

```

1 #coding: Shift_JIS
2
3 # コマンド引数で指定するICD 1 0 抽出条件にマッチするICD病名をもつ（疑いを除
- く）患者を検索し、
4 # その患者IDに関する検査結果テーブル名を取得して、その検査結果テーブル名に登
- 録されている
5 # 検査結果数がおおよそ指定件数以上である検査の場合には、その患者の検査結果レコ
- ードを抽出し
6 # CSV用に出力する
7 # 数値検査だけ出力する
8 # 検体材料として静脈血だけを対象とする
9
10
11 import os
12 import psycopg2
13 from psycopg2 import sql
14 import sys # モジュール属性 argv を取得するため
15 from datetime import datetime
16
17 # 患者IDからその患者の検査結果を含むすべての検査結果テーブル名をタプルで返す
18 def get_pt2obs(dbcon, pid):
19     with dbcon.cursor() as cursor:
20         cursor.execute("select * from patient_observation_rel where patient_id =
- (%s)" , [pid])
21         rs = cursor.fetchall()
22         cursor.close()
23         return rs;
24
25 # 検査結果テーブルタプル（患者IDとペアになっている）のすべての検査レコードを一
- 定件数以上であれば出力する
26 def out_obsConditioned(hdstr, dStart,dEnd, dbcon, pt2obs,query3,
- threshold=100000,obsFetchLimit=10, limitOutputLines=10):
-
27 #   sqlStr = "select n_live_tup from pg_stat_user_tables where relname=(%s)"
28   for obsrec in pt2obs:   # 全レコード処理
29       patient_id = obsrec[0]
30       obsTableName = obsrec[1]           # obsrec[0]: patient_id,  obsrec[1]:
-       tablename

```

```

31     if obsTableName.endswith("81") or obsTableName.endswith("82") or
-     obsTableName.endswith("84") or obsTableName.endswith("85"):
32         out_obs(hdstr, dStart,dEnd, dbcon, obsTableName, patient_id,
-         query3,obsFetchLimit, limitOutputLines)
33     return
34
35 # 指定された1つの検査結果テーブルのすべての検査レコードを出力する
36 def out_obs(hstr, dStart,dEnd, dbcon, obsTableName, pid,
- query3,fetchLimit,limitOutputLines):
37     localCode = (obsTableName)[12:]
38     outlines =0
39     with dbcon.cursor('queyObs') as cursoObsTable:
40         cursoObsTable.execute(sql.SQL(query3).format(sql.Identifier(obsTableName))
-         , {'pid':pid,'startdis':dStart, 'enddis':dEnd })
41     while True:
42         rows = cursoObsTable.fetchmany(fetchLimit)
43         if len(rows) <= 0:
44             break;
45         for rec in rows:
46             print(hstr , end="")
47             print(localCode , end="")
48 #             sys.stderr.write ( localCode)
49
50             for i in range(len(rec)):
51                 print(", ", end="")
52                 print(rec[i], end="")
53 #                 sys.stderr.write ( ", ")
54 #                 sys.stderr.write ( str(rec[i]))
55             print()
56 #             sys.stderr.write ( "%n")
57         if len(rows) < fetchLimit:
58             break;
59         outlines +=len(rows)
60         if outlines >=limitOutputLines:
61             break;
62     cursoObsTable.close()
63     return
64

```

```

65 # プログラム本体メインルーチン
66
67 file_name1 = 'sqlICD.sql'
68 file_name2 = 'sqlDis.sql'
69 file_name3 = 'selectObs.sql'
70
71
72 with open(file_name1, 'r') as f1: # ICDサマリテーブル検索用
73     query1 = f1.read()
74 f1.close()
75
76 with open(file_name2, 'r') as f2: # 個別の病名テーブル（交換コード別テーブル）検索
77     query2 = f2.read()
78 f2.close()
79
80 with open(file_name3, 'r') as f3: # 個別の検査結果テーブル（検査ローカルコード別テ
81     query3 = f3.read()
82 f3.close()
83
84 argvs = sys.argv # コマンドライン引数を格納したリストの取得
85 argc = len(argvs) # 引数の個数
86
87
88 # psycopg2 の execute ではバインドパラメータを %(変数名)s で指定する必要がある
89
90 url = 'postgresql://ssmix2summary_101_2:dS2qbXef@192.41.180.250:5432/
91     - smmix2summary_101_2db'
92
93 limitOutputLines1 = 100000
94 limitOutputLines2 = 100000 # for each disCode
95 limitOutputLines3 = 100000 # for each obsCode
96 threshold = 100000
97 icdFetchLimit = 1000
98 disFetchLimit = 1000
99 obsFetchLimit = 1000
100 datetime19000101 = datetime(1900,1,1)

