A Data Warehousing Primer

Roland Bouman (Strukton Rail)
http://rpbouman.blogspot.com/

Starring Sakila

Topics

Starring Sakila

- Terminology
  - Business Intelligence
  - Data Warehouse
  - Dimensional Model
  - Star Schema
  - OLAP
  - Cube

Data Warehousing Terminology

What is Business Intelligence?

- Business Intelligence (BI)
  - Skills, technologies, applications and practices to acquire a better understanding of the commercial context of your business.
- Data Warehouse
- Dimensional Model
- Star Schema
- OLAP
- Cube

- Business Intelligence
- Data Warehouse
  - A database designed to support Business Intelligence
- Dimensional Model
- Star Schema
- OLAP
- Cube

What is a Data Warehouse?

- Business Intelligence
- Data Warehouse
- Dimensional Model
  - A logical data model that divides data in two kinds: Facts and Dimensions
- Star Schema
- OLAP
- Cube

What is the Dimensional Model?

- Business Intelligence
- Data Warehouse
- Dimensional Model
- Star Schema
  - Physical implementation of the Dimensional Model on a RDBMS which maps a dimension to a single table
- OLAP
- Cube

What is a Star Schema?

- Business Intelligence
- Data Warehouse
- Dimensional Model
- Star Schema
- OLAP
  - On-Line Analytical Processing: querying muli-dimensional data, cornerstone of most BI applications
- Cube

What is OLAP?

- Business Intelligence
- Data Warehouse
- Dimensional Model
- Star Schema
- OLAP
- Cube
  - Multi-dimensional data structure suitable for OLAP queries

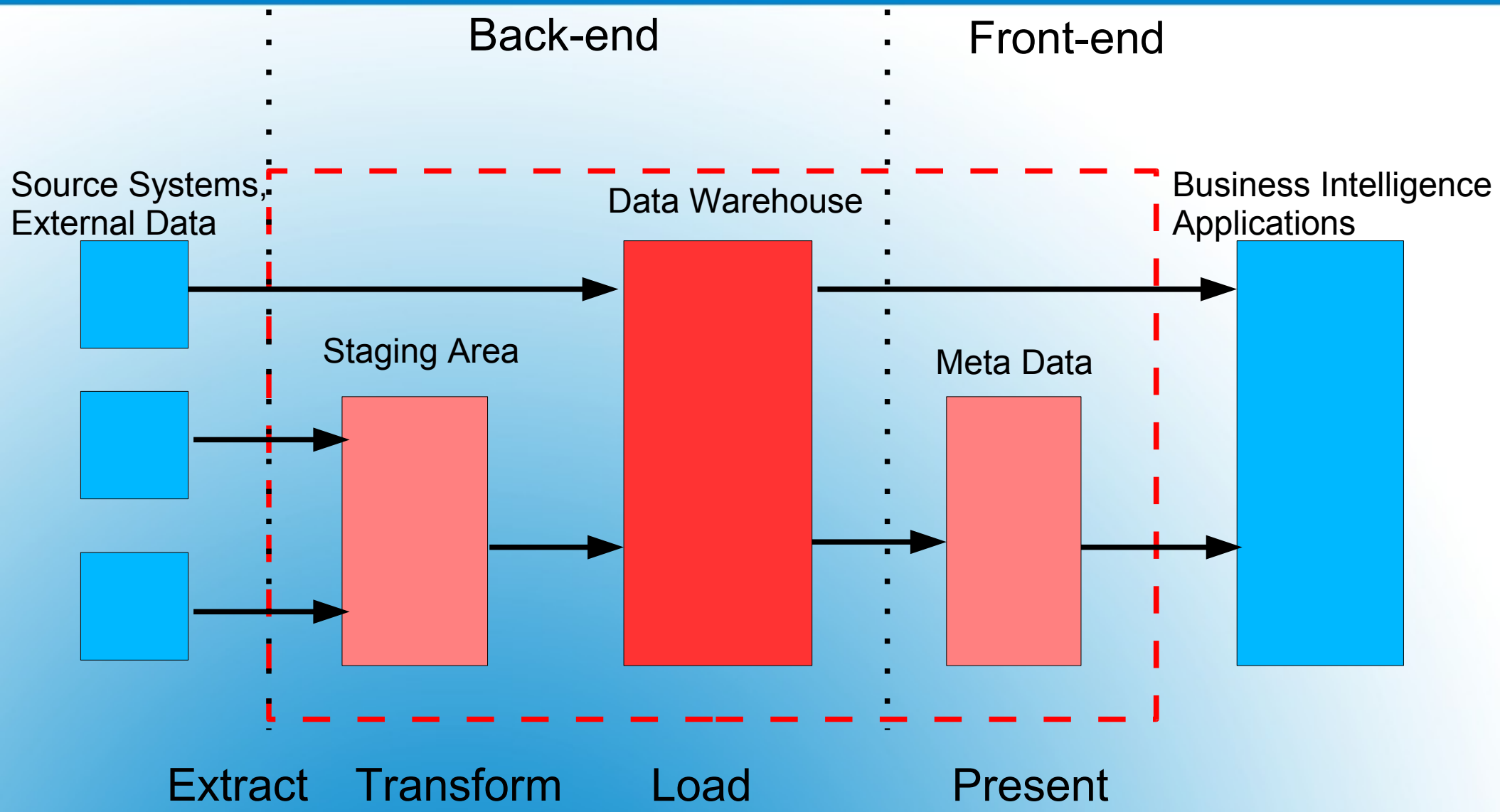
What is a Cube

Business Intelligence

Understanding your Business

- Front end Applications:
  - Reports
  - Charts and Graphs
  - OLAP Pivot tables
  - Data Mining
  - Dashboards
- Back end, Infrastructure
  - ETL
    - Extract
    - Transformation
    - Load
  - Data Warehouse
  - Data Mart
  - Metadata
  - ROLAP Cube

Business Intelligence



High Level BI Architecture

Data Warehouse

Business Intelligence Database

- Ultimately, it's just a Relational Database
  - Tables, Columns, Keys...
- ...But designed for BI applications
  - Ease of use
  - Performance
- Data from various source systems
  - Integration, Standardization, Data cleaning
  - Add and maintain history

Data Warehouse

OLTP vs OLAP: Application Characterization

- OLTP
  - Operational
  - 'Always' on
  - All kinds of users
  - Many users
  - Directly supports business process
  - Keep a Record of Current status
- OLAP
  - Tactical, Strategic
  - Periodically Available
  - Managers, Directors
  - Few(er) users
  - Decision support, long-term planning
  - Maintain history

- OLTP
  - Subject Oriented
  - Add, Modify, Remove single rows
  - Human data entry
  - Queries for small sets of rows with all their details
  - Standard queries
- OLAP
  - Aspect Oriented
  - Bulk load, rarely modify, never remove
  - Automated ETL jobs
  - Scan large sets to return aggregates over arbitrary groups
  - Ad-hoc queries

OLTP vs OLAP: data processing

OLTP vs OLAP: database schema organization

- OLTP
  - Entity-Relationship model
  - Entities, Attributes, Relationships
  - Foreign key constraints
  - Indexes to increase performance
  - Normalized to 3NF or BCNF
- OLAP
  - Dimensional model
  - Facts, Dimensions, Hierarchies
  - Ref. integrity ensured in loading process
  - Scans on Fact table obliterates indexes
  - Denormalized Dimensions (<= 1NF)

Dimensional Model

Organizing data to suit Business Intelligence

- Two kinds of data
  - Facts
  - Dimensions

The Dimensional Model

The Dimensional Model: Facts

- Facts
  - Measures/Metrics of a Business Process
  - Typical Metrics
    - Cost, Units Sold, Profit

The Dimensional Model: Dimensions

- Dimensions
  - Describe aspects of Business Process
  - Dimensions typically not inter-dependent
  - Who? What? Where? When? Why?
  - Typical Dimensions:
    - Customer (who?), Product (what?), Date/Time (when?)

- Dimension Attributes organized in Hierarchies
  - Date dimension examples:
    - Year, Quarter, Month, Day
    - Year, Week, Day
- Metrics typically numeric and additive
- Navigate fact data
  - Choose particular values for dimension
  - Aggregate fact data at chosen level of hierarchy

The Dimensional Model: Navigating Facts with Dimensions

Date Dimension →		2008 Q4			
↓ Location Dimension		All Months	October	November	December
All locations		$ 3850	$ 1000	$ 1350	$ 1500
America	All America	$ 2050	$500	$ 750	$ 800
	North	$ 1275	$ 300	$ 500	$ 475
	South	$ 775	$ 200	$ 250	$ 325
Europe	All Europe	$ 1800	$ 500	$ 600	$ 700
	East	$ 800	$ 250	$ 250	$ 300
	West	$ 1000	$ 250	$ 350	$ 400

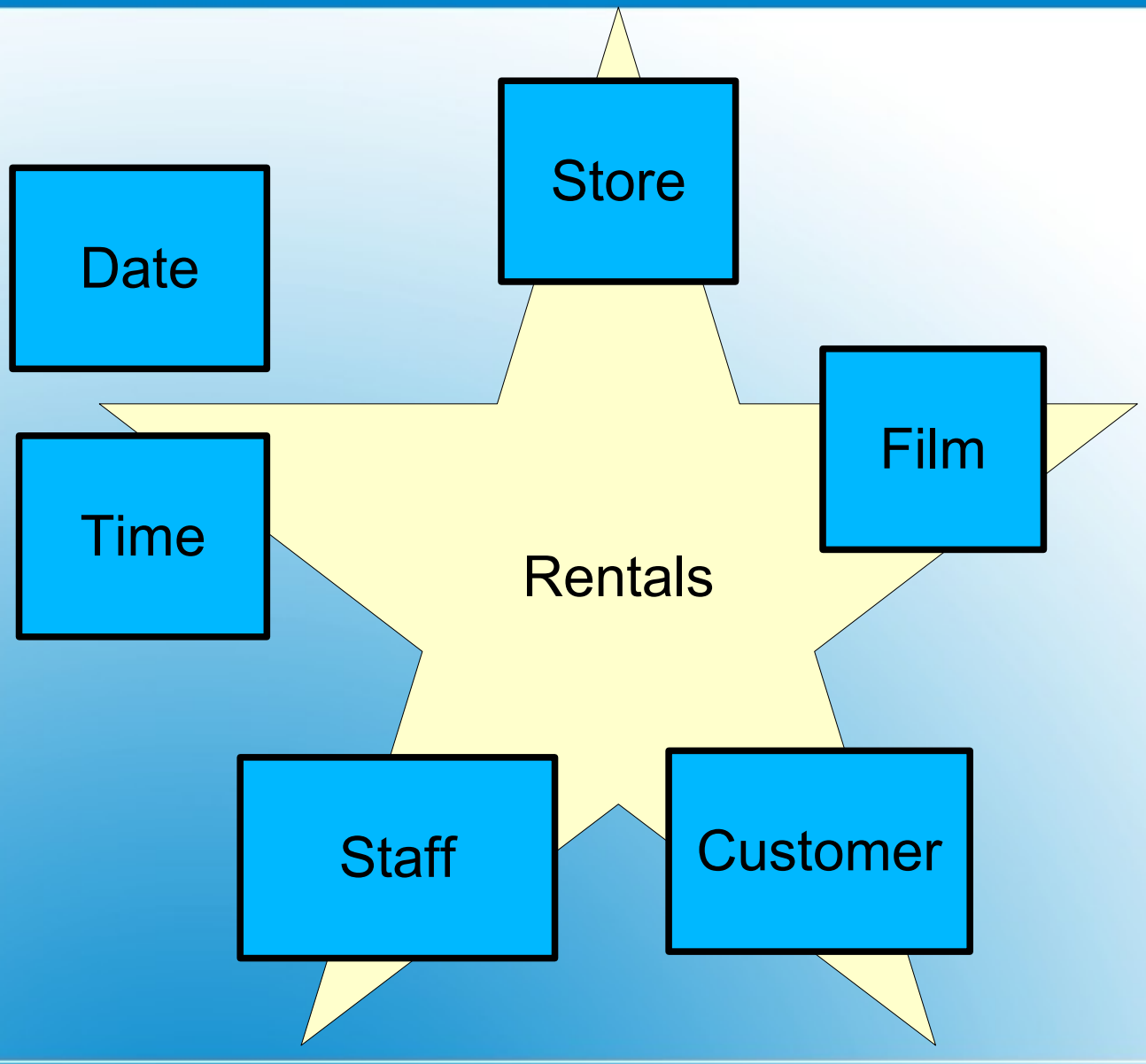
Dimensional Example: Crosstab

Star Schema

Dimensional Model Implementation

Stars Schema Characteristics

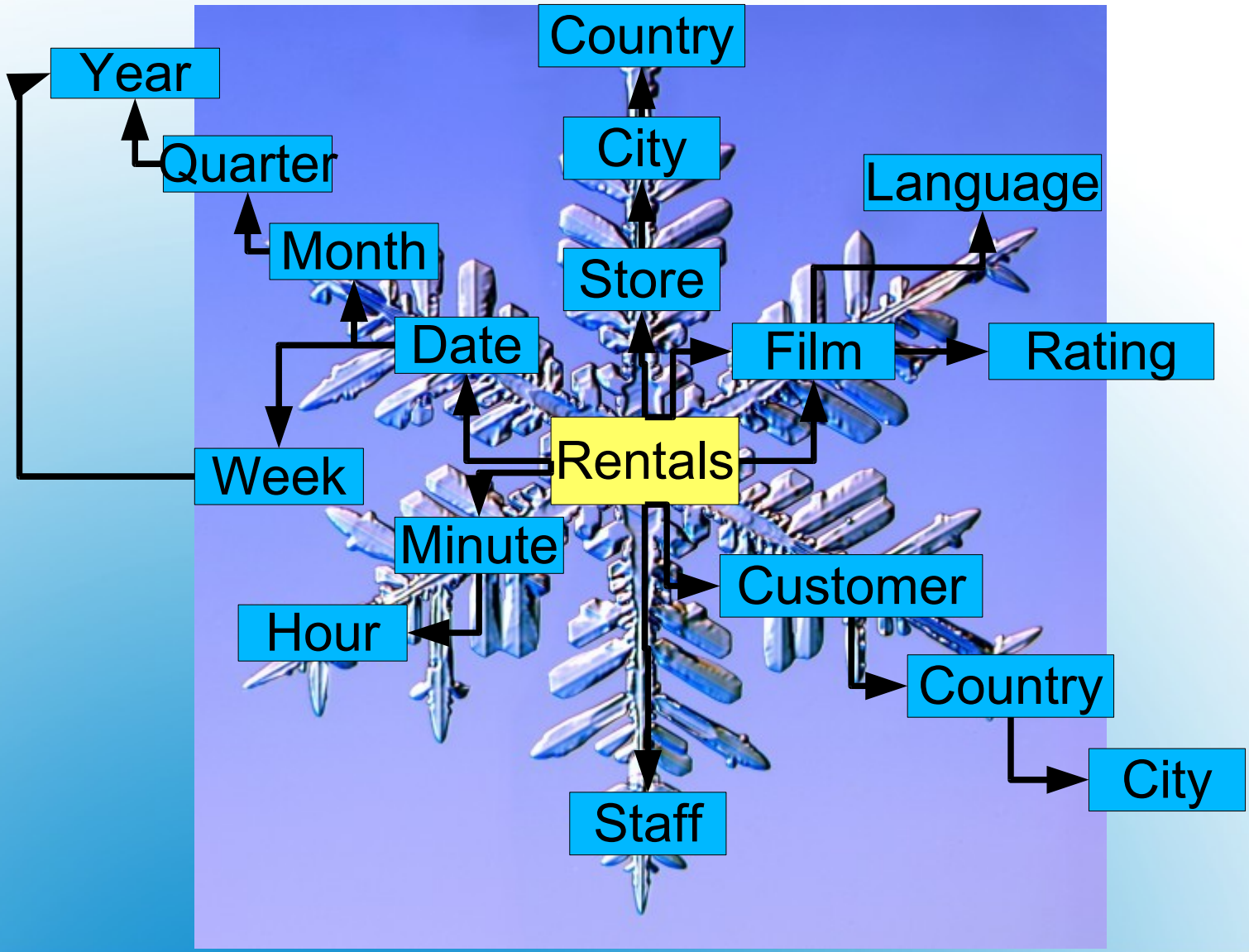
- Central Fact Table
  - Columns for storing Metrics
  - 'Foreign Key' columns point to Dimension
  - Typically normalized and not pre-aggregated
- Dimension maps to a Dimension table
  - Surrogate key
  - Descriptive attributes organized in hierarchies
  - No Foreign Keys to other tables
  - Typically heavily denormalized



