FINAL PROJECT

Fall 2017 CSCI E-59

Description

Database design report for CSCI E-59 final project.

Charles Mateer
cmateer@g.harvard.edu

Table of Contents

Final Project

Description

The goal of this project is to create a database to power the backend of a REST API for a web application I would like to prototype. The subject matter of the database does not actually have anything to do with my work and is more of a hobby project of a somewhat silly nature that I believe has been both fun and educational to implement.

The database stores *simulated* near real-time (NRT), crowd-sourced, location-based information about… smells. Yes, smells. The idea came to me while my significant other and I were walking around downtown Richmond amongst the myriad of diverse and often offensive smells that come with living in a city. What started off as a recurring joke soon became a more concrete desire to create a map of "smells" in the City with associated descriptive information such as magnitude, persistence, subjective description, and an overall semi-objective quantitative ranking on a Likert scale.

The general idea is to provide the database for a web/mobile application for users of the application, who come across a distinct smell while walking around the City, to record their current location and input the above mentioned descriptive information about the smell they have encountered. This information is then aggregated with other user input to strengthen or negate other user scores within that "Smellscape" (US Census Blocks) essentially providing an average ranking. For example the more "votes" that location has the more weight that point will have. Ideally, this would lead to a rich map of "Smellscapes" in the City denoted cartographically on an interactive map by varying colors and sizes based on the aggregated user input. Users also would have the ability to comment on specific Smellscapes and subscribe to those Smellscapes to see the associated Smell Profiles that make up a Smellscape.

Changes Since the Proposal

Overall, not much has changed since my proposal. The main differences you will see is that I have adopted an underscore case (i.e. table_name) naming convention rather than my original camel case (i.e. TableName) that I used in some aspects of my original design proposal.

Additionally, I originally had foreign keys in the Smellscape entity for subscription and comment log tables. I believe this was incorrect to do and rather those FKs should be for the Smellscape and be in the respective subscription log and comment log entity. I have made this change in my diagrams and in my schema.

I had originally planned to use JSON (specifically the GeoJSON specification) to store spatial data but then opted to use the SDO_Geometry data type upon learning that the Spatial Data and Graph capability was installed on our course's Oracle instance. I had initially still planned to use JSON as temporary storage for spatial data and then use JSON to SDO conversion functions to load data into spatial

database tables, but I learned that our course's version of Oracle does not yet natively support JSON data types. Therefore, I opted to use well-known text (WKT) geometries as explained in the Special Features section of this document.

Given that I decided to use spatial data types as my special feature (described below in the Special Features section) I opted to not use hashed passwords as I had originally planned. I ended up spending a large amount of time researching how to implement spatial data types and rather than continue to research another special feature (hashed passwords) I decided to use plain text passwords and to focus on the design elements of the database instead.

Lastly, I have opted for numeric type IDs rather than the alphabetic ones in my original design.

Listing of Bugs

I did not run into too many bugs but more so have had to add certain assumptions to my project for it to work. These mainly revolve around the spatial component of the database (Smell Profile and Smellscape entities).

Initially, I had planned to only add Smellscapes (US Census Block geometries) for Richmond, VA which would come with the assumption that any points used to test the database would either need to be within the bounds of the city or should be outright rejected by the database.

I then discovered that SQLPLUS has a limitation whereby the maximum line length allowable is 2,499. This limitation caused several of my geometry INSERT statements (explained below in the Special Features section) to fail and not be added. You will see in the spool file error messages such as:

SP2-0027: Input is too long (> 2499 characters) - line ignored.

In some instances, this also caused the next line below it to fail as only a truncated version of the line before it was run:

SP2-0734: unknown command beginning "1, -77.437..." - rest of line ignored.

As such, several of the larger Richmond, VA block geometries (since the larger geometries had too many vertices and thus their line length was too long) did not end up getting added to the database. I had initially started to try to identify and comment out these lines, but it became apparent that that process was going to be painstaking and long and would overall detract from me focusing on the more design centric elements of the database.

Therefore, I decided to leave the Smellscape table as is which essentially means it has "holes" in it that will be different than what is seen in the images below (in the Special Features section). This can cause issues as well because the one hundred or so test points I automatically generated will not all work as intended even though they are within the spatial boundary of Richmond. I have tried to provide points I know will work within my inserts in the database as well as in the comments so that the individual/individuals grading this assignment can test out working points. When using the stored procedure for loading a Smell Profile you will see that it failed based on an exception block and a database output message.

