SQL 作业解答

所有代码可见根目录 code 文件夹`w5_sql.sql`

查看源码

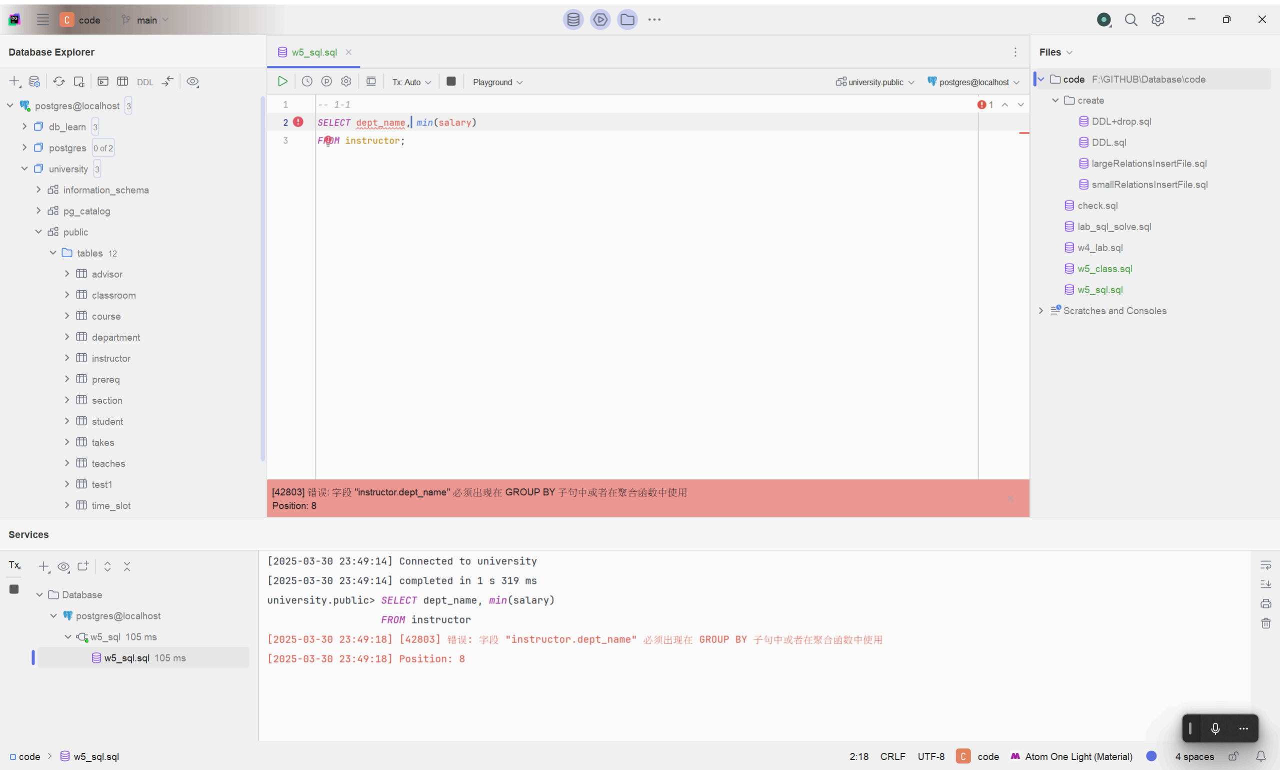
1 题目一

请问下面的 SQL 语句是否合法？用实验验证你的想法。你从实验结果能得到什么结论？

```sql
SELECT dept_name, min(salary)
FROM instructor;

SELECT dept_name, min(salary)
FROM instructor
GROUP BY dept_name
HAVING name LIKE '%at%';

SELECT dept_name
FROM instructor
WHERE AVG(salary) > 20000;
```

1. 不合法，因为 min 函数需要作用在组上，而这里没有 group by。

```sql
SELECT dept_name, min(salary)
FROM instructor;
```



