3. Proiectarea depozitelor de date

Cuprins

3. Proiectarea depozitelor de date

Etapele procesului de proiectare a unui depozit de date

- **Etapa 1**
  - Definirea modelului de business
- **Etapa 2**
  - Definirea modelului logic
- **Etapa 3**
  - Definirea modelului dimensional
- **Etapa 4**
  - Definirea modelului fizic

- ❖ Proiectarea depozitului de date este orientată către nevoile utilizatorilor finali.
- ❖ O proiectare bine planificată trebuie să permită dezvoltarea ulterioară a depozitului de date și schimbări în funcție de evoluția nevoilor utilizatorilor.

- Presupunem că o companie a decis să construiască un depozit de date. Pentru aceasta s-au definit cerințele afacerii și s-a stabilit scopul aplicației, apoi a fost creat modelul conceptual. În continuare trebuie să se creeze modelul logic, respectiv modelul fizic al depozitului de date. Adică, trebuie definite:
  - conținutul specific de date;
  - relațiile în interiorul grupurilor de date și între acestea;
  - transformările de date necesare depozitului;
  - frecvența de actualizare a datelor.
- Proiectarea unui depozit de date include două tipuri de proiectări:
  - proiectarea logică;
  - proiectare fizică.

3.1. Proiectarea logică a depozitelor de date

- Proiectarea logică este conceptuală și abstractă. Încă nu se lucrează cu detaliile de implementare fizică, ci doar cu definirea tipurilor de informații necesare depozitului și a relațiilor dintre acestea.

- O tehnică utilizată pentru modelarea informației logice a cerințelor companiilor este modelul entitate-relație. Acest model implică identificarea:
  - obiectelor (entitățile),
  - proprietăților obiectelor (atributele),
  - modul în care obiectele sunt conectate unele de celelalte (relațiile).
- În modelarea dimensională, în loc să descoperim unități atomice de informații (precum entitățile și atributele) și toate relațiile dintre acestea, trebuie să identificăm ce informație aparține:
  - tabela de fapte;
  - tabelelor dimensiune asociate tabelei de fapte.
- Rezultatele proiectării logice a unui depozit de date sunt următoarele:
  - un set de entități și atribute corespunzătoare tabelelor de fapte și tabelelor dimensiune;
  - un model de date operaționale care provin dintr-o anumită sursă de informații și care sunt transferate într-o structură orientată pe subiect, în cadrul depozitului de date vizat.
- Pentru proiectarea logică a depozitului de date se pot folosi:
  - *Oracle Warehouse Builder*;
  - *Oracle Designer*.

3.1.1. Obiectele depozitului de date

- Obiectele principale ale depozitului de date sunt:
  - tabelele de fapte;
  - tabelele dimensiune.

Tabelele de fapte

- Sunt cele mai mari tabele din schema depozitului de date.
- Exemple de tabele de fapte pot fi:
  - *vânzări*,
  - *costuri*,
  - *stocuri,*
  - *profituri* etc.

- O tabelă de fapte are de obicei două tipuri de coloane:
  - coloane care conțin date numerice
    - sunt denumite măsuri sau metrici
  - coloane care sunt chei externe ce referă tabelele dimensiune.

❖ Din punct de vedere al informației stocate, tabela de fapte conține fapte corespunzătoare nivelurilor de bază (cu detalii) sau fapte care au fost agregate.

❖ Din punct de vedere al atributelor, tabela de fapte conține măsuri și chei externe.

- O tabelă de fapte poate conține:
  - date aditive, care pot fi agregate prin simpla lor adunare aritmetică;
    - exemplu: vânzările
  - semi-aditive, care pot fi agregate de-a lungul unor dimensiuni și nu pot fi agregate de-a lungul altora;
    - exemplu: nivelurile de inventariere
  - non-aditive, care nu pot fi adunate.
    - exemplu: mediile

❖ De obicei tabelele de fapte conțin numai date care au același nivel de agregare, adică date aditive.

❖ Din punctul de vedere al modelării, cheia primară a tabelei de fapte este de obicei o cheie compusă care este formată din toate cheile externe.

Tabelele dimensiune

- Conțin date statice.
- Stochează informația care este folosită pentru interogări.
- Sunt de obicei textuale, descriptive și pot fi folosite ca titluri pentru înregistrările din mulțimea de rezultate.
- Exemple de tabele dimensiune pot fi:
  - *clienți* (*cumpărători*),
  - *produse*,
  - *regiuni*,
  - *timp*.

❖ Datele tabelelor dimensiune sunt de cele mai multe ori colectate la cel mai mic nivel al detaliului și apoi sunt agregate la nivele superioare, care sunt necesare pentru analiză.

3.1.2. Schemele depozitului de date

Schema stea

- Proiectarea sub formă de stea este cea mai simplă și mai naturală metodă de proiectare a depozitului de date.
- Se numește astfel deoarece diagrama se aseamănă cu o stea.
  - Centrul stelei constă dintr-una sau mai multe tabele de fapte, iar celelalte noduri reprezintă tabele dimensiune, așa cum sunt reprezentate în figura următoare.

