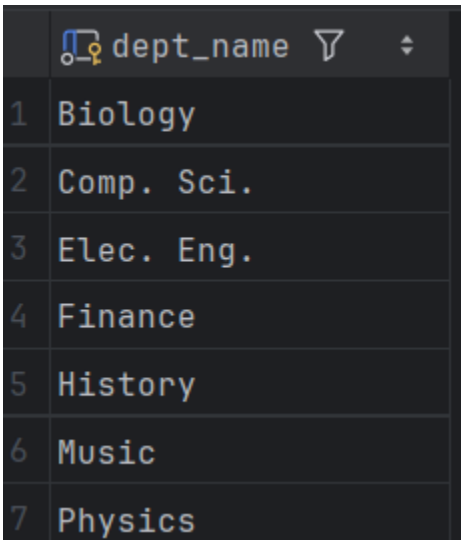
#1. List all departments:

SELECT dept_name FROM department;



	dept_name
1	Biology
2	Comp. Sci.
3	Elec. Eng.
4	Finance
5	History
6	Music
7	Physics

#2. Find the names of all instructors:

SELECT name FROM instructor;

1	Srinivasan
2	Wu
3	Mozart
4	Einstein
5	El Said
6	Gold
7	Katz
8	Califieri
9	Singh
10	Crick
11	Brandt
12	Kim

#3. Retrieve all courses offered by the Biology department:

```
SELECT title
FROM course
WHERE dept_name = 'Biology';
```

	title
1	Intro. to Biology
2	Genetics
3	Computational Biology

#4. Show the capacity of the 'Watson' building:

```
SELECT capacity
FROM classroom
WHERE building = 'Watson';
```

	capacity
1	30
2	50

#5. List all students in the 'Comp. Sci.' department:

```
SELECT name
FROM student
WHERE dept_name = 'Comp. Sci.';
```

	name
1	Zhang
2	Shankar
3	Williams
4	Brown

#6. Find all courses with credits greater than 3:

```
SELECT title
FROM course
WHERE credits > 3;
```

	title
1	Intro. to Biology
2	Genetics
3	Intro. to Computer Science
4	Game Design
5	Physical Principles

#7. Retrieve the names and salaries of instructors earning more than 70,000:

```
SELECT name, salary
FROM instructor
WHERE salary > 70000;
```

	name	salary
1	Wu	90000.00
2	Einstein	95000.00
3	Gold	87000.00
4	Katz	75000.00
5	Singh	80000.00
6	Crick	72000.00
7	Brandt	92000.00
8	Kim	80000.00

#8. Display all sections offered in the 'Fall' semester of 2017:

```
SELECT *
FROM section
WHERE semester = 'Fall' AND year = 2017;
```

	course_id	sec_id	semester	year	building	room_number	time_slot_id
1	CS-101	1	Fall	2017	Packard	101	H
2	CS-347	1	Fall	2017	Taylor	3128	A
3	PHY-101	1	Fall	2017	Watson	100	A

#9. Find the names of students who have earned more than 50 total credits:

```
SELECT name
FROM student
WHERE tot_cred > 50;
```

	name
1	Zhang
2	Brandt
3	Chavez
4	Peltier
5	Williams
6	Brown
7	Aoi
8	Bourikas
9	Tanaka

#10. List all courses along with their prerequisites:SELECT c.title AS course, p.prereq_id
FROM course c
JOIN prereq p ON c.course_id = p.course_id;

	course_id
1	BIO-301
2	BIO-399
3	CS-190
4	CS-315
5	CS-319
6	CS-347
7	EE-181

#11. Display the room numbers in the 'Taylor' building:

```
SELECT room_number
FROM classroom
WHERE building = 'Taylor';
```

	room_number
1	3128

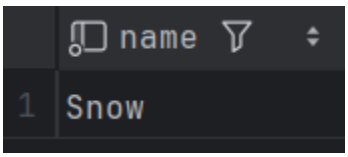
#12. Retrieve the name and department of all instructors:

```
SELECT name, dept_name
FROM instructor;
```

	name	dept_name
1	Srinivasan	Comp. Sci.
2	Wu	Finance
3	Mozart	Music
4	Einstein	Physics
5	El Said	History
6	Gold	Physics
7	Katz	Comp. Sci.
8	Califieri	History
9	Singh	Finance
10	Crick	Biology
11	Brandt	Comp. Sci.
12	Kim	Elec. Eng.

