数据库第13周作业

19336035 陈梓乐

---

1. **学习游标的使用（cursor，参考《Introduction to PL/SQL》一书第 6 章），利用游标实现“查询公司中工资最高的三位员工”**

```
set serveroutput on;

declare
    cursor most_sal is
        select ename, sal from emp
        order by sal desc;
    sal emp.sal%type;
    ename emp.ename%type;
    i integer;
begin
    open most_sal;
    for i in 1..3 loop
        fetch most_sal into ename, sal;
        dbms_output.put_line(ename||' '||sal);
    end loop;
    close most_sal;
end;
/
```

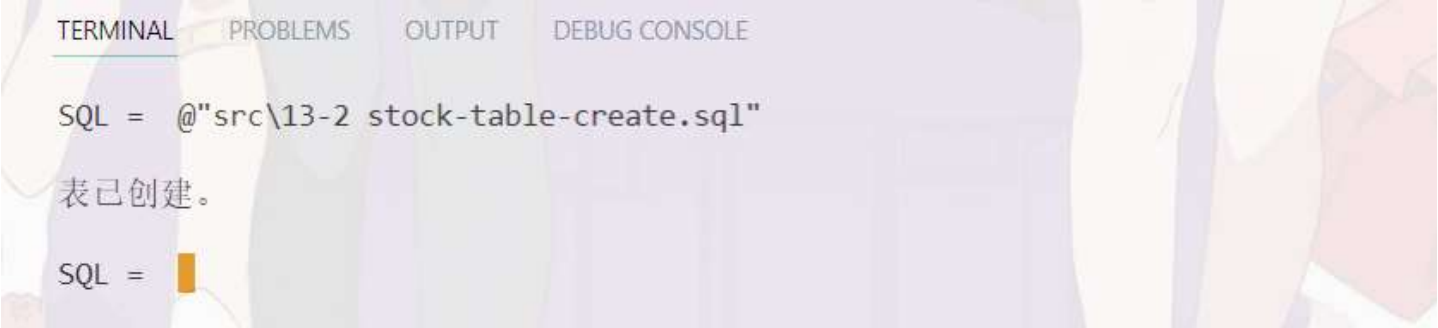
2. **把“课程资源”中的“上证指数历史数据”导入到 Oracle，完成以下任务（尽量使用 1 条 SQL语句完成）**

- **参考课程幻灯片第 48 页，求出所有“黄金交叉”（日期）和“死亡交叉”**
- **参考课程幻灯片第 51 页，找出所有“连升三天”“连跌三天”的日期**

导入数据

- 创建表格：

```
create table ss001 (
    day     char(10),
    open    number(12, 6),
    high    number(12, 6),
    low     number(12, 6),
    close   number(12, 6),
    adjclose    number(12, 6),
    volume  number(12, 6)
);
```



- 导入数据

```
options(skip=1,rows=4096)
load data
infile "data/000001.SS.csv"
truncate
INTO table ss001
fields terminated by ","
(day,open,high,low,close,adjclose,volume)
```

- 展示导入效果

```
SQL =  select * from ss001 where rownum <= 3;

DAY              OPEN       HIGH        LOW      CLOSE   ADJCLOSE     VOLUME
---------- ---------- ---------- ---------- ---------- ---------- ----------
1998-01-19 1215.40796 1216.98706   1205.479 1213.12903 1213.12903          0
1998-01-20   1211.276 1215.56995 1195.38196 1203.71497 1203.71497          0
1998-01-21 1201.98804 1209.68897 1198.22803   1200.203   1200.203          0

SQL =
```