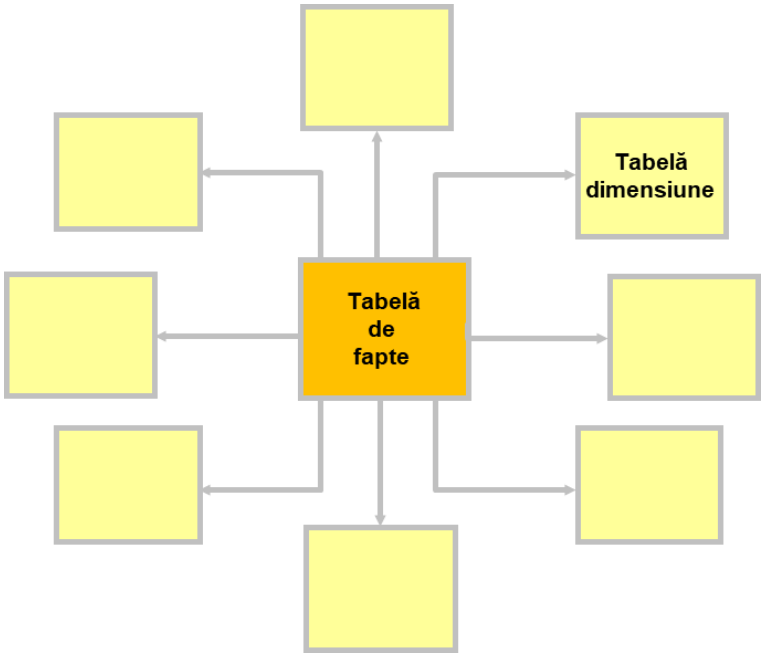

Fig. 3.1. Reprezentarea grafică a schemei stea (*star schema*)

❖ Se poate observa că numai un singur *join* stabilește relațiile dintre tabelul de fapte și oricare alt tabel dimensiune.

❖ O proiectare stea optimizează performanțele prin simplitatea interogărilor și prin furnizarea unui timp de răspuns rapid. Toată informația despre fiecare nivel este reținută într-o singură înregistrare.

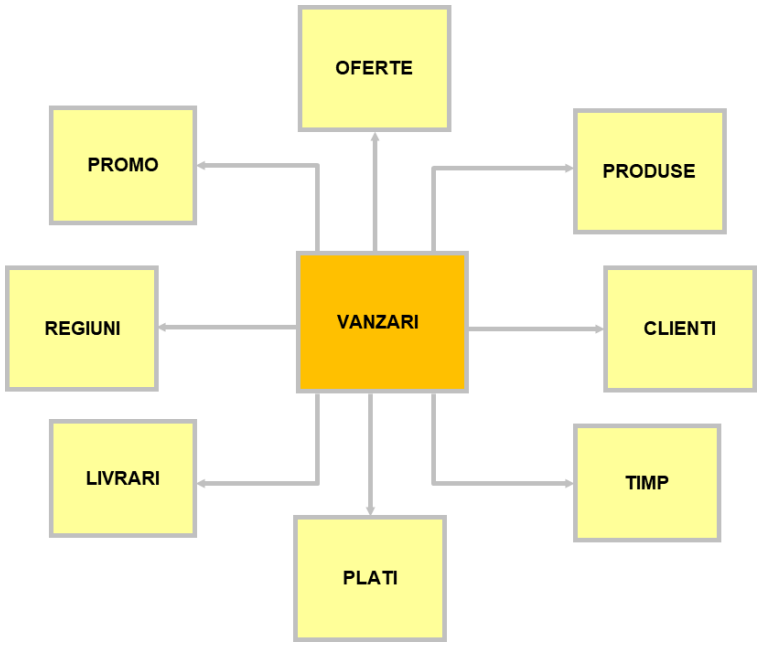

Fig. 3.2. Exemplu de schemă stea

Schema fulg

- Este o schemă stea în care o parte dintre tabelele dimensiune sunt normalizate.
- Organizarea sub formă de schemă fulg determină:
  - redundanță redusă
  - spațiu ocupat redus
- Se numește astfel deoarece reprezentarea grafică seamănă cu un fulg de zăpadă.

