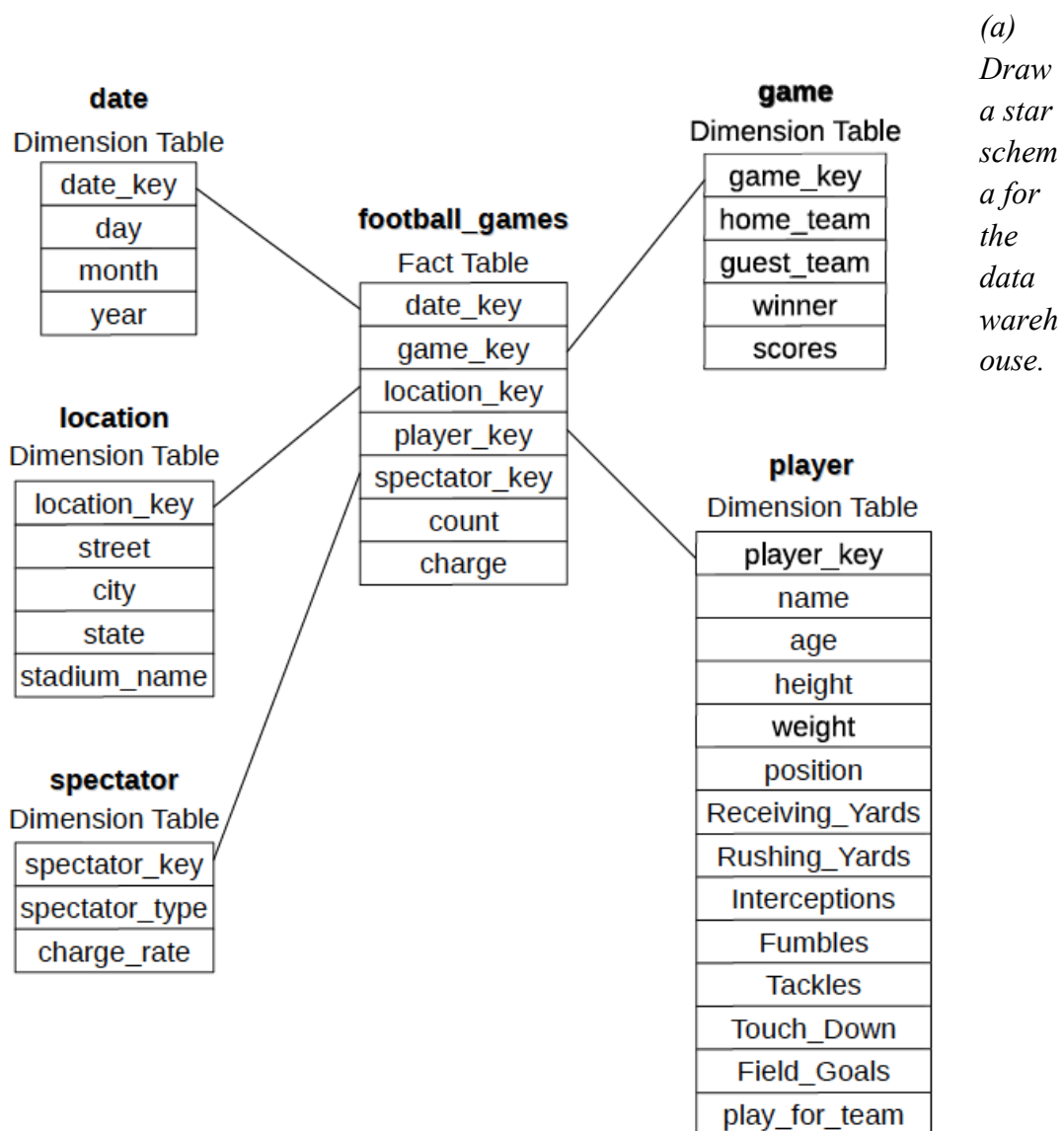
