

DATABASE MANAGEMENT SYSTEM PROJECT

IPL SCORER – CRICKIE

BY:

PRATHESH N

2018103576

**TABLES, ER DIAGRAMS, RELATIONAL SCHEMA, TRIGGERS,
PROCEDURES, FUNCTIONS, VIEWS, CURSORS**

INTRODUCTION:

IPL is a biggest cricketing event happening every year. This biggest T20 tournament conducted between various franchises, attracts a greater viewership than any other event in India. Because of this, several other organizations develop various other applications and softwares related to IPL like scoring app , fantasy league etc... which not only attracts user's attraction but also brings in rapid growth in those organizations.

OBJECTIVES:

- To provide instantaneous score to the clients those who use this software
- To know about Players personal info, stats ..
- To know information about the current season
- To get information about various franchises participated every year and their stats like win-loss statistics etc..
- Seasonal orange cap, purple cap and winners

SOFTWARE REQUIREMENTS:

Front-end : HTML,CSS,JS,jQuery

Back-end : PHP, MySQL

Database Server:10.1.38-MariaDB

Database Tools:XAMPP version 7.2.16

TABLES:

NAME: TEAM,TEAM2

DESCRIPTION: Gives information about individual scores for a particular match

Column Name	Data Type	Null	Default	Foreign Key
MATCH_ID	int(11)	No	None	MATCHES (MATCH_ID)
TEAM	int(11)	Yes	NULL	FRANCHISE (FID)
PNAME	varchar(30)	No	None	
RUNS	int(11)	Yes	0	
BALLS	int(11)	Yes	0	
OVERS	decimal(2,1)	Yes	0.0	
MAIDEN	int(11)	Yes	0	
RUN	int(11)	Yes	0	
WICKET	int(11)	Yes	0	
O_ST	decimal(2,1)	Yes	0.0	
B_ST	decimal(2,1)	Yes	0.0	
FLDR	varchar(30)	Yes	NULL	PLAYER (PNAME)
WAY_OUT	varchar(10)	Yes	NULL	
ORDER_OF_BAT	int(11)	Yes	0	
YEAR	int(11)	No	None	MATCHES (YEAR)
FOW	int(11)	Yes	0	
FOW_B	decimal(3,1)	Yes	0.0	
FOW_R	int(11)	Yes	0	
BWLR	varchar(30)			PLAYER (PNAME)

NAME: MATCHES

DESCRIPTION: Holds information about list of matches every year

Column Name	Data Type	Null	Default	Foreign key
MATCH_ID	int(11)	No	None	
TEAM1	int(11)	Yes	NULL	FRANCHISE (FID)
TEAM2	int(11)	Yes	NULL	FRANCHISE (FID)
YEAR	int(11)	No	2020	
STATUS	int(11)	Yes	0	
WINNER	int(11)	Yes	NULL	FRANCHISE (FID)

NAME: CAPWK

DESCRIPTION: Have captain's and wicketkeeper's of the teams participating in a match

Column Name	Data Type	Null	Default	Foreign key
MATCH_ID	int(11)	No	None	MATCHES (MATCH_ID)
YEAR	Int(11)	No	None	MATCHES (YEAR)
CAP1	varchar(30)	Yes	NULL	PLAYER (PNAME)
CAP2	varchar(30)	Yes	NULL	PLAYER (PNAME)
WK1	varchar(30)	Yes	NULL	PLAYER (PNAME)
WK2	varchar(30)	Yes	NULL	PLAYER (PNAME)

NAME: STATS

DESCRIPTION: Maintains player's stats

Column Name	Data Type	Null	Default	Foreign Key
PNAME		No	None	PLAYER (PNAME)
RUNS	int(11)	Yes	0	
BALLS_FACED	int(11)	Yes	0	
STRIKE_RATE	decimal(5,2)	Yes	0.00	
WICKETS	int(11)	Yes	0	
OVERS_BOWLED	decimal(5,1)	Yes	0.0	
RUNS_C	int(11)	Yes	0	
ECONOMY	decimal(5,2)	Yes	0.00	
MAIDEN	int(11)	Yes	0	
THREE_HAUL	int(11)	Yes	0	
FIVE_HAUL	int(11)	Yes	0	
FIFTIES	int(11)	Yes	0	
THIRTIES	int(11)	Yes	0	
HUNDREDS	int(11)	Yes	0	
NOS	int(11)	Yes	0	
CATCHES	int(11)	Yes	0	
STUMPINGS	int(11)	Yes	0	
RUN_OUTS	int(11)	Yes	0	
MATCHES	int(11)	Yes	0	
INNINGS_B	int(11)	Yes	0	
INNINGS_BO	int(11)	Yes	0	
AVERAGE	decimal(5,2)	Yes	0.00	