#13. Find all students with a total credit of 0:

```
SELECT name
FROM student
WHERE tot_cred = 0;
```



	name
1	Snow

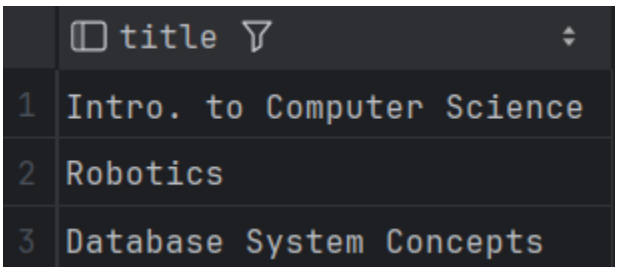
#14. Show all sections that take place in the 'Packard' building:

```
SELECT *
FROM section
WHERE building = 'Packard';
```

	course_id	sec_id	semester	year	building	room_number	time_slot_id
1	CS-101	1	Fall	2017	Packard	101	H
2	CS-101	1	Spring	2018	Packard	101	F
3	FIN-201	1	Spring	2018	Packard	101	B
4	MU-199	1	Spring	2018	Packard	101	D

#15. List all courses taught by 'Srinivasan':

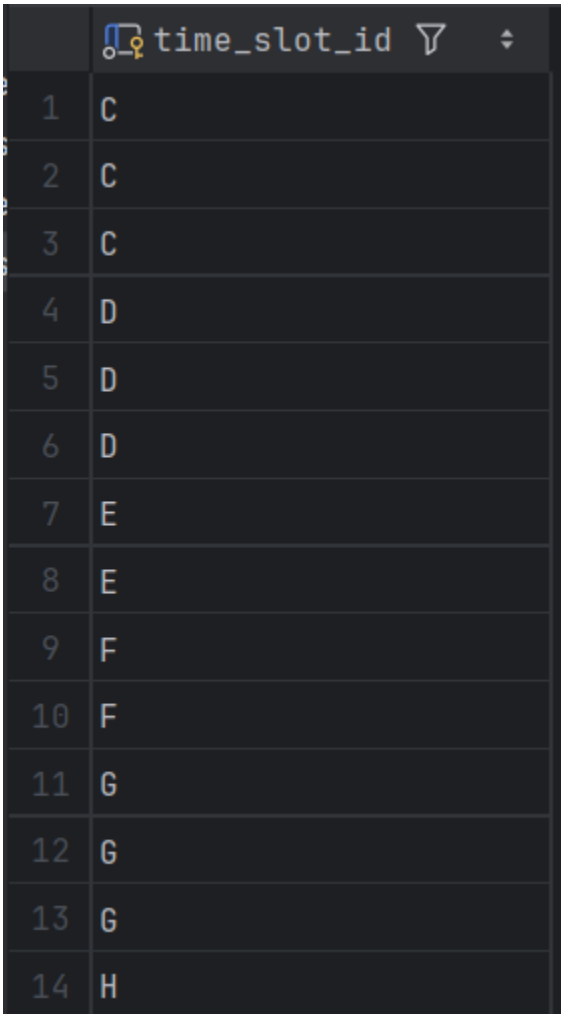
```
SELECT c.title
FROM course c
JOIN teaches t ON c.course_id = t.course_id
JOIN instructor i ON t.ID = i.ID
WHERE i.name = 'Srinivasan';
```



	title
1	Intro. to Computer Science
2	Robotics
3	Database System Concepts

#16. Retrieve all time slots that start at or after 10 AM.

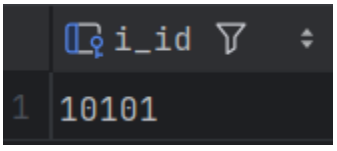
```
select time_slot_id from time_slot
where start_hr >= 10;
```



	time_slot_id
1	C
2	C
3	C
4	D
5	D
6	D
7	E
8	E
9	F
10	F
11	G
12	G
13	G
14	H

#17. Find all advisors for student '12345'.

```
select advisor.i_id from advisor
join student on advisor.s_ID = student.ID
where student.ID = 12345;
```



	i_id
1	10101

#18. List all departments located in the 'Taylor' building.

```
SELECT DISTINCT d.dept_name
FROM department d
JOIN classroom c ON d.building = c.building
WHERE c.building = 'Taylor';
```

	dept_name
1	Comp. Sci.
2	Elec. Eng.

