

数据库原理实验报告

(2023-2024 年度第二学期)

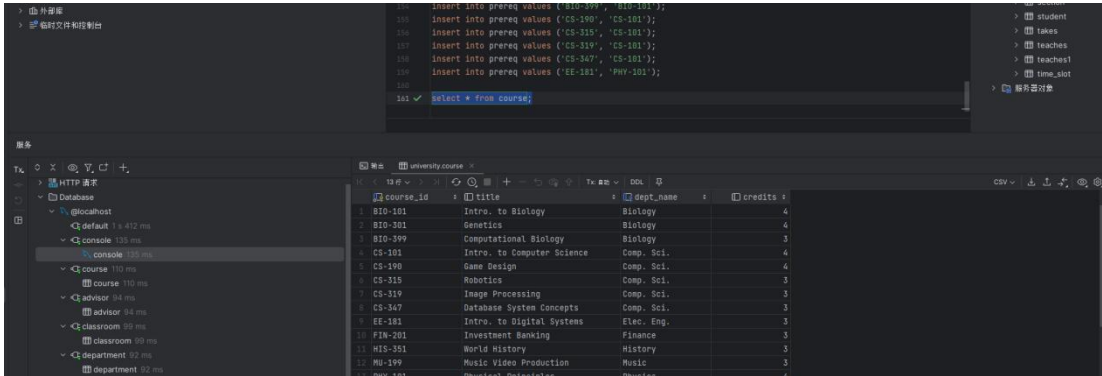
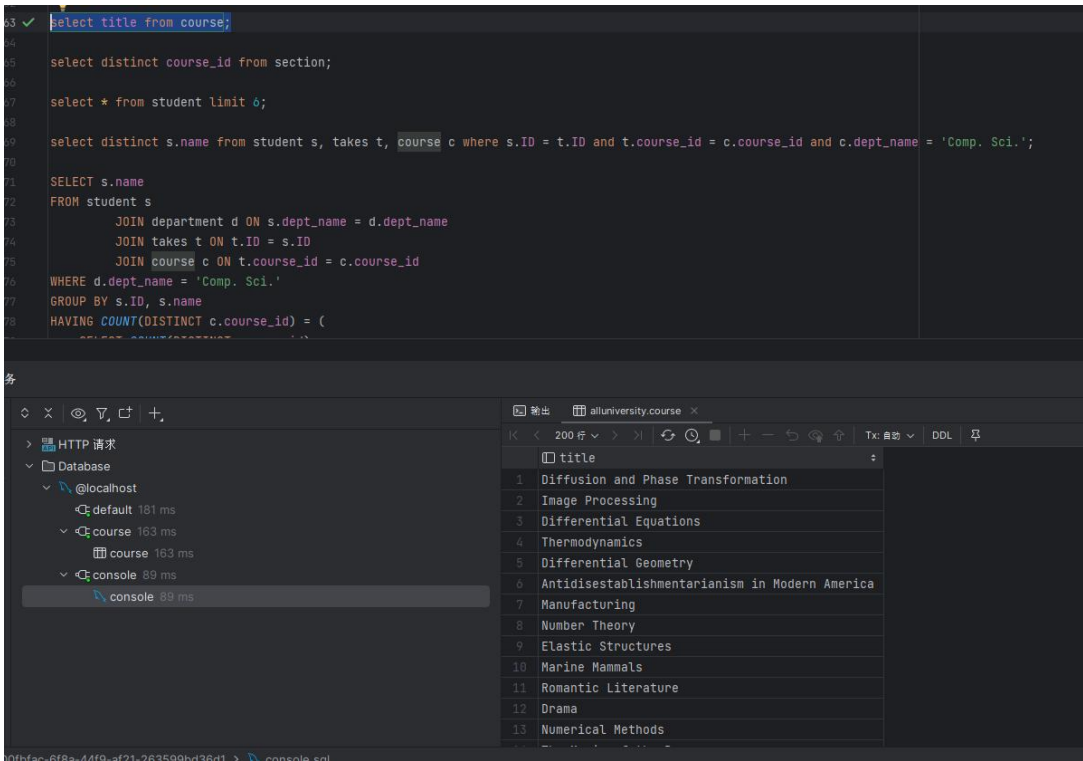
学院：_____软件学院_____

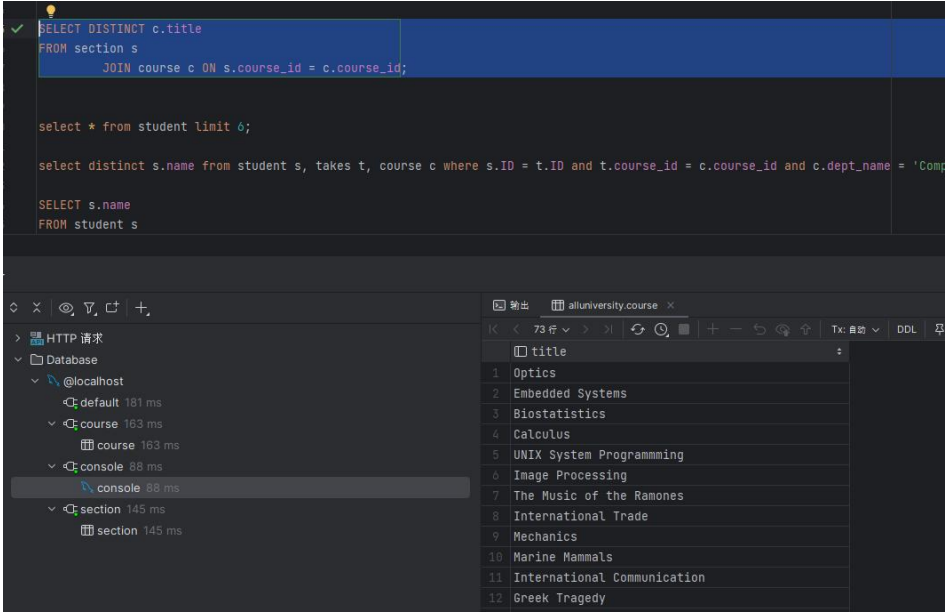
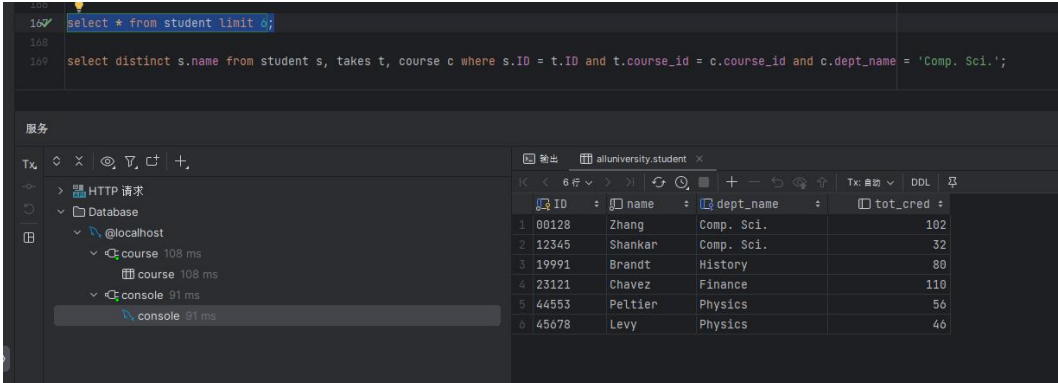
专业：_____软件工程_____

