**TEMĂ**

**1. Identificați în diagrama Entitate-Relație utilizată în proiectul prezentat la materia Baze de**

**Date din anul I, o relație din tip many-to-many. Dacă nu aveți o astfel de relație în acest**

**proiect, atunci definiți una folosind tema aleasă în acel proiect:**

**a. descrieți entitățile și relația dintre acestea;**

**b. realizați diagrama Entitate-Relație doar pentru această parte din proiect;**

**c. descrieți modul de transformare al acestei relații în diagrama conceptuală, precizând**

**toate cheile primare, cheile externe și alte atribute esențiale;**

**d. realizați diagrama conceptuală doar pentru această parte din proiect;**

**e. pe baza diagramei conceptuale de la punctul d, definiți în SQL tabelele și toate**

**constrângerile necesare;**

**f. adaptați una dintre cerințele exercițiilor 4-12 pentru diagrama obținută la punctul d**

**(formulați cerința înlimbaj natural, inserați 5-10 înregistrări în fiecare tabelă utilizată,**

**apoi rezolvați cererea propusă în SQL).**

**Comenzi și Rețete**

Un client poate comanda multiple rețete în aceeași comandă. Comanda va fi înregistrată în sistem de către casier. O comandă poate fi făcută prin telefon în care caz va avea o adresă și un cost de livrare.

O rețetă are un nume și un preț, și este făcută din ingrediente.

O comandă poate conține multiple rețete, și o rețetă poate fi în multiple comenzi (many to many).

**Diagrama ER restrânsă la Comenzi, Rețete și relația many-to-many**

Pentru a permite multiple instanțe din aceeași rețeta într-o comandă (i.e. două pizze) e necesar un câmp „nr”.

RETETA

id\_reteta#

nume

pret

M(0)

M(0)

nr

COMANDA

id\_comanda#

data

LIVRARE

adresa

cost

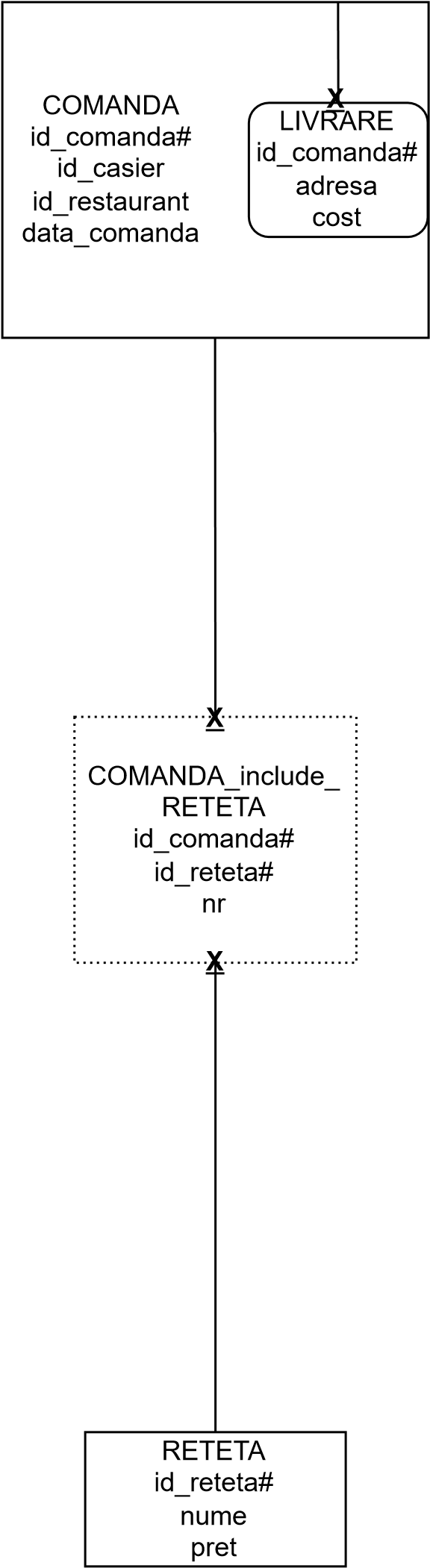
ISA

1:1(0)

**Diagramă conceptuală restrânsă la Comenzi, Rețete, și relația many-to-many**

Relația many-to-many va fi reprezentată printr-un tabel asociativ. Tabelul va conține o pereche de chei externe (id\_comanda#, id\_reteta#) referindu-se la tabelele COMANDA și RETETA. Acea pereche va fi cheie primară compusă în tabelul asociativ. Atributul „nr” va vi inclus în tabelul asociativ deasemenea.

