1. SELECT \* FROM STUDENT
2. SELECT DISTINCT ORAS FROM STUDENT WHERE ORAS IS NOT NULL;
3. SELECT NUME, PRENUME FROM STUDENT WHERE NUME = 'Petrov';
4. SELECT \* FROM STUDENT WHERE ANUL = 3 AND BURSA > 0;
5. SELECT CURS\_ID, CURS\_DEN, SEMESTRU, ORE FROM CURS;
6. SELECT \* FROM NOTE\_EXAM WHERE CURS\_ID = 12;
7. SELECT ANUL, PRENUME, NUME, BURSA FROM STUDENT;
8. SELECT CURS\_DEN, ORE FROM CURS WHERE SEMESTRU = 4;
9. SELECT DISTINCT NOTA FROM NOTE\_EXAM WHERE NOTA IS NOT NULL;
10. SELECT NUME FROM STUDENT WHERE ANUL >= 3
11. SELECT NUME, PRENUME, ANUL FROM STUDENT WHERE BURSA > 140;
12. SELECT \* FROM CURS WHERE ORE > 30;
13. SELECT \* FROM UNIVERSITATE WHERE RATING > 300;
14. SELECT NUME, PRENUME, ANUL FROM STUDENT WHERE BURSA >= 100 AND ORAS = 'Balti';
15. SELECT STUDENT.NUME, NOTA FROM STUDENT INNER JOIN NOTE\_EXAM ON STUDENT.ST\_ID = NOTE\_EXAM.ST\_ID WHERE NOTA IN (9, 10);
16. SELECT \* FROM CURS WHERE ORE > 30 AND ORE < 40;
17. SELECT \* FROM STUDENT WHERE NUME LIKE 'P%';
18. SELECT CURS\_ID FROM NOTE\_EXAM WHERE DATA\_EXAM BETWEEN '10-01-1999' AND '20-01-1999';
19. SELECT \* FROM CURS WHERE CURS\_ID IN (SELECT CURS\_ID FROM NOTE\_EXAM WHERE ST\_ID IN (12, 32) AND NOTA >= 5);
20. SELECT \* FROM CURS WHERE CURS\_DEN LIKE 'I%';
21. SELECT \* FROM STUDENT WHERE NUME LIKE 'I%' OR NUME LIKE 'C%';
22. SELECT \* FROM NOTE\_EXAM WHERE NOTA IS NULL;
23. SELECT \* FROM NOTE\_EXAM WHERE NOTA IS NOT NULL;
24. /\*SELECT
25. CAST(ST\_ID AS varchar) + '; ' +
26. UPPER(NUME) + '; '+ UPPER(PRENUME) + '; '+
27. CAST(BURSA AS varchar) + '; '+
28. CAST(ANUL AS varchar) + '; '+
29. CASE WHEN ORAS IS NULL THEN 'ЗАСЕКРЕЧЕНО' ELSE UPPER(ORAS) END + '; '+
30. CASE WHEN DATA\_N IS NULL THEN 'ЗАСЕКРЕЧЕНО' ELSE CAST(DATA\_N AS varchar) END + '; '+
31. CAST(UNIV\_ID AS varchar)
32. FROM STUDENT\*/
33. /\*SELECT
34. LEFT(PRENUME, 1) + '.' + NUME + '; ' +
35. CASE WHEN ORAS IS NULL THEN '' ELSE 'localitatea – ' + UPPER(ORAS) + '; ' END +
36. CASE WHEN DATA\_N IS NULL THEN '' ELSE 'născut la ' + CAST(DATA\_N AS varchar) END
37. FROM STUDENT\*/
38. /\*SELECT
39. LOWER(LEFT(PRENUME, 1)) + '.' + LOWER(NUME) + '; ' +
40. CASE WHEN ORAS IS NULL THEN '' ELSE 'localitatea – ' + LOWER(ORAS) + '; ' END +
