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 505

ANUL II MASTER, SEMESTRUL I

**Exercitiul 1: Crearea bazei de date OLTP și a utilizatorilor**

Crearea bazei de date OLTP

drop table client cascade constraints;

create table client(

id\_client number(10) generated by default on null as identity,

nume varchar2(100),

tip\_client varchar2(50) not null,

data\_inscriere date default sysdate not null,

data\_incetare date,

constraint client\_pk primary key(id\_client),

constraint client\_uk unique (nume),

constraint client\_ck check (tip\_client in ('PF','PJ')));

drop table cont cascade constraints;

create table cont(

id\_cont number(10) generated by default on null as identity,

tip\_cont varchar2(10) not null,

nume\_cont varchar2(20) not null,

sold number(10,2) default 0 not null,

data\_creare date default sysdate not null,

data\_inchidere date,

cod\_client number(10) not null,

constraint cont\_pk primary key(id\_cont),

constraint cont\_uk1 unique (cod\_client, tip\_cont),

constraint cont\_uk2 unique (nume\_cont),

constraint cont\_ck check (tip\_cont in ('CURENT', 'ECONOMII', 'IMPRUMUT')),

constraint cnt\_fk\_clnt foreign key (cod\_client) references client(id\_client));

drop table card cascade constraints;

create table card(

id\_card number(10) generated by default on null as identity,

cod\_cont number(10) not null,

tip\_card varchar2(10) not null,

data\_emitere date default sysdate not null,

data\_expirare date,

numar\_card varchar2(30),

constraint card\_pk primary key(id\_card),

constraint card\_uk unique(numar\_card),

constraint card\_ck check (tip\_card in ('DEBIT', 'CREDIT')),

constraint card\_fk\_cont foreign key (cod\_cont) references cont(id\_cont));

drop table tranzactii cascade constraints;

create table tranzactii(

id\_tranzactie number(10) generated by default on null as identity,

suma number(10,2) not null,

cod\_cont\_debitor number(10) not null,

cod\_cont\_creditor number(10) not null,

data\_initiere date not null,

data\_procesare date,

stare varchar2(50) not null,

constraint trn\_pk primary key(id\_tranzactie),

constraint trn\_ck check (stare in ('Initiat', 'Procesat', 'Eroare')),

constraint trn\_fk\_cont\_db foreign key (cod\_cont\_debitor) references cont(id\_cont),

constraint trn\_fk\_cont\_cr foreign key (cod\_cont\_creditor) references cont(id\_cont));

drop table comerciant cascade constraints;

create table comerciant(

id\_comerciant number(10) generated by default on null as identity,

cod\_client number(10),

nume varchar2(100) not null,

data\_inscriere date not null,

data\_incetare date,

constraint com\_pk primary key(id\_comerciant),

constraint com\_fk\_clnt foreign key (cod\_client) references client(id\_client)

);

drop table locatie cascade constraints;

create table locatie (

id\_locatie number(10) generated by default on null as identity,

strada varchar2(100),

oras varchar2(100),

tara varchar2(100),

site varchar2(100),

constraint loc\_pk primary key(id\_locatie));

drop table canal\_plata cascade constraints;

create table canal\_plata(

cod\_comerciant number(10) not null,

cod\_locatie number(10) not null,

tip\_echipament varchar2(10) not null,

cod\_cont number(10) not null,

data\_inceput date not null,

data\_incetare date,

constraint cnl\_plt\_uk unique(cod\_comerciant, cod\_cont),

constraint cnl\_plt\_ck check (tip\_echipament in ('POS', 'ONLINE')),

constraint cnl\_plt\_fk\_comrt foreign key (cod\_comerciant) references comerciant(id\_comerciant),

constraint cnl\_plt\_fk\_cont foreign key (cod\_cont) references cont(id\_cont),

constraint cnl\_plt\_fk\_loc foreign key (cod\_locatie) references locatie(id\_locatie));

insert into client(nume, tip\_client, data\_inscriere, data\_incetare) values ('Popescu Ion', 'PF', sysdate - 10, null);

insert into client(nume, tip\_client, data\_inscriere, data\_incetare) values ('Alexandru Petre', 'PF', sysdate - 5, null);

insert into client(nume, tip\_client, data\_inscriere, data\_incetare) values ('Auchan', 'PJ', sysdate - 100 , null);

insert into client(nume, tip\_client, data\_inscriere, data\_incetare) values ('Carrefour', 'PJ', sysdate - 200, null);

insert into client(nume, tip\_client, data\_inscriere, data\_incetare) values ('George Banica', 'PF', sysdate - 1, null);

insert into client(nume, tip\_client, data\_inscriere, data\_incetare) values ('Mega Image', 'PJ', sysdate - 300, null);

insert into client(nume, tip\_client, data\_inscriere, data\_incetare) values ('Camil Ressu', 'PF', sysdate - 300, sysdate);

insert into client(nume, tip\_client, data\_inscriere, data\_incetare) values ('Mihail Sadoveanu', 'PF', sysdate, null);

insert into client(nume, tip\_client, data\_inscriere, data\_incetare) values ('Alexandru Vlahuta', 'PF', sysdate-1, null);

insert into client(nume, tip\_client, data\_inscriere, data\_incetare) values ('Penny', 'PJ', sysdate-500, null);

insert into client(nume, tip\_client, data\_inscriere, data\_incetare) values ('Glovo', 'PJ', sysdate-50, null);

commit;

insert into comerciant(nume, cod\_client, data\_inscriere) values ('Auchan',

(select id\_client from client where nume = 'Auchan'), sysdate - 100);

insert into comerciant(nume, cod\_client, data\_inscriere) values ('Carrefour',

(select id\_client from client where nume = 'Carrefour'), sysdate - 200);

insert into comerciant(nume, cod\_client, data\_inscriere) values ('Mega Image',

(select id\_client from client where nume = 'Mega Image'), sysdate - 300);

insert into comerciant(nume, cod\_client, data\_inscriere) values ('Penny',

(select id\_client from client where nume = 'Penny'), sysdate - 500);

insert into comerciant(nume, cod\_client, data\_inscriere) values ('Glovo',

(select id\_client from client where nume = 'Glovo'), sysdate - 50);

commit;

insert into cont(tip\_cont, sold, nume\_cont, cod\_client) values ('CURENT', 25.39, 'RO49POCB1V3100759384', (select id\_client from client where nume = 'Popescu Ion'));

insert into cont(tip\_cont, sold, nume\_cont, cod\_client) values ('CURENT', 120.45, 'RO49POCD2X7100869495', (select id\_client from client where nume = 'Alexandru Petre'));

insert into cont(tip\_cont, sold, nume\_cont, cod\_client) values ('CURENT', 67.85, 'RO49POCG5Z4200292728', (select id\_client from client where nume = 'George Banica'));

insert into cont(tip\_cont, sold, nume\_cont, cod\_client) values ('CURENT', 143.20, 'RO49POCE6T5300303839', (select id\_client from client where nume = 'Camil Ressu'));

