Q1. Perform the different JOIN operations on the tables created for your Application

domain.

Specification:

a. The Operations should be performed on tables with logical sense and mention

valid justification

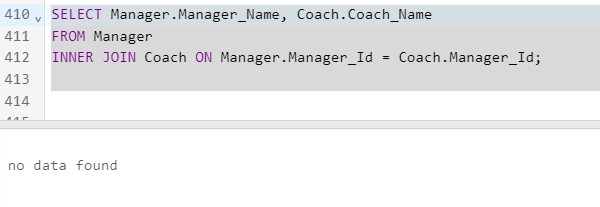
b. At least 5 Queries should be demonstrated to join 2 Tables, 3 Tables and 4 Tables.

c. All the types of Joins should be demonstrated (Cross Join, Inner Join, Left outerJoin, Right Outer Join and Self Join)

d. Write a SQL query using Outer Join which returns the records from minimum of 3Tables.

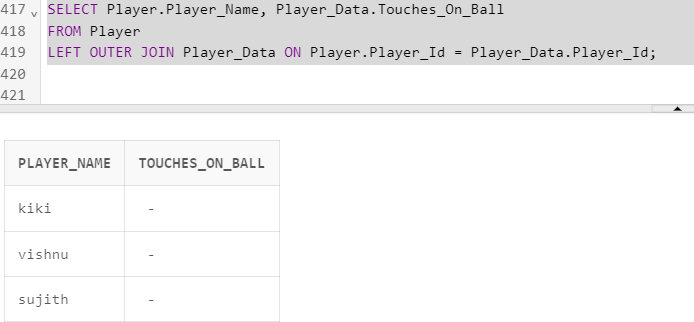
* INNER JOIN: Returns only the rows that have matching values in both tables.
* LEFT OUTER JOIN (or LEFT JOIN): Returns all the rows from the left table and the matching rows from the right table. If there's no match, NULL values are returned for the right table's columns.
* RIGHT OUTER JOIN (or RIGHT JOIN): Similar to LEFT JOIN, but returns all the rows from the right table and the matching rows from the left table. If there's no match, NULL values are returned for the left table's columns.
* FULL OUTER JOIN: Returns all rows when there is a match in either the left or right table. If there's no match, NULL values are returned for the columns of the table without a match.
* SELF JOIN: When a table is joined with itself, usually with the help of aliases, to create a relationship between rows within the same table.
* CROSS JOIN: Returns the Cartesian product of two tables, resulting in all possible combinations of rows from both tables.

1. Inner Join between Manager and Coach



Justification: The query retrieves the names of managers and their corresponding coaches using an inner join.

2. Left Outer Join between Player and Player\_Data



justification: This query shows the player names and the number of touches on the ball, including players who may not have player data

3. Right Outer Join between Assistant\_coach and Gym\_Trainer

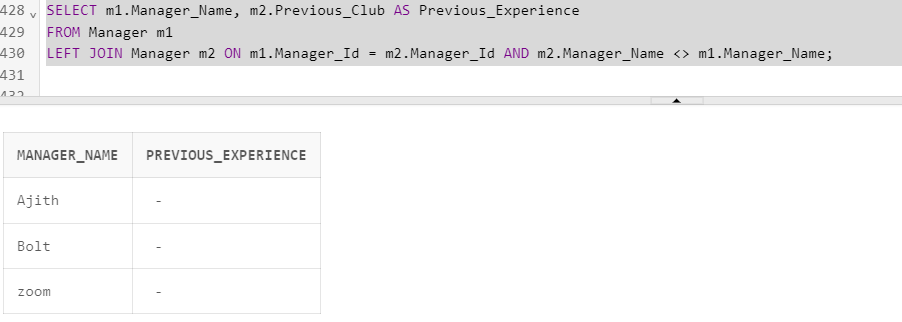
SELECT Assistant\_coach.Assist\_Coach\_Name, Gym\_Trainer.Gym\_Trainer\_Name

