MY INTERNSHIP PROJECT

INTERN: STOIAN ALIN-BOGDAN

27 AUG 2021

TABLE OF CONTENTS

GOALS

- Use as much knowledge from the theoretical part as possible (not directly)
- Learn as much as possible (don't focus on only one thing)
- Use as much of my previous knowledge as possible

MYSQL

Why MySQL?

- See how I adapt to a new DB
- Very relevant DB (historically and today) - LAMP stack
- Useful for mixed case databases – like mine!
- MyISAM vs InnoDB
- I like it

MYSQL ON LINUX (WSL)

- DB management & administration is much easier on Linux!
- WSL 2.0 (Ubuntu 20.04 LTS)
- Installation is trivial (LAMP stack support)

```
alstoian@EN617210:~/mysql$ sudo service --status-all
[sudo] password for alstoian:
 [ - ]  apache-htcacheclean
 [ - ]  apache2
 [ - ]  apparmor
 [ ? ]  apport
 [ - ]  atd
 [ ? ]  binfmt-support
 [ - ]  console-setup.sh
 [ - ]  cron
 [ ? ]  cryptdisks
 [ ? ]  cryptdisks-early
 [ - ]  dbus
 [ ? ]  hwclock.sh
 [ + ]  irqbalance
 [ - ]  iscsid
 [ - ]  keyboard-setup.sh
 [ ? ]  kmod
 [ - ]  lvm2
 [ - ]  lvm2-lvmpolld
 [ - ]  multipath-tools
 [ + ]  mysql
 [ + ]  open-iscsi
 [ - ]  open-vm-tools
 [ ? ]  plymouth
 [ ? ]  plymouth-log
 [ - ]  procps
 [ - ]  rsync
 [ - ]  rsyslog
 [ - ]  screen-cleanup
 [ - ]  ssh
 [ ? ]  ubuntu-fan
 [ - ]  udev
 [ - ]  ufw
 [ - ]  unattended-upgrades
 [ - ]  uuidd
 [ - ]  x11-common
```

```
[--] 7) 2021-08-19T06:05:16.369255Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.26-0ubuntu0.20.04.2'  socket: '/var/run/mysqld/mysqld.sock'  port:
 3306  (Ubuntu).
[--] 8) 2021-08-19T06:05:16.369019Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '127.0.0.1' port: 33060, socket: /var/run/mysqld/mysqlx.sock
[--] 9) 2021-08-18T12:34:38.659692Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.26-0ubuntu0.20.04.2'  socket: '/var/run/mysqld/mysqld.sock'  port:
 3306  (Ubuntu).
[--] 10) 2021-08-18T12:34:38.659625Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '127.0.0.1' port: 33060, socket: /var/run/mysqld/mysqlx.sock
[--] 1 shutdown(s) detected in /home/alstoian/mysql/error.log
[--] 1) 2021-08-18T12:34:33.739876Z 0 [System] [MY-010910] [Server] /usr/sbin/mysqld: Shutdown complete (mysqld 8.0.26-0ubuntu0.20.04.2)  (Ubuntu).

-------- Storage Engine Statistics -------------------------------------------
[--] Status: +ARCHIVE +BLACKHOLE +CSV -FEDERATED +InnoDB +MEMORY +MRG_MYISAM +MyISAM +PERFORMANCE_SCHEMA
[--] Data in InnoDB tables: 1.1G (Tables: 10)
[--] Data in MyISAM tables: 25.6M (Tables: 7)
[OK] Total fragmented tables: 0

-------- Analysis Performance Metrics ----------------------------------------
[--] innodb_stats_on_metadata: OFF
[OK] No stat updates during querying INFORMATION_SCHEMA.

-------- Security Recommendations --------------------------------------------
[--] Skipped due to unsupported feature for MySQL 8

-------- CVE Security Recommendations ----------------------------------------
[OK] NO SECURITY CVE FOUND FOR YOUR VERSION

-------- Performance Metrics -------------------------------------------------
[--] Up for: 6h 9m 5s (98 q [0.004 qps], 31 conn, TX: 826K, RX: 26K)
[--] Reads / Writes: 98% / 2%
[--] Binary logging is enabled (GTID MODE: OFF)
[--] Physical Memory     : 6.1G
[--] Max MySQL memory    : 10.6G
[--] Other process memory: 0B
[--] Total buffers: 304.0M global + 70.0M per thread (151 max threads)
[--] P_S Max memory usage: 72B
[--] Galera GCache Max memory usage: 0B
[OK] Maximum reached memory usage: 584.0M (9.38% of installed RAM)
[!!] Maximum possible memory usage: 10.6G (174.61% of installed RAM)
[!!] Overall possible memory usage with other process exceeded memory
[OK] Slow queries: 2% (2/98)
[OK] Highest usage of available connections: 2% (4/151)
[!!] Aborted connections: 3.23%  (1/31)
[!!] name resolution is active : a reverse name resolution is made for each new connection and can reduce performance
[--] Query cache have been removed in MySQL 8
[OK] Sorts requiring temporary tables: 0% (0 temp sorts / 270 sorts)
[!!] Joins performed without indexes: 103
[OK] Temporary tables created on disk: 0% (0 on disk / 517 total)
[OK] Thread cache hit rate: 87% (4 created / 31 connections)
[OK] Table cache hit rate: 88% (6K hits / 7K requests)
[OK] table_definition_cache(615) is upper than number of tables(343)
[OK] Open file limit used: 0% (4/1K)
```

