Spatial Database Management
GEP 664 / GEP 380
Class #5: Data processing

Frank Donnelly

Dept of EEGS, Lehman College CUNY

Spring 2019

Today's Topics

ETL

Extract : data is pulled from various sources
Transform : data is cleaned and harmonized to fit the target
Load : data is loaded into the target database

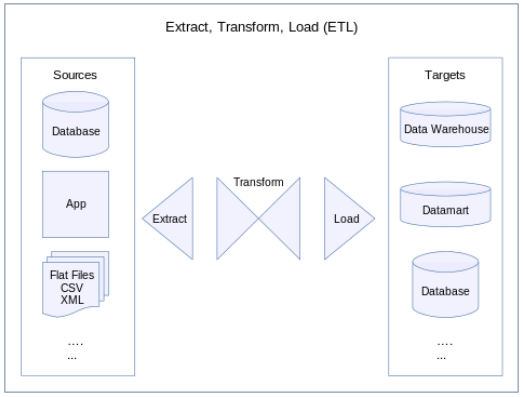


Image source:
https://commons.wikimedia.org/wiki/File:Extract,_Transform,_Load_Data_Flow_Diagram.svg

Data Interchange Formats

Machine-readable data has an inherent, basic structure so computers can parse it. Data files can fall into three categories:

Plain text : characters of readable data without any graphical representation. Can be viewed anywhere.

Rich text : characters of readable data with mark-up to denote style, organization, and semantics. Can be viewed anywhere, but may need specific software to compile for proper display and processing.

Binary file : data is encoded as binary objects. Requires specific software to view and interpret.

Plain Text

- Simple format
- Characters are encoded using a standard
  - UTF-8, LATIN-1, Windows-1252
- Can be viewed in any operating system
- Can be viewed in any software (text editors, spreadsheets, word processors, web browsers, etc.)
- Good format for digital preservation

Image source: https://commons.wikimedia.org/wiki/Category:ASCII#/media/File:ASCII-Table.svg