FROM Assistant\_coach

RIGHT OUTER JOIN Gym\_Trainer ON Assistant\_coach.Assist\_Coach\_Id = Gym\_Trainer.Coach\_Id;

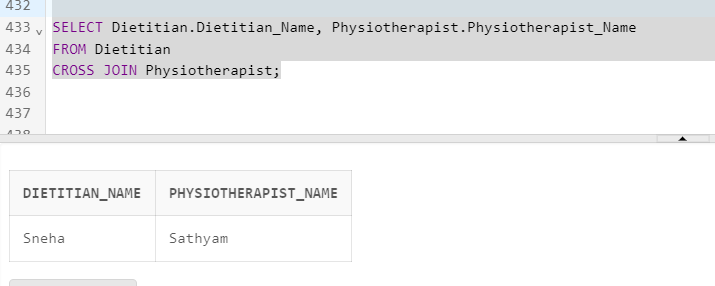
Justification: This query displays the names of assistant coaches and gym trainers, including gym trainers who may not have an assistant coach..

4. Self Join on Manager table to find Managers with Previous Experience



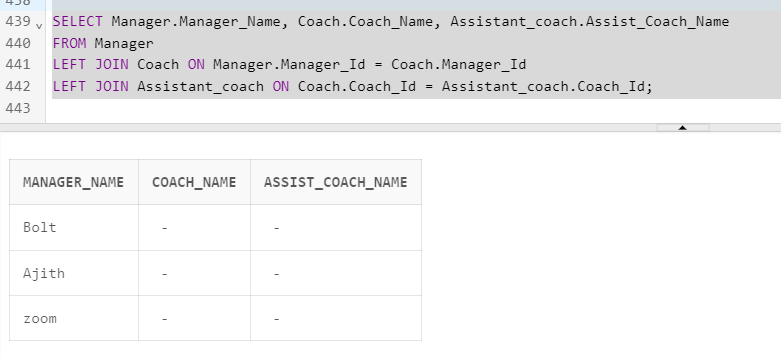
Justification: This query finds managers and their previous experience by self-joining the Manager table.

5. Cross Join between Dietitian and Physiotherapist



Justification: This query performs a cross join between dietitians and physiotherapists, creating combinations of all dietitians and all physiotherapists.

6. Outer Join involving Manager, Coach, and Assistant\_coach (at least 3 tables)



Justification: This query uses two outer joins to show the names of managers, coaches, and assistant coaches, including cases where any of these roles might be missing.Justification: This query uses two outer joins to show the names of managers, coaches, and assistant coaches, including cases where any of these roles might be missing.

e. Mention the difference between Join and Set operations used in SQL.

Difference between JOIN and Set operations in SQL:

JOIN Operations: JOIN operations are used to combine rows from two or more tables based on a related column between them. JOINs are used to retrieve data from multiple tables by linking rows in one table with corresponding rows in another. The result of a JOIN operation includes columns from all joined tables.

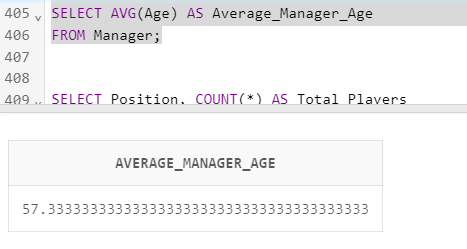
Set Operations: Set operations, such as UNION, INTERSECT, and EXCEPT/MINUS, involve combining the results of two or more SELECT queries. These operations are applied to the result sets of individual queries and don't necessarily require a related column. Set operations help merge, intersect, or exclude rows from multiple result sets. In summary, JOIN operations focus on combining rows from tables based on matching columns, while set operations combine rows from the results of separate SELECT queries.

