

# Pattern Patent

ETL

Northwestern

Previously on Pattern Patent ...

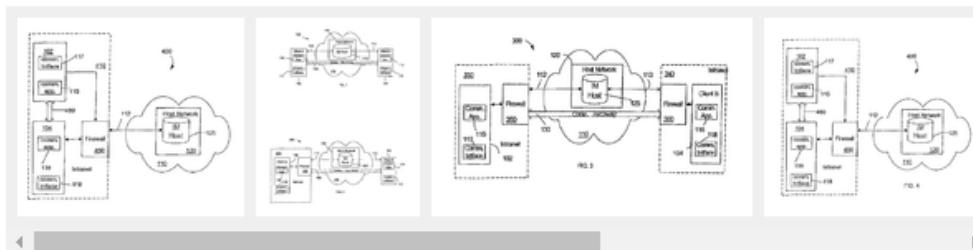
- *Would you afford the time cost?*  
Check out Hamlet App
- *Is your case likely to be patented?*  
Check out Hamlet App
- *Which attorney are you talking to?*  
Check out HesJustNotThatIntoYou App
- *Would you want to live around other inventors?*  
If yes, move to CA, Ivy League, RTP areas
- *Do you know the examiner's sweet spot?*  
25, 135, 837 rules

# Implicit population of access control lists

## Abstract

Communication applications may include lists of users with which a user of the application communicates. If two users of a communications application each include the other user on their user lists, an implicit trust may be established between the users. For example, if user A includes user B in her list and user B includes user A in his list, then it may be determined that each user knows and/or trusts the other user. As a result, a connection or communications pathway may be automatically created between the client devices of the users to facilitate communications between the users based on the implicit trust.

## Images (7)



## Classifications

H04L63/0272 Virtual private networks

<https://patents.google.com/patent/USRE45254>

USRE45254E1

United States

[Download PDF](#)

[Find Prior Art](#)

[Similar](#)

Inventor: [James A. Roskind](#)

Current Assignee : [Facebook Inc](#)

## Worldwide applications

2002 • US 2007 • US 2009 • US 2013 • US 2014 • US

Application US13/907,761 events ②

2002-12-31 • Priority to US10/334,142

2013-05-31 • Application filed by Facebook Inc

2014-11-18 • Publication of USRE45254E1

2014-11-18 • Application granted

Status • Active

2022-12-31 • Anticipated expiration

Description	Claims (33)	Show Dependent
<strong>CROSS-REFERENCE TO RELATED APPLICATIONS</strong>		
This application is a continuation application of and claims priority to U.S. application Ser. No. 11/782,461, now U.S. Pat. No. 7,490,238, filed Jul. 24, 2007, which is a continuation of U.S. application Ser. No. 10/334,142, now U.S. Pat. No. 7,263,614, filed on Dec. 31, 2002, the entire contents all of which are hereby incorporated by reference.		
<strong>TECHNICAL FIELD</strong>		
The following description relates to network communications.		
<strong>BACKGROUND</strong>		
With the rapid proliferation and affordability of computers, the Internet has become the communications medium of choice for many users. Although the Internet is a public medium, techniques have been developed for using the Internet to enable private communications between networks. One such private communications technique is used to enable instant messaging.		
Instant messaging allows users to rapidly communicate with other users of a communications network. Generally, client messaging software runs on a client A device <b>102</b> and provides a communications interface for entry of a message. The intended message recipient may be entered manually or may be selected from a user list, such as a "Buddy List"™ from America Online, Inc. Instant messaging may be used to communicate text messages, images, and sounds or voice.		
<strong>SUMMARY</strong>		
In one general aspect, messaging applications, systems, and methods may be used to automatically configure a communications pathway based on an implicit trust between users. Each user of a communications application may have a user list that identifies other users to which a message may be sent. If two users of the communications application each include the other user on their user lists, an implicit trust may be inferred between the users. For example, if user A includes user B in her user list and user B includes user A in his user list, then it may be inferred or determined that each user knows and/or implicitly trusts the other user. As a result, a connection or communications pathway may be automatically created and/or configured between the client devices of the users to facilitate communications between the users based on the implicit trust.		
	1. A method comprising:  logging, by a server, a first client device into a server;  logging, by the server, a second client device into the server;  accessing, using by the server, a first user list associated with a first user of the first client device;  accessing, using by the server, a second user list associated with a second user of the second client device, wherein the first user list and the second user list are maintained separately from one another;  analyzing, using by the server, the accessed first user list to determine whether an identifier of the second user is included in the first user list;  analyzing, using by the server, the accessed second user list to determine whether an identifier of the first user is included in the second user list; and  regulating, using by the server, a communications pathway communications between the first client device and the second client device based on both the determination of whether the identifier of the first user is included on the second user list and the determination of whether the identifier of the second user is included on the first user list.  ...	
	...	
	...	
	...	
	...	
	...	
	...	
	9. A system comprising:	

# Extract

## Data Source

<https://www.uspto.gov/learning-and-resources/electronic-data-products/patent-examination-research-dataset-public-pair>  
<https://www.uspto.gov/learning-and-resources/electronic-data-products/patent-claims-research-dataset>

## Data Files

Each of the files below can be downloaded in either Stata-14 (DTA) or CSV format.

Download a full set of data files (2014): [\[.dta format\]](#) (5.42 GB) | [\[.csv format\]](#) (4.33 GB)

Download a full set of data files (2015): [\[.dta format\]](#) (5.56 GB) | [\[.csv format\]](#) (4.99 GB)

Download a full set of data files (2016): [\[.dta format\]](#) (4.98 GB) | [\[.csv format\]](#) (4.36 GB)

Download a full set of data files (2017): [\[.dta format\]](#) (5.37 GB) | [\[.csv format\]](#) (4.8 GB)

Download a full set of data files (2019): [\[.dta format\]](#) (9.4 GB) | [\[.csv format\]](#) (7.87 GB)

Download individual data files (the direct download pages are here: [2014](#), [2015](#), [2016](#), [2017](#), [2019](#)).

File Name	2014		2016		2017		2019	
application_data	DTA 1.53 GB	CSV 585 MB	DTA 1.1 GB	CSV 681	DTA 1.01 MB	CSV 657	DTA 1.03 GB	CSV 774 MB
all_inventors	DTA 229 MB	CSV 225 MB	DTA 348 MB	CSV 347	DTA 485 MB	CSV 499	DTA 427 MB	CSV 417 MB
transactions	DTA 2.55 GB	CSV 2.45 GB	DTA 2.02 GB	CSV 1.91 GB	DTA 2.21 GB	CSV 2.09 GB	DTA 2.56 GB	CSV 1.65 GB
event_codes	DTA 75 KB	CSV 21.2 KB	DTA 36.4 KB	CSV 22.8 KB	DTA 37.8 KB	CSV 23.5 KB	DTA 40.7 KB	CSV 23.3 KB
status_codes	DTA 8.56 KB	CSV 3.53 KB	DTA 5.87 KB	CSV 3.74 KB	DTA 6.01 KB	CSV 3.74 KB	No data	No data
continuity_parents	DTA 49.9 MB	CSV 48.7 MB	DTA 73.2 MB	CSV 58 MB	DTA 79 MB	CSV 63.1 MB	DTA 102 MB	CSV 80.2 MB

