

UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
SPECIALIZAREA INFORMATICĂ

**DATA WAREHOUSE & BUSINESS
INTELLIGENCE
- PROIECT -**

Autor

Tiberius Coman
George Banica
Daniela Alexandra Constantin
Sebastian Alexandru Velciu

Profesor titular

Lect. Dr. GABRIELA MIHAI

GRUPA 505
ANUL II MASTER, SEMESTRUL I

UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
SPECIALIZAREA INFORMATICĂ

**DATA WAREHOUSE & BUSINESS
INTELLIGENCE
- ANALIZA -**

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

Prezentarea concisă a bazei de date (utilizarea ei) :

Sistemul de plăți permite crearea de conturi și carduri aferente pentru persoane fizice pe care îi vom numi generic clienți. Aceștia pot face cumpărături la diferiți comercianți care sunt înregistrați în prealabil în sistem. Comercianții trebuie să fie la rândul lor clienți ai aplicației și necesită înregistrarea unui dispozitiv POS prin care clienții pot face cumpărături. De asemenea în cazul în care comercianții nu dețin un POS fizic pot înregistra site-ul pe care îl folosesc pentru a comercializa produsele. Acest sistem este cu circuit închis în sensul că, nu se acceptă tranzacții pentru clienți ale căror carduri nu sunt cunoscute sau comercianți ale căror dispozitive nu sunt regăsite în sistem.

Exercitiul 2

a) Realizarea diagramei entitate-relație (ERD).

În urma analizei modelului s-au creat următoarele relații:

CLIENT(id_client#, nume, tip_client, data_inscriere, data_incetare);

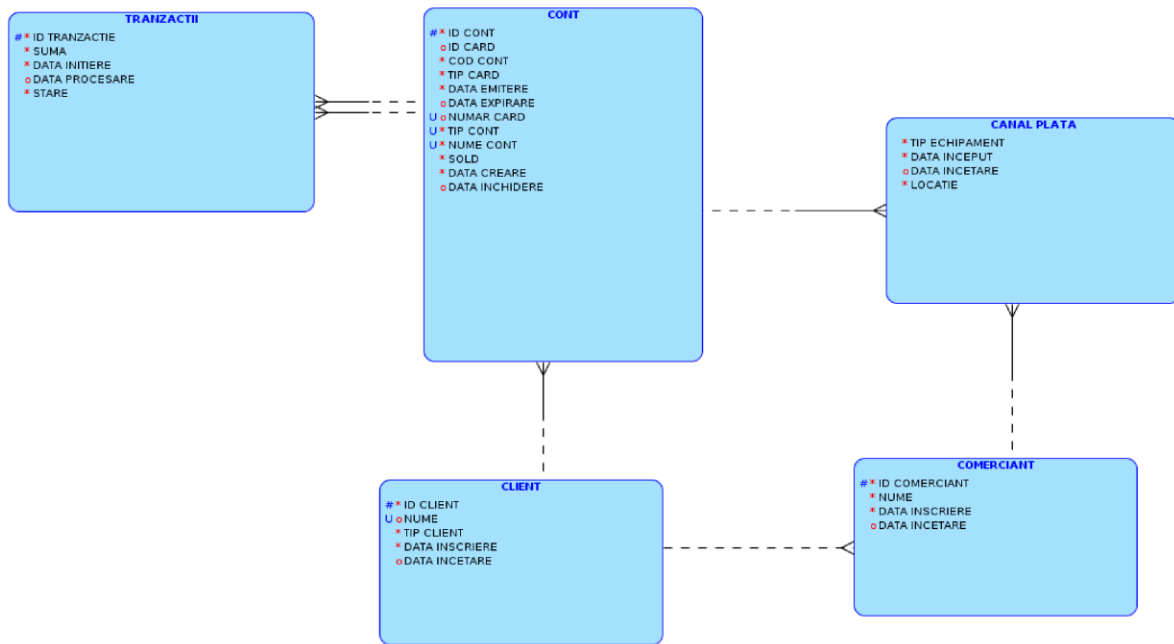
CONT(id_cont#, id_card#, tip_card, data_emitere, data_expirare, numar_card, tip_cont, nume_cont, sold, data_creare, data_inchidere);

TRANZACTII(id_tranzactie#, suma, cod_cont_debitor, cod_cont_creditor, data_initiere, data_procesare, stare);

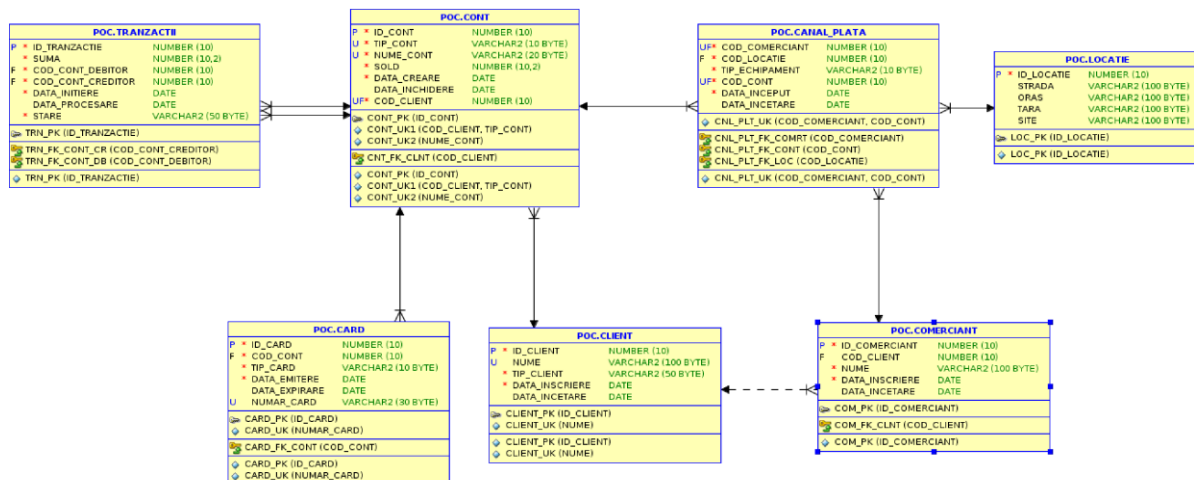
COMERCIANT(id_comerciant#, cod_client, nume, data_inscriere, data_incetare);

CANAL_PLATA(cod_comerciant, tip echipament, cod_cont, data_inceput, data_incetare, locatie);

b) Transformarea sistemului conceptual într-un design logic, subliniind relațiile dintre
tabele, cheile primare și străine (externe).



c) Implementarea tabelor în Oracle, folosind chei primare, constrângeri de referință și
domeniu



Exercitiul 3

1. Proiectarea logica de depozitului de date

Business Intelligence pentru sistemele de plată poate reprezenta utilizarea datelor pentru a obține informații despre sistemele de plată și operațiunile acestora. Scopul este de a înțelege performanța și eficiența sistemelor de plată și de a lua decizii bazate pe date care pot îmbunătăți experiența generală pentru utilizatori și pot crește profitabilitatea sistemului.

Unele aplicații comune ale business intelligence în sistemele de plată includ:

- **Optimizarea plăților:** sistemul de BI poate fi utilizat pentru a analiza datele tranzacțiilor pentru a identifica blocajele și ineficiența proceselor de plată și pentru a face îmbunătățiri bazate pe date care cresc eficiența și reduc timpul de procesare.
- **Analiza comportamentului clienților:** sistemul de BI poate fi utilizat pentru a analiza comportamentul și preferințele clienților, pentru a înțelege nevoile acestora și pentru a face recomandări bazate pe date pentru noi produse și servicii;
- **Gestionarea riscurilor:** Business intelligence poate fi utilizat pentru a analiza datele de plată pentru a identifica și gestiona riscul, inclusiv riscul de credit, riscul de piață și riscul operațional;
- **Tendențele de plată și forecasting:** datele și analizele pot fi utilizate pentru a înțelege tendințele de plată și pentru a face predicții despre plățile și veniturile viitoare.