Q2. Write at least 5 compound SQL statements to demonstrate at least any 5 of thefollowing Aggregate functions on the tables created for your application domain (AVG,SUM, COUNT, MIN, MAX, MEDIAN, CORR and STDDEV) and appropriately use

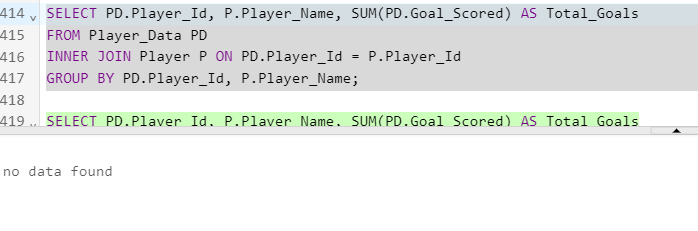
GROUPBY and HAVING clause

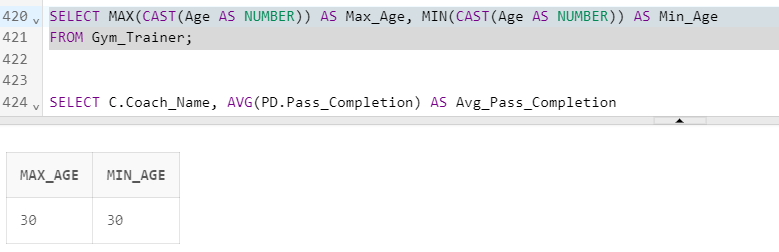
Specification:

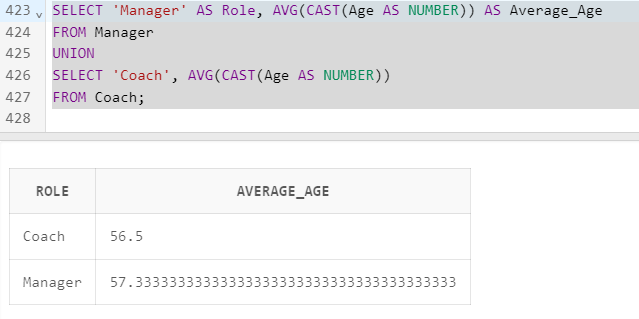
1. The operations should be performed on tables with logical sense and mention valid justification.





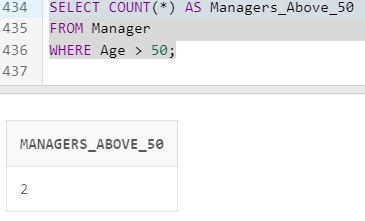




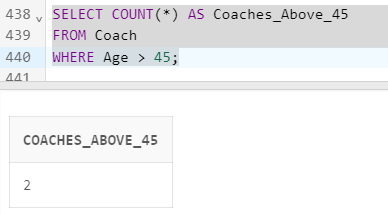


1. At least 1 Compound SQL Query with a suitable aggregate function should be demonstrated for all the Tables.

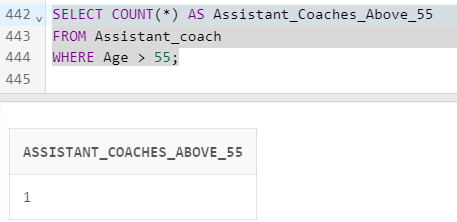
Manager Table - Count the number of managers with an age greater than 50:



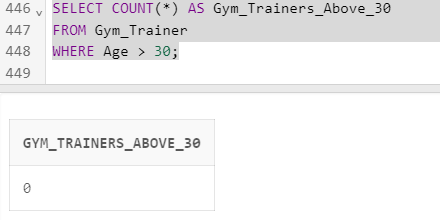
Coach Table - Count the number of coaches with an age greater than 45



