**SQL CHECK YOUR SELF – 2**

SELECT first\_name FROM employees;

SELECT first\_name, last\_name FROM employees;

SELECT first\_name FROM employees;

**SQL CHECK YOUR SELF – 3**

1. Please write a query for selecting the **unique values of "first\_name" column** in the **"student\_info"** table.

SELECT DISTINCT first\_name FROM student\_info;

1. Please write a query for selecting the **unique values of the "last\_name" column** in the **"student\_info"** table.

SELECT DISTINCT last\_name FROM student\_info;

1. Please write a query for selecting the **unique values of "state\_name" column** in the **"student\_info"** table.

SELECT DISTINCT state\_name FROM student\_info;

1. Please write a query for selecting the **unique values of both "state\_name" and "county\_name" columns** in the **"student\_info"** table.

SELECT DISTINCT state\_name, county\_name FROM student\_info;

1. Please write a query for selecting the **unique values of both "field" and "start\_date" columns** in the **"student\_info"** table.

SELECT DISTINCT field, start\_date FROM student\_info;

**SQL CHECK YOUR SELF – 4 ; sözel soru cevaplar buraya alınmadı.**

**SQL CHECK YOURK SELF – 5**

1. Please write a query to return the salary of employees from highest to lowest. Show just first name, last name and salary of the employees.

SELECT first\_name, last\_name, salary FROM employees ORDER BY salary DESC;

1. Please write a query to return all records from employees table sorted by gender (display females first). Next, sort the result table by first names in descending order. **Note:** Write a single query

SELECT \* FROM employees ORDER BY gender ASC, first\_name DESC;

1. Which employee has been working in the company for the longest period?

SELECT \* FROM employees ORDER BY hire\_date ASC LIMIT 1;

**SQL CHECK YOURK SELF – 6**

1. Please write a query to return female employees whose salary is higher than $70,000?

**SELECT \***

**FROM employees**

**WHERE gender = "Female" AND salary>70000**

1. Please write a query to return employees whose job title is web developer or data scientist.

**SELECT \***

**FROM employees**

**WHERE job\_title = 'Data Scientist' OR job\_title = 'Web Developer';**

1. Please write a query to return all employees whose job title is not Data Scientist?

**SELECT \***

**FROM employees**

**WHERE NOT job\_title = 'Data Scientist'**

**SQL CHECK YOURK SELF – 7**

1. Please write a query to return employees whose salary is between $75,000 and $95,000?

SELECT \*

FROM employees

WHERE salary BETWEEN 75000 AND 95000;

1. Please write a query to return employees whose salary is not between $75,000 and $95,000. Sort it by salary in descending order.

SELECT \*

FROM employees

WHERE salary NOT BETWEEN 75000 AND 95000

ORDER BY salary DESC;

1. Please write a query to return employees who have joined the company from January 1, 2018 to December 31, 2018. Sort it by hire date in ascending order.

SELECT \*

FROM employees

WHERE hire\_date BETWEEN '2018-01-01' AND '2018-12-31'

ORDER BY hire\_date;

**CHECK YOUR SELF – 8**

1. You have decided to build a project team from students. The project team will be composed of students from the fields of Data Analysis, Data Science and DevOps. You are searching for the right candidates.

SELECT \* FROM student\_info WHERE field IN ('Data Analysis', 'Data Science', 'DevOps');

1. Please modify your previous query to return all possible candidates from Virginia state.

SELECT \* FROM student\_info WHERE field IN ('Data Analysis', 'Data Science', 'DevOps') AND state = 'Virginia';

**CHECK YOUR SELF – 9**

1. Please write a query to return the students whose field is Data Science or Data Analysis. Your query should return the first name, last name and field of each student and should be ordered by the first name in descending order.

SELECT first\_name, last\_name, field

FROM student\_info

WHERE field LIKE 'Data%'

ORDER BY first\_name DESC;

1. Please write a query to return the last names including'er' letters.

SELECT last\_name

FROM student\_info

WHERE last\_name LIKE '%er%';

1. Please write a query to match 'r' letter in"Eric" in the first name column. Just return the first name column.

SELECT first\_name

FROM student\_info

WHERE first\_name LIKE 'E\_ic';

ASSİNGMENT;

1. Write a query that displays InvoiceId, CustomerId and total dollar amount for each invoice, sorted first by CustomerId (in ascending order), and then by total dollar amount  (in descending order).

2. Write a query that displays InvoiceId, CustomerId and total dollar amount for each invoice, but this time sorted first by total dollar amount (in descending order), and then by CustomerId (in ascending order).

3. Compare the results of these two queries above. How are the results different when you switch the column you sort on first? (Explain it in your own words.)

4. Write a query to pull the first 10 rows and all columns from the invoices table that have a dollar amount of total greater than or equal to 10.  
  
5. Write a query to pull the first 5 rows and all columns from the invoices table that have a dollar amount of total less than 10.  
  
6. Find all track names that start with 'B' and end with 's'.  
  
7. Use the invoices table to find all information regarding invoices whose billing address is USA or Germany or Norway or Canada and invoice date is at any point in 2010, sorted from newest to oldest.

