



DE- TASK 4

Bytewise Ltd



JULY 1, 2024
JAHANZEB AHMED

DE - Task 2

Table Queries:

```
CREATE TABLE mypublic."cities"(  
    city_id VARCHAR(100) PRIMARY KEY,  
    name VARCHAR(100),  
    province VARCHAR(100),  
    country VARCHAR(100),  
    status VARCHAR(100)  
);
```

```
CREATE TABLE mypublic."categories"(  
    category_id VARCHAR(100) PRIMARY KEY,  
    name VARCHAR(100),  
    status VARCHAR(100),  
    description VARCHAR(200)  
);
```

```
CREATE TABLE mypublic."customers"(  
    customer_id VARCHAR(100) PRIMARY KEY,  
    name VARCHAR(100),  
    phone VARCHAR(50),  
    location VARCHAR(100),  
    status VARCHAR(50)  
);
```

```
CREATE TABLE mypublic."items"(  
    item_id VARCHAR(100) PRIMARY KEY,
```

```
        order_id VARCHAR(100),  
        product_id VARCHAR(100),  
        amount FLOAT,  
        status VARCHAR(30),  
        item_timestamp TIMESTAMP  
    );
```

```
CREATE TABLE mypublic."orders"(  
    order_id VARCHAR(100) PRIMARY KEY,  
    customer_id VARCHAR(100),  
    status VARCHAR(30),  
    order_timestamp TIMESTAMP,  
    total_amount FLOAT  
);
```

```
CREATE TABLE mypublic."products"(  
    product_id VARCHAR(100) PRIMARY KEY,  
    name VARCHAR(30),  
    supplier_id VARCHAR(100),  
    category VARCHAR(30),  
    price FLOAT,  
    stock_available INT,  
    status VARCHAR(30),  
    product_createtimestamp TIMESTAMP  
);
```

```
CREATE TABLE mypublic."suppliers"(  
    supplier_id VARCHAR(100) PRIMARY KEY,  
    name VARCHAR(100),
```

```
    phone VARCHAR(50),  
    location VARCHAR(50),  
    status VARCHAR(30),  
    category VARCHAR(20)  
);
```

Importing Tables into Database:

Example:

```
COPY mypublic."cities" FROM 'C:\Users\DELL\OneDrive\Desktop\New Database\cities.csv' CSV HEADER;
```

Q1: Write a query to fetch all customer names and sort them alphabetically. (Topic Sorting)

Query:

```
SELECT name AS customers_name FROM mypublic.customers  
  
ORDER BY name;
```

Result:

Data Output		Messages	Notifications
<div> <div>≡+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> <div>SQL</div> </div>			
	customers_name character varying (100) 🔒		
1	Aaron Hayes		
2	Aaron Jones		
3	Aaron Schwartz		
4	Abigail Reed		
5	Adam David		
6	Adam Johns		
7	Adam Martinez		
8	Alan Massey		
9	Alex Smith		
Total rows: 1000 of 1000		Query complete 00:00:00.105	

Q2: Write a query to fetch all product names and their prices, sorted by price from low to high. (Topic Sorting)

Query:

```
SELECT name AS product_name, price
FROM mypublic.products
ORDER BY price;
```

Result:

Data Output Messages Notifications		
<div> <div> <div>+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> <div>SQL</div> </div> </div>		
	product_name character varying (30) 🔒	price double precision 🔒
1	deal	11.1
2	interview	12.37
3	step	12.72
4	especially	12.74
5	game	12.78
6	level	13.59
7	include	13.75
8	their	16.27
9	build	17.25
Total rows: 1000 of 1000		Query complete 00:00:00.113

Q3: Write a query to fetch supplier names that start with the letter 'A' and sort them by their names. (Topic Sorting with Operators and Wildcards)

Query:

```
SELECT name AS supplier_names
FROM mypublic.suppliers
WHERE name LIKE 'A%'
ORDER BY name;
```

Result:

Data Output		Messages	Notifications
<div> <div>≡+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> <div>SQL</div> </div>			
	supplier_names character varying (100) <div>🔒</div>		
1	Acosta-Freeman		
2	Adams-Roberts		
3	Adams-Schwartz		
4	Adams, Hull and Wise		
5	Adams, Rojas and Morgan		
6	Adkins and Sons		
7	Aguilar-Carter		
8	Aguilar Ltd		
9	Aguirre, Taylor and Gregory		
Total rows: 46 of 46		Query complete 00:00:00.272	

Q4: Write a query to fetch all items and sort them by their status, placing NULL values first.

