

LATVIJAS UNIVERSITĀTE  
FIZIKAS UN MATEMĀTIKAS FAKULTĀTE  
DATORIKAS NODAĻA

**DATU NOLIKTAVU ATBALSTS DATUBĀZU PĀRVALDĪBAS  
SISTĒMĀ DB2**

MAĢISTRA DARBS

Autors: **Anna Ļivčaka**

Stud. apl. Nr. DatZ010118

Darba vadītājs: Kārlis Podnieks, LU profesors

RĪGA 2007

## ANOTĀCIJA

Šajā maģistra darbā tiek apskatīts datu noliktavu atbalsts – kompānijas *IBM* datu bāzu pārvaldības sistēmā *DB2*. Dots divu vadošo firmu *IBM* un *Oracle* datu bāzu produktu vispārējs salīdzinājums, kā arī tiek salīdzināti datu noliktavu realizācijas risinājumi šajos produktos. Tiek dots *DB2* datu noliktavu atbalstošo funkciju apraksts, ko piedāvā *DB2 Warehouse Edition*. Tiek parādīti *DB2 Warehouse Edition* demonstrācijas programmas izmēģināšanas rezultāti. Ir aprakstīts datu noliktavu risinājums specifiska banka nozarē. Kā potenciāls lietojums parādīta datu noliktavu projektēšanas problēma konkrētā bankā.

## ANNOTATION

This master work deals with data warehouse support – DB2 Database Management System from IBM company. It views a general product comparison of two leading companies IBM and ORACLE, also it compares data warehouse realization solutions. This work provides the description of DB2 data warehouse functions, which are offered by DB2 Warehouse Edition. It shows DB2 Warehouse Edition demo programm performance results. It describes the solution of data warehouses in a specific bank branch. It shows data warehouse design problem in particular bank.

## AUTOREFERĀTS

Šajā darbā tiek apskatīta datu noliktavu atbalsts – datu bāzu pārvaldības sistēma *DB2*. Tā ir programmēšanas produktu saime *IBM* kompānijas informācijas pārvaldības jomā. Balstoties uz [1] informāciju, tiek dots īss ievads par to, kas ir *DB2* sistēma, kādi tai ir rīki.

Balstoties uz [2], [3], [4], [5] informāciju, tiek parādīts *IBM* un *Oracle* firmu salīdzināšanas, kas bāzējas uz biznesa inteligences, SQL valodas izmantošanas, *DB2* un *Oracle* pielietošanas Linux operētājsistēmā salīdzināšanas.

Balstoties uz [6], [9] informāciju, apskatīti *IBM* un *Oracle* jaunie izstrādājumi: *DB2 Warehouse 9* zem nosaukuma *Viper* un *Oracle Database Vault*. Ir parādīts, ka divu datu bāzu atbalstu sistēmu salīdzināšana ir neviennozīmīga un dažreiz ir grūti novērtēt, kura no tām ir labāka.

Balstoties uz [7], [8], [10], [11], [12], [13], [14] informāciju, parādītas lietotāju atsauces par darbu uz lieldatoriem, īpašuma izmaksām un darbu ar kanāliem, un *DB2* konkurentu *Oracle* skats uz drošību. Viedokļi veidoti dažādos laika brīžos un līdz ar to ir atšķirīgi.

Balstoties uz [15], [16], [17] informāciju, tiek parādītas *DB2 Data Warehouse Edition* funkcijas, kuras šis produkts piedāvā datu noliktavu atbalstam.

Balstoties uz [18], ir apskatīta *DB2* demonstrācijas programma un izmēģinātas dažas funkcijas tajā. Ir salīdzināta efektivitāte ar funkciju izmantošanu un bez tām. Tiek parādīti eksperimentu rezultāti.

Balstoties uz [19], [20] informāciju, tiek dots banku datu noliktavu risinājuma pārskats.

Kā konkrēts piemērs no autores pieredzes, ir aprakstīts kādai informācijai jābūt konkrētas bankas datu noliktavā un tiek dots bankas datu noliktavas aptuvenā shēma.

# SATURS

APZĪMĒJUMU SARAKSTS.....	7
IEVADS.....	8
1. DB2 SALĪDZINĀJUMS AR ORACLE.....	9
1.1. Ievads datu bāzes pārvaldības sistēmā DB2.....	9
1.2. Biznesa inteliģence .....	11
1.3. SQL valodas izmantošana .....	14
1.4. DB2 un Oracle salīdzināšana Linux sistēmā.....	18
1.4.1. Izmantošanas vienkāršība.....	18
1.4.2. Īpašuma pilnas izmaksa.....	20
1.4.3. Drošība.....	21
1.4.4. Mērogojamība. ....	22
1.5. Jaunākais izstrādājums DB2 9.....	22
1.5.1. DB2 9 datu pārvaldības metodes.....	23
1.5.2. DB2 Viper salīdzināšana ar Oracle: kompresija un drošība.....	24
1.6. Oracle jaunākais izstrādājums Oracle Database Vault.....	26
1.7. Lietotāju atsauksmes.....	27
1.7.1. Lietotāju atsauksmes par lieldatoriem.....	27
1.7.2. Lietotāju atsauksmes par īpašuma izmaksām.....	28
1.7.3. Par darbu ar kanālu.....	29
2. DB2 DWE EDITION FUNKCIJAS.....	31
2.1. DWE veidošanas funkcijas.....	33
2.2. Datu pārveidošanas funkcijas.....	34
2.3. OLAP funkcijas.....	36
2.4. Izraces funkcijas.....	38
2.5. Iekļautas analītiskas funkcijas.....	40
3. DB2 DWE IZMANTOŠANA .....	42
3.1. Apkopojumu tabulas .....	43
3.2. Lineāras regresiju funkcijas .....	45
3.3. Kubu un agregāciju operatori.....	48
3.4. Ranga funkcijas.....	53
3.5. Daudzdimensiju klasteru meklēšana.....	54
4. Datu noliktavu risinājums bankām.....	59
4.1. BDW modelis.....	60

4.2. Biznesa risinājumu šabloni.....	60
4.3. Lietojumprogrammu risinājumu šabloni.....	61
4.4. BDW Projektu skati.....	62
4.5. BDW komponentes un Basel II arhitektūra.....	62
5. BANKU DATU NOLIKTAVAS VEIDOŠANA.....	64
SECINĀJUMI.....	67
IZMANTOTĀ LITERATŪRA .....	68

## APZĪMĒJUMU SARAKSTS

ODBC - *Open Database Connectivity*. Atklāta saskarne datubāzu piekļuvei.

AS/400 - *Application System/400*. Serveris, kas strādā ar tīmekļa lietojumprogrammām.

BI – *Business Intelligence*. Datu noliktavu, datu vitrīnu un analītiskas iekārtas, kas dabū datus lēmumu pieņemšanai.

ESE – *Enterprise Server Edition*. DB2 izlaidums korporatīvai datu noliktavai.

RAC – *Real Application Clusters*. Oracle korporācijas programmatūra, kas ļauj vairākiem lietotājiem vienā datubāzē strādāt ar kādu programmu vienlaicīgi.

DWE – *Data Warehouse Edition*. Izlaidums datu noliktavai.

SQW – *SQL Warehousing Tool*. Rīks, ar kura palīdzību var pārvaldīt un pārvietot datu noliktavas datus.

DDL - *Data Definition Language*. Valodu saime, ko izmanto datu bāzes struktūras aprakstam.

MQT – *Materialized Query tables*. Tabulas, kas satur agregācijas datus, kas atbilst OLAP struktūrām.

MDC – *Multidimensional Clustering*. Daudzdimensiju klasteru meklēšana.

BDW – *Banking Data Warehouse*. Datu noliktavu risinājums bankām.

BST – *Business Solution Templates*. Biznesa risinājumu šabloni.

AST – *Application Solution Templates*. Lietojumprogrammu risinājumu šabloni.

## IEVADS

Šajā darbā tiek apskatīta datu noliktavu atbalsts – datu bāzu pārvaldības sistēma *DB2*, un tās lietojums. Tā ir programmēšanas produktu ģimene *IBM* kompānijas informācijas pārvaldības jomā. Tiek dots īss ievads par to, kas ir *DB2* sistēma, kādi tai ir rīki. Kaut biežāk runājot par *DB2* tiek domāta relāciju datu bāzu pārvaldības sistēma (*DBPS*), *DB2 Universal Database (DB2 UDB)*, ko izstrādā un izlaiž *IBM* kompānija.

*DB2* ir gara vēsture, un, kā daži uzskata, tā ir kļuvusi par pirmo *DBPS*, kas izmanto *SQL* valodu. Šī *DBPS* saņēma nosaukumu *DB2* 1982. gadā, kad tika izlaista tās relīze, kā arī *SQL* lieldatoriem.

Pirmā nodaļā tiek dots neliels ievads, kur īsi tiek pastāstīts par *DB2* un tās rīkiem. Ir pateikts par pamatrisinājumiem datu noliktavu izstrādē un tiek salīdzināta *DB2* un *Oracle* biznesa inteliģence, t.i. datu noliktavu, datu vitrīnu un analītiskas iekārtas, kas dabū datus lēmumu pieņemšanas atbalstam. Tiek salīdzinātas *SQL* valodu izmantošanas īpašības *DB2* un *Oracle* sistēmās un *DB2* un *Oracle* izmantošana *Linux* sistēmā. Sistēmas tiek salīdzinātas pēc izmantošanas vienkāršības, īpašuma izmaksām, drošības un mērogojamības. Tiek stāstīts par jaunāku *IBM* izstrādājumu - *DB2 Warehouse 9* zem nosaukuma *Viper*, tiek izteikti lietotāju viedokļi, kas runā par labu *DB2*. Arī tiek stāstīts par jaunāku *Oracle* izstrādājumu - *Oracle Database Vault*, kas parāda, ka *Oracle* piemīt īpašības, kas tika uzskatītas par *DB2* privilēģiju. Tiek parādītas lietotāju atsauksmes par saistītām ar *DB2* lietām.

Otrā nodaļā tiek stāstīts par *DB2 Data Warehouse Edition* funkcijām, kuras šis produkts piedāvā datu noliktavu atbalstam. Tās ir *Data Warehouse Edition* veidošanas, datu pārveidošanas funkcijas, *OLAP*, izraces un iekļautas analīzes funkcijas.

Trešā nodaļā ir praktiski parādītas biznesa inteliģences funkciju izmēģināšana, ko piedāvā *DB2* demonstrācijas programma. Tiek izpētītas apkopojumu tabulu funkcijas, lineāras funkcijas, kubu un agregāciju funkcijas, rangu funkcijas un daudzdimensiju klasteru meklēšana. Tiek veikts vaicājumu salīdzinājums ar funkciju izmantošanu un bez tām. Tiek parādīti eksperimentu rezultāti.

Ceturtnā nodaļā ir stāstīts par datu noliktavu risinājumu bankām – *Banking Data Warehouse*. Tiek aprakstītas *Banking Data Warehouse* komponentes.

Piektā nodaļā īsi pastāstīts par to, kāda veida informācijai jābūt attēlotai bankas datu noliktavā un tiek dots procedūru saraksts, ko veic banka. Ir uzzīmēta un aprakstīta bankas datu noliktavas modeļa aptuvena shēma.

# 1. DB2 SALĪDZINĀJUMS AR ORACLE

## 1.1. Ievads datu bāzes pārvaldības sistēmā DB2

Vienkāršākajā gadījumā klienta-servera informācijas sistēma sastāv no divām pamatkomponentēm [1]:

- datu bāzes serveris, kas pārvalda datus un izpilda vaicājumus no klientu lietojumprogrammām;
- klienta lietojumprogrammas, kas nodrošina lietotāja saskarni un sūta vaicājumus uz serveri.

DB2 Universal Database [1] ir datu bāzes pārvaldības sistēma, ko izmanto datu bāzu serveris. Serveris ir lietojumprogramma vai operētājsistēmas dienests. Tas var manipulēt ar failiem, kur glabājas dati, izpildīt vaicājumus, atbalstīt datu integritāti, nodrošināt datu piekļuvi, veidot rezerves kopijas, veidot operāciju protokolus, kas tika saistīti ar datu izmaiņām.

Projektēt datus priekš *DB2 Universal Database*, kā arī realizēt apgrieztu projektēšanu, var gandrīz ar visu vispopulārāko datu projektēšanas līdzekļu palīdzību - *ERwin (Computer Associates)*, *Visible Analyst (Visible Systems Corporation)*, *PowerDesigner (Sybase)*, *System Architect (Popkin Software)*, *Visio Enterprise (Microsoft)* [1]. *ERwin* un *Visible Analyst* ne tikai ģenerē parastu scenāriju tabulu, indeksu, serveru ierobežojumu, triggeru un procedūru veidošanai, bet arī ļauj ģenerēt skriptus priekš *DB2*, kas ņem vērā tādus specifiskus šīs DBPS fiziskas atmiņas organizēšanas īpašības, kā tabulu telpas un atgrūšanās segmentu vadība, kā arī pārvalda datu bloķēšanas režīmus.

*DB2 Universal Database* atbalsta datu noliktavas būvēšanu un ne tikai realizē datu importu noliktavā, bet klientu daļas sastāvā satur arī *Data Warehouse Center* [1], kas ļauj pārvaldīt visus procesus, saistītus ar datu noliktavu veidošanu un atbalstu, iekļaujot datu noliktavas veidošanu, datu noliktavu „zvaigznes” tipa shēmu būvēšanu, izejošo datu avotu noteikšanu, kā arī sarakstu veidošanu, saskaņā ar kuriem datu noliktavas dati tiks atjaunoti.

Sākot ar *DB2 Universal Database 7.2* versiju, *Data Warehouse Center* atbalsta arī nekorekto datu pārbaudes un labošanas iespēju.

Lēmumu pieņemšanai bieži vien tiek izmantota OLAP – vairākdimensiju agregātu datu kopu veidošanas tehnoloģija, kas ir datu noliktavu satura apstrādes rezultāts. OLAP-noliktavas pēdējā laikā ir plaši izplatītas lēmumu pieņemšanas atbalsta sistēmās.

OLAP atbalstam *DB2 Universal Database* servera sastāvā ietilpst *OLAP Starter Kit*, kas ļauj veidot vairākdimensiju datu bāzes. *OLAP Starter Kit*, kā arī atsevišķi uzstādāmais *DB2 OLAP Server*, integrē *Hyperion Essbase* ar *DB2 Universal Database* OLAP-servera kodolu.

Servera *Essbase* pamatā ir vairākzvaigžņu arhitektūra, un tas ir domāts priekš daudzlietotāju pieejai OLAP-datiem un analītisko rēķinu īstenošanai. Tas var tikt izmantots kopā ar dažādiem datu avotiem un analītiskiem pielikumiem. *Hyperion Essbase* atklātā arhitektūra atbalsta piekļuvi šī servera OLAP-datiem ar dažādu līdzekļu palīdzību – elektronisko tabulu, vaicājumu būvētāju, atskaišu ģeneratoru, ļaujot šiem līdzekļiem būt klientu pielikumu lomā. *Essbase* atbalsta vairāk nekā 50 klientu produktu, piemēram, *Crystal Reports* un *Crystal Info (Crystal Decisions)*, OLAP-datu apskatu līdzekļus, tādus kā *Business Objects* un *Cognos PowerPlay*, vadošo ražotāju elektroniskās tabulas.

Ērtu administrēšanas utilītu esamība dažreiz ir viens no nozīmīgiem faktoriem DBPS izvēlē. *DB2 Universal Database* klientu daļā ir sekojošie produkti [1]:

*Control Center* – datu bāzu administrēšanas produkts, kuru pārvalda *DB2 Universal Database*, tam ir saskarne, kas ir līdzīga *Windows Explorer* saskarnei. Ar *Control Center* palīdzību var veidot jaunas datu bāzes, tabulu telpas, tabulas, indeksus, triggerus, skatus, organizēt replikācijas starp dažādiem *DB2* serveriem, manipulēt ar lietotājiem un grupām.

*Client Configuration Assistant* – domāts *DB2* klientu daļas piekļuves konfigurācijai pie lokāliem attālinātiem serveriem, ļauj manipulēt pieejamo *DB2* datu bāzu savienojumu sarakstu, aprakstīt tos kā ODBC-datu avotus, mainīt piekļuves paroles tiem.

*Event Monitor* – domāts informācijas krājumam par to, kas notiek datu bāzēs ekspluatācijas laikā. Tā ļauj reģistrēt agrāk noteiktus notikumus, piemēram, savienojumu ar datu bāzi, SQL-vaicājumu izpildi utt., un ar tiem saistītus datus, piemēram, procesora ielādes laiku, bloķēšanas laiku, pielikuma vārdu, kas iniciēja notikumu u.c.

*Event Analyser* – ļauj analizēt datus, kas tika pierakstīti ar *Event Monitor* palīdzību, ar nolūku saņemt statistiskus datus un datu bāzes servera produktivitātes novērtējumu.

*Command Center* – ļauj pārvaldīt visus administrēšanas uzdevumus, tajā skaitā arī ielādēt augstākminētus administratīvus produktus, kā arī izpildīt SQL-vaicājumus ar vaicājumu veidošanas vizuāla līdzekļa palīdzību *SQL Assist*. Var arī apskatīt vaicājuma izpildes plānu un saņemt ziņas par katru operāciju, kas tika veikta izpildes laikā.

Citi *DB2 Universal Database* produkti:

*DB2 Query Patroller* – vaicājumu veidošanas un resursu pārvaldības līdzekļu kopa lēmumu pieņemšanas sistēmai. Šis produkts saņem ODBC-vaicājumus no klienta, analizē tos, un dinamiski ievieto dažādos mezglos;

*DB2 Connect* – līdzeklis dažādu klientu ar *DB2* un *AS/400* savienojumu pārvaldībai;

*DB2 Relational Connect* – līdzeklis relāciju un nerelāciju datu piekļuvei, kas tiek novietotas dažādās platformās, kā vienīgam datu attēlam;

*DB2 Data Links Manager* – līdzeklis papildus failu pārvaldībai, kurus pieslēdz pie *DBPS*.

Arī ir pieejami bezmaksas līdzekļi datu migrācijai no *Microsoft Access* uz *DB2*, kā arī no *DBPS Oracle, Microsoft, Sybase* un *Informix*.

## 1.2. Biznesa inteliģence

Salīdzināt datu bāzes [2, 25.lpp] pārvaldības sistēmas nav viegli, jo:

- datu bāzu funkcionalitāte un sarežģītība ir ļoti plašas un padziļināta salīdzināšana prasa daudz laika;
- dažu īpašību nozīmīgums var tikt novērtēts pēc praktiskās pieredzes, izvēršanas un izmēģināšanas (t.i. pēc izpildes iedarbības);
- salīdzināt sistēmas pēc kādas īpašības ir grūti, jo tām ir potenciāli dažādi implementācijas dziļumi un vispārējas atšķirības arhitektūras pieejā un produkta filozofijā;
- detalizētas īpašības bieži vien nevar tikt salīdzinātas pa tiešo.

*IBM* kompānijas mērķis – integrēto programmatūras produktu un servisu kopuma nodrošināšana, kas tika veidotas uz vienīgās arhitektūras. Datu noliktavu pamats ir *DBPS DB2*. *IBM* priekšrocības ir tā, ka dati, kuri ir jāizvelk no operatīvas datubāzes un ievietot datu noliktavā, atrodas *IBM* sistēmās. Tas saka par programmatūras produktu ciešo integrāciju.

Tiek piedāvāti trīs risinājumi datu noliktavām [3]:

- Izolēta datu vitrīna – domāta atsevišķo uzdevumu risināšanai bez sakara ar korporācijas kopīgo datu noliktavu;
- Atkarīga datu vitrīna – analogiski izolētai vitrīnai, bet datu avoti atrodas zem centralizētas pārvaldības;
- Globāla datu noliktava. Korporatīva datu noliktava, kas tiek pilnīgi centralizēti kontrolēta un pārvaldīta. Globāla datu noliktava var paplašināties centralizēti un saturēt no vairākiem tīkla novietotiem datu tirgum.

*Oracle* risinājums datu noliktavu jomā pamatojas uz diviem faktoriem: pašas kompānijas produktu plašs sortiments un partneru darbība *Warehouse Technology Initiative* programmā. *Oracle* iespējas datu noliktavu jomā tiek bāzētas uz sekojošām sastāvdaļām [3]:

- relāciju DBPS, kas pastāvīgi pilnveidojas datu noliktavu vajadzību apmierinājumam;
- gatavu pielikumu kopa, kas nodrošina datu noliktavas izstrādes iespēju;
- augsts tehnoloģisks potenciāls kompānijas datu analīzes jomā;
- vairāku produktu, ko ražo citas kompānijas, pieejamība.

