**Data Cleaning With OpenRefine :**

All the 4 dirty datasets were loaded into the OpenRefine and cleaning operation was performed. The cleaning operation is captured in the json file that is generated by OpenRefine.

1. **Menu.csv**

**Number of records – 17, 547**

* 1. Please note that 90+ steps were performed on the dataset and it was the most dirty dataset of all. Some steps were repeated and data set was cleaned. So, it was disffilcult to capture all the numbers in the process.
  2. Generic : All the columns where numeric fields were present were converted to number type. Date fields were converted to date type and text types to text.
     1. ***name*** :
        1. Removed leading and trailing spaces.
        2. Collapsed consecutive white spaces.
        3. Converted the values to titlecase.
        4. Clustering was used to group different data values and appropriate action were taken on the values. All these below methods were used to cluster the data.
           1. Method : key collision

fingerprint

ngram-fingerprint

metaphone-3

cologne-phonetic

Daiktch-Mokotoff

Beider-Morse

* + - * 1. Nearest neighbor

levenshtein

ppm

* + - 1. Used GREL to remove values like “ , ( ) { } \* [ ] ‘. ? ;
      2. All these above steps were repeated multiple times as each clustering and and cleaning open up an opportunity to apply the transaformation again.
    1. ***sponsor*** :
       1. Removed leading and trailing spaces.
       2. Collapsed consecutive white spaces.
       3. Converted the values to titlecase.
       4. Clustering was used to group different data values and appropriate action were taken on the values. All these below methods were used to cluster the data.
          1. Method : key collision

fingerprint

ngram-fingerprint

metaphone-3

cologne-phonetic

Daiktch-Mokotoff

Beider-Morse

* + - * 1. Nearest neighbor

levenshtein

ppm

* + - 1. Used GREL to remove values like “ , ( ) { } \* [ ] ‘. ? \ \ ;
      2. All these above steps were repeated multiple times as each clustering and and cleaning open up an opportunity to apply the transaformation again.
    1. ***occasion*** :
       1. Removed leading and trailing spaces.
       2. Collapsed consecutive white spaces.
       3. Converted the values to titlecase.
       4. Used GREL to remove values like “ , ( ) { } \* [ ] ‘. ? \ \ ;
       5. Clustering was used to group different data values and appropriate action were taken on the values. All these below methods were used to cluster the data.
          1. Method : key collision

fingerprint

ngram-fingerprint

metaphone-3

cologne-phonetic

Daiktch-Mokotoff

Beider-Morse

* + - * 1. Nearest neighbor

levenshtein

ppm

* + - 1. The opportuinity to clean the data based on the clustering was very helpful. Data like ‘Anniv’, "ANNIVERSARYERSARY;" etc. were clubbed together as ‘Anniversary’. This was one of the example from the list of values those were cleaned.
      2. Used GREL to remove values like “ , ( ) { } \* [ ] ‘. ? \ \
      3. All these above steps were repeated multiple times as each clustering and and cleaning open up an opportunity to apply the transformation again.
    1. ***event, venue*** :
       1. Removed leading and trailing spaces.
       2. Collapsed consecutive white spaces.
       3. Converted the values to titlecase.
       4. Used GREL to remove values like “ , ( ) { } \* [ ] ‘. ? \ \
       5. Clustering was used to group different data values and appropriate action were taken on the values. All these below methods were used to cluster the data.
          1. Method : key collision

fingerprint

ngram-fingerprint

metaphone-3

cologne-phonetic

Daiktch-Mokotoff

Beider-Morse

* + - * 1. Nearest neighbor

levenshtein

ppm

* + - 1. Used GREL to remove values like “ , ( ) { } \* [ ] ‘. ? \ \
      2. All these above steps were repeated multiple times as each clustering and and cleaning open up an opportunity to apply the transformation again.
    1. ***place***
       1. Removed leading and trailing spaces.
       2. Collapsed consecutive white spaces.
       3. Used GREL to remove values like “ , ( ) { } \* [ ] ‘. ? \ \
       4. Clustering was used to group different data values and appropriate action were taken on the values. All these below methods were used to cluster the data.
          1. Method : key collision

