## Creating Fact and Dimension Tables

CREATE TABLE product

(

pid serial NOT NULL,

pkey integer NOT NULL,

pname character varying(30) ,

unit\_price character varying(30),

category character varying(40),

CONSTRAINT product\_pkey PRIMARY KEY (pid)

)

CREATE TABLE store

(

sid serial NOT NULL,

skey integer NOT NULL,

sname character varying(30) ,

city character varying(30),

state character varying(40),

country character varying(30),

region character varying(30),

CONSTRAINT store\_pkey PRIMARY KEY (sid)

)

CREATE TABLE time

(

tid serial NOT NULL,

tkey timestamp without time zone NOT NULL,

day character varying(30) NOT NULL,

month character varying(10) NOT NULL,

quarter character varying(2) NOT NULL,

year integer NOT NULL,

CONSTRAINT time\_pkey PRIMARY KEY (tid)

)

CREATE TABLE sales

(

pid integer NOT NULL REFERENCES product(pid),

sid integer NOT NULL REFERENCES store(sid),

tid integer NOT NULL REFERENCES time(tid),

units\_sold integer,

total\_sales integer,

CONSTRAINT sales\_pkey PRIMARY KEY (pid,sid,tid)

)

## -- Inserting Into Fact and Dimension Tables

insert into product(pkey,pname,unit\_price,category)values(1,'Hand Sanitizers','100','Health'),(2,'Soap','100','Health')

insert into time(tkey, day, month, quarter, year)values

(timestamp'2017-01-26 10:23:54',date\_part('day', timestamp '2017-08-26 10:23:54'),date\_part('month', timestamp '2017-01-26 10:23:54'),concat('Q',date\_part('quarter', timestamp '2017-01-26 10:23:54')),date\_part('year', timestamp '2017-01-26 10:23:54')),

(timestamp'2017-04-26 10:23:54',date\_part('day', timestamp '2017-08-26 10:23:54'),date\_part('month', timestamp '2017-04-26 10:23:54'),concat('Q',date\_part('quarter', timestamp '2017-04-26 10:23:54')),date\_part('year', timestamp '2017-04-26 10:23:54')),

(timestamp'2017-05-26 10:23:54',date\_part('day', timestamp '2017-08-26 10:23:54'),date\_part('month', timestamp '2017-05-26 10:23:54'),concat('Q',date\_part('quarter', timestamp '2017-05-26 10:23:54')),date\_part('year', timestamp '2017-05-26 10:23:54'))

(timestamp'2020-11-19 10:23:54',date\_part('day', timestamp '2019-11-23 11:23:54'),date\_part('month', timestamp '2019-11-23 11:23:54'),concat('Q',date\_part('quarter', timestamp '2019-11-23 11:23:54')),date\_part('year', timestamp '2019-11-23 11:23:54')),

(timestamp'2020-11-23 11:23:54',date\_part('day', timestamp '2019-11-23 11:23:54'),date\_part('month', timestamp '2019-11-23 11:23:54'),concat('Q',date\_part('quarter', timestamp '2019-11-23 11:23:54')),date\_part('year', timestamp '2019-11-23 11:23:54')),

(timestamp'2019-11-23 11:23:54',date\_part('day', timestamp '2019-11-23 11:23:54'),date\_part('month', timestamp '2019-11-23 11:23:54'),concat('Q',date\_part('quarter', timestamp '2019-11-23 11:23:54')),date\_part('year', timestamp '2019-11-23 11:23:54')),

(timestamp'2018-01-23 10:23:54',date\_part('day', timestamp '2018-01-23 10:23:54'),date\_part('month', timestamp '2018-01-23 10:23:54'),concat('Q',date\_part('quarter', timestamp '2018-01-23 10:23:54')),date\_part('year', timestamp '2018-01-23 10:23:54')),