## Data Files

Download full set of 2014 data files [\[.dta format\]](#) (11.2 GB) | [\[.csv format\]](#) (9.32 GB)

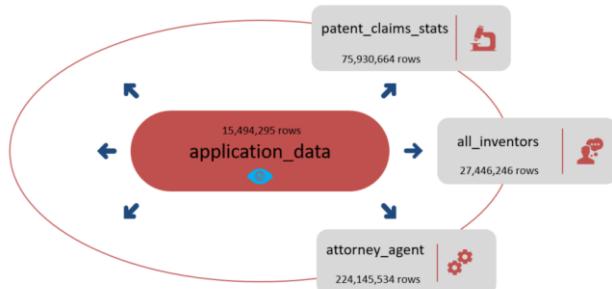
Download individual data files:

File Name	2014	
	DTA	CSV
patent_claims_fulltext	5.45 GB	4.41 GB
patent_claims_stats	821 MB	452 MB
patent_document_stats	119 MB	90.3 MB
ppgpub_claims_fulltext	4.21 GB	3.79 GB
ppgpub_claims_stats	570 MB	530 MB
ppgpub_document_stats	81.6 MB	75 MB

The direct download page is [here](#).

Note: The DTA (Stata dataset) files are saved in the Stata-13 data file format.

Note: The code used to parse the Patent Application Full-Tout and Distant Grant Full-Tout files and



# Transform

Data cleaning and transformation (cleaning, joining, filtering, grouping, merging, aggregating, etc) are performed in Python

```
[13]: p_word_ct = p.agg({"word_ct": "sum"})  
p_word_ct.tail(5)
```

```
[13]:          word_ct  
patent_number  
RE45249    1665  
RE45250    1138  
RE45251    2154  
RE45252    4597  
RE45254    2825
```

```
[14]: p_char_ct = p.agg({"char_ct": "sum"})  
p_char_ct.tail(5)
```

```
[14]:          char_ct  
patent_number  
RE45249    10245  
RE45250    7417  
RE45251    13356  
RE45252    29228  
DEAEDEA    17750
```

# Load

## PostgreSQL/Beaver

```
--ALTER TABLE application_cleaned_all ADD CONSTRAINT fk_application_cleaned_all_uspc_class FOREIGN KEY( uspc_class )
REFERENCES "uspc_class_all" ("uspc_class");

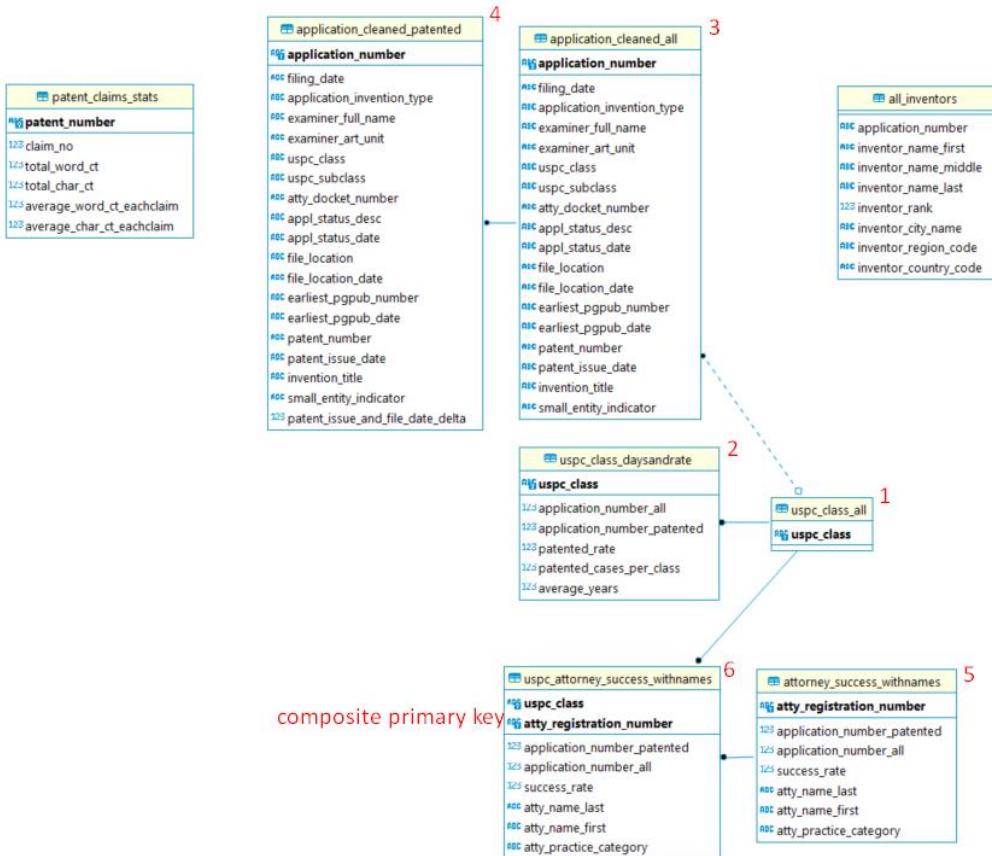
--ALTER TABLE "uspc_class_daysandrate" ADD CONSTRAINT "fk_uspc_class_daysandrate_uspc_class" FOREIGN KEY("uspc_class")
REFERENCES "uspc_class_all" ("uspc_class");

--ALTER TABLE "uspc_attorney_success_withnames" ADD CONSTRAINT "fk_uspc_attorney_success_withnames_uspc_class" FOREIGN KEY("uspc_class")
REFERENCES "uspc_class_all" ("uspc_class");

--ALTER TABLE "uspc_attorney_success_withnames" ADD CONSTRAINT "fk_uspc_attorney_success_withnames_atty_registration_number" FOREIGN KEY("atty_registration_number")
REFERENCES "attorney_success_withnames" ("atty_registration_number");

-- --
--SELECT * FROM patent_claims_stats;
SELECT * FROM all_inventors;
--SELECT * FROM uspc_class_all where uspc_class = '060';
SELECT * FROM uspc_class_all;
SELECT * FROM application_cleaned_patented;
SELECT * FROM application_cleaned_all;
SELECT * FROM uspc_class_daysandrate;
SELECT * FROM attorney_success_withnames;
SELECT * FROM uspc_attorney_success_withnames;
SELECT * FROM patent_claims_stats;
```

# Load



# Load

DBBeaver 21.0.4 - <postgres> mol.sql

File Edit Navigate Search SQL Editor Database Window Help

ETL\_USPTO public <postgres> mol.sql fk\_application\_cleaned\_patented\_application\_number

Enter a part of table name here

postgres - localhost:5432

SQL Auto Commit Rollback

ALTER TABLE "application\_cleaned\_all" ADD CONSTRAINT "fk\_application\_cleaned\_all\_uspc\_class" FOREIGN KEY("uspc\_class") REFERENCES "uspc\_class\_all" ("uspc\_class");