NAME: TOSS

DESCRIPTION: To know information about the team which won the toss and their choice

Column Name	Data Type	Null	Default	Foreign key
MATCH_ID	int(11)	No	None	MATCHES (MATCH_ID)
YEAR	int(11)	No	None	MATCHES (YEAR)
TOSS	int(11)	Yes	None	FRANCHISE (FID)
CHOICE	varchar(30)	Yes	NULL	

NAME: SCORE

DESCRIPTION: To know the total score of both teams participating in a match

Column Name	Data Type	Null	Default	Foreign key
MATCH_ID	int(11)	No	None	MATCHES (MATCH_ID)
YEAR	int(11)	No	None	MATCHES (YEAR)
TEAM	int(11)	No	None	FRANCHISE (FID)
INNINGS	int(11)	Yes	None	
RUNS	int(11)	Yes	0	
WICKET	int(11)	Yes	0	
BALLS	Int(11)	Yes	0	
OVERS	Int(11)	Yes	0	
M_ST	Int(11)	Yes	0	

NAME: PLAYER

DESCRIPTION: This table has a List of all the players played in the IPL

Column Name	Data Type	Null	Default	Foreign key
PNAME	varchar(30)	No	None	
NATION	varchar(20)	Yes	NULL	
DOB	date	Yes	NULL	
ROLE	varchar(30)	Yes	NULL	
BATTING_S	varchar(30)	Yes	NULL	
BOWLING_S	varchar(30)	Yes	NULL	
SEARCH	Int(11)	Yes	NULL	

NAME: PLAYER_BIO

DESCRIPTION: This table has a List of all the players and their stats in each season

Column Name	Data Type	Null	Default	Foreign key
PNAME	varchar(30)	No	None	PLAYER (PNAME)
YEAR	Int(11)	No	None	
FID_PLAYED	Int(11)	Yes	NULL	FRANCHISE (FID)
RUNS	int(11)	Yes	0	
INNINGS_B	int(11)	Yes	0	
WICKETS	Int(11)	Yes	0	
INNINGS_BO	Int(11)	Yes	0	

NAME: FRANCHISE

DESCRIPTION: This table has a List of all the franchises participated in the ipl so far

Column Name	Data Type	Null	Default	Foreign key
FID	Int(11)	No	None	
FNAME	varchar(30)	No	None	
F_OWNER	varchar(30)	Yes	NULL	
NO_OF_TR	int(11)	Yes	0	

NAME: TEAM_BIO

DESCRIPTION: This table has a List of all the Franchises and their stats in each season

Column Name	Data Type	Null	Default	Foreign key
FID	int(11)	No	None	FRANCHISE (FID)
YEAR	Int(11)	No	None	
MATCHES	int(11)	Yes	0	
WINS	int(11)	Yes	0	
LOSES	Int(11)	Yes	0	
DRAWS	Int(11)	Yes	0	
NO_RES	Int(11)	Yes	0	

NAME: ORANGE_CAP

DESCRIPTION: This table gives the information about each season's highest run-getter

Column Name	Data Type	Null	Default	Foreign key
PNAME	varchar(30)	Yes	NULL	PLAYER (PNAME)
YEAR	Int(11)	No	None	

NAME: PURPLE_CAP

DESCRIPTION: This table gives the information about each season's highest wicket-taker

