8.1.3 Searchable encryption

This section presents the demonstrator for the Searchable Encryption (SE) module. The demonstrator runs on a single machine and emulates both the CLARUS user and the CSP (that runs an instance of a PostgreSQL server). The communication between the user and the server is done by means of the SQL syntax.

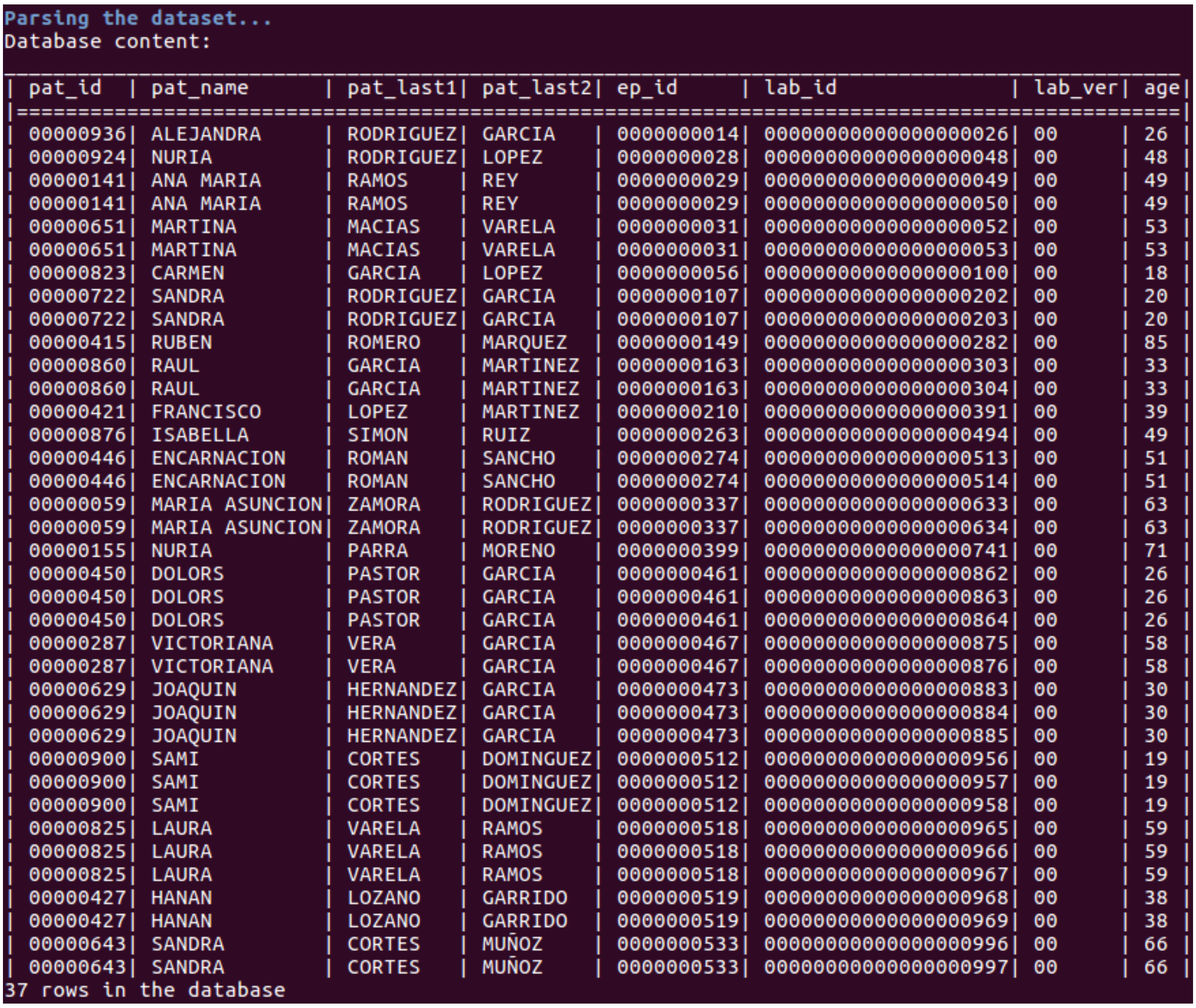
8.1.3.1 Storyboard

The CLARUS user would like to store a privacy-sensitive (SQL) database (such as the e-health database developed in the project) in an untrusted cloud server. To preserve data confidentiality and patients' privacy against this cloud server, the user will resort to the Searchable Encryption module, in order to be still able to search keywords over the protected database.

In this scenario, the user will first enter the Storage phase of the Searchable Encryption procedure (see section 4.4). This phase generates the key material for the encryption of the data content and the construction of a secure search index that will enable the cloud to search for some keywords, on behalf of the CLARUS user, without knowing which keywords are searched. In this phase, the user also defines the attributes that can be subject to range queries. At a later point, the user will generate an SQL search query (SELECT query) that will be transformed by the Searchable encryption module into an "encrypted SQL query" (noted SE query) that hides information about the search criteria. The cloud server executes this SE query and sends back to the user the encrypted search results, that is, the records in the database that match the search criteria. Finally, the user invokes the decryption mechanism of the SE module to retrieve the decrypted search results.

8.1.3.2 Step by step demo

Figure 65 shows the original e-health database, that has been reduced to a single table and a couple of records for the demonstrator purposes. This database, named lab_simple, consists of 8 attributes: pat_id, pat_name, pat_last1, pat_last2, ep_id, lab_ver and age. The CLARUS user would like to search records that present particular value of pat_name, pat_last2 and age. For example, he/she would like to search for the records of patients whose pat_name is SANDRA or whose pat_last2 is GARCIA etc.



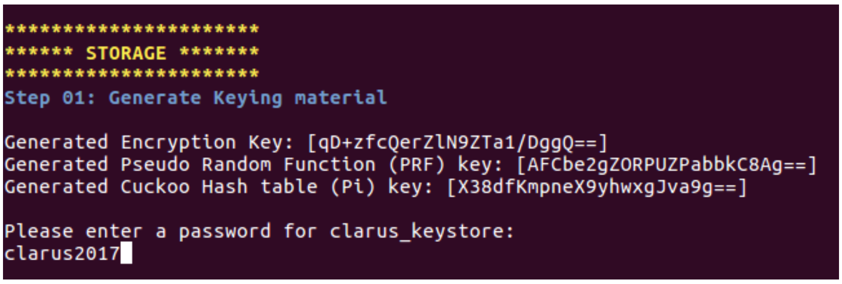
```
Parsing the dataset...
Database content:
_________________________________________________________________________________________________
| pat_id  | pat_name       | pat_last1| pat_last2| ep_id     | lab_id              | lab_ver| age|
|===============================================================================================|
| 00000936| ALEJANDRA      | RODRIGUEZ| GARCIA   | 0000000014| 00000000000000000026| 00     | 26 |
| 00000924| NURIA          | RODRIGUEZ| LOPEZ    | 0000000028| 00000000000000000048| 00     | 48 |
| 00000141| ANA MARIA      | RAMOS    | REY      | 0000000029| 00000000000000000049| 00     | 49 |
| 00000141| ANA MARIA      | RAMOS    | REY      | 0000000029| 00000000000000000050| 00     | 49 |
| 00000651| MARTINA        | MACIAS   | VARELA   | 0000000031| 00000000000000000052| 00     | 53 |
| 00000651| MARTINA        | MACIAS   | VARELA   | 0000000031| 00000000000000000053| 00     | 53 |
| 00000823| CARMEN         | GARCIA   | LOPEZ    | 0000000056| 00000000000000000100| 00     | 18 |
| 00000722| SANDRA         | RODRIGUEZ| GARCIA   | 0000000107| 00000000000000000202| 00     | 20 |
| 00000722| SANDRA         | RODRIGUEZ| GARCIA   | 0000000107| 00000000000000000203| 00     | 20 |
| 00000415| RUBEN          | ROMERO   | MARQUEZ  | 0000000149| 00000000000000000282| 00     | 85 |
| 00000860| RAUL           | GARCIA   | MARTINEZ | 0000000163| 00000000000000000303| 00     | 33 |
| 00000860| RAUL           | GARCIA   | MARTINEZ | 0000000163| 00000000000000000304| 00     | 33 |
| 00000421| FRANCISCO      | LOPEZ    | MARTINEZ | 0000000210| 00000000000000000391| 00     | 39 |
| 00000876| ISABELLA       | SIMON    | RUIZ     | 0000000263| 00000000000000000494| 00     | 49 |
| 00000446| ENCARNACION    | ROMAN    | SANCHO   | 0000000274| 00000000000000000513| 00     | 51 |
| 00000446| ENCARNACION    | ROMAN    | SANCHO   | 0000000274| 00000000000000000514| 00     | 51 |
| 00000059| MARIA ASUNCION| ZAMORA   | RODRIGUEZ| 0000000337| 00000000000000000633| 00     | 63 |
| 00000059| MARIA ASUNCION| ZAMORA   | RODRIGUEZ| 0000000337| 00000000000000000634| 00     | 63 |
| 00000155| NURIA          | PARRA    | MORENO   | 0000000399| 00000000000000000741| 00     | 71 |
| 00000450| DOLORS         | PASTOR   | GARCIA   | 0000000461| 00000000000000000862| 00     | 26 |
| 00000450| DOLORS         | PASTOR   | GARCIA   | 0000000461| 00000000000000000863| 00     | 26 |
| 00000450| DOLORS         | PASTOR   | GARCIA   | 0000000461| 00000000000000000864| 00     | 26 |
| 00000287| VICTORIANA     | VERA     | GARCIA   | 0000000467| 00000000000000000875| 00     | 58 |
| 00000287| VICTORIANA     | VERA     | GARCIA   | 0000000467| 00000000000000000876| 00     | 58 |
| 00000629| JOAQUIN        | HERNANDEZ| GARCIA   | 0000000473| 00000000000000000883| 00     | 30 |
| 00000629| JOAQUIN        | HERNANDEZ| GARCIA   | 0000000473| 00000000000000000884| 00     | 30 |
| 00000629| JOAQUIN        | HERNANDEZ| GARCIA   | 0000000473| 00000000000000000885| 00     | 30 |
| 00000900| SAMI           | CORTES   | DOMINGUEZ| 0000000512| 00000000000000000956| 00     | 19 |
| 00000900| SAMI           | CORTES   | DOMINGUEZ| 0000000512| 00000000000000000957| 00     | 19 |
| 00000900| SAMI           | CORTES   | DOMINGUEZ| 0000000512| 00000000000000000958| 00     | 19 |
| 00000825| LAURA          | VARELA   | RAMOS    | 0000000518| 00000000000000000965| 00     | 59 |
| 00000825| LAURA          | VARELA   | RAMOS    | 0000000518| 00000000000000000966| 00     | 59 |
| 00000825| LAURA          | VARELA   | RAMOS    | 0000000518| 00000000000000000967| 00     | 59 |
| 00000427| HANAN          | LOZANO   | GARRIDO  | 0000000519| 00000000000000000968| 00     | 38 |
| 00000427| HANAN          | LOZANO   | GARRIDO  | 0000000519| 00000000000000000969| 00     | 38 |
| 00000643| SANDRA         | CORTES   | MUÑOZ    | 0000000533| 00000000000000000996| 00     | 66 |
| 00000643| SANDRA         | CORTES   | MUÑOZ    | 0000000533| 00000000000000000997| 00     | 66 |
37 rows in the database
```