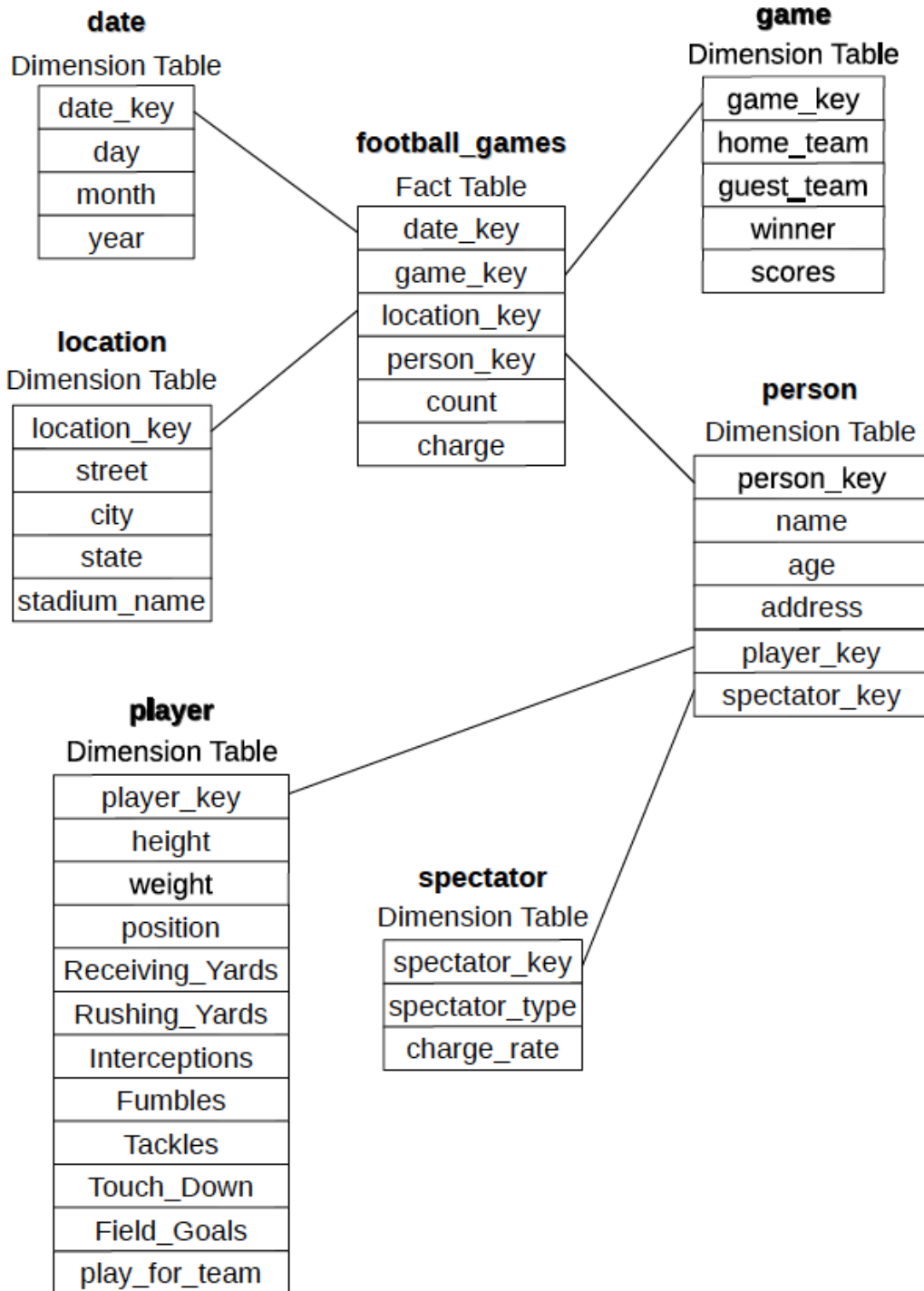


