

一、实验题目：内置函数与多表查询

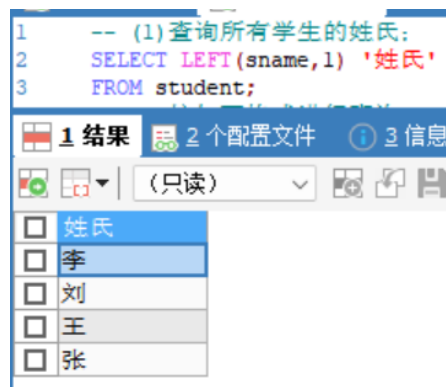
二、实验目的

1. 熟练掌握 MySQL 内置函数的使用方法，包括字符串函数、日期函数和集函数。
2. 熟练掌握连接查询；

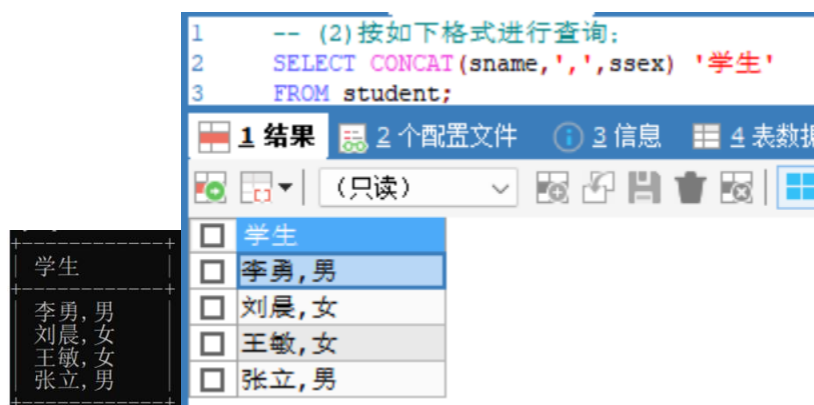
三、实验内容（利用截屏和文字对实验过程进行说明）

1. 字符串函数：

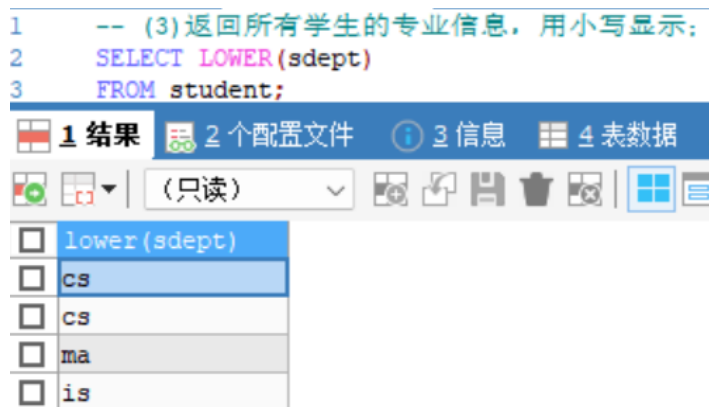
(1) 查询所有学生的姓氏；



(2) 按如下格式进行查询；



(3) 返回所有学生的专业信息，用小写显示；



(4) 查询所有课程的名字, 以及课程名所占用的字节数;

```
1  -- (4) 查询所有课程的名字, 以及课程名所占用的字节数:
2  SELECT cname, LENGTH(cname) '占用字节数'
3  FROM course
```

1 结果	2 个配置文件	3 信息	4 表数据	5 信息
(只读)				
cname	占用字节数			
数据库	9			
数学	6			
信息系统	12			
操作系统	12			
数据结构	12			
数据处理	12			
PASCAL语言	12			

(5) 查询所有课程的详细信息, 并将所有课程名中的“数据”替换为“Data”;

```
1  -- (5) 查询所有课程的详细信息, 并将所有课程名中的“数据”替换为“Data”:
2  SELECT REPLACE(cname, '数据', 'Data') '替换后'
3  FROM course;
```

1 结果	2 个配置文件	3 信息	4 表数据	5 信息
(只读)				
替换后				
Data库				
数学				
信息系统				
操作系统				
Data结构				
Data处理				
PASCAL语言				

