**INTRODUCTION**

EATSOME in SF is a web application, which allows the restaurant business users to analyze how well the restaurant is doing, from there past inspections or violations. This analysis will give business user’s a chance to view the history of their examinations in order to help them in improvise with their business so that they could gain more profits and attract more consumers.

This application also helps the consumers get an idea of how hygienic the food is at particular restaurant in SF.

The Health Department has developed an inspection report and scoring system. After conducting an inspection of the facility, the Health Inspector calculates a score based on the violations observed. Violations can fall into

**High-risk category**: records specific violations that directly relate to the transmission of food borne illnesses, the adulteration of food products and the contamination of food-contact surfaces.

**Moderate risk category**: records specific violations that are of a moderate risk to the public health and safety.

**Low risk category**: records violations that are low risk or have no immediate risk to the public health and safety.

The scorecards issued by the inspector are maintained at the food establishment and are available to the public in this dataset.

Depending on the score and violations, they might conduct follow-up inspections and revise their score. Every restaurant in San Francisco City has a yearly, surprise inspection in order to remain in operation. At inspection, points are assigned for every violation, dependent on violation type and category. The number of points is summed to the restaurant’s score.

We analyzed Violation, Inspection and Score information in relation to Business over different parameters with details obtained from the health inspection data set.

**TECHNOLOGY STACK AND DATA SET**

To develop the above described web application we have used the following technologies

* HTML 5, CSS, JavaScript, JQUERY, AJAX, Bootstrap – front end
* PHP – backend
* MYSQL – data store
* Apache web server – deployment server
* Google chart API – charting library
* User invokes data operation through http request
* Web server serves the Web browser with dynamic and static requests through PHP code running on server

**DATA SET**

We got this data set in the following below link

<https://data.sfgov.org/Health-and-Social-Services/Restaurant-Scores/stya-26eb>

Data was pretty much well organized, but it had lots of null values as well as redundancies. We had to clean the data and performed normalization. All our data tables are in 3rd normal form.

**OVERVIEW OF YOUR DATAMODELS**

We have persisted our data in a relational data model. Below listed are the operational database tables.

* **Business** - this table describes the restaurant details
* **business\_violation** - this table describes violations associated with each business.
* **business\_violation\_details**- this table describes the risk category associated with each business\_violation.

**Business** 🡪 business\_owner, business\_phone

* Business owner is a master table. Each business owner can hold one or more businesses. So it’s many to many relationships between both of them.
* Each business owner can have more than one phone number, so it’s a multi-valued attribute. So it’s one to many relationships between business and business\_phone.

**Business** 🡪 business\_voilation, business\_violation\_detail

Each business could have one or many business violations and each business\_violation is associated with one business\_violation\_details.

* Between business and business\_violation it’s one to many relation ship
* Between business\_violation and business\_violation\_detail its one to one relationship.

**Inspection** - This table describes inspection type, score obtained and date on which the inspection is performed on a particular business.

* Each business can have one or more than one inspection performed. Relationship between business and inspection is one to many.

**Violation**\_**catalog**-This table lists out the list of violations, which are listed out by the health inspection department. Each violation catalog is associated with one violation category

**Violation\_category**-This table lists out the high level of violations

**score\_legend**- This table lists out the score ranges and the description of the score. After each inspection each restaurant is associated with score, which is categorized in good, bad or worst based on the score ranges listed out in this table.

**ER DIAGRAM**

Below diagram depicts our ER diagram for the above web application. All the entities and description have been listed out in the above section. Following list depicts the relationship between each entitiy.

A business owner can own one or more businesses and each business can be owned by one or more business owners (many to many relation ship).