Star Schema example: Sakila Rentals

Stars Schema Characteristics

- Star schema is 'just' an implementation
  - Optimized for simplicity
  - Optimized for performance (?)
  - Heavily denormalized dimensions
- Snowflake: Star Schema Alternative
  - Still a dimensional model
  - Still a central fact table
  - Normalized dimensions
  - Easier maintenance of dimensions



Snow Flake example: Sakila Rentals

Desinging Star Schemas

Starring Sakila

- Select Business Process
  - Sales, Purchase, Storage, Transport, ...
- Define Facts and Key Metrics
  - Facts: Key Event in Business Process
  - Metrics (Fact Attributes): Count or Amount
- Choose Dimensions and Hierarchies
  - What? When? Where?
  - Who? Why?

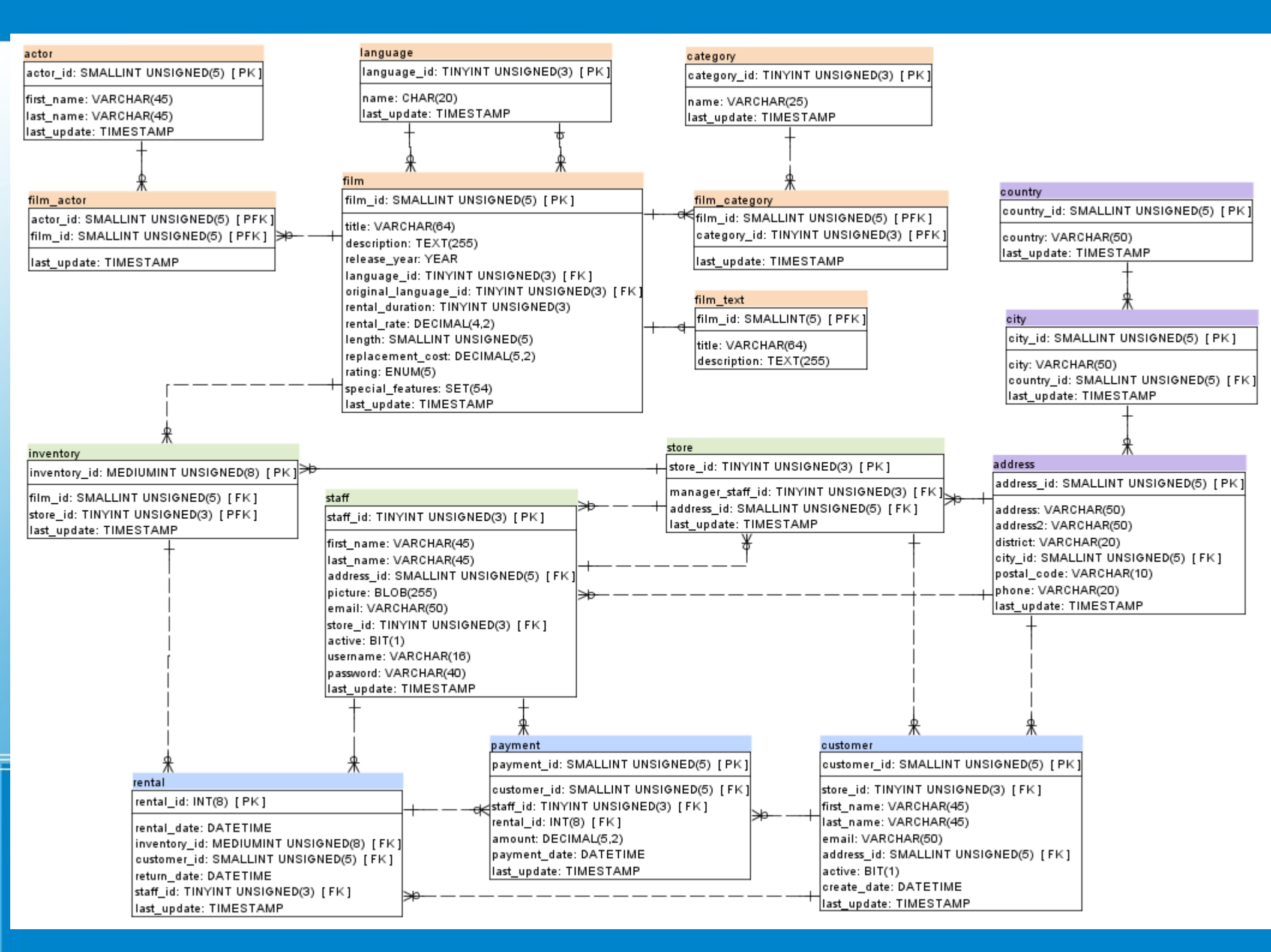
Dimensional Design

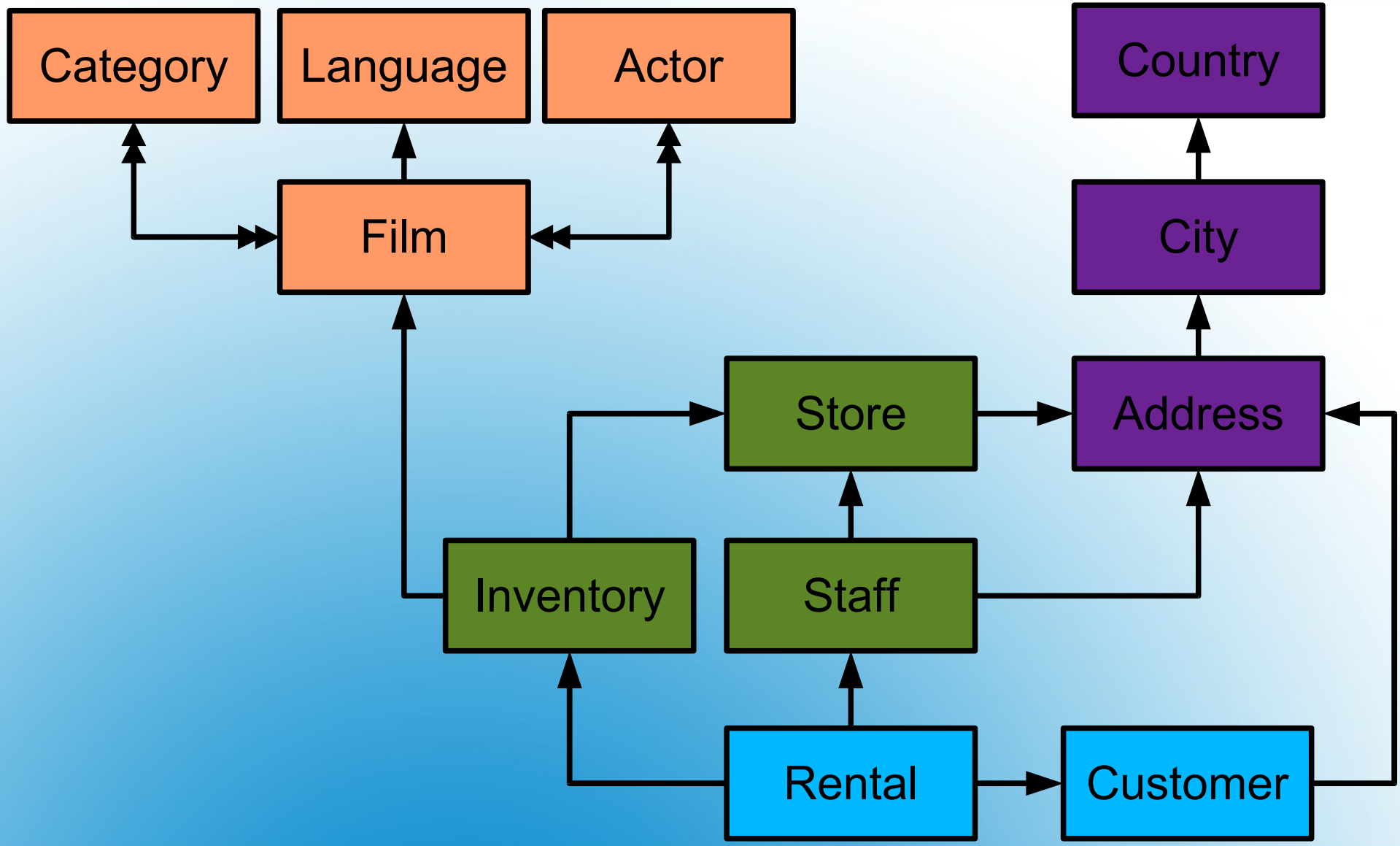
Dimensional Model example

- MySQL Sample Database
  - http://dev.mysql.com/doc/sakila/en/sakila.html
- DVD rental business
  - Overly simplified database schema
- Typical OLTP database

- Rental Business Process
  - Customer visits store, picks DVD
  - DVD taken out of store inventory by staff member
  - Customer returns home and enjoys DVD
  - Customer returns to store with DVD
  - DVD returned to staff member
  - Staff member collects payment made by customer

