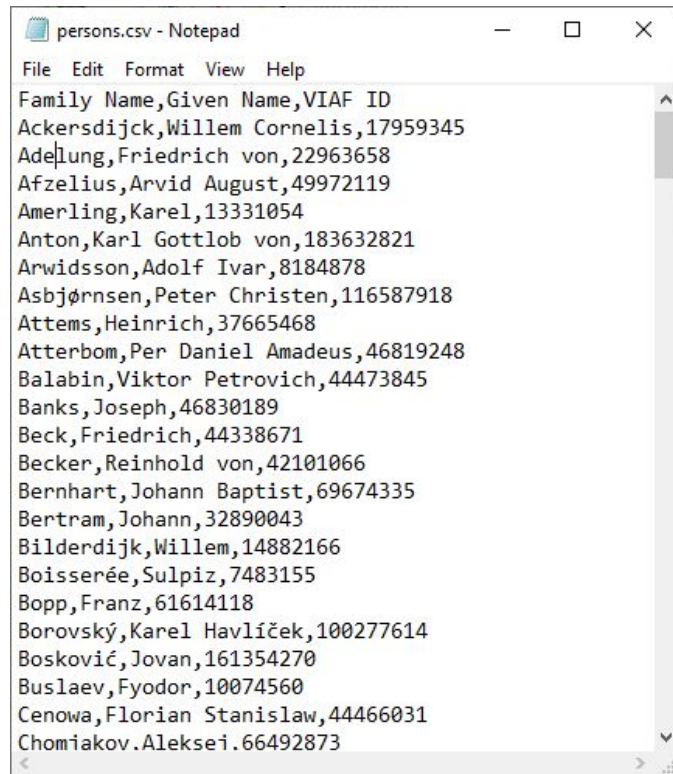


Accelerating CSV Analysis Using a Databases Approach

Arjun Rawal
CMSC 33550

CSV Files are Still the Default

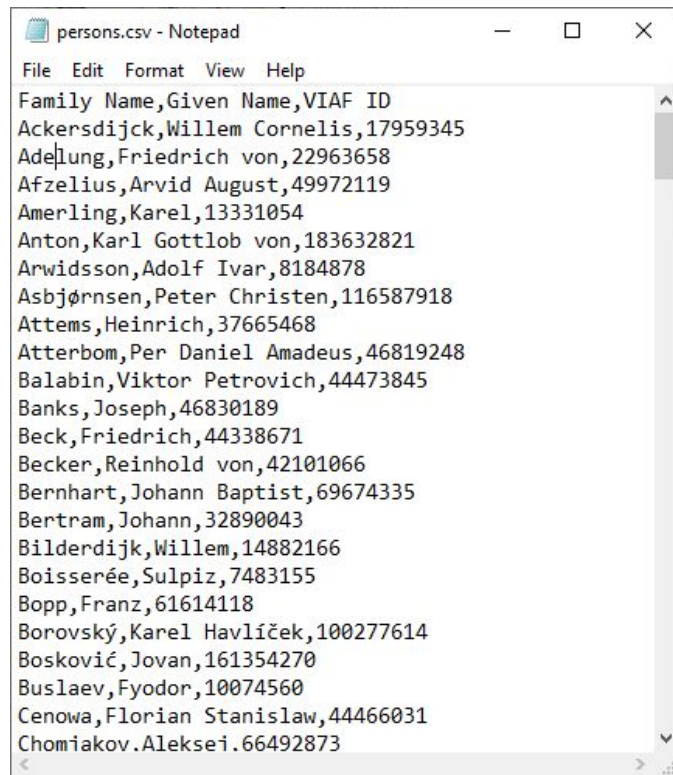
- Even HDF5 and Apache Parquet around, CSV files remain the easiest to use and most supported



```
persons.csv - Notepad
File Edit Format View Help
Family Name,Given Name,VIAF ID
Ackersdijck,Willem Cornelis,17959345
Adelung,Friedrich von,22963658
Afzelius,Arvid August,49972119
Amerling,Karel,13331054
Anton,Karl Gottlob von,183632821
Arwidsson,Adolf Ivar,8184878
Asbjørnsen,Peter Christen,116587918
Attems,Heinrich,37665468
Atterbom,Per Daniel Amadeus,46819248
Balabin,Viktor Petrovich,44473845
Banks,Joseph,46830189
Beck,Friedrich,44338671
Becker,Reinhold von,42101066
Bernhart,Johann Baptist,69674335
Bertram,Johann,32890043
Bilderdijk,Willem,14882166
Boisserée,Sulpiz,7483155
Bopp,Franz,61614118
Borovský,Karel Havlíček,100277614
Bosković,Jovan,161354270
Buslaev,Fyodor,10074560
Cenowa,Florian Stanislaw,44466031
Chomiakov,Aleksei,66492873
```

