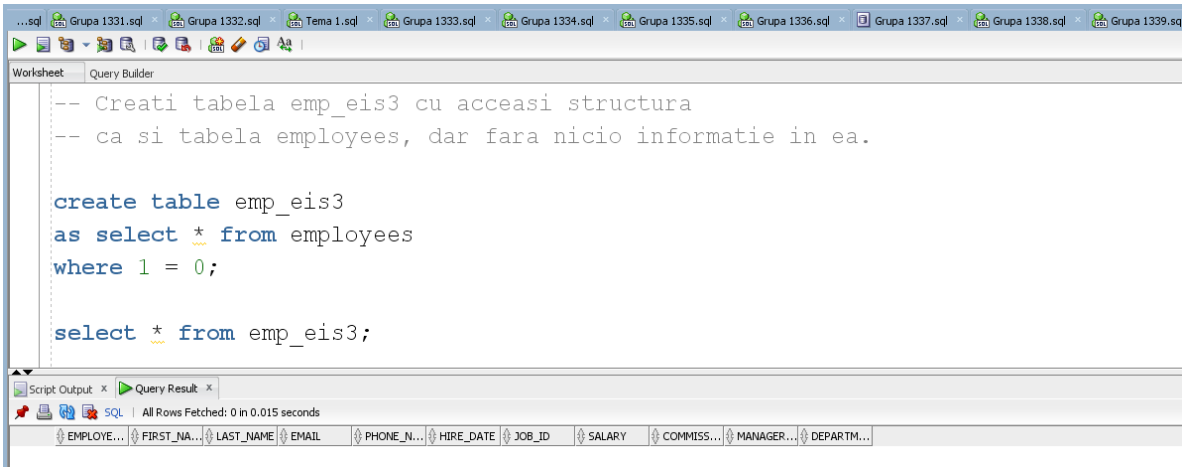
TEMA 2 (LABORATOR BD)

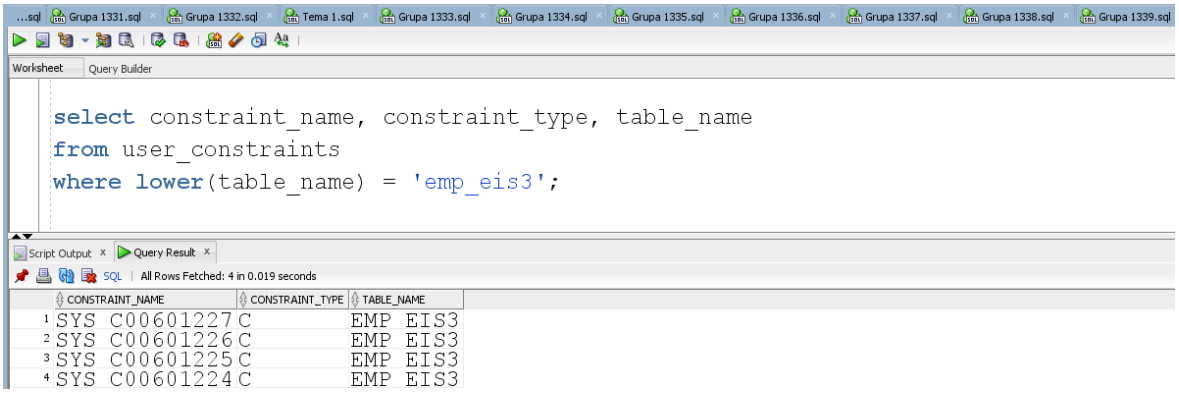
EXERCITIUL 1

create table emp_eis3
as select * from employees
where 1 = 0;

Conditia (where 1=0) este falsa intotdeauna, asadar nu se vor copia datele din tabel, ci doar structura acestuia.

