

## Report #3 (SQL) 모범답안

- ❖ 본 답안은 유일한 답이 아니며, 다양한 형태로 다르게 표현될 수 있습니다.  
혹, 답안에 오류가 있더라도 그 책임은 교수가 지지 않습니다.  
오답을 발견할 경우 제일 먼저 수정 답안을 올린 학생에게는 보너스 점수 드립니다.  
(단, 철자가 틀렸거나, 세미콜론이 누락되거나 등의 사소한 오류 말고 논리적 오류에 한함)

가. 과목명에 '구조'가 들어 있는 과목번호와 과목명을 찾아라.

---

```
SQL> Run SQL Command Line

SQL> select course_id, title
  2  from course
  3  where title like '%구조%';
```

COURSE_ID	TITLE
C102	자료구조
C302	컴퓨터구조

```
SQL>
```

나. 2012년도 1학기에 강의가 없는 교수의 이름을 찾아라.

---

```
SQL> Run SQL Command Line

SQL> select name
  2  from professor
  3  where prof_id not in (select prof_id
  4                          from class
  5                          where year = 2012 and semester = 1);

NAME
-----
고희석
김태석
```

다. 2012년도 1학기에 한 과목도 수강하지 않은 학생의 학번과 이름, 학과명을 찾아라.

SQL> Run SQL Command Line

```
SQL> select stu_id, name, dept_name
2  from student, department
3  where student.dept_id = department.dept_id and
4  stu_id not in (select distinct stu_id
5                  from class, takes
6                  where class.class_id=takes.class_id
7                  and year = 2012 and semester = 1);
```

STU_ID	NAME	DEPT_NAME
1234567	김영웅	컴퓨터공학과
1292305	김우주	산업공학과
1292501	박철수	전자공학과
1292502	백태성	전자공학과

## 라. 학과별 학생 수를 찾아라. (학과명, 학생수)

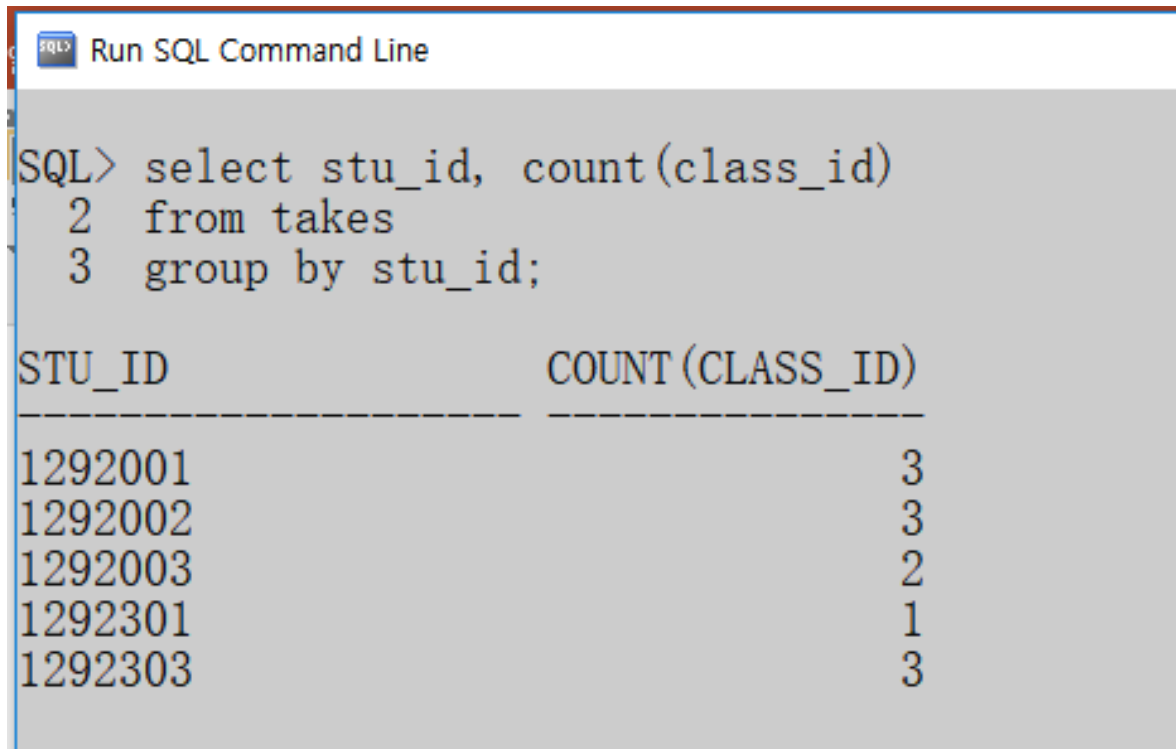
---

```
SQL> select dept_name, count(*)
  2  from student, department
  3  where student.dept_id = department.dept_id
  4  group by dept_name;
```