Business and Inspection – Many to many

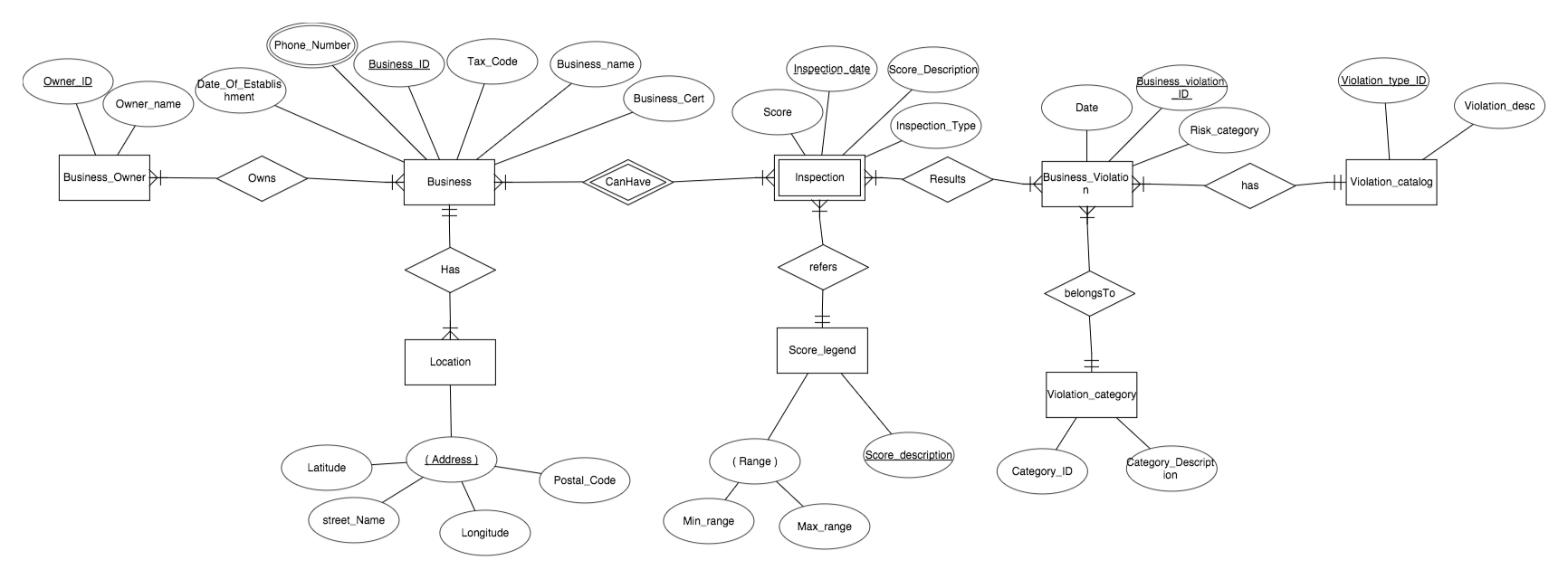
Inspection and business\_violation – Many to many

Business\_violation and violation catalog – one to many

Business and location – one to many

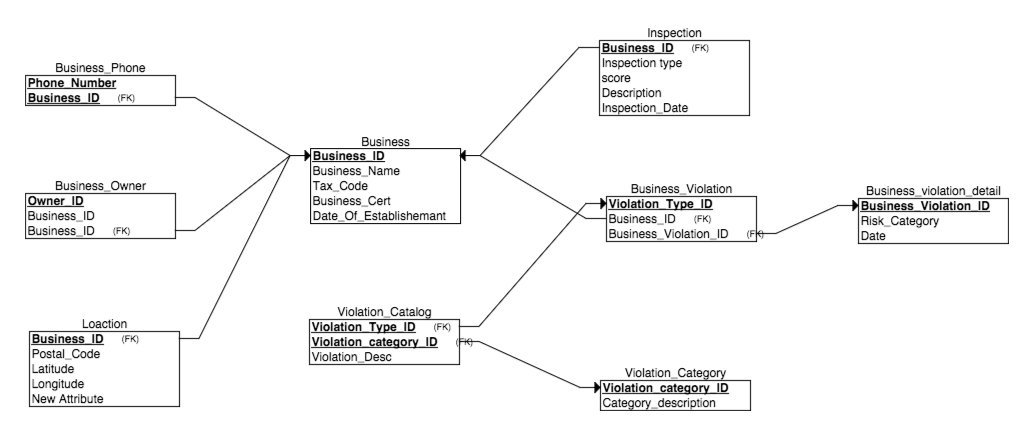
Inspection and score\_legend – one to many