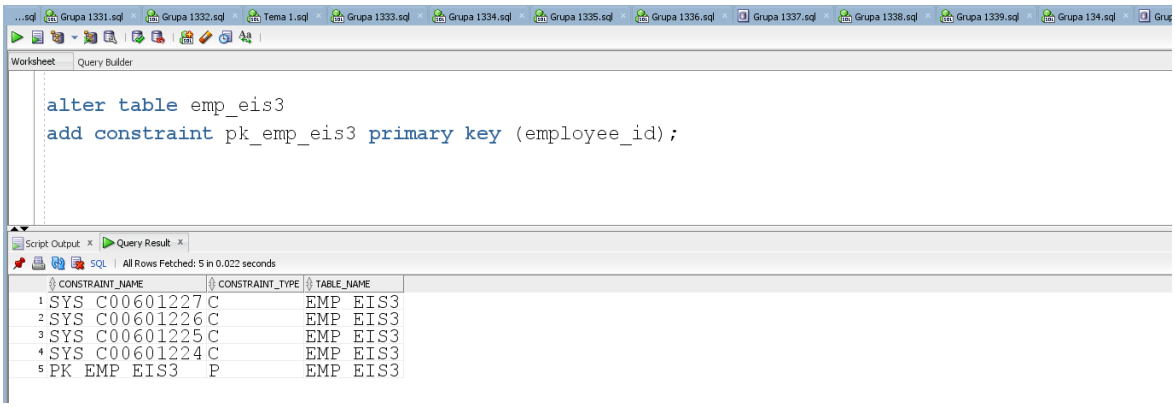


Cheia primara, cheile externe si unele constrangeri nu au fost copiate si ele, acestea trebuie setate de noi. Au fost copiate doar constrangerile de not null.



Pentru exercitiile urmatoare nu avem nevoie de chei externe, asadar nu e necesar sa le setam acum. Constrangerea pentru minimul unui salariu nu afecteaza nici ea cu nimic celelalte exercitii. Cheia primara insa este elementara.

alter table emp_eis3
add constraint pk_emp_eis3 primary key (employee_id);



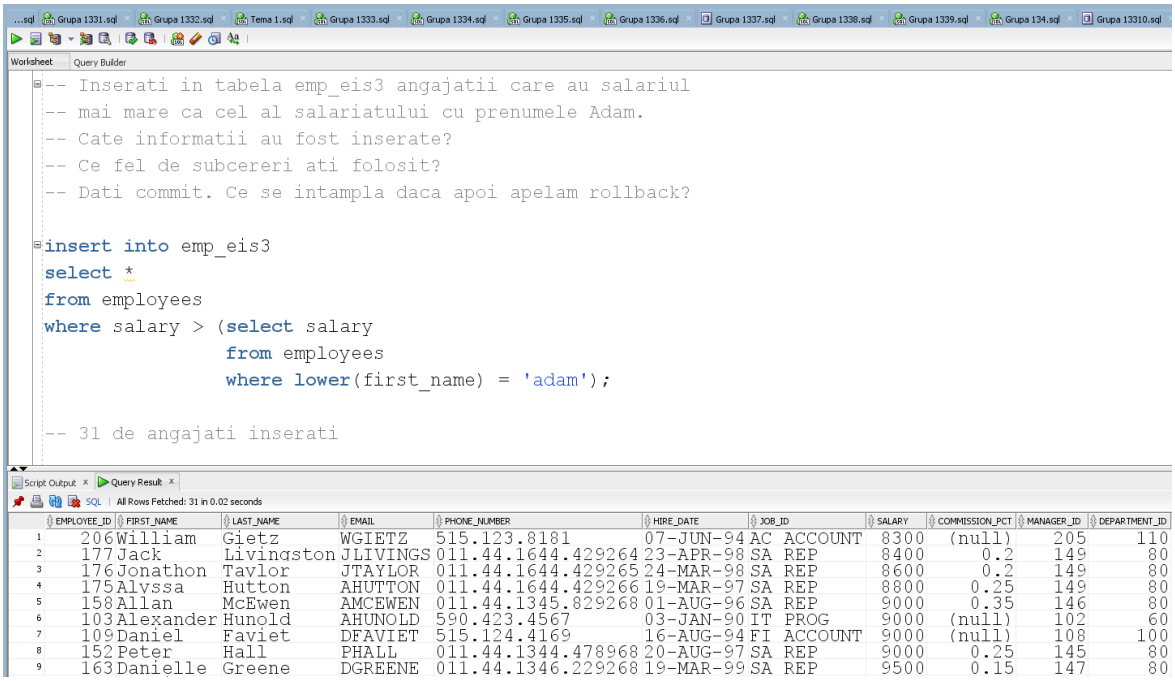
```
alter table emp_eis3
add constraint pk_emp_eis3 primary key (employee_id);
```

