

# ***PROIECT DATA WAREHOUSE***

~ Modul implementare baza de date ~

## 1. Crearea bazei de date OLTP și a utilizatorilor

--Baza noastra de date va avea 3 tipuri de utilizatori in aplicatie

- utilizatorul manager care va introduce schema si cu toate datele din aplicatie.
- acesta va avea acces la toate datele din schema oltp prin SELECT,UPDATE,DELETE,INSERT
- Managerul are posibilitatea de a introduce rezervari,modifica utilizatori, update pe rezervari.
- Practic are acces sa faca tot ce vrea pe schema
- acesta va avea acces si in olap pentru a vizualiza rapoartele scoase dar
- si de a modifica datele din tabele dupa propriul interes.

```
show con_name;
alter session set container= orclpdb;
show con_name;
ALTER PLUGGABLE DATABASE orclpdb open;
```

```
CREATE USER dw_manager IDENTIFIED BY mng_pass;
GRANT CREATE SESSION TO dw_manager;
GRANT CREATE ANY TABLE TO dw_manager;
GRANT CREATE ANY INDEX TO dw_manager;
GRANT CREATE VIEW TO dw_manager;
GRANT CREATE TRIGGER TO dw_manager;
GRANT SELECT ANY TABLE TO dw_manager;
GRANT DELETE ANY TABLE TO dw_manager;
GRANT UPDATE ANY TABLE TO dw_manager;
GRANT ALTER ANY TABLE TO dw_manager;
GRANT UNLIMITED TABLESPACE TO dw_manager;
--Pentru a vizualiza privilegiile adaugate putem folosi aceasta cerere asupra utlizatorului creat.
SELECT *
FROM session_privs;
-- intro schema si datele furnizate pentru schema
--SCHEMA CU TABELE TREBUIE RULATE IN dw_manager
```

```
--script creare schema oltp.txt
--DATELE DIN TABELE TREBUIE INTRODUSE IN dw_manager
--script inserare date oltp.txt
```

```
-- urmatorul tip va fi de tip admin care va avea posibilitatea de UPDATE peste toate tabele din schema OLTP
-- fara a avea posibilitatea de a sterge orice tip de inregistrare deoarece si inregistrarile neconforme pot reprezenta
-- un interes pentru manager.
```

```
CREATE USER dw_admin IDENTIFIED BY admin_pass;
GRANT CREATE SESSION TO dw_admin;
```

```
GRANT SELECT ANY TABLE TO dw_admin;
GRANT DELETE ANY TABLE TO dw_admin;
GRANT UPDATE ANY TABLE TO dw_admin;
GRANT ALTER ANY TABLE TO dw_admin;
```

```
--daca dorim sa oferim doar anumite privilegii mai restrictive asupra anumitor tabele putem folosi comanda urmatoare
```

```
--GRANT UPDATE ON dw_manager.rezervare TO dw_admin;
```

```
--sau daca dorim sa nu mai folosim anumite privilegii precum cel de mai putem folosi comanda
```

```
--REVOKE DELETE ON dw_manager.rezervare FROM dw_admin;
```

```
User DW_ADMIN created.
```

```
--pentru a accesa un tabel trebuie sa folosim dw_manager.numa_tabel deoarece altfel nu merge
```

```
Grant succeeded.
```

```
-- iar ultimul tip de utilizator este cel de utilizator care are
```

```
-- posibilitatea sa vizualizeze hotelurile si sa introduca date in rezervari.
```

```
Grant succeeded.
```

```
-- acesta nu avea acces la baza de date.
```

```
Grant succeeded.
```

```
--ca si SYS putem rula urmatoarea cerere pentru a vizualiza care sunt care sunt privilegiile oferite
```

```
SELECT substr(grantee,1,20) grantee, owner,substr(table_name,1,15) table_name, grantor, privilege
```

```
FROM DBA_TAB_PRIVS
```

```
WHERE grantee like 'DW_%';
```

```
Grant succeeded.
```

```
Grant succeeded.
```

```
--cu aceasta comanda putem vedea doar privilegiile mai restrictive.
```

```
>>Query Run In:Query Result 3
```

```

SET FEEDBACK 1
SET NUMWIDTH 10
SET LINESIZE 80
SET TRIMSPOOL ON
SET TAB OFF
SET PAGESIZE 100
SET ECHO OFF
REM *****
REM Create the UTILIZATOR table to hold users information for application
Prompt ***** Creating UTILIZATOR table ....

```

```

CREATE TABLE utilizator
( id_utilizator NUMBER GENERATED ALWAYS as IDENTITY(START WITH 1 INCREMENT BY 1),
  nume_utilizator VARCHAR(30) CONSTRAINT nume_utilizator_nn NOT NULL,
  hash_parola VARCHAR(25) CONSTRAINT hash_parola_utilizator_nn NOT NULL,
  nume_complet VARCHAR(30) CONSTRAINT nume_complet_utilizator_nn NOT NULL,
  telefon VARCHAR(15) CONSTRAINT telefon_utilizator_nn NOT NULL,
  email VARCHAR(50) CONSTRAINT email_utilizator_nn NOT NULL,
  data_nasterii DATE CONSTRAINT data_nasterii_utilizator_nn NOT NULL,
  gen VARCHAR(20) DEFAULT NULL,
  stare_civila VARCHAR(20) DEFAULT NULL);

```

```

CREATE UNIQUE INDEX id_utilizator_index
ON utilizator (id_utilizator);

```

```

ALTER TABLE utilizator
ADD ( CONSTRAINT id_nume_utilizator_pk PRIMARY KEY (id_utilizator)) ;

```

```

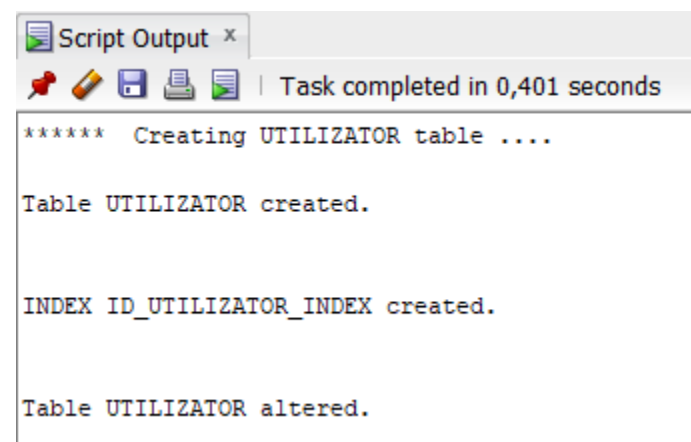
CREATE UNIQUE INDEX id_utilizator_index
ON utilizator (id_utilizator);

```

```

ALTER TABLE utilizator
ADD ( CONSTRAINT id_nume_utilizator_pk PRIMARY KEY (id_utilizator)) ;

```



```
REM *****
REM Create the REZERVARE table to hold information for reservation of users
```

Prompt \*\*\*\*\* Creating REZERVARE table ....

```
CREATE TABLE rezervare
( id_rezervare NUMBER GENERATED ALWAYS as IDENTITY(START WITH 1 INCREMENT BY 1),
  id_client NUMBER CONSTRAINT id_client_utilizator_nn NOT NULL,
  data_inceput DATE CONSTRAINT data_inceput_rezervare_nn NOT NULL,
  data_sfarsit DATE CONSTRAINT data_sfarsit_rezervare_nn NOT NULL,
  data_efectuarii DATE);
```

```
CREATE UNIQUE INDEX id_rezervare_index
ON rezervare (id_rezervare);
```

```
ALTER TABLE rezervare
ADD ( CONSTRAINT id_rezervare_pk PRIMARY KEY (id_rezervare)) ;
```

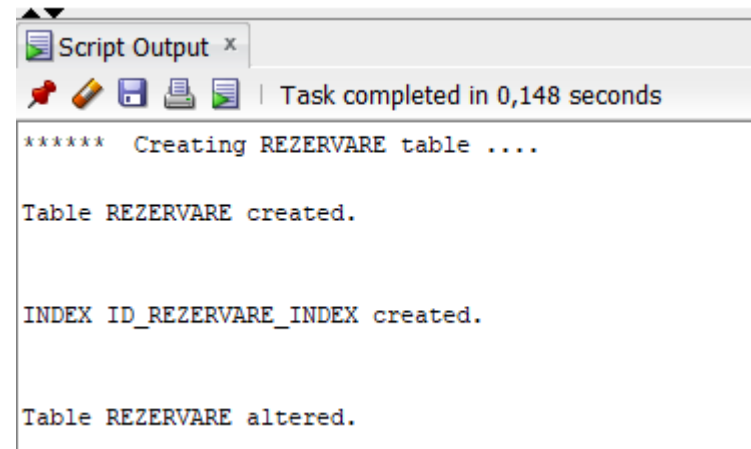
```
REM *****
REM Create the REZERVARE CAMERA table to hold information about rooms
of users
```

Prompt \*\*\*\*\* Creating REZERVARE CAMERA table ....

```
CREATE TABLE rezervare_camera
( id_rezervare NUMBER CONSTRAINT id_rezervare_atribuie_nn NOT
NULL,
  id_camera NUMBER CONSTRAINT id_camera_atribuie_nn NOT NULL);
```

```
CREATE UNIQUE INDEX id_rezervare_camera_index
ON rezervare_camera (id_rezervare,id_camera);
```

```
ALTER TABLE rezervare_camera
ADD ( CONSTRAINT id_rezervare_camera__pk PRIMARY KEY (id_rezervare,id_camera));
```



Script Output x

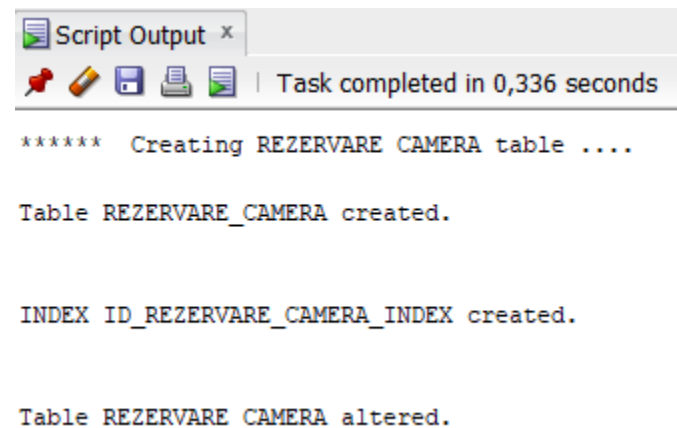
Task completed in 0,148 seconds

```
***** Creating REZERVARE table ....

Table REZERVARE created.

INDEX ID_REZERVARE_INDEX created.

Table REZERVARE altered.
```



Script Output x

Task completed in 0,336 seconds

```
***** Creating REZERVARE CAMERA table ....

Table REZERVARE_CAMERA created.

INDEX ID_REZERVARE_CAMERA_INDEX created.

Table REZERVARE_CAMERA altered.
```