41. CASE WHEN DATA\_N IS NULL THEN '' ELSE 'născut la ' + CONVERT(varchar, DATA\_N, 106) END
42. FROM STUDENT\*/
43. /\*SELECT
44. PRENUME + ' ' + NUME + ' s-a născut în anul ' +
45. CASE WHEN DATA\_N IS NULL THEN 'ЗАСЕКРЕЧЕНО' ELSE LEFT(DATA\_N, 4) END
46. FROM STUDENT;\*/
47. /\*SELECT
48. PRENUME + ' ' + NUME + ' s-a născut în anul ' +
49. CASE WHEN DATA\_N IS NULL THEN 'ЗАСЕКРЕЧЕНО' ELSE LEFT(DATA\_N, 4) END
50. FROM STUDENT
51. WHERE ANUL IN (1, 2, 4);\*/
52. SELECT NUME, BURSA \* 100 AS BURSA FROM STUDENT WHERE BURSA > 0;
53. SELECT 'Cod-' + CAST(UNIV\_ID AS varchar) + '; ' + UNIV\_DEN + '-or.' + ORAS + '; Reiting-' + CAST(RATING AS varchar) FROM UNIVERSITATE;
54. SELECT 'Cod-' + CAST(UNIV\_ID AS varchar) + '; ' + UNIV\_DEN + '-or.' + ORAS + '; Reiting-' + CAST(ROUND(RATING, -2) AS varchar) FROM UNIVERSITATE;
55. SELECT AVG(CAST(NOTA AS float)) FROM NOTE\_EXAM
56. SELECT COUNT(\*) FROM STUDENT
57. 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;

SELECT COUNT(DISTINCT ST\_ID) FROM NOTE\_EXAM WHERE CURS\_ID = 20 AND NOTA >= 5;

1. SELECT COUNT(DISTINCT CURS\_ID) FROM NOTE\_EXAM
2. 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;
3. SELECT \* FROM STUDENT WHERE NUME LIKE 'I%' ORDER BY NUME ASC;
4. SELECT CURS\_DEN, MAX(SEMESTRU) FROM CURS GROUP BY CURS\_ID, CURS\_DEN;
5. SELECT DATA\_EXAM, COUNT(DISTINCT ST\_ID) FROM NOTE\_EXAM WHERE NOTA >= 5 GROUP BY DATA\_EXAM;
6. /\*SELECT
7. CURS\_ID,
8. CASE WHEN MONTH(DATA\_EXAM) >= 9 THEN YEAR(DATA\_EXAM) ELSE YEAR(DATA\_EXAM) - 1 END AS an\_de\_studii,
9. AVG(CAST(NOTA AS float)) AS AVG\_NOTA
10. FROM NOTE\_EXAM
11. GROUP BY
12. CURS\_ID,
13. CASE WHEN MONTH(DATA\_EXAM) >= 9 THEN YEAR(DATA\_EXAM) ELSE YEAR(DATA\_EXAM) - 1 END;\*/
14. SELECT ST\_ID, AVG(CAST(NOTA AS float)) FROM NOTE\_EXAM GROUP BY ST\_ID;
15. SELECT EXAM\_ID, AVG(CAST(NOTA AS float)) FROM NOTE\_EXAM GROUP BY EXAM\_ID;
16. SELECT EXAM\_ID, COUNT(ST\_ID) FROM NOTE\_EXAM WHERE NOTA >= 5 GROUP BY EXAM\_ID;
17. /\*SELECT
18. CASE WHEN MONTH(DATA\_EXAM) >= 9 THEN YEAR(DATA\_EXAM) ELSE YEAR(DATA\_EXAM) - 1 END AS an\_de\_studii,
19. COUNT(DISTINCT CURS\_ID) AS COUNT\_CURS
20. FROM NOTE\_EXAM
21. GROUP BY
