**IPL DATABASE**

**DDL**

**Database Creation:-**

CREATE DATABASE "IPL database"

WITH

OWNER = postgres

ENCODING = 'UTF8'

LC\_COLLATE = 'en-US'

LC\_CTYPE = 'en-US'

LOCALE\_PROVIDER = 'libc'

TABLESPACE = pg\_default

CONNECTION LIMIT = -1

IS\_TEMPLATE = False;

**Create ENUM types:-**

CREATE TYPE leadership\_role\_enum AS ENUM ('None', 'Captain', 'ViceCaptain');

CREATE TYPE staff\_role\_enum AS ENUM ('Head Coach', 'Batting Coach', 'Bowling Coach', 'Fielding Coach', 'Physio', 'Other');

**Creating Franchise Table:-**

CREATE TABLE Franchise (

franchise\_id SERIAL PRIMARY KEY,

franchise\_name VARCHAR(200) NOT NULL,

city VARCHAR(45) NOT NULL,

trophy\_count INT DEFAULT 0

);

**Creating Player Table:-**

CREATE TABLE Player (

player\_id SERIAL PRIMARY KEY,

player\_name VARCHAR(200) NOT NULL,

role VARCHAR(100) NOT NULL,

contract\_amount INT NOT NULL,

leadership\_role leadership\_role\_enum DEFAULT 'None',

franchise\_id INT,

FOREIGN KEY (franchise\_id) REFERENCES Franchise(franchise\_id) ON DELETE CASCADE

);

**Creating Owner Table:-**

CREATE TABLE Owner (

owner\_id SERIAL PRIMARY KEY,

owner\_name VARCHAR(200) NOT NULL,

group\_name VARCHAR(200),

city VARCHAR(200),

franchise\_id INT,

FOREIGN KEY (franchise\_id) REFERENCES Franchise(franchise\_id) ON DELETE CASCADE

);

**Creating Stadium Table:-**

CREATE TABLE Stadium (

stadium\_id SERIAL PRIMARY KEY,

stadium\_name VARCHAR(200) NOT NULL,

city VARCHAR(200) NOT NULL,

stadium\_area INT NOT NULL,

total\_seat INT NOT NULL,

franchise\_id INT,

FOREIGN KEY (franchise\_id) REFERENCES Franchise(franchise\_id) ON DELETE CASCADE

);

**Creating Sponsor Table:-**

CREATE TABLE Sponsor (

sponsor\_id SERIAL PRIMARY KEY,

sponsor\_name VARCHAR(200) NOT NULL

);

**Many-to-Many Relationship: Sponsor & Franchise Table :-**

CREATE TABLE SponsorFranchise (

sponsor\_id INT,

franchise\_id INT,

contract\_start\_date DATE NOT NULL,

contract\_end\_date DATE NOT NULL,

sponsorship\_amount INT NOT NULL,

PRIMARY KEY (sponsor\_id, franchise\_id),

FOREIGN KEY (sponsor\_id) REFERENCES Sponsor(sponsor\_id) ON DELETE CASCADE,

FOREIGN KEY (franchise\_id) REFERENCES Franchise(franchise\_id) ON DELETE CASCADE,

CHECK (contract\_end\_date > contract\_start\_date)

);

**Creating Supporting Staff Table:-**

CREATE TABLE SupportingStaff (

staff\_id SERIAL PRIMARY KEY,

staff\_name VARCHAR(200) NOT NULL,

role staff\_role\_enum NOT NULL,

franchise\_id INT,

FOREIGN KEY (franchise\_id) REFERENCES Franchise(franchise\_id) ON DELETE CASCADE

);

**Creating Winner Table:-**

Create table Winner(

winner\_id SERIAL PRIMARY KEY,

franchise\_id INT NOT NULL,

winning\_year INT NOT NULL,

captain\_name Varchar(100) not null,

player\_is\_active Boolean Not null default true,

runner\_up VARCHAR(200) NOT NULL,

man\_of\_match VARCHAR(100) NOT NULL,

player\_of\_tournament VARCHAR(100) NOT NULL,

FOREIGN KEY (franchise\_id) REFERENCES Franchise(franchise\_id) ON DELETE CASCADE

);

**INSERTINFG VALUES IN TABLE:-**

-- Insert data into Franchise Table

INSERT INTO Franchise (franchise\_name, city, trophy\_count) VALUES

('Chennai Super Kings', 'Chennai', 5),

('Mumbai Indians', 'Mumbai', 5),

('Royal Challengers Bangalore', 'Bangalore', 0),

('Kolkata Knight Riders', 'Kolkata', 3),

('Delhi Capitals', 'Delhi', 0);

INSERT INTO Franchise (franchise\_name, city, trophy\_count) VALUES

('Rajasthan Royals','Rajasthan',1),

('Punjab Kings','Punjab',0),

('Sunrisers Hyderabad','Hyderabad',1),

('Lucknow Supergiants','Lucknow',0),

('Gujart Titans', 'Gujart',1),

('Deccan Chargers','Hyderabad',1);

-- Insert data into Player Table

INSERT INTO Player (player\_name, role, contract\_amount, leadership\_role, franchise\_id) VALUES

('MS Dhoni', 'Wicketkeeper-Batsman', 4000000, 'None', 1),

('Virat Kohli', 'Batsman', 21000000, 'None', 3),

('Jasprit Bumrah ', 'Bowler', 18000000, 'ViceCaptain', 2),

('Andre Russell', 'All-rounder', 12000000, 'None', 4),

('KL Rahul', 'Wicketkeeper-Batsman', 16000000, 'Captain', 5);

INSERT INTO Player (player\_name, role, contract\_amount, leadership\_role, franchise\_id) VALUES

('Sanju Samson','Wicketkeeper-Batsman',18000000,'Captain',6),

('Arshdeep Singh', 'Bowler',18000000,'None','7'),

('Abhishek Sharma','Batsman',14000000,'None',8),

('Rishab Pant','Wicketkeeper-Batsman',27000000,'Captain',9),

('Jos Butler','Batsman',15750000,'ViceCaptain',10);

-- Insert data into Owner Table

INSERT INTO Owner (owner\_name, group\_name, city, franchise\_id) VALUES

('N. Srinivasan', 'India Cements', 'Chennai', 1),

('Mukesh Ambani', 'Reliance Industries', 'Mumbai', 2),

('United Spirits', 'United Spirits Group', 'Bangalore', 3),

('Shah Rukh Khan', 'Red Chillies Entertainment', 'Kolkata', 4),

('Parth Jindal', 'JSW Group', 'Delhi', 5);

