实验三 数据查询（单表查询）

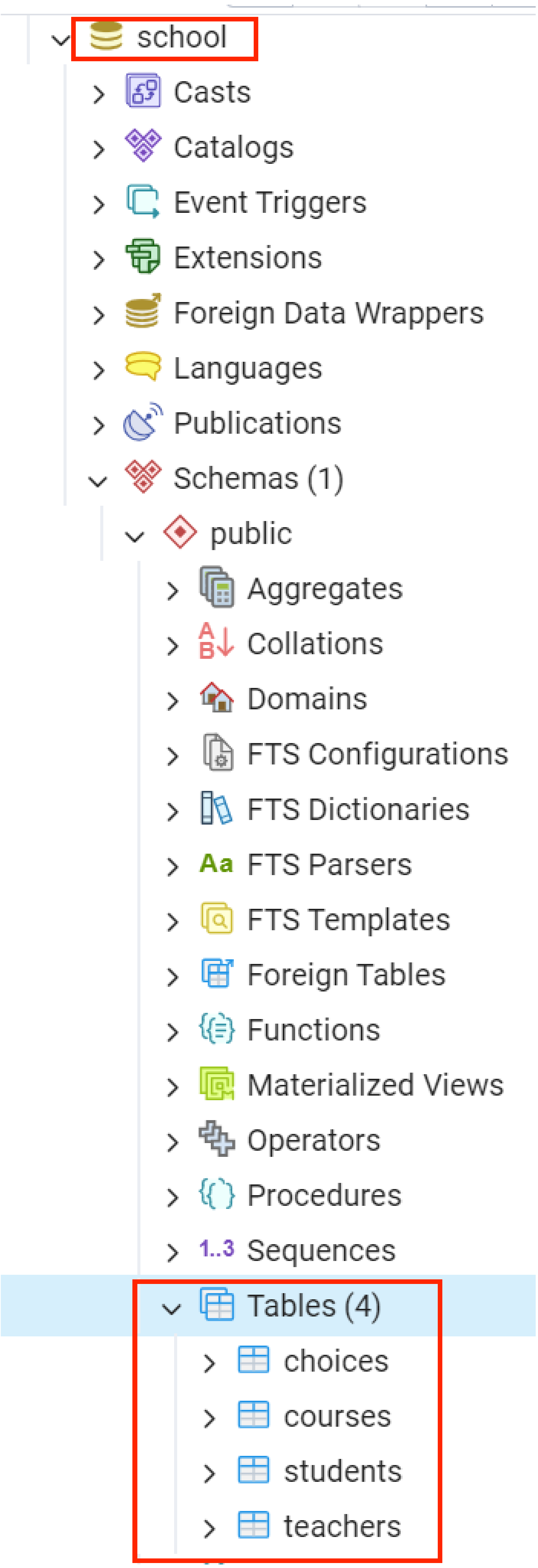
【姓名】谢宇桐　　【学号】 22336259　　【专业】计算机科学与技术

1.实验目的

熟悉SQL语句的数据查询语言，能够使用SQL语句对数据库进行单表查询。

2.实验环境

已安装完成Postgre SQL，在pgAdmin 4进行编辑。并已配置好schoo数据库的四张表格。



- school
  - Casts
  - Catalogs
  - Event Triggers
  - Extensions
  - Foreign Data Wrappers
  - Languages
  - Publications
  - Schemas (1)
    - public
      - Aggregates
      - Collations
      - Domains
      - FTS Configurations
      - FTS Dictionaries
      - FTS Parsers
      - FTS Templates
      - Foreign Tables
      - Functions
      - Materialized Views
      - Operators
      - Procedures
      - Sequences
      - Tables (4)
        - choices
        - courses
        - students
        - teachers

3.实验内容

本节实验的主要内容包括：

- ● 查询的目标表达式为所有列、指定列或指定列的运算。
- ● 使用DISTINCT保留字消除重复行。
- ● 对查询结果排序和分组。
- ● 集合分组使用集函数进行各项统计。