ASCII TABLE

Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	[NULL]
1	1	1	1	[START OF HEADING]
2	2	10	2	[START OF TEXT]
3	3	11	3	[END OF TEXT]
4	4	100	4	[END OF TRANSMISSION]
5	5	101	5	[ENQUIRY]
6	6	110	6	[ACKNOWLEDGE]
7	7	111	7	[BELL]
8	8	1000	10	[BACKSPACE]
9	9	1001	11	[HORIZONTAL TAB]
10	A	1010	12	[LINE FEED]
11	B	1011	13	[VERTICAL TAB]
12	C	1100	14	[FORM FEED]
13	D	1101	15	[CARRIAGE RETURN]
14	E	1110	16	[SHIFT OUT]
15	F	1111	17	[SHIFT IN]
16	10	10000	20	[DATA LINK ESCAPE]
17	11	10001	21	[DEVICE CONTROL 1]
18	12	10010	22	[DEVICE CONTROL 2]
19	13	10011	23	[DEVICE CONTROL 3]
20	14	10100	24	[DEVICE CONTROL 4]
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]
22	16	10110	26	[SYNCHRONOUS IDLE]
23	17	10111	27	[ENG OF TRANS. BLOCK]
24	18	11000	30	[CANCEL]
25	19	11001	31	[END OF MEDIUM]
26	1A	11010	32	[SUBSTITUTE]
27	1B	11011	33	[ESCAPE]
28	1C	11100	34	[FILE SEPARATOR]
29	1D	11101	35	[GROUP SEPARATOR]
30	1E	11110	36	[RECORD SEPARATOR]
31	1F	11111	37	[UNIT SEPARATOR]
32	20	100000	40	[SPACE]
33	21	100001	41	!
34	22	100010	42	"
35	23	100011	43	#
36	24	100100	44	$
37	25	100101	45	%
38	26	100110	46	&
39	27	100111	47	'
40	28	101000	50	(
41	29	101001	51	)
42	2A	101010	52	*
43	2B	101011	53	+
44	2C	101100	54	,
45	2D	101101	55	-
46	2E	101110	56	.
47	2F	101111	57	/
48	30	110000	60	0
49	31	110001	61	1
50	32	110010	62	2
51	33	110011	63	3
52	34	110100	64	4
53	35	110101	65	5
54	36	110110	66	6
55	37	110111	67	7
56	38	111000	70	8
57	39	111001	71	9
58	3A	111010	72	:
59	3B	111011	73	;
60	3C	111100	74	<
61	3D	111101	75	=
62	3E	111110	76	>
63	3F	111111	77	?
64	40	1000000	100	@
65	41	1000001	101	A
66	42	1000010	102	B
67	43	1000011	103	C
68	44	1000100	104	D
69	45	1000101	105	E
70	46	1000110	106	F
71	47	1000111	107	G
72	48	1001000	110	H
73	49	1001001	111	I
74	4A	1001010	112	J
75	4B	1001011	113	K
76	4C	1001100	114	L
77	4D	1001101	115	M
78	4E	1001110	116	N
79	4F	1001111	117	O
80	50	1010000	120	P
81	51	1010001	121	Q
82	52	1010010	122	R
83	53	1010011	123	S
84	54	1010100	124	T
85	55	1010101	125	U
86	56	1010110	126	V
87	57	1010111	127	W
88	58	1011000	130	X
89	59	1011001	131	Y
90	5A	1011010	132	Z
91	5B	1011011	133	[
92	5C	1011100	134	\
93	5D	1011101	135	]
94	5E	1011110	136	^
95	5F	1011111	137	_
96	60	1100000	140	`
97	61	1100001	141	a
98	62	1100010	142	b
99	63	1100011	143	c
100	64	1100100	144	d
101	65	1100101	145	e
102	66	1100110	146	f
103	67	1100111	147	g
104	68	1101000	150	h
105	69	1101001	151	i
106	6A	1101010	152	j
107	6B	1101011	153	k
108	6C	1101100	154	l
109	6D	1101101	155	m
110	6E	1101110	156	n
111	6F	1101111	157	o
112	70	1110000	160	p
113	71	1110001	161	q
114	72	1110010	162	r
115	73	1110011	163	s
116	74	1110100	164	t
117	75	1110101	165	u
118	76	1110110	166	v
119	77	1110111	167	w
120	78	1111000	170	x
121	79	1111001	171	y
122	7A	1111010	172	z
123	7B	1111011	173	{
124	7C	1111100	174	|
125	7D	1111101	175	}
126	7E	1111110	176	~
127	7F	1111111	177	[DEL]

Fixed-width

Plain Text

Each row is a record. All values have a set length and must be parsed using their starting positions. Header row possible but uncommon. File extensions vary.

```
DE10001KENT COUNTY          800 162310
DE10003NEW CASTLE COUNTY    494 538479
DE10005SUSSEX COUNTY        1196197145
```

Delimited

Plain Text

Each row is a record. All values separated by a delimiter. Most common are commas, tabs, and pipes. Header row optional. File extensions vary, commonly .csv, .txt, .tsv.

```
USPS,STATE,COUNTY,NAME,AREA,POP2010
DE,10,001,KENT COUNTY,800,162310
DE,10,003,NEW CASTLE COUNTY,494,538479
DE,10,005,SUSSEX COUNTY,1196,197145
```

Values may be quoted to preserve text and to escape characters. Quotes can be applied to all text, or only to text that must be escaped.

```
DE,10,003,''NEW CASTLE COUNTY, DELAWARE'',494,538479
```

Rich Text

AKA formatted text. Same as plain text, except:

- Characters are marked-up to add meaning for organization, description, or display
- Mark-up can include <tags >around values </tag >
- Or punctuation that describes relationships 'element' : {value1, value2}
- While richer and readily readable, complexity requires tools for processing

