MAP BASICS: PART 1

Justin Gould (gould29@purdue.edu)

HONR 25315: Foundations of Geospatial Analytics

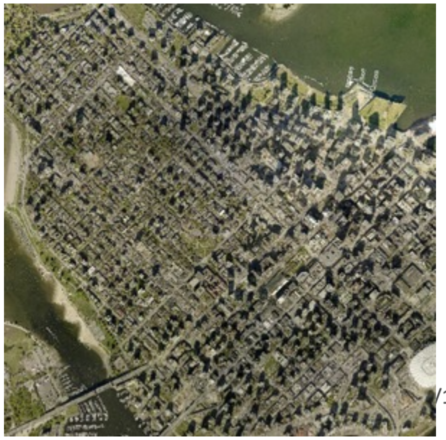
Fall 2021

Topics

- Part 1: What are maps? (Chapter 1 of *PostGIS in Action*)
  - Graphs as maps
  - Data which make up maps
  - Spatial data types
  - Spatial databases
- Part 2: Basic map creation and visualization
  - Preprocessing spatial data
  - Map design principles
  - Mapping via Python
  - Mapping via QGIS

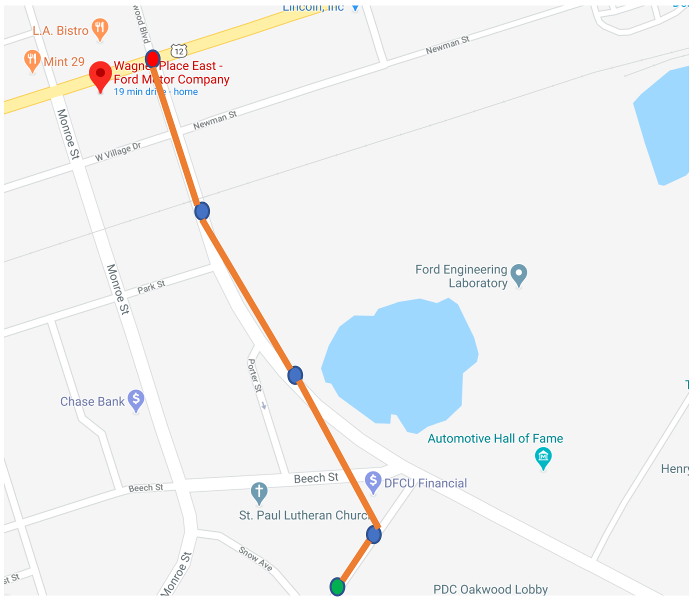
What are Maps?

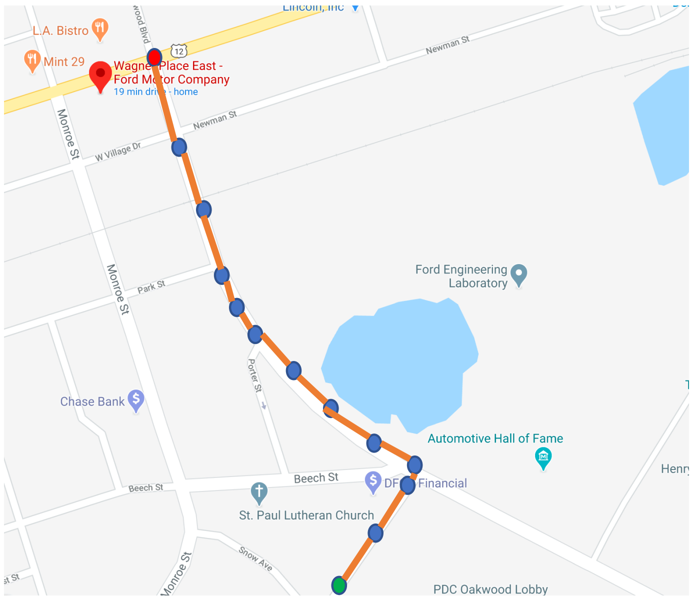
- Geospatial data are the collective data and associated technology containing geographic or locational components
  - Coordinates, address, city, satellite imagery, etc.
- Vector vs. Raster
  - Vector
    - Graphical representations of the real world: points, lines, and polygons
    - Connecting **points** create **lines**, and connecting lines that create an enclosed area are **polygons**
  - Raster
    - Data that is presented in a grid of pixels…typically refers to imagery (Google Maps)
- Each "segment" on a map can store values: color, unit of measurement, etc.



