

## <데이터베이스기초 과제 2>

정보통계학과 20210853 이민희

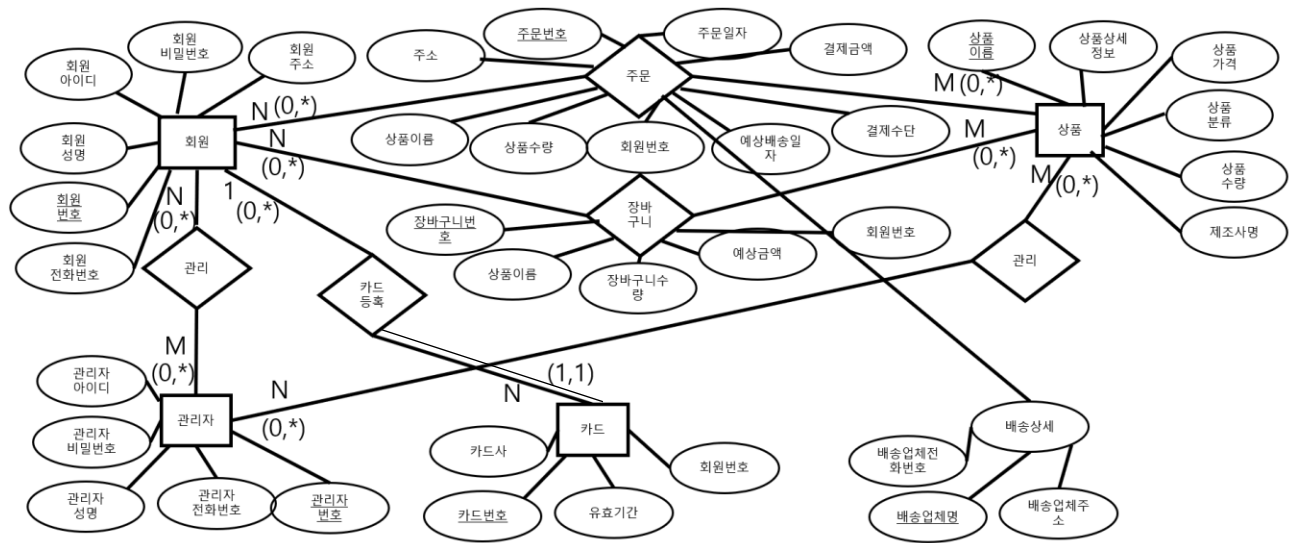
### 1. 데이터베이스가 사용되는 정보시스템 선택

옷 쇼핑물 정보시스템

### 2. 문제 정의 (객체, 속성, 관계 모두 정의-요구사항 10개정도) (최소한 4개 이상의 entity 필요, 6개 이상의 테이블)

- 1) 쇼핑몰에 회원으로 가입을 하려면 회원 아이디, 회원 비밀번호, 회원 성명, 회원 주소, 회원 전화번호를 입력해야 한다.
- 2) 쇼핑몰에 회원으로 가입이 되면 회원번호가 부여된다.
- 3) 쇼핑몰에는 관리자의 관리자 아이디, 관리자 비밀번호, 관리자 성명, 관리자 전화번호, 가 저장되어 있다.
- 4) 관리자가 등록되면 관리자가 부여된다.
- 5) 관리자는 상품, 회원을 관리한다.
- 6) 쇼핑몰에서는 보유한 상품들에 대해 상품명, 상품가격, 상품분류, 상품수량, 제조사명, 상품상세정보를 저장하고 있다.
- 7) 회원은 상품을 장바구니에 담을 수 있다. 장바구니에는 장바구니번호, 상품명, 상품수량, 예상금액, 회원번호가 저장 되어있다.
- 8) 회원은 상품을 주문할 때, 결제수단, 배송지를 입력해야 한다.
- 9) 회원이 주문을 하면 주문번호, 배송업체명, 상품명, 주문일자, 예상배송일자, 결제금액이 결정된다.
- 10) 배송상세에서 배송업체명, 배송업체전화번호, 배송업체주소를 확인할 수 있다.
- 11) 회원은 결제수단에 신용카드를 등록할 수 있다. 신용카드를 등록할 때에는 카드사, 카드번호, 유효기간을 입력해야 한다.