找出黄金交叉与死亡交叉日期

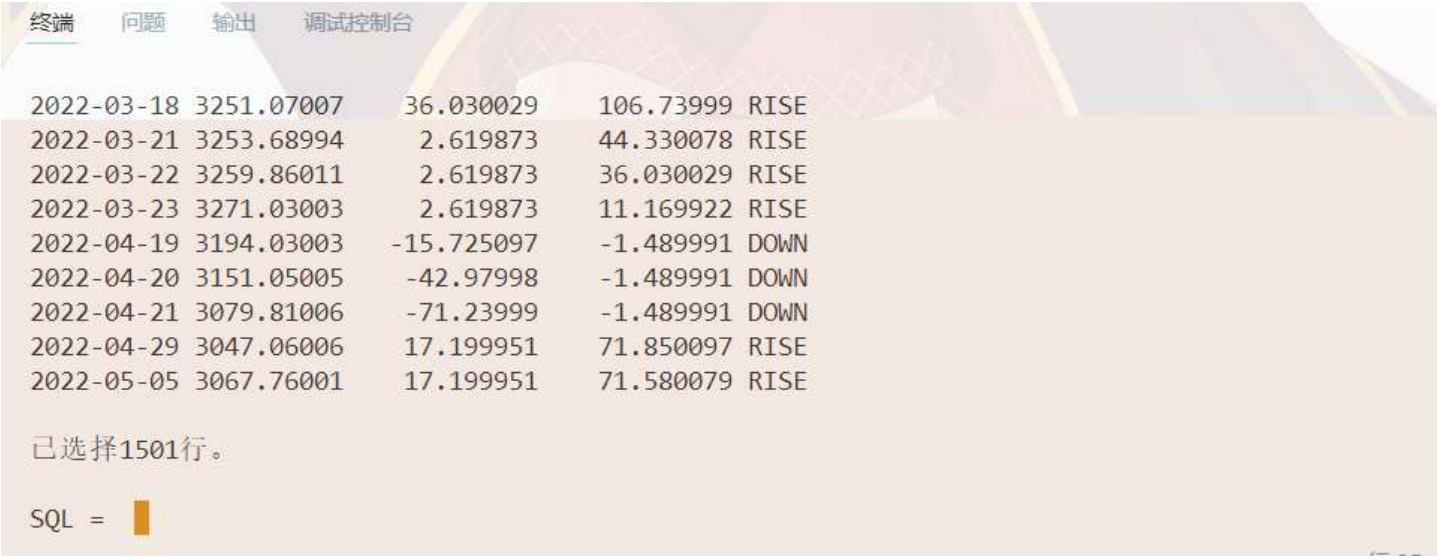
```sql
with analysis1 as (
    select
        day,
        close,
        avg(close) over (
            order by day
            rows between 10 preceding and 1 preceding
        ) MA10,
        avg(close) over (
            order by day
            rows between 30 preceding and 1 preceding
        ) MA30
    from ss001
), analysis2 as (
    select
        day, close, MA10, MA30,
        lag(MA10 - MA30) over(order by day) LAST_DAY,
        MA10 - MA30 TODAY
    from analysis1
), analysis3 as (
    select day,close,MA10,MA30,LAST_DAY,TODAY,
    case
        when LAST_DAY < 0 and TODAY > 0 then 'GOLDEN'
        when LAST_DAY > 0 and TODAY < 0 then 'DEAD'
        else null
    end OPERATE
    from analysis2
)
select * from analysis3
where OPERATE is not null;
```

```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

SQL =  @"src\13-3 operate-day.sql"

DAY              CLOSE       MA10       MA30   LAST_DAY      TODAY OPERAT
---------- ---------- ---------- ---------- ---------- ---------- ------
1997-07-23 1198.86096 1170.52511  1165.0783 -2.3753474 5.44681047 GOLDEN
1997-08-19 1132.90503  1167.0338 1169.12321 3.65689287 -2.0894044 DEAD
1997-09-02 1226.30298 1185.53949  1177.8423  -.6145753  7.6971882 GOLDEN
1997-09-24 1110.01196   1174.434 1187.39666  2.4487387 -12.962663 DEAD
1997-10-24 1178.30603 1155.12301 1150.11727 -2.8128988 5.00574143 GOLDEN
1997-11-25 1125.52795 1176.67462 1177.34654  5.5831828 -.67191977 DEAD
1997-12-23 1171.25403 1159.23129 1156.25827 -.55471597  2.9730144 GOLDEN
1998-03-02 1188.30603  1218.7869 1222.02347 1.11580007 -3.2365641 DEAD
1998-04-01 1254.96497 1204.04449 1203.89567 -3.7187458 .148824167 GOLDEN
1998-06-18 1370.68506 1375.05299 1377.07303 5.10343013 -2.0200358 DEAD
1998-09-16 1252.13794  1188.5679 1181.49594 -3.7391724 7.07196047 GOLDEN
1998-10-28 1225.92896 1229.02168 1229.20116 2.50795083 -.17948403 DEAD
1998-11-10 1286.55896 1236.91079 1233.06153  -.8342367  3.8492635 GOLDEN
1998-12-07 1209.54102 1248.48961 1254.04487 .121439533 -5.5552613 DEAD
```