INSERT INTO OWNER (OWNER\_NAME, GROUP\_NAME, CITY, FRANCHISE\_ID) VALUES

('Manoj Badale', 'Royal Multisport Pvt. Ltd.', 'Jaipur', 6),

('Preity Zinta', 'KPH Dream Cricket Pvt. Ltd.', 'Mohali', 7),

('Kalanithi Maran', 'Sun TV Network', 'Hyderabad', 8),

('Sanjiv Goenka', 'RPSG Group', 'Lucknow', 9),

('Torrent', 'Torrent Group', 'Ahmedabad', 10);

-- Insert data into Stadium Table

INSERT INTO Stadium (stadium\_name, city, stadium\_area, total\_seat, franchise\_id) VALUES

('M. A. Chidambaram Stadium', 'Chennai', 45000, 50000, 1),

('Wankhede Stadium', 'Mumbai', 40000, 55000, 2),

('M. Chinnaswamy Stadium', 'Bangalore', 42000, 48000, 3),

('Eden Gardens', 'Kolkata', 45000, 66000, 4),

('Arun Jaitley Stadium', 'Delhi', 35000, 41000, 5);

INSERT INTO Stadium (stadium\_name, city, stadium\_area, total\_seat, franchise\_id) VALUES

('Sawai Mansingh Stadium', 'Jaipur',25000,30000, 6), -- Rajasthan Royals

('Punjab Cricket Association IS Bindra Stadium', 'Mohali',40000, 27000, 7), -- Punjab Kings

('Rajiv Gandhi International Cricket Stadium', 'Hyderabad',35000, 55000, 8), -- Sunrisers Hyderabad

('BRSABV Ekana Cricket Stadium', 'Lucknow',47000, 50000, 9), -- Lucknow Super Giants

('Narendra Modi Stadium', 'Ahmedabad',100000, 132000, 10); -- Gujarat Titans

-- Insert data into Sponsor Table

INSERT INTO Sponsor (sponsor\_name) VALUES

('Etihad Airways'),

('Slice'),

('Qatar Airways'),

('SIX5SIX'),

('Ebix');

-- Insert data into SponsorFranchise Table (Many-to-Many)

INSERT INTO SponsorFranchise (sponsor\_id, franchise\_id, contract\_start\_date, contract\_end\_date, sponsorship\_amount) VALUES

(1, 1, '2023-01-01', '2025-12-31', 20000000),

(2, 2, '2022-01-01', '2024-12-31', 18000000),

(3, 3, '2023-05-01', '2026-04-30', 22000000),

(4, 4, '2021-06-01', '2025-05-31', 15000000),

(5, 5, '2022-03-01', '2024-10-31', 17000000);

-- Insert data into SupportingStaff Table

INSERT INTO SupportingStaff (staff\_name, role, franchise\_id) VALUES

('Stephen Fleming', 'Head Coach', 1),

('Mahela Jayawardene', 'Head Coach', 2),

('Sanjay Bangar', 'Batting Coach', 3),

('Brendon McCullum', 'Head Coach', 4),

('Ricky Ponting', 'Head Coach', 5);

INSERT INTO SupportingStaff (staff\_name, role, franchise\_id) VALUES

('Kumar Sangakkara', 'Head Coach', 6), -- Rajasthan Royals

('Trevor Bayliss', 'Head Coach', 7), -- Punjab Kings

('Daniel Vettori', 'Head Coach', 8), -- Sunrisers Hyderabad

('Justin Langer', 'Head Coach', 9), -- Lucknow Super Giants

('Ashish Nehra', 'Head Coach', 10), -- Gujarat Titans

('Lasith Malinga', 'Bowling Coach', 6),

('Charl Langeveldt', 'Bowling Coach', 7),

('Dale Steyn', 'Bowling Coach', 8),

('Morne Morkel', 'Bowling Coach', 9),

('Ryan Harris', 'Bowling Coach', 10),

('Amol Muzumdar', 'Batting Coach', 6),

('Wasim Jaffer', 'Batting Coach', 7),

('Brian Lara', 'Batting Coach', 8),

('Gary Kirsten', 'Batting Coach', 10);

--Insert data into winner table

INSERT INTO Winner(franchise\_id, winning\_year, captain\_name, player\_is\_active, runner\_up, man\_of\_match, player\_of\_tournament)

VALUES

(6, 2008, 'Shane Warne', FALSE, 'Chennai Super Kings', 'Yusuf Pathan', 'Shane Watson'),

(7, 2009, 'Adam Gilchrist', FALSE, 'Royal Challengers Bangalore', 'Anil Kumble', 'Adam Gilchrist'),

(1, 2010, 'MS Dhoni', TRUE, 'Mumbai Indians', 'Suresh Raina', 'Sachin Tendulkar'),

(1, 2011, 'MS Dhoni', TRUE, 'Royal Challengers Bangalore', 'Murali Vijay', 'Chris Gayle'),

(4, 2012, 'Gautam Gambhir', FALSE, 'Chennai Super Kings', 'Manvinder Bisla', 'Sunil Narine'),

(2, 2013, 'Rohit Sharma', TRUE, 'Chennai Super Kings', 'Kieron Pollard', 'Shane Watson'),

(4, 2014, 'Gautam Gambhir', FALSE, 'Kings XI Punjab', 'Manish Pandey', 'Glenn Maxwell'),

(2, 2015, 'Rohit Sharma', TRUE, 'Chennai Super Kings', 'Rohit Sharma', 'Andre Russell'),

(8, 2016, 'David Warner', FALSE, 'Royal Challengers Bangalore', 'Ben Cutting', 'Virat Kohli'),

(2, 2017, 'Rohit Sharma', TRUE, 'Rising Pune Supergiants', 'Krunal Pandya', 'Ben Stokes'),

(1, 2018, 'MS Dhoni', TRUE, 'Sunrisers Hyderabad', 'Shane Watson', 'Sunil Narine'),

(2, 2019, 'Rohit Sharma', TRUE, 'Chennai Super Kings', 'Jasprit Bumrah', 'Andre Russell'),

(2, 2020, 'Rohit Sharma', TRUE, 'Delhi Capitals', 'Trent Boult', 'Jofra Archer'),

