SQL

ALTER TABLE statement is used to modify the structure of an existing table in a relational database.

Example:

```
Код                                                    ⎘

    ALTER TABLE Customers
    ADD Email VARCHAR(255);
```

SELECT statement is used to select data from a database.

Example

Return data from the Customers table:

```
SELECT CustomerName, City FROM Customers;
```

SELECT DISTINCT statement is used to return only distinct (different) values.

Example

Select all the different countries from the "Customers" table:

```
SELECT DISTINCT Country FROM Customers;
```

EXPLAIN SELECT statement is used to analyze and understand the execution plan of a given SQL query

EXPLAIN [WITH_RECOMMENDATIONS] SQL_statement [;]

https://learn.microsoft.com/queries/explain-transact

DROP TABLE statement is used to drop an existing table in a database.

Example

```
DROP TABLE Shippers;
```

DELETE statement is a Data Manipulation Language (DML) command used to remove one or more rows from a table in a database

Example

```
DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';
```

TRUNCATE statement is a Data Definition Language (DDL) command used to remove all rows from a table, effectively resetting it to an empty state while keeping its structure intact.

Example

```
TRUNCATE TABLE Categories;
```

WHERE statement is used to extract only those records that fulfill a specified condition

Example

Select all customers from Mexico:

```
SELECT * FROM Customers
WHERE Country='Mexico';
```

GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country"

GROUP BY statement is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group the result-set by one or more columns

Example

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country;
```

HAVING statement is used to filter the results of a GROUP BY query based on the values of aggregate functions

Example

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
HAVING COUNT(CustomerID) > 5;
```

LEFT JOIN keyword returns all records from the left table (table1), and the matching records from the right table (table2). The result is 0 records from the right side, if there is no match.
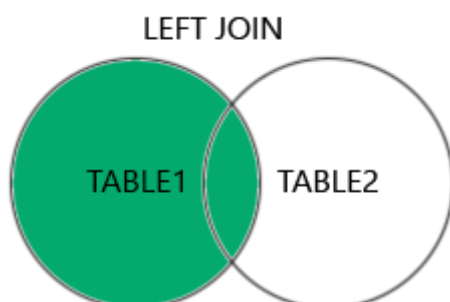

LEFT JOIN

RIGHT JOIN keyword returns all records from the right table (table2), and the matching records from the left table (table1). The result is 0 records from the left side, if there is no match.

RIGHT JOIN
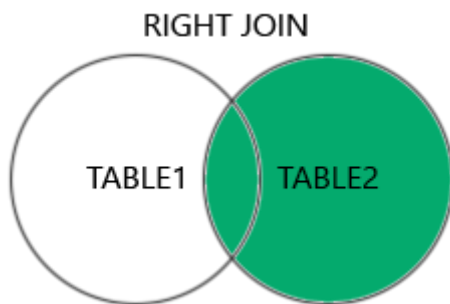
TABLE1   TABLE2

(FULL) OUTER JOIN keyword returns all records when there is a match in left (table1) or right (table2) table records.

FULL OUTER JOIN

TABLE1   TABLE2
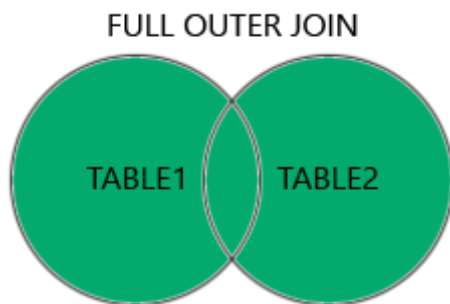
INNER JOIN keyword selects records that have matching values in both tables.

INNER JOIN

TABLE1   TABLE2
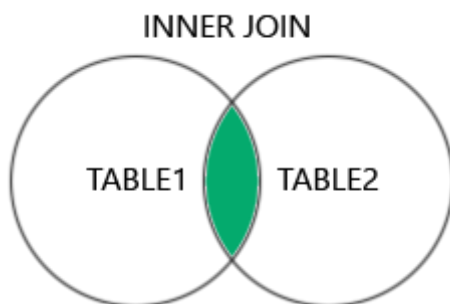
NATURAL JOIN is a type of join, it automatically combines rows from two or more tables based on columns that share the same name and data type in all participating tables. It doesn't require an explicit ON clause to specify the join condition. Instead, the database automatically identifies and uses all common columns for the join.

ITEM_ID	ITEM_NAME	ITEM_UNIT	COMPANY_ID
1	Chex Mix	Pcs	16
6	Cheez-It	Pcs	15
2	BN Biscuit	Pcs	15
3	Mighty Munch	Pcs	17
4	Pot Rice	Pcs	15
5	Jaffa Cakes	Pcs	18
7	Salt n Shake	Pcs	-

COMPANY_ID	COMPANY_NAME	COMPANY_CITY
18	Order All	Boston
15	Jack Hill Ltd	London
16	Akas Foods	Delhi
17	Foodies.	London
19	sip-n-Bite.	New York

** Same column came once

COMPANY_ID	ITEM_ID	ITEM_NAME	ITEM_UNIT	COMPANY_NAME	COMPANY_CITY
16	1	Chex Mix	Pcs	Akas Foods	Delhi
15	6	Cheez-It	Pcs	Jack Hill Ltd	London
15	2	BN Biscuit	Pcs	Jack Hill Ltd	London
17	3	Mighty Munch	Pcs	Foodies.	London
15	4	Pot Rice	Pcs	Jack Hill Ltd	London
18	5	Jaffa Cakes	Pcs	Order All	Boston

ASC keyword delegates to ascending order

DESC keyword delegates to descending order

OVER (PARTITION BY ...) keyword component, enabling the performance of calculations across a set of rows related to the current row, without collapsing the individual rows into a single result

AS command is used to rename a column or table with an alias.
An alias only exists for the duration of the query.