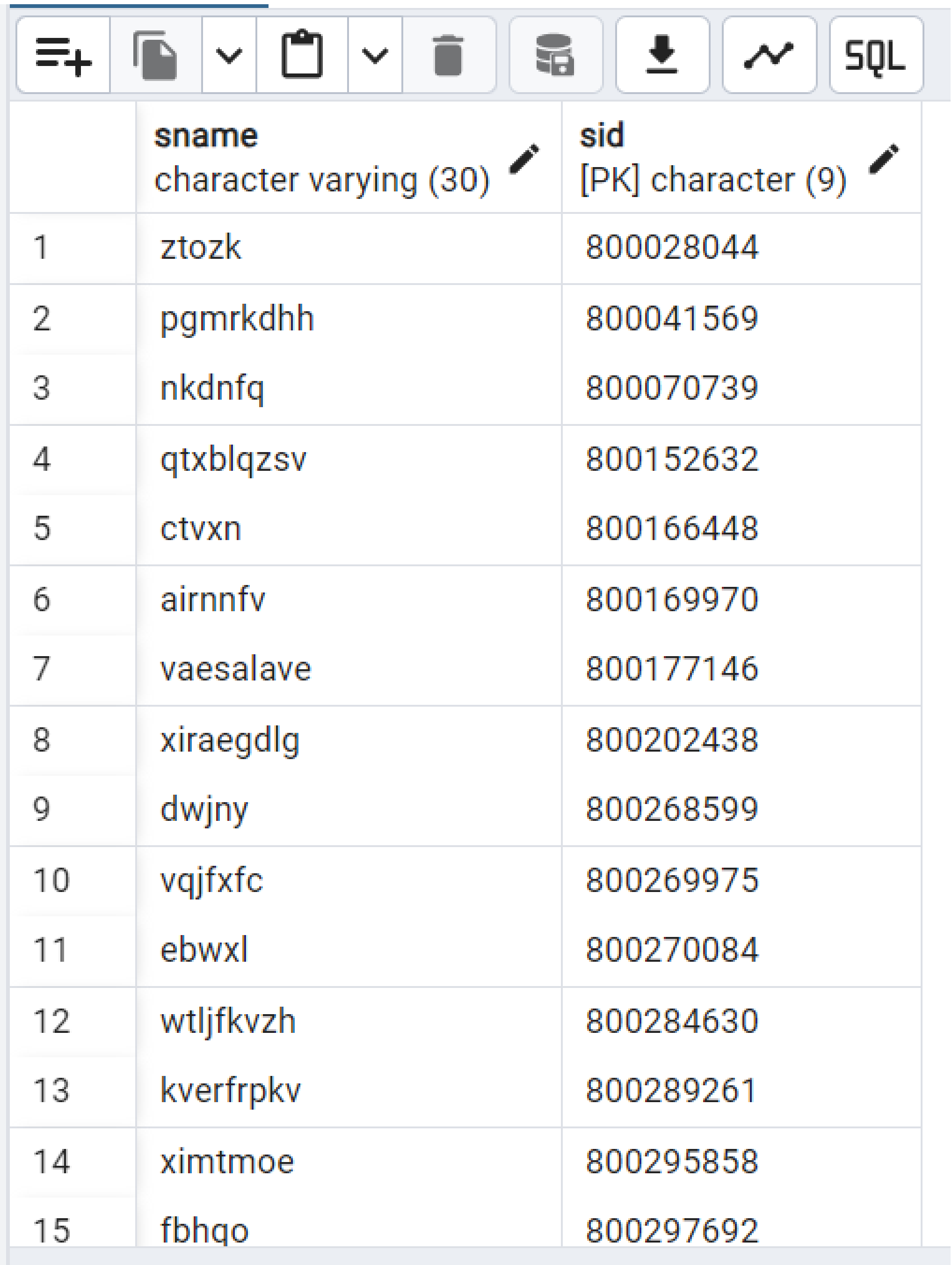
4.实验步骤（遇到的问题为灰色字和注释）

在数据库中，存在这样的关系：学生可以选择课程。一个课程对应一个教师。在表CHOICES中保存学生的选课记录。

(1) 查询年级为2001的所有学生的名称，按编号升序排列

SQL中空格很重要，若空格不规范则会报错

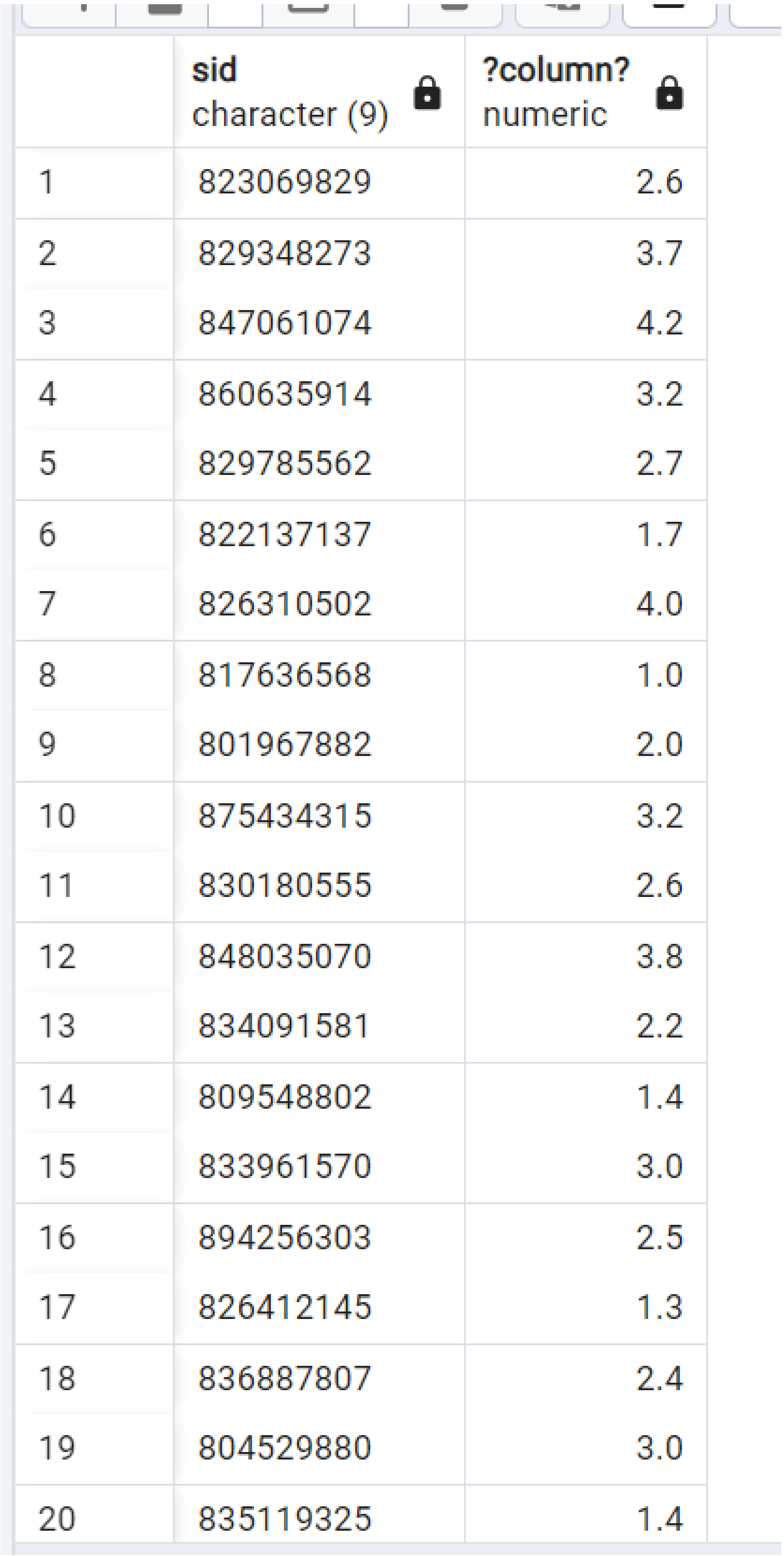
```sql
SELECT sname, sid
FROM STUDENTS
WHERE grade='2001'
ORDER BY sid;
```



	sname character varying (30)	sid [PK] character (9)
1	ztozk	800028044
2	pgmrkdhh	800041569
3	nkdnfq	800070739
4	qtxblqzsv	800152632
5	ctvxn	800166448
6	airnnfv	800169970
7	vaesalave	800177146
8	xiraegdlg	800202438
9	dwjny	800268599
10	vqjfxfc	800269975
11	ebwxl	800270084
12	wtljfkvzh	800284630
13	kverfrpkv	800289261
14	ximtmoe	800295858
15	fbhqo	800297692

(2) 查询学生的选课成绩合格的课程成绩，并把成绩换算为积点(60分对应积点为1,每增加1分，积点增加0.1);

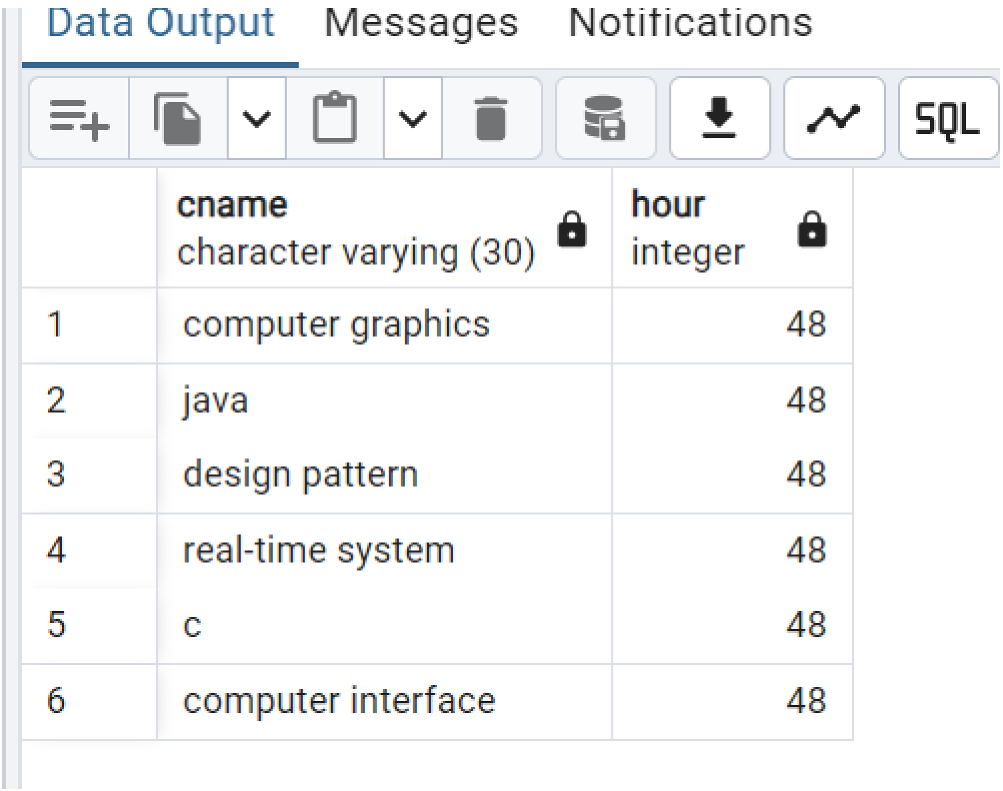
```
SELECT sid,(score-60)*0.1+1
FROM CHOICES
WHERE score>=60;
```



	sid character (9)	?column? numeric
1	823069829	2.6
2	829348273	3.7
3	847061074	4.2
4	860635914	3.2
5	829785562	2.7
6	822137137	1.7
7	826310502	4.0
8	817636568	1.0
9	801967882	2.0
10	875434315	3.2
11	830180555	2.6
12	848035070	3.8
13	834091581	2.2
14	809548802	1.4
15	833961570	3.0
16	894256303	2.5
17	826412145	1.3
18	836887807	2.4
19	804529880	3.0
20	835119325	1.4

