CS 486/586 Midterm
Spring 2022

SUBMITTED BY:- PARTH PARASHAR

Updates & Clarifications:

- Q9: You do not have to use UNION for this question. Removed ~~*Use UNION in your answer. (Using UNION is required.)*~~

Instructions.

This file contains the Midterm Exam for CS 486-586 - Spring 2022.

To complete this exam, please make a copy of the google doc and edit it. When you are ready to turn in your exam, please save the doc as a PDF and turn it in on Canvas.

The midterm is ***due by midnight on Friday May 6***. The extended amount of time for the exam is intended so that students can work the exam into their schedules. I anticipate the exam will take on the order of 6-15 hours.

The exam is required to be done individually. This exam is ***open book, open notes, open Internet, open everything.*** However, if you use material from the Internet or sources other than the textbook, course slides, your notes, or the PostgreSQL documentation, ***you must cite the web site or material that you used.*** For example, you could include a link to the web site you used or the name of a book you used in your answer.

200 points total.

Please do be sure to read questions completely and note that some questions ask you to include the results in addition to the query. Please pay attention to

the "Include in your answer" notes in the questions. Please do not overcomplicate your answers and write your answers in the most optimized way you can possible.

Sport Schema:

Sports

Id	Name	Type	METs	DifficultyRank
1	Volleyball	team	4.0	20
2	Football	team	8.0	10
3	Basketball	team	6.0	4
4	Weight lifting	individual	6.0	44
5	Boxing	individual	7.0	1

Players

Id	Name	Sport	Age	WeightClass
1	Arin	2	22	2
2	Inaya	1	34	2
3	Ashley	5	40	3
4	Charlie	4	35	4
5	Owen	3	52	1
6	Li	2	19	3
7	Zaina	3	20	1

Calories

id	Sport	WeightClass	AverageCalories
1	1	2	285
2	2	2	498
3	2	3	623
4	3	1	365

5	3	2	715
6	4	4	526
7	5	3	434

Notes:

Football refers to international football or soccer
METs: Metabolic Equivalents
Weight class: could be a number from 1-5
AverageCalories: Refers to the averaged amount of calories burnt by an individual in a specific weight class during one hour.
Sources:
https://burned-calories.com/sport/
https://community.plu.edu/~chasega/met.html
https://healthysportindex.com/report/physical-activity/
https://www.espn.com/espn/page2/sportSkills

Q1. (15 pts)

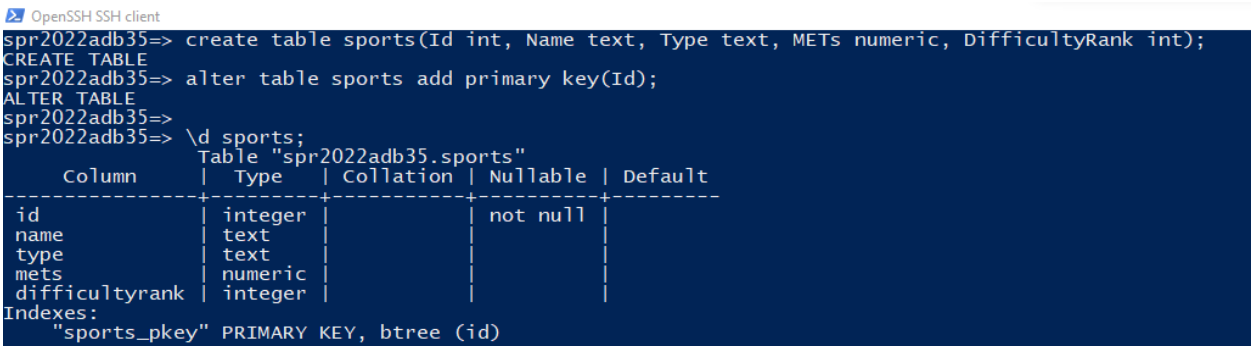
Write Create Table commands to create the three tables above in your database and create a primary key for each table. Add any foreign keys as appropriate. Include in your answer the table creation commands and any command to create keys and foreign keys. Read the notes below and include criteria according to your answer.