共210行，其中有105行是黄金交叉点。

找出连续涨跌三天的日子

```
with t1 as (
    select day, close,
        close - lag(close) over (order by day) gradient
    from ss001
), t2 as (
    select day, close,
        min(gradient) over (
            order by day
            rows between 2 preceding and current row
        ) min_gradient,
        max(gradient) over (
            order by day
            rows between 2 preceding and current row
        ) max_gradient
    from t1
)
select
    day, close, min_gradient, max_gradient,
    case
        when min_gradient > 0 then 'RISE'
        when max_gradient < 0 then 'DOWN'
        else null
    end status
from t2
where min_gradient * max_gradient > 0 and day >= '1997-07-05';
```



共1501行，其中连续涨3天的日子有864天。

3. **参考课程幻灯片第 68 页。**

- **先随机生成模拟数据，包含手机号，时间戳（只需要考虑在同天内即可），基站 id，每一行数据表示手机定期与最接近的基站联系了一次**
- **假设某个特定号码沿着时间轴联系了 A 站，接着再联系 B 站（也可能是再次联系 A），那么他在 A 的停留时间就是前后两个时间戳之差。求每个号码当天连接时间最长的三个基站**

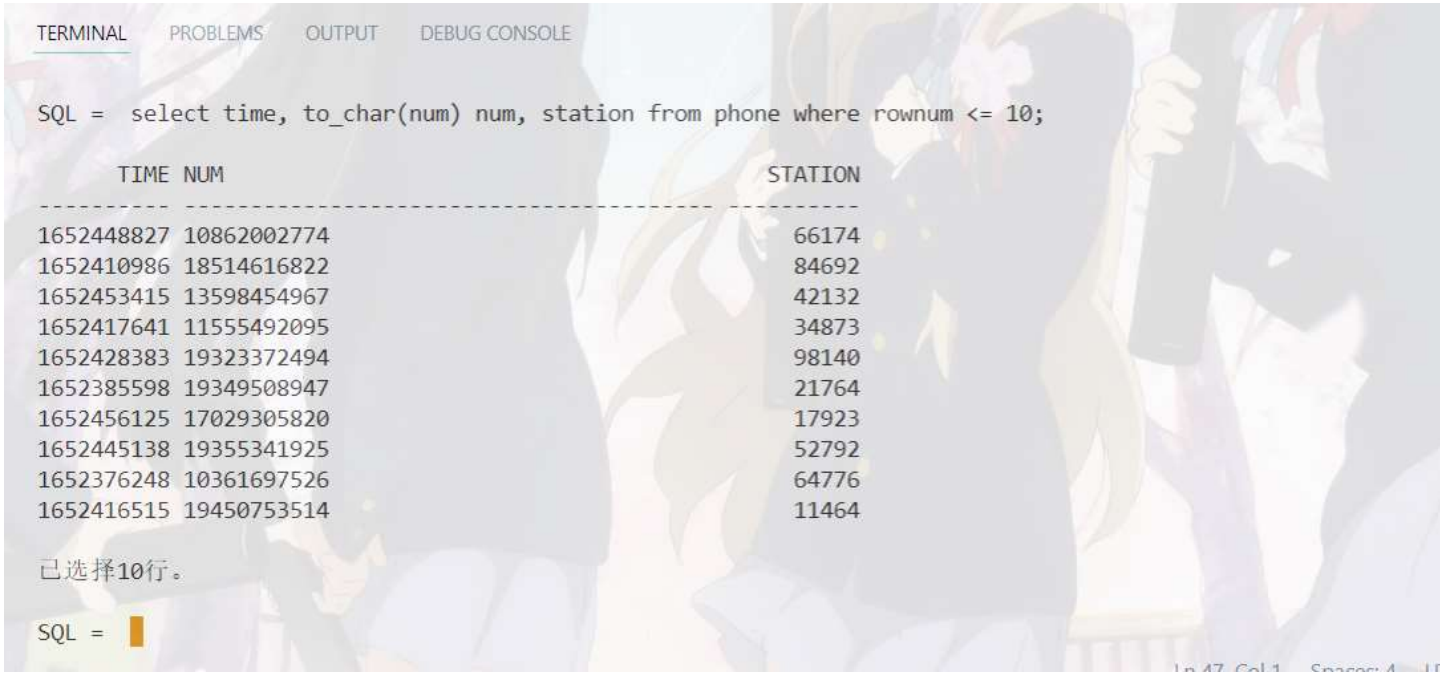
建立表格

```
create table phone (
    time number(16),
    num number(11),
    station number(5)
);
```

