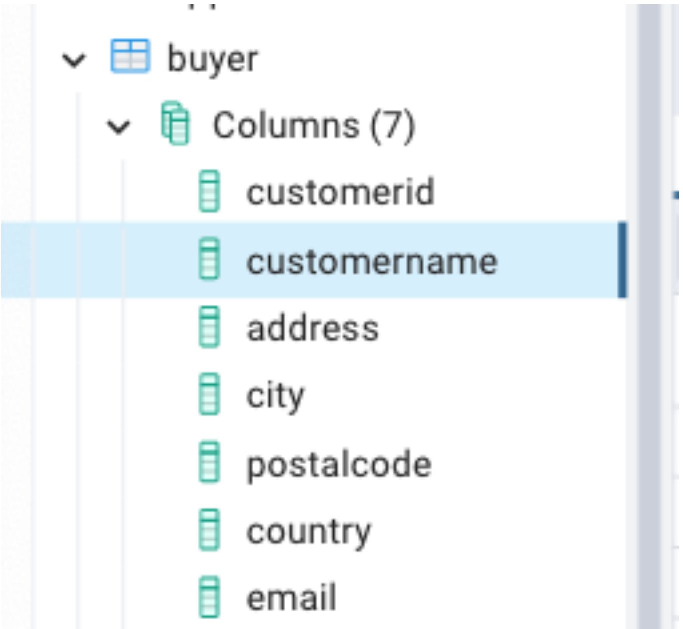
Name: Tashi palden

Lesson 9: Insert and Update

1. **Create** a table with the following parameters:
    - CustomerID
    - CustomerName
    - Address
    - City
    - PostalCode
    - Country
    - Email

```
create table buyer

(

    customerID serial Primary key,

    CustomerName character varying,

    Address character varying,

    City character varying,

    PostalCode char  ,

    Country character varying,

    Email character varying

);
```



buyer

Columns (7)

customerid

customername

address

city

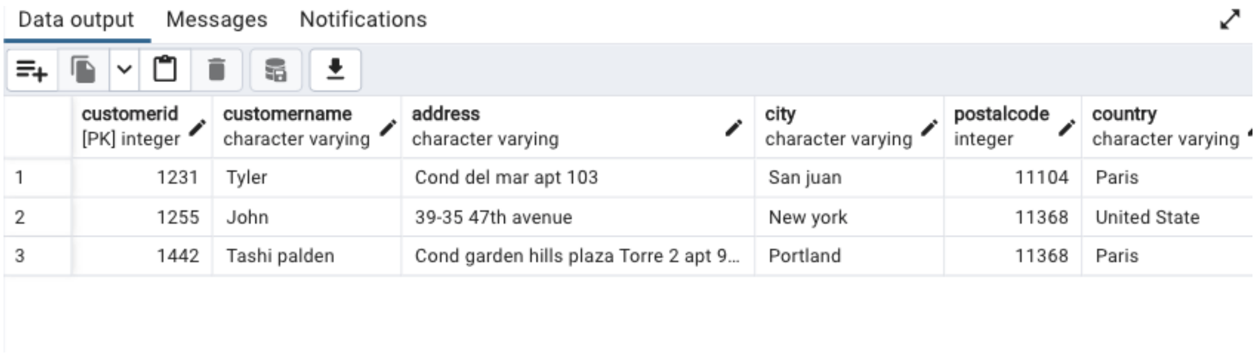
postalcode

country

email

2. Insert 3 rows of data into these columns using **INSERT**. The data you insert should make sense for the column.

```
insert into buyer (customerID, CustomerName,Address, City, PostalCode,Country,Email)
values
(1231,'Tyler','Cond del mar apt 103','San juan',11104,'Paris','maddog@gmail.com'),
(1255, 'John','39-35 47th avenue','New york','11368','United State','John53@gmail.com'),
( 1442,'Tashi palden','Cond garden hills plaza Torre 2 apt 905','Portland',11368,'Paris',
 'Tahipaden07@gmail.com');
```