ALTER TABLE "uspc\_class\_daysandrate" ADD CONSTRAINT "fk\_uspc\_class\_daysandrate\_uspc\_class" FOREIGN KEY("uspc\_class") REFERENCES "uspc\_class\_all" ("uspc\_class");

ALTER TABLE "uspc\_attorney\_success\_withnames" ADD CONSTRAINT "fk\_uspc\_attorney\_success\_withnames\_uspc\_class" FOREIGN KEY("uspc\_class") REFERENCES "uspc\_class\_all" ("uspc\_class");

ALTER TABLE "uspc\_attorney\_success\_withnames" ADD CONSTRAINT "fk\_uspc\_attorney\_success\_withnames\_atty\_registration\_number" FOREIGN KEY("atty\_registration\_number") REFERENCES "attorney\_success\_withnames" ("atty\_registration\_number");

--

--SELECT \* FROM patent\_claims\_stats;

SELECT \* FROM all\_inventors;

SELECT \* FROM uspc\_class\_all WHERE uspc\_class = '060';

SELECT \* FROM uspc\_class\_all;

SELECT \* FROM application\_cleaned\_patented;

SELECT \* FROM application\_cleaned\_all;

SELECT \* FROM uspc\_class\_daysandrate;

SELECT \* FROM attorney\_success\_withnames;

SELECT \* FROM uspc\_attorney\_success\_withnames;

SELECT \* FROM patent\_claims\_stats;

uspc\_class\_all 1

uspc\_class

1	nan
2	380
3	000
4	419
5	179
6	999
7	423
8	178
9	376
10	075
11	114
12	102
13	095
14	367
15	327
16	073
17	057

Record

Save Cancel Script | 200+ Rows: 1

CST en Writable Smart Insert 175: 1 [29] Sel: 29 | 1

200 row(s) fetched - 1ms

# Load

DBeaver 21.0.4 - <postgres> m01.sql

File Edit Navigate Search SQL Editor Database Window Help

Database Nav... Projects ETL\_USPTO public <postgres> m01.sql fk\_application\_cleaned\_patented application\_number

Enter a part of table name here

postgres - localhost:5432

```
ALTER TABLE "application_cleaned_patented" ADD CONSTRAINT "fk_application_cleaned_patented_application_number" FOREIGN KEY( "application_number")
REFERENCES "application_cleaned_all" ( "application_number");

ALTER TABLE "application_cleaned_all" ADD CONSTRAINT "fk_application_cleaned_all_uspc_class" FOREIGN KEY("uspc_class")
REFERENCES "uspc_class_all" ("uspc_class");

ALTER TABLE "uspc_class_daysandrate" ADD CONSTRAINT "fk_uspc_class_daysandrate_uspc_class" FOREIGN KEY("uspc_class")
REFERENCES "uspc_class_all" ("uspc_class");

ALTER TABLE "uspc_attorney_success_withnames" ADD CONSTRAINT "fk_uspc_attorney_success_withnames_uspc_class" FOREIGN KEY("uspc_class")
REFERENCES "uspc_class_all" ("uspc_class");

ALTER TABLE "uspc_attorney_success_withnames" ADD CONSTRAINT "fk_uspc_attorney_success_withnames_atty_registration_number" FOREIGN KEY("atty_registration_number")
REFERENCES "attorney_success_withnames" ("atty_registration_number");

--SELECT * FROM patent_claims_stats;
SELECT * FROM all_inventors;
SELECT * FROM uspc_class_all WHERE uspc_class = '060';
SELECT * FROM uspc_class_all;
SELECT * FROM application_cleaned_patented;
SELECT * FROM application_cleaned_all;
SELECT * FROM uspc_class_daysandrate;
SELECT * FROM attorney_success_withnames;
SELECT * FROM uspc_attorney_success_withnames;
SELECT * FROM patent_claims_stats;
```

uspc\_class\_daysandrate 1

SELECT \* FROM uspc\_class\_daysandrate

uspc_class	application_number_all	application_number_patented	patented_rate	patented_cases_per_class	average_years
514	196,500	50,752	25.828	50,751	2,759,671,9735
435	168,540	46,133	27,3721	46,132	3,317,657,4903
424	168,511	52,691	31,2686	52,667	3,153,648,4987
001	166,150	7	0.0042	7	3,587,666,9276
370	150,331	78,732	52,3724	78,730	3,636,339,6314
257	146,513	79,510	54,2682	79,508	2,508,770,3417
428	126,593	46,650	36,8504	46,635	2,761,977,561
705	120,092	30,762	25,6154	30,761	4,907,268,999
455	111,943	56,758	50,7026	56,757	3,278,620,6807
345	109,589	51,559	47,0476	51,556	3,488,615,0663
438	103,545	54,882	53,003	54,882	2,403,032,5086
709	98,728	46,468	47,0667	46,468	4,297,851,519
707	81,713	40,059	49,024	40,059	3,945,486,5717
D14	75,143	72,799	96,8806	72,797	1,581,450,5754
340	74,416	32,703	43,9462	32,678	2,681,863,2116
382	72,508	37,231	51,3474	37,229	3,180,077,8374
600	72,074	28,795	39,952	28,795	3,669,334,8113

Save Cancel Script | 200+ Rows: 1

CST en Writable Smart Insert 178: 1 [37] Sel: 37: 1

200 row(s) fetched - 3ms (+3ms)

# Load

DBeaver 21.0.4 - <postgres> mo1.sql

File Edit Navigate Search SQL Editor Database Window Help

Database Navi... Projects ETL\_USPTO public <postgres> mo1.sql fk\_application\_cleaned\_patented\_application\_number

Enter a part of table name here

postgres - localhost:5432

```
ALTER TABLE "application_cleaned_patented" ADD CONSTRAINT "fk_application_cleaned_patented_application_number" FOREIGN KEY( "application_number")
REFERENCES "application_cleaned_all" ( "application_number");

ALTER TABLE "application_cleaned_all" ADD CONSTRAINT "fk_application_cleaned_all_uspc_class" FOREIGN KEY("uspc_class")
REFERENCES "uspc_class_all" ("uspc_class");

ALTER TABLE "uspc_class_daysandrate" ADD CONSTRAINT "fk_uspc_class_daysandrate_uspc_class" FOREIGN KEY("uspc_class")
REFERENCES "uspc_class_all" ("uspc_class");

ALTER TABLE "uspc_attorney_success_withnames" ADD CONSTRAINT "fk_uspc_attorney_success_withnames_uspc_class" FOREIGN KEY("uspc_class")
REFERENCES "uspc_class_all" ("uspc_class");

ALTER TABLE "uspc_attorney_success_withnames" ADD CONSTRAINT "fk_uspc_attorney_success_withnames_atty_registration_number" FOREIGN KEY("atty_registration_number")
REFERENCES "attorney_success_withnames" ("atty_registration_number");

-- 
--SELECT * FROM patent_claims_stats;
SELECT * FROM all_inventors;
SELECT * FROM uspc_class_all WHERE uspc_class = '060';
SELECT * FROM uspc_class_all;
SELECT * FROM application_cleaned_patented;
SELECT * FROM application_cleaned_all;
SELECT * FROM uspc_class_daysandrate;
SELECT * FROM attorney_success_withnames;
SELECT * FROM uspc_attorney_success_withnames;
SELECT * FROM patent_claims_stats;
```