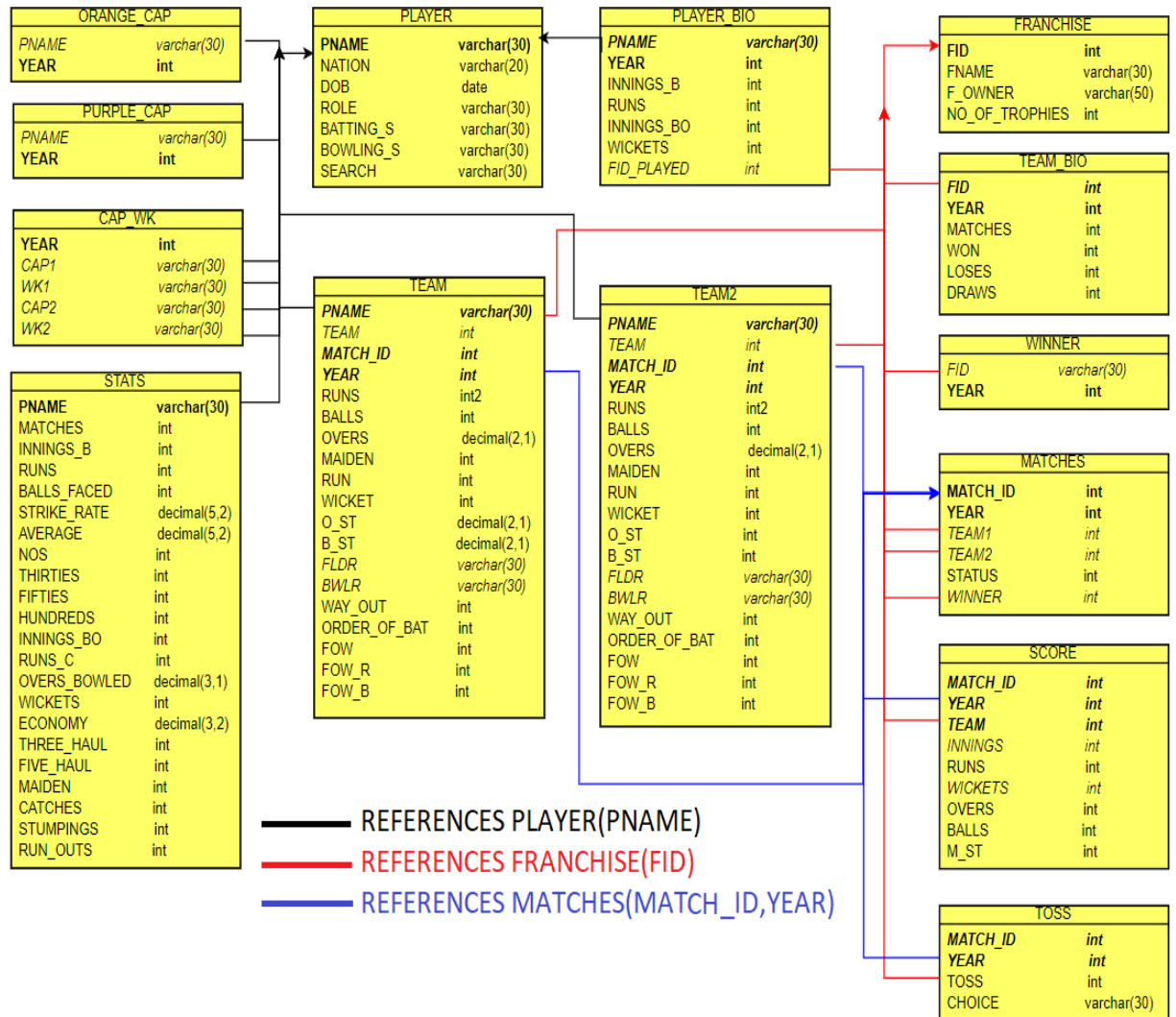
Column Name	Data Type	Null	Default	Foreign key
PNAME	varchar(30)	Yes	NULL	PLAYER (PNAME)
YEAR	Int(11)	No	None	

NAME: WINNER

DESCRIPTION: This table gives the information about each season's title-winner

Column Name	Data Type	Null	Default	Foreign key
FID	int(11)	Yes	NULL	FRANCHISE (FID)
YEAR	Int(11)	No	None	

ER DIAGRAMS:



TRIGGERS:

1) CREATE OR REPLACE TRIGGER **PB_UPD**

AFTER UPDATE ON **team**

FOR EACH ROW

BEGIN

IF OLD.RUNS <> NEW.RUNS OR OLD.WICKET <> NEW.WICKET THEN

UPDATE **PLAYER_BIO** SET PLAYER_BIO.RUNS = RUNS - OLD.RUNS + NEW.RUNS, PLAYER_BIO.WICKETS = WICKETS-OLD.WICKET + NEW.WICKET WHERE PLAYER_BIO.PNAME = OLD.PNAME;