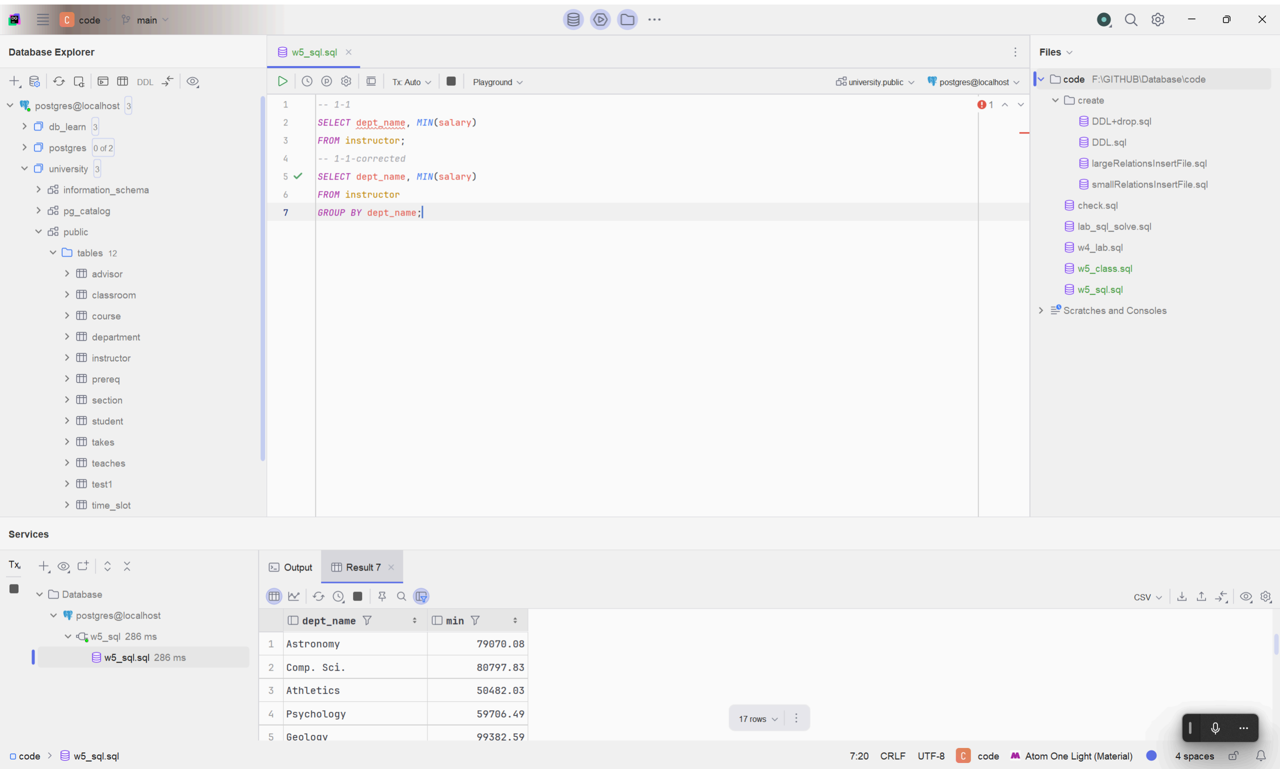
```sql
-- 1-1
SELECT dept_name, min(salary)
FROM instructor;
```

[42803] 错误: 字段 "instructor.dept_name" 必须出现在 GROUP BY 子句中或者在聚合函数中使用
Position: 8

```
[2025-03-30 23:49:14] Connected to university
[2025-03-30 23:49:14] completed in 1 s 319 ms
university.public> SELECT dept_name, min(salary)
                   FROM instructor
[2025-03-30 23:49:18] [42803] 错误: 字段 "instructor.dept_name" 必须出现在 GROUP BY 子句中或者在聚合函数中使用
[2025-03-30 23:49:18] Position: 8
```

修正:

```sql
SELECT dept_name, min(salary)
FROM instructor
GROUP BY dept_name;
```

