

# Week 6: Data Cleaning and Window Functions in SQL

Parch and Posey Database

---

## Data Cleaning

### LEFT and RIGHT

1. In the `accounts` table, there is a column holding the website for each company. The last three digits specify what type of web address they are using. Pull these extensions and provide how many of each website type exist in the accounts table.
2. There is much debate about how much the name (or even the first letter of a company name) matters. Use the accounts table to pull the first letter of each company name to see the distribution of company names that begin with each letter (or number).
3. What is the number of company names that start with a vowel and consonant letters?

### POSITION

4. Use the `accounts` table to create first and last name columns that hold the first and last names for the `primary_poc`.

### CONCAT

5. Each company in the `accounts` table wants to create an email address for each `primary_poc`. The email address should be the first name of the `primary_poc` last name `primary_poc @ company name .com`. (e.g. `tamara.tuma@walmart.com`)
6. You may have noticed that in the previous solution some of the company names include spaces, which will certainly not work in an email address. See if you can create an email address that will work by removing all of the spaces in the account name, but otherwise your solution should be just as in the previous question.
7. We would also like to create an initial password, which they will change after their first log in. The password will be a combination of:
  - the first letter of the `primary_poc`'s first name (lowercase),
  - the last letter of their first name (lowercase),
  - the first letter of their last name (uppercase),
  - the last letter of their last name (uppercase),
  - the number of letters in their first name,
  - the number of letters in their last name, and
  - the name of the company they are working with, no spaces
  - the forth and fifth digit of their sales rep id

## Window Functions

8. For the orders table, create a new column which shows the total number of transactions for all accounts.
9. Update the previous query to create two new column: (1) `over_all_total_by_account_id`, and (2) `overall_count_by_account_id` (without using Group By).
10. Create a running total of `standard_amt_usd` (in the orders table) over order time.
11. Create a running total of `standard_amt_usd` (in the orders table) over order time for each month.
12. **YOUR TURN** - Create a running total of `standard_qty` (in the orders table) over order time for each year.

## Ranking data: `RAW_NUMBER()` and `RANK()`, `DENSE_RANK()`

13. For account with id 1001, use the `row_number()`, `rank()` and `dense_rank()` to rank the transactions by the number of standard paper purchased.
14. For each account, use the `row_number()`, `rank()` and `dense_rank()` to rank the transactions by the number of standard paper purchased.
15. **Your Turn** Select the id, account\_id, and standard\_qty variable from the orders table, then create a column called dense\_rank that ranks this standard\_qty amount of paper for each account. In addition, create a sum\_std\_qty which gives you the running total for account. Repeat the last task to get the avg, min, and max.
16. Give an alias for the window function in the previous question, and call it `account_window`.

## NTILES

17. Use the NTILE functionality to divide the accounts into 4 levels in terms of the amount of standard\_qty for their orders. Your resulting table should have the account\_id, the occurred\_at time for each order, the total amount of standard\_qty paper purchased, and one of four levels in a standard\_quartile column.
18. **YOUR TURN** Use the NTILE functionality to divide the accounts into two levels in terms of the amount of gloss\_qty for their orders. Your resulting table should have the account\_id, the occurred\_at time for each order, the total amount of gloss\_qty paper purchased, and one of two levels in a gloss\_half column.