **2**) identifikaciju i analizu ciljeva, korisničkih zahteva i oblasti pokrivenosti poslovanja *DW* sistemom, **3**) projektovanje koncepcije

arhitekture i dugoročnog razvoja *DW* sistema, **4)** identifikaciju i analizu izvora podataka, **5)** analizu neusaglašenosti zahteva i mogućnosti izvora podataka, **6)** strateško opredeljenje metoda i tehnoloških osnova, **7)** definisanje strategije unapređenja znanja, **8)** izradu plana razvoja *DW* sistema, **9)** procenu cene i isplativosti.

Razvoj inkrementa: razvoj pojedinačnog *DM*-a nudi **1)** fazni *bottom-up* razvoj koji se uklapa u strategiju razvoja ukupnog *EDW*-a, **2)** id-aciju ciljeva i potreba korisnika jedne teme, **3)** id-aciju (ne)funkcionalnih zahteva *DM*-a, **4)** id-aciju izvora podataka za *DM*, **5)** izbor ili razvoj *DM* sa lokalnim *MDR* i *MDE* softverom, **6)** izbor i uspostavljanje arhitekture *DM*, **7)** pristup podacima putem *web* tehnologija.

Pilot projekat: obezbeđuje **1)** razvoj "karakterističnog" prvoizabranog *DM* sistema (dovoljno složenog, bliskog korisnicima po f-jama i koncepciji, pogodnog za brz razvoj sa niskim rizikom, sa ograničenim trajanjem), **2)** testiranje strategije i koncepcije razvoja *EDW* na pilot rešenju, **3)** korekciju inicijalne strategije, **4)** dodatnu motivaciju za uključivanje novih korisnika.

05

Projektovanje *DW* sistema, analiza i specifikacija korisničkih zahteva.

Analiza i specifikacija korisničkih zahteva: na početku je važno shvatiti probleme u poslovanju i pitanja na koje klijent želi da odgovori - iz tog razloga, prvi korak je upoznavanje sa načinom poslovanja klijenta (id-acija poslovnih procesa), nakon čega sledi upoznavanje sa postojećim inf. sistemom.

Identifikacija poslovnih procesa najčešće se izvodi putem ankete i intervjuisanja klijenta, čime se analizira hijerarhijska organizacija, kao i planiranje trajanja procesa prikupljanja korisničkih zahteva. Nakon svakog intervjua izvodi se sumiranje i definicija kriterijuma uspeha pri realizaciji poslovnih zahteva intervjuisane osobe. Intervjui se grupišu u srodne celine koje odgovaraju poslovnim procesima kompanije.

Specifikacija poslovnog modela podrazumeva **1)** identifikaciju i specifikiranje (ciljeva i kritičnih faktora uspeha, procesa poslovanja i tema, nadležnosti, pravila i ograničenja, metrike, i drugih), **2)** ulazne specifikacije i izvore informacija (intervjui, dokumentacija, znanja, literatura), **3)** rezultate bitne za projektovanje logičkog modela baze, **4)** specifikaciju zahtevanih dimenzija i mera.

06

Pogledi, materijalizovani pogledi, agregacija podataka.

Pogledi u *DW* sistemima: SQL pogled predstavlja virtuelnu tabelu čiji se sadržaj generiše upitom, a može se koristiti u drugim upitima kao i za definisanje novih pogleda. Pogledi se koriste u *DSS*-ovima kako bi se omogućilo skoncentrisanje samo na potrebne podatke, dok sa druge strane sami upiti u *DSS*-ovima često zahtevaju agregirane podatke izvedene iz tabela činjenica.

Materijalizovani pogledi: reč je o tabeli ili skupu tabela sa agregiranim podacima čiji se sadržaj generiše na osnovu sadržaja baznih tabela *DW* baze; čak se i sama *DW* baza može posmatrati kao materijalizovan pogled, i to nad *OLTP* sistemima i spoljnim izvorima podataka. Pri kreiranju ovakvih pogleda uzima se u obzir koliko vrsta upita može biti

pokriveno pogledom, frekvencija pokretanja pokrivenih upita, očekivano poboljšanje performansi, prostorna zahtevnost, usložnjavanje osvežavanje *DW* baze, i drugo.

