



# ORACLE 12C的安装配置

数据库原理与应用

# 1. 安装概览

- Oracle主要用到的3个工具：sqlplus(命令行工具)，SQL developer(图形界面工具)和 Database Configuration Assistant(配置工具)



```
管理员: 命令提示符 - sqlplus
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation。保留所有权利。

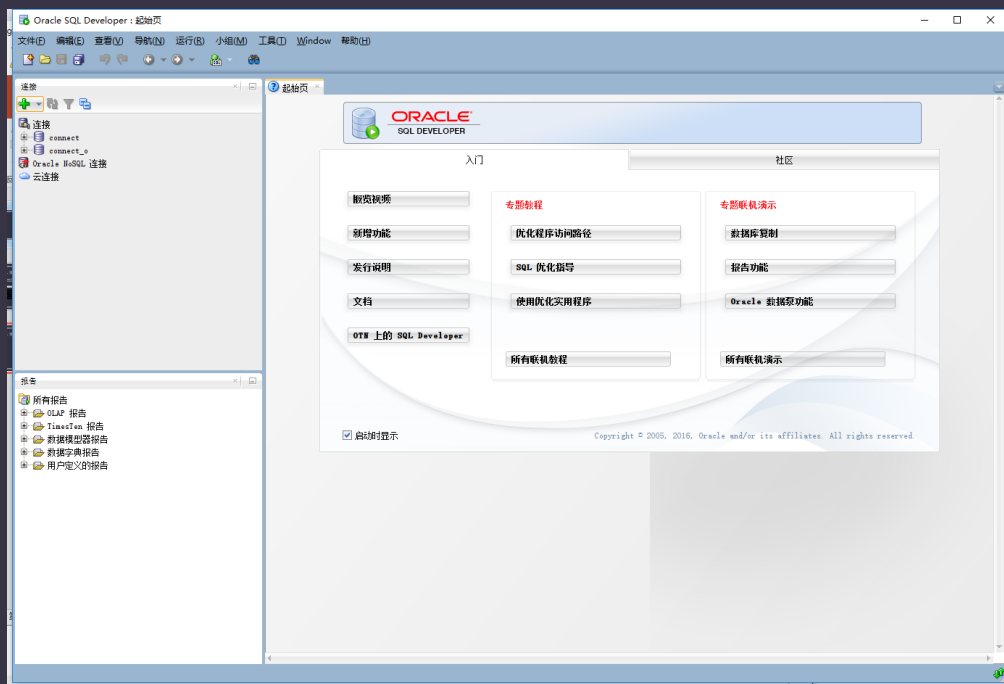
C:\WINDOWS\system32>sqlplus

SQL*Plus: Release 12.2.0.1.0 Production on Wed Jun 7 13:07:14 2017

Copyright (c) 1982, 2016, Oracle. All rights reserved.

Enter user-name:
```

# 1. 安装概览



## 2. 安装注意事项

### 1. 在选择类型的时候选择“桌面类”

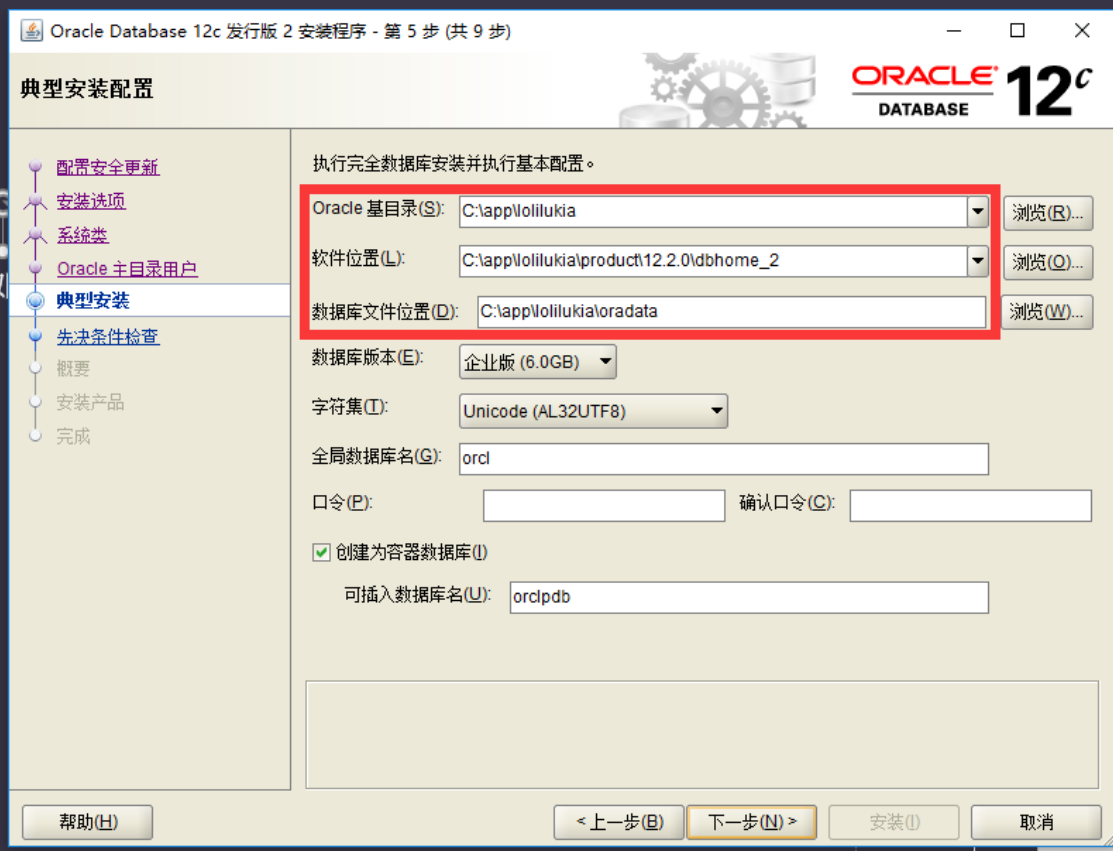


## 2. 安装注意事项

2. 建议在oracle主目录用户选择时选择“创建新windows用户”，并且需要设置与当前windows用户不同的用户名，并且请牢记口令，在后续创建连接的时候会作为“主目录用户口令”用到

## 2. 安装注意事项

3. 典型安装中，尽量不要修改默认的软件位置，如果一定要改，只能修改oracle基目录



Oracle Database 12c 发行版 2 安装程序 - 第 5 步 (共 9 步)

典型安装配置

执行完全数据库安装并执行基本配置。

Oracle 基目录(S): C:\app\lolilukia 浏览(B)...

