Querying (using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.)

Instruction Sheet

Step 1: Create a table department and instructor table

Step 2: Insert data into department and instructor table

Step 3: Create a table section table and insert data into section table.

Step 4: Write the following SQL queries using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT

* Write a SQL query to find names of instructors with salary greater than that of **some**(some clause) (at least one) instructor in the Computer Science department.
* Write a SQL query to find the names of all instructors whose salary is greater than the salary of all instructors in the Biology department.(any clause)
* Write a SQL query to find the names of all instructors whose salary is greater than the salary of all instructors in the Biology department.(all clause)
* Write a SQL Query to find the name all instructors whose name is “Mozart” , Einstein”(IN clause)
* Write a SQL Query to find the name all instructors whose name is neither “Mozart” nor Einstein”(NOT IN clause)
* Write a SQL query to find all courses taught in both the Fall 2009 semester and in the Spring 2010 semester”(**Exists clause**)
* Write a SQL query to find all courses taught in both the Fall 2009 semester and in the Spring 2010 semester”(NOT **Exists clause**)
* Write a SQL query to find courses that ran in Fall 2009 or in Spring 2010 (union and union all)

Write a following simple SQL queries

* Write a SQL query to find the all details of instructors.
* Write a SQL query to find the names of all instructors :
* Write a SQL query to find the department names of all instructors.
* Write a SQL query to find the department names of all instructors, and remove duplicates.
* Write a SQL query to find all instructors in Comp. Sci. dept
* Write a SQL query to find all instructors in Comp. Sci. dept with salary > 80000
* Write a SQL query to find the Cartesian product *instructor X teaches*
* Write a SQL query to find the names of all instructors who have taught some course and the

course\_id

* Write a SQL query to Find the names of all instructors in the Art department who have taught some course and the course\_id
* Write a SQL query to Find the names of all instructors who have a higher salary than  
   some instructor in 'Comp. Sci'.
* Write a SQL query to find the names of all instructors whose name includes the substring “dar”.
* Write a SQL query to find the list in alphabetic order the names of all instructors
* Write a SQL query to find the names of all instructors with salary between $90,000  
  and $100,000 (that is, ≥ $90,000 and ≤ $100,000)
* Write a SQL query to find all instructors whose salary is null*.*Write a SQL query to delete all instructors from the finance department
* Write a SQL query to d*elete all tuples in the instructor relation for those instructors associated with a department located in the Watson building.*
* Write a SQL query to remove all tuples from the instructorrelation

Querying (using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.)

The SQL ANY and ALL Operators

The ANY and ALL operators allow you to perform a comparison between a single column value and a range of other values.

The SQL ANY Operator

The ANY operator:

* returns a boolean value as a result
* returns TRUE if ANY of the subquery values meet the condition

ANY means that the condition will be true if the operation is true for any of the values in the range.

ANY Syntax

SELECT *column\_name(s)* FROM *table\_name* WHERE *column\_name operator* ANY  
  (SELECT *column\_name*FROM *table\_name*WHERE *condition*)

**Note:** The operator must be a standard comparison operator (=, <>, !=, >, >=, <, or <=).

The SQL ALL Operator

The ALL operator:

* returns a boolean value as a result
* returns TRUE if ALL of the subquery values meet the condition
* is used with SELECT, WHERE and HAVING statements

ALL means that the condition will be true only if the operation is true for all values in the range.

ALL Syntax With SELECT

SELECT ALL *column\_name(s)* FROM *table\_name* WHERE *condition*;

ALL Syntax With WHERE or HAVING

SELECT *column\_name(s)* FROM *table\_name* WHERE *column\_name operator* ALL  
  (SELECT *column\_name*FROM *table\_name*WHERE *condition*);

**Note:** The *operator* must be a standard comparison operator (=, <>, !=, >, >=, <, or <=).

Create a table department and instructor table

Insert data into department and instructor table

Schema - department(dept\_ name, building, budget)

mysl>create table department

(dept\_name char(20),

building char(15),

budget numeric (12,2) check (budget > 0),

primary key (dept\_name));

mysql> desc department;