Figure 65. Original database

Storage phase

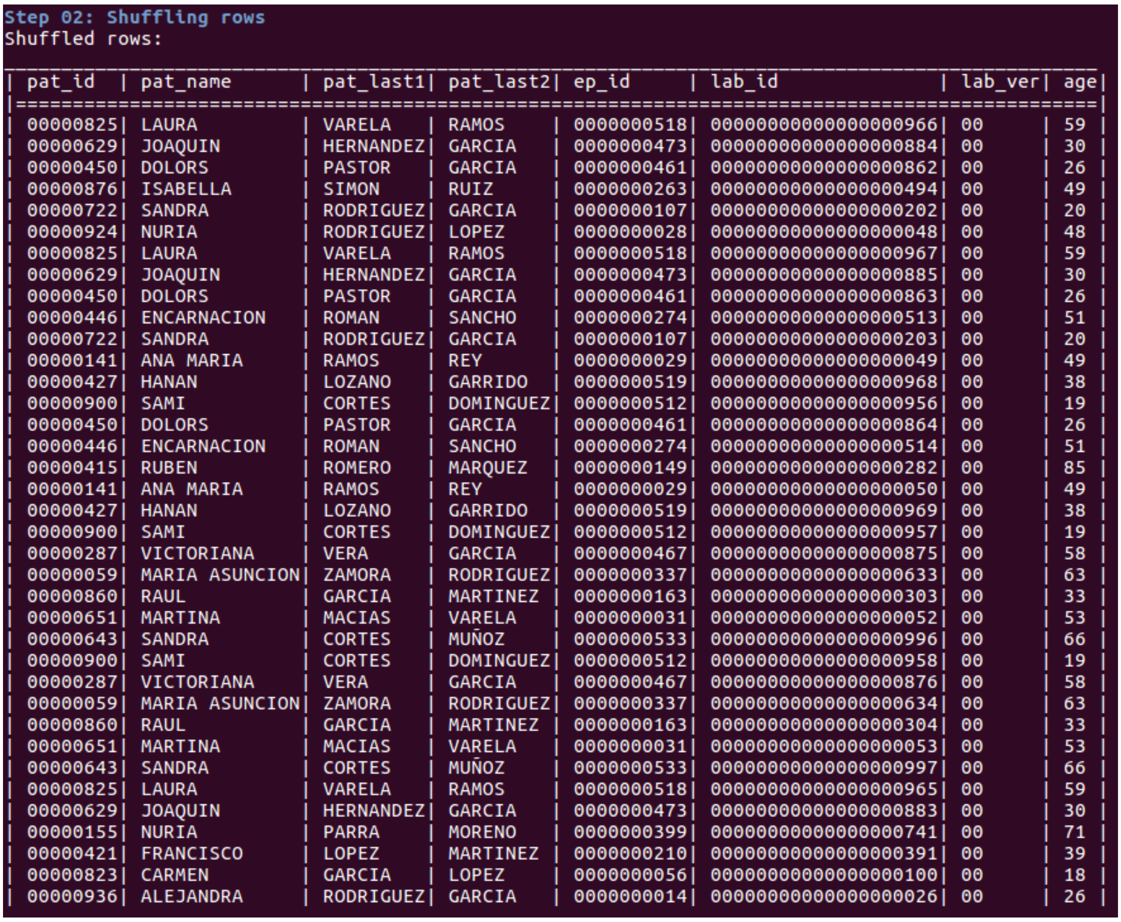
The first step of the SE module operations is the generation of the key material as shown in Figure 66. The module asks the user for a password that will be used to generate the keys ans to instantiate and access the keystore.

Figure 66 depicts the second step of the storage phase that consists in shuffling the records in the database. This step helps in obfuscating the ordering of the records in the original table. This permutation is invertible.



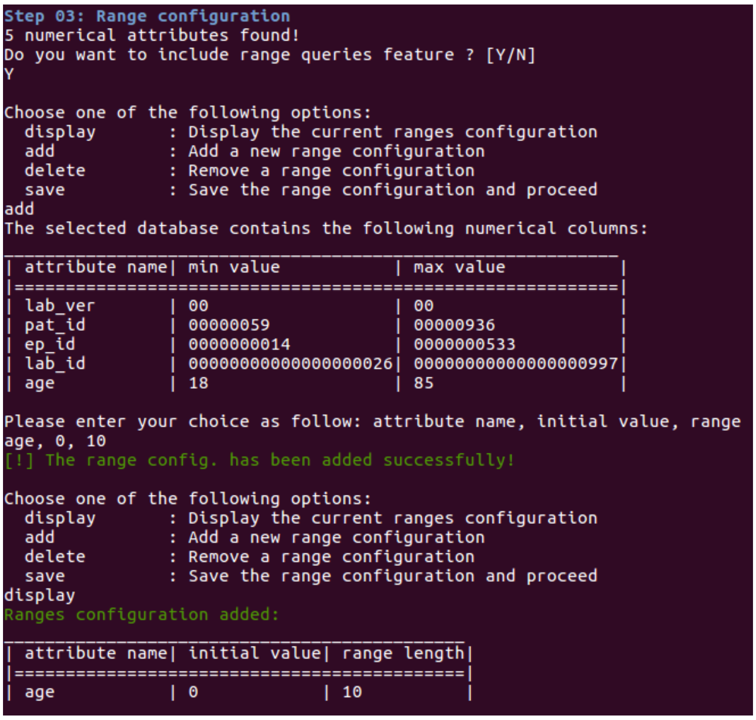
```
**********************
****** STORAGE *******
**********************
Step 01: Generate Keying material

Generated Encryption Key: [qD+zfcQerZlN9ZTa1/DggQ==]
Generated Pseudo Random Function (PRF) key: [AFCbe2gZORPUZPabbkC8Ag==]
Generated Cuckoo Hash table (Pi) key: [X38dfKmpneX9yhwxgJva9g==]

Please enter a password for clarus_keystore:
clarus2017
```

