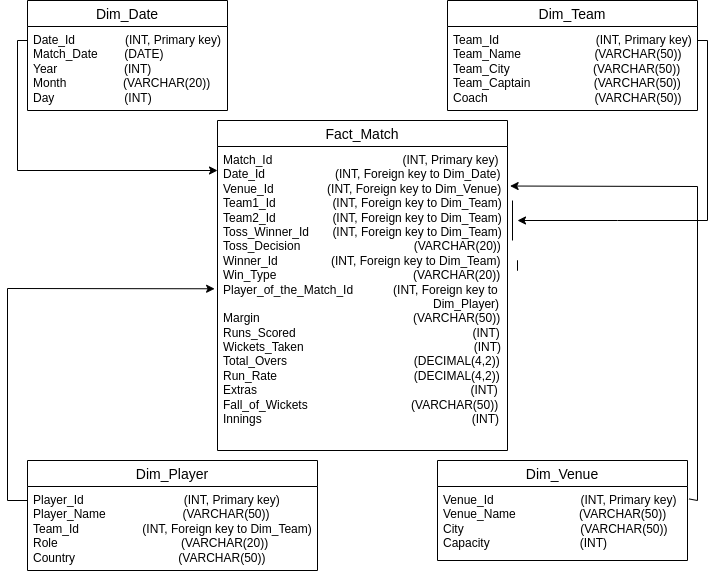
**Q1. IPL tournament**

Ans: DWH diagram (Star Schema)



**List of Dimensions and their definition:**

**All the dimensions shall be maintained in SCD-Type1 fashion**

1. **Dim\_Date Table:- This table would contain the date of each IPL match**

CREATE TABLE Dim\_Date (

Date\_Id INT PRIMARY KEY,

Match\_Date DATE,

Year INT,

Month VARCHAR(20),

Day INT

);

1. **Dim\_Team Table:- This table would contain information about the teams that play in IPL**

CREATE TABLE Dim\_Team (

Team\_Id INT PRIMARY KEY,

Team\_Name VARCHAR(50),

Team\_City VARCHAR(50),

Team\_Captain VARCHAR(50),

Coach VARCHAR(50)

);

1. **Dim\_Player Table:- This table would contain information about the players in IPL**

CREATE TABLE Dim\_Player (

Player\_Id INT PRIMARY KEY,

Player\_Name VARCHAR(50),

Team\_Id INT,

Role VARCHAR(20),

Country VARCHAR(50),

FOREIGN KEY (Team\_Id) REFERENCES Dim\_Team(Team\_Id)

);

1. **Dim\_Venue Table:- This table would contain information about the venues where IPL matches are played**

CREATE TABLE Dim\_Venue (

Venue\_Id INT PRIMARY KEY,

Venue\_Name VARCHAR(50),

City VARCHAR(50),

Capacity INT

);

1. **Fact\_Match Table:- This table would contain information about each IPL match**

CREATE TABLE Fact\_Match (

Match\_Id INT PRIMARY KEY,

Date\_Id INT,

Venue\_Id INT,

Team1\_Id INT,

Team2\_Id INT,

Toss\_Winner\_Id INT,

Toss\_Decision VARCHAR(20),

Winner\_Id INT,

Win\_Type VARCHAR(20),

Player\_of\_the\_Match\_Id INT,

Margin VARCHAR(50),

Runs\_Scored INT,

Wickets\_Taken INT,

Total\_Overs DECIMAL(4,2),

Run\_Rate DECIMAL(4,2),

Extras INT,

Fall\_of\_Wickets VARCHAR(50),

Innings INT,

FOREIGN KEY (Date\_Id) REFERENCES Dim\_Date(Date\_Id),

FOREIGN KEY (Venue\_Id) REFERENCES Dim\_Venue(Venue\_Id),

FOREIGN KEY (Team1\_Id) REFERENCES Dim\_Team(Team\_Id),

FOREIGN KEY (Team2\_Id) REFERENCES Dim\_Team(Team\_Id),

FOREIGN KEY (Toss\_Winner\_Id) REFERENCES Dim\_Team(Team\_Id),

FOREIGN KEY (Winner\_Id) REFERENCES Dim\_Team(Team\_Id),

FOREIGN KEY (Player\_of\_the\_Match\_Id) REFERENCES Dim\_Player(Player\_Id)

);

The Fact\_Match table would also contain columns for various match statistics such as runs scored, wickets taken, and other performance metrics.

