**Database Systems – V5 - Spring 2021**

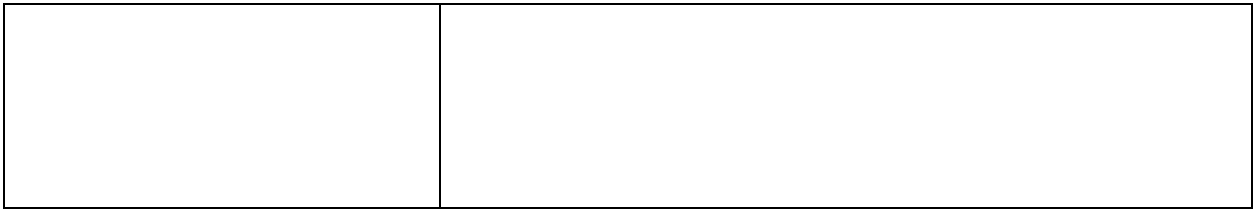
**Instructor Name: Mehr un Nisa**

**Submission Deadline: Monday 27th May,2021** before class

**Name: Mehroz Mustafa S-ID: F2019266178  
  
Assignment Submission Guidelines:**

* Assignments will not be accepted after deadline.
* Copied assignments will be marked straight zero. No tolerance.

1. **Notes:**



|  |  |  |  |
| --- | --- | --- | --- |
| SUM() | MIN() | FW**G**HOS | HAVING [condition] |
| MAX() | AVG() | ORDER BY [colname] |  |
| COUNT() |  | GROUP BY [colname] |  |

1. **Joins Examples**

Problem Set 1

*Given tables created with these commands:*

CREATE TABLE A (a int);

CREATE TABLE B (b int);

INSERT INTO A VALUES (1), (2), (3), (4);

INSERT INTO B VALUES (3), (4), (5), (6);

*What’s the output for each of the following:*

SELECT \* FROM A INNER JOIN B ON A. a = B. b;

a | b

--+--

3 | 3

4 | 4

SELECT \* FROM A LEFT OUTER JOIN B ON A. a = B. b;

a | b

--+-----

1 | null

2 | null

3 | 3

4 | 4

SELECT \* FROM A RIGHT OUTER JOIN B ON A. a = B. b;

a | b

-----+----

3 | 3

4 | 4

null | 5

null | 6

SELECT \* FROM A FULL OUTER JOIN B ON A. a = B. b;

a | b

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

1 | null

2 | null

3 | 3

4 | 4

null | 6

null | 5

1. **SQL Practice**

Problem Set 2

CREATE TABLE Movies (

id int,

name varchar(30),

budget int,

gross int,

rating int,

year int,

PRIMARY KEY (id)

);

CREATE TABLE Actors (

id int,

name varchar(30),

age int,

PRIMARY KEY (id)

);

CREATE TABLE ActsIn (

mid int,

aid int,

FOREIGN KEY (mid) REFERENCES Movies (id), FOREIGN KEY (aid) REFERENCES Actors (id), PRIMARY KEY (mid, aid)

);

What is the number of movies, and the average rating of all movie that the actor” Patrick Stewart” has appeared in?

SELECT count (\*), avg(rating)

FROM Movies as M, ActsIn as AI, Actors as

AWHERE M.id = AI.mid

AND A.id = AI. aid AND A.name = “Patrick Stewart”;

What is the minimum age of an actor who has appeared in a movie where the gross of the movie has been over $1,000,000,000?

SELECT min (age)

FROM Movies as M, ActsIn as AI, Actors as

AWHERE M.id = AI. mid AND AI. aid = A.id

AND gross > 1000000000;

1. **Self Join**

Problem Set 3

*Consider the following over simplified Employee table*

CREATE TABLE Employees (

id int,

bossOf int

);

Suppose all employees have an id which is not null. How would we find the id of all employees who are the boss of at least one other employee?

We want select table alias. col name (e2.id in this case) because otherwise our output would

contain a row for each pair of id, bossOf. This means we would get each boss to employee

pairing whereas we only want the boss ids.

*What do we select? (select \* vs select table alias.col name)*

Consider the case with employees (1, NULL), (2, NULL), (5, 1), (5, 2), (5, NULL), (3, NULL). How many employees is id = 5 the boss of?

This question has a typo, and is malformed.

1. **Subqueries**

Problem Set 4

*Consider the following tables*

*CREATE TABLE Movie (*

movie\_name VARCHAR (75),

movie\_id INT,

director\_id INT,

year\_released INT,

budget INT,

PRIMARY KEY (movie\_id),

FOREIGN KEY (director\_id) REFERENCES Director(director\_id)

*);*

CREATE TABLE Director (

director\_id INT,

director\_name VARCHAR (75),

director\_country VARCHAR (75),

PRIMARY KEY (director\_id)

*);*

Find the id and name of all directors who have directed more than 20 movies.

