Overview of Topics

1. Introduction
   - Operational vs. Warehouse

2. Multidimensional Data
   - Data model & schema

3. Queries
   - OLAP Queries
   - CUBE Operator
   - Window Operator

4. Implementation Algorithms
   - Bitmap Index
   - MOLAP vs ROLAP

5. Materialized Views
   - Query answering using views
   - View maintenance
   - View selection

6. Constructing a Data Warehouse (ETL)
5. Views and Decision Support (25.8,9)

- In large databases, *precomputation* is necessary for fast response times
  - Examples: google, brain
- Example: Precompute daily sums for the cube.
  - What can be derived from those precomputations?
- These precomputed queries are called *materialized views*
  - (Another name: indexed views in SQL Server.)
**Materialized View Example**

**Mat. View**

```
CREATE VIEW DailySum(date, sumamt) AS
SELECT date, SUM(amt)
FROM Times Join Sales USING(timeid)
GROUP BY date
```

**Query**

```
SELECT week, SUM(amt)
FROM Times Join Sales USING(timeid)
GROUP BY week
```

**Modified Query**

```
SELECT week, SUM(sumamt)
FROM Times Join DailySum USING (date)
GROUP BY week
```
Views

- A view is a named query
- A materialized view is the stored result of a query

Why are views interesting?
Views

- A view is a named query

- A materialized view is the stored result of a query

Why are views interesting?

- Query optimization

- Independence of the physical and logical views

- Access rights management
Using a View for a Query

Query:

```
Select Advises.prof, Advises.student, Registered.quarter
From Registered, Teaches, Advises
Where Registered.c-number = Teaches.c-number and
    Registered.quarter = Teaches.quarter and
    Advises.prof = Teaches.prof and Advises.student =
    Registered.student and Registered.quarter >= "winter98"
```

V1:

```
Select Registered.student, Teaches.prof, Registered.quarter
From Registered, Teaches
Where Registered.c-number = Teaches.c-number and
    Registered.quarter = Teaches.quarter and
    Registered.quarter > "winter97".
```
Query:

Quarter \geq \text{“winter 98”}

V1

Quarter \geq \text{“winter 97”}
Query:

- Quarter \(\geq\) “winter 98”

V2

- Quarter \(\geq\) “winter 98”
Query:

Quarter >= “winter 98”

V3

Quarter >= “winter 99”
Refreshing Materialized Views

- How often should we refresh the materialized view?
- Many enterprises refresh warehouse data only weekly/nightly, so can afford to completely rebuild their materialized views.
- Others want their warehouses to be current, so materialized views must be updated incrementally if possible.
- Let's look at some simple examples.
Maintaining Materialized Views (25.10)

- Incremental view maintenance
  - make changes in view that correspond to changes in the base tables

- Example: View V = \texttt{SELECT a FROM R}
  - How is V modified if r is inserted to R?
  - How is V modified if r is deleted from R?
Maintaining Materialized Views

- Consider \( V = R \bowtie S \)
  - How is \( V \) modified if \( r \) is inserted to \( R \)?
  - How is \( V \) modified if \( r \) is deleted from \( R \)?

- Consider \( V = \text{SELECT } g, \text{ COUNT}(*) \text{ FROM } R \text{ GROUP BY } g \)
  - How is \( V \) modified if \( r \) is inserted to \( R \)?
  - How is \( V \) modified if \( r \) is deleted from \( R \)
  - What if the aggregate is \( \text{AVG}, \text{MIN}, \text{MAX} \)

- For more general cases, see [348] in the textbook