

[sql] programmers sql high score2

holy

<2023-08-03 Thu>

Contents

problem1: 강원도에 위치한 생산공장 목록 출력하기(level1)

강원도에 위치한 생산공장 목록 출력하기

Description

다음은 식품공장의 정보를 담은 `FOOD_FACTORY` 테이블입니다. `FOOD_FACTORY` 테이블은 다음과 같으며 `FACTORY_ID`, `FACTORY_NAME`, `ADDRESS`, `TLNO` 는 각각 공장 ID, 공장 이름, 주소, 전화번호를 의미합니다.

Column name	Type	Nullable
<code>FACTORY_ID</code>	<code>VARCHAR(10)</code>	FALSE
<code>FACTORY_NAME</code>	<code>VARCHAR(50)</code>	FALSE
<code>ADDRESS</code>	<code>VARCHAR(100)</code>	FALSE
<code>TLNO</code>	<code>VARCHAR(20)</code>	TRUE

문제

`FOOD_FACTORY` 테이블에서 강원도에 위치한 식품공장의 공장 ID, 공장 이름, 주소를 조회하는 SQL문을 작성해주세요. 이때 결과는 공장 ID를 기준으로 오름차순 정렬해주세요.

FOOD_FACTORY 테이블이 다음과 같을 때

FACTORY_ID	FACTORY_NAME	ADDRESS	TLNO
FT19980003	(주)맛있는라면	강원도 정선군 남면 칠현로 679	033-431-3122
FT19980004	(주)맛있는기름	경기도 평택시 포승읍 포승공단순환로 245	031-651-2410
FT20010001	(주)맛있는소스	경상북도 구미시 1공단로7길 58-11	054-231-2121
FT20010002	(주)맛있는통조림	전라남도 영암군 미암면 곤미현로 1336	061-341-5210
FT20100001	(주)맛있는차	전라남도 장성군 서삼면 장산리 233-1 번지	061-661-1420
FT20100002	(주)맛있는김치	충청남도 아산시 탕정면 탕정면로 485	041-241-5421
FT20100003	(주)맛있는음료	강원도 원주시 문막읍 문막공단길 154	033-232-7630
FT20100004	(주)맛있는국	강원도 평창군 봉평면 진조길 227-35	033-323-6640
FT20110001	(주)맛있는밥	경기도 화성시 팔탄면 가재리 34번지	031-661-1532
FT20110002	(주)맛있는과자	광주광역시 북구 하서로 222	062-211-7759

SQL을 실행하면 다음과 같이 출력되어야 합니다.

FACTORY_ID	FACTORY_NAME	ADDRESS
FT19980003	(주)맛있는라면	강원도 정선군 남면 칠현로 679
FT20100003	(주)맛있는음료	강원도 원주시 문막읍 문막공단길 154
FT20100004	(주)맛있는국	강원도 평창군 봉평면 진조길 227-35

풀이

푸는 순서는 다음과 같다. from부터 푼다. 이문제에서 어려울것은 없다.

from FOOD_FACTORY

select FACTORY_ID, FACTORY_NAME, ADDRESS from
FOOD_FACTORY

select FACTORY_ID, FACTORY_NAME, ADDRESS from
FOOD_FACTORY order by FACTORY_ID asc

problem2: 흉부외과 또는 일반 외과 의사 목록 출력하기
(level1)

증상별 의사 목록 출력하기					
증상별 의사 목록 출력하기					
증상별 의사 목록 출력하기					
DR_NAME DR_ID LCNS_NO HRE_YMD MCSP_CD TNO					
우리 0202000029 LC00010001 2009-03-01 CS 07054802011					
제작 0202000001 LC00010002 2009-07-01 CS 07053220222					
조광호 0202010123 LC00099200 2017-03-01 GS 07054969210					
자기 0202000001 LC00010003 2010-03-01 NP 07034229818					
황나 0202000231 LC00010004 2015-11-01 GS 07040840278					
진판 0202000012 LC00010002 2010-03-01 FM 07034042290					
나경 0202000013 LC00010005 2010-03-01 CS 07094848291					
오윤관 0202000031 LC00010032 2010-11-01 GS 07098428997					
최도민 0202000032 LC00010032 2010-03-01 GS 0702388922					
정 0202000039 LC00010052 2010-07-01 GS 0705930758					