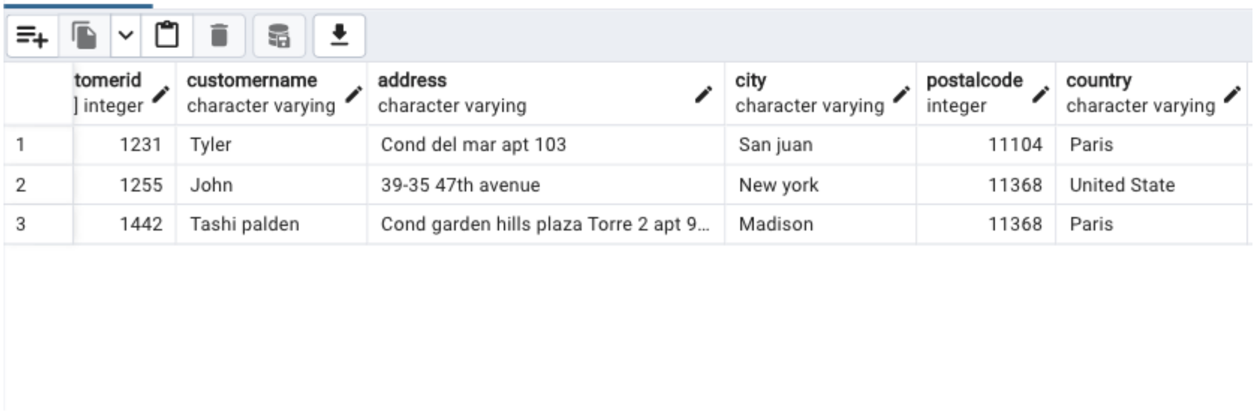


	customerid [PK] integer	customername character varying	address character varying	city character varying	postalcode integer	country character varying
1	1231	Tyler	Cond del mar apt 103	San juan	11104	Paris
2	1255	John	39-35 47th avenue	New york	11368	United State
3	1442	Tashi palden	Cond garden hills plaza Torre 2 apt 9...	Portland	11368	Paris

3 Use an **UPDATE** to modify any portion of the data

```
update buyer

set city='Madison'
```

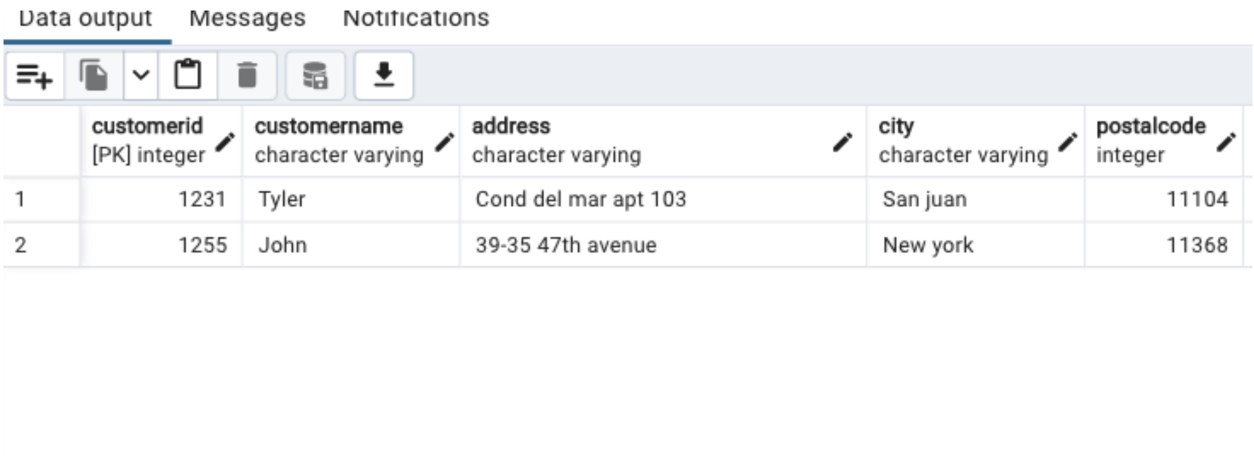
where customername like 'Tashi palden%'



	tomerid] integer	customername character varying	address character varying	city character varying	postalcode integer	country character varying
1	1231	Tyler	Cond del mar apt 103	San juan	11104	Paris
2	1255	John	39-35 47th avenue	New york	11368	United State
3	1442	Tashi palden	Cond garden hills plaza Torre 2 apt 9...	Madison	11368	Paris