(1, 2021, 'MS Dhoni', TRUE, 'Kolkata Knight Riders', 'Faf du Plessis', 'Harshal Patel'),

(10, 2022, 'Hardik Pandya', TRUE, 'Rajasthan Royals', 'Hardik Pandya', 'Jos Buttler'),

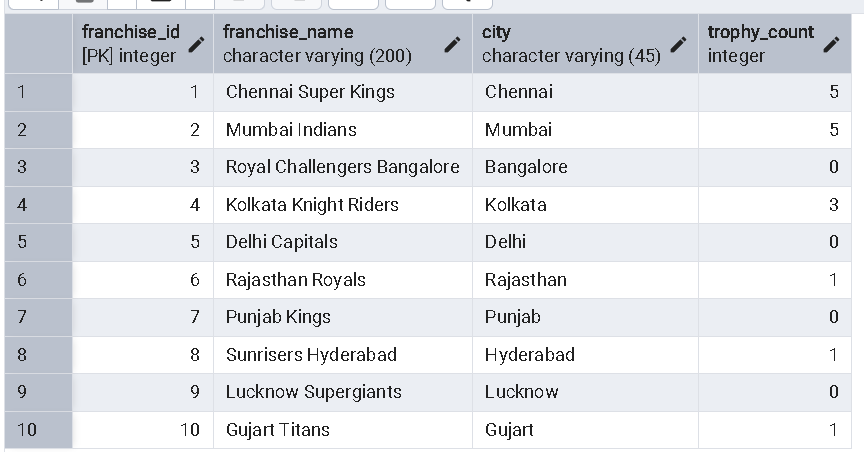
(1, 2023, 'MS Dhoni', TRUE, 'Gujarat Titans', 'Devon Conway', 'Shubman Gill'),

(4, 2024, 'Shreyas Iyer', TRUE, 'Sunrisers Hyderabad', 'Mitchell Starc', 'Sunil Narine');

**//VIEWING TABLE:-**

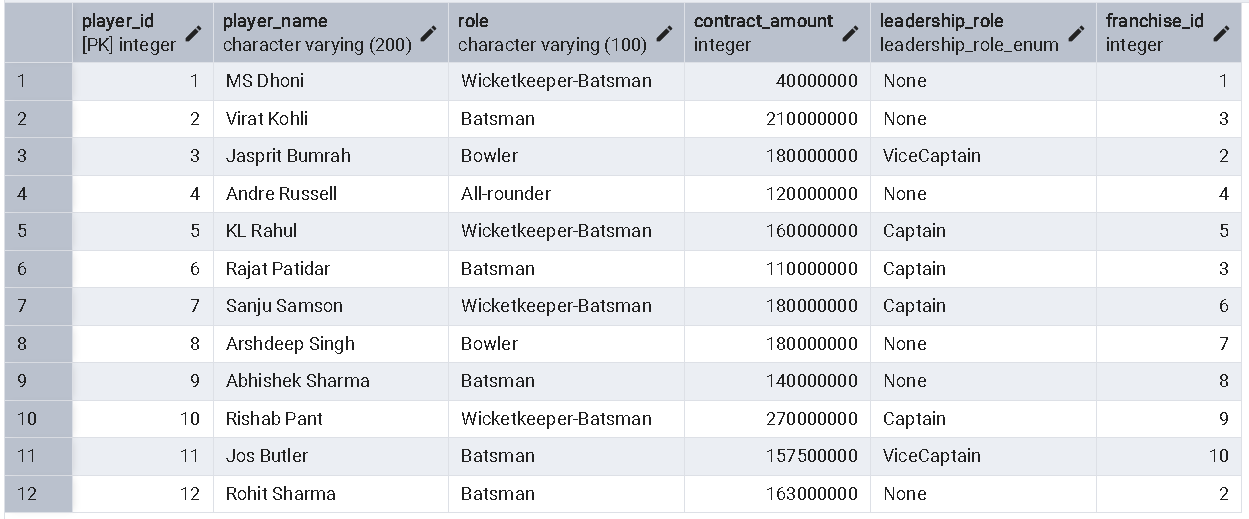
**Franchise Table = It is used to store information about franchises, their city and trophy count.**

**select \* from Franchise order by franchise\_id limit 10;**

****

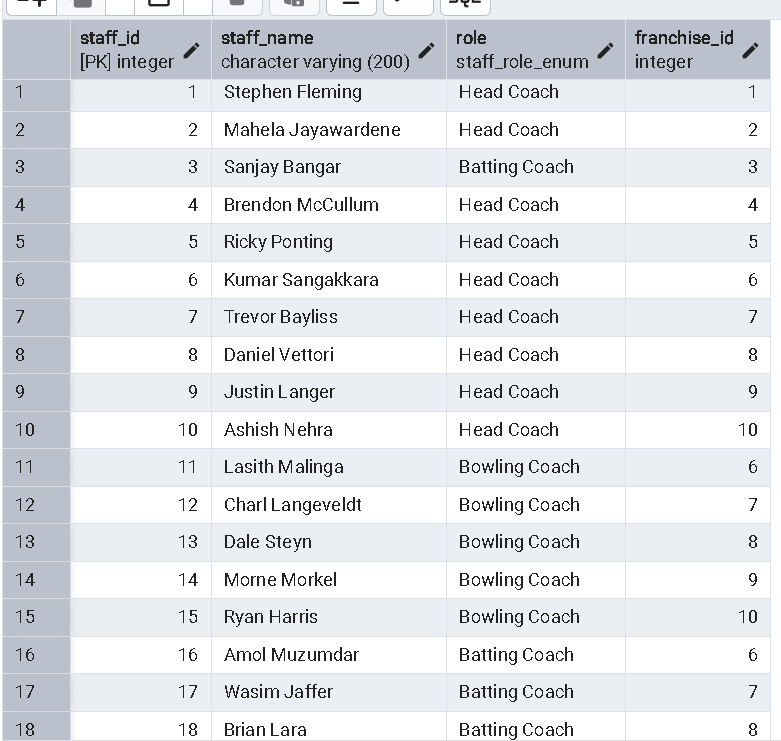
**Player table = It is used to stored information about player , its role and contract amount.**

