

Proiectarea, implementarea și
exploatarea unei Baze de Date pentru o
rețea de socializare

Cuprins

1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.....	3
2. Constraineri impuse asupra modelului.....	3
3. Descrierea entităților.....	3
4. Descrierea relațiilor.....	4
5. Descrierea atributelor.....	5
6. Diagrama Entitate - Relație.....	5
7. Diagrama conceptuală.....	6
8. Enumerarea schemelor relaționale.....	7
9. Realizarea normalizării pana la forma normală 3.....	8
10. Crearea unei secvențe.....	12
11. Crearea tabelelor și inserarea datelor.....	13
1. Tabel UTILIZATORI.....	14
2. Tabel POSTARI.....	15
3. Tabel SCOR.....	17
4. Tabel ALERGEN.....	18
5. Tabel CATEGORII.....	20
6. Tabel POSTARE_CATEGORIE.....	21
7. Tabel URMARESTE.....	24
8. Tabel COMENTARII.....	25
9. Tabel MENU.....	26
10. Tabel APRECIAZA.....	29
11. Tabel LISTA_DORINTE.....	30
12. Tabel INSIGNE.....	32
13. Tabel GRUPURI.....	34
14. Tabel GRUP_UTILIZATOR.....	35
15. Tabel UTILIZATOR_INSIGNA.....	36
16. Tabel POSTARE_ALERGEN.....	39
17. Tabel UTILIZATOR_ALERGEN.....	41
12. Cereri SQL.....	42
13. Operații de actualizare si suprimare a datelor.....	48

14. Crearea unei vizualizări complexe.....	50
--	----

1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare

Se dorește implementarea unei aplicații de tip rețea de socializare care să promoveze în rândul utilizatorilor dorința de a mâncă mai informat și, astfel, mai echilibrat și mai sănătos. Rețeaua de socializare va fi o sursă de inspirație pentru mese mai diversificate și un sprijin pentru cei care aleg să urmeze un stil de viață mai sănătos, devenind membri într-o comunitate care împărtășește același scop.

Utilizatorii au posibilitatea de a crea postări (titlu, o descriere a preparatului, o imagine, eventual un link către locul din care s-a inspirat rețeta sau de unde a fost achiziționat preparatul, informații nutriționale, etc.).

Pe zi, un utilizator poate crea o singură postare. Postările se încadrează în anumite categorii (dulciuri, salate, feluri principale, milkshake-uri, etc.) și pot primi punctaje de la alți utilizatori. De asemenea, utilizatorii pot face comentarii la postări și pot adăuga postări la "lista de dorințe". Fiecare utilizator își poate crea meniuri alcătuite din maxim 5 postări. Pentru activitatea lor în aplicație, utilizatorii primesc insigne care să-i motiveze să rămână implicați. Fiecare utilizator poate menționa în profilul său alimente la care este alergic pentru o mai bună filtrare a postărilor vizualizate, iar fiecare postare va preciza ce alergeni conține preparatul promovat. Utilizatorii se pot afilia unui sau mai multor grupuri.

2. Constraineri impuse asupra modelului

Pentru a accesa aplicația, utilizatorii au nevoie să se înregistreze. Un utilizator poate crea un unic cont pe o adresă de mail. Pe zi, un utilizator poate crea o singură postare. Două postări diferite nu pot face referire la aceeași rețetă (să aibă același link către o rețetă).

3. Descrierea entităților

Entitatea UTILIZATORI cuprinde datele utilizatorilor aplicației, identificăți printr-un cod numeric (nu folosim username-ul pe post de cheie primară pentru a nu face căutare pe un sir de caractere). Entitatea POSTĂRI se referă la postările create de utilizatori, care au, de asemenea, drept cheie primară un cod numeric, pentru facilitarea folosirii acesteia drept cheie secundară în cadrul altor entități. ALERGENI reține diferite ingrediente evitate de utilizatori, precizate în contul fiecărui și în postările care le conțin, pentru a putea filtra postările astfel încât relevanța pentru utilizator să crească. Fiecare astfel de ingredient va avea un cod numeric drept identificator unic. Postările se încadrează în categorii, cuprinse de entitatea CATEGORII și identificate după un cod numeric. Pentru activitatea lor în aplicație, utilizatorii primesc

INSIGNE, având cheia primară un cod de numere. Utilizatorii pot crea sau se pot alia diferitelor GRUPURI din aplicație, comunități strânse în jurul unor interese comune. Fiecare grup primește la creare un sir unic de numere. Utilizatorii pot face comentarii la postări, grupate în entitatea COMENTARII. Fiecare comentariu este identificat unic printr-o cheie compusă din cheia primară a utilizatorului, cheia primară a postării și data la care a fost creat comentariul, astfel încât un utilizator poate comenta de mai multe ori aceeași postare. Un meniu din entitatea MENIURI primește un cod numeric unic, pentru a evita cheia primară compusă dată de cheia primară a utilizatorului și cheile primare ale celor 5 postări incluse într-un meniu (utilizatorul poate dori într-un meniu mai puțin de 5 postări, ceea ce ar da valori de null nepermise în compunerea cheii primare). Entitatea SCOR cuprinde gradele de apreciere pe care le poate avea un utilizator pentru o postare, iar cheia primară este reprezentată de numărul de stele.

4. Descrierea relațiilor

UTILIZATORI -creează- POSTĂRI: Un utilizator poate crea mai multe postări sau niciuna. Fiecare postare trebuie să aparțină unui utilizator și numai unui.

UTILIZATORI -listă_dorințe- POSTĂRI: Un utilizator poate "aprecia" (adăuga la "lista de dorințe" mai multe postări, dar nu este obligat să o facă. O postare poate fi apreciată de mai mulți utilizatori.

UTILIZATORI - POSTĂRI - SCOR: Un utilizator poate da același scor mai multor postări. Un utilizator poate da unei postări un singur scor, dar nu este obligat să o facă. O postare poate primi același scor de la mai mulți utilizatori.

UTILIZATORI –creează- COMENTARII: Un utilizator poate face mai multe comentarii, iar un comentariu trebuie să aparțină unui unic utilizator.

COMENTARII -făcute la- POSTĂRI: Un comentariu aparține unei postări și numai uneia, o postare poate avea mai multe comentarii. Relația are ca atribut textul comentariului.

UTILIZATORI -creează- MENIU: Un utilizator poate crea mai multe meniuri, dar un meniu trebuie să aparțină unui utilizator și numai unui.

MENIU -conține- POSTĂRI: Un meniu poate conține mai multe postări, dar trebuie să aibă minim una. O postare poate fi inclusă în unul sau mai multe meniuri.

POSTĂRI -apartine- CATEGORII: O postare aparține sau nu unei categorii, putând fi inclusă și în mai multe categorii. În schimb, o categorie poate cuprinde mai multe categorii, dar trebuie să conțină minim o postare.

UTILIZATORI -precizează- ALERGENI: Un utilizator poate sau nu preciza ingrediente evitate, iar fiecare alergen poate fi evitat de unul sau mai mulți utilizatori.

ALERGENI -precizează în- POSTĂRI: Un alergen poate fi precizat în mai multe postări sau în niciuna, o postare poate face referire la mai mulți alergeni pe care îi conține.

UTILIZATORI -urmăresc- UTILIZATORI: Un utilizator poate urmări mai mulți utilizatori, un utilizator poate fi urmărit de mai mulți utilizatori.

UTILIZATORI -apartin- GRUPURI: Un utilizator poate face parte din mai multe grupuri, dar nu este o condiție obligatorie. Un grup trebuie să aibă cel puțin un utilizator (administratorul).

UTILIZATORI -dețin- INSIGNE: Un utilizator poate deține mai multe insigne, iar o insignă poate fi oferită mai multor utilizatori.

5. Descrierea atributelor

Utilizatorii vor avea precizat un nume, un prenume, un username, un email (șiruri de caractere) și data creării contului (de tip dată). Atributul email nu poate fi NULL și este unic, la fel și username-ul.

Postările vor conține coloane pentru reținerea creatorului postării (FK), data creării postării (de tip dată), titlul, descrierea, link-ul către o rețetă și link-ul către o imagine sugestivă pentru postare (șiruri de caractere) și numărul de kilocalorii pentru o porție (număr).

Alergenii și categoriile vor preciza câte o denumire de tip sir de caractere, insignele vor mai avea și o descriere, iar comentariile vor conține un text.

Grupurile vor avea o trimitere către id-ul administratorului, o denumire și o descriere de tip sir de caractere, precum și data înființării de tip dată.

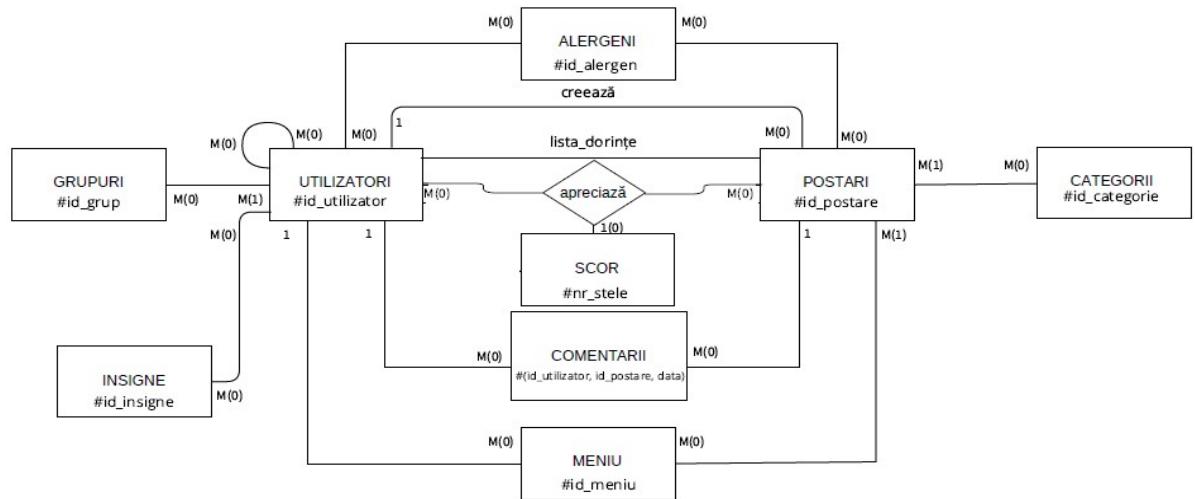
Meniurile vor avea o referință la id-ul utilizatorului căruia îi aparține, cinci atrbute pentru trimitere către postările incluse (pot avea și valori de null) și numărul total de kilocalorii conținute de meniu (de tip număr).

Scorul va avea pe lângă numărul de steluțe (între 1 și 5 inclusiv) o descriere a gradului de apreciere (de tip sir de caractere). De exemplu, 5 steluțe - "Ador (Un preparat extraordinar. Gust excepțional. Trebuie încercat neapărat.)".

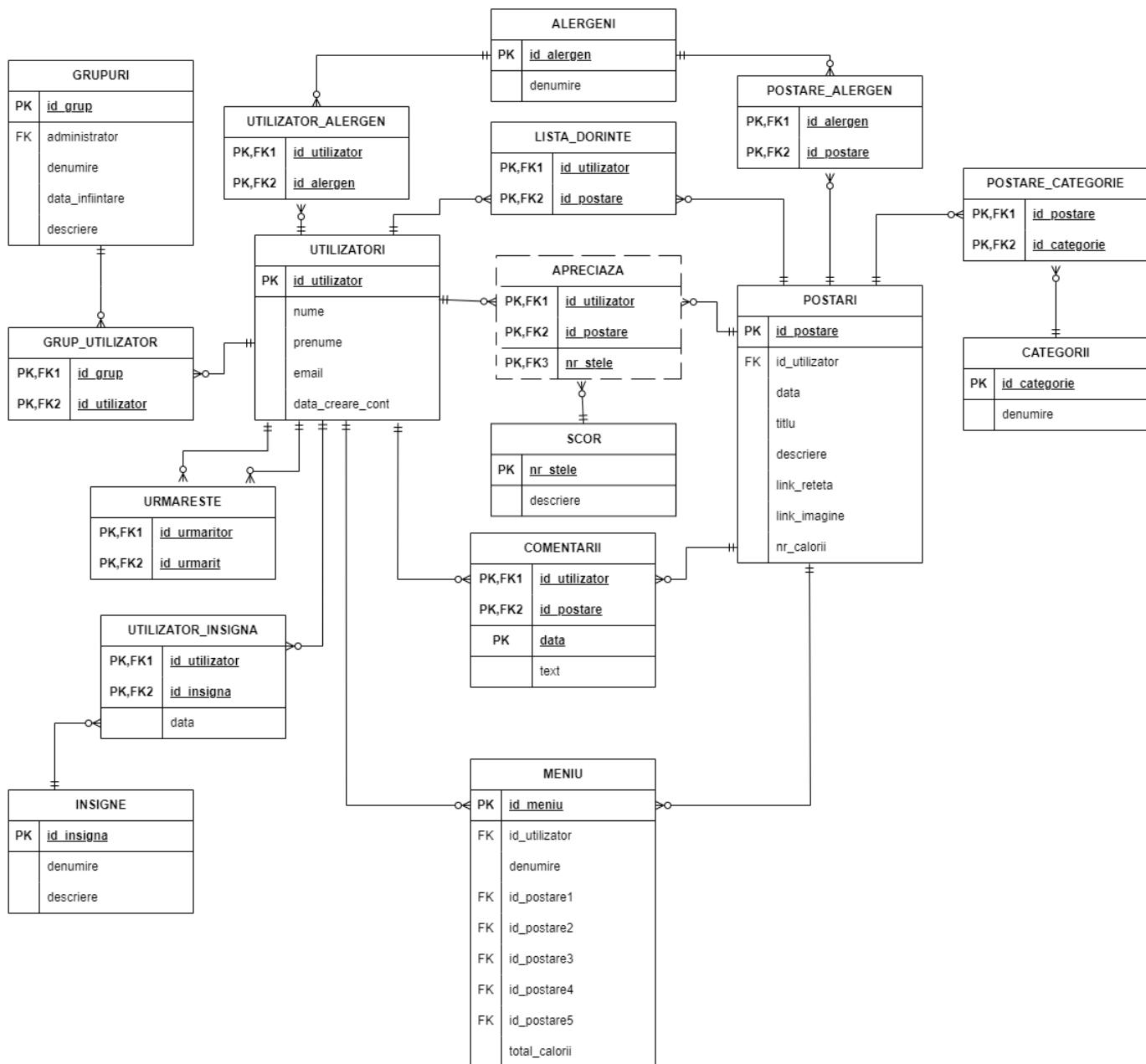
6. Diagrama Entitate - Relație



DIAGRAMA ENTITATE-RELATIE



7. Diagrama conceptuală



8. Enumerarea schemelor relaționale

UTILIZATORI (id_utilizator, nume, prenume, username, email, data_creare_cont)

