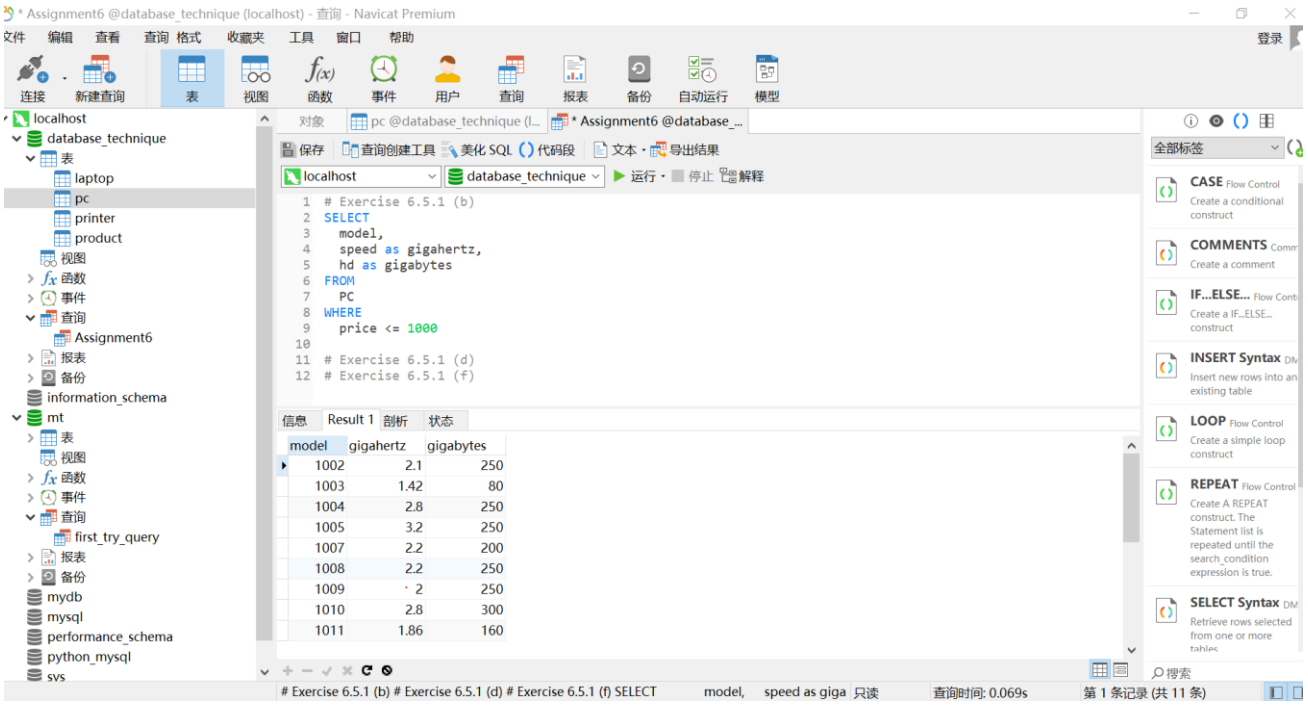
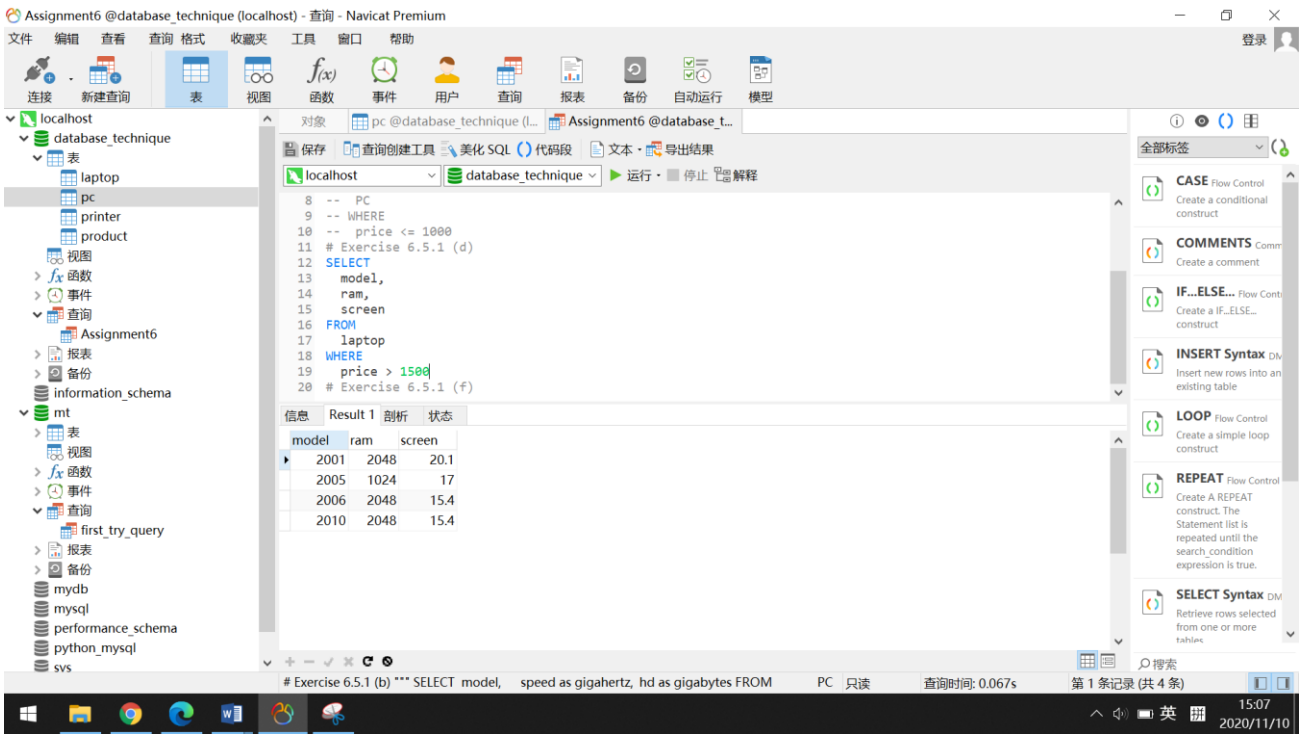
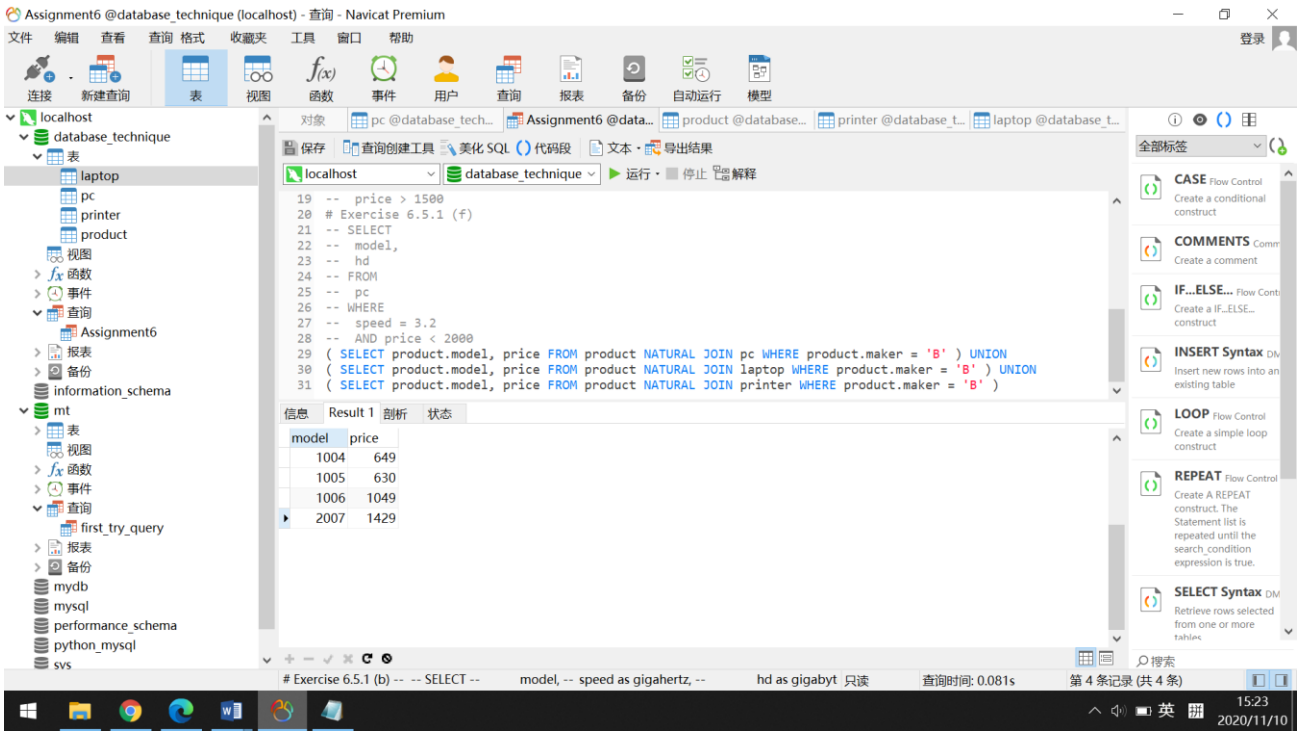
