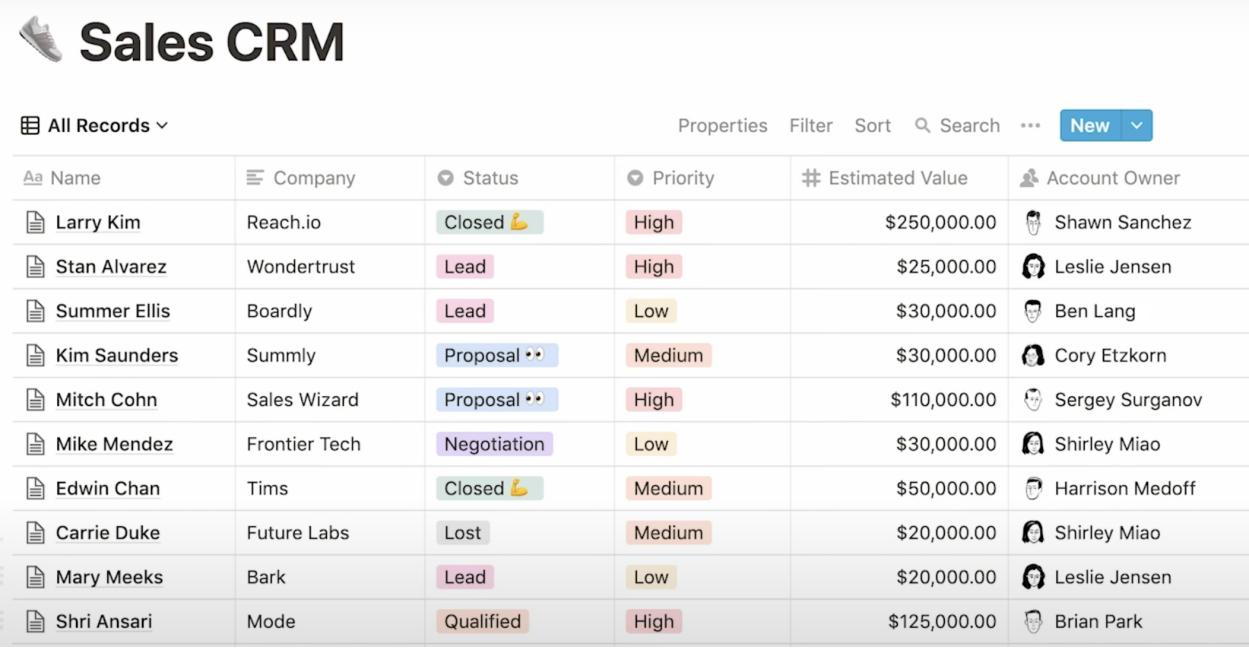
# Notion Table View and Filters

In this assignment, you will implement some of the read-only functionality of the Notion table view UI and even extend the filter operator functionality. Consider how you would exhibit clean code principles including reusability and extensibility. The UI doesn't need to be beautiful but it should the UX should be intuitive enough in terms of usability.




| Name | Company | Status | Priority | Estimated Value | Account Owner |
|---|---|---|---|---|---|
| Larry Kim | Reach.io | Closed 💪 | High | $250,000.00 | Shawn Sanchez |
| Stan Alvarez | Wondertrust | Lead | High | $25,000.00 | Leslie Jensen |
| Summer Ellis | Boardly | Lead | Low | $30,000.00 | Ben Lang |
| Kim Saunders | Summly | Proposal 👀 | Medium | $30,000.00 | Cory Etzkorn |
| Mitch Cohn | Sales Wizard | Proposal 👀 | High | $110,000.00 | Sergey Surganov |
| Mike Mendez | Frontier Tech | Negotiation | Low | $30,000.00 | Shirley Miao |
| Edwin Chan | Tims | Closed 💪 | Medium | $50,000.00 | Harrison Medoff |
| Carrie Duke | Future Labs | Lost | Medium | $20,000.00 | Shirley Miao |
| Mary Meeks | Bark | Lead | Low | $20,000.00 | Leslie Jensen |
| Shri Ansari | Mode | Qualified | High | $125,000.00 | Brian Park |

## Getting Started

You can refer to this blog on creating a simple app integrated with Notion. You may follow the code here to set up your own basic server.

## Submission

Please dockerize your application for ease of start up and testing. Push your code to Github and share the link as your submission. Feel free to share a README file with anything you'd like us to take note of from your submission.

## Requirements

1. Build a table view UI for Notion databases
    a. Implement a basic table view given a Notion database as input

b. Support sorting (see Sort doc)

c. Support rearrangement and resizing of columns - expected behavior:

   i. Click and hold the column headings to drag them left or right

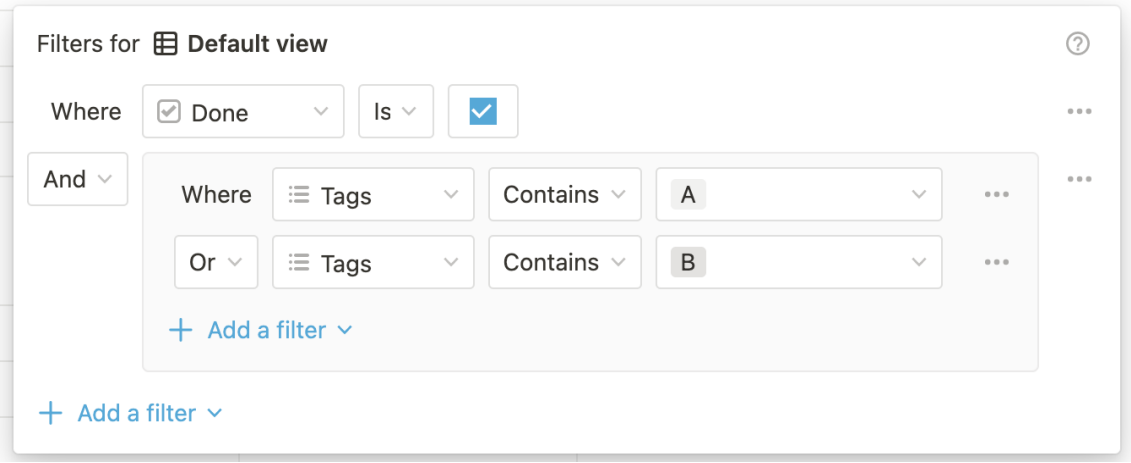
   ii. Resize columns by hovering over their edges, and dragging right or left

2. Build a Notion filter UI for supporting database filters (see Filters doc)

   a. Support the property types `checkbox`, `date`, `multi_select`, `number`, `rich_text`, `select`, `timestamp`, `status`

   b. Support Compound filters with filter groups (see Compound filter conditions doc)

      i. The Notion API doc notes that it only supports two levels of nesting on compound filter conditions. Implement the filters such that the restriction on the levels of nesting is configurable e.g. could be increased to 3, 4, or more




   c. Implement unit tests for the Compound filters

## Stretch Goals

1. Implement the NOT operator for compound filter conditions. Support compound filter conditions that contain only filter operators where the Notion API offers the logical negation e.g. !(`is_empty`) is `is_not_empty`, !(`less_than`) is `greater_than_or_equal_to`

   a. For the filter conditions where Notion does not offer the logical negation, implement validation logic that prompts the user that the NOT operator is

unsupported with the given compound filter conditions

b. For example: !((`datePropertyX` is after “2023-01-01” AND `textPropertyY` ends with “.com”) OR `textPropertyZ` starts with “www.”) should indicate “Unsupported conditions for NOT: `ends with`, `starts with`”

c. Include unit test cases for the NOT operator logic