fingerprint

ngram-fingerprint

metaphone-3

cologne-phonetic

Daiktch-Mokotoff

Beider-Morse

* + - * 1. Nearest neighbor

levenshtein

ppm

* + - 1. Used GREL to remove values like “ , ( ) { } \* [ ] ‘. ? \ \
      2. All these above steps were repeated multiple times as each clustering and and cleaning open up an opportunity to apply the transformation again.
    1. ***physical\_descrription***
       1. Removed leading and trailing spaces.
       2. Collapsed consecutive white spaces.
       3. Used GREL to remove values like “ , ( ) { } \* [ ] ‘. ? \ \
       4. Clustering was used to group different data values and appropriate action were taken on the values. All these below methods were used to cluster the data.
       5. The opportunity to clean the data was minimal.
       6. The cleaning was not necessary and could have been left as it is.
    2. ***occasion***
       1. Removed leading and trailing spaces.
       2. Collapsed consecutive white spaces.
       3. Used GREL to remove values like “ , ( ) { } \* [ ] ‘. ? \ \
       4. Clustering was used to group different data values and appropriate action were taken on the values. All these below methods were used to cluster the data.
          1. Method : key collision

fingerprint

ngram-fingerprint

metaphone-3

cologne-phonetic

Daiktch-Mokotoff

Beider-Morse

* + - * 1. Nearest neighbor

levenshtein

ppm

* + - 1. Used GREL to remove values like “ , ( ) { } \* [ ] ‘. ? \ \
      2. All these above steps were repeated multiple times as each clustering and and cleaning open up an opportunity to apply the transformation again
    1. ***notes***
       1. Removed leading and trailing spaces.
       2. Collapsed consecutive white spaces.
       3. Used GREL to remove values like “ , ( ) { } \* [ ] ‘. ? \ \
       4. Few clusters were available for cleaning considering the nature of data avalable.
       5. Opportunity was less.
    2. ***call\_number***
       1. Few clusters were available for cleaning considering the nature of data avalable.
       2. Opportunity was less.
    3. ***keywords, language, location, location\_type***
       1. The columns were removed because of lack of data in them.
    4. ***date***
       1. Converted the column to Date type.
       2. Formatted the data to ISO standard - toString(toDate(value),"yyyy-MM-dd")
    5. ***currency***
       1. Removed leading and trailing spaces.
       2. Collapsed consecutive white spaces.
       3. Converted the values to titlecase.
    6. ***status***
       1. Removed leading and trailing spaces.
       2. Collapsed consecutive white spaces.
       3. Converted the values to titlecase.
    7. ***page\_count, dish\_count***
       1. No scope for transformation other than changing the column type to number type.
    8. ***currency\_symbol***
       1. No Cleaning.

1. **Dish.csv**

**Number of records – 428,082**

* 1. Please note that because of the size of the dataset, the OpenRefine tool was not able to apply any Facets. Because of this, the mess in the data was not possible to identify.
  2. With careful observation, many dirty data were cleaned.
  3. Generic : All the columns where numeric fields were present were converted to number type. Date fields were converted to date type and text types to text.
     1. ***name*** :
        1. Removed leading and trailing spaces. – Rows Affected - 6582
        2. Collapsed consecutive white spaces. - Rows Affected - 9288
        3. Converted the values to titlecase.
        4. Clustering was not possible.
        5. Used GREL to remove values like “ , ( ) { } \* [ ] ‘. ? -
           1. Rows Affected – 7009, 2568, 5 1801, 18, 309, 33015 (different types of characters)
     2. ***id, menus\_appeared, times\_appeared, lowest\_price, highest\_price :***
        1. Removed leading and trailing spaces.
        2. Collapsed consecutive white spaces.
        3. Converted the values to numbers.
     3. ***first\_appeared, last\_appeared***
        1. Date transformations.