Figure 66. Key generation



```
Step 02: Shuffling rows
Shuffled rows:
________________________________________________________________________________________________
| pat_id  | pat_name      | pat_last1| pat_last2| ep_id     | lab_id             | lab_ver| age|
|===============================================================================================|
| 00000825| LAURA         | VARELA   | RAMOS    | 0000000518| 00000000000000000966| 00     | 59 |
| 00000629| JOAQUIN       | HERNANDEZ| GARCIA   | 0000000473| 00000000000000000884| 00     | 30 |
| 00000450| DOLORS        | PASTOR   | GARCIA   | 0000000461| 00000000000000000862| 00     | 26 |
| 00000876| ISABELLA      | SIMON    | RUIZ     | 0000000263| 00000000000000000494| 00     | 49 |
| 00000722| SANDRA        | RODRIGUEZ| GARCIA   | 0000000107| 00000000000000000202| 00     | 20 |
| 00000924| NURIA         | RODRIGUEZ| LOPEZ    | 0000000028| 00000000000000000048| 00     | 48 |
| 00000825| LAURA         | VARELA   | RAMOS    | 0000000518| 00000000000000000967| 00     | 59 |
| 00000629| JOAQUIN       | HERNANDEZ| GARCIA   | 0000000473| 00000000000000000885| 00     | 30 |
| 00000450| DOLORS        | PASTOR   | GARCIA   | 0000000461| 00000000000000000863| 00     | 26 |
| 00000446| ENCARNACION   | ROMAN    | SANCHO   | 0000000274| 00000000000000000513| 00     | 51 |
| 00000722| SANDRA        | RODRIGUEZ| GARCIA   | 0000000107| 00000000000000000203| 00     | 20 |
| 00000141| ANA MARIA     | RAMOS    | REY      | 0000000029| 00000000000000000049| 00     | 49 |
| 00000427| HANAN         | LOZANO   | GARRIDO  | 0000000519| 00000000000000000968| 00     | 38 |
| 00000900| SAMI          | CORTES   | DOMINGUEZ| 0000000512| 00000000000000000956| 00     | 19 |
| 00000450| DOLORS        | PASTOR   | GARCIA   | 0000000461| 00000000000000000864| 00     | 26 |
| 00000446| ENCARNACION   | ROMAN    | SANCHO   | 0000000274| 00000000000000000514| 00     | 51 |
| 00000415| RUBEN         | ROMERO   | MARQUEZ  | 0000000149| 00000000000000000282| 00     | 85 |
| 00000141| ANA MARIA     | RAMOS    | REY      | 0000000029| 00000000000000000050| 00     | 49 |
| 00000427| HANAN         | LOZANO   | GARRIDO  | 0000000519| 00000000000000000969| 00     | 38 |
| 00000900| SAMI          | CORTES   | DOMINGUEZ| 0000000512| 00000000000000000957| 00     | 19 |
| 00000287| VICTORIANA    | VERA     | GARCIA   | 0000000467| 00000000000000000875| 00     | 58 |
| 00000059| MARIA ASUNCION| ZAMORA   | RODRIGUEZ| 0000000337| 00000000000000000633| 00     | 63 |
| 00000860| RAUL          | GARCIA   | MARTINEZ | 0000000163| 00000000000000000303| 00     | 33 |
| 00000651| MARTINA       | MACIAS   | VARELA   | 0000000031| 00000000000000000052| 00     | 53 |
| 00000643| SANDRA        | CORTES   | MUÑOZ    | 0000000533| 00000000000000000996| 00     | 66 |
| 00000900| SAMI          | CORTES   | DOMINGUEZ| 0000000512| 00000000000000000958| 00     | 19 |
| 00000287| VICTORIANA    | VERA     | GARCIA   | 0000000467| 00000000000000000876| 00     | 58 |
| 00000059| MARIA ASUNCION| ZAMORA   | RODRIGUEZ| 0000000337| 00000000000000000634| 00     | 63 |
| 00000860| RAUL          | GARCIA   | MARTINEZ | 0000000163| 00000000000000000304| 00     | 33 |
| 00000651| MARTINA       | MACIAS   | VARELA   | 0000000031| 00000000000000000053| 00     | 53 |
| 00000643| SANDRA        | CORTES   | MUÑOZ    | 0000000533| 00000000000000000997| 00     | 66 |
| 00000825| LAURA         | VARELA   | RAMOS    | 0000000518| 00000000000000000965| 00     | 59 |
| 00000629| JOAQUIN       | HERNANDEZ| GARCIA   | 0000000473| 00000000000000000883| 00     | 30 |
| 00000155| NURIA         | PARRA    | MORENO   | 0000000399| 00000000000000000741| 00     | 71 |
| 00000421| FRANCISCO     | LOPEZ    | MARTINEZ | 0000000210| 00000000000000000391| 00     | 39 |
| 00000823| CARMEN        | GARCIA   | LOPEZ    | 0000000056| 00000000000000000100| 00     | 18 |
| 00000936| ALEJANDRA     | RODRIGUEZ| GARCIA   | 0000000014| 00000000000000000026| 00     | 26 |
```

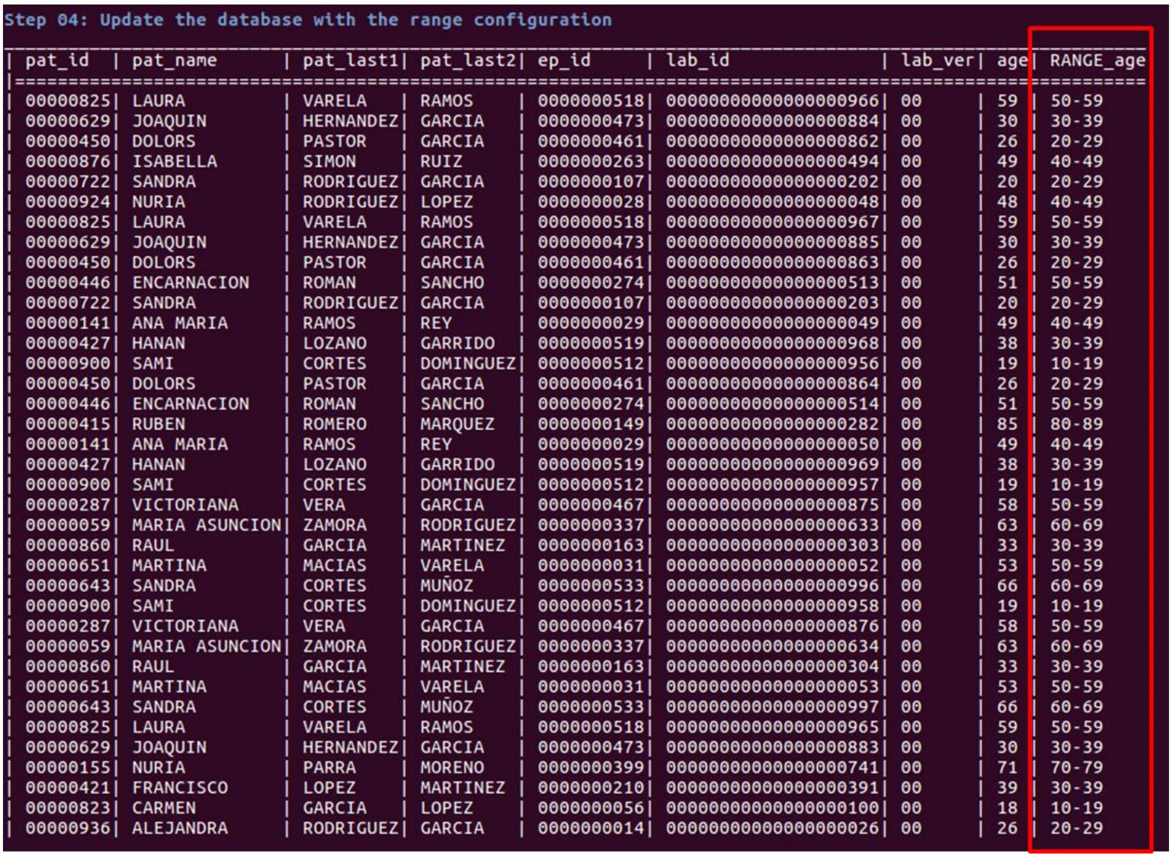
Figure 67. Shuffling records



```
Step 03: Range configuration
5 numerical attributes found!
Do you want to include range queries feature ? [Y/N]
Y

Choose one of the following options:
  display       : Display the current ranges configuration
  add           : Add a new range configuration
  delete        : Remove a range configuration
  save          : Save the range configuration and proceed
add
The selected database contains the following numerical columns:
_______________________________________________________________
| attribute name| min value             | max value             |
|=============================================================|
| lab_ver       | 00                    | 00                    |
| pat_id        | 00000059              | 00000936              |
| ep_id         | 0000000014            | 0000000533            |
| lab_id        | 000000000000000000026| 000000000000000000997|
| age           | 18                    | 85                    |

Please enter your choice as follow: attribute name, initial value, range
age, 0, 10
[!] The range config. has been added successfully!

Choose one of the following options:
  display       : Display the current ranges configuration
  add           : Add a new range configuration
  delete        : Remove a range configuration
  save          : Save the range configuration and proceed
display
Ranges configuration added:
______________________________________________
| attribute name| initial value| range length|
|============================================|
| age           | 0            | 10           |
```