Datu noliktavu, datu vitrīnu un analītiskas iekārtas, kas dabū datus lēmumu pieņemšanas atbalstam sauc par Biznesa Inteliģenci (BI) [2, 47.lpp]. BI saprotama kā tehnoloģija, kas palīdz uzņēmumam saprast biznesu, analizējot biznesa operācijas. Lai to paveiktu, ir jāsavāc informācija par biznesa aktivitātēm un klientu uzvedību, parasti no dažādiem avotiem (t.i. no iekšējām IT sistēmām, darba virsmas datiem, un iespējams ārēji no partneru un tirgus pētījuma). Šī informācija pāriet integrētā datubāzē (datu noliktavā) un tiek uzturēta uz esošās bāzes. Tad šai informācijai var pielietot speciālas analīzes tehnikas, lai ieskatītos savā biznesā, kas vedīs pie uzlabošanas novērojuma.

Lai atbalstītu BI izpildi, ir vajadzība pēc daudzām tehnoloģijām. Daudzus gadus BI tika implementēta ar rīku kolekciju no daudziem dažādiem ražotājiem. Tagad BI tehnoloģijas daudz stingrāk integrētas. Datubāzu ražotāji spēlē svarīgu lomu BI platformu nostiprināšanā: datu noliktavas satur ļoti lielus datu sējumus un tām jāatbalsta eksromptvaicājumi no daudzajiem lietotājiem, kuru domeini ir sarakstīti vadošajā DBPS. DBPS pati nevar adresēt pilnus BI apgabalus, bet tiem laukiem, kas nepieder DBPS kodola kompetences, jābūt adresētiem. Kamēr pilnas integrācijas pieeja liekas pievilcīga no izmantošanas redzes viedokļa, šādi risinājumi nevar sasniegt augsta līmeņa tehnoloģijas, kuras atbilst konkrētām biznesa prasībām un tām pretī nostājas labākas tehnoloģijas. Tomēr nebūs pārāk lietderīgi vērtēt DBPS iespējas tikai no DBPS perspektīvas.

Atkarībā no individuālām prasībām, datu noliktava var būt ļoti liela (vairāk nekā 1 TB), datu bāzes pieaug līdz OLTP datu bāzu izmēram. Vaicājumi, kas izpildās noliktavās ir sarežģītāki un resursi ir intensīvāki par OLTP transakcijām. Kad jāpārbauda atbilstību prasības pēc lieliem datu apjomiem ar komplekso analīzi, mērogojamības un izpildes iespējas spēlēs svarīgāku lomu, nekā tieša BI funkcionalitāte.

Integrētai BI platformai jāatbalsta sekojošie apgabali:

- Datu noliktavas apkalpošana. Apkalpošana iekļauj iespējas projektēt, kā arī izvilkēt, pārvadīt, pārveidot un tīrīt datus no dažādām avotu sistēmām noliktavas shēmā. Ir vēlams plašs loks priekšdefinētu pārveidošanu, un platformai arī jāatbalsta individuāli paplašinājumi.

- Datu noliktavas pārvaldība. Galvenā problēma datu noliktavas pārvaldībā ir datu pārvaldība. Tas iekļauj gan relāciju, gan dimensiju modelēšanas stratēģijas. Repozitorijā atrodas metadati. Metadati apraksta no kura avota dati tiek ņemti un kā tie tiek izmantoti. Ir jābūt pieejamām saskarnēm lai veiktu šo metadatu apmaiņu ar citām datu noliktavas platformas komponentēm. Šo komponentu optimizāciju veic datu noliktavas pārvaldība.
- Piekļuve datu noliktavai. Analīzes metodes lietotāju biznesa prasību atbalstam iekļauj vaicājumus un atskaišu veidošanu, OLAP un datizrāci. Kompleksām BI lietojumprogrammām ir vajadzīgi modificējamie kubi, prognozēšana un what-if analīze. No datubāzes perspektīvas tas galvenokārt iekļauj atbalstītus API un lasīšanas/rakstīšanas pieeju relāciju un dimensiju datiem.

*DB2* piedāvā speciālu paketi *DB2 Data Warehouse Edition* (DWE), kas ir pieejams BI mērķa funkcionalitātei. Tomēr *Oracle* trūkst šis speciālais piedāvājums. *DB2* DWE piedāvā plašas iespējas BI funkcionalitātei par konkurētspējīgām cenām.

Integrēts avotu atbalsts [2,48.lpp.]. *DB2 Warehouse Center* ir *DB2 UDB* tehniskā īpatnība un nodrošina caurlaidību lokālam serverim. *DB2 Warehouse Manager* ir *DB2 UDB* opcija, kas paplašina *Warehouse Center* līdz novietojumam, kas nav Warehouse Server un nodrošina ETL funkcionalitāti. Abi šie produkti atbalsta lielu skaitu dažādu avotu tipus ieskaitot visus vadošos DBPS. *DB2 Manager* piedāvā līdzekļus vairākām platformām attālinātu datu piekļuves atbalstam. Opcija *DB2 Information Integrator* pievieno *DB2*-integrēto atbalstu priekš *Oracle*, *Teradata* un *Sysbase*, atļaujot vaicājumu optimizāciju un pārrakstīšanu.

Datu pārveidošanās [2, 49.lpp.]. Sistēmā *DB2* var implementēt individuālas glabājamās procedūras, kā arī izvēlēties no dažādiem priekšdefinētiem pārveidojumiem. Gan *Oracle*, gan *DB2* ir spējīgi darboties ar datu tīrīšanu.

Relāciju datu noliktavas pārvaldība [2, 49.lpp.]. Abas DBPS atbalsta relāciju shēmas un piedāvā *SQL* paplašinājumus relāciju datu analīzei OLAP veidā: kubu veida agregācijas ar augšupejošo analīzi un statiskām funkcijām. Tas iekļauj *SQL* paplašinājumus grupēšanai, rēķināšanai, rekursīvai atlasei un rezultātu kopas agregāciju. *DB2* un *Oracle* pārvalda ar *SQL*-standartu elastīgu, stingru implementāciju, tomēr *Oracle* piedāvā arī nestandartus papildinājumus. *DB2 Cube Views* ļauj definēt individuālus agregācijas skriptus visos dimensionālās shēmas līmeņos. *Oracle 10g* ir ļauti aprēķini atmiņas iekšienē, kompleksie aprēķini daudzdimensiju masīvos.

Dimensiju datu noliktavas pārvaldība [2, 49.lpp]. *Oracle* pieeja pamatojas uz *Oracle Express* daudzdimensionālu atmiņu, tajā ir arī daudzdimensiju OLAP kuba arhitektūra. Papildus metadatu slānis dod iespēju konceptuālam kubu aprakstam, nodrošinot kubu glabāšanu relāciju formātā. Agregāti relāciju un daudzdimensiju OLAP kubos ir atbalstīti ar materializētiem skatiem, tomēr speciālai sintaksei OLAP saskarnes atbalstam jābūt izmantotai veidojot tos uz daudzdimensiju OLAP datiem. *Oracle 9i* konceptuāli vienādi materiāli skati uz ROLAP un MOLAP datiem netika atzīti ar optimizētāju, sintakses nesakritības dēļ materiālu skatu uzbūvē. *Oracle 10g* atrisināja šo problēmu ar „*Query Equivalence*”, piedāvājot iespēju pieņemt divus dažādus vaicājumu sintakses tipus kā konceptuāli vienādus, nodrošinot vaicājumu pārrakstu, kas tika rakstīts Oracle-specifiskajā relāciju-dimensiju sintaksē uz OLAP un otrādi.

*DB2 Cube Views* padara *DB2* saprotamu OLAP. *Cube Views* pievieno OLAP metadatus priekš *DB2*. *DB2 Cube Views* optimizē labākas iespējamās materializētas vaicājumu tabulas atbalstot daudzdimensiju vaicājumus. BI rīku gala lietotāji, veidojot SQL pieprasījumus var redzēt novērojamu izpildes paātrinājumu. *DB2* optimizētājs paplašina materializēto vaicājumu tabulu izmantošanu. *Cube Views* ir apvienots ar daudzu biznesa partneru produktiem, kas atbalsta pieeju *DB2*, ieskaitot *DB2 OLAP Server*, *Cognos*, *Business Objects* un *Microstrategy*. *DB2 OLAP Server* ir *IBM* MOLAP piedāvājums. *Cube Views* ROLAP priekšrocība ir vienkāršā izmantošana un mērogojamība lieliem datu sējumiem.

Metadatu pārvaldība [2, 50.lpp]. Gan *DB2*, gan *Oracle* atbalsta kopīgo datu noliktavas modeļu standartu un apmaiņas ar noliktavu metadatiem. Ar *Cube Views DB2* piedāvā metadatu savienojumu ar vadošiem OLAP ražotājiem. *DB2* atbalsta metadatus arī ar informāciju katalogu.

### **1.3. SQL valodas izmantošana**

Abas datu bāzes pārvaldības sistēmas, gan *Oracle*, gan *DB2* izmanto SQL valodu. Tomēr šo sistēmu SQL valodas īpašības un funkcijas atšķirās. *DB2* un *Oracle* iespējas nav simetriskas, katrā sistēmā SQL valodai ir priekšrocības un trūkumi. Tātad šo divu sistēmu funkcionālas iespējas kopumā ir ekvivalentas. Nevar viennozīmīgi teikt kurai no šīm divām sistēmām SQL valoda ir efektīvākā. Kuru no sistēmām jāizvēlas savu biznesa problēmu risinājumā, ir atkarīgs no katra izstrādātāja vajadzībām un prioritātēm.

Salīdzinot SQL valodas izmantošanu datu bāzes pārvaldības sistēmās *DB2* un *Oracle*, ir vajadzīgs apskatīt katru valodas aspektu. Šajā darbā SQL valoda šajās sistēmā tiek salīdzināta pēc sekojošiem kritērijiem:

- Datu apraksts;
- Vaicājumi;
- Skati;
- Trigeri;
- SQL procedūru paplašinājumi;
- Transakciju vadība;
- Piekļuves pārvaldība [4].

Apskatīsim šos kritērijus sīkāk:

Datu apraksts. *DB2* atbalsta iebūvētu datu tipu kopu, kas pieņemts daudzās DBPS. *Oracle* nodrošina visu skaitlisku tipu vienoto realizāciju ar tipu NUMBER (n, p), bet pie tām apstrāda arī standartu skaitlisku tipu deklarāciju, un automātiski pārvērš tos par NUMBER. Visu skaitlisku tipu vienota realizācija *Oracle* sistēmā veido iespēju mainīt parametrus un to realizāciju (n, p) operatorā ALTER TABLE.

Datumu un laiku realizācijai *DB2* atbalsta standartus datu/laika tipus: DATE, TIME, TIMESTAMP. *Oracle* ir vienīgs tips – DATE, kas atbilst standartam TIMETAMP, citu tipu *Oracle* nepazīst. Datu/laika līdzekļi šajos DBPS ir būtiski atšķirīgi, funkcionāli pilni un ne pārāk vienkārši izmantošanā. *DB2* piedāvā interesantāka datu/laika aritmētikas iespēja, ar laika vienību izmantošanu.

Tipu pārveidošanu *DB2* izpildās ar standartam funkcijām, kuru vārdi sakrīt ar mērķa tipu vārdiem un universālas pārveidošanas standartu operāciju CAST. *Oracle* ir tikai trīs formāta pārveidošanas funkcijas – TO\_CHAR, TO\_NUMBER, TO\_DATE, kas ir funkcionāli bagātākas par standartiem, bet sarežģītākas izmantošanā.

Integritātes ierobežojumi, kas tika uzdoti datu aprakstā, abās DBPS pārsvarā ir vienādas. *DB2* dod lielas izvēles iespējas pie darbību noteikšanas, kurus izpilda norāžu integritātes nodrošinājumam pie ierakstu dzēšanas/izmaiņām.

Vaicājumi. *DB2* SQL valodai salīdzinājumā ar *Oracle* ir dažas SQL papildus iespējas, kas palīdz formulēt labāk saprotamus un kompakus vaicājumus. *DB2* SQL priekšrocību vidū ir sekojošais:

- skalāru apakšvaicājumu izmantošanas iespēja SELECT teikuma kolonnu elementu izskatā;
- nosacījuma izteiksme – CASE;

- ierobežojumu neesamība uz vaicājumu izmantošanu salīdzinājuma predikātos (tai skaitā var salīdzināt divu apakšvaicājumu rezultātus, kas nav iespējams *Oracle*).

Tabulu savienošana *Oracle* izpildās pēc vērtību vienādības predikāta kolonnās WHERE izteikumā. *DB2* piedāvā gan šo iespēju, gan JOIN operāciju, kas nosaka savienojumu jau FROM izteikumā. Ārējo savienojumu izpildīšanai *Oracle* tika ieviesta speciāla operācija (+), kas ļauj noteikt vai nu kreiso, vai nu labo savienojumu (bet ne abus kopā). *DB2* tās pašas JOIN operācijas paplašinājums nodrošina kreiso, labo, kā arī pilnu ārēju savienojumu.

Abas DBPS atbalsta arī hijerahiskus vaicājumus, kuru izpildīšana jau atšķirās no relāciju modeļa. *Oracle* šim nolūkam tika ieviesta īpaša sintakse, kas iekļauj atslēgas vārdus CONNECT BY, START WITH, PRIOR. *DB2* risina šo pašu uzdevumu ar laicīgu skatu pielietošanu ar vienīgu atslēgas vārdu WITH. Kaut hierarhiskie vaicājumi *DB2* ir mazāk kompakti, nekā *Oracle*, tomēr tie ir labāk saprotami programmētājiem, kuri darbojas ar rekursijām. Pie tām laicīgu skatu funkcionalitāte nav tikai hierarhiskā atlase, tos arī ērti izmantot kā sarežģītu vaicājumu saīsināšanas līdzekli.

Skati. Skatu veidošanā abu DBPS iespējas ir praktiski vienādas, tās atbilst CREATE VIEW operatora standartam iespējām un atšķiras tikai ar to vaicājumu formulējumiem, kas nosaka skatu. Abas DBPS paplašina skatu izmaiņu standartus ierobežojumus ar to, ka ļauj mainīt skatus, kuru definīcija jau satur ieliktus vaicājumus. *DB2* mainīti skati ir arī tie skati, kuru definīcijā satur apvienojumu (UNION ALL) un divus vai vairākas bāzes tabulas. *Oracle* ļauj mainīt dažus skatus, kas satur tabulu savienojumu savā definīcijā (tajos gadījumos, kad skata atslēga ir arī bāzes tabulas atslēga). Šī iespēja *Oracle* ir ierobežota un sarežģīta saprašanai, tomēr tā ir ļoti svarīga.

Trigeri. Standartu CREATE TRIGGER operatora sintaksi abas DBPS papildina ar savām modifikācijām.

*Oracle* dod iespēju veidot trigerus ne tikai bāzes tabulām, bet arī skatiem. Aktivizācijas nosacījums INSTEAD OF (tikai skatu trigieriem) dod iespēju izpildīt dažas izmaiņas pat tiem skatiem, kas formāli nav modificējamie. *Oracle* ir vairāki ierobežojumi trigera papildus nosacījuma izpildei (WHEN izteikums), kas tiek kompensētas ar to, ka trigera darbības tiek uzdotas ar PL/SQL operatoru bloku, kas paplašina trigera loģiskas iespējas.

*DB2* ir tabulu modifikāciju operatoru izpildes ierobežojums BEFORE-trigeros, kas tomēr nav būtisks funkcionalitātes ierobežojums. Trigera darbībās ir iespējama norāde ne tikai uz vecu, bet arī uz jaunu modificēta ieraksta vērtību, kas ir iespējams ne tikai *DB2*, bet arī *Oracle*, un uz modificētas tabulas jaunu un vecu vērtību. Trigera darbības tiek realizētas ar vienu SQL operatoru vai operatoru bloku, kas izpildās kā atomārs. Operatoru saraksts, kas

ir pieļaujams triggerī, tiek paplašināts ar operatoriem SET un SIGNAL SQLSTATE, arī triggera darbību loģiskās iespējas ir ierobežotas ar izteikuma WHEN teikuma CASE iespējām (ar cik iespējams sarežģītāku nosacījumu).

SQL procedūru paplašinājumi. Principiāla atšķirība ir tā, ka *Oracle* PL/SQL bieži vien tiek pozicionēts kā patstāvīga programmēšanas valoda ar iebūvētu SQL, bet *DB2* procedūru valoda principiāli ir SQL paplašinājums un savās bāzes konstrukcijās (nosacījumi, izteiksmes, u.c.) izmanto SQL sintaksi un iespējas.

Tranzakciju vadība. No četriem izolēšanas līmeņiem, ko nosaka SQL standarts, *DB2* nodrošina aptuvenu vai tiešu atbilstību visiem līmeņiem ( kaut to nosaukumi nesakrīt ar standartiem), bet *Oracle* nodrošina tikai divus standartu līmeņus – READ COMMITTED un SERIALIZABLE. Apmierinot standarta prasības šo divu līmeņu realizācija iekš DBPS ļoti atšķiras, kas noved pie vienādu transakciju atšķirīgas uzvedības dažādās DBPS. *DB2* paralēlas pieejas konflikts vienmēr noved pie bloķēšanas, bet *Oracle* parasti lasāma transakcija saņem datu veidu, kas tika fiksēts uz tās izpildes sākuma momentu. Izolēšanas nodrošinājumam *DB2* pielieto „klasisko” bloķēšanu mehānismu, *Oracle* pielieto vairākversiju modeli. Abas DBPS vienādi reaģē uz strupceļiem, pabeidzot vienu no konfliktējošām transakcijām.

Izolēšanas vadības iespējas *Oracle* var uzskatīt par elastīgākiem, jo izolēšanas līmeni var uzstādīt ne tikai visam seansam, bet arī atsevišķai transakcijai.

Piekļuves pārvaldība. Piekļuves pārvaldības koncepcijas *DB2* un *Oracle* ir pilnīgi dažādas. *Oracle* pilnīgi pārvalda drošību, tām sekojot lietotāji ir datubāzes objekti. *DB2* piekļuves pārvaldība integrēta ar drošības pārvaldību operētājsistēmā/tīklā, lietotāji un grupas reģistrējas aiz DBPS. DBPS objekti ir ieraksti par privilēģijām, kas tika dotas lietotājiem un grupām. No šejienes var rasties situācija: pārvaldības vidē (*DB2 Command Center*) var izveidot jaunu lietotāju un dot viņam piekļuves privilēģijas kādam objektam, bet šis lietotājs nevar paveikt autentifikāciju pie savienojuma ar datu bāzi, jo viņš nav pierēģistrēts sistēmā.

*Oracle* koncepcija realizē visu privilēģiju pārvaldību praktiski viendimensijas modelī. Ir tikai divu tipu privilēģijas (sistēmas un objektu privilēģijas) un visu viena tipa privilēģiju pārvaldības līdzekļi ir vienādi (GRANT/REVOKE). Pat divām privilēģiju tipiem tie atšķirās tikai detaļās.

*DB2* privilēģiju pārvaldība ir vairāk „tipizēta”. Katrai objektu klasei ir izdalīta sava privilēģiju kopa. Augšēja līmeņa objektiem privilēģiju jēdziens tiek aizvietots ar pilnvaru jēdzienu, kuru pārvaldība tiek veikta tikai ar ārējiem līdzekļiem attiecībā uz DBPS – lietotāju iekļaušana pilnvarotās administratīvās grupās. Pārvaldības līdzeklis - GRANT/REVOKE.

*Oracle* privilēģiju grupēšanu vada bez lomu palīdzības (datu bāzu objektu), un *DB2* ar lietotāju grupu palīdzību (kas tiek pārvaldīti aiz *DBPS*). Abos gadījumos tiek nodrošināta vienāda funkcionalitāte.

Pārvaldības vides – *Oracle Navigator* un *DB2 Control Center* ir praktiski vienādas pēc iespējām un pēc funkcionalitātes efektivitātes. Šo vidu nesimetriskās atšķirības ir piekļuves iespēja tabulu datiem iekš *Navigator* un funkcijas *SQL Assist* esamība iekš *Control Center*.

Interaktīva *SQL* vide *DB2 Control Center* ir bagātāka un ērtāka nekā *Oracle SQL\*Plus*. Tomēr *Control Center* ir resursu ietilpīgāka lietojumprogramma, un tās izmantošana uz mazjaudīgiem datoriem var izraisīt problēmas. *DB2* ir arī *Command Line Processor* un *Command Window* – interaktīvas *SQL* vides bez servisa, bet ar minimālām resursu prasībām.

## 1.4. *DB2* un *Oracle* salīdzināšana *Linux* sistēmā

Šajā nodaļā salīdzināsim datu bāzu programmatūras paketes *Linux* opērētājsistēmai: *Oracle 9i* versija 9.2 Enterprise un *IBM DB2* versija 8.1.2 ESE. Salīdzināšanu veiks, pamatoties uz īpašuma izmaksām, izmantošanas vienkāršības, drošuma un mērogošanas [5, 1.lpp.].

Modernā datubāze spēlē kritisku lomu jebkurā biznesā, lielajā vai mazajā. Spēja nodrošināt drošu, nebojātu un efektīvu datu apstrādi organizācijai ir liela prioritāte jebkuram IT vadītājam. Viņi izvēlas glabāt korporatīvus datus uz *Linux*-bāzētiem datu bāzu serveriem to spēka, drošuma, zemu izmaksu un pieejamības dēļ.

