

## CINS 370 Team Project Assignment 4

### Phase 4: Advanced SQL

By Group 1

The first 5 pages of this project is showing the screenshots of the datasets we used in this project from phase 1. There are 5 different tables that we loaded in sql, named as Population, Immigration, Income, Product and Poverty Table. We have used the same data to answer all the questions for this project.

#### Population Table

```
|SQL> SELECT * FROM POPULATION;
```

| COUNTRY | STATE                | TOTAL_POPULATION |
|---------|----------------------|------------------|
| USA     | California           | 39776830         |
| USA     | Texas                | 28704330         |
| USA     | Florida              | 21312211         |
| USA     | New York             | 19862512         |
| USA     | Pennsylvania         | 12823989         |
| USA     | Illinois             | 12768320         |
| USA     | Ohio                 | 11694664         |
| USA     | Georgia              | 10545138         |
| USA     | North Carolina       | 10390149         |
| USA     | Michigan             | 9991177          |
| USA     | New Jersey           | 9832872          |
| USA     | Virginia             | 8525660          |
| USA     | Washington           | 7530552          |
| USA     | Arizona              | 7123898          |
| USA     | Massachusetts        | 6895917          |
| USA     | Tennessee            | 6782564          |
| USA     | Indiana              | 6699629          |
| USA     | Missouri             | 6135888          |
| USA     | Maryland             | 6079602          |
| USA     | Wisconsin            | 5818049          |
| USA     | Colorado             | 5684203          |
| USA     | Minnesota            | 5628162          |
| USA     | South Carolina       | 5088916          |
| USA     | Alabama              | 4888949          |
| USA     | Louisiana            | 4682509          |
| USA     | Kentucky             | 4472265          |
| USA     | Oregon               | 4199563          |
| USA     | Oklahoma             | 3940521          |
| USA     | Connecticut          | 3588683          |
| USA     | Iowa                 | 3160553          |
| USA     | Utah                 | 3159345          |
| USA     | Nevada               | 3056824          |
| USA     | Arkansas             | 3020327          |
| USA     | Mississippi          | 2982785          |
| USA     | Kansas               | 2918515          |
| USA     | New Mexico           | 2890708          |
| USA     | Nebraska             | 1932549          |
| USA     | West Virginia        | 1883077          |
| USA     | Idaho                | 1753860          |
| USA     | Hawaii               | 1426393          |
| USA     | New Hampshire        | 1350575          |
| USA     | Maine                | 1341582          |
| USA     | Montana              | 1062330          |
| USA     | Rhode Island         | 1061712          |
| USA     | Delaware             | 971180           |
| USA     | South Dakota         | 877790           |
| USA     | North Dakota         | 755238           |
| USA     | Alaska               | 738068           |
| USA     | District of Columbia | 703608           |
| USA     | Vermont              | 623960           |
| USA     | Wyoming              | 573720           |

```
51 rows selected.
```

# Immigration Table

SQL> select \* from immigration;

| PORT_NAME                | STATE        | PORT_CODE | BORDER           | CROSSING_DATE | CONTAINER                   | VALUE  | LOCATION                              |
|--------------------------|--------------|-----------|------------------|---------------|-----------------------------|--------|---------------------------------------|
| Nogales                  | Arizona      | 2604      | US-Mexico Border | 01-MAR-19     | Truck Containers Empty      | 5928   | POINT (-110.93361 31.340279999999996) |
| Scobury                  | Montana      | 3309      | US-Canada Border | 01-MAR-19     | Truck Containers Full       | 3      | POINT (-105.44272 48.80707)           |
| Del Bonita               | Montana      | 3322      | US-Canada Border | 01-MAR-19     | Personal Vehicles           | 1548   | POINT (-112.32481 48.63274)           |
| Bridgewater              | Maine        | 127       | US-Canada Border | 01-MAR-19     | Truck Containers Empty      | 319    | POINT (-67.84262 46.41923)            |
| Santa Teresa             | New Mexico   | 2488      | US-Mexico Border | 01-MAR-19     | Personal Vehicle Passengers | 109348 | POINT (-106.67983 31.850000000000005) |
| Otay Mesa                | California   | 2568      | US-Mexico Border | 01-MAR-19     | Buses                       | 916    | POINT (-117.85333 32.57333)           |
| San Ysidro               | California   | 2584      | US-Mexico Border | 01-MAR-19     | Buses                       | 3329   | POINT (-117.8266699999998 32.54306)   |
| Nogales                  | Arizona      | 2604      | US-Mexico Border | 01-MAR-19     | Bus Passengers              | 17479  | POINT (-110.93361 31.3402799999996)   |
| Lukeville                | Arizona      | 2682      | US-Mexico Border | 01-MAR-19     | Personal Vehicles           | 46128  | POINT (-112.2150000000001 31.8825)    |
| Roseau                   | Minnesota    | 3426      | US-Canada Border | 01-MAR-19     | Personal Vehicles           | 2658   | POINT (-95.81074 48.77518)            |
| Skagway                  | Alaska       | 3183      | US-Canada Border | 01-MAR-19     | Personal Vehicle Passengers | 3077   | POINT (-135.38656 59.4719699999999)   |
| Fortuna                  | North Dakota | 3417      | US-Canada Border | 01-MAR-19     | Truck Containers Empty      | 26     | POINT (-103.77986 48.91086)           |
| Vanceboro                | Maine        | 185       | US-Canada Border | 01-MAR-19     | Trains                      | 28     | POINT (-67.42955 45.55984)            |
| Baudette                 | Minnesota    | 3424      | US-Canada Border | 01-MAR-19     | Bus Passengers              | 196    | POINT (-94.72512 48.70731)            |
| Sarles                   | North Dakota | 3489      | US-Canada Border | 01-MAR-19     | Buses                       | 9      | POINT (-98.99467 48.941650000000004)  |
| Fort Kent                | Maine        | 110       | US-Canada Border | 01-MAR-19     | Bus Passengers              | 13     | POINT (-68.58458 47.26878)            |
| Grand Portage            | Minnesota    | 3613      | US-Canada Border | 01-MAR-19     | Truck Containers Full       | 697    | POINT (-89.68472800000001 47.96389)   |
| Buffalo-Niagara Falls    | New York     | 904       | US-Canada Border | 01-MAR-19     | Train Passengers            | 1099   | POINT (-79.05694 43.09444)            |
| Calexico                 | California   | 2583      | US-Mexico Border | 01-MAR-19     | Pedestrians                 | 346158 | POINT (-115.49806000000001 32.67889)  |
| Massena                  | New York     | 704       | US-Canada Border | 01-MAR-19     | Personal Vehicle Passengers | 118275 | POINT (-74.74 44.98944)               |
| Otay Mesa                | California   | 2568      | US-Mexico Border | 01-MAR-19     | Pedestrians                 | 366812 | POINT (-117.85333 32.57333)           |
| Pinecreek                | Minnesota    | 3425      | US-Canada Border | 01-MAR-19     | Truck Containers Empty      | 14     | POINT (-95.81074 48.77518)            |
| Calexico East            | California   | 2587      | US-Mexico Border | 01-MAR-19     | Truck Containers Full       | 20256  | POINT (-115.48433000000001 32.67524)  |
| Otay Mesa                | California   | 2566      | US-Mexico Border | 01-MAR-19     | Truck Containers Full       | 63197  | POINT (-117.85333 32.57333)           |
| Otay Mesa                | California   | 2586      | US-Mexico Border | 01-MAR-19     | Truck Containers Empty      | 22225  | POINT (-117.85333 32.57333)           |
| Douglas                  | Arizona      | 2601      | US-Mexico Border | 01-MAR-19     | Truck Containers Full       | 1639   | POINT (-109.54472 31.34423999999995)  |
| Grand Portage            | Minnesota    | 3613      | US-Canada Border | 01-MAR-19     | Personal Vehicle Passengers | 30064  | POINT (-89.68472800000001 47.96389)   |
| Northgate                | North Dakota | 3466      | US-Canada Border | 01-MAR-19     | Pedestrians                 | 7      | POINT (-102.39634 48.89848)           |
| Port Huron               | Michigan     | 3886      | US-Canada Border | 01-MAR-19     | Rail Containers Empty       | 16839  | POINT (-82.47889 42.97583)            |
| Sarles                   | North Dakota | 3489      | US-Canada Border | 01-MAR-19     | Truck Containers Full       | 54     | POINT (-98.99457 48.941650000000004)  |
| Northgate                | North Dakota | 3466      | US-Canada Border | 01-JAN-19     | Personal Vehicles           | 724    | POINT (-102.39634 48.89848)           |
| Tecate                   | California   | 2565      | US-Mexico Border | 01-JAN-19     | Personal Vehicle Passengers | 169601 | POINT (-116.62667 32.57722)           |
| Calexico East            | California   | 2587      | US-Mexico Border | 01-JAN-19     | Bus Passengers              | 8329   | POINT (-115.48433000000001 32.67524)  |
| Hannah                   | North Dakota | 3488      | US-Canada Border | 01-JAN-19     | Truck Containers Empty      | 7      | POINT (-98.701 48.97029)              |
| Detroit                  | Michigan     | 3891      | US-Canada Border | 01-JAN-19     | Trains                      | 150    | POINT (-83.19222 42.383690000000001)  |
| Douglas                  | Arizona      | 2491      | US-Mexico Border | 01-JAN-19     | Buses                       | 34     | POINT (-109.54472 31.34423999999995)  |
| Trout River              | New York     | 715       | US-Canada Border | 01-JAN-19     | Trucks                      | 1081   | POINT (-73.44253 44.990610000000005)  |
| Norton                   | Vermont      | 211       | US-Canada Border | 01-JAN-19     | Rail Containers Empty       | 235    | POINT (-71.79528000000002 45.01)      |
| Derby Line               | Vermont      | 269       | US-Canada Border | 01-JAN-19     | Personal Vehicles           | 25796  | POINT (-72.09944 45.005)              |
| Wildhorse                | Montana      | 3323      | US-Canada Border | 01-JAN-19     | Personal Vehicle Passengers | 1187   | POINT (-109.67761 48.54863)           |
| Beecher Falls            | Vermont      | 266       | US-Canada Border | 01-JAN-19     | Personal Vehicles           | 2828   | POINT (-71.49664 45.00888)            |
| Buffalo-Niagara Falls    | New York     | 981       | US-Canada Border | 01-JAN-19     | Trucks                      | 73144  | POINT (-79.05694 43.09444)            |
| Limestone                | Maine        | 118       | US-Canada Border | 01-JAN-19     | Trucks                      | 15     | POINT (-67.82788 46.99461)            |
| Andrade                  | California   | 2562      | US-Mexico Border | 01-JAN-19     | Personal Vehicle Passengers | 95295  | POINT (-114.63402 32.7394)            |
| Pembina                  | North Dakota | 3481      | US-Canada Border | 01-JAN-19     | Truck Containers Full       | 16593  | POINT (-97.24333 48.96639)            |
| Eastport                 | Idaho        | 3302      | US-Canada Border | 01-JAN-19     | Truck Containers Empty      | 633    | POINT (-116.1882799999998 48.99944)   |
| Sault Sainte Marie       | Michigan     | 3883      | US-Canada Border | 01-JAN-19     | Rail Containers Empty       | 1881   | POINT (-84.29889 46.47828)            |
| Roma                     | Texas        | 2310      | US-Mexico Border | 01-JAN-19     | Trucks                      | 841    | POINT (-99.01833 26.43631)            |
| Anacortes                | Washington   | 3010      | US-Canada Border | 01-JAN-19     | Personal Vehicle Passengers | 1156   | POINT (-122.6173899999999 48.49988)   |
| Eagle Pass               | Texas        | 2303      | US-Mexico Border | 01-JAN-19     | Truck Containers Full       | 12834  | POINT (-109.49917 28.70889)           |
| Vanceboro                | Maine        | 185       | US-Canada Border | 01-JAN-19     | Personal Vehicles           | 1355   | POINT (-67.42955 45.55984)            |
| Scobury                  | Montana      | 3309      | US-Canada Border | 01-JAN-19     | Truck Containers Empty      | 18     | POINT (-105.44272 48.80707)           |
| Blaine                   | Washington   | 3084      | US-Canada Border | 01-JAN-19     | Rail Containers Full        | 9614   | POINT (-122.74583 48.99389)           |
| Beecher Falls            | Vermont      | 266       | US-Canada Border | 01-JAN-19     | Truck Containers Full       | 30     | POINT (-71.49664 45.00888)            |
| Tecate                   | California   | 2585      | US-Mexico Border | 01-JAN-19     | Personal Vehicles           | 90098  | POINT (-116.62667 32.57722)           |
| Alexandria Bay           | New York     | 788       | US-Canada Border | 01-JAN-19     | Truck Containers Empty      | 1529   | POINT (-75.91886 44.32583)            |
| Calais                   | Maine        | 115       | US-Canada Border | 01-JAN-19     | Trains                      | 10     | POINT (-67.27917 45.18889)            |
| Roosville                | Montana      | 3318      | US-Canada Border | 01-JAN-19     | Trucks                      | 797    | POINT (-115.04512000000001 48.88171)  |
| Willow Creek             | Montana      | 3325      | US-Canada Border | 01-JAN-19     | Personal Vehicle Passengers | 385    | POINT (-109.67761 48.54863)           |
| Hightgate Springs-Alburg | Vermont      | 212       | US-Canada Border | 01-JAN-19     | Rail Containers Full        | 1263   | POINT (-73.18583 44.979440000000004)  |
| Point Roberts            | Washington   | 3017      | US-Canada Border | 01-JAN-19     | Trucks                      | 865    | POINT (-123.87994 48.96978)           |
| Roseau                   | Minnesota    | 3426      | US-Canada Border | 01-JAN-19     | Truck Containers Empty      | 84     | POINT (-95.81074 48.77518)            |
| Point Roberts            | Washington   | 3017      | US-Canada Border | 01-JAN-19     | Truck Containers Empty      | 674    | POINT (-123.87994 48.96978)           |
| Ferry                    | Washington   | 3013      | US-Canada Border | 01-JAN-19     | Truck Containers Empty      | 4      | POINT (-118.58774 48.88627)           |
| Roosville                | Montana      | 3318      | US-Canada Border | 01-JAN-19     | Bus Passengers              | 212    | POINT (-115.04512000000001 48.88171)  |
| Sumas                    | Washington   | 3089      | US-Canada Border | 01-JAN-19     | Trucks                      | 12137  | POINT (-122.26361 49.80028)           |
| Walhalla                 | North Dakota | 3407      | US-Canada Border | 01-JAN-19     | Personal Vehicles           | 2160   | POINT (-97.91778 48.92333)            |
| Ambrose                  | North Dakota | 3410      | US-Canada Border | 01-JAN-19     | Personal Vehicles           | 84     | POINT (-103.48222 48.95389)           |
| Medawaska                | Maine        | 189       | US-Canada Border | 01-JAN-19     | Truck Containers Empty      | 146    | POINT (-68.3271 47.35446)             |

1798 rows selected.

SQL> select count(\*) from immigration;

COUNT(\*)

1798

SQL> select count(\*) as "Number of Records" from immigration;

Number of Records

1798

SQL>

## Income Table

```
[SQL]> COLUMN COUNTRY_CODE HEADING "COUNTRY_CODE"
[SQL]> COLUMN COUNTRY_CODE FORMAT A15
[SQL]> select * from income;
```

| COUNTRY_CODE | REGION                    | INCOME_GROUP        | COUNTRY            |
|--------------|---------------------------|---------------------|--------------------|
| NOR          | Europe & Central Asia     | High income         | Norway             |
| NPL          | South Asia                | Low income          | Nepal              |
| NRU          | East Asia & Pacific       | High income         | Nauru              |
| NZL          | East Asia & Pacific       | High income         | New Zealand        |
| PAK          | South Asia                | Lower middle income | Pakistan           |
| PAN          | Latin America & Caribbean | Upper middle income | Panama             |
| PER          | Latin America & Caribbean | Upper middle income | Peru               |
| PHL          | East Asia & Pacific       | Lower middle income | Philippines        |
| PLW          | East Asia & Pacific       | Upper middle income | Palau              |
| PNG          | East Asia & Pacific       | Lower middle income | Papua New Guinea   |
| POL          | Europe & Central Asia     | High income         | Poland             |
| PRI          | Latin America & Caribbean | High income         | Puerto Rico        |
| PRK          | East Asia & Pacific       | Low income          | "Korea             |
| PRT          | Europe & Central Asia     | High income         | Portugal           |
| PRY          | Latin America & Caribbean | Upper middle income | Paraguay           |
| PYF          | East Asia & Pacific       | High income         | French Polynesia   |
| ROU          | Europe & Central Asia     | Upper middle income | Romania            |
| RUS          | Europe & Central Asia     | Upper middle income | Russian Federation |
| RWA          | Sub-Saharan Africa        | Low income          | Rwanda             |
| SDN          | Sub-Saharan Africa        | Lower middle income | Sudan              |
| SEN          | Sub-Saharan Africa        | Low income          | Senegal            |
| SGP          | East Asia & Pacific       | High income         | Singapore          |
| SLB          | East Asia & Pacific       | Lower middle income | Solomon Islands    |
| SLE          | Sub-Saharan Africa        | Low income          | Sierra Leone       |
| BHS          | Latin America & Caribbean | High income         | "Bahamas           |
| BLR          | Europe & Central Asia     | Upper middle income | Belarus            |
| BLZ          | Latin America & Caribbean | Upper middle income | Belize             |
| BMU          | North America             | High income         | Bermuda            |
| BOL          | Latin America & Caribbean | Lower middle income | Bolivia            |
| BRA          | Latin America & Caribbean | Upper middle income | Brazil             |
| BRB          | Latin America & Caribbean | High income         | Barbados           |
| BRN          | East Asia & Pacific       | High income         | Brunei Darussalam  |
| BTN          | South Asia                | Lower middle income | Bhutan             |
| BWA          | Sub-Saharan Africa        | Upper middle income | Botswana           |
| CAN          | North America             | High income         | Canada             |
| CHE          | Europe & Central Asia     | High income         | Switzerland        |
| CHI          | Europe & Central Asia     | High income         | Channel Islands    |
| CHL          | Latin America & Caribbean | High income         | Chile              |
| CHN          | East Asia & Pacific       | Upper middle income | China              |
| CIV          | Sub-Saharan Africa        | Lower middle income | Côte d'Ivoire      |
| CMR          | Sub-Saharan Africa        | Lower middle income | Cameroon           |
| COD          | Sub-Saharan Africa        | Low income          | "Congo             |
| COG          | Sub-Saharan Africa        | Lower middle income | "Congo             |
| COL          | Latin America & Caribbean | Upper middle income | Colombia           |
| COM          | Sub-Saharan Africa        | Low income          | Comoros            |
| CPV          | Sub-Saharan Africa        | Lower middle income | Cabo Verde         |
| CRI          | Latin America & Caribbean | Upper middle income | Costa Rica         |
| CUB          | Latin America & Caribbean | Upper middle income | Cuba               |
| CWU          | Latin America & Caribbean | High income         | Cura?ao            |
| CYM          | Latin America & Caribbean | High income         | Cayman Islands     |
| CYP          | Europe & Central Asia     | High income         | Cyprus             |
| CZE          | Europe & Central Asia     | High income         | Czech Republic     |
| DEU          | Europe & Central Asia     | High income         | Germany            |
| DMA          | Latin America & Caribbean | Upper middle income | Dominica           |
| DNK          | Europe & Central Asia     | High income         | Denmark            |
| DOM          | Latin America & Caribbean | Upper middle income | Dominican Republic |
| ECU          | Latin America & Caribbean | Upper middle income | Ecuador            |
| ERI          | Sub-Saharan Africa        | Low income          | Eritrea            |
| ESP          | Europe & Central Asia     | High income         | Spain              |
| EST          | Europe & Central Asia     | High income         | Estonia            |
| ETH          | Sub-Saharan Africa        | Low income          | Ethiopia           |
| FIN          | Europe & Central Asia     | High income         | Finland            |
| MMR          | East Asia & Pacific       | Lower middle income | Myanmar            |
| MNE          | Europe & Central Asia     | Upper middle income | Montenegro         |
| MNG          | East Asia & Pacific       | Lower middle income | Mongolia           |
| MOZ          | Sub-Saharan Africa        | Low income          | Mozambique         |
| MRT          | Sub-Saharan Africa        | Lower middle income | Mauritania         |
| MUS          | Sub-Saharan Africa        | Upper middle income | Mauritius          |
| MWI          | Sub-Saharan Africa        | Low income          | Malawi             |
| MYS          | East Asia & Pacific       | Upper middle income | Malaysia           |
| NAM          | Sub-Saharan Africa        | Upper middle income | Namibia            |
| NCL          | East Asia & Pacific       | High income         | New Caledonia      |
| NER          | Sub-Saharan Africa        | Low income          | Niger              |
| NGA          | Sub-Saharan Africa        | Lower middle income | Nigeria            |
| NIC          | Latin America & Caribbean | Lower middle income | Nicaragua          |
| NLD          | Europe & Central Asia     | High income         | Netherlands        |

187 rows selected.

```
[SQL]> select count(*) as "Number of Records" from income;
```

```
Number of Records
-----
187
```

SQL> ■

## Product Table

```
[SQL> COLUMN PRODUCT_COUNTRY_CODE HEADING "PRODUCT_COUNTRY_CODE"
[SQL> COLUMN PRODUCT_COUNTRY_CODE FORMAT A20
[SQL> select * from product;
```

| PRODUCT_COUNTRY_CODE | COUNTRY                  | REGION                            | ITEM_TYPE       | ORDER_ID  |
|----------------------|--------------------------|-----------------------------------|-----------------|-----------|
| FJI                  | Fiji                     | Australia and Oceania             | Personal Care   | 600370490 |
| USA                  | United States of America | North America                     | Meat            | 908136594 |
| USA                  | United States of America | North America                     | Personal Care   | 551057326 |
| USA                  | United States of America | North America                     | Baby Food       | 984673964 |
| USA                  | United States of America | North America                     | Clothes         | 699368035 |
| USA                  | United States of America | North America                     | Office Supplies | 811701095 |
| USA                  | United States of America | North America                     | Beverages       | 799003732 |
| CAN                  | Canada                   | North America                     | Vegetables      | 185941302 |
| JPN                  | Japan                    | Asia                              | Cereal          | 161442649 |
| TCD                  | Chad                     | Sub-Saharan Africa                | Fruits          | 645713555 |
| ARM                  | Armenia                  | Europe                            | Cereal          | 683458888 |
| ERI                  | Eritrea                  | Sub-Saharan Africa                | Cereal          | 679414975 |
| MNE                  | Montenegro               | Europe                            | Clothes         | 208630645 |
| JAM                  | Jamaica                  | Central America and the Caribbean | Vegetables      | 266467225 |
| FJI                  | Fiji                     | Australia and Oceania             | Vegetables      | 118598544 |
| TGO                  | Togo                     | Sub-Saharan Africa                | Clothes         | 451018930 |
| MNE                  | Montenegro               | Europe                            | Snacks          | 220003211 |
| GRC                  | Greece                   | Europe                            | Household       | 702186715 |
| SDN                  | Sudan                    | Sub-Saharan Africa                | Cosmetics       | 544485270 |
| MDV                  | Maldives                 | Asia                              | Fruits          | 714135285 |
| MNE                  | Montenegro               | Europe                            | Clothes         | 448685348 |
| EST                  | Estonia                  | Europe                            | Office Supplies | 405997025 |
| GRL                  | Greenland                | North America                     | Beverages       | 414244067 |
| CPV                  | Cape Verde               | Sub-Saharan Africa                | Clothes         | 821912881 |
| SEN                  | Senegal                  | Sub-Saharan Africa                | Household       | 247802054 |
| BGR                  | Bulgaria                 | Europe                            | Clothes         | 888999934 |
| MNG                  | Mongolia                 | Asia                              | Clothes         | 778478332 |
| GRD                  | Grenada                  | Central America and the Caribbean | Cereal          | 430390187 |
| GRD                  | Grenada                  | Central America and the Caribbean | Beverages       | 397877871 |
| SEN                  | Senegal                  | Sub-Saharan Africa                | Beverages       | 683927953 |
| GRL                  | Greenland                | North America                     | Fruits          | 469839179 |
| TCD                  | Chad                     | Sub-Saharan Africa                | Meat            | 357222878 |
| MUS                  | Mauritius                | Sub-Saharan Africa                | Personal Care   | 118002879 |
| HND                  | Honduras                 | Central America and the Caribbean | Office Supplies | 499009597 |
| BEN                  | Benin                    | Sub-Saharan Africa                | Fruits          | 564666470 |
| GRC                  | Greece                   | Europe                            | Baby Food       | 294499957 |
| MHL                  | Marshall Islands         | Australia and Oceania             | Beverages       | 666678130 |
| GRD                  | Grenada                  | Central America and the Caribbean | Baby Food       | 641018617 |
| LUX                  | Luxembourg               | Europe                            | Meat            | 775278842 |
| ZWE                  | Zimbabwe                 | Sub-Saharan Africa                | Meat            | 85545134  |
| CHN                  | China                    | Asia                              | Vegetables      | 737816321 |
| ATG                  | Antigua and Barbuda      | Central America and the Caribbean | Vegetables      | 165835034 |
| GTM                  | Guatemala                | Central America and the Caribbean | Baby Food       | 576264083 |
| THA                  | Thailand                 | Asia                              | Snacks          | 670878255 |
| SGP                  | Singapore                | Asia                              | Cereal          | 435146415 |
| PRK                  | North Korea              | Asia                              | Snacks          | 522371423 |
| AUT                  | Austria                  | Europe                            | Office Supplies | 141977107 |
| JPN                  | Japan                    | Asia                              | Baby Food       | 823699796 |
| ZWE                  | Zimbabwe                 | Sub-Saharan Africa                | Beverages       | 567588317 |
| LTU                  | Lithuania                | Europe                            | Fruits          | 594003999 |
| LUX                  | Luxembourg               | Europe                            | Baby Food       | 393628669 |
| NOR                  | Norway                   | Europe                            | Household       | 326714789 |
| NZL                  | New Zealand              | Australia and Oceania             | Household       | 398511302 |
| UKR                  | Ukraine                  | Europe                            | Personal Care   | 185177838 |
| TWN                  | Taiwan                   | Asia                              | Personal Care   | 865658832 |
| ITA                  | Italy                    | Europe                            | Cereal          | 622791612 |
| FIN                  | Finland                  | Europe                            | Personal Care   | 409774005 |
| SDN                  | Sudan                    | Sub-Saharan Africa                | Office Supplies | 800084340 |
| HRV                  | Croatia                  | Europe                            | Snacks          | 637521445 |
| MRT                  | Mauritania               | Sub-Saharan Africa                | Beverages       | 186196649 |
| NZL                  | New Zealand              | Australia and Oceania             | Baby Food       | 680533778 |
| PAK                  | Pakistan                 | Middle East and North Africa      | Beverages       | 275269162 |
| POL                  | Poland                   | Europe                            | Household       | 795451629 |
| LTU                  | Lithuania                | Europe                            | Cereal          | 986442506 |
| PAK                  | Pakistan                 | Middle East and North Africa      | Cereal          | 563915622 |
| TLS                  | East Timor               | Australia and Oceania             | Cosmetics       | 663857305 |
| MHL                  | Marshall Islands         | Australia and Oceania             | Fruits          | 692566382 |
| CUB                  | Cuba                     | Central America and the Caribbean | Household       | 576654183 |
| GRL                  | Greenland                | North America                     | Baby Food       | 313044536 |
| LUX                  | Luxembourg               | Europe                            | Personal Care   | 418973767 |
| BGR                  | Bulgaria                 | Europe                            | Cereal          | 181045520 |
| MNG                  | Mongolia                 | Asia                              | Vegetables      | 693743550 |
| DOM                  | Dominican Republic       | Central America and the Caribbean | Clothes         | 716849601 |
| FIN                  | Finland                  | Europe                            | Personal Care   | 841291654 |
| BHS                  | The Bahamas              | Central America and the Caribbean | Cereal          | 450268065 |
| GRD                  | Grenada                  | Central America and the Caribbean | Meat            | 918334138 |

329 rows selected.

```
[SQL> select count(*) as "Number of Records" from product;
```

Number of Records

329

SQL> ■

## Poverty Table

```
SQL> SET PAGESIZE 200
SQL> SET LINESIZE 200
SQL> select * from poverty;
```

| STATE                | LOWER_LEVEL_PEOPLE | UPPER_LEVEL_PEOPLE | LOWER_BOUND_CHILDREN | UPPER_BOUND_CHILDREN |
|----------------------|--------------------|--------------------|----------------------|----------------------|
| Alabama              | 784517             | 820009             | 253694               | 272124               |
| Alaska               | 75273              | 83583              | 23957                | 28201                |
| Arizona              | 1002773            | 1043901            | 324723               | 347577               |
| Arkansas             | 463966             | 487800             | 149187               | 162569               |
| California           | 5111205            | 5217133            | 1586526              | 1645300              |
| Colorado             | 552126             | 588110             | 143486               | 160312               |
| Connecticut          | 326360             | 350942             | 89324                | 101814               |
| Delaware             | 116181             | 127003             | 32862                | 37720                |
| District of Columbia | 104105             | 114899             | 29285                | 35421                |
| Florida              | 2860804            | 2942948            | 829310               | 872538               |
| Georgia              | 1503877            | 1568261            | 513320               | 550706               |
| Hawaii               | 126481             | 137921             | 32442                | 38702                |
| Idaho                | 203631             | 220243             | 60371                | 69421                |
| Illinois             | 1543257            | 1594083            | 470646               | 500986               |
| Indiana              | 841903             | 879603             | 263205               | 285029               |
| Iowa                 | 319716             | 337542             | 85662                | 94830                |
| Kansas               | 325420             | 345322             | 97150                | 108566               |
| Kentucky             | 723378             | 753748             | 211037               | 226431               |
| Louisiana            | 875670             | 911984             | 293085               | 312963               |
| Maine                | 141958             | 152868             | 32560                | 37530                |
| Maryland             | 541127             | 571971             | 157132               | 173598               |
| Massachusetts        | 675076             | 710810             | 173590               | 191332               |
| Michigan             | 1351618            | 1395098            | 404866               | 427744               |
| Minnesota            | 505806             | 528890             | 142348               | 156088               |
| Mississippi          | 561749             | 585685             | 187869               | 200803               |
| Missouri             | 776721             | 809281             | 240522               | 259136               |
| Montana              | 125024             | 136208             | 32971                | 37675                |
| Nebraska             | 191439             | 206529             | 59898                | 68064                |
| Nevada               | 379792             | 405922             | 122765               | 135439               |
| New Hampshire        | 95275              | 105825             | 22708                | 27858                |
| New Jersey           | 862408             | 901856             | 259622               | 281804               |
| New Mexico           | 378032             | 399548             | 118269               | 129821               |
| New York             | 2700018            | 2771138            | 793426               | 831246               |
| North Carolina       | 1442873            | 1493191            | 461659               | 491217               |
| North Dakota         | 70249              | 77865              | 17896                | 21426                |
| Ohio                 | 1551281            | 1599521            | 493056               | 521182               |
| Oklahoma             | 589591             | 612231             | 193877               | 207017               |
| Oregon               | 521125             | 548567             | 133798               | 148368               |
| Pennsylvania         | 1524709            | 1572405            | 430076               | 457254               |
| Rhode Island         | 118956             | 131460             | 34517                | 39665                |
| South Carolina       | 735356             | 765362             | 234515               | 251123               |
| South Dakota         | 103234             | 111810             | 31772                | 36182                |
| Tennessee            | 961862             | 998926             | 303257               | 323607               |
| Texas                | 4036102            | 4124652            | 1494968              | 1556920              |
| Utah                 | 284801             | 307475             | 90671                | 103091               |
| Vermont              | 61032              | 69104              | 13200                | 16112                |
| Virginia             | 856763             | 896113             | 247583               | 268493               |
| Washington           | 781803             | 817547             | 219520               | 240494               |
| West Virginia        | 316875             | 336129             | 83372                | 91830                |
| Wisconsin            | 627066             | 653282             | 175854               | 191036               |
| Wyoming              | 57678              | 64960              | 15694                | 18938                |

51 rows selected.

## Part 1: Oracle SQL Commands

We have used **Phase 1 Queries** to answer these questions.

### 1. How many people crossed the US border between the year 2018-2019?

Answer: The total number of vehicles that crossed the US border between the year 2018-2019 are **6890229**.

With the help of the aggregate function, **GROUP BY** and **INNER JOIN**, we get the final result

```
SQL> SELECT POPULATION.COUNTRY,SUM(IMMIGRATION.VALUE) AS "TOTAL CROSSINGS" FROM POPULATION RIGHT JOIN IMMIGRATION ON POPULATION.STATE = IMMIGRATION.STATE WHERE IMMIGRATION.CROSSING_DATE
| 2 BETWEEN to_date('01-JAN-18','DD-MON-yy') AND to_date('01-JAN-19','DD-MON-YY') GROUP BY COUNTRY;
```

| COUNTRY | TOTAL CROSSINGS |
|---------|-----------------|
| USA     | 6890229         |

### 2. How many people are below the poverty line in every state of the US?

Answer: The column “lower\_level\_people” shows **people under the poverty line** and column “lower\_bound\_children” shows **children under the poverty line**.