### 3. ER Diagram 그리기



### 4. 관계 스키마로 변형

회원(회원번호, 회원아이디, 회원비밀번호, 회원성명, 회원주소, 회원전화번호)

관리자(관리자번호, 관리자아이디, 관리자비밀번호, 관리자성명, 관리자전화번호)

상품(상품이름, 상품가격, 상품분류, 상품수량, 제조사명, 상품상세정보)

장바구니(장바구니번호, 상품이름, 장바구니수량, 예상금액, 회원번호)

\*상품이름은 foreign key, primary key

주문(주문번호, 상품이름, 회원번호, 주문일자, 예상배송일자, 결제금액, 결제수단, 배송업체명, 주소)

\*상품이름은 foreign key, primary key

배송상세 (배송업체명, 배송업체전화번호, 배송업체주소)

카드정보(카드사, 카드번호, 유효기간, 회원번호)

### 5. 각 테이블당 최소 10개 이상 데이터 입력

테이블 생성

```
1 • create database shopping;
2 • use shopping;
3
4
5 • create table users(
6     usernumber integer primary key,
7     userid varchar(20) not null,
8     userpassword varchar(20) not null,
9     username varchar(20) not null,
10    useraddress varchar(100),
11    userphone varchar(20) );
12
13 • create table managers(
14    managernumber varchar(20) primary key,
15    managerid varchar(20) not null,
16    managerpassword varchar(20) not null,
17    managername varchar(20) not null,
18    managerphone varchar(20) not null );

• create table product (
    proname varchar(20) not null primary key,
    proprice integer not null,
    procategory varchar(20) not null,
    proavailable integer not null,
    promanuname varchar(20) not null,
    prodetail varchar(100)
);

• create table basket (
    basnum varchar(20),
    proname varchar(20) not null,
    bascount integer,
    basprice integer not null,
    usernumber integer not null,
    primary key (basnum, proname),
    foreign key(proname) references product(proname),
    foreign key(usernumber) references users(usernumber)
);
```

- ⊖ 

```
CREATE TABLE delieverydetail (  
  delname varchar(20) not null primary key,  
  delphonenum varchar(20) not null,  
  deladdress varchar(100) not null  
);
```
  
- ⊖ 

```
create table card (  
  usernumber integer not null,  
  cardcompany varchar(20),  
  cardid varchar(20) not null primary key,  
  expiredate date,  
  foreign key(usernumber) references users(usernumber)  
);
```
  
- ⊖ 

```
create table orders (  
  ordernumber integer,  
  usernumber integer not null,  
  proname varchar(20) not null,  
  delname varchar(20) not null,  
  orderdate date not null,  
  deliverydate date not null,  
  orderprice integer not null,  
  payment varchar(20) not null,  
  address varchar(100) not null,  
  primary key (ordernumber, proname),  
  foreign key(proname) references product(proname),  
  foreign key(usernumber) references users(usernumber),  
  foreign key(delname) references delieverydetail(delname)  
);
```

## 1) users 테이블 데이터 입력

Limit to 50000 rows

```

69
70 • INSERT INTO users VALUES (1, 'jisung1', 'qwert', '박지성', '영국 맨체스터', '000-5000-0001');
71 • INSERT INTO users VALUES (2, 'yuna2', '1235', '김연아', '대한민국 서울', '000-6000-0001');
72 • INSERT INTO users VALUES (3, 'miran7', 'pass', '장미란', '대한민국 강원도', '000-7000-0001');
73 • INSERT INTO users VALUES (4, 'sinsu5', 'werd', '추신수', '미국 클리블랜드', '000-8000-0001');
74 • INSERT INTO users VALUES (5, 'heungmin', 'worrd', '손흥민', '영국 토트넘', '000-1235-1234');
75 • INSERT INTO users VALUES (6, 'jisun3', 'qwerty', '박지선', '영국 맨체스터', '000-5000-0001');
76 • INSERT INTO users VALUES (7, 'yua2', '12356', '김유아', '대한민국 서울', '000-3000-0001');
77 • INSERT INTO users VALUES (8, 'misu7', 'passs', '장미수', '대한민국 강원도', '000-4000-0001');
78 • INSERT INTO users VALUES (9, 'sinsung', 'werdd', '추신성', '미국 클리블랜드', '000-5000-0001');
79 • INSERT INTO users VALUES (10, 'sejin', 'wordd', '박세진', '대한민국 대전', '000-1225-1234');
80

