**SQL DATABASE**

1. CREATE DATABASE *databasename*;
2. DROP DATABASE *databasename*;
3. CREATE TABLE *table\_name*(  
   *column1 datatype*,  
   *column2 datatype*,  
   *column3 datatype*,  
      ....  
   );
4. CREATE TABLE *new\_table\_name* AS  
       SELECT *column1, column2,...*  
       FROM *existing\_table\_name*  
       WHERE ....;
5. DROP TABLE *table\_name*;
6. TRUNCATE TABLE *table\_name*;
7. ALTER TABLE *table\_name*  
   ADD *column\_name datatype*;
8. ALTER TABLE *table\_name*  
   DROP COLUMN *column\_name*;
9. ALTER TABLE *table\_name*  
   ALTER COLUMN *column\_name datatype*;
10. CREATE TABLE *table\_name*(  
    *column1 datatype* *constraint*,  
    *column2 datatype* *constraint*,  
    *column3 datatype* *constraint*,  
        ....  
    );
11. CREATE TABLE Persons (  
        ID int NOT NULL,  
        LastName varchar(255) NOT NULL,  
        FirstName varchar(255) NOT NULL,  
        Age int  
    );
12. CREATE TABLE Persons (  
        ID int NOT NULL,  
        LastName varchar(255) NOT NULL,  
        FirstName varchar(255),  
        Age int,  
        UNIQUE (ID)  
    );
13. CREATE TABLE Persons (  
        ID int NOT NULL,  
        LastName varchar(255) NOT NULL,  
        FirstName varchar(255),  
        Age int,  
        CONSTRAINT UC\_Person UNIQUE (ID,LastName)  
    );
14. ALTER TABLE Persons  
    ADD UNIQUE (ID);
15. ALTER TABLE Persons  
    ADD CONSTRAINT UC\_Person UNIQUE (ID,LastName);
16. ALTER TABLE Persons  
    DROP INDEX UC\_Person;
17. CREATE TABLE Persons (  
        ID int NOT NULL,  
        LastName varchar(255) NOT NULL,  
        FirstName varchar(255),  
        Age int,  
        PRIMARY KEY (ID)  
    );
18. CREATE TABLE Persons (  
        ID int NOT NULL,  
        LastName varchar(255) NOT NULL,  
        FirstName varchar(255),  
        Age int,  
        CONSTRAINT PK\_Person PRIMARY KEY (ID,LastName)  
    );
19. ALTER TABLE Persons  
    ADD PRIMARY KEY (ID);
20. ALTER TABLE Persons  
    ADD CONSTRAINT PK\_Person PRIMARY KEY (ID,LastName);
21. ALTER TABLE Persons  
    DROP PRIMARY KEY;
22. CREATE TABLE Orders (  
        OrderID int NOT NULL,  
        OrderNumber int NOT NULL,  
        PersonID int,  
        PRIMARY KEY (OrderID),  
        FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)  
    );
23. CREATE TABLE Orders (  
        OrderID int NOT NULL,  
        OrderNumber int NOT NULL,  
        PersonID int,  
        PRIMARY KEY (OrderID),  
        CONSTRAINT FK\_PersonOrder FOREIGN KEY (PersonID)  
        REFERENCES Persons(PersonID)  
    );
24. ALTER TABLE Orders  
    ADD FOREIGN KEY (PersonID) REFERENCES Persons(PersonID);
25. ALTER TABLE Orders  
    ADD CONSTRAINT FK\_PersonOrder  
    FOREIGN KEY (PersonID) REFERENCES Persons(PersonID);
26. ALTER TABLE Orders  
    DROP FOREIGN KEY FK\_PersonOrder;
27. CREATE TABLE Persons (  
        ID int NOT NULL,  
        LastName varchar(255) NOT NULL,  
        FirstName varchar(255),  
        Age int,  
        CHECK (Age>=18)  
    );
28. CREATE TABLE Persons (  
        ID int NOT NULL,  
        LastName varchar(255) NOT NULL,  
        FirstName varchar(255),  
        Age int,  
        City varchar(255),  
        CONSTRAINT CHK\_Person CHECK (Age>=18 AND City='Sandnes')  
    );
29. ALTER TABLE Persons  
    ADD CHECK (Age>=18);
30. ALTER TABLE Persons  
    ADD CONSTRAINT CHK\_PersonAge CHECK (Age>=18 AND City='Sandnes');
31. ALTER TABLE Persons  
    DROP CHECK CHK\_PersonAge;
32. CREATE TABLE Persons (  
        ID int NOT NULL,  
        LastName varchar(255) NOT NULL,  
        FirstName varchar(255),  
        Age int,  
        City varchar(255) DEFAULT 'Sandnes'  
    );
33. CREATE TABLE Orders (  
        ID int NOT NULL,  
        OrderNumber int NOT NULL,  
        OrderDate date DEFAULT GETDATE()  
    );
34. ALTER TABLE Persons  
    ALTER City SET DEFAULT 'Sandnes';
35. ALTER TABLE Persons  
    ALTER City DROP DEFAULT;
36. CREATE INDEX *index\_name*  
    ON *table\_name* (*column1*, *column2*, ...);
37. CREATE UNIQUE INDEX *index\_name*  
    ON *table\_name* (*column1*, *column2*, ...);
38. ALTER TABLE *table\_name*DROP INDEX *index\_name*;
39. CREATE TABLE Persons (  
        ID int NOT NULL AUTO\_INCREMENT,  
        LastName varchar(255) NOT NULL,  
        FirstName varchar(255),  
        Age int,  
        PRIMARY KEY (ID)  
    );
40. ALTER TABLE Persons AUTO\_INCREMENT=100;
41. INSERT INTO Persons (FirstName,LastName)  
    VALUES ('Lars','Monsen');
42. CREATE VIEW view\_name AS  
    SELECT column1, column2, ...  
    FROM table\_name  
    WHERE condition;
43. CREATE OR REPLACE VIEW view\_name AS  
    SELECT column1, column2, ...  
    FROM table\_name  
    WHERE condition;
44. DROP VIEW view\_name;
45. INJECTION
46. SELECT LastName,FirstName,Address FROM Persons  
    WHERE Address IS NULL
47. SELECT LastName,FirstName,Address FROM Persons  
    WHERE Address IS NOT NULL
48. SELECT column\_name, aggregate\_function(column\_name)  
    FROM table\_name  
    WHERE column\_name operator value  
    GROUP BY column\_name;
49. SELECT column\_name, aggregate\_function(column\_name)  
    FROM table\_name  
    WHERE column\_name operator value  
    GROUP BY column\_name  
    HAVING aggregate\_function(column\_name) operator value;