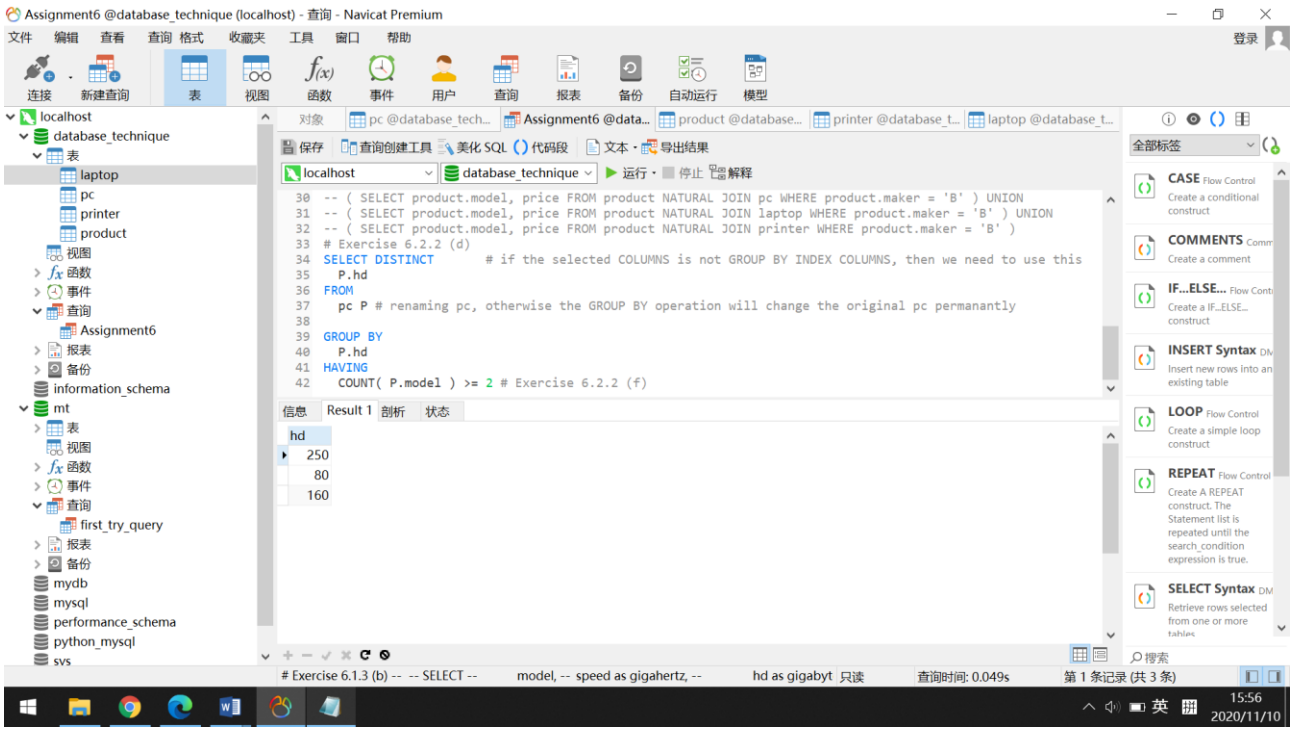
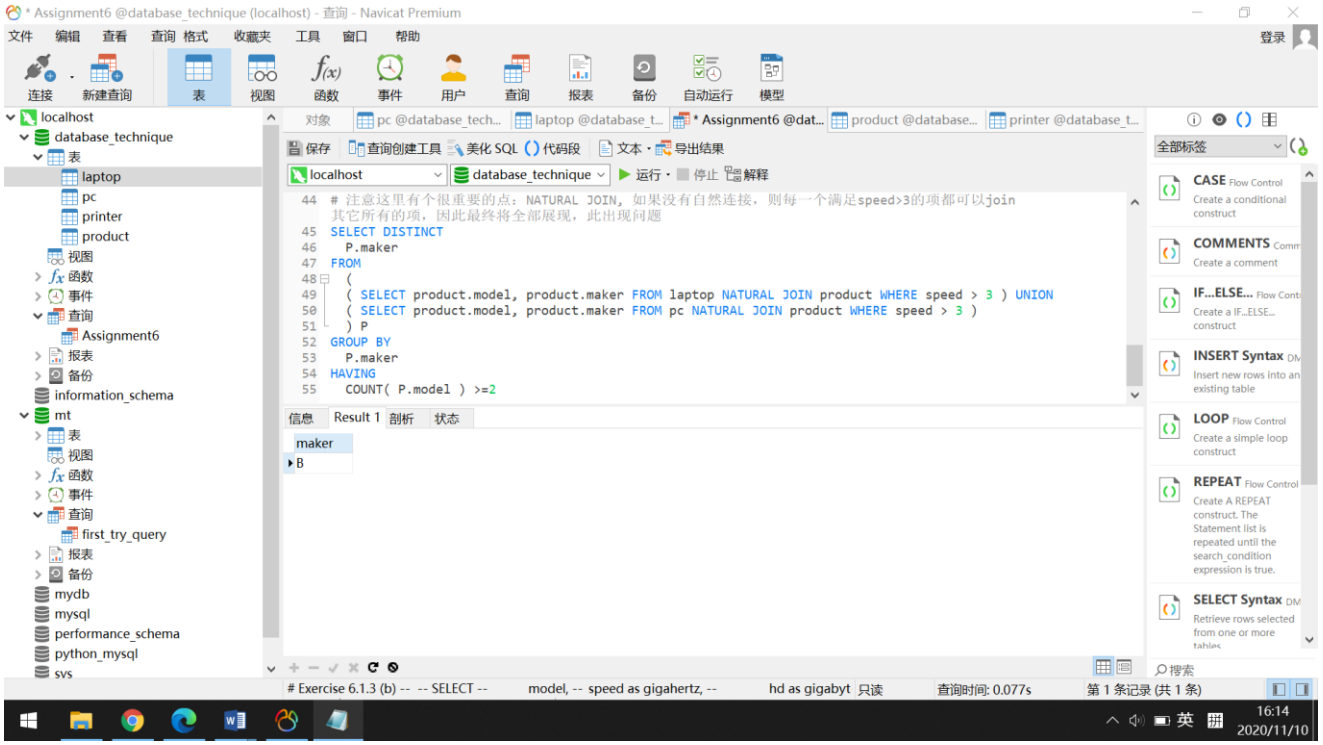
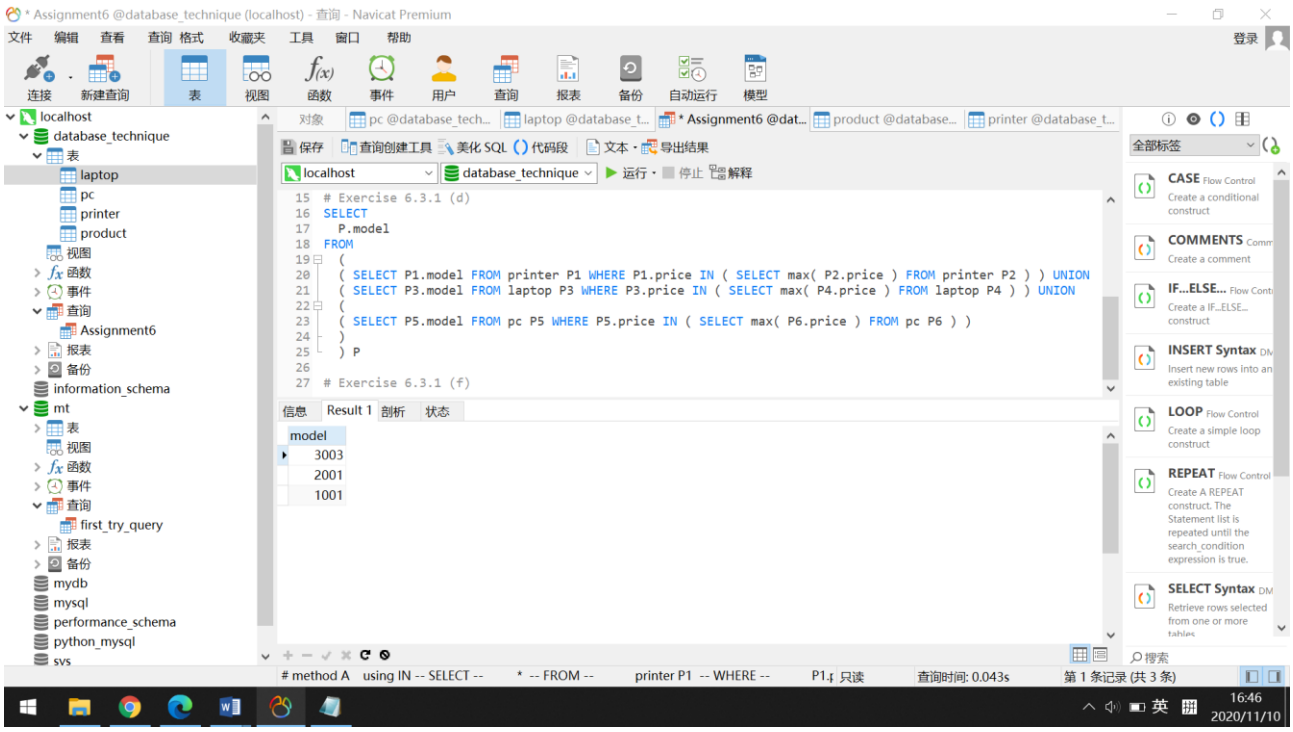

