GENERAL REMARKS

1. **Duplicate rows in static tables:** In all tables containing static information, there may be several lines for the same store/product, although one row actually contains all the information. This applies to the tables "Store Info", "Product Categories", "Product Infos Static", “Bundle Infos" and “Sales Locations”. This is due to the fact that the information on the products can change. If this is the case, an additional row is created with the current information. In most cases, it makes sense to filter the respective table so that it only contains the latest entry for the respective product. This is easily possible using the "scrapingDate" column. The older information nevertheless remains in each table on purpose, as this could still be of interest for certain analyses.
2. **Keys connecting tables:** itemNumber, storeId
3. **Columns in JSON-format:** Several columns contain strings in the so-called JSON-Format. Examples are the column ‘extras’ in the “Opening Hours” table or “itemNumbersComponents” in the table “Bundle Infos”. These can be loaded using the json.loads() command, e.g.,

   ```
   json.loads(dfBundleInfos.itemNumbersComponents.iloc[0]),
   ```

   which returns the list `['20485765', '90494747']`.
4. **Datatypes:** After you have loaded one of the data tables, make sure that all columns have the correct data type. In this case, applying mathematical operations to these columns can lead to unexpected (incorrect) results. Two errors are common:
   a. Sometimes numeric columns like *stock* or *ratingCount* are regarded as strings rather than numbers. The datatype can be changed to a numeric type using `pd.to_numeric(dfPriceData['ratingCount'])`, for example.
   b. Date columns are often regarded as pure strings rather than the datetime format. The datatype can be changed to datetime format using

      ```
      pd.to_datetime(dfPriceData['date'], format='%Y-%m-%d')
      ```
5. **NONE Values:** Sometimes price or stock data is given as *None*. Usually this means that the product was not sold at this date.

STORE OPENING HOURS

- The table OpeningHours contains the column *extras* which contains data in the json format, e.g., {"smaland": [{"heading": "Aktuell geschlossen"}]}. In order to be able to use the information given here, meaningful transformations have to be applied first. NOTE: As of June 11, 2024, the above entries about the smaland being closed in Augsburg is the only relevant entry, anyway. As a consequence, not too much can be learnt from this information.

PRODUCT INFO

- The *productId* column is only required to be able to assign the variants listed in the variantsProductIds column to the item number in question. Variants are usually different color versions of the same product, e.g., a Billy shelf in black or wood colors instead of the white version.
- The *itemNoGlobal* column is not (!) used as a key for connecting to other tables.

PRODUCT CATEGORY PATH

	itemNumber	categoryKey	categoryName	categoryDepth	scrapingDate
62469	00507278	fu001	Möbel	1	2023-05-31
62470	00507278	fu003	Sofas	2	2023-05-31
62471	00507278	57527	Récamieren	3	2023-05-31
62472	00507278	47359	Relaxsessel	4	2023-05-31
114038	00507278	fu002	Stühle	2	2023-08-02
114039	00507278	fu006	Sessel & Récamieren	3	2023-08-02
114040	00507278	16239	Sessel	4	2023-08-02
114041	00507278	47359	Relaxsessel	5	2023-08-02

As in the other tables, which contain static information, there can be rows for an itemNumber at several points in time, provided the original information has been updated by IKEA. In this case, it is again important to ensure that only the rows with the most recent time entries are used.

In contrast to the other tables with static information, there are always several rows for an itemNumber due to the different depths of the categories. This must be taken into account when filtering this table. As seen in the above example, the updates can also be done separately for specific depths.

PRICE DATA

The prices of each product are identical for each store. As such, there is no *storeId* column in this table.

STOCK DATA

1. **Stock values:** The stock values given here are identical to the number of available units in the specific store as depicted on a product's page on the IKEA website. They are measured at 3 am every day, so they refer to the stock value at the end of the prior day.
2. **Sales values:** The difference in stock values between successive days can be interpreted as the sales value of the specific day, but with caution because returns are not factored in. It is a good approximation of reality, especially for fast-moving products, nevertheless. On days on which a product is restocked due to a delivery, it is not possible to determine the sales quantity using the difference between the stock values of successive days.
3. **Restocking:** A positive difference in the stock values between successive days can be interpreted as a restocking, but again with caution. For slow-moving products in particular, a positive difference can also simply result from customer returns. As a rule, a threshold should therefore be set above which a positive difference in stock values is to be interpreted as restocking. This threshold value must either be selected as a minimal relative difference of the stock values or determined individually for each product.
4. **Stock values of Bundles (important!):** Many products are a bundle of different single items, e.g.: https://www.ikea.com/de/de/p/billy-oxberg-buecherregal-mit-tueren-weiss-s29281066/
   This is a BILLY bookcase with two doors, which consists of a BILLY shelf and 2 OXBERG doors. The displayed value of available units simply corresponds to the minimum of the two individual products. Accordingly, a change in the available units for bundles cannot be interpreted as a sales quantity, as the number of available units also changes if, for example, OXBERG doors are sold independently of the bundle under consideration. Inventory management analyses should, therefore, be limited to products that are not bundles. The

BundleInfos table can be used to filter out all itemNumbers that represent a bundle

BUNDLE INFOS

Each itemNumber listed here is a bundle. In the column *itemNumbersComponents* a list of all itemNumbers that constitute the bundle at hand is given in JSON-format.

BUNDLE INFOS INVERSE

Each itemNumber listed here is a product that is not a bundle. In the column *itemNumbersBundles,* a list of all bundles that contain the item at hand is given in JSON-format.

SALES LOCATIONS

For each itemNumber that is not a bundle (!), this table shows where the product can be found in the respective IKEA store.

Not each itemNumber (that is not a bundle) has a sales location in every store. Some products are simply not sold in some stores.