SQL queries to generate insightful business metrics:

1. **Total matches played in each year:**

SELECT year, COUNT(\*) AS num\_matches

FROM Dim\_Date d

JOIN Fact\_Match f ON d.Date\_Id = f.Date\_Id

GROUP BY year;

1. **Total runs scored by each team in each year:**

SELECT year, t.Team\_Name, SUM(runs) AS total\_runs

FROM Dim\_Team t

JOIN Fact\_Match f ON t.Team\_Id = f.Team1\_Id OR t.Team\_Id = f.Team2\_Id

JOIN (SELECT Match\_Id, SUM(runs\_scored) AS runs

FROM Fact\_Match

GROUP BY Match\_Id) fm ON f.Match\_Id = fm.Match\_Id

JOIN Dim\_Date d ON f.Date\_Id = d.Date\_Id

GROUP BY year, t.Team\_Name;

1. **Top 10 players with the highest batting average:**

SELECT p.Player\_Name, AVG(batting\_average) AS avg\_batting\_avg

FROM Dim\_Player p

JOIN (SELECT Player\_Id, SUM(runs\_scored) AS total\_runs, COUNT(\*) AS num\_innings,

SUM(CASE WHEN out\_type IS NULL THEN 0 ELSE 1 END) AS num\_outs,

(SUM(runs\_scored) / (CASE WHEN SUM(CASE WHEN out\_type IS NULL THEN 0 ELSE 1 END) = 0 THEN 1 ELSE SUM(CASE WHEN out\_type IS NULL THEN 0 ELSE 1 END) END)) AS batting\_average