(3) 查询课时是48或64的课程的名称；

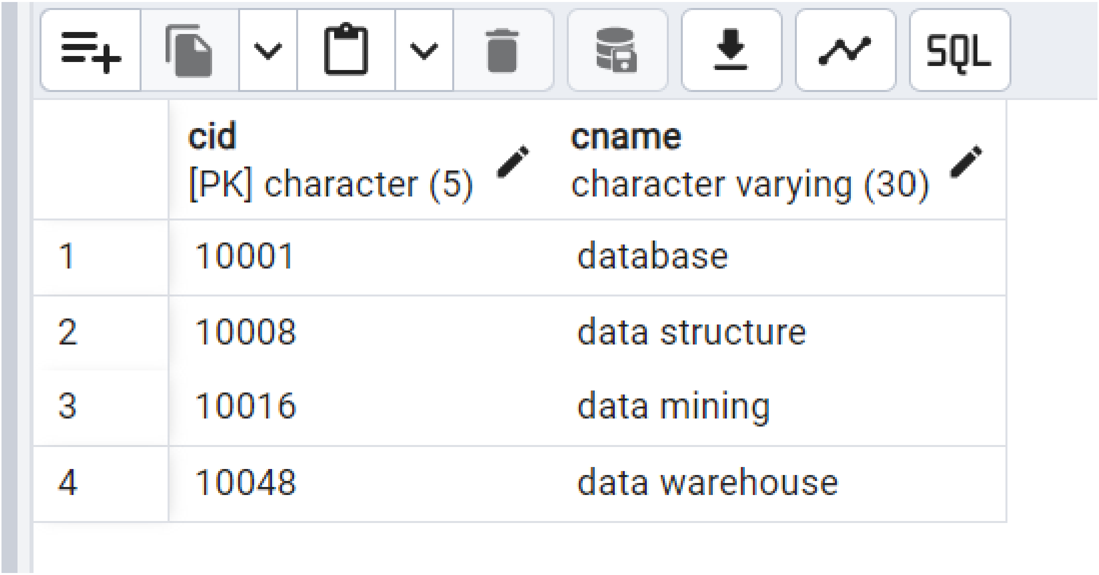
```
SELECT cname,hour
FROM COURSES
WHERE hour=48 or hour=64;
```



	cname character varying (30)	hour integer
1	computer graphics	48
2	java	48
3	design pattern	48
4	real-time system	48
5	c	48
6	computer interface	48

(4) 查询所有课程名称中含有data的课程编号；

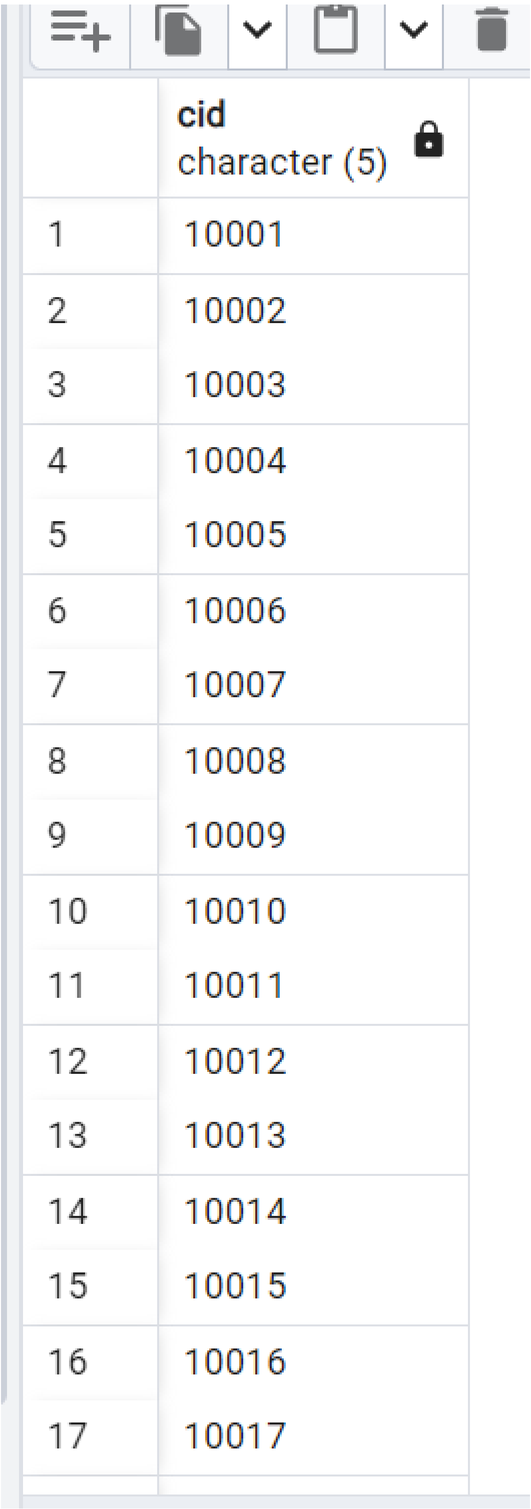
```
SELECT cid,cname
FROM COURSES
WHERE cname like '%data%';
```



	cid [PK] character (5)	cname character varying (30)
1	10001	database
2	10008	data structure
3	10016	data mining
4	10048	data warehouse

(5) 查询所有选课记录的课程号(不重复显示)；

```
SELECT DISTINCT cid
FROM CHOICES;
```

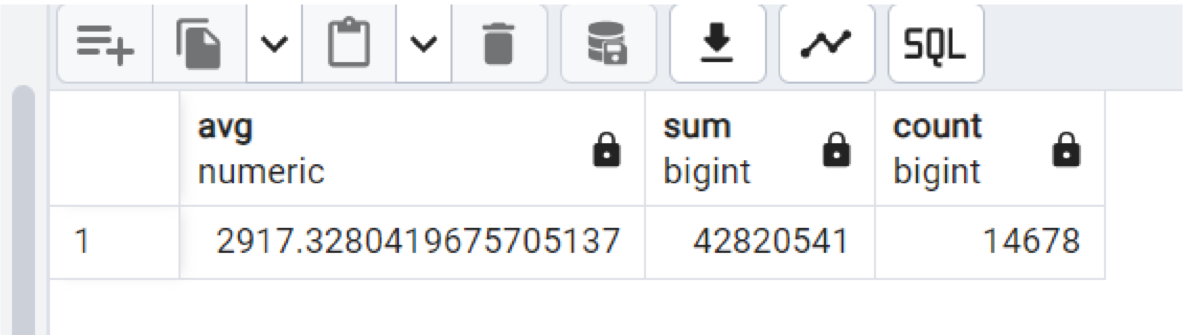


	cid character (5)
1	10001
2	10002
3	10003
4	10004
5	10005
6	10006
7	10007
8	10008
9	10009
10	10010
11	10011
12	10012
13	10013
14	10014
15	10015
16	10016
17	10017

可以注意到distinct的作用不只是去重，也有按顺序排列的作用

(6) 统计所有老师的平均工资；

```
SELECT AVG(salary),SUM(salary),COUNT(salary)
FROM TEACHERS;
```

