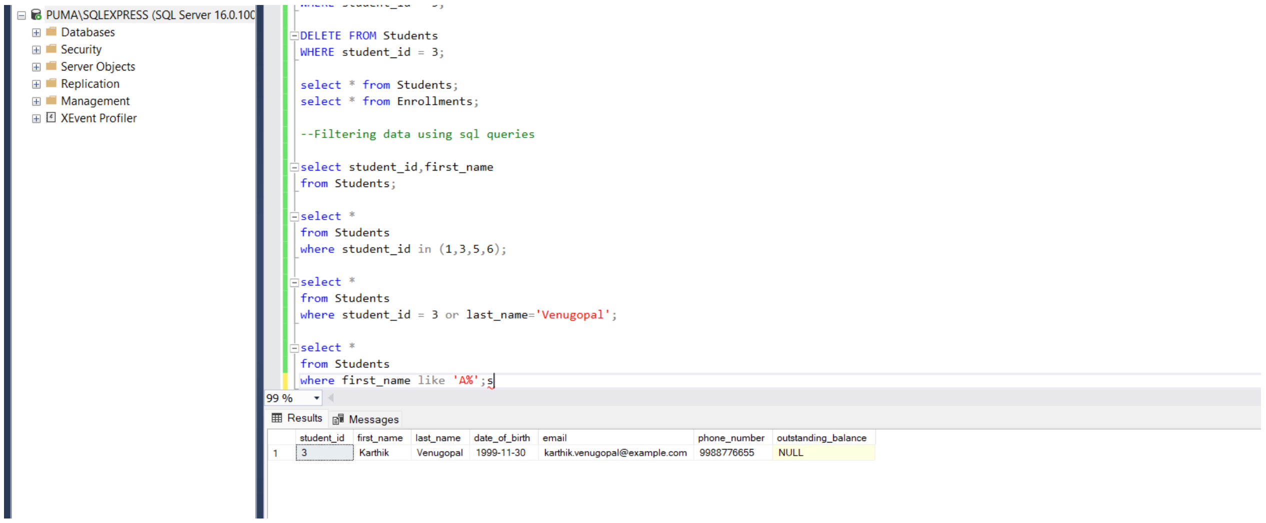
Manipulating Data using DML

Name: Aathirainathan P

Date: 05-11-2024

---

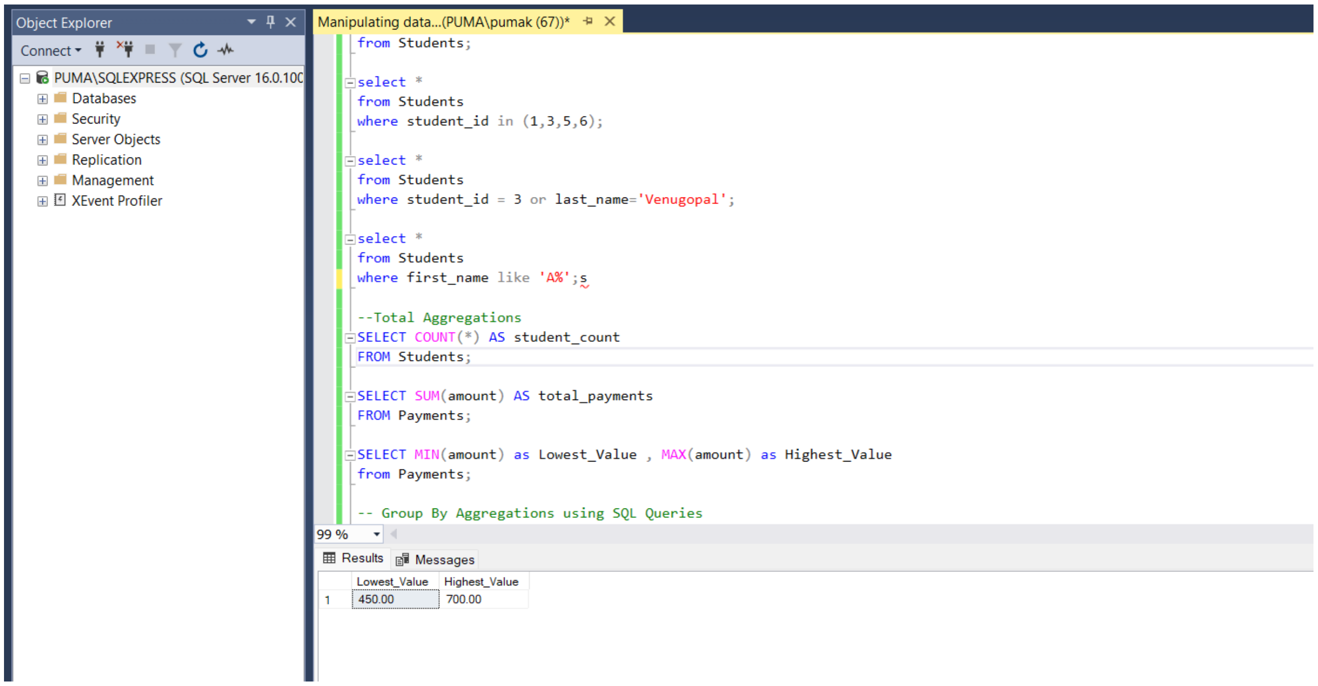
1. Filtering data using sql queries:



```sql
DELETE FROM Students
WHERE student_id = 3;

select * from Students;
select * from Enrollments;

--Filtering data using sql queries

select student_id,first_name
from Students;

select *
from Students
where student_id in (1,3,5,6);

select *
from Students
where student_id = 3 or last_name='Venugopal';

select *
from Students
where first_name like 'A%';s
```