Deoarece COMANDA este în relație many-to-one cu CASIER si RESTAURANT, va conține chei externe pentru acele tabele.

Pentru a reprezenta subclasa LIVRARE, se va folosi un tabel de subclasă cu atribute (id\_comanda#, adresa, cost).

**Creare tabele în SQL**

CREATE SEQUENCE ID\_RETETA\_SEQ START WITH 1 INCREMENT BY 1;

CREATE SEQUENCE ID\_COMANDA\_SEQ START WITH 1 INCREMENT BY 1;

CREATE TABLE RETETA (

  id\_reteta NUMBER(10) DEFAULT ID\_RETETA\_SEQ.nextval PRIMARY KEY,

  nume VARCHAR2(50) NOT NULL,

  pret NUMBER(10,2) NOT NULL

);

CREATE TABLE COMANDA (

  id\_comanda NUMBER(10) DEFAULT ID\_COMANDA\_SEQ.nextval PRIMARY KEY,

  id\_casier NUMBER(10) NOT NULL,

  id\_restaurant NUMBER(10) NOT NULL,

  data\_comanda DATE DEFAULT SYSDATE NOT NULL,

  FOREIGN KEY (id\_casier) REFERENCES CASIER(id\_angajat),

  FOREIGN KEY (id\_restaurant) REFERENCES RESTAURANT(id\_restaurant)

);