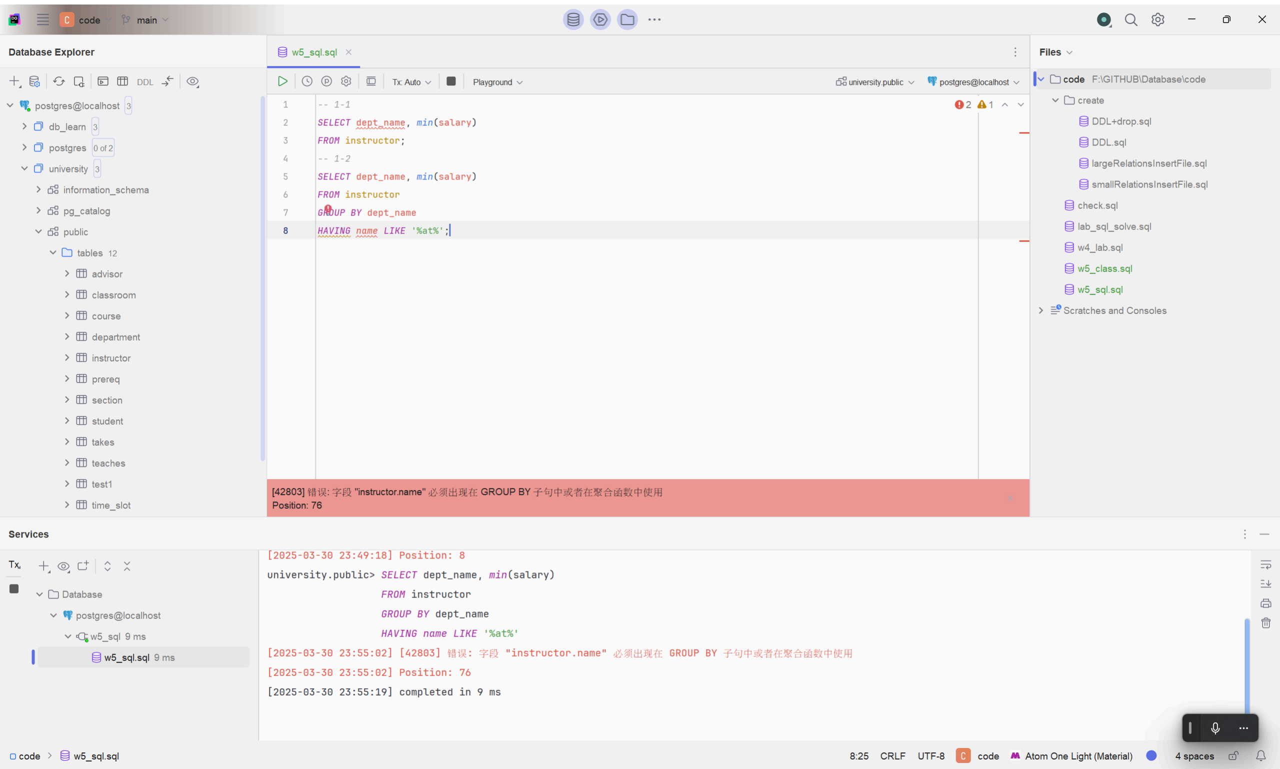


```sql
-- 1-1
SELECT dept_name, MIN(salary)
FROM instructor;
-- 1-1-corrected
SELECT dept_name, MIN(salary)
FROM instructor
GROUP BY dept_name;
```

	dept_name	min
1	Astronomy	79070.08
2	Comp. Sci.	80797.83
3	Athletics	50482.03
4	Psychology	59706.49
5	Geology	99382.59

2. 不合法, Having 的字段 name 不在 group by 中, 会产生歧义, 因为 group by 后一个 dept_name 组内的 name 有多个值, 无法判断使用哪个 name 值来过滤.

```sql
SELECT dept_name, min(salary)
FROM instructor
GROUP BY dept_name
HAVING name LIKE '%at%';
```



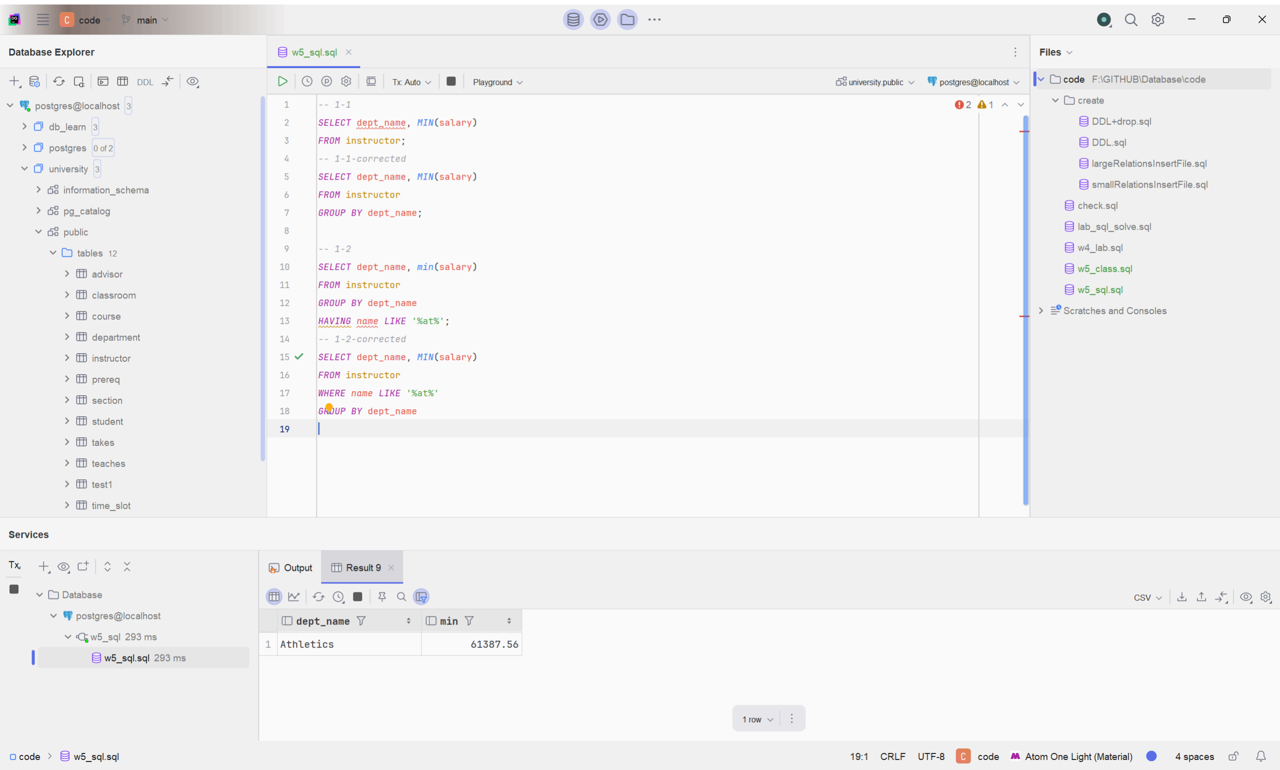
```sql
-- 1-1
SELECT dept_name, min(salary)
FROM instructor;
-- 1-2
SELECT dept_name, min(salary)
FROM instructor
GROUP BY dept_name
HAVING name LIKE '%at%';
```

