

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Oracle的点餐系统的数据库设计 | | |
| 课程 | Oracle数据库应用 | | |
| 学 院 | 信息科学与工程学院 | | |
| 专 业 | 软件工程 | 年级 | 2018级 |
| 学生姓名 | 杨辰露 | 学号 | 201810414105 |
| 指导教师 | 赵卫东 | 职称 | 副教授 |

|  |  |  |  |
| --- | --- | --- | --- |
| **评分项** | **评分标准** | **满分** | **得分** |
| 文档整体 | 文档内容详实、规范，美观大方 | 10 |  |
| 表设计 | 表，表空间设计合理，数据合理 | 20 |  |
| 用户管理 | 权限及用户分配方案设计正确 | 20 |  |
| PL/SQL设计 | 存储过程和函数设计正确 | 30 |  |
| 备份方案 | 备份方案设计正确 | 20 |  |
| **得分合计** | | |  |

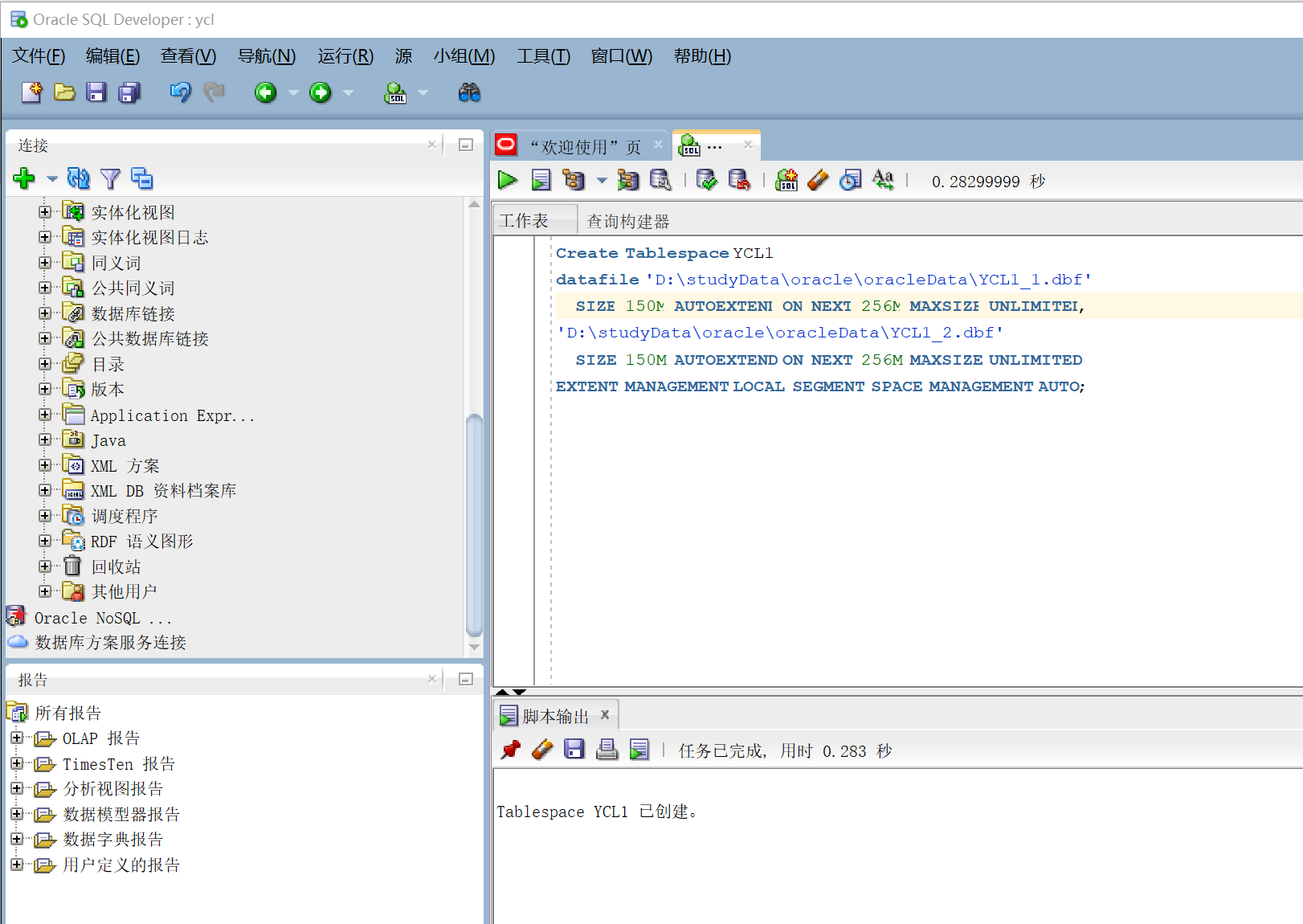
2021 年 6 月 1 日

在互联网经济飞速发展的时代，网络化企业管理也在其带领下快速兴起，开发一款自主点餐系统会受到众多商家的青睐。现如今市场上的人力资源价格是非常高昂的，一款自主点餐系统可以减少餐厅的人力开销，将服务员从繁忙的点餐过程中解脱出来，将厨师从重复制菜的烦恼中解脱出来，并减少了高峰期用餐时点餐出错的几率，同时减少了餐厅定期更新印制菜单的开销，提高了餐厅的档次和品位。

## 设计表空间使用方案

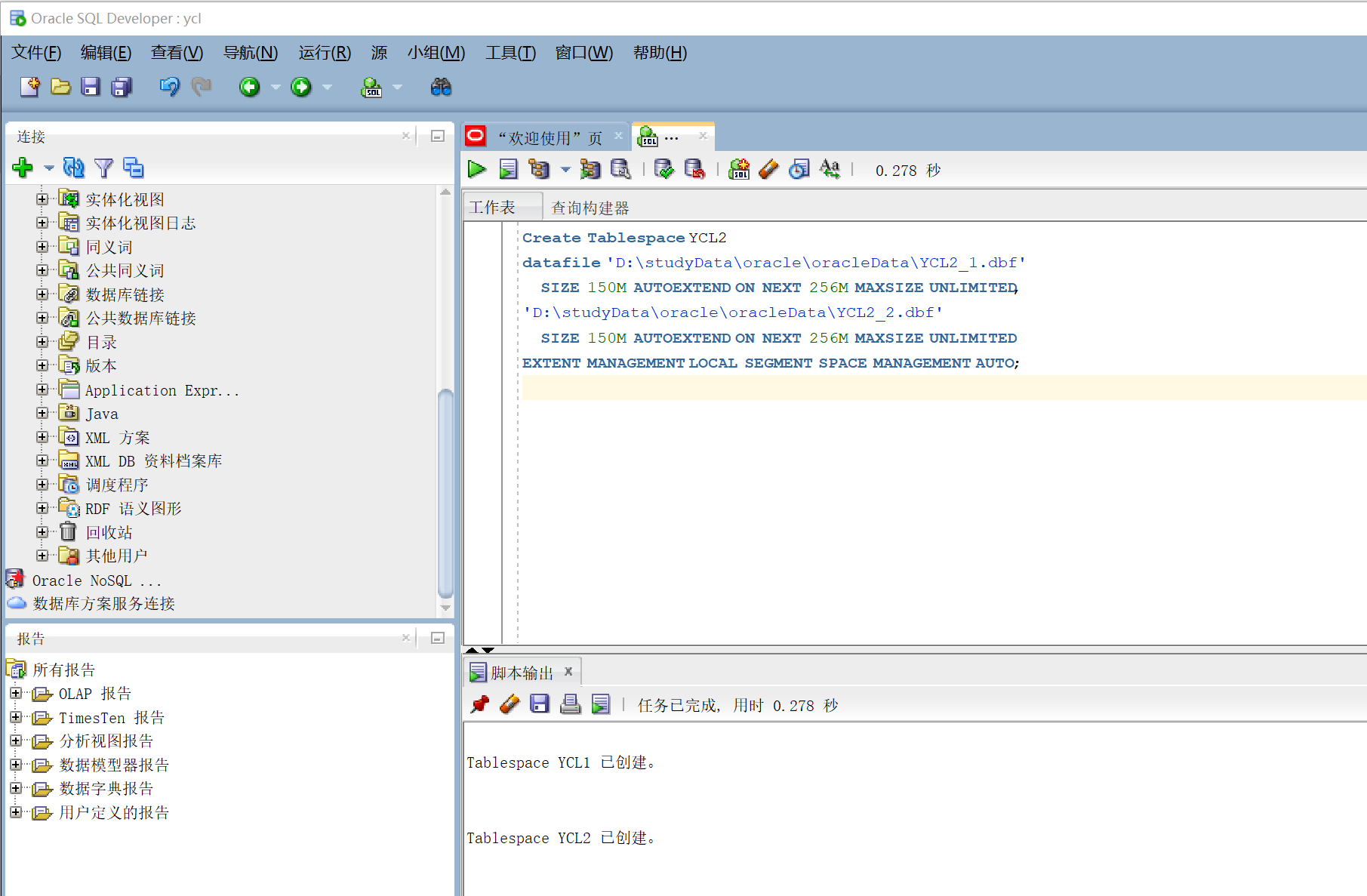
创建表空间YCL1，大小为150M

1. Create Tablespace YCL1
2. datafile 'D:\studyData\oracle\oracleData\YCL1\_1.dbf'
3. SIZE 150M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED,
4. 'D:\studyData\oracle\oracleData\YCL1\_2.dbf'
5. SIZE 150M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED
6. EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;

