nextleap

SQL 6 Data Analyst Fellowship

Saksham Arora

Software Engineer

Microsoft

[Saksham Arora - Microsoft | LinkedIn]

Database design for a data analyst refers to the process of structuring and organizing data in a way that makes it easy to retrieve, analyze, and generate insights. A well-designed database allows a data analyst to efficiently query data, create reports, and make data-driven decisions.

Key Aspects of Database Design:

1. **Data Modeling:** Identifying the key entities (objects) and their relationships. This involves creating conceptual, logical, and physical models to represent how data will be stored and accessed.
2. **Normalization:** Organizing the data to minimize redundancy and improve data integrity. This process ensures that data is stored in separate tables based on their relationships.
3. **Indexing:** Creating indexes on columns that are frequently queried to improve performance.
4. **Constraints and Keys:** Using primary keys (unique identifiers for records) and foreign keys (linking relationships between tables) to maintain data accuracy and consistency.
5. **Data Warehousing:** Structuring the database for analytical purposes, typically through data warehouses or star/snowflake schemas that optimize reporting and analysis.

Approaches to processing data

OLTP

Online Transaction Processing

OLAP

Online Analytical Processing

OLAP vs. OLTP

	OLTP	OLAP
Purpose	support daily transactions	report and analyze data
Design	application-oriented	subject-oriented
Data	up-to-date, operational	consolidated, historical
Size	snapshot, gigabytes	archive, terabytes
Queries	simple transactions & frequent updates	complex, aggregate queries & limited updates
Users	thousands	hundreds