SQL materijalizovani pogled predstavlja vrstu bazne tabele koja se formira i ažurira (što se dešava posredno - osvežavanjem) preuzimanjem (agregiranjem) podataka iz jedne ili više drugih tabela. Ovakav pogled može biti indeksiran, dok osvežavanje sadržaja može biti trenutno (pri izvođenju transakcija nad originalnim podacima) ili odloženo, i inkrementalno ("brzo") ili kompletno (reinicijalizacija čitavog sadržaja materijalizovanog pogleda).

Agregacija podataka: podrazumevaju sumarne podatke po zadatim dimenzijama; ovakvi podaci su redundantni ali neophodni za efikasnu podršku različitih upita i analiza. Čuvaju se u *DW* bazi, a generišu se (izračunavaju) u ECTL procesima na osnovu formiranih činjeničnih podataka, i često uz primenu skupovnih f-ja (SUM, COUNT, MIN, MAX, AVG, STDDEV). Nivoi agregacije uključuju bilo koju kombinaciju dimenzija i bilo koji nivo u hijerarhijskoj strukturi dimenzije.

Projektovanje *DW* šeme baze podataka sa agregiranim podacima podrazumeva specifikaciju **1)** atributa agregiranih podataka, **2)** nivoa agregacije, **3)** algoritma za agregiranje podataka, **4)** načina izračunavanja i memorisanja agregiranih vrednosti, **5)** implementaciju agregacije u *DW* šemu i ECTL procese, **6)** optimizaciju upita sa agregiranim podacima i **7)** praćenje upotrebe agregiranih podataka.

07

Vrste materijalizacije pogleda.

Vrste materijalizacije pogleda: Moguće vrste materijalizacije pogleda obuhvataju **1)** kroz program, **2)** izvedeni pogled, **3)** kroz podatke, **4)** kroz indeks, **5)** kombinaciju podataka i indeksa, i **6)** agregirani pogled.

Materijalizacija kroz program (*pure program*) - SQL definicija pogleda u samom programu, upit se izvršava svaki put na zahtev korisnika.

Izvedeni pogled (*derived data view*) - izvedeni podaci se generišu upitom i kreiranjem materijalizovanog pogleda koji se koristi u daljim upitima i trenutno osvežava. Omogućena je optimizacija upita, dok očuvanje konzistencije podataka materijalizovanog pogleda zahteva posebno procesorsko vreme.

Materijalizacija kroz podatke (*pure data view*) - putem materijalizovanog pogleda bez uključenog osvežavanja (presek stanja - *snapshot*), ili sa periodičnim obnavljanjem. Očuvanje konzistencije podataka zahteva posebno procesorsko vreme.

Materijalizacija kroz indeks (*pure index*) - indeks i sam predstava vrstu materijalizovanog pogleda (sadrži replicirane vrednosti atributa sa adresama torki iz baznih tabela, osvežava se trenutno prilikom ažuriranja); koristi se za poboljšanje performansi upita sa uslovom selekcije koja obuhvata indeksirane attribute, upita sa operacijom spajanja, kao i kompletne realizacije određenih upita nad indeksom.

Kombinacija podataka i indeksa (*hybrid data and index*) - kombinacija "izvedenog pogleda" i "kroz indeks"; budući da se često materijalizuju zahtevane vrednosti atributa, očekuje se postizanje boljih performansi upita nad tako pripremljenim atributima (projektovani sadržaj tabele zauzima manje prostora).

Agregirani pogled (*aggregate view*) - reč je o najširem shvatanju agregacije podataka, obuhvata sledeće slučajeve: celu DW bazu nad zvezdastom šemom (ili nekim drugim oblikom šeme), oblik materijalizacije "izvedeni pogled" i oblik "kroz podatke". Klasifikacija: **1**) *join aggregate view*, **2**) *single table aggregate view* i **3**) *join only aggregate view*.

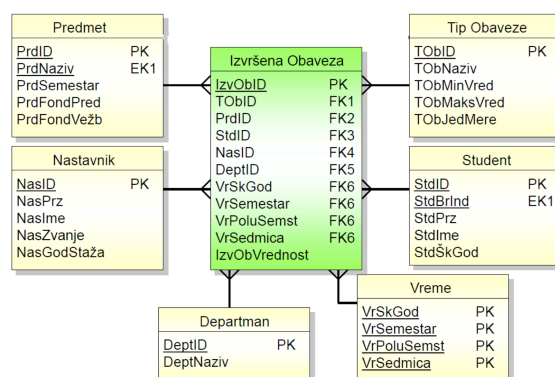
Sam izbor pogleda za materijalizaciju je često kompleksan problem budući da postoji širok spektar upita i pogleda koji bi se mogli koristiti u vezi sa tim upitima. S druge strane, uvođenje materijalizovanih pogleda ima uticaj na povećanje memorijskog prostora i produženje vremena za održavanje DW baze podataka. Cilj je izabrati mali ali pogodan skup pogleda koji će podržati najveći broj upita označenih kao značajni.