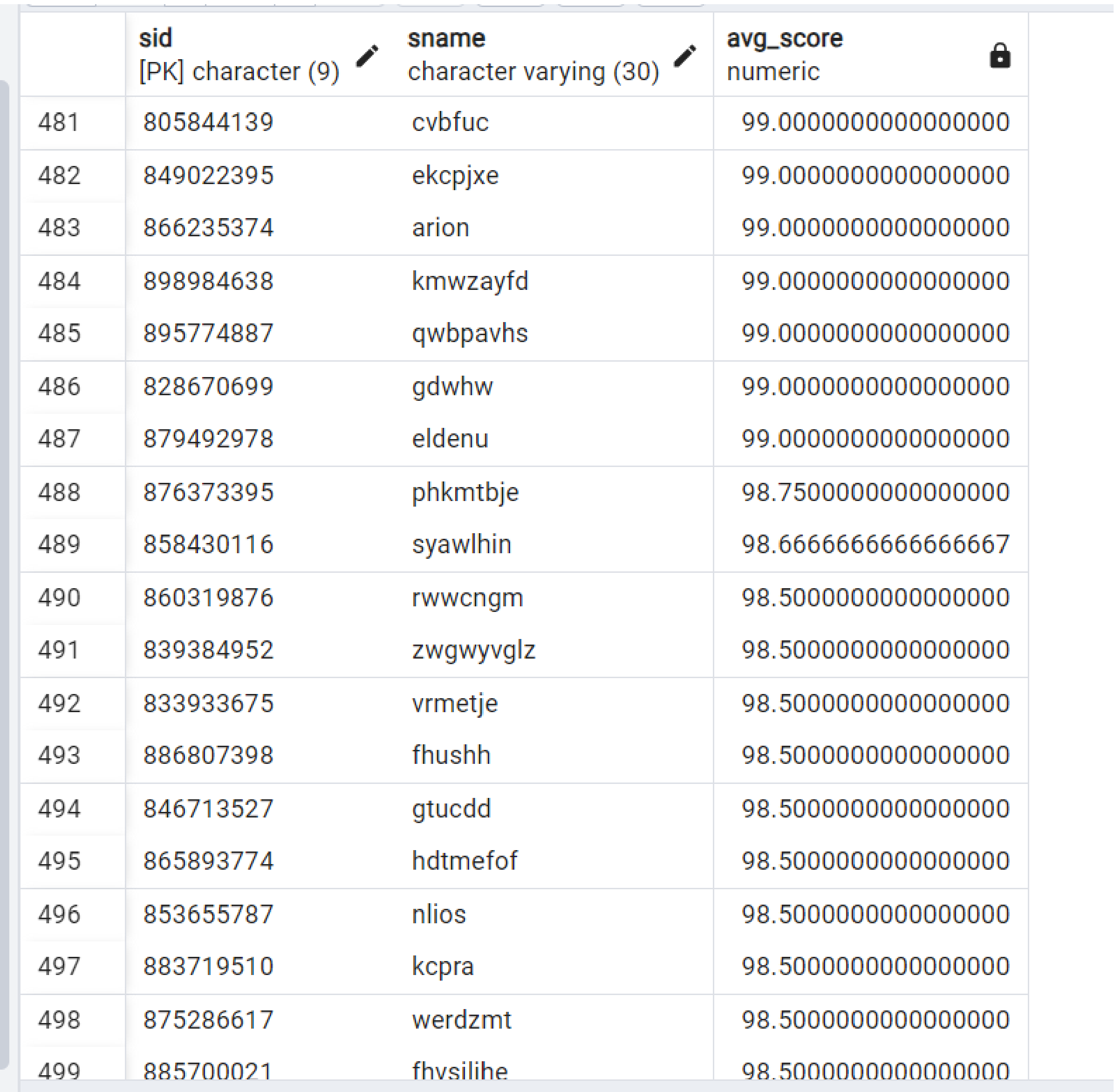


	avg numeric	sum bigint	count bigint
1	2917.3280419675705137	42820541	14678

SUM和COUNT用来验证结果是否正确，验证得到是正确的。

(7) 查询所有学生的编号，姓名和平均成绩，按总平均成绩降序排列；

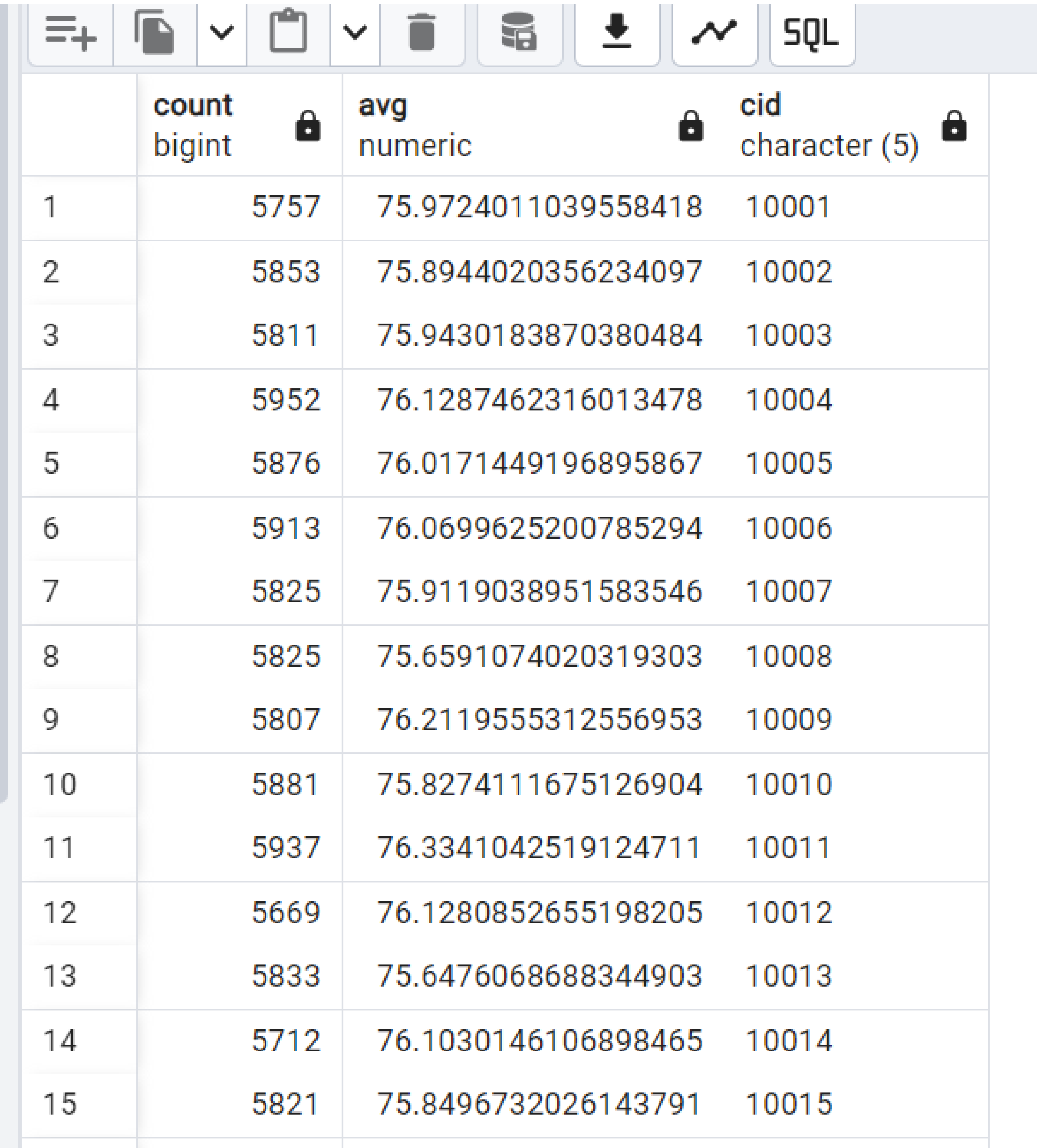
```
SELECT STUDENTS.sid, sname, AVG(CHOICES.score) AS avg_score
FROM STUDENTS, CHOICES
WHERE CHOICES.sid = STUDENTS.sid
GROUP BY STUDENTS.sid, sname
HAVING AVG(CHOICES.score) IS NOT NULL --之前我使用的语句没有这一句，这会使显示出来的avg_score为NULL，原因应为学生没有选的课成绩为NULL，不排除未选的课算出来的AVG就为NULL，因此需要加上这一句把分数为NULL的课排除掉，即可选出他们的平均成绩
ORDER BY avg_score DESC;
```



	sid [PK] character (9)	sname character varying (30)	avg_score numeric
481	805844139	cvbfuc	99.0000000000000000
482	849022395	ekcpjxe	99.0000000000000000
483	866235374	arion	99.0000000000000000
484	898984638	kmwzayfd	99.0000000000000000
485	895774887	qwbpavhs	99.0000000000000000
486	828670699	gdwhw	99.0000000000000000
487	879492978	eldenu	99.0000000000000000
488	876373395	phkmtbje	98.7500000000000000
489	858430116	syawlhin	98.6666666666666667
490	860319876	rwwcngm	98.5000000000000000
491	839384952	zwgwyvglz	98.5000000000000000
492	833933675	vrmetje	98.5000000000000000
493	886807398	fhushh	98.5000000000000000
494	846713527	gtucdd	98.5000000000000000
495	865893774	hdtmefof	98.5000000000000000
496	853655787	nlios	98.5000000000000000
497	883719510	kcpra	98.5000000000000000
498	875286617	werdzmt	98.5000000000000000
499	885700021	fhvsilihe	98.5000000000000000

