Tabelul STUDENT

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ST\_ ID | NUME | PRENUME | BURSA | ANUL | ORAS | DATA\_N | UNIV\_ID |
| 1 | Popescu | Ion | 150 | 1 | Оrhei | 3/12/1982 | 10 |
| 3 | Ionescu | Petru | 200 | 3 | Cahul | 1/12/1980 | 10 |
| 6 | Sima | Vadim | 150 | 4 | Chişinău | 7/06/1979 | 22 |
| 10 | Osman | Nicolai | 0 | 2 | Briceni | 8/12/1981 | 10 |
| 12 | Casta | Olga | 250 | 2 | Lipcani | 1/05/1981 | 10 |
| 265 | Mantes | Andrei | 0 | 3 | Bălţi | 5/11/1979 | 10 |
| 32 | Colesnic | Petru | 150 | 5 | Drochia | NULL | 14 |
| 654 | Luca | Artur | 200 | 3 | Bălţi | 1/12/1981 | 10 |
| 276 | Petrescu | Elena | 200 | 4 | NULL | 5/08/1981 | 22 |
| 55 | Bănescu | Vald | 250 | 5 | Bălţi | 7/01/1980 | 10 |

ST\_ID — cod numeric ce identifică studentul,

NUME — numele studentului,

PRENUME — prenumele studentului,

BURSA — bursa primită de student,

ANUL — anul în care învaţă studentul,

ORAS — oraşul în care locuieşte studentul,

DATA\_N — data de naştere a studentului,

UNIV\_ID — cod numeric ce identifică universitatea în care învaţă studentul.

Tabelul PROFESOR

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PR\_ ID | NUME | PRENUME | ORAS | UNIV\_ID |
| 24 | Iordachi | Boris | Bălţi | 10 |
| 46 | Nuca | Ion | Bălţi | 10 |
| 74 | Lăsescu | Pavel | Chişinău | 22 |
| 108 | Stănescu | Nicolai | Chişinău | 22 |
| 276 | Negrescu | Victor | Bălţi | 10 |
| 328 | Sorescu | Andrei | Оrhei | 10 |

PR\_ID — cod numeric ce identifică profesorul,

NUME — numele profesorului,

PRENUME — prenumele profesorului,

ORAS — oraşul în care locuieşte profesorul,

UNIV\_ID — cod numeric ce identifică universitatea în care lucrează profesorul.

Tabelul CURS (Cursurile predate)

|  |  |  |  |
| --- | --- | --- | --- |
| CURS\_ID | CURS\_DEN | ORE | SEMESTRU |
| 10 | Informatica | 56 | 1 |
| 22 | Fizica | 34 | 1 |
| 43 | Matematica | 56 | 2 |
| 56 | Istoria | 34 | 4 |
| 94 | Limba Engleză | 56 | 3 |
| 73 | Educaţia fizică | 34 | 5 |

CURS\_ID — idntificatorul cursului predat,

CURS\_DEN — dumirea cursului predat,

ORE — numărul de ore planificate pentru cursul dat,

SEMESTRU — semestrul în care se citeşte cursul.

Tabelul UNIVERSITATE

|  |  |  |  |
| --- | --- | --- | --- |
| UNIV\_ID | UNIV\_DEN | RATING | ORAS |
| 22 | USM | 606 | Chişinău |
| 10 | USB | 296 | Bălţi |
| 11 | ULIM | 345 | Chişinău |
| 32 | USC | 416 | Cahul |
| 14 | USD | 326 | Drochia |
| 15 | USC | 368 | Chişinău |
| 18 | ВГМА | 327 | Bălţi |

UNIV\_ID — identificatorul universităţii,

UNIV\_DEN — denumirea universităţii,

RATING — reitingul universităţii,

ORAS — oraşul unde se află universitatea.

Tabelul NOTE\_EXAM (Notele de la examene)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EXAM\_ID | ST\_ID | CURS\_ID | NOTA | DATA \_ EXAM |
| 145 | 12 | 10 | 10 | 12/01/2000 |
| 34 | 32 | 10 | 9 | 23/01/2000 |
| 75 | 55 | 10 | 9 | 05/01/2000 |
| 238 | 12 | 22 | 6 | 17/06/1999 |
| 639 | 55 | 22 | NULL | 22/06/1999 |
| 43 | 6 | 22 | 5 | 18/01/2000 |

EXAM\_ID — identificatorul examenului,

ST\_ID — identificatorul studentului,

CURS\_ID — identificatorul cursului,

NOTA — nota de la examen,

DATA\_EXAM — data promovării examenului.

Tabelul CURS\_PROF (Cursurile profesorilor)

|  |  |
| --- | --- |
| PR \_ID | CURS\_ID |
| 24 | 24 |
| 46 | 46 |
| 74 | 74 |
| 108 | 108 |
| 276 | 276 |
| 328 | 328 |

PR \_ID — identificatorul profesorului,

CURS\_ID — identificatorul cursului predat.

