Birla Institute of Technology & Science, Pilani, K. K. BIRLA Goa campus
Database Systems (CS F212)
Midsemester Online Exam, 26/02/2020
Maximum marks : 20   Duration : 90 Minutes

INSTRUCTIONS:

1. This online exam contains nine questions, which are divided into Parts A and B. All questions are compulsory.

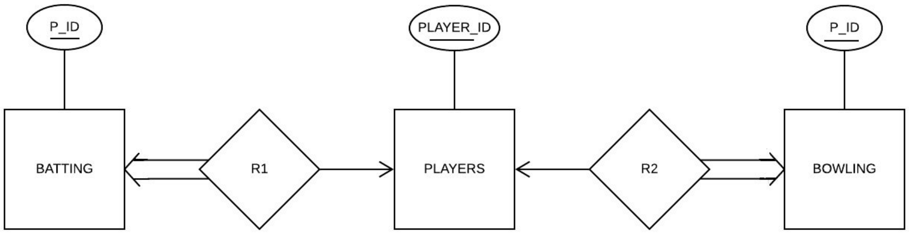
2. Download “db212midsem.sql” file from the course website.

3. Import the downloaded sql script from the command prompt. The sql script will create a database called “db212midsem”. Also, it will create three tables in db212midsem database.

4. You must submit two SQL files: StudentIdXXXXG_PartA.sql and StudentIdXXXXG_PartB.sql, which must contain the queries for Part A and Part B respectively.

Questions:

The tables in the db212midsem database are named: PLAYERS, BATTING and BOWLING. The PLAYERS contain details about cricket players. The BATTING and BOWLING tables contain the batting and bowling scores of the players. The player ID is the primary key in all the three databases.



Part A

The queries for Part A must be saved in StudentIdXXXXG_PartA.sql file.

1. Write a query to add a column BOUNDARIES to the BATTING table. The default value of the BOUNDARIES column should be zero. (1 Mark)

2. Write a query to update the BOUNDARIES column (in the BATTING table) with the total number of fours and sixes (i.e. FOURS+SIXES). The value in the BOUNDARIES column must be updated for only those rows where the RUNS value is greater than 20. (1 Mark)

3. Alter the table definition(s) such that deletion of any player from PLAYERS table should not be permitted if it leads to inconsistent data. Also, any updates in PLAYER_ID (in PLAYERS table) should maintain data consistency across all table(s). Save the output of "show create table" for the tables PLAYERS, BATTING and BOWLING. (3 Marks)

Part B

The queries for Part B must be saved in StudentIdXXXXG_PartB.sql file.

4. Write a query to display the players who debuted more than five years ago and have taken at least two wickets. (Note: Debut is the date on which a player played his/her first match.) (2 Marks)

5. Write a query to display the name of the player who scored the 2nd highest (batting) runs without using any aggregate functions and without using any LIMIT clause. (3 Marks)

6. Write a query to display the name of the Indian Player having 5th highest (batting) runs using only LIMIT clause(s). (2 Marks)

7. Write a query to display the TEAM whose overall batting Run_Rate is more than 7, where Run_Rate = (total no. of runs * 6) ÷ (total no. of balls).
(2 Marks)

8. Write a query to display the Name, Nationality and Bowling Type of players who are either Indian or Right-arm Bowlers but NOT BOTH. Your query should not use the OR operator anywhere. (3 Marks)

9. Write a query using EXISTS/NOT EXISTS to display the Names and Fours scored by Allrounders of MI team who are either Indians or West Indians.
(3 Marks)

Solutions

1. ***SQL COMMAND:*** *alter table BATTING add column BOUNDARIES int default 0;*

2. ***SQL COMMAND:*** *update BATTING set BOUNDARIES=FOURS+SIXES where RUNS>20;*

3. ***Show create table BATTING output may vary depending on the way SQL command was executed. Below are the options***

3.1.1. SQL COMMAND: ALTER TABLE BATTING ADD CONSTRAINT FOREIGN KEY (P_ID) REFERENCES PLAYERS(PLAYER_ID) ON UPDATE CASCADE

