CHECK_UP TABLE INSERT:

```
insert into Check-up
values ('2019-12-09', 100, 180, 45, 98, 'None', 22, 19),
('2020-11-06', 120, 170, 40, 99, 'ACL Tear', 21, 11),
('2020-10-29', , 110, 165, 30, 95, 'None', 22, 19),
('2021-03-15', 115, 185, 50, 96, 'None', 21, 16),
('2018-11-09', 100, 180, 45, 97, 'None', 25, 18),
('2021-12-18', 120, 177, 40, 99, 'ACL Tear', 21, 11),
('2020-03-29', , 114, 167, 37, 95, 'PCL tear', 22, 16),
('2020-05-12', 101, 187, 45, 98, 'None', 22, 19),
('2019-05-19', 117, 189, 42, 96, 'None', 29, 13),
('2019-06-08', 113, 165, 40, 99, 'ACL Tear', 21, 11)
```

UPDATE CHECK_UP:

```
set d_comment= '%s'
where player_id =
```

GOES_FOR TABLE INSERT:

```
insert into goes_for
values ('2019-12-09', 170),
('2020-11-06', 131),
('2020-10-29',33),
('2021-03-15, 12),
('2018-11-09', 333),
('2021-12-18', 90),
('2020-03-29',1126),
('2020-05-12', 543),
('2019-05-19', 2000),
('2019-06-08', 337)
```

SCHEDULE_DATA TABLE INSERT:

```
insert into schedule
values (200, '2021-12-14','Tuesday','4pm',2, 'Training_1'),
(201, '2021-12-16','Thursday','8am',3, 'Fitness'),
(202, '2021-12-16','Thursday','6pm',2, 'Practice_game'),
(206, '2021-12-17','Friday',6am,2, 'Gym')
```

UPDATES TABLE INSERT:

```
insert into updates
values (202, 5) ,(200,4), (201,6), (206,9)
```

DISPLAY PLAYER PROFILE:

```
Select * from player
where player_id = %s
```

DELETE PLAYER:

```
delete from player
where player_id = %s
```

STORED PROCEDURES

1. Get overall average for a player in all matches

```
delimiter $$
create procedure get_player_average(in in_player_id int)
begin
select player_id, avg(points_scored) as avg_points,
avg(rebounds) as avg_rebouds,
avg(assists) as avg_assists, avg(steals) as avg_steals,
avg(blocked_shots) as avg_blocked_shots,
avg(personal_fouls) as avg_fouls
from has_match_stats
group by player_id;
end $$
delimiter ;
```

2. Get overall team average in a year vs some team

```
delimiter $$
create procedure get_team_average(in year int, in opponent_team varchar(1000) )
begin
select year,opponent_team, avg(home_wins) as avg_home_wins
from team as t natural join plays as p natural join matches as m
where t.year = year and m.opponent_team = opponent_team
group by year, opponent_team;
end $$
delimiter ;
```

3. Top 3 scores in every match

```
delimiter $$
create procedure top_3_in_every_match(in match_id int )
begin
select player_id, player_name
from has_match_stats as hm natural join player as p
where hm.match_id = match_id
order by desc
limit 3
end $$
delimiter ;
```

4. Show stats using player_id and match_id

```
delimiter $$
create procedure stats_in_match_by_player(in match_id int, in player_id int)
begin
select *
from has_match_stats as hm
where hm.match_id = match_id and hm.player_id = player_id
end $$
delimiter ;
```

5. Show health data to dietician using player_id

```
delimiter $$
create procedure health_data_dietician(in player_id int)
begin
select p.player_name,check_up_date,bp,cholestrol,bpm,oxygen
from check_up as cp natural join goes_for as gf natural join player as p
where gf.player_id = player_id
end $$
delimiter ;
```

VIEWS

1. View for the manager to see list of players with expiring contracts in the current year

```
CREATE or REPLACE VIEW Contracts_expiring AS
SELECT p.player_id, p.player_name, c.fa_year, c.salary, c.aav
FROM players as p natural join contract as c
WHERE Year(curdate()) = c.end_year
```

2. View for the coach to see a list of all players in the current team

```
CREATE or REPLACE VIEW list_of_current_players AS
SELECT player_id, player_name, team_id, year
FROM players natural join teams
WHERE Year(curdate()) = year
```

3. View for physiotherapist to see list of injured players

```
CREATE or REPLACE VIEW list_of_injured_players AS
SELECT player_id, player_name, status, injuries
FROM players natural join goes_for natural join Check_up
WHERE status = 'injured'
```

4. View for current day schedule

```
create or replace view today_schedule as
select *
from schedule
where practice_date = curdate()
```

5.View for valid schedule

```
create or replace view valid_schedule as
select *
from schedule
where practice_date > curdate()
```

FUNCTIONS

1. Total number of matches played by a player

```
delimiter $$
create function total_matches_played(player_id int)
returns int
deterministic
begin
declare matches_played int;
select count(*) into matches_played
from has_match_stats as hms
where hms.player_id = player_id;


return matches_played;
end $$
delimiter ;
```

2. Total fouls committed in career

```
delimiter $$
create function total_fouls_player(player_id int)
returns int
deterministic
begin
declare total_fouls int;
select sum(personal_fouls) into total_fouls
from has_match_stats as hms
where hms.player_id = player_id;


return total_fouls;
end $$
delimiter ;
```

3.Total points scored in career

```
delimiter $$
create function total_points_scored(player_id int)
returns int
deterministic
begin
declare total_points int;
select sum(points_scored) into total_points
from has_match_stats as hms
where hms.player_id = player_id;

return total_points;
end $$
delimiter ;
```

TRANSACTION
1.If physiotherapist comments the player is

then change player status to active

```
create procedure update_player_status (player_id_ int)
begin
declare exit handler for sqlexception rollback;


start transaction;


update Contract
set approval_status = "Yes"
where player_id = player_id_ ;


update player
set status = "Active"
where player_id =player_id _ ;


commit;
end
```