```

Result Grid

	usernumber	userid	userpassword	username	useraddress	userphone
▶	1	jisung1	qwert	박지성	영국 맨체스터	000-5000-0001
	2	yuna2	1235	김연아	대한민국 서울	000-6000-0001
	3	miran7	pass	장미란	대한민국 강원도	000-7000-0001
	4	sinsu5	werd	추신수	미국 클리블랜드	000-8000-0001
	5	heungmin	worrd	손흥민	영국 토트넘	000-1235-1234
	6	jisun3	qwerty	박지선	영국 맨체스터	000-5000-0001
	7	yua2	12356	김유아	대한민국 서울	000-3000-0001
	8	misu7	passs	장미수	대한민국 강원도	000-4000-0001
	9	sinsung	werdd	추신성	미국 클리블랜드	000-5000-0001
	10	sejin	wordd	박세진	대한민국 대전	000-1225-1234
*	NULL	NULL	NULL	NULL	NULL	NULL

basket 1 card 2 delieverydetail 3 managers 4 orders 5 product 6 users 7 x

## 2) managers 테이블 데이터 입력

Limit to 50000 rows

```

80
81 • INSERT INTO managers VALUES (1, 'jisu1', 'qwer', '박지수', '001-5000-0001');
82 • INSERT INTO managers VALUES (2, 'yun2', '1235', '김연', '001-6000-0001');
83 • INSERT INTO managers VALUES (3, 'rose7', 'pass', '장미', '001-7000-0001');
84 • INSERT INTO managers VALUES (4, 'sinsunh5', 'werd', '추신선', '001-8000-0001');
85 • INSERT INTO managers VALUES (5, 'heungjin', 'worrd', '손흥진', '001-1235-1234');
86 • INSERT INTO managers VALUES (6, 'jisun3', 'qwerty', '유지선', '001-5000-0001');
87 • INSERT INTO managers VALUES (7, 'yua2', '12356', '유유아', '001-3000-0001');
88 • INSERT INTO managers VALUES (8, 'misu7', 'passs', '김미수', '001-4000-0001');
89 • INSERT INTO managers VALUES (9, 'sirsung', 'werdd', '강신성', '001-5000-0001');
90 • INSERT INTO managers VALUES (10, 'seojin', 'wordd', '이세진', '001-1225-1234');
91

```

Result Grid

	managernumber	managerid	managerpassword	managername	managerphone
▶	1	jisu1	qwer	박지수	001-5000-0001
	10	seojin	wordd	이세진	001-1225-1234
	2	yun2	1235	김연	001-6000-0001
	3	rose7	pass	장미	001-7000-0001
	4	sinsunh5	werd	추신선	001-8000-0001
	5	heungjin	worrd	손흥진	001-1235-1234
	6	jisun3	qwerty	유지선	001-5000-0001
	7	yua2	12356	유유아	001-3000-0001
	8	misu7	passs	김미수	001-4000-0001
	9	sirsung	werdd	강신성	001-5000-0001
*	NULL	NULL	NULL	NULL	NULL