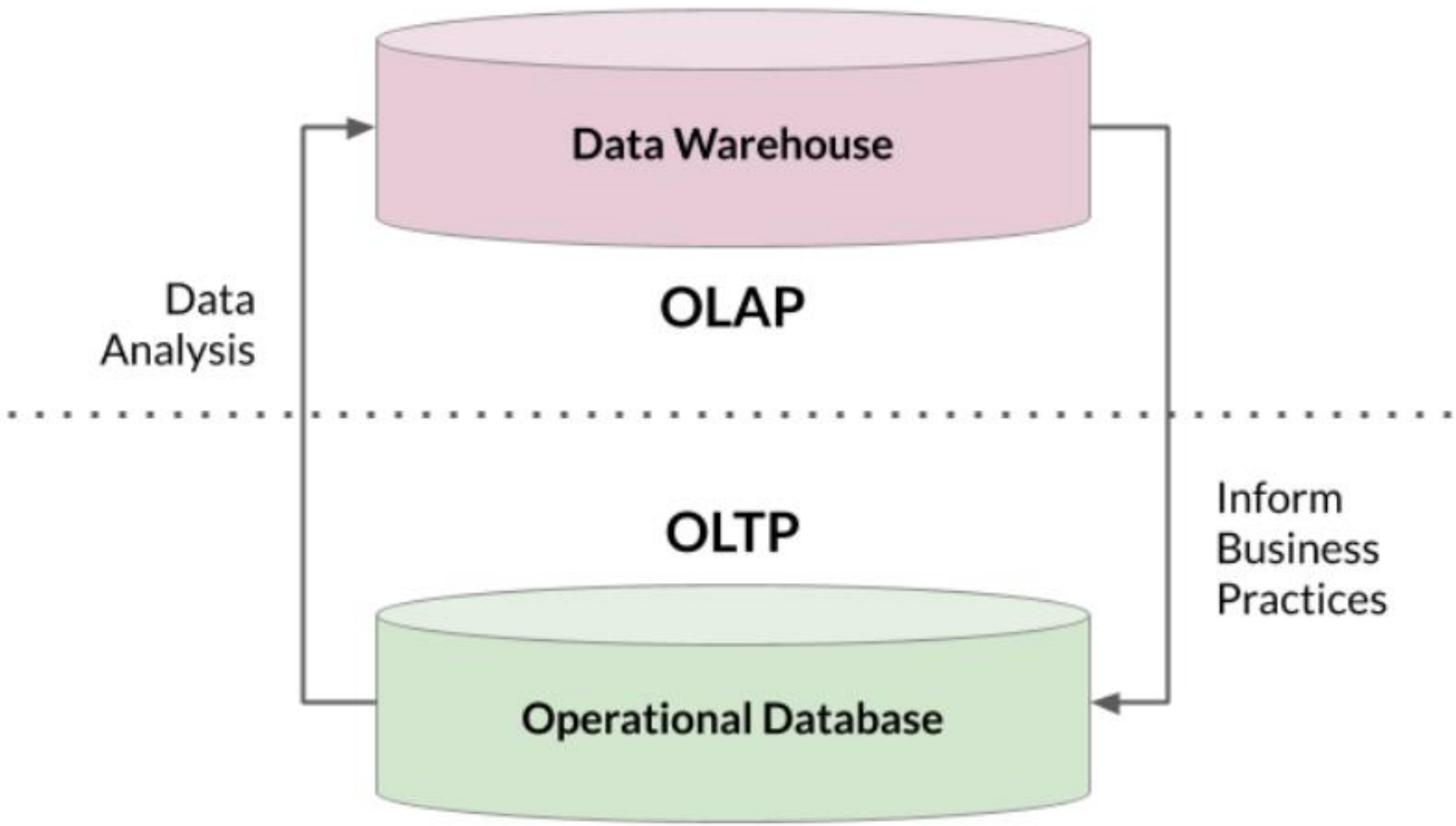
Some concrete examples

OLTP tasks

- Find the price of a book
- Update latest customer transaction
- Keep track of employee hours

OLAP tasks

- Calculate books with best profit margin
- Find most loyal customers
- Decide employee of the month


Data Warehouse
Data
Analysis
OLAP
OLTP
Inform
Business
Practices
Operational Database

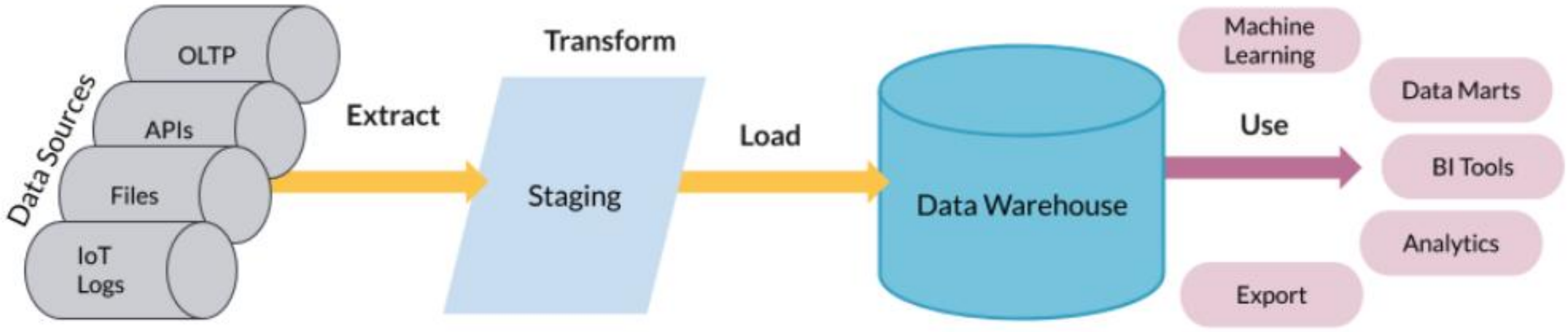
Storing data beyond traditional databases

- **Traditional databases**
  - For storing real-time relational structured data ? **OLTP**
- **Data warehouses**
  - For analyzing archived structured data ? **OLAP**
- **Data lakes**
  - For storing data of all structures = flexibility and scalability
  - For analyzing **big data**

