

数据库 2022M03 作业 实验报告

19200132 郭俊杰

1. 实验目的

熟悉 PostgreSQL 数据库管理系统的安装、配置和基本操作，为后续 SQL 数据库语言学习和数据库应用系统开发准备实验环境。

熟悉 PostgreSQL 数据库管理系统加载示例数据集的方法，为后续 SQL 数据库语言学习准备用于测试的示例数据库。

2. 实验平台

操作系统：Windows 2000/ XP/7/8/10/11 或者 Linux

数据库管理系统：PostgreSQL

2. 实验内容

(1) 安装和启动

- A. 在选定的操作系统上安装 PostgreSQL;
- B. 操作启动和停止数据库服务 (PostgreSQL-x64-13) 的方法;
- C. 掌握 PostgreSQL 的登录参数 (主机、端口、默认数据库、用户名、密码)，完成系统登录。

(2) 使用 psql.exe 执行文件登录数据库服务器

(3) 使用 sql shell 登录数据库服务器，输入：主机、端口、默认数据库、用户名、密码，完成登录。

(4) 学习配置 postgresql.conf 文件

将 PostgreSQL 的所有消息输出由中文转为英文，说明配置方法。

验证方法为：使用 sql shell 命令提示符窗口中查询所有的数据库对象，命令为“\d”，查看输出信息是否含有中文，也可以采用其他方法。

(5) 根据已提供的 PostgreSQL 示例数据库 dvdrental.tar 文件，实现该示例数据的导入。要求采用两种方式实现：pgAdmin 图形界面程序和命令行方式。

验证方法：执行 `select * from actor;` 返回查询结果。

(6) 试图理解 PostgreSQL 示例数据库导入的原理，并将发现写入报告的“实验总结”部分。

4. 要求

(1) 独立完成，严禁相互抄袭（如有发现抄袭和被抄袭均判为 0 分），以及从网络上直接摘抄别人的观点和总结（该行为将影响报告成绩）。

(2) 实验报告符合学术写作的排版要求，请参考群文件中的“报告模板.docx”和“参考文献格式.docx”的排版格式。

(3) 实验报告内容详实，采用图文混合的方式叙述安装和配置过程。

Tip: Win+Shift+S 在 Windows 中可以快速截屏。

实验报告

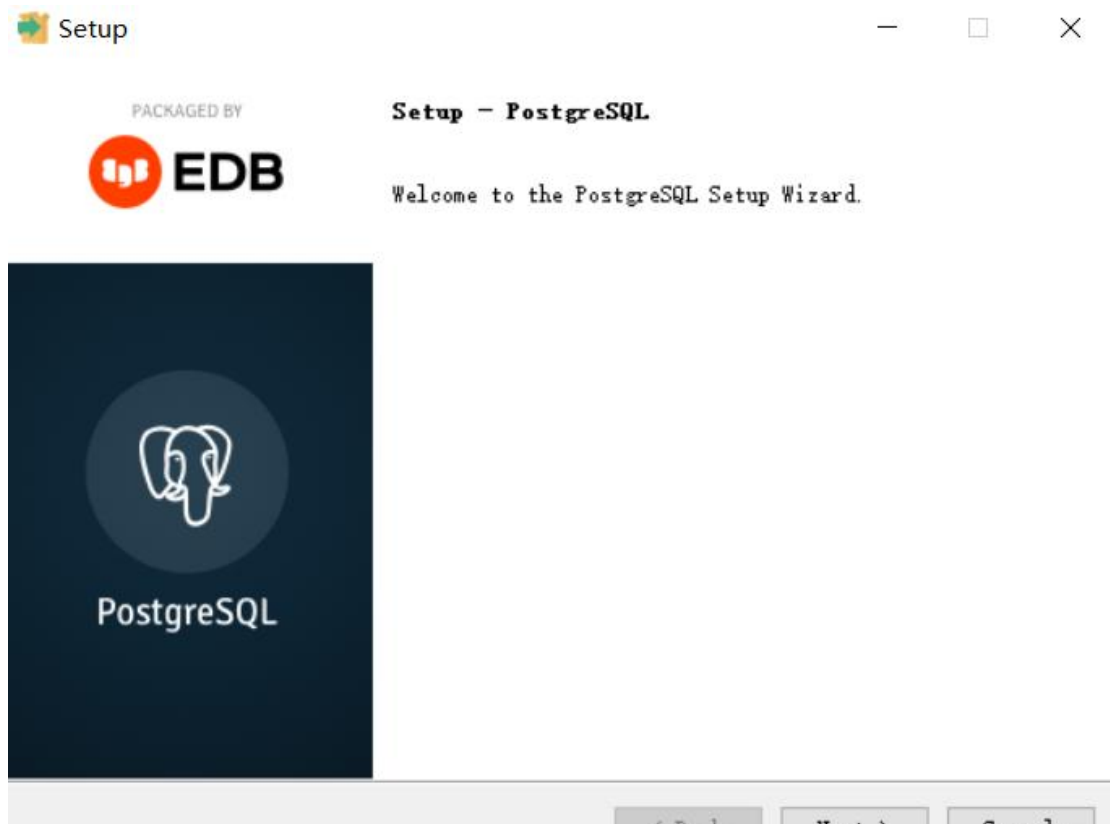
一、实验环境

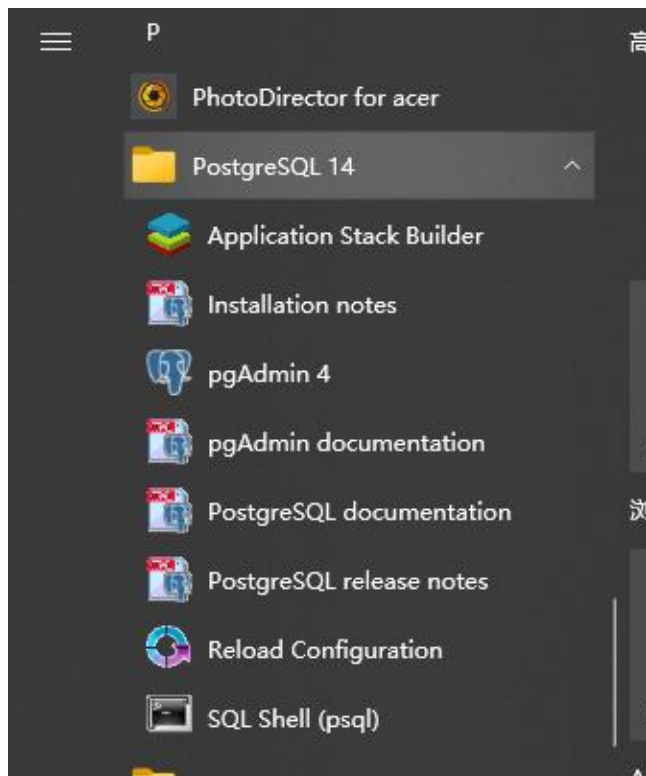
1. 操作系统：Windows 10 家庭中文版
2. 数据库管理软件（含版本号）：PostgreSQL 14
3. 其他工具：无

