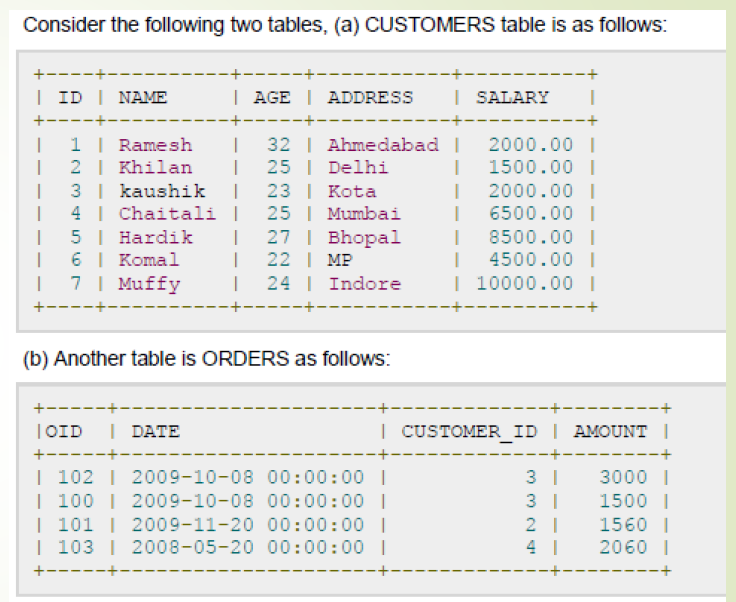
Joins Practice:



SHOW CUSTOMER NAMES , AMOUNT, SALARY if they have placed order

INNER JOIN:

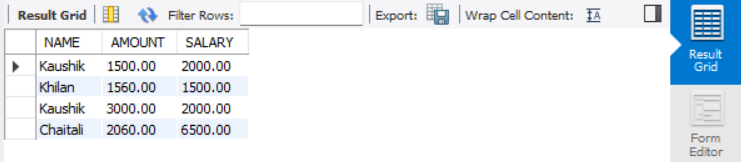
A type of join that combines rows from two or more tables based on a related column between them and returns the rows where there is a match.

SELECT NAME , ORDERS.AMOUNT , SALARY

FROM CUSTOMERS

INNER JOIN ORDERS

ON CUSTOMERS.ID = ORDERS.CUSTOMER\_ID ;



LEFT JOIN:

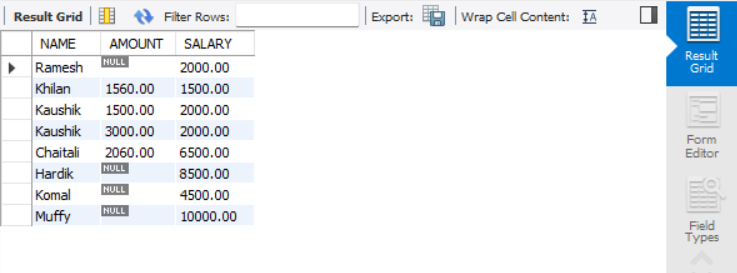
A LEFT JOIN returns all rows from the left table (the first table mentioned in the query), along with matching rows from the right table

SELECT NAME , ORDERS.AMOUNT , SALARY

FROM CUSTOMERS

LEFT JOIN ORDERS

ON CUSTOMERS.ID = ORDERS.CUSTOMER\_ID ;



RIGHT JOIN:

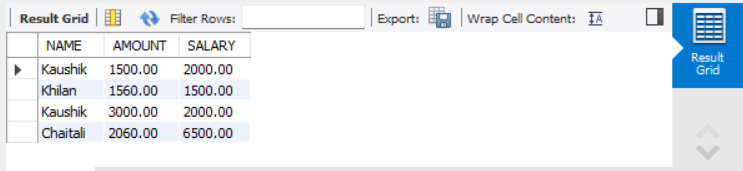
A RIGHT JOIN returns all rows from the RIGHT table (the first table mentioned in the query), along with matching rows from the LEFT table

SELECT NAME , ORDERS.AMOUNT , SALARY

FROM CUSTOMERS

RIGHT JOIN ORDERS

ON CUSTOMERS.ID = ORDERS.CUSTOMER\_ID ;



NATURAL JOIN:

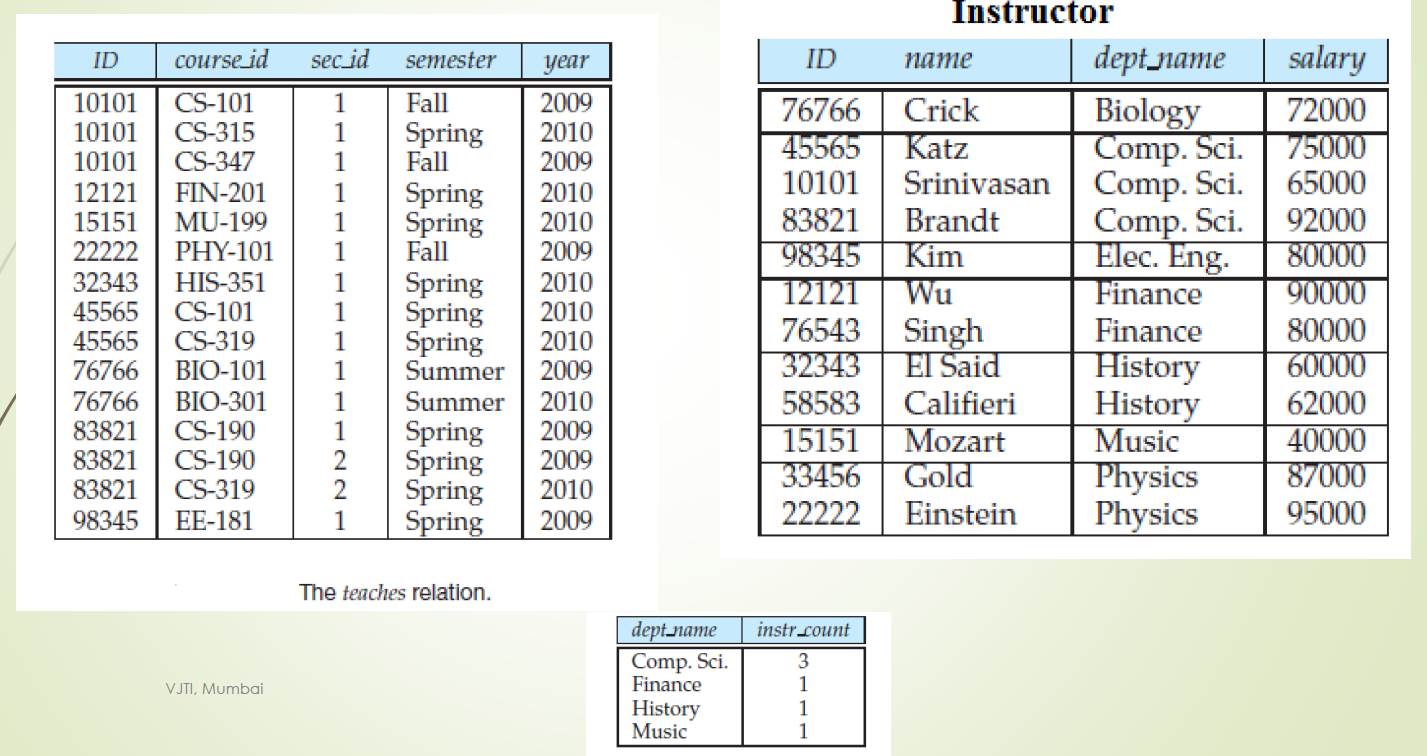
No need of providing the ON condition, this automatically joins the table based to the same column name. only will work correctly when the column names are same, INNER JOIN

SELECT \* FROM ORDERS1 ;

SELECT NAME , ORDERS.AMOUNT, ORDERS.ID , SALARY

FROM CUSTOMERS

NATURAL JOIN ORDERS;



Find the number of instructors in each department who teach a course in theSpring 2010 semester

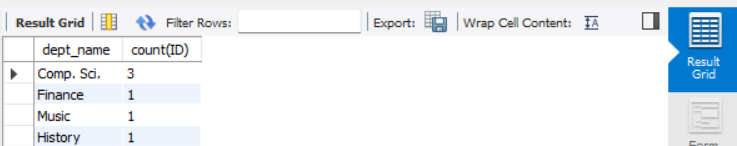
SELECT dept\_name ,count(ID) from Instructor

NATURAL JOIN teaches

WHERE teaches.year = 2010

AND teaches.semester = 'Spring'

GROUP BY dept\_name;



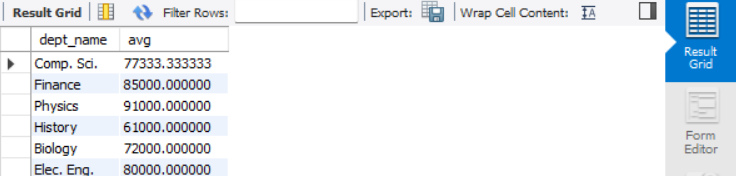
Department name where the average salary of the instructors is more than $42,000.