Figure 68. Range configuration

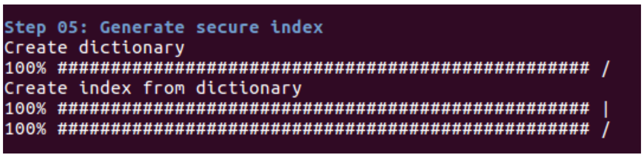
In the third step of the storage phase operated by the SE module, the user is requested to configure the columns that contain numerical attributes, such as the patients' age. As seen in Figure 68, the module automatically detects these numerical attributes and displays to the user their names, minimum and maximum values. If needed, the user specifies which attributes will be searched by means of a range query of the form attribute>=value and attribute<=value. Note that the operator can also be "<" or ">". Furthermore, the user defines the range and the initial value from which the range intervals are computed. For instance, as shown in Figure 68, the user specifies a range of 10 for the attribute age, starting from the initial value 0. This means that all the values of the attribute age in the original database will be distributed in range intervals of length 10 of the form [0-9], [10, 19], [20, 29], etc. Figure 68 shows the resulting database, in which the SE module adds the extra column "RANGE_age" that specifies for each record the range of the corresponding attribute age As specified in D3.2, this new attribute will help to transform the range query problem into a simple keyword search problem: searching for age between 20 and 30 is reduced to searching for keywords RANGE_age='20-29' and age='30'.



```
Step 04: Update the database with the range configuration

-----------------------------------------------------------------------------------------------------------
| pat_id  | pat_name      | pat_last1| pat_last2| ep_id     | lab_id              | lab_ver| age| RANGE_age
|==========================================================================================================
| 00000825| LAURA         | VARELA   | RAMOS    | 0000000518| 00000000000000000966| 00     | 59 | 50-59
| 00000629| JOAQUIN       | HERNANDEZ| GARCIA   | 0000000473| 00000000000000000884| 00     | 30 | 30-39
| 00000450| DOLORS        | PASTOR   | GARCIA   | 0000000461| 00000000000000000862| 00     | 26 | 20-29
| 00000876| ISABELLA      | SIMON    | RUIZ     | 0000000263| 00000000000000000494| 00     | 49 | 40-49
| 00000722| SANDRA        | RODRIGUEZ| GARCIA   | 0000000107| 00000000000000000202| 00     | 20 | 20-29
| 00000924| NURIA         | RODRIGUEZ| LOPEZ    | 0000000028| 00000000000000000048| 00     | 48 | 40-49
| 00000825| LAURA         | VARELA   | RAMOS    | 0000000518| 00000000000000000967| 00     | 59 | 50-59
| 00000629| JOAQUIN       | HERNANDEZ| GARCIA   | 0000000473| 00000000000000000885| 00     | 30 | 30-39
| 00000450| DOLORS        | PASTOR   | GARCIA   | 0000000461| 00000000000000000863| 00     | 26 | 20-29
| 00000446| ENCARNACION   | ROMAN    | SANCHO   | 0000000274| 00000000000000000513| 00     | 51 | 50-59
| 00000722| SANDRA        | RODRIGUEZ| GARCIA   | 0000000107| 00000000000000000203| 00     | 20 | 20-29
| 00000141| ANA MARIA     | RAMOS    | REY      | 0000000029| 00000000000000000049| 00     | 49 | 40-49
| 00000427| HANAN         | LOZANO   | GARRIDO  | 0000000519| 00000000000000000968| 00     | 38 | 30-39
| 00000900| SAMI          | CORTES   | DOMINGUEZ| 0000000512| 00000000000000000956| 00     | 19 | 10-19
| 00000450| DOLORS        | PASTOR   | GARCIA   | 0000000461| 00000000000000000864| 00     | 26 | 20-29
| 00000446| ENCARNACION   | ROMAN    | SANCHO   | 0000000274| 00000000000000000514| 00     | 51 | 50-59
| 00000415| RUBEN         | ROMERO   | MARQUEZ  | 0000000149| 00000000000000000282| 00     | 85 | 80-89
| 00000141| ANA MARIA     | RAMOS    | REY      | 0000000029| 00000000000000000050| 00     | 49 | 40-49
| 00000427| HANAN         | LOZANO   | GARRIDO  | 0000000519| 00000000000000000969| 00     | 38 | 30-39
| 00000900| SAMI          | CORTES   | DOMINGUEZ| 0000000512| 00000000000000000957| 00     | 19 | 10-19
| 00000287| VICTORIANA    | VERA     | GARCIA   | 0000000467| 00000000000000000875| 00     | 58 | 50-59
| 00000059| MARIA ASUNCION| ZAMORA   | RODRIGUEZ| 0000000337| 00000000000000000633| 00     | 63 | 60-69
| 00000860| RAUL          | GARCIA   | MARTINEZ | 0000000163| 00000000000000000303| 00     | 33 | 30-39
| 00000651| MARTINA       | MACIAS   | VARELA   | 0000000031| 00000000000000000052| 00     | 53 | 50-59
| 00000643| SANDRA        | CORTES   | MUÑOZ    | 0000000533| 00000000000000000996| 00     | 66 | 60-69
| 00000900| SAMI          | CORTES   | DOMINGUEZ| 0000000512| 00000000000000000958| 00     | 19 | 10-19
| 00000287| VICTORIANA    | VERA     | GARCIA   | 0000000467| 00000000000000000876| 00     | 58 | 50-59
| 00000059| MARIA ASUNCION| ZAMORA   | RODRIGUEZ| 0000000337| 00000000000000000634| 00     | 63 | 60-69
| 00000860| RAUL          | GARCIA   | MARTINEZ | 0000000163| 00000000000000000304| 00     | 33 | 30-39
| 00000651| MARTINA       | MACIAS   | VARELA   | 0000000031| 00000000000000000053| 00     | 53 | 50-59
| 00000643| SANDRA        | CORTES   | MUÑOZ    | 0000000533| 00000000000000000997| 00     | 66 | 60-69
| 00000825| LAURA         | VARELA   | RAMOS    | 0000000518| 00000000000000000965| 00     | 59 | 50-59
| 00000629| JOAQUIN       | HERNANDEZ| GARCIA   | 0000000473| 00000000000000000883| 00     | 30 | 30-39
| 00000155| NURIA         | PARRA    | MORENO   | 0000000399| 00000000000000000741| 00     | 71 | 70-79
| 00000421| FRANCISCO     | LOPEZ    | MARTINEZ | 0000000210| 00000000000000000391| 00     | 39 | 30-39
| 00000823| CARMEN        | GARCIA   | LOPEZ    | 0000000056| 00000000000000000100| 00     | 18 | 10-19
| 00000936| ALEJANDRA     | RODRIGUEZ| GARCIA   | 0000000014| 00000000000000000026| 00     | 26 | 20-29
```

Figure 69. Augmented database for range queries

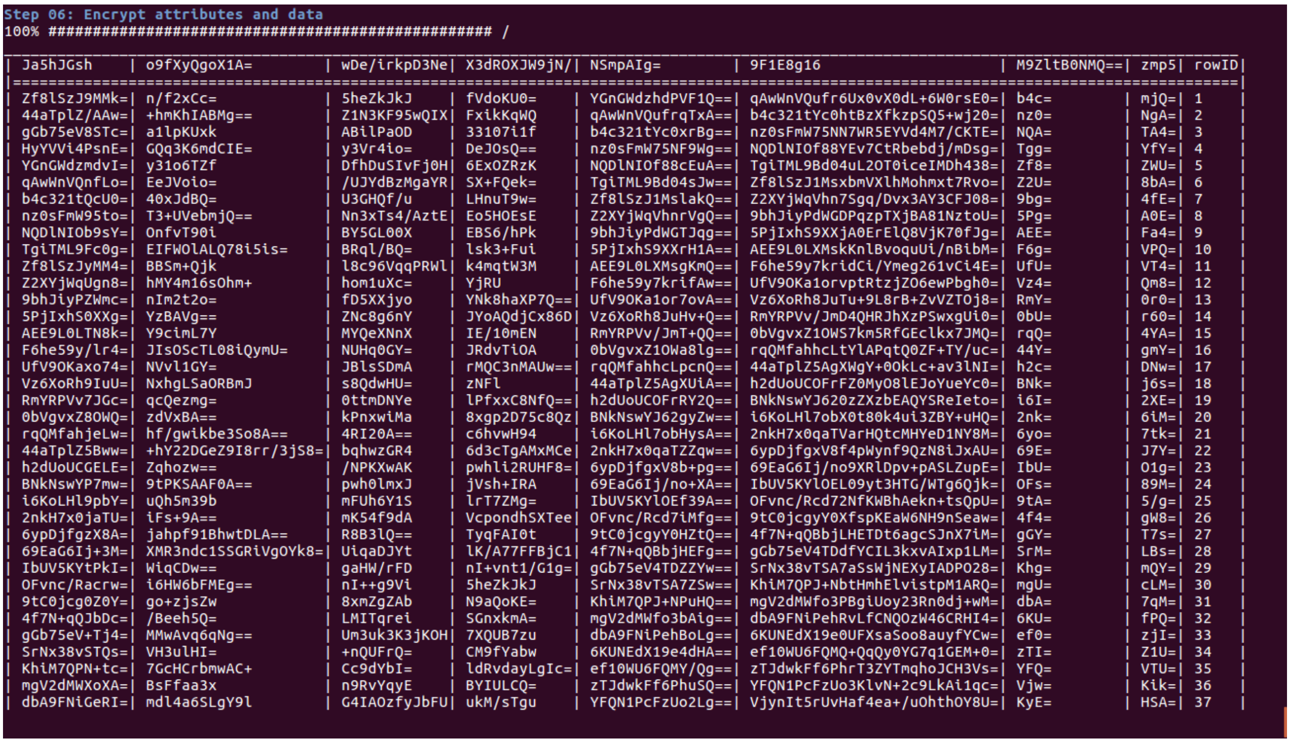
The fifth step in the storage phase (Figure 70) is the generation of the secure search index, as described in details in D3.2. The SE module first creates a dictionary of the distinct keywords in the database of the form attribute='value'. It then creates the index from this dictionary. The index is protected against the curious cloud server such that it does not infer which keywords are included in the index.



```
Step 05: Generate secure index
Create dictionary
100% ################################################## /
Create index from dictionary
100% ################################################## |
100% ################################################## /
```