END IF;

END;

OBJECTIVE: Whenever a batsman scores runs or a bowler takes a wicket it also gets updated in **PLAYER_BIO** table

2) CREATE OR REPLACE TRIGGER **NOT_OUT**

AFTER UPDATE ON **SCORE**

FOR EACH ROW

BEGIN

IF NEW.M_ST = 2 AND OLD.M_ST = 1 THEN

IF NEW.INNINGS = 1 THEN

UPDATE **STATS** SET NOS=NOS+1 WHERE PNAME IN(SELECT PNAME FROM **TEAM** WHERE O_ST = 1.0 OR O_ST = 1.5);

END IF;

IF NEW.INNINGS = 2 THEN

UPDATE **STATS** SET NOS=NOS+1 WHERE PNAME IN(SELECT PNAME FROM **TEAM2** WHERE O_ST =1.0 OR O_ST=1.5);

END IF;

END IF;

END

OBJECTIVE: Whenever an inning ends the Not-Out batsmen's NOS (number of Not Outs) will be incremented in **STATS**

3) CREATE OR REPLACE TRIGGER **WIN_LOSS**

AFTER UPDATE ON **matches**

FOR EACH ROW

BEGIN

IF OLD.MATCH_ID <= 28 THEN

IF NEW.STATUS = 2 AND OLD.STATUS <> 2 THEN

UPDATE **team_bio** SET MATCHES=MATCHES+1 WHERE FID=OLD.TEAM1 OR FID=OLD.TEAM2;

UPDATE **team_bio** SET DRAWS=DRAWS+1 WHERE FID=OLD.TEAM1 OR FID=OLD.TEAM2;

END IF;

IF NEW.WINNER IS NOT NULL AND OLD.WINNER IS NULL THEN

UPDATE **team_bio** SET WINS=WINS+1 WHERE FID=NEW.WINNER;

UPDATE **team_bio** SET DRAWS=DRAWS-1 WHERE FID=OLD.TEAM1 OR FID=OLD.TEAM2;

UPDATE **team_bio** SET LOSES=LOSES+1 WHERE (FID=OLD.TEAM1 OR FID=OLD.TEAM2) AND FID<>NEW.WINNER;

END IF;

END IF;

END;

OBJECTIVE: Whenever a match ends the respective team which had won lost will be updated in **TEAM_BIO**

4) CREATE OR REPLACE TRIGGER **MT_STAT**

AFTER UPDATE ON **MATCHES**

FOR EACH ROW

BEGIN

IF OLD.STATUS = 0 AND NEW.STATUS = 1 THEN

UPDATE **STATS** SET MATCHES=MATCHES+1 WHERE PNAME IN (SELECT PNAME FROM TEAM WHERE MATCH_ID=NEW.MATCH_ID
UNION SELECT PNAME FROM TEAM2 WHERE MATCH_ID = NEW.MATCH_ID);

END IF;

END;

OBJECTIVE: This is a very simple trigger to increment the matches in **STATS** of every individual player selected in the squad

5) CREATE OR REPLACE TRIGGER **MDN_TEAM1**

BEFORE UPDATE ON **team2**

FOR EACH ROW

BEGIN

DECLARE RET INT;

DECLARE RT INT;

IF OLD.OVERS <> NEW.OVERS THEN

SET RT = **MDEN**(NEW.MATCH_ID, NEW.RUN, OLD.OVERS, NEW.OVERS);

IF RT = 1 THEN

UPDATE MDN SET MAIDEN = NEW.RUN WHERE MATCH_ID = NEW.MATCH_ID;

END IF;

SET RET=**MDEN_C1**(NEW.MATCH_ID, NEW.RUN, OLD.OVERS, NEW.OVERS);

IF RET = 1 THEN

SET NEW.MAIDEN = OLD.MAIDEN + 1;

UPDATE **STATS** SET MAIDEN = MAIDEN + 1 WHERE PNAME = NEW.PNAME;

END IF;

END IF;

END

OBJECTIVE: This checks whether the just concluded over is a maiden or not if yes it updates in **STATS**.