01

Strukture činjeničnih i dimenzionih podataka, hijerarhije dimenzija.

Zvezdasta šema, šema tipa pahuljice i sazvežđa, šema sa agregiranim podacima.

Zvezdasta šema: naziva se i *star* šema, a sastoji se od struktura činjeničnih podataka (tipovi entiteta / šeme relacija činjenica, sa primarnim ključem, atributima i domenima mera) i dimenzionih podataka (tipovi entiteta / šeme relacija dimenzija, sa skupovima ključeva i primarnim ključevima, atributima i hijerarhijama dimenzija). Karakterišu je **1**) lako razumljiva struktura, **2**) mogućnost sprovođenja multidimenzionalnih analiza, **3**) podrška od strane velikog broja alata za izveštavanje i analizu. Uslovi normalizacije: **1**) očekivano zadovoljava uslov 1NF, **2**) šeme relacija činjenica najčešće zadovoljavaju BCNF, **3**) šeme relacija dimenzija mogu ali često ne zadovoljavaju uslove za NF već od 1NF.



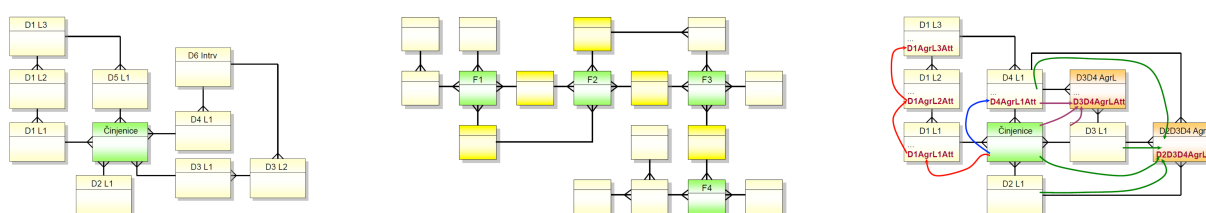
Slika 1: Zvezdasta šema.

Strukture činjeničnih podataka: šema relacije tabela činjenica naziva se i tabela detaljnih podataka; nalazi se uvek u centru *star* šeme i referencira tabele dimenzija putem stranih ključeva koji predstavljaju primarne ključeve dimenzija. Sadrži attribute mera koji predstavljaju numeričke podatke, a često je reč o agregiranim (sumarnim) podacima. Relacija može sadržati velik broj torki, količina podataka brzo raste i tipično je aditivna. Uobičajno sadrži jedan - primarni ključ, gde je reč o veštačkom ključu (nova vrednost se generiše inkrementacijom prethodne). Razlikuju se dva tipa: **1**) *regular fact table* - sa uključenim merama (figuriraju atributi koji reprezentuju mere - vrednosti činjenica), i **2**) *factless fact table* - bez uključenih mera (služi praćenju događaja - ukoliko torka postoji događaj se desio, inače nije, pri čemu upiti najčešće koriste COUNT f-ju). Vrste mera činjenica (prema mogućnosti sumiranja po dimenzijama): aditivne, semiaditivne i neaditivne mere. Vrste mera činjenica (prema izvoru generisanja vrednosti): osnovne, izvedene mere činjenica.

Strukture dimenzionih podataka: reprezentuju kontekst činjenica i povezane su sa šemama relacija istih; sadrže attribute koji opisuju samu dimenziju ili reprezentuju agregirane numeričke podatke po dimenziji. Relacija dimenzije može biti različitog obima. Može sadržati više ekvivalentnih ključeva, ali se samo jedan bira za primarni koji po strukturi može biti veštački (preferabilno) ili prirodni. Na izbor ključa utiču između ostalog i **1**) granularnost (savetuje se izbor za nivo finije granularnosti od potreba), **2**) nejednaka jedinična dimenzija u

različitim izvorima podataka (nemogućnost lakog "poravnavanja" granularnosti dimenzije na željeni nivo), **3**) različiti sistemi označavanja (problem pojave istih entiteta pod različitim vrednostima ključa), **4**) mogućnost modifikacije podataka dimenzije (podela, spajanje, modifikacija - savetuje se ublažavanje problema finijom granulacijom).

