6. Obiecte *Dimension*

Cuprins

6.1. Definirea obiectelor *dimension* 2
Specificarea nivelurilor ierarhice 3
Specificarea ierarhiilor 4
Specificarea dependențelor 4
6.2. Obiecte *dimension* cu ierarhii multiple 7
6.3. Obiecte *dimension* cu ierarhii normalizate 8
6.3. Operații asupra obiectelor *dimension* 9
Validarea obiectelor *dimension* 11
Modificarea obiectelor *dimension* 12
Revalidarea obiectelor *dimension* 13
Eliminarea obiectelor *dimension* 13
Bibliografie 14

6. Obiecte *Dimension*

- O dimensiune reprezintă o structură necesară pentru clasificarea datelor cu scopul de a răspunde unor probleme specifice de *business*.
- Informațiile stocate într-un depozit de date sunt în două categorii: fapte și dimensiuni.
- Informațiile dimensionale sunt stocate într-o tabelă dimensiune, iar cu ajutorul obiectului *dimension* datele pot fi organizate și grupate în ierarhii.
  - Acestea determină relații *1*:*n* între atribute sau grupări de atribute (niveluri ierarhice) care nu pot fi reprezentate prin constrângeri de integritate.
  - Trecerea la un nivel superior al ierarhiei se numește *rolling up*, iar trecerea la un nivel inferior al ierarhiei se numește *drilling down*.
  - De obicei, analiza datelor pornește de la cel mai înalt nivel ierarhic și coboară treptat către nivelurile inferioare, atât timp cât se justifică o astfel de analiză.

6.1. Definirea obiectelor *dimension*

- Pentru a putea defini un obiect *dimension*, în baza de date trebuie să existe tabelele cu date dimensionale.
  - De exemplu, pentru crearea unui obiect *dimension* care va defini ierarhii la nivel de zone geografice una sau mai multe tabele trebuie să existe și să furnizeze informații despre orașe, țări, subregiuni, regiuni. Într-o diagramă de tip stea sau fulg aceste informații dimensionale deja există.
- După de ce au fost identificate informațiile dimensionale și definite nivelurile obiectului *dimension*, urmează să se stabilească relațiile ierarhice dintre acestea.
  - De exemplu, nivelul *oras* are ca părinte nivelul *tara*, acesta la rândul său are ca părinte nivelul *subregiune* care are ca părinte nivelul *regiune.* Această informație ierarhică va fi stocată în obiectul *dimension*.
- Există cazuri în care informațiile dimensionale necesare pentru definirea obiectului *dimension* sunt stocate în mai mult de o tabelă (dimensiunea este normalizată sau parțial normalizată).
  - În această situație trebuie să se identifice modul în care aceste tabele sunt conectate. Această legătură trebuie să garanteze că fiecare înregistrare copil este în relație cu o singură înregistrare părinte. Aceasta se poate asigura prin constrângeri de integritate referențială.

- În cazul în care dimensiunile sunt denormalizate, toate informațiile dimensionale se regăsesc în aceeași tabelă dimensiune.
  - În această situație atributele copil trebuie să determine în mod unic atributele părinte.
- Dimensiunile pot fi create utilizând comanda *CREATE DIMENSION*, care are următoarea formă generală:

```
CREATE DIMENSION [schema.]nume_obiect_dimension
  level_clause
  [ level_clause ]...
  { hierarchy_clause
  | attribute_clause
  }
    [ hierarchy_clause
    | attribute_clause
    ]... ;
```

Specificarea nivelurilor ierarhice

- În cadrul comenzii *CREATE DIMENSION* se folosește clauza *LEVEL* pentru a asocia fiecărui nivel ierarhic un nume.

Exemplul 6.1

Utilizând comanda *CREATE DIMENSION* specificați nivelurile ierarhice ale obiectului *dimension* denumit *prod_dim*:

produs → *categorie* → *domeniu* → *subgrupa* → *grupa* → *raion*

Un exemplu de informații ierarhice poate fi următorul:

Tastatura M xt20 (denumirea *produsului*) →
Tastaturi M (denumirea categoriei din care face parte acest produs) →
Tastaturi (denumirea domeniului din care face parte această categorie) →
Componente PC (denumirea subgrupei din care face parte acest domeniu) →
Calculatoare (denumirea grupei din care face parte această subgrupă) →
IT (denumirea raionului din care face parte această grupă)

```
CREATE DIMENSION prod_dim
LEVEL produs      IS (produse.id_produs)
LEVEL categorie   IS (produse.categorie_5)
LEVEL domeniu     IS (produse.categorie_4)
LEVEL subgrupa    IS (produse.categorie_3)
```

```
LEVEL grupa       IS (produse.categorie_2)
LEVEL raion       IS (produse.categorie_1)
...;
```

Fiecare nivel specificat în comandă trebuie să corespundă unui atribut dintr-o tabelă a depozitului de date. De exemplu, în acest caz, nivelul *produs* este identificat de atributul *id_produs* din tabela *produse*, iar nivelul *categorie* corespunde atributului *categorie_5* din aceeași tabelă.

În acest exemplu, tabela *produse* este denormalizată, toate atributele necesare definirii obiectului *dimension* existând în aceeași tabelă, dar aceasta nu este o condiție obligatorie pentru a putea defini obiecte *dimension*.

Specificarea ierarhiilor

- Următorul pas în definirea obiectului *dimension* îl reprezintă declararea relațiilor dintre niveluri folosind clauza *HIERARCHY* a comenzii *CREATE DIMENSION*.
- O relație ierarhică reprezintă o dependență funcțională de la un nivel al structurii la altul. Folosind numele nivelurilor definite anterior, se declară ierarhia dimensiunii. Astfel, fiecărui nivel copil i se asociază un singur nivel părinte și numai unul.
- Ierarhiile permit implementarea dependentelor de tip *1*:*n* între nivelurile ierarhice.

Exemplul 6.2

Prin următoarea clauză se declară ierarhia numită *i_prod* care definește relația dintre nivelurile *produs*, *categorie*, *domeniu*, *subgrupă*, *grupă* și *raion*.

```
HIERARCHY h_prod (
      produs      CHILD OF
      categorie   CHILD OF
      domeniu     CHILD OF
      subgrupa    CHILD OF
      grupa       CHILD OF
      raion) ...
```

Specificarea dependențelor

- În plus de relațiile de tip *1*:*n* ce pot fi specificate prin ierarhii, obiectele *dimension* permit și definirea dependențelor unidirecționale între atribute.
- Clauza *ATTRIBUTE ... DETERMINES* a comenzii *CREATE DIMENSION* permite specificarea atributelor care sunt unic determinate de un nivel ierarhic.

Exemplul 6.3

Prin următoarele clauze se declară dependențe unidirecționale între nivelurile ierarhice și alte atribute ale tabelei dimensiune *produse.*

```
ATTRIBUTE produs DETERMINES
     (produse.denumire, produse.descriere, produse.um)
ATTRIBUTE categorie DETERMINES (produse.categorie5_denumire)
ATTRIBUTE domeniu DETERMINES (produse.categorie4_denumire)
ATTRIBUTE subgrupa DETERMINES (produse.categorie3_denumire)
ATTRIBUTE grupa DETERMINES (produse.categorie2_denumire)
ATTRIBUTE raion DETERMINES (produse.categorie1_denumire)
```

De exemplu, prin clauza *ATTRIBUTE produs DETERMINES* (*produse.denumire*, *produse.descriere*, *produse.um*) se specifică faptul că unei valori a atributului *produs* îi corespunde o singură valoare a atributelor *denumire*, *descriere*, respectiv *um* (unitate de măsură).

De remarcat că este vorba despre o dependență unidirecțională. Există doar certitudinea că pentru o valoare specifică a atributului *produs* (de exemplu, 507) se va găsi exact o singură valoare pentru *um* (bucată). Nu se poate determina o valoare pentru produs care să rezulte din *um*, deoarece există mai multe produse care au aceeași unitate de măsură.