MYSQLTUNER

```
key_buffer -> default = 16M, recommended: innoDB: 70% physical memory, myISAM: 25%.
-> 4G
innodb_sort_buffer_size -> default = 10M -> 20M
sort_buffer_size -> default = 262k -> 2M
myisam_sort_buffer_size -> default = 8M -> 16M
read_buffer_size -> default = 131k -> 1M
read_rnd_buffer_size -> default = 262k -> 1M
join_buffer_size -> default = 262k -> 2M
bulk_insert_buffer_size -> default = 8M -> 16M

slow_query_log -> default = OFF -> ON
slow_query_log_file -> default = /var/lib/mysql/$USER-slow.log ->
~/mysql/slowqueries.log
long_query_time -> default = 10s -> 5s
log_error -> default = /var/log/mysql/error.log -> /home/alstoian/mysql/error.log
general_log_file -> default = /var/lib/mysql/USER.log ->
/home/alstoian/mysql/general.log
log_slow_extra -> default = OFF -> ON
```

```
innodb_buffer_pool_size -> default = 128M -> 256M
innodb_flush_log_at_trx_commit -> default = 1 -> 2
innodb_thread_concurrency -> default = 0 -> 8
```

..AND MANY OTHERS PERSISTED VARS ->

Data Internship

```
root@EN617210:/var/lib/mysql# python3 -m json.tool mysqld-auto.cnf
{
    "Version": 1,
    "mysql_server": {
        "innodb_thread_concurrency": {
            "Value": "8",
            "Metadata": {
                "Timestamp": 1628871726000323,
                "User": "alstoian",
                "Host": "localhost"
            }
        },
        "general_log": {
            "Value": "ON",
            "Metadata": {
                "Timestamp": 1628751661162599,
                "User": "alstoian",
                "Host": "localhost"
            }
        },
        "log_slow_extra": {
            "Value": "ON",
            "Metadata": {
                "Timestamp": 1628752338900576,
                "User": "alstoian",
                "Host": "localhost"
            }
        },
        "slow_query_log": {
            "Value": "ON",
            "Metadata": {
                "Timestamp": 1628680061923515,
                "User": "alstoian",
                "Host": "localhost"
            }
        },
        "key_buffer_size": {
            "Value": "3999997952",
            "Metadata": {
                "Timestamp": 1628674647243263,
                "User": "alstoian",
                "Host": "localhost"
            }
        },
        "long_query_time": {
            "Value": "5.000000",
            "Metadata": {
                "Timestamp": 1628680327032619,
```