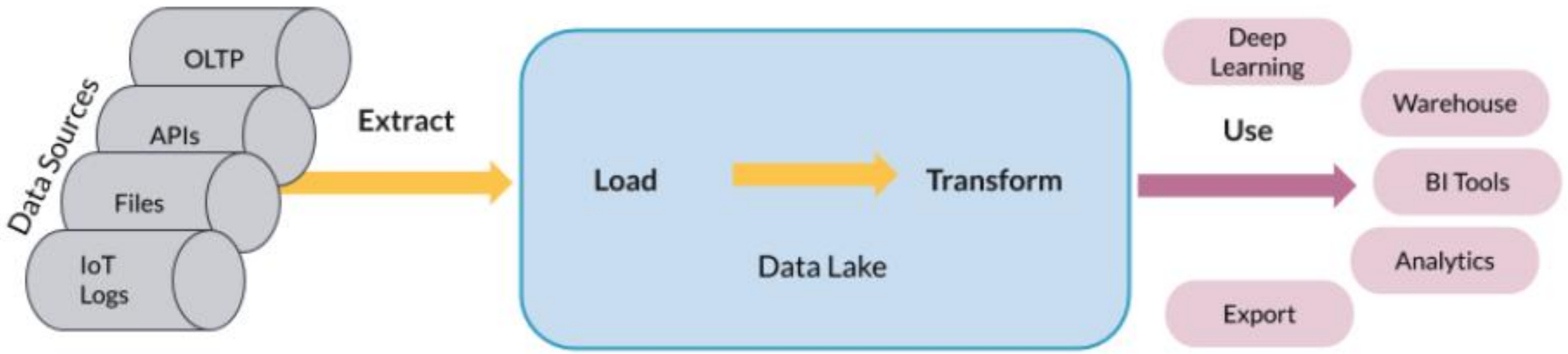
Data warehouses

- Optimized for analytics - **OLAP**
  - Organized for reading/aggregating data
  - Usually read-only
- Contains data from multiple sources
- Massively Parallel Processing (MPP)
- Typically uses a denormalized schema and dimensional modeling

ETL


Data Sources
OLTP
APIs
Files
IoT
Logs
Extract
Transform
Staging
Load
Data Warehouse
Use
Machine Learning
Data Marts
BI Tools
Analytics
Export

ELT

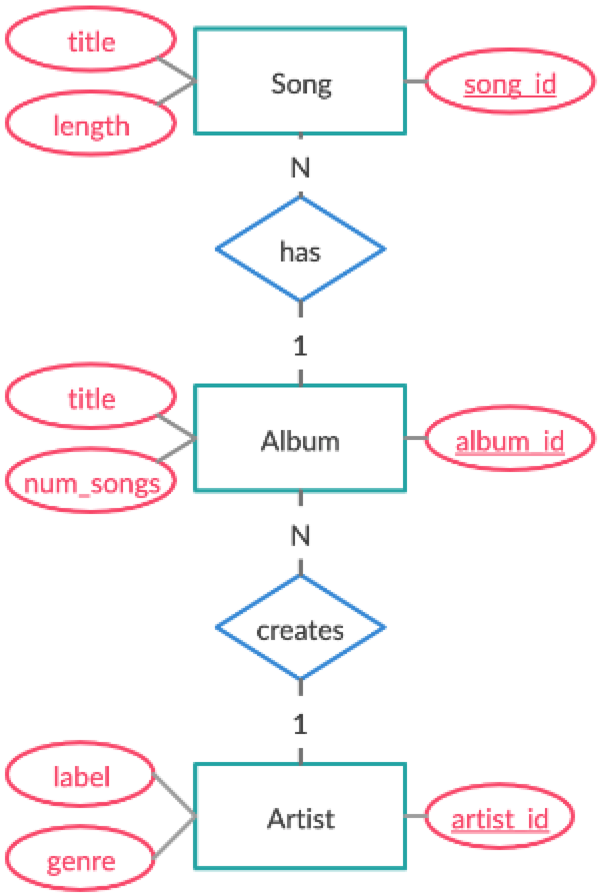

Data Sources
OLTP
APIs
Files
IoT
Logs
Extract
Load
Transform
Data Lake
Use
Deep Learning
Warehouse
BI Tools
Analytics
Export

Data modeling

Process of creating a *data model* for the data to be stored

1. **Conceptual data model**: describes entities, relationships, and attributes
   - *Tools:* data structure diagrams, e.g., entity-relational diagrams and UML diagrams
2. **Logical data model**: defines tables, columns, relationships
   - *Tools:* database models and schemas, e.g., relational model and star schema
3. **Physical data model**: describes physical storage
   - *Tools*: partitions, CPUs, indexes, backup systems and tablespaces