3. Finally, write a statement to **delete** one row of data.

   delete from buyer where customerid=1442

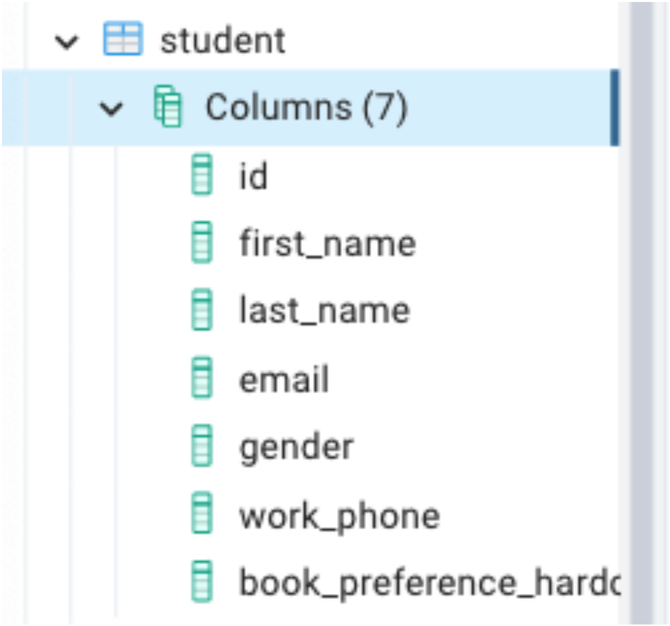


	customerid [PK] integer	customername character varying	address character varying	city character varying	postalcode integer
1	1231	Tyler	Cond del mar apt 103	San juan	11104
2	1255	John	39-35 47th avenue	New york	11368

1. Using the following Link https://github.com/niteen11/cuny_lagcc_micro_credential_data_analytics/tree/main/Track%20A/Unit%205%20-%20SQL_%20Relational%20Databases/guided%20exercise

First you have to create a table than upload the data ,safe the table in to your Laptop and change the path accordingly.usr the following link for creating table, https://github.com/niteen11/cuny_lagcc_micro_credential_data_analytics/blob/main/Track%20A/Unit%205%20-%20SQL_%20Relational%20Databases/guided%20exercise/student.sql

```
DROP TABLE IF EXISTS student;

CREATE TABLE student

(

    id serial PRIMARY KEY,

    first_name character varying,

    last_name character varying,

    email character varying,

    gender character varying,

    work_phone character varying,

    book_preference_hardcopy boolean

);


copy student(first_name,last_name,email,gender,work_phone,book_preference_hardcopy)

--set the path for file location of student_data.csv

from '/Applications/PostgreSQL 14/Student_data.csv'

delimiter ',' CSV header
```



student

Columns (7)