[42803] 错误: 字段 "instructor.name" 必须出现在 GROUP BY 子句中或者在聚合函数中使用
Position: 76

```
[2025-03-30 23:49:18] Position: 8
university.public> SELECT dept_name, min(salary)
                   FROM instructor
                   GROUP BY dept_name
                   HAVING name LIKE '%at%'
[2025-03-30 23:55:02] [42803] 错误: 字段 "instructor.name" 必须出现在 GROUP BY 子句中或者在聚合函数中使用
[2025-03-30 23:55:02] Position: 76
[2025-03-30 23:55:19] completed in 9 ms
```

修正:

```sql
SELECT dept_name, MIN(salary)
FROM instructor
WHERE name LIKE '%at%'
GROUP BY dept_name;
```

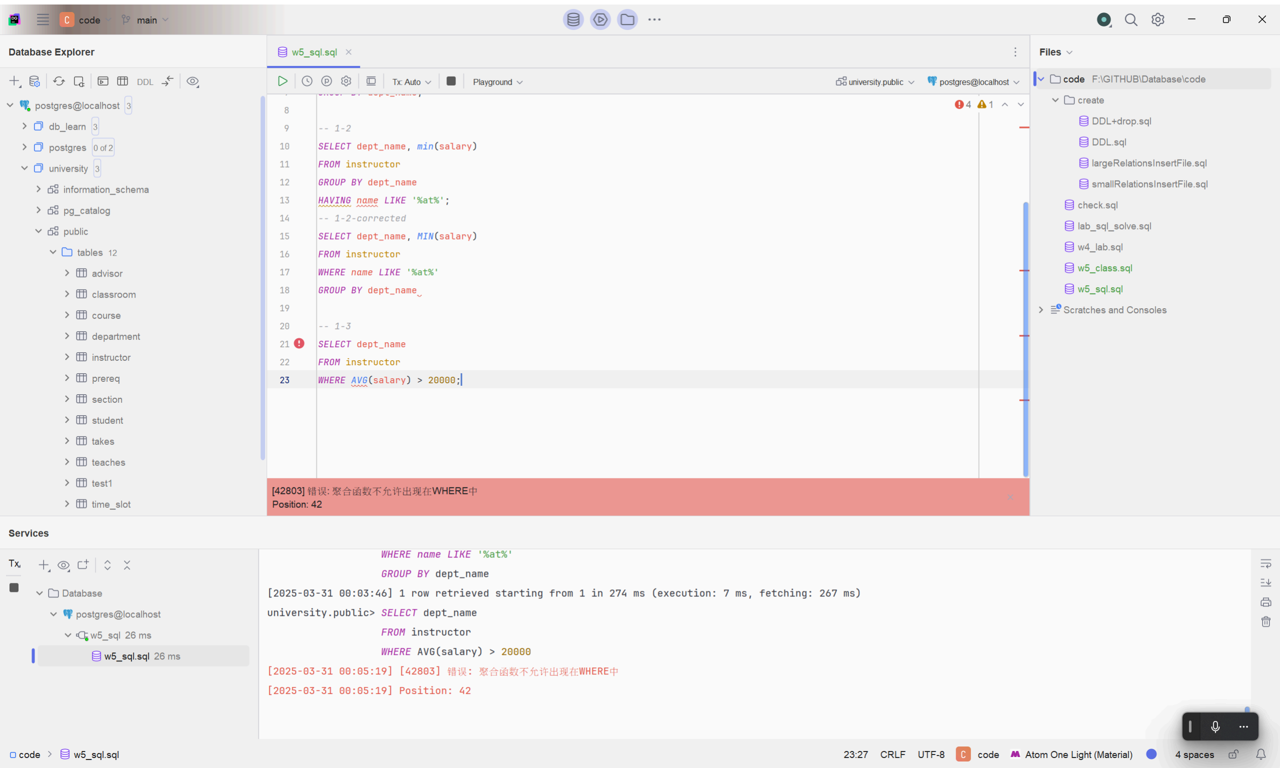


```sql
-- 1-1
SELECT dept_name, MIN(salary)
FROM instructor;
-- 1-1-corrected
SELECT dept_name, MIN(salary)
FROM instructor
GROUP BY dept_name;

-- 1-2
SELECT dept_name, min(salary)
FROM instructor
GROUP BY dept_name
HAVING name LIKE '%at%';
-- 1-2-corrected
SELECT dept_name, MIN(salary)
FROM instructor
WHERE name LIKE '%at%'
GROUP BY dept_name
```