Notes:

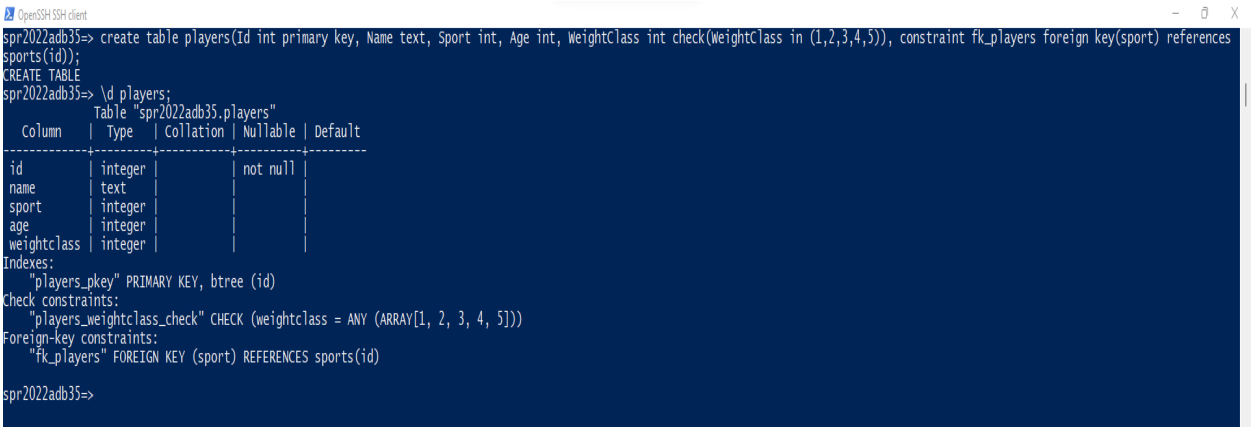
- You may create the primary keys and foreign keys either as part of the table creation statement or using Alter Table to add the keys after you create the table
- Assume that the names in all tables can be of arbitrary length
- Assume that the weight class is a number from 1-5
- Assume that metabolic equivalents (METs) are floating point numbers
- All other fields should be assumed to be integers

Answer: -

create table sports(Id int, Name text, Type text, METs numeric, DifficultyRank int);
alter table sports add primary key(Id);



create table players(Id int primary key, Name text, Sport int, Age int, WeightClass int check(WeightClass in (1,2,3,4,5)), constraint fk_players foreign key(sport) references sports(id));



Create table calories(Id int PRIMARY KEY, sport int,weightclass int CHECK(weightclass IN(1,2,3,4,5)), averagecalories int,constraint fk_calories FOREIGN KEY(sport) REFERENCES
sports(id));

```
spr2022adb35=> Create table calories(Id int PRIMARY KEY, sport int,weightclass int CHECK(weightclass IN(1,2,3,4,5)), averagecalories int,constraint fk_calories FOREIGN KEY(sport) REFERENCES
sports(id));
CREATE TABLE
spr2022adb35=> \d calories;
                Table "spr2022adb35.calories"
     Column      |  Type   | Collation | Nullable | Default
-----------------+---------+-----------+----------+---------
 id              | integer |           | not null |
 sport           | integer |           |          |
 weightclass     | integer |           |          |
 averagecalories | integer |           |          |
Indexes:
    "calories_pkey" PRIMARY KEY, btree (id)
Check constraints:
    "calories_weightclass_check" CHECK (weightclass = ANY (ARRAY[1, 2, 3, 4, 5]))
Foreign-key constraints:
    "fk_calories" FOREIGN KEY (sport) REFERENCES sports(id)

spr2022adb35=>
```

Q2. (10 pts) Create csv files for the data shown in the tables above. Load all the data into the tables. You may use either the command line or the phppgadmin gui. Include your work in your answer.

