# Database System Concepts

**Shikha Chamoli**

**Schema**

## Diagram Description automatically generated with medium confidence

**SQL Queries**

Write queries in SQL, on the University schema, to answer each of the following questions:

1. Find the names of all the instructors from Biology department

|  |
| --- |
| select \* from instructor where dept\_name="Biology"; |

Graphical user interface, application

Description automatically generated

1. Find the names of courses in Computer science department which have 3 credits

|  |
| --- |
| select \* from course where dept\_name= "Comp. Sci." and credits=3; |

Graphical user interface, application, Word

Description automatically generated

1. For the student with ID 12345 (or any other value), show all course\_id and title of all courses registered for by the student.

|  |
| --- |
| select ID, T.course\_id, title from (select \* from takes where ID="12345" ) as T left join course on T.course\_id = course.course\_id |

Graphical user interface, text

Description automatically generated

1. As above, but show the total number of credits for such courses (taken by that student). Don't display the tot\_creds value from the student table, you should use SQL aggregation on courses taken by the student.

|  |
| --- |
| select ID, sum(credits) from ((select \* from takes where ID="12345" ) as T left join course on T.course\_id = course.course\_id) group by ID |

Graphical user interface, text, application

Description automatically generated

1. As above, but display the total credits for each of the students, along with the ID of the student; don't bother about the name of the student. (Don't bother about students who have not registered for any course, they can be omitted)

|  |
| --- |
| select \* from  (select ID, sum(credits) as TotalCredits from  (takes left join course on takes.course\_id = course.course\_id)  group by ID) as T  where TotalCredits>0 |

Graphical user interface, application

Description automatically generated

1. Find the names of all students who have taken any Comp. Sci. course ever (there should be no duplicate names)

|  |
| --- |
| select distinct name from  (select ID, takes.course\_id, dept\_name from  (takes left join course on takes.course\_id = course.course\_id)) as T  left join student on T.ID = student.ID |

Graphical user interface, text, application

Description automatically generated

1. Display the IDs of all instructors who have never taught a couse (Notesad1) Oracle uses the keyword minus in place of except; (2) interpret "taught" as "taught or is scheduled to teach")

|  |
| --- |
| select ID from instructor where ID not in (select ID from teaches) |

Graphical user interface, text, application

Description automatically generated

1. As above, but display the names of the instructors also, not just the IDs.

|  |
| --- |
| select ID, name from instructor where ID not in (select ID from teaches) |

Graphical user interface, text, application, email

Description automatically generated

Credits:

These are the queries from the book –

**Database System Concepts *Seventh Edition***

[**Avi Silberschatz**](http://www.cs.yale.edu/homes/avi), [**Henry F. Korth**](http://www.cse.lehigh.edu/~korth)**,** [**S. Sudarshan**](http://www.cse.iitb.ac.in/~sudarsha)

<https://www.db-book.com/university-lab-dir/exercises-dir/>