Special Features

The special feature I chose to utilize was the use of the Oracle Spatial Data and Graph software to store spatial data types, create spatial metadata and indexes, and to utilize spatial querying functions. I utilized the SDO_Geometry spatial data type in the Smellscape and Smell_Proifle entities to store point geometry (for Smell Profiles) and polygon geometry for Smellscapes.

In order to bulk add Richmond, VA 2010 Census Blocks (obtained from the US Census Bureau as Esri Shapefiles) I used a Python script to cycle through the shapefile's thousands of features and output a well-formed string of text that contained both the FIPS code (unique identifier for geopolitical boundaries in the United States) and the polygon geometries in well-known text (WKT) format (which is supported by the Open Geospatial Consortium). This string was formatted to be a series of SQL INSERT statements that would use the SDO_UTIL.FROM_WKTGEOEMETRY Oracle function to convert WKT geometry into SDO_Geometry and add the feature and FIPS code to the database.

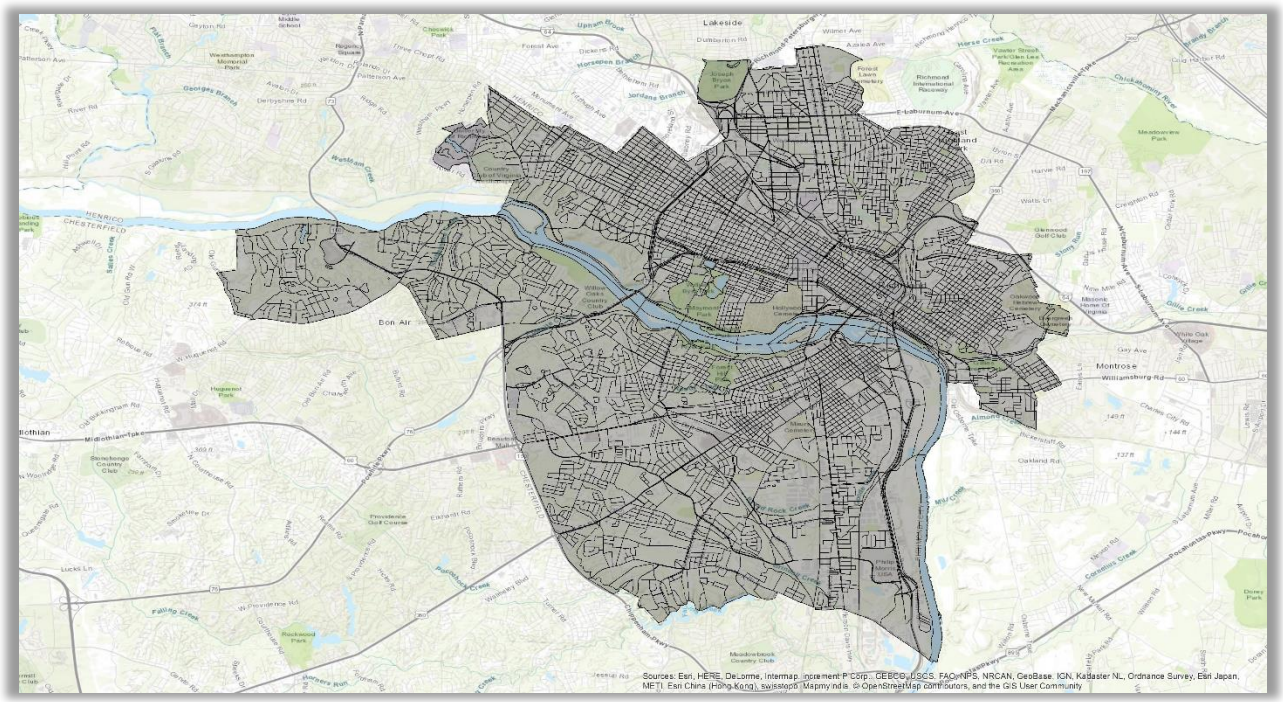
I then utilized a random point generation algorithm in Esri's ArcMap software to generate point geometry randomly within the Richmond, VA spatial boundary. These points were planned to be used as test data. I used a similar (like the one mentioned above) Python function to extract the WKT geometry for these points.