Līdz ar ātru *Linux* operētājsistēmu attīstīšanos, tika piedāvāts arī nodrošinājums *Linux* serveru datu bāzu programmatūrai. *IBM DB2* priekš *Linux* ļauj organizācijām izmantot *Linux* spēku un drošumu ar datu bāzu risinājumiem, ar racionālām un drošām cenām.

### 1.4.1. *Izmantošanas vienkāršība*

Datu bāzu administratori dod priekšroku *DB2* tāpēc, ka tā pieprasa mazāk sagatavošanas soļus nekā *Oracle* pirms programmatūra tiks uzinstalēta uz servera. *DB2* ir arī efektīvāka instalēšanas procesa gaitā, jo tā prasa mazāk soļu un mazāk laika, nekā lēnāka un apgrūtinošāka *Oracle* novietojums. Vienreiz uzinstalējot, *DB2* ļauj veikt līdzīgus uzdevumus ātri un vienkārši; *DB2* priekšrocība ir pat tad, ja darbojas ar lieliem datu bāzu klasteriem.

*DB2* labums ir vienkāršāka apmācība, samazinātas apmācības izmaksas un samazinātas atbalsta izmaksas. Apskatīsim sīkāk dažus salīdzināšanas aspektus:

Instalācija uz atsevišķas mašīnas [5, 1.lpp]. Instalācijas gaitā, *DB2* saglabā laiku un piepūles nodrošinot viegli izmantojamu uzstādīšanas procedūru. *DB2* aizņem pusi no soļiem un trešdaļu no laika lai uzinstalētu to pašu, ko prasa *Oracle*. *Oracle* pieprasa papildus soļus lai veidotu specializētu lietotāja kontus uz servera. Datu bāzu administratoram jāveido specialus diska uzstādīšanas punktus un jākonfigurē papildus vides mainīgie. Šīs apgrūtināšanas palielina instalēšanas izmaksas, atbalsta sabrukumu un instalācijas neveiksmes iespēju.

Klasteru instalācija [5, 1.lpp]. Daudzie modernie datu centri darbojas ar datu bāzu serveru klasteriem lai palielinātu veiktspēju un pieejamību. Klasteru pārvaldība ar *DB2* ir drošāka, ātrāka un efektīvāka nekā ar *Oracle*.

*Oracle* datu bāzes instalācija izmantojot *Real Application Clusters* (RAC) ir daudz sarežģītāka un prasa ievērojami vairāk laika nekā *DB2* dublikāts. *Oracle* RAC klasterizēta instalācija ir lēnāka, tai ir daudz atsevišķu instanču un tā prasa papildus komponentu instalāciju. Vismaz trijām galvenajām papildus programmatūras komponentēm jābūt uzinstalētām uz dalītas disku sistēmas un jābūt konfigurētām lai darbotos ar *Oracle* klasteru; tas ir papildinājumā sarežģītai pamatinstalācijai. *DB2* vienkārši pieprasa lai instalācija atkārtotos katra mezglā ar nedaudziem vienkāršiem soļiem lai konfigurētu klasteru.

Vēl *Oracle* RAC dokumentācija ir lielāka par 1000 lappusēm. Tas ir pārāk liels dokumentācijas apjoms administratoram, lai meklētu informāciju. Dokumentācijas izmērs arī rāda uz RAC tehnoloģijas izmantošanas sarežģītību.

Pretēji *DB2* klasterizēšanu viegli konfigurēt un tas prasa daudz mazāk dokumentācijas. Administratoriem ir labāk būt veiksmīgiem datu bāzu serveru klasterizēšanā ar vienkāršu un tīru *DB2* instalāciju.

Vienreiz uzkonfigurējot *DB2* servera klasteru, to ir vieglāk pārvaldīt ar *DB2* un spēcīgām grafiskām lietotāju saskarnēm administrēšanas rīkiem. Šie rīki padara serveru klasteru pārvaldību tikpat vienkāršu kā vienīga atsevišķa servera pārvaldība.

Administrācija [5, 2.lpp]. *DB2* administrēšanas rīki, kas ir bāzēti uz grafiskās lietotāju saskarnes dod vienkāršāku datu bāzes pārvaldības iespēju un rezultātu ar zemākām izmaksām.

Piemērām, *DB2* ļauj piekļūt lietotājam, instancei vai datubāzei caur grafiska lietotāja saskarnes rīku. *Oracle* datu bāzes administratoram jāatver konsoles logu un ar roku jāievada komandas. Šis gadījums ar *Oracle* palielina kļūdas varbūtību un nejaušas pārtrauces iespēju.

*DB2* arī nodrošina vairāk rīku datu bāzu administrēšanai. Konfigurēšanas konsultēšanas rīki un veiktspējas rīki ir pieejami tikai *DB2* datu bāzēm. *Oracle* sistēmai nav rīku, kas dublē nozīmīgu funkcionalitāti un tā paļaujas uz rokas optimizāciju vai vedņa tipa rīkiem, kas nav tik spējīgi.

*DB2* īpašības var rekomendēt vai pat veikt korektīvu darbību vairākās situācijās, piemēram tad, kad tabula paliek pilna ar atmiņas utilizāciju. Šī īpašība uzlabo datu bāzu administratoru produktivitāti un samazinās paļaušanās uz viņu specifiskām prasmēm.

#### **1.4.2. Īpašuma pilnas izmaksas**

*DB2* piedāvā nozīmīgu vērtību un dara to par mazāku maksu; tas samazina izdevumus un papildina *Linux* veiktspēju. *DB2* uz *Linux* ir pārāka par *Oracle* divās sfērās: administrēšanas un licenzēšanas izmaksas. Apskatīsim tās sīkāk:

Administratīvās izmaksas [5, 2.lpp]. *Linux* operētājsistēmā vidējas *Oracle* datu bāzes instalācija aizņem apmēram 20-30 minūtes vairāk nekā ekvivalenta *DB2* instalācija uz vienas un tās pašas aparatūras. Šis laiks atbilst tieši personāla apmaksai, kas asociētas ar instalāciju. Saliekot ar efektīvāku pirms instalācijas uzdevumu sarakstu, kas nodrošināts ar *DB2*, organizācija var izvairīties no nozīmīgas darbaspēka izmaksas no sava IT budžeta.

Iespēja ātri būvēt *DB2* *Linux* serverus samazina laiku, kas administratoriem jāpatērē, lai nodibinātu jaunu datu bāzes serveru. Tas samazina administratīvās izmaksas organizācijai ar datu bāzu serveru klasteriem.

Licenzēšanas izmaksas [5, 3.lpp]. Liela sākotnējo izmaksu daļa jebkurā datubāzē ir programmatūras licenzēšanas izmaksas. Papildus citām priekšrocībām *IBM* nodrošina klientus ar reālu finansu saglabāšanu, pērkot licences.

Organizācija, kas izvēlas *Oracle*, tērēs ievērojami daudz naudas iegūstot un uzturot datu bāzes programmatūru. Lielajam un mazajam biznesam, sevišķi, kad izmanto klasterizētas sistēmas, *IBM* licenzēšanas izmaksas ir daudz zemākas, nekā ekvivalentai *Oracle* programmatūrai. Dažreiz pirkt *DB2* ir trīs reizes lētāk nekā pirkt *Oracle*.

Piemēram, bāzējoties uz cenu lapas, kas bija pieejama 6. maijā 2003 gadā, *Oracle* IT nodaļai bija jāmaksā \$15,000USD lai licenzētu datu bāzes programmatūru atsevišķajam serverim priekš 50 cilvēkiem. Tāda pati instalācija, izmantojot *IBM DB2* priekš *Linux* programmatūras maksāja tikai \$5,449USD. Tas attēlo 63% samazināšanas programmatūras licenzēšanas budžetā šai instalācijai.

Izvēle izmantot *DB2 UDB Express* uz *Linux* piedāvā vairāk taupīšanas maza mēroga sistēmām. *DB2* ir racionālākas cenas un sniedzamie risinājumi neatkarīgi no datubāzes izmēra.

Aparatūras izmaksas [5, 3.lpp]. *Oracle* RAC tehnoloģija pieprasa augstākus aparatūras kapitālieguldījumus serveru klasteriem. Papildus tīkla pieslēgumi, serveri un dalīta diska apakšsistēmas ir pieprasītas lai izmantotu šo īpašību. Šīs papildus izmaksas palielina instalācijas izmaksas, ko izmanto *Oracle* tehnoloģija.

### **1.4.3. Drošība**

*Linux* nodrošina organizāciju ar drošu sistēmu ar augstu veiktspēju. Daudziem *Linux* serveriem tika mērīts darbības laiks mēnešos un gados. *DB2* palīdz iedarbināt augstu pieejamību uzticamā risinājumā. *Oracle* sistēmai ir aizsardzības un drošības problēmu skaits, kas padara to par sliktu izvēli uzņēmuma līmenim, it sevišķi klasterizētam datubāzes risinājumam uz *Linux*.

*Oracle* komponente *RAC* ir taisni saistīta ar vienu no nopietnajiem dienas garuma pārtraucēm tiešsaistes režīma avialīniju uzskaites sistēmai. Par spīti *Oracle* labākajiem atbalsta sasniegumiem, šīs vietnes atjaunošana pieprasīja varonīgus un riskantus lēmumus lai pārveidotu tiešsaistes režīma lietojumprogrammu. Pilna funkcionalitāte bija rekonstruēta tikai tad, kad lietojumprogramma tika mainīta lai likvidētu *RAC* tehnoloģiju.

*Oracle* atzīst, ka tai ir maza pieredze datu bāzu serveru lielajos klasteros *Linux* sistēmā, vai uz jebkādas citas platformas. Larrijs Elisons, no *Oracle* komandas, savā runā San Francisco *Linux World* saka „Mums nav daudz pieredzes ar reāli lielām produkciju sistēmām, vairākiem par 16 mezgliem”. Tas dod saprast IT darbiniekam, ka ir jāizvēlas mērogojumu, uzņēmuma līmeņa sistēmu [5, 3.lpp].

*DB2* nodrošina spēcīgākas klasterizēšanas tehnoloģijas, kas nosaka mērogu lineārā veidā. Tas ir daudz drošāk ar vienkārši izmantojamu saskarni, kas samazina pārtrauces operatora kļūdu dēļ un ļauj viegli pielāgot lietojumprogrammas mainīgiem ielādes nosacījumiem. *DB2* klasteri spēj ātri atjaunoties pēc neveiksmēm un nodrošināt atjaunošanas laikus salīdzinot ar *Oracle*, kas to var izdarīt par papildus samaksu.

#### 1.4.4. Mērogojamība

*DB2* mērogojamība [5, 4.lpp] uz *Linux* un *Oracle* klasteri tika salīdzināti, izmantojot publiski pieejamu veiktspējas informāciju. Ideālā vidē datu bāzes mēris lineārā veidā jo vairāk izpildes spēka tiek pievienots klasteram. *Oracle* demonstrē nelineāru mērogojamību, jo klasteram tika pievienoti serveri. Kā rezultāts, kapitālieguldījumi papildus serveriem strauji samazinās. Kaut gan, *DB2* demonstrē gandrīz lineāru mērogošanu, ka datu bāzes serveri tika papildināti klasteram. Faktiski *DB2* ir sertificēts ap 1000 mezglu klasteru izmantošanā. *DB2* palīdz padarīt datu bāzes serveru klasterus gandrīz bez maksas.

### 1.5. Jaunākais izstrādājums *DB2 9*

2006 gadā kompānija *IBM* ražoja produktu *DB2 Warehouse 9* [6] zem nosaukuma *Viper*, kas īstenoja visnozīmīgākos panākumus datubāzu tehnoloģiju attīstībā vairāk nekā par divdesmit gadiem. *DB2 9* izlaidums pabeidz *IBM* piecgadīgo projektu, kas pārver tradicionālas statiskās datu bāzu tehnoloģijas interaktīva, dināmiska datu bāzu servera koncepcijā, kas dod klientiem pilnvērtīgākas iespējas visu informāciju tipu pārvaldībai, tādu kā dokumenti, audio- un videofaili, attēli, tīmekļa vietnes un XML-transakcijas ar ciparu parakstiem. Jauns *IBM* datu bāzu serveris pirmo reizi šajā jomā ļauj nodrošināt relāciju un XML-datu brīvu, paralēlu kustību neatkarīgi no formāta, platformas un izvietojuma.

*DBPS DB2 9* tiek realizēti trīs nozīmīgie tehnoloģiskie panākumi – patentēta tehnoloģija *pureXML*, datu saspiešanas tehnoloģija *Venom* un autonomas datu pārvaldības priekšējie līdzekļi. Līdzīgs jaunu funkcionālu iespēju apvienojums, kas tika papildināts ar pilnvērtīgāku atjaunošanas spēju avārijas gadījumā, ļaus paātrināt informāciju vaicājumu izpildi un dos klientiem neierobežotu pieeju informācijai. Pie tām jauna *DB2* versija nodrošina pieeju informācijai, kas glabājas parastajās datu bāzēs *Oracle* un *MySQL*. *Oracle* un *MySQL* analogisko iespēju nepiedāvā.

*IBM* pirmo reizi izlaiž uzreiz trīs *DB2* redakcijas – korporatīviem klientiem, maza un vidēja biznesa kompānijām (*DB2 Express*) un izstrādātājiem (*DB2 Express-C*).

*IBM Information Management* nodaļas ģenerālais vadītājs saka: „*DB2 Viper* sola mainīt situāciju datu bāzu jomā. Mēs atvērām jaunu datu bāzu serveru tehnoloģiju ēru, kas nostiprinās mūsu investīcijas un iniciatīvas, vērstas uz klientu nodrošinājumu paplašināt biznesu, efektīvi izmantojot informāciju pēc vajadzībām” [6]. Klienti un partneri, kuri veica *DBPS DB2 9* iepriekšējo testēšanu un vērtēšanu, deva pozitīvas atsauksmes. *Zurich*

*Insurance, N.A.*, vadošais apdrošināšanu pakalpojumu piegādātājs, kas apkalpo lielas korporācijas un klientus no maza un vidēja biznesa sfēras, ir nolēmis izmantot *DB2 9* lai saņemtu pieeju jaunajai tehnoloģijai *pureXML*. *Zurich Insurance, N.A.* kompānijas datu bāzu vecākais administrators saka: „Apdrošināšanas jomā palielinājās vajadzības pēc dokumentācijas milzīgu masīvu glabāšanas XML formātā. *pureXML* tehnoloģija nodrošina augstu elastības līmeni definējot lietotāja funkcijas un izpildot ārējas operācijas ar XML-datiem, kas glabājas vecajās datu bāzēs. Iespēja glabāt vienu un to pašu XML-dokumentu dažādas versijas ar dažādām struktūrām vienā tabulā ļaus netērēt daudz laika izpildot ar roku izvilšanas un atkārtotas ielādes operācijas pie mūsu XML-struktūras izmaiņām” [6].

### **1.5.1. DB2 9 datu pārvaldības metodes**

DBPS *DB2* tiek realizēti daudzi pilnveidojumi un tehnoloģiskie panākumi, kurus nevar piedāvāt neviens no konkurentiem. Tie ir iespēja transformēt un pilnveidot glabāšanas, pārvaldīšanas un informācijas piekļuves metodes [6]. tajā skaitā:

Tehnoloģija *pureXML*. Tā ļauj klientiem caurspīdīgi pārvaldīt ne tikai parastus relāciju datus, bet arī XML-datus, bez nepieciešamības tos pārformatēt vai izvietot lielu objektu iekšienē datu bāzē. Šī unikālā *Viper* iespēja ļauj klientiem paaugstināt informācijas gatavības līmeni, darba ātrumu ar to un izmantot dažādus datus pie administratīvo izdevumu straujas ierobežošanas, kas ir raksturīgi informācijas pārvaldības esošām metodēm. *pureXML* tehnoloģija ļaus izstrādātājiem ievērojami atvieglināt un paātrināt pielikumu veidošanu, kam ir iespēja iegūt pieeju gan relāciju, tā arī XML-datiem.

Datu saspiešanas tehnoloģija *Venom*. Tā ļauj datu bāzu administratoriem izmantot zema līmeņa datu objektu kompresiju, panākot ievērojamu diska vietas ekonomiju, ievada/izvada operāciju un vajadzību pēc operatīvas atmiņas apjomu samazinājumu – it īpaši lielajām tabulām ar atkārtojamu datu secību. Pateicoties *Venom* tehnoloģijai DBPS *DB2* ļauj izmantot *Linux*, *UNIX* un *Windows* vidēs datu kompresijas līdzekļus, līdzīgus tiem, kurus izmanto uz lieldatoriem. *Venom* tehnoloģija papildina DBPS *DB2* tabulu vērtību kompresijas, indeksu kompresiju vairākdimensiju klasteros un datu kompresija rezerves kopēšanā esošas tehnoloģijas. Šī tehnoloģija kardinālā veidā atšķir *DB2* no *Oracle*, kas piedāvā novecojušos kompresijas līdzekļus tabulu līmenī, un no *SQL Server*, kas vispār nepiedāvā nekādu kompresijas nozīmīgu līdzekļu.

Glabāšanas resursu autonoma vadība. Tā piedāvā klientiem iespēju taupīt laiku un naudu pateicoties glabāšanas vadības uzdevumu automatizācijai, kas pašlaik prasa daudzas izmaiņu ieviešanas procedūras, ko administratori veic ar rokām.

Drošības līmeņa paaugstināšana. DB2 piedāvā piekļuves datiem unikālo vadības līdzekli – *Label Based Access Control* (LBAC), kas piedāvā ļoti elastīgas iespējas drošību hierarhiju noteikšanai un klasifikāciju līmeņu, lai pēc tam pielietotu šīs noteikšanās pārvaldāmai informācijai. Jauna iezīmju piešķiršanas iespēja piedāvā piekļuves konfidencialiem datiem, kas glabājas datu bāzē, vadības metodes. Tiek ieviests drošības administratora pilnvaru līmenis, kas apvieno vairākas privilēģijas drošības jomā zem viena lietotāja un nodrošina stingrāku kontroli par to, kas saņem piekļuvi informācijai.

Pilnveidotie atjaunošanas līdzekļi avāriju gadījumā. Tie paredz pārtrauktu operāciju atjaunošanas iespēju, ļauj taupīt laiku pie datu bāzes atjaunošanas. Tiek realizēts pāradresētas atjaunošanas procedūru izpildes paplašināts atbalsts ar scenāriju izmantošanu, kas tiek automātiski veidoti no esošiem rezerves paraugiem. Līdzīgas funkcionālas iespējas nodrošina DBPS DB2 lielu drošumu un atjaunošanas procesa elastību salīdzinājumā ar jebkuru citu datu bāzu serveru un piedāvā pilnīgāku lēmumu datu atjaunošanai.

Datu sekciju veidošanu priekšējās iespējas. Tie kļuva par svarīgu inovāciju datu pārvaldības pilnveidošanas un informācijas gatavības līmeņa paaugstināšanas jomā. DB2 9 ir pirmā DBPS, kas nodrošina triju izplatītu sekciju veidošanas metožu vienlaicīgu atbalstu: sekcionēšana pa diapazoniem, daudzdimensiju klasterizācija un hešings. Šī iespēja ļauj organizācijām izvēlēties sistematizācijas un savas informācijas sakārtošanas metodi, kas vislabāk atbilst biznesa individuālām prasībām, paaugstinot pielikumu produktivitāti.

Atmiņas automatizētas pārvaldības līdzekļi. Tie tiek piedāvāti visās DB2 9 redakcijās. Šī unikālā atmiņas pārvaldības iespēja nodrošina darba slodzes samazinājumu un var tikt izmantota sarežģītās vidēs ar vairākiem datu bāzu eksemplāriem, atvieglojot administrēšanas uzdevumus un ļaujot kvalificētiem datu bāzu administratoriem izpildīt jebkuru no atmiņas parametriem rokas uzstādījumu.

### **1.5.2. DB2 Viper salīdzināšana ar Oracle: kompresija un drošība**

Salīdzināsim DB2 un Oracle kompresijas [7, 7.lpp.] un drošības īpašības. Oracle ļauj kompresēt tabulas vai tabulu partīcijas. Kaut Oracle kompresē plaši izplatītas vērtības lapu līmenī, DB2 kompresēšana ir tabulu līmenī. Tas nozīmē kā atkārtotās vērtības atsevišķajā lapā būs aizvietotas ar simbolu Oracle sistēmā. Šis simbols tiek saglabāts lapas virsrakstā.

Tāpēc *Oracle* kompresēšanas pieejai ir sekojošie trūkumi:

- *Oracle* atbalsta kompresiju tikai vairākām ielādēm (bet *DB2* atbalsta kompresiju ielādēm, ierakstīšanai un importam);
- Ja šeit ir blīva vērtību atkārtošana caur visu tabulu, tad tie būs saglabāti daudz reižu katras lapas virsrakstā;
- Ja dati nav sakārtoti, tad varētu būt atkārtotie šabloni tabulā, bet ne uz katras lapas, tāpēc *Oracle* ignorēs šīs kompresijas.