ANSWER

1) SELECT InvoiceId, CustomerId, total FROM invoices ORDER BY CustomerId ASC, total DESC;

2) SELECT InvoiceId, CustomerId, total FROM invoices ORDER BY total DESC, CustomerId ASC;

3) Firstly, it sorts the first command, then it sorts the other command in accordance with the first command. So if the command order changes, our table will be changed,

4) SELECT \* FROM invoices WHERE total >= 10 LIMIT 10;

5) SELECT \* FROM invoices WHERE total < 10 LIMIT 5;

6) SELECT \* FROM tracks WHERE name like 'B%s';

7) SELECT \* FROM invoices WHERE BillingCountry IN('USA', 'Germany', 'Norway', 'Canada') AND InvoiceDate BETWEEN '2010-01-01' AND '2010-12-31' ORDER BY InvoiceDate DESC;

AGGREGATE FUNCTİONS AND JOİNS,

**CHECK YOUR SELF – 1**

1. Please write a query to return the number of students as "number\_of\_students" from West Virginia State.

SELECT COUNT (\*) AS number\_of\_students

FROM student\_info

WHERE state = 'West Virginia';

1. Please write a query to return the number of female students as "number\_of\_students" from Virginia State.

SELECT COUNT(\*) AS number\_of\_students

FROM student\_info

WHERE state = 'Virginia' AND gender = 'F';

1. Please write a query to return distinct number of job titles as "number\_of\_titles" in the company.

SELECT COUNT (DISTINCT job\_title) AS number\_of\_titles

FROM employees

**CHECK YOUR SELF – 2**

1. Please write a query to return the earliest start date as 'earliest\_date"' among the female students

SELECT MIN(start\_date) AS earliest\_date

FROM student\_info

WHERE gender = 'F'

1. Please rewrite the previous query to return the earliest start date among the female students without using the MIN function. Your query should return just a single column. (Note: Please don't use AS keyword in the query)

SELECT start\_date

FROM student\_info

WHERE gender = 'F'

ORDER BY start\_date

LIMIT 1;

1. Please write a query to return the newest start date as 'newest\_date' among the male students.

SELECT MAX(start\_date) AS newest\_date

FROM student\_info

WHERE gender = 'M'

1. Please rewrite the previous query to return the newest start date among the male students. Your query should return just a single column. Please don't use AS keyword and MAX function in the query.

SELECT start\_date

FROM student\_info

WHERE gender = 'M'

ORDER BY start\_date DESC

LIMIT 1;

**CHECK YOUR SELF – 3**

1. Please write a query to return the total grade as 'total\_grade'.

SELECT SUM(grade) AS total\_grade

FROM student\_table;

1. Please write a query to return the total grade in Mathematics lesson as 'total\_grade'.

SELECT SUM(grade) AS total\_grade

FROM student\_table

WHERE lesson ='Mathematics'

1. Please write a query to return the average grade as 'average\_grade'.

SELECT AVG(grade) AS average\_grade

FROM student\_table

1. Please write a query to return the average grade in Physics lesson as 'average\_grade'.

SELECT AVG(grade) AS average\_grade

FROM student\_table

WHERE lesson = 'Physics'

**CHECK YOUR SELF – 4**

1. Please write a query to return the total salary of each department. The result table's column headers will be dept\_name and total\_salary.

SELECT name, MAX (AlbumId)

AS max\_albümıd

FROM tracks

GROUP BY name

ORDER BY max\_albümıd ASC

**CHECKYOUR SELF -6**

1. Please write a query to return the departments of the employees who have a salary higher than 80000. Show just first name, last name, salary and department of the employees.

SELECT

employees.first\_name,

employees.last\_name,

employees.salary,

departments.dept\_name

FROM employees

INNER JOIN departments

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

WHERE employees.salary>80000;

1. Please write a query to return the departments of female employees. Show just first name, last name, salary and department of the employees.

SELECT

employees.first\_name,

employees.last\_name,

employees.salary,

departments.dept\_name

FROM employees

INNER JOIN departments

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

WHERE employees.gender='Female';

***Check your self 7***

1. Using the Left JOIN method, write a query to return the departments of all male employees in the "employees" table. Show just first name, last name, department and department ID of the employees.

SELECT

employees.first\_name,

employees.last\_name,

departments.dept\_name,

departments.dept\_id

FROM employees

LEFT JOIN departments

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

WHERE employees.gender = 'Male'

1. Higher Than 75 Marks

SELECT Name

FROM STUDENTS

WHERE Marks > 75

ORDER BY substr(Name, -3, 3),

ID ASC;

1. Employee Names

SELECT name

FROM Employee

ORDER BY name;

\*\*\* Write a query that prints a list of employee names (i.e.: the name attribute) for employees in **Employee** having a salary greater than  per month who have been employees for less than  months. Sort your result by ascending employee\_id.

SELECT name

FROM Employee

WHERE salary>2000 AND months<10

ORDER BY employee\_id;