**BDO vs. DD (DWH)**

**Diferența dintre BDO și DD(DWH)**

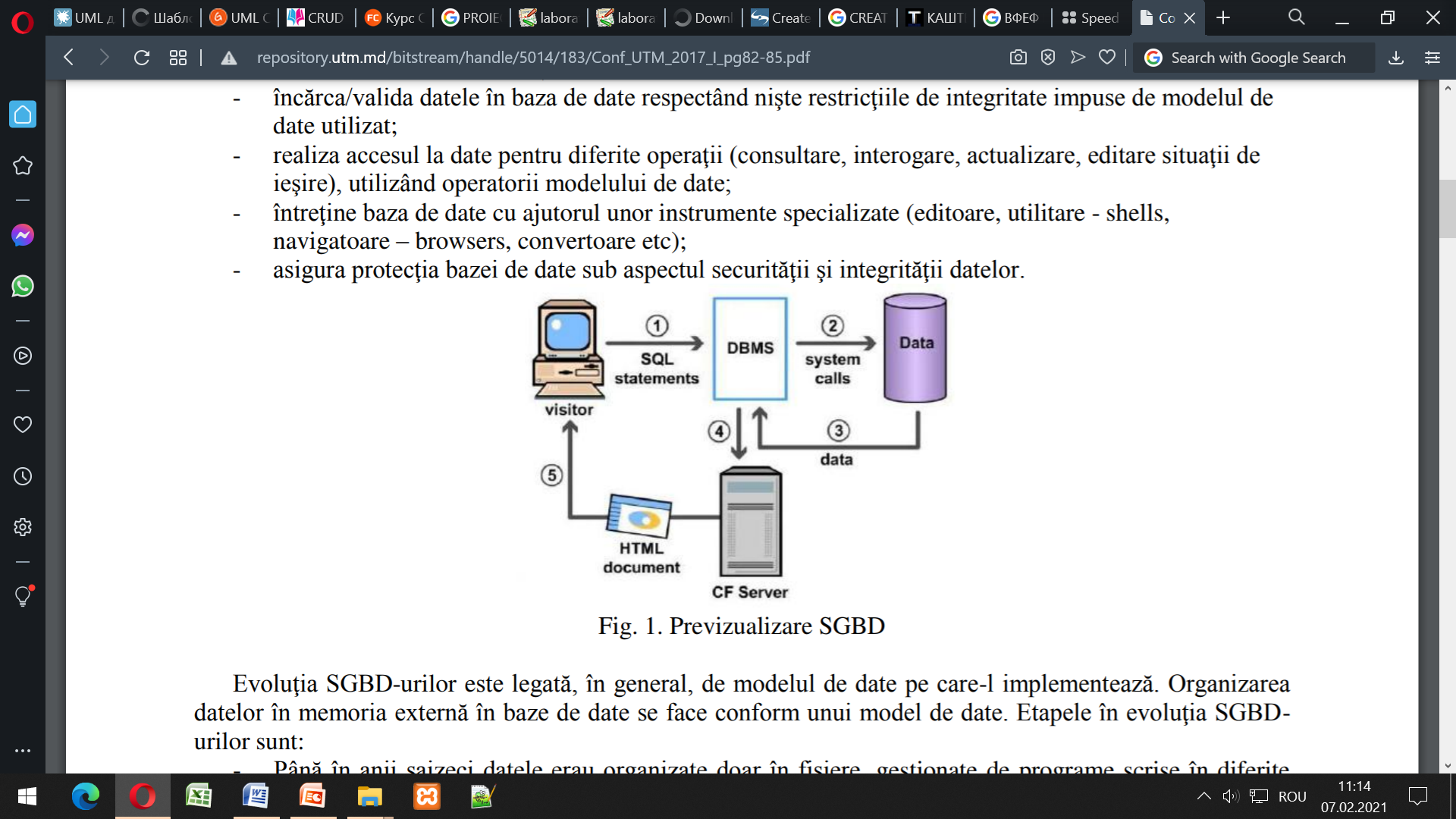
* **Ce este baza de date operațională (BDO)?**
* **Ce este un Depozit de Date (DD), sau Data WareHouse (DWH)?**
* **Ce semnificatie are expresia “Modelare multidimensională”**
* **Diferența dintre BDO și DD**

**Ce este o baza de date operațională (BDO)?**

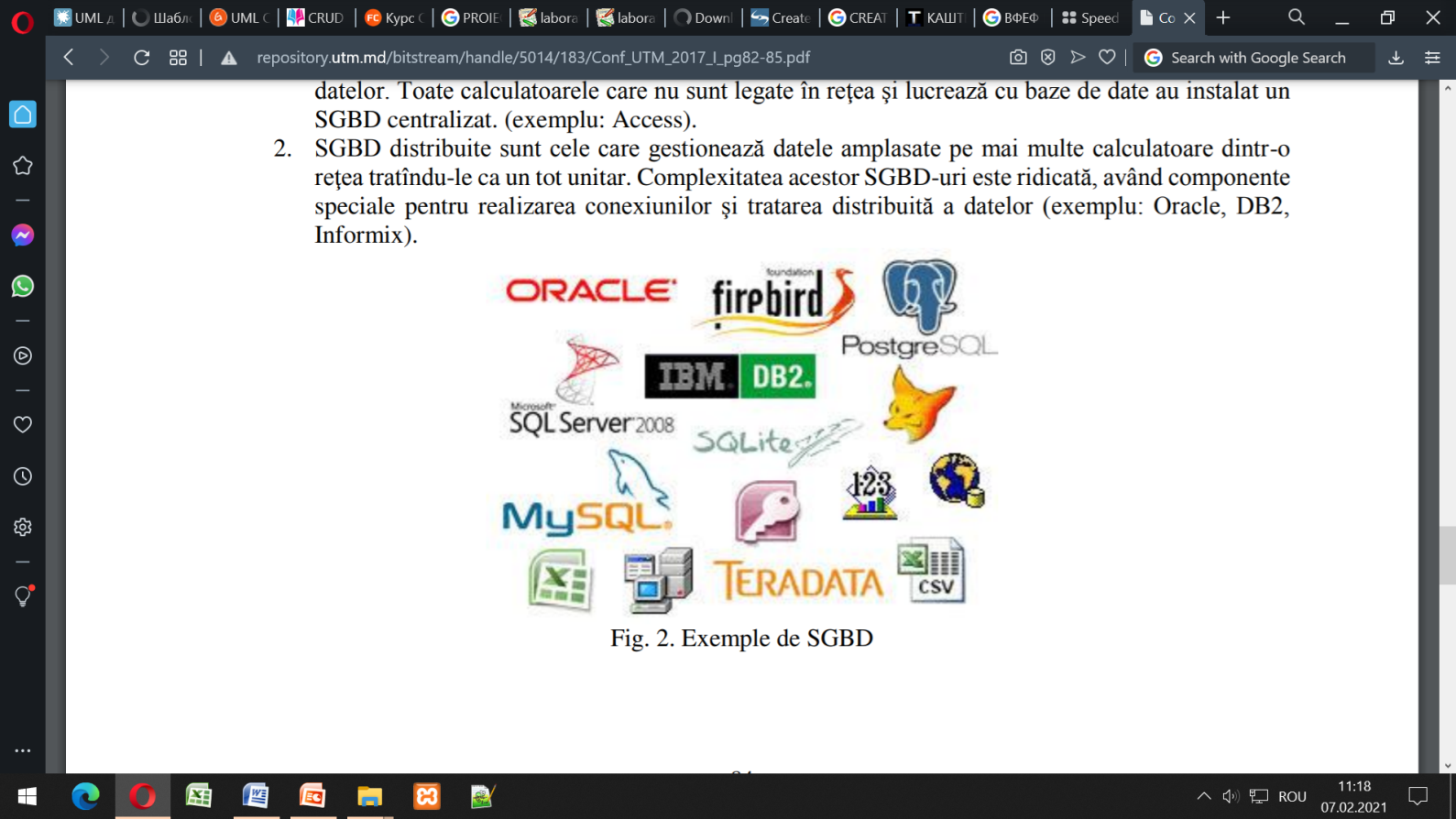
***O BDO este pur și simplu o colecție de date, care fiind procesate ne oferă informații planificate, informație operațională (folositoare, practică pentru procesul decizional curent!!!).***

O **BDO** ***este*** un ***set de liste conexe*** de ***articole similare***. **Datele** ***sunt*** adesea structurate astfel încât să fie ușor de gestionat. De exemplu, o BDO FCIM, DIIS ar avea mai multe tabele ca: profesori, studenți și aule/laboratoare în care fiecare tabel ar avea înregistrări care specifică ***date*** **CURENTE** despre fiecare articol, care fiind procesate, permit să obținem **informatii curente** despre comportamentul Obiectelor din DS **care redau** ***contentul informațional*** sau ***Cerințele de business ale managementului***, adică echipei de administrare care dirijeaza cu procesul current de învățământ.