(timestamp'2018-04-18 09:23:54',date\_part('day', timestamp '2018-04-18 09:23:54'),date\_part('month', timestamp '2018-04-18 09:23:54'),concat('Q',date\_part('quarter', timestamp '2018-04-18 09:23:54')),date\_part('year', timestamp '2018-04-18 09:23:54')),

(timestamp'2018-06-19 08:23:54',date\_part('day', timestamp '2018-06-19 08:23:54'),date\_part('month', timestamp '2018-06-19 08:23:54'),concat('Q',date\_part('quarter', timestamp '2018-06-19 08:23:54')),date\_part('year', timestamp '2018-06-19 08:23:54')),

(timestamp'2018-11-22 09:23:54',date\_part('day', timestamp '2018-11-22 09:23:54'),date\_part('month', timestamp '2018-11-22 09:23:54'),concat('Q',date\_part('quarter', timestamp '2018-11-22 09:23:54')),date\_part('year', timestamp '2018-11-22 09:23:54')),

(timestamp'2016-05-25 08:23:54',date\_part('day', timestamp '2016-05-25 08:23:54'),date\_part('month', timestamp '2016-05-25 08:23:54'),concat('Q',date\_part('quarter', timestamp '2016-05-25 08:23:54')),date\_part('year', timestamp '2016-05-25 08:23:54')),

(timestamp'2015-11-24 11:23:54',date\_part('day', timestamp '2015-11-24 11:23:54'),date\_part('month', timestamp '2015-11-24 11:23:54'),concat('Q',date\_part('quarter', timestamp '2015-11-24 11:23:54')),date\_part('year', timestamp '2015-11-24 11:23:54'))

insert into store(skey, sname, city, state, country, region)values

(1,'Bandra Stores','Mumbai','MH','India','West'),

(2,'ABC Stores','Surat','GJ','India','West'),

(3,'QWERTY Stores','Mumbai','MH','India','West'),

(4,'XYZ Stores','Surat','GJ','India','West')

insert into sales( pid, sid, tid, units\_sold, total\_sales) values

(1,4,11,1,190),

(1,1,12,26,100),

(1,2,13,36,120)

(1,4,8,1,190),

(1,1,8,26,100),

(1,2,8,36,120)

(1,1,1,10,100),

(2,1,2,20,200),

(1,2,3,30,300),

(2,2,4,40,400),

(1,1,5,50,500),

(2,1,6,10,10),

(1,2,7,15,150),

(2,2,8,16,160),

(1,1,9,17,170),

(2,1,10,18,180)

## --Viewing Fact and Dimension Tables

select \* from product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| pid | pkey | pname | unit\_price | category |
| 1 | 1 | Hand Sanitizer | 100 | Health |
| 2 | 2 | Soap | 100 | Commercial |

select \* from store

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| sid | skey | sname | city | state | country | region |
| 1 | 1 | Bandra Stores | Mumbai | MH | India | West |
| 2 | 2 | ABC Stores | Surat | GJ | India | West |
| 3 | 3 | QWERTY Stores | Mumbai | MH | India | West |
| 4 | 4 | XYZ Stores | Surat | GJ | India | West |

select \* from time

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| tid | tkey | day | month | quarter | year |
| 1 | 19-02-2020 10:23 | 19 | 2 | Q1 | 2020 |
| 2 | 23-02-2020 11:23 | 23 | 2 | Q1 | 2020 |
| 3 | 23-11-2019 11:23 | 23 | 11 | Q4 | 2019 |
| 4 | 23-01-2018 10:23 | 23 | 1 | Q1 | 2018 |
| 5 | 18-04-2018 09:23 | 18 | 4 | Q2 | 2018 |
| 6 | 19-06-2018 08:23 | 19 | 6 | Q2 | 2018 |
| 7 | 22-11-2018 09:23 | 22 | 11 | Q4 | 2018 |
| 8 | 26-08-2017 10:23 | 26 | 8 | Q3 | 2017 |
| 9 | 25-05-2016 08:23 | 25 | 5 | Q2 | 2016 |
| 10 | 24-11-2015 11:23 | 24 | 11 | Q4 | 2015 |
| 11 | 26-01-2017 10:23 | 26 | 1 | Q1 | 2017 |
| 12 | 26-04-2017 10:23 | 26 | 4 | Q2 | 2017 |
| 13 | 26-05-2017 10:23 | 26 | 5 | Q2 | 2017 |

