JOINING TABLES

Join tables on condition	SELECT Columns
	FROM Table1, Table2
	WHERE [condition]

Inner Join (Same as above)	SELECT Columns
	FROM Table1
	INNER JOIN Table2
	ON [condition]

Shorthand for inner join	SELECT Columns
	FROM Table1
	JOIN Table2
	ON [condition]

Left Join: All rows in left (Table1) table plus matching rows in right (Table2) table.	SELECT Columns
	FROM Table1
	LEFT JOIN Table2
	ON Table1.Column = Table2.Column

Using an alias (AS)	SELECT T1.Col1, T2.Col2
	FROM Table1 AS T1
	JOIN Table2 AS T2
	ON T1.Id = T2.Id

Conditional Join	SELECT Columns FROM Table1 JOIN Table2 ON [condition1] AND [condition2]

Join multiple tables	SELECT Columns
	FROM Table1 AS T1
	JOIN Table2 AS T2
	  ON [condition]
	JOIN Table3 AS T3
	  ON [condition]

Self Join	SELECT Columns
	FROM Table AS T1
	JOIN Table AS T2
	ON condition;

SUBQUERIES

In WHERE Clause	WHERE [columns] [operator] (subquery)

Example	SELECT *
	FROM Movie
	WHERE Year = (SELECT MIN(Year) FROM Movie)

With INSERT	INSERT INTO TableName(col1, col2, …) (subquery);
Example	INSERT INTO OldMovie (SELECT * FROM Movie WHERE ReleaseDate < 1968)
Using IN keyword	WHERE [columns] [NOT] IN (subquery)

Example	SELECT *
	FROM Movie
	WHERE Id IN (SELECT BestMovieId FROM Director)

With ANY/ALL	WHERE column [operator] ANY/ALL (subquery)

ALL Example	SELECT * FROM Director
	WHERE MovieCount > ALL (SELECT MovieCount
	  FROM Director
	  WHERE PlaceOfBirth = 'USA'
	)

With EXISTS	WHERE [NOT] EXISTS (subquery)

Example	SELECT Date
	FROM Orders
	WHERE EXISTS (SELECT ProductName FROM Products
	WHERE Products.ProductID = Orders.ProductID AND
	Price < 20);

In FROM clause	SELECT Columns
	FROM (subquery1) AS S1, (subquery2) AS S2, …

In JOIN	JOIN (subquery) AS Alias [ON condition]
Correlated subquery	(SELECT Column FROM Innertable WHERE OuterTable.Column = InnerTable.column)

Example: Using outer table key column inside subquery	UPDATE Table1 AS T1 SET Column = (
	SELECT Column
	FROM Table2 AS T2
	WHERE T1.Key = T2.Key
	)

CTES

Syntax	WITH cte AS (query)

Reference CTE within a query	WITH cte as (query)
	SELECT columns
	FROM cte
	WHERE *conditions*

Example: CTE named 'aps' which gets the average order amount grouped by store is joined with orders table to list store orders along with the average order	WITH aps AS
	  (SELECT store, AVG(amount) AS average_order
	   FROM orders
	   GROUP BY store)
	SELECT o.store, o.amount, aps.average_order
	FROM orders o
	JOIN aps
	ON o.store = aps.store;

Multiple CTEs	WITH
	   cte1 AS (query),
	   cte2 AS (query), ...

Example	WITH
	   cte1 AS (query),
	   cte2 AS (query)
	SELECT * FROM cte1
	WHERE cte1.column = cte2.column;

Rename columns	WITH cte (col1, col2, …) AS (query)

Example: We get 2 columns from the CTE, Ids and the total number of orders per Id as NumOrders. The average value of NumOrders is then returned as "Avg Sales Per Person"	WITH sales_cte (id, num_orders)
	AS
	(
	    SELECT SalesPersonID, COUNT(*)
	    FROM Sales
	    WHERE SalesPersonID IS NOT NULL
	    GROUP BY SalesPersonID
	)
	SELECT AVG(num_orders) AS "Avg Sales Per Person"
	FROM sales_cte;