

## IPL SQL D.A

Create database IPL;

Use IPL;

CREATE TABLE deliveries

```
(
    match_id INT NOT NULL,
    inning INT NOT NULL,
    batting_team VARCHAR(255) NOT NULL,
    bowling_team VARCHAR(255) NOT NULL,
    over_no INT NOT NULL,
    ball INT NOT NULL,
    batsman VARCHAR(255) NOT NULL,
    non_striker VARCHAR(255) NOT NULL,
    bowler VARCHAR(255) NOT NULL,
    is_super_over INT NOT NULL,
    wide_run INT NOT NULL,
    bye_run INT NOT NULL,
    no_ball_run INT NOT NULL,
    penalty_run INT NOT NULL,
    batsman_run INT NOT NULL,
    extra_run INT NOT NULL,
    total_run INT NOT NULL,
    player_dismissed VARCHAR(255) NOT NULL,
    dismissal_kind VARCHAR(255) NOT NULL,
    fielder VARCHAR(255) NOT NULL,
    PRIMARY KEY (match_id, inning, over_no, ball)
);
```

CREATE TABLE matches

```
(
    id INT NOT NULL,
    season INT NOT NULL,
    city VARCHAR(255) NOT NULL,
    team_1 VARCHAR(255) NOT NULL,
    team_2 VARCHAR(255) NOT NULL,
    toss_winner VARCHAR(255) NOT NULL,
    toss_decision VARCHAR(255) NOT NULL,
    result VARCHAR(255) NOT NULL,
    dl_applied INT NOT NULL,
    winner VARCHAR(255) NOT NULL,
    win_by_runs INT NOT NULL,
    win_by_wickets INT NOT NULL,
    player_of_match VARCHAR(255) NOT NULL,
    venue VARCHAR(255) NOT NULL,
    umpire_1 VARCHAR(255) NOT NULL,
    umpire_2 VARCHAR(255) NOT NULL,
    umpire_3 VARCHAR(255) NOT NULL,
    PRIMARY KEY (id)
);
```

```
Select * from deliveries;
Select * from matches;
```

-- WHAT ARE THE TOP 5 PLAYERS WITH THE MOST PLAYER OF THE MATCH AWARDS?

```
Select player_of_match, count(*) as match_award
from matches
group by player_of_match
order by match_award desc
limit 5;
```

-- HOW MANY MATCHES WERE WON BY EACH TEAM IN EACH SEASON?

```
Select season, winner, count(*) as won_match
from matches
group by season, winner;
```

-- WHAT IS THE AVERAGE STRIKE RATE OF BATSMEN IN THE IPL DATASET?

```
Select (avg(total_runs/total_balls)* 100) as avg_strike_rate
from (
Select batsman, sum(batsman_runs) as total_runs,
count(ball) as total_balls
from deliveries
group by batsman) as individual_strike_rates;
```

-- WHAT IS THE NUMBER OF MATCHES WON BY EACH TEAM BATTING FIRST VERSUS BATTINGSECOND?

```
SELECT batsman,(total_runs * 100.0 / total_balls) AS strike_rate
FROM (SELECT batsman,
SUM(batsman_runs) AS total_runs,
COUNT(*) AS total_balls
FROM deliveries
GROUP BY batsman) AS subquery
WHERE total_runs >= 200
ORDER BY strike_rate DESC
LIMIT 1;
```

-- WHICH BATSMAN HAS THE HIGHEST STRIKE RATE (MINIMUM 200 RUNS SCORED)?

```
select batsman,(sum(batsman_runs)*100/count(*))
as strike_rate
from deliveries group by batsman
having sum(batsman_runs)>=200
order by strike_rate desc
limit 1;
```

-- HOW MANY TIMES HAS EACH BATSMAN BEEN DISMISSED BY THE BOWLER 'MALINGA'?

```
select batsman,count(*) as total_dismissals
from deliveries
where player_dismissed is not null
and bowler='SL Malinga'
group by batsman;
```