dept_name	min
Athletics	61387.56

3. 不合法, 因为聚合函数不能在 where 中使用, 且必须要在 group by 中确定聚合的前置条件.

```sql
SELECT dept_name
FROM instructor
WHERE AVG(salary) > 20000;
```



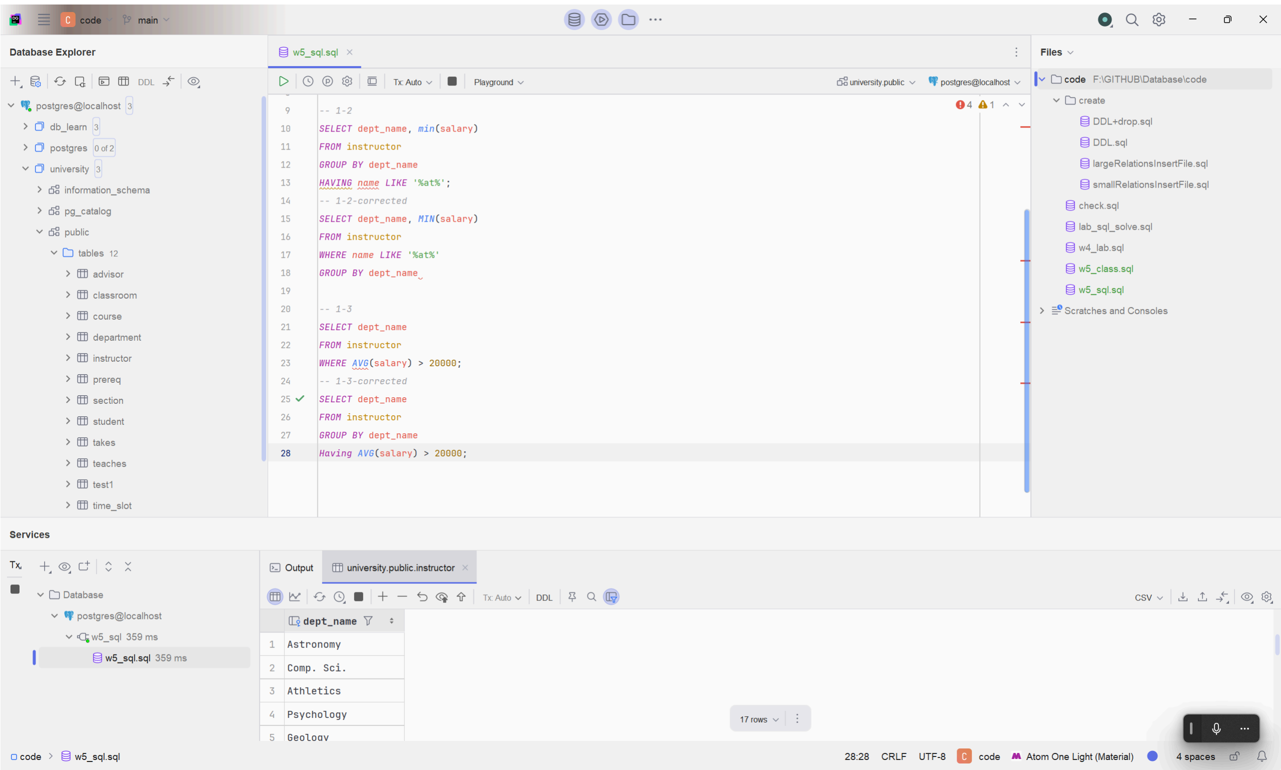
```sql
-- 1-2
SELECT dept_name, min(salary)
FROM instructor
GROUP BY dept_name
HAVING name LIKE '%at%';
-- 1-2-corrected
SELECT dept_name, MIN(salary)
FROM instructor
WHERE name LIKE '%at%'
GROUP BY dept_name

-- 1-3
SELECT dept_name
FROM instructor
WHERE AVG(salary) > 20000;
```