POSTARI (id_postare, id_utilizator, data, titlu, descriere, link_reteta, link_imagine, nr_kcalorii)

CATEGORII (id_categorie, denumire)

ALERGENI (id_alergen, denumire)

GRUPURI (id_grup, administrator, denumire, descriere, data_infiintare)

INSIGNE (id_insigna, denumire, descriere)

SCOR (nr_stele, descriere)

COMENTARII (id_utilizator, id_postare, data, text)

MENIU (id_menu, id_utilizator, denumire, id_postare1, id_postare2, id_postare3, id_postare4, id_postare5, total_kcalorii)

APRECIAZA (id_utilizator, id_postare, nr_stele)

LISTA_DORINTE (id_utilizator, id_postare)

UTILIZATOR_ALERGEN (id_utilizator, id_alergen)

POSTARE_ALERGEN (id_alergen, id_postare)

POSTARE_CATEGORIE (id_postare, id_categorie)

UTILIZATOR_INSIGNA (id_utilizator, id_insigna)

GRUP_UTILIZATOR (id_grup, id_utilizator)

URMARESTE (id_utilizator1, id_utilizator2)

9. Realizarea normalizării pana la forma normală 3

POSTARI (id_postare, id_utilizator, data, titlu, descriere, link_reteta, link_imagine, nr_kcalorii)

Exemplu:

id_postare	id_utilizator	data	titlu	descriere	Link_reteta	Link_imagine	Nr_kcalorii
1	1	13-05-2023	Paine din faina de hrisca	"O reteta delicioasa de paine, fara gluten, pe care tocmai am incercat-o. Am ramas surprinsa de cat de pufoasa este si de gustul placut amarui."	http://www.veggiefit.ro/2018/04/paine-de-hrisca-fara-gluten-fermentata.html	https://2.bp.blogspot.com/-OFwT3zMOiAI/Wtr9JbA5JfI/AAAAAAAAN10/3L_LqJcwSJ4tIE-7YgUDbWcYE_LTnR--vwCLcBGAs/s1600/painehrisca2.jpg , https://www.juliana.ro/wp-content/uploads/2023/03/Paine-pufoasa-hrisca-vegana-fara-gluten.jpg	90

Relația nu este în FN1 din cauza valorilor multiple ale atributului link_imagine, inserate în aceeași celulă a tabelului.

Normalizare la FN1:

id_postare	id_utilizator	data	titlu	descriere	Link_reteta	Link_imagine	Nr_kcalorii
1	1	13-05-2023	Paine din faina de hrisca	"O reteta delicioasa de paine, fara gluten, pe care tocmai am incercat-o. Am ramas surprinsa de cat de pufoasa este si de gustul placut amarui."	http://www.veggiefit.ro/2018/04/paine-de-hrisca-fara-gluten-fermentata.html	https://2.bp.blogspot.com/-OFwT3zMOiAI/Wtr9JbA5JfI/AAAAAAAAN10/3L_LqJcwSJ4tIE-7YgUDbWcYE_LTnR--vwCLcBGAs/s1600/painehrisca2.jpg	90

1	1	13-05-2023	Paine din faina de hrisca	"O reteta delicioasa de paine, fara gluten, pe care tocmai am incercat-o. Am ramas surprinsa de cat de pufoasa este si de gustul placut amarui."	http://www.veggiefit.ro/2018/04/paine-de-hrisca-fara-gluten-fermentata.html	https://www.juliana.ro/wp-content/uploads/2023/03/Paine-pufoasa-hrisca-vegana-fara-gluten.jpg	90
---	---	------------	---------------------------	--	---	---	----

POSTARI (id_postare, id_utilizator, data, titlu, nume_utilizator, prenume_utilizator)

(id_postare) → (id_utilizator, data, titlu, descriere)

(id_utilizator) → (nume_utilizator, prenume_utilizator)

Exemplu:

id_postare	id_utilizator	data	titlu	nume_utilizator	prenume_utilizator
1	1	13-05-2023	Paine din faina de hrisca	Shona	Carolina
2	1	17-05-2023	Inghetata de fructe	Shona	Carolina

Relația nu este în FN3, deoarece atributele nume_utilizator și prenume_utilizator (care nu fac parte din cheia candidat) nu depind direct de cheia candidat id_postare. Id_postare determină id_utilizator, id_utilizator determină (nume_utilizator, prenume_utilizator) - dependență tranzitivă

Normalizare la FN3:

id_postare	id_utilizator	data	titlu
1	1	13-05-2023	Paine din faina de hrisca
2	1	17-05-2023	Inghetata de fructe
id_utilizator	nume_utilizator	prenume_utilizator	
1	Shona	Carolina	
1	Shona	Carolina	

UTILIZATOR_COMENTARIU (id_utilizator, nume, prenume, id_postare, data, text)

(id_utilizator) → (nume, prenume)

(id_utilizator, id_postare, data) → (text)

Relația nu este în FN2, deoarece attributele nume și prenume, care nu fac parte din cheia candidat (id_utilizator, id_postare, data) depind doar de o parte din cheia candidat (de id_utilizator). - dependență parțială

Exemplu:

id_utilizator	nume	prenume	id_postare	data	text
1	Flanagan	Michaela	1	18-05-2023	"Am incercat si eu aceasta reteta si este delicioasa."
1	Flanagan	Michaela	2	20-05-2023	"Un desert forte racoritor si sanatos. Perfect pentru zilele de vara."

Normalizare la FN2:

id_utilizator	nume	prenume
1	Flanagan	Michaela
1	Flanagan	Michaela

id_postare	id_utilizator	data	text
1	1	18-05-2023	"Am incercat si eu aceasta reteta si este delicioasa."
2	1	20-05-2023	"Un desert forte racoritor si sanatos. Perfect pentru zilele de vara."

10. Crearea unei secvențe

CREATE SEQUENCE sequence_PK

START WITH 1

INCREMENT BY 1

NOCYCLE;

Welcome Page docker_project

Worksheet Query Builder

```
CREATE SEQUENCE sequence_PK
START WITH 1
INCREMENT BY 1
NOCYCLE;
```

Script Output

Task completed in 0,141 seconds

Sequence SEQUENCE_PK created.

11. Crearea tabelelor și inserarea datelor

1. Tabel UTILIZATORI

```
CREATE TABLE utilizatori (
    id_utilizator NUMBER(10) PRIMARY KEY,
    nume VARCHAR2(30),
    prenume VARCHAR2(30),
    username VARCHAR(20) NOT NULL UNIQUE,
    email VARCHAR2(50) NOT NULL UNIQUE,
    data_creare_cont DATE DEFAULT SYSDATE
);
```

```
INSERT INTO utilizatori
VALUES (sequence_PK.nextval, 'Shona', 'Carolina', 'CarolinaS72', 'carolinash@gmail.com', '07-JUN-2019');
```

```
INSERT INTO utilizatori
VALUES (sequence_PK.nextval, 'Mayelis', 'Jacob', 'jacobM85', 'jacobm@gmail.com', '20-MAR-2020');
```

```
INSERT INTO utilizatori
VALUES (sequence_PK.nextval, 'Kunto', 'Adam', 'adamkunto', 'adamk@gmail.com', '13-JUL-2019');
```

```
INSERT INTO utilizatori
VALUES (sequence_PK.nextval, 'Jacobus', 'Charles', 'charles94', 'charlesj@gmail.com', SYSDATE);
```

```
INSERT INTO utilizatori
VALUES (sequence_PK.nextval, 'Toria', 'Adriene', 'adrieneT', 'adrienet@gmail.com', '15-DEC-2020');
```

```
INSERT INTO utilizatori
VALUES (sequence_PK.nextval, 'Stan', 'Faustino', 'faustinoStan', 'faustinoS@gmail.com', '12-NOV-2019');
```

```
INSERT INTO utilizatori
VALUES (sequence_PK.nextval, 'Sebastien', 'Denis', 'Sdenis96', 'deniss@gmail.com', '09-FEB-2019');
```

```
INSERT INTO utilizatori
VALUES (sequence_PK.nextval, 'Parvan', 'Enzo', 'enzoP87', 'enzop@gmail.com', '10-OCT-2018');
```

INSERT INTO utilizzatori

```
VALUES (sequence_PK.nextval, 'Hermann', 'Charlene', 'CharleneH89', 'charleneh@gmail.com',  
'18-APR-2021');
```

INSERT INTO utilizzatori

```
VALUES (sequence_PK.nextval, 'Slavko', 'Veronika', 'veronikaS92', 'veronikas@gmail.com',  
'02-JAN-2021');
```

The screenshot shows the Oracle SQL Developer interface with two tabs open:

- Worksheet**: Contains the initial SQL code for creating the table and inserting data.
- Query Builder**: Contains the modified SQL code where the first two insert statements have been removed.

Below the tabs is a toolbar with icons for running scripts, executing queries, and viewing output.

The results grid at the bottom displays the data inserted into the table:

ID_UTILIZZATORI	NUME	PRENOME	USERNAME	EMAIL	DATA_CREARE_CONT
1	Shona	Carolina	CarolinaS72	carolinasm@gmail.com	07-JUN-19
2	Mayelis	Jacob	jacobM85	jacobm@gmail.com	20-MAR-20
3	Kunto	Adam	adamkunto	adamk@gmail.com	13-JUL-19
4	Jacobus	Charles	charles94	charlesj@gmail.com	13-MAY-23
5	Stan	Faustina	faustinoStan	faustinos@gmail.com	12-NOV-19
6	Sebastien	Denis	Sdenis96	deniss@gmail.com	09-FEB-19
7	Parvan	Enzo	enzoP87	enzop@gmail.com	10-OCT-18
8	Hermann	Charlene	CharleneH89	charleneh@gmail.com	18-APR-21
9	Slavko	Veronika	veronikaS92	veronikas@gmail.com	02-JAN-21

2. Tabel POSTARI

```
CREATE TABLE postari (
    id_postare NUMBER(20) PRIMARY KEY,
    id_utilizator NUMBER(10),
    data DATE,
    titlu VARCHAR2(100),
    descriere VARCHAR2(500),
    link_reteta VARCHAR2(200) UNIQUE,
    link_imagine VARCHAR2(200),
    nr_kalorii NUMBER(6),
    CONSTRAINT post_ut_fk FOREIGN KEY (id_utilizator) REFERENCES utilizatori(id_utilizator) ON DELETE
SET NULL,
    CONSTRAINT ut_data_uk UNIQUE (id_utilizator, data)
);
```

INSERT INTO postari

```
VALUES (sequence_PK.nextval, 1, '08-MAY-2022', 'Paine din faina de hrisca ', 'O reteta delicioasa de
paine, fara gluten, pe care tocmai am incercat-o. Am ramas surprinsa de cat de pufoasa este si de gustul
placut amarui.', 'http://www.veggiefit.ro/2018/04/paine-de-hrisca-fara-gluten-fermentata.html',
'https://2.bp.blogspot.com/-OFwT3zMOiAI/Wtr9JbA5JfI/AAAAAAAANIO/3L\_LqJcwSJ4tIE-7YgUDbWcYELTnR--vwCLcBGAs/s1600/painehrisca2.jpeg', 90 );
```

INSERT INTO postari

```
VALUES (sequence_PK.nextval, 5, '24-JUL-2022', 'Inghetata de fructe ', 'Un desert racoritor perfect
pentru o zi calduroasa de vara. Si pe deasupra si sanatos...', 'https://www.bbcgoodfood.com/recipes/avocado-strawberry-ices',
'https://images.immediate.co.uk/production/volatile/sites/30/2020/08/avocado-strawberry-ices-copy-a3dcaf8.jpg', 94 );
```

INSERT INTO postari

```
VALUES (sequence_PK.nextval, 2, '10-DEC-2021', 'Supa vegana de dovleac', 'Sa ne amintim de zilele
placute de inceput de toamna cu aceasta supa colorata si aromata.', 'https://www.bbcgoodfood.com/recipes/vegan-pumpkin-soup',
'https://images.immediate.co.uk/production/volatile/sites/30/2020/08/vegan-pumpkin-soup-1d458f8.jpg', 256);
```

INSERT INTO postari

```
VALUES (sequence_PK.nextval, 1, '22-JUN-2022', 'Smoothie cu mere si zmeura', 'Un smoothie pentru
micul dejun. perfect pentru a-ti incepe ziua plin de energie.', 'https://www.bbcgoodfood.com/recipes/raspberry-and-apple-smoothie',
'https://images.immediate.co.uk/production/volatile/sites/30/2020/08/raspberry-smoothie-e687525.jpg', 106);
```

INSERT INTO postari

```
VALUES (sequence_PK.nextval, 7, '19-DEC-2022', 'Tort vegan', 'O reteta vegana delicioasa pentru clasicalul
tort de ciocolata. Pentru ca acum se apropie sarbatorile...',
```

```
'https://www.bbcgoodfood.com/recipes/vegan-birthday-cake',
'https://images.immediate.co.uk/production/volatile/sites/30/2020/08/vegan-birthday-cake-2-177c457.jpg', 697 );
```

INSERT INTO postari

```
VALUES (sequence_PK.nextval, 9, '22-MAY-2021', 'Briose fara gluten', 'Briose simple cu aroma de vanilie. Prietenii au fost foarte incantati.', 'https://www.bbcgoodfood.com/recipes/gluten-free-cupcakes', 'https://images.immediate.co.uk/production/volatile/sites/30/2022/09/GF-Cupcakes-bd4b78a.jpg', 430 );
```