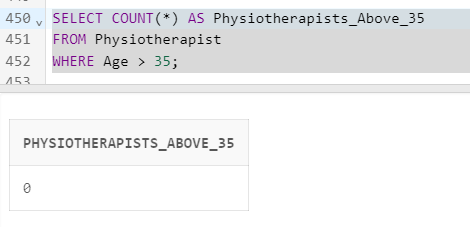
Assistant\_coach Table - Count the number of assistant coaches with an age greater than 55:



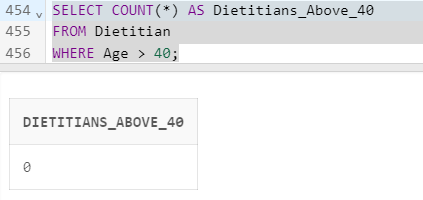
Gym\_Trainer Table - Count the number of gym trainers with an age greater than 30:



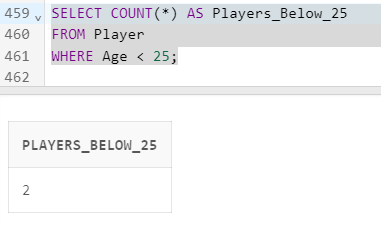
Physiotherapist Table - Count the number of physiotherapists with an age greater than 35:



Dietitian Table - Count the number of dietitians with an age greater than 40



Player Table - Count the number of players with an age less than 25:



create table Football (

Admin\_Id varchar2(20) constraint Admin\_Id\_pk primary key,

Admin\_Name varchar2(50)not null,

Manager\_Name varchar2(50) not null,

Manager\_Id varchar2(50) not null,

Player\_Id varchar2(30)not null,

Game\_Data varchar2(50)not null);

create table Manager (

Manager\_Name varchar2(50) NOT NULL,

Manager\_Id varchar2(30) constraint Manager\_Id\_pk primary key,

Address varchar2(80)NOT NULL,

Email varchar2(60)NOT NULL,

contact\_No varchar2(50)NOT NULL,

Age varchar2(30)NOT NULL,

Achievements varchar2(100)NOT NULL,

Previous\_Club varchar2(80)NOT NULL,

Admin\_Id varchar2(20) references Football( Admin\_Id));

create table Coach (

Coach\_Name varchar2(50) NOT NULL,

Coach\_Id varchar2(30) constraint Coach\_Id\_pk primary key,

Address varchar2(80)NOT NULL,

Email varchar2(60)NOT NULL,

Contact\_No varchar2(50)NOT NULL,

Age varchar2(30)NOT NULL,

Achievements varchar2(100)NOT NULL,

Previous\_Club varchar2(80)NOT NULL,

Manager\_Id varchar2(20) references Manager( Manager\_Id));

create table Assistant\_coach (

Assist\_Coach\_Name varchar2(50)NOT NULL,

Assist\_Coach\_Id varchar2(30) constraint Assistant\_coach\_Id\_pk primary key,

Address varchar2(80)NOT NULL,

Email varchar2(60)NOT NULL,

Contact\_No varchar2(50)NOT NULL,

Age varchar2(30)NOT NULL,

Achievements varchar2(100)NOT NULL,

Previous\_Club varchar2(80)NOT NULL,

Coach\_Id varchar2(20) references Coach( Coach\_Id));

create table Gym\_Trainer (

Gym\_Trainer\_Name varchar2(50) not null,

Gym\_Trainer\_Id varchar2(30) constraint Gym\_Trainer\_pk primary key,

Address varchar2(80),

Email varchar2(60),

Contact\_No varchar2(50),

Age varchar2(30),

Special\_Workout\_Player varchar2(100),

Previous\_Club varchar2(80));

create table Physiotherapist (

Physiotherapist\_Name varchar2(50)NOT NULL,

Physiotherapist\_Id varchar2(30) constraint Physiotherapist\_Id\_pk primary key,

Address varchar2(80)NOT NULL,

Email varchar2(60)NOT NULL,

Contact\_No varchar2(50)NOT NULL,

Age varchar2(30)NOT NULL,

Special\_Exercise\_Player varchar2(100)NOT NULL,

Previous\_Club varchar2(80)NOT NULL,

Assist\_Coach\_Id varchar2(20) references Assistant\_coach(Assist\_Coach\_Id));