1. Să se formuleze o interogare care ar afişa toată informaţia despre studenţi.

SELECT\*

FROM STUDENT

1. Să se formuleze o interogare care ar afişa lista oraşelor în care locuiesc studenţii.

SELECT DISTINCT ORAS

FROM STUDENT

WHERE ORAS IS NOT NULL

1. Să se formuleze o interogare care ar afişa numele şi prenumele studenţilor ce au numele Petrov.

SELECT NUME, PRENUME

FROM STUDENT

WHERE NUME =’PETROV’

1. Să se formuleze o interogare care ar afişa lista studenţii din anul trei ce au bursă.

SELECT\*

FROM STUDENT

WHERE ANUL = 3 AND BURSA>0

1. Să se formuleze o interogare care ar afişa identificatorului (codului numeric) cursului, denumirea lui, semestrul în care el este predat precum şi numărul de ore planificat pentru acest curs pentru toate cursurile din tabelul Cursuri.

SELECT\*

FROM CURS

1. Să se formuleze o interogare care ar afişa toate înregistrărilor din tabelul NOTE, pentru cursul cu numărul de identificare egal cu 12.

SELECT\*

FROM NOTE\_EXAM

WHERE CURS\_ID=12

1. Să se formuleze o interogare care ar afişa informaţia din tabelul STUDENT, ordonînd cîmpurile în felul următor: ANUL, PRENUME, NUME, BURSA.

SELECT ANUL, PRENUME, NUME, BURSA

FROM STUDENT

1. Să se formuleze o interogare care ar afişa denumirea cursului, numărul de ore planificate pentru toate cursurile predate în semestrul patru.

SELECT CURS\_DEN, ORE

FROM CURS

WHERE SEMESTRU = 4

1. Să se formuleze o interogare care ar afişa informaţia din cîmpul NOTE a tabelei NOTE\_EXAM pentru fiecare student evitînd duplicatele.

SELECT DISTINCT NOTA

FROM NOTE EXAM

WHERE NOTA IS NOT NULL

1. Să se formuleze o interogare care ar afişa numele studenţilor ce învaţă în anul trei şi mai mare.

SELECT NUME  
FROM STUDENT

WHERE ANUL >= 3

1. Să se formuleze o interogare care ar afişa numele, prenumele şi anul de studii a studenţilor ce au bursa mai mare ca 140 lei.

SELECT NUME, PRENUME, ANUL

FROM STUDENT

WHERE BURSA > 140

1. Să se formuleze o interogare care ar afişa lista cursurilor pentru care au fost planificate mai mult de 30 ore.

SELECT \*

FROM CURS

WHERE ORE>30

1. Să se formuleze o interogare care ar afişa lista universităţilor reitingul cărora depăşeşte 300 puncte.

SELECT \*

FROM UNIVERSITATE

WHERE RATING>300

1. Să se formuleze o interogare (la tabela STUDENT) care ar afişa numele, prenumele şi anul de studii a studenţilor ce au bursa egală sau mai mare ca 100 şi care sînt originari din Bălţi.

SELECT NUME, PRENUME, ANUL

FROM STUDENT

WHERE BURSA >= 100 AND ORAS =’BALTI’

1. Să se formuleze o interogare care ar afişa numele doar acelor studenţi ce au note de 9 şi 10.

SELECT NUME, NOTA

FROM STUDENT INNER JOIN NOTE\_EXAM

ON STUDENT.ST\_ID=NOTE\_EXAM.ST\_ID

WHERE NOTA IN (9,10)

1. Să se formuleze o interogare care ar afişa cursurile pentru care numărul de ore planificate este mai mare ca 30, dar mai mic ca 40.

SELECT \*

FROM CURS

WHERE ORE>30 AND ORE <40

1. Să se formuleze o interogare care ar afişa informaţia despre studenţii numele cărora începe cu „P”.

SELECT \*

FROM STUDENT

WHERE NUME LIKE’P%’

1. Să se formuleze o interogare care ar afişa codul numeric al cursurilor examenele cărora au fost promovate între 10 şi 20 ianuarie 1999.

SELECT CURS\_ID

FROM NOTE\_EXAM

WHERE DATA\_EXAM>’1999-01-10’ AND DATA\_EXAM<’1999-01-20’

1. Să se formuleze o interogare care ar afişa informaţia despre acele cursuri la care examenele au fost susţinute de către studenţii cu numărul de indetificare 12 şi 32.

SELECT \*

FROM CURS

WHERE CURS\_ID IN (SELECT CURS\_ID

FROM NOTE\_EXAM

WHERE ST\_ID IN (12,32) AND NOTA >=5)

1. Să se formuleze o interogare care ar afişa lista cursurilor denumirea cărora începe cu „I”.

SELECT \*

FROM CURS

WHERE CURS\_DEN LIKE ‘I%’

1. Să se formuleze o interogare care ar afişa informaţia despre studenţii la care numele începe cu „I” sau „C”.

SELECT \*

FROM STUDENT

WHERE LEFT(NUME, 1) = ‘I’ OR LEFT(NUME, 1) = ’C’

1. Să se formuleze o interogare care ar afişa înregistrările din tabelul NOTE\_EXAM la care lipseşte valoarea pentru cîmpul NOTE.

SELECT \*

FROM NOTE\_EXAM

WHERE NOTA IS NULL

1. Să se formuleze o interogare care ar afişa înregistrările din tabelul NOTE\_EXAM la care este specificată valoarea pentru cîmpul NOTE.

