SQL*Plus 기반의 SQL 질의 언어 실습1

빅 데이터 마이닝 연구실
컴퓨터공학과
한양대학교

목 차

- SQL*Plus 질의
- 테이블 스키마정의, 내용, 입력
- SQL 질의 언어 실습

SQL*Plus 질의

- Select : 질의의 결과 속성들을 나열하는 데 사용한다.
- From : 질의를 수행하기 위해 접근 해야 하는 테이블 들을 나열한다.
- Where : From 절에 있는 테이블 속성들을 포함하는 조건이 [illegible]

– 컴퓨터공학과에서 급여가 $70,000가 넘는 모든 교수의 이름을 구하라.

```
SQL> select * from instructor;

ID         NAME                                 DEPT_NAME                                 SALARY
---------- ------------------------------------ ---------------------------------------- ----------
10101      Srinivasan                           Comp. Sci.                                 65000
12121      Wu                                   Finance                                    90000
15151      Mozart                               Music                                      40000
22222      Einstein                             Physics                                    95000
32343      El Said                              History                                    60000
33456      Gold                                 Physics                                    87000
45565      Katz                                 Comp. Sci.                                 75000
58583      Califieri                            History                                    62000
76543      Singh                                Finance                                    80000
76766      Crick                                Biology                                    72000
83821      Brandt                               Comp. Sci.                                 92000
98345      Kim                                  Elec. Eng.                                 80000

12 rows selected.

SQL> select name
  2  from instructor
  3  where dept_name = 'Comp. Sci.' and salary > 70000;

NAME
------------------------------------
Katz
Brandt
```

SQL*Plus 질의 (cont'd)

- 카티션 곱 : 두 릴레이션 으로부터 *가능한 모든 투플*의 쌍을 출력한다.

  – 컴퓨터공학과의 교수와 모든 수업코드의 조합을 출력하라.

```
SQL> select name, course_id
  2  from instructor, teaches
  3  where instructor.dept_name = 'Comp. Sci.';

NAME                                 COURSE_ID
------------------------------------ ----------
Srinivasan                           CS-101
Srinivasan                           CS-315
Srinivasan                           CS-347
Srinivasan                           FIN-201
Srinivasan                           MU-199
Srinivasan                           PHY-101
Srinivasan                           HIS-351
Srinivasan                           CS-101
Srinivasan                           CS-319
Srinivasan                           BIO-101
Srinivasan                           BIO-301
Srinivasan                           CS-190
Srinivasan                           CS-190
Srinivasan                           CS-319
Srinivasan                           EE-181
Katz                                 CS-101
Katz                                 CS-315
Katz                                 CS-347
Katz                                 FIN-201
Katz                                 MU-199
```

```
SQL> select course_id
  2  from teaches;

COURSE_ID
----------
CS-101
CS-315
CS-347
FIN-201
MU-199
PHY-101
HIS-351
CS-101
CS-319
BIO-101
BIO-301
CS-190
CS-190
CS-319
EE-181

15 rows selected.

SQL>
```

SQL*Plus 질의 (cont'd)

- 자연 조인 : 두 릴레이션의 스키마에서 나타나는 *속성의 값이 같은 투플의 짝만*을 고려한다.

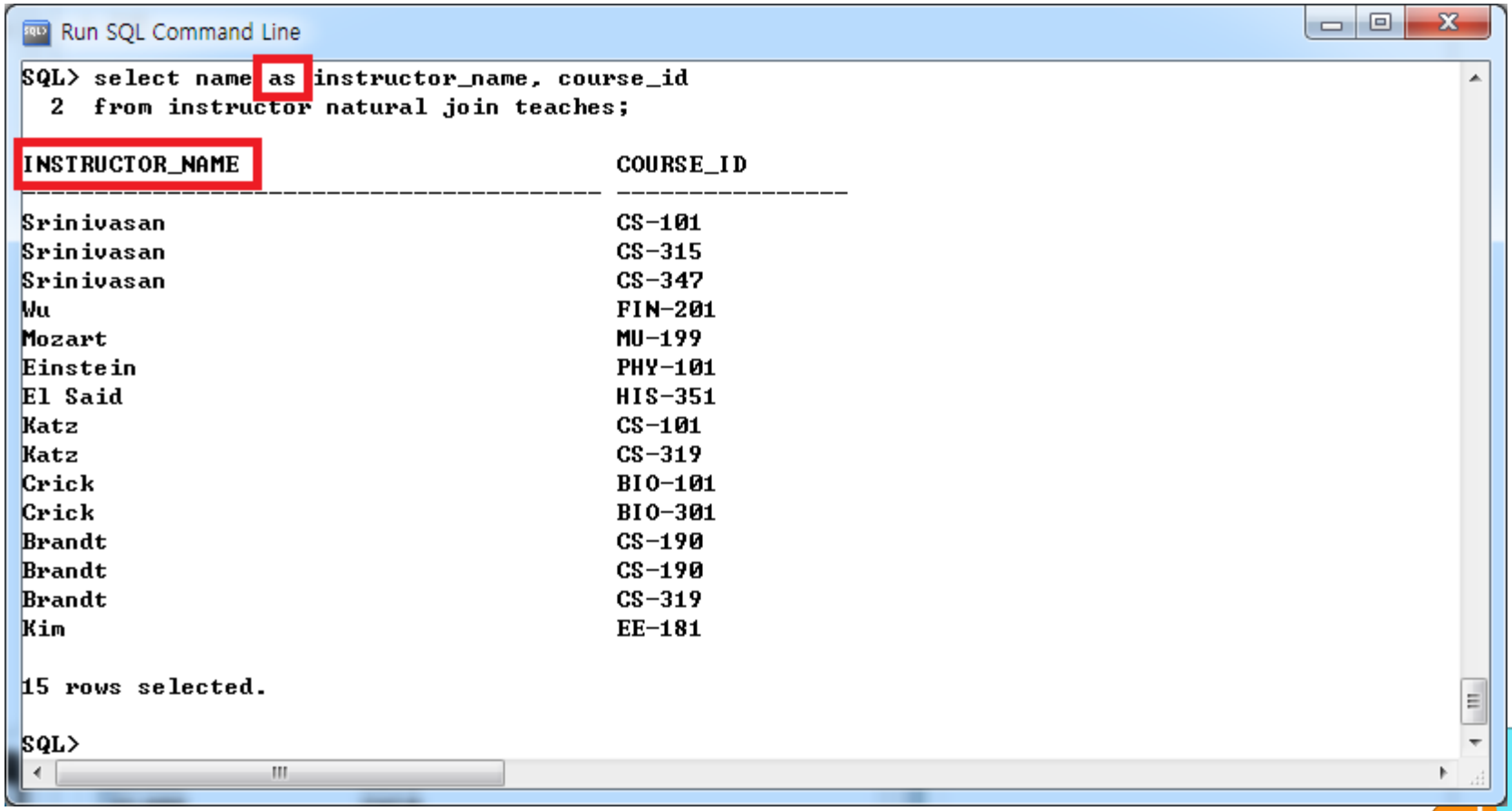
– 컴퓨터공학과 수업을 하는 모든 교수에 대해, 그들의 이름과 그들이 가르치는 모든 수업의 수업 아이디를 구하라.

```
SQL> select name, course_id
  2  from instructor natural join teaches
  3  where dept_name = 'Comp. Sci.';

NAME                                 COURSE_ID
------------------------------------ ----------
Srinivasan                           CS-101
Srinivasan                           CS-315
Srinivasan                           CS-347
Katz                                 CS-101
Katz                                 CS-319
Brandt                               CS-190
Brandt                               CS-190
Brandt                               CS-319

8 rows selected.

SQL>
```

```
SQL> select name, course_id
  2  from instructor, teaches
  3  where instructor.ID = teaches.ID
  4  and instructor.dept_name = 'Comp. Sci.';

NAME                                 COURSE_ID
------------------------------------ ----------
Srinivasan                           CS-101
Srinivasan                           CS-315
Srinivasan                           CS-347
Katz                                 CS-101
Katz                                 CS-319
Brandt                               CS-190
Brandt                               CS-190
Brandt                               CS-319

8 rows selected.

SQL>
```

