

# Business Intelligence

## Cursul 6



Conf. Bologa Ana-Ramona  
ASE, Bucuresti

# Agenda

---

1. **Modelul de date asociativ**
2. **Solutii agile de business intelligence**
3. **Baze de date orientate pe coloane in contextul Big data**

# Modelul de date asociativ

---

- ❑ Este modelul de date utilizat de instrumentul **QlikView**
- ❑ Acest model **nu face distincție** între **atributele-fapte** și atributele-**dimensiuni**
- ❑ Cuvantul "asociativ" pune accent pe modul în care sunt legate seturile de date între ele
- ❑ Modelul este construit în jurul conceptului de **SET DE DATE**, cu tabelele logice asociate
- ❑ Utilizează tehnologia "in-memory", seturile de date sunt încărcate în memoria RAM comprimate, utilizând un script de încărcare

# Caracteristicile modelului asociativ

---

- ❑ Modelul de date este persistent si reactioneaza ca un intreg la interogariile utilizatorului; **o selectie afecteaza intreaga schema**
- ❑ Se elimina necesitatea de a dezvolta **ierarhii, hipercuburi** si **preagregari** de date
- ❑ Nu este necesara utilizarea unui **limbaj de interogare a datelor**
- ❑ Nu este necesara utilizarea unui **limbaj de definire a datelor**
- ❑ Fiecare comanda de incarcare sau de selectie genereaza o **tabela logica** in timpul procesului de incarcare
- ❑ **Agregarile** pot fi realizate:
  - in scriptul de incarcare si
  - in etapa de definire a interfetei, permitand o interactiune mai completa cu datele decat in SQL
- ❑ **Adaptabilitata** la cerintele de business si flexibilitate in analiza (orice valoare a oricarui atribut poate fi punct de plecare in analiza)
- ❑ **Proiectarea mai rapida** a modelului

# Asocierile

---

- ❑ Asocierile între tabele logice sunt **generate automat** în timpul procesului de încărcare a datelor pe baza potrivirilor de nume de coloane din tabelele logice
- ❑ Orice câmpuri cu același nume din două sau mai multe tabele vor fi asociate; relațiile între tabelele logice **nu reflecta, de obicei, relații de tip cheie externă**;
- ❑ asocierile dintre tabele sunt similare joinurilor externe complete (*full outer join*)
- ❑ Dacă există mai mult de 1 câmp cu același nume se creează o **CHEIE SINTETICĂ**

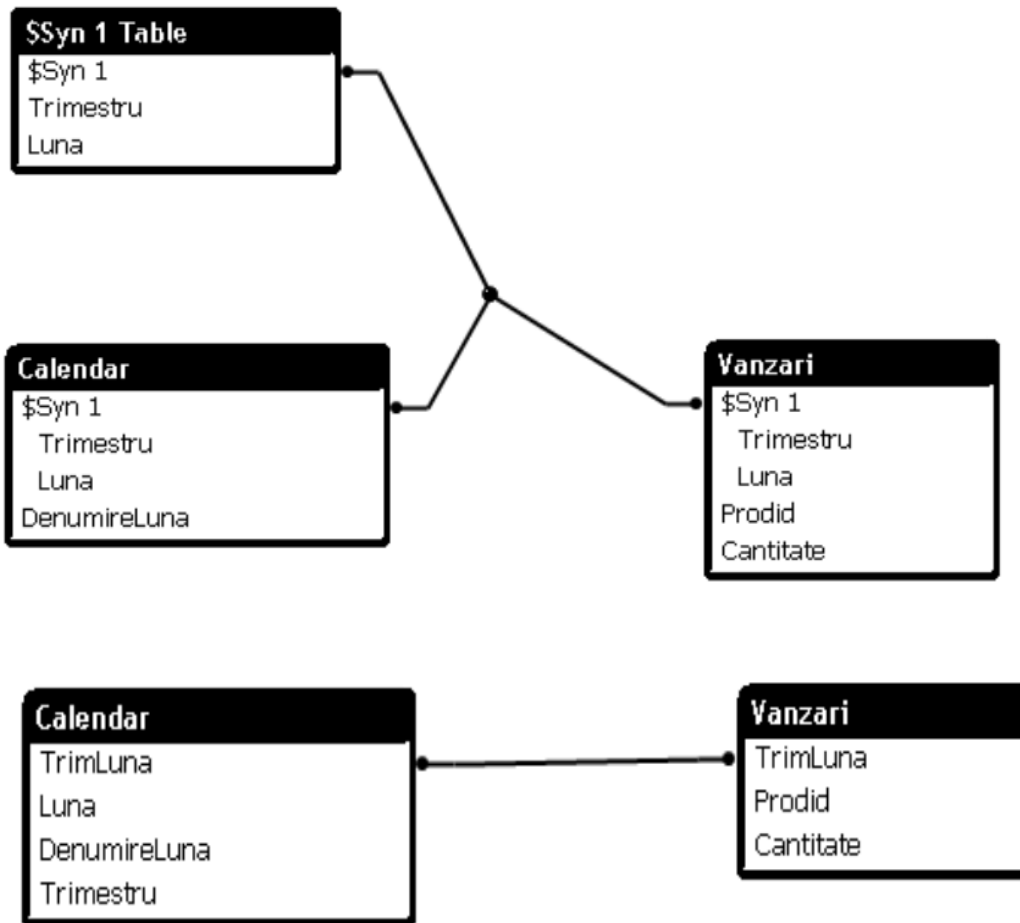
# Cheia sintetica

---

- ❑ Contine **toate combinatiile posibile** ale atributelor comune tabelelor
- ❑ Solicita intens resurse si fac modelul de date greu de urmarit
- ❑ Solutii:
  - a) **Redenumirea atributelor** cheii sintetice *care nu sunt parte a asocierii* intre seturile de date
  - b) **Stergerea atributelor comune** intr-unul din seturile de date (daca nu sunt necesare in ambele seturi)
  - c) **Crearea de chei compuse** prin **concatenarea campurilor comune celor doua seturi** de date, urmata de stergerea atributelor comune din scriptul de incarcare
  - d) Concatenarea tabelelor logice sau utilizarea **tabelelor logice de legatura**