insert into cont(tip\_cont, sold, nume\_cont, cod\_client) values ('CURENT', 70.50, 'RO49POCJ5S5300303839', (select id\_client from client where nume = 'Mihail Sadoveanu'));

insert into cont(tip\_cont, sold, nume\_cont, cod\_client) values ('CURENT', 93.82, 'RO49POCH5K5300303839', (select id\_client from client where nume = 'Alexandru Vlahuta'));

insert into cont(tip\_cont, sold, nume\_cont, cod\_client) values ('CURENT', 398.00, 'RO49POCE3Y8200970506', (select id\_client from client where nume = 'Auchan'));

insert into cont(tip\_cont, sold, nume\_cont, cod\_client) values ('CURENT', 450.30, 'RO49POCF4W9300181617', (select id\_client from client where nume = 'Carrefour'));

insert into cont(tip\_cont, sold, nume\_cont, cod\_client) values ('CURENT', 750.20, 'RO49POCH5P8400460329', (select id\_client from client where nume = 'Mega Image'));

insert into cont(tip\_cont, sold, nume\_cont, cod\_client) values ('CURENT', 556.20, 'RO49POCL6D9500570430', (select id\_client from client where nume = 'Penny'));

insert into cont(tip\_cont, sold, nume\_cont, cod\_client) values ('CURENT', 950.80, 'RO49POCU7M4600981541', (select id\_client from client where nume = 'Glovo'));

commit;

insert into card(tip\_card, cod\_cont, data\_emitere, numar\_card) values

('DEBIT', (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Popescu Ion'),

sysdate - 100, '4012888888881881');

insert into card(tip\_card, cod\_cont, data\_emitere, numar\_card) values ('CREDIT',

(select id\_cont from cont, client where cod\_client = id\_client and nume = 'Alexandru Petre'),

sysdate - 30, '5019717010103742');

insert into card(tip\_card, cod\_cont, data\_emitere, numar\_card) values ('DEBIT',

(select id\_cont from cont, client where cod\_client = id\_client and nume = 'George Banica'),

sysdate - 80, '5610591081018250');

insert into card(tip\_card, cod\_cont, data\_emitere, numar\_card) values ('CREDIT',

(select id\_cont from cont, client where cod\_client = id\_client and nume = 'Camil Ressu'),

sysdate - 60, '6721691081018250');

insert into card(tip\_card, cod\_cont, data\_emitere, numar\_card) values ('DEBIT',

(select id\_cont from cont, client where cod\_client = id\_client and nume = 'Mihail Sadoveanu'),

sysdate - 55, '5721691082019361');

insert into card(tip\_card, cod\_cont, data\_emitere, numar\_card) values ('CREDIT',

(select id\_cont from cont, client where cod\_client = id\_client and nume = 'Alexandru Vlahuta'),

sysdate - 45, '5832701071019361');

commit;

insert into locatie(tara, oras, strada) values ('Romania', 'Bucuresti', 'Bacovia 12'); -- Auchan

insert into locatie(tara, oras, strada) values ('Romania', 'Constanta', 'Arghezi 20'); -- Carrefour

insert into locatie(tara, oras, strada) values ('Romania', 'Brasov', 'Rebreanu 4'); -- Mega Image

insert into locatie(tara, oras, strada) values ('Romania', 'Timisoara', 'Eminescu 56'); -- Penny

insert into locatie(site) values ('http://glovo.ro'); -- Glovo

commit;

insert into canal\_plata(cod\_comerciant, tip\_echipament, cod\_cont, data\_inceput, cod\_locatie) values

((select id\_comerciant from comerciant where nume = 'Auchan'), 'POS',

(select id\_cont from cont, client where cod\_client = id\_client and nume = 'Auchan'),

sysdate - 30, (select id\_locatie from locatie where oras = 'Bucuresti'));

insert into canal\_plata(cod\_comerciant, tip\_echipament, cod\_cont, data\_inceput, cod\_locatie) values

((select id\_comerciant from comerciant where nume = 'Carrefour'), 'POS',

(select id\_cont from cont, client where cod\_client = id\_client and nume = 'Carrefour'),

sysdate - 20, (select id\_locatie from locatie where oras = 'Constanta'));

insert into canal\_plata(cod\_comerciant, tip\_echipament, cod\_cont, data\_inceput, cod\_locatie) values

((select id\_comerciant from comerciant where nume = 'Mega Image'), 'POS',

(select id\_cont from cont, client where cod\_client = id\_client and nume = 'Mega Image'),

sysdate - 10, (select id\_locatie from locatie where oras = 'Brasov'));

insert into canal\_plata(cod\_comerciant, tip\_echipament, cod\_cont, data\_inceput, cod\_locatie) values

((select id\_comerciant from comerciant where nume = 'Penny'), 'POS',

(select id\_cont from cont, client where cod\_client = id\_client and nume = 'Penny'),

sysdate - 40, (select id\_locatie from locatie where oras = 'Timisoara'));

insert into canal\_plata(cod\_comerciant, tip\_echipament, cod\_cont, data\_inceput, cod\_locatie) values

((select id\_comerciant from comerciant where nume = 'Glovo'), 'ONLINE',

(select id\_cont from cont, client where cod\_client = id\_client and nume = 'Glovo'),

sysdate - 50, (select id\_locatie from locatie where site = 'http://glovo.ro'));

commit;

Insert into CANAL\_PLATA (COD\_COMERCIANT,COD\_LOCATIE,TIP\_ECHIPAMENT,COD\_CONT,DATA\_INCEPUT,DATA\_INCETARE) values (1,1,'POS',7,to\_date('26-DEC-22','DD-MON-RR'),null);

Insert into CANAL\_PLATA (COD\_COMERCIANT,COD\_LOCATIE,TIP\_ECHIPAMENT,COD\_CONT,DATA\_INCEPUT,DATA\_INCETARE) values (2,2,'POS',8,to\_date('05-JAN-23','DD-MON-RR'),null);

Insert into CANAL\_PLATA (COD\_COMERCIANT,COD\_LOCATIE,TIP\_ECHIPAMENT,COD\_CONT,DATA\_INCEPUT,DATA\_INCETARE) values (3,3,'POS',9,to\_date('15-JAN-23','DD-MON-RR'),null);

Insert into CANAL\_PLATA (COD\_COMERCIANT,COD\_LOCATIE,TIP\_ECHIPAMENT,COD\_CONT,DATA\_INCEPUT,DATA\_INCETARE) values (4,4,'POS',10,to\_date('16-DEC-22','DD-MON-RR'),null);

Insert into CANAL\_PLATA (COD\_COMERCIANT,COD\_LOCATIE,TIP\_ECHIPAMENT,COD\_CONT,DATA\_INCEPUT,DATA\_INCETARE) values (5,5,'ONLINE',11,to\_date('06-DEC-22','DD-MON-RR'),null);

