IPL DELIVERIES DATA ANALYSIS USING SQL

By SAHIL VERMA

Q1 WRITE A QUERY TO LIST ALL DELIVERIES WITH THEIR MATCH_ID, INNING, OVER, AND BALL.

```
-- SAHIL VERMA
select match_id, inning, overs, ball
from ipldeliveries;
```

Result Grid | Filter Rows:

match_id	inning	overs	ball
1	1	1	1
1	1	1	2
1	1	1	3
1	1	1	4
1	1	1	5
1	1	1	6
1	1	1	7
1	1	2	1

Q2 WRITE A QUERY TO DISPLAY ALL COLUMNS FOR DELIVERIES BOWLED BY A SPECIFIC BOWLER, J BUMRAH.

```
-- SAHIL VERMA
select * from ipldeliveries
where bowler='JJ Bumrah';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

match_id	inning	batting_team	bowling_team	overs	ball	batsman	non_striker	bowler	is_super_over	wi
2	2	Rising Pune Supergiant	Mumbai Indians	5	1	AM Rahane	SPD Smith	JJ Bumrah	0	0
2	2	Rising Pune Supergiant	Mumbai Indians	5	2	SPD Smith	AM Rahane	JJ Bumrah	0	0
2	2	Rising Pune Supergiant	Mumbai Indians	5	3	AM Rahane	SPD Smith	JJ Bumrah	0	0
2	2	Rising Pune Supergiant	Mumbai Indians	5	4	AM Rahane	SPD Smith	JJ Bumrah	0	0
2	2	Rising Pune Supergiant	Mumbai Indians	5	5	SPD Smith	AM Rahane	JJ Bumrah	0	0
2	2	Rising Pune Supergiant	Mumbai Indians	5	6	SPD Smith	AM Rahane	JJ Bumrah	0	0
2	2	Rising Pune Supergiant	Mumbai Indians	12	1	BA Stokes	SPD Smith	JJ Bumrah	0	0

Q3 WRITE A QUERY TO LIST DELIVERIES WHERE THE BATTING_TEAM IS MUMBAI INDIANS.

```
-- SAHIL VERMA
select * from ipldeliveries
where batting_team = 'Mumbai Indians';
```

match_id	inning	batting_team	bowling_team	overs	ball	batsman	non_striker	bowler	is_super_over
2	1	Mumbai Indians	Rising Pune Supergiant	1	1	PA Patel	JC Buttler	AB Dinda	0
2	1	Mumbai Indians	Rising Pune Supergiant	1	2	PA Patel	JC Buttler	AB Dinda	0
2	1	Mumbai Indians	Rising Pune Supergiant	1	3	PA Patel	JC Buttler	AB Dinda	0
2	1	Mumbai Indians	Rising Pune Supergiant	1	4	JC Buttler	PA Patel	AB Dinda	0
2	1	Mumbai Indians	Rising Pune Supergiant	1	5	PA Patel	JC Buttler	AB Dinda	0
2	1	Mumbai Indians	Rising Pune Supergiant	1	6	PA Patel	JC Buttler	AB Dinda	0
2	1	Mumbai Indians	Rising Pune Supergiant	1	7	PA Patel	JC Buttler	AB Dinda	0

Q4 WRITE A QUERY TO FIND ALL DELIVERIES WHERE THE BATSMAN SCORED MORE THAN 4 RUNS OFF A SINGLE BALL.

```
-- SAHIL VERMA
select * from ipldeliveries
where batsman_runs>4;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

overs	ball	batsman	non_striker	bowler	is_super_over	wide_runs	bye_runs	legbye_runs	noball_runs	penalty_runs	batsman_runs	extra_runs
2	4	DA Warner	S Dhawan	A Choudhary	0	0	0	0	0	0	6	0
8	4	MC Henriques	S Dhawan	TM Head	0	0	0	0	0	0	6	0
13	2	Yuvraj Singh	MC Henriques	A Choudhary	0	0	0	0	0	0	6	0
15	3	Yuvraj Singh	MC Henriques	S Aravind	0	0	0	0	0	0	6	0
15	5	MC Henriques	Yuvraj Singh	S Aravind	0	0	0	0	0	0	6	0
18	1	DJ Hooda	Yuvraj Singh	A Choudhary	0	0	0	0	0	0	6	0
19	3	Yuvraj Singh	DJ Hooda	TS Mills	0	0	0	0	0	0	6	0

Q5 WRITE A QUERY TO COUNT THE TOTAL NUMBER OF DELIVERIES IN THE DATASET.

```
31      -- SAHIL VERMA
32 •    select count(*) total_deliveries from ipldeliveries;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total_deliveries
150460

Q6 WRITE A QUERY TO FIND THE AVERAGE NUMBER OF RUNS SCORED PER OVER.

