**Name: Mozeb Ahmed Khan**

**Roll No: 20F-0161**

**Section: BCS-4A**

**Course: Database System Course**

**Assignment No: 04**

**Part 1 - Joins**

**Task 1:**

Select CONCAT\_WS('\_', S.first\_name, S.last\_name) As full\_name, Sum(G.grades) As total\_score

From Graded G Inner Join Students S

On G.student\_ID=S.student\_ID

And 2 >= (SELECT COUNT(DISTINCT grades) FROM Graded B WHERE B.grades >= G.grades)

Group by G.student\_ID, S.first\_name, S.last\_name, G.grades

ORDER BY G.grades DESC;

**Screenshot:**

Graphical user interface, text, application

Description automatically generated

**Task 2: Self Join**

Select A.student\_ID, A.exam\_id, A.grades

From Graded A, Graded B

Where A.grades=B.grades

And A.grades < (Select avg(B.grades) From Graded B)

Group by A.exam\_id,A.student\_ID,A.ID,A.grades;

**Screenshot:**

Graphical user interface, text

Description automatically generated

Graphical user interface, text

Description automatically generated

Graphical user interface, text

Description automatically generated

**Task 3:**

Select CONCAT\_WS('\_', S.first\_name, S.last\_name) As full\_name, G.student\_id,G.grades

from (Select exam\_id, MAX(grades) As High\_Marks From Graded Group By exam\_id) E

Inner Join Graded G

On E.exam\_id = G.exam\_id

And E.high\_Marks = G.grades

Inner Join Students S

On S.student\_ID=G.student\_ID;

**Screenshot:**

Graphical user interface, text, application

Description automatically generated

**Task 4:**

**Screenshot:**

**Task 5:**

SELECT Exams.\*

FROM ((Exams

JOIN Graded ON Exams.exam\_id = Graded.exam\_id)

JOIN Students ON Graded.student\_ID = Students.student\_ID And Students.first\_name='Corey');

**Screenshot:**

Graphical user interface, application

Description automatically generated

**Part 2 – Sub-queries**

**Create, Insert And Display Databases:**

**Paintings:**

Create Table Paintings(

id int NOT NULL,

painting\_name VARCHAR(25) NOT NULL,

artist\_id int NOT NULL,

listed\_price float NOT NULL,

PRIMARY KEY (id)

);

INSERT INTO Paintings(id,painting\_name,artist\_id,listed\_price)

values (11,'Miracle',1,300.00), (12,'Sunshine',1,700.00),

(13,'Pretty Women',2,2800.00), (14,'Handsome Man',2,2300.00),

(15,'Barbie',3,250.00), (16,'Cool Painting',3,5000.00),

(17,'Black Square#1000',3,50.00), (18,'Mountains',4,1300.00);

Graphical user interface

Description automatically generated

**Artists:**

Create Table Artists(

id int NOT NULL,

first\_name VARCHAR(25) NOT NULL,

last\_name VARCHAR(25) NOT NULL,

PRIMARY KEY (id)

);

INSERT INTO Artists(id,first\_name,last\_name)

values (1,'Thomas','Black'), (2,'Kate','Simth'),

(3,'Natalie','Wein'), (4,'Francesco','Benelli');

Graphical user interface, table

Description automatically generated

**Collectors:**

Create Table Collectors(

id int NOT NULL,

first\_name VARCHAR(25) NOT NULL,

last\_name VARCHAR(25) NOT NULL,

PRIMARY KEY (id)

);

INSERT INTO Collectors(id,first\_name,last\_name)

values (101,'Brandon','Cooper'), (102,'Laura','Fisher'),

(103,'Christina','Buffet'), (104,'Steve','Stevenson');

Graphical user interface, application

Description automatically generated

**Sales:**

Create Table Sales(

id int NOT NULL,

Sales\_date date NOT NULL,

painting\_id int NOT NULL,

artist\_id int NOT NULL,

collector\_id int NOT NULL,

sales\_price float NOT NULL,

PRIMARY KEY (id),

Foreign Key(id) references Paintings(id),

Foreign Key(id) references Artists(id),

Foreign Key(id) references Collectors(id)

);

INSERT INTO Sales(id,Sales\_date,painting\_id,artist\_id,collector\_id,sales\_price)

Values (1001,'01-NOV-2021',13,2,104,2500.00), (1002,'10-NOV-2021',14,2,102,2300.00),

(1003,'10-NOV-2021',11,1,102,300.00), (1004,'15-NOV-2021',16,3,103,4000.00),

(1005,'22-NOV-2021',15,3,103,200.00), (1006,'22-NOV-2021',17,3,103,50.00);

Graphical user interface

Description automatically generated with low confidence

**Task 1:**

Select painting\_name, listed\_price

From Paintings

Where listed\_price > (Select Avg(listed\_price) From Paintings );

**Screenshot:**

Text, email

Description automatically generated

**Task 2:**

Select CONCAT\_WS('\_', first\_name, last\_name) As full\_name

From Collectors

Where id In (Select collector\_id From Sales);

**Screenshot:**

Graphical user interface, text, email

Description automatically generated

**Task 3:**

Select artist\_id,Sum(sales\_price) As total\_sales

From Sales

Where artist\_id In (Select artist\_id From Sales)

Group By artist\_id

Order By Sum(sales\_price) DESC;

**Screenshot:**

Graphical user interface, text

Description automatically generated

**Task 4:**

Select CONCAT\_WS('\_', C.first\_name, C.last\_name) As full\_name,Count(S.collector\_id) As total\_purchased

From Sales S, Collectors C

Where C.id In (Select S.collector\_id From Sales)

Group By S.collector\_id,C.first\_name, C.last\_name;

**Screenshot:**

Graphical user interface, text, application, email

Description automatically generated

**Task 5:**

Select CONCAT\_WS('\_', first\_name, last\_name) As full\_name

From Artists

Where id Not In (Select artist\_id From Sales);

**Screenshot:**

Graphical user interface, text, email

Description automatically generated

**The end. Thank you.**