-- WHAT IS THE AVERAGE PERCENTAGE OF BOUNDARIES (FOURS AND SIXES COMBINED) HIT BY EACH BATSMAN?

```
select batsman,avg(case when batsman_runs=4 or batsman_runs=6
then 1 else 0 end)*100 as average_boundaries
from deliveries group by batsman;
```

-- WHAT IS THE AVERAGE NUMBER OF BOUNDARIES HIT BY EACH TEAM IN EACH SEASON?

```
select season,batting_team,avg(fours+sixes) as average_boundaries
from(select season,match_id,batting_team,
sum(case when batsman_runs=4 then 1 else 0 end)as fours,
sum(case when batsman_runs=6 then 1 else 0 end) as sixes
from deliveries,matches
where deliveries.match_id=matches.id
group by season,match_id,batting_team) as team_bounseries
group by season,batting_team;
```

-- WHAT IS THE HIGHEST PARTNERSHIP (RUNS) FOR EACH TEAM IN EACH SEASON?

```
select season,batting_team,max(total_runs) as highest_partnership
from(select season,batting_team,partnership,sum(total_runs) as total_runs
from(select season,match_id,batting_team,over_no,
sum(batsman_runs) as partnership,sum(batsman_runs)+sum(extra_runs) as total_runs
from deliveries,matches where deliveries.match_id=matches.id
group by season,match_id,batting_team,over_no) as team_scores
group by season,batting_team,partnership) as highest_partnership
group by season,batting_team;
```

-- How MANY EXTRAS (WIDES & NO-BALLS) WERE BOWLED BY EACH TEAM IN EACH MATCH?

```
select m.id as match_no,d.bowling_team,
sum(d.extra_runs) as extras
from matches as m
join deliveries as d on d.match_id=m.id
where extra_runs>0
group by m.id,d.bowling_team;
```

-- WHICH BOWLER HAS THE BEST BOWLING FIGURES (MOST WICKETS TAKEN) IN A SINGLE EACH MATCH?

```
select m.id as match_no,d.bowler,count(*) as wickets_taken
from matches as m
join deliveries as d on d.match_id=m.id
where d.player_dismissed is not null
group by m.id,d.bowler
```

```
order by wickets_taken desc
limit 1;
```

-- HOW MANY MATCHES RESULTED IN A WIN FOR EACH TEAM IN EACH CITY?

```
select m.city,case when m.team1=m.winner then m.team1
when m.team2=m.winner then m.team2
else 'draw'
end as winning_team,
count(*) as wins
from matches as m
join deliveries as d on d.match_id=m.id
where m.result!='Tie'
group by m.city,winning_team;
```

-- HOW MANY TIMES DID EACH TEAM WIN THE TOSS IN EACH SEASON?

```
select season,toss_winner,count(*) as toss_wins
from matches group by season,toss_winner;
```

-- HOW MANY MATCHES DID EACH PLAYER WIN THE "PLAYER OF THE MATCH" AWARD MATCH?

```
select player_of_match,count(*) as total_wins
from matches
where player_of_match is not null
group by player_of_match
order by total_wins desc;
```

-- WHAT IS THE AVERAGE NUMBER OF RUNS SCORED IN EACH OVER OF THE INNINGS IN

```
select m.id,d.inning,d.over_no,
avg(d.total_runs) as average_runs_per_over
from matches as m
join deliveries as d on d.match_id=m.id
group by m.id,d.inning,d.over_no;
```

-- WHICH TEAM HAS THE HIGHEST TOTAL SCORE IN A SINGLE MATCH?

```
select m.season,m.id as match_no,d.batting_team,
sum(d.total_runs) as total_score
from matches as m
join deliveries as d on d.match_id=m.id
group by m.season,m.id,d.batting_team
order by total_score desc
limit 1;
```

-- WHICH BATSMAN HAS SCORED THE MOST RUNS IN A SINGLE MATCH?

```
select m.season,m.id as match_no,d.batsman,
sum(d.batsman_runs) as total_runs
from matches as m
join deliveries as d on d.match_id=m.id
group by m.season,m.id,d.batsman
order by total_runs desc
```

limit 1;