basket 1 card 2 delieverydetail 3 managers 4 x orders 5 product 6 users 7

### 3) product 테이블 데이터 입력

Limit to 50000 rows

```

91
92 • INSERT INTO product VALUES('반팔티 슬림핏', 10000, '상의', 3, '난닝구', '면 함유율: 100%, 제조국: 중국');
93 • INSERT INTO product VALUES('반팔티 오버핏', 13000, '상의', 6, '난닝구', '면 함유율: 80%, 제조국: 중국');
94 • INSERT INTO product VALUES('반팔티 기본핏', 12000, '상의', 8, '난닝구', '면 함유율: 100%, 제조국: 일본');
95 • INSERT INTO product VALUES('롱스커트 데님', 35000, '하의', 23, '포에버영', '면 함유율: 100%, 제조국: 중국');
96 • INSERT INTO product VALUES('롱스커트 쉬폰', 36000, '하의', 66, '포에버영', '폴리에스터 함유율: 100%, 제조국: 중국');
97 • INSERT INTO product VALUES('롱스커트 실크', 36000, '하의', 65, '포에버영', '실크 함유율: 100%, 제조국: 몽골');
98 • INSERT INTO product VALUES('롱스커트 면', 35600, '하의', 77, '김씨네옷', '면 함유율: 100%, 제조국: 말레이시아');
99 • INSERT INTO product VALUES('레깅스', 13000, '하의', 3, '기묘한옷', '스판 함유율: 100%, 제조국: 중국');
100 • INSERT INTO product VALUES('조거팬츠', 27500, '하의', 5, '안녕한옷', '면 함유율: 80%, 제조국: 미국');
101 • INSERT INTO product VALUES('청바지', 33000, '하의', 6, '미즈', '면 함유율: 100%, 제조국: 미국');
102

```

Result Grid

	praname	proprice	procategory	proavailable	promanuname	prodetail
▶	레깅스	13000	하의	3	기묘한옷	스판 함유율: 100%, 제조국: 중국
	롱스커트 데님	35000	하의	23	포에버영	면 함유율: 100%, 제조국: 중국
	롱스커트 면	35600	하의	77	김씨네옷	면 함유율: 100%, 제조국: 말레이시아
	롱스커트 쉬폰	36000	하의	66	포에버영	폴리에스터 함유율: 100%, 제조국: 중국
	롱스커트 실크	36000	하의	65	포에버영	실크 함유율: 100%, 제조국: 몽골
	반팔티 기본핏	12000	상의	8	난닝구	면 함유율: 100%, 제조국: 일본
	반팔티 슬림핏	10000	상의	3	난닝구	면 함유율: 100%, 제조국: 중국
	반팔티 오버핏	13000	상의	6	난닝구	면 함유율: 80%, 제조국: 중국
	조거팬츠	27500	하의	5	안녕한옷	면 함유율: 80%, 제조국: 미국
	청바지	33000	하의	6	미즈	면 함유율: 100%, 제조국: 미국
•	NULL	NULL	NULL	NULL	NULL	NULL

basket 1 card 2 delieverydetail 3 managers 4 orders 5 product 6 x users 7

### 4) basket 테이블 데이터 입력

Limit to 50000 rows

```

102
103 • INSERT INTO basket VALUES(1, '반팔티 슬림핏', 1, 10000, 3);
104 • INSERT INTO basket VALUES(2, '반팔티 오버핏', 1, 13000, 4);
105 • INSERT INTO basket VALUES(3, '반팔티 기본핏', 1, 12000, 5);
106 • INSERT INTO basket VALUES(4, '롱스커트 데님', 1, 35000, 7);
107 • INSERT INTO basket VALUES(5, '롱스커트 쉬폰', 1, 36000, 6);
108 • INSERT INTO basket VALUES(6, '롱스커트 실크', 1, 36000, 1);
109 • INSERT INTO basket VALUES(7, '롱스커트 면', 1, 35600, 2);
110 • INSERT INTO basket VALUES(8, '레깅스', 1, 13000, 10);
111 • INSERT INTO basket VALUES(9, '조거팬츠', 1, 27500, 8);
112 • INSERT INTO basket VALUES(10, '청바지', 1, 33000, 9);
113