Business\_violation and violation\_category – one to many



**RELATIONAL SCHEMA**

Below diagram depicts relational schema for our web application

****

**STAR SCHEMA**

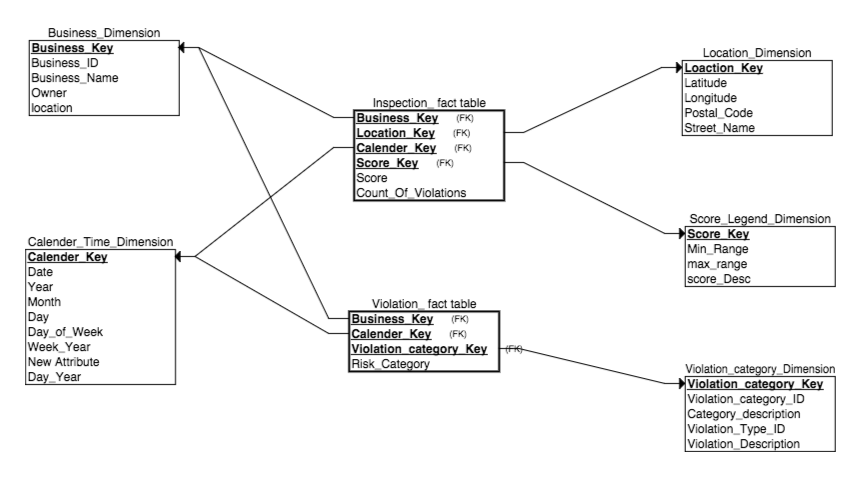
Below diagram depicts star schema where in we have business\_dimension, calendar\_time (drill up and drill down queries), location\_dimension, score\_legend\_dimension, violation\_category\_dimension and two-fact table inspection\_fact and violation\_fact for performing OLAP and OLTP operations.

**Business\_dimension**: This table holds information about the following data, business\_id, business\_name, owner and location.

**Location\_dimension**: This table holds information about locations of all the business.

**Calendar\_dimension**: This table holds the generic information about the dates, quarter, and week etc.

**Score\_legend\_dimension**: this table holds info about min and max range of score and their description

****

**Violation\_category\_time**: holds information about different violations, category and their description.

**Inspection\_fact\_table**: This fact table helps in calculating different (slice/dice/pivot) like

* Score of a particular business, in a specific location
* Score description of a particular business, in a specific location
* Number of violations of a particular business, in particular location in a particular quarter
* Number of violations of a specific business in a quarter/month, year
* Score of a business in quarter/month/year

**Violation\_fact\_table**: This fact table helps in calculating different (slice/dice/pivot) like

* Risk category of a particular business in a particular quarter/month/year
* Number of businesses in a particular violation category
* Number of businesses in a particular violation category in quarter/month/year

**DATA OPERATIONS AND QUERIES**

We have performed DDL and DML operations on our data set and incorporated these operations with our backend.

1. To list the businesses and their distinct inspection dates – (**select**)

select DISTINCT STR\_TO\_DATE(`Inspection\_Date`, '%Y%m%d'),Business\_ID from inspection

2. Number of violations for a particular business – (**select**)

[SELECT](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/select.html) [COUNT](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_count)(business\_violation.Business\_ID),business\_violation.Business\_Violation\_ID,violation\_catalog.violation\_category\_ID FROM `violation\_catalog`,business\_violation WHEREbusiness\_violation.Violation\_Type\_ID =violation\_catalog.Violation\_Type\_ID GROUP by violation\_catalog.violation\_category\_ID