SELECT \*

FROM NOTE\_EXAM

WHERE NOTA IS NOT NULL

1. Să se formuleze o interogare pentru tabela STUDENT astfel încît tabelul rezultat să conţină un cîmp care ar conţine lista valorilor din toate cîmpurile acestei tabele delimitate prin „;”, valorile caracteriale să fie afişate numai cu majuscule, de ex.: 10;Casta;Olga;0;Bălţi;8/12/1981;10.

SELECT CAST (ST\_ID AS NVARCHAR)+

’;’+

UPPER(NUME, 1,1)+

LOWER(SUBSTRING(NUME,2LEN(NUME)))+

’;’+

UPPER(SUBSTRING(PRENUME,1,1))+

LOWER(SUBSTRING(PRENUME, 2,LEN(PRENUME)))+

’;’+

CAST(BURSA AS NVARCHAR)+

’;’+

UPPER(SUBSTRING(ISNULL(ORAS\_S, ‚NU ESTE INDICAT’),1,1))+

LOWER(SUBSTRING(ISNULL(ORAS\_S, ‚NU ESTE INDICAT’), 2,LEN(ISNULL(ORAS\_S, ‚NU ESTE INDICAT’))))+

’;’+

CAST (CONVERT(VARCHAR,ISNULL(DATA\_N, ’ ’),103)AS NVARCHAR)+

’;’+

CAST(NIV\_ID AS NVARCHAR)AS REZULTAT

FROM STUDENT

1. Să se formuleze o interogare care ar afişa informaţia din tabelul STUDENT astfel încît tabelul rezultat să conţină un singur cîmp ce ar conţine informaţie în următorul format O.CASTA; localitatea – BĂLŢI; născut la 8.12.1981.

SELECT SUBSTRING(PRENUME,1,1)+

’;’+

UPPER(NUME)’+; LOCALITATEA- ’+ UPPER(ISNULL(ORAS, ’NU ESTE INDICAT’))+’;NASCUT(A) LA ’+CAST (CONVERT(VARCHAR,ISNULL (DATA\_N, ’ ’),104)AS NVARCHAR)AS REZULTAT

FROM STUDENT

1. Să se formuleze o interogare care ar afişa informaţia din tabelul STUDENT astfel încît tabelul rezultat să conţină un singur cîmp ce ar conţine informaţie în următorul format o.casta; localitatea – bălţi; născut la 8-dec-1981.

SELECT LOWER (SUBSTRING(PRENUME,1,1))+

’;’+

LOWER(NUME)’+; LOCALITATEA- ’+ LOWER(ISNULL(ORAS, ’NU ESTE INDICAT’))+’; NASCUT(A) LA ’ +

CAST (CONVERT(VARCHAR,ISNULL (DATA\_N, ’ ’),106)AS NVARCHAR)

FROM STUDENT

1. Să se formuleze o interogare care ar afişa din tabelul STUDENT astfel încît tabelul rezultat să conţină un singur cîmp ce ar conţine informaţie în următorul format Olga Casta s-a născut în anul 1981.

SELECT UPPER (SUBSTRING(NUME,1,1))+ LOWER (SUBSTRING(NUME, 2, LEN(NUME)))+ ’ ’ +

UPPER (SUBSTRING(PRENUME,1,1))+LOWER(SUBSTRING(PRENUME, 2,LEN(PRENUME)))+

’ S-A NASCUT IN ANUL ’+CAST(YEAR(IS NULL(DATA\_N, ’ ’))AS NVARCHAR) AS RESULTAT

FROM STUDENT

1. La punctul precedent mai apare o condiţie: studenţii trebuie să fie din anul 1, 2 sau 4.

SELECT UPPER (SUBSTRING(NUME,1,1))+ LOWER (SUBSTRING(NUME, 2, LEN(NUME)))+ ’ ’ +

UPPER (SUBSTRING(PRENUME,1,1))+LOWER(SUBSTRING(PRENUME, 2,LEN(PRENUME)))+

’ S-A NASCUT IN ANUL ’+CAST(YEAR(IS NULL(DATA\_N, ’ ’))AS NVARCHAR) AS RESULTAT

FROM STUDENT

WHERE ANUL IN (1, 2, 4)

1. Să se formuleze o interogare care ar afişa numle studenţilor ce au bursă, precum şi bursă mărită de 100 de ori.

SELECT NUME, BURSA\*100  
FROM STUDENT

WHERE BURSA > 0

1. Să se formuleze o interogare care ar afişa pentru tabelul UNIVERSITATE astfel încît tabele rezultat ar conţine un singur cîmp cu următorul format: Cod-10; USB-or.Bălţi;Reiting-296.

SELECT ’COD-’+CAST(UNIV\_ID AS NVARCHAR)+

’;’+

UNIV\_DEN+’-OR.’+UPPER(SUBSTRING(ISNULL(ORAS,’NU ESTE INDICAT’),1,1))+

LOWER(SUBSTRING(ISNULL(ORAS,’NU ESTE INDICAT’),2,LEN(ISNULL(ORAS,’NU ESTE INDICAT’)))+

’;REITING- ’CAST(RATING AS NVARCHAR)) AS REZULTAT

FROM UNIVERSITATE

WHERE UNIV\_ID=10

1. În punctul precedent mai apare o condiţie: valoarea reitingului trebuie aproximată la primul simbol.