FROM Fact\_Match

GROUP BY Player\_Id) b ON p.Player\_Id = b.Player\_Id

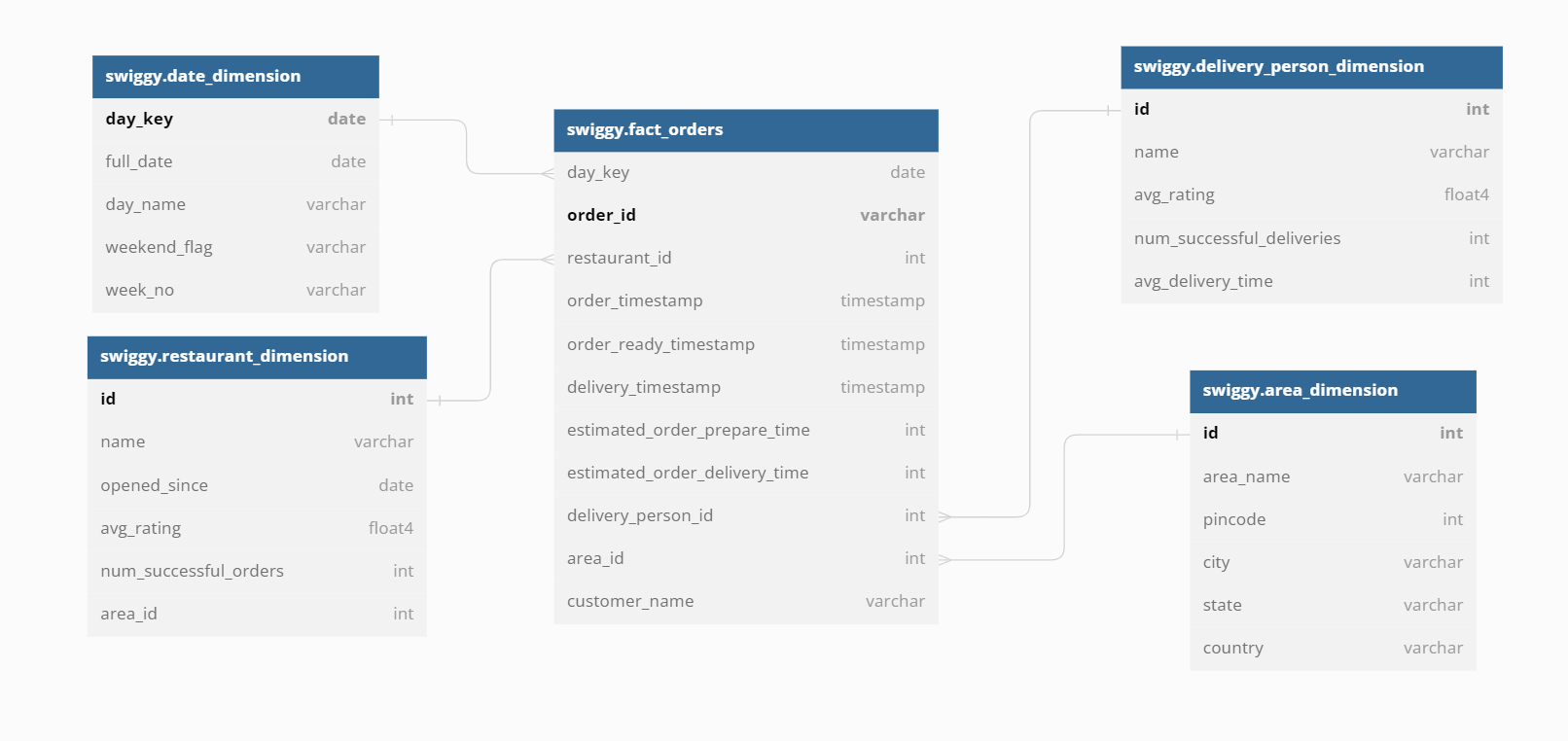
GROUP BY p.Player\_Name

ORDER BY avg\_batting\_avg DESC

LIMIT 10;

**Q2. Swiggy DWH**

Ans: DWH diagram (Star Schema)



**List of Dimensions and their definition:**

**All the dimensions shall be maintained in SCD-Type1 fashion**

1. **swiggy.restaurant\_dimension**

swiggy.restaurant\_dimension {

id int [pk]

name varchar

opened\_since date

avg\_rating float4

num\_successful\_orders int

area\_id int

}

1. **swiggy.area\_dimension**

swiggy.area\_dimension {

id int [pk]

area\_name varchar

pincode int

city varchar

state varchar

country varchar

}

1. **swiggy.delivery\_person\_dimension**

swiggy.delivery\_person\_dimension {

id int [pk]

name varchar

avg\_rating float4

num\_successful\_deliveries int

avg\_delivery\_time int

}

1. **swiggy.date\_dimension**

swiggy.date\_dimension {

day\_key date [pk]

full\_date date

day\_name varchar

weekend\_flag varchar

week\_no varchar

}

1. **swiggy.fact\_orders**

swiggy.fact\_orders {

day\_key date [ref: > swiggy.date\_dimension.day\_key]

order\_id varchar [pk]

restaurant\_id int [ref: > swiggy.restaurant\_dimension.id]

order\_timestamp timestamp

order\_ready\_timestamp timestamp

delivery\_timestamp timestamp

estimated\_order\_prepare\_time int

estimated\_order\_delivery\_time int

delivery\_person\_id int [ref: > swiggy.delivery\_person\_dimension.id]

area\_id int [ref: > swiggy.area\_dimension.id]

customer\_name varchar

}

SQL queries to generate insightful business metrics:

1. **What is the area wise percentage of orders delivered on time?**

with cte as

(

select a.area\_name, a.pincode, count(\*) ontime\_deliveries from swiggy.area\_dim a

inner join

swiggy.fact\_orders f

on a.id= f.area\_id

where TIMESTAMPADD(MINUTE,estimated\_order\_delivery\_time,order\_timestamp) <= order\_timestamp

group by a.area\_name, a.pincode

),

cte1 as

(

select a.area\_name, a.pincode, count(\*) total\_deliveries from swiggy.area\_dim a

inner join

swiggy.fact\_orders f

on a.id= f.area\_id

group by a.area\_name, a.pincode

)

select cte.area\_name,cte.pincode, (ontime\_deliveries\*100/total\_deliveries) area\_wise\_percentage

from cte

inner join cte1

on cte.area\_name=cte1.area\_name and cte.pincode=cte1.pincode;

