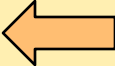


RELATIONAL DBMS EXTENSIONS FOR DW

- SQL extensions
- Index and storage structures
- Star query physical plans
- Materialized views
- Optimization techniques for star queries with grouping and aggregations
- Query rewriting to use materialized views 

THE QUERY REWRITING PROBLEM

A **materialized view** V is a query with a result that is materialized and stored in a table.

The **query rewriting problem**: *given a query Q and a materialized view V , is it possible to rewrite the query (plan of) Q using V ?*

The query Q is not rewritten if the plan without V has a lower estimated computational cost than the plan using V , or if the plan with another materialized view has a lower cost.

- In general, a query rewritten in terms of a materialized view improves the execution of the query, because most of the query result has been precomputed.
- We omit cost-based evaluation of query plans in this course, see *Advanced Databases*.

Who write a query is not necessarily aware of materialized views, but the query optimizer consider the possibility of rewrite a query to use one of the available materialized views.

THE STAR SCHEMA: ASSUMPTIONS

Foreign and primary keys are the only attributes of the star schema with the same names and data types.

The fact and dimensional tables have attributes without null values.

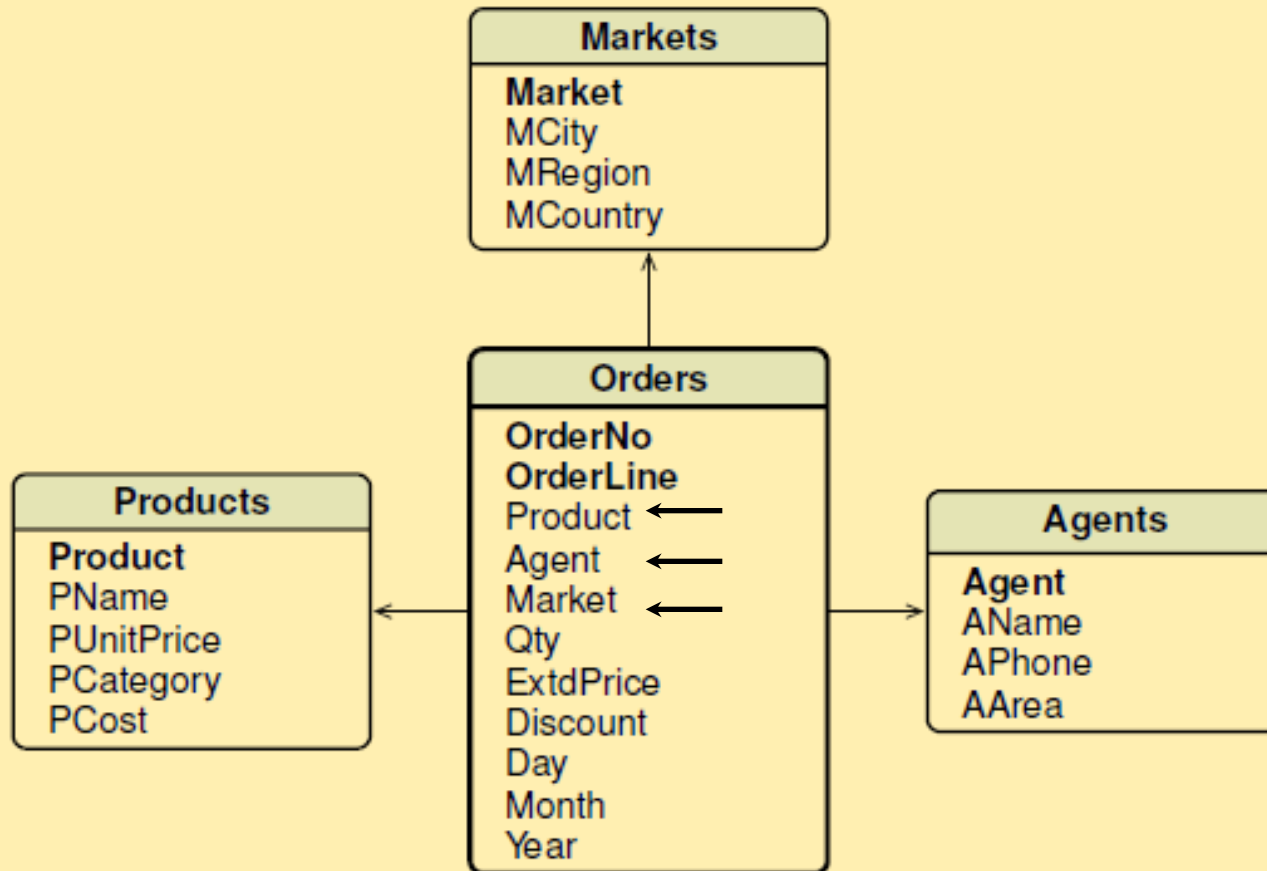
Star query only, namely natural joins of fact and dimension tables

- The joins are lossless and non-duplicating (a row in the fact table matches with one and only one row in the joined dimension tables)

Query rewriting in commercial DBMS make a number of such assumptions, e.g., by restricting the forms of admissible materialized views

