Cleaning Tables

Cleaned user_cards_raw

- Created table user_cards_staging1.
- Checked for duplicates (non-found)
- Converted card_index and card_user to integer.
- Separated card_expires to 2 new columns card_expiry_month and card_expiry_year.
- Converted card_cvv to integer.
- Added new column card_has_chip_bool which converts card_has_chip to Boolean.
- Converted cards_issued to integer.
- Extracted numbers only from credit_limit.
- Separated acct_open_date to 2 new columns acc_open_month and acc_open_year.
- Converted year_pin_last_changed to integer.
- Added new column card_on_dark_web_bool which converts card_on_dark_web to Boolean.
- Dropped columns card_expires, card_has_chip, acct_open_date, card_on_dark_web.
- Renamed column card_has_chip_bool to card_has_chip.
- Renamed column card_on_dark_web_bool to card_on_dark_web.
- Renamed table user_cards_clean.

Cleaned user_info_raw

- Created table user_info_staging1.
- Checked for duplicates (non-found)
- Converted user_age, retirement_age, birth_year, birth_month to integer.
- Replaced blanks with nulls in apartment.
- Converted apartments and zipcode to integer.
- Converted latitude, longitude to numeric.
- Extracted digits only from per_capita_income_zipcode, yearly_income_person, total_debt.
- Converted per_capita_income_zipcode, yearly_income_person, total_debt, fico_score, “Num Credit Cards” to integer.
- Renamed “Num Credit Cards” to num_credit_cards.
- Renamed table to user_info_clean.

Cleaned credit_card_transactions_raw

- Created table transactions_staging1 and filled with credit_card_transactions_raw.
- Converted trans_user, trans_card, trans_year,trans_month,trans_day to integer.
- Converted trans_time to time.
- Created table transactions_staging2 and filled with transactions_staging1.
- Extracted digits and “.” , “-“ from amount.
- Converted amount to numeric.
- Converted merchant_name to bigint.
- Converted zip to numeric.
- Converted mcc to integer.
- Checked and found duplicates, however decided not to remove since there is possibility that a transaction was made by the same person within the same minute at same location, since we do not include seconds in trans_time we cannot verify.
- Renamed transactions_staging2 to transactions_clean.

Understanding Tables Relationship

- Verified total row numbers for user_info_clean matched max(transactions_clean.trans_user) and max(user_cards_clean.card_user)
- Original user_info_clean did not have row index so added in myself.
- Evaluated whether each row in user_info_clean represents a unique user referenced by transactions_clean.trans_user and user_cards_clean.card_user.

```
select t1.card_user, count(t1.card_index), t2.num_credit_cards
from user_cards_clean t1 join user_info_clean t2 on t1.card_user = t2.user_index
group by t1.card_user, t2.num_credit_cards
having count(t1.card_index) <> t2.num_credit_cards
order by t1.card_user asc;
```

- This assumption could not be validated, as code up above was not empty table.
- Therefore, user_info_clean is treated as independent user metadata rather than a strict parent table.
- Evaluated whether each distinct transactions_clean.trans_user represents each user_cards_clean.card_user.

```
select t1.trans_user, max(t1.trans_card), max(t2.card_index)
from transactions_clean t1 join user_cards_clean t2 on t1.trans_user = t2.card_user
group by t1.trans_user
having max(t1.trans_card) <> max(t2.card_index)
order by t1.trans_user;
```

- The assumption is accepted since code above return 2 rows most likely case was card was never used since `max(t1.trans_card) was less then max(t2.card_index)`
- It is reasonable to assume that each distinct transactions_clean.trans_user corresponds to user_cards_clean.card_user, though the relationship is not formally enforced.