Assignment 2

Data Collection and Preparation for the Analysis of Churn Reasons

The increasing number of cancelled subscriptions has put Zenara, a music streaming service, in a difficult position. Investigating this situation and identifying the underlying causes has highlighted the need for a detailed sales and subscription data analysis. The first step involves determining the available data related to individual subscribers in the company's relational database (using SQLite technology). At the request of Zenara's management, the sales team collected and briefly described the recorded data, which is shown in Table 1.

Table	Field Name	Field Description
Subscriptions	SubscriptionID [PK]	Subscription ID
	STATUS_PORTFOLIO [FK]	Subscription status in the portfolio system
	STATUS_REASON [FK]	Reason for subscription status in the portfolio system
	TECHNICAL_COMMENCEMENT_DATE	Technical start date of the subscription
	VERSION_START_DATE	The start date of the current subscription version in the portfolio system
	PRODUCT	Product code
	PRODUCT_TYPE (Old/New)	Product type (Old/New)
	AGENCY	Name of the customer service agency providing the subscription
	REGION	Subscriber's state of residence
	CITY	Subscriber's city of residence
	DEVICE	The primary device used for the application
	MONTHLY_FEE	Monthly subscription fee (in USD)
	ClientID [FK]	The ID of the subscriber
StatusCodes	STATUS_PORTFOLIO [PK]	Subscription status in the portfolio system (all possible codes listed)
	Description	Textual description of the status codes
ChurnCodes	STATUS_REASON [PK]	Reason for subscription status in the portfolio system (all possible codes listed)
	Description	Textual description of the reason codes
Client	ClientID [PK]	Customer ID

	BIRTH_DATE	Subscriber's birthdate
	Sex	Subscriber's registered biological sex

Table 1: Scope of Subscriber Data
Legend: PK = Primary Key; FK = Foreign Key

In the second step, the available data must be reviewed and interpreted before the in-depth analyses to identify the necessary data preparation steps. The service provider has been recording its subscribers' data into a database, which is contained in the SQLite database ***zenara_subscriptions.db***, since February 2020, when new services (products) were introduced due to the curfew restrictions caused by the coronavirus epidemic. Before answering questions from the business side, let's review the available data.

Task 2.1

a) Identify the data type for the fields in the ***Subscriptions*** table. Distinguish between four data types in this exercise:
   - Categorical (Nominal)
   - Date
   - Discrete numerical value (small set of values compared to the number of observations)
   - Continuous numeric (large set of values compared to the number of observations)

b) Let's examine the distribution of the TECHNICAL_COMMENCEMENT_DATE, PRODUCT, MONTHLY_FEE and BIRTH_DATE fields.
   - Among the given fields, for categorical and date-type fields, simply use the appropriate aggregation operation to examine the data values and their frequencies; for discrete and continuous value fields, use the pandas data frame histogram method.
   - Try to give possible explanations for the possible anomalies, and in the following exercises, investigate what the real cause of the possible anomalies might be.

Morgan asked the analytical team to answer several questions in preparation for the drop-out survey. *During the analyses, it should be considered that the data was extracted on 01.03.2024.*

Task 2.2

Calculate each subscriber's age (as a whole) at the time of the query and add the result as a new field to the Subscriptions table. When calculating, consider that the date of birth and other data of subscribers born before 1980 are recorded in an older customer data management system.

The client's age should be given as a part of the result obtained from the formula (date of query - date of birth)/365.25.

Task 2.3

Subscribers can set the device from which they prefer to use *Zenara* services. The company distinguishes between three main types of devices:

- **Mobile Device**: any mobile device (e.g. phone, tablet).
- **TV**: for use via smart TV.
- **Browser**: use the *Zenara* web application from an internet browser.

In the sales department's records, these three device types are stored under different codes appearing in the DEVICE field.

Your task is to unify the codes in a new field (within the *Subscriptions* table) so that only the three cases above are distinguished by the new coding!

Task 2.4

For subscriptions served by a Central Online Customer Contact Agency, only the subscriber's city was recorded as geographic data, while for other subscriptions, only the subscriber's state was recorded. In a new field, the two-letter abbreviation of the subscriber's state is uniformly entered according to the ANSI standard.

For a solution, you can use the tables in the relevant Wikipedia entries :[1]

- https://en.wikipedia.org/wiki/List_of_United_States_cities_by_population

[1] For the solution, we can assume that the population of the subscriber's city is at least 100,000. Incidentally, in the US, the use of counties and other administrative units in addition to the state would be very important in the management of place names. There can be identical place names in the same state but different counties.