Dimensional Model example

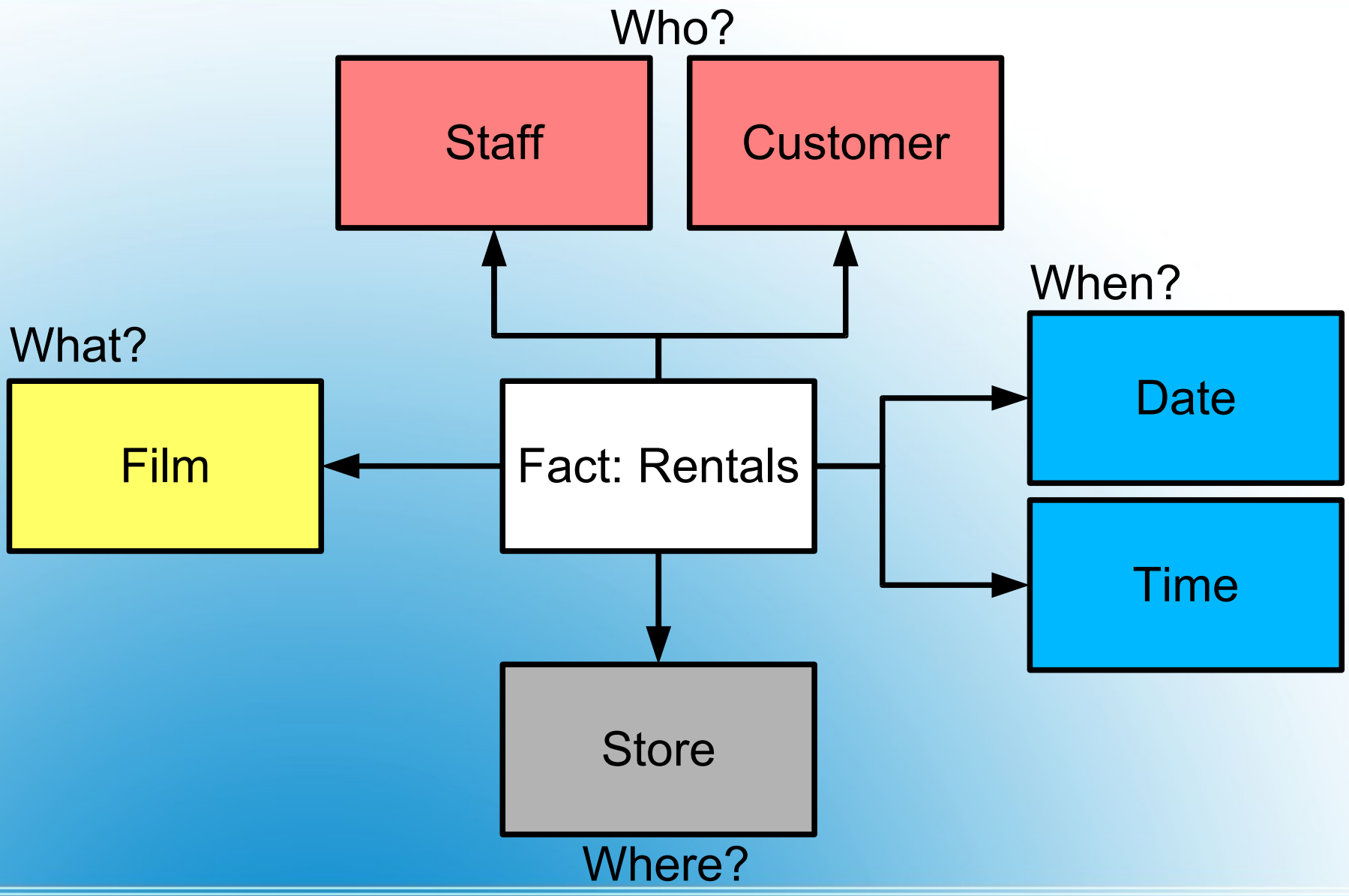

actor
actor_id: SMALLINT UNSIGNED(5) [PK]
first_name: VARCHAR(45)
last_name: VARCHAR(45)
last_update: TIMESTAMP
language
language_id: TINYINT UNSIGNED(3) [PK]
name: CHAR(20)
last_update: TIMESTAMP
category
category_id: TINYINT UNSIGNED(3) [PK]
name: VARCHAR(25)
last_update: TIMESTAMP
film_actor
actor_id: SMALLINT UNSIGNED(5) [PFK]
film_id: SMALLINT UNSIGNED(5) [PFK]
last_update: TIMESTAMP
film
film_id: SMALLINT UNSIGNED(5) [PK]
title: VARCHAR(64)
description: TEXT(255)
release_year: YEAR
language_id: TINYINT UNSIGNED(3) [FK]
original_language_id: TINYINT UNSIGNED(3) [FK]
rental_duration: TINYINT UNSIGNED(3)
rental_rate: DECIMAL(4,2)
length: SMALLINT UNSIGNED(5)
replacement_cost: DECIMAL(5,2)
rating: ENUM(5)
special_features: SET(54)
last_update: TIMESTAMP
film_category
film_id: SMALLINT UNSIGNED(5) [PFK]
category_id: TINYINT UNSIGNED(3) [PFK]
last_update: TIMESTAMP
country
country_id: SMALLINT UNSIGNED(5) [PK]
country: VARCHAR(50)
last_update: TIMESTAMP
film_text
film_id: SMALLINT(5) [PFK]
title: VARCHAR(64)
description: TEXT(255)
city
city_id: SMALLINT UNSIGNED(5) [PK]
city: VARCHAR(50)
country_id: SMALLINT UNSIGNED(5) [FK]
last_update: TIMESTAMP
inventory
inventory_id: MEDIUMINT UNSIGNED(8) [PK]
film_id: SMALLINT UNSIGNED(5) [FK]
store_id: TINYINT UNSIGNED(3) [PFK]
last_update: TIMESTAMP
store
store_id: TINYINT UNSIGNED(3) [PK]
manager_staff_id: TINYINT UNSIGNED(3) [FK]
address_id: SMALLINT UNSIGNED(5) [FK]
last_update: TIMESTAMP
address
address_id: SMALLINT UNSIGNED(5) [PK]
address: VARCHAR(50)
address2: VARCHAR(50)
district: VARCHAR(20)
city_id: SMALLINT UNSIGNED(5) [FK]
postal_code: VARCHAR(10)
phone: VARCHAR(20)
last_update: TIMESTAMP
staff
staff_id: TINYINT UNSIGNED(3) [PK]
first_name: VARCHAR(45)
last_name: VARCHAR(45)
address_id: SMALLINT UNSIGNED(5) [FK]
picture: BLOB(255)
email: VARCHAR(50)
store_id: TINYINT UNSIGNED(3) [FK]
active: BIT(1)
username: VARCHAR(16)
password: VARCHAR(40)
last_update: TIMESTAMP
payment
payment_id: SMALLINT UNSIGNED(5) [PK]
customer_id: SMALLINT UNSIGNED(5) [FK]
staff_id: TINYINT UNSIGNED(3) [FK]
rental_id: INT(8) [FK]
amount: DECIMAL(5,2)
payment_date: DATETIME
last_update: TIMESTAMP
customer
customer_id: SMALLINT UNSIGNED(5) [PK]
store_id: TINYINT UNSIGNED(3) [FK]
first_name: VARCHAR(45)
last_name: VARCHAR(45)
email: VARCHAR(50)
address_id: SMALLINT UNSIGNED(5) [FK]
active: BIT(1)
create_date: DATETIME
last_update: TIMESTAMP
rental
rental_id: INT(8) [PK]
rental_date: DATETIME
inventory_id: MEDIUMINT UNSIGNED(8) [FK]
customer_id: SMALLINT UNSIGNED(5) [FK]
return_date: DATETIME
staff_id: TINYINT UNSIGNED(3) [FK]
last_update: TIMESTAMP



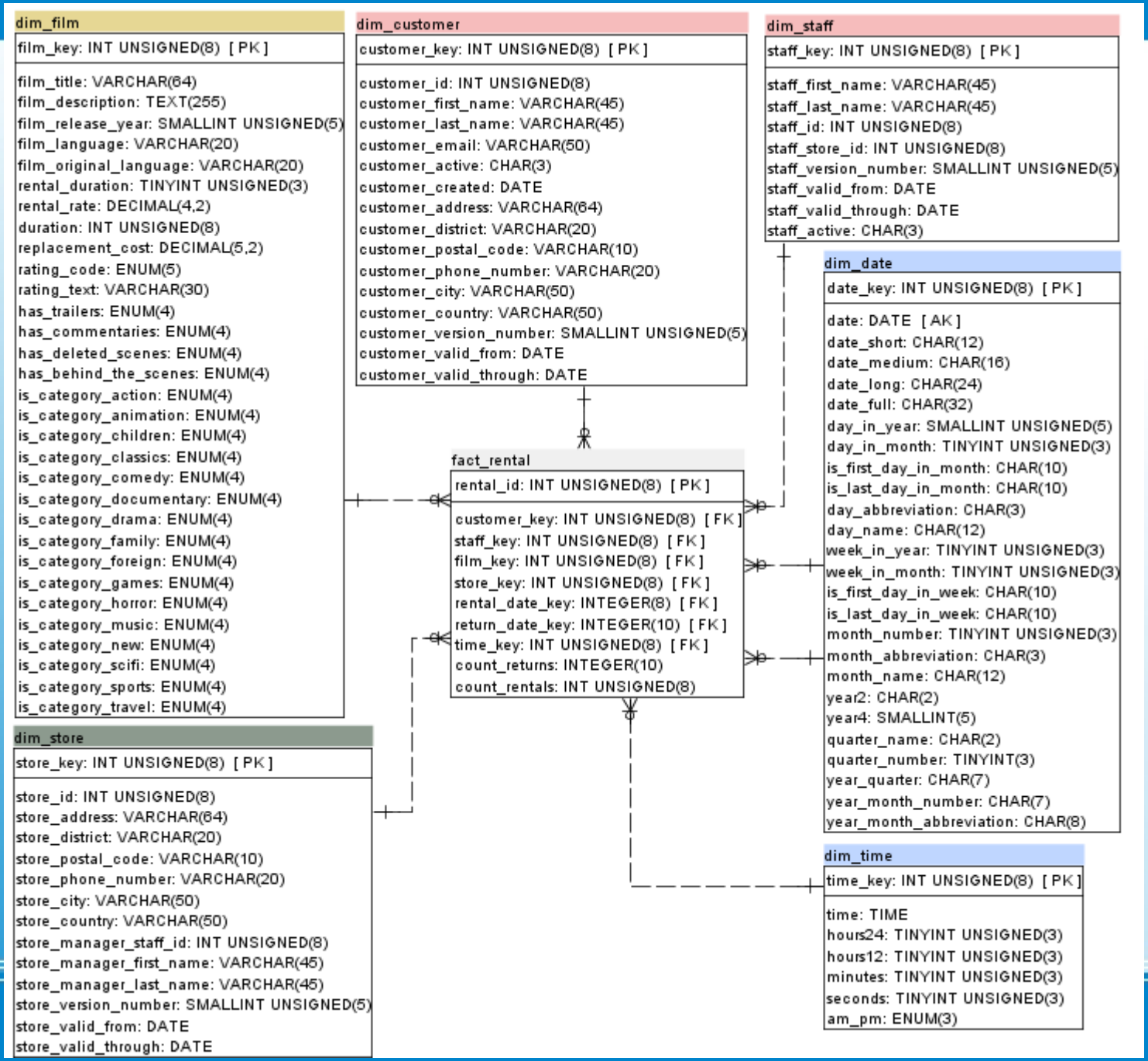
3NF Source schema: Sakila Rentals

Example Business Process: Rentals

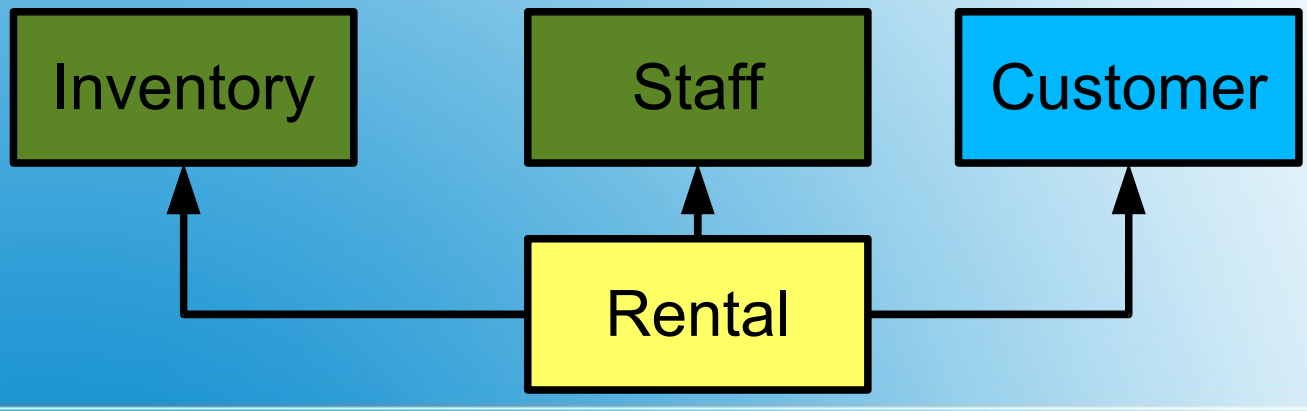
- Select Business Process
  - Rentals
- Identify Facts
  - Count (number of rentals)
  - Rental Duration
- Choose Dimensions
  - What: Films
  - When: Rental, Return
  - Who: Customer, Staff
  - Where: Store