班级：_____552104_____

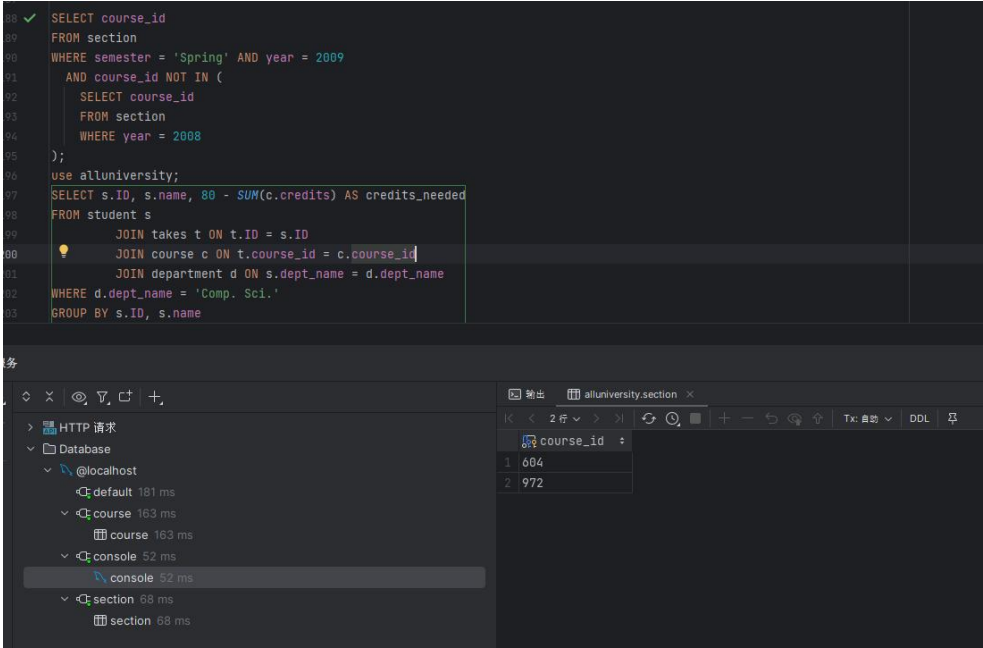
学号：_____55210425_____

姓名：_____朱家顺_____

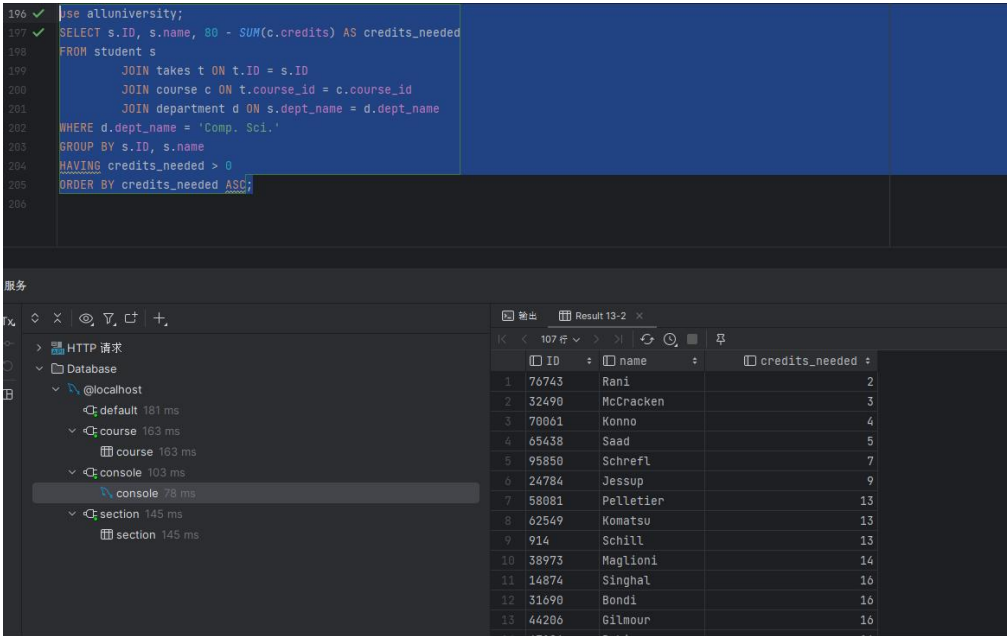
实 验 环 境	Windows 11 家庭中文版（22000.1936）;mysql Ver 8.0.31 for Win64 on x86_64 (MySQL Community Server - GPL)
实 验 题 目 1	从课程表（course）中查询所有课程信息。
答案	select * from course;
效果 截 图	
实 验 题 目 2	从课程表中（course）查询课程名。
答案	SELECT title FROM course;
效果 截 图	
实 验 题	从课程段（section）表中查询课程名称，要求消除值相同的那些行。

目 3	
答案	<pre>SELECT DISTINCT c.title FROM section s JOIN course c ON s.course_id = c.course_id;</pre>
效果截图	
实验题目 4	从学生（student）表中查询所有的信息，要求只显示查询结果的前 6 行数据。
答案	<pre>select * from student limit 6;</pre>
效果截图	
实验题目 5	查询选了所有计算机学院开设课程的学生的姓名。
答案	<pre>SELECT s.name FROM student s JOIN department d ON s.dept_name = d.dept_name</pre>

	<pre>JOIN takes t ON t.ID = s.ID JOIN course c ON t.course_id = c.course_id WHERE d.dept_name = 'Comp. Sci.' GROUP BY s.ID, s.name HAVING COUNT(DISTINCT c.course_id) = (SELECT COUNT(DISTINCT course_id) FROM course WHERE dept_name = 'Comp. Sci.');</pre>
效果截图	 A screenshot of a SQL IDE interface. The top pane shows a SQL query with line numbers 70 to 82. The query is a complex JOIN statement filtering for 'Comp. Sci.' and grouping by student ID and name, with a HAVING clause that compares the count of distinct course IDs to a subquery. The bottom pane shows the 'Output' window with a table of results containing four rows: Meyl, Winter, Senn, and Ikeda.
实验题目 6	查询 2019 年春季开课,但 2018 年不开课的课程的编号。(改为 2009 和 2008)
答案	<pre>SELECT course_id FROM section WHERE semester = 'Spring' AND year = 2019 AND course_id NOT IN (SELECT course_id FROM section</pre>

	<div>WHERE year = 2018</div> <div>);</div>
效果截图	<div></div>
实验题目 7	<div>假设毕业要求为修够 80 学分，请统计计算机学院内学生距离毕业要求还差多少学分，并按所差分数的升序排列。</div>
答案	<div>SELECT s. ID, s. name, 80 - SUM(c. credits) AS credits_needed</div> <div>FROM student s</div> <div>JOIN takes t ON t. ID = s. ID</div> <div>JOIN course c ON t. course_id = c. course_id</div> <div>JOIN department d ON s. dept_name = d. dept_name</div> <div>WHERE d. dept_name = 'Comp. Sci.'</div> <div>GROUP BY s. ID, s. name</div> <div>HAVING credits_needed > 0</div> <div>ORDER BY credits_needed ASC;</div>

效果截图



The screenshot shows a SQL IDE with a query editor and a results pane. The query in the editor is as follows:

```
196 use alluniversity;
197 SELECT s.ID, s.name, 80 - SUM(c.credits) AS credits_needed
198 FROM student s
199 JOIN takes t ON t.ID = s.ID
200 JOIN course c ON t.course_id = c.course_id
201 JOIN department d ON s.dept_name = d.dept_name
202 WHERE d.dept_name = 'Comp. Sci.'
203 GROUP BY s.ID, s.name
204 HAVING credits_needed > 0
205 ORDER BY credits_needed ASC;
```

The results pane shows a table with 3 columns: ID, name, and credits_needed. The table contains 13 rows of data.