Once spatial data were loaded via WKT I had to update the *feature_geometry.SDO_SRID* attribute for each feature as well as add spatial metadata to the user_sdo_geom_metadata table for each table and column with spatial data.

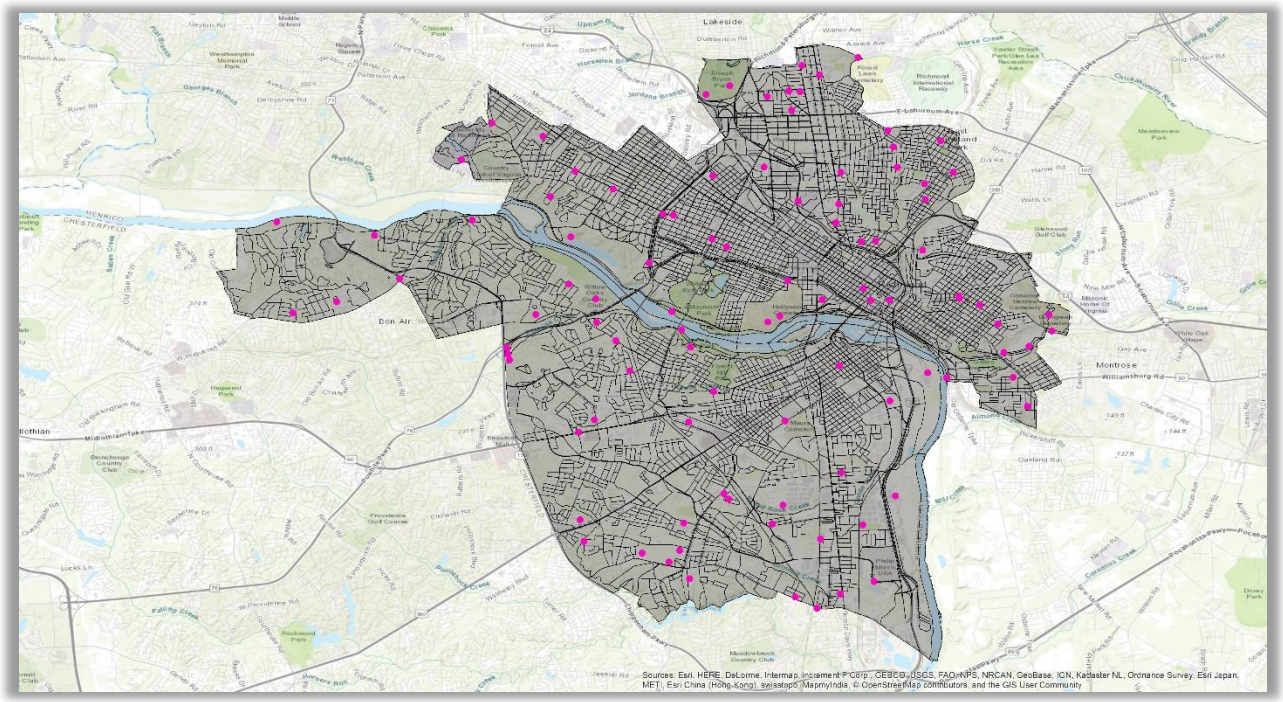
The spatial metadata additions described above include information such as spatial reference system (in this case World Geodetic System 84 (WGS 84) – a common reference system that applications like Google Maps uses) and grid dimensions (in this case the bounds of the longitude/latitude system).

Once the metadata had been added I was able to initiate the creation of spatial R-Tree indexes which allow for spatial querying and quicker searching. The combination of the metadata and the index creation allowed me to then use the spatial CONTAINS function to add not just text based querying but also spatial querying to my SQL scripts. I utilized this spatial querying to check if a given Smell Profile was within a Smellscape and return the associated FIPS code if it was.