Script Output | Query Result | All Rows Fetched: 5 in 0.022 seconds

	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME
1	SYS_C00601227	C	EMP_EIS3
2	SYS_C00601226	C	EMP_EIS3
3	SYS_C00601225	C	EMP_EIS3
4	SYS_C00601224	C	EMP_EIS3
5	PK_EMP_EIS3	P	EMP_EIS3

EXERCITIUL 2

```
insert into emp_eis3
select *
from employees
where salary > (select salary
                from employees
                where lower(first_name) = 'adam');
```



```
-- Inserati in tabela emp_eis3 angajatii care au salariul
-- mai mare ca cel al salariatului cu prenumele Adam.
-- Cate informatii au fost inserate?
-- Ce fel de subcereri ati folosit?
-- Dati commit. Ce se intampla daca apoi apelam rollback?

insert into emp_eis3
select *
from employees
where salary > (select salary
                from employees
                where lower(first_name) = 'adam');

-- 31 de angajati inserati
```

Script Output | Query Result | All Rows Fetched: 31 in 0.02 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	206	William	Gietz	WGIETZ	515.123.8181	07-JUN-94	AC_ACCOUNT	8300	(null)	205	110
2	177	Jack	Livingston	JLIVINGS	011.44.1644.429264	23-APR-98	SA_REP	8400	0.2	149	80
3	176	Jonathon	Taylor	JTAYLOR	011.44.1644.429265	24-MAR-98	SA_REP	8600	0.2	149	80
4	175	Alyssa	Hutton	AHUTTON	011.44.1644.429266	19-MAR-97	SA_REP	8800	0.25	149	80
5	158	Allan	McEwen	AMCEWEN	011.44.1345.829268	01-AUG-96	SA_REP	9000	0.35	146	80
6	103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000	(null)	102	60
7	109	Daniel	Faviet	DFAVIET	515.124.4169	16-AUG-94	FI_ACCOUNT	9000	(null)	108	100
8	152	Peter	Hall	PHALL	011.44.1344.478968	20-AUG-97	SA_REP	9000	0.25	145	80
9	163	Danielle	Greene	DGREENE	011.44.1346.229268	19-MAR-99	SA_REP	9500	0.15	147	80

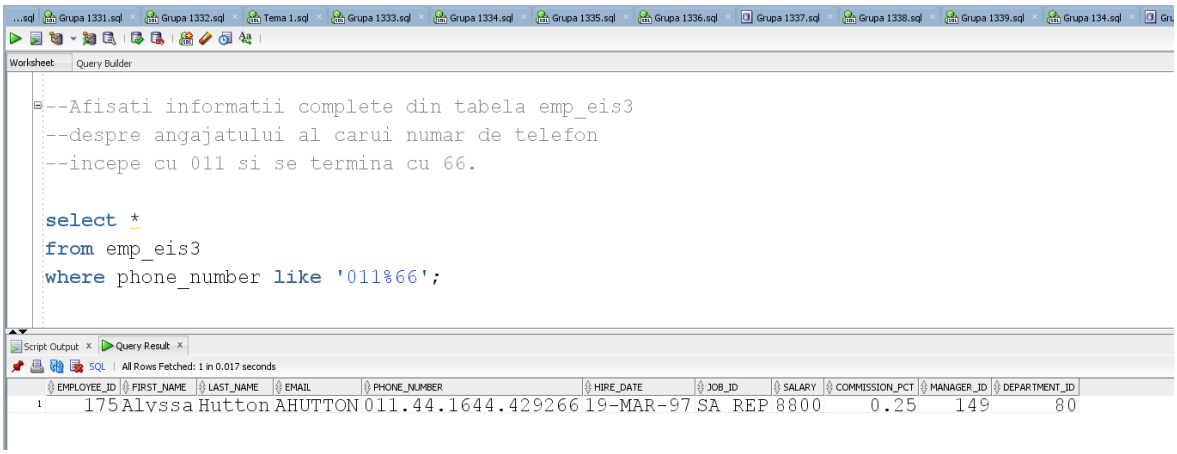
Au fost inserate 31 de informatii.

Am folosit o subcerere monolinie, necorelata. Aceasta este executata prima si returneaza salariul angajatului cu prenumele Adam (8200). Apoi sunt selectati angajatii cu salariul mai mare si inserati in tabel. Cel mai mic salariu din tabelul nostru este 8300.