ID	name	credits_needed
1	76743 Rani	2
2	32490 McCracken	3
3	70061 Konno	4
4	65438 Saad	5
5	95850 Schrefl	7
6	24784 Jessup	9
7	58081 Pelletier	13
8	62549 Komatsu	13
9	914 Schill	13
10	38973 Maglioni	14
11	14874 Singhal	16
12	31690 Bondi	16
13	44286 Gilmour	16

实验题目 8

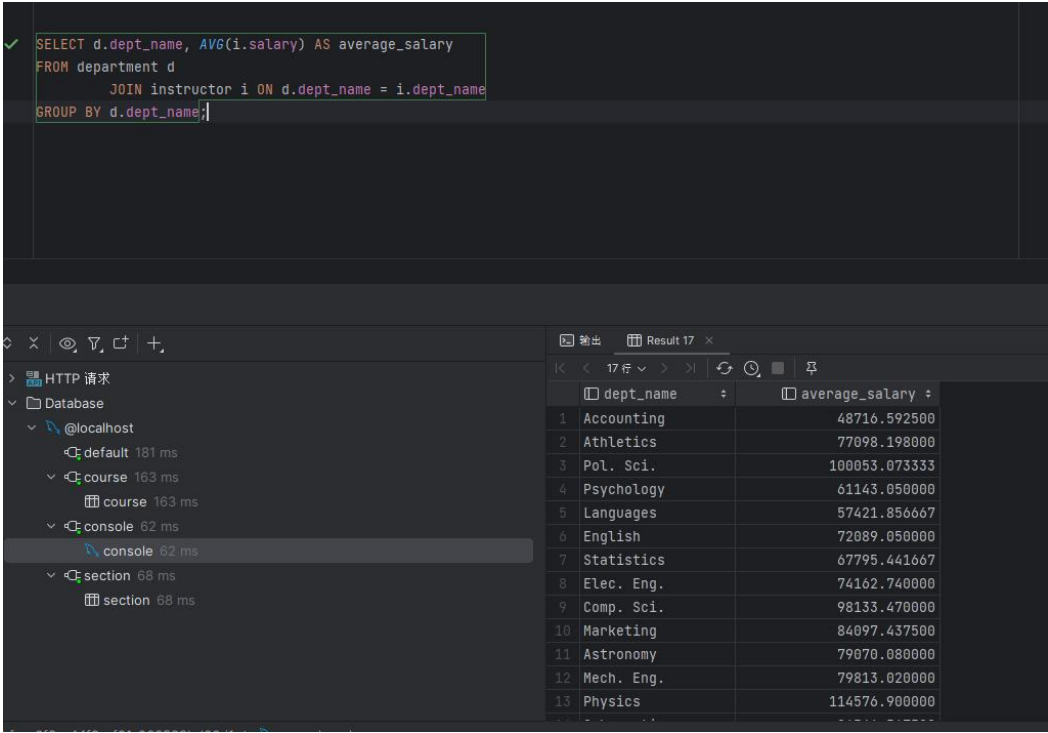
统计 2019 年春季所开课程段选课人数的最大值。

答案

```
SELECT MAX(student_count) AS max_enrollment
FROM (
    SELECT COUNT(*) AS student_count
    FROM section
    WHERE semester = 'Spring' AND year = 2019
    GROUP BY course_id, sec_id, semester, year
) AS enrollment_counts;
```

<div>效果截图</div>	 <p>The screenshot shows a SQL IDE interface. The top pane displays a SQL query: <code>SELECT MAX(student_count) AS max_enrollment FROM (SELECT COUNT(*) AS student_count FROM section WHERE semester = 'Spring' AND year = 2009 GROUP BY course_id, sec_id, semester, year) AS enrollment_counts;</code>. The bottom pane is split into two sections. The left section, titled '服务' (Services), shows a tree view of database connections and tables, including 'default', 'course', 'console', and 'section'. The right section, titled '输出' (Output), shows the query results in a table with one row and one column, 'max_enrollment', with the value '1'.</p>
<div>实验题目 9</div>	<p>统计各个学院老师的平均年薪。</p>
<div>答案</div>	<pre>SELECT d.dept_name, AVG(i.salary) AS average_salary FROM department d JOIN instructor i ON d.dept_name = i.dept_name GROUP BY d.dept_name;</pre>

效果截图



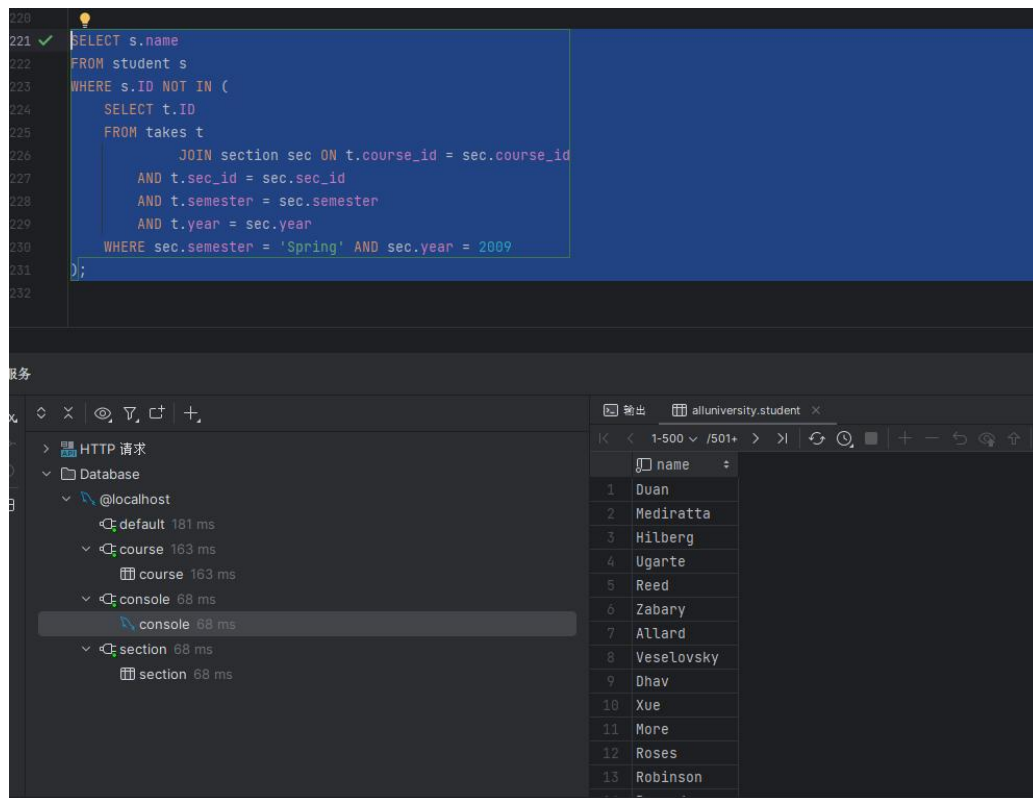
实验题目 10

请输出没有选择 2019 年春季开课的课程段的学生的姓名。（2009）

答案

```
SELECT s.name
FROM student s
WHERE s.ID NOT IN (
    SELECT t.ID
    FROM takes t
          JOIN section sec ON t.course_id = sec.course_id
          AND t.sec_id = sec.sec_id
          AND t.semester = sec.semester
          AND t.year = sec.year
    WHERE sec.semester = 'Spring' AND sec.year = 2009
);
```


效果截图



实验题目 11

请输出“张三”指导的学生在 2019 年春季开设课程段中所获得的总学分。
（我们默认只要选择某个 course 的一个 section，就可以获得这个 course 的学分；若选择同一个 course 的多个 section，也只获得 1 次这个 course 的学分）。（Mingoz, 2009, Fall）