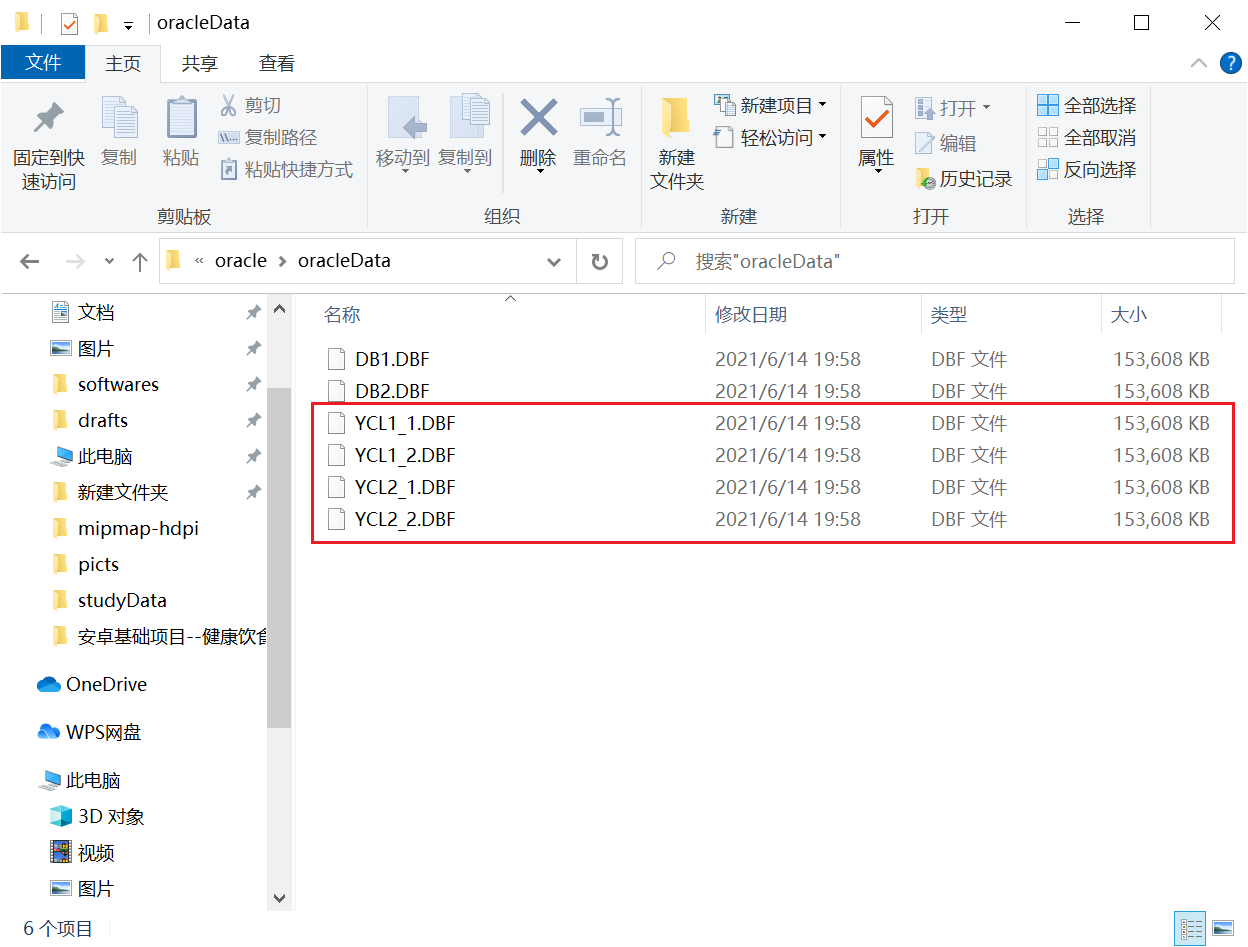


创建表空间YCL2，大小也为150M

1. Create Tablespace YCL2
2. datafile 'D:\studyData\oracle\oracleData\YCL2\_1.dbf'
3. SIZE 150M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED,
4. 'D:\studyData\oracle\oracleData\YCL2\_2.dbf'
5. SIZE 150M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED
6. EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;