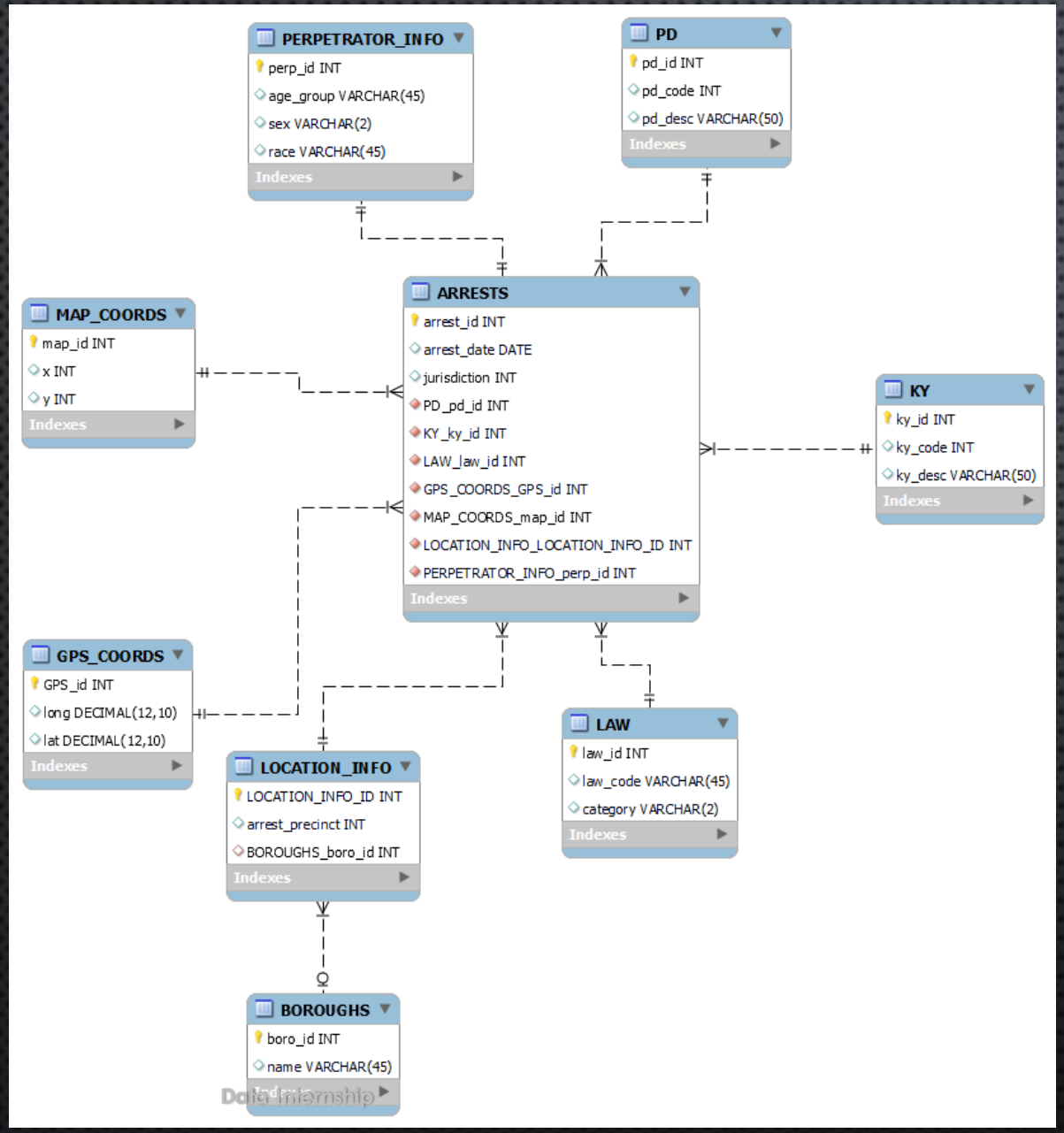
MY DATASET

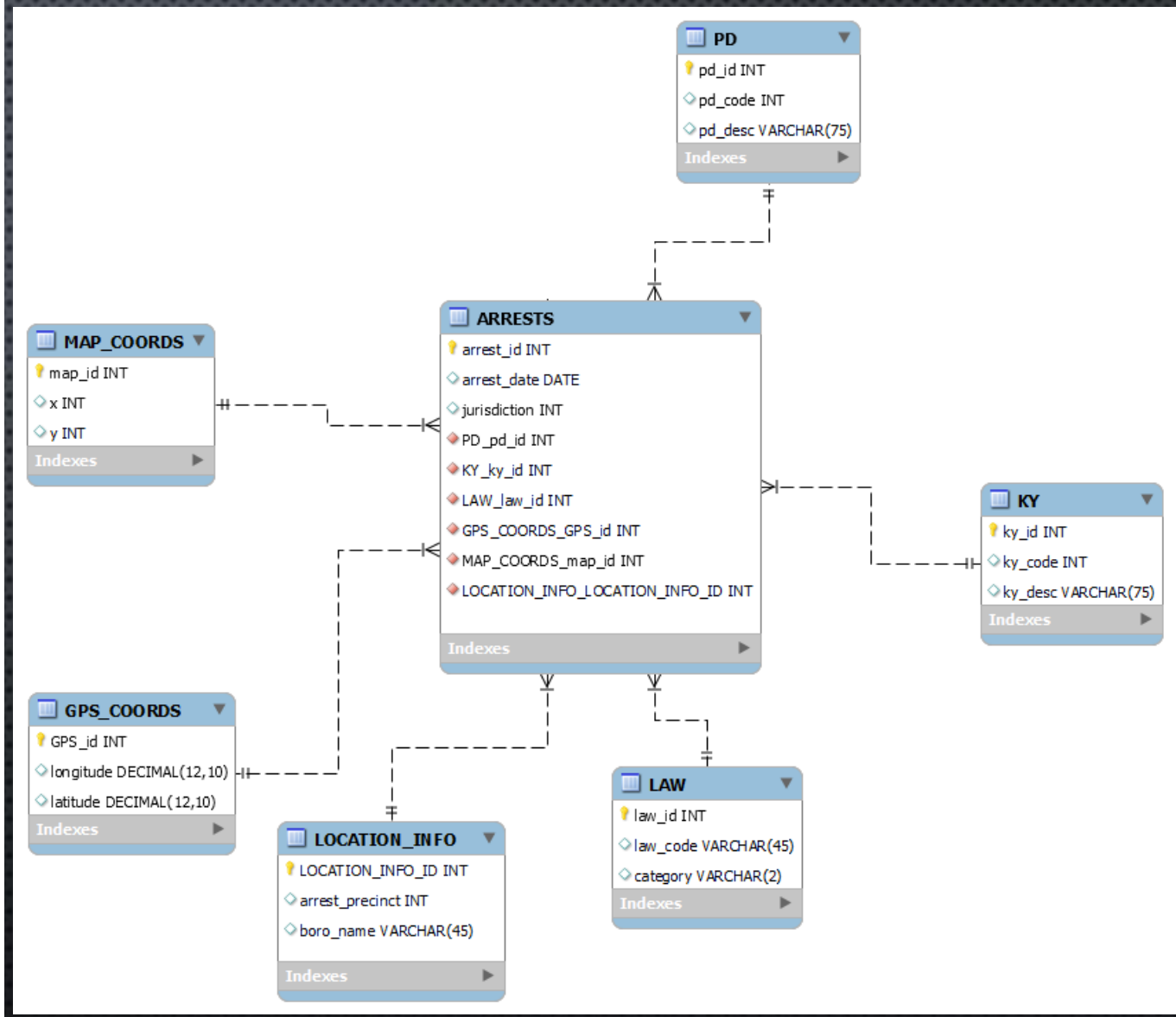
- New York arrests 2006-2020
- https://data.cityofnewyork.us/Public-Safety/NYPD-Arrests-Data-Historic-/8h9b-rp9u
- 5.2 million rows, 19 columns -> needs normalization!