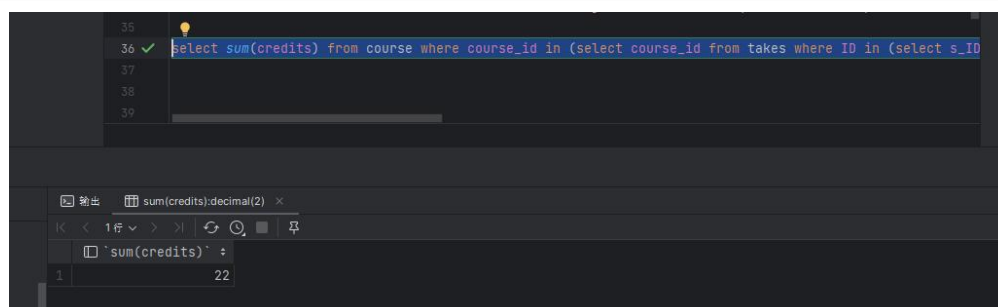
答案

```

select sum(credits) from course where course_id in (select course_id
from takes where ID in (select s_ID as ID from advisor where (s_ID
in (select ID from takes where year=2009 and semester='Fall')) and
(i_ID in (select ID from instructor where name='Mingoz')) and
semester='Fall' and year=2009);

```

效果截图



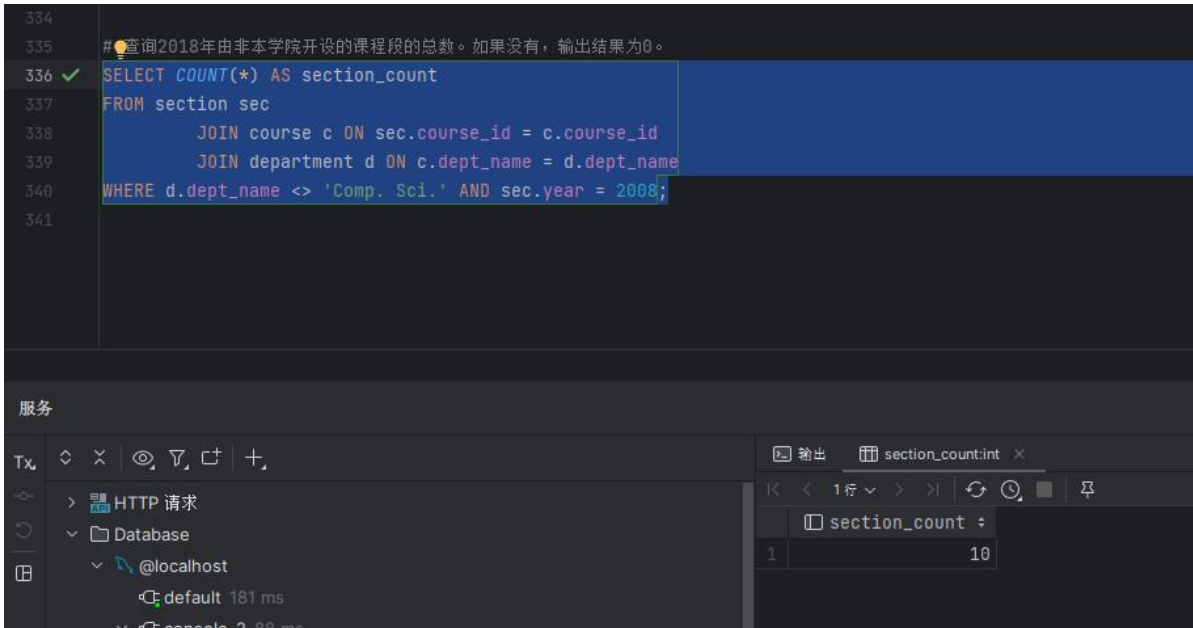
实 验 题 目 12	请输出“数据库系统原理”和“离散数学”的共同的先修课程的 ID。（Cost Accounting 和 Environmental Law）
答案	<pre>select distinct prereq_id from prereq where course_id in (select course_id from course where title='Cost Accounting' or title='Environmental Law');</pre>
效 果 截 图	 <p>The screenshot shows a SQL IDE with a query editor and a results pane. The query in the editor is: <code>select distinct prereq_id from prereq where course_id in (select course_id from course where title='Cost Accounting' or title='Environmental Law');</code>. The results pane shows the output of the query, which is a list of prereq_id values: 130 and 324.</p>
实 验 题 目 13	请统计 2016 年-2018 年, 计算机学院每年开设的课程段的数量。(2006-2008)
答案	<pre>SELECT year, COUNT(*) AS course_count FROM section sec JOIN course c ON sec.course_id = c.course_id JOIN department d ON c.dept_name = d.dept_name WHERE d.dept_name = 'Comp. Sci.' AND year BETWEEN 2006 AND 2008 GROUP BY year;</pre>

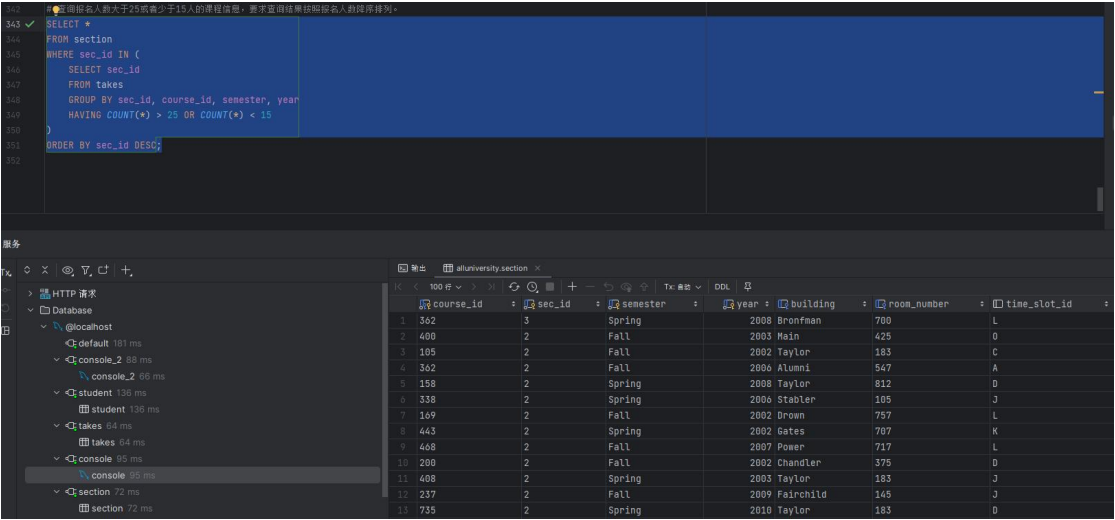
<p>效果截图</p>	
<p>实验题目 14</p>	<p>查询课程编号不为“004”、“007”、“013”的课程编号和课程名称。</p>
<p>答案</p>	<pre>SELECT course_id, title FROM course WHERE course_id NOT IN ('004', '007', '013');</pre>
<p>效果截图</p>	
<p>实验题目 15</p>	<p>查询课程名以字母 D 开始，以“e”结尾的课程信息。(G y)</p>
<p>答案</p>	<pre>SELECT * FROM course WHERE title LIKE 'G%y';</pre>

效果截图	
实验题目 16	查询课程名以“制作”两字作为中间字的课程信息。（要求“制作”不做开头和结尾）（and）
答案	<pre>SELECT * FROM course WHERE title LIKE '%and%' ;</pre>
效果截图	
实验题目 17	查询姓名第二个字为“宝”的学生信息。（a）
答案	<pre>SELECT * FROM student WHERE name LIKE '_a%' ;</pre>