QUERY REWRITING TO USE MATERIALIZED VIEWS

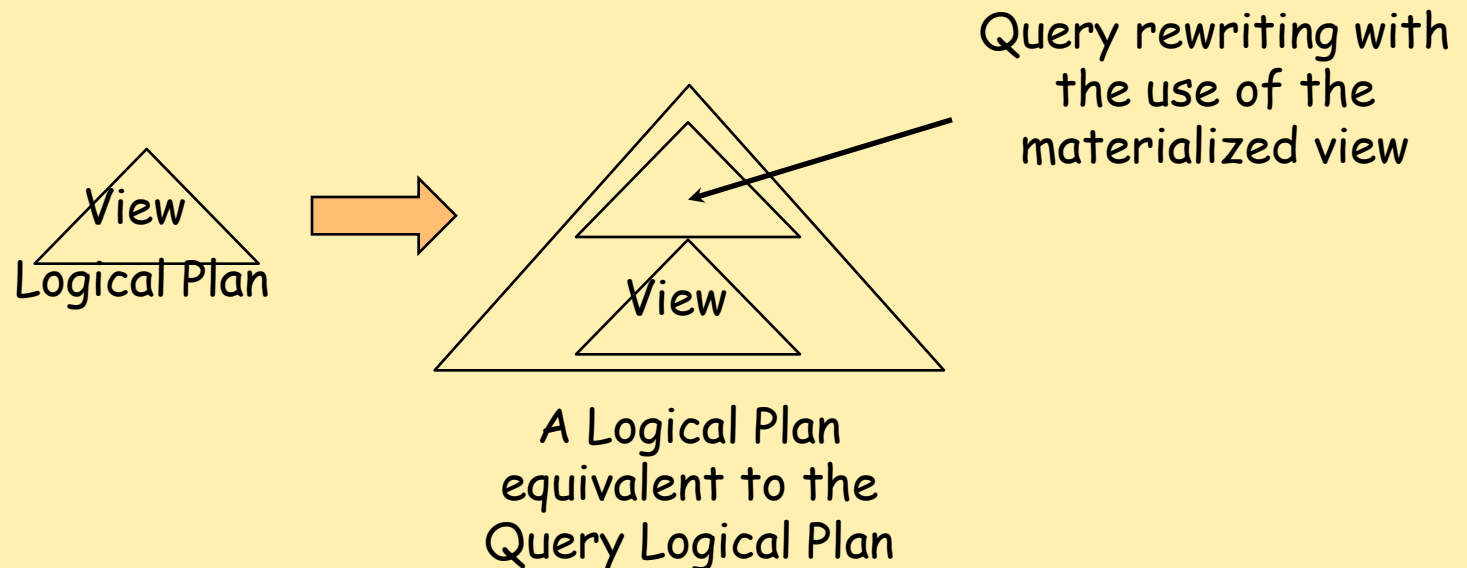
Let us consider a database with a **Star Schema**.



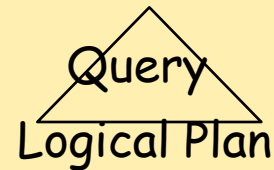
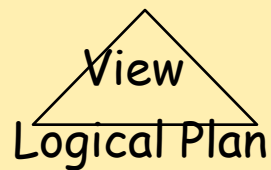
QUERY REWRITING TO USE MATERIALIZED VIEWS: APPROACH 1



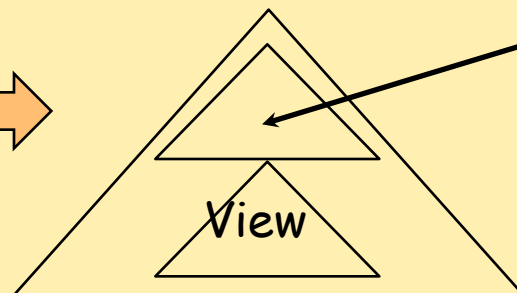
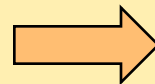
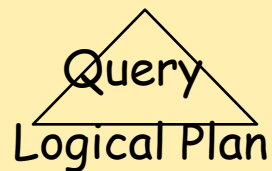
First approach. Add a logical plan to the top of the View logical plan so that the global plan is equivalent to the Query logical plan.



QUERY REWRITING TO USE MATERIALIZED VIEWS: APPROACH 2



Second approach. Transform the Query logical plan so that a **bottom portion** of it is **equivalent to the View logical plan**.



Query rewriting with the use of the materialized view

The Query Logical Plan with a bottom equivalent to View Logical Plan

EXAMPLE

A trivial case

$$Q = \sigma_{A \gamma_{SUM(C)} AS S}(R) \quad V = \sigma_{A \gamma_{SUM(C)} AS S}(R)$$

A simple case

$$Q = \sigma_{A \gamma_{SUM(C)} AS SQ}(R) \quad V = \sigma_{A, B \gamma_{SUM(C)} AS SV}(R)$$

Assume that $A \rightarrow A, B$ and $A, B \rightarrow A$ hold.

$$|V| = |Q| ?$$

In general, that $g(Q) \rightarrow g(V)$ and $g(V) \rightarrow g(Q)$ hold.

Is Q rewritable using V ?

R

A	B	C
a	2	2
a	2	3
b	3	4
b	3	5

$$Q = \pi_{A, SV} \sigma_{SQ} \left(\underbrace{\sigma_{A, B \gamma_{SUM(C)} AS SV}(R)}_V \right)$$

EXAMPLE

Another case

$$Q = A \gamma_{\text{SUM}(C)} \text{ASSQ}(R)$$

$$V = A, B \gamma_{\text{SUM}(C)} \text{ASSV}(R)$$

Assume that $g(V) \rightarrow g(Q)$ only holds.

$|V| > |Q|$?