INSERT INTO postari

```
VALUES (sequence_PK.nextval, 10, '23-APR-2021', 'Salata de paste', 'O salata de paste foarte rapid si usor de facut, pentru serile in care nu mai ai chef sa gatesti.', 'https://www.bbcgoodfood.com/recipes/deli-pasta-salad', 'https://images.immediate.co.uk/production/volatile/sites/30/2020/08/recipe-image-legacy-id-338583_11-1796dd4.jpg', 426 );
```

INSERT INTO postari

```
VALUES (sequence_PK.nextval, 7, '03-JAN-2023', 'Keto pizza', 'Daca tii dieta keto, aceasta varianta de pizza va fi perfecta pentru un rasfat culinar.', 'https://www.bbcgoodfood.com/recipes/keto-pizza', 'https://images.immediate.co.uk/production/volatile/sites/30/2022/01/Keto-Pizza-17a5061.jpg', 402 );
```

INSERT INTO postari

```
VALUES (sequence_PK.nextval, 8, '11-JUN-2019', 'Risotto cu rosii coapte, mozzarella si oregano', 'Exterior crocant, miez aromat, ingrediente delicioase. Aceasta reteta de risotto merita incercata.', 'https://www.bbcgoodfood.com/recipes/baked-tomato-mozzarella-basil-risotto', 'https://images.immediate.co.uk/production/volatile/sites/30/2021/04/Baked-tomato-mozzarella-and-basil-risotto-368464f.jpg', 504 );
```

INSERT INTO postari

```
VALUES (sequence_PK.nextval, 1, '22-FEB-2020', 'Cheesecake cu banane coapte', 'Un cheesecake delicios si cu o cantitate scazut a de zahar. O varianta excelenta de desert pentru persoanele cu diabet.', 'https://www.bbcgoodfood.com/recipes/baked-banana-cheesecake', 'https://images.immediate.co.uk/production/volatile/sites/30/2020/08/baked-banana-cheesecake-dac4a04.jpg', 299 );
```

3. Tabel SCOR

```
CREATE TABLE scor (
    nr_stele NUMBER(1) PRIMARY KEY,
    descriere VARCHAR2(70)
);
```

INSERT INTO scor

```
VALUES (sequence_PK.nextval, 'Nu as repeta experienta.');
```

```

INSERT INTO scor
VALUES (sequence_PK.nextval, 'Acceptabil, dar ar trebui imbunatatita reteta.');

INSERT INTO scor
VALUES (sequence_PK.nextval, 'O reteta buna, sanatoasa, dar cu nimic speciala.');

INSERT INTO scor
VALUES (sequence_PK.nextval, 'Chiar mi-a placut. Cred ca pe viitor o voi gati din nou.');

INSERT INTO scor
VALUES (sequence_PK.nextval, 'Extraordinara. O reteta memorabila. Trebuie neaparat incercata.');

```

```

79: -- tabel SCOR --
80: CREATE TABLE scor (
81:     nr_stele NUMBER(1) PRIMARY KEY,
82:     descriere VARCHAR2(70)
83: );
84: INSERT INTO scor
85: VALUES (sequence_PK.nextval, 'Nu as repeta experienta.');
86: INSERT INTO scor
87: VALUES (sequence_PK.nextval, 'Acceptabil, dar ar trebui imbunatatita reteta.');
88: INSERT INTO scor
89: VALUES (sequence_PK.nextval, 'O reteta buna, sanatoasa, dar cu nimic speciala.');
90: INSERT INTO scor
91: VALUES (sequence_PK.nextval, 'Chiar mi-a placut. Cred ca pe viitor o voi gati din nou.');
92: INSERT INTO scor
93: VALUES (sequence_PK.nextval, 'Extraordinara. O reteta memorabila. Trebuie neaparat incercata.');
94:
95: SELECT * FROM scor;

```

Script Output | Query Result | All Rows Fetched: 5 in 0,006 seconds

NR_STELE	DESCRIERE
1	Nu as repeta experienta.
2	Acceptabil, dar ar trebui imbunatatita reteta.
3	O reteta buna, sanatoasa, dar cu nimic speciala.
4	Chiar mi-a placut. Cred ca pe viitor o voi gati din nou.
5	Extraordinara. O reteta memorabila. Trebuie neaparat incercata.

4. Tabel ALERGEN

```

CREATE TABLE alergen (
    id_alergen NUMBER(3) PRIMARY KEY,
    descriere VARCHAR2(50)
);

INSERT INTO alergen
VALUES (sequence_PK.nextval, 'gluten');

INSERT INTO alergen
VALUES (sequence_PK.nextval, 'lactate');

INSERT INTO alergen

```

```
VALUES (sequence_PK.nextval, 'fructe cu coaja lemoasa');

INSERT INTO alergen
VALUES (sequence_PK.nextval, 'oua');

INSERT INTO alergen
VALUES (sequence_PK.nextval, 'zahar');

INSERT INTO alergen
VALUES (sequence_PK.nextval, 'telina');

INSERT INTO alergen
VALUES (sequence_PK.nextval, 'carne');

INSERT INTO alergen
VALUES (sequence_PK.nextval, 'soia');

INSERT INTO alergen
VALUES (sequence_PK.nextval, 'hrisca');

INSERT INTO alergen
VALUES (sequence_PK.nextval, 'banana');

INSERT INTO alergen
VALUES (sequence_PK.nextval, 'ovaz');

INSERT INTO alergen
VALUES (sequence_PK.nextval, 'orez');

INSERT INTO alergen
VALUES (sequence_PK.nextval, 'capsuna');

INSERT INTO alergen
VALUES (sequence_PK.nextval, 'mustar');
```

```

worksheet  Query Builder
121: INSERT INTO alergen
122: VALUES (sequence_PK.nextval, 'banana');
123: INSERT INTO alergen
124: VALUES (sequence_PK.nextval, 'ovaz');
125: INSERT INTO alergen
126: VALUES (sequence_PK.nextval, 'orez');
127: INSERT INTO alergen
128: VALUES (sequence_PK.nextval, 'capsuna');
129: INSERT INTO alergen
130: VALUES (sequence_PK.nextval, 'mustar');
131:
132: SELECT * FROM alergen;

```

Script Output X | Query Result X | SQL | All Rows Fetched: 14 in 0,006 seconds

ID_ALERGEN	DESCRIERE
1	gluten
2	lactate
3	fructe cu coaja lemnosa
4	oua
5	zahar
6	telina
7	carne
8	soia
9	hrisca
10	banana
11	ovaz
12	orez
13	capsuna
14	mustar

5. Tabel CATEGORII

```

CREATE TABLE categorii (
    id_categorie NUMBER(5) PRIMARY KEY,
    denumire VARCHAR2(50)
);

INSERT INTO categorii
VALUES (sequence_PK.nextval, 'Ciorbe si supe');

INSERT INTO categorii
VALUES (sequence_PK.nextval, 'Salate');

INSERT INTO categorii
VALUES (sequence_PK.nextval, 'Idei pentru cina');

INSERT INTO categorii
VALUES (sequence_PK.nextval, 'Pizza si Paste');

INSERT INTO categorii
VALUES (sequence_PK.nextval, 'Garnituri');

INSERT INTO categorii
VALUES (sequence_PK.nextval, 'Deserturi');

```

```
INSERT INTO categorii  
VALUES (sequence_PK.nextval, 'Bauturi');
```

```
INSERT INTO categorii  
VALUES (sequence_PK.nextval, 'Inghetate');
```

```
INSERT INTO categorii  
VALUES (sequence_PK.nextval, 'Preparate vegane');
```

```
INSERT INTO categorii  
VALUES (sequence_PK.nextval, 'Preparate cu fructe');
```

Worksheet Query Builder

```
-- tabel CATEGORII --
CREATE TABLE categorii (
    id_categorie NUMBER(5) PRIMARY KEY,
    denumire VARCHAR2(50)
);
INSERT INTO categorii
VALUES (sequence_PK.nextval, 'Ciorbe si supe');
INSERT INTO categorii
VALUES (sequence_PK.nextval, 'Salate');
INSERT INTO categorii
VALUES (sequence_PK.nextval, 'Idei pentru cina');
INSERT INTO categorii
VALUES (sequence_PK.nextval, 'Pizza si Paste');
INSERT INTO categorii
VALUES (sequence_PK.nextval, 'Garnituri');
INSERT INTO categorii
VALUES (sequence_PK.nextval, 'Deserturi');
INSERT INTO categorii
VALUES (sequence_PK.nextval, 'Bauturi');
INSERT INTO categorii
VALUES (sequence_PK.nextval, 'Inghetate');
INSERT INTO categorii
VALUES (sequence_PK.nextval, 'Preparate vegane');
INSERT INTO categorii
VALUES (sequence_PK.nextval, 'Preparate cu fructe');

INSERT INTO categorii
VALUES (sequence_PK.nextval, 'Garnituri');
INSERT INTO categorii
VALUES (sequence_PK.nextval, 'Deserturi');
INSERT INTO categorii
VALUES (sequence_PK.nextval, 'Bauturi');
INSERT INTO categorii
VALUES (sequence_PK.nextval, 'Inghetate');
INSERT INTO categorii
VALUES (sequence_PK.nextval, 'Preparate vegane');
INSERT INTO categorii
VALUES (sequence_PK.nextval, 'Preparate cu fructe');

SELECT * FROM categorii;
```

Script Output Query Result

All Rows Fetched: 10 in 0,012 seconds

ID_CATEGORIE	DENUMIRE
1	Ciorbe si supe
2	Salate
3	Idei pentru cina
4	Pizza si Paste
5	Garnituri
6	Deserturi
7	Bauturi
8	Inghetate
9	Preparate vegane
10	Preparate cu fructe

6. Tabel POSTARE_CATEGORIE

```
CREATE TABLE postare_categorie (
    id_postare NUMBER(20),
    id_categorie NUMBER(5),
    CONSTRAINT post_cat_pk PRIMARY KEY (id_postare, id_categorie),
    CONSTRAINT post_fk FOREIGN KEY (id_postare) REFERENCES postari(id_postare) ON DELETE CASCADE,
```

```
    CONSTRAINT cat_fk FOREIGN KEY(id_categorie) REFERENCES categorii(id_categorie) ON DELETE
CASCADE
);
INSERT INTO postare_categorie
VALUES (3,1);

INSERT INTO postare_categorie
VALUES (7,2);

INSERT INTO postare_categorie
VALUES (3,3);

INSERT INTO postare_categorie
VALUES (7,3);

INSERT INTO postare_categorie
VALUES (8,3);

INSERT INTO postare_categorie
VALUES (9,3);

INSERT INTO postare_categorie
VALUES (7,4);

INSERT INTO postare_categorie
VALUES (8,4);

INSERT INTO postare_categorie
VALUES (7,5);

INSERT INTO postare_categorie
VALUES (9,5);

INSERT INTO postare_categorie
VALUES (2,6);

INSERT INTO postare_categorie
VALUES (4,6);

INSERT INTO postare_categorie
VALUES (5,6);

INSERT INTO postare_categorie
VALUES (6,6);

INSERT INTO postare_categorie
VALUES (10,6);
```

```
INSERT INTO postare_categorie  
VALUES (4,7);
```

```
INSERT INTO postare_categorie  
VALUES (2,8);
```