Hijerarhija dimenzije: višenivovska, hijerarhijska struktura dimenzionih podataka - odnos između tipova entiteta organizovan po nivoima hijerarhije 1:N, definisan nizom funkcionalnih zavisnosti među atributima dimenzije (npr. država-region-grad). Načini modelovanja obuhvataju **1**) normalizovane strukture (jedan nivo - jedna šema relacije), **2**) delimično normalizovane strukture (više nivoa - jedna šema relacije), **3**) potpuno denormalizovane strukture (svi nivoi - jedna šema relacije). Hijerarhija dimenzija se koristi za analizu činjenica na različitim nivoima agregacije (*drill down / up*), po različitim hijerarhijama i za obezbeđenje optimizacije upita.



Slika 2: Šema pahuljice, sazvežđa i sa agregiranim podacima.

Šema tipa pahuljice: *snowflake* šema je varijanta zvezdaste šeme gde se javljaju hijerarhije dimenzija putem normalizovane strukture. Prednosti: izbegavanje logičkih problema usled denormalizovanih tabela, struktura eksplicitno iskazuje hijerarhije i moguće nivoe agregacije. Mane: problem performansi upita.

Šema tipa sazvežđa: reč je o kombinaciji više zvezdastih šema koje dele zajedničke dimenzije.

Šema sa agregiranim podacima: obuhvata sumarne podatke po zadatim dimenzijama - redundantni su ali neophodni za efikasnu podršku različitih upita i analiza, izračunavaju se u ECTL procesima. Agregacija je moguća po bilo kojoj kombinaciji dimenzija i u bilo kom nivou hijerarhijske strukture dimenzije.

02

ECTL procesi - ekstrakcija i transformacija podataka.

E(C)TL proces: **1**) *extraction* (selektovanje podataka iz različitih izvora), **2**) *cleaning & transformation* (validacija, pročišćavanje, integracija, vremensko označavanje), **3**) *loading* (punjenje). Postoje različita rešenja za podršku ECTL procesa, među kojima su upotreba gotovih softverskih paketa, ili sopstvenih softverskih rešenja. Tehnološke osnove za ETL uključuju direktnu upotrebu 3GL programskih jezika (C, C++, Java, itd.), različitih *utility* softverskih alata (sa funkcionalnostima *Export*, *Import*, *Load*), naprednih mogućnosti SQL jezika (upiti tipa `CREATE TABLE ... AS SELECT ...`), i konačno posrednika (*Gateway* interfejsi).

Ekstrakcija podataka: obuhvata selekciju i preuzimanje izvornih podataka iz različitih izvora u koje spadaju produkcionni podaci (*OLTP* sistemi), arhivski podaci, interni izvori podataka (*XML*, *DOCX* i slični fajlovi) i eksterni izvori podataka. Sama ekstrakcija može biti potpuna ili

inkrementalna (samo podaci koji su novi u odnosu na poslednji unos), odnosno u radnom režimu (*online*) ili van njega (*offline*).

Transformacija podataka: podrazumeva validaciju, pročišćavanje, integraciju i vremensko označavanje podataka - često najzahtevniji i najkompleksniji deo *ECTL* procesa. Realizuje se u okviru *Data Staging Area (DSA)* i ima ključni uticaj na obezbeđenje kvaliteta podataka u *DW* sistemu. Samo *DSA* područje može biti pozicionirano zajedno sa *OLTP* serverom (*on-site staging*), na posebnom serveru (*standalone remote staging*) i zajedno sa *DW* serverom (*remote staging*). Pozicioniranje postupka transformacije može biti paralelno sa ekstrakcijom, posle nje ali pre punjenja *DW*-a, paralelno sa punjenjem i kombinacijom navedenog.

Greške na koje se nailazi u ovoj fazi najčešće se tiču: **1**) ograničenja ključa, **2**) drugih ograničenja, **3**) pojave atributa sinonima ili homonima, **4**) ograničenja domena istih atributa, **5**) istih vrednosti atributa, **6**) ograničenja ključa za istu klasu realnih objekata, **7**) izostavljenih podataka. pristupi u rešavanju problema: **1**) tolerisati greške (preuzimanje "prljavih" podataka), **2**) ignorisati greške ("prljavi" podaci se ne ubacuju u *DW*), **3**) suštinsko rešavanje grešaka i neusaglašenosti (najteže, ali vodi ka kvalitetnom *DW*-u).