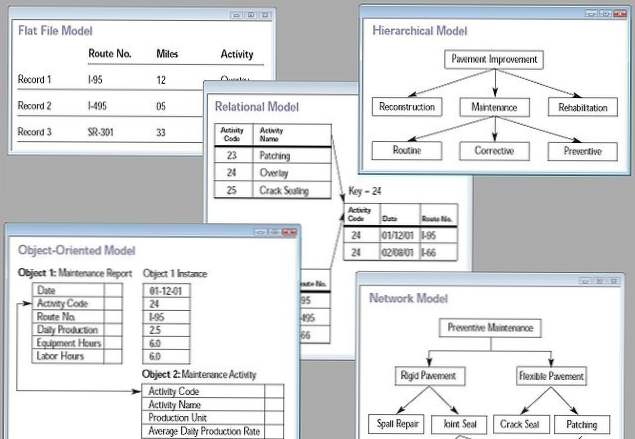
O BDO implică adesea un sistem software numit **S**istem de **G**estionare a **B**azelor de **D**ate (SGBD/DBMS) care este responsabil pentru stocarea și gestionarea datelor din BDO.



Printre (SGBD/DBMS) pot fi: *MySQL, Oracle, Microsoft SQL Server*, Informics, PostgreSQL etc.

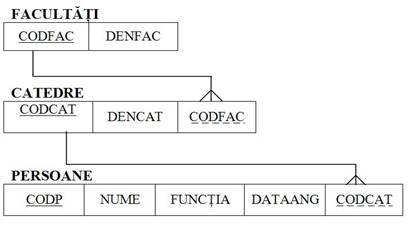


### Termeni cheie ai unei BDO, SGBD/DBMS

**

## Ce este BDO

**O BDO este o colecție de date care sunt stocate în tabele**. Fiecare tabel este alcătuit din rânduri și coloane. Coloanele reprezintă atribute. De exemplu, tabelul studentului din BDO FCIM/DIIS poate avea atribute precum id, nume, oraș, locație, mark1, mark2 etc. **Fiecare rând din baza de date reprezintă o singură entitate.** În acest scenariu, fiecare rând înseamnă un set complet de date ce descriu un singur student. De obicei, menținerea unei baze de date este benefică deoarece organizează seturi de date similar pentru mai multe entități, date care sunt dependente, inteconectate. De exemplu, o BDO FCIM/DIIS poate avea tabele separate pentru a menține detalii despre profesori, studenți, cursuri etc…



**Astfel,** O BDO conține o colecție de date, ce exprimă echivalentul unui model real supus procesului de modelare, pentru a obține o structură posibil de realizat intr-un produs soft, aplicatii Desktop sau Web, cu ajutorul instrumentelor informatice de programare, pentru administrarea datelor ce descriu domeniul real modelat in procesul operational de management.

**BDO ajută la realizarea funcționalităților de bază ale unei organizații.**

Un **S**istem de **G**estionare a **B**azelor de **D**ate (SGBD/DBMS) este software-ul care ajută la gestionarea bazelor de date. Unele (SGBD/DBMS) populare includ MySQL, MSSQL, Oracle și PostgreSQL etc. Utilizatorul poate scrie interogări în Limbajul Structurat de Interogare (SQL) pentru a manipula datele din baza de date.

Astfel, un SGBD/DBMS este un software care permite utilizatorilor să creeze, să manipuleze și să administreze BDO.

**Obiectivele și funcțiile unui SGBD**

Obiectivul general al unui SGBD este de a furniza suportul software complet pentru dezvoltarea de aplicaţii informatice cu baze de date. Fiind un mediu specializat, SGBD satisface cerinţele informaţionale ale utilizatorului într-un mod optim.

**Astfel, el asigură minimizarea costului de prelucrare a datelor, reduce timpul de răspuns (timp util), asigură flexibilitate şi deschidere mare aplicaţiei, asigură protecţie sporită a datelor. Pentru ca un produs software să atingă performanţele de mai sus şi să fie un SGBD, el trebuie să asigure un set minim de obiective/cerinte/particularitati (propuse de Codd):**

- **Asigurarea independenţei datelor.** Se spune că o aplicaţie informatică depinde de date, dacă modificarea structurii de memorare a datelor sau a strategiei de acces la date afectează şi aplicaţia.

- **Redundanţă minimă şi controlată a datelor**. Spre deosebire de sistemele clasice (cu fişiere) de prelucrare automată a datelor, stocarea informaţiilor în bazele de date se face astfel încât datele să nu fie multiplicate. - Facilităţi de utilizare a datelor. Această facilitate presupune ca SGBD-ul să aibă nişte componente specializate pentru diferite operaţii de utilizare.

- **Protecţia datelor**. În sistemele de bază de date, protecţia datelor se asigură sub două aspecte: securitatea şi integritatea.

- **Partajabilitatea datelor.** Partajabilitatea datelor se referă nu numai la aspectul asigurării accesului mai multor utilizatori la aceleaşi date, ci şi la posibilitatea dezvoltării unor aplicaţii fără a se modifica structura bazei de date.

- **Legăturile între date.** Legăturile între date corespund asocierilor care se pot realiza între obiectele unei aplicaţii informatice.

- **Performanţele globale.**

Performanţele globale ale aplicaţiei sunt influenţate de SGBD. Acesta trebuie să gestioneze un volum mare de date de o complexitate ridicată, într-un anumit timp de acces rezonabil pentru diferiţi utilizatori. Funcțiile unui SGBD sunt:

- **Descrierea datelor.** SGBD-ul, prin această funcţie, permite definirea structurii bazei de date cu ajutorul limbajului de definire a datelor (LDD).

- **Funcţia de manipulare a datelor**. Funcţia de manipulare este cea mai complexă şi realizează actualizarea şi regăsirea datelor din baza de date, cu ajutorul limbajului de manipulare a datelor (LMD). - Utilizarea bazei de date. Funcţia de utilizare asigură mulţimea interfeţelor necesare pentru comunicarea tuturor utilizatorilor cu baza de date.

- **Funcţia de administrare.** Funcţia de administrare este complexă şi de competenţa administratorului bazei de date. Administratorul, care are o bogată experienţă de analiză, proiectare şi programare, organizează şi administrează baza de date în toate etapele de realizare a acesteia.

**Procesul de executare a interogărilor în BDO se numește** **OLTP** **sau procesarea online a tranzacțiilor**. Prin urmare, o BDO utilizează OLTP. În general, o bază de date ajută la aranjarea unui set de date pentru o aplicație de administrare a ei cu un instrument SGBD/DBMS **prin intermediul interogărilor, denumit OLTP.**

**OLTP** este prescurtarea pentru **P**rocesarea **T**ranzacțiilor **O**nline, care este de obicei marcată de interogări nu atît de complexe, dar care sunt efectuate într-un volum mai mare. OLTP va procesa un număr mare de tranzacții online, deși scurte, în timp real, tratând astfel de tipuri de interogări precum **Citire, Inserare, Actualizare și Ștergere, adică CRUD.**

**ARHITECTURA OLTP:**

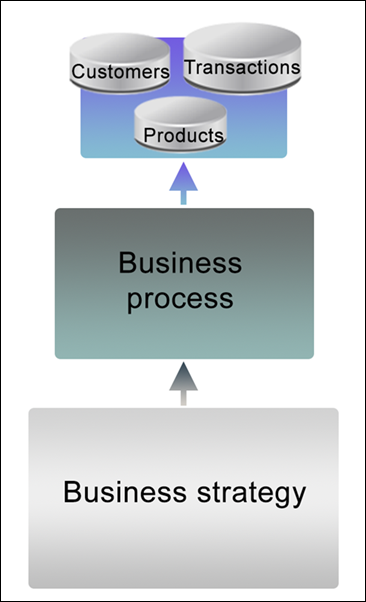
**Sistemul OLTP** se bazează pe o **arhitectură pe** **3 niveluri**.

**Arhitectura pe 3 niveluri** presupune organizarea aplicațiilor în trei niveluri de **calcul logic și fizic**:

1. **Nivelul de prezentare (UI) FRONT-END.**
2. **Nivelul aplicației (*prelucrarea* datelor).**
3. **Nivelul de date (stocarea și gestionarea datelor, operatii CRUD) BACK-END.**

Fiecare dintre cele trei niveluri este independent și are propria infrastructură, dezvoltare și intervale de actualizare și nu afectează celelalte niveluri.

**Următoarea diagramă arată arhitectura OLTP:**



**Unde:**

**Strategie de afaceri**. Conducerea de vârf a unei entități dintr-un DS dezvoltă de obicei ***strategia de afaceri***/*logica de business*. Strategia de afaceri tratează problemele care “afectează” compania în ansamblu.