```

```

100 datetime21001231 = datetime(2100,12,31)
101 try:
102     conn = psycopg2.connect(url)
103     sys.stderr.write("successfully connected"+"%n")
104     tablename = "icd10_summary"
105     if argc <2 :
106         condvalue = 'E21%' # 対象とするICD10パターン
107     else :
108         icd10Pattern=argvs[1]
109         condvalue = icd10Pattern
110     sys.stderr.write("ICD Pattern"+condvalue+"%n")
111
112     with conn.cursor('queryEach') as cursorDisSummary:
113         cursorDisSummary.execute(sql.SQL(query1).format(sql.Identifier(tablename)),
114             - {'condval':condvalue} )
115         lines = 0
116         while True:
117             rowsDisSummary = cursorDisSummary.fetchmany(icdFetchLimit)
118             sys.stderr.write("enter loop%n")
119             sys.stderr.write("rowsDisSummary="+str(len(rowsDisSummary))+ "%n")
120             if len(rowsDisSummary) <= 0:
121                 break;
122             for recDisSummary in rowsDisSummary: # 全レコード処理
123                 lines +=1
124                 sys.stderr.write (str(lines))
125                 for i in range(len(recDisSummary)): # 全カラム処理
126                     sys.stderr.write(",")
127                     sys.stderr.write(str(recDisSummary[i]))
128                 sys.stderr.write ( "%n")
129             #
130             if recDisSummary[2] != "disease_none" :
131                 with conn.cursor('queryDisease') as cursorDisTable:
132                     cursorDisTable.execute(sql.SQL(query2).format(sql.Identifier(
133                         - recDisSummary[2])) ) # recDisSummary[2]は病名交換用
134                     - コード別の病名テーブル名
135                     dislines=0;
136                     while True:
137                         disrows = cursorDisTable.fetchmany(disFetchLimit)

```

```

135     if len(disrows) <= 0:
136         break;
137     sys.stderr.write("disrows="+str(len(disrows))+ "¥n")
138     for disrec in disrows:    # 全レコード処理
139         dislines +=1
140         header =""
141     #     print(dislines, end="")
142     #     sys.stderr.write(str(dislines))
143         header +=str(dislines)
144     for j in range(len(disrec)):
145         #         # 全カラム処理
146         #         print(",", end="")
147         #         print(disrec[j], end="")
148         #         sys.stderr.write(",")
149         #         sys.stderr.write(str(disrec[j]))
150         #         header +="," + str(disrec[j])
151     #     print()
152     #     sys.stderr.write ( "¥n")Fstr
153         header +=","
154
155         patientid = disrec[2]
156         # disrec[2]はpatient_id
157         # patientidから関連する検査結果テーブルをすべてチェック
158         disStart = disrec[8]
159         if disrec[9] is None or disrec[7] ==
160             datetime19000101:
161             disEnd = datetime21001231
162         else:
163             disEnd = disrec[9]
164         rows = get_pt2obs(conn, patientid)
165         out_obsConditioned(header,
166             disStart,disEnd,conn, rows,query3,
167             threshold,obsFetchLimit,limitOutputLines3)
168     if len(disrows) < disFetchLimit:
169         sys.stderr.write("break:
170             len(disrows):"+str(len(disrows))+ " <
171             disFetchLimit¥n")

```

```

165         break;
166         if dislines >= limitOutputLines2 :
167             break;
168             cursorDisTable.close()
169         if len(rowsDisSummary) < icdFetchLimit:
170             sys.stderr.write("break: len(rowsDisSummary): "+
-             str(len(rowsDisSummary))+"<icdFetchLimit¥n")
171             break;
172         if lines >= limitOutputLines1 :
173             break;
174         cursorDisSummary.close()
175     except Exception as e:
176         print(e)
177     finally:
178         conn.close()
179
180

```