What are Maps?

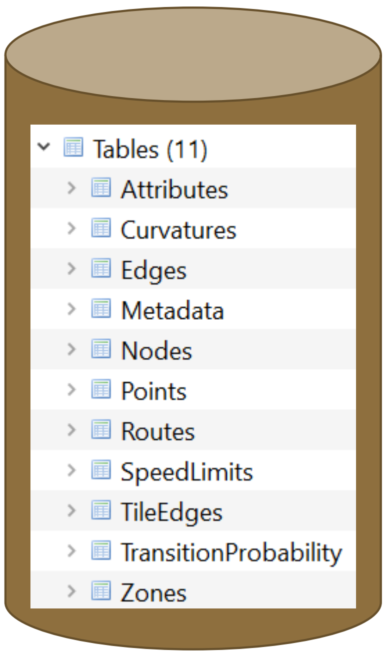
- Think of a map as a graph, with nodes and edges (edges connecting nodes)
  - A straight line drawn between nodes = edge
  - More data (ideally at each change in road curvature, get a new point) = cleaner map





What are Maps?

- Maps can also be a database



Curvatures
- edgeId
- startDist
- heading
- curvature
- tripId

Edges
- id
- tileId
- fromNodeId
- toNodeId
- hwyTypeId
- length
- travelDirection
- frequency
- defaultZone

Nodes
- id
- lat
- lon
- geoHash

Routes
- id
- routeId
- name
- idx
- nd

TransitionProbability
- eFromId
- eToId
- txId
- destId
- freq

Points
- edgeId
- idx
- lat
- lon

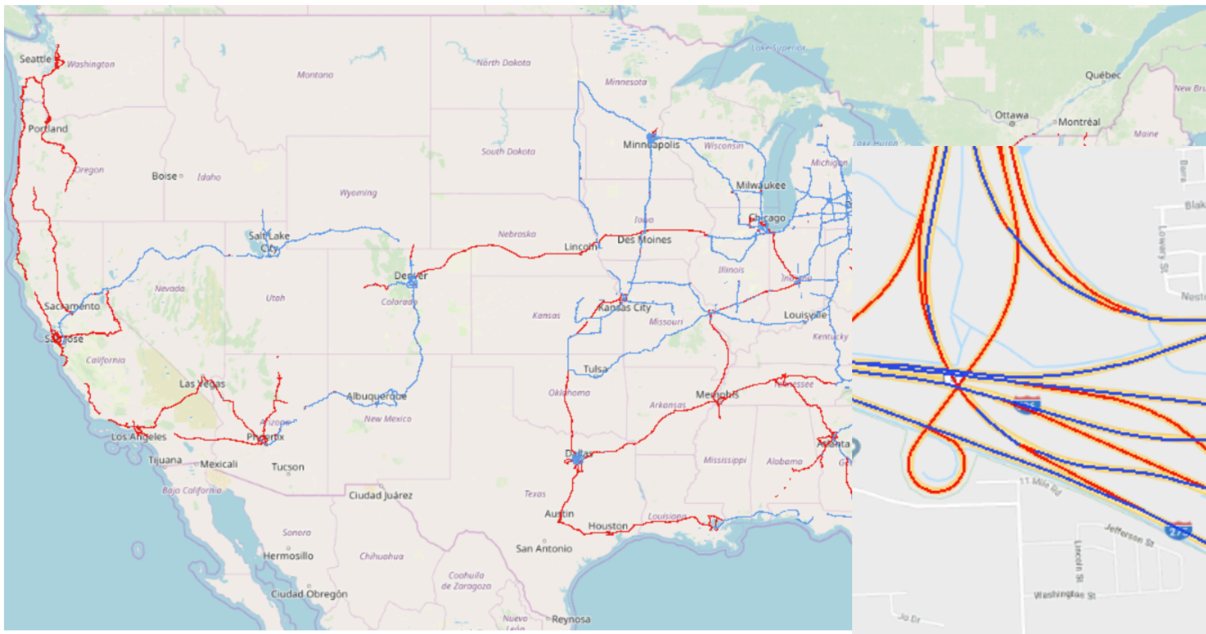
TileEdges
- tileId
- edgeId
- startPointIdx