Drošības jautājumā tiek salīdzināti *IBM* un *Oracle* korporāciju viedokļi.

*IBM* viedoklis par drošību [7, 23.lpp.] :

*IBM* drošība:

- Integrēta aparatūras un programmatūras drošība
- Resursu piekļuves vadības administrācija
- Viengabala šifrēšana un atslēgas vadība
- Mazāk nekā 10 drošības saistītu gabalu pēdējās 10 gados

*Oracle* drošība:

- *Oracle* problēmas izraisīja klientu prasības pēc drošības
- *Oracle* drošība nav integrēta ar operētājsistēmām
- *Oracle* nosūta vairāk nekā 100 gabalus vienā kvartālā
- Bieži labojumi nozīmē augstas administrēšanas un operāciju izmaksas.

*Oracle* viedoklis par drošību [8, 6.lpp]:

*IBM* drošība:

- Drošība ārpus datubāzes padara *DB2* viegli ievainojamu lietotājiem, kuriem izgāzās drošība
- Klienti pērk datu bāzi ar mazo ārpus-kastes drošību, un pēc tam papildina pirkumu ar drošības produktiem. Pieprasītie produkti un pakalpojumi rezultātā maksā dārgi
- Nav neatkarīgas *DB2* validācijas
- Augsta liela termiņa īpašuma izmaksas tāpēc, ka klientiem jāmaksā par datu bāzes produktu, drošības produktiem un pieprasītiem pakalpojumiem – modificēšanas un atbalsta pakalpojumi visiem šiem produktiem

*Oracle* drošība:

- *Oracle* nodrošina industrijas vadošus drošības īpašības datu bāzes produkta iekšienē, padarot tā, lai būtu grūti nogāzt drošību

- *Oracle* datu bāzes drošība atrodas klāt pati par sevi saprotama, bez pieprasījuma lai klienti licenzētu atsevišķus drošības produktus būtiskām, novērtētām drošības īpašībām
- Neatkarīgie drošības novērtējumi validē īstu drošības implementāciju *Oracle* relāciju datu bāzu pārvaldības sistēmās
- Klientiem nav obligāti ne pirkt papildus produktus atslēgas drošības īpašībām, ne maksāt par šādu produktu modificēšanu un atbalstu.

## **1.6. *Oracle* jaunākais izstrādājums *Oracle Database Vault***

*Oracle Database Vault* [9] aizsargās korporatīvo datu bāzi no iekšējiem uzbrukumiem.

*Oracle* kompānija ražoja produktu *Oracle Database Vault* - vadošais aizsardzības un pieejas ierobežošanas līdzeklis svarīgiem datiem un lietojumprogrammām. Aizsardzības mehānismi, kas realizēti produktā, bāzējas uz likumu izmantošanu uz aizsardzības jomām.

Likumi nozīmē lietotāju pilnvaru ierobežošanu, uzskaitot viņa amata pienākumus un jomas ļauj dot pieeju specifiskām lietojumprogrammu kopām un datu bāzu objektiem. Šo divu pieeju esamība ļauj organizēt maksimāli elastīgu pieejas kontroli. Piemēram, administrators nevar ieviest izmaiņas datubāzē, kad atrodas aiz Intranet korporatīva tīkla, kā arī īstenot jebkuras citas operācijas ar noliktavu ārpus darba laika.

*Oracle Database Vault* nepieļaus informācijas zudumus, izslēdzot nesankcionētas pieejas lietotāju konfidenciāliem datiem, kuriem ir īpašas pilnvaras (piemēram, datu bāzes administrators). Organizācijas vadība var ieviest likumus, saskaņā ar kuriem uzdevumu risināšanai, kurām domāta pieeja noslēpuma informācijai, būs vajadzīga divu darbinieku esamība. Tādā veidā, kompānija būs droši aizsargāta no ļaunprātīgiem darbinieku darbībām un iekšējiem uzbrukumiem.

Produkts garantē drošības izplatītu normatīvu ievērošanu. Šie normatīvi ir: *Sarbanes-Oxley Act*, *Health Insurance Portability and Accountability Act*, *Japan's Personal Information Protection Act* un *European Union Privacy and Electronic Communications Directive*.

## 1.7. Lietotāju atsauksmes

### 1.7.1. Lietotāju atsauksmes par lieldatoriem

„Neskatoties uz visu tehnoloģisku progresu, kas gāja no 70-to gadu paša sākuma, tie paši *IBM* lieldatori un tās pašas 70-to gadu programmas arī tagad kalpo ASV vairākos uzņēmumos.

Tādai parādībai ir daži iemesli. Pirmkārt šādās sistēmās tiek uzkrāti neizvietoti vairāku gadu desmitu dati un zināšanas, kas ir dārgi un grūti pārnest jaunās datorsistēmās un struktūrās.

Otrkārt, tieši šīs mantotas sistēmas piedāvā lielāko daļu visvērtīgākai analīzei. Tagad viss, kas tika veikts ar *IBM DB2* u.c. sistēmu līdzekļiem, savākts uz magnētiskām lentām skaitļošanas centros. Tie ir vērtīgākie dati, kas paliks par derīgiem bieži un vienmēr. Un tāpēc paliek skaidrs, ka pat nelieli pārtraukumi vai aizkavējumi izmantošanā, ko mantoja dati, maksā pārāk dārgi; to neattaisno nekāda tālāka operativitāte vai datu pārnese uz jaunākām ergonomiskām platformām” [10].

Analītiķi no *Robert Frances Group* pārliecināti, ka lieldatora tehniskās priekšrocības pārvērš to par informācijas kodolu, kuram pie tam ir enerģiju lietošanas un atdzesēšanās priekšrocības. Pēc analītiķu viedokļa:

„Glabāšanas serveru/sistēmu konsolidāciju tehnoloģiju, kā arī sadalījums nodaļās un virtuālizācijās jaunā paaudze – tā ir modeļa pareizības kārtējais apliecinājums, kas tradicionāli tiek izmantots uz lieldatoriem. Bez tā IT-vadītājiem jāievēro, ka lieldators – tas ir optimāls līdzeklis pielikumu hostingiem un centrāla mezgla realizācijai, kas nodrošina informācijas drošību, servera pūla pārvaldību un darba slodzes/datu konsolidāciju” [11].

*Robert Frances Group* izdarīja sekojošus secinājumus:

Liieldatoriem ir unikālie tehniskie raksturojumi: tā ir visjaudīgāka komerciāli pieejama skaitļošanas platforma kopīgiem mērķiem, liieldatoriem ir daži iebūvēti specializēti procesori ievada/izvada pārvaldībai, utt. [11].

Programmu produkcija *IBM DB2* daudzu gadu garumā ir līderis DBPS vidū. Darba slodzes izpilde *DB2* liieldatorā – pats par sevi ir pietiekams pamats lai turpinātu liieldatoru lietošanu. Rezerves kopēšanas/atjaunošanas jaudīgas iespējas, jaudīga funkcionalitāte un datu aizsardzības augsts līmenis, cenu priekšrocības – tās ir iespējas, kas *DB2* attīstītas labāk nekā *Oracle*.

### ***1.7.2. Lietotāju atsauksmes par īpašuma izmaksām***

2002 gadā saskaņā ar vadošas analītiskas firmas publicētu atskaiti, datu bāzu pārvaldības sistēma *IBM DB2* kļūst par pievilcīgu platformu elektroniska biznesa kompānijām, kas tiecās pēc datubāzu administrēšanas uzlabošanas un pēc īpašuma pilnas izmaksas samazināšanas. Arvien lielāks kompāniju skaits grib nodrošināt maksimālu finanšu atdevi no saviem ieguldījumiem informāciju tehnoloģijās, kā arī izmantojamo produktu ilgtermiņa stabilitāti datu pārvaldībai. DBPS elektroniskā biznesā, tādas kā *DB2*, svārgaku lomu spēlē kompānijas elektroniska biznesa infrastruktūras fundamentā [12].

*D.H. Brown* kompānijas atskaitē, kas saucās „*IBM DB2 Universal Database vs. Oracle 9i: Total Cost of Ownership*” [13] (*IBM DB2 UDB* un *Oracle 9i*: īpašuma pilnas izmaksas salīdzināšana) tika izanalizēti dažādi faktori, kas ietekmē uz īpašuma pilnām izmaksām priekš *DB2* un *Oracle 9i*, tajā skaitā: datubāzu administrēšanas efektivitāte, vadība, produktivitāte un cena. *D.H. Brown* analītiķi nāca pie secinājuma, ka *DB2* visās konfigurācijās ar ārējo izvēršanu Internetā iznāk uz pusi lētāk, nekā *Oracle*, piedāvā pasūtītājiem lielu efektivitātes līmeni pie datubāzu pārvaldības un no ekonomiskuma redzes viedokļa ir rekomendēta izvēle uz piecu gadu periodu. Īpašuma pilnas izmaksas *DB2* pasūtītājiem ir zemākas par *Oracle 9i* par 30% [13].

Atskaitē arī tika izanalizēti divu risinājumu raksturojumi un priekšrocības. DBPS *IBM DB2* pārsniedza *Oracle 9i* tādos aspektos kā uzstādīšanas vienkāršība, vaicājumu optimizācija un kopīga sadalītu datu bāzu arhitektūra. Šīs priekšrocības piedāvā pasūtītājiem augstāku automatizācijas līmeni, kas glābj datubāzu administratoru no tehniskām sarežģītībām, pateicoties tam daudzus uzdevumus var dot mazāk kvalificētam administrācijas personālam, bet vaicājumu pārvaldību – gala lietotājiem.

Ņujorkas kompānija *Clarity Incentive Systems*, kas speciālizējas uz maksājumu sistēmu izstrādes un maksājumu transakciju apstrādes attīstītājiem un šauri speciālizētiem tirgum, izmanto DBPS *IBM DB2* rēķinu, transakciju un maksājumu apstrādes sistēmu lietotāju informāciju pārvaldībai. *Clarity* izvēljas *DB2* pēc zemas īpašuma pilnas izmaksas, augstās vadības iespējām un drošuma.

„Viens no mūsu galvenajiem uzdevumiem ir tas lai izvēlētos DBPS ar vismazākajām īpašuma pilnām izmaksām – un mēs izvēlamies *DB2*, - saka Karls Enslijs, kompānijas *Clarity* informācijas dienesta direktors. – Izvēles procesā mēs konstatējam, ka *DB2/Linux* risinājums, kas maksā 250000 dolaru strādā ātrāk un labāk nekā *Oracle/Solaris* risinājums par 1,5 miljoniem, un tagad mēs jūtam nozīmīgu ekonomiju. Bez tam, ar izpildes

paaugstinājuma un vaicājumu optimizācijas tehnoloģijas iebūvētām funkcijām mums pietiek ar mazāku datu bāzu administratoru skaitu” [12].

*D.H. Brown* atskaitē arī tika minētas sekojošās *DB2* priekšrocības pret *Oracle 9i* [13]:

- Datu bāzu administratora darba efektivitāte: *DB2* nodrošina augstāku automatizācijas līmeni salīdzinot ar *Oracle 9i*. Kopumā, pateicoties *DB2* efektivitātei operatīvas transakciju apstrādes sistēmām tiek nodrošināta ikgadēja darba resursu ekonomija 6% izmērā, lielajām OLTP sistēmām – 15%, datu bāzēm ar Interneta atbalstu – 20%, datu noliktavām – 18%. Bez tam, klasteru atbalsta iespēju realizācijai *DB2* prasa ap trešdaļu cilvēk-stundu, kas tiek prasīts *Oracle 9i* ekvivalentai *RAC* uzstādīšanai. *DB2* arī piedāvā 25% priekšrocību kadru aspektā uz darba apjoma samazināšanas un nepieciešamas apmācības rēķina [13, 2.lpp].
- Īpašuma pilnas izmaksas piegādes līmenis: Programmatūras nodrošinājuma kopīgo izmaksu, apkalpojumu un darba resursu analīzei tika apskatīti trīs īpašuma pilno izmaksu scenāriji: OLTP sistēmām, lielajām OLTP sistēmām un Internet pielikumiem. *DB2* atrada īpašuma pilnas izmaksas priekšrocību visos trijos scenārijos: izmaksu līmenis izrādījās par 20-30% procentu zemāks nekā *Oracle* [13, 2.lpp].
- Komplektēšana: *DB2* pievieno papildus izmaksas par biznesa zināšanu atbalsta, savienojumu ar lieldatoriem un lietojumprogrammu izstrādes sākuma līmeņa produktu iekļaušanu pamatkomplektā [13, 1.lpp].

### **1.7.3. Par darbu ar kanālu**

Reselleri ir apmierināti ar DBPS *IBM DB2* kvalitāti un drošumu, bet tās darbs ar kanāliem [14] programmās patīk tiem vēl vairāk.

*IBM* ir kļuvusi pirmā sešos no astoņiem kritērijiem darbam ar kanālu – tas ir pietiekams, lai nodrošināt tai izšķirošu priekšrocību kanāla apmierinātībā un labāku atzīmi 75,7 punktu pret konkurentiem (*Oracle*, *Microsoft* un *MySQL*).

Sans Patalano, „*en technologies*” integratoru kompānijas prezidents saka: „*IBM* darba ar kanāliem programma – tas ir labākais, ar ko man bija darīšana 20 gadu laikā. Tika prasīts daudz laika, lai to panāktu” [14].

Īpaši augstās atzīmes *IBM* saņēma pēc tādiem kritērijiem, kā reselleru informēšana, partneru programmu kvalitāte, atbalsts no piegādātāja un atgriezeniska saite.

Kad trīs gadu atpakaļ *IBM* sāka runāt par to, lai vērstu visas pārdošanas vidējā un mazā biznesā caur kanālu, Patalano pret tām attiecās skeptiski. Viņš saka: „Viņi pierādīja, ka mums nebija taisnība: *IBM* pārvadā caur kanālu visu vidējam un mazajam biznesam” [14].

Brajans Boksmans, kompānijas *Syscom* realizācijas un marketinga vice-prezidents saka: „*IBM* nodrošina vispusīgu atbalstu, vai nu tā ir pirmspārdošanas vai pēcpārdošanas vai saistīta ar pasūtītāja atbalstu. *IBM* mūs atbalsta visā, ko mēs prasām” [14].

Tehniskā līmenī *IBM*, *Oracle* un *Microsoft* ir nopietni konkurenti. Kaut *Oracle* šajā līmenī pārvārēja *IBM* par 0,2 punktiem pateicoties nozīmīgai priekšrocībai mērogojamībā, tomēr *MySQL* un *Microsoft* izrādās stiprākie cenas/izpildes attiecībā, un *IBM DB2* saņēma visaugstākās atzīmes par kvalitāti un drošumu.

Boksmans saka: „*Syscom* izmanto *DB2* tāpēc kā tai ir funkcionāla savienojamība ar citām *IBM* lietojumprogrammām, tajā skaitā *Content Manager*, kas padara to mērogojamu. Kritiski svarīgajās lietojumprogrammās, kad tā ir pareizi ieviesta, tā ir visspēcīgāka. Mūsu lietojumprogrammas ir ļoti noslogotas ar datiem un lielu saturu, un mums ir vajadzīgs mērogojams, funkcionāli saderīgs risinājums, kas ir spējīgs apstrādāt milzīgus datu apjomus” [14].

Uzskatāmības pēc apskatīsim tabulu, kur tiek uzskaitīts tehniskais līmenis, darbs ar kanālu un kopīgais vērtējums:

	Tehniskais līmenis	Darbs ar kanālu	Kopīgais vērtējums
IBM	82,2	71,5	75,5
Oracle	82,4	68,7	74,0
Microsoft	81,0	68,6	73,3
MySQL	77,9	65,7	70,4

## 2. DB2 DWE EDITION FUNKCIJAS

*IBM DB2 DWE* nodrošina visu, kas ir vajadzīgs elastīgas, mērogojamas datu noliktavas realizācijai, lai veidotu dinamisku datu noliktavu. *DB2 DWE* ir sekojošās īpašības:

- Tā vienkāršo korporatīvas datu noliktavas izstrādi, izvēršanu un uzturēšanu ar aptverošu integrētu risinājumu;
  - Vienkāršo BI procesu ar gatavām analītiska risinājuma komponentēm;
  - Uzlabo datu noliktavas produktivitāti ar spēcīgam *DB2* optimizācijas īpašībām.
- [15]

Nodrošinot daudz iespēju, kas neietilpst tradicionālās noliktavās, *DB2 DWE* ir plaša platforma, kas palīdz datu noliktavas arhitektoriem un administrātoriem efektīvi nokonstruēt, izvērst un uzturēt korporatīvo datu noliktavu. Šīs iespējas ir:

- DB2 datu servera bāze. *IBM DB2 9* platforma ir *DB2 DWE* risinājuma pamats. *DB2 9* nodrošina augstu izpildes efektivitāti dažādas slodzes vaicājumu apstrādei. *DB2 9* ir risinājums dināmisko datu noliktavu veidošanā, jo tam ir tādas īpašības kā datu sadalīšana, jaunu rindu kompresija, daudzdimensionāla klasteru veidošana un materializētas vaicājumu tabulas.
- Modelēšanas un konstruēšanas rīki. *DB2 DWE* iekļauj modelēšanas un konstruēšanas rīkus, pamatoties uz sadarbību ar *IBM Rational Data Architect* programmatūru, kas ļauj konstruēt, modelēt un apgriezt fiziskās datubāzu shēmas. Tas nodrošina daudz iespēju, iekļaujot fizisko datu modelēšanu vienai datubāzei vai veselai korporācijai; modeļu validāciju; atbilstības analīzi; izmaiņu pārvaldību.
- Iebūvēta datu pārvietošana un transformācija. Tas ir grafisks rīks datu noliktavas konstruēšanai un administrēšanai. Rīks piedāvā datu plūsmu, vadības un transformācijas operatoru paleti, kas ir nokompilēti iekš *DB2* specifiskām SQL operācijām un var tikt izpildītas iekš noliktavas. Šis rīks nodrošina datu noliktavas analītiskas struktūras sagatavi un aizpildīšanu, kas ir nepieciešams datu izracei, daudzdimensionālai un iebūvētai analīzei.
- Datu izrace un vizuālizācija. *DB2 DWE* ir iekļautas spēcīgas datu izraces iespējas lai nodrošinātu integrēto analīzi korporatīvā datu noliktavā. *DB2 DWE* izstrādes vidē ir atbalstīti standarti datu izraces modelēšanas algoritmi (klasteru veidošana, asociācijas, klasifikācija un prognozēšana). Šie algoritmi var tikt importēti PMML standartā formātā no trešdaļas modelēšanas rīkiem. Datu izraces modeļi var tikt izpildīti tehniskā vidē, lai nodrošinātu reāla laika datu

ierakstu skaitīšanu. Papildus tam vēl ir nodrošinātas prezentāciju komponentes lai veidotu datu izraces rezultātu vizuālo analīzi.

- Tiešsaistes analītiska apstrāde (OLAP), konstruēšana un optimizācija. *DB2 DWE* nodrošina tiešo atbalstu optimizētai OLAP analīzei pret dinamisku datu noliktavu. *DB2 DWE* programmatūras rīki ļauj veidot, rediģēt, eksportēt un izvērst OLAP modeļus virs relāciju datu noliktavas shēmas. Tas arī veido optimizācijas rekomendācijas lai uzlabotu integrēto OLAP rīku izpildi. Šīs īpašības nodrošina dziļu integrāciju ar iebūvētām analītiskām iespējām un relāciju kubu veidošanu optimizētai pārlūkprogrammu bāzētai iebūvētai OLAP analīzei.
- Iebūvēta analīze. *IBM Alphabox* programmatūra nodrošina integrētas, tīmekļa izvietotas analītiskas lietojumprogrammas, kas var nodrošināt pareizu daudzdimensionālu vaicājumu un rēķināšanas atbalstu visos datu noliktavas slāņos. Relāciju kubu serveris un cieša integrācija ar *DB2 DWE OLAP* rīkiem un datu izraces īpašībām ļauj *Alphabox* programmatūrai atbalstīt *end-to-end* analīzi dinamiskās datu noliktavas risinājumam.
- Uzlabota kompresija. Ar *DB2 DWE* var uzbūvēt unikālu atmiņas optimizācijas tehnoloģiju iekš *DB2 9*, kas samazina gan izmaksas, gan diska vietu, kas ir asociētas ar relāciju datu glabāšanu. Ekonomija no šīs koncepcijas var būt daudznozīmīga: testi uz izmantotiem datiem datu noliktavu etalonā demonstrēja no 45 līdz 69 procentu ekonomijas diska vietā. Labāka diska utilizācija un vaicājumu ātruma uzlabošana tika pievienoti kā kompresijas labums.
- Slodzes kontrole. *DB2 DWE* nodrošina SQL vaicājumu izpildes kontroli un pārraudzības rīkus, lai efektīvāk kontrolētu datu noliktavas slodzi. Caur vaicājumu klasifikāciju un sliekšņu profiliem, SQL vaicājumi var būt prioritizēti, noplānoti un izpildīti tādā veidā, kas optimizē pieejamās sistēmas resursus un caurlaidspēju. [15]

*DB2 DWE* piedāvā piecas funkciju kategorijas [16]:

- DWE veidošanas funkcijas: darbojas ar avotu un mērķa metadatiem, veido fizisko datu modeli. Fizisko modeli izmanto citās funkcijās.
- Datu pārveidošanas funkcijas: Izvērš, modelē un pārrauga datu noliktavas lietojumprogrammas, kuras veido iekš *Design Studio*. Apskata statistiku un ieraksta žurnālā to, kas saistās ar procesiem un izpildes kļūdām.
- OLAP funkcijas: Importē un eksportē kubu modeļus, izmanto OLAP. Parāda kuba modeļa metadatu saturu - tabulas, savienojumus, mērījumus un atribūtus.