二、实验内容及其完成情况

（1）安装和启动

- A. 在选定的操作系统上安装 PostgreSQL；



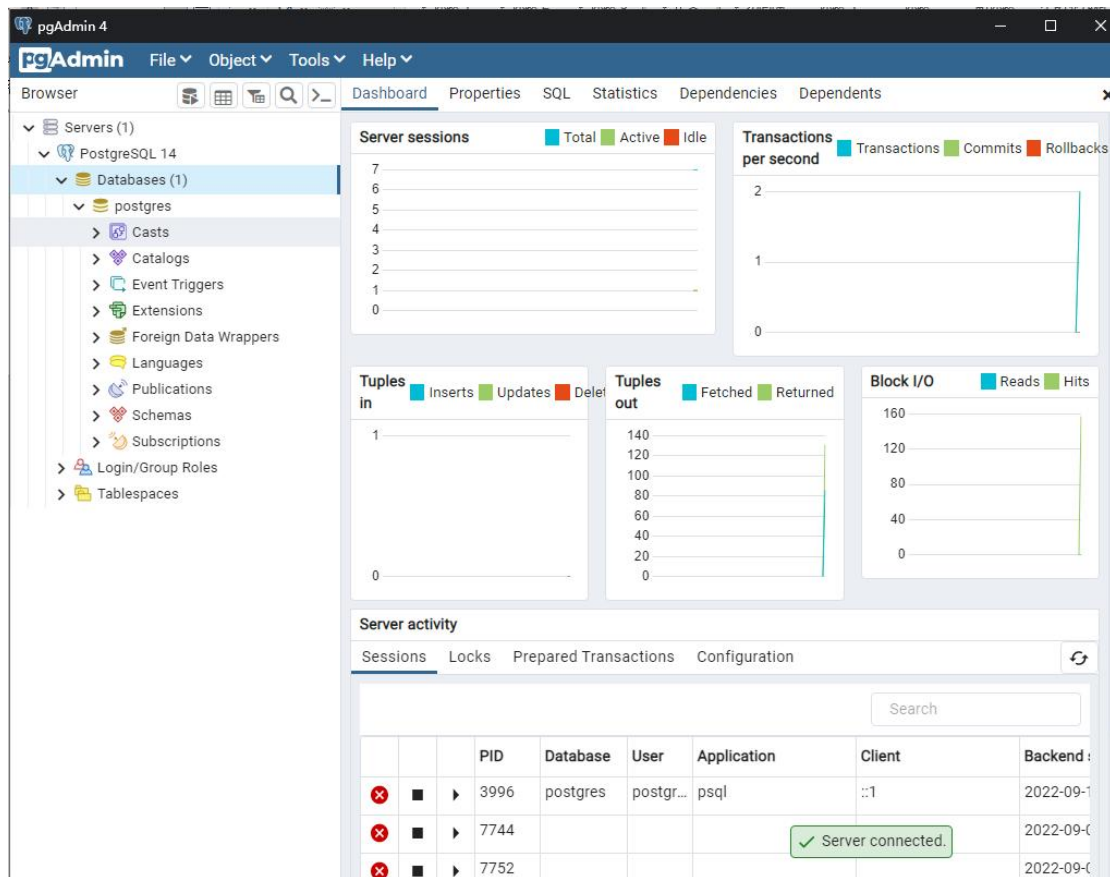


B. 操作启动和停止数据库服务 (PostgreSQL-x64-13) 的方法;

1. 通过 SQL Shell



2. 通过 pgAdmin

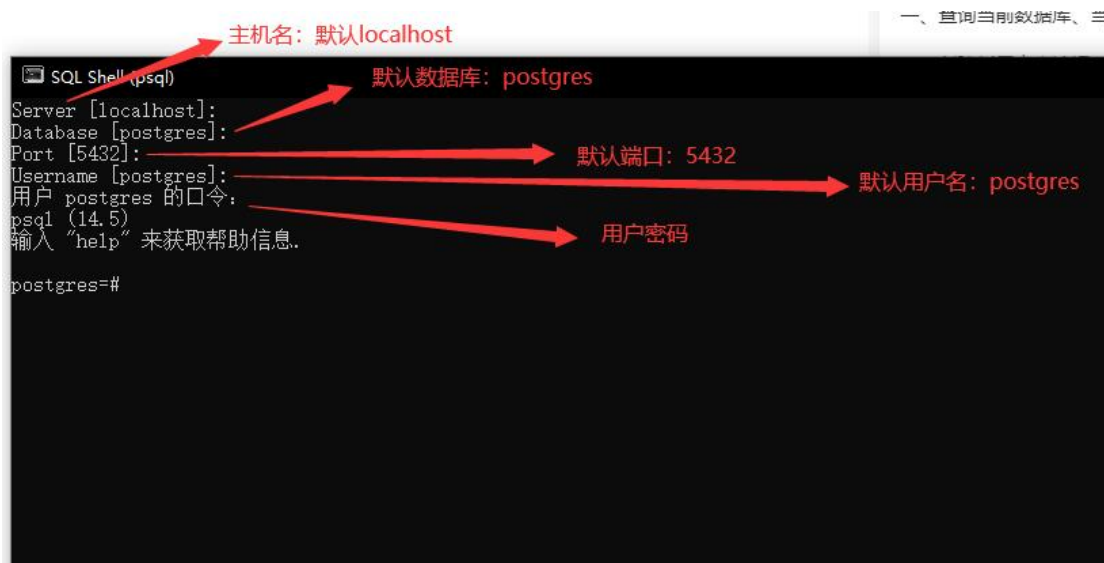


C. 掌握 PostgreSQL 的登录参数（主机、端口、默认数据库、用户名、密码），完成系统登录。

主机：默认值 localhost

端口：默认值 5432

默认数据库：postgres



(2) 使用 psql.exe 执行文件登录数据库服务器

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
用户 postgres 的口令:
psql (14.5)
输入 "help" 来获取帮助信息.
postgres=#
```

(3) 使用 sql shell 登录数据库服务器，输入：主机、端口、默认数据库、用户名、密码，完成登录。