CREATE TABLE COMANDA\_include\_RETETA (

  id\_comanda NUMBER(10) NOT NULL,

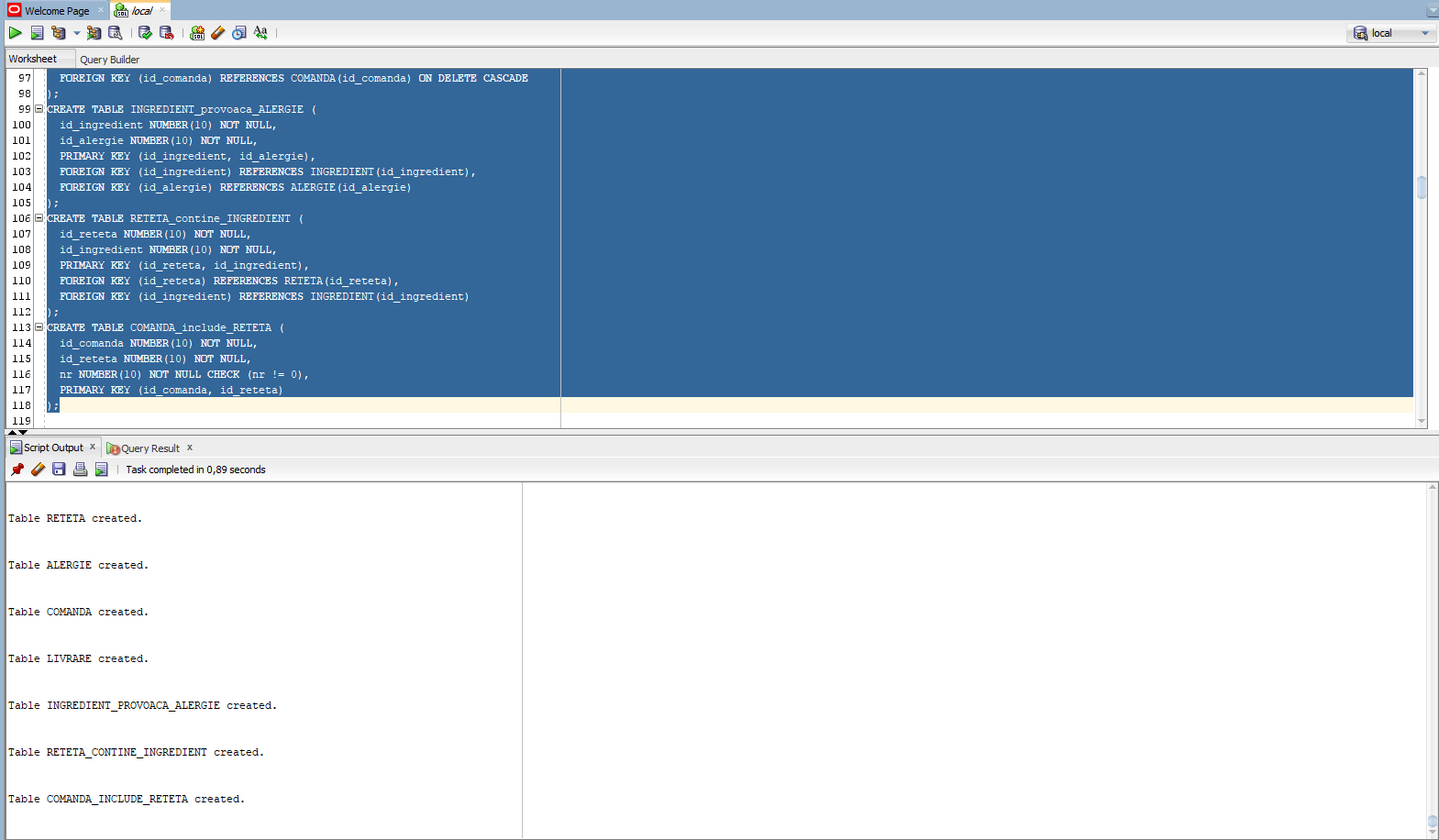
  id\_reteta NUMBER(10) NOT NULL,

  nr NUMBER(10) NOT NULL CHECK (nr != 0),

  PRIMARY KEY (id\_comanda, id\_reteta)

);

(screenshot-ul include mai mult cod deoarece COMANDA are chiei externe ce îl leagă la CASIER și RESTAURANT, deci nu pot fi create în izolare)



Cerință adaptată de la 9: câte vânzări au fost, și care a fost venitul, pentru fiecare rețetă, la fiecare restaurant?

SELECT id\_restaurant, id\_reteta, nume nume\_reteta, SUM(nr) total\_vanzari, SUM(nr) \* pret total\_venit FROM RETETA

LEFT JOIN COMANDA\_include\_RETETA

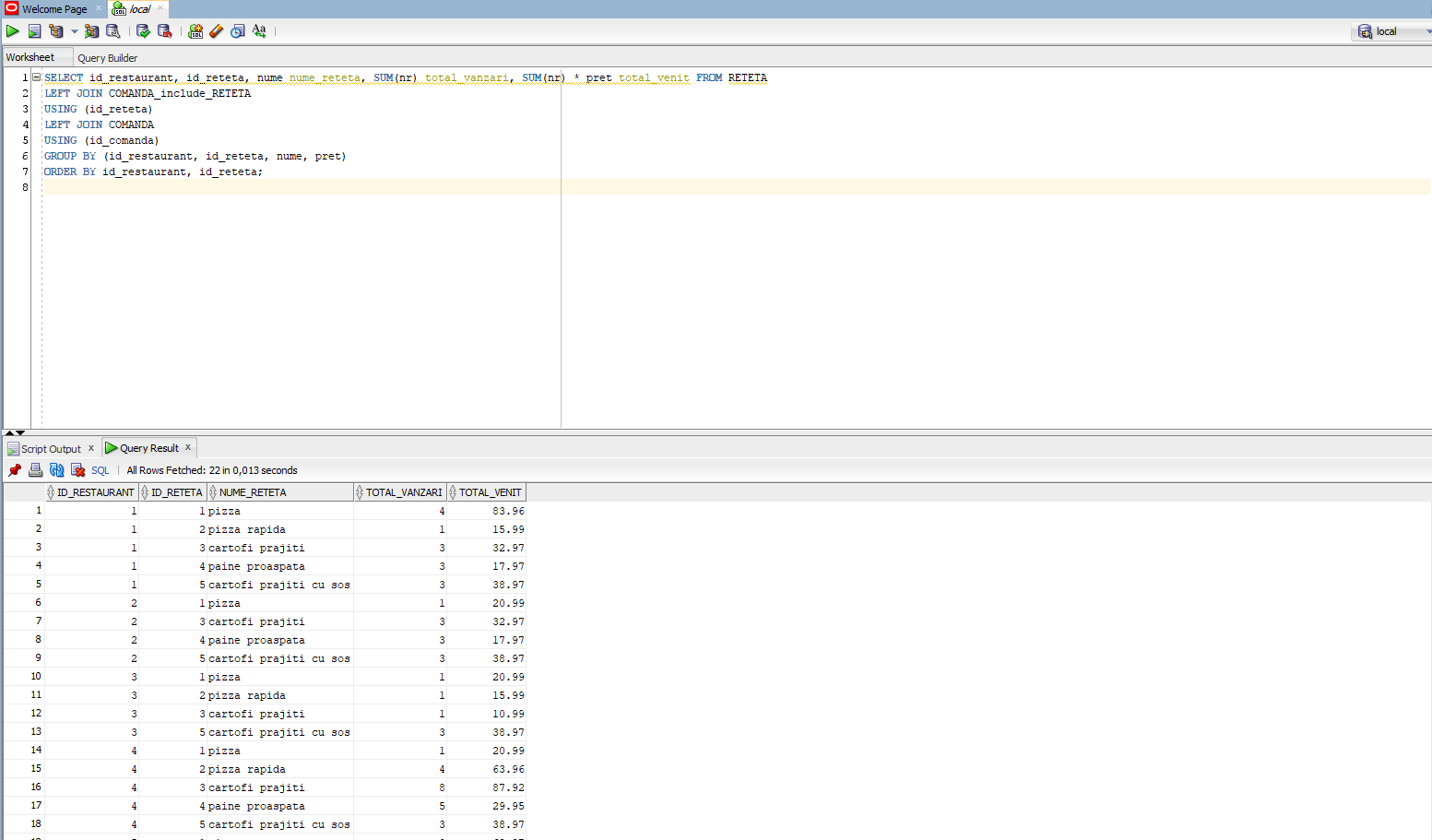
USING (id\_reteta)