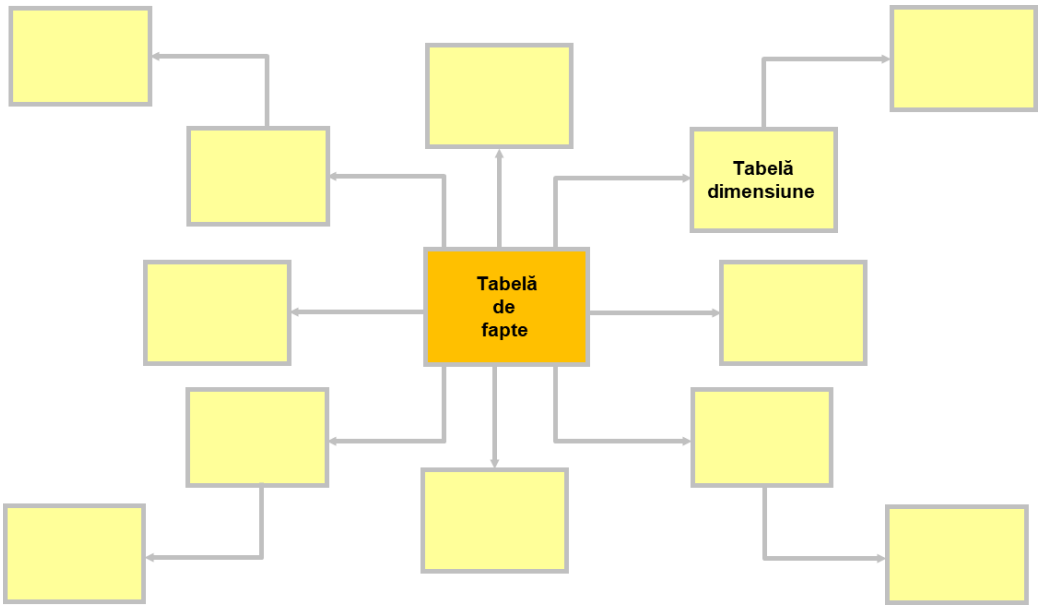

Fig. 3.3. Reprezentarea grafică a schemei fulg (*snowflake schema*)

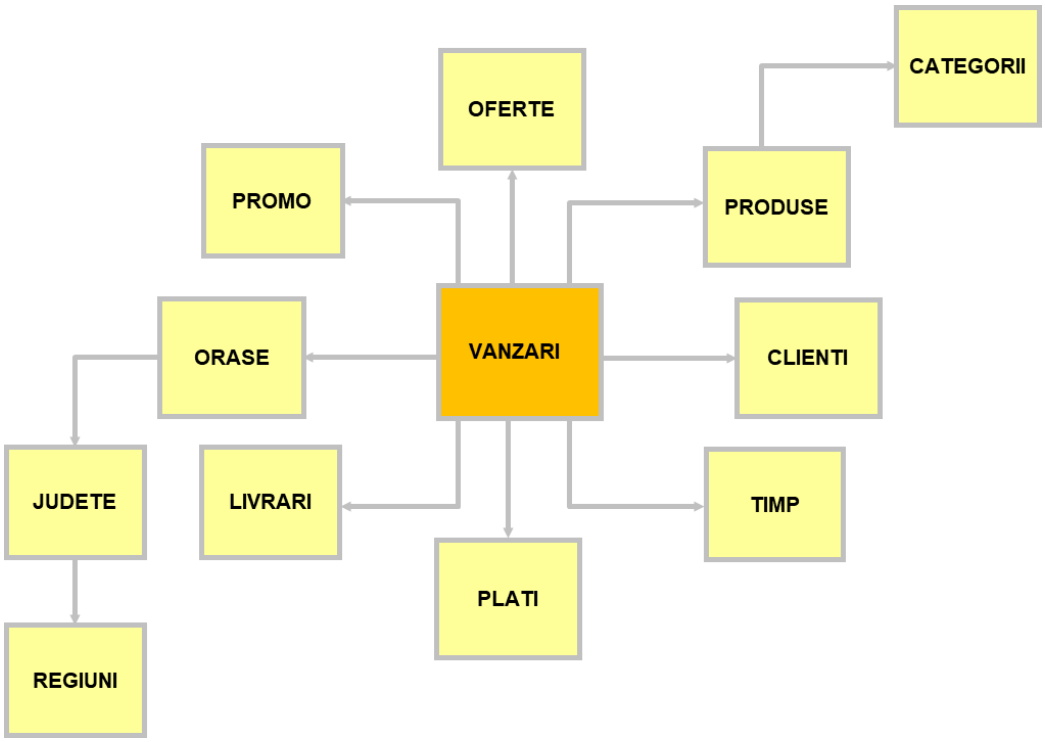

Fig. 3.4. Exemplu de schemă fulg

3.1.3. Tipuri speciale de scheme ale depozitului de date

Scheme stea cu mai multe tabele de fapte

- În unele cazuri, sunt necesare mai multe tabele de fapte.
- Această situație poate să apară deoarece:
  - tabelele de fapte conțin fapte independente;
  - periodicitatea cu care se încarcă noile date în sistem este diferită;
    - de exemplu, anumite fapte se încarcă săptămânal, iar altele lunar
  - pentru a mări performanța;
    - în cazul depozitelor de date foarte mari, tabele de fapte stochează date agregate la niveluri diferite
    - se creează tabele diferite de fapte pentru diferite niveluri de agregare
    - de exemplu, o tabelă de fapte va stoca valoarea totală a vânzărilor zilnice, o altă tabelă de fapte va stoca valoarea totală a vânzărilor săptămânale, iar o alta va stoca valoarea totală a vânzărilor lunare
  - pentru a defini relațiile de tip *many-to-many* dintre anumite tabele dimensiune.