Log

Out

application\_cleaned\_all 1

SELECT \* FROM application\_cleaned\_all

application_number	filling_date	application_invention_type	examiner_full_name	examiner_art_unit	uspc_class	uspc_subclass	atty_docket_number	appl_status_desc	appl_status_date	file_date
1	200161	Utility	[NULL]	[NULL]	[nan]	[NULL]	[NULL]	[NULL]	[NULL]	MISSIN
2	2022946	Utility	[NULL]	[NULL]	[nan]	[NULL]	[NULL]	[NULL]	[NULL]	MISSIN
3	2032617	Utility	[NULL]	[NULL]	[nan]	[NULL]	[NULL]	[NULL]	[NULL]	MISSIN
4	2045760	1948-08-23	[NULL]	[NULL]	[nan]	[NULL]	[NULL]	Patented File - (Old Case Added for File Tracking Purposes)	1986-04-25	FILE RE
5	2048602	[NULL]	[NULL]	[NULL]	[nan]	[NULL]	[NULL]	Patented File - (Old Case Added for File Tracking Purposes)	1984-03-22	FILE RE
6	2081957	Utility	[NULL]	[NULL]	[nan]	[NULL]	[NULL]	[NULL]	[NULL]	MISSIN
7	2089162	Utility	[NULL]	[NULL]	[nan]	[NULL]	[NULL]	[NULL]	[NULL]	MISSIN
8	2107244	1936-10-23	UTILITY	GREGORY, BERNARR E	3662	380	026000	Patented Case	2000-09-20	FILE RE
9	2122612	1937-01-27	[NULL]	[NULL]	000	000000	[NULL]	Patented File - (Old Case Added for File Tracking Purposes)	1992-10-13	FILE RE
10	2122769	1949-10-21	[NULL]	[NULL]	[nan]	[NULL]	[NULL]	Patented File - (Old Case Added for File Tracking Purposes)	1987-11-04	FILE RE
11	2209192	1927-07-28	[NULL]	[NULL]	[nan]	[NULL]	[NULL]	ABANDONED - RESTORED	1988-05-03	FILE RE
12	2255669	Utility	[NULL]	[NULL]	[nan]	[NULL]	[NULL]	[NULL]	[NULL]	MISSIN
13	2262037	[NULL]	[NULL]	[NULL]	[nan]	[NULL]	[NULL]	[NULL]	[NULL]	FILE RE
14	2296389	Utility	[NULL]	[NULL]	[nan]	[NULL]	[NULL]	[NULL]	[NULL]	MISSIN
15	2334093	Utility	[NULL]	[NULL]	[nan]	[NULL]	[NULL]	[NULL]	[NULL]	MISSIN
16	2356006	Utility	[NULL]	[NULL]	[nan]	[NULL]	[NULL]	[NULL]	[NULL]	MISSIN

Save Cancel Script | 200+ Rows: 1 | CST en Writable | Smart Insert | 177 : 1 [38] | Sel: 38 | 1 | 200 row(s) fetched - 5ms (+1ms)

Project - General

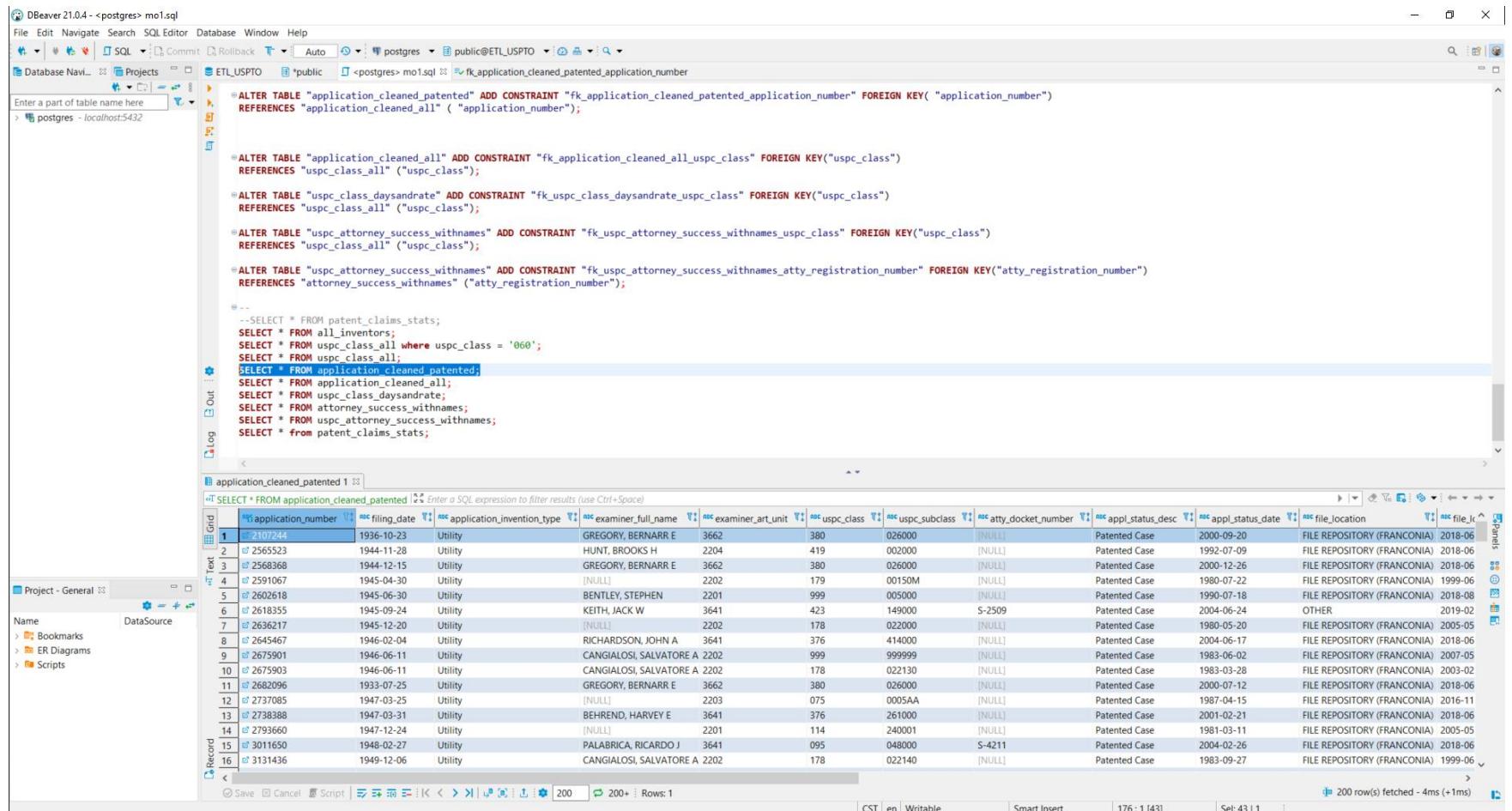
Name DataSource

Bookmarks

ER Diagrams

Scripts

## Load



## Load

The screenshot shows the DBBeaver interface with the following details:

- Top Bar:** DBBeaver 21.0.4 - postgres> mol.sql
- Menu:** File Edit Navigate Search SQL Editor Database Window Help
- Toolbar:** Back Forward Home Refresh Save Commit Rollback Auto Open Close Connect Disconnect
- Database Navigator:** Shows 'ETL\_USPTO' and 'public' schemas.
- Script Editor:** Contains a series of `ALTER TABLE` statements and a block of `SELECT` statements. The `SELECT` block retrieves data from various tables including `all_inventors`, `uspc_class_all`, and `patent_claims_stats`.
- Results Grid:** A table titled 'all\_inventors 1' showing 17 rows of inventor data. The columns are: application\_number, inventor\_name\_first, inventor\_name\_middle, inventor\_name\_last, inventor\_rank, inventor\_city\_name, inventor\_region\_code, and inventor\_country\_code.
- Project Explorer:** Shows a 'General' project with 'Bookmarks', 'ER Diagrams', and 'Scripts'.
- Status Bar:** Shows '200 row(s) fetched - 5ms (+1ms)'.

application_number	inventor_name_first	inventor_name_middle	inventor_name_last	inventor_rank	inventor_city_name	inventor_region_code	inventor_country_code	
1	5692088	TROY	E.	PLUNK	1	Bedford	MA	US
2	5692088	JAMES	E.	HOPSON	2	Wellesley	MA	US
3	5692179	PIETRO		VENTRESCA	1	Littleton	MA	US
4	5692179	FRANCIS	J.	O'HARA	2	Bedford	MA	US
5	5692179	CHESTER	J.	HUNT	3	Melrose	MA	US
6	5692179	DOMINIC	V.	RESTAGNO	4	Waltham	MA	US
7	5692179	KENNETH	R.	TURNER	5	Atkinson	NH	US
8	6106811	BARRY	N.	LEVITT	1	FRAMINGHAM	MA	US
9	6106811	THOMAS	H.	CROCKER	2	BURLINGTON	MA	US
10	6106811	RICHARD	A.	BECKERLEG	3	BOXFORD	MA	US
11	6106811	DANIEL	O.	EYRING	4	ACTON	MA	US
12	6179664	JOHN		ALLAN	1	ELLON	[NULL]	GB
13	6209457	JOANNE	B.	GERG	1	ARLINGTON HGTS.	IL	US
14	6214417	HORST		BUHL	1	WEINSTADT	[NULL]	DE
15	6307951	AMILCARE		BOVO	1	MILANO	[NULL]	IT
16	6307951	FRANCESCO		MILONI	2	GAGLIANO	[NULL]	IT
17	6307951	GIUSEPPE		VALSECCHI	3	LECCO	[NULL]	IT

## Load

## Load

## Load

# Load

```
--MAKE AN APP: I am interested in finding the uspc_class X. 074 is Machine element or mechanism  
--my_interest = "074"  
SELECT * FROM uspc_attorney_success_withnames  
where uspc_class = '074'  
order by "application_number_all" desc;
```

# HesJustNotThatIntoYou App

	uspc_class	atty_registration_number	application_number_patented	application_number_all	success_rate	atty_name_last	atty_name_first	atty_practice_category
1	074	24618		149	288	0.5173611111	Oblon	Norman
2	074	22115		163	266	0.6127819549	Schwartz	Arthur
3	074	25479		153	261	0.5862068966	Schwaab	Richard
4	074	18879		108	253	0.4268774704	Mion	John
5	074	24854		134	252	0.5317460317	Neustadt	Arthur
6	074	26257		145	242	0.59%		
7	074	19980		150	241	0.6%		
8	074	18366		92	235	0.39%		
9	074	27075		131	234	0.55%		
10	074	26001		139	233	0.59%		

# Hamlet App

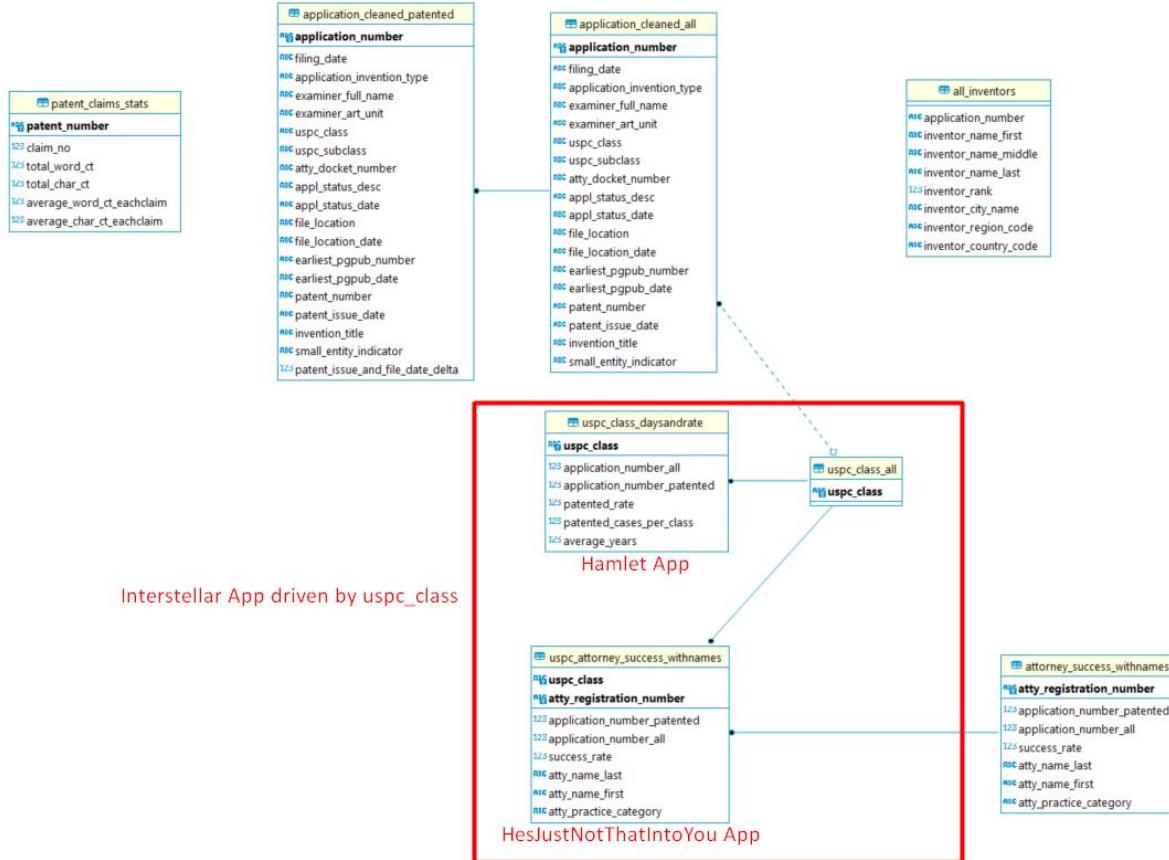
	uspc_class	application_number_all	application_number_patented	patented_rate	patented_cases_per_class	average_years
1	074	25,947	10,678	41.1531	10,674	2.5146537098