**2. What is the area from which most orders are received?**

with cte as

(

select a.area\_name, a.pincode, count(\*) total\_deliveries from swiggy.area\_dim a

inner join

swiggy.fact\_orders f

on a.id= f.area\_id

group by a.area\_name, a.pincode

)

Select \* from cte

Order by total\_deliveries desc limit 1;

3.**Who’s fault was there in a particular area for delayed deliveries (restaurant or delivery person)?**

select a.area\_name, a.pincode,

case

when TIMESTAMPADD(MINUTE, estimated\_order\_prepare\_time,order\_timestamp) > order\_ready\_timestamp then ‘Restaurant’s fault’

when TIMESTAMPADD(MINUTE, estimated\_order\_delivery\_time,order\_timestamp) > delivery\_timestamp then ‘Delivery person’s fault’

end Delay\_Reason

from swiggy.fact\_orders f

inner join

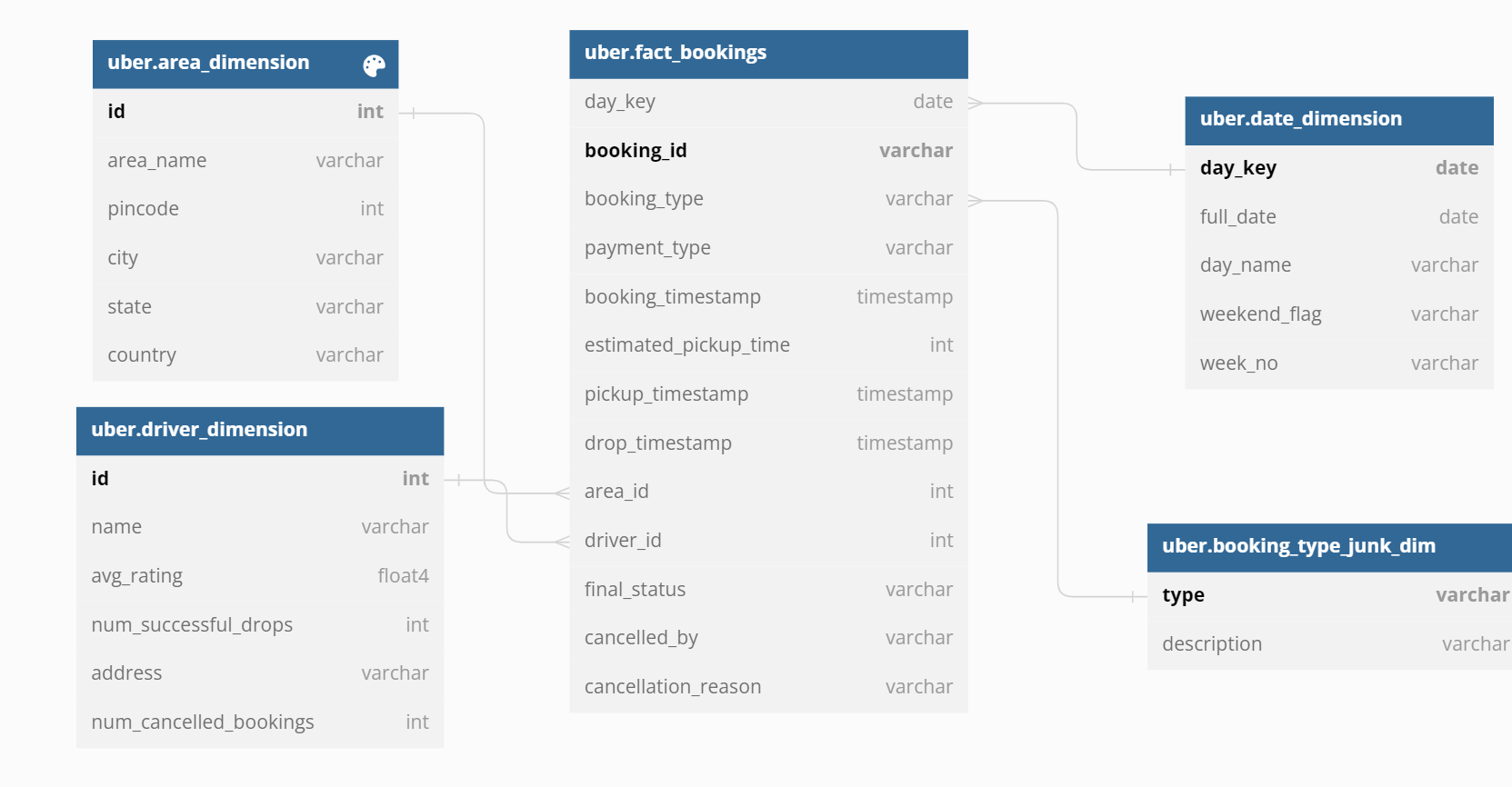
swiggy.area\_dim a

on a.id= f.area\_id

where a.area\_name=’Kondapur’ and a.pincode=500084

**Q3. Uber DWH**

Ans: **DWH diagram (Star Schema)**



**List of Dimensions and their definition:**

**All the dimensions shall be maintained in SCD-Type1 fashion**

Table uber.area\_dimension {

id int [pk]

area\_name varchar

pincode int

city varchar

state varchar

country varchar

}

Table uber.driver\_dimension {

id int [pk]

name varchar

avg\_rating float4

num\_successful\_drops int

address varchar

num\_cancelled\_bookings int

}

Table uber.date\_dimension {

day\_key date [pk]

full\_date date

day\_name varchar

weekend\_flag varchar

week\_no varchar

}