**select \* from player order by player\_id;**

****

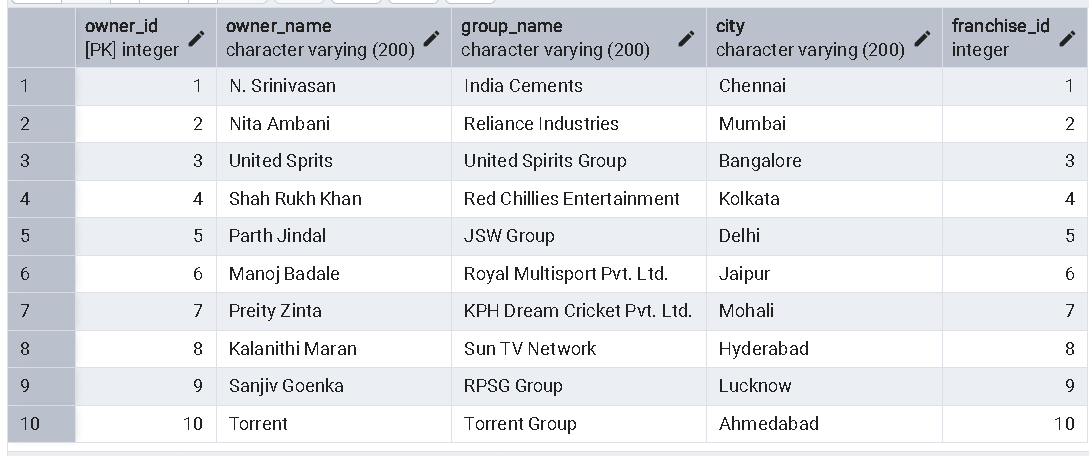
**Staff table = It is used to store info about supporting staff and their role in team.**

**select \* from Supportingstaff order by staff\_id;**

****

**Owner table = It is used to store information about their owner and owner group.**

**select \* From Owner order by owner\_id;**

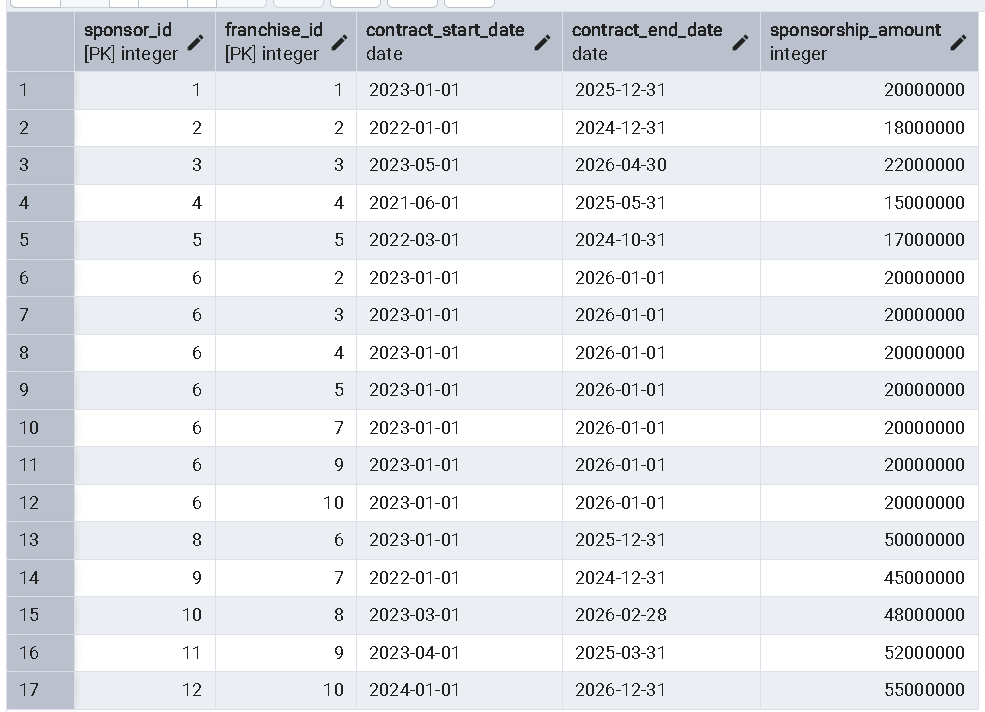
****

**Sponsor table = It is used to store info about sponsor of each franchise and their sponsorship amount for each franchise.**

**select \* From sponsor order by sponsor\_id;**

****

**select \* from SponsorFranchise order by sponsor\_id;**

****

**Stadium table = It is used to store information about each stadium of particular franchises.**

**select \* from Stadium order by stadium\_id;**

****

**Winner table = It is used to store info about Ipl winning team , runner up , man of match and player of tournament of that ipl year.**