SELECT dept\_name, avg(salary) as avg

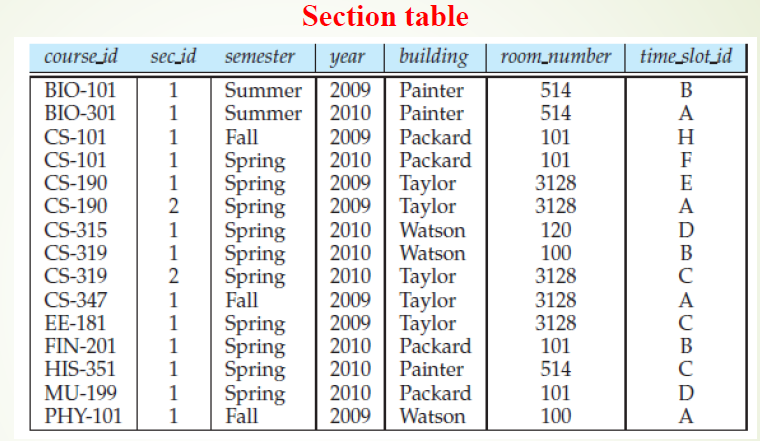
FROM Instructor

GROUP BY dept\_name

HAVING avg > 42000



HAVING IS USED AFTER GROUP BY



Find all the courses taught in the both the Fall 2009 and Spring 2010 semesters.

SELECT course\_id

FROM course\_sections

WHERE year = 2009 and semester = 'Fall' and

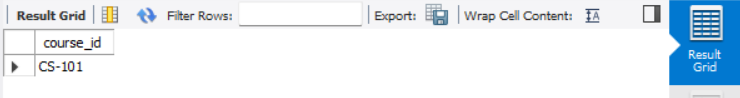
course\_id IN (

SELECT course\_id

FROM course\_sections

WHERE year = 2010 and semester = 'Spring'

) ;



Find all the courses taught in the Fall 2009 semester but not in the Spring 2010semester.

SELECT course\_id

FROM course\_sections

WHERE year = 2009 and semester = 'Fall' and

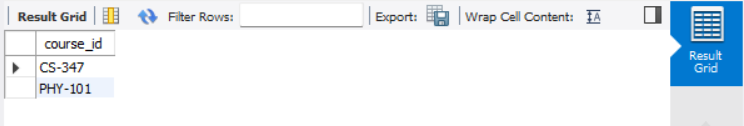
course\_id NOT IN (

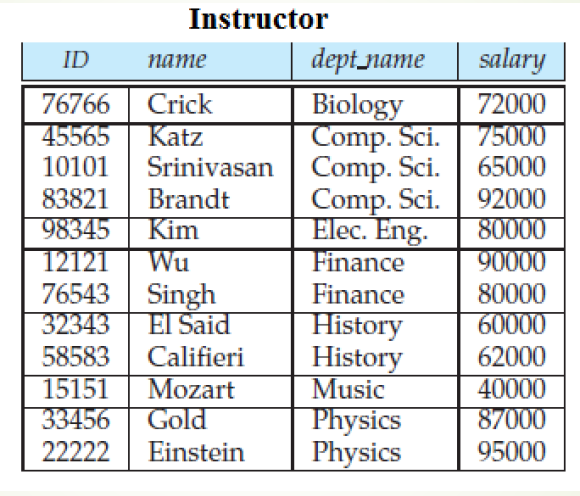
SELECT course\_id

FROM course\_sections

WHERE year = 2010 and semester = 'Spring'

) ;