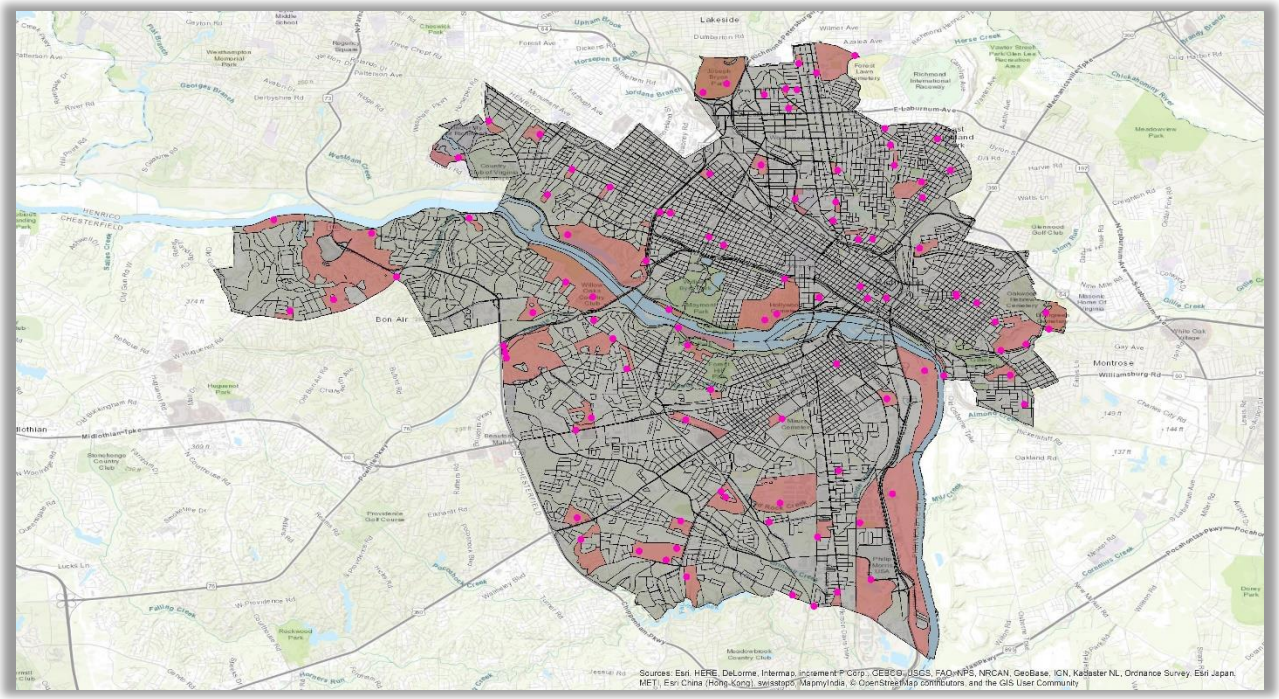
Richmond, Virginia Census Blocks



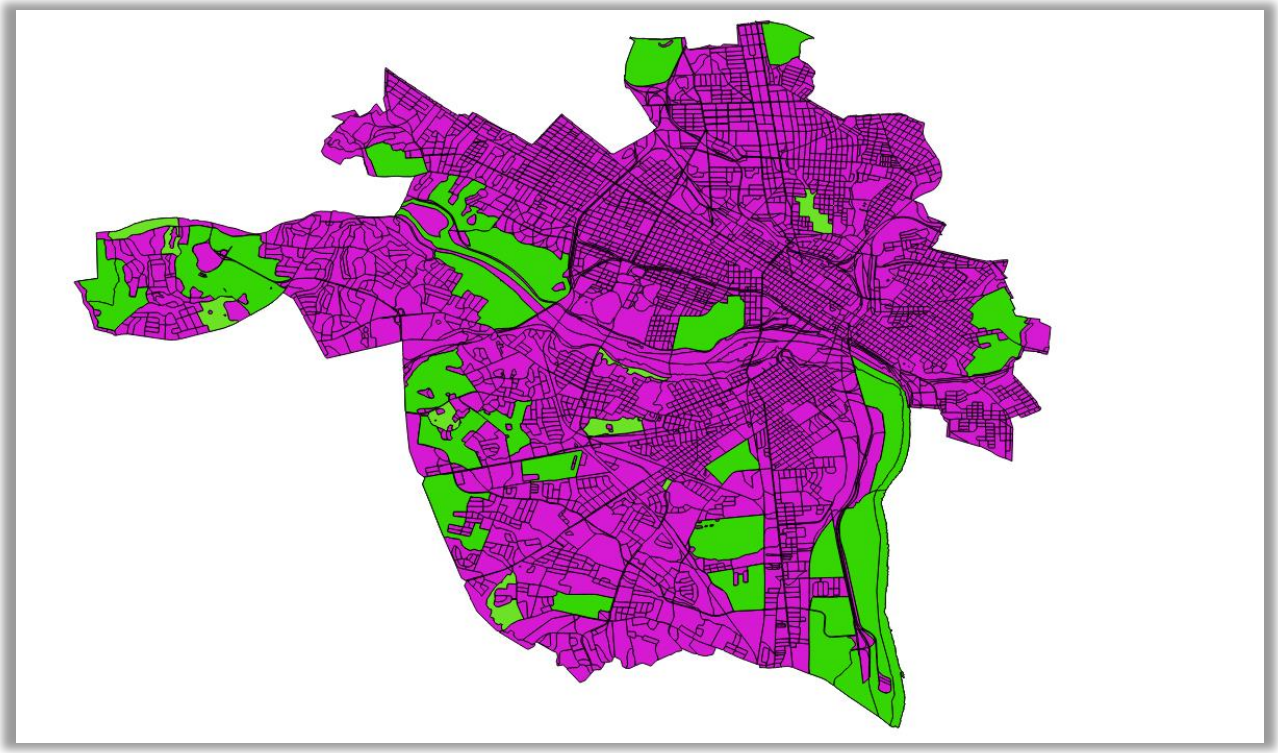
Randomly Generated Points



Expected Intersection Areas



More Likely Outcome Based on Issues Mentioned in the Bugs Section (Green Areas Likely Missing from Dataset in Database)



Next Steps

To get this project fully functional I would need to develop a web front end and develop some middleware for the database and web application to talk to each other. Ultimately, I would like for the app to handle a lot of the spatial components of the application and data entry automatically. For example, at the moment, the database code (through a stored procedure) handles placing a Smell Profile into a specific Smellscape based on spatial relationship. I still must provide the initial coordinate of the Smell Profile manually though whereas I would like for this to be accomplished via the user's current GPS location.