DEPT_NAME	COUNT (*)
컴퓨터공학과	4
산업공학과	3
전자공학과	2

마. 학번별 수강과목 수를 찾아라. (학번, 수강과목수)

---



The screenshot shows a window titled "Run SQL Command Line". Inside, a SQL query is entered and executed. The query is: `SQL> select stu_id, count(class_id)  
2 from takes  
3 group by stu_id;`. The results are displayed in a table with two columns: `STU_ID` and `COUNT (CLASS_ID)`. The results are as follows:

STU_ID	COUNT (CLASS_ID)
1292001	3
1292002	3
1292003	2
1292301	1
1292303	3

바. 가장 최근에 임용된 교수의 이름과 재직연수를 찾아라.

SQL Plus

```
SQL>
SQL> select name, 2023-year_emp
  2   from professor
  3   where year_emp >= all (select year_emp
  4                           from professor);
```

NAME	2023-YEAR_EMP
박철재	16

SQL Plus

```
SQL>
SQL> select name, 2023-year_emp
  2   from professor
  3   where year_emp = (select max(year_emp)
  4                       from professor);
```

NAME	2023-YEAR_EMP
박철재	16

사. 같은 학과, 같은 주소를 갖는 학생 이름의 쌍을 찾아라. (이때, 동일인이 쌍으로 나와서도 안되며, 한번 나온 쌍은 순서를 바꿔서 나와서도 안된다.

```
SQL> Run SQL Command Line

SQL> select s1.name, s2.name
      2  from student s1, student s2
      3  where s1.dept_id = s2.dept_id and
      4  s1.address = s2.address and
      5  s1.name < s2.name;
```