SELECT ’COD-’+CAST(UNIV\_ID AS NVARCHAR)+

’;’+

UNIV\_DEN+’-OR.’+UPPER(SUBSTRING(ISNULL(ORAS,’NU ESTE INDICAT’),1,1))+

LOWER(SUBSTRING(ISNULL(ORAS,’NU ESTE INDICAT’),2,LEN(ISNULL(ORAS,’NU ESTE INDICAT’)))+

’;REITING- ’CAST(ROUND(RATING, -2) AS NVARCHAR)) AS REZULTAT

FROM UNIVERSITATE

WHERE UNIV\_ID=10

1. Să se formuleze o interogare care ar determina nota medie a tuturor studenţilor.

SELECT AVG (CAST(NOTA AS FLOAT)) AS AVERAGE

FROM NOTE\_EXAM

1. Să se formuleze o interogare care ar determina despre cîţi studenţi se păstrează informaţia.

SELECT COUNT(\*) AS TOTAL\_ST

FROM STUDENT

1. Să se formuleze o interogare care ar determina nota maximă primită de fiecare student.

SELECT STUDENT.ST\_ID, MAX(NOTA)

FROM STUDENT LEFT JOIN NOTE\_EXAM ON STUDENT.ST\_ID=NOTE\_EXAM.ST\_ID

GROUP BY STUDENT.ST\_ID

1. Să se formuleze o interogare care ar determina numărul de studenţi care au susţinut examen la cursul cu numărul de identificare 20.

SELECT COUNT(DISTINCT ST\_ID)

FROM NOTE\_EXAM

WHERE CURS\_ID=20 AND NOTA>=5

1. Să se înscrie o interogare care ar calcula numărul de cursuri diferite după tabelul NOTE\_EXAM.

SELECT COUNT(DISTINCT CURS\_ID) AS TOTAL\_CURS

FROM NOTE\_EXAM

1. Să se formuleze o interogare care ar afişa identificatorul studentului şi nota lui minimală.

SELECT STUDENT.ST\_ID, MIN(NOTA)

FROM STUDENT LEFT JOIN NOTE\_EXAM ON STUDENT.ST\_ID=NOTE\_EXAM.ST\_ID

GROUP BY STUDENT.ST\_ID

1. Să se formuleze o interogare care ar afişa primul studentul (după ordinea alfabetică) numele căruia începe cu „I”.

SELECT TOP I NUME

FROM STUDENT

WHERE NUME LIKE ’I%’

ORDER BY NAME ASC

1. Să se formuleze o interogare care ar afişa denumirea cursului (pentru fiecare curs) şi numărul maximal al semestrului în care a fost predat acest curs.

SELECT CURS\_DEN, MAX(SEMESTRU) AS TOTAL

FROM CURS

GROUP BY CURS\_DEN, SEMESTRU

1. Să se formuleze o interogare care ar afişa pentru fiecare dată de promovare a examenului numărul de studenţi care au susţinut acest examen.

SELECT DATA\_EXAM, COUNT(DISTINCT ST\_ID) AS STUDENTI

FROM NOTE\_EXAM

WHERE NOTA > 5

GROUP BY DATA\_EXAM

1. Să se formuleze o interogare care ar nota medie pentru fiecare an de studii pentru fiecare curs.

SELECT CURS\_ID

CASE WHEN MONTH(DATA\_EXAM)>=9 THEN YEAR(DATA\_EXEM) ELSE YEAR(DATA\_EXEM)-1END AS YEAR,

AVG(CAST(NOTA AS FLOAT))AS MEDIA

FROM NOTE\_EXAM

GROUP BY CURS\_ID,

CASE WHEN MONTH(DATA\_EXAM)>=9

THEN YEAR(DATA\_EXEM) ELSE YEAR(DATA\_EXEM)-1END

1. Să se formuleze o interogare care ar determina nota medie pentru fiecare student.

SELECT NUME, PRENUME, AVG(CAST(NOTA AS FLOAT)) AS AVERAGE\_ST

FROM NOTE\_EXAM INNER JOIN STUDENT ON NOTE\_EXAM.ST\_ID=STUDENT.ST\_ID

GROUP BY NUME, PRENUME

1. Să se formuleze o interogare care ar determina nota medie pentru fiecare examen.

SELECT EXAM\_ID, AVG(CAST(NOTA AS FLOAT)) AS AVERAGE\_EXAM

FROM NOTE\_EXAM

GROUP BY EXAM\_ID

1. Să se formuleze o interogare care ar determina pentru fiecare examen numărul de studenţi ce l-au susţinut.

SELECT EXAM\_ID, COUNT(ST\_ID) AS NR\_STUDENT

FROM NOTE\_EXAM

WHERE NOTA>=5

GROUP BY EXAM\_ID

1. Să se formuleze o interogare care ar determina pentru fiecare an de studii numărul de cursuri studiate pe parcursul acestui an.

SELECT 1 AS AN, COUNT(CURS\_ID) AS NR\_CURS

FROM CURS

WHERE SEMESTRU IN (1,2)

UNION SELECT 2, COUNT(CURS\_ID)

FROM CURS

WHERE SEMESTRU IN (3,4)

UNION SELECT 3, COUNT(CURS\_ID)

FROM CURS

1. Să presupunem că a fost mărită bursa cu 20%. Să se proiecteze o interogare la tabelul STUDENT, care ar afişa identificatorul studentului, numele lui şi bursa mărită. Datele afişate trebuie ordonate: a) după bursă; b) în ordinea alfabetică a numelui studenţilor.