id

first_name

last_name

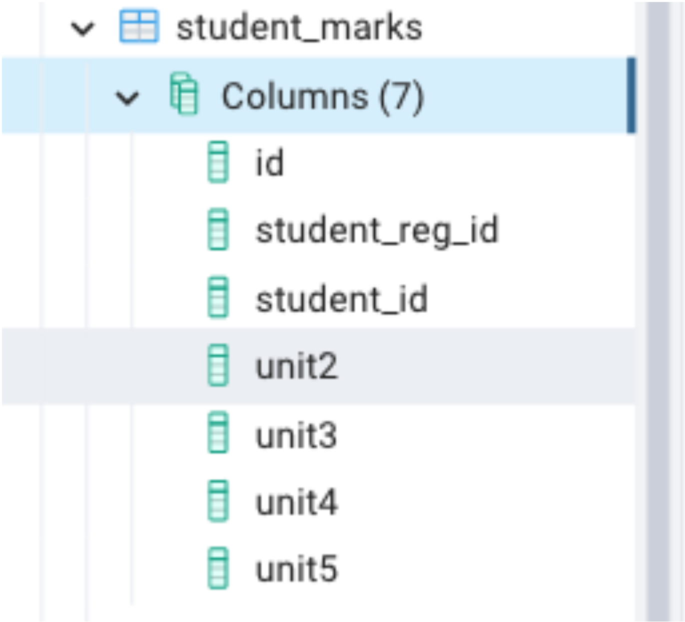
email

gender

work_phone

book_preference_hardc

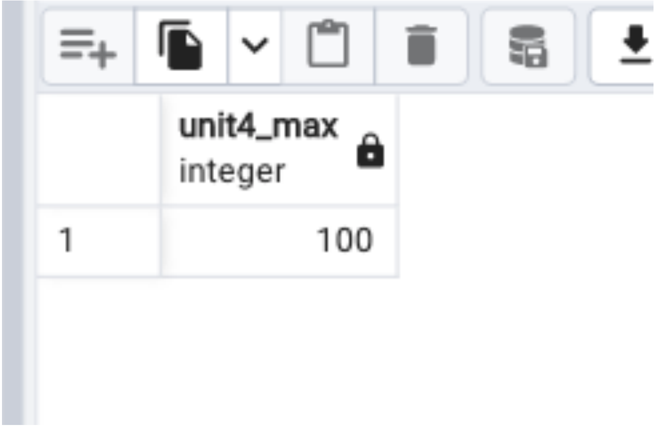
```
DROP TABLE IF EXISTS student_marks;

CREATE TABLE student_marks

(

  id serial PRIMARY KEY,

  student_reg_id integer,

  student_id integer,

        unit2 integer,

        unit3 integer,

        unit4 integer,

        unit5 integer

);


copy student_marks(student_reg_id,student_id,unit2,unit3,unit4,unit5)

--set the path for file location of student_marks.csv

from '/Applications/PostgreSQL 14/student_marks.csv'

delimiter ',' CSV header
```



student_marks
Columns (7)
id
student_reg_id
student_id
unit2
unit3
unit4
unit5

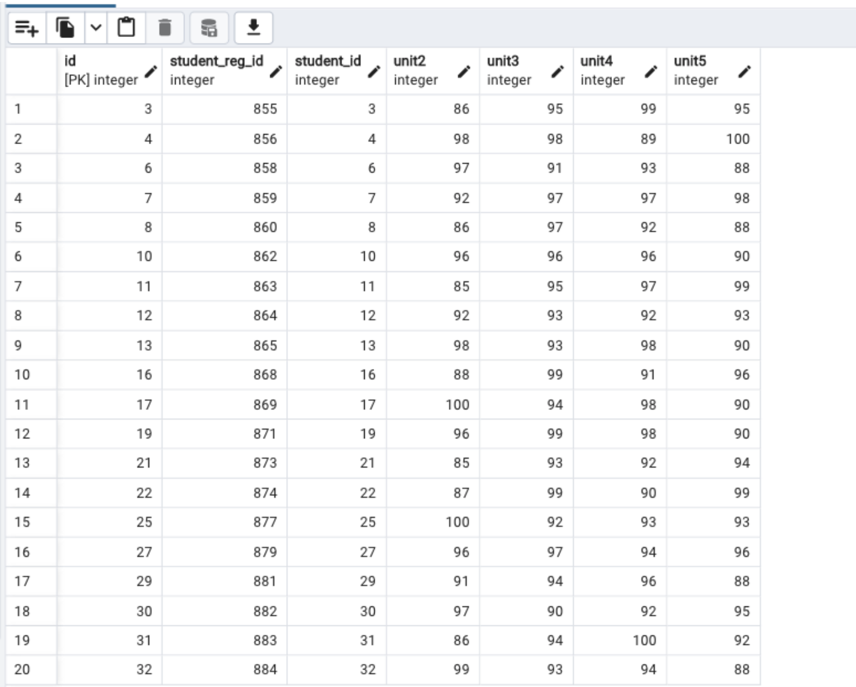
Sample questions:

```
-- students with the highest marks in Unit 4


select max(unit4)as unit4_max
from student_marks
```



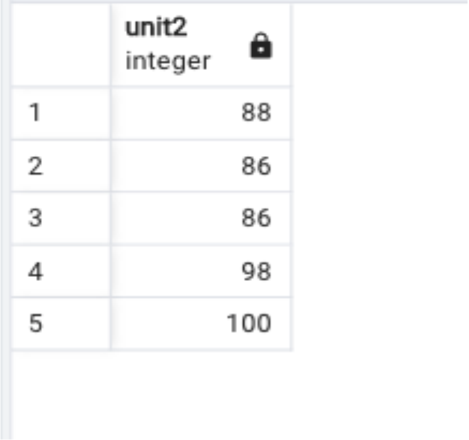
	unit4_max integer
1	100

```
-- students scored between 89 and 100 unit4

select * from student_marks
where unit4>=89 and unit4<=100
limit 20;
```