1. **MenuItem.csv**

**Number of records – 1,334,779**

* 1. Please note that because of the size of the dataset 118MB, the OpenRefine tool was not able to apply any Facets.
  2. Data was mostly clean.
  3. Some generic transformations applied.
  4. Text transform on 1334779 cells in column menu\_page\_id: value.toNumber()
  5. Text transform on 888520 cells in column price: value.toNumber()
  6. Text transform on 91979 cells in column high\_price: value.toNumber()
  7. Text transform on 1334538 cells in column dish\_id: value.toNumber()
  8. Text transform on 1334779 cells in column xpos: value.toNumber()
  9. Text transform on 1334779 cells in column ypos: value.toNumber()

1. **MenuPage.csv**

**Number of records – 66,937**

* 1. Data was mostly clean.
  2. Few generic transfortmation was applied.
  3. Text transform on 66937 cells in column id: value.toNumber()
  4. Text transform on 66937 cells in column menu\_id: value.toNumber()
  5. Text transform on 65735 cells in column page\_number: value.toNumber()
  6. Text transform on 66914 cells in column image\_id: value.toNumber()
  7. Text transform on 66608 cells in column full\_height: value.toNumber()
  8. Text transform on 66608 cells in column full\_width: value.toNumber()
  9. Text transform on 1 cells in column uuid: value.toLowercase()