软件位置(L): C:\app\lolilukia\product12.2.0\dbhome\_2 浏览(O)...

数据库文件位置(D): C:\app\lolilukia\oradata 浏览(W)...

数据库版本(E): 企业版 (6.0GB)

字符集(I): Unicode (AL32UTF8)

全局数据库名(G): orcl

口令(P): 确认口令(C):

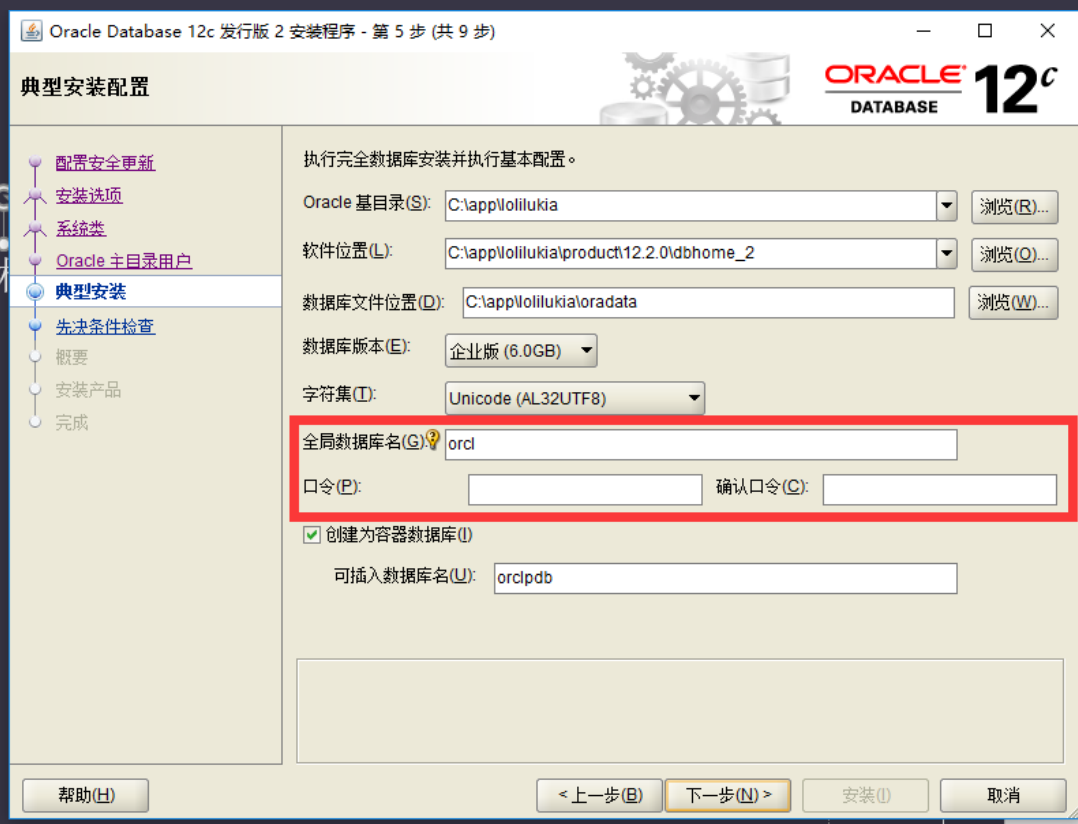
☒ 创建为容器数据库(U)