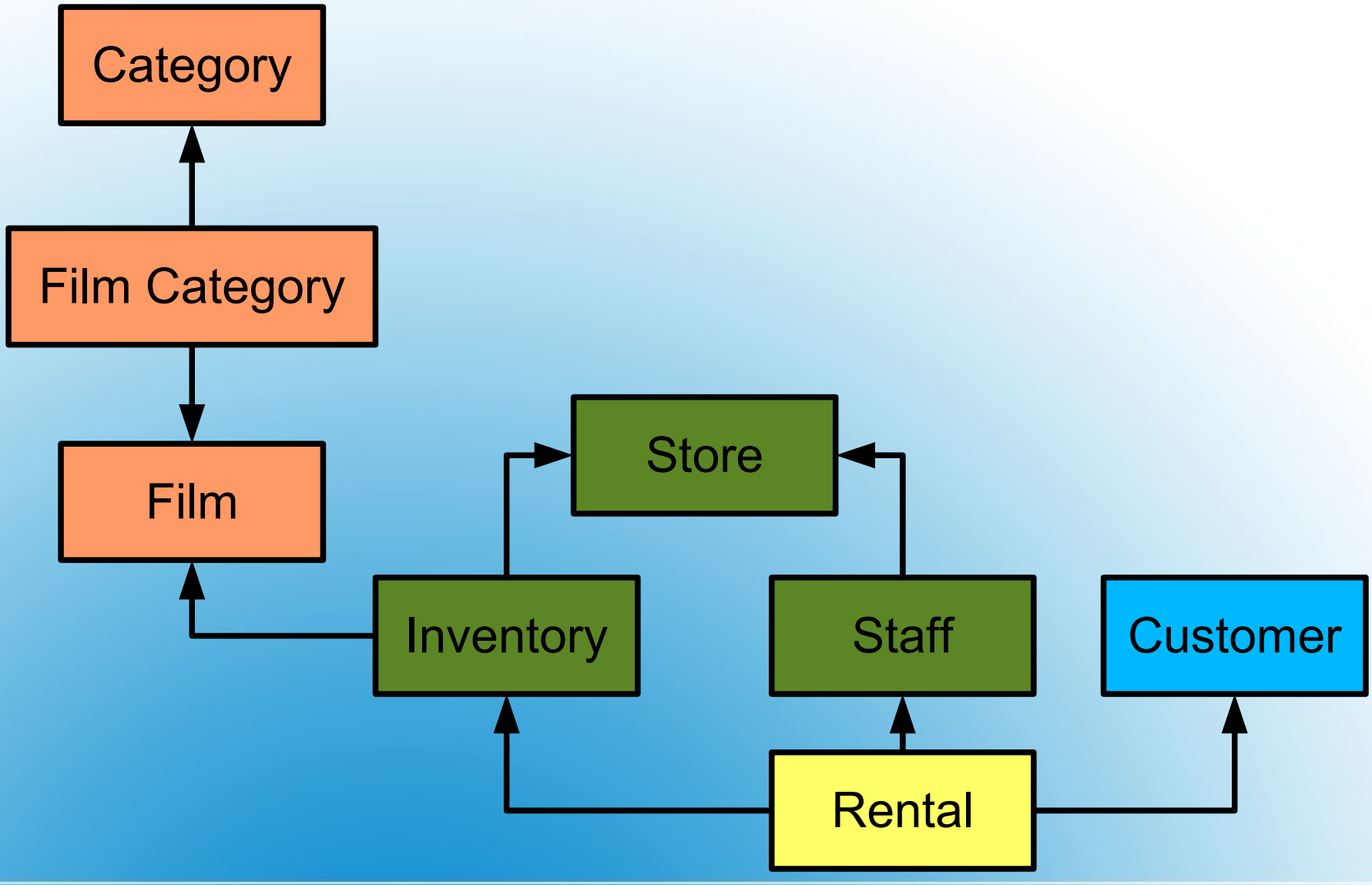
Target Star Schema



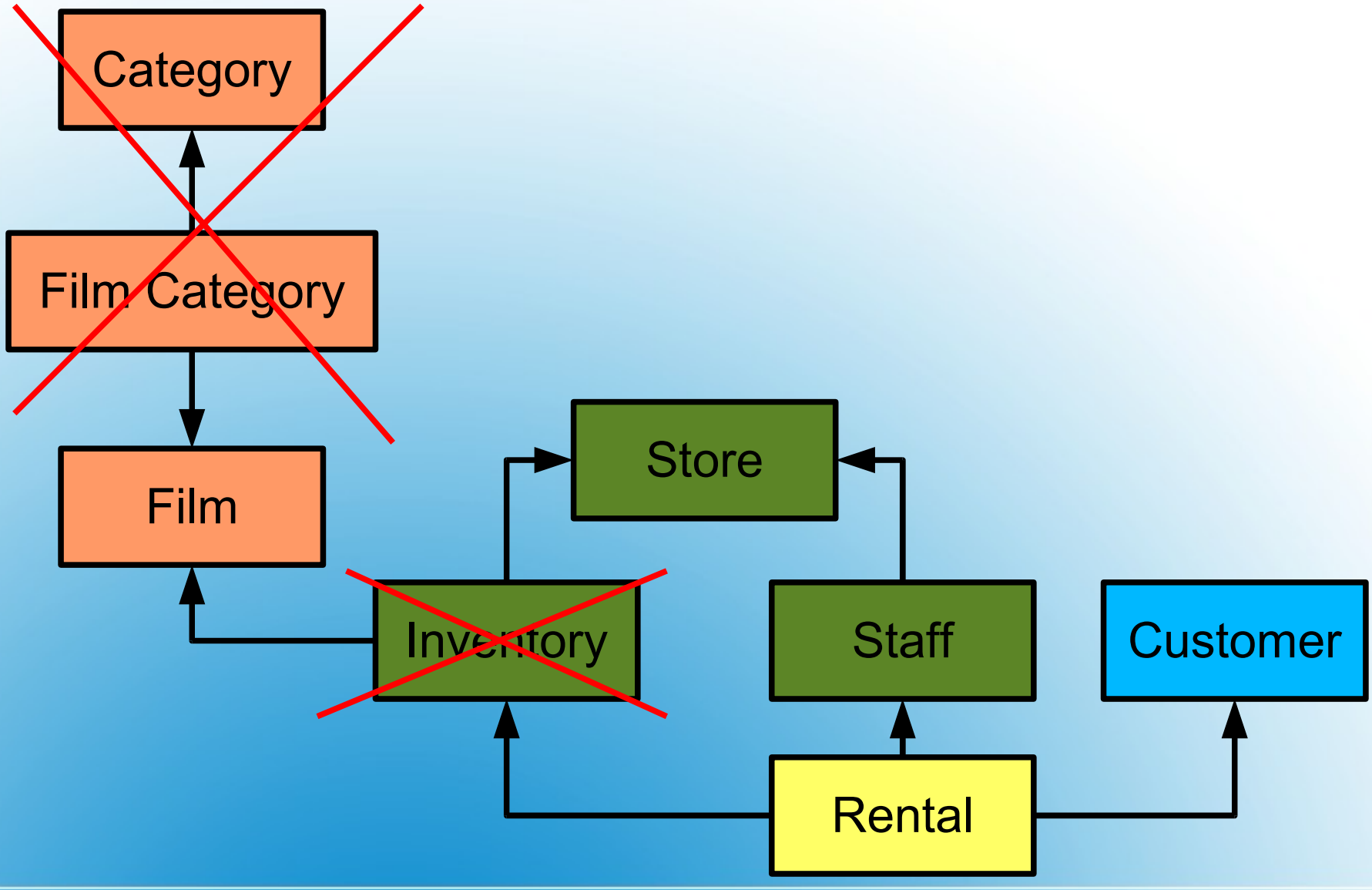
Rental Star Schema



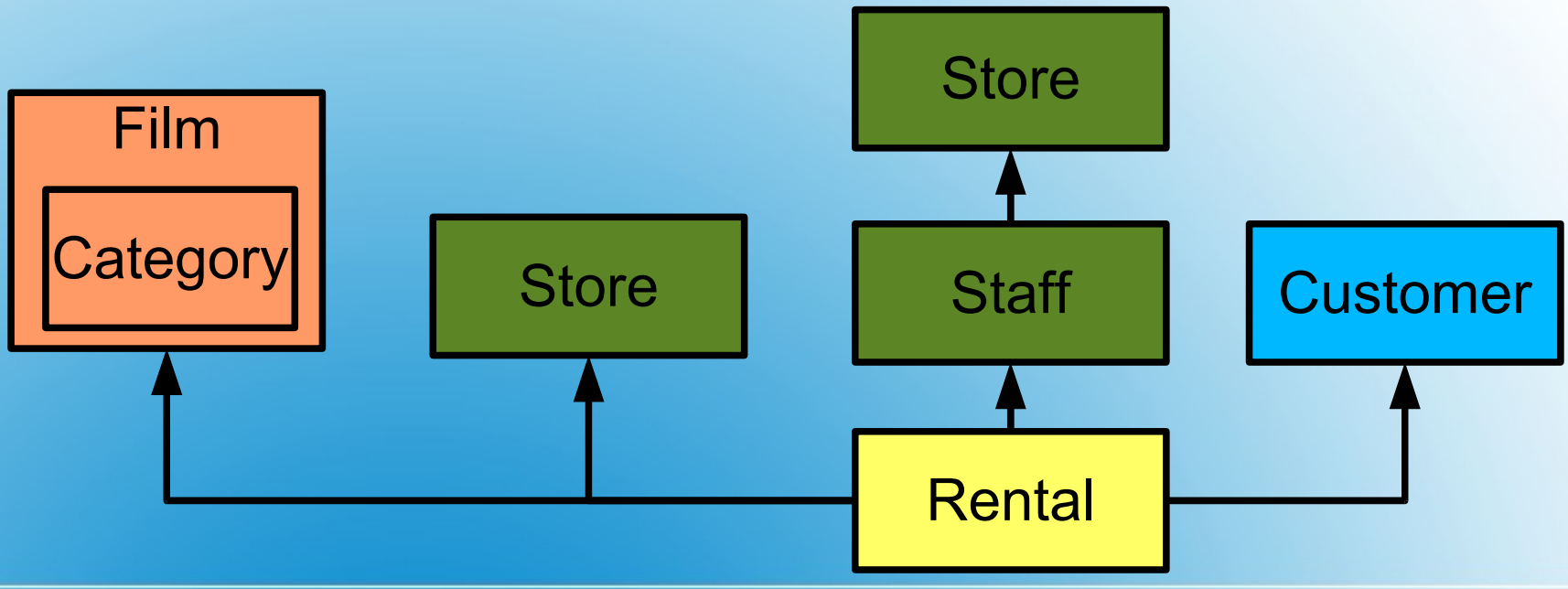
A star is born: Rentals 3NF



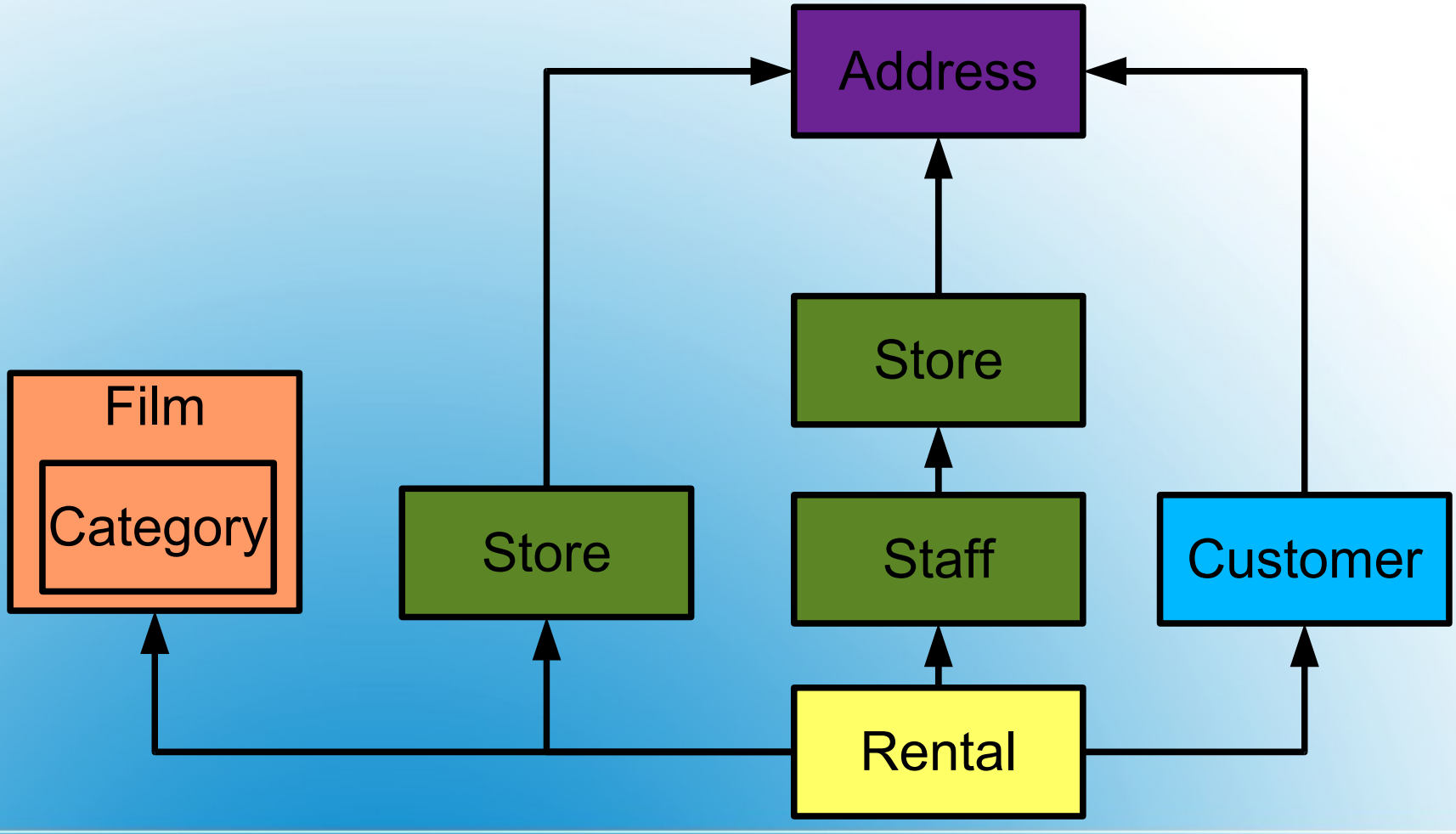
A star is born: Rentals 3NF



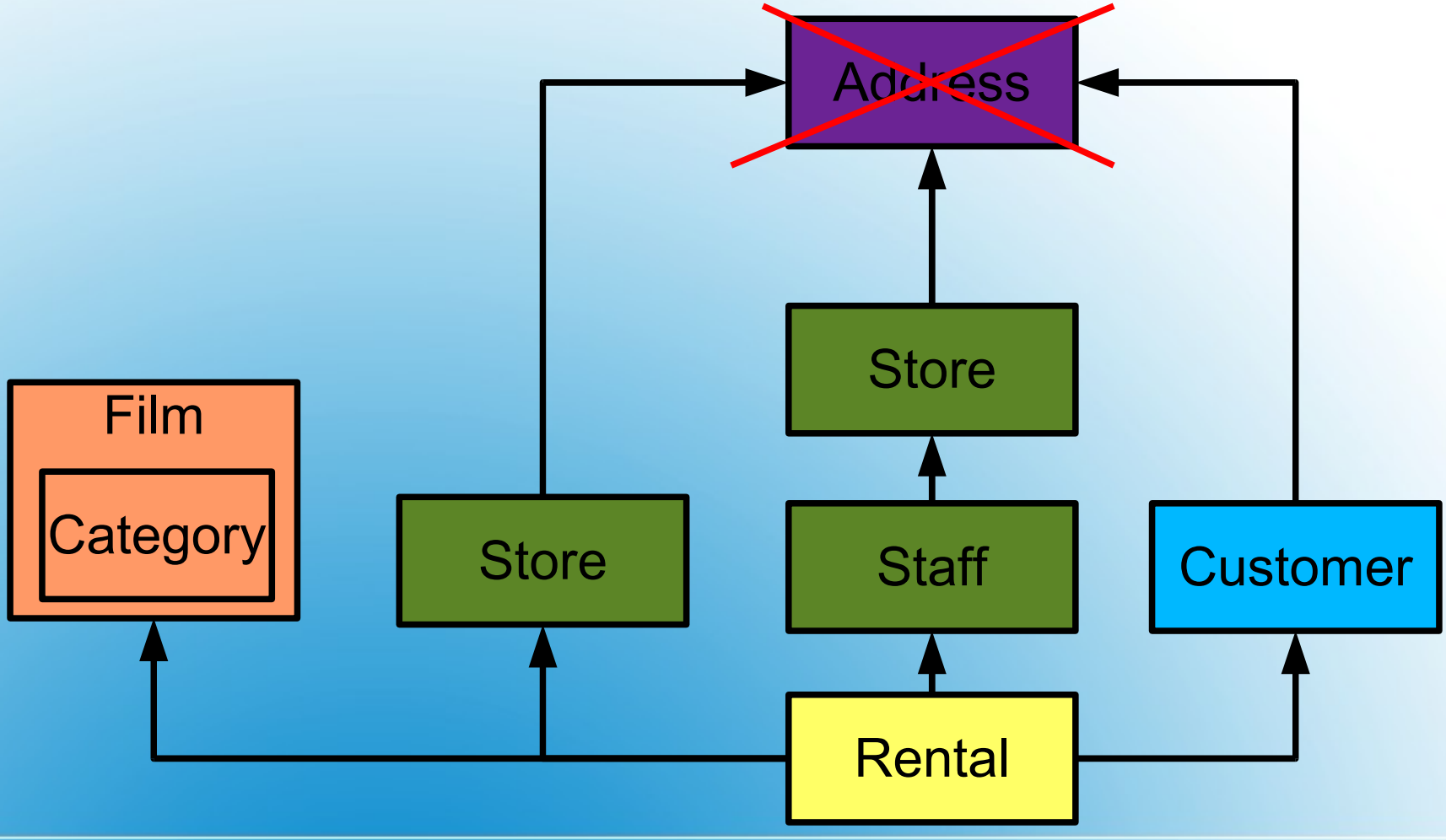
A star is born: Denormalize



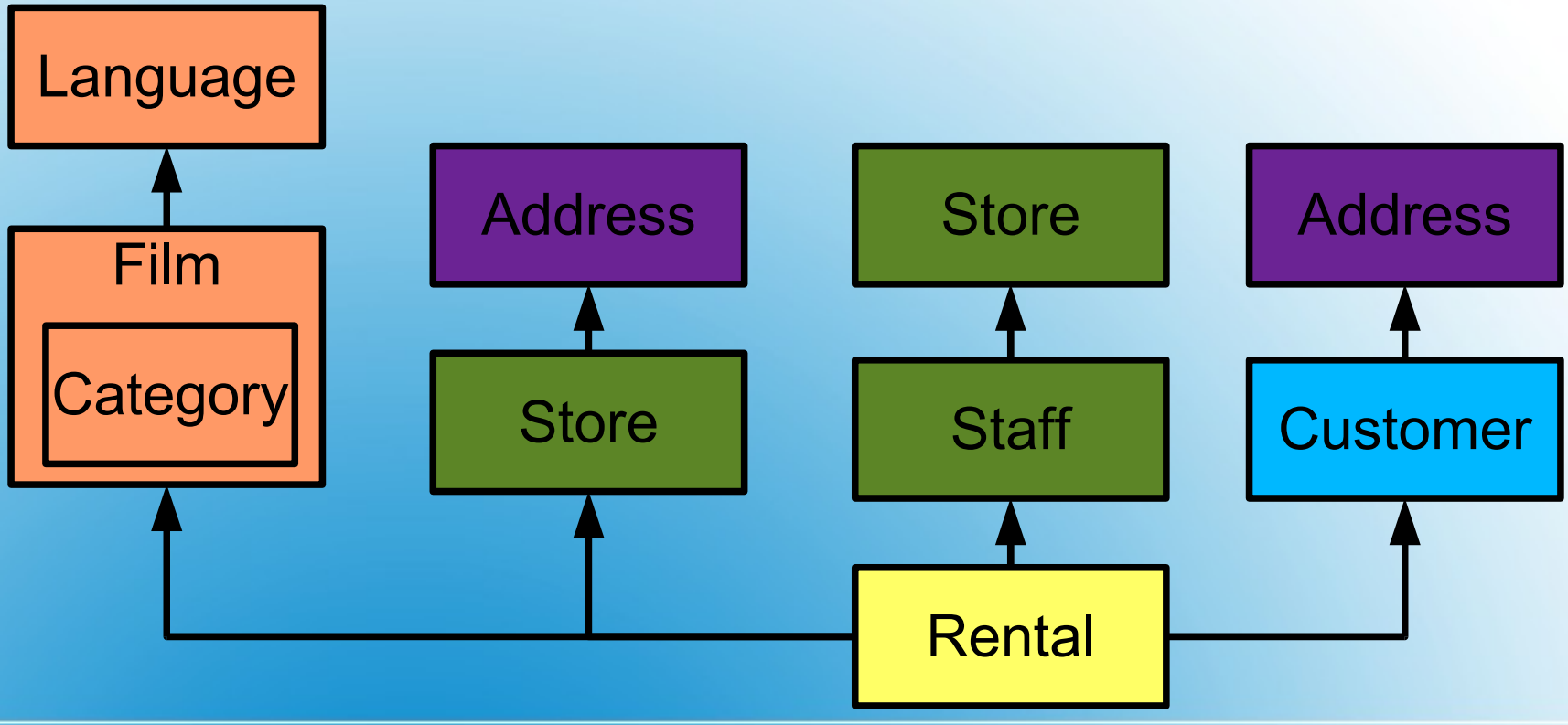