Daca dam commit si apoi dam rollback, informatiile vor ramane in tabel. Daca dadeam rollback fara sa dam commit, tabelul era gol, asa cum l-am setat inainte sa inseram informatii.

EXERCITIUL 3

```
select *
from emp_eis3
where phone_number like '011%66';
```



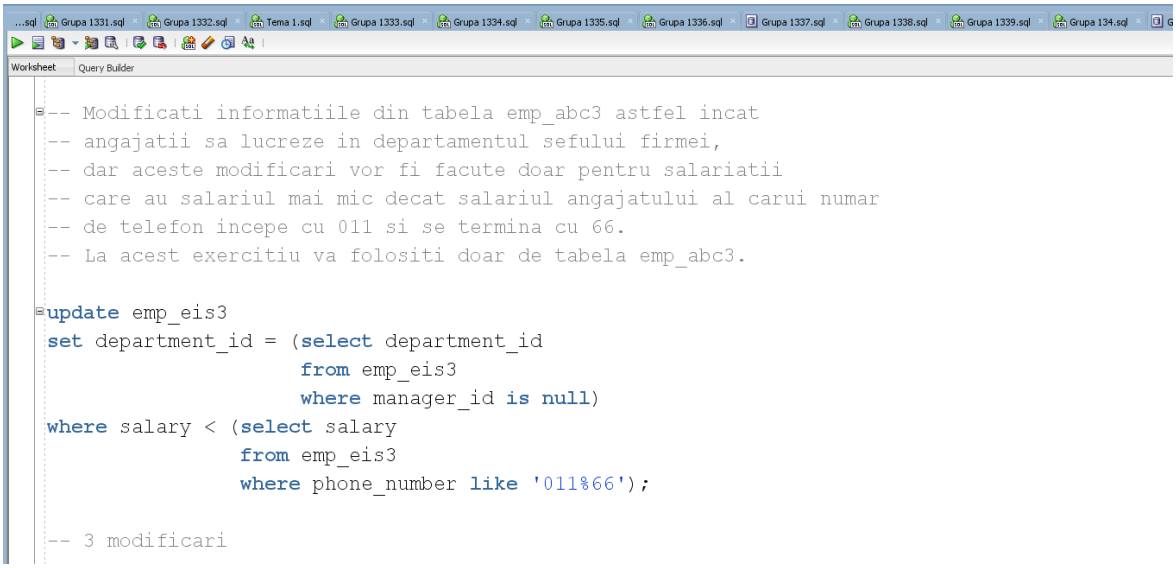
```
--Afisati informatii complete din tabela emp_eis3
--despre angajatului al carui numar de telefon
--incepe cu 011 si se termina cu 66.

select *
from emp_eis3
where phone_number like '011%66';
```

Script Output | Query Result | All Rows Fetched: 1 in 0.017 seconds

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1 175	Alyssa	Hutton	AHUTTON	011.44.1644.429266	19-MAR-97	SA_REP	8800	0.25	149	80

```
update emp_eis3
set department_id = (select department_id
                     from emp_eis3
                     where manager_id is null)
where salary < (select salary
                from emp_eis3
                where phone_number like '011%66');
```



```
-- Modificati informatiile din tabela emp_abc3 astfel incat
-- angajatii sa lucreze in departamentul sefului firmei,
-- dar aceste modificari vor fi facute doar pentru salariatii
-- care au salariul mai mic decat salariul angajatului al carui numar
-- de telefon incepe cu 011 si se termina cu 66.
-- La acest exercitiu va folositi doar de tabela emp_abc3.

update emp_eis3
set department_id = (select department_id
                     from emp_eis3
                     where manager_id is null)
where salary < (select salary
                from emp_eis3
                where phone_number like '011%66');

-- 3 modificari
```