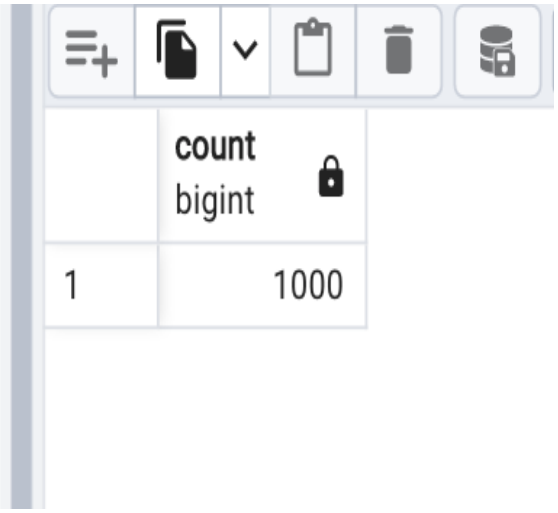
	id [PK] integer	student_reg_id integer	student_id integer	unit2 integer	unit3 integer	unit4 integer	unit5 integer
1	3	855	3	86	95	99	95
2	4	856	4	98	98	89	100
3	6	858	6	97	91	93	88
4	7	859	7	92	97	97	98
5	8	860	8	86	97	92	88
6	10	862	10	96	96	96	90
7	11	863	11	85	95	97	99
8	12	864	12	92	93	92	93
9	13	865	13	98	93	98	90
10	16	868	16	88	99	91	96
11	17	869	17	100	94	98	90
12	19	871	19	96	99	98	90
13	21	873	21	85	93	92	94
14	22	874	22	87	99	90	99
15	25	877	25	100	92	93	93
16	27	879	27	96	97	94	96
17	29	881	29	91	94	96	88
18	30	882	30	97	90	92	95
19	31	883	31	86	94	100	92
20	32	884	32	99	93	94	88

```
Open ended questions:
-- Take a closer look at the tables that you created and come up with 10 different scenarios/ questions and form SQL
-- Ask your colleagues

--select the distinct
select unit2
from student_marks
limit 5;
```



	unit2 integer
1	88
2	86
3	86
4	98
5	100

```
-- count

select count(unit4)

from student_marks
```

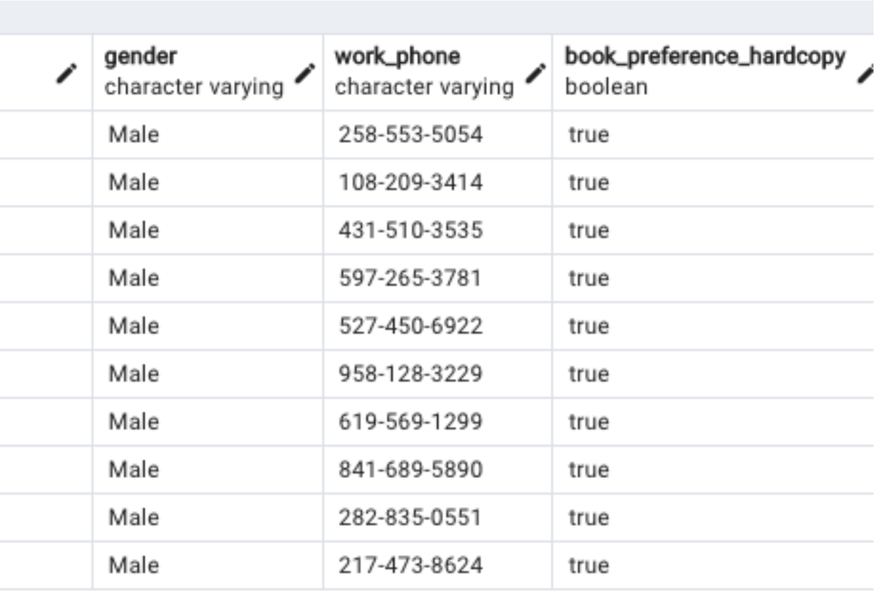


	count bigint
1	1000

--filetring rows

select * from student

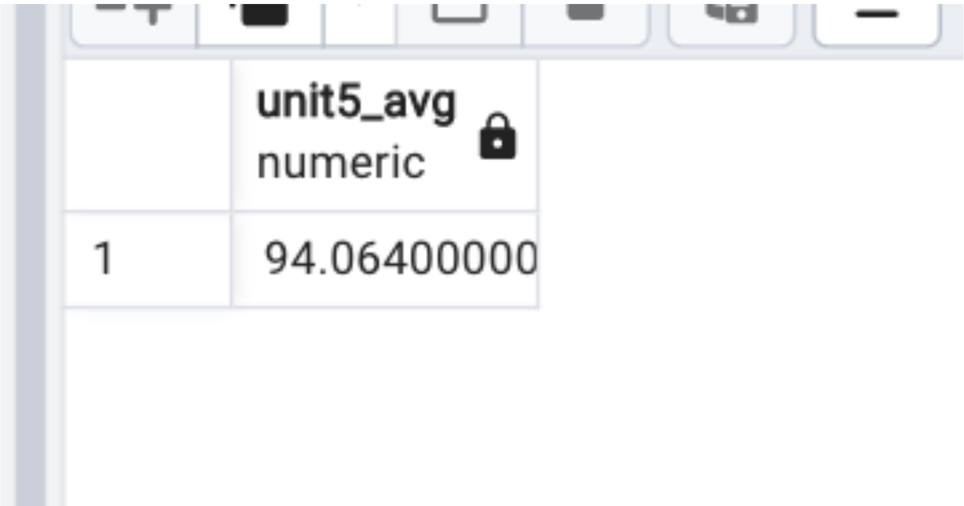
where gender ='Male' and book_preference_hardcopy='true'



