



# 实验报告

实验（二）

姓 名	熊恪峥
学 号	22920202204622
日 期	2023年4月3日
学 院	信息学院
课程名称	数据库

## 实验（二）

### 目录

1	实验2.1	1
2	实验2.2	7

## 1 实验2.1

1. 查询年级为2001的所有学生的名称并按编号升序排列。

```
select * from STUDENTS where grade=2001 order by sid asc
```

图 1: 运行结果

	sid	sname	email	grade
1	800028044	ztozk	r36919a@laykh.gov	2001
2	800041569	pgarkdth	xpql2vc@urjtp.edu	2001
3	800070739	nkdnfq	pio7n@sci.com	2001
4	800152632	qtxblqzsv	tdvtn@nau.edu	2001
5	800166448	ctvxn	5b6zrz@pplh.org	2001
6	800169970	airrnfv	ytkyo@xeh.org	2001
7	800177146	vaasalave	tlafdd@ypzxr.gov	2001
8	800202438	xiraegdlg	_fov_i@hmc.edu	2001
9	800268599	dvjny	9pax@ejlk.com	2001
10	800269975	vqjxfic	i5w9ba@spu.com	2001
11	800270084	ebxrl	vp02q@ekze.gov	2001
12	800284630	wtljfivzh	cc33gi@tqs.org	2001
13	800289261	kverfrpkv	3if97lp@tzn.net	2001
14	800295858	xlatasa	e3.us@aki.com	2001

	sid	cid	score	gpa
1	823069829	10037	76	2
2	829348273	10010	87	3
3	847061074	10025	92	4
4	860635914	10039	82	3
5	829785562	10028	77	2
6	822137137	10011	67	1
7	826310502	10005	90	4
8	817636568	10047	60	1
9	801967882	10021	70	2
10	875434315	10048	82	3
11	830180555	10016	76	2
12	848035070	10007	88	3
13	834091581	10049	72	2
14	809548802	10002	64	1

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）。

思路：使用CASE语句分情况讨论，将60一下映射为0，60以上映射为(score-50)/10，即1-5。

```
SELECT sid, cid, score, CASE WHEN score >= 60 THEN (score - 50) / 10 ELSE 0 END
AS gpa FROM CHOICES WHERE score >= 60;
```

图 2: 运行结果

	cname
1	computer graphics
2	java
3	design pattern
4	real-time system
5	c
6	computer interface

	cid
1	10001
2	10008

3. 查询课时是48或64的课程名称。

```
SELECT cname FROM COURSES WHERE hour IN (48, 64);
```

4. 查询所有课程名称中含有data的课程编号。

思路：使用通配符%，表示任意多个字符。

```
SELECT cid FROM COURSES WHERE cname LIKE '%data%';
```

5. 查询所有选课记录的课程号（不重复显示）。

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

6. 统计所有教师的平均工资。

图 3: 运行结果

	cid
1	10008
2	10019
3	10018
4	10040
5	10011
6	10028
7	10035

	avg_salary
1	3417

图 4: 运行结果

	tid	avg_score
1	259380467	90
2	272507136	89
3	241568555	89
4	266524091	88
5	265299360	88
6	260955336	88
7	294109744	88

(无列名)	(无列名)
1	5985 75
2	6074 75
3	5969 76
4	6102 75
5	6086 76
6	6042 75
7	6104 76
8	5916 76
9	6069 76

```
SELECT AVG(salary) AS avg_salary FROM TEACHERS;
```

7. 查询所有教师的编号及选修其课程的学生们的平均成绩，按平均成绩降序排列。

```
SELECT TEACHERS.tid, AVG(CHOICES.score) AS avg_score FROM TEACHERS JOIN CHOICES
ON TEACHERS.tid = CHOICES.tid JOIN STUDENTS ON STUDENTS.sid = CHOICES.sid GROUP
BY TEACHERS.tid ORDER BY avg_score desc
```

图 5: 运行结果

100 %	
结果	消息
	cid
1	10008
2	10019
3	10018
4	10040
5	10011
6	10028
7	10035
8	10021

100 %	
结果	消息
	sid
1	893821981

8. 统计各个课程的选课人数和平均成绩。

```
SELECT COUNT(*),AVG(CHOICES.score) from CHOICES GROUP BY cid
```