FUNCTIONS & PROCEDURES USED: MDEN(), MDEN_C1().

6) CREATE OR REPLACE TRIGGER **INS_STAT**

AFTER UPDATE ON **team**

FOR EACH ROW

BEGIN

DECLARE RET INT;

CALL **INNING**(NEW.PNAME,NEW.YEAR,OLD.O_ST,NEW.O_ST,OLD.OVERS,NEW.OVERS);

UPDATE **STATS** SET stats.RUNS=(stats.RUNS+NEW.RUNS-OLD.RUNS) WHERE PNAME=NEW.PNAME;

UPDATE **STATS** SET stats.BALLS_FACED=(stats.BALLS_FACED+NEW.BALLS-OLD.BALLS) WHERE PNAME=NEW.PNAME;

UPDATE **STATS** SET stats.OVERS_BOWLED=(stats.OVERS_BOWLED+NEW.OVERS-OLD.OVERS) WHERE PNAME=NEW.PNAME;

UPDATE **STATS** SET stats.WICKETS=(stats.WICKETS+NEW.WICKET-OLD.WICKET) WHERE PNAME=NEW.PNAME;

UPDATE **STATS** SET stats.RUNS_C=(stats.RUNS_C+NEW.RUN-OLD.RUN) WHERE PNAME=NEW.PNAME;

CALL **MILESTONE** (NEW.PNAME, OLD.RUNS, NEW.RUNS);

CALL **MILESTONE1** (NEW.PNAME, OLD.WICKET, NEW.WICKET);

SET RET=FLD_ST (NEW.FLDR, OLD.FLDR, NEW.WAY_OUT);

END;

OBJECTIVES: This updates runs, balls, overs, wickets, runs conceded instantaneously .

This also increments thirties, fifties, hundreds if anything respectively is reached.

This increments 3w haul and 5w haul if there are any .

This also sets the fielder and bowler, if a batsman is out

FUNCTIONS & PROCEDURES USED: INNING(), MILESTONE(), MILESTONE1(), FLD_ST()

7) CREATE OR REPLACE TRIGGER **STR_ECO_AVG**

BEFORE UPDATE ON **stats**

FOR EACH ROW

BEGIN

IF NEW.BALLS_FACED <> 0 THEN

SET NEW.STRIKE_RATE = **ST_EC_AV**(1,NEW.RUNS, NEW.BALLS_FACED, NEW.INNINGS_B, NEW.NOS,
NEW.OVERS_BOWLED, NEW.RUNS_C);

END IF;

IF NEW.INNINGS_B<>NEW.NOS THEN

SET NEW.AVERAGE = **ST_EC_AV**(2,NEW.RUNS, NEW.BALLS_FACED, NEW.INNINGS_B, NEW.NOS, NEW.OVERS_BOWLED,
NEW.RUNS_C);

END IF;

IF NEW.OVERS_BOWLED <> 0.0 THEN

SET NEW.ECONOMY = **ST_EC_AV**(3,NEW.RUNS, NEW.BALLS_FACED, NEW.INNINGS_B, NEW.NOS, NEW.OVERS_BOWLED,
NEW.RUNS_C);

END IF;

END;

OBJECTIVE: This just sets the derived attributes strike rate, economy, average in **STATS**

FUNCTIONS & PROCEDURES USED: ST_EC_AV ()

8) CREATE OR REPLACE TRIGGER **ORG_PURP**

AFTER UPDATE ON **player_bio**

FOR EACH ROW

BEGIN

IF OLD.RUNS <> NEW.RUNS AND **ORS_PURP_FUNC**(1, NEW.YEAR, NEW.RUNS, NEW.WICKETS) THEN

UPDATE **ORANGE_CAP** SET PNAME=NEW.PNAME WHERE YEAR=NEW.YEAR;

END IF;

IF OLD.WICKETS <> NEW.WICKETS AND **ORS_PURP_FUNC**(2, NEW.YEAR, NEW.RUNS, NEW.WICKETS) THEN

UPDATE **PURPLE_CAP** SET PNAME=NEW.PNAME WHERE YEAR=NEW.YEAR;

END IF;

END;

OBJECTIVE: This checks whether there is a change in order in orange cap and purple cap, if so it updates the same

FUNCTIONS & PROCEDURES USED: ORS_PURP_FUNC()