- Izraces funkcijas: Atbalsta datu izraces modeļus un darbības ar tiem. Izraces vizualizācijas rīks nodrošina grafisku rezultātu reprezentāciju izraces modelim.
- Iekļautas analīzes funkcijas: Veido vizuālas komponentes un lietojumprogrammas, kas apstrādā citu komponentu analītiskas struktūras. Analītiskas komponentes ir iebūvētas tīmekļa programmas.

## 2.1. DWE veidošanas funkcijas

Datu struktūru jebkurai biznesa risinājumam būvē ar datu modeļiem. Ir trīs datu modeļu tipi: konceptuāla, loģiska un fiziska. Īsi apskatīsim katru datu modeli:

- Konceptuālais datu modelis ir augsta līmeņa biznesa skats uz datiem. Tas identificē datu entītijas un relācijas starp tiem, kurus izmanto biznesā. Tas palīdz saprast un izstrādāt loģisko un fizisko datu modeļus.
- Loģiskais datu modelis ir datu modeļa definīcija. Šajā modelī identificē datu entītijas un visas relācijas starp tām. Entītijas apraksta ar specifiskiem atribūtiem, datu tipiem un atslēgām.
- Fiziskais datu modelis ir loģiska modeļa fiziska realizācija datu bāzu pārvaldības sistēmā. Tas nosaka konfigurāciju, atslēgas, indeksus un ierobežojumus. Šīs detaļas glabājas datubāzes struktūrās. [17, 4.nod.]

Fizisko modelēšanu veic ar IBM *Rational Data Architect* [17, 2.1.1.sad.] rīku. To izmanto lai izstrādātu un uzturētu fiziskus datu modeļus. Lietojumprogrammu izstrādātāji un datu noliktavu arhitekti izmanto šo rīku darbam ar fiziskiem datu modeļiem avotu un mērķa datubāzēm.

Fiziskais datu modelis – tas principā ir metadati, kas attēlo datubāzes fizisko raksturojumu. Izmantojot datu modeļa grafisko redaktoru, var izveidot datu modeli. To arī var veikt otrādi – izveidot datubāzi un no izveidotas bāzes noģenerēt fizisko modeli. Ja datubāzē kaut kas pamainījas, tad var ar salīdzināt datu modeli ar datubāzi un ģenerēt tikai izmaiņas.

Fizisko modeli var izmantot arī lai nodrošināt metadatu informāciju citām DWE funkcionālām komponentēm, tādām kā *SQL Warehousing* (SQW) rīks.

Fiziskais modelis definē datu avotu shēmu. Datu modelis attēlo datu tabulas, kolonnas un attiecības starp tabulām. Iekš DWE Design Studio ir komponentes, ar kurām veido fizisko modeli un tad ģenerē SQL skriptus no šī modeļa.

Fizisko modeli ierobežo saskaņā ar mērķa koncepciju. DWE fiziskie modeļi ir ierobežoti saskaņā ar relāciju modeli. Var modelēt tikai tos objektus, kurus atbalsta mērķa datubāze.

Fiziskus modeļus var izmantot lai veidotu jaunas shēmas vai mainītu jau esošās shēmas datubāzē.

Fizisku modeļu elementi ir:

- Datubāzes (viena datubāze uz datu modeli)
- Tabulas
- Skati
- Primāras atslēgas un ārējas atslēgas
- Indeksi
- Glabājamās procedūras un funkcijas
- *DB2* tabulu telpas un bufera pūli [17, 4.1. sad.]

Fiziskais modelis, kas izstrādāts iekš *Design Studio*, parasti ir realizēts kā entītiju-relāciju diagramma. Modeļi var vizuāli attēlot ar *IE* vai *UML* palīdzību.

## 2.2. Datu pārveidošanas funkcijas

Būvējot BI platformu, ir svarīgi pārvaldīt un pārvietot datus apkārt datu noliktavas, kā arī pārveidojot datus pēc savām vajadzībām. SQW [17, 2.1.1. sad.] ir grafisks rīks, ar kura palīdzību var veikt šos uzdevumus. To izmanto lai izveidotu datu kustību un transformāciju plūsmas datu noliktavā. SQW rīks automātiski ģenerē SQL kodu, pamatoties uz vizuāla operatora plūsmām, kurus modelē iekš DWE *Design Studio*.

SQW ir divi plūsmu pamattipi: datu plūsma un vadības plūsma.

Ar SQW var izstrādāt loģiskus plūsmu modeļus. Šie modeļi attēlo darbības, kuras ir jāpielieto datiem, kad tie pārvietojas no avotiem uz mērķa vietu. Šīs plūsmas ir datu plūsmas [17, 6.1.1. sad.]. Vairākumam datu plūsmu ir avoti un mērķa vietas, kurus attēlo relāciju tabulas datu noliktavā. Avoti un mērķa vietas var arī būt dažādi faili un attālinātas relāciju datu glabātuves. Operatori datu plūsmā attēlo augstāka līmeņa darbību, kura jāpielieto datiem. No šī modeļa ģenerē *DB2* SQL kodu. Šim kodam veido izpildes plānu. Šis ģenerētais SQL kods ir izpildāms *DB2* mašīnās. Ir arī specializētas datu plūsmas, ko sauc par izraces plūsmām. Izraces plūsmas palīdz sagatavot datus datu izracei, izpildīt datu izraces algoritmus un strādāt ar datu izraces modeli.

Var izstrādāt vairākas datu vai izraces plūsmas kādai noliktavas lietojumprogrammai. Tā var saturēt dažas atkarības, piemēram, izpildot kādu konkrētu plūsmu, būs jāizpilda citas

plūsmas. Iekš DWE ir grafiskais redaktors, kurā var definēt saistītas datu plūsmas un izraces plūsmu izpildes kārtību. Tās ir vadības plūsmas [17, 6.1.1. sad.]. Vadības plūsmas var noteikt datu un izraces plūsmu kārtību un izpildīt citu datu apstrādes funkcijas: operētājsistēmas skriptus, DB2 SQL skriptus, FTP, e-pasta atsūtīšanas paziņojumus, un citas. Vadības plūsma satur datu plūsmu un ne-datu plūsmu darbības. Piemēram, komandas izpildi un faila sagaidīšanu. Vadības plūsma arī ģenerē vadības plūsmas plāna izpildes grafu, kur tiek nolasīti un izpildīti vajadzīgie mezgli, atkarībā no iepriekšējās darbības rezultāta.

DWE SQW arī nodrošina servera komponenti, kas atbalsta datu pārvietošanas un pārveidošanas plūsmu izvietojumu, izpildi un pārvaldību izpildlaika vidē. Tāda vide var būt tests, kvalitātes nodrošināšanas sistēma vai produkciju sistēma. Izpildlaika vides lietotāja saskarne ir tīmekļa pārlūkprogramma. Šī programma nodrošina piekļuvi prasītām funkcijām lai pārvaldītu DWE izpildlaika vidi un iekļauj datu pārvietošanas un pārveidošanas plūsmu pārvaldības funkcijas.

DWE SQW ir arhitektūras komponentes gan DWE izstrādes vidē, gan DWE izpildes vidē. SQW izstrādes komponentes ir daļa no DWE integrētas izstrādes vides, DWE Design Studio.

SQW elementi iekš *Design Studio* ir datu plūsmu redaktors un vadības plūsmu redaktors. Šie grafiskie redaktori sastāv no ikonām, kas attēlo vairākus operāciju tipus. Operācijas ir saistītas ar plūsmu bultiņām, tas attēlo datu plūsmu vai pārvaldību starp operācijām. Plūsmas informācija vai īpašības glabājas DWE *Design Studio* metadatu pamatstruktūrā.

Izpildlaika videi ir sekojošās iespējas:

- Komunikācijas ar lietotāju DWE administrācijas konsolē un administratora vaicājumu izpilde;
- Metadatu pārraudzība noplānota procesa izpildei, kas ir bāzēts uz darbības noplānota laika vai pabeigšanas, un darbības iesniegšana serverī;
- Izpildāmo darbību pārraudzīšana, to metadatu izpildes un pabeigšanas ieraksts žurnālā. [17, 6.1.2. sad.]

SQW izpildlaika vidē metadati glabājas DB2 relāciju datu bāzē. Šie metadati sastāv no informācijas par datu, datu izraces, vadības plūsmām, datu savienojamības informācijas serveru informācijas, plānošanas informācijas, izpildlaika statistikas utt.

Ir arī pārveidošanas serveri. SQW darbību gadījumā tas var būt DB2 datu bāzes serveris, kas ir iezīmēts kā izpildes datubāze.

Ar SQW izstrāde parasti notiek sekojoši. Izstrādā datu pārvietošanas un pārveidošanas plūsmu kopu iekš DWE *Design Studio* un tad izveido izvēršanas paketi. Šī izvēršanas pakete

var tikt izvērsta vairākās izpildlaika vidēs. Izvēšanas pakete ir izstrādāta vienu reizi iekš DWE *Design Studio*. Tas ir iteratīvs process, un kad tas būs pabeigts, izvēšanas pakete būs izveidota. Pirmo reizi tas ir izvērsts līdz sistēmas testam vai kvalitātes nodrošināšanas videi lai to notestētu. Ja tests veiksmīgi pabeidzas, tad tā pati izvēšanas pakete tiks secīgi izvērsta līdz produkcijai [17, 6.1.3. sad.].

### 2.3. OLAP funkcijas

DWE OLAP [17, 5.2. sad.] ir pievienojama DB2 UDB īpatnība. Tā uzlabo DB2 UDB spēju izpildīt OLAP apstrādi. DWE OLAP vēl sauc par DB2 *Cube Views*. Ar DWE OLAP datubāze kļūst daudzdimensionāli informēta, pievienojot OLAP metadatus DB2 UDB katalogu tabulām. OLAP metadati apraksta relāciju tabulu dimensiju struktūru. Metadati nodrošina *DB2 UDB* pamatu kas tos optimizē priekš OLAP rīkiem un lietojumprogrammām.

Ir rīks, kas optimizē analīzē dimensiju modeli un rekomendētus agregācijas tabulas, kas ir pazīstami kā DB2 MQT. Šis rīks ir DWE OLAP Optimization Advisor. MQT satur aprēķinātus datus, kas atbilst OLAP struktūrām. Tas uzlabo OLAP rīku un lietojumprogrammu vaicājumu izpildi.

DWE OLAP ir integrēta pieeja OLAP, kas uzlabo DB2 vaicājumu izpildi. DWE OLAP ir sekojošās iespējas:

- Palīdz DB2 UDB optimizētājam pārrakstīt ienākošus vaicājumus uz MQTs, kurus rekomendē DWE OLAP.
- Palīdz lietojumprogrammām un rīkiem apstrādāt DB2 dimensionālus datus.
- Būvē visus vaicājumus, kas izmanto agregātus datus, ieskaitot OLAP veida vaicājumus, MOLAP kubu ielādes vaicājumus un relāciju tabulu eksromptu analīzi iekš DB2 UDB.

Ir daži iemesli, kāpēc organizācijas izvēlas DWE OLAP:

- Izmanto SQL datu noliktavu projektu izvēšanā
- Atbalsta skaitļošanas rīkus, kas nodrošina komponentu sadarbību
- Dod centralizētas OLAP metadatu pārvaldības iespēju
- Uzlabo datu noliktavas izpildi.

DWE OLAP apraksta metadatu objektus līdzīgi daudzdimensiju modeļu struktūrām atsevišķajos zvaigznes shēmās un sniegpārslas shēmās.

DWE modelī tiek izmantoti sekojoši metadatu objekti [17, 5.2.1. sad.]:

- Kuba modelis - zvaigznes vai sniegpārslas shēmas attēlojums. Kuba modelis sagrupē vajadzīgus dimensiju objektus apkārt centrālam faktam. Tas

apraksta visus datus, kas ir saistīti ar mērījumu kolekciju. Datubāzē kuba modelis saistās ar zvaigznes shēmu vai ar sniegpārslas shēmu. Kuba modelis atsaucās uz vienu faktu objektu un vienu vai vairākām dimensijām.

- Faktu objekti – sagrupē saistītus mērījumus specifiskām lietojumprogrammām. Kuba modelī faktu objektus izmanto kā zvaigznes vai sniegpārslas shēmas centru. Faktu objekti glabā informāciju par atribūtiem un par atribūtiem ar savienojumiem. Atribūtus izmanto faktu-dimensiju savienojumos. Atribūtus ar savienojumiem izmanto lai sakārtotu mērījumu atbilstības starp vairākām datubāzes tabulām. Pie tam faktu objekts glabā sevī atribūtu un savienojumu kopu.
- Mērījums – definē izmēra entītiju. Mērījumu izmanto faktu objektos. Mērījumi apzīmē kolonnas ar skaitliskām vērtībām faktu tabulā.
- Dimensija – saistīto atribūtu kolekcija, kas kopā apraksta datu aspektu. Atribūti no vienas vai vairāku dimensiju tabulām atsaucās uz dimensiju. Tomēr, ja izmanto atribūtus no daudzdimensiju tabulām, tad starp tām tabulām jābūt savienojumiem. Un tad šiem savienojumiem jāatsaucas uz dimensijām. Uz dimensiju var atsaukties viena vai vairākas hierarhijas un var saistīt relācijas starp atribūtiem.
- Atribūts – atbilst vienai kolonnai tabulā vai izteiksmē. Izteiksme parasti ir kolonnu kopu, vai citu atribūtu kombinācija.
- Savienojums – analogisks tabulu savienojumam SQL vaicājumā. Savienojumam ir tips un kārdinalitāte. Savienojumus var izmantot dimensijās lai savienotu dimensiju tabulas kopā, vai kuba modelī lai savienotu kuba modeļa dimensijas ar to faktu objektu, vai faktu objekta iekšienē savienotu vairākas faktu tabulas.
- Hierarhija – definē attiecības starp vienu vai vairāku atribūtu kopu, kas ir sagrupēti pa līmeņiem dimensiju vai kuba modelī.
- Līmenis – sastāv no atribūtiem, kas ir saistīti kā viens loģisks solis hierarhijas kārtībā. Attiecības starp atribūtiem vienā līmenī parasti definē ar funkcionālu atkarību.
- Kubs – specifiskā kuba modeļa apakškopa vai instance. Kuba fakti un kuba dimensijas ir to apakškopas, kas atsaucās uz kuba modeli. Daudzus kubus var nolasīt ar vienu SQL teikumu.
- Kubu fakti – uz tiem atsaucās faktu objektu mērījumu apakškopas.
- Kubu dimensijas – uz tām atsaucās dimensiju atribūtu apakškopas. Uz tam atsaucās arī atsevišķas kuba hierarhijas.

- Kuba hierarhija – uz to atsaucās hierarhijas atribūtu apakškopas, kur atribūtu kārtībai jāsakrīt ar kārtību hierarhijā.
- Kuba līmenis – līmeņa apakškopa, ko izmanto kubā. Uz kuba līmeni atsaucās līmenis, no kura tas ir atvasināts. Kuba līmenis manto līmeņa atslēgu atribūtus, kas ir atvasināti no augstāka līmeņa.

## 2.4. Izraces funkcijas

Datu analīzes tehnikas ir: tradicionālā statistika, OLAP un datubāzes vaicājumi. Šīs tehnikas parasti asociējas ar datu izraci. Šīs metodes ir hipotētiskās, jo šajā gadījumā labi jāsaprot sistēma lai formulētu precīzus vaicājumus vai noteiktu un testētu precīzu hipotēzi datos.

Datu izraci var definēt kā vērtīgu šablonu un relāciju atklāšanas vai modelēšanas procesu. Datu izraci var saukt par atklāšanu, jo šīs tehnikas var atrast un raksturot nezināmas relācijas. [17, 7.1. sad.]

Datu izrace ir iteratīvs process. Atrodot datus, vizuālizējot modeli un interpretējot biznesa problēmu, var uzzināt par datiem un paslēptiem šabloniem un attiecībām. Šīs ieskatīšanās ļauj labot nepareizas vērtības, pārveidot esošus laukus lai attēlotu to nozīmi skaidrāk, savākt papildus datus. [17, 7.2. sad.]

Vadošās kompānijas gūst lielo peļņu no datu izraces izmantošanas. Datu izrace palīdz tiem pieņemt labākus biznesa lēmumus. Bet šī pieeja ir sarežģīta, parasti tā darbojas atsevišķi no centrālā datu repozitorija un BI struktūras. Analīze un lēmumu pieņemšana ir ļoti svarīgas, tāpēc DWE produktā ir iebūvēta datu izraces tehnika. Šī metodoloģija ir pieejama datubāzes vidē caur portāliem, atskaišu rīkiem un citām lietojumprogrammām. Par datu izraces platformu var kļūst vaicājumu rīks, kas var eksportēt SQL.

Iebūvētai datu izracei ir atklāšanas izraces metodēm: asociācijas, secīgi šabloni un klasteru meklēšana. Šīs metodes ir salīdzinoši viegli izpildīt, izskaidrot un lietot tos biznesa lietojumprogrammās. Prognozējošas datu izraces metodes ir klasificēšana un vērtību prognoze. Tās metodes arī var realizēt ar iebūvētu izraci, bet šīs metodes ir grūtāk izpildīt un izskaidrot. Biznesa analītiķi izmanto prognozējošās metodes savos rēķinos. Parasti savos uzdevumos viņi izmanto statistiskus modeļus.

Datu noliktava iebūvētai datu izracei ir atsevišķs, centralizēts un visaptverošs datu avots. Var izvairīties no daudzām problēmām, kas ir raksturīgas datu pārvietošanā un dublēšanā caur centralizētu datu resursu vadību. Ja darbības notiek datubāzē, tad ir augsta caurlaidība un plašas datu noliktavas platformas, ar kurām veic BI analīzi.

BI analīzi var tikt efektīvi atbalstīt ar definīcijas apgabala vai projekta datu struktūrām. Piemēram, lai atbalstītu notiekošo klienta segmentācijas projektu, jābūt izveidotai datu vitrīnai. Tas palīdz regulāri veikt datu pārveidojumus – aizpildīt un atjaunot tabulas, kas satur klienta datus vajadzīgā struktūrā analīzei. Datu vitrīnas uztur dažāda veida vajadzīgus datus un atjauno tos. Tapēc datu vitrīnas samazina kopējas analīzes laiku.

DB2 vidē ir izveidoti PMML un SQL standarti, jo ir vajadzīga datu izraces integrēšana. PMML palīdz izteikt datu izraces modeli kā XML objektu. Tādu objektu var pagriezt starp analītiskām vidēm bez modeļa pārbuvēšanas vai pārkodēšanas. PMML arī vienkāršo modeļa atjaunošanas procesu bez pārkodēšanas un pārtestēšanas. Tas ir tāpēc, ka modeļa realizācijas laiks ievērojami samazinās. Analītiķis var ātri pārlicināties vai viņš drīkst izmantot aktuālus modeļus pieņemot lēmumus [17, 7.3. sad.].

DWE ir trīs DB2 datu izraces komponentes: modelēšana, vizuālizēšana un rēķināšana.

Modelēšana [17, 7.4.1. sad.] ir DB2 SQL lietojumprogrammu izstrādes saskarne, kas ir realizēts kā DB2 paplašinātājs. Modelēšana ir pieejama grafiski caur DWE Design Studio lai uzbūvētu datu izraces modeļus no datubāzes informācijas.

Modelēšana piedāvā piecus datu izraces metodes:

- Asociēšanas: Kādas vienības līdzības eksistē datos?
- Secības: Kādi secības šabloni eksistē datos?
- Klasteru meklēšana: Kādas grupas vai segmenti eksistē datos?
- Lēmuma koka klasifikācija: Kā prognozēt kategoriskās vērtības?
- Regresija: Kā prognozēt skaitliskās vērtības?