의사 목록 출력하기					
의사 목록 출력하기					
의사 목록 출력하기					
DR_NAME DR_ID MCSP_CD HRE_YMD					
나경 0202000013 CS 2010-03-01					
정 0202000039 GS 2010-07-01					
조광호 0202010123 GS 2017-03-01					
진판 0202000012 GS 2010-03-01					
최도민 0202000032 GS 2010-03-01					
우리 0202000029 GS 2009-03-01					
제작 0202000001 GS 2009-07-01					
자기 0202000001 GS 2010-03-01					
황나 0202000231 GS 2015-11-01					
오윤관 0202000031 GS 2010-11-01					
최스민 0202000032 GS 2010-03-01					
제작 0202000001 GS 2009-07-01					
우리 0202000029 GS 2009-03-01					

증상별 의사 목록 출력하기					
증상별 의사 목록 출력하기					
증상별 의사 목록 출력하기					
DR_NAME DR_ID MCSP_CD HRE_YMD					
나경 0202000013 CS 2010-03-01					
정 0202000039 GS 2010-07-01					
조광호 0202010123 GS 2017-03-01					
진판 0202000012 GS 2010-03-01					
최도민 0202000032 GS 2010-03-01					
우리 0202000029 GS 2009-03-01					
제작 0202000001 GS 2009-07-01					
자기 0202000001 GS 2010-03-01					
황나 0202000231 GS 2015-11-01					
오윤관 0202000031 GS 2010-11-01					
최스민 0202000032 GS 2010-03-01					
제작 0202000001 GS 2009-07-01					
우리 0202000029 GS 2009-03-01					

풀이

problem3: 서울에 위치한 식당 목록 출력하기(level4)

서울에 위치한 식당 목록 출력하기		
Description		
서울에 위치한 식당 목록을 출력하세요. REST_ID는 식당 ID, REST_NAME은 식당 이름입니다. EX) REST_ID REST_NAME ADDRESS TEL 음식점 ID, 식당 이름, 주소, 휴대폰 번호를 출력합니다.		
Favorites , PARKING_LOT , ADDRESS , TEL , 음식점 ID, 식당 이름, 주소, 휴대폰 번호, 팔고자 하는 주택 정보입니다.		
Column name	Type	
REST_ID	VARCHAR(10)	FALSE
REST_NAME	VARCHAR(100)	FALSE
FOOD_TYPE	VARCHAR(100)	TRUE
VIEW_CNT	NUMBER	TRUE
FAVORITES	NUMBER	TRUE
PARKING_LOT	VARCHAR(10)	TRUE
ADDRESS	VARCHAR(100)	TRUE
TEL	VARCHAR(100)	TRUE

Review		
Description		
REST_ID는 식당 ID, MEMBER_ID는 회원 ID, REVIEW_SCORE는 평균 점수, REVIEW_TEXT는 리뷰 내용입니다.		
REVIEW_ID , REVIEW_DATE 는 각각 리뷰 ID, 사용 ID, 등록 일자입니다.		
Column name	Type	
REVIEW_ID	VARCHAR(10)	FALSE
REST_ID	VARCHAR(10)	TRUE
MEMBER_ID	VARCHAR(100)	TRUE
REVIEW_SCORE	NUMBER	TRUE
REVIEW_TEXT	VARCHAR(1000)	TRUE
REVIEW_DATE	DATE	TRUE

음식		
Description		
REST_ID는 식당 ID, MEMBER_ID는 회원 ID, REVIEW_SCORE는 평균 점수, REVIEW_TEXT는 리뷰 내용입니다.		
REVIEW_ID , REVIEW_DATE 는 각각 리뷰 ID, 사용 ID, 등록 일자입니다.		
Column name	Type	
REVIEW_ID	VARCHAR(10)	FALSE
REST_ID	VARCHAR(10)	TRUE
MEMBER_ID	VARCHAR(100)	TRUE
REVIEW_SCORE	NUMBER	TRUE
REVIEW_TEXT	VARCHAR(1000)	TRUE
REVIEW_DATE	DATE	TRUE

www_review		
Description		
www_review_id, MEMBER_ID, REVIEW_SCORE, REVIEW_DATE		
www_review_id는 리뷰 ID, MEMBER_ID는 회원 ID, REVIEW_SCORE는 평균 점수, REVIEW_DATE는 등록 일자입니다.		
Column name	Type	
www_review_id	VARCHAR(10)	FALSE
MEMBER_ID	VARCHAR(100)	TRUE
REVIEW_SCORE	NUMBER	TRUE
REVIEW_DATE	DATE	TRUE