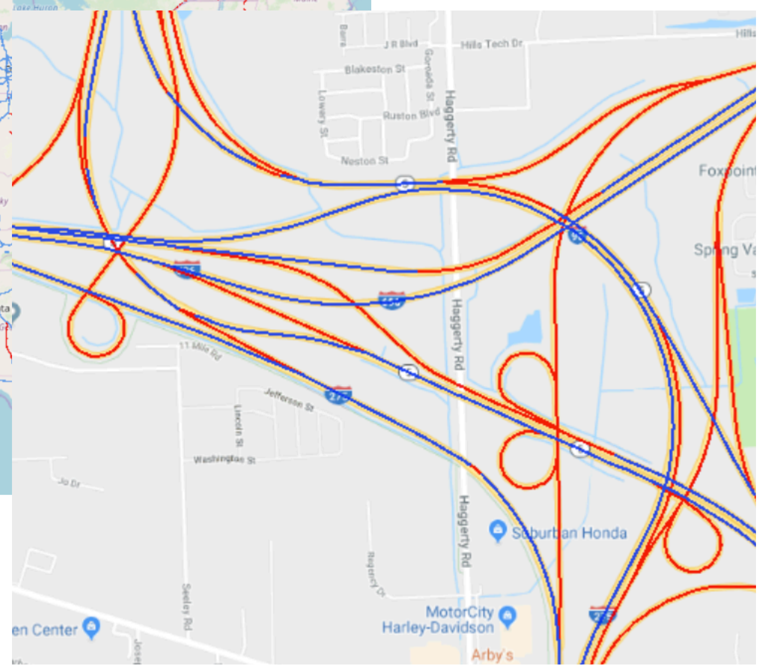
Metadata
- name
- value

Attributes
- edgeId
- startDist
- name
- value

What are Maps?

- Visualizing map database and attributes associated with vector data (lines and points) – *think back to the idea of a* ***graph***





Segment ID

Segment length

Min lane width

Max lane width

Left avg LMC

Right avg LMC

ADAS availability

Has pothole

Construction

…

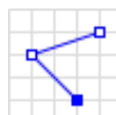
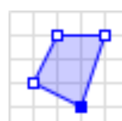
…

…

PURDUE UNIVERSITY

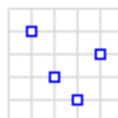
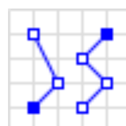
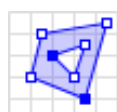
Geometry (Spatial Data) Types

- How can we codify spatial data and represent vector geometry in code?
  - Geometric objects: 2D (x, y), 3D (x, y, z (i.e., altitude))

Type	Definition	Shapes	Examples	
Point	A given space (lon lat)		`{ "type": "Point", "coordinates": [30, 10] }`	POINT(-83.2456381 42.3061845)
Linestring	Connected series of points		`{ "type": "LineString", "coordinates": [ [30, 10], [10, 30], [40, 40] ] }`	LINESTRING(POINT, POINT, POINT, …)
Polygon	Closed shape defined by a connected sequence of (lon lat) pairs		`{ "type": "Polygon", "coordinates": [ [[30, 10], [40, 40], [20, 40], [10, 20], [30, 10]] ] }`	POLYGON((POINT, POINT, …))

Geometry (Spatial Data) Types

- How can we codify spatial data and represent vector geometry in code?
  - Geometric objects: 2D (x, y), 3D (x, y, z (i.e., altitude))

Type	Definition	Shapes	Examples	
Multipoint	Ordered collection of points		`{ "type": "MultiPoint", "coordinates": [ [10, 40], [40, 30], [20, 20], [30, 10] ] }`	MULTIPOINT(POINT, POINT, POINT, …)
Multilinestring	A collection of > 1 linestrings		`{ "type": "MultiLineString", "coordinates": [ [[10, 10], [20, 20], [10, 40]], [[40, 40], [30, 30], [40, 20], [30, 10]] ] }`	MULTILINESTRING(LS, LS, LS, …)
Multipolygon	A collection of polygons that consists which construct from exterior ring and hole list tuples		`{ "type": "MultiPolygon", "coordinates": [ [ [[30, 20], [45, 40], [10, 40], [30, 20]] ], [ [[15, 5], [40, 10], [10, 20], [5, 10], [15, 5]] ] ] }`	MULTIPOLYGON(POLYGON, POLYGON, POLYGON, …)