效果截图	 <p>The screenshot shows a SQL query in the editor: <code>SELECT * FROM student WHERE name LIKE 'a%';</code>. The results pane displays a table with columns ID, name, dept_name, and tot_cred, containing 12 rows of student data.</p>
实验题目 18	查询不姓“刘”的学生信息。(A)
答案	<pre>SELECT * FROM student WHERE name NOT LIKE 'A%';</pre>
效果截图	 <p>The screenshot shows a SQL query in the editor: <code>SELECT * FROM student WHERE name NOT LIKE 'A%';</code>. The results pane displays a table with columns ID, name, dept_name, and tot_cred, containing 13 rows of student data.</p>
实验题目 19	查询那些在 2018 年有至少两个课程段的课程的 ID。(2008)
答案	<pre>SELECT course_id FROM section WHERE year = 2008 HAVING COUNT(*) >= 2;</pre>

<p>效果截图</p>	
<p>实验题目 20</p>	<p>查询计算机学院学生选了非本学院老师开设的属于本学院的课程的情况，统计这些学生的 ID, 姓名，总学分。</p>
<p>答案</p>	<p>select distinct student.ID,name,tot_cred from student join (select * from takes natural join course) as abc where student.ID=abc.ID and student.dept_name!=abc.dept_name and student.dept_name='Comp.Sci.';</p>
<p>效果截图</p>	
<p>实验题目 21</p>	<p>查询 2019 年春季开设的，选课人数少于 25 并且多于 15 人的课程段信息。</p>
<p>答案</p>	<p>select count(distinct(course_id)) from course natural join section where year=2008 and dept_name != 'Comp.Sci.';</p>
<p>效果截图</p>	

实验题目 22	查询 2018 年由非本学院开设的课程段的总数。如果没有，输出结果为 0。		
答案	<pre>SELECT COUNT(*) AS section_count FROM section sec JOIN course c ON sec.course_id = c.course_id JOIN department d ON c.dept_name = d.dept_name WHERE d.dept_name <> 'Comp. Sci.' AND sec.year = 2008;</pre>		
效果截图	 The screenshot shows a SQL IDE interface. The top pane displays a SQL query: <pre>SELECT COUNT(*) AS section_count FROM section sec JOIN course c ON sec.course_id = c.course_id JOIN department d ON c.dept_name = d.dept_name WHERE d.dept_name <> 'Comp. Sci.' AND sec.year = 2008;</pre> The bottom pane shows the execution results in a table with one row: <table><tr><th>section_count</th></tr><tr><td>10</td></tr></table>	section_count	10
section_count			
10			
实验题目 23	查询报名人数大于 25 或者少于 15 人的课程信息，要求查询结果按照报名人数降序排列。		
答案	<pre>SELECT * FROM section WHERE sec_id IN (SELECT sec_id FROM takes GROUP BY sec_id, course_id, semester, year HAVING COUNT(*) > 25 OR COUNT(*) < 15</pre>		

	<p>)</p> <p>ORDER BY sec_id DESC;</p>
效果截图	
实验题目 24	查询与计算机学院同处于一座大楼的其他学院的老师的平均工资。（Finance 经济学院）
答案	<pre>SELECT AVG(salary) AS avg_salary FROM instructor WHERE dept_name IN (SELECT dept_name FROM department WHERE building IN (SELECT building FROM department WHERE dept_name = 'Finance')) AND dept_name <> 'Finance';</pre>

效果截图	
实验题目 25	给在“匡亚明”大楼内办公的老师工资增加到原来工资的 1.5 倍。(Palmer)
答案	<pre>UPDATE instructor SET salary = salary * 1.5 WHERE dept_name IN (SELECT dept_name FROM department WHERE building = 'Palmer');</pre>
效果截图	

```
[2023-06-08 20:12:51] 在 27 ms (execution: 6 ms, fetching: 21 ms) 内检索到从 1 开始的 1 行
alluniversity> UPDATE instructor
SET salary = salary * 1.5
WHERE dept_name IN (
SELECT dept_name
FROM department
WHERE building = 'Palmer'
)

[2023-06-08 20:15:33] [01000][1265] Data truncated for column 'salary' at row 2
[2023-06-08 20:15:33] [01000][1265] Data truncated for column 'salary' at row 3
[2023-06-08 20:15:33] [01000][1265] Data truncated for column 'salary' at row 5
[2023-06-08 20:15:33] 27 ms 中有 5 行受到影响
```

实验题目 26

统计各个学院的学生数量。

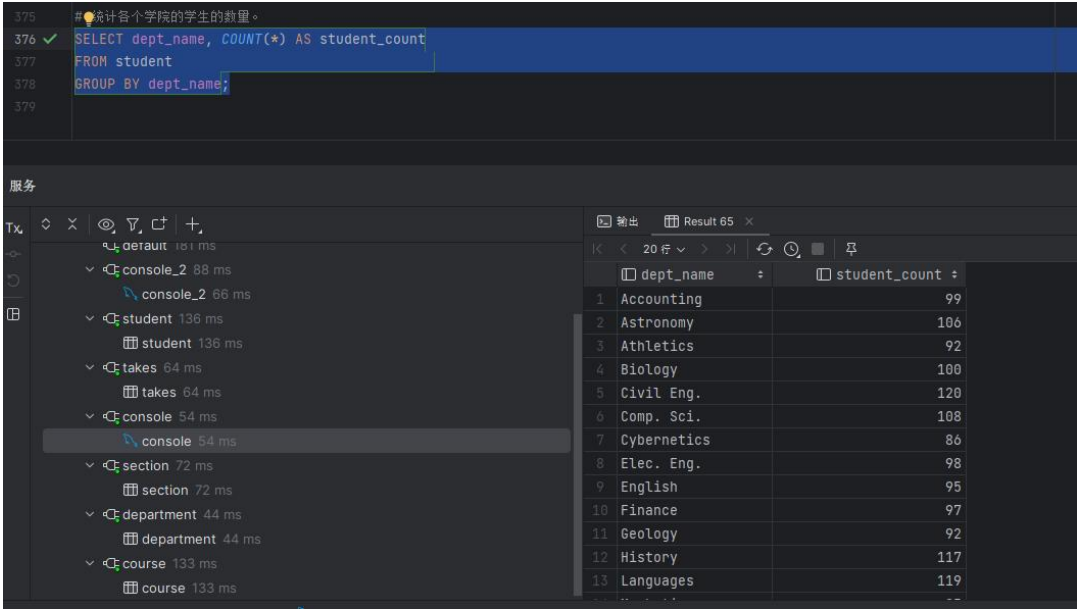
答案

SELECT dept_name, COUNT(*) AS student_count

FROM student

GROUP BY dept_name;

效果图



dept_name	student_count
Accounting	99
Astronomy	106
Athletics	92
Biology	100
Civil Eng.	120
Comp. Sci.	108
Cybernetics	86
Elec. Eng.	98
English	95
Finance	97
Geology	92
History	117
Languages	119

实验题目 27

统计计算机学院中所获总学分排名前 10 位的学生信息。

答案

SELECT s.ID, s.name, s.dept_name, SUM(c.credits) AS total_credits

FROM student s

JOIN takes t ON s.ID = t.ID

JOIN section sec ON t.course_id = sec.course_id

AND t.sec_id = sec.sec_id

AND t.semester = sec.semester

AND t.year = sec.year

JOIN course c ON sec.course_id = c.course_id

JOIN department d ON s.dept_name = d.dept_name

WHERE d.dept_name = 'Comp. Sci.'

