

INTRODUCTION TO SQL

Table Name

Title 1	Title 2	Title 3	
Data	Data	Data	← Row
Data	Data	Data	← Row
Data	Data	Data	← Row
Data	Data	Data	← Row
↑ Column	↑ Column	↑ Column	

KEYWORDS

→ SELECT

select any data in any column

→ FROM

when selecting data, FROM says which Table

SELECT name FROM people;

- Selecting all name column data in the people table.

SELECT name, birthday FROM people;

- Selecting all name and birthday column data in the people table.

→ LIMIT says us how many rows will select

SELECT name FROM people LIMIT 10;

- Selecting first 10 rows name column data in the people table.

→ DISTINCT finds unique values data

→ COUNT says how many data are there

→ WHERE

Keyword allows filter based search with operators

OPERATORS

=	equal
<>	not equal
<	less than
>	greater than
<=	less than or equals
>=	greater than or equals

NOTES
WHERE keyword always comes after the FROM keyword

`SELECT name FROM people WHERE name = 'Musa';`

Selecting "Musa" names in the people table

IS NULL

Shows null values

IS NOT NULL

Shows not null values

`SELECT name FROM people WHERE job IS NULL;`

Selecting name who haven't job description in the people table

→ WHERE AND Keyword allows multiple filter search with "AND".

`SELECT name FROM people WHERE country = 'Turkey' AND language = 'English';`

Selecting names who from Turkey and know English in people table

→ WHERE OR

`SELECT name FROM customer WHERE country = 'Turkey' OR language = 'English';`

Selecting names who from Turkey or know English in customer table

→ BETWEEN keyword filter values in a specified list

```
SELECT name FROM manager WHERE age BETWEEN 45 AND 50;
```

Selecting names who ages are between 45 and 50 from manager table

→ WHERE IN keyword allows you filter many conditions

```
SELECT name FROM people WHERE age IN (2, 5, 6, 7, 12, 24);
```

Selecting names who ages are 2, 5, 6, 7, 12, 24 from people table

→ LIKE / NOT LIKE % / _

LIKE keyword finds a same data with given value with '%' or '_' operator

```
SELECT name FROM company WHERE name LIKE 'Musa%';
```

Selecting name like Musa and others some

Musa
Musak
Musac
Musaqa
Musakta
⋮

CAUTION
(%)

```
SELECT name FROM name WHERE name LIKE 'M_sa';
```

Musa

Masa

Mtsa

Müsa

⋮

CAUTION
(-)

Mathematical Expressions

AVG () average value

```
SELECT AVG(budget) FROM films;
```

MAX () highest value

```
SELECT MAX(budget) FROM films;
```

MIN () minimum value

```
SELECT MIN(age) FROM people;
```

SUM () total all values

→ AS keyword avoids code duplications and helps more readable code.
(ked teknonimi önlər)

`SELECT MAX(parameters) AS max_per`

MAX(parameters) \Leftrightarrow max_per



ASC
(Ascending)

→ ORDER BY keyword sort and group us results

`SELECT title FROM films ORDER BY releaseYear DESC`

Selecting alphabetically sorting titles from films table

→ GROUP BY allows group us results like:
(male/female) Descending
(Azələr)

→ HAVING

`SELECT title FROM films GROUP BY release-year HAVING
COUNT(title) > 10;`

qillere göre gruplanmış filmlerin 10'den fazla adının listelen
which more than 10 films were released for every year.