Geometry (Spatial Data) Types

- As we've seen, spatial data are typically represented as strings or numeric values:
  - **Well-known text** (**WKT**) is a text markup language for representing vector geometry objects:
    - "POINT(LON LAT)"
    - "LINESTRING(POINT, POINT, …)"
  - **GeoJSON** is a format for encoding a variety of geographic data structures:

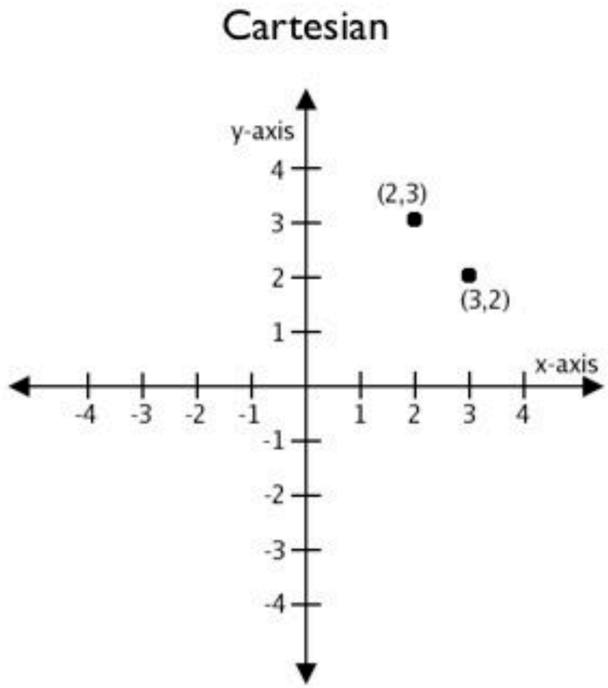
```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [125.6, 10.1]
  },
  "properties": {
    "name": "Dinagat Islands"
  }
}
```

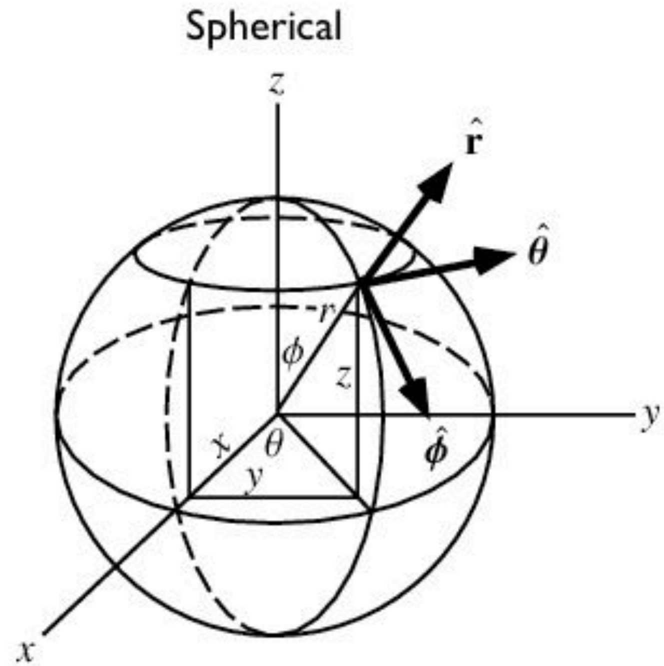
Coordinates and Coordinate Systems

- Coordinate systems contained within EPSG Registry
  - Standard = WGS84 (World Geodetic System 1984)
    - Latitude/longitude coordinate system based on the Earth's center of mass
    - For example, raw GPS data from vehicles comes in this coordinate system
  - Web Mercator Projection (EPSG:3857) = used for display by web-based maps
    - Variant of WGS84, but can be represented in meters

- ~6,000 coordinate systems registered through EPSG Registry
  - States, countries can have their own coordinate system
  - A coordinate system is a projection of spatial data on a flat surface
  - Since there is no perfect way to transpose a curved surface to a flat surface without some distortion, many different map projections exist that provide different properties.