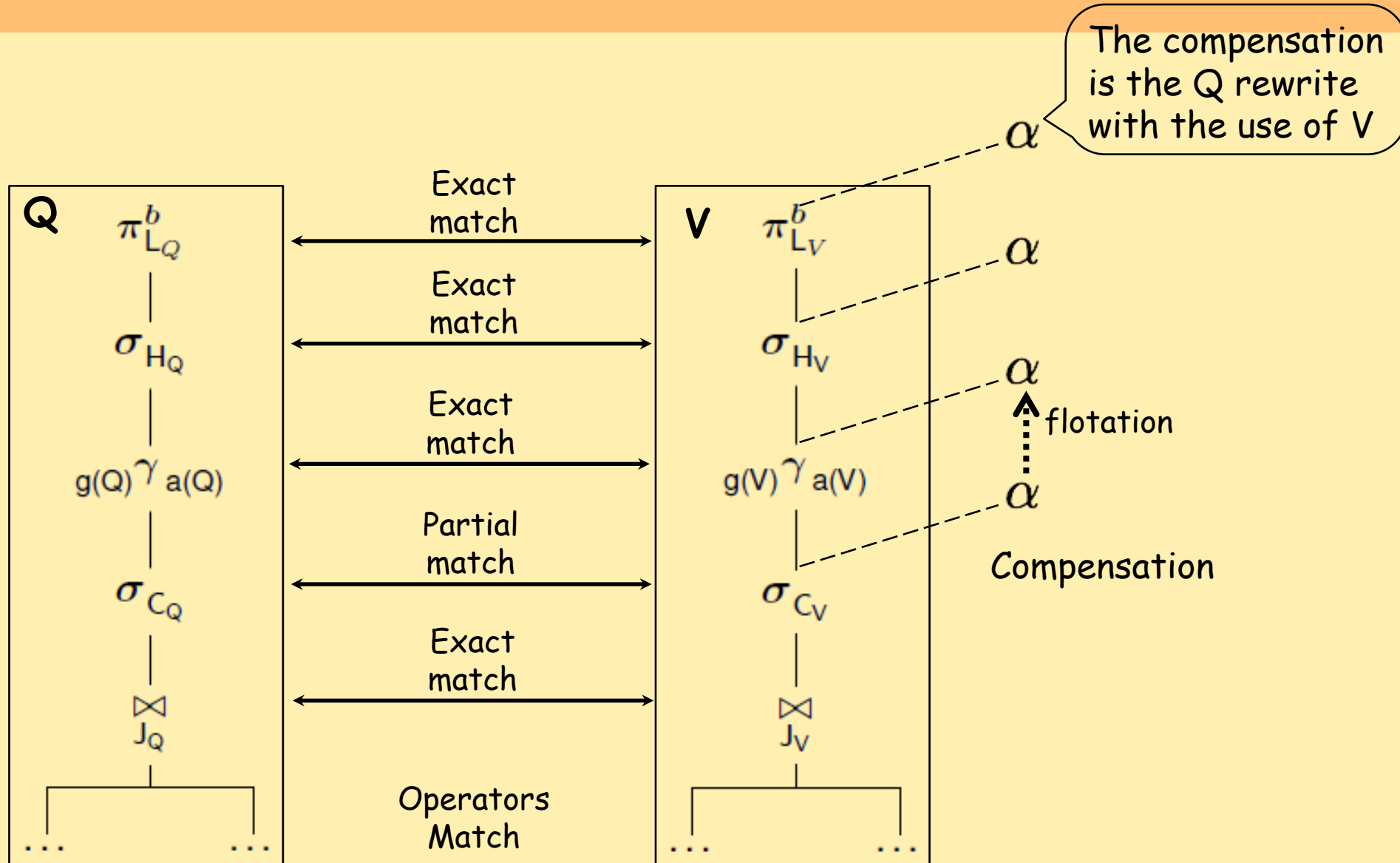
Is Q rewritable using V ?

$$Q = A \gamma_{\text{SUM}(SV)} \text{ASSQ} \left(\underbrace{A, B \gamma_{\text{SUM}(C)} \text{ASSV}(R)}_V \right)$$

R

A	B	C
a	2	2
a	2	3
a	3	2
a	3	3
b	2	4
b	2	5
b	3	4
b	3	5

FIRST APPROACH: MATCHES AND COMPENSATIONS



FIRST APPROACH: MATCHES AND COMPENSATIONS

The algorithm idea.

The Q and V logical plans operators are pair-wise compared bottom-up, and if they are not **equivalent** a **compensation** is added to the view operator, that is a set of logical operations that have to be performed on the view operator to produce the same result.

When the matching view operator **has an operand with a compensation**, it must float on the operator because it must be included in the compensation for the operator match.

The final compensation on the view root is the rewriting of the query

EXAMPLE: IS Q REWRITABLE WITH V ?

```
SELECT
FROM
WHERE
```

```
Market, Year, C
Markets NATURAL JOIN V
MRegion = 'Toscana';
```

Rewritten Q

Rewritable!