SQL*Plus 질의 (cont'd)

- As : 결과 릴레이션에서 속성의 이름을 *재명명* 한다.
  (from절에서 사용할 때는 한 칸 띄고 as없이 입력하면 된다.)



SQL*Plus 질의 (cont'd)

- Like : *패턴 일치*를 수행할 수 있다.
  퍼센트 (%) : 어떠한 부분 *문자열*과도 일치한다.
  밑줄 (_) : 어떠한 한 *문자*와도 일치한다.

  – 이름에 'tso'이라는 부분 문자열이 포함 된 건물의 모든 학과의 이름을 구하라.

```
SQL> select * from department;

DEPT_NAME                            BUILDING                        BUDGET
------------------------------------ ------------------------------ ----------
Biology                              Watson                          90000
Comp. Sci.                           Taylor                         100000
Elec. Eng.                           Taylor                          85000
Finance                              Painter                        120000
History                              Painter                         50000
Music                                Packard                         80000
Physics                              Watson                          70000

7 rows selected.

SQL> select dept_name
  2  from department
  3  where building like '%tso%';

DEPT_NAME
------------------------------------
Biology
Physics

SQL> _
```

SQL*Plus 질의 (cont'd)

- Group by : 모든 속성이 같은 값을 가지는 투플들을 *하나의 그룹*으로 만들어 준다.

– 각 학과의 평균 급여를 구하라

```
SQL> select dept_name, salary
  2  from instructor;

DEPT_NAME                                SALARY
---------------------------------------- ----------
Comp. Sci.                                65000
Finance                                   90000
Music                                     40000
Physics                                   95000
History                                   60000
Physics                                   87000
Comp. Sci.                                75000
History                                   62000
Finance                                   80000
Biology                                   72000
Comp. Sci.                                92000
Elec. Eng.                                80000

12 rows selected.

SQL> _
```

```
SQL> select dept_name, avg(salary) as avg_salary
  2  from instructor
  3  group by dept_name;

DEPT_NAME                                AVG_SALARY
---------------------------------------- ----------
Elec. Eng.                                80000
Physics                                   91000
Comp. Sci.                             77333.3333
Finance                                   85000
Biology                                   72000
Music                                     40000
History                                   61000

7 rows selected.
```

SQL*Plus 질의 (cont'd)

- Order by : 질의의 결과 투플들이 *정렬된 순서*로 나타나도록 한다.

  – 교수급여는 내림차순으로 교수이름은 오름차순으로 급여와 이름을 구하라.

```
SQL> select salary, name from instructor;

    SALARY NAME
---------- --------------------------------
     65000 Srinivasan
     90000 Wu
     40000 Mozart
     95000 Einstein
     60000 El Said
     87000 Gold
     75000 Katz
     62000 Califieri
     80000 Singh
     72000 Crick
     92000 Brandt
     80000 Kim

12 rows selected.
```

```
SQL> select salary, name from instructor
  2  order by salary desc, name asc;

    SALARY NAME
---------- --------------------------------
     95000 Einstein
     92000 Brandt
     90000 Wu
     87000 Gold
     80000 Kim
     80000 Singh
     75000 Katz
     72000 Crick
     65000 Srinivasan
     62000 Califieri
     60000 El Said
     40000 Mozart

12 rows selected.
```

SQL*Plus 질의 (cont'd)

- Having by : group by 절의 기준 항목이나, 집계 함수를 이용한 *조건 표시*를 할 수 있다.

– 평균 급여가 $42,000을 넘는 학과의 교수들의 평균 급여를 구하라.

```
SQL> select dept_name, avg(salary) as avg_salary
  2  from instructor
  3  group by dept_name;

DEPT_NAME                              AVG_SALARY
-------------------------------------- ----------
Elec. Eng.                                  80000
Physics                                     91000
Comp. Sci.                             77333.3333
Finance                                     85000
Biology                                     72000
Music                                       40000
History                                     61000

7 rows selected.
```

```
SQL> select dept_name, avg(salary) as avg_salary
  2  from instructor
  3  group by dept_name
  4  having avg(salary) > 42000;

DEPT_NAME                              AVG_SALARY
-------------------------------------- ----------
Elec. Eng.                                  80000
Physics                                     91000
Comp. Sci.                             77333.3333
Finance                                     85000
Biology                                     72000
History                                     61000

6 rows selected.

SQL>
```

SQL 질의 언어 실습 (Subqueries)

- 하위 쿼리 구문

```
SELECT      select_list
FROM        table
WHERE       expr operator
                    (SELECT     select_list
                     FROM       table) ;
```

- 사용법
  - 하위 쿼리는 group by 절 안에서 사용할 수 없습니다.