**select \* from Winner order by winner\_id;**

****

**//QUERIES:-**

**-- 1. Group players by their franchise id --**

**SELECT f.franchise\_name,f.franchise\_id, STRING\_AGG(p.player\_name , ' , ') AS players**

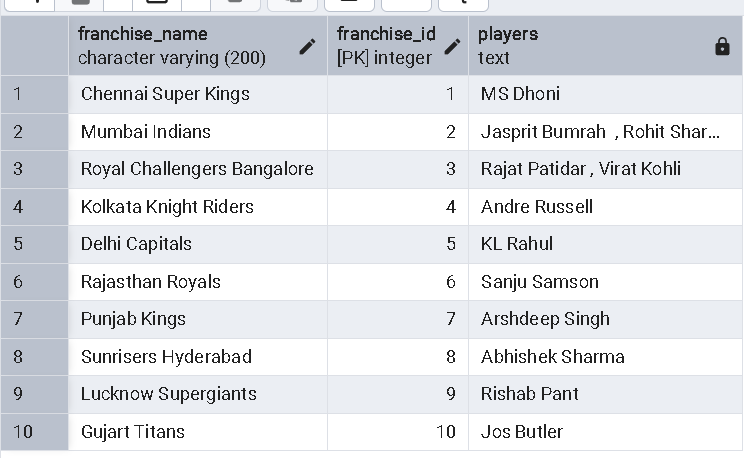
**FROM Franchise f**

**LEFT JOIN Player p ON f.franchise\_id = p.franchise\_id**

**GROUP BY f.franchise\_name, f.franchise\_id**

**ORDER BY FRANCHISE\_id**

**limit 10;**

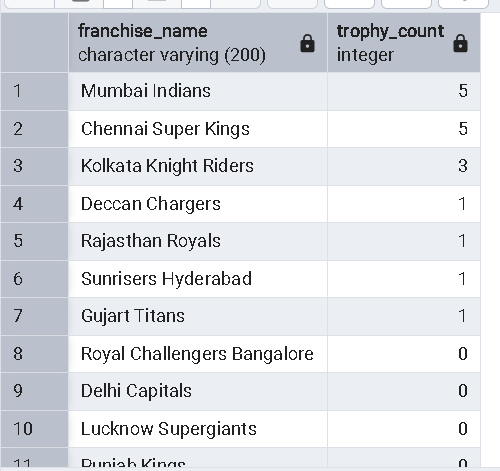
****

**-- 2. Total Trophies Won by Each Franchise--**

**SELECT franchise\_name, trophy\_count**

**FROM Franchise**

**ORDER BY trophy\_count DESC;**

****

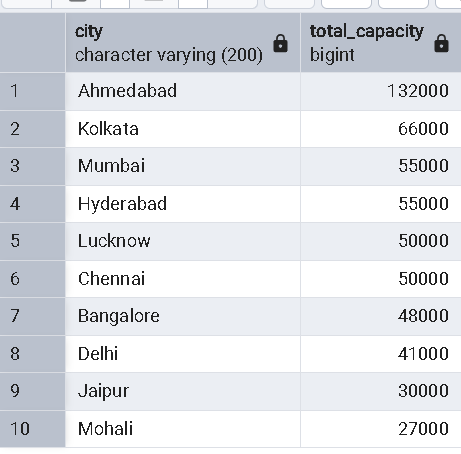
**--3. Total Seating Capacity Per City of Stadium--**

SELECT city, sum(total\_seat) AS total\_capacity

FROM Stadium

GROUP BY city

ORDER BY total\_capacity DESC;



**--4. Total Players Owned by Each Franchise--**

SELECT f.franchise\_name,f.franchise\_id, COUNT(p.player\_id) AS total\_players

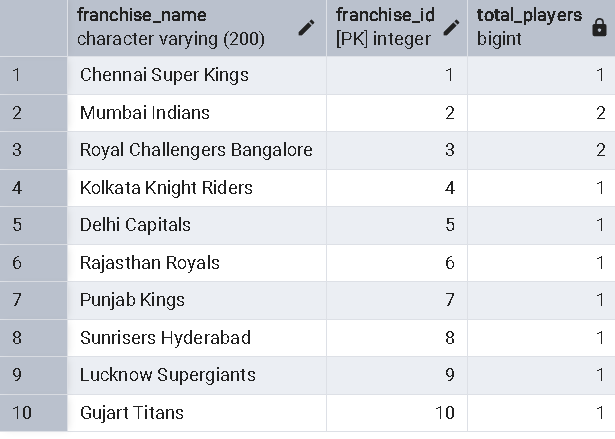
FROM Franchise f

LEFT JOIN Player p ON f.franchise\_id = p.franchise\_id

GROUP BY f.franchise\_name,f.franchise\_id

ORDER BY franchise\_id

limit 10;



**--5. Total Owner of each Franchise--**

SELECT f.franchise\_name,f.franchise\_id, COUNT(o.owner\_id) AS total\_owners

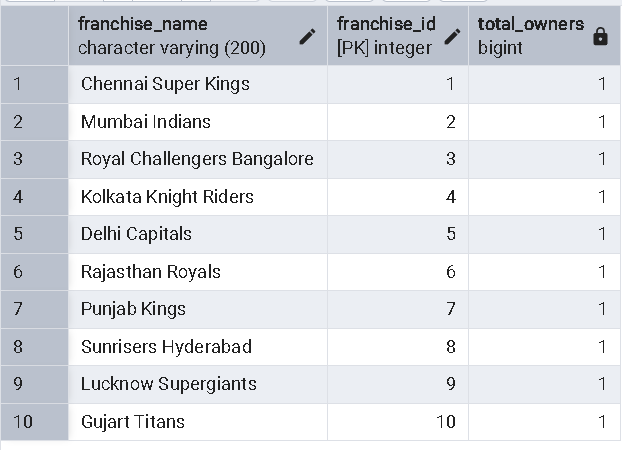
FROM Franchise f

LEFT JOIN Owner o ON f.franchise\_id = o.franchise\_id

GROUP BY f.franchise\_name,f.franchise\_id

ORDER BY f.franchise\_id

limit 10;



**--6. Total amount spend by each franchise--**

select f.franchise\_name,f.franchise\_id, sum(p.contract\_amount) as total\_amount

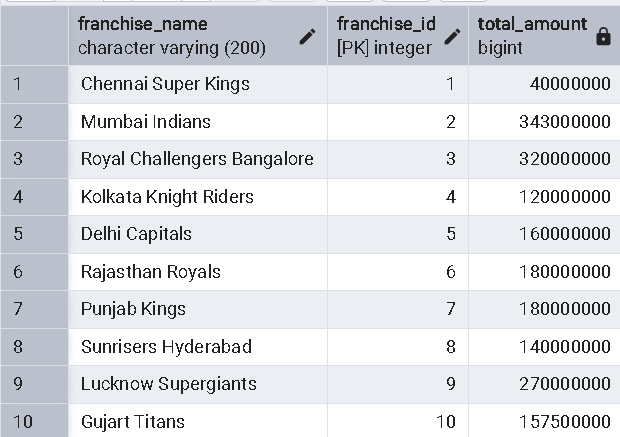
from franchise f

join player p on f.franchise\_id = p.franchise\_id

group by f.franchise\_name,f.franchise\_id

order by franchise\_id

limit 10;



**--7. Find the Most Valuable Players Based on Contract Amount--**

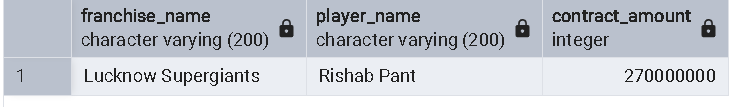
SELECT f.franchise\_name, p.player\_name, p.contract\_amount

FROM Franchise f

JOIN Player p ON f.franchise\_id = p.franchise\_id

ORDER BY p.contract\_amount DESC

LIMIT 1;



**-- Find all Players Who Are Vice-Captains--**

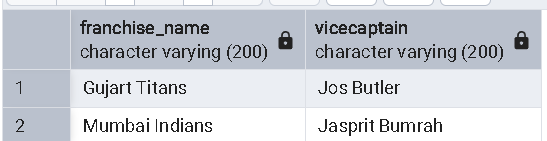
SELECT f.franchise\_name, p.player\_name as vicecaptain

