

Title	Assignment 03: Information Management (GISC 6354)
Handed Out	Thursday, February 08, 2024
Name	Victoria Ebeh
Email	Vae230000@utdallas.edu

Q1: Open the Online SQL interpreter (<https://www.db-book.com/db7/university-labdir/sqljs.html>)

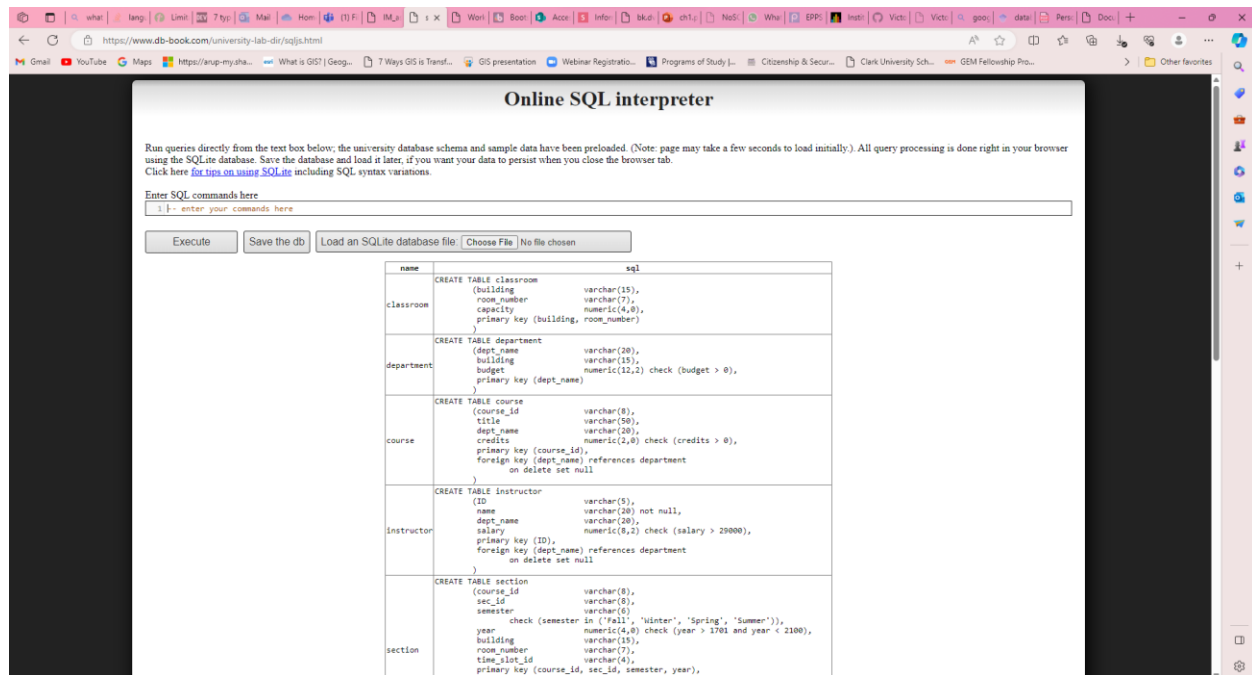


Figure 1: Image of opened online SQL interpreter.

Q2: Write SQL codes to get a list of:

- i. **Students IDs (hint: from the takes relation)**
`SELECT DISTINCT ID FROM student`
- ii. **Instructors**
`SELECT * FROM instructor`
OR
`SELECT ID, name, dept_name FROM instructor`
- iii. **Department**
`SELECT dept_name FROM department`

Q3: Write in SQL codes to do following queries:

- i. Find the ID and name of each student who has taken at least one Comp. Sci. course; make sure there are no duplicate names in the result.