(8) 统计各个课程的选课人数和平均成绩；

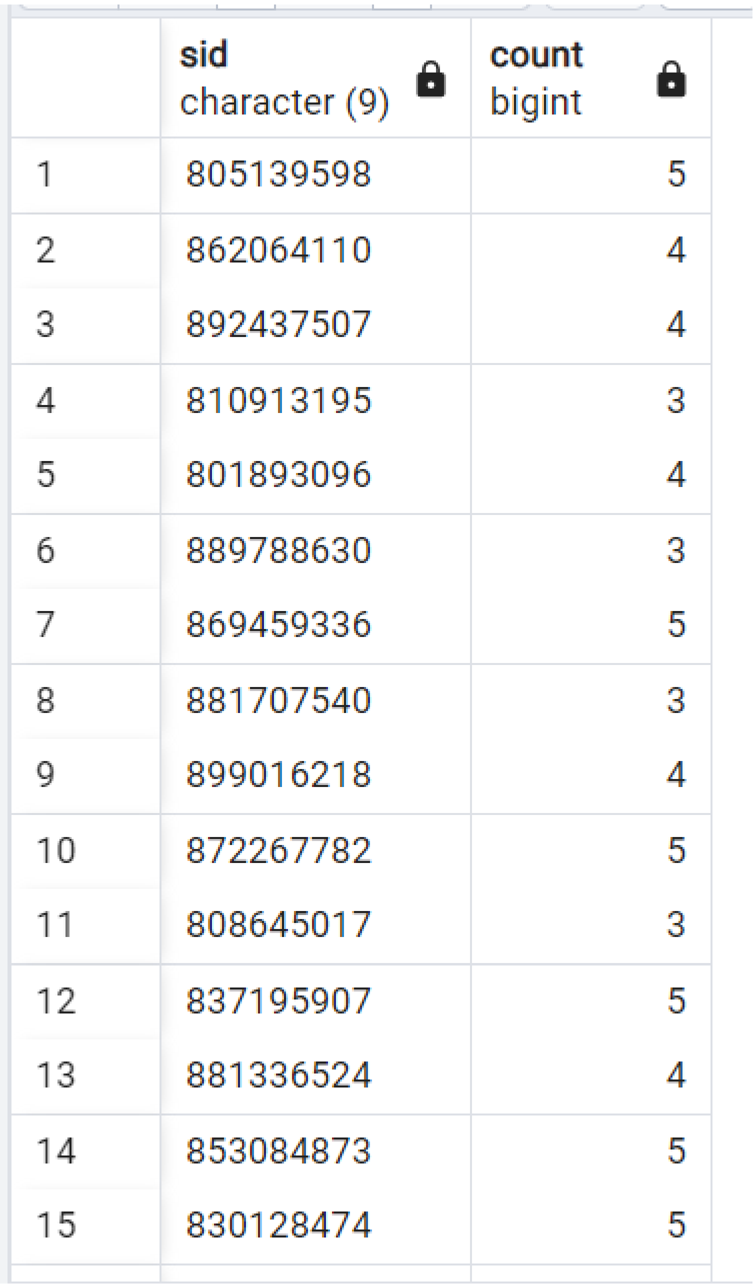
```
SELECT COUNT(distinct sid),AVG(score), cid
FROM CHOICES
GROUP BY cid;
```



	count bigint	avg numeric	cid character (5)
1	5757	75.9724011039558418	10001
2	5853	75.8944020356234097	10002
3	5811	75.9430183870380484	10003
4	5952	76.1287462316013478	10004
5	5876	76.0171449196895867	10005
6	5913	76.0699625200785294	10006
7	5825	75.9119038951583546	10007
8	5825	75.6591074020319303	10008
9	5807	76.2119555312556953	10009
10	5881	75.8274111675126904	10010
11	5937	76.3341042519124711	10011
12	5669	76.1280852655198205	10012
13	5833	75.6476068688344903	10013
14	5712	76.1030146106898465	10014
15	5821	75.8496732026143791	10015

(9) 查询至少选修了三门课程的学生编号。

```
SELECT sid,COUNT(cid)
FROM CHOICES
GROUP BY sid
HAVING COUNT(cid) >= 3;
```



	sid character (9)	count bigint
1	805139598	5
2	862064110	4
3	892437507	4
4	810913195	3
5	801893096	4
6	889788630	3
7	869459336	5
8	881707540	3
9	899016218	4
10	872267782	5
11	808645017	3
12	837195907	5
13	881336524	4
14	853084873	5
15	830128474	5

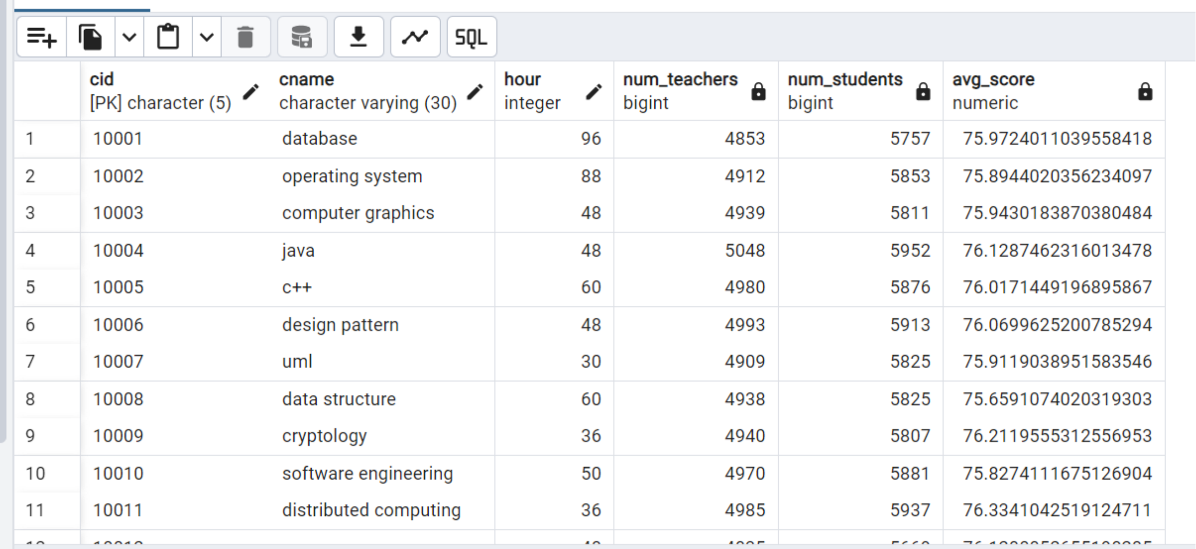
至此，课内实验成功完成。

5.“自我实践”实验步骤

(1) 查询全部课程的详细记录；

```
SELECT COURSES.cid,
       COURSES.cname,
       COURSES.hour,
       COUNT(DISTINCT TEACHERS.tid) AS num_teachers,
       COUNT(DISTINCT STUDENTS.sid) AS num_students,
       AVG(CHOICES.score) AS avg_score
FROM COURSES
JOIN CHOICES ON COURSES.cid = CHOICES.cid
LEFT JOIN STUDENTS ON CHOICES.sid = STUDENTS.sid
LEFT JOIN TEACHERS ON CHOICES.tid = TEACHERS.tid
GROUP BY COURSES.cid, COURSES.cname, COURSES.hour;
```