22. CASE WHEN MONTH(DATA\_EXAM) >= 9 THEN YEAR(DATA\_EXAM) ELSE YEAR(DATA\_EXAM) - 1 END;\*/
23. SELECT ST\_ID, NUME, BURSA \* 1.2 AS bursa\_mărită FROM STUDENT ORDER BY BURSA ASC, NUME ASC;
24. SELECT ST\_ID, MIN(NOTA) AS MIN, MAX(NOTA) AS MAX FROM NOTE\_EXAM GROUP BY ST\_ID;
25. SELECT SEMESTRU, CURS\_DEN, CURS\_ID FROM CURS ORDER BY SEMESTRU DESC, ORE ASC;
26. SELECT DATA\_EXAM, SUM(NOTA) AS SUM\_NOTA FROM NOTE\_EXAM GROUP BY DATA\_EXAM ORDER BY SUM\_NOTA DESC;
27. SELECT DATA\_EXAM, AVG(CAST(NOTA AS float)) AS AVG, MIN(NOTA) AS MIN, MAX(NOTA) AS MAX FROM NOTE\_EXAM GROUP BY DATA\_EXAM ORDER BY AVG DESC, MIN DESC, MAX DESC;
28. SELECT NOTA FROM NOTE\_EXAM WHERE ST\_ID = (SELECT ST\_ID FROM STUDENT WHERE NUME = 'Popescu');
29. /\*SELECT NUME FROM STUDENT WHERE ST\_ID IN
30. (SELECT ST\_ID FROM NOTE\_EXAM WHERE CURS\_ID = 101 AND NOTA >
31. (SELECT AVG(CAST(NOTA AS float)) FROM NOTE\_EXAM WHERE CURS\_ID = 101));\*/
32. /\*SELECT NUME FROM STUDENT WHERE ST\_ID IN
33. (SELECT ST\_ID FROM NOTE\_EXAM WHERE CURS\_ID = 102 AND NOTA <
34. (SELECT AVG(CAST(NOTA AS float)) FROM NOTE\_EXAM WHERE CURS\_ID = 102));\*/
35. SELECT COUNT(DISTINCT CURS\_ID) FROM NOTE\_EXAM WHERE ST\_ID IN (SELECT ST\_ID FROM NOTE\_EXAM WHERE NOTA >= 5 GROUP BY ST\_ID HAVING COUNT(NOTA) > 20);
36. /\*SELECT NUME, ST\_ID FROM
37. (SELECT NUME, ST\_ID, ORAS, BURSA FROM STUDENT) AS T1 INNER JOIN
38. (SELECT ORAS AS ORAS1, MAX(BURSA) AS BURSA1 FROM STUDENT WHERE ORAS IS NOT NULL GROUP BY ORAS) AS T2
39. ON ORAS = ORAS1 AND BURSA = BURSA1;\*/
40. SELECT NUME, ST\_ID FROM STUDENT WHERE ORAS NOT IN (SELECT DISTINCT ORAS FROM UNIVERSITATE);
41. SELECT NUME, ST\_ID FROM STUDENT WHERE ORAS != (SELECT ORAS FROM UNIVERSITATE WHERE UNIVERSITATE.UNIV\_ID = STUDENT.UNIV\_ID);
42. SELECT NUME FROM STUDENT WHERE ST\_ID IN (SELECT DISTINCT ST\_ID FROM NOTE\_EXAM WHERE NOTA < 5);
43. SELECT DISTINCT CURS\_ID FROM NOTE\_EXAM WHERE NOTA >= 5 GROUP BY EXAM\_ID, CURS\_ID HAVING COUNT(ST\_ID) > 1;
44. SELECT ST\_ID FROM NOTE\_EXAM WHERE NOTA >= 5 GROUP BY ST\_ID HAVING COUNT(\*) > 1;
45. SELECT \* FROM STUDENT WHERE UNIV\_ID IN (SELECT UNIV\_ID FROM UNIVERSITATE WHERE RATING > 300);