FROM Franchise f

JOIN Player p ON f.franchise\_id = p.franchise\_id

WHERE p.leadership\_role = 'ViceCaptain'

order by franchise\_name;



**--8. Find All Supporting Staff of Each Franchise--**

SELECT f.franchise\_name,f.franchise\_id, ARRAY\_AGG(s.staff\_name) AS staff\_members

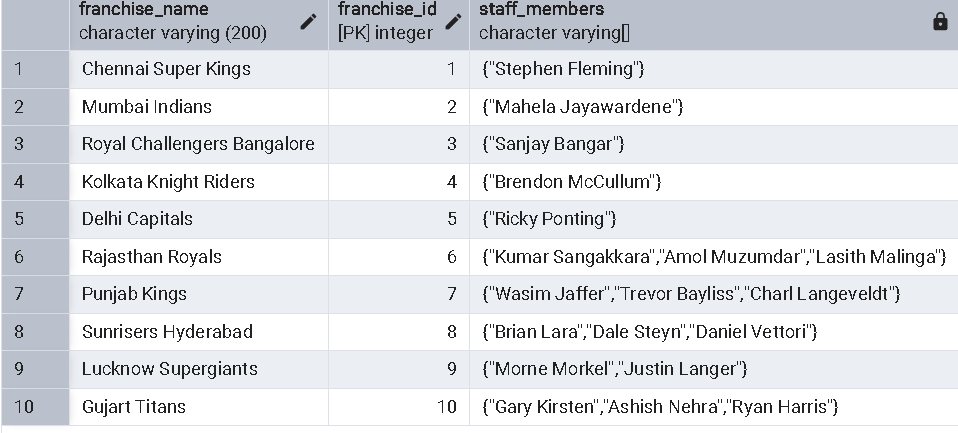
FROM Franchise f

LEFT JOIN SupportingStaff s ON f.franchise\_id = s.franchise\_id

GROUP BY f.franchise\_name,f.franchise\_id

order by franchise\_id

limit 10;



**--9. Find Stadiums Linked to Each Franchise--**

SELECT f.franchise\_name,f.franchise\_id, ARRAY\_AGG(s.stadium\_name) AS stadiums

FROM Franchise f

LEFT JOIN Stadium s ON f.franchise\_id = s.franchise\_id

GROUP BY f.franchise\_id,f.franchise\_name

order by f.franchise\_id;



**--10. Find All Owners Linked to Each Franchise--**

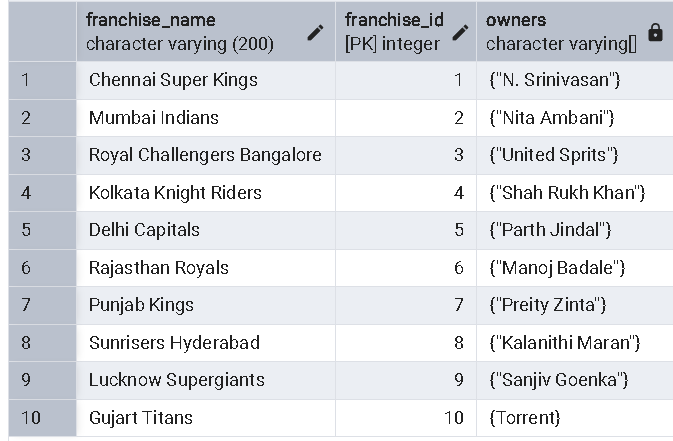
SELECT f.franchise\_name,f.franchise\_id, ARRAY\_AGG(o.owner\_name) AS owners

FROM Franchise f

LEFT JOIN Owner o ON f.franchise\_id = o.franchise\_id

GROUP BY f.franchise\_name,f.franchise\_id

order by franchise\_id;



**--11. Finding sponsor for each franchise--**

SELECT f.franchise\_name, s.sponsor\_name, sf.contract\_start\_date, sf.contract\_end\_date, sf.sponsorship\_amount

FROM SponsorFranchise sf

JOIN Franchise f ON sf.franchise\_id = f.franchise\_id

JOIN Sponsor s ON sf.sponsor\_id = s.sponsor\_id

ORDER BY f.franchise\_name;



**--12. Finding all batsman in each team**

SELECT f.franchise\_name,

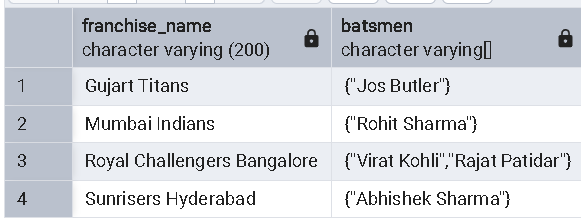
ARRAY\_AGG(p.player\_name) AS batsmen

FROM Franchise f

JOIN Player p ON f.franchise\_id = p.franchise\_id

WHERE p.role = 'Batsman'

GROUP BY f.franchise\_name;



**--13. Finding all bowler in each team**

SELECT f.franchise\_name,

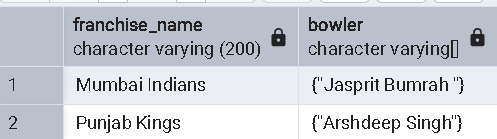
ARRAY\_AGG(p.player\_name) AS bowler

FROM Franchise f

JOIN Player p ON f.franchise\_id = p.franchise\_id

WHERE p.role = 'Bowler'

GROUP BY f.franchise\_name;



**--14. Finding all wickerkeeper in each team**

SELECT f.franchise\_name,

ARRAY\_AGG(p.player\_name) AS wicketkeeper

FROM Franchise f

JOIN Player p ON f.franchise\_id = p.franchise\_id

WHERE p.role = 'Wicketkeeper-Batsman'

GROUP BY f.franchise\_name;



**--15. Find all allrounder in each team--**

SELECT f.franchise\_name,

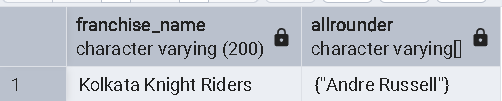
ARRAY\_AGG(p.player\_name) AS allrounder

FROM Franchise f

JOIN Player p ON f.franchise\_id = p.franchise\_id

WHERE p.role = 'All-rounder'