**Procesul de afaceri.** ***Un set de activități*** și sarcini care realizează un obiectiv al entitații din DS în activitățile ei cotediene.

**Clienți, Tranzacții etc.** ***Baze de date*** care stochează informații despre produse, tranzacții, clienți, furnizori etc.

**Exemple OLTP**

Sistemele de baze de date OLTP sunt utilizate în mod obișnuit pentru diferite **tranzacții online** scurte.

**DEF:** O tranzacție, în contextul unei baze de date, este o unitate logică de program/cod, care este executată independent pentru ***preluarea datelor*** sau ***actualizări***.

**ÎN BAZELE DE DATE RELAȚIONALE**, TRANZACȚIILE CU BAZE DE DATE TREBUIE SĂ FIE **ATOMICE, CONSISTENTE, IZOLATE ȘI DURABILE - REZUMATE CA ACRONIM ACID**.

**OLTP procesează interogări și returnează rezultatele** **aproape în timp real**.

***Exemplele de utilizare a OLTP includ:***

1. Online banking
2. Adăugarea articolelor în coș în magazinele online
3. Rezervarea unui bilet
4. Trimiterea unui mesaj text
5. Întroducerea unei comenzii
6. Bancomat
7. Introducerea rezultatelor unui sondaj
8. Managerii vizualizează și actualizează detaliile clienților
9. Etc.

**AVANTAJE SI DEZAVANTAJE OLTP**

|  |  |
| --- | --- |
| **Avantage** | **Dezavantage** |
| Platformă unică | Dependență mare de personal |
| Consecvența bazei de date | Informații sensibile |
| Restricții de securitate | Risc mare de pierdere a datelor |
| Permite extinderea bazei de clienți | Permite modificări concurente ale datelor |
| Modificări opportune/in orice timp ale tranzacțiilor | Număr limitat de interogări |
| **Suport pentru baze de date mari pentru DWH** |  |
| Atomicitate |  |
| Facilitează manipularea datelor |  |

Scopul principal pentru sistemele OLTP este de a efectua o procesare rapidă a interogărilor, stocarea si asigurarea integrității datelor în medii cu acces multiplu. Eficiența sistemelor OLTP este determinată de numărul de tranzacții pe secundă. Pentru a efectua o serie vastă de tranzacții de succes în sistemul OLTP, aceste tranzacții trebuie să fie atomice, consistente, izolate și durabile (ACID ***have to be atomic, consistent, isolated, and durable***).

**Exemplu de OLTP.** O interogare selectează toate problemele deținute de un utilizator cu ID = 1 cu stările „Rezolvat”, „Publicat”, „Închis”, „Terminat”, le sortează, le grupează pe proiecte și le afișează numărul.

**SELECT** tab2.`key` **AS** project\_key, tab1.`status` **AS** issue\_status, **COUNT**(\*) **AS** t\_count

**FROM** jira\_issues **AS** **tab1**

**LEFT** **JOIN** jira\_\_project **AS** **tab2**

**ON** tab1.project\_id = tab2.id

**WHERE** tab1.user\_assigne = 1

**AND** tab1.`status` **IN** ('Resolved','Published','Closed','Done')

**GROUP** **BY** project\_key, tab1.`status`

**ORDER** **BY** project\_key

**In concluzie:**  OLTP înseamnă Procesarea Tranzacțiilor Online, care este de obicei marcată de interogări mult mai puțin complexe, care sunt efectuate într-un volum mai mare. OLTP va procesa un număr mare de **tranzacții online**, deși scurte, în timp real, tratând astfel de tipuri de interogări precum **Citire, Inserare, Actualizare și Ștergere (CRUD)**

### Ce sunt tranzactiile ?

In sistemele de baze de date tranzactiile sunt un concept important si foarte necesar in anumite cazuri.

**Tranzactiile inseamna suport din partea serverului de baze de date pentru tratarea unei secvente de instructiuni sql ca un grup unitar**.

**Practic, una din consecintele principale este ca aceste instructiuni sql se executa toate cu succes, sau nici una.**  
Un exemplu usor de inteles pentru necesitatea tranzactiilor este cel in care lucram cu baza de date a unei banci, si avem un tabel **conturi** in care operam mutarea unei **sume** dintr-un cont in altul.  
Cele doua query-uri arata astfel:  
**query 1** : UPDATE conturi SET sold = sold - 100 WHERE clientId = 23;  
**query 2** : UPDATE conturi SET sold = sold + 100 WHERE clientId = 499;  
Daca dintr-un motiv query 2 nu este executat (pana de curent, eroare de aplicatie, etc) atunci scade soldul primului client dar suma nu se muta in contul celui de-al doilea client, lucru ce nu corespunde cu realitatea si apare astfel o inconsistenta a bazei de date. **Grupand aceste doua query-uri intr-o tranzactie**, serverul MySQL executa aceste instructiuni dar nu le face permanente, astfel incat daca una din ele esueaza instructiunile pot fi anulate (un fel de undo). **Plus ca, in timp ce o tranzactie se desfasoara, ceilalti clienti ce se conecteaza la serverul MySQL nu citesc datele modificate decat daca tranzactia este incheiata.**

### Tranzactiile in MySQL.

**Tranzactiile au aparut in MySQL incepand cu versiunea 4.0 si sunt suportate de tipul de tabele InnoDB.** In MySQL exista mai multe tipuri de **motoare de stocare** sau "storage engines" cum le numeste echipa MySQL. Cele mai folosite sunt **MyISAM**, tipul implicit pentru tabelele noi create. InnoDB are ca principal avantaj fata de MyISAM suportul pentru tranzactii. **Deci, inainte sa testati sau sa folositi tranzactiile, asigurati-va ca lucrati pe un tabel InnoDB**. In MySQL exista urmatoarele instructiuni principale ce ne ajuta sa lucram cu tranzactii:

* **START TRANSACTION;** sau **BEGIN;**- incepe o tranzactie
* **COMMIT;** - finalizeaza tranzactia curenta
* **ROLLBACK** - daca tranzactia nu s-a finalizat, putem sa anulam instructiunile executate pana atunci din cadrul tranzactiei

**NOTĂ:** MySQL lucreaza in mod implicit in modul AUTOCOMMIT, adica automat fiecare instructiune ce nu face parte dintr-o tranzactie este finalizata (ca si cum am specifica COMMIT; la sfarsitul ei). Pot sa schimb acest comportament cu instructiunea sql: SET AUTOCOMMIT = 0;  
Pentru a vedea la un moment dat ce valoare are variabila AUTOCOMMIT (0 sau 1) pot folosi instructiunea: **SELECT @@autocommit;**

### Folosirea tranzactiilor in MySQL din PHP

Extensiile **mysqli** si **PDO** permit lucrul cu tranzactiile MySQL. Functiile **mysqli** pentru folosirea tranzactiilor sunt:  
  
1. **mysqli\_autocommit** - seteaza modul autocommit pentru conexiunea $link Pseudocod bool ***mysqli\_autocommit ( mysqli $link , bool $mode )***

2. **mysqli\_commit** - echivalent cu instructiunea **COMMIT** - finalizeaza o tranzactie Pseudocod bool

***mysqli\_commit ( mysqli $link )***

3. **mysqli\_rollback** - echivalent cu **ROLLBACK** - anuleaza tranzactia curenta si query-urile executate pana atunci Pseudocodbool

***mysqli\_rollback ( mysqli $link )***

**Exemplu:**

#### PHP

**<?php**

include("conectare.php");

$rollback = 0;

$sql[] = 'DELETE FROM utilizatori WHERE utilizatorId = 1';

$sql[] = 'DELETE FROM mesaje WHERE utilizatorId = 1';

mysqli\_autocommit($link, false);

foreach ($sql as $s) {

$ok = mysqli\_query($link, $s);

if ($ok === FALSE) $rollback = 1; // daca un query returneaza FALSE (a esuat), variabila rollback devine 1

}

if ($rollback == 1) {

mysqli\_rollback($link);

} else {

mysqli\_commit($link);

}

**?>**

#### Nota

**Desi tranzactiile reprezinta un element important al bazelor de date, intr-o aplicatia web medie probabil nu e nevoie in general de tranzactii.** Daca in schimb se lucreaza cu date financiare sau alte date sensibile in care e extrem important sa se pastreze consistenta datelor si baza de date (implicit website-ul) este foarte accesata, si mai mult, exista probabilitatea reala ca mai multi utilizatori sa citeasca/modifice acelasi randuri dintr-un tabel atunci tranzactiile devin necesare. Daca insa folositi tranzactii, trebuie sa lucrati cu tabele InnoDB.