(6) 自行测试其它字符串函数的功能。

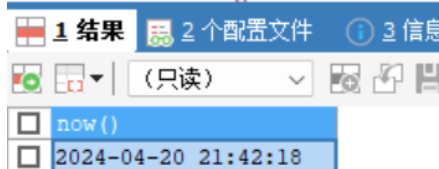
```
1  -- (6) 自行测试其它字符串函数的功能。
2  SELECT REPEAT(cno, 3) '重复三次'
3  FROM course;
```

1 结果	2 个配置文件	3 信息	4 表
(只读)			
重复三次			
111			
222			
333			
444			
555			
666			
777			

2. 日期和时间函数:

(1) 查询当前的日期时间;

```
1  -- (1) 查询当前的日期时间:
2  SELECT NOW();
```

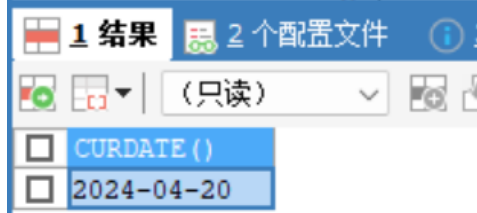


The screenshot shows the results of the SQL query `SELECT NOW();`. The interface includes tabs for '1 结果' (Results), '2 个配置文件' (2 configuration files), and '3 信息' (3 information). Below the tabs, there are icons for execution, a dropdown menu set to '(只读)' (Read-only), and icons for saving, printing, and refreshing. The results pane shows two rows: the first row is the function name `now()`, and the second row shows the current date and time: `2024-04-20 21:42:18`.

now()
2024-04-20 21:42:18

(2) 查询当前的日期;

```
1  -- (2) 查询当前的日期:
2  SELECT CURDATE();
```

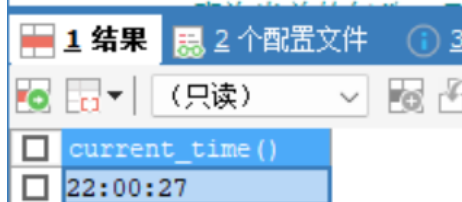


The screenshot shows the results of the SQL query `SELECT CURDATE();`. The interface includes tabs for '1 结果' (Results), '2 个配置文件' (2 configuration files), and '3 信息' (3 information). Below the tabs, there are icons for execution, a dropdown menu set to '(只读)' (Read-only), and icons for saving, printing, and refreshing. The results pane shows two rows: the first row is the function name `CURDATE()`, and the second row shows the current date: `2024-04-20`.

CURDATE()
2024-04-20

(3) 查询当前的时间;

```
1  -- (3) 查询当前的时间:
2  SELECT CURRENT_TIME();
```




The screenshot shows the results of the SQL query `SELECT CURRENT_TIME();`. The interface includes tabs for '1 结果' (Results), '2 个配置文件' (2 configuration files), and '3 信息' (3 information). Below the tabs, there are icons for execution, a dropdown menu set to '(只读)' (Read-only), and icons for saving, printing, and refreshing. The results pane shows two rows: the first row is the function name `current_time()`, and the second row shows the current time: `22:00:27`.

current_time()
22:00:27

(4) 查询当前的年份、月份、日;

```
1  -- (4) 查询当前的年份、月份、日:
2  SELECT
3      YEAR(CURDATE()) AS nowYear,
4      MONTH(CURDATE()) AS nowMonth,
5      DAY(CURDATE()) AS nowDay;
```



The screenshot shows the results of the SQL query that extracts the year, month, and day from the current date. The interface includes tabs for '1 结果' (Results), '2 个配置文件' (2 configuration files), '3 信息' (3 information), and '4 表数据' (4 table data). Below the tabs, there are icons for execution, a dropdown menu set to '(只读)' (Read-only), and icons for saving, printing, and refreshing. The results pane shows a table with three columns: `nowYear`, `nowMonth`, and `nowDay`. The data row shows the values `2024`, `4`, and `20` respectively.

nowYear	nowMonth	nowDay
2024	4	20

(5) 查询 7 天后的日期、7 天前的日期;

```
1  -- (5) 查询7天后的日期、7天前的日期:
2  SELECT DATE_ADD(CURDATE(), INTERVAL 7 DAY) '七天后的日期';
```



The screenshot shows the results of the SQL query `SELECT DATE_ADD(CURDATE(), INTERVAL 7 DAY) '七天后的日期';`. The interface includes tabs for '1 结果' (Results), '2 个配置文件' (2 configuration files), '3 结果' (3 results), '4 信息' (4 information), '5 表数据' (5 table data), and '6 信息' (6 information). Below the tabs, there are icons for execution, a dropdown menu set to '(只读)' (Read-only), and icons for saving, printing, and refreshing. The results pane shows two rows: the first row is the text `七天后的日期`, and the second row shows the date `2024-04-27`.