SELECT DISTINCT student.ID, name **FROM** student, course, takes **WHERE** course.course_id = takes.course_id **AND** course.dept_name = 'Comp. Sci.';

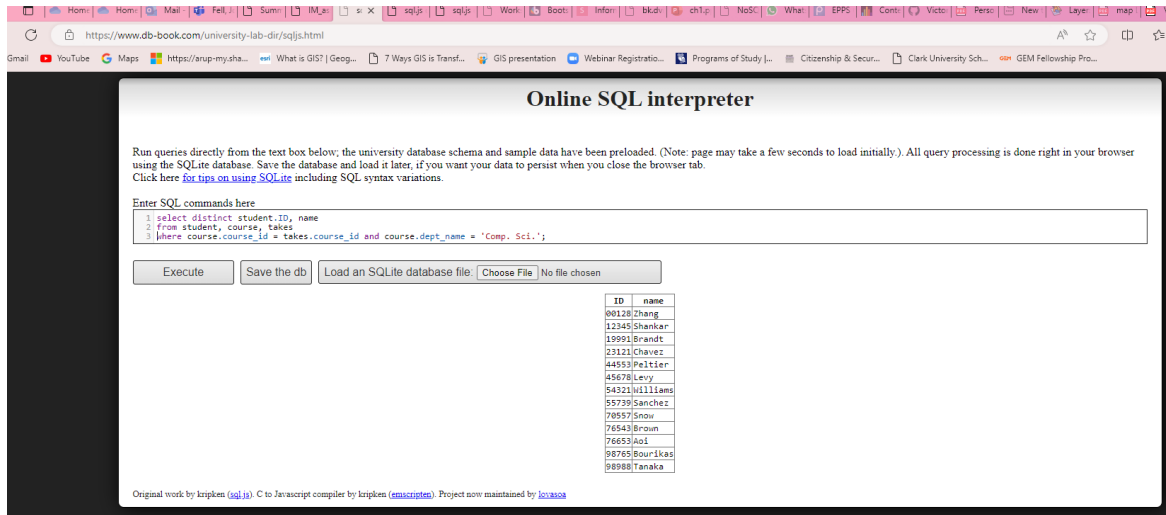


Figure 2: query results for students that have taken at least one Comp.Sci course

- ii. Add grades to the list

SELECT DISTINCT student.ID, name, grades **FROM** student, course, takes **WHERE** course.course_id = takes.course_id **AND** course.dept_name = 'Comp. Sci.';

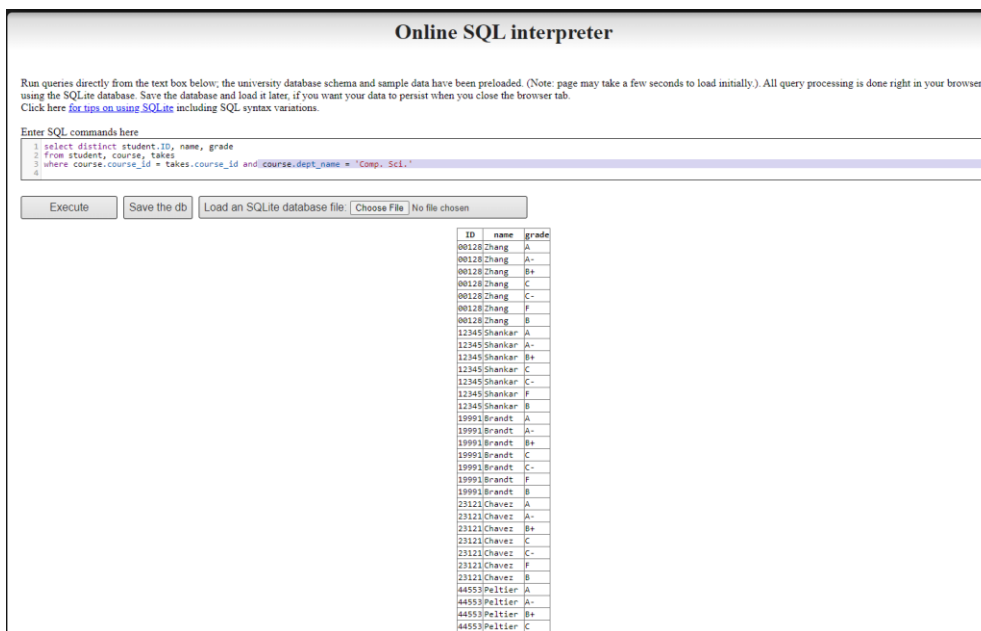


Figure 3: query results for students that have taken at least one Comp.Sci course with grades