REM \*\*\*\*\*  
REM Create the CAMERA table to hold informations about rooms

Prompt \*\*\*\*\* Creating CAMERA table ....

```
CREATE TABLE camera
( id_camera NUMBER GENERATED ALWAYS as IDENTITY(START WITH 1 INCREMENT BY 1),
  id_hotel NUMBER CONSTRAINT id_hotel_camera_nn NOT NULL,
  nr_camera NUMBER,
  nr_etaj NUMBER,
  nr_paturi_duble NUMBER CONSTRAINT nr_paturi_duble_camera_nn NOT NULL,
  nr_paturi_simple NUMBER CONSTRAINT nr_paturi_simple_camera_nn NOT NULL,
  are_terasa NUMBER(1) CONSTRAINT are_terasa_camera_nn NOT NULL,
  are_televizor NUMBER(1) CONSTRAINT are_televizor_camera_nn NOT NULL,
  pret_per_noapte NUMBER CONSTRAINT pret_per_noapte_camera_nn NOT NULL);
```

```
CREATE UNIQUE INDEX id_camera_index
ON camera (id_camera);
```

```
ALTER TABLE camera
ADD ( CONSTRAINT id_camera_camera_pk PRIMARY KEY (id_camera)) ;
```

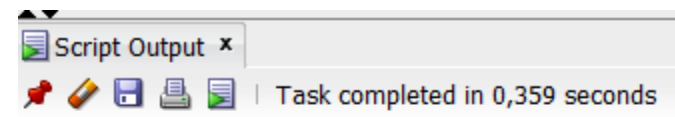
REM \*\*\*\*\*  
REM Create the HOTEL table to hold information of hotels

Prompt \*\*\*\*\* Creating HOTEL table ....

```
CREATE TABLE hotel
( id_hotel NUMBER GENERATED ALWAYS as IDENTITY(START WITH 1
INCREMENT BY 1),
  nume VARCHAR2(50) CONSTRAINT nume_hotel_hotel_nn NOT NULL,
  nr_stele NUMBER CONSTRAINT nr_stele_nn NOT NULL,
  id_zona NUMBER CONSTRAINT id_zona_hotel_nn NOT NULL,
  are_mic_dejun_inclus NUMBER(1) CONSTRAINT are_mic_dejun_inclus_hotel_nn
NOT NULL);
```

```
CREATE UNIQUE INDEX id_hotel_index
ON hotel (id_hotel);
```

```
ALTER TABLE hotel
ADD ( CONSTRAINT id_hotel_hotel_pk PRIMARY KEY (id_hotel)) ;
```

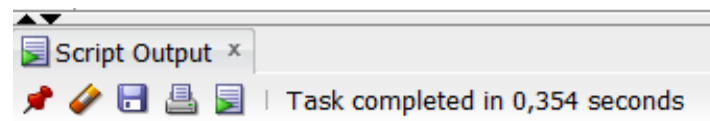


\*\*\*\*\* Creating CAMERA table ....

Table CAMERA created.

INDEX ID\_CAMERA\_INDEX created.

Table CAMERA altered.



\*\*\*\*\* Creating HOTEL table ....

Table HOTEL created.

INDEX ID\_HOTEL\_INDEX created.

Table HOTEL altered.

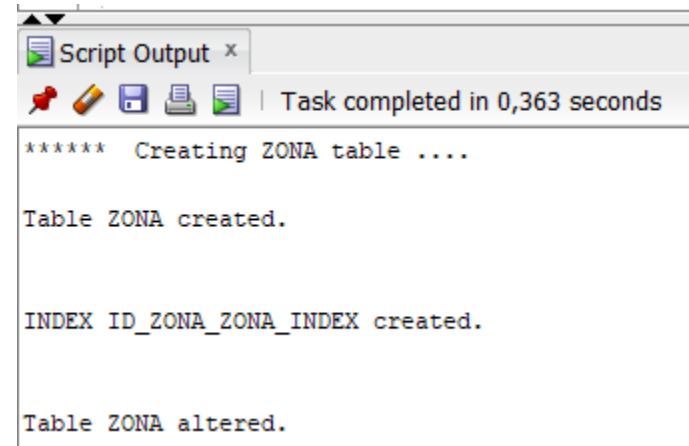
REM \*\*\*\*\*  
REM Create the ZONA table to hold information the zones where the hotels was build

Prompt \*\*\*\*\* Creating ZONA table ....

```
CREATE TABLE zona
( id_zona NUMBER GENERATED ALWAYS as IDENTITY(START WITH 1 INCREMENT BY 1),
  regiune VARCHAR2(50),
  judet VARCHAR2(50) CONSTRAINT judet_zona_nn NOT NULL,
  localitate VARCHAR2(50) CONSTRAINT localitate_zona_nn NOT NULL,
  pozitie VARCHAR2(50) CONSTRAINT pozitie_zona_nn NOT NULL);
```

```
CREATE UNIQUE INDEX id_zona_zona_index
ON zona (id_zona);
```

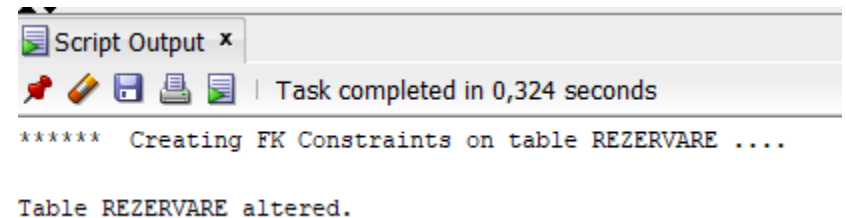
```
ALTER TABLE zona
ADD ( CONSTRAINT id_zona_zona_pk PRIMARY KEY (id_zona)) ;
```



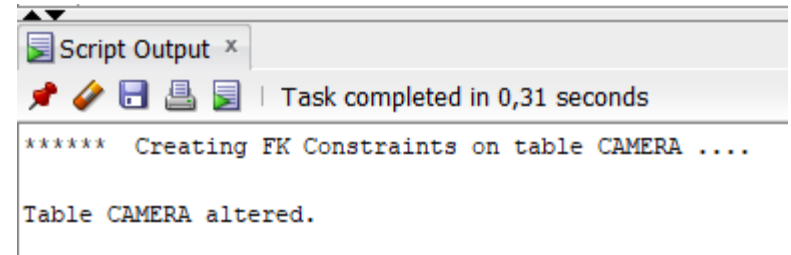
REM \*\*\*\*\*Introducerea de FK tabelului\*\*\*\*\* REZERVARE\*\*\*\*\*

Prompt \*\*\*\*\* Creating FK Constraints on table REZERVARE ....

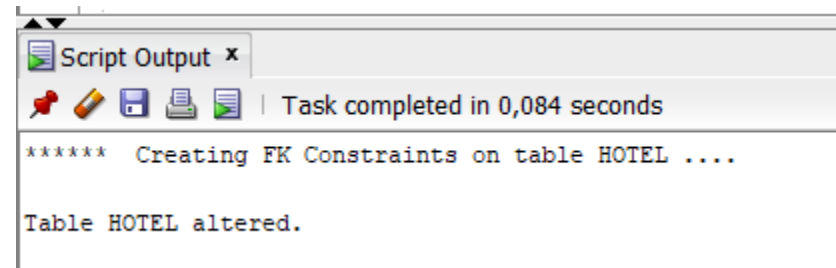
```
ALTER TABLE rezervare
add constraint fk_id_client_id_utilizator FOREIGN KEY(id_client) REFERENCES utilizator(id_utilizator);
```



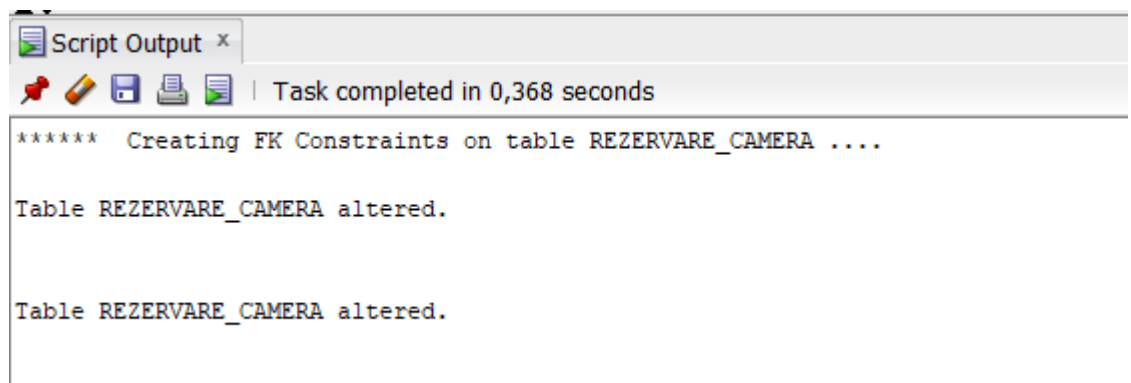
```
REM *****Introducerea de FK tabelului***** CAMERA*****
Prompt ***** Creating FK Constraints on table CAMERA ....
ALTER TABLE camera
add constraint fk_camera_hotel FOREIGN KEY(id_hotel) REFERENCES
hotel(id_hotel);
```



```
REM *****Introducerea de FK tabelului***** HOTEL*****
Prompt ***** Creating FK Constraints on table HOTEL ....
ALTER TABLE hotel
add constraint fk_hotel_zona FOREIGN KEY(id_zona) REFERENCES
zona(id_zona);
```



```
REM *****Introducerea de FK tabelului***** REZERVARE_CAMERA*****
Prompt ***** Creating FK Constraints on table REZERVARE_CAMERA ....
ALTER TABLE rezervare_camera
ADD CONSTRAINT fk_rezervare_camera_rezervare FOREIGN KEY(id_rezervare) REFERENCES rezervare(id_rezervare);
ALTER TABLE rezervare_camera
ADD CONSTRAINT fk_rezervare_camera_camera FOREIGN KEY (id_camera) REFERENCES camera(id_camera);
```



## 2. (0,25p) Generarea datelor și inserarea acestora în tabele.

---Inserarea datelor in tabelul ZONA---

```
INSERT INTO ZONA (JUDET, LOCALITATE, POZITIE) VALUES ('Mures', 'Targu Mures', 'centrala'); COMMIT;
INSERT INTO ZONA (JUDET, LOCALITATE, POZITIE) VALUES ('Brasov', 'Bran', 'centrala'); COMMIT;
INSERT INTO ZONA (JUDET, LOCALITATE, POZITIE) VALUES ('Brasov', 'Bran', 'periferica'); COMMIT;
INSERT INTO ZONA (JUDET, LOCALITATE, POZITIE) VALUES ('Bucuresti', 'Bucuresti', 'centrala'); COMMIT;
INSERT INTO ZONA (JUDET, LOCALITATE, POZITIE) VALUES ('Prahova', 'Sinaia', 'centrala'); COMMIT;
```

The screenshot displays a database query tool interface. At the top, there are two tabs: 'Worksheet' and 'Query Builder'. The 'Worksheet' tab is active, showing a SQL script. The script starts with a comment '---Inserarea datelor in tabelul ZONA---' followed by five INSERT statements for the ZONA table. The statements insert data for Mures, Brasov, and Bucharest. Below the script, there is a 'Script Output' section. It shows the execution progress with icons for running, saving, and refreshing. A status bar indicates 'Task completed in 0,381 seconds'. The output area shows four lines, each stating '1 row inserted.', corresponding to the first four INSERT statements in the script.

```
---Inserarea datelor in tabelul ZONA---
INSERT INTO ZONA (JUDET, LOCALITATE, POZITIE) VALUES ('Mures', 'Targu Mures', 'centrala');
INSERT INTO ZONA (JUDET, LOCALITATE, POZITIE) VALUES ('Brasov', 'Bran ', 'centrala');
INSERT INTO ZONA (JUDET, LOCALITATE, POZITIE) VALUES ('Brasov', 'Bran ', 'periferica');
INSERT INTO ZONA (JUDET, LOCALITATE, POZITIE) VALUES ('Bucharest', 'Bucharest', 'centrala');
INSERT INTO ZONA (JUDET, LOCALITATE, POZITIE) VALUES ('Prahova', 'Sinaia ', 'centrala');
INSERT INTO ZONA (JUDET, LOCALITATE, POZITIE) VALUES ('Bucharest', 'Bucharest ', 'periferica');
INSERT INTO ZONA (JUDET, LOCALITATE, POZITIE) VALUES ('Sibiu ', 'Sibiu ', 'centrala');
INSERT INTO ZONA (JUDET, LOCALITATE, POZITIE) VALUES ('Iasi', 'Iasi ', 'centrala');
```