# Interstellar App

```
select *  
from uspc_attorney_success_withnames  
join uspc_class_daysandrate  
on uspc_attorney_success_withnames.uspc_class = uspc_class_daysandrate.uspc_class  
where uspc_attorney_success_withnames.uspc_class = '074'  
order by uspc_attorney_success_withnames.application_number_all desc;
```

	uspc_class	atty_registration_number	application_number_patented	application_number_all	success_rate	atty_name_last	atty_name_first	atty_pract	uspc_class	application	application	patented_rate	patented	average_years
1	074	24618		149	288	0.5173611111	Oblon	Norman	074	25,947	10,678	41.1531	10,674	2.5146537098
2	074	22115		163	266	0.6127819549	Schwartz	Arthur	074	25,947	10,678	41.1531	10,674	2.5146537098
3	074	25479		153	261	0.5862068966	Schwaab	Richard	074	25,947	10,678	41.1531	10,674	2.5146537098
4	074	18879		108	253	0.4268774704	Mion	John	074	25,947	10,678	41.1531	10,674	2.5146537098
5	074	24854		134	252	0.5317460317	Neustadt	Arthur	074	25,947	10,678	41.1531	10,674	2.5146537098
6	074	26257		145	242	0.5991735557	Blumenthal	David	074	25,947	10,678	41.1531	10,674	2.5146537098
7	074	19980		150	241	0.62406639	Jeffery	Donald	074	25,947	10,678	41.1531	10,674	2.5146537098

# Load



# Load

Out

```

select *
from patent_claims_stats
join application_cleaned_patented
on patent_claims_stats.patent_number = application_cleaned_patented.patent_number
join all_inventors
on application_cleaned_patented.application_number = all_inventors.application_number
where patent_claims_stats.patent_number = 'RE45254';

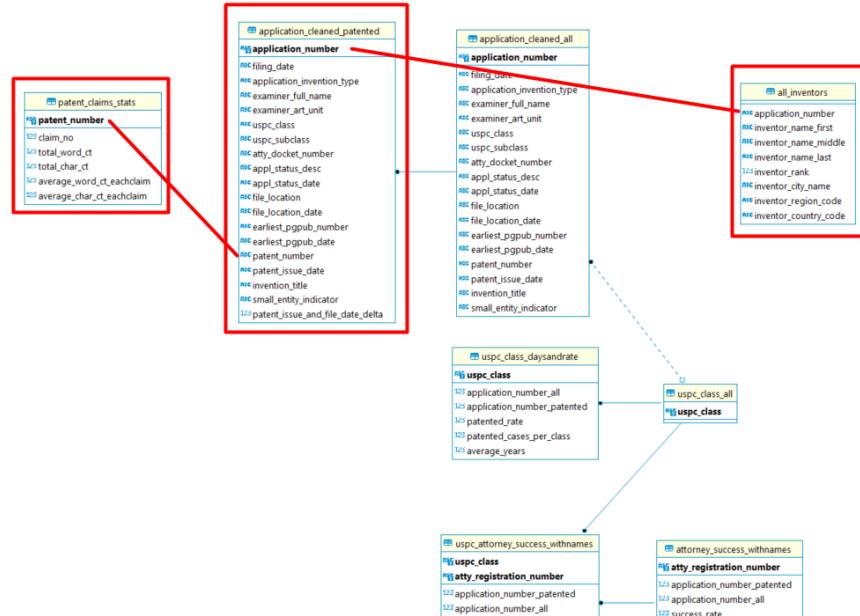
```

Log

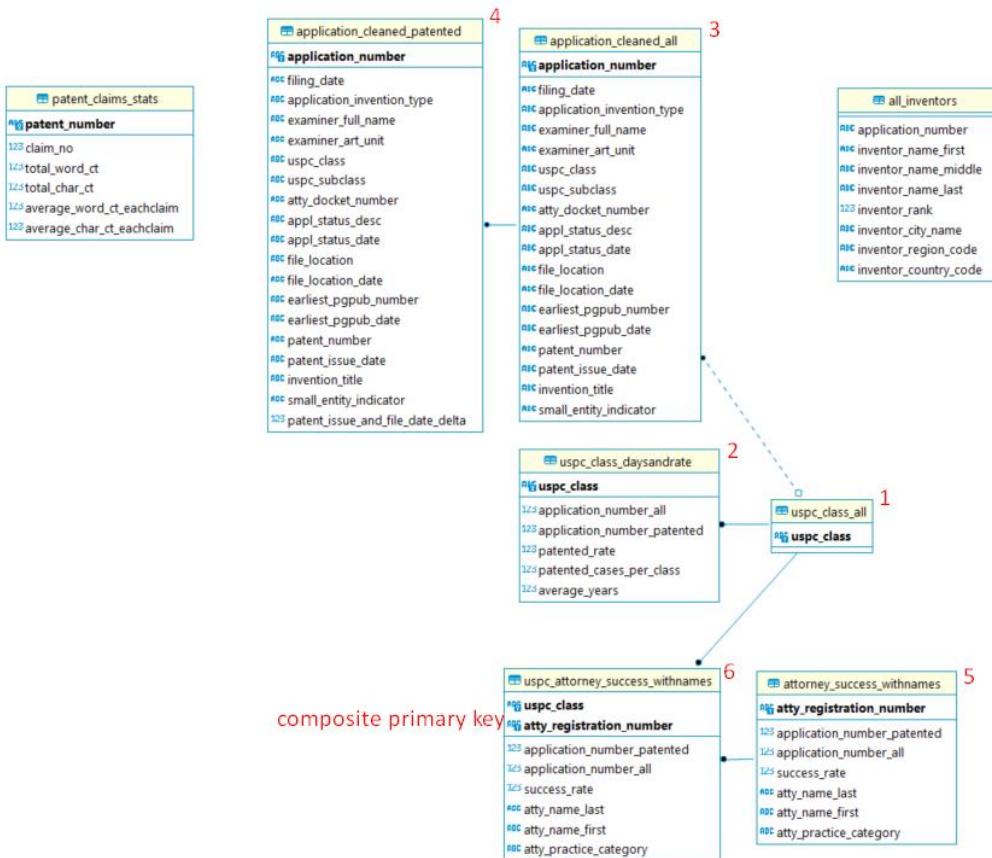
patent\_claims\_stats(+ 1)

	patent_number	claim_no	total_word_ct	total_char_ct	average_word_ct_eachclaim	average_char_ct_eachclaim	application_number	filng_date	application_invention_type	examiner_full_name	examiner_art_unit
1	RE45254	33	2,825	17,752	86	538	13907761	2013-05-31	Re-Issue	POWERS, WILLIAM S	2434

Enter a SQL expression to filter results (use Ctrl+Space)

