

## Triggers

--Booking

```
CREATE TABLE BOOK_trig_table(  
Match_no INT NOT NULL,  
player_id INT NOT NULL,  
playschedule GS NOT NULL,  
booking_time INT NOT NULL,  
booking_type BT NOT NULL  
);
```

Create or Replace Function BOOK\_trig\_func() Returns Trigger as \$BOOK\_trig\_table\$  
Declare

```
    playerin PLAYER%rowtype;  
    playerout PLAYER%rowtype;  
    pl PLAYER%rowtype;  
    m MATCH%rowtype;  
    b booking%rowtype;
```

Begin

```
    for pl in select* from PLAYER  
    LOOP
```

```
        if(new.PLAYER_ID_IN = pl.PLAYER_ID) then  
            playerin = pl;
```

```
        End if;
```

```
        if(new.PLAYER_ID_OUT = pl.PLAYER_ID) then  
            playerout = pl;
```

```
        End if;
```

```
    End LOOP;
```

```
    for m in select* from MATCH  
    LOOP
```

```
        if(m.MATCH_NUMBER = new.MATCH_NUMBER) then
```

```
            if(playerin.TEAM_NAME != m.HOME_TEAM) then
```

```
                if(playerin.TEAM_NAME != m.AWAY_TEAM) then
```

```
                    Insert into BOOK_trig_table
```

```
Values(New.MATCH_NUMBER,New.PLAYER_ID,NEW.PLAY_SCHEDULE,NEW.BOOKIN_TIME,New.BOOK  
ING_TYPE);
```

```
                    Return NULL;
```

```
                End if;
```

```
            End if;
```

```
            if(playerout.TEAM_NAME != m.HOME_TEAM) then
```

```
                if(playerout.TEAM_NAME != m.AWAY_TEAM) then
```

```
                    Insert into BOOK_trig_table
```

```
Values(New.MATCH_NUMBER,New.PLAYER_ID,NEW.PLAY_SCHEDULE,NEW.BOOKIN_TIME,New.BOOK  
ING_TYPE);
```

```
                    Return NULL;
```

```
                End if;
```

```
            End if;
```

```
            for b in SELECT* from BOOKING  
            LOOP
```

```
                if(b.PLAYER_ID = m.PLAYER_ID) then
```

```
                    if(b.BOOKING_TYPE = 'RED') then
```

```
                        Insert into BOOK_trig_table
```

```
Values(New.MATCH_NUMBER,New.PLAYER_ID,NEW.PLAY_SCHEDULE,NEW.BOOKIN_TIME,New.BOOK  
ING_TYPE);
```

```

                                Triggers
                                Return NULL;
                                end if;
                                end if;
                                end LOOP;
                                End if;
                                End loop;
RETURN NEW;
END;$BOOK_trig_table$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER BOOK_trig_table
BEFORE INSERT ON BOOKING
FOR EACH ROW EXECUTE PROCEDURE BOOK_trig_func();

```

```

--captian
CREATE TABLE CAPTAIN_trig_table(
team_name Varchar(50) NOT NULL,
player_id INT NOT NULL,
start_date varchar(10) NOT NULL,
end_date varchar(10) NOT NULL
);

```

Create or Replace Function CAPTAIN\_trig\_func() Returns Trigger as  
\$CAPTAIN\_trig\_table\$

```

Declare
    p PLAYER%rowtype;
    c CAPTAIN%rowtype;
Begin
    for p in SELECT* from PLAYER
    LOOP
        if(new.PLAYER_ID = p.PLAYER_ID) then
            if(new.TEAM_NAME != p.TEAM_NAME) then
                INSERT INTO CAPTAIN_trig_table
values(new.TEAM_NAME,new.PLAYER_ID,new.START_DATE,new.END_DATE);
                RETURN NULL;
            end if;
        end if;
    end loop;
    if('08/12/2017'>new.START_DATE) then
        INSERT INTO CAPTAIN_trig_table
values(new.TEAM_NAME,new.PLAYER_ID,new.START_DATE,new.END_DATE);
        RETURN NULL;
    end if;
    if('05/13/2018'>new.END_DATE) then
        INSERT INTO CAPTAIN_trig_table
values(new.TEAM_NAME,new.PLAYER_ID,new.START_DATE,new.END_DATE);
        RETURN NULL;
    end if;
end if;