www_member		
Description		
www_member_id, MEMBER_ID, MEMBER_NAME, MEMBER_PASSWORD, MEMBER_EMAIL, MEMBER_PHONE, MEMBER_ADDRESS, MEMBER_GENDER, MEMBER_BIRTHDAY, MEMBER_IS_DELETED		
www_member_id는 회원 ID, MEMBER_ID는 회원 ID, MEMBER_NAME은 회원 이름, MEMBER_PASSWORD는 회원 비밀번호, MEMBER_EMAIL은 회원 이메일, MEMBER_PHONE은 회원 전화번호, MEMBER_ADDRESS는 회원 주소, MEMBER_GENDER는 회원 성별, MEMBER_BIRTHDAY는 회원 생년월일, MEMBER_IS_DELETED는 회원 탈퇴 여부입니다.		
Column name	Type	
www_member_id	VARCHAR(10)	FALSE
MEMBER_ID	VARCHAR(100)	TRUE
MEMBER_NAME	VARCHAR(100)	TRUE
MEMBER_PASSWORD	VARCHAR(100)	TRUE
MEMBER_EMAIL	VARCHAR(100)	TRUE
MEMBER_PHONE	VARCHAR(100)	TRUE
MEMBER_ADDRESS	VARCHAR(100)	TRUE
MEMBER_GENDER	VARCHAR(10)	TRUE
MEMBER_BIRTHDAY	DATE	TRUE
MEMBER_IS_DELETED	NUMBER	TRUE

www_member_log		
Description		
www_member_log_id, MEMBER_ID, MEMBER_IP, MEMBER_LOGGED_IN_AT, MEMBER_LOGGED_OUT_AT		
www_member_log_id는 로그인 기록 ID, MEMBER_ID는 회원 ID, MEMBER_IP는 로그인 IP주소, MEMBER_LOGGED_IN_AT는 로그인 일자, MEMBER_LOGGED_OUT_AT는 로그아웃 일자입니다.		
Column name	Type	
www_member_log_id	VARCHAR(10)	FALSE
MEMBER_ID	VARCHAR(100)	TRUE
MEMBER_IP	VARCHAR(100)	TRUE
MEMBER_LOGGED_IN_AT	DATE	TRUE
MEMBER_LOGGED_OUT_AT	DATE	TRUE

www_member_log_out		
Description		
www_member_log_out_id, MEMBER_ID, MEMBER_IP, MEMBER_LOGGED_OUT_AT		
www_member_log_out_id는 로그아웃 기록 ID, MEMBER_ID는 회원 ID, MEMBER_IP는 로그아웃 IP주소, MEMBER_LOGGED_OUT_AT는 로그아웃 일자입니다.		
Column name	Type	
www_member_log_out_id	VARCHAR(10)	FALSE
MEMBER_ID	VARCHAR(100)	TRUE
MEMBER_IP	VARCHAR(100)	TRUE
MEMBER_LOGGED_OUT_AT	DATE	TRUE

풀이

problem4: 조건에 맞는 도서 리스트 출력하기(level1)

도서에 대한 표준 목록 출력하기		
Description		
도서에 대한 표준 목록을 출력하세요. CATEGORY는 카테고리, AUTHOR_ID는 저작자 ID, PRICE는 가격입니다.		
BOOK_ID , BOOK_NAME , AUTHOR_ID , PRICE 는 각각 도서 ID, 도서 이름, 저작자 ID, 가격입니다.		
Column name	Type	
BOOK_ID	VARCHAR(10)	FALSE
CATEGORY	VARCHAR(100)	FALSE
AUTHOR_ID	INTEGER	FALSE
PRICE	INTEGER	FALSE
PUBLISHED_DATE	DATE	FALSE

책 목록			
BOOK_ID	CATEGORY	AUTHOR_ID	PRICE
1	경제	1	50000
2	경제	2	90000
3	경제	2	110000
4	경제	3	80000
5	경제	1	100000

고객에게 판매된 책 목록입니다. 판매일은 2021-01-10입니다.

BOOK_ID	PUBLISHED_DATE
1	2021-01-10
2	2021-01-10
3	2021-01-10
4	2021-01-10
5	2021-01-10

판매된 책은 모두 판매되었거나 예상 판매되는 상태입니다.