可插入数据库名(U): orclpdb

帮助(H) < 上一步(B) 下一步(N) > 安装(I) 取消

## 2. 安装注意事项

4. 全局数据库名默认为orcl，这与之后SID属性值相吻合，管理口令同样需要牢记，忘记的话可以通过sqlplus进行找回



## 2. 安装注意事项

5. 然后进行安装，如果不慎断电或中止，可以打开前面提到的配置工具Database Configuration Assistant删除数据库进行再次安装





# 3. 问题及相关解决方案

配置Database Configuration Assistant失败

也就是生成数据库实例的时候出错，先开启监听lsnrctl start，然后手动打开Database Configuration Assistant，重新创建



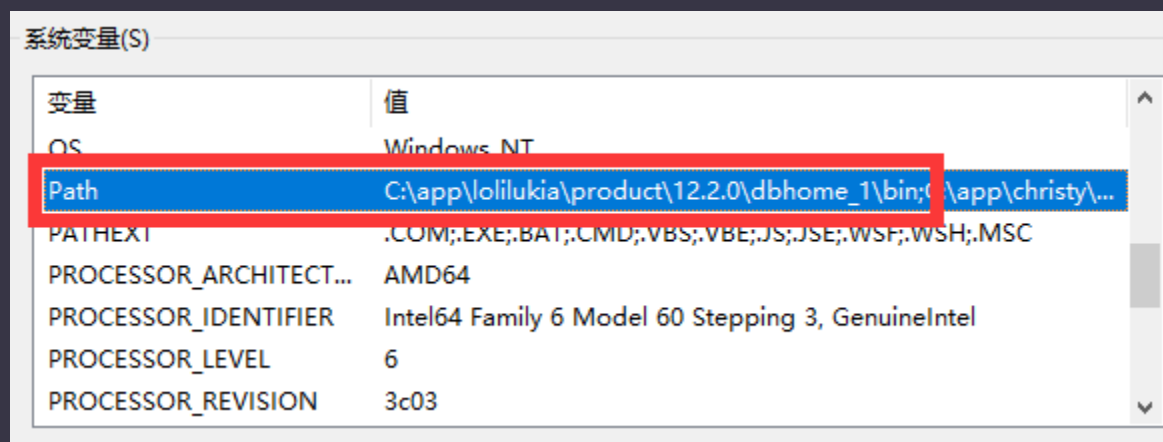
# 3. 问题及相关解决方案

## Sqlplus的使用

打开管理员命令行，输入sqlplus

1. 经典问题：sqlplus命令找不到

解决方案： 将oracle的bin目录添加到系统变量path中  
形如C:\app\xxx\product\12.2.0\dbhome\_1\bin



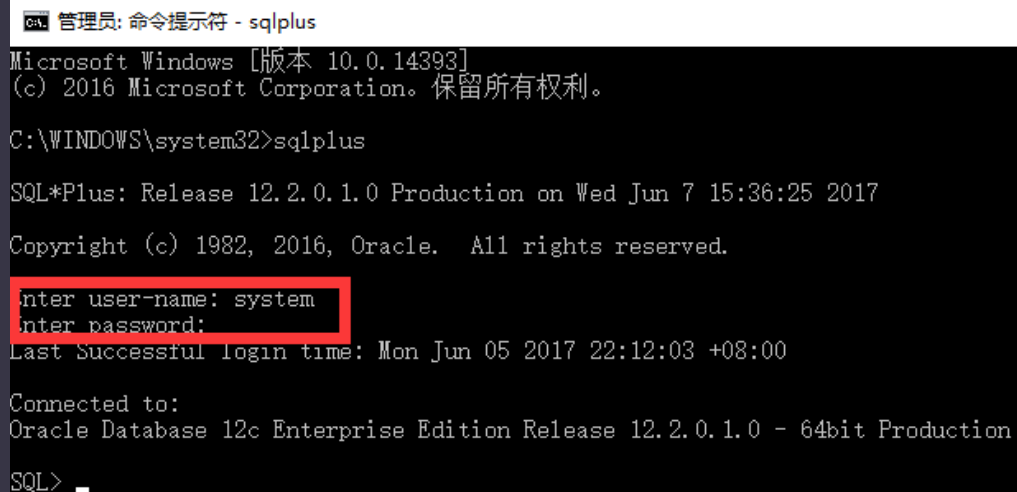
# 3. 问题及相关解决方案

## Sqlplus的使用

2. 使用普通管理员的账号，登陆

用户名: system

密码: （前面设置的管理口令）



```
管理员: 命令提示符 - sqlplus
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation. 保留所有权利。

C:\WINDOWS\system32>sqlplus

SQL*Plus: Release 12.2.0.1.0 Production on Wed Jun 7 15:36:25 2017

Copyright (c) 1982, 2016, Oracle. All rights reserved.

Enter user-name: system
Enter password:
Last Successful login time: Mon Jun 05 2017 22:12:03 +08:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
SQL> _
```

# 3. 问题及相关解决方案

## Sqlplus的使用

### 3. 忘记管理口令

1) 无密码登陆 sqlplus /nolog

2) 连接sysdba并且修改密码

conn /as sysdba

alter user system identified by 1234; (这里1234是新密码)

```
C:\WINDOWS\system32>sqlplus /nolog
```

```
SQL*Plus: Release 12.2.0.1.0 Production on Wed Jun 7 15:50:31 2017
```

```
Copyright (c) 1982, 2016, Oracle. All rights reserved.
```

```
SQL> conn /as sysdba
```

```
Connected.
```

```
SQL> alter user system identified by 1234;
```

```
User altered.
```