PROCEDURES:

1) CREATE OR REPLACE PROCEDURE **INNING**(IN NAME VARCHAR(30), IN YR INT, IN OBT DECIMAL(3,1), IN NBT DECIMAL(3,1), IN OBLT DECIMAL(3,1), IN NBLT DECIMAL(3,1))

BEGIN

IF(OBT = 0 AND (NBT = 1.0 OR NBT = 1.5)) THEN

UPDATE **STATS** SET INNINGS_B=INNINGS_B+1 WHERE PNAME=NAME;

UPDATE **PLAYER_BIO** SET INNINGS_B=INNINGS_B+1 WHERE PNAME=NAME AND YEAR=YR;

END IF;

IF(OBLT = 0.0 AND NBLT =0.1) THEN

UPDATE **STATS** SET INNINGS_BO=INNINGS_BO+1 WHERE PNAME=NAME;

UPDATE **PLAYER_BIO** SET INNINGS_BO=INNINGS_BO+1 WHERE PNAME=NAME AND YEAR=YR;

END IF;

END;

OBJECTIVE: If a batsman comes to the crease his batting inning will be incremented and if a bowler bowls atleast one ball then his bowling inning will be incremented.

2) CREATE OR REPLACE PROCEDURE **MILESTONE**(IN NAME VARCHAR(30),IN ORU INT,IN NR INT)

BEGIN

IF NR >= 30 AND ORU < 30 THEN

UPDATE **STATS** SET THIRTIES=THIRTIES+1 WHERE PNAME=NAME;

END IF;

IF NR >= 50 AND ORU < 50 THEN

UPDATE **STATS** SET THIRTIES=THIRTIES-1 WHERE PNAME=NAME;

UPDATE **STATS** SET FIFTIES=FIFTIES+1 WHERE PNAME=NAME;

END IF;

IF NR >= 100 AND ORU < 100 THEN

UPDATE **STATS** SET FIFTIES=FIFTIES-1 WHERE PNAME=NAME;

UPDATE **STATS** SET HUNDREDS=HUNDREDS+1 WHERE PNAME=NAME;

END IF;

END;

OBJECTIVE: If a batsman crosses 30 or 50 or 100 this procedure will increment that milestone in **STATS**

3) CREATE OR REPLACE PROCEDURE **MILESTONE1**(IN NAME VARCHAR(30),IN OWK INT,IN NWK INT)

BEGIN

IF NWK = 3 AND OWK = 2 THEN

 UPDATE **STATS** SET THREE_HAUL=THREE_HAUL+1 WHERE PNAME=NAME;

END IF;

IF NWK = 5 AND OWK = 4 THEN

 UPDATE **STATS** SET THREE_HAUL=THREE_HAUL-1 WHERE PNAME=NAME;

 UPDATE **STATS** SET FIVE_HAUL=FIVE_HAUL+1 WHERE PNAME=NAME;

END IF;

END;

OBJECTIVE: If a bowler bags a 3 or 5 wicket haul this procedure will update it in **STATS**

4) CREATE OR REPLACE PROCEDURE **CAP_WK** ()

BEGIN

DECLARE NAME VARCHAR(30);

DECLARE MID INT;

DECLARE YR INT;

DECLARE WIN INT;

DECLARE TEAM INT;

DECLARE FIN INT;

DECLARE CRS CURSOR FOR SELECT P.PNAME, M.MATCH_ID, P.FID_PLAYED, P.YEAR, M.WINNER FROM **PLAYER_BIO** P, **CAPWK** C, **MATCHES** M WHERE M.MATCH_ID = C.MATCH_ID AND STATUS = 2 AND M.YEAR=P.YEAR AND M.YEAR=C.YEAR AND (P.PNAME=C.CAP1 OR P.PNAME=C.CAP2);

```

DECLARE CONTINUE HANDLER FOR NOT FOUND SET FIN=1;

UPDATE CAPTAIN SET MATCHES=0,WON=0,DRAW=0;

OPEN CRS;

LABEL: LOOP

    FETCH CRS INTO NAME,MID,TEAM,YR,WIN;

    IF FIN=1 THEN

        LEAVE LABEL;

    END IF;

    IF NAME NOT IN (SELECT PNAME FROM CAPTAIN) THEN

        INSERT INTO CAPTAIN VALUES(NAME,0,0,0);

    END IF;

    IF WIN IS NULL THEN

        UPDATE CAPTAIN SET MATCHES=MATCHES+1,DRAW=DRAW+1 WHERE PNAME=NAME;

    ELSEIF WIN <> TEAM THEN

        UPDATE CAPTAIN SET MATCHES=MATCHES+1 WHERE PNAME=NAME;

    ELSEIF WIN = TEAM THEN

        UPDATE CAPTAIN SET MATCHES=MATCHES+1,WON=WON+1 WHERE PNAME=NAME;

    END IF;

END LOOP;

CLOSE CRS;

END