BOOK_ID	PUBLISHED_DATE
1	2021-01-15
2	2021-01-15
3	2021-01-15

판매된 책은 모두 예상 판매되는 상태입니다.

풀이

problem5: 파일로 만든 아이스크림 고르기(level1)

파일로 만든 아이스크림		
파일은 고객이 아이스크림을 구매할 때 고객ID와 아이스크림ID, 판매금액을 기록하는 형식으로 되어 있습니다. 고객ID는 1001에서 1010까지이며, 아이스크림ID는 1001에서 1010까지입니다. 판매금액은 1000에서 100000까지입니다. 고객ID와 아이스크림ID는 같은 범위입니다. 판매금액은 고객ID와 아이스크림ID에 따라 다릅니다.		
고객ID	아이스크림ID	판매금액
1001	1001	1000
1002	1002	2000
1003	1003	3000
1004	1004	4000
1005	1005	5000
1006	1006	6000
1007	1007	7000
1008	1008	8000
1009	1009	9000
1010	1010	100000

파일에는 고객ID와 아이스크림ID, 판매금액이 기록되어 있습니다. 판매금액은 고객ID와 아이스크림ID에 따라 다르게 설정되었습니다.

고객ID	아이스크림ID	판매금액
1001	1001	1000
1002	1002	2000
1003	1003	3000
1004	1004	4000
1005	1005	5000
1006	1006	6000
1007	1007	7000
1008	1008	8000
1009	1009	9000
1010	1010	100000

파일에는 고객ID와 아이스크림ID, 판매금액이 기록되어 있습니다. 판매금액은 고객ID와 아이스크림ID에 따라 다르게 설정되었습니다.

풀이

problem6: 평균 일일 대여 요금 구하기(level1)

평균 일일 대여 요금		
파일은 고객ID와 대여일자, 반납일자를 기록하는 형식으로 되어 있습니다. 고객ID는 1001에서 1010까지이며, 대여일자는 2020-01-01에서 2020-12-31까지입니다. 대여일자는 같은 범위입니다. 반납일자는 같은 범위입니다. 고객ID와 대여일자는 같은 범위입니다. 고객ID와 반납일자는 같은 범위입니다.		
고객ID	대여일자	반납일자
1001	2020-01-01	2020-01-01
1002	2020-01-01	2020-01-02
1003	2020-01-01	2020-01-03
1004	2020-01-01	2020-01-04
1005	2020-01-01	2020-01-05
1006	2020-01-01	2020-01-06
1007	2020-01-01	2020-01-07
1008	2020-01-01	2020-01-08
1009	2020-01-01	2020-01-09
1010	2020-01-01	2020-01-10

파일에는 고객ID와 대여일자, 반납일자가 기록되어 있습니다. 대여일자는 같은 범위입니다. 반납일자는 같은 범위입니다.

고객ID	대여일자	반납일자
1001	2020-01-01	2020-01-01
1002	2020-01-01	2020-01-02
1003	2020-01-01	2020-01-03
1004	2020-01-01	2020-01-04
1005	2020-01-01	2020-01-05
1006	2020-01-01	2020-01-06
1007	2020-01-01	2020-01-07
1008	2020-01-01	2020-01-08
1009	2020-01-01	2020-01-09
1010	2020-01-01	2020-01-10

파일에는 고객ID와 대여일자, 반납일자가 기록되어 있습니다. 대여일자는 같은 범위입니다. 반납일자는 같은 범위입니다.

풀이

problem7: 조건에 부합하는 중고거래 댓글 조회하기(level1)

```
/* 문제7: 조건에 부합하는 중고거래 댓글 조회하기(level1) */

-- 1. 중고거래 댓글 목록 조회하는 쿼리 작성
-- 조건: 판매자 이름이 '한국인'인 경우
-- 출력: 판매자 이름, 댓글 내용, 댓글 등록일자, 댓글 등록시간
```

```
SELECT * FROM USED_GOODS_COMMENT;
```

```
SELECT * FROM USED_GOODS_COMMENT WHERE SELLER_NAME = '한국인';
```

풀이

problem8: 인기 있는 아이스크림(level1)

```
/* 문제8: 인기 있는 아이스크림 조회하기(level1)

-- 1. 판매량이 많은 아이스크림 목록 조회하는 쿼리 작성
-- 조건: 판매량이 2000 이상인 경우
-- 출력: 아이스크림 이름, 판매량, 판매량 등록일자, 판매량 등록시간
```

```
SELECT * FROM ICECREAM_ORDER;
```

```
SELECT * FROM ICECREAM_ORDER WHERE QUANTITY >= 2000;
```