create table Dietitian (

Dietitian\_Name varchar2(50)NOT NULL,

Dietitian\_Id varchar2(30) constraint Dietitian\_Id\_pk primary key,

Address varchar2(80)NOT NULL,

Email varchar2(60)NOT NULL,

Contact\_No varchar2(50)NOT NULL,

Age varchar2(30)NOT NULL,

Special\_Diet\_player varchar2(100)NOT NULL,

Previous\_Club varchar2(80)NOT NULL,

Assist\_Coach\_Id varchar2(20) references Assistant\_coach(Assist\_Coach\_Id));

create table Game\_Data (

Game\_Id varchar2(30) constraint Game\_Id\_pk primary key,

Ball\_Possession varchar2(30)NOT NULL,

Keyy\_Chances varchar2(30)NOT NULL,

Goal\_Scored varchar2(20)NOT NULL,

Pass\_Completion varchar2(20)NOT NULL,

Interception varchar2(30)NOT NULL,

Assists varchar2(30)NOT NULL,

Yellow\_Card varchar2(30)NOT NULL,

Break\_The\_Defence varchar2(30)NOT NULL,

Admin\_Id varchar2(20) references Football( Admin\_Id));

create table Player (

Player\_Name varchar2(50)NOT NULL,

Player\_Id varchar2(30) constraint Player\_Id\_pk primary key,

Address varchar2(80)NOT NULL,

Email varchar2(60)NOT NULL,

Contact\_No varchar2(50)NOT NULL,

Age varchar2(30)NOT NULL,

Position varchar2(80)NOT NULL,

Admin\_Id varchar2(20) references Football( Admin\_Id));

create table Player\_Data (

Player\_Data\_Id varchar2(30) constraint Player\_Data\_Id\_pk primary key,

Health\_Details varchar2(50)NOT NULL,

Touches\_On\_Ball varchar2(50)NOT NULL,

Pass\_Completion varchar2(50)NOT NULL,

Pass\_Received varchar2(50)NOT NULL,

Pass\_Missed varchar2 (50)NOT NULL,

Goal\_Scored varchar2 (50)NOT NULL,

Key\_Pass varchar2 (50)NOT NULL,

Interceptions varchar2(50)NOT NULL,

Play\_Time varchar2(50)NOT NULL,

Position\_Of\_Player varchar2(50) NOT NULL,

Player\_Id varchar2(20) references Player( Player\_Id)

);

"Insert into the first table football"

insert into Football(

Admin\_Id,

Admin\_Name,

Manager\_Name,

Manager\_Id,

Player\_Id,

Game\_Data)values ('008','Reeve','Ajith','Man001','Pla001','Gd001');

"Insert into the first table Manager"

insert into Manager(

Manager\_Name,

Manager\_Id,

Address,

Email,contact\_No,

Age,

Achievements,

Previous\_Club)

values ('Ajith','Man004','abc','xyz@gmail.com','1234567899','50','premier league','mancity');

insert into Manager(

Manager\_Name,

Manager\_Id,

Address,

Email,contact\_No,

Age,

Achievements,

Previous\_Club)

values ('Bolt','Man002','agbc','xyzz@gmail.com','1234567599','54','ligue1 league','man united');

insert into Manager(

Manager\_Name,

Manager\_Id,

Address,

Email,contact\_No,

Age,

Achievements,

Previous\_Club)

values ('zoom','Man003','bbc','yyz@gmail.com','1234567898','68','champions league','mancity');