GROUP BY s.ID, s.name, s.dept_name

ORDER BY total_credits DESC

LIMIT 10;

效果截图

380 #按计算机学院中所获总学分排名前10位的学生信息。

381 ✓ SELECT s.ID, s.name, s.dept_name, SUM(c.credits) AS total_credits

382 FROM student s

383 JOIN takes t ON s.ID = t.ID

384 JOIN section sec ON t.course_id = sec.course_id

385 AND t.sec_id = sec.sec_id

386 AND t.semester = sec.semester

387 AND t.year = sec.year

388 JOIN course c ON sec.course_id = c.course_id

389 JOIN department d ON s.dept_name = d.dept_name

390 WHERE d.dept_name = 'Comp. Sci.'

391 GROUP BY s.ID, s.name, s.dept_name

392 ORDER BY total_credits DESC

393 LIMIT 10;

394

395 #

服务

default 101 ms

console_2 88 ms

console_2 86 ms

student 136 ms

student 136 ms

takes 64 ms

takes 64 ms

console 67 ms

console 67 ms

section 72 ms

section 72 ms

department 44 ms

department 44 ms

输出 Result 66

ID	name	dept_name	total_credits
1 47379	Wakamiya	Comp. Sci.	86
2 76743	Rani	Comp. Sci.	78
3 32490	McCracken	Comp. Sci.	77
4 70061	Konno	Comp. Sci.	76
5 65438	Saad	Comp. Sci.	75
6 95850	Schrefl	Comp. Sci.	73
7 24784	Jessup	Comp. Sci.	71
8 62549	Komatsu	Comp. Sci.	67
9 58081	Pelletier	Comp. Sci.	67
10 914	Schill	Comp. Sci.	67

实验题目 28

查询“李四”老师在 2017-2019 年开设课程段的数量。(2007-2009, ” Dale”)

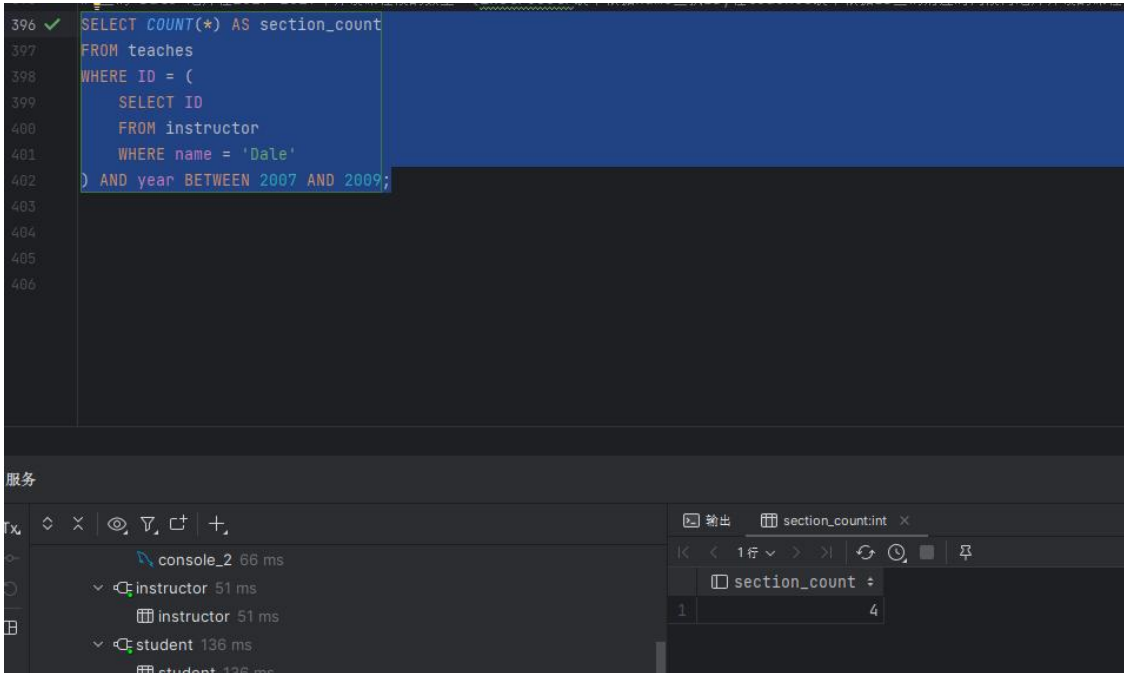
答案

SELECT COUNT(*) AS section_count

FROM teaches

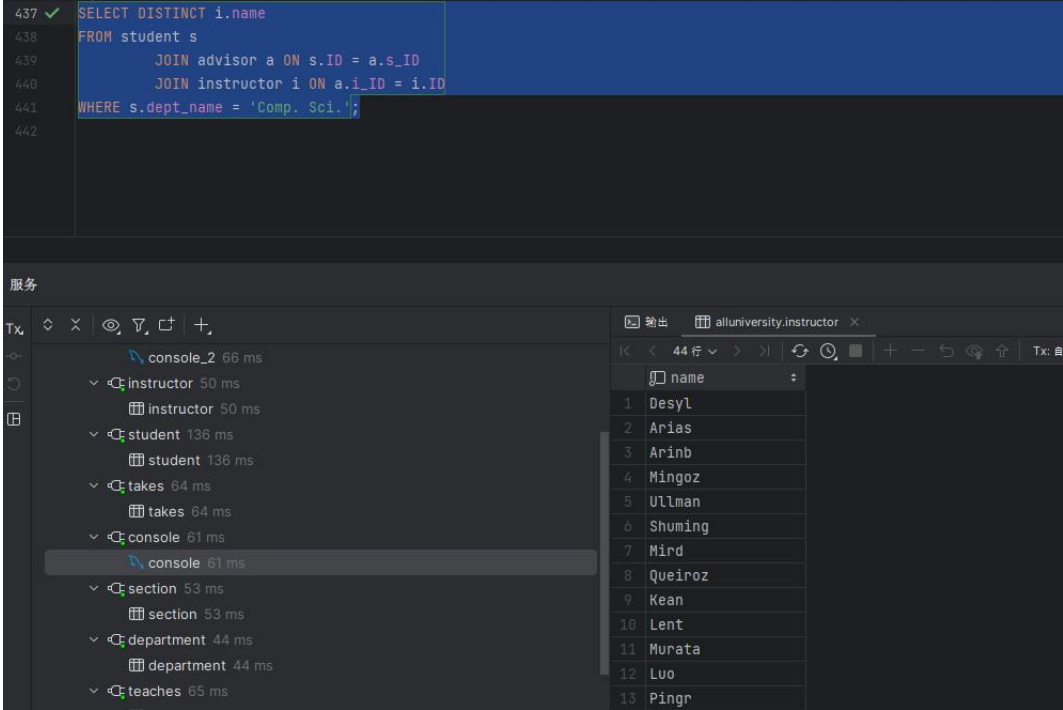
WHERE ID = (

SELECT ID

	<pre>FROM instructor WHERE name = 'Dale') AND year BETWEEN 2007 AND 2009;</pre>		
效果截图	 <p>The screenshot shows a SQL IDE with a query editor and a results pane. The query in the editor is:</p> <pre>396 SELECT COUNT(*) AS section_count 397 FROM teaches 398 WHERE ID = (399 SELECT ID 400 FROM instructor 401 WHERE name = 'Dale' 402) AND year BETWEEN 2007 AND 2009; 403 404 405 406</pre> <p>The results pane shows a table with one row and one column:</p> <table><tr><th>section_count</th></tr><tr><td>4</td></tr></table>	section_count	4
section_count			
4			
实验题目 29	查询计算机学院老师中比生物学院工资最高的老师工资低，但是比生物学院最低工资高的老师的信息。（生物学院，英语学院）		
答案	<pre>SELECT * FROM instructor WHERE dept_name = 'Biology' AND salary < (SELECT MAX(salary) FROM instructor WHERE dept_name = 'English') AND salary > (SELECT MIN(salary) FROM instructor WHERE dept_name = 'English'</pre>		