- În figura *Fig. 3.5* tabela de fapte *vanzari* conține informații corespunzătoare vânzărilor realizate în anul curent, iar tabela de fapte *vanzari_istoric* conține informații corespunzătoare vânzărilor realizate anul trecut.

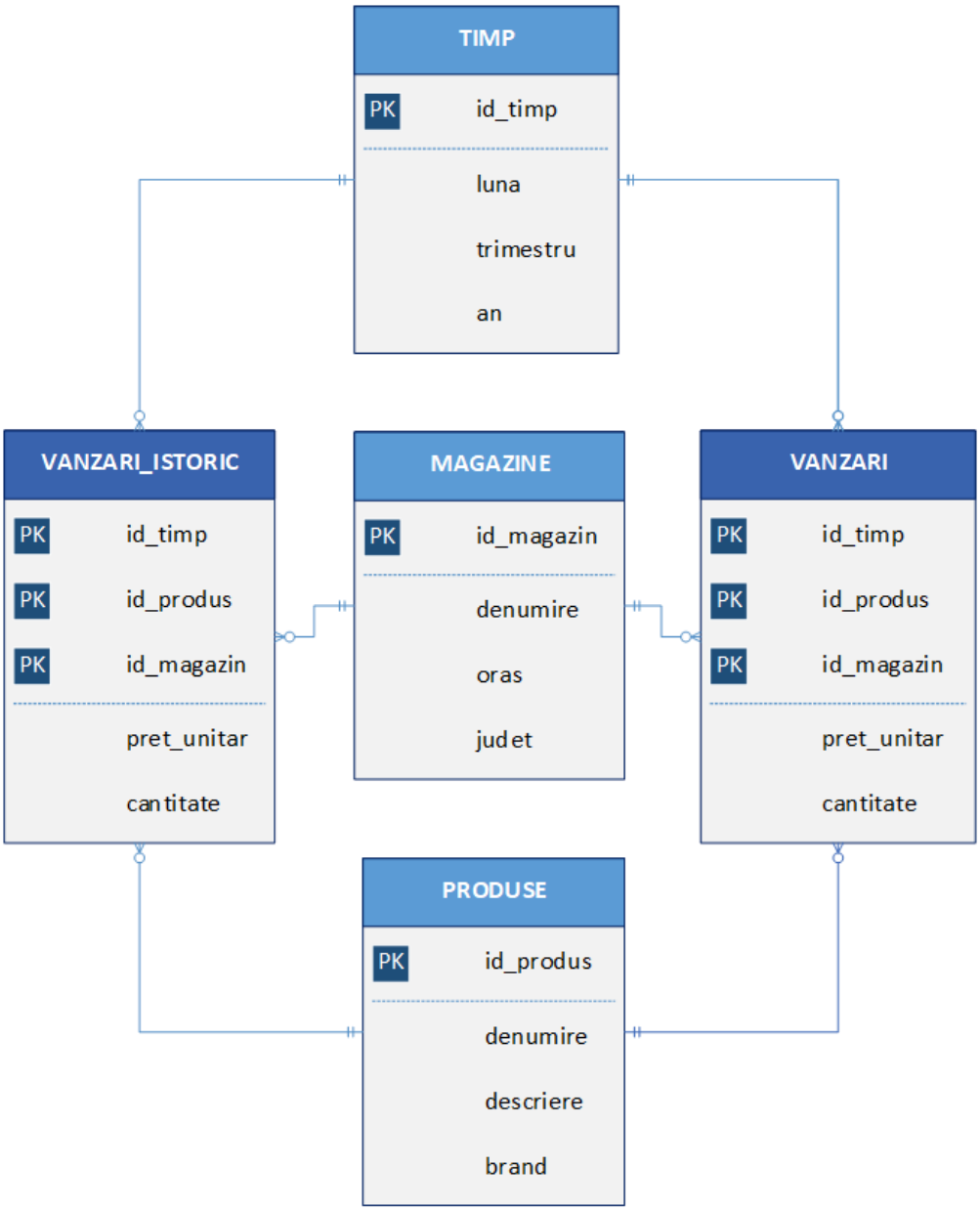

Fig. 3.5. Exemplu de schemă stea cu două tabele de fapte

- În figura *Fig. 3.6* tabela *produse_categorii* este o tabelă asociativă, implementând relația *many-to-many* dintre entitățile *produse* și *categorii.*