**Facts and their definition:**

Table uber.fact\_bookings {

day\_key date [ref: > uber.date\_dimension.day\_key]

booking\_id varchar [pk]

booking\_type varchar

payment\_type varchar

booking\_timestamp timestamp

estimated\_pickup\_time int

pickup\_timestamp timestamp

drop\_timestamp timestamp

area\_id int [ref: > uber.area\_dimension.id]

driver\_id int [ref: > uber.driver\_dimension.id]

final\_status varchar

cancelled\_by varchar

cancellation\_reason varchar

}

**Insightful reports:**

1. **What is the area wise count of ride bookings on a particular day?**

select count(\*) num\_of\_rides,f.booking\_type , a.area\_name, a.city, a.state from

uber.fact\_bookings f left join uber.area\_dimension a

on f.area\_id=a.id

where f.day\_key='2022-01-05'

group by f.booking\_type , a.area\_name, a.city, a.state;

1. **Percentage of rides which have abnormally long pickup time daywise?**

with cte as

(

select f.day\_key, count(\*) delayed\_pickup\_num from uber.fact\_orders f

inner join

uber.area\_dim a

on a.id= f.area\_id

where TIMESTAMPADD(MINUTE,estimated\_pickup\_time,booking\_timestamp)

> pickup\_timestamp group by f.day\_key

),

cte1 as

(

select f.day\_key, count(\*) total\_drops from uber.fact\_orders f

inner join

uber.area\_dim a

on a.id= f.area\_id

where f.final\_status='Dropped successfully'

group by f.day\_key

)

select cte.day\_key, (cte.delayed\_pickup\_num\*100/cte1.total\_drops)

day\_wise\_percentage

from cte

inner join cte1

on cte.day\_key=cte1.day\_key;

