1-Tut : **SQL Query - 8**

Send Feedback

Problem Statement:

Enlist the email ids of all the interns along with their names.

Information about the table:

Attributes list:

	DATA TYPES
Emp_ID :- Employee ID	INT
Emp_name :- Employee Full Name	VARCHAR
Dept :- Department Employee is working in.	VARCHAR
Contract :- Full Time employee or Intern.	VARCHAR
Email :- Employee official mail ID.	VARCHAR
HomeTown:- Employees HomeTown.	VARCHAR

Table **Emp_data**:

Emp_ID	Emp_name	Dept	Contract	Email	HomeTown
546	Rakesh Matam	D1	FTE	fabcd1@xyz.com	Patna
1111	Kuldeep Ravaliya	D3	Intern	intdef1@xyz.com	Indore
670	Sugam Sehgal	D4	FTE	fabcd3@xyz.com	Himachal
1110	Sumit Mishra	D3	Intern	intdef2@xyz.com	Patna
890	Lokesh Daga	D2	FTE	fabcd5@xyz.com	Bikaner
700	Rakesh Matam	D3	FTE	fabcd6@xyz.com	Ludhiana
1251	Ram Kumar	D4	FTE	fabcd7@xyz.com	Guwahati
1300	Shayam Singh	D2	Intern	intdef3@xyz.com	Ludhiana
245	Neelabh Shukla	D4	FTE	fabcd8@xyz.com	Kota
210	Barkha Singh	D3	FTE	fabcd9@xyz.com	Mumbai
500	Rohan Arora	D5	Intern	intdef4@xyz.com	Jalandhar

Output Table Structure:

```
     emp_name      |       email
-------------------+-----------------
```

Query : **SELECT Emp_name,Email FROM Emp_data WHERE Contract = 'Intern';**

2-Tut : SQL Query - 9

Send Feedback

Problem Statement:

Fetch the records of all employees of Department D3 with the FTE offer

Information about the table:

Attributes list:

Table **Emp_data**:

Output Table Structure:

```
 emp_id |   emp_name   | dept | contract |     email      | hometown
--------+--------------+------+----------+----------------+----------
```

Query : **SELECT * FROM Emp_data WHERE dept = 'D3' AND contract = 'FTE';**

3-Tut : SQL Query - 10

Send Feedback

Problem Statement:

Fetch all the records of the employees working for department D1 or D3.

Query : **SELECT * FROM emp_data WHERE dept = 'D1' OR dept = 'D3';**

4-Tut : SQL Query - 12

Send Feedback

Problem Statement:

Fetch all the records of the employees working for department D1 or D3 using the IN clause

Query : **SELECT * FROM Emp_data WHERE dept IN ('D1','D3');**

5-Tut : SQL Query - 13

Send Feedback

Problem Statement:

Fetch all the records of employees that neither work for department D1 nor for D2.

Query : **select * from emp_data where dept not in ('D1','D2');**

6-Tut : SQL Query - 14

Send Feedback

Problem Statement:

List down the employee name and department that are either from Mumbai or Jalandhar and hold an employee Id numbered less than 900.

Query : **SELECT Emp_name,Dept FROM Emp_data WHERE HomeTown IN ('Mumbai','Jalandhar') AND Emp_ID < 900;**

7-Tut : **SQL Query - 17**

Send Feedback

Problem Statement:

Fetch all the records with Email ID's starting from 'fab'.

Query : **SELECT * FROM Emp_data WHERE Email LIKE 'fab%';**

8-Tut : SQL Query - 18

Send Feedback

Problem Statement:

List down all the employee id's and names whose Email contains 'bcd' and belongs to department D3 or D4 but aren't from Himachal, Guwahati.

Query : SELECT Emp_ID,Emp_name FROM Emp_data WHERE Email LIKE '%bcd%' AND Dept IN ('D3','D4') AND HomeTown NOT IN ('Himachal','Guwahati');

9-Tut : SQL Query - 23

Send Feedback

Problem Statement:

List down the Order ID's and their respective Ordering time, arranged in Ascending order by ordering time.