* Assignment6 @database_technique (localhost) - 查询 - Navicat Premium
文件 编辑 查看 查询 格式 收藏夹 工具 窗口 帮助
登录
连接 新建查询 表 视图 函数 事件 用户 查询 报表 备份 自动运行 模型
localhost
database_technique
表
laptop
pc
printer
product
视图
函数
事件
查询
Assignment6
报表
备份
information_schema
mt
表
视图
函数
事件
查询
first_try_query
报表
备份
mydb
mysql
performance_schema
python_mysql
sys
对象
pc @database_technique (l...
* Assignment6 @database_...
保存 查询创建工具 美化 SQL 代码段 文本 导出结果
localhost database_technique 运行 停止 解释
1 # Exercise 6.5.1 (b)
2 SELECT
3 model,
4 speed as gigahertz,
5 hd as gigabytes
6 FROM
7 PC
8 WHERE
9 price <= 1000
10
11 # Exercise 6.5.1 (d)
12 # Exercise 6.5.1 (f)
信息 Result 1 剖析 状态
model gigahertz gigabytes
1002 2.1 250
1003 1.42 80
1004 2.8 250
1005 3.2 250
1007 2.2 200
1008 2.2 250
1009 2 250
1010 2.8 300
1011 1.86 160
全部标签
CASE Flow Control
Create a conditional construct
COMMENTS
Create a comment
IF...ELSE...
Create a IF...ELSE... construct
INSERT Syntax
Insert new rows into an existing table
LOOP Flow Control
Create a simple loop construct
REPEAT Flow Control
Create A REPEAT construct. The Statement list is repeated until the search_condition expression is true.
SELECT Syntax
Retrieve rows selected from one or more tables
搜索
Exercise 6.5.1 (b) # Exercise 6.5.1 (d) # Exercise 6.5.1 (f) SELECT model, speed as giga 只读 查询时间: 0.069s 第 1 条记录 (共 11 条)