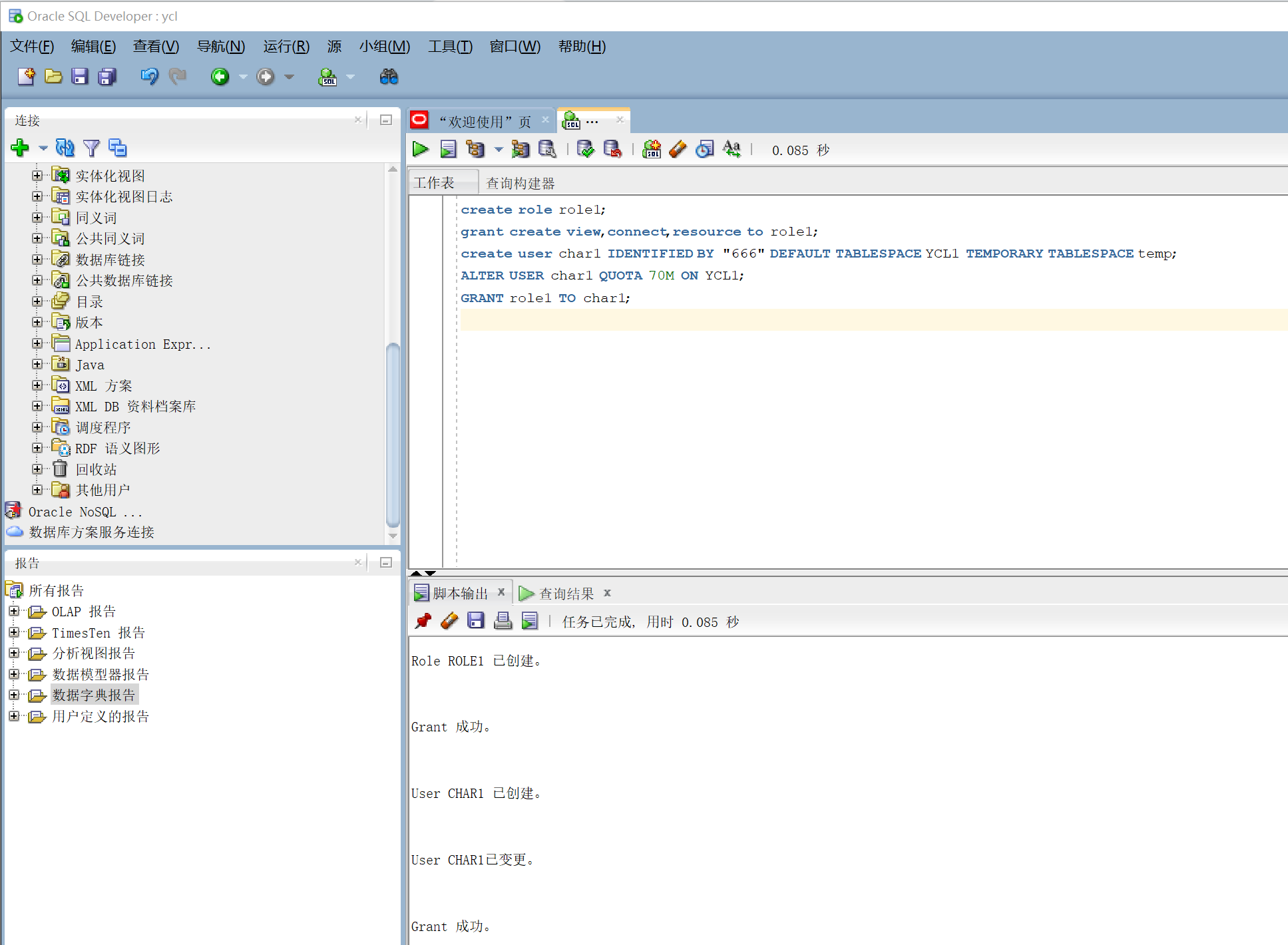
查看：



## 设计权限及用户分配方案

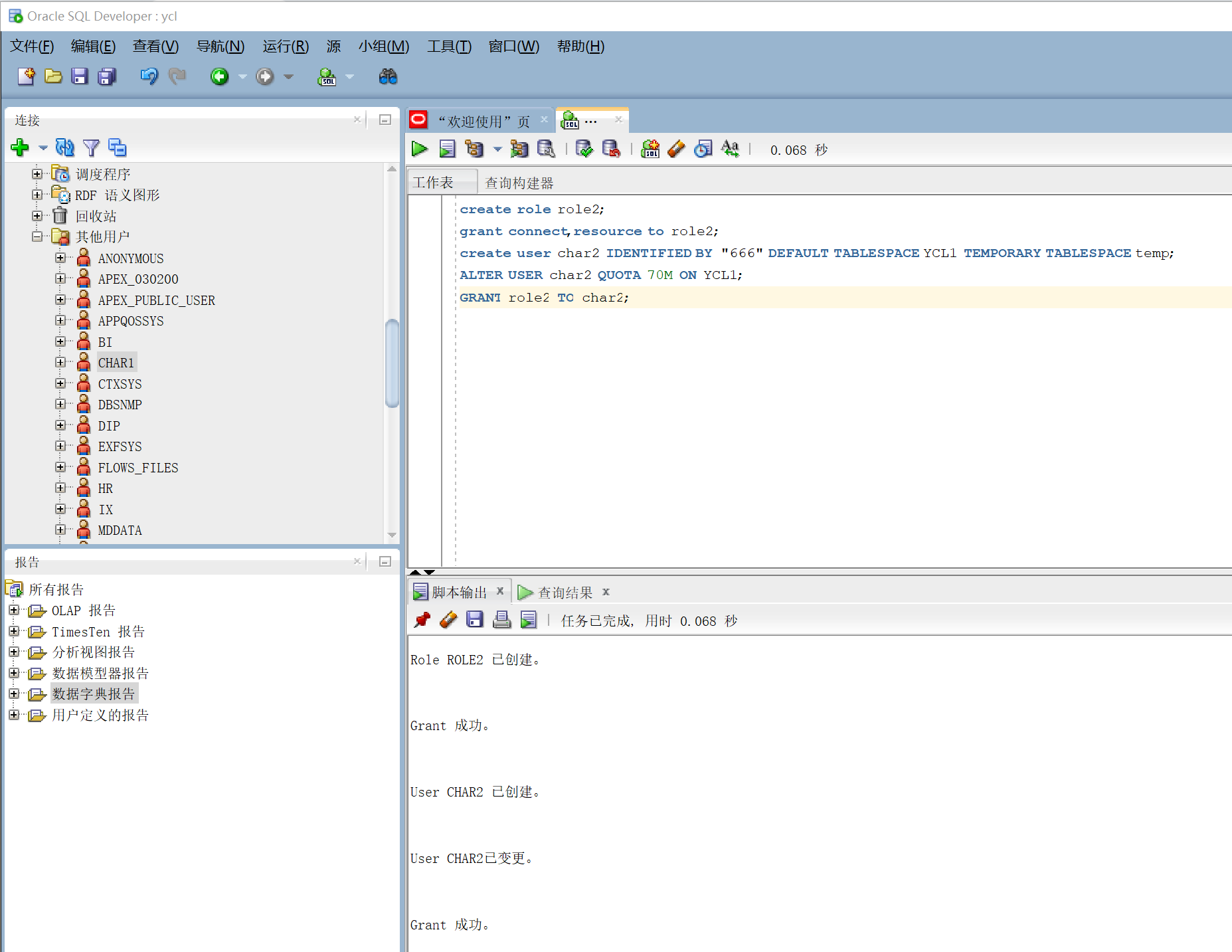
创建角色role1，并将create view,connect,resource权限授予role1；再创建用户char1，将角色qhl1授权给用户qhl\_1

1. *-- 创建角色role1*
2. create role role1;
3. *-- 将create view,connect,resource权限授予role1*
4. grant create view,connect,resource to myrole;
5. *-- 创建用户char1（默认使用表空间YCL1）*
6. create user char1 IDENTIFIED BY 666 DEFAULT TABLESPACE YCL1 TEMPORARY TABLESPACE temp;
7. *-- 分配70M空间给char1*
8. ALTER USER char1 QUOTA 70M ON YCL1;
9. *-- 并将角色qhl1授权给用户qhl\_1*
10. GRANT role1 TO char1;



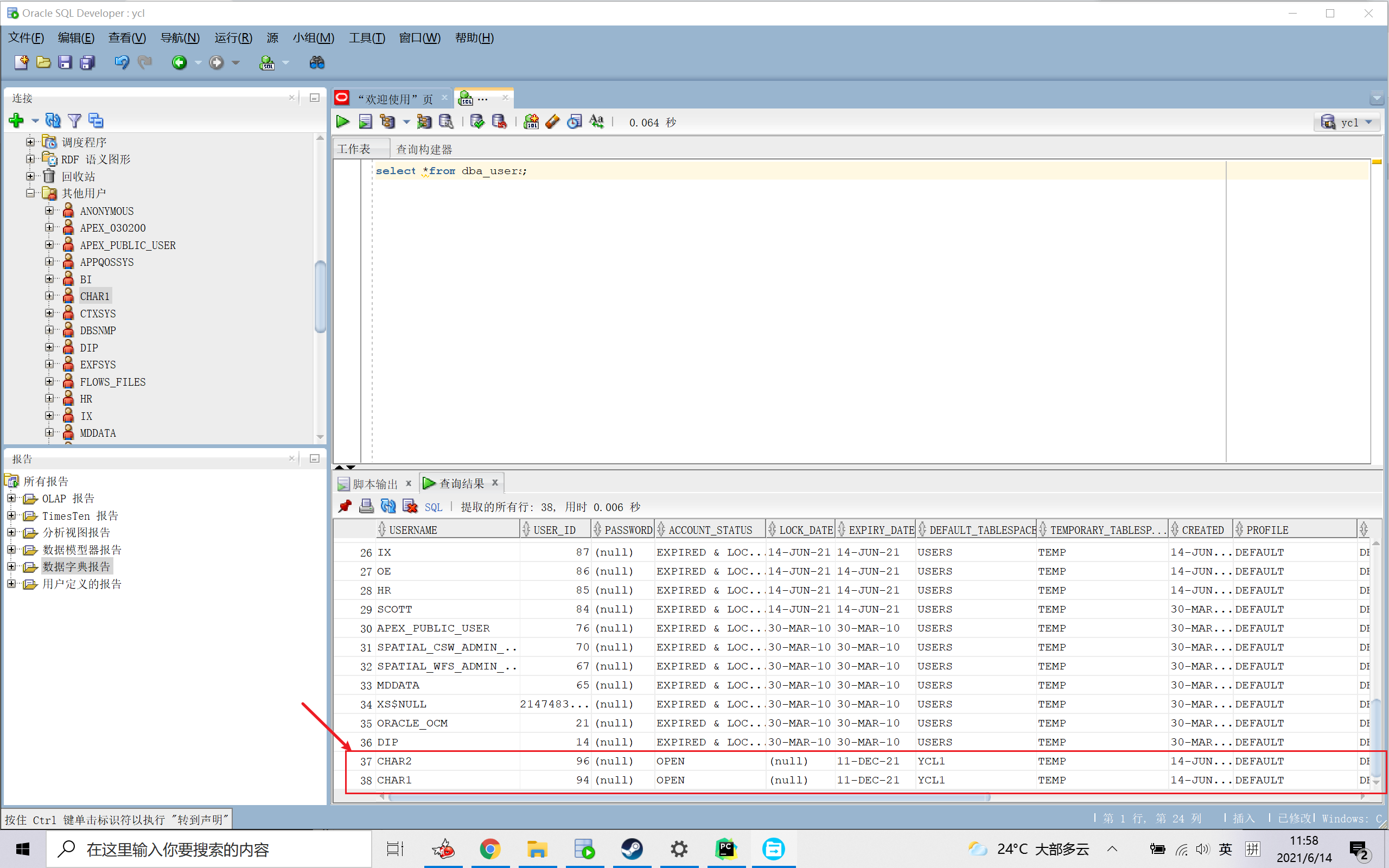
创建角色role2，并将connect,resource权限授予role2；再创建用户char2，将角色qhl2授权给用户qhl\_2

1. *-- 创建角色role2*
2. create role role2;
3. *-- 将connect,resource权限授予role2*
4. grant connect,resource to role2;
5. *-- 创建用户char2（默认使用表空间YCL1）*
6. create user char2 IDENTIFIED BY 666 DEFAULT TABLESPACE YCL1 TEMPORARY TABLESPACE temp;
7. *-- 分配70M空间给char2*
8. ALTER USER char2 QUOTA 70M ON YCL1;
9. *-- 并将角色role2授权给用户char2*
10. GRANT role2 TO char2;



