SQL VIGNETTES

JUNE 8, 2017

Felix Kabo
fkabo@umich.edu

INSTITUTE FOR SOCIAL RESEARCH
UNIVERSITY OF MICHIGAN
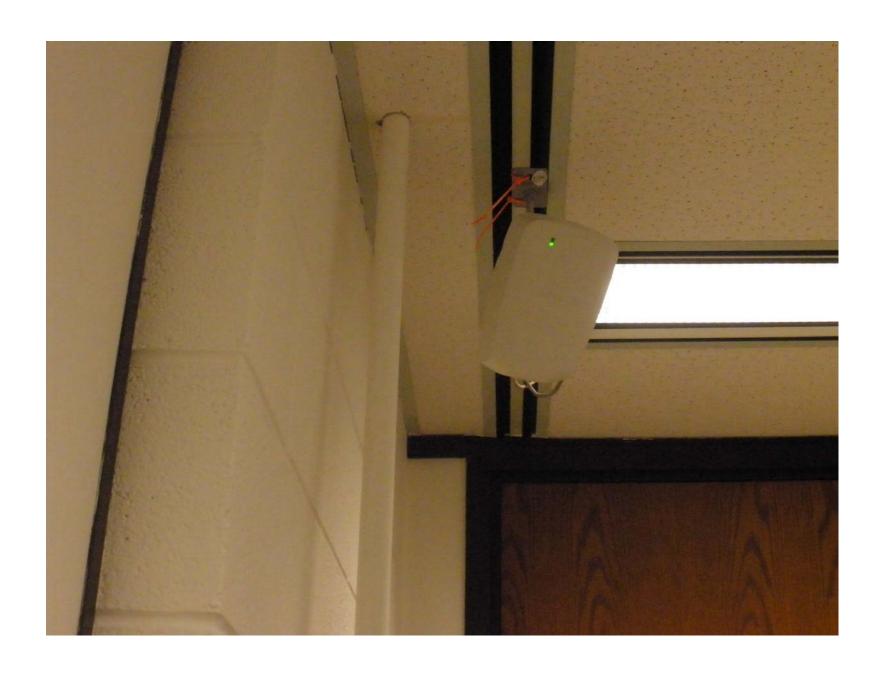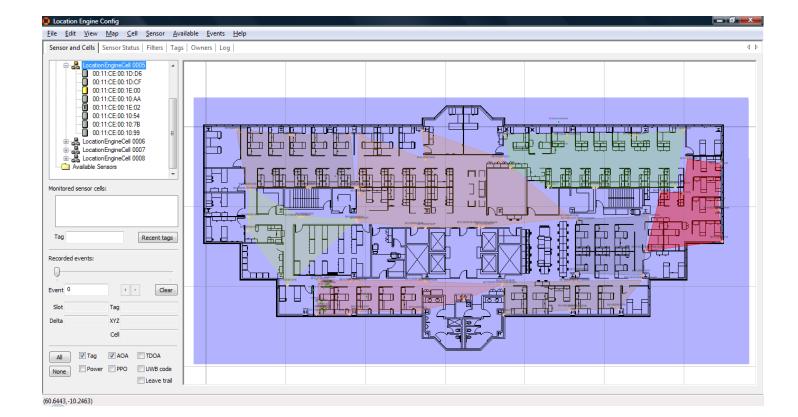
1

LOCATION TRACKING

- Real-time data were collected using ultrawide-band (UWB) location-tracking technology

- Study conducted at BRX, subunit of large global manufacturing enterprise in the Midwest of the U.S.

- Over 35 million rows of data (location tracking points) for study period

- Kabo, F. W. (2016). "A Model of Potential Encounters in the Workplace: The Relationships of Homophily, Spatial Distance, Organizational Structure, and Perceived Networks." *Environment & Behavior*.

tag	eventdatetime	x	y	z
TID78082	9/16/2009 15:28:53	13.58	-21.02	1.27
TID78082	9/16/2009 15:28:53	13.58	-21.02	1.27
TID78082	9/16/2009 15:28:53	13.58	-21.02	1.27
TID78082	9/16/2009 15:28:53	13.58	-21.02	1.27
TID78082	9/16/2009 15:28:53	13.58	-21.02	1.27
TID78084	9/24/2009 14:55:53	19.36	-21.01	0.73
TID78084	9/24/2009 14:55:53	19.39	-21.08	0.78
TID78084	9/24/2009 14:55:54	19.41	-21.13	0.82
TID78084	9/24/2009 14:55:54	19.43	-21.22	0.86
TID78084	9/24/2009 14:55:54	19.43	-21.32	0.9
TID78084	9/24/2009 14:55:54	19.44	-21.4	0.94
TID78084	9/24/2009 14:55:55	19.44	-21.47	0.97
TID78084	9/24/2009 14:55:51	19.35	-20.89	0.67
TID78084	9/24/2009 14:55:51	19.36	-20.91	0.69
TID78084	9/24/2009 14:55:51	19.36	-20.93	0.69
TID78084	9/24/2009 14:55:51	19.35	-20.95	0.7
TID78084	9/24/2009 14:55:52	19.33	-20.95	0.69
TID78084	9/24/2009 14:55:52	19.33	-20.94	0.69
TID78084	9/24/2009 14:56:08	19.45	-21.64	1.12
TID78084	9/24/2009 14:56:08	19.45	-21.58	1.09