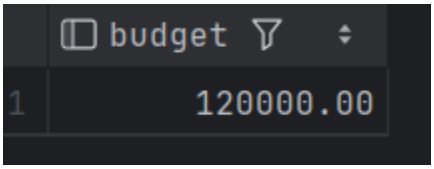
#19. Show all students along with their assigned advisors:

```
SELECT s.name AS student_name, a.i_ID AS advisor_id
FROM student s
JOIN advisor a ON s.ID = a.s_ID;
```

	student_name	advisor_id
1	Shankar	10101
2	Peltier	22222
3	Levy	22222
4	Zhang	45565
5	Brown	45565
6	Chavez	76543
7	Tanaka	76766
8	Aoi	98345
9	Bourikas	98345

#20. Retrieve the budget of the 'Finance' department:

```
SELECT budget
FROM department
WHERE dept_name = 'Finance';
```



	budget
1	120000.00

#21. Find the names of students who took 'CS-101' in 'Fall 2017':

```
SELECT s.name
FROM student s
JOIN takes t ON s.ID = t.ID
WHERE t.course_id = 'CS-101' AND t.semester = 'Fall' AND t.year = 2017;
```

	name
1	Zhang
2	Shankar
3	Levy
4	Williams
5	Brown
6	Bourikas

#22. List the titles of courses taught by 'Crick':

```
SELECT c.title
FROM course c
JOIN teaches t ON c.course_id = t.course_id
JOIN instructor i ON t.ID = i.ID
WHERE i.name = 'Crick';
```

	title
1	Intro. to Biology
2	Genetics

#23. Retrieve the total budget for all departments in the 'Taylor' building:

```
SELECT SUM(d.budget) AS total_budget
FROM department d
JOIN classroom c ON d.building = c.building
WHERE c.building = 'Taylor';
```

	total_budget
1	185000.00

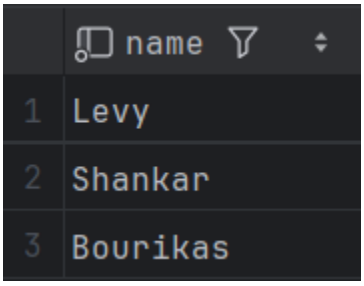
#24. Find the average salary of instructors in each department:

```
SELECT dept_name, AVG(salary) AS avg_salary
FROM instructor
GROUP BY dept_name;
```

	dept_name	avg_salary
1	Biology	72000.000000
2	Comp. Sci.	77333.333333
3	Elec. Eng.	80000.000000
4	Finance	85000.000000
5	History	61000.000000
6	Music	40000.000000
7	Physics	91000.000000

#25. Show all students who are taking 'Spring 2018' courses in the 'Watson' building:

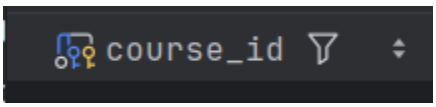
```
SELECT DISTINCT s.name
FROM student s
JOIN takes t ON s.ID = t.ID
JOIN section sec ON t.course_id = sec.course_id AND t.sec_id = sec.sec_id
WHERE sec.building = 'Watson' AND sec.semester = 'Spring' AND sec.year = 2018;
```



	name
1	Levy
2	Shankar
3	Bourikas

#26. List all courses that have more than one prerequisite:

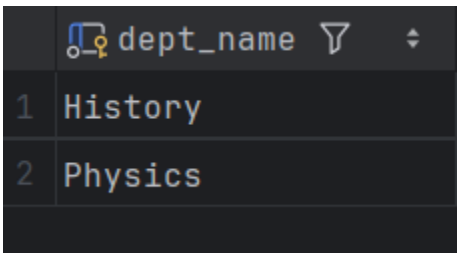
```
SELECT course_id
FROM prereq
GROUP BY course_id
HAVING COUNT(prereq_id) > 1;
```



course_id

#27. Find the departments with a budget less than 75,000:

```
SELECT dept_name
FROM department
WHERE budget < 75000;
```



	dept_name
1	History
2	Physics

#28. Retrieve all time slots that span more than an hour:

```
SELECT *
FROM time_slot
WHERE (end_hr * 60 + end_min) - (start_hr * 60 + start_min) > 60;
```

	time_slot_id	day	start_hr	start_min	end_hr	end_min
1	E	R	10	30	11	45
2	E	T	10	30	11	45
3	F	R	14	30	15	45
4	F	T	14	30	15	45
5	H	W	10	0	12	30