```
SQL Shell (psql)
Server [localhost]: localhost
Database [postgres]: postgres
Port [5432]: 5432
Username [postgres]: postgres
用户 postgres 的口令:
psql (14.5)
输入 "help" 来获取帮助信息.
postgres=#
```

(4) 学习配置 postgresql.conf 文件

将 PostgreSQL 的所有消息输出由中文转为英文，说明配置方法。

验证方法为：使用 sql shell 命令提示符窗口中查询所有的数据库对象，命令为“\d”，查看输出信息是否含有中文，也可以采用其他方法。

①打开 postgresql.conf 文件

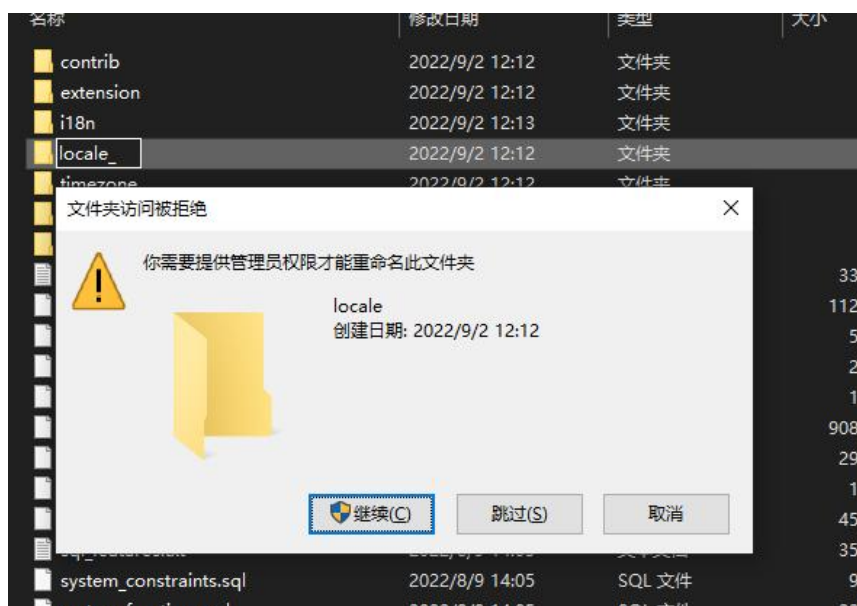
```
1 #
2 # PostgreSQL configuration file
3 #
4 #
5 # This file consists of lines of the form:
6 #
7 #   name = value
8 #
9 # (The "=" is optional.) Whitespace may be used. Comments are introduced with
10 # "s" anywhere on a line. The complete list of parameter names and allowed
11 # values can be found in the PostgreSQL documentation.
12 #
13 # The commented-out settings shown in this file represent the default values.
14 # Re-commenting a setting is NOT sufficient to revert it to the default value;
15 # you need to reload the server.
16 #
17 # This file is read on server startup and when the server receives a SIGHUP
18 # signal. If you edit the file on a running system, you have to SIGHUP the
19 # server for the changes to take effect, run "pg_ctl reload", or execute
20 # "SELECT pg_reload_conf()". Some parameters, which are marked below,
21 # require a server shutdown and restart to take effect.
22 #
23 # Any parameter can also be given as a command-line option to the server, e.g.,
24 # "postgres -c log_connections=on". Some parameters can be changed at run time
25 # with the "SET" SQL command.
26 #
27 # Memory units: B = bytes          Time units: us = microseconds
28 #               KB = kilobytes       ms = milliseconds
29 #               MB = megabytes        s = seconds
30 #               GB = gigabytes        min = minutes
31 #               TB = terabytes        h = hours
32 #                                   d = days
33 #
34 #
35 # FILE LOCATIONS
36 #
37 #
38 # The default values of these variables are driven from the -D command-line
39 # option or PGDATA environment variable, represented here as ConfigDir.
40 #
41 #
42 # Data directory:
43 #
44 # This is the default location for the database cluster.
45 #
46 #
47 #
48 #
49 #
50 #
51 #
52 #
53 #
54 #
55 #
56 #
57 #
58 #
59 #
60 #
61 #
62 #
63 #
64 #
65 #
66 #
67 #
68 #
69 #
70 #
71 #
72 #
73 #
74 #
75 #
76 #
77 #
78 #
79 #
80 #
81 #
82 #
83 #
84 #
85 #
86 #
87 #
88 #
89 #
90 #
91 #
92 #
93 #
94 #
95 #
96 #
97 #
98 #
99 #
100 #
101 #
102 #
103 #
104 #
105 #
106 #
107 #
108 #
109 #
110 #
111 #
112 #
113 #
114 #
115 #
116 #
117 #
118 #
119 #
120 #
121 #
122 #
123 #
124 #
125 #
126 #
127 #
128 #
129 #
130 #
131 #
132 #
133 #
134 #
135 #
136 #
137 #
138 #
139 #
140 #
141 #
142 #
143 #
144 #
145 #
146 #
147 #
148 #
149 #
150 #
151 #
152 #
153 #
154 #
155 #
156 #
157 #
158 #
159 #
160 #
161 #
162 #
163 #
164 #
165 #
166 #
167 #
168 #
169 #
170 #
171 #
172 #
173 #
174 #
175 #
176 #
177 #
178 #
179 #
180 #
181 #
182 #
183 #
184 #
185 #
186 #
187 #
188 #
189 #
190 #
191 #
192 #
193 #
194 #
195 #
196 #
197 #
198 #
199 #
200 #
201 #
202 #
203 #
204 #
205 #
206 #
207 #
208 #
209 #
210 #
211 #
212 #
213 #
214 #
215 #
216 #
217 #
218 #
219 #
220 #
221 #
222 #
223 #
224 #
225 #
226 #
227 #
228 #
229 #
230 #
231 #
232 #
233 #
234 #
235 #
236 #
237 #
238 #
239 #
240 #
241 #
242 #
243 #
244 #
245 #
246 #
247 #
248 #
249 #
250 #
251 #
252 #
253 #
254 #
255 #
256 #
257 #
258 #
259 #
260 #
261 #
262 #
263 #
264 #
265 #
266 #
267 #
268 #
269 #
270 #
271 #
272 #
273 #
274 #
275 #
276 #
277 #
278 #
279 #
280 #
281 #
282 #
283 #
284 #
285 #
286 #
287 #
288 #
289 #
290 #
291 #
292 #
293 #
294 #
295 #
296 #
297 #
298 #
299 #
300 #
301 #
302 #
303 #
304 #
305 #
306 #
307 #
308 #
309 #
310 #
311 #
312 #
313 #
314 #
315 #
316 #
317 #
318 #
319 #
320 #
321 #
322 #
323 #
324 #
325 #
326 #
327 #
328 #
329 #
330 #
331 #
332 #
333 #
334 #
335 #
336 #
337 #
338 #
339 #
340 #
341 #
342 #
343 #
344 #
345 #
346 #
347 #
348 #
349 #
350 #
351 #
352 #
353 #
354 #
355 #
356 #
357 #
358 #
359 #
360 #
361 #
362 #
363 #
364 #
365 #
366 #
367 #
368 #
369 #
370 #
371 #
372 #
373 #
374 #
375 #
376 #
377 #
378 #
379 #
380 #
381 #
382 #
383 #
384 #
385 #
386 #
387 #
388 #
389 #
390 #
391 #
392 #
393 #
394 #
395 #
396 #
397 #
398 #
399 #
400 #
401 #
402 #
403 #
404 #
405 #
406 #
407 #
408 #