# 3. 问题及相关解决方案

如果sqlplus无法正常运行

(1) 首先检查该数据库实例是否存在

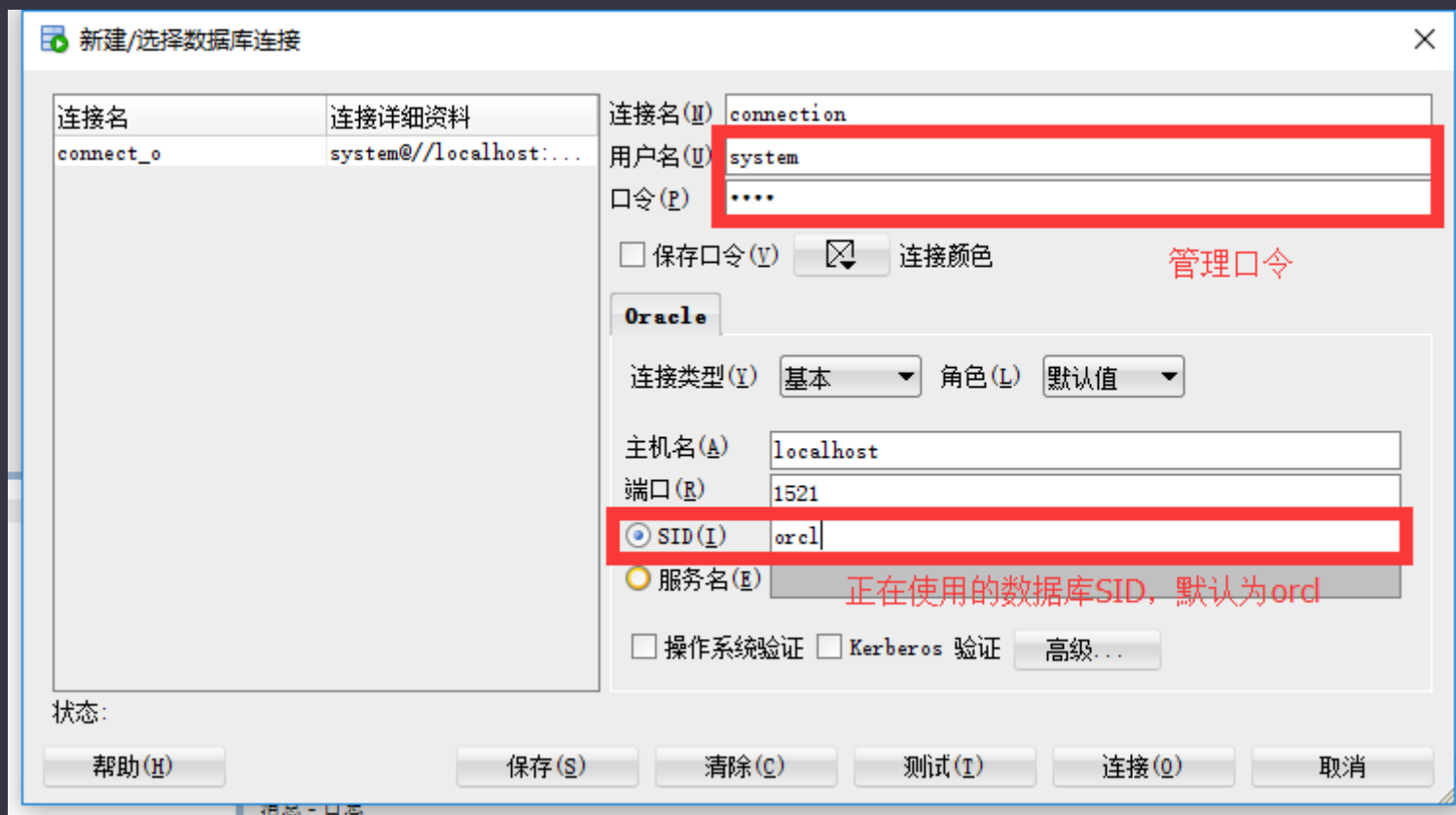
方法：在服务中检查OracleServiceORCL是否存在（如果在安装的时候没有修改默认数据库名），如果存在，则说明实例存在；如果不存在，则打开Database Configuration Assistant创建数据库

(2) 如果问题未解决，将SID设置成要访问的数据库

打开命令行输入默认为： `set ORACLE_SID=orcl`

# 4. SQL Developer

## 1. 建立连接



# 4. SQL Developer

## 2. 导入sql文件

sql文件包含了一系列sql语句，直接执行sql文件可以将大量的数据库操作快速而可重复执行

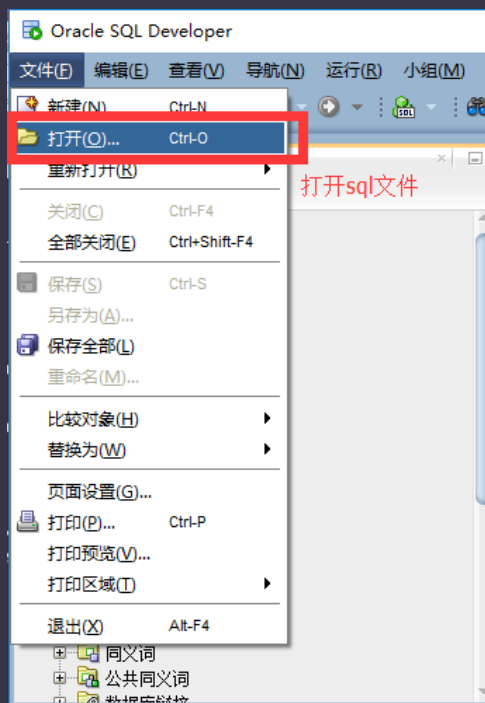
sqlplus可以进行少量的sql语句执行，但不适合执行大量的语句，尤其当文件中sql语句出现问题时，sqlplus无法定位问题的位置

建议使用sql developer来对数据库进行大规模的操作

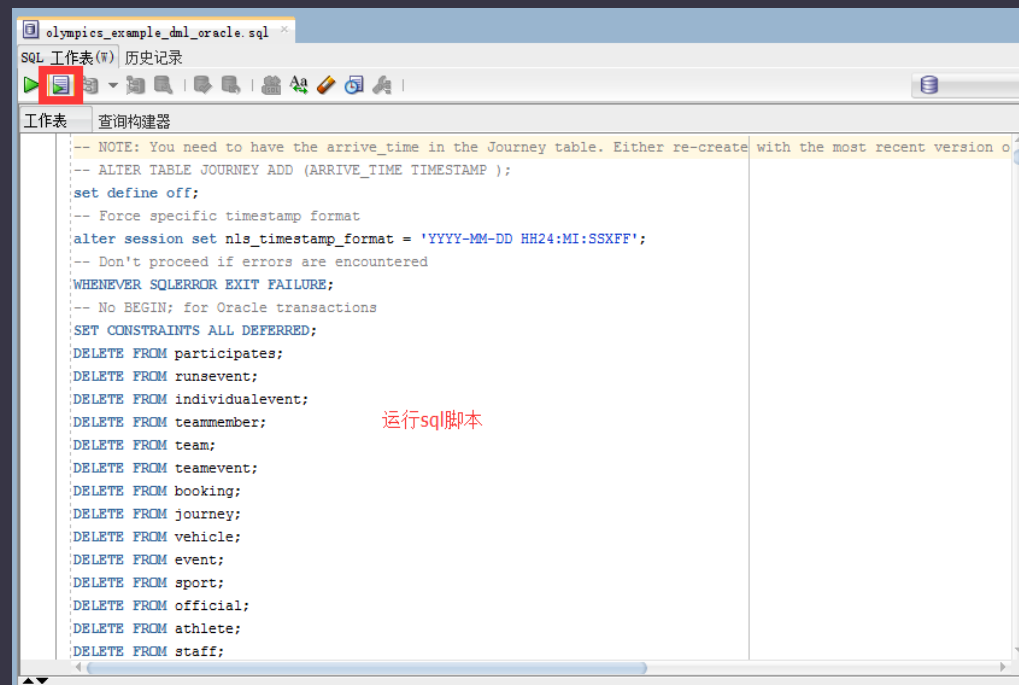
# 4. SQL Developer

## 2. 导入sql文件

### (1) 打开sql文件



### (2) 执行结果





# 4. SQL Developer

## 3. 对数据的图形界面操作（增删改查）

测试时使用

### （1）查

olympics\_example\_dml\_oracle.sql MEMBER

列 数据 约束条件 授权 统计信息 触发器 闪回 相关性 详细资料 分区 索引 SQL

排序: 过滤器: 操作...