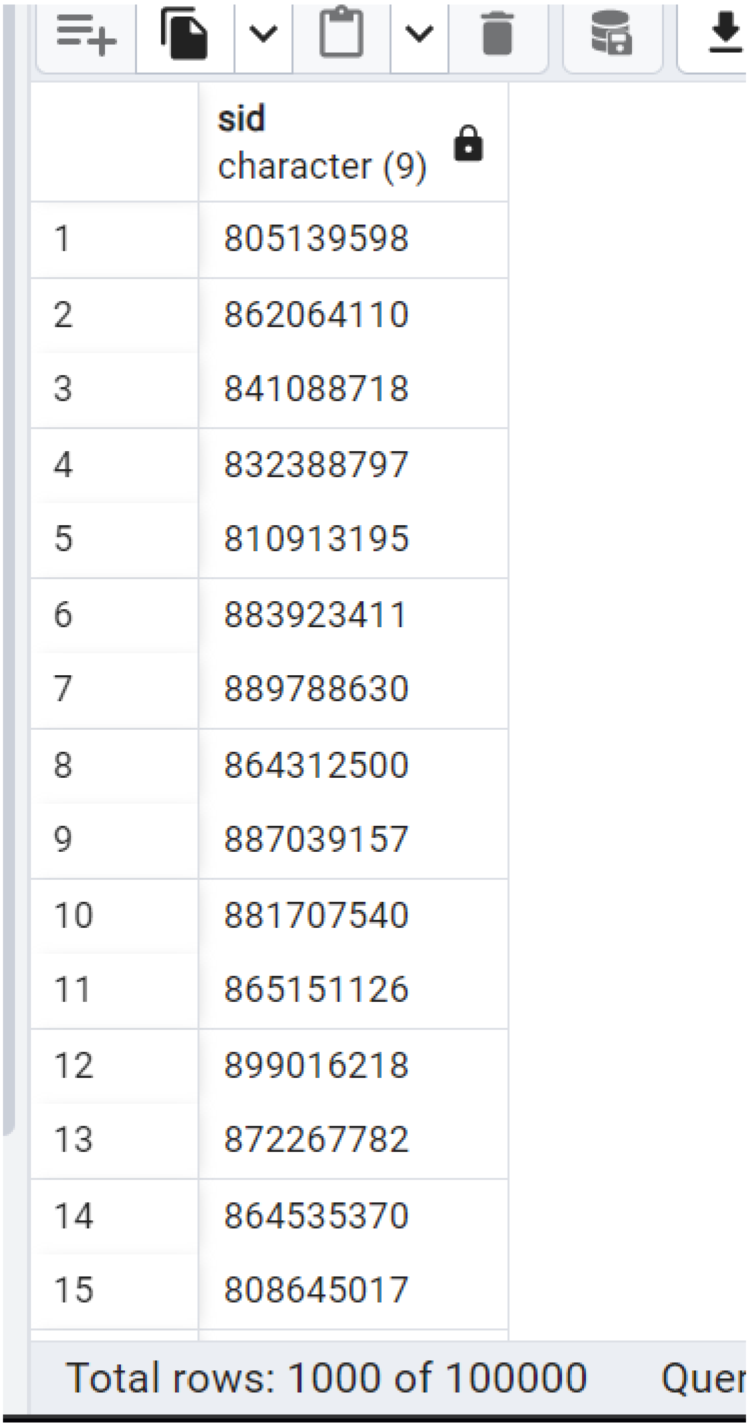
这个代码显示了课程的编号，名称，课时时长，教师人数，选课学生人数以及平均成绩。



	cid [PK] character (5)	cname character varying (30)	hour integer	num_teachers bigint	num_students bigint	avg_score numeric
1	10001	database	96	4853	5757	75.9724011039558418
2	10002	operating system	88	4912	5853	75.8944020356234097
3	10003	computer graphics	48	4939	5811	75.9430183870380484
4	10004	java	48	5048	5952	76.1287462316013478
5	10005	c++	60	4980	5876	76.0171449196895867
6	10006	design pattern	48	4993	5913	76.0699625200785294
7	10007	uml	30	4909	5825	75.9119038951583546
8	10008	data structure	60	4938	5825	75.6591074020319303
9	10009	cryptology	36	4940	5807	76.2119555312556953
10	10010	software engineering	50	4970	5881	75.8274111675126904
11	10011	distributed computing	36	4985	5937	76.3341042519124711

(2) 查询所有有选修课的学生的编号；

```
SELECT DISTINCT sid
FROM CHOICES;
```

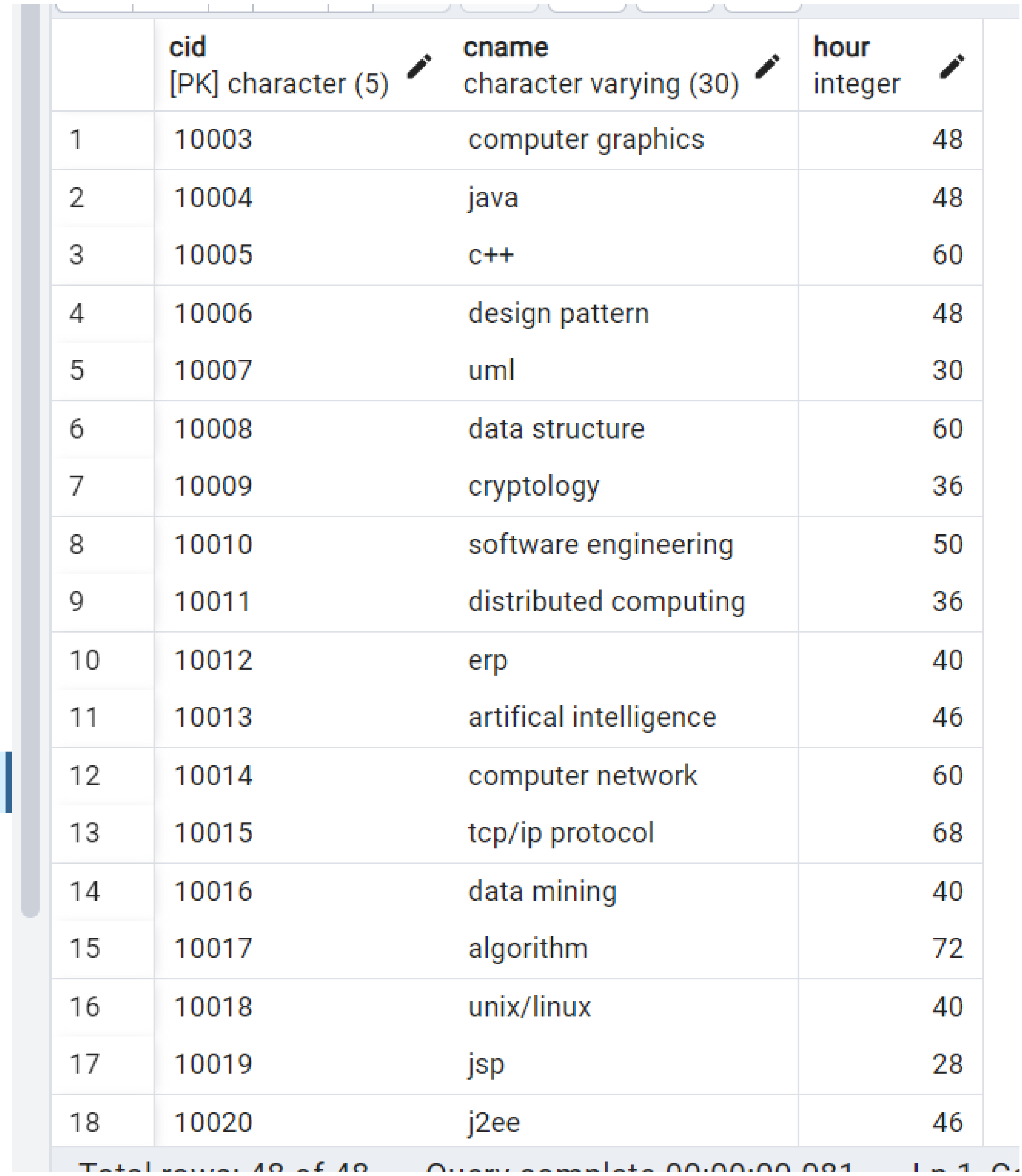


	sid character (9)
1	805139598
2	862064110
3	841088718
4	832388797
5	810913195
6	883923411
7	889788630
8	864312500
9	887039157
10	881707540
11	865151126
12	899016218
13	872267782
14	864535370
15	808645017

Total rows: 1000 of 100000 Quer

(3) 查询课时<88(小时)的课程的编号；

```
SELECT cid,cname,hour
FROM COURSES
WHERE hour<88;
```