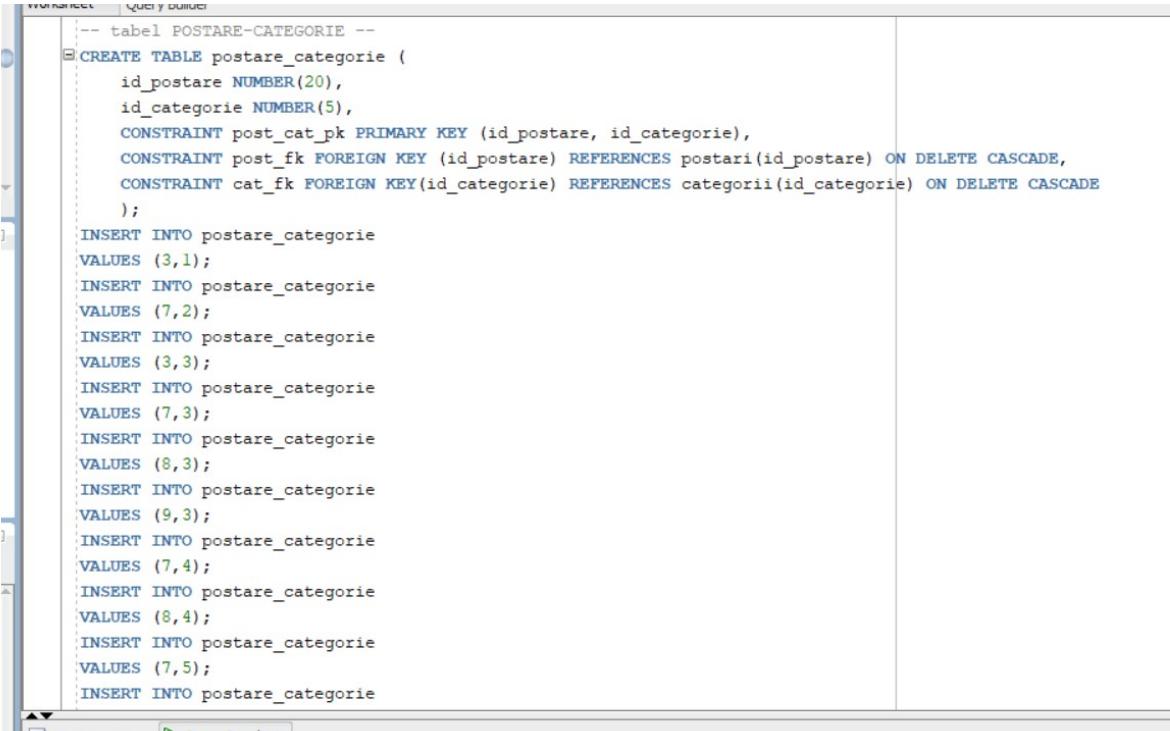
```
INSERT INTO postare_categorie  
VALUES (3,9);
```

```
INSERT INTO postare_categorie  
VALUES (7,9);
```

```
INSERT INTO postare_categorie  
VALUES (2,10);
```

```
INSERT INTO postare_categorie  
VALUES (4,10);
```

```
INSERT INTO postare_categorie  
VALUES (10,10);
```



```
-- tabel POSTARE-CATEGORIE --  
CREATE TABLE postare_categorie (  
    id_postare NUMBER(20),  
    id_categorie NUMBER(5),  
    CONSTRAINT post_cat_pk PRIMARY KEY (id_postare, id_categorie),  
    CONSTRAINT post_fk FOREIGN KEY (id_postare) REFERENCES postari(id_postare) ON DELETE CASCADE,  
    CONSTRAINT cat_fk FOREIGN KEY(id_categorie) REFERENCES categorii(id_categorie) ON DELETE CASCADE  
);  
INSERT INTO postare_categorie  
VALUES (3,1);  
INSERT INTO postare_categorie  
VALUES (7,2);  
INSERT INTO postare_categorie  
VALUES (3,3);  
INSERT INTO postare_categorie  
VALUES (7,3);  
INSERT INTO postare_categorie  
VALUES (8,3);  
INSERT INTO postare_categorie  
VALUES (9,3);  
INSERT INTO postare_categorie  
VALUES (7,4);  
INSERT INTO postare_categorie  
VALUES (8,4);  
INSERT INTO postare_categorie  
VALUES (7,5);  
INSERT INTO postare_categorie
```

```

Worksheet   Query Builder
VALUES (1,5);
INSERT INTO postare_categorie
VALUES (9,5);
INSERT INTO postare_categorie
VALUES (2,6);
INSERT INTO postare_categorie
VALUES (4,6);
INSERT INTO postare_categorie
VALUES (5,6);
INSERT INTO postare_categorie
VALUES (6,6);
INSERT INTO postare_categorie
VALUES (10,6);
INSERT INTO postare_categorie
VALUES (4,7);
INSERT INTO postare_categorie
VALUES (2,8);
INSERT INTO postare_categorie
VALUES (3,9);
INSERT INTO postare_categorie
VALUES (7,9);
INSERT INTO postare_categorie
VALUES (2,10);
INSERT INTO postare_categorie
VALUES (4,10);
INSERT INTO postare_categorie
VALUES (10,10);

SELECT *
FROM postare_categorie;

Script Output x | Query Result x | All Rows Fetched: 22 in 0,007 seconds
ID_POSTARE | ID_CATEGORIE |
1           | 2             | 6
2           | 2             | 8
3           | 2             | 10
4          | 3             | 1
5          | 3             | 3
6          | 3             | 9
7          | 4             | 6
8          | 4             | 7
9          | 4             | 10
10         | 5             | 6
11         | 6             | 6
12         | 7             | 2
13         | 7             | 3
14         | 7             | 4
15         | 7             | 5
16         | 7             | 9
17         | 8             | 3
18         | 8             | 4
19         | 9             | 3
20         | 9             | 5
21         | 10            | 6
22         | 10            | 10

```

7. Tabel URMARESTE

```

CREATE TABLE urmareste (
    id_urmaritor NUMBER(10),
    id_urmarit NUMBER(10),
    CONSTRAINT urmarest_pk PRIMARY KEY (id_urmaritor, id_urmarit),
    CONSTRAINT urmaritor_fk FOREIGN KEY (id_urmaritor) REFERENCES utilizatori(id_utilizator) ON DELETE CASCADE,
    CONSTRAINT urmarit_fk FOREIGN KEY (id_urmarit) REFERENCES utilizatori(id_utilizator) ON DELETE CASCADE
);

```

```

INSERT INTO urmareste
VALUES (5,1);

```

```

INSERT INTO urmareste
VALUES (10,7);

```

```

INSERT INTO urmareste
VALUES (7,10);

```

```

INSERT INTO urmareste
VALUES (9,7);

```

```

INSERT INTO urmareste
VALUES (5,1);

```

```

INSERT INTO urmareste

```

```
VALUES (2,7);
```

```
INSERT INTO urmareste  
VALUES (7,2);
```

```
INSERT INTO urmareste  
VALUES (8,10);
```

The screenshot shows the Oracle SQL Developer interface with three main panes:

- Top Left Window (Query Builder):** Contains the following SQL code:

```
-- tabel URMARESTE --  
CREATE TABLE urmareste (  
    id_urmaritor NUMBER(10),  
    id_urmarit NUMBER(10),  
    CONSTRAINT urmreste_pk PRIMARY KEY (id_urmaritor, id_urmarit),  
    CONSTRAINT urmaritor_fk FOREIGN KEY (id_urmaritor) REFERENCES utilizatori(id_utilizator) ON DELETE CASCADE,  
    CONSTRAINT urmarit_fk FOREIGN KEY (id_urmarit) REFERENCES utilizatori(id_utilizator) ON DELETE CASCADE  
);  
INSERT INTO urmareste  
VALUES (5,1);  
INSERT INTO urmareste  
VALUES (10,7);  
INSERT INTO urmareste  
VALUES (7,10);  
INSERT INTO urmareste  
VALUES (9,7);  
INSERT INTO urmareste  
VALUES (5,1);  
INSERT INTO urmareste  
VALUES (2,7);  
INSERT INTO urmareste  
VALUES (7,2);  
INSERT INTO urmareste  
VALUES (8,10);  
  
SELECT *  
FROM urmareste;
```
- Bottom Left Window (Query Builder):** Contains the same SQL code as the top window.
- Bottom Right Window (Script Output / Query Result):** Shows the results of the SELECT query. The table has columns ID_URMARITOR and ID_URMARIT, with the following data:

ID_URMARITOR	ID_URMARIT
1	2
2	5
3	7
4	7
5	8
6	9
7	10

8. Tabel COMENTARII

```
CREATE TABLE comentarii (  
    id_utilizator NUMBER(10),  
    id_postare NUMBER(20),  
    data DATE,  
    text VARCHAR2(500),
```

```
CONSTRAINTS com_pk PRIMARY KEY (id_utilizator, id_postare),
CONSTRAINT com_ut_fk FOREIGN KEY (id_utilizator) REFERENCES utilizatori(id_utilizator),
CONSTRAINT com_post_fk FOREIGN KEY (id_postare) REFERENCES postari(id_postare)
);
```

INSERT INTO comentarii

```
VALUES (1, 5, '7-DEC-2022', 'Ador acest tort vegan. L-am preparat pentru familia si prietenii mei pentru masa de Craciun. A fost absolut delicios si a avut mare succes. Il voi prepara si pentru invitati de Revelion. Va fi surpriza serii.');
```

INSERT INTO comentarii

```
VALUES (8, 4, '25-JUN-2022', 'Am incercat aceasta reteta imediat cum am vazut postarea. Imi era pofta de ceva dulce si racoritor. Racoritor a fost, dar i-am adaugat si miere pentru a fi putin mai dulce.');
```

INSERT INTO comentarii

```
VALUES (5, 4, '28-JUN-2022', 'Cand am vazut combinatia de mere cu zmeura, am fost putin sceptica... as fi pus alte fructe de sezon. Dar am incercat. Si nu regret. Gustul este surprinzator.');
```

INSERT INTO comentarii

```
VALUES (2, 1, '15-MAY-2022', 'Am intoleranta la gluten si sunt in cautarea unui inlocuitor de paine. Din pacate, nici acesta reteta nu este ceea ce trebuie. Pinea este densa si cu un gust puternic amar. Nici aspectul nu este prea placut.');
```

INSERT INTO comentarii

```
VALUES (9, 7, '01-MAY-2021', 'Foarte usoara si satioasa aceasta reteta de paste. Si foarte usor de facut.');
```

```

-- tabel COMENTARII --
CREATE TABLE comentarii (
    id_utilizator NUMBER(10),
    id_postare NUMBER(20),
    data DATE,
    text VARCHAR2(500),
    CONSTRAINT com_pk PRIMARY KEY (id_utilizator, id_postare),
    CONSTRAINT com_ut_fk FOREIGN KEY (id_utilizator) REFERENCES utilizatori(id_utilizator),
    CONSTRAINT com_post_fk FOREIGN KEY (id_postare) REFERENCES postari(id_postare)
);
INSERT INTO comentarii
VALUES (1, 5, '7-DEC-2022', 'Ador acest tort vegan. L-am preparat pentru familia si prietenii mei pentru masa de Craciun. A fost absolut delicios si a av');
INSERT INTO comentarii
VALUES (8, 4, '25-JUN-2022', 'Am incercat aceasta reteta imediat cum am vazut postarea. Imi era pofta de ceva dulce si racoritor. Racoritor a fost, dar i');
INSERT INTO comentarii
VALUES (5, 4, '28-JUN-2022', 'Cand am vazut combinatia de mere cu zmeura, am fost putin sceptica... as fi pus alte fructe de sezon. Dar am incercat. Si n');
INSERT INTO comentarii
VALUES (2, 1, '15-MAY-2022', 'Am intoleranta la gluten si sunt in cautarea unui inlocuitor de paine. Din pacate, nici acesta reteta nu este ceea ce trebuie');
INSERT INTO comentarii
VALUES (9, 7, '01-MAY-2021', 'Foarte usoara si saticasa aceasta reteta de paste. Si foarte usor de facut.');
SELECT *
FROM comentarii;

```

Worksheet **Query Builder**

```

INSERT INTO comentarii
VALUES (1, 5, '7-DEC-2022', 'Ador acest tort vegan. L-am preparat pentru familia si prietenii mei pentru masa de Craciun. A fost absolut delicios si a av');
INSERT INTO comentarii
VALUES (8, 4, '25-JUN-2022', 'Am incercat aceasta reteta imediat cum am vazut postarea. Imi era pofta de ceva dulce si racoritor. Racoritor a fost, dar i');
INSERT INTO comentarii
VALUES (5, 4, '28-JUN-2022', 'Cand am vazut combinatia de mere cu zmeura, am fost putin sceptica... as fi pus alte fructe de sezon. Dar am incercat. Si n');
INSERT INTO comentarii
VALUES (2, 1, '15-MAY-2022', 'Am intoleranta la gluten si sunt in cautarea unui inlocuitor de paine. Din pacate, nici acesta reteta nu este ceea ce trebuie');
INSERT INTO comentarii
VALUES (9, 7, '01-MAY-2021', 'Foarte usoara si saticasa aceasta reteta de paste. Si foarte usor de facut.');

SELECT *
FROM comentarii;

```

Script Output **Query Result**

All Rows Fetched: 5 in 0,012 seconds

ID_UTILIZATOR	ID_POSTARE	DATA	TEXT
1	1	5-07-DEC-22	Ador acest tort vegan. L-am preparat pentru familia si prietenii mei pentru masa de Craciun. A fost absolut delicios si a av
2	8	4-25-JUN-22	Am incercat aceasta reteta imediat cum am vazut postarea. Imi era pofta de ceva dulce si racoritor. Racoritor a fost, dar i
3	5	4-28-JUN-22	Cand am vazut combinatia de mere cu zmeura, am fost putin sceptica... as fi pus alte fructe de sezon. Dar am incercat. Si n
4	2	11-15-MAY-22	Am intoleranta la gluten si sunt in cautarea unui inlocuitor de paine. Din pacate, nici acesta reteta nu este ceea ce trebuie
5	9	7-01-MAY-21	Foarte usoara si saticasa aceasta reteta de paste. Si foarte usor de facut.

9. Tabel MENIU

CREATE TABLE meniu (

```

id_menu NUMBER(5),
id_utilizator NUMBER(10),
denumire VARCHAR2(50),
id_postare1 NUMBER(20),
id_postare2 NUMBER(20),
id_postare3 NUMBER(20),
id_postare4 NUMBER(20),
id_postare5 NUMBER(20),
total_kcalorii NUMBER(7),
CONSTRAINT meniu_pk PRIMARY KEY (id_menu),
CONSTRAINT p1_fk FOREIGN KEY (id_postare1) REFERENCES postari(id_postare),
CONSTRAINT p2_fk FOREIGN KEY (id_postare2) REFERENCES postari(id_postare),
CONSTRAINT p3_fk FOREIGN KEY (id_postare3) REFERENCES postari(id_postare),
CONSTRAINT p4_fk FOREIGN KEY (id_postare4) REFERENCES postari(id_postare),

```

```
CONSTRAINT p5_fk FOREIGN KEY (id_postare5) REFERENCES postari(id_postare),
CONSTRAINT meniu_ut_fk FOREIGN KEY (id_utilizator) REFERENCES utilizatori(id_utilizator)
);
```

```
INSERT INTO meniu
VALUES (sequence_PK.nextval, 1, 'Cina de vara', 3, 7, 6, NULL, NULL, (
    SELECT SUM(nr_kcalorii)
    FROM postari
    WHERE id_postare IN (3, 7, 6)
));
```

```
INSERT INTO meniu
VALUES (sequence_PK.nextval, 10, 'Cina italieneasca', 8, 7, 10, 2, NULL, (
    SELECT SUM(nr_kcalorii)
    FROM postari
    WHERE id_postare IN (8, 7, 10, 2)
));
```

```
INSERT INTO meniu
VALUES (sequence_PK.nextval, 1, 'Pranz aromat', 3, 1, 9, 4, NULL, (
    SELECT SUM(nr_kcalorii)
    FROM postari
    WHERE id_postare IN (3, 1, 9, 4)
));
```

```
INSERT INTO meniu
VALUES (sequence_PK.nextval, 3, 'Petrecere', 8, 7, 10, 2, 5, (
    SELECT SUM(nr_kcalorii)
    FROM postari
    WHERE id_postare IN (8, 7, 10, 2, 5)
));
```

```
INSERT INTO meniu
VALUES (sequence_PK.nextval, 9, 'Gustarele pentru ceai', 6, 10, 2, 4, NULL, (
    SELECT SUM(nr_kcalorii)
    FROM postari
    WHERE id_postare IN (6, 10, 2, 4)
));
```

```
INSERT INTO meniu
VALUES (sequence_PK.nextval, 8, 'Gustare rapida', 6, 4, NULL, NULL, NULL, (
    SELECT SUM(nr_kcalorii)
    FROM postari
    WHERE id_postare IN (6, 4)
));
```

```
INSERT INTO meniu
VALUES (sequence_PK.nextval, 7, 'Cina usoara', 1, 3, NULL, NULL, NULL, (
    SELECT SUM(nr_kcalorii)
    FROM postari
    WHERE id_postare IN (1, 3)
));
```

```
INSERT INTO meniu
VALUES (sequence_PK.nextval, 4, 'Pranz', 9, 10, NULL, NULL, NULL, (
```

```

        SELECT SUM(nr_kcalorii)
        FROM postari
        WHERE id_postare IN (9, 10)
    ));

INSERT INTO meniu
VALUES (sequence_PK.nextval, 4, 'Pranz vegan', 3, 5, NULL, NULL, NULL, (
        SELECT SUM(nr_kcalorii)
        FROM postari
        WHERE id_postare IN (3, 5)
    ));

INSERT INTO meniu
VALUES (sequence_PK.nextval, 3, 'Gustare dulce', 10, 4, NULL, NULL, NULL, (
        SELECT SUM(nr_kcalorii)
        FROM postari
        WHERE id_postare IN (10, 4)
    ));