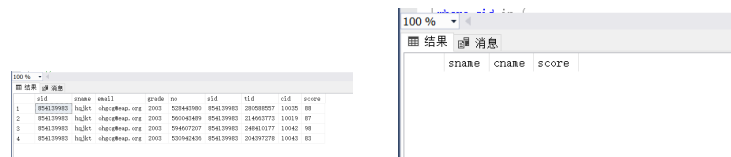
9. 查询至少选修了三门课程的学生编号。

```
SELECT sid from CHOICES GROUP BY sid HAVING COUNT(*)>=3 select * from CHOICES
where sid=812917218
```

10. 查询编号800009026的学生所选的全部课程的课程名和成绩。

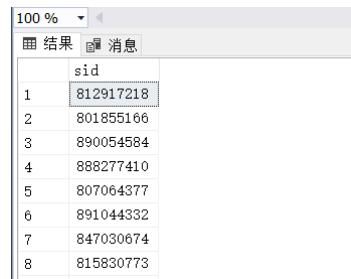
```
SELECT cname,score FROM CHOICES JOIN COURSES ON CHOICES.cid = COURSES.cid where
sid = 800009026
```

图 6: 运行结果

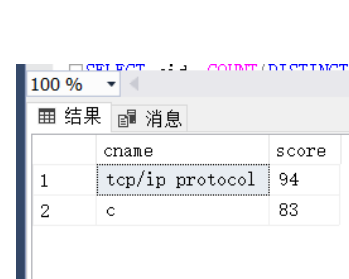


cid	cname	seat	grade	no	sid	tid	cid	score
854139983	hqlt	hqlt@exp_vra	2003	528443980	854139983	285088857	10025	88
854139983	hqlt	hqlt@exp_vra	2003	560043489	854139983	214463773	10019	87
854139983	hqlt	hqlt@exp_vra	2003	544607237	854139983	244615177	10042	86
854139983	hqlt	hqlt@exp_vra	2003	530942436	854139983	204297278	10043	83

图 7: 运行结果



sid
812917218
801855166
890054584
888277410
807064377
891044332
847030674
815830773

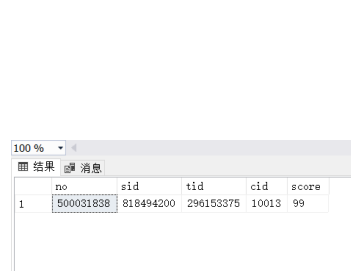


cname	score
tcp/ip protocol	94
c	83

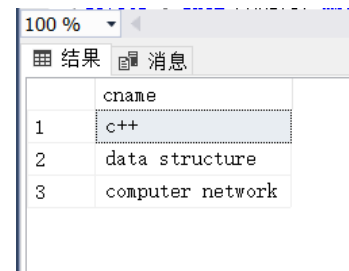
11. 查询所有选修了database的学生的编号。

```
SELECT sid FROM CHOICES where cid in (select cid from COURSES where cname like 'database') select * from CHOICES where sid=870899566 select * from COURSES
```

图 8: 运行结果



no	sid	tid	cid	score
500031838	818494200	296153375	10013	99



cname
c++
data structure
computer network

12. 求出选择了同一个课程的学生数。

```
SELECT cid, COUNT(DISTINCT sid) AS num_of_students FROM CHOICES GROUP BY cid;
```

```
SELECT COUNT(DISTINCT c1.sid) AS num_of_students, c1.cid FROM CHOICES c1 INNER JOIN CHOICES c2 ON c1.cid = c2.cid AND c1.sid <> c2.sid GROUP BY c1.cid;
```

13. 求出至少被两名学生选修的课程编号。

```
SELECT cid FROM CHOICES GROUP BY cid HAVING COUNT(*)>=2 select * from CHOICES where cid=10008
```

14. 查询选修了编号800009026的学生所选的某个课程的学生编号。

```
select top(1) sid from CHOICES where cid in ( select cid from CHOICES where sid = 800009026 ) order by NEWID()
```

15. 查询学生的基本信息及选修课程编号和成绩。

```
select * from STUDENTS JOIN CHOICES ON STUDENTS.sid = CHOICES.sid where STUDENTS.sid=854139983
```

