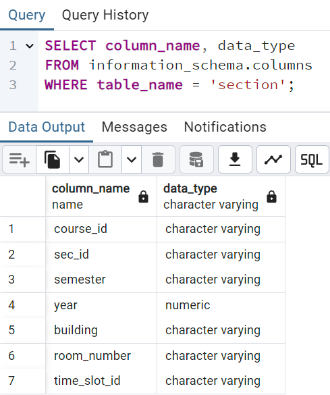
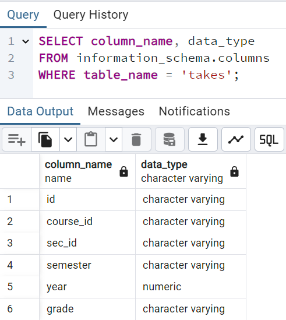
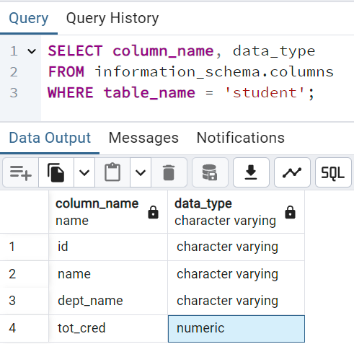
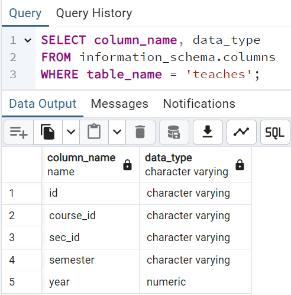
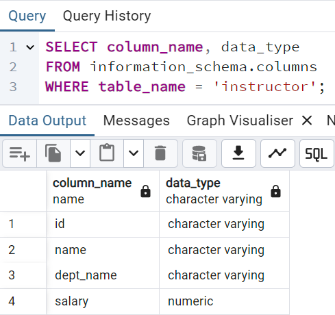
**2022141460051 王晨盛**

-- 我知道可以不用附结果图 ，但为了方便

**I. Write the following queries in SQL, using the university schema. (Execute script files create\_schema.sql and insert\_data.sql.)**

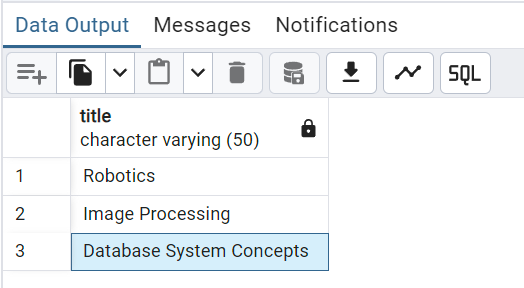
********

1. **Find the titles of courses in the Comp.Sci. department that have 3 credits.**

SELECT title

FROM course

WHERE dept\_name = 'Comp. Sci.' AND credits = 3;



1. **Find the IDs of all students who were taught by an instructor whose name is Einstein; make sure there are no duplicates in the result.**

SELECT DISTINCT \* --stu.id

FROM student stu

JOIN takes ta ON stu.id = ta.id

JOIN section sec ON ta.course\_id = sec.course\_id AND ta.sec\_id = sec.sec\_id AND ta.semester = sec.semester AND ta.year = sec.year

JOIN teaches tea ON sec.course\_id = tea. course\_id AND sec.sec\_id = tea.sec\_id AND sec.semester = tea.semester AND sec.year = tea.year

JOIN instructor ins ON tea.id = ins.id

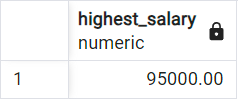
WHERE ins.name = 'Einstein';

"44553" "Peltier" "Physics" 56 "44553" "PHY-101" "1" "Fall" 2017 "B-" "PHY-101" "1" "Fall" 2017 "Watson" "100" "A" "22222" "PHY-101" "1" "Fall" 2017 "22222" "Einstein" "Physics" 95000.00

1. **Find the highest salary of any instructor.**

SELECT MAX(salary) AS highest\_salary

FROM instructor;



1. **Find all instructors earning the highest salary (there may be more than one with the same salary).**