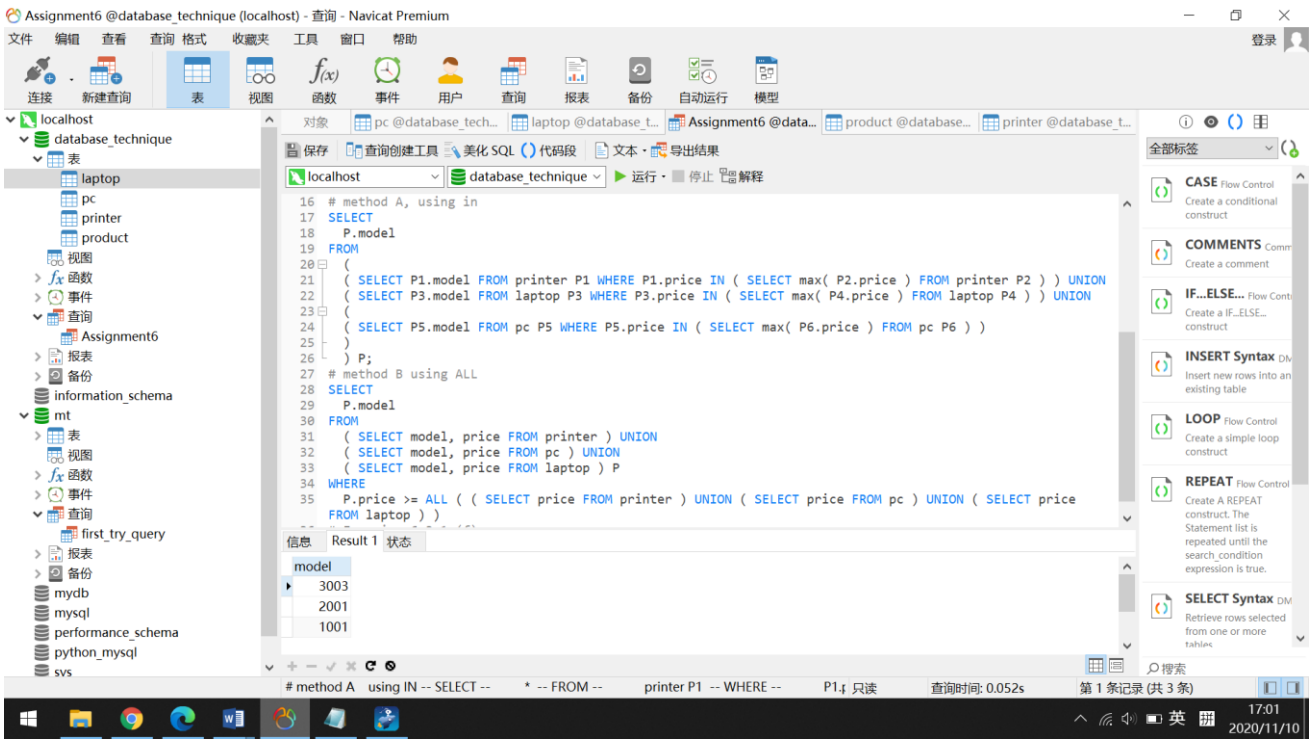

Assignment6 @database_technique (localhost) - 查询 - Navicat Premium
文件 编辑 查看 查询 格式 收藏夹 工具 窗口 帮助
登录
连接 新建查询 表 视图 函数 事件 用户 查询 报表 备份 自动运行 模型
localhost
database_technique
表
laptop
pc
printer
product
视图
函数
事件
查询
Assignment6
报表
备份
information_schema
mt
表
视图
函数
事件
查询
first_try_query
报表
备份
mydb
mysql
performance_schema
python_mysql
sys
对象
pc @database_technique (l...
Assignment6 @database_t...
保存 查询创建工具 美化 SQL 代码段 文本 导出结果
localhost database_technique 运行 停止 解释
8 -- PC
9 -- WHERE
10 -- price <= 1000
11 # Exercise 6.5.1 (d)
12 SELECT
13 model,
14 ram,
15 screen
16 FROM
17 laptop
18 WHERE
19 price > 1500
20 # Exercise 6.5.1 (f)
信息 Result 1 剖析 状态
model ram screen
2001 2048 20.1
2005 1024 17
2006 2048 15.4
2010 2048 15.4
全部标签
CASE Flow Control
Create a conditional construct
COMMENTS
Create a comment
IF...ELSE...
Create a IF...ELSE... construct
INSERT Syntax
Insert new rows into an existing table
LOOP Flow Control
Create a simple loop construct
REPEAT Flow Control
Create A REPEAT construct. The Statement list is repeated until the search_condition expression is true.
SELECT Syntax
Retrieve rows selected from one or more tables
搜索
Exercise 6.5.1 (b) "" SELECT model, speed as gigahertz, hd as gigabytes FROM PC 只读 查询时间: 0.067s 第 1 条记录 (共 4 条)
15:07
2020/11/10

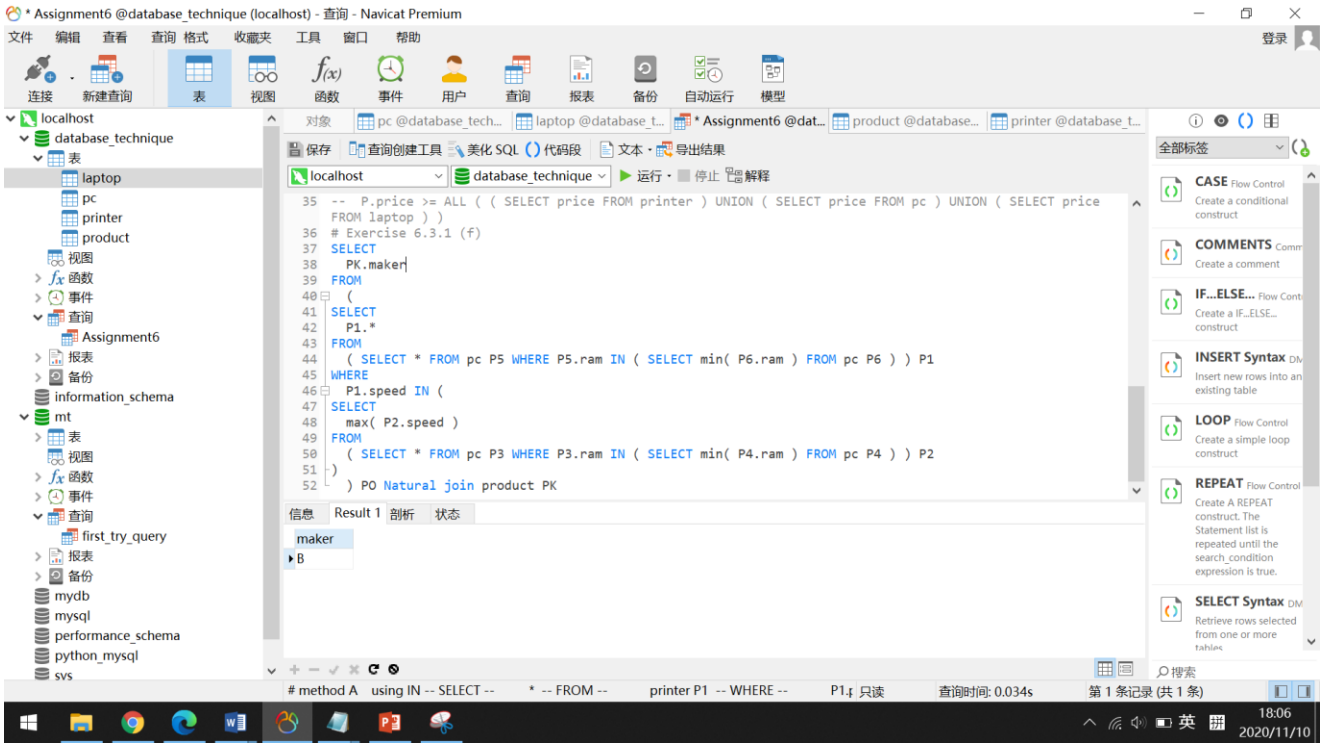

Assignment6 @database_technique (localhost) - 查询 - Navicat Premium
文件 编辑 查看 查询 格式 收藏夹 工具 窗口 帮助
登录
连接 新建查询 表 视图 函数 事件 用户 查询 报表 备份 自动运行 模型
localhost
database_technique
表
laptop
pc
printer
product
视图
函数
事件
查询
Assignment6
报表
备份
information_schema
mt
表
视图
函数
事件
查询
first_try_query
报表
备份
mydb
mysql
performance_schema
python_mysql
sys
对象
pc @database_tech...
Assignment6 @data...
product @database...
printer @database_t...
laptop @database_t...
保存 查询创建工具 美化 SQL 代码段 文本 导出结果
localhost database_technique 运行 停止 解释
19 -- price > 1500
20 # Exercise 6.5.1 (f)
21 -- SELECT
22 -- model,
23 -- hd
24 -- FROM
25 -- pc
26 -- WHERE
27 -- speed = 3.2
28 -- AND price < 2000
29 (SELECT product.model, price FROM product NATURAL JOIN pc WHERE product.maker = 'B') UNION
30 (SELECT product.model, price FROM product NATURAL JOIN laptop WHERE product.maker = 'B') UNION
31 (SELECT product.model, price FROM product NATURAL JOIN printer WHERE product.maker = 'B')
信息 Result 1 剖析 状态
model price
1004 649
1005 630
1006 1049
2007 1429
全部标签
CASE Flow Control Create a conditional construct
COMMENTS Create a comment
IF...ELSE... Create a IF...ELSE... construct
INSERT Syntax Insert new rows into an existing table
LOOP Flow Control Create a simple loop construct
REPEAT Flow Control Create A REPEAT construct. The Statement list is repeated until the search_condition expression is true.
SELECT Syntax Retrieve rows selected from one or more tables
搜索
Exercise 6.5.1 (b) -- -- SELECT -- model, -- speed as gigahertz, -- hd as gigabyt 只读
查询时间: 0.081s
第 4 条记录 (共 4 条)
英
15:23
2020/11/10