Script Output x

Task completed in 0,381 seconds

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

---Inserarea datelor in tabelul HOTEL(dependent de ZONA)---

```
INSERT INTO HOTEL (NUME, NR_STELE, ID_ZONA, ARE_MIC_DEJUN_INCLUS) VALUES ('Hotel Privo', 4, 1, 0); COMMIT;
INSERT INTO HOTEL (NUME, NR_STELE, ID_ZONA, ARE_MIC_DEJUN_INCLUS) VALUES ('Conacul Bratescu', 4, 2, 1);
COMMIT;
INSERT INTO HOTEL (NUME, NR_STELE, ID_ZONA, ARE_MIC_DEJUN_INCLUS) VALUES ('Transylvanian Inn', 3, 3, 1);
COMMIT;
INSERT INTO HOTEL (NUME, NR_STELE, ID_ZONA, ARE_MIC_DEJUN_INCLUS) VALUES ('The Mansion Boutique Hotel', 4, 4,
1); COMMIT;
INSERT INTO HOTEL (NUME, NR_STELE, ID_ZONA, ARE_MIC_DEJUN_INCLUS) VALUES ('Ioana Hotel', 5, 5, 1); COMMIT;
```



Worksheet Query Builder

---Inserarea datelor in tabelul HOTEL(dependent de ZONA)---

```
INSERT INTO HOTEL (NUME, NR_STELE, ID_ZONA, ARE_MIC_DEJUN_INCLUS) VALUES ('Hotel Privo',4,1,0);
INSERT INTO HOTEL (NUME, NR_STELE, ID_ZONA, ARE_MIC_DEJUN_INCLUS) VALUES ('Conacul Bratescu', 4, 2, 1);
INSERT INTO HOTEL (NUME, NR_STELE, ID_ZONA, ARE_MIC_DEJUN_INCLUS) VALUES ('Transylvanian Inn', 3, 3, 1);
INSERT INTO HOTEL (NUME, NR_STELE, ID_ZONA, ARE_MIC_DEJUN_INCLUS) VALUES ('The Mansion Boutique Hotel', 4, 4, 1);
INSERT INTO HOTEL (NUME, NR_STELE, ID_ZONA, ARE_MIC_DEJUN_INCLUS) VALUES ('Ioana Hotel', 5, 5, 1);
INSERT INTO HOTEL (NUME, NR_STELE, ID_ZONA, ARE_MIC_DEJUN_INCLUS) VALUES ('Epoque Hotel Relais Chateaux', 5, 6, 0);
INSERT INTO HOTEL (NUME, NR_STELE, ID_ZONA, ARE_MIC_DEJUN_INCLUS) VALUES ('Rosen Villa Sibiu', 1, 7, 1);
INSERT INTO HOTEL (NUME, NR_STELE, ID_ZONA, ARE_MIC_DEJUN_INCLUS) VALUES ('Hotel International Iasi', 4, 8, 1);
```

Script Output x

Task completed in 0,83 seconds

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

---Inserarea datelor in tabelul CAMERA(dependent de HOTEL)---

```
INSERT INTO CAMERA (ID_HOTEL, NR_CAMERA, NR_ETAJ, NR_PATURI_DUBLE, NR_PATURI_SIMPLE, ARE_TERASA,
ARE_TELEVIZOR, PRET_PER_NOAPTE)
VALUES (76, 312, 3, 0, 3, 1, 0, 450); COMMIT;
```

```
INSERT INTO CAMERA (ID_HOTEL, NR_CAMERA, NR_ETAJ, NR_PATURI_DUBLE, NR_PATURI_SIMPLE, ARE_TERASA,
ARE_TELEVIZOR, PRET_PER_NOAPTE)
VALUES (78, 107, 1, 0, 1, 1, 0, 100); COMMIT;
```

```
INSERT INTO CAMERA (ID_HOTEL, NR_CAMERA, NR_ETAJ, NR_PATURI_DUBLE, NR_PATURI_SIMPLE, ARE_TERASA,
ARE_TELEVIZOR, PRET_PER_NOAPTE)
VALUES (44, 405, 4, 0, 2, 1, 0, 180); COMMIT;
```

Worksheet

Query Builder

---Inserarea datelor in tabelul CAMERA(dependent de HOTEL)---

INSERT INTO CAMERA (ID\_HOTEL, NR\_CAMERA, NR\_ETAJ, NR\_PATURI\_DUBLE, NR\_PATURI\_SIMPLE, ARE\_TERASA, ARE\_TELEVIZOR, PRET\_PER\_NOAPTE)  
VALUES (76, 312, 3, 0, 3, 1, 0, 450);

INSERT INTO CAMERA (ID\_HOTEL, NR\_CAMERA, NR\_ETAJ, NR\_PATURI\_DUBLE, NR\_PATURI\_SIMPLE, ARE\_TERASA, ARE\_TELEVIZOR, PRET\_PER\_NOAPTE)  
VALUES (78, 107, 1, 0, 1, 1, 0, 100);

INSERT INTO CAMERA (ID\_HOTEL, NR\_CAMERA, NR\_ETAJ, NR\_PATURI\_DUBLE, NR\_PATURI\_SIMPLE, ARE\_TERASA, ARE\_TELEVIZOR, PRET\_PER\_NOAPTE)  
VALUES (44, 405, 4, 0, 2, 1, 0, 180);

Script Output x

Task completed in 1,639 seconds

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

-----

---Inserarea datelor in tabelul UTILIZATOR---

```
INSERT INTO UTILIZATOR (NUME_UTILIZATOR, HASH_PAROLA, NUME_COMPLET, TELEFON, EMAIL, DATA_NASTERII, GEN, STARE_CIVILA)
VALUES ('bernard_noble', 'kr102h8', 'Bernard Noble', '+40 710 024 027', 'kstoltenberg@yahoo.com', to_date('13-Oct-1976', 'DD-MON-RR'), 'masculin', 'necasatorit '); COMMIT;
```

```
INSERT INTO UTILIZATOR (NUME_UTILIZATOR, HASH_PAROLA, NUME_COMPLET, TELEFON, EMAIL, DATA_NASTERII, GEN, STARE_CIVILA)
VALUES ('isis_saunders', 'PA3pnYV', 'Isis Saunders', '+40 713 721 929', 'orn.skye@huels.com', to_date('6-Apr-2001', 'DD-MON-RR'), 'feminin', 'casatorita'); COMMIT;
```

```
INSERT INTO UTILIZATOR (NUME_UTILIZATOR, HASH_PAROLA, NUME_COMPLET, TELEFON, EMAIL, DATA_NASTERII, GEN, STARE_CIVILA)
VALUES ('micaela_gillespie', 'G7fSXy8', 'Micaela Gillespie', '+40 713 037 240', 'stuart22@yahoo.com', to_date('19-Apr-1966', 'DD-MON-RR'), 'feminin', 'necasatorita'); COMMIT;
```

```
INSERT INTO UTILIZATOR (NUME_UTILIZATOR, HASH_PAROLA, NUME_COMPLET, TELEFON, EMAIL, DATA_NASTERII, GEN, STARE_CIVILA)
VALUES ('gilbert_mccarty', 'ek1Rwcq', 'Gilbert Mccarty', '+40 711 666 147', 'gskiles@altenwerth.com', to_date('31-Jul-1969', 'DD-MON-RR'), 'masculin', 'necasatorit '); COMMIT;
```

Worksheet Query Builder

```
---Inserarea datelor in tabelul UTILIZATOR---
```

```
INSERT INTO UTILIZATOR (NUME_UTILIZATOR, HASH_PAROLA, NUME_COMPLET, TELEFON, EMAIL, DATA_NASTERII, GEN, STARE_CIVILA)
VALUES ('bernard_noble', 'kr1O2h8', 'Bernard Noble', '+40 710 024 027', 'kstoltenberg@yahoo.com', to_date('13-Oct-1976', 'DD-MON-RR'), 'male', 'necasatorit ');

INSERT INTO UTILIZATOR (NUME_UTILIZATOR, HASH_PAROLA, NUME_COMPLET, TELEFON, EMAIL, DATA_NASTERII, GEN, STARE_CIVILA)
VALUES ('isis_saunders', 'PA3pnYV', 'Isis Saunders', '+40 713 721 929', 'orn.skye@huelis.com', to_date('6-Apr-2001', 'DD-MON-RR'), 'female', 'casatorita');

INSERT INTO UTILIZATOR (NUME_UTILIZATOR, HASH_PAROLA, NUME_COMPLET, TELEFON, EMAIL, DATA_NASTERII, GEN, STARE_CIVILA)
```

Script Output x

Task completed in 0,768 seconds

```
1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.
```

---Inserarea datelor in tabelul REZERVARE(dependent de UTILIZATOR)---

```
INSERT INTO REZERVARE (ID_CLIENT, DATA_INCEPUT, DATA_SFARSIT)
VALUES (89, to_date('2-Jul-2023', 'DD-MON-RR'), to_date('9-Jul-2023', 'DD-MON-RR')); COMMIT;
```

```
INSERT INTO REZERVARE (ID_CLIENT, DATA_INCEPUT, DATA_SFARSIT)
VALUES (49, to_date('4-Feb-2025', 'DD-MON-RR'), to_date('13-Feb-2025', 'DD-MON-RR')); COMMIT;
```

```
INSERT INTO REZERVARE (ID_CLIENT, DATA_INCEPUT, DATA_SFARSIT)
VALUES (4, to_date('24-Apr-2023', 'DD-MON-RR'), to_date('27-Apr-2023', 'DD-MON-RR')); COMMIT;
```

Worksheet

Query Builder

---

Inserarea datelor in tabelul REZERVARE(dependent de UTILIZATOR)---

INSERT INTO REZERVARE (ID\_CLIENT, DATA\_INCEPUT, DATA\_SFARSIT)

VALUES (89, to\_date('2-Jul-2023', 'DD-MON-RR'), to\_date('9-Jul-2023', 'DD-MON-RR'));

INSERT INTO REZERVARE (ID\_CLIENT, DATA\_INCEPUT, DATA\_SFARSIT)

VALUES (49, to\_date('4-Feb-2025', 'DD-MON-RR'), to\_date('13-Feb-2025', 'DD-MON-RR'));

INSERT INTO REZERVARE (ID\_CLIENT, DATA\_INCEPUT, DATA\_SFARSIT)

VALUES (4, to\_date('24-Apr-2023', 'DD-MON-RR'), to\_date('27-Apr-2023', 'DD-MON-RR'));