Answer: -

The first step is to make the CSV files. The CSV files are given below: -

The second step is to get the files into the linux system using the scp command as given below: -

Go to directory where the files are present,

Then use the scp command as given below: -

scp * parth2@131.252.208.103:/u/parth2/Desktop/DBMS/midterm/

This follows the following format: -

scp * <host profile name>@<Host IP address>:<Directory where files need to be copied>

```
PS C:\Users\vivek\OneDrive\Desktop\Spring-2022\db\Files for initial Database\Mid-term>
PS C:\Users\vivek\OneDrive\Desktop\Spring-2022\db\Files for initial Database\Mid-term> scp * parth2@131.252.208.103:/u/parth2/Desktop/DBMS/midterm/
parth2@131.252.208.103's password:
calories.csv                                                                                  100%  117     4.7KB/s   00:00
players.csv                                                                                   100%  141     3.1KB/s   00:00
sports.csv                                                                                    100%  162     5.7KB/s   00:00
PS C:\Users\vivek\OneDrive\Desktop\Spring-2022\db\Files for initial Database\Mid-term>
PS C:\Users\vivek\OneDrive\Desktop\Spring-2022\db\Files for initial Database\Mid-term>
PS C:\Users\vivek\OneDrive\Desktop\Spring-2022\db\Files for initial Database\Mid-term>
PS C:\Users\vivek\OneDrive\Desktop\Spring-2022\db\Files for initial Database\Mid-term>
PS C:\Users\vivek\OneDrive\Desktop\Spring-2022\db\Files for initial Database\Mid-term>
```

The last step is to use the copy command to copy the files into the database.
\copy sports from sports.csv with csv header;
\copy players from players.csv with csv header;
\copy calories from calories.csv with csv header;

OpenSSH SSH client

```
spr2022adb35=> \copy sports from sports.csv with csv header;
COPY 5
spr2022adb35=> \copy players from players.csv with csv header;
COPY 7
```

OpenSSH SSH client

```
spr2022adb35=> \copy calories from calories.csv with csv header;
COPY 7
spr2022adb35=>
```

The results can be verified by using the select * query commands as given below: -

Q3. (10 pts) Write a SQL query to make a list of player's name, sport name, and sport type they play.

Answer: -

select p.name, s.name, s.type from sports s inner join players p on s.id = p.sport;

```
spr2022adb35=> select p.name, s.name, s.type from sports s inner join players p on s.id = p.sport;
  name   |      name       |    type
---------+-----------------+------------
 Arin    | Football        | team
 Inaya   | Volleyball      | team
 Ashley  | Boxing          | individual
 Charlie | Weight lifting  | individual
 Owen    | Basketball      | team
 Li      | Football        | team
 Zaina   | Basketball      | team
(7 rows)
```

For questions 4 - 7, make sure to use one of the following topics from the following list that we learned in the class to answer the questions from 4 to 7.

a) **Three-table join**
b) **Aggregates**
c) **Group by and Having**

d) Subquery

Q4. (10 pts) Find the name of the players and the average calories they burn in one hour for those players who play a sport with a difficulty rank lower than 10.

Include your **typed** query and the result in your answer and a **screenshot** of your results.

Answer: -

Select p.name, SUM(c.averagecalories) as averagecalories from sports s inner join players p on s.id=p.sport inner join calories c on s.id=c.sport where s.difficultyrank<10 group by p.name;

```
spr2022adb35=>
spr2022adb35=>
spr2022adb35=> Select p.name, SUM(c.averagecalories) as averagecalories
spr2022adb35-> from sports s inner join players p
spr2022adb35-> on s.id=p.sport inner join calories c on s.id=c.sport
spr2022adb35-> where s.difficultyrank<10 group by p.name;
  name   | averagecalories
---------+-----------------
 Owen    |            1080
 Ashley  |             434
 Zaina   |            1080
(3 rows)

spr2022adb35=>
```

Q5. (10 pts) Write a query to list names of all sports that at least 2 players play.

