

t0Pn01: PostgreSQL API #1: ecpg (Embedded SQL)

```
1 #include <stdio.h>
2 #include <string.h>
3 EXEC SQL include sqlca;
4
5 EXEC SQL whenever sqlerror do sql_error("PostgreSQL error--");
6 void sql_error(char *pre_message)
7 {   char    error_message[512];
8     size_t  buffer_length,
9           message_length;
10    EXEC SQL WHENEVER SQLERROR CONTINUE;
11    printf("\n%s\n", pre_message);
12    sqlprint();
13    EXEC SQL ROLLBACK WORK;
14    exit(1);
15 }
16
17 int main(void)
18 {
19     EXEC SQL BEGIN DECLARE SECTION;
20     char  job_number[5],
21          job_name[40];
22     EXEC SQL END DECLARE SECTION;
23
24     EXEC SQL CONNECT to tcp:postgresql://basil:5432/spj;
25     printf("\nConnected to PostgreSQL.\n");
26
27     EXEC SQL DECLARE londonjobs CURSOR FOR
28         SELECT jno, jname FROM j WHERE city = 'London';
29
30     EXEC SQL OPEN londonjobs;
31
32     printf("\n\nThe London job numbers and names:\n\n");
33     printf("Job Number    Job Name\n");
34     printf("-----      ----- \n");
35
36     EXEC SQL WHENEVER NOT FOUND DO break;
37     for (;;) {
38         EXEC SQL FETCH londonjobs INTO :job_number, :job_name;
39         printf("%-10s   %-8s\n", job_number, job_name);
40     }
41     EXEC SQL CLOSE londonjobs;
42     EXEC SQL COMMIT;
43
44     printf("\nThe program is complete.\n\n");
45     return 0;
46 }
```

t0Pn02: PostgreSQL API #1: ecpg (Dynamic SQL)

```

1 #include <stdio.h>
2 #include <string.h>
3 EXEC SQL include sqlca;
4
5 EXEC SQL whenever sqlerror do sql_error("PostgreSQL error--");
6 void sql_error(char *pre_message)
7 {   char    error_message[512];
8     size_t  buffer_length, message_length;
9     EXEC SQL WHENEVER SQLERROR CONTINUE;
10    printf("\n%s\n", pre_message);    sqlprint();
11    EXEC SQL ROLLBACK WORK;
12    exit(1);
13 }
14
15 int main(void)
16 {   int    i;
17     EXEC SQL BEGIN DECLARE SECTION;
18     char  job_city[20], job_number[5], job_name[40], dynstmt[80];
19     EXEC SQL END DECLARE SECTION;
20
21     EXEC SQL CONNECT to tcp:postgresql://basil:5432/spj;
22     printf("\nConnected to PostgreSQL.\n");
23
24     printf("\n\nThis program will tell you the job names and job numbers\n"
25           "of the jobs located in a particular city.\n\n");
26     printf("Please enter the location of a job (ex: London): ");
27     fgets(job_city,20,stdin);  *(strchr(job_city,'\n')) = '\0';
28
29     strcpy(dynstmt, "SELECT jno, jname FROM j WHERE city = :the_city");
30     EXEC SQL PREPARE S FROM :dynstmt;
31     EXEC SQL DECLARE londonjobs CURSOR FOR S;
32
33     EXEC SQL OPEN londonjobs USING :job_city;
34
35     printf("\n\nThe London job numbers and names:\n\n");
36     printf("Job Number   Job Name\n"); printf("-----   ----- \n");
37
38     EXEC SQL WHENEVER NOT FOUND DO break;
39     for (i=0;;i++) {
40         EXEC SQL FETCH IN londonjobs INTO :job_number, :job_name;
41         printf("%-10s   %-8s\n", job_number, job_name);
42     }
43     printf("\nQuery returned %d row%s.\n\n", i, (i == 1) ? "" : "s");
44
45     EXEC SQL CLOSE londonjobs;
46     EXEC SQL COMMIT;
47     printf("\nThe program is complete.\n\n");
48     return 0;
49 }

```