**Si un ultim moment legat de ceea ce realizam current in MODULUL 1 PHP POO +MVC + CI**

**PRIN ABORDAREA DE MAI SUS NOI AM EFECTUAT SEPARATIA BUSINESS-LOGICII DE LOGICA DE PROGRAMARE**

Un site web obișnuit are trei tipuri de funcționalități:

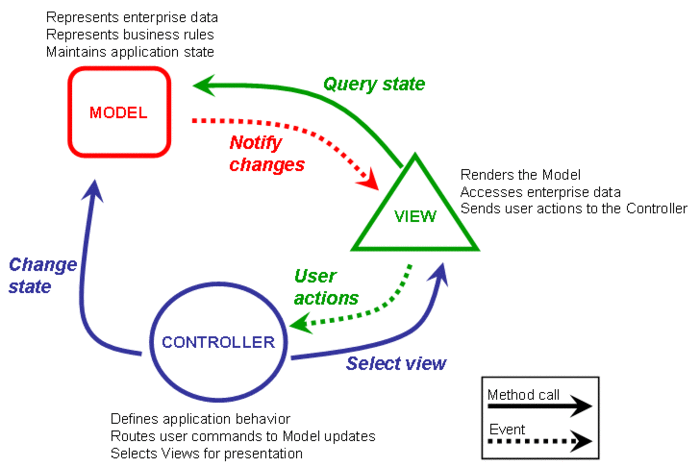
1. **codul de implementare a logicii de afaceri,**
2. **codul de implementare a interacțiunii cu utilizatorul și**
3. **codul de redare a paginii HTML (vizualizări).**

Ceva vreme in urmă, în mediul de programare PHP, ***programatorii obișnuiau să combine aceste trei tipuri de cod într-un singur script PHP destul de maricel***, ceea ce făcea testarea și menținerea unui astfel de cod extrem de dificilă, mai ales dacă scrieți un site mare.

Acum când stim să utilizăm **PHP orientat pe obiecte**, putem împărți codul initial/procedural **în clase**. **Modelul/Paternul** **Model-View-Controller (MVC)** este doar un ***set de sfaturi despre cum să vă organizați aplicațiile într-un mod ușor de lucrat.***

În MVC, clasele care **conțin logica de afaceri** sunt numite **modele**, fragmentele de cod pentru redarea paginilor HTML sunt numite **vizualizăr**i, iar clasele responsabile pentru comunicarea cu utilizatorul sunt numite **controale.**

**Aceasta abordare ne permite sî precăutăm o altă structura a aplicațiilor după cum urmează (este vorba de paternal MVC):**

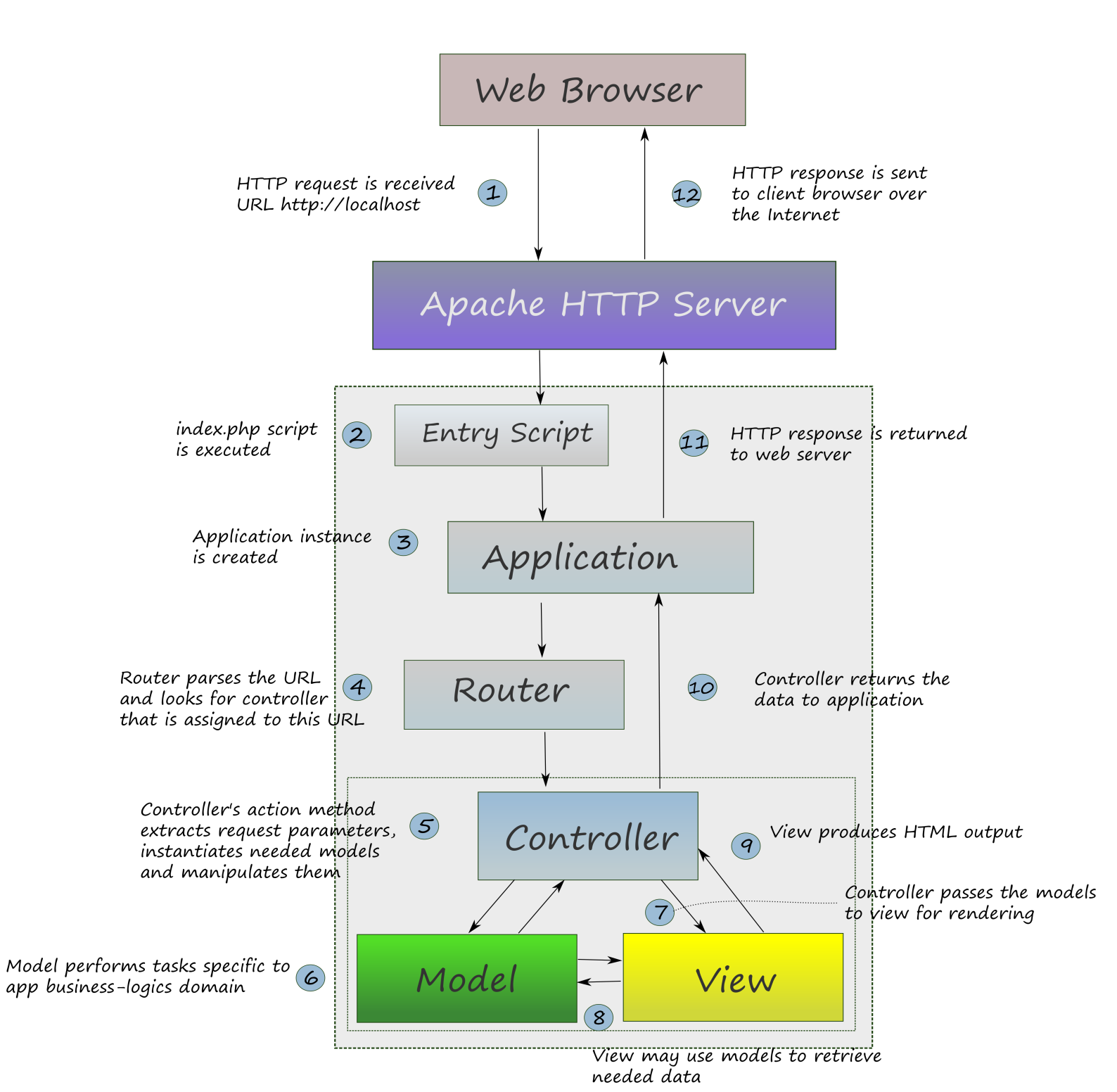
****

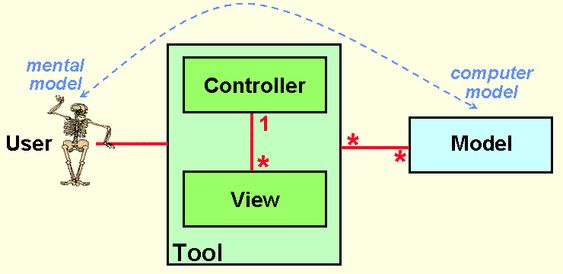
**Sau mai aproape de “ratiunea unui programator”**

**Astfel,** sarcina principală a paternului MVC este de a separa **logica de afaceri (modele)** de **vizualizarea acesteia (vizualizări**). **Aceasta se mai numește și împărțirea responsabilității - fiecare component a codului script PHP este antrenata numai în propriile sarcini.**

Separând **modelele** de **vizualizări**, reducem numărul de dependențe dintre ele. Prin urmare, modificările efectuate într-o legătură vor avea cel mai mic impact asupra altora.

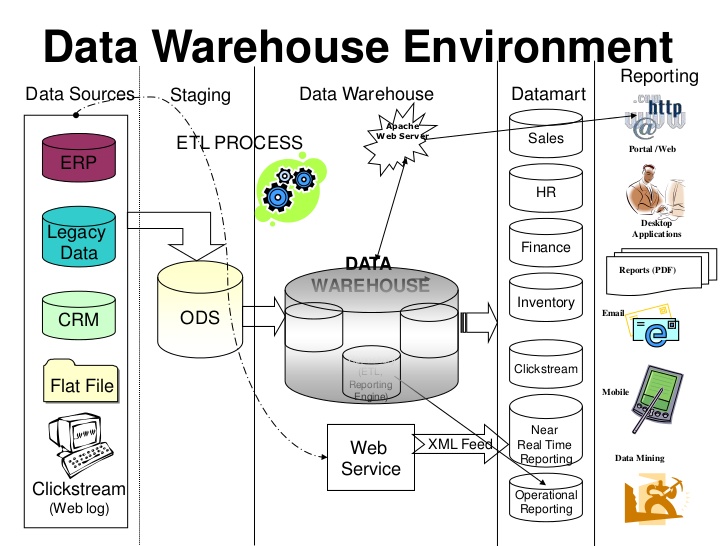
Această separare îmbunătățește, de asemenea, reutilizarea codului. De exemplu, putem crea mai multe reprezentări vizuale pentru aceleași modele. Pentru a înțelege mai bine cum funcționează acest lucru, să ne amintim că orice site web este doar un program PHP care primește o solicitare HTTP de la un server web și trimite un răspuns HTTP. In figura ce urmeaza este prezentata o solicitare HTTP de către o aplicație MVC și cum este generat un răspuns:



**Important: **

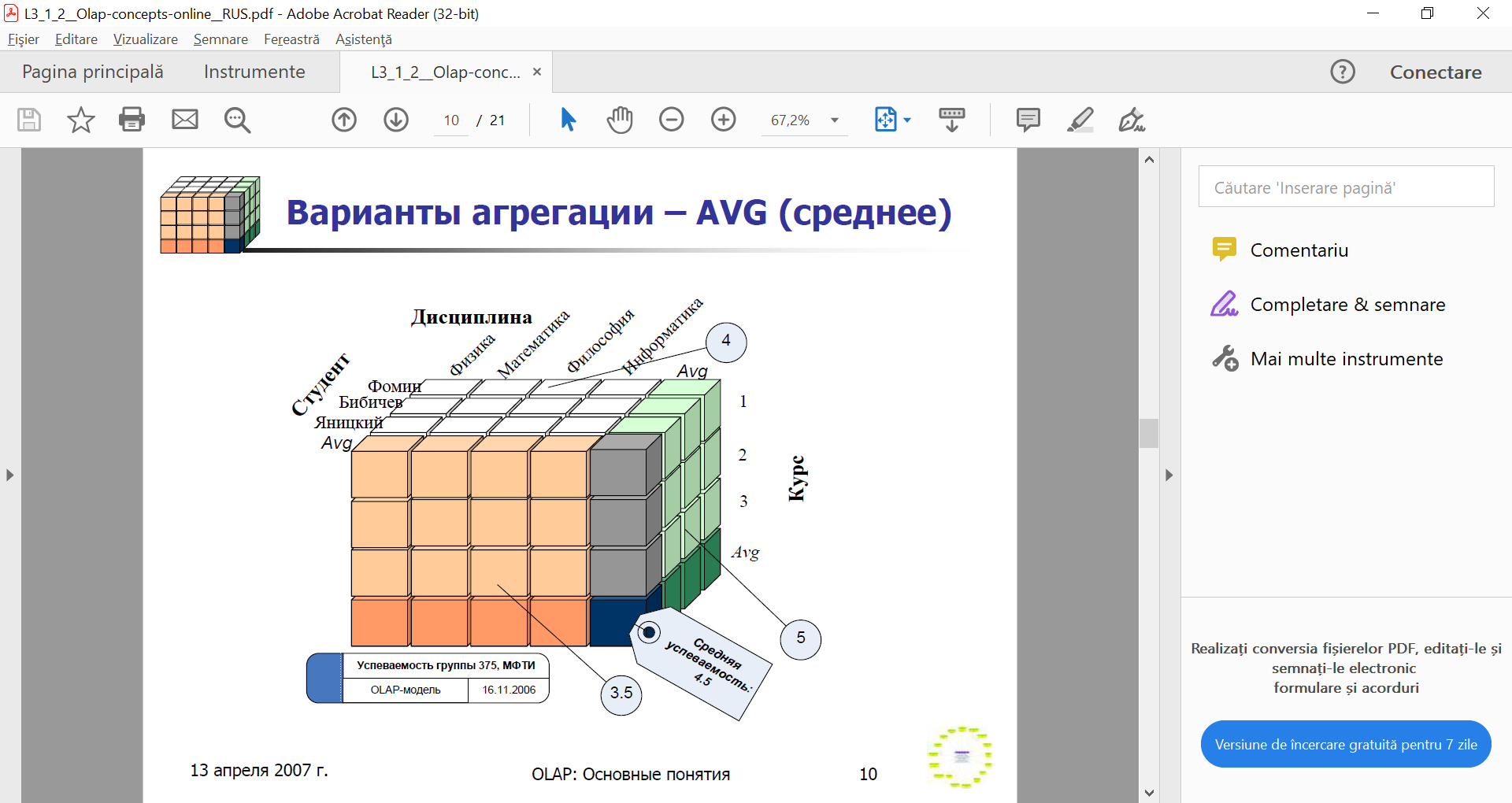
**!!**

**Ce este Depozitul de Date (DD), sau Data WareHouse (DWH)?**

****

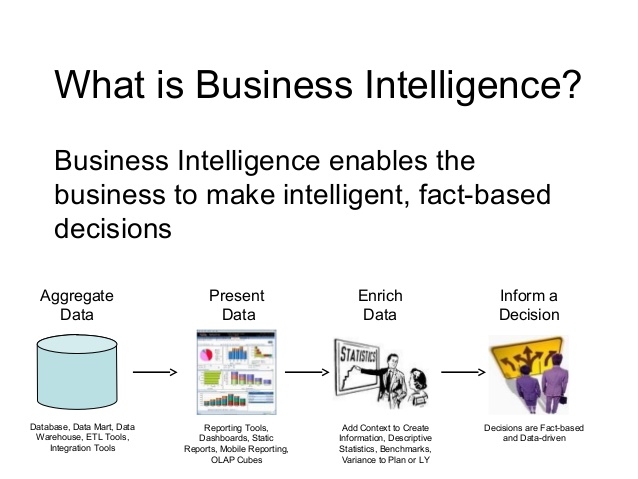
**Ce este Depozitul de Date (DD), sau Data WareHouse (DWH)?**

Un DD(DWH) este ***un tip special de organizare*** si ***un proces special de stocare, integrare/agregare și procesare a datelor,*** care este utilizat pentru ***un proces special de*** ***analiza a datelor***.



În DD ***seturile colectete, stocate și integrate de date au un character istoric***, **NU** current/operațional, și este ***format/colectat*** ***din mai multe sisteme informaționale***, ***realizate pe mai multe platforme operaționale***, iar ***informația procesată*** este oferită ***la unul sau mai multe centre de decizie***, ***vitrine/marts de date***, *cu* ***character de comportament operațional*** *sau* ***strategic.***

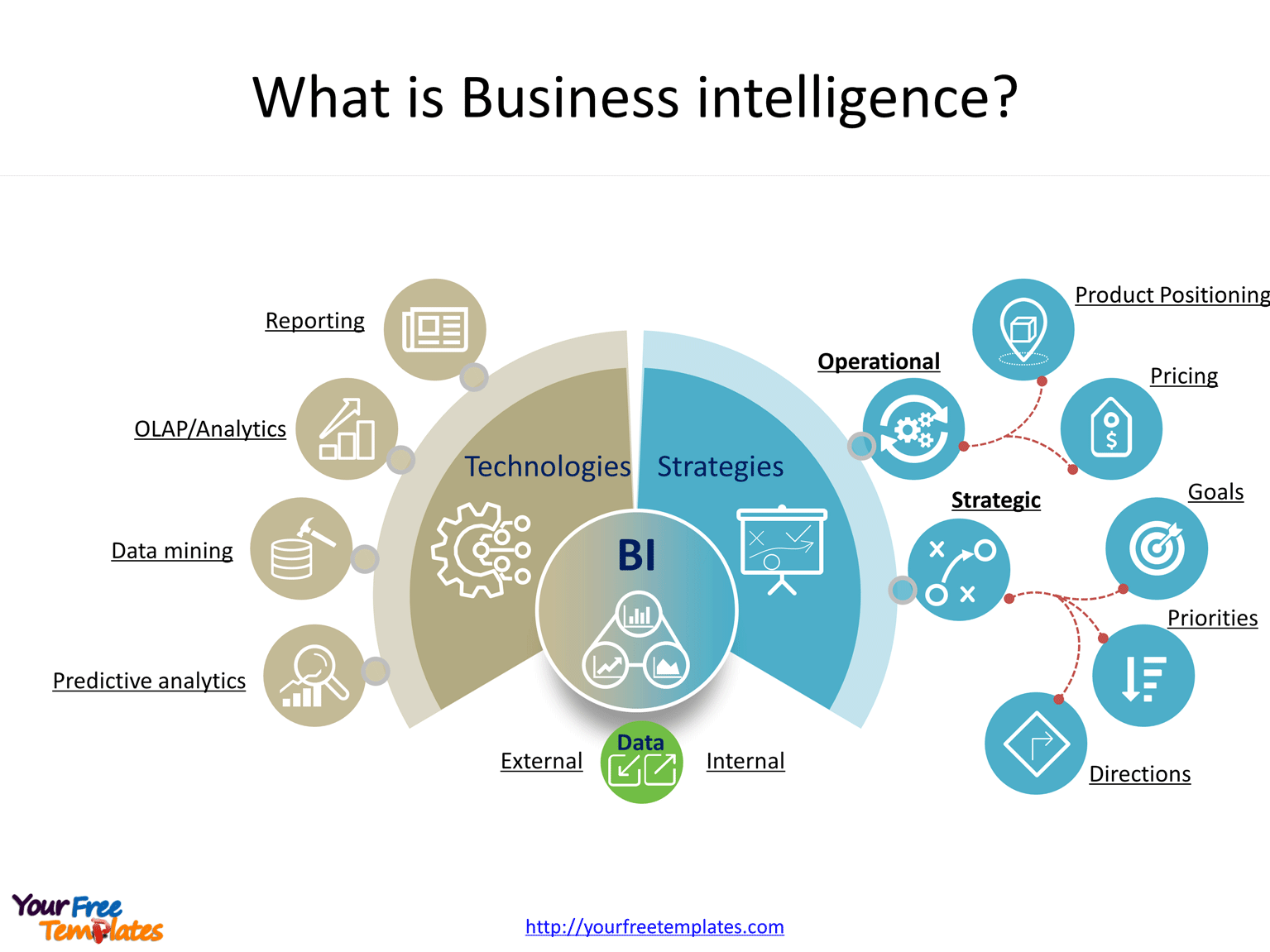
Pe de altă parte, DD(DWH) este ***un sistem de support pentru raportare și analiză*** a datelor; **este componenta principală a business intelligence – BI /Intelegența in afaceri!!/.**