```

The screenshot shows the Oracle SQL Developer interface with a query editor titled "Querry builder" containing the following SQL code:

```

CREATE TABLE meniu (
    id_menu NUMBER(5),
    id_utilizator NUMBER(10),
    denumire VARCHAR2(50),
    id_postare1 NUMBER(20),
    id_postare2 NUMBER(20),
    id_postare3 NUMBER(20),
    id_postare4 NUMBER(20),
    id_postare5 NUMBER(20),
    total_kcalorii NUMBER(7),
    CONSTRAINT meniu_pk PRIMARY KEY (id_menu),
    CONSTRAINT p1_fk FOREIGN KEY (id_postare1) REFERENCES postari(id_postare),
    CONSTRAINT p2_fk FOREIGN KEY (id_postare2) REFERENCES postari(id_postare),
    CONSTRAINT p3_fk FOREIGN KEY (id_postare3) REFERENCES postari(id_postare),
    CONSTRAINT p4_fk FOREIGN KEY (id_postare4) REFERENCES postari(id_postare),
    CONSTRAINT p5_fk FOREIGN KEY (id_postare5) REFERENCES postari(id_postare),
    CONSTRAINT meniu_ut_fk FOREIGN KEY (id_utilizator) REFERENCES utilizatori(id_utilizator)
);

```

INSERT INTO meniu

```

VALUES (sequence_PK.nextval, 1, 'Cina de vara', 3, 7, 6, NULL, NULL, (
        SELECT SUM(nr_kcalorii)
        FROM postari
        WHERE id_postare IN (3, 7, 6)
    ));

```

INSERT INTO meniu

```

VALUES (sequence_PK.nextval, 10, 'Cina italieneasca', 8, 7, 10, 2, NULL, (
    ...
));

```

Worksheet Query Builder

```

    ;
    INSERT INTO meniu
    VALUES (sequence_PK.nextval, 1, 'Cina de vara', 3, 7, 6, NULL, NULL, (
        SELECT SUM(nr_kcalorii)
        FROM postari
        WHERE id_postare IN (3, 7, 6)
    ));

    INSERT INTO meniu
    VALUES (sequence_PK.nextval, 10, 'Cina italieneasca', 8, 7, 10, 2, NULL, (
        SELECT SUM(nr_kcalorii)
        FROM postari
        WHERE id_postare IN (8, 7, 10, 2)
    ));

    INSERT INTO meniu
    VALUES (sequence_PK.nextval, 1, 'Franz aromat', 3, 1, 9, 4, NULL, (
        SELECT SUM(nr_kcalorii)
        FROM postari
        WHERE id_postare IN (3, 1, 9, 4)
    ));

    INSERT INTO meniu
    VALUES (sequence_PK.nextval, 3, 'Petrecere', 8, 7, 10, 2, 5, (
        SELECT SUM(nr_kcalorii)
        FROM postari
        WHERE id_postare IN (8, 7, 10, 2, 5)
    ));

    INSERT INTO meniu
    VALUES (sequence_PK.nextval, 9, 'Gustarele pentru ceai', 6, 10, 2, 4, NULL, (
        SELECT SUM(nr_kcalorii)
        FROM postari
        WHERE id_postare IN (6, 10, 2, 4)
    ));

    INSERT INTO meniu
    VALUES (sequence_PK.nextval, 4, 'Pranz vegan', 3, 5, NULL, NULL, NULL, (
        SELECT SUM(nr_kcalorii)
        FROM postari
        WHERE id_postare IN (3, 5)
    ));

    INSERT INTO meniu
    VALUES (sequence_PK.nextval, 3, 'Gustare dulce', 10, 4, NULL, NULL, NULL, (
        SELECT SUM(nr_kcalorii)
        FROM postari
        WHERE id_postare IN (10, 4)
    ));

```

Script Output | Query Result | SQL | All Rows Fetched: 10 in 0,003 seconds

ID_MENUU	ID_UTILIZATOR	DENUMIRE	ID_POSTARE1	ID_POSTARE2	ID_POSTARE3	ID_POSTARE4	ID_POSTARE5	TOTAL_KCALORII
1	1	1 Cina de vara	3	7	6	(null)	(null)	1112
2	2	10 Cina italieneasca	8	7	10	2	(null)	1221
3	3	1 Franz aromat	3	1	9	4	(null)	956
4	4	9 Gustarele pentru ceai	6	10	2	4	(null)	929
5	5	8 Gustare rapida	6	4	(null)	(null)	(null)	536
6	6	7 Cina usoara	1	3	(null)	(null)	(null)	346
7	7	4 Franz	9	10	(null)	(null)	(null)	803
8	8	4 Pranz vegan	3	5	(null)	(null)	(null)	953
9	9	3 Gustare dulce	10	4	(null)	(null)	(null)	405
10	10	3 Petrecere	8	7	10	2	5	1918

10. Tabel APRECIAZA

```

CREATE TABLE apreciaza (
    id_utilizator NUMBER(10),
    id_postare NUMBER(20),
    nr_stele NUMBER(1),
    CONSTRAINT apr_pk PRIMARY KEY (id_utilizator, id_postare, nr_stele),
    CONSTRAINT apr_ut_fk FOREIGN KEY (id_utilizator) REFERENCES utilizatori(id_utilizator),
    CONSTRAINT apr_post_fk FOREIGN KEY (id_postare) REFERENCES postari(id_postare),

```

```
CONSTRAINT apr_st_fk FOREIGN KEY (nr_stele) REFERENCES scor(nr_stele)
);

INSERT INTO apreciaza
VALUES (1, 5, 5);
INSERT INTO apreciaza
VALUES (8, 4, 3);
INSERT INTO apreciaza
VALUES (5, 4, 5);
INSERT INTO apreciaza
VALUES (2, 1, 1);
INSERT INTO apreciaza
VALUES (9, 7, 4);
INSERT INTO apreciaza
VALUES (2, 8, 5);
INSERT INTO apreciaza
VALUES (9, 3, 5);
INSERT INTO apreciaza
VALUES (4, 10, 4);
INSERT INTO apreciaza
VALUES (1, 9, 5);
INSERT INTO apreciaza
VALUES (3, 7, 4);
```

The screenshot shows the Oracle SQL Developer interface. The top window is a 'Worksheet' titled 'Query Builder' containing the following SQL script:

```

117 CREATE TABLE apreciaza (
118     id_utilizator NUMBER(10),
119     id_postare NUMBER(20),
120     nr_stele NUMBER(1),
121     CONSTRAINT apr_pk PRIMARY KEY (id_utilizator, id_postare, nr_stele),
122     CONSTRAINT apr_ut_fk FOREIGN KEY (id_utilizator) REFERENCES utilizatori(id_utilizator),
123     CONSTRAINT apr_post_fk FOREIGN KEY (id_postare) REFERENCES postari(id_postare),
124     CONSTRAINT apr_st_fk FOREIGN KEY (nr_stele) REFERENCES scor(nr_stele)
125 );
126 INSERT INTO apreciaza
127 VALUES (1, 5, 5);
128 INSERT INTO apreciaza
129 VALUES (8, 4, 3);
130 INSERT INTO apreciaza
131 VALUES (5, 4, 5);
132 INSERT INTO apreciaza
133 VALUES (2, 1, 1);
134 INSERT INTO apreciaza
135 VALUES (9, 7, 4);
136 INSERT INTO apreciaza
137 VALUES (2, 8, 5);
138 INSERT INTO apreciaza
139 VALUES (9, 3, 5);
140 INSERT INTO apreciaza
141 VALUES (4, 10, 4);
142 SELECT * FROM apreciaza;

```

The bottom window is also a 'Worksheet' titled 'Query Builder' containing the following SQL script:

```

VALUES (2, 1, 1);
INSERT INTO apreciaza
VALUES (9, 7, 4);
INSERT INTO apreciaza
VALUES (2, 8, 5);
INSERT INTO apreciaza
VALUES (9, 3, 5);
INSERT INTO apreciaza
VALUES (4, 10, 4);
INSERT INTO apreciaza
VALUES (1, 9, 5);
INSERT INTO apreciaza
VALUES (3, 7, 4);
SELECT * FROM apreciaza;

```

Below these windows is a 'Script Output' tab showing the results of the last query:

All Rows Fetched: 10 in 0,003 seconds

ID_UTILIZATOR	ID_POSTARE	NR_STELE
1	1	5
2	1	9
3	2	1
4	2	8
5	3	7
6	4	10
7	5	4
8	8	4
9	9	3
10	9	7

11. Tabel LISTA_DORINTE

```

CREATE TABLE lista_dorinte (
    id_utilizator NUMBER(10),
    id_postare NUMBER(20),
    CONSTRAINT ld_pk PRIMARY KEY (id_utilizator, id_postare),
    CONSTRAINT ld_ut_fk FOREIGN KEY (id_utilizator) REFERENCES utilizatori(id_utilizator),
    CONSTRAINT ld_post_fk FOREIGN KEY (id_postare) REFERENCES postari(id_postare)
);

```

```
INSERT INTO lista_dorinte
```

```

VALUES (1,5);
INSERT INTO lista_dorinte
VALUES (2,8);
INSERT INTO lista_dorinte
VALUES (5,4);
INSERT INTO lista_dorinte
VALUES (9,3);
INSERT INTO lista_dorinte
VALUES (3,8);
INSERT INTO lista_dorinte
VALUES (7,5);
INSERT INTO lista_dorinte
VALUES (10,5);
INSERT INTO lista_dorinte
VALUES (3,4);
INSERT INTO lista_dorinte
VALUES (7,4);
INSERT INTO lista_dorinte
VALUES (4,1);

```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor containing the SQL script for creating the `LISTA_DORINTE` table and inserting data into it. The table has two columns: `ID_UTILIZATOR` and `ID_POSTARE`. The data inserted consists of 10 rows, each with a unique ID_UTILIZATOR value and a corresponding ID_POSTARE value.

ID_UTILIZATOR	ID_POSTARE
1	5
2	8
3	4
4	8
5	1
6	4
7	4
8	5
9	3
10	5

Below the code editor, there are tabs for "Script Output" and "Query Result". The "Query Result" tab is selected, showing the message "All Rows Fetched: 10 in 0,002 seconds".

12. Tabel INSIGNE

```
CREATE TABLE insigne (
```

```
id_insigna NUMBER(5),
denumire VARCHAR2(50),
descriere VARCHAR2(200),
CONSTRAINT ins_pk PRIMARY KEY (id_insigna)
);
```

```
INSERT INTO insigne
VALUES (sequence_PK.nextval, 'Bine ai venit!', 'Felicitari! Ai primit prima ta insigna pentru crearea contului.');
```

```
INSERT INTO insigne
VALUES (sequence_PK.nextval, 'Sa facem cunostinta!', 'Felicitari! Ai primit insigna "Sa facem cunostinta!" pentru primul tau followed!');
```

```
INSERT INTO insigne
VALUES (sequence_PK.nextval, 'Pofta buna!', 'Felicitari! Ai primit insigna "Pofta buna!" pentru crearea primului tau meniu.');
```

```
INSERT INTO insigne
VALUES (sequence_PK.nextval, 'Unde-s multi, puterea creste!', 'Felicitari! Ai primit insigna "Unde-s multi, puterea creste!" pentru intrarea in primul tau grup.');
```

```
INSERT INTO insigne
VALUES (sequence_PK.nextval, 'Ador!', 'Felicitari! Ai primit insigna "Ador!" pentru adaugarea primei postari in lista de dorinte.');
```

```
INSERT INTO insigne
VALUES (sequence_PK.nextval, 'Inspiratie!', 'Felicitari! Ai primit insigna "Inspiratie!" pentru crearea primei tale postari.');
```

```
INSERT INTO insigne
VALUES (sequence_PK.nextval, 'Sa impartasim!', 'Felicitari! Ai primit insigna "Sa impartasim!" pentru crearea a 3 postari.');
```

```
INSERT INTO insigne
VALUES (sequence_PK.nextval, 'Experimentat!', 'Felicitari! Ai primit insigna "Experimentat" pentru crearea a 10 postari.');
```

```
INSERT INTO insigne
VALUES (sequence_PK.nextval, 'Carte de bucate!', 'Felicitari! Ai primit insigna "Carte de bucate!" pentru crearea a 100 de postari.');
```

```
INSERT INTO insigne
VALUES (sequence_PK.nextval, 'Leader in devenire!', 'Felicitari! Ai primit insigna "Leader in devenire!" pentru crearea primului tau grup.');
```