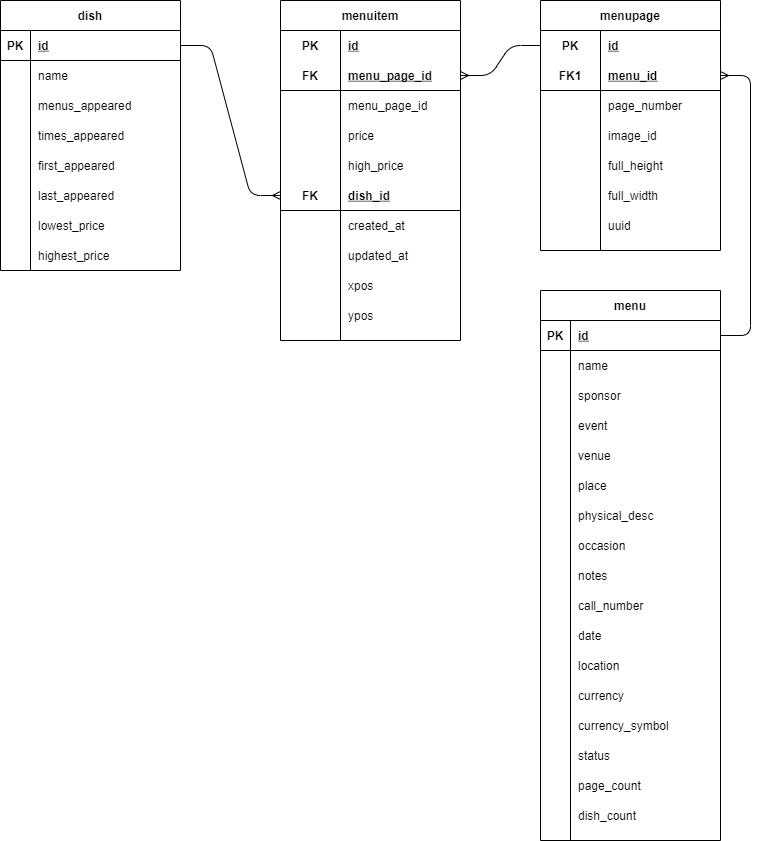
**Data Quality Improvements :**

|  |  |  |
| --- | --- | --- |
| **File** | **Column** | **Quality Change Measure** |
| Menu.csv | id | Text transform on 676 cells in column sponsor: grel:value.replace(/\?/,'').replace(/\(|\)/,'').replace(/\"/,'') |
|  |  | Text transform on 474 cells in column sponsor: grel:value.replace(/\\|\//,'').replace(/\"/,'').replace(/\[|\]/,'') |
|  |  | Text transform on 8275 cells in column sponsor: value.toTitlecase() |
|  |  | Clustering :  key collision -  fingerprint  Mass edit 3937 cells in column sponsor  ngram-fingerprint  Mass edit 1787 cells in column sponsor  metaphone-3  Mass edit 524 cells in column sponsor Undo  cologne-phonetic  Mass edit 1676 cells in column sponsor  Daitch-Mokotof  Mass edit 163 cells in column sponsor Undo  Beider-Morse  Mass edit 23 cells in column sponsor Undo  nearest-neighbor -  lavenshtein  Mass edit 1617 cells in column sponsor Undo  ppm  0 |
|  | name | Text transform on 28 cells in column name: grel:value.replace(/\?/,'').replace(/\(|\)/,'').replace(/\"/,'')  Text transform on 139 cells in column name: grel:value.replace(/\\|\//,'').replace(/\"/,'').replace(/\[|\]/,'').replace(/\\*/,'') |
|  |  | Clustering :  key collision -  fingerprint  Mass edit 556 cells in column sponsor  ngram-fingerprint  Mass edit 622 cells in column sponsor  metaphone-3  Mass edit 762 cells in column sponsor Undo  cologne-phonetic  Mass edit 4 cells in column sponsor  Daitch-Mokotof  Mass edit 12 cells in column sponsor Undo  Beider-Morse  Mass edit 0 cells in column sponsor Undo  nearest-neighbor -  lavenshtein  Mass edit 7 cells in column sponsor Undo  ppm  60 |
|  | event | Text transform on 9393 cells in column event: value.toString() |
|  |  | Text transform on 99 cells in column event: grel:value.replace(/\?/,'').replace(/\(|\)/,'').replace(/\"/,'') |
|  |  | Text transform on 90 cells in column event: grel:value.replace(/\\|\//,'').replace(/\"/,'').replace(/\[|\]/,'') |
|  |  | Text transform on 7826 cells in column event: value.toTitlecase() |
|  |  | Clustering :  key collision -  fingerprint  Mass edit 4027cells in column sponsor  ngram-fingerprint  Mass edit 2325 cells in column sponsor  metaphone-3  Mass edit 3057 cells in column sponsor Undo  cologne-phonetic  Mass edit 2297 cells in column sponsor  Daitch-Mokotof  Mass edit 98 cells in column sponsor Undo  Beider-Morse  Mass edit 2 cells in column sponsor Undo  nearest-neighbor -  lavenshtein  Mass edit 407 cells in column sponsor Undo  ppm  0 |
|  | venue | Text transform on 9449 cells in column venue: value.toString() |
|  |  | Text transform on 8096 cells in column venue: value.toTitlecase |
|  |  | Clustering:  fingerprint  Mass edit 2126 cells in column sponsor  ngram-fingerprint  Mass edit 21 cells in column sponsor  metaphone-3  Mass edit 5044 cells in column sponsor Undo  cologne-phonetic  Mass edit 0 cells in column sponsor  Daitch-Mokotof  Mass edit 505 cells in column sponsor Undo  Beider-Morse  Mass edit 0 cells in column sponsor Undo  nearest-neighbor -  lavenshtein  Mass edit 0 cells in column sponsor Undo  ppm  Mass edit 7 cells in column sponsor Undo |
|  | physical\_description | Text transform on 2777 cells in column physical\_description: value.toString() |
|  |  | Text transform on 38 cells in column physical\_description: value.replace(/\s+/,' ') |
|  |  | Clustering:  fingerprint  Mass edit 1514 cells in column sponsor  ngram-fingerprint  Mass edit 4575 cells in column sponsor  metaphone-3  Mass edit 0 cells in column sponsor Undo  cologne-phonetic  Mass edit 2 cells in column sponsor  Daitch-Mokotof  Mass edit 0 cells in column sponsor Undo  Beider-Morse  Mass edit 0 cells in column sponsor Undo  nearest-neighbor -  lavenshtein  Mass edit 0 cells in column sponsor Undo  ppm  Mass edit 0 cells in column sponsor Undo |
|  | Occasion | Text transform on 13744 cells in column occasion: value.toString() |
|  |  | Text transform on 508 cells in column sponsor: grel:value.replace(/\?/,'').replace(/\(|\)/,'').replace(/\"/,'') |
|  |  | Text transform on 474 cells in column sponsor: grel:value.replace(/\\|\//,'').replace(/\"/,'').replace(/\[|\]/,'') |