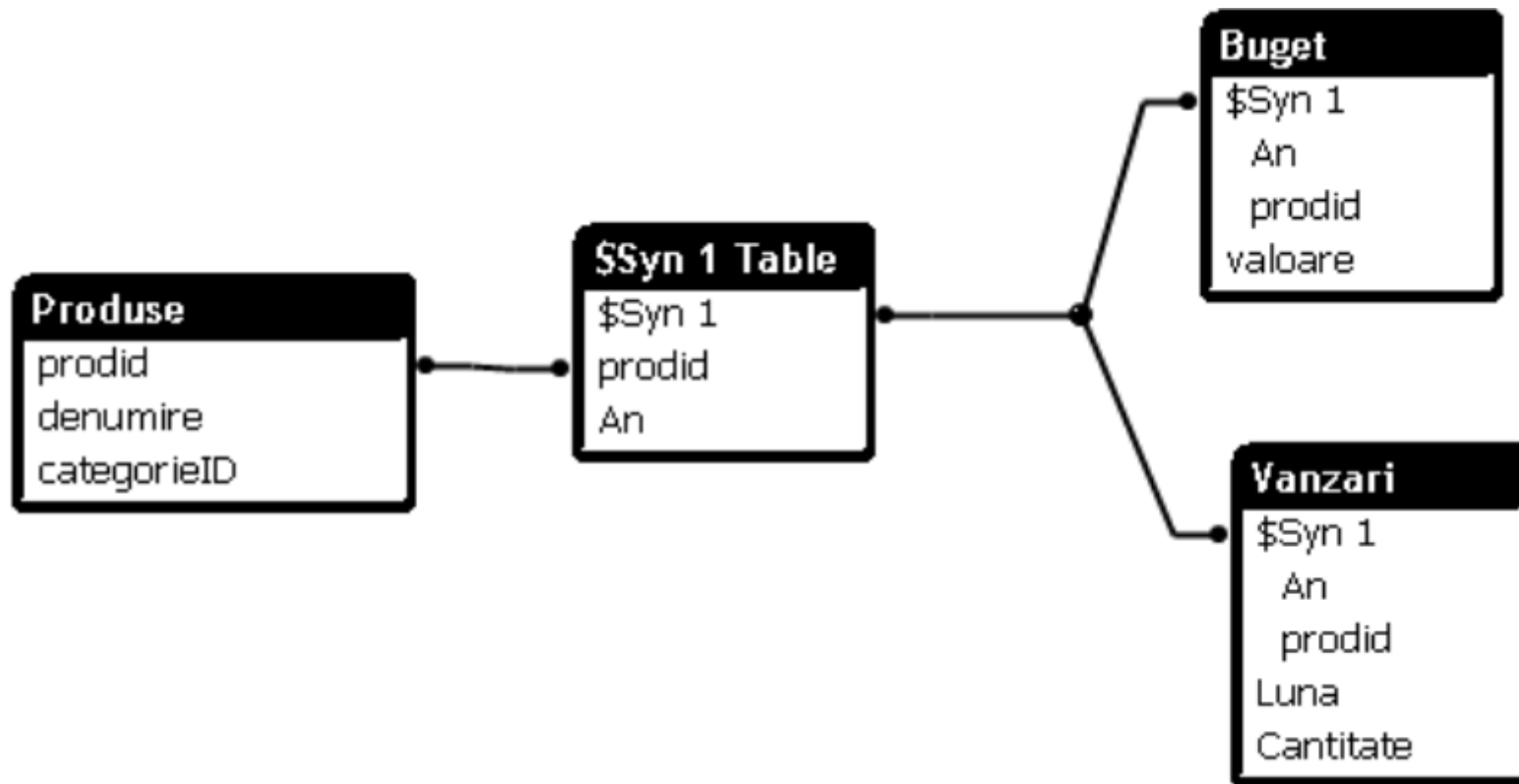
# Cheie sintetica – cheie concatenata

---



# Cheie sintetica – ex. Tabele de legatura

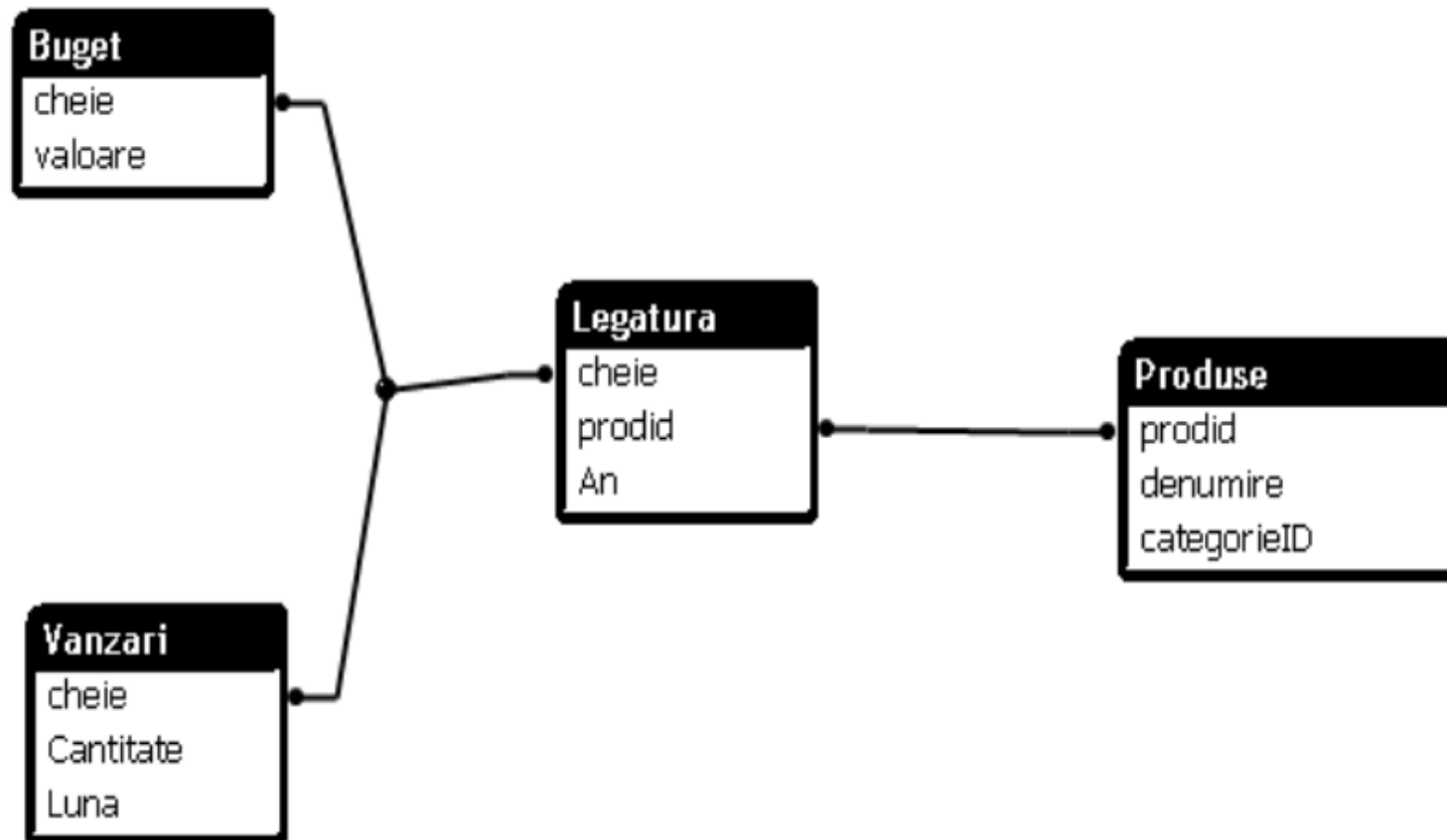
---





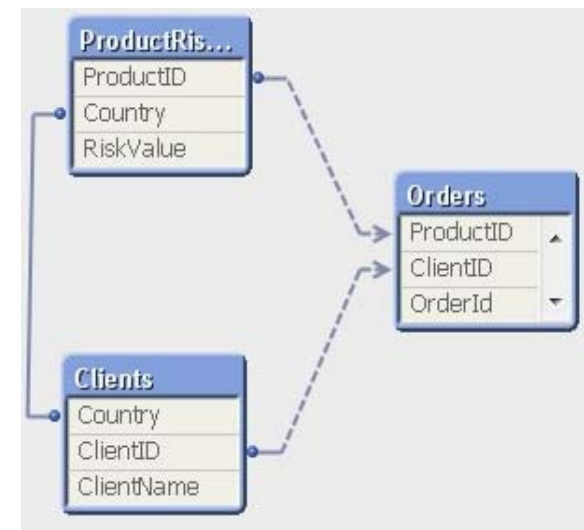
# Cheie sintetica – ex. Tabele de legatura

---



# Referinte circulare

- ❑ Este o bucla logica ce apare atunci cand exista 2 sau **mai multe cai de asociere** intre 3 sau mai multe tabele.
- Poate conduce la **rezultate ambigue**
- **Mesaj de avertizare** in QlikView, care va seta una dintre tabele ca slab cuplata (***loosly coupled***)- nu se mai fac inferente logice prin intermediul ei
- ❑ Frecvent cand in modelul asociativ se incarca **tabele de fapte multiple** cu **granularitati diferite** si care au mai **multe dimensiuni comune**