16. 查询学号850955252的学生的姓名和选修的课程名及成绩。

图 9: 运行结果

	sname
1	tkbzqduq
2	efmgfprz
3	msqbykxi
4	jnbluzg
5	lndjfyinc
6	cadcsskc
7	minujyxfb
8	narsw
9	asczkv
10	...

	sname
1	...

```
SELECT s.sname, c.cname, ch.score FROM STUDENTS s, CHOICES ch, COURSES c WHERE
s.sid = ch.sid AND ch.cid = c.cid AND s.sid = '850955252';
```

17. 查询与学号850955252的学生同年级的所有学生资料。

```
SELECT * FROM STUDENTS where grade in (SELECT grade from STUDENTS where sid='850955252')
```

图 10: 运行结果

	cid	cname	hour
1	10004	java	48

	sname
1	vnbqzsuv
2	waqcj
3	aoahudi
4	cxjng
5	efmgfprz
6	ocofw
7	ptqno
8	goqiv

18. 查询所有有选课的学生们的详细信息。

```
select * from STUDENTS where sid in ( select distinct sid from CHOICES )
```

19. 查询没有学生选的课程编号。

```
select * from COURSES where cid not in ( select distinct cid from CHOICES )
```

图 11: 运行结果

	sid	sname	email	grade
1	800028044	ztozk	r36919a@laykh.gov	2001
2	800041569	pgarkdhh	xpq12vc@hjtj.edu	2001
3	800070739	rkdnfq	pto7n@sci.com	2001
4	800152632	qtblqzsv	tdvhn@neu.edu	2001
5	800166448	ctvxn	5b6zcx@pplh.org	2001
6	800169970	airmfv	ytkyox@eh.org	2001
7	800177146	vaesalave	tla7d@ypxrr.gov	2001
8	800202438	xiraegdlg	_fow_i@hbc.edu	2001
9	800268599	dwjny	9pax@ejk.com	2001
10	800269975	vqjxfic	15w9ba@spu.com	2001
11	800270084	ebwal	vp02qa@ekze.gov	2001
12	800284630	vtljkvzh	cc33gi@tqos.org	2001
13	800289261	kverfrpkv	3f597lp@tzn.net	2001
14	800295858	viatase	g3.usaki.com	2001

	sid	sname	email	grade
1	800008585	ehlycg	nachi0@uic.com	1999
2	800014678	fmvgrig	pikh@cccbb.com	1996
3	800015960	yqjhake	ko7y7n@fgp.com	1995
4	800017736	vupposq	vfi@jiqnn.edu	1998
5	800020890	orcghz	gfd9v9x@ipe.net	1998
6	800026053	inagsga	t0z2@bdja.edu	1993
7	800032997	vdfvfx	a6xnz2@lbdnd.edu	1993
8	800036362	yahvv	h5_x@fqc.edu	2002
9	800042511	rwlnliqf	s6flyf@tos.org	1991
10	800046262	ldounsoa	ojbyj0b@ulvix.org	1998
11	800051082	hfxjzme	ek_hid9@cvz.com	2002
12	800060416	lnfjda	dow0@uqrni.com	2002
13	800064534	xagzveisc	ijbm1@vry.gov	1996

20. 查询选修了与C++的课时一样课程的学生名称。

```
select sname from STUDENTS where sid in ( select distinct sid from CHOICES where
cid in ( select distinct cid from COURSES where hour in ( select hour from COURSES
where cname like 'c++' ) ) )
```

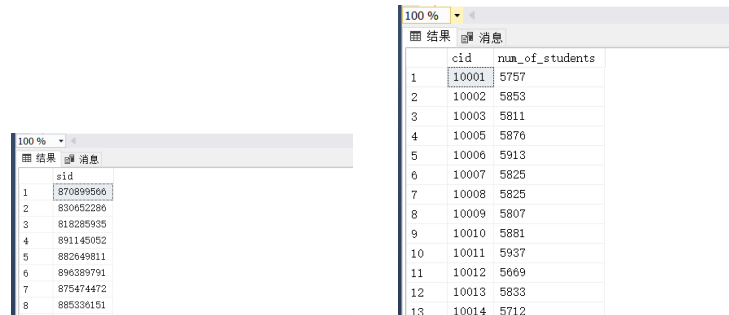
21. 找出选修课程成绩最好的选课记录。

```
select top (1) * from CHOICES order by score DESC
```