Market, Year γ COUNT(*) AS C

$\sigma_{MRegion = 'Toscana'}$

Orders Markets

$\pi^b_{Market, Year, C}$
 $\sigma_{MRegion = 'Toscana'}$

Markets Market, Year γ COUNT(*) AS C

$\sigma_{MRegion = 'Toscana'}$

Markets Orders

```
SELECT Market, Year, COUNT(*) AS C
FROM Orders NATURAL JOIN Markets
WHERE MRegion = 'Toscana'
GROUP BY Market, Year;
```

Q

```
SELECT Market, Year, COUNT(*) AS c
FROM Orders
GROUP BY Market, Year;
```

V

FIRST APPROACH

$$A_Q = \gamma_{G_Q}(\sigma_{C_Q}(\bowtie R_Q)) \quad A_V = \gamma_{G_V}(\sigma_{C_V}(\bowtie R_V))$$
$$R_V \subseteq R_Q$$

REWRITING ALGORITHM

1. (\bowtie)

If the joins do not **match** (ie, $R_V \subset R_Q$), a **compensation** is added to the view operator as follows:

Let $W = R_Q - R_V$ be the tables in Q but not in V , then the compensation is the

$$\text{join } \alpha_{\bowtie} = (\bowtie W (A_V(\bowtie)))$$


Root of compensation tree = $\bowtie R_V$

Condition: the compensation can float on γ_{G_V} if G_V contains the foreign keys for the tables W . **Otherwise Q is not rewritable**

EXAMPLE: IS Q REWRITABLE WITH V ?

Rewritable!

Market, Year γ COUNT(*) AS C

$\sigma_{MRegion = 'Toscana'}$



Orders Markets

Market, Year γ COUNT(*) AS C



Markets

Orders

} $A_V(\bowtie)$

```
SELECT Market, Year, COUNT(*) AS C
FROM Orders NATURAL JOIN Markets
WHERE MRegion = 'Toscana'
GROUP BY Market, Year;
```

Q

```
SELECT Market, Year, COUNT(*) AS c
FROM Orders
GROUP BY Market, Year;
```

V

FIRST APPROACH

$$A_Q = \gamma_{G_Q}(\sigma_{C_Q}(\bowtie R_Q)) \quad A_V = \gamma_{G_V}(\sigma_{C_V}(\bowtie R_V))$$
$$R_V \subseteq R_Q$$

REWRITING ALGORITHM

2. (σ)

If on the operand exists a compensation it floats on the operator.

If the selections do not **match**, and $C_Q = C_V \wedge C$, then **compensation** to add is

$$\alpha_\sigma = \sigma_C(\underbrace{A_V(\sigma_V)}_{\text{Root of compensation tree}})$$

Root of compensation tree

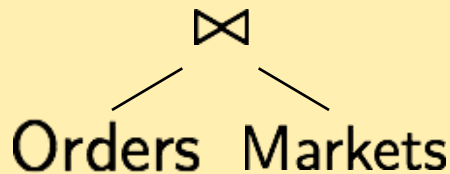
Condition: the compensation can float on γ_{G_V} if C uses only attributes in G_V or attributes of tables with foreign keys in G_V . **Otherwise Q is not rewritable**

EXAMPLE: IS Q REWRITABLE WITH V ?

Rewritable!

Market, Year γ COUNT(*) AS C

$\sigma_{MRegion = 'Toscana'}$

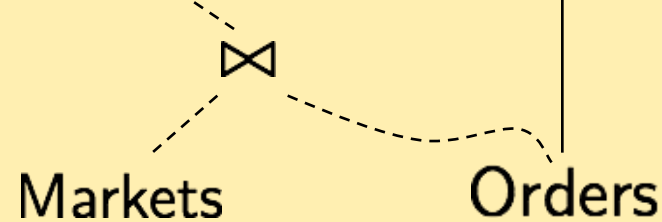


$A_V(\sigma_V)$



Market, Year γ COUNT(*) AS C

$\sigma_{MRegion = 'Toscana'}$



```
SELECT Market, Year, COUNT(*) AS C
FROM Orders NATURAL JOIN Markets
WHERE MRegion = 'Toscana'
GROUP BY Market, Year;
```

Q

```
SELECT Market, Year, COUNT(*) AS c
FROM Orders
GROUP BY Market, Year;
```

V

FIRST APPROACH

$$A_Q = \gamma_{G_Q}(\sigma_{C_Q}(\bowtie R_Q)) \quad A_V = \gamma_{G_V}(\sigma_{C_V}(\bowtie R_V))$$
$$R_V \subseteq R_Q$$

REWRITING ALGORITHM

see Lecture Notes for other aggregates

3. $(_{g(Q)}\gamma_{SUM(A) AS S})$

If on the operand exists a compensation it floats on the operator.

Case WITHOUT grouping

The rewriting is without grouping when the groupings in Q and V partition data into the same number of groups, e.g., if

$$g(Q) \rightarrow g(V) \quad \text{and} \quad g(V) \rightarrow g(Q)$$

Root of compensation tree

The **compensation** to add is $\alpha = \pi^b_{g(Q) \cup \{S\}}(\overbrace{A_V(\gamma_{G_V})})$

EXAMPLE: IS Q REWRITABLE WITH V ?