SELECT \*

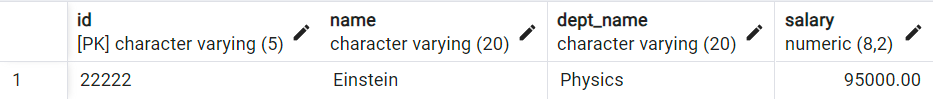
FROM instructor

WHERE salary = (

SELECT MAX(salary)

FROM instructor

);

****

1. **Find the enrollment（选课人数） of each section that was offered in Fall 2017.**

SELECT sec.course\_id, sec.sec\_id, sec.semester, sec.year, COUNT(\*) AS enrollment

FROM section sec

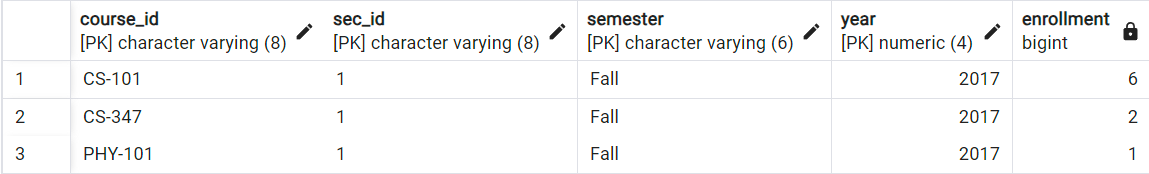
JOIN takes ta ON sec.course\_id = ta.course\_id AND sec.sec\_id = ta.sec\_id AND sec.semester = ta.semester AND sec.year = ta.year

WHERE sec.semester = 'Fall'

AND sec.year = 2017

GROUP BY sec.course\_id, sec.sec\_id, sec.semester, sec.year

ORDER BY sec.course\_id, sec.sec\_id;



1. **Find the maximum enrollment, across all sections, in Fall 2017.**

SELECT MAX(enrollment) AS max\_enrollment

FROM (

SELECT COUNT(\*) AS enrollment

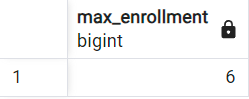
FROM takes ta

JOIN section ON ta.course\_id = section.course\_id AND ta.sec\_id = section.sec\_id AND ta.semester = section.semester AND ta.year = section.year

WHERE section.semester = 'Fall' AND section.year = 2017

GROUP BY section.course\_id, section.sec\_id, section.semester, section.year

) AS subquery;



**7. Find the sections that had the maximum enrollment in Fall 2017.**

SELECT sec.course\_id, sec.sec\_id, sec.semester, sec.year

FROM section sec

JOIN takes ta ON sec.course\_id = ta.course\_id

AND sec.sec\_id = ta.sec\_id

AND sec.semester = ta.semester

AND sec.year = ta.year

WHERE sec.semester = 'Fall'

AND sec.year = 2017

GROUP BY sec.course\_id, sec.sec\_id, sec.semester, sec.year

HAVING COUNT(ta.id) = (

SELECT MAX(enrollment)

FROM (

SELECT COUNT(\*) AS enrollment

FROM section sec

JOIN takes ta ON sec.course\_id = ta.course\_id

AND sec.sec\_id = ta.sec\_id

AND sec.semester = ta.semester

AND sec.year = ta.year

WHERE sec.semester = 'Fall' AND sec.year = 2017

GROUP BY sec.course\_id, sec.sec\_id

) AS enrollments

)

ORDER BY sec.course\_id, sec.sec\_id;