DB DESIGN

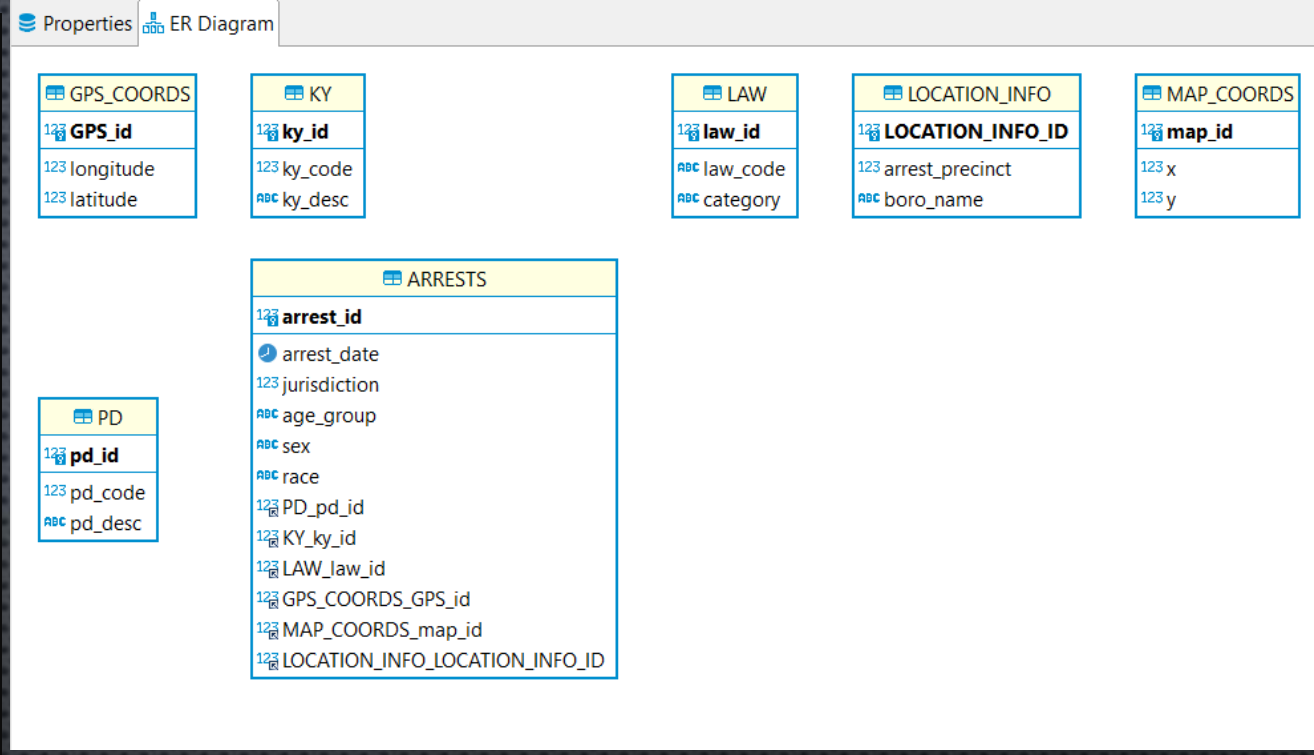
- Why two DBs?
- Main, live, Reporting, static – only from the last year (ETL!)
- No relations on Reporting?