22. 找出和课程UML或课程C++的课时一样课程名称。

```
select cname from COURSES where hour in ( select hour from COURSES where cname
like 'C++' or cname like 'UML' )
```

图 12: 运行结果



	sid
1	870899566
2	830652286
3	818285935
4	891145052
5	882649811
6	896389791
7	875474472
8	885336151

	cid	num_of_students
1	10001	5757
2	10002	5853
3	10003	5811
4	10005	5876
5	10006	5913
6	10007	5825
7	10008	5825
8	10009	5807
9	10010	5881
10	10011	5937
11	10012	5669
12	10013	5833
13	10014	5712

23. 查询所有选修编号10001的课程的学生的姓名。

```
select sname from STUDENTS where sid in( select distinct sid from CHOICES where
cid = 10001 )
```

24. 查询选修了所有课程的学生姓名。

思路一、查询条件“为不存在一门课程，该学生没选”

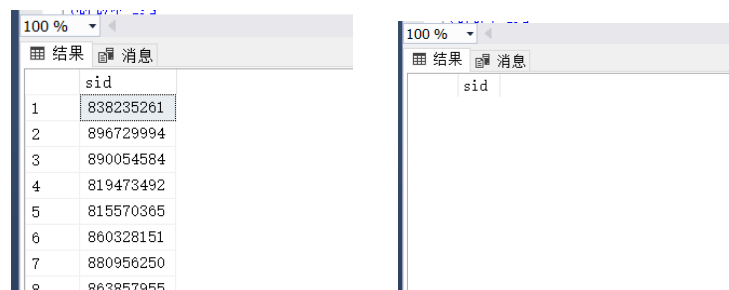
思路二、查询学生不同的选课ID数量和COURSE表中课程数量相等的学生

两种做法如下：

```
SELECT sid from CHOICES where not exists( select * from COURSES where cid not
in (select CHOICES.cid from CHOICES where CHOICES.sid = sid) )
```

```
SELECT sname FROM STUDENTS WHERE sid IN ( SELECT sid FROM CHOICES GROUP BY sid
HAVING COUNT(DISTINCT cid) = ( SELECT COUNT(*) FROM COURSES ) );
```

图 13: 运行结果



	sid
1	838235261
2	896729994
3	890054584
4	819473492
5	815570365
6	860328151
7	880956250
8	863857955

	sid
1	

25. 利用集合运算，查询选修课程C++或选修课程Java的学生的编号。

```
SELECT sid FROM CHOICES WHERE cid = (SELECT cid FROM COURSES WHERE cname = 'C++')
UNION SELECT sid FROM CHOICES WHERE cid = (SELECT cid FROM COURSES WHERE cname
= 'Java');
```

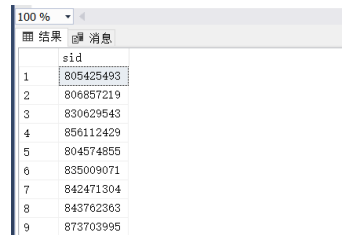
26. 实现集合交运算，查询既选修课程C++又选修课程Java的学生的编号。

```
SELECT sid FROM CHOICES WHERE cid = (SELECT cid FROM COURSES WHERE cname = 'C++')  
INTERSECT SELECT sid FROM CHOICES WHERE cid = (SELECT cid FROM COURSES WHERE cname  
= 'Java');
```

27. 实现集合减运算，查询选修课程C++而没有选修课程Java的学生的编号。

```
SELECT sid FROM CHOICES WHERE cid = (SELECT cid FROM COURSES WHERE cname = 'C++')  
EXCEPT SELECT sid FROM CHOICES WHERE cid = (SELECT cid FROM COURSES WHERE cname  
= 'Java');
```

图 14: 运行结果



The screenshot shows a database query result window with a title bar at 100% zoom. It contains a table with two columns: 'sid' and a list of 9 student IDs. The table is titled '结果' (Results) and '消息' (Messages). The data is as follows:

	sid
1	805425493
2	806857219
3	830629543
4	856112429
5	804574855
6	835009071
7	842471304
8	843762363
9	873703995