OUTPUT : "show create table BATTING"

BATTING	CREATE TABLE `BATTING` (`P_ID` varchar(15) NOT NULL, `RUNS` int DEFAULT NULL, `BALLS` int DEFAULT NULL, `FOURS` int DEFAULT NULL, `SIXES` int DEFAULT NULL, `DISMISSAL` varchar(12) DEFAULT NULL, `BOUNDARIES` int DEFAULT '0', PRIMARY KEY (`P_ID`), CONSTRAINT `BATTING_ibfk_1` FOREIGN KEY (`P_ID`) REFERENCES `PLAYERS` (`PLAYER_ID`) ON UPDATE CASCADE) ENGINE=InnoDB DEFAULT CHARSET=latin1

3.1.2. SQL COMMAND: ALTER TABLE BATTING ADD CONSTRAINT FK_PID1 FOREIGN KEY (P_ID) REFERENCES PLAYERS(PLAYER_ID) ON DELETE RESTRICT ON UPDATE CASCADE

OUTPUT : show create table BATTING

Table	Create Table	
BATTING	CREATE TABLE `BATTING` (`P_ID` varchar(15) NOT NULL, `RUNS` int DEFAULT NULL, `BALLS` int DEFAULT NULL, `FOURS` int DEFAULT NULL, `SIXES` int DEFAULT NULL, `DISMISSAL` varchar(12) DEFAULT NULL, `BOUNDARIES` int DEFAULT '0', PRIMARY KEY (`P_ID`), CONSTRAINT `FK_PID1` FOREIGN KEY (`P_ID`) REFERENCES `PLAYERS` (`PLAYER_ID`) ON DELETE RESTRICT ON UPDATE CASCADE) ENGINE=InnoDB DEFAULT CHARSET=latin1	

3.1.3. SQL COMMAND: ALTER TABLE BOWLING ADD CONSTRAINT FOREIGN KEY (P_ID) REFERENCES PLAYERS(PLAYER_ID) ON UPDATE CASCADE

OUTPUT : show create table BOWLING

Table	Create Table	
BOWLING	CREATE TABLE `BOWLING` (`P_ID` varchar(15) NOT NULL, `OVERS` int DEFAULT NULL, `MAIDENS` int DEFAULT NULL, `RUNS` int DEFAULT NULL,	

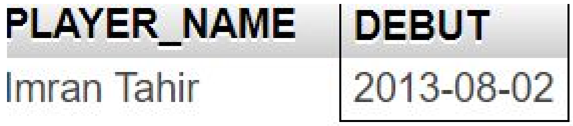
	`WICKETS` int DEFAULT NULL, `DOTS` int DEFAULT NULL, `WIDES` int DEFAULT NULL, PRIMARY KEY (`P_ID`), CONSTRAINT `BOWLING_ibfk_1` FOREIGN KEY (`P_ID`) REFERENCES `PLAYERS` (`PLAYER_ID`) ON UPDATE CASCADE) ENGINE=InnoDB DEFAULT CHARSET=latin1	

3.1.4. SQL COMMAND: ***ALTER TABLE BOWLING ADD CONSTRAINT FK_PID1 FOREIGN KEY (P_ID) REFERENCES PLAYERS(PLAYER_ID) ON DELETE RESTRICT ON UPDATE CASCADE***

OUTPUT : show create table BOWLING

BOWLING	CREATE TABLE `BOWLING` (`P_ID` varchar(15) NOT NULL, `OVERS` int DEFAULT NULL, `MAIDENS` int DEFAULT NULL, `RUNS` int DEFAULT NULL, `WICKETS` int DEFAULT NULL, `DOTS` int DEFAULT NULL, `WIDES` int DEFAULT NULL, PRIMARY KEY (`P_ID`), CONSTRAINT `FK_PID1` FOREIGN KEY (`P_ID`) REFERENCES `PLAYERS` (`PLAYER_ID`) ON DELETE RESTRICT ON UPDATE CASCADE) ENGINE=InnoDB DEFAULT CHARSET=latin1