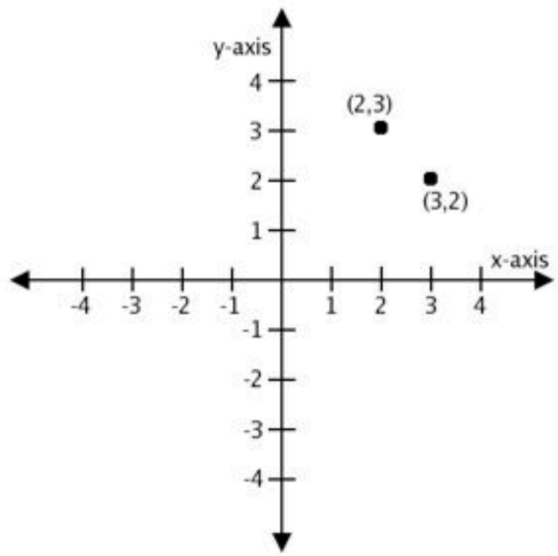
Geography: Cartesian vs. Spherical





Geography: Cartesian vs. Spherical

- Calculating Cartesian distance
  - Cartesian points are on a plane with 2 dimensions: x (latitude) and y (longitude)
  - You can calculate the shortest path (in degrees, in our case), as you would any two points on a plane



Geography: Cartesian vs. Spherical

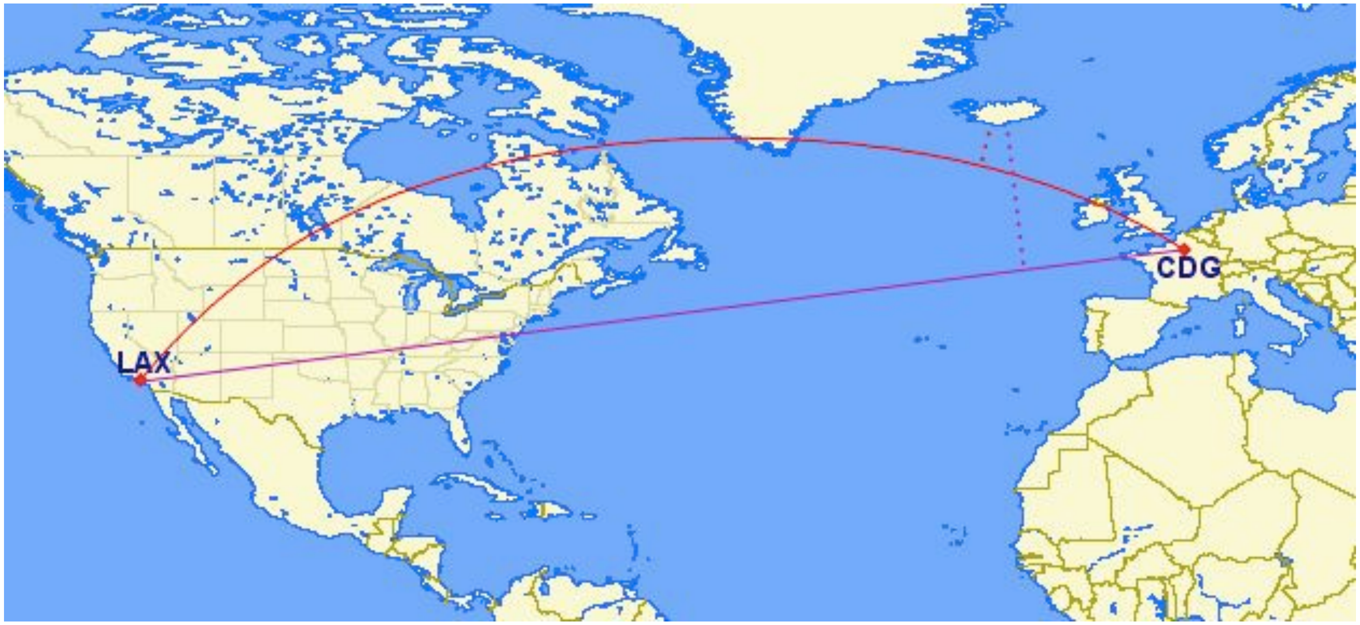
- Calculating Spherical distance
  - Since our earth is round, calculating distance between 2 points is more challenging
  - Haversine formula:

$$\Delta\hat{\sigma} = 2\arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos\phi_s\cos\phi_f\sin^2\left(\frac{\Delta\lambda}{2}\right)}\right).$$