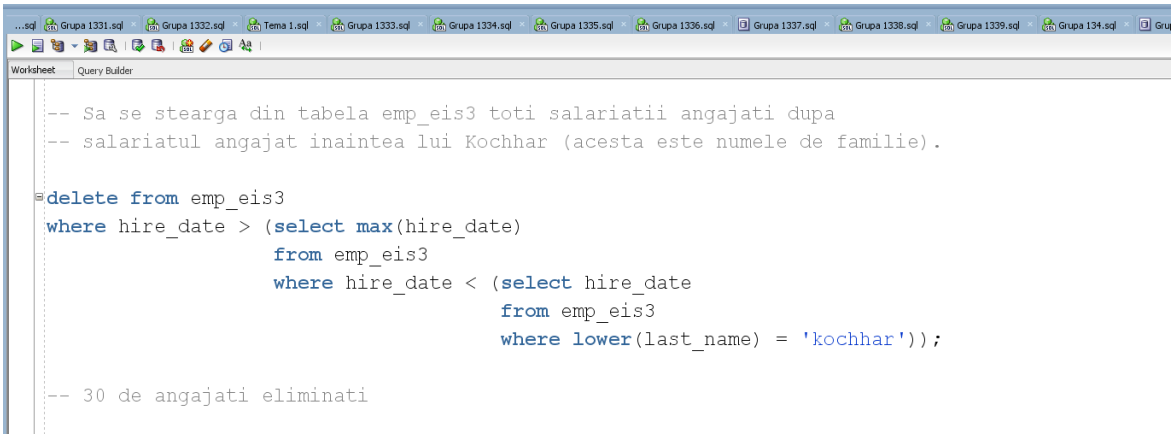
Au fost modificate 3 informatii.

Am folosit doua subcereri monolinii, necorelate. Prima este pentru a selecta departamentul sefului, departament ce urmeaza sa fie setat angajatilor. A doua selecteaza salariul angajatului al carui numar de telefon incepe cu 011 si se termina cu 66. Angajatul este unic deoarece tabelul din care am copiat datele avea constrangere de unicitate pentru numarul de telefon. Cand rulam comanda se va modifica departamentul doar pentru angajatii ce respecta conditia din where, mai exact cei care au salariul mai mic decat salariul lui Alvssa Hutton (8800).

Dupa ce am rulat comanda, departamentele celor trei sunt 90 (departamentul sefului). Daca dam rollback, departamentele lor vor reveni la cele initiale: 110, respective 80. Pentru a seta modificarea facuta va trebui sa dam commit.

EXERCITIUL 4

```
delete from emp_eis3
where hire_date > (select max (hire_date)
                   from emp_eis3
                   where hire_date < (select hire_date
                                      from emp_eis3
                                      where lower(last_name) = 'kochhar'));
```

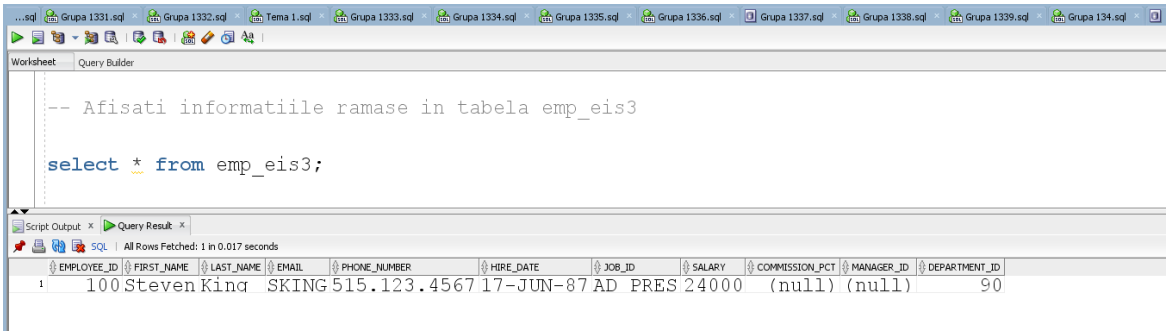


Au fost sterse 30 de informatii deoarece inaintea lui Kochhar era angajata o singura persoana, seful firmei. A ramas un singur angajat in tabel, seful firmei.

Am folosit doua subcereri necorelate. A doua subcerere returneaza data de angajare a lui Kochhar, iar prima subcerere returneaza maximul dintre angajarile facute inaintea lui Kochhar. Acest maxim va indica exact salariatul angajat inaintea lui Kochhar. Se vor sterge salariatii angajati dupa salariatul angajat inaintea lui Kochhar.

EXERCITIUL 5

select * from emp_eis3;



drop table emp_eis3;