XML

Rich Text

Hierarchical, nested-structure where mark-up supplies meaning to values. XML is the general structure for defining limitless of vocabularies (XHTML, KML, GPL...)

```
<record>
    <state usps='DE'>
        <code>10</code>
        <county>
            <code>003</code>
            <name>New Castle County</name>
            <area type='sqmi'>494</area>
            <population yr=2010>538479</population>
        </county>
    </state>
</record>
```

JSON

Rich Text

JavaScript Object Notation, data stored as attribute-value pairs that allows for nesting and implicit type notation.

```
{
    "state": {
        "usps": "DE",
        "code": "10",
        "county": {
            "code": "003",
            "name": "New Castle County"
            "area": 494,
            "pop2010" : 538479
        }
    }
}
```

SQL

Rich Text

File contains instructions and the data for creating and populating tables in a database.

```
BEGIN;
CREATE TABLE de_counties (
usps varchar(2),
state varchar(2),
county varchar(3),
name text,
area numeric (5,1),
pop2010 integer,
CONSTRAINT pkde PRIMARY KEY (state,county)
);

INSERT INTO 'de_counties' VALUES ('DE','10','001','KENT COUNTY',800,162310);
INSERT INTO 'de_counties' VALUES ('DE','10','003','NEW CASTLE COUNTY',494,
538479);
INSERT INTO 'de_counties' VALUES ('DE','10','005','SUSSEX
COUNTY',1196,197145);
COMMIT;
```

Spreadsheets

Binary

- These formats are not encoded in plain text and require specific software to open and manipulate
- Common data formats include spreadsheets (Excel .xls and .xlsx, Calc .ods) and dBase files .dbf
- Import / export support for these formats varies with different database packages
- Spreadsheet programs are capable of saving files as delimited text

Today's Topics

COPY command

The SQL standard does NOT include commands for importing and exporting data. Implementations are database-specific

- The PostgreSQL COPY command is non-standard SQL for importing delimited text data
- SQL COPY works on localhost but NOT on client machines connecting to a remote server
- The psql \copy command is a non-SQL command that works locally or remotely
- The pgAdmin interface also has user-friendly GUI tools for import and export

COPY Caveats

Regardless of which approach you use, you *must create a table in the database first*, and then copy data into it. The order of the columns and data matters. Approaches:

1. Create the perfect table in the database. Clean your data up in an external program to the maximum extent. Load it in.
2. Create a staging table in the database. Clean your data up in an external program to the minimum extent. Load it in. Do more cleaning and transformation in the staging table. Create the perfect table, load from the staging table, delete staging table.

Import / Export data with COPY

Import: Comma-delimited with header row

```
COPY nyc.weather_staging
FROM 'C:\user\weatherdata\newobservs.csv' WITH
   DELIMITER AS ',' CSV HEADER;
```

Export: Tab-delimited with header row, and optional command to put quotes around id column (to preserve as text)

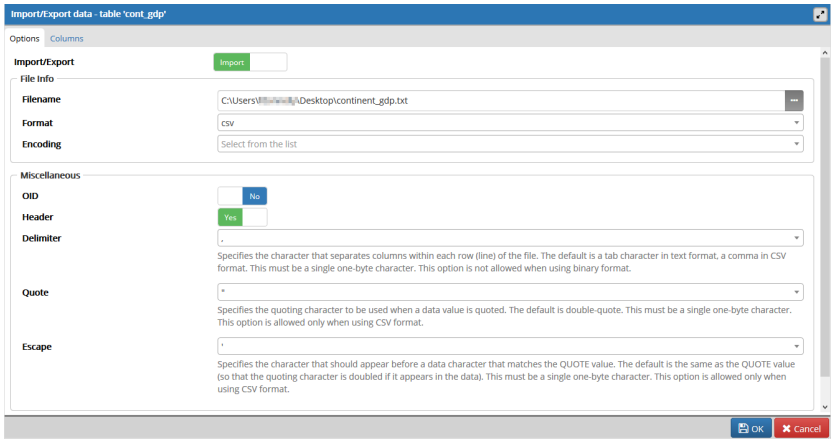
```
COPY nyc.weather_staging
TO 'C:\user\weatherdata\newobservs.txt' WITH DELIMITER
   AS '\t' CSV HEADER FORCE QUOTE station_id;
```

Make sure to move data files to directory BEFORE launching pgadmin / psql; it won't detect files moved there after launch.