Astfel, ne propun ca modelul nostru să poată permite:

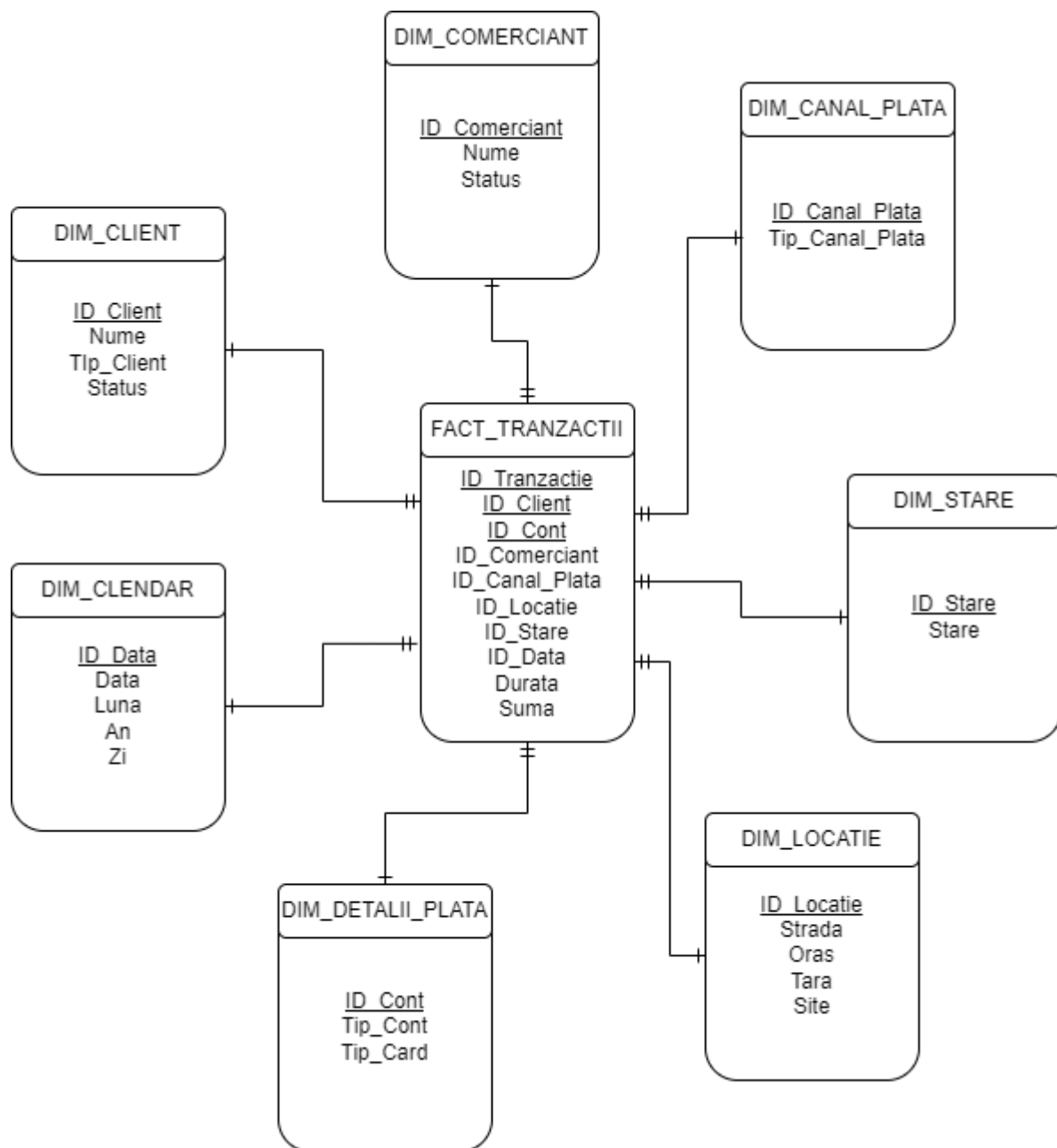
- Analiza sumelor tranzacționate în funcție de clienți, comercianți, modalități de plată și locații sau timp;
- Analiza duratelor de soluționare a transferurilor;
- Analiza numărului de transferuri eșuate.

Pentru a obține abilitatea de a analiza aceste metrice, vom propune următorul model pentru depozitul de date:

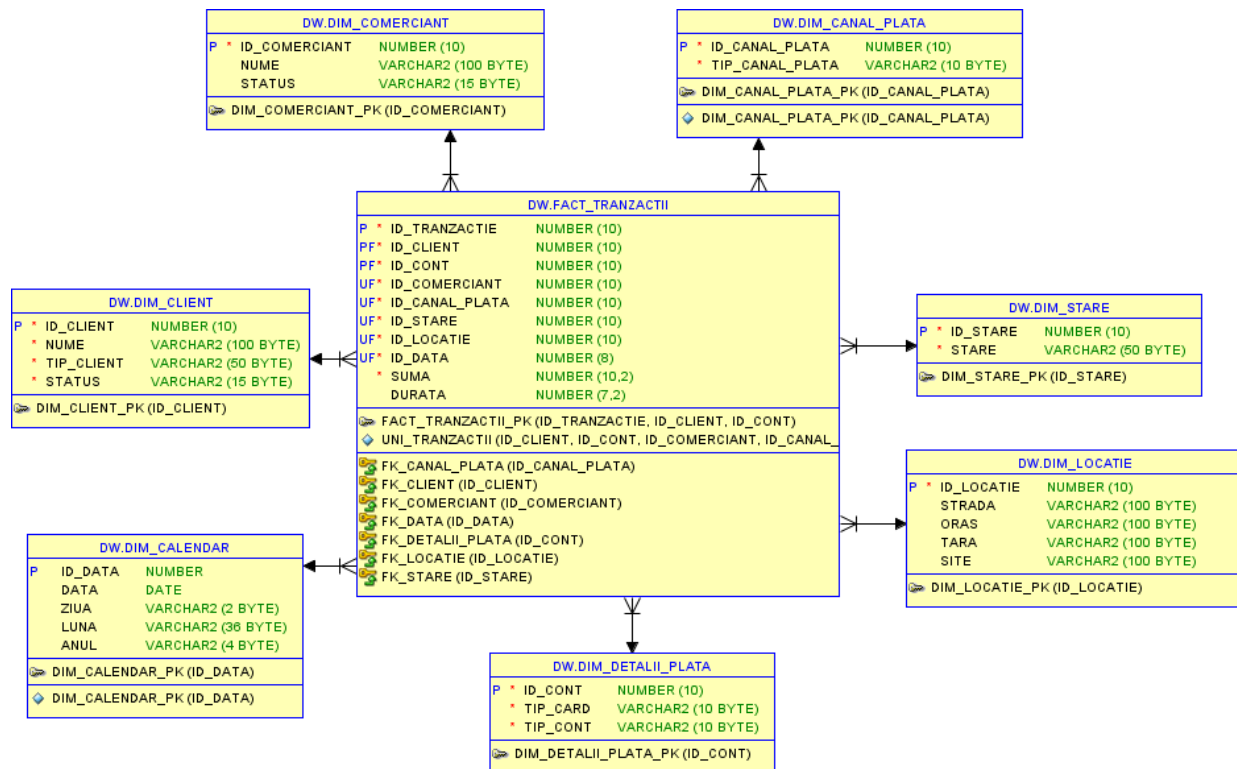
- **FACT_TRANZACȚII:**
 - **ID_Transactie** - câmp ce provine din tabela TRANZACȚII, ne va ajuta să identificăm tranzacția în cadrul careia s-au realizat plata și încasarea sumelor;
 - **ID_Client** - câmp ce provine din tabela CLIENT, stabilește o legătură cu DIM_CLIENT și ne va ajuta să analizăm datele în funcție de clienți;
 - **ID_Cont** - câmp ce provine din tabela CONT, stabilește o legătură cu DIM_DETALII_PLATA și ne va ajuta să analizăm datele în funcție de tipul conturilor și tipurile de carduri.