DWE modelēšanas arhitektūra ir sekojoša. Modelēšana nodrošina SQL glabājamo procedūru kopu un lietotāja definētas funkcijas. Tas ir vajadzīgs, lai uzbūvētu modeli un glabātu to DB2 tabulā. Šīs procedūras un funkcijas diezgan vieglas izraces procedūras. Šādas procedūras automātiski izpilda izraces uzdevumus, kuri nosaka kāda tipa modeli uzbūvēt, kādus parametrus iestādīt, kur atrodas dati un datu iestādīšanas (piemēram, kādu kolonnu izmantot modelī). DB2 tabula satur kolonnas, kas attēlo visu ierakstu atribūtu uzvedību un atbildi vai rezultātu. Šādas tabulas izmanto kā datu avotus modeļa būvēšanai, prognozējošai izracei un validācijai (testēšanai).

Vizuālizēšana [17, 7.4.2. sad.] ir *Java* lietojumprogramma, kas izmanto SQL lai izsaukt un grafiski attēlot PMML modeļus. Vizuālizēšana nodrošina analītiķim pieeju modeļa kvalitātei, un viņš var nolemt kā uzlabot modeli regulējot to saturu vai parametrus un izskaidrojot biznesa rezultātus. Izmantojot vizuālizāciju, analītiķis var izpētīt gatavu modeli.

Ir specialie vizualizātori katrai izraces modelēšanas metodei. Tām ir dažāda veida informācija, kas palīdz novērtēt modeļa kvalitāti un izskaidrot to no biznesa viedokļa.

Vizualizācija var arī attēlot PMML modeļus, kurus ģenerē ar citiem rīkiem. Tas ir tad, kad modelim ir atbilstošie vizuālizācijas paplašinājumi - kvalitātes informācija vai sadalīšanas statistika. Daži no šiem modeļu tiptiem satur maz informācijas un tāpēc nav parāk labi attēloti vizuālizācijā.

Rēķināšana [17, 7.4.3. sad.] ir realizēta kā *DB2* paplašinātājs. Tā pielieto PMML modeļus lielajām datubāzēm, datubāzu apakškopām, vai atsevišķiem ierakstiem. Rēķināšana atbalsta visus modeļu tipus, kas atbalsta PMML modeļus:

- Asociēšanas
- Secības
- Klasteru meklēšana (sadalīta, centrālā)
- Klasifikācija (lēmumu koka, loģiskā regresija)
- Regresija (lineāra, polinomiāla, pārveidošana)

DWE rēķināšanas arhitektūra ir sekojoša. PMML modeļi veido un glabā ar modelēšanas komponenti vai veido ar citu lietojumprogrammu un importē *DB2* ar rēķināšanas SQL importa funkciju. Rēķināšanai var piekļūt caur *Design Studio* un pielietot PMML modeļi jauniem datiem. Rēķināšanas skaitlis tiek piešķirts katram ierakstam, kas ir atkarīgs no izraces modeļa tipa. Piemēram, ar klasteru meklēšanas modeļi, šo skaitli vislabāk ir piekārtot dotajam ierakstam. Rezultāti būs ierakstīti jaunajā *DB2* tabulā vai skatā, kur tam var piekļūt citas lietojumprogrammas - atskaišu veidošanas vai OLAP analīzes programma. Rēķināšanā izmanto *DB2* paralēlo apstrādi.

## 2.5. Iekļautas analītiskas funkcijas

DWE *Alphabox* [17, 8.1. sad.] nodrošina iespēju strauji veidot Tīmekļa-bāzētas analītiskas lietojumprogrammas. Šīs lietojumprogrammas var būt piemērotas korporatīvā infrastruktūrā. Tām ir iespēja sasniegt lietotājus gan iekš, gan ārpus korporācijas. Lietojumprogrammas, kas ir uzbūvētas ar *DB2 Alphabox*, izpildās standartajos tīmekļa pārlukprogrammās. Tā nodrošinot reāla laika izpildi, daudzdimensiju analīzi tīmekļa pārlukprogrammā.

Iekš DWE V9.1 *DB2 Alphabox* veido saskarni, caur kuru business skatās uz savas korporācijas datiem. To izmanto lai izstrādātu un atbalstītu informācijas ielādi un nodrošinātu daudzdimensiju datu analīzi.

*DB2 Alphabox* ir izstrādāts kā arhitektūras palīg līdzeklis, kas piegādā analītiskus risinājumus. Lai atvieglotu instalāciju, *DB2 Alphabox* nodrošina grafisko saskarni lai varētu instalēt *DB2 Alphabox* serveri un administrācijas lapas. Ir pieejama analītisko risinājumu

centralizēta administrācija caur administrācijas lapu kopu. Šīs lapas ļauj lietojumprogrammu izstrādātājiem pārvaldīt *DB2 Alphabox* pakalpojumus, kas izmanto tos pašus resursus.

Piemēram, lietojumprogrammu izstrādātāji izmanto *DB2 Alphabox* administrācijas lapas kā ērtu veidu jaunu lietojumprogrammu reģistrēšanai un uzstādīšanai.

*DB2 Alphabox* administrācijas lapas var tikt izmantotas lai konfigurētu *specifiskās DB2 Alphabox* iestādīšanas: datu avotus, relāciju kubus, grupas, un *DB2 Alphabox* servera iestādīšanas. Šo lietojumprogrammu var administrēt caur tīmekļa lapām zem *DB2 Alphabox* mājas lapas vai caur standartu komandu konsoli. Administratori var veidot lietotājus, datu avotus un citus *DB2 Alphabox* objektus, izmantojot kādu no šiem metodēm. Tas nodrošina attālināto servera administrēšanu.

*DB2 Alphabox* atbalsta lietotāju komunikāciju. Izstrādātājs vai gala lietotājs drīkst izvēlēties analītisko lietojumprogrammu, kuru nodrošina *Alphabox*. To var izvērst pēc savām vajadzībām. Tas ļauj visiem lietotājiem atrast analītiskus risinājumus, neatkarīgi no tīkla limita. Tas arī ļauj optimizēt lietojumprogrammas atkarībā no lietotāja prasībām. [17, 7.4.3. sad.]

### 3. DB2 DWE IZMANTOŠANA

Lai pamēģinātu un izpētītu *DB2* īpašības, var apskatīt *DB2 demo* versiju. Šī programma ir domāta *DB2 9* funkcionalitātes demonstrācijai [18]. Lai izmantotu šo programmu, ir vajadzīgas *Windows* operētājsistēma (2000/XP/2003) un pārlūkprogramma (*Internet Explorer*, *Firefox*, vai cita piemērota pārlūkprogramma) un pieeja *DB2* sistēmai ar parauga datubāzi. Parauga datubāze *SAMPLE* var tikt uzinstalēta uz jebkura servera, ko atbalsta *DB2*.

Programma *DB2DEMO* ļauj izpētīt vairākas īpašības. Šīs īpašības ir:

- SQL īpašības
- Objektu-relāciju īpašības
- BI
- Pārvaldība
- Izpilde, u.c.

Apskatīsim sīkāk *DB2DEMO* programmas īpašības, kas attiecas tieši uz BI. Lai izveidotu datu noliktavu, ir nepieciešams izpētīt, kādas BI funkcijas piedāvā *DB2 demo* programma. Tā satur īpašības, kas var palīdzēt analizēt lielu datu apjomu. Šajā programmā ir piemēri, uz kuriem tos var izmēģināt, bet pirms tam, ir jāaizpilda tabulas parauga datubāzē. Vēl ir jāpievērš uzmanība tam, ka tabulām, kas ir izmantotas piemēros, pieprasa vai nu standartu transakcijas tabulas veidošanu vai nu privāto transakciju tabulu. Pirmā gadījumā tiks noģenerētas standartu transakciju tabulas, kas satur 5 veikalus, kas pārdod precis ar 10000 rindām. Otrā gadījumā izmanto tad, ja vajag pamainīt veikala vai produkta vārdu vai tad, ja ir jāpamaina ģenerētu ierakstu skaitu. Tabulas jāģenerē tikai vienu reizi un tad tās var atkal izmantot. BI funkciju grupas ir: apkopojumu tabulas, kubu/augšupejošās funkcijas, ranga funkcijas, statistikas, neatbilstības un daudzdimensiju klasteru meklēšana. Vairākās funkcijās ir jāizmanto transakciju tabula – *TRANSACTIONS*, ar sekojošu struktūru:

Tabula *TRANSACTIONS*

store	char	10	Veikals
quarter	char	2	Kvartāls
item	char	10	Preces nosaukums
sales	int		Pārdošanu skaits

Tabulas TRANSACTIONS satura fragments ir šāds:

STORE	QUARTER	ITEM	SALES
Sydney	Q1	Columbian	1
Sydney	Q2	Mocha	585
Sydney	Q1	Columbian	822
New York	Q2	Columbian	710
New York	Q1	Mocha	91
New York	Q1	Mocha	988
Sydney	Q3	Mocha	8
Berlin	Q1	Columbian	601
Berlin	Q1	Mocha	450
Rockwood	Q3	Kona	783
Sydney	Q4	Columbian	875
Rockwood	Q2	Java	539
Sydney	Q4	Java	862
Rockwood	Q3	Kona	996
New York	Q4	Mocha	266
New York	Q3	Java	375
Rockwood	Q2	Kona	8
Berlin	Q4	Columbian	587
Sydney	Q3	Java	484
New York	Q3	Kona	457
Berlin	Q3	Mocha	599
Rockwood	Q1	Mocha	572
Sydney	Q3	Kona	425

### 3.1. Apkopoju tabulas

Apkopoju tabulas var izmantot lai uzlabotu vaicājumu izpildi. Var veidot tabulu, kurā var ierakstīt atvasinātu rezultātu un atjaunot šo rezultātu, ja ir nepieciešams. Piemēram, tādā tabulā var glabāt maksimālo darbinieka algu kādā kompānijā un glabāt šo informāciju tā lai tas vienmēr būtu aktuāla. Tad, kad lietotājam vajadzēs apskatīt vislielāko darbinieka algu, būs atgriezts rezultāts no apkopoju tabulas, kas ir savā kārtā pārrēķināts no oriģinālas darbinieku tabulas.

Apkopoju tabulas var tikt veidotas lai paturētu vienkāršo vaicājumu rezultātus vai savienojumu kolekciju no vairākām tabulām.

Sekojošais SQL teikums veido apkopoju tabulu, kas pamatojas uz pārdošanu summu un pārdoto vienību skaitu no transakciju tabulas. Šāda tabula var tikt veidota no DDL vai struktūras perspektīvas:

```

drop table storesum;
set current refresh age 0;
create summary table storesum as (
    select store, quarter, item, sum(sales) as total, count(*) as qty
    from transactions
    group by store, quarter, item)
data initially deferred refresh deferred not logged initially
in userspace1 ;

```

Lai aizpildītu tabulu ar aktuālām datiem, jāizmanto komanda REFRESH: refresh table storesum;

Apkopojumu tabula sākumā tiek veidota no oriģinālas tabulas, tomēr tajā nav nekādu ierakstu, veidojas tikai struktūra. REFRESH komanda nosaka, ka apkopojumu tabula jāaizpilda no jauna. Šī komanda izmet visus ierakstus no apkopojumu tabulas un tad aizpilda to ar aktuāliem datiem, t.i. ar tiem, kas atrodas oriģinālā tabulā. Šī komanda nav automātiska un tai jābūt izpildītai regulāros intervālos lai atjaunotu ierakstus pēc vajadzības. Kad apkopojumu tabula eksistē sistēmā, tas no tās var atlasīt datus un tai var veidot indeksus tieši tāpat kā jebkuram citam tabulas objektam.

Var arī apskatīt vaicājumu izpildes grafus ar apkopojumu tabulu un bez tās salīdzinot kas ir efektīvāks. Šim mērķim var apskatīt arī tekstuālo skaidrojumu. Ja salīdzina skaidrojumus, tad iznāk, ka ar apkopojumu tabulu strādāt ir ātrāk nekā ar oriģinālo tabulu.

Iespējama arī apkopojumu tabulu alternatīva izmantošana. No apkopojumu tabulas arī var veidot atvasinājumu. Tajā vietā lai rakstītu

```

select store, quarter, item, avg(sales) as "Average" from transactions
group by store, quarter, item

```

var uzrakstīt

```

select store, quarter, item, total/qty as "Average" from storesum
group by store, quarter, item

```

, kas strādās ātrāk.

Darbībās ar apkopojumu tabulām vēl ir tāda iespēja, kā kuba izpildes parādīšana. Kubi un agregācijas arī var tikt izrēķināti priekš izveidotām apkopojumu tabulām. To var panākt ar komandu CUBE(„GROUP BY saraksts”).

Šajā demo programmā ir arī tāda opcija, kā apkopojumu tabulu atjaunošana, pievienojot ierakstus bāzē. Tas izskatās pēc automātiskām apkopojumu tabulām. Ideja ir sekojoša: dzēst apkopojumu tabulu Storesum un veidot to no jauna ar REFRESH IMMEDIATE opciju. Tas ļauj DB2 izmantot iekšējo loģiku lai modificētu apkopojumu tabulu, kad no oriģinālas tabulas dati tiek dzēsti vai ierakstīti. Tādas tabulas tiek sauktas par dinamiskām apkopojumu tabulām.

Lietotāju uzturētu apkopojumu tabulu opcija ļauj lietotājiem veidot savu apkopotu informāciju un saņemt DB2, lai izmantotu to kā daļu no sava optimizācijas procesa. Lai strādātu ar šādām tabulām, ir jāizslēdz integritātes īpašības. Ir arī jāizslēdz visas apkopojumu tabulas, jo aktuāls select teikums var paņemt datus no citām apkopojumu tabulām, kas nav vajadzīgas. Tomēr optimizācijai jābūt ieslēgtai lai apkopojumu tabula tiktu atpazīta. Tādās tabulās var mainīt statistiku lai parādītu nekorektu informāciju. Ir iespējams izskatīt no jauna vaicājumu lai apskatītu kas notiek, kad ieslēdz optimizācijas īpašību. Tāda vaicājuma rezultātiem jābūt atšķirīgiem no tiem, kurus saņem pirms optimizācijas ieslēgšanas. Tas izceļ faktu, ka ja padod nekorektu informāciju apkopojumu tabulai, DB2 to izmantos optimizācijas laikā.

### 3.2 Lineāras regresiju funkcijas

DB2 ir funkcijas, kas var tikt izmantotas lai analizētu tabulas datus. Tādas funkcijas ļauj uzstādīt grafikus datu kopai lai ģenerētu punktus un zīmētu grafikus caur to (pamatojoties uz tabulas datiem). Lai varētu izpildīt demo piemērus, ir jāģenerē koordinātu tabula XYCOORDS. Priekš tam ir jāizvēlas viena no tabulu veidošanas komandām:

➤ Lineārie dati

Šī komanda ģenerē x un y koordinātes, pamatojoties uz funkciju  $y=2x+5$ . Dotajā gadījumā SQL izmanto vienkāršus tabulu izteikumus un rekursiju lai ģenerētu datus:

```
create table xycoords(  
  x int,  
  y int)  
in userspace1;  
insert into xycoords
```

```

with temp1(x) as
(
  values (0)
  union all
  select x+1 from temp1 where x < 10
)
select x, 2*x + 5
  from temp1;

```

Tabula XYCOORDS aizpildās šādi:

X	Y
0	5
1	7
2	9
3	11
4	13
5	15
6	17
7	19
8	21
9	23
10	25

➤ Nejausie (gadījuma) dati

Šī datu kopa ir pilnīgi nejauša, x un y pieņem vērtības no 0 līdz 10. Katru reizi izpildot komandu, tā izdod dažādus rezultātus:

```

create table xycoords(
  x int,
  y int)
  in userspace1;
insert into xycoords
  with temp1(x) as
  (
    values (0)
    union all
    select x+1 from temp1 where x < 10
  )
  select x, int(rand()*10)
  from temp1;

```

Izpildot šo komandu, tabula XYCOORDS aizpildīsies šādi, tomēr katru reizi aizpildīsies dažādi, jo Y kolonna pieņem gadījuma vērtības:

X	Y
0	7
1	4
2	3
3	7
4	5
5	3
6	9
7	6
8	1
9	3
10	1

Vienu reizi uzģenerējot datu kopu, ir jāizpilda komandas, kas parāda kā dati atbilst lineārā vienādojumā  $y=ax+b$ . Var pārslēgties starp nejaušiem lineāriem datiem, pārveidojot tabulu ar vienu no šīm divām komandām.

Var apskatīt arī ģenerēto vienādojumu. Nākamā komanda ģenerē „vienādojumu” no funkcijas parametriem, kas tika noģenerēti no iepriekšējās funkcijas. Parasti kad saņem rezultātu, tad ievieto to hosta mainīgos. SQL komanda veido simbolu rindu, kas veido vienādojumu  $ax+b$ , apskatot koeficientus un ievietojot tos a un b vērtībās:

```
select DEC(REGR_SLOPE(y,x),5,2) AS SLOPE,DEC(REGR_ICPT(y,x),5,2) AS
INTERCEPT
from xycoords;
```

Ar lineāriem datiem rezultāts būs šāds:

SLOPE	INTERCEPT
2,00	5,00

Ar gadījuma datiem rezultāts būs šāds:

SLOPE	INTERCEPT
-0,34	6,18

Šāda vienādojuma a un b koeficienti arī ir izrēķināmi:

```
select DEC(REGR_SLOPE(y,x),15,2) as "Slope (a)" from xycoords;
select DEC(REGR_INTERCEPT(y,x),15,2) as "Intercept (b)" from xycoords;
```

Demo programma ļauj arī atrast vidējas  $x$  un  $y$  vērtības lai salīdzinātu ar kādiem statistiskiem datiem, datu pāru skaitu, kas tika izmantoti funkcijas aprēķināšanā.  $X$  un  $y$  variānces summu arī risina lai izmantotu to tālākajā statistikā.

Izņemot lineāras funkcijas vēl ir statistiskas funkcijas. Statiskās funkcijas var tikt izmantotas lai analizētu savus datus. Šīs funkcijas var izmantot kombinācijā ar regresijas funkcijām lai salīdzinātu datus ar lineāras funkcijas grafiku. Lai izmēģinātu demo versijas piemērus tiek piedāvāts izveidot tabulas, gan lineāriem datiem, gan nejaušiem datiem, analogiski tām, kas ir domātas lineāru regresiju funkcijām. Statistiskās funkcijas tāpat kā lineāras ļauj izrēķināt variānci, kovariānci un korelāciju. Vēl ir funkcija, kas atgriež skaitļu kopas standartu novirzi.

### **3.3. Kubu un agregāciju operatori**

Kubu un agregāciju operatori dod lietotājiem jaunas SQL īpašības lai ļautu veikt daudzdimensiju analīzi. Parasti ir vajadzīgi OLAP rīki lai veiktu tādus rēķinus, bet šī programma veic to pati.