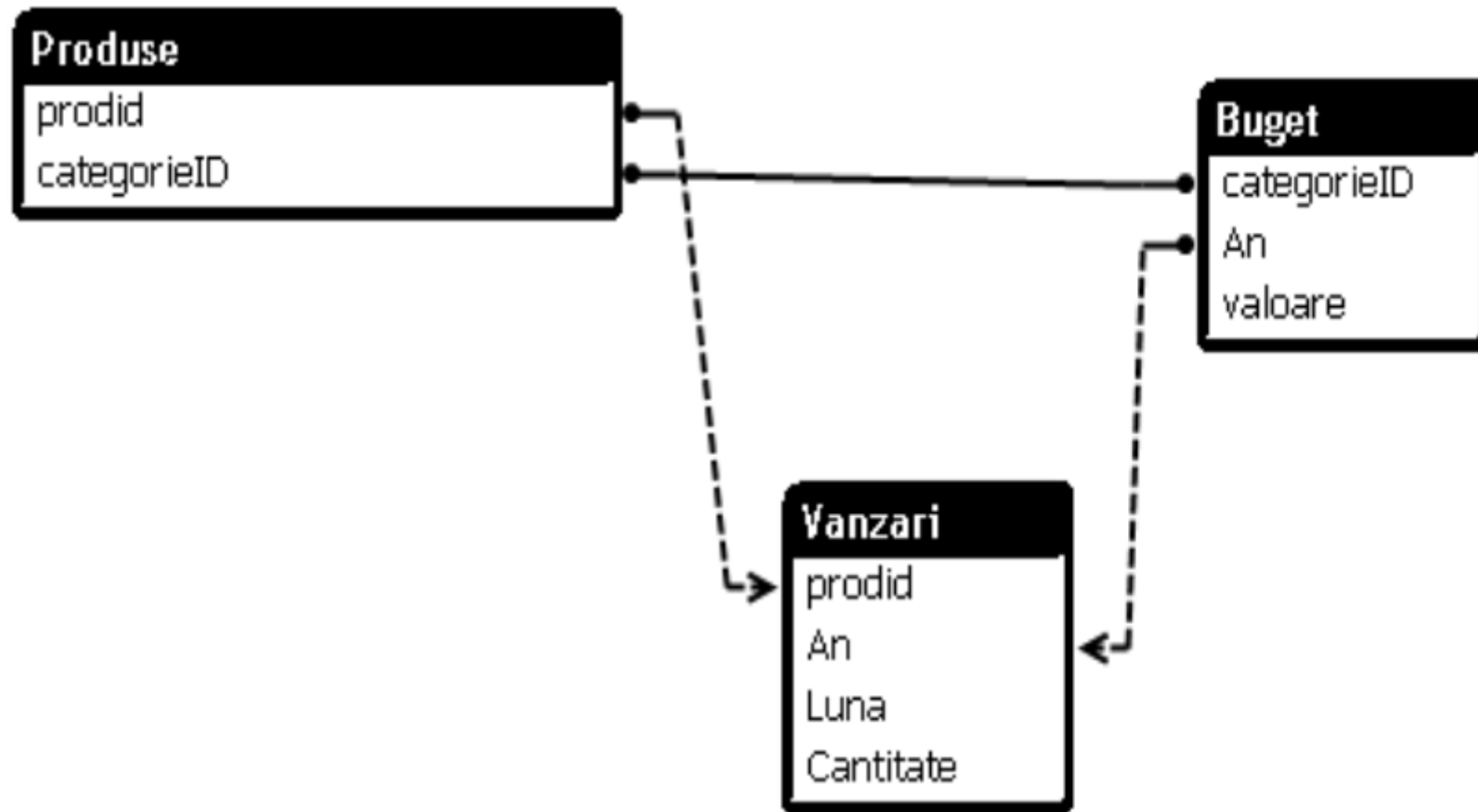
# Eliminarea dependentei circulare

---

## □ Solutii:

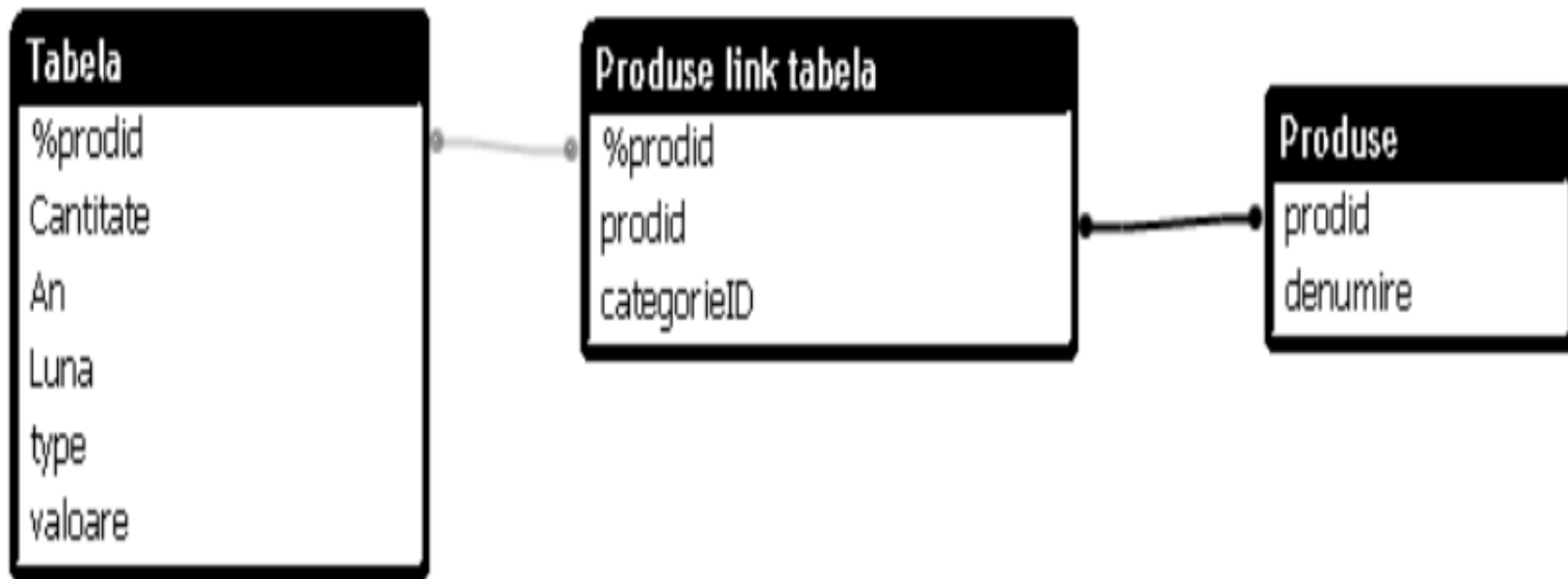
1. **Eliminarea unuia din campuri** daca nu este necesar in toate seturile de date (dar creeaza asocieri in plus)
2. **Redenumirea campurilor** care duc la aparitia dependentei circulare, daca stim ca se refera la entitati diferite
3. **Concatenarea** a 2 tabele, daca acestea contin aproximativ aceleasi date, cu ajutorul unor **CHEI GENERICI**, care trebuie sa fie mapate pe chei reale. Cheile generice contin alte valori decat cele ale campurilor, ca o modalitate de a grupa si inregistrari care inca nu exista
4. Utilizarea **tabelelor logice de legatura**, in cazul in care avem campuri cu aceeasi denumire (dimensiuni) iar campurile cu indicatori sunt diferiti intre tabele

# Dependente circulare - exemplu



# Dependente circulare – cheie generica

---



# Agenda

---

1. **Modelul de date asociativ**
2. **Solutii agile de business intelligence**
3. **Baze de date orientate pe coloane in contextul Big data**