+----------------+-------------------+-------+------+-----------+--------+

| Field | Type | Null | Key | Default | Extra |

+----------------+-------------------+-------+------+-----------+--------+

| dept\_name | char(20) | NO | PRI | NULL | |

| building | char(15) | YES | | NULL | |

| budget | decimal(12,2) | YES | | NULL | |

+----------------+--------------------+------+------+-----------+--------+

3 rows in set (0.00 sec)

mysql>insert into department values('Biology','Watson',90000.00);

mysql>insert into department values('Comp.Sci','Taylor',100000.00);

mysql>insert into department values('Ele.Eng','Taylor',85000.00);

mysql>insert into department values('Finance','Painter',120000.00);

mysql>insert into department values('History','Painter',50000.00);

mysql>insert into department values('Music','Packard',80000.00);

mysql>insert into department values('Physics','Watson',70000.00);

mysql>commit;

mysql> select \* from department;

+----------------+------------+---------------+

| dept\_name | building | budget |

+----------------+------------+---------------+

| Biology | Watson | 90000.00 |

| Comp.Sci | Taylor | 100000.00|

| Elec.Eng | Taylor | 85000.00 |

| Finance | Painter | 120000.00 |

| History | Painter | 50000.00 |

| Music | Packard | 80000.00 |

| Physics | Watson | 70000.00 |

+--------------- +------------+----------------+

7 rows in set (0.00 sec)

Schema - instructor(ID, name, dept\_ name, salary)

mysql>create table instructor

(ID char (5),

name char (20) not null,

dept\_name char (20),

salary numeric (8,2) check (salary > 29000),

primary key (ID), foreign key (dept\_name) references department(dept\_name) );

mysql> desc instructor;

+-----------------+------------------+------+--------+-----------+--------+

| Field | Type | Null | Key | Default | Extra |

+-----------------+---------------- --+------+--------+----------+---------+

| ID | char(5) | NO | PRI | NULL | |

| name | char(20) | NO | | NULL | |

| dept\_name | char(20) | YES | MUL | NULL | |

| salary | decimal(8,2) | YES | | NULL | |

+-----------------+------------------+-------+--------+----------+---------+

4 rows in set (0.01 sec)

mysql> insert into instructor values('10101','Srinivasan','Comp.Sci',65000);

mysql> insert into instructor values('12121','Wu','Finance',90000);

mysql> insert into instructor values('15151','Mozart','Music',40000);

mysql> insert into instructor values('22222','Einstein','Physics',95000);

mysql> insert into instructor values('32343','El Said','History',60000);

mysql> insert into instructor values('33456','Gold','Physics',87000);

mysql> insert into instructor values('45565','Katz','Comp.Sci',75000);

mysql> insert into instructor values('58583','Califieri','History',62000);

mysql> insert into instructor values('76543','Singh','Finance',80000);

mysql> insert into instructor values('76766','Crick','Biology',72000);

mysql> insert into instructor values('83821','Brandt','Comp.Sci',92000);

mysql> insert into instructor values('98345','Kim','Elec.Eng',80000);

mysl>commit;

Write the following SQL queries using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT

mysql> select \* from instructor;

+-------+------------+-----------+----------+

| ID | name | dept\_name | salary |

+-------+------------+-----------+----------+

| 10101 | Srinivasan | Comp.Sci | 65000.00 |

| 12121 | Wu | Finance | 90000.00 |

| 15151 | Mozart | Music | 40000.00 |

| 22222 | Einstein | Physics | 95000.00 |

| 32343 | El Said | History | 60000.00 |

| 33456 | Gold | Physics | 87000.00 |

| 45565 | Katz | Comp.Sci | 75000.00 |

| 58583 | Califieri | History | 62000.00 |

| 76543 | Singh | Finance | 80000.00 |

| 76766 | Crick | Biology | 72000.00 |

| 83821 | Brandt | Comp.Sci | 92000.00 |

| 98345 | Kim | Elec.Eng | 80000.00 |

+-------+------------+-----------+----------+

12 rows in set (0.00 sec)

Write a SQL query to find names of instructors with salary greater than that of **some**(some clause) (at least one) instructor in the Computer Science department.

mysql> select name from instructor where salary > some (select salary from instructor where dept\_name = 'Comp.Sci');

+----------+

| name |

+----------+

| Wu |

| Einstein |

| Gold |

| Katz |

| Singh |

| Crick |

| Brandt |

| Kim |

+----------+

8 rows in set (0.00 sec)

mysql>

Write a SQL query to find names of instructors with salary greater than that of **some**(**without using some clause)** (at least one) instructor in the Computer Science department.