409 #
410 #
411 #
412 #
413 #
414 #
415 #
416 #
417 #
418 #
419 #
420 #
421 #
422 #
423 #
424 #
425 #
426 #
427 #
428 #
429 #
430 #
431 #
432 #
433 #
434 #
435 #
436 #
437 #
438 #
439 #
440 #
441 #
442 #
443 #
444 #
445 #
446 #
447 #
448 #
449 #
450 #
451 #
452 #
453 #
454 #
455 #
456 #
457 #
458 #
459 #
460 #
461 #
462 #
463 #
464 #
465 #
466 #
467 #
468 #
469 #
470 #
471 #
472 #
473 #
474 #
475 #
476 #
477 #
478 #
479 #
480 #
481 #
482 #
483 #
484 #
485 #
486 #
487 #
488 #
489 #
490 #
491 #
492 #
493 #
494 #
495 #
496 #
497 #
498 #
499 #
500 #
501 #
502 #
503 #
504 #
505 #
506 #
507 #
508 #
509 #
510 #
511 #
512 #
513 #
514 #
515 #
516 #
517 #
518 #
519 #
520 #
521 #
522 #
523 #
524 #
525 #
526 #
527 #
528 #
529 #
530 #
531 #
532 #
533 #
534 #
535 #
536 #
537 #
538 #
539 #
540 #
541 #
542 #
543 #
544 #
545 #
546 #
547 #
548 #
549 #
550 #
551 #
552 #
553 #
554 #
555 #
556 #
557 #
558 #
559 #
560 #
561 #
562 #
563 #
564 #
565 #
566 #
567 #
568 #
569 #
570 #
571 #
572 #
573 #
574 #
575 #
576 #
577 #
578 #
579 #
580 #
581 #
582 #
583 #
584 #
585 #
586 #
587 #
588 #
589 #
590 #
591 #
592 #
593 #
594 #
595 #
596 #
597 #
598 #
599 #
600 #
601 #
602 #
603 #
604 #
605 #
606 #
607 #
608 #
609 #
610 #
611 #
612 #
613 #
614 #
615 #
616 #
617 #
618 #
619 #
620 #
621 #
622 #
623 #
624 #
625 #
626 #
627 #
628 #
629 #
630 #
631 #
632 #
633 #
634 #
635 #
636 #
637 #
638 #
639 #
640 #
641 #
642 #
643 #
644 #
645 #
646 #
647 #
648 #
649 #
650 #
651 #
652 #
653 #
654 #
655 #
656 #
657 #
658 #
659 #
660 #
661 #
662 #
663 #
664 #
665 #
666 #
667 #
668 #
669 #
670 #
671 #
672 #
673 #
674 #
675 #
676 #
677 #
678 #
679 #
680 #
681 #
682 #
683 #
684 #
685 #
686 #
687 #
688 #
689 #
690 #
691 #
692 #
693 #
694 #
695 #
696 #
697 #
698 #
699 #
700 #
701 #
702 #
703 #
704 #
705 #
706 #
707 #
708 #
709 #
710 #
711 #
712 #
713 #
714 #
715 #
716 #
717 #
718 #
719 #
720 #
721 #
722 #
723 #
724 #
725 #
726 #
727 #
728 #
729 #
730 #
731 #
732 #
733 #
734 #
735 #
736 #
737 #
738 #
739 #
740 #
741 #
742 #
743 #
744 #
745 #
746 #
747 #
748 #
749 #
750 #
751 #
752 #
753 #
754 #
755 #
756 #
757 #
758 #
759 #
760 #
761 #
762 #
763 #
764 #
765 #
766 #
767 #
768 #
769 #
770 #
771 #
772 #
773 #
774 #
775 #
776 #
777 #
778 #
779 #
780 #
781 #
782 #
783 #
784 #
785 #
786 #
787 #
788 #
789 #
790 #
791 #
792 #
793 #
794 #
795 #
796 #
797 #
798 #
799 #
800 #
801 #
802 #
803 #
804 #
805 #
806 #
807 #
808 #
809 #
810 #
811 #
812 #
813 #
814 #
815 #
816 #
817 #
818 #
819 #
820 #
821 #
822 #
823 #
824 #
825 #
826 #
827 #
828 #
829 #
830 #
831 #
832 #
833 #
834 #
835 #
836 #
837 #
838 #
839 #
840 #
841 #
842 #
843 #
844 #
845 #
846 #
847 #
848 #
849 #
850 #
851 #
852 #
853 #
854 #
855 #
856 #
857 #
858 #
859 #
860 #
861 #
862 #
863 #
864 #
865 #
866 #
867 #
868 #
869 #
870 #
871 #
872 #
873 #
874 #
875 #
876 #
877 #
878 #
879 #
880 #
881 #
882 #
883 #
884 #
885 #
886 #
887 #
888 #
889 #
890 #
891 #
892 #
893 #
894 #
895 #
896 #
897 #
898 #
899 #
900 #
901 #
902 #
903 #
904 #
905 #
906 #
907 #
908 #
909 #
910 #
911 #
912 #
913 #
914 #
915 #
916 #
917 #
918 #
919 #
920 #
921 #
922 #
923 #
924 #
925 #
926 #
927 #
928 #
929 #
930 #
931 #
932 #
933 #
934 #
935 #
936 #
937 #
938 #
939 #
940 #
941 #
942 #
943 #
944 #
945 #
946 #
947 #
948 #
949 #
950 #
951 #
952 #
953 #
954 #
955 #
956 #
957 #
958 #
959 #
960 #
961 #
962 #
963 #
964 #
965 #
966 #
967 #
968 #
969 #
970 #
971 #
972 #
973 #
974 #
975 #
976 #
977 #
978 #
979 #
980 #
981 #
982 #
983 #
984 #
985 #
986 #
987 #
988 #
989 #
990 #
991 #
992 #
993 #
994 #
995 #
996 #
997 #
998 #
999 #
1000 #
```

②找到 lc_messages，并将其修改

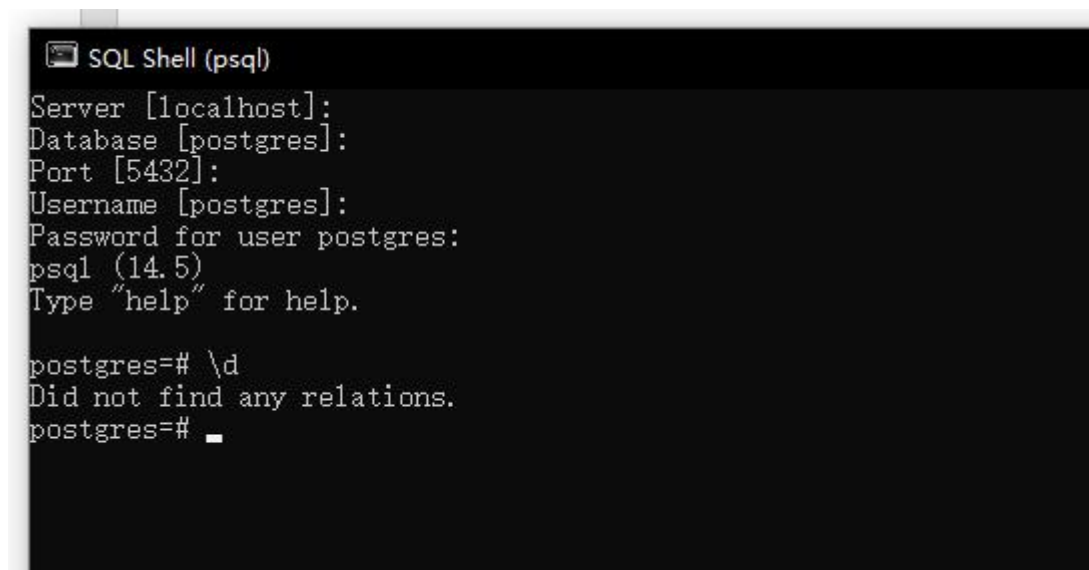
```
706 #client_encoding = sql_ascii          # actually, defaults to database
707 # encoding
708
709 # These settings are initialized by initdb, but they can be changed.
710 lc_messages = 'Chinese (Simplified)_China.936'      # locale for system error message
711 # strings
712 lc_monetary = 'Chinese (Simplified)_China.936'      # locale for monetary formatting
713 lc_numeric = 'Chinese (Simplified)_China.936'      # locale for number formatting
714 lc_time = 'Chinese (Simplified)_China.936'          # locale for time formatting
715
716 # default configuration for text search
717 default_text_search_config = 'pg_catalog.simple'
```