SELECT ST\_ID, NUME, PRENUME,(STUDENT.BURSA \* 1.2) AS BURSA\_MARITA

FROM STUDENT

ORDER BY BURSA, NUME ASC

1. Să se formuleze o interogare care pentru fiecare student să se determine din tabelul NOTE\_EXAM nota maximală şi nota minimală şi să le afişeze împreună cu identificatorul studentului.

SELECT ST\_ID, MAX(NOTA) AS MAXIMUM, MIN(NOTA) AS MINIMUM

FROM NOTE\_EXAM

GROUP BY ST\_ID

1. Să se formuleze o interogare care ar afişa lista cursurilor studiate afişate în ordinea descreşterii semestrilor şi în ordinea crescătoare a numărului de ore planificte pentru cursul respectiv. În tabelul rezultat cîmpul numărul semestrului trebuie să fie primul, după care denumirea şi identificatorul cursului.

SELECT SEMESTRU, CURS\_DEN, CURS\_ID

FROM CURS

GROUP BY SEMESTRU DESC, ORE ASC

1. Să se formuleze o interogare care ar calcula suma notelor studenţilor pentru fiecare dată de susţinere a examenelor şi prezintă rezultatul în ordinea descrescătoare a acestor sume.

SELECT DATA\_EXAM, SUM(NOTA) AS SUM

FROM NOTE\_EXAM

GROUP BY DATA\_EXAM

ORDER BY SUMA DESC

1. Să se formuleze o interogare care ar afişa valoarea medie, minimală şi maximală a notelor studenţilor pentru fiecare dată de susţinerea a examenului şi prezintă rezultatele în ordinea descrescătoare a acestor valori.

SELECT DATA\_EXAM, MAX(NOTA) AS MAXIM, MIN(NOTA) AS MININ, AVG(CAST(NOTA AS FLOAT)) AS AVERAGE

FROM NOTE\_EXAM

GROUP BY DATA\_EXAM

ORDER BY MEDIA DESC, MINIM DESC, MAXIM DESC

1. Să se proiecteze o interogare cu subinterogare care ar afişa notele primite de cătrestudentul cu numele „Popescu”. Presupunem că nu este cunoscut identificatorul acestui student.

SELECT NOTA

FROM NOTE\_EXAM

WHERE ST\_ID=(SELECT ST\_ID FROM STUDENT WHERE NUME=’POPESCU’)

1. Să se formuleze o interogare care ar afişa numele studenţilor care au primit la cursul cu identificatorul 101 o notă mai mare decît nota medie generală.

SELECT NUME, CURS\_ID, NOTA

FROM NOTE\_EXAM JOIN STUDENT

ON NOTE\_EXAM.ST\_ID=STUDENT.ST\_ID

WHERE NOTA>(SELECT AVG (NOTA)

FROM NOTE\_EXAM)

AND CURS\_ID=101

1. Să se formuleze o interogare care ar afişa numele studenţilor care au primit la cursul cu identificatorul 102 o notă mai mică decît nota medie generală.

SELECT NUME, CURS\_ID, NOTA

FROM NOTE\_EXAM JOIN STUDENT

ON NOTE\_EXAM.ST\_ID=STUDENT.ST\_ID

WHERE NOTA>(SELECT AVG (NOTA)

FROM NOTE\_EXAM) AND CURS\_ID=102

1. Să se formuleze o interogare care ar afişa numărul de cursuri la care a fost examinaţi studenţi ce au susţinut mai mult de 20 de examene.

SELECT COUNT (DISTINCT CURS\_ID) AS CURS

FROM NOTE\_EXAM

WHERE ST\_ID IN(SELECT ST\_ID FROM NOTE\_EXAM WHERE NOTA >=5

GROUP BY ST ID

HAVING COUNT (NOTA)>20)

1. Să se formuleze o interogare care ar afişa numele şi identificatorul studenţilor la care bursa este cea mai mare dintre bursele primite de studenţii din acelaşi oraş.

SELECT NUME,ST\_ID,BURSA,ORAS

FROM(SELECT NUME,ST\_ID,ORAS,BURSA FROM STUDENT) AS T1

INNER JOIN (SELECT ORAS AS ORASS1, MAX(BURSA)

AS BURSA1 FROM STUDENT WHERE ORAS IS NOT NULL

GROUP BY ORAS) AS T2

ON ORAS = ORASS1 AND BURSA = BURSA1

1. Să se formuleze o interogare care ar afişa numele şi identificatorul studenţilor care locuiesc în oraş în care nu este universitate.

SELECT NUME,ST\_ID

FROM STUDENT AS STUD

WHERE ORAS\_S NOT IN(SELECT DISTINCT ORAS FROM UNIVERSITATE)

1. Să se formuleze o interogare care ar afişa numele şi identificatorul studenţilor despre care se ştie că locuiesc într-un alt oraş de universitatea unde învaţă.