**II.You will be working with the following schemas （Execute script files e2c.txt and e2i.txt.）:**

**Employee (SSN, name, salary, DNo)**

**Department (DNo, dept\_name, mgr\_SSN)**

**Project (PNo, location, proj\_name, DNo)**

**HourLog (SSN, PNo, hours)**

**The Employee relation provides a list of employees with their SSN, name, salary, and department number (DNo).**

**The SSN is unique for each employee. Each employee belongs to only one department. The Department relation**

**contains a list of the departments for the company. Its schema includes a unique department number called DNo.**

**It also includes the name of the department (dept\_name) and the social security number of the department's**

**manager (mgr\_SSN). Each department has a only one manager. The Project relation includes a unique project**

**number (PNo), location, the project name (proj\_name) and the department(DNo) to be responsible for it.**

**An employee can be assigned to any number (including zero) projects. Each project has at least one person**

**assigned to it. Finally, the HourLog relation lists for each project the number of hours of work for each**

**employee who is assigned to that project. The key of this relation is SSN and PNo.**

**Write SQL statements to perform the following commands.**

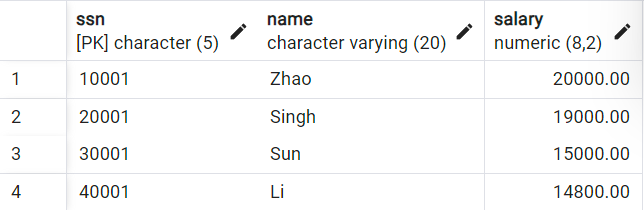
1. **List all managers’ SSN, name and salary.**

SELECT DISTINCT e.ssn, e.name, e.salary

FROM employee e

JOIN department d ON e.ssn = d.mgr\_ssn

ORDER BY e.ssn;



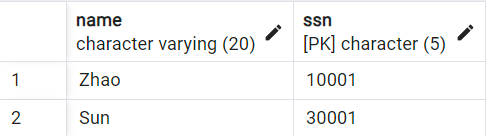
1. **Find the name and the SSN of everyone who works more than 100 hours on any project; make sure there are no duplicates in the result.**

SELECT DISTINCT e.name, e.ssn

FROM employee e

JOIN hourlog h ON e.ssn = h.ssn

WHERE h.hours > 100;



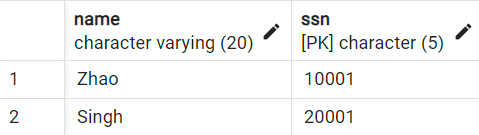
1. **Find the name and the SSN of everyone who works on at least two projects.**

SELECT e.name, e.ssn

FROM employee e

JOIN hourlog h ON e.ssn = h.ssn

GROUP BY e.ssn, e.name

****

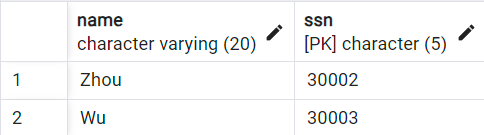
1. **HAVING COUNT(DISTINCT h.pno) >= 2;4. Find the name and the SSN of everyone who have never worked on any projects.**

SELECT e.name, e.ssn

FROM employee e

LEFT JOIN hourlog h ON e.ssn = h.ssn

WHERE h.ssn IS NULL;



1. **Find the name and the SSN of everyone who works on all projects that "Singh" (the name of an employee) works on.**

SELECT e.name, e.ssn

FROM employee e

JOIN hourlog h ON e.ssn = h.ssn

WHERE h.pno IN (

SELECT pno

FROM hourlog

WHERE ssn = (

SELECT ssn

FROM employee

WHERE name = 'Singh'

)

)

GROUP BY e.ssn, e.name

HAVING COUNT(DISTINCT h.pno) = (

SELECT COUNT(\*)

FROM hourlog

WHERE ssn = (

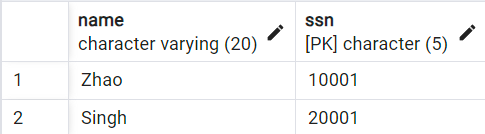
SELECT ssn

FROM employee

WHERE name = 'Singh'

)

);



1. **Find the name and the SSN of everyone who works on all projects that his deparment is responsible for them.**

SELECT e.name, e.ssn

FROM employee e

JOIN hourlog h ON e.ssn = h.ssn

JOIN project p ON h.pno = p.pno

WHERE e.dno = p.dno

GROUP BY e.ssn, e.name

HAVING COUNT(DISTINCT h.pno) = (

SELECT COUNT(\*)

FROM project pr

WHERE pr.dno = e.dno

);

