The SQL SELECT Statement

The SELECT statement is used to select data from a database.

The data returned is stored in a result table, called the result-set.

SELECT Syntax

SELECT *column1*,*column2, ...*  
FROM *table\_name*;

Here, column1, column2, ... are the field names of the table you want to select data from. If you want to select all the fields available in the table, use the following syntax:

SELECT \* FROM *table\_name*;

The SQL WHERE Clause

The WHERE clause is used to filter records.

It is used to extract only those records that fulfill a specified condition.

WHERE Syntax

SELECT *column1*,*column2, ...*  
FROM *table\_name*  
WHERE *condition*;

**Note:** The WHERE clause is not only used in SELECT statements, it is also used in UPDATE, DELETE, etc.!

The SQL ORDER BY Keyword

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

The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

ORDER BY Syntax

SELECT *column1*,*column2, ...*  
FROM *table\_name*  
ORDER BY *column1, column2, ...*ASC|DESC;

The SQL COUNT(), AVG() and SUM() Functions

The COUNT() function returns the number of rows that matches a specified criterion.

COUNT() Syntax

SELECT COUNT(*column\_name*)  
FROM *table\_name*  
WHERE *condition*;

Select count(\*) from Products where CategoryId=2;

The AVG() function returns the average value of a numeric column.

AVG() Syntax

SELECT AVG(*column\_name*)  
FROM *table\_name*  
WHERE *condition*;

The SUM() function returns the total sum of a numeric column.

SUM() Syntax

SELECT SUM(*column\_name*)  
FROM *table\_name*  
WHERE *condition*;

The SQL GROUP BY Statement

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

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

GROUP BY Syntax

SELECT *column\_name(s)*  
FROM *table\_name*  
WHERE *condition*  
GROUP BY column\_name(s)  
ORDER BY column\_name(s);

Select categoryId, count(\*) from Products group by categoryId;

Fiyatı 20’den fazla ve Ürün sayısı 10’dan az olanları listelemek için;

Select categoryId, count(\*) from Products where unitPrice>20 group by categoryId having count(\*)<10;

SQL INNER JOIN Keyword

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

INNER JOIN Syntax—eşleşmeyen tablo varsa onu getirmez

SELECT *column\_name(s)*  
FROM *table1*  
INNER JOIN *table2*ON *table1.column\_name*=*table2.column\_name*;



SQL LEFT JOIN Keyword

The 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 Syntax

SELECT *column\_name(s)*  
FROM *table1*  
LEFT JOIN *table2*ON *table1.column\_name*=*table2.column\_name*;

**Note:** In some databases LEFT JOIN is called LEFT OUTER JOIN.



SQL RIGHT JOIN Keyword

The 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 Syntax

SELECT *column\_name(s)*  
FROM *table1*  
RIGHT JOIN *table2*ON *table1.column\_name*=*table2.column\_name*;

**Note:** In some databases RIGHT JOIN is called RIGHT OUTER JOIN.



SQL FULL OUTER JOIN Keyword

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

**Tip:** FULL OUTER JOIN and FULL JOIN are the same.

FULL OUTER JOIN Syntax

SELECT *column\_name(s)*  
FROM *table1*  
FULL OUTER JOIN *table2*ON *table1.column\_name*=*table2.column\_name*WHERE *condition*;



**Note:** FULL OUTER JOIN can potentially return very large result-sets!