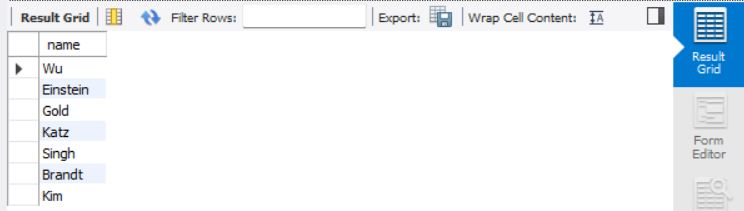


Find the names of all instructors whose salary is greater than at least one instructor in the Biology department

SELECT T.name

FROM Instructor AS T, Instructor AS S

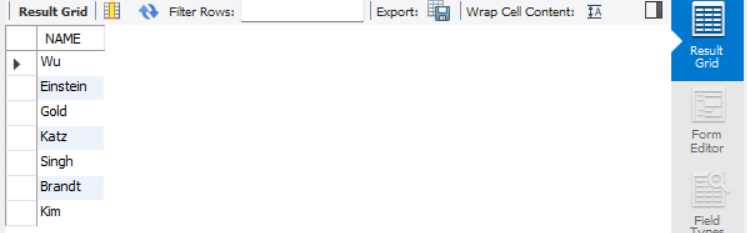
WHERE T.salary > S.salary AND S.dept\_name = 'Biology';



-OR- VIA NESTED QUERY

SELECT NAME FROM Instructor

WHERE salary > (SELECT salary FROM INSTRUCTOR WHERE dept\_name = 'Biology' ) ;



Find the department that have the highest average salary.

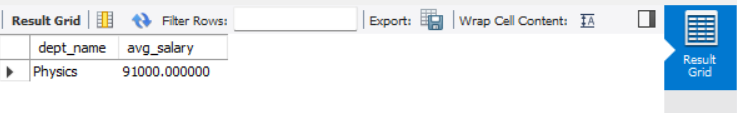
SELECT dept\_name , avg(salary) as salary FROM Instructor

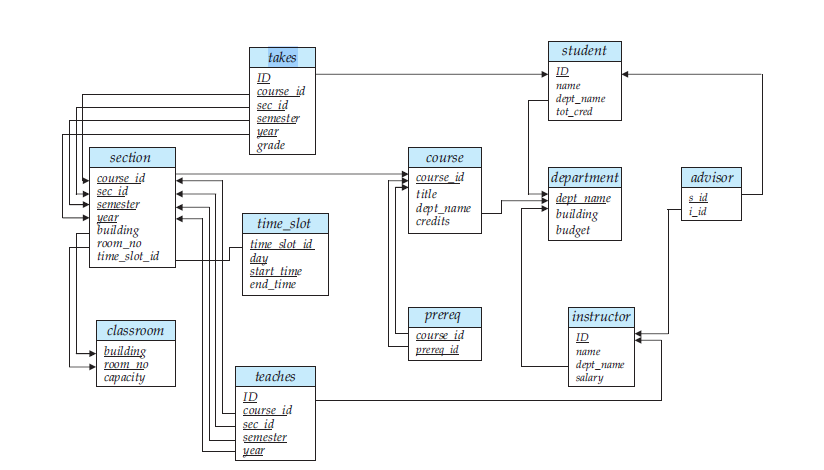
GROUP BY dept\_name

HAVING salary = max(select avg (salary)

from instructor

group by dept name);





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

SELECT course\_id FROM course WHERE credits = 3

Find the highest salary of any instructor.

SELECT max(salary) FROM instructor

Find all instructors earning the highest salary.

select ID from instructor

where salary = (select max(salary) from instructor)

Find the enrolment (count) of each section that was offered in Autumn 2009

SELECT sec\_id, COUNT(course\_id)

NATURAL JOIN TAKES

WHERE takes.year= 2009 semester = ’Autumn’

GROUP BY sec\_id

Find the maximum enrolment, across all sections, in Autumn 2009.

SELECT max(enrolment) FROM

(SELECT sec\_id , count(course\_id) as enrolment

FROM section INNER JOIN takes

WHERE year = 2009 and semester = ‘Autumn’)

For each course section offered in 2009, find the average total credits (*tot cred*) of all students enrolled in the section, if the section had at least 2 students.

SELECT

t.sec\_id AS section,

AVG(s.tot\_cred) AS avg\_total\_credits

FROM

takes AS t

NATURAL JOIN

student AS s

WHERE

t.year = 2009

GROUP BY

t.sec\_id

HAVING

COUNT(DISTINCT t.ID) >= 2;