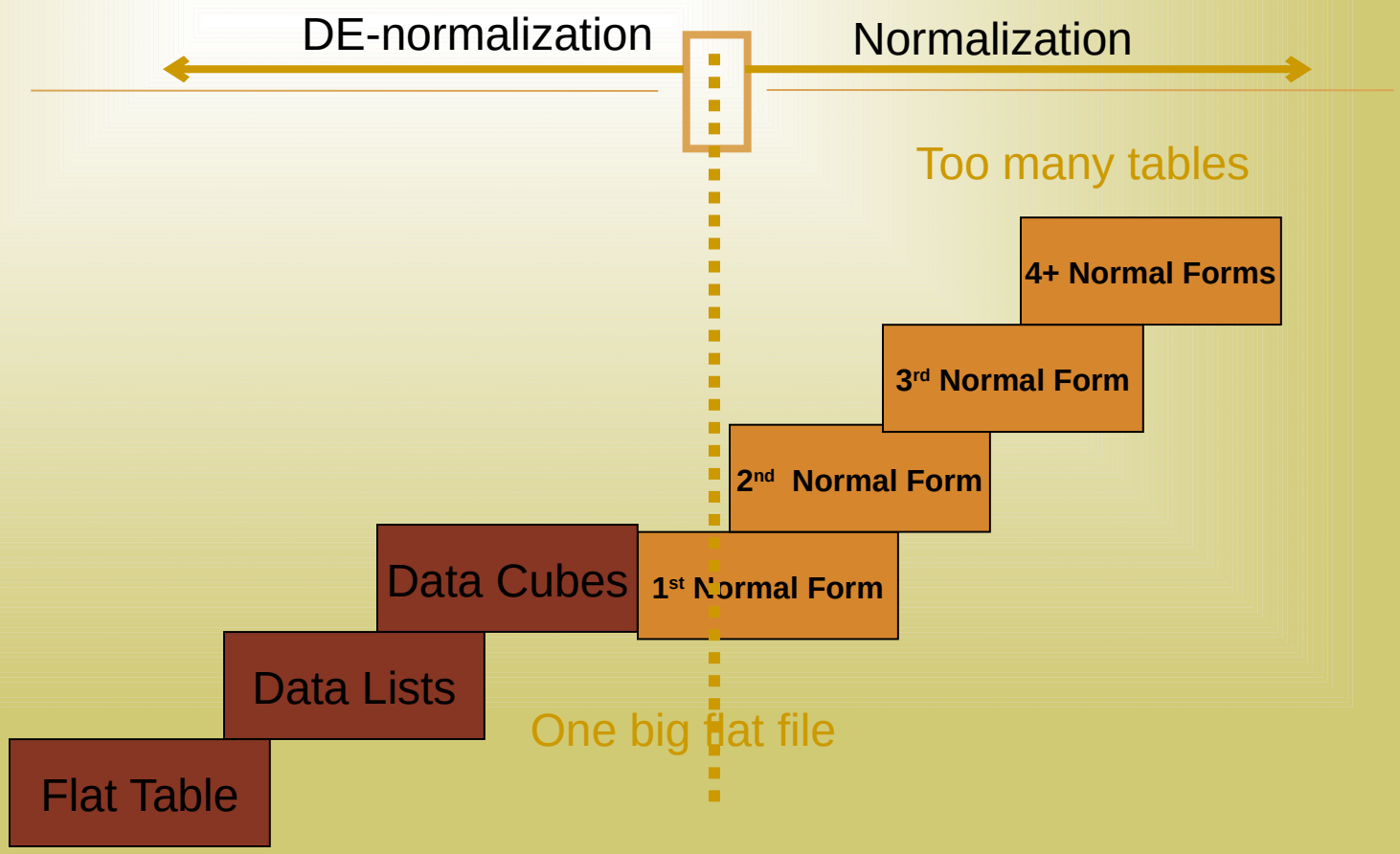
Data Warehousing and Data Mining

Sajid Majeed

Lecture-6

De-normalization

Striking a balance between “good” & “evil”



What is De-normalization?

- Denormalization is the process of taking a normalized database and modifying table structures to allow controlled redundancy for increased database performance.

What is De-normalization?

- the aim is to enhance performance without loss of information.

- Normalization is a rule of thumb in DBMS, but in DSS ease of use is achieved by way of denormalization.

- De-normalization comes in many flavors, such as combining tables, splitting tables, adding data etc., but all done very carefully.

Why De-normalization In DSS?

- Bringing “close” dispersed but related data items.
- Very early studies showed performance difference in orders of magnitude for different number de-normalized tables and rows per table.
- The level of de-normalization should be carefully considered.

How De-normalization improves performance?

De-normalization specifically improves performance by either:

- Reducing the number of tables and hence the reliance on joins, which consequently speeds up performance.
- Reducing the number of joins required during query execution.