Script Output x

Task completed in 0,735 seconds

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

-----  
---Inserarea datelor in tabelul REZERVARE\_CAMERA---

```
INSERT INTO REZERVARE_CAMERA (ID_REZERVARE, ID_CAMERA)
VALUES (95, 167);
```

```
INSERT INTO REZERVARE_CAMERA (ID_REZERVARE, ID_CAMERA)
VALUES (93, 54);
```

```
INSERT INTO REZERVARE_CAMERA (ID_REZERVARE, ID_CAMERA)
VALUES (47, 140);
```

```
INSERT INTO REZERVARE_CAMERA (ID_REZERVARE, ID_CAMERA)
VALUES (93, 71);
```

```
INSERT INTO REZERVARE_CAMERA (ID_REZERVARE, ID_CAMERA)
VALUES (10, 154);
```

```
INSERT INTO REZERVARE_CAMERA (ID_REZERVARE, ID_CAMERA)
VALUES (4, 120);
```

Worksheet Query Builder

```
-----  
--Inserarea datelor in tabelul REZERVARE_CAMERA--  
  
INSERT INTO REZERVARE_CAMERA (ID_REZERVARE, ID_CAMERA)  
VALUES (55, 167);  
  
INSERT INTO REZERVARE_CAMERA (ID_REZERVARE, ID_CAMERA)  
VALUES (53, 54);  
  
INSERT INTO REZERVARE_CAMERA (ID_REZERVARE, ID_CAMERA)  
VALUES (47, 140);  
  
INSERT INTO REZERVARE_CAMERA (ID_REZERVARE, ID_CAMERA)  
VALUES (52, 71);
```

Script Output x

Task completed in 0,986 seconds

```
1 row inserted.  
  
1 row inserted.  
1 row inserted.  
1 row inserted.  
1 row inserted.
```

```
--GENEREAZA O DATA ALEATOARE A EFECTUARII PENTRU FIECARE REZERVARE  
UPDATE rezervare SET data_efectuarii=TO_DATE(TRUNC(DBMS_RANDOM.VALUE(TO_CHAR(DATE'2022-01-01','J'),TO_CHAR(DATE'2022-12-31','J'))),'J'); COMMIT;
```

Worksheet Query Builder

```
--GENEREAZA O DATA ALEATOARE A EFECTUARII PENTRU FIECARE REZERVARE  
UPDATE rezervare SET data_efectuarii=TO_DATE(TRUNC(DBMS_RANDOM.VALUE(TO_CHAR(DATE'2022-01-01','J'),TO_CHAR(DATE'2022-12-31','J'))),'J'); COMMIT;
```

Script Output x

Task completed in 0,272 seconds

```
99 rows updated.  
  
Commit complete.
```

--SETEAZA REGIUNILE ASOCIATE JUDETELOR

UPDATE zona

SET regiune='Transilvania'

WHERE judet='Cluj' OR judet='Brasov' OR judet='Sibiu' OR judet='Mures'; COMMIT;

UPDATE zona

SET regiune='Maramures'

WHERE judet='Maramures' OR judet='Satu Mare'; COMMIT;

UPDATE zona

SET regiune='Muntenia'

WHERE judet='Prahova' OR judet='Ilfov' OR judet='Arges' OR judet='Buzau' OR judet='Bucuresti'; COMMIT;

UPDATE zona

SET regiune='Oltenia'

WHERE judet='Valcea'; COMMIT;

UPDATE zona

SET regiune='Dobrogea'

WHERE judet='Constanta'; COMMIT;

Worksheet

Query Builder

```
--SETEAZA REGIUNILE ASOCIATE JUDETELOR
```

```
UPDATE zona
```

```
SET regiune='Transilvania'
```

```
WHERE judet='Cluj' OR judet='Brasov' OR judet='Sibiu' OR judet='Mures'; COMMIT;
```

```
UPDATE zona
```

```
SET regiune='Maramures'
```

```
WHERE judet='Maramures' OR judet='Satu Mare'; COMMIT;
```

```
UPDATE zona
```

```
SET regiune='Muntenia'
```

```
WHERE judet='Prahova' OR judet='Ilfov' OR judet='Arges' OR judet='Buzau' OR judet='Bucuresti'; COMMIT;
```

```
UPDATE zona
```

```
SET regiune='Oltenia'
```

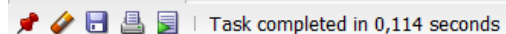
```
WHERE judet='Valcea'; COMMIT;
```

```
UPDATE zona
```

```
SET regiune='Dobrogea'
```

```
WHERE judet='Constanta'; COMMIT;
```

Script Output x

Task completed in 0,114 seconds

14 rows updated.

Commit complete.

5 rows updated.

Commit complete.

6 rows updated.

Commit complete.

### 3. (0,5p) Crearea bazei de date depozit și a utilizatorilor

Schema bazei de date depozit a fost introdusă în conexiunea utilizatorului dw\_manager. Acest utilizator a fost creat asemănător cu utilizatorul pentru baza de date OLTP.

```
CREATE TABLE perioada_rezervare_OLAP(  
id_perioada NUMBER(8,0) GENERATED ALWAYS as IDENTITY(START WITH 1 INCREMENT BY 1),  
zi_din_luna_inceput NUMBER(2,0) CONSTRAINT zi_din_luna_inceput_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
luna_inceput CHAR(3) CONSTRAINT luna_inceput_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
an_inceput NUMBER(4,0) CONSTRAINT an_inceput_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
zi_din_saptamana_inceput CHAR(3) CONSTRAINT zi_din_saptamana_inceput_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
zi_din_an_inceput NUMBER(3,0) CONSTRAINT zi_din_an_inceput_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
zi_din_luna_sfarsit NUMBER(2,0) CONSTRAINT zi_din_luna_sfarsit_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
luna_sfarsit CHAR(3) CONSTRAINT luna_sfarsit_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
an_sfarsit NUMBER(4,0) CONSTRAINT an_sfarsit_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
zi_din_saptamana_sfarsit CHAR(3) CONSTRAINT zi_din_saptamana_sfarsit_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
zi_din_an_sfarsit NUMBER(3,0) CONSTRAINT zi_din_an_sfarsit_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
durata_in_zile NUMBER(2,0) CONSTRAINT durata_in_zile_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
PRIMARY KEY(id_perioada)  
);
```

The screenshot displays a database management interface with two main panes. The top pane, titled 'Worksheet' and 'Query Builder', contains the SQL script for creating the 'perioada\_rezervare\_OLAP' table. The script defines various numeric and character fields with constraints and a primary key. The bottom pane, titled 'Script Output', shows the execution results, indicating that the task was completed successfully in 0.064 seconds and that the table 'PERIODADA\_REZERVARE\_OLAP' was created.

```
Worksheet Query Builder
```

```
CREATE TABLE perioada_rezervare_OLAP(  
id_perioada NUMBER(8,0) GENERATED ALWAYS as IDENTITY(START WITH 1 INCREMENT BY 1),  
zi_din_luna_inceput NUMBER(2,0) CONSTRAINT zi_din_luna_inceput_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
luna_inceput CHAR(3) CONSTRAINT luna_inceput_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
an_inceput NUMBER(4,0) CONSTRAINT an_inceput_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
zi_din_saptamana_inceput CHAR(3) CONSTRAINT zi_din_saptamana_inceput_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
zi_din_an_inceput NUMBER(3,0) CONSTRAINT zi_din_an_inceput_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
zi_din_luna_sfarsit NUMBER(2,0) CONSTRAINT zi_din_luna_sfarsit_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
luna_sfarsit CHAR(3) CONSTRAINT luna_sfarsit_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
an_sfarsit NUMBER(4,0) CONSTRAINT an_sfarsit_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
zi_din_saptamana_sfarsit CHAR(3) CONSTRAINT zi_din_saptamana_sfarsit_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
zi_din_an_sfarsit NUMBER(3,0) CONSTRAINT zi_din_an_sfarsit_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
durata_in_zile NUMBER(2,0) CONSTRAINT durata_in_zile_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
PRIMARY KEY(id_perioada)  
);
```

Script Output x

Task completed in 0,064 seconds

Table PERIODADA\_REZERVARE\_OLAP created.



```

CREATE TABLE tip_camera_OLAP(
    id_tip_camera NUMBER(8) GENERATED ALWAYS as IDENTITY(START WITH 1 INCREMENT BY 1),
    nr_paturi_duble NUMBER(1) CONSTRAINT nr_paturi_duble_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,
    nr_paturi_simple NUMBER(1) CONSTRAINT nr_paturi_simple_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,
    are_terasa NUMBER(1) CONSTRAINT are_terasa_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,
    are_televizor NUMBER(1) CONSTRAINT are_televizor_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,
    PRIMARY KEY (id_tip_camera)
);

```







Worksheet	Query Builder
<pre> CREATE TABLE tip_camera_OLAP(     id_tip_camera NUMBER(8) GENERATED ALWAYS as IDENTITY(START WITH 1 INCREMENT BY 1),     nr_paturi_duble NUMBER(1) CONSTRAINT nr_paturi_duble_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,     nr_paturi_simple NUMBER(1) CONSTRAINT nr_paturi_simple_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,     are_terasa NUMBER(1) CONSTRAINT are_terasa_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,     are_televizor NUMBER(1) CONSTRAINT are_televizor_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,     PRIMARY KEY (id_tip_camera) ); </pre>	
<div>  Script Output x         </div> <div>      Task completed in 0,047 seconds </div>	

Table TIP\_CAMERA\_OLAP created.

```






CREATE TABLE hotel_OLAP
(
    id_hotel NUMBER(8),
    nume VARCHAR2(50) CONSTRAINT nume_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,
    nr_stele NUMBER(1) CONSTRAINT nr_stele_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,
    regiune VARCHAR2(50),
    judet VARCHAR2(20) CONSTRAINT judet_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,
    localitate VARCHAR(20) CONSTRAINT localitate_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,
    pozitie VARCHAR(20) CONSTRAINT pozitie_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,
    nr_camere NUMBER(3),
    are_mic_dejun_inclus NUMBER(1) CONSTRAINT are_mic_dejun_inclus_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,
    PRIMARY KEY (id_hotel)
)

```

```

PARTITION BY LIST (nr_stele)
(PARTITION o_stea VALUES (1),
PARTITION doua_stele VALUES (2),
PARTITION trei_stele VALUES (3),
PARTITION patru_stele VALUES (4),
PARTITION cinci_stele VALUES (5));

```

Worksheet	Query Builder
<pre> CREATE TABLE hotel_OLAP (     id_hotel NUMBER(8),     nume VARCHAR2(50) CONSTRAINT nume_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,     nr_stele NUMBER(1) CONSTRAINT nr_stele_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,     regiune VARCHAR2(50),     judet VARCHAR2(20) CONSTRAINT judet_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,     localitate VARCHAR(20) CONSTRAINT localitate_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,     pozitie VARCHAR(20) CONSTRAINT pozitie_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,     nr_camere NUMBER(3),     are_mic_dejun_inclus NUMBER(1) CONSTRAINT are_mic_dejun_inclus_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,     PRIMARY KEY (id_hotel) ) PARTITION BY LIST (nr_stele) (PARTITION o_stea VALUES (1), PARTITION doua_stele VALUES (2), PARTITION trei_stele VALUES (3), PARTITION patru_stele VALUES (4), PARTITION cinci_stele VALUES (5)); </pre>	
<div> <div>Script Output x</div> <div>      </div> </div> <div>Task completed in 0,081 seconds</div>	
<div>Table HOTEL_OLAP created.</div>	

```

CREATE TABLE tip_client_OLAP(
    id_tip_client NUMBER(8) GENERATED ALWAYS as IDENTITY(START WITH 1 INCREMENT BY 1),
    varsta NUMBER(3) CONSTRAINT varsta_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,
    gen VARCHAR(20),
    stare_civila VARCHAR(20),
    PRIMARY KEY (id_tip_client)
);