**!**

**1 2 3 4**

DD oferă performanțe ridicate pentru ***interogări analitice***. De obicei, BDO este utilizată de comunitatea managerială în **activități curente** de ***realizare tactică a unor strategii de lucru în procesul decizional***, utilizarea DD se efectuiază ***la nivel de top management***, în **activități de planificare-dezvoltare strategică pentru procesul decizional**.



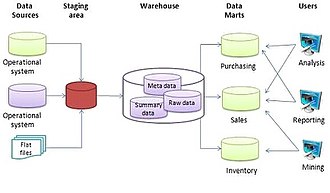
**Business Intelligence ajută la îmbunătățirea afacerii**. **Convertește datele în informații utile pentru analiza afacerilor. De asemenea, ajută la crearea de informații despre întreprinderi pentru utilizatorii finali.**

**!**

DD susține procesul de business intelligence, -BI. O companie poate avea diferite baze de date de la Oracle, MySQL etc. DAR, din aceste baze de date, ***NU este posibilă vizualizarea directă a datelor integrate/aggregate***

Un DD oferă o soluție pentru această problemă. **Este nevoie de *date* pentru a realiza *integrarea/agregarea* și *prelucrarea* *lor*.** Doar aceste date pot fi utilizate pentru vizualizare.

**Un DD sunt orientate pe subiect**, cu date integrate/agregate, variante de timp (difrite orizonturi de timp) și ***nonvolatile (nu dispar, sunt stabile).***

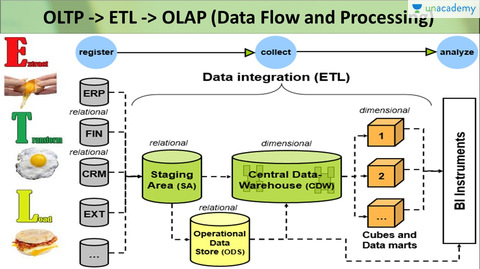


**Procesul de extragere, transformare și încărcare a datelor din mai multe surse de date, inclusive si din diferite BDO în depozit se numește** **ETL /Extract –Transform-Loading/.**

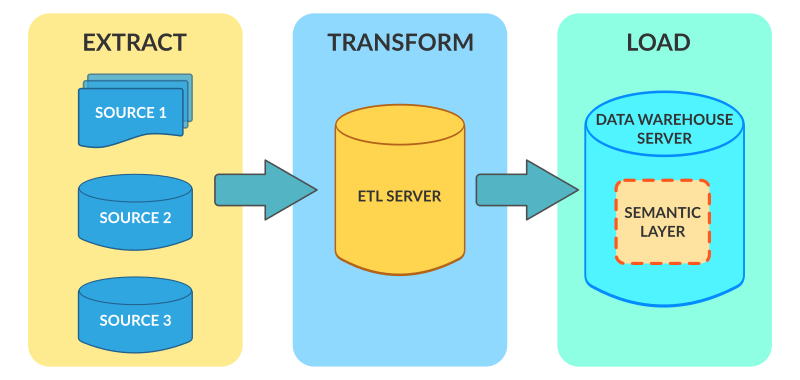
**ERP (Enterprise Resource Planning)** se referă la un sistem complet, care integrează toate funcționalitățile esențiale pentru managementul unei companii. Un ERP este capabil să automatizeze și să computerizeze procesele și informațiile întregii organizații.

<https://www.fluentis.com/ro/enterprise-resource-planning/>

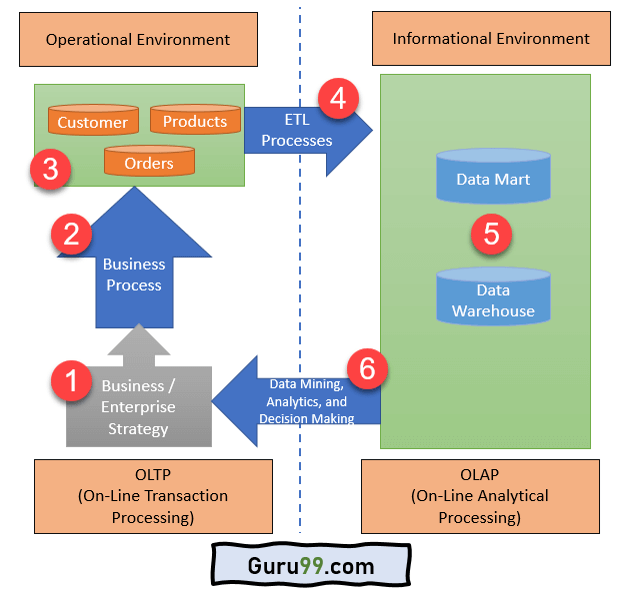
**CRM înseamnă Customer Relationship Management** (în traducere, Managementul relațiilor cu clienții). CRM se referă la un sistem unitar de date și procese de lucru (de obicei un [soft CRM](https://minicrm.ro/)) ce este legat de interacțiunile dintre membrii unei echipe și clienți sau potențiali clienți (lead-uri). <https://www.minicrm.ro/ce-este-un-crm/>



***A staging area*** - place or ***assembly point*** en route to a destination.



**Apoi, Depozitul De Date este utilizată pentru *efectuarea analizei datelor* utilizând strategia OLAP**. Este o metodă de procesare a datelor prin ***interogări analitice online (MS SQL SERVER Datamining, VS Datamining, Power BI, QlikView, Weka, R, Tableau etc.)***

****** În zilele noastre, practic toate organizațiile nu pot decât să accepte **transmiterea și procesarea de date de volume mari si foarte mari**, ca fiind cea mai valoroasă resursă de profit din afacerea lor. Datorită exploatării și gestionării datelor, companiile sunt capabile să aplice analize și **să colecteze informații esențiale pentru previziuni și strategii viitoare**.

Astfel de ***tehnologii precum OLAP și OLTP*** sunt deosebit de utile in cazurile mentionate mai sus.