- **ID_Comerciant** - câmp ce provine din tabela COMERCIENT și stabilește legătura cu DIM_COMERCIENT și ne va ajuta sa analizăm datele în funcție de comercianti;
- **ID_Canal_Plata** - câmp generat în funcție de modalitatea de plata, codul 1 pentru plata la POS, codul 2 pentru plăți ONLINE pentru coloana Tip_Echipament din tabela CANAL_DE_PLATA, stabilește legătura cu DIM_CANAL_DE_PLATA;
- **ID_Locatie** - câmp ce provine din tabla LOCATIE și stabilește legătura cu DIM_LOCATIE și ne va ajuta sa analizăm datele în funcție de locurile unde s-au făcut transferurile.
- **ID_Stare** - câmp generat în funcție de starea transferului, codul 1 pentru transferuri INIȚIATE, codul 2 pentru transferuri PROCESATE, codul 3 pentru transferuri ERONATE, pe baza coloanei STARE din table TRANZACȚII, stabilește legătura cu DIM_STARE.
- **ID_Data** - câmp generat pe baza coloanei DATA_INIȚIERE din tabela TRANZACȚII, în format "YYYYMMDD", va fi folosit pentru a stabili legătura cu DIM_CALENDAR și ne va permite sa analizăm datele în funcție de timp.
- **Suma** - câmp ce provine din tabela TRANZACȚII, va avea valoare negativa pentru plăți și valoare pozitivă pentru încasări;
- **Durata** - câmp calculat pe baza coloanelor DATA_INIȚIERE și DATA_PROCESARE din tabela TRANZACȚII, reprezentând durata în minute de la inițiere până la procesare.
- **DIM_CLIENT:**
 - **ID_Client** - câmp ce provine din tabela CLIENT, este identificatorul unic pentru un client al sistemului de plata;
 - **Nume** - câmp ce provine din tabela CLIENT, reprezintă numele clientului și va permite construirea unor rapoarte sau vizualizări în funcție de clienți.
 - **Tip_Client** - câmp ce provine din tabela CLIENT, reprezintă statutul juridic al clientului (persoana fizică sau persoana juridică), va permite construirea de vizualizări în funcție de această clasificare.
 - **Status** - câmp calculat pe baza coloanelor Data_Inscriere și Data_Incetare din tabela CLIENT și va oferi informații despre statutul actual al clientului, dacă este client actual sau a încetat utilizarea sistemului de plata.
- **DIM_COMERCIENT:**
 - **ID_Comerciant** câmp ce provine din tabela COMERCIENT, este identificatorul unic pentru un comerciant al sistemului de plata;
 - **Nume** - câmp ce provine din tabela COMERCIENT, reprezintă numele comerciantului și va permite construirea unor rapoarte sau vizualizări în funcție de comercianti;
 - **Status** - câmp calculat pe baza coloanelor Data_Inscriere și Data_Incetare din tabela COMERCIENT și va oferi informații despre statutul actual al clientului, dacă este client actual sau a încetat utilizarea sistemului de plata.
- **DIM_CANAL_PLATA:**
 - **ID_Canal_Plata** - câmp generat pe baza coloanei Tip_Echipament din tabela CANAL_PLATA cu scopul de a deveni un identificator unic pentru metodele de plata în DIM_CANAL_PLATA. În același mod, pentru câmpul ID_Canal_Plata este populat în FACT_TRANZACTII, astfel încât să putem stabili o legătură dintre cele două tabele, pentru a permite analiza datelor în funcție de modalitățile de plata;

- **Tip_Canal_Plata** - câmp ce provine din tabela CANAL_PLATA, reprezinta modalitatea de plata si va permite construirea unor rapoarte sau vizualizari în funcție de aceasta.
- **DIM_STARE:**
 - **ID_Stare** - câmp generat pe baza coloanei Stare din tabela TRANZACTII cu scopul de a deveni un identificator unic pentru starea transferurilor în DIM_STARE. În același mod, pentru câmpul ID_Stare este populat în FACT_TRANZACTII, astfel incat sa putem stabili o legatura dintre cele doua tabele, pentru a permite analiza datelor în funcție de stările transferurilor;
 - **Stare** - câmp ce provine din tabela TRANZACTII, reprezinta starea unui transfer si va permite construirea unor rapoarte sau vizualizari în funcție de aceasta.
- **DIM_LOCAȚIE:**
 - **ID_Locatie** - câmp ce provine din tabela LOCAȚIE, este identificatorul unic pentru un comerciant al sistemului de plata;
 - **Oraș, Țara, Strada, Site** - câmpuri ce provin din tabela LOCAȚIE fără a aplica transformări asupra datelor;
- **DIM_DETALII_PLATA:**
 - **ID_Cont** - câmp ce provine din tabela CONT, este identificatorul unic pentru un cont în cadrul sistemului de plata și ne va permite sa stabilim o legatura cu FACT_TRANZACTII pentru analiza datelor în funcție de tipurile conturilor și cardurilor folosite pentru transferuri;
 - **Tip_Cont** - câmp ce provine din tabela CONT, reprezinta tipul contului din / către a fost făcut transferul;
 - **Tip_Card** - câmp ce provine din tabela CARD, reprezinta tipul cardului din / către a fost făcut transferul.
- **DIM_CALENDAR:**
 - **ID_Data** - câmp generat ce contine data in format "YYYYMMDD", prin care putem identifica unic o data și prin care putem stabili o legătură cu FACT_TRANZACTII;
 - **Data** - câmp generat ce contine data în format "DD/MM/YYYY"
 - **Luna** - câmp generat ce contine luna în format "Month";
 - **An** - câmp generat ce contine anul în format "YYYY";
 - **Zi** - câmp generat ce contine anul în format "DD";



Exercitiul 4



Exercitiul 5

C1: Avem tabela de fapte FACT_TRANZACTII având următoarele atribute: ID#, ID_Transactie, ID_Client, ID_Cont, ID_Comerciant, ID_Canal_Plata, ID_Stare, ID_Locatie, ID_Data, Durata, Suma. Coloanele ID_Client, ID_Cont, ID_Comerciant, ID_Canal_Plata, ID_Stare, ID_Locatie si ID_Data identifică în mod unic o înregistrare din această tabelă. Un mod potrivit pentru un depozit de date de a defini aceasta constrângere de unicitate este prezentat in documentul „Sursa”, punctul 5 (cod referință c1).

Justificare: Am ales aceasta abordare deoarece opțiunea DISABLE ne scutește de costurile unui index unic (care nu a fost creat), dar in același timp asigura unicitatea datelor. Având in vedere faptul ca tabela FACT_TRANZACTII este actualizata printr-un proces ETL, perechea de constrângeri DISABLE VALIDATE nu incomodează: nu mai sunt permise operații LMD, constrângerea nu este activa, dar s-a realizat validarea datelor.

C2: Într-o schemă stea, constrângerile de cheie externă validează relațiile dintre tabela de fapte și tabelele dimensiune. Tabela de fapte FACT_TRANZACTII conține atributele ID_Client, ID_Cont, ID_Comerciant, ID_Canal_Plata, ID_Stare si ID_Locatie care referă atributele cu același nume din tabelele dimensiune DIM_CLIENT, DIM_DETALII_PLATA, DIM_COMERCIANT, DIM_CANAL_PLATA, DIM_STARE, respectiv DIM_LOCATIE. Pentru depozitele de date, o maniera potrivita de a defini constrângerile de cheie externa este exemplificata in documentul „ Sursa ”, punctul 5 (cod referință c2-c7).

