

Problem Statement 1:

1. Data Detective (Introduction to Data Transformation & Functions):

Objective:

- Understand the concept of data transformation and its importance in data analysis.
- Master basic SQL functions for data manipulation and cleaning.

Task:

1. Use a sample dataset from your chosen database management software or a provided one. The dataset could represent customer information, sales data, or any relevant data.

-- Create the Customers table

```
CREATE TABLE Customers1 (  
    CustomerID INT IDENTITY PRIMARY KEY,  
    FirstName VARCHAR(50),  
    LastName VARCHAR(50),  
    Email VARCHAR(100),  
    BirthDate DATE,  
    JoinDate DATE,  
    City VARCHAR(50)  
);
```

-- Insert data into the Customers table

```
INSERT INTO Customers1 (FirstName, LastName, Email, BirthDate, JoinDate, City)  
VALUES  
(  
    'John', 'Doe', 'john.doe@example.com', '1985-04-12', '2020-06-01', 'New York'),  
(  
    'Jane', 'Smith', 'jane.smith@example.com', '1990-07-22', '2021-02-15', 'Los  
    Angeles'),  
(  
    'Alice', 'Johnson', 'alice.j@example.com', '1975-11-30', '2019-03-10',
```

'Chicago'),

('Bob', 'Lee', 'bob.lee@example.com', '1982-02-18', '2022-05-30', 'San Francisco'),

('Charlie', 'Brown', 'charlie.b@example.com', '1992-09-08', '2020-10-25', 'Seattle'),

('David', 'Williams', 'david.w@example.com', '1988-01-15', '2021-07-20', 'Houston'),

('Emma', 'Taylor', 'emma.t@example.com', '1993-05-27', '2019-08-14', 'Phoenix'),

('Michael', 'Brown', 'michael.b@example.com', '1980-03-10', '2020-11-05', 'Philadelphia'),

('Olivia', 'Martinez', 'olivia.m@example.com', '1995-12-06', '2022-03-22', 'San Diego'),

('Ethan', 'Anderson', 'ethan.a@example.com', '1989-07-30', '2021-01-10', 'Dallas'),

('Sophia', 'Thomas', 'sophia.t@example.com', '1991-04-18', '2020-04-20', 'Austin'),

('James', 'Jackson', 'james.j@example.com', '1983-06-25', '2019-09-12', 'San Jose'),

('Mia', 'White', 'mia.w@example.com', '1992-11-14', '2020-12-30', 'Columbus'),

('Alexander', 'Harris', 'alex.h@example.com', '1987-08-02', '2021-02-27', 'Fort Worth'),

('Isabella', 'Clark', 'isabella.c@example.com', '1994-05-09', '2019-03-05', 'Charlotte'),

('Daniel', 'Lewis', 'daniel.l@example.com', '1981-12-19', '2020-07-22', 'Detroit'),

('Amelia', 'Robinson', 'amelia.r@example.com', '1986-03-05', '2021-05-16', 'El Paso'),

('Matthew', 'Walker', 'matthew.w@example.com', '1990-08-21', '2019-06-10', 'Memphis'),

('Emily', 'Young', 'emily.y@example.com', '1984-10-15', '2020-11-07', 'Boston'),

('Jacob', 'Allen', 'jacob.a@example.com', '1995-01-30', '2021-03-25', 'Seattle'),

('Harper', 'King', 'harper.k@example.com', '1989-07-11', '2020-01-18', 'Denver'),

('Ava', 'Green', 'ava.g@example.com', '1993-04-07', '2019-10-12', 'Nashville'),

('Elijah', 'Baker', 'elijah.b@example.com', '1991-09-22', '2020-02-28',

```

'Oklahoma City'),

('Abigail', 'Nelson', 'abigail.n@example.com', '1985-05-04', '2021-06-15', 'Las
Vegas'),

( 'Logan', 'Carter', 'logan.c@example.com', '1994-10-27', '2019-11-20',
'Portland'),

('Charlotte', 'Mitchell', 'charlotte.m@example.com', '1987-02-03', '2020-05-12',
'Louisville'),

('Lucas', 'Perez', 'lucas.p@example.com', '1992-06-13', '2021-07-08',
'Baltimore'),

('Avery', 'Roberts', 'avery.r@example.com', '1988-03-26', '2020-08-14',
'Milwaukee'),

('Henry', 'Turner', 'henry.t@example.com', '1990-09-01', '2019-01-25',
'Albuquerque'),

('Ella', 'Phillips', 'ella.p@example.com', '1984-12-08', '2021-10-03',
'Tucson'),

('Benjamin', 'Campbell', 'benjamin.c@example.com', '1993-05-21', '2020-03-15',
'Fresno'),

('Scarlett', 'Parker', 'scarlett.p@example.com', '1991-08-16', '2019-12-22',
'Sacramento'),

('Mason', 'Evans', 'mason.e@example.com', '1986-11-01', '2021-04-19', 'Kansas
City'),

('Liam', 'Edwards', 'liam.e@example.com', '1995-07-28', '2020-09-07',
'Atlanta');

```

2. Explore various functions for data manipulation:

String Functions: Use LOWER, UPPER, or SUBSTRING to clean and format text data (e.g., converting all names to uppercase).