```

Worksheet Query Builder

```
CREATE TABLE tip_client_OLAP(  
  id_tip_client NUMBER(8) GENERATED ALWAYS as IDENTITY(START WITH 1 INCREMENT BY 1),  
  varsta NUMBER(3) CONSTRAINT varsta_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  gen VARCHAR(20),  
  stare_civila VARCHAR(20),  
  PRIMARY KEY (id_tip_client)  
);
```

Script Output x

Task completed in 0,07 seconds

Table TIP\_CLIENT\_OLAP created.

```
CREATE TABLE moment_efectuare_rezervare_OLAP(  
  id_moment_efectuare NUMBER(8,0) GENERATED ALWAYS as IDENTITY(START WITH 1 INCREMENT BY 1),  
  zi_din_luna NUMBER(2,0) CONSTRAINT zi_din_luna_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  luna CHAR(3) CONSTRAINT luna_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  an NUMBER(4,0) CONSTRAINT an_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  zi_din_saptamana CHAR(3) CONSTRAINT zi_din_saptamana_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  zi_din_an NUMBER(3,0) CONSTRAINT zi_din_an_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  --ora_aprox NUMBER(2,0),  
  PRIMARY KEY (id_moment_efectuare)  
);
```

Worksheet Query Builder

```
CREATE TABLE moment_efectuare_rezervare_OLAP(  
  id_moment_efectuare NUMBER(8,0) GENERATED ALWAYS as IDENTITY(START WITH 1 INCREMENT BY 1),  
  zi_din_luna NUMBER(2,0) CONSTRAINT zi_din_luna_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  luna CHAR(3) CONSTRAINT luna_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  an NUMBER(4,0) CONSTRAINT an_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  zi_din_saptamana CHAR(3) CONSTRAINT zi_din_saptamana_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  zi_din_an NUMBER(3,0) CONSTRAINT zi_din_an_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  --ora_aprox NUMBER(2,0),  
  PRIMARY KEY (id_moment_efectuare)  
);
```

Script Output x

Task completed in 0,06 seconds

Table MOMENT\_EFECTUARE\_REZERVARE\_OLAP created.

```
DROP TABLE rezervare_camera_OLAP;  
CREATE TABLE rezervare_camera_OLAP(  
  id_rezervare NUMBER(8,0) CONSTRAINT id_rezervare_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  id_hotel NUMBER(8,0) CONSTRAINT id_hotel_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  id_perioada NUMBER(8,0) CONSTRAINT id_perioada_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  id_moment_efectuare NUMBER(8,0) CONSTRAINT id_moment_efectuare_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  id_tip_client NUMBER(8,0) CONSTRAINT id_tip_client_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  id_tip_camera NUMBER(8,0) CONSTRAINT id_tip_camera_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  pret NUMBER CONSTRAINT pret_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE  
);
```

Worksheet Query Builder

```
CREATE TABLE rezervare_camera_OLAP(  
  id_rezervare NUMBER(8,0) CONSTRAINT id_rezervare_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  id_hotel NUMBER(8,0) CONSTRAINT id_hotel_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  id_perioada NUMBER(8,0) CONSTRAINT id_perioada_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  id_moment_efectuare NUMBER(8,0) CONSTRAINT id_moment_efectuare_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  id_tip_client NUMBER(8,0) CONSTRAINT id_tip_client_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  id_tip_camera NUMBER(8,0) CONSTRAINT id_tip_camera_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,  
  pret NUMBER CONSTRAINT pret_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE  
);
```

Script Output x

Task completed in 0,062 seconds

Table REZERVARE\_CAMERA\_OLAP created.

```
ALTER TABLE rezervare_camera_OLAP  
ADD FOREIGN KEY(id_hotel) REFERENCES hotel_OLAP(id_hotel);  
  
ALTER TABLE rezervare_camera_OLAP  
ADD FOREIGN KEY(id_perioada) REFERENCES perioada_rezervare_OLAP(id_perioada);  
  
ALTER TABLE rezervare_camera_OLAP  
ADD FOREIGN KEY(id_moment_efectuare) REFERENCES moment_efectuare_rezervare_OLAP(id_moment_efectuare);  
  
ALTER TABLE rezervare_camera_OLAP  
ADD FOREIGN KEY(id_tip_camera) REFERENCES tip_camera_OLAP(id_tip_camera);  
  
ALTER TABLE rezervare_camera_OLAP  
ADD FOREIGN KEY(id_tip_client) REFERENCES tip_client_OLAP(id_tip_client);
```

```
ALTER TABLE rezervare_camera_OLAP
ADD FOREIGN KEY(id_hotel) REFERENCES hotel_OLAP(id_hotel);

ALTER TABLE rezervare_camera_OLAP
ADD FOREIGN KEY(id_perioada) REFERENCES perioada_rezervare_OLAP(id_perioada);

ALTER TABLE rezervare_camera_OLAP
ADD FOREIGN KEY(id_moment_efectuare) REFERENCES moment_efectuare_rezervare_OLAP(id_moment_efectuare);

ALTER TABLE rezervare_camera_OLAP
ADD FOREIGN KEY(id_tip_camera) REFERENCES tip_camera_OLAP(id_tip_camera);

ALTER TABLE rezervare_camera_OLAP
ADD FOREIGN KEY(id_tip_client) REFERENCES tip_client_OLAP(id_tip_client);
```

## Script Output x


 Task completed in 0,165 seconds

Table REZERVARE\_CAMERA\_OLAP altered.

Table REZERVARE\_CAMERA\_OLAP altered.

Table REZERVARE\_CAMERA\_OLAP altered.

Table REZERVARE\_CAMERA\_OLAP altered.

Table REZERVARE\_CAMERA\_OLAP altered.

#### 4. (0,5p) Popularea cu informații a bazei de date depozit folosind ca sursă datele din baza de date OLTP

```
INSERT INTO perioada_rezervare_OLAP(zi_din_luna_inceput, luna_inceput, an_inceput, zi_din_saptamana_inceput,
zi_din_an_inceput, zi_din_luna_sfarsit, luna_sfarsit, an_sfarsit, zi_din_saptamana_sfarsit, zi_din_an_sfarsit,
durata_in_zile)
SELECT DISTINCT TO_NUMBER(TO_CHAR(data_inceput,'DD')), TO_CHAR(data_inceput,'MON'),
TO_NUMBER(TO_CHAR(data_inceput,'YYYY')),
TO_CHAR(data_inceput,'DY'),TO_NUMBER(TO_CHAR(data_inceput,'DDD')),
TO_NUMBER(TO_CHAR(data_sfarsit,'DD')), TO_CHAR(data_sfarsit,'MON'), TO_NUMBER(TO_CHAR(data_sfarsit,'YYYY')),
TO_CHAR(data_sfarsit,'DY'),TO_NUMBER(TO_CHAR(data_sfarsit,'DDD')),
data_sfarsit - data_inceput
FROM rezervare; COMMIT;
```

The screenshot shows a database query tool interface. At the top, there are tabs for 'Worksheet' and 'Query Builder'. Below them, a text area contains the SQL script. The script is as follows:

```
INSERT INTO perioada_rezervare_OLAP(zi_din_luna_inceput, luna_inceput, an_inceput, zi_din_saptamana_inceput, zi_din_an_inceput, zi_din_luna_sfarsit, luna_sfarsit, an_sfarsit, zi_din_saptamana_sfarsit, zi_din_an_sfarsit,
durata_in_zile)
SELECT DISTINCT TO_NUMBER(TO_CHAR(data_inceput,'DD')), TO_CHAR(data_inceput,'MON'), TO_NUMBER(TO_CHAR(data_inceput,'YYYY')),
TO_CHAR(data_inceput,'DY'),TO_NUMBER(TO_CHAR(data_inceput,'DDD')),
TO_NUMBER(TO_CHAR(data_sfarsit,'DD')), TO_CHAR(data_sfarsit,'MON'), TO_NUMBER(TO_CHAR(data_sfarsit,'YYYY')),
TO_CHAR(data_sfarsit,'DY'),TO_NUMBER(TO_CHAR(data_sfarsit,'DDD')),
data_sfarsit - data_inceput
FROM rezervare; COMMIT;
```

Below the script area, there is a 'Script Output' window. It shows the following messages:

```
97 rows inserted.

Commit complete.
```

```
INSERT INTO moment_efectuare_rezervare_OLAP(zi_din_luna, luna, an, zi_din_saptamana, zi_din_an)
SELECT DISTINCT TO_NUMBER(TO_CHAR(data_efectuarii,'DD')), TO_CHAR(data_efectuarii,'MON'),
TO_NUMBER(TO_CHAR(data_efectuarii,'YYYY')),TO_CHAR(data_efectuarii,'DY'),TO_NUMBER(TO_CHAR(data_efectuarii,'DD
D'))--,
--TO_NUMBER(TO_CHAR(data_efectuarii,'HH')) + CASE WHEN TO_NUMBER(TO_CHAR(data_efectuarii,'MI')) > 29 THEN 1
ELSE 0 END AS ora_aprox
FROM rezervare; COMMIT;
```

Worksheet Query Builder

```
INSERT INTO moment_efectuare_rezervare_OLAP(zi_din_luna, luna, an, zi_din_saptamana, zi_din_an)
SELECT DISTINCT TO_NUMBER(TO_CHAR(data_efectuarii,'DD')), TO_CHAR(data_efectuarii,'MON'),
TO_NUMBER(TO_CHAR(data_efectuarii,'YYYY')),TO_CHAR(data_efectuarii,'DY'),TO_NUMBER(TO_CHAR(data_efectuarii,'DDD'))--,
--TO_NUMBER(TO_CHAR(data_efectuarii,'HH')) + CASE WHEN TO_NUMBER(TO_CHAR(data_efectuarii,'MI')) > 29 THEN 1 ELSE 0 END AS ora_aprox
FROM rezervare; COMMIT;
```

Script Output x

Task completed in 0,095 seconds

84 rows inserted.

Commit complete.

```
INSERT INTO hotel_OLAP(id_hotel, nume, regiune, judet, localitate, pozitie, nr_stele, are_mic_dejun_inclus)
SELECT id_hotel, nume, regiune, judet, localitate, pozitie, nr_stele, are_mic_dejun_inclus
FROM hotel JOIN zona
USING(id_zona); COMMIT;
```

Worksheet Query Builder

```
INSERT INTO hotel_OLAP(id_hotel, nume, regiune, judet, localitate, pozitie, nr_stele, are_mic_dejun_inclus)
SELECT id_hotel, nume, regiune, judet, localitate, pozitie, nr_stele, are_mic_dejun_inclus
FROM hotel JOIN zona
USING(id_zona); COMMIT;
```

Script Output x

Task completed in 0,691 seconds

100 rows inserted.

Commit complete.

```
INSERT INTO tip_client_OLAP(varsta,gen,stare_civila)
SELECT DISTINCT FLOOR(MONTHS_BETWEEN(SYSDATE,data_nasterii)/12) AS varsta, gen, stare_civila
FROM utilizator; COMMIT;
```








Worksheet

Query Builder

```
INSERT INTO tip_client_OLAP(varsta,gen,stare_civila)
SELECT DISTINCT FLOOR(MONTHS_BETWEEN(SYSDATE,data_nasterii)/12) AS varsta, gen, stare_civila
FROM utilizator; COMMIT;
```

Script Output x

     | Task completed in 0,089 seconds

84 rows inserted.