|  |  | Text transform on 2954 cells in column occasion: grel:value.replace(/\\|\//,'').replace(/\"/,'').replace(/\[|\]/,'').replace(/;/,'').replace(/\\*/,'') |
|  |  | Clustering ;  fingerprint  Mass edit 969 cells in column sponsor  ngram-fingerprint  Mass edit 272 cells in column sponsor  metaphone-3  Mass edit 1393 cells in column sponsor  cologne-phonetic  Mass edit 693 cells in column sponsor  Daitch-Mokotof  Mass edit 215 cells in column sponsor  Beider-Morse  Mass edit 0 cells in column sponsor  nearest-neighbor -  lavenshtein  Mass edit 281 cells in column sponsor  ppm  Mass edit 716 cells in column sponsor  Text transform on 3752 cells in column occasion: value.toTitlecase() |
|  | notes | Text transform on 195 cells in column notes: value.replace(/\s+/,' ')  Text transform on 9460 cells in column notes: value.toTitlecase() |
|  |  | Clustering:  fingerprint  Mass edit 1355 cells in column sponsor  ngram-fingerprint  Mass edit 320 cells in column sponsor  metaphone-3  Mass edit 0 cells in column sponsor  cologne-phonetic  Mass edit 268 cells in column sponsor  Daitch-Mokotof  Mass edit 0 cells in column sponsor  Beider-Morse  Mass edit 0 cells in column sponsor  nearest-neighbor -  lavenshtein  Mass edit 0 cells in column sponsor  ppm  Mass edit 0 cells in column sponsor |
|  |  | Text transform on 851 cells in column sponsor: grel:value.replace(/\?/,'').replace(/\(|\)/,'').replace(/\"/,'') |
|  |  | Text transform on 267 cells in column notes: grel:value.replace(/\\|\//,'').replace(/\"/,'').replace(/\[|\]/,'') |
|  |  | Text transform on 6034 cells in column notes: grel:value.replace(/\;$/,'') |
|  | call\_number | Clustering :  Mass edit 0 cells in column sponsor  ngram-fingerprint  Mass edit 0 cells in column sponsor  metaphone-3  Mass edit 0 cells in column sponsor  cologne-phonetic  Mass edit 0 cells in column sponsor  Daitch-Mokotof  Mass edit 0 cells in column sponsor  Beider-Morse  Mass edit 0 cells in column sponsor  nearest-neighbor -  lavenshtein  Mass edit 0 cells in column sponsor  ppm  Mass edit 0 cells in column sponsor |
|  | date | Text transform on 16958 cells in column date: value.toDate()  Text transform on 16958 cells in column date: grel:toString(toDate(value),"yyyy-MM-dd") |
|  | status | Text transform on 17547 cells in column status: value.toTitlecase() |
|  | page\_count, dish\_count | Text transform on 17547 value.toNumber() |
| Menupage.csv | id | Text transform on 66937 cells in column id: value.toNumber() |
|  | menu\_id | Text transform on 66937 cells in column menu\_id: value.toNumber() |
|  | page\_number | Text transform on 65735 cells in column page\_number: value.toNumber() |
|  | image\_id | Text transform on 66914 cells in column image\_id: value.toNumber() |
|  | full\_height | Text transform on 66608 cells in column full\_height: value.toNumber() |
|  | full\_width | Text transform on 66608 cells in column full\_width: value.toNumber() |
|  | uuid | Text transform on 1 cells in column uuid: value.toLowercase() |
| Menuitem.csv | menu\_page\_id | Text transform on 1334779 cells in column menu\_page\_id: value.toNumber() |
|  | price | Text transform on 888520 cells in column price: value.toNumber() |
|  | high\_price | Text transform on 91979 cells in column high\_price: value.toNumber() |
|  | dish\_id | Text transform on 1334538 cells in column dish\_id: value.toNumber() |
|  | xpos | Text transform on 1334779 cells in column xpos: value.toNumber() |
|  | ypos | Text transform on 1334779 cells in column ypos: value.toNumber() |
| Dish.csv | name | Removed leading and trailing spaces. – Rows Affected - 6582  Collapsed consecutive white spaces. - Rows Affected - 9288  Converted the values to titlecase.  Clustering was not possible.  Used GREL to remove values like “ , ( ) { } \* [ ] ‘. ? -  Rows Affected – 7009, 2568, 5 1801, 18, 309, 33015 (different types of characters) |
|  | id, menus\_appeared, times\_appeared, lowest\_price, highest\_price | Removed leading and trailing spaces.  Collapsed consecutive white spaces.  Converted the values to numbers. |
|  | first\_appeared, last\_appeared | Date transformations. (mostly all records) |
| SQL |  | Quality improvements by SQL is given below. |