LEFT JOIN COMANDA

USING (id\_comanda)

GROUP BY (id\_restaurant, id\_reteta, nume, pret)

ORDER BY id\_restaurant, id\_reteta;



(PS: cred că tema anterioară a fost numită *151*­*\_...s2* în loc de *251­\_...s2* din greșeală. Scuze pentru inconveniență)

**Exerciții pe baza de date HR**

/\*

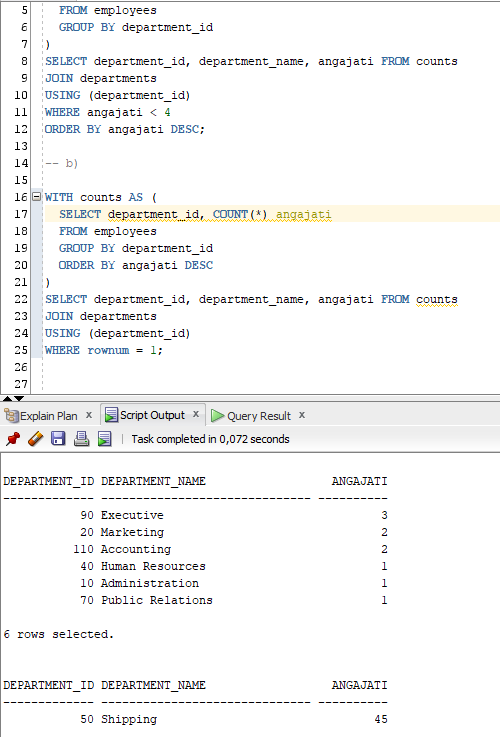
24. Sa se afiseze codul, numele departamentului si numarul de angajati care lucreaza in

acel departament pentru:

a) departamentele in care lucreaza mai putin de 4 angajati;

b) departamentul care are numarul maxim de angajati.