SELECT ST\_ID,NUME

FROM STUDENT INNER JOIN UNIVERSITATE

ON STUDENT.UNIV\_ID = UNIVERSITATE.UNIV\_ID

WHERE STUDENT.ORAS != UNIVERSITATE.ORAS

1. Să se formuleze o interogare care ar determina numele studenţilor ce au obţinut cel puţin o notă negativă.

SELECT ST\_ID,NUME

FROM STUDENT

WHERE ST\_ID IN (SELECT DISTINCT ST\_ID

FROM NOTE\_EXAM

WHERE NOTA < 5)

1. Să se formuleze o interogare care ar determina identificatorul cursurilor examenul la are a fost susţinut nu de un student ci de cîţiva.

SELECT CURS\_ID

FROM NOTE\_EXAM

WHERE NOTA >= 5

GROUP BY CURS\_ID

HAVING COUNT (ST\_ID)>1

1. Să se formuleze o interogare care ar afişa identificatorul studenţilor ce au susţinut mi mult de un examen.

SELECT ST\_ID,COUNT(NOTA) AS PASSED

FROM NOTE\_EXAM

WHERE NOTA >= 5

GROUP BY ST\_ID

HAVING COUNT (NOTA) >= 1

1. Să se formuleze o interogare care ar afişa studenţii ce studiază în universitate reitingul căreia este mai mare de 300.

SELECT \*

FROM STUDENT

WHERE UNIV\_ID IN (SELECT UNIV\_ID

FROM UNIVERSITATE

WHERE RATING > 300)

1. Să se formuleze o interogare care ar afişa lista studenţilor care locuiesc în oraş unde este universitate, dar care nu învaţă în aceasă universitate.

SELECT \*

FROM STUDENT

WHERE STUDENT.ORAS IN (SELECT DISTINCT ORAS

FROM UNIVERSITATE)

AND STUDENT.ORAS != (SELECT ORAS

FROM UNIVERSITATE WHERE UNIV\_ID = STUDENT.UNIV\_ID)

1. Să se formuleze o interogare pe baza tabelului CURS care ar afişa lista cursurilor examenele la care au fost susţinute cel puţin de un student.

SELECT \*

FROM CURS

WHERE CURS\_ID IN (SELECT CURS\_ID

FROM NOTE\_EXAM

WHERE NOTA >= 5

GROUP BY CURS\_ID

HAVING COUNT(ST\_ID)>=1 )

1. Să se formuleze o interogare care ar afişa informaţia despre studenţii care învaţă la universitatea din oraşul în care locuiesc.

SELECT \*

FROM STUDENT

WHERE ORAS\_S = (SELECT ORAS

FROM UNIVERSITATE

WHERE UNIV\_ID = STUDENT.UNIV\_ID)

1. Să se formulze o interogare care ar afişa identificatorii studenţilor ce au note care sunt mai mari cel puţin de o notă obţinută de ei primite pe 6 decembrie 1999.

SELECT DISTINCT ST\_ID

FROM NOTE\_EXAM

WHERE NOTA > (SELECT NOTA

FROM NOTE\_EXAM

WHERE DATA\_EXAM = ’06-12-1999’)

1. Să se formuleze o interogare care ar afişa lista universităţilor cu un reiting mai mare decît reitingului oricărei universităţi din Bălţi.

SELECT \*

FROM UNIVERSITATE

WHERE RATING > ALL (SELECT MAX(RATING)

FROM UNIVERSITATE

WHERE ORAS = ‘BALTI’)

1. Să se formulze o interogare care ar selecta informaţia despre universităţile cu un reiting egal sau mai mare ca reitingul USB Bălţi.

SELECT \*

FROM UNIVERSITATE

WHERE (UNIV \_DEN != ‘USB’

OR ORAS != ‘BALTI’)

AND RATING >= (SELECT RATING

FROM UNIVERSITATE

WHERE UNIV\_DEN = ‘USB’

AND ORAS = ‘BALTI’)

1. Să se formulze o interogare care ar selecta informaţia despre studenţii la care în oraşul unde locuies nu este nici o universitate.

SELECT \*

FROM STUDENT

WHERE ORAS NOT IN (SELECT DISTINCT ORAS FROM UNIVERSITATE)

1. Să se formulze o interogare pe baza tabelei NOTE\_EXAM care ar afişa denumirea cursurilor la care valoarea cîmpului NOTE depăşeşte orice valoare a aceluiaşi cîmp pentru cursul cu identificatorul egal cu 105.

SELECT DISTINCT CURS\_ID

FROM NOTE\_EXAM

WHERE NOTA > (SELECT MAX(NOTA)

FROM NOTE\_EXAM

WHERE CURS\_ID = 105)

1. Să se formuleze o interogare care ar afişa numele studentului ce a susţinut examenul precum şi identificatorul cursului la care a susţinut examenul.

SELECT DISTINCT NUME,CURS\_ID

FROM STUDENT INNER JOIN NOTE\_EXAM

ON STUDENT.ST\_ID = NOTE\_EXAM.ST\_ID

WHERE NOTA >= 5

1. Să se formulze o interogare care ar afişa informaţia despre toţi studenţii, iar pentru cei ce au susţinut examen – identificatorul cursului.

