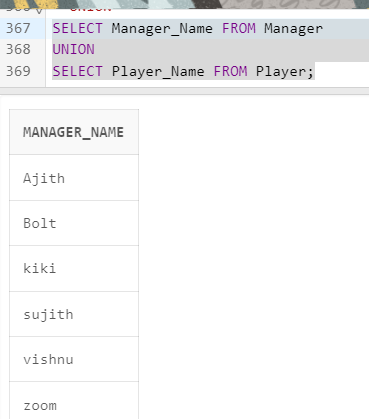
Q1. Write compound SQL statements to demonstrate the following SET operations on the

tables created for your Application domain.

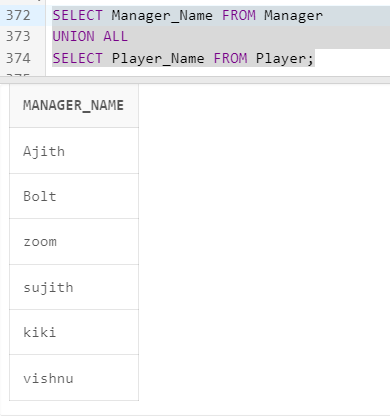
UNION, UNION ALL, INTERSECT and MINUS

Specifications:



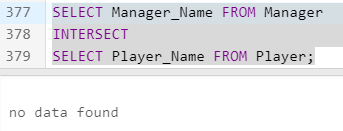
Union: The union operation combines the result table of two or more select queries into a single table, removing duplicate rows, in this case table between manager and player.

So that the administrator can see all the managers and players in the program.



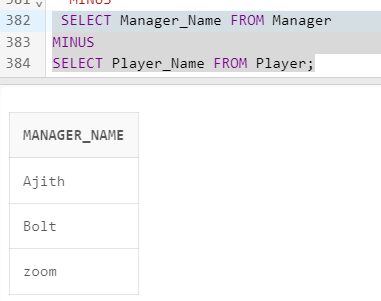
Union all: this operations combines the result set of two or more select queries into a single result set, it doesn’t remove duplicates.

In this case we performing the union all from manager and player.



Intersect: This operation retrieves the common rows that exist in the result of two or more select queries.

This case from manager and player as there are no common values it shows no data.



Minus: it shows the rows from the first result set that are not present it the second result set.

Here in between manager and players

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

valid justification.

Union: The union operation combines the result table of two or more select queries into a single table, removing duplicate rows, in this case table between manager and player.

So that the administrator can see all the managers and players in the program.

In this case we performing the union all from manager and player.

Union all: this operations combines the result set of two or more select queries into a single result set, it doesn’t remove duplicates.

Intersect: This operation retrieves the common rows that exist in the result of two or more select queries.

This case from manager and player as there are no common values it shows no data.

Minus: it shows the rows from the first result set that are not present it the second result set.

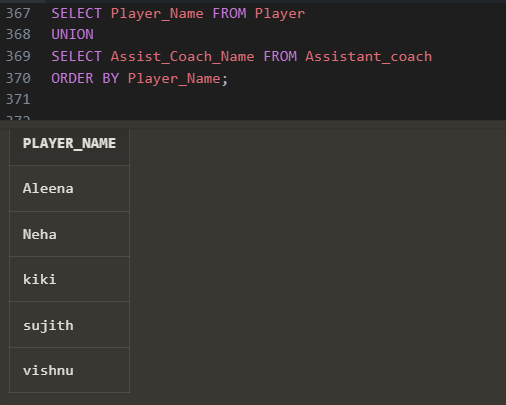
Here in between manager and players

b. What is the difference between UNION and UNION ALL?

Union: it gives a result with unique rows, and the duplicates will be removed .

Union all: includes all rows from both result sets, including duplicates.

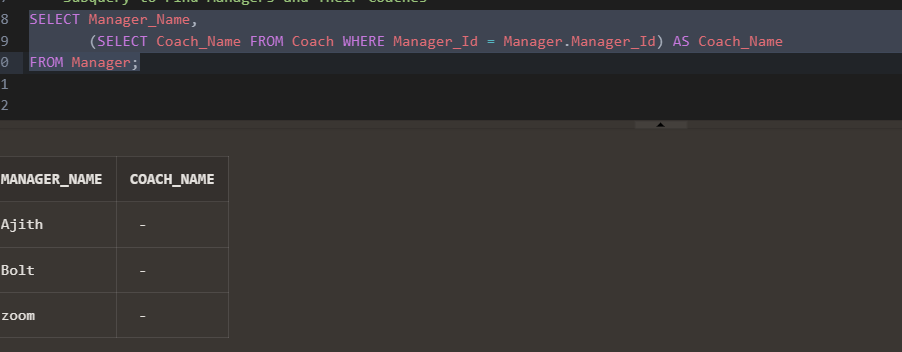
c. Show how to sort the result set of a compound query which performs SET operations.

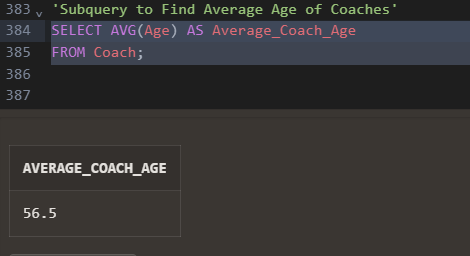


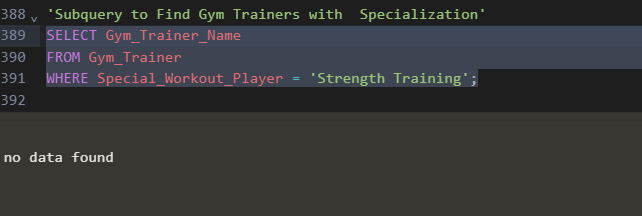
Q2. Demonstrate the usage of Sub queries to implement the operations to be performedon the tables created for your Application domain.