*Demo* versija piedāvā apskatīt regulāru summu piemērus. Piemēram, ja lietotājs veic pārdošanu analīzi, tad viņam būtu jāizmanto dažāda veida summas. Izpildot nākamo vaicājumu, var ieraudzīt informāciju par 4 šķiru kafijas pārdošanu 4 veikalos 4 gada kvartālos:

```
select store, quarter, item, sum(sales) as summa from transactions
group by store, quarter, item;
```

Rezultātā būs 64 ieraksti:

STORE	QUARTER	ITEM	4
Berlin	Q1	Columbian	71179
Berlin	Q1	Java	60815
Berlin	Q1	Kona	69520
Berlin	Q1	Mocha	68489
Berlin	Q2	Columbian	61769
Berlin	Q2	Java	55479
Berlin	Q2	Kona	46225
Berlin	Q2	Mocha	71368
Berlin	Q3	Columbian	59609
Berlin	Q3	Java	48829
Berlin	Q3	Kona	57831
Berlin	Q3	Mocha	60548
Berlin	Q4	Columbian	65180
Berlin	Q4	Java	51745
Berlin	Q4	Kona	69435
Berlin	Q4	Mocha	63364
New York	Q1	Columbian	58636
New York	Q1	Java	65147
New York	Q1	Kona	61963
New York	Q1	Mocha	53817
New York	Q2	Columbian	60014
New York	Q2	Java	74101
New York	Q2	Kona	59817
New York	Q2	Mocha	63134
New York	Q3	Columbian	67174
New York	Q3	Java	60835
New York	Q3	Kona	56888
New York	Q3	Mocha	57199
New York	Q4	Columbian	62913
New York	Q4	Java	65894
New York	Q4	Kona	57123
New York	Q4	Mocha	62856

Rockwood	Q1	Columbian	71422
Rockwood	Q1	Java	56862
Rockwood	Q1	Kona	72475
Rockwood	Q1	Mocha	57269
Rockwood	Q2	Columbian	58735
Rockwood	Q2	Java	60554
Rockwood	Q2	Kona	71278
Rockwood	Q2	Mocha	60769
Rockwood	Q3	Columbian	67026
Rockwood	Q3	Java	48545
Rockwood	Q3	Kona	69120
Rockwood	Q3	Mocha	54606
Rockwood	Q4	Columbian	67738
Rockwood	Q4	Java	57547
Rockwood	Q4	Kona	64194
Rockwood	Q4	Mocha	69713
Sydney	Q1	Columbian	120425
Sydney	Q1	Java	142498
Sydney	Q1	Kona	109526
Sydney	Q1	Mocha	131997
Sydney	Q2	Columbian	135071
Sydney	Q2	Java	129032
Sydney	Q2	Kona	122706
Sydney	Q2	Mocha	128161
Sydney	Q3	Columbian	137624
Sydney	Q3	Java	131879
Sydney	Q3	Kona	131003
Sydney	Q3	Mocha	129501
Sydney	Q4	Columbian	143624
Sydney	Q4	Java	121926
Sydney	Q4	Kona	117570
Sydney	Q4	Mocha	131452

Tomēr summēšana no dažādiem gabaliem prasa daudz laika. Izmantojot ROLLUP operatoru, var veidot summēšanu no zemāka līmeņa, t.i. pārdošanu skaita līdz augstākajam līmenim, t.i. veikala vienā SELECT teikumā. To var panākt, izpildot šādu vaicājumu:

```
select store, quarter, item, sum(sales) from transactions
group by rollup(store, quarter, item);
```

Šī vaicājuma rezultātā būs tas pats, kas ir pagājušā vaicājumā un tam nāk klāt papildus ieraksti. Pirmais ieraksts attēlo summu pa visām pārdošanām, nākamās 4 rindās attēlo summas pa visām pārdošanām katrā veikalā. Pēc tam ir 16 rindas, kuras attēlo summas pa visām pārdošanām katrā veikalā un katrā kvartālā. Tas izskatīsies šādi:

STORE	QUARTER	ITEM	4
5040744			
Berlin			981385
New York			987511
Rockwood			1007853
Sydney			2063995
Berlin	Q1		270003
Berlin	Q2		234841
Berlin	Q3		226817
Berlin	Q4		249724
New York	Q1		239563
New York	Q2		257066
New York	Q3		242096
New York	Q4		248786
Rockwood	Q1		258028
Rockwood	Q2		251336
Rockwood	Q3		239297
Rockwood	Q4		259192
Sydney	Q1		504446
Sydney	Q2		514970
Sydney	Q3		530007
Sydney	Q4		514572

Summēšana ar kuba operatoru notiek sekojoši: CUBE operators veic summēšanu tālāk par vienu līmeni summējot pāri citām dimensijām arī, ne tikai no zemāka līmeņa uz augstāko. Šajā gadījumā tiek saņemta summēšanas informācija pēc šķiras, veikala un kvartāla dažādās dimensijās. Vaicājums izskatās šādi:

```
select store, quarter, item, sum(sales) from transactions
group by cube(store, quarter, item);
```

Šī vaicājuma rezultātā būs tas pats, kas ir vaicājumā ar ROLLUP operatoru un tām nāk klāt papildus ieraksti. Tie attēlo katras kafijas šķiras pārdošanas summas katrā kvartālā un summas un kopējās katra kvartāla pārdošanas summas. Arī ir attēlotas kopējas summas pēc katras kafijas šķiras.

Tas izskatīsies šādi:

STORE	QUARTER	ITEM	4
Q1	Columbiar	321662	
Q1	Java	325322	
Q1	Kona	313484	
Q1	Mocha	311572	
Q1		1272040	
Q2	Columbiar	315589	
Q2	Java	319166	
Q2	Kona	300026	
Q2	Mocha	323432	
Q2		1258213	
Q3	Columbiar	331433	
Q3	Java	290088	
Q3	Kona	314842	
Q3	Mocha	301854	
Q3		1238217	
Q4	Columbiar	339455	
Q4	Java	297112	
Q4	Kona	308322	
Q4	Mocha	327385	
Q4		1272274	
Columbian	1308139		
Java	1231688		
Kona	1236674		
Mocha	1264243		

Lai parādītu rezultātiem dažāda līmeņa grupēšanu, izmanto grupēšanu pēc grupēšanas lauku kopām. Vaicājumam ir grupēšanas kopas veikals un kvartāls, veikals un šķira, šķira. Viena no grupēšanas kopām ir tukšā (), un tas nozīmē, ka rezultātam pievienojas viens ieraksts, kur parādīta tikai pārdošanas summa pa visiem datiem. Vaicājums izskatās sekojoši:

```
select store, quarter, item, sum(sales) from transactions
group by grouping sets
((store,quarter),
 (store,item),
 (item), ());
```

Rezultātā parādīsies visu pārdošanu kopsumma, kopsummas pa katrai kafijas šķirai, kopsummas pa visiem veikaliem un šķirām, kopsummas pa veikaliem un kvartāliem. Tas izskatīsies šādi:

STORE	QUARTER	ITEM	4
5040744			
Columbian	1308139		
Java	1231688		
Kona	1236674		
Mocha	1264243		
Berlin		Columbian	257737
New York		Columbian	248737
Rockwood		Columbian	264921
Sydney		Columbian	536744
Berlin		Java	216868
New York		Java	265977
Rockwood		Java	223508
Sydney		Java	525335
Berlin		Kona	243011
New York		Kona	235791
Rockwood		Kona	277067
Sydney		Kona	480805
Berlin		Mocha	263769
New York		Mocha	237006
Rockwood		Mocha	242357
Sydney		Mocha	521111
Berlin	Q1		270003
Berlin	Q2		234841
Berlin	Q3		226817
Berlin	Q4		249724
New York	Q1		239563
New York	Q2		257066
New York	Q3		242096
New York	Q4		248786
Rockwood	Q1		258028
Rockwood	Q2		251336
Rockwood	Q3		239297
Rockwood	Q4		259192
Sydney	Q1		504446
Sydney	Q2		514970
Sydney	Q3		530007
Sydney	Q4		514572

### 3.4. Ranga funkcijas

OLAP funkcijās ir iekļauta iespēja atgriezt rangu un rindu numurēšanu vaicājumā. Tajās funkcijas var būt starp izvēlētām kolonnām SELECT-sarakstā vai ORDER BY daļā.

Ranga funkcijas rēķina rindu kārtību pēc kāda sadalīšanas kritērija. Rindām, kuras pēc vajadzīgas pazīmes neatšķiras, piešķir vienādu rangu. Rezultātus var sakārtot ar tukšumiem un bez tiem. Šī situācija rodas tad, kad divām vai vairākām rindām ir vienādi rangi.

Ir divas ranga komandas: RANK un DENSERANK. Ja izmanto komandu RANK, tad gadījumā kad diviem ierakstiem ir vienāda pazīme, tad rangu secībā būs viens vai vairāki tukšumi. Ja izmanto komandu DENSERANK, tad šādā gadījumā tukšumu nebūs. Abas šīs funkcijas sāk numerāciju ar skaitli 1.

Rangu funkciju pētīšanai apskatīsim tabulu EMPLOYEE, kuru kolonnas ir LASTNAME, SALARY. Mērķis ir apreķināt rangu katram darbiniekam pēc lieluma. Tabulas EMPLOYEE saturs ir šāds:

LASTNAME	SALARY
Brown	1000
Daniels	500
Jones	1000
Richards	3000
Stevens	2000

Apskatīsim vaicājumu ar komandu RANK:

```
select lastname, salary, rank() over (order by salary desc) as rank_salary  
from employee  
order by lastname;
```

Izmantojot funkciju RANK, rezultāts būs šāds:

LASTNAME	SALARY	RANK_SALARY
Brown	1000	3
Daniels	500	5
Jones	1000	3
Richards	3000	1
Stevens	2000	2

Rangu secībā ir tukšums, izlaists rangs 4, jo ir divi ieraksti ar rangu 3.

Ja tajā pašā vaicājumā RANK vietā izmantot funkciju DENSERANK, tad rezultāts būs šāds:

LASTNAME	SALARY	RANK_SALARY
Brown	1000	3
Daniels	500	4
Jones	1000	3
Richards	3000	1
Stevens	2000	2

Šī funkcija neatstāj tukšumus, neatkarīgi no tā, cik ir ierakstu ar vienādu rangu.

Šajā funkciju sadaļā ir arī funkcija ROWNUMBER. Šī funkcija rēķina rindas secības numuru, sākot no 1. Ja SELECT teikumā nav ORDER BY daļas, tad ierakstus atgriež tādā secībā, kuru nosaka SELECT-sarakstā.

Piemērs tādām vaicājumam ir:

```
select rownumber() over (order by salary) as number,  
       lastname, salary from employee  
order by lastname;
```

Rezultāts būs šāds:

NUMBER	LASTNAME	SALARY
3	Brown	1000
1	Daniels	500
2	Jones	1000
5	Richards	3000
4	Stevens	2000

Šis vaicājums atgriež piekārtotu skaitli, uzvārdu un algu. Rezultāts ir sakārtots pēc uzvārda. Ja no šajā vaicājuma izmestu „order by lastname” izteiksmi, tad rezultātā būs saraksts, kas ir sakārtots pēc algas lieluma. Atšķirībā no RANK un DENSERANK funkcijām, rezultātā nav rindu ar vienādu numuru.

### 3.5. Daudzdimensiju klasteru meklēšana

Daudzdimensiju klasteru meklēšana (MDC) ir paredzēta datu noliktavām un lielām datubāzēm. Tā palīdz nodrošināt daudzdimensiju datu ilgstošu automātisku klasteru meklēšanu. Klasteru meklēšana palīdz uzlabot vaicājumu izpildi, samazina datu uzturēšanas (piemēram, reorganizācijas vai indeksu uzturēšanas) operāciju skaitu INSERT, UPDATE un DELETE laikā.

MDC nodrošina lai tabulai būtu sameklēti klasteri vairākās dimensijās reizē. MDC tabula organizē rindas uz diskiem secīgu lappušu blokos, piemēram, tā lai rindām blokā būtu vienādas dimensiju vērtības. Visiem blokiem ir vienāds lappušu skaits. MDC tabulas dimensijas nosaka tad, kad veido tabulu. Katrai dimensijai automātiski izveidojas bloka indekss. Dimensiju kopai veidojas salikts bloku indekss. Dimensija var būt kolonna vai kolonnu secība.

MDC tabulām pielieto komandu RUNSTATS, kas ir vajadzīga lai padotu optimizētājam MDC struktūras papildus informāciju.

Lai pamēģinātu MDC funkcijas, ir jāizveido tabula tempMDC:

```
create table tempMDC
( empno int,
  dept int,
  div int )
in userspace1
#
begin atomic
  declare i int default 0;
  while i < 10000 do
    insert into tempMDC values
      ( i,
        int(rand()*5)+1,
        int(rand()*10)+1 );
    set i = i + 1;
  end while;
end;
```

Šajā tabulā ir ierakstīti 10000 ieraksti, departamenta kolonnas vērtības ir gadījuma skaitļi no 1 līdz 5, un divīzijas kolonnas vērtības ir gadījuma skaitļi no 1 līdz 10.

Lai izpētītu atšķirības, kas ir vaicājumam no MDC tabulas un vaicājumam no tabulas bez MDC, veidosim divas tabulas: empMDC un empNoMDC, kuru kolonnas attēlos darbinieka numuru, departamentu un divīziju.

```

create table empMDC
( empno int,
  dept int,
  div int )
in userspace1
organize by dimensions(dept,div);

```

Šis skripts veido tabulu empMDC, kurai ir divas dimensijas – departaments un divīzija. Tajā ierakstīsim visus datus no tabulas tempMDC:

```

insert into empMDC
select * from tempMDC;
call admin_cmd('RUNSTATS ON TABLE %[schema].EMPMDC AND INDEXES
ALL');

```

Tabulas empNoMDC struktūra ir tieši tāda kā empMDC. Tā arī tiek aizpildīta ar tabulas tempDMC vērtībām. Bet tai nav pielietota komanda RUNSTATS.

Kad tabulas empMDC un empNoMDC ir gatavas un aizpildītas ar 10000 katra, var apskatīt kā notika klasteru meklēšana. Attēlosim to tabulu fragmentus, lai apskatītu, kas mainās veicot klasteru meklēšanu. Tabulas empNoMDC dati ir šādi:

EMPNO	DEPT	DIV
0	5	5
1	2	7
2	5	7
3	4	6
4	1	5
5	3	10
6	2	7
7	1	5
8	5	5
9	4	10
10	5	7
11	1	8
12	1	2
13	4	10
14	1	2
15	3	2
16	4	4
17	4	9
18	4	6
19	4	2
20	3	9
21	5	10
22	3	2
23	2	6

Tabulā empMDC dati izskatās citādi:

EMPNO	DEPT	DIV
0	5	5
8	5	5
32	5	5
52	5	5
101	5	5
383	5	5
389	5	5
466	5	5
611	5	5
785	5	5
814	5	5
837	5	5
850	5	5
855	5	5
1004	5	5
1024	5	5
1035	5	5
1037	5	5
1048	5	5
1063	5	5
1100	5	5
1162	5	5
1166	5	5
1204	5	5

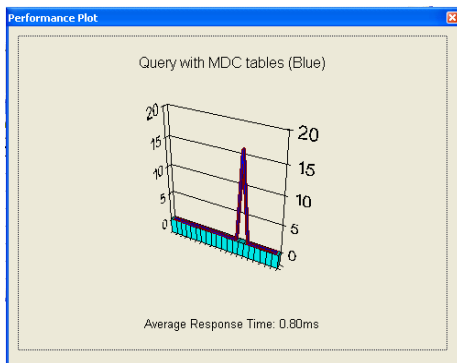
Šeit var pamanīt, ka dati ir sakārtoti. Tas tika veidots pēc sekojoša principa. Sakumā ir visas rindas, sākot no 0-ta darbinieka numura, kuru departaments un divīzija sakrīt ar 0-ta darbinieka departamentu un divīziju. Pēc tam iet ieraksts ar vismazāko darbinieka numuru no tiem, kurš vēl nav parādījušas tabulā un tām seko visi ieraksti ar tādu departamenta un divīzijas vērtību pāri. Tā turpinās ar katru departamenta un divīzijas vērtību pāri. Var iedomāties, ka katrs departamenta un divīzijas vērtību pāris veido datu apakškopu. Katrā apakškopā ieraksti ir sakārtoti pēc darbinieku numuriem pieaugšanas secībā.

Var uzzīmēt vaicājumu izpildes grafus un salīdzināt viena un tā paša vaicājuma izpildi, izmantojot katru no šīm divām tabulām.

Ar nākama skripta palīdzību tiks zīmēts izpildes grafs vaicājumam ar tabulas empMDC izmantošanu:

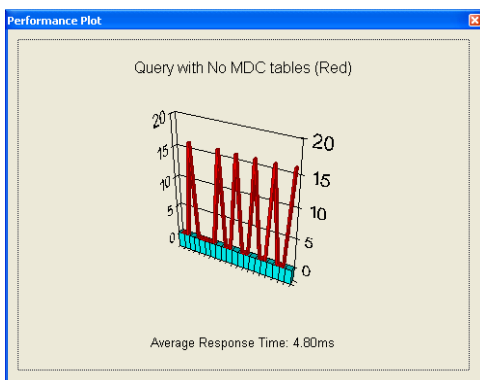
```
%graph  
color=blue  
repeat=20  
title="Query with MDC tables (Blue)"  
sql="select div, dept, count(*) from empMDC"
```

where dept = 5 and div = 2  
group by div, dept" ;



Lai uzzīmētu grafu vaicājumam ar tabulas empNoMDC izmantošanu, uzrakstīsim šādu skriptu:

```
%graph  
color=red  
repeat=20  
title="Query with No MDC tables (Red)"  
sql="select div, dept, count(*) from empNoMDC  
where dept = 5 and div = 2  
group by div, dept" ;
```



Salīdzinot šos zīmējumus, var pamanīt, ka MDC izmantošana paātrina vaicājumu izpildi.

## 4. DATU NOLIKTAVU RISINĀJUMS BANKĀM

Biznesa un finanšu organizācijas galvenokārt ir saistītas ar zināma veida riskiem. Šādu organizāciju mērķis ir iegūt maksimālo peļņu. Lēmumu pieņemšanā lielu lomu spēlē riska smaguma un naudas summu aprēķini. Organizācijas izmanto marketinga galveno ideju – lemj kuriem produktiem jāpaaugstina cena un kam tie jāpārdod. [19, 3.lpp.]

Ir svarīgi sakārtot datu struktūras, kas attiecas uz risku un finanšu datiem. Tie dati ir:

- Transakciju dati
- Naudas vērtību dati
- Klienta dati

Transakciju dati attēlo transakciju tipus un finanšu rezultātus. Naudas vērtību dati parāda cik lielas naudas summas ir saistītas ar transakcijām, cenas un cenu izmaiņas. Klienta dati attēlo klientus, kuri ņem kredītu bankā vai veic naudas piegādi bankā.

Šos datus var apskatīt dažādos veidos. Bet ne visus no tiem ir viegli attēlot riska un finanšu sistēmās. Šādu sfēru pārvaldībai ir vajadzīgi sakarīgi skati. Lai to panāktu, ir vajadzīga integrēta datu vide, kas atbalsta lēmumu pieņemšanu un atskaišu veidošanu. Finanšu organizācijās šādas prasības ir plaši izplatītas.

*IBM Banking Data Warehouse* (BDW) ir risinājums, kas ir izstrādāts šādām uzņēmumu datu integrācijas vidēm lai veidotu datu noliktavu. BDW palīdz veikt biznesa analīzi, peļņas analīzi, risku analīzi un citas operācijas. BDW versija 3.4 ir *Basel II* (bankas darbību regulēšanas kritēriju noteikumi) un IFRS/IAS (grāmatvedības standarts) atbalsts. Daudzas finanšu organizācijas izmanto BDW lai atbalstītu integrēto datu prasības visos biznesa aspektos.

BDW izmanto integrētu datu centru ar vienu apkopotu skatu uz datiem. BDW arhitektūrai ir šādas īpašības [19, 5.lpp.]:

- Integrēta risku un izaicinājumu informācija
- Elastība, kas palīdz saņemt jaunas prasības
- Ātra reakcija uz jaunām prasībām
- Iespēja apstrādāt datus biznesam
- Saskaņa datu izmantošanā

BDW palīdz finanšu organizācijām būvēt datu noliktavas risinājumus saskaņā ar uzstādītajiem mērķiem. Tie var būt gan kādas nodaļas datu vitrīnas, gan korporatīva datu noliktava. DBW ietver datu noliktavas tehnisko struktūru, kas atbilst dažām biznesa atskaišu un analīzes vajadzībām [19, 8.lpp.].

Šādas vajadzības ir:

- Izdevīgums
- Marketinga attiecības
- Regulējoša atbilstība
- Risks
- Aktīvu un pasīvu pārvaldība

BDW atbalsta *Basel II* prasības datiem. Šim nolūkam to realizē daudzas finanšu organizācijas. Ar BDW tās veic visādas atskaites mazumtirdzniecībai un vairumtirdzniecībai. BDW ir daži modeļi. Tās ir finanšu organizācijas datu noliktavas izstrādes komponentes. Šīs komponentes ir:

- Banku datu noliktavas modelis
- Biznesa risinājumu šabloni
- Lietojumprogrammu risinājumu šabloni
- BDW projektu skati.

#### **4.1. BDW modelis**

BDW modelis [19, 9.lpp.] ir entītijū-relāciju datu modelis. Tajā modelī ir vēsturiskie un atomārie datu noliktavas dati. Tā ir BI struktūra, kas atbalsta vairākus biznesa virzienus un analītiskās funkcijas. Parasti to pielieto vidējām un lielajām finanšu organizācijām. Šīs struktūras mērķis ir nodrošināt datu integrēšanas centru. Tāds centrs samazina izstrādes izmaksas tāpēc, ka nodrošina biznesa inteliģences funkcionalitāti. Tas ir tāpēc, ka datus var ievietot integrēšanas centrā vienu reizi un tad var atkal izmantot BI izstrādi un operācijas.

IBM nodrošina noklusēto fizisko datubāzes projektējumu, kas ir ģenerēts no loģiska entītijū-relāciju datu modeļa. BDW modelis ir datu struktūra, kas ir vajadzīga finanšu organizācijām lai atbalstītu kredītu risku un operāciju risku pieejas.

#### **4.2. Biznesa risinājumu šabloni.**

BDW satur biznesa risinājumu šablonus [19, 10.lpp.] (BST). Tie atspoguļo vispārējos vaicājumu tipus specifiskajās biznesa jomās, ko pielieto daudz lietotāju. BDW atbalsta arī citas prasības – atskaišu veidošanu, datu izraci un lēmumu pieņemšanu.

BST ir trīs galvenās komponentes:

- Atkal izmantotie finanšu mērījumi
- Atkal izmantotās finanšu dimensijas
- Grupēšana virs mērījumiem un dimensijām konfigurējamās BST.