```

Result Grid

	basnum	praname	bascount	basproprice	usernumber
▶	1	반팔티 슬림핏	1	10000	3
	10	청바지	1	33000	9
	2	반팔티 오버핏	1	13000	4
	3	반팔티 기본핏	1	12000	5
	4	롱스커트 데님	1	35000	7
	5	롱스커트 쉬폰	1	36000	6
	6	롱스커트 실크	1	36000	1
	7	롱스커트 면	1	35600	2
	8	레깅스	1	13000	10
	9	조거팬츠	1	27500	8
•	NULL	NULL	NULL	NULL	NULL

basket 1 x card 2 delieverydetail 3 managers 4 orders 5 product 6 users 7



## 5) orders 테이블 데이터 입력

Database: madang\_book | madang\_book | 마당서점\_연습\_학생용 | 마당서점\_view\_학생용 | Univ\_DEMS | Univ\_Query | shopping\_query

Limit to 50000 rows

```

126
127 • INSERT INTO orders VALUES (1, 1, '반말티 술집', '한전택배2', STR_TO_DATE('2022-06-03', '%Y-%m-%d'), STR_TO_DATE('2022-06-04', '%Y-%m-%d'), 10000, 'card', '대한민국 서울');
128 • INSERT INTO orders VALUES (2, 1, '반말티 술집', '한전택배1', STR_TO_DATE('2022-06-02', '%Y-%m-%d'), STR_TO_DATE('2022-06-02', '%Y-%m-%d'), 10000, 'card', '대한민국 서울');
129 • INSERT INTO orders VALUES (3, 2, '반말티 오버핏', '한전택배3', STR_TO_DATE('2022-05-29', '%Y-%m-%d'), STR_TO_DATE('2022-06-03', '%Y-%m-%d'), 13000, 'card', '대한민국 서울');
130 • INSERT INTO orders VALUES (4, 3, '반말티 오버핏', '대한택배1', STR_TO_DATE('2022-05-31', '%Y-%m-%d'), STR_TO_DATE('2022-06-03', '%Y-%m-%d'), 13000, 'card', '영국 토트넘');
131 • INSERT INTO orders VALUES (5, 4, '반말티 기본핏', '대한택배2', STR_TO_DATE('2022-05-29', '%Y-%m-%d'), STR_TO_DATE('2022-06-04', '%Y-%m-%d'), 12000, 'cash', '영국 토트넘');
132 • INSERT INTO orders VALUES (6, 1, '반말티 기본핏', '대한택배3', STR_TO_DATE('2022-05-29', '%Y-%m-%d'), STR_TO_DATE('2022-06-03', '%Y-%m-%d'), 12000, 'cash', '영국 토트넘');
133 • INSERT INTO orders VALUES (7, 4, '롱스커트 데님', '대한택배4', STR_TO_DATE('2022-05-30', '%Y-%m-%d'), STR_TO_DATE('2022-06-09', '%Y-%m-%d'), 35000, 'card', '대한민국 강원도');
134 • INSERT INTO orders VALUES (8, 3, '롱스커트 데님', '한전통운1', STR_TO_DATE('2022-06-01', '%Y-%m-%d'), STR_TO_DATE('2022-06-10', '%Y-%m-%d'), 35000, 'cash', '대한민국 서울');
135 • INSERT INTO orders VALUES (9, 2, '롱스커트 위문', '한전통운2', STR_TO_DATE('2022-05-30', '%Y-%m-%d'), STR_TO_DATE('2022-06-07', '%Y-%m-%d'), 36000, 'card', '대한민국 서울');
136 • INSERT INTO orders VALUES (10, 3, '롱스커트 위문', '한전통운3', STR_TO_DATE('2022-05-29', '%Y-%m-%d'), STR_TO_DATE('2022-06-03', '%Y-%m-%d'), 36000, 'card', '대한민국 강원도');