```
708
709 # These settings are initialized by initdb, but they can be changed.
710 lc_messages = 'C'                                  # locale for system error message
711 # strings
712 lc_monetary = 'Chinese (Simplified)_China.936'      # locale for monetary formatting
```

③在安装目录打开 share 文件夹，找到 locale 后重命名为 locale_



④验证工作

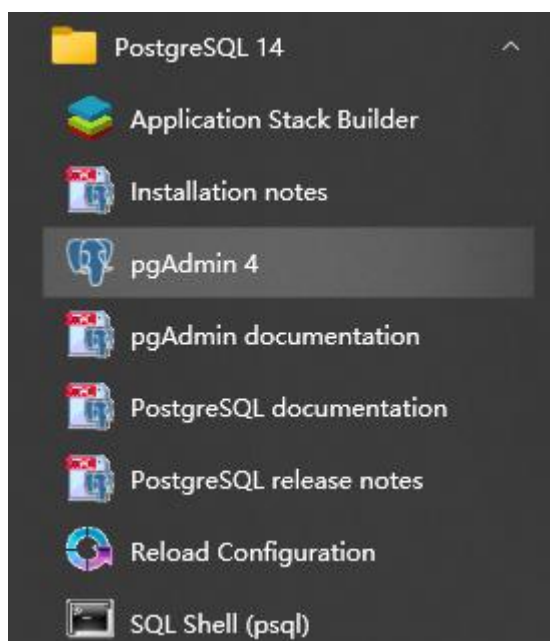


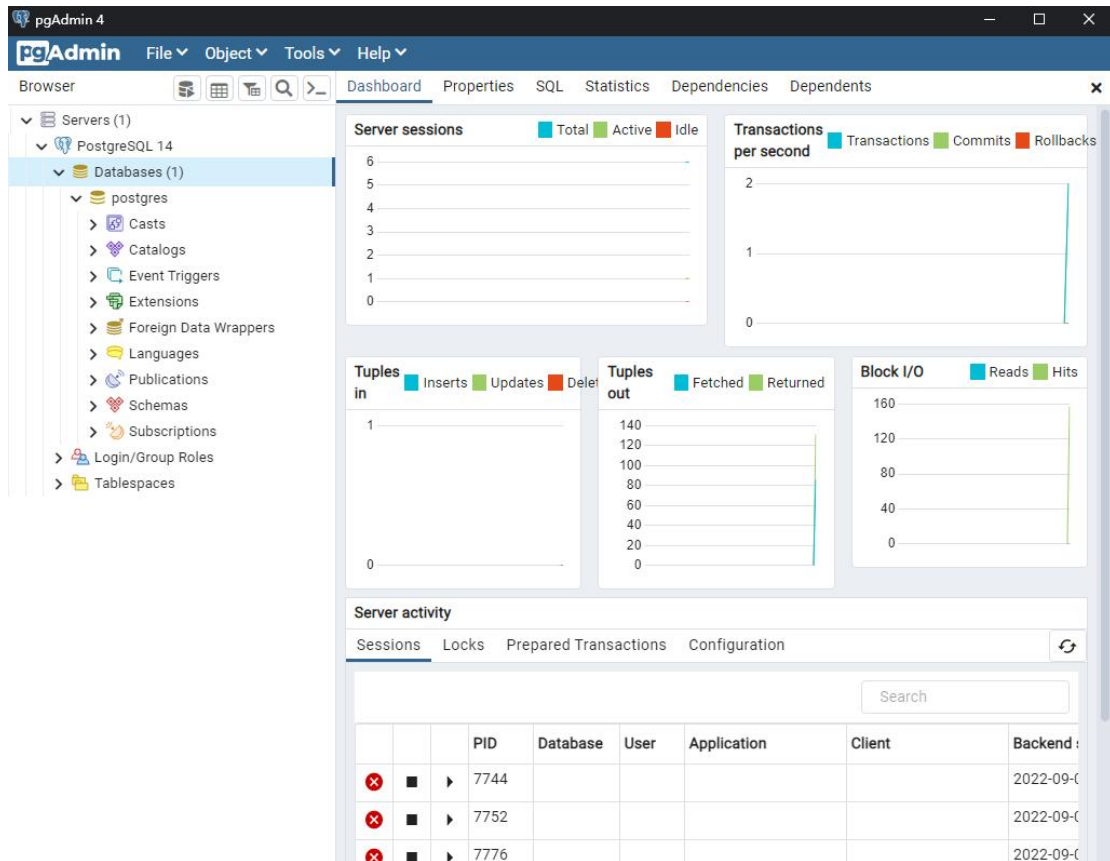
```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:
psql (14.5)
Type "help" for help.