Example

```
SELECT CustomerID AS ID, CustomerName AS Customer
FROM Customers;
```

VIEW keyword is used to define and manage database views

```
Код

CREATE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

ANY means that the condition will be true if the operation is true for any of the values in the range

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator ANY
  (SELECT column_name
  FROM table_name
  WHERE condition);
```

ALL means that the condition will be true only if the operation is true for all values in the range.

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator ALL
  (SELECT column_name
  FROM table_name
  WHERE condition);
```

ORDER BY keyword is used to sort the result-set in ascending or descending order

Example

Sort the products by price:

```
SELECT * FROM Products
ORDER BY Price;
```

LAG statement that is used to access to data from a previous row within the same result set without requiring a self-join

```
Код

SELECT
    Product,
    SalesYear,
    SalesAmount,
    LAG(SalesAmount, 1, 0) OVER (PARTITION BY Product ORDER BY SalesYear) AS Previ
FROM
    ProductSales;
```

CASCADE keyword is a referential action used within foreign key constraints to maintain data integrity across related tables.

ON DELETE CASCADE: When a row in the parent table is deleted, all corresponding rows in the child table that reference the deleted parent row will also be automatically deleted.

```
Код

    CREATE TABLE Orders (
        OrderID INT PRIMARY KEY,
        CustomerID INT
    );

    CREATE TABLE OrderDetails (
        DetailID INT PRIMARY KEY,
        OrderID INT,
        ProductName VARCHAR(50),
        FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) ON DELETE CASCADE
    );
```

PRIMARY KEY keyword in SQL defines a constraint that uniquely identifies each record in a table.

```sql
CREATE TABLE Persons (
    ID int NOT NULL PRIMARY KEY,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int
);
```

SERIAL PRIMARY KEY is a common way to define an auto-incrementing primary key column, particularly in PostgreSQL

```sql
Код

CREATE TABLE users (
    id SERIAL PRIMARY KEY,
    username VARCHAR(50) NOT NULL UNIQUE,
    email VARCHAR(100)
);
```

UNIQUE KEY is a constraint applied to one or more columns in a table to ensure that all values in that column or set of columns are unique.

```sql
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    CONSTRAINT UC_Person UNIQUE (ID,LastName)
);
```

CANDIDATE KEY Any column or set of columns that can uniquely identify a record in a table. All candidate keys are super keys, but not all super keys are candidate keys.

SUPER KEY set of attributes that uniquely identifies tuples (rows) in a table. It can include redundant attributes that are not necessary for uniqueness.

ROLLBACK keyword in SQL is a Transaction Control Language (TCL) command used to undo changes made within the current transaction that have not yet been permanently saved (committed) to the database.

```
ROLLBACK { TRAN | TRANSACTION }
    [ transaction_name | @tran_name_variable
    | savepoint_name | @savepoint_variable ]
[ ; ]
```

COMMIT keyword in SQL is a Transaction Control Language (TCL) command used to permanently save the changes made during a transaction to the database. A transaction is a sequence of SQL statements treated as a single, indivisible unit of work.

```
syntaxsql

COMMIT [ { TRAN | TRANSACTION }
    [ transaction_name | @tran_name_variable ] ]
    [ WITH ( DELAYED_DURABILITY = { OFF | ON } ) ]
[ ; ]
```

SAVEPOINT keyword in SQL is used within a transaction to create a named marker, allowing for partial rollbacks. This means that if an error occurs or a change needs to be undone during a complex transaction, you can revert to a specific SAVEPOINT without having to roll back the entire transaction.

```
syntaxsql

SAVE { TRAN | TRANSACTION } { savepoint_name | @savepoint_variable }
[ ; ]
```

DDL stands for Data Definition Language, a subset of SQL used to define and manage the structure of a database. It includes commands like CREATE, ALTER, and DROP to build, modify, and delete database objects such as tables, indexes, and views.

DML stands for Data Manipulation Language, is a subset of SQL used to manage and modify data within a database. It includes commands to insert, update, delete, and retrieve data, allowing users to work with the actual information stored in tables. Examples of DML commands include SELECT, INSERT, UPDATE, and DELETE

Binary Search Tree algorithm
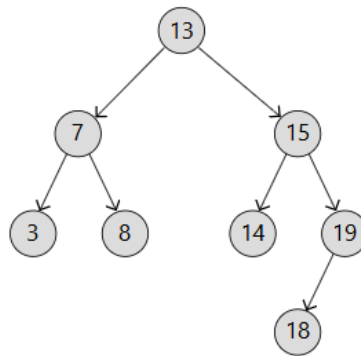
Binary Search Tree (BST)

all values in the left subtree are less than the node's value, and
all values in the right subtree are greater than the node's value

```
sql                                                    Копіювати код

Inorder (sorted): [20, 30, 40, 50, 60, 70, 80]

Tree structure:
└─ 50
     ┌─ 70
     │   └─ 80
     └─ 30
          ┌─ 40
          └─ 20

Search 40 -> found: True path: [50, 30, 40]
Search 25 -> found: False path: [50, 30, 20]

Deleting 70 (node with two children)...
Inorder after deletion: [20, 30, 40, 50, 60, 80]
└─ 50
     ┌─ 80
     │
     └─ 30
          ┌─ 40
          └─ 20
```