) ;
效果截图	<div><div><div>405 ✓ SELECT * 406 FROM instructor 407 WHERE dept_name = 'Biology' 408 AND salary < (409 SELECT MAX(salary) 410 FROM instructor 411 WHERE dept_name = 'English' 412) AND salary > (413 SELECT MIN(salary) 414 FROM instructor 415 WHERE dept_name = 'English' 416); 417 418 419</div><div>服务</div><div><div>Tx</div><div><div>console_2 66 ms</div><div><div>instructor 50 ms</div><div>instructor 50 ms</div><div>student 136 ms</div><div>student 136 ms</div></div></div><div>输出 alluniversity.instructor x</div><div><div>1 行</div><div>1</div><div>2</div><div>3</div><div>4</div><div>5</div><div>6</div><div>7</div><div>8</div><div>9</div><div>10</div><div>11</div><div>12</div><div>13</div><div>14</div><div>15</div><div>16</div><div>17</div><div>18</div><div>19</div><div>20</div><div>21</div><div>22</div><div>23</div><div>24</div><div>25</div><div>26</div><div>27</div><div>28</div><div>29</div><div>30</div><div>31</div><div>32</div><div>33</div><div>34</div><div>35</div><div>36</div><div>37</div><div>38</div><div>39</div><div>40</div><div>41</div><div>42</div><div>43</div><div>44</div><div>45</div><div>46</div><div>47</div><div>48</div><div>49</div><div>50</div><div>51</div><div>52</div><div>53</div><div>54</div><div>55</div><div>56</div><div>57</div><div>58</div><div>59</div><div>60</div><div>61</div><div>62</div><div>63</div><div>64</div><div>65</div><div>66</div><div>67</div><div>68</div><div>69</div><div>70</div><div>71</div><div>72</div><div>73</div><div>74</div><div>75</div><div>76</div><div>77</div><div>78</div><div>79</div><div>80</div><div>81</div><div>82</div><div>83</div><div>84</div><div>85</div><div>86</div><div>87</div><div>88</div><div>89</div><div>90</div><div>91</div><div>92</div><div>93</div><div>94</div><div>95</div><div>96</div><div>97</div><div>98</div><div>99</div><div>100</div><div>101</div><div>102</div><div>103</div><div>104</div><div>105</div><div>106</div><div>107</div><div>108</div><div>109</div><div>110</div><div>111</div><div>112</div><div>113</div><div>114</div><div>115</div><div>116</div><div>117</div><div>118</div><div>119</div><div>120</div><div>121</div><div>122</div><div>123</div><div>124</div><div>125</div><div>126</div><div>127</div><div>128</div><div>129</div><div>130</div><div>131</div><div>132</div><div>133</div><div>134</div><div>135</div><div>136</div><div>137</div><div>138</div><div>139</div><div>140</div><div>141</div><div>142</div><div>143</div><div>144</div><div>145</div><div>146</div><div>147</div><div>148</div><div>149</div><div>150</div><div>151</div><div>152</div><div>153</div><div>154</div><div>155</div><div>156</div><div>157</div><div>158</div><div>159</div><div>160</div><div>161</div><div>162</div><div>163</div><div>164</div><div>165</div><div>166</div><div>167</div><div>168</div><div>169</div><div>170</div><div>171</div><div>172</div><div>173</div><div>174</div><div>175</div><div>176</div><div>177</div><div>178</div><div>179</div><div>180</div><div>181</div><div>182</div><div>183</div><div>184</div><div>185</div><div>186</div><div>187</div><div>188</div><div>189</div><div>190</div><div>191</div><div>192</div><div>193</div><div>194</div><div>195</div><div>196</div><div>197</div><div>198</div><div>199</div><div>200</div><div>201</div><div>202</div><div>203</div><div>204</div><div>205</div><div>206</div><div>207</div><div>208</div><div>209</div><div>210</div><div>211</div><div>212</div><div>213</div><div>214</div><div>215</div><div>216</div><div>217</div><div>218</div><div>219</div><div>220</div><div>221</div><div>222</div><div>223</div><div>224</div><div>225</div><div>226</div><div>227</div><div>228</div><div>229</div><div>230</div><div>231</div><div>232</div><div>233</div><div>234</div><div>235</div><div>236</div><div>237</div><div>238</div><div>239</div><div>240</div><div>241</div><div>242</div><div>243</div><div>244</div><div>245</div><div>246</div><div>247</div><div>248</div><div>249</div><div>250</div><div>251</div><div>252</div><div>253</div><div>254</div><div>255</div><div>256</div><div>257</div><div>258</div><div>259</div><div>260</div><div>261</div><div>262</div><div>263</div><div>264</div><div>265</div><div>266</div><div>267</div><div>268</div><div>269</div><div>270</div><div>271</div><div>272</div><div>273</div><div>274</div><div>275</div><div>276</div><div>277</div><div>278</div><div>279</div><div>280</div><div>281</div><div>282</div><div>283</div><div>284</div><div>285</div><div>286</div><div>287</div><div>288</div><div>289</div><div>290</div><div>291</div><div>292</div><div>293</div><div>294</div><div>295</div><div>296</div><div>297</div><div>298</div><div>299</div><div>300</div><div>301</div><div>302</div><div>303</div><div>304</div><div>305</div><div>306</div><div>307</div><div>308</div><div>309</div><div>310</div><div>311</div><div>312</div><div>313</div><div>314</div><div>315</div><div>316</div><div>317</div><div>318</div><div>319</div><div>320</div><div>321</div><div>322</div><div>323</div><div>324</div><div>325</div><div>326</div><div>327</div><div>328</div><div>329</div><div>330</div><div>331</div><div>332</div><div>333</div><div>334</div><div>335</div><div>336</div><div>337</div><div>338</div><div>339</div><div>340</div><div>341</div><div>342</div><div>343</div><div>344</div><div>345</div><div>346</div><div>347</div><div>348</div><div>349</div><div>350</div><div>351</div><div>352</div><div>353</div><div>354</div><div>355</div><div>356</div><div>357</div><div>358</div><div>359</div><div>360</div><div>361</div><div>362</div><div>363</div><div>364</div><div>365</div><div>366</div><div>367</div><div>368</div><div>369</div><div>370</div><div>371</div><div>372</div><div>373</div><div>374</div><div>375</div><div>376</div><div>377</div><div>378</div><div>379</div><div>380</div><div>381</div><div>382</div><div>383</div><div>384</div><div>385</div><div>386</div><div>387</div><div>388</div><div>389</div><div>390</div><div>391</div><div>392</div><div>393</div><div>394</div><div>395</div><div>396</div><div>397</div><div>398</div><div>399</div><div>400</div><div>401</div><div>402</div><div>403</div><div>404</div></div></div></div></div>
实验题目 30	请统计 2019 年计算机学院比生物学院多开设了几个课程段。(2009, Civil Eng., Athletics)
答案	<pre>SELECT SUM(CASE WHEN c.dept_name = 'Civil Eng.' THEN 1 ELSE 0 END) AS civil_eng_count, SUM(CASE WHEN c.dept_name = 'Athletics' THEN 1 ELSE 0 END) AS ath_count, SUM(CASE WHEN c.dept_name = 'Civil Eng.' THEN 1 ELSE 0 END) - SUM(CASE WHEN c.dept_name = 'Athletics' THEN 1 ELSE 0 END) AS difference FROM section sec JOIN course c ON sec.course_id = c.course_id WHERE sec.year = 2009;</pre>