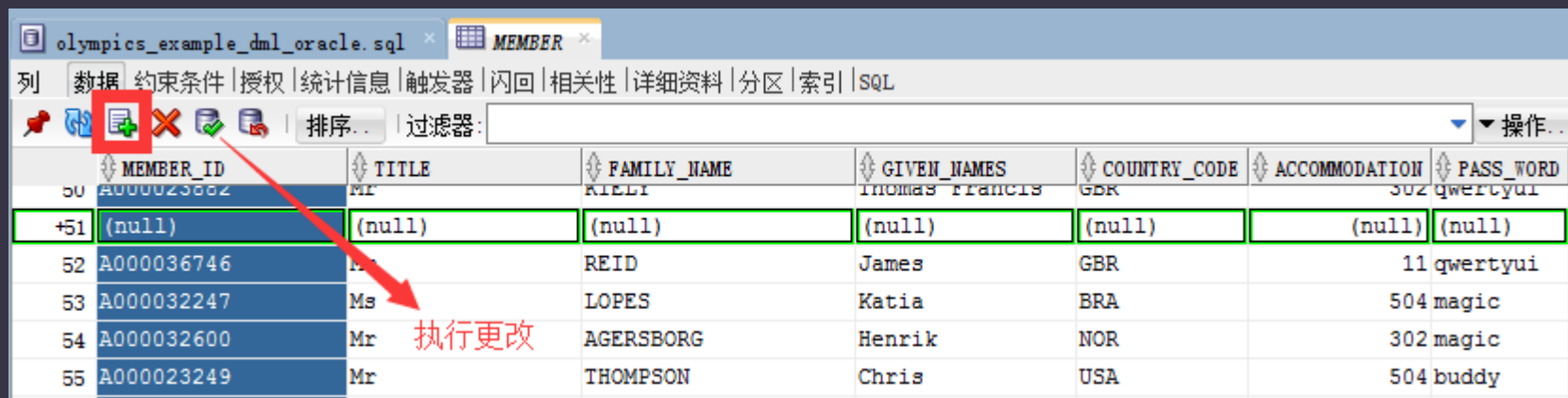
	MEMBER_ID	TITLE	FAMILY_NAME	GIVEN_NAMES	COUNTRY_CODE	ACCOMMODATION	PASS_WORD
1	A000037555	Mr	MAHONY	Harold Sergerson	ZZX		11 lover
2	A000028752	Mr	PEREZ MU桂Z	Alfonso	ESP	503	xxxxxxx
3	A000033482	Mr	WEST	Allen Tarwater	USA	302	xxxxxxx
4	A000022180	Ms	REVE JIMENEZ	Odalys	CUB		11 777777
5	A000042040	Mr	KURKA	Jan	TCH	302	777777
6	A000043387	Mr	MOEL	J.	USA	503	canada
7	A000040101	Mr	BARTHELS	Wolfgang	EUA		11 canada
8	A000035843	Mr	ROMAN	Mauro	ITA	302	canada
9	A000032144	Mr	H腰NER	Reinhard	GDR		11 nathan

# 4. SQL Developer

## 3. 对数据的图形界面操作（增删改查）

测试时使用

### （2）增



The screenshot shows the SQL Developer interface with a table named 'MEMBER'. The table has columns: MEMBER\_ID, TITLE, FAMILY\_NAME, GIVEN\_NAMES, COUNTRY\_CODE, ACCOMMODATION, and PASS\_WORD. A new row is being added, highlighted in green, with values: (null), (null), (null), (null), (null), (null), (null). A red arrow points to the '执行更改' (Execute Changes) button in the toolbar, indicating the next step to save the new row.

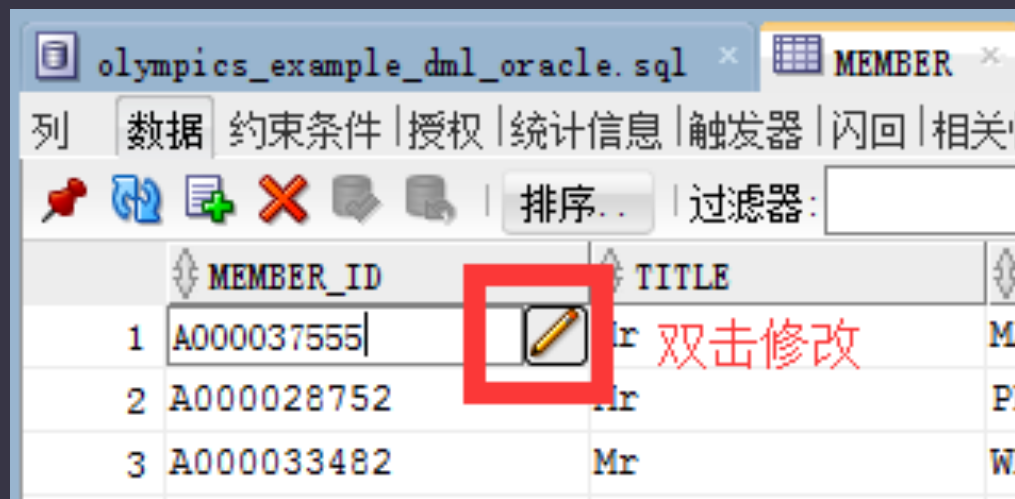
	MEMBER_ID	TITLE	FAMILY_NAME	GIVEN_NAMES	COUNTRY_CODE	ACCOMMODATION	PASS_WORD
50	A000023882	Mr	KIELI	THOMAS FRANCIS	GBR	302	qwertyui
+51	(null)	(null)	(null)	(null)	(null)	(null)	(null)
52	A000036746	Mr	REID	James	GBR	11	qwertyui
53	A000032247	Ms	LOPES	Katia	BRA	504	magic
54	A000032600	Mr	AGERSBORG	Henrik	NOR	302	magic
55	A000023249	Mr	THOMPSON	Chris	USA	504	buddy