"Insert into the first table Coach"

insert into Coach(

Coach\_Name,

Coach\_Id,

Address,

Email,

contact\_No,

Age,

Achievements

,Previous\_Club)

values ('Senthil','Coa003','abc','xyz@gmail.com','1234567899','58','Isl league',' city');

insert into Coach(

Coach\_Name,

Coach\_Id,

Address,

Email,

contact\_No,

Age,

Achievements

,Previous\_Club)

values ('Nadhan\_sir','Coa002','abc','xyzzz@gmail.com','1234547899','55','Isl league',' gg city');

"Insert into the first table Assistant coach"

insert into Assistant\_coach(

Assist\_coach\_Name,

Assist\_coach\_Id,

Address,

Email,

contact\_No,

Age,

Achievements,

Previous\_Club)

values ('Neha','Acoa004','abc','xyz@gmail.com','1234567899','60','Christ league','jain');

insert into Assistant\_coach(

Assist\_coach\_Name,

Assist\_coach\_Id,

Address,

Email,

contact\_No,

Age,

Achievements,

Previous\_Club)

values ('Aleena','Acoa002','abccc','xysz@gmail.com','1234563899','40','tow league','savior');

"Insert into the first table Gym Trainer"

insert into Gym\_Trainer(

Gym\_Trainer\_Name,

Gym\_Trainer\_Id,

Address,

Email,

contact\_No,

Age,

Special\_Workout\_Player,

Previous\_Club)

values ('Christina','Gt002','abc','xyz@gmail.com','1234567899','30','stregthening','mancity');

"Insert into the first table Physiotherapist"

insert into physiotherapist (

physiotherapist\_Name,

physiotherapist\_Id,

Address,

Email,

contact\_No,

Age,

Special\_Exercise\_Player,

Previous\_Club)

values ('Sathyam','Phy001','abc','xyz@gmail.com','1234567899','30','Condition','mancity');

"Insert into the first table Dietitian"

insert into Dietitian (

Dietitian\_Name,

Dietitian\_Id,

Address,

Email,

contact\_No,Age,

Special\_Diet\_Player,

Previous\_Club)

values ('Sneha','Die002','abc','xyz@gmail.com','1234567899','40','Renal','mancity');

"Insert into the first table Game data"

insert into Game\_Data

( Game\_Id,

Ball\_Possession,

Keyy\_Chances,

Goal\_Scored,

Pass\_Completion,

Interception,Assists,Yellow\_Card,Break\_The\_Defence)

values ('Pl01','10','5','4','7','8','10','10','8');

"Insert into the first table Player"

insert into Player(Player\_Name,

Player\_Id,

Address,

Email,

contact\_No,

Age,

position)

values ('sujith','pl05','abc','xxyz@gmail.com','1234567899','20','RW');

insert into Player(Player\_Name,

Player\_Id,

Address,

Email,

contact\_No,

Age,

position)

values ('vishnu','pl002','abec','xtyz@gmail.com','1234277899','20','lW');

insert into Player(Player\_Name,

Player\_Id,

Address,

Email,

contact\_No,

Age,

position)

values ('kiki','pl003','abc','xysz@gmail.com','1234544899','27','cb');

"Insert into the first table Player\_Data"

insert into Player\_Data (

Player\_Data\_Id,

Health\_Details,

Touches\_On\_Ball,

Pass\_Completion,

Pass\_Received,

Pass\_Missed,

Goal\_Scored,

Key\_Pass,

Interceptions,

Play\_Time,

Position\_Of\_Player)

values ('Pld001','fit','5','8','18','5','2','7','4','12','RW');

insert into Player\_Data (

Player\_Data\_Id,

Health\_Details,

Touches\_On\_Ball,

Pass\_Completion,

Pass\_Received,

Pass\_Missed,

Goal\_Scored,

Key\_Pass,

Interceptions,

Play\_Time,

Position\_Of\_Player)

values ('Pld002','fit','6','5','19','5','2','8','4','19','RW');

insert into Player\_Data (

Player\_Data\_Id,

Health\_Details,

Touches\_On\_Ball,

Pass\_Completion,

Pass\_Received,

Pass\_Missed,

Goal\_Scored,

Key\_Pass,

Interceptions,

Play\_Time,

Position\_Of\_Player)

values ('Pld003','un\_fit','8','8','18','12','1','9','14','15','RW');

select\*from Football;

select \*from Manager;

select\*from Coach;

select\*from Assistant\_coach;

select\*from Gym\_Trainer;

select\*from Physiotherapist;

select\*from Dietitian;

select\*from Game\_Data;

select\*from Player;

select\*from Player\_Data;