풀이

problem9: 3월에 태어난 여성 회원 목록 출력하기(level2)

```
/*
 * 문제 제작자: 정재우 교수님
 * 문제 번호: 9
 * 문제 이름: 3월에 태어난 여성 회원 목록 출력하기
 * 문제 내용: 회원 목록을 출력하는 프로그램을 작성하세요. 회원 목록은 회원번호, 이름, 성별, 생년월일, 주민등록번호로 구성됩니다.
 * 문제 조건: 회원 목록은 회원번호, 이름, 성별, 생년월일, 주민등록번호로 구성됩니다.
 * 문제 예상점수: 10점
 */
#include <stdio.h>
#include <string.h>

struct Member {
    int member_id;
    char name[10];
    char gender;
    int birth;
    char ssn[10];
};

int main() {
    struct Member members[10];
    int i, count = 0;

    // 회원 정보 입력
    for (i = 0; i < 10; i++) {
        printf("회원 %d 번째 이름과 주민등록번호를 입력하세요: ", i + 1);
        scanf("%s", members[i].name);
        printf("회원 %d 번째 성별을 입력하세요: ", i + 1);
        scanf(" %c", &members[i].gender);
        printf("회원 %d 번째 생년월일을 입력하세요: ", i + 1);
        scanf("%d", &members[i].birth);
        printf("회원 %d 번째 주민등록번호를 입력하세요: ", i + 1);
        scanf("%s", members[i].ssn);
    }

    // 3월생 여성을 찾는다.
    for (i = 0; i < 10; i++) {
        if ((members[i].birth / 100) == 3 && (members[i].gender == 'F')) {
            count++;
            printf("3월에 태어난 여성 회원 목록입니다.\n");
            printf("회원 번호: %d, 이름: %s, 성별: %c, 생년월일: %d, 주민등록번호: %s\n", i + 1, members[i].name, members[i].gender, members[i].birth, members[i].ssn);
        }
    }

    if (count == 0) {
        printf("3월에 태어난 여성 회원이 없습니다.\n");
    }

    return 0;
}
```

```
/*
 * 문제 제작자: 정재우 교수님
 * 문제 번호: 9
 * 문제 이름: 3월에 태어난 여성 회원 목록 출력하기
 * 문제 내용: 회원 목록을 출력하는 프로그램을 작성하세요. 회원 목록은 회원번호, 이름, 성별, 생년월일, 주민등록번호로 구성됩니다.
 * 문제 조건: 회원 목록은 회원번호, 이름, 성별, 생년월일, 주민등록번호로 구성됩니다.
 * 문제 예상점수: 10점
 */
#include <stdio.h>
#include <string.h>

struct Member {
    int member_id;
    char name[10];
    char gender;
    int birth;
    char ssn[10];
};

int main() {
    struct Member members[10];
    int i, count = 0;

    // 회원 정보 입력
    for (i = 0; i < 10; i++) {
        printf("회원 %d 번째 이름과 주민등록번호를 입력하세요: ", i + 1);
        scanf("%s", members[i].name);
        printf("회원 %d 번째 성별을 입력하세요: ", i + 1);
        scanf(" %c", &members[i].gender);
        printf("회원 %d 번째 생년월일을 입력하세요: ", i + 1);
        scanf("%d", &members[i].birth);
        printf("회원 %d 번째 주민등록번호를 입력하세요: ", i + 1);
        scanf("%s", members[i].ssn);
    }

    // 3월생 여성을 찾는다.
    for (i = 0; i < 10; i++) {
        if ((members[i].birth / 100) == 3 && (members[i].gender == 'F')) {
            count++;
            printf("3월에 태어난 여성 회원 목록입니다.\n");
            printf("회원 번호: %d, 이름: %s, 성별: %c, 생년월일: %d, 주민등록번호: %s\n", i + 1, members[i].name, members[i].gender, members[i].birth, members[i].ssn);
        }
    }

    if (count == 0) {
        printf("3월에 태어난 여성 회원이 없습니다.\n");
    }

    return 0;
}
```