3. Number of businesses in each violation category (Analytical query - **slice**)

[SELECT](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/select.html) [COUNT](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_count)(Violation\_Fact\_Table.Business\_Key)as No\_of\_Violations,Violation\_Category\_Dimension.Category\_Description FROM`Violation\_Fact\_Table`,Violation\_Category\_Dimension WHERE Violation\_Fact\_Table.Violation\_Category\_Key=Violation\_Category\_Dimension.Violation\_Category\_Key GROUP BYViolation\_Category\_Dimension.Violation\_Category\_ID

4. Average Score of each business after an inspection (**Slice**)

SELECT Business\_Dimension.Business\_ID ,[avg](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html" \l "function_avg)(Inspection\_Fact\_Table.Score) as AVG\_Score FROM `Inspection\_Fact\_Table`,Business\_Dimension where Inspection\_Fact\_Table.Business\_Key=Business\_Dimension.Business\_Key GROUP BY Business\_Dimension.Business\_ID

5. Number of businesses in each score range (**Dice**)

SELECT countSixty as "50-60",countEighty as "60-80",countHundred as "80-100" FROM (SELECT COUNT(\*) AS countSixty FROM (SELECT Business\_Dimension.Business\_ID ,avg(Inspection\_Fact\_Table.Score) as AVG\_Score FROM `Inspection\_Fact\_Table`,Business\_Dimension where Inspection\_Fact\_Table.Business\_Key=Business\_Dimension.Business\_Key GROUP BY Business\_Dimension.Business\_ID HAVING AVG\_Score BETWEEN 50 AND 60) AS sixty)AS sixtyTable

INNER JOIN

(SELECT COUNT(\*) AS countEighty FROM (SELECT Business\_Dimension.Business\_ID ,avg(Inspection\_Fact\_Table.Score) as AVG\_Score FROM `Inspection\_Fact\_Table`,Business\_Dimension where Inspection\_Fact\_Table.Business\_Key=Business\_Dimension.Business\_Key GROUP BY Business\_Dimension.Business\_ID HAVING AVG\_Score BETWEEN 60 AND 80) AS eighty )AS eightyTable

INNER JOIN

(SELECT COUNT(\*) AS countHundred FROM (SELECT Business\_Dimension.Business\_ID ,avg(Inspection\_Fact\_Table.Score) as AVG\_Score FROM `Inspection\_Fact\_Table`,Business\_Dimension where Inspection\_Fact\_Table.Business\_Key=Business\_Dimension.Business\_Key GROUP BY Business\_Dimension.Business\_ID HAVING AVG\_Score BETWEEN 80 AND 100) AS hundred )AS hundredTable

6. Number of violations for each business (**pivot**)

[SELECT](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/select.html) Business\_Dimension.Business\_ID,[COUNT](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_count)(Violation\_Fact\_Table.Violation\_Category\_Key) as No\_of\_violations FROM `Violation\_Fact\_Table`,Business\_Dimension WHEREViolation\_Fact\_Table.Business\_Key = Business\_Dimension.Business\_Key GROUP BY Business\_Dimension.Business\_ID

7. Number of business in each violation range i.e number of businesses who have violations between 0-20 – (**dice**)

SELECT countTwenty as "0-20",countForty as "20-40",countSixty as "40-60",countEighty as "60-80",countHundred as "80-100" FROM

(SELECT COUNT(\*) AS countTwenty FROM (SELECT COUNT(Violation\_Fact\_Table.Violation\_Category\_Key) as No\_of\_violations FROM `Violation\_Fact\_Table`,Business\_Dimension WHERE Violation\_Fact\_Table.Business\_Key = Business\_Dimension.Business\_Key GROUP BY Business\_Dimension.Business\_ID HAVING No\_of\_violations BETWEEN 0 AND 20) AS twenty )AS twentyTable