Information about the table:

Attributes list:

	DATA TYPES
order_id :- ID of orders done by customers.	VARCHAR
ordered_time :- Time at which orders were placed.	DATE
shipping_time :- Shipping time for those orders.	DATE
cost :- Cost of the product ordered.	INT
cust_id :- Customer ID for every order.	VARCHAR

Table e_transactions:

order_id	ordered_time	shipping_time	cost	cust_id
CN70101	2021-02-22	2021-02-28	5679	Cid065
CN70102	2021-02-22	2021-02-25	7999	Cid019
CN70103	2021-06-14	2021-06-18	1300	Cid07
CN70104	2021-07-07	2021-07-11	13299	Cid098
CN70105	2021-07-07	2021-07-12	1754	Cid032

Output Table Structure

```
+----------+--------------+
| order_id | ordered_time |
+----------+--------------+
```

Query : SELECT order_id,ordered_time FROM e_transactions ORDER BY ordered_time ASC;

10-Tut : SQL Query - 24

Send Feedback

Problem Statement:

Arrange the above-given data in descending order by Shipping time.

Query : SELECT * FROM e_transactions ORDER BY Shipping_time DESC;

11-Tut : SQL Query - 25

Send Feedback

Problem Statement:

Fetch out all the records but in descending order by ordering time and in case of similar order times sort in ascending order w.r.t. shipping time.

Query : SELECT * FROM e_transactions ORDER BY ordered_time DESC, shipping_time ASC;

12-Tut : SQL Query - 26

Send Feedback

Problem Statement:

List down all the order details in ascending order by cost and whose cost is less than 5000.

Query : SELECT * FROM e_transactions AS e WHERE e.cost < 5000;

13-Tut : SQL Query - 27

Send Feedback

Problem Statement:

List down the orders ids with their shipping time which were ordered before 30th June 2021 sort them in ascending order w.r.t. cost and in descending order w.r.t. time the purchase was made.

Query : SELECT order_id, shipping_time FROM e_transactions AS e WHERE e.ordered_time < '2021-06-30' ORDER BY e.cost ASC, e.ordered_time DESC;

14-Tut : SQL Query - 28

Send Feedback

Problem Statement:

List down all the details of the orders made in February 2021 or July 2021, also sort them in ascending order by their price.

Query 1 : SELECT * FROM e_transactions as e WHERE e.ordered_time BETWEEN '2021-02-01' AND '2021-02-28' OR e.ordered_time BETWEEN '2021-07-01' AND '2021-07-31' ORDER BY e.cost ASC;

Query 2 : SELECT * FROM e_transactions WHERE ordered_time LIKE '%-02-%' OR ordered_time LIKE '%-07-%' ORDER BY cost;

ASSIGNMENT :

15-Ass : SQL Query

Send Feedback

Given below is the Employee Data for a XYZ organisation, as visible we are provided with employee details with columns like,

	DATA TYPES
Emp_ID :- Employee ID	INT
Emp_name :- Employee Full Name	VARCHAR
Dept :- Department Employee is working in.	VARCHAR
Contract :- Full Time employee or Intern.	VARCHAR
Email :- Employee official mail ID.	VARCHAR
HomeTown:- Employees HomeTown.	VARCHAR

Table Emp_data:-

Emp_id	Emp_name	Dept	Contract	Email	HomeTown
546	Rakesh Matam	D1	FTE	fabcd1@xyz.com	Patna
1111	Kuldeep Ravaliya	D3	Intern	intdef1@xyz.com	Indore
670	Sugam Sehgal	D4	FTE	fabcd3@xyz.com	Himachal
1110	Sumit Mishra	D3	Intern	intdef2@xyz.com	Patna
890	Lokesh Daga	D2	FTE	fabcd5@xyz.com	Bikaner
700	Rakesh Matam	D3	FTE	fabcd6@xyz.com	Ludhiana
1251	Ram Kumar	D4	FTE	fabcd7@xyz.com	Guwahati
1300	Shayam Singh	D2	Intern	intdef3@xyz.com	Ludhiana
245	Neelabh Shukla	D4	FTE	fabcd8@xyz.com	Kota
210	Barkha Singh	D3	FTE	fabcd9@xyz.com	Mumbai
500	Rohan Arora	D5	Intern	intdef4@xyz.com	Jalandhar

Who are the employees with FTE offers but working with either department D1 or D3.