단일 행 서브쿼리	다중 행 서브쿼리
=	IN
>	> ANY , > ALL
<	< ANY , < ALL

SQL 질의 언어 실습 (Subqueries)

- 단일 행 서브쿼리
  - Select 문장에서 오직 하나의 행(값)만 반환하는 쿼리이다.
  - 단일 행 연산자(=,>, >=, <, <=, <>, !=) 만 사용된다.

```
Ex)  SELECT   last_name, job_id, salary
     FROM     employees
     WHERE    job_id =
                    (SELECT   job_id
                     FROM     employees
                     WHERE    employee_id = 141)
     AND      Salary >
                    SELECT   salary
                     FROM     employees
                     WHERE    employee_id = 143) ;
```

ST_CLERK

2600

LAST_NAME	JOB_ID	SALARY
Rajs	ST_CLERK	35000
Davies	ST_CLERK	31000

SQL 질의 언어 실습 (Subqueries)

- 다중 행 서브 쿼리
  - 하나 이상의 행을 반환하는 서브쿼리이다.
  - 단일 행 연산자를 사용하지 못하며, 다중 행 연산자(IN, NOT IN, ANY, ALL, EXISTS)만 사용이 가능하다.

```
SELECT      employee_id, last_name, job_id, salary
FROM        employees
WHERE       salary < ANY
                    (SELECT     salary
                     FROM       employees
                     WHERE      job_id = 'IT_PROG' )
AND         job_id <> 'IT_PROG' ;
```

9000, 6000, 4200

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
124	Mourgos	ST_MAN	5800
141	Rajs	ST_CLERK	3500
142	Davies	ST_CLERK	3100
143	Matos	ST_CLERK	2600
144	Vargas	ST_CLERK	2500

....

```
SELECT      employee_id, last_name, job_id, salary
FROM        employees
WHERE       salary < ALL
                    (SELECT     salary
                     FROM       employees
                     WHERE      job_id = 'IT_PROG' )
AND         job_id <> 'IT_PROG' ;
```

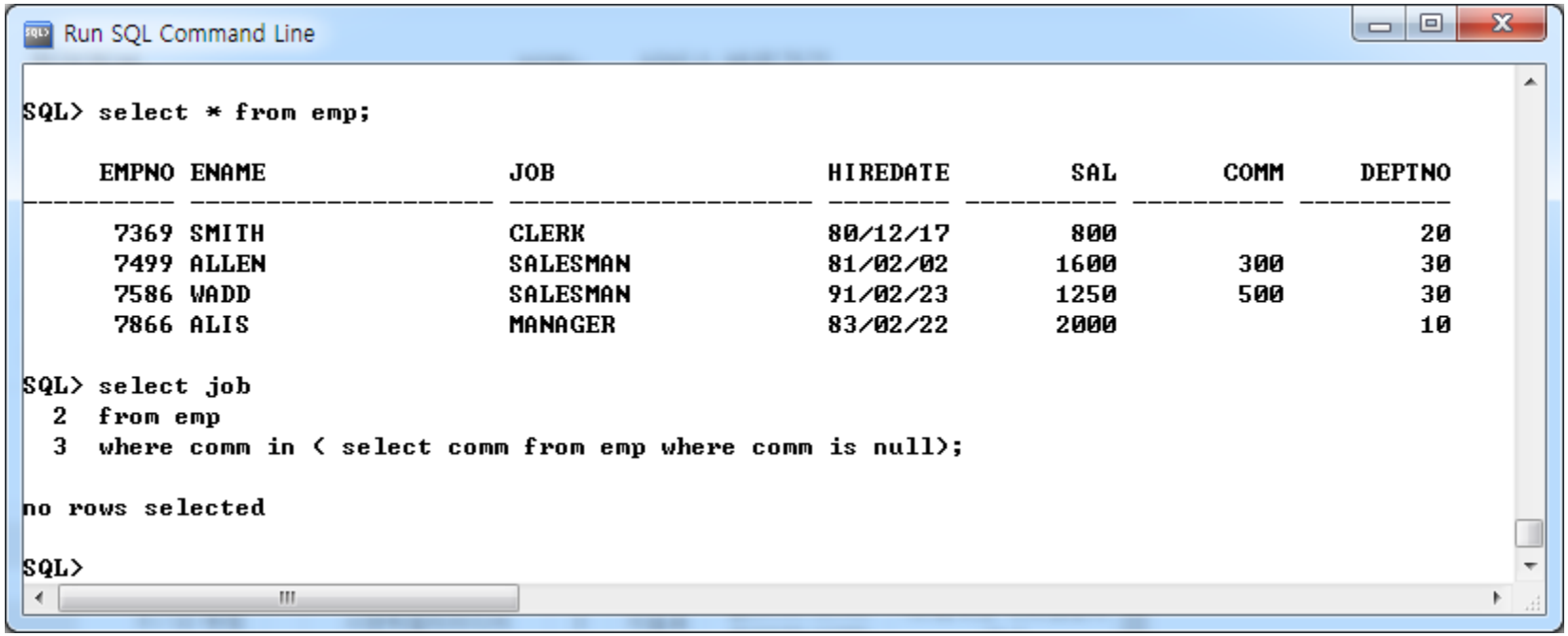
9000, 6000, 4200

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
141	Rajs	ST_CLERK	3500
142	Davies	ST_CLERK	3100
143	Matos	ST_CLERK	2600
144	Vargas	ST_CLERK	2500

....

SQL 질의 언어 실습 (Subqueries)

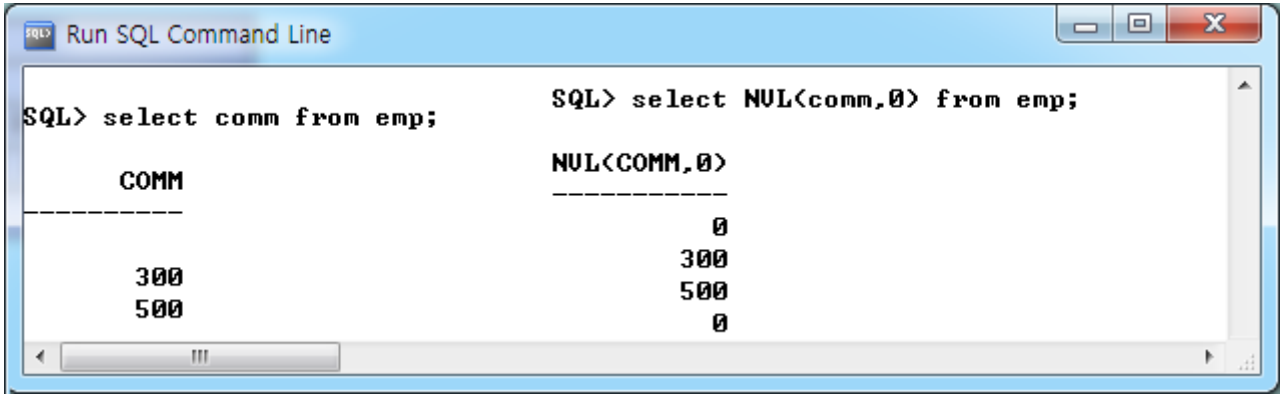
- 서브쿼리 안에 널 값
  - 서브쿼리에서 NULL 값이 반환 되면 주 쿼리 에서는 어떠한 행도 반환되지 않는다.