MyISAM!


PERPETRATOR_INFO
perp_id INT
age_group VARCHAR(45)
sex VARCHAR(2)
race VARCHAR(45)
Indexes
PD
pd_id INT
pd_code INT
pd_desc VARCHAR(50)
Indexes
MAP_COORDS
map_id INT
x INT
y INT
Indexes
ARRESTS
arrest_id INT
arrest_date DATE
jurisdiction INT
PD_pd_id INT
KY_ky_id INT
LAW_law_id INT
GPS_COORDS_GPS_id INT
MAP_COORDS_map_id INT
LOCATION_INFO_LOCATION_INFO_ID INT
PERPETRATOR_INFO_perp_id INT
Indexes
KY
ky_id INT
ky_code INT
ky_desc VARCHAR(50)
Indexes
GPS_COORDS
GPS_id INT
long DECIMAL(12,10)
lat DECIMAL(12,10)
Indexes
LOCATION_INFO
LOCATION_INFO_ID INT
arrest_precinct INT
BOROUGHS_boro_id INT
Indexes
LAW
law_id INT
law_code VARCHAR(45)
category VARCHAR(2)
Indexes
BOROUGHS
boro_id INT
name VARCHAR(45)
Indexes


PD
pd_id INT
pd_code INT
pd_desc VARCHAR(75)
Indexes
MAP_COORDS
map_id INT
x INT
y INT
Indexes
ARRESTS
arrest_id INT
arrest_date DATE
jurisdiction INT
PD_pd_id INT
KY_ky_id INT
LAW_law_id INT
GPS_COORDS_GPS_id INT
MAP_COORDS_map_id INT
LOCATION_INFO_LOCATION_INFO_ID INT
Indexes
KY
ky_id INT
ky_code INT
ky_desc VARCHAR(75)
Indexes
GPS_COORDS
GPS_id INT
longitude DECIMAL(12,10)
latitude DECIMAL(12,10)
Indexes
LOCATION_INFO
LOCATION_INFO_ID INT
arrest_precinct INT
boro_name VARCHAR(45)
Indexes
LAW
law_id INT
law_code VARCHAR(45)
category VARCHAR(2)
Indexes

```
Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use main;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+------------------+
| Tables_in_main   |
+------------------+
| ARRESTS          |
| BOROUGHS         |
| GPS_COORDS       |
| KY               |
| LAW              |
| LOCATION_INFO    |
| MAP_COORDS       |
| PD               |
| PERPETRATOR_INFO |
+------------------+
9 rows in set (0.00 sec)

mysql> use report;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+------------------+
| Tables_in_report |
+------------------+
| ARRESTS          |
| GPS_COORDS       |
| KY               |
| LAW              |
| LOCATION_INFO    |
| MAP_COORDS       |
| PD               |
+------------------+
7 rows in set (0.00 sec)

mysql>
```



NO RELATIONS ON REPORTING DB BECAUSE OF MYISAM

DATA INGESTION

- Jupyter notebook
- Python
- Pandas

Dataframes make it easy to split the massive input .csv

- Helper function to generate INSERT statements (for stress testing, curiosity, testing variables, real life performance, etc.)
- No direct connection to the database (overhead!)

```
In [2]: df = pd.read_csv(r"C:\Users\alstoian\Scripts\NYPD_Arrests_Historic_Final.csv")
```

```
In [3]: df
```

Out[3]:

	ARREST_KEY	ARREST_DATE	PD_CD	PD_DESC	KY_CD	OFNS_DESC	LAW_CODE	LAW_CAT_CD	ARREST_BORO	ARREST_F
0	192799737	01/26/2019	177.0	SEXUAL ABUSE	116.0	SEX CRIMES	PL 1306503	F	M	25
1	149117452	01/06/2016	153.0	RAPE 3	104.0	RAPE	PL 1302503	F	K	67
2	190049060	11/15/2018	157.0	RAPE 1	104.0	RAPE	PL 1303501	F	K	77
3	24288194	09/13/2006	203.0	TRESPASS 3, CRIMINAL	352.0	CRIMINAL TRESPASS	PL 140100E	M	K	77
4	189182271	10/24/2018	153.0	RAPE 3	104.0	RAPE	PL 1302503	F	M	5
...	...	...	...	...	...	...	...	...	...	...
5125601	207601040	01/08/2020	273.0	TAMPERING 1,CRIMINAL	121.0	CRIMINAL MISCHIEF & RELATED OF	PL 1452000	F	M	1
5125602	206891807	01/01/2020	113.0	MENACING,UNCLASSIFIED	344.0	ASSAULT 3 & RELATED OFFENSES	PL 1201401	M	K	90
5125603	207760542	01/11/2020	339.0	LARCENY,PETIT FROM OPEN AREAS,	341.0	PETIT LARCENY	PL 1552500	M	M	13
5125604	206896678	01/01/2020	105.0	STRANGULATION 1ST	106.0	FELONY ASSAULT	PL 1211200	F	Q	111
5125605	206908101	01/02/2020	782.0	WEAPONS, POSSESSION, ETC	236.0	DANGEROUS WEAPONS	PL 2650101	M	Q	115