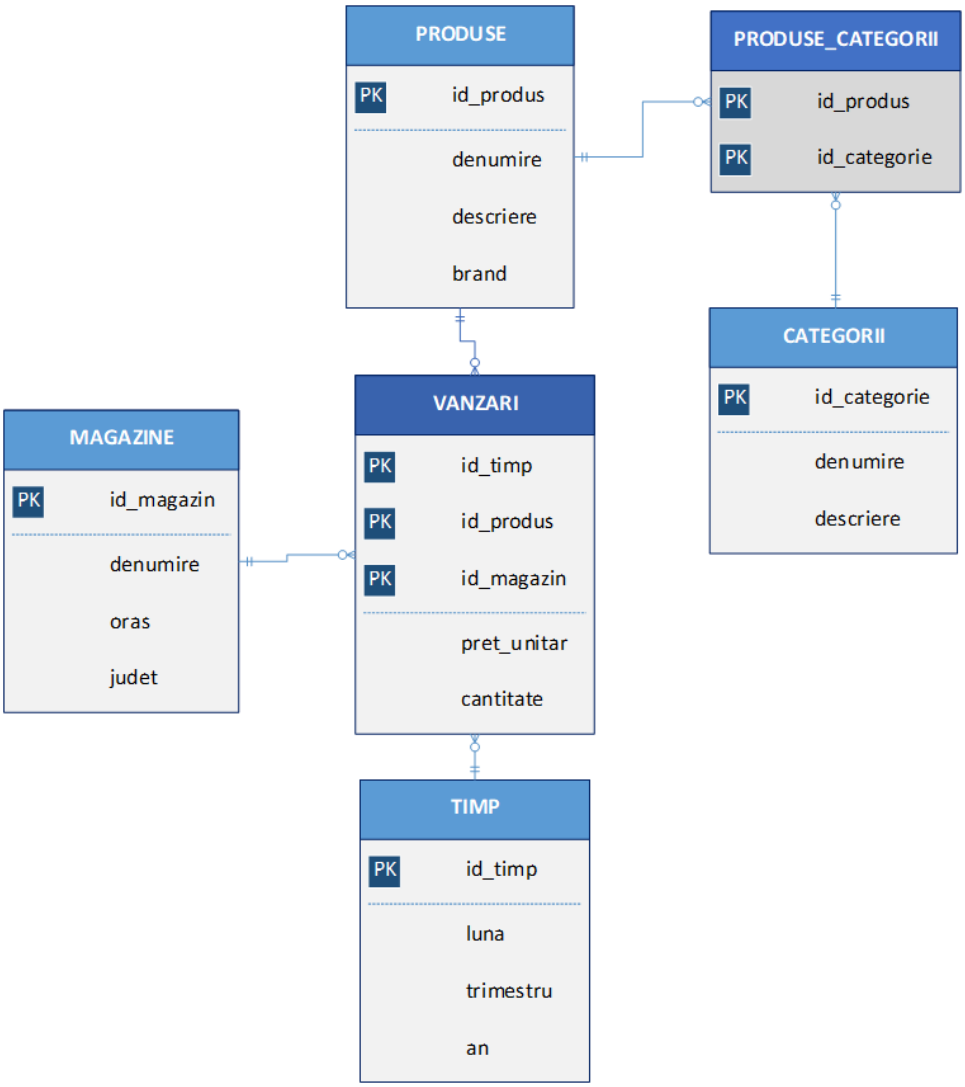

Fig. 3.6. Exemplu de schemă stea cu două tabele de fapte

Scheme stea cu tabele dimensiune secundare

- Există situații în care se dorește menținere anumitor informații într-o tabelă dimensiune, cu scopul de a defini ulterior ierarhii.
- Dacă este necesar să se reducă spațiul ocupat, atunci se va normaliza tabelă dimensiune, iar în urma acestui proces se vor obține mai multe tabele dimensiune secundare.

- Deși normalizarea poate reduce spațial ocupat, aceasta implică și reducerea performanței, respectiv a utilizării avantajelor ce se obțin datorită schemei stea standard.