	gender character varying	work_phone character varying	book_preference_hardcopy boolean
	Male	258-553-5054	true
	Male	108-209-3414	true
	Male	431-510-3535	true
	Male	597-265-3781	true
	Male	527-450-6922	true
	Male	958-128-3229	true
	Male	619-569-1299	true
	Male	841-689-5890	true
	Male	282-835-0551	true
	Male	217-473-8624	true

--find the avg

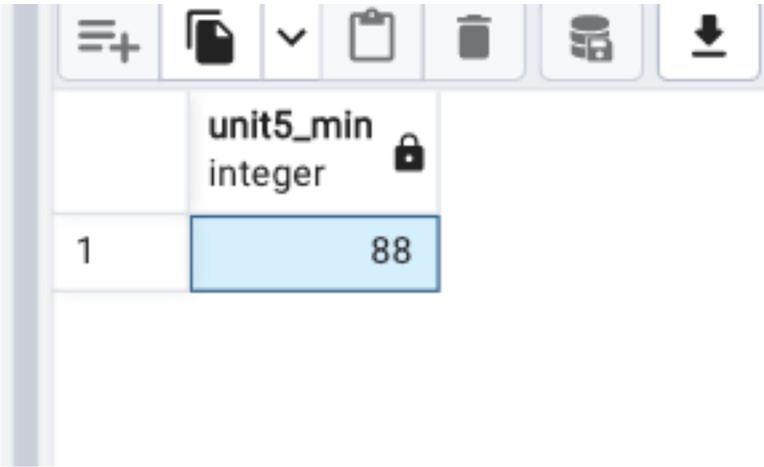
select avg(unit5) as unit5_avg from student_marks



	unit5_avg numeric
1	94.06400000

--find the min

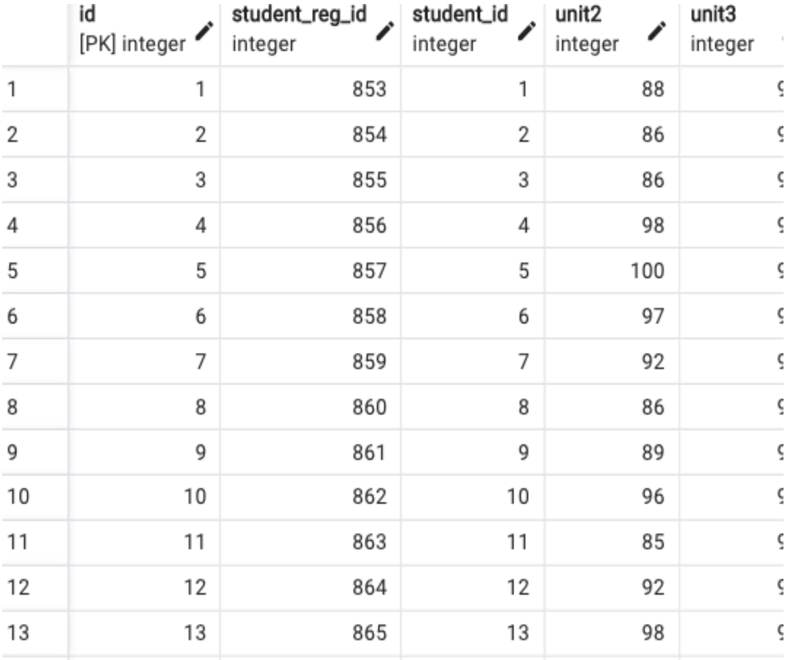
select min(unit5) as unit5_min from student_marks



