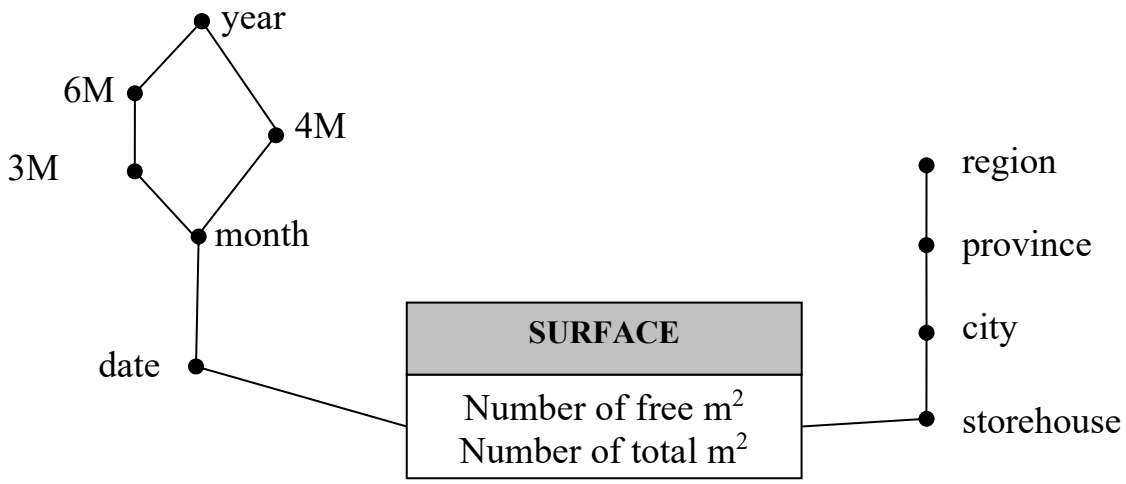
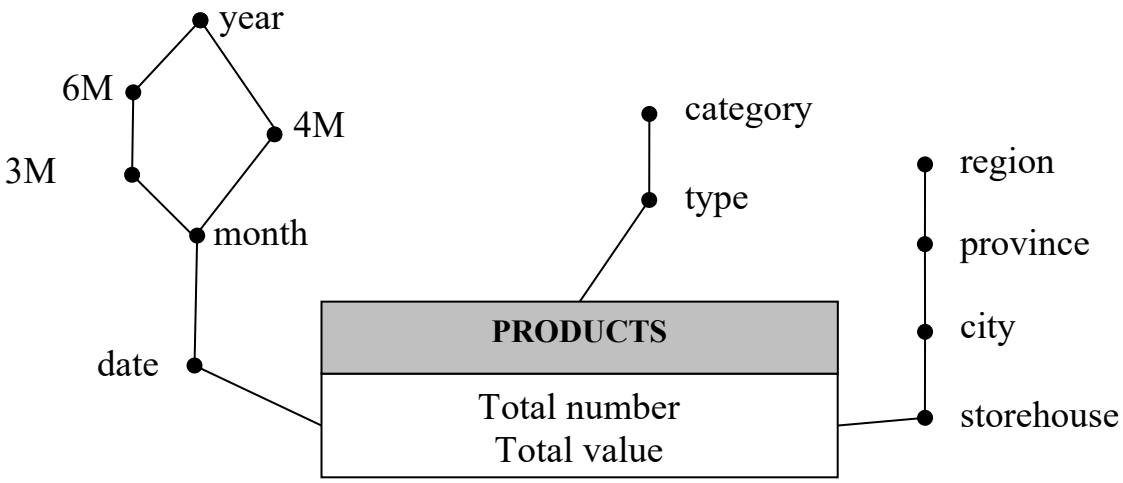
Data Base and Data Mining Group of Politecnico di Torino

Data Science and Database Technology

Politecnico di Torino

Conceptual design





Logical design

Primary keys are underlined.