Solution for CIS 452/552 Assignment Three

1. A data warehouse for previous NCAA football games has five dimensions (date, game, location, player, and spectator), and two measures (count and charge). Charge is what a spectator pays when watching a game in a given date and location. Count is number of tickets sold. The spectators can be students, faculty, adults, seniors (older than 65), children (younger than 13), with each category having its own charge rate. The player information includes name, age, height, weight, position, the numbers of Receiving Yards, Rushing Yards, Interceptions, Fumbles, Tackles, Touch Downs, and Field Goals in a game.



(b) If we treat spectators and players as persons who all have name, address, age. For example, players in one game can be spectators of another game. Please design a new schema to represent this. Use DMQL to define the new schema.



DMQL answer:

```
define cube NACC[Time, Game, Location, People]:
```

```
count = count(*), charge = charge_rate
```

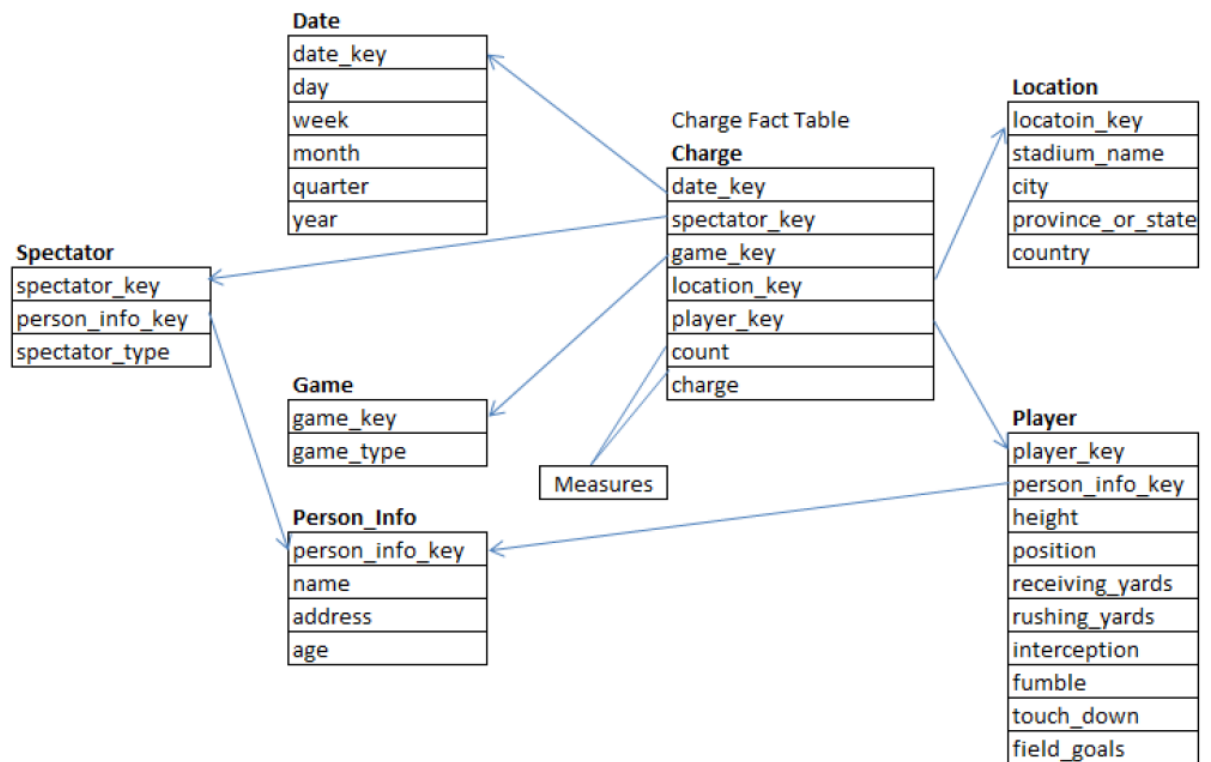
```
define dimension Time as (time_key, day, day_of_week, month, quarter, year)
```

```
define dimension Game as (game_key, game_name, host_team, away_team)
```

```
define dimension Location as (location_key, country, state, city, street, unit)
```

```
define dimension Person as (people_key, name, age, address, spectator  
(spectator_key, charge_rate, type), player(player_key, height, weight, position,  
receiving_yards, rushing_yards, interceptions, fumbles, Trackles, Touch_Downs,  
field_goals) )
```

another option:



(c) Starting with the base cuboid [date, game, location, player, spectator], what specific OLAP operations (e.g., roll-up from quarter to year) should one perform in order to list the total charge paid by all students at Autzen Stadium in the fall of 2016 and Justin Herbert was one of players?