# Dezavantajele sist. BI tradiționale

<b>Dezavantajele sistemelor BI tradiționale</b>	<b>Probleme</b>
Volum uriaș de date duplicate	a. inconsistența datelor b. calitatea datelor
Instrumente diferite pentru task-uri diferite	a. metadate diferite, nepartajabile b. rezultate inconsistente
Modelele (relațional și multidimensional) destul de rigide	a. flexibilitate limitată la schimbare b. suport limitat pentru analiza datelor nestructurate
Metodologie de dezvoltare în cascadă	a. durata de dezvoltare mare b. utilizatorii nu sunt implicați în ciclul de dezvoltare c. inflexibilitate la modificările cerințelor analitice d. testarea doar la sfârșitul ciclului de dezvoltare

# O soluție BI agilă ...

---

## □ Forrester Research:

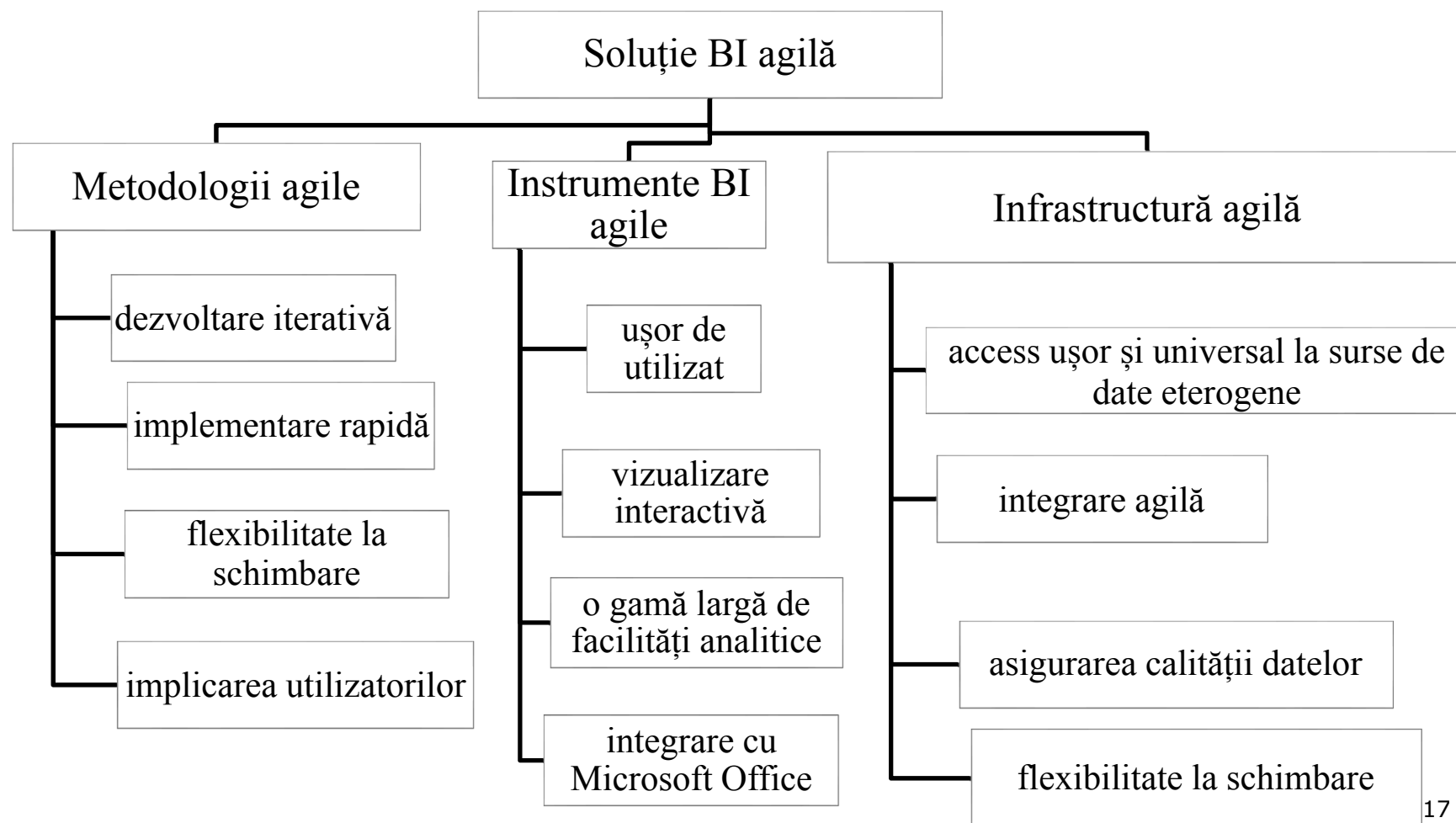
- „o nouă abordare ce combină procese, metodologii, structura organizațională, instrumente și tehnologii care permit decidenților de la nivel strategic, tactic și operațional de a se adapta ușor la modificările cerințelor de business”

