11,12,13,16 exercitii.

--SCRIE CATE CURSURI,SEDINTE,PROIECTE(IN TOTAL ) PREDA COREGRAFUL NASCUT PE 1 IANUARIE 1998

SELECT COUNT(\*)

FROM ((COREGRAF INNER JOIN CURS ON COREGRAF.ID\_ANG=CURS.ID\_ANG) INNER JOIN PROIECT ON (COREGRAF.ID\_ANG=PROIECT.ID\_ANG))JOIN SEDINTA\_PRIVATA ON (COREGRAF.ID\_ANG=SEDINTA\_PRIVATA.ID\_ANG)

WHERE COREGRAF.DATA\_NASTERE='1-JAN-1998';

--Sterge proiectele ale caror zi de desfasurare a trecut,care si-au vandut toate locurile sau care nu au vandut destule locuri la timp, foloseste sysdate si least si e o subcerere de suprimare a datelor

(adaugam ce urmeaza sa stergem)

INSERT INTO PROIECT VALUES(7,'bulgaria',14,'3-JUN-2021',2,250,10,25,2);

INSERT INTO PROIECT VALUES(11,'bulgaria',14,'3-JUN-2021',2,250,10,25,2);

INSERT INTO PROIECT VALUES(6,'TIMIS',14,'5-JUN-2021',2,250,10,13,2);

INSERT INTO PROIECT VALUES(9,'TIMISA',14,'5-JUN-2021',2,250,10,12,2);

INSERT INTO PROIECT VALUES(8,'TIMIS',14,'15-JUN-2021',2,250,10,0,2);

INSERT INTO PROIECT VALUES(10,'TIMIS',14,'15-JUN-2021',2,250,10,0,2);

DELETE FROM PROIECT

WHERE PROIECT\_ID IN (SELECT PROIECT\_ID

FROM PROIECT

WHERE (LEAST(SYSDATE,ZIUA\_P)=ZIUA\_P) or ((least(sysdate,ziua\_p)=sysdate and least(sysdate+2,ziua\_p)=ziua\_p) and nr\_locuri>11) or nr\_locuri=0);

--Sterge rezervarile facute de client la cursurile la care nu au voie din cauza limitei de varsta, foloseste conversia datelor si e o subcerere de suprimare

DELETE FROM REZERVARE

WHERE REZERVARE\_ID in (SELECT REZERVARE\_ID

FROM ((REZERVARE INNER JOIN CLIENTI ON CLIENTI.CLIENT\_ID=REZERVARE.CLIENT\_ID) INNER JOIN CURS ON CURS.CURS\_ID=REZERVARE.CURS\_ID) INNER JOIN TIPDANS ON TIPDANS.TIPDANS\_ID=CURS.TIPDANS\_ID

WHERE (TO\_NUMBER(TO\_CHAR(SYSDATE,'YYYY'))- TO\_NUMBER(TO\_CHAR(DATA\_NASTERE,'YYYY')))<LIMITA\_VARSTA);

(adaugam ce stergem)

INSERT INTO REZERVARE VALUES(11,15,17,NULL,NULL);

INSERT INTO REZERVARE VALUES(12,15,18,NULL,NULL);

--Afla numarul de locuri valabile pt rezervare de la cursuri atunci cand un loc este rezervat si le ordoneaza crescator dupa acel nr

SELECT a.idcc cursid,a.nrl vechi,a.nrl-a.scad nou

FROM (SELECT B.NR scad,B.idc idcc,CURS.NR\_LOCURI nrl

FROM (SELECT COUNT(\*) NR,curs.curs\_id idc

FROM CURS right JOIN REZERVARE ON CURS.CURS\_ID=REZERVARE.CURS\_ID

group by curs.curs\_id)B INNER JOIN CURS ON B.idc=CURS.CURS\_ID)a

ORDER BY a.nrl-a.scad;

--STERGE REZERVARILE ANULATE

ALTER TABLE ANULARE

ADD REZERVARE\_ID INTEGER;

ALTER TABLE ANULARE

ADD CONSTRAINT REZERVARE\_ID

FOREIGN KEY (REZERVARE\_ID)

REFERENCES REZERVARE (REZERVARE\_ID) ON DELETE CASCADE; (NU LE LEGASEM INITIAL)

TRUNCATE TABLE ANULARE;

INSERT INTO ANULARE VALUES(10,'LUNI',2);

INSERT INTO ANULARE VALUES(14,'MARTI',4);

INSERT INTO ANULARE VALUES(9,'LUNI',6);

INSERT INTO ANULARE VALUES(13,'MARTI',8);

