5. Materialized Views

Basically pre-computed queries to increase performance. They are directly **stored** in the data warehouse.

TODO

Physical Design

It focuses on designing the physical layout of the data warehouse.
Several index types to choose from, such as **bitmap**,**join**,**bitmapped join** and B^+**-tree**.
Also, materialized views should be exploitable by the query optimizer.

ETL - Extraction, Transformation and Loading

- **Prepares** data to be loaded into the data warehouse
- **Eased** by exploiting the staging area
- **Performed** when **DW** is first loaded and during **periodical** DW refresh.

Extraction

It's the data acquisition from sources.
Two methods to extract data:

- **static:** snapshot of operational data, typically performed during first DW population
- **incremental:** selection of updates that took place after last extraction
  - **immediate:** when changes happen in an operational database, these changes propagates immediately to DW
  - **deferred:** changes are first transferred into the staging area

The **extraction** depends on how operational data is collected:

- **historical:** all modifications are stored for a given time, used in bank transactions and it's operationally simple.
  Here a **deferred** approach is more appropriate.
- **partly historical:** only a limited number of states is stored and it's operationally complex.
- **transient:** it keeps only the current data state, used for stock inventory and it's operationally complex.
  Here an **immediate** approach is more appropriate.

Incremental extraction

Several **approaches available:**

- **application assisted:** data changes are captured by ad hoc applications
- **log based:** data changes are captured by log data, accessed by APIs
- **trigger based:** triggers are used to capture interesting data changes
- **timestamp based:** a timestamp is used to mark data changes