	student_id	first_name	last_name	date_of_birth	email	phone_number	outstanding_balance
1	3	Karthik	Venugopal	1999-11-30	karthik.venugopal@example.com	9988776655	NULL

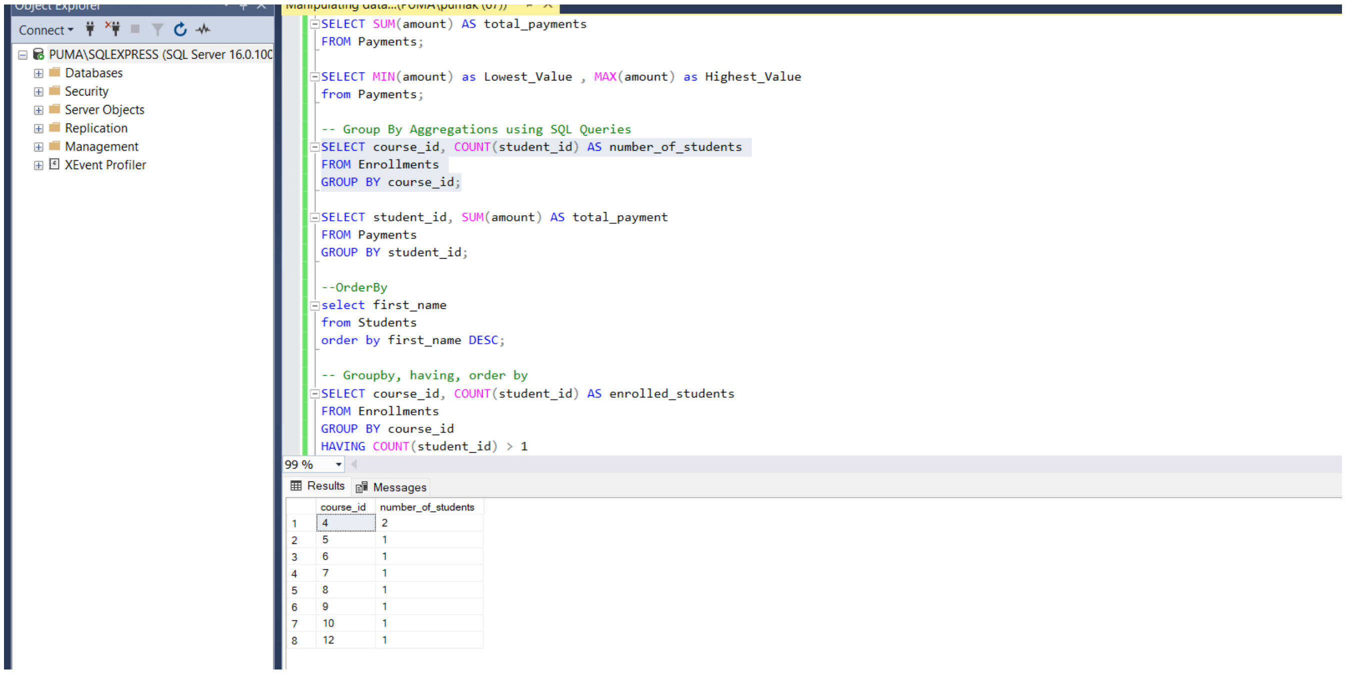
2. Total Aggregations:



```sql
from Students;

select *
from Students
where student_id in (1,3,5,6);

select *
from Students
where student_id = 3 or last_name='Venugopal';

select *
from Students
where first_name like 'A%';s

--Total Aggregations
SELECT COUNT(*) AS student_count
FROM Students;

SELECT SUM(amount) AS total_payments
FROM Payments;

SELECT MIN(amount) as Lowest_Value , MAX(amount) as Highest_Value
from Payments;

-- Group By Aggregations using SQL Queries
```