Justificare: Atunci când se încărca un depozit de date, de obicei nu este nevoie de validarea constrângerile, deoarece datele au fost încărcate în bulk dintr-o zonă de pregătire în care a fost efectuată toată validarea datelor. Astfel, am salvat timp prețios declarând constrângerile fără a le valida. Crearea unei constrângeri în NOVALIDATE este imediată și nu depinde de dimensiunea tabelului. Constrângerea garantează că nu se vor introduce rânduri în tabela de fapte fără un rând corespunzător în dimensiune. Cu toate acestea, deoarece Oracle nu a validat rezultatul în sine, putem seta atributul RELY. Pe lângă RELY NOVALIDATE, implicit, constrângerea este definită cu opțiunea ENABLE, adică constrângerea este activă și va determina verificarea integrității datelor în cazul operațiilor LDM realizate asupra tabelelor implicate.

C3: Tabela de fapte FACT_TRANZACTII conține atributul ID_Data care referă atributul cu același nume din tabela dimensiune DIM_CALENDAR. Pentru depozitele de date, o alternativă de definire a constrângerii de cheie externă între cele două tabele se găsește în documentul „Sursa”, punctul 5, sub codul de referință c8.

Justificare: Opțiunea RELY permite optimizatorului să utilizeze constrângerea pentru a determina un plan optim, fără consum de resurse.

Exercitiul 6

Crearea indexului global:

```
drop index calendar_idx;
create index calendar_idx on dim_calendar (luna) global partition by hash(luna) partitions 12;
```

Script Output x

Task completed in 0.301 seconds

Index CALENDAR_IDX dropped.

Index CALENDAR_IDX created.

Consultare dictionarul datelor pentru verificarea crearii partițiilor pe index:

```
select * from user_ind_partitions where index_name = 'CALENDAR_IDX';
```

INDEX_NAME	COMPOSITE	PARTITION_NAME	SUBPARTITION_COUNT	HIGH_VALUE	HIGH_VALUE_LENGTH	PARTITION_POSITION	STATUS	TABLESPACE_NAME	PCT_FREE	INI_TRANS	MAX_TRANS	INITIAL_EXTENT
CALENDAR_IDX NO		SYS_P609	0 (null)	0	0	1	USABLE	USERS	10	2	255	65536
CALENDAR_IDX NO		SYS_P610	0 (null)	0	0	2	USABLE	USERS	10	2	255	65536
CALENDAR_IDX NO		SYS_P611	0 (null)	0	0	3	USABLE	USERS	10	2	255	65536
CALENDAR_IDX NO		SYS_P612	0 (null)	0	0	4	USABLE	USERS	10	2	255	65536
CALENDAR_IDX NO		SYS_P613	0 (null)	0	0	5	USABLE	USERS	10	2	255	65536
CALENDAR_IDX NO		SYS_P614	0 (null)	0	0	6	USABLE	USERS	10	2	255	65536
CALENDAR_IDX NO		SYS_P615	0 (null)	0	0	7	USABLE	USERS	10	2	255	65536
CALENDAR_IDX NO		SYS_P616	0 (null)	0	0	8	USABLE	USERS	10	2	255	65536
CALENDAR_IDX NO		SYS_P617	0 (null)	0	0	9	USABLE	USERS	10	2	255	65536
CALENDAR_IDX NO		SYS_P618	0 (null)	0	0	10	USABLE	USERS	10	2	255	65536
CALENDAR_IDX NO		SYS_P619	0 (null)	0	0	11	USABLE	USERS	10	2	255	65536
CALENDAR_IDX NO		SYS_P620	0 (null)	0	0	12	USABLE	USERS	10	2	255	65536

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

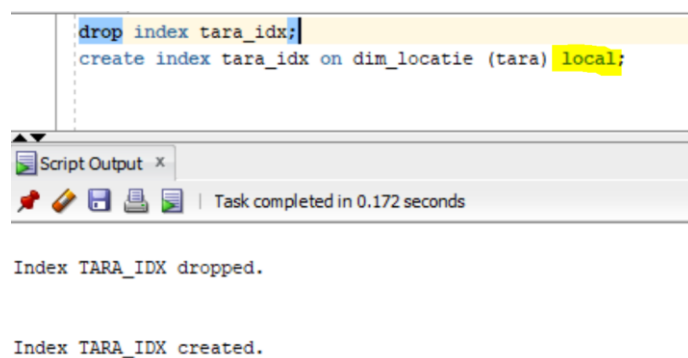
table(dbms_xplan.display('plan_table', 's1_index_global','serial'));
```

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 :



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

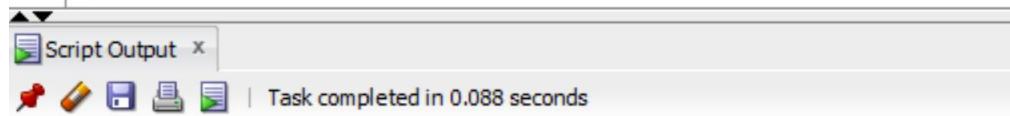
```
-----
```

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:

```
drop index dim_client_bmp;  
CREATE bitmap INDEX dim_client_bmp ON dim_client (tip_client);
```



```
Index DIM_CLIENT_BMP dropped.
```

```
INDEX DIM_CLIENT_BMP created.
```

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
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_COMERCIENT	1	11	0 (0)	00:00:01	

Exercitiul 7

Pentru a putea stabili care vor fi obiectele dimensiune pe care le vom crea, vom incepe prin a mentiona ce sunt acestea si la ce ne pot ajuta.

Dimensiunea este o structură care clasifică datele pentru a le permite utilizatorilor să răspundă la întrebările de afaceri. Dimensiunile utilizate în mod obișnuit sunt clienții, produsele și timpul. De exemplu, fiecare canal de vânzări al unui comerciant poate colecta și stoca date privind vânzările. Managementul lanțului de vânzare poate construi un depozit de date pentru a analiza vânzările produselor sale în toate magazinele de-a lungul timpului și pentru a răspunde la întrebări precum:

- Care este efectul promovării unui produs comparativ cu unul care nu este promovat?
- Care sunt vânzările unui produs înainte și după o promoție?
- Cum afectează o promovare diferitele canale de distribuție?

În baza de date Oracle, informațiile dimensionale în sine sunt stocate într-un tabel de dimensiuni. În plus, obiectul dimensiune ajută la organizarea și gruparea informațiilor dimensionale în ierarhii. Aceasta reprezintă relații naturale 1:n între coloane sau grupuri de coloane (nivelurile unei ierarhii) care nu pot fi reprezentate cu condiții de constrângere. Urcarea unui nivel în ierarhie se numește acumularea datelor și coborârea la un nivel în ierarhie se numește detaliere a datelor.

Obiectele dimensiune nu trebuie definite neapărat. Cu toate acestea, creerea lor poate aduce beneficii semnificative, deoarece ajută la rescrierea interogărilor să efectueze tipuri mai complexe de rescriere.

În cazul acestui proiect au fost identificate 2 obiecte dimensiune: locație, timp. Ambele dimensiuni sunt normalizate (nu este stocată în mai mult de un tabel, sau altfel zis, nu are referințe la alte tabele).

Aceste obiecte vor avea următoarele nivele și ierarhie:

- Locație: locație_id -> stradă -> oraș -> țară
- Timp: data_id -> dată -> anul

Pentru dimensiunea locație, clauza `ATTRIBUTE ... DETERMINES` se referă la a pune în legătură `locație_id` cu celelalte câmpuri, și anume: `stradă`, `oraș`, `țară` și `site`. Aceasta este o determinare unidirecțională. Este doar garantat că pentru un anumit id, o să fie găsită exact o valoare potrivită pentru `stradă`, `oraș`, `țară` și `site`. Însă nu se poate determina un id pentru un anumit oraș spre exemplu.

Similar este și în cazul dimensiunii timp, un singur lucru diferă aici, și anume faptul că putem crea mai multe relații prin atribute de această dată. Motivul pentru care putem face acest lucru este faptul că atât `data_id` cât și `data` sunt unice în tabelul `dim_calendar`. Prin urmare, câmpurile: `id_data`, `data`, `ziua`, `luna`, `anul` pot fi determinate în mod direct atât prin `data_id` cât și prin `data`.

Nu in ultimul rand, informațiile unui obiect dimensiune sunt doar declarative și nu sunt impuse de baza de date. Dacă relațiile descrise de dimensiuni sunt incorecte, pot apărea rezultate incorecte. Prin urmare, ar trebui să fie verificate periodic relațiile specificate de CREATE DIMENSION utilizând procedura DBMS_DIMENSION.VALIDATE_DIMENSION.

Exercitiul 8

Definire partitie pe tabela de fapte :