CSV Files are Still the Default

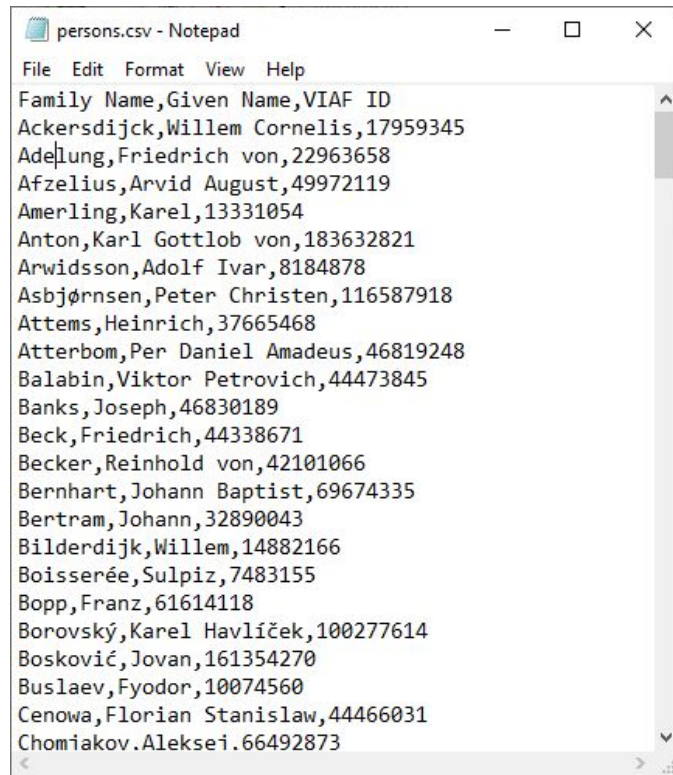
- Even HDF5 and Apache Parquet around, CSV files remain the easiest to use and most supported
- Libraries like Pandas allow for larger than memory CSV files, but processing is extremely slow (SELECT * FROM table1 where id > 1000)



```
persons.csv - Notepad
File Edit Format View Help
Family Name,Given Name,VIAF ID
Ackersdijck,Willem Cornelis,17959345
Adelung,Friedrich von,22963658
Afzelius,Arvid August,49972119
Amerling,Karel,13331054
Anton,Karl Gottlob von,183632821
Arwidsson,Adolf Ivar,8184878
Asbjørnsen,Peter Christen,116587918
Attems,Heinrich,37665468
Atterbom,Per Daniel Amadeus,46819248
Balabin,Viktor Petrovich,44473845
Banks,Joseph,46830189
Beck,Friedrich,44338671
Becker,Reinhold von,42101066
Bernhart,Johann Baptist,69674335
Bertram,Johann,32890043
Bilderdijk,Willem,14882166
Boisserée,Sulpiz,7483155
Bopp,Franz,61614118
Borovský,Karel Havlíček,100277614
Bosković,Jovan,161354270
Buslaev,Fyodor,10074560
Cenowa,Florian Stanislaw,44466031
Chomiakov,Aleksei,66492873
```

CSV Files are Still the Default

- Even HDF5 and Apache Parquet around, CSV files remain the easiest to use and most supported
- Libraries like Pandas allow for larger than memory CSV files, but processing is extremely slow (SELECT * FROM table1 where id > 1000)
- Every selection query requires reading every line of the file (~2 min for a 7 GB file)



```
persons.csv - Notepad
File Edit Format View Help
Family Name,Given Name,VIAF ID
Ackersdijck,Willem Cornelis,17959345
Adelung,Friedrich von,22963658
Afzelius,Arvid August,49972119
Amerling,Karel,13331054
Anton,Karl Gottlob von,183632821
Arwidsson,Adolf Ivar,8184878
Asbjørnsen,Peter Christen,116587918
Attems,Heinrich,37665468
Atterbom,Per Daniel Amadeus,46819248
Balabin,Viktor Petrovich,44473845
Banks,Joseph,46830189
Beck,Friedrich,44338671
Becker,Reinhold von,42101066
Bernhart,Johann Baptist,69674335
Bertram,Johann,32890043
Bilderdijk,Willem,14882166
Boisserée,Sulpiz,7483155
Bopp,Franz,61614118
Borovský,Karel Havlíček,100277614
Bosković,Jovan,161354270
Buslaev,Fyodor,10074560
Cenowa,Florian Stanislaw,44466031
Chomiakov,Aleksei.66492873
```

Pandas Wastes Space



- Pandas data is often stored inefficiently, increasing memory usage and slowing down larger than memory computations

Pandas Wastes Space



- Pandas data is often stored inefficiently, increasing memory usage and slowing down larger than memory computations
- To reduce data usage, we automatically categorize columns, and shrink numerical data types

Pandas Wastes Space



- Pandas data is often stored inefficiently, increasing memory usage and slowing down larger than memory computations
- To reduce data usage, we automatically categorize columns, and shrink numerical data types
- We use Feather, a storage format of the Apache Arrow in memory columnar store to hold both a columnar and entire frame