*mysql>select distinct T.name from instructor as T, instructor as S  
 where T.salary > S.salary and S.dept\_name =* 'Comp.Sci'*;*

mysql> select distinct T.name from instructor as T, instructor as S where T.salary > S.salary and S.dept\_name = 'Comp.Sci';

+----------+

| name |

+----------+

| Wu |

| Einstein |

| Gold |

| Katz |

| Singh |

| Crick |

| Brandt |

| Kim |

+----------+

8 rows in set (0.00 sec)

mysql> select name from instructor where salary < some (select salary from instructor where dept\_name = 'Comp.Sci');

mysql> select name from instructor where salary = some (select salary from instructor where dept\_name = 'Comp.Sci');

mysql> select name from instructor where salary <> some (select salary from instructor where dept\_name = 'Comp.Sci');

Find the names of all instructors whose salary is greater than the salary of all instructors in the Biology department.(all clause)

mysql> select name from instructor where salary > all (select salary from instructor where dept\_name = 'Biology');

+----------+

| name |

+----------+

| Wu |

| Einstein |

| Gold |

| Katz |

| Singh |

| Brandt |

| Kim |

+----------+

7 rows in set (0.00 sec)

Find the names of all instructors whose salary is less than the salary of all instructors in the Computer Science department.(all clause)

mysql> select name from instructor where salary < all (select salary from instructor where dept\_name = 'Comp.Sci');

+-----------+

| name |

+-----------+

| Mozart |

| El Said |

| Califieri |

+-----------+

3 rows in set (0.00 sec)

mysql>

mysql> select name from instructor where salary = all (select salary from instructor where dept\_name = 'Comp.Sci');

Empty set (0.00 sec)

mysql> select name from instructor where salary > any (select salary from instructor where dept\_name = 'Comp.Sci');

+----------+

| name |

+----------+

| Wu |

| Einstein |

| Gold |

| Katz |

| Singh |

| Crick |

| Brandt |

| Kim |

+----------+

8 rows in set (0.00 sec)

mysql> select name from instructor where salary = any (select salary from instructor where dept\_name = 'Comp.Sci');

+------------+

| name |

+------------+

| Srinivasan |

| Katz |

| Brandt |

+------------+

3 rows in set (0.00 sec)

mysql>

mysql> select name from instructor where salary < any (select salary from instructor where dept\_name = 'Comp.Sci');

+------------+

| name |

+------------+

| Srinivasan |

| Wu |

| Mozart |

| El Said |

| Gold |

| Katz |

| Califieri |

| Singh |

| Crick |

| Kim |

+------------+

10 rows in set (0.00 sec)

Write a SQL Query to find the name all instructors whose name is “Mozart” , Einstein”

(Using **IN** clause)

mysql> select distinct name from instructor where name IN ('Mozart', 'Einstein');

+----------+

| name |

+----------+

| Mozart |

| Einstein |

+----------+

2 rows in set (0.00 sec)

Write a SQL Query to find the name all instructors whose name is neither “Mozart” nor Einstein”

(Using **NOT** **IN** clause)

mysql> select distinct name from instructor where name NOT IN ('Mozart', 'Einstein');

+------------+

| name |

+------------+

| Srinivasan |

| Wu |

| El Said |

| Gold |

| Katz |

| Califieri |

| Singh |

| Crick |

| Brandt |

| Kim |

+------------+

10 rows in set (0.00 sec)

Write a SQL Query to find all instructors details who works in a department.