```
35      -- SAHIL VERMA
36 •    select match_id,overs, avg(total_runs) as avg_runs
37      from ipldeliveries
38      group by match_id,overs
39      order by avg_runs desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

match_id	overs	avg_runs
66	18	4.3333
104	20	4.0000
250	20	4.0000
367	20	3.9000
9	20	3.8333
557	15	3.8333
22	15	3.7692
2	20	3.7500
9	19	3.6667
60	20	3.6667

Q7 WRITE A QUERY TO LIST THE TOP 5 BATSMEN WITH THE HIGHEST TOTAL RUNS SCORED.

```
42      -- SAHIL VERMA
43 •    select batsman, sum(total_runs) as total_runs_inall_matches
44      from ipldeliveries
45      group by batsman
46      order by total_runs_inall_matches desc limit 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

batsman	total_runs_inall_matches
SK Raina	4745
V Kohli	4588
G Gambhir	4388
RG Sharma	4334
DA Warner	4213

Q8 WRITE A QUERY TO LIST ALL PAIRS OF BOWLERS WHO BOWLED IN THE SAME MATCH. INCLUDE COLUMNS FOR MATCH_ID, BOWLER1, AND BOWLER2.

```sql
-- SAHIL VERMA
select d1.match_id, d1.bowler as bowler1, d2.bowler as bowler2
from ipldeliveries d1
join ipldeliveries d2 on d1.match_id=d2.match_id and d1.bowler<d2.bowler
order by d1.match_id, bowler1, bowler2;
```

UNFORTUNATELY THIS ERROR CODE IS APPEARING:

✔	18	13:06:59	select batsman, sum(total_runs) as total_runs_inall_matches from ipldeliveries group by batsman order by total_r...	5 row(s) returned
✖	19	13:07:44	select d1.match_id, d1.bowler as bowler1, d2.bowler as bowler2 from ipldeliveries d1 join ipldeliveries d2 on d1....	Error Code: 2013. Lost connection to MySQL server during query
✖	20	13:10:56	select d1.match_id, d1.bowler as bowler1, d2.bowler as bowler2 from ipldeliveries d1 inner join ipldeliveries d2 o...	Error Code: 2013. Lost connection to MySQL server during query
✖	21	13:11:38	select d1.match_id, d1.bowler as bowler1, d2.bowler as bowler2 from ipldeliveries d1 cross join ipldeliveries d2 ...	Error Code: 2013. Lost connection to MySQL server during query

Q9 WRITE A QUERY TO FIND THE TOTAL NUMBER OF DELIVERIES BOWLED BY EACH BOWLER. DISPLAY BOWLER AND THE COUNT OF DELIVERIES.

```
56      -- SAHIL VERMA
57 •    select bowler, count(*) as no_of_deliveries
58      from ipldeliveries
59      group by bowler
60      order by no_of_deliveries desc;
```

Result Grid | Filter Rows: | Export: | Wra

bowler	no_of_deliveries
Harbhajan Singh	2989
A Mishra	2703
SL Malinga	2694
P Kumar	2637
PP Chawla	2594
R Ashwin	2359
Z Khan	2276
R Vinay Kumar	2161
DW Steyn	2159

Q10 WRITE A QUERY USING A CASE STATEMENT TO CATEGORIZE DELIVERIES INTO THREE CATEGORIES BASED ON RUNS SCORED:
DOT BALL FOR 0 RUNS, SINGLE FOR 1 RUN, AND MULTIPLE FOR MORE THAN 1 RUN.

```
64      -- SAHIL VERMA
65      select total_runs,
66      case when total_runs=0 then 'Dot Ball'
67      when total_runs=1 then 'Single'
68      when total_runs>1 then 'Multiple'
69      end as Delivery_Categories
70      from ipldeliveries;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Cont

total_runs	Delivery_Categories
0	Dot Ball
0	Dot Ball
4	Multiple
0	Dot Ball
2	Multiple
0	Dot Ball
1	Single
1	Single

Q11 WRITE A QUERY TO ADD A NEW COLUMN IS_BOUNDARY TO THE DELIVERIES TABLE THAT INDICATES IF THE DELIVERY RESULTED IN A BOUNDARY (4 OR 6 RUNS).

```sql
-- SAHIL VERMA
alter table ipldeliveries
add column is_boundary varchar(10);

update ipldeliveries
set is_boundary= case
when batsman_runs=4 then 'FOUR'
when batsman_runs=6 then 'SIX'
else ' '
end;

select * from ipldeliveries;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

uper_over	wide_runs	bye_runs	legbye_runs	noball_runs	penalty_runs	batsman_runs	extra_runs	total_runs	player_dismissed	dismissal_kind	fielder	is_boundary
	0	0	0	0	0	4	0	4				FOUR
	0	0	0	0	0	0	0	0				
	0	0	0	0	0	4	0	4				FOUR
	0	0	0	0	0	4	0	4				FOUR

Q12 WRITE A QUERY USING AN ADVANCED FUNCTION TO FIND THE OVER WITH THE HIGHEST TOTAL RUNS SCORED.