*The following query is used for the analysis of people under the poverty line vs total population in every state of US*

```
SQL> select country,population.state,lower_level_people,lower_bound_children,total_population from poverty inner join population on poverty.state = population.state;
```

| COUNTRY | STATE                | LOWER_LEVEL_PEOPLE | LOWER_BOUND_CHILDREN | TOTAL_POPULATION |
|---------|----------------------|--------------------|----------------------|------------------|
| USA     | California           | 5111205            | 1586526              | 39776830         |
| USA     | Texas                | 4936182            | 1494968              | 28704338         |
| USA     | Florida              | 2868884            | 829319               | 21312211         |
| USA     | New York             | 2700018            | 793426               | 19862512         |
| USA     | Pennsylvania         | 1524789            | 430076               | 12823989         |
| USA     | Illinois             | 1542357            | 478646               | 12768328         |
| USA     | Ohio                 | 1551281            | 493856               | 11694664         |
| USA     | Georgia              | 1503877            | 513320               | 10545138         |
| USA     | North Carolina       | 1442873            | 461659               | 10396149         |
| USA     | Michigan             | 1351618            | 404866               | 9991177          |
| USA     | New Jersey           | 862408             | 259622               | 9832872          |
| USA     | Virginia             | 856763             | 247583               | 8525668          |
| USA     | Washington           | 781883             | 219520               | 7530552          |
| USA     | Arizona              | 1802773            | 324723               | 7123898          |
| USA     | Massachusetts        | 675876             | 173590               | 6895917          |
| USA     | Tennessee            | 961862             | 303257               | 6782564          |
| USA     | Indiana              | 841983             | 263205               | 6699629          |
| USA     | Missouri             | 776721             | 240522               | 6135888          |
| USA     | Maryland             | 541127             | 157132               | 6879602          |
| USA     | Wisconsin            | 627866             | 175854               | 5818849          |
| USA     | Colorado             | 552126             | 143486               | 5684203          |
| USA     | Minnesota            | 5085806            | 142348               | 5628162          |
| USA     | South Carolina       | 735356             | 234515               | 5088916          |
| USA     | Alabama              | 784517             | 253674               | 4888949          |
| USA     | Louisiana            | 875678             | 293885               | 4682589          |
| USA     | Kentucky             | 723378             | 211037               | 4472265          |
| USA     | Oregon               | 521125             | 133798               | 4199563          |
| USA     | Oklahoma             | 589591             | 193877               | 3940521          |
| USA     | Connecticut          | 326368             | 89324                | 3588683          |
| USA     | Iowa                 | 319716             | 85662                | 3160553          |
| USA     | Utah                 | 284881             | 90671                | 3159345          |
| USA     | Nevada               | 379792             | 122765               | 3956824          |
| USA     | Arkansas             | 463966             | 149187               | 3820327          |
| USA     | Mississippi          | 561749             | 187869               | 2982785          |
| USA     | Kansas               | 325428             | 97158                | 2918515          |
| USA     | New Mexico           | 378832             | 118269               | 2899708          |
| USA     | Nebraska             | 191439             | 59898                | 1932549          |
| USA     | West Virginia        | 316875             | 83372                | 1803877          |
| USA     | Idaho                | 203631             | 60371                | 1753860          |
| USA     | Hawaii               | 126481             | 32442                | 1426393          |
| USA     | New Hampshire        | 95275              | 22708                | 1356575          |
| USA     | Maine                | 141958             | 32568                | 1341582          |
| USA     | Montana              | 128024             | 32971                | 1862338          |
| USA     | Rhode Island         | 118956             | 34517                | 1861712          |
| USA     | Delaware             | 116181             | 32862                | 971188           |
| USA     | South Dakota         | 103234             | 31772                | 877798           |
| USA     | North Dakota         | 70249              | 17896                | 755238           |
| USA     | Alaska               | 75273              | 23957                | 738068           |
| USA     | District of Columbia | 104185             | 29285                | 703608           |
| USA     | Vermont              | 61032              | 13208                | 623968           |
| USA     | Wyoming              | 57678              | 15694                | 573728           |

51 rows selected.

### 3. How many countries lie under a high Income group?

Answer: **64 Countries** lie under a high income group.

We get the number of countries in a high income group by **cutting the Income table horizontally** with filtering countries in the high income group and then summarizing the records from the Income table by their income groups. We used the following query to solve this question:

```
SQL> select count(*) as "Number of countries",income_group from income where income_group = 'High income' group by income_group;  
Number of countries INCOME_GROUP  
-----  
64 High income  
SQL>
```

### 4. Name the top 5 richest states in the US.

Answer: **California, Texas, Florida, New York, Georgia**

We solve this question by using Complex Query: JOIN with TWO SUBQUERIES

Columns determining the result:

**Upper\_level\_people** : Shows Number of People above poverty line;

**Upper\_Bound\_Children** : Shows Number of Children above poverty line;

First we sorted all the US states by number of upper level people and upper bound children in descending order by using **join and a subquery**.

```
SQL> select * from (select * from (select population.state,population.country,upper_level_people,upper_bound_children  from poverty inner join population on poverty.state = population.state  
| 2 order by upper_level_people desc) order by upper_bound_children desc);  
STATE COUNTRY UPPER_LEVEL_PEOPLE UPPER_BOUND_CHILDREN  
California USA 5217133 1645300  
Texas USA 4124652 1556926  
Florida USA 2942948 872538  
New York USA 2771138 831246  
Georgia USA 1568261 550766  
Ohio USA 1595921 521182  
Illinois USA 1594883 509866  
North Carolina USA 1493191 491217  
Pennsylvania USA 1572405 457254  
Michigan USA 139598 427744  
Arizona USA 1043901 347577  
Tennessee USA 998926 323667  
Louisiana USA 911984 312963  
Indiana USA 879603 285029  
New Jersey USA 901856 281804  
Alabama USA 820809 272124  
Virginia USA 896113 268493  
Missouri USA 809281 259136  
South Carolina USA 765362 251123  
Washington USA 817547 248494  
Kentucky USA 753748 226431  
Oklahoma USA 612231 207017  
Mississippi USA 585685 208883  
Massachusetts USA 710819 191332  
Wisconsin USA 653282 191036  
Maryland USA 571971 173598  
Arkansas USA 487800 162569  
Colorado USA 588118 169312  
Minnesota USA 528899 156088  
Oregon USA 548567 148368  
Nevada USA 405922 135439  
New Mexico USA 399548 129821  
Kansas USA 345322 108566  
Utah USA 307475 103991  
Connecticut USA 360942 101814  
Iowa USA 337542 94838  
West Virginia USA 336129 91339  
Idaho USA 220743 69421  
Nebraska USA 206529 68064  
Rhode Island USA 131466 39665  
Hawaii USA 137921 38762  
Delaware USA 127003 37728  
Montana USA 136208 37675  
Maine USA 152868 37538  
South Dakota USA 111818 36182  
District of Columbia USA 114899 35421  
Alaska USA 83583 28261  
New Hampshire USA 105825 27858  
North Dakota USA 77865 21426  
Wyoming USA 64960 18938  
Vermont USA 69104 16112
```

Secondly, we used another **subquery** and **rownum** on the above shown result to get top 5 records and hence we get the top richest states in the US.

|            |         |                    |                      |
|------------|---------|--------------------|----------------------|
| STATE      | COUNTRY | UPPER_LEVEL_PEOPLE | UPPER_BOUND_CHILDREN |
| California | USA     | 5217133            | 1645300              |
| Texas      | USA     | 4124652            | 1556920              |
| Florida    | USA     | 2942948            | 872538               |
| New York   | USA     | 2771138            | 831246               |
| Georgia    | USA     | 1568261            | 550706               |

**Assumption:** The richer states are the ones which have more people and children above the poverty line than states which have a lesser number of people and children above the poverty line . We have ignored the number of people and children below the poverty line in comparison.

### Steps for Creating an index on table Product for a faster lookup:

We will choose column “Item Type” from product table for Indexing;

Number of records in Product Table having Item Type as “Meat”: 65

To view all indexes of a table, you query from the `all_indexes` view.:

Here the table product has 2 Indexes

|   |
|---|
| SQL> select index_name, index_type, visibility, table_name from all_indexes where table_name='PRODUCT'; |
| INDEX_NAME  |
| INDEX_TYPE  |
| VISIBILITY  |
| TABLE_NAME  |
| ORDER_ID_PK   |
| NORMAL  |
| VISIBLE   |
| PRODUCT   |
| SYS_C007028   |
| NORMAL  |
| VISIBLE   |
| PRODUCT   |

Suppose, you often want to look up members by the last name and you find that the query is quite slow. To speed up the lookup, you create an index for the `Item Type` column:

## You will see that the table Product has 3 Indexes now

```

SQL> select index_name, index_type, visibility, table_name from all_indexes where table_name='PRODUCT';
INDEX_NAME          INDEX_TYPE          VISIBILITY TABLE_NAME
-----  -----  -----
ORDER_ID_PK          NORMAL             VISIBLE    PRODUCT
SYS_C007028          NORMAL             VISIBLE    PRODUCT

SQL> create index item_type_i on product(item_type);

Index created.

SQL> select index_name, index_type, visibility, table_name from all_indexes where table_name='PRODUCT';
INDEX_NAME          INDEX_TYPE          VISIBILITY TABLE_NAME
-----  -----  -----
ORDER_ID_PK          NORMAL             VISIBLE    PRODUCT
SYS_C007028          NORMAL             VISIBLE    PRODUCT
ITEM_TYPE_I          NORMAL             VISIBLE    PRODUCT

SQL> 

```

The following statement finds products whose Item Type is Meat:

```

TCD Chad               Sub-Saharan Africa      Meat           857222878   34
TUR Turkey             Middle East and North Africa  Meat           681298188   48
ARM Armenia            Europe                   Meat           489148938   52
ZWE Zimbabwe            Sub-Saharan Africa      Meat           219834612   141
MNG Mongolia           Asia                     Meat           573378455   142
COG Republic of the Congo Sub-Saharan Africa      Meat           738839423   157
MDA Moldova            Europe                   Meat           971916991   161
KIR Kiribati           Australia and Oceania  Meat           139540883   186
TON Tonga              Australia and Oceania  Meat           355682824   266
VUT Vanuatu            Australia and Oceania  Meat           614994323   216
GIN Guinea             Sub-Saharan Africa      Meat           251621949   233

PRO COUNTRY           REGION                ITEM_TYPE      ORDER_ID PRODUCT_ID
SWZ Swaziland          Sub-Saharan Africa      Meat           947620856   252
CRI Costa Rica         Central America and the Caribbean  Meat           433627212   256
PRK North Korea         Asia                   Meat           277568137   268
GIN Guinea              Sub-Saharan Africa      Meat           807281672   278
MRT Mauritania          Sub-Saharan Africa      Meat           386125295   323
LUX Luxembourg          Europe                  Meat           775727442   338
ZWE Zimbabwe            Sub-Saharan Africa      Meat           855445134   331
GRD Grenada             Central America and the Caribbean  Meat           918334138   374
TUR Turkey              Middle East and North Africa  Meat           867222821   388
TON Tonga              Australia and Oceania  Meat           389678895   389
BEL Belgium             Europe                  Meat           768364982   398

PRO COUNTRY           REGION                ITEM_TYPE      ORDER_ID PRODUCT_ID
LCA Saint Lucia        Central America and the Caribbean  Meat           479447878   394
TTO Tobago              Sub-Saharan Africa      Meat           479447925   398
ZAF South Africa        Sub-Saharan Africa      Meat           287922542   402
CUB Cuba                Central America and the Caribbean  Meat           925504094   424
NIC Nicaragua           Central America and the Caribbean  Meat           652889438   458
MLI Mali                Sub-Saharan Africa      Meat           286014386   494
AND Andorra             Europe                  Meat           834460818   499
GNQ Equatorial Guinea  Sub-Saharan Africa      Meat           565798747   519
EST Estonia             Europe                  Meat           175257527   524
GHA Ghana               Sub-Saharan Africa      Meat           588854308   529
NER Niger               Sub-Saharan Africa      Meat           156619393   558

PRO COUNTRY           REGION                ITEM_TYPE      ORDER_ID PRODUCT_ID
POL Poland              Europe                  Meat           841492497   588
BEN Benin               Sub-Saharan Africa      Meat           348827071   588
BEN Benin               Sub-Saharan Africa      Meat           372845788   586
MDA Moldova             Europe                  Meat           687521983   589
IRL Ireland              Europe                  Meat           287395112   594
RWA Rwanda              Sub-Saharan Africa      Meat           884493243   603
JPN Japan                Asia                     Meat           551720589   621
SLV El Salvador          Central America and the Caribbean  Meat           957123613   622
PAK Pakistan             Asia                     Meat           801893799   652
COM Comoros             Sub-Saharan Africa      Meat           794969689   654

65 rows selected.

SQL> select count(*) as "Number of records" from product where item_type = 'Meat';
Number of records
65
SQL> 

```

**To check if a query uses the index for lookup or not, we follow these steps:**

**First, add the EXPLAIN PLAN FOR clause immediately before the SQL statement:**

```
[SQL> explain plan for select * from product where item_type = 'Meat';
Explained.
SQL> ]
```

**This explains the execution plan into the plan\_table table.**

**Then, use the DBMS\_XPLAN.DISPLAY() procedure to show the content of the plan\_table:**

```
SQL> select plan_table_output from table(dbms_xplan.display());
```

```
PLAN_TABLE_OUTPUT
```

```
-----
```

```
Plan hash value: 427209646
```

| Id   Operation                   | Name    | Rows | Bytes           | Cost (%CPU)     | Time |
|----------------------------------|---------|------|-----------------|-----------------|------|
| 0   SELECT STATEMENT             | PRODUCT | 81   | 4131            | 5 (0)  00:00:01 |      |
| * 1   TABLE ACCESS FULL  PRODUCT | 81      | 4131 | 5 (0)  00:00:01 |                 |      |

```
Predicate Information (identified by operation id):
```

```
PLAN_TABLE_OUTPUT
```

```
-----
```

```
1 - filter("ITEM_TYPE"='Meat')
```

```
13 rows selected.
```

```
SQL> ]
```

**You can see the number of rows are 13 here which are comparatively less to 65 earlier with filter “item\_type” and value “Meat”.**

**Hence this way indexes make retrieval faster**

### **PHASE 3 QUERIES**

**GROUP BY WITH FILTER (Horizontal Cut)**

```
[SQL> select count(*) as
[ 2 "Number of countries" from income where region = 'Sub-Saharan Africa' group by income_group;
Number of countries
-----
26
7
1
12
```

## INNER JOIN QUERY WITH FILTER (Horizontal Cut)

```
[SQL]> Select Immigration.PORT_CODE, Population.STATE, POPULATION.COUNTRY, IMMIGRATION.CROSSING_DATE FROM POPULATION INNER JOIN IMMIGRATION ON POPULATION.STATE = IMMIGRATION.STATE WHERE ROWNUM<=10;

PORT_CODE STATE      COUNTRY CROSSING_DATE
-----  -----
2684 Arizona    USA      01-MAR-19
3389 Montana   USA      01-MAR-19
3322 Montana   USA      01-MAR-19
127 Maine     USA      01-MAR-19
2488 New Mexico USA      01-MAR-19
2586 California USA      01-MAR-19
2584 California USA      01-MAR-19
2684 Arizona   USA      01-MAR-19
2682 Arizona   USA      01-MAR-19
3426 Minnesota USA      01-MAR-19

10 rows selected.
```

## Part 2. System Tables

- The following is the system table for getting all tables, and the tablespace names by **user\_tables**. The query we used is:

```
[SQL]> select table_name,tablespace_name from user_tables
[ 2 ;

TABLE_NAME          TABLESPACE_NAME
-----  -----
INCOME              SYSTEM
PRODUCT             SYSTEM
POPULATION          SYSTEM
IMMIGRATION         SYSTEM
POVERTY              SYSTEM

SQL>
```

- The following is the system table for getting a description of relational tables which are accessible to the user by **all\_tables**. The query we used is:

```
[SQL]> select owner,table_name,tablespace_name,status from all_tables;
OWNER        TABLE_NAME          TABLESPACE_NAME      STATUS      ADHAAR          IMMIGRATION      SYSTEM      VALID
SYS          DUAL               SYSTEM                VALID      ADHAAR          POVERTY          SYSTEM      VALID
SYS          SYSTEM_PRIVILEGE_MAP SYSTEM                VALID      APEX_940000     WHV_FLOW_TEMP_TABLE SYSTEM      VALID
SYS          TABLE_PRIVILEGE_MAP SYSTEM                VALID      APEX_940000     WHV_FLOW_LOV_TEMP SYSTEM      VALID
SYS          STMT_AUDIT_OPTION_MAP SYSTEM               VALID      MDSYS          SDO_TOPO_DATA      VALID
SYS          AUDIT_ACTIONS       SYSTEM               VALID      MDSYS          SDO_TOPO_RELATION_DATA VALID
SYS          WRRS_REPLAY_CALL_FILTER SYSTEM              VALID      MDSYS          SDO_TOPO_TRANSACT_DATA VALID
SYS          HS_BULKLOAD_VIEW_OBJ SYSTEM              VALID      MDSYS          SDO_CS_CONTEXT_INFORMATION VALID
SYS          HS_PARALLEL_METADATA SYSTEM              VALID      MDSYS          SDO_TXN_IDX_EXHD_UPD_RGN VALID
SYS          HS_PARTITION_COL_NAME SYSTEM              VALID      MDSYS          SDO_TXN_IDX_DELETES VALID
SYS          HS_PARTITION_COL_TYPE SYSTEM              VALID      MDSYS          SDO_TXN_IDX_INSERTS VALID
SYSTEM        HELP               SYSTEM               VALID      MDSYS          SDO_ST_TOLERANCE VALID
CTKSYS       DR$OBJECT_ATTRIBUTE SYSAUX              VALID      XDB            XDB$XID_IMP_T      VALID
CTKSYS       DR$POLICY_TAB       SYSAUX              VALID      SYS            KUS_DATAPUMP_MASTER_10_1 VALID
CTKSYS       DR$THS              SYSAUX              VALID      SYS            KUS_DATAPUMP_MASTER_11_1 VALID
CTKSYS       DR$THS_PHRASE       SYSAUX              VALID      SYS            KUS_DATAPUMP_MASTER_11_1_0_7 VALID
MDSYS        SRNSNAMESPACE_TABLE SYSAUX              VALID      SYS            KUS_DATAPUMP_MASTER_11_2 VALID
MDSYS        OGIS_SPATIAL_REFERENCE_SYSTEMS SYSAUX              VALID      SYS            IMPDP_STATS      VALID
MDSYS        OGIS_GEOMETRY_COLUMNS  SYSAUX              VALID      SYS            ODCI_PMO_ROWIDS$ VALID
MDSYS        SDO_UNITS_OF_MEASURE   SYSAUX              VALID      SYS            ODCI_WARNINGS$ VALID
MDSYS        SDO_PRIME_MERIDIANS   SYSAUX              VALID      SYS            ODCI_SECOBJ$ VALID
MDSYS        SDO_ELLIPSOIDS       SYSAUX              VALID      SYS            KUS_LIST_FILTER_TEMP_2 VALID
MDSYS        SDO_DATUMS          SYSAUX              VALID      SYS            KUS_LIST_FILTER_TEMP VALID
MDSYS        SDO_COORD_SYS       SYSAUX              VALID      SYS            KUSNODES          VALID
MDSYS        SDO_COORD_AXIS_NAMES SYSAUX              VALID      SYSTEM          OL$NODES          VALID
MDSYS        SDO_COORD_AXES      SYSAUX              VALID      SYSTEM          OL$HINTS          VALID
MDSYS        SDO_COORD_REF_SYS   SYSAUX              VALID      SYSTEM          OL$              VALID
MDSYS        SDO_COORD_OP_METHODS SYSAUX              VALID      SYS            PLAN_TABLE$      VALID
MDSYS        SDO_COORD_OPS       SYSAUX              VALID      SYS            WRIS_ADV_ASA_RECO_DATA VALID
MDSYS        SDO_PREFERRED_OPS_SYSTEM SYSAUX              VALID      SYS            PSTUBTBL        VALID
MDSYS        SDO_PREFERRED_OPS_USER  SYSAUX              VALID      SYS            76 rows selected.
MDSYS        SDO_COORD_OP_PATHS   SYSAUX              VALID      76
MDSYS        SDO_COORD_OP_PARAMS  SYSAUX              VALID      -----
MDSYS        SDO_COORD_OP_PARAM_USE SYSAUX              VALID      -----
MDSYS        SDO_COORD_OP_PARAM_VALS SYSAUX              VALID      -----
MDSYS        SDO_CS_SNAPSHOT      SYSAUX              VALID      -----
MDSYS        NTV2_XML_DATA       SYSAUX              VALID      -----
MDSYS        SDO_CS_DESIGNGRAPHIC_PLUG_HEIGHT SYSAUX              VALID      -----
MDSYS        SDO_PROJECTIONS_OLD_SNAPSHOT SYSAUX              VALID      -----
MDSYS        SDO_ELLIPSOIDS_OLD_SNAPSHOT SYSAUX              VALID      -----
MDSYS        SDO_DATUMS_OLD_SNAPSHOT SYSAUX              VALID      -----
```

[SQL]> select count(\*) as "Number of Records" from all\_tables;

Number of Records

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

- The following is the system table for getting all information related to constraints by using **all\_constraints**.  
The query we used is:

```
SQL> COLUMN CONSTRAINT_TYPE FORMAT A15;
SQL> select owner,constraint_name,constraint_type from all_constraints;
```

| OWNER  | CONSTRAINT_NAME         | CONSTRAINT_TYPE |
|--------|-------------------------|-----------------|
| SYS    | SYS_C00480              | C               |
| SYS    | SYS_C00479              | C               |
| SYS    | SYS_C00478              | C               |
| SYSTEM | HELP_TOPIC_SEQ          | P               |
| SYSTEM | SYS_C003646             | C               |
| SYSTEM | SYS_C003645             | C               |
| SYS    | SYS_C00477              | C               |
| SYS    | SYS_C00476              | C               |
| SYS    | SYS_C00475              | C               |
| SYS    | SYS_C00474              | C               |
| SYS    | SYS_C00473              | C               |
| SYS    | SYS_C001461             | C               |
| SYS    | SYS_C001460             | C               |
| SYS    | SYS_C003487             | U               |
| SYS    | SYS_C003486             | U               |
| SYS    | HS_PARALLEL_METADATA_PK | P               |
| SYS    | SYS_C003484             | C               |
| SYS    | SYS_C003483             | C               |
| SYS    | SYS_C003482             | C               |
| SYS    | SYS_C003481             | C               |
| SYS    | SYS_C003480             | C               |
| SYS    | SYS_C003479             | C               |
| SYS    | SYS_C003478             | C               |
| CTXSYS | DRCSOAT_NAME            | U               |
| CTXSYS | DRCSOAT_KEY             | P               |
| CTXSYS | SYS_C003684             | U               |
| CTXSYS | SYS_C003683             | P               |
| CTXSYS | SYS_C003682             | C               |
| CTXSYS | SYS_C003681             | C               |
| CTXSYS | SYS_C003680             | C               |
| CTXSYS | SYS_C003679             | C               |

| OWNER  | CONSTRAINT_NAME         | CONSTRAINT_TYPE |
|--------|-------------------------|-----------------|
| ADHAAR | SYS_C006987             | C               |
| ADHAAR | SYS_C006988             | C               |
| ADHAAR | SYS_C007009             | P               |
| ADHAAR | SYS_C007008             | C               |
| ADHAAR | SYS_C007007             | C               |
| ADHAAR | SYS_C007006             | C               |
| ADHAAR | SYS_C007024             | R               |
| ADHAAR | SYS_C007023             | C               |
| ADHAAR | SYS_C007022             | C               |
| ADHAAR | SYS_C007021             | C               |
| ADHAAR | SYS_C007019             | C               |
| ADHAAR | SYS_C007020             | C               |
| ADHAAR | PRODUCT_COUNTRY_CODE_FK | R               |
| ADHAAR | ORDER_ID_PK             | P               |
| ADHAAR | SYS_C006996             | C               |
| ADHAAR | SYS_C006995             | C               |
| ADHAAR | SYS_C006994             | C               |
| ADHAAR | SYS_C006993             | C               |
| ADHAAR | SYS_C006992             | C               |

216 rows selected.

```
SQL> select count(*) as "Number of records" from all_constraints;
```

| Number of records |
|-------------------|
| 216               |

```
SQL> █
```

- The following is the system table for getting constraint definitions on tables in the current user's schema:  
**user\_constraints**

```
SQL> select owner,constraint_name,constraint_type from USER_CONSTRAINTS;
OWNER          CONSTRAINT_NAME           CONSTRAINT_TYPE
-----          -----
ADHAAR          SYS_C006987                C
ADHAAR          SYS_C006988                C
ADHAAR          SYS_C006989                C
ADHAAR          SYS_C006990                C
ADHAAR          COUNTRY_CODE_PK            P
ADHAAR          SYS_C006992                C
ADHAAR          SYS_C006993                C
ADHAAR          SYS_C006994                C
ADHAAR          SYS_C006995                C
ADHAAR          SYS_C006996                C
ADHAAR          ORDER_ID_PK               P
ADHAAR          PRODUCT_COUNTRY_CODE_FK    R
ADHAAR          SYS_C007006                C
ADHAAR          SYS_C007007                C
ADHAAR          SYS_C007008                C
ADHAAR          SYS_C007009                P
ADHAAR          SYS_C007010                C
ADHAAR          SYS_C007011                C
ADHAAR          SYS_C007012                C
ADHAAR          SYS_C007013                C
ADHAAR          SYS_C007014                C
ADHAAR          SYS_C007015                C
ADHAAR          SYS_C007016                C
ADHAAR          SYS_C007019                C
ADHAAR          SYS_C007018                R
ADHAAR          SYS_C007020                C
ADHAAR          SYS_C007021                C
ADHAAR          SYS_C007022                C
ADHAAR          SYS_C007023                C
ADHAAR          SYS_C007024                R
30 rows selected.

SQL>
```

- The following is the system table for getting privileges which the user currently has set under SESSION\_PRIVS. The query we used is:

```
SQL> SELECT * FROM SESSION_PRIVS;

PRIVILEGE
-----
CREATE SESSION
UNLIMITED TABLESPACE
CREATE TABLE
CREATE CLUSTER
CREATE SEQUENCE
CREATE PROCEDURE
CREATE TRIGGER
CREATE TYPE
CREATE OPERATOR
CREATE INDEXTYPE

10 rows selected.

SQL>
```

## Changing Degree of a Table

Degree is defined as the number of threads per instance for scanning the table

**Before:** The degree of the table Poverty is 1

```
[SQL> select table_name,degree from user_tables;
```

| TABLE_NAME  | DEGREE |
|-------------|--------|
| INCOME      | 1      |
| PRODUCT     | 1      |
| POPULATION  | 1      |
| IMMIGRATION | 1      |
| POVERTY     | 1      |

```
SQL> █
```

**After:** The degree of Table Poverty is 8

```
[SQL> select table_name,degree from user_tables;
```

| TABLE_NAME  | DEGREE |
|-------------|--------|
| INCOME      | 1      |
| PRODUCT     | 1      |
| POPULATION  | 1      |
| IMMIGRATION | 1      |
| POVERTY     | 8      |

```
SQL> █
```

## Deleting entry from constraints table

```
[SQL> SELECT TABLE_NAME,CONSTRAINT_NAME,OWNER FROM ALL_CONSTRAINTS WHERE TABLE_NAME = 'POVERTY' AND OWNER = 'ADHAAR';
```

| TABLE_NAME | CONSTRAINT_NAME | OWNER  |
|------------|-----------------|--------|
| POVERTY    | SYS_C007024     | ADHAAR |
| POVERTY    | SYS_C007023     | ADHAAR |
| POVERTY    | SYS_C007022     | ADHAAR |
| POVERTY    | SYS_C007021     | ADHAAR |
| POVERTY    | SYS_C007020     | ADHAAR |
| POVERTY    | SYS_C007019     | ADHAAR |

```
6 rows selected.
```

```
[SQL> DELETE FROM ALL_CONSTRAINTS WHERE TABLE_NAME = 'PRODUCT' AND OWNER = 'ADHAAR' AND CONSTRAINT_NAME = 'ORDER_ID_PK';
DELETE FROM ALL_CONSTRAINTS WHERE TABLE_NAME = 'PRODUCT' AND OWNER = 'ADHAAR' AND CONSTRAINT_NAME = 'ORDER_ID_PK'
```

```
*
```

```
ERROR at line 1:
```

```
ORA-01752: cannot delete from view without exactly one key-preserved table
```

## Part 3: Schema Constraints

### CASCADE DELETE QUERIES:

#### Description:

Parent Table : Income

Primary Key : country\_code

Child Table: Product

Foreign Key : product\_country\_code

Here product\_country\_code references country\_code from parent table.

```
SQL> alter table product
  2  drop constraint product_country_code_fk;
Table altered.

SQL> alter table product
  2
SQL> alter table product
  2  add constraint product_country_code_fk foreign key (product_country_code)
  3  references income(country_code)
  4  on delete cascade;
Table altered.

SQL> 
```

```
SQL> select * from product where product_country_code = 'USA';
PRODUCT_COUNTRY_CODE COUNTRY          REGION        ITEM_TYPE          ORDER_ID
-----  -----  -----  -----  -----
USA      United States of America  North America  Meat            988136594
USA      United States of America  North America  Personal Care    551857326
USA      United States of America  North America  Baby Food       984673964
USA      United States of America  North America  Clothes          699368035
USA      United States of America  North America  Office Supplies  811701095
USA      United States of America  North America  Beverages       799003732

6 rows selected.

SQL> select * from income where country_code = USA;
select * from income where country_code = USA
*
ERROR at line 1:
ORA-00904: "USA": invalid identifier

SQL> select * from income where country_code = 'USA';
COUNTRY_CODE  REGION        INCOME_GROUP      COUNTRY
-----  -----  -----  -----
USA          North America  High income      United States
```

```
SQL> delete from income where country_code = 'USA';
1 row deleted.

SQL> select * from income where country_code = 'USA';
no rows selected

SQL> select * from product where product_country_code = 'USA';
no rows selected

SQL> 
```

Here the columns deleted from parent table income have also deleted columns from child table product by delete cascade;

## Referential Integrity constraint violations:

**Delete:** The income is parent table with country\_code as primary key and product is child table with product\_country\_code as foreign key so deleting a record from parent table is a violation

```
SQL> delete from income where country_code = 'UZB';
delete from income where country_code = 'UZB'
*
ERROR at line 1:
ORA-02292: integrity constraint (TESTUSER.SYS_C007043) violated - child record found

SQL> ■
```

**Update:** The income is parent table with country\_code as primary key and product is child table with product\_country\_code as foreign key so updating a record in parent table is a violation

```
SQL> Update income set country_code = 'IND' where country_code = 'VUT';
Update income set country_code = 'IND' where country_code = 'VUT'
*
ERROR at line 1:
ORA-00001: unique constraint (ADHAAR.COUNTRY_CODE_PK) violated

SQL> ■
```

**Insert:** The income is parent table with country\_code as primary key and product is child table with product\_country\_code as foreign key so inserting a record in child table with no corresponding entry in parent table is a violation

```
SQL> Insert into product(PRODUCT_COUNTRY_CODE,REGION,COUNTRY,ITEM_TYPE,ORDER_ID) values ('MYS','East Asia & Pacific','Malaysia','Meat',600370591);
Enter value for pacific:
old : 1: Insert into product(PRODUCT_COUNTRY_CODE,REGION,COUNTRY,ITEM_TYPE,ORDER_ID) values ('MYS','East Asia & Pacific','Malaysia','Meat',600370591)
new : 1: Insert into product(PRODUCT_COUNTRY_CODE,REGION,COUNTRY,ITEM_TYPE,ORDER_ID) values ('MYS','East Asia ','Malaysia','Meat',600370591)
Insert into product(PRODUCT_COUNTRY_CODE,REGION,COUNTRY,ITEM_TYPE,ORDER_ID) values ('MYS','East Asia ','Malaysia','Meat',600370591)
*
ERROR at line 1:
ORA-02291: integrity constraint (ADHAAR.PRODUCT_COUNTRY_CODE_FK) violated -
parent key not found

SQL> ■
```

## SET RESTRICT AND SET DEFAULT IS NOT SUPPORTED IN ORACLE 11g

Link - [http://www.dba-oracle.com/bk\\_on\\_delete\\_restrict\\_on\\_delete\\_no\\_action\\_tips.htm](http://www.dba-oracle.com/bk_on_delete_restrict_on_delete_no_action_tips.htm)

| sets the department_id column value to null for employees in this department.  |   |   |
|--|---|---|
| Table 5-3 outlines the DML statements allowed by the different referential actions on the key values in the parent table, and the foreign key values in the child table.             |   |   |
| <b>Table 5-3 DML Statements Allowed by Update and Delete No Action</b>   |   |   |
| DML Statement  | Issued Against Parent Table   | Issued Against Child Table  |
| INSERT   | Always OK if the parent key value is unique   | OK only if the foreign key value exists in the parent key or is partially or all null |
| UPDATE NO ACTION   | Allowed if the statement does not leave any rows in the child table without a referenced parent key value | Allowed if the new foreign key value still references a referenced key value          |
| DELETE NO ACTION   | Allowed if no rows in the child table reference the parent key value                                      | Always OK   |
| DELETE CASCADE   | Always OK   | Always OK   |
| DELETE SET NULL  | Always OK   | Always OK   |
| <b>Note:</b><br>Other referential actions not supported by FOREIGN KEY integrity constraints of Oracle Database can be enforced using database triggers. See "Overview of Triggers". |   |   |
| <b>See Also:</b><br><a href="#">Oracle Database SQL Language Reference</a> to learn about the ON DELETE clause   |   |   |

## Part 4: Oracle Sequence Object

To create Oracle Sequence objects we have to follow the following **5 steps**:

**Step 1:** Adding auto increment column “product\_id” to the table product

```
[SQL> ALTER TABLE product
  2 ADD product_id NUMBER(30) UNIQUE;
Table altered.

SQL> ]
```

**Step 2:** Adding sequence object starting from 1

```
[SQL> CREATE SEQUENCE products_sequence;
Sequence created.

SQL> ]
```

```
[SQL> DROP SEQUENCE products_sequence;
Sequence dropped.

[SQL> CREATE SEQUENCE products_sequence START WITH 1;
Sequence created.

SQL> ]
```

**Step 3:** Creating trigger on insert to table product so that auto increment values that can be added for column “product\_id”.

```
[SQL> CREATE OR REPLACE TRIGGER products_on_insert
  2 BEFORE INSERT ON product
  3 FOR EACH ROW
  4 BEGIN
  5 SELECT products_sequence.nextval
  6 INTO :new.product_id
  7 FROM dual;
  8 END;
  9 /
Trigger created.

SQL> ]
```

#### Step 4: Successful loading of csv into table product by using sql loader.

```
[ubuntu@ip-172-31-90-165:~$ sqlldr adhaar/AdhaarChico17 log='product.log' control='load_product_data.ctl'  
SQL*Loader: Release 11.2.0.2.0 - Production on Fri Apr 17 21:43:14 2020  
Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.  
  
Commit point reached - logical record count 64  
Commit point reached - logical record count 128  
Commit point reached - logical record count 192  
Commit point reached - logical record count 256  
Commit point reached - logical record count 320  
Commit point reached - logical record count 384  
Commit point reached - logical record count 448  
Commit point reached - logical record count 512  
Commit point reached - logical record count 576  
Commit point reached - logical record count 640  
Commit point reached - logical record count 704  
Commit point reached - logical record count 768  
Commit point reached - logical record count 832  
Commit point reached - logical record count 896  
Commit point reached - logical record count 960  
Commit point reached - logical record count 1000  
ubuntu@ip-172-31-90-165:~$ ]
```