Conceptual - ER diagram



Logical - schema

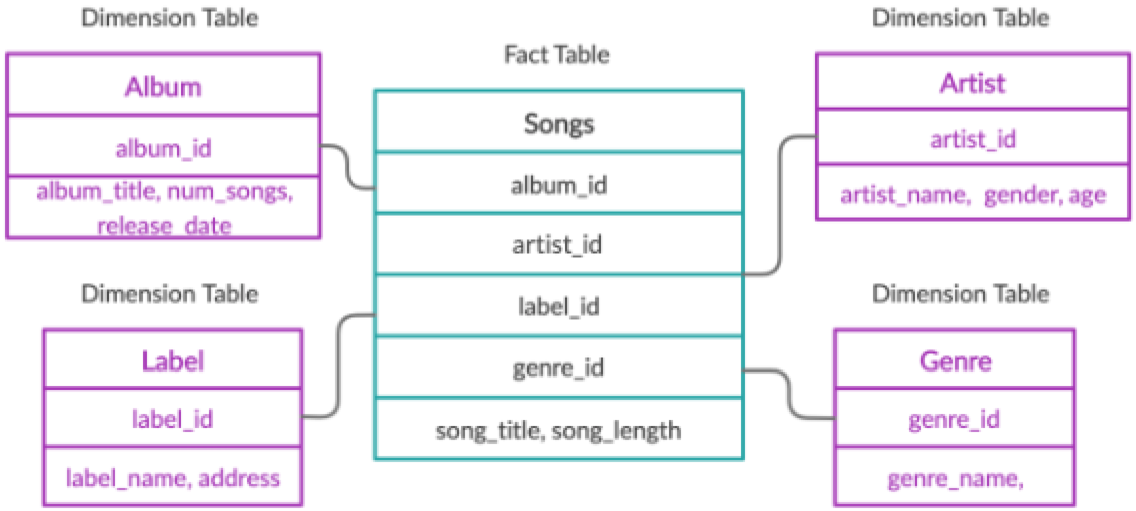
Beyond the relational model

Dimensional modeling

Adaptation of the relational model for data warehouse design

- Optimized for **OLAP** queries: aggregate data, not updating (OLTP)
- Built using the star schema
- Easy to interpret and extend schema

Elements of dimensional modeling



Fact tables

- Decided by business use-case
- Holds records of a metric
- Changes regularly
- Connects to dimensions via foreign keys

Dimension tables

- Holds descriptions of attributes
- Does not change as often

Organize by:

- What is being analyzed?
- How often do entities change?

Star schema

Dimensional modeling: star schema

Fact tables

- Holds records of a metric
- Changes regularly
- Connects to dimensions via foreign keys