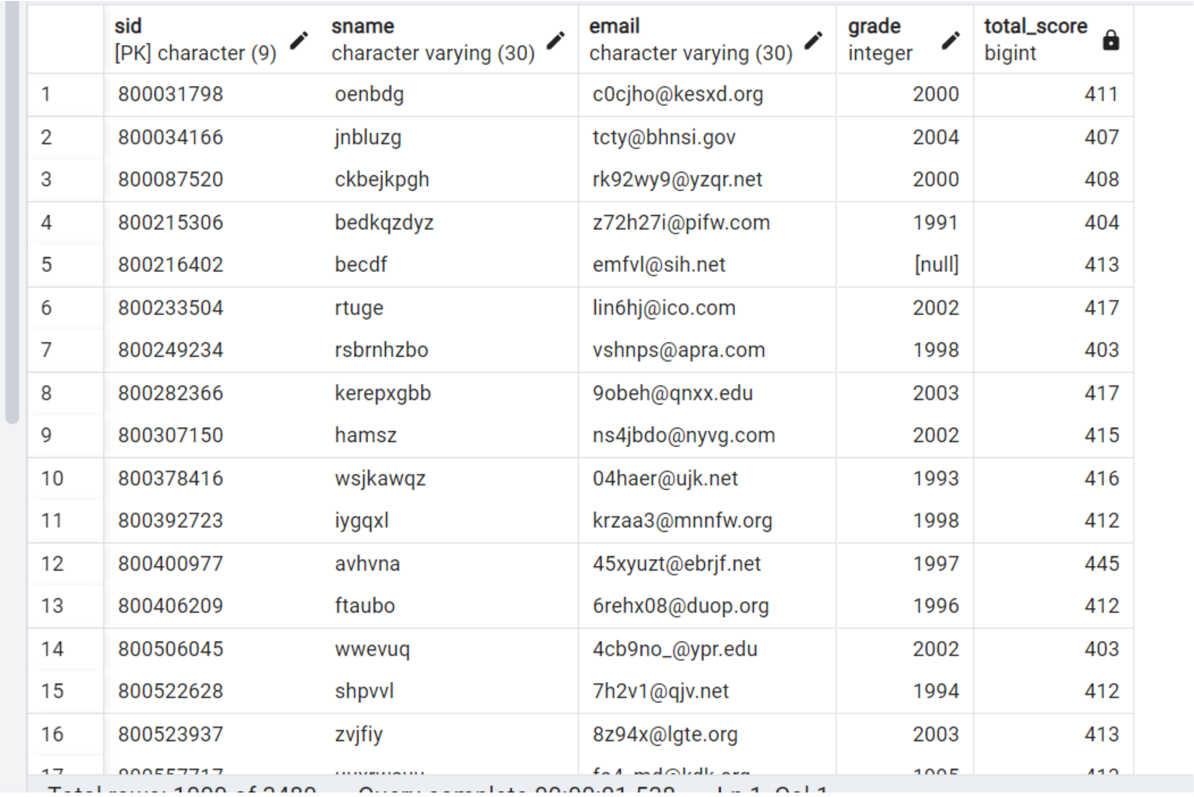
	cid [PK] character (5)	cname character varying (30)	hour integer
1	10003	computer graphics	48
2	10004	java	48
3	10005	c++	60
4	10006	design pattern	48
5	10007	uml	30
6	10008	data structure	60
7	10009	cryptology	36
8	10010	software engineering	50
9	10011	distributed computing	36
10	10012	erp	40
11	10013	artifical intelligence	46
12	10014	computer network	60
13	10015	tcp/ip protocol	68
14	10016	data mining	40
15	10017	algorithm	72
16	10018	unix/linux	40
17	10019	jsp	28
18	10020	j2ee	46

Total rows: 48 of 48 Query complete 00:00:00.081 Ln 1, C

(4) 请找出总分超过400分的学生；

***不能用在GROUP BY 语句中**

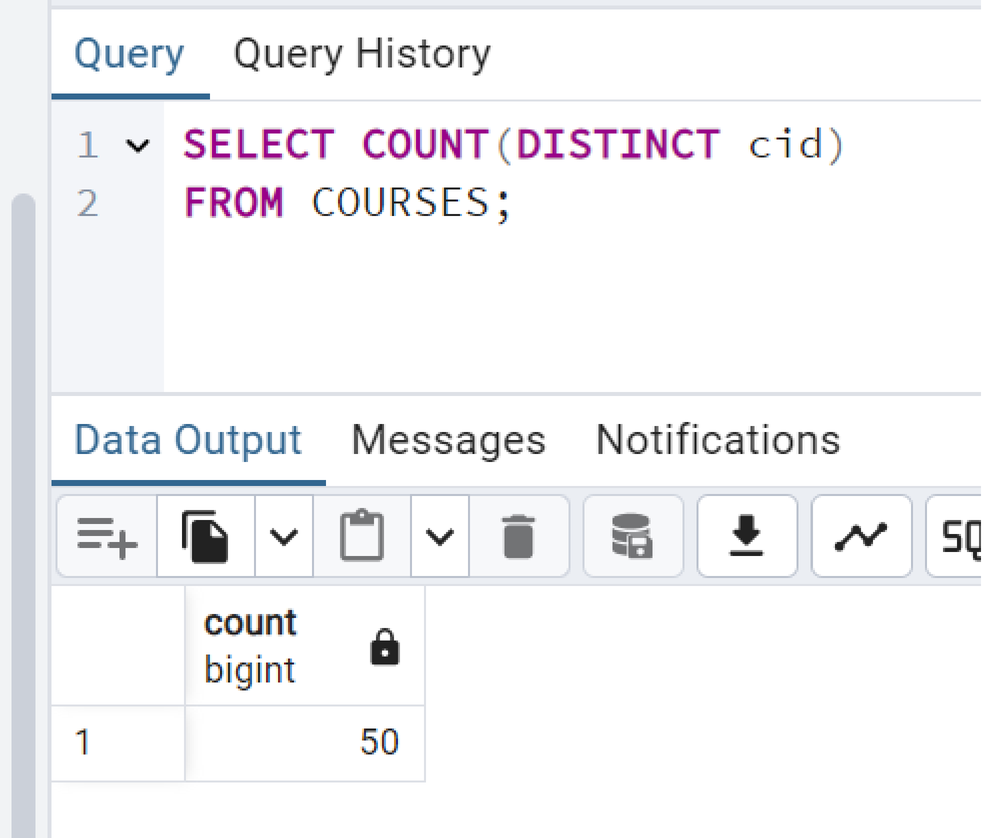
```
SELECT STUDENTS.*,SUM(CHOICES.score) AS total_score
FROM STUDENTS
JOIN CHOICES ON STUDENTS.sid = CHOICES.sid
GROUP BY STUDENTS.sid
HAVING SUM(CHOICES.score) > 400;
```



	sid [PK] character (9)	sname character varying (30)	email character varying (30)	grade integer	total_score bigint
1	800031798	oenbdg	c0cjho@kesxd.org	2000	411
2	800034166	jnbluzg	tcty@bhnsi.gov	2004	407
3	800087520	ckbejkpgh	rk92wy9@yzqr.net	2000	408
4	800215306	bedkqzdyz	z72h27i@pifw.com	1991	404
5	800216402	becdf	emfvl@sih.net	[null]	413
6	800233504	rtuge	lin6hj@ico.com	2002	417
7	800249234	rsbrnhzbo	vshnps@apra.com	1998	403
8	800282366	kerepxgbb	9obeh@qnxx.edu	2003	417
9	800307150	hamsz	ns4jbdo@nyvg.com	2002	415
10	800378416	wsjkawqz	04haer@ujk.net	1993	416
11	800392723	iygqxl	krzaa3@mnnfw.org	1998	412
12	800400977	avhvna	45xyuzt@ebrjf.net	1997	445
13	800406209	ftaubo	6rehx08@duop.org	1996	412
14	800506045	wwevuq	4cb9no_@ypr.edu	2002	403
15	800522628	shpvvl	7h2v1@qjv.net	1994	412
16	800523937	zvjfiy	8z94x@lgte.org	2003	413

(5) 查询课程的总数；

```sql
SELECT COUNT(DISTINCT cid)
FROM COURSES;
```



```sql
SELECT COUNT(DISTINCT cid)
FROM COURSES;
```