## □ Data Warehousing Institute:

- „include tehnologii cum ar fi *self-service BI, BI bazat pe cloud, tablouri de bord* care permit utilizatorilor să analizeze datele mult **mai rapid** și să le **ajusteze în funcție de modificările cerințelor de business**”



# Elemente care promovează o soluție BI agilă



# Utilizarea unei metodologii de dezvoltare agilă

---

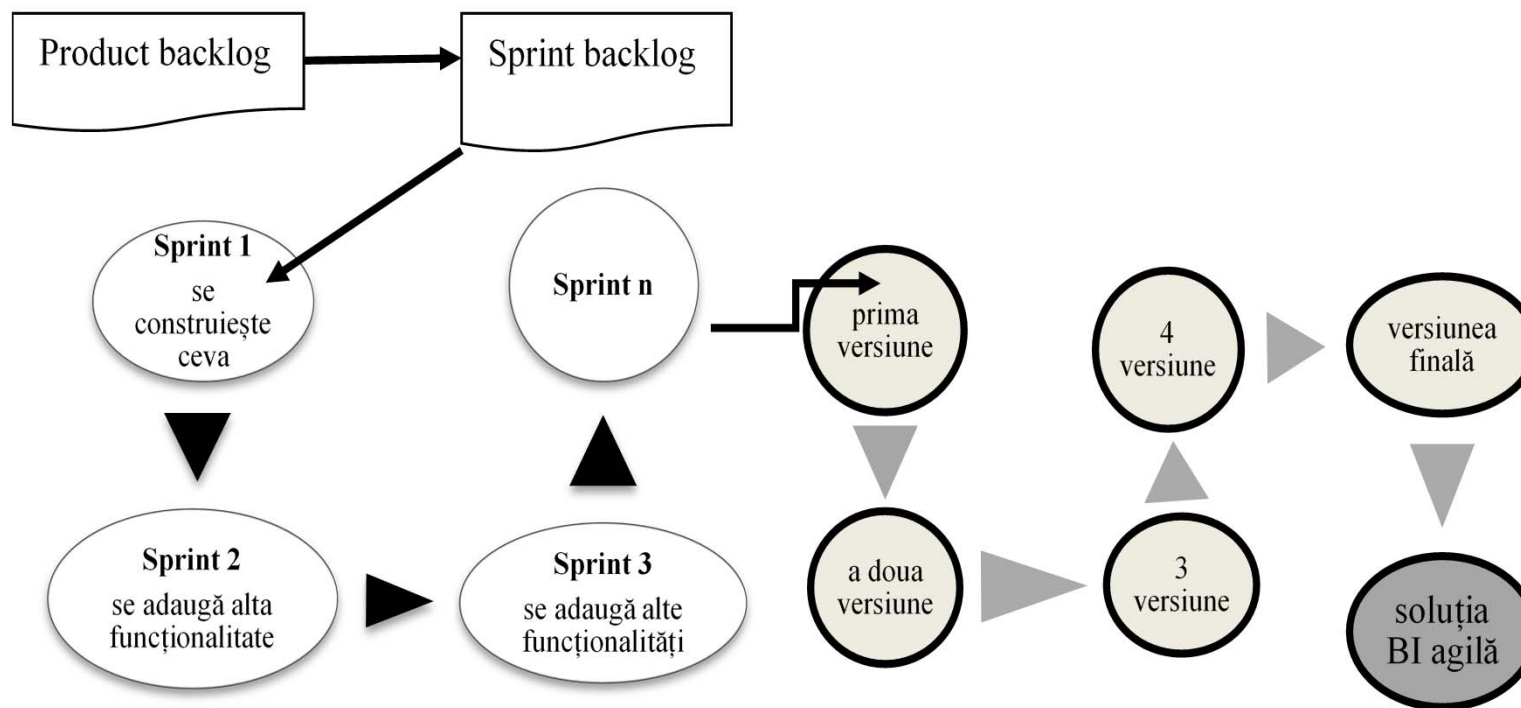
- ❑ **Scopul** unei soluții BI agilă este de a „fi dezvoltată rapid și de a reacționa rapid la modificările cerințelor de business”
- ❑ Există un număr mare de **metodologii agile** cum ar fi:
  - ❑ Scrum, Extreme Programming, Crystal, Dynamic Systems Development, Lean, etc.
- ❑ Cele mai utilizate metodologii agile pentru dezvoltarea soluțiilor BI sunt:
  - ❑ Scrum,
  - ❑ Extreme Scoping și
  - ❑ Agile Data Warehousing.

# Metodologia SCRUM

---

- ❑ Cerințele analitice sunt împărțite în mai multe "**user story**".
- ❑ Aceste "**user story**" sunt utilizate pentru a stabili "**product backlog**", adică o listă cu toate [cerințele analitice](#), ordonate de utilizatori, în funcție de priorități.
- ❑ "**Sprint backlog**" este o [listă de task-uri](#) pe care echipa trebuie să le execute într-un "**sprint**"/ciclu (1-2 săptămâni).
- ❑ Un "**sprint**" include ca etape:
  - identificarea cerințelor,
  - analiză,
  - proiectare,
  - dezvoltare și
  - testare.
- ❑ La sfârșitul unui "**sprint**" există un **livrabil** (de exemplu, un raport/tablou de bord).
- ❑ Membrii echipei analizează stadiul proiectului într-un "**daily scrum**".