<div>效果截图</div>	
<div>实验题目 31</div>	<div>请统计 2019 年春季，没有学生选课的课程段的数量。若没有，请输出 0。</div>
<div>答案</div>	<pre>SELECT COUNT(*) AS section_count FROM section WHERE sec_id NOT IN (SELECT sec_id FROM takes WHERE year = 2009 AND semester = 'Spring');</pre>
<div>效果截图</div>	
<div>实验题目 32</div>	<div>请输出计算机学院学生的指导老师的姓名，去掉重复信息。</div>

答案	<pre>SELECT DISTINCT i.name FROM student s JOIN advisor a ON s.ID = a.s_ID JOIN instructor i ON a.i_ID = i.ID WHERE s.dept_name = 'Comp. Sci.';</pre>
效果截图	 <p>The screenshot displays a SQL query in a dark-themed IDE. The query is: <code>SELECT DISTINCT i.name FROM student s JOIN advisor a ON s.ID = a.s_ID JOIN instructor i ON a.i_ID = i.ID WHERE s.dept_name = 'Comp. Sci.';</code>. Below the query editor, the 'Services' pane shows a tree of database objects with execution times. The 'Results' pane on the right shows a table with one column, 'name', and 13 rows of instructor names: Desyl, Arias, Arinb, Mingo, Ullman, Shuming, Mird, Queiroz, Kean, Lent, Murata, Luo, and Pingr.</p>
实验题目 33	将低于整个学校的平均预算的各个学院的预算提高到原来的 1.2 倍。
答案	<pre>UPDATE department d JOIN (SELECT AVG(budget) AS avg_budget FROM department) avg_dept SET d.budget = d.budget * 1.2 WHERE d.budget < avg_dept.avg_budget;</pre>

效果截图	<pre>[2023-06-08 20:45:51] [HY000][1093] You can't specify target table 'department' for update in FROM clause alluniversity> UPDATE department d JOIN (SELECT AVG(budget) AS avg_budget FROM department) avg_dept SET d.budget = d.budget * 1.2 WHERE d.budget < avg_dept.avg_budget [2023-06-08 20:46:39] [01000][1265] Data truncated for column 'budget' at row 1 [2023-06-08 20:46:39] [01000][1265] Data truncated for column 'budget' at row 2 [2023-06-08 20:46:39] [01000][1265] Data truncated for column 'budget' at row 3 [2023-06-08 20:46:39] [01000][1265] Data truncated for column 'budget' at row 4 [2023-06-08 20:46:39] [01000][1265] Data truncated for column 'budget' at row 5 [2023-06-08 20:46:39] [01000][1265] Data truncated for column 'budget' at row 6 [2023-06-08 20:46:39] [01000][1265] Data truncated for column 'budget' at row 8</pre>
实验题目 34	统计“李四”老师指导的学生数量。若没有，请输出 0。(Bourrier)
答案	<pre>SELECT COUNT(*) AS student_count FROM advisor WHERE i_ID = (SELECT ID FROM instructor WHERE name = 'Bourrier');</pre>
效果截图	
实验题目 35	查询已经选了“数据库原理”这门课的所有直接先修课程的学生信息(Rock and Roll)
答案	<pre>SELECT *</pre>


```
FROM student

WHERE ID IN (

    SELECT ID

    FROM takes

    WHERE course_id IN (

        SELECT prereq_id

        FROM prereq

        WHERE course_id = (

            SELECT course_id

            FROM course

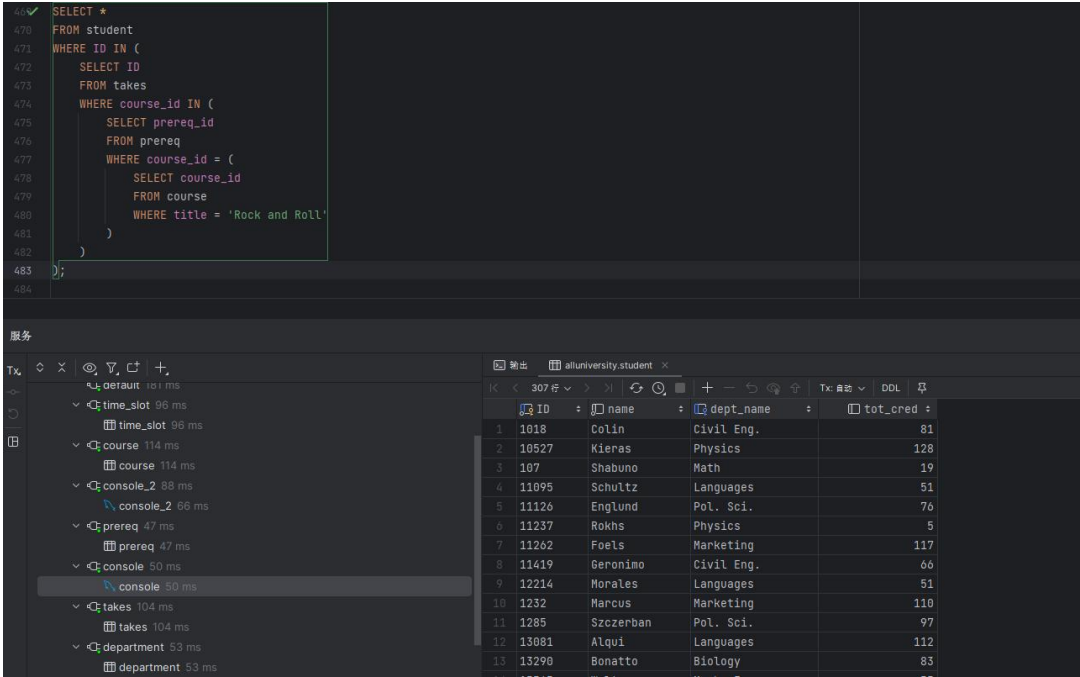
            WHERE title = 'Rock and Roll'

        )

    )

);
```

效果截图



实验题目 36

查询 2019 年春季选了自己指导教师开设的课程段的学生的姓名、指导老师的姓名和课程段的 ID。(2009,)

答案

```
SELECT s.name, i.name, t.course_id
FROM student s
      JOIN takes t ON s.ID = t.ID
      JOIN instructor i ON t.ID = i.ID
WHERE t.course_id IN (
    SELECT course_id
    FROM teaches
    WHERE ID = (
        SELECT ID
        FROM instructor
        WHERE name = 'Atanassov'
    ) AND sec_id = (
        SELECT sec_id
        FROM section
        WHERE semester = 'Spring' AND year = 2009
    )
);
```

效果截图

486 ✓
487
488
489
490
491
492
493
494
495
496

```
SELECT s.name, i.name, t.sec_id
FROM student s
      JOIN advisor a ON s.ID = a.s_ID
      JOIN instructor i ON a.i_ID = i.ID
      JOIN teaches t ON i.ID = t.ID
      JOIN takes tk ON t.course_id = tk.course_id
      AND t.sec_id = tk.sec_id
      AND t.semester = tk.semester
      AND t.year = tk.year
WHERE tk.year = 2009 AND tk.semester = 'Spring';
```

服务

Tx

default 101 ms

time_slot 96 ms

time_slot 96 ms

course 114 ms

course 114 ms

console_2 88 ms

console_2 66 ms

prereq 47 ms

prereq 47 ms

console 47 ms

console 47 ms

takes 104 ms

takes 104 ms

department 53 ms

department 53 ms

输出 Result 93

s.name ÷ i.name ÷ sec_id ÷

1 Januszewski Atanassov 1

2 Holn Atanassov 1

3 Simon Atanassov 1

4 Fournier Atanassov 1

5 Hawkins Atanassov 1

6 Leitner Atanassov 1

7 Milner Atanassov 1

8 Rougemont Atanassov 1

9 Soni Atanassov 1

10 Theodores Atanassov 1

11 Ryder Atanassov 1

12 Tsantis Atanassov 1

13 Resa Atanassov 1