1. **Daywise, Areawise, peak time of the day (in terms of hour of the day)?**

with cte as

(

select f.day\_key, hour(f.booking\_timestamp) hour\_of\_booking

a.area\_name, a.city, a.state,

count(\*) num\_of\_rides,

from

uber.fact\_bookings f

inner join uber.area\_dimension a

on f.area\_id=a.id

group by f.day\_key , hour(f.booking\_timestamp), a.area\_name, a.city, a.state

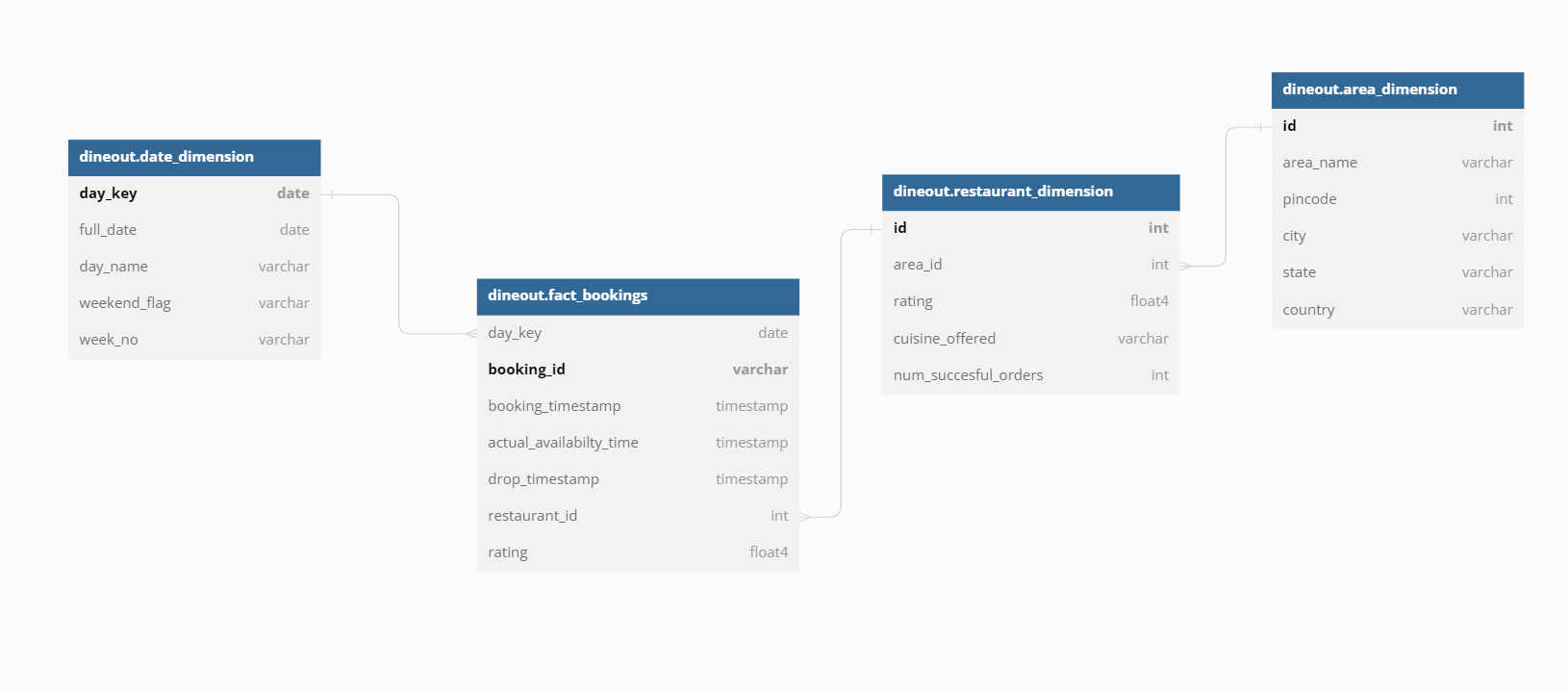
order by count(\*) desc limit 1

)

select \* from cte

**Q4. Dineout Datawarehouse**

Ans: **DWH diagram (Snowflake Schema)**



**List of Dimensions and their definition:**

**All the dimensions shall be maintained in SCD-Type1 fashion**

Table dineout.restaurant\_dimension {

id int [pk]

area\_id int [ref: > dineout.area\_dimension.id]

rating float4

cuisine\_offered varchar

num\_succesful\_orders int

}

Table dineout.area\_dimension {

id int [pk]

area\_name varchar

pincode int

city varchar

state varchar

country varchar

}

Table dineout.date\_dimension {

day\_key date [pk]

full\_date date

day\_name varchar

weekend\_flag varchar

week\_no varchar

}

**Facts and their definition:**

Table dineout.fact\_bookings {

day\_key date [ref: > dineout.date\_dimension.day\_key]

booking\_id varchar [pk]

booking\_timestamp timestamp

est\_availability\_time timestamp

actual\_availabilty\_time timestamp

drop\_timestamp timestamp

restaurant\_id int [ref: > dineout.restaurant\_dimension.id]

rating float4

final\_status varchar

}

**Insightful reports:**

**1. What is the area wise percentage of tables provided to the customer on time?**

with cte as

(

select a.area\_name, a.city,a.state, count(\*) ontime\_bookings from

dineout.fact\_bookings f

inner join

dineout.area\_dim a

on a.id= f.area\_id

where actual\_availabilty\_time <= est\_availabilty\_time

group by a.area\_name, a.city,a.state

),

cte1 as

(

select a.area\_name, a.city,a.state, count(\*) total\_bookings from

dineout.fact\_bookings f

inner join

dineout.area\_dim a

on a.id= f.area\_id

where f.final\_status='success'