[42803] 错误: 聚合函数不允许出现在WHERE中
Position: 42

```
                WHERE name LIKE '%at%'
                GROUP BY dept_name
[2025-03-31 00:03:46] 1 row retrieved starting from 1 in 274 ms (execution: 7 ms, fetching: 267 ms)
university.public> SELECT dept_name
                FROM instructor
                WHERE AVG(salary) > 20000
[2025-03-31 00:05:19] [42803] 错误: 聚合函数不允许出现在WHERE中
[2025-03-31 00:05:19] Position: 42
```

修正:

```sql
SELECT dept_name
FROM instructor
GROUP BY dept_name
HAVING AVG(salary) > 20000;
```



```sql
-- 1-2
SELECT dept_name, min(salary)
FROM instructor
GROUP BY dept_name
HAVING name LIKE '%at%';
-- 1-2-corrected
SELECT dept_name, MIN(salary)
FROM instructor
WHERE name LIKE '%at%'
GROUP BY dept_name

-- 1-3
SELECT dept_name
FROM instructor
WHERE AVG(salary) > 20000;
-- 1-3-corrected
SELECT dept_name
FROM instructor
GROUP BY dept_name
Having AVG(salary) > 20000;
```

```
  dept_name
1 Astronomy
2 Comp. Sci.
3 Athletics
4 Psychology
5 Geology
```

2 题目二

1. 找到工资最高员工的名字，假设工资最高的员工只有一位（至少两种写法）。

2. 找到工资最高员工的名字，假设工资最高的员工有多位（试试多种写法）。

3. 解释下面四句。

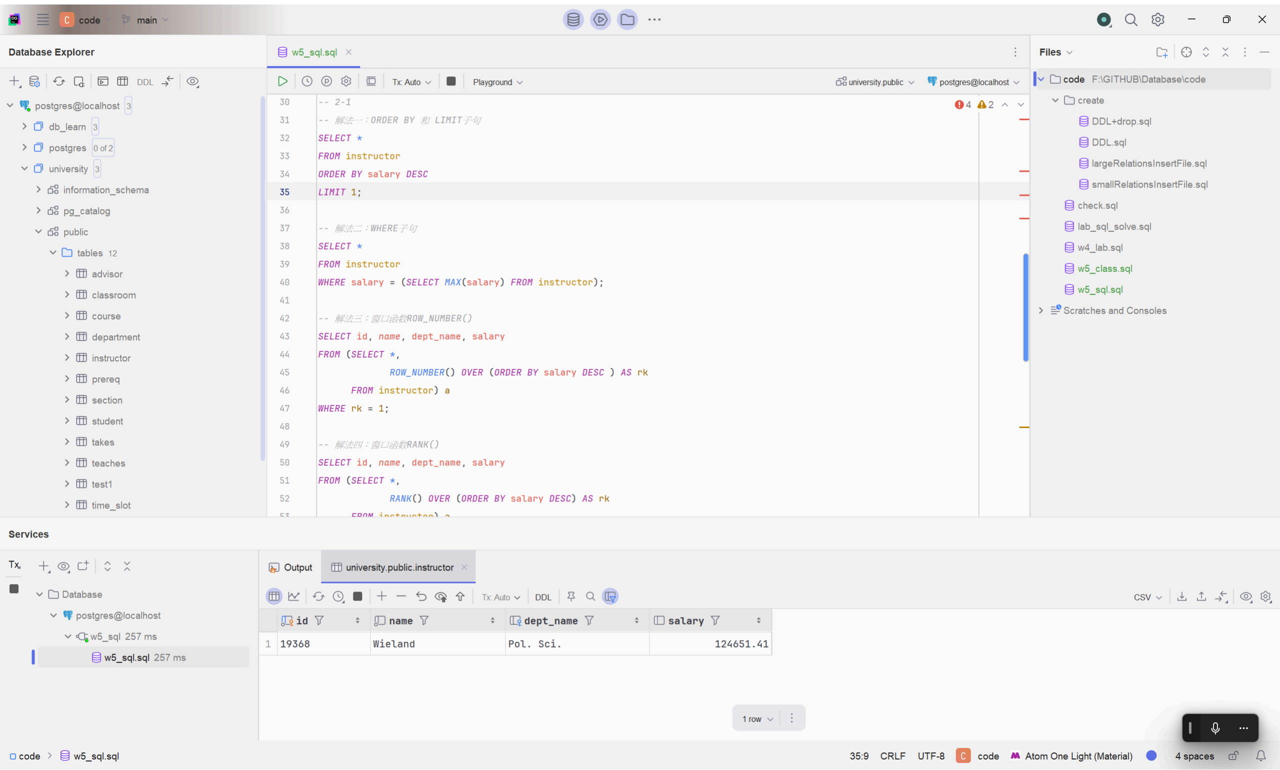
```sql
SELECT 1 IN (1);

SELECT 1 = (1);

SELECT (1, 2) = (1, 2);

SELECT (1) IN (1, 2);
```