$\Delta\sigma$	Interior Spherical Angle
$\Delta\phi$	Latitude1 - Latitude2
ϕ_s	Latitude1
ϕ_f	Latitude2
$\Delta\lambda$	Longitude1 - Longitude2

Geography: Cartesian vs. Spherical

- Calculate the distance between Los Angeles, California and Paris, France…

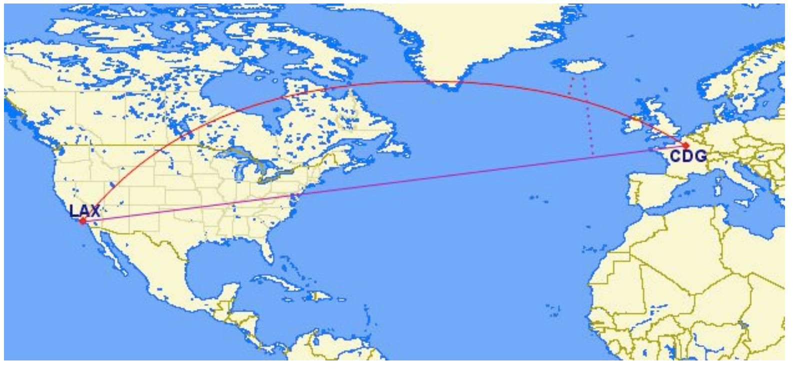


Geography: Cartesian vs. Spherical

- Calculate the distance between Los Angeles, California and Paris, France…

Distance between points **Create cartesian point** **Coordinate system code for Cartesian**

```
SELECT
    ST_Distance(
        ST_GeometryFromText('POINT(-118.4107 33.9415)', 4326),
        ST_GeometryFromText('POINT(2.5457 49.0096)', 4326)
    );
>> 121.891338 (degrees)
```

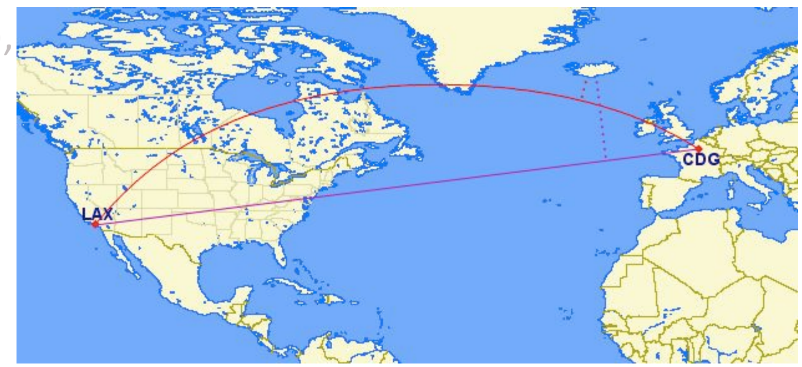


Geography: Cartesian vs. Spherical

- Calculate the distance between Los Angeles, California and Paris, France…

Distance between points **Create spherical point**

```
SELECT
    ST_Distance(
        ST_GeometryFromText('POINT(-118.4107 33.9415)', 4326),
        ST_GeometryFromText('POINT(2.5457 49.0096)', 4326)
    );
>> 121.891338 (degrees)
```




```
SELECT
    ST_Distance(
        ST_GeographyFromText('POINT(-118.4107 33.9415)'),
        ST_GeographyFromText('POINT(2.5457 49.0096)')
    );
>> 9102760.908043034 (meters)
```

Spatial Databases

- Our book's definition:
  - "A *spatial database* is a database with column data types specifically designed to store objects in space—these data types can be added to database tables."
  - "The information stored is usually geographic in nature, such as a point location or the boundary of a lake."
  - "A spatial database is often used as just a storage container for spatial data."