SELECT DISTINCT STUDENT. \*,

CASE WHEN NOTA >= 5 THEN CURS\_ID ELSE NULL END AS CURS

FROM STUDENT LEFT JOIN NOTE\_EXAM ON STUDENT.ST\_ID = NOTE\_EXAM.ST\_ID

1. Să se formulze o interogare care ar afişa numele tuturor studenţilor care au încercat să susţină examen, precum şi denumirea cursului.

SELECT NUME, CURS\_DEN

FROM STUDENT INNER FOIN (SELECT DISTINCT ST\_ID,CURS\_ID

FROM NOTE \_EXAM) AS ST ON

STUDENT.ST\_ID INNER JOIN CURS ON ST.CURS\_ID = CURS.CURS\_ID

1. Să se formulze o interogare care pentru fiecare student ar afişa lista cursurilor la care studentul a obţinut note de 9 şi 10.

SELECT ST\_ID, CURS\_ID

FROM NOTE\_EXAM

WHERE NOTA IN (9,10)

ORDER BY ST\_ID

1. Să se formuleze o interogare care ar afişa denumirea cursurilor la studenţii au primit note bune (8,9 şi 10). Tabelul rezultat trebuie să conţină numele studenţilor, denumirea cursului şi nota primită.

SELECT NUME, CURS\_DEN, NOTA

FROM STUDENT INNER JOIN NOTE\_EXAM ON STUDENT.ST\_ID = NOTE\_EXAM.ST\_ID INNER JOIN CURS ON CURS.CURS\_ID = NOTE\_EXAM.CURS\_ID

WHERE NOTA IN (8, 9, 10)

1. Să se formuleze o interogare care ar afişa lista universităţilor cu un reiting mai mare de 300, împreună cu valoarea maximală a bursei primită de studenţii acestor universităţi.

SELECT RATING.UNIV\_ID, BURSA.MAX\_BURSA

FROM (SELECT UNIV\_ID FROM UNIVERSITATE WHERE RATING > 300) AS RATING LEFT JOIN(SELECT UNIV\_ID, MAX(BURSA) AS MAX\_BURSA

FROM STUDENT

GROUP BY UNIV\_ID) AS BURSA ON RATING.UNIV\_ID = BURSA.UNIV\_ID

1. Să se formulze o interogare pentru afişarea listei studenţilor (în ordinea alfabetică) împreună cu reitingul universităţii unde învaţă, incluzînd în listă şi acei studenţi care în baza de date nu este indicată universitatea unde îşi face studiile.

SELECT NUME,PRENUME, RATING

FROM STUDENT LEFT JOIN UNIVERSITATE ON STUDENT.UNIV\_ID = UNIVERSITATE.UNIV\_ID

ORDER BY NUME, PRENUME

1. Să se formuleze o interogre care ar afişa numele studenţilor cu acelaşi prenume.

SELECT S1.NUME, S1.PRENUME

FROM STUDENT AS S1 INNER JOIN (SELECT NUME, PRENUME FROM STUDENT) AS S2 ON S1.PRENUME = S2.PRENUME

WHERE S1 PRENUME = S2.PRENUME GROUP BY S1.PRENUME, S1.NUME

HAVING COUNT(S1.ST\_ID) > 1

ORDER BY S1.PRENUME

1. Să se formulze o interogare care ar afişa toate perechile de nume a studenţilor care locuiesc în acelaşi oraş, cu excepţia combinaţiei numelui studenţilor cu ei înseşi (Popescu – Popescu) şi a combinaţiilor ce diferă după ordinea numelor (Popescu – Ionescu sau Ionescu – Popescu).

SELECT DISTINCT S1.NUME, S2.NUME

FROM STUDENT AS S1 CROSS JOIN (SELECT NUME, ORAS FROM STUDENT) AS S2

WHERE S1.ORAS = S2.ORAS AND S1.NUME != S2.NUME

1. Să se formulze o interogare care ar afişa lista tuturor perechilor de denumiri de univeristăţi care se află în acelaşi oraş.

SELECT DISTINCT S1.UNIV\_DEN, S2.UNIV\_DEN

FROM UNIVERSITATE AS S1 CROSS JOIN (SELECT UNIV\_DEN, ORAS FROM UNIVERSITATE) AS S2

WHERE S1.ORAS = S2.ORAS AND S1.UNIV\_DEN != S2.UNIV\_DEN

1. Să se formulze o interogare care permite vizualizarea denumirii universităţilor şi oraşelor unde se află aceste universităţi pentru care reitingul este egal sau depăşeşte reitingul univesităţii din Bălţi.

SELECT UNIV\_DEN, ORAS

FROM UNIVERSITATE

WHERE RATING >= (SELECT MIN(RATING) FROM UNIVERSITATE WHERE ORAS = ‘BALTI’)

1. Să înscrie instrucţiunea care ar adăuga în tabela CURS o nouă înregistrare pentru un curs nou cu următoarele valori pentru cîmpuri: semestrul – 4; denumirea – Algebra; numărul de ore – 72; identificatorul – 201.

INSERT CURS(CURS\_ID, CURS\_DEN, ORE, SEMESTRU)

VALUES (201, ‘ALGEBRA’, 72, 4)

1. Să se adauge o înregistrare pentru un student nou pe care îl cheamă Osman Nicolai, care îşi face studiile la USB în anul I de studii şi care locuieşte în oraşul Bălţi, iar date despre data de naştere şi bursă nu se cunosc.