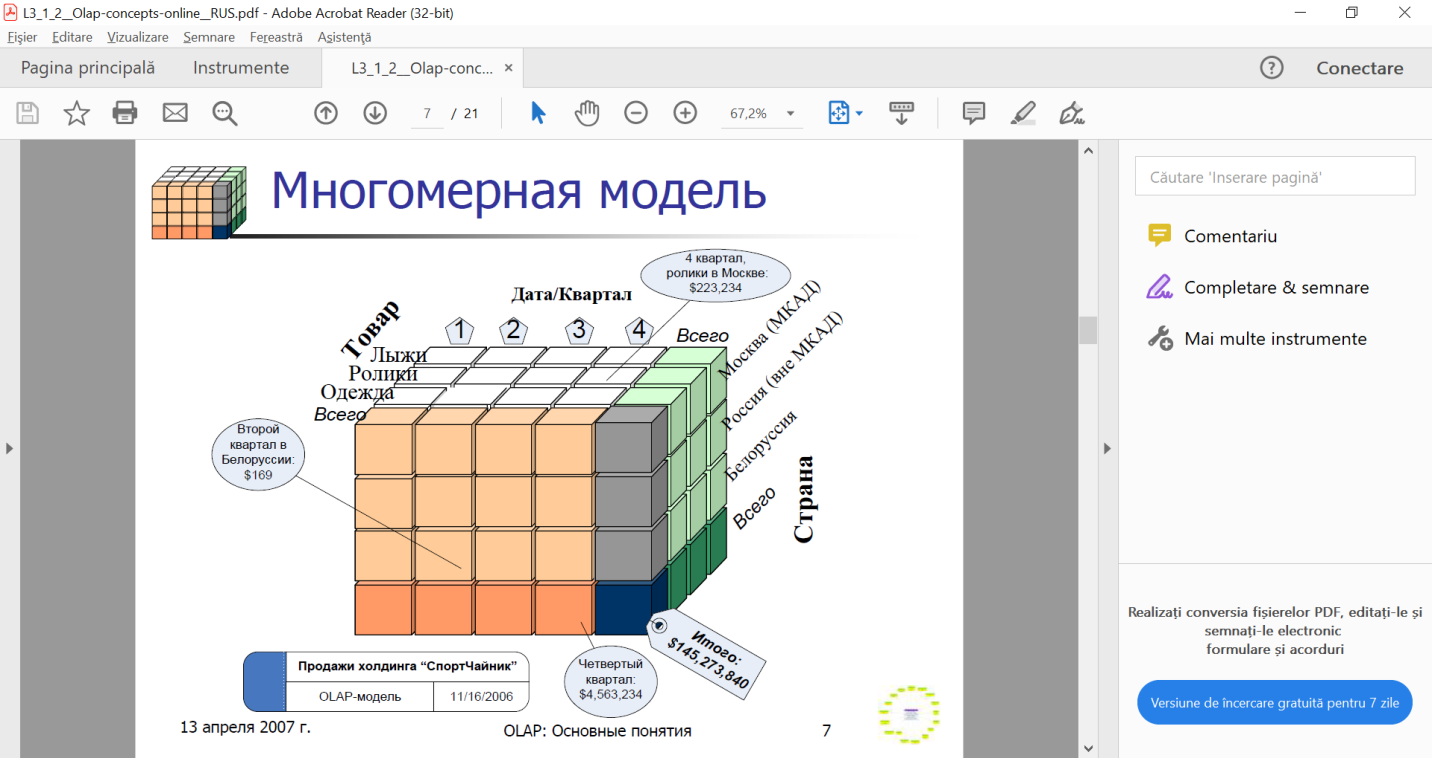
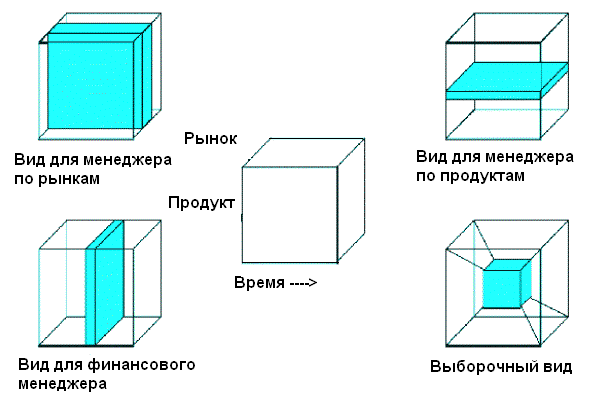
Deci, **OLTP și OLAP** sunt sisteme de procesare online pentru operarea datelor din mediul de afaceri.



**Ce este OLAP?** Pe scurt, este pentru ***Prelucrare Analitică Online*** a datelor. **Dar nu oricum si nu orice**!!!

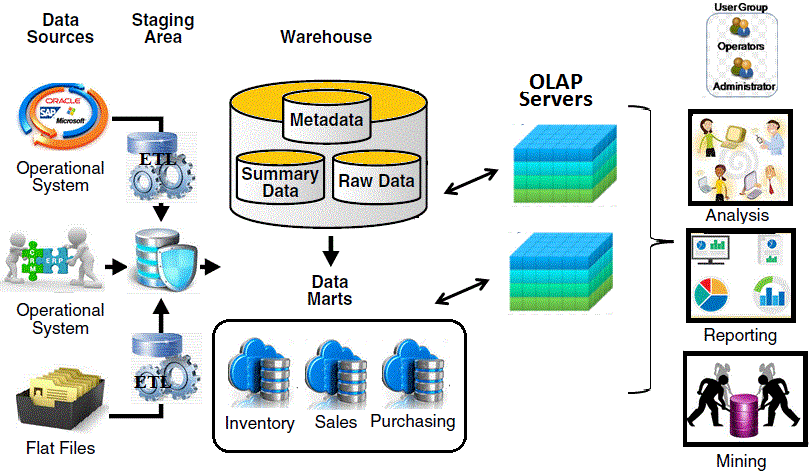
**Această tehnologie analizează o cantitate mare de date păstrate într-un DEPOZIT DE DATE.**

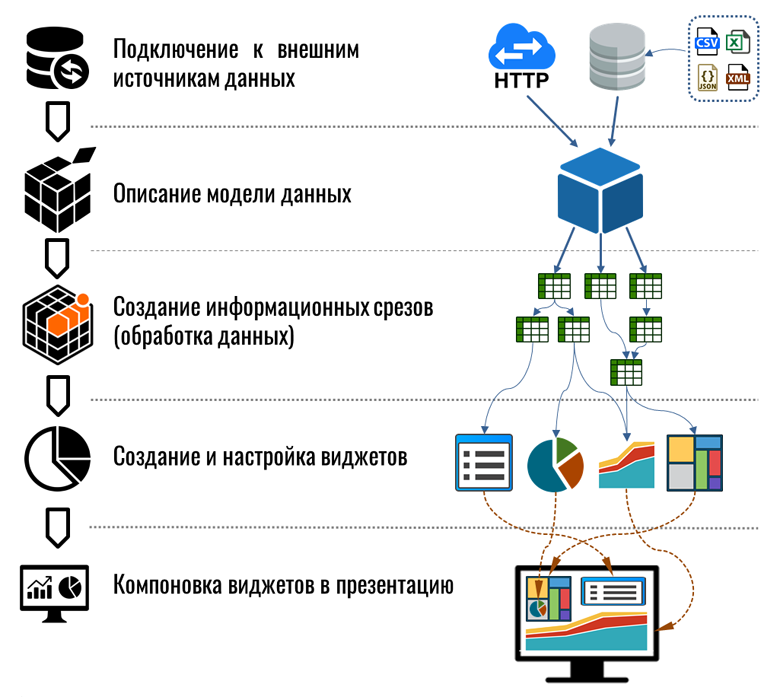
***În mediul OLAP***, toate datele sunt prelucrate prin intermediul unor ***cuburi virtuale de date multidimensionale***, permițându-le utilizatorilor să le detalieze, să le consolideze și să le rotească astfel încât să le poată vedea din diferite perspective și să analizeze tendințele.

**Arhitectura software OLAP include două componente de bază:**

1. ***Server OLAP:*** asigură stocarea datelor, efectuând operațiunile necesare pe acesta și **formarea unui model multidimensional la nivel conceptual.**
2. ***Client de procesare analitică online a datelor multidimensionale:*** prezintă utilizatorului o interfață cu modelul de date multidimensional, oferindu-i posibilitatea de a manipula convenabil datele pentru a efectua sarcini de analiză.





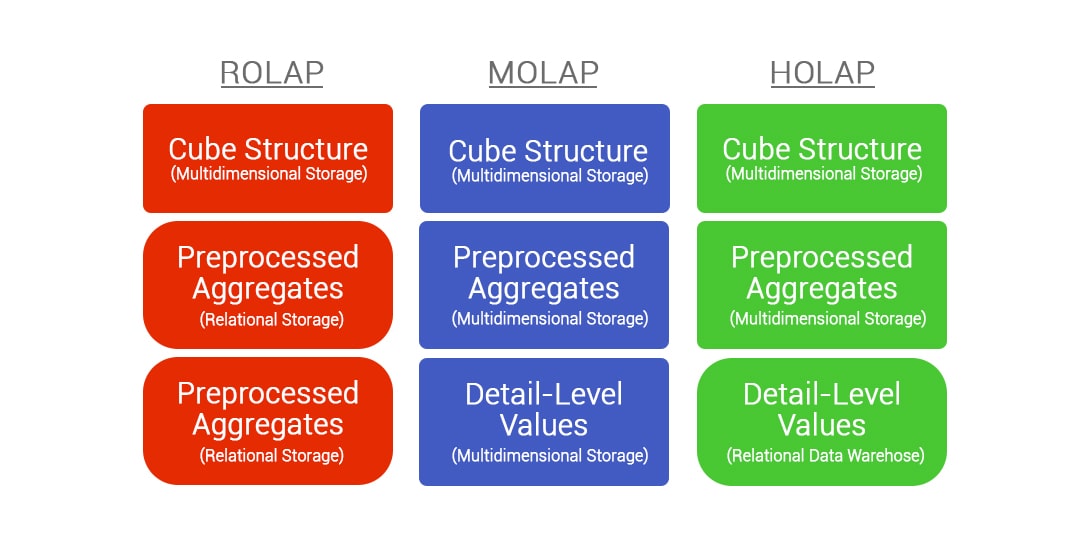
***Procedura de umplere a sistemului OLAP*** din ***sistemele sursă*** are loc utilizând ***ETL (Extract-Transform-Load***). Aceste acțiuni simplifică gestionarea datelor și reduc simultan efortul depus.