## 2 实验2.2

1. 查询所有选课记录的成绩并将它换算为五分制（满分5分，合格3分），注意SCORE取NULL值的情况。

思路：使用CASE WHEN THEN END语句处理NULL，将其替换成0

```
SELECT no, sid, tid, cid, CASE WHEN score IS NULL THEN NULL WHEN score >= 60 THEN
(score-60)/10+2 ELSE 0.0 END AS gpa FROM CHOICES;
```

2. 通过查询选修编号10028的课程的学生的人数，其中成绩合格的学生人数，不合格的学生人数，讨论NULL值的特殊含义。

```
SELECT COUNT(*) AS total_students, SUM(CASE WHEN score IS NULL THEN 0 ELSE 1 END)
AS scored_students, SUM(CASE WHEN score >= 60 THEN 1 ELSE 0 END) AS passed_students,
SUM(CASE WHEN score >= 60 THEN 0 WHEN score IS NULL THEN 0 ELSE 1 END) AS failed_students
FROM CHOICES WHERE cid = '10028';
```

在这道题的条件下，NULL值的含义可能是缺考、缓考等特殊情况导致的没有成绩。

图 15: 运行结果

no	sid	tid	cid	five_scale_score
1	500000058	823009829	24096497	3.0
2	500000258	829848273	24096410	4.0
3	500000984	860962352	234145410	0.0
4	500001270	847061074	292045491	5.0
5	500002143	886025914	238811498	4.0
6	500002371	829785982	273199168	3.0
7	500002715	822137137	218923966	2.0
8	500003252	828310502	247846042	5.0
9	500004144	817636568	253205179	2.0
10	500005519	813520169	226385492	0.023
11	500005925	801967882	234419511	3.0
12	500005967	975434313	229646385	4.0
13	500006353	831380555	218440500	3.0

	total_students	scored_students	passed_students	failed_students
1	6042	5967	4912	730

3. 通过实验检验在使用ORDER BY进行排序时，取NULL的项是否出现在结果中？如果有，在什么位置？

```
SELECT score FROM CHOICES ORDER BY score ASC;
```

运行以上代码，可以发现ASC排序时最先出现，DESC排序时会最后出现

4. 在上面的查询过程中如果加上保留字DISTINCT会有什么效果？

```
SELECT DISTINCT score FROM CHOICES ORDER BY score ASC;
```

运行以上代码，可以发现会保留一个NULL值的项。

图 16: 运行结果

	score
1	NULL
2	50
3	51
4	52
5	53
6	54
7	55
8	60
9	61
10	62
11	63
12	64

100 %

SELECT score FROM CHOICES ORDER BY score ASC;

结果 消息

score

1 NULL

2 50

3 51

4 52

5 53

6 54

7 55

8 60

9 61

10 62

11 63

12 64

13 65

14 66

5. 通过实验说明使用分组GROUP BY对取值为NULL的项的处理。

```
SELECT score, COUNT(*) FROM CHOICES GROUP BY score;
```

会将NULL值的元组单独放在一组中。

6. 结合分组，使用集合函数求每个同学的平均分、总的选课记录数、最高成绩、最低成绩和总成绩。

```
SELECT sid, AVG(score) AS avg_score, COUNT(*) AS total_records, MAX(score) AS
max_score, MIN(score) AS min_score, SUM(score) AS sum_score FROM CHOICES GROUP BY
sid;
```

如果考虑NULL，可以将其替换成0。函数ISNULL可以将NULL值替换成0。

```
SELECT sid, AVG(ISNULL(score, 0)) AS avg_score, COUNT(*) AS total_records, MAX(ISNULL(score,
0)) AS max_score, MIN(ISNULL(score, 0)) AS min_score, SUM(ISNULL(score, 0)) AS
sum_score FROM CHOICES GROUP BY sid;
```

图 17: 运行结果

100 %	结果	消息
score	(无列名)	
23	90	5898
24	96	5907
25	65	5912
26	79	5734
27	73	5771
28	85	5847
29	62	5765
30	99	5839
31	76	5877
32	NULL	23441
33	82	5915
34	88	5848
35	53	5788
36	71	5881

