|  |
| --- |
| Restaurant Chain Database |
| Sam YuDatabase ManagementRobert PerkoskiApril 19, 2017 |

# Table of Contents

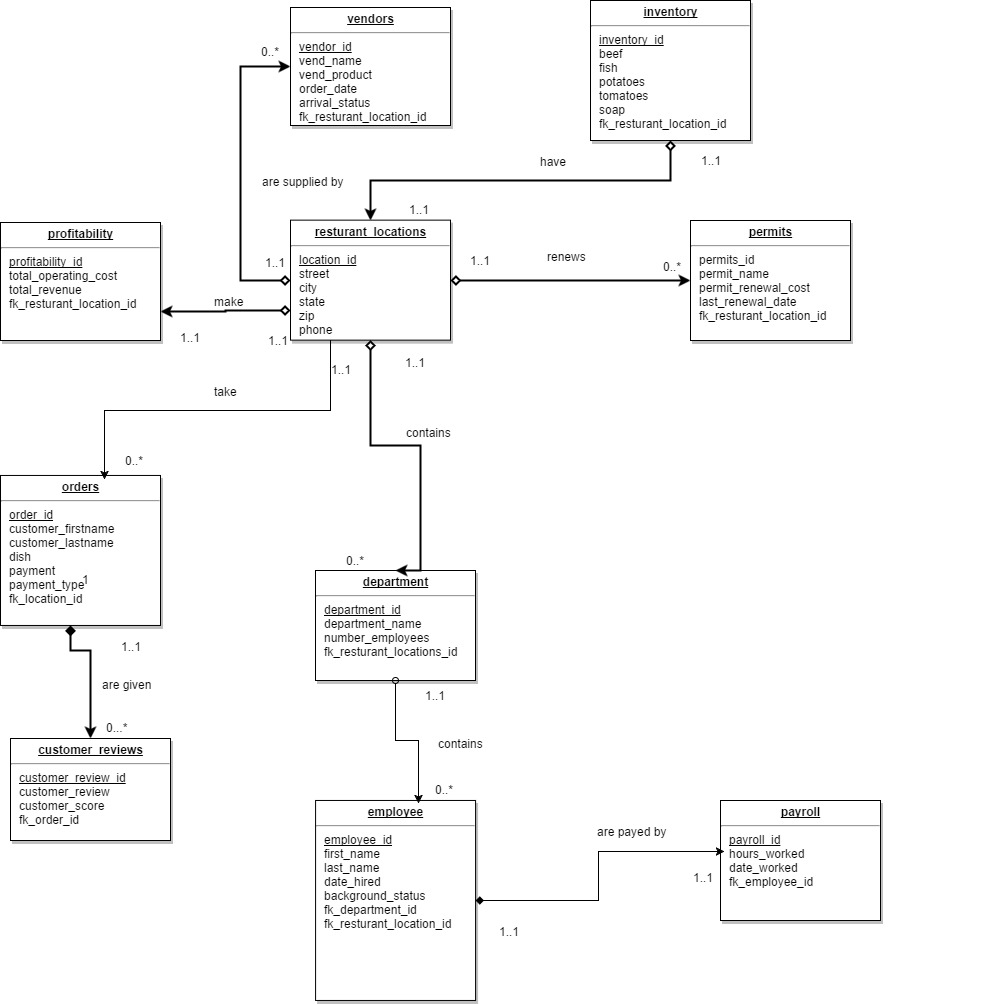
1. Introduction
2. UML Diagram
3. Business Rules
4. Entity and Attribute Descriptions
5. 10 Useful Questions to Ask
6. Closing Statement

# Introduction

This project is a UML compliant E-R model of a restaurant chain’s database. The database is designed to record and track the normal day to day operations of any restaurant chain. Using the records from this database, users can track employee records, inventory stock, sales and much more!

Hopefully with this database model any restaurant chain can monitor and improve its sales with order volume and customer reviews. The chain can also reduce inventory waste and shortages with a vendor and supply data. Another use of the database is to track overall revenue, total cost and most importantly profit. Finally, the chain can manage human resources with records of background checks, employment length and department overhead.

# UML Diagram



# Business Rules

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity 1 | Entity 2 | Cardinality on 1 | Cardinality on 2 | Business Rules |
| Restaurant\_locations | vendors | 1..1 | 0..\* | Restaurant can 0 or have many vendors.  Vendors can have 1 restaurant |
| Restaurant\_locations | inventory | 1..1 | 1..1 | Restaurants can have 1 inventory  An inventory can have 1 restaurant |
| Restaurant\_locations | departments | 1..1 | 0..\* | Restaurants can have 0 or many departments. Departments have 1 restaurant |
| Restaurant\_locations | profitability | 1..1 | 1..1 | Restaurants have 1 profitability chart. Profitability charts have 1 restaurant |
| Restaurant\_locations | orders | 1..1 | 0..\* | Restaurants can have 0 many or many orders. Orders are from 1 resturant |
| Restaurant\_locations | permits | 1..1 | 0..\* | Restaurants can have 0 or many permits. Permits are for 1 resturant |
| Orders | Customer\_reviews | 1..1 | 0..\* | Orders can have 0 or many reviews. Reviews can come from only 1 order |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| departments | employees | 1..1 | 0..\* | Departments can have 0 or many employees. Employees can only be from 1 department. |
| Employees | payroll | 1..1 | 1..1 | Employees have 1 payroll. Payroll record is specific to 1 employee. |