# Utilizarea metodologie SCRUM pentru dezvoltarea soluțiilor BI

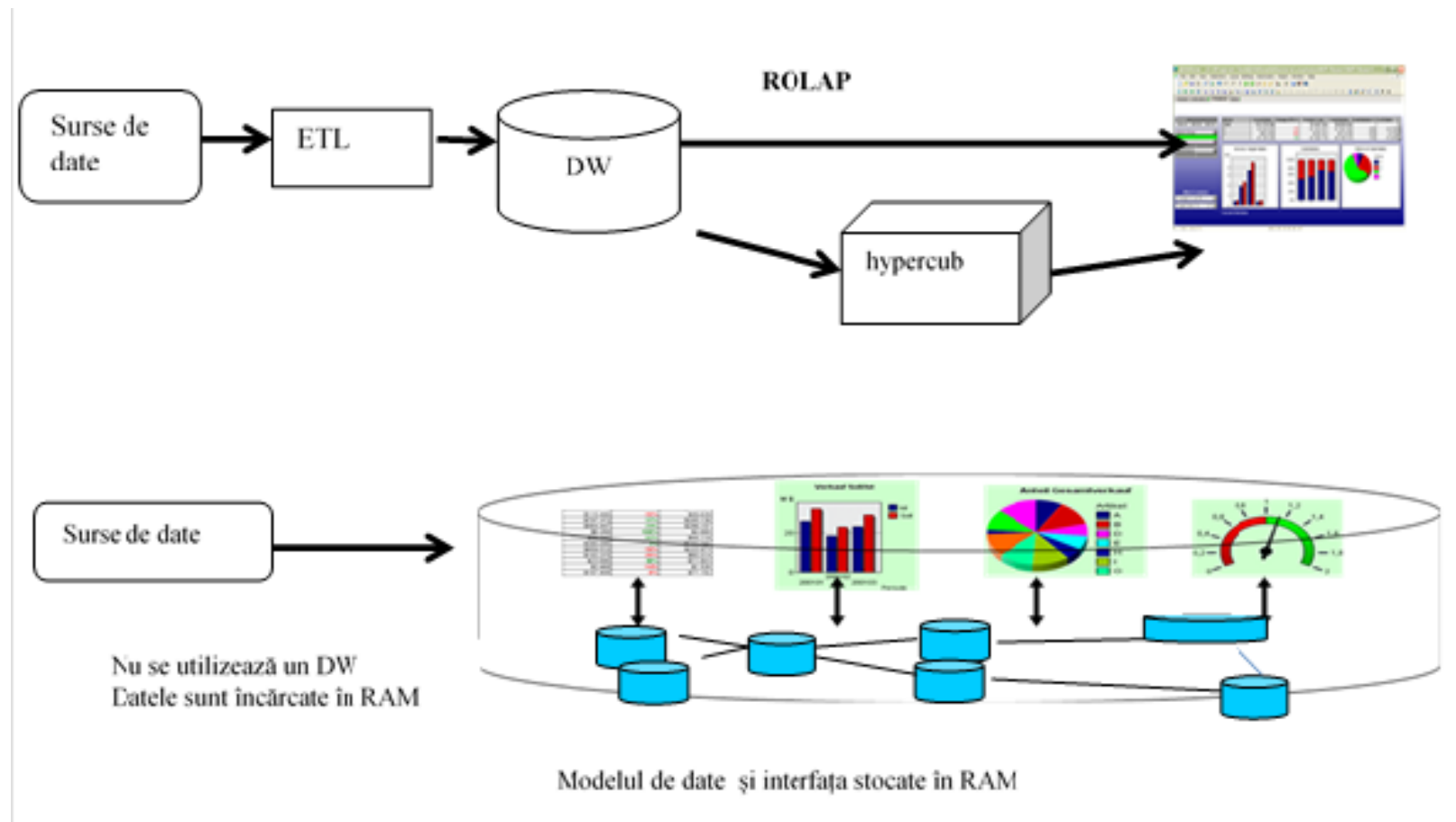


# BI „in-memory”

---

- Are potențial de a oferi agilitate soluțiilor BI
- Obiectiv: eliminarea stocării pe disc a datelor
- Tehnologiile BI „in-memory” sunt mai **rapide**, deoarece încarcă tot setul de date analizat în memoria RAM
- Se elimină, de asemenea, nevoia de a construi agregate și de a le stoca în cuburi/tabele de agregate, precum și de proiecta cuburi/scheme stea complexe.
- Dar memoria **RAM** este mult mai scumpă decât discul și sistemele pe **64 biți** au o limită teoretică de **16.3 exabytes de RAM**
- Majoritatea tehnologiilor BI „in-memory” utilizează **tehnici de compresie complexe și stocare pe coloane**, pentru a îmbunătăți eficiența compresiei

# Soluție BI tradițională versus soluție BI „in-memory”



# Soluții BI “in-memory”

Soluția	Caracteristici	Exemple	Limbajul de interogare	Modelul de date
<b>1.In-memory OLAP</b>	-cub MOLAP încărcat în memorie	-IBM <b>Cognos-Applix</b> (TM1) -Actuate BIRT - tehnologia Dynamic Cubes - <b>Cognos BI</b> versiune 10.1	- <b>MDX</b> sau alt limbaj de interogare multidimensional	-cub n-dimensional
<b>2.In-memory ROLAP</b>	-metadate ROLAP încărcate în memorie	- <b>MicroStrategy</b>	<b>SQL</b>	-model dimensional -cub n-dimensional
<b>3.O bază de date orientată pe coloane cu tehnici de compresie complexe</b>	- stochează datele într-o bază de date orientată pe coloane	<b>Tableau Software</b>	- <b>VizQL</b> –limbaj declarativ	-se pot accesa baze de date relaționale/multidimensionale
<b>4.In memory spreadsheet</b>	- foaie de calcul tabelar încărcată în memorie	- <b>PowerPivot</b> utilizează VertiPaq (stocare pe coloane)	- <b>DAX</b> (Data Analysis Expression).	-nu cere modelarea datelor

# Soluții BI “in-memory” (cont)

Soluția	Caracteristici	Exemple	Limbajul de interogare	Modelul de date
<b>5.Model de date asociativ</b>	<ul style="list-style-type: none"> <li>-stochează datele într-un model “asociativ” încărcat în memorie</li> <li>-toate joncțiunile și calculele se fac în timp real</li> <li>-scripturi pentru încărcarea și transformarea datelor</li> <li>-stocare pe coloane cu tehnici de compresie (raport 10:1)</li> </ul>	<p><b>QlikView</b> include Expressor – instrument ETL</p>	<ul style="list-style-type: none"> <li>-Limbaj de script pentru încărcarea datelor și generarea automată a modelului</li> <li>-tehnologia <b>AQL/</b> Associative Logic -nu are limbaj de interogare/definire</li> </ul>	<ul style="list-style-type: none"> <li>-fără agregări, ierarhii, cuburi</li> <li>-poate accesa schema stea/fulg de zăpadă/cuburi</li> </ul>
<b>6.Soluție hibridă cu tehnici de compresie</b>	<p>Bază de date relațională + bază de date orientată pe coloane</p>	<p><b>Oracle Exalytics</b> In-memory (include Essbase, in -memory TimesTen database) <b>SAP HANA</b></p>	<p><b>SQL</b></p>	<ul style="list-style-type: none"> <li>-modelul dimensional</li> <li>-cub n-dimensional</li> </ul>
<b>7.Soluție de stocare hibridă (disk + RAM)</b>	<ul style="list-style-type: none"> <li>-MOLAP - stochează agregatele și datele atomice pe disc</li> <li>-Tabular stochează datele atomice în memorie, pe coloane</li> </ul>	<p><b>SQL Server 2012</b> -motor xVelocity</p>	<ul style="list-style-type: none"> <li>-<b>MDX</b> pentru multidimensional</li> <li>- <b>DAX</b> pentru tabular</li> </ul>	<ul style="list-style-type: none"> <li>-schema stea pentru MOLAP</li> <li>-modelare relațională pentru modul tabular</li> </ul>