Python packages and functions that can be used in the solution

Packages

- pandas: https://pandas.pydata.org/
- numpy: https://numpy.org/
- sqlite3: https://docs.python.org/3/library/sqlite3.html
- beautifulsoup4: https://www.crummy.com/software/BeautifulSoup/bs4/doc/
- urllib.request: https://docs.python.org/3/library/urllib.request.html#module-urllib.request

Task 2.0

- sqlite3.connect: https://docs.python.org/3/library/sqlite3.html
- sqlite3.cursor: https://docs.python.org/3/library/sqlite3.html
- sqlite3.execute: https://docs.python.org/3/library/sqlite3.html
- sqlite3.fetchall: https://docs.python.org/3/library/sqlite3.html
- pandas.read_sql_query: https://pandas.pydata.org/docs/reference/api/pandas.read_sql_query.html
- pandas.DataFrame.info: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.info.html
- sqlite3.close: https://docs.python.org/3/library/sqlite3.html
- pandas.DataFrame.copy: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.copy.html

Task 2.1

- pandas.DataFrame.groupby: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.groupby.html
- pandas.DataFrame.count: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.count.html
- pandas.DataFrame.plot.bar: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.bar.html
- pandas.to_datetime: https://pandas.pydata.org/docs/reference/api/pandas.to_datetime.html
- pandas.Series.dt.to_period: https://pandas.pydata.org/docs/reference/api/pandas.Series.dt.to_period.html
- pandas.DataFrame.plot: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.html
- pandas.DataFrame.hist: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.hist.html

Task 2.2

- pandas.Series.str.split: https://pandas.pydata.org/docs/reference/api/pandas.Series.str.split.html
- pandas.DataFrame.info: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.info.html
- pandas.to_numeric: https://pandas.pydata.org/docs/reference/api/pandas.to_numeric.html

- pandas.Series.str.replace: https://pandas.pydata.org/docs/reference/api/pandas.Series.str.replace.html
- numpy.where: https://numpy.org/doc/stable/reference/generated/numpy.where.html
- pandas.Series.str.strip: https://pandas.pydata.org/docs/reference/api/pandas.Series.str.strip.html
- pandas.unique: https://pandas.pydata.org/docs/reference/api/pandas.unique.html
- pandas.DataFrame: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html
- pandas.DataFrame.merge: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.merge.html
- pandas.DataFrame.rename: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.rename.html
- pandas.to_datetime: https://pandas.pydata.org/docs/reference/api/pandas.to_datetime.html
- dict: https://www.w3schools.com/python/ref_func_dict.asp
- pandas.Series.dt.days: https://pandas.pydata.org/docs/reference/api/pandas.Series.dt.days.html
- numpy.floor: https://numpy.org/doc/stable/reference/generated/numpy.floor.html
- pandas.DataFrame.astype: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.astype.html

Task 2.3

- pandas.DataFrame: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html
- pandas.DataFrame.merge: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.merge.html
- pandas.DataFrame.drop: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.drop.html
- pandas.DataFrame.rename: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.rename.html

Task 2.4

- urllib.request.Request: https://docs.python.org/3/library/urllib.request.html
- urllib.request.urlopen.read: https://docs.python.org/3/library/urllib.request.html
- bs4.BeautifulSoup: https://www.crummy.com/software/BeautifulSoup/bs4/doc/
- pandas.read_html: https://pandas.pydata.org/docs/reference/api/pandas.read_html.html
- pandas.DataFrame.info: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.info.html
- pandas.DataFrame.columns: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.columns.html
- pandas.Series.str.split: https://pandas.pydata.org/docs/reference/api/pandas.Series.str.split.html
- pandas.DataFrame.loc: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.loc.html
- pandas.Series.str.title: https://pandas.pydata.org/docs/reference/api/pandas.Series.str.title.html

- pandas.DataFrame.merge: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.merge.html
- pandas.DataFrame.isna: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.isna.html
- len: https://www.w3schools.com/python/ref_func_len.asp
- pandas.DataFrame.duplicated: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.duplicated.html
- pandas.DataFrame.drop_duplicates: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.drop_duplicates.html
- numpy.where: https://numpy.org/doc/stable/reference/generated/numpy.where.html
- pandas.DataFrame.drop: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.drop.html