group by a.area\_name. a.city,a.state

)

select a.area\_name, a.city,a.state, (cte.ontime\_bookings

\*100/cte1.total\_bookings) area\_wise\_percentage

from cte

inner join cte1

on cte.area\_name =cte1. area\_name

and cte.city =cte1.city

and cte.state =cte1.state

**2. How many bookings (percentage) were cancelled because of**

**delayed table availabilty, area wise?**

with cte as

(

select a.area\_name, a.city,a.state, count(\*) late\_bookings from

dineout.fact\_bookings f

inner join

dineout.area\_dim a

on a.id= f.area\_id

where actual\_availabilty\_time > est\_availabilty\_time

and final\_status=’cancelled’

group by a.area\_name, a.city,a.state

),

cte1 as

(

select a.area\_name, a.city,a.state, count(\*) total\_bookings from

dineout.fact\_bookings f

inner join

dineout.area\_dim a

on a.id= f.area\_id

group by a.area\_name. a.city,a.state

)

select a.area\_name, a.city,a.state, (cte.late\_bookings

\*100/cte1.total\_bookings) area\_wise\_percentage

from cte

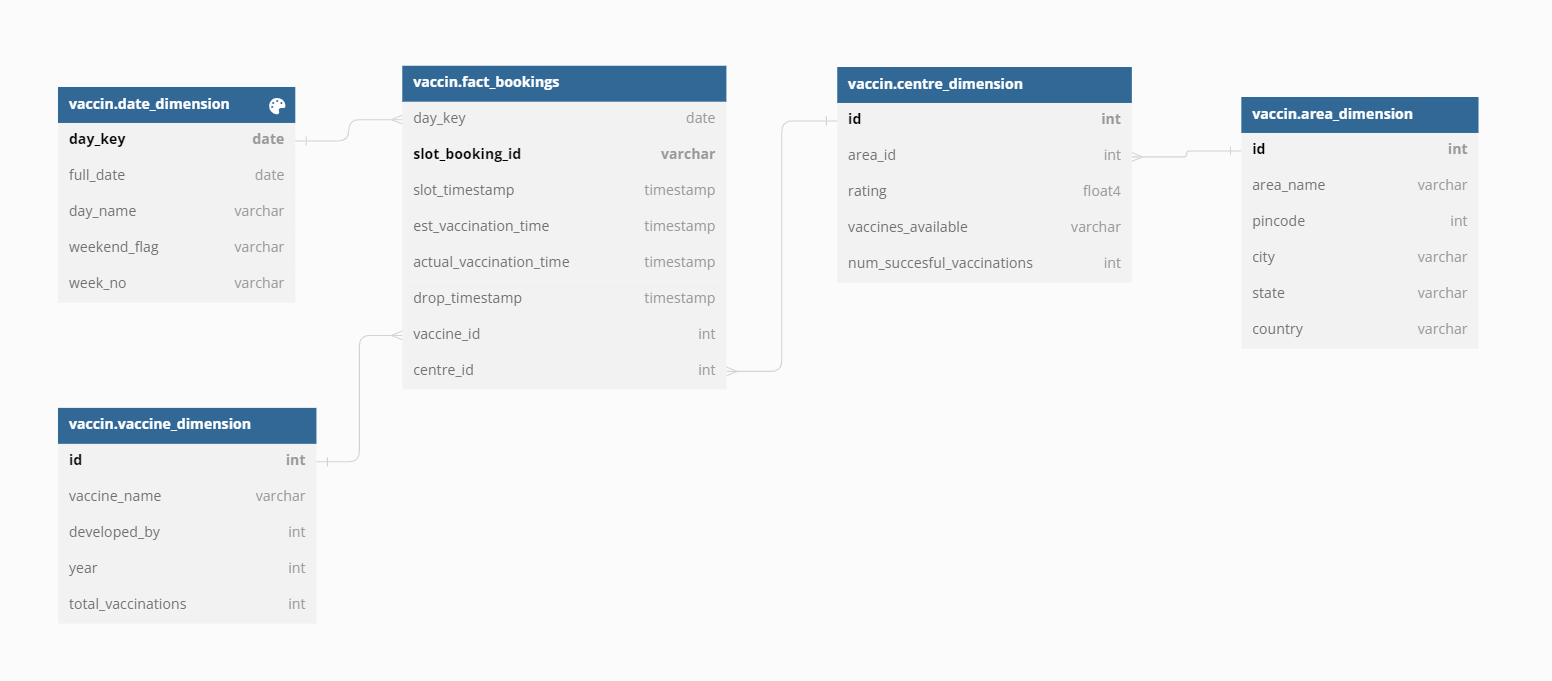
inner join cte1

on cte.area\_name =cte1. area\_name

and cte.city =cte1.city

and cte.state =cte1.state;

**Q5. Covid Vaccination Datawarehouse**

Ans: **DWH diagram (Snowflake Schema)**

**List of Dimensions and their definition:**

**All the dimensions shall be maintained in SCD-Type1 fashion**

Table vaccin.centre\_dimension {

id int [pk]

area\_id int [ref: > vaccin.area\_dimension.id]

rating float4

vaccines\_available varchar

num\_succesful\_vaccinations int

}

Table vaccin.area\_dimension {

id int [pk]

area\_name varchar

pincode int

city varchar

state varchar

country varchar

}

Table vaccin.date\_dimension {

day\_key date [pk]

full\_date date

day\_name varchar

weekend\_flag varchar

week\_no varchar

}

Table vaccin.date\_dimension {

day\_key date [pk]

full\_date date

day\_name varchar

weekend\_flag varchar

week\_no varchar

}

**Facts and their definition:**

Table vaccin.fact\_vaccinations {

day\_key date [ref: > vaccin.date\_dimension.day\_key]