\*/

-- a)

WITH counts AS (

  SELECT department\_id, COUNT(\*) angajati

  FROM employees

  GROUP BY department\_id

)

SELECT department\_id, department\_name, angajati FROM counts

JOIN departments

USING (department\_id)

WHERE angajati < 4

ORDER BY angajati DESC;

-- b)

WITH counts AS (

  SELECT department\_id, COUNT(\*) angajati

  FROM employees

  GROUP BY department\_id

  ORDER BY angajati DESC

)

SELECT department\_id, department\_name, angajati FROM counts

JOIN departments

USING (department\_id)

WHERE rownum = 1;

/\*

25. S? se ob?in? num?rul departamentelor care au cel pu?in 15 angaja?i

\*/

WITH counts AS (

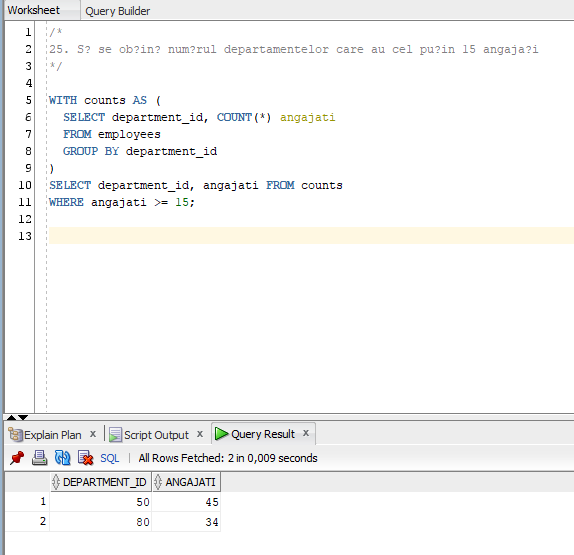
  SELECT department\_id, COUNT(\*) angajati

  FROM employees

  GROUP BY department\_id

)

SELECT department\_id, angajati FROM counts

WHERE angajati >= 15;

/\*

26. Sa se afiseze salariatii care au fost angajati �n aceea?i zi a lunii �n care cei mai multi dintre

salariati au fost angajati.

\*/

WITH hire\_days AS (

  SELECT employee\_id, EXTRACT(DAY FROM hire\_date) hire\_day FROM employees

),

hire\_days\_count AS (

  SELECT hire\_day, COUNT(\*) hire\_count FROM hire\_days

  GROUP BY hire\_day

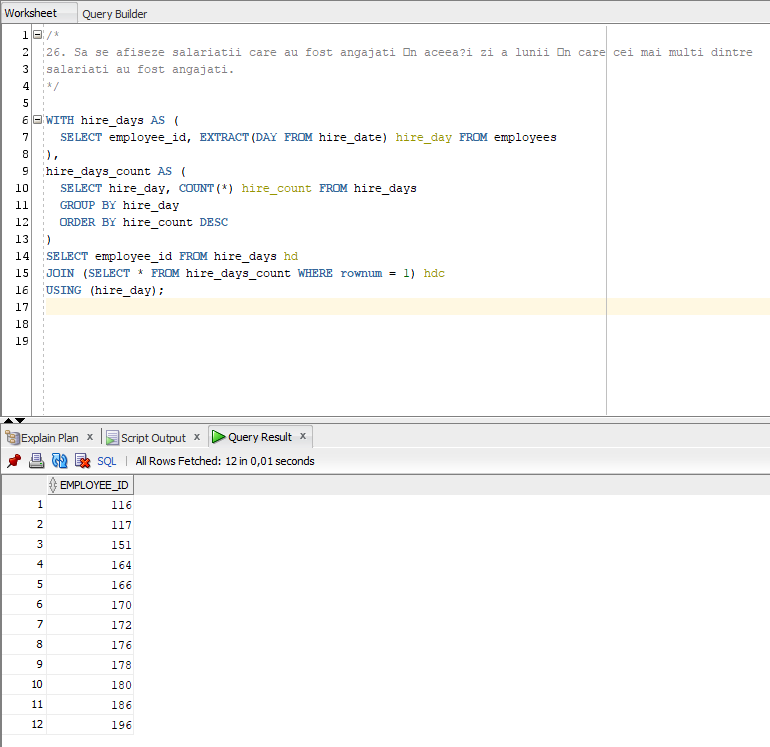
  ORDER BY hire\_count DESC

)

SELECT employee\_id FROM hire\_days hd

JOIN (SELECT \* FROM hire\_days\_count WHERE rownum = 1) hdc

USING (hire\_day);