Exemplul 6.4

Comanda completă de creare a obiectului *dimension* denumit *prod_dim* este următoarea:

```
CREATE DIMENSION prod_dim
   LEVEL produs      IS (produse.id_produs)
   LEVEL categorie   IS (produse.categorie_5)
   LEVEL domeniu     IS (produse.categorie_4)
   LEVEL subgrupa    IS (produse.categorie_3)
   LEVEL grupa       IS (produse.categorie_2)
   LEVEL raion       IS (produse.categorie_1)
   HIERARCHY h_prod (
        produs      CHILD OF
        categorie   CHILD OF
        domeniu     CHILD OF
        subgrupa    CHILD OF
        grupa       CHILD OF
        raion)
   ATTRIBUTE produs DETERMINES
        (produse.denumire, produse.descriere, produse.um)
   ATTRIBUTE categorie DETERMINES (produse.categorie5_denumire)
```

```
ATTRIBUTE domeniu DETERMINES (produse.categorie4_denumire)
ATTRIBUTE subgrupa DETERMINES (produse.categorie3_denumire)
ATTRIBUTE grupa DETERMINES (produse.categorie2_denumire)
ATTRIBUTE raion DETERMINES (produse.categorie1_denumire);
```

- Proiectarea, crearea și întreținerea obiectelor *dimension* reprezintă o parte importantă din proiectarea, crearea și administrarea schemei depozitului de date.

- Odată creat un obiect *dimension*, se verifică îndeplinirea următoarelor cerințe:
  - între părinte și copil trebuie să existe o relație *1:n* (un părinte poate avea unul sau mai mulți copii, dar un copil are un singur părinte);
  - între nivelurile ierarhice și atributele specificate trebuie să existe o relație *1:1*;
  - dacă atributele unor niveluri părinte-copil se află în tabele diferite, atunci legătura dintre acestea necesită de asemenea o relație de tip *1:n* (fiecare înregistrare a tabelei copil trebuie să se afle în relație cu o singură înregistrare din tabela părinte; această relație este mai puternică decât utilizarea doar a unei constrângeri de integritate referențiale, deoarece necesită în plus valori *not null* pentru cheia externă);
  - atributele fiecărui nivel ierarhic trebuie să nu accepte valori *null*;
  - ierarhiile unui obiect *dimension* pot să nu aibă legături între ele, dar atributele specificate pentru nivelurile ierarhice nu pot fi asociate cu mai mult unui obiect *dimension*;
  - relațiile dintre nivelurile unui obiect *dimension* nu pot forma cicluri în reprezentarea grafică **(**de exemplu, un nivel ierarhic nu poate fi pus în relație cu el însuși, în mod direct sau indirect).

- Dacă datele stocate în baza de date încalcă relațiile dintre atribute ce sunt implementate printr-un obiect *dimension*, obiectul va fi creat fără erori.
  - Restricțiile nu sunt aplicate în mod asemănător constrângerilor de integritate.
  - Pentru validarea relațiilor implementate prin obiecte *dimension* se poate utiliza procedura *VALIDATE_DIMENSION* a pachetului predefinit *DBMS_DIMENSION.*

6.2. Obiecte *dimension* cu ierarhii multiple

- În definiția unui obiect *dimension* se pot specifica mai multe ierarhii.

Exemplul 6.5

Presupunem că se doresc informații despre vânzările unui anumit produs în funcție de timp. Primul pas îl constituie identificarea tabelei dimensiune care conține informații despre timp. În următoarea etapă se definește un obiect *dimension* în care vor fi specificate nivelurile ierarhice pentru care se va raporta valoarea vânzărilor și ierarhiile corespunzătoare acestora.

De exemplu, pentru raportare ar putea fi necesare următoarele două reprezentări ierarhice:

- ierarhia *an - trimestru – luna - zi*;
- ierarhia *an_fiscal - trimestru_fiscal - luna_fiscala - saptamana_fiscala - zi*.