#29. Display the names of instructors who have taught in the 'Taylor' building:

```
SELECT DISTINCT i.name
FROM instructor i
JOIN teaches t ON i.ID = t.ID
JOIN section sec ON t.course_id = sec.course_id AND t.sec_id =
sec.sec_id
WHERE sec.building = 'Taylor';
```

	name
1	Brandt
2	Srinivasan
3	Kim

#30. Show the highest total credits among all students:
SELECT *MAX*(tot_cred) AS highest_credits
FROM student;

	highest_credits
1	120

#31. Find the instructor teaching the most courses:

SELECT i.name, COUNT(t.course_id) AS course_count

FROM instructor i

JOIN teaches t ON i.ID = t.ID

GROUP BY i.ID

ORDER BY course_count DESC

LIMIT 1;

	name	course_count
1	Srinivasan	3

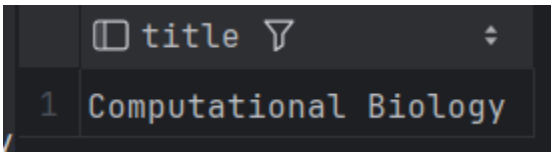
#32 List the courses that no students are enrolled in:

SELECT c.title

FROM course c

LEFT JOIN takes t ON c.course_id = t.course_id

WHERE t.ID IS NULL;



	title
1	Computational Biology

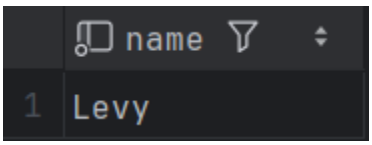
#33 Retrieve the names of students who failed at least one course:

SELECT DISTINCT s.name

FROM student s

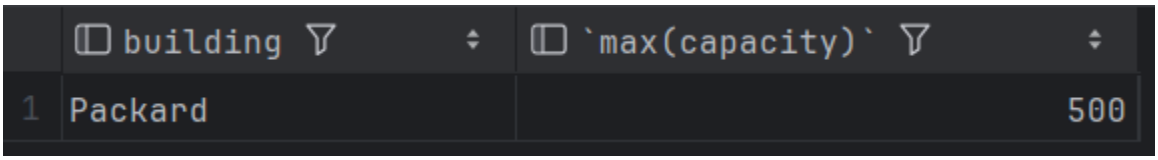
JOIN takes t ON s.ID = t.ID

WHERE t.grade = 'F';



	name
1	Levy

34. Find the building with the largest total classroom capacity.SELECT s.building, *max*(capacity)
FROM section AS s
JOIN classroom AS c
ON s.room_number = c.room_number
GROUP BY building
LIMIT 1;



	building	`max(capacity)`
1	Packard	500

35. List the names of students advised by 'Einstein'.

```
SELECT s.name
FROM student AS s
JOIN advisor AS a
ON s.ID = a.s_ID
JOIN instructor AS i
ON a.i_ID = i.ID
WHERE i.name = 'Einstein';
```

	name
1	Peltier
2	Levy

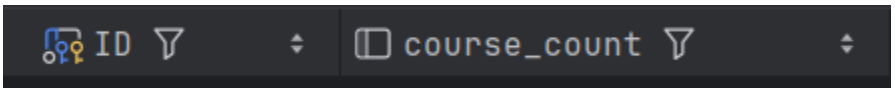
36. Display the titles of courses taken by 'Zhang'.

```
SELECT DISTINCT c.title
FROM takes T
JOIN student S
ON t.ID = s.ID
JOIN course C
ON c.course_id = t.course_id
where s.name = 'Zhang';
```

	title
1	Intro. to Computer Science
2	Database System Concepts

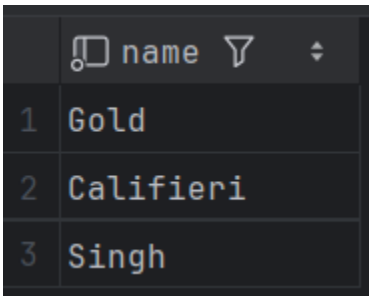
37. Show the names of students who have completed all courses in the 'Comp. Sci.' department.

```
WITH total_cs_courses AS (
    SELECT COUNT(*) AS total_count
    FROM course
    WHERE dept_name = 'Comp. Sci.'
)
SELECT T.ID, COUNT(DISTINCT T.course_id) AS course_count
FROM takes T
JOIN course C
  ON T.course_id = C.course_id
WHERE C.dept_name = 'Comp. Sci.'
  AND T.grade != 'F'
  AND T.grade IS NOT NULL
GROUP BY T.ID
HAVING COUNT(DISTINCT T.course_id) = (SELECT total_count FROM
total_cs_courses);
```