`https://www.postgresql.org/docs/10/sql-copy.html`

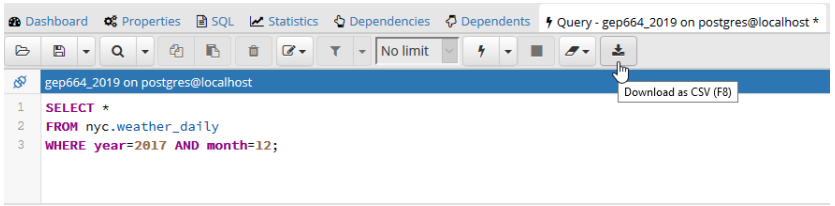
Import / Export Data with pgAdmin

Create empty table with structure. Right click on table, choose Import/Export. Must specify: Import/Export, Filename, Format, Header, Delimiter, Quote.



GUI for Exporting Query Results

In pgAdmin use the SQL window, write a statement, and when executing write the result directly to a text/csv format.



Change export options under File - Preferences - SQL Editor - CSV Output

Importing & Exporting Data into PostgreSQL

- Use the COPY command for delimited text files
- Use the pgAdmin GUI to load text or binary files
- With PostGIS use the shapefile loader plugin for shapefiles and DBFs
- QGIS DB Manager can also be used for loading shapefiles
- Use backup and restore for SQL dump files
- For tiny datasets use manual INSERT statements

Today's Topics

Data Formats

Data Import and Export

Data Cleaning

Next Class

Cleaning Goals

- End up with rows of data and columns of attributes of a single data type
- Eliminate presentational aspects (headers, footers, free text)
- Eliminate totals and subtotals
- Assign unique identifiers
- Separate footnotes from data
- Shuffle the order of columns
- Combine or separate values
- Standardize values
- Summarize (aggregate) and transform (pivot) data

Spreadsheets vs Databases

Spreadsheets and databases have similar functions for cleaning and re-organizing data, but for databases the data must be well-structured in order to import it.

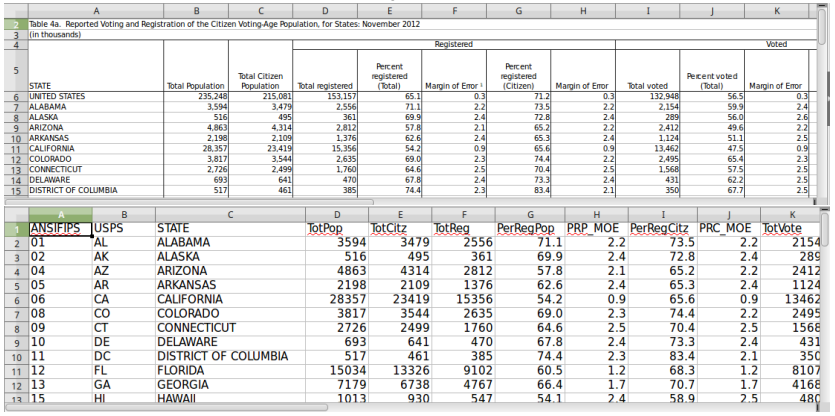


Table 4a. Reported Voting and Registration of the Citizen Voting-Age Population, for States: November 2012
(in thousands)