4 Guidelines for De-normalization

1. Carefully do a cost-benefit analysis (frequency of use, additional storage, join time).

2. Do a data requirement and storage analysis.

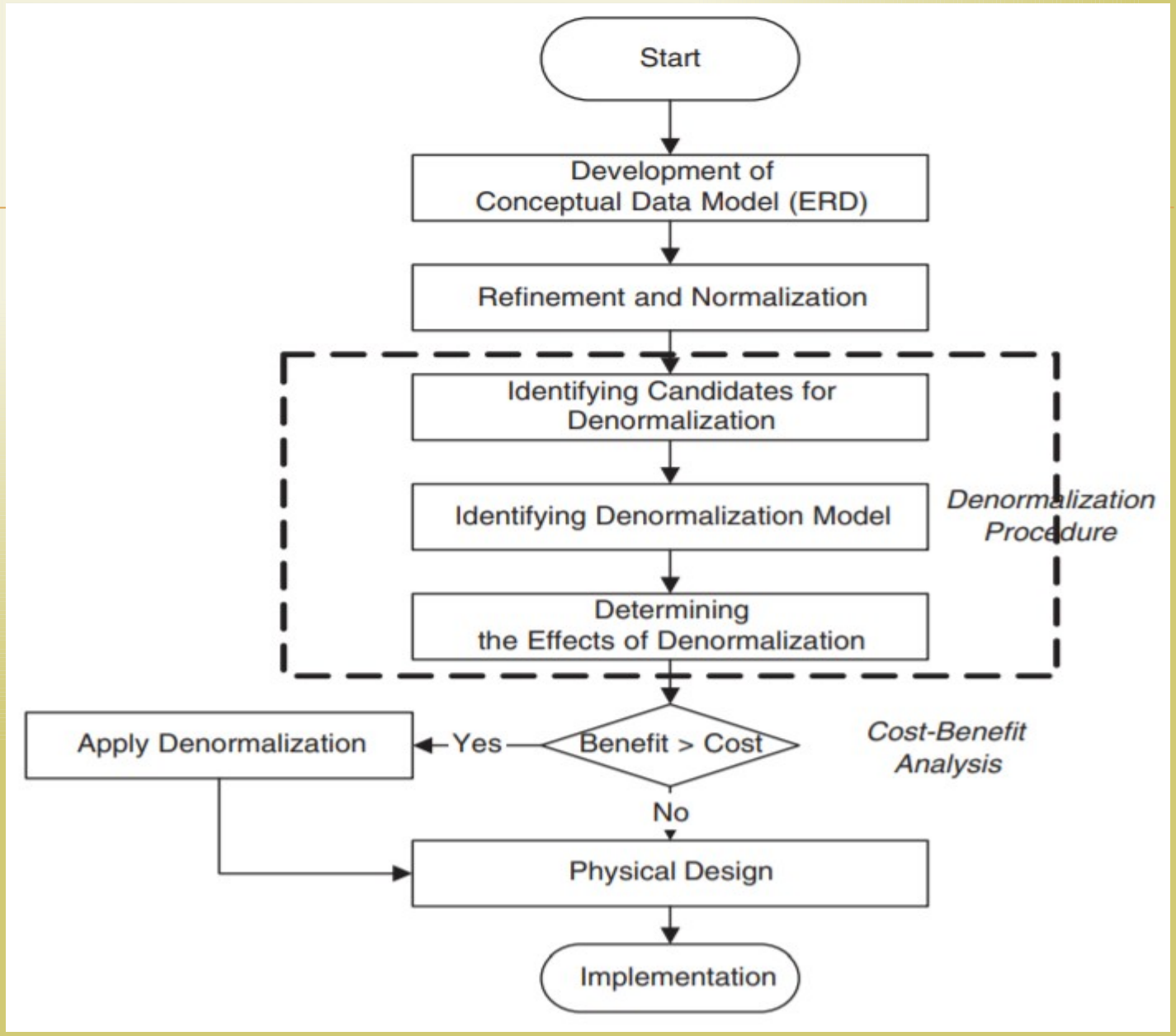
3. When in doubt, don’t denormalize.

Areas for Applying De-Normalization Techniques

- Dealing with the abundance of star schemas.
  - Fast aggregate (sum, average etc.) results and complicated calculations.
  - Multidimensional analysis (e.g. geography) in a complex hierarchy.
  - Dealing with few updates but many join queries.

De-normalization will ultimately affect the database size and query performance.

Database design cycle incorporating denormalization



Five principal De-normalization techniques

1. Collapsing Tables.
- Two entities with a One-to-One relationship.
- Two entities with a Many-to-Many relationship.

2. Splitting Tables (Horizontal/Vertical Splitting).

3. Pre-Joining.

4. Adding Redundant Columns (Reference Data).

5. Derived Attributes (Summary, Total, Balance etc).

Collapsing Tables

normalized

ColA	ColB

ColA	ColC

denormalized

ColA	ColB	ColC

- Reduced storage space.
- Reduced update time.
- Does not changes business view.
- Reduced foreign keys.