- What does this mean?
  - A (more than likely) relational database which supports querying geographic and non-geographic features via SQL to gain insights into, and manipulate, your data.

*Spatial Databases: An Example**

- Scenario:
  - Ice cream entrepreneurs Jen and Barry have opened their business and now need a database to track orders.
- What data do they collect?
  - When taking an order, they record the customer's name, the details of the order such as the flavors and quantities of ice cream needed, the date the order is needed, and the delivery address.
- What does the spatial database need to answer for Jen and Barry?

1. Which orders are due to be shipped within the next two days?
2. Which flavors must be produced in greater quantities?

- **<u>What are some fields we should include in the database for Jen and Barry?</u>**

* Courtesy of GEOG 868 – Penn State

*Spatial Databases: An Example**

- First attempt:

Customer	Order	DeliveryDate	DeliveryAdd
Eric Cartman	1 vanilla, 2 chocolate	12/1/11	101 Main St
Bart Simpson	10 chocolate, 10 vanilla, 5 strawberry	12/3/11	202 School Ln
Stewie Griffin	1 rocky road	12/3/11	303 Chestnut St
Bart Simpson	3 mint chocolate chip, 2 strawberry	12/5/11	202 School Ln
Hank Hill	2 coffee, 3 vanilla	12/8/11	404 Canary Dr
Stewie Griffin	5 rocky road	12/10/11	303 Chestnut St

- **Is this table schema acceptable? Why or why not?**

* Courtesy of GEOG 868 – Penn State

Spatial Databases: An Example*

- Ok, let's address the order flavor issue…

Customer	Flavor1	Qty1	Flavor2	Qty2	Flavor3	Qty3	DeliveryDate	DeliveryAdd
Eric Cartman	vanilla	1	chocolate	2			12/1/11	101 Main St
Bart Simpson	chocolate	10	vanilla	10	strawberry	5	12/3/11	202 School Ln
Stewie Griffin	rocky road	1					12/3/11	303 Chestnut St
Bart Simpson	mint chocolate chip	3	strawberry	2			12/5/11	202 School Ln
Hank Hill	coffee	2	vanilla	3			12/8/11	404 Canary Dr
Stewie Griffin	rocky road	5					12/10/11	303 Chestnut St

- **Is this table schema acceptable? Why or why not?**

* Courtesy of GEOG 868 – Penn State

Spatial Databases: An Example*

- Third time's the charm?

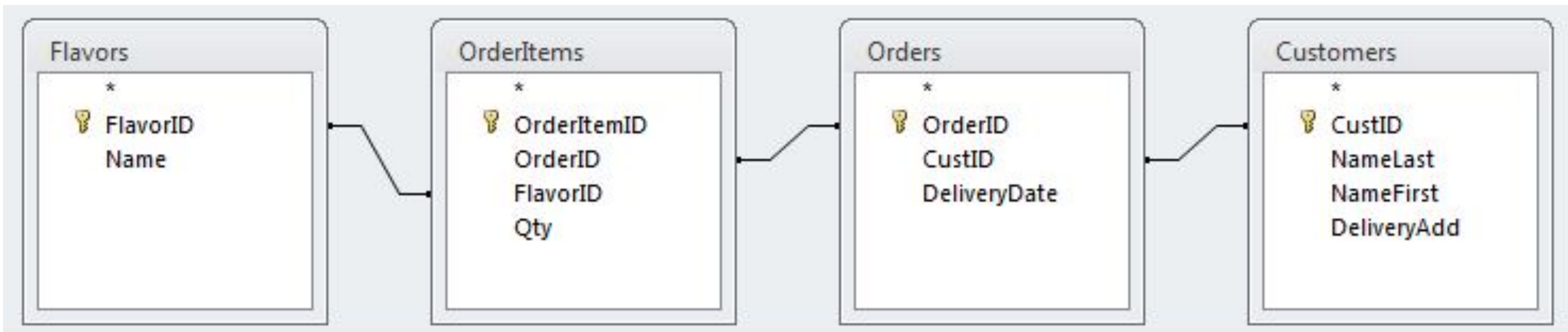
Customer	Flavor	Qty	DeliveryDate	DeliveryAdd
Eric Cartman	vanilla	1	12/1/11	101 Main St
Eric Cartman	chocolate	2	12/1/11	101 Main St
Bart Simpson	chocolate	10	12/3/11	202 School Ln
Bart Simpson	vanilla	10	12/3/11	202 School Ln
Bart Simpson	strawberry	5	12/3/11	202 School Ln
Stewie Griffin	rocky road	1	12/3/11	303 Chestnut St
Hank Hill	coffee	2	12/8/11	404 Canary Dr
Hank Hill	vanilla	3	12/8/11	404 Canary Dr
Stewie Griffin	rocky road	5	12/10/11	303 Chestnut St