Evaluating on Real World Data

- CA police dataset (6.7 GB mix of string, date, integer)
- 2-core Intel Processor, 8GB RAM

Evaluating on Real World Data

- CA police dataset (6.7 GB mix of string, date, integer)
- 2-core Intel Processor, 8GB RAM
- Compared standard
 - `read_csv`
 - `read_csv` with iterator
 - `read_fast`

Evaluating on Real World Data

- CA police dataset (6.7 GB mix of string, date, integer)
- 2-core Intel Processor, 8GB RAM

- Compared standard

- read_csv

```
df = pd.read_csv(FILE)
```

- read_csv with iterator

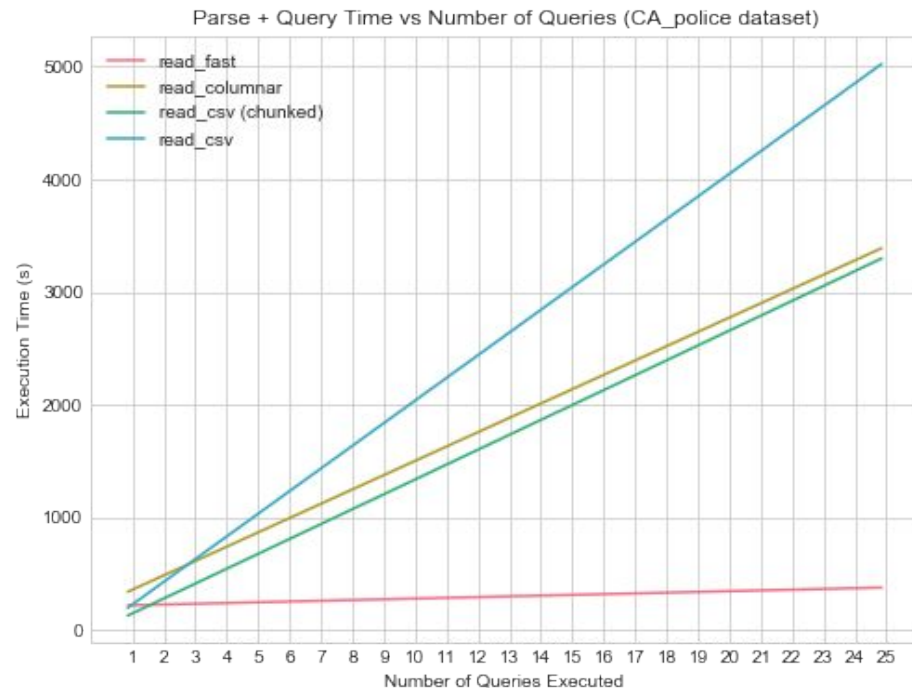
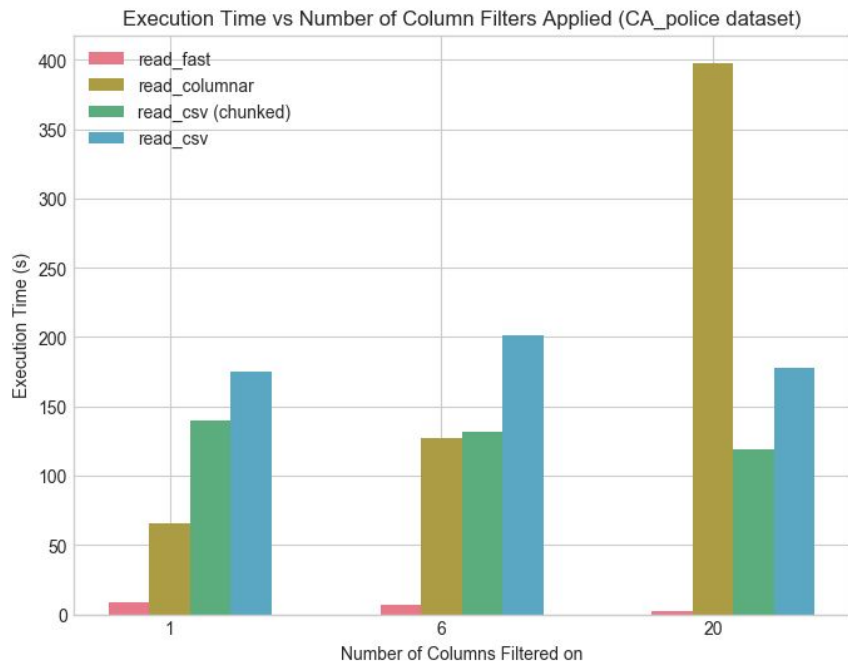
```
for i, df_p in enumerate(pd.read_csv(FILE, iterator=True, chunksize=CHUNK_SIZE)):
```

- read_fast

```
num_chunks = write_chunks(FILE, CHUNK_SIZE)
```

```
df_read = read_fast(FEATHER_DIR, FILE, PREDS, COLS, num_chunks)
```

Speedup on Multiple Columns, Multiple Queries



Questions?