Insert into CANAL\_PLATA (COD\_COMERCIANT,COD\_LOCATIE,TIP\_ECHIPAMENT,COD\_CONT,DATA\_INCEPUT,DATA\_INCETARE) values (4,1,'POS',1,to\_date('27-DEC-22','DD-MON-RR'),null);

Insert into CANAL\_PLATA (COD\_COMERCIANT,COD\_LOCATIE,TIP\_ECHIPAMENT,COD\_CONT,DATA\_INCEPUT,DATA\_INCETARE) values (3,3,'POS',3,to\_date('16-JAN-23','DD-MON-RR'),null);

Insert into CANAL\_PLATA (COD\_COMERCIANT,COD\_LOCATIE,TIP\_ECHIPAMENT,COD\_CONT,DATA\_INCEPUT,DATA\_INCETARE) values (3,4,'POS',4,to\_date('17-DEC-22','DD-MON-RR'),null);

Insert into CANAL\_PLATA (COD\_COMERCIANT,COD\_LOCATIE,TIP\_ECHIPAMENT,COD\_CONT,DATA\_INCEPUT,DATA\_INCETARE) values (1,5,'ONLINE',5,to\_date('07-DEC-22','DD-MON-RR'),null);

Insert into CANAL\_PLATA (COD\_COMERCIANT,COD\_LOCATIE,TIP\_ECHIPAMENT,COD\_CONT,DATA\_INCEPUT,DATA\_INCETARE) values (2,5,'ONLINE',6,to\_date('07-DEC-22','DD-MON-RR'),null);

Insert into CANAL\_PLATA (COD\_COMERCIANT,COD\_LOCATIE,TIP\_ECHIPAMENT,COD\_CONT,DATA\_INCEPUT,DATA\_INCETARE) values (4,1,'POS',2,to\_date('27-DEC-22','DD-MON-RR'),null);

commit;

insert into tranzactii(suma, cod\_cont\_debitor, cod\_cont\_creditor, data\_initiere, data\_procesare, stare) values

(25.43, (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Alexandru Petre'), (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Auchan'),

sysdate - 30, sysdate -30 + 1/24/60, 'Procesat');

insert into tranzactii(suma, cod\_cont\_debitor, cod\_cont\_creditor, data\_initiere, data\_procesare, stare) values

(125.43, (select id\_cont from cont, client where cod\_client = id\_client and nume = 'George Banica'), (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Glovo'),

sysdate - 30, sysdate -30 + 2 \* 1/24/60, 'Procesat');

insert into tranzactii(suma, cod\_cont\_debitor, cod\_cont\_creditor, data\_initiere, data\_procesare, stare) values

(125.43, (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Mihail Sadoveanu'), (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Carrefour'),

sysdate - 30, sysdate -30 + 0.5 \* 1/24/60, 'Procesat');

insert into tranzactii(suma, cod\_cont\_debitor, cod\_cont\_creditor, data\_initiere, data\_procesare, stare) values

(925.43, (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Alexandru Vlahuta'), (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Penny'),

sysdate - 30, sysdate -30 + 0.25 \* 1/24/60, 'Eroare');

insert into tranzactii(suma, cod\_cont\_debitor, cod\_cont\_creditor, data\_initiere, data\_procesare, stare) values

(325.43, (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Alexandru Vlahuta'), (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Mega Image'),

sysdate, null, 'Initiat');

insert into tranzactii(suma, cod\_cont\_debitor, cod\_cont\_creditor, data\_initiere, data\_procesare, stare) values

(325.43, (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Camil Ressu'), (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Auchan'),

sysdate, null, 'Initiat');

insert into tranzactii(suma, cod\_cont\_debitor, cod\_cont\_creditor, data\_initiere, data\_procesare, stare) values

(65.13, (select id\_cont from cont, client where cod\_client = id\_client and nume = 'George Banica'), (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Glovo'),

sysdate, null, 'Initiat');

insert into tranzactii(suma, cod\_cont\_debitor, cod\_cont\_creditor, data\_initiere, data\_procesare, stare) values

(1975.33, (select id\_cont from cont, client where cod\_client = id\_client and nume = 'George Banica'), (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Penny'),

sysdate - 30, sysdate -30 + 0.25 \* 1/24/60, 'Eroare');

insert into tranzactii(suma, cod\_cont\_debitor, cod\_cont\_creditor, data\_initiere, data\_procesare, stare) values

(25.43, (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Popescu Ion'), (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Penny'),

sysdate - 30, sysdate -30 + 1/24/60, 'Procesat');

insert into tranzactii(suma, cod\_cont\_debitor, cod\_cont\_creditor, data\_initiere, data\_procesare, stare) values

(125.43, (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Popescu Ion'), (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Auchan'),

sysdate - 30, sysdate -30 + 2 \* 1/24/60, 'Procesat');

commit;

insert into tranzactii(suma, cod\_cont\_debitor, cod\_cont\_creditor, data\_initiere, data\_procesare, stare) values

(25.43, (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Popescu Ion'), (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Penny'),

sysdate - 1, sysdate -1 + 1/24/60, 'Procesat');

insert into tranzactii(suma, cod\_cont\_debitor, cod\_cont\_creditor, data\_initiere, data\_procesare, stare) values

(125.43, (select id\_cont from cont, client where cod\_client = id\_client and nume = 'George Banica'), (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Carrefour'),

sysdate - 2, sysdate - 2 + 2 \* 1/24/60, 'Procesat');

insert into tranzactii(suma, cod\_cont\_debitor, cod\_cont\_creditor, data\_initiere, data\_procesare, stare) values

(125.43, (select id\_cont from cont, client where cod\_client = id\_client and nume = 'George Banica'), (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Mega Image'),

sysdate - 5, sysdate - 5 + 2 \* 1/24/60, 'Procesat');

insert into tranzactii(suma, cod\_cont\_debitor, cod\_cont\_creditor, data\_initiere, data\_procesare, stare) values

(125.43, (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Mihail Sadoveanu'), (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Glovo'),

sysdate - 7, sysdate - 7 + 2 \* 1/24/60, 'Procesat');

insert into tranzactii(suma, cod\_cont\_debitor, cod\_cont\_creditor, data\_initiere, data\_procesare, stare) values

(15.0, (select id\_cont from cont, client where cod\_client = id\_client and nume = 'George Banica'), (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Auchan'),

sysdate - 30, sysdate -30 + 1/24/60, 'Procesat');

insert into tranzactii(suma, cod\_cont\_debitor, cod\_cont\_creditor, data\_initiere, data\_procesare, stare) values

(15.0, (select id\_cont from cont, client where cod\_client = id\_client and nume = 'George Banica'), (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Glovo'),

sysdate - 30, sysdate -30 + 2 \* 1/24/60, 'Procesat');

insert into tranzactii(suma, cod\_cont\_debitor, cod\_cont\_creditor, data\_initiere, data\_procesare, stare) values