```

OBJECTIVES: This procedure evaluates the win-loss statistics of a captain.

```

5) CREATE OR REPLACE PROCEDURE CAP_WK1()

BEGIN

DECLARE NAME VARCHAR(30);

DECLARE CAT INT;

DECLARE STMP INT;

DECLARE RO INT;

DECLARE FIN INT;

DECLARE CRS1 CURSOR FOR SELECT DISTINCT P.PNAME, S.CATCHES, S.STUMPINGS, S.RUN_OUTS FROM PLAYER_BIO P, CAPWK C,
SCORE S WHERE M.YEAR=C.YEAR AND (P.PNAME=C.WK1 OR P.PNAME=C.WK2) AND S.PNAME=P.PNAME;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET FIN=1;


UPDATE CAPTAIN SET MATCHES=0,WON=0,DRAW=0;

OPEN CRS1;

LABEL: LOOP

    FETCH CRS1 INTO NAME, CAT, STMP, RO;

    IF FIN=1 THEN

        LEAVE LABEL;

    END IF;

    IF NAME NOT IN (SELECT PNAME FROM WICKET_KP) THEN

        INSERT INTO WICKET_KP VALUES(NAME,0,0,0);

    END IF;

    UPDATE WICKET_KP SET CATCHES=CAT, STUMPINGS=STMP, TOT=CAT+STMP+RO WHERE PNAME=NAME;

END LOOP;

CLOSE CRS1;

END

```

OBJECTIVE: This just evaluates a keeper's total catches, stumpings ..

FUNCTIONS:

1) CREATE OR REPLACE FUNCTION **MDEN**(MID INT,RNS INT,OBL DECIMAL(3,1),NBL DECIMAL(3,1)) RETURNS INT

BEGIN

SET OBL=((OBL*10) MOD 10);

SET NBL=((NBL*10) MOD 10);

IF OBL=0 AND NBL=1 THEN

RETURN 1;

ELSE

RETURN 0;

END IF;

END

OBJECTIVE: This function is used to update the run once a over completes , for checking whether the upcoming over is a maiden or not

2) CREATE OR REPLACE FUNCTION **MDEN_C1**(MID INT,RNS INT,OBL DECIMAL(3,1),NBL DECIMAL(3,1)) RETURNS INT

BEGIN

DECLARE RN INT;

SET OBL=((OBL*10) MOD 10);

SET NBL=((NBL*10) MOD 10);

IF OBL = 5 AND NBL=0 THEN

SELECT MAIDEN INTO RN FROM MDN WHERE MATCH_ID=MID;

IF RN = RNS THEN

RETURN 1;

END IF;

END IF;

RETURN 0;

END

OBJECTIVE: This function is used to check whether the current over ends or not ,if so it checks whether the run at the beginning and the end are same, if same it returns 1

```
3) CREATE OR REPLACE FUNCTION FLD_ST (OFLDR VARCHAR (30), NFLDR VARCHAR (30) , WO VARCHAR (30))
RETURNS INT
BEGIN
IF NFLDR IS NOT NULL AND OFLDR IS NULL THEN
    IF WO = "STUMPED" THEN
        UPDATE STATS SET STUMPINGS=STUMPINGS+1 WHERE PNAME=NFLDR;
    END IF;
    IF WO = "CATCH" THEN
        UPDATE STATS SET CATCHES=CATCHES+1 WHERE PNAME=NFLDR;
    END IF;
    IF WO = "RUN OUT" THEN
        UPDATE STATS SET RUN_OUTS=RUN_OUTS+1 WHERE PNAME=NFLDR;
    END IF;
    RETURN 0;
ELSE
    RETURN 1;
END IF;
END;
```

OBJECTIVE: This sets the fielder and bowler for a batsman who has just gone out.