查看已经创建的用户

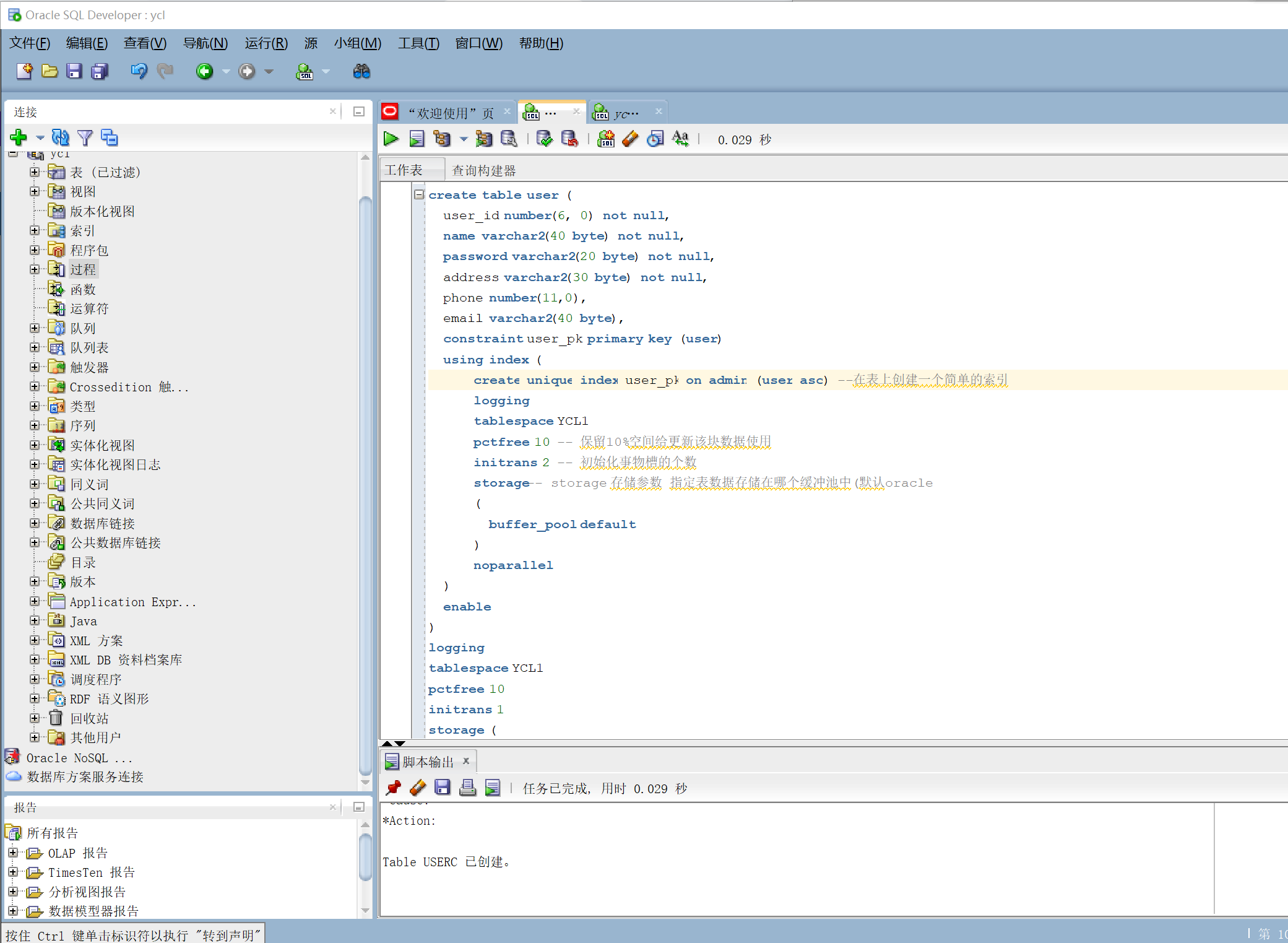
1. select \*from dba\_user;



## 3.设计表

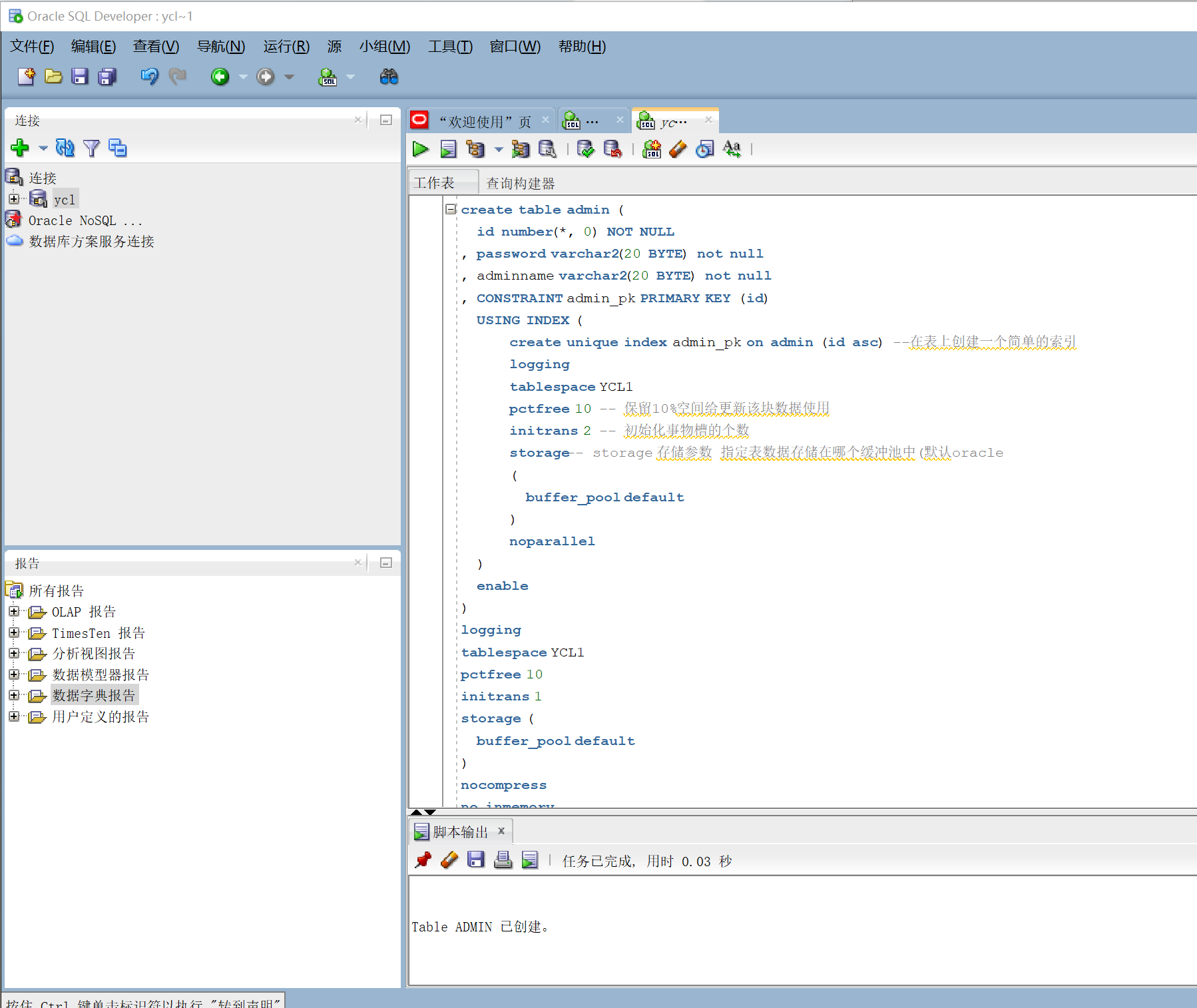
### 3.1设计用户表

1. create table user (
2. user\_id number(6, 0) not null,
3. name varchar2(40 byte) not null,
4. password varchar2(20 byte) not null
5. address varchar2(30 byte) not null
6. phone number(11,0),
7. email varchar2(40 byte),
8. constraint user\_pk primary key (userid)
9. using index (
10. create unique index admin\_pk on admin (userid asc) *--在表上创建一个简单的索引*
11. logging
12. tablespace YCL1
13. pctfree 10 *-- 保留10%空间给更新该块数据使用*
14. initrans 2 *-- 初始化事物槽的个数*
15. storage*-- storage 存储参数 指定表数据存储在哪个缓冲池中(默认oracle*
16. (
17. buffer\_pool default
18. )
19. noparallel
20. )
21. enable
22. )
23. logging
24. tablespace YCL1
25. pctfree 10
26. initrans 1
27. storage (
28. buffer\_pool default
29. )
30. nocompress
31. no inmemory
32. noparallel;