(15.0, (select id\_cont from cont, client where cod\_client = id\_client and nume = 'George Banica'), (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Carrefour'),

sysdate - 30, sysdate -30 + 0.5 \* 1/24/60, 'Procesat');

insert into tranzactii(suma, cod\_cont\_debitor, cod\_cont\_creditor, data\_initiere, data\_procesare, stare) values

(15.0, (select id\_cont from cont, client where cod\_client = id\_client and nume = 'George Banica'), (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Penny'),

sysdate - 30, sysdate -30 + 0.5 \* 1/24/60, 'Procesat');

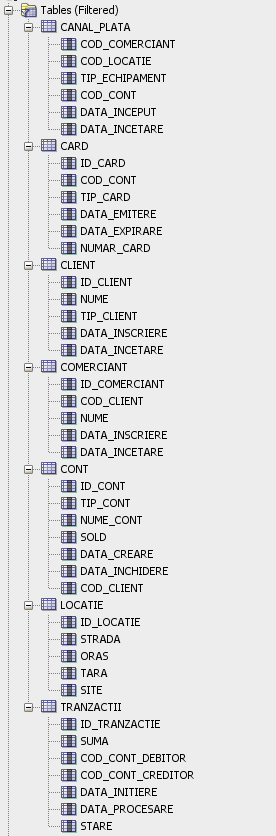
insert into tranzactii(suma, cod\_cont\_debitor, cod\_cont\_creditor, data\_initiere, data\_procesare, stare) values

(15.0, (select id\_cont from cont, client where cod\_client = id\_client and nume = 'George Banica'), (select id\_cont from cont, client where cod\_client = id\_client and nume = 'Mega Image'),

sysdate - 30, sysdate -30 + 0.5 \* 1/24/60, 'Procesat');

commit;

Captura de ecran SQLDeveloper care demonstreaza crearea tabelelor:



Crearea utilizatorilor:

CREATE USER DW

IDENTIFIED BY aSyifijdasfnlsjaJNSc;

GRANT CREATE SESSION TO DW;

GRANT CONNECT, DBA TO DW;

GRANT CREATE DIMENSION TO DW;

CREATE USER DB

IDENTIFIED BY ufbivgfbdDHASU832fjdl;

GRANT CREATE SESSION TO DB;

GRANT CONNECT, DBA TO DB;

GRANT SELECT ON DB.TRANZACTII TO DW;

GRANT SELECT ON DB.CLIENTI TO DW;

GRANT SELECT ON DB.CARD TO DW;

GRANT SELECT ON DB.CONT TO DW;

GRANT SELECT ON DB.COMERCIANT TO DW;

GRANT SELECT ON DB.CANAL\_PLATA TO DW;

GRANT SELECT ON DB.LOCATIE TO DW;

**Exercitiul 2: Generarea datelor și inserarea acestora în tabele (puteți utiliza ca interfață o aplicație creată anul trecut la celelalte materii)**

//TODO

**Exercitiul 3: Crearea bazei de date depozit și a utilizatorilor**

Crearea bazei de date depozit :

-- DROP TABLE FACT\_TRANZACTII;

-- DROP TABLE DIM\_CALENDAR;

-- DROP TABLE DIM\_CANAL\_PLATA;

-- DROP TABLE DIM\_LOCATIE;

-- DROP TABLE DIM\_COMERCIANT;

-- DROP TABLE DIM\_DETALII\_PLATA;

-- DROP TABLE DIM\_CLIENT;

-- DROP TABLE DIM\_STARE;

CREATE TABLE DIM\_STARE (

ID\_Stare NUMBER(10) NOT NULL PRIMARY KEY,

Stare VARCHAR2(50 BYTE) NOT NULL

);

CREATE TABLE DIM\_LOCATIE (

ID\_Locatie NUMBER(10) PRIMARY KEY,

Strada VARCHAR(100) DEFAULT NULL,

Oras VARCHAR(100) DEFAULT NULL,

Tara VARCHAR(100) DEFAULT NULL,

Site VARCHAR(100) DEFAULT NULL

) PARTITION by list (oras) (

PARTITION TM

VALUES

('Timisoara'),

PARTITION B

VALUES

('Bucuresti'),

PARTITION CT

VALUES

('Constanta'),

PARTITION BV

VALUES

('Brasov'),

PARTITION nedefinit

VALUES

(DEFAULT)

);

drop index tara\_idx;

create index tara\_idx on dim\_locatie (tara) local;

CREATE TABLE DIM\_DETALII\_PLATA (

ID\_Cont NUMBER(10) PRIMARY KEY,

Tip\_Card VARCHAR(10) NOT NULL,

Tip\_Cont VARCHAR(10) NOT NULL

);

CREATE TABLE DIM\_COMERCIANT (

ID\_Comerciant NUMBER(10) PRIMARY KEY,

Nume VARCHAR(100),

STATUS VARCHAR(15)

);

CREATE TABLE DIM\_CANAL\_PLATA (

ID\_Canal\_Plata NUMBER(10) NOT NULL,

Tip\_Canal\_Plata VARCHAR(10) NOT NULL

);

DROP TABLE DIM\_CALENDAR;

CREATE TABLE DIM\_CALENDAR AS

SELECT

TO\_NUMBER(

TO\_CHAR(

TO\_DATE('31/12/2020', 'DD/MM/YYYY') + NUMTODSINTERVAL(n, 'day'),

'YYYYMMDD'

)

) AS ID\_Data,

TO\_DATE('31/12/2020', 'DD/MM/YYYY') + NUMTODSINTERVAL(n, 'day') AS Data,

TO\_CHAR(

TO\_DATE('31/12/2020', 'DD/MM/YYYY') + NUMTODSINTERVAL(n, 'day'),

'DD'

) AS Ziua,

TO\_CHAR(

TO\_DATE('31/12/2020', 'DD/MM/YYYY') + NUMTODSINTERVAL(n, 'day'),

'Month'

) AS Luna,

TO\_CHAR(

TO\_DATE('31/12/2020', 'DD/MM/YYYY') + NUMTODSINTERVAL(n, 'day'),

'YYYY'

) AS Anul

FROM

(

SELECT

LEVEL n

FROM

dual connect by LEVEL <= 2000

);

drop index calendar\_idx;

create index calendar\_idx on dim\_calendar (luna) global partition by hash(luna) partitions 12;

DROP TABLE FACT\_TRANZACTII;

CREATE TABLE FACT\_TRANZACTII (

ID\_Tranzactie NUMBER(10) NOT NULL,

ID\_Client NUMBER(10) NOT NULL,

ID\_Cont NUMBER(10) NOT NULL,

ID\_Comerciant NUMBER(10) NOT NULL,

ID\_Canal\_Plata NUMBER(10) NOT NULL,

ID\_Stare NUMBER(10) NOT NULL,

ID\_Locatie NUMBER(10) NOT NULL,

ID\_Data NUMBER(8) NOT NULL,

Suma NUMBER(10, 2) NOT NULL,

Durata NUMBER(7, 2) DEFAULT NULL,

PRIMARY KEY(ID\_Tranzactie, ID\_Client, ID\_Cont)