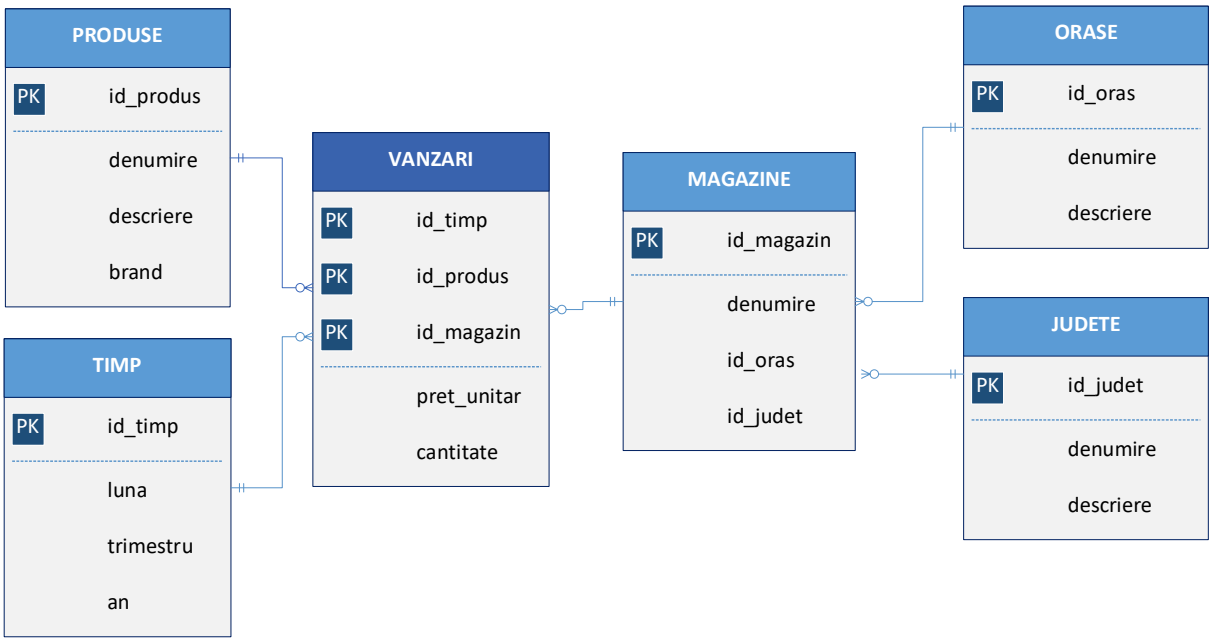

Fig. 3.7. Exemplu de schemă stea cu tabele dimensiune secundare

Scheme multi-stea

- În funcție de modul în care este definită cheia primară a tabelei de fapte se disting două tipuri de schemă stea:
  - schemă stea simplă
    - cheia primară a tabelei de fapte este compusă din cheile externe ce referă tabelele dimensiune
  - schemă multi-stea
    - unicitatea nu este obținută prin compunerea cheilor externe ce referă tabelele dimensiune;
    - cheia primară a tabelei de fapte este compusă din cheile externe ce referă tabelele dimensiune și alte coloane suplimentare.

- În figura *Fig. 3.8* cheia primară a tabelei de fapte *vanzari* este compusă din:
  - cheile externe *id_magazin*, *id_produs* și *id_timp*
  - atributul *nr_factura*
    - face parte din cheia primară, dar nu este cheie externă;
    - permite identificarea unică a unei înregistrări din tabelă.

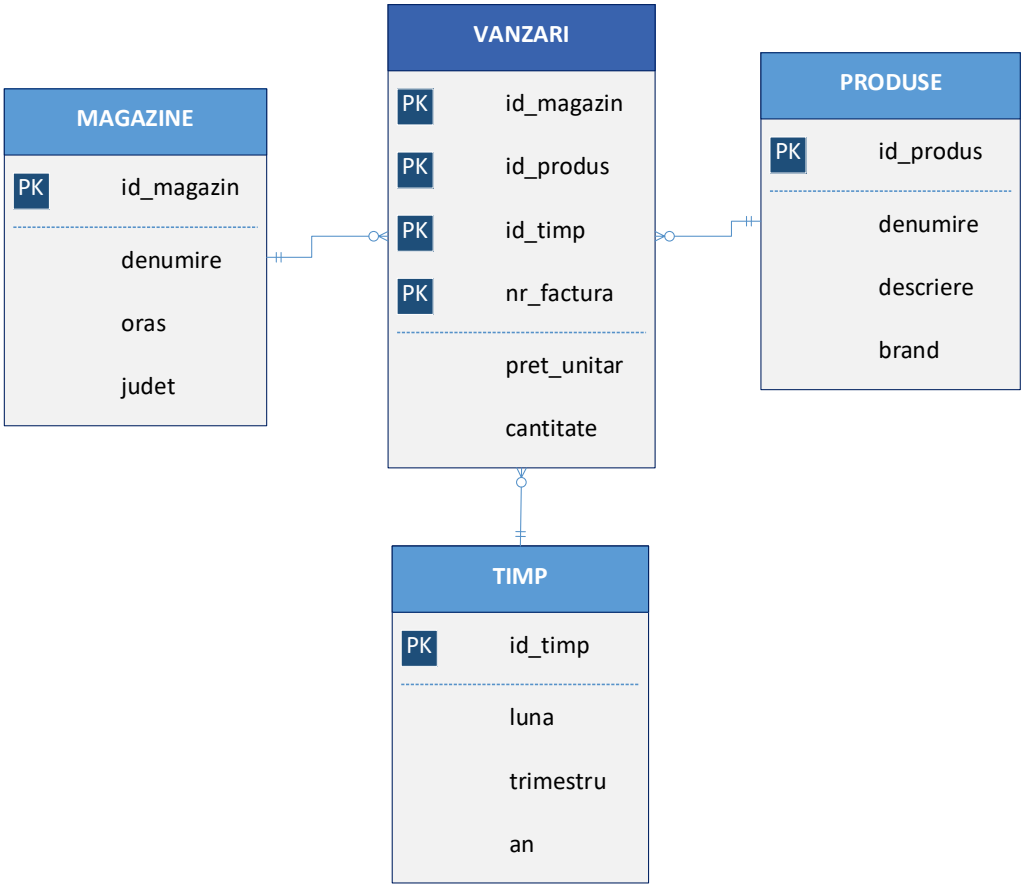

Fig. 3.8. Exemplu de schemă multi-stea

Exemplu de schemă ce combină mai multe concepte

- În practică conceptele enunțate anterior se pot combina într-o singură diagramă de tip stea.
- În figura *Fig. 3.9* este ilustrată o diagramă de tip stea ce conține:
  - tabele de fapte ce folosesc în comun anumite tabele dimensiune;
  - tabele dimensiune ce sunt tabele dimensiune principale pentru anumite tabele de fapte, respectiv tabele dimensiune secundare pentru alte tabele de fapte;
  - normalizare tip fulg.