1.

```sql
-- 解法一：ORDER BY 和 LIMIT子句
SELECT *
FROM instructor
ORDER BY salary DESC
LIMIT 1;

-- 解法二：WHERE子句
SELECT *
FROM instructor
WHERE salary = (SELECT MAX(salary) FROM instructor);

-- 解法三：窗口函数ROW_NUMBER()
```

```sql
SELECT id, name, dept_name, salary
FROM (SELECT *,
             ROW_NUMBER() OVER (ORDER BY salary DESC ) AS rk
      FROM instructor) a
WHERE rk = 1;

-- 解法四：窗口函数RANK()
SELECT id, name, dept_name, salary
FROM (SELECT *,
             RANK() OVER (ORDER BY salary DESC) AS rk
      FROM instructor) a
WHERE rk = 1;

-- 解法五：窗口函数DENSE_RANK()
SELECT id, name, dept_name, salary
FROM (SELECT *,
             DENSE_RANK() OVER (ORDER BY salary DESC) AS rk
      FROM instructor) a
WHERE rk = 1;

-- 解法六：HAVING子句
SELECT id, name, dept_name, salary
FROM instructor
GROUP BY id, name, dept_name, salary
HAVING salary = (SELECT MAX(salary) FROM instructor);

-- 解法七：ALL关键字
SELECT *
FROM instructor
WHERE salary >= ALL (SELECT salary FROM instructor);

-- 解法八：  JOIN子句
SELECT i.*
FROM instructor i
         JOIN (SELECT MAX(salary) AS max_salary FROM instructor) m
              ON i.salary = m.max_salary;

```

经过验证,上述所有解法输出相同



```
-- 2-1
-- 解法一：ORDER BY 和 LIMIT子句
SELECT *
FROM instructor
ORDER BY salary DESC
LIMIT 1;

-- 解法二：WHERE子句
SELECT *
FROM instructor
WHERE salary = (SELECT MAX(salary) FROM instructor);

-- 解法三：窗口函数ROW_NUMBER()
SELECT id, name, dept_name, salary
FROM (SELECT *,
             ROW_NUMBER() OVER (ORDER BY salary DESC ) AS rk
      FROM instructor) a
WHERE rk = 1;

-- 解法四：窗口函数RANK()
SELECT id, name, dept_name, salary
FROM (SELECT *,
             RANK() OVER (ORDER BY salary DESC) AS rk
```

```
  id     name     dept_name   salary
1 19368  Wieland  Pol. Sci.   124651.41
```