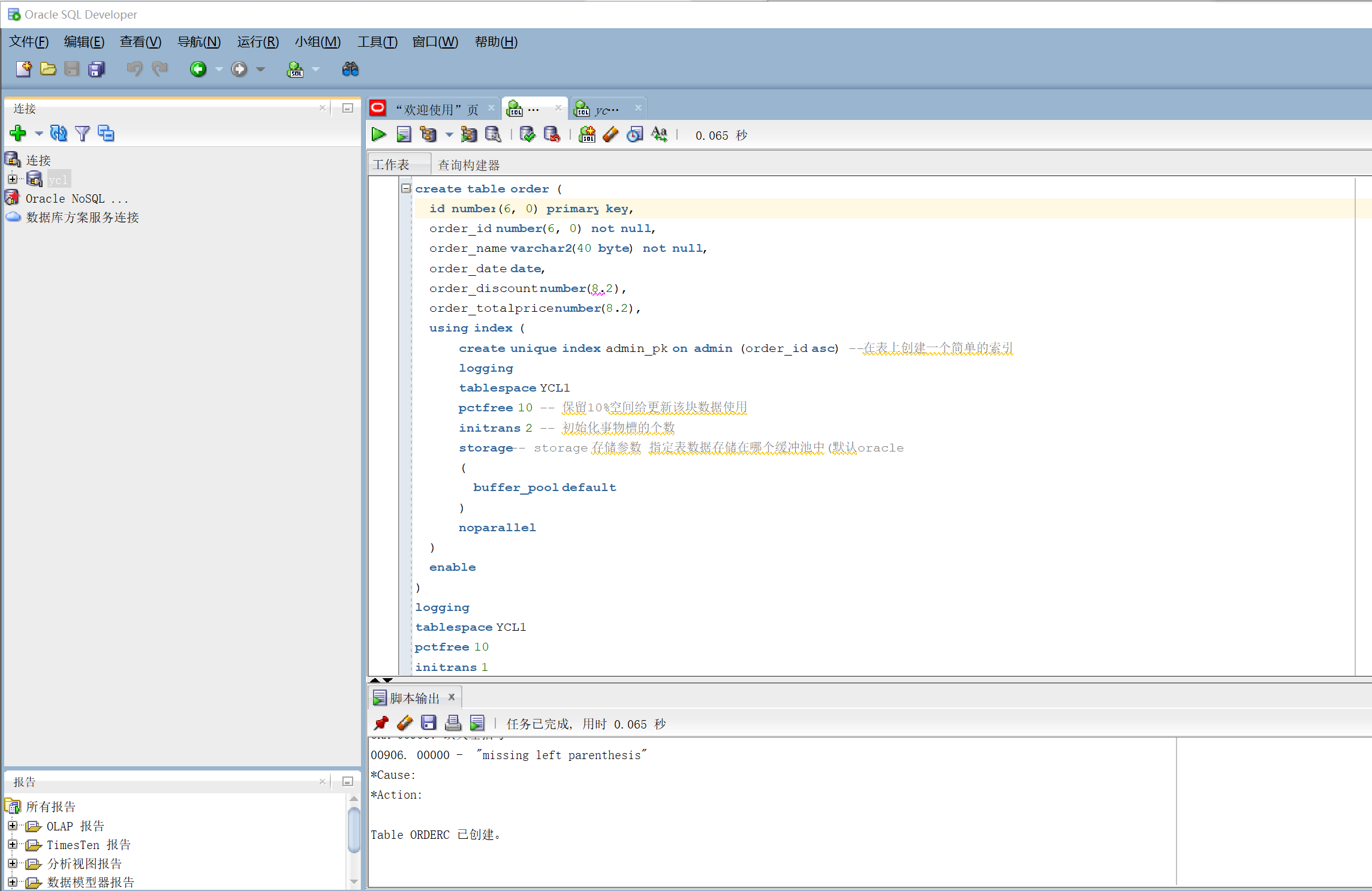
### 3.2设计管理员表

1. create table admin (
2. id number(\*, 0) NOT NULL
3. , password varchar2(20 BYTE) not null
4. , adminname varchar2(20 BYTE) not null
5. , CONSTRAINT admin\_pk PRIMARY KEY (id)
6. USING INDEX (
7. create unique index admin\_pk on admin (id asc) *--在表上创建一个简单的索引*
8. logging
9. tablespace YCL1
10. pctfree 10 *-- 保留10%空间给更新该块数据使用*
11. initrans 2 *-- 初始化事物槽的个数*
12. storage*-- storage 存储参数 指定表数据存储在哪个缓冲池中(默认oracle*
13. (
14. buffer\_pool default
15. )
16. noparallel
17. )
18. enable
19. )
20. logging
21. tablespace YCL1
22. pctfree 10
23. initrans 1
24. storage (
25. buffer\_pool default
26. )
27. nocompress
28. no inmemory
29. noparallel;



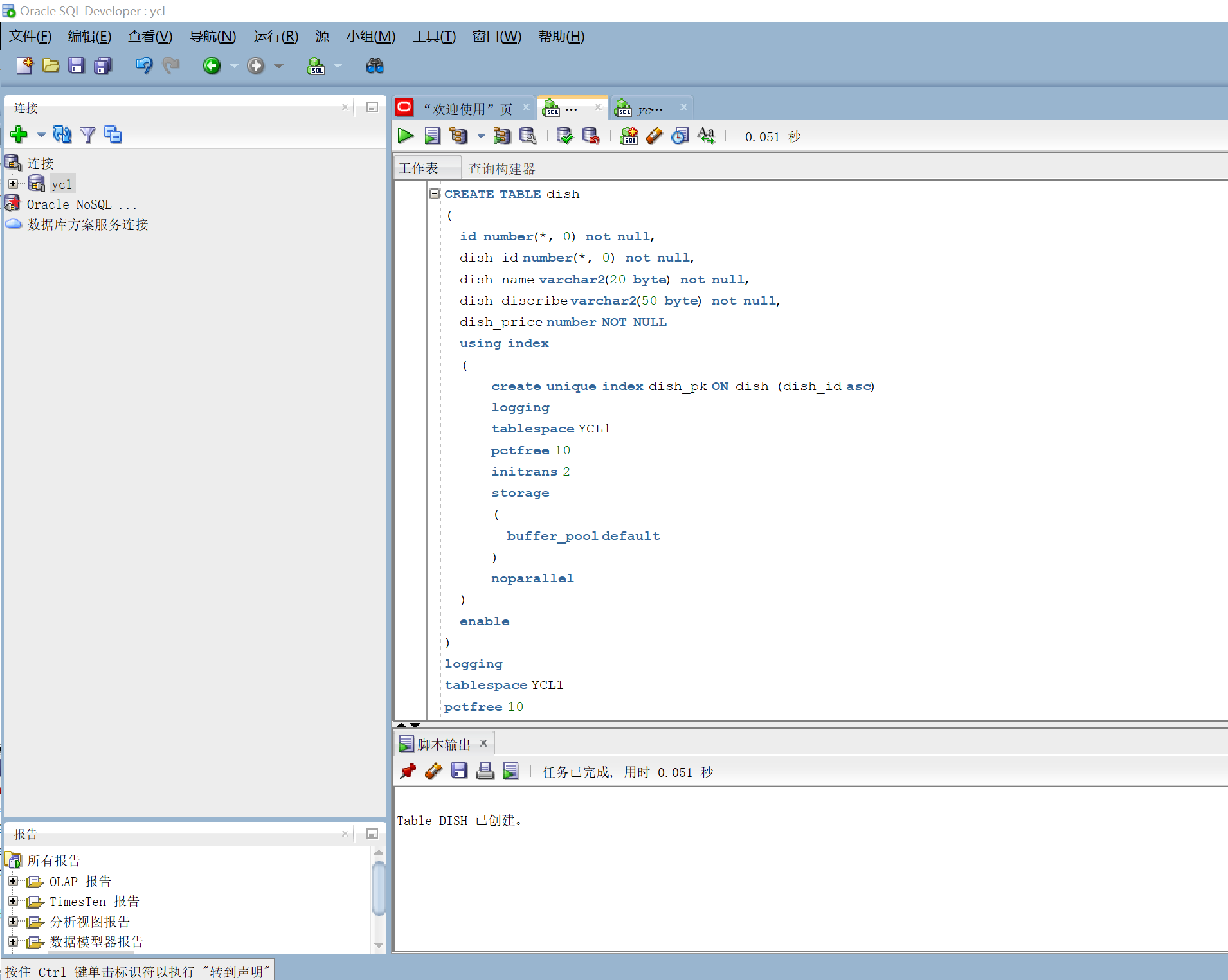
### 3.3设计订单表

1. create table order (
2. id number(6, 0) primary key,
3. order\_id number(6, 0) not null,
4. order\_name varchar2(40 byte) not null,
5. order\_date date,
6. order\_discount number(8.2),
7. order\_totalprice number(8.2),
8. using index (
9. create unique index admin\_pk on admin (order\_id asc) *--在表上创建一个简单的索引*
10. logging
11. tablespace YCL1
12. pctfree 10 *-- 保留10%空间给更新该块数据使用*
13. initrans 2 *-- 初始化事物槽的个数*
14. storage*-- storage 存储参数 指定表数据存储在哪个缓冲池中(默认oracle*
15. (
16. buffer\_pool default
17. )
18. noparallel
19. )
20. enable
21. )
22. logging
23. tablespace YCL1
24. pctfree 10
25. initrans 1
26. storage (
27. buffer\_pool default
28. )
29. nocompress
30. no inmemory
31. noparallel;