select \* from sales

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| pid | sid | tid | units\_sold | total\_sales |
| 1 | 1 | 1 | 10 | 100 |
| 2 | 1 | 2 | 20 | 200 |
| 1 | 2 | 3 | 30 | 300 |
| 2 | 2 | 4 | 40 | 400 |
| 1 | 1 | 5 | 50 | 500 |
| 2 | 1 | 6 | 10 | 10 |
| 1 | 2 | 7 | 15 | 150 |
| 2 | 2 | 8 | 16 | 160 |
| 1 | 1 | 9 | 17 | 170 |
| 2 | 1 | 10 | 18 | 180 |
| 2 | 3 | 8 | 10 | 10 |
| 1 | 4 | 8 | 1 | 190 |
| 1 | 1 | 8 | 26 | 100 |
| 1 | 2 | 8 | 36 | 120 |
| 1 | 4 | 11 | 1 | 190 |
| 1 | 1 | 12 | 26 | 100 |
| 1 | 2 | 13 | 36 | 120 |

-------------------------------------------------------------------------------------------------------------------------------------

## -- Q1) Describe all DT

-- Ans.

SELECT \* FROM information\_schema.COLUMNS WHERE TABLE\_NAME = 'product';

SELECT \* FROM information\_schema.COLUMNS WHERE TABLE\_NAME = 'store';

SELECT \* FROM information\_schema.COLUMNS WHERE TABLE\_NAME = 'time';

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| column\_name | ordinal\_position | column\_default | is\_nullable | data\_type | character\_maximum\_length | character\_octet\_length | numeric\_precision | numeric\_precision\_radix | numeric\_scale | datetime\_precision | … |
| tid | 1 | nextval('time\_tid\_seq'::regclass) | NO | integer | NULL | NULL | 32 | 2 | 0 | NULL | … |
| tkey | 2 | NULL | NO | timestamp without time zone | NULL | NULL | NULL | NULL | NULL | 6 | … |
| day | 3 | NULL | NO | character varying | 30 | 120 | NULL | NULL | NULL | NULL | … |
| month | 4 | NULL | NO | character varying | 10 | 40 | NULL | NULL | NULL | NULL | … |
| quarter | 5 | NULL | NO | character varying | 2 | 8 | NULL | NULL | NULL | NULL |  |
| year | 6 | NULL | NO | integer | NULL | NULL | 32 | 2 | 0 | NULL |  |

## -- Q2) Describe FT

-- Ans.

SELECT \* FROM information\_schema.COLUMNS WHERE TABLE\_NAME = 'sales';

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| table\_name | column\_name | ordinal\_position | column\_default | is\_nullable | data\_type | character\_maximum\_length | character\_octet\_length | numeric\_precision |
| sales | pid | 1 | NULL | NO | integer | NULL | NULL | 32 |
| sales | sid | 2 | NULL | NO | integer | NULL | NULL | 32 |
| sales | tid | 3 | NULL | NO | integer | NULL | NULL | 32 |
| sales | units\_sold | 4 | NULL | YES | integer | NULL | NULL | 32 |
| sales | total\_sales | 5 | NULL | YES | integer | NULL | NULL | 32 |

## -- Q3) Create basic cube by considering all queries

-- Ans.