# 4. SQL Developer

## 3. 对数据的图形界面操作（增删改查）

测试时使用

（3）改

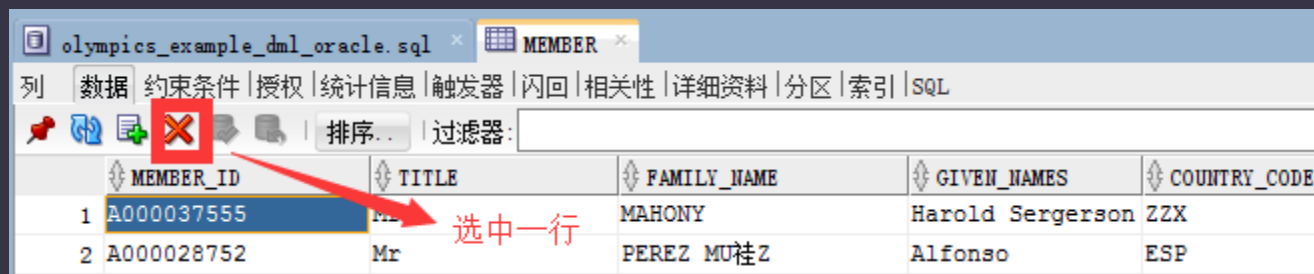


# 4. SQL Developer

## 3. 对数据的图形界面操作（增删改查）

测试时使用

### （4）删

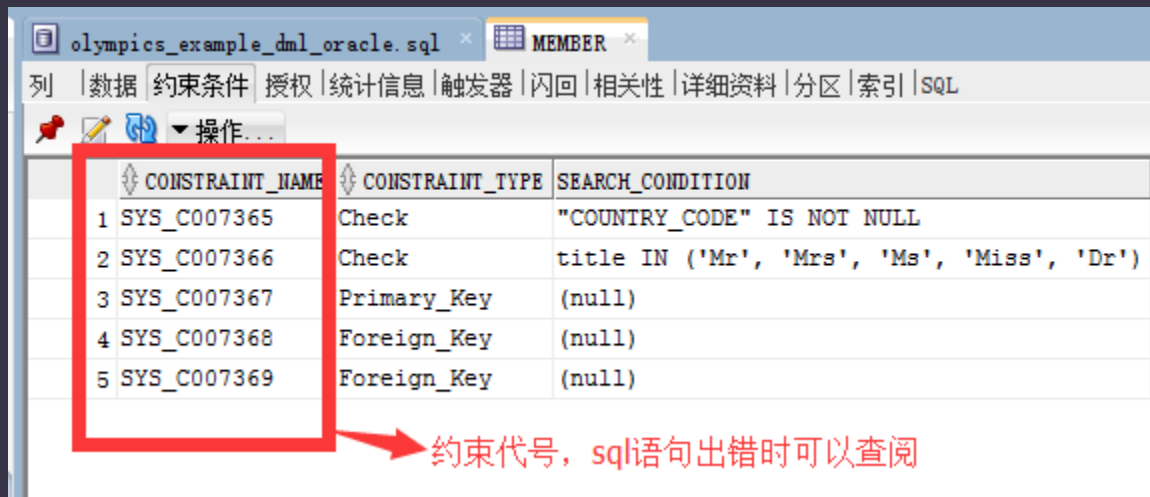


The screenshot shows the SQL Developer interface with the 'olympics\_example\_dml\_oracle.sql' file open. The 'MEMBER' table is selected, and the '数据' (Data) tab is active. The toolbar shows the delete icon (X) highlighted with a red box. A red arrow points from the delete icon to the first row of the table, which is also highlighted. The text '选中一行' (Select one row) is written in red next to the arrow.

	MEMBER_ID	TITLE	FAMILY_NAME	GIVEN_NAMES	COUNTRY_CODE
1	A000037555	Mr	MAHONY	Harold Sergerson	ZZX
2	A000028752	Mr	PEREZ MU桂Z	Alfonso	ESP

# 4. SQL Developer

## 4. 查看约束



olympics\_example\_dml\_oracle.sql x MEMBER x

列 | 数据 | 约束条件 | 授权 | 统计信息 | 触发器 | 闪回 | 相关性 | 详细资料 | 分区 | 索引 | SQL

操作...

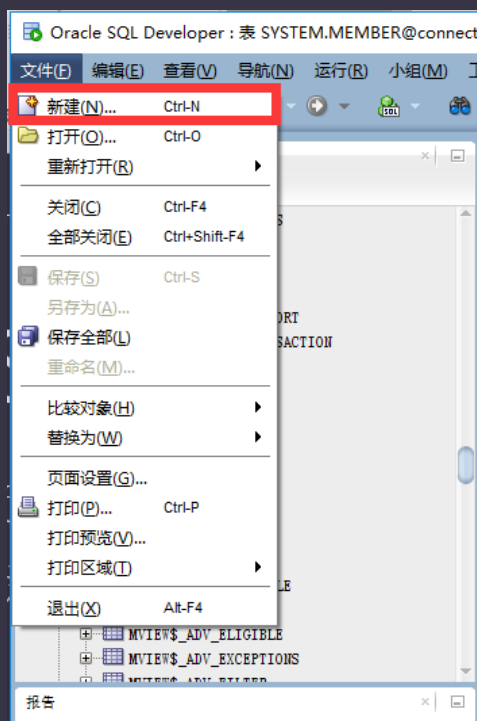
	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
1	SYS_C007365	Check	"COUNTRY_CODE" IS NOT NULL
2	SYS_C007366	Check	title IN ('Mr', 'Mrs', 'Ms', 'Miss', 'Dr')
3	SYS_C007367	Primary_Key	(null)
4	SYS_C007368	Foreign_Key	(null)
5	SYS_C007369	Foreign_Key	(null)