```

```

                                Triggers
for c in SELECT * from CAPTAIN
    LOOP
        if(new.TEAM_NAME = c.TEAM_NAME) then
            c.END_DATE = new.SATRT_DATE;
        end if;
    end LOOP;
RETURN NEW;
END;$CAPTAIN_trig_table$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER CAPTAIN_trig_table
BEFORE INSERT ON CAPTAIN
FOR EACH ROW EXECUTE PROCEDURE CAPTAIN_trig_func();

```

```

--captianu
Create or Replace Function CAPTAINU_trig_func() Returns Trigger as
$CAPTAINU_trig_table$
Begin
RETURN NULL;
END;$CAPTAINU_trig_table$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER CAPTAINU_trig_table
BEFORE UPDATE ON CAPTAIN
FOR EACH ROW EXECUTE PROCEDURE CAPTAINU_trig_func();

```

```

--goal
CREATE TABLE GOAL_trig_table(
goal_id INT NOT NULL,
goal_time INT NOT NULL,
goal_type GT NOT NULL,
goal_schedule GS NOT NULL,
goal_half INT NOT NULL,
assisted_by int not null,
match_no int not null,
scored_by int not null
);

```

```

Create or Replace Function GOAL_trig_func() Returns Trigger as $GOAL_trig_table$
Declare
    p PLAYER%rowtype;
    s substitution%rowtype;
    m MATCH%rowtype;
    b booking%rowtype;
    ps PLAYER%rowtype;
    pa PLAYER%rowtype;
Begin

```

```

Triggers
for p in select* from player
LOOP
    if(p.PLAYER_ID = new.SCORED_BY) then
        ps = p;
    end if;
    if(p.PLAYER_ID = new.SCORED_BY) then
        pa = p;
    end if;
end loop;
for m in Select* from match
LOOP
    if(m.MATCH_NUMBER = new.MATCH_NUMBER) then
        if(ps.TEAM_NAME != m.HOME_TEAM) then
            if(ps.TEAM_NAME != m.AWAY_TEAM) then
                Insert into GOAL_trig_table
Values(New.GOAL_ID,New.GOAL_TIME,New.GOAL_TYPE,NEW.GOAL_SCHEDULE,NEW.GOAL_HALF,N
ew.ASSISTED_BY,New.MATCH_NUMBER,New.SCORED_BY);
                Return NULL;
            End if;
        End if;
        if(pa.TEAM_NAME != m.HOME_TEAM) then
            if(pa.TEAM_NAME != m.AWAY_TEAM) then
                Insert into GOAL_trig_table
Values(New.GOAL_ID,New.GOAL_TIME,New.GOAL_TYPE,NEW.GOAL_SCHEDULE,NEW.GOAL_HALF,N
ew.ASSISTED_BY,New.MATCH_NUMBER,New.SCORED_BY);
                Return NULL;
            End if;
        End if;
    End if;
end loop;
for s in select* from substitution
LOOP
    if(s.MATCH_NUMBER=NEW.MATCH_NUMBER) then
        if(s.PLAYER_ID_OUT = NEW.SCORED_BY) then
            if(s.TIME_IN_OUT<new.GOAL_TIME) then
                Insert into SUB_trig_table
Values(New.MATCH_NUMBER,New.PLAYER_ID_IN,New.PLAYER_ID_OUT,NEW.PLAY_SCHEDULE,NEW
.TIME_IN_OUT);
                RETURN NULL;
            End If;
        End If;
    End if;
    if(s.MATCH_NUMBER=NEW.MATCH_NUMBER) then
        if(s.PLAYER_ID_OUT = NEW.ASSISTED_BY) then
            if(s.TIME_IN_OUT<new.GOAL_TIME) then
                Insert into SUB_trig_table
Values(New.MATCH_NUMBER,New.PLAYER_ID_IN,New.PLAYER_ID_OUT,NEW.PLAY_SCHEDULE,NEW
.TIME_IN_OUT);
                RETURN NULL;
            End If;
        End If;
    End if;