-- pgAdmin4

create table C1 as

select \* from sales

natural join products

natural join store

natural join time

SELECT C1.category,C1.state,C1.year,sum(C1.total\_sales) as Total\_Sales

FROM C1

GROUP BY CUBE ( C1.category,C1.state,C1.year);

-- or

SELECT C1.category,C1.state,C1.year,sum(C1.total\_sales) as Total\_Sales

FROM C1

GROUP BY GROUPING SETS ((C1.category,C1.state,C1.year),

(C1.category,C1.state),

( C1.category,C1.year),

( C1.state,C1.year),

( c1.category),

( C1.state),

( C1.year),

()) ;

-- pgAdmin3

(SELECT C1.category, C1.state, C1.year,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 group by C1.category, C1.state, C1.year)

union all

(SELECT C1.category, C1.state, NULL ,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 group by C1.category, C1.state)

union all

(SELECT C1.category, NULL, C1.year,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 group by C1.category,C1.year)

union all

(SELECT NULL, C1.state, C1.year,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 group by C1.state, C1.year)

union all

(SELECT C1.category, NULL, NULL,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 group by C1.category)

union all

(SELECT NULL, C1.state, NULL,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 group by C1.state)

union all

(SELECT NULL, NULL, C1.year,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 group by C1.year)

union all

(SELECT NULL,NULL,NULL, sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 )

-- output

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| category | state | year | units\_sold | total\_sales |
| NULL | NULL | NULL | 362 | 3000 |
| Health | GJ | NULL | 119 | 1070 |
| Commercial | MH | NULL | 58 | 400 |
| Health | MH | NULL | 129 | 970 |
| Commercial | GJ | NULL | 56 | 560 |
| Health | NULL | NULL | 248 | 2040 |
| Commercial | NULL | NULL | 114 | 960 |
| Commercial | MH | 2015 | 18 | 180 |
| Health | GJ | 2019 | 30 | 300 |
| Health | MH | 2018 | 50 | 500 |
| Health | MH | 2016 | 17 | 170 |
| Health | MH | 2017 | 52 | 200 |
| Health | GJ | 2017 | 74 | 620 |
| Commercial | GJ | 2017 | 16 | 160 |
| Commercial | MH | 2017 | 10 | 10 |
| Commercial | MH | 2018 | 10 | 10 |
| Commercial | MH | 2020 | 20 | 200 |
| Health | MH | 2020 | 10 | 100 |
| Commercial | GJ | 2018 | 40 | 400 |
| Health | GJ | 2018 | 15 | 150 |
| NULL | MH | 2017 | 62 | 210 |
| NULL | GJ | 2017 | 90 | 780 |
| NULL | MH | 2016 | 17 | 170 |
| NULL | MH | 2015 | 18 | 180 |
| NULL | MH | 2018 | 60 | 510 |
| NULL | GJ | 2018 | 55 | 550 |
| NULL | GJ | 2019 | 30 | 300 |
| NULL | MH | 2020 | 30 | 300 |
| NULL | MH | NULL | 187 | 1370 |
| NULL | GJ | NULL | 175 | 1630 |
| Health | NULL | 2020 | 10 | 100 |
| Commercial | NULL | 2015 | 18 | 180 |
| Health | NULL | 2017 | 126 | 820 |
| Health | NULL | 2018 | 65 | 650 |
| Commercial | NULL | 2017 | 26 | 170 |
| Commercial | NULL | 2020 | 20 | 200 |
| Health | NULL | 2019 | 30 | 300 |
| Health | NULL | 2016 | 17 | 170 |
| Commercial | NULL | 2018 | 50 | 410 |
| NULL | NULL | 2017 | 152 | 990 |
| NULL | NULL | 2016 | 17 | 170 |
| NULL | NULL | 2018 | 115 | 1060 |
| NULL | NULL | 2020 | 30 | 300 |
| NULL | NULL | 2015 | 18 | 180 |
| NULL | NULL | 2019 | 30 | 300 |

## -- Q4) Describe cube

-- Ans.