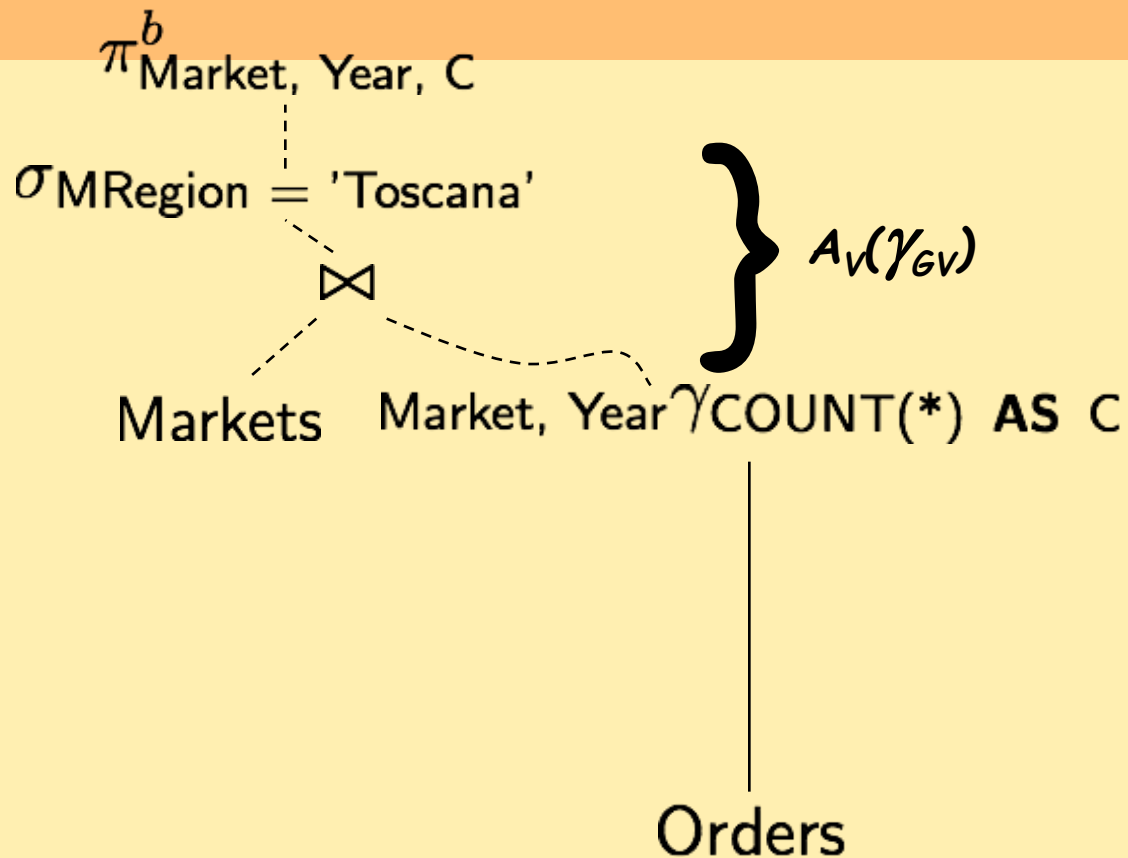
Rewritable!

Market, Year γ COUNT(*) AS C

$\sigma_{MRegion = 'Toscana'}$



Orders Markets



```
SELECT Market, Year, COUNT(*) AS C
FROM Orders NATURAL JOIN Markets
WHERE MRegion = 'Toscana'
GROUP BY Market, Year;
```

Q

```
SELECT Market, Year, COUNT(*) AS c
FROM Orders
GROUP BY Market, Year;
```

V

FIRST APPROACH

$$A_Q = \gamma_{G_Q}(\sigma_{C_Q}(\bowtie R_Q)) \quad A_V = \gamma_{G_V}(\sigma_{C_V}(\bowtie R_V))$$
$$R_V \subseteq R_Q$$

REWRITING ALGORITHM

3. $(g(Q) \gamma_{SUM(A) AS S})$

If on the operand exists a compensation it floats on the operator.

Case WITH grouping

The rewriting is without grouping when the groupings V partitions data into more groups than the grouping in Q, e.g., if

$$g(V) \rightarrow g(Q)$$

Root of compensation tree

The **compensation** to add is $\alpha = g(Q) \gamma_{SUM(S') AS S} (\overbrace{A_V(\gamma_{G_V})})$
where **SUM(A) AS S'** is in G_V .

EXERCISE: IS Q REWRITABLE WITH V ?

Rewritable ?

Market γ COUNT(*) AS C

σ MRegion = 'Toscana'



Orders Markets

Market, Year γ COUNT(*) AS C

Orders

```
SELECT Market, COUNT(*) AS C
FROM Orders NATURAL JOIN Markets
WHERE MRegion = 'Toscana'
AND Year = 2013
GROUP BY Market;
```

Q

```
SELECT Market, Year, COUNT(*) AS c
FROM Orders
GROUP BY Market, Year;
```

V

EXERCISE AT HOME: IS Q REWRITABLE WITH V ?

Rewritable ?

Market γ COUNT(*) AS C

σ MRegion = 'Toscana' AND Year = 2013



Orders Markets

Market, Year γ COUNT(*) AS C

Orders

```
SELECT Market, COUNT(*) AS C
FROM Orders NATURAL JOIN Markets
WHERE MRegion = 'Toscana'
      AND Year = 2013
GROUP BY Market;
```

Q

```
SELECT Market, Year, COUNT(*) AS c
FROM Orders
GROUP BY Market, Year;
```

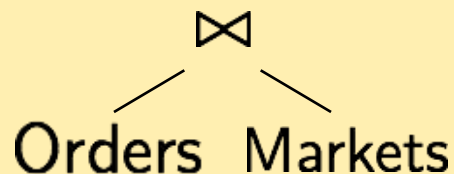
V

EXERCISE: IS Q REWRITABLE WITH V ?

Rewritable ?