Commit complete.






```
INSERT INTO tip_camera_OLAP(nr_paturi_duble, nr_paturi_simple,are_terasa,are_televizor)
SELECT DISTINCT nr_paturi_duble, nr_paturi_simple,are_terasa,are_televizor
FROM camera; COMMIT;
```

Worksheet

Query Builder

```
INSERT INTO tip_camera_OLAP(nr_paturi_duble, nr_paturi_simple,are_terasa,are_televizor)
SELECT DISTINCT nr_paturi_duble, nr_paturi_simple,are_terasa,are_televizor
FROM camera; COMMIT;
```

Script Output x

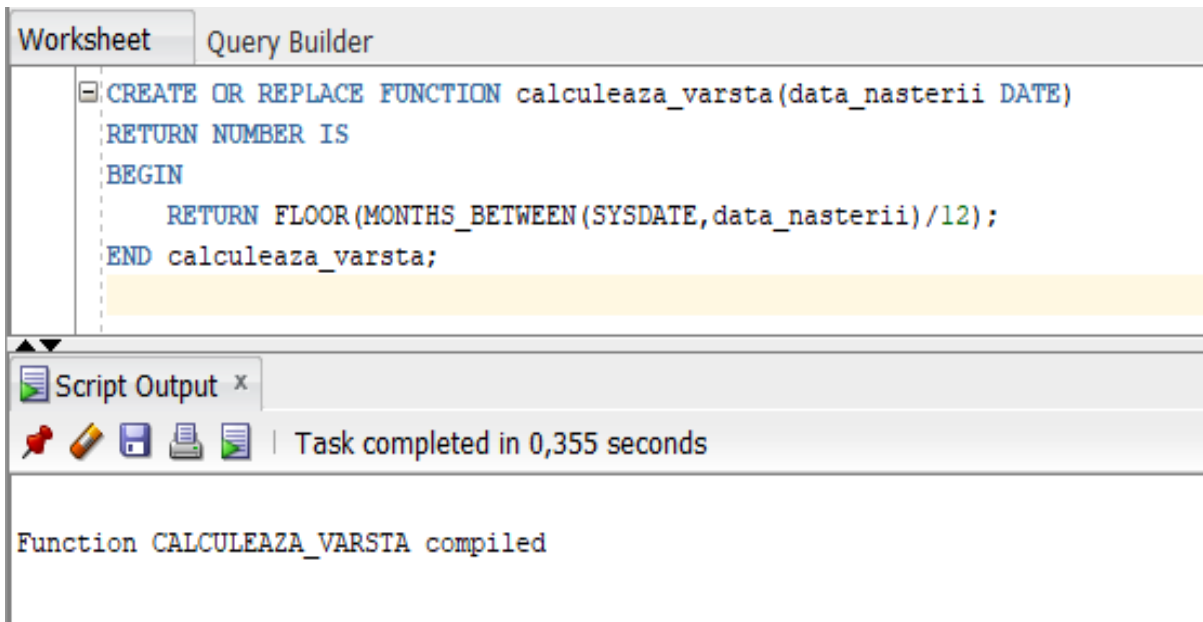
     | Task completed in 0,07 seconds

36 rows inserted.

Commit complete.

---functie ce calculeaza varsta clientului-

```
CREATE OR REPLACE FUNCTION calculeaza_varsta(data_nasterii DATE)
RETURN NUMBER IS
BEGIN
    RETURN FLOOR(MONTHS_BETWEEN(SYSDATE,data_nasterii)/12);
END calculeaza_varsta;
```



The screenshot displays a software interface with two main sections. The top section, titled "Worksheet" and "Query Builder", contains a SQL script in a text editor. The script is for creating or replacing a function named "calculeaza\_varsta" that takes a date "data\_nasterii" as input and returns a number. The script uses the "FLOOR" and "MONTHS\_BETWEEN" functions to calculate the age in years. The bottom section, titled "Script Output", shows the result of executing the script: "Function CALCULEAZA\_VARSTA compiled". Above the output text, there is a status bar indicating "Task completed in 0,355 seconds" and several icons (a red pushpin, a yellow pencil, a blue floppy disk, a printer, and a green document).

Worksheet Query Builder

```
CREATE OR REPLACE FUNCTION calculeaza_varsta(data_nasterii DATE)
RETURN NUMBER IS
BEGIN
    RETURN FLOOR(MONTHS_BETWEEN(SYSDATE,data_nasterii)/12);
END calculeaza_varsta;
```

Script Output x

Task completed in 0,355 seconds

Function CALCULEAZA\_VARSTA compiled

---functie ce cauta id-ul perioadei de rezervare---

```
CREATE OR REPLACE FUNCTION gaseste_id_perioada_OLAP(p_zi_din_an_inceput  
perioada_rezervare_OLAP.zi_din_an_inceput%TYPE, p_an_inceput perioada_rezervare_OLAP.an_inceput%TYPE,  
p_zi_din_an_sfarsit  
perioada_rezervare_OLAP.zi_din_an_sfarsit%TYPE, p_an_sfarsit perioada_rezervare_OLAP.an_sfarsit%TYPE)  
RETURN perioada_rezervare_OLAP.id_perioada%TYPE IS  
v_id_gasit perioada_rezervare_OLAP.id_perioada%TYPE;  
BEGIN  
    SELECT id_perioada  
    INTO v_id_gasit  
    FROM perioada_rezervare_OLAP  
    WHERE zi_din_an_inceput=p_zi_din_an_inceput  
    AND an_inceput=p_an_inceput  
    AND zi_din_an_sfarsit=p_zi_din_an_sfarsit  
    AND an_sfarsit= p_an_sfarsit;  
    RETURN v_id_gasit;  
END gaseste_id_perioada_OLAP;
```

The screenshot displays a database IDE interface. At the top, there are two tabs: 'Worksheet' and 'Query Builder'. The 'Worksheet' tab is active, showing the PL/SQL code for the function 'gaseste\_id\_perioada\_OLAP'. The code is color-coded, with keywords in blue and identifiers in black. Below the code editor, there is a 'Script Output' window. It contains a message: 'Task completed in 0,096 seconds'. At the bottom of the IDE, a status bar indicates 'Function GASESTE\_ID\_PERIOADA\_OLAP compiled'.

```
CREATE OR REPLACE FUNCTION gaseste_id_perioada_OLAP(p_zi_din_an_inceput perioada_rezervare_OLAP.zi_din_an_inceput%TYPE, p_an_inceput perioada_rezervare_OLAP.an_inceput%TYPE,  
p_zi_din_an_sfarsit perioada_rezervare_OLAP.zi_din_an_sfarsit%TYPE, p_an_sfarsit perioada_rezervare_OLAP.an_sfarsit%TYPE)  
RETURN perioada_rezervare_OLAP.id_perioada%TYPE IS  
v_id_gasit perioada_rezervare_OLAP.id_perioada%TYPE;  
BEGIN  
    SELECT id_perioada  
    INTO v_id_gasit  
    FROM perioada_rezervare_OLAP  
    WHERE zi_din_an_inceput=p_zi_din_an_inceput  
    AND an_inceput=p_an_inceput  
    AND zi_din_an_sfarsit=p_zi_din_an_sfarsit  
    AND an_sfarsit= p_an_sfarsit;  
    RETURN v_id_gasit;  
END gaseste_id_perioada_OLAP;
```

Script Output x

Task completed in 0,096 seconds

Function GASESTE\_ID\_PERIOADA\_OLAP compiled

---functie ce cauta id-ul momentului rezervarii---

```
CREATE OR REPLACE FUNCTION gaseste_id_moment_efectuare_OLAP(p_zi_din_an
moment_efectuare_rezervare_OLAP.zi_din_an%TYPE, p_an moment_efectuare_rezervare_OLAP.an%TYPE)
RETURN moment_efectuare_rezervare_OLAP.id_moment_efectuare%TYPE IS
v_id_gasit moment_efectuare_rezervare_OLAP.id_moment_efectuare%TYPE;
BEGIN
SELECT id_moment_efectuare
INTO v_id_gasit
FROM moment_efectuare_rezervare_OLAP
WHERE an=p_an
AND zi_din_an=p_zi_din_an;
RETURN v_id_gasit;
END gaseste_id_moment_efectuare_OLAP;
```

The screenshot shows a database IDE interface. At the top, there is a 'Worksheet' tab and a 'Query Builder' tab. The 'Worksheet' tab is active, displaying a SQL script. The script is a PL/SQL function named 'gaseste\_id\_moment\_efectuare\_OLAP'. The script is as follows:

```
CREATE OR REPLACE FUNCTION gaseste_id_moment_efectuare_OLAP(p_zi_din_an moment_efectuare_rezervare_OLAP.zi_din_an%TYPE, p_an moment_efectuare_rezervare_OLAP.an%TYPE)
RETURN moment_efectuare_rezervare_OLAP.id_moment_efectuare%TYPE IS
v_id_gasit moment_efectuare_rezervare_OLAP.id_moment_efectuare%TYPE;
BEGIN
SELECT id_moment_efectuare
INTO v_id_gasit
FROM moment_efectuare_rezervare_OLAP
WHERE an=p_an
AND zi_din_an=p_zi_din_an;
RETURN v_id_gasit;
END gaseste_id_moment_efectuare_OLAP;
```

Below the script, there is a 'Script Output' tab. The output shows the result of the script execution:

```
Function GASESTE_ID_MOMENT_EFECTUARE_OLAP compiled
```

The output also includes a status bar at the bottom indicating 'Task completed in 0,078 seconds'.

---functie ce cauta id-ul camerei---

```
CREATE OR REPLACE FUNCTION gaseste_id_tip_camera_OLAP(p_nr_paturi_duble tip_camera_OLAP.nr_paturi_duble%TYPE,  
p_nr_paturi_simple tip_camera_OLAP.nr_paturi_duble%TYPE,  
p_are_terasa tip_camera_OLAP.are_terasa%TYPE,  
p_are_televizor tip_camera_OLAP.are_televizor%TYPE)  
RETURN tip_camera_OLAP.id_tip_camera%TYPE IS  
v_id_gasit tip_camera_OLAP.id_tip_camera%TYPE;  
BEGIN  
    SELECT id_tip_camera  
    INTO v_id_gasit  
    FROM tip_camera_OLAP  
    WHERE nr_paturi_duble=p_nr_paturi_duble  
    AND nr_paturi_simple=p_nr_paturi_simple  
    AND are_terasa=p_are_terasa  
    AND are_televizor=p_are_televizor;  
    RETURN v_id_gasit;  
END gaseste_id_tip_camera_OLAP;
```

The screenshot shows a database query builder interface with a 'Worksheet' tab and a 'Query Builder' tab. The 'Query Builder' tab is active, displaying the SQL code for creating a function. The code is as follows:

```
CREATE OR REPLACE FUNCTION gaseste_id_tip_camera_OLAP(p_nr_paturi_duble tip_camera_OLAP.nr_paturi_duble%TYPE, p_nr_paturi_simple tip_camera_OLAP.nr_paturi_duble%TYPE,  
p_are_terasa tip_camera_OLAP.are_terasa%TYPE, p_are_televizor tip_camera_OLAP.are_televizor%TYPE)  
RETURN tip_camera_OLAP.id_tip_camera%TYPE IS  
v_id_gasit tip_camera_OLAP.id_tip_camera%TYPE;  
BEGIN  
    SELECT id_tip_camera  
    INTO v_id_gasit  
    FROM tip_camera_OLAP  
    WHERE nr_paturi_duble=p_nr_paturi_duble  
    AND nr_paturi_simple=p_nr_paturi_simple  
    AND are_terasa=p_are_terasa  
    AND are_televizor=p_are_televizor;  
    RETURN v_id_gasit;  
END gaseste_id_tip_camera_OLAP;
```

Below the code editor, there is a 'Script Output' tab showing the result of the execution: 'Task completed in 0,114 seconds' and 'Function GASESTE\_ID\_TIP\_CAMERA\_OLAP compiled'.