2. 首先再插入一条数据

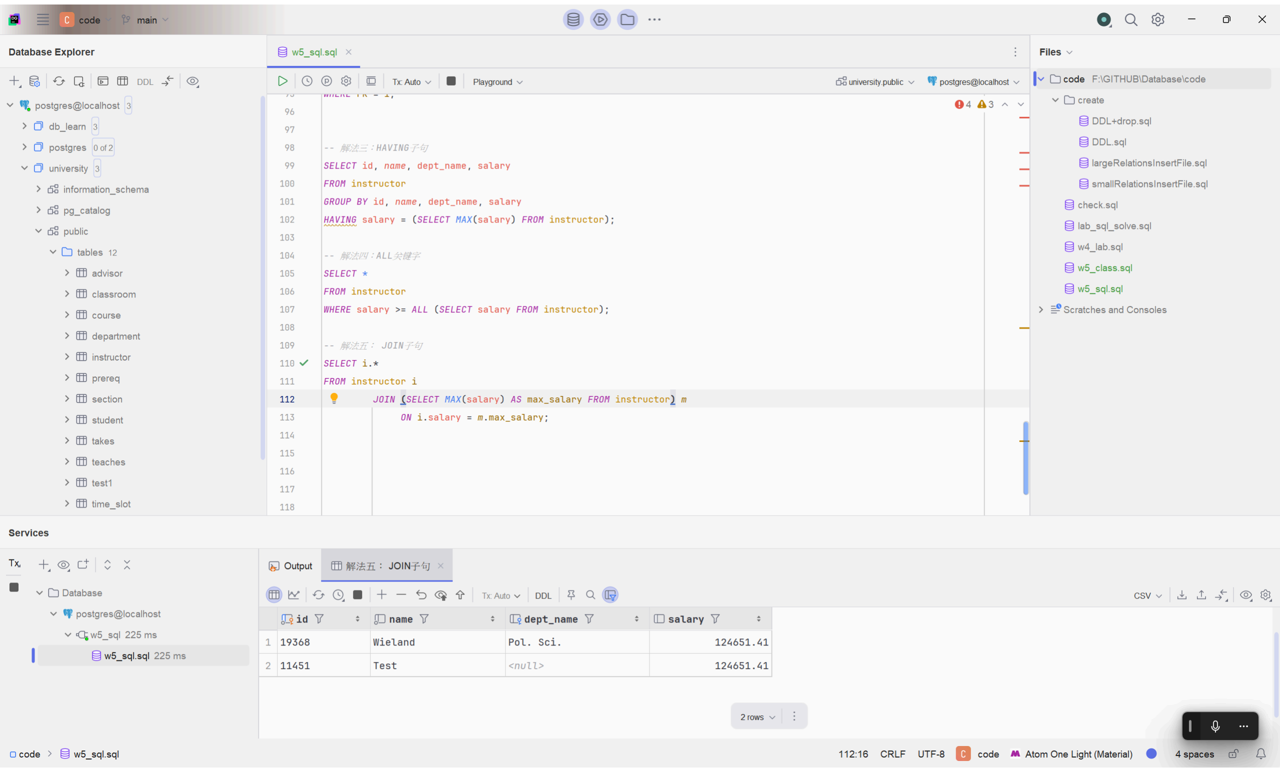
```sql
INSERT INTO instructor (id, name, dept_name, salary)
VALUES (11451, 'Test', NULL, 124651.41);
```

接下来,展示有多个最高工资的员工时可使用的解法:(即去掉第一问的解法一、三、四)

```sql
-- 解法一: WHERE子句
SELECT *
FROM instructor
WHERE salary = (SELECT MAX(salary) FROM instructor);

-- 解法二: 窗口函数RANK()
SELECT id, name, dept_name, salary
FROM (SELECT *,
             RANK() OVER (ORDER BY salary DESC) AS rk
      FROM instructor) a
WHERE rk = 1;


-- 解法三: HAVING子句
SELECT id, name, dept_name, salary
FROM instructor
GROUP BY id, name, dept_name, salary
HAVING salary = (SELECT MAX(salary) FROM instructor);

-- 解法四: ALL关键字
SELECT *
FROM instructor
WHERE salary >= ALL (SELECT salary FROM instructor);

-- 解法五:  JOIN子句
SELECT i.*
```

```sql
FROM instructor i
         JOIN (SELECT MAX(salary) AS max_salary FROM instructor) m
              ON i.salary = m.max_salary;

```

经过验证,上述所有解法输出相同



```sql
-- 解法三：HAVING子句
SELECT id, name, dept_name, salary
FROM instructor
GROUP BY id, name, dept_name, salary
HAVING salary = (SELECT MAX(salary) FROM instructor);

-- 解法四：ALL关键字
SELECT *
FROM instructor
WHERE salary >= ALL (SELECT salary FROM instructor);

-- 解法五： JOIN子句
SELECT i.*
FROM instructor i
         JOIN (SELECT MAX(salary) AS max_salary FROM instructor) m
              ON i.salary = m.max_salary;
```

	id	name	dept_name	salary
1	19368	Wieland	Pol. Sci.	124651.41
2	11451	Test	<null>	124651.41

3. 首先经过测试这四句全部返回 TRUE

- 第一句

```sql
SELECT 1 IN (1);
```

检查标量 1 是否在只有1这个元素的集合 {1} 中, 显然满足, 返回 TRUE

- 第二句

```sql
SELECT 1 = (1);
```

检查标量 1 是否等于(1),由于单元素将被视为标量而不是元组, 因此相等, 返回 TRUE

- 第三句

```sql
SELECT (1, 2) = (1, 2);
```

检查元组 (1, 2) 是否等于元组 (1,2), 显然满足, 返回 TRUE

- 第四句

```sql
SELECT (1) IN (1, 2);
```

检查标量 1 是否在集合 {1, 2} 中, 显然满足, 返回 TRUE