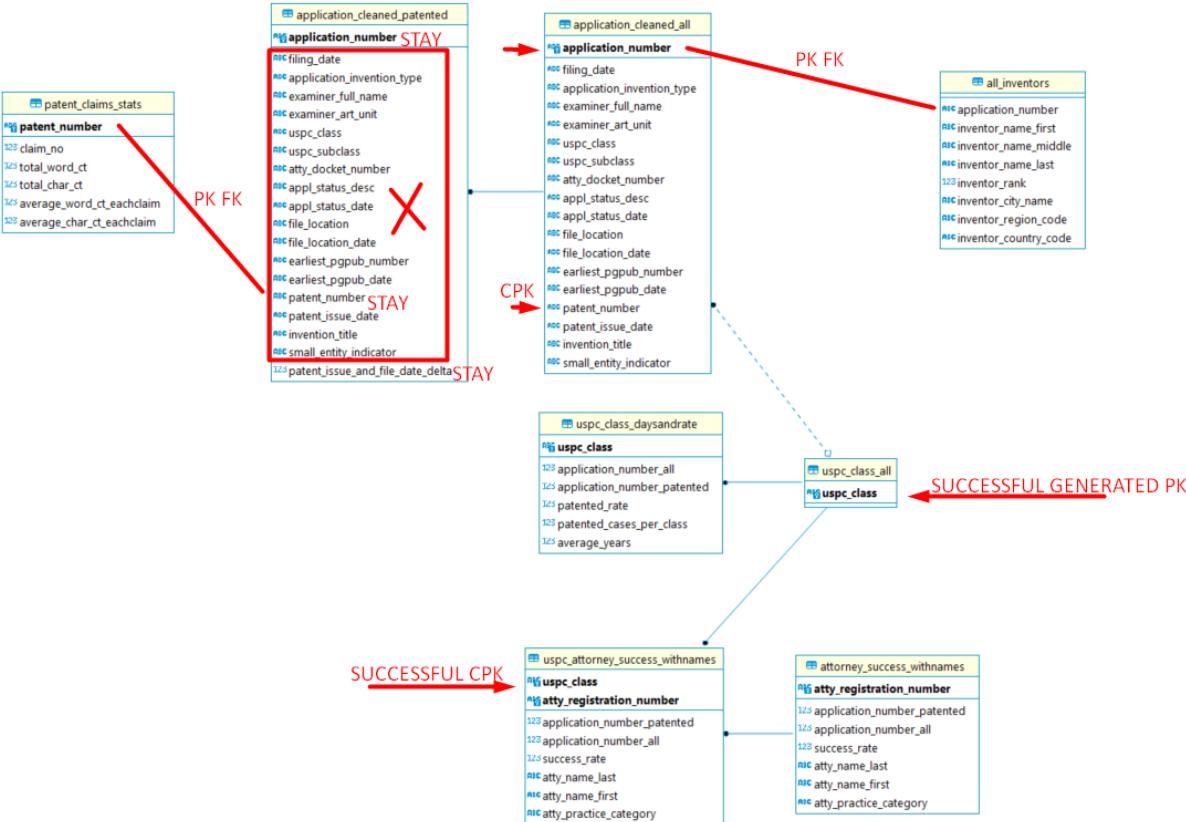


# Load



Version I

# Load



Version II

SQL Like
SQL Wildcards
SQL In
SQL Between
SQL Aliases
SQL Joins
SQL Inner Join
SQL Left Join
SQL Right Join
SQL Full Join
SQL Self Join
SQL Union
SQL Group By
SQL Having
SQL Exists
SQL Any, All
SQL Select Into
SQL Insert Into Select
SQL Case
SQL Null Functions
SQL Stored Procedures
SQL Comments
SQL Operators
SQL Database
SQL Create DB
SQL Drop DB
SQL Backup DB
SQL Create Table
SQL Drop Table
SQL Alter Table
SQL Constraints
SQL Not Null
SQL Unique
SQL Primary Key
SQL Foreign Key
SQL Check
SQL Default
SQL Index
SQL Auto Increment
SQL Dates
SQL Views
SQL Triggers

## SQL FOREIGN KEY Constraint

[◀ Previous](#)[Next ▶](#)

### SQL FOREIGN KEY Constraint

The **FOREIGN KEY** constraint is used to prevent actions that would destroy links between tables.

A **FOREIGN KEY** is a field (or collection of fields) in one table, that refers to the **PRIMARY KEY** in another table.

The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table.

Look at the following two tables:

Persons Table

PersonID	LastName	FirstName	Age
1	Hansen	Ola	30
2	Svendson	Tove	23
3	Pettersen	Kari	20

Orders Table

OrderID	OrderNumber	PersonID
1	77895	3
2	44678	3
3	22456	2
4	24562	1

Notice that the "PersonID" column in the "Orders" table points to the "PersonID" column in the "Persons" table.

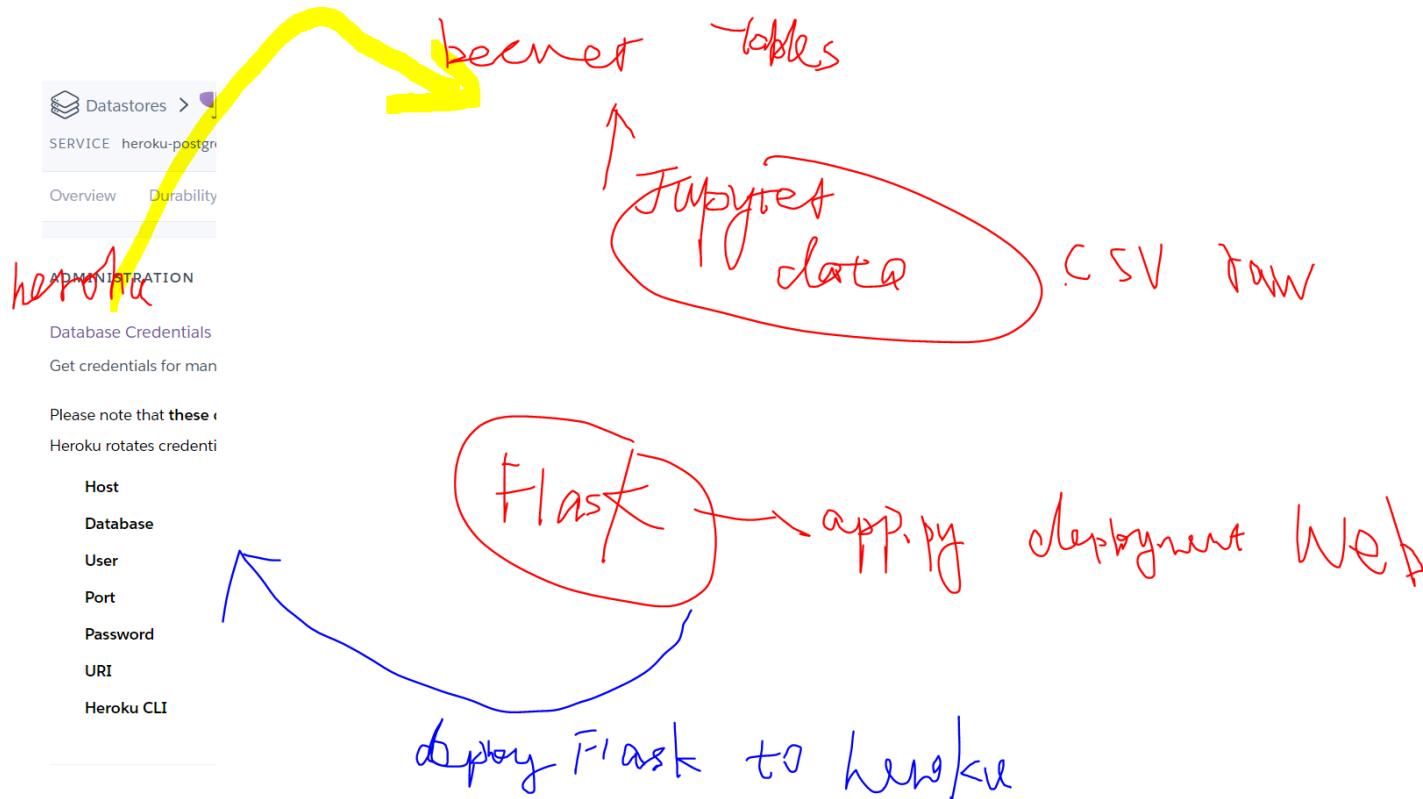
The "PersonID" column in the "Persons" table is the **PRIMARY KEY** in the "Persons" table.