**Data Cleaning With SQL & Quality Improvements:**

Once most of the cleaning was completed using Pyhton and OpenRefine, SQLite Stuido was used to load the data into different tables and do additional data validation (Integrity Constraint Violation) and cleaning where possible. As most of the data were cleaned using OpenRefine, the scope for cleaning was miminlal. Below is the detail of the ER Diagrams, Integrity Constraint Violation Vaidation and few Cleaning steps perfomed using SQLite.

**ER Diagram :**



**Table Definitions/ Schema:**

**Table :** *menu*

*CREATE TABLE menu (*

*id BIGINT PRIMARY KEY*

*NOT NULL,*

*name TEXT,*

*sponsor TEXT,*

*event TEXT,*

*venue TEXT,*

*place TEXT,*

*physical\_desc TEXT,*

*occasion TEXT,*

*notes TEXT,*

*call\_number TEXT,*

*date DATE,*

*location TEXT,*

*currency TEXT,*

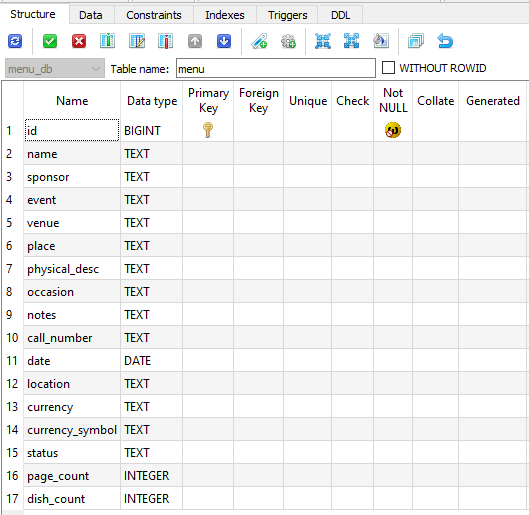
*currency\_symbol TEXT,*

*status TEXT,*

*page\_count INTEGER,*

*dish\_count INTEGER*

*);*



**Table :** *menuitem*

*CREATE TABLE menuitem (*

*id BIGINT PRIMARY KEY*

*NOT NULL,*

*menu\_page\_id BIGINT REFERENCES menupage (id),*

*price DOUBLE,*

*high\_price DOUBLE,*

*dish\_id BIGINT REFERENCES dish (id),*

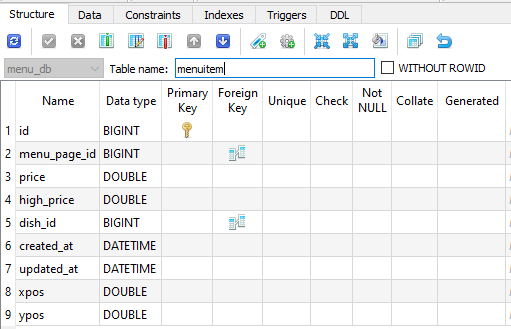
*created\_at DATETIME,*

*updated\_at DATETIME,*

*xpos DOUBLE,*

*ypos DOUBLE*

*);*



**Table :** *menupage*

*CREATE TABLE menupage (*

*id BIGINT PRIMARY KEY*

*NOT NULL,*

*menu\_id BIGINT REFERENCES menu (id),*

*page\_number INTEGER,*

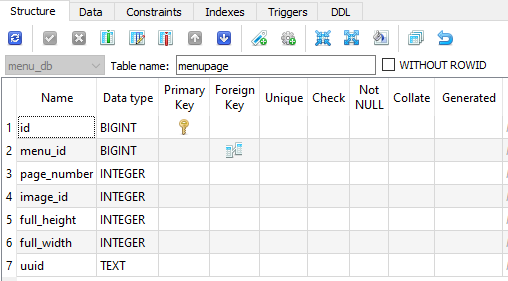
*image\_id INTEGER,*

*full\_height INTEGER,*

*full\_width INTEGER,*

*uuid TEXT*

*);*



**Table :** *dish*

*CREATE TABLE dish (*

*id BIGINT PRIMARY KEY*

*NOT NULL,*

*name TEXT,*

*menus\_appeared INTEGER,*

*times\_appeared INTEGER,*

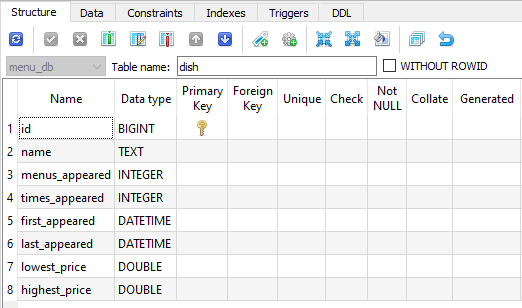
*first\_appeared DATETIME,*

*last\_appeared DATETIME,*

*lowest\_price DOUBLE,*

*highest\_price DOUBLE*

*);*



**Integrity Constraint Check :**

1. **Criteria:** id shouldn’t have dupliacte or NULL Values on menu table.
2. ***Table*** *:* ***menu***

***Query:***

*SELECT id, COUNT(id) AS cnt\_id*

*FROM menu*

*GROUP BY id*

*HAVING COUNT(id) > 1;*

***Resut*** *: 0 Records*

***Action*** *: No action.*

1. ***NULL Value Check:***