---functie ce cauta id-ul clientului---

```
CREATE OR REPLACE FUNCTION gaseste_id_tip_client_OLAP (p_varsta tip_client_OLAP.varsta%TYPE, p_gen
tip_client_OLAP.gen%TYPE,p_stare_civila tip_client_OLAP.stare_civila%TYPE)
    RETURN tip_client_OLAP.id_tip_client%TYPE IS
    v_id_gasit tip_client_OLAP.id_tip_client%TYPE;
BEGIN
    SELECT id_tip_client
    INTO v_id_gasit
    FROM tip_client_OLAP
    WHERE varsta=p_varsta
    AND gen=p_gen
    AND stare_civila=p_stare_civila;
    RETURN v_id_gasit;
END gaseste_id_tip_client_OLAP;
```

The screenshot displays a database IDE interface. At the top, there are two tabs: 'Worksheet' and 'Query Builder'. The 'Worksheet' tab is active, showing a SQL script. The script is as follows:

```
CREATE OR REPLACE FUNCTION gaseste_id_tip_client_OLAP (p_varsta tip_client_OLAP.varsta%TYPE, p_gen tip_client_OLAP.gen%TYPE, p_stare_civila tip_client_OLAP.stare_civila%TYPE)
    RETURN tip_client_OLAP.id_tip_client%TYPE IS
    v_id_gasit tip_client_OLAP.id_tip_client%TYPE;
BEGIN
    SELECT id_tip_client
    INTO v_id_gasit
    FROM tip_client_OLAP
    WHERE varsta=p_varsta
    AND gen=p_gen
    AND stare_civila=p_stare_civila;
    RETURN v_id_gasit;
END gaseste_id_tip_client_OLAP;
```

Below the script editor, there is a 'Script Output' tab. It shows the result of the script execution: 'Task completed in 0,063 seconds'. At the bottom of the IDE, a status bar indicates 'Function GASESTE\_ID\_TIP\_CLIENT\_OLAP compiled'.

```

INSERT INTO
rezervare_camera_OLAP(id_rezervare,id_hotel,id_perioada,id_moment_efectuare,id_tip_camera,id_tip_client,pret)
SELECT DISTINCT id_rezervare,id_hotel,

gaseste_id_perioada_OLAP(TO_NUMBER(TO_CHAR(data_inceput,'DDD')),TO_NUMBER(TO_CHAR(data_inceput,'YYYY')),

TO_NUMBER(TO_CHAR(data_sfarsit,'DDD')),TO_NUMBER(TO_CHAR(data_sfarsit,'YYYY'))) AS id_perioada,

gaseste_id_moment_efectuare_OLAP(TO_NUMBER(TO_CHAR(data_efectuarii,'DDD')),TO_NUMBER(TO_CHAR(data_efectuarii,'
YYYY'))) AS id_moment_efectuare,

gaseste_id_tip_camera_OLAP(camera.nr_paturi_duble,camera.nr_paturi_simple,camera.are_terasa,camera.are_televiz
or) AS id_tip_camera,

gaseste_id_tip_client_OLAP(calculeaza_varsta(utilizator.data_nasterii),utilizator.gen,utilizator.stare_civila)
AS id_tip_client,

camera.pret_per_noapte * (rezervare.data_sfarsit - rezervare.data_inceput) AS pret
FROM utilizator JOIN rezervare
ON utilizator.id_utilizator=rezervare.id_client
JOIN rezervare_camera
USING(id_rezervare)
JOIN camera
USING(id_camera)
JOIN hotel
USING(id_hotel); COMMIT;

```

```
INSERT INTO rezervare_camera_OLAP(id_rezervare,id_hotel,id_perioada,id_moment_efectuare,id_tip_camera,id_tip_client,pret)
SELECT DISTINCT id_rezervare,id_hotel,
    gaseste_id_perioada_OLAP(TO_NUMBER(TO_CHAR(data_inceput,'DDD')),TO_NUMBER(TO_CHAR(data_inceput,'YYYY')),
        TO_NUMBER(TO_CHAR(data_sfarsit,'DDD')),TO_NUMBER(TO_CHAR(data_sfarsit,'YYYY'))) AS id_perioada,
    gaseste_id_moment_efectuare_OLAP(TO_NUMBER(TO_CHAR(data_efectuarii,'DDD')),TO_NUMBER(TO_CHAR(data_efectuarii,'YYYY'))) AS id_moment_efectuare,
    gaseste_id_tip_camera_OLAP(camera.nr_paturi_duble,camera.nr_paturi_simple,camera.are_terasa,camera.are_televizor) AS id_tip_camera,
    gaseste_id_tip_client_OLAP(calculeaza_varsta(utilizator.data_nasterii),utilizator.gen,utilizator.stare_civila) AS id_tip_client,
    camera.pret_per_noapte * (rezervare.data_sfarsit - rezervare.data_inceput) AS pret
FROM utilizator JOIN rezervare
ON utilizator.id_utilizator=rezervare.id_client
JOIN rezervare_camera
USING(id_rezervare)
JOIN camera
USING(id_camera)
JOIN hotel
USING(id_hotel); COMMIT;
```

## Script Output x

Task completed in 0,341 seconds

149 rows inserted.

Commit complete.



## 5. (0,5p) Definirea constrângerilor

Constrângerile au fost definite încă de la momentul creării tabelelor bazei de date depozit.

Exemple:

```
luna_inceput CHAR(3) CONSTRAINT luna_inceput_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,
```

```
nr_paturi_duble NUMBER(1) CONSTRAINT nr_paturi_duble_nn_OLAP NOT NULL RELY DISABLE  
NOVALIDATE,
```

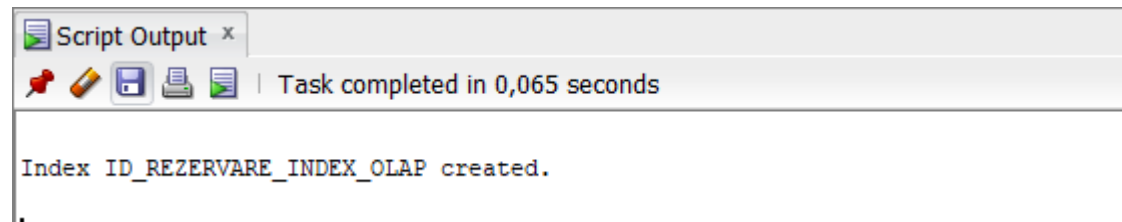
```
nr_stele NUMBER(1) CONSTRAINT nr_stele_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,
```

```
varsta NUMBER(3) CONSTRAINT varsta_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,
```

```
zi_din_an NUMBER(3,0) CONSTRAINT zi_din_an_nn_OLAP NOT NULL RELY DISABLE NOVALIDATE,
```

## 6. (1p) Definirea indecșilor și a cererilor SQL însoțite de planul de execuție al acestora (din care să reiasă ca optimizorul utilizează eficient indecșii definiți)

```
CREATE INDEX id_rezervare_index_OLAP  
ON rezervare_camera_OLAP(id_rezervare);
```



```
CREATE BITMAP INDEX luna_an_efectuare_bjindex_OLAP  
ON rezervare_camera_OLAP(moment_efectuare_OLAP.luna,moment_efectuare_OLAP.an)  
FROM moment_efectuare_OLAP,rezervare_camera_OLAP  
WHERE rezervare_camera_OLAP.id_moment_efectuare= moment_efectuare_OLAP.id_moment_efectuare;
```

```
CREATE BITMAP INDEX localitate_pozitie_bjindex_OLAP  
ON rezervare_camera_OLAP(hotel.localitate,hotel.pozitie)  
FROM hotel_OLAP, rezervare_camera_OLAP  
WHERE rezervare_camera_OLAP.id_hotel = hotel_OLAP.id_hotel;
```

```
CREATE BITMAP INDEX nr_paturi_duble_nr_paturi_simple_bjindex_OLAP  
ON rezervare_camera_OLAP(tip_camera_OLAP.nr_paturi_duble,tip_camera_OLAP.nr_paturi_simple)  
FROM tip_camera_OLAP, rezervare_camera_OLAP  
WHERE tip_camera_OLAP.id_tip_camera= rezervare_camera_OLAP.id_tip_camera;
```

WorksheetQuery Builder

ON rezervare\_camera\_OLAP(id rezervare);

CREATE BITMAP INDEX luna\_an\_efectuare\_bjindex\_OLAP

ON rezervare\_camera\_OLAP(moment\_efectuare\_rezervare\_OLAP.luna,moment\_efectuare\_rezervare\_OLAP.an)

FROM moment\_efectuare\_rezervare\_OLAP, rezervare\_camera\_OLAP

WHERE rezervare\_camera\_OLAP.id\_moment\_efectuare= moment\_efectuare\_rezervare\_OLAP.id\_moment\_efectuare;

CREATE BITMAP INDEX localitate\_pozitie\_bjindex\_OLAP

ON rezervare\_camera\_OLAP(hotel\_OLAP.localitate,hotel\_OLAP.pozitie)

FROM hotel\_OLAP, rezervare\_camera\_OLAP

WHERE rezervare\_camera\_OLAP.id\_hotel = hotel\_OLAP.id\_hotel;

CREATE BITMAP INDEX nr\_paturi\_duble\_nr\_paturi\_simple\_bjindex\_OLAP

ON rezervare\_camera\_OLAP(tip\_camera\_OLAP.nr\_paturi\_duble,tip\_camera\_OLAP.nr\_paturi\_simple)

FROM tip\_camera\_OLAP, rezervare\_camera\_OLAP

WHERE tip\_camera\_OLAP.id\_tip\_camera= rezervare\_camera\_OLAP.id\_tip\_camera;

Script Output x

Task completed in 0.096 seconds

INDEX LOCALITATE\_POZITIE\_BJINDEX\_OLAP created.

INDEX LUNA\_AN\_EFECTUARE\_BJINDEX\_OLAP created.

INDEX NR\_PATURI\_DUBLE\_NR\_PATURI\_SIMPLE\_BJINDEX\_OLAP created.