풀이

problem10: 12세 이하인 여자환자 목록 출력하기(level1)

```
/*
 * 문제 제작자: 정재우 교수님
 * 문제 번호: 10
 * 문제 이름: 12세 이하인 여자환자 목록 출력하기
 * 문제 내용: 환자 목록을 출력하는 프로그램을 작성하세요. 환자 목록은 환자번호, 이름, 성별, 나이, 진료과목으로 구성됩니다.
 * 문제 조건: 환자 목록은 환자번호, 이름, 성별, 나이, 진료과목으로 구성됩니다.
 * 문제 예상점수: 10점
 */
#include <stdio.h>
#include <string.h>

struct Patient {
    int patient_id;
    char name[10];
    char gender;
    int age;
    char specialty[10];
};

int main() {
    struct Patient patients[10];
    int i, count = 0;

    // 환자 정보 입력
    for (i = 0; i < 10; i++) {
        printf("환자 %d 번째 이름과 진료과목을 입력하세요: ", i + 1);
        scanf("%s", patients[i].name);
        printf("환자 %d 번째 성별을 입력하세요: ", i + 1);
        scanf(" %c", &patients[i].gender);
        printf("환자 %d 번째 나이를 입력하세요: ", i + 1);
        scanf("%d", &patients[i].age);
        printf("환자 %d 번째 진료과목을 입력하세요: ", i + 1);
        scanf("%s", patients[i].specialty);
    }

    // 12세 이하 여성을 찾는다.
    for (i = 0; i < 10; i++) {
        if ((patients[i].age <= 12) && (patients[i].gender == 'F')) {
            count++;
            printf("12세 이하인 여자환자 목록입니다.\n");
            printf("환자 번호: %d, 이름: %s, 성별: %c, 나이: %d, 진료과목: %s\n", i + 1, patients[i].name, patients[i].gender, patients[i].age, patients[i].specialty);
        }
    }

    if (count == 0) {
        printf("12세 이하인 여자환자가 없습니다.\n");
    }

    return 0;
}
```

```
/*
 * 문제 제작자: 정재우 교수님
 * 문제 번호: 10
 * 문제 이름: 12세 이하인 여자환자 목록 출력하기
 * 문제 내용: 환자 목록을 출력하는 프로그램을 작성하세요. 환자 목록은 환자번호, 이름, 성별, 나이, 진료과목으로 구성됩니다.
 * 문제 조건: 환자 목록은 환자번호, 이름, 성별, 나이, 진료과목으로 구성됩니다.
 * 문제 예상점수: 10점
 */
#include <stdio.h>
#include <string.h>

struct Patient {
    int patient_id;
    char name[10];
    char gender;
    int age;
    char specialty[10];
};

int main() {
    struct Patient patients[10];
    int i, count = 0;

    // 환자 정보 입력
    for (i = 0; i < 10; i++) {
        printf("환자 %d 번째 이름과 진료과목을 입력하세요: ", i + 1);
        scanf("%s", patients[i].name);
        printf("환자 %d 번째 성별을 입력하세요: ", i + 1);
        scanf(" %c", &patients[i].gender);
        printf("환자 %d 번째 나이를 입력하세요: ", i + 1);
        scanf("%d", &patients[i].age);
        printf("환자 %d 번째 진료과목을 입력하세요: ", i + 1);
        scanf("%s", patients[i].specialty);
    }

    // 12세 이하 여성을 찾는다.
    for (i = 0; i < 10; i++) {
        if ((patients[i].age <= 12) && (patients[i].gender == 'F')) {
            count++;
            printf("12세 이하인 여자환자 목록입니다.\n");
            printf("환자 번호: %d, 이름: %s, 성별: %c, 나이: %d, 진료과목: %s\n", i + 1, patients[i].name, patients[i].gender, patients[i].age, patients[i].specialty);
        }
    }

    if (count == 0) {
        printf("12세 이하인 여자환자가 없습니다.\n");
    }

    return 0;
}
```

풀이

problem11: 모든 레코드 조회하기(level1)

풀이

problem12: 재구매가 일어난 상품과 회원 리스트 구하기
(level2)

풀이

problem13: 역순 정렬하기(level1)

풀이

problem14: 오프라인/온라인 판매 데이터 통합하기(level4)

풀이

problem15: 아픈 동물 찾기(level1)

풀이

problem16: 어린 동물 찾기(level1)

풀이

problem17: 동물의 아이디와 이름(level1)

풀이

problem18: 여러기준으로 정렬하기(level1)

풀이

problem19: 상위 n개 레코드(level1)

풀이

problem20: 조건에 맞는 회원수 구하기(level1)

풀이