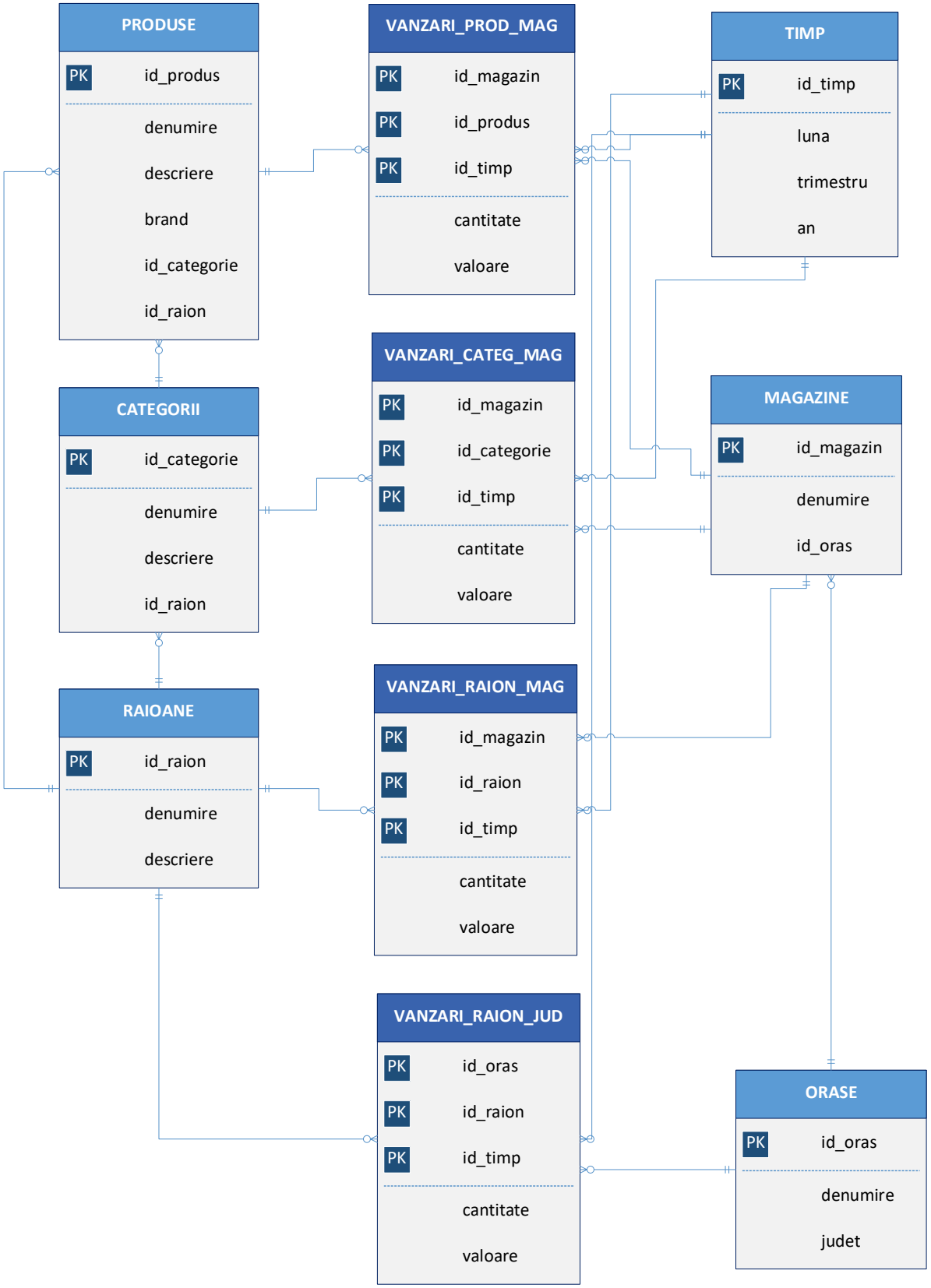

Fig. 3.9. Exemplu de schemă stea ce combină mai multe concepte de modelare

❖ Presupunem că avem organizat un depozit de date al cărui subiect este axat pe vânzările realizate, scopul acestuia fiind determinarea valorii totale a vânzărilor realizate în timp în funcție de anumiți indicatori. În acest scop, tabela de fapte *vanzari* va conține informații despre facturi. În sistemul sursă *OLTP* tabela *facturi* conține atributul *status*, care poate avea una dintre următoarele valori: *validată*, *anulată*, *închisă*.

1. Ce status au facturile încărcate în tabela de fapte?
2. Considerați că este utilă o tabelă dimensiune despre facturi? Justificați printr-un exemplu.
3. Atributul *status* este un atribut care ar trebui să se regăsească în tabela de fapte sau într-o tabelă dimensiune?
4. În cazul concret expus anterior, atributul *status* ar trebui să facă parte din diagramă?

