UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

SPECIALIZAREA INFORMATICĂ

   
 

**DATA WAREHOUSE & BUSINESS INTELLIGENCE**    
**- SCRIPTURI -**

Autor

Tiberius Coman

George Banica

Daniela Alexandra Constantin

Sebastian Alexandru Velciu

Profesor titular

Lect. Dr. GABRIELA MIHAI

 

   
GRUPA 405

ANUL I MASTER, SEMESTRUL I

2.

CREATE OR REPLACE PROCEDURE ADD\_LOCATION(valoare\_tara IN VARCHAR2, valoare\_oras IN VARCHAR2, valoare\_strada IN VARCHAR2, valoare\_site IN VARCHAR2) IS

BEGIN

INSERT INTO locatie(tara, oras, strada, site) VALUES (valoare\_tara, valoare\_oras, valoare\_strada, valoare\_site);

COMMIT;

END;

DROP PROCEDURE ADD\_CONT;

CREATE OR REPLACE PROCEDURE ADD\_CONT(valoare\_tip\_cont IN VARCHAR2, valoare\_nume\_cont IN VARCHAR2,

valoare\_sold IN NUMBER, valoare\_cod\_client IN NUMBER) IS

BEGIN

INSERT INTO cont(tip\_cont, nume\_cont, sold, cod\_client)

VALUES (valoare\_tip\_cont, valoare\_nume\_cont, valoare\_sold, valoare\_cod\_client);

COMMIT;

END;

CREATE OR REPLACE PROCEDURE ADD\_COMERCIANT(valoare\_cod\_client IN NUMBER, valoare\_nume IN VARCHAR2,

valoare\_data\_inscriere IN DATE, valoare\_data\_incetare IN DATE) IS

BEGIN

INSERT INTO comerciant(cod\_client, nume, data\_inscriere, data\_incetare) VALUES (valoare\_cod\_client, valoare\_nume,

valoare\_data\_inscriere, valoare\_data\_incetare);

COMMIT;

END;

CREATE OR REPLACE PROCEDURE ADD\_CLIENT(valoare\_nume IN VARCHAR2, valoare\_tip\_client IN VARCHAR2,

valoare\_data\_inscriere IN DATE, valoare\_data\_incetare IN DATE) IS

BEGIN

INSERT INTO client(nume, tip\_client, data\_inscriere, data\_incetare) VALUES (valoare\_nume, valoare\_tip\_client,

valoare\_data\_inscriere, valoare\_data\_incetare);

COMMIT;

END;

DROP PROCEDURE ADD\_CARD;

CREATE OR REPLACE PROCEDURE ADD\_CARD(valoare\_cod\_cont IN NUMBER, valoare\_tip\_card IN VARCHAR2,

valoare\_data\_emitere IN DATE, valoare\_data\_expirare IN DATE, valoare\_numar\_card IN VARCHAR2) IS

BEGIN

INSERT INTO card(cod\_cont, tip\_card, data\_emitere, data\_expirare, numar\_card) VALUES (valoare\_cod\_cont, valoare\_tip\_card,

valoare\_data\_emitere, valoare\_data\_expirare, valoare\_numar\_card);

COMMIT;

END;

CREATE OR REPLACE PROCEDURE ADD\_CANAL\_PLATA(valoare\_cod\_comerciant IN NUMBER, valoare\_cod\_locatie IN NUMBER,

valoare\_tip\_echipament IN VARCHAR2, valoare\_cod\_cont IN NUMBER, valoare\_data\_inceput IN DATE) IS

BEGIN

INSERT INTO canal\_plata(cod\_comerciant, cod\_locatie, tip\_echipament, cod\_cont, data\_inceput) VALUES (valoare\_cod\_comerciant, valoare\_cod\_locatie,

valoare\_tip\_echipament, valoare\_cod\_cont, valoare\_data\_inceput);

COMMIT;

END;

CREATE OR REPLACE PROCEDURE ADD\_TRANZACTIE(valoare\_suma IN NUMBER, valoare\_cod\_cont\_debitor IN NUMBER,

valoare\_cod\_cont\_creditor IN NUMBER, valoare\_data\_initiere IN DATE, valoare\_data\_procesare IN DATE, valoare\_stare IN VARCHAR2) IS

BEGIN

INSERT INTO tranzactii(suma, cod\_cont\_debitor, cod\_cont\_creditor, data\_initiere, data\_procesare, stare)

VALUES (valoare\_suma, valoare\_cod\_cont\_debitor, valoare\_cod\_cont\_creditor, valoare\_data\_initiere, valoare\_data\_procesare, valoare\_stare);

COMMIT;

END;

5.

--(c1)

ALTER TABLE FACT\_TRANZACTII

ADD CONSTRAINT uni\_tranzactii

UNIQUE(ID\_Client, ID\_Cont, ID\_Comerciant, ID\_Canal\_Plata, ID\_Stare, ID\_Locatie, ID\_Data)

DISABLE VALIDATE;

--(c2)

ALTER TABLE DIM\_CLIENT RENAME CONSTRAINT SYS\_C007748 TO DIM\_CLIENT\_PK;

ALTER TABLE DIM\_CLIENT MODIFY CONSTRAINT DIM\_CLIENT\_PK RELY NOVALIDATE;

ALTER TABLE FACT\_TRANZACTII ADD CONSTRAINT fk\_client

FOREIGN KEY (ID\_Client)

REFERENCES DIM\_CLIENT (ID\_Client)

ENABLE NOVALIDATE;

--(c3)

ALTER TABLE DIM\_DETALII\_PLATA RENAME CONSTRAINT SYS\_C007751 TO DIM\_DETALII\_PLATA\_PK;

ALTER TABLE DIM\_DETALII\_PLATA MODIFY CONSTRAINT DIM\_DETALII\_PLATA\_PK RELY NOVALIDATE;