导入数据

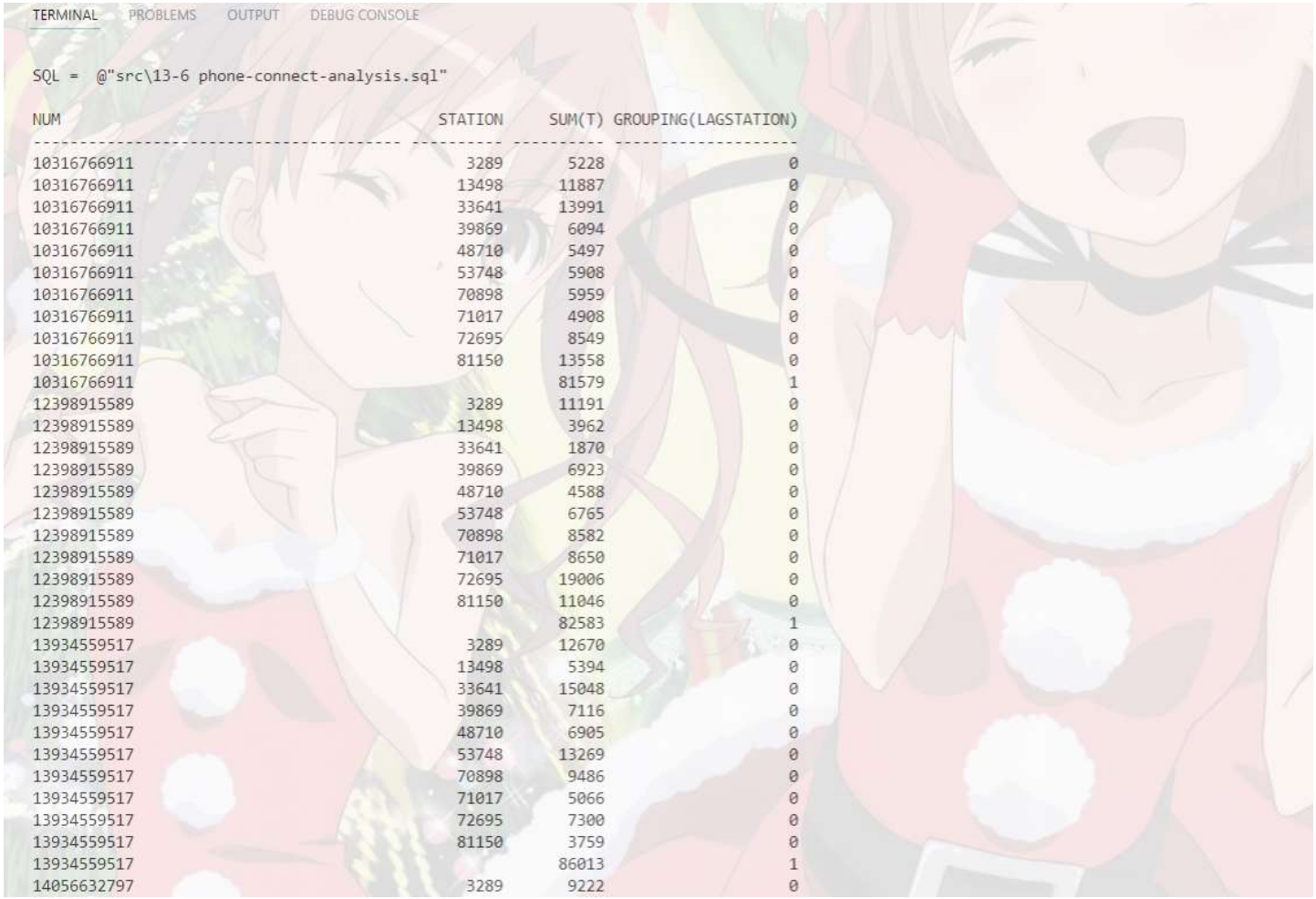
设有5个手机号，共10个基站，产生300条连接记录，导入代码如下：

```
import cx_Oracle, random
con = cx_Oracle.connect('scott/tiger@127.0.0.1/orcl')

# 手机号表
phone_number = [
    int(1e10) + random.randint(0, 1e10)
    for i in range(5)
]

# 基站表
station = [
    random.randint(0, 1e5)
    for i in range(10)
]

to_insert = [(
    random.randint(1652371200, 1652457600),
    random.choice(phone_number),
    random.choice(station))
    for i in range(300)
]

with open('data.csv', 'w') as f:
    for i in to_insert:
        f.write(','.join(map(str, i)))
        f.write('\n')

cur = con.cursor()
cur.bindarraysize = 10000
cur.setinputsizes(int, int, int)
cur.executemany("insert into phone values (:1, :2, :3)", to_insert)
con.commit()
```