***Query:***

*SELECT \* FROM menu*

*WHERE id IS NULL;*

***Result:*** *0 such records.*

***Action*** *: No action.*

1. **Criteria:** id shouldn’t have dupliacte or NULL Values on menuitem table.
2. ***Table*** *:* ***menuitem***

***Query:***

*SELECT id, COUNT(id) AS cnt\_id*

*FROM menu*

*GROUP BY id*

*HAVING COUNT(id) > 1;*

***Resut*** *: 0 Records*

***Action*** *: No action.*

1. ***NULL Value Check:***

***Query:***

*SELECT \* FROM menuitem*

*WHERE id IS NULL;*

***Result:*** *0 such records.*

***Action*** *: No action.*

1. **Criteria:** id shouldn’t have dupliacte or NULL Values on dish table.
2. ***Table*** *:* ***menuitem***

***Query:***

*SELECT id, COUNT(id) AS cnt\_id*

*FROM dish*

*GROUP BY id*

*HAVING COUNT(id) > 1;*

***Resut*** *: 0 Records*

***Action*** *: No* ***action.***

1. ***NULL Value Check:***

***Query:***

*SELECT \* FROM dish*

*WHERE id IS NULL;*

***Result:*** *0 such records.*

***Action*** *: No action.*

1. **Criteria:** id shouldn’t have dupliacte or NULL Values on menupage table.
2. ***Table*** *:* ***menuitem***

***Query:***

*SELECT id, COUNT(id) AS cnt\_id*

*FROM menupage*

*GROUP BY id*

*HAVING COUNT(id) > 1;*

***Resut*** *: 0 Records*

***Action*** *: No action.*

1. ***NULL Value Check:***

***Query:***

*SELECT \* FROM menupage*

*WHERE id IS NULL;*

***Result:*** *0 such records.*

***Action*** *: No action.*

**Cleaning Using SQL:**

1. **Table** : ***menu***
   1. Identify and delete irrelevant records.

**Query:**

*SELECT COUNT(\*)*

*FROM menu*

*WHERE name = ‘’*

*AND sponsor= ‘’;*

**Result*:***1619

**Obervation :** 1680 records where both name and sponsor are blank don’t have most of its other columns not populated. So, all such recors were deleted since they don’t hold enough data for those records to be relevant.

**Action :** Delete such records.

**Query:**

*DELETE FROM menu*

*WHERE name = ‘’*

*AND sponsor= ‘’;*

**Result*:***1619 records were deleted.

* 1. Default columns where most of the data are missing so that they can be ised for categorizing.

**Query:**

*SELECT COUNT(\*)*

*FROM MENU*

*WHERE NAME='';*

**Result*:***12729

Action : Default such records with value as ‘Unavailable’.

