Explore Data Warehouses

Xujia Qin

Jun 15th, 2025

Q1: Fact Tables, Star Schemas, and OLAP in Relational Databases

Fact Tables

- Store **numeric business metrics** (e.g., sales, revenue, quantities).
- Each row represents a measurable event (like a transaction or shipment).
- Use **foreign keys** to link to **dimension tables** (e.g., Date, Product, Customer).

Star Schema

- A simple, denormalized design with:
  - A **central fact table** (containing the measurable data).
  - Connected directly to **dimension tables** (descriptive attributes).
- Visually resembles a **star**—hence the name!
- Optimized for fast querying and reporting.

Can a Transactional Database be Used for OLAP?

- Transactional (OLTP) databases are optimized for speed and data integrity during inserts/updates—great for daily business operations. However, they are **not suitable** for OLAP due to:
  - Poor performance with large, complex queries.
  - Normalized schemas not optimized for aggregations with redundent data
  - Overwhelming in CPU workload, OOM, or I/O bottleneck issues.
- Therefore, **OLTP systems should not be used to house OLAP**. Instead, ETL implementation is good into a dedicated warehouse design.

Why Use a Relational DB + Star Schema vs. NoSQL?

- While **NoSQL** systems (like MongoDB, Cassandra) offer horizontal scalability and flexibility, **relational databases with star schemas** are still preferred for OLAP because:
  - Powerful **query optimization** and **indexing**
  - Strong **data integrity** and constraints with ACID compliance
  - Better performance for complex **joins and aggregations** with star schemas
  - Mature **BI tool** (e.g., Tableau, Power BI, Looker)

Q2: Data Warehouse vs. Data Mart vs. Data Lake

Feature	Data Warehouse	Data Mart	Data Lake
Who uses it?	Big bosses & analysts	Specific Department (e.g.:Marketing/Sales teams)	Data engineers
Data Type	Clean, organized data	Clean, focused data	Raw, messy data
Speed	Pretty fast	Super fast	Slow (needs ETL cleaning)
Cost	High	Moderate	Low

What They Actually Do

- **Data Warehouse**: Like a company's main library - all central repository for integrated, the important, cleaned-up data in one place.
- **Data Mart**: Like a department's bookshelf - just a focused subset of a data warehouse intended for a specific team.
- **Data Lake**: Like a giant storage unit for raw, unstructured, or semi-structured data.

Real-Life Example

In my experience working on a data engineering internship, 1. Dumped all the raw data into an **AWS S3 data lake** (the "storage unit"as data lake) 2. Cleaned it up and moved it to **Snowflake** (the "main library" as datawarehouse) 3. Gave the marketing team their own **data mart** (the "special bookshelf")

Video Reference (https://www.youtube.com/watch?v=-bSkREem8dM)

Q3: Designing a Fact Table for the Sakila Database

Analytical Goal

I aim to analyze rental revenue and activity over time, broken down by staff, customer, and store. This will support monthly/quarterly revenue trends.

Star Schema Overview

We design a fact table named `fact_rental_revenue` and connect it with four dimension tables:

Table	Type	Description
`fact_rental_revenue`	Fact	Metrics such as revenue and late days
`dim_date`	Dimension	Rental date in various formats
`dim_customer`	Dimension	Customer information
`dim_staff`	Dimension	Staff who processed the rental
`dim_store`	Dimension	Store location

Fact Table: `fact_rental_revenue`

Each row corresponds to one rental transaction.

Column Name	Type	Description
rental_id	INT	Primary key (from rental table)
date_id	INT	Foreign key to dim_date
customer_id	INT	Foreign key to dim_customer
staff_id	INT	Foreign key to dim_staff
store_id	INT	Foreign key to dim_store
rental_amount	REAL	Amount paid for the rental (from payment)
late_return_days	INT	Return delay in days (return - rental date)

Dimension: dim_date

Column	Description
date_id	Surrogate primary key
date	Date (YYYY-MM-DD)
day	Day of month
month	Month number
year	Year
weekday	Weekday name
is_weekend	Boolean

Dimension: dim_customer

Column	Description
customer_id	Customer ID (PK)
first_name	
last_name	
email	
active	TRUE/FALSE

Dimension: dim_staff

Column	Description
staff_id	Staff ID (PK)
first_name	
last_name	
store_id	Foreign key to store

Dimension: dim_store

Column	Description
store_id	Store ID (PK)
address_id	To join with location
manager_id	

Q4: Create and Populate OLAP Fact Table in SakilaOLAP.db

In this step, 1. Create a new OLAP database: SakilaOLAP.db. 2. Design and create a fact table: fact_rental_revenue. 3. Extract relevant rental facts from the original sakila.db. 4. Populate the fact table in the new OLAP database. —

---

1. Connect to Databases

```
## [1] "Database Connected!"
```

---

2. Create Fact Table in OLAP Database

```
## [1] "Fact table is created!"
```

---

3. Extract and Transform Facts from sakila.db

I'll join the following tables:

- rental for rental info
- payment for rental revenue
- inventory for store ID
- staff and customer for foreign keys

```
##   rental_id    date_id customer_id staff_id store_id rental_amount
## 1         1 2005-05-24         130        1        1          2.99
## 2         2 2005-05-24         459        1        2          2.99
## 3         3 2005-05-24         408        1        2          3.99
## 4         4 2005-05-24         333        2        1          4.99
## 5         5 2005-05-24         222        1        2          6.99
## 6         6 2005-05-24         549        1        1          0.99
##   late_return_days
## 1                1
## 2                3
## 3                7
## 4                9
## 5                8
## 6                2
```

---

4. Load Data into OLAP Fact Table

```
##   row_count
## 1     16044
```

5. Preview Final Fact Table

```
##    rental_id    date_id customer_id staff_id store_id rental_amount
## 1          1 2005-05-24         130        1        1          2.99
## 2          2 2005-05-24         459        1        2          2.99
## 3          3 2005-05-24         408        1        2          3.99
## 4          4 2005-05-24         333        2        1          4.99
## 5          5 2005-05-24         222        1        2          6.99
## 6          6 2005-05-24         549        1        1          0.99
## 7          7 2005-05-24         269        2        2          1.99
## 8          8 2005-05-24         239        2        1          4.99
## 9          9 2005-05-25         126        1        1          4.99
## 10        10 2005-05-25         399        2        2          5.99
##    late_return_days
## 1                 1
## 2                 3
## 3                 7
## 4                 9
## 5                 8
## 6                 2
## 7                 4
## 8                 3
## 9                 3
## 10                6
```

Q5: Rental Analytics — Average Rental Amount Per Month (year 2005)

```
##   year_month avg_rental_amount
## 1    2005-05              4.17
## 2    2005-06              4.17
## 3    2005-07              4.23
## 4    2005-08              4.23
```

formatted table

Year-Month	Average Rental Amount ($)
2005-05	4.17
2005-06	4.17
2005-07	4.23
2005-08	4.23

Disconnect from Both Databases

```
## [1] "Database is disconnected!"
```