INNER JOIN

(SELECT COUNT(\*) AS countForty FROM (SELECT COUNT(Violation\_Fact\_Table.Violation\_Category\_Key) as No\_of\_violations FROM `Violation\_Fact\_Table`,Business\_Dimension WHERE Violation\_Fact\_Table.Business\_Key = Business\_Dimension.Business\_Key GROUP BY Business\_Dimension.Business\_ID HAVING No\_of\_violations BETWEEN 20 AND 40) AS forty )AS fortyTable

INNER JOIN

(SELECT COUNT(\*) AS countSixty FROM (SELECT COUNT(Violation\_Fact\_Table.Violation\_Category\_Key) as No\_of\_violations FROM `Violation\_Fact\_Table`,Business\_Dimension WHERE Violation\_Fact\_Table.Business\_Key = Business\_Dimension.Business\_Key GROUP BY Business\_Dimension.Business\_ID HAVING No\_of\_violations BETWEEN 40 AND 60) AS sixty )AS sixtyTable

INNER JOIN

(SELECT COUNT(\*) AS countEighty FROM (SELECT COUNT(Violation\_Fact\_Table.Violation\_Category\_Key) as No\_of\_violations FROM `Violation\_Fact\_Table`,Business\_Dimension WHERE Violation\_Fact\_Table.Business\_Key = Business\_Dimension.Business\_Key GROUP BY Business\_Dimension.Business\_ID HAVING No\_of\_violations BETWEEN 60 AND 80) AS eighty )AS eightyTable

INNER JOIN

(SELECT COUNT(\*) AS countHundred FROM (SELECT COUNT(Violation\_Fact\_Table.Violation\_Category\_Key) as No\_of\_violations FROM `Violation\_Fact\_Table`,Business\_Dimension WHERE Violation\_Fact\_Table.Business\_Key = Business\_Dimension.Business\_Key GROUP BY Business\_Dimension.Business\_ID HAVING No\_of\_violations BETWEEN 80 AND 100) AS hundred )AS hundredTable

8. Violations for a particular business (**dice**)

[SELECT](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/select.html) Business\_Dimension.business\_name,Business\_Dimension.Owner,location\_dimension.Street\_Name, [COUNT](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_count)(DISTINCT Violation\_Fact\_Table.Violation\_Category\_Key) asNo\_of\_violations FROM `Violation\_Fact\_Table`,Business\_Dimension,Inspection\_Fact\_Table,location\_dimension WHERE Violation\_Fact\_Table.Business\_Key =Business\_Dimension.Business\_Key [and](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) Business\_Dimension.Business\_Key = Inspection\_Fact\_Table.Business\_Key [and](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and)Inspection\_Fact\_Table.Location\_Key=location\_dimension.Location\_Key [and](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) Business\_Dimension.Business\_ID = 10

9. Number of time a business is inspected (Analysis Query)

[SELECT](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/select.html) Business\_Dimension.Business\_ID,[COUNT](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_count)(Inspection\_Fact\_Table.Calendar\_Key) as No\_of\_times\_Inspected FROM `Inspection\_Fact\_Table`,Business\_Dimension whereInspection\_Fact\_Table.Business\_Key=Business\_Dimension.Business\_Key GROUP BY Business\_Dimension.Business\_ID

10. Number of times a business is inspected in a particular year. (**Drill up**)

[SELECT](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/select.html) Business\_Dimension.Business\_ID,Calendar\_TIME\_DIMENSION.YEAR,[COUNT](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_count)(Calendar\_TIME\_DIMENSION.Calendar\_Key)AS No\_of\_Times\_Inspected\_A\_Year FROMInspection\_Fact\_Table,Business\_Dimension,Calendar\_TIME\_DIMENSION WHERE Inspection\_Fact\_Table.Business\_Key=Business\_Dimension.Business\_Key [and](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and)Inspection\_Fact\_Table.Calendar\_Key=Calendar\_TIME\_DIMENSION.Calendar\_Key GROUP BY Business\_Dimension.Business\_ID,Calendar\_TIME\_DIMENSION.YEAR