/\*

27. Sa se afiseze numele si salariul celor mai prost platiti angajati din fiecare departament

\*/

SELECT e1.first\_name || ' ' || e1.last\_name name, e1.department\_id, e1.salary

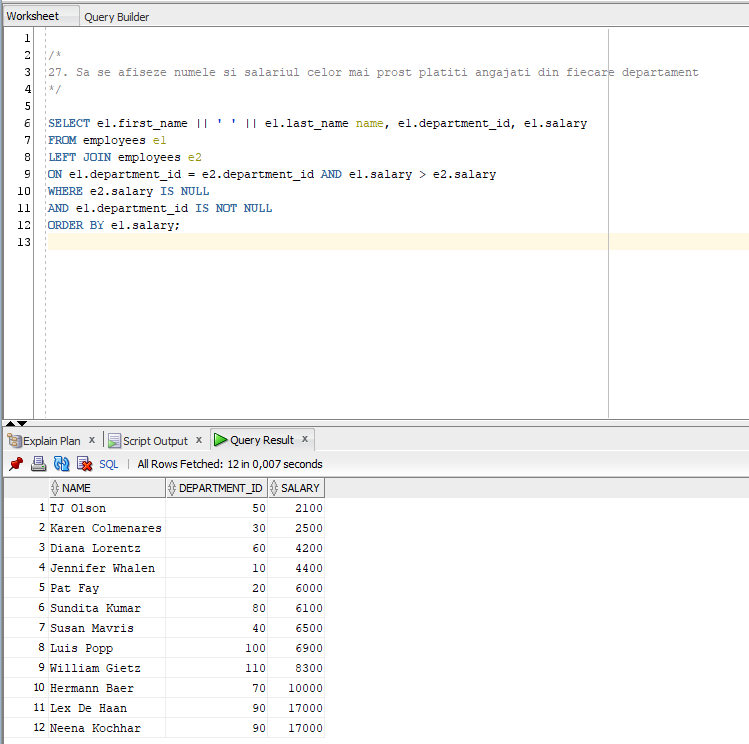
FROM employees e1

LEFT JOIN employees e2

ON e1.department\_id = e2.department\_id AND e1.salary > e2.salary

WHERE e2.salary IS NULL

AND e1.department\_id IS NOT NULL

ORDER BY e1.salary;

/\*

28. S? se detemine primii 10 cei mai bine pl?ti?i angaja?i

\*/

WITH sorted AS (

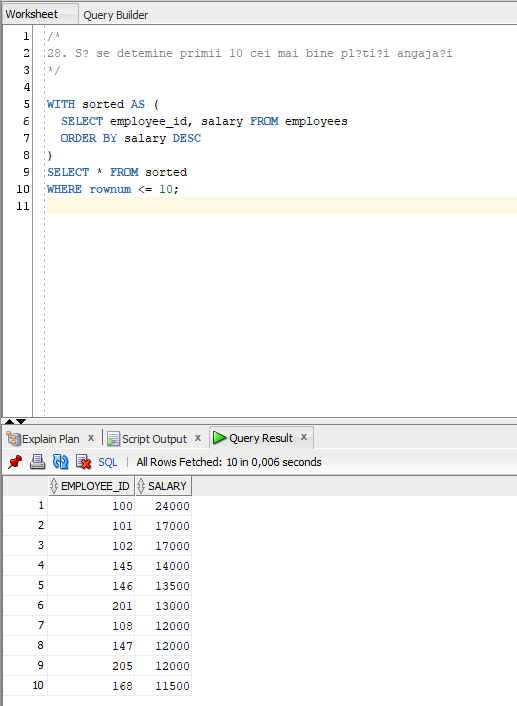
  SELECT employee\_id, salary FROM employees

  ORDER BY salary DESC

)

SELECT \* FROM sorted

WHERE rownum <= 10;