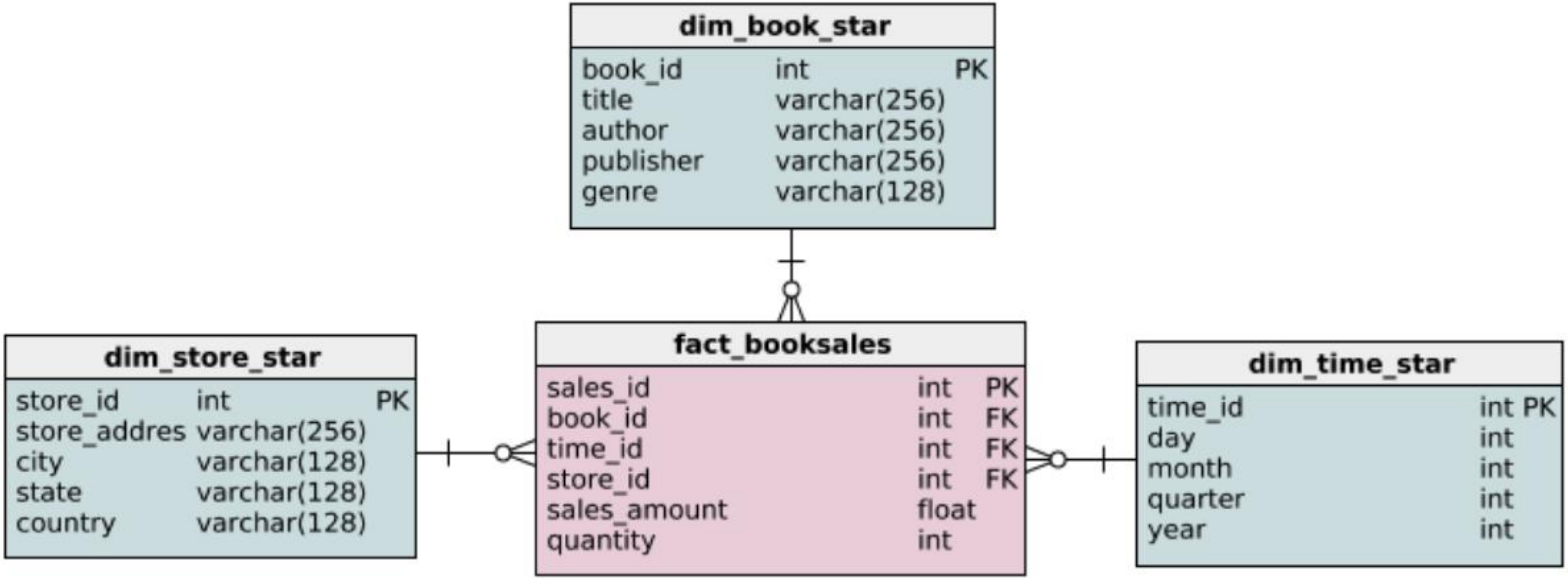
Dimension tables

- Holds descriptions of attributes
- Does not change as often

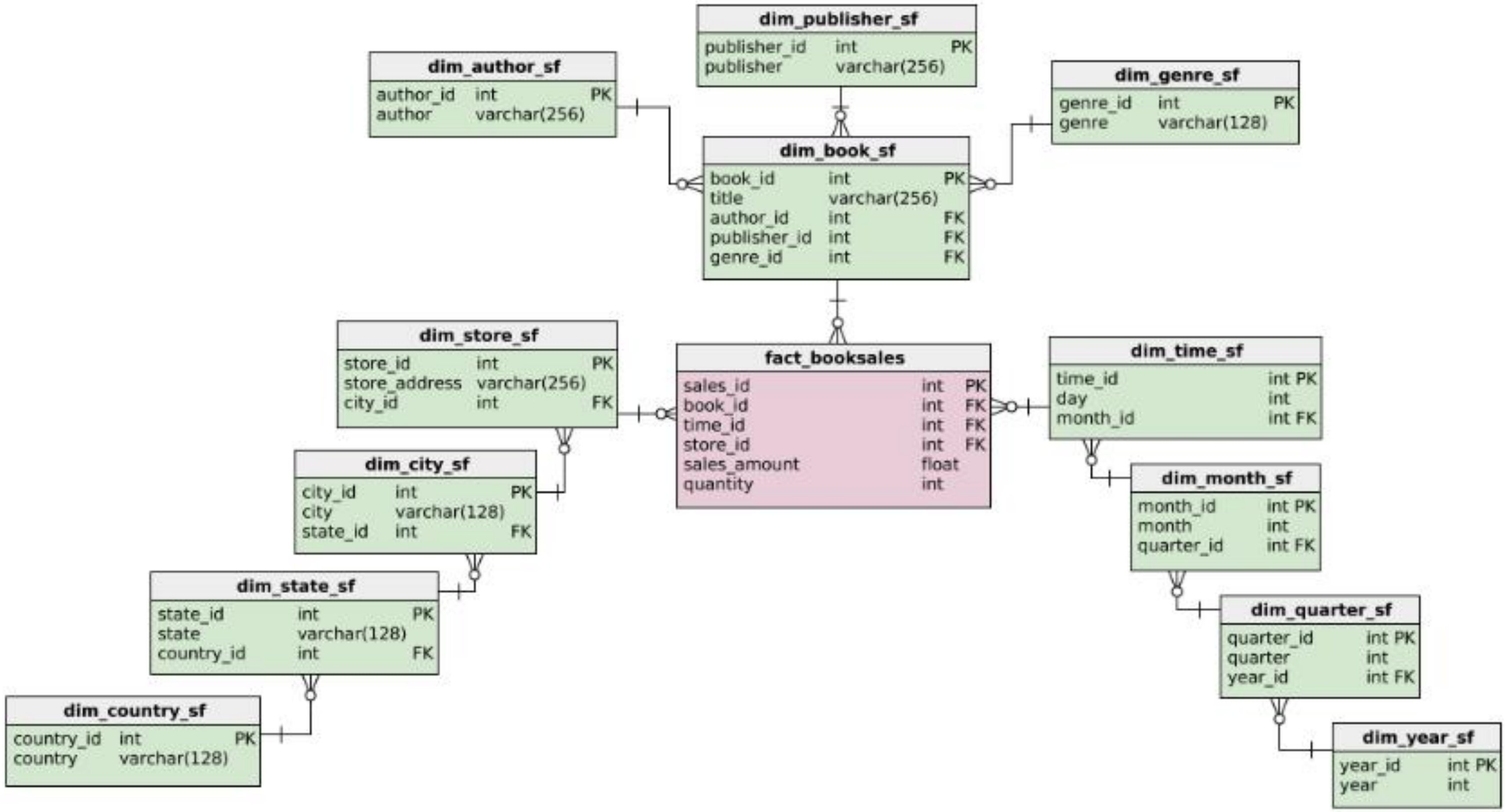
Example:

- Supply books to stores in USA and Canada
- Keep track of book sales

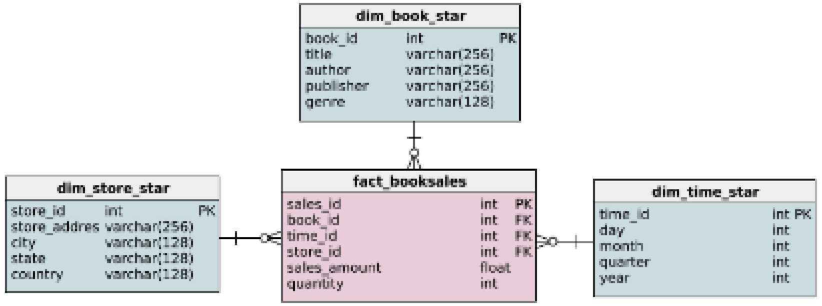
Star schema example



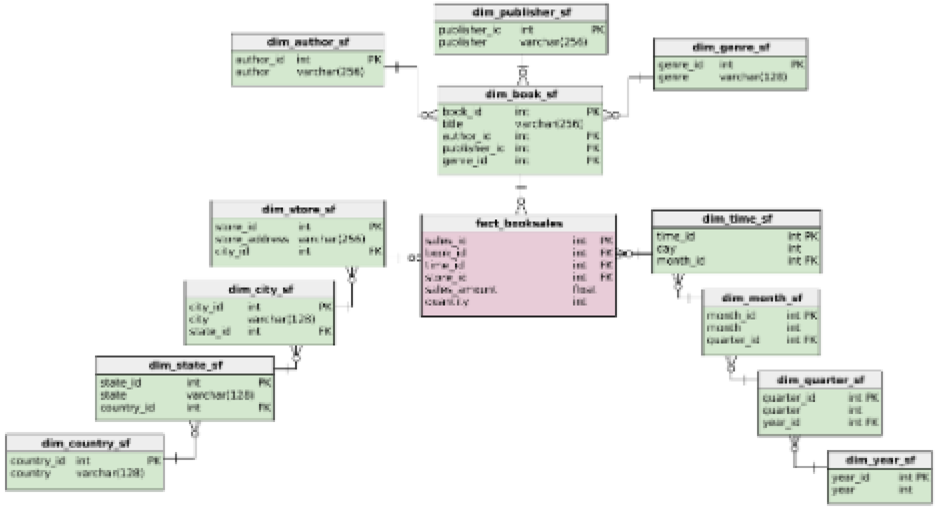
Snowflake schema (an extension)



Same fact table, different dimensions



Star schemas: one dimension



Snowflake schemas: more than one dimension

Because dimension tables are ***normalized***

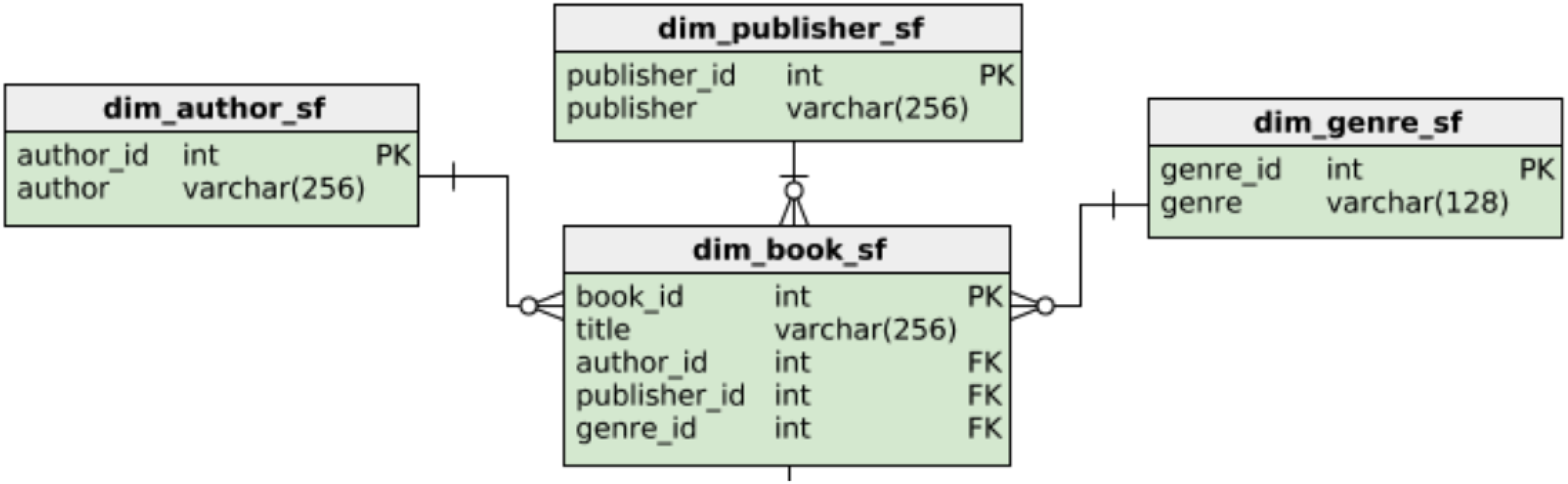
Book dimension of the star schema

dim_book_star		
book_id	int	PK
title	varchar(256)	
author	varchar(256)	
publisher	varchar(256)	
genre	varchar(128)	