约束代号, sql语句出错时可以查阅

# 4. SQL Developer

## 5. 执行复杂的sql语句

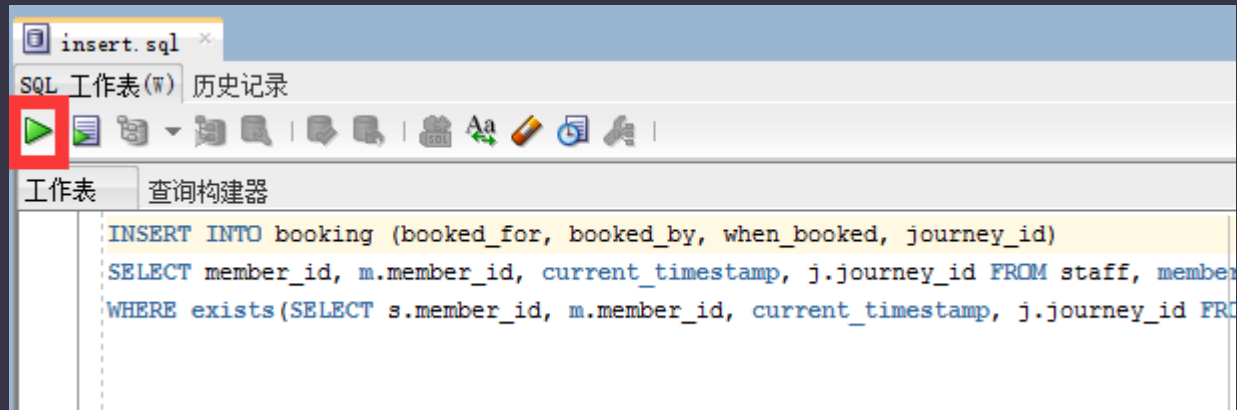
### (1) 新建sql文件



# 4. SQL Developer

## 5. 执行复杂的sql语句

### (2) 执行语句



数据库安装及基本配置介绍完毕





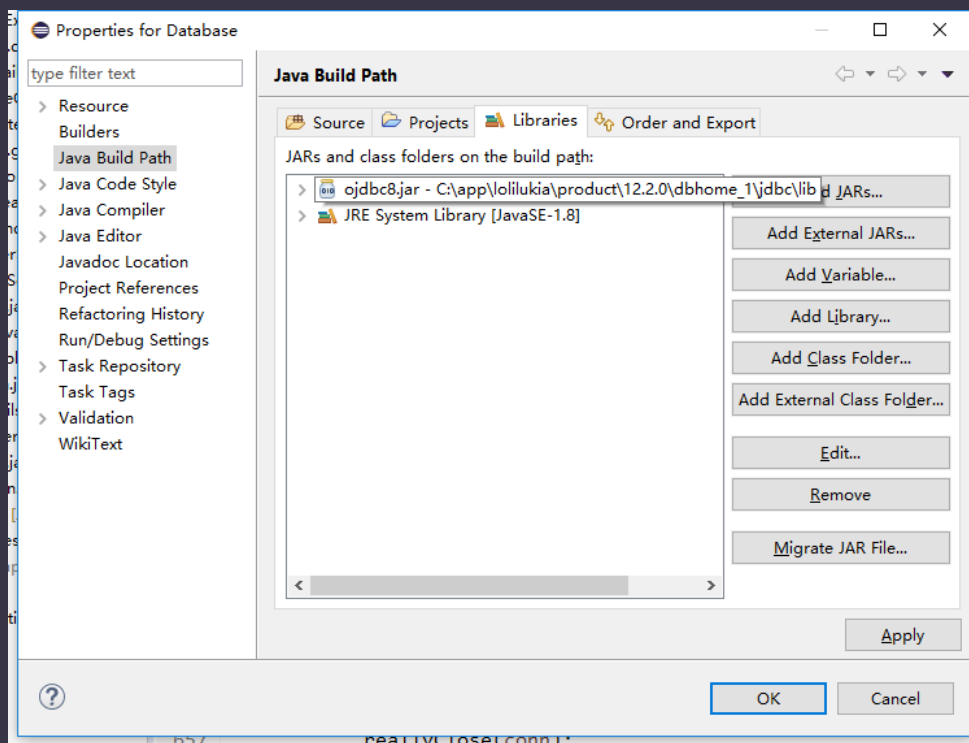
# ORACLE 12C在JDBC的应用

数据库原理与应用

# 1. 配置

IDE环境: eclipse(Java-SE 1.8)

(1) 导入ojdbc的jar包, 路径为 (oracle基目录\jdbc\lib), 文件名为ojdbc8.jar



# 1. 配置

- (2) 以一个为运动员预订行程的数据库程序为例，数据库配置文件olympicsdb.properties

```
olympicsdb.properties
1# VENDOR
2# either postgresql or oracle
3dbvendor = oracle → 数据库名称
4
5# SERVER ADDRESS
6# SIT PostgreSQL server is soit-db-pro-2.ucc.usyd.edu.au
7# Local servers are at localhost
8address = localhost
9
10# PORT
11# default for PostgreSQL is 5432
12# default for Oracle is 1521
13port = 1521 → 端口号
14
15# USER DETAILS
16# This is the user name and password with which
17# to connect to the database server
18# UG student accounts have user y17i2x20_YOURUNIKEY
19# PG student accounts have user YOURUNIKEY
20# Alternatively can use info2120public/info2120public for INFO2x20
21# PostgreSQL or comp9120public/comp9120public for COMP9120 Oracle
22username = system
23userpass = 1234 → 用户名和管理口令
24
25# DBNAME
26# (PostgreSQL) The name of the database on the server
27# By default usually the same as the user name
28# (Oracle) This is the connection identifier. If you have a local server you
29# probably need the default "ORCL". The SIT server uses "COMP9120"
30dbname = olympics → 数据库实例名称
31
32
```