STATE	Total Population	Total Citizen Population	Registered: Total registered	Registered: Percent registered (Total)	Registered: Margin of Error	Registered: Percent registered (Citizen)	Registered: Margin of Error	Voted: Total voted	Voted: Percent voted (Total)	Voted: Margin of Error
UNITED STATES	235,248	215,081	153,157	65.1	0.3	71.2	0.3	132,948	56.5	0.3
ALABAMA	3,594	3,479	2,556	71.1	2.2	73.5	2.2	2,154	59.9	2.4
ALASKA	516	495	361	69.9	2.4	72.8	2.4	289	56.0	2.6
ARIZONA	4,863	4,314	2,812	57.8	2.1	65.2	2.2	2,412	49.6	2.2
ARKANSAS	2,198	2,109	1,376	62.6	2.4	65.3	2.4	1,124	51.1	2.5
CALIFORNIA	28,357	23,419	15,356	54.2	0.9	65.6	0.9	13,462	47.5	0.9
COLORADO	3,817	3,544	2,635	69.0	2.3	74.4	2.2	2,495	65.4	2.3
CONNECTICUT	2,726	2,499	1,760	64.6	2.5	70.4	2.5	1,568	57.5	2.5
DELAWARE	693	641	470	67.8	2.4	73.3	2.4	431	62.2	2.5
DISTRICT OF COLUMBIA	517	461	385	74.4	2.3	83.4	2.1	350	67.7	2.5

ANSIFIPS	USPS	STATE	TotPop	TotCitz	TotReg	PerRegPop	PRP_MOE	PerRegCitz	PRC_MOE	TotVote
01	AL	ALABAMA	3594	3479	2556	71.1	2.2	73.5	2.2	2154
02	AK	ALASKA	516	495	361	69.9	2.4	72.8	2.4	289
04	AZ	ARIZONA	4863	4314	2812	57.8	2.1	65.2	2.2	2412
05	AR	ARKANSAS	2198	2109	1376	62.6	2.4	65.3	2.4	1124
06	CA	CALIFORNIA	28357	23419	15356	54.2	0.9	65.6	0.9	13462
08	CO	COLORADO	3817	3544	2635	69.0	2.3	74.4	2.2	2495
09	CT	CONNECTICUT	2726	2499	1760	64.6	2.5	70.4	2.5	1568
10	DE	DELAWARE	693	641	470	67.8	2.4	73.3	2.4	431
11	DC	DISTRICT OF COLUMBIA	517	461	385	74.4	2.3	83.4	2.1	350
12	FL	FLORIDA	15034	13326	9102	60.5	1.2	68.3	1.2	8107
13	GA	GEORGIA	7179	6738	4767	66.4	1.7	70.7	1.7	4168
15	HI	HAWAII	1013	930	547	54.1	2.4	58.9	2.5	480

Excel is Lousy with CSV Files

You should NEVER doubleclick on a CSV file to open it in Excel - Excel makes bad assumptions about formatting data. This:

```
USPS,STATE,COUNTY,NAME
DE,10,001,KENT COUNTY
DE,10,003,NEW CASTLE COUNTY
DE,10,005,SUSSEX COUNTY
```

Becomes this:

USPS	STATE	COUNTY	NAME
DE	10	1	KENT COUNTY
DE	10	3	NEW CASTLE COUNTY
DE	10	5	SUSSEX COUNTY

Opening CSVs in Spreadsheets

To preserve ID codes or other values as text, to prevent loss of leading zeros and value transformation:

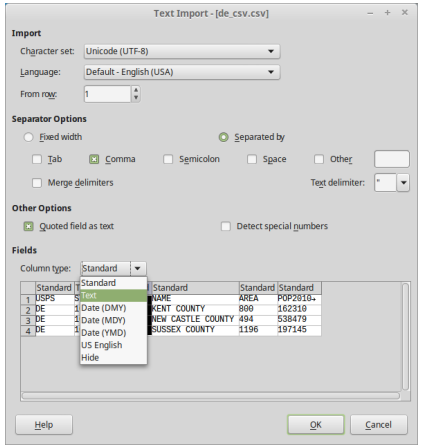
MS Excel

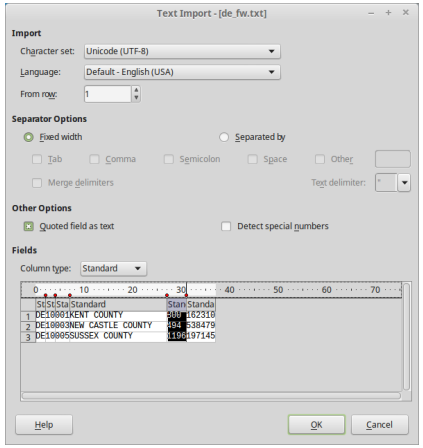
Open the software to a blank project. Go to Data - Import From Text, and for each ID column set the column type to text, then open.

Libre Office Calc

Is smarter. Doubleclick on files to open them; for delimited text you will automatically be prompted to designate columns as text. Can also select File - Open from within the software.

Importing Delimited and Fixed Width





Common String Operations

In spreadsheet (ss) and database (db):

Transform data types: TEXT and VALUES (ss), CAST (db)

Combine multiple: CONCATENATE (ss), double pipes col1 | | col2 (db)

Split strings into multiple values: Text to values tool (ss), SPLIT_PART (db)

Return subset based on position: LEFT, RIGHT, MID (ss), LEFT, RIGHT, SUBSTR (db)

Replace text with something else: Find and replace feature (ss), REPLACE or SET - UPDATE (db)

Remove trailing and leading white space: TRIM (both)

Convert cases: UPPER, LOWER (both)

Capitalize: PROPER (ss), INITCAP (db)

Spreadsheets and Logic

In a spreadsheet use the IF statement to make decisions, and parentheses to nest them

state	county	scode	ccode
California	Alpine	6	3
California	San Mateo	6	81
Delaware	New Castle	10	3
Pennsylvania	Philadelphia	42	101

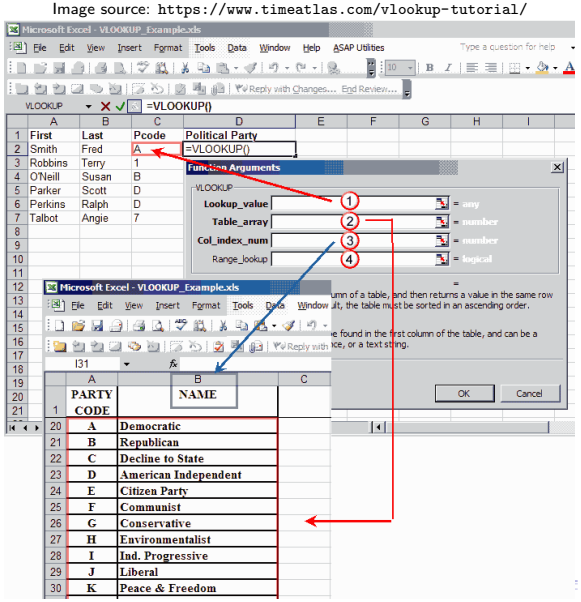
```
=IF(len(c2)=1,concatenate(''0'',c2),c2)
=IF(len(d2)=3,d2,IF(len(d2)=2,concatenate(''0'',d2),
   concatenate(''00'',d2)))
```

state	county	scode	ccode
California	Alpine	06	003
California	San Mateo	06	081
Delaware	New Castle	10	003
Pennsylvania	Philadelphia	42	101