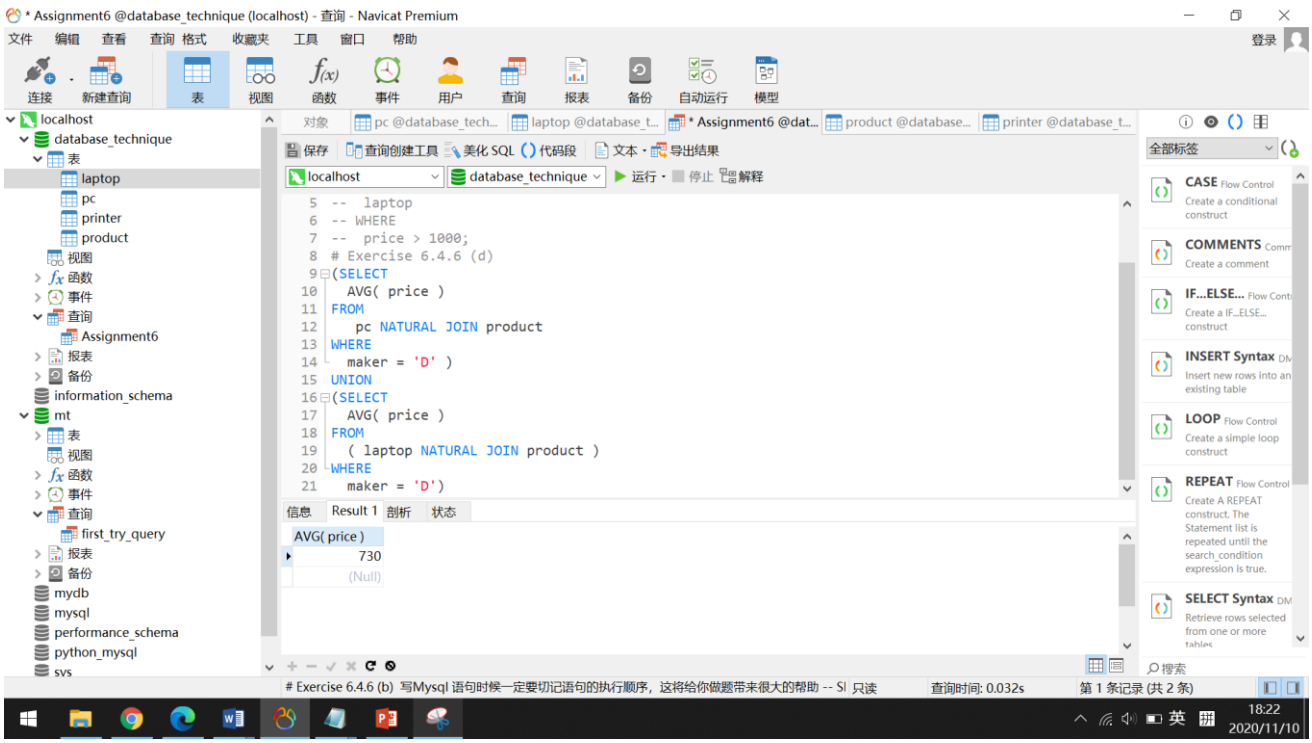

Assignment6 @database_technique (localhost) - 查询 - Navicat Premium
文件 编辑 查看 查询 格式 收藏夹 工具 窗口 帮助
登录
连接 新建查询 表 视图 函数 事件 用户 查询 报表 备份 自动运行 模型
localhost
database_technique
表
laptop
pc
printer
product
视图
函数
事件
查询
Assignment6
报表
备份
information_schema
mt
表
视图
函数
事件
查询
first_try_query
报表
备份
mydb
mysql
performance_schema
python_mysql
sys
对象
pc @database_tech...
Assignment6 @data...
product @database...
printer @database_t...
laptop @database_t...
保存 查询创建工具 美化 SQL 代码段 文本 导出结果
localhost database_technique 运行 停止 解释
30 -- (SELECT product.model, price FROM product NATURAL JOIN pc WHERE product.maker = 'B') UNION
31 -- (SELECT product.model, price FROM product NATURAL JOIN laptop WHERE product.maker = 'B') UNION
32 -- (SELECT product.model, price FROM product NATURAL JOIN printer WHERE product.maker = 'B')
33 # Exercise 6.2.2 (d)
34 SELECT DISTINCT # if the selected COLUMNS is not GROUP BY INDEX COLUMNS, then we need to use this
35 P.hd
36 FROM
37 pc P # renaming pc, otherwise the GROUP BY operation will change the original pc permanantly
38
39 GROUP BY
40 P.hd
41 HAVING
42 COUNT(P.model) >= 2 # Exercise 6.2.2 (f)
信息 Result 1 剖析 状态
hd
250
80
160
全部标签
CASE Flow Control Create a conditional construct
COMMENTS Create a comment
IF...ELSE... Create a IF...ELSE... construct
INSERT Syntax Insert new rows into an existing table
LOOP Flow Control Create a simple loop construct
REPEAT Flow Control Create A REPEAT construct. The Statement list is repeated until the search_condition expression is true.
SELECT Syntax Retrieve rows selected from one or more tables
搜索
Exercise 6.1.3 (b) -- -- SELECT -- model, -- speed as gigahertz, -- hd as gigabyt 只读
查询时间: 0.049s
第 1 条记录 (共 3 条)
英
15:56
2020/11/10