5125606 rows × 18 columns

This is what was left after data cleaning

endava

In [143]:
```
boroughs = df['ARREST_BORO'].unique();
boroughs_dict = {'boro_id':[x for x in range(0, len(boroughs))], 'name': boroughs}
print(boroughs_dict)
boroughs_df = pd.DataFrame(boroughs_dict)
boroughs_df
```

```
{'boro_id': [0, 1, 2, 3, 4], 'name': array(['K', 'M', 'Q', 'B', 'S'], dtype=object)}
```

Out[143]:

	boro_id	name
0	0	K
1	1	M
2	2	Q
3	3	B
4	4	S

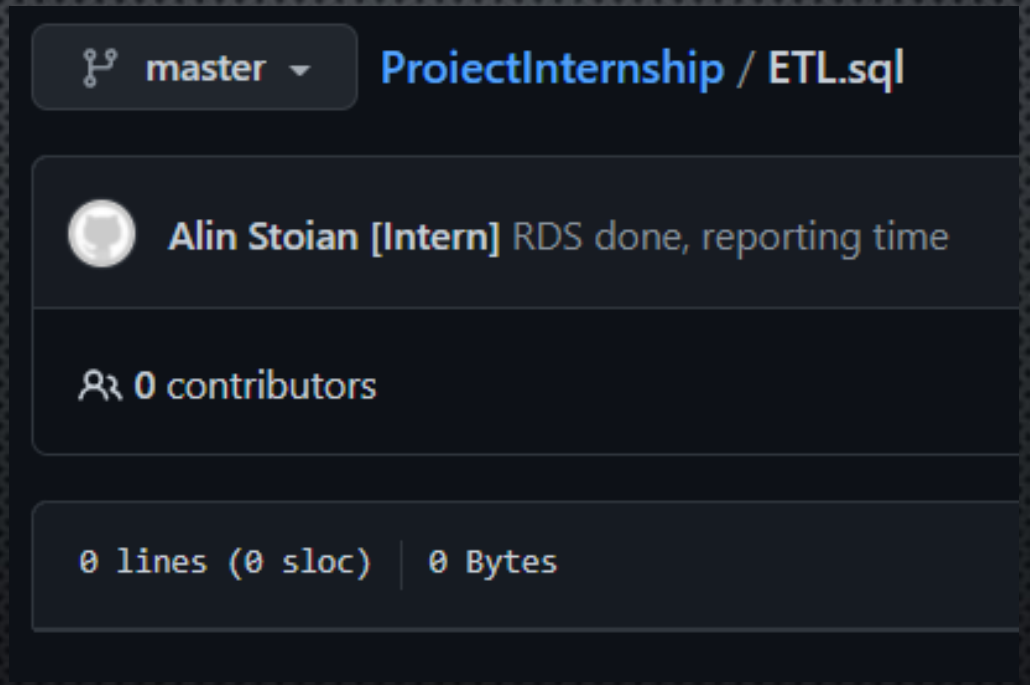
In [6]:
```
#Functie helper pentru generare de fisiere .sql cu INSERT statement-uri
def SQL_INSERT_STATEMENT_FROM_DATAFRAME(SOURCE, TARGET, FILE):
    sql_texts = []
    fout = open(FILE, 'a')
    fout.truncate(0)
    for index, row in SOURCE.iterrows():
        sql_texts.append('INSERT INTO '+TARGET+' ('+ str(', '.join(SOURCE.columns))+ ') VALUES '+ str(tuple(row.values))+';')
    for item in sql_texts:
        fout.write(str(item)+'\n')
    fout.close()
```

In [137]:
```
SQL_INSERT_STATEMENT_FROM_DATAFRAME(boroughs_df, 'BOROUGHS', 'insert_boroughs.sql')
```

```
1  INSERT INTO BOROUGHS (boro_id, name) VALUES (0, 'M');
2  INSERT INTO BOROUGHS (boro_id, name) VALUES (1, 'K');
3  INSERT INTO BOROUGHS (boro_id, name) VALUES (2, 'B');
4  INSERT INTO BOROUGHS (boro_id, name) VALUES (3, 'Q');
5  INSERT INTO BOROUGHS (boro_id, name) VALUES (4, 'S');
6
```