slot\_booking\_id varchar [pk]

slot\_timestamp timestamp

est\_vaccination\_time timestamp

actual\_vaccination\_time timestamp

drop\_timestamp timestamp

vaccine\_id int [ref: > vaccin.vaccine\_dimension.id]

centre\_id int [ref: > vaccin.centre\_dimension.id]

}

**Insightful reports:**

**1. What is the area wise percentage of vaccinations done on time?**

with cte as

(

select a.area\_name, a.city,a.state, count(\*) ontime\_vaccinations from

vaccin.fact\_vaccinations f

inner join

vaccin.centre\_dimension c

on c.id= f.centre\_id

inner join

vaccin.area\_dimension a

on a.id= c.area\_id

where actual\_vaccination\_time <= est\_vaccination\_time

group by a.area\_name, a.city,a.state

),

cte1 as

(

select a.area\_name, a.city,a.state, count(\*) total\_vaccinations from

vaccin.fact\_vaccinations f

inner join

vaccin.centre\_dimension c

on c.id= f.centre\_id

inner join

vaccin.area\_dimension a

on a.id= c.area\_id

group by a.area\_name. a.city,a.state

)

select a.area\_name, a.city,a.state, (cte.ontime\_vaccinations

\*100/cte1.total\_vaccinations) area\_wise\_percentage

from cte

inner join cte1

on cte.area\_name =cte1. area\_name

and cte.city =cte1.city

and cte.state =cte1.state

**2. What was the peak time of slot bookings, area wise , day wise?**

with cte as

(

select a.area\_name, a.city,a.state,f.day\_key,f.day\_name,hour(f.slot\_timestamp) hour,

count(\*) over (partition by a.area\_name,

a.city,a.state,f.day\_key,f.day\_name,hour(f.slot\_timestamp)) total from

vaccin.fact\_vaccinations f

inner join vaccin.date\_dimension d

d.day\_key=f.day\_key

inner join

vaccin.centre\_dimension c

on c.id= f.centre\_id

inner join

vaccin.area\_dimension a

on a.id= c.area\_id

),

cte1 as

(

select

cte.area\_name, cte.city,cte.state,cte.day\_key,cte.day\_name,cte.hour

hour,

dense\_rank() over(partition by cte.area\_name,

cte.city,cte.state,cte.day\_key,cte.day\_name order by cte.total desc) rnk

from cte

)

select \*

from cte1

where rnk=1