# 1. 配置

(3) 连接数据库，第一步读出配置文件里的属性

```
Properties props = new Properties();  
try {  
    props.load(config);  
} catch (IOException e) {  
    throw new OlympicsDBException("Couldn't read config data",e);  
}
```

```
dbUser = props.getProperty("username");  
dbPass = props.getProperty("userpass");  
String port = props.getProperty("port");  
String dbname = props.getProperty("dbname");  
String server = props.getProperty("address");
```

将配置文件里的属性读出

# 1. 配置

(3) 连接数据库，第一步读出配置文件里的属性，第二步填入属性，第三步启动驱动创建连接

```
Properties props = new Properties();
try {
    props.load(config);
} catch (IOException e) {
    throw new OlympicsDBException("Couldn't read config data",e);
}
```

```
dbUser = props.getProperty("username");
dbPass = props.getProperty("userpass");
String port = props.getProperty("port");
String dbname = props.getProperty("dbname");
String server = props.getProperty("address");
```

将配置文件里的属性读出

```
} else if ("oracle".equals(vendor)) {
    Class.forName("oracle.jdbc.driver.OracleDriver");
    connstring = "jdbc:oracle:thin:@" + server + ":" + port + ":" + dbname;
} else throw new OlympicsDBException("Unknown database vendor: " + vendor);
```

```
Connection conn;
conn = DriverManager.getConnection(connstring, dbUser, dbPass);
return conn;
```

# 数据操作

## (1) PreparedStatement类

对于只执行一次的请求，使用Statement会更快，对于执行很多次的语句，PreparedStatement在多次执行的性能就会体现，并且可以一次发送多个查询，用法如下：

```
Connection conn = getConnection();
```

```
String sql = "select * from member where member_id=? and pass_word=?";  
PreparedStatement pre = conn.prepareStatement(sql);  
pre.setString(1, member);  
pre.setString(2, String.valueOf(password));  
ResultSet result = pre.executeQuery();
```

通过设置占位符的值来补全整个sql语句

# 数据操作

## (1) PreparedStatement类

当sql语句执行时，ResultSet务必要执行next()方法，才将result的值指向第一行数据

```
if (result.next() && result_t.next()) {  
    details = new HashMap<String, Object>();  
    // Populate with record data  
    details.put("member_type", result_t.getString(1));  
    System.out.println(result_t.getString(1));  
}  
result.close();  
pre.close();  
conn.close();
```

result.getString(1)代表执行sql语句查询的第一个字段，该字段读出为String类型

整个方法结束时，需要关闭相关的ResultSet，PreparedStatement和Connection

# 数据操作

## (1) 增删改查操作

查询操作：以下操作是找出某个event\_id代表的体育赛事运动员的名字，姓氏，国籍和奖牌获得情况，其中奖牌返回时设置了别名，并最后按照运动员姓氏排序

```
conn = getConnection();
String event = "select given_names, family_name, c.country_name, "
    + "CASE WHEN medal='G' THEN 'Gold' WHEN medal='S' THEN 'Silver' WHEN medal='B' THEN 'Bronze' END AS medal "
    + "from member join participates p on member.member_id=p.athlete_id "
    + "join country c on member.country_code=c.country_code "
    + "where event_id=? "
    + "ORDER BY family_name ";
pre = conn.prepareStatement(event);
pre.setString(1, String.valueOf(eventId));
result = pre.executeQuery();
while(result.next()){
    HashMap<String,Object> result1 = new HashMap<String,Object>();
    result1.put("participant", result.getString(2)+"", "+result.getString(1));
    result1.put("country_name", result.getString(3));
    result1.put("medal", result.getString(4));
    results.add(result1);
}
```



# 数据操作

## (1) 增删改查操作

### 新增操作

下面展现的例子是，某个member\_id的员工(staff 表)为某个member\_id(member表)的用户预订一个某个journey\_id的行程，其中booking中1,3,4字段都是外键，此时直接插入数据就违反了主外键完整性约束，插入的时候需要使用insert into table(field1,...) select语句

```
String sql2 = "INSERT INTO booking (booked_for, booked_by, when_booked, journey_id) "
    + "SELECT m.member_id, s.member_id, "
    + "to_timestamp(to_char(current_timestamp, 'yyyy-MM-dd hh24:mi:ss'), 'yyyy-MM-dd hh24:mi:ss'), "
    + "j.journey_id FROM staff s, member m, DUAL, journey j where s.member_id=? "
    + "AND m.member_id=? AND j.journey_id=? ";
PreparedStatement pre2 = conn.prepareStatement(sql2);
pre2.setString(1, byStaff);
pre2.setString(2, forMember);
pre2.setInt(3, journeyId);
pre2.executeUpdate();
```

# 数据操作

## (1) 增删改查操作

### 修改操作

还是上面那个例子，在预订成功之后，journey表中的已预订字段需要加一，以下是更新操作的具体写法：

```
String sql4 = "Update journey set nbooked=nbooked+1 where journey_id=?";  
PreparedStatement pre4 = conn.prepareStatement(sql4);  
pre4.setInt(1, journeyId);  
pre4.executeUpdate();
```

注：修改操作和新增操作不返回result，在所有操作完成时，需要提交事务

```
conn.commit();
```

在catch语句中，如果执行不成功，则执行事务回滚

```
conn.rollback();
```

谢谢！