# **SQL FUNCTIONS**

1. Aggregate Functions

[AVG()](https://www.w3schools.com/sql/sql_func_avg.asp)

[COUNT()](https://www.w3schools.com/sql/sql_func_count.asp)

[FIRST()](https://www.w3schools.com/sql/sql_func_first.asp)

[LAST()](https://www.w3schools.com/sql/sql_func_last.asp)

[MAX()](https://www.w3schools.com/sql/sql_func_max.asp)

[MIN()](https://www.w3schools.com/sql/sql_func_min.asp)

[ROUND()](https://www.w3schools.com/sql/sql_func_round.asp)

[SUM()](https://www.w3schools.com/sql/sql_func_sum.asp)

1. **String Functions**

CHARINDEX

CONCAT()

LEFT()

[LEN() / LENGTH()](https://www.w3schools.com/sql/sql_func_len.asp)

[LOWER() / LCASE()](https://www.w3schools.com/sql/sql_func_lcase.asp)

LTRIM()

[SUBSTRING() / MID()](https://www.w3schools.com/sql/sql_func_mid.asp)

PATINDEX()

REPLACE()

RIGHT()

RTRIM()

[UPPER() / UCASE()](https://www.w3schools.com/sql/sql_func_ucase.asp)

1. **Date Functions**

[NOW()](https://www.w3schools.com/sql/func_now.asp)

[CURDATE()](https://www.w3schools.com/sql/func_curdate.asp)

[CURTIME()](https://www.w3schools.com/sql/func_curtime.asp)

[DATE()](https://www.w3schools.com/sql/func_date.asp)

[EXTRACT()](https://www.w3schools.com/sql/func_extract.asp)

[DATE\_ADD()](https://www.w3schools.com/sql/func_date_add.asp)

[DATE\_SUB()](https://www.w3schools.com/sql/func_date_sub.asp)

[DATEDIFF()](https://www.w3schools.com/sql/func_datediff_mysql.asp)

[DATE\_FORMAT()](https://www.w3schools.com/sql/func_date_format.asp)