#### Table PRODUCT:

```
984 Rows successfully loaded.  
16 Rows not loaded due to data errors.  
0 Rows not loaded because all WHEN clauses were failed.  
0 Rows not loaded because all fields were null.
```

```
Space allocated for bind array: 82560 bytes(64 rows)  
Read buffer bytes: 1048576
```

```
Total logical records skipped: 0  
Total logical records read: 1000  
Total logical records rejected: 16  
Total logical records discarded: 0
```

```
Run began on Fri Apr 17 21:43:14 2020  
Run ended on Fri Apr 17 21:43:14 2020
```

```
Elapsed time was: 00:00:00.18  
CPU time was: 00:00:00.01  
ubuntu@ip-172-31-90-165:~$ ]
```

**Step 5: Now you can see the auto increment values for column “product\_id” starting from 1.**

```
SQL> SET PAGESIZE 100
SQL> SET LINESIZE 500
SQL> select * from product order by product_id asc;
```

| PRO_COUNTRY                  | REGION                            | ITEM_TYPE       | ORDER_ID  | PRODUCT_ID |
|------------------------------|-----------------------------------|-----------------|-----------|------------|
| USA United States of America | North America                     | Meat            | 908136594 | 1          |
| USA United States of America | North America                     | Personal Care   | 551057326 | 2          |
| USA United States of America | North America                     | Baby Food       | 984673964 | 3          |
| USA United States of America | North America                     | Clothes         | 699368035 | 4          |
| USA United States of America | North America                     | Office Supplies | 811701895 | 5          |
| USA United States of America | North America                     | Beverages       | 799003732 | 6          |
| USA United States of America | North America                     | Cosmetics       | 686800706 | 7          |
| LBY Libya                    | Middle East and North Africa      | Vegetables      | 185941302 | 8          |
| CAN Canada                   | North America                     | Baby Food       | 246222341 | 9          |
| LBY Libya                    | Middle East and North Africa      | Cereal          | 161442649 | 10         |
| JPN Japan                    | Asia                              | Fruits          | 645713555 | 11         |
| TCD Chad                     | Sub-Saharan Africa                | Cereal          | 683458888 | 12         |
| ARM Armenia                  | Europe                            | Cereal          | 679414975 | 13         |
| ERI Eritrea                  | Sub-Saharan Africa                | Clothes         | 208630645 | 14         |
| MNE Montenegro               | Europe                            | Vegetables      | 266467225 | 15         |
| JAM Jamaica                  | Central America and the Caribbean | Vegetables      | 118598544 | 16         |
| FJI Fiji                     | Australia and Oceania             | Clothes         | 451010930 | 17         |
| TGO Togo                     | Sub-Saharan Africa                | Clothes         | 220003211 | 18         |
| MNE Montenegro               | Europe                            | Snacks          | 702186715 | 19         |
| GRC Greece                   | Europe                            | Household       | 544485270 | 20         |
| SDN Sudan                    | Sub-Saharan Africa                | Cosmetics       | 714135205 | 21         |
| MDV Maldives                 | Asia                              | Fruits          | 448685348 | 22         |
| MNE Montenegro               | Europe                            | Clothes         | 40597025  | 23         |
| EST Estonia                  | Europe                            | Office Supplies | 414244867 | 24         |
| GRL Greenland                | North America                     | Beverages       | 821912801 | 25         |
| CPV Cape Verde               | Sub-Saharan Africa                | Clothes         | 247802054 | 26         |
| SEN Senegal                  | Sub-Saharan Africa                | Household       | 880999934 | 27         |
| BGR Bulgaria                 | Europe                            | Clothes         | 127468717 | 28         |
| DZA Algeria                  | Middle East and North Africa      | Personal Care   | 770478332 | 29         |
| MNG Mongolia                 | Asia                              | Clothes         | 430390107 | 30         |
| GRD Grenada                  | Central America and the Caribbean | Cereal          | 397877871 | 31         |
| GRD Grenada                  | Central America and the Caribbean | Beverages       | 683927953 | 32         |
| SEN Senegal                  | Sub-Saharan Africa                | Beverages       | 469839179 | 33         |
| GRL Greenland                | North America                     | Fruits          | 357222878 | 34         |
| TCD Chad                     | Sub-Saharan Africa                | Meat            | 118002879 | 35         |
| MUS Mauritius                | Sub-Saharan Africa                | Personal Care   | 944415509 | 36         |
| MAR Morocco                  | Middle East and North Africa      | Beverages       | 499009597 | 37         |
| HND Honduras                 | Central America and the Caribbean | Office Supplies | 564646470 | 38         |
| BEN Benin                    | Sub-Saharan Africa                | Fruits          |           |            |

| PRO_COUNTRY              | REGION                       | ITEM_TYPE  | ORDER_ID  | PRODUCT_ID |
|--------------------------|------------------------------|------------|-----------|------------|
| MNG Mongolia             | Asia                         | Clothes    | 308168065 | 975        |
| PLW Palau                | Australia and Oceania        | Household  | 884216010 | 976        |
| MCO Monaco               | Europe                       | Snacks     | 858611428 | 977        |
| FJI Fiji                 | Australia and Oceania        | Cereal     | 903278148 | 978        |
| MLI Mali                 | Sub-Saharan Africa           | Beverages  | 410452497 | 979        |
| LBR Liberia              | Sub-Saharan Africa           | Cereal     | 642683303 | 980        |
| CHE Switzerland          | Europe                       | Beverages  | 682831895 | 981        |
| WSM Samoa                | Australia and Oceania        | Baby Food  | 584872101 | 982        |
| NPL Nepal                | Asia                         | Meat       | 919890248 | 983        |
| AZE Azerbaijan           | Middle East and North Africa | Snacks     | 534085166 | 984        |
| GEO Georgia              | Europe                       | Baby Food  | 590768182 | 985        |
| ARE United Arab Emirates | Middle East and North Africa | Vegetables | 524363124 | 986        |
| FIN Finland              | Europe                       | Household  | 289606320 | 987        |
| PRT Portugal             | Europe                       | Cereal     | 811546599 | 988        |

984 rows selected.

```
SQL> select count(*) as "Number of records" from product;
```

Number of records

```
984
```

```
SQL> ■
```

## Part 5: Multiple Database

In this part we are implementing the access between multiple databases schemas by explaining the steps involved.

Tables of user : Adhaar

```
SQL> SELECT owner, table_name FROM all_tables where owner='ADHAAR';

OWNER          TABLE_NAME
-----          -----
ADHAAR          INCOME
ADHAAR          PRODUCT
ADHAAR          POPULATION
ADHAAR          IMMIGRATION
ADHAAR          POVERTY
```

Created another user : testuser

```
ubuntu@ip-172-31-90-165:~$ sqlplus sys as sysdba

SQL*Plus: Release 11.2.0.2.0 Production on Sat Apr 18 02:56:18 2020

Copyright (c) 1982, 2011, Oracle. All rights reserved.

Enter password:

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL> create user testuser identified by test123;

User created.

SQL> grant connect,resource to test123;
grant connect,resource to test123
*
ERROR at line 1:
ORA-01917: user or role 'TEST123' does not exist

SQL> grant connect,resource to testuser;

Grant succeeded.

SQL> $
```

## Tables of user : testuser

```
SQL> select owner,table_name from all_tables where owner='TESTUSER'  
| 2 ;  
  
OWNER          TABLE_NAME  
-----  
TESTUSER        PRODUCT  
  
SQL> ■
```

```
SQL> describe adhaar.income;  
Name          Null?    Type  
-----  
COUNTRY_CODE  NOT NULL CHAR(3)  
REGION        NOT NULL VARCHAR2(25)  
INCOME_GROUP  NOT NULL VARCHAR2(25)  
COUNTRY       NOT NULL VARCHAR2(20)  
  
SQL> describe testuser.product  
SP2-0734: unknown command beginning "describe te..." - rest of line ignored.  
SQL> describe testuser.product  
Name          Null?    Type  
-----  
PRODUCT_COUNTRY_CODE  NOT NULL CHAR(3)  
COUNTRY        NOT NULL VARCHAR2(25)  
REGION         NOT NULL VARCHAR2(35)  
ITEM_TYPE      NOT NULL VARCHAR2(35)  
ORDER_ID       NUMBER(20)  
PRODUCT_ID     NUMBER(30)  
  
SQL> ■
```

Now we are **granting security grants** to both Income and Product tables. For Income table it's **user:adhaar** and for the product table its **user:testuser**

```
SQL> grant references on testuser.product to adhaar;  
Grant succeeded.  
  
SQL> grant references on adhaar.income to testuser;  
Grant succeeded.  
  
SQL> ■
```

In this step we are **adding foreign key** to the product table which is the child table and income table is the parent in this case.

```
SQL> ALTER TABLE testuser.product  
| 2 ADD FOREIGN KEY (product_country_code)  
| 3 references adhaar.income(country_code);  
  
Table altered.  
  
SQL> ■
```

In this step we are performing **JOIN** on both the tables

| COUNTRY           | INCOME_GROUP        | ITEM_TYPE       |
|-------------------|---------------------|-----------------|
| Canada            | High income         | Vegetables      |
| Japan             | High income         | Cereal          |
| Chad              | Low income          | Fruits          |
| Armenia           | Lower middle income | Cereal          |
| Eritrea           | Low income          | Cereal          |
| Montenegro        | Upper middle income | Clothes         |
| Jamaica           | Upper middle income | Vegetables      |
| Fiji              | Upper middle income | Vegetables      |
| Togo              | Low income          | Clothes         |
| Montenegro        | Upper middle income | Snacks          |
| Greece            | High income         | Household       |
| Sudan             | Lower middle income | Cosmetics       |
| Maldives          | Upper middle income | Fruits          |
| Montenegro        | Upper middle income | Clothes         |
| Estonia           | High income         | Office Supplies |
| Greenland         | High income         | Beverages       |
| Cabo Verde        | Lower middle income | Clothes         |
| Senegal           | Low income          | Household       |
| Bulgaria          | Upper middle income | Clothes         |
| Mongolia          | Lower middle income | Clothes         |
| Grenada           | Upper middle income | Cereal          |
| Grenada           | Upper middle income | Beverages       |
| Senegal           | Low income          | Beverages       |
| Greenland         | High income         | Fruits          |
| Chad              | Low income          | Meat            |
| Mauritius         | Upper middle income | Personal Care   |
| Honduras          | Lower middle income | Office Supplies |
| Benin             | Low income          | Fruits          |
| Greece            | High income         | Baby Food       |
| Jamaica           | Upper middle income | Beverages       |
| Equatorial Guinea | Upper middle income | Office Supplies |

## Part 6: Visualization

### Phase 1 Queries Answered

#### Query 1. How many countries lie under a high income group?

Answer: There are **63 countries** which lie in the High Income Group all over the world. *We used the following query to answer this question.*

| Number of countries | INCOME_GROUP |
|---------------------|--------------|
| 63                  | High income  |

Additionally, the number of countries in the **Upper middle income group** is **48**.

*We used the following query to answer this question.*

| Number of countries | INCOME_GROUP        |
|---------------------|---------------------|
| 48                  | Upper middle income |

The number of countries in the **Lower middle income group** is 45.

We used the following query to answer this question.

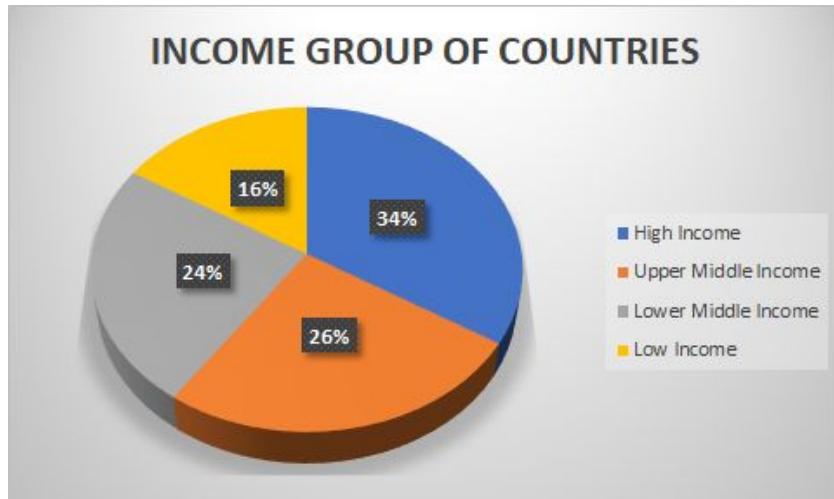
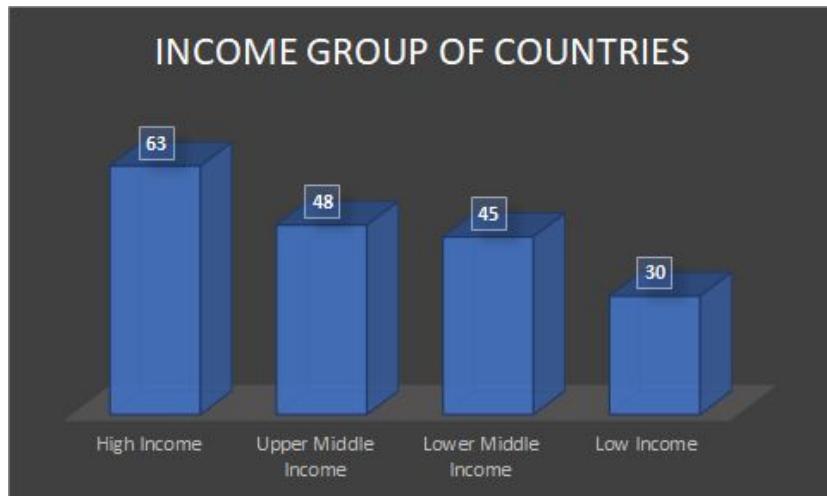
```
SQL> select count(*) as "Number of countries",Income_group from income where income_group = 'Lower middle income' group by Income_group;  
Number of countries INCOME_GROUP  
-----  
45 Lower middle income  
SQL> █
```

The number of countries in the **Low income group** is 30.

We used the following query to answer this question.

```
SQL> select count(*) as "Number of countries",Income_group from income where income_group = 'Low income' group by Income_group;  
Number of countries INCOME_GROUP  
-----  
30 Low income  
SQL> █
```

Bar and Pie graph for the above query are as follows:



## Query 2. Name the top 5 richest states in the US.

Answer - Below are the approximate Percentages of people living in the top 5 richest states in the US

**California - 31%**

**Texas - 25%**

**Florida - 18%**

**New York - 16%**

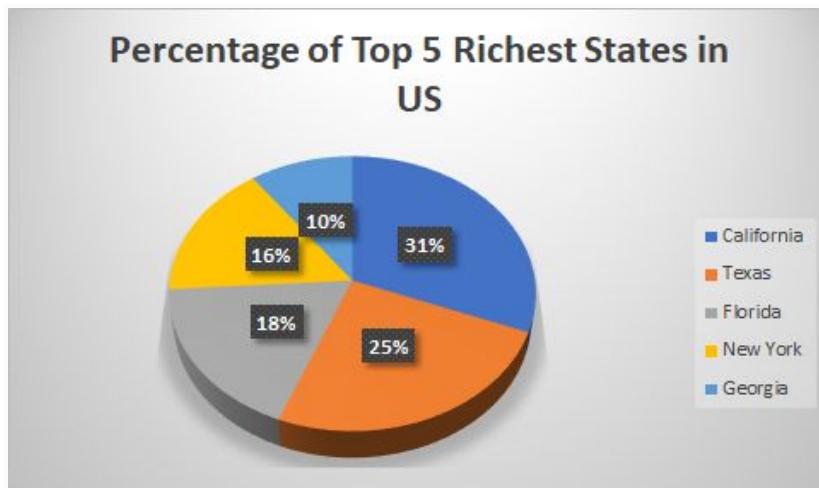
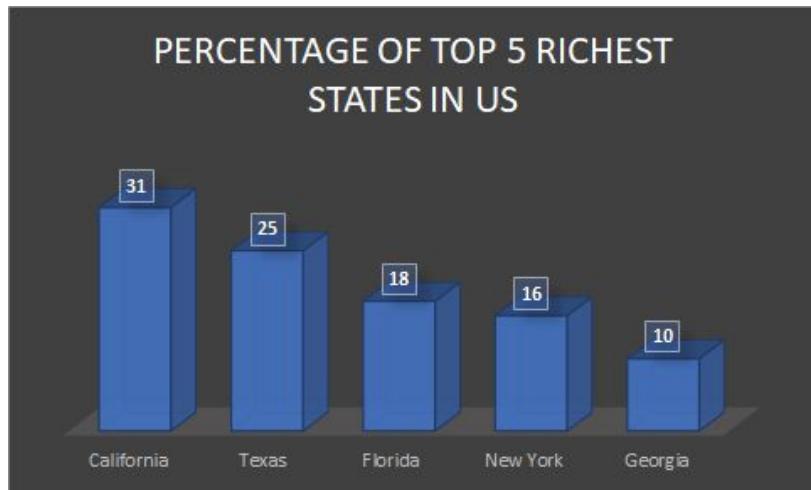
**Georgia - 10%**

*The above analysis is based on the value of Upper\_level\_people column in table*

*Income which represents the number of people living above the poverty line. We used the following query to answer this problem:*

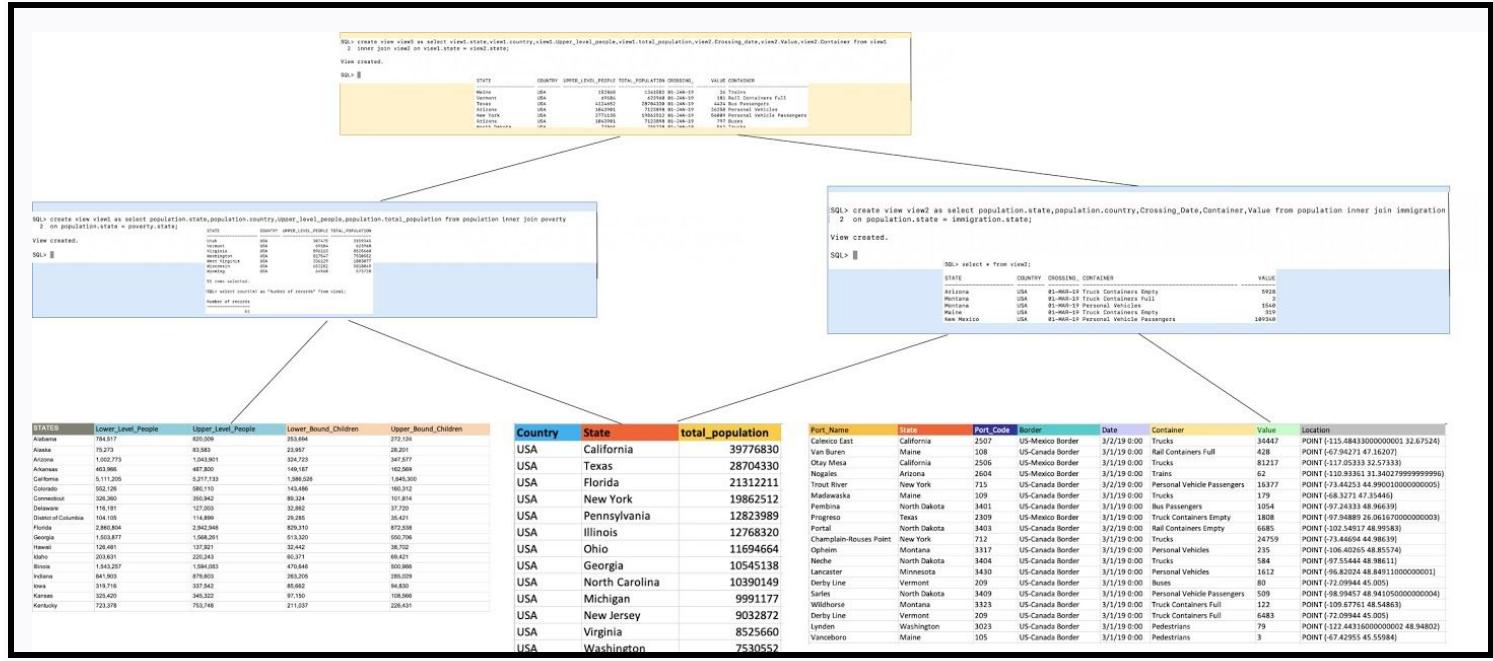
|            |         |                    |                      |
|------------|---------|--------------------|----------------------|
| STATE      | COUNTRY | UPPER_LEVEL_PEOPLE | UPPER_BOUND_CHILDREN |
| California | USA     | 5217133            | 1645300              |
| Texas      | USA     | 4124652            | 1556928              |
| Florida    | USA     | 2942948            | 872538               |
| New York   | USA     | 2771138            | 831246               |
| Georgia    | USA     | 1568261            | 580706               |

Bar graph and pie chart are as follows:



## Part 7: Layered Design with Views

View Diagram shows the **3 datasets on the bottom row**. **View 1** and **View 2** in the middle row and **View 3** on the top row.



We have posted the bigger size screenshots and enlarged images from the above table down below in the following:

**View 1** is created by a **Join of Table Population and Table Poverty**

**View 1 screenshots of Query**

```

SQL> create view view1 as select population.state,population.country,Upper_Level_People,population.Total_Population from population inner join poverty
  2  on population.state = poverty.state;
View created.

SQL>
    
```

## **View 1 screenshot of Output**

|                |         |                    |                  |
|----------------|---------|--------------------|------------------|
| Georgia        | USA     | 1568261            | 10545138         |
| <hr/>          |         |                    |                  |
| STATE          | COUNTRY | UPPER_LEVEL_PEOPLE | TOTAL_POPULATION |
| Hawaii         | USA     | 137921             | 1426393          |
| Idaho          | USA     | 228243             | 1753868          |
| Illinois       | USA     | 1594083            | 12768328         |
| Indiana        | USA     | 879603             | 6699629          |
| Iowa           | USA     | 337542             | 3168553          |
| Kansas         | USA     | 345322             | 2918515          |
| Kentucky       | USA     | 753748             | 4472265          |
| Louisiana      | USA     | 911984             | 4682509          |
| Maine          | USA     | 152868             | 1341582          |
| Maryland       | USA     | 571971             | 6079602          |
| Massachusetts  | USA     | 710810             | 6895917          |
| <hr/>          |         |                    |                  |
| STATE          | COUNTRY | UPPER_LEVEL_PEOPLE | TOTAL_POPULATION |
| Michigan       | USA     | 1395098            | 9991177          |
| Minnesota      | USA     | 528890             | 5628162          |
| Mississippi    | USA     | 585685             | 2982785          |
| Missouri       | USA     | 809281             | 6135888          |
| Montana        | USA     | 136208             | 1062338          |
| Nebraska       | USA     | 206529             | 1932549          |
| Nevada         | USA     | 405922             | 3056824          |
| New Hampshire  | USA     | 105825             | 1350575          |
| New Jersey     | USA     | 901856             | 9032872          |
| New Mexico     | USA     | 399548             | 2090708          |
| New York       | USA     | 2771138            | 19862512         |
| <hr/>          |         |                    |                  |
| STATE          | COUNTRY | UPPER_LEVEL_PEOPLE | TOTAL_POPULATION |
| North Carolina | USA     | 1493191            | 10390149         |
| North Dakota   | USA     | 77865              | 755238           |
| Ohio           | USA     | 1599521            | 11694664         |
| Oklahoma       | USA     | 612231             | 3940521          |
| Oregon         | USA     | 548567             | 4199563          |
| Pennsylvania   | USA     | 1572405            | 12823989         |
| Rhode Island   | USA     | 131460             | 1061712          |
| South Carolina | USA     | 765362             | 5088916          |
| South Dakota   | USA     | 111810             | 877798           |
| Tennessee      | USA     | 998926             | 6782564          |
| Texas          | USA     | 4124652            | 28704330         |
| <hr/>          |         |                    |                  |
| STATE          | COUNTRY | UPPER_LEVEL_PEOPLE | TOTAL_POPULATION |
| Utah           | USA     | 307475             | 3159345          |
| Vermont        | USA     | 69104              | 623968           |
| Virginia       | USA     | 896113             | 8525668          |
| Washington     | USA     | 817547             | 7530552          |
| West Virginia  | USA     | 336129             | 1803077          |
| Wisconsin      | USA     | 653282             | 5818049          |
| Wyoming        | USA     | 64960              | 573728           |

51 rows selected.

|SQL> select count(\*) as "Number of records" from view;

Number of records

51

SQL> █

## **View 2 is view created by a Join of Tables Population and Immigration**

View 2 Screenshot of the Query:

SQL> create view view2 as select population.state,population.country,Crossing\_Date,Container,Value from population inner join immigration

2 on population.state = immigration.state;

View created.

SQL> █

## View 2 screenshot of Output

| STATE        | COUNTRY | CROSSING_CONTAINER                    | VALUE  |
|--------------|---------|---------------------------------------|--------|
| Arizona      | USA     | 01-MAR-19 Truck Containers Empty      | 5928   |
| Montana      | USA     | 01-MAR-19 Truck Containers Full       | 3      |
| Montana      | USA     | 01-MAR-19 Personal Vehicles           | 1540   |
| Maine        | USA     | 01-MAR-19 Truck Containers Empty      | 319    |
| New Mexico   | USA     | 01-MAR-19 Personal Vehicle Passengers | 109340 |
| California   | USA     | 01-MAR-19 Buses                       | 916    |
| California   | USA     | 01-MAR-19 Buses                       | 3329   |
| Arizona      | USA     | 01-MAR-19 Bus Passengers              | 17479  |
| Arizona      | USA     | 01-MAR-19 Personal Vehicles           | 46128  |
| Minnesota    | USA     | 01-MAR-19 Personal Vehicles           | 2658   |
| Alaska       | USA     | 01-MAR-19 Personal Vehicle Passengers | 3077   |
| North Dakota | USA     | 01-MAR-19 Truck Containers Empty      | 26     |
| Maine        | USA     | 01-MAR-19 Trains                      | 28     |
| Minnesota    | USA     | 01-MAR-19 Bus Passengers              | 196    |
| North Dakota | USA     | 01-MAR-19 Buses                       | 9      |
| Maine        | USA     | 01-MAR-19 Bus Passengers              | 13     |
| Minnesota    | USA     | 01-MAR-19 Truck Containers Full       | 697    |
| New York     | USA     | 01-MAR-19 Train Passengers            | 1899   |
| California   | USA     | 01-MAR-19 Pedestrians                 | 346158 |
| New York     | USA     | 01-MAR-19 Personal Vehicle Passengers | 118275 |
| California   | USA     | 01-MAR-19 Pedestrians                 | 306812 |
| Minnesota    | USA     | 01-MAR-19 Truck Containers Empty      | 14     |
| California   | USA     | 01-MAR-19 Truck Containers Full       | 20256  |
| California   | USA     | 01-MAR-19 Truck Containers Full       | 63197  |
| California   | USA     | 01-MAR-19 Truck Containers Empty      | 22225  |
| Arizona      | USA     | 01-MAR-19 Truck Containers Full       | 1639   |
| Minnesota    | USA     | 01-MAR-19 Personal Vehicle Passengers | 30864  |
| North Dakota | USA     | 01-MAR-19 Pedestrians                 | 7      |
| Michigan     | USA     | 01-MAR-19 Rail Containers Empty       | 16839  |
| North Dakota | USA     | 01-MAR-19 Truck Containers Full       | 54     |
| North Dakota | USA     | 01-MAR-19 Personal Vehicles           | 20513  |
| North Dakota | USA     | 01-MAR-19 Truck Containers Empty      | 328    |
| Washington   | USA     | 01-MAR-19 Truck Containers Full       | 129    |
| New Mexico   | USA     | 01-MAR-19 Truck Containers Full       | 8698   |

|              |     |                                       |        |
|--------------|-----|---------------------------------------|--------|
| Maine        | USA | 01-JAN-19 Trains                      | 26     |
| Vermont      | USA | 01-JAN-19 Rail Containers Full        | 181    |
| Texas        | USA | 01-JAN-19 Bus Passengers              | 4424   |
| Arizona      | USA | 01-JAN-19 Personal Vehicles           | 26258  |
| New York     | USA | 01-JAN-19 Personal Vehicle Passengers | 56089  |
| Arizona      | USA | 01-JAN-19 Buses                       | 797    |
| North Dakota | USA | 01-JAN-19 Trucks                      | 562    |
| Michigan     | USA | 01-JAN-19 Personal Vehicle Passengers | 478349 |
| California   | USA | 01-JAN-19 Personal Vehicles           | 48673  |
| Montana      | USA | 01-JAN-19 Truck Containers Full       | 138    |
| Maine        | USA | 01-JAN-19 Rail Containers Full        | 40     |
| New York     | USA | 01-JAN-19 Personal Vehicles           | 28098  |
| Minnesota    | USA | 01-JAN-19 Personal Vehicles           | 6067   |
| North Dakota | USA | 01-JAN-19 Personal Vehicles           | 724    |
| California   | USA | 01-JAN-19 Personal Vehicle Passengers | 169681 |
| California   | USA | 01-JAN-19 Bus Passengers              | 8320   |
| North Dakota | USA | 01-JAN-19 Truck Containers Empty      | 7      |
| Michigan     | USA | 01-JAN-19 Trains                      | 150    |
| Arizona      | USA | 01-JAN-19 Buses                       | 34     |
| New York     | USA | 01-JAN-19 Trucks                      | 1081   |
| Vermont      | USA | 01-JAN-19 Rail Containers Empty       | 235    |
| Vermont      | USA | 01-JAN-19 Personal Vehicles           | 25796  |
| Montana      | USA | 01-JAN-19 Personal Vehicle Passengers | 1187   |
| Vermont      | USA | 01-JAN-19 Personal Vehicles           | 2820   |
| New York     | USA | 01-JAN-19 Trucks                      | 73144  |
| Maine        | USA | 01-JAN-19 Trucks                      | 15     |
| California   | USA | 01-JAN-19 Personal Vehicle Passengers | 95295  |
| North Dakota | USA | 01-JAN-19 Truck Containers Full       | 16593  |
| Idaho        | USA | 01-JAN-19 Truck Containers Empty      | 633    |
| Michigan     | USA | 01-JAN-19 Rail Containers Empty       | 1881   |
| Texas        | USA | 01-JAN-19 Trucks                      | 841    |
| Washington   | USA | 01-JAN-19 Personal Vehicle Passengers | 1156   |
| Texas        | USA | 01-JAN-19 Truck Containers Full       | 12834  |
| Maine        | USA | 01-JAN-19 Personal Vehicles           | 1355   |
| Montana      | USA | 01-JAN-19 Truck Containers Empty      | 18     |
| Washington   | USA | 01-JAN-19 Rail Containers Full        | 9614   |
| Vermont      | USA | 01-JAN-19 Truck Containers Full       | 30     |
| California   | USA | 01-JAN-19 Personal Vehicles           | 90090  |
| New York     | USA | 01-JAN-19 Truck Containers Empty      | 1529   |
| Maine        | USA | 01-JAN-19 Trains                      | 10     |
| Montana      | USA | 01-JAN-19 Trucks                      | 797    |
| Montana      | USA | 01-JAN-19 Personal Vehicle Passengers | 385    |
| Vermont      | USA | 01-JAN-19 Rail Containers Full        | 1263   |
| Washington   | USA | 01-JAN-19 Trucks                      | 865    |
| Minnesota    | USA | 01-JAN-19 Truck Containers Empty      | 84     |
| Washington   | USA | 01-JAN-19 Truck Containers Empty      | 674    |
| Washington   | USA | 01-JAN-19 Truck Containers Empty      | 4      |
| Montana      | USA | 01-JAN-19 Bus Passengers              | 212    |
| Washington   | USA | 01-JAN-19 Trucks                      | 12137  |
| North Dakota | USA | 01-JAN-19 Personal Vehicles           | 2160   |
| North Dakota | USA | 01-JAN-19 Personal Vehicles           | 84     |
| Maine        | USA | 01-JAN-19 Truck Containers Empty      | 146    |

1798 rows selected.

SQL> select count(\*) as "Number of records" from view2;

Number of records

-----  
1798

SQL> ■

## View 3 is a join of View1 and View2

### View 3 Screenshot of Query

```
SQL> create view view3 as select view1.state,view1.country,view1.upper_level_people,view1.total_population,view2.crossing_date,view2.value,view2.container from view1
  2 inner join view2 on view1.state = view2.state;
```

View created.