ALTER TABLE FACT\_TRANZACTII ADD CONSTRAINT fk\_detalii\_plata

FOREIGN KEY (ID\_Cont)

REFERENCES DIM\_DETALII\_PLATA (ID\_Cont)

ENABLE NOVALIDATE;

--(c4)

ALTER TABLE DIM\_COMERCIANT RENAME CONSTRAINT SYS\_C007752 TO DIM\_COMERCIANT\_PK;

ALTER TABLE DIM\_COMERCIANT MODIFY CONSTRAINT DIM\_COMERCIANT\_PK RELY NOVALIDATE;

ALTER TABLE FACT\_TRANZACTII ADD CONSTRAINT fk\_comerciant

FOREIGN KEY (ID\_Comerciant)

REFERENCES DIM\_COMERCIANT (ID\_Comerciant)

ENABLE NOVALIDATE;

--(c5)

ALTER TABLE DIM\_CANAL\_PLATA ADD CONSTRAINT DIM\_CANAL\_PLATA\_PK PRIMARY KEY(ID\_CANAL\_PLATA) RELY NOVALIDATE;

ALTER TABLE FACT\_TRANZACTII ADD CONSTRAINT fk\_canal\_plata

FOREIGN KEY (ID\_Canal\_Plata)

REFERENCES DIM\_CANAL\_PLATA (ID\_Canal\_Plata)

ENABLE NOVALIDATE;

--(c6)

ALTER TABLE DIM\_STARE RENAME CONSTRAINT SYS\_C007744 TO DIM\_STARE\_PK;

ALTER TABLE DIM\_STARE MODIFY CONSTRAINT DIM\_STARE\_PK RELY NOVALIDATE;

ALTER TABLE FACT\_TRANZACTII ADD CONSTRAINT fk\_stare

FOREIGN KEY (ID\_Stare)

REFERENCES DIM\_STARE (ID\_Stare)

ENABLE NOVALIDATE;

--(c7)

ALTER TABLE DIM\_LOCATIE RENAME CONSTRAINT SYS\_C007753 TO DIM\_LOCATIE\_PK;

ALTER TABLE DIM\_LOCATIE MODIFY CONSTRAINT DIM\_LOCATIE\_PK RELY NOVALIDATE;

ALTER TABLE FACT\_TRANZACTII ADD CONSTRAINT fk\_locatie

FOREIGN KEY (ID\_Locatie)

REFERENCES DIM\_LOCATIE (ID\_Locatie)

ENABLE NOVALIDATE;

--(c8)

ALTER TABLE DIM\_CALENDAR ADD CONSTRAINT DIM\_CALENDAR\_PK PRIMARY KEY(ID\_Data) RELY NOVALIDATE;

ALTER TABLE FACT\_TRANZACTII ADD CONSTRAINT fk\_data

FOREIGN KEY (ID\_Data)

REFERENCES DIM\_CALENDAR (ID\_Data)

RELY DISABLE NOVALIDATE;

7.

-- create dimensions

DROP DIMENSION locatie;

CREATE DIMENSION locatie

LEVEL locatie\_id IS (dim\_locatie.id\_locatie)

LEVEL strada IS (dim\_locatie.strada)

LEVEL oras IS (dim\_locatie.oras)

LEVEL tara IS (dim\_locatie.tara)

HIERARCHY ierarhie\_locatie (locatie\_id CHILD OF strada CHILD OF oras CHILD OF tara)

ATTRIBUTE locatie\_info LEVEL locatie\_id DETERMINES

(dim\_locatie.strada, dim\_locatie.oras, dim\_locatie.tara, dim\_locatie.site);

DROP DIMENSION timp;

CREATE DIMENSION timp

LEVEL data\_id IS (dim\_calendar.id\_data)

LEVEL data IS (dim\_calendar.data)

LEVEL anul IS (dim\_calendar.anul)

HIERARCHY ierarhie\_timp (data\_id CHILD OF data CHILD OF anul)

ATTRIBUTE timp\_id\_info LEVEL data\_id DETERMINES

(dim\_calendar.id\_data, dim\_calendar.data, dim\_calendar.ziua, dim\_calendar.luna, dim\_calendar.anul)

ATTRIBUTE timp\_data\_info LEVEL data DETERMINES

(dim\_calendar.id\_data, dim\_calendar.data, dim\_calendar.ziua, dim\_calendar.luna, dim\_calendar.anul);

-- display dimensions

SET SERVEROUTPUT ON FORMAT WRAPPED; --to improve the display of info

EXECUTE DBMS\_DIMENSION.DESCRIBE\_DIMENSION('locatie');

EXECUTE DBMS\_DIMENSION.DESCRIBE\_DIMENSION('timp');

-- stergerea exceptiilor vechi (doar daca este necesar)

DELETE (SELECT \* FROM dimension\_exceptions);

-- validate dimensions

-- used to create dimensions exceptions table

@utldim.sql

EXECUTE DBMS\_DIMENSION.VALIDATE\_DIMENSION ('locatie', FALSE, TRUE, 'validare locatie');

EXECUTE DBMS\_DIMENSION.VALIDATE\_DIMENSION ('timp', FALSE, TRUE, 'validare timp');

-- verificarea exceptiilor

SELECT \* FROM dimension\_exceptions;

-- match-uirea exceptiilor in tabela pentru identificarea randurilor cu probleme

SELECT \* FROM dim\_locatie

WHERE rowid IN (SELECT bad\_rowid

FROM dimension\_exceptions

WHERE statement\_id = 'validare locatie');

SELECT \* FROM dim\_calendar

WHERE rowid IN (SELECT bad\_rowid

FROM dimension\_exceptions

WHERE statement\_id = 'validare timp');