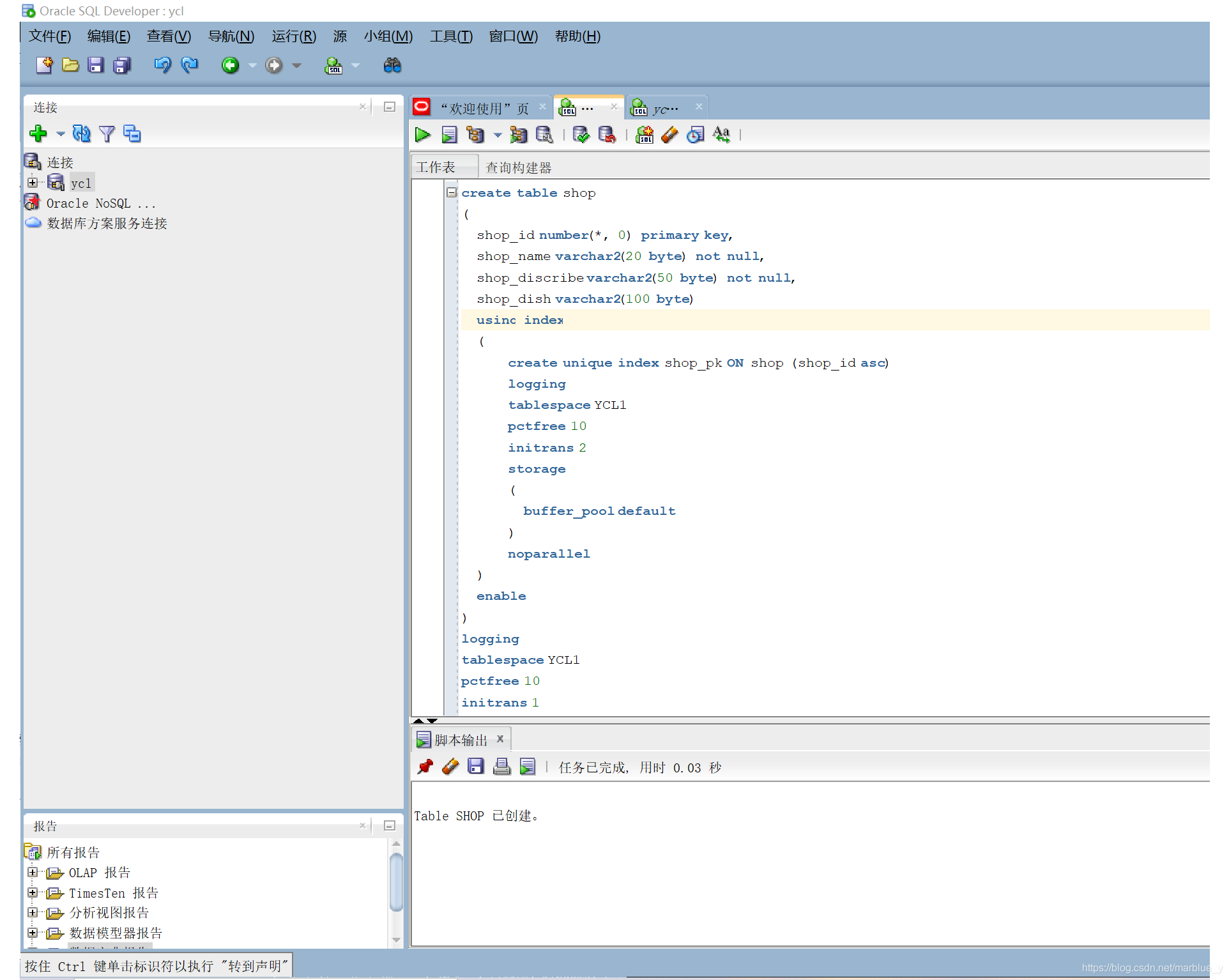
### 3.4设计菜品表

1. CREATE TABLE dish
2. (
3. id number(\*, 0) not null,
4. dish\_id number(\*, 0) not null,
5. dish\_name varchar2(20 byte) not null,
6. dish\_discribe varchar2(50 byte) not null,
7. dish\_price number NOT NULL
8. using index
9. (
10. create unique index dish\_pk ON dish (dish\_id asc)
11. logging
12. tablespace YCL1
13. pctfree 10
14. initrans 2
15. storage
16. (
17. buffer\_pool default
18. )
19. noparallel
20. )
21. enable
22. )
23. logging
24. tablespace YCL1
25. pctfree 10
26. initrans 1
27. storage (
28. buffer\_pool default
29. )
30. nocompress
31. no inmemory
32. noparallel;

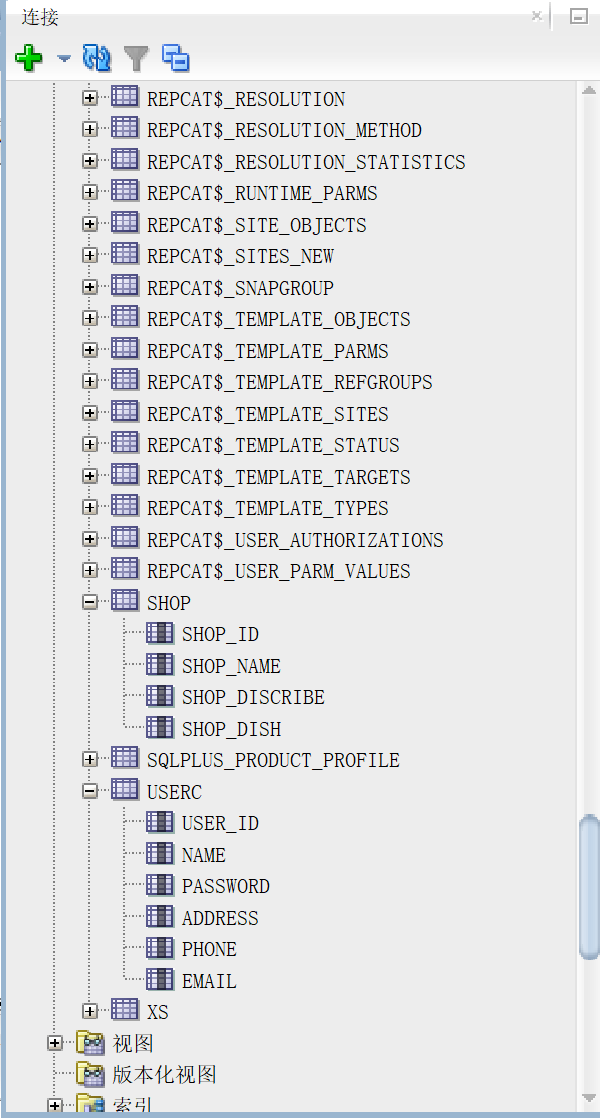
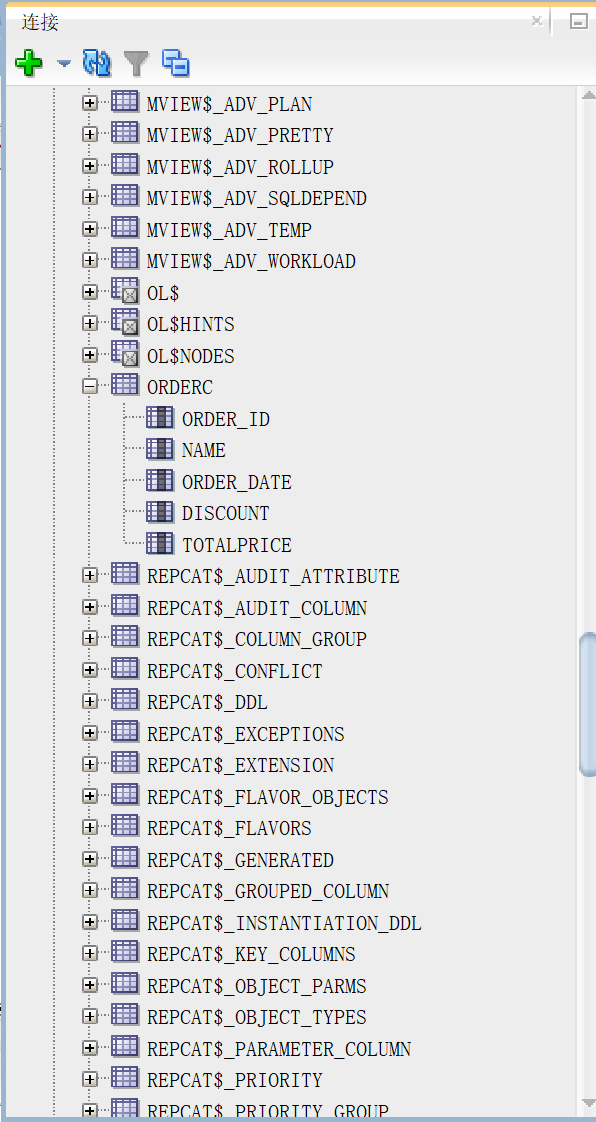
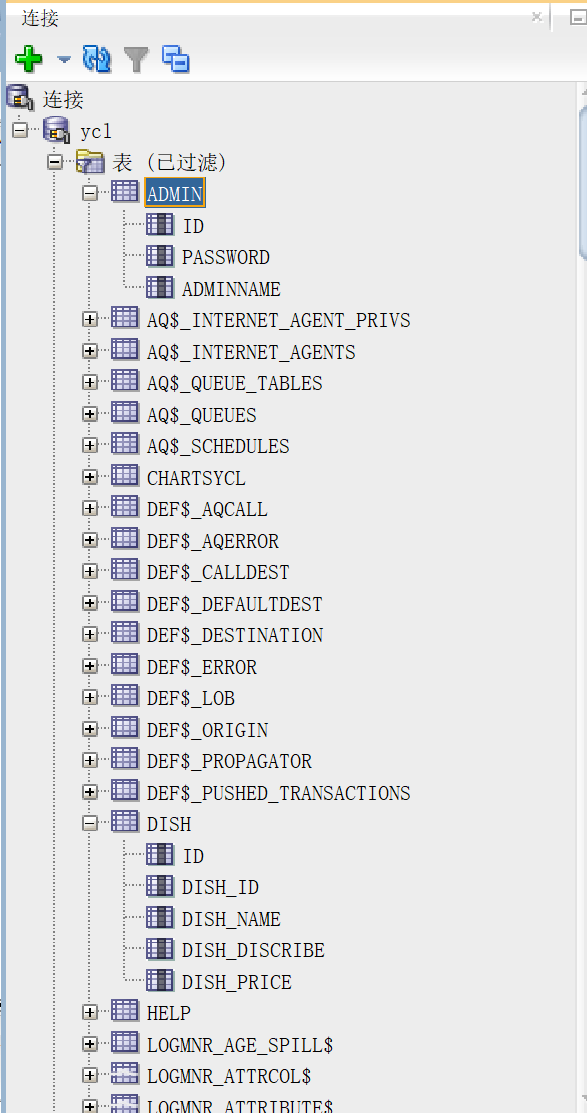


### 3.5设计店铺表

1. create table shop
2. (
3. shop\_id number(\*, 0) primary key,
4. shop\_name varchar2(20 byte) not null,
5. shop\_discribe varchar2(50 byte) not null,
6. shop\_dish varchar2(100 byte)
7. using index
8. (
9. create unique index shop\_pk ON shop (shop\_id asc)
10. logging
11. tablespace YCL1
12. pctfree 10
13. initrans 2
14. storage
15. (
16. buffer\_pool default
17. )
18. noparallel
19. )
20. enable
21. )
22. logging
23. tablespace YCL1
24. pctfree 10
25. initrans 1
26. storage (
27. buffer\_pool default
28. )
29. nocompress
30. no inmemory
31. noparallel;