postgres=# \d
Did not find any relations.
postgres=#
```

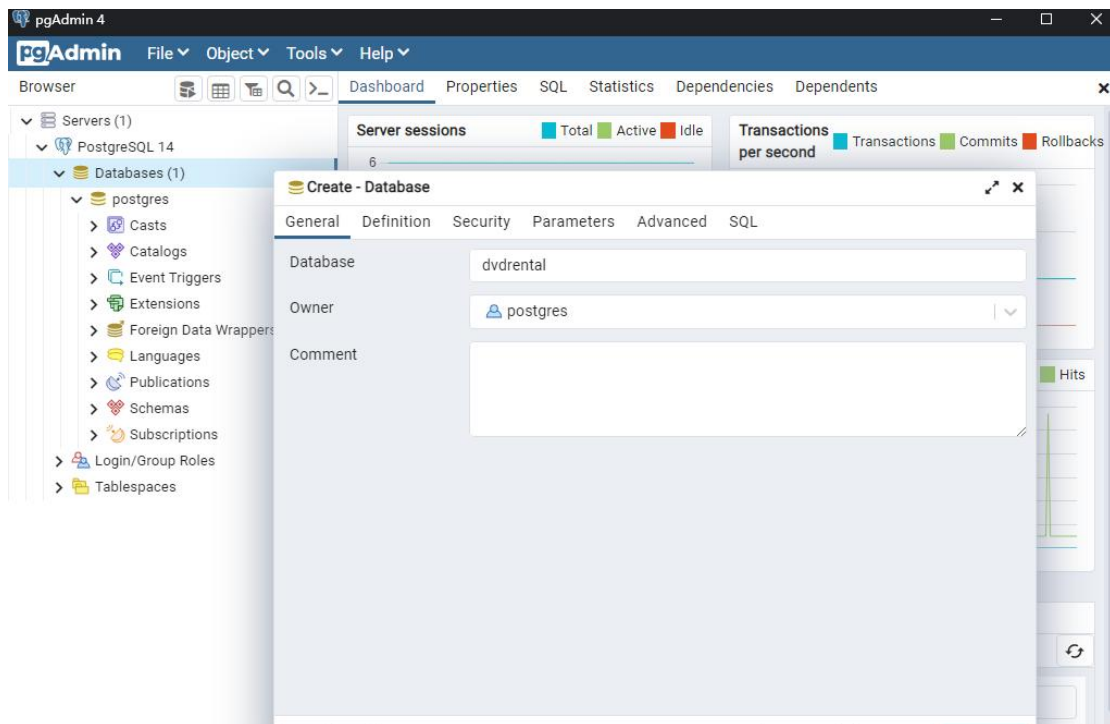
(5). pgAdmin 方法

①打开 pgAdmin 4 并且输入密码登录

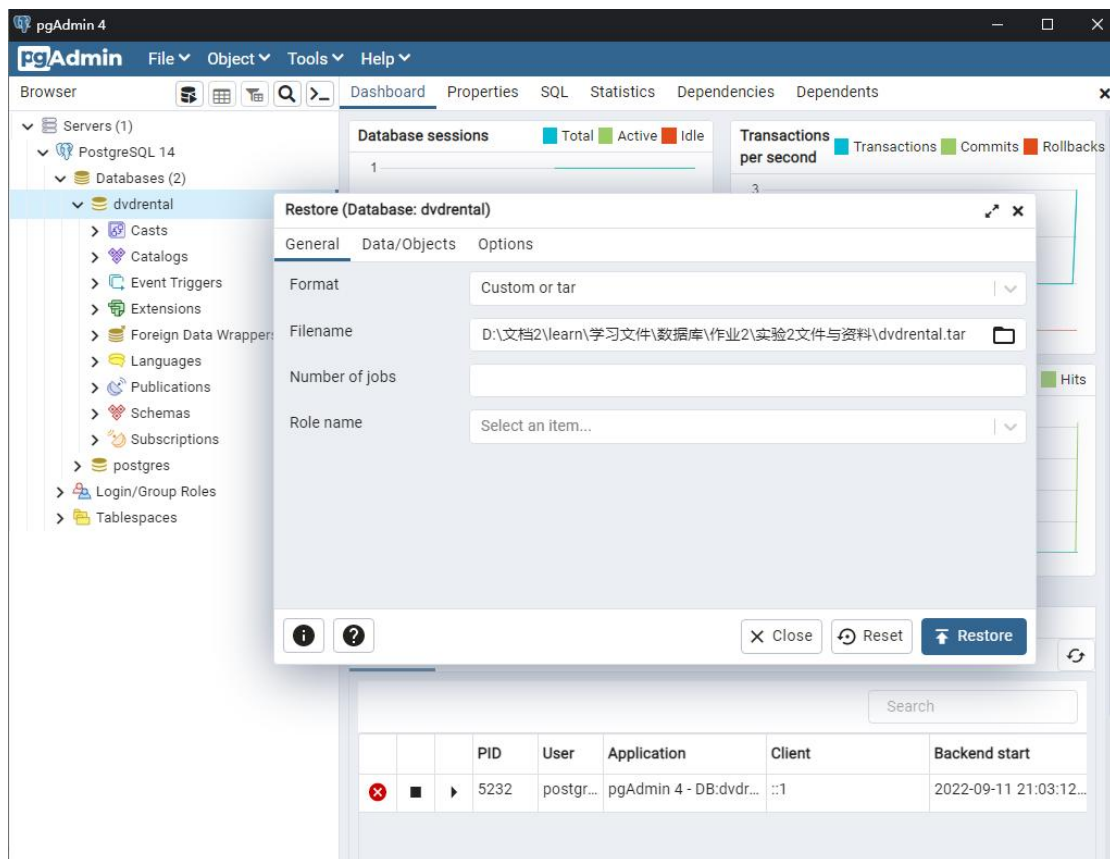




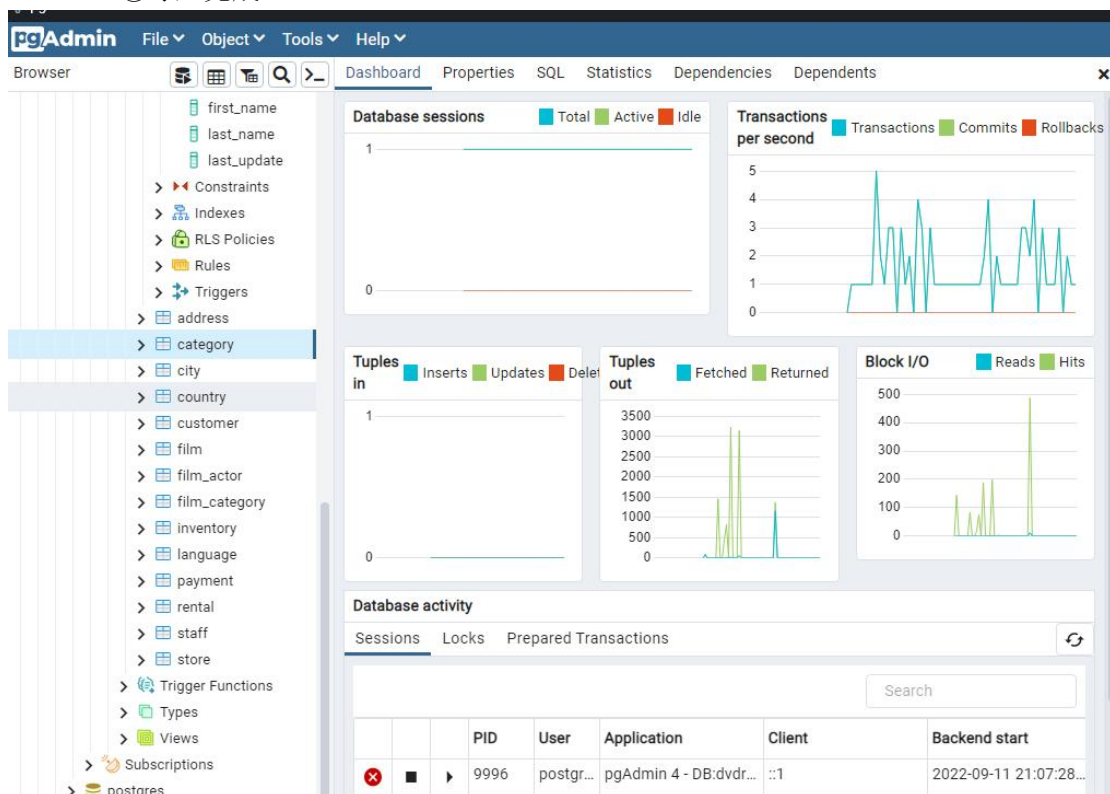
②创建数据库



③使用 restore 导入实例



⑤导入完成



(6). 命令行方法

①命令行登录