A star is born: Denormalize



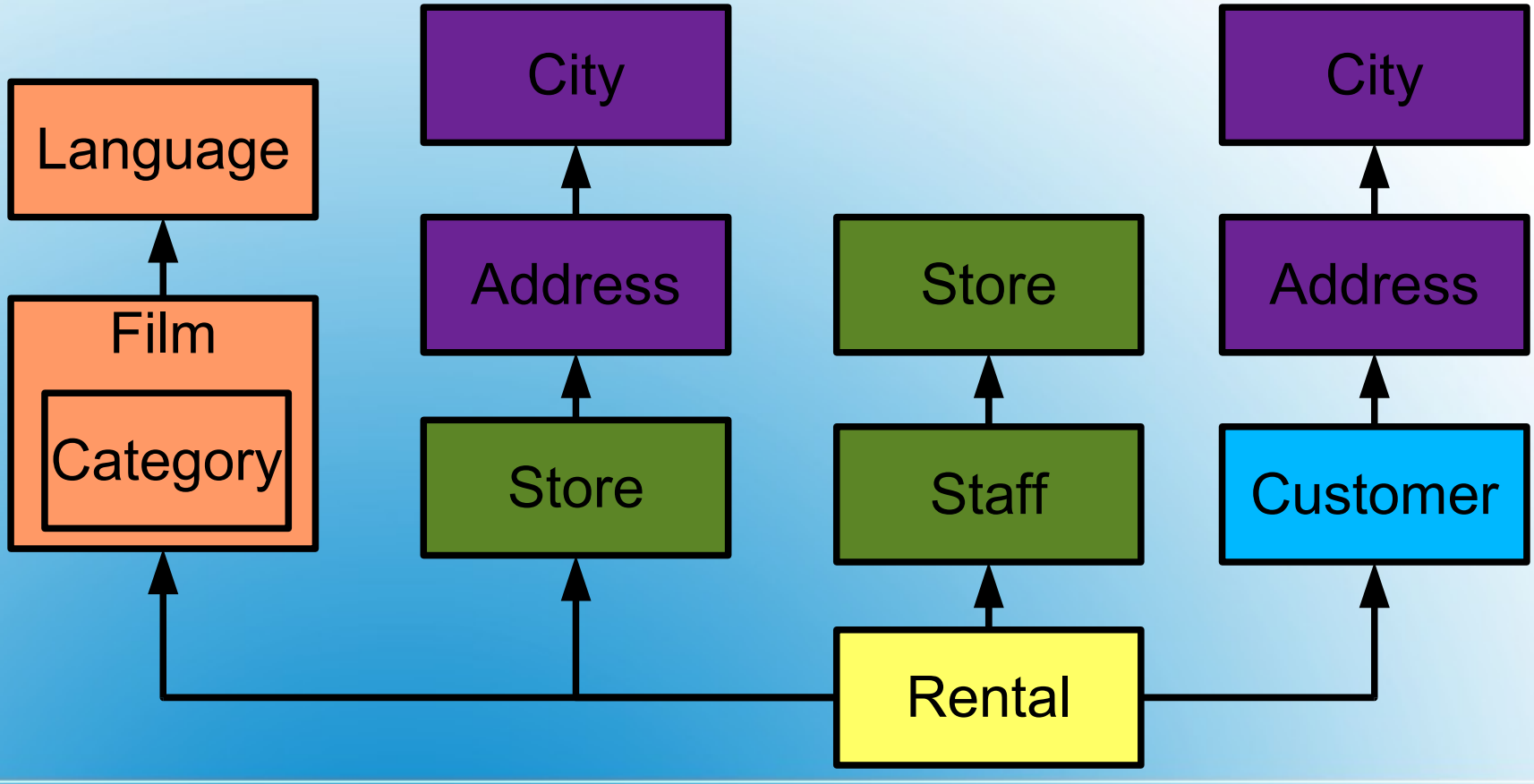
A star is born



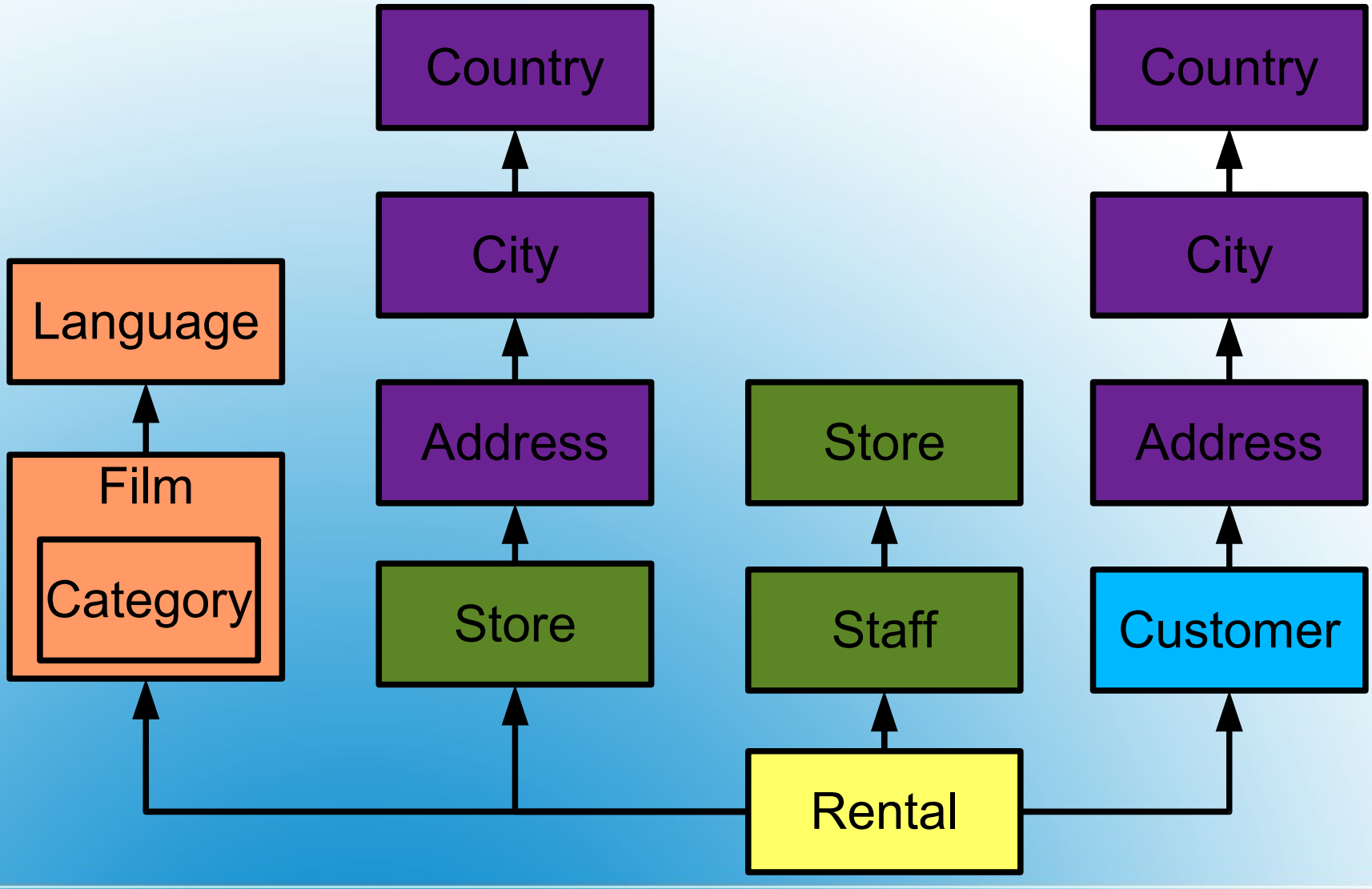
A star is born: Denormalize



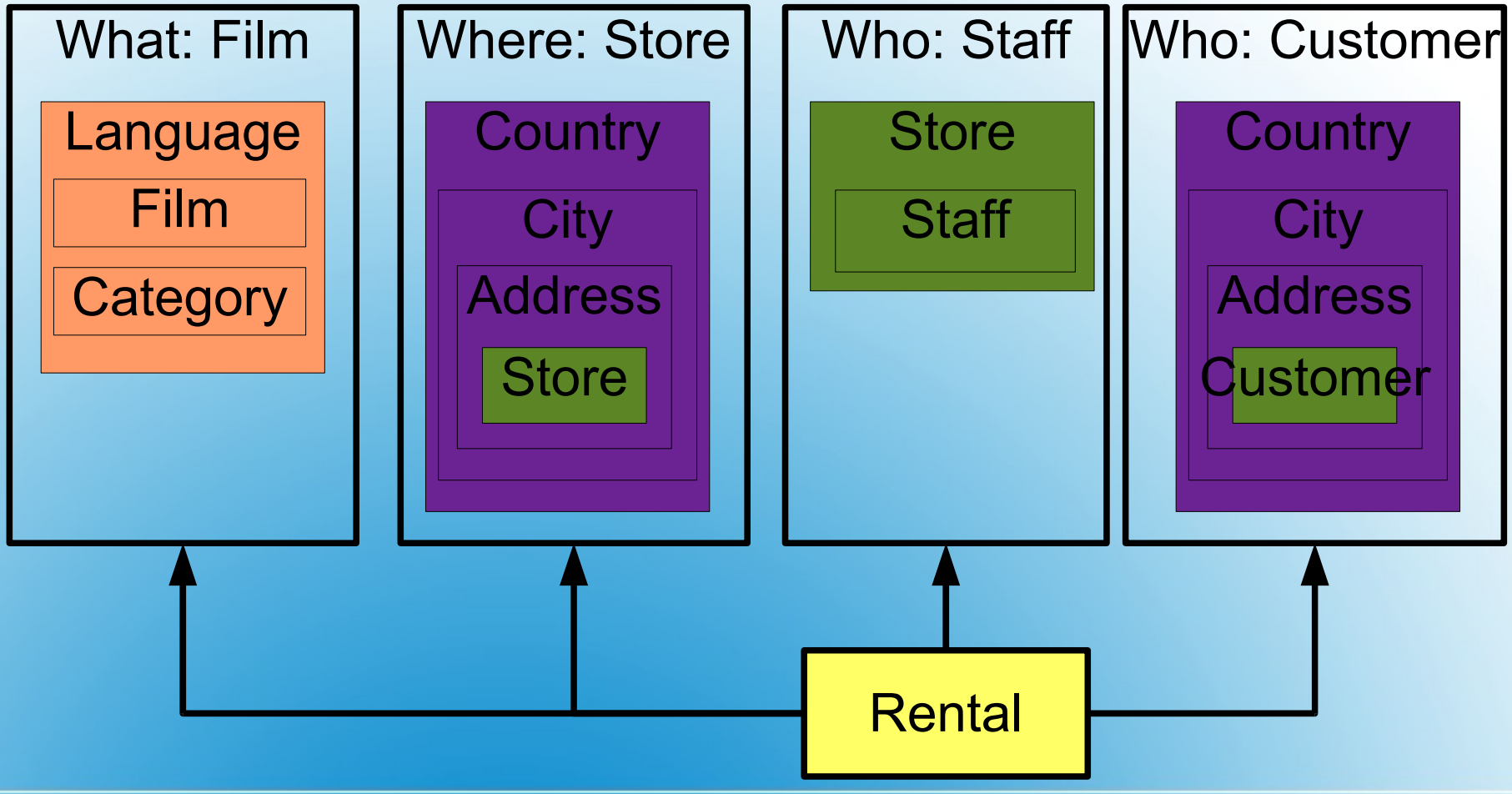
A star is born: Denormalize



A star is born: Denormalize



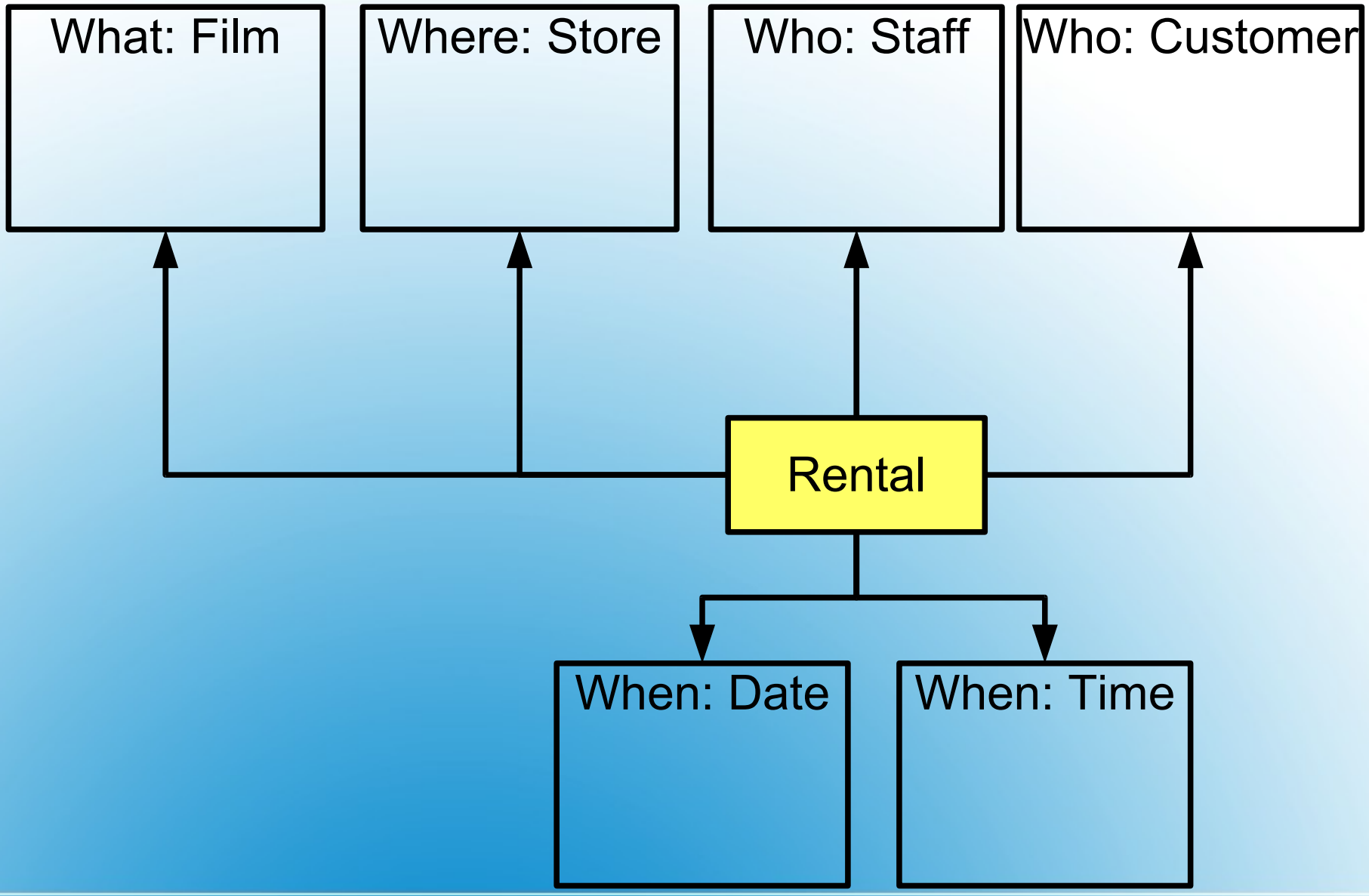
A star is born: Rental Snowflake



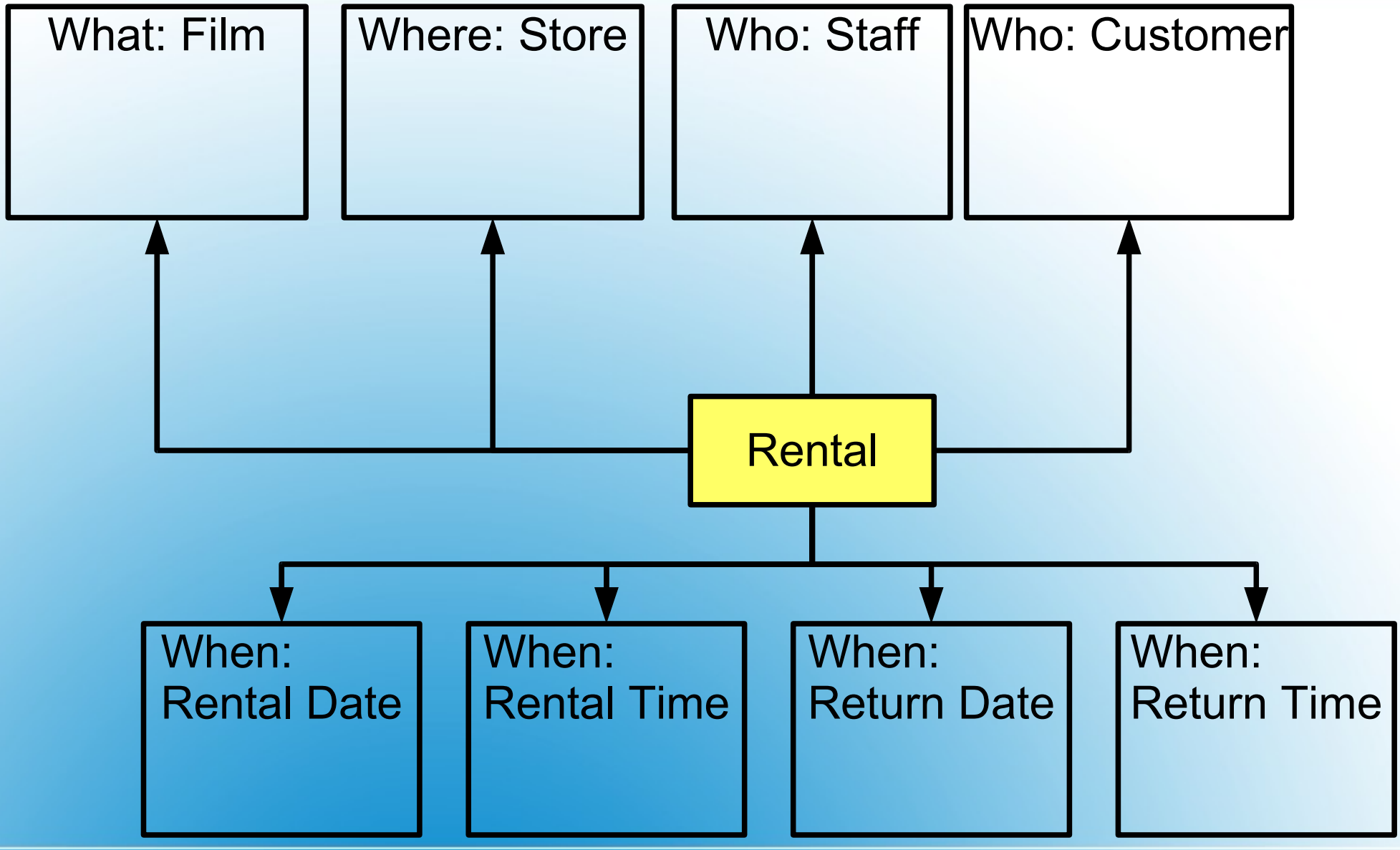
A star is born: Rental Star Schema