# Attributes and Entities

|  |  |  |
| --- | --- | --- |
| Resturant\_location |  |  |
| Attribute | Data type | description |
| Location\_id | Int | Primary key |
| street | varchar | The street name |
| City | varchar | The city name |
| State | varchar | The state name |
| zip | int | Zip code name |
| phone | double | Phone number for resturant |

|  |  |  |
| --- | --- | --- |
| profitability |  |  |
| Profitability\_id | Int | Primary key |
| Total\_operating\_cost | Double | Total cost of operation |
| Total\_revenue | Double | Total revenue made |
| Fk\_resturant\_location\_id | Int | Foreign key |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Vendor |  |  |
| Vend\_name | varchar | Name of vendor |
| Vend\_product | varchar | Type of product |
| Order\_date | datetime | Date of order |
| Arrival\_status | boolean | Has the order arrived yet |
| inventory |  |  |
| Inventory\_id | int | Primary key |

|  |  |  |
| --- | --- | --- |
| tomtatoes | varchar | Amount of tomatoes |
| soap | varchar | Amount of soap |
| beef | varchar | Amount of beef |
| fish | Varchar | Amount of fish |
| Potatoes | varchar | Amount of potatoes |
| Tomatoes  Fk\_resturant\_location\_id | Varchar  int | Amount of tomatoes  Foreign key |

|  |  |  |
| --- | --- | --- |
| Permits |  |  |
| Permits\_id | Int | Primary key |
| Permit\_name | Varchar | Name of permit |
| Permit\_renewal\_cost | Double | Cost to renew permit |
| Last\_renewal\_date | datetime | Last time permit was renewed |
| Fk\_resturant\_location\_id | Int | Foreign key |
| orders |  |  |
| Orders\_id | int | Primary key |

|  |  |  |
| --- | --- | --- |
| Customer\_firstname | varchar | First name of customer |
| Customer\_lastname | varchar | Last name of customer |
| dish | Varchar | Name of dish |
| Payment | Double | Amount payed |
| Payment\_type | Varchar | Type of payment cash or visa |
| Fk\_location\_id | Int | Foreign key |
| Customer\_reviews |  |  |
| Customer\_review\_id | int | Primary key |

|  |  |  |
| --- | --- | --- |
| Customer\_review | Varchar | The text comment by customer |
| Customer\_score | int | Score out of 10 |
| Fk\_order\_id | Int | Foreign key |
| Department |  |  |
| Department\_id | Int | Primary key |
| Department\_name | Varchar | Name of department |
| Number\_employees | Int | Number of employees in the departmentf |
| Fk\_resturant\_locations\_id | Int | Foreign key |

|  |  |  |
| --- | --- | --- |
| employee |  |  |
| Employee\_id | Int | Primary key |
| First\_name | Varchar | Employee first name |
| Last\_name | Varchar | Employee last name |
| Date\_hired | Datetime | Date employee was hired |
| Background\_status | Varchar | Current criminal background status of employees |
| Fk\_department\_id | Int | Foreign key |
| Fk\_resturant\_location\_id | Int | Foreign key |

|  |  |  |
| --- | --- | --- |
| Payroll |  |  |
| Payroll\_id | Int | Primary key |
| Hours\_worked | Int | Hours worked that week |
| Date\_worked | Datetime | The start of the week |
| Fk\_employee\_id | Int | Foreign key |
|  |  |  |
|  |  |  |
|  |  |  |

# List of Questions

1. What departments are there in the company and what job titles are there?
2. What is the operating cost of each restaurant and where are they located?
3. What restaurants currently don’t have any vendors?
4. What is the highest revenue of the 10 restaurants and lowest revenue of the 10 restaurants?
5. What employees has similar backgrounds and have not yet had a background check?
6. What orders from vendors have not arrived yet?
7. What branches need to reorder soap and have low stocks?
8. Find top 3 most profitability branches for distributing company awards.

# Closing Statement

The hardest part of this project is populating the database. Although relatively simple in its coding complexity the sheer amount of lines required to populate a ten table, ten attributes per table database is massive. Almost 500 lines of code to populate the database. Another part I found difficult is relationships between different attributes. In theory everything has a proper relationship and cardinalities. But when actually trying to implement it, it’s a head scratching task.