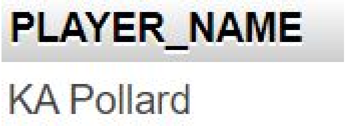
4. ***SQL COMMAND:*** ***SELECT PLAYER_NAME, DEBUT FROM PLAYERS, BOWLING WHERE PLAYER_ID=P_ID AND TIMESTAMPDIFF(YEAR, DEBUT, CURDATE())>=5 AND WICKETS>1;***



PLAYER_NAME	DEBUT
Imran Tahir	2013-08-02

5. ***SQL COMMAND*** ***SELECT PLAYER_NAME FROM PLAYERS WHERE PLAYER_ID IN (SELECT P_ID FROM (SELECT P_ID, RUNS FROM BATTING WHERE RUNS < ANY (SELECT RUNS FROM BATTING)) AS T WHERE RUNS >= ALL (SELECT RUNS FROM BATTING WHERE RUNS < ANY (SELECT RUNS FROM BATTING)));***

OUTPUT :

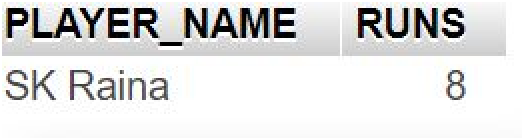


PLAYER_NAME
KA Pollard

6. ***SQL COMMAND:***

```
select * from (SELECT PLAYER_NAME,RUNS FROM PLAYERS,BATTING where P_ID=PLAYER_ID
and NATIONALITY="India" order by RUNS desc limit 5) as T order by RUNS limit 1;
```

OUTPUT :

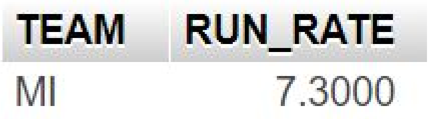


PLAYER_NAME	RUNS
SK Raina	8

7. ***SQL COMMAND:***

```
select TEAM,sum(RUNS)/(sum(BALLS)/6) as RUN_RATE from BATTING,PLAYERS where P_ID=PLAYER_ID group by TEAM
having RUN_RATE>7
```

OUTPUT:



TEAM	RUN_RATE
MI	7.3000

8. ***SQL COMMAND:***

```
SELECT P.PLAYER_NAME, P.NATIONALITY, P.BOWL FROM PLAYERS AS P, ( SELECT PLAYER_NAME FROM PLAYERS WHERE
PLAYER_NAME NOT IN ( SELECT PLAYER_NAME FROM PLAYERS WHERE NATIONALITY != 'India' AND BOWL NOT LIKE
'%Right-arm%' )) AS T1, ( SELECT PLAYER_NAME FROM PLAYERS WHERE PLAYER_NAME NOT IN ( SELECT PLAYER_NAME
FROM PLAYERS WHERE NATIONALITY = 'India' AND BOWL LIKE '%Right-arm%' )) AS T2 WHERE
P.PLAYER_NAME=T1.PLAYER_NAME AND T1.PLAYER_NAME=T2.PLAYER_NAME;
```

9. ***SQL COMMAND:***

```
SELECT PLAYER_NAME, T3.FOURS, NATIONALITY, ROLE, TEAM FROM PLAYERS AS T2, BATTING AS T3 WHERE
T2.PLAYER_ID=T3.P_ID AND NATIONALITY in ('India','West Indies') and ROLE='Allrounder' and TEAM='MI' and Exists (SELECT
T1.FOURS FROM BATTING AS T1 where T2.PLAYER_ID=T1.P_ID);
```

OUTPUT:

```
+-------------+-------+-------------+------------+------+
| PLAYER_NAME | FOURS | NATIONALITY | ROLE       | TEAM |
+-------------+-------+-------------+------------+------+
| KA Pollard  |     3 | West Indies | Allrounder | MI   |
| KH Pandya   |     0 | India       | Allrounder | MI   |
| HH Pandya   |     1 | India       | Allrounder | MI   |
+-------------+-------+-------------+------------+------+
```