Figure 70. Creation of the secure search index

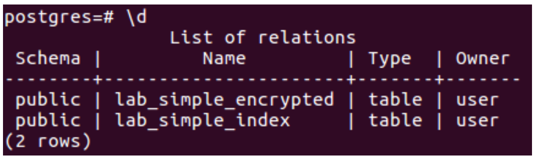
Finally, the last step of the SE module encrypts the database with a semantically secure symmetric encryption scheme, entry by entry, such that no entry yields the same ciphertext. The resulting encrypted database is illustrated in Figure 71. Note that for each record, the SE module also adds an extra column “RowID” that simply takes the order of the record in the encrypted database.



```
Step 06: Encrypt attributes and data
100% ################################################### /

___________________________________________________________________________________________________________________________________________
| Ja5hJGsh    | o9fXyQgoX1A=        | wDe/irkpD3Ne| X3dROXJW9jN/| NSmpAIg=        | 9F1E8g16                    | M9ZltB0NMQ==| zmp5| rowID|
|==========================================================================================================================================|
| Zf8lSzJ9MMk=| n/f2xCc=            | 5heZkJkJ    | fVdoKU0=    | YGnGWdzhdPVF1Q==| qAwWnVQufr6Ux0vX0dL+6W0rsE0=| b4c=        | mjQ=| 1    |
| 44aTplZ/AAw=| +hmKhIABMg==        | Z1N3KF95wQIX| FxikKqWQ    | qAwWnVQufrqTxA==| b4c321tYc0htBzXfkzpSQ5+wj20=| nz0=        | NgA=| 2    |
| gGb75eV8STc=| a1lpKUxk            | ABilPaOD    | 33107i1f    | b4c321tYc0xrBg==| nz0sFmW75NN7WR5EYVd4M7/CKTE=| NQA=        | TA4=| 3    |
| HyYVVi4PsnE=| GQq3K6mdCIE=        | y3Vr4io=    | DeJOsQ==    | nz0sFmW75NF9Wg==| NQDlNIOf88YEv7CtRbebdj/mDsg=| Tgg=        | YfY=| 4    |
| YGnGWdzmdvI=| y31o6TZf            | DfhDuSIvFj0H| 6ExOZRzK    | NQDlNIOf88cEuA==| TgiTML9Bd04uL2OT0iceIMDh438=| Zf8=        | ZWU=| 5    |
| qAwWnVQnfLo=| EeJVoio=            | /UJYdBzMgaYR| SX+FQek=    | TgiTML9Bd04sJw==| Zf8lSzJ1MsxbmVXlhMohmxt7Rvo=| Z2U=        | 8bA=| 6    |
| b4c321tQcU0=| 40xJdBQ=            | U3GHQf/u    | LHnuT9w=    | Zf8lSzJ1MslakQ==| Z2XYjWqVhn7Sgq/Dvx3AY3CFJ08=| 9bg=        | 4fE=| 7    |
| nz0sFmW95to=| T3+UVebmjQ==        | Nn3xTs4/AztE| Eo5HOEsE    | Z2XYjWqVhnrVgQ==| 9bhJiyPdWGDPqzpTXjBA81NztoU=| 5Pg=        | A0E=| 8    |
| NQDlNIOb9sY=| OnfvT90i            | BY5GL00X    | EBS6/hPk    | 9bhJiyPdWGTJqg==| 5PjIxhS9XXjA0ErElQ8VjK70fJg=| AEE=        | Fa4=| 9    |
| TgiTML9Fc0g=| EIFWOlALQ78i5is=    | BRql/BQ=    | lsk3+Fui    | 5PjIxhS9XXrH1A==| AEE9L0LXMskKnlBvoquUi/nBibM=| F6g=        | VPQ=| 10   |
| Zf8lSzJyMM4=| BBSm+Qjk            | l8c96VqqPRWl| k4mqtW3M    | AEE9L0LXMsgKmQ==| F6he59y7kridCi/Ymeg261vCi4E=| UfU=        | VT4=| 11   |
| Z2XYjWqUgn8=| hMY4m16sOhm+        | hom1uXc=    | YjRU        | F6he59y7krifAw==| UfV9OKa1orvptRtzjZO6ewPbgh0=| Vz4=        | Qm8=| 12   |
| 9bhJiyPZWmc=| nIm2t2o=            | fD5XXjyo    | YNk8haXP7Q==| UfV9OKa1or7ovA==| Vz6XoRh8JuTu+9L8rB+ZvVZTOj8=| RmY=        | 0r0=| 13   |
| 5PjIxhS0XXg=| YzBAVg==            | ZNc8g6nY    | JYoAQdjCx86D| Vz6XoRh8JuHv+Q==| RmYRPVv/JmD4QHRJhXzPSwxgUi0=| 0bU=        | r60=| 14   |
| AEE9L0LTN8k=| Y9cimL7Y            | MYQeXNnX    | IE/10mEN    | RmYRPVv/JmT+QQ==| 0bVgvxZ1OWS7km5RfGEclkx7JMQ=| rqQ=        | 4YA=| 15   |
| F6he59y/lr4=| JIsOScTL08iQymU=    | NUHq0GY=    | JRdvTiOA    | 0bVgvxZ1OWa8lg==| rqQMfahhcLtYlAPqtQ0ZF+TY/uc=| 44Y=        | gmY=| 16   |
| UfV9OKaxo74=| NVvl1GY=            | JBlsSDmA    | rMQC3nMAUw==| rqQMfahhcLpcnQ==| 44aTplZ5AgXWgY+0OkLc+av3lNI=| h2c=        | DNw=| 17   |
| Vz6XoRh9IuU=| NxhgLSaORBmJ        | s8QdwHU=    | zNFl        | 44aTplZ5AgXUiA==| h2dUoUCOFrFZ0MyO8lEJoYueYc0=| BNk=        | j6s=| 18   |
| RmYRPVv7JGc=| qcQezmg=            | 0ttmDNYe    | lPfxxC8NfQ==| h2dUoUCOFrRY2Q==| BNkNswYJ620zZXzbEAQYSReIeto=| i6I=        | 2XE=| 19   |
| 0bVgvxZ8OWQ=| zdVxBA==            | kPnxwiMa    | 8xgp2D75c8Qz| BNkNswYJ62gyZw==| i6KoLHl7obX0t80k4ui3ZBY+uHQ=| 2nk=        | 6iM=| 20   |
| rqQMfahjeLw=| hf/gwikbe3So8A==    | 4RI20A==    | c6hvwH94    | i6KoLHl7obHysA==| 2nkH7x0qaTVarHQtcMHYeD1NY8M=| 6yo=        | 7tk=| 21   |
| 44aTplZ5Bww=| +hY22DGeZ9I8rr/3jS8=| bqhwzGR4    | 6d3cTgAMxMCe| 2nkH7x0qaTZZqw==| 6ypDjfgxV8f4pWynf9QzN8iJxAU=| 69E=        | J7Y=| 22   |
| h2dUoUCGELE=| Zqhozw==            | /NPKXwAK    | pwhli2RUHF8=| 6ypDjfgxV8b+pg==| 69EaG6Ij/no9XRlDpv+pASLZupE=| IbU=        | O1g=| 23   |
| BNkNswYP7mw=| 9tPKSAAF0A==        | pwh0lmxJ    | jVsh+IRA    | 69EaG6Ij/no+XA==| IbUV5KYlOEL09yt3HTG/WTg6Qjk=| OFs=        | 89M=| 24   |
| i6KoLHl9pbY=| uQh5m39b            | mFUh6Y1S    | lrT7ZMg=    | IbUV5KYlOEf39A==| OFvnc/Rcd72NfKWBhAekn+tsQpU=| 9tA=        | 5/g=| 25   |
| 2nkH7x0jaTU=| iFs+9A==            | mK54f9dA    | VcpondhSXTee| OFvnc/Rcd7iMfg==| 9tC0jcgyY0XfspKEaW6NH9nSeaw=| 4f4=        | gW8=| 26   |
| 6ypDjfgzX8A=| jahpf91BhwtDLA==    | R8B3lQ==    | TyqFAI0t    | 9tC0jcgyY0HZtQ==| 4f7N+qQBbjLHETDt6agcSJnX7iM=| gGY=        | T7s=| 27   |
| 69EaG6Ij+3M=| XMR3ndc1SSGRiVgOYk8=| UiqaDJYt    | lK/A77FFBjC1| 4f7N+qQBbjHEFg==| gGb75eV4TDdfYCIL3kxvAIxp1LM=| SrM=        | LBs=| 28   |
| IbUV5KYtPkI=| WiqCDw==            | gaHW/rFD    | nI+vnt1/G1g=| gGb75eV4TDZZYw==| SrNx38vTSA7aSsWjNEXyIADPO28=| Khg=        | mQY=| 29   |
| OFvnc/Racrw=| i6HW6bFMEg==        | nI++g9Vi    | 5heZkJkJ    | SrNx38vTSA7ZSw==| KhiM7QPJ+NbtHmhElvistpM1ARQ=| mgU=        | cLM=| 30   |
| 9tC0jcg0Z0Y=| go+zjsZw            | 8xmZgZAb    | N9aQoKE=    | KhiM7QPJ+NPuHQ==| mgV2dMWfo3PBgiUoy23Rn0dj+wM=| dbA=        | 7qM=| 31   |
| 4f7N+qQJbDc=| /Beeh5Q=            | LMITqrei    | SGnxkmA=    | mgV2dMWfo3bAig==| dbA9FNiPehRvLfCNQOzW46CRHI4=| 6KU=        | fPQ=| 32   |
| gGb75eV+Tj4=| MMwAvq6qNg==        | Um3uk3K3jKOH| 7XQUB7zu    | dbA9FNiPehBoLg==| 6KUNEdX19e0UFXsaSoo8auyfYCw=| ef0=        | zjI=| 33   |
| SrNx38vSTQs=| VH3ulHI=            | +nQUFrQ=    | CM9fYabw    | 6KUNEdX19e4dHA==| ef10WU6FQMQ+QqQy0YG7q1GEM+0=| zTI=        | Z1U=| 34   |
| KhiM7QPN+tc=| 7GcHCrbmwAC+        | Cc9dYbI=    | ldRvdayLgIc=| ef10WU6FQMY/Qg==| zTJdwkFf6PhrT3ZYTmqhoJCH3Vs=| YFQ=        | VTU=| 35   |
| mgV2dMWXoXA=| BsFfaa3x            | n9RvYqyE    | BYIULCQ=    | zTJdwkFf6PhuSQ==| YFQN1PcFzUo3KlvN+2c9LkAi1qc=| Vjw=        | Kik=| 36   |
| dbA9FNiGeRI=| mdl4a6SLgY9l        | G4IAOzfyJbFU| ukM/sTgu    | YFQN1PcFzUo2Lg==| VjynIt5rUvHaf4ea+/uOhthOY8U=| KyE=        | HSA=| 37   |
```

Figure 71. Encrypted database

At the end of the storage phase, the SE module uploads the encrypted database and the search index in the remote PostgreSQL server. Figure 72 shows the content of the PostgreSQL server. The latter stores the encrypted database (lab_simple_encrypted) as well as the index in a table named lab_simple_index.