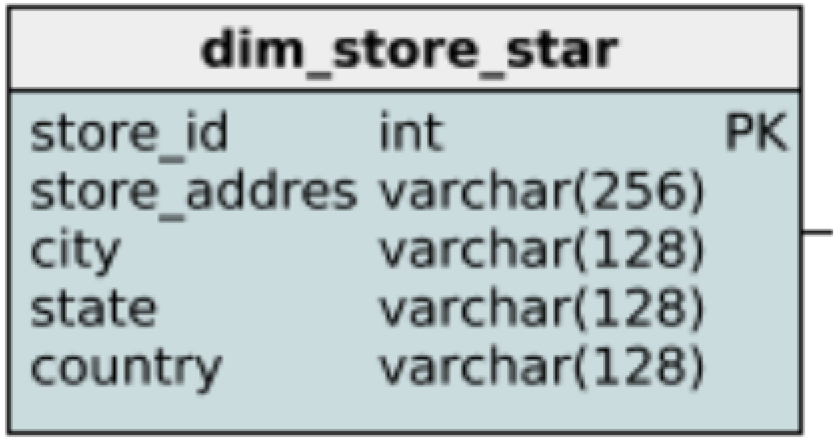
Most likely to have repeating values:

- Author
- Publisher
- Genre

Book dimension of the snowflake schema

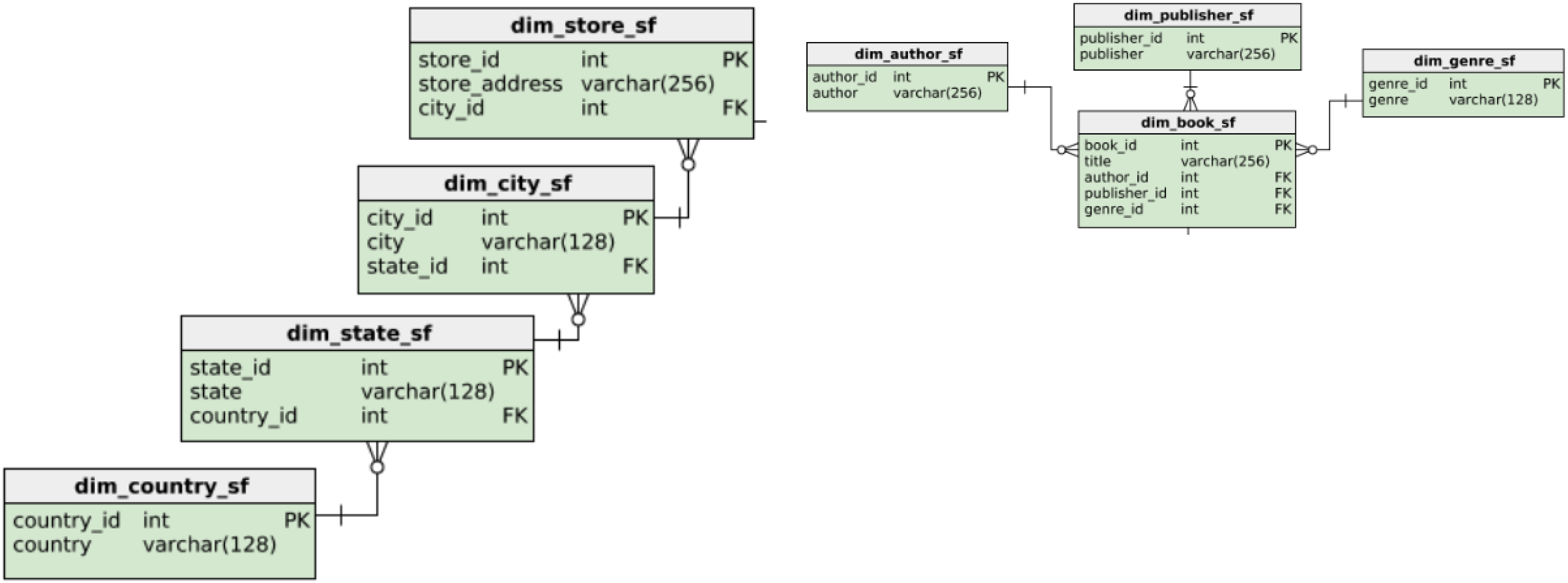


Store dimension of the star schema



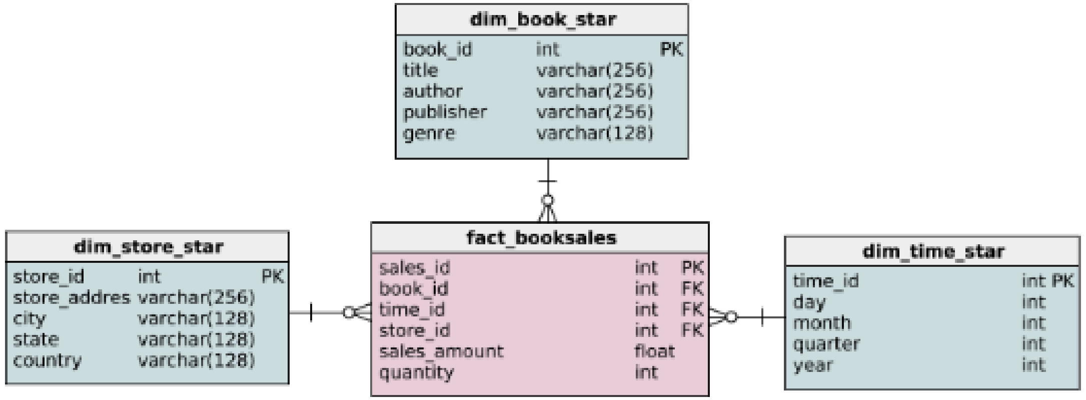