处理数据

```
with analysis1 as (
    select
        to_char(num) num, time, station,
        lag(station) over (
           partition by num
            order by time
        ) lagstation,
        time - lag(time) over (
            partition by num
            order by time
        ) t
    from phone
), analysis2 as (
    select * from analysis1
    where t is not null
)
select  num, lagstation station,
        sum(t), grouping(lagstation)
from analysis2
group by rollup(num, lagstation);
```



```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

SQL =  @"src\13-6 phone-connect-analysis.sql"

NUM                                 STATION     SUM(T) GROUPING(LAGSTATION)
---------------------------------- ---------- ---------- -------------------
10316766911                              3289       5228                   0
10316766911                             13498      11887                   0
10316766911                             33641      13991                   0
10316766911                             39869       6094                   0
10316766911                             48710       5497                   0
10316766911                             53748       5908                   0
10316766911                             70898       5959                   0
10316766911                             71017       4908                   0
10316766911                             72695       8549                   0
10316766911                             81150      13558                   0
10316766911                                        81579                   1
12398915589                              3289      11191                   0
12398915589                             13498       3962                   0
12398915589                             33641       1870                   0
12398915589                             39869       6923                   0
12398915589                             48710       4588                   0
12398915589                             53748       6765                   0
12398915589                             70898       8582                   0
12398915589                             71017       8650                   0
12398915589                             72695      19006                   0
12398915589                             81150      11046                   0
12398915589                                        82583                   1
13934559517                              3289      12670                   0
13934559517                             13498       5394                   0
13934559517                             33641      15048                   0
13934559517                             39869       7116                   0
13934559517                             48710       6905                   0
13934559517                             53748      13269                   0
13934559517                             70898       9486                   0
13934559517                             71017       5066                   0
13934559517                             72695       7300                   0
13934559517                             81150       3759                   0
13934559517                                        86013                   1
14056632797                              3289       9222                   0
```

找出连接时间前三的基站

```sql
with analysis1 as (
    select
        to_char(num) num, time, station,
        lag(station) over (
           partition by num
            order by time
        ) lagstation,
        time - lag(time) over (
            partition by num
            order by time
        ) t
    from phone
), analysis2 as (
    select * from analysis1
    where t is not null
), analysis3 as (
    select  num, lagstation station,
            sum(t) s, grouping(lagstation) g,
            rank() over (partition by num order by sum(t) desc) r
    from analysis2
    group by rollup(num, lagstation)
)
select num, station, s time from analysis3
where r <= 4 and r >= 2;
```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE

```
SQL =  @"src\13-7 phone-connect-most.sql"

NUM                                        STATION       TIME
---------------------------------------- ---------- ----------
10316766911                                 33641      13991
10316766911                                 81150      13558
10316766911                                 13498      11887
12398915589                                 72695      19006
12398915589                                  3289      11191
12398915589                                 81150      11046
13934559517                                 33641      15048
13934559517                                 53748      13269
13934559517                                  3289      12670
14056632797                                 72695      13877
14056632797                                 71017      12642
14056632797                                 39869      12275
17159510523                                  3289      13597
17159510523                                 70898      12055
17159510523                                 81150      11567

已选择15行。
```