* SELECT \* FROM table\_name;
* SELECT CustomerName, City FROM Customers;
* SELECT DISTINCT column1, column2, ...FROM table\_name;
* SELECT COUNT(DISTINCT Country) FROM Customers;
* SELECT \* FROM Customers WHERE Country='Mexico';
* SELECT \* FROM CustomersWHERE Country='Germany'ANDCity='Berlin'
* SELECT \* FROM Customers  
  WHERE Country='Germany' AND Country='USA'; same as below

SELECT \* FROM Customers  
WHERE Country in('Germany','USA')

* SELECT \* FROM Customers  
  WHERE Country='Germany' OR Country='Spain';
* SELECT \* FROM Customers  
  WHERE NOT Country='Germany';
* SELECT \* FROM Customers  
  WHERE Country='Germany' AND (City='Berlin' OR City='München');
* SELECT \* FROM Customers  
  ORDER BY Country ASC, CustomerName DESC;
* SELECT CustomerName, ContactName, Address  
  FROM Customers  
  WHERE Address IS NULL;
* DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste'
* SELECT \* FROM student\_table WHERE grade > 70 LIMIT 2; (Show Two Row)
* select \* from employees where gender='Female' and salary>'70000';
* SELECT \* FROM employees

WHERE salary NOT BETWEEN 80000 AND 90000;

same below

SELECT \* FROM employees

WHERE salary < 80000 OR salary > 90000;

* Select \* from employees

where salary not between 75000 and 95000

order by salary desc;

* Select \* from invoices

where (BillingCountry = 'USA' or BillingCountry = 'Germany')

and (InvoiceDate BETWEEN '2010-01-01' AND '2010-12-31')

ORDER By InvoiceDate DESC

* SELECT \* FROM employees

WHERE job\_title = 'Data Scientist' OR job\_title = 'Business Analyst';

Same below

SELECT \* FROM employees WHERE job\_title

IN ('Data Scientist', 'Business Analyst');

* SELECT \* FROM employees WHERE job\_title

NOT IN ('Operations Director', 'HR Manager', 'Sales Manager');

* Select \* from track where name like('B%') and name LIKE('%S')
* Select first\_name,last\_name,field from student\_info

where field like('Data Science') or field like('Data Analysis') order by first\_name desc;

* SELECT COUNT(field) AS count\_of\_field

FROM student\_info;

Ans: count\_of\_field

32

* SELECT MIN/MAX(salary) AS lowest\_salary

FROM employees WHERE gender = 'Female';

* SELECT MIN/MAX(hire\_date) AS earliest\_date

FROM employees;

* SELECT SUM/AVG(salary) AS total\_salary

FROM employees WHERE gender = 'Female'

* SELECT gender, COUNT(gender) FROM employees

GROUP BY gender;

gender COUNT(gender)

---------- -------------

Female 4

Male 6

* Select employees.emp\_id, employees.first\_name, employees.salary, departments.dept\_name

from employees

INNER JOIN departments

ON employees.emp\_id=departments.emp\_id;

**WHERE Methodun da SUM AVG vs kullanilmaz.**

**Group By da Min Max vs. kullanilmaz.**

Exercise:

1. How many tracks does each album have? Your solution should include Album id and its number of tracks sorted from highest to lowest.

**SELECT AlbumId, COUNT(Name)**

**from tracks**

**GROUP BY AlbumId**

**ORDER BY COUNT(Name)DESC;**

1. Find the album title of the tracks. Your solution should include track name and its album title.

**SELECT tracks.Name, albums.Title**

**FROM tracks**

**INNER JOIN albums**

**ON tracks.AlbumId=albums.AlbumId;**

1. Find the minimum duration of the track in each album. Your solution should include album id, album title and duration of the track sorted from highest to lowest.

**SELECT tracks.AlbumId, albums.Title, min(tracks.Milliseconds)**

**FROM tracks**

**INNER JOIN albums**

**ON tracks.AlbumId=albums.AlbumId**

**GROUP BY albums.AlbumId**

**ORDER BY min(tracks.Milliseconds) DESC;**

1. Find the total duration of each album. Your solution should include album id, album title and its total duration sorted from highest to lowest.

**SELECT tracks.AlbumId, albums.Title, sum(tracks.Milliseconds)**

**FROM tracks**

**INNER JOIN albums**

**ON tracks.AlbumId=albums.AlbumId**

**GROUP BY albums.AlbumId**

**ORDER BY sum(tracks.Milliseconds) DESC;**

1. Based on the previous question, find the albums whose total duration is higher than 70 minutes. Your solution should include album title and total duration.

**SELECT albums.Title, sum(tracks.Milliseconds)**

**FROM tracks**

**INNER JOIN albums**

**ON tracks.AlbumId=albums.AlbumId**

**GROUP BY albums.AlbumId**

**HAVING SUM(tracks.Milliseconds)>4200000**

**ORDER BY sum(tracks.Milliseconds)DESC;**