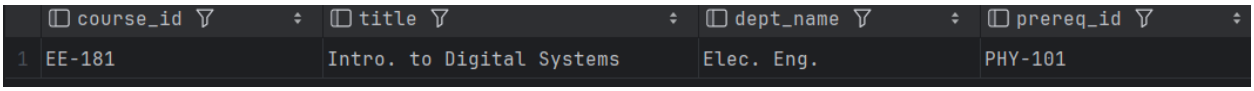
38. Retrieve all instructors who have never taught a course.

```
SELECT i.name

FROM instructor i

LEFT JOIN teaches t ON i.ID = t.ID

WHERE t.course_id IS NULL;
```



39. Find all courses that have prerequisites from a different department.

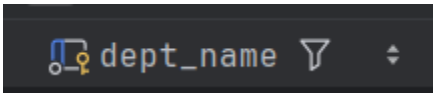
```
SELECT c.course_id, c.title, c.dept_name, p.prereq_id
FROM course c
JOIN prereq p ON c.course_id = p.course_id
JOIN course prereq_course ON p.prereq_id = prereq_course.course_id
WHERE c.dept_name != prereq_course.dept_name;
```



	course_id	title	dept_name	prereq_id
1	EE-181	Intro. to Digital Systems	Elec. Eng.	PHY-101

40. List all departments where no students are enrolled.

```
SELECT d.dept_name
FROM department d
LEFT JOIN student s ON d.dept_name = s.dept_name
WHERE s.ID IS NULL;
```

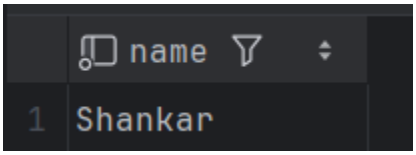


dept_name

41. Find students who are taking all courses taught by 'Srinivasan'.

```
 SELECT s.name
FROM student s
WHERE NOT EXISTS (
   SELECT t.course_id
   FROM teaches t
   JOIN instructor i ON t.ID = i.ID
   WHERE i.name = 'Srinivasan'
   EXCEPT
   SELECT tk.course_id
   FROM takes tk
```

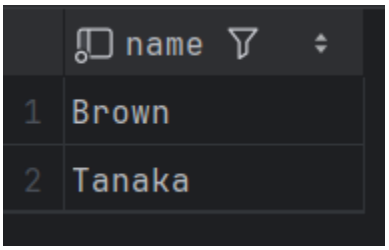
```
    WHERE tk.ID = s.ID
);
```



	name
1	Shankar

42. Retrieve the name of the student with the highest GPA (assume grades map to points: A=4, B=3, etc.).

```
    WITH GPA_Calculation AS ( SELECT s.ID, s.name, SUM(CASE grade
WHEN 'A' THEN 4 WHEN 'B' THEN 3 WHEN 'C' THEN 2 WHEN 'D' THEN 1
WHEN 'F' THEN 0 ELSE 0 END) * 1.0 / COUNT(*) AS GPA FROM student s
JOIN takes t ON s.ID = t.ID WHERE grade IS NOT NULL GROUP BY s.ID,
s.name ) SELECT name FROM GPA_Calculation WHERE GPA = (SELECT
MAX(GPA) FROM GPA_Calculation);
```



	name
1	Brown
2	Tanaka

43. List the courses with overlapping time slots.

```
SELECT c1.course_id AS Course1, c2.course_id AS Course2 FROM
section s1 JOIN section s2 ON s1.time_slot_id = s2.time_slot_id AND
s1.course_id <> s2.course_id JOIN course c1 ON s1.course_id =
c1.course_id JOIN course c2 ON s2.course_id = c2.course_id;
```

Course1	Course2
FIN-201	BIO-101
CS-319	BIO-101
PHY-101	BIO-301
CS-347	BIO-301
CS-190	BIO-301
PHY-101	CS-190
CS-347	CS-190
BIO-301	CS-190
MU-199	CS-315
FIN-201	CS-319
BIO-101	CS-319
HIS-351	CS-319
EE-181	CS-319
PHY-101	CS-347
CS-190	CS-347
BIO-301	CS-347
HIS-351	EE-181
CS-319	EE-181
CS-319	FIN-201
BIO-101	FIN-201
EE-181	HIS-351
CS-319	HIS-351