11. Distribution of score on different years for particular restaurant (shows how the restaurant improved or got worse) -- **DICE**

[SELECT](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/select.html) [AVG](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_avg)(Inspection\_Fact\_Table.Score),Calendar\_TIME\_DIMENSION.YEAR FROM `Inspection\_Fact\_Table`,Business\_Dimension,Calendar\_TIME\_DIMENSION WHERE Inspection\_Fact\_Table.Calendar\_Key=Calendar\_TIME\_DIMENSION.Calendar\_Key [and](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) Inspection\_Fact\_Table.Business\_Key=Business\_Dimension.Business\_Key [and](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and)Business\_Dimension.Business\_ID=10 GROUP BY Calendar\_TIME\_DIMENSION.YEAR

12. Scores for a particular restaurant in each month (**drill down**)

[SELECT](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/select.html) Business\_Dimension.business\_name,Calendar\_TIME\_DIMENSION.MONTH,Inspection\_Fact\_Table.Score FROM Inspection\_Fact\_Table,Calendar\_TIME\_DIMENSION,Business\_Dimensionwhere Inspection\_Fact\_Table.Calendar\_Key=Calendar\_TIME\_DIMENSION.Calendar\_Key [and](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) Business\_Dimension.Business\_Key=Inspection\_Fact\_Table.Business\_Key [AND](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and)Calendar\_TIME\_DIMENSION.YEAR ="2014" [and](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) Business\_Dimension.Business\_ID = 10 GROUP BY Calendar\_TIME\_DIMENSION.MONTH

13. Number of times a particular business is inspected in each month (**drill down**)

[SELECT](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/select.html) Business\_Dimension.business\_name,Calendar\_TIME\_DIMENSION.MONTH,[COUNT](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_count)(Inspection\_Fact\_Table.Business\_Key) FROM`Inspection\_Fact\_Table`,Calendar\_TIME\_DIMENSION,Business\_Dimension where Inspection\_Fact\_Table.Calendar\_Key=Calendar\_TIME\_DIMENSION.Calendar\_Key [and](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and)Business\_Dimension.Business\_Key=Inspection\_Fact\_Table.Business\_Key [AND](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) Calendar\_TIME\_DIMENSION.YEAR ="2013" [and](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) Business\_Dimension.Business\_ID = 54 GROUP BY Calendar\_TIME\_DIMENSION.MONTH

14. Number of good restaurants in a particular location (**dice**) — for all the businesses

select DISTINCT(b.bn) from (select score\_legend\_dimension.score\_desc,a.business\_name as "bn",a.score,a.ln from score\_legend\_dimension,(select business\_name,Inspection\_Fact\_Table.Score as "score",location\_dimension.location\_name as "ln" from Business\_Dimension, location\_dimension,Inspection\_Fact\_Table,score\_legend\_dimension where Business\_Dimension.Business\_Key=Inspection\_Fact\_Table.Business\_Key and location\_dimension.Location\_Key=Inspection\_Fact\_Table.Location\_Key and Inspection\_Fact\_Table.Score\_Key=score\_legend\_dimension.score\_key) a

where a.score >=score\_legend\_dimension.min\_range and a.score<=score\_legend\_dimension.max\_range and score\_legend\_dimension.score\_desc="good" and a.ln="22nd street") b