七天后的日期
2024-04-27

```

1  -- (5) 查询7天后的日期、7天前的日期:
2  SELECT DATE_SUB(CURDATE(),INTERVAL 7 DAY) '七天前的日期';

```

1 结果 2 个配置文件 3 信息 4 表数据 5 信息

(只读)

七天前的日期
2024-04-13

(6) 查询你从出生到现在生活了多少天;

```

1  -- (6) 查询你从出生到现在生活了多少天:
2  SELECT DATEDIFF(CURDATE(),'2004-11-6') '生活了多少天';

```

1 结果 2 个配置文件 3 信息 4 表数据 5 信息

(只读)

生活了多少天
7105

(7) 创建一张留言表, 保存每条留言的编号(自动编号, 主码)、内容(非空)、留言时间(非空), 并插入若干条留言;

```

1  -- (7) 创建一张留言表, 保存每条留言的编号(自动编号, 主码)、内容(非空)、留言时间(非空), 并插入若干条留言:
2  CREATE TABLE comments(
3      id INT AUTO_INCREMENT PRIMARY KEY,
4      content TEXT NOT NULL,
5      comment_time TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
6  );
7  INSERT INTO comments(content) VALUES ('这是第一条留言内容');
8  INSERT INTO comments(content) VALUES ('这是第二条留言内容');
9  INSERT INTO comments(content) VALUES ('这是第三条留言内容');

```

1 信息 2 表数据 3 信息

限制行 第一行: 0 行数

id	content	comment_time
1	这是第一条留言内容	2024-04-20 22:12:11
2	这是第二条留言内容	2024-04-20 22:12:11
3	这是第三条留言内容	2024-04-20 22:12:11
*	(Auto) (NULL)	OK CURRENT_TIMESTAMP

(8) 查询所有留言信息, 包括编号、内容、留言的日期(不显示时间);

```

1  -- (8) 查询所有留言信息, 包括编号、内容、留言的日期(不显示时间):
2  SELECT id,content,DATE(comment_time)
3  FROM comments;

```

1 结果 2 个配置文件 3 信息 4 表数据 5 信息

(只读)

id	content	date(comment_time)
1	这是第一条留言内容	2024-04-20
2	这是第二条留言内容	2024-04-20
3	这是第三条留言内容	2024-04-20

(9) 查询 1 分钟内的所有留言信息。

```

1  -- (9)查询1分钟内的所有留言信息。
2  SELECT id,content,comment_time
3  FROM comments
4  WHERE TIMESTAMPDIFF(MINUTE,comment_time,NOW())<= 1;

```

1 结果			2 个配置文件	3 信息	4 表数据	5 信息
(只读)						
id	content	comment_time				
10	这是第四条留言内容	2024-04-20 22:17:54				

3. 连接查询：

(1) 查询所有姓李的学生的选课情况，要求显示学号、姓名、所在系、课程名、成绩；

```

1  -- (1)查询所有姓李的学生的选课情况，要求显示学号、姓名、所在系、课程名、成绩：
2  SELECT sc.sno,sname,sdept,cname,grade
3  FROM student,sc,course
4  WHERE sc.sno=student.sno
5         AND sc.cno=course.cno
6         AND sname LIKE '李%';

```

1 结果

2 个配置文件

3 信息

4 表数据

5 信息

(只读)

(2) 查找选修了没有先行课的课程的学生姓名；

```

1  -- (2)查找选修了没有先行课的课程的学生姓名：
2  SELECT DISTINCT sname
3  FROM student,sc,course
4  WHERE sc.cno=course.cno
5         AND sc.sno=student.sno
6         AND cpno IS NULL