Assigning Codes and Groups

VLOOKUP

Image source: https://www.timeatlas.com/vlookup-tutorial/



Shift Values to Columns

Pivot Table

uid	address	city	country	iso	stat_code	stat_text	accept
1	15 Winterlev [illegible]	Point Lonsdale	Australia	AUS	13	Conditional Accept	A
2	Eastern [illegible]	Dhaka	Bangladesh	BGD	8	Accept MS INTL	A
3	20/11, Shu [illegible]	Dhaka	Bangladesh	BGD	8	Accept MS INTL	A
4	House#41 [illegible]	Dhaka	Bangladesh	BGD	10	Deny	D
5	Apt# F4, [illegible]	Dhaka	Bangladesh	BGD	10	Deny	D
6	Denisovskaya [illegible]	Minsk	Belarus	BLR	13	Conditional Accept	A
7	44, Padre [illegible]	Botucatu	Brazil	BRA	17	Deny Incomplete	D
8	Des. Otavio [illegible]	Curitiba	Brazil	BRA	17	Deny Incomplete	D
9	70 Theodore	Thornhill	Canada	CAN	11	Ineligible	D
10	1773 Grosvenor [illegible]	Mississauga	Canada	CAN	2	Accept FT MBA INTL	A
11	65 Forest [illegible]	Toronto	Canada	CAN	10	Deny	D
12	160 George [illegible]	Sarnia	Canada	CAN	8	Accept MS INTL	A
13	6328 Larkin [illegible]	Vancouver	Canada	CAN	8	Accept MS INTL	A
14	10740 Tre [illegible]	Richmond	Canada	CAN	8	Accept MS INTL	A
15	7-16-4 Chest [illegible]	Chongqing	China	CHN	10	Deny	D
16	No.5 Zhang [illegible]	Guangzhou	China	CHN	8	Accept MS INTL	A
17	XinHe Street	WenZhou	China	CHN	8	Accept MS INTL	A
18	Room 404 [illegible]	Shaoxing	China	CHN	8	Accept MS INTL	A
19	19-106 Unde [illegible]	Shanghai	China	CHN	12	Conditional Accept MS UG	A
20	Room 510, [illegible]	Beijing	China	CHN	10	Deny	D
21	2F, Interc [illegible]	Beijing	China	CHN	8	Accept MS INTL	A
22	2F,Interchan [illegible]	Beijing	China	CHN	10	Deny	D
23	No.2 Yard, [illegible]	Beijing	China	CHN	13	Conditional Accept	A
24	36-1-502, [illegible]	Beijing	China	CHN	8	Accept MS INTL	A
25	Room 1904, [illegible]	Beijing	China	CHN	10	Deny	D

Count - uid	accept		
country	iso	A	D	Total Result
Australia	AUS	1		1
Bangladesh	BGD	2	2	4
Belarus	BLR	1		1
Brazil	BRA		2	2
Canada	CAN	4	2	6
China	CHN	129	63	192
Denmark	DNK		2	2
Dominica	DMA	1		1
Germany	DEU	2		2
Ghana	GHA	1		1

Creating Aggregates

- Possible in a spreadsheet with formulas (SUMIF, COUNTIF), but awkward
- Can be done with pivot table
- (In Excel you can also reverse pivot to split up values)
- May be simplest to create in the database with GROUP BY

Cleaning Exercise - Copper Data

- Data on mines and smelters from the USGS Data Catalog
- Data on imports and exports from UNdata (Comtrade)

Image source: https://en.wikipedia.org/wiki/Copper

OpenRefine

Free, open source tool that's great for cleaning messy data and handling larger datasets that spreadsheets can't manage.

- `http://openrefine.org/`
- Simple binary executable file
- Runs in any web browser
- Takes spreadsheets and delimited text as input
- Use text facet tool to standardize values and fix mis-spellings
- Use numeric facet tool to identify text buried in numeric columns

Today's Topics

Data Formats

Data Import and Export

Data Cleaning

Next Class

Due Next Class

The following are due at the beginning of our next class:

Assignment #5

Posted on the course website

Readings for Class #6

Listed in the syllabus, in the *PostGIS In Action* book

PostGIS in Action Readings

READ preface, Chapters 1, 2, & 5

But in these chapters you can skim or skip the following:

- 1.4.5: Read, but don't install OpenJUMP
- 2.2.7: Polyhedrals and Tins
- 2.2.8 Curved geometries
- 5.2 OpenJUMP
- 5.4 & 5.5 uDig & gvSIG