The front end of this application would have a significant mapping component and I would need to use some of the queries I have developed to pass those spatial geometries from the database to a mapping API for display. I have some front-end programming experience and believe I could accomplish this with time. Additionally, I am taking my second web development class here at HES this upcoming semester, so I may be able to accomplish some of these tasks via that course.

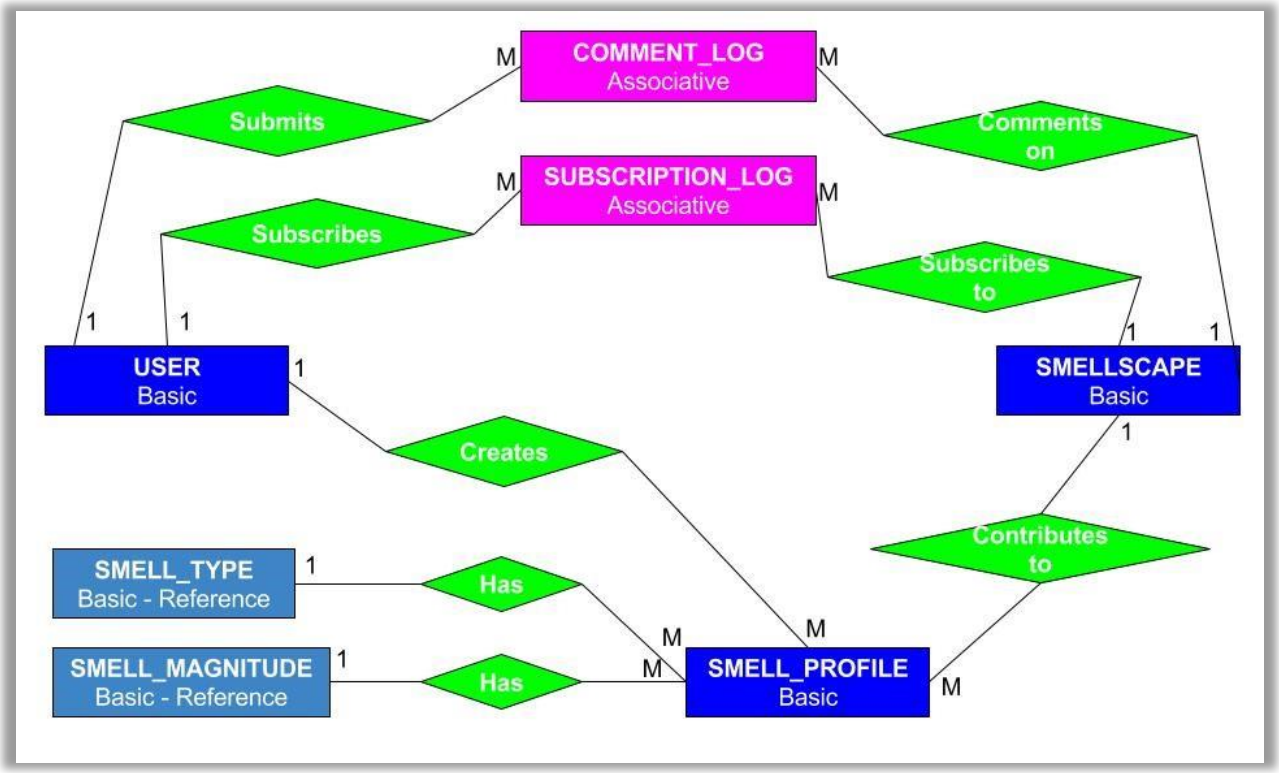
Lastly, one of my main interests will be to try to port the functionality of this database over to different RDBMSs. I have enjoyed learning Oracle but believe that doing this sort of porting will help me to better understand the differences between Oracle and other RDBMSs which I use more frequently at work and in hobbies such as SQL Server and MY SQL.