Facts

SURFACE (storehouseID, timeID, m2free, m2tot)
PRODUCTS (storehouseID, timeID, typeID, totNumber, totValue)

Dimensions

TIME (timeID, date, month, 3M, 4M, 6M, year) → shared both facts
TYPES (typeID, type, category) → only for Products fact
STOREHOUSES (storehouseID, storehouse, city, province, region) → shared both facts

Query A

```
select
    storehouse, date, sum(totValue),
    avg( sum(totValue) ) over (partition by storehouse order by date range between interval '6' day preceding and current row)
from
    products p, storehouses sh, time t
where
    p.storehouseID=sh.storehouseID and p.timeID=t.timeID and
    t.3M=1/2013 and sh.city='Turin'
group by
    storehouseID, storehouse, date;
```

NB: averaging the daily total value over the last week could be done using the *sum(sum(totValue)/7)* expression, which handles missing days as if their *totValue* were 0, while the proposed solution fills missing values with the week average; furthermore note that *totValue* is a level measure, thus there should be no missing values in the data warehouse.

Query B

```
select
    city, date,
    sum(m2free)/sum(m2tot)*100,
    rank() over (order by sum(m2free)/sum(m2tot) asc)
from
    surface s, storehouses sh, time t
where
    s.storehouseID=sh.storehouseID and s.timeID=t.timeID and t.year=2014
group by
    city, date;
```

Query C

```
select
    storehouse, date, (m2free/m2tot)*100,
from
    products p, storehouses sh, time t
where
    p.storehouseID=sh.storehouseID and p.timeID=t.timeID and
    t.month>=1/2014 and t.month<=6/2014
group by
    storehouseID, storehouse, date;
```

Query D

```
select
    storehouse, month,
    sum(totValue)/count(distinct date)
from
    products p, storehouses sh, time t
where
    p.storehouseID=sh.storehouseID and p.timeID=t.timeID and t.year=2013
group by
    storehouseID, storehouse, month;
```

Alternative solution:

```
select distinct
    storehouse, month,
    avg( sum(totValue) ) over (partition by storehouse, month)
from
    products p, storehouses sh, time t
where
    p.storehouseID=sh.storehouseID and p.timeID=t.timeID and t.year=2013
group by
    storehouseID, storehouse, date, month;
```

NB: the DISTINCT command does **not** remove rows with the same storehouse; it removes duplicate rows considering all attribute values of each row.

Query E

```
select
    region, sum(totValue)/count(distinct date)
from
    products p, storehouses sh, time t
where
    p.storehouseID=sh.storehouseID and p.timeID=t.timeID and t.year=2015
group by
    region;
```

Alternative solution:

```
select distinct
    region, avg(sum(totValue)) over (partition by region)
from
    products p, storehouses sh, time t
where
    p.storehouseID=sh.storehouseID and p.timeID=t.timeID and t.year=2015
group by
    region, date;
```

Query F

```
select distinct
    region, month,
    avg(sum(m2free)/sum(m2tot)*100) over (partition by region, month)
from
    surface s, storehouses sh, time t
where
    s.storehouseID=sh.storehouseID and s.timeID=t.timeID and t.year=2014
group by
    region, month, date;
```

Materialized view

Query 1

```
select
    province, month,
    sum(totValue),
    sum(totNumber)
from
    products p, storehouses s, time t
where
    p.storehouseID=s.storehouseID and p.timeID=t.timeID and (t.year=2015 or t.year=2015) and region='Piedmont'
group by
    province, month;
```

Query 2

```
select
    province, 6M,
    sum(totValue)/count(distinct month),
from
    products p, storehouses s, time t
where
    p.storehouseID=s.storehouseID and p.timeID=t.timeID and t.year=2013
group by
    province, 6M;
```

Query 3

```
select
    province, 3M,
    sum(totValue),
from
    products p, storehouses s, time t
where
    p.storehouseID=s.storehouseID and p.timeID=t.timeID and (t.year=2015 or t.year=2015) and region='Piedmont'
group by
    region, 3M;
```

View

```
select
    province, month, region, 3M, 6M, year
    sum(totValue),
    sum(totNumber)
from
    products p, storehouses s, time t
where
    p.storehouseID=s.storehouseID and p.timeID=t.timeID
group by
    province, month, region, 3M, 6M, year;
```

Reductions:
Storehouses → Province
Date → Month
No type dimension: reduction of 1/100

Card: 90 x 12 = 1080 << 18250k → a materialized view on this query is convenient.