SQL 질의 언어 실습 (Subqueries)

- NVL 함수는 NULL을 0 또는 다른 값으로 변환하기 위한 함수이다.

- NVL 사용법
  - NVL(컬럼명, 널값을 바꿀값)



테이블 스키마 정의

- 테이블 스키마

  Product (maker, model, type)

  PC (model, speed, ram, hd, cd, price)

  Laptop (model, speed, ram, hd, screen, price)

  Printer (model, color, type, price)

  - model ➔ primary key
  - Data Type
    - maker, type, cd, color : char(20)
    - model, speed, ram, price : integer, number(38)
    - hd, screen : number

테이블 내용

Product

maker	model	type	maker	model	type
A	1001	pc	D	2001	laptop
A	1002	pc	D	2002	laptop
A	1003	pc	D	2003	laptop
B	1004	pc	D	3001	printer
B	1005	pc	D	3003	printer
B	3002	printer	E	2004	laptop
B	3004	printer	E	2008	laptop
C	1006	pc	F	2005	laptop
C	1007	pc	G	2006	laptop
D	1008	pc	G	2007	laptop
D	1009	pc	H	3005	printer
D	1010	pc	I	3006	printer

PC

model	speed	ram	hd	cd	price
1001	133	16	1.6	6x	1595
1002	120	16	1.6	6x	1399
1003	166	24	2.5	6x	1899
1004	166	32	2.5	8x	1999
1005	166	16	2.0	8x	1999
1006	200	32	3.1	8x	2099
1007	200	32	3.2	8x	2349
1008	180	32	2.0	8x	3699
1009	200	32	2.5	8x	2599
1010	160	16	1.2	8x	1495

테이블 내용

Laptop

model	*speed*	*ram*	*hd*	*screen*	*price*
2001	100	20	1.10	9.5	1999
2002	117	12	0.75	11.3	2499
2003	117	32	1.00	10.4	3599
2004	133	16	1.10	11.2	3499
2005	133	16	1.00	11.3	2599
2006	120	8	0.81	12.1	1999
2007	150	16	1.35	12.1	4799
2008	120	16	1.10	12.1	2099

Printer

model	*color*	*type*	*price*
3001	true	ink-jet	1999
3002	true	ink-jet	2499
3003	false	laser	3599
3004	false	laser	3499
3005	false	ink-jet	2599
3006	true	dry	1999

Table 생성

schema.sql - 메모장

파일(F) 편집(E) 서식(O) 도움말(H)

```
create table Product (
        maker           char(20)        not null,
        model           integer         not null,
        type            char(20)        not null,
        primary key (model)
);

create table PC (
        model           integer         not null,
        speed           integer         not null,
        ram             integer         not null,
        hd              number          not null,
        cd              char(20)        not null,
        price           integer         not null,
        primary key (model)
);

create table Laptop (
        model           integer         not null,
        speed           integer         not null,
        ram             integer         not null,
        hd              number          not null,
        screen          number          not null,
        price           integer         not null,
        primary key (model)
);

create table Printer (
        model           integer         not null,
        color           char(20)        not null,
        type            char(20)        not null,
        price           integer         not null,
        primary key (model)
);
```

Insert 및 drop 문

PC.sql - 메모장

파일(F) 편집(E) 서식(O) 도움말(H)

```
insert into PC values (1001 ,133, 16, 1.6, '6x',1595);
insert into PC values (1002 ,120, 16, 1.6, '6x',1399);
insert into PC values (1003 ,166, 24, 2.5, '6x',1899);
insert into PC values (1004 ,166, 32, 2.5, '8x',1999);
insert into PC values (1005 ,166, 16, 2.0, '8x',1999);
insert into PC values (1006 ,200, 32, 3.1, '8x',2099);
insert into PC values (1007 ,200, 32, 3.2, '8x',2349);
insert into PC values (1008 ,180, 32, 2.0, '8x',3699);
insert into PC values (1009 ,200, 32, 2.5, '8x',2599);
insert into PC values (1010 ,160, 16, 1.2, '8x',1495);
```

Laptop.sql - 메모장

파일(F) 편집(E) 서식(O) 도움말(H)

```
insert into Laptop values (2001, 100, 20, 1.10,  9.5, 1999);
insert into Laptop values (2002, 117, 12, 0.75, 11.3, 2499);
insert into Laptop values (2003, 117, 32, 1.00, 10.4, 3599);
insert into Laptop values (2004, 133, 16, 1.10, 11.2, 3499);
insert into Laptop values (2005, 133, 16, 1.00, 11.3, 2599);
insert into Laptop values (2006, 120,  8, 0.81, 12.1, 1999);
insert into Laptop values (2007, 150, 16, 1.35, 12.1, 4799);
insert into Laptop values (2008, 120, 16, 1.10, 12.1, 2099);
```

Printer.sql - 메모장

파일(F) 편집(E) 서식(O) 도움말(H)

```
insert into PRINTER values (3001 ,  'true', 'ink-jet', 1999);
insert into PRINTER values (3002 ,  'true', 'ink-jet', 2499);
insert into PRINTER values (3003 , 'false',   'laser', 3599);
insert into PRINTER values (3004 , 'false',   'laser', 3499);
insert into PRINTER values (3005 , 'false', 'ink-jet', 2599);
insert into PRINTER values (3006 ,  'true',     'dry', 1999);
```

droptable.sql - 메모장

파일(F) 편집(E) 서식(O) 도움말(H)

```
drop table Product;
drop table PC;
drop table Laptop;
drop table Printer;
```