15. How well a particular restaurant is doing – **dice**

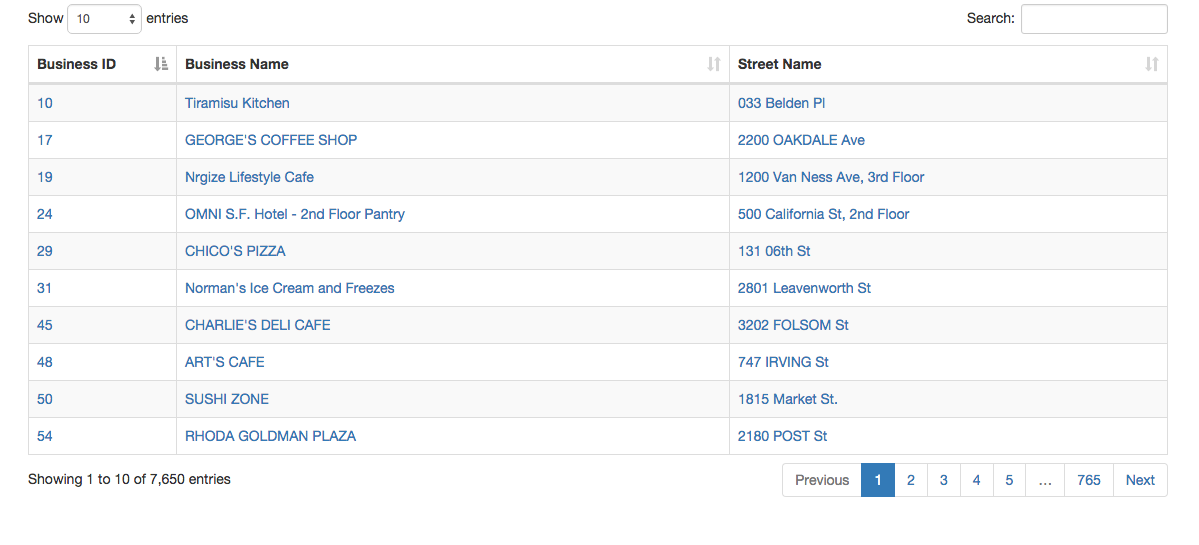
select count(score\_legend\_dimension.score\_desc),score\_legend\_dimension.score\_desc,a.business\_name as "bn",a.score,a.ln from score\_legend\_dimension,(select business\_name,Inspection\_Fact\_Table.Score as "score",location\_dimension.location\_name as "ln" from Business\_Dimension, location\_dimension,Inspection\_Fact\_Table,score\_legend\_dimension where Business\_Dimension.Business\_Key=Inspection\_Fact\_Table.Business\_Key and location\_dimension.Location\_Key=Inspection\_Fact\_Table.Location\_Key and Inspection\_Fact\_Table.Score\_Key=score\_legend\_dimension.score\_key) a

where a.score >=score\_legend\_dimension.min\_range and a.score<=score\_legend\_dimension.max\_range  GROUP BY a.business\_name

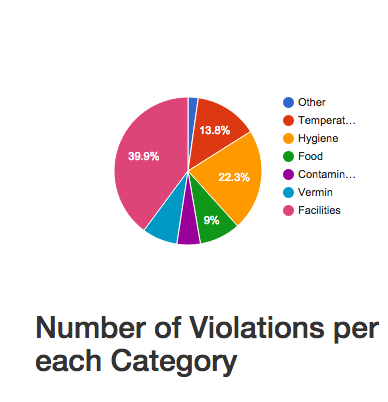
**USER ACTIONS AND SCREEN SHOTS**

Below figure depicts the starting page (index.php)- where in user is able to view all the restaurant businesses in SF and he can click on a particular business ID to view a particular restaurant details.

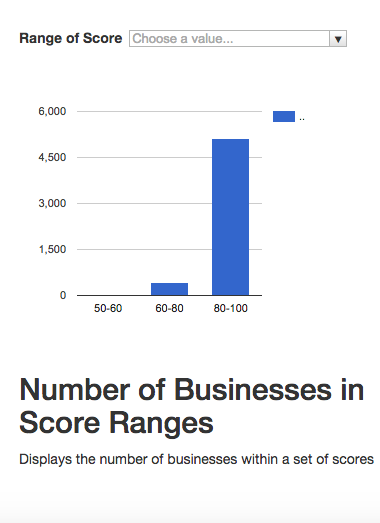
1. On the right side corner of the page, user has the option to search for a restaurant. That action takes him to a particular restaurant business page.