Nakon prečišćavanja podaci se transformišu, što obično uključuje **1**) opremanje podataka vremenskom dimenzijom i svođenje na istu vremensku osu, **2**) spajanje različito identifikovanih torki u jednu (kao posledica grublje granularnosti npr.), **3**) razdvajanje jedne torke na više različitih (kao posledica finije granularnosti npr.), **4**) integracija podataka (spajanje isto identifikovanih torki u jednu).

Alati za prečišćavanje i transformaciju podataka su zasnovani na navedenim tehnološkim osnovanama *ETL* procesa i mogu predstavljati ugrađenje funkcionalnosti u druge alate za *ETL* proces: **1**) *data migration tools* - primena jednostavnih pravila za transformaciju podataka, **2**) *data scrubbing tools* - sofisticiraniji alati koji implementiraju i primenjuju različita pravila iz domena primene, i **3**) *data auditing tools* - primenjuju tehnike rudarenja podataka u pronalaženju netipičnih uzoraka podataka.

03

ECTL procesi - punjenje i održavanje podataka.

Punjenje podataka: inicijalno punjenje predstavlja jednokratnu proceduru preuzimanja istorijskih podataka i početnog formiranja *DW* baze (i dimenzija i činjenica); obuhvata veliku količinu podataka i zahteva potpunu ekstrakciju i posebne procedure prečišćavanja podataka, kao i kompleksnu obradu u pred- ili postprocesiranju, te zato može dugo da traje.

Sa stanovišta preuzetih i transformisanih podataka, punjenje može biti **1**) potpuno - uvek se pripremaju i pune kompletni podaci u *DW* što može biti vremenski i resorno zahtevan pristup, ili **2**) inkrementalno - pripremaju se i pune samo novopridodati podaci što je naravno manje zahtevno. Sa stanovišta pristupa *DW* sistemu punjenje može biti van radnog vremena (*offline*) ili *online*.

Alati za punjenje *DW*-ova su zasnovani na navedenim odlikama *ETL* procesa i bazirani su na iskorišćavanju arhitektura i tehnika paralelizacije i vremenskog smicanja (eng. *pipelining*) operacija punjenja, budući da je često potrebno obraditi veliku količinu podataka u kratkom vremenu. Samo punjenje često prate sortiranje, agregacija, indeksiranje, kreiranje

materijalizovanih pogleda, kao i izvršavanje različitih administrativnih zadataka (arhiviranje, osvežavanje meta podataka, publikovanje).

Za punjenje je vezan i termin *load window*, što označava minimalno vreme potrebno za realizaciju svih zadataka punjenja podataka. Nastoji se da ono bude što kraće te je od važnosti izbor trenutka realizacije pratećih zadataka - npr. ne savetuje se generisanje podataka (npr. indeksa) paralelano za punjenjem jer znatno produžava trajanje procesa.

Osvežavanje podataka: podrazumeva regularno ponovljivu proceduru preuzimanja novih podataka i osvežavanje *DW*-a, te samim tim obuhvata manju količinu podataka nego pri inicijalnom punjenju. Zahteva se inkrementalna ekstrakcija i jednostavnije procedure prečišćavanja, ali mora trajati relativno kratko i izvršava se tokom noći).

Osnovni načini realizacije su **1)** propagacija ažuriranja - korišćenje mehanizama za osvežavanje repliciranih podataka (putem *trigger*-a baze podataka, materijalizovanih pogleda i replikacije, ili replikacije zasnovane na tokovima podataka), i **2)** inkrementalno punjenje - preuzimanje inkremenata iz izvora i punjenjem *DW*-a (zamenom kompletnih podataka, poređenjem instanci, sa vremenski obeleženim razlikama, ili upotrebom transakcionog dnevnika). U oba slučaja izvor može biti *OLTP* ili *DSA* područje i u oba slučaja se nastoji da se osvežavanje lokalizuje samo na podatke koji su predmet izmena.

Osvežavanje karakterišu **1)** *load window* (ograničenje na maksimalno trajanje osvežavanja), **2)** frekvencija i obim podataka za osvežavanje (periodičnost sprovođenja osvežavanja - manja količina podataka, kraći *load window*, bolja ažurnost u *DW*), **3)** pristup detektovanju promena na izvoru i **4)** način i procedura osvežavanja.