```

ID_Data NUMBER(8) NOT NULL,
Suma NUMBER(10, 2) NOT NULL,
Durata NUMBER(7, 2) DEFAULT NULL
partition by range(id_data)
interval(1)
(partition partitie_initiala values less than (20220101));

```

Table FACT_TRANZACTII dropped.

Table FACT_TRANZACTII created.

Partitionarea te tip interval este o extensie a partitionarii de tip range. Partitiile noi sunt create automat de catre SGBD. atunci cand coloana pe care s-a definit regula de partitionare “depaseste” toate valorile. Cel puțin 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_NAME	COMPOSITE	PARTITION_NAME	SUBPARTITION_COUNT	HIGH_VALUE	HIGH_VALUE_LENGTH	PARTITION_POSITION	TABLESPACE_NAME	PCT_FREE
1 FACT_TRANZACTII NO		PARTITIE_INITIALA		0 20220101	8	1 USERS		10
2 FACT_TRANZACTII NO		SYS_P541		0 20221232	8	2 USERS		10
3 FACT_TRANZACTII NO		SYS_P546		0 20230124	8	3 USERS		10
4 FACT_TRANZACTII NO		SYS_P545		0 20230126	8	4 USERS		10
5 FACT_TRANZACTII NO		SYS_P544		0 20230129	8	5 USERS		10
6 FACT_TRANZACTII NO		SYS_P542		0 20230130	8	6 USERS		10
7 FACT_TRANZACTII NO		SYS_P543		0 20230131	8	7 USERS		10


```
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 :

```
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)  
);
```

Script Output x Query Result x Explain Plan x Explain Plan 1 x Query Res

Task completed in 0.12 seconds

Table DIM_LOCATIE created.

Verificarea crearii partiilor prin consultarea dictionarului datelor.

```
select * from user_tab_partitions;
```

TABLE_NAME	COMPOSITE	PARTITION_NAME	SUBPARTITION_COUNT	HIGH_VALUE	HIGH_VALUE_LENGTH	PARTITION_POSITION	TABLESPACE_NAME	PCT_FREE	PCT_USED	INI_TRANS
5 DIM_LOCATIE	NO	TM		0 'Timisoara'	11	1	USERS	10	(null)	1
6 DIM_LOCATIE	NO	B		0 'Bucuresti'	11	2	USERS	10	(null)	1
7 DIM_LOCATIE	NO	CT		0 'Constanta'	11	3	USERS	10	(null)	1
8 DIM_LOCATIE	NO	BV		0 'Brasov'	8	4	USERS	10	(null)	1
9 DIM_LOCATIE	NO	NEDEFINIT		0 default	7	5	USERS	10	(null)	1

```
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

Pentru acest exercitiu vom declara urmatoarea cerere in limbaj natural:

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

Planul este sa cream o vizualizare materializata cu join-uri si agregate care sa corespunda cererii noastre. Mai apoi vom colecta statisticile si vom seta parametrii pentru rescrierea cererii.

Exercitiul 10

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

```
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 efectuate dupa numarul de tranzactii efectuate doar cu POS din anul 2022(doar debitarile).

```
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 debitarile).

```
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 debitarile).

```
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 debitarile).

```
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;
```

UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
SPECIALIZAREA INFORMATICĂ

**DATA WAREHOUSE & BUSINESS
INTELLIGENCE
- SCURSA -**

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_creatie 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(num, tip_client, data_inscriere, data_incetare) values ('Popescu
Ion', 'PF', sysdate - 10, null);

insert into client(num, tip_client, data_inscriere, data_incetare) values ('Alexandru
Petre', 'PF', sysdate - 5, null);

insert into client(num, tip_client, data_inscriere, data_incetare) values ('Auchan',
'PJ', sysdate - 100 , null);

insert into client(num, tip_client, data_inscriere, data_incetare) values
('Carrefour', 'PJ', sysdate - 200, null);

insert into client(num, tip_client, data_inscriere, data_incetare) values ('George
Banica', 'PF', sysdate - 1, null);

```

```

insert into client(num, tip_client, data_inscriere, data_incetare) values ('Mega
Image', 'PJ', sysdate - 300, null);

insert into client(num, tip_client, data_inscriere, data_incetare) values ('Camil
Ressu', 'PF', sysdate - 300, sysdate);

insert into client(num, tip_client, data_inscriere, data_incetare) values ('Mihail
Sadoveanu', 'PF', sysdate, null);

insert into client(num, tip_client, data_inscriere, data_incetare) values ('Alexandru
Vlahuta', 'PF', sysdate-1, null);

insert into client(num, tip_client, data_inscriere, data_incetare) values ('Penny',
'PJ', sysdate-500, null);

insert into client(num, tip_client, data_inscriere, data_incetare) values ('Glovo',
'PJ', sysdate-50, null);

commit;

```

```

insert into comerciant(num, cod_client, data_inscriere) values ('Auchan',
(select id_client from client where num = 'Auchan'), sysdate - 100);

insert into comerciant(num, cod_client, data_inscriere) values ('Carrefour',
(select id_client from client where num = 'Carrefour'), sysdate - 200);

insert into comerciant(num, cod_client, data_inscriere) values ('Mega Image',
(select id_client from client where num = 'Mega Image'), sysdate - 300);

insert into comerciant(num, cod_client, data_inscriere) values ('Penny',
(select id_client from client where num = 'Penny'), sysdate - 500);

insert into comerciant(num, cod_client, data_inscriere) values ('Glovo',
(select id_client from client where num = 'Glovo'), sysdate - 50);

commit;

```

```

insert into cont(tip_cont, sold, num_cont, cod_client) values ('CURENT', 25.39,
'RO49POCB1V3100759384', (select id_client from client where num = 'Popescu Ion'));

insert into cont(tip_cont, sold, num_cont, cod_client) values ('CURENT', 120.45,
'RO49POCD2X7100869495', (select id_client from client where num = 'Alexandru
Petre'));

insert into cont(tip_cont, sold, num_cont, cod_client) values ('CURENT', 67.85,
'RO49POCG5Z4200292728', (select id_client from client where num = 'George Banica'));

insert into cont(tip_cont, sold, num_cont, cod_client) values ('CURENT', 143.20,
'RO49POCE6T5300303839', (select id_client from client where num = 'Camil Ressu'));

insert into cont(tip_cont, sold, num_cont, cod_client) values ('CURENT', 70.50,
'RO49POCJ5S5300303839', (select id_client from client where num = '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
```

```

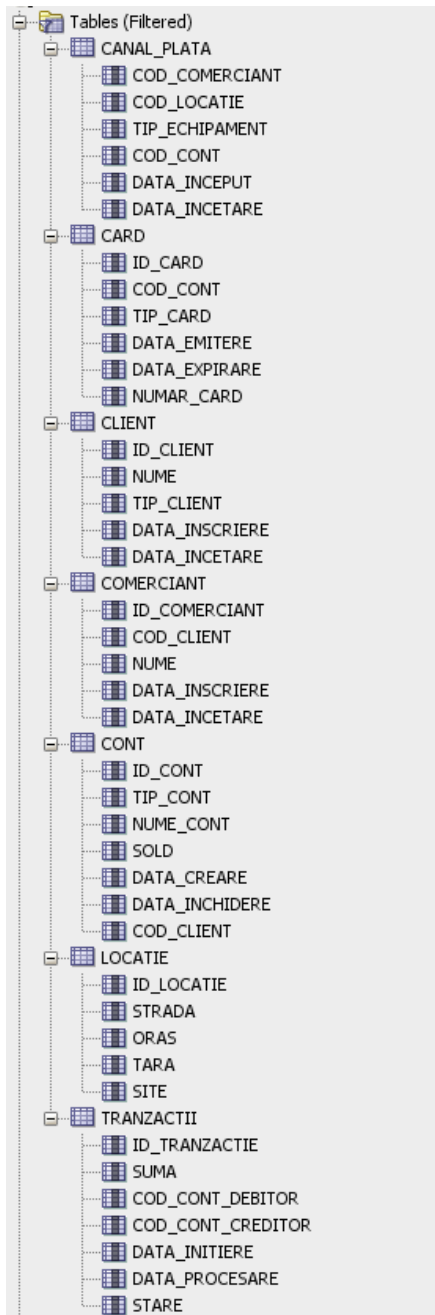
(15.0, (select id_cont from cont, client where cod_client = id_client and nume =
'George Banica'), (select id_cont from cont, client where cod_client = id_client and
nume = 'Mega Image'),

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)