- iv. Find the ID and name of each student who has not taken any course offered before 2017.

```
SELECT DISTINCT s.ID, s.name
FROM student s, takes
WHERE NOT EXISTS (
  SELECT 1
  FROM takes t
  JOIN section sec ON t.course_id = sec.course_id AND t.sec_id = sec.sec_id
  WHERE t.ID = s.ID AND sec.year < 2017
);
```

Online SQL interpreter

Run queries directly from the text box below; the university database schema and sample data have been preloaded. (Note: page may take a few seconds to load initially.). All query processing is done right in your browser using the SQLite database. Save the database and load it later, if you want your data to persist when you close the browser tab. Click here [for tips on using SQLite](#) including SQL syntax variations.

Enter SQL commands here

```
1 SELECT DISTINCT s.ID, s.name
2 FROM student s, takes
3 WHERE NOT EXISTS (
4   SELECT 1
5   FROM takes t
6   JOIN section sec ON t.course_id = sec.course_id AND t.sec_id = sec.sec_id
7   WHERE t.ID = s.ID AND sec.year < 2017
8 );
9
10
```

Execute Save the db Load an SQLite database file: Choose File No file chosen

ID	name
00128	Zhang
12345	Shankar
19991	Brandt
23121	Chavez
44551	Peltier
45678	Levy
54321	Williams
55739	Sanchez
70557	Snow
70543	Brown
76653	Adi
98765	Bourikas
98988	Tanaka

Original work by kripken ([sql.js](#)). C to Javascript compiler by kripken ([smucripten](#)). Project now maintained by [louisroa](#)

Figure 4: Query results for students who havent taken a course before 2017

- v. For each department, find the maximum salary of instructors in that department. You may assume that every department has at least one instructor.

```
SELECT dept_name, MAX(salary) AS max_salary
FROM instructor
GROUP BY dept_name
```

Online SQL interpreter

Run queries directly from the text box below; the university database schema and sample data have been preloaded. (Note: page may take a few seconds to load initially.). All query processing is done right in your browser using the SQLite database. Save the database and load it later, if you want your data to persist when you close the browser tab. Click here [for tips on using SQLite](#) including SQL syntax variations.

Enter SQL commands here

```
1 SELECT dept_name, MAX(salary) AS max_salary
2 FROM instructor
3 GROUP BY dept_name
4
```

Execute Save the db Load an SQLite database file: Choose File No file chosen

dept_name	max_salary
Biology	72000
Comp. Sci.	92000
Elec. Eng.	80000
Finance	90000
History	62000
Music	40000
Physics	95000

Original work by kripken ([sql.js](#)). C to Javascript compiler by kripken ([smucripten](#)). Project now maintained by [louisroa](#)

Figure 5: Maximum salary of instructors across departments.

v. Find the lowest, across all departments, of the per-department maximum salary computed by the preceding query.

```
SELECT MIN(max_salary) AS lowest_max_salary
FROM (
  SELECT MAX(salary) AS max_salary
  FROM instructor
  GROUP BY dept_name
) AS department_max_salaries;
```

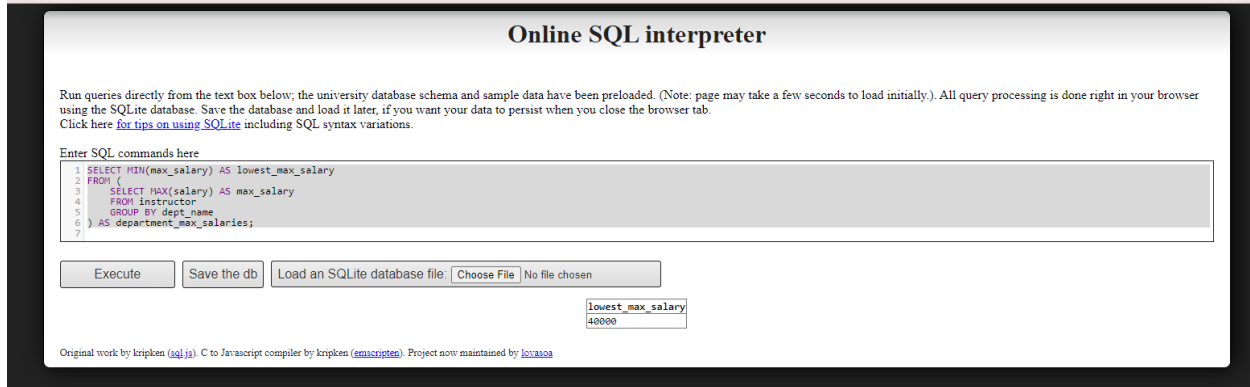


Figure 6: lowest salary from maximum salary query results.