Održavanje meta podataka: meta podaci se kreiraju u procesu razvoja *DW* sistema i predstavlja projektnu dokumentaciju pohranjenu u repozitorijumu (u struktuiranom vidu). Održavanje se vrši u *ETL* procesu, a administriranje podrazumeva definisanje i uspostavu odgovornosti nad podacima i procedurama.

04

Osnovne karakteristike OLAP sistema.

Definicija OLAP sistema: ovi sistemi se bave neposrednom analitičkom obradom podataka - reč je o multidimenzionalnoj obradi, budući da se primenjuje "rotiranje" i "detaljizacija" u prikazu do zahtevanog nivoa, zatim kreiranje i interaktivno istraživanje agregiranih podataka, sprovođenje analiza trendova i izuzetaka, i konačno izvršavanje analitičkih modela (predviđanja i regresione analize). *OLAP* transakcije predstavljaju pretežno ad hoc upite takve da mogu da obuhvate i nekoliko hiljada torki, i da zahtevaju spajanje torki i primenu f-ja agregacije - zbog toga je poželjno da se *OLAP* transakcije izvode nad *DW* bazom podataka.

Osnovne karakteristike: *OLAP* sisteme karakterišu **1)** multidimenzione strukture podataka, **2)** primena multidimenzionih tehnika analize, **3)** repozitorijum meta podataka, **4)** UI prilagođen znanju i mogućnostima korisnika, **5)** primena višeslojnih i distribuiranih K/S arhitektura.

Multidimenzione strukture podataka su odlika svakog *OLAP* sistema, a poželjne su mogućnosti kreiranja neograničenog broja dimenzija, nivoa agregacije, analitičkih f-ja nad tim strukturama, mogućnost dinamičkog upravljanja retkim matricama i fleksibilnost u oblikovanju različitih struktura. Opšti pristupi realizaciji su putem hiperkočke (gde se svaka

dimenzija podataka pridružuje jednoj ivici kocke, u fizičkom smislu reč je o višedimenzionalnim nizovima) i putem relacionog modela podataka (zvezdasta šema *DW* baze).

Arhitektura *OLAP* sistema uopšte se sastoji od *OLAP* baze podataka (*OLAP* servera), što fizički može biti *DW* ili *OLTP* baza, ili pak alat za izgradnju i eksploataciju struktura hiperkocke, i *OLAP* front-end komponente (*OLAP* klijent) za podršku multidimenzionalne analize i prikaza podataka. *OLAP* server može biti realizovan kao tradicionalni RDBMS, specijalizovani SQL server, ROLAP, MOLAP ili HOLAP server.

Tradicionalni RDBMS serveri nisu primarno namenjeni da efikasno podrže *OLAP* upite i baze podataka obima GB-TB budući da ne poseduju analitičke f-je, ali mogu u određenoj meri biti iskorišćeni u f-ji podrške *OLAP* sistema.

Specijalizovani SQL server ima za cilj bolju podršku *OLAP* upita nad relaciono orijentisanom multidimenzionalnom strukturom podataka, i kao takav nudi proširenje SQL-a novim f-jama (ROLLUP, CUBE, WINDOW), unapređenje koncepata fizičke organizacije podataka, i unapređenje SQL optimizatora upita.

Relational OLAP server predstavlja međusloj između "pozadinske" relacione baze (*DW* ili *OLTP*) i *OLAP* klijenta; transformiše i optimizuje SQL *OLAP* upite za izvršavanje nad pozadinskom bazom, i to tako što identifikuje poglede koje treba realizovati kao materijalizovane, transformiše SQL upite da koriste materijalizovane poglede ili da efikasnije agregiraju tražene podatke, i obezbeđuje transformaciju pogleda na strukturu *star* šeme.

Multidimensional OLAP server takođe predstavlja međusloj između "pozadinske" relacione baze, ali ova varijanta pruža direktnu podršku multidimenzionalnim strukturama tipa hiperkocke - konkretno, reč je o upotrebi višedimenzionalnih nizova u cilju implementacije hiperkocki, podaci su "keširani" u kockama i nezavisni od pozadinske baze, *OLAP* upiti se izvršavaju direktno nad kockom. Po pravilu, generisanje hiperkocke traje dugo ali se upiti potom brzo izvršavaju.

Hybrid OLAP server predstavlja kombinaciju ROLAP i MOLAP servera, gde je pozadinska baza ROLAP server i *DW* baza, a agregirani podaci su u vidu MOLAP servera i hiperkocke.