- Something is missing....
  - Who ? (Customer, Staff)
  - What ? (Film)
  - Where ? (Store)
  - .... ?

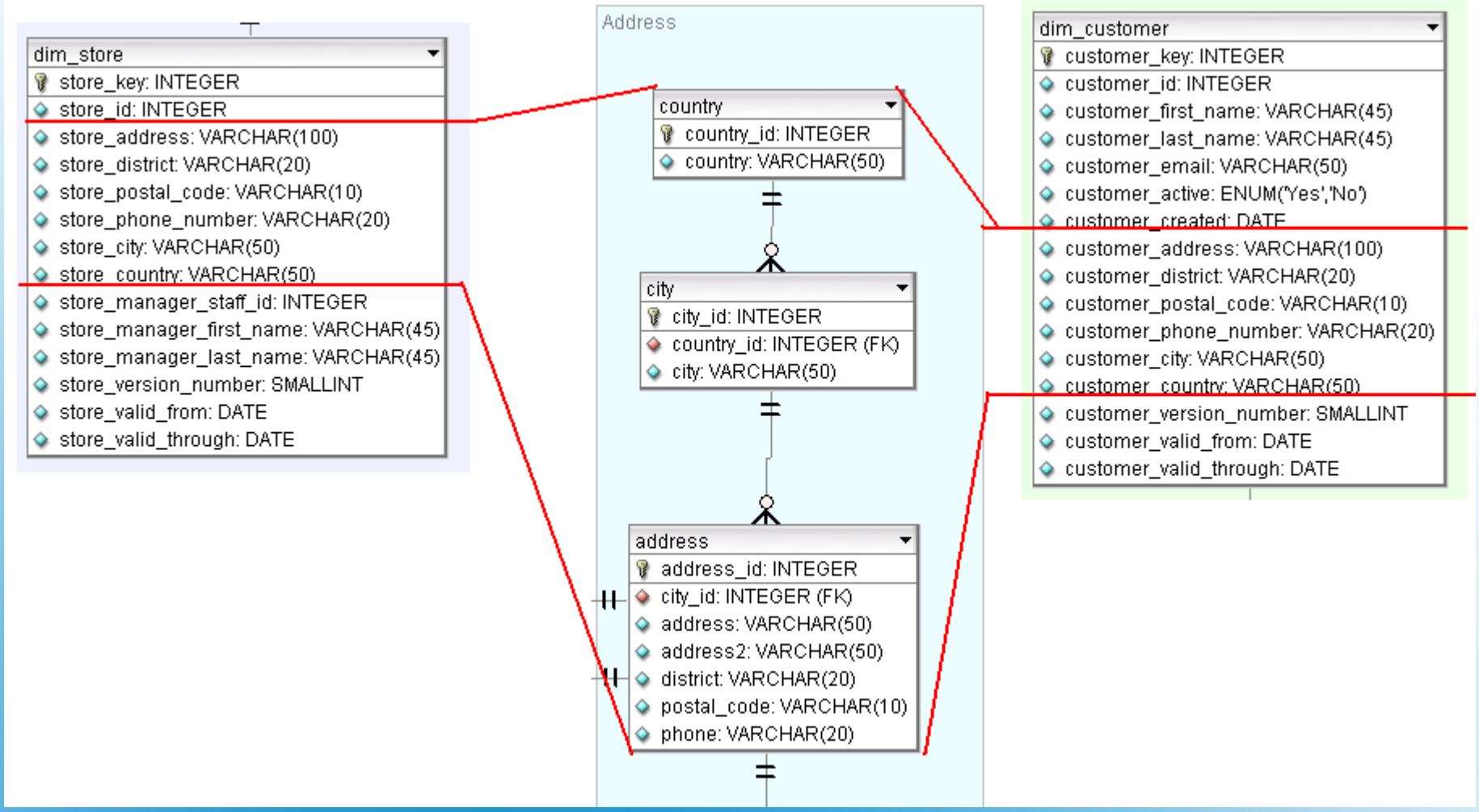
Dimensional Design



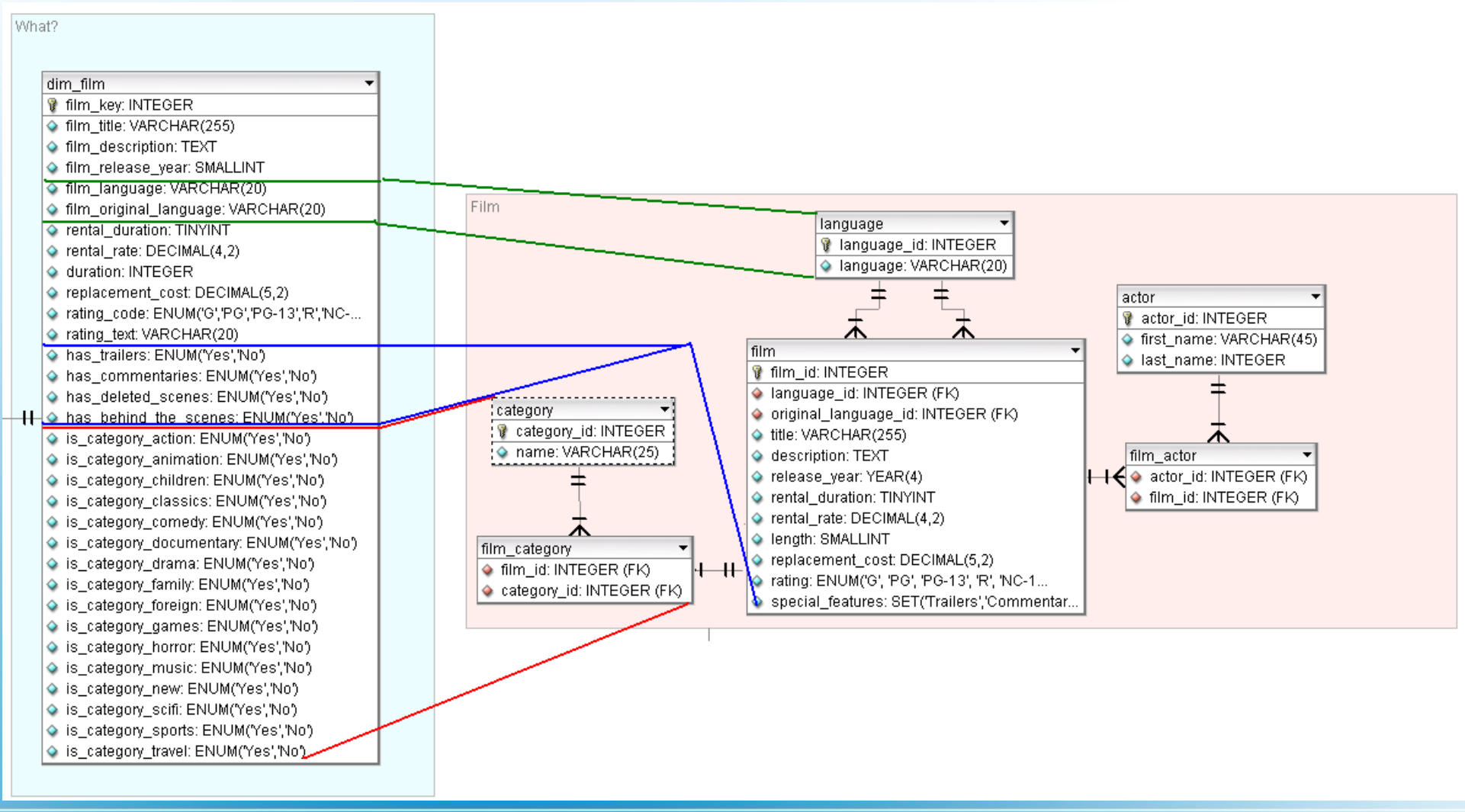
A star is born:
Rental Date and Time



Role Playing: Date/Time for both Rentals and Returns



Denormalization through Joins



Denormalization through Flattening (Repeating Group)