Original Work Statement

I hereby certify that this project was prepared especially for this course, and that this or a similar version has not been submitted to any other course.

Full Data Model

ER Model



Record Diagrams

user_account Entity					
user_id	first_name	last_name	username	email	user_password
PK					
1	Angel	Jarvis	ajarvis	ajarvis@gmail.com	{Password}
2	Kayree	Chang	kchang	kchang@gmail.com	{Password}
3	Geanie	McJeanie	gmcjeanie	gmcjeanie@gmail.com	{Password}

smell_profile Entity							
fips_code	user_id	latitude	longitude	feature_geom	type_id	magnitude_id	open_description
PK	**PK**						
FK	**FK**				**FK**	**FK**	
2	1	38.5738	-79.2784	SDO Geom	01	02	Dryer sheets, with a hint of jasmine. Very subtle but it has lingered for some time now.
4	2	38.4892	-79.0192	SDO Geom	03	10	A smell most foul! Post-rain sewer runoff mixed with old socks.
1	3	38.1167	-79.2897	SDO Geom	02	05	Sort of chemically... I can't say it was bad, but it wasn't good either.

smellscape Entity	
fips_code	feature_geometry
PK	
517600411003000	{SDO geometry object}
517600411003011	{SDO geometry object}
517600107002002	{SDO geometry object}

subscription_log Entity			
subscription_id	user_id	fips_code	subscription_date
PK			
	FK	**FK**	
1	1	517600411003000	5/23/2016: 00:23:15
2	1	517600411003011	7/15/2017: 02:50:02
3	2	517600411003000	10/20/2017: 15:27:44

comment_log Entity				
comment_id	user_id	fips_code	comment_date	comment_desc
PK				
	FK	**FK**		
1	1	517600411003000	5/23/2016: 00:23:15	Wow! I totally smelled that smell too!
2	1	517600411003011	7/15/2017: 02:50:02	Yep... Can confirm... Very subtle.
3	3	517600411003000	10/20/2017: 15:27:44	Yuck!

Smell_type Entity	
type_ID	type_description
PK	
01	Good
02	Neutral
03	Bad

smell_magnitude Entity		
magnitude_ID	magnitude_description	magnitude_weight
PK		
01	{Short description for weakest}	0.1
02	{Short description}	0.2
03	{Short description}	0.3
04	{Short description}	0.4
05	{Short description}	0.5
06	{Short description}	0.6
07	{Short description}	0.7
08	{Short description}	0.8
09	{Short description}	0.9
10	{Short description for strongest}	1

Outline of Included Files

Primary files in order in which they should be run.

Smell_map_ddl.sql

Set_spatial_metadata.sql (only needs to be run once and has already been run)

Smell_map_load.sql

Smell_map_qc.sql

Smell_map_queries.sql