	unit5_min integer
1	88

--sorting the single columns

select * from student_marks order by id



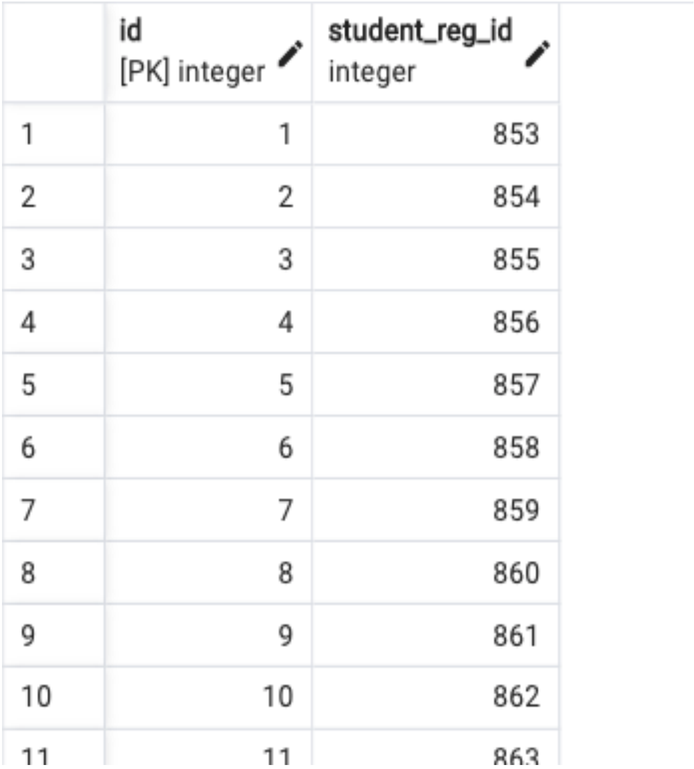
	id [PK] integer	student_reg_id integer	student_id integer	unit2 integer	unit3 integer
1	1	853	1	88	9
2	2	854	2	86	9
3	3	855	3	86	9
4	4	856	4	98	9
5	5	857	5	100	9
6	6	858	6	97	9
7	7	859	7	92	9
8	8	860	8	86	9
9	9	861	9	89	9
10	10	862	10	96	9
11	11	863	11	85	9
12	12	864	12	92	9
13	13	865	13	98	9

--sorting the multiple columns

select id , student_reg_id

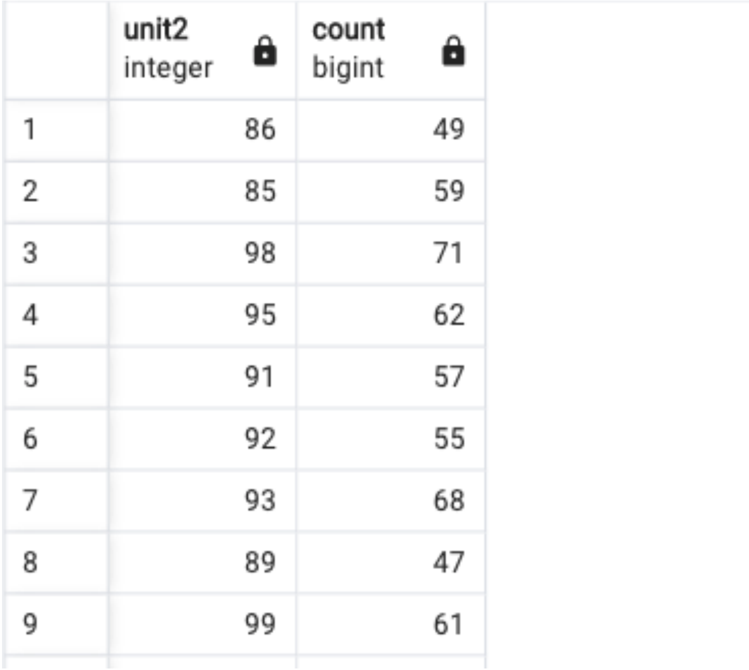
from student_marks

order by id, student_reg_id;