05

OLAP upiti - grupisanje, agregacije, detaljizacije (objasniti na primeru ili slici).

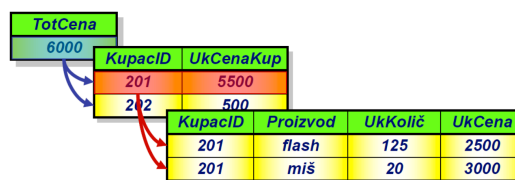
Grupisanje i agregacija: grupisanje podataka po zadatom kriterijumu jednakosti, uz agregiranje po grupama primenom klauzula GROUP BY i HAVING, ili skupovnih f-ja (listing 1).

```
SELECT
    COUNT(*) AS "tsQuantity",
    MIN(tourCreated) AS "tsCreatedFirst",
    MAX(tourCreated) AS "tsCreatedLast",
    NVP_SRC_Customer_custId AS "NVP_DW_Customer_custId"
FROM NVP_SRC_Tour
GROUP BY NVP_SRC_Customer_custId;
```

Listing 1. SQL upit iz praktičnog zadatka, primer grupisanja i agregacije.

Agregacija i detaljizacija: temelje se na dimenzijama i njihovim hijerarhijama; obuhvataju

1) agregaciju - *roll-up* (redukovanje dimenzija hiperkočke i agregacijom podataka - "odbacuje" se dimenzija najnižeg nivoa i vrši se sumiranje po odbačenoj dimenziji), i **2**) detaljizaciju - *drill-down* (inverzna operacija u odnosu na *roll-up*, obezbeđuje se detaljni prikaz po vraćenoj dimenziji, slika 2).



Slika 2. Drill-down postupak.

06

OLAP upiti - agregacione f-je, rangiranje, unakrsno tabeliranje (primer ili slika).

Agregacione funkcije: intenzivno se upotrebljavaju u OLAP upitima, resursno su zahtevne i često zahtevaju sortirane podatke. Optimizacija se vrši upotrebom agregiranih vrednosti finije u izračunavanju vrednosti grublje granularnosti (što, nažalost, nije uvek primenjivo). Vrste ovakvih f-ja: **1**) distributivne - mogu se izračunavati postupno rekursivnom primenom nad podskupovima podataka (COUNT, SUM, MAX, MIN), **2**) algebarske - mogu se izračunavati postupno primenom drugih agregacionih f-ja (AVG, STDDEV, MinN, MaxN, Center_Of_Mass), **3**) holističke - ne računaju se postupno već uvek "iz početka" nad celim skupom podataka (Median, MostFrequent). Primer je dat u listingu 1.

```
SELECT emp_num, sales,
       RANK() OVER (ORDER BY sales) AS rank FROM sales;
```

```
-- emp_num    sales    rank
--      101      2,000      1
--      102      2,400      2
--      103      2.400      2
```

Listing 2. Primer RANK f-je.

Rangiranje: f-ja RANK je OLAP f-ja za rangiranje koja računa rang za svaki red u OLAP prozoru - povratna vrednost je prirodan broj zasnovan na ORDER BY izrazu u OVER klauzuli (sa IBM-

```
CREATE VIEW order_stats AS
SELECT
    category_name,
    status,
    order_id
FROM
    order_items
INNER JOIN orders USING (order_id)
INNER JOIN prod USING (product_id)
INNER JOIN prod_cat USING (cat_id);

SELECT * FROM order_stats
PIVOT(
    COUNT(order_id)
    FOR category_name
    IN ('CPU',
        'Video Card',
        'Mother Board',
        'Storage')
)
ORDER BY status;
```

Listing 3. Primer PIVOT f-je.

ovog sajta budući da na slajdovima ovog nema, listing 2: <https://www.ibm.com/docs/en/informix-servers/12.10?topic=expressions-rank-function>).

Unakrsno tabeliranje: ovaj metod se zasniva na grupisanju podataka tako da se razume korelacija između istih, a takođe se koriste i za prikaz kako se korelacija menja od jednog grupisanja do drugog. Najčešće se koristi u statističkoj analizi kako bi se pronašli obrasci, trendovi i verovatnoće unutar sirovih podataka (pronađeno na netu budući da na slajdovima ovog nema).

07

Tehnike indeksiranja - bitmap indeksi (vrste i primene).