(Using exists clause)

mysql> select \* from instructor where exists (select \* from department where department.dept\_name=instructor.dept\_name);

+-------+------------+-----------+----------+

| ID | name | dept\_name | salary |

+-------+------------+-----------+----------+

| 10101 | Srinivasan | Comp.Sci | 65000.00 |

| 12121 | Wu | Finance | 90000.00 |

| 15151 | Mozart | Music | 40000.00 |

| 22222 | Einstein | Physics | 95000.00 |

| 32343 | El Said | History | 60000.00 |

| 33456 | Gold | Physics | 87000.00 |

| 45565 | Katz | Comp.Sci | 75000.00 |

| 58583 | Califieri | History | 62000.00 |

| 76543 | Singh | Finance | 80000.00 |

| 76766 | Crick | Biology | 72000.00 |

| 83821 | Brandt | Comp.Sci | 92000.00 |

| 98345 | Kim | Elec.Eng | 80000.00 |

+-------+------------+-----------+----------+

12 rows in set (0.01 sec)

mysql>select \* from instructor where exists (select \* from department where department.dept\_name=instructor.dept\_name);

mysql> select \* from instructor where exists (select dept\_name from department where department.dept\_name=instructor.dept\_name);

+-------+------------+-----------+----------+

| ID | name | dept\_name | salary |

+-------+------------+-----------+----------+

| 10101 | Srinivasan | Comp.Sci | 65000.00 |

| 12121 | Wu | Finance | 90000.00 |

| 15151 | Mozart | Music | 40000.00 |

| 22222 | Einstein | Physics | 95000.00 |

| 32343 | El Said | History | 60000.00 |

| 33456 | Gold | Physics | 87000.00 |

| 45565 | Katz | Comp.Sci | 75000.00 |

| 58583 | Califieri | History | 62000.00 |

| 76543 | Singh | Finance | 80000.00 |

| 76766 | Crick | Biology | 72000.00 |

| 83821 | Brandt | Comp.Sci | 92000.00 |

| 98345 | Kim | Elec.Eng | 80000.00 |

+-------+------------+-----------+----------+

12 rows in set (0.00 sec)

*mysql> select \* from instructor where not exists (select dept\_name from department where department.dept\_name=instructor.dept\_name);*

*Empty set (0.00 sec)*

*Schema - section*(*course\_ id*, *sec\_ id*, *semester*, *year*, *building*, *room\_ number*, *time\_ slot\_ id*)

mysql>create table section

(course\_id char (8),

sec\_id char (8),

semester char (6) check (semester in ('Fall','Winter','Spring','Summer')),

year numeric (4,0) check (year > 1701 and year < 2100),

building char (15),

room\_number char (7),

time\_slot\_id char (4),

primary key (course\_id, sec\_id, semester, year),

foreign key (course\_id) references course(course\_id)

on delete cascade, foreign key (building,room\_number) references classroom(building,room\_number) on delete set null);

*mysql> desc section;*

*+--------------+--------------+------+-----+---------+-------+*

*| Field | Type | Null | Key | Default | Extra |*

*+--------------+--------------+------+-----+---------+-------+*

*| course\_id | char(8) | NO | PRI | NULL | |*

*| sec\_id | char(8) | NO | PRI | NULL | |*

*| semester | char(6) | NO | PRI | NULL | |*

*| year | decimal(4,0) | NO | PRI | NULL | |*

*| building | char(15) | YES | MUL | NULL | |*

*| room\_number | char(7) | YES | | NULL | |*

*| time\_slot\_id | char(4) | YES | | NULL | |*

*+--------------+--------------+------+-----+---------+-------+*

mysql> insert into section values('BIO-101',1,'Summer',2009,'Painter',514,'B');

mysql> insert into section values('BIO-301',1,'Summer',2010,'Painter',514,'A');

mysql> insert into section values('CS-101',1,'Fall',2009,'Packard',101,'H');

mysql> insert into section values('CS-101',1,'Spring',2010,'Packard',101,'F');

mysql> insert into section values('CS-190',1,'Spring',2009,'Taylor',3128,'E');

mysql> insert into section values('CS-190',2,'Spring',2009,'Taylor',3128,'A');

mysql> insert into section values('CS-315',1,'Spring',2010,'Watson',120,'D');

mysql> insert into section values('CS-319',1,'Spring',2010,'Watson',100,'B');