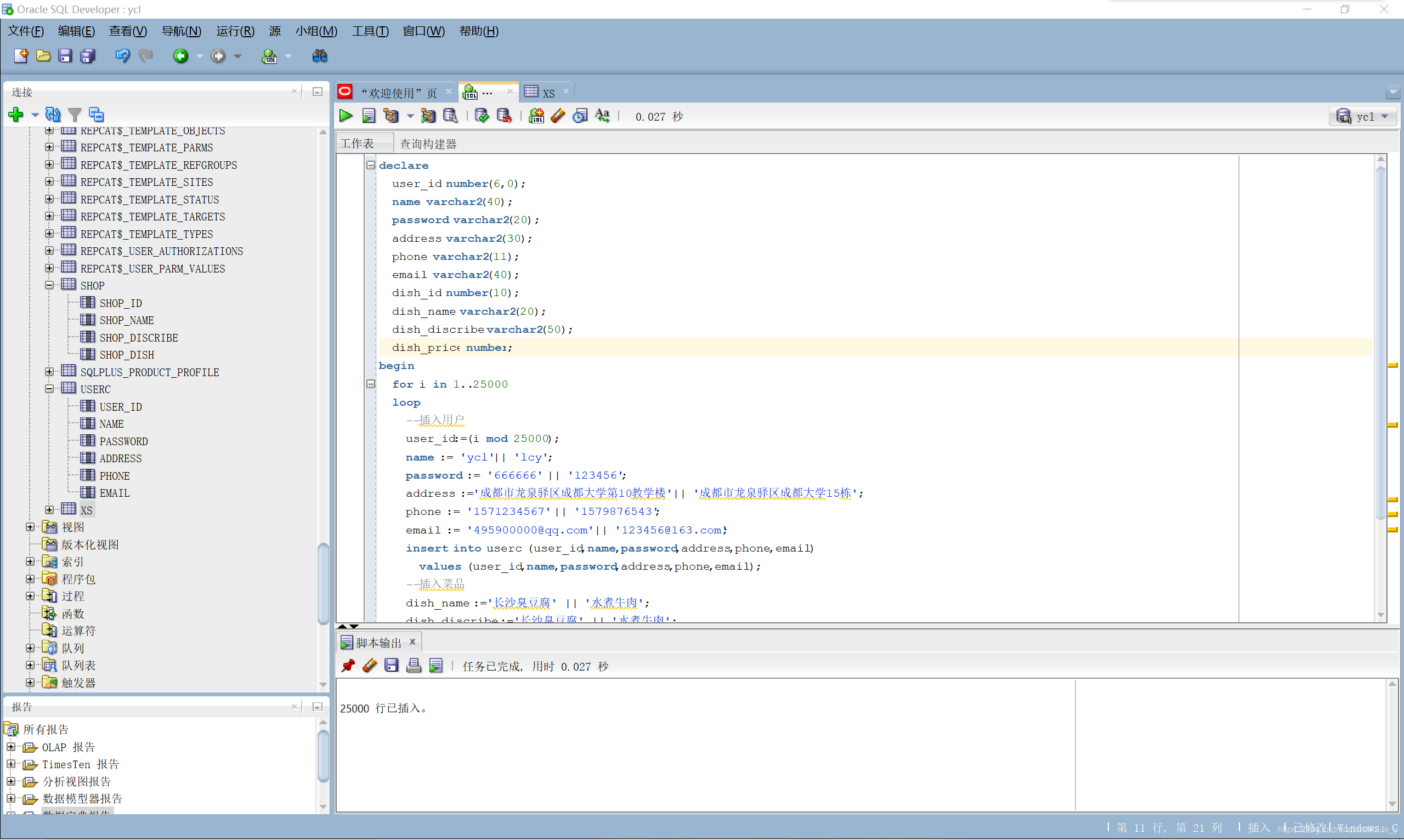


### 3.6查看已经创建好的表



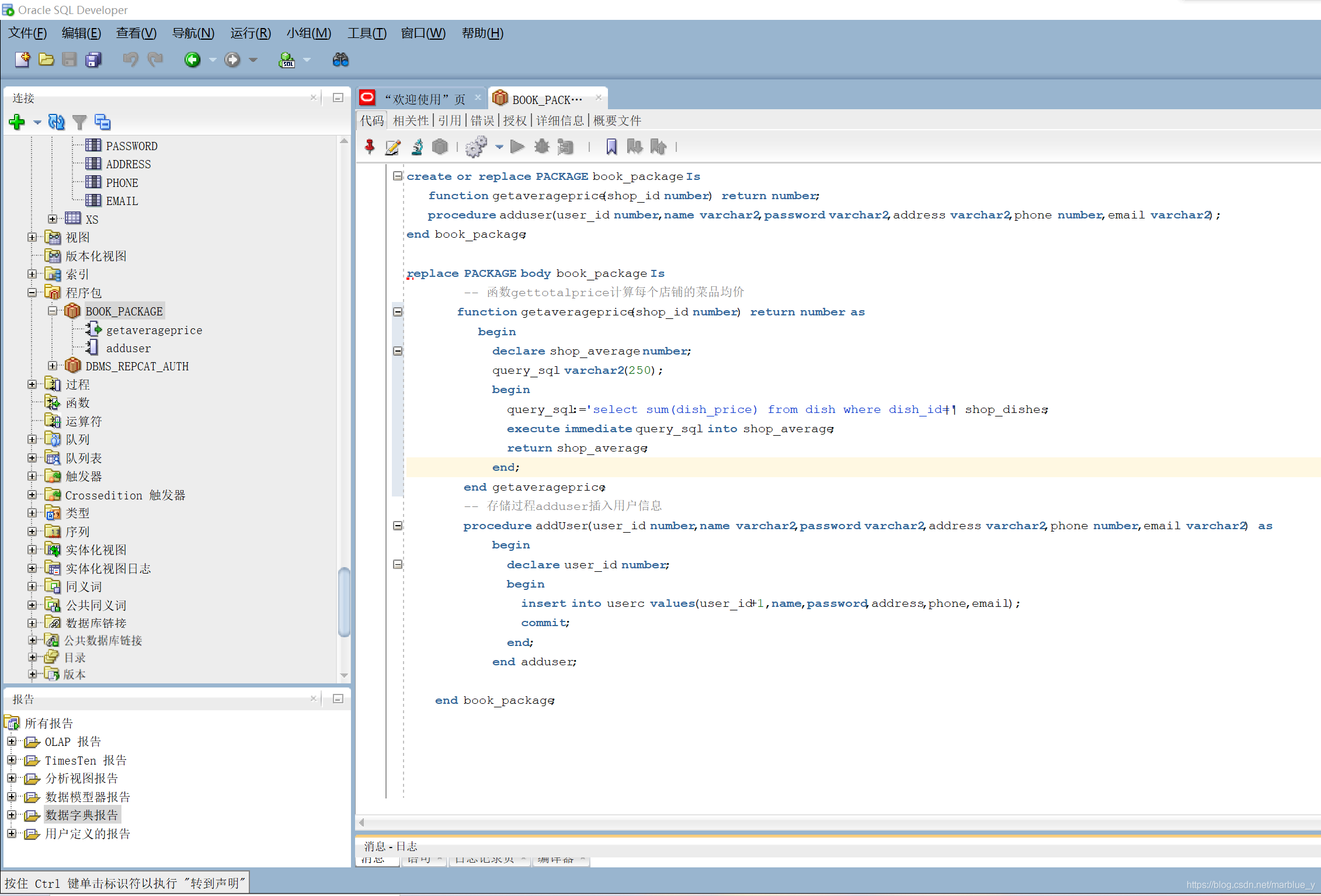
### 3.7插入用户与菜品数据

1. declare
2. user\_id number(6,0);
3. name varchar2(40);
4. password varchar2(20);
5. address varchar2(30);
6. phone varchar2(11);
7. email varchar2(40);
8. dish\_id number(10),
9. dish\_name varchar2(20),
10. dish\_discribe varchar2(50),
11. dish\_price number
12. begin
13. for i in 1..25000
14. loop
15. *--插入用户*
16. user\_id:=(i mod 25000);
17. name := 'ycl'|| 'lcy';
18. password := '666666' || '123456';
19. address :='成都市龙泉驿区成都大学第10教学楼'|| '成都市龙泉驿区成都大学15栋';
20. phone := '1571234567' || '1579876543';
21. email := '495900000@qq.com' || '123456@163.com';
22. insert into userc (user\_id,name,password,address,phone,email)
23. values (user\_id,name,password,address,phone,email);
24. *--插入菜品*
25. dish\_name :='长沙臭豆腐' || '水煮牛肉';
26. dish\_discribe :='长沙臭豆腐' || '水煮牛肉';
27. dish\_price :='20' || '25';
28. insert into dish(dish\_id,dish\_name,dish\_discribe,dish\_price)
29. values (i,dish\_name,dish\_discribe,dish\_price);
30. end loop;
31. end;