Jednostavni bitmap indeks: indeksira se atribut (kolona) A tabele T sa n torki, pri čemu postoji d različitih vrednosti za A u T; za svaku vrednost A se uvodi bit vektor dužine n koji sadrži n binarnih vrednosti (za svaku torku tabele po bit), što je primer primene tehnike realtivnog adresiranja torki tabele; posmatra se vrednost $A = a$ i odgovarajući vektor: i -ta pozicija ima vrednost 1 (zako za i -tu torku t_i u tabeli važi $t_i(A) = a$) ili 0 (za $t_i(A) \neq a$). Skup svih d bit vektora indeksiranog atributa predstavlja bitmap indeks.

Data je tabela Student({StId, StProg, StTip}, {StId}), indeksiraju se StProg i StTip:

```
CREATE BITMAP INDEX IDX_Std1 ON Studnet(StProg)
```

```
CREATE BITMAP INDEX IDX_Std2 ON Studnet(StTip)
```

Rezultat su sledeći indeksi:

indeks za StProg		
mat	rn	it
1	0	0
0	1	0
0	0	1
1	0	0
1	0	0

indeks za StTip		
d	m	o
0	0	1
1	0	0
1	0	0
0	1	0
0	0	1

Student		
StId	StProg	StTip
13	mat	o
17	rn	d
21	it	d
45	mat	m
34	mat	o

Prednost: efikasna podrška realizacije upita sa skupovnim f-jama i logičkim operatorima (CPU direktno operiše nad bit vektorima, takođe je omogućena prethodna priprema upita nad indeksima). **Mane:** nepogodan za indeksiranje atributa visoke selektivnosti kada je broj torki velik (linearan porast dužine bit vektora sa porastom broja torki indeksirane tabele n i broja različitih vrednosti indeksiranog atributa d). Upotrebljava se kod atributa sa niskim stepenom selektivnosti (ako je faktor d/n niži od 20%).

Bit slice indeks: reč je o bitmap matrici od n vrsta i m kolona - za n torki u indeksiranoj tabeli, gde i -ta vrsta predstavlja jednu indeksiranu vrednost u binarno kodiranom obliku koja se pojavljuje u i -toj torki tabele. Postoji analogija sa bitmap indeksom koji bi bio formiran sa d vektora, za d mogućih vrednosti iz domena indeksa od kojih je svaki dužine n . Svaki niz od d po bitova prikazuje se u binarno kodiranom ("upakovanom") obliku.

Prednosti: pogodan je za indeksiranje atributa sa većim brojem mogućih vrednosti iz domena (u kom slučaju zauzima manje prostora od običnog bitmap indeksa), efikasnost upita je zadržana na istom nivou kao kod običnih bitmap indeksa.

Mana: algoritmi za korišćenje *bit slice*-a su složeni.

Data je tabela Student čiji je *bit slice* indeks dat sa desne strane:

Student			<i>bit slice</i> indeks nad BrIsp				
StId	Ime	BrIsp	2^4	2^3	2^2	2^1	2^0
35	Ana	23	1	0	1	1	1
21	Marko	16	1	0	0	0	0
14	Ivan	9	0	1	0	1	0
45	Sanja	5	0	0	1	1	0
18	Dunja	20	1	0	1	0	0

Kodirani bitmap indeks: ponovo je reč o bitmap matrici od n vrsta i m kolona, gde za n torki u indeksiranoj tabeli svaka i -ta vrsta kodira indeksiranu vrednost u i -toj torci putem niza bitova dužine m , pri čemu je d broj različitih aktuelnih vrednosti indeksiranog atributa; konačno formira se m kodiranih vektora, svaki dužine n . Koristi se zajedno sa tabelom kodiranih vrednosti i primenljiv je nad atributima bilo kog tipa.

$d(\text{ime})=4, m=\log_2 4=2$				Student			Kodirani bitmap idx		
Ime	2^2	2^1	2^0	StId	Ime	BrIsp	2^2	2^1	2^0
Dejan	0	0	0	10	Draga	9	0	0	1
Draga	0	0	1	77	Dejan	13	0	0	0
Goran	0	1	0	13	Ivana	16	0	1	1
Ivana	0	1	1	1	Goran	4	0	1	0

Prilikom ažuriranja kodiranog bitmap indeksa dodavanjem novih torki u tabelu činjenica dolazi do produžavanja svih vektora bitmap indeksa, budući da broj vektora bitmap indeksa nad atributom A zavisi od broja aktuelnih vrednosti d .