# BI tradițional versus BI agil

CRITERIU	BI TRADIȚIONAL	BI AGIL
<b>Cerințele de business</b>	-sunt clare, bine definite, nu se modifică semnificativ in timp	- nu sunt clar definite - se modifică frecvent
<b>Modul de integrare</b>	- instrumente ETL - date replicate	-virtualizarea datelor/surse în cloud
<b>Tipul de date</b>	date istorice	date în timp real
<b>Actualizarea datelor</b>	sfârșitul zilei	aproape în timp real
<b>Livrarea informațiilor</b>	durează foarte mult	rapidă
<b>Surse de date</b>	-structurate (BDR/BDMD), fișiere Excel, semistructurate	structurate, semistructurate, nestructurate, Big data
<b>Metodologia de dezvoltare</b>	în cascadă	metodologii agile
<b>Ciclul de dezvoltare</b>	-inflexibil, nu permite modificări -durează prea mult	-permite modificări -timpul de dezvoltare mult mai mic
<b>Instrumente BI</b>	tradiționale (raportare, cereri ad-hoc, OLAP)	instrumente BI agile

# Agenda

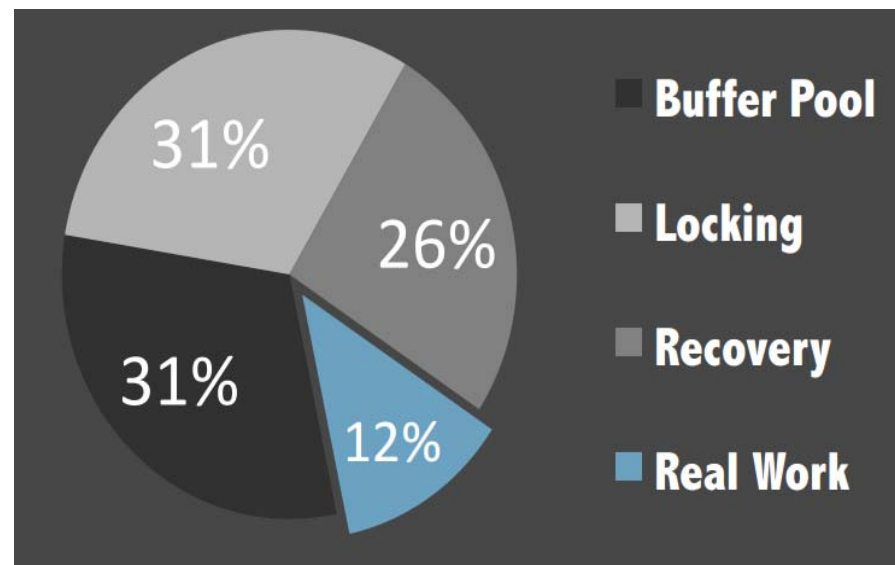
---

1. **Modelul de date asociativ**
2. **Solutii agile de business intelligence**
3. **Baze de date orientate pe coloane in contextul Big data**

# Dificultati in analiza datelor

---

- ❑ **Big data** – volumul, varietatea, diversitatea – genereaza dificultati in culegerea, curatarea si procesarea datelor utilizand BD clasice
- ❑ Michael Stonebraker, profesor MIT – modelul relational clasic este invechit, prea lent



# In 2010...

---

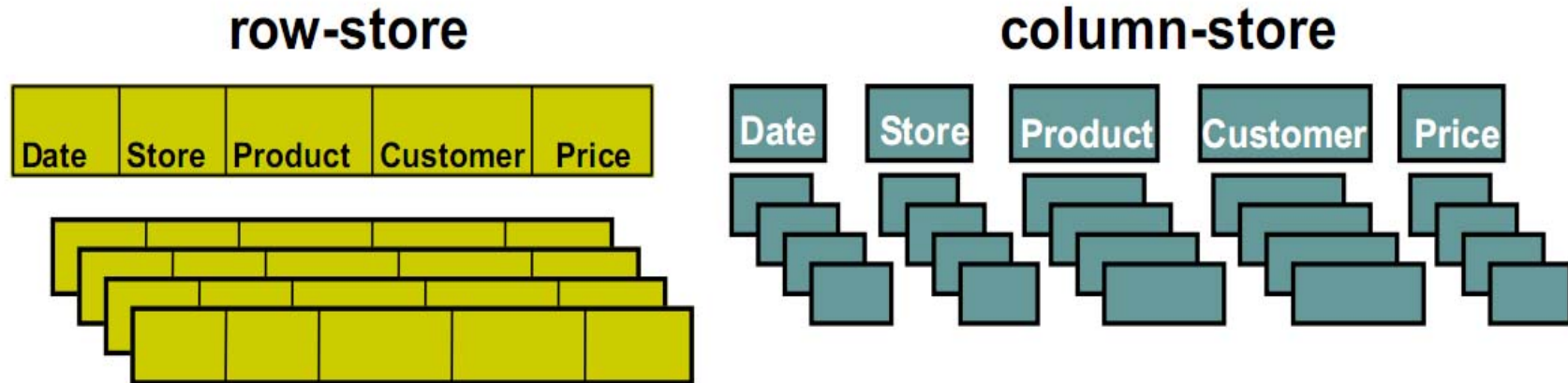
- Un **studiu al IDC** prognoza ca in urmatoorii **5 ani**:
  - Majoritatea depozitelor de date vor fi stocate **pe coloane**;
  - Cele mai multe BD pentru OLTP (On-Line Transaction Processing) vor fi completate sau inlocuite de o **baza de date in memory**;
  - Cele mai multe servere de baze de date mari vor realiza scalabilitate orizontala prin **clusterizare**;
  - Multe dintre problemele cu colectarea datelor si raportare vor fi rezolvate cu baze de date care nu vor avea **nici o schema formala**

# Bazele de date orientate pe coloane

---

- In **1969** – **TAXIR** - destinat domeniului biologiei.
- In **1976**, sistemul **RAPID** pentru procesarea datelor provenite din recensământul populației și al locuințelor din Canada.
- **Sybase IQ** , aparut la **începutul anilor '90**
  - Solutie foarte performanta de BI
  - Multa vreme singurul SGBD orientat pe coloane disponibil comercial.