```
Windows PowerShell
PS D:\Softwares\PostgreSQL\bin> .\psql.exe postgresql://postgres:123456@localhost:5432/postgres
psql (14.5)
Type "help" for help.

postgres=#
```

②导入 dvdrental. tar 数据库

```
Windows PowerShell
PS D:\Softwares\PostgreSQL\bin> .\pg_restore.exe -U postgres -d dvdrental D:\文档2\learn\学习文件\数据库\作业2\实验2文件与资料\dvdrental.tar
pg_restore: while PROCESSING TOC:
pg_restore: from TOC entry 3: 2615 2200 SCHEMA public postgres
pg_restore: error: could not execute query: ERROR: schema "public" already exists
Command was: CREATE SCHEMA public;

pg_restore: from TOC entry 548: 1247 16778 TYPE mpaa_rating postgres
pg_restore: error: could not execute query: ERROR: type "mpaa_rating" already exists
Command was: CREATE TYPE mpaa_rating AS ENUM (
    'G',
    'PG',
    'PG-13',
    'R',
    'NC-17'
);

pg_restore: from TOC entry 551: 1247 16789 DOMAIN year postgres
pg_restore: error: could not execute query: ERROR: type "year" already exists
Command was: CREATE DOMAIN year AS integer
CONSTRAINT year_check CHECK (((VALUE >= 1901) AND (VALUE <= 2155)));

pg_restore: from TOC entry 232: 1255 16791 FUNCTION _group_concat(text, text) postgres
pg_restore: error: could not execute query: ERROR: function "_group_concat" already exists with same argument types
Command was: CREATE FUNCTION _group_concat(text, text) RETURNS text
LANGUAGE sql IMMUTABLE
AS $$
SELECT CASE
    WHEN $2 IS NULL THEN $1
    WHEN $1 IS NULL THEN $2
    ELSE $1 || ',' || $2
END
$$;