SQLQuery2.sql - AC8...ikeStores (sa (73)) * SQLQuery1.sql - AC8...ikeStores (sa (72))

```

select * from Customers1
select customerid,UPPER(firstname) Firstname,UPPER(lastname) Lastname from customers1

```

100 %

Results Messages

	customerid	Firstname	Lastname
1	1	JOHN	DOE
2	2	JANE	SMITH
3	3	ALICE	JOHNSON
4	4	BOB	LEE
5	5	CHARLIE	BROWN
6	6	DAVID	WILLIAMS
7	7	EMMA	TAYLOR
8	8	MICHAEL	BROWN
9	9	OLIVIA	MARTINEZ
10	10	ETHAN	ANDERSON

SQLQuery2.sql - AC8...ikeStores (sa (73)) * SQLQuery1.sql - AC8...ikeStores (sa (72))

```

select * from Customers1
--Converting all names to uppercase using UPPER function
select customerid,UPPER(firstname) Firstname,UPPER(lastname) Lastname from customers1
--Converting all names to lower using lower function
select customerid,LOWER(firstname) Firstname,LOWER(lastname) Lastname from customers1

```

100 %

Results Messages

	customerid	Firstname	Lastname
1	1	john	doe
2	2	jane	smith
3	3	alice	johnson
4	4	bob	lee
5	5	charlie	brown
6	6	david	williams
7	7	emma	taylor
8	8	michael	brown
9	9	olivia	martinez
10	10	ethan	anderson

Query executed successfully. AC8DDA201F1A5B8 (16.0 RTM) sa (73) BikeStores 00:00:00 34 rows

SQLQuery2.sql - AC8...ikeStores (sa (73))* SQLQuery1.sql - AC8...ikeStores (sa (72))

```

select * from customers1
--Converting all names to uppercase using UPPER function
select customerid,UPPER(firstname) Firstname,UPPER(lastname) Lastname from customers1
--Converting all names to lower using lower function
select customerid,Lower(firstname) Firstname,lower(lastname) Lastname from customers1
--substring function
select Firstname,SUBSTRING(firstname,1,2) Substr from customers1

```

100 %

Results Messages

	Firstname	Substr
1	John	Jo
2	Jane	Ja
3	Alice	Al
4	Bob	Bo
5	Charlie	Ch
6	David	Da
7	Emma	Em
8	Michael	Mi
9	Olivia	Ol
10	Ethan	Et

Query executed successfully. AC8DDA201F1A5B8 (16.0 RTM) sa (73) BikeStores 00:00:00 34 rows

o Date Functions: Apply YEAR, MONTH, or DAY functions to extract specific date components for further analysis.

SQLQuery2.sql - AC8...ikeStores (sa (73))* SQLQuery1.sql - AC8...ikeStores (sa (72))

```

select Firstname,SUBSTRING(firstname,1,2) Substr from customers1
--Date Functions: Apply YEAR, MONTH, or DAY functions to extract specific date components for further analysis.
select customerid,joindate,year(joindate) Join_Year,MONTH(joindate) Join_Month,Day(joindate) Join_day from customers1

```

100 %

Results Messages

	customerid	joindate	Join_Year	Join_Month	Join_day
1	1	2020-06-01	2020	6	1
2	2	2021-02-15	2021	2	15
3	3	2019-03-10	2019	3	10
4	4	2022-05-30	2022	5	30
5	5	2020-10-25	2020	10	25
6	6	2021-07-20	2021	7	20
7	7	2019-08-14	2019	8	14
8	8	2020-11-05	2020	11	5
9	9	2022-03-22	2022	3	22
10	10	2021-01-10	2021	1	10

Query executed successfully. AC8DDA201F1A5B8 (16.0 RTM) sa (73) BikeStores 00:00:00 34 rows

o Aggregate Functions (COUNT & COUNT DISTINCT): Calculate the total number of records and the number of distinct values in a specific column.

--Aggregate Functions (COUNT & COUNT DISTINCT): Calculate the total number of records and the number of distinct values in a specific column
--count function
select count(*) TotalNoCustomers from customers1

100 %

Results Messages

	TotalNoCustomers
1	34

Query executed successfully. AC8DDA201F1A5B8 (16.0 RTM) sa (73) BikeStores 00:00:00 1 rows

select count(distinct(city)) City_count from customers1

100 %

Results Messages

	City_count
1	33

Query executed successfully. AC8DDA201F1A5B8 (16.0 RTM) sa (73) BikeStores 00:00:00 1 rows