```
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_ume_cont IN VARCHAR2,
valoare_sold IN NUMBER, valoare_cod_client IN NUMBER) IS

BEGIN

    INSERT INTO cont(tip_cont, ume_cont, sold, cod_client)

    VALUES (valoare_tip_cont, valoare_ume_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;

```

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;

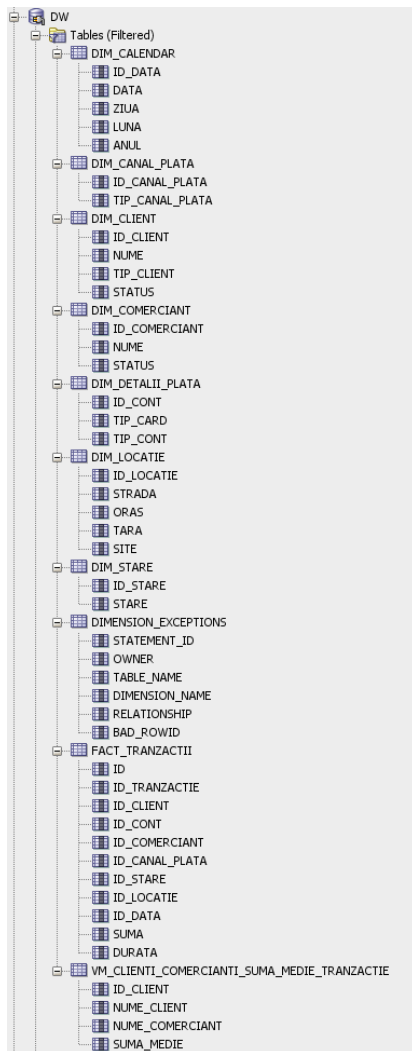
CREATE TABLE FACT_TRANZACTII (
    ID_Transactie NUMBER(10) NOT NULL,
    ID_Client NUMBER(10) NOT NULL,
    ID_Cont NUMBER(10) NOT NULL,
    ID_Comerciant NUMBER(10) NOT NULL,
    ID_Canal_Plata NUMBER(10) NOT NULL,
    ID_Stare NUMBER(10) NOT NULL,
    ID_Locatie NUMBER(10) NOT NULL,
    ID_Data NUMBER(8) NOT NULL,
    Suma NUMBER(10, 2) NOT NULL,
    Durata NUMBER(7, 2) DEFAULT NULL,
    PRIMARY KEY(ID_Transactie, 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_Transactie,
        ID_Client,
        ID_Cont,
        ID_Comerciant,
        ID_Canal_Plata,
        ID_Stare,
        ID_Locatie,
        ID_Data,
        Suma,
        Durata
    )

```

```

SELECT
    ID_Transactie,
    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:

19	UNI_TRANZACTII	Unique	(null)	(null)	(null)	(null)	(null)	(null)	DISABLED	NOT DEFERRABLE	VALIDATED	USER NAME	(null)	(null)	29-01-2023	(null)	(null)
1	FK_CANAL_PLATA	Foreign_Key	(null)	DW	DIM_CANAL_PLATA	DIM_CANAL_PLATA_PK	NO ACTION	ENABLED	NOT DEFERRABLE	NOT VALIDATED	USER NAME	(null)	(null)	29-01-2023	(null)	(null)	
2	FK_CLIENT	Foreign_Key	(null)	DW	DIM_CLIENT	DIM_CLIENT_PK	NO ACTION	ENABLED	NOT DEFERRABLE	NOT VALIDATED	USER NAME	(null)	(null)	29-01-2023	(null)	(null)	
3	FK_COMERCANT	Foreign_Key	(null)	DW	DIM_COMERCANT	DIM_COMERCANT_PK	NO ACTION	ENABLED	NOT DEFERRABLE	NOT VALIDATED	USER NAME	(null)	(null)	29-01-2023	(null)	(null)	
4	FK_DATA	Foreign_Key	(null)	DW	DIM_CALENDAR	DIM_CALENDAR_PK	NO ACTION	DISABLED	NOT DEFERRABLE	NOT VALIDATED	USER NAME	(null)	RELY	29-01-2023	(null)	(null)	
5	FK_DETALII_PLATA	Foreign_Key	(null)	DW	DIM_DETALII_PLATA	DIM_DETALII_PLATA_PK	NO ACTION	ENABLED	NOT DEFERRABLE	NOT VALIDATED	USER NAME	(null)	(null)	29-01-2023	(null)	(null)	
6	FK_LOCATIE	Foreign_Key	(null)	DW	DIM_LOCATIE	DIM_LOCATIE_PK	NO ACTION	ENABLED	NOT DEFERRABLE	NOT VALIDATED	USER NAME	(null)	(null)	29-01-2023	(null)	(null)	
7	FK_STARE	Foreign_Key	(null)	DW	DIM_STARE	DIM_STARE_PK	NO ACTION	ENABLED	NOT DEFERRABLE	NOT VALIDATED	USER NAME	(null)	(null)	29-01-2023	(null)	(null)	

Oracle 19 db

DB

DW

DIM_CALENDAR

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	CONSTRAINT_NAME	CONSTRAINT...	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATUS	DEFERRABLE	VALIDATED	GENERATED	BAD	RELY	LAST_CHANGE	INDEX_OWNER	INDEX_NAME	INVALID	VIEW_REL
1	DIM_CALENDAR_PK	Primary_Key	(null)	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE	NOT VALIDATED	USER NAME	(null)	RELY	29-01-2023	DW	DIM_CALENDAR_PK	(null)	(null)

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:

```
drop index calendar_idx;  
create index calendar_idx on dim_calendar (luna) global partition by hash(luna) partitions 12;
```

Script Output x

Task completed in 0.301 seconds

Index CALENDAR_IDX dropped.

Index CALENDAR_IDX created.

Consultare dictionarul datelor pentru verificarea crearii partiilor pe index:

```
select * from user_ind_partitions where index_name = 'CALENDAR_IDX';
```

pt Output x Query Result x

SQL All Rows Fetched: 12 in 0.11 seconds

INDEX_NAME	COMPOSITE	PARTITION_NAME	SUBPARTITION_COUNT	HIGH_VALUE	HIGH_VALUE_LENGTH	PARTITION_POSITION	STATUS	TABLESPACE_NAME	PCT_FREE	INI_TRANS	MAX_TRANS	INITIAL_EXTENT
CALENDAR_IDX NO		SYS_P609	0 (null)		0	1	USABLE	USERS	10	2	255	65536
CALENDAR_IDX NO		SYS_P610	0 (null)		0	2	USABLE	USERS	10	2	255	65536
CALENDAR_IDX NO		SYS_P611	0 (null)		0	3	USABLE	USERS	10	2	255	65536
CALENDAR_IDX NO		SYS_P612	0 (null)		0	4	USABLE	USERS	10	2	255	65536
CALENDAR_IDX NO		SYS_P613	0 (null)		0	5	USABLE	USERS	10	2	255	65536
CALENDAR_IDX NO		SYS_P614	0 (null)		0	6	USABLE	USERS	10	2	255	65536
CALENDAR_IDX NO		SYS_P615	0 (null)		0	7	USABLE	USERS	10	2	255	65536
CALENDAR_IDX NO		SYS_P616	0 (null)		0	8	USABLE	USERS	10	2	255	65536
CALENDAR_IDX NO		SYS_P617	0 (null)		0	9	USABLE	USERS	10	2	255	65536
CALENDAR_IDX NO		SYS_P618	0 (null)		0	10	USABLE	USERS	10	2	255	65536
CALENDAR_IDX NO		SYS_P619	0 (null)		0	11	USABLE	USERS	10	2	255	65536
CALENDAR_IDX NO		SYS_P620	0 (null)		0	12	USABLE	USERS	10	2	255	65536

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
```

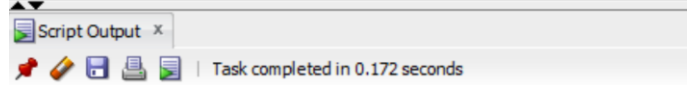
```
table(dbms_xplan.display('plan_table', 's1_index_global','serial'));
```

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 :

```
drop index tara_idx;
create index tara_idx on dim_locatie (tara) local;
```



Index TARA_IDX dropped.

Index TARA_IDX created.

Folosire index:

EXPLAIN PLAN

```
SET STATEMENT_ID = 's1_index_local' FOR
select * from dim_locatie where tara = 'Romania';
```

SELECT plan_table_output

FROM

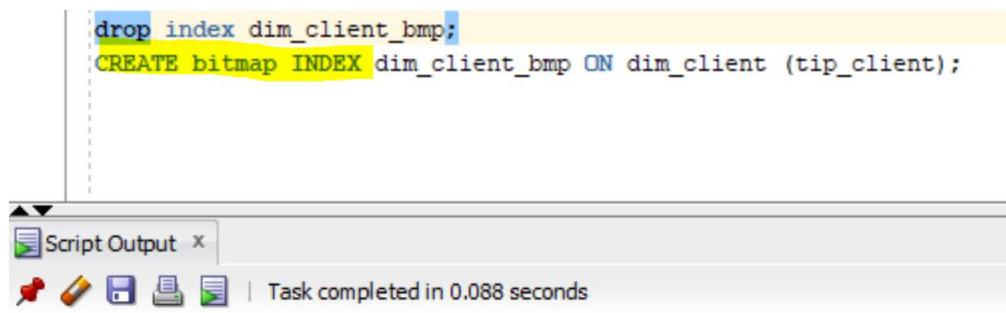
```
table(dbms_xplan.display('plan_table', 's1_index_local','serial'));
```

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:

```
drop index dim_client_bmp;
CREATE bitmap INDEX dim_client_bmp ON dim_client (tip_client);
```



Index DIM_CLIENT_BMP dropped.

INDEX DIM_CLIENT_BMP created.

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 = 'sl_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| |
|---|---|---|---|---|---|---|---|---|
| 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 :

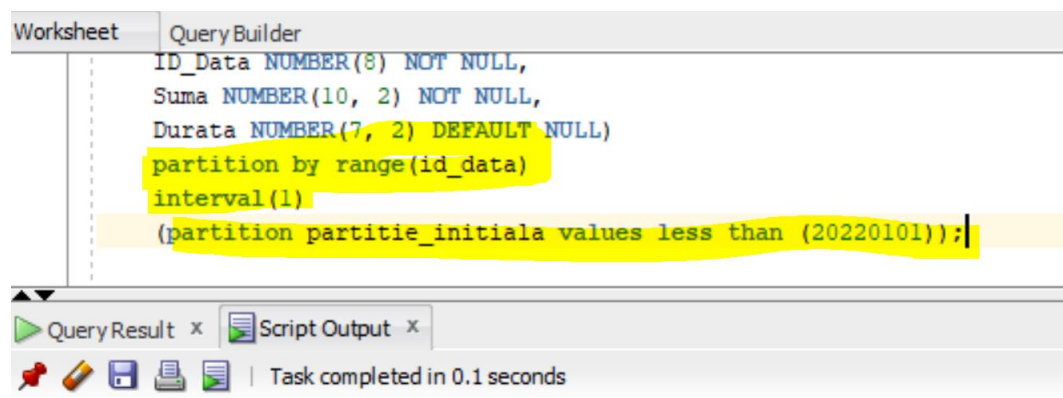


Table FACT_TRANZACTIII dropped.

Table FACT_TRANZACTIII created.

Partitionarea te tip interval este o extensie a partitionarii de tip range. Partitiile noi sunt create autmat 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:

```
select * from user_tab_partitions;
```

	TABLE_NAME	COMPOSITE	PARTITION_NAME	SUBPARTITION_COUNT	HIGH_VALUE	HIGH_VALUE_LENGTH	PARTITION_POSITION	TABLESPACE_NAME	PCT_FREE
1	FACT_TRANZACTIII	NO	PARTITIE_INITIALA		0 20220101	8	1	USERS	10
2	FACT_TRANZACTIII	NO	SYS_P541		0 20221232	8	2	USERS	10
3	FACT_TRANZACTIII	NO	SYS_P546		0 20230124	8	3	USERS	10
4	FACT_TRANZACTIII	NO	SYS_P545		0 20230126	8	4	USERS	10
5	FACT_TRANZACTIII	NO	SYS_P544		0 20230129	8	5	USERS	10
6	FACT_TRANZACTIII	NO	SYS_P542		0 20230130	8	6	USERS	10
7	FACT_TRANZACTIII	NO	SYS_P543		0 20230131	8	7	USERS	10

```
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 :

```
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)  
 );
```

Script Output x Query Result x Explain Plan x Explain Plan 1 x Query Res

Task completed in 0.12 seconds

Table DIM_LOCATIE created.

Verificarea crearii partitiilor prin consultarea dictionarului datelor.

```
select * from user_tab_partitions;
```

TABLE_NAME	COMPOSITE	PARTITION_NAME	SUBPARTITION_COUNT	HIGH_VALUE	HIGH_VALUE_LENGTH	PARTITION_POSITION	TABLESPACE_NAME	PCT_FREE	PCT_USED	INI_TRANS
5 DIM_LOCATIE	NO	TM		0 'Timisoara'	11	1	USERS	10	(null)	1
6 DIM_LOCATIE	NO	B		0 'Bucuresti'	11	2	USERS	10	(null)	1
7 DIM_LOCATIE	NO	CT		0 'Constanta'	11	3	USERS	10	(null)	1
8 DIM_LOCATIE	NO	BV		0 'Brasov'	8	4	USERS	10	(null)	1
9 DIM_LOCATIE	NO	NEDEFINIT		0 default	7	5	USERS	10	(null)	1

