1. select 'ENTRY\_ID' from dt\_account;

2. select \* from dt\_account where "ENTRY\_ID" is not null;

select \* from dt\_account where "ENTRY\_ID" is not null limit 6 offset 10;

select \* from dt\_account where "ENTRY\_ID" is not null order by login\_id asc limit 6 offset 10 ;

3.select account\_id, to\_char(last\_login\_date, 'yyyy/mm/dd hh24:mi:ss') from dt\_account;

select account\_id, to\_char(last\_login\_date, 'yyyy/mm/dd hh24:mi:ss') from dt\_account where last\_login\_date between(now() - interval '1 month') and now();

4.select account\_id, to\_char( last\_login\_date, 'yyyy/mm/dd hh24:mi:ss') from dt\_account;

select account\_id, to\_char( last\_login\_date, 'yyyy/mm/dd hh24:mi:ss')

from dt\_account where last\_login\_date between (now() - cast (exer\_type || 'month' as interval)) and now();

5.select \* from dt\_account where lower(mail\_address)='takku@test.com';

select \* from dt\_account where mail\_address='takku@test.com' or

mail\_address='takku@test.com';

6.select \* from dt\_account where login\_id like '%o%' or login\_id like '%O%';

select \* from dt\_account where lower(login\_id) like '%o%';

select \* from dt\_account where login\_id ilike '%o%';

7.select \* from dt\_account\_info where lower(exer\_comment)=

'psql to\n\'postgres\'';

8. select exer\_name from dt\_account where exer\_name ilike '%?%';

select replace(exer\_name, '?', '@HG@') from dt\_account where exer\_name

ilike '%?%';

select exer\_name from dt\_account where position('?' in exer\_name) > 0;

select replace(exer\_name, '?', '@HG@') from dt\_account where

position('?' in exer\_name) > 0;

position('?' in exer\_name) 返回？在exer\_name中的位置，如果在exer\_name中 没有？，则返回0；

9.select \* from dt\_account where login\_id ilike '%-%';

select \* from dt\_account where position('-' in login\_id) > 0;

select substring(login\_id, position('-' in login\_id) + 1)) from dt\_account where login\_id ilike '%-%';

10.select \* from dt\_account;

select \* from dt\_account where last\_login\_date between '2007-06-24

12:00:00' and '2007-06-24 21:00:00';

结束时间直接使用下一个时段，例如结束时时8点，就用9点。

表连接查询：

11.

select account\_id, exer\_type\_name from dt\_account

join mt\_exer\_type on(exer\_type=exer\_type\_id);

select account\_id, exer\_type\_name, guild\_name from dt\_account

join mt\_exer\_type on(exer\_type=exer\_type\_id)

left join dt\_guild using(guild\_id);

select account\_id, exer\_type\_name, guild\_name, entry\_name from dt\_account

join mt\_exer\_type on(exer\_type=exer\_type\_id)

left join dt\_guild using(guild\_id)

left join mt\_entry on(dt\_account."ENTRY\_ID" = mt\_entry.entry\_id);

12.

select account\_id, login\_id, exer\_name, exer\_type, sex\_flg sex\_name, guild\_id, "ENTRY\_ID"

from dt\_account;

select account\_id, login\_id, exer\_name, exer\_type, guild\_id, "ENTRY\_ID",

sex\_flg sex\_name

from dt\_account;

注释：case用法

\* case前面必须有“，”逗号，此处sex\_fly后面（其别名sex\_name)有逗号。

\* case必须接from语句

以上认为不对。

select account\_id, login\_id, exer\_name, exer\_type, guild\_id, "ENTRY\_ID",

sex\_flg sex\_name,

case

when sex\_flg='m' then '男性'

when sex\_flg='f' then '女性'

end

from dt\_account;

select account\_id, login\_id, exer\_name, exer\_type\_name, guild\_id, "ENTRY\_ID",

sex\_flg sex\_name,

case

when sex\_flg='m' then '男性'

when sex\_flg='f' then '女性'

end

from dt\_account

join mt\_exer\_type on (exer\_type=exer\_type\_id);

老师所给答案:

select dt\_account.acount\_id, login\_id, exer\_name,

CASE WHEN sex\_flg = 'm'

THEN '男性'

WHEN sex\_flg = 'f'

THEN '女性'

ELSE ''

END

AS sex\_name from dt\_account;

13:显示mt\_product 的所有数据，他关联mt\_skill的两个字段skill\_id\_0, skill\_id\_1用对应的

skill\_name 表示出来。

select product\_id, product\_name, product\_price, skill\_id\_0, skill\_name\_0, skill\_id\_1, skill\_name\_1

from(select product\_id product\_id\_0, product\_name product\_name\_0,

product\_price product\_price\_0, skill\_id\_0 skill\_id\_0\_0,

skill\_name skill\_name\_0 from mt\_product mt\_product\_0

left join mt\_skill mt\_skill\_1 on(skill\_id\_0 = skill\_id)

) as ss1 join(select mt\_product\_1.\*, skill\_name skill\_name\_1 from mt\_product

mt\_product\_1 left join mt\_skill mt\_skill\_2 on(skill\_id\_1 = skill\_id)

) as ss2

on(product\_id = product\_id\_0);

用子查询时：所用的子表必须有一个名字，例如这里的ss1 和 ss2.