pg_restore: from TOC entry 243: 1255 16792 FUNCTION film_in_stock(integer, integer) postgres
pg_restore: error: could not execute query: ERROR: function "film_in_stock" already exists with same argument types
Command was: CREATE FUNCTION film_in_stock(p_film_id integer, p_store_id integer, OUT p_film_count integer) RETURNS SETOF integer
LANGUAGE sql
AS $$
SELECT inventory_id
FROM inventory
WHERE film_id = $1
AND store_id = $2
AND inventory_in_stock(inventory_id);
$ $;
```

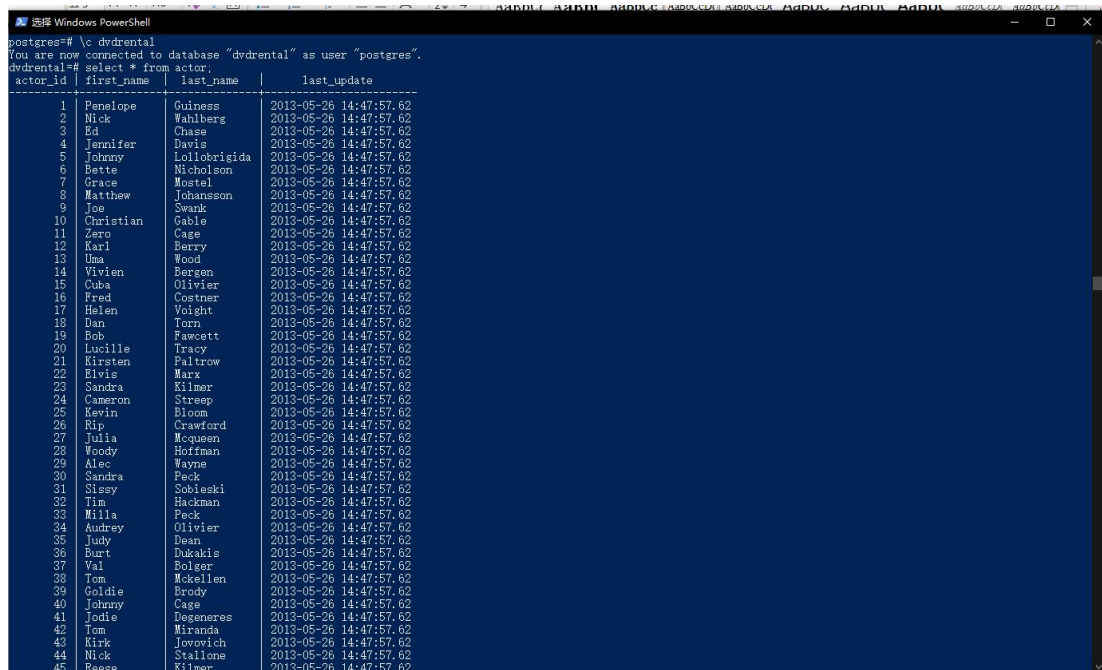
③启动并登录 postgresQL

```

pg_restore: warning: errors ignored on restore: 128
PS D:\Softwares\PostgreSQL\bin> .\psql.exe
Password for user ABC:
psql: error: connection to server at "localhost" (::1), port 5432 failed: FATAL: password authentication failed for user "ABC"
PS D:\Softwares\PostgreSQL\bin> .\psql.exe postgresql://postgres:123456@localhost:5432/postgres
psql (14.5)
Type 'help' for help.

```

④切换数据库为 dvdrental 并且执行 `select * from actor;` 返回查询结果查看已导入的数据



The screenshot shows a Windows PowerShell terminal window with the following content:

```

postgres=# \c dvdrental
You are now connected to database "dvdrental" as user "postgres".
dvdrental=# select * from actor;

```

actor_id	first_name	last_name	last_update
1	Penelope	Guinness	2013-05-26 14:47:57.62
2	Nick	Wahlberg	2013-05-26 14:47:57.62
3	Ed	Chase	2013-05-26 14:47:57.62
4	Jennifer	Davis	2013-05-26 14:47:57.62
5	Johnny	Lolobrigida	2013-05-26 14:47:57.62
6	Bette	Nicholson	2013-05-26 14:47:57.62
7	Grace	Mostel	2013-05-26 14:47:57.62
8	Matthew	Johansson	2013-05-26 14:47:57.62
9	Joe	Swank	2013-05-26 14:47:57.62
10	Christian	Gable	2013-05-26 14:47:57.62
11	Zero	Cage	2013-05-26 14:47:57.62
12	Karl	Berry	2013-05-26 14:47:57.62
13	Uma	Wood	2013-05-26 14:47:57.62
14	Vivien	Bergen	2013-05-26 14:47:57.62
15	Cuba	Olivier	2013-05-26 14:47:57.62
16	Fred	Costner	2013-05-26 14:47:57.62
17	Helen	Voight	2013-05-26 14:47:57.62
18	Dan	Turn	2013-05-26 14:47:57.62
19	Bob	Fancett	2013-05-26 14:47:57.62
20	Lucille	Tracy	2013-05-26 14:47:57.62
21	Kirsten	Paltrow	2013-05-26 14:47:57.62
22	Elvis	Marx	2013-05-26 14:47:57.62
23	Sandra	Kilmer	2013-05-26 14:47:57.62
24	Cameron	Streep	2013-05-26 14:47:57.62
25	Kevin	Bloom	2013-05-26 14:47:57.62
26	Rip	Crawford	2013-05-26 14:47:57.62
27	Julia	McQueen	2013-05-26 14:47:57.62
28	Woody	Hoffman	2013-05-26 14:47:57.62
29	Alec	Wayne	2013-05-26 14:47:57.62
30	Sandra	Peck	2013-05-26 14:47:57.62
31	Sissy	Sobieski	2013-05-26 14:47:57.62
32	Tim	Hackman	2013-05-26 14:47:57.62
33	Milla	Peck	2013-05-26 14:47:57.62
34	Audrey	Olivier	2013-05-26 14:47:57.62
35	Judy	Dean	2013-05-26 14:47:57.62
36	Burt	Dukakis	2013-05-26 14:47:57.62
37	Val	Bolger	2013-05-26 14:47:57.62
38	Tom	McKellen	2013-05-26 14:47:57.62
39	Goldie	Brody	2013-05-26 14:47:57.62
40	Johnny	Cage	2013-05-26 14:47:57.62
41	Jodie	Degeneres	2013-05-26 14:47:57.62
42	Tom	Miranda	2013-05-26 14:47:57.62
43	Rink	Jovich	2013-05-26 14:47:57.62
44	Nick	Stallone	2013-05-26 14:47:57.62
45	Reese	Kilmer	2013-05-26 14:47:57.62

三、实验总结

此次实验于我来说受益良多，我成功安装了 PostgreSQL，并且进行了一定的测试，首先我成功打开 postgresSQL 界面，然后根据自身之前设置的密码进行登录并且了解了常用的登陆参数，比如端口、默认数据库、用户名等等，此次实验开启了我实践的第一步，我也对数据库的认识也进一步加深了。

数据库导入原理：对于 PostgreSQL 示例数据库导入的原理，我在解压 tar 文件以后发现有 16 个 dat 文件（数据文件）和 1 个 sql 文件，在导入过程中，数据库读取 sql 文件，逐条执行其中的 sql 指令；15 个 dat 文件对应 15 个表；toc 文件存放了一些还原的配置信息，比如还原顺序，包含哪些表等 PostgreSQL 数据库数据导入的核心一般都使用 COPY 命令，相对于 MySQL 去一条条的执行 insert 命令来说，COPY 命令十分迅速 COPY 在 PostgreSQL 表和标准文件系统文件之间移动数据。COPY TO 将表的内容复制到文件中，而 COPY FROM 将数据从文件复制到表中(将数据追加到表中已经存在的内容)。COPY TO 还可以复制 SELECT 查询的结果。