SQL> ■

### View 3 screenshot of Output

| STATE        | COUNTRY | UPPER_LEVEL_PEOPLE | TOTAL_POPULATION | CROSSING_DATE | VALUE   | CONTAINER                   |
|--------------|---------|--------------------|------------------|---------------|---------|-----------------------------|
| Arizona      | USA     | 3043901            | 7123890          | 01-MAR-19     | 5928    | Truck Containers Empty      |
| Montana      | USA     | 1362688            | 1862330          | 01-MAR-19     | 5       | Truck Containers Full       |
| Montana      | USA     | 1362688            | 1862330          | 01-MAR-19     | 1548    | Personal Vehicles           |
| Maine        | USA     | 152868             | 1341582          | 01-MAR-19     | 319     | Truck Containers Empty      |
| New Mexico   | USA     | 395948             | 1093462          | 01-MAR-19     | 1093462 | Personal Vehicle Passengers |
| California   | USA     | 5217133            | 39778380         | 01-MAR-19     | 916     | Buses                       |
| California   | USA     | 5217133            | 39778380         | 01-MAR-19     | 3329    | Buses                       |
| Alaska       | USA     | 1043901            | 19862512         | 01-MAR-19     | 1219    | Train Passengers            |
| Arizona      | USA     | 1043901            | 7123890          | 01-MAR-19     | 46128   | Personal Vehicles           |
| Minnesota    | USA     | 528898             | 6428162          | 01-MAR-19     | 2658    | Personal Vehicles           |
| Alaska       | USA     | 817547             | 1043901          | 01-MAR-19     | 307     | Personal Vehicle Passengers |
| North Dakota | USA     | 77865              | 7552328          | 01-MAR-19     | 26      | Truck Containers Empty      |
| Maine        | USA     | 152868             | 1341582          | 01-MAR-19     | 28      | Train Passengers            |
| Minnesota    | USA     | 528898             | 1043901          | 01-MAR-19     | 194     | Bus Passengers              |
| North Dakota | USA     | 77865              | 7552328          | 01-MAR-19     | 9       | Buses                       |
| Maine        | USA     | 152868             | 1341582          | 01-MAR-19     | 13      | Bus Passengers              |
| Minnesota    | USA     | 528898             | 1043901          | 01-MAR-19     | 497     | Personal Vehicle Passengers |
| New York     | USA     | 2771138            | 19862512         | 01-MAR-19     | 1899    | Train Passengers            |
| California   | USA     | 5217133            | 39778380         | 01-MAR-19     | 346158  | Pedestrians                 |
| North Dakota | USA     | 77865              | 7552328          | 01-MAR-19     | 12072   | Personal Vehicle Passengers |
| California   | USA     | 5217133            | 39778380         | 01-MAR-19     | 306812  | Pedestrians                 |
| Minnesota    | USA     | 528898             | 6428162          | 01-MAR-19     | 14      | Truck Containers Empty      |
| Califonia    | USA     | 5217133            | 39778380         | 01-MAR-19     | 20856   | Personal Vehicles           |
| California   | USA     | 5217133            | 39778380         | 01-MAR-19     | 63197   | Truck Containers Full       |
| California   | USA     | 5217133            | 39778380         | 01-MAR-19     | 22225   | Truck Containers Empty      |
| Alaska       | USA     | 1043901            | 19862512         | 01-MAR-19     | 1493    | Personal Vehicle Passengers |
| Minnesota    | USA     | 528898             | 6428162          | 01-MAR-19     | 30864   | Personal Vehicle Passengers |
| North Dakota | USA     | 77865              | 7552328          | 01-MAR-19     | 7       | Pedestrians                 |
| Maine        | USA     | 152868             | 1341582          | 01-MAR-19     | 16890   | Truck Containers Empty      |
| Minnesota    | USA     | 528898             | 1043901          | 01-MAR-19     | 54      | Truck Containers Full       |
| North Dakota | USA     | 77865              | 7552328          | 01-MAR-19     | 28513   | Personal Vehicles           |
| North Dakota | USA     | 77865              | 7552328          | 01-MAR-19     | 298     | Train Containers Empty      |
| Washington   | USA     | 817547             | 7588528          | 01-MAR-19     | 129     | Truck Containers Full       |
| New Mexico   | USA     | 395948             | 1043901          | 01-MAR-19     | 8698    | Truck Containers Full       |
| Vermont      | USA     | 619104             | 19862512         | 01-MAR-19     | 471     | Personal Vehicle Passengers |
| Texas        | USA     | 4124652            | 20784530         | 01-MAR-19     | 1       | Buses                       |
| New York     | USA     | 2771138            | 19862512         | 01-MAR-19     | 682     | Rail Containers Empty       |
| California   | USA     | 5217133            | 39778380         | 01-MAR-19     | 28859   | Personal Vehicle Passengers |
| Michigan     | USA     | 1395989            | 9991177          | 01-MAR-19     | 92794   | Truck Containers Full       |
| Maine        | USA     | 152868             | 1341582          | 01-MAR-19     | 8       | Buses                       |
| California   | USA     | 5217133            | 39778380         | 01-MAR-19     | 9285    | Personal Vehicles           |
| North Dakota | USA     | 77865              | 7552328          | 01-MAR-19     | 14112   | Personal Vehicles           |
| Montana      | USA     | 1362688            | 1043901          | 01-MAR-19     | 35      | Trucks                      |
| Idaho        | USA     | 152868             | 1341582          | 01-MAR-19     | 318     | Trucks                      |
| Montana      | USA     | 1362688            | 1043901          | 01-MAR-19     | 1235    | Personal Vehicle Passengers |
| North Dakota | USA     | 77865              | 7552328          | 01-MAR-19     | 2837    | Personal Vehicle Passengers |
| Texas        | USA     | 4124652            | 20784530         | 01-MAR-19     | 814     | Truck Containers Empty      |
| North Dakota | USA     | 77865              | 7552328          | 01-MAR-19     | 173     | Rail Containers Empty       |
| Washington   | USA     | 817547             | 7588528          | 01-MAR-19     | 19      | Truck Containers Empty      |
| Washington   | USA     | 817547             | 7588528          | 01-MAR-19     | 562     | Trucks                      |
| North Dakota | USA     | 77865              | 7552328          | 01-MAR-19     | 169681  | Personal Vehicle Passengers |
| Maine        | USA     | 152868             | 1341582          | 01-MAR-19     | 8328    | Bus Passengers              |
| North Dakota | USA     | 77865              | 7552328          | 01-MAR-19     | 7       | Truck Containers Empty      |
| Michigan     | USA     | 1395989            | 9991177          | 01-MAR-19     | 150     | Trains                      |
| California   | USA     | 5217133            | 39778380         | 01-MAR-19     | 24      | Buses                       |
| Arizona      | USA     | 1843901            | 7123890          | 01-MAR-19     | 1081    | Trucks                      |
| New York     | USA     | 2771138            | 19862512         | 01-MAR-19     | 235     | Rail Containers Empty       |
| Vermont      | USA     | 619104             | 623960           | 01-JAN-19     | 623960  | Personal Vehicles           |
| Vermont      | USA     | 619104             | 623960           | 01-JAN-19     | 25796   | Personal Vehicles           |
| Montana      | USA     | 1362688            | 1062330          | 01-JAN-19     | 1187    | Personal Vehicle Passengers |
| North Dakota | USA     | 77865              | 7552328          | 01-JAN-19     | 2820    | Personal Vehicles           |
| California   | USA     | 5217133            | 39778380         | 01-JAN-19     | 73144   | Trucks                      |
| California   | USA     | 5217133            | 39778380         | 01-JAN-19     | 1156    | Personal Vehicle Passengers |
| North Dakota | USA     | 77865              | 7552328          | 01-JAN-19     | 12834   | Rail Containers Full        |
| Maine        | USA     | 152868             | 1341582          | 01-JAN-19     | 1341582 | Personal Vehicles           |
| Michigan     | USA     | 1395989            | 9991177          | 01-JAN-19     | 95295   | Personal Vehicle Passengers |
| Arizona      | USA     | 1843901            | 7123890          | 01-JAN-19     | 16593   | Truck Containers Full       |
| New York     | USA     | 2771138            | 19862512         | 01-JAN-19     | 633     | Truck Containers Empty      |
| Vermont      | USA     | 619104             | 623960           | 01-JAN-19     | 1881    | Rail Containers Empty       |
| Montana      | USA     | 1362688            | 1062330          | 01-JAN-19     | 841     | Trucks                      |
| North Dakota | USA     | 77865              | 7552328          | 01-JAN-19     | 90098   | Personal Vehicles           |
| Texas        | USA     | 4124652            | 20784530         | 01-JAN-19     | 1529    | Truck Containers Empty      |
| Washington   | USA     | 817547             | 7588528          | 01-JAN-19     | 10      | Trains                      |
| Texas        | USA     | 4124652            | 20784530         | 01-JAN-19     | 1355    | Personal Vehicles           |
| Maine        | USA     | 152868             | 1341582          | 01-JAN-19     | 18      | Truck Containers Empty      |
| Montana      | USA     | 1362688            | 1062330          | 01-JAN-19     | 9614    | Rail Containers Full        |
| Washington   | USA     | 817547             | 7588528          | 01-JAN-19     | 30      | Truck Containers Full       |
| Vermont      | USA     | 619104             | 623960           | 01-JAN-19     | 90098   | Personal Vehicles           |
| California   | USA     | 5217133            | 39778380         | 01-JAN-19     | 1529    | Truck Containers Empty      |
| New York     | USA     | 2771138            | 19862512         | 01-JAN-19     | 797     | Trucks                      |
| Maine        | USA     | 152868             | 1341582          | 01-JAN-19     | 385     | Personal Vehicle Passengers |
| Montana      | USA     | 1362688            | 1062330          | 01-JAN-19     | 1263    | Rail Containers Full        |
| Montana      | USA     | 1362688            | 1062330          | 01-JAN-19     | 865     | Trucks                      |
| Vermont      | USA     | 619104             | 623960           | 01-JAN-19     | 84      | Truck Containers Empty      |
| Washington   | USA     | 817547             | 7588528          | 01-JAN-19     | 674     | Truck Containers Empty      |
| Minnesota    | USA     | 528898             | 628162           | 01-JAN-19     | 4       | Truck Containers Empty      |
| Washington   | USA     | 817547             | 7588528          | 01-JAN-19     | 212     | Bus Passengers              |
| Montana      | USA     | 1362688            | 1062330          | 01-JAN-19     | 12137   | Trucks                      |
| Washington   | USA     | 817547             | 7588528          | 01-JAN-19     | 2160    | Personal Vehicles           |
| North Dakota | USA     | 77865              | 7552328          | 01-JAN-19     | 84      | Personal Vehicles           |
| North Dakota | USA     | 77865              | 7552328          | 01-JAN-19     | 146     | Truck Containers Empty      |

1798 rows selected.

```
SQL> select count(*) as "Number of records" from view3;
```

Number of records

1798

SQL> ■

Immigration Table has a record with state as Maine and Value 146 and has 1798 records

Screenshot of the output

|                         |              |      |                  |           |                             |   |
|-------------------------|--------------|------|------------------|-----------|-----------------------------|---|
| Northgate               | North Dakota | 3406 | US-Canada Border | 01-JAN-19 | Personal Vehicles           | 724 POINT (-102.39634 48.89848)           |
| Tecate                  | California   | 2565 | US-Mexico Border | 01-JAN-19 | Personal Vehicle Passengers | 169681 POINT (-116.62667 32.57722)        |
| Calexico East           | California   | 2507 | US-Mexico Border | 01-JAN-19 | Bus Passengers              | 8320 POINT (-115.48433000000001 32.67524) |
| Hannah                  | North Dakota | 3408 | US-Canada Border | 01-JAN-19 | Truck Containers Empty      | 7 POINT (-98.701 48.97029)                |
| Detroit                 | Michigan     | 3801 | US-Canada Border | 01-JAN-19 | Trains                      | 158 POINT (-83.10222 42.38306000000001)   |
| Douglas                 | Arizona      | 2601 | US-Mexico Border | 01-JAN-19 | Buses                       | 34 POINT (-109.54472 31.344439999999995)  |
| Trout River             | New York     | 715  | US-Canada Border | 01-JAN-19 | Trucks                      | 1081 POINT (-73.44253 44.99801000000001)  |
| Norton                  | Vermont      | 211  | US-Canada Border | 01-JAN-19 | Rail Containers Empty       | 235 POINT (-71.79528000000002 45.01)      |
| Derby Line              | Vermont      | 209  | US-Canada Border | 01-JAN-19 | Personal Vehicles           | 28796 POINT (-72.89944 45.0065)           |
| Wildhorse               | Montana      | 3323 | US-Canada Border | 01-JAN-19 | Personal Vehicle Passengers | 1187 POINT (-109.67761 48.54863)          |
| Beecher Falls           | Vermont      | 206  | US-Canada Border | 01-JAN-19 | Personal Vehicles           | 2824 POINT (-71.49664 45.00888)           |
| Buffalo-Niagara Falls   | New York     | 981  | US-Canada Border | 01-JAN-19 | Trucks                      | 73144 POINT (-79.05694 43.89444)          |
| Limestone               | Maine        | 118  | US-Canada Border | 01-JAN-19 | Trucks                      | 15 POINT (-67.82788 46.99941)             |
| Andrade                 | California   | 2502 | US-Mexico Border | 01-JAN-19 | Personal Vehicle Passengers | 95295 POINT (-114.63402 32.7394)          |
| Pembina                 | North Dakota | 3401 | US-Canada Border | 01-JAN-19 | Truck Containers Full       | 16593 POINT (-97.24333 48.96639)          |
| Eastport                | Idaho        | 3302 | US-Canada Border | 01-JAN-19 | Truck Containers Empty      | 633 POINT (-116.1802799999998 48.99944)   |
| Sault Sainte Marie      | Michigan     | 3803 | US-Canada Border | 01-JAN-19 | Rail Containers Empty       | 1881 POINT (-84.29889 46.47028)           |
| Roma                    | Texas        | 2310 | US-Mexico Border | 01-JAN-19 | Trucks                      | 841 POINT (-99.81833 26.49361)            |
| Anacortes               | Washington   | 3010 | US-Canada Border | 01-JAN-19 | Personal Vehicle Passengers | 1156 POINT (-122.6173899999999 48.49988)  |
| Eagle Pass              | Texas        | 2303 | US-Mexico Border | 01-JAN-19 | Truck Containers Full       | 12834 POINT (-100.49917 28.70889)         |
| Vanceboro               | Maine        | 185  | US-Canada Border | 01-JAN-19 | Personal Vehicles           | 1355 POINT (-67.42955 45.55984)           |
| Scobey                  | Montana      | 3309 | US-Canada Border | 01-JAN-19 | Truck Containers Empty      | 18 POINT (-105.44272 48.88707)            |
| Blaine                  | Washington   | 3004 | US-Canada Border | 01-JAN-19 | Rail Containers Full        | 9614 POINT (-122.74583 48.99389)          |
| Beecher Falls           | Vermont      | 206  | US-Canada Border | 01-JAN-19 | Truck Containers Full       | 38 POINT (-71.49664 45.00888)             |
| Tecate                  | California   | 2505 | US-Mexico Border | 01-JAN-19 | Personal Vehicles           | 98099 POINT (-116.62667 32.57722)         |
| Alexandria Bay          | New York     | 708  | US-Canada Border | 01-JAN-19 | Truck Containers Empty      | 1529 POINT (-75.91886 44.33583)           |
| Calais                  | Maine        | 115  | US-Canada Border | 01-JAN-19 | Trains                      | 10 POINT (-67.27917 45.18889)             |
| Roosville               | Montana      | 3318 | US-Canada Border | 01-JAN-19 | Trucks                      | 797 POINT (-115.04512000000001 48.88171)  |
| Willow Creek            | Montana      | 3325 | US-Canada Border | 01-JAN-19 | Personal Vehicle Passengers | 385 POINT (-109.67761 48.54863)           |
| Highgate Springs-Alburg | Vermont      | 212  | US-Canada Border | 01-JAN-19 | Rail Containers Full        | 1263 POINT (-73.18583 44.97944000000004)  |
| Point Roberts           | Washington   | 3017 | US-Canada Border | 01-JAN-19 | Trucks                      | 865 POINT (-123.07994 48.96978)           |
| Roseau                  | Minnesota    | 3426 | US-Canada Border | 01-JAN-19 | Truck Containers Empty      | 84 POINT (-95.81074 48.77518)             |
| Point Roberts           | Washington   | 3017 | US-Canada Border | 01-JAN-19 | Truck Containers Empty      | 674 POINT (-123.07994 48.96978)           |
| Ferry                   | Washington   | 3013 | US-Canada Border | 01-JAN-19 | Truck Containers Empty      | 4 POINT (-118.58774 48.88627)             |
| Roosville               | Montana      | 3318 | US-Canada Border | 01-JAN-19 | Bus Passengers              | 212 POINT (-115.04512000000001 48.88171)  |
| Sumas                   | Washington   | 3009 | US-Canada Border | 01-JAN-19 | Trucks                      | 12137 POINT (-122.26361 49.00028)         |
| Walhalla                | North Dakota | 3407 | US-Canada Border | 01-JAN-19 | Personal Vehicles           | 2160 POINT (-97.91778 48.92333)           |
| Ambrose                 | North Dakota | 3410 | US-Canada Border | 01-JAN-19 | Personal Vehicles           | 84 POINT (-103.48222 48.95389)            |
| Madawaska               | Maine        | 109  | US-Canada Border | 01-JAN-19 | Truck Containers Empty      | 146 POINT (-68.3271 47.35446)             |

1798 rows selected.

SQL> select count(\*) from immigration;

COUNT(\*)

-----  
1798

SQL> select count(\*) as "Number of Records" from immigration;

Number of Records

-----  
1798

SQL>

**View 3** is derived from Immigration table that has a record with state Maine and value **146** and **1798** records

Screenshot of the the output are as follows:

```
Maine          USA        152868    1341582 01-JAN-19      26 Trains
Vermont        USA        69104     623960 01-JAN-19      181 Rail Containers Full
Texas          USA        4124652   28704330 01-JAN-19     4424 Bus Passengers
Arizona        USA        1043901   7123898 01-JAN-19      26258 Personal Vehicles
New York       USA        2771138   19862512 01-JAN-19      56809 Personal Vehicle Passengers
Arizona        USA        1043901   7123898 01-JAN-19      797 Buses
North Dakota   USA        77865     755238 01-JAN-19      562 Trucks
Michigan       USA        1395098   9991177 01-JAN-19      478349 Personal Vehicle Passengers
California     USA        5217133   39776830 01-JAN-19     48673 Personal Vehicles
Montana        USA        136208    1862338 01-JAN-19      138 Truck Containers Full
Maine          USA        152868    1341582 01-JAN-19      48 Rail Containers Full
New York       USA        2771138   19862512 01-JAN-19      28098 Personal Vehicles
Minnesota      USA        528890    5628162 01-JAN-19      6067 Personal Vehicles
North Dakota   USA        77865     755238 01-JAN-19      724 Personal Vehicles
California     USA        5217133   39776830 01-JAN-19      169601 Personal Vehicle Passengers
California     USA        5217133   39776830 01-JAN-19      8328 Bus Passengers
North Dakota   USA        77865     755238 01-JAN-19      7 Truck Containers Empty
Michigan       USA        1395098   9991177 01-JAN-19      150 Trains
Arizona        USA        1043901   7123898 01-JAN-19      34 Buses
New York       USA        2771138   19862512 01-JAN-19      1081 Trucks
Vermont        USA        69104     623960 01-JAN-19      238 Rail Containers Empty
Vermont        USA        69104     623960 01-JAN-19      25796 Personal Vehicles
Montana        USA        136208    1862330 01-JAN-19      1187 Personal Vehicle Passengers
Vermont        USA        69104     623960 01-JAN-19      2820 Personal Vehicles
New York       USA        2771138   19862512 01-JAN-19      73144 Trucks
Maine          USA        152868    1341582 01-JAN-19      15 Trucks
California     USA        5217133   39776830 01-JAN-19      95295 Personal Vehicle Passengers
North Dakota   USA        77865     755238 01-JAN-19      16593 Truck Containers Full
Idaho          USA        229243    1753860 01-JAN-19      633 Truck Containers Empty
Michigan       USA        1395098   9991177 01-JAN-19      1881 Rail Containers Empty
Texas          USA        4124652   28704330 01-JAN-19      841 Trucks
Washington     USA        817547    7530552 01-JAN-19      1156 Personal Vehicle Passengers
Texas          USA        4124652   28704330 01-JAN-19      12834 Truck Containers Full
Maine          USA        152868    1341582 01-JAN-19      1355 Personal Vehicles
Montana        USA        136208    1862330 01-JAN-19      18 Truck Containers Empty
Washington     USA        817547    7530552 01-JAN-19      9614 Rail Containers Full
Vermont        USA        69104     623960 01-JAN-19      30 Truck Containers Full
California     USA        5217133   39776830 01-JAN-19      90098 Personal Vehicles
New York       USA        2771138   19862512 01-JAN-19      1529 Truck Containers Empty
Maine          USA        152868    1341582 01-JAN-19      10 Trains
Montana        USA        136208    1862330 01-JAN-19      797 Trucks
Montana        USA        136208    1862330 01-JAN-19      385 Personal Vehicle Passengers
Vermont        USA        69104     623960 01-JAN-19      1263 Rail Containers Full
Washington     USA        817547    7530552 01-JAN-19      865 Trucks
Minnesota      USA        528890    5628162 01-JAN-19      84 Truck Containers Empty
Washington     USA        817547    7530552 01-JAN-19      674 Truck Containers Empty
Washington     USA        817547    7530552 01-JAN-19      4 Truck Containers Empty
Montana        USA        136208    1862330 01-JAN-19      212 Bus Passengers
Washington     USA        817547    7530552 01-JAN-19      12137 Trucks
North Dakota   USA        77865     755238 01-JAN-19      2168 Personal Vehicles
North Dakota   USA        77865     755238 01-JAN-19      84 Personal Vehicles
Maine          USA        152868    1341582 01-JAN-19      146 Truck Containers Empty

1798 rows selected.

SQL> select count(*) as "Number of records" from view3;
Number of records
-----
1798

SQL>
```

After deleting the record from table with state maine and value 146 from table immigration-

```
SQL> delete from immigration where State='Maine' and value=146;
1 row deleted.

Walhalla      North Dakota      3407 US-Canada Border      01-JAN-19 Personal Vehicles      2168 POINT (-97.91778 48.92333)
Ambrose       North Dakota      3410 US-Canada Border      01-JAN-19 Personal Vehicles      84 POINT (-103.48222 48.95389)

1797 rows selected.

SQL> select count(*) from immigration as "Number of Records";
select count(*) from immigration as "Number of Records"
*
ERROR at line 1:
ORA-00933: SQL command not properly ended

SQL> select count(*) as "Number of Records" from immigration;
Number of Records
-----
1797

SQL>
```

The record with state Maine and value 146 is also deleted from view3 and has 1797 records

Screenshots -

| STATE        | COUNTRY | UPPER_LEVEL_People | TOTAL_POPULATION | CROSSING_ | VALUE  | CONTAINER                   |
|--------------|---------|--------------------|------------------|-----------|--------|-----------------------------|
| Maine        | USA     | 152868             | 1341582          | 01-JAN-19 | 26     | Trains                      |
| Vermont      | USA     | 69184              | 623968           | 01-JAN-19 | 181    | Rail Containers Full        |
| Texas        | USA     | 4124652            | 28704338         | 01-JAN-19 | 4424   | Bus Passengers              |
| Arizona      | USA     | 1643981            | 7123898          | 01-JAN-19 | 26258  | Personal Vehicles           |
| New York     | USA     | 2771138            | 19862512         | 01-JAN-19 | 56089  | Personal Vehicle Passengers |
| Arizona      | USA     | 1643981            | 7123898          | 01-JAN-19 | 797    | Buses                       |
| North Dakota | USA     | 77865              | 755238           | 01-JAN-19 | 562    | Trucks                      |
| Michigan     | USA     | 1395098            | 9991177          | 01-JAN-19 | 478349 | Personal Vehicle Passengers |
| California   | USA     | 5217133            | 39776838         | 01-JAN-19 | 48673  | Personal Vehicles           |
| Montana      | USA     | 136208             | 1862338          | 01-JAN-19 | 138    | Truck Containers Full       |
| Maine        | USA     | 152868             | 1341582          | 01-JAN-19 | 40     | Rail Containers Full        |
| New York     | USA     | 2771138            | 19862512         | 01-JAN-19 | 28098  | Personal Vehicles           |
| Minnesota    | USA     | 528890             | 5628162          | 01-JAN-19 | 6067   | Personal Vehicles           |
| North Dakota | USA     | 77865              | 755238           | 01-JAN-19 | 724    | Personal Vehicles           |
| California   | USA     | 5217133            | 39776838         | 01-JAN-19 | 169661 | Personal Vehicle Passengers |
| California   | USA     | 5217133            | 39776838         | 01-JAN-19 | 8328   | Bus Passengers              |
| North Dakota | USA     | 77865              | 755238           | 01-JAN-19 | 7      | Truck Containers Empty      |
| Michigan     | USA     | 1395098            | 9991177          | 01-JAN-19 | 188    | Trains                      |
| Arizona      | USA     | 1643981            | 7123898          | 01-JAN-19 | 34     | Passes                      |
| New York     | USA     | 2771138            | 19862512         | 01-JAN-19 | 1081   | Trucks                      |
| Vermont      | USA     | 69184              | 623968           | 01-JAN-19 | 235    | Rail Containers Empty       |
| Vermont      | USA     | 69184              | 623968           | 01-JAN-19 | 25796  | Personal Vehicles           |
| Montana      | USA     | 136208             | 1862338          | 01-JAN-19 | 1187   | Personal Vehicle Passengers |
| Vermont      | USA     | 69184              | 623968           | 01-JAN-19 | 2828   | Personal Vehicles           |
| New York     | USA     | 2771138            | 19862512         | 01-JAN-19 | 73144  | Trucks                      |
| Maine        | USA     | 152868             | 1341582          | 01-JAN-19 | 15     | Trucks                      |
| California   | USA     | 5217133            | 39776838         | 01-JAN-19 | 95295  | Personal Vehicle Passengers |
| North Dakota | USA     | 77865              | 755238           | 01-JAN-19 | 16593  | Truck Containers Full       |
| Idaho        | USA     | 228243             | 1753868          | 01-JAN-19 | 633    | Truck Containers Empty      |
| Michigan     | USA     | 1395098            | 9991177          | 01-JAN-19 | 1881   | Rail Containers Empty       |
| Texas        | USA     | 4124652            | 28704338         | 01-JAN-19 | 841    | Trucks                      |
| Washington   | USA     | 817547             | 7538552          | 01-JAN-19 | 1156   | Personal Vehicle Passengers |
| Texas        | USA     | 4124652            | 28704338         | 01-JAN-19 | 12834  | Truck Containers Full       |
| Maine        | USA     | 152868             | 1341582          | 01-JAN-19 | 1355   | Personal Vehicles           |
| Montana      | USA     | 136208             | 1862338          | 01-JAN-19 | 18     | Truck Containers Empty      |
| Washington   | USA     | 817547             | 7538552          | 01-JAN-19 | 9614   | Rail Containers Full        |
| Vermont      | USA     | 69184              | 623968           | 01-JAN-19 | 30     | Truck Containers Full       |
| California   | USA     | 5217133            | 39776838         | 01-JAN-19 | 90098  | Personal Vehicles           |
| New York     | USA     | 2771138            | 19862512         | 01-JAN-19 | 1529   | Truck Containers Empty      |
| Maine        | USA     | 152868             | 1341582          | 01-JAN-19 | 10     | Trains                      |
| Montana      | USA     | 136208             | 1862338          | 01-JAN-19 | 797    | Trucks                      |
| Montana      | USA     | 136208             | 1862338          | 01-JAN-19 | 385    | Personal Vehicle Passengers |
| Virginia     | USA     | 1643981            | 7123898          | 01-JAN-19 | 1265   | Rail Containers Full        |
| Washington   | USA     | 817547             | 7538552          | 01-JAN-19 | 865    | Trucks                      |
| Minnesota    | USA     | 528890             | 5628162          | 01-JAN-19 | 84     | Truck Containers Empty      |
| Washington   | USA     | 817547             | 7538552          | 01-JAN-19 | 674    | Truck Containers Empty      |
| Washington   | USA     | 817547             | 7538552          | 01-JAN-19 | 4      | Truck Containers Empty      |
| Montana      | USA     | 136208             | 1862338          | 01-JAN-19 | 212    | Bus Passengers              |
| Washington   | USA     | 817547             | 7538552          | 01-JAN-19 | 12137  | Trucks                      |
| North Dakota | USA     | 77865              | 755238           | 01-JAN-19 | 2160   | Personal Vehicles           |
| North Dakota | USA     | 77865              | 755238           | 01-JAN-19 | 84     | Personal Vehicles           |

1797 rows selected.

SQL> select count(\*) as "Number of Records" from view3;

Number of Records

1797

SQL> █

## Part 8: Pig on Hadoop

This is previously done on Dataset, Poverty.csv and State\_Population.csv

### Uploading Poverty.csv and State.csv

hadoop fs -put Poverty.csv P.txt

hadoop fs -put State.csv S.txt

Pig

Grunt>ls

```
[Viraj's-air:370 virajsonvane]$ sudo scp -i ~/Downloads/370/Vikey.pem ~/Downloads/370/Poverty.csv centos@ec2-54-84-183-153.compute-1.amazonaws.com:~/  
[Password: ]  
Poverty.csv  
[Viraj's-air:370 virajsonvane]$ sudo scp -i ~/Downloads/370/Vikey.pem ~/Downloads/370/State.csv centos@ec2-54-84-183-153.compute-1.amazonaws.com:~/  
State.csv  
[Viraj's-air:370 virajsonvane]$ sudo ssh -i "Vikey.pem" centos@ec2-54-84-183-153.compute-1.amazonaws.com  
Last login: Fri Apr 10 19:15:00 2020 from c-24-5-59-63.hsd1.ca.comcast.net  
[centos@ip-172-31-31-189: ~]$ pig  
1987.csv  dataf.txt mydata.txt  pig_158642034363.log  pig_1586498803344.log  pig_1586499854932.log  pig_1586549828165.log  pig_1586557202621.log  totalmiles.pig  
Count.txt  data2.txt  pig_1586400888306.log  pig_1586446694846.log  pig_15864993056794.log  pig_1586552082795.log  pig_1586568785839.log  
d1.txt  dept.txt  pig_1586421492518.log  pig_1586467425081.log  pig_1586499311047.log  pig_158650095359.log  pig_158655789917.log  Poverty.csv  
d2.txt  exp.txt  pig_1586421639428.log  pig_15864676093208.log  pig_1586499301504.log  pig_158656943954.log  pig_1586557162832.log  State.csv  
[centos@ip-172-31-31-189: ~]$ hadoop fs -put Poverty.csv P.txt  
[centos@ip-172-31-31-189: ~]$ hadoop fs -put State.csv S.txt  
[centos@ip-172-31-31-189: ~]$ pig  
2020-04-10 19:32:33.975 [main] INFO org.apache.pig.Main - Logging error messages to: /home/centos/pig_1586561553372.log  
2020-04-10 19:32:33.651 [main] INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: hdfs://localhost:8020  
2020-04-10 19:32:34.218 [main] INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to map-reduce job tracker at: localhost:8021  
grunt>ls
```

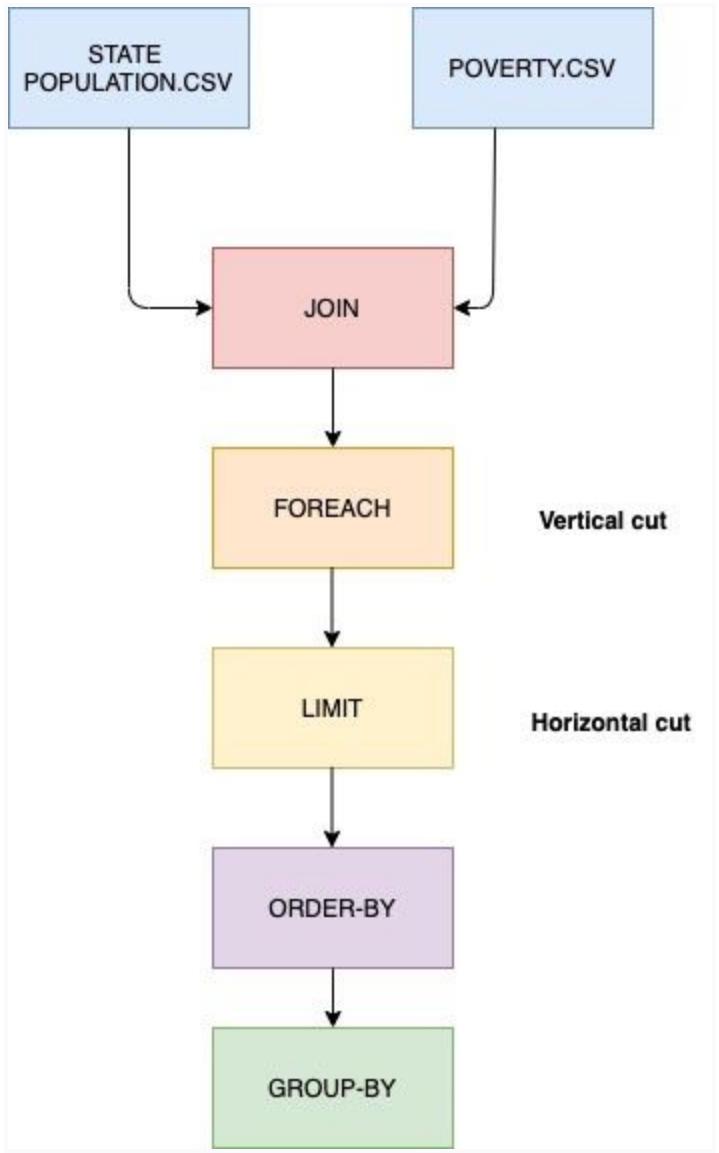
### Overview of Poverty table

```
grunt> cat P.txt  
Alabama,784517,820009,253694,272124  
Alaska,75273,83583,23957,28201  
Arizona,1002773,1043901,324723,347577  
Arkansas,463966,487800,149187,162569  
California,5111205,5217133,1586526,1645300  
Colorado,552126,580110,143486,160312  
Connecticut,326360,350942,89324,101814  
Delaware,116181,127003,32862,37720  
District of Columbia,104105,114899,29285,35421  
Florida,2860804,2942948,829310,872538  
Georgia,1503877,1568261,513320,550706  
Hawaii,126481,137921,32442,38702  
Idaho,203631,220243,60371,69421  
Illinois,1543257,1594083,470646,500986  
Indiana,841903,879603,263205,285029  
Iowa,319716,337542,85662,94830  
Kansas,325420,345322,97150,108566  
Kentucky,723378,753748,211037,226431  
Louisiana,875670,911984,293085,312963  
Maine,141958,152868,32560,37530  
Maryland,541127,571971,157132,173598  
Massachusetts,675076,710810,173590,191332  
Michigan,1351618,1395098,404866,427744  
Minnesota,505806,528890,142348,156088  
Mississippi,561749,585685,187869,200803  
Missouri,776721,809281,240522,259136  
Montana,125024,136208,32971,37675  
National,42342619,42824683,13229339,13477065  
Nebraska,191439,206529,59898,68064  
Nevada,379792,405922,122765,135439  
New_Hampshire,95275,105825,22708,27858  
New_Jersey,862408,901856,259622,281804  
New_Mexico,378032,399548,118269,129821  
New_York,2700018,2771138,793426,831246  
North_Carolina,1442873,1493191,461659,491217  
North_Dakota,70249,77865,17896,21426  
Ohio,1551281,1599521,493056,521182  
Oklahoma,589591,612231,193877,207017  
Oregon,521125,548567,133798,148368  
Pennsylvania,1524709,1572405,430076,457254  
Rhode_Island,118956,131460,34517,39665  
South_Carolina,735356,765362,234515,251125  
South_Dakota,103234,111810,31772,36182  
Tennessee,961862,998926,303257,323607  
Texas,4036102,4124652,1494968,1556920  
Utah,284801,307475,90671,103091  
Vermont,61032,69104,13200,16112  
Virginia,856763,896113,247583,268493  
Washington,781803,817547,219520,240494  
West_Virginia,316875,336129,83372,91830  
Wisconsin,627066,653282,175854,191036  
Wyoming,57678,64960,15694,18938
```