ETL FROM MAIN TO REPORTING

- MISTAKES HAPPEN…



FINAL RESULT

```
mysql> SELECT table_schema AS "Database",
ROUND(SU    -> ROUND(SUM(data_length + index_length) / 1024 / 1024, 2) AS "Size (MB)"
    -> FROM information_schema.TABLES
    -> GROUP BY table_schema;
+--------------------+-----------+
| Database           | Size (MB) |
+--------------------+-----------+
| mysql              |      2.63 |
| information_schema |      0.00 |
| performance_schema |      0.00 |
| sys                |      0.02 |
| main               |   1110.89 |
| report             |     25.57 |
+--------------------+-----------+
6 rows in set (0.20 sec)
```

CLOUD

- Script to backup reporting database – crontabbed
- Backup to EC2 instance
- EC2 instance to AWS RDS MySQL - crontabbed

```
#!/bin/bash
if test -f "./backup.sql.gz"; then
        rm ./backup.sql.gz
fi

mysqldump --user=alstoian --password=Admin1234 --add-drop-table --databases report --verbose > backup.sql

if test -f "./backup.sql"; then
        gzip ./backup.sql
        if test -f "./mykey2.pem"; then
                scp -i mykey2.pem backup.sql.gz ubuntu@ec2-3-121-184-109.eu-central-1.compute.amazonaws.com:/home/ubuntu
        else
                echo "no key found"
        fi
else
        echo "backup failed!"
fi
```

<- ON PREM

```
#!/bin/bash

if test -f "./backup.sql.gz"; then
        gunzip backup.sql.gz
else
        echo "no backup archive found!"
fi

pass=`cat /usr/local/etc/my-pass`

if test -f "./backup.sql"; then
        mysql --verbose -h mysql-rds.ctrc9p26yyt8.eu-central-1.rds.amazonaws.com --user=alstoian --password=$pass report < backup.sql
else
        echo "sql script not found!"
fi
```

<- EC2

EC2 > Instances > i-015d0c6f173c6ff4d

Instance summary for i-015d0c6f173c6ff4d Info

Updated less than a minute ago

Connect | Instance state | Actions

Instance ID
i-015d0c6f173c6ff4d

IPv6 address
–

Private IPv4 DNS
ip-172-30-1-73.eu-central-1.compute.internal

VPC ID
vpc-0daabc94d219e9f23

Subnet ID
subnet-011fd848aeb044e13

Public IPv4 address
3.121.184.109 | open address

Instance state
Running

Instance type
t2.micro

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Private IPv4 addresses
172.30.1.73

Public IPv4 DNS
ec2-3-121-184-109.eu-central-1.compute.amazonaws.com | open address

Elastic IP addresses
–

IAM Role
–

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Security details

IAM Role
–

Security groups
sg-0e8f678afe581f30b (finalfinalmysqlSec)

Owner ID
447227867751

Launch time
Thu Aug 19 2021 14:11:53 GMT+0300 (Eastern European Summer Time)

sg-0e8f678afe581f30b - finalfinalmysqlSec

Actions

Details

Security group name	Security group ID	Description	VPC ID
finalfinalmysqlSec	sg-0e8f678afe581f30b	Created by RDS management console	vpc-0daabc94d219e9f23

Owner	Inbound rules count	Outbound rules count
447227867751	6 Permission entries	1 Permission entry

Inbound rules | Outbound rules | Tags

You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer

Inbound rules (6)

Manage tags | Edit inbound rules

Filter security group rules

1

Security group rule...	IP version	Type	Protocol	Port range	Source	Description
sgr-0a2f1355cd3e88658	IPv4	SSH	TCP	22	81.180.208.111/32	ssh-for-ec2
sgr-035dd62bdf550e7...	IPv4	MYSQL/Aurora	TCP	3306	172.30.1.177/32	ec2-access
sgr-04a778df2121466...	IPv4	MYSQL/Aurora	TCP	3306	172.29.98.86/32	wsl-access
sgr-0c4e7d03ae37a4648	IPv4	MYSQL/Aurora	TCP	3306	81.180.209.187/32	my-access
sgr-06d11b3413eb7b	IPv4	MYSQL/Aurora	TCP	3306	81.180.208.111/32	my-access2

mysql-rds

Modify | Actions

Summary

DB identifier mysql-rds	CPU 2.46%	Status Available	Class db.t2.micro
Role Instance	Current activity 0 Connections	Engine MySQL Community	Region & AZ eu-central-1a

Connectivity & security | Monitoring | Logs & events | Configuration | Maintenance & backups | Tags