- **Is this table schema acceptable? Why or why not?**

* Courtesy of GEOG 868 – Penn State

Spatial Databases: An Example*

- Let's split data into four entities: Customers, Flavors, Orders, and Order Items



* Courtesy of GEOG 868 – Penn State

Spatial Databases: An Example*

FlavorID	Name
1	vanilla
2	chocolate
3	strawberry
4	rocky road
5	mint chocolate chip
6	coffee

OrderID	CustID	DeliveryDate
1	1	12/1/11
2	2	12/3/11
3	3	12/3/11
4	2	12/5/11
5	4	12/8/11
6	3	12/10/11

OrderItemID	OrderID	FlavorID	Qty
1	1	1	1
2	1	2	2
3	2	2	10
4	2	1	10
5	2	3	5
6	3	4	1
7	4	5	3
8	4	3	2
9	5	6	2
10	5	1	3
11	6	4	5

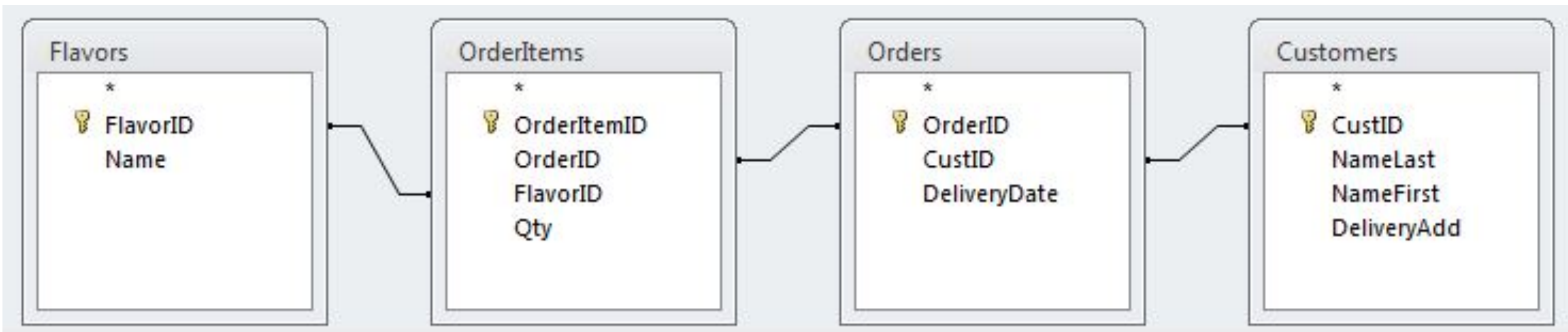
CustID	NameLast	NameFirst	DeliveryAdd
1	Cartman	Eric	101 Main St
2	Simpson	Bart	202 School Ln
3	Griffin	Stewie	303 Chestnut St
4	Hill	Hank	404 Canary Dr

PURDUE UNIVERSITY

* Courtesy of GEOG 868 – Penn State

*Spatial Databases: An Example**

- An order would leverage all four tables:



Ice Cream Flavors:
- **Flavor ID**
- Name of flavor

Order Details:
- Order Item ID
- **Order ID**
- **Flavor ID**
- Qty of ice cream flavor

Order Info:
- **Order ID**
- **Customer info**
- Delivery date

Customer Info:
- **Customer ID**
- Name
- Address

* Courtesy of GEOG 868 – Penn State