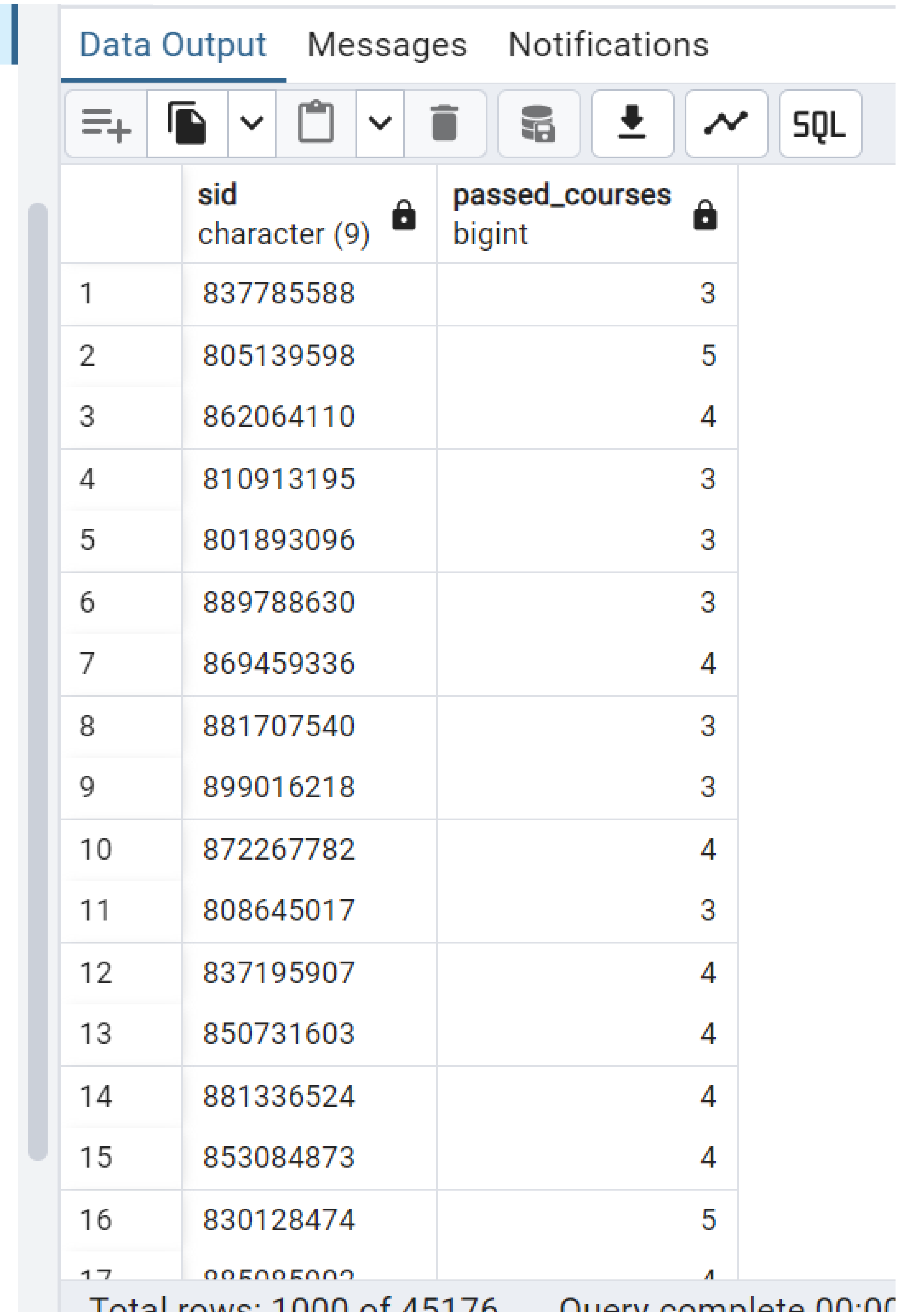
	count bigint
1	50

(6) 查询所有课程和选修该课程的学生总数；

此步骤我在第一步已进行展示，这里不再进行赘述

(7) 查询选修成绩合格的课程超过两门的学生编号；

```sql
SELECT sid,COUNT(*) AS passed_courses
FROM CHOICES
WHERE score >= 60
GROUP BY sid
HAVING COUNT(*) > 2;
```

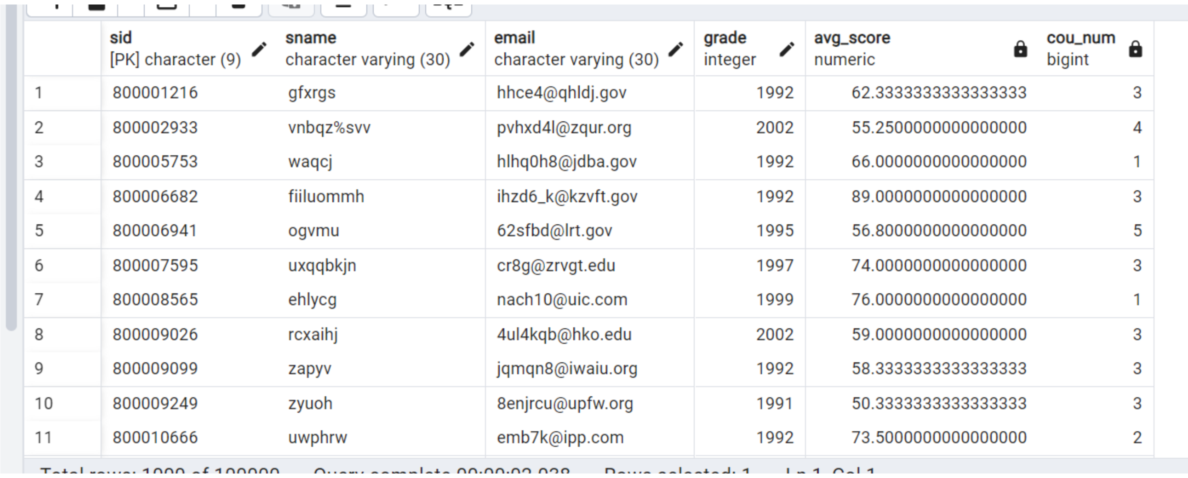


	sid character (9)	passed_courses bigint
1	837785588	3
2	805139598	5
3	862064110	4
4	810913195	3
5	801893096	3
6	889788630	3
7	869459336	4
8	881707540	3
9	899016218	3
10	872267782	4
11	808645017	3
12	837195907	4
13	850731603	4
14	881336524	4
15	853084873	4
16	830128474	5
17	885085002	4

Total rows: 1000 of 45176   Query complete 00:00

(8) 统计各个学生的选修课程数目和平均成绩。

```
SELECT STUDENTS.*, AVG(COALESCE(score, 0)) AS avg_score, COUNT(CHOICES.*) AS
cou_num
FROM STUDENTS, CHOICES
WHERE CHOICES.sid = STUDENTS.sid
GROUP BY STUDENTS.sid;
```



	sid [PK] character (9)	sname character varying (30)	email character varying (30)	grade integer	avg_score numeric	cou_num bigint
1	800001216	gfxrgs	hhce4@qhldj.gov	1992	62.3333333333333333	3
2	800002933	vnbqz%svv	pvhxd4l@zqur.org	2002	55.2500000000000000	4
3	800005753	waqcj	hlhq0h8@jdba.gov	1992	66.0000000000000000	1
4	800006682	fiiluommh	ihzd6_k@kzvft.gov	1992	89.0000000000000000	3
5	800006941	ogvmu	62sfbd@lrt.gov	1995	56.8000000000000000	5
6	800007595	uxqqbkjn	cr8g@zrvgt.edu	1997	74.0000000000000000	3
7	800008565	ehlycg	nach10@uic.com	1999	76.0000000000000000	1
8	800009026	rcxaihj	4ul4kqb@hko.edu	2002	59.0000000000000000	3
9	800009099	zapyv	jqmqn8@iwaiu.org	1992	58.3333333333333333	3
10	800009249	zyuoh	8enjrcu@upfw.org	1991	50.3333333333333333	3
11	800010666	uwphrw	emb7k@ipp.com	1992	73.5000000000000000	2

Total rows: 1000 of 100000 Query complete 00:00:02.038 Rows selected: 1 Ln 1, Col 1

在这里因为成绩包含NULL 值，若将这些 NULL 值视为 0，使用 COALESCE 函数将 NULL 转换为 0。