```
88      -- SAHIL VERMA
89  •   select overs, total_runs_per_over from (select overs,
90      sum(total_runs) as total_runs_per_over,
91      dense_rank() over(order by sum(total_runs) desc) as over_rank
92      from ipldeliveries
93      group by overs)  as over_ranking
94      limit 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

overs	total_runs_per_over
18	10899

Q13 CREATE A VIEW NAMED HIGHSCORINGOVERS THAT INCLUDES OVERS WHERE THE TOTAL RUNS SCORED ARE MORE THAN 15.

```
97      -- SAHIL VERMA
98  •   create view HighScoringOvers as
99      select match_id, overs, sum(total_runs) as total_runs_in_all
100     from ipldeliveries
101     group by match_id, overs
102     having total_runs_in_all>15;
103
104 •   select * from HighScoringOvers;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

match_id	overs	total_runs_in_all
1	1	18
1	2	17
1	3	22
1	4	19
1	6	23
1	8	24
1	9	20
1	10	17

Q14 WRITE A QUERY USING A WINDOW FUNCTION TO RANK DELIVERIES BASED ON THEIR TOTAL_RUNS WITHIN EACH MATCH_ID.

```
107     -- SAHIL VERMA
108 •   select match_id,
109     sum(total_runs) as total_of_runs,
110     dense_rank() over(order by sum(total_runs) desc) as run_ranking
111     from ipldeliveries
112     group by match_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

match_id	total_of_runs	run_ranking
206	469	1
50	453	2
61	447	3
259	433	4
68	431	5
562	431	5
516	428	6
41	422	7