vi. Add names to the list

```
SELECT name, MIN(max_salary) AS lowest_max_salary
FROM (
  SELECT name, MAX(salary) AS max_salary
  FROM instructor
  GROUP BY dept_name
) AS department_max_salaries;
```

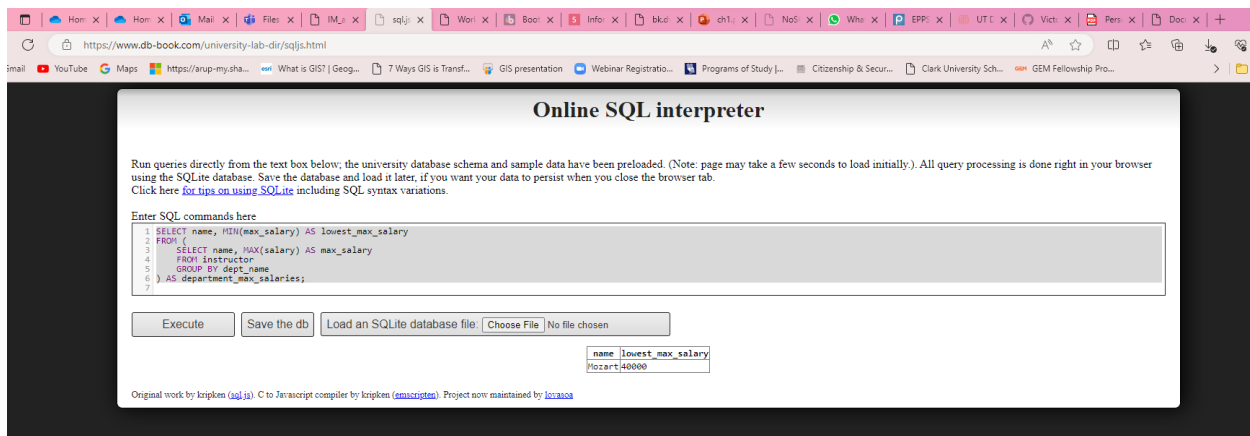


Figure 7: lowest salary from maximum salary query results and names

Q4: Find an instructor (with name and ID) who has never given an A grade in any course she or he has taught. (Instructors who have never taught a course trivially satisfy this condition.)

```
SELECT DISTINCT i.ID, i.name
FROM instructor i
LEFT JOIN teaches t ON i.ID = t.ID
LEFT 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 t.ID IS NULL OR tk.grade <> 'A';
```

Online SQL interpreter

Run queries directly from the text box below; the university database schema and sample data have been preloaded. (Note: page may take a few seconds to load initially.) All query processing is done right in your browser using the SQLite database. Save the database and load it later, if you want your data to persist when you close the browser tab. Click here [for tips on using SQLite](#) including SQL syntax variations.

Enter SQL commands here

```
1 SELECT DISTINCT i.ID, i.name
2 FROM instructor i
3 LEFT JOIN teaches t ON i.ID = t.ID
4 LEFT 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
5 WHERE t.ID IS NULL OR tk.grade <> 'A';
6
7
```

Execute Save the db Load an SQLite database file: Choose File No file chosen

ID	name
10101	Srinivasan
12121	Hu
15151	Mozart
22222	Einstein
32343	El Said
33456	Gold
45665	Katz
56563	Califlieri
76543	Singh
83821	Brandt
98345	Kim

Original work by kripken ([sql.js](#)). C to Javascript compiler by kripken ([emsccripten](#)). Project now maintained by [luccasoz](#)

Figure 8: Instructors that have not rewarded A grade