1. **NULL() Functions**

SELECT ProductName,UnitPrice\*(UnitsInStock+IFNULL(UnitsOnOrder,0))  
FROM Products

SELECT ProductName,UnitPrice\*(UnitsInStock+COALESCE(UnitsOnOrder,0))  
FROM Products

# **SQL SYNTAX**

1. SELECT column1, column2, ...  
   FROM table\_name;
2. SELECT \* FROM table\_name;
3. SELECT DISTINCT column1,column2, ...  
   FROM table\_name;
4. SELECT COUNT(DISTINCT column1) FROM table\_name;
5. SELECT column1,column2, ...  
   FROM table\_name  
   WHERE condition;
6. SELECT column1,column2, ...  
   FROM table\_name  
   WHERE condition1 AND condition2 AND condition3 ...;
7. SELECT column1,column2, ...  
   FROM table\_name  
   WHERE condition1 OR condition2 OR condition3 ...;
8. SELECT column1,column2, ...  
   FROM table\_name  
   WHERE NOT condition;
9. SELECT column1,column2, ...  
   FROM table\_name  
   ORDER BY column1, column2, ...ASC|DESC;
10. INSERT INTO table\_name (column1,column2,column3, ...)  
    VALUES (value1,value2,value3, ...);
11. INSERT INTO table\_name  
    VALUES (value1,value2,value3, ...);
12. UPDATE table\_name  
    SET column1=value1,column2=value2, ...  
    WHERE condition;
13. DELETE FROM table\_name  
    WHERE condition;
14. DELETE \* FROM table\_name;
15. SELECT column\_name(s)  
    FROM table\_nameWHERE condition  
    LIMIT number;
16. SELECT MIN(column\_name)  
    FROM table\_name  
    WHERE condition;
17. SELECT MAX(column\_name)  
    FROM table\_name  
    WHERE condition;
18. SELECT COUNT(column\_name)  
    FROM table\_name  
    WHERE condition;
19. SELECT AVG(column\_name)  
    FROM table\_name  
    WHERE condition;
20. SELECT SUM(column\_name)  
    FROM table\_name  
    WHERE condition;
21. SELECT column1, column2, ...  
    FROM table\_name  
    WHERE columnN LIKE pattern;
22. SELECT column\_name(s)  
    FROM table\_name  
    WHERE column\_name IN (value1,value2, ...);
23. SELECT column\_name(s)  
    FROM table\_name  
    WHERE column\_name IN (SELECTSTATEMENT);
24. SELECT column\_name(s)  
    FROM table\_name  
    WHERE column\_nameBETWEEN value1 AND value2;
25. SELECT column\_name AS alias\_name  
    FROM table\_name;
26. SELECT column\_name(s)  
    FROM table\_nameAS alias\_name;
27. SELECT column\_name(s)  
    FROM table1  
    INNER JOIN table2  
    ON table1.column\_name=table2.column\_name;
28. SELECT column\_name(s)  
    FROM table1  
    JOIN table2  
    ON table1.column\_name=table2.column\_name;
29. SELECT column\_name(s)  
    FROM table1  
    LEFT JOIN table2  
    ON table1.column\_name=table2.column\_name;
30. SELECT column\_name(s)  
    FROM table1  
    LEFT OUTER JOIN table2  
    ON table1.column\_name=table2.column\_name;
31. SELECT column\_name(s)  
    FROM table1  
    RIGHT JOIN table2  
    ON table1.column\_name=table2.column\_name;
32. SELECT column\_name(s)  
    FROM table1  
    RIGHT OUTER JOIN table2  
    ON table1.column\_name=table2.column\_name;
33. SELECT column\_name(s)  
    FROM table1  
    FULL OUTER JOIN table2  
    ON table1.column\_name=table2.column\_name;
34. SELECT column\_name(s) FROM table1  
    UNION  
    SELECT column\_name(s) FROM table2;
35. SELECT column\_name(s) FROM table1  
    UNION ALL  
    SELECT column\_name(s) FROM table2;
36. SELECT \*  
    INTO newtable [IN externaldb]  
    FROM table1;
37. SELECT column\_name(s)  
    INTO newtable [IN externaldb]  
    FROM table1;
38. INSERT INTO table2  
    SELECT \* FROM table1;
39. INSERT INTO table2  
    (column\_name(s))  
    SELECT column\_name(s)  
    FROM table1;