NAME	NAME
김영웅	김정현
김광식	김정현
김광식	김영웅



## 1. 모든 고객의 계좌번호, 이름, 그리고 예금 잔액을 검색하라.

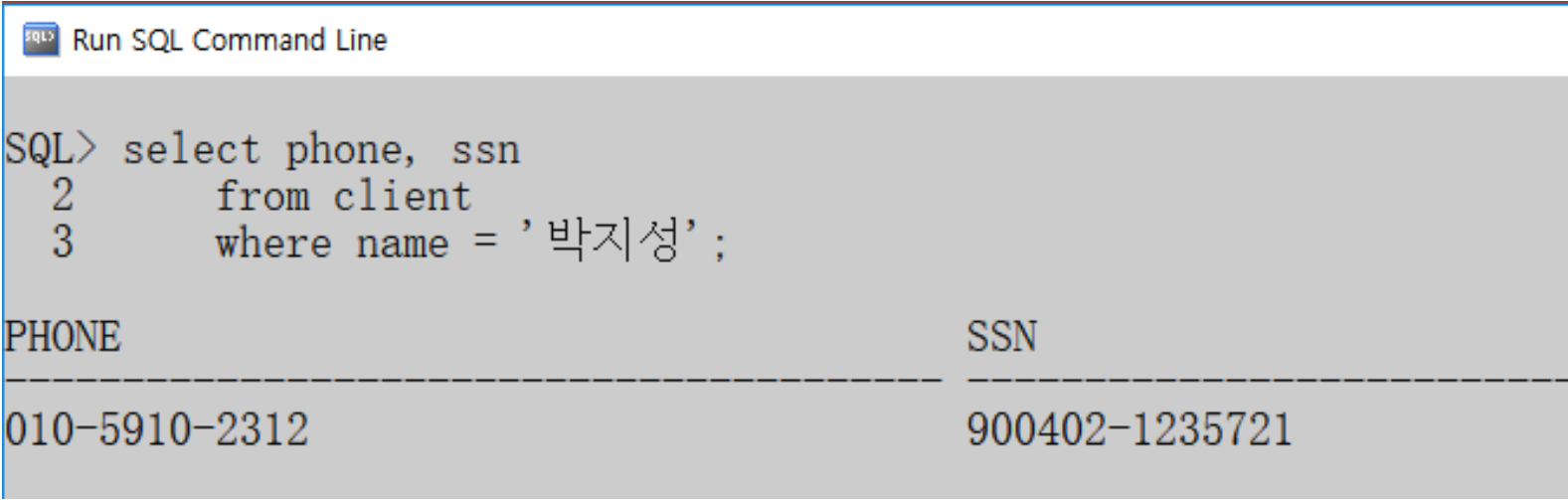
```
SQL> select deposit_num, name, balance
2      from client left outer join deposit
3      using(ssn);
```

DEPOSIT_NUM	NAME	BALANCE
101	강지선	120000
110	홍순태	1200000
103	홍순태	560000
108	홍순태	320000
102	이기상	2300000
104	박지성	870000
	이영순	
107	황현희	1900000
109	박선희	560000
105	차두리	9000
100	김기식	330000
106	김기식	110000
111	정성태	900000

13 rows selected.

## 2. 이름이 '박지성'인 고객의 전화번호와 주민등록번호를 검색하라.

---



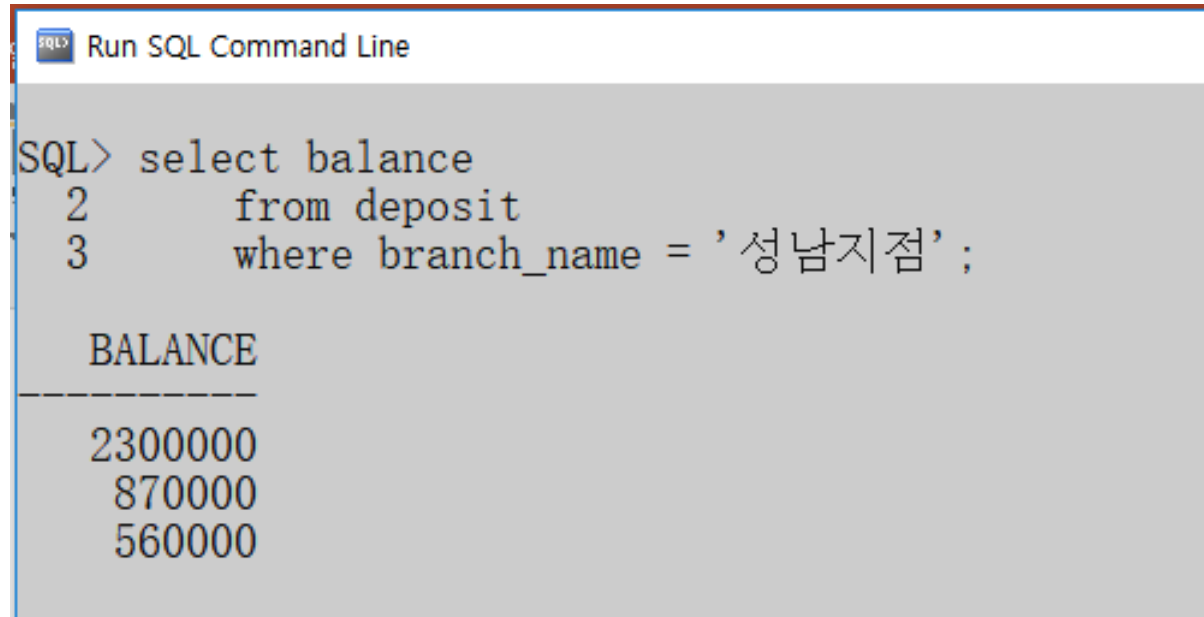
The screenshot shows a window titled "Run SQL Command Line". Inside, a SQL query is entered and executed. The query selects the phone and SSN from a table named 'client' where the name is '박지성'. The results are displayed in a table with two columns: PHONE and SSN. The phone number is 010-5910-2312 and the SSN is 900402-1235721.

```
SQL> select phone, ssn
      2      from client
      3      where name = '박지성';
```