product.sql - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
insert into Product values ('A', 1001, 'pc');
insert into Product values ('A', 1002, 'pc');
insert into Product values ('A', 1003, 'pc');
insert into Product values ('B', 1004, 'pc');
insert into Product values ('B', 1006, 'pc');
insert into Product values ('B', 3002, 'printer');
insert into Product values ('B', 3004, 'printer');
insert into Product values ('C', 1005, 'pc');
insert into Product values ('C', 1007, 'pc');
insert into Product values ('D', 1008, 'pc');
insert into Product values ('D', 1009, 'pc');
insert into Product values ('D', 1010, 'pc');
insert into Product values ('D', 2001, 'laptop');
insert into Product values ('D', 2002, 'laptop');
insert into Product values ('D', 2003, 'laptop');
insert into Product values ('D', 3001, 'printer');
insert into Product values ('D', 3003, 'printer');
insert into Product values ('E', 2004, 'laptop');
insert into Product values ('E', 2008, 'laptop');
insert into Product values ('F', 2005, 'laptop');
insert into Product values ('G', 2006, 'laptop');
insert into Product values ('G', 2007, 'laptop');
insert into Product values ('H', 3005, 'printer');
insert into Product values ('I', 3006, 'printer');
```

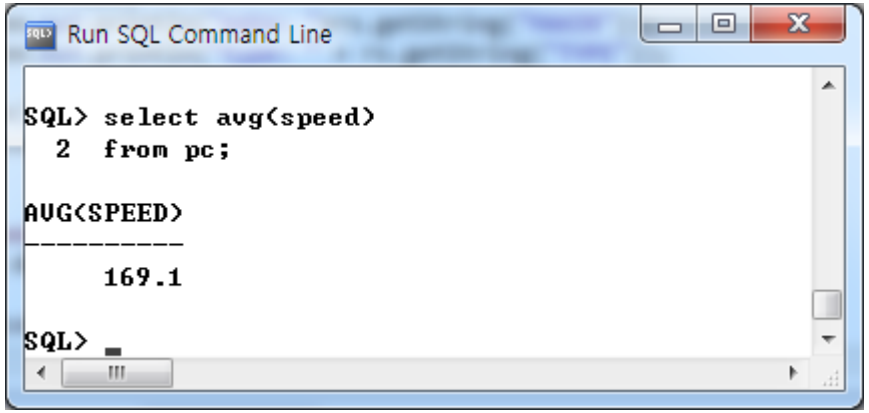
SQL 질의 언어 실습

- SQL> start schema
- SQL> start product
- SQL> start pc
- SQL> start laptop
- SQL> start printer

SQL 질의 언어 실습 (SELECT 문)

- 문제1) PC의 평균 속도를 구하라.
  (hint: avg이용)

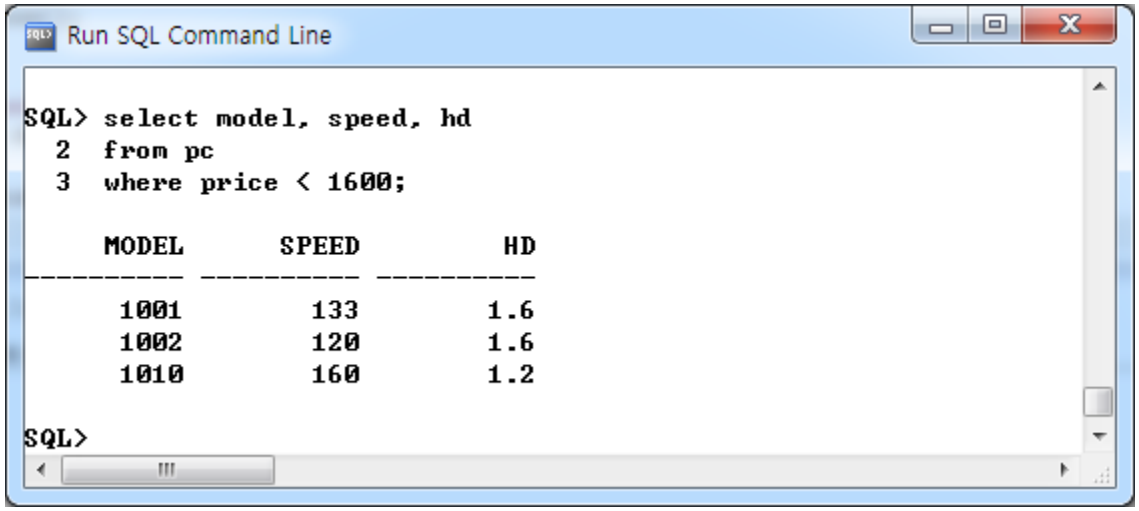
SQL 질의 언어 실습 (SELECT 문)



SQL 질의 언어 실습 (SELECT 문)

- 문제 2) 가격이 $1600 미만인 모든 PC의 모델 번호, 속도, 하드 디스크 용량을 구하라

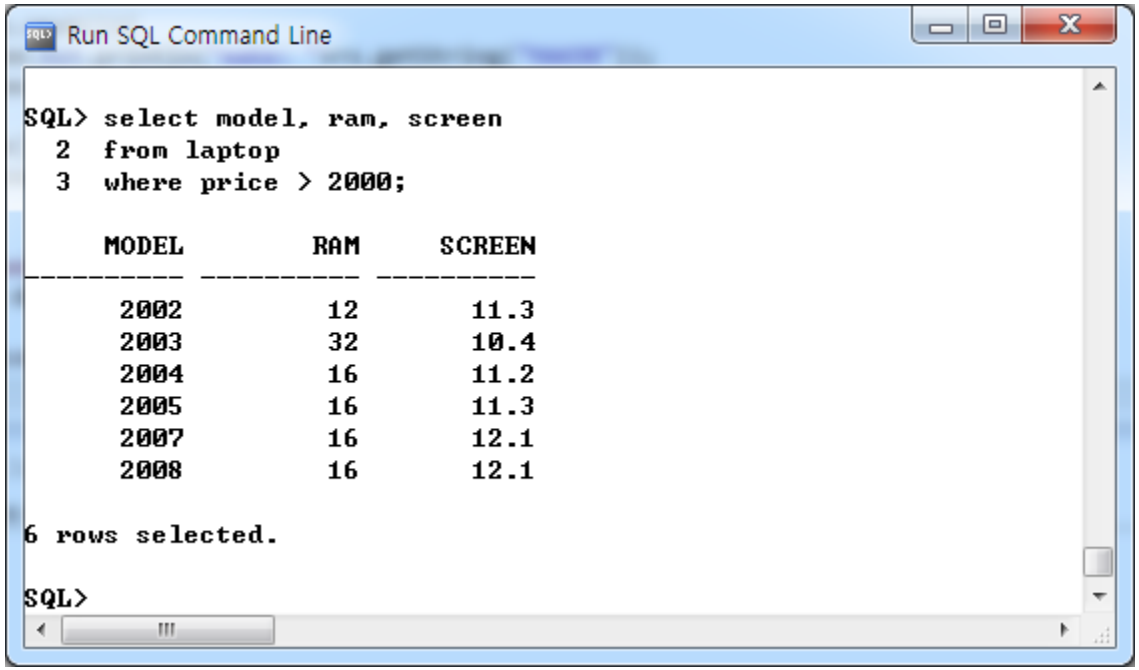
SQL 질의 언어 실습 (SELECT 문)



SQL 질의 언어 실습 (SELECT 문)

- 문제 3) 가격이 $2000을 초과하는 laptop들의 모델 번호, 메모리 용량, 화면 크기를 찾아라

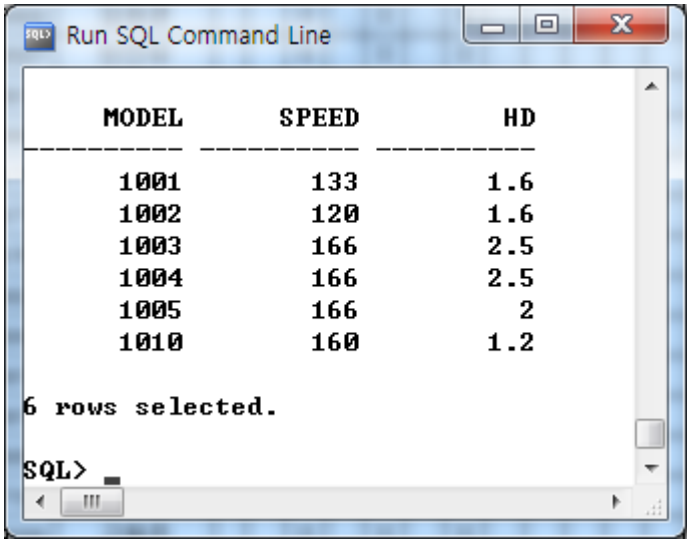
SQL 질의 언어 실습 (SELECT 문)



SQL 질의 언어 실습 (SELECT 문)

- 문제 **4) 6**배속이나 **8**배속의 **CD**를 가지고 있으며 가격이 **$2000**미만인 **PC**들의 모델 번호, 속도, 하드 디스크 용량을 구하라.