GROUP BY f.franchise\_name;



**--16. Finding teams that won ipl more than two time--**

select \* from winner;

SELECT f.franchise\_name,COUNT(w.winner\_id) AS total\_wins

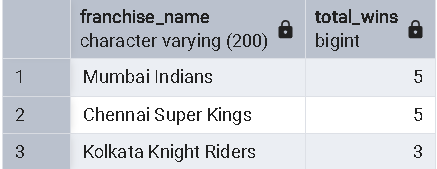
FROM Winner w

JOIN Franchise f ON w.franchise\_id = f.franchise\_id

GROUP BY f.franchise\_name

HAVING COUNT(w.winner\_id) > 2

ORDER BY total\_wins DESC;



**--17. Finding all captain who have won most ipl titles--**

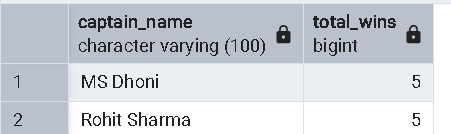
SELECT captain\_name, COUNT(winner\_id) AS total\_wins

FROM Winner

GROUP BY captain\_name

ORDER BY total\_wins DESC

Limit 2;



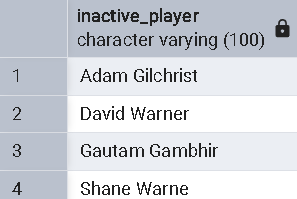
**--18. Finding all captain who is not playing ipl--**

Select distinct Captain\_name as inactive\_player

from winner

where player\_is\_active = False

order by captain\_name;



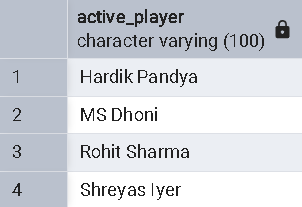
**--19. Finding all captain who is still playing ipl--**

Select distinct (captain\_name) as active\_player

from winner

where player\_is\_active = true

order by captain\_name;



**--20. Total IPL title wins by each franchise--**

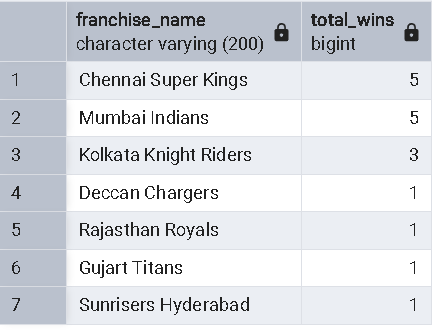
Select f.franchise\_name , count(w.winner\_id) as total\_wins

from winner w

join Franchise f on w.franchise\_id = f.franchise\_id

group by f.franchise\_name

order by total\_wins desc;



**--21. Stored procedure for inserting player--**

create or replace procedure inserting\_player(

p\_player\_name varchar,

p\_role varchar,

p\_contract\_amount int,

p\_leadership\_role leadership\_role\_enum,

p\_franchise\_id int)

language plpgsql

AS $$

begin

insert into player(player\_name,role,contract\_amount,leadership\_role,franchise\_id)

values(p\_player\_name,p\_role,p\_contract\_amount,p\_leadership\_role,p\_franchise\_id);

end;

$$;

select \* from player;

call inserting\_player('Rohit Sharma','Batsman',163000000,'None',2);

**--21. Stored procedure for inserting stadium--**

create or replace procedure inserting\_stadium(

p\_stadium\_name varchar,

p\_city varchar,

p\_stadium\_area int,

p\_total\_seat int,

p\_franchise\_id int)

language plpgsql

AS $$

begin

insert into stadium(stadium\_name,city,stadium\_area,total\_seat,franchise\_id)

values(p\_stadium\_name,p\_city,p\_stadium\_area,p\_total\_seat,p\_franchise\_id);

end;

$$;

call inserting\_stadium();

**--22.Stored procedure for updating player--**

CREATE OR REPLACE PROCEDURE updating\_player(

p\_player\_id INT,

p\_player\_name VARCHAR DEFAULT NULL,

p\_role VARCHAR DEFAULT NULL,

p\_contract\_amount INT DEFAULT NULL,

p\_leadership\_role leadership\_role\_enum DEFAULT NULL,

p\_franchise\_id INT DEFAULT NULL

)

LANGUAGE plpgsql

AS $$

BEGIN

UPDATE player

SET

player\_name = COALESCE(p\_player\_name, player\_name),

role = COALESCE(p\_role, role),

contract\_amount = COALESCE(p\_contract\_amount, contract\_amount),

leadership\_role = COALESCE(p\_leadership\_role, leadership\_role),

franchise\_id = COALESCE(p\_franchise\_id, franchise\_id)

WHERE player\_id = p\_player\_id;

END;

$$;

**--23. Updating stadium using stored procedure--**

CREATE OR REPLACE PROCEDURE update\_stadium(

p\_stadium\_id INT,

p\_stadium\_name VARCHAR DEFAULT NULL,

p\_city VARCHAR DEFAULT NULL,

p\_stadium\_area INT DEFAULT NULL,

p\_total\_seat INT DEFAULT NULL,

p\_franchise\_id INT DEFAULT NULL

)

LANGUAGE plpgsql

AS $$

BEGIN

UPDATE Stadium

SET

stadium\_name = COALESCE(p\_stadium\_name, stadium\_name),

city = COALESCE(p\_city, city),

stadium\_area = COALESCE(p\_stadium\_area, stadium\_area),

total\_seat = COALESCE(p\_total\_seat, total\_seat),

franchise\_id = COALESCE(p\_franchise\_id, franchise\_id)

WHERE stadium\_id = p\_stadium\_id;

END;

$$;

CALL updating\_player(12,null,null,null,'Captain'::leadership\_role\_enum,null);

Call updating\_player(12,null,null,null,'None',null);

**--24. Stored procedure for deleting player--**

create or replace procedure deleting\_player(p\_player\_id int)

language plpgsql

as $$

begin

delete from player where player\_id = p\_player\_id;

end;

$$;

select \* from player;

select \* from stadium;

call deleting\_player(12);

**--25. Deleting stadium using stored procedure--**

CREATE OR REPLACE PROCEDURE delete\_stadium(p\_stadium\_id INT)

LANGUAGE plpgsql

AS $$

BEGIN

DELETE FROM Stadium WHERE stadium\_id = p\_stadium\_id;