```

## Triggers

```
End Loop;
for b in SELECT* from BOOKING
LOOP
    if(b.PLAYER_ID = NEW.SCORED_BY) then
        if(b.BOOKING_TYPE = 'RED') then
            if(b.BOOKING_TIME < NEW.GOAL_TIME) then
                Insert into SUB_trig_table
Values(New.MATCH_NUMBER,New.PLAYER_ID_IN,New.PLAYER_ID_OUT,NEW.PLAY_SCHEDULE,NEW
.TIME_IN_OUT);
                RETURN NULL;
            end if;
        end if;
    end if;
    if(b.PLAYER_ID = NEW.ASSISTED_BY) then
        if(b.BOOKING_TYPE = 'RED') then
            if(b.BOOKING_TIME < NEW.GOAL_TIME) then
                Insert into SUB_trig_table
Values(New.MATCH_NUMBER,New.PLAYER_ID_IN,New.PLAYER_ID_OUT,NEW.PLAY_SCHEDULE,NEW
.TIME_IN_OUT);
                RETURN NULL;
            end if;
        end if;
    end if;
end loop;
RETURN NEW;
END;$GOAL_trig_table$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER GOAL_trig_table
BEFORE INSERT ON GOALS
FOR EACH ROW EXECUTE PROCEDURE GOAL_trig_func();
```

```
--player
CREATE TABLE PLAYER_trig_table(
player_id INT NOT NULL,
player_name Varchar(50) NOT NULL,
team_name Varchar(50) NOT NULL,
age INT NOT NULL,
nationality varchar(20) NOT NULL,
jursey_no INT NOT NULL
);
```

```
Create or Replace Function PLAYER_trig_func() Returns Trigger as
$PLAYER_trig_table$
Declare
    p PLAYER%rowtype;
Begin
    for p in SELECT* from PLAYER
    LOOP
```

```

Triggers
    if(new.TEAM_NAME = p.TEAM_NAME) then
        if(new.JERSEY_NO = p.JERSEY_NO) then
            INSERT INTO PLAYER_trig_table
values(new.PLAYER_ID,new.PLAYER_NAME,new.TEAM_NAME,new.AGE,new.NATIONALITY,NEW.J
ERSeY_NO);
            RETURN NULL;
        end if;
    end if;
END LOOP;
RETURN NEW;
END;$PLAYER_trig_table$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER PLAYER_trig_table
BEFORE INSERT ON PLAYER
FOR EACH ROW EXECUTE PROCEDURE PLAYER_trig_func();

```

```

--playeru
CREATE TABLE PLAYERU_trig_table(
player_id INT NOT NULL,
player_name Varchar(50) NOT NULL,
team_name Varchar(50) NOT NULL,
age INT NOT NULL,
nationality varchar(20) NOT NULL,
jurseY_no INT NOT NULL
);

```

Create or Replace Function PLAYERU\_trig\_func() Returns Trigger as  
\$PLAYERU\_trig\_table\$

Declare

```

    p PLAYER%rowtype;

```

Begin

```

    for p in SELECT* from PLAYER
    LOOP

```

```

        if(new.TEAM_NAME = p.TEAM_NAME) then
            if(new.JERSEY_NO = p.JERSEY_NO) then
                INSERT INTO PLAYER_trig_table

```

```

values(new.PLAYER_ID,new.PLAYER_NAME,new.TEAM_NAME,new.AGE,new.NATIONALITY,NEW.J
ERSeY_NO);

```

```

            RETURN NULL;

```

```

        end if;

```

```

    end if;

```

```

    END LOOP;

```

```

RETURN NEW;

```

```

END;$PLAYERU_trig_table$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER PLAYERU_trig_table
BEFORE UPDATE ON PLAYER
FOR EACH ROW EXECUTE PROCEDURE PLAYERU_trig_func();

```

## Triggers

--Substitution

```
CREATE TABLE SUB_trig_table(  
Match_no INT NOT NULL,  
player_id_in INT NOT NULL,  
player_id_out INT NOT NULL,  
playschedule GS NOT NULL,  
time_inout INT NOT NULL  
);
```