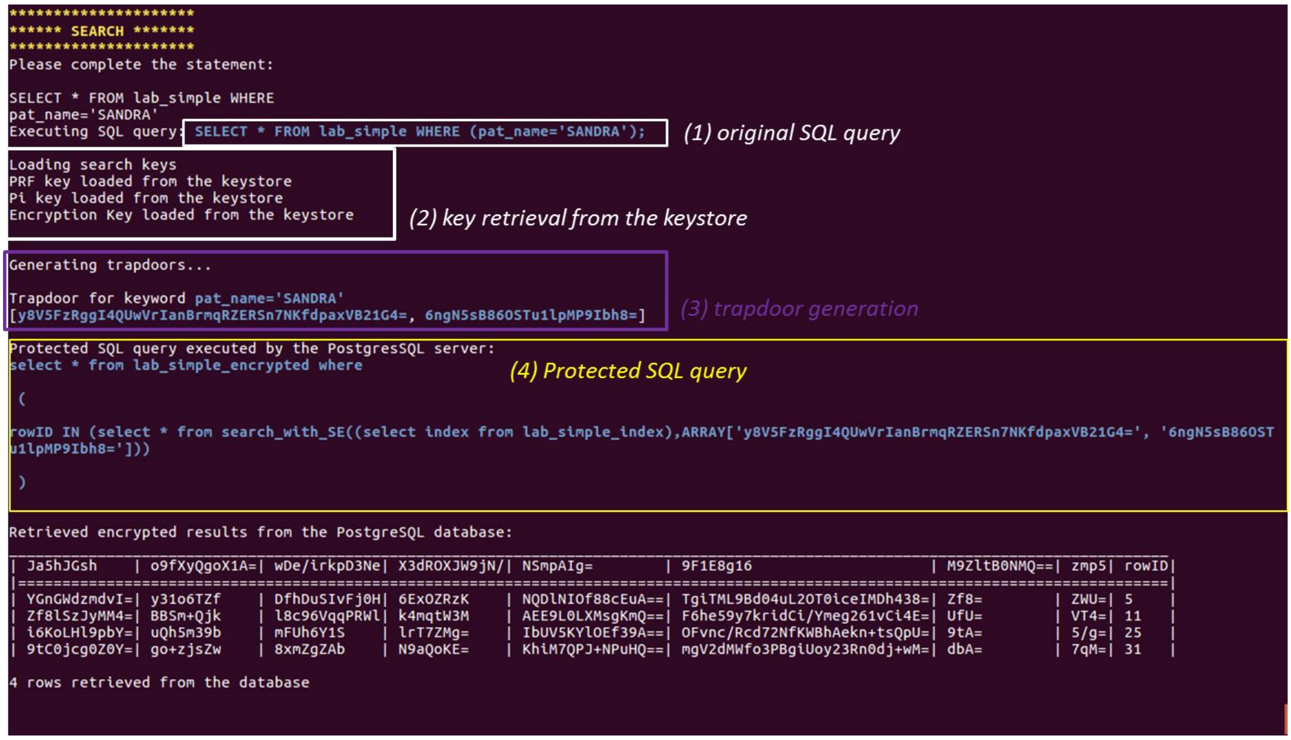


```
postgres=# \d
             List of relations
 Schema |         Name         | Type  | Owner
--------+----------------------+-------+-------
 public | lab_simple_encrypted | table | user
 public | lab_simple_index     | table | user
(2 rows)
```

Figure 72. Database and index uploaded to the PostgreSQL server

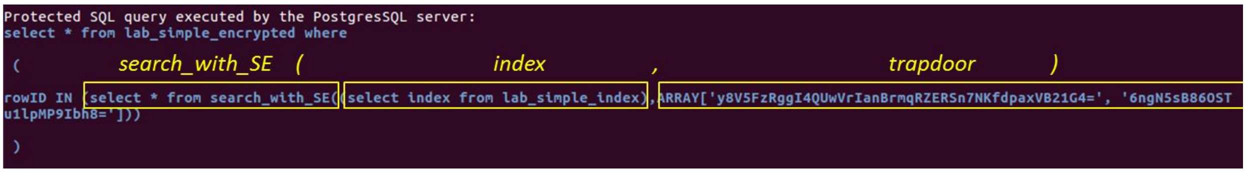
Search phase

Searching a keyword using the SE module consists in a challenge-response protocol between the CLARUS user and the PostgreSQL server. Figure 73 depicts the transcript of the search operation. (1) The user first creates a (plaintext) SQL query such as SELECT * FROM lab_simple WHERE pat_name='SANDRA'. This query targets all the records whose attribute pat_name takes the value SANDRA. To transform this query into a secure SE query (the encrypted version of the plaintext SQL query), the SE module first retrieves the keys generated during the storage phase (2). Consequently, based on these keys and the search criterion pat_name='SANDRA', the SE module generates the trapdoor, that is, the secure search token, that will enable the cloud to search for this criterion, without knowing the content of the SE query (3). Finally, the SE module forms the SE query by replacing the table name by the name of the encrypted table, and by replacing the where statement by the following statement (simplified for ease of exposition): RowID IN search_with_SE(index, trapdoor) (4).



```
*********************
****** SEARCH *******
*********************
Please complete the statement:

SELECT * FROM lab_simple WHERE
pat_name='SANDRA'
Executing SQL query: SELECT * FROM lab_simple WHERE (pat_name='SANDRA');
Loading search keys
PRF key loaded from the keystore
Pi key loaded from the keystore
Encryption Key loaded from the keystore

Generating trapdoors...

Trapdoor for keyword pat_name='SANDRA'
[y8V5FzRggI4QUwVrIanBrmqRZERSn7NKfdpaxVB21G4=, 6ngN5sB86OSTu1lpMP9Ibh8=]
Protected SQL query executed by the PostgresSQL server:
select * from lab_simple_encrypted where

 (

rowID IN (select * from search_with_SE((select index from lab_simple_index),ARRAY['y8V5FzRggI4QUwVrIanBrmqRZERSn7NKfdpaxVB21G4=', '6ngN5sB86OST
u1lpMP9Ibh8=']))

 )

Retrieved encrypted results from the PostgreSQL database:
______________________________________________________________________________________________________________________________________
| Ja5hJGsh    | o9fXyQgoX1A=| wDe/irkpD3Ne| X3dROXJW9jN/| NSmpAIg=        | 9F1E8g16                    | M9ZltB0NMQ==| zmp5| rowID|
|====================================================================================================================================|
| YGnGWdzmdvI=| y31o6TZf    | DfhDuSIvFj0H| 6ExOZRzK    | NQDlNIOf88cEuA==| TgiTML9Bd04uL2OT0iceIMDh438=| Zf8=        | ZWU=| 5    |
| Zf8lSzJyMM4=| BBSm+Qjk    | l8c96VqqPRWl| k4mqtW3M    | AEE9L0LXMsgKmQ==| F6he59y7kridCi/Ymeg261vCi4E=| UfU=        | VT4=| 11   |
| i6KoLHl9pbY=| uQh5m39b    | mFUh6Y1S    | lrT7ZMg=    | IbUV5KYlOEf39A==| OFvnc/Rcd72NfKWBhAekn+tsQpU=| 9tA=        | 5/g=| 25   |
| 9tC0jcg0Z0Y=| go+zjsZw    | 8xmZgZAb    | N9aQoKE=    | KhiM7QPJ+NPuHQ==| mgV2dMWfo3PBgiUoy23Rn0dj+wM=| dbA=        | 7qM=| 31   |

4 rows retrieved from the database
```

Figure 73. Simple keyword search query



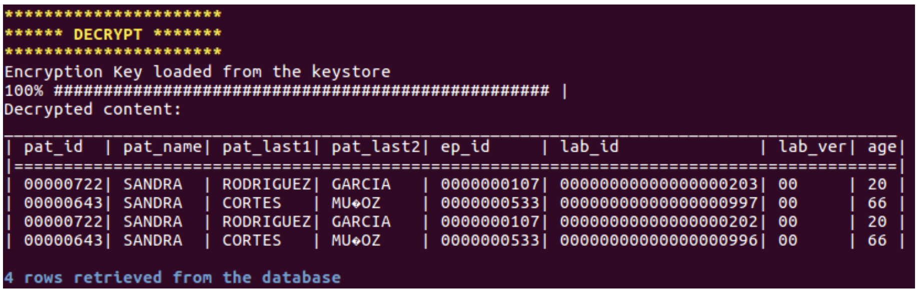
```
Protected SQL query executed by the PostgresSQL server:
select * from lab_simple_encrypted where

 (

rowID IN (select * from search_with_SE((select index from lab_simple_index),ARRAY['y8V5FzRggI4QUwVrIanBrmqRZERSn7NKfdpaxVB21G4=', '6ngN5sB86OST
u1lpMP9Ibh8=']))

 )
```

Figure 74. Protected SQL query

The function search_with_SE is the actual search function executed by the PostgreSQL server. It looks up the secure search index based on the trapdoor generated by the SE module. This function returns as output the list of RowID of the records that match the search query. The predicate RowID IN in the protected SQL query fetches the records from the outsourced encrypted database whose attribute RowID is included in the search results output by the function search_with_SE.