```

ordernumber	userid	praname	delname	orderdate	deliverydate	orderprice	payment	address
1	1	반말티 술집	한전택배2	2022-06-03	2022-06-04	10000	card	대한민국 서울
2	1	반말티 술집	한전택배1	2022-06-02	2022-06-02	10000	card	대한민국 서울
3	2	반말티 오버핏	한전택배3	2022-05-29	2022-06-03	13000	card	대한민국 서울
4	3	반말티 오버핏	대한택배1	2022-05-31	2022-06-03	13000	card	영국 토트넘
5	4	반말티 기본핏	대한택배2	2022-05-29	2022-06-04	12000	cash	영국 토트넘
6	1	반말티 기본핏	대한택배3	2022-05-29	2022-06-03	12000	cash	영국 토트넘
7	4	롱스커트 데님	대한택배4	2022-05-30	2022-06-09	35000	card	대한민국 강원도
8	3	롱스커트 데님	한전통운1	2022-06-01	2022-06-10	35000	cash	대한민국 서울
9	2	롱스커트 위문	한전통운2	2022-05-30	2022-06-07	36000	card	대한민국 서울
10	3	롱스커트 위문	한전통운3	2022-05-29	2022-06-03	36000	card	대한민국 강원도

basket 73 | card 74 | deliverydetail 75 | managers 76 | orders 77 | product 78 | users 79 | Result 80 | users 81 | product 82 | deliverydetail 83

## 6) deliverydetail 테이블 데이터 입력

Limit to 50000 rows

```

113
114 • INSERT INTO deliverydetail VALUES ('대한택배1', '111-5000-0001', '대한민국 강원도');
115 • INSERT INTO deliverydetail VALUES ('대한택배2', '222-5000-0001', '대한민국 강원도');
116 • INSERT INTO deliverydetail VALUES ('대한택배3', '333-5000-0001', '대한민국 강원도');
117 • INSERT INTO deliverydetail VALUES ('대한택배4', '444-5000-0001', '대한민국 강원도');
118 • INSERT INTO deliverydetail VALUES ('한전택배1', '011-7000-0001', '대한민국 대전');
119 • INSERT INTO deliverydetail VALUES ('한전택배2', '021-7000-0001', '대한민국 대전');
120 • INSERT INTO deliverydetail VALUES ('한전택배3', '031-7000-0001', '대한민국 대전');
121 • INSERT INTO deliverydetail VALUES ('한전통운1', '011-5000-0001', '대한민국 강원도');
122 • INSERT INTO deliverydetail VALUES ('한전통운2', '011-5000-0001', '대한민국 강원도');
123 • INSERT INTO deliverydetail VALUES ('한전통운3', '011-5000-0001', '대한민국 강원도');

```

delname	delphonenum	deladdress
대한택배1	111-5000-0001	대한민국 강원도
대한택배2	222-5000-0001	대한민국 강원도
대한택배3	333-5000-0001	대한민국 강원도
대한택배4	444-5000-0001	대한민국 강원도
한전택배1	011-7000-0001	대한민국 대전
한전택배2	021-7000-0001	대한민국 대전
한전택배3	031-7000-0001	대한민국 대전
한전통운1	011-5000-0001	대한민국 강원도
한전통운2	011-5000-0001	대한민국 강원도
한전통운3	011-5000-0001	대한민국 강원도

basket 1 | card 2 | deliverydetail 3 | managers 4 | orders 5 | product 6 | users 7

## 7) card 테이블 데이터 입력

The screenshot shows a SQL IDE with a script editor containing 11 INSERT statements for the 'card' table. The statements use the STR\_TO\_DATE function to convert string dates to actual dates. Below the script editor, the 'Result Grid' shows the data inserted into the table.

usernumber	cardcompany	cardid	expiredate
7	비씨카드	000-12345-003431	2029-11-03
9	하나카드	010-12345-003431	2027-06-03
1	비씨카드	011-12345-003431	2026-06-03
1	비씨카드	011-12445-003431	2024-06-30
1	비씨카드	014-12345-003431	2023-06-03
2	삼성카드	016-12345-003431	2028-05-03
1	신세계카드	023-12345-003431	2026-05-03
3	비씨카드	111-12345-003431	2026-04-03
5	비자카드	111-22345-003431	2026-07-03
NULL	NULL	NULL	NULL

At the bottom, there are tabs for different tables: basket 1, card 2 (selected), deliverydetail 3, managers 4, orders 5, product 6, and users 7.