# Depozite pe linii si depozite pe coloane



- In depozitele orientate pe linii, datele sunt stocate pe disc inregistrare dupa inregistrare
- In depozitele orientate pe coloane datele sunt stocate pe disc coloana dupa coloana

# De ce depozite orientate pe coloane?

---

- Punand informatiile similare impreuna, **minimizeaza timpul pentru citirea discului**
- Pot fi **mult mai rapide pentru anumite tipuri de aplicatii**
  - Incarca doar coloanele necesare intr-o interogare
  - Efectele cache sunt mai bune
  - Compresie mai buna (valori similare ale atributelor intr-o coloana)
- Pot functiona mai lent pentru alte aplicatii:
  - OLTP cu multe linii de inserat

# Depozite pe linii, depozite pe coloane

Depozit pe linii	Depozit pe coloane
(+) Se adauga/modifica usor inregistrari	(+) Trebuie sa citeasca doar datele relevante
(-) Pot citi date care nu sunt necesare	(-) Scrierile de tupluri necesita accesari multiple

- depozitele orientate pe coloane sunt potrivite pentru **depozite mari de date** in care se realizeaza intensiv **operatii de citire**, sau in care aceste operatii sunt preponderente: DW, OLAP/DSS



# Optimizarea executiei orientate pe coloane

---

- **Optimizarile** sunt diferite in cazul bazelor de date orientate pe coloane
  - Compresie - entropia scazuta -> rate de compresie ridicate
  - Materializare intarziata
  - Iterarea blocurilor
  - Join invizibil

# Tehnici de procesare paralela

---

- ❑ **Massive Parallel Processing -MPP** (*grid computing* sau *computer cluster*)– fiecare dintre procesoare e conectat la propria structura persistenta de stocare, datele fiind distribuite; pot fi adaugate un numar nelimitat de procesoare
- ❑ **Symmetric Multi-Processing – SMP**- mai multe procesoare identice se conecteaza la o singura memorie partajata, partajeaza sistemele de I/O si sunt controlate de o singura instanta de sistem de operare care le trateaza in mod egal

# Exemple pe piata?

---

Big Table

Vertica

SAP HANA

# VERTICA

- un SGBD **relational, distribuit, paralel**
- unul dintre putinele SGBD-uri care este utilizat pe scara larga în sisteme critice de business
- peste 500 de implementari de productie ale Vertica, cel puțin 3 dintre ele avand peste 1 petabyte dimensiune
- Vertica a fost conceput în mod explicit pentru **sarcini analitice**

Decimal	
Value	SI
1000	k kilo-
1000 <sup>2</sup>	M mega-
1000 <sup>3</sup>	G giga-
1000 <sup>4</sup>	T tera-
1000 <sup>5</sup>	P peta-
1000 <sup>6</sup>	E exa-
1000 <sup>7</sup>	Z zetta-
1000 <sup>8</sup>	Y yotta-

# Avantaje Vertica

---

- ❑ este conceput pentru a **reduce operatiile de I/E pe disc** - abordarea orientata pe coloane
- ❑ este scris nativ pentru suport **grid computing**
- ❑ interogariile sunt de 50-200 de ori mai rapide decât la bazele de date orientate pe linii.
- ❑ **arhitectura MPP** ofera o **scalabilitate** mai buna - poate fi realizata prin adaugarea de noi servere în arhitectura grid.
- ❑ utilizeaza mai multi **algoritmi de compresie**, în functie de tipul de date, cardinalitate si ordinea de sortare a fiecărei coloane (selectat automat prin esantionare)
- ❑ raport de compresie 8-13 ori fata de date originale

# Modelul de date Vertica

---

- Dpdv **logic** – datele sunt privite ca tabele de coloane
- Dpdv **fizic** - datele sunt organizate in **proiectii**, subseturi de date sortate ale unei tabele; pot exista oricate proiectii cu diverse combinatii de coloane si ordini de sortare
- Cel putin o **superproiectie** care sa contina toate coloanele tabelei de referinta
- un **sistem de stocare distribuita** complet implementat, care atribuie tuplurile pe diferite noduri de calcul.
- Suporta **INSERT, UPDATE, DELETE** pentru actualizarea datelor si toate **comenzile SQL de interogare** a datelor

# BigTable

- un sistem de stocare distribuit pentru gestiunea datelor structurate, proiectat pentru volume foarte mari de date (petabytes) repartizate pe mii de servere
- proiectele Google: indexarea Web, Google Earth, Gmail, YouTube si Gogle Finance
- adesea folosit impreuna cu MapReduce
- Toleranta la erori, persistenta
- Scalabil
  - Mii de servere
  - Terabytes de date in-memory
  - Petabyte de date pe disc
  - Milioane de citiri/scrieri pe secunda, scanari eficiente
- Auto-gestionabil
  - Serverele se pot adauga/ sterge dinamic
  - Serverele fac ajustari la dezechilibre in incarcare

# Modelul de date

- nu ofera un model de date complet relational:
  - Nu avem restrictii de integritate la nivel de tabela
  - Nu avem tranzactii multirand
- datele sunt indexate folosind denumirile liniilor si coloanelor care pot fi siruri de caractere arbitrare
- BigTable mapeaza doua valori sir de caractere arbitrare (**cheie linie** , **cheie coloana**) si o **marca de timp**, realizand o mapare tridimensionala intr-un sir asociat de octeti



# Tablete

---

- Tabelele de mari dimensiuni sunt sparte dupa linii in **TABLETE**
  - Tabletele pastreaza intervale contigue de linii
  - Clientii pot alege cheile liniilor pentru a obtine localizarea
  - Scopul este de a segmenta in tablete de **100-200 MB**
- Fiecare **masina server** gestioneaza de obicei in jur de **100 tablete**
  - **Recuperare rapida**: Fiecare din 100 masini preiau o tableta pentru masina care cade
  - **Echilibrarea fina a incarcarii**
    - Migrarea tabelelor de pe masini supraincarcate
    - **Masinile Master** iau deciziile referitoare la echilibrarea incarcarii

# SAP HANA

---

- ❑ Platforma pentru analize in timp real si aplicatii in timp real
- ❑ Baza de date **pe coloane, in-memory**
- ❑ **Motor de calcul in-memory**
- ❑ Componente software optimizate pentru hardware-ul de la Dell, Cisco, IBM, HP, Fujitsu si Intel , foloseste din plin **memoriile flash**
- ❑ Conceput în jurul unei **arhitecturi multi-core** (cel puțin 1.000 de nuclee)
- ❑ Implementeaza **paralelizarea dinamica** si **partitionare dinamica**, atât pentru OLAP cat si pentru volumul de lucru OLTP, prin **algoritmi genetici**

# Avantaje

---

- Motorul SAP foloseste un **depozit pe coloane in memorie** pentru a obtine performanța în operațiunile de scanare, grupare si de agregare. => scaneaza **2 milioane de înregistrari pe miliseconda pe nucleu** si peste 10 milioane de agregari complexe calculate pe secunda pe nucleu.
- Din 2013 – SAP HANA in cloud