  (hint: where절에 **or**와 **and**사용**)**

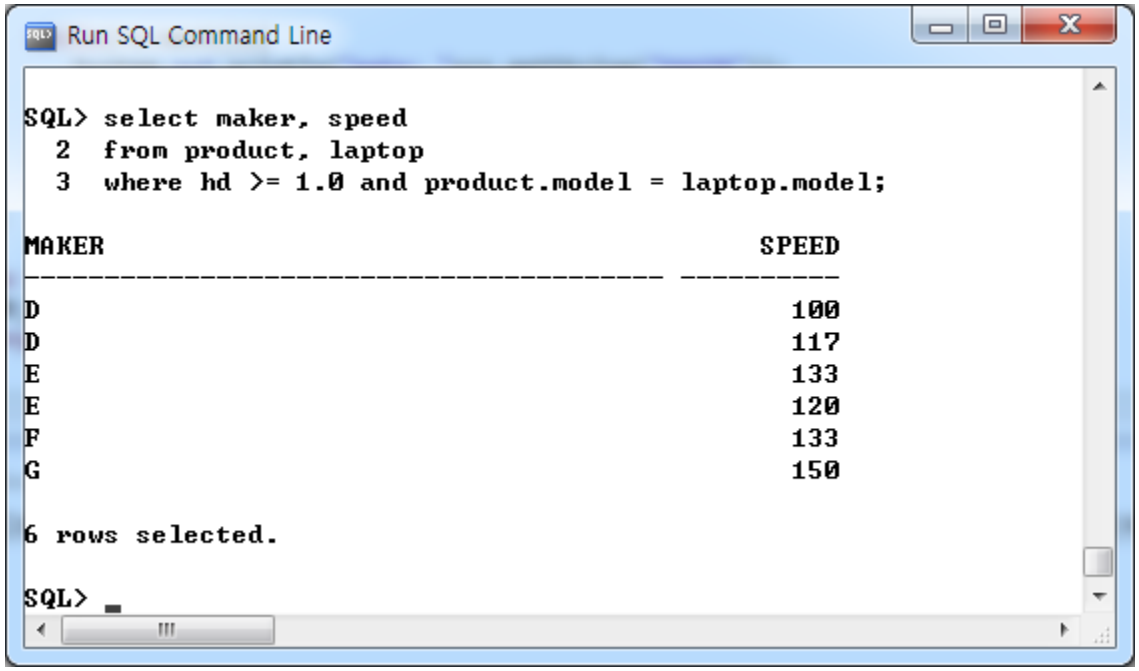
SQL 질의 언어 실습 (SELECT 문)



SQL 질의 언어 실습 (카티션 곱)

- 문제 5) 적어도 1기가 바이트의 용량을 지닌 하드 디스크가 내장된 랩탑의 속도와 그 제조업체를 구하라.
  (hint: 조인이 필요, where절에는 두개의 조건이 필요)

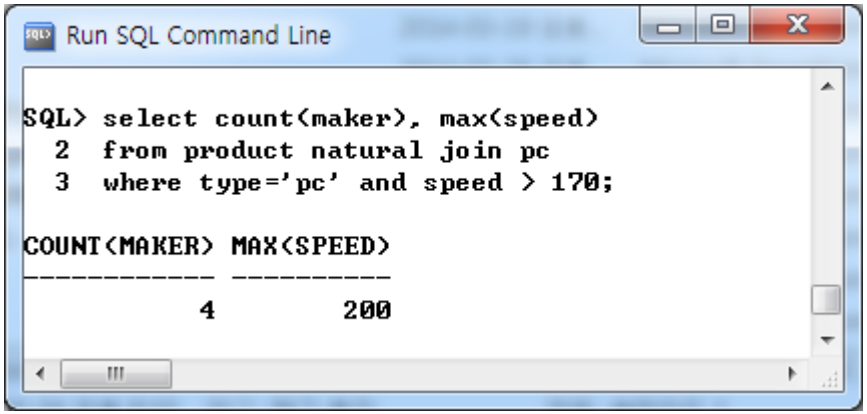
SQL 질의 언어 실습 (카티션 곱)



SQL 질의 언어 실습 (natural join)

- 문제 6) type이 pc이고 speed가 170보다 큰 maker의 개수와 최대 speed를 구하시오.
  (hint : natural join을 사용하고, count, max를 사용해라)

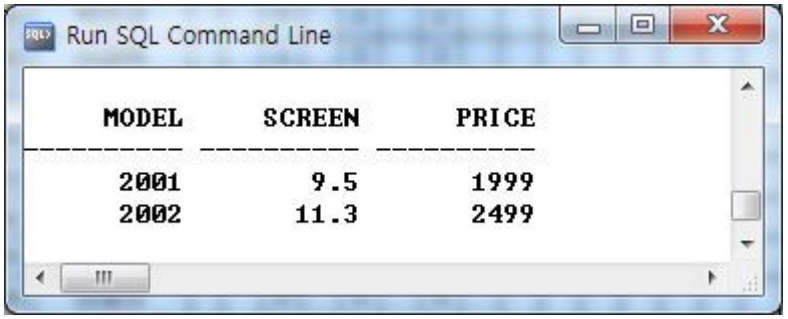
SQL 질의 언어 실습 (natural join)



SQL 질의 언어 실습 (natural join)

- 문제 7) maker가 D이고 모델번호가 2003보다 작은 Laptop의 model과 screen, price를 나타내라
(hint : natural join을 사용해라)

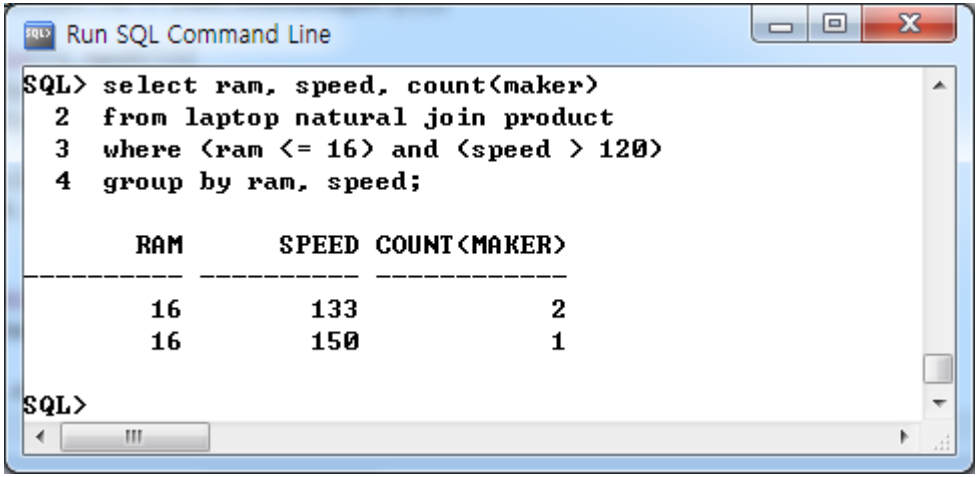
SQL 질의 언어 실습 (natural join)



SQL 질의 언어 실습 (group by)

- 문제 8) ram이 16보다 작거나 같고, speed가 120보다 큰 laptop의 ram, speed와 maker의 개수를 구하시오.
  (hint : natural join과 group by, count를 이용해라)

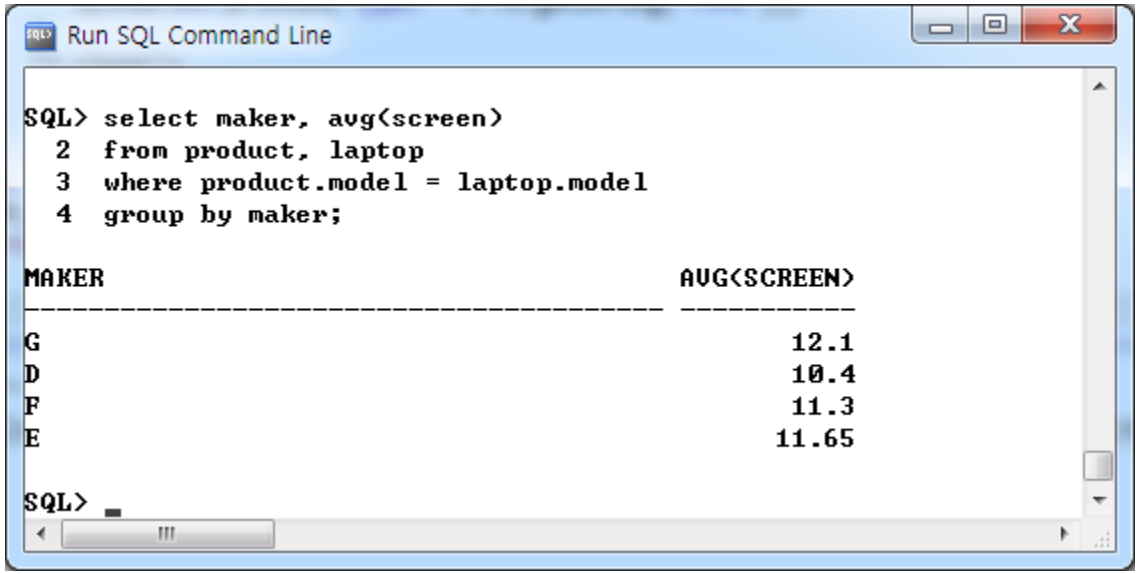
SQL 질의 언어 실습 (group by)



SQL 질의 언어 실습 (group by)

- 문제 9) 각 제조업체별로 랩탑의 평균 화면 크기를 구하라. (hint: avg이용, group by 이용)

