"a short description of the decisions you made and cleaning steps you took."

Decisions we've made

We decided to break the original table called `listings` on the website into tables containing information about the `host(hostInfo, hostResponses)` and tables containing information about the `listings(listingInfo, NeighbourhoodInfo, ListingPrice, ListingPolicy)`, because we know that one host can have multiple listings, then information about the host like `host_name`, `host_about` and many others would be redundant information.

We also broke information about the `host` and information about host responses to customers into two tables (`hostInfo` and `hostResponses`), because we know the some hosts may not have responses information. We also broke information about the listings into multiple tables because some listings don't have certain information. Table `Score` and table `Review` contains scores and review of a particular listing, and again, not all listings contain such information.

When importing the data, we decided to populate tables in sql using the postrgreSQL `\copy` command because some of our tables have more than 3000 rows, and it wouldn't be feasible to use the insert into command.

For the table '`ListingPolicy`':
We did not notice that '`instant_bookable`' was of '`t`' or '`f`' inputs. So we changed its type from '`integer`' to '`boolean`' in schema.

For the table '`ListingPrice`':
We changed the type of `price`, `weekly_price`... from '`float`' to '`varchar`' and removed the `check` statements, as we noticed that these prices are of form '$1489.20' with a leading '$' character. We create an extra view (using a query) to hold the converted data from 'varchar' to 'numeric'.

Cleaning steps

We know that our data won't perfectly follow our original schema when we first import it, because it may violate the primary constraint or the `not null` constraint. So we need to clean the data to make it fit into the schema. We first created schemas without these constraints and then import the data. Then we use sql commands to remove rows that would violate the `not null` constraints, and also remove duplicate rows so that the primary key constraint wouldn't be violated. We also deleted the rows that would violate the foreign key constraint. Then we exported all the tables with cleaned data into csv files, and then import these tables into the original schema.