/\*

29. S? se afi?eze codul, numele departamentului ?i suma salariilor pe departamente

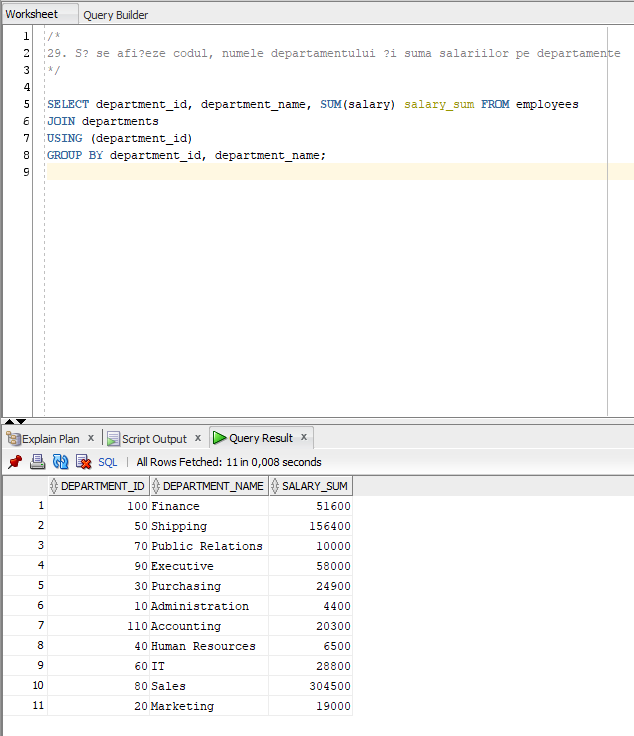
\*/

SELECT department\_id, department\_name, SUM(salary) salary\_sum FROM employees

JOIN departments

USING (department\_id)

GROUP BY department\_id, department\_name;



/\*

30. S? se afi?eze informa?ii despre angaja?ii al c?ror salariu dep??e?te valoarea medie a

salariilor colegilor s?i de departament.

\*/

WITH department\_avgs AS (

 SELECT department\_id, AVG(salary) avg FROM employees

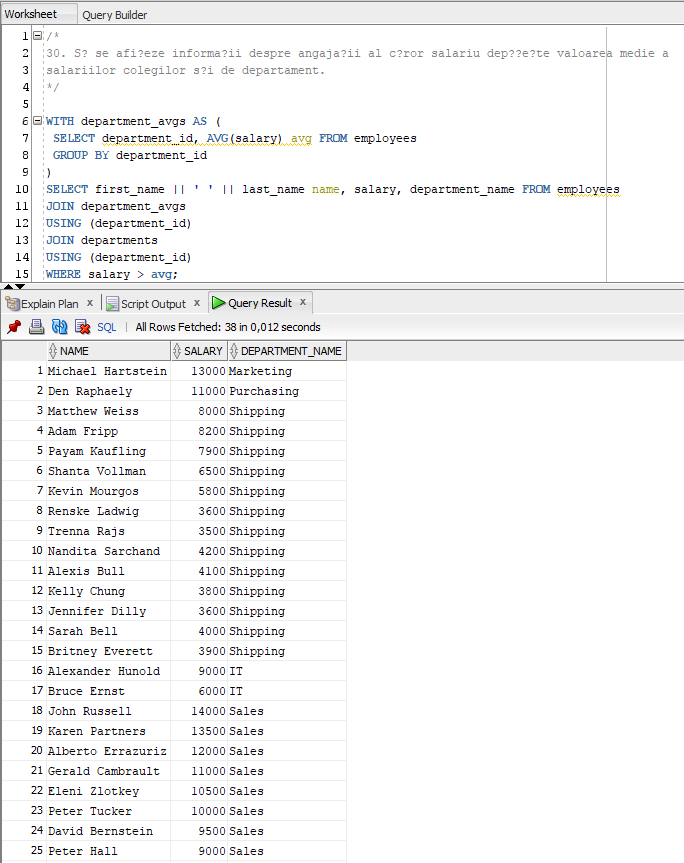
 GROUP BY department\_id

)

SELECT first\_name || ' ' || last\_name name, salary, department\_name FROM employees

JOIN department\_avgs

USING (department\_id)

JOIN departments

USING (department\_id)

WHERE salary > avg;

/\*

31. Sa se afiseze numele si salariul celor mai prost platiti angajati din fiecare departament.

\*/

-- Alta varianta.

WITH min\_salaries AS (

  SELECT department\_id, MIN(salary) min\_salary FROM employees

  GROUP BY (department\_id)

)

SELECT first\_name || ' ' || last\_name name, department\_id, salary FROM employees

JOIN min\_salaries

USING (department\_id)

WHERE salary = min\_salary

ORDER BY salary;