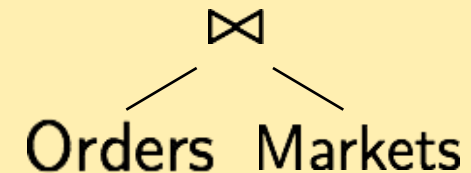
MCity γ COUNT(*) AS C

σ MRegion = 'Toscana' AND Year = 2013



Market, Year γ COUNT(*) AS C

σ MRegion = 'Toscana'



```
SELECT MCity, COUNT(*) AS C
FROM Orders NATURAL JOIN Markets
WHERE MRegion = 'Toscana'
      AND Year = 2013
GROUP BY MCity; Q
```

```
SELECT Market, Year, COUNT(*) AS c
FROM Orders NATURAL JOIN Markets
WHERE MRegion = 'Toscana'
GROUP BY Market, Year; V
```

$$A_Q = \gamma_{G_Q}(\sigma_{C_Q}(\bowtie R_Q)) \quad A_V = \gamma_{G_V}(\sigma_{C_V}(\bowtie R_V))$$

$$R_V \subseteq R_Q$$

REWRITING ALGORITHM

3. $(g(Q)\gamma_{SUM(A) AS S})$

If on the operand exists a compensation it floats on the operator.

Case WITH grouping

The rewriting is without grouping when the groupings V partitions data into more groups than the grouping in Q , e.g., if

$$g(V) \rightarrow g(Q)$$

Root of compensation tree

The **compensation** to add is $\alpha = g(Q)\gamma_{SUM(S') AS S}(\overbrace{A_V(\gamma_{G_V}) \bowtie R})$

where the $g(Q)$ attributes missing in $A_V(\gamma_{G_V})$ can be retrieved with a lossless and non-duplicating join of $A_V(\gamma_{G_V})$ with R , and **SUM(A) AS S'** is in G_V .

For example, $A_V(\gamma_{G_V})$ contains the foreign keys f_k of R .

IS Q REWRITABLE WITH V ?

Q: **SELECT** Market, SUM(Qty)
FROM Orders
GROUP BY Market;

V: **SELECT** Market, SUM(Qty)
FROM Orders **NATURAL JOIN** Markets
WHERE MRegion = 'Toscana'
GROUP BY Market;

Q: **SELECT** SUM(Qty) **AS** Q
FROM Orders
GROUP BY Market;

V: **SELECT** SUM(Qty) **AS** Q
FROM Orders;

Q: **SELECT** SUM(Qty) **AS** Q
FROM Orders
GROUP BY Market;

V: **SELECT** MAX(Qty) **AS** Q
FROM Orders
GROUP BY Market;

SECOND APPROACH TO QUERY REWRITING: EXAMPLE

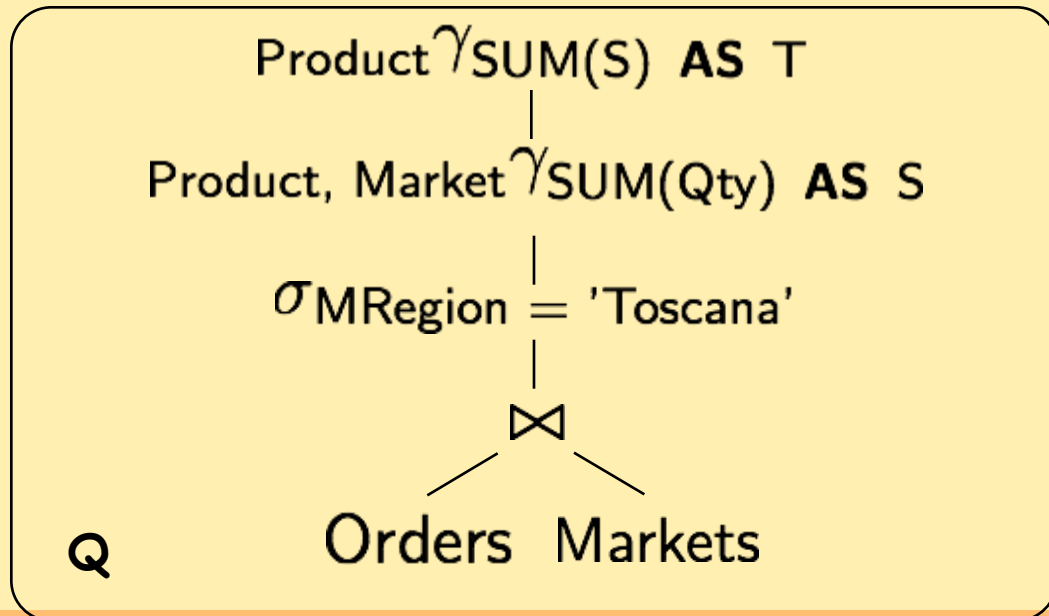
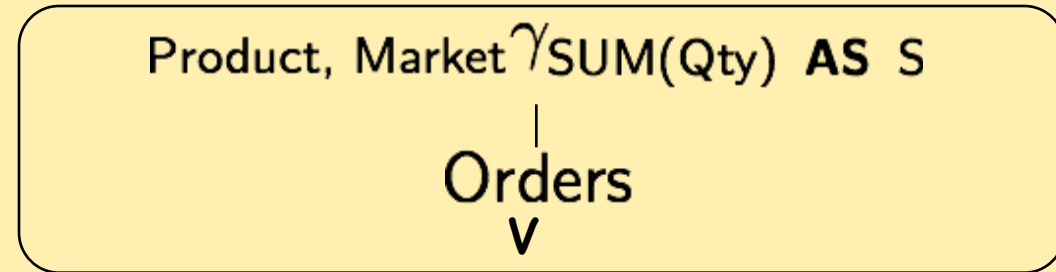
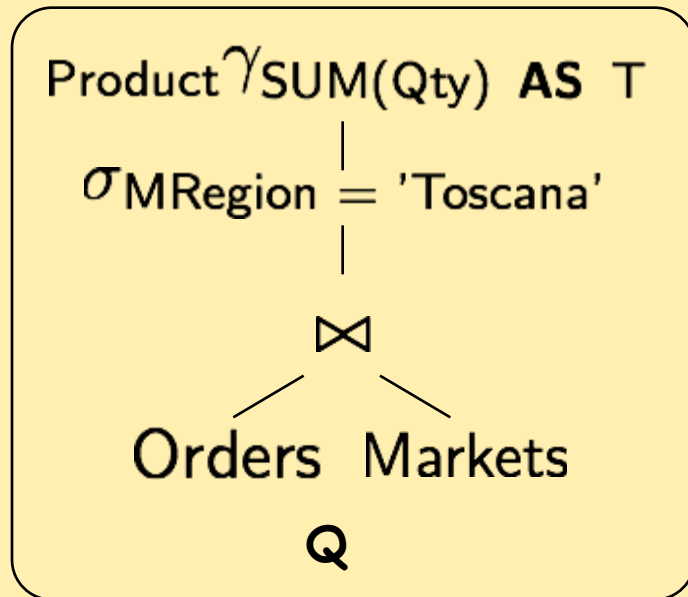
Q: **SELECT** Product, SUM(Qty) **AS** T
FROM Orders **NATURAL JOIN** Markets
WHERE MRegion = 'Toscana'
GROUP BY Product;