Include your **typed** query and the result in your answer and a **screenshot** of your results.

Answer: -

Select s.Name from players p inner join sports s on p.sport=s.id group by s.id,s.name having count(p.id)>=2;

```
spr2022adb35=> Select s.Name from players p inner join sports s on p.sport=s.id group by s.id,s.name having count(p.id)>=2;
    name
------------
 Football
 Basketball
(2 rows)
```

Q6. (10 pts)

Write two queries:

1) Count the number of players older than 33

2) Find the average age of players older than 33 . Use only the players table.

Include your **typed** query and the result in your answer and a **screenshot** of your results.

Answer: -

1) Select count(age) from players where age>33;

```
spr2022adb35=> Select count(age) from players where age>33;
 count
-------
     4
(1 row)
```

2) Select avg(age) from players where age>33;

```
spr2022adb35=> Select avg(age) from players where age>33;
         avg
---------------------
 40.2500000000000000
(1 row)
```

Q7. (10 pts) Write a SQL query to find the sport names and difficulty rank of the sports that Charlie does NOT play.

Include your **typed** query and the result in your answer and a **screenshot** of your results.

Answer: -

Select name,difficultyRank from sports where Id not in(select sport from players where name='Charlie');

```
spr2022adb35=> Select name,difficultyRank from sports where Id not in(select sport from players where name='Charlie');
    name    | difficultyrank
------------+----------------
 Volleyball |             20
 Football   |             10
 Basketball |              4
 Boxing     |              1
(4 rows)

spr2022adb35=>
```

Q8. (15 pts) Write two SQL queries to find average MET's of sports that Inaya and Li play. One query should use a subquery in the WHERE clause, one query should not use a subquery.

Include your **typed** query and the result in your answer and a **screenshot** of your results.

Answer: -

Using subquery: -

Select AVG(METs) as AverageMETs from sports where Id in(select sport from players where Name in('Inaya','Li'));

```
spr2022adb35=> Select AVG(METs) as AverageMETs from sports where Id in(select sport from players where Name in('Inaya','Li'));
    averagemets
--------------------
 6.0000000000000000
(1 row)

spr2022adb35=>
```

Without subquery: -

Select AVG(s.mets) as AvgMETs from players p inner join sports s on p.sport=s.id where p.name in ('Inaya','Li');

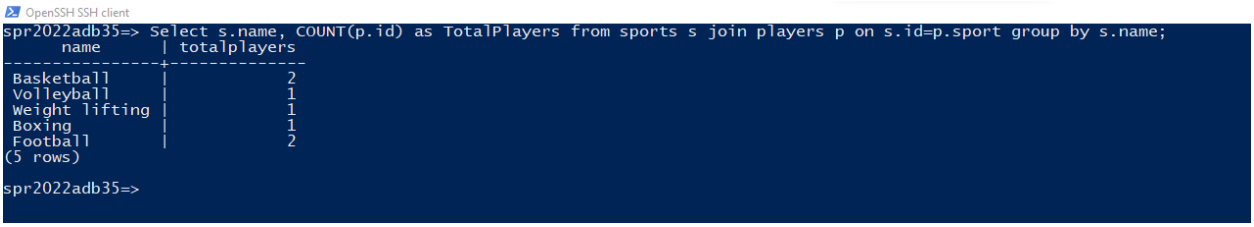
```
spr2022adb35=> Select AVG(s.mets) as AvgMETs from players p inner join sports s on p.sport=s.id where p.name in ('Inaya','Li');
      avgmets
--------------------
 6.0000000000000000
(1 row)

spr2022adb35=>
```

Q9: (15 pts) Write a query to find for each sport, the number of players who play that sport. List sport name and count in your result. Name the count column TotalPlayers.