) PARTITION by RANGE(id\_data) INTERVAL(1) (

PARTITION partitie\_initiala

VALUES

less than (20220101)

);

CREATE TABLE DIM\_CLIENT

(

ID\_Client NUMBER(10,0) PRIMARY KEY,

Nume VARCHAR2(100) NOT NULL,

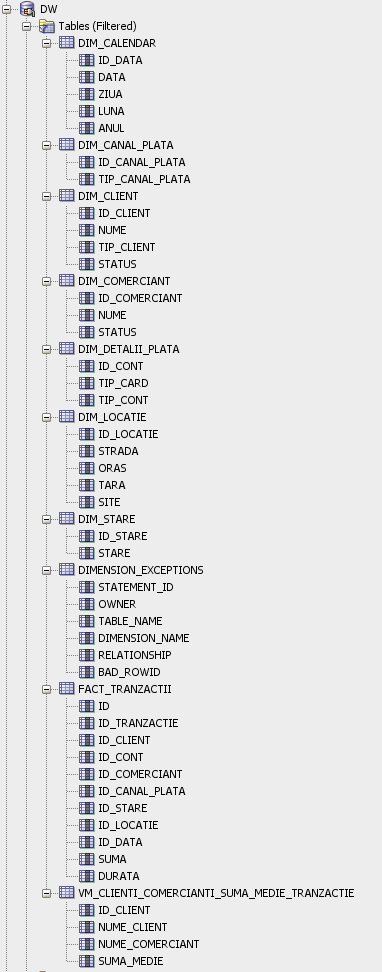
Tip\_Client VARCHAR2(50) NOT NULL,

Status VARCHAR2(15) NOT NULL

);

CREATE bitmap INDEX dim\_client\_bmp ON dim\_client (tip\_client);

Captura de ecran care demonstreaza crearea tabelelor din cadrul bazei de date de tip depozit :



*Obs : Pentru cearea utilizatorilor -> exercitiul 1*

**Exercitiul 4: Popularea cu informații a bazei de date depozit folosind ca sursă datele din baza de date OLTP**

-- DIM\_STARE

TRUNCATE TABLE DIM\_STARE;

INSERT INTO

DIM\_STARE

SELECT

DISTINCT CASE

WHEN Stare = 'Initiat' THEN 1

WHEN Stare = 'Procesat' THEN 2

WHEN Stare = 'Eroare' THEN 3

ELSE NULL

END AS ID\_Stare,

Stare

FROM

db.Tranzactii;

-- DIM\_CLIENT

TRUNCATE TABLE DIM\_CLIENT;

INSERT INTO

DIM\_CLIENT

SELECT

ID\_Client,

Nume,

Tip\_Client,

CASE

WHEN Data\_Incetare IS NOT NULL THEN 'Incetat'

ELSE 'Inscris'

END AS STATUS

FROM

db.client;

-- DIM\_COMERCIANT

TRUNCATE TABLE DIM\_COMERCIANT;

INSERT INTO

DIM\_COMERCIANT

SELECT

ID\_Comerciant,

Nume,

CASE

WHEN Data\_Incetare IS NOT NULL THEN 'Incetat'

ELSE 'Inscris'

END AS STATUS

FROM

db.comerciant;

-- DIM\_DETALII\_PLATA

TRUNCATE TABLE DIM\_DETALII\_PLATA;

INSERT INTO

DIM\_DETALII\_PLATA

SELECT

ID\_Cont,

Tip\_Cont,

Tip\_Card

FROM

db.cont ct

INNER JOIN db.card cd ON ct.ID\_Cont = cd.COD\_Cont;

-- DIM\_Locatie

TRUNCATE TABLE DIM\_LOCATIE;

INSERT INTO

DIM\_LOCATIE

SELECT

ID\_Locatie,

Strada,

Oras,

Tara,

Site

FROM

db.locatie;

-- DIM\_CANAL\_PLATA

TRUNCATE TABLE DIM\_CANAL\_PLATA;

INSERT INTO

DIM\_CANAL\_PLATA

SELECT

DISTINCT CASE

WHEN Tip\_Echipament = 'POS' THEN 1

WHEN Tip\_Echipament = 'ONLINE' THEN 2

ELSE NULL

END AS ID\_Canal\_Plata,

Tip\_Echipament AS Tip\_Canal\_Plata

FROM

db.canal\_plata;

-- FACT\_TRANZACTII

INSERT INTO

FACT\_TRANZACTII (

ID\_Tranzactie,

ID\_Client,

ID\_Cont,

ID\_Comerciant,

ID\_Canal\_Plata,

ID\_Stare,

ID\_Locatie,

ID\_Data,

Suma,

Durata

)

SELECT

ID\_Tranzactie,

ID\_Client,

ID\_Cont,

ID\_Comerciant,

CASE

WHEN Tip\_Echipament = 'POS' THEN 1

WHEN Tip\_Echipament = 'ONLINE' THEN 2

ELSE NULL

END AS ID\_Canal\_Plata,

CASE

WHEN Stare = 'Initiat' THEN 1

WHEN Stare = 'Procesat' THEN 2

WHEN Stare = 'Eroare' THEN 3

ELSE NULL

END AS ID\_Stare,

ID\_Locatie,

TO\_CHAR(DATA\_INITIERE, 'YYYYMMDD') AS ID\_Data,

- SUMA,

TO\_CHAR(

(DATA\_PROCESARE - DATA\_INITIERE) \* 1440,

'99999.99'

) AS Durata

FROM

db.tranzactii t

LEFT JOIN db.cont co ON t.COD\_CONT\_DEBITOR = co.ID\_CONT

LEFT JOIN db.client cl ON co.COD\_CLIENT = cl.ID\_CLIENT

LEFT JOIN db.canal\_plata cp ON co.ID\_CONT = cp.COD\_CONT

LEFT JOIN db.locatie loc ON cp.COD\_LOCATIE = loc.ID\_LOCATIE

LEFT JOIN db.comerciant cm ON cp.COD\_COMERCIANT = cm.ID\_COMERCIANT

UNION

ALL

SELECT

ID\_Tranzactie,

ID\_Client,

ID\_Cont,

ID\_Comerciant,

CASE

WHEN Tip\_Echipament = 'POS' THEN 1

WHEN Tip\_Echipament = 'ONLINE' THEN 2

ELSE NULL

END AS ID\_Canal\_Plata,

CASE

WHEN Stare = 'Initiat' THEN 1

WHEN Stare = 'Procesat' THEN 2

WHEN Stare = 'Eroare' THEN 3

ELSE NULL

END AS ID\_Stare,

ID\_Locatie,

TO\_CHAR(DATA\_INITIERE, 'YYYYMMDD') AS ID\_Data,

SUMA,

TO\_CHAR(

(DATA\_PROCESARE - DATA\_INITIERE) \* 1440,

'99999.99'

