When closely examined output of the first query after explain analyze, the function 'get\_fiscal\_year(date)' is executed for every customer for every date.

For the same date, it is going back and forth to the 'get\_fiscal\_year(date)’ function to retrieve fiscal\_year again and again, which is increasing the execution time.

Hence I created a dim\_date table where for one date you have the corresponding fiscal year column. (dim\_date serves the purpose of a lookup table)

A lookup table (LUT) stores precomputed values to quickly retrieve data based on an index, improving computation efficiency by avoiding repetitive calculations.

I created an empty dim\_date table with columns:

* Calender\_date
* Fiscal\_year

Fiscal year is a generated column whose formula

= year(date\_add(calender\_date, interval 4 month))

In an excel file, I input possible calendar dates needed for us as per fact\_sales\_monthly table and saved the file as .csv and imported those calender dates to our empty dim\_date table in mysql. The fiscal\_year column values have automatically generated as per the formula given.

-

*creating dim\_date and joining with this table and avoid using the function 'get\_fiscal\_year()' to reduce the amount of time taking to run the query*

SELECT

s.date,

s.customer\_code,

s.product\_code,

p.product, p.variant,

s.sold\_quantity,

g.gross\_price as gross\_price\_per\_item,

ROUND(s.sold\_quantity\*g.gross\_price,2) as gross\_price\_total,

pre.pre\_invoice\_discount\_pct

FROM fact\_sales\_monthly s

JOIN dim\_date dt

ON dt.calendar\_date = s.date

JOIN dim\_product p

ON s.product\_code=p.product\_code

JOIN fact\_gross\_price g

ON g.fiscal\_year=dt.fiscal\_year

AND g.product\_code=s.product\_code

JOIN fact\_pre\_invoice\_deductions as pre

ON pre.customer\_code = s.customer\_code AND

pre.fiscal\_year=dt.fiscal\_year

WHERE

dt.fiscal\_year=2021

LIMIT 1500000;

1. **Execution time reduced to 3 seconds** when optimized the query by creating a lookup table