Based on Apna College SQL Notes (Your PDF)

SQL (notes) (1)

TOPIC 1 — DATABASE QUERIES

1. Write a query to create a new database named collegeDB.
2. Create a database only if it does not already exist.
3. Delete a database only if it exists.-where it there we use drop database database_name;
4. Display all databases present in MySQL.
5. Select and use the database collegeDB.

TOPIC 2 — CREATE TABLE

6. Create a table named student with columns (rollno, name, marks, grade, city).
7. Create a table named course with columns (student_id, course_name).
8. Display all tables inside the selected database.

TOPIC 3 — INSERT STATEMENTS

9. Insert 6 rows into the student table.
10. Insert multiple records into the student table using one INSERT statement.
11. Insert a new course entry for any student.
12. Insert a record where grade is NULL.
13. Insert values into only selected columns of student table.

TOPIC 4 — SELECT BASIC

14. Display all students from the student table.
15. Display only the name and marks of all students.
16. Display unique cities from the student table.
17. Display rollno and name for all students.

TOPIC 5 — WHERE CLAUSE

18. Display students who scored more than 85 marks.

19. Display students from the city “Delhi”.

20. Display students whose marks lie between 70 and 90.

21. Display students whose name starts with the letter 'a'.

22. Display students whose grade is NOT ‘A’.

23. Display students whose city is either 'Pune' or 'Mumbai'.

24. Display students whose marks are less than 50.

TOPIC 6 — OPERATORS (AND, OR, NOT, IN, LIKE, BETWEEN)

25. Display students who are from 'Delhi' AND have marks above 80.

26. Display students who are from 'Delhi' OR from 'Mumbai'.

27. Display students whose grade is NOT 'F'.

28. Display students whose marks are NOT between 70 and 90.

29. Display students whose city is IN ('Delhi', 'Pune').

30. Display students whose name ends with letter 'h'.

TOPIC 7 — ORDER BY

31. Display students sorted by marks in descending order.

32. Display students sorted by name in ascending order.

33. Sort students by city and then by marks.

34. Sort students by grade (A → F).

TOPIC 8 — LIMIT

35. Display the top 3 highest scoring students.

36. Display the bottom 2 students with lowest marks.

37. Display the first 4 students from the table.

TOPIC 9 — AGGREGATE FUNCTIONS

38. Find the total number of students.

39. Find the highest marks scored by any student.

40. Find the lowest marks scored.

41. Find the average marks of students.

42. Find the total sum of all marks.

43. Count how many students are from ‘Delhi’.

TOPIC 10 — GROUP BY

44. Count the number of students in each city.

45. Find the average marks of students in each city.

46. Count the number of students in each grade.

47. Display total marks grouped by grade.

TOPIC 11 — HAVING

48. Display cities where average marks are greater than 80.

49. Display grades that have more than 2 students.

50. Display cities that have at least 2 students.

TOPIC 12 — UPDATE

51. Update marks of the student with rollno = 101 to 90.

52. Update the grade of all students scoring above 90 to 'A'.

53. Change the city of student rollno = 103 to ‘Pune’.

54. Update marks of all students from ‘Delhi’ by adding +5.

TOPIC 13 — DELETE

55. Delete all students whose marks are less than 30.

56. Delete students whose grade is ‘F’.

57. Delete student with rollno = 105.

58. Delete all records from student table.

TOPIC 14 — ALTER TABLE

59. Add a new column email to the student table.

60. Delete the column grade from the student table.

61. Change datatype of marks to FLOAT.

62. Rename table student to learners.

TOPIC 15 — JOINS

63. Perform INNER JOIN on student and course tables.

64. Perform LEFT JOIN on student and course tables.

65. Perform RIGHT JOIN on student and course tables.

66. Perform FULL JOIN using UNION.

67. Display left exclusive join (students not having a course).

68. Display right exclusive join (courses not having a student).

TOPIC 16 — SELF JOIN

69. Perform a self join to show students from the same city.

70. Use self join to find duplicate city entries.

TOPIC 17 — UNION

71. Combine names of students from two sections using UNION.

72. Combine Delhi students from two tables using UNION ALL.

TOPIC 18 — SUBQUERIES

73. Display students who scored above class average.

74. Display students whose marks are greater than marks of rollno 103.

75. Display names of students having even roll numbers using subquery.

76. Find the highest marks of students from Mumbai using subquery.

TOPIC 19 — VIEWS

77. Create a view showing name, marks, city.

78. Create a view for only 'Delhi' students.

79. Display all data from the created view.

80. Delete a view from the database.

TOPIC 20 — KEYS & CONSTRAINTS

81. Create a table with PRIMARY KEY constraint.

82. Create a table with UNIQUE constraint on email.

83. Add FOREIGN KEY student_id referencing student(rollno).

84. Add CHECK constraint to ensure marks > 0.

TOPIC 21 — TRUNCATE

85. Truncate the student table to delete all records but keep structure.