PHONE	SSN
010-5910-2312	900402-1235721

3. 지점 이름이 '성남지점'인 지점을 통해 개설된 모든 예금의 잔액을 검색하라.

---



The screenshot shows a window titled "Run SQL Command Line" with a SQL prompt. The query entered is: `SQL> select balance  
2 from deposit  
3 where branch_name = '성남지점';`. The results are displayed in a table with a single column labeled "BALANCE" and three rows of values: 2300000, 870000, and 560000.

```
SQL> select balance  
2      from deposit  
3      where branch_name = '성남지점';  
  
BALANCE  
-----  
2300000  
870000  
560000
```

#### 4. 지점장 이름이 '고희경'인 지점의 이름과 주소를 검색하라.

---

```
SQL> select branch_name, address
2      from branch
3      where branch_head = '고희경';
```

BRANCH_NAME	ADDRESS
제주지점	제주

## 5. 지점 이름이 '광주지점'인 지점의 지점장 이름과 주소를 검색하라.

---

```
SQL> Run SQL Command Line

SQL> select branch_head, address
2      from branch
3      where branch_name = '광주지점';
```

BRANCH_HEAD	ADDRESS
김기백	광주

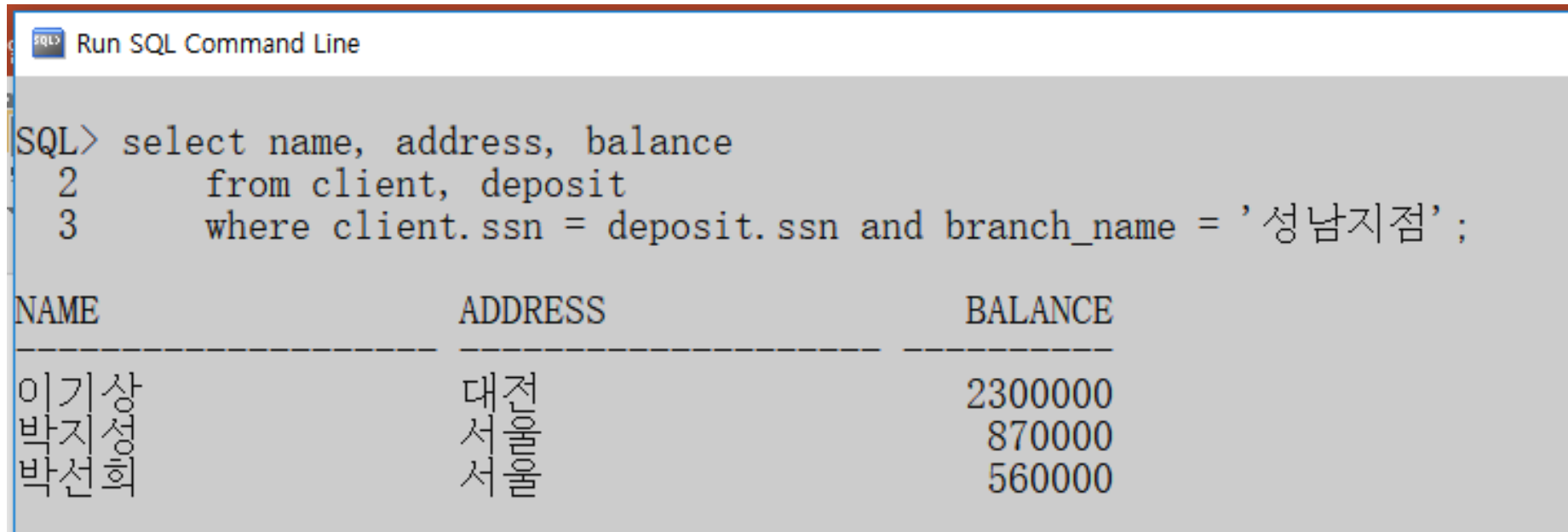
6. 이름이 '김기식'인 고객이 소유한 예금의 계좌번호, 개설지점의 이름, 잔액을 검색하라.