Include your **typed** query and the result in your answer and a **screenshot** of your results.

Answer: -

Select s.name, COUNT(p.id) as TotalPlayers from sports s join players p on s.id=p.sport group by s.name;



Q10: (15 pts) Write two relational algebra expressions to find the name of the sport, METs (Metabolic Equivalents) value, and the amount of calories burnt in a sport that Charlie plays.

The two relational algebra expressions **must be different**, but guarantee to give the **same result**.

Answer: - $\pi_{S.Name,\ S.METs,\ C.AverageCalories}(\sigma_{S.Id\ =\ P.Sport\ \ AND\ S.Id\ =\ C.Sport\ AND\ P.Name\ =\ 'Charlie'}$ $(Sports\ S \quad Players\ P \quad Calories\ C))$

$\pi_{S.Name,\ S.METs,\ C.AverageCalories}(Players\ P \bowtie P.Name\ =\ 'Charlie'\ AND\ S.Id\ =\ P.sport$ $Sports\ S \bowtie S.Id\ =\ C.Sport\ Calories\ C)$

Q11: (15 pts) Write a SQL query to find the name of the sport(s) that has the most player(s).

Include your **typed** query and the result in your answer and a **screenshot** of your results.

Answer: - select s.name from sports s where s.id IN (select p.sport from players p group by p.sport having count(*) = (select max(fav_s.fav_s_count) from (select count(*) as fav_s_count from players p group by p.sport) as fav_s));

```
spr2022adb35=> select s.name from sports s where s.id IN (select p.sport from players p group by p.sport having count(*) = (select max(fav_s.fav_s_count) from (select count(*) as fav_s_count
 from players p group by p.sport) as fav_s));
    name
------------
 Basketball
 Football
(2 rows)

spr2022adb35=>
```

Q12: (10 pts) Give an example of a query that can be expressed in SQL, but which cannot be expressed in pure relational algebra. Write the SQL and explain why the query cannot be expressed in RA. (You can refer to video lectures)

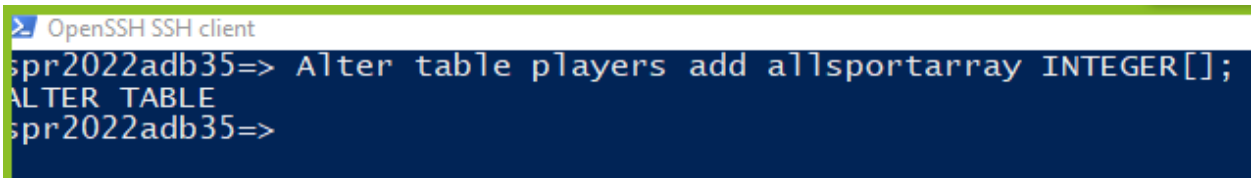
Answer: - select weightclass from players;
This query can be expressed in SQL but cannot be expressed in relational algebra because of the presence of duplicate entries which will be omitted by the Relational algebra queries.
This can further be explained by the fact that Relational algebra removes duplicates as it considers it as a set (while using basic project operator).
To find the correct answer, we need bag versions of the operators.

Q13: (10 pts) Some players may play different sports; for example, Owen and Zaina play both Basketball and Weight lifting.
Modify the Sport Schema by adding a column called allSportsArray that uses an *Array Type* to allow a player to play multiple sports.
Write DDL statements to modify the schema and write insert or update commands to insert rows showing sports (sport ids) Owen and Zaina plays.
Note: DDL or Data Definition Language (DDL) Statements, let's you make alteration to schema, it can be as simple as adding a new column to a table, inserting values in a table, changing values, or dropping a column, etc.

Include your **typed** query and the result in your answer and a **screenshot** of your results.
No need to drop the existing sport column, you may leave it.