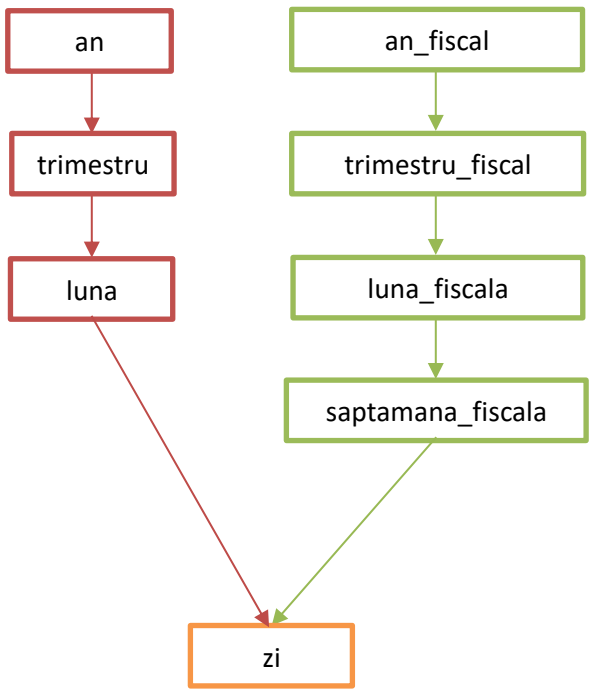

Fig. 6.1 Reprezentarea ierarhiilor obiectului *dimension* denumit *timp_dim*

Pornind de la această reprezentare, se poate construi obiectul *dimension* denumit *timp_dim*, folosind comanda *CREATE DIMENSION*.

```
CREATE DIMENSION timp_dim
LEVEL zi              IS TIMP.data
```

```
LEVEL sapt_fiscala  IS TIMP.saptamana_fiscala
LEVEL luna          IS TIMP.luna
LEVEL luna_fiscala  IS TIMP.luna_fiscala
LEVEL trimestru     IS TIMP.trimestru
LEVEL trim_fiscal   IS TIMP.trimestru_fiscal
LEVEL an            IS TIMP.an
LEVEL an_fiscal     IS TIMP.an_fiscal
HIERARCHY h1_rollup    (
             zi        CHILD OF
             luna      CHILD OF
             trimestru CHILD OF
             an)
HIERARCHY h2_rollup    (
             zi            CHILD OF
             sapt_fiscala  CHILD OF
             luna_fiscala  CHILD OF
             trim_fiscal   CHILD OF
             an_fiscal)
ATTRIBUTE zi DETERMINES
   (TIMP.nume_zi,
    TIMP.nr_zi_in_saptamana,
    TIMP.nr_zi_in_luna,
    TIMP.nr_zi_in_an)
ATTRIBUTE luna DETERMINES (TIMP.luna_nume)
ATTRIBUTE luna_fiscala DETERMINES (TIMP.luna_nume);
```

6.3. Obiecte *dimension* cu ierarhii normalizate

- Se pot crea obiecte *dimension* folosind atribute din mai multe tabele.
- Ierarhiile unui obiect *dimension* pot fi normalizate sau denormalizate.
  - Dacă nivelurile unei ierarhii provin din aceeași tabelă, atunci ierarhia este complet denormalizată.
  - Dacă nivelurile unei ierarhii provin din mai multe tabele, atunci această ierarhie se numește normalizată.
- Clauza *JOIN KEY* a comenzii *CREATE DIMENSION* specifică modul de joncțiune între nivelurile ierarhiei.

Exemplul 6.6

Presupunem că sunt necesare informații despre valoarea vânzărilor raportate la domiciliul clienților.

Datele necesare sunt stocate în tabelele *clienti* și *tari*.

Obiectul *dimension* denumit *clienti_dim* este normalizat, deoarece atributele din definiția sa provin din tabele diferite.

```
CREATE DIMENSION clienti_dim
        LEVEL client     IS (clienti.client_id)
        LEVEL oras       IS (clienti.oras)
        LEVEL judet      IS (clienti.judet)
        LEVEL tara       IS (tari.id_tara)
        LEVEL subregiune IS (tari.subregiune)
        LEVEL regiune    IS (tari.regiune)
        HIERARCHY h_rollup (
           client          CHILD OF
           oras            CHILD OF
           judet           CHILD OF
           tara            CHILD OF
           subregiune      CHILD OF
           regiune
           JOIN KEY (clienti.tara_id)
                     REFERENCES (tari.id_tara)
            );
```

6.3. Operații asupra obiectelor *dimension*

- Obiectele *dimension* pot fi vizualizate folosind fie pachetul *DEMO_DIM*, fie folosind *Oracle Enterprize Manager*.
- Pachetul *DEMO_DIM* se poate crea rulând fișierul *smdim.sql*, aflat în *$ORACLE_HOME/rdbms/demo*.
  - Pentru afișarea obiectelor *dimension* se pot utiliza procedurile *PRINT_DIM* și *PRINT_ALLDIMS*.