***Interogările la DD*** pot fi effectuate prin intermediul fie Tranzact SQL fie prin limbajul MDX - **MultiDimensional EXpressions (MDX) este un limbaj de interogare pentru baze de date OLAP, la fel cum SQL este un limbaj de interogare pentru baze de date relaționale.**

Aceste interogari pot fi relative mici, dar si destul de extinse și au nevoie de mult timp pentru a le realiza.

***Interogările OLAP, utilizând MDX,*** permite agregate și găzduiesc toate datele unui anumit subset. Astfel, utilizatorii sunt capabili să analizeze tendințele, mediile, agregările etc.

***Aplicațiile OLAP sunt exploatate cel mai des prin tehnici de Data Mining.*** Baza de date OLAP permite configurarea datelor după criteriu, după care utilizatorul dorește ca acestea să fie vizualizate. ***OLAP este de obicei realizat prin 3 tipuri:*** **MOLAP, ROLAP și HOLAP**.



**MOLAP**, care este o prelucrare analitică online multidimensională, este un tip standard de OLAP și este adesea menționat ca OLAP în sine. **ROLAP** își ia numele de la „***relațional***”, deoarece operează în mod specific cu baze de date relaționale.

**HOLAP** este un hibrid, iar diferența dintre această bază de date OLAP și baza de date relațională, precum și între multidimensională este că își propune să ***unească atât avantajele MOLAP, cât și ROLAP.***

**Exemplu de OLAP**: Un exemplu simplu de OLAP interogare pentru un utilizator ar fi obtinerea unui tablou care conține date despre **vânzări** în **funcție** de **teritoriu** și **sex (Cub cu 4 dimensiuni)**. ***Interogarea MDX (****SQL este pentru bazele de date "tradiționale" (OLTP). MDX (****M****ulti****D****imensional E****X****pressions) este doar pentru baze de date multidimensionale (OLAP). Problema este că arată foarte asemănătoare, dar....*), ***în acest caz, ar arăta în felul următor***:

**SELECT**

**NON EMPTY VISUALTOTALS(HIERARCHIZE(HIERARCHIZE([Employee].[Gender].Levels(0).Members)), '\* - Total\*\*') DIMENSION PROPERTIES PARENT\_UNIQUE\_NAME, HIERARCHY\_UNIQUE\_NAME, CUSTOM\_ROLLUP, UNARY\_OPERATOR, KEY0,**

**MEMBER\_TYPE ON 0,**

**NON EMPTY**

**VISUALTOTALS(HIERARCHIZE(HIERARCHIZE([Sales Territory].[Sales Territory Country].Levels(0).Members)), '\* - Total\*\*') DIMENSION PROPERTIES PARENT\_UNIQUE\_NAME, HIERARCHY\_UNIQUE\_NAME, CUSTOM\_ROLLUP, UNARY\_OPERATOR, KEY0, MEMBER\_TYPE ON 1**

**FROM**

**Adventure Works]**

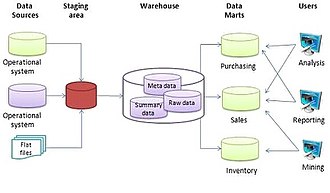
**CELL PROPERTIES BACK\_COLOR, CELL\_ORDINAL, FORE\_COLOR, FONT\_NAME, FONT\_SIZE, FONT\_FLAGS, FORMAT\_STRING, VALUE, FORMATTED\_VALUE, UPDATEABLE, ACTION\_TYPE**

În cele din urmă, datele analizate pot fi încărcate într-un instrument de vizualizare a datelor pentru ***utilizatorii de afaceri***, cum ar fi ***analiștii de date (data analysts)***, ***cercetătorii de date (data scientists) și managerii (BI)***, pentru a lua cunoștințe despre afaceri si a decide.

În plus, datele din DD pot fi împărțite în date asa numite ***vitrine de date/marts***. Ultimile baze de date conțin date pentru ***anumiți utilizatori (segmente specializate de date pentru servicii specific!!!)***.

**De exemplu,** ***departamentul de vânzări*** poate utiliza ***vânzările mart.*** Departamentul de marketing poate folosi ***marketing mart*** și așa mai departe. Aceste baze de date furnizează mai multă securitate și integritate a datelor (sunt cu proprietati mai apropiate de BDO locale).

**!!**!



**!!**!

De obicei, DD se află într-o **locație separată** de BDO. **Un lucru important este faptul că un Depozit de Date NU este un produs pe care o companie o poate achiziționa. Acesta ar trebui să fie proiectat în conformitate cu cerințele companiei.**

# DIFERENȚA DINTRE BDO ȘI DD

Principala diferență dintre BDO și DD este: **o BDO este o colecție organizată de date conexe care stochează datele într-un format tabelar, în timp ce DD este o locație centrală de *date consolidate*, stocate din mai multe surse de date, inclusiv si din BDO.**

**DIFERENȚELE CHEIE**

1. O ***BDO*** ***stochează datele curente*** în timp ce un DD ***stochează date istorice***.
2. O ***BDO*** se schimbă adesea din cauza ***actualizărilor frecvente*** asupra acesteia și, prin urmare, ***NU poate fi folosită pentru analiză sau decizie de lungă durată, adică cu character strategic ci mai degrabă cu character tactic***. Dintr-un DD sunt extrase date ce se raportează pentru a analiza și a lua decizii cu character de ***comportament strategic, de lungă durată***. Astfel, o ***BDO este mai dinamică*** in raport cu operațiile asupra datelor, in comparație cu un ***DD, in care acestea sunt mai mult statice***, decît dinamice.
3. O BDO generală este utilizată pentru ***procesarea tranzacțională*** online (**OLTP**) a datelor, în timp ce un DD este utilizat pentru ***procesarea analitică*** online (**OLAP**).
4. ***Tabelele*** dintr-o ***BDO sunt normalizate*** pentru a obține o stocare eficientă, în timp ce un ***DD este de obicei denormalizat pentru a realiza interogarea mai rapid.***
5. ***Înterogările analitice*** sunt mult mai rapide pe un DD decât pe o ***BDO***.
6. O ***BDO*** ***conține*** ***date extrem de detaliate***, în timp ce un ***DD conține date rezumate***.
7. O ***BDO oferă o vedere relațională detaliată a unui model real, în timp*** ce ***un DD oferă o vedere multidimensională rezumată***.
8. O BDO poate efectua o mulțime de ***tranzacții simultane***, în timp ce ***un DD nu este proiectat pentru astfel de sarcini***.

## Alte diferențe dintre BDO și DD

### Definiție

***Baza de date*** este o colecție organizată de date conexe care stochează datele într-un format tabelar.

***Un depozit de date,*** în schimb, este o locație centrală care stochează date consolidate din mai multe baze de date.

### Tipul de date

***Baza de date*** conține ***date detaliate***, în timp ce ***depozitul de date*** conține date ***sintetizate***.

### Metoda de procesare

***Baza de date*** utilizează procesarea tranzacțiilor online (OLTP), în timp ce ***depozitul de date*** utilizează procesarea analitică online (OLAP).

### Utilizare

***Baza de date*** ajută la efectuarea operațiunilor fundamentale ale unei afaceri, în timp ce ***depozitul de date*** ajută la analizarea afacerii.

### Viteză și precizie

***Depozitul de date*** este mai rapid și mai precis decât ***baza de date*** normală.

### Orientare

***Baza de date*** este orientată spre aplicație. ***Depozitul de date*** este orientat spre subiect. DD clasifică și stochează mai degrabă pe subiect, decât pe aplicație.

### Tabele și conexiuni

Tabelele și intrările dintr-o bază de date sunt complexe deoarece sunt ***normalizate.*** Acest lucru ***minimizează redundanța datelor***. Pe de altă parte, masa și îmbinările sunt simple în depozitul de date deoarece sunt ***de-normalizate.*** Aceasta înseamnă reducerea timpului de răspuns pentru interogările analitice.

### METODA DE PROIECTARE

### Modelarea relațiilor dintre entități contribuie la crearea unei baze de date. Tehnicile de modelare multidimensională a datelor contribuie la crearea unui depozit de date.

### Concluzie

**Diferența dintre BDO și DD** este aceea că baza de date este o colecție organizată de date conexe care **stochează datele într-un format tabel,** în timp ce un depozit de date este o locație centralizată de date, care **stochează date consolidate/integrate/agregate** din mai multe baze de date operationale si nu numai. Pe scurt, baza de date contribuie la ***efectuarea operațiunilor fundamentale ale unei afaceri***, în timp ce depozitul de date ajută ***la analitica afacerii (c***apacitate mintală de a disocia părțile componente ale unui întreg).