- City
- State
- Country

Store dimension of the snowflake schema

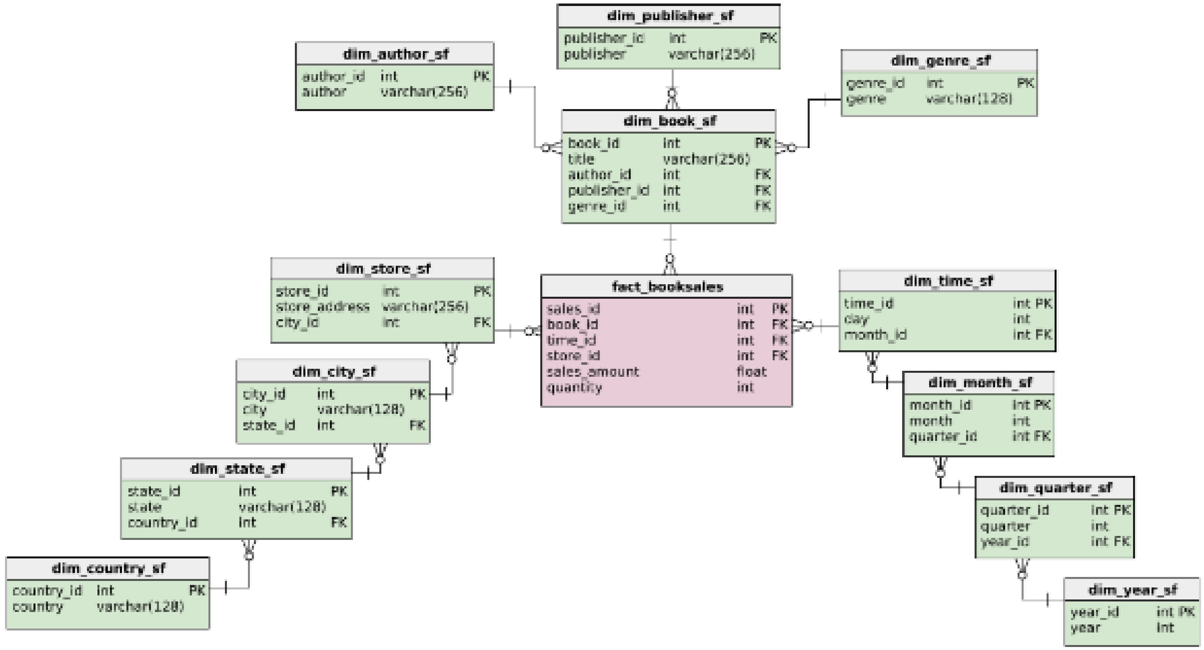


Back to our book store example

Denormalized: star schema



Normalized: snowflake schema




SKELLYSCOO
I have several questions.

Feed us back!