Answer:

Roll-Up

On Date

(from days to quarters)

Roll-Up

For spectator

(from individuals to types)

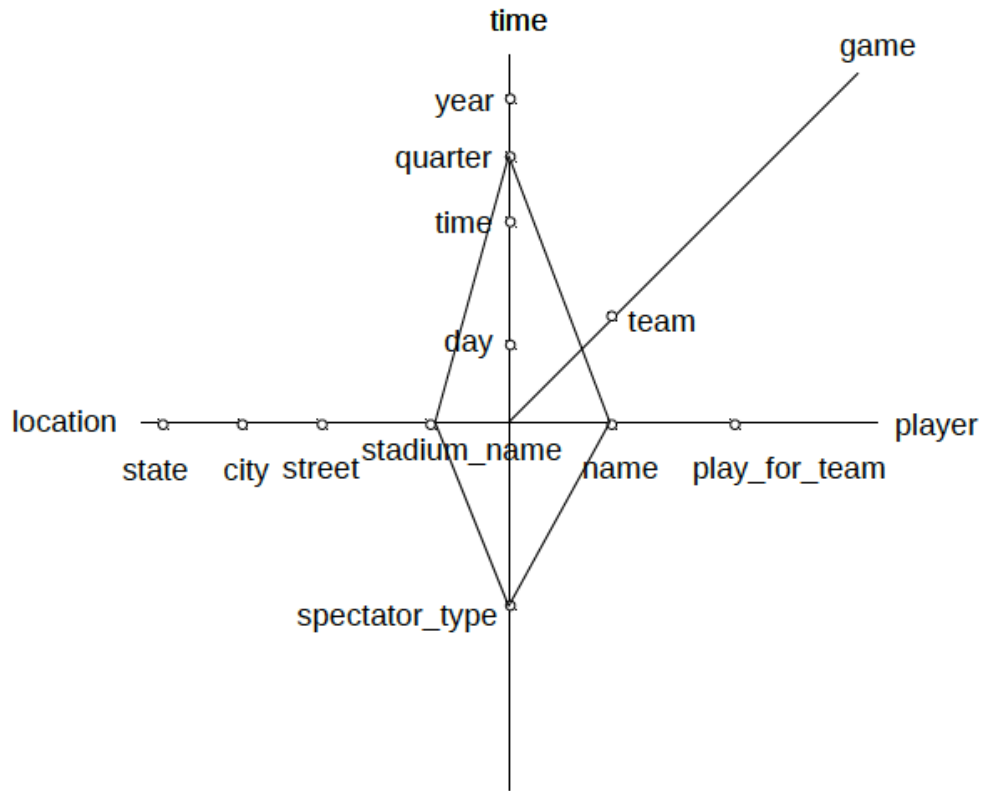
Dice

For

(location = "Autzen Stadium") and (Date = "2016 Fall") and (Player = "Justin Herbert") and (spectator_type = "students")

(d) Using a star network query model to represent your design in (a) and the query in (c). Can you write down the query in (c) with SQL? If yes, show your SQL query. If not, explain why.

Answer:



SQL Query:

```
Select sum(sf.charge)
```

```
from Sales_Fact sf, Time t, Game g, Location l, Player p, Spectators sp,
```

```
where P.name = " Justin Herbert " and T.year = "2016" and L.name = "Autzen Stadium  
" and T.quarter = "fall" and (G.host_team = P.team or G.away_team = P.team ) and  
sp.type = "students"
```

It requires join operation of 6 relations, which is time and resources consuming.

2. We talked about virtual data warehouse and data mediator in the class. Are they exactly the same thing? If not, what's the difference? Whether the virtual data warehouse can support the typical OLAP operations such as roll-up and drill-down?

Answer:

In traditional heterogeneous database integration problem, we build wrappers or mediators on top of heterogeneous databases, the mediators don't actually store or

aggregate the data in the bottom level of databases, they either transform query of from user into low level database query language or transform the answers from databases to users. What's different in virtual data warehouse is we provide a set of materialized views over operational databases. Therefore, even virtual data warehouses do not need to store all historical data from operational databases, they store some of them through materialized views, which can be used for direct query and analysis. Therefore, virtual data warehouses can still support typical OLAP operations, such as roll-up and possibly drill down, but it depends how much levels of data they store in the materialized views.

3. Suppose that a data warehouse contains 20 dimensions, each with about five levels of granularity. Users are mainly interested in four particular dimensions, each having three frequently accessed levels for rolling up and drilling down. How would you design a data cube structure to support this preference efficiently?

Answer:

I will design a data cube by materializing (pre-computing) the four dimension and three level of granularity for each dimension. The total number of cuboids are $4^4 = 256$ which is an acceptable number. Therefore, we partially materialize the data cube for all the four dimensions and three level foot-prints.