	Lowest_Value	Highest_Value
1	450.00	700.00

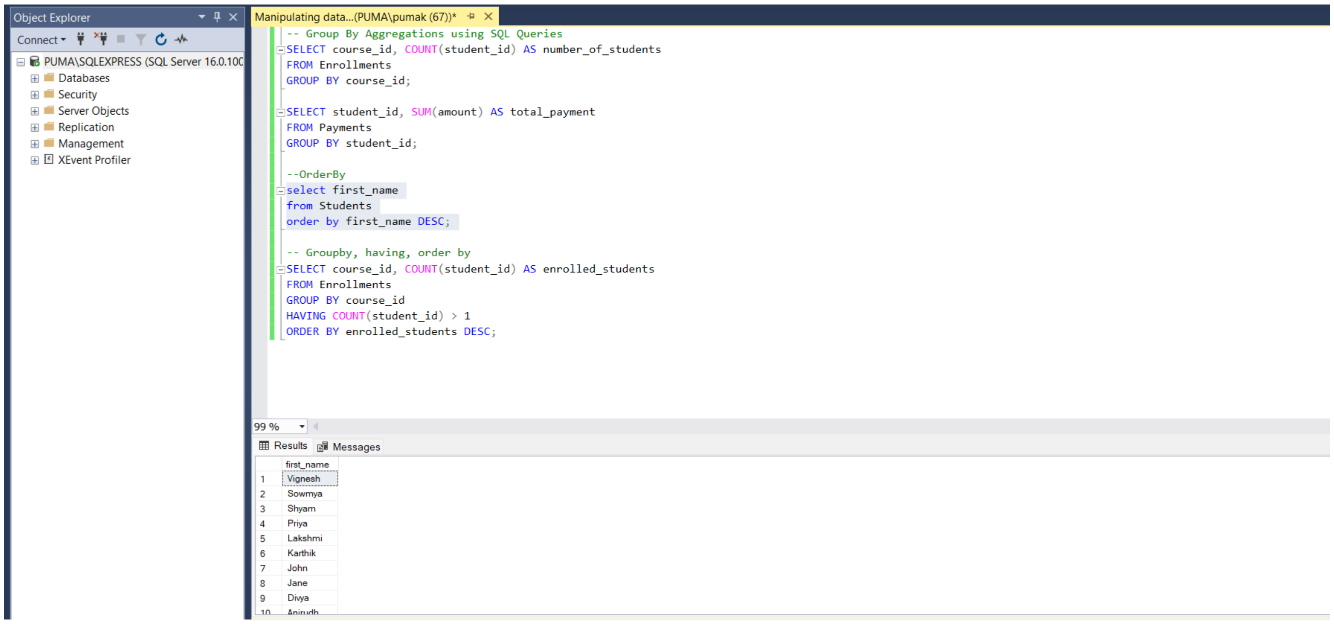
3. Group By Aggregations using SQL Queries:



```sql
SELECT SUM(amount) AS total_payments
FROM Payments;

SELECT MIN(amount) as Lowest_Value , MAX(amount) as Highest_Value
from Payments;

-- Group By Aggregations using SQL Queries
SELECT course_id, COUNT(student_id) AS number_of_students
FROM Enrollments
GROUP BY course_id;

SELECT student_id, SUM(amount) AS total_payment
FROM Payments
GROUP BY student_id;

--OrderBy
select first_name
from Students
order by first_name DESC;

-- Groupby, having, order by
SELECT course_id, COUNT(student_id) AS enrolled_students
FROM Enrollments
GROUP BY course_id
HAVING COUNT(student_id) > 1
```