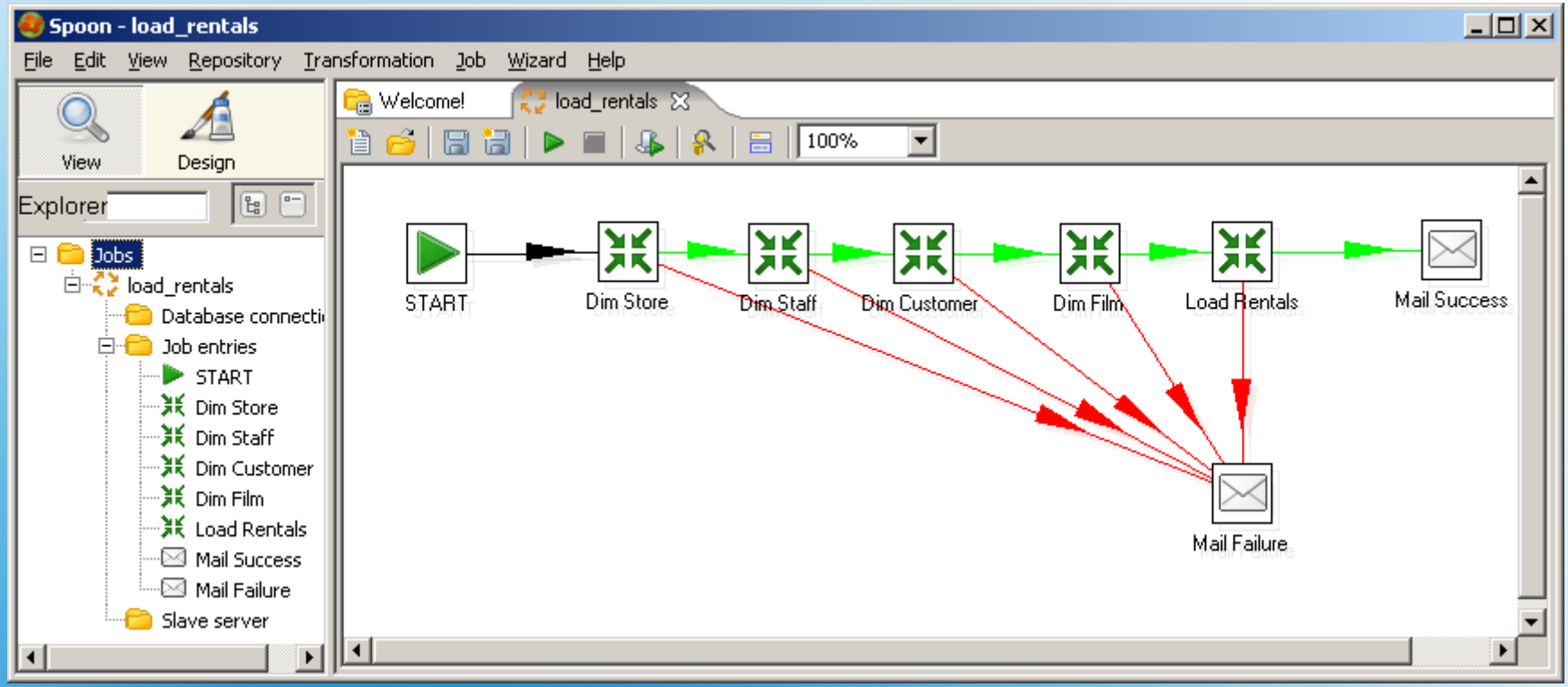
Loading a Data Warehouse

ETL with Pentaho Data Integration

- Pentaho Data Integration
  - sourceforge.net/projects/pentaho/
- ETL and much more
- Transformations:
  - Extract, Load and Transform
- Jobs:
  - Organize multiple transformations to a complete ETL process
- > 30 RDBMS-es, > 130 Transformation Steps

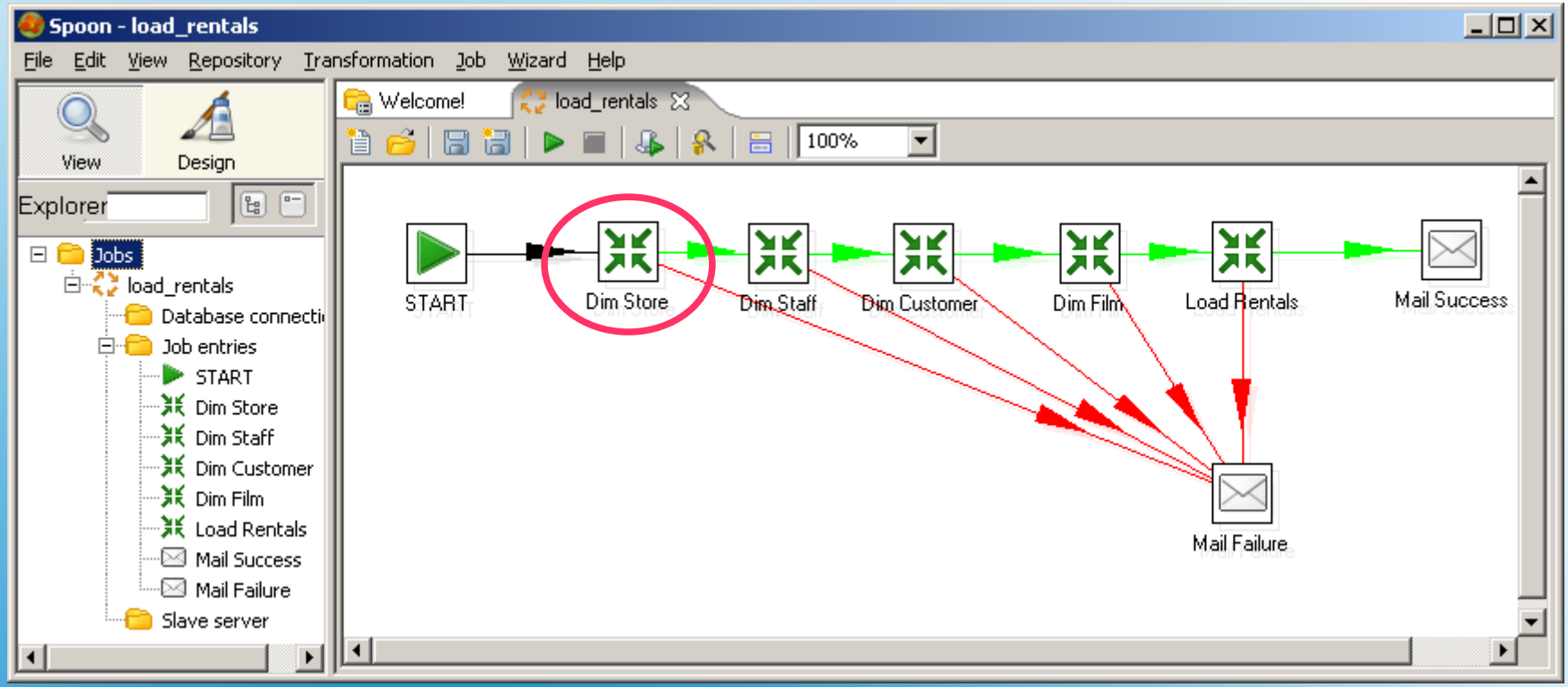
Dimensional Design

- First load dimensions, finally load fact
- Mail notification in case of success / failure



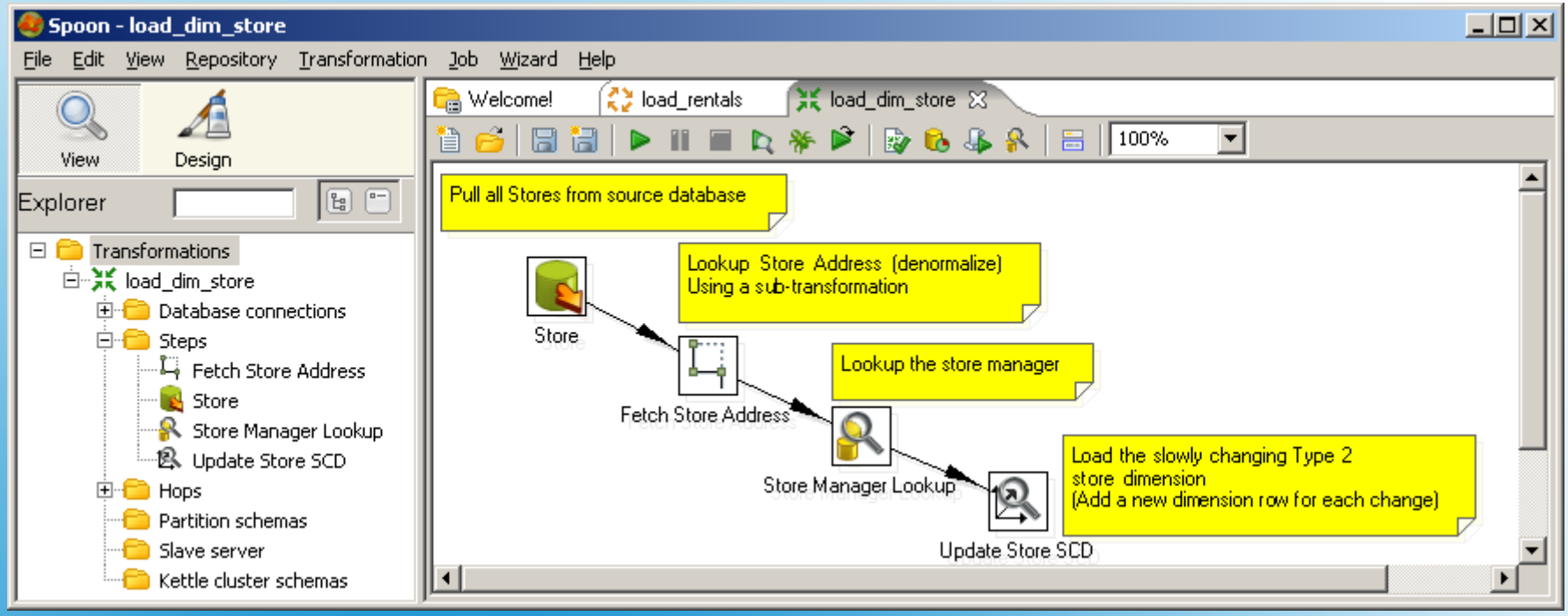
Job: Rental ETL Process

- First load dimensions, finally load fact
- Mail notification in case of success / failure



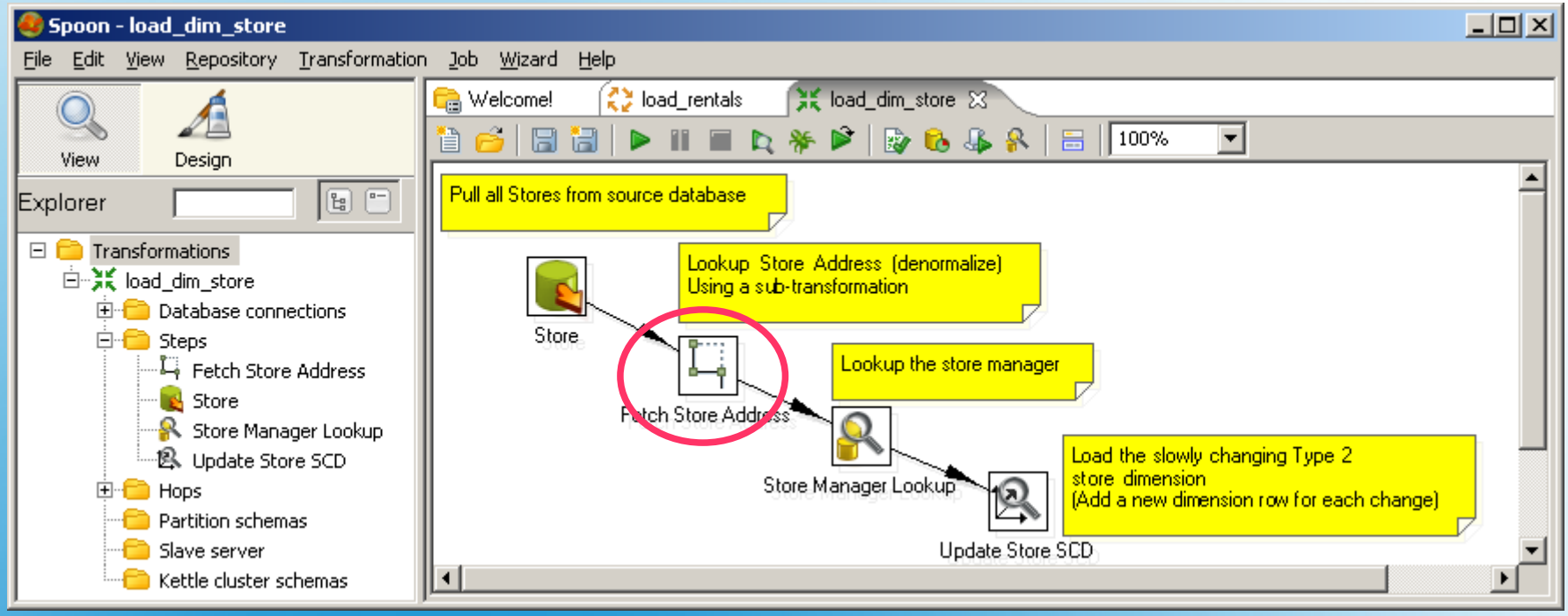
Job: Rental ETL Process

- Get store, lookup address (subtransformation) and manager
- Load store dimension table



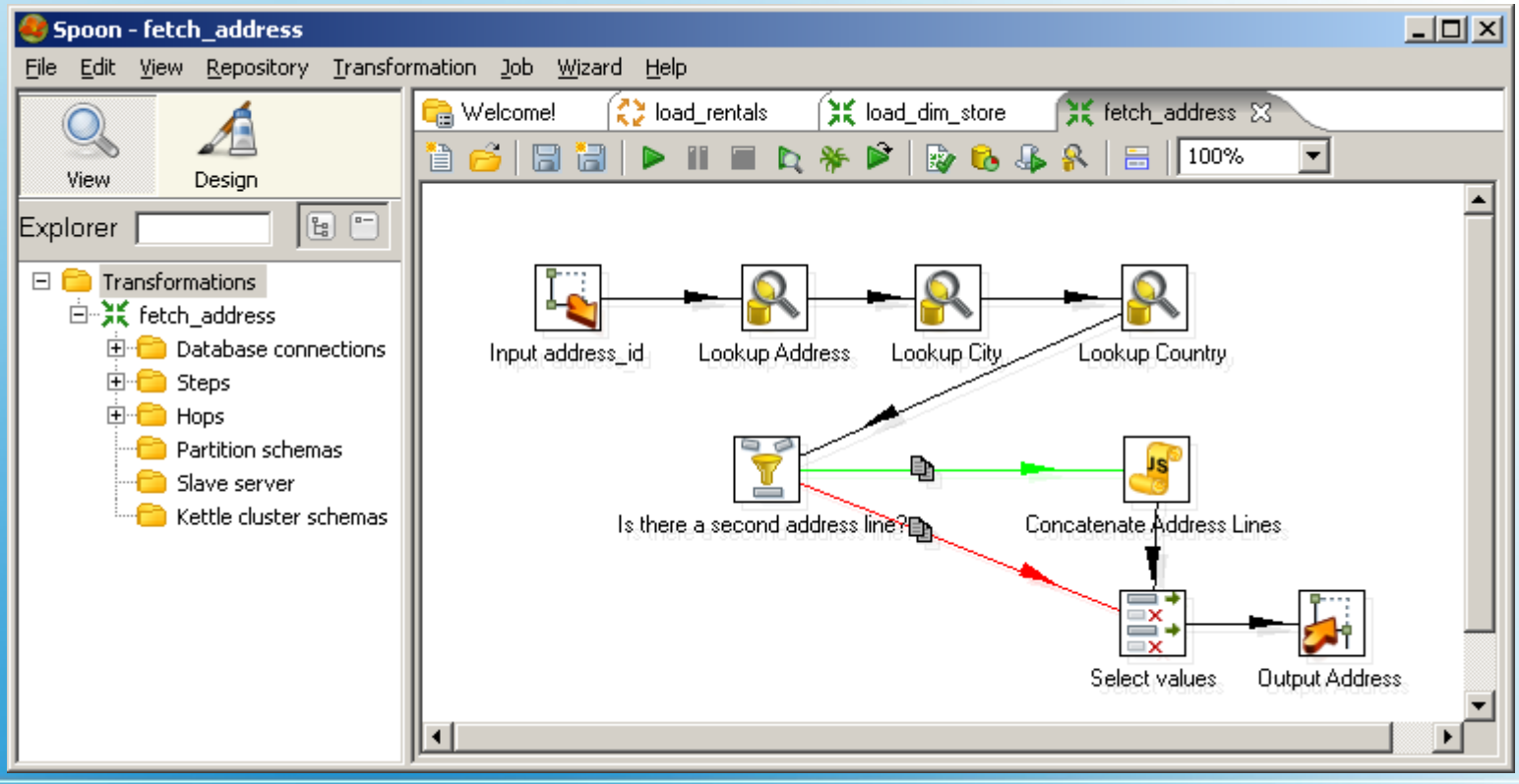
Job: Rental ETL Process

- Get store, lookup address (subtransformation) and manager
- Load store dimension table