V: **SELECT** Product, Market,
SUM(Qty) **AS** S
FROM Orders
GROUP BY Product, Market;

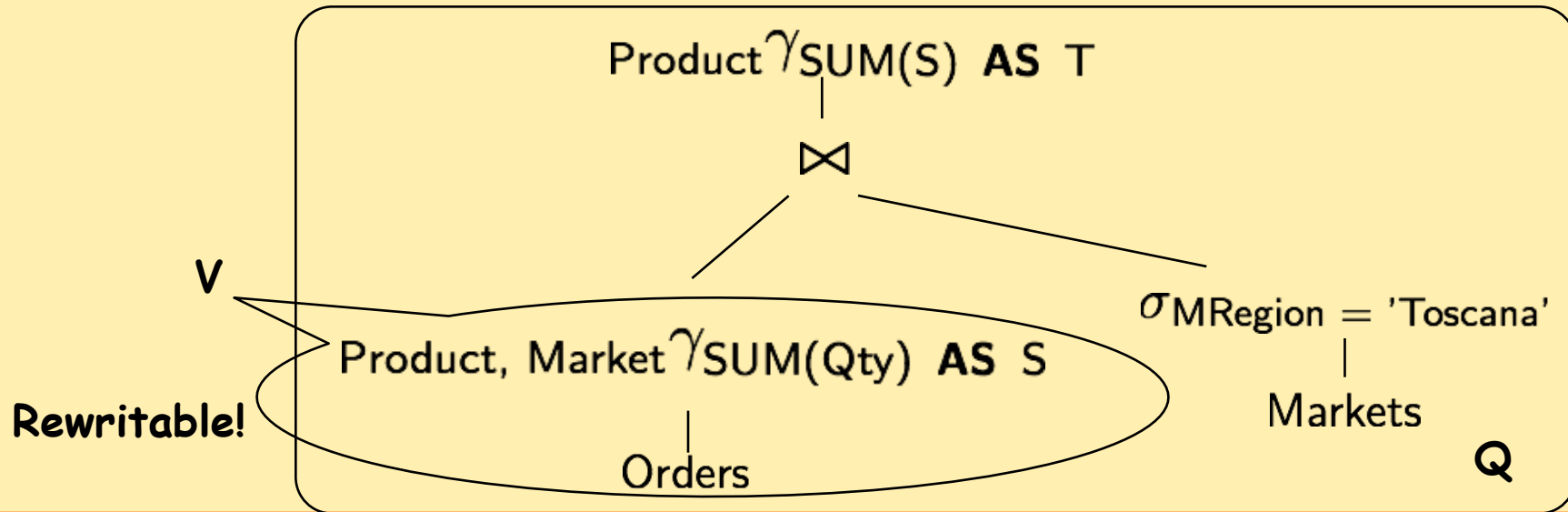
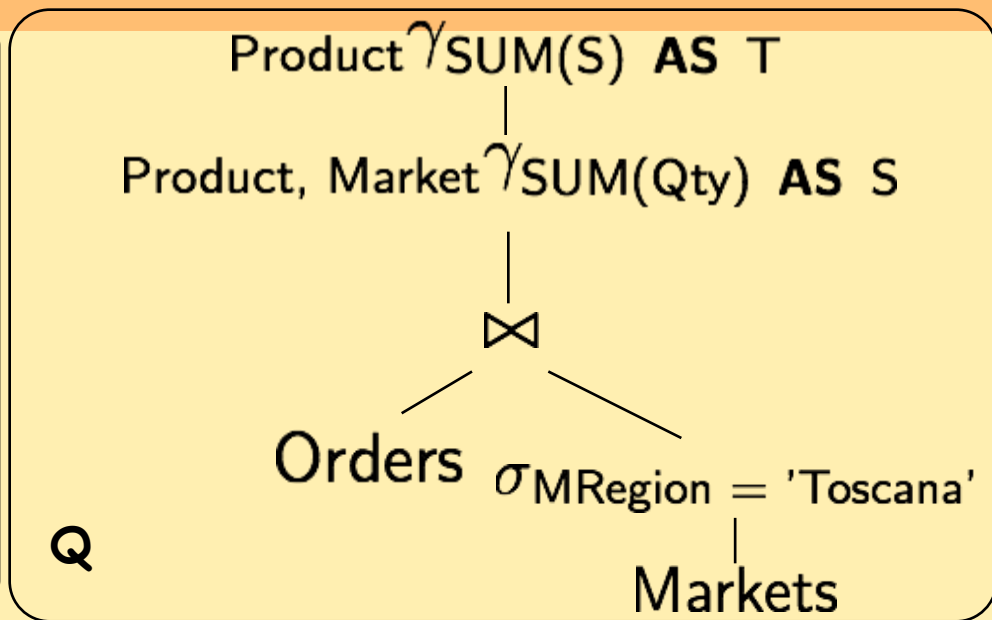
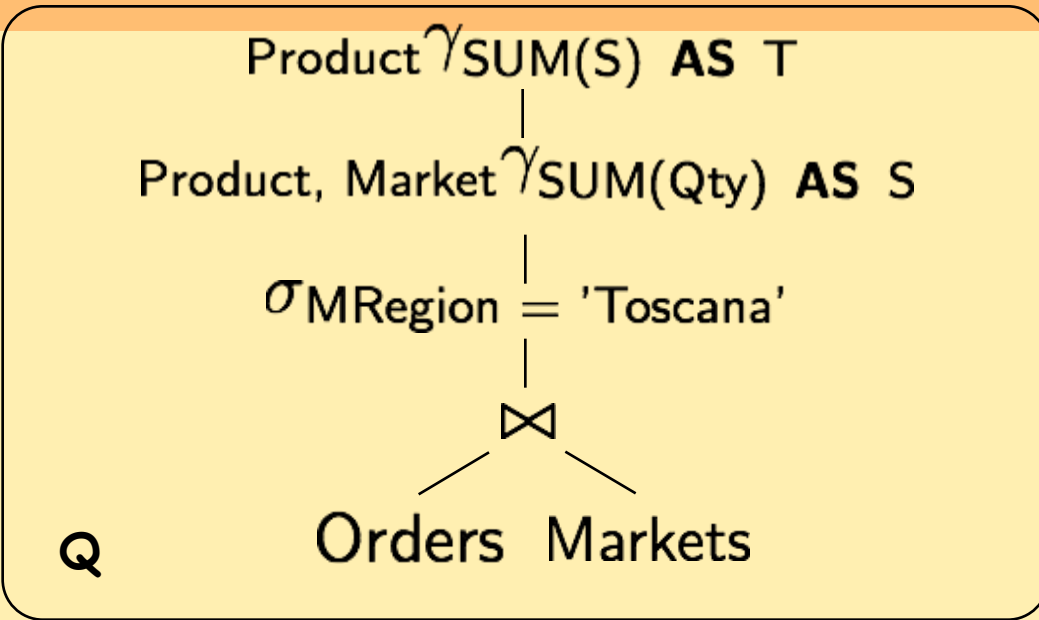
Rewritable?