## Overview of State table

```
[grunt] > cat S.txt
USA,California,39776830
USA,Texas,28704330
USA,Florida,21312211
USA,New_York,19862512
USA,Pennsylvania,12823989
USA,Illinois,12768320
USA,Ohio,11694664
USA,Georgia,10545138
USA,North_Carolina,10390149
USA,Michigan,9991177
USA,New_Jersey,9032872
USA,Virginia,8525660
USA,Washington,7530552
USA,Arizona,7123898
USA,Massachusetts,6895917
USA,Tennessee,6782564
USA,Indiana,6699629
USA,Missouri,6135888
USA,Maryland,6079602
USA,Wisconsin,5818049
USA,Colorado,5684203
USA,Minnesota,5628162
USA,South_Carolina,5088916
USA,Alabama,4888949
USA,Louisiana,4682509
USA,Kentucky,4472265
USA,Oregon,4199563
USA,Oklahoma,3940521
USA,Connecticut,3588683
USA,Iowa,3160553
USA,Utah,3159345
USA,Nevada,3056824
USA,Arkansas,3020327
USA,Mississippi,2982785
USA,Kansas,2918515
USA,New_Mexico,2090708
USA,Nebraska,1932549
USA,West_Virginia,1803077
USA,Idaho,1753860
USA,Hawaii,1426393
USA,New_Hampshire,1350575
USA,Maine,1341582
USA,Montana,1062330
USA,Rhode_Island,1061712
USA,Delaware,971180
USA,South_Dakota,877790
USA,North_Dakota,755238
USA,Alaska,738068
USA,District_of_Columbia,703608
USA,Vermont,623960
USA,Wyoming,573720
```

## Flow Diagram:



## Loading Poverty table in to A;

A = Load 'P.txt' using pigstorage(',') as (State:Chararray,LowerPeople:int, Upperpeople:int, LowerChildren:int, UpperChildren:int)

Dump A:

```
grunt> A = LOAD 'P.txt' USING PigStorage(',') AS (State:chararray,LowerPeople:int,UpperPeople:int,LowerChildren:int,UpperChildren:int);
grunt> DUMP A;
2020-01-10 22:27:23 [INFO] TNEC: Dumping data from A to console
(Alabama,784517,820009,253694,272124)
(Alaska,75273,83583,23957,28201)
(Arizona,1002773,1043901,324723,347577)
(Arkansas,463966,487800,149187,162569)
(California,5111205,5217133,1586526,1645300)
(Colorado,552126,580110,143486,160312)
(Connecticut,326360,350942,89324,101814)
(Delaware,116181,127003,32862,37720)
(District of Columbia,104105,114899,29285,35421)
(Florida,2860804,2942948,829310,872538)
(Georgia,1503877,1568261,513320,550706)
(Hawaii,126481,137921,32442,38702)
(Idaho,203631,220243,60371,69421)
(Illinois,1543257,1594083,470646,500986)
(Indiana,841903,879603,263205,285029)
(Iowa,319716,337542,85662,94830)
(Kansas,325420,345322,97150,108566)
(Kentucky,723378,753748,211037,226431)
(Louisiana,875670,911984,293085,312963)
(Maine,141958,152868,32560,37530)
(Maryland,541127,571971,157132,173598)
(Massachusetts,675076,710810,173590,191332)
(Michigan,1351618,1395098,404866,427744)
(Minnesota,505806,528890,142348,156088)
(Mississippi,561749,585685,187869,200803)
(Missouri,776721,809281,240522,259136)
(Montana,125024,136208,32971,37675)
(National,42342619,42824683,13229339,13477065)
(Nebraska,191439,206529,59898,68064)
(Nevada,379792,405922,122765,135439)
(New_Hampshire,95275,105825,22708,27858)
(New_Jersey,862408,901856,259622,281804)
(New_Mexico,378032,399548,118269,129821)
(New_York,2700018,2771138,793426,831246)
(North_Carolina,1442873,1493191,461659,491217)
(North_Dakota,70249,77865,17896,21426)
(Ohio,1551281,1599521,493056,521182)
(Oklahoma,589591,612231,193877,207017)
(Oregon,521125,548567,133798,148368)
(Pennsylvania,1524709,1572405,430076,457254)
(Rhode_Island,118956,131460,34517,39665)
(South_Carolina,735356,765362,234515,251123)
(South_Dakota,103234,111810,31772,36182)
(Tennessee,961862,998926,303257,323607)
(Texas,4036102,4124652,1494968,1556920)
(Utah,284801,307475,90671,103091)
(Vermont,61032,69104,13200,16112)
(Virginia,856763,896113,247583,268493)
(Washington,781803,817547,219520,240494)
(West_Virginia,316875,336129,83372,91830)
(Wisconsin,627066,653282,175854,191036)
(Wyoming,57678,64960,15694,18938)
```

## Loading State table in to B:

B = Load 'S.txt' using pigstorage('') as (Country:Chararray,State:Chararray,Population:int)

Dump B:

```
[grunt> B = LOAD 'S.txt' USING PigStorage('') AS (Country:chararray,State:chararray,Population:int);
[grunt> DUMP B;
```

```
(USA,California,39776830)
(USA,Texas,28704330)
(USA,Florida,21312211)
(USA,New_York,19862512)
(USA,Pennsylvania,12823989)
(USA,Illinois,12768320)
(USA,Ohio,11694664)
(USA,Georgia,10545138)
(USA,North_Carolina,10390149)
(USA,Michigan,9991177)
(USA,New_Jersey,9032872)
(USA,Virginia,8525660)
(USA,Washington,7530552)
(USA,Arizona,7123898)
(USA,Massachusetts,6895917)
(USA,Tennessee,6782564)
(USA,Indiana,6699629)
(USA,Missouri,6135888)
(USA,Maryland,6079602)
(USA,Wisconsin,5818049)
(USA,Colorado,5684203)
(USA,Minnesota,5628162)
(USA,South_Carolina,5088916)
(USA,Alabama,4888949)
(USA,Louisiana,4682509)
(USA,Kentucky,4472265)
(USA,Oregon,4199563)
(USA,Oklahoma,3940521)
(USA,Connecticut,3588683)
(USA,Iowa,3160553)
(USA,Utah,3159345)
(USA,Nevada,3056824)
(USA,Arkansas,3020327)
(USA,Mississippi,2982785)
(USA,Kansas,2918515)
(USA,New_Mexico,2090708)
(USA,Nebraska,1932549)
(USA,West_Virginia,1803077)
(USA,Idaho,1753860)
(USA,Hawaii,1426393)
(USA,New_Hampshire,1350575)
(USA,Maine,1341582)
(USA,Montana,1062330)
(USA,Rhode_Island,1061712)
(USA,Delaware,971180)
(USA,South_Dakota,877790)
(USA,North_Dakota,755238)
(USA,Alaska,738068)
(USA,District_of_Columbia,703608)
(USA,Vermont,623960)
(USA,Wyoming,573720)
```

## Preview of Poverty Table

Illustrate A:

| A | State:chararray | LowerPeople:int | UpperPeople:int | LowerChildren:int | UpperChildren:int |
|---|-----------------|-----------------|-----------------|-------------------|-------------------|
|   | Washington      | 781803          | 817547          | 219520            | 240494            |

## Preview of State Table

Illustrate B:

| B | Country:chararray | State:chararray | Population:int |
|---|-------------------|-----------------|----------------|
|   | USA               | Arkansas        | 3020327        |

## Applying Left Outer Join on Both the table using State

J = JOIN A by State left outer, B by state;

```
grunt> J = JOIN A BY State LEFT OUTER, B BY State;
grunt> Describe J;
J: (A::State: chararray,A::LowerPeople: int,A::UpperPeople: int,A::LowerChildren: int,A::UpperChildren: int,B::Country: chararray,B::State: chararray,B::Population: int)
grunt> DUMP J;
```

```
(Iowa,319716,337542,85662,94830,USA,Iowa,3160553)
(Ohio,1551281,1599521,493056,521182,USA,Ohio,11694664)
(Utah,284801,307475,90671,103091,USA,Utah,3159345)
(Idaho,203631,220243,60371,69421,USA,Idaho,1753860)
(Maine,141958,152868,32560,37530,USA,Maine,1341582)
(Texas,4036102,4124652,1494968,1556920,USA,Texas,28704330)
(Alaska,75273,83583,23957,28201,USA,Alaska,738068)
(Hawaii,126481,137921,32442,38702,USA,Hawaii,1426393)
(Kansas,325420,345322,97150,108566,USA,Kansas,2918515)
(Nevada,379792,405922,122765,135439,USA,Nevada,3056824)
(Oregon,521125,548567,133798,148368,USA,Oregon,4199563)
(Alabama,784517,820009,253694,272124,USA,Alabama,4888949)
(Arizona,1002773,1043901,324723,347577,USA,Arizona,7123898)
(Florida,2860804,2942948,829310,872538,USA,Florida,21312211)
(Georgia,1503877,1568261,513320,550706,USA,Georgia,10545138)
(Indiana,841903,879603,263205,285029,USA,Indiana,6699629)
(Montana,125024,136208,32971,37675,USA,Montana,1062330)
(Vermont,61032,69104,13200,16112,USA,Vermont,623960)
(Wyoming,57678,64960,15694,18938,USA,Wyoming,573720)
(Arkansas,463966,487800,149187,162569,USA,Arkansas,3020327)
(Colorado,552126,580110,143486,160312,USA,Colorado,5684203)
(Delaware,116181,127003,32862,37720,USA,Delaware,971180)
(Illinois,1543257,1594083,470646,500986,USA,Illinois,12768320)
(Kentucky,723378,753748,211037,226431,USA,Kentucky,4472265)
(Maryland,541127,571971,157132,173598,USA,Maryland,6079602)
(Michigan,1351618,1395098,404866,427744,USA,Michigan,9991177)
(Missouri,776721,809281,240522,259136,USA,Missouri,6135888)
(National,42342619,42824683,13229339,13477065,...)
(Nebraska,191439,206529,59898,68064,USA,Nebraska,1932549)
(New_York,2700018,2771138,793426,831246,USA>New_York,19862512)
(Oklahoma,589591,612231,193877,207017,USA,Oklahoma,3946521)
(Virginia,856763,896113,247583,268493,USA,Virginia,8525660)
(Louisiana,875670,911984,293085,312963,USA,Louisiana,4682509)
(Minnesota,505806,528890,142348,156088,USA,Minnesota,5628162)
(Tennessee,961862,998926,303257,323607,USA,Tennessee,6782564)
(Wisconsin,627066,653282,175854,191936,USA,Wisconsin,5818049)
(California,5111205,5217133,1586526,1645300,USA,California,39776830)
(New_Jersey,862408,901856,259622,281804,USA,New_Jersey,9032872)
(New_Mexico,378032,399548,118269,129821,USA,New_Mexico,2090708)
(Washington,781803,817547,219520,240494,USA,Washington,7530552)
(Connecticut,326360,350942,89324,101814,USA,Connecticut,3588683)
(Mississippi,561749,585685,187869,200803,USA,Mississippi,2982785)
(North_Dakota,70249,77865,17896,21426,USA,North_Dakota,755238)
(Pennsylvania,1524709,1572405,430076,457254,USA,Pennsylvania,12823989)
(Rhode_Island,118956,131460,34517,39665,USA,Rhode_Island,1061712)
(South_Dakota,103234,111810,31772,36182,USA,South_Dakota,877790)
(Massachusetts,675076,710810,173590,191332,USA,Massachusetts,6895917)
(New_Hampshire,95275,105825,22708,27858,USA,New_Hampshire,1350575)
(West_Virginia,316875,336129,83372,91830,USA,West_Virginia,1803077)
(North_Carolina,1442873,1493191,461659,491217,USA,North_Carolina,10390149)
(South_Carolina,735356,765362,234515,251123,USA,South_Carolina,5088916)
(District_of_Columbia,104105,114899,29285,35421,...)
```

## Applying Foreach on Join using State, Population, LowerChildren.

G = Foreach J generate State,Population,LowerChildren;

Describe G:

Dump G:

```
[grunt> G = FOREACH J GENERATE B::State,B::Population,A::LowerChildren;
[grunt> Describe G;
G: (B::State: chararray,B::Population: int,A::LowerChildren: int)
[grunt> DUMP G;
```

```
(Iowa,3160553,85662)
(Ohio,11694664,493056)
(Utah,3159345,90671)
(Idaho,1753860,60371)
(Maine,1341582,32560)
(Texas,28704330,1494968)
(Alaska,738068,23957)
(Hawaii,1426393,32442)
(Kansas,2918515,97150)
(Nevada,3056824,122765)
(Oregon,4199563,133798)
(Alabama,4888949,253694)
(Arizona,7123898,324723)
(Florida,21312211,829310)
(Georgia,10545138,513320)
(Indiana,6699629,263205)
(Montana,1062330,32971)
(Vermont,623960,13200)
(Wyoming,573720,15694)
(Arkansas,3020327,149187)
(Colorado,5684203,143486)
(Delaware,971180,32862)
(Illinois,12768320,470646)
(Kentucky,4472265,211037)
(Maryland,6079602,157132)
(Michigan,9991177,404866)
(Missouri,6135888,240522)
(.,13229339)
(Nebraska,1932549,59898)
(New_York,19862512,793426)
(Oklahoma,3940521,193877)
(Virginia,8525660,247583)
(Louisiana,4682509,293085)
(Minnesota,5628162,142348)
(Tennessee,6782564,303257)
(Wisconsin,5818049,175854)
(California,39776830,1586526)
(New_Jersey,9032872,259622)
(New_Mexico,2090708,118269)
(Washington,7530552,219520)
(Connecticut,3588683,89324)
(Mississippi,2982785,187869)
(North_Dakota,755238,17896)
(Pennsylvania,12823989,430076)
(Rhode_Island,1061712,34517)
(South_Dakota,877790,31772)
(Massachusetts,6895917,173590)
(New_Hampshire,1350575,22708)
(West_Virginia,1803077,83372)
(North_Carolina,10390149,461659)
(South_Carolina,5088916,234515)
(.,29285)
```

## Applying Limit on G of 50

L = Limit G 50:

DUMP L:

```
[grunt> L = Limit G 50;
[grunt> DUMP L;
```

```
(Delaware,971180,32862)
(Montana,1062330,32971)
(Rhode_Island,1061712,34517)
(Nebraska,1932549,59898)
(Idaho,1753860,60371)
(West_Virginia,1803077,83372)
(Iowa,3160553,85662)
(Connecticut,3588683,89324)
(Utah,3159345,90671)
(Kansas,2918515,97150)
(New_Mexico,2090708,118269)
(Nevada,3056824,122765)
(Oregon,4199563,133798)
(Minnesota,5628162,142348)
(Colorado,5684203,143486)
(Arkansas,3020327,149187)
(Maryland,6079602,157132)
(Massachusetts,6895917,173590)
(Wisconsin,5818049,175854)
(Mississippi,2982785,187869)
(Oklahoma,3940521,193877)
(Kentucky,4472265,211037)
(Washington,7530552,219520)
(Missouri,6135888,240522)
(Virginia,8525660,247583)
(Alabama,4888949,253694)
(New_Jersey,9032872,259622)
(Indiana,6699629,263205)
(Louisiana,4682509,293085)
(Tennessee,6782564,303257)
(Arizona,7123898,324723)
(Michigan,9991177,404866)
(Pennsylvania,12823989,430076)
(North_Carolina,10390149,461659)
(Illinois,12768320,470646)
(Ohio,11694664,493056)
(Georgia,10545138,513320)
(New_York,19862512,793426)
(Florida,21312211,829310)
(Texas,28704330,1494968)
(California,39776830,1586526)
(.,13229339)
(Vermont,623960,13200)
(Wyoming,573720,15694)
(North_Dakota,755238,17896)
(New_Hampshire,1350575,22708)
(Alaska,738068,23957)
(South_Dakota,877790,31772)
(Hawaii,1426393,32442)
(Maine,1341582,32560)
```

## Applying Order by on Obtained Limited L

O = Order L by State ASC;

Describe O:

Dump O:

```
[grunt> O = ORDER L By State ASC;
[grunt> O = ORDER L By State ASC;
[grunt> describe O;
O: {B::State: chararray,B::Population: int,A::LowerChildren: int}
[grunt> Dump O;
```

```
(,,13229339)
(Alabama,4888949,253694)
(Alaska,738068,23957)
(Arizona,7123898,324723)
(Arkansas,3020327,149187)
(California,39776830,1586526)
(Colorado,5684203,143486)
(Connecticut,3588683,89324)
(Delaware,971180,32862)
(Florida,21312211,829310)
(Georgia,10545138,513320)
(Hawaii,1426393,32442)
(Idaho,1753860,60371)
(Illinois,12768320,470646)
(Indiana,6699629,263205)
(Iowa,3160553,85662)
(Kansas,2918515,97150)
(Kentucky,4472265,211037)
(Louisiana,4682509,293085)
(Maine,1341582,32560)
(Maryland,6079602,157132)
(Massachusetts,6895917,173590)
(Michigan,9991177,404866)
(Minnesota,5628162,142348)
(Mississippi,2982785,187869)
(Missouri,6135888,240522)
(Montana,1062330,32971)
(Nebraska,1932549,59898)
(Nevada,3056824,122765)
(New_Hampshire,1350575,22708)
(New_Jersey,9032872,259622)
(New_Mexico,2090708,118269)
(New_York,19862512,793426)
(North_Carolina,10390149,461659)
(North_Dakota,755238,17896)
(Ohio,11694664,493056)
(Oklahoma,3940521,193877)
(Oregon,4199563,133798)
(Pennsylvania,12823989,430076)
(Rhode_Island,1061712,34517)
(South_Dakota,877790,31772)
(Tennessee,6782564,303257)
(Texas,28704330,1494968)
(Utah,3159345,90671)
(Vermont,623960,13200)
(Virginia,8525660,247583)
(Washington,7530552,219520)
(West_Virginia,1803077,83372)
(Wisconsin,5818049,175854)
(Wyoming,573720,15694)
```

## Applying Group By State

```
[grunt> Gm = GROUP 0 BY B::State;
[grunt> DUMP Gm;
```

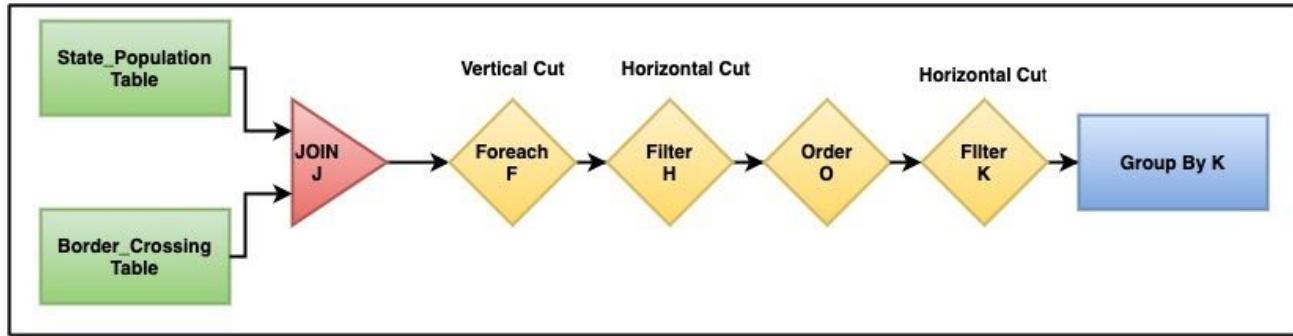
```
(Iowa,((Iowa,3160553,85662)))
(Ohio,((Ohio,11694664,493056)))
(Utah,((Utah,3159345,90671)))
(Idaho,((Idaho,1753860,60371)))
(Maine,((Maine,1341582,32560)))
(Texas,((Texas,28704330,1494968)))
(Alaska,((Alaska,738068,23957)))
(Hawaii,((Hawaii,1426393,32442)))
(Kansas,((Kansas,2918515,97150)))
(Nevada,((Nevada,3056824,122765)))
(Oregon,((Oregon,4199563,133798)))
(Alabama,((Alabama,4888949,253694)))
(Arizona,((Arizona,7123898,324723)))
(Florida,((Florida,21312211,829310)))
(Georgia,((Georgia,10545138,513320)))
(Indiana,((Indiana,6699629,263205)))
(Montana,((Montana,1062330,32971)))
(Vermont,((Vermont,623960,13200)))
(Wyoming,((Wyoming,573720,15694)))
(Arkansas,((Arkansas,3020327,149187)))
(Colorado,((Colorado,5684203,143486)))
(Delaware,((Delaware,971180,32862)))
(Illinois,((Illinois,12768320,470646)))
(Kentucky,((Kentucky,4472265,211037)))
(Maryland,((Maryland,6079602,157132)))
(Michigan,((Michigan,9991177,404866)))
(Missouri,((Missouri,6135888,240522)))
(Nebraska,((Nebraska,1932549,59898)))
(New_York,((New_York,19862512,793426)))
(Oklahoma,((Oklahoma,3940521,193877)))
(Virginia,((Virginia,8525660,247583)))
(Louisiana,((Louisiana,4682509,293085)))
(Minnesota,((Minnesota,5628162,142348)))
(Tennessee,((Tennessee,6782564,303257)))
(Wisconsin,((Wisconsin,5818049,175854)))
(California,((California,39776830,1586526)))
(New_Jersey,((New_Jersey,9032872,259622)))
(New_Mexico,((New_Mexico,2090708,118269)))
(Washington,((Washington,7530552,219520)))
(Connecticut,((Connecticut,3588683,89324)))
(Mississippi,((Mississippi,2982785,187869)))
(North_Dakota,((North_Dakota,755238,17896)))
(Pennsylvania,((Pennsylvania,12823989,430076)))
(Rhode_Island,((Rhode_Island,1061712,34517)))
(South_Dakota,((South_Dakota,877790,31772)))
(Massachusetts,((Massachusetts,6895917,173590)))
(New_Hampshire,((New_Hampshire,1350575,22708)))
(West_Virginia,((West_Virginia,1803077,83372)))
(North_Carolina,((North_Carolina,10390149,461659)))
```

## Applying Group By LowerChildren

```
[grunt> Dg = GROUP 0 BY LowerChildren;
[grunt> describe Dg;
Dg: {group: int,0: {(B::State: chararray,B::Population: int,A::LowerChildren: int))}}
[grunt> DUMP Dg;
```

```
(13200, { (Vermont,623960,13200)})
(15694, { (Wyoming,573720,15694)})
(17896, { (North_Dakota,755238,17896)})
(22708, { (New_Hampshire,1350575,22708)})
(23957, { (Alaska,738068,23957)})
(31772, { (South_Dakota,877790,31772)})
(32442, { (Hawaii,1426393,32442)})
(32560, { (Maine,1341582,32560)})
(32862, { (Delaware,971180,32862)})
(32971, { (Montana,1062330,32971)})
(34517, { (Rhode_Island,1061712,34517)})
(59898, { (Nebraska,1932549,59898)})
(60371, { (Idaho,1753860,60371)})
(83372, { (West_Virginia,1803077,83372)})
(85662, { (Iowa,3160553,85662)})
(89324, { (Connecticut,3588683,89324)})
(90671, { (Utah,3159345,90671)})
(97150, { (Kansas,2918515,97150)})
(118269, { (New_Mexico,2090708,118269)})
(122765, { (Nevada,3056824,122765)})
(133798, { (Oregon,4199563,133798)})
(142348, { (Minnesota,5628162,142348)})
(143486, { (Colorado,5684203,143486)})
(149187, { (Arkansas,3020327,149187)})
(157132, { (Maryland,6079602,157132)})
(173590, { (Massachusetts,6895917,173590)})
(175854, { (Wisconsin,5818049,175854)})
(187869, { (Mississippi,2982785,187869)})
(193877, { (Oklahoma,3940521,193877)})
(211037, { (Kentucky,4472265,211037)})
(219520, { (Washington,7530552,219520)})
(240522, { (Missouri,6135888,240522)})
(247583, { (Virginia,8525660,247583)})
(253694, { (Alabama,4888949,253694)})
(259622, { (New_Jersey,9032872,259622)})
(263205, { (Indiana,6699629,263205)})
(293085, { (Louisiana,4682509,293085)})
(303257, { (Tennessee,6782564,303257)})
(324723, { (Arizona,7123898,324723)})
(404866, { (Michigan,9991177,404866)})
(430076, { (Pennsylvania,12823989,430076)})
(461659, { (North_Carolina,10390149,461659)})
(470646, { (Illinois,12768320,470646)})
(493056, { (Ohio,11694664,493056)})
(513320, { (Georgia,10545138,513320)})
(793426, { (New_York,19862512,793426)})
(829310, { (Florida,21312211,829310)})
(1494968, { (Texas,28704330,1494968)})
(1586526, { (California,39776830,1586526)})
(13229339, { (,13229339)})
```

## Flow Diagram:



Now adding, new Dataset BorderCrossing Cross.csv,

### Uploading Border\_cross.csv and State.csv

`hadoop fs -put Cross.csv Cross.txt`

```
[centos@ip-172-31-86-109 ~]$ hadoop fs -put Cross.csv Cross.txt
```

### Overview of table Cross.txt

```
[centos@ip-172-31-86-109 ~]$ cat Cross.csv
Port,Name,State,Port Code,Border,Date,Container,Nos People,Location
Calexico,East,California,2507,US-Mexico Border,3/2/19 0:00,Trucks,34447,POINT (-115.48433000000001 32.67524)
Van Buren,Maine,108,US-Canada Border,3/1/19 0:00,Rail Containers Full,428,POINT (-67.94271 47.16207)
Otay Mesa,California,2506,US-Mexico Border,3/1/19 0:00,Trucks,81217,POINT (-117.05333 32.57333)
Nogales,Arizona,2604,US-Mexico Border,3/1/19 0:00,Trains,62,POINT (-110.93361 31.340279999999996)
Trout River,New York,715,US-Canada Border,3/2/19 0:00,Personal Vehicle Passengers,16377,POINT (-73.44253 44.990010000000005)
Madawaska,Maine,109,US-Canada Border,3/1/19 0:00,Trucks,179,POINT (-68.3271 47.35446)
Pembina,North Dakota,3401,US-Canada Border,3/1/19 0:00,Bus Passengers,1054,POINT (-97.24333 48.96639)
Progreso,Texas,2309,US-Mexico Border,3/1/19 0:00,Truck Containers Empty,1808,POINT (-97.94889 26.061670000000003)
Portal,North Dakota,3403,US-Canada Border,3/2/19 0:00,Rail Containers Empty,6685,POINT (-102.54917 48.99583)
Champlain-Rouses Point,New York,712,US-Canada Border,3/1/19 0:00,Trucks,24759,POINT (-73.44694 44.98639)
Opheim,Montana,3317,US-Canada Border,3/1/19 0:00,Personal Vehicles,235,POINT (-106.40265 48.85574)
Neché,North Dakota,3404,US-Canada Border,3/1/19 0:00,Trucks,584,POINT (-97.55444 48.98611)
Lancaster,Minnesota,3430,US-Canada Border,3/1/19 0:00,Personal Vehicles,1612,POINT (-96.82024 48.84911000000001)
Derby Line,Vermont,209,US-Canada Border,3/1/19 0:00,Buses,80,POINT (-72.09944 45.005)
Sarles,North Dakota,3409,US-Canada Border,3/1/19 0:00,Personal Vehicle Passengers,509,POINT (-98.99457 48.941050000000004)
Wildhorse,Montana,3323,US-Canada Border,3/1/19 0:00,Truck Containers Full,122,POINT (-109.67761 48.54863)
Derby Line,Vermont,209,US-Canada Border,3/1/19 0:00,Truck Containers Full,6483,POINT (-72.09944 45.005)
Lynden,Washington,3023,US-Canada Border,3/1/19 0:00,Pedestrians,79,POINT (-122.44316000000002 48.94802)
Vanceboro,Maine,105,US-Canada Border,3/1/19 0:00,Pedestrians,3,POINT (-67.42955 45.55984)
San Ysidro,California,2504,US-Mexico Border,3/1/19 0:00,Bus Passengers,7779,POINT (-117.0266699999998 32.54306)
Scobey,Montana,3309,US-Canada Border,3/1/19 0:00,Personal Vehicles,325,POINT (-105.44272 48.80707)
Beecher Falls,Vermont,206,US-Canada Border,3/1/19 0:00,Trucks,981,POINT (-71.49664 45.00808)
Calais,Maine,115,US-Canada Border,3/1/19 0:00,Train Passengers,30,POINT (-67.27917 45.18889)
Massena,New York,704,US-Canada Border,3/1/19 0:00,Truck Containers Full,1489,POINT (-74.74 44.98944)
Oroville,Washington,3019,US-Canada Border,3/2/19 0:00,Buses,12,POINT (-119.43444 48.93917)
Hansboro,North Dakota,3415,US-Canada Border,3/1/19 0:00,Truck Containers Full,1,POINT (-99.38250000000001 48.95)
Ferry,Washington,3013,US-Canada Border,3/1/19 0:00,Personal Vehicles,601,POINT (-118.58774 48.88627)
Pembina,North Dakota,3401,US-Canada Border,3/1/19 0:00,Trucks,19150,POINT (-97.24333 48.96639)
Tecate,California,2505,US-Mexico Border,3/1/19 0:00,Truck Containers Empty,1993,POINT (-116.62667 32.57722)
Eastport,Idaho,3302,US-Canada Border,3/1/19 0:00,Trains,101,POINT (-116.1802799999998 48.99944)
Walhalla,North Dakota,3407,US-Canada Border,3/1/19 0:00,Personal Vehicles,2399,POINT (-97.91778 48.92333)
Calais,Maine,115,US-Canada Border,3/1/19 0:00,Trains,15,POINT (-67.27917 45.18889)
Roma,Texas,2310,US-Mexico Border,3/1/19 0:00,Truck Containers Empty,740,POINT (-99.01833 26.40361)
Naco,Arizona,2603,US-Mexico Border,3/1/19 0:00,Truck Containers Empty,139,POINT (-109.9475 31.335280000000004)
Boquillas,Texas,2410,US-Mexico Border,3/1/19 0:00,Pedestrians,4362,POINT (-102.95 29.190000000000005)
Raymond,Montana,3301,US-Canada Border,3/1/19 0:00,Buses,1,POINT (-104.575 48.99667)
Porthill,Idaho,3308,US-Canada Border,3/1/19 0:00,Truck Containers Empty,282,POINT (-116.49828 48.99282)
Norton,Vermont,211,US-Canada Border,3/1/19 0:00,Personal Vehicles,1350,POINT (-71.79528000000002 45.01)
Sault Sainte Marie,Michigan,3803,US-Canada Border,3/1/19 0:00,Truck Containers Empty,769,POINT (-84.29889 46.47028)
Antler,North Dakota,3413,US-Canada Border,3/1/19 0:00,Truck Containers Full,9,POINT (-101.28194 48.97083)
Otay Mesa,California,2506,US-Mexico Border,3/1/19 0:00,Personal Vehicles,573597,POINT (-117.05333 32.57333)
Eastport,Idaho,3302,US-Canada Border,3/1/19 0:00,Pedestrians,200,POINT (-116.1802799999998 48.99944)
Rio Grande City,Texas,2307,US-Mexico Border,3/1/19 0:00,Pedestrians,4762,POINT (-98.82 26.37944)
Portal,North Dakota,3403,US-Canada Border,3/1/19 0:00,Trains,135,POINT (-102.54917 48.99583)
Del Rio,Texas,2302,US-Mexico Border,3/1/19 0:00,Truck Containers Full,5295,POINT (-100.89639 29.3625)
Sasabe,Arizona,2606,US-Mexico Border,3/1/19 0:00,Personal Vehicle Passengers,4658,POINT (-111.54139 31.48861)
Sault Sainte Marie,Michigan,3803,US-Canada Border,3/1/19 0:00,Truck Containers Full,3362,POINT (-84.29889 46.47028)
Highgate Springs-Alburg,Vermont,212,US-Canada Border,3/1/19 0:00,Trains,25,POINT (-73.10583 44.979440000000004)
Madawaska,Maine,109,US-Canada Border,3/1/19 0:00,Personal Vehicle Passengers,44387,POINT (-68.3271 47.35446)
Sasabe,Arizona,2606,US-Mexico Border,3/1/19 0:00,Pedestrians,23,POINT (-111.54139 31.48861)
Del Bonita,Montana,3322,US-Canada Border,3/1/19 0:00,Truck Containers Full,39,POINT (-112.32481 48.63274)
Vanceboro,Maine,105,US-Canada Border,3/1/19 0:00,Personal Vehicles,1476,POINT (-67.42955 45.55984)
Neché,North Dakota,3404,US-Canada Border,3/1/19 0:00,Buses,2,POINT (-97.55444 48.98611)
Houlton,Maine,106,US-Canada Border,3/1/19 0:00,Truck Containers Full,5778,POINT (-67.84083 46.12611)
Norton,Vermont,211,US-Canada Border,3/1/19 0:00,Trains,19,POINT (-71.79528000000002 45.01)
```