-- pgadmin 4

explain SELECT C1.category,C1.state,C1.year,sum(C1.total\_sales) as Total\_Sales

FROM C1

GROUP BY CUBE ( C1.category,C1.state,C1.year);

-- or

explain SELECT C1.category,C1.state,C1.year,sum(C1.total\_sales) as Total\_Sales

FROM C1

GROUP BY GROUPING SETS ((C1.category,C1.state,C1.year),

(C1.category,C1.state),

( C1.category,C1.year),

( C1.state,C1.year),

( c1.category),

( C1.state),

( C1.year),

()) ;

-- pgadmin 3

EXPLAIN (SELECT C1.category, C1.state, C1.year,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 group by C1.category, C1.state, C1.year)

union all

(SELECT C1.category, C1.state, NULL ,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 group by C1.category, C1.state)

union all

(SELECT C1.category, NULL, C1.year,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 group by C1.category,C1.year)

union all

(SELECT NULL, C1.state, C1.year,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 group by C1.state, C1.year)

union all

(SELECT C1.category, NULL, NULL,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 group by C1.category)

union all

(SELECT NULL, C1.state, NULL,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 group by C1.state)

union all

(SELECT NULL, NULL, C1.year,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 group by C1.year)

union all

(SELECT NULL,NULL,NULL, sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 )

-- output –

|  |
| --- |
| QUERY PLAN |
| MixedAggregate (cost=0.00..21.71 rows=631 width=208) |
| Hash Key: category, state, year |
| Hash Key: category, state |
| Hash Key: category |
| Hash Key: state, year |
| Hash Key: state |
| Hash Key: year, category |
| Hash Key: year |
| Group Key: () |
| -> Seq Scan on c1 (cost=0.00..10.90 rows=90 width=204) |

## -- Q5) Give sales cube for ( all quarters(q1,q2,q3,q4), all product categories(ci,c2,c3,c4), all

## -- store states(s1,s2,s3,s4)) for year 2017

-- Ans.

-- pgadmin 4

SELECT C1.category,C1.state,C1.quarter,sum(C1.total\_sales) as Total\_Sales

FROM C1

WHERE C1.year = 2017

GROUP BY GROUPING SETS ((C1.category,C1.state,C1.quarter),

(C1.category,C1.state),

( C1.category,C1.quarter),

( C1.state,C1.quarter),

( c1.category),

( C1.state),

( C1.quarter),

()) ;

-- or

SELECT C1.category,C1.state,C1.quarter,sum(C1.total\_sales) as Total\_Sales

FROM C1

WHERE C1.year = 2017

GROUP BY CUBE(C1.category,C1.state,C1.quarter)

--pgadmin 3

(SELECT C1.category, C1.state, C1.quarter,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 WHERE C1.year = 2017 group by C1.category, C1.state, C1.quarter)

union all

(SELECT C1.category, C1.state, NULL ,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 WHERE C1.year = 2017 group by C1.category, C1.state)

union all

(SELECT C1.category, NULL, C1.quarter,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 WHERE C1.year = 2017 group by C1.category,C1.quarter)

union all

(SELECT NULL, C1.state, C1.quarter,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 WHERE C1.year = 2017 group by C1.state, C1.quarter)

union all

(SELECT C1.category, NULL, NULL,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 WHERE C1.year = 2017 group by C1.category)

union all

(SELECT NULL, C1.state, NULL,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 WHERE C1.year = 2017 group by C1.state)

union all

(SELECT NULL, NULL, C1.quarter,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 WHERE C1.year = 2017 group by C1.quarter)

union all

(SELECT NULL,NULL,NULL, sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 WHERE C1.year = 2017)