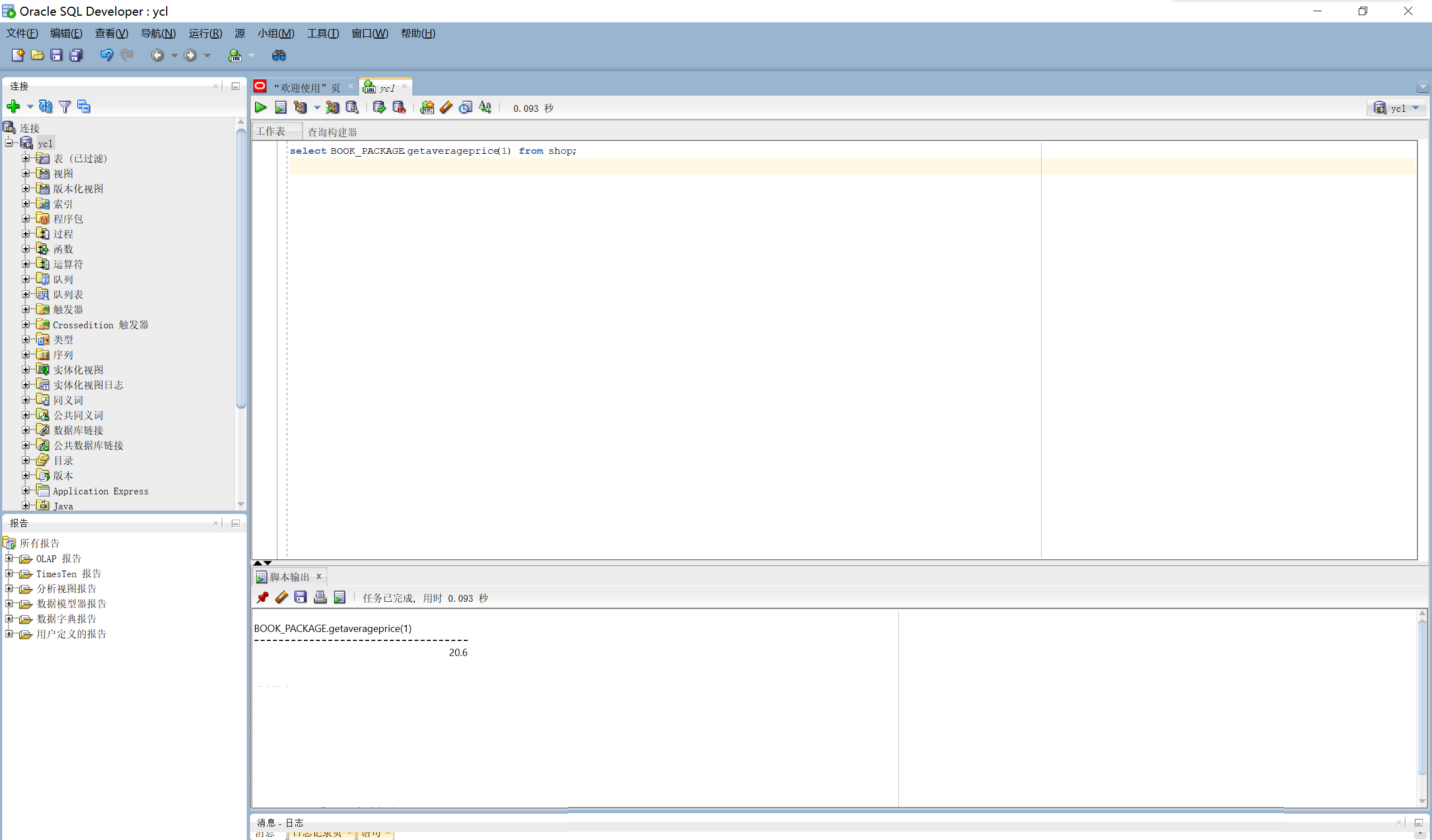
## 4.创建程序包、存储过程、函数执行分析计划

1. create or replace PACKAGE book\_package Is
2. function getaverageprice(shop\_id number) return number;
3. procedure adduser(user\_id number,name varchar2,password varchar2,address varchar2,phone number,email varchar2);
4. end book\_package;
5. create or replace PACKAGE body book\_package Is
6. *-- 函数gettotalprice计算每个店铺的菜品均价*
7. function getaverageprice(shop\_id number) return number as
8. begin
9. declare shop\_average number;
10. query\_sql varchar2(250);
11. begin
12. query\_sql:='select sum(dish\_price) from dish where dish\_id=' || shop\_dishes;
13. execute immediate query\_sql into shop\_average;
14. return shop\_average;
15. end;
16. end getaverageprice;
17. *-- 存储过程adduser插入用户信息*
18. procedure addUser(user\_id number,name varchar2,password varchar2,address varchar2,phone number,email varchar2)
19. begin
20. declare user\_id number;
21. begin
22. insert into userc values(user\_id+1,name,password,address,phone,email);
23. commit;
24. end;
25. end adduser;
26. end book\_package;



存储过程、函数执行分析

1. select BOOK\_PACKAGE.getaverageprice(1) from shop;



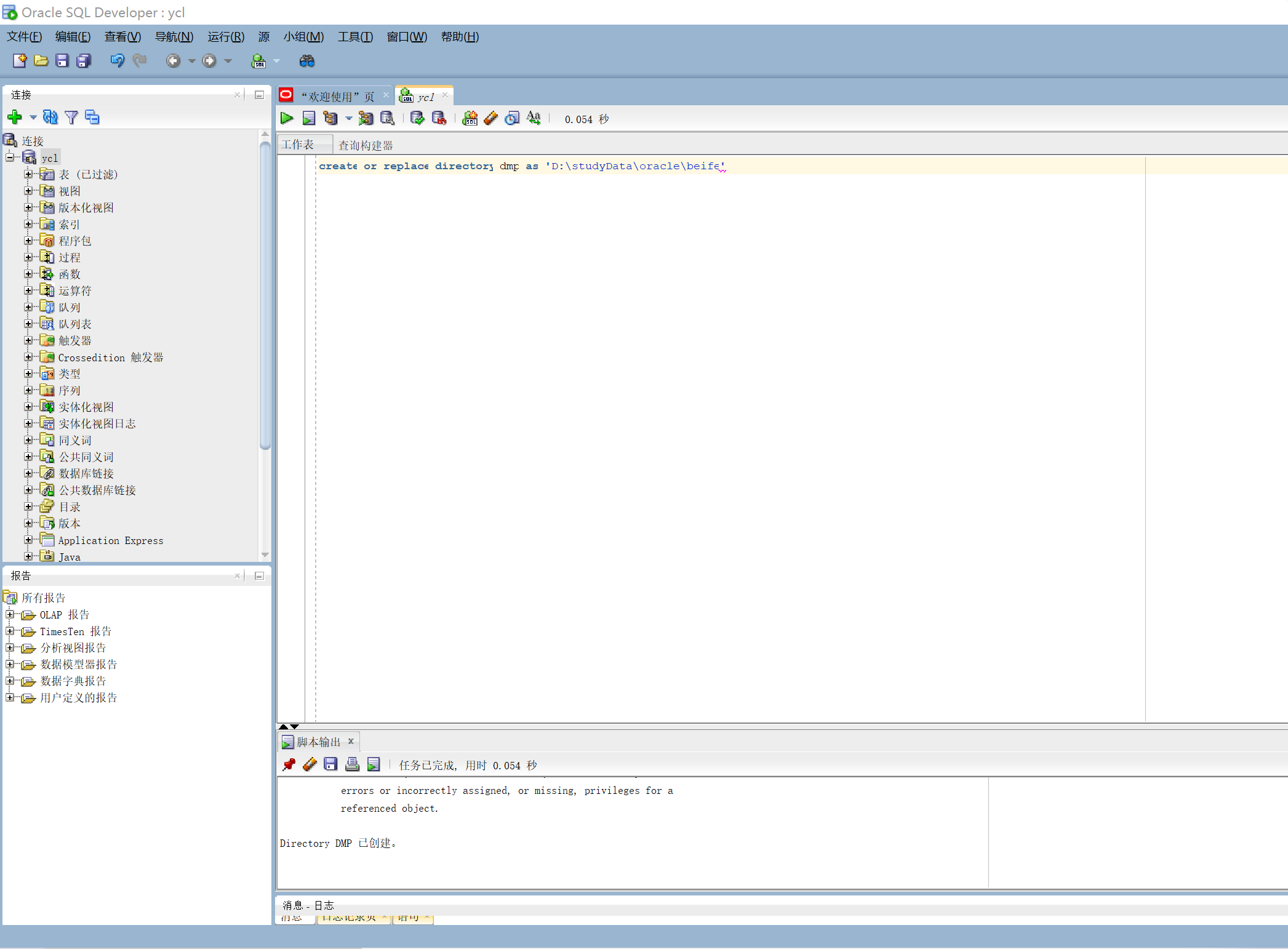
使用存储过程adduser插入用户数据

1. set serveroutput on
2. declare
3. begin
4. BOOK\_PACKAGE.adduser(20,'ycl','666666','地球',110,'163.com');
5. end

## 5.设计备份方案

建立备份目录

1. create or replace directory dmp as 'D:\studyData\oracle\beifen'



1. @echo off
2. set BACKUPDATE=%date:~0,4%%date:~5,2%%date:~8,2%
3. set USER=system
4. set PASSWORD=root
5. set DATABASE=ycl
6. beifen'%USER%/%PASSWORD%' directory=dmp dumpfile=data\_%BACKUPDATE%.dmp schemas=%DATABASE%
7. exit

执行以上批处理文件 ，将dmp文件保存到指定目录移动批处理命令,将D:\studyData\oracle\beifen的所有文件移动到D:\studyData\oracle\oracleData下

1. move D:\studyData\oracle\beifen\*.\*  D:\studyData\oracle\oracleData

删除批处理命令：

1. forfiles /p D:\studyData\oracle\beifen /m \*.\* /d -10 /c "cmd /c del @file"

以Windows中的计划任务程序来触发备份脚本

