**Musical Instruments Shop and Rental**

A **shop** of musical instruments, which is also working as **rent shop** needs a database for storing their data and the information about purchases and rentals.

*NOTE: This is not a very suitable DB for application, it’s not a big online shop, but rather a small physical one. That fact affects the overall usability of this DB, and influenced the implementation of OLAP scheme, meaning “Why would someone build a Data Warehouse for small business like that?”. Anyway it was noticed too late and one’s best was done.*

This project implements a complete database solution for Musical Instruments Shop and Rental. It consists of two main components:

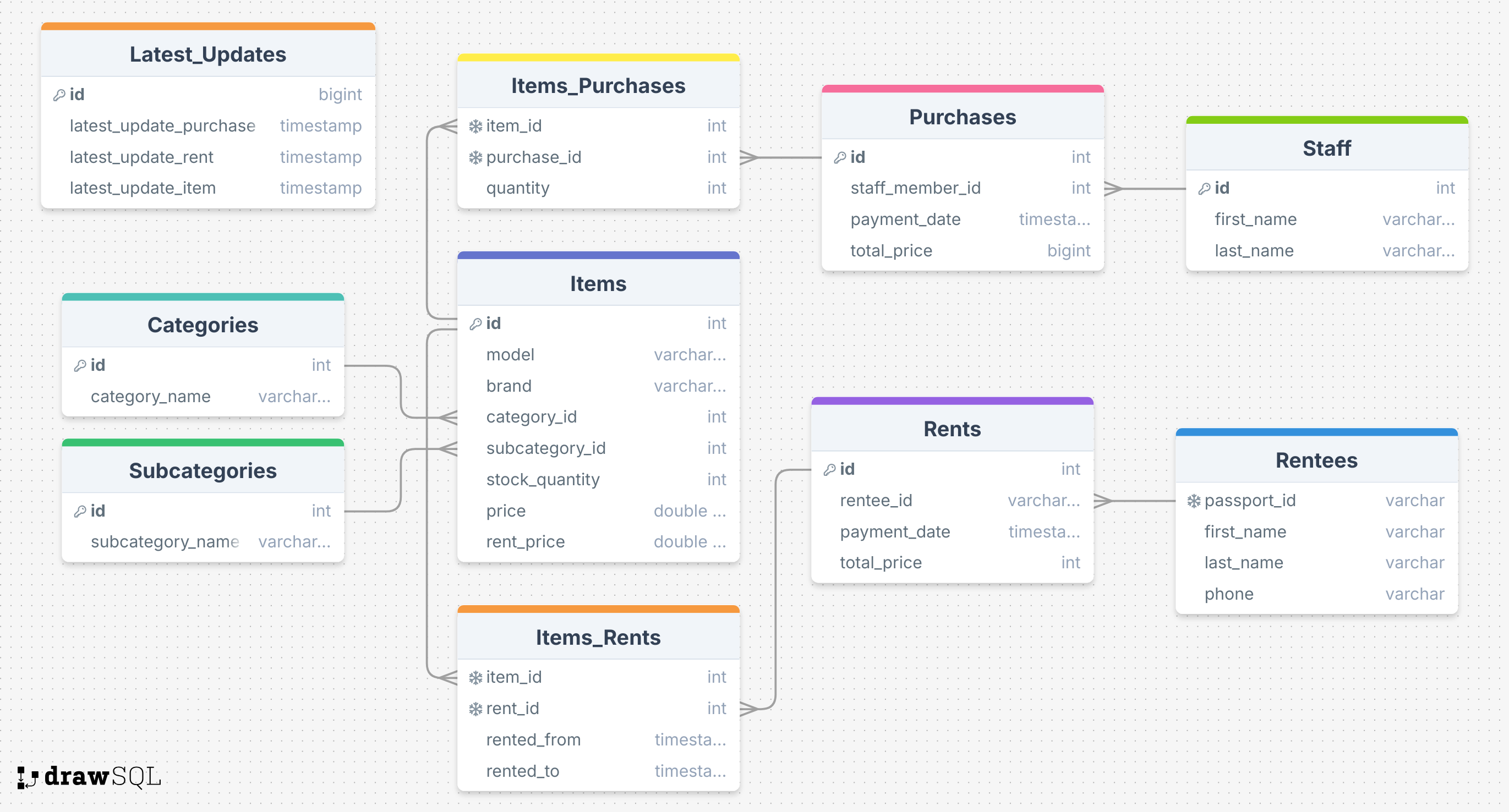
1. **OLTP (Online Transaction Processing)**: Designed to support real-time operational activities.
2. **OLAP (Online Analytical Processing)**: Designed for analytical data processing and reporting.