Create or Replace Function SUB\_trig\_func() Returns Trigger as \$SUB\_trig\_table\$  
Declare

```
pin PLAYS_ON%rowtype;  
pout PLAYS_ON%rowtype;  
playerin PLAYER%rowtype;  
playerout PLAYER%rowtype;  
s substitution%rowtype;  
p PLAYS_ON%rowtype;  
pl PLAYER%rowtype;  
m MATCH%rowtype;  
count integer;
```

Begin

```
for s in select* from substitution  
LOOP
```

```
if(s.MATCH_NUMBER=NEW.MATCH_NUMBER) then  
if(S.PLAYER_ID_OUT = NEW.PLAYER_ID_OUT) then  
Insert into SUB_trig_table
```

```
Values(New.MATCH_NUMBER,New.PLAYER_ID_IN,New.PLAYER_ID_OUT,NEW.PLAY_SCHEDULE,NEW  
.TIME_IN_OUT);
```

```
RETURN NULL;
```

```
End If;
```

```
if(S.PLAYER_ID_OUT = NEW.PLAYER_ID_IN)then  
Insert into SUB_trig_table
```

```
Values(New.MATCH_NUMBER,New.PLAYER_ID_IN,New.PLAYER_ID_OUT,NEW.PLAY_SCHEDULE,NEW  
.TIME_IN_OUT);
```

```
RETURN NULL;
```

```
End If;
```

```
if(S.PLAYER_ID_IN = NEW.PLAYER_ID_IN) then  
Insert into SUB_trig_table
```

```
Values(New.MATCH_NUMBER,New.PLAYER_ID_IN,New.PLAYER_ID_OUT,NEW.PLAY_SCHEDULE,NEW  
.TIME_IN_OUT);
```

```
RETURN NULL;
```

```
End If;
```

```
End if;
```

```
End Loop;
```

```
for p in select* from PLAYS_ON  
LOOP
```

```
if(new.PLAYER_ID_IN = p.PLAYER_ID) then  
pin = p;
```

```
END if;
```

```
if(new.PLAYER_ID_OUT = p.PLAYER_ID) then
```

```

Triggers
        pout = p;
    End if;
END LOOP;
if(pin.PID != pout.PID) then
    Insert into SUB_trig_table
Values(New.MATCH_NUMBER,New.PLAYER_ID_IN,New.PLAYER_ID_OUT,NEW.PLAY_SCHEDULE,NEW
.TIME_IN_OUT);
    RETURN NULL;
End if;
for pl in select* from PLAYER
LOOP
    if(new.PLAYER_ID_IN = pl.PLAYER_ID) then
        playerin = pl;
    End if;
    if(new.PLAYER_ID_OUT = pl.PLAYER_ID) then
        playerout = pl;
    End if;
End LOOP;
if(playerin.TEAM_NAME != playerout.TEAM_NAME) then
    Insert into SUB_trig_table
Values(New.MATCH_NUMBER,New.PLAYER_ID_IN,New.PLAYER_ID_OUT,NEW.PLAY_SCHEDULE,NEW
.TIME_IN_OUT);
    Return NULL;
End if;
for m in select* from MATCH
LOOP
    if(m.MATCH_NUMBER = new.MATCH_NUMBER) then
        if(playerin.TEAM_NAME != m.HOME_TEAM) then
            if(playerin.TEAM_NAME != m.AWAY_TEAM) then
                Insert into SUB_trig_table
Values(New.MATCH_NUMBER,New.PLAYER_ID_IN,New.PLAYER_ID_OUT,NEW.PLAY_SCHEDULE,NEW
.TIME_IN_OUT);
            Return NULL;
        End if;
    End if;
    if(playerout.TEAM_NAME != m.HOME_TEAM) then
        if(playerout.TEAM_NAME != m.AWAY_TEAM) then
            Insert into SUB_trig_table
Values(New.MATCH_NUMBER,New.PLAYER_ID_IN,New.PLAYER_ID_OUT,NEW.PLAY_SCHEDULE,NEW
.TIME_IN_OUT);
        Return NULL;
    End if;
End if;
End if;
End loop;
RETURN NEW;
END;$SUB_trig_table$ LANGUAGE plpgsql;

CREATE TRIGGER SUB_trig_table
BEFORE INSERT ON SUBSTITUTION
FOR EACH ROW EXECUTE PROCEDURE SUB_trig_func();

```