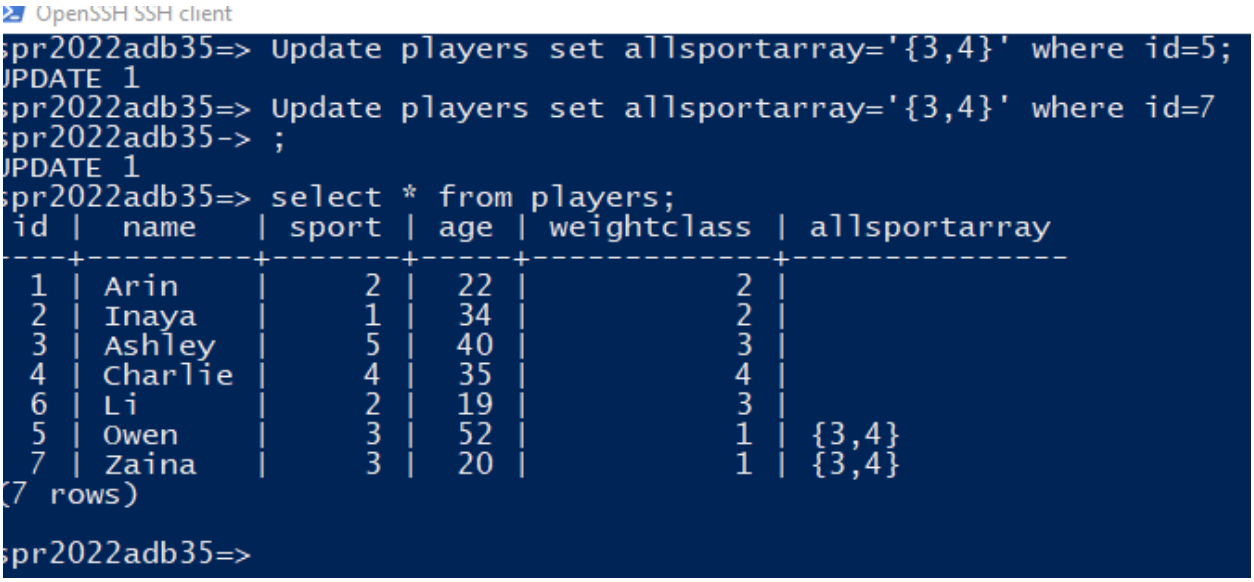
Answer: -

Alter table players add allsportarray INTEGER[];



Update player set allsportarray='{3,4}'where id=5;

Update players set allsportarray='{3,4}' where id=7;



Q14: (10 pts) First, insert the information from the sport column into the new allSportsArray column for all players other than Owen and Zaina. Using this new allSportsArray column, find all players that do weight lifting as one of their sports.

Include your **typed** query and the result in your answer and a **screenshot** of your results.

Answer: -

Update players set allsportarray='{2}' where id=1;

Update players set allsportarray='{1}' where id=2;

Update players set allsportarray='{5}' where id=3;

Update players set allsportarray='{4}' where id=4;
Update players set allsportarray='{2}' where id=6;
select * from players;
Select p.name from players p, sports s where s.id=any(allsportarray) and s.name='Weight lifting';

```
spr2022adb35=> Update players set allsportarray='{2}' where id=1;
UPDATE 1
spr2022adb35=> Update players set allsportarray='{1}' where id=2;
UPDATE 1
spr2022adb35=> Update players set allsportarray='{5}' where id=3;
UPDATE 1
spr2022adb35=> Update players set allsportarray='{4}' where id=4;
UPDATE 1
spr2022adb35=> Update players set allsportarray='{2}' where id=6;
UPDATE 1
spr2022adb35=> select * from players;
 id |  name   | sport | age | weightclass | allsportarray
----+---------+-------+-----+-------------+---------------
  5 | Owen    |     3 |  52 |           1 | {3,4}
  7 | Zaina   |     3 |  20 |           1 | {3,4}
  1 | Arin    |     2 |  22 |           2 | {2}
  2 | Inaya   |     1 |  34 |           2 | {1}
  3 | Ashley  |     5 |  40 |           3 | {5}
  4 | Charlie |     4 |  35 |           4 | {4}
  6 | Li      |     2 |  19 |           3 | {2}
(7 rows)
```

```
spr2022adb35=> Select p.name from players p, sports s where s.id=any(allsportarray) and s.name='Weight lifting';
  name
---------
 Owen
 Zaina
 Charlie
(3 rows)
```