100 %						
消息						
	sid	avg_score	total_records	max_score	min_score	sum_score
1	834984317	73	4	90	53	292
2	871390593	81	3	92	61	243
3	845379956	73	2	79	67	146
4	805050629	46	2	71	62	133
5	86124477	92	2	94	92	185
6	861315623	NULL	2	NULL	NULL	NULL
7	848752380	80	5	94	69	404
8	820928013	74	4	92	51	297
9	830717441	86	2	92	68	162
10	895200397	73	5	89	51	292
11	816522768	74	3	84	61	222
12	885920900	80	1	80	80	80
13	851274287	87	4	89	88	175

7. 查询成绩小于60的选课记录，统计总数、平均分、最大值和最小值。

```
SELECT COUNT(*) AS total_records, AVG(ISNULL(score, 0)) AS avg_score, MAX(ISNULL(score,
0)) AS max_score, MIN(ISNULL(score, 0)) AS min_score FROM CHOICES WHERE ISNULL(score,
0) < 60;
```

8. 采用嵌套查询的方式，利用比较运算符和谓词ALL的结合来查询表COURSES中最少的课时。假设数据库中只有一个记录的时候，使用前面的方法会得到什么结果，为什么？

```
SELECT MIN(hour) FROM COURSES WHERE hour <= ALL (SELECT hour FROM COURSES WHERE
hour > 0);
```

如果数据库中只有一个记录，那么子查询SELECT hour FROM COURSES将返回该记录的课时，并且它是该表课时的最小值。因此，主查询中WHERE子句的条件 hour >= ALL(SELECT hour FROM COURSES)也将成立，因此查询结果将为该记录的课时值。

图 18: 运行结果

100 %

结果

消息

	sid	avg_score	total_records	max_score	min_score	sum_score
1	812937218	77	5	99	53	386
2	801805166	87	4	94	76	349
3	888277410	73	5	90	65	305
4	890054684	77	3	99	51	231
5	807064277	88	4	99	77	354
6	883247944	32	2	65	0	65
7	801644532	93	3	99	85	280
8	807443390	75	2	90	60	150
9	884999158	93	1	93	93	93
10	847030574	86	5	79	0	282
11	848032458	49	2	99	0	99
12	844908195	91	2	99	83	182
13	819830773	89	4	87	77	323
14	801440608	71	1	71	71	71

100 %

	total_records	avg_score	max_score	min_score
1	58734	31	55	0

9. 创建一个学生表S(NO, SID, SNAME), 教师表T(NO, TID, TNAME)作为实验用的表。其中NO分别是这两个表的主键, 其他键允许为空。向S插入元组(1, 0129871001, 王小明)、(2, 0129871002, 李兰)、(3, 0129871005, NULL)、(4, 0129871004, 关红); 向T插入元组(1, 100189, 王小明)、(2, 100180, 李小)、(3, 100121, NULL)、(4, 100128, NULL)。对这两个表作对姓名的等值连接运算, 找出既是老师又是学生的人员的学生编号和老师编号。

```
CREATE TABLE S ( NO INT PRIMARY KEY, SID CHAR(10), SNAME VARCHAR(20) );
CREATE TABLE T ( NO INT PRIMARY KEY, TID CHAR(10), TNAME VARCHAR(20) );

INSERT INTO S(NO, SID, SNAME) VALUES (1, '0129871001', '王小明'), (2, '0129871002', '李兰'), (3, '0129871005', NULL), (4, '0129871004', '关红');
INSERT INTO T(NO, TID, TNAME) VALUES (1, '100189', '王小明'), (2, '100180', '李小'), (3, '100121', NULL), (4, '100128', NULL);
SELECT * FROM S
SELECT * FROM T

SELECT S.SID, T.TID FROM S JOIN T ON S.SNAME = T.TNAME WHERE S.SNAME IS NOT NULL
AND T.TNAME IS NOT NULL;
```

图 19: 运行结果

The screenshot displays two query result windows. The left window shows the result of a query on table S, with a single row containing the value 48. The right window shows the result of a query on table T, with a single row containing the value 100189. Below these, a third window shows the result of a join query, displaying a single row with the values 0129871001 and 100189.

NO	SID	SNAME
1	0129871001	王小明
2	0129871002	李兰
3	0129871005	NULL
4	0129871004	关红

NO	TID	TNAME
1	100189	王小明
2	100180	李小
3	100121	NULL
4	100128	NULL

SID	TID
0129871001	100189