SQL> Run SQL Command Line

```
SQL> select deposit_num, branch_name, balance  
2  from client, deposit  
3  where client.ssn = deposit.ssn and name = '김기식';
```

DEPOSIT_NUM	BRANCH_NAME	BALANCE
100	서울지점	330000
106	대구지점	110000

7. '성남지점'에서 계좌를 개설한 고객의 이름과 주소, 그리고 예금 잔액을 검색하라.



The screenshot shows a SQL Command Line window with the following content:

```
SQL> select name, address, balance
2      from client, deposit
3      where client.ssn = deposit.ssn and branch_name = '성남지점';
```

NAME	ADDRESS	BALANCE
이기상	대전	2300000
박지성	서울	870000
박선희	서울	560000

8. '성남지점'에서 계좌를 개설한 고객 중 김씨 성을 가진 고객의 이름과 예금 잔액을 검색하라.

---

SQL> Run SQL Command Line

```
SQL> select name, balance
2      from client, deposit
3      where client.ssn = deposit.ssn and branch_name = '성남지점'
4      and name like '김%';

no rows selected
```



## 9. 예금 잔액이 10만원 이상인 계좌를 소유한 고객의 이름을 검색하라.

```
Run SQL Command Line

SQL> select name
  2     from client, deposit
  3     where client.ssn = deposit.ssn and balance >= 100000;

NAME
-----
강지선
홍순태
홍순태
홍순태
홍순태
이기상
박지성
황현희
박선희
김기식
김기식
정성태

11 rows selected.
```

## 10. 예금 잔액이 10만원 이상인 계좌가 개설된 지점의 이름과 지점장 이름을 검색하라.

SQL> Run SQL Command Line

```
SQL> select distinct branch.branch_name, branch_head  
2      from deposit, branch  
3      where deposit.branch_name = branch.branch_name and balance >= 100000;
```

BRANCH_NAME	BRANCH_HEAD
-------------	-------------

광주시점	김기백
대전지점	이연희
성남지점	박찬주
서울지점	강동희
대구지점	김기식
제주지점	고희경

6 rows selected.

## 11. 예금을 개설한 지점의 지점장과 이름이 같은 고객이 소유한 예금의 계좌번호, 잔액, 그리고 개설지점 이름을 검색하라.

Run SQL Command Line

```
SQL> select deposit_num, balance, branch.branch_name
2      from client, deposit, branch
3      where client.ssn = deposit.ssn and deposit.branch_name = branch.branch_name
4            and name = branch.branch_head;
```

DEPOSIT_NUM	BALANCE	BRANCH_NAME
106	110000	대구지점

## 12. '서울지점'에서 계좌를 개설한 고객들 중에서 남자 고객의 이름과 예금 잔액을 검색하라.

```
SQL> Run SQL Command Line

SQL> select name, balance
2      from client, deposit
3      where client.ssn = deposit.ssn and branch_name = '서울지점'
4      and client.ssn like '%-1%';
```