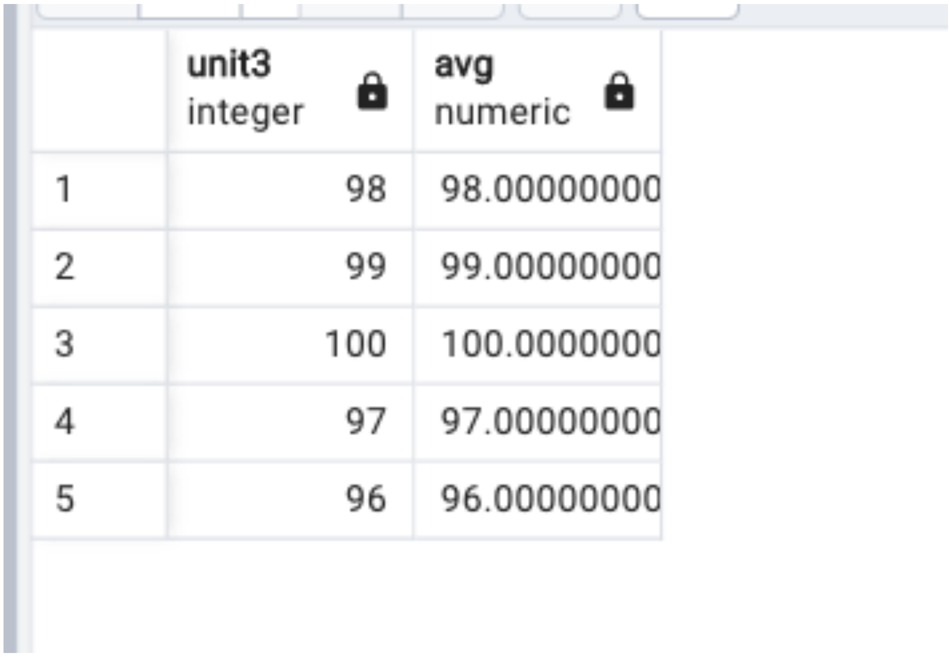
	id [PK] integer	student_reg_id integer
1	1	853
2	2	854
3	3	855
4	4	856
5	5	857
6	6	858
7	7	859
8	8	860
9	9	861
10	10	862
11	11	863

```
--show group by and count

select unit2, count(unit2) from student_marks

group by unit2
```



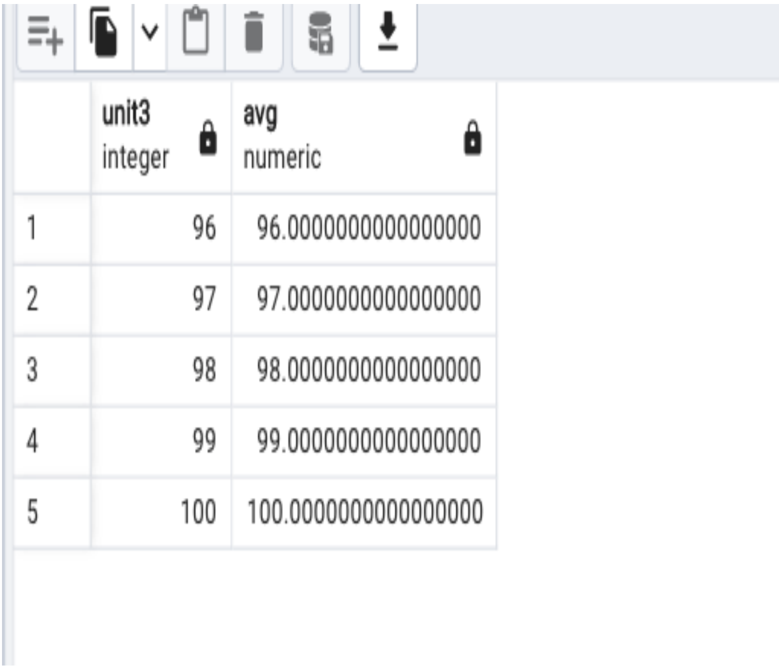
	unit2 integer	count bigint
1	86	49
2	85	59
3	98	71
4	95	62
5	91	57
6	92	55
7	93	68
8	89	47
9	99	61

```
--show the having function

select unit3, avg(unit3)

from student_marks

group by unit3

having avg(unit3)>95;
```



	unit3 integer	avg numeric
1	98	98.00000000
2	99	99.00000000
3	100	100.0000000
4	97	97.00000000
5	96	96.00000000

```
--show the order by

select unit3, avg(unit3)

from student_marks

group by unit3

having avg(unit3)>95

order by unit3
```



	unit3 integer	avg numeric
1	96	96.0000000000000000
2	97	97.0000000000000000
3	98	98.0000000000000000
4	99	99.0000000000000000
5	100	100.0000000000000000