mysql> insert into section values('CS-319',2,'Spring',2010,'Taylor',3128,'C');

mysql> insert into section values('CS-347',1,'Fall',2009,'Taylor',3128,'A');

mysql> insert into section values('EE-181',1,'Spring',2009,'Taylor',3128,'C');

mysql> insert into section values('FIN-201',1,'Spring',2010,'Packard',101,'B');

mysql> insert into section values('HIS-351',1,'Spring',2010,'Painter',514,'C');

mysql> insert into section values('MU-199',1,'Spring',2010,'Packard',101,'D');

mysql> insert into section values('PHY-101',1,'Fall',2009,'Watson',100,'A');

mysql> commit;

*mysql> select \* from section;*

mysql>

mysql> select \* from section;

+-----------+--------+----------+------+----------+-------------+--------------+

| course\_id | sec\_id | semester | year | building | room\_number | time\_slot\_id |

+-----------+--------+----------+------+----------+-------------+--------------+

| BIO-101 | 1 | Summer | 2009 | Painter | 514 | B |

| BIO-301 | 1 | Summer | 2010 | Painter | 514 | A |

| CS-101 | 1 | Fall | 2009 | Packard | 101 | H |

| CS-101 | 1 | Spring | 2010 | Packard | 101 | F |

| CS-190 | 1 | Spring | 2009 | Taylor | 3128 | E |

| CS-190 | 2 | Spring | 2009 | Taylor | 3128 | A |

| CS-315 | 1 | Spring | 2010 | Watson | 120 | D |

| CS-319 | 1 | Spring | 2010 | Watson | 100 | B |

| CS-319 | 2 | Spring | 2010 | Taylor | 3128 | C |

| CS-347 | 1 | Fall | 2009 | Taylor | 3128 | A |

| EE-181 | 1 | Spring | 2009 | Taylor | 3128 | C |

| FIN-201 | 1 | Spring | 2010 | Packard | 101 | B |

| HIS-351 | 1 | Spring | 2010 | Painter | 514 | C |

| MU-199 | 1 | Spring | 2010 | Packard | 101 | D |

| PHY-101 | 1 | Fall | 2009 | Watson | 100 | A |

+-----------+--------+----------+------+----------+-------------+--------------+

15 rows in set (0.01 sec)

mysql>

Write a SQL query to find all courses taught in both the Fall 2009 semester and in the Spring 2010 semester”(**Exists clause**)

mysql> select course\_id from section as S where semester = 'Fall' and year = 2009 and

exists (select \* from section as T where semester = 'Spring' and year= 2010

and S.course\_id = T.course\_id);

+-----------+

| course\_id |

+-----------+

| CS-101 |

+-----------+

1 row in set (0.00 sec)

mysql>

mysql> select course\_id from section as S where semester = 'Fall' and year = 2009 and

not exists (select \* from section as T where semester = 'Spring' and year= 2010

and S.course\_id = T.course\_id);

+-----------+

| course\_id |

+-----------+

| CS-347 |

| PHY-101 |

+-----------+

2 rows in set (0.00 sec)

**mysql>**

Find courses that ran in Fall 2009 or in Spring 2010(union )

mysql> (select course\_id from section where semester='Fall' and year=2009)

union

(select course\_id from section where semester='Spring' and year=2010);

+-----------+

| course\_id |

+-----------+

| CS-101 |

| CS-347 |

| PHY-101 |

| FIN-201 |

| MU-199 |

| HIS-351 |

| CS-319 |

| CS-315 |

+-----------+

8 rows in set (0.00 sec)

mysql>

Find courses that ran in Fall 2009 or in Spring 2010(union all)

mysql> (select course\_id from section where semester='Fall' and year=2009)

-> union all

-> (select course\_id from section where semester='Spring' and year=2010);

+-----------+

| course\_id |

+-----------+

| CS-101 |

| CS-347 |

| PHY-101 |

| CS-101 |

| FIN-201 |

| MU-199 |

| HIS-351 |

| CS-319 |

| CS-319 |

| CS-315 |

+-----------+

10 rows in set (0.00 sec)