**Query:**

*UPDATE menu*

*SET name = 'Unavialble'*

*WHERE name = '';*

Result*:* 12729 records were updated.

* 1. Default name and sponsor coumn where most of the data are like (Restaurant Andor Location Not Given./ Restaurant Name Andor Location Not Given/ Not Given) so that they can be used for categorizing

**Query:**

SELECT COUNT(\*) FROM menu

WHERE name LIKE '%NOT GIVEN%'

**Result***:* 138

**Action :** Default such records with value as ‘Unavailable’.

**Query:**

*UPDATE menu*

*SET name = 'Unavialble'*

*WHERE name LIKE '%NOT GIVEN%';*

**Result***:* 138 records were updated.

**Query:**

*SELECT COUNT(\*) FROM menu*

*WHERE sponsor LIKE '%NOT GIVEN%';*

Result*:* 216

**Query:**

*UPDATE menu*

*SET sponsor = 'Unavialble'*

*WHERE sponsor LIKE '%NOT GIVEN%';*

**Result*:***216 records were updated.

1. **Table** : ***dish***
   1. **Identify and delete irrelevant records.**

**Query**:

*SELECT count(\*) FROM dish*

*WHERE times\_appeared = '';*

**Result*:***0

**Query**:

*SELECT count(\*) FROM dish*

*WHERE menus\_appeared = '';*

**Result*:* 0**

Action : None. No such records were found.

* 1. **Obervation :** 1680 records where both name and sponsor are blank don’t have most of its other columns not populated. So, all such recors were deleted since they don’t hold enough data for those records to be relevant.

**Query***:*

*SELECT COUNT(\*) FROM menu*

*WHERE name = ‘’*

*AND sponsor= ‘’;*

**Result*:***1680 records found

**Action :** Delete such records.

**Query**:

*DELETE FROM menu*

*WHERE name = ‘’*

*AND sponsor= ‘’;*

**Result*:***1680 records were deleted.

* 1. Last appeared\_date had years beoyond 2021.

**Query:**

*SELECT COUT+NT(\*) FROM dish*

*WHERE last\_appeared > '2021';*

Result*:* 180

**Action :** Delete such records.

*DELETE FROM dish*

*WHERE last\_appeared > '2021';*

**Result*:***180 records were deleted.

* 1. Last appeared\_date had years beoyond 2021.

**Query:**

*SELECT COUT+NT(\*) FROM dish*

*WHERE last\_appeared > '2021';*

**Result*:***180

**Action :** Delete such records.

*DELETE FROM dish*

*WHERE last\_appeared > '2021';*

**Result*:***180 records were deleted.

* 1. highest\_price column contains a lot spaces. Update them to 0.

**Query :**

*SELECT \* FROM dish*

*WHERE highest\_price = ' '*

**Result:** 29,098

**Action :** Default the spaces to 0.

**Query :**

*UPDATE dish*

*SET highest\_price = 0*

*WHERE highest\_price =* '';

**Result:** 29,098 records updated.

* 1. Check the same for lowest\_price.

**Query :**

*SELECT \* FROM dish*

*WHERE lowest\_price = ' '*

**Result:** 0

**Action :** None.

**Query :**

*UPDATE dish*

*SET lowest\_price = 0*

*WHERE highest\_price = '';*

**Result:** 446,259 records updated.

1. **Table** : ***menuitem***
   1. high\_price column contains a lot spaces. Update them to 0.

**Query :**

SELECT \* FROM menuitem

WHERE high\_price = ' '

**Result:** 1,242,800

**Action :** Default the spaces to 0.

**Query :**

*UPDATE menuitem*

*SET high\_price = 0*

*WHERE high\_price =* '';

**Result:** 1,242,800 records updated.

* 1. price column contains a lot spaces. Update them to 0.

**Query :**

SELECT \* FROM menuitem

WHERE price = ' '

**Result:** 446,259

**Action :** Default the spaces to 0.

**Query :**

*UPDATE menuitem*

*SET price = 0*

*WHERE price =* '';

**Result:** 446,259 records updated.

1. **Table** : ***menupage***

No relevant data found to be cleaned as such.