46. SELECT \* FROM STUDENT WHERE ORAS IN (SELECT DISTINCT ORAS FROM UNIVERSITATE) AND ORAS != (SELECT ORAS FROM UNIVERSITATE WHERE UNIV\_ID = STUDENT.UNIV\_ID);
47. SELECT \* FROM CURS WHERE CURS\_ID IN (SELECT CURS\_ID FROM NOTE\_EXAM WHERE NOTA >= 5 GROUP BY CURS\_ID HAVING COUNT(ST\_ID) > 0);
48. SELECT \* FROM STUDENT WHERE ORAS = (SELECT ORAS FROM UNIVERSITATE WHERE UNIV\_ID = STUDENT.UNIV\_ID);
49. SELECT DISTINCT ST\_ID FROM NOTE\_EXAM WHERE NOTA > (SELECT MIN(NOTA) FROM NOTE\_EXAM WHERE DATA\_EXAM = '06-12-1999');
50. SELECT \* FROM UNIVERSITATE WHERE RATING > (SELECT MAX(RATING) FROM UNIVERSITATE WHERE ORAS = 'Balti');
51. SELECT \* FROM UNIVERSITATE WHERE (UNIV\_DEN != 'USB' OR ORAS != 'Balti') AND RATING >= (SELECT RATING FROM UNIVERSITATE WHERE UNIV\_DEN = 'USB' AND ORAS = 'Balti');
52. SELECT \* FROM STUDENT WHERE ORAS NOT IN (SELECT DISTINCT ORAS FROM UNIVERSITATE);
53. SELECT DISTINCT CURS\_ID FROM NOTE\_EXAM WHERE NOTA > (SELECT MAX(NOTA) FROM NOTE\_EXAM WHERE CURS\_ID = 105);
54. SELECT DISTINCT NUME, CURS\_ID FROM STUDENT INNER JOIN NOTE\_EXAM ON STUDENT.ST\_ID = NOTE\_EXAM.ST\_ID WHERE NOTA >= 5;
55. SELECT DISTINCT STUDENT.\*, CASE WHEN NOTA >= 5 THEN CURS\_ID ELSE NULL END AS CURS\_ID FROM STUDENT LEFT JOIN NOTE\_EXAM ON STUDENT.ST\_ID = NOTE\_EXAM.ST\_ID;
56. SELECT NUME, CURS\_DEN FROM STUDENT INNER JOIN (SELECT DISTINCT ST\_ID, CURS\_ID FROM NOTE\_EXAM) AS T1 ON STUDENT.ST\_ID = T1.ST\_ID INNER JOIN CURS ON T1.CURS\_ID = CURS.CURS\_ID;
57. SELECT ST\_ID, CURS\_ID FROM NOTE\_EXAM WHERE NOTA IN (9, 10) ORDER BY ST\_ID;
58. 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);
59. SELECT T1.UNIV\_ID, T2.MAX\_BURSA FROM (SELECT UNIV\_ID FROM UNIVERSITATE WHERE RATING > 300) AS T1 LEFT JOIN (SELECT UNIV\_ID, MAX(BURSA) AS MAX\_BURSA FROM STUDENT GROUP BY UNIV\_ID) AS T2 ON T1.UNIV\_ID = T2.UNIV\_ID;
60. SELECT NUME, PRENUME, RATING FROM STUDENT LEFT JOIN UNIVERSITATE ON STUDENT.UNIV\_ID = UNIVERSITATE.UNIV\_ID ORDER BY NUME, PRENUME;
61. SELECT T1.PRENUME, T1.NUME FROM STUDENT AS T1 INNER JOIN STUDENT AS T2 ON T1.PRENUME = T2.PRENUME AND T1.ST\_ID != T2.ST\_ID;
62. 78 и 79) сделаны не полностью (не знаю)