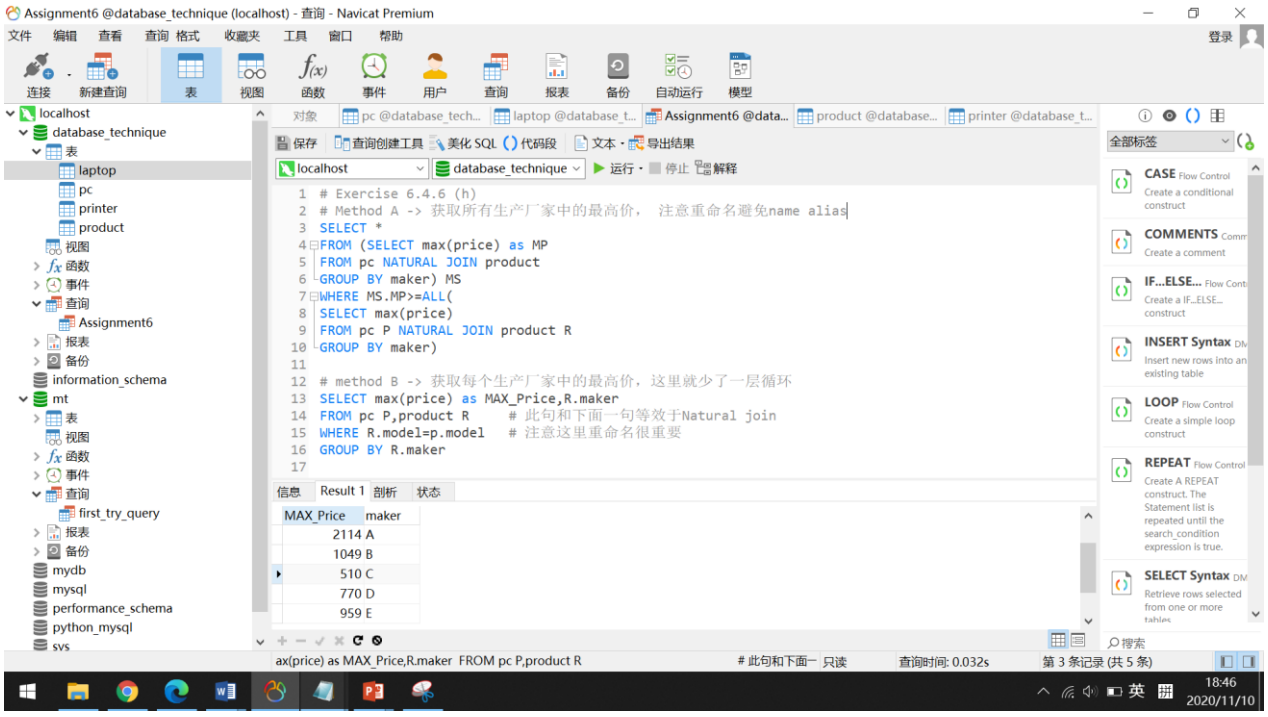

* Assignment6 @database_technique (localhost) - 查询 - Navicat Premium
文件 编辑 查看 查询 格式 收藏夹 工具 窗口 帮助
登录
连接 新建查询 表 视图 函数 事件 用户 查询 报表 备份 自动运行 模型
localhost
database_technique
表
laptop
pc
printer
product
视图
函数
事件
查询
Assignment6
报表
备份
information_schema
mt
表
视图
函数
事件
查询
first_try_query
报表
备份
mydb
mysql
performance_schema
python_mysql
sys
对象
pc @database_tech...
laptop @database_t...
* Assignment6 @dat...
product @database...
printer @database_t...
保存 查询创建工具 美化 SQL 代码段 文本 导出结果
localhost database_technique 运行 停止 解释
44 # 注意这里有个很重要的点：NATURAL JOIN，如果没有自然连接，则每一个满足speed>3的项都可以join
其它所有的项，因此最终将全部展现，此出现问题
45 SELECT DISTINCT
46 P.maker
47 FROM
48 (
49 (SELECT product.model, product.maker FROM laptop NATURAL JOIN product WHERE speed > 3) UNION
50 (SELECT product.model, product.maker FROM pc NATURAL JOIN product WHERE speed > 3)
51) P
52 GROUP BY
53 P.maker
54 HAVING
55 COUNT(P.model) >=2
信息 Result 1 剖析 状态
maker
B
全部标签
CASE Flow Control Create a conditional construct
COMMENTS Create a comment
IF...ELSE... Flow Control Create a IF...ELSE... construct
INSERT Syntax DML Insert new rows into an existing table
LOOP Flow Control Create a simple loop construct
REPEAT Flow Control Create A REPEAT construct. The Statement list is repeated until the search_condition expression is true.
SELECT Syntax DML Retrieve rows selected from one or more tables
搜索
Exercise 6.1.3 (b) -- -- SELECT -- model, -- speed as gigahertz, -- hd as gigabyt 只读
查询时间: 0.077s
第 1 条记录 (共 1 条)
16:14
2020/11/10

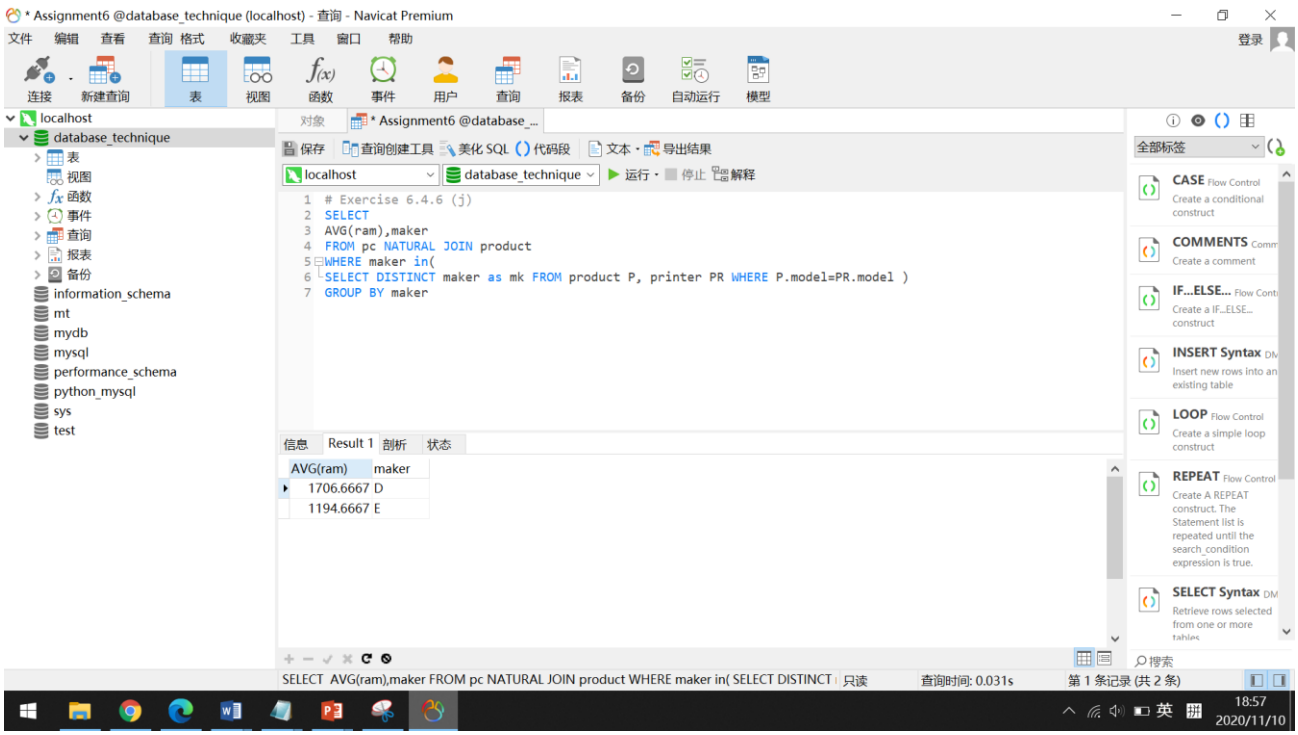

* Assignment6 @database_technique (localhost) - 查询 - Navicat Premium
文件 编辑 查看 查询 格式 收藏夹 工具 窗口 帮助
登录
连接 新建查询 表 视图 函数 事件 用户 查询 报表 备份 自动运行 模型
localhost
database_technique
表
laptop
pc
printer
product
视图
函数
事件
查询
Assignment6
报表
备份
information_schema
mt
表
视图
函数
事件
查询
first_try_query
报表
备份
mydb
mysql
performance_schema
python_mysql
sys
对象
pc @database_tech...
laptop @database_t...
* Assignment6 @dat...
product @database...
printer @database_t...
保存 查询创建工具 美化 SQL 代码段 文本 导出结果
localhost database_technique 运行 停止 解释
15 # Exercise 6.3.1 (d)
16 SELECT
17 P.model
18 FROM
19 (
20 (SELECT P1.model FROM printer P1 WHERE P1.price IN (SELECT max(P2.price) FROM printer P2)) UNION
21 (SELECT P3.model FROM laptop P3 WHERE P3.price IN (SELECT max(P4.price) FROM laptop P4)) UNION
22 (
23 (SELECT P5.model FROM pc P5 WHERE P5.price IN (SELECT max(P6.price) FROM pc P6))
24)
25) P
26
27 # Exercise 6.3.1 (f)
信息 Result 1 剖析 状态
model
3003
2001
1001
全部标签
CASE Flow Control Create a conditional construct
COMMENTS Create a comment
IF...ELSE... Flow Control Create a IF...ELSE... construct
INSERT Syntax DML Insert new rows into an existing table
LOOP Flow Control Create a simple loop construct
REPEAT Flow Control Create A REPEAT construct. The Statement list is repeated until the search_condition expression is true.
SELECT Syntax DML Retrieve rows selected from one or more tables
搜索
method A using IN -- SELECT -- * -- FROM -- printer P1 -- WHERE -- P1.p 只读
查询时间: 0.043s
第 1 条记录 (共 3 条)
16:46
2020/11/10