) AS Durata

FROM

db.tranzactii t

LEFT JOIN db.cont co ON t.COD\_CONT\_CREDITOR = co.ID\_CONT

LEFT JOIN db.client cl ON co.COD\_CLIENT = cl.ID\_CLIENT

LEFT JOIN db.canal\_plata cp ON co.ID\_CONT = cp.COD\_CONT

LEFT JOIN db.locatie loc ON cp.COD\_LOCATIE = loc.ID\_LOCATIE

LEFT JOIN db.comerciant cm ON cp.COD\_COMERCIANT = cm.ID\_COMERCIANT;

**Exercitiul 5: Definirea constrângerilor**

--(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

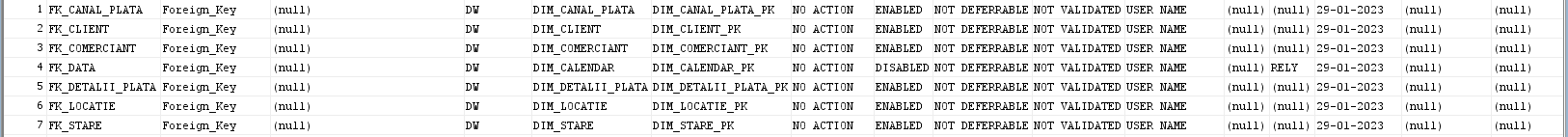
FOREIGN KEY (ID\_Data)

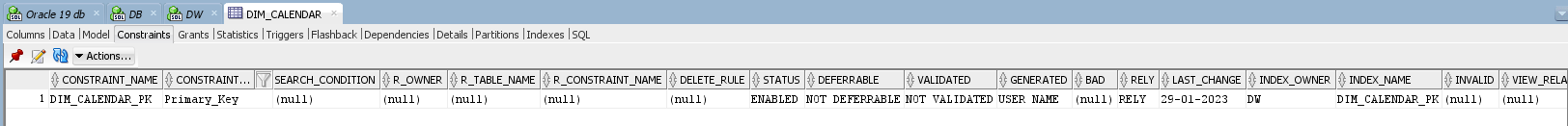
REFERENCES DIM\_CALENDAR (ID\_Data)

RELY DISABLE NOVALIDATE;

Capturi de ecran care demonstreaza crearea constrangerilor:







**Exercitiul 6: Definirea indecșilor și a cererilor SQL însoțite de planul de execuție al acestora(din care să reiasă ca optimizatorul utilizează eficient indecșii definiți)**

Creare index global:Graphical user interface, text, application, email

Description automatically generated

Consulatare dictionarul datelor pentru verificarea crearii partitiilor pe index:Graphical user interface, application

Description automatically generated

Folosire index :

EXPLAIN PLAN

SET STATEMENT\_ID = 's1\_index\_global' FOR

select /\*+ index(dim\_calendar calendar\_idx) \*/ \* from dim\_calendar where luna= 'June';

SELECT plan\_table\_output

FROM

table(dbms\_xplan.display('plan\_table', 's1\_index\_global','serial'));

Plan hash value: 2382551087

---------------------------------------------------------------------------------------------------------------------

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time | Pstart| Pstop |

---------------------------------------------------------------------------------------------------------------------

| 0 | SELECT STATEMENT | | 167 | 5344 | 8 (0)| 00:00:01 | | |

| 1 | PARTITION HASH SINGLE | | 167 | 5344 | 8 (0)| 00:00:01 | 7 | 7 |

| 2 | TABLE ACCESS BY INDEX ROWID BATCHED| DIM\_CALENDAR | 167 | 5344 | 8 (0)| 00:00:01 | | |

|\* 3 | INDEX RANGE SCAN | CALENDAR\_IDX | 167 | | 1 (0)| 00:00:01 | 7 | 7 |

---------------------------------------------------------------------------------------------------------------------

Creare index local :

Graphical user interface, text, application, email

Description automatically generated

Folosire index:

EXPLAIN PLAN

SET STATEMENT\_ID = 's1\_index\_local' FOR

select \* from dim\_locatie where tara = 'Romania';

SELECT plan\_table\_output

FROM

table(dbms\_xplan.display('plan\_table', 's1\_index\_local','serial'));

Plan hash value: 1361974882

--------------------------------------------------------------------------------------------------------------------------

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time | Pstart| Pstop |

--------------------------------------------------------------------------------------------------------------------------

| 0 | SELECT STATEMENT | | 4 | 124 | 8 (0)| 00:00:01 | | |

| 1 | PARTITION LIST ALL | | 4 | 124 | 8 (0)| 00:00:01 | 1 | 5 |

| 2 | TABLE ACCESS BY LOCAL INDEX ROWID BATCHED| DIM\_LOCATIE | 4 | 124 | 8 (0)| 00:00:01 | 1 | 5 |

|\* 3 | INDEX RANGE SCAN | TARA\_IDX | 4 | | 4 (0)| 00:00:01 | 1 | 5 |

--------------------------------------------------------------------------------------------------------------------------

Bitmap index:

Graphical user interface, text, application, email

Description automatically generated

Folosire index:

analyze index dim\_client\_bmp compute statistics;

EXPLAIN PLAN

SET STATEMENT\_ID = 's1\_index\_bmp' FOR

select /\*+ index(dim\_client dim\_client\_bmp) \*/ \* from dim\_client where tip\_client = 'PF';

SELECT plan\_table\_output

FROM

table(dbms\_xplan.display('plan\_table', 's1\_index\_bmp','serial'));

Plan hash value: 2457481617

------------------------------------------------------------------------------------------------------

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time |

------------------------------------------------------------------------------------------------------

| 0 | SELECT STATEMENT | | 6 | 156 | 7 (0)| 00:00:01 |

| 1 | TABLE ACCESS BY INDEX ROWID BATCHED| DIM\_CLIENT | 6 | 156 | 7 (0)| 00:00:01 |

| 2 | BITMAP CONVERSION TO ROWIDS | | | | | |

|\* 3 | BITMAP INDEX SINGLE VALUE | DIM\_CLIENT\_BMP | | | | |

Bitmap join index:

create bitmap index bmp\_join\_idx on fact\_tranzactii (suma) from fact\_tranzactii f, dim\_comerciant d

where f.id\_comerciant = d.id\_comerciant local;

analyze index bmp\_join\_idx compute statistics;

alter session set star\_transformation\_enabled = true;

Folosire bitmap join index:

EXPLAIN PLAN

SET STATEMENT\_ID = 's1\_index\_join\_bmp' FOR

select /\*+ STAR\_TRANSFORMATION \*/

/\*+ FACT(fact\_tranzactii) \*/

f.suma, d.nume

from fact\_tranzactii f, dim\_comerciant d

where f.id\_comerciant = d.id\_comerciant

and suma = 50;