Connectivity & security

Endpoint & port

Endpoint
mysql-rds.ctrc9p26yyt8.eu-central-1.rds.amazonaws.com

Port
3306

Networking

Availability zone
eu-central-1a

VPC
vpc-0daabc94d219e9f23

Subnet group
default-vpc-0daabc94d219e9f23

Subnets
subnet-074abd267759e06db
subnet-0c9dbc70ccba5334b
subnet-011fd848aeb044e13

Security

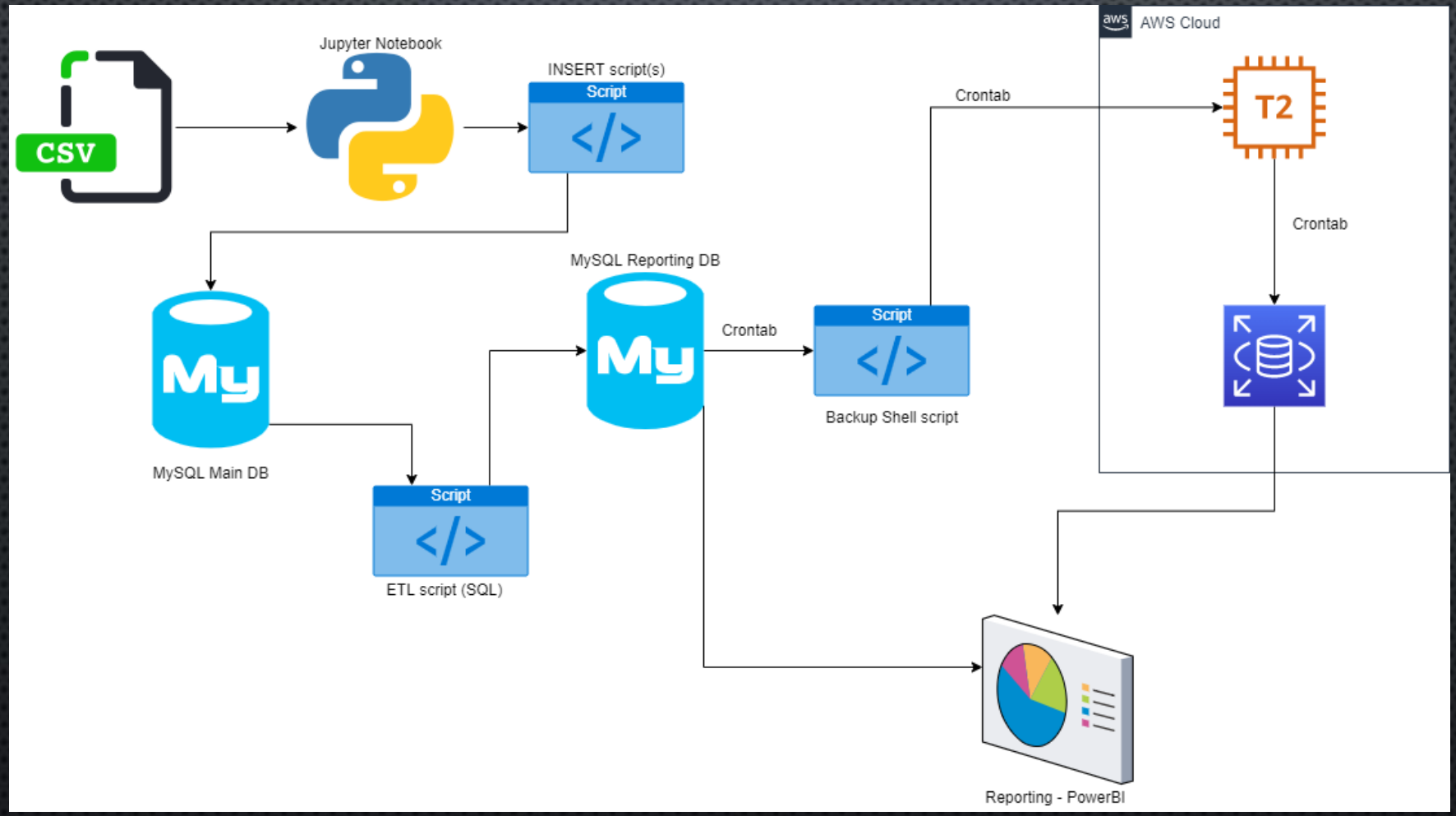
VPC security groups
finalfinalmysqlSec (sg-0e8f678afe581f30b)
(active)

Public accessibility
Yes

Certificate authority
rds-ca-2019

Certificate authority date
August 22, 2024, 08:08 (UTC±8:08)

MY PIPELINE


CSV
Jupyter Notebook
INSERT script(s)
Script
AWS Cloud
Crontab
T2
Crontab
MySQL Reporting DB
MySQL Main DB
Crontab
Script
Backup Shell script
Script
ETL script (SQL)
Reporting - PowerBI