INSERT INTO ANULARE VALUES(15,'LUNI',1);

INSERT INTO ANULARE VALUES(11,'LUNI',12);

DELETE FROM REZERVARE

WHERE REZERVARE\_ID in (SELECT REZERVARE.REZERVARE\_ID

FROM (REZERVARE INNER JOIN ANULARE ON REZERVARE.REZERVARE\_ID=ANULARE.REZERVARE\_ID) INNER JOIN CURS ON REZERVARE.CURS\_ID=CURS.CURS\_ID

WHERE CURS.ORA>ANULARE.ORA\_ANULARE+1);

SELECT\*

FROM REZERVARE;

SELECT\*

FROM ANULARE;

INSERT INTO REZERVARE VALUES(2,1,18,NULL,NULL);

INSERT INTO REZERVARE VALUES(4,3,2,NULL,NULL);

INSERT INTO REZERVARE VALUES(6,5,4,NULL,NULL);

INSERT INTO REZERVARE VALUES(8,2,17,NULL,NULL);

13— sequence

CREATE SEQUENCE sequence\_1

start with 14

increment by 1

minvalue 14

maxvalue 100

cycle;

INSERT INTO REZERVARE VALUES(sequence\_1.nextval,10,2,NULL,NULL);

INSERT INTO REZERVARE VALUES(sequence\_1.nextval,8,2,NULL,NULL);

16 --AFISEAZA TOATE CURSURILE CU NR LOR DE REZERVARI UNDE PRETUL E DE 40 LEI SAU CLIENTULL NU ARE NUMELE DIANA,MARIA,ALINA

SELECT CURS.CURS\_ID,COUNT(REZERVARE.CURS\_ID)

FROM ((REZERVARE RIGHT OUTER JOIN CURS ON CURS.CURS\_ID=REZERVARE.CURS\_ID) LEFT OUTER JOIN TIPDANS ON TIPDANS.TIPDANS\_ID=CURS.TIPDANS\_ID)LEFT OUTER JOIN CLIENTI ON CLIENTI.CLIENT\_ID=REZERVARE.CLIENT\_ID

WHERE TIPDANS.PRET=40 OR CLIENTI.NUME NOT IN ('DIANA','MARIA','ALINA')

GROUP BY CURS.CURS\_ID

ORDER BY CURS.CURS\_ID;

--AFISEAZA NUMELE CU LITERA MARE SI PRENUMELE DOAR IN LITERE MICI ALE CLIENTILOR CARE AU FACUT REZERVARI SI AU ABONAMENT SAU VOUCHER CU UN ID MAI MARE DECAT UNU-SUBCERERE NESINCRONIZATA

SELECT INITCAP(NUME),LOWER(PRENUME)

FROM CLIENTI

WHERE CLIENT\_ID IN (SELECT CLIENT\_ID

FROM REZERVARE) AND NVL(VOUCHER\_ID,ABONAMENT\_ID)>1;

--AFISEAZA ID-UL CURSULUI CU DENUMIREA TIPULUI DE DANS SI NIVELUL ACESTUIA(INCEPATOR,AVANSAT,INTERMEDIAR)

SELECT CURS.CURS\_ID,TIPDANS.DENUMIRE, DECODE(CURS.DIFICULTATE,1,'INCEPATOR',2,'INTERMEDIAR',3,'AVANSAT',0)

FROM CURS INNER JOIN TIPDANS ON CURS.TIPDANS\_ID=TIPDANS.TIPDANS\_ID

ORDER BY CURS.CURS\_ID;

--SAU CU CASE

SELECT CURS.CURS\_ID,TIPDANS.DENUMIRE, (CASE DIFICULTATE

WHEN 1 THEN 'INCEPATOR'

WHEN 2 THEN 'INTERMEDIAR'

WHEN 3 THEN 'AVANSAT'

ELSE '0'

END)

FROM CURS INNER JOIN TIPDANS ON CURS.TIPDANS\_ID=TIPDANS.TIPDANS\_ID

ORDER BY CURS.CURS\_ID;

--scrieti id-ul cursurilor care au fost rezervate de 4 oameni

WITH PATRUREZERVARI(IDC) as (SELECT B.idc

FROM (SELECT COUNT(\*) NR,curs.curs\_id idc

FROM CURS right JOIN REZERVARE ON CURS.CURS\_ID=REZERVARE.CURS\_ID

group by curs.curs\_id)B

WHERE B.NR=4)

SELECT CURS\_ID

FROM CURS,PATRUREZERVARI

WHERE CURS\_ID=IDC;