Exemplul 6.7

- Procedura *PRINT_DIM* permite afișarea unui obiect *dimension* al cărui nume este specificat ca parametru:

```
EXECUTE DEMO_DIM.PRINT_DIM ('TIMP_DIM');
```

Exemplul 6.8

- Pentru a vizualiza informații despre toate obiectele *dimension* care au fost definite se folosește procedura *PRINT_ALLDIMS*.

```
EXECUTE DEMO_DIM.PRINT_ALLDIMS;
```

Validarea obiectelor *dimension*

- Informația unui obiect *dimension* este doar la nivel declarativ, fără nicio restricție legată de baza de date. Dacă relațiile descrise de un obiect *dimension* sunt incorecte, atunci și rezultatele pot fi incorecte. De aceea, ar trebui ca relațiile specificate prin comanda *CREATE DIMENSION* să fie verificate periodic folosind procedura *VALIDATE_ DIMENSION* din pachetul *DBMS_DIMENSION*.
- Această procedură are 4 parametri:
  - *dimension* (numele proprietarului și numele obiectului *dimension*, în formatul *owner.name*);
  - *incremental* (*TRUE*, dacă se dorește verificarea doar a înregistrărilor noi din tabelele utilizate de obiectul *dimension*);
  - *check_nulls* (*TRUE*, pentru a verifica dacă toate coloanele corespunzătoare nivelurilor ierarhice sunt *not nulll*);
  - *statement_id* (un cod unic de identificare asociat rezultatului).

Exemplul 6.9

```
EXECUTE DBMS_DIMENSION.VALIDATE_DIMENSION
      ('MASTER_DW.TIMP_DIM', FALSE, TRUE, 'st_id')
```

- Dacă procedura *VALIDATE_DIMENSION* generează erori, atunci acestea vor fi plasate într-o tabelă sistem denumită *DIMENSION_EXCEPTIONS*.

Exemplul 6.10

```
SELECT *
FROM   DIMENSION_EXCEPTIONS
WHERE  STATEMENT_ID = 'st_id';

STATEMENT_ID OWNER      TABLE_NAME  DIMENSION_NAME  RELATIONSHIP  BAD_ROWID
------------ ---------  ----------- --------------  ------------  ------------------
st_id        MASTER_DW  TIMP        TIMP_DIM        CHILD OF      AAAAuwAAJAAAARwAAA
```

Exemplul 6.11

O metodă utilă constă în interogarea *rowid*-ului înregistrării invalide, pentru a identifica exact înregistrarea care încalcă constrângerea. În acest exemplu, obiectul *dimension* denumit *timp_dim* utilizează tabela *timp.* Folosind *rowid*-urile, putem identifica exact înregistrările care încalcă restricțiile.

```
SELECT *
FROM   timp
WHERE  ROWID IN (SELECT BAD_ROWID
                 FROM   DIMENSION_EXCEPTIONS
                 WHERE  STATEMENT_ID = 'st_id');
```

Exemplul 6.12

Pentru a șterge rezultatele din tabela sistem *DIMENSION_EXCEPTIONS* se pot utiliza comenzile:

```
DELETE FROM DIMENSION_EXCEPTIONS
WHERE  STATEMENT_ID = 'st_id';
COMMIT;

TRUNCATE TABLE DIMENSION_EXCEPTIONS;
```

Modificarea obiectelor *dimension*

- Folosind comanda *ALTER DIMENSION*, se pot adăuga sau șterge niveluri, ierarhii sau atribute dintr-un obiect *dimension*.
- Dacă se încearcă ștergerea unei componente a unui obiect *dimension* de care depind alte componente ale acesteia, atunci comanda eșuează.
- Pentru a verifica starea unei dimensiuni, se verifică conținutul coloanei *INVALID* a vizualizării *USER_DIMENSIONS* din dicționarul datelor.

Exemplul 6.13

```
ALTER DIMENSION timp_dim
DROP HIERARCHY h2_rollup;

ALTER DIMENSION timp_dim
DROP ATTRIBUTE luna_fiscala;

ALTER DIMENSION timp_dim
DROP LEVEL an_fiscal;

ALTER DIMENSION timp_dim
ADD   LEVEL saptamana IS timp.saptamana;
```

Revalidarea obiectelor *dimension*

- Pentru revalidarea unui obiect *dimension* se folosește opțiunea *COMPILE*:

Exemplul 6.14

```
ALTER DIMENSION timp_dim COMPILE;
```

Eliminarea obiectelor *dimension*

- Un obiect *dimension* se elimină folosind instrucțiunea *DROP DIMENSION*.

Exemplul 6.15

```
DROP DIMENSION timp_dim;
```

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