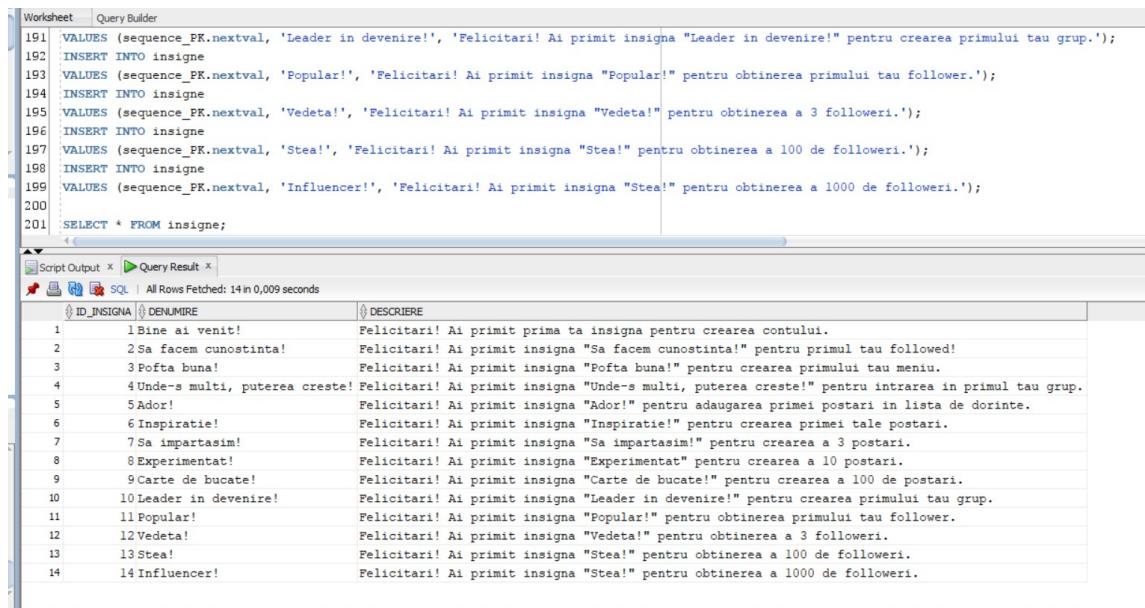
```
INSERT INTO insigne
```

VALUES (sequence_PK.nextval, 'Popular!', 'Felicitari! Ai primit insigna "Popular!" pentru obtinerea primului tau follower.');

INSERT INTO insigne
 VALUES (sequence_PK.nextval, 'Vedeta!', 'Felicitari! Ai primit insigna "Vedeta!" pentru obtinerea a 3 followeri.');

INSERT INTO insigne
 VALUES (sequence_PK.nextval, 'Stea!', 'Felicitari! Ai primit insigna "Stea!" pentru obtinerea a 100 de followeri.');

INSERT INTO insigne
 VALUES (sequence_PK.nextval, 'Influencer!', 'Felicitari! Ai primit insigna "Stea!" pentru obtinerea a 1000 de followeri.');



The screenshot shows the Oracle SQL Developer interface. The top pane is a 'Worksheet' containing a script with numbered lines from 191 to 201. Lines 191-198 show the insertion of four new insignes into the 'insigne' table. Line 199 shows the insertion of the 'Influencer!' insigna. Line 200 is a placeholder. Line 201 is a 'SELECT * FROM insigne;' query. The bottom pane is a 'Query Result' window showing the contents of the 'insigne' table. The table has two columns: 'ID_INSIGNA' and 'DESCRIERE'. The data is as follows:

ID_INSIGNA	DESCRIERE
1	1Bine ai venit!
2	2Sa facem cunostinta!
3	3Pofta buna!
4	4Unde-s multi, puterea create!
5	5Ador!
6	6Inspiratie!
7	7Sa impartasim!
8	8Experimentat!
9	9Carte de bucate!
10	10Leader in devenire!
11	11Popular!
12	12Vedeta!
13	13Stea!
14	14Influencer!
	Felicitari! Ai primit prima ta insigna pentru crearea contului.
	Felicitari! Ai primit insigna "Sa facem cunostinta!" pentru primul tau follower.
	Felicitari! Ai primit insigna "Pofta buna!" pentru crearea primului tau meniu.
	Felicitari! Ai primit insigna "Unde-s multi, puterea create!" pentru intrarea in primul tau grup.
	Felicitari! Ai primit insigna "Ador!" pentru adaugarea primei postari in lista de dorinte.
	Felicitari! Ai primit insigna "Inspiratie!" pentru crearea primei tale postari.
	Felicitari! Ai primit insigna "Sa impartasim!" pentru crearea a 3 postari.
	Felicitari! Ai primit insigna "Experimentat" pentru crearea a 10 postari.
	Felicitari! Ai primit insigna "Carte de bucate!" pentru crearea a 100 de postari.
	Felicitari! Ai primit insigna "Leader in devenire!" pentru crearea primului tau grup.
	Felicitari! Ai primit insigna "Popular!" pentru obtinerea primului tau follower.
	Felicitari! Ai primit insigna "Vedeta!" pentru obtinerea a 3 followeri.
	Felicitari! Ai primit insigna "Stea!" pentru obtinerea a 100 de followeri.
	Felicitari! Ai primit insigna "Stea!" pentru obtinerea a 1000 de followeri.

13. Tabel GRUPURI

```
CREATE TABLE grupuri (
    id_grup NUMBER(5),
    administrator NUMBER(10),
    denumire VARCHAR2(50),
    descriere VARCHAR2(500),
    data_infiintare DATE,
    CONSTRAINT gr_pk PRIMARY KEY (id_grup),
    CONSTRAINT gr_admin_fk FOREIGN KEY (administrator) REFERENCES utilizatori(id_utilizator)
);
```

INSERT INTO grupuri

VALUES (sequence_PK.nextval, 2, 'Preparate vegane', 'Vegan nu inseamna sa mananci mereu acelasi meniu. Preparatele pot fi delicioase si foarte colorate. Sa gatim creativ!', '03-JUN-2020');

INSERT INTO grupuri

VALUES (sequence_PK.nextval, 7, 'Savuros si sanatos', 'Mancaruri sanatoase si foarte gustoase. A manca sanatos nu inseamna mai putin savuros.', '10-FEB-2020');

INSERT INTO grupuri

VALUES (sequence_PK.nextval, 8, 'Arome italienesti', 'Gustul mediteranean in preparate sanatoase. ', '15-MAR-2020');

INSERT INTO grupuri

VALUES (sequence_PK.nextval, 9, 'Dulciuri sanatoase', 'A manca sanatos nu inseamna sa renuntam la dulciuri. Haideti sa ne bucuram de momente dulci in continuare!', '04-MAY-2022');

INSERT INTO grupuri

VALUES (sequence_PK.nextval, 10, 'Preparate de vara', 'Sa ne bucuram impreuna de prospetimea si diversitatea legumelor si fructelor specifice verii, impartasind si gatind preparate delicioase', '07-JUN-2020');

```

91 -- tabel GRUPURI --
92 CREATE TABLE grupuri (
93     id_grup NUMBER(5),
94     administrator NUMBER(10),
95     denumire VARCHAR2(50),
96     descriere VARCHAR2(500),
97     data_infiintare DATE,
98     CONSTRAINT gr_pk PRIMARY KEY (id_grup),
99     CONSTRAINT gr_admin_fk FOREIGN KEY (administrator) REFERENCES utilizatori(id_utilizator)
100 );
101 INSERT INTO grupuri
102 VALUES (sequence_PK.nextval, 2, 'Preparate vegane', 'Vegan nu inseamna sa mananci mereu acelasi meniu. Preparatele pot fi delicioase si foarte colorate. Sa gatim creativ!', '03-JUN-2020');
103 INSERT INTO grupuri
104 VALUES (sequence_PK.nextval, 7, 'Savuros si sanatos', 'Mancaruri sanatoase si foarte gustoase. A manca sanatos nu inseamna mai putin savuros.', '10-FEB-2020');
105 INSERT INTO grupuri
106 VALUES (sequence_PK.nextval, 8, 'Arome italienesti', 'Gustul mediteranean in preparate sanatoase. ', '15-MAR-2020');
107 INSERT INTO grupuri
108 VALUES (sequence_PK.nextval, 9, 'Dulciuri sanatoase', 'A manca sanatos nu inseamna sa renuntam la dulciuri. Haideti sa ne bucuram de momente dulci in continuare!', '04-MAY-2022');
109 INSERT INTO grupuri
110 VALUES (sequence_PK.nextval, 10, 'Preparate de vara', 'Sa ne bucuram impreuna de prospetimea si diversitatea legumelor si fructelor specifice verii, impartasind si gatind preparate delicioase', '07-JUN-2020');

```

Script Output | Query Result | Query Result 1 | All Rows Fetched: 5 in 0,003 seconds

ID_GRUP	ADMINISTRATOR	DENUMIRE	DESCRIERE
1	1	2 Preparate vegane	Vegan nu inseamna sa mananci mereu acelasi meniu. Preparatele pot fi delicioase si foarte colorate. Sa gatim creativ!
2	2	7 Savuros si sanatos	Mancaruri sanatoase si foarte gustoase. A manca sanatos nu inseamna mai putin savuros.
3	3	8 Arome italienesti	Gustul mediteranean in preparate sanatoase.
4	4	9 Dulciuri sanatoase	A manca sanatos nu inseamna sa renuntam la dulciuri. Haideti sa ne bucuram de momente dulci in continuare!
5	5	10 Preparate de vara	Sa ne bucuram impreuna de prospetimea si diversitatea legumelor si fructelor specifice verii, impartasind si gatind preparate delicioase

14. Tabel GRUP_UTILIZATOR

```

CREATE TABLE GRUP_UTILIZATOR (
    id_grup NUMBER(10),
    id_utilizator NUMBER(10),
    CONSTRAINT p_u_pk PRIMARY KEY (id_grup, id_utilizator)
);

```

INSERT INTO grup_utilizator

```

VALUES ( 1, 2 );
INSERT INTO grup_utilizator
VALUES ( 1, 4 );
INSERT INTO grup_utilizator
VALUES ( 2, 9 );
INSERT INTO grup_utilizator
VALUES ( 2, 7 );
INSERT INTO grup_utilizator
VALUES ( 3, 2 );
INSERT INTO grup_utilizator
VALUES ( 3, 5 );
INSERT INTO grup_utilizator
VALUES ( 4, 1 );
INSERT INTO grup_utilizator
VALUES ( 5, 6 );
INSERT INTO grup_utilizator
VALUES ( 5, 3 );
INSERT INTO grup_utilizator
VALUES ( 5, 4 );
INSERT INTO grup_utilizator
VALUES ( 5, 10 );

```

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Worksheet' tab is active, displaying the following SQL code:

```

-- tabel GRUP_UTILIZATOR --
CREATE TABLE GRUP_UTILIZATOR (
    id_grup NUMBER(10),
    id_utilizator NUMBER(10),
    CONSTRAINT p_u_pk PRIMARY KEY (id_grup, id_utilizator)
);
INSERT INTO grup_utilizator
VALUES ( 1, 2 );
INSERT INTO grup_utilizator
VALUES ( 1, 4 );
INSERT INTO grup_utilizator
VALUES ( 2, 9 );
INSERT INTO grup_utilizator
VALUES ( 2, 7 );
INSERT INTO grup_utilizator
VALUES ( 3, 2 );
INSERT INTO grup_utilizator
VALUES ( 3, 5 );
INSERT INTO grup_utilizator
VALUES ( 4, 1 );
INSERT INTO grup_utilizator
VALUES ( 5, 6 );
INSERT INTO grup_utilizator
VALUES ( 5, 3 );
INSERT INTO grup_utilizator
VALUES ( 5, 4 );
INSERT INTO grup_utilizator
VALUES ( 5, 10 );

```

On the right, the 'Query Result' tab is active, showing the output of the SELECT statement:

```

SELECT * FROM GRUP_UTILIZATOR;

```

ID_GRUP	ID_UTILIZATOR
1	1
2	1
3	2
4	2
5	3
6	3
7	4
8	5
9	5
10	5
11	5

15. Tabel UTILIZATOR_INSIGNA

```

CREATE TABLE utilizator_insigna (
    id_utilizator NUMBER(10),

```

```

id_insigna NUMBER(5),
data DATE,
CONSTRAINT u_i_pk PRIMARY KEY (id_utilizator, id_insigna),
CONSTRAINT u_i_ut_fk FOREIGN KEY (id_utilizator) REFERENCES utilizatori(id_utilizator),
CONSTRAINT u_i_in_fk FOREIGN KEY (id_insigna) REFERENCES insigne(id_insigna)
);

INSERT INTO utilizator_insigna
VALUES ( 1, 1, (SELECT data_creare_cont FROM utilizatori WHERE id_utilizator = 1));
INSERT INTO utilizator_insigna
VALUES ( 2, 1, (SELECT data_creare_cont FROM utilizatori WHERE id_utilizator = 2));
INSERT INTO utilizator_insigna
VALUES ( 3, 1, (SELECT data_creare_cont FROM utilizatori WHERE id_utilizator = 3));
INSERT INTO utilizator_insigna
VALUES ( 4, 1, (SELECT data_creare_cont FROM utilizatori WHERE id_utilizator = 4));
INSERT INTO utilizator_insigna
VALUES ( 5, 1, (SELECT data_creare_cont FROM utilizatori WHERE id_utilizator = 5));
INSERT INTO utilizator_insigna
VALUES ( 6, 1, (SELECT data_creare_cont FROM utilizatori WHERE id_utilizator = 6));
INSERT INTO utilizator_insigna
VALUES ( 7, 1, (SELECT data_creare_cont FROM utilizatori WHERE id_utilizator = 7));
INSERT INTO utilizator_insigna
VALUES ( 8, 1, (SELECT data_creare_cont FROM utilizatori WHERE id_utilizator = 8));
INSERT INTO utilizator_insigna
VALUES ( 9, 1, (SELECT data_creare_cont FROM utilizatori WHERE id_utilizator = 9));
INSERT INTO utilizator_insigna
VALUES ( 10, 1, (SELECT data_creare_cont FROM utilizatori WHERE id_utilizator = 10));
INSERT INTO utilizator_insigna
VALUES ( 1, 3, '23-MAY-2020');
INSERT INTO utilizator_insigna
VALUES ( 3, 3, '20-SEP-2019');
INSERT INTO utilizator_insigna
VALUES ( 4, 3, '20-MAY-2023');
INSERT INTO utilizator_insigna
VALUES ( 7, 3, '10-FEB-2019');
INSERT INTO utilizator_insigna
VALUES ( 8, 3, '13-OCT-2018');
INSERT INTO utilizator_insigna
VALUES ( 9, 3, '22-APR-2021');
INSERT INTO utilizator_insigna
VALUES ( 10, 3, '3-APR-2021');
INSERT INTO utilizator_insigna
VALUES ( 1, 5, '08-JUN-2019');
INSERT INTO utilizator_insigna
VALUES ( 2, 5, '22-MAR-2020');
INSERT INTO utilizator_insigna
VALUES ( 3, 5, '15-JUL-2019');
INSERT INTO utilizator_insigna