'join'

SELECT Manager.Manager\_Name, Coach.Coach\_Name

FROM Manager

INNER JOIN Coach ON Manager.Manager\_Id = Coach.Manager\_Id;

SELECT Player.Player\_Name, Player\_Data.Touches\_On\_Ball

FROM Player

LEFT OUTER JOIN Player\_Data ON Player.Player\_Id = Player\_Data.Player\_Id;

SELECT Assistant\_coach.Assist\_Coach\_Name, Gym\_Trainer.Gym\_Trainer\_Name

FROM Assistant\_coach

RIGHT OUTER JOIN Gym\_Trainer ON Assistant\_coach.Assist\_Coach\_Id = Gym\_Trainer.Coach\_Id;

SELECT m1.Manager\_Name, m2.Previous\_Club AS Previous\_Experience

FROM Manager m1

LEFT JOIN Manager m2 ON m1.Manager\_Id = m2.Manager\_Id AND m2.Manager\_Name <> m1.Manager\_Name;

SELECT Dietitian.Dietitian\_Name, Physiotherapist.Physiotherapist\_Name

FROM Dietitian

CROSS JOIN Physiotherapist;

SELECT Manager.Manager\_Name, Coach.Coach\_Name, Assistant\_coach.Assist\_Coach\_Name

FROM Manager

LEFT JOIN Coach ON Manager.Manager\_Id = Coach.Manager\_Id

LEFT JOIN Assistant\_coach ON Coach.Coach\_Id = Assistant\_coach.Coach\_Id;

'avg count'

SELECT AVG(Age) AS Average\_Manager\_Age

FROM Manager;

SELECT Position, COUNT(\*) AS Total\_Players

FROM Player

GROUP BY Position;

SELECT PD.Player\_Id, P.Player\_Name, SUM(PD.Goal\_Scored) AS Total\_Goals

FROM Player\_Data PD

INNER JOIN Player P ON PD.Player\_Id = P.Player\_Id

GROUP BY PD.Player\_Id, P.Player\_Name;

SELECT MAX(CAST(Age AS NUMBER)) AS Max\_Age, MIN(CAST(Age AS NUMBER)) AS Min\_Age

FROM Gym\_Trainer;

SELECT 'Manager' AS Role, AVG(CAST(Age AS NUMBER)) AS Average\_Age

FROM Manager

UNION

SELECT 'Coach', AVG(CAST(Age AS NUMBER))

FROM Coach;

"compound sql"

SELECT COUNT(\*) AS Managers\_Above\_50

FROM Manager

WHERE Age > 50;

SELECT COUNT(\*) AS Coaches\_Above\_45

FROM Coach

WHERE Age > 45;

SELECT COUNT(\*) AS Assistant\_Coaches\_Above\_55

FROM Assistant\_coach

WHERE Age > 55;

SELECT COUNT(\*) AS Gym\_Trainers\_Above\_30

FROM Gym\_Trainer

WHERE Age > 30;

SELECT COUNT(\*) AS Physiotherapists\_Above\_35

FROM Physiotherapist

WHERE Age > 35;

SELECT COUNT(\*) AS Dietitians\_Above\_40

FROM Dietitian

WHERE Age > 40;

SELECT COUNT(\*) AS Players\_Below\_25

FROM Player

WHERE Age < 25;