Mērījumi ir, piemēram, galvenie izpildes rādītāji. Tādi rādītāji, piemēram, ir: klientu skaits, pārvaldīto naudas līdzekļu kopsumma un transakciju skaits. Katrs mērījums ir pilnībā definēts un to var izmantot gan atsevišķi, gan kā komponenti, kas darbojas kopā ar formulu. Savukārt tā formula var sadarboties ar lielāku formulu. Mērījumu, kuru izmanto formulā, nodrošina kontekstjūtīgām rēķināšanas īpašībām. Tas nozīmē, ka vienā formulā mērījumu var pieskaitīt kopsummai, bet cita formulā tam jābūt atņemtam no kopsummas. Šī mērījumu atkalizmantošana garantē biznesa mērījumu atbilstību organizācijai. Tas ir palīgīdzeklis BI pārvaldībā. Mērījumus ir vislabāk izmantot tad, kad tos salīdzina zem dažādiem virsrakstiem. Piemēram, depozītu kopsummu salīdzina pa pēdējiem 12 mēnešiem, klientu skaitu salīdzina pa ģeogrāfiskiem reģioniem. BST dimensijas nodrošina tos virsrakstus, pa kuriem klasificēt un salīdzināt mērījumus. Dimensijas var atkal izmantot dažādās BST. To var izmantot analīzei.

### 4.3. Lietojumprogrammu risinājumu šabloni

Lietojumprogrammu risinājumu šabloni [19, 14.lpp.] (AST) parāda, kā *Basel II* rīka komponentu datu saturs pārklāj BDW modeli. BST ir saistīti ar atskaišu funkcijām BDW modelī, bet AST ir saistītas ar citām funkcijām šajā modelī. AST ir katrai riska komponentei. AST ir:

- Kredītu riskiem
- Saistību nepildīšanas varbūtībai
- Saistību nepildīšanas atklāšanai
- Termiņiem
- Gaidītiem zaudējumiem un nodrošināšanām
- Operāciju riskiem.

Katrs AST dod katra riska komponentei pilnu datu prasību klasificēšanu. Tas ir noteikts Basel dokumentācijā. Ir arī katra AST elementa detalizēta atbilstību attēlošana ekvivalentiem BDW modeļa entītijām un atribūtiem. AST palīdz finanšu organizācijām analizēt katras riska komponentes prasības specifiskām vajadzībām. Kad analīze ir pabeigta, finanšu organizācijas meklēs atbilstību starp riska prasībām un BDW modeli. Tad var identificēt datu noliktavas daļu, kas ir vajadzīga lai darbotos ar risku aprēķināšanu.

#### 4.4. BDW Projektu skati

BDW projektu skati [19, 15.lpp.] ir biznesa objektu jomas skatu sērija. Tā aptver visas BDW komponentes. BDW projekta skati ļauj lietotājiem saprast datu kopu, ko pieprasa BST specifiskajām biznesa vajadzībām vai AST specifiskajām datu rēķināšanas prasībām. BDW 3.4 ir projektu skati priekš *Basel II*.

Katrs projektu skats balstās uz izvēlēta BST vai AST. Lai nosūtītu kādus datus vai atskaišu prasības *Basel II*, ir jāizvēlas elementu apakškopa. Katru BDW projekta skata sfēru var paplašināt lai iekļautu definētas atbilstības starp BST AST un BDW modeļiem.

#### 4.5. BDW komponentes un Basel II arhitektūra

BDW komponentes sadarbojās kā papildus satura modeļi, kas risina dažādus pārvaldības informācijas biznesa problēmas un datu arhitektūras problēmas. Atsevišķa modeļa komponentes ievieš arhitektūras struktūrā kā informācijas struktūru. Informācijas struktūras modeļa elementi vienā šūniņā atbilst citiem elementiem citās šūniņās. Tas nozīmē, ka datu vitrīnas pamatmērījums var atbilst BDW modeļa atribūtam, kas savukārt ir datu avots.. Un tam avotam ir jābūt ielādētam datu vitrīnā. Tas attēlo atbilstību starp informāciju, ko pieprasa biznesa lietotājs un tehniskā lietotāja datu glabātuvi. Tā ir atbilstība starp mērījumu un datubāzes atribūtu. Risinot tādas problēmas, finanšu organizācijas pārvalda informāciju un risina BI problēmas. Šīs problēmas [19, 17.lpp.] ir:

- Atrast datu avotus
- Noteikt kā tos pārveidot un sakopot
- Uzlabot datu kvalitātes pārvaldību organizācijā

BDW arhitektūra [19, 18.lpp.] sastāv no sešām daļām, ko atbalsta *Basel II*. Tās ir:

- Datu avoti – iekšējie un ārējie vajadzīgo datu avoti priekš *Basel II*
- Izvilksana – datu izvilksanas process vai tehnoloģija. Datus izvelk no dažādiem avotiem.
- Korporatīva datu noliktava – repozitorijs, kur ir savākti visi detalizētie dati, kas ir vajadzīgi priekš *Basel II*
- Pārveidošana un aprēķini – dažādu aprēķinu pārvaldība
- Datu vitrīnas – sakopotie dati atskaišu veidošanai vai analīzei
- Atskaišu veidošana – *Basel II* atskaišu veidošana un piegāde dažādām lietotāju grupām.

BDW modelis nodrošina korporatīvas datu noliktavas projektējumu. BST nodrošina pamatu *Basel II* atskaitēm un analīzei. AST apraksta prasības datu struktūras kredītu riska komponentēm. BDW projektu skati nodrošina filtrētus skatus.

## 5. BANKU DATU NOLIKTAVAS VEIDOŠANA

Saskaņā ar *Basel II* bankām jākontrolē sekojošās lietas:

- Kredītu riski
- Tirgu riski (procentu, valūtas, fondu, u.c.)
- Nesabalansētas likviditātes riski
- Operāciju riski

Tādēļ datu noliktavā jāattēlo darījumu, maksājumu, valūtu, portfeļu, klientu, kredītu operāciju, operāciju risku informācija.

Ir vēlams lai darījumi būtu saistīti ar maksājumu grafikiem. Tas palīdz prognozēt maksājumu plūsmas, kontrolēt likviditātes riska un aprēķināt procentu risku. Informāciju par valūtu un tās kursiem izmanto lai izsekotu kursu izmaiņas un prognozētu fondu riskus. Lai noskaidrotu kredītu riskus, tad izmanto klientu un kredītu operāciju datus. Aizņēmēja kredībspēja un kredīta produkta risks nosaka kredītu risku. Ar finansu norēķinu palīdzību var novērtēt klientu kredībspēju. Šīm nolūkam var palīdzēt arī aizņēmēju organizācijas finanšu pamatkoeficienti, naudas līdzekļu kustības plāni, kredītu reitingi ar izmaiņu vēsturi, kredītu vēsture.

Kredītam novērtē spreadu. Tas ietekmē produkta cenu. Kredīta spreads ir saistīts ar kredīta produkta riska novērtējumu. Kredīta spreda aprēķināšanai ir vajadzīga sekojošā informācija:

- Kredīta līguma parametrus (darījuma termiņš, resursu cena)
- Kredīta nodrošināšana
- Trešo personu atbalsts (garantijas un rekomendācijas)

Riskus novērtē ar operāciju risku datiem. Priekš tam noliktavā ielādē sekojošu informāciju:

- Banku operāciju klasifikācija pēc darbības virzieniem
- Visi banku operāciju dati katras operācijas bruto ienākuma aprēķināšanai
- Koeficientu bāze rezervēta kapitāla lieluma aprēķināšanai
- u.c. [20]

Visu šo informāciju uzkrāj datu noliktavā vēsturiskā griezumā. To izmanto risku pārvaldībai un atskaišu veidošanai. Datu noliktavā ir ielādēta informācija no vairākiem avotiem. Tie avoti ir:

Automatizēta banku sistēma un ārējie avoti. No automatizēta banku sistēmas integrē darījumu datus ar operāciju (darījumu) grafikiem; grāmatvedības informāciju, sarakstus,

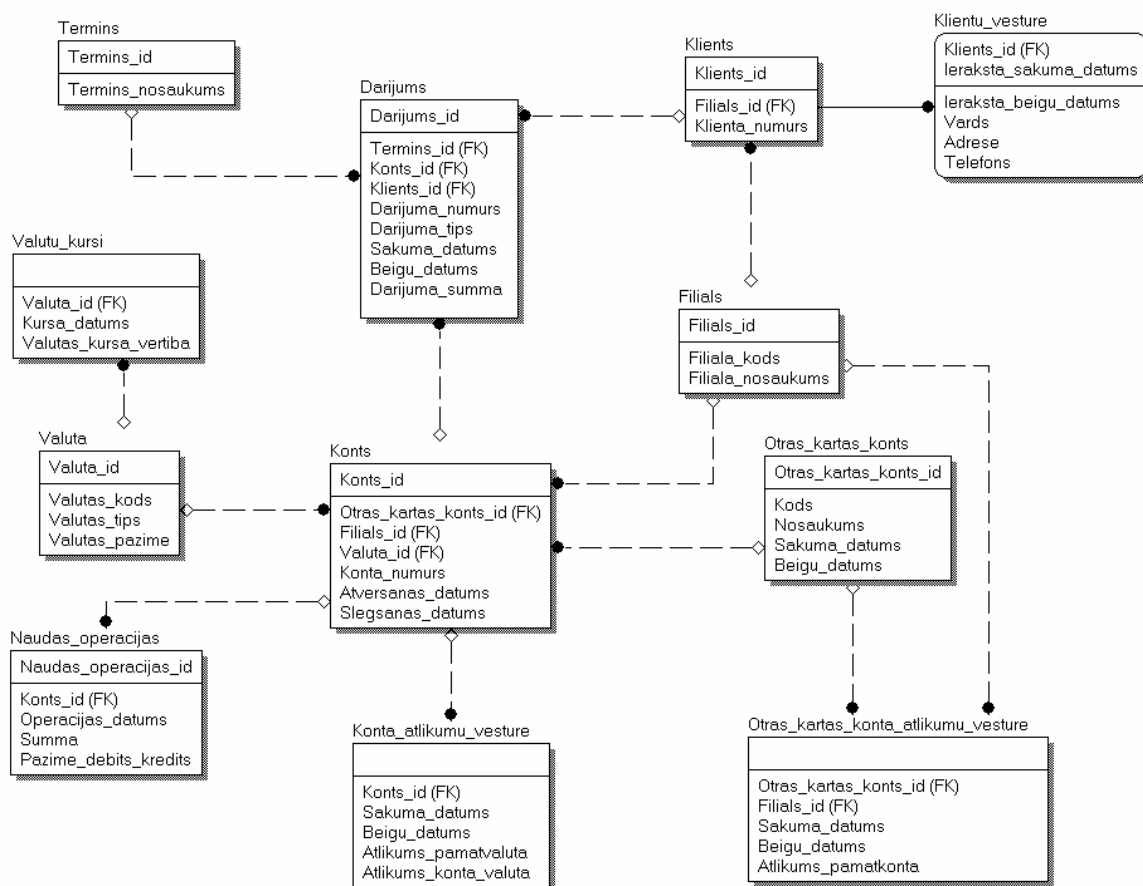
valūtas kursus, aizņēmēju datus. No tirdzniecības sistēmām integrē likmes, finanšu instrumentu kotējumus.

Datu noliktava realizē savākšanas, unifikācijas un datu glabāšanas funkcijas risku novērtēšanai. Tā nodrošina sekojošās procedūras:

- Līgumu portfeļu uzskaitē
- Virsgrāmatas uzskaitē
- Pārvaldības uzskaitē (aktīvu un pasīvu pārvaldības struktūru, ienākumu un izmaksu veidošanās)
- Klientu reģistra uzskaitē
- Tirgu informācijas vēsturisko arhīvu uzskaitē
- Operāciju risku statistikas uzskaitē
- Operāciju(darījumu) limitu saskaņošana produktu, klientu vai cita griezumā
- Aktīvu un pasīvu stāvokļu modelēšana maksājumu kalendāru pamatā
- Atskaišu veidošana [20]

Ar datu noliktavas datiem realizē risku aprēķinu un analīzi. Tās ļauj izveidot atskaites iekšējai pārvaldībai un auditam.

Apskatīsim bankas noliktavas piemēru. Veidosim modeli, kas attēlos nepieciešamu informāciju. Šī ir aptuvena bankas datu noliktavas shēma. Shēmas centrā ir attēlota darījumu tabula. Vienu darījumu ar banku slēdz viens klients uz kādu termiņu, norādot vienu kontu. Klientam ir vēstures tabula, kurā ir apsekot klientu rekvizītu izmaiņas laika gaitā. Klientam var mainīties vārds, adrese un telefons. Kontam ir numurs un otrās kārtas konts, kas ir vienāds ar konta numura pirmajiem pieciem cipariem. Kontam ir noteikta valūta. Valūtai ir valūtas kurss, kurš var laika gaitā mainīties. Valūtai ir pazīme vai tā ir pamatvalūta vai nē. Ir arī filiāls, kurā tas konts ir pierēģistrēts. Kontam un otrās kārtas kontam ir naudas summu atlikumu vēsture, kur var atsekot, cik naudas paliek konta. Noliktavā attēlotas naudas operācijas parāda cik liela naudas summa ir paņemta no konta vai uzlikta uz konta (to nosaka kredīta/debita pazīme) noteiktajā datumā.



## SECINĀJUMI

Šajā darbā tika apskatītas *IBM DB2* un *Oracle* produktu īpašības. Parādīts kā laika gaitā produkti attīstās un līdz ar to produktu priekšrocības mainās.

Kaut tika mēģināts salīdzināt abus šos produktus, tomēr tos nevar salīdzināt adekvāti, jo produktu izvēle atkarīga no kompānijas mērķiem un prioritātēm.

*IBM DB2 DWE* ir izplatīts produkts datu noliktavu veidošanā. Šim produktam ir arī risinājumi atsevišķām darbību sfērām, tajā skaitā arī bankām.

Darba rezultāti parāda, ka *DB2* ir laba izvēle datu noliktavu atbalstam.

## IZMANTOTĀ LITERATŪRA

1. Елманова Н. Часть 1. DB2 Universal Database - КомпьютерПресс 9'2001. [tiešsaiste]. [atsauce 17.05.2007.] Pieejams: <http://www.compress.ru/Archive/CP/2001/9/12/>
2. IBM DB2 UDB V8.2 / Oracle10g / Microsoft SQL Server 2000 – a technical comparison / Aut. kol. Bloemen J., Brunner G. – Germany: Bad Homburg, 2004. Pieejams: <ftp://ftp.software.ibm.com/software/data/highlights/dbmscomparison.pdf>
3. Кузнецов С. Обзор возможностей применения ведущих СУБД для построения хранилищ данных (DataWarehouse), 1998. [tiešsaiste]. [atsauce 17.05.2007.]. Pieejams: <http://www.citforum.ru/database/kbd98/glava15.shtml>
4. Деревянко А. Сравнительный анализ возможностей двух ведущих промышленных СУБД - Системы обробки інформації. Збірник наукових праць. Вип.6(16) - Харків: НАНУ, ПАНМ, ХВУ, 2001. - с.33-38. [tiešsaiste]. [atsauce 17.05.2007.]. Pieejams: [http://khpi-iip.mipk.kharkiv.edu/library/org/02/02\\_5.html](http://khpi-iip.mipk.kharkiv.edu/library/org/02/02_5.html)
5. Linux Databases: DB2 vs. Oracle by Branham Group - Executive Summary – Branham Group Inc., 2003. Pieejams: [ftp://ftp.software.ibm.com/software/data/db2/linux/db2oracle\\_exec.pdf](ftp://ftp.software.ibm.com/software/data/db2/linux/db2oracle_exec.pdf)
6. ITWare Решения для бизнеса: Пресс-релизы. [tiešsaiste]. [atsauce 17.05.2007.]. Pieejams: <http://itware.com.ua/pr/10610>
7. „DB2 „Viper” Foundation”, white paper, IBM Corp.[http://www-5.ibm.com/at/events/power/pdf/02\\_viper\\_foundation.pdf](http://www-5.ibm.com/at/events/power/pdf/02_viper_foundation.pdf)
8. Technical Comparison of Oracle Database vs. IBM DB2 UDB: Focus on Security – U.S.A.: Redwood Shores, 2000. Pieejams: [http://www.oracle.com/technology/deploy/security/oracle9ir2/pdf/CWP\\_9IVSDB\\_SECURITY.PDF](http://www.oracle.com/technology/deploy/security/oracle9ir2/pdf/CWP_9IVSDB_SECURITY.PDF)
9. Софт@Mail.Ru: Новости софта. [tiešsaiste]. [atsauce 17.05.2007.]. Pieejams: [http://soft.mail.ru/pressrl\\_page.php?id=15280](http://soft.mail.ru/pressrl_page.php?id=15280)
10. Закрома корпораций. Часть I. Что, зачем, как – СЮ, 2003. Pieejams: <http://www.cio-world.ru/offline/2003/12/24997>
11. ERP NEWS: Серверные технологии и сети. [tiešsaiste]. [atsauce 17.05.2007.]. Pieejams: [http://erpnews.ru/details.php?image\\_id=1225](http://erpnews.ru/details.php?image_id=1225)
12. Ведущая аналитическая компания признает, что СУБД от IBM легче в управлении и обеспечивает заказчикам более низкий уровень полной стоимости владения, 2002. [tiešsaiste]. [atsauce 17.05.2007.]. Pieejams:

<http://www.pool.kiev.ua/clients/poolhome.nsf/3fd9948be217d5dac225692000262223/6364043e12166702c2256b7200289534!OpenDocument>

13. “IBM DB2 Universal Database vs. Oracle9i: Total Cost of Ownership”, white paper, IBM Corp., Jan. 2002. Pieejams:

<ftp://ftp.software.ibm.com/software/emea/de/db2/DB2vsOracle.pdf>

14. Кэмпбелл С. СУБД – ИТ Бизнес CRN. , №15 (212), 2004. [tiešsaiste]. [atsauce 17.05.2007.]. Pieejams: <http://www.crn.ru/numbers/reg-numbers/detail.php?ID=9882>

15. DB2 Data Warehouse Edition *The top of the line IBM DB2 Edition for data warehousing, OLTP, and mixed workloads* [tiešsaiste]. – [atsauce 17.05.2007].

Pieejams: <http://www-306.ibm.com/software/data/db2/dwe/features.html>

16. DB2 Data Warehouse Edition *The top of the line IBM DB2 Edition for data warehousing, OLTP, and mixed workloads*. Components [tiešsaiste]. – [atsauce

17.05.2007]. Pieejams: [http://www-](http://www-306.ibm.com/software/data/db2/dwe/administration.html)

[306.ibm.com/software/data/db2/dwe/administration.html](http://www-306.ibm.com/software/data/db2/dwe/administration.html)

17. Leveraging DB2 Data Warehouse Edition for Business Intelligence / Aut.kol. Ballard C., Beaton A. – IBM Corp. 2006. Pieejams:

<http://www.redbooks.ibm.com/redbooks/SG247274/wwhelp/wwhimpl/js/html/wwhelp.htm>

18. IBM Corp., *DB2 demo program for DB2 9*. Pieejams: <http://www-128.ibm.com/developerworks/db2/library/demos/db2demo/>

19. “Banking Data Warehouse and the Basel II Accord,” white paper, IBM Corp., Dec. 2005. Pieejams: [http://t1d.www-](http://t1d.www-03.cacheibm.com/industries/financialservices/doc/content/bin/fss_bdw_baseII_rev_BDW34027.pdf)

[03.cacheibm.com/industries/financialservices/doc/content/bin/fss\\_bdw\\_baseII\\_rev\\_BDW34027.pdf](http://t1d.www-03.cacheibm.com/industries/financialservices/doc/content/bin/fss_bdw_baseII_rev_BDW34027.pdf)

20. Королева Е. Создание Хранилища договоров – первый шаг к автоматизации управления рисками - "Банки и технологии", 2007, №1. [tiešsaiste].

[atsauce 17.05.2007.]. Pieejams: <http://www.iso.ru/cgi-bin/main/public.cgi?id=209>

## DOKUMENTĀRĀ LAPA

Maģistra darbs DATU NOLIKTAVU ATBALSTS DATUBĀZU PĀRVALDĪBAS SISTĒMĀ  
DB2

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: \_\_\_\_\_  
(Autora paraksts)

Ar savu parakstu apliecinu, ka esmu lasījis augšminēto maģistra darbu un atzīstu to par p i e m ē r o t u / n e p i e m ē r o t u (nevajadzīgo svītrot) aizstāvēšanai Latvijas Universitātes datorzinātņu maģistrantūrā.

Darba vadītājs(-ja): \_\_\_\_\_  
(Vadītāja paraksts)

Darbs iesniegts Datorikas nodaļā \_\_\_\_\_  
(Iesniegšanas datums)

Ar šo es apliecinu, ka darba elektroniskā versija ir augšupielādēta LU informatīvajā sistēmā.

Metodiķe: \_\_\_\_\_  
(Metodiķes paraksts)

Recenzents: \_\_\_\_\_  
(Recenzenta paraksts)

Darbs aizstāvēts maģistra gala pārbaudījuma komisijas sēdē  
\_\_\_\_\_ prot. Nr. \_\_\_\_\_, vērtējums \_\_\_\_\_  
(Darba aizstāvēšanas datums)

Komisijas sekretārs: \_\_\_\_\_  
(Sekretāra paraksts)