INSERT STUDENT (ST\_ID, NUME, PRENUME, BURSA, ORAS, DATA\_N, UNIV\_ID, ANUL)

VALUES (123, ‘OSMAN’, ‘NICOLAI’, NULL, ‘BALTI’, NULL, 10, 1)

1. Să se înscrie o instrucţiune pentru a elimina din tabela NOTE\_EXAM toate înregistrările în care se păstrează notele studentului cu numărul de identificare egal cu 100.

DELETE NOTE\_EXAM WHERE ST\_ID = 100

1. Să se înscrie o instrucţiune care ar mari reitingul universităţilor din Cahul cu 5.

UPDATE UNIVERSITATE SET RATING = RATING +5

WHERE (ORAS = ‘CAHUL’)

1. Să se înscrie o instrucţiune care ar fi necesară în caz că Popescu se mută cu traiul în Bălţi.

UPDATE STUDENT SET ORAS = ‘BALTI’ WHERE NUME = ‘POPESCU’

1. Să se înscrie o instruţiune care ar adăuga în tabelul STUDENTI (tabelul STUDENTI are aceeaşi structură ca şi STUDENT) datele despre toţi studenţii ce învaţă la Chişinău.

SELECT \* INTO STUDENT

FROM STUDENT

WHERE ORAS = ‘CHISINAU’

1. Să se înscrie o instruţiune care ar adăuga în tabelul STUDENTI (tabelul STUDENTI are aceeaşi structură ca şi STUDENT) datele despre toţi studenţii ce au susţinut cu succes mai mult de 5 examene.

INSERT INTO STUDENT SELECT \*

FROM STUDENT

WHERE ST\_ID IN (SELECT ST\_ID

FROM NOTE\_EXAM

WHERE NOTA >= 5

GROUP BY ST\_ID

HAVING GOUNT(\*) > 5)

1. Să se înscrie o instruţiune care ar elimina din tabelul CURS acele cursuri la care nici un student nu a obţinut nici o notă.

DELETE FROM CURS

WHERE CURS\_ID NOT IN (SELECT DISTINCT CURS\_ID

FROM NOTE\_EXAM

WHERE NOTA IS NOT NULL)

1. Să se înscrie o instruţiune care ar permite mărirea bursei cu 20% pentru acei studenţi care au o sumă a notelor ce depăşeşte 50 puncte.

UPDATE STUDENT SET BURSA = \* 1.2 WHERE ST\_ID IN (SELECT ST\_ID

FROM NOTE\_EXAM

GROUP BY ST\_ID

HAVING SUM(NOTA) >50)

1. Să se înscrie o instruţiune CREATE TABLE pentru crearea tabelului PROFESOR.

CREATE TABLE PROFESSOR (PR\_ID SMALLINT PRIMARY KEY, NUME NVARCHAR(10) NOT NULL,

PRENUME NVARCHAR(10) NOT NULL,

ORAS NVARCHAR(10),

UNIV\_ID SMALLINT FOREIGN KEY (UNIV\_ID) REFERENCES

UNIVERSITATEA(UNIV\_ID))

1. Să se înscrie o instruţiune CREATE TABLE pentru crearea tabelului CURS.

CREATE TABLE CURS (CURS\_ID SMALINT PRIMARY KEY,

CURS\_DEN NVARCHAR(20),

ORE INT,

SEMESTRU INT)

1. Să se înscrie o instruţiune CREATE TABLE pentru crearea tabelului UNIVERSITATE.

CREATE TABLE UNIVERSITATE ( UNIV\_ID INT PRIMARY KEY,

UNIV\_DEN NVARCHAR(20),

RATING INT,

ORAS NVARCHAR(20))

1. Să se înscrie o instruţiune CREATE TABLE pentru crearea tabelului NOTE\_EXAM.

CREATE TABLE NOTE\_EXAM(EXAM\_ID SMALLINT PRIMARY KEY,

ST\_ID SMALLINT FOREIGN KEY (ST\_ID) REFERNCES STUDENT(ST\_ID),

CURS\_ID SMALLINT FOREIGN KEY (CURS\_ID) REFERENCES CURS(CURS\_ID),

NOTA SMALLINT,

DATA\_EXAM DATE DEFAULT GETDATE())

1. Să se înscrie o instruţiune CREATE TABLE pentru crearea tabelului CURS\_PROF.

CREATE TABLE CURS\_PROF(PR\_ID SMALLINT FOREIGN KEY (PR\_ID) REFERENCES PROFESOR(PR\_ID),CURS\_ID SMALLINT FORIGN KEY (CURS\_ID) REFERENCES CURS (CURS\_ID))

1. Să se înscrie o instruţiune care ar mări viteza de selectare a datelor despre studenţi după anul de studii.

SELECT \*

FROM STUDENT

ORDER BY ANUL, NUME, PRENUME

1. Să se definească un index care ar permite pentru fiecare student mai repede de regăsit notele obţinute de el grupate după data examinării.

SELECT ST\_ID, NOTA, DATA \_ EXAM

FROM NOTE\_EXAM

WHERE ST\_ID IN (SELECT ST\_ID

FROM STUDENT )

GROUP BY DATA \_ EXAM