## Continuing output

```
Detroit,Michigan,3801,US-Canada Border,7/1/17 0:00,Bus Passengers,15877,POINT (-83.10222 42.38306000000001)
Pembina,North Dakota,3401,US-Canada Border,7/1/17 0:00,Truck Containers Empty,2452,POINT (-97.24333 48.96639)
Columbus,New Mexico,2406,US-Mexico Border,7/1/17 0:00,Personal Vehicles,38635,POINT (-107.63944 31.827499999999997)
Jackman,Maine,104,US-Canada Border,7/1/17 0:00,Buses,11,POINT (-70.255 45.62389)
Fortuna,North Dakota,3417,US-Canada Border,7/1/17 0:00,Buses,1,POINT (-103.77986 48.91086)
Frontier,Washington,3020,US-Canada Border,7/1/17 0:00,Truck Containers Full,661,POINT (-117.78134000000001 48.910160000000005)
Hidalgo,Texas,2305,US-Mexico Border,7/1/17 0:00,Pedestrians,188792,POINT (-98.26278 26.1)
Norton,Vermont,211,US-Canada Border,7/1/17 0:00,Personal Vehicles,4624,POINT (-71.79528000000002 45.01)
Opheim,Montana,3317,US-Canada Border,7/1/17 0:00,Personal Vehicles,412,POINT (-106.48265 48.85574)
Beecher Falls,Vermont,206,US-Canada Border,7/1/17 0:00,Truck Containers Full,51,POINT (-71.49664 45.00808)
Sarles,North Dakota,3409,US-Canada Border,7/1/17 0:00,Buses,16,POINT (-98.99457 48.94105000000004)
Buffalo-Niagara Falls,New York,901,US-Canada Border,7/1/17 0:00,Rail Containers Empty,3214,POINT (-79.05694 43.09444)
Fort Kent,Maine,110,US-Canada Border,7/1/17 0:00,Trucks,718,POINT (-68.58458 47.26878)
Baudette,Minnesota,3424,US-Canada Border,7/1/17 0:00,Trucks,663,POINT (-94.72512 48.70731)
Portal,North Dakota,3403,US-Canada Border,7/1/17 0:00,Trucks,6618,POINT (-102.54917 48.99583)
Houlton,Maine,106,US-Canada Border,7/1/17 0:00,Personal Vehicles,27503,POINT (-67.84083 46.12611)
Laurier,Washington,3016,US-Canada Border,7/1/17 0:00,Trucks,544,POINT (-118.22302 48.99892)
El Paso,Texas,2402,US-Mexico Border,7/1/17 0:00,Rail Containers Empty,4587,POINT (-106.48639 31.75861000000004)
Norton,Vermont,211,US-Canada Border,7/1/17 0:00,Truck Containers Empty,277,POINT (-71.79528000000002 45.01)
Bridgewater,Maine,127,US-Canada Border,7/1/17 0:00,Trucks,604,POINT (-67.84262 46.41923)
Eastport,Idaho,3302,US-Canada Border,7/1/17 0:00,Rail Containers Empty,697,POINT (-116.1802799999998 48.99944)
Madawaska,Maine,109,US-Canada Border,7/1/17 0:00,Bus Passengers,141,POINT (-68.3271 47.35446)
Otay Mesa,California,2506,US-Mexico Border,7/1/17 0:00,Personal Vehicles,660099,POINT (-117.05333 32.57333)
Eastport,Idaho,3302,US-Canada Border,7/1/17 0:00,Personal Vehicles,12056,POINT (-116.1802799999998 48.99944)
Vanceboro,Maine,105,US-Canada Border,7/1/17 0:00,Personal Vehicles,2768,POINT (-67.42955 45.55984)
Roosville,Montana,3318,US-Canada Border,7/1/17 0:00,Personal Vehicles,31543,POINT (-115.04512000000001 48.88171)
Frontier,Washington,3020,US-Canada Border,7/1/17 0:00,Pedestrians,12,POINT (-117.78134000000001 48.91016000000005)
Raymond,Montana,3301,US-Canada Border,7/1/17 0:00,Personal Vehicle Passengers,6529,POINT (-104.575 48.99667)
Eastport,Idaho,3302,US-Canada Border,7/1/17 0:00,Trucks,5202,POINT (-116.1802799999998 48.99944)
Sarles,North Dakota,3409,US-Canada Border,7/1/17 0:00,Truck Containers Empty,35,POINT (-98.99457 48.94105000000004)
Laurier,Washington,3016,US-Canada Border,7/1/17 0:00,Trains,12,POINT (-118.22302 48.99892)
Eastport,Idaho,3302,US-Canada Border,7/1/17 0:00,Truck Containers Full,4077,POINT (-116.1802799999998 48.99944)
San Luis,Arizona,2608,US-Mexico Border,7/1/17 0:00,Bus Passengers,20,POINT (-114.78139000000002 32.48694)
Sweetgrass,Montana,3310,US-Canada Border,7/1/17 0:00,Rail Containers Empty,762,POINT (-111.95972 48.99611000000001)
Progreso,Texas,2309,US-Mexico Border,7/1/17 0:00,Personal Vehicle Passengers,248337,POINT (-97.94889 26.06167000000003)
Antler,North Dakota,3413,US-Canada Border,7/1/17 0:00,Personal Vehicles,742,POINT (-101.28194 48.97083)
Point Roberts,Washington,3017,US-Canada Border,7/1/17 0:00,Personal Vehicles,91745,POINT (-123.07994 48.96978)
Eagle Pass,Texas,2303,US-Mexico Border,7/1/17 0:00,Pedestrians,85166,POINT (-100.49917 28.70889)
Baudette,Minnesota,3424,US-Canada Border,7/1/17 0:00,Personal Vehicle Passengers,34090,POINT (-94.72512 48.70731)
Norton,Vermont,211,US-Canada Border,7/1/17 0:00,Train Passengers,32,POINT (-71.79528000000002 45.01)
Portal,North Dakota,3403,US-Canada Border,7/1/17 0:00,Personal Vehicle Passengers,24344,POINT (-102.54917 48.99583)
Sweetgrass,Montana,3310,US-Canada Border,7/1/17 0:00,Bus Passengers,1463,POINT (-111.95972 48.99611000000001)
Ketchikan,Alaska,3102,US-Canada Border,7/1/17 0:00,Personal Vehicles,399,POINT (-131.64031 55.33897)
Presidio,Texas,2403,US-Mexico Border,7/1/17 0:00,Personal Vehicle Passengers,138460,POINT (-104.37167 29.56056)
Boundary,Washington,3015,US-Canada Border,7/1/17 0:00,Pedestrians,4,POINT (-117.89655000000002 48.54648)
Nogales,Arizona,2604,US-Mexico Border,7/1/17 0:00,Truck Containers Full,12247,POINT (-110.93361 31.340279999999996)
Lancaster,Minnesota,3430,US-Canada Border,7/1/17 0:00,Personal Vehicles,3458,POINT (-96.82024 48.84911000000001)
Hidalgo,Texas,2305,US-Mexico Border,7/1/17 0:00,Personal Vehicle Passengers,836177,POINT (-98.26278 26.1)
Lynden,Washington,3023,US-Canada Border,7/1/17 0:00,Trucks,3124,POINT (-122.44316000000002 48.94802)
Ogdensburg,New York,701,US-Canada Border,7/1/17 0:00,Truck Containers Empty,708,POINT (-75.48167 44.70667)
Vanceboro,Maine,105,US-Canada Border,7/1/17 0:00,Rail Containers Empty,947,POINT (-67.42955 45.55984)
Sweetgrass,Montana,3310,US-Canada Border,7/1/17 0:00,Buses,45,POINT (-111.95972 48.99611000000001)
Highgate Springs-Alburg,Vermont,212,US-Canada Border,7/1/17 0:00,Bus Passengers,10298,POINT (-73.10583 44.979440000000004)
Ogdensburg,New York,701,US-Canada Border,7/1/17 0:00,Bus Passengers,204,POINT (-75.48167 44.70667)
Champlain-Rouses Point,New York,712,US-Canada Border,7/1/17 0:00,Rail Containers Empty,714,POINT (-73.44694 44.98639)
Fort Fairfield,Maine,107,US-Canada Border,7/1/17 0:00,Personal Vehicle Passengers,17043,POINT (-67.82965 46.76891)
Wildhorse,Montana,3323,US-Canada Border,7/1/17 0:00,Truck Containers Empty,1,POINT (-109.67761 48.54863)
Del Rio,Texas,2302,US-Mexico Border,7/1/17 0:00,Trucks,5834,POINT (-100.89639 29.3625)
Madawaska,Maine,109,US-Canada Border,7/1/17 0:00,Personal Vehicle Passengers,52188,POINT (-68.3271 47.35446)grunt> ■
```

End

Here we are loading State table in to B;

B = Load 'S.txt' using pigstorage(',') as (Country:Chararray,State:Chararray,Population:int)

Dump B:

```
grunt>B = LOAD 'S.txt' USING PigStorage(',') AS (Country:chararray,State:chararray,Population:int);
```

```
(USA,California,39776830)
(USA,Texas,28704330)
(USA,Florida,21312211)
(USA,New_York,19862512)
(USA,Pennsylvania,12823989)
(USA,Illinois,12768320)
(USA,Ohio,11694664)
(USA,Georgia,10545138)
(USA,North_Carolina,10390149)
(USA,Michigan,9991177)
(USA,New_Jersey,9032872)
(USA,Virginia,8525660)
(USA,Washington,7530552)
(USA,Arizona,7123898)
(USA,Massachusetts,6895917)
(USA,Tennessee,6782564)
(USA,Indiana,6699629)
(USA,Missouri,6135888)
(USA,Maryland,6079602)
(USA,Wisconsin,5818049)
(USA,Colorado,5684203)
(USA,Minnesota,5628162)
(USA,South_Carolina,5088916)
(USA,Alabama,4888949)
(USA,Louisiana,4682509)
(USA,Kentucky,4472265)
(USA,Oregon,4199563)
(USA,Oklahoma,3940521)
(USA,Connecticut,3588683)
(USA,Iowa,3160553)
(USA,Utah,3159345)
(USA,Nevada,3056824)
(USA,Arkansas,3020327)
(USA,Mississippi,2982785)
(USA,Kansas,2918515)
(USA,New_Mexico,2090708)
(USA,Nebraska,1932549)
(USA,West_Virginia,1803077)
(USA,Idaho,1753860)
(USA,Hawaii,1426393)
(USA,New_Hampshire,1350575)
(USA,Maine,1341582)
(USA,Montana,1062330)
(USA,Rhode_Island,1061712)
(USA,Delaware,971180)
(USA,South_Dakota,877790)
(USA,North_Dakota,755238)
(USA,Alaska,738068)
(USA,District_of_Columbia,703608)
(USA,Vermont,623960)
(USA,Wyoming,573720)
```

End output

Here we are loading State table in to A;

A = Load 'Cross.txt' using pigstorage(',') as (Country:Chararray,State:Chararray,Population:int)

Dump B:

```
grunt> A = LOAD 'Cross.txt' USING PigStorage(',') AS (Port_name:chararray, State:chararray, PortCode:int, Border:chararray, Date:chararray, container:chararray, Nos_People:int, Location:chararray);
grunt> G = limit A 55;;
grunt> DUMP G;
```

```
(Port Name,State,,Border,Date,Container,,Location)
(Calexico East,California,2507,US-Mexico Border,3/2/19 0:00,Trucks,34447,POINT (-115.48433000000001 32.67524))
(Van Buren,Maine,108,US-Canada Border,3/1/19 0:00,Rail Containers Full,428,POINT (-67.94271 47.16207))
(Otay Mesa,California,2506,US-Mexico Border,3/1/19 0:00,Trucks,81217,POINT (-117.05333 32.57333))
(Nogales,Arizona,2604,US-Mexico Border,3/1/19 0:00,Trains,62,POINT (-110.93361 31.340279999999996))
(Trout River,New York,715,US-Canada Border,3/2/19 0:00,Personal Vehicle Passengers,16377,POINT (-73.44253 44.990010000000005))
(Madawaska,Maine,109,US-Canada Border,3/1/19 0:00,Trucks,179,POINT (-68.3271 47.35446))
(Pembina,North Dakota,3401,US-Canada Border,3/1/19 0:00,Bus Passengers,1054,POINT (-97.24333 48.96639))
(Progreso,Texas,2309,US-Mexico Border,3/1/19 0:00,Truck Containers Empty,1808,POINT (-97.94889 26.061670000000003))
(Portal,North Dakota,3403,US-Canada Border,3/2/19 0:00,Rail Containers Empty,6685,POINT (-102.54917 48.99583))
(Champlain-Rouses Point,New York,712,US-Canada Border,3/1/19 0:00,Trucks,24759,POINT (-73.44694 44.98639))
(Opheim,Montana,3317,US-Canada Border,3/1/19 0:00,Personal Vehicles,235,POINT (-106.40265 48.85574))
(Neche,North Dakota,3404,US-Canada Border,3/1/19 0:00,Trucks,584,POINT (-97.55444 48.98611))
(Lancaster,Minnesota,3430,US-Canada Border,3/1/19 0:00,Personal Vehicles,1612,POINT (-96.82024 48.84911000000001))
(Derby Line,Vermont,209,US-Canada Border,3/1/19 0:00,Buses,80,POINT (-72.09944 45.005))
(Sarles,North Dakota,3409,US-Canada Border,3/1/19 0:00,Personal Vehicle Passengers,509,POINT (-98.99457 48.941050000000004))
(Wildhorse,Montana,3323,US-Canada Border,3/1/19 0:00,Truck Containers Full,122,POINT (-109.67761 48.54863))
(Derby Line,Vermont,209,US-Canada Border,3/1/19 0:00,Truck Containers Full,6483,POINT (-72.09944 45.005))
(Lynden,Washington,3023,US-Canada Border,3/1/19 0:00,Pedestrians,79,POINT (-122.44316000000002 48.94802))
(Vanceboro,Maine,105,US-Canada Border,3/1/19 0:00,Pedestrians,3,POINT (-67.42955 45.55984))
(San Ysidro,California,2504,US-Mexico Border,3/1/19 0:00,Bus Passengers,7779,POINT (-117.0266699999998 32.54306))
(Scobey,Montana,3309,US-Canada Border,3/1/19 0:00,Personal Vehicles,325,POINT (-105.44272 48.88707))
(Beecher Falls,Vermont,206,US-Canada Border,3/1/19 0:00,Trucks,901,POINT (-71.49664 45.00808))
(Calais,Maine,115,US-Canada Border,3/1/19 0:00,Train Passengers,30,POINT (-67.27917 45.18889))
(Massena,New York,704,US-Canada Border,3/1/19 0:00,Truck Containers Full,1489,POINT (-74.74 44.98944))
(Oroville,Washington,3019,US-Canada Border,3/2/19 0:00,Buses,12,POINT (-119.43444 48.93917))
(Hansboro,North Dakota,3415,US-Canada Border,3/1/19 0:00,Truck Containers Full,1,POINT (-99.38250000000001 48.95))
(Ferry,Washington,3013,US-Canada Border,3/1/19 0:00,Personal Vehicles,601,POINT (-118.58774 48.88627))
(Pembina,North Dakota,3401,US-Canada Border,3/1/19 0:00,Trucks,19150,POINT (-97.24333 48.96639))
(Tecate,California,2505,US-Mexico Border,3/1/19 0:00,Truck Containers Empty,1993,POINT (-116.62667 32.57722))
(Eastport,Idaho,3302,US-Canada Border,3/1/19 0:00,Trains,101,POINT (-116.1802799999998 48.99944))
(Walhalla,North Dakota,3407,US-Canada Border,3/1/19 0:00,Personal Vehicles,2399,POINT (-97.91778 48.92333))
(Calais,Maine,115,US-Canada Border,3/1/19 0:00,Trains,15,POINT (-67.27917 45.18889))
(Roma,Texas,2310,US-Mexico Border,3/1/19 0:00,Truck Containers Empty,748,POINT (-99.01833 26.40361))
(Naco,Arizona,2603,US-Mexico Border,3/1/19 0:00,Truck Containers Empty,139,POINT (-109.9475 31.335280000000004))
(Boquillas,Texas,2418,US-Mexico Border,3/1/19 0:00,Pedestrians,4362,POINT (-102.95 29.19000000000005))
(Raymond,Montana,3301,US-Canada Border,3/1/19 0:00,Buses,1,POINT (-104.575 48.99667))
(Porthill,Idaho,3308,US-Canada Border,3/1/19 0:00,Truck Containers Empty,282,POINT (-116.49828 48.99282))
(Norton,Vermont,211,US-Canada Border,3/1/19 0:00,Personal Vehicles,1350,POINT (-71.79528000000002 45.01))
(Sault Sainte Marie,Michigan,3803,US-Canada Border,3/1/19 0:00,Truck Containers Empty,769,POINT (-84.29889 46.47028))
(Antler,North Dakota,3413,US-Canada Border,3/1/19 0:00,Truck Containers Full,9,POINT (-101.28194 48.97083))
(Otay Mesa,California,2506,US-Mexico Border,3/1/19 0:00,Personal Vehicles,573597,POINT (-117.05333 32.57333))
(Eastport,Idaho,3302,US-Canada Border,3/1/19 0:00,Pedestrians,200,POINT (-116.1802799999998 48.99944))
(Rio Grande City,Texas,2307,US-Mexico Border,3/1/19 0:00,Pedestrians,4762,POINT (-98.82 26.37944))
(Portal,North Dakota,3403,US-Canada Border,3/1/19 0:00,Trains,135,POINT (-102.54917 48.99583))
(Del Rio,Texas,2302,US-Mexico Border,3/1/19 0:00,Truck Containers Full,5295,POINT (-100.89639 29.3625))
(Sasabe,Arizona,2606,US-Mexico Border,3/1/19 0:00,Personal Vehicle Passengers,4658,POINT (-111.54139 31.48861))
(Sault Sainte Marie,Michigan,3803,US-Canada Border,3/1/19 0:00,Truck Containers Full,3362,POINT (-84.29889 46.47028))
(Highgate Springs-Alburg,Vermont,212,US-Canada Border,3/1/19 0:00,Trains,25,POINT (-73.10583 44.97944900000004))
(Madawaska,Maine,109,US-Canada Border,3/1/19 0:00,Personal Vehicle Passengers,44387,POINT (-68.3271 47.35446))
(Sasabe,Arizona,2606,US-Mexico Border,3/1/19 0:00,Pedestrians,23,POINT (-111.54139 31.48861))
(Del Bonita,Montana,3322,US-Canada Border,3/1/19 0:00,Truck Containers Full,39,POINT (-112.32481 48.63274))
(Vanceboro,Maine,105,US-Canada Border,3/1/19 0:00,Personal Vehicles,1476,POINT (-67.42955 45.55984))
(Neche,North Dakota,3404,US-Canada Border,3/1/19 0:00,Buses,2,POINT (-97.55444 48.98611))
(Houlton,Maine,106,US-Canada Border,3/1/19 0:00,Truck Containers Full,5778,POINT (-67.84083 46.12611))
(Norton,Vermont,211,US-Canada Border,3/1/19 0:00,Trains,19,POINT (-71.79528000000002 45.01))
(Del Rio,Texas,2302,US-Mexico Border,3/1/19 0:00,Personal Vehicle Passengers,259047,POINT (-100.89639 29.3625))
(Houlton,Maine,106,US-Canada Border,3/1/19 0:00,Bus Passengers,669,POINT (-67.84083 46.12611))
(Metaline Falls,Washington,3025,US-Canada Border,3/1/19 0:00,Personal Vehicles,2300,POINT (-117.36037000000002 48.85798))
```

continue

continue

```
(Detroit,Michigan,3801,US-Canada Border,7/1/17 0:00,Bus Passengers,15877,POINT (-83.18222 42.38306000000001))
(Pembina,North Dakota,3401,US-Canada Border,7/1/17 0:00,Truck Containers Empty,2452,POINT (-97.24333 48.96639))
(Columbus,New Mexico,2406,US-Mexico Border,7/1/17 0:00,Personal Vehicles,38635,POINT (-107.63944 31.82749999999997))
(Jackman,Maine,104,US-Canada Border,7/1/17 0:00,Buses,11,POINT (-70.255 45.62389))
(Fortuna,North Dakota,3417,US-Canada Border,7/1/17 0:00,Buses,1,POINT (-103.77906 48.91086))
(Frontier,Washington,3020,US-Canada Border,7/1/17 0:00,Truck Containers Full,661,POINT (-117.78134000000001 48.910160000000005))
(Hidalgo,Texas,2305,US-Mexico Border,7/1/17 0:00,Pedestrians,188792,POINT (-98.26278 26.1))
(Norton,Vermont,211,US-Canada Border,7/1/17 0:00,Personal Vehicles,4624,POINT (-71.79528000000002 45.01))
(Opheim,Montana,3317,US-Canada Border,7/1/17 0:00,Personal Vehicles,412,POINT (-106.40265 48.85574))
(Beecher Falls,Vermont,206,US-Canada Border,7/1/17 0:00,Truck Containers Full,51,POINT (-71.49664 45.00808))
(Sarles,North Dakota,3409,US-Canada Border,7/1/17 0:00,Buses,16,POINT (-98.99457 48.94105000000004))
(Buffalo-Niagara Falls,New York,901,US-Canada Border,7/1/17 0:00,Rail Containers Empty,3214,POINT (-79.05694 43.09444))
(Fort Kent,Maine,110,US-Canada Border,7/1/17 0:00,Trucks,718,POINT (-68.58458 47.26878))
(Baudette,Minnesota,3424,US-Canada Border,7/1/17 0:00,Trucks,663,POINT (-94.72512 48.70731))
(Portal,North Dakota,3403,US-Canada Border,7/1/17 0:00,Trucks,6618,POINT (-102.54917 48.99583))
(Houlton,Maine,106,US-Canada Border,7/1/17 0:00,Personal Vehicles,27503,POINT (-67.84083 46.12611))
(Laurier,Washington,3016,US-Canada Border,7/1/17 0:00,Trucks,544,POINT (-118.22302 48.99892))
(El Paso,Texas,2402,US-Mexico Border,7/1/17 0:00,Rail Containers Empty,4587,POINT (-106.48639 31.758610000000004))
(Norton,Vermont,211,US-Canada Border,7/1/17 0:00,Truck Containers Empty,277,POINT (-71.79528000000002 45.01))
(Bridgewater,Maine,127,US-Canada Border,7/1/17 0:00,Trucks,604,POINT (-67.84262 46.41923))
(Eastport,Idaho,3302,US-Canada Border,7/1/17 0:00,Rail Containers Empty,697,POINT (-116.1802799999998 48.99944))
(Madawaska,Maine,189,US-Canada Border,7/1/17 0:00,Bus Passengers,141,POINT (-68.3271 47.35446))
(Otay Mesa,California,2506,US-Mexico Border,7/1/17 0:00,Personal Vehicles,660099,POINT (-117.05333 32.57333))
(Eastport,Idaho,3302,US-Canada Border,7/1/17 0:00,Personal Vehicles,12056,POINT (-116.1802799999998 48.99944))
(Vanceboro,Maine,105,US-Canada Border,7/1/17 0:00,Personal Vehicles,2768,POINT (-67.42955 45.55984))
(Roosville,Montana,3318,US-Canada Border,7/1/17 0:00,Personal Vehicles,31543,POINT (-115.04512000000001 48.88171))
(Frontier,Washington,3020,US-Canada Border,7/1/17 0:00,Pedestrians,12,POINT (-117.78134000000001 48.91016000000005))
(Raymond,Montana,3301,US-Canada Border,7/1/17 0:00,Personal Vehicle Passengers,6529,POINT (-104.575 48.99667))
(Eastport,Idaho,3302,US-Canada Border,7/1/17 0:00,Trucks,5202,POINT (-116.1802799999998 48.99944))
(Sarles,North Dakota,3409,US-Canada Border,7/1/17 0:00,Truck Containers Empty,35,POINT (-98.99457 48.94105000000004))
(Laurier,Washington,3016,US-Canada Border,7/1/17 0:00,Trains,12,POINT (-118.22302 48.99892))
(Eastport,Idaho,3302,US-Canada Border,7/1/17 0:00,Truck Containers Full,4077,POINT (-116.1802799999998 48.99944))
(San Luis,Arizona,2608,US-Mexico Border,7/1/17 0:00,Bus Passengers,28,POINT (-114.78139000000002 32.48694))
(Sweetgrass,Montana,3310,US-Canada Border,7/1/17 0:00,Rail Containers Empty,762,POINT (-111.95972 48.99611000000001))
(Progreso,Texas,2309,US-Mexico Border,7/1/17 0:00,Personal Vehicle Passengers,248337,POINT (-97.94889 26.06167000000003))
(Antler,North Dakota,3413,US-Canada Border,7/1/17 0:00,Personal Vehicles,242,POINT (-101.28194 48.97083))
(Point Roberts,Washington,3017,US-Canada Border,7/1/17 0:00,Personal Vehicles,91745,POINT (-123.07994 48.96978))
(Eagle Pass,Texas,2303,US-Mexico Border,7/1/17 0:00,Pedestrians,85166,POINT (-100.49917 28.70889))
(Baudette,Minnesota,3424,US-Canada Border,7/1/17 0:00,Personal Vehicle Passengers,34090,POINT (-94.72512 48.70731))
(Norton,Vermont,211,US-Canada Border,7/1/17 0:00,Train Passengers,32,POINT (-71.79528000000002 45.01))
(Portal,North Dakota,3403,US-Canada Border,7/1/17 0:00,Personal Vehicle Passengers,24344,POINT (-102.54917 48.99583))
(Sweetgrass,Montana,3310,US-Canada Border,7/1/17 0:00,Bus Passengers,1463,POINT (-111.95972 48.99611000000001))
(Ketchikan,Alaska,3102,US-Canada Border,7/1/17 0:00,Personal Vehicles,399,POINT (-131.64031 55.33897))
(Presidio,Texas,2403,US-Mexico Border,7/1/17 0:00,Personal Vehicle Passengers,138460,POINT (-104.37167 29.56056))
(Boundary,Washington,3015,US-Canada Border,7/1/17 0:00,Pedestrians,4,POINT (-117.89655000000002 48.54648))
(Nogales,Arizona,2604,US-Mexico Border,7/1/17 0:00,Truck Containers Full,12247,POINT (-110.93361 31.34027999999996))
(Lancaster,Minnesota,3430,US-Canada Border,7/1/17 0:00,Personal Vehicles,3458,POINT (-96.82024 48.84911000000001))
(Hidalgo,Texas,2305,US-Mexico Border,7/1/17 0:00,Personal Vehicle Passengers,836177,POINT (-98.26278 26.1))
(Lynden,Washington,3023,US-Canada Border,7/1/17 0:00,Trucks,3124,POINT (-122.44316000000002 48.94802))
(Ogdensburg,New York,701,US-Canada Border,7/1/17 0:00,Truck Containers Empty,708,POINT (-75.48167 44.70667))
(Vanceboro,Maine,105,US-Canada Border,7/1/17 0:00,Rail Containers Empty,947,POINT (-67.42955 45.55984))
(Sweetgrass,Montana,3310,US-Canada Border,7/1/17 0:00,Buses,45,POINT (-111.95972 48.99611000000001))
(Highgate Springs-Alburg,Vermont,212,US-Canada Border,7/1/17 0:00,Bus Passengers,10298,POINT (-73.10583 44.97944000000004))
(Ogdensburg,New York,701,US-Canada Border,7/1/17 0:00,Bus Passengers,204,POINT (-75.48167 44.70667))
(Champlain-Rouses Point,New York,712,US-Canada Border,7/1/17 0:00,Rail Containers Empty,714,POINT (-73.44694 44.98639))
(Fort Fairfield,Maine,107,US-Canada Border,7/1/17 0:00,Personal Vehicle Passengers,17043,POINT (-67.82965 46.76891))
(Wildhorse,Montana,3323,US-Canada Border,7/1/17 0:00,Truck Containers Empty,1,POINT (-109.67761 48.54863))
(Del Rio,Texas,2302,US-Mexico Border,7/1/17 0:00,Trucks,5834,POINT (-100.89639 29.3625))
(Madawaska,Maine,189,US-Canada Border,7/1/17 0:00,Personal Vehicle Passengers,52188,POINT (-68.3271 47.35446))
```

End output

Preview of State Table is as shown below with one data and its attributes.

Illustrate B:

| B | Country:chararray | State:chararray | Population:int |
|---|-------------------|-----------------|----------------|
|   | USA               | Arkansas        | 3020327        |

Preview of Boardercross Table is as shown below with one data and its attributes

Illustrate A:

| A | Port_name:chararray | State:chararray | PortCode:int | Border:chararray | Date:chararray | container:chararray    | Nos_People:int | Location:chararray                   |
|---|---------------------|-----------------|--------------|------------------|----------------|------------------------|----------------|--------------------------------------|
|   | Sweetgrass          | Montana         | 3310         | US-Canada Border | 5/1/18 0:00    | Truck Containers Empty | 2209           | POINT (-111.95972 48.99611000000001) |

Here are applying **Left Outer Join** on Both the table using **State**, to get output as below

J = JOIN A by State left outer, B by state;

```
grunt> J = Join A by State Left outer, B by State;
grunt> Describe J;
J: {A::Country: chararray,A::State: chararray,A::Population: int,B::Port_name: chararray,B::State: chararray,B::PortCode: int,B::Border: chararray,B::Date: chararray,B::Container: chararray,B::Nos_People: int,B::Location: chararray}
```

```
(USA,Iowa,3160553,,,...)
(USA,Ohio,11694664,Toledo-Sandusky,Ohio,4105,US-Canada Border,8/1/18 0:00,Personal Vehicles,122,POINT (-83.53303000000001 41.6648))
(USA,Ohio,11694664,Toledo-Sandusky,Ohio,4105,US-Canada Border,6/1/18 0:00,Personal Vehicles,98,POINT (-83.53303000000001 41.6648))
(USA,Ohio,11694664,Toledo-Sandusky,Ohio,4105,US-Canada Border,5/1/18 0:00,Personal Vehicles,17,POINT (-83.53303000000001 41.6648))
(USA,Ohio,11694664,Toledo-Sandusky,Ohio,4105,US-Canada Border,7/1/18 0:00,Personal Vehicles,111,POINT (-83.53303000000001 41.6648))
(USA,Ohio,11694664,Toledo-Sandusky,Ohio,4105,US-Canada Border,9/1/18 0:00,Personal Vehicles,33,POINT (-83.53303000000001 41.6648))
(USA,Utah,3159345,,,...)
(USA,Idaho,1753860,Porthill,Idaho,3308,US-Canada Border,1/1/19 0:00,Buses,3,POINT (-116.49828 48.99282))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,7/1/17 0:00,Truck Containers Full,4077,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,7/1/17 0:00,Trucks,5202,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,7/1/17 0:00,Personal Vehicles,12056,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,7/1/17 0:00,Rail Containers Empty,697,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Porthill,Idaho,3308,US-Canada Border,7/1/17 0:00,Truck Containers Empty,594,POINT (-116.49828 48.99282))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,7/1/17 0:00,Personal Vehicles,12703,POINT (-116.49828 48.99282))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,3/1/19 0:00,Trains,101,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Porthill,Idaho,3308,US-Canada Border,3/1/19 0:00,Truck Containers Empty,282,POINT (-116.49828 48.99282))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,3/1/19 0:00,Pedestrians,200,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,7/1/17 0:00,Buses,8,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Porthill,Idaho,3308,US-Canada Border,7/1/17 0:00,Truck Containers Full,37,POINT (-116.49828 48.99282))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,8/1/17 0:00,Trains,87,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,8/1/17 0:00,Truck Containers Empty,1028,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,8/1/17 0:00,Rail Containers Empty,643,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,8/1/17 0:00,Train Passengers,180,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Porthill,Idaho,3308,US-Canada Border,8/1/17 0:00,Pedestrians,31,POINT (-116.49828 48.99282))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,8/1/17 0:00,Trucks,5120,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,8/1/17 0:00,Pedestrians,180,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Porthill,Idaho,3308,US-Canada Border,8/1/17 0:00,Truck Containers Full,50,POINT (-116.49828 48.99282))
(USA,Idaho,1753860,Porthill,Idaho,3308,US-Canada Border,3/1/19 0:00,Trucks,310,POINT (-116.49828 48.99282))
(USA,Idaho,1753860,Porthill,Idaho,3308,US-Canada Border,8/1/17 0:00,Trucks,603,POINT (-116.49828 48.99282))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,8/1/17 0:00,Personal Vehicles,16735,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,3/1/19 0:00,Train Passengers,200,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Porthill,Idaho,3308,US-Canada Border,8/1/17 0:00,Bus Passengers,84,POINT (-116.49828 48.99282))
(USA,Idaho,1753860,Porthill,Idaho,3308,US-Canada Border,8/1/17 0:00,Buses,3,POINT (-116.49828 48.99282))
(USA,Idaho,1753860,Porthill,Idaho,3308,US-Canada Border,8/1/17 0:00,Personal Vehicles,13250,POINT (-116.49828 48.99282))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,8/1/17 0:00,Truck Containers Full,4276,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,8/1/17 0:00,Rail Containers Full,8821,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,8/1/17 0:00,Personal Vehicle Passengers,40997,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Porthill,Idaho,3308,US-Canada Border,8/1/17 0:00,Personal Vehicle Passengers,25242,POINT (-116.49828 48.99282))
(USA,Idaho,1753860,Porthill,Idaho,3308,US-Canada Border,8/1/17 0:00,Truck Containers Empty,553,POINT (-116.49828 48.99282))
(USA,Idaho,1753860,Porthill,Idaho,3308,US-Canada Border,9/1/17 0:00,Trucks,404,POINT (-116.49828 48.99282))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,9/1/17 0:00,Truck Containers Full,3882,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,9/1/17 0:00,Bus Passengers,310,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,9/1/17 0:00,Pedestrians,174,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,9/1/17 0:00,Personal Vehicle Passengers,18722,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,9/1/17 0:00,Rail Containers Full,8791,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Porthill,Idaho,3308,US-Canada Border,9/1/17 0:00,Personal Vehicles,10630,POINT (-116.49828 48.99282))
(USA,Idaho,1753860,Porthill,Idaho,3308,US-Canada Border,9/1/17 0:00,Truck Containers Full,31,POINT (-116.49828 48.99282))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,9/1/17 0:00,Train Passengers,174,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,9/1/17 0:00,Trains,90,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,9/1/17 0:00,Trucks,4591,POINT (-116.1802799999998 48.99944))
(USA,Idaho,1753860,Eastport,Idaho,3302,US-Canada Border,9/1/17 0:00,Truck Containers Empty,841,POINT (-116.1802799999998 48.99944))
```