Decryption phase

Finally, the SE module decrypts the encrypted records that have been retrieved by the PostgreSQL server in a privacy preserving manner. To do so, the module retrieves the key material from the keystore and decrypts entry by entry the search results, as depicted in Figure 75.



```
**********************
****** DECRYPT *******
**********************
Encryption Key loaded from the keystore
100% ################################################### |
Decrypted content:
___________________________________________________________________________________
| pat_id   | pat_name| pat_last1| pat_last2| ep_id      | lab_id              | lab_ver| age|
|==================================================================================|
| 00000722| SANDRA   | RODRIGUEZ| GARCIA    | 0000000107| 00000000000000000203| 00     | 20 |
| 00000643| SANDRA   | CORTES    | MU◆OZ     | 0000000533| 00000000000000000997| 00     | 66 |
| 00000722| SANDRA   | RODRIGUEZ| GARCIA    | 0000000107| 00000000000000000202| 00     | 20 |
| 00000643| SANDRA   | CORTES    | MU◆OZ     | 0000000533| 00000000000000000996| 00     | 66 |

4 rows retrieved from the database
```

Figure 75. Decrypted search results (simple keyword search)

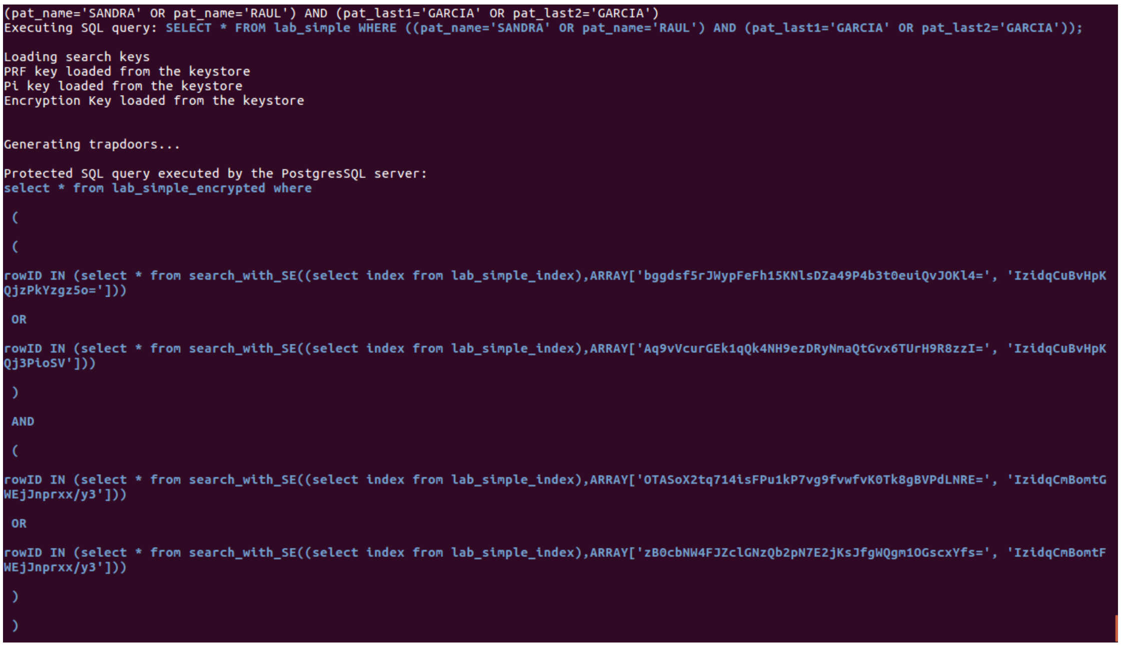
Boolean queries

The SE module allows for more complex types of search queries, namely Boolean queries and range queries, while being efficient and secure.

Figure 78 shows an example of the following Boolean query:

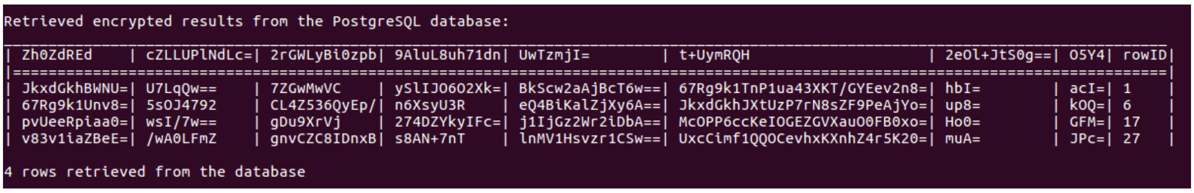
```
SELECT * FROM lab_simple WHERE (pat_name=’SANDRA’ OR pat_name=’RAUL’) AND
(pat_last1=’GARCIA’ OR pat_last2=’GARCIA’).
```

This example query aims at showing that the SE module handles brackets and Boolean operators (AND, OR). In particular, Figure 76 shows that for each element of the query of the form attribute=’value’, the SE module computes a corresponding trapdoor. Thereafter, these trapdoors are combined in the protected SE query with the same brackets and Boolean operators as in the plaintext query. In other words, the PostgreSQL will execute the function search_with_SE as many times that trapdoors are generated. Then for the search results for each of these executions are combined together by the server according to the Boolean operators. Namely, if the Boolean operator is an AND, then the server computes the intersection of the search results; if it is an OR, the server computes their union. Figure 77 shows the search results for this Boolean query.



```
(pat_name='SANDRA' OR pat_name='RAUL') AND (pat_last1='GARCIA' OR pat_last2='GARCIA')
Executing SQL query: SELECT * FROM lab_simple WHERE ((pat_name='SANDRA' OR pat_name='RAUL') AND (pat_last1='GARCIA' OR pat_last2='GARCIA'));

Loading search keys
PRF key loaded from the keystore
Pi key loaded from the keystore
Encryption Key loaded from the keystore


Generating trapdoors...

Protected SQL query executed by the PostgresSQL server:
select * from lab_simple_encrypted where

 (

 (

rowID IN (select * from search_with_SE((select index from lab_simple_index),ARRAY['bggdsf5rJWypFeFh15KNlsDZa49P4b3t0euiQvJOKl4=', 'IzidqCuBvHpK
QjzPkYzgz5o=']))

 OR

rowID IN (select * from search_with_SE((select index from lab_simple_index),ARRAY['Aq9vVcurGEk1qQk4NH9ezDRyNmaQtGvx6TUrH9R8zzI=', 'IzidqCuBvHpK
Qj3PioSV']))

 )

 AND

 (

rowID IN (select * from search_with_SE((select index from lab_simple_index),ARRAY['OTASoX2tq714isFPu1kP7vg9fvwfvK0Tk8gBVPdLNRE=', 'IzidqCmBomtG
WEjJnprxx/y3']))

 OR

rowID IN (select * from search_with_SE((select index from lab_simple_index),ARRAY['zB0cbNW4FJZclGNzQb2pN7E2jKsJfgWQgm1OGscxYfs=', 'IzidqCmBomtF
WEjJnprxx/y3']))

 )

 )
```

Figure 76. Boolean query

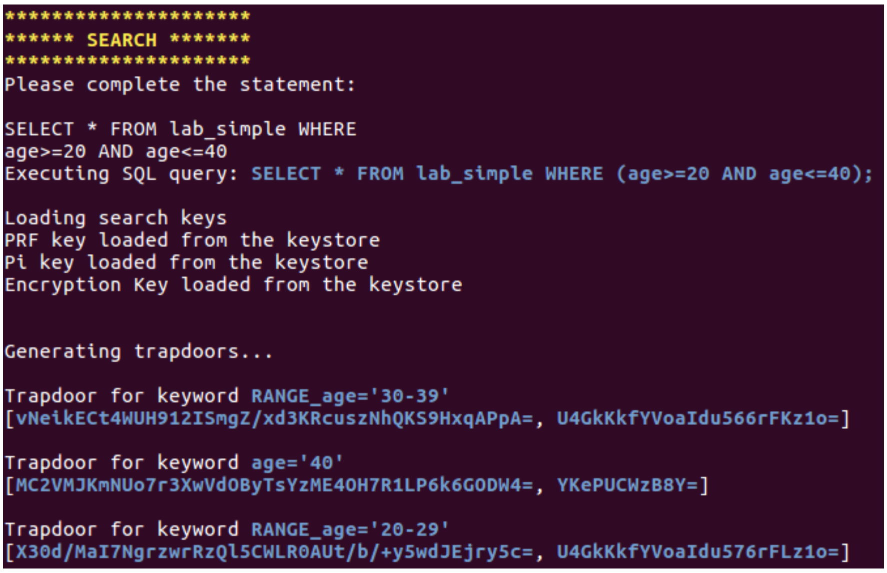


```
Retrieved encrypted results from the PostgreSQL database:
_________________________________________________________________________________________________________________________
| Zh0ZdREd    | cZLLUPlNdLc=| 2rGWLyBi0zpb| 9AluL8uh71dn| UwTzmjI=        | t+UymRQH                    | 2eOl+JtS0g==| O5Y4| rowID|
|========================================================================================================================|
| JkxdGkhBWNU=| U7LqQw==    | 7ZGwMwVC    | ySlIJO6O2Xk=| BkScw2aAjBcT6w==| 67Rg9k1TnP1ua43XKT/GYEev2n8=| hbI=        | acI=| 1    |
| 67Rg9k1Unv8=| 5sOJ4792    | CL4Z536QyEp/| n6XsyU3R    | eQ4BiKalZjXy6A==| JkxdGkhJXtUzP7rN8sZF9PeAjYo=| up8=        | kOQ=| 6    |
| pvUeeRpiaa0=| wsI/7w==    | gDu9XrVj    | 274DZYkyIFc=| j1IjGz2Wr2iDbA==| McOPP6ccKeIOGEZGVXauO0FB0xo=| Ho0=        | GFM=| 17   |
| v83v1iaZBeE=| /wA0LFmZ    | gnvCZC8IDnxB| s8AN+7nT    | lnMV1Hsvzr1CSw==| UxcCimf1QQOCevhxKXnhZ4r5K20=| muA=        | JPc=| 27   |

4 rows retrieved from the database
```