**OLTP Database Schema**

The OLTP database is designed in 3NF and consists of the following tables:

* **ITEMS**: Stores information about the items that are present in shop, Includes information about prices, prices for rental per week, brand, presence on stock.
* **CATEGORIES**: Represents instruments categories (guitars, keys, etc).
* **SUBCATEGORIES**: More detailed information about items categories.
* **STAFF**: Represents the employees that work in shop.
* **RENTEES**: Represents the people which have rented instruments, holds their names, passport data and phone, if present.
* **RENTS**: Contains rental information – who was the rentee, when the payment for rental service was made, price for the whole rental term (rental\_price \* weeks).
* **ITEMS\_RENTS**: Service table, which helps to implement many-to-many relational ship between item and rental act (one item can be in many rentals and one rental service can consist of many items). It also has information about period of rental term of one item.
* **PURCHASES**: Contains purchase information – who was the employee, who sold the instrument (*NOTE: no information about who bought it as it was seen as obsolete*), when the payment was made, price for the whole purchase (sum of prices of items in purchase \* their number).
* **ITEMS\_PURCHASES**: Service table, which helps to implement many-to-many relational ship between item and purchase act (one item can be in many purchases and one purchase can consist of many items). It also has information about number of same items in a purchase.
* **LATEST\_UPDATES**: Service table not connected to anything, needed for updating existing data with new one.

Another copy of schema can be found in repository.



**OLTP Database Schema**

The OLAP database follows a **snowflake schema** and includes the following tables:

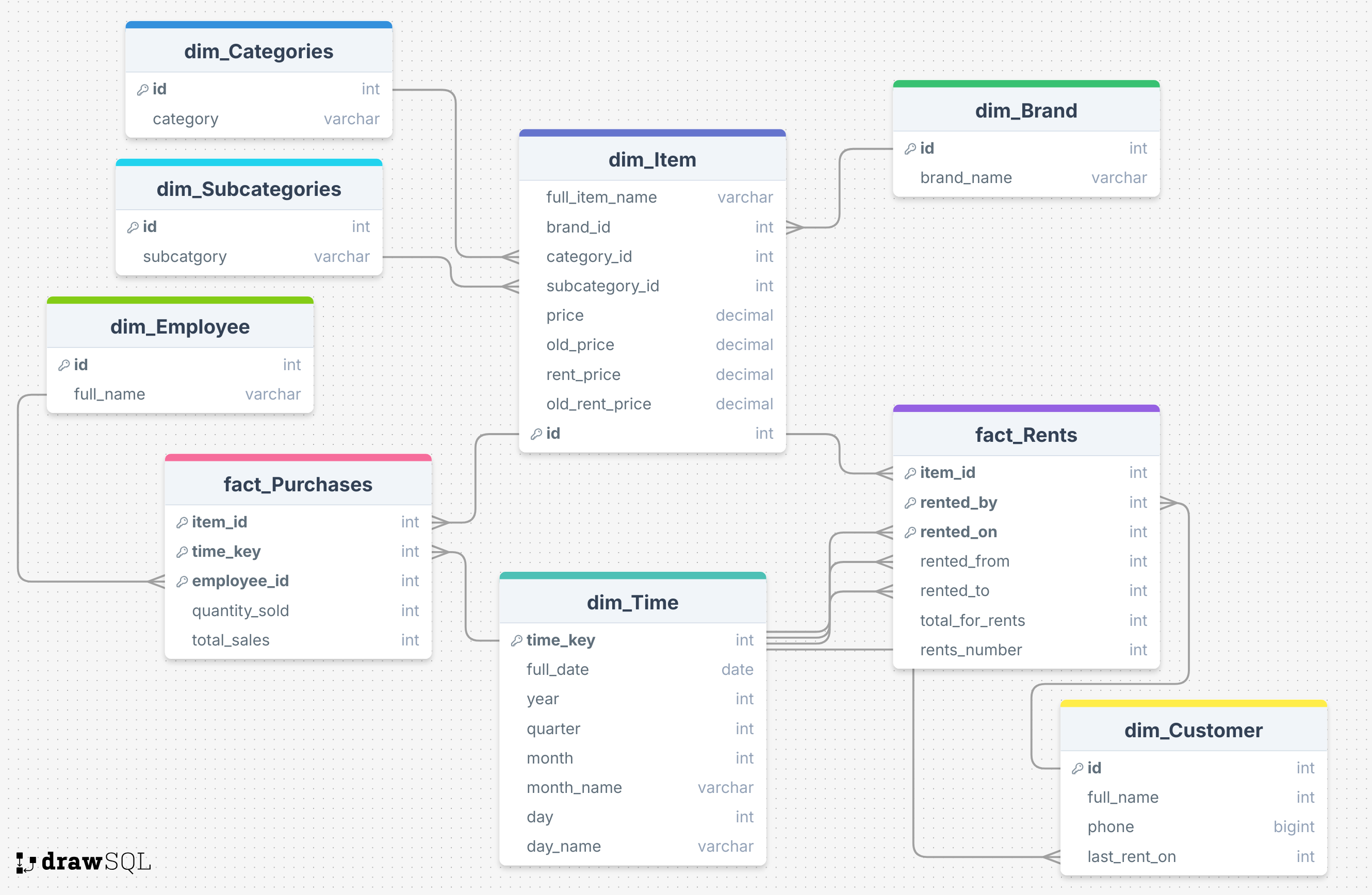
### **Fact Tables**

* **FACT\_PURCHASES**: Records purchases.
* **FACT\_RENTS**: Records rents.

### **Dimension Tables**

* **DIM\_CUSTOMER**: Stores customer information with SCD Type 2 for tracking last rental over time.
* **DIM\_PRODUCT**: Contains items details. Designed as SCD Type 2, as it can track the information about changing price of the instruments.
* **DIM\_CATEGORY**: Represents items category hierarchy.
* **DIM\_SUBCATEGORY**: Represents items subcategory hierarchy.
* **DIM\_TIME**: Captures time-related information for analytics.
* **DIM\_BRAND**: Stores brand details.
* **DIM\_EMPLOYEE**: Stores brand details.

Another copy of schema can be found in repository.



**Setup instructions**

*NOTE: In every script there are comments that might make easier to understand the code behind it. Though it still may be challenging as no obvious optimization or clever refactoring was made. But I hope it helps!*

OLTP creation:

* **Run the ‘OLTP create tables’ script**

ETL process:

* **Open ‘load data script’ and change placeholders with file paths from the OLTP data directory**
* **You can add new data to the files and run script again. However there are some restrictions (see Data instructions) on adding new data**, **if you want to test it.**

Query to OLTP solution:

* **Run the query in OLTP directory**

OLAP creation:

* **Run the ‘OLTP create tables’ script**

**Data instructions**

I tried to create as much ‘raw data’ as possible. All data is needed to be in .csv format.

* **Customers:** includes passport information, full name and phone alternatives,
* **Employees:** only employee names.
* **Items:** Full name, sub\category, prices and delivered date – date when item was shipped to the shop. The shipping date is criterion based on which item will be added while rerunning the script or not:
  + New item with shipping date > last remembered date in this file => item is added, date is updated.
  + Old item with shipping date > last remembered date in this file => old item is found (quantity + 1), date is updated.
* **Rents:** Who rented, what rented from and until when, when paid. Also ONE RENT is defined in script as one or several rows with SAME date\_time and SAME rentee passport. In real word terms = one customer has rented >=1 product. The ‘when paid’ date is criterion based on which item will be added while rerunning the script or not, just like in **Items**.
* **Purchases:** Who sold an item, what and when. Also ONE PURCHASE is defined in script as one or several rows with SAME date\_time and SAME employee name. In real word terms = one staff member has sold >=1 product to one customer. The when ‘date’ is criterion based on which item will be added while rerunning the script or not, just like in **Items**.