continue

## Continue output

```
(Detroit,Michigan,3801,US-Canada Border,7/1/17 0:00,Bus Passengers,15877,POINT (-83.10222 42.383060000000001))
(Pembina,North Dakota,3401,US-Canada Border,7/1/17 0:00,Truck Containers Empty,2452,POINT (-97.24333 48.96639))
(Columbus,New Mexico,2406,US-Mexico Border,7/1/17 0:00,Personal Vehicles,38635,POINT (-107.63944 31.82749999999997))
(Jackman,Maine,184,US-Canada Border,7/1/17 0:00,Buses,11,POINT (-70.255 45.62389))
(Fortuna,North Dakota,3417,US-Canada Border,7/1/17 0:00,Buses,1,POINT (-103.77906 48.91086))
(Frontier,Washington,3020,US-Canada Border,7/1/17 0:00,Truck Containers Full,661,POINT (-117.78134000000001 48.910160000000005))
(Hidalgo,Texas,2305,US-Mexico Border,7/1/17 0:00,Pedestrians,188792,POINT (-98.26278 26.1))
(Norton,Vermont,211,US-Canada Border,7/1/17 0:00,Personal Vehicles,4624,POINT (-71.79528000000002 45.01))
(Opheim,Montana,3317,US-Canada Border,7/1/17 0:00,Personal Vehicles,412,POINT (-106.40265 48.85574))
(Beecher Falls,Vermont,206,US-Canada Border,7/1/17 0:00,Truck Containers Full,51,POINT (-71.49664 45.00808))
(Sarles,North Dakota,3409,US-Canada Border,7/1/17 0:00,Buses,16,POINT (-98.99457 48.94105000000004))
(Buffalo-Niagara Falls,New York,901,US-Canada Border,7/1/17 0:00,Rail Containers Empty,3214,POINT (-79.05694 43.09444))
(Fort Kent,Maine,110,US-Canada Border,7/1/17 0:00,Trucks,718,POINT (-68.58458 47.26878))
(Baudette,Minnesota,3424,US-Canada Border,7/1/17 0:00,Trucks,663,POINT (-94.72512 48.70731))
(Portal,North Dakota,3403,US-Canada Border,7/1/17 0:00,Trucks,6618,POINT (-102.54917 48.99583))
(Houlton,Maine,186,US-Canada Border,7/1/17 0:00,Personal Vehicles,27503,POINT (-67.84083 46.12611))
(Laurier,Washington,3016,US-Canada Border,7/1/17 0:00,Trucks,544,POINT (-118.22302 48.99892))
(El Paso,Texas,2402,US-Mexico Border,7/1/17 0:00,Rail Containers Empty,4587,POINT (-106.48639 31.758610000000004))
(Norton,Vermont,211,US-Canada Border,7/1/17 0:00,Truck Containers Empty,277,POINT (-71.79528000000002 45.01))
(Bridgewater,Maine,127,US-Canada Border,7/1/17 0:00,Trucks,604,POINT (-67.84262 46.41923))
(Eastport,Idaho,3302,US-Canada Border,7/1/17 0:00,Rail Containers Empty,697,POINT (-116.1802799999998 48.99944))
(Madawaska,Maine,109,US-Canada Border,7/1/17 0:00,Bus Passengers,141,POINT (-68.3271 47.35446))
(Otay Mesa,California,2506,US-Mexico Border,7/1/17 0:00,Personal Vehicles,660099,POINT (-117.05333 32.57333))
(Eastport,Idaho,3302,US-Canada Border,7/1/17 0:00,Personal Vehicles,12056,POINT (-116.1802799999998 48.99944))
(Vanceboro,Maine,105,US-Canada Border,7/1/17 0:00,Personal Vehicles,2768,POINT (-67.42955 45.55984))
(Roosville,Montana,3318,US-Canada Border,7/1/17 0:00,Personal Vehicles,31543,POINT (-115.04512000000001 48.88171))
(Frontier,Washington,3020,US-Canada Border,7/1/17 0:00,Pedestrians,12,POINT (-117.78134000000001 48.91016000000005))
(Raymond,Montana,3301,US-Canada Border,7/1/17 0:00,Personal Vehicle Passengers,6529,POINT (-104.575 48.99667))
(Eastport,Idaho,3302,US-Canada Border,7/1/17 0:00,Trucks,5202,POINT (-116.1802799999998 48.99944))
(Sarles,North Dakota,3409,US-Canada Border,7/1/17 0:00,Truck Containers Empty,35,POINT (-98.99457 48.94105000000004))
(Laurier,Washington,3016,US-Canada Border,7/1/17 0:00,Trains,12,POINT (-118.22302 48.99892))
(Eastport,Idaho,3302,US-Canada Border,7/1/17 0:00,Truck Containers Full,4077,POINT (-116.1802799999998 48.99944))
(San Luis,Arizona,2608,US-Mexico Border,7/1/17 0:00,Bus Passengers,20,POINT (-114.78139000000002 32.48694))
(Sweetgrass,Montana,3310,US-Canada Border,7/1/17 0:00,Rail Containers Empty,762,POINT (-111.95972 48.99611000000001))
(Progreso,Texas,2309,US-Mexico Border,7/1/17 0:00,Personal Vehicle Passengers,248337,POINT (-97.94889 26.06167000000003))
(Antler,North Dakota,3413,US-Canada Border,7/1/17 0:00,Personal Vehicles,742,POINT (-101.28194 48.97083))
(Point Roberts,Washington,3017,US-Canada Border,7/1/17 0:00,Personal Vehicles,91745,POINT (-123.07994 48.96978))
(Eagle Pass,Texas,2303,US-Mexico Border,7/1/17 0:00,Pedestrians,85166,POINT (-100.49917 28.70889))
(Baudette,Minnesota,3424,US-Canada Border,7/1/17 0:00,Personal Vehicle Passengers,34090,POINT (-94.72512 48.70731))
(Norton,Vermont,211,US-Canada Border,7/1/17 0:00,Train Passengers,32,POINT (-71.79528000000002 45.01))
(Portal,North Dakota,3403,US-Canada Border,7/1/17 0:00,Personal Vehicle Passengers,24344,POINT (-102.54917 48.99583))
(Sweetgrass,Montana,3310,US-Canada Border,7/1/17 0:00,Bus Passengers,1463,POINT (-111.95972 48.99611000000001))
(Ketchikan,Alaska,3102,US-Canada Border,7/1/17 0:00,Personal Vehicles,399,POINT (-131.64031 55.33897))
(Presidio,Texas,2403,US-Mexico Border,7/1/17 0:00,Personal Vehicle Passengers,138460,POINT (-104.37167 29.56056))
(Boundary,Washington,3015,US-Canada Border,7/1/17 0:00,Pedestrians,4,POINT (-117.89655000000002 48.54648))
(Nogales,Arizona,2604,US-Mexico Border,7/1/17 0:00,Truck Containers Full,12247,POINT (-110.93361 31.34027999999996))
(Lancaster,Minnesota,3430,US-Canada Border,7/1/17 0:00,Personal Vehicles,3458,POINT (-96.82024 48.84911000000001))
(Hidalgo,Texas,2305,US-Mexico Border,7/1/17 0:00,Personal Vehicle Passengers,836177,POINT (-98.26278 26.1))
(Lynden,Washington,3023,US-Canada Border,7/1/17 0:00,Trucks,3124,POINT (-122.44316000000002 48.94882))
(Ogdensburg,New York,701,US-Canada Border,7/1/17 0:00,Rail Containers Empty,708,POINT (-75.48167 44.70667))
(Vanceboro,Maine,105,US-Canada Border,7/1/17 0:00,Rail Containers Empty,947,POINT (-67.42955 45.55984))
(Sweetgrass,Montana,3310,US-Canada Border,7/1/17 0:00,Buses,45,POINT (-111.95972 48.99611000000001))
(Highgate Springs-Alburg,Vermont,212,US-Canada Border,7/1/17 0:00,Bus Passengers,10298,POINT (-73.10583 44.97944000000004))
(Ogdensburg,New York,701,US-Canada Border,7/1/17 0:00,Bus Passengers,204,POINT (-75.48167 44.70667))
(Champlain-Rouses Point,New York,712,US-Canada Border,7/1/17 0:00,Rail Containers Empty,714,POINT (-73.44694 44.98639))
(Fort Fairfield,Maine,107,US-Canada Border,7/1/17 0:00,Personal Vehicle Passengers,17043,POINT (-67.82965 46.76891))
(Wildhorse,Montana,3323,US-Canada Border,7/1/17 0:00,Truck Containers Empty,1,POINT (-109.67761 48.54863))
(Del Rio,Texas,2302,US-Mexico Border,7/1/17 0:00,Trucks,5834,POINT (-100.89639 29.3625))
(Madawaska,Maine,109,US-Canada Border,7/1/17 0:00,Personal Vehicle Passengers,52188,POINT (-68.3271 47.35446))
```

## End output

## Applying Foreach on Join using State, Population, Port\_name, Date, container,Nos.

F = Foreach J generate A::State,A::Population,B::Port\_name, Date, B::container, B::Nos\_people;

Describe F:

Dump F:

```
grunt> F = Foreach J Generate A::State, A::Population, B::Port_name, B::Date, B::container, B::Nos_People;
grunt> describe F;
F: {A::State: chararray,A::Population: int,B::Port_name: chararray,B::Date: chararray,B::container: chararray,B::Nos_People: int}
```

```
(Iowa,3160553,,,)
(Ohio,11694664,Toledo-Sandusky,8/1/18 0:00,Personal Vehicles,122)
(Ohio,11694664,Toledo-Sandusky,6/1/18 0:00,Personal Vehicles,90)
(Ohio,11694664,Toledo-Sandusky,5/1/18 0:00,Personal Vehicles,17)
(Ohio,11694664,Toledo-Sandusky,7/1/18 0:00,Personal Vehicles,111)
(Ohio,11694664,Toledo-Sandusky,9/1/18 0:00,Personal Vehicles,33)
(Utah,3159345,,,)
(Idaho,1753860,Porthill,1/1/19 0:00,Buses,3)
(Idaho,1753860,Eastport,7/1/17 0:00,Truck Containers Full,4077)
(Idaho,1753860,Eastport,7/1/17 0:00,Trucks,5202)
(Idaho,1753860,Eastport,7/1/17 0:00,Personal Vehicles,12056)
(Idaho,1753860,Eastport,7/1/17 0:00,Rail Containers Empty,697)
(Idaho,1753860,Porthill,7/1/17 0:00,Truck Containers Empty,594)
(Idaho,1753860,Porthill,7/1/17 0:00,Personal Vehicles,12703)
(Idaho,1753860,Eastport,3/1/19 0:00,Trains,101)
(Idaho,1753860,Porthill,3/1/19 0:00,Truck Containers Empty,282)
(Idaho,1753860,Eastport,3/1/19 0:00,Pedestrians,200)
(Idaho,1753860,Eastport,7/1/17 0:00,Buses,8)
(Idaho,1753860,Porthill,7/1/17 0:00,Truck Containers Full,37)
(Idaho,1753860,Eastport,8/1/17 0:00,Trains,87)
(Idaho,1753860,Eastport,8/1/17 0:00,Truck Containers Empty,1028)
(Idaho,1753860,Eastport,8/1/17 0:00,Rail Containers Empty,643)
(Idaho,1753860,Eastport,8/1/17 0:00,Train Passengers,180)
(Idaho,1753860,Porthill,8/1/17 0:00,Pedestrians,31)
(Idaho,1753860,Eastport,8/1/17 0:00,Trucks,5120)
(Idaho,1753860,Eastport,8/1/17 0:00,Pedestrians,180)
(Idaho,1753860,Porthill,8/1/17 0:00,Truck Containers Full,50)
(Idaho,1753860,Porthill,3/1/19 0:00,Trucks,310)
(Idaho,1753860,Porthill,8/1/17 0:00,Trucks,603)
(Idaho,1753860,Eastport,8/1/17 0:00,Personal Vehicles,16735)
(Idaho,1753860,Eastport,3/1/19 0:00,Train Passengers,200)
(Idaho,1753860,Porthill,8/1/17 0:00,Bus Passengers,84)
(Idaho,1753860,Porthill,8/1/17 0:00,Buses,3)
(Idaho,1753860,Porthill,8/1/17 0:00,Personal Vehicles,13250)
(Idaho,1753860,Eastport,8/1/17 0:00,Truck Containers Full,4276)
(Idaho,1753860,Eastport,8/1/17 0:00,Rail Containers Full,8821)
(Idaho,1753860,Eastport,8/1/17 0:00,Personal Vehicle Passengers,40997)
(Idaho,1753860,Porthill,8/1/17 0:00,Personal Vehicle Passengers,25242)
(Idaho,1753860,Porthill,8/1/17 0:00,Truck Containers Empty,553)
(Idaho,1753860,Porthill,9/1/17 0:00,Trucks,404)
(Idaho,1753860,Eastport,9/1/17 0:00,Truck Containers Full,3882)
(Idaho,1753860,Eastport,9/1/17 0:00,Bus Passengers,310)
(Idaho,1753860,Eastport,9/1/17 0:00,Pedestrians,174)
(Idaho,1753860,Eastport,9/1/17 0:00,Personal Vehicle Passengers,18722)
(Idaho,1753860,Eastport,9/1/17 0:00,Rail Containers Full,8791)
(Idaho,1753860,Porthill,9/1/17 0:00,Personal Vehicles,10630)
(Idaho,1753860,Porthill,9/1/17 0:00,Truck Containers Full,31)
(Idaho,1753860,Eastport,9/1/17 0:00,Train Passengers,174)
(Idaho,1753860,Eastport,9/1/17 0:00,Trains,90)
(Idaho,1753860,Eastport,9/1/17 0:00,Trucks,4591)
(Idaho,1753860,Eastport,9/1/17 0:00,Truck Containers Empty,841)
```

Continue

```
(Washington,7530552,Danville,7/1/17 0:00,Personal Vehicle Passengers,6123)
(Washington,7530552,Blaine,12/1/18 0:00,Train Passengers,4475)
(Washington,7530552,Laurier,2/1/18 0:00,Rail Containers Full,257)
(Washington,7530552,Boundary,8/1/18 0:00,Trucks,34)
(Washington,7530552,Metaline Falls,1/1/18 0:00,Truck Containers Empty,608)
(Washington,7530552,Point Roberts,8/1/18 0:00,Truck Containers Full,178)
(Washington,7530552,Sumas,2/1/18 0:00,Buses,26)
(Washington,7530552,Danville,8/1/18 0:00,Truck Containers Empty,71)
(Washington,7530552,Frontier,8/1/17 0:00,Pedestrians,7)
(Washington,7530552,Lynden,8/1/17 0:00,Pedestrians,124)
(Washington,7530552,Oroville,8/1/18 0:00,Truck Containers Empty,869)
(Washington,7530552,Sumas,2/1/18 0:00,Bus Passengers,694)
(Washington,7530552,Laurier,8/1/18 0:00,Personal Vehicle Passengers,10897)
(Washington,7530552,Lynden,1/1/19 0:00,Trucks,3386)
(Washington,7530552,Sumas,2/1/18 0:00,Train Passengers,33)
(Washington,7530552,Blaine,2/1/18 0:00,Rail Containers Full,6029)
(Washington,7530552,Sumas,8/1/18 0:00,Personal Vehicle Passengers,179464)
(Washington,7530552,Sumas,8/1/18 0:00,Personal Vehicles,93397)
(Washington,7530552,Lynden,2/1/18 0:00,Pedestrians,65)
(Washington,7530552,Ferry,8/1/18 0:00,Truck Containers Empty,27)
(Washington,7530552,Port Angeles,2/1/18 0:00,Bus Passengers,162)
(Washington,7530552,Frontier,3/1/19 0:00,Trucks,1873)
(Washington,7530552,Danville,2/1/18 0:00,Truck Containers Full,5)
(Washington,7530552,Anacortes,8/1/18 0:00,Personal Vehicle Passengers,9678)
(Washington,7530552,Laurier,1/1/18 0:00,Train Passengers,28)
(Washington,7530552,Blaine,1/1/18 0:00,Train Passengers,3715)
(Washington,7530552,Port Angeles,2/1/18 0:00,Personal Vehicle Passengers,4281)
(Washington,7530552,Point Roberts,8/1/18 0:00,Trucks,1338)
(Washington,7530552,Anacortes,8/1/18 0:00,Personal Vehicles,3529)
(Washington,7530552,Frontier,8/1/18 0:00,Truck Containers Empty,1337)
(Washington,7530552,Danville,2/1/18 0:00,Personal Vehicle Passengers,3672)
(Washington,7530552,Port Angeles,8/1/18 0:00,Truck Containers Full,35)
(Washington,7530552,Lynden,1/1/18 0:00,Trucks,3338)
(Washington,7530552,Metaline Falls,8/1/18 0:00,Buses,7)
(Washington,7530552,Blaine,1/1/19 0:00,Personal Vehicles,282837)
(Washington,7530552,Anacortes,1/1/19 0:00,Personal Vehicles,496)
(Washington,7530552,Point Roberts,8/1/18 0:00,Truck Containers Empty,1160)
(Washington,7530552,Blaine,8/1/18 0:00,Bus Passengers,37280)
(Washington,7530552,Laurier,1/1/18 0:00,Rail Containers Full,218)
(Connecticut,3588683,„„)
(Mississippi,2982785,„„)
(North_Dakota,755238,„„)
(Pennsylvania,12823989,„„)
(Rhode_Island,1061712,„„)
(South_Dakota,877790,„„)
(Massachusetts,6895917,„„)
(New_Hampshire,1350575,„„)
(West_Virginia,1803077,„„)
(North_Carolina,10390149,„„)
(South_Carolina,5088916,„„)
(District_of_Columbia,703608,„„)
grunt> █
```

End output

## Applying Filter F by container.

H = Filter F by container == 'Buses':

Describe H:

Dump H:

```
grunt> H = Filter F By container == 'Buses';
grunt> describe H;
H: {A::State: chararray,A::Population: int,B::Port_name: chararray,B::Date: chararray,B::container: chararray,B::Nos_People: int}
grunt> DUMP H;
```

```
(Idaho,1753860,Porthill,1/1/19 0:00,Buses,3)
(Idaho,1753860,Eastport,7/1/17 0:00,Buses,8)
(Idaho,1753860,Porthill,8/1/17 0:00,Buses,3)
(Idaho,1753860,Eastport,9/1/17 0:00,Buses,8)
(Idaho,1753860,Eastport,3/1/19 0:00,Buses,2)
(Idaho,1753860,Porthill,9/1/17 0:00,Buses,2)
(Idaho,1753860,Eastport,10/1/17 0:00,Buses,16)
(Idaho,1753860,Porthill,3/1/19 0:00,Buses,1)
(Idaho,1753860,Porthill,11/1/17 0:00,Buses,2)
(Idaho,1753860,Eastport,11/1/17 0:00,Buses,10)
(Idaho,1753860,Eastport,12/1/17 0:00,Buses,3)
(Idaho,1753860,Eastport,1/1/18 0:00,Buses,7)
(Idaho,1753860,Eastport,2/1/19 0:00,Buses,6)
(Idaho,1753860,Eastport,2/1/18 0:00,Buses,9)
(Idaho,1753860,Porthill,2/1/19 0:00,Buses,2)
(Idaho,1753860,Eastport,3/1/18 0:00,Buses,5)
(Idaho,1753860,Porthill,4/1/18 0:00,Buses,3)
(Idaho,1753860,Eastport,4/1/18 0:00,Buses,2)
(Idaho,1753860,Porthill,5/1/18 0:00,Buses,1)
(Idaho,1753860,Eastport,5/1/18 0:00,Buses,8)
(Idaho,1753860,Porthill,6/1/18 0:00,Buses,3)
(Idaho,1753860,Eastport,6/1/18 0:00,Buses,1)
(Idaho,1753860,Eastport,7/1/18 0:00,Buses,9)
(Idaho,1753860,Porthill,8/1/18 0:00,Buses,1)
(Idaho,1753860,Eastport,8/1/18 0:00,Buses,4)
(Idaho,1753860,Porthill,9/1/18 0:00,Buses,1)
(Idaho,1753860,Eastport,9/1/18 0:00,Buses,8)
(Idaho,1753860,Eastport,10/1/18 0:00,Buses,10)
(Idaho,1753860,Porthill,10/1/18 0:00,Buses,2)
(Idaho,1753860,Eastport,1/1/19 0:00,Buses,6)
(Idaho,1753860,Porthill,11/1/18 0:00,Buses,1)
(Idaho,1753860,Eastport,11/1/18 0:00,Buses,3)
(Idaho,1753860,Porthill,12/1/18 0:00,Buses,1)
(Idaho,1753860,Eastport,12/1/18 0:00,Buses,2)
```

continue

(Washington,7530552,Sumas,9/1/17 0:00,Buses,37)  
(Washington,7530552,Frontier,9/1/17 0:00,Buses,9)  
(Washington,7530552,Laurier,6/1/18 0:00,Buses,1)  
(Washington,7530552,Port Angeles,5/1/18 0:00,Buses,16)  
(Washington,7530552,Sumas,11/1/18 0:00,Buses,41)  
(Washington,7530552,Lynden,8/1/17 0:00,Buses,2)  
(Washington,7530552,Metaline Falls,8/1/17 0:00,Buses,1)  
(Washington,7530552,Frontier,8/1/17 0:00,Buses,7)  
(Washington,7530552,Point Roberts,6/1/18 0:00,Buses,20)  
(Washington,7530552,Oroville,6/1/18 0:00,Buses,4)  
(Washington,7530552,Anacortes,5/1/18 0:00,Buses,1)  
(Washington,7530552,Sumas,12/1/18 0:00,Buses,21)  
(Washington,7530552,Port Angeles,8/1/17 0:00,Buses,8)  
(Washington,7530552,Laurier,3/1/19 0:00,Buses,1)  
(Washington,7530552,Point Roberts,8/1/17 0:00,Buses,15)  
(Washington,7530552,Frontier,5/1/18 0:00,Buses,8)  
(Washington,7530552,Point Roberts,12/1/18 0:00,Buses,30)  
(Washington,7530552,Laurier,12/1/18 0:00,Buses,2)  
(Washington,7530552,Sumas,8/1/17 0:00,Buses,7)  
(Washington,7530552,Anacortes,6/1/18 0:00,Buses,4)  
(Washington,7530552,Metaline Falls,4/1/18 0:00,Buses,8)  
(Washington,7530552,Blaine,12/1/18 0:00,Buses,877)  
(Washington,7530552,Frontier,4/1/18 0:00,Buses,8)  
(Washington,7530552,Blaine,8/1/17 0:00,Buses,1763)  
(Washington,7530552,Laurier,4/1/18 0:00,Buses,3)  
(Washington,7530552,Oroville,12/1/18 0:00,Buses,11)  
(Washington,7530552,Anacortes,4/1/18 0:00,Buses,1)  
(Washington,7530552,Frontier,6/1/18 0:00,Buses,4)  
(Washington,7530552,Frontier,2/1/18 0:00,Buses,31)  
(Washington,7530552,Frontier,8/1/18 0:00,Buses,6)  
(Washington,7530552,Blaine,8/1/18 0:00,Buses,1628)  
(Washington,7530552,Lynden,2/1/19 0:00,Buses,3)  
(Washington,7530552,Blaine,2/1/18 0:00,Buses,949)  
(Washington,7530552,Frontier,7/1/17 0:00,Buses,2)  
(Washington,7530552,Point Roberts,8/1/18 0:00,Buses,6)  
(Washington,7530552,Sumas,2/1/19 0:00,Buses,20)  
(Washington,7530552,Oroville,2/1/19 0:00,Buses,13)  
(Washington,7530552,Oroville,1/1/18 0:00,Buses,11)  
(Washington,7530552,Frontier,1/1/18 0:00,Buses,29)  
(Washington,7530552,Laurier,1/1/19 0:00,Buses,3)  
(Washington,7530552,Point Roberts,1/1/18 0:00,Buses,44)  
(Washington,7530552,Metaline Falls,1/1/18 0:00,Buses,15)  
(Washington,7530552,Oroville,2/1/18 0:00,Buses,21)  
(Washington,7530552,Boundary,2/1/18 0:00,Buses,1)  
(Washington,7530552,Lynden,8/1/18 0:00,Buses,10)  
(Washington,7530552,Oroville,8/1/18 0:00,Buses,9)  
(Washington,7530552,Port Angeles,8/1/18 0:00,Buses,12)  
(Washington,7530552,Boundary,1/1/18 0:00,Buses,1)  
(Washington,7530552,Oroville,1/1/19 0:00,Buses,13)  
(Washington,7530552,Sumas,2/1/18 0:00,Buses,26)  
(Washington,7530552,Metaline Falls,8/1/18 0:00,Buses,7)

End output

## Applying Order by on H by State in ascending order

O = Order H by State ASC:

Describe O:

Dump O:

```
grunt> O = Order H by State ASC;
grunt> Describe O;
O: {A::State: chararray,A::Population: int,B::Port_name: chararray,B::Date: chararray,B::container: chararray,B::Nos_People: int}
grunt> DUMP O;
```

```
(Alaska,738068,Dalton Cache,7/1/17 0:00,Buses,12)
(Alaska,738068,Ketchikan,9/1/18 0:00,Buses,1)
(Alaska,738068,Skagway,9/1/18 0:00,Buses,1505)
(Alaska,738068,Alcan,10/1/18 0:00,Buses,1)
(Alaska,738068,Dalton Cache,8/1/18 0:00,Buses,18)
(Alaska,738068,Alcan,9/1/18 0:00,Buses,13)
(Alaska,738068,Skagway,9/1/17 0:00,Buses,1413)
(Alaska,738068,Alcan,8/1/18 0:00,Buses,48)
(Alaska,738068,Alcan,4/1/18 0:00,Buses,1)
(Alaska,738068,Dalton Cache,11/1/18 0:00,Buses,1)
(Alaska,738068,Dalton Cache,9/1/18 0:00,Buses,5)
(Alaska,738068,Skagway,5/1/18 0:00,Buses,1851)
(Alaska,738068,Alcan,9/1/17 0:00,Buses,8)
(Alaska,738068,Skagway,8/1/18 0:00,Buses,2768)
(Alaska,738068,Alcan,5/1/18 0:00,Buses,12)
(Alaska,738068,Skagway,6/1/18 0:00,Buses,2687)
(Alaska,738068,Ketchikan,6/1/18 0:00,Buses,4)
(Alaska,738068,Dalton Cache,7/1/18 0:00,Buses,16)
(Alaska,738068,Dalton Cache,6/1/18 0:00,Buses,9)
(Alaska,738068,Skagway,7/1/18 0:00,Buses,2818)
(Alaska,738068,Dalton Cache,8/1/17 0:00,Buses,13)
(Alaska,738068,Ketchikan,7/1/18 0:00,Buses,4)
(Alaska,738068,Dalton Cache,5/1/18 0:00,Buses,4)
(Alaska,738068,Skagway,8/1/17 0:00,Buses,3027)
(Alaska,738068,Alcan,6/1/18 0:00,Buses,47)
(Alaska,738068,Alcan,7/1/18 0:00,Buses,60)
(Alaska,738068,Dalton Cache,10/1/17 0:00,Buses,2)
(Alaska,738068,Dalton Cache,9/1/17 0:00,Buses,4)
(Alaska,738068,Ketchikan,8/1/18 0:00,Buses,2)
(Alaska,738068,Skagway,10/1/18 0:00,Buses,18)
(Alaska,738068,Ketchikan,8/1/17 0:00,Buses,1)
(Alaska,738068,Skagway,7/1/17 0:00,Buses,2840)
(Alaska,738068,Alcan,7/1/17 0:00,Buses,65)
(Arizona,7123898,Lukeville,7/1/18 0:00,Buses,59)
(Arizona,7123898,Douglas,7/1/18 0:00,Buses,258)
(Arizona,7123898,Naco,3/1/18 0:00,Buses,3)
(Arizona,7123898,Douglas,2/1/19 0:00,Buses,44)
(Arizona,7123898,Douglas,3/1/18 0:00,Buses,200)
(Arizona,7123898,Lukeville,3/1/18 0:00,Buses,55)
(Arizona,7123898,San Luis,7/1/18 0:00,Buses,11)
(Arizona,7123898,Nogales,7/1/18 0:00,Buses,830)
(Arizona,7123898,San Luis,3/1/18 0:00,Buses,19)
(Arizona,7123898,Nogales,2/1/19 0:00,Buses,717)
(Arizona,7123898,San Luis,2/1/19 0:00,Buses,17)
(Arizona,7123898,Douglas,4/1/18 0:00,Buses,190)
```

Continue

```
(Washington,7530552,Sumas,9/1/17 0:00,Buses,37)
(Washington,7530552,Frontier,9/1/17 0:00,Buses,9)
(Washington,7530552,Laurier,6/1/18 0:00,Buses,1)
(Washington,7530552,Port Angeles,5/1/18 0:00,Buses,16)
(Washington,7530552,Sumas,11/1/18 0:00,Buses,41)
(Washington,7530552,Lynden,8/1/17 0:00,Buses,2)
(Washington,7530552,Metaline Falls,8/1/17 0:00,Buses,1)
(Washington,7530552,Frontier,8/1/17 0:00,Buses,7)
(Washington,7530552,Point Roberts,6/1/18 0:00,Buses,20)
(Washington,7530552,Oroville,6/1/18 0:00,Buses,4)
(Washington,7530552,Anacortes,5/1/18 0:00,Buses,1)
(Washington,7530552,Sumas,12/1/18 0:00,Buses,21)
(Washington,7530552,Port Angeles,8/1/17 0:00,Buses,8)
(Washington,7530552,Laurier,3/1/19 0:00,Buses,1)
(Washington,7530552,Point Roberts,8/1/17 0:00,Buses,15)
(Washington,7530552,Frontier,5/1/18 0:00,Buses,8)
(Washington,7530552,Point Roberts,12/1/18 0:00,Buses,30)
(Washington,7530552,Laurier,12/1/18 0:00,Buses,2)
(Washington,7530552,Sumas,8/1/17 0:00,Buses,7)
(Washington,7530552,Anacortes,6/1/18 0:00,Buses,4)
(Washington,7530552,Metaline Falls,4/1/18 0:00,Buses,8)
(Washington,7530552,Blaine,12/1/18 0:00,Buses,877)
(Washington,7530552,Frontier,4/1/18 0:00,Buses,8)
(Washington,7530552,Blaine,8/1/17 0:00,Buses,1763)
(Washington,7530552,Laurier,4/1/18 0:00,Buses,3)
(Washington,7530552,Oroville,12/1/18 0:00,Buses,11)
(Washington,7530552,Anacortes,4/1/18 0:00,Buses,1)
(Washington,7530552,Frontier,6/1/18 0:00,Buses,4)
(Washington,7530552,Frontier,2/1/18 0:00,Buses,31)
(Washington,7530552,Frontier,8/1/18 0:00,Buses,6)
(Washington,7530552,Blaine,8/1/18 0:00,Buses,1628)
(Washington,7530552,Lynden,2/1/19 0:00,Buses,3)
(Washington,7530552,Blaine,2/1/18 0:00,Buses,949)
(Washington,7530552,Frontier,7/1/17 0:00,Buses,2)
(Washington,7530552,Point Roberts,8/1/18 0:00,Buses,6)
(Washington,7530552,Sumas,2/1/19 0:00,Buses,20)
(Washington,7530552,Oroville,2/1/19 0:00,Buses,13)
(Washington,7530552,Oroville,1/1/18 0:00,Buses,11)
(Washington,7530552,Frontier,1/1/18 0:00,Buses,29)
(Washington,7530552,Laurier,1/1/19 0:00,Buses,3)
(Washington,7530552,Point Roberts,1/1/18 0:00,Buses,44)
(Washington,7530552,Metaline Falls,1/1/18 0:00,Buses,15)
(Washington,7530552,Oroville,2/1/18 0:00,Buses,21)
(Washington,7530552,Boundary,2/1/18 0:00,Buses,1)
(Washington,7530552,Lynden,8/1/18 0:00,Buses,10)
(Washington,7530552,Oroville,8/1/18 0:00,Buses,9)
(Washington,7530552,Port Angeles,8/1/18 0:00,Buses,12)
(Washington,7530552,Boundary,1/1/18 0:00,Buses,1)
(Washington,7530552,Oroville,1/1/19 0:00,Buses,13)
(Washington,7530552,Sumas,2/1/18 0:00,Buses,26)
(Washington,7530552,Metaline Falls,8/1/18 0:00,Buses,7)
```

end output

# Applying Filter again on O By State California

k = Filter O by State == 'California':

Describe k:

Dump k:

```
grunt> k = Filter O By State == 'California';
grunt> Describe k;
k: {A::State: chararray,A::Population: int,B::Port_name: chararray,B::Date: chararray,B::container: chararray,B::Nos_People: int}
grunt> DUMP k;
```

```
(California,39776830,Otay Mesa,6/1/18 0:00,Buses,2521)
(California,39776830,Otay Mesa,2/1/19 0:00,Buses,878)
(California,39776830,San Ysidro,6/1/18 0:00,Buses,2748)
(California,39776830,Calexico East,4/1/18 0:00,Buses,228)
(California,39776830,Calexico East,12/1/18 0:00,Buses,193)
(California,39776830,Otay Mesa,4/1/18 0:00,Buses,2244)
(California,39776830,San Ysidro,4/1/18 0:00,Buses,2732)
(California,39776830,San Ysidro,12/1/18 0:00,Buses,2995)
(California,39776830,San Ysidro,2/1/19 0:00,Buses,2952)
(California,39776830,Calexico East,7/1/18 0:00,Buses,212)
(California,39776830,Calexico East,3/1/18 0:00,Buses,235)
(California,39776830,Otay Mesa,3/1/18 0:00,Buses,990)
(California,39776830,San Ysidro,3/1/18 0:00,Buses,2749)
(California,39776830,Otay Mesa,7/1/18 0:00,Buses,3239)
(California,39776830,Otay Mesa,2/1/18 0:00,Buses,1727)
(California,39776830,San Ysidro,2/1/18 0:00,Buses,2395)
(California,39776830,San Ysidro,7/1/18 0:00,Buses,2632)
(California,39776830,Calexico East,2/1/18 0:00,Buses,191)
(California,39776830,Otay Mesa,8/1/18 0:00,Buses,3249)
(California,39776830,Calexico East,1/1/18 0:00,Buses,216)
(California,39776830,San Ysidro,8/1/18 0:00,Buses,2590)
(California,39776830,Otay Mesa,1/1/18 0:00,Buses,2173)
(California,39776830,Calexico East,8/1/18 0:00,Buses,223)
(California,39776830,San Ysidro,1/1/18 0:00,Buses,2663)
(California,39776830,Calexico East,9/1/18 0:00,Buses,211)
(California,39776830,San Ysidro,12/1/17 0:00,Buses,2794)
(California,39776830,Calexico East,12/1/17 0:00,Buses,228)
(California,39776830,Otay Mesa,12/1/17 0:00,Buses,2298)
(California,39776830,San Ysidro,9/1/18 0:00,Buses,2531)
(California,39776830,Otay Mesa,9/1/18 0:00,Buses,2929)
(California,39776830,Otay Mesa,11/1/17 0:00,Buses,2458)
(California,39776830,Otay Mesa,1/1/19 0:00,Buses,1351)
(California,39776830,Calexico East,3/1/19 0:00,Buses,105)
(California,39776830,Calexico East,10/1/18 0:00,Buses,265)
(California,39776830,San Ysidro,11/1/17 0:00,Buses,2788)
(California,39776830,Calexico East,11/1/17 0:00,Buses,223)
(California,39776830,San Ysidro,10/1/18 0:00,Buses,3076)
(California,39776830,San Ysidro,1/1/19 0:00,Buses,3110)
(California,39776830,Otay Mesa,10/1/18 0:00,Buses,3422)
(California,39776830,Otay Mesa,10/1/17 0:00,Buses,2690)
(California,39776830,San Ysidro,10/1/17 0:00,Buses,2890)
(California,39776830,Calexico East,10/1/17 0:00,Buses,308)
(California,39776830,Calexico East,11/1/18 0:00,Buses,196)
(California,39776830,Otay Mesa,9/1/17 0:00,Buses,2241)
(California,39776830,Otay Mesa,11/1/18 0:00,Buses,2802)
(California,39776830,Calexico East,9/1/17 0:00,Buses,271)
(California,39776830,San Ysidro,11/1/18 0:00,Buses,2296)
(California,39776830,San Ysidro,9/1/17 0:00,Buses,2911)
(California,39776830,Calexico East,8/1/17 0:00,Buses,221)
(California,39776830,Otay Mesa,8/1/17 0:00,Buses,2502)
(California,39776830,Calexico East,1/1/19 0:00,Buses,208)
(California,39776830,Otay Mesa,12/1/18 0:00,Buses,2089)
(California,39776830,San Ysidro,8/1/17 0:00,Buses,2838)
(California,39776830,San Ysidro,3/1/19 0:00,Buses,3329)
(California,39776830,Calexico East,7/1/17 0:00,Buses,238)
(California,39776830,Otay Mesa,3/1/19 0:00,Buses,916)
(California,39776830,San Ysidro,7/1/17 0:00,Buses,2968)
(California,39776830,Otay Mesa,5/1/18 0:00,Buses,2148)
(California,39776830,San Ysidro,5/1/18 0:00,Buses,2651)
(California,39776830,Calexico East,5/1/18 0:00,Buses,234)
(California,39776830,Calexico East,6/1/18 0:00,Buses,206)
(California,39776830,Calexico East,2/1/19 0:00,Buses,137)
```