SELECT D.director\_id, D.director\_name FROM Director AS D, Movie AS M WHERE D.director\_id = M.director\_id GROUP BY D.director\_name, D.director\_idHAVING COUNT(\*) > 20;

For each director, find the corresponding movie that has the highest budget.

WITH MovieMaxBudget AS (SELECT M.director\_id AS director\_id, max(M.budget) AS max\_budget FROM Movie M GROUP BY M.director\_id)SELECT D.director\_name, M.movie\_name FROM Movie AS M, Director AS D, MovieMaxBudget AS MMB WHERE M.director\_id = D.director\_id AND D.director\_id = MMB.director\_id AND MMB.max\_budget = M.budget;

*Consider the following tables*

CREATE TABLE Class (

dept VARCHAR(6),

number INTEGER,

title VARCHAR(75),

PRIMARY KEY (dept, number)

);

CREATE TABLE Instructor (

username VARCHAR(8),

fname VARCHAR(50),

lname VARCHAR(50),

started\_on CHAR(10),

PRIMARY KEY (username)

);

CREATE TABLE Teaches (

username VARCHAR(8),

dept VARCHAR(6),

number INTEGER,

PRIMARY KEY (username, dept, number),

FOREIGN KEY (username) REFERENCES Instructor(username), FOREIGN KEY (dept, number) REFERENCES Class(dept, number)

);

How many classes are being taught by at least one instructor?

SELECT count (\*) AS class\_count

FROM (SELECT 1 FROM Teaches

GROUP BY dept, number) X;

Which instructors teach more than 1 class? Give the username, first name, and last name of these instructors.

SELECT I. username, I.fname, I.lname

FROM Instructor AS I WHERE 1 < (SELECT COUNT (\*)

FROM Teaches AS T WHERE T. username = I. username);

SELECT I. username, I.fname, I.lname

FROM Instructor AS I, Teaches AS T

WHERE I. username = T. username

GROUP BY I. username, I.fname, I.lname

HAVING COUNT (\*) > 1;

Which CSE courses do neither Dr. Levy (‘levy’) nor Dr. Wetherall (‘djw’) teach? Give the department, number, and title of these courses.

SELECT \* FROM Class AS C

WHERE C. dept = 'CSE' AND

C. number NOT IN (SELECT C. number

FROM Class AS C, Teaches AS T

WHERE C. dept = T. dept AND

C. number = T. number AND

(T. username = 'levy' OR

T. username = 'djw'));

*Consider the following tables*

Clinic(cid, name, street, state)

Equipment(eid, type, model)

Assignment(cid, eid)

Finds the count of clinics that do not have a fridge (of model 1004) assigned to it.

SELECT COUNT (\*)

FROM Clinic AS C

WHERE NOT EXISTS (

SELECT \* FROM Assignment AS A, Equipment AS E

WHERE C.cid = A.cid

AND A.eid = E.eid

AND E.type = ’Fridge’

AND E.model = 1004);

**Problem:** You are the lead organizer of managing the program committee for a conference at UMT. Thefollowing database stores information about papers submitted to the conference (table Paper), reviewers on the program committee (table Reviewer), and the assignment of reviewers to papers (table Reviews). Each reviewer on the program committee will have to review a set of papers. Each paper will be reviewed by some subset of reviewers.

Paper (pid, title)

Reviewer (rid, name)

Reviews (rid, pid)

* pid is a unique paper identifier and the primary key of the Paper table.
* rid is a unique reviewer identifier and the primary key of the Reviewer table.
* Reviews.rid is a foreign key that references Reviewer.rid.
* Reviews.pid is a foreign key that references Paper.pid.
* A reviewer is assigned zero or more papers.
* A paper is assigned zero or more reviewers.

1. Write a SQL query that finds all papers with fewer than three reviewers assigned to them. The output of the query should be a list of paper titles. The result should include papers without any reviewers assigned to them.

Answer (write a SQL query): -

SELECT P. title FROM Paper P

LEFT OUTER JOIN Reviews X ON P.pid = X.pid

GROUP BY P.pid, P. title

HAVING count (\*) < 3

1. Write a SQL query that finds the reviewers with the most papers assigned to them. There can be more than one such reviewer. The output of the query should be a list of reviewer names. A reviewer should be listed if no other reviewer has strictly more papers to review.

Answer (write a SQL query): -

SELECT R. name FROM Reviewer R, Reviews X1

WHERE R. rid = X1.rid

GROUP BY R. rid, R. name

HAVING count (\*) >= ALL (SELECT count (\*)

FROM Reviews X2GROUP BY X2. rid)