Query : SELECT * FROM Emp_data WHERE Contract = 'FTE' AND Dept IN ('D1','D3');

16-Ass : SQL Query

Send Feedback

Given below is the Employee Data for a XYZ organisation, as visible we are provided with employee details with columns like,

	DATA TYPES
Emp_ID :- Employee ID	INT
Emp_name :- Employee Full Name	VARCHAR
Dept :- Department Employee is working in.	VARCHAR
Contract :- Full Time employee or Intern.	VARCHAR
Email :- Employee official mail ID.	VARCHAR
HomeTown:- Employees HomeTown.	VARCHAR

Table Emp_data:-

Emp_id	Emp_name	Dept	Contract	Email	HomeTown
546	Rakesh Matam	D1	FTE	fabcd1@xyz.com	Patna
1111	Kuldeep Ravaliya	D3	Intern	intdef1@xyz.com	Indore
670	Sugam Sehgal	D4	FTE	fabcd3@xyz.com	Himachal
1110	Sumit Mishra	D3	Intern	intdef2@xyz.com	Patna
890	Lokesh Daga	D2	FTE	fabcd5@xyz.com	Bikaner
700	Rakesh Matam	D3	FTE	fabcd6@xyz.com	Ludhiana
1251	Ram Kumar	D4	FTE	fabcd7@xyz.com	Guwahati
1300	Shayam Singh	D2	Intern	intdef3@xyz.com	Ludhiana
245	Neelabh Shukla	D4	FTE	fabcd8@xyz.com	Kota
210	Barkha Singh	D3	FTE	fabcd9@xyz.com	Mumbai
500	Rohan Arora	D5	Intern	intdef4@xyz.com	Jalandhar

List down all the Departments. (No repetitions allowed)

Query : SELECT DISTINCT Dept FROM Emp_data;

17-Ass : SQL Query

Send Feedback

Given below is the Employee Data for a XYZ organisation, as visible we are provided with employee details with columns like,

List down the emails of all employees with their names whose hometown is from any of these: Patna, Ludhiana, Bikaner.
Order of attributes for output table- Emp_name, Email

Query : SELECT Emp_name,Email FROM Emp_data WHERE HomeTown IN ('Patna','Ludhiana','Bikaner');

18-Ass : SQL Query - 27

Send Feedback

Problem Statement:
List down the orders ids with their shipping time which were ordered before 30th June 2021 sort them in ascending order w.r.t. cost and in descending order w.r.t. time the purchase was made.
Information about the table:
Attributes list:

	DATA TYPES
order_id :- ID of orders done by customers.	VARCHAR
ordered_time :- Time at which orders were placed.	DATE
shipping_time :- Shipping time for those orders.	DATE
cost :- Cost of the product ordered.	INT
cust_id :- Customer ID for every order.	VARCHAR

Table e_transactions:

order_id	ordered_time	shipping_time	cost	cust_id
CN70101	2021-02-22	2021-02-28	5679	Cid065
CN70102	2021-02-22	2021-02-25	7999	Cid019
CN70103	2021-06-14	2021-06-18	1300	Cid07
CN70104	2021-07-07	2021-07-11	13299	Cid098
CN70105	2021-07-07	2021-07-12	1754	Cid032

Output Table Structure

```
order_id | shipping_time
---------+--------------
```

Query : SELECT order_id,shipping_time FROM e_transactions WHERE ordered_time < '2021-06-30' ORDER BY cost ASC, ordered_time DESC;

19-Ass : SQL Query

Send Feedback

Given below is the e-transaction data for an e-commerce website. As visible we are provided with order details done by different customers. Here,

	DATA TYPES
order_id :- ID of orders done by customers.	VARCHAR
ordered_time :- Time at which orders were placed.	DATE
shipping_time :- Shipping time for those orders.	DATE
cost :- Cost of the product ordered.	INT
cust_id :- Customer ID for every order.	VARCHAR

Table e_transactions:-

order_id	ordered_time	shipping_time	cost	cust_id
CN70101	2021-02-22	2021-02-28	5679	Cid065
CN70102	2021-02-22	2021-02-25	7999	Cid019
CN70103	2021-06-14	2021-06-18	1300	Cid07
CN70104	2021-07-07	2021-07-11	13299	Cid098
CN70105	2021-07-07	2021-07-12	1754	Cid032

List down all the details of the orders made except for the ones whose cost were 1300,7999 and sort them in ascending order by the price.

Query : SELECT * FROM e_transactions WHERE cost NOT IN (1300,7999) ORDER BY cost ASC;