SELECT plan\_table\_output

FROM

table(dbms\_xplan.display('plan\_table', 's1\_index\_join\_bmp','serial'));

Plan hash value: 3558226070

--------------------------------------------------------------------------------------------------------------------------------

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time | Pstart| Pstop |

--------------------------------------------------------------------------------------------------------------------------------

| 0 | SELECT STATEMENT | | 1 | 37 | 1 (0)| 00:00:01 | | |

| 1 | NESTED LOOPS | | 1 | 37 | 1 (0)| 00:00:01 | | |

| 2 | NESTED LOOPS | | 1 | 37 | 1 (0)| 00:00:01 | | |

| 3 | PARTITION RANGE ALL | | 1 | 26 | 1 (0)| 00:00:01 | 1 |1048575|

|\* 4 | TABLE ACCESS BY LOCAL INDEX ROWID BATCHED| FACT\_TRANZACTII | 1 | 26 | 1 (0)| 00:00:01 | 1 |1048575|

| 5 | BITMAP CONVERSION TO ROWIDS | | | | | | | |

|\* 6 | BITMAP INDEX SINGLE VALUE | BMP\_JOIN\_IDX | | | | | 1 |1048575|

|\* 7 | INDEX UNIQUE SCAN | SYS\_C008471 | 1 | | 0 (0)| 00:00:01 | | |

| 8 | TABLE ACCESS BY INDEX ROWID | DIM\_COMERCIANT | 1 | 11 | 0 (0)| 00:00:01 | | |

--------------------------------------------------------------------------------------------------------------------------------

**Exercitiul 7: Definirea  obiectelor  de  tip  dimensiune,  validarea  acestora (din care să reiasă că datele respectă constrângerile impuse prin aceste tipuri de obiecte)**

-- 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');

**Exercitiul 8: Definirea partițiilor; definirea cererilor SQL însoțite de planul de execuție al acestora din care să reiasă ca optimizorul utilizează eficient partițiile**

Definire partitie pe tabela de fapte :Graphical user interface, text, application, email

Description automatically generated

Partitionarea te tip interval este o extensie a partitionarii de tip range. Partitiile noi sunt create autmomat de catre SGBD. atunci cand coloana pe care s-a definit regula de partitionare “depaseste” toate valorile. Cel putin o partitie initiala trebuie creata. Valoarea cheii de partitie interval determină valoarea superioara a partitiilor interval, care se numește punct de tranziție, iar serverul de baze de date creează partiții de interval noi pentru datele care au o valoare mai mare decat punctul de tranziție.

Se pot defini pana la 1,048,757 partitii la nivel de tabel. Avand in vedere ca partitionarea este la nivel de zi se pot stoca intr-un tabel 2, 873 de ani.

Verificarea crearii partitiilor prin consultarea dictionarului datelor:Table

Description automatically generated

analyze table fact\_tranzactii compute statistics;

Table FACT\_TRANZACTII analyzed.

EXPLAIN PLAN

SET STATEMENT\_ID = 'st\_fact\_partition'

FOR

select \* from fact\_tranzactii where id\_data = 20221231;

SELECT plan\_table\_output

FROM

table(dbms\_xplan.display('plan\_table','st\_fact\_partition','serial'));

Plan hash value: 3811362893

----------------------------------------------------------------------------------------------------------

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time | Pstart| Pstop |

----------------------------------------------------------------------------------------------------------

| 0 | SELECT STATEMENT | | 24 | 648 | 274 (0)| 00:00:01 | | |

| 1 | PARTITION RANGE SINGLE| | 24 | 648 | 274 (0)| 00:00:01 | 1132 | 1132 |

|\* 2 | TABLE ACCESS FULL | FACT\_TRANZACTII | 24 | 648 | 274 (0)| 00:00:01 | 1132 | 1132 |

----------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):

---------------------------------------------------

2 – filter("ID\_DATA"=20221231)

EXPLAIN PLAN

SET STATEMENT\_ID = 'st\_fact\_partition\_pruning'

FOR

select \* from fact\_tranzactii where id\_data between 20230123 and 20230128;

SELECT plan\_table\_output

FROM

table(dbms\_xplan.display('plan\_table','st\_fact\_partition\_pruning','serial'));

Plan hash value: 370717546

------------------------------------------------------------------------------------------------------------

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time | Pstart| Pstop |

------------------------------------------------------------------------------------------------------------

| 0 | SELECT STATEMENT | | 6 | 168 | 820 (1)| 00:00:01 | | |

| 1 | PARTITION RANGE ITERATOR| | 6 | 168 | 820 (1)| 00:00:01 | 10024 | 10029 |

|\* 2 | TABLE ACCESS FULL | FACT\_TRANZACTII | 6 | 168 | 820 (1)| 00:00:01 | 10024 | 10029 |

------------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):

---------------------------------------------------

2 - filter("ID\_DATA">=20230123 AND "ID\_DATA"<=20230128)

Definire partitie pe tabela de dimensiune locatie :Graphical user interface, text, application, chat or text message

Description automatically generated

Verificarea crearii partitiilor prin consultarea dictionarului datelor.Graphical user interface, text, application

Description automatically generated

analyze table dim\_locatie compute statistics;

Table DIM\_LOCATIE analyzed.

EXPLAIN PLAN

SET STATEMENT\_ID = 'st\_dim\_partition'

FOR

select \* from dim\_locatie where oras = 'Timisoara'

SELECT plan\_table\_output

FROM

table(dbms\_xplan.display('plan\_table','st\_dim\_partition','serial'));

Plan hash value: 3299633729

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time | Pstart| Pstop |

| 0 | SELECT STATEMENT | | 1 | 36 | 274 (0)| 00:00:01 | | |

| 1 | PARTITION LIST SINGLE| | 1 | 36 | 274 (0)| 00:00:01 | KEY | KEY |

| 2 | TABLE ACCESS FULL | DIM\_LOCATIE | 1 | 36 | 274 (0)| 00:00:01 | 1 | 1 |

EXPLAIN PLAN

SET STATEMENT\_ID = 'st\_dim\_partition\_pruning'

FOR

select \* from dim\_locatie where oras in ('Timisoara', 'Constanta')

SELECT plan\_table\_output

FROM

table(dbms\_xplan.display('plan\_table','st\_dim\_partition\_pruning','serial'));

Plan hash value: 1889622376

-----------------------------------------------------------------------------------------------------

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time | Pstart| Pstop |

-----------------------------------------------------------------------------------------------------

| 0 | SELECT STATEMENT | | 2 | 54 | 547 (0)| 00:00:01 | | |

| 1 | PARTITION LIST INLIST| | 2 | 54 | 547 (0)| 00:00:01 |KEY(I) |KEY(I) |

| 2 | TABLE ACCESS FULL | DIM\_LOCATIE | 2 | 54 | 547 (0)| 00:00:01 |KEY(I) |KEY(I) |

-----------------------------------------------------------------------------------------------------

