

Implement View as description below:

- 1) On left there should be list of tables and on the right there should be grid area

The screenshot shows a user interface with a sidebar on the left and a main content area on the right. The sidebar contains a search bar labeled "Filter by Table Name" with a magnifying glass icon. Below the search bar is a tree view of tables under a category "imp". The tree view includes "employee_salary", "employee", "employee", and "patients". Under "Employee", there are "test", "tag1", "test2", "test3", "test4", "test5", "test6", and "HHHHHH". Under "Aldar", there is "Aldar". At the bottom of the sidebar is a button "Browse Workbooks" with a right-pointing arrow. The main content area is a large, empty grid with a light gray background and a vertical grid line running down its center. In the top right corner of the grid area, there are two small icons: a "X" and a "refresh" symbol.

- 2) User should be able to drag any table from list and drop to grid. As user drops the table, table should be created to the available space on the grid.

Eg. Dropped one table

The screenshot shows the same user interface as the previous one, but with a table "employee_salary" now present in the grid area. The sidebar remains the same, showing the tree view of tables. The main content area now displays the "employee_salary" table details. A modal window titled "employee_salary" is open, showing four columns: "age" (Data Type: INTEGER), "emp_id" (Data Type: INTEGER), and "experience" (Data Type: INTEGER). There is also a "Column" header with a checked checkbox. At the bottom of the modal, there is a link "Scroll to see more columns" with a double slash icon. The "X" icon in the top right corner of the modal can be used to close it.

Eg. Dropped other table

The screenshot shows a data management interface with a sidebar on the left containing a tree view of datasets and a search bar. The main area displays two tables side-by-side:

- employee_salary** table:
 - Column: age, Data Type: INTEGER
 - Column: emp_id, Data Type: INTEGER
 - Column: experience, Data Type: INTEGER
- employee** table:
 - Column: department_id, Data Type: int
 - Column: email, Data Type: string
 - Column: employee_id, Data Type: int

Both tables have a "Scroll to see more columns" message at the bottom.

3) User should be able to change position of table by dragging the table

The screenshot shows the same data management interface, but the **employee_salary** table has been moved to the top position. The **employee** table remains in its original position below it.

Eg: Drop table. (Patients table added to the available space after employee got dragged)

The screenshot shows a database management interface with a sidebar for filtering table names. Three tables are displayed in separate windows:

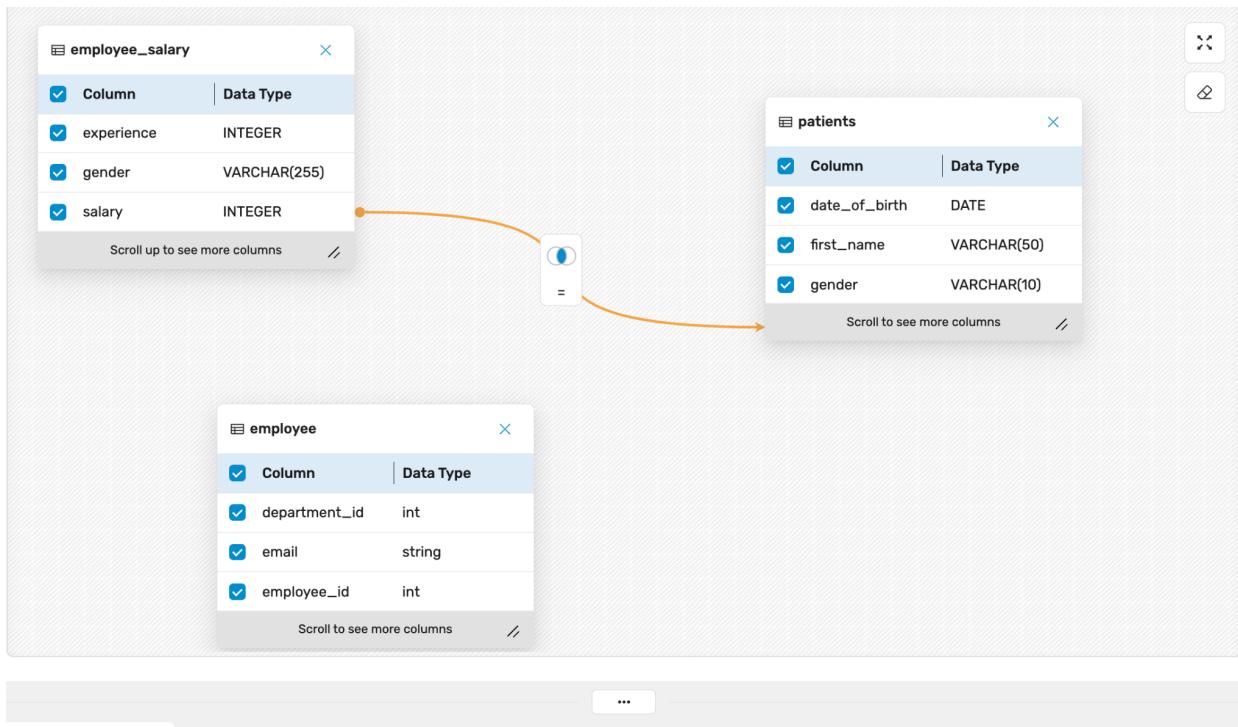
- employee_salary** (Columns: age, emp_id, experience, Data Type: INTEGER)
- patients** (Columns: date_of_birth, first_name, gender, Data Type: DATE, VARCHAR(50), VARCHAR(10))
- employee** (Columns: department_id, email, employee_id, Data Type: int, string, int)

The **employee** table is highlighted with a blue border and has a hand cursor icon over it, indicating it is being dragged. A message "Scroll to see more columns" is visible at the bottom of each table window.

- 4) If Dropped table already exist to the grid either Highlight the already existing Table or show alert.
- 5) Table can be resized
- 6) User Should be able to drag one column of a table to the other column of other table. When do so there should be a connection line created

The screenshot shows the same database interface as above, but with a specific interaction. The **experience** column from the **employee_salary** table is being dragged to the **age** column in the **patients** table. A yellow connection line is drawn from the source column to the target column. A small circular icon with a plus sign is placed near the connection point, indicating a successful connection or a proposed action.

7) Connection line should be accommodated correctly on scroll of the table



8) User should be able to remove table from Grid, as he remove Table all connection lines linked to that table will be removed as well

Note:

- 1) Any action visible in design which is not described above can be ignored
- 2) On Left hand side listing no need of hierarchy, Simply tables can be listed.

Mock data structure tables:

```
tables: [
  {
    id: 'unique_table_id',
    name: 'table_name',
    columns: [
      {
        column_id: 'unique_id',
```

```
        name: 'column_name',
        column_data_type: 'data_type'
    },
    {
        column_id: 'unique_id',
        name: 'column_name',
        column_data_type: 'data_type'
    }
]

},
{
    id: 'unique_table_id',
    name: 'table_name',
    columns: [
        {
            column_id: 'unique_id',
            name: 'column_name',
            column_data_type: 'data_type'
        },
        {
            column_id: 'unique_id',
            name: 'column_name',
            column_data_type: 'data_type'
        }
    ]
}
]
```