Query:

```
SELECT *
FROM mypublic.items
ORDER BY status NULLS FIRST;
```

Result:

Query

Query History

1

2

3

4

5

SELECT *

FROM mypublic.items

ORDER BY status NULLS FIRST;

Scratch Pad

Data Output

Messages

Notifications

Q5: Write a query to fetch all products, sort them first by category and then by price in descending order.

Query:

```
SELECT *
FROM mypublic.products
ORDER BY category, price DESC;
```

Result:

Query History
Scratch Pad x

```

1 SELECT *
2 FROM mypublic.products
3 ORDER BY category, price DESC;
4

```

Data Output
Messages
Notifications

	product_id [PK] character varying (100)	name character varying (30)	supplier_id character varying (100)	category character varying (30)	price double precision	stock_available integer	status character varying (30)
1	12ea3377-cca5-4965-a41c-9a30998310...	of	3f89d404-d78f-4aca-8c67-ee4e43467d45	a	302.68	979	InActive
2	494c69e7-db73-42fd-9eef-7880822585a5	offer	01abe709-0567-47dc-ae92-3b077c44b8...	able	385.18	137	Active
3	b8adc046-eef0-4f58-9c7a-7c6f58a2d628	share	6252ec97-c226-4c94-8b70-7ce1642719...	able	76.71	283	InActive
4	a63e349b-fb6e-4222-aca0-5079603843bf	wide	16157d6a-f40a-4140-b233-c8e5e8542b...	about	398.27	851	Out of Stock
5	e61f1dad-1ec7-4114-aaf5-7a449f91c7b4	factor	c9a89c46-4f89-4cc4-a804-be28738fc04b	about	91.72	974	Out of Stock
6	2d61572a-ad68-4eb9-a327-cbde171ce7...	middle	f292cfe7-c77d-4dd4-8b75-b5eb72f08ece	above	337.62	812	InActive
7	e37e126c-f0d4-4b67-8efc-5d4226a23a18	door	7b539636-fcdb-4749-9e31-f9e38c83cec8	above	281.2	275	InActive
8	dad2a61c-41d1-49ea-96b1-6c78daa687...	Congress	c2721d21-27ef-474c-82a6-5dcf1e3b6fcd	above	93.24	928	Active
9	ef209ah5-5d43-4496-8669-8071c78hR8	vnah	2ae9ah92-e46f-4c86-h155-cn020004c8	accent	88.18	339	InActive

Total rows: 1000 of 1000 Query complete 00:00:00.470

Q6: Write a query to fetch all customer names and phone numbers, but sort them by the last four digits of their phone numbers in ascending order. (Hint use sorting with substrings)

Query:

```
SELECT name, phone
```

FROM mypublic.customers

```
ORDER BY substr(phone, length(phone)-4);
```

Result:

Query Query History

```
1  SELECT name, phone
2  FROM mypublic.customers
3  ORDER BY substr(phone, length(phone)-4);
```

Data Output Messages Notifications

	name character varying (100)	phone character varying (50)
1	Ashley Cobb	001-587-449-0043
2	Erin Ward	+1-640-549-0044
3	Abigail Reed	203-341-0115
4	Curtis Holt	587-441-0140
5	Bradley Weeks	(918)461-0244
6	Lisa Ford	001-677-627-0290
7	Mary Taylor	(458)396-0347
8	Brent Petty	704-826-0368
9	Kendra Quinn	637-991-0372
Total rows: 1000 of 1000		Query complete 00:00:00.188