## 7. (1p) Definirea obiectelor de tip dimensiune, validarea acestora (din care să reiasă că datele respectă constrângerile impuse prin aceste tipuri de obiecte)

```
CREATE DIMENSION perioada_rezervare_dim_OLAP
LEVEL zi_din_an_inceput IS perioada_rezervare_OLAP.zi_din_an_inceput
LEVEL zi_din_an_sfarsit IS perioada_rezervare_OLAP.zi_din_an_sfarsit
ATTRIBUTE zi_din_an_inceput DETERMINES ( perioada_rezervare_OLAP.zi_din_luna_inceput,
perioada_rezervare_OLAP.luna_inceput, perioada_rezervare_OLAP.zi_din_saptamana_inceput)
ATTRIBUTE zi_din_an_sfarsit DETERMINES ( perioada_rezervare_OLAP.zi_din_luna_sfarsit,
perioada_rezervare_OLAP.luna_sfarsit, perioada_rezervare_OLAP.zi_din_saptamana_sfarsit);
```

The screenshot displays a software interface with two main sections. The top section, titled 'Query Builder', contains a script for creating a dimension named 'perioada\_rezervare\_dim\_OLAP'. The script defines two levels: 'zi\_din\_an\_inceput' and 'zi\_din\_an\_sfarsit', and two attributes that determine values based on month, year, and week. The bottom section, titled 'Script Output', shows the result of the script execution: 'Task completed in 0,202 seconds' and 'Dimension PERIOADA\_REZERVARE\_DIM\_OLAP created.'

Worksheet	Query Builder
<pre>CREATE DIMENSION perioada_rezervare_dim_OLAP LEVEL zi_din_an_inceput IS perioada_rezervare_OLAP.zi_din_an_inceput LEVEL zi_din_an_sfarsit IS perioada_rezervare_OLAP.zi_din_an_sfarsit ATTRIBUTE zi_din_an_inceput DETERMINES ( perioada_rezervare_OLAP.zi_din_luna_inceput, perioada_rezervare_OLAP.luna_inceput, perioada_rezervare_OLAP.zi_din_saptamana_inceput) ATTRIBUTE zi_din_an_sfarsit DETERMINES ( perioada_rezervare_OLAP.zi_din_luna_sfarsit, perioada_rezervare_OLAP.luna_sfarsit, perioada_rezervare_OLAP.zi_din_saptamana_sfarsit);</pre>	

Script Output x

Task completed in 0,202 seconds

Dimension PERIOADA\_REZERVARE\_DIM\_OLAP created.

```
CREATE DIMENSION moment_efectuare_rezervare_dim_OLAP
LEVEL zi_din_an IS moment_efectuare_rezervare_OLAP.zi_din_an
ATTRIBUTE zi_din_an DETERMINES ( moment_efectuare_rezervare_OLAP.zi_din_luna,
moment_efectuare_rezervare_OLAP.luna, moment_efectuare_rezervare_OLAP.zi_din_saptamana);
```

The screenshot shows a software interface with two main panes. The top pane, titled 'Worksheet' and 'Query Builder', contains a SQL script for creating a dimension. The script is as follows:

```
CREATE DIMENSION moment_efectuare_rezervare_dim_OLAP
LEVEL zi_din_an IS moment_efectuare_rezervare_OLAP.zi_din_an
ATTRIBUTE zi_din_an DETERMINES ( moment_efectuare_rezervare_OLAP.zi_din_luna, moment_efectuare_rezervare_OLAP.luna, moment_efectuare_rezervare_OLAP.zi_din_saptamana);
```

The bottom pane, titled 'Script Output', shows the result of the script execution: 'Task completed in 0,081 seconds' and 'Dimension MOMENT\_EFECTUARE\_REZERVARE\_DIM\_OLAP created.'

```
--Ierarhia de mai jos este problematica deoarece exista posibilitatea ca numele a doua localitati din judete
diferite sa coincida
CREATE DIMENSION hotel_dim_OLAP
LEVEL regiune IS hotel_OLAP.regiune
LEVEL judet IS hotel_OLAP.judet
LEVEL localitate IS hotel_OLAP.localitate
HIERARCHY judet_localitate (
    localitate CHILD OF
    judet CHILD OF
    regiune
);
```

Worksheet Query Builder

```
CREATE DIMENSION hotel_dim_OLAP
LEVEL regiune IS hotel_OLAP.regiune
LEVEL judet IS hotel_OLAP.judet
LEVEL localitate IS hotel_OLAP.localitate
HIERARCHY judet_localitate (
    localitate CHILD OF
    judet CHILD OF
    regiune
);
```

Script Output x

Task completed in 0,142 seconds

Dimension HOTEL\_DIM\_OLAP created.

```
CREATE DIMENSION id_rezervare_dim_OLAP
LEVEL id_rezervare IS rezervare_camera_OLAP.id_rezervare
ATTRIBUTE id_rezervare DETERMINES (rezervare_camera_OLAP.id_perioada_rezervare,
rezervare_camera_OLAP.id_moment_efectuare_rezervare, rezervare_camera_OLAP.id_hotel,
rezervare_camera_OLAP.id_tip_client);
```

Worksheet Query Builder

```
CREATE DIMENSION id_rezervare_dim_OLAP
LEVEL id_rezervare IS rezervare_camera_OLAP.id_rezervare
ATTRIBUTE id_rezervare DETERMINES (rezervare_camera_OLAP.id_perioada, rezervare_camera_OLAP.id_moment_efectuare, rezervare_camera_OLAP.id_hotel, rezervare_camera_OLAP.id_tip_client);
```

Script Output x

Task completed in 0,065 seconds

Dimension ID\_REZERVARE\_DIM\_OLAP created.

8. (1p) Definirea partițiilor; definirea cererilor SQL însoțite de planul de execuție al acestora din care să reiasă ca optimizorul utilizează eficient partițiile.

```
-pentru hotel
PARTITION BY LIST (nr_stele)
(PARTITION o_stea VALUES (1),
PARTITION doua_stele VALUES (2),
PARTITION trei_stele VALUES (3),
PARTITION patru_stele VALUES (4),
PARTITION cinci_stele VALUES (5));

-pentru rezervare_camera
PARTITION BY RANGE(pret)
(PARTITION pret_sub_1000 VALUES LESS THAN (1000),
PARTITION pret_sub_2500 VALUES LESS THAN (2500),
PARTITION pret_sub_5000 VALUES LESS THAN (5000),
PARTITION pret_peste_5000 VALUES LESS THAN (MAXVALUE));
```

9. (2p) Optimizarea cererii SQL propusă în etapa de analiză

a. (1p) planul de execuție ales de optimizorul bazat pe cost (explicație etape parcurse)

Se consideră cererea SQL ce afișează numele hotelurilor ce au avut suma rezervărilor cuprinsă între

```
SELECT nume, SUM(pret) AS suma
FROM hotel_OLAP, rezervare_camera_OLAP
where rezervare_camera_olap.id_hotel = hotel_olap.id_hotel
GROUP BY nume
HAVING SUM(pret) between 1000 and 2000;
```

Worksheet		Query Builder	
		<pre>SELECT nume, SUM(pret) AS suma FROM hotel_OLAP, rezervare_camera_OLAP where rezervare_camera_olap.id_hotel = hotel_olap.id_hotel GROUP BY nume HAVING SUM(pret) between 1000 and 2000;</pre>	
Script Output x		Query Result x	
		SQL   All Rows Fetched: 9 in 0,032 seconds	
NUME		SUMA	
1 Hohe Rinne Paltinis Hotel Spa		1720	
2 Pensiunea Ioana		1410	
3 Opera Plaza Hotel		1440	
4 Ramada by Wyndham Oradea		1520	
5 UpperHouse		1600	
6 K+K Hotel Elisabeta		1300	
7 Casa Vacanza Brasov		1000	
8 Ivana Apart Hotel		1020	
9 Hotel Central Park		1360	

b. (1p) sugestii de optimizare a cererii, specificând planul de execuție obținut

Se poate crea o vizualizare materializată astfel:

```
CREATE MATERIALIZED VIEW pret_rezervare_camera_mw
BUILD IMMEDIATE
REFRESH FORCE
ENABLE QUERY REWRITE
AS
SELECT nume,
       SUM(pret) AS suma
FROM hotel_OLAP, rezervare_camera_OLAP
WHERE rezervare_camera_olap.id_hotel = hotel_olap.id_hotel
GROUP BY nume
HAVING SUM(pret) BETWEEN 10 AND 4000;
```



Worksheet Query Builder

```

CREATE MATERIALIZED VIEW pret_rezervare_camera_mw
BUILD IMMEDIATE
REFRESH FORCE
ENABLE QUERY REWRITE
AS
SELECT nume,
       SUM(pret) AS suma
FROM hotel_OLAP, rezervare_camera_OLAP
WHERE rezervare_camera_olap.id_hotel = hotel_olap.id_hotel
GROUP BY nume
HAVING SUM(pret) BETWEEN 10 AND 4000;

```

Script Output x

Task completed in 0,138 seconds

Materialized view PRET\_REZERVARE\_CAMERA\_MW created.

Iar cererea se poate rescrie astfel:

```

SELECT nume, suma
FROM pret_rezervare_camera_mw
WHERE suma>100 AND suma<2000;

```

Worksheet Query Builder

```

SELECT nume, suma
FROM pret_rezervare_camera_mw
WHERE suma>100 AND suma<2000;

```

Script Output x Query Result x

All Rows Fetched: 17 in 0,012 seconds

NUME	SUMA
1 Pensiunea Christiana	570
2 Hohe Rinne Paltinis Hotel Spa	1720
3 Pensiunea Ioana	1410
4 Hotel Casa Wagner	990
5 Opera Plaza Hotel	1440
6 Hotel Arnia	210
7 Ramada by Wyndham Oradea	1520
8 UpperHouse	1600
9 Casa Rozelor boutique hotel	720
10 K+K Hotel Elisabeta	1300
11 Platinia Hotel	770
12 Casa Vacanza Brasov	1000
13 Golden Tulip Ana Dome	190
14 Ivana Apart Hotel	1020
15 Mandachi Hotel Spa	740
16 Hotel Central Park	1360
17 Pension Bellagio	450

## 10. (2p) Crearea rapoartelor cu complexitate diferită (la acest nivel vor fi scripturi SQL, fără reprezentare grafică)

```
SELECT moment_efectuare_rezervare_olap.luna lunaefectuare, COUNT(tip_camera_olap.id_tip_camera) numar,
GROUPING_ID (luna) grouping_id
FROM rezervare_camera_olap, moment_efectuare_rezervare_olap, tip_camera_olap
WHERE rezervare_camera_olap.id_tip_camera= tip_camera_olap.id_tip_camera
AND rezervare_camera_olap.id_moment_efectuare= moment_efectuare_rezervare_olap.id_moment_efectuare
GROUP BY ROLLUP ( moment_efectuare_rezervare_olap.luna );
```

Worksheet		Query Builder	
		<pre>SELECT moment_efectuare_rezervare_olap.luna lunaefectuare, COUNT(tip_camera_olap.id_tip_camera) numar, GROUPING_ID (luna) grouping_id FROM rezervare_camera_olap, moment_efectuare_rezervare_olap, tip_camera_olap WHERE rezervare_camera_olap.id_tip_camera= tip_camera_olap.id_tip_camera AND rezervare_camera_olap.id_moment_efectuare= moment_efectuare_rezervare_olap.id_moment_efectuare GROUP BY ROLLUP ( moment_efectuare_rezervare_olap.luna );</pre>	
Script Output x		Query Result x	
		SQL   All Rows Fetched: 13 in 0,029 seconds	
LUNAEFFECTUARE	NUMAR	GROUPING_ID	
1 APR	13	0	
2 AUG	10	0	
3 DEC	10	0	
4 FEB	8	0	
5 IAN	19	0	
6 IUL	21	0	
7 IUN	10	0	
8 MAI	9	0	
9 MAR	13	0	
10 NOI	14	0	
11 OCT	16	0	
12 SEP	6	0	