```

1 结果

2 个配置文件

3 信息

4 表数据

(只读)

<input type="checkbox"/>	sname
<input type="checkbox"/>	李勇
<input type="checkbox"/>	刘晨

(3) 查询比刘晨大的学生姓名、年龄以及刘晨的年龄；


```

1  -- (3)查询比刘晨大的学生姓名、年龄以及刘晨的年龄:
2  SELECT sname,sage
3  FROM student
4  WHERE sage>(
5  SELECT sage
6  FROM student
7  WHERE sname='刘晨')
8  UNION ALL
9  SELECT sname,sage
10 FROM student
11 WHERE sname='刘晨';

```

1 结果		2 个配置文件	3 信息	4 表数据	5 信息
(只读)					
<input type="checkbox"/>	sname	sage			
<input type="checkbox"/>	李勇	20			
<input type="checkbox"/>	刘晨	19			

4. 集函数的使用:

(1) 查询各学生的平均成绩,并按成绩由高到低排序;

```

1  -- (1)查询各学生的平均成绩,并按成绩由高到低排序:
2  SELECT AVG(grade) '平均成绩'
3  FROM sc
4  GROUP BY sno
5  ORDER BY AVG(grade) DESC;

```

1 结果		2 个配置文件	3 信息	4 表数据	5 信息
(只读)					
<input type="checkbox"/>	平均成绩				
<input type="checkbox"/>	88.3333				
<input type="checkbox"/>	85.0000				

(2) 查询平均成绩在 86 分以上的学生的姓名;

```

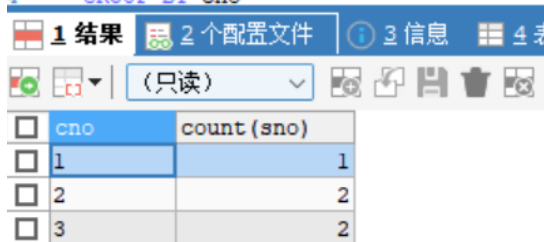
1  -- (2)查询平均成绩在86分以上的学生的姓名:
2  SELECT sname
3  FROM sc,student
4  WHERE sc.sno=student.sno
5  GROUP BY sc.sno
6  HAVING AVG(grade)>86;

```

1 结果		2 个配置文件	3 信息	4 表数据
(只读)				
<input type="checkbox"/>	sname			
<input type="checkbox"/>	李勇			

(3) 查询选修了课程的学生的人数;

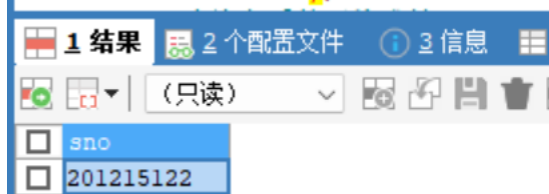
```
1  -- (3) 查询选修了课程的学生的人数:
2  SELECT cno, COUNT(sno)
3  FROM sc
4  GROUP BY cno
```



cno	count(sno)
1	1
2	2
3	2

(4) 查询 2 号课最高分学生的学号;

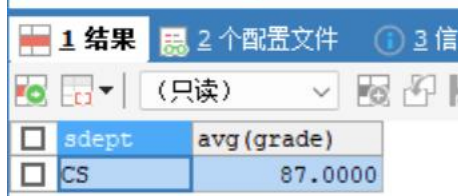
```
1  -- (4) 查询2号课最高分学生的学号:
2  SELECT sno
3  FROM sc
4  WHERE cno=2 AND grade=(
5      SELECT MAX(grade)
6      FROM sc
7      WHERE cno=2);
```



sno
201215122

(5) 查询各系的平均成绩。

```
1  -- (5) 查询各系的平均成绩。
2  SELECT sdept, AVG(grade)
3  FROM sc, student
4  WHERE sc.sno=student.sno
5  GROUP BY sdept;
```



sdept	avg(grade)
CS	87.0000

四、实验分析与总结（遇到的困难及解决方法、对知识点的理解）

分析：本次实验所用到的函数在实验指导中都有介绍，结合 csdn 对这些函数的详细讲解，可以正常使用；集函数和连接查询在课堂上也有学习和练习，总体来讲没有产生太多困难。