SECOND APPROACH TO QUERY REWRITING: EXAMPLE

Transform the Query logical plan so that a **bottom portion** of it is equivalent to the **View logical plan**.



SECOND APPROACH TO QUERY REWRITING: EXAMPLE



SECOND APPROACH TO QUERY REWRITING: EXAMPLE

Q: **SELECT** Product, SUM(Qty) **AS** T
FROM Orders **NATURAL JOIN** Markets
WHERE MRegion = 'Toscana'
GROUP BY Product;

V: **SELECT** Product, Market,
SUM(Qty) **AS** S
FROM Orders
GROUP BY Product, Market;

Rewritable!

Rewriting of Q using the second approach (but also using the first approach)

SELECT Product, SUM(S) **AS** T
FROM V **NATURAL JOIN** Markets
WHERE MRegion = 'Toscana'
GROUP BY Product;

In general, the first and the second approach may yield different rewriting!

STRUCTURE OF EXAMS

- [DW] A. Albano, S. Ruggieri. [Decision Support Databases Essentials](#), University of Pisa, 2 December 2021
- [DB] A. Albano. [DB Essentials](#) and [solutions to exercises](#), University of Pisa, 1 December 2020. This is a (partial) translation (in English) from the book [Fondamenti di basi di dati](#) (in Italian, free download).
- Examples of [written exams with solutions](#) and [written exam](#).

- **Written test (2 hours)**

- See website for

Examples of written text

Dates and registration

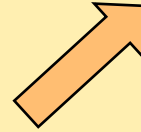
- Grading ≥ 17 admission to the oral part

- **Oral part**

- Discussion of written part
- Open questions on all topics of the course
- Questions/small exercises using JRS and/or SQL Server

- **Overall Decision Support Systems (12 ECTS)**

- Final grade is the average of the two modules DSD and LDS



TIME FOR FILLING STUDENT'S QUESTIONNAIRES

<https://esami.unipi.it>

The image shows a screenshot of the VALUTAMI website. The header features the university logo and the text 'VALUTAMI VALUTazione della didattica ed iscrizione agli esami'. The left sidebar has a menu with items: Home, Exams, Subscribe, Subscriptions, Calendar by course, Calendar by session, Evaluation (highlighted with a red box and a red arrow), Syllabi, and Support. The main content area displays a blue banner with the following text: 'Dal 25/01 e' disponibile il questionario relativo ai servizi e all'organizzazione della didattica. Tutti gli studenti sono inviati a partecipare esprimendo la propria opinione. Il questionario è anonimo, si compone di 13 domande e deve essere compilato una sola volta. Vai alla pagina'.