The "PersonID" column in the "Orders" table is a **FOREIGN KEY** in the "Orders" table.

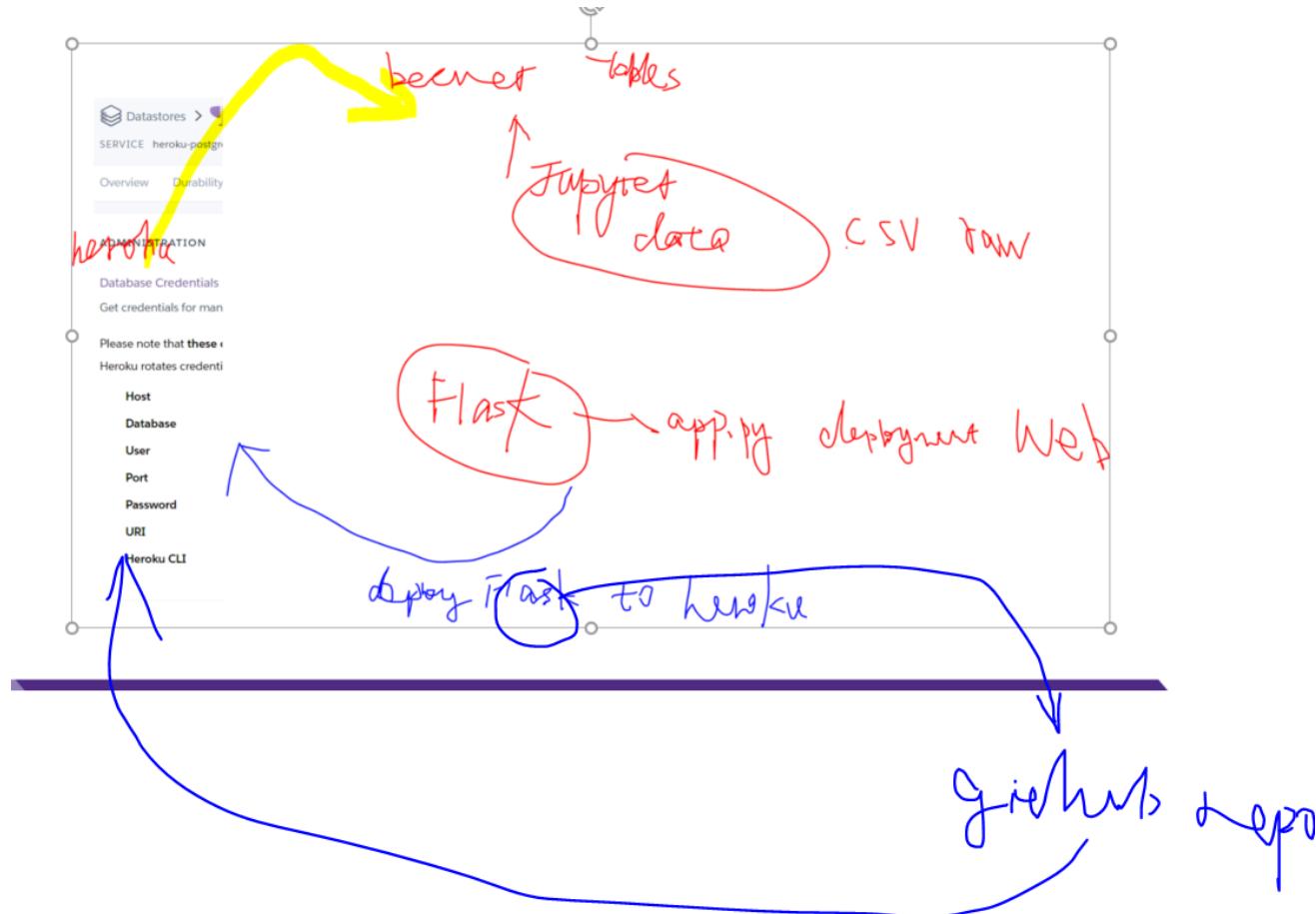
The **FOREIGN KEY** constraint prevents invalid data from being inserted into the foreign key column, because it has to be one of the values contained in the parent table.

- Primary key has to be not null AND unique AND covers all the values in foreign key – generate a dataframe that covers all values of a column and use as PK. Probably requires outer join from all tables to cover all the unique values for a column. FK can contain null.
- Composite primary key can not contain any null in any primary key column even if every row is unique. If really need to use every unique row as a primary key with a few null values in it, create an index to represent the rows and use as PK.
- Null is a bad input, prefer 0 or 000000000 which helps to make CPK in my case. Drop duplicate rows in my case.
- Better not to have duplicated columns in different tables.
- In Python, when there are mixed types (mixed float, string, int) in one column, make sure to convert them to one data type string before unique()
- Graph database
- Bear the PostgreSQL schema rules in mind when Transforming in Python.

# Deploy



# Deploy



## **Copyright and Trademark Notice**

All material in this presentation, including design, text, images, charts, are owned by Liang Gong, protected by US copyright and trademark laws. All rights are reserved by Liang Gong. Content may not be copied, reproduced, transmitted, distributed, downloaded or transferred in any form or by any means without Liang Gong's prior written consent, and with express attribution to Liang Gong. The only exception is that one temporary copy may be downloaded into a single computer's memory and one unaltered permanent copy may be used by the viewer for personal and non-commercial use only, with an attached copy of Liang Gong 's written consent. No part of the downloaded copy may be subsequently reproduced, disseminated or transferred. Copyright infringement is a violation of federal law subject to criminal and civil penalties. (For permission to reprint, please contact Liang Gong via email at [gongliangtju@gmail.com](mailto:gongliangtju@gmail.com))

# Thank You Teachers!

# Questions?