END;

$$;

**--26. function for getting trophy count for each franchise--**

create or replace function trophy\_count(f\_franchise\_id int)

returns table(franchise\_id int,franchise\_name varchar,city varchar, trophy\_count int)

language plpgsql

As $$

Begin

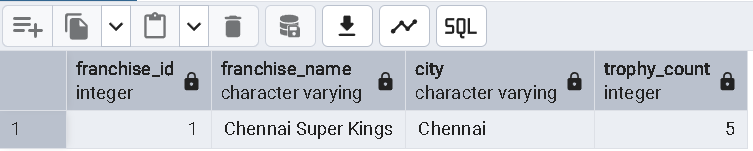
Return query

Select f.franchise\_id,f.franchise\_name,f.city,f.trophy\_count from franchise f where f.franchise\_id = f\_franchise\_id;

End;

$$;

select \* from trophy\_count(1);



**--27. Stored procedure for getting trophies for all franchise:-**

CREATE OR REPLACE PROCEDURE get\_franchises\_by\_trophies()

LANGUAGE plpgsql

AS $$

BEGIN

-- Fetch franchise name and trophy count in descending order

EXECUTE 'SELECT franchise\_name, trophy\_count FROM Franchise ORDER BY trophy\_count DESC';

END;

$$;

**--28. function for getting captain for each franchise--**

CREATE OR REPLACE FUNCTION get\_captains\_for\_each\_franchise()

RETURNS TABLE (franchise\_id INT, franchise\_name VARCHAR, captain\_name VARCHAR)

LANGUAGE plpgsql

AS $$

BEGIN

RETURN QUERY

SELECT

f.franchise\_id,

f.franchise\_name,

p.player\_name AS captain\_name

FROM Franchise f

JOIN Player p ON f.franchise\_id = p.franchise\_id

WHERE p.leadership\_role = 'Captain'

ORDER BY f.franchise\_id;

END;

$$;

select \* from get\_captains\_for\_each\_franchise();



**29. Stored Function to Get Total Income for Each Franchise**

CREATE OR REPLACE FUNCTION get\_total\_income\_for\_each\_franchise()

RETURNS TABLE (franchise\_id INT, franchise\_name VARCHAR, total\_income BIGINT)

LANGUAGE plpgsql

AS $$

BEGIN

RETURN QUERY

SELECT

f.franchise\_id,

f.franchise\_name,

COALESCE(SUM(sf.sponsorship\_amount), 0) AS total\_income

FROM Franchise f

LEFT JOIN SponsorFranchise sf ON f.franchise\_id = sf.franchise\_id

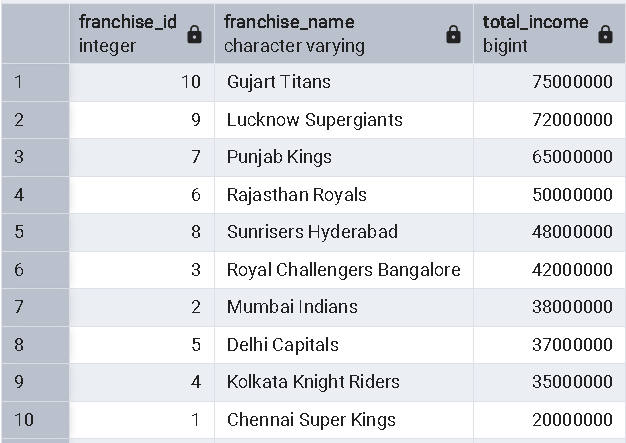
GROUP BY f.franchise\_id, f.franchise\_name

ORDER BY total\_income DESC;

END;

$$;

SELECT \* FROM get\_total\_income\_for\_each\_franchise();



--**30. Function for total people in each franchise—**

**CREATE OR REPLACE FUNCTION get\_total\_people\_in\_franchise()**

**RETURNS TABLE (**

**franchise\_id INT,**

**franchise\_name VARCHAR,**

**total\_players INT,**

**total\_staff INT,**

**total\_people INT**

**)**

**LANGUAGE plpgsql**

**AS $$**

**BEGIN**

**RETURN QUERY**

**SELECT**

**f.franchise\_id,**

**f.franchise\_name,**

**COALESCE(CAST(COUNT(DISTINCT p.player\_id) AS INT), 0) AS total\_players,**

**COALESCE(CAST(COUNT(DISTINCT s.staff\_id) AS INT), 0) AS total\_staff,**

**COALESCE(CAST(COUNT(DISTINCT p.player\_id) AS INT), 0) + COALESCE(CAST(COUNT(DISTINCT s.staff\_id) AS INT), 0) AS total\_people**

**FROM Franchise f**

**LEFT JOIN Player p ON f.franchise\_id = p.franchise\_id**

**LEFT JOIN SupportingStaff s ON f.franchise\_id = s.franchise\_id**

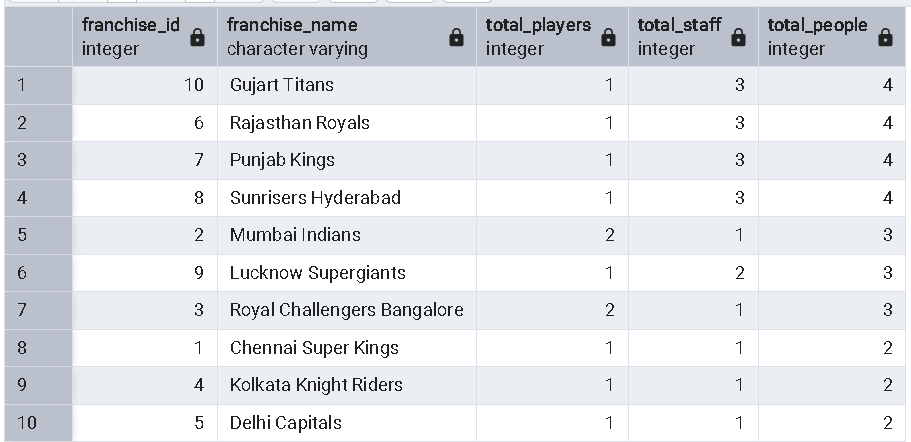
**GROUP BY f.franchise\_id, f.franchise\_name**

**ORDER BY total\_people DESC;**

**END;**

**$$;**

**SELECT \* FROM get\_total\_people\_in\_franchise();**

****