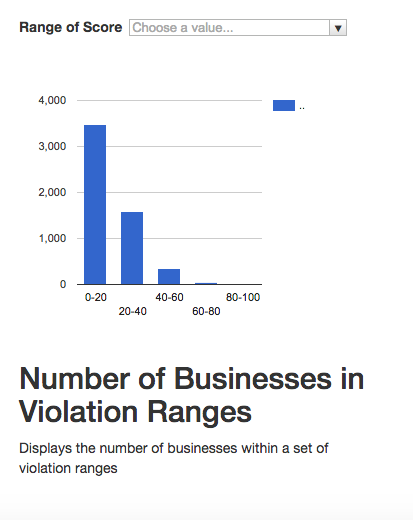
2. And on the same page, when user scrolls down, user can view different analytical perspective of data. Like the one shown below – dice operation performed. The below diagram shows number of violations in each category for all the businesses in SF. i.e., 39.9% of businesses in SF are violating facilities.



3. The below diagram shows slice. It depicts number of business in a particular score range. User has an option to select a value from the drop down i.e., number of restaurants in 20-40 score range etc etc



4. Slice operation- below diagram depicts number of businesses who has violation between 20-40. User has an option to select a range from the drop down and view the results.



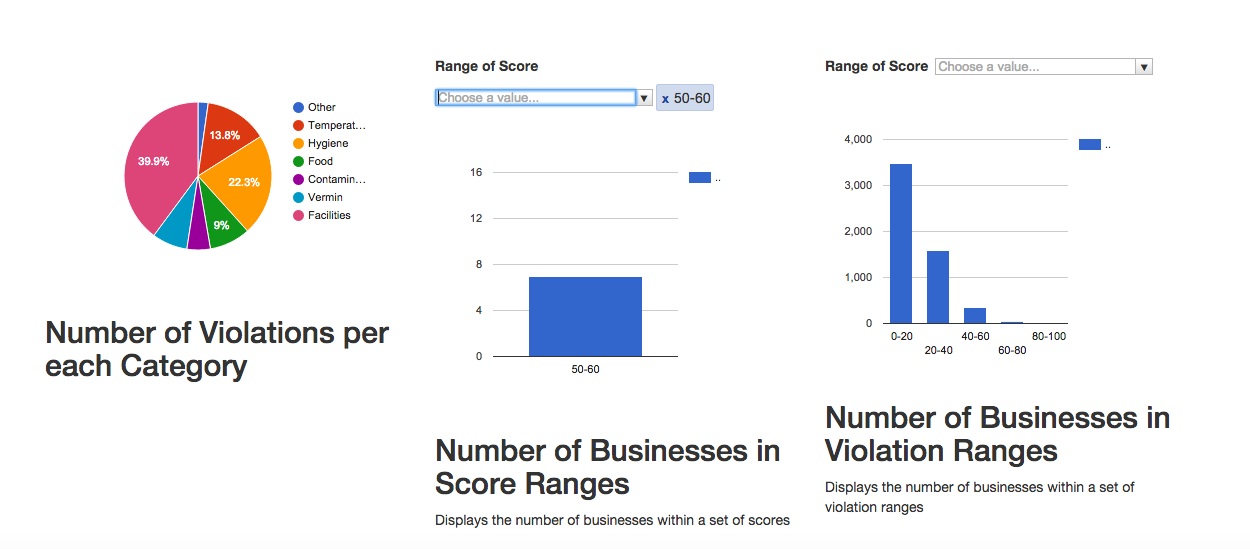
5. The following operations are showed on the first page

Select – displays all the business in tabular form

Dice - Number of business in each violation category

Slice - Number of businesses in particular scores range

Slice - Number of businesses in a set of violation ranges



6. After user selects a particular business, the below page is displayed.

* It shows business details in a tabular form
* Average scores per year/per month – drill up and drill down
* Number of inspections per year/per month – drill up and drill down
* Number of violations in each violation category for this particular business - pivot