**Exercitiul 9: Optimizarea cererii SQL propusă în etapa de analiză**

--9 Formularea în limbaj natural a unei cereri SQL complexe care va fi optimizată în următoarea etapă,

--folosind tehnici specifice bazelor de date depozit

--Cerere SQL in limbaj natural:

--"Sa se afiseze pentru fiecare client suma medie a tuturor tranzactiilor efectuate la fiecare comerciant in parte."

select t.ID\_Client as ID\_CLIENT, c.Nume as NUME\_CLIENT, co.Nume as NUME\_COMERCIANT, AVG(ABS(t.SUMA)) as SUMA\_MEDIE

from FACT\_TRANZACTII t, DIM\_CLIENT c, DIM\_COMERCIANT co

where t.ID\_Client=c.ID\_Client

and t.ID\_Comerciant=co.ID\_Comerciant

group by t.ID\_CLIENT, c.Nume, co.Nume

order by t.ID\_Client

Graphical user interface

Description automatically generated with low confidence

--Crearea vizualizarii materializate

CREATE MATERIALIZED VIEW vm\_clienti\_comercianti\_suma\_medie\_tranzactie

BUILD IMMEDIATE

REFRESH COMPLETE

ON DEMAND

ENABLE QUERY REWRITE

AS

select t.ID\_Client as ID\_CLIENT, c.Nume as NUME\_CLIENT, co.Nume as NUME\_COMERCIANT, AVG(ABS(t.SUMA)) as SUMA\_MEDIE

from FACT\_TRANZACTII t, DIM\_CLIENT c, DIM\_COMERCIANT co

where t.ID\_Client=c.ID\_Client

and t.ID\_Comerciant=co.ID\_Comerciant

group by t.ID\_CLIENT, c.Nume, co.Nume

order by t.ID\_Client

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

--Colectare statistici

ANALYZE TABLE FACT\_TRANZACTII COMPUTE STATISTICS;

BEGIN DBMS\_STATS.GATHER\_TABLE\_STATS ('DW','VM\_CLIENTI\_COMERCIANTI\_SUMA\_MEDIE\_TRANZACTIE',

estimate\_percent=>20,block\_sample=>TRUE,cascade=>TRUE);

END;

Graphical user interface, application

Description automatically generated

--Setare parametrii pt rescriere

ALTER SESSION SET QUERY\_REWRITE\_ENABLED = TRUE;

ALTER SESSION SET QUERY\_REWRITE\_INTEGRITY = enforced;

--Afisare plan executie

EXPLAIN PLAN

SET STATEMENT\_ID ='st\_clienti\_comercianti\_suma'

FOR

select t.ID\_Client as ID\_CLIENT, c.Nume as NUME\_CLIENT, co.Nume as NUME\_COMERCIANT, AVG(ABS(t.SUMA)) as SUMA\_MEDIE

from FACT\_TRANZACTII t, DIM\_CLIENT c, DIM\_COMERCIANT co

where t.ID\_Client=c.ID\_Client

and t.ID\_Comerciant=co.ID\_Comerciant

group by t.ID\_CLIENT, c.Nume, co.Nume

order by t.ID\_Client;

SELECT plan\_table\_output

FROM table(dbms\_xplan.display('plan\_table','st\_clienti\_comercianti\_suma','serial'));

Text

Description automatically generated with medium confidence

--Rescriere cerere

select ID\_CLIENT, NUME\_CLIENT, NUME\_COMERCIANT, SUMA\_MEDIE

from VM\_CLIENTI\_COMERCIANTI\_SUMA\_MEDIE\_TRANZACTIE

**Graphical user interface, application, table

Description automatically generated**

**Exercitiul 10: Crearea rapoartelor cu complexitate diferită(la acest nivel vor fi scripturi SQL, fără reprezentare grafică)**

1. Sa se afle care sunt primele 3 luni ale anului 2022, in functie de volumul tranzactiilor(doar debitariile).

SELECT dim\_calendar.luna, SUM (fact\_tranzactii.suma) AS suma\_totala

FROM fact\_tranzactii

JOIN dim\_calendar ON fact\_tranzactii.id\_data = dim\_calendar.id\_data

WHERE dim\_calendar.anul = 2022 AND fact\_tranzactii.suma > 0

GROUP BY dim\_calendar.luna

ORDER BY suma\_totala DESC

FETCH FIRST 3 ROWS ONLY;

2. Sa se afle primele 3 luni si sumele aferente tranzactiilor effectuate dupa numarul de tranzactii efectuate doar cu POS din anul 2022(doar debitariile).

SELECT COUNT(\*) AS nr\_plati, luna, SUM (fact\_tranzactii.suma) AS suma\_totala

FROM fact\_tranzactii

JOIN dim\_calendar ON fact\_tranzactii.id\_data = dim\_calendar.id\_data

JOIN dim\_canal\_plata ON fact\_tranzactii.id\_canal\_plata = dim\_canal\_plata.id\_canal\_plata

WHERE dim\_canal\_plata.id\_canal\_plata = 1 AND fact\_tranzactii.suma > 0 AND dim\_calendar.anul = 2022

GROUP BY dim\_calendar.luna

ORDER BY nr\_plati DESC

FETCH FIRST 3 ROWS ONLY;

3. Sa se afle tipul canalului de plata a primelor 10 tranzactii dupa suma(doar debitariile).

SELECT dim\_canal\_plata.tip\_canal\_plata, fact\_tranzactii.suma

FROM fact\_tranzactii

JOIN dim\_canal\_plata ON fact\_tranzactii.id\_canal\_plata = dim\_canal\_plata.id\_canal\_plata

WHERE fact\_tranzactii.suma > 0

ORDER BY fact\_tranzactii.suma DESC

FETCH FIRST 10 ROWS ONLY;

4. Sa se afle comerciantul cu media cea mai mare a duratei tranzactiilor(doar debitariile).

SELECT dim\_comerciant.nume, AVG(fact\_tranzactii.durata) as medie\_durata

FROM dim\_comerciant

JOIN fact\_tranzactii ON fact\_tranzactii.id\_comerciant = dim\_comerciant.id\_comerciant

WHERE fact\_tranzactii.suma > 0

GROUP BY dim\_comerciant.nume

ORDER BY medie\_durata DESC

FETCH FIRST 3 ROWS ONLY;

5. Sa se afle numele clientului care a facut cea mai mare achizitie din toti anii(doar debitariile).

SELECT dim\_client.nume, MAX(fact\_tranzactii.suma) as suma\_maxima

FROM fact\_tranzactii

JOIN dim\_client ON fact\_tranzactii.id\_client = dim\_client.id\_client

WHERE fact\_tranzactii.suma > 0

GROUP BY dim\_client.nume

ORDER BY suma\_maxima DESC

FETCH FIRST 3 ROWS ONLY;