Job: Rental ETL Process

- Get address, lookup city and country
- Concatenate address if necessary



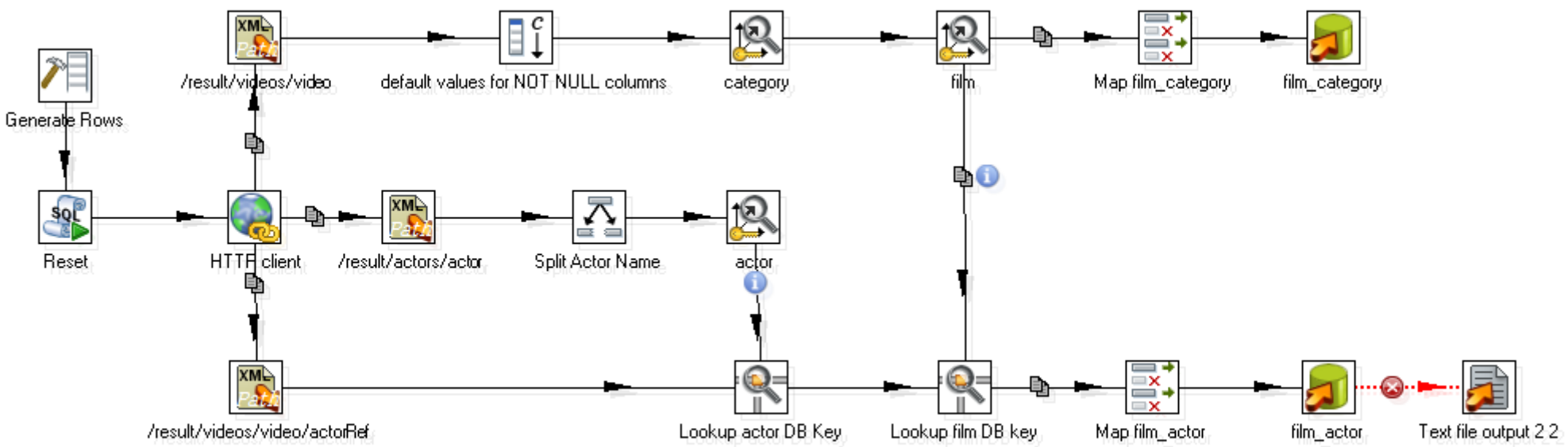
Job: Rental ETL Process

- This was just a simple example
- More complex example: importing XML

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <actors>
    <actor id="00000015">Anderson, Jeff</actor>
    <actor id="00000015">Anderson, Jeff</actor>
    ..
  </actors>
  <videos>
    <video>
      <title>The Fugitive</title>
      <genre>action</genre>
      ....
    </video>
    ...
  </videos>
</result>
```

Job: Rental ETL Process

- This was just a simple example
- More complex example: importing XML



Job: Rental ETL Process

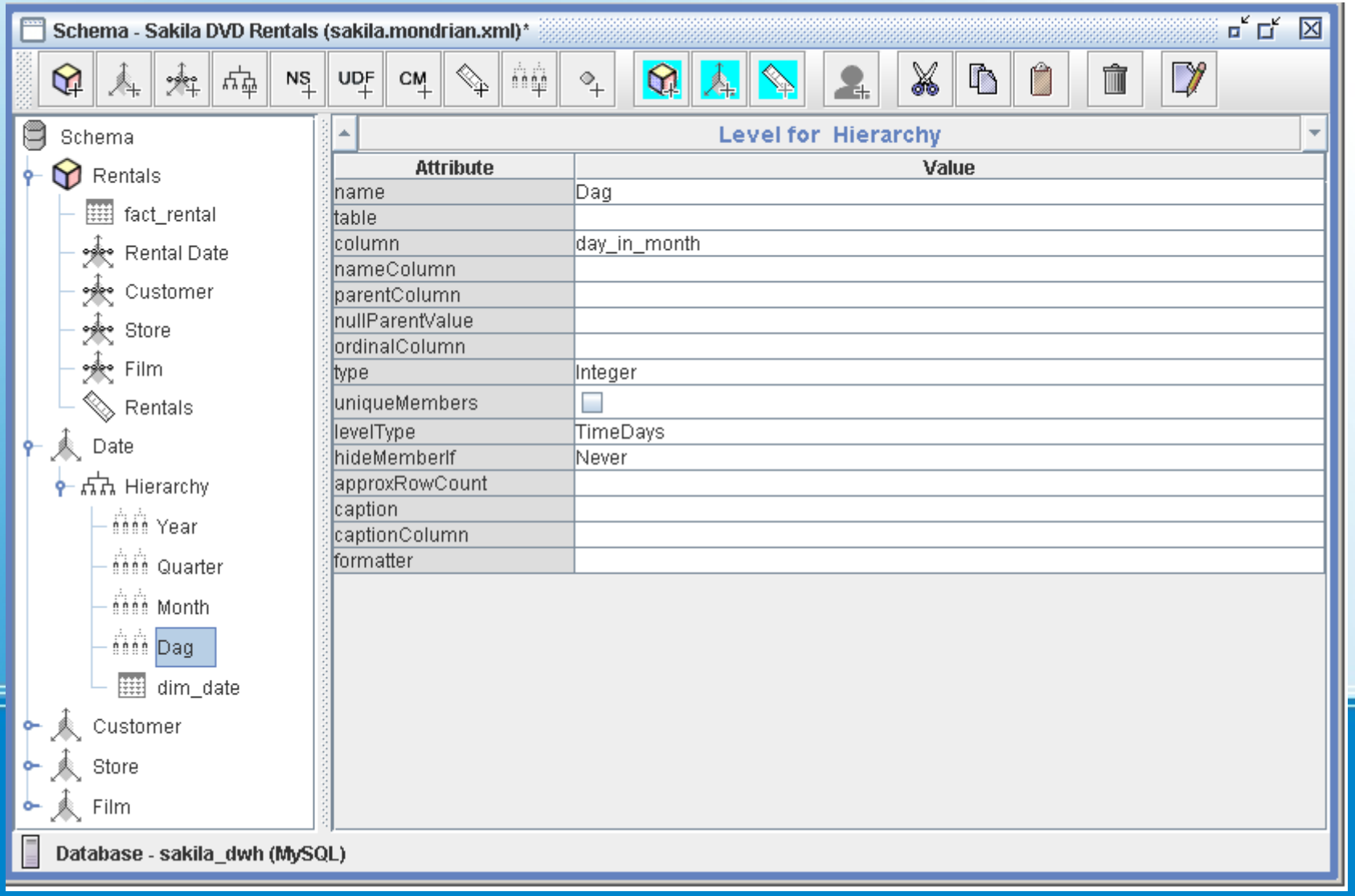
OLAP

OLAP Pivot Table with Pentaho Analysis Services

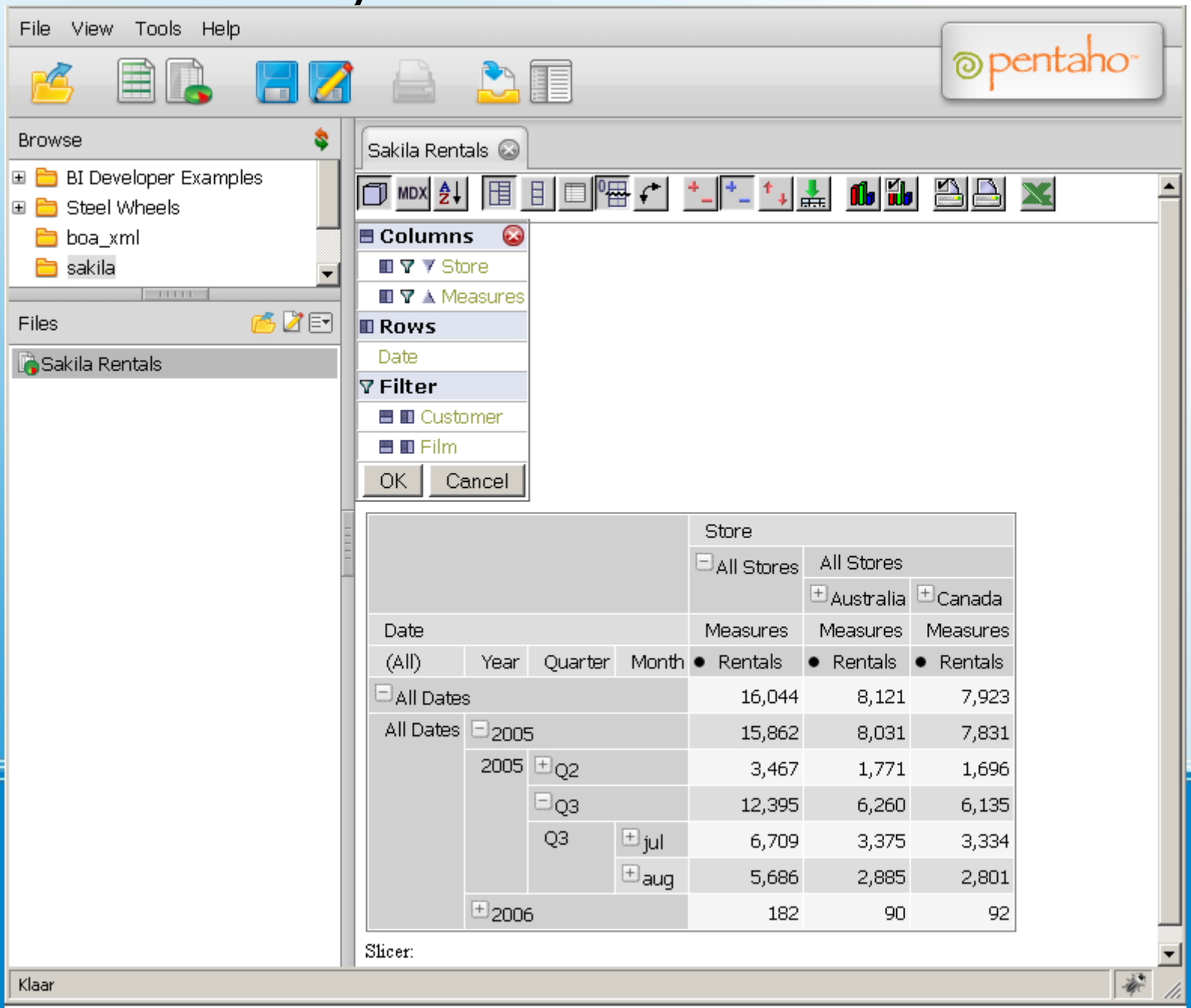
- Pentaho Analysis Services
  - Part of Pentaho BI Server
  - sourceforge.net/projects/pentaho/
  - Based on Mondrian ROLAP server
  - sourceforge.net/projects/mondrian/

Dimensional Design

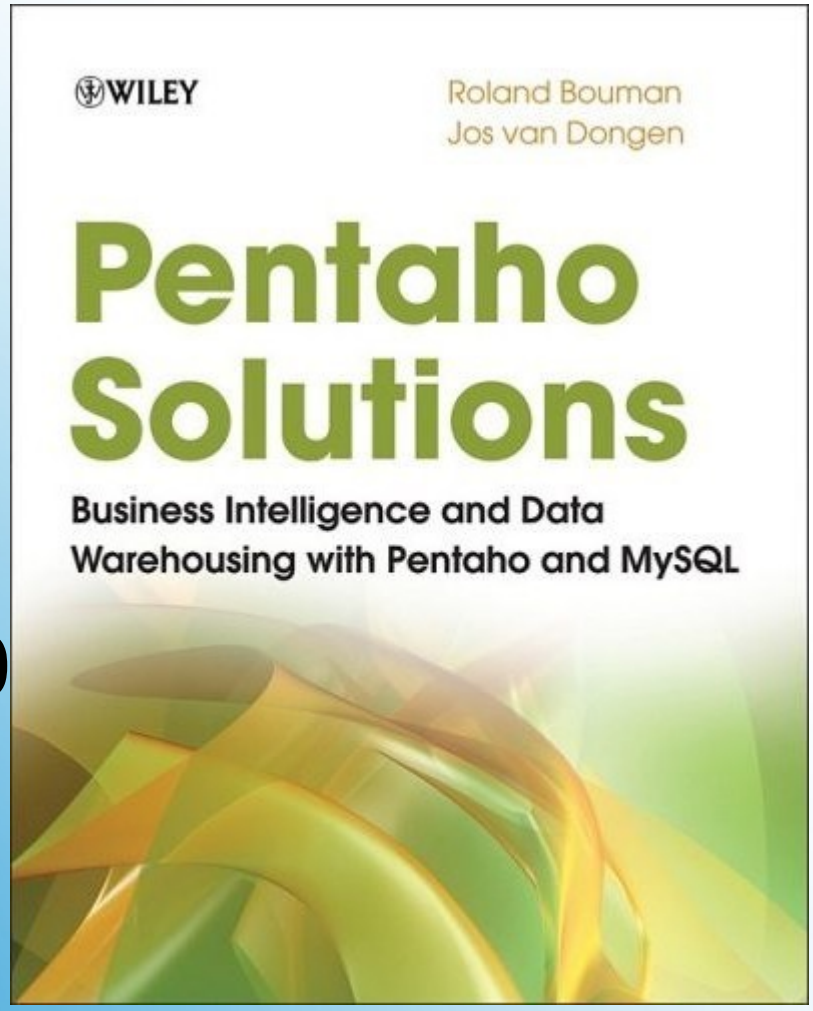
- Pentaho Schema Workbench
  - Map data warehouse tables to a logical Cube



- Pentaho Analysis View:



- Pentaho Solutions
  - Wiley
  - ISBN 978-0-470-48432-6
  - September 2009
  - 630+ page paperback
  - Amazon pre-order $31.50
  - Regular: $50.00



Upcoming Book: Pentaho Solutions