--output

|  |  |  |  |
| --- | --- | --- | --- |
| category | state | quarter | total\_sales |
| NULL | NULL | NULL | 990 |
| Health | GJ | Q2 | 120 |
| Commercial | MH | Q3 | 10 |
| Commercial | GJ | Q3 | 160 |
| Health | GJ | Q1 | 190 |
| Health | GJ | Q3 | 310 |
| Health | MH | Q2 | 100 |
| Health | MH | Q3 | 100 |
| Health | MH | NULL | 200 |
| Health | GJ | NULL | 620 |
| Commercial | MH | NULL | 10 |
| Commercial | GJ | NULL | 160 |
| Commercial | NULL | NULL | 170 |
| Health | NULL | NULL | 820 |
| NULL | GJ | Q1 | 190 |
| NULL | GJ | Q2 | 120 |
| NULL | MH | Q3 | 110 |
| NULL | GJ | Q3 | 470 |
| NULL | MH | Q2 | 100 |
| NULL | GJ | NULL | 780 |
| NULL | MH | NULL | 210 |
| Health | NULL | Q2 | 220 |
| Commercial | NULL | Q3 | 170 |
| Health | NULL | Q1 | 190 |
| Health | NULL | Q3 | 410 |
| NULL | NULL | Q3 | 580 |
| NULL | NULL | Q2 | 220 |
| NULL | NULL | Q1 | 190 |

## Q6) Write appropriate SQL queries for slice operation, Give the sales of MH or s1 state stores for all product categories, for all quarter for the year 2018

-- Ans.

-- pgadmin 4

SELECT C1.category,C1.quarter,sum(C1.total\_sales) as Total\_Sales

FROM C1

WHERE C1.state = 'MH' and C1.year = 2018

GROUP BY CUBE ( C1.category,C1.quarter)

-- OR

SELECT C1.category,C1.quarter,sum(C1.total\_sales) as Total\_Sales

FROM C1

WHERE C1.state = 'MH' and C1.year = 2018

GROUP BY GROUPING SETS (( C1.category,C1.quarter),

( c1.category),

( C1.quarter),

()) ;

-- PGADMIN 3

(SELECT C1.category,C1.quarter,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 WHERE C1.state = 'MH' and C1.year = 2018 group by C1.category, C1.quarter)

union all

(SELECT C1.category,NULL ,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 WHERE C1.state = 'MH' and C1.year = 2018 group by C1.category)

union all

(SELECT NULL,C1.quarter,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 WHERE C1.state = 'MH' and C1.year = 2018 group by C1.quarter)

union all

(SELECT NULL,NULL, sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 WHERE C1.state = 'MH' and C1.year = 2018)

-- Output

|  |  |  |
| --- | --- | --- |
| category | quarter | total\_sales |
| NULL | NULL | 510 |
| Commercial | Q2 | 10 |
| Health | Q2 | 500 |
| NULL | Q2 | 510 |
| Commercial | NULL | 10 |
| Health | NULL | 500 |

## Q7) Write appropriate SQL queries for dice operation Give the sales of MH or s1 state and GJ or s2 for product categories c1 and c2, for quarter q1 and q2 for the year 2018

-- Ans.

-- PGADMIN 4

SELECT C1.category,C1.state,C1.quarter,sum(C1.total\_sales) as Total\_Sales

FROM C1

WHERE (C1.category='Health' or C1.category='Commercial') and (C1.state = 'MH' or C1.state = 'GJ') and C1.year = 2018 and (C1.quarter = 'Q1' or C1.quarter = 'Q2')

GROUP BY CUBE (C1.category,C1.state,C1.quarter)

-- OR

SELECT C1.category,C1.state,C1.quarter,sum(C1.total\_sales) as Total\_Sales

FROM C1

WHERE (C1.category='Health' or C1.category='Commercial') and (C1.state = 'MH' or C1.state = 'GJ') and C1.year = 2018 and (C1.quarter = 'Q1' or C1.quarter = 'Q2')

GROUP BY GROUPING SETS ((C1.category,C1.state,C1.quarter),

(C1.category,C1.state),

( C1.category,C1.quarter),

( C1.state,C1.quarter),

( c1.category),

( C1.state),

( C1.quarter),

()) ;

-- PGADMIN 3

(SELECT C1.category, C1.state, C1.quarter,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1

WHERE (C1.category='Health' or C1.category='Commercial') and (C1.state = 'MH' or C1.state = 'GJ') and C1.year = 2018 and (C1.quarter = 'Q1' or C1.quarter = 'Q2')

group by C1.category, C1.state, C1.quarter)

union all

(SELECT C1.category, C1.state, NULL ,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1