```

```
VALUES ( 4, 5, '08-JUN-2022');
INSERT INTO utilizator_insigna
VALUES ( 5, 5, '20-DEC-2020');
INSERT INTO utilizator_insigna
VALUES ( 7, 5, '10-MAR-2019');
INSERT INTO utilizator_insigna
VALUES ( 9, 5, '23-APR-2021');
INSERT INTO utilizator_insigna
VALUES ( 10, 5, '20-FEB-2022');
INSERT INTO utilizator_insigna
VALUES ( 1, 6, '22-FEB-2020');
INSERT INTO utilizator_insigna
VALUES ( 2, 6, '10-DEC-2021');
INSERT INTO utilizator_insigna
VALUES ( 5, 6, '24-JUL-2022');
INSERT INTO utilizator_insigna
VALUES ( 7, 6, '19-DEC-2022');
INSERT INTO utilizator_insigna
VALUES ( 8, 6, '11-JUN-2019');
INSERT INTO utilizator_insigna
VALUES ( 9, 6, '22-MAY-2021');
INSERT INTO utilizator_insigna
VALUES ( 10, 6, '23-APR-2021');
INSERT INTO utilizator_insigna
VALUES ( 1, 7, '22-JUN-2022');
INSERT INTO utilizator_insigna
VALUES ( 1, 4, '10-MAY-2022');
INSERT INTO utilizator_insigna
VALUES ( 2, 4, '2-JUN-2020');
INSERT INTO utilizator_insigna
VALUES ( 3, 4, '10-DEC-2021');
INSERT INTO utilizator_insigna
VALUES ( 4, 4, '17-SEP-2020');
INSERT INTO utilizator_insigna
VALUES ( 5, 4, '09-JAN-2021');
INSERT INTO utilizator_insigna
VALUES ( 6, 4, '07-OCT-2020');
INSERT INTO utilizator_insigna
VALUES ( 7, 4, '11-APR-2020');
INSERT INTO utilizator_insigna
VALUES ( 9, 4, '13-FEB-2020');
INSERT INTO utilizator_insigna
VALUES ( 10, 4, '24-NOV-2023');
INSERT INTO utilizator_insigna
VALUES ( 2, 2, '22-JAN-2023');
INSERT INTO utilizator_insigna
VALUES ( 8, 4, '18-JUL-2022');
INSERT INTO utilizator_insigna
```

```

VALUES ( 2, 10, '03-JUN-2020');
INSERT INTO utilizator_insigna
VALUES ( 7, 10, '10-FEB-2020');
INSERT INTO utilizator_insigna
VALUES ( 8, 10, '15-MAR-2020');
INSERT INTO utilizator_insigna
VALUES ( 9, 10, '04-MAY-2022');
INSERT INTO utilizator_insigna
VALUES ( 10, 10, '07-JUN-2020');
INSERT INTO utilizator_insigna
VALUES ( 1, 11, '03-APR-2022');
INSERT INTO utilizator_insigna
VALUES ( 2, 11, '10-MAY-2022');
INSERT INTO utilizator_insigna
VALUES ( 7, 11, '13-JUN-2022');
INSERT INTO utilizator_insigna
VALUES ( 10, 11, '17-AUG-2023');
INSERT INTO utilizator_insigna
VALUES ( 7, 12, '05-SEP-2023);

```

The screenshot shows two windows side-by-side. On the left is the 'Query Result' window displaying the output of a SELECT query. On the right is the 'Table Editor' window showing the data in the UTILIZATOR_INSIGNA table.

Query Result Window:

ID_UTILIZATOR	ID_INSIGNA	DATA
1	1	1 07-JUN-19
2	2	1 20-MAR-20
3	3	1 13-JUL-19
4	4	1 13-MAY-23
5	5	1 15-DEC-20
6	6	1 12-NOV-19
7	7	1 09-FEB-19
8	8	1 10-OCT-18
9	9	1 18-APR-21
10	10	1 02-JAN-21
11	1	3 23-MAY-20
12	3	3 20-SEP-19
13	4	3 20-MAY-23
14	7	3 10-FEB-19
15	8	3 13-OCT-18
16	9	3 22-APR-21
17	10	3 03-APR-21
18	1	5 08-JUN-19
19	2	5 22-MAR-20
20	3	5 15-JUL-19
21	4	5 08-JUN-22
22	5	5 20-DEC-20
23	7	5 10-MAR-19
24	9	5 23-APR-21
25	10	5 20-MAR-20

Table Editor Window:

ID_UTILIZATOR	ID_INSIGNA	DATA
29	7	6 19-DEC-22
30	8	6 11-JUN-19
31	9	6 22-MAY-21
32	10	6 23-APR-21
33	1	7 22-JUN-22
34	1	4 10-MAY-22
35	2	4 02-JUN-20
36	3	4 10-DEC-21
37	4	4 17-SEP-20
38	5	4 09-JAN-21
39	6	4 07-OCT-20
40	7	4 11-APR-20
41	9	4 13-FEB-20
42	10	4 24-NOV-23
43	2	2 22-JAN-23
44	8	4 18-JUL-22
45	2	10 03-JUN-20
46	7	10 10-FEB-20
47	8	10 15-MAR-20
48	9	10 04-MAY-22
49	1	11 03-APR-22
50	2	11 10-MAY-22
51	7	11 13-JUN-22
52	10	11 17-AUG-23
53	7	12 05-SEP-23
54	10	10 07-JUN-20

16. Tabel POSTARE_ALERGEN

```
CREATE TABLE postare_alergen (
    id_postare NUMBER(20),
    id_alergen NUMBER(3),
    CONSTRAINT p_a_pk PRIMARY KEY (id_alergen, id_postare),
    CONSTRAINT p_a_a_fk FOREIGN KEY (id_alergen) REFERENCES alergen(id_alergen),
    CONSTRAINT p_a_p_fk FOREIGN KEY (id_postare) REFERENCES postari(id_postare)
);

INSERT INTO postare_alergen
VALUES (1,9);
INSERT INTO postare_alergen
VALUES (2,5);
INSERT INTO postare_alergen
VALUES (2,13);
INSERT INTO postare_alergen
VALUES (3,8);
INSERT INTO postare_alergen
VALUES (4,8);
INSERT INTO postare_alergen
VALUES (5,1);
INSERT INTO postare_alergen
VALUES (5,2);
INSERT INTO postare_alergen
VALUES (5,11);
INSERT INTO postare_alergen
VALUES (6,2);
INSERT INTO postare_alergen
VALUES (6,5);
INSERT INTO postare_alergen
VALUES (6,4);
INSERT INTO postare_alergen
VALUES (7,1);
INSERT INTO postare_alergen
VALUES (7,7);
INSERT INTO postare_alergen
VALUES (8,3);
INSERT INTO postare_alergen
VALUES (8,12);
INSERT INTO postare_alergen
VALUES (8,4);
INSERT INTO postare_alergen
VALUES (9,12);
INSERT INTO postare_alergen
VALUES (9,1);
INSERT INTO postare_alergen
VALUES (9,2);
INSERT INTO postare_alergen
VALUES (10,2);
```

```

INSERT INTO postare_alergen
VALUES (10,11);
INSERT INTO postare_alergen
VALUES (10,3);
INSERT INTO postare_alergen
VALUES (10,10);
INSERT INTO postare_alergen
VALUES (10,4);

```

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Script Output' tab displays the SQL code for creating the table and inserting data. On the right, the 'Query Result' tab shows the resulting data in a table format.

ID_POSTARE	ID_ALERGEN
1	5
2	7
3	9
4	5
5	6
6	9
7	10
8	8
9	10
10	6
11	8
12	10
13	2
14	6
15	7
16	3
17	4
18	1
19	10
20	5
21	10
22	8
23	9
24	2

17. Tabel UTILIZATOR_ALERGEN

```

CREATE TABLE utilizator_alergen (
    id_utilizator NUMBER(20),
    id_alergen NUMBER(3),
    CONSTRAINT u_a_pk PRIMARY KEY (id_alergen, id_utilizator),
    CONSTRAINT u_a_a_fk FOREIGN KEY (id_alergen) REFERENCES alergen(id_alergen),
    CONSTRAINT u_a_p_fk FOREIGN KEY (id_utilizator) REFERENCES utilizatori(id_utilizator)
);

INSERT INTO utilizator_alergen
VALUES (1, 1);
INSERT INTO utilizator_alergen
VALUES (1, 2);
INSERT INTO utilizator_alergen
VALUES (2, 2);

```

```

INSERT INTO utilizator_alergen
VALUES (2, 8);
INSERT INTO utilizator_alergen
VALUES (4, 1);
INSERT INTO utilizator_alergen
VALUES (4, 6);
INSERT INTO utilizator_alergen
VALUES (4, 14);
INSERT INTO utilizator_alergen
VALUES (8, 1);
INSERT INTO utilizator_alergen
VALUES (9, 1);
INSERT INTO utilizator_alergen
VALUES (9, 3);

```

The screenshot shows two panes in Oracle SQL Developer. The left pane displays the SQL code for creating the UTILIZATOR_ALERGEN table and inserting data. The right pane shows the resulting data in a grid and the SQL query used to retrieve it.

```

281 -- tabel UTILIZATOR_ALERGEN --
282 CREATE TABLE utilizator_alergen (
283     id_utilizator NUMBER(20),
284     id_alergen NUMBER(3),
285     CONSTRAINT u_a_pk PRIMARY KEY (id_alergen, id_utilizator),
286     CONSTRAINT u_a_a_fk FOREIGN KEY (id_alergen) REFERENCES alergen(id_alergen),
287     CONSTRAINT u_a_p_fk FOREIGN KEY (id_utilizator) REFERENCES utilizatori(id_utilizator)
288 );
289 SELECT * FROM UTILIZATORI JOIN POSTARI USING(ID_UTILIZATOR);
290 INSERT INTO utilizator_alergen
291 VALUES (1, 1);
292 INSERT INTO utilizator_alergen
293 VALUES (1, 2);
294 INSERT INTO utilizator_alergen
295 VALUES (2, 2);
296 INSERT INTO utilizator_alergen
297 VALUES (2, 8);
298 INSERT INTO utilizator_alergen
299 VALUES (4, 1);
300 INSERT INTO utilizator_alergen
301 VALUES (4, 6);
302 INSERT INTO utilizator_alergen
303 VALUES (4, 14);
304 INSERT INTO utilizator_alergen
305 VALUES (8, 1);
306 INSERT INTO utilizator_alergen
307 VALUES (9, 1);
308 INSERT INTO utilizator_alergen
309 VALUES (9, 3);
310
311 SELECT * FROM utilizator_alergen;
312
313

```

ID_UTILIZATOR	ID_ALERGEN
1	1
2	4
3	8
4	9
5	1
6	2
7	9
8	4
9	2
10	4
11	14

12. Cereri SQL

- a) Să se afișeze denumirea grupului, și numele complet al administratorului pentru grupul cel mai activ (media numărului de postări per utilizator din grup este cea mai mare). Se iau în calcul doar grupurile care au cel puțin doi utilizatori înscrise.

Cererea utilizează:

- bloc WITH

- grupări de date cu subcereri nesincronizate în care intervin cel puțin 3 tabele, funcții grup, filtrare la nivel de grupuri (în cadrul aceleiași cereri)

-- Pentru fiecare grup, pentru fiecare utilizator calculăm numărul de postări create în ultimii doi ani

```

WITH nr_post AS(
SELECT gu.id_grup, gu.id_utilizator, COUNT(p.id_postare) nr_postari
FROM grup_utilizator gu JOIN postari p ON (gu.id_utilizator = p.id_utilizator)
WHERE TO_CHAR(data, 'YYYY') >= TO_CHAR(SYSDATE, 'YYYY')-2
GROUP BY gu.id_utilizator, gu.id_grup
)

SELECT g.denumire, u.nume || ' ' || u.prenume "Nume administrator"
FROM grupuri g JOIN (
    -- pentru fiecare grup care are cel putin doi utilizatori inscrisi, calculam media de postari
    SELECT id_grup, AVG(nr_postari)
    FROM nr_post
    GROUP BY id_grup
    HAVING COUNT(id_utilizator)>=2
    ORDER BY AVG(nr_postari) DESC
) ap ON (g.id_grup = ap.id_grup)

JOIN utilizatori u ON (u.id_utilizator = g.administrator)
-- selectam doar primul grup din ordinea descrescatoare a grupurilor in functie de activitate
WHERE ROWNUM < 2;

```

```

Worksheet | Query Builder
72 -- sa se afiseze denumirea grupului, si numele complet al administratorului pentru grupul cel mai activ ( media numarului de postari per utilizator din grup)
73 -- Se iau in calcul doar grupurile care au cel putin doi utilizatori inscrisi.
74 -- Crearea utilizeaza:
75 -- - bloc WITH
76 -- - grupuri de date cu subcereri nesincronizate in care intervin cel putin 3 table, functii grup, filtrare la nivel de grupuri (in cadrul aceleiasi cerere)
77 |
78 -- Pentru fiecare grup, pentru fiecare utilizator calculam numarul de postari create in ultimii doi ani
79 WITH nr_post AS(
80   SELECT gu.id_grup, gu.id_utilizator, COUNT(p.id_postare) nr_postari
81   FROM grup_utilizator gu JOIN postari p ON (gu.id_utilizator = p.id_utilizator)
82   WHERE TO_CHAR(data, 'YYYY') >= TO_CHAR(SYSDATE, 'YYYY')-2
83   GROUP BY gu.id_utilizator, gu.id_grup
84 )
85
86   SELECT g.denumire, u.nume || ' ' || u.prenume "Nume administrator"
87   FROM grupuri g JOIN (
88     -- pentru fiecare grup care are cel putin doi utilizatori inscrisi, calculam media de postari
89     SELECT id_grup, AVG(nr_postari)
90     FROM nr_post
91     GROUP BY id_grup
92     HAVING COUNT(id_utilizator)>=2
93     ORDER BY AVG(nr_postari) DESC
94   )ap ON (g.id_grup = ap.id_grup)

```

Script Output | Query Result | Query Result 1 | All Rows Fetched: 1 in 0,01 seconds

DENUMIRE	Nume administrator
1 Savuros si sanatos	Sebastien Denis

b) Să se afișeze titlul postării care este cea mai populară, adică cea care apare în lista de dorințe a celor mai mulți utilizatori.

