第四章

第四章 SQL是关于集合的

4.1 以面向集合的思维方式来思考

在线小游戏：www.setgame.com/puzzle/set.htm

4.1.1 从面向过程转变为基于集合的思维方式

eg:

```
select distinct employee_id
from job_history j1
where not exists
(select null from job_history j2
where j2.employee_id = j1.employee_id
and round(month_between(j2.start_date, j2.end_date)/12,0) <>
round(months_between(j1.start_date,j1.end_date)/12,0);
select employee_id
from job_history
group by employee_id
having min(round(months_between(start_date,end_date)/12,0)) =
max(round(months_between(start_date, end_date)/12,0));
```

eg: 过程化与基于集合的方法的对比

```
set autotrace on
select distinct employee_id
from job_history j1
where not exists
    (select null
        from job_history j2
    where j2.employee_id = j1.employee_id
    and round(months_between(j2.start_date,j2.end_date)/12,0) <>
    round(months_between(j1.start_date,j1.end_date)/12,0));

select employee_id
    from job_history
    group by employee_id
    having min(round(months_between(start_date,end_date)/12,0)) =
    max(round(months_between(start_date,end_date)/12,0));
```

4.1.2 面向过程 vs. 基于集合的思维方式：一个例子

eg: show the list of order dates for customer 102

```
select customer_id, order_date from orders where customer_id = 102;
```

eg: determine the order_date prior to the current row's order_date

```
select customer_id, order_date, lag(order_date, 1, order_date) over (partition by customer_id order by order_date) as
prev_order_date from orders where customer_id = 102;
```

eg: determine the days between each order

```
select trunc(order_date) - trunc(prev_order_date) days_between
from
(
select customer_id, order_date,
    lag(order_date, 1, order_date)
    over (partition by customer_id order by order_date)
    as prev_order_date
```

```
from orders
where customer_id = 102
);

eg: put it together wth an AVG function to get the final answer
select avg(trunc(order_date) - trunc(prev_order_date)) avg_days_between
from
(
select customer_id, order_date,
    lag(order_date, 1, order_date)
    over (partition by customer_id order by order_date)
    as prev_order_date
from orders
where customer_id = 102
);

eg: 基于集合的思维方式
select (max(trunc(order_date)) - min(trunc(order_date))) / count(*) as avg_days_between
from orders
where customer_id = 102;
```

4.2 集合运算

UNION

UNION ALL

MINUS

INTERSECT

```
eg:
select color from table1
union
select color from table2;
eg:
select color from table1
union all
select color from table2;
eg:
select color from table1
minus
select color from table2;
eg: minus queries are equivalent to not exists queries
select distinct color from table1
where not exists (select null from table2 where table2.color = table1.color);
eg:
select color from table2
minus
select color from table1;
eg:
select color from table1
minus
select color from table3;
eg:
select color from table1
intersect
select color from table2;
eg:
select color from table1
intersect
select color from table3;
```

eg: 空值与集合运算

```
select null from dual
union
select null from dual;
```

eg:

```
select null from dual
union all
select null from dual;
```

eg:

```
select null from dual
intersect
select null from dual;
```

eg:

```
select null from dual
minus
select null from dual;
```

eg:

```
select 1 from dual
union
select null from dual;
```

eg:

```
select 1 from dual
union all
select null from dual;
```

eg:

```
select 1 from dual
intersect select null from dual;
select 1 from dual
minus select null from dual;
```

4.3.3 空值与 group by 和 order by

eg:

```
select comm, count(*) ctr from scott.emp group by comm;
```

eg:

```
select comm, count(*) ctr from scott.emp group by comm order by comm;
```

eg:

```
select comm, count(*) ctr from scott.emp group by comm order by comm nulls first;
```

eg:

```
select ename, sal, comm from scott.emp order by comm, ename;
```

4.3.4 空值与聚合函数

eg:

```
select count(*) row_ct, count(comm) comm_ct, avg(comm) avg_comm, min(comm) min_comm, max(comm) max_comm, sum(comm) sum_comm from
scott.emp;
```