Assignment6 @database_technique (localhost) - 查询 - Navicat Premium
localhost
database_technique
laptop
pc
printer
product
Assignment6
first_try_query
mydb
mysql
performance_schema
python_mysql
sys
16 # method A, using in
17 SELECT
18 P.model
19 FROM
20 (
21 (SELECT P1.model FROM printer P1 WHERE P1.price IN (SELECT max(P2.price) FROM printer P2)) UNION
22 (SELECT P3.model FROM laptop P3 WHERE P3.price IN (SELECT max(P4.price) FROM laptop P4)) UNION
23 (
24 (SELECT P5.model FROM pc P5 WHERE P5.price IN (SELECT max(P6.price) FROM pc P6))
25)
26) P;
27 # method B using ALL
28 SELECT
29 P.model
30 FROM
31 (SELECT model, price FROM printer) UNION
32 (SELECT model, price FROM pc) UNION
33 (SELECT model, price FROM laptop) P
34 WHERE
35 P.price >= ALL ((SELECT price FROM printer) UNION (SELECT price FROM pc) UNION (SELECT price
FROM laptop))
model
3003
2001
1001
method A using IN -- SELECT -- * -- FROM -- printer P1 -- WHERE -- P1.price 只读 查询时间: 0.052s 第 1 条记录 (共 3 条)
CASE Flow Control Create a conditional construct
COMMENTS Create a comment
IF...ELSE... Create a IF...ELSE... construct
INSERT Syntax Insert new rows into an existing table
LOOP Flow Control Create a simple loop construct
REPEAT Flow Control Create A REPEAT construct. The Statement list is repeated until the search_condition expression is true.
SELECT Syntax Retrieve rows selected from one or more tables
17:01
2020/11/10


* Assignment6 @database_technique (localhost) - 查询 - Navicat Premium
localhost
database_technique
laptop
pc
printer
product
Assignment6
first_try_query
mydb
mysql
performance_schema
python_mysql
sys
35 -- P.price >= ALL ((SELECT price FROM printer) UNION (SELECT price FROM pc) UNION (SELECT price
FROM laptop))
36 # Exercise 6.3.1 (f)
37 SELECT
38 PK.maker
39 FROM
40 (
41 SELECT
42 P1.*
43 FROM
44 (SELECT * FROM pc P5 WHERE P5.ram IN (SELECT min(P6.ram) FROM pc P6)) P1
45 WHERE
46 P1.speed IN (
47 SELECT
48 max(P2.speed)
49 FROM
50 (SELECT * FROM pc P3 WHERE P3.ram IN (SELECT min(P4.ram) FROM pc P4)) P2
51)
52) PO Natural join product PK
maker
B
method A using IN -- SELECT -- * -- FROM -- printer P1 -- WHERE -- P1.price 只读 查询时间: 0.034s 第 1 条记录 (共 1 条)
CASE Flow Control Create a conditional construct
COMMENTS Create a comment
IF...ELSE... Create a IF...ELSE... construct
INSERT Syntax Insert new rows into an existing table
LOOP Flow Control Create a simple loop construct
REPEAT Flow Control Create A REPEAT construct. The Statement list is repeated until the search_condition expression is true.
SELECT Syntax Retrieve rows selected from one or more tables
18:06
2020/11/10


* Assignment6 @database_technique (localhost) - 查询 - Navicat Premium
-- laptop
-- WHERE
-- price > 1000;
Exercise 6.4.6 (d)
(SELECT
AVG(price)
FROM
pc NATURAL JOIN product
WHERE
maker = 'D')
UNION
(SELECT
AVG(price)
FROM
(laptop NATURAL JOIN product)
WHERE
maker = 'D')
AVG(price)
730
(Null)
Exercise 6.4.6 (b) 写Mysql 语句时候一定要切记语句的执行顺序，这将给你做题带来很大的帮助 -- Sl 只读
查询时间: 0.032s
第 1 条记录 (共 2 条)


Assignment6 @database_technique (localhost) - 查询 - Navicat Premium
Exercise 6.4.6 (h)
Method A -> 获取所有生产厂家中的最高价， 注意重命名避免name alias
SELECT *
FROM (SELECT max(price) as MP
FROM pc NATURAL JOIN product
GROUP BY maker) MS
WHERE MS.MP>=ALL(
SELECT max(price)
FROM pc P NATURAL JOIN product R
GROUP BY maker)
method B -> 获取每个生产厂家中的最高价，这里就少了一层循环
SELECT max(price) as MAX_Price,R.maker
FROM pc P,product R # 此句和下面一句等效于Natural join
WHERE R.model=p.model # 注意这里重命名很重要
GROUP BY R.maker
MAX_Price maker
2114 A
1049 B
510 C
770 D
959 E
ax(price) as MAX_Price,R.maker FROM pc P,product R
此句和下面一 只读
查询时间: 0.032s
第 3 条记录 (共 5 条)



保存 查询创建工具 美化 SQL 代码段 文本 导出结果

localhost database_technique 运行 停止 解释

```
1  INSERT INTO product SELECT
2  'm',
3  model+ 1100,
4  "laptop"
5  FROM
6    product
7  WHERE
8    type = 'pc';
9
10 INSERT INTO laptop SELECT
11 model + 1100,
12 speed,
13 ram,
14 hd,
15 17,
16 price + 500
17 FROM
18   PC;
```

信息 剖析 状态

```
INSERT INTO laptop SELECT
model + 1100,
speed,
ram,
hd,
17,
price + 500
FROM
       PC
> Affected rows: 26
> 时间: 0.025s
```

```sql
DELETE FROM laptop  WHERE model IN (
 # layer 2 # 电脑的maker 在几家里面选择
SELECT R2.model FROM product R2 WHERE R2.maker IN (
 # layer 3
 SELECT DISTINCT R.maker FROM product R WHERE R.maker NOT IN
 ( SELECT R2.maker FROM product R2 WHERE R2.type = 'printer' )
   )
)
DELETE FROM product  WHERE model IN (SELECT K_model from (
 # layer 2 # 先通过再更新product值
SELECT R3.model as K_model FROM product R3 WHERE R3.maker IN (
 # layer 3
 SELECT DISTINCT R4.maker FROM product R4 WHERE R4.maker NOT IN
 ( SELECT R3.maker FROM product R3 WHERE R3.type = 'printer' )
   )
) AS a )
```