Q15 WRITE A QUERY TO CALCULATE THE CUMULATIVE COUNT OF RUNS SCORED IN EACH MATCH SORTED BY OVER.

```
114     -- SAHIL VERMA
115     -- Q15  Write a query to calculate the cumulative count of runs scored in each match sorted by over.
116 •   SELECT match_id, overs,
117     SUM(total_runs) OVER (PARTITION BY match_id ORDER BY overs) AS cumulative_runs
118     FROM ipldeliveries
119     ORDER BY match_id, overs;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

match_id	overs	cumulative_runs
1	1	18
1	1	18
1	1	18
1	2	35
1	2	35
1	2	35
1	2	35
1	2	35
1	2	35
1	2	35
1	2	35
1	2	35

Q16 WRITE A STORED PROCEDURE TO UPDATE THE RUNS SCORED ON A DELIVERY GIVEN ITS MATCH_ID, INNING, OVER, BALL, AND NEW RUNS.

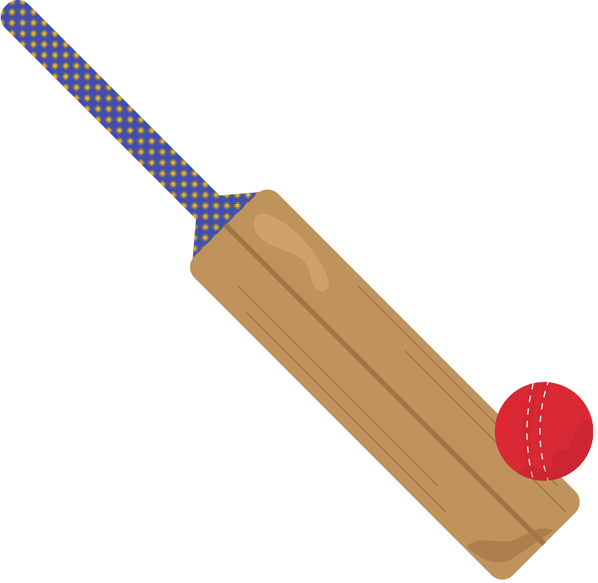
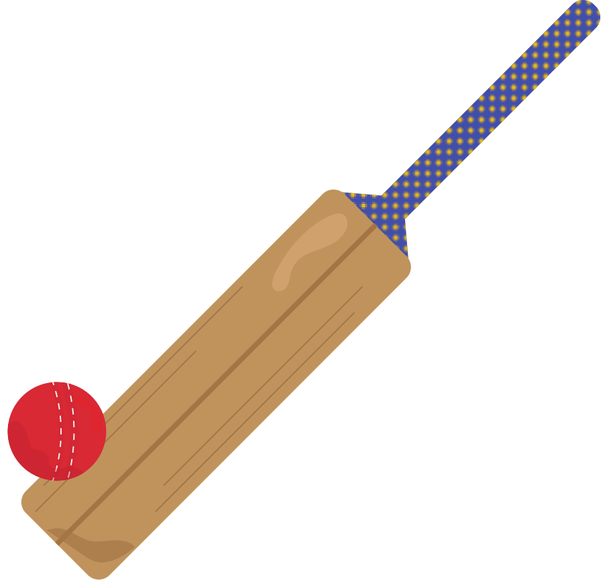
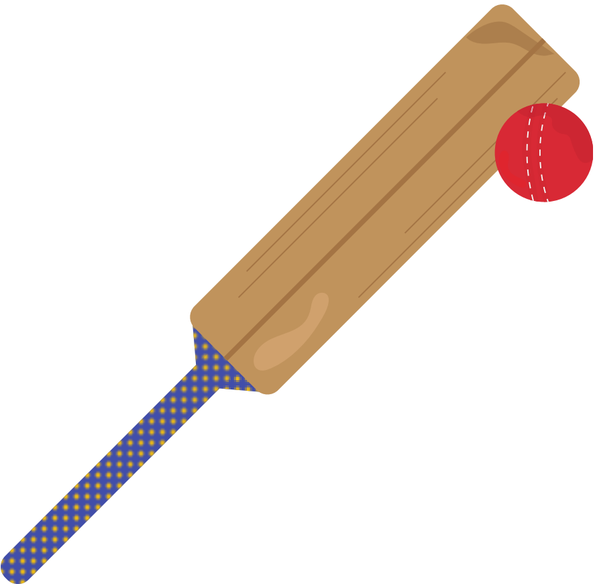
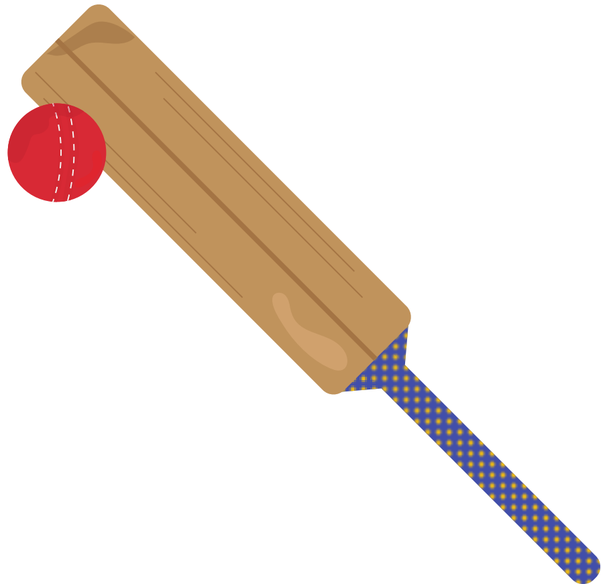
```
-- SAHIL VERMA
-- Q16 Write a stored procedure to update the runs scored on a delivery given its match_id, inning, over, ball, and new runs.
delimiter //
create procedure update_runs(
    in p_match_id int,
    in p_inning int,
    in p_over int,
    in p_ball int,
    in p_new_runs int
)
begin
    update ipldeliveries
    set batsman_runs = p_new_runs
    where match_id = p_match_id
      and inning = p_inning
      and overs = p_over
      and ball = p_ball;
end //

delimiter ;
call update_runs(1,1,5,3,4);
```

Q18. WRITE A QUERY TO FIND PAIRS OF DELIVERIES IN THE SAME OVER WHERE ONE DELIVERY RESULTED IN MORE RUNS THAN THE OTHER. DISPLAY COLUMNS FOR MATCH_ID, INNING, OVER, BALL1, RUNS1, BALL2, AND RUNS2.

```
152     -- SAHIL VERMA
153     -- Q18. Write a query to find pairs of deliveries in the same over where one delivery resulted in more runs than the other.
154      -- Display columns for match_id, inning, over, ball1, runs1, ball2, and runs2.
155     select a.match_id, a.overs, a.ball as ball1, b.ball as ball2, a.batsman_runs as runs1, b.batsman_runs as runs2
156     from ipldeliveries a
157     join ipldeliveries b on a.match_id = b.match_id and a.overs = b.overs and a.inning = b.inning and a.ball < b.ball
158     where a.batsman_runs > b.batsman_runs
159     order by a.match_id, a.overs, a.ball, b.ball;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

match_id	overs	ball1	ball2	runs1	runs2
1	1	1	2	1	0
1	1	1	3	1	0
1	1	3	4	4	0
1	1	3	5	4	0
1	1	3	6	4	0
1	1	3	7	4	0


THANKYOU