Cererea utilizează:

- subcerere nesincronizată în clauza FROM

SELECT titlu

FROM (

SELECT COUNT(l.id_utilizator) nr, l.id_postare, p.titlu

FROM lista_dorinte l

JOIN postari p ON (p.id_postare = l.id_postare)

GROUP BY l.id_postare, p.titlu

ORDER BY nr DESC

)

WHERE ROWNUM < 2;

```

1 -- Sa se afiseze titlul postarii care este cea mai populara, adica cea care apare in lista de dorinte a celor mai multi utilizatori
2 -- Cererea utilizeaza:
3 -- - subcerere nesincronizata in clauza FROM
4
5 SELECT titlu
6   FROM (
7     SELECT COUNT(l.id_utilizator).nr, l.id_postare, p.titlu
8       FROM lista_dorinte l
9      JOIN postari p ON (p.id_postare = l.id_postare)
10     GROUP BY l.id_postare, p.titlu
11    ORDER BY nr DESC
12  )
13 WHERE ROWNUM < 2;

```

Script Output | Query Result | Query Result 1 |

SQL | All Rows Fetched: 1 in 0,003 seconds

TITLU
1 Tort vegan

c) Încercăm să observăm comportamentul utilizatorilor în aplicație, mai exact perioada de timp de la crearea contului pana la crearea unui grup (utilizatorul poate crea un singur grup). Astfel un utilizator s-a grăbit (statut = prea repede) daca încă nu are nicio luna încheiată în aplicație, este PERFECT daca are un an împlinit, altfel este OK (daca este între 0 și 12 luni sau peste și dacă încă nu și-a creat grup încă).

Se cere să se afișeze id-ul de utilizator, username-ul si statutul pentru fiecare utilizator. De asemenea, să se afișeze denumirea grupului, dacă descrierea acestuia conține 'sanatos/sanatoasa' sau ceva asemănător, '-' altfel. Rezultatele să se ordoneze descrescător după lungimea username-ului.

Cererea utilizează:

- ordonări si utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)
- 2 funcții pe șiruri de caractere (INSTR, UPPER), 2 funcții pe date calendaristice (TRUNC, MONTHS_BETWEEN)

```

1 SELECT id_utilizator, username, DECODE(TRUNC(MONTHS_BETWEEN(NVL(data_infiintare,data_creare_cont), data_creare_cont)),0,'prea repede', 12, 'perfect', 'OK')
2   FROM utilizatori
3  LEFT JOIN grupuri ON (administrator = id_utilizator)
4 ORDER BY LENGTH(username) DESC;

```

Script Output | Query Result |

SQL | All Rows Fetched: 10 in 0,018 seconds

ID_UTILIZATOR	USERNAME	STATUT	DENUMIRE
1	6faustinoStan	prea repede -	
2	9CharleneH89	perfect	Dulciuri sanatoase
3	1CarolinaS72	prea repede -	
4	10veronicaS92	OK	Preparate de vară
5	3adamkunto	prea repede -	
6	4charles94	prea repede -	
7	7Sdenis96	perfect	Savuros și sanatos
8	2jacobM85	OK	Preparate vegane
9	5adrieneT	prea repede -	
10	8enzoP87	OK	Arme italienești

```

SELECT id_utilizator, username,
DECODE(TRUNC(MONTHS_BETWEEN(NVL(data_infiintare,data_creare_cont), data_creare_cont)),0,'prea
repede', 12, 'perfect', 'OK') "STATUT", NVL2(INSTR(UPPER(descriere),'SANATO'),denumire, ' - ') denumire
FROM utilizatori

```

```
LEFT JOIN grupuri ON (administrator = id_utilizator)
```

```
ORDER BY LENGTH(username) DESC;
```

- d) Pentru fiecare utilizator, să se afișeze cate comentarii a făcut la postările create de cei pe care îi urmărește și denumirea grupului în care se află cei mai mulți din următorii lui.

Cererea utilizează:

- subcereri sincronizate în care intervin cel puțin 3 tabele

The screenshot shows the Oracle SQL Developer interface. In the top window (Worksheet), there is a complex SQL query. Below it, in the bottom window (Query Result), the results of the query are displayed in a table.

ID_UTILIZATOR	NR_COMENTARIU	GRUP
1	1	0 (null)
2	2	1 Savuros si sanatos
3	3	0 (null)
4	4	0 (null)
5	5	3 Dulciuri sanatoase
6	6	0 (null)
7	7	1 Preparete vegane
8	8	1 Preparete de vară
9	9	1 Savuros si sanatos
10	10	1 Savuros si sanatos

```
SELECT u.id_utilizator,
```

```
(
```

```
    SELECT COUNT(*)
```

```
    FROM comentarii c
```

```
    JOIN postari p ON (p.id_postare = c.id_postare)
```

```
    JOIN urmareste ur ON (p.id_utilizator = ur.id_urmarit)
```

```
    WHERE ur.id_urmaritor = u.id_utilizator
```

```
) nr_comentarii,
```

```
(
```

```
    SELECT denumire
```

```
    FROM grupuri
```

```

WHERE id_grup = (
    SELECT id_grup FROM (
        -- calculam pentru fiecare grup numarul de urmaritori ai utilizatorului curent care se afla in acel grup,
        -- asezam in ordine descrescatoare a numarului
        SELECT gu.id_grup, COUNT(gu.id_utilizator)
            FROM grup_utilizator gu JOIN urmareste ur ON (gu.id_utilizator =
ur.id_urmarit)
            WHERE ur.id_urmaritor = u.id_utilizator
            GROUP BY gu.id_grup
            ORDER BY COUNT(gu.id_utilizator) DESC
    )
    WHERE ROWNUM < 2
)
) grup
FROM utilizatori u;

```

e) Să se afișeze pentru fiecare postare în parte ratingul său, împreună cu titlul. Ratingul se stabilește astfel: media scorului: 5 stele - Extraordinar, 4 stele - Aproape perfect, 3 stele - Se poate mai bine, 2 stele - Dezamigator, 1 stea - De evitat, dacă nu există aprecieri - Nu a primit scor.

Cererea utilizează:

- clauza WITH
- expresie CASE

The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying a SQL query. The 'Query Result' tab shows the execution results, which is a table with two columns: 'ID_POSTARE' and 'RATING'. The table contains 8 rows of data, each with a post ID and a corresponding rating description.

```

-- calculam pentru fiecare postare media scorului
WITH avg_stele AS (
  SELECT id_postare, AVG(nr_stele) avg
  FROM apreciaza
  GROUP BY id_postare
)
SELECT p.id_postare,
(
  CASE
    WHEN a.avg = 5 THEN p.titlu || ' - Extraordinar'
    WHEN a.avg = 4 THEN p.titlu || ' - Aproape perfect'
    WHEN a.avg = 3 THEN p.titlu || ' - Se poate mai bine'
    WHEN a.avg = 2 THEN p.titlu || ' - Dezamagitor'
    WHEN a.avg = 1 THEN p.titlu || ' - De evitat'
    ELSE p.titlu || ' - Nu a primit scor'
  END
) rating
FROM postari p JOIN avg_stele a ON (p.id_postare = a.id_postare);

```

ID_POSTARE	RATING
1	1 Paine din faina de hrisca - De evitat
2	7 Salata de paste - Aproape perfect
3	8 Keto pizza - Extraordinar
4	5 Tort vegan - Extraordinar
5	10 Cheesecake cu banane coapte - Aproape perfect
6	4 Smoothie cu mere si zmeura - Aproape perfect
7	9 Risotto cu rosii coapte, mozzarella si oregano - Extraordinar
8	3 Supa vegana de dovleac - Extraordinar

-- calculam pentru fiecare postare media scorului

```

WITH avg_stele AS (
  SELECT id_postare, AVG(nr_stele) avg
  FROM apreciaza
  GROUP BY id_postare
)
SELECT p.id_postare,
(
  CASE
    WHEN a.avg = 5 THEN p.titlu || ' - Extraordinar'
    WHEN a.avg = 4 THEN p.titlu || ' - Aproape perfect'
    WHEN a.avg = 3 THEN p.titlu || ' - Se poate mai bine'
    WHEN a.avg = 2 THEN p.titlu || ' - Dezamagitor'
    WHEN a.avg = 1 THEN p.titlu || ' - De evitat'
    ELSE p.titlu || ' - Nu a primit scor'
  END
) rating
FROM postari p JOIN avg_stele a ON (p.id_postare = a.id_postare);

```

FROM postari p JOIN avg_stele a ON (p.id_postare = a.id_postare);

13. Operații de actualizare și suprimare a datelor

```
--STERGEM TOATE COMENTARIILE LA POSTARILE CARE AU PRIMIT MACAR O DATA MAI PUTIN DE DOUA STELE
```

```
--SUPRIMAREA DATELOR CU SUBCERERI
```

```
DELETE FROM comentarii
```

```
WHERE id_postare IN (
```

```
    SELECT id_postare
```

```
        FROM apreciaza
```

```
        WHERE nr_stele <= 2 );
```

The screenshot shows a 'Worksheet' tab in Oracle SQL Developer. The code in the query builder is as follows:

```
1 | SELECT *  
2 | FROM comentarii;  
3 |  
4 | --STERGEM TOATE COMENTARIILE LA POSTARILE CARE AU PRIMIT MACAR O DATA MAI PUTIN DE DOUA STELE  
5 |  
6 | DELETE FROM comentarii  
7 | WHERE id_postare IN (  
8 |     SELECT id_postare  
9 |         FROM apreciaza  
10|         WHERE nr_stele <= 2 );  
11|  
12| SELECT *  
13| FROM comentarii;  
14| ROLLBACK;
```

```
--STERGEM DIN LISTA DE DORINTE A UTILIZATORULUI CU USERNAME-UL 'ADAMKUNTO' POSTAREA CE CONTINE IN TITLU 'PIZZA'
```

```
--SUPRIMAREA DATELOR CU SUBCERERI
```

```
DELETE FROM lista_dorinte
```

```

WHERE id_utilizator =
    SELECT id_utilizator
    FROM utilizatori
    WHERE UPPER(username) = 'ADAMKUNTO'
)

AND id_postare =
    SELECT id_postare
    FROM postari
    WHERE UPPER(titlu) LIKE '%PIZZA%'
);

```

```

1
2
3 --STERGEM DIN LISTA DE DORINTE A UTILIZATORULUI CU USERNAME-UL 'ADAMKUNTO' POSTAREA CE CONTINE IN TITLU 'PIZZA'
4 DELETE FROM lista_dorinte
5 WHERE id_utilizator =
6     SELECT id_utilizator
7     FROM utilizatori
8     WHERE UPPER(username) = 'ADAMKUNTO'
9 )
10 AND id_postare =
11     SELECT id_postare
12     FROM postari
13     WHERE UPPER(titlu) LIKE '%PIZZA%'
14 );

```

```

--FACEM UPDATE LA TABELUL MENIU ASTFEL INCAT ID_POSTARE3 DIN MENIUL CU DENUMIREA
'GUSTARE DULCE' SA CONTINA POSTAREA CU TITLUL 'BRIOSE FARÀ GLUTEN'

-- ACTUALIZAREA DATELOR CU SUBCERERI

UPDATE meniu

SET id_postare3 =
    SELECT id_postare
    FROM postari
    WHERE UPPER(titlu) = 'BRIOSE FARÀ GLUTEN'

)
WHERE UPPER(denumire) LIKE 'GUSTARE DULCE';

```

```

32  SELECT * FROM MENU;
33
34
35
36 --FACEM UPDATE LA TABELUL MENU ASTEPTA INCAT ID_POSTARE3 DIN MENUUL CU DENUMIREA 'GUSTARE DULCE' SA CONTINA POSTAREA CU TITLUL 'BRIOSE FARA GLUTEN'
37
38
39 UPDATE menu
40 SET id_postare3 =
41     (SELECT id_postare
42      FROM postari
43      WHERE UPPER(titlu) = 'BRIOSE FARA GLUTEN'
44    )
45 WHERE UPPER(denumire) LIKE 'GUSTARE DULCE';
46

```

14. Crearea unei vizualizări complexe

O operatie LMD permisa pe o vizualizare complexa este SELECT, iar una nepermisa este DELETE.

```

CREATE VIEW PreparateFaraGluten AS (
    SELECT p.titlu
    FROM postari p
    WHERE NOT EXISTS (
        SELECT 'x'
        FROM postare_alergen pa
        WHERE pa.id_postare = p.id_postare
        AND pa.id_alergen = (
            SELECT id_alergen
            FROM alergen
            WHERE UPPER(descriere) = 'GLUTEN'
        )
    );

```

Worksheet Query Builder

```
CREATE VIEW PreparateFaraGluten AS (
    SELECT p.titlu
    FROM postari p
    WHERE NOT EXISTS (
        SELECT 'x'
        FROM postare_alergen pa
        WHERE pa.id_postare = p.id_postare
            AND pa.id_alergen = (
                SELECT id_alergen
                FROM alergen
                WHERE UPPER(descriere) = 'GLUTEN'
            )
    );
SELECT * FROM PreparateFaraGluten;
```

Script Output x | Query Result x

All Rows Fetched: 7 in 0,013 seconds

TITLU
1 Paine din faina de hrisca
2 Inghetata de fructe
3 Supa vegana de dovleac
4 Smoothie cu mere si zmeura
5 Briose fara gluten
6 Keto pizza
7 Cheesecake cu banane coapte