老师所给答案：

select \* from mt\_product left join mt\_skill mt\_skill\_0

on(mt\_skill\_0.skill\_id = mt\_product.skill\_id\_0)

left join mt\_skill mt\_skill\_1

on(mt\_skill\_1.skill\_id = mt\_product.skill\_id\_1);

14.含有"script"关键字的技能(mt\_skill)的产品(mt\_product)的价值(product\_price)总和是多少。

select sum(product\_price) from (select product\_price, skill\_id\_by\_0, skill\_id\_by\_1 from(select product\_id, product\_name, product\_price, skill\_id\_by\_0, skill\_id\_by\_1 from mt\_product left

join (select skill\_id skill\_id\_by\_0 from mt\_skill where skill\_name ilike '%script%') mt\_skill\_0

on(mt\_product.skill\_id\_0 = mt\_skill\_0.skill\_id\_by\_0)

left join (select skill\_id skill\_id\_by\_1 from mt\_skill where skill\_name ilike '%script%') mt\_skill\_1

on(mt\_skill\_1.skill\_id\_by\_1 = mt\_product.skill\_id\_1)) as sst3

where (skill\_id\_by\_0 is not null) or (skill\_id\_by\_1 is not null)) as sst4;

15。计算每个会员(dt\_account)所拥有的商品(mt\_product)的总价值(product\_price\*product\_count)

select sum(the\_product) from

(

select account\_id, product\_price\*product\_count the\_product from

(select account\_id, product\_id product\_id\_by\_1, product\_count from dt\_account\_product)

ss1 join

(select product\_id product\_id\_by\_2, product\_price from mt\_product)

ss2

on product\_id\_by\_1= product\_id\_by\_2

) ss3 group by(account\_id);

显示会员的全部信息:

select \* from dt\_account join

(

select account\_id, sum(the\_product) from

(

select account\_id, product\_price\*product\_count the\_product from

(select account\_id, product\_id product\_id\_by\_1, product\_count from dt\_account\_product)

ss1 join

(select product\_id product\_id\_by\_2, product\_price from mt\_product)

ss2

on product\_id\_by\_1= product\_id\_by\_2

) ss3 group by(account\_id)

) as ss4

on(ss4.account\_id = dt\_account.account\_id);

16.将dt\_account dt\_guild mt\_entry mt\_exer\_type 进行连接后 只能得到299条数据。

而 dt\_account中一共有300条数据。找出这条数据，他有何特点。

select \* from dt\_account where account\_id not in

(select account\_id from

(select account\_id, exer\_type\_name, guild\_name, entry\_name from dt\_account

join mt\_exer\_type on(exer\_type=exer\_type\_id)

left join dt\_guild using(guild\_id)

left join mt\_entry on(dt\_account."ENTRY\_ID" = mt\_entry.entry\_id)

) as ss1

);

17.计算每个会员所拥有的product的总价值和dt\_account\_info中的experience\_point的比率。

其中sum\_product\_price时分子，experience\_point是分子。

select ss5.\*,

case

when experience\_point <> 0 then all\_price / experience\_point

else 0

end

as all\_price\_experience\_point

from dt\_account\_info join

(

select \* from dt\_account join

(

select account\_id account\_id\_1, sum(the\_product) all\_price from

(

select account\_id, product\_price\*product\_count the\_product from

(select account\_id, product\_id product\_id\_by\_1, product\_count from dt\_account\_product)

ss1 join

(select product\_id product\_id\_by\_2, product\_price from mt\_product)

ss2

on product\_id\_by\_1= product\_id\_by\_2

) ss3 group by(account\_id)

) as ss4

on(ss4.account\_id\_1 = dt\_account.account\_id)

) as ss5 on (ss5.account\_id\_1 = dt\_account\_info.account\_id);

18.将dt\_account的account\_id和exer\_name组合成一个新字段输出，形式是'\accounnt\_id\'exer\_name

select '\\' || account\_id || '\\\'' || exer\_name from dt\_account;

19.将dt\_account的exer\_name和他所属的guild\_name组合成一个新字段输出。形如

exer\_name@guild\_name 例如：alter96@神勇

注意：guild\_id可能为null

select exer\_name || '@' ||

(case

when cast(guild\_id as varchar) is not null then cast(guild\_id as varchar)

when cast(guild\_id as varchar) is null then ''

end

)

from

(select account\_id, exer\_name, t1.guild\_id, guild\_name from dt\_account t1

left join dt\_guild t2 on(t1.guild\_id = t2.guild\_id)

) as ss1;