4) CREATE OR REPLACE FUNCTION **ST_EC_AV**(I INT,RUN INT,BALLS INT,INN INT,NOS INT,OVERS DECIMAL(3,1),RUN_C INT) RETURNS DECIMAL(5,2)

BEGIN

DECLARE RET DECIMAL(5,2);

IF I = 1 THEN

SET RET = (RUN/BALLS)*100;

ELSEIF I = 2 THEN

SET RET = RUN/(INN-NOS);

ELSE

SET RET = RUN_C/(((OVERS*10) MOD 10)+(((OVERS*10) DIV 10)*6))*6;

END IF;

RETURN RET;

END;

OBJECTIVE: This calculates strike rate, economy, average and returns those values.

5) CREATE OR REPLACE FUNCTION **ORG_PURP_FUNC**(I INT,YEAR INT,RUN INT,WICKET INT) RETURNS INT

BEGIN

DECLARE A INT;

IF I = 1 THEN

SELECT RUNS INTO A FROM **ORANGE_CAP** O, **PLAYER_BIO** P WHERE P.YEAR=YEAR AND O.YEAR=YEAR AND O.PNAME=P.PNAME;

IF RUN > A THEN

RETURN 1;

END IF;

ELSE

SELECT WICKETS INTO A FROM **PURPLE_CAP** O, **PLAYER_BIO** P WHERE P.YEAR=YEAR AND O.YEAR=YEAR AND O.PNAME=P.PNAME;

IF WICKET > A THEN

RETURN 1;

END IF;

END IF;

RETURN 0;

END;

OBJECTIVE: This makes sure that orange and purple caps are maintained and updated instantaneously

VIEWS:

1) CREATE OR REPLACE VIEW **MOST_RUNS** AS SELECT PNAME BATSMEN,MATCHES M,INNINGS_B I,RUNS R,AVERAGE A FROM STATS ORDER BY RUNS DESC LIMIT 10;

OBJECTIVE: To know the top 10 run scorers

2) CREATE OR REPLACE VIEW **MOST_WICKETS** AS SELECT PNAME BOWLER,MATCHES M,INNINGS_BO I,WICKETS W, ECONOMY E FROM STATS ORDER BY WICKETS DESC LIMIT 10;

OBJECTIVE: To know the top 10 wicket takers

3) CREATE OR REPLACE VIEW **MOST_RUNS** AS SELECT * FROM STATS WHERE PNAME IN (SELECT PNAME FROM PLAYER_BIO WHERE FID_PLAYED=\$t AND YEAR=\$yr) AND RUNS>0 AND \$vr>0 ORDER BY \$vr DESC LIMIT 10

4) CREATE OR REPLACE VIEW **MOST_OVERS** AS SELECT * FROM STATS WHERE PNAME IN (SELECT PNAME FROM PLAYER_BIO WHERE FID_PLAYED=\$t AND YEAR=\$yr) AND OVERS_BOWLED>0.0 AND \$vr>0 ORDER BY \$vr DESC

(THE ABOVE TWO VIEWS WERE USED IN CLIENTS/stats.php)

CURSORS:

1) DECLARE **CRS1** CURSOR FOR SELECT DISTINCT P.PNAME, S.CATCHES, S.STUMPINGS, S.RUN_OUTS FROM **PLAYER_BIO** P, **CAPWK** C, **SCORE** S WHERE M.YEAR=C.YEAR AND (P.PNAME=C.WK1 OR P.PNAME=C.WK2) AND S.PNAME=P.PNAME;

OBJECTIVE: To display all the captain's winning stats

USED IN: **CAP_WK1()**

2) DECLARE **CRS** CURSOR FOR SELECT P.PNAME, M.MATCH_ID, P.FID_PLAYED, P.YEAR, M.WINNER FROM **PLAYER_BIO** P, **CAPWK** C, **MATCHES** M WHERE M.MATCH_ID = C.MATCH_ID AND STATUS = 2 AND M.YEAR=P.YEAR AND M.YEAR=C.YEAR AND (P.PNAME=C.CAP1 OR P.PNAME=C.CAP2);

OBJECTIVE: To display all the wicket keeper's glove works

USED IN: **CAP_WK()**