信息 剖析 状态

```
DELETE FROM product  WHERE model IN (SELECT K_model from (
# layer 2 # 先通过再更新product值
SELECT R3.model as K_model FROM product R3 WHERE R3.maker IN (
# layer 3
SELECT DISTINCT R4.maker FROM product R4 WHERE R4.maker NOT IN
( SELECT R3.maker FROM product R3 WHERE R3.type = 'printer' )
      )
) AS a )
> Affected rows: 28
> 时间: 0.05s
```

```sql
# Exercise 6.5.1 (f)
UPDATE pc
SET ram = ram * 2,
hd = hd + 10
```

信息 剖析 状态

```
# Exercise 6.5.1 (f)
UPDATE pc set ram=ram*2,hd=hd+10
> Affected rows: 26
> 时间: 0.094s
```

注意： Exercise 6.5.1 (d)中， 先通过 select product 选择出 model，再更新 product 值,这个在 Mysql 下是不行的，在其它 SQL 平台上可以

解决方法：select 的结果再通过一个中间表 select 多一次，就可以避免这个错误 --> SELECT K_model from (原始内容) AS a

Code →

```
# Exercise 6.1.3 (b)

SELECT
	model,
	speed as gigahertz,
	hd as gigabytes
FROM
	PC
WHERE
	price <= 1000

# Exercise 6.1.3 (d)
SELECT
	model,
	ram,
	screen
FROM
	laptop
WHERE
	price > 1500

# Exercise 6.1.3 (f)
SELECT
	model,
	hd
FROM
	pc
WHERE
	speed = 3.2
	AND price < 2000



# Exercise 6.2.2 (b)
( SELECT product.model, price FROM product NATURAL JOIN pc WHERE product.maker = 'B' ) UNION
( SELECT product.model, price FROM product NATURAL JOIN laptop WHERE product.maker = 'B' ) UNION
( SELECT product.model, price FROM product NATURAL JOIN printer WHERE product.maker = 'B' )
# Exercise 6.2.2 (d)
SELECT DISTINCT# if the selected COLUMNS is not GROUP BY INDEX COLUMNS, then
```

we need to use this
P.hd
FROM
pc P # renaming pc, otherwise the GROUP BY operation will change the original pc permanantly

GROUP BY
P.hd
HAVING
COUNT(P.model) >= 2

Exercise 6.2.2 (f)
注意这里有个很重要的点：NATURAL JOIN, 如果没有自然连接，则每一个满足 speed>3 的项都可以 join 其它所有的项，因此最终将全部展现，此出现问题
SELECT DISTINCT
P.maker
FROM
(
(SELECT product.model, product.maker FROM laptop NATURAL JOIN product WHERE speed > 3) UNION
(SELECT product.model, product.maker FROM pc NATURAL JOIN product WHERE speed > 3)
) P
GROUP BY
P.maker
HAVING
COUNT(P.model) >=2

Exercise 6.3.1 (b)
method A using IN
SELECT
*
FROM
printer P1
WHERE
P1.price IN (SELECT max(P2.price) FROM printer P2);
method B, using ALL
SELECT
*
FROM
printer P1
WHERE
P1.price >= ALL (SELECT P2.price FROM printer P2)

```
# Exercise 6.3.1 (d)
# method A, using in
SELECT
	P.model
FROM
	(
	( SELECT P1.model FROM printer P1 WHERE P1.price IN ( SELECT max( P2.price ) FROM printer P2 ) ) UNION
	( SELECT P3.model FROM laptop P3 WHERE P3.price IN ( SELECT max( P4.price ) FROM laptop P4 ) ) UNION
	(
	( SELECT P5.model FROM pc P5 WHERE P5.price IN ( SELECT max( P6.price ) FROM pc P6 ) )
	)
	) P;
-- # method B using ALL
SELECT
	P.model
FROM
	( SELECT model, price FROM printer ) UNION
	( SELECT model, price FROM pc ) UNION
	( SELECT model, price FROM laptop ) P
WHERE
	P.price >= ALL ( ( SELECT price FROM printer ) UNION ( SELECT price FROM pc ) UNION ( SELECT price FROM laptop ) )

# Exercise 6.3.1 (f)
SELECT
	PK.maker
FROM
	(
SELECT
	P1.*
FROM
	( SELECT * FROM pc P5 WHERE P5.ram IN ( SELECT min( P6.ram ) FROM pc P6 ) ) P1
WHERE
	P1.speed IN (
SELECT
	max( P2.speed )
FROM
	( SELECT * FROM pc P3 WHERE P3.ram IN ( SELECT min( P4.ram ) FROM pc P4 ) ) P2
)
```

```
    ) PO Natural join product PK
```

Exercise 6.4.6 (b)　写 Mysql 语句时候一定要切记语句的执行顺序，这将给你做题带来很大的帮助

```
SELECT
    AVG( speed )
FROM
    laptop
WHERE
    price > 1000;

# Exercise 6.4.6 (d)
(SELECT
    AVG( price )
FROM
     pc NATURAL JOIN product
WHERE
    maker = 'D' )
UNION
(SELECT
    AVG( price )
FROM
    ( laptop NATURAL JOIN product )
WHERE
    maker = 'D')

# Exercise 6.4.6 (f)
SELECT
    maker,
    AVG( screen )
FROM
    laptop
    NATURAL JOIN product
GROUP BY
    maker # Exercise 6.4.6 (h)
# Method A ->  获取所有生产厂家中的最高价，  注意重命名避免 name alias
SELECT
    *
FROM
    ( SELECT max( price ) AS MP FROM pc NATURAL JOIN product GROUP BY maker )
MS
```

```
WHERE
    MS.MP >= ALL ( SELECT max( price ) FROM pc P NATURAL JOIN product R GROUP BY maker ) # method B -> 获取每个生产厂家中的最高价，这里就少了一层循环
SELECT
    max( price ) AS MAX_Price,
    R.maker
FROM
    pc P,
    product R #  此句和下面一句等效于 Natural join

WHERE
    R.model = p.model #  注意这里重命名很重要

GROUP BY
    R.maker

# Exercise 6.4.6 (j)
SELECT
    AVG( ram ),
    maker
FROM
    pc
    NATURAL JOIN product
WHERE
    maker IN ( SELECT DISTINCT maker AS mk FROM product P, printer PR WHERE P.model = PR.model )
GROUP BY
Maker



# Exercise 6.5.1 (b)
INSERT INTO product SELECT
'm',
model+ 1100,
"laptop"
FROM
    product
WHERE
    type = 'pc';

INSERT INTO laptop SELECT
model + 1100,
speed,
```

```
ram,
hd,
17,
price + 500
FROM
    PC;

# Exercise 6.5.1 (d)
# layer 1   # 删除特定笔记本电脑
DELETE FROM laptop   WHERE model IN (
# layer 2 # 电脑的 maker 在几家里面选择
SELECT R2.model FROM product R2 WHERE R2.maker IN (
# layer 3
SELECT DISTINCT R.maker FROM product R WHERE R.maker NOT IN
( SELECT R2.maker FROM product R2 WHERE R2.type = 'printer' )
    )
)
DELETE FROM product   WHERE model IN (
# layer 2 # 先通过 select product 选择出 model, 再更新 product 值,这个在 Mysql 下是不行的
# 解决方法：select 的结果再通过一个中间表 select 多一次，就可以避免这个错误   --> SELECT K_model from (原始内容) AS a
SELECT K_model from (
SELECT R3.model as K_model FROM product R3 WHERE R3.maker IN (
# layer 3
SELECT DISTINCT R4.maker FROM product R4 WHERE R4.maker NOT IN
( SELECT R3.maker FROM product R3 WHERE R3.type = 'printer' ) ) ) AS a
)

# Exercise 6.5.1 (f)
UPDATE pc
SET ram = ram * 2,
hd = hd + 10
```