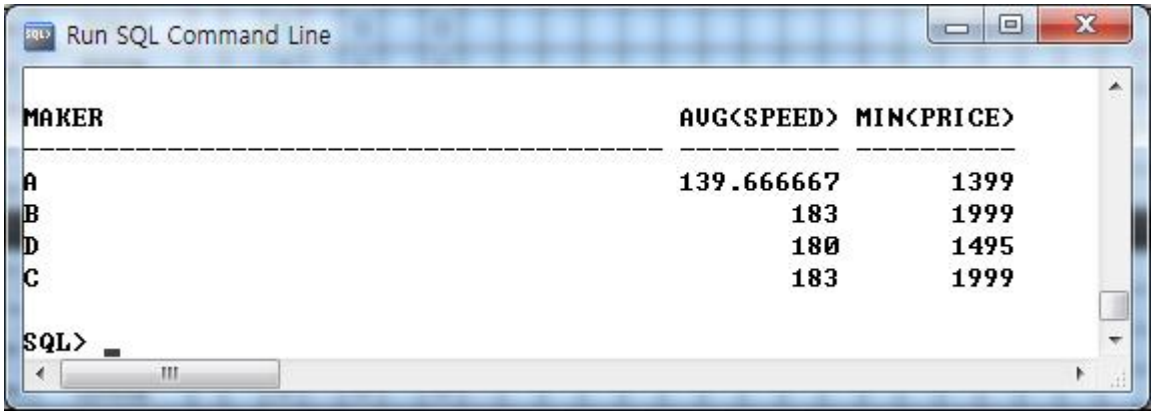
SQL 질의 언어 실습 (group by)



SQL 질의 언어 실습 (group by)

- 문제 **10) PC**의 메이커별 평균 속도와 최저가격을 나타내라

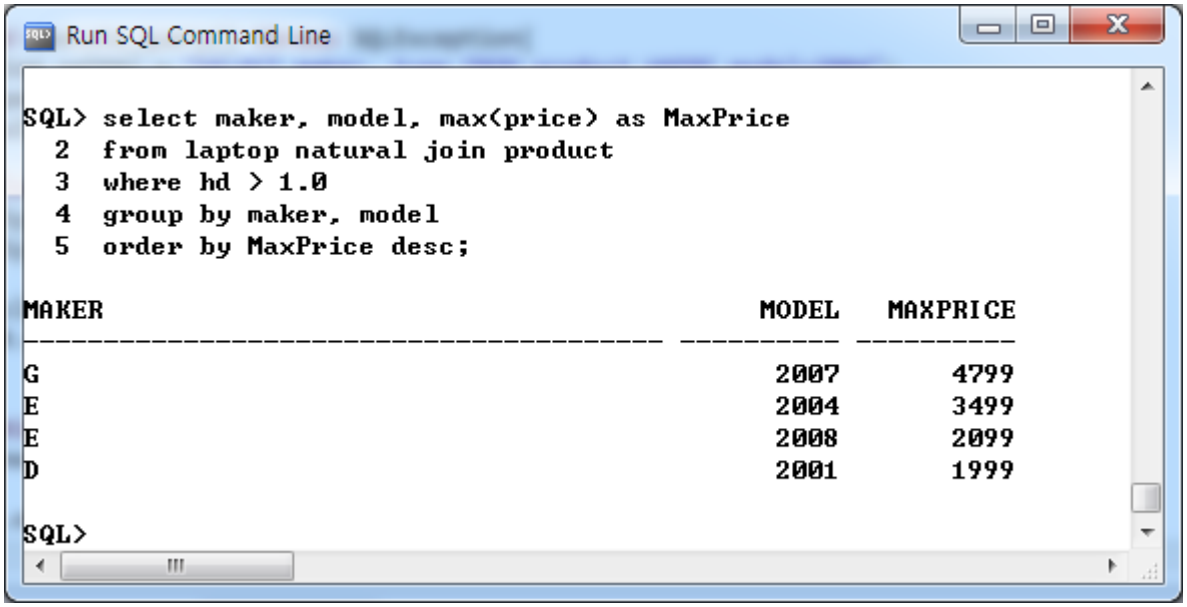
SQL 질의 언어 실습 (group by)



SQL 질의 언어 실습 (order by)

- 문제 11) laptop의 hd가 1.0 초과인 maker, model, 최대가격을 구하고 최대가격을 기준으로 내림차순 하시오.
  (hint : natural join을 사용하고, group by, order by를 사용해라)

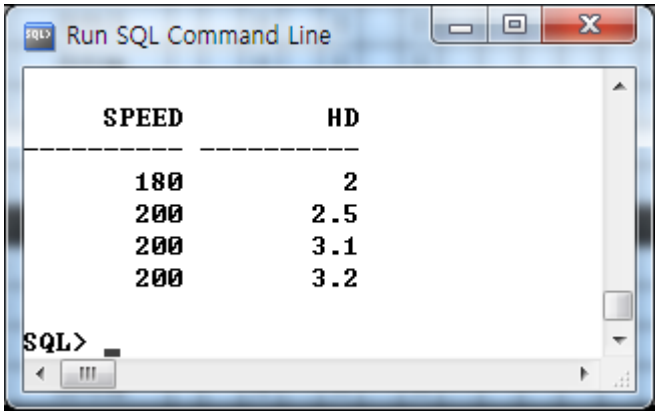
SQL 질의 언어 실습 (order by)



SQL 질의 언어 실습 (order by)

- 문제 **12) speed**가**180**이상이고 **price**가 **2000**보다 큰 **pc**의 **speed**와 **hd**를 **pc**의 **speed**로 정렬하여라

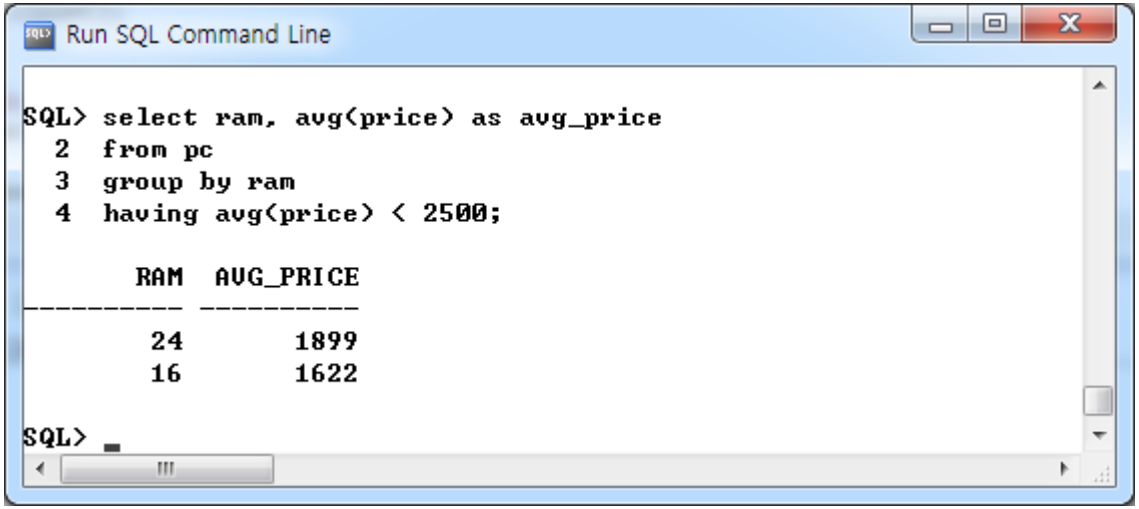
SQL 질의 언어 실습 (order by)



SQL 질의 언어 실습 (having)

- 문제 13) PC의 ram 크기 별 평균 가격이 2500보다 적은 ram과 그 가격을 나타내라

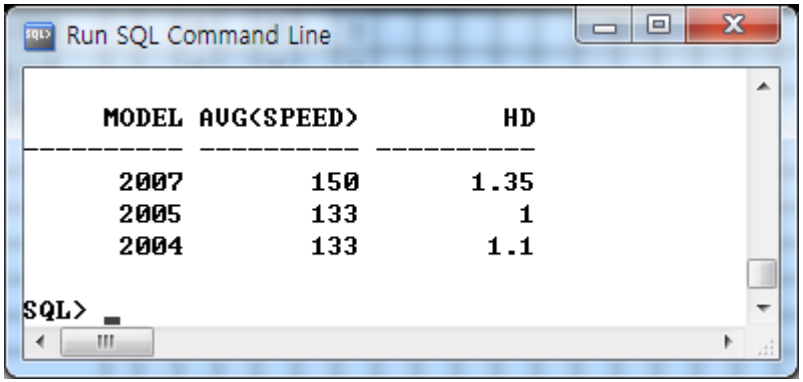
SQL 질의 언어 실습 (having)



SQL 질의 언어 실습 (having)

- 문제 **14) laptop**의 평균 **speed**가 **130**보다 큰 **model, hd,** 평균 **speed**를 구하시오.

SQL 질의 언어 실습 (having)



실습

- 문제 4, 7, 10, 12, 14번에 대한 답을 구하고 쿼리와 결과를 캡쳐를 하여 HY-IN에 업로드