WHICH WAS THE BUSIEST DAY?

```sql
use Location_Tracking;
go

select * from data_formatted;
go
--35,027,838 rows

--get/construct parts of date from "datetime"
select datename(dy, eventdatetime) as day_of_year,
datename(month, eventdatetime) as month_of_event,
datename(day, eventdatetime) as day_of_event,
count(eventdatetime) as num_events
from data_formatted
group by datename(dy, eventdatetime), datename(month, eventdatetime),
datename(day, eventdatetime)
order by count(eventdatetime) desc;◄-------------------
go
--71 rows
```

day_of_year	month_of_event	day_of_event	num_events
285	October	12	1,425,544
274	October	1	1,212,382
299	October	26	1,203,120
278	October	5	1,200,886
281	October	8	1,191,689
271	September	28	1,190,308
308	November	4	1,151,409
279	October	6	1,130,660
287	October	14	1,111,939
306	November	2	1,090,669
282	October	9	1,087,659
280	October	7	1,082,942
289	October	16	1,053,833
286	October	13	1,047,012
272	September	29	1,044,482
307	November	3	1,041,883
292	October	19	1,027,397
303	October	30	1,013,813
268	September	25	998,798
313	November	9	984,123

2

RULE-BREAKING & POSITIVE REWARDS

"Cartesian Products usually don't provide useful information and often result in mistakes that can hurt your database developer career. Learn to spot Cartesian Joins and banish them from your SELECT queries **forever**."

http://www.databasejournal.com/features/mysql/article.php/3901221/Identifying-and-Eliminating-the-Dreaded-Cartesian-Product.htm
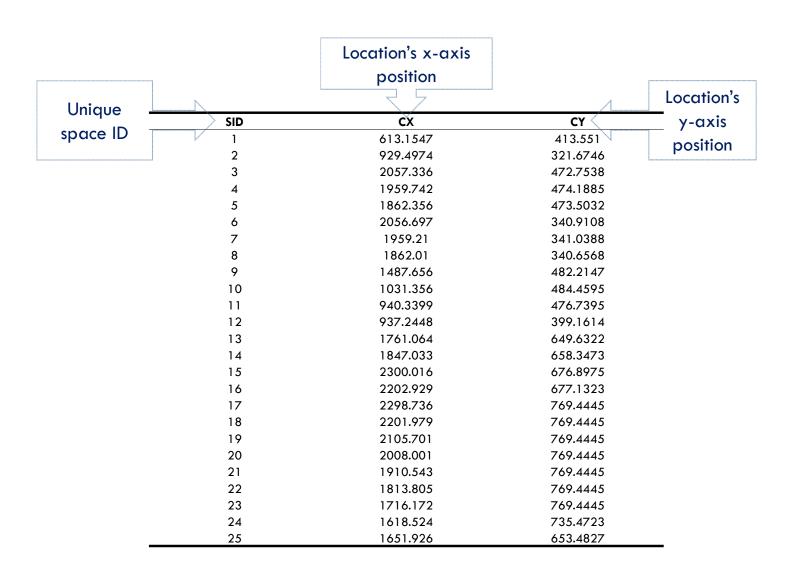
BEHOLD…THE CROSS-JOIN

Cross-join is the combination of each row from the first table with each row from the second table…all possible combinations!!

Would you ever need to do this?

DISTANCE FROM PERSON A TO PERSON B

An office layout that has been converted into a convex spatial network (dark interior areas are circulation or service spaces).

	Unique space ID		Location's x-axis position		Location's y-axis position

SID	CX	CY
1	613.1547	413.551
2	929.4974	321.6746
3	2057.336	472.7538
4	1959.742	474.1885
5	1862.356	473.5032
6	2056.697	340.9108
7	1959.21	341.0388
8	1862.01	340.6568
9	1487.656	482.2147
10	1031.356	484.4595
11	940.3399	476.7395
12	937.2448	399.1614
13	1761.064	649.6322
14	1847.033	658.3473
15	2300.016	676.8975
16	2202.929	677.1323
17	2298.736	769.4445
18	2201.979	769.4445
19	2105.701	769.4445
20	2008.001	769.4445
21	1910.543	769.4445
22	1813.805	769.4445
23	1716.172	769.4445
24	1618.524	735.4723
25	1651.926	653.4827

CREATE THE DATABASE & TABLE

```sql
--create the database without specifying any files
create database icos2017;
go

use icos2017;
go

create table locations(
sid int not null,
cx float null,
cy float null
);
go

bulk insert locations
from 'H:\ICOS 2017\Locations 06072017.txt'
with (firstrow=2);
--50 rows
```

COMBINE SELF- & CROSS-JOIN

```sql
--Combination of self- & cross-joins
--create two instances of same table in "from" clause...use aliases
select * from locations a
cross join locations b
where a.sid <> b.sid;
go
--2,450 rows


select a.sid as from_id, a.cx as from_x, a.cy
as from_y, b.sid as to_id, b.cx as to_x, b.cy
as to_y
into edgelist
from locations a
cross join locations b
where a.sid <> b.sid;
go
--2,450 rows
```

COMPUTE THE DISTANCE

```sql
alter table edgelist
add distance float;
go

update edgelist
set distance = sqrt((square(from_x-to_x) +
square(from_y-to_y)));
go
--2,450 rows

select from_id, to_id, distance
from edgelist
order by from_id, to_id;
go
--2,450 rows
```

DONE!

from_id	to_id	distance
1	2	329.41
1	3	1445.39
1	4	1347.95
1	5	1250.64
1	6	1445.37
1	7	1348.01
1	8	1250.98
1	9	877.19
1	10	424.17
1	11	333.23
1	12	324.41
1	13	1171.93
1	14	1257.93
1	15	1707.29
1	16	1611.48
1	17	1722.74
1	18	1628.20
1	19	1534.39
1	20	1439.53
1	21	1345.32