❖ De data aceasta, presupunem că avem organizat un depozit de date al cărui subiect este axat pe vânzările realizate, dar scopul acestuia este determinarea motivelor care implică anularea facturilor validate și efectuarea de statistici referitoare la facturile anulate versus facturile închise, în funcție de anumiți indicatori. Ca și în contextul anterior, tabela de fapte *vanzari* va conține informații despre facturi, iar datele referitoare la acestea vor fi extrase dintr-un sistem sursă *OLTP* în care tabela *facturi* conține atributul *status* ce poate avea una dintre următoarele valori: *validată*, *anulată*, *închisă*.

5. Ce status au facturile încărcate în tabela de fapte?
6. Considerați că este utilă o tabelă dimensiune despre facturi/status facturi?
7. În contextul actual, atributul *status* ar trebui să facă parte din diagramă?

❖ În sistemul sursă *OLTP* tabela *facturi* conține și atributul *tip_factura* care poate avea una dintre următoarele valori: *normală*, *avans*, *rest_plată*, *corecție*, *storno*. În funcție de subiectul de interes pe baza căruia este organizat depozitul de date, în tabela de fapte *vanzari* se vor încărca anumite tipuri de facturi, se pot introduce sau nu date agregate, se va crea sau nu o tabelă dimensiune asociată facturilor.

8. Ce subiect definește depozitul de date și ce tipuri de informații pot fi extrase din acesta dacă în tabela de fapte se încarcă toate tipurile de facturi?

9. Există situații în care ar fi utilă introducerea datelor agregate referitoare la aceeași factura?

10. Este utilă definirea unei tabele dimensiune referitoare la facturi/tipuri facturi?

❖ În exemplele anterioare se poate observa că în funcție de subiect și obiective, depozitul de date este organizat diferit și populat doar cu anumite tipuri de informație.

11. Care este soluția de implementare prin care ne putem asigura că avem la dispoziție toate informațiile necesare pentru a putea crea *data mart*-uri în orice moment de timp (ulterior definirii inițiale a depozitului de date, fără a modifica structura acestuia)?

3.2. Proiectarea fizică a depozitelor de date

- În timpul etapei de proiectare logică se definește un model pentru depozitul bazei de date, care constă din entități, atribute și relații.
- În timpul procesului de proiectare fizică schemele sunt transpuse în structuri ale bazei de date:
  - entitățile în tabele;
  - relațiile în constrângeri de chei externe;
  - atributele în coloane;
  - identificatorii unici în constrângeri de chei primare.

❖ În cadrul etapei corespunzătoare proiectării fizice trebuie să se stabilească cel mai eficient mod de a stoca și de a regăsi obiectele, de a le prelucra din perspectiva realizării copiilor de siguranță și a restaurării (*backup/recovery*).

Bibliografie

1. Connolly T.M., Begg C.E., *Database Systems*: *A Practical Approach to Design, Implementation and Management*, 5th edition, Pearson Education, 2005
2. Dollinger R., Andron L., *Baze de date și gestiunea tranzacțiilor*, Editura Albastră, Cluj-Napoca, 2004
3. Inmon W.H., *Building the Data Warehouse*, 4th Edition, Wiley, 2005
4. Kimball R., *The Data Warehouse Toolkit*, 3rd Edition, Wiley, 2013
5. Kimball R., Ross M., Thornthwaite W., Mundy J., Becker B., *The Data Warehouse Lifecycle Toolkit*, 2nd Edition Wiley, 2008
6. Oracle and/or its affiliates, *Oracle Database Concepts*, 1993, 2024
7. Oracle and/or its affiliates, *Oracle Database Administrator's Guide*, 2001, 2025
8. Oracle and/or its affiliates, *Oracle Database Performance Tuning Guide*, 2013, 2025
9. Oracle and/or its affiliates, *Oracle Database SQL Language Reference*, 1996, 2025
10. Oracle and/or its affiliates, *Oracle Database PL/SQL Language Reference*, 1996, 2025
11. Oracle and/or its affiliates, *Oracle Database: SQL Tuning Workshop*, 2010, 2025
12. Oracle and/or its affiliates, *Oracle OLAP Customizing Analytic Workspace Manager*, 2006, 2019
13. Oracle and/or its affiliates, *Oracle OLAP DML Reference*, 1994, 2019
14. Oracle and/or its affiliates, *Oracle OLAP User's Guide*, 2003, 2019
15. Oracle and/or its affiliates, *Oracle Warehouse Builder Concepts*, 2000, 2021
16. Oracle and/or its affiliates, *Oracle Warehouse Builder Data Modeling, ETL, and Data Quality Guide*, 2000, 2021
17. Oracle and/or its affiliates, *Oracle Database Data Warehousing Guide*, 2001, 2021
18. Oracle University, *Oracle Database*: *PL/SQL Fundamentals*, *Student Guide*, 2009, 2025
19. Poe V., Klauer P., Brobst S., *Building A Data Warehouse for Decision Support*, 2nd Edition, Prentice Hall; 1997
20. Popescu I., Alecu A., Velcescu L., Florea (Mihai) G., *Programare avansată în Oracle9i*, Ed. Tehnică, 2004