NAME	BALANCE
김기식	330000
홍순태	1200000

### 13. 주민등록번호상의 생일이 3월인 모든 고객의 이름과 소유한 예금의 계좌번호를 검색하라.

//예금이 있는 고객만 조회

```
SQL> select name, deposit_num
2      from client, deposit
3      where client.ssn=deposit.ssn and
4            client.ssn like '__03%';
```

NAME	DEPOSIT_NUM
홍순태	103
홍순태	108
홍순태	110
정성태	111

//예금이 없는 고객도 조회

```
SQL> select name, deposit_num
2      from client left outer join deposit
3      using (ssn)
4      where ssn like '__03%';
```

NAME	DEPOSIT_NUM
홍순태	103
홍순태	108
홍순태	110
이영순	
정성태	111

## 14. 자신의 주소와 같은 지점에 계좌를 소유하고 있는 고객의 이름과 예금 잔액을 검색하라.

```
SQL> Run SQL Command Line

SQL> select name, balance
2      from client, deposit, branch
3      where client.ssn = deposit.ssn and deposit.branch_name = branch.branch_name
4      and client.address = branch.address;

NAME                BALANCE
-----
김기식                330000
홍순태                1200000
```

## 15. ‘성남지점’과 거래하고 있는 고객의 숫자를 검색하라.

---

```
SQL> Run SQL Command Line

SQL> select count(distinct ssn)
      2      from deposit
      3      where branch_name = '성남지점';

COUNT(DISTINCTSSN)
-----
                    3
```

## 16. 각 지점별 잔액의 총합을 검색하라.

```
SQL> select branch_name, sum(balance)
2      from deposit
3      group by branch_name;
```

BRANCH_NAME	SUM(BALANCE)
대구지점	119000
대전지점	120000
제주지점	900000
광주지점	880000
서울지점	3430000
성남지점	3730000

6 rows selected.



## 17. 고객 이름별 예금 잔액의 총합을 검색하라.

```
Run SQL Command Line

SQL> select name, sum(balance)
      2      from client, deposit
      3      where client.ssn = deposit.ssn
      4      group by name;

NAME                                SUM(BALANCE)
-----
박선희                             560000
정성태                             900000
이기상                             2300000
박지성                             870000
김기식                             440000
강지선                             120000
홍순태                             2080000
황현희                             1900000
차두리                             9000

9 rows selected.
```

## 18. 잔액의 합이 100만원 이상인 지점 이름과 잔액의 합을 검색하라.

```
SQL> select branch_name, sum(balance)
2      from deposit
3      group by branch_name
4      having sum(balance) >= 1000000;
```

BRANCH_NAME	SUM(BALANCE)
서울지점	3430000
성남지점	3730000

## 19. 지점별로 예금 잔액이 100만원 이상인 고객의 숫자를 검색하라.

Run SQL Command Line

```
SQL> select branch_name, count(distinct ssn)
2      from deposit
3      where balance >= 1000000
4      group by branch_name;
```

BRANCH_NAME	COUNT(DISTINCTSSN)
서울지점	2
성남지점	1

// 이 문제는 논란의 소지가 있습니다. 한 명의 고객이 동일지점에 두 개의 계좌를 갖고 있는데 잔고가 각각 500000, 600000일 때 이 고객을 카운터해야 하는지에 대해 문제가 분명치 않습니다.

## 20. 예금 계좌를 소유하고 있지 않은 고객의 이름과 전화번호를 검색하라.

SQL> Run SQL Command Line

```
SQL> select name, phone
2      from client
3      where ssn not in (select distinct ssn from deposit);
```

NAME	PHONE
이영순	010-9876-2323

SQL Plus

```
SQL>
SQL> select name, phone
2  from client left outer join deposit
3  using (ssn)
4  where deposit_num is null;
```

NAME	PHONE
이영순	010-9876-2323