## Applying Group By on k By Date

Gm = Group k by Date:

Describe Gm:

Dump Gm:

```
grunt> Gm = Group k by B::Date;
grunt> describe Gm;
Gm: {group: chararray,k: {(A::State: chararray,A::Population: int,B::Port_name: chararray,B::Date: chararray,B::container: chararray,B::Nos_People: int)})}
grunt> DUMP Gm;
```

```
(1/1/18 0:00,{{(California,39776830,San Ysidro,1/1/18 0:00,Buses,2663),(California,39776830,Otay Mesa,1/1/18 0:00,Buses,2173),(California,39776830,Calexico East,1/1/18 0:00,Buses,216))}}(1/1/19 0:00,{{(California,39776830,San Ysidro,1/1/19 0:00,Buses,3110),(California,39776830,Otay Mesa,1/1/19 0:00,Buses,1351),(California,39776830,Calexico East,1/1/19 0:00,Buses,208))}}(2/1/18 0:00,{{(California,39776830,Calexico East,2/1/18 0:00,Buses,191),(California,39776830,San Ysidro,2/1/18 0:00,Buses,2395),(California,39776830,Otay Mesa,2/1/18 0:00,Buses,1727))}}(2/1/19 0:00,{{(California,39776830,Calexico East,2/1/19 0:00,Buses,137),(California,39776830,Otay Mesa,2/1/19 0:00,Buses,878),(California,39776830,San Ysidro,2/1/19 0:00,Buses,2952))}}(3/1/18 0:00,{{(California,39776830,Otay Mesa,3/1/18 0:00,Buses,990),(California,39776830,Calexico East,3/1/18 0:00,Buses,235),(California,39776830,San Ysidro,3/1/18 0:00,Buses,2749))}}(3/1/19 0:00,{{(California,39776830,Calexico East,3/1/19 0:00,Buses,105),(California,39776830,Otay Mesa,3/1/19 0:00,Buses,916),(California,39776830,San Ysidro,3/1/19 0:00,Buses,3329))}}(4/1/18 0:00,{{(California,39776830,Calexico East,4/1/18 0:00,Buses,220),(California,39776830,San Ysidro,4/1/18 0:00,Buses,2732),(California,39776830,Otay Mesa,4/1/18 0:00,Buses,2244))}}(5/1/18 0:00,{{(California,39776830,Calexico East,5/1/18 0:00,Buses,234),(California,39776830,San Ysidro,5/1/18 0:00,Buses,2651),(California,39776830,Otay Mesa,5/1/18 0:00,Buses,2148))}}(6/1/18 0:00,{{(California,39776830,Otay Mesa,6/1/18 0:00,Buses,2521),(California,39776830,San Ysidro,6/1/18 0:00,Buses,2748),(California,39776830,Calexico East,6/1/18 0:00,Buses,206))}}(7/1/17 0:00,{{(California,39776830,San Ysidro,7/1/17 0:00,Buses,2968),(California,39776830,Calexico East,7/1/17 0:00,Buses,238))}}(7/1/18 0:00,{{(California,39776830,Calexico East,7/1/18 0:00,Buses,212),(California,39776830,Otay Mesa,7/1/18 0:00,Buses,3239),(California,39776830,San Ysidro,7/1/18 0:00,Buses,2632))}}(8/1/17 0:00,{{(California,39776830,Calexico East,8/1/17 0:00,Buses,221),(California,39776830,San Ysidro,8/1/17 0:00,Buses,2838),(California,39776830,Otay Mesa,8/1/17 0:00,Buses,2502))}}(8/1/18 0:00,{{(California,39776830,San Ysidro,8/1/18 0:00,Buses,2590),(California,39776830,Calexico East,8/1/18 0:00,Buses,223),(California,39776830,Otay Mesa,8/1/18 0:00,Buses,3249))}}(9/1/17 0:00,{{(California,39776830,San Ysidro,9/1/17 0:00,Buses,2911),(California,39776830,Otay Mesa,9/1/17 0:00,Buses,2241),(California,39776830,Calexico East,9/1/17 0:00,Buses,271))}}(9/1/18 0:00,{{(California,39776830,Calexico East,9/1/18 0:00,Buses,211),(California,39776830,San Ysidro,9/1/18 0:00,Buses,2531),(California,39776830,Otay Mesa,9/1/18 0:00,Buses,2929))}}(10/1/17 0:00,{{(California,39776830,Calexico East,10/1/17 0:00,Buses,308),(California,39776830,San Ysidro,10/1/17 0:00,Buses,2890),(California,39776830,Otay Mesa,10/1/17 0:00,Buses,2690))}}(10/1/18 0:00,{{(California,39776830,Otay Mesa,10/1/18 0:00,Buses,3422),(California,39776830,San Ysidro,10/1/18 0:00,Buses,3076),(California,39776830,Calexico East,10/1/18 0:00,Buses,265))}}(11/1/17 0:00,{{(California,39776830,Calexico East,11/1/17 0:00,Buses,223),(California,39776830,Otay Mesa,11/1/17 0:00,Buses,2458),(California,39776830,San Ysidro,11/1/17 0:00,Buses,2788))}}(11/1/18 0:00,{{(California,39776830,San Ysidro,11/1/18 0:00,Buses,2296),(California,39776830,Otay Mesa,11/1/18 0:00,Buses,2802),(California,39776830,Calexico East,11/1/18 0:00,Buses,196))}}(12/1/17 0:00,{{(California,39776830,Otay Mesa,12/1/17 0:00,Buses,2298),(California,39776830,Calexico East,12/1/17 0:00,Buses,228),(California,39776830,San Ysidro,12/1/17 0:00,Buses,2794))}}(12/1/18 0:00,{{(California,39776830,Otay Mesa,12/1/18 0:00,Buses,2089),(California,39776830,Calexico East,12/1/18 0:00,Buses,193),(California,39776830,San Ysidro,12/1/18 0:00,Buses,2995))}}grunt> illustrate Gm;
```

## Part 9: Store Procedure

Firstly creating table named :customer with Attributes as ID, Name, AccountNo and Branch

Command:

Command: SQL>Select\*from customer;

| ID | NAME   | ACCOUNTNO | AGE | BRANCH     |
|----|--------|-----------|-----|------------|
| 1  | John   | 1212      | 52  | Chico      |
| 2  | Helen  | 1235      | 32  | SanJose    |
| 3  | Drek   | 4531      | 25  | Sanfransco |
| 4  | Martin | 3331      | 27  | Longbeach  |
| 5  | Megan  | 1111      | 20  | Fullerton  |

Command: SQL>Describe customer;

| Name      | Null? | Type         |
|-----------|-------|--------------|
| ID        |       | NUMBER(38)   |
| NAME      |       | VARCHAR2(10) |
| ACCOUNTNO |       | NUMBER(38)   |
| AGE       |       | NUMBER(38)   |
| BRANCH    |       | VARCHAR2(20) |

- Setting v\_id = ID, v\_name = Name, v\_acc = AccountNo, v\_age = v\_age and v\_branch = Branch to match coding parameters with that of table attributes.
- Modifying given code, to get **Branch** of Customer with **ID = 1**.
- Using **While loop** to get **Age** of all the Customer and using **If condition** to get **Name** of the Customer whose AccountNo is greater than 1000
- I had a lot of trouble with the procedure shows compiler error which was due to wrong syntax being used.

## Main Code:

## Code output:

```
[Enter user-name: mansi
[Enter password:
Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
[SQL> @Code.sql
Procedure created.

[SQL> set serveroutput on
[SQL> call mygame();
Branch : Chico
Age :52
Age :32
Age :25
Age :27
Age :20
Name:Megan
Call completed.
```

## Part 10: PL/SQL Cursor

### Implicit Cursor example 1:

```
[SQL> declare
[ 2  cnt number(3);
[ 3  begin
[ 4  update customer set AccountNo = 0000 where age >50;
[ 5  cnt := SQL%Rowcount;
[ 6  dbms_output.put_line(cnt || 'Successfully row updation');
[ 7  end;
[ 8 /
1Successfully row updation
PL/SQL procedure successfully completed.

[SQL> select * from customer
[ 2 ;


| ID | NAME   | ACCOUNTNO | AGE | BRANCH     |
|----|--------|-----------|-----|------------|
| 1  | John   | 0         | 52  | Chico      |
| 2  | Helen  | 1235      | 32  | SanJose    |
| 3  | Drek   | 4531      | 25  | Sanfransco |
| 4  | Martin | 3331      | 27  | Longbeach  |
| 5  | Megan  | 1111      | 20  | Fullerton  |
| 6  | Chris  | 2121      | 23  | Sandiego   |
| 7  | Danny  | 2000      | 21  | Oroville   |
| 8  | Nicole | 2001      | 22  | Freemont   |


8 rows selected.
```

- In this Code we use implicit cursor **SQL%Rowcount** to get count of the rows that are inserted.
- The AccountNo of the customer is set to zero when Age of the customer is greater than 50.
- And cnt is used to get affected rows using **SQL%Rowcount**.
- Hence in the output John AccountNo now equals zero.

### Implicit Cursor example 2 :

```
[SQL> select * from customer
[ 2 ;

      ID NAME      ACCOUNTNO      AGE BRANCH
-----  -----
    1 John          0            52 Chico
    2 Helen        1235         32 SanJose
    3 Drek         4531         25 Sanfransco
    4 Martin       3331         27 Longbeach
    5 Megan        1111         20 Fullerton
    6 Chris         2121         23 Sandiego
    7 Danny         2000         21 Oroville
    8 Nicole       2001         22 Freemont

8 rows selected.

[SQL> declare
[ 2   cnt number(3);
[ 3   begin
[ 4     update customer set Branch = '-----' where AccountNo = 0;
[ 5   cnt := SQL%Rowcount;
[ 6   dbms_output.put_line(cnt || ': Succesfully row updation');
[ 7   end;
[ 8 /
1: Succesfully row updation

PL/SQL procedure successfully completed.

[SQL> select * from customer
[ 2 ;

      ID NAME      ACCOUNTNO      AGE BRANCH
-----  -----
    1 John          0            52 -----
    2 Helen        1235         32 SanJose
    3 Drek         4531         25 Sanfransco
    4 Martin       3331         27 Longbeach
    5 Megan        1111         20 Fullerton
    6 Chris         2121         23 Sandiego
    7 Danny         2000         21 Oroville
    8 Nicole       2001         22 Freemont

8 rows selected.
```

- In this Code also we use implicit cursor **SQL%Rowcount** to get the count of the rows that are inserted.
- The Branch of the customer is set to '----' when AccountNo of the customer is zero or NULL. And cnt is used to get affected rows using **SQL%Rowcount**.
- Hence, in output John Branch now equals to Null or '-----'.

### Explicit Cursor 1:

The Customer table on which we use the explicit cursor is as follows,

```
[SQL> select * from customer
[ 2 ;
```

| ID | NAME   | ACCOUNTNO | AGE | BRANCH     |
|----|--------|-----------|-----|------------|
| 1  | John   | 0         | 52  | -----      |
| 2  | Helen  | 1235      | 32  | SanJose    |
| 3  | Drek   | 4531      | 25  | Sanfransco |
| 4  | Martin | 3331      | 27  | Longbeach  |
| 5  | Megan  | 1111      | 20  | Fullerton  |
| 6  | Chris  | 2121      | 23  | Sandiego   |
| 7  | Danny  | 2000      | 21  | Oroville   |
| 8  | Nicole | 2001      | 22  | Freemont   |

```
8 rows selected.
```

```
[SQL> describe customer
```

| Name      | Null? | Type         |
|-----------|-------|--------------|
| ID        |       | NUMBER(38)   |
| NAME      |       | VARCHAR2(10) |
| ACCOUNTNO |       | NUMBER(38)   |
| AGE       |       | NUMBER(38)   |
| BRANCH    |       | VARCHAR2(20) |

- In this code we use an explicit cursor named **tbl\_cursor%Rowcount** and **tbl\_cursor% Notfound**.
- We set v\_id = customer.ID, v\_name = customer.Name, v\_branch = customer.Branch to match table attributes with these variables.
- We are using one **tbl\_cursor** to fetch data from the customer table like ID,name and Branch of customers.
- And the other **tbl\_cursor** to run the loop till count greater than 8. Hence, from that loop we obtain output with ID, Name and Branch of all the customers.

```

SQL> Declare
 2 v_id customer.ID%type;
 3 v_name customer.Name%type;
 4 v_branch customer.Branch%type;
 5
 6 Cursor tbl_cursor IS
 7 Select ID,Name,Branch
 8 From customer;
 9
10 begin
11 open tbl_cursor;
12 LOOP
13   Fetch tbl_cursor into v_id, v_name, v_branch;
14   Exit when tbl_cursor%ROWCOUNT > 8 or tbl_cursor%NOTFOUND;
15   dbms_output.put_line(' '||v_id || ' : ' || v_name || ' : ' || v_branch);
16 END LOOP;
17 CLOSE tbl_cursor;
18 END;
19 /
1 : John : -----
2 : Helen : SanJose
3 : Drek : Sanfransco
4 : Martin : Longbeach
5 : Megan : Fullerton
6 : Chris : Sandiego
7 : Danny : Oroville
8 : Nicole : Freemont

```

PL/SQL procedure successfully completed.

## Part 11. PL/SQL Trigger

The datatable used for trigger is customer describe as follow,

```

SQL> select * from customer;

      ID NAME      ACCOUNTNO      AGE BRANCH
----- -----  -----
    1 John        1212        52 Chico
    2 Helen       1235        32 SanJose
    3 Drek        4531        25 Sanfransco
    4 Martin      3331        27 Longbeach
    5 Megan       1111        20 Fullerton
    6 Chris        2121        23 Sandiego
    7 Danny        2000        21 Oroville

```

7 rows selected.

```

SQL> describe customer;
 Name          Null?    Type
-----  -----
ID           NUMBER(38)
NAME         VARCHAR2(10)
ACCOUNTNO    NUMBER(38)
AGE          NUMBER(38)
BRANCH       VARCHAR2(20)

```

## Trigger1:

```
[SQL> create or replace trigger customer_insert
[ 2 after insert on customer
[ 3 begin
[ 4 dbms_output.put_line('Successfully row is Inserted in to table Customer.');
[ 5 end;
[ 6 /

Trigger created.

[SQL> set serveroutput on;
[SQL> insert into customer
[ 2 values (8,'Nicole',2001,22,'Freemont');
Successfully row is Inserted in to table Customer.

1 row created.
```

- Here we use trigger name **customer\_trigger**.
- The trigger get activated when a row is inserted into the table
- The trigger displays 'Successfully row is inserted into table customer,'
- The output of the table show newly inserted row ID = 8.

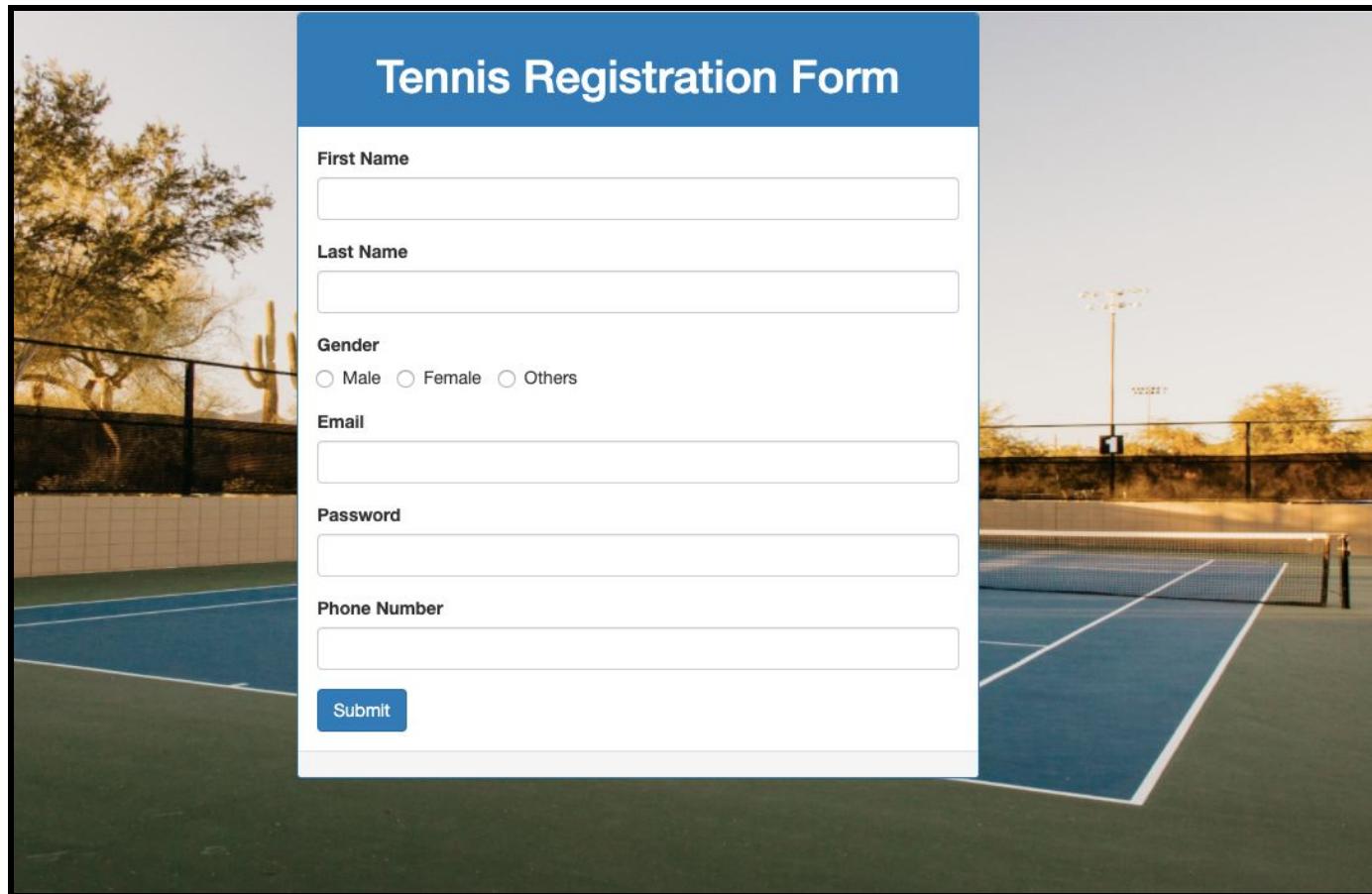
```
[SQL> select * from customer;

ID NAME      ACCOUNTNO     AGE BRANCH
----- -----  -----
 1 John        1212        52 Chico
 2 Helen       1235        32 SanJose
 3 Drek        4531        25 Sanfransco
 4 Martin      3331        27 Longbeach
 5 Megan       1111        20 Fullerton
 6 Chris        2121        23 Sandiego
 7 Danny        2000        21 Oroville
 8 Nicole      2001        22 Freemont

8 rows selected.
```

## Part 12: Live User Interface

User Interface:



The image shows a "Tennis Registration Form" window centered over a photograph of a tennis court at sunset. The form has a blue header with the title "Tennis Registration Form". It contains fields for "First Name" (with a text input box), "Last Name" (with a text input box), "Gender" (with three radio button options: Male, Female, Others), "Email" (with a text input box), "Password" (with a text input box), and "Phone Number" (with a text input box). A "Submit" button is located at the bottom left of the form. The background photograph shows a blue tennis court with white lines, a net, and a tall light pole. In the distance, there are trees and a fence under a clear sky.

### Tennis Registration Form

First Name

Last Name

Gender

Male  Female  Others

Email

Password

Phone Number

Submit

*None of our group members knew how to use HTML but rather than skipping these parts we decided to give it a try and it turned out that the user interface part was the most fun out of all other parts. We used Xampp and overall we learned a little about HyperText Markup Language.*

HTML code for the above user interface:

```
<!DOCTYPE html>
<html>
<head>
    <title>Registration Page</title>
    <link rel="stylesheet" type="text/css" href="bootstrap.css" />
</head>
<body>
    <div class="container">
        <div class="row col-md-6 col-md-offset-3">
            <div class="panel panel-primary">
                <div class="panel-heading text-center">
                    <h1>Tennis Registration Form</h1>
                </div>
                <div class="panel-body">
                    <form action="connect.php" method="post">
                        <div class="form-group">
                            <label for="firstName">First Name</label>
                            <input
                                type="text"
                                class="form-control"
                                id="firstName"
                                name="firstName"
                            />
                            <label for="lastName">Last Name</label>
                            <input
                                type="text"
                                class="form-control"
                                id="lastName"
                                name="lastName"
                            />
                        </div>
                        <div class="form-group">
                            <label for="gender">Gender</label>
                            <div>
                                <label for="male" class="radio-inline">
                                    <input
                                        type="radio"
                                        name="gender"
                                        value="Male"
                                        id="male"
                                    />Male</label>
                                >
                                <label for="female" class="radio-inline">
                                    <input
                                        type="radio"
                                        name="gender"
                                        value="Female"
                                        id="female"
                                    />Female</label>
                                >
                                <label for="others" class="radio-inline">
                                    <input
                                        type="radio"
                                        name="gender"
                                        value="Other"
                                        id="others"
                                    />Others</label>
                                >
                            </div>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>
</body>
```

```

        | </form>
        | </div>
        | <div class="panel-footer text-right">
        | </div>
        | </div>
    </div>
<div class="toast" role="alert" aria-live="assertive" aria-atomic="true">
<div class="toast-header">
<strong class="mr-auto">Bootstrap</strong>
<small>11 mins ago</small>
<button type="button" class="ml-2 mb-1 close" data-dismiss="toast" aria-label="Close">
| <span aria-hidden="true">&times;</span>
</button>
</div>
<div class="toast-body">
</div>
</div>
</body>
</html>

```

## PHP CODE FOR THE ABOVE USER INTERFACE

### Connect.php

```

?php
$firstName = $_POST['firstName'];
$lastName = $_POST['lastName'];
$gender = $_POST['gender'];
$email = $_POST['email'];
$password = $_POST['password'];
$number = $_POST['number'];

$conn = new mysqli('localhost', 'root', '', 'test');

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "INSERT INTO registration ".
    "(firstName, lastName, gender, email, password, number) "."VALUES ".
    "('$firstName', '$lastName', '$gender', '$email', '$password', '$number')";

if (mysqli_query($conn, $sql)) {
    echo "New record created successfully".<br>;
} else {
    echo "Error: " . $sql . "" . mysqli_error($conn);
}

$sql2 = 'SELECT * FROM registration';
$result = mysqli_query($conn, $sql2);

```

```

if (mysqli_num_rows($result) > 0) {
    while($row = mysqli_fetch_assoc($result)) {
        echo "<br>";
        echo "FirstName: " . $row["firstName"] . "<br>";
        echo "LastName: " . $row["lastName"] . "<br>";
        echo "Gender: " . $row["gender"] . "<br>";
        echo "Email: " . $row["email"] . "<br>";
        echo "Password: " . $row["password"] . "<br>";
        echo "PhoneNumber: " . $row["number"] . "<br>";

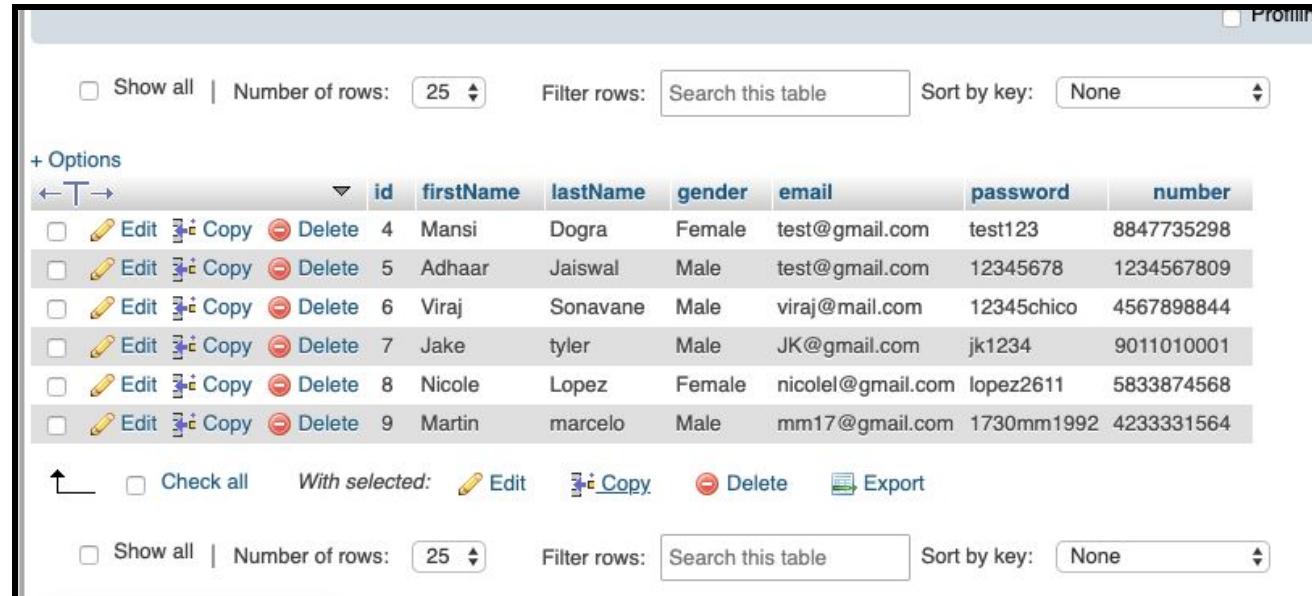
    }
} else {
    echo "0 results";
}

$conn->close();
?>

```

## MYSQL

The reflected database output for when Submit button is being pressed each time:



A screenshot of a MySQL database table interface. The table has columns: id, firstName, lastName, gender, email, password, and number. The data is as follows:

|                          | id | firstName | lastName | gender | email             | password   | number     |
|--------------------------|----|-----------|----------|--------|-------------------|------------|------------|
| <input type="checkbox"/> | 4  | Mansi     | Dogra    | Female | test@gmail.com    | test123    | 8847735298 |
| <input type="checkbox"/> | 5  | Adhaar    | Jaiswal  | Male   | test@gmail.com    | 12345678   | 1234567809 |
| <input type="checkbox"/> | 6  | Viraj     | Sonavane | Male   | viraj@mail.com    | 12345chico | 4567898844 |
| <input type="checkbox"/> | 7  | Jake      | tyler    | Male   | JK@gmail.com      | jk1234     | 9011010001 |
| <input type="checkbox"/> | 8  | Nicole    | Lopez    | Female | nicolet@gmail.com | lopez2611  | 5833874568 |
| <input type="checkbox"/> | 9  | Martin    | marcelo  | Male   | mm17@gmail.com    | 1730mm1992 | 4233331564 |

At the bottom, there are buttons for 'Check all', 'With selected:', 'Edit', 'Copy', 'Delete', and 'Export'. There are also links to 'Show all' and 'Number of rows' (set to 25), and search/filter options.

## Structure of database:

Describe table test:

The screenshot shows the phpMyAdmin interface for the 'test' database. The left sidebar lists databases: information\_schema, mysql, performance\_schema, phpmyadmin, test, and a New folder containing registration. The registration table is selected in the 'Table structure' tab. The table has 7 columns: id, firstName, lastName, gender, email, password, and number. The 'id' column is defined as int(11) with AUTO\_INCREMENT, while others are varchar(50). The 'gender' column uses enum('Male', 'Female', 'Others'). The 'password' column uses utf8mb4\_general\_ci collation. The 'number' column is bigint(10). Below the table structure, there's an 'Indexes' section showing a primary key named 'PRIMARY' using BTREE type. At the bottom, there are buttons for Add, Print, Propose table structure, Track table, Move columns, Normalize, and a Go button.

| # | Name      | Type                             | Collation          | Attributes | Null | Default | Comments | Extra          | Action             |
|---|-----------|----------------------------------|--------------------|------------|------|---------|----------|----------------|--------------------|
| 1 | id        | int(11)                          |                    |            | No   | None    |          | AUTO_INCREMENT | Change  Drop  More |
| 2 | firstName | varchar(50)                      | utf8mb4_general_ci |            | No   | None    |          |                | Change  Drop  More |
| 3 | lastName  | varchar(50)                      | utf8mb4_general_ci |            | No   | None    |          |                | Change  Drop  More |
| 4 | gender    | enum('Male', 'Female', 'Others') | utf8mb4_general_ci |            | No   | None    |          |                | Change  Drop  More |
| 5 | email     | varchar(50)                      | utf8mb4_general_ci |            | No   | None    |          |                | Change  Drop  More |
| 6 | password  | varchar(20)                      | utf8mb4_general_ci |            | No   | None    |          |                | Change  Drop  More |
| 7 | number    | bigint(10)                       |                    |            | No   | None    |          |                | Change  Drop  More |

| Action     | Keyname | Type  | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|------------|---------|-------|--------|--------|--------|-------------|-----------|------|---------|
| Edit  Drop | PRIMARY | BTREE | Yes    | No     | id     | 6           | A         | No   |         |

## *OUTPUT AFTER FILLING IN THE USER INTERFACE FORM AND USING BUTTONS*

FirstName: Mansi  
LastName: Dogra  
Gender: Female  
Email: test@gmail.com  
Password: test123  
PhoneNumber: 8847735298

FirstName: Adhaar  
LastName: Jaiswal  
Gender: Male  
Email: test@gmail.com  
Password: 12345678  
PhoneNumber: 1234567809

FirstName: Viraj  
LastName: Sonavane  
Gender: Male  
Email: viraj@mail.com  
Password: 12345chico  
PhoneNumber: 4567898844

FirstName: Jake  
LastName: tyler  
Gender: Male  
Email: JK@gmail.com  
Password: jk1234  
PhoneNumber: 9011010001

FirstName: Nicole  
LastName: Lopez  
Gender: Female  
Email: nicolel@gmail.com  
Password: lopez2611  
PhoneNumber: 5833874568

FirstName: Martin  
LastName: marcelo  
Gender: Male  
Email: mm17@gmail.com  
Password: 1730mm1992  
PhoneNumber: 4233331564

## Part 13. Combine Tools

We made a connection to Oracle Aws Instance using Python. At first we connected python to oracle using cx\_Oracle module. We have also shown the columns country\_code and Country from the income table in the output.

*None of our group members knew this either and we had spent the longest time on zoom meetings to figure out how to make connections as it timed out on us more than 20 times at least. Finally we got it after 2 days of struggle, we are very glad that we attempted all 13 parts of this project.*

```
ubuntu@ip-172-31-90-165:~$ python
Python 2.7.17 (default, Nov  7 2019, 10:07:09)
[GCC 7.4.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import cx_Oracle
>>> dsn_tns = cx_Oracle.makedsn('ec2-54-163-87-221.compute-1.amazonaws.com', '1521', 'XE')
>>> conn = cx_Oracle.connect(user='adhaar',password='AdhaarChico17', dsn=dsn_tns)
>>> c.execute('select count(*) from income')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'c' is not defined
>>> c = conn.cursor()
>>> c.execute('select count(*) from income')
<cx_Oracle.Cursor on <cx_Oracle.Connection to adhaar@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=ec2-54-163-87-221.compute-1.amazonaws.com)(PORT=1521))(CONNECT_DATA=(SID=XE)))>>
>>> for row in c:
...     print row[0], "--", row[1]
...
185 -
Traceback (most recent call last):
  File "<stdin>", line 2, in <module>
IndexError: tuple index out of range
>>> c = conn.cursor()
>>> c.execute('select country_code,country from income')
<cx_Oracle.Cursor on <cx_Oracle.Connection to adhaar@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=ec2-54-163-87-221.compute-1.amazonaws.com)(PORT=1521))(CONNECT_DATA=(SID=XE)))>>
>>> for row in c:
...     print row[0], "--", row[1]
... conn.close()
  File "<stdin>", line 3
    conn.close()
 ^
SyntaxError: invalid syntax
>>> c = conn.cursor()
>>> c.execute('select country_code,country from income')
<cx_Oracle.Cursor on <cx_Oracle.Connection to adhaar@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=ec2-54-163-87-221.compute-1.amazonaws.com)(PORT=1521))(CONNECT_DATA=(SID=XE)))>>
>>> for row in c:
...     print row
...
('NOR', 'Norway\r')
('NPL', 'Nepal\r')
('NRU', 'Nauru\r')
('NZL', 'New Zealand\r')
('PAK', 'Pakistan\r')
('PAN', 'Panama\r')
('PER', 'Peru\r')
('PHL', 'Philippines\r')
('PLW', 'Palau\r')
('PNG', 'Papua New Guinea\r')
('POL', 'Poland\r')
('PRI', 'Puerto Rico\r')
('PRK', 'Korea')
('prt', 'Portugal\r')
('PYR', 'Paraguay\r')
('PYF', 'French Polynesia\r')
('ROU', 'Romania\r')
('RUS', 'Russian Federation\r')
('RWA', 'Rwanda\r')
('SDN', 'Sudan\r')
('SEN', 'Senegal\r')
('SGP', 'Singapore\r')
('SLB', 'Solomon Islands\r')
('SLE', 'Sierra Leone\r')
('SLV', 'El Salvador\r')
('SMR', 'San Marino\r')
```

### GROUP MEMBERS

- Adhaar Jaiswal
- Viraj Sonavane
- Mansi Dogra