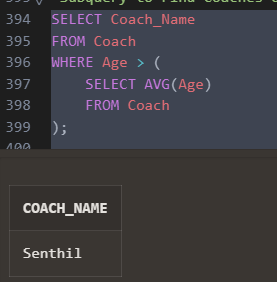
Specification:

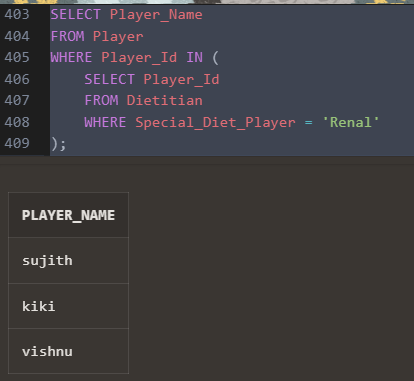
1. At least 5 Queries should be demonstrated.

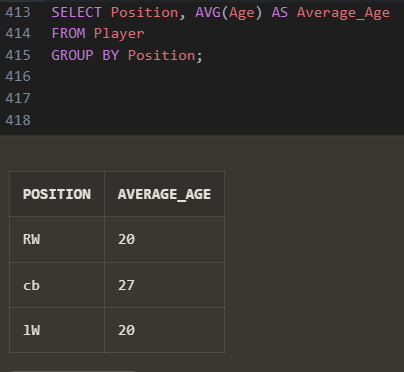












1. All tables should be used in at least 1 Compound SQL Query to perform a specified operation.

SELECT Player\_Name

FROM Player

WHERE Manager\_Id IN (

SELECT Coach.Manager\_Id

FROM Coach

WHERE Previous\_Club = 'city'

);

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

with this query we can connect the player table to the manager table which are very distinctly related to each other.

d. Mention the types of subqueries and its differences.

Single-row subquery:

1. Returns a single value

* Used with comparison operators
* The outer query expects a single value from the subquery
* Eg; finding employees with a salary greater than the average salary.

1. Multiple-row subquery

* Returns multiple rows
* Used with operators like in, any or all
* Compares a value with a set of values from the subquery.
* Eg; finding products that have been ordered more than once.

1. Correlated subquery

* Returns values for each row of the outer query.
* References colums from the outer query within the subquery
* Typically used with exits or not exists clauses
* Eg; checking if a manager has employees in a certain age range

1. Nested subquery

* Contains a subquery inside another subquery
* Can be of any type single row, multiple row, correlated
* Allows for complex filtering and calculations
* Eg; finding managers with employees whose average salary is above the company's average salary

Differences

* Single-row returns a single value
* Multiple rows returns multiple values
* Correlated returns values for each row of the outer query
* Nested contains a subquery inside another subquery

Usage

* Single-row comparing a value with a single value
* Multiple rows comparing a value with multiple values
* Correlated checking conditions for each row using values from the outer query
* Nested performing complex filtering or calculations involving subqueries

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;

-- Retrieve a list of unique names from Managers and players

SELECT Manager\_Name FROM Manager

UNION

SELECT Player\_name FROM Player;

-- Retrieve a list of all names from Managers and Player, including duplicates

SELECT Manager\_Name FROM Manager

UNION ALL

SELECT Player\_name FROM Player;

-- Retrieve Manager names who are also Player

SELECT Manager\_Name FROM Manager

INTERSECT

SELECT Player\_name FROM Player;

-- Retrieve Manager names who are not Player

SELECT Manager\_Name FROM Manager

MINUS

SELECT Player\_name FROM Player;

SELECT Player\_Name FROM Player

UNION

SELECT Assist\_Coach\_Name FROM Assistant\_coach

ORDER BY Player\_Name;

SELECT Manager\_Name,

(SELECT Coach\_Name FROM Coach WHERE Manager\_Id = Manager.Manager\_Id) AS Coach\_Name

FROM Manager;

SELECT AVG(Age) AS Average\_Coach\_Age

FROM Coach;

SELECT Gym\_Trainer\_Name

FROM Gym\_Trainer

WHERE Previous\_Club IS NULL;

SELECT Coach\_Name

FROM Coach

WHERE Age > (

SELECT AVG(Age)

FROM Coach

);

SELECT Player\_Name

FROM Player

WHERE Player\_Id IN (

SELECT Player\_Id

FROM Dietitian

WHERE Special\_Diet\_Player = 'Renal'

);

SELECT Position, AVG(Age) AS Average\_Age

FROM Player

GROUP BY Position;

SELECT DISTINCT M.Manager\_Name, AC.Assist\_Coach\_Name

FROM Manager M

JOIN Coach C ON M.Manager\_Id = C.Manager\_Id

JOIN Assistant\_coach AC ON C.Coach\_Id = AC.Coach\_Id

JOIN Player P ON C.Manager\_Id = P.Manager\_Id

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

WHERE PD.Health\_Details = 'fit';

SELECT Player\_Name

FROM Player

WHERE Manager\_Id IN (

SELECT Coach.Manager\_Id

FROM Coach

WHERE Previous\_Club = 'city'

);