63. SELECT T1.ORAS, T1.NUME AS NUME1, T2.NUME AS NUME2 FROM STUDENT AS T1 INNER JOIN STUDENT AS T2 ON T1.ORAS = T2.ORAS AND T1.NUME != T2.NUME
64. SELECT T1.ORAS, T1.UNIV\_DEN AS DEN1, T2.UNIV\_DEN AS DEN2 FROM UNIVERSITATE AS T1 INNER JOIN UNIVERSITATE AS T2 ON T1.ORAS = T2.ORAS AND T1.UNIV\_DEN != T2.UNIV\_DEN ORDER BY ORAS;
65. SELECT UNIV\_DEN, ORAS FROM UNIVERSITATE WHERE RATING >= (SELECT MIN(RATING) FROM UNIVERSITATE WHERE ORAS = 'Balti');
66. INSERT INTO CURS VALUES (201, 'Algebra', 72, 4);
67. INSERT INTO STUDENT VALUES (100, 'Osman', 'Nicolai', NULL, 1, 'Bălţi', NULL, 10);
68. DELETE FROM NOTE\_EXAM WHERE ST\_ID = 100;
69. UPDATE UNIVERSITATE SET RATING = RATING + 5 WHERE ORAS = 'Cahul';
70. UPDATE STUDENT SET ORAS = 'Bălţi' WHERE NUME = 'Popescu';
71. INSERT INTO STUDENTI SELECT \* FROM STUDENT WHERE ORAS = 'Chişinău';
72. INSERT INTO STUDENTI SELECT \* FROM STUDENT WHERE ST\_ID IN (SELECT ST\_ID FROM NOTE\_EXAM WHERE NOTA >= 5 GROUP BY ST\_ID HAVING COUNT(\*) > 5);
73. DELETE FROM CURS WHERE CURS\_ID NOT IN (SELECT DISTINCT CURS\_ID FROM NOTE\_EXAM WHERE NOTA IS NOT NULL);
74. UPDATE STUDENT SET BURSA = BURSA \* 1.2 WHERE ST\_ID IN (SELECT ST\_ID FROM NOTE\_EXAM GROUP BY ST\_ID HAVING SUM(NOTA) > 50);
75. /\*CREATE TABLE PROFESOR (
76. PR\_ID INT NOT NULL PRIMARY KEY,
77. NUME VARCHAR(20),
78. PRENUME VARCHAR(20),
79. ORAS VARCHAR(20),
80. UNIV\_ID INT REFERENCES UNIVERSITATE
81. );\*/
82. /\*CREATE TABLE CURS (
83. CURS\_ID INT NOT NULL PRIMARY KEY,
84. CURS\_DEN VARCHAR(20),
85. ORE INT,
86. SEMESTRU INT
87. );\*/
88. /\*CREATE TABLE UNIVERSITATE (
89. UNIV\_ID INT NOT NULL PRIMARY KEY,
90. UNIV\_DEN VARCHAR(20),
91. RATING INT,
92. ORAS VARCHAR(20)
93. );\*/
94. /\*CREATE TABLE NOTE\_EXAM (
95. EXAM\_ID INT,
96. ST\_ID INT REFERENCES STUDENT,
97. CURS\_ID INT REFERENCES CURS,
98. NOTA INT,
99. DATA\_EXAM DATE,
100. PRIMARY KEY (EXAM\_ID, ST\_ID)
101. );\*/
102. /\*CREATE TABLE CURS\_PROF (
103. PR\_ID INT REFERENCES PROFESOR,
104. CURS\_ID INT REFERENCES CURS
105. );\*/
106. SELECT \* FROM STUDENT ORDER BY ANUL, NUME, PRENUME
107. SELECT DATA\_EXAM, NUME, PRENUME, NOTE\_EXAM.ST\_ID, NOTA FROM NOTE\_EXAM INNER JOIN STUDENT ON NOTE\_EXAM.ST\_ID = STUDENT.ST\_ID ORDER BY DATA\_EXAM, NUME, PRENUME