Q15: (10 pts). Create an enum type named 'FitnessTypeCodeEnum' to categorize sports. Enum values allowed should be: P, B, C, ME, S, BCME, BMES. Add a column Finess to the Sports table of type FitnessTypeCodeEnum. Update the table to add the Fitness for each sport.

Include in your answer: enum creation command; a sample update statement and your new table. Include your **typed** query and the result in your answer and a **screenshot** of your results.

Note: Count Volleyball as P, Football, Basketball, Boxing as BCME, and weight lifting as BMES.

(**P**: Performance related, **B:** Body composition, **C:** Cardiorespiratory, **ME:** Muscular Endurance, **S:** Muscular Strength, **BCME:** Body composition, Cardiorespiratory, Muscular Endurance, **BMES:** Body composition, Muscular Endurance, Muscular Strength)

Postgres enum documentation: https://www.postgresql.org/docs/14/datatype-enum.html

Answer: -

CREATE TYPE FitnessTypeCodeEnum as ENUM('P','B','C','ME','S','BMES');

Alter table sports add Fitness FitnessTypeCodeEnum;

select * from sports;

OpenSSH SSH client

```
spr2022adb35=> CREATE TYPE FitnessTypeCodeEnum as ENUM('P','B','C','ME','S','BMES');
CREATE TYPE
spr2022adb35=> Alter table sports add Fitness FitnessTypeCodeEnum;
ALTER TABLE
spr2022adb35=> select * from sports;
 id |      name      |    type    | mets | difficultyrank | fitness
----+----------------+------------+------+----------------+---------
  1 | Volleyball     | team       |    4 |             20 |
  2 | Football       | team       |    8 |             10 |
  3 | Basketball     | team       |    6 |              4 |
  4 | Weight lifting | individual |    6 |             44 |
  5 | Boxing         | individual |    7 |              1 |
(5 rows)

spr2022adb35=>
```

Q16: (15 pts). Find the name and mets value of the sport(s) with the highest Metabolic Equivalents (METs) value in each fitness category.

Include your **typed** query and the result in your answer and a **screenshot** of your results.

Answer: -

SELECT name, max(METs) FROM sports GROUP BY Fitness,sports.name;

OpenSSH SSH client

```
spr2022adb35=> SELECT name, max(METs) FROM sports GROUP BY Fitness,sports.name;
      name      | max
----------------+-----
 Basketball     |   6
 Volleyball     |   4
 Weight lifting |   6
 Boxing         |   7
 Football       |   8
(5 rows)
```

Q17: (10 pts) The table/schema below representing a list of teams is not normalized. Please normalize the table/schema.

In your answer, include table names and attribute names as in the first line below. Underline keys and indicate any foreign keys you would create.

Team(id, name, coachId, coachName)

id	name	CoachId	CoachName
1	LFC	20	Jurgen Klopp
2	UTD	21	Ralf Rangnick
3	LFC	25	Jurgen Klopp

Answer: -

We can convert this given table into a normal form as given: -
As we know that id is the primary key for the Team table and there is no foreign key given in this particular scenario,
Therefore, we will add a foreign key to the table:-

Team(id, name, coachID, coachName, players)
Here, players is referenced as a foreign key :- Team.players references players.id