WHERE (C1.category='Health' or C1.category='Commercial') and (C1.state = 'MH' or C1.state = 'GJ') and C1.year = 2018 and (C1.quarter = 'Q1' or C1.quarter = 'Q2')

group by C1.category, C1.state)

union all

(SELECT C1.category, NULL, C1.quarter,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1

WHERE (C1.category='Health' or C1.category='Commercial') and (C1.state = 'MH' or C1.state = 'GJ') and C1.year = 2018 and (C1.quarter = 'Q1' or C1.quarter = 'Q2')

group by C1.category,C1.quarter)

union all

(SELECT NULL, C1.state, C1.quarter,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1

WHERE (C1.category='Health' or C1.category='Commercial') and (C1.state = 'MH' or C1.state = 'GJ') and C1.year = 2018 and (C1.quarter = 'Q1' or C1.quarter = 'Q2')

group by C1.state, C1.quarter)

union all

(SELECT C1.category, NULL, NULL,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1

WHERE (C1.category='Health' or C1.category='Commercial') and (C1.state = 'MH' or C1.state = 'GJ') and C1.year = 2018 and (C1.quarter = 'Q1' or C1.quarter = 'Q2')

group by C1.category)

union all

(SELECT NULL, C1.state, NULL,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1

WHERE (C1.category='Health' or C1.category='Commercial') and (C1.state = 'MH' or C1.state = 'GJ') and C1.year = 2018 and (C1.quarter = 'Q1' or C1.quarter = 'Q2')

group by C1.state)

union all

(SELECT NULL, NULL, C1.quarter,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1

WHERE (C1.category='Health' or C1.category='Commercial') and (C1.state = 'MH' or C1.state = 'GJ') and C1.year = 2018 and (C1.quarter = 'Q1' or C1.quarter = 'Q2')

group by C1.quarter)

union all

(SELECT NULL,NULL,NULL, sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1

WHERE (C1.category='Health' or C1.category='Commercial') and (C1.state = 'MH' or C1.state = 'GJ') and C1.year = 2018 and (C1.quarter = 'Q1' or C1.quarter = 'Q2')

)

-- output

|  |  |  |  |
| --- | --- | --- | --- |
| category | state | quarter | total\_sales |
| NULL | NULL | NULL | 910 |
| Commercial | MH | NULL | 10 |
| Commercial | GJ | NULL | 400 |
| Health | MH | NULL | 500 |
| Commercial | NULL | NULL | 410 |
| Health | NULL | NULL | 500 |
| Commercial | GJ | Q1 | 400 |
| Commercial | MH | Q2 | 10 |
| Health | MH | Q2 | 500 |
| NULL | MH | Q2 | 510 |
| NULL | GJ | Q1 | 400 |
| NULL | GJ | NULL | 400 |
| NULL | MH | NULL | 510 |
| Commercial | NULL | Q1 | 400 |
| Commercial | NULL | Q2 | 10 |
| Health | NULL | Q2 | 500 |
| NULL | NULL | Q2 | 510 |
| NULL | NULL | Q1 | 400 |

## Q8) Write appropriate SQL queries for drilldown operation

## Give the sales of MH or s1 state for all product categories, for all months for the year 2018

-- PGADMIN 4

SELECT C1.category,C1.month,sum(C1.total\_sales) as Total\_Sales

FROM C1

WHERE C1.state = 'MH' and C1.year = 2018

GROUP BY CUBE (C1.category,C1.month)

-- OR

SELECT c1.category, c1.month, sum(c1.total\_sales) as Total\_Sales

FROM c1

WHERE c1.state = 'MH' and c1.year=2018

GROUP BY GROUPING SETS (( c1.category,c1.month),

( c1.category),

( c1.month),

()) ;