```
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
```

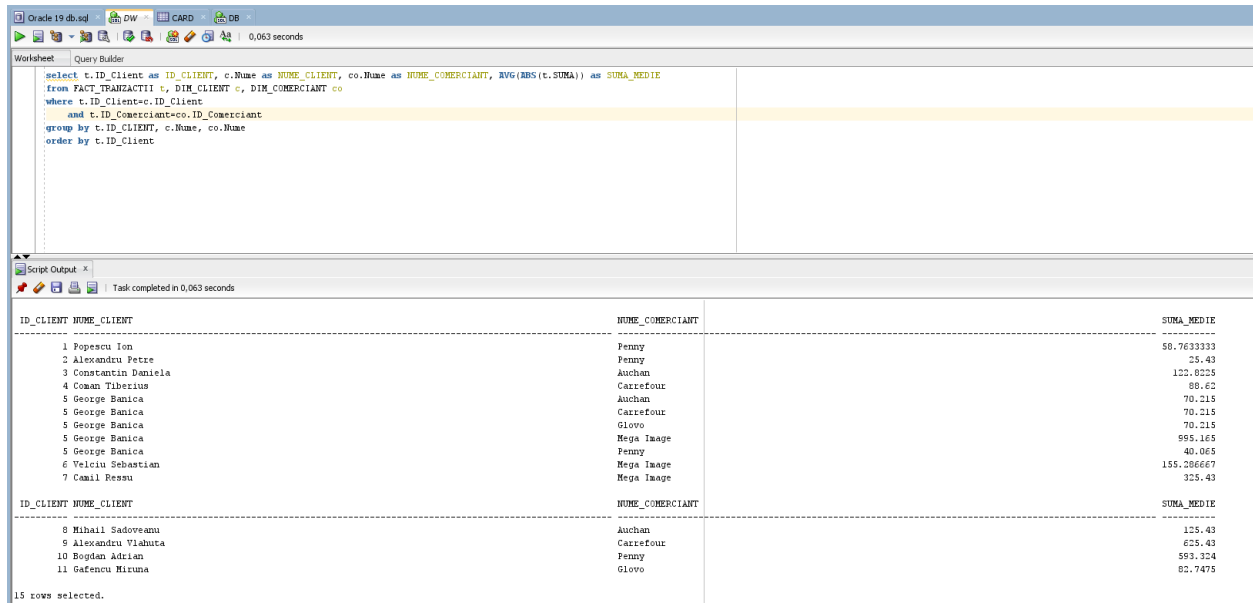
```
from FACT_TRANZACTII t, DIM_CLIENT c, DIM_COMERCIANT co
```

```
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
```



The screenshot shows the Oracle SQL Developer interface. The top pane displays the following SQL query:

```
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
```

The bottom pane shows the query results in a table with 15 rows. The columns are ID_CLIENT, NUME_CLIENT, NUME_COMERCIANT, and SUMA_MEDIE.

ID_CLIENT	NUME_CLIENT	NUME_COMERCIANT	SUMA_MEDIE
1	Popescu Ion	Penny	50.7633333
2	Alexandru Petre	Penny	25.43
3	Constantin Daniela	Auchan	122.8225
4	Coan Tiberius	Carrefour	66.62
5	George Banica	Auchan	70.215
5	George Banica	Carrefour	70.215
5	George Banica	Glovo	70.215
5	George Banica	Mega Image	955.165
5	George Banica	Penny	40.065
6	Velciu Sebastian	Mega Image	155.286667
7	Camil Resu	Mega Image	325.43
8	Mihail Sadoreanu	Auchan	125.43
9	Alexandru Vlahuta	Carrefour	625.43
10	Bogdan Adrian	Penny	593.324
11	Gafencu Miruna	Glovo	82.7475

15 rows selected.

--Crearea vizualizarii materializate

Script Output x

Task completed in 0,562 seconds

Materialized view VM_CLIENTI_COMERCIANTI_SUMA_MEDIE_TRANZACTIE created.

[illegible]

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;
```

The screenshot shows the Oracle SQL Developer interface. The top pane displays the table statistics for 'VM_CLIENTI_COMERCIANTI_SUMA_MEDIE_TRANZACTIE'. The bottom pane shows the 'Column Statistics' for the same table.

Name	Value
1 ROWS_ROW	15
2 BLOCKS	4
3 AVG_ROW_LEN	33
4 SAMPLE_SIZE	15
5 LAST_ANALYZED	31-01-2023
6 LAST_ANALYZED_SINCE	31-01-2023

OWNER	TABLE_NAME	COLUMN_NAME	NUM_DISTINCT	LOW_VALUE	HIGH_VALUE	DENSITY	NUM_NULLS	NUM_BUCKETS	LAST_ANALYZED	SAMPLE_SIZE	GLOBAL_STATS	USER_STATS	NOTES	AVG_COL_LEN	HISTOGRAM	SCOPE
DW	VM_CLIENTI_COMERCIANTI_SUMA_MEDIE_TRANZACTIE	ID_CLIENT	11	C10C	C10C	0.09090909090909	0	0	31-01-2023	15	YES	NO	STATS_ON_LOAD	15	NONE	SHAPED
DW	VM_CLIENTI_COMERCIANTI_SUMA_MEDIE_TRANZACTIE	NUME_CLIENT	11	4166457061064727520506374728	56563638732036362173746916E	0.09090909090909	0	0	31-01-2023	15	YES	NO	STATS_ON_LOAD	15	NONE	SHAPED
DW	VM_CLIENTI_COMERCIANTI_SUMA_MEDIE_TRANZACTIE	NUME_COMERCIANT	5	4175636616E	506566E75	0.2	0	0	31-01-2023	15	YES	NO	STATS_ON_LOAD	0	NONE	SHAPED
DW	VM_CLIENTI_COMERCIANTI_SUMA_MEDIE_TRANZACTIE	SUMA_MEDIE	13	C11A2C	C06A60133	0.0769230769230769	0	0	31-01-2023	15	YES	NO	STATS_ON_LOAD	0	NONE	SHAPED

--Setare parametrilor 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'));
```

Oracle 19 db.sql x DW x DB x Oracle 19 db x FACT_TRANZACTII x

0,18799999 seconds

Worksheet Query Builder

```

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'));

```

Script Output x

Task completed in 0,188 seconds

Explained.

PLAN_TABLE_OUTPUT

Plan hash value: 2472764427

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		15	495	4 (25)	00:00:01
1	SORT ORDER BY		15	495	4 (25)	00:00:01
2	MAT_VIEW REWRITE ACCESS FULL	VM_CLIENTI_COMERCIANTI_SUMA_MEDIE_TRANZACTIE	15	495	3 (0)	00:00:01

9 rows selected.

--Rescriere cerere

```

select ID_CLIENT, NUME_CLIENT, NUME_COMERCIANT, SUMA_MEDIE
from VM_CLIENTI_COMERCIANTI_SUMA_MEDIE_TRANZACTIE

```

Oracle 19 db.sql x DW x DB x Oracle 19 db x VM_CLIENTI_COMERCIANTI_SUMA_MEDIE_TRANZACTIE x

0,052 seconds

Worksheet Query Builder

```

select ID_CLIENT, NUME_CLIENT, NUME_COMERCIANT, SUMA_MEDIE
from VM_CLIENTI_COMERCIANTI_SUMA_MEDIE_TRANZACTIE

```

Script Output x

Task completed in 0,052 seconds

ID_CLIENT	NUME_CLIENT	NUME_COMERCIANT	SUMA_MEDIE
1	Popescu Ion	Penny	50.7633333
2	Alexandru Petre	Penny	25.43
3	Constantin Daniela	Auchan	122.0225
4	Coman Tiberius	Carrefour	80.62
5	George Banica	Auchan	70.215
5	George Banica	Carrefour	70.215
5	George Banica	Glovo	70.215
5	George Banica	Mega Image	995.165
5	George Banica	Penny	40.065
6	Velciu Sebastian	Mega Image	155.206667
7	Camil Rensu	Mega Image	325.43
8	Mihail Sadorreanu	Auchan	125.43
9	Alexandru Vlahuta	Carrefour	625.43
10	Bogdan Adrian	Penny	553.324
11	Gafencu Miruna	Glovo	82.7475

15 rows selected.

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 debitarile).

```
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 efectuate dupa numarul de tranzactii efectuate doar cu POS din anul 2022(doar debitarile).

```
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 debitarile).

```
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 debitarile).

```
SELECT dim_comerciant.numa, 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.numa
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 debitarile).

```
SELECT dim_client.numa, 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.numa
ORDER BY suma_maxima DESC
FETCH FIRST 3 ROWS ONLY;
```