	course_id	number_of_students
1	4	2
2	5	1
3	6	1
4	7	1
5	8	1
6	9	1
7	10	1
8	12	1

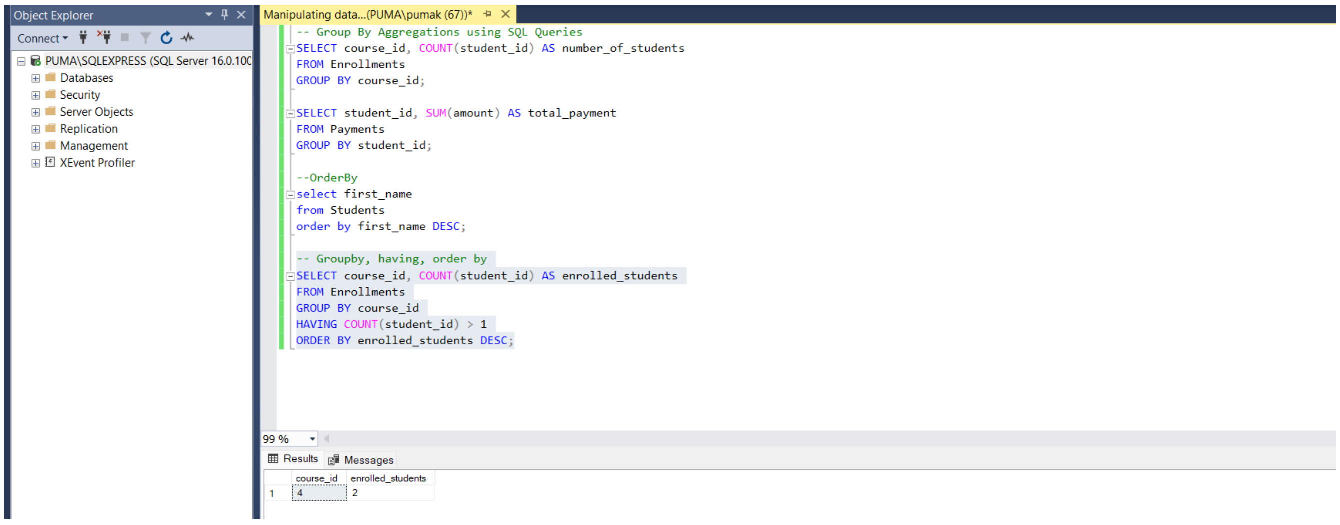
4. OrderBy:



```sql
-- Group By Aggregations using SQL Queries
SELECT course_id, COUNT(student_id) AS number_of_students
FROM Enrollments
GROUP BY course_id;

SELECT student_id, SUM(amount) AS total_payment
FROM Payments
GROUP BY student_id;

--OrderBy
select first_name
from Students
order by first_name DESC;

-- Groupby, having, order by
SELECT course_id, COUNT(student_id) AS enrolled_students
FROM Enrollments
GROUP BY course_id
HAVING COUNT(student_id) > 1
ORDER BY enrolled_students DESC;
```

	first_name
1	Vignesh
2	Sowmya
3	Shyam
4	Priya
5	Lakshmi
6	Karthik
7	John
8	Jane
9	Divya
10	Anirudh

5. Groupby, having, order by:



```sql
-- Group By Aggregations using SQL Queries
SELECT course_id, COUNT(student_id) AS number_of_students
FROM Enrollments
GROUP BY course_id;

SELECT student_id, SUM(amount) AS total_payment
FROM Payments
GROUP BY student_id;

--OrderBy
select first_name
from Students
order by first_name DESC;

-- Groupby, having, order by
SELECT course_id, COUNT(student_id) AS enrolled_students
FROM Enrollments
GROUP BY course_id
HAVING COUNT(student_id) > 1
ORDER BY enrolled_students DESC;
```

	course_id	enrolled_students
1	4	2