Figure 77. Search results (Boolean queries)

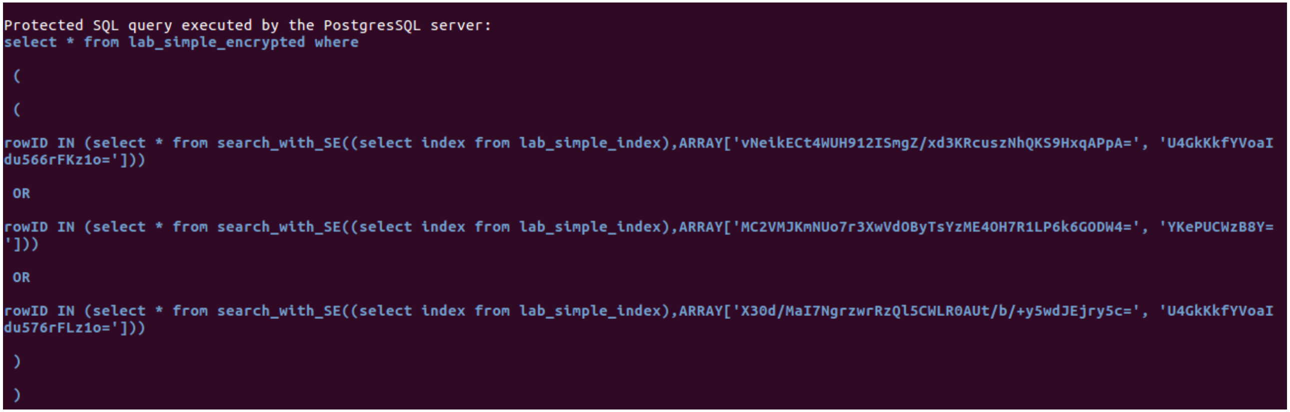
Range queries

As regards range queries, the SE module expands the SQL query of the type “age>=20 and age<=40” into several (simple) keyword search queries, as illustrated in Figure 78. In this case, the SE module identifies in the range query the intervals that have been added in the database during the storage phase. In the example of Figure 78, the SE module identifies the intervals [20-29] and [30-39]. It then computes the trapdoors for the keywords “RANGE_age=’[20-29]’” and “RANGE_age=’[30-39]’”. For the remaining values that are outside these intervals, the SE module computes trapdoors for singletons. In our example, the module generates a trapdoor for the singleton “age=40”. Thereafter, the trapdoors are combined with the “OR” operator, as illustrated in Figure 79. The search results of this query are shown in Figure 80.



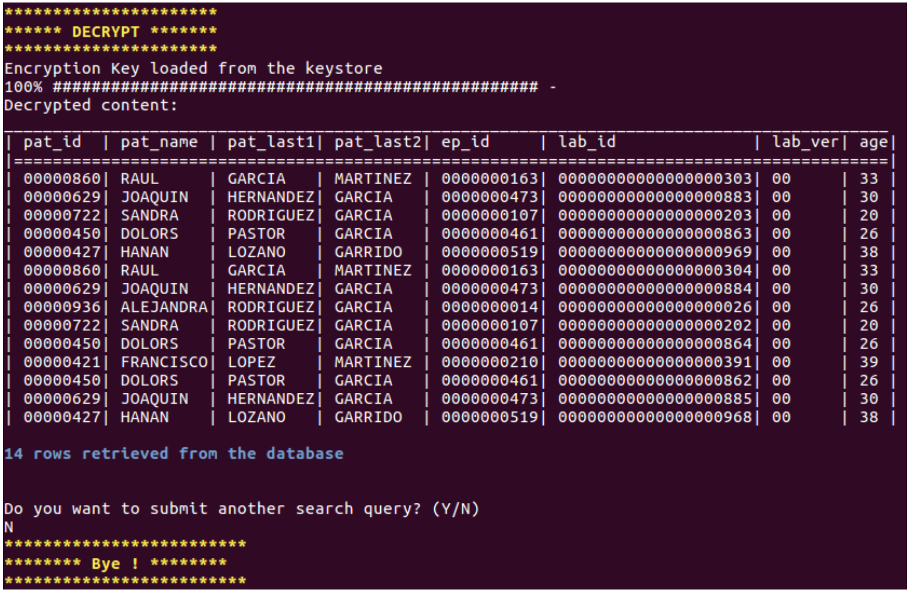
```
**********************
****** SEARCH *******
**********************
Please complete the statement:

SELECT * FROM lab_simple WHERE
age>=20 AND age<=40
Executing SQL query: SELECT * FROM lab_simple WHERE (age>=20 AND age<=40);

Loading search keys
PRF key loaded from the keystore
Pi key loaded from the keystore
Encryption Key loaded from the keystore


Generating trapdoors...

Trapdoor for keyword RANGE_age='30-39'
[vNeikECt4WUH912ISmgZ/xd3KRcuszNhQKS9HxqAPpA=, U4GkKkfYVoaIdu566rFKz1o=]

Trapdoor for keyword age='40'
[MC2VMJKmNUo7r3XwVdOByTsYzME4OH7R1LP6k6GODW4=, YKePUCWzB8Y=]

Trapdoor for keyword RANGE_age='20-29'
[X30d/MaI7NgrzwrRzQl5CWLR0AUt/b/+y5wdJEjry5c=, U4GkKkfYVoaIdu576rFLz1o=]
```

Figure 78. Trapdoor generation for range queries



```
Protected SQL query executed by the PostgresSQL server:
select * from lab_simple_encrypted where

 (

 (

rowID IN (select * from search_with_SE((select index from lab_simple_index),ARRAY['vNeikECt4WUH912ISmgZ/xd3KRcuszNhQKS9HxqAPpA=', 'U4GkKkfYVoaI
du566rFKz1o=']))

 OR

rowID IN (select * from search_with_SE((select index from lab_simple_index),ARRAY['MC2VMJKmNUo7r3XwVdOByTsYzME4OH7R1LP6k6GODW4=', 'YKePUCWzB8Y=
']))

 OR

rowID IN (select * from search_with_SE((select index from lab_simple_index),ARRAY['X30d/MaI7NgrzwrRzQl5CWLR0AUt/b/+y5wdJEjry5c=', 'U4GkKkfYVoaI
du576rFLz1o=']))

 )

 )
```

Figure 79. SE query for range queries



```
**********************
****** DECRYPT *******
**********************
Encryption Key loaded from the keystore
100% ################################################## -
Decrypted content:
_____________________________________________________________________________________________
| pat_id   | pat_name | pat_last1| pat_last2| ep_id      | lab_id              | lab_ver| age|
|============================================================================================|
| 00000860| RAUL     | GARCIA    | MARTINEZ | 0000000163| 00000000000000000303| 00      | 33 |
| 00000629| JOAQUIN  | HERNANDEZ| GARCIA    | 0000000473| 00000000000000000883| 00      | 30 |
| 00000722| SANDRA   | RODRIGUEZ| GARCIA    | 0000000107| 00000000000000000203| 00      | 20 |
| 00000450| DOLORS   | PASTOR    | GARCIA    | 0000000461| 00000000000000000863| 00      | 26 |
| 00000427| HANAN    | LOZANO    | GARRIDO  | 0000000519| 00000000000000000969| 00      | 38 |
| 00000860| RAUL     | GARCIA    | MARTINEZ | 0000000163| 00000000000000000304| 00      | 33 |
| 00000629| JOAQUIN  | HERNANDEZ| GARCIA    | 0000000473| 00000000000000000884| 00      | 30 |
| 00000936| ALEJANDRA| RODRIGUEZ| GARCIA    | 0000000014| 00000000000000000026| 00      | 26 |
| 00000722| SANDRA   | RODRIGUEZ| GARCIA    | 0000000107| 00000000000000000202| 00      | 20 |
| 00000450| DOLORS   | PASTOR    | GARCIA    | 0000000461| 00000000000000000864| 00      | 26 |
| 00000421| FRANCISCO| LOPEZ     | MARTINEZ | 0000000210| 00000000000000000391| 00      | 39 |
| 00000450| DOLORS   | PASTOR    | GARCIA    | 0000000461| 00000000000000000862| 00      | 26 |
| 00000629| JOAQUIN  | HERNANDEZ| GARCIA    | 0000000473| 00000000000000000885| 00      | 30 |
| 00000427| HANAN    | LOZANO    | GARRIDO  | 0000000519| 00000000000000000968| 00      | 38 |

14 rows retrieved from the database


Do you want to submit another search query? (Y/N)
N
*************************
******** Bye ! ********
*************************
```

Figure 80. Decrypted search results for range queries