-- PGADMIN 3

(SELECT C1.category, C1.month,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 WHERE C1.state = 'MH' and C1.year = 2018 group by C1.category, C1.month)

union all

(SELECT C1.category, NULL ,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 WHERE C1.state = 'MH' and C1.year = 2018 group by C1.category)

union all

(SELECT NULL, C1.month,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 WHERE C1.state = 'MH' and C1.year = 2018 group by C1.month)

union all

(SELECT NULL, NULL, sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 WHERE C1.state = 'MH' and C1.year = 2018)

--OUTPUT

|  |  |  |
| --- | --- | --- |
| category | month | total\_sales |
| NULL | NULL | 510 |
| Health | 4 | 500 |
| Commercial | 6 | 10 |
| Commercial | NULL | 10 |
| Health | NULL | 500 |
| NULL | 6 | 10 |
| NULL | 4 | 500 |

## Q9) Write appropriate SQL queries rollup operation Give the sales of MH or s1 state stores for all product categories, for all years 2018, 2017, 2016, and 2015

-- PGADMIN 4

SELECT C1.category,C1.year,sum(C1.total\_sales) as Total\_Sales

FROM C1

WHERE C1.state = 'MH'

GROUP BY ROLLUP (C1.category,C1.year)

-- OR

SELECT c1.category, c1.year, sum(c1.total\_sales) as Total\_Sales

FROM c1

WHERE c1.state = 'MH'

GROUP BY GROUPING SETS (( c1.category,c1.year),

( c1.category),

( c1.year),

()) ;

-- PGADMIN 3

(SELECT C1.category,C1.year,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 WHERE C1.state = 'MH' group by C1.category, C1.year)

union all

(SELECT C1.category,NULL ,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 WHERE C1.state = 'MH' group by C1.category)

union all

(SELECT NULL,C1.year,sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 WHERE C1.state = 'MH' group by C1.year)

union all

(SELECT NULL,NULL, sum (C1.units\_sold) as units\_sold, sum (C1.total\_sales) as total\_sales FROM C1 WHERE C1.state = 'MH')

-- OUTPUT

|  |  |  |
| --- | --- | --- |
| category | year | total\_sales |
| NULL | NULL | 1370 |
| Commercial | 2018 | 10 |
| Health | 2017 | 200 |
| Commercial | 2020 | 200 |
| Health | 2016 | 170 |
| Commercial | 2017 | 10 |
| Health | 2018 | 500 |
| Commercial | 2015 | 180 |
| Health | 2020 | 100 |
| Commercial | NULL | 400 |
| Health | NULL | 970 |

## Q 10) Give the sales of all state stores for product categories c1, for all years 2018, 2017, 2016, and 2015

-- Ans.

-- PGADMIN 3,4

SELECT

state,

SUM (total\_sales) FILTER (WHERE year=2018) AS "2018",

SUM (total\_sales) FILTER (WHERE year=2017) AS "2017",

SUM (total\_sales) FILTER (WHERE year=2016) AS "2016",

SUM (total\_sales) FILTER (WHERE year=2015) AS "2015"

FROM c1

WHERE category = 'Health'

GROUP BY state ;

--OUTPUT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| state | 2018 | 2017 | 2016 | 2015 |
| GJ | 150 | 620 | NULL | NULL |
| MH | 500 | 200 | 170 | NULL |

--OR

SELECT

state,

SUM (total\_sales) FILTER (WHERE year=2018) AS "2018",

SUM (total\_sales) FILTER (WHERE year=2017) AS "2017",

SUM (total\_sales) FILTER (WHERE year=2016) AS "2016",

SUM (total\_sales) FILTER (WHERE year=2015) AS "2015"

FROM c1

WHERE category = ‘Commercial’

GROUP BY state ;

--OUTPUT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| state | 2018 | 2017 | 2016 | 2015 |
| GJ | 400 | 160 | NULL | NULL |
| MH | 10 | 10 | NULL | 180 |

## Q 11) Calculate total no. of cuboids for above problem.

L1=5-1 = 4, L2=4-1=3, L3=6-1=5

N=3

**So, this cube will contain (5\*4\*6) = 120 cuboids**