## 6. 질의 최소 10개 이상 만들고 결과 출력

(일반 질의 3, 복합질의 3, aggregation 함수질의 3, 기타)

1) 쇼핑몰에 있는 상품의 총 개수를 보이시오.

The screenshot shows a SQL IDE with a script editor containing a single SQL query. Below the script editor, the 'Result Grid' shows the result of the query.

```

167      -- (1) 쇼핑몰에 있는 상품의 총 개수를 보이시오.
168 •    select count(*) as product_num from product;
  
```

product_num
10

2) 모든 고객의 이름, 주소를 보이시오.



```

170      -- (2) 박세진 회원의 전화번호를 찾으시오.
171  •   select userphone from users where username='박세진';
172

```

The screenshot shows a database query result in a 'Result Grid' window. The query is: `select userphone from users where username='박세진';`. The result grid has two columns: 'userphone' and a value '000-1225-1234'.

userphone
000-1225-1234

3) 가격이 35000원 이상인 상품의 이름을 보이시오.

```

173      -- (3) 가격이 35000원 이상인 상품의 이름을 보이시오.
174  •   select prname from product where proprice>=35000;

```

The screenshot shows a database query result in a 'Result Grid' window. The query is: `select prname from product where proprice>=35000;`. The result grid has two columns: 'prname' and a list of product names: '롱스커트 데님', '롱스커트 면', '롱스커트 쉬폰', '롱스커트 실크', and 'NULL'.

prname
롱스커트 데님
롱스커트 면
롱스커트 쉬폰
롱스커트 실크
NULL

4) 대한민국 강원도에 위치한 배송업체명과 배송업체전화번호를 보이시오.

```

176      -- (4) 대한민국 강원도에 위치한 배송업체명과 배송업체전화번호를 보이시오.
177  •   select delname, delphonenum from delieverydetail where deladdress = '대한민국 강원도';
178

```

The screenshot shows a database query result in a 'Result Grid' window. The query is: `select delname, delphonenum from delieverydetail where deladdress = '대한민국 강원도';`. The result grid has two columns: 'delname' and 'delphonenum'. The results are: '대한택배1' (111-5000-0001), '대한택배2' (222-5000-0001), '대한택배3' (333-5000-0001), '대한택배4' (444-5000-0001), '한전통운1' (011-5000-0001), '한전통운2' (011-5000-0001), '한전통운3' (011-5000-0001), and 'NULL' (NULL).

delname	delphonenum
대한택배1	111-5000-0001
대한택배2	222-5000-0001
대한택배3	333-5000-0001
대한택배4	444-5000-0001
한전통운1	011-5000-0001
한전통운2	011-5000-0001
한전통운3	011-5000-0001
NULL	NULL

5) 쇼핑몰 회원이름별 장바구니 총예상금액을 보이시오

```

181      -- (5) 쇼핑몰 회원이름별 장바구니 총예상금액을 보이시오.
182 • select ( select username
183             from users
184             where users.usernumber=basket.usernumber) 'name', sum(basprice) 'totalbasprice'
185   from basket
186   group by basket.usernumber;
187

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	name	totalbasprice		
▶	박지성	36000		
	김연아	35600		
	장미란	10000		
	추신수	13000		
	손흥민	12000		
	박지선	36000		
	김유아	35000		
	장미수	27500		
	추신성	33000		
	박세진	13000		

6) 평균 상품가격보다 낮은 상품을 보이시오.

```

188      -- (6) 평균 상품가격보다 낮은 상품을 보이시오.
189 • select proname, proprice
190   from product
191   where proprice <= (select avg(proprice) from product);
192

```

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
	praname	proprice			
▶	레깅스	13000			
	반팔티 기본핏	12000			
	반팔티 슬림핏	10000			
	반팔티 오버핏	13000			
*	NULL	NULL			

7) 쇼핑몰 주문번호별 배송업체명을 보이시오.

```

192
193      -- (7) 쇼핑몰 주문번호별 배송업체명을 보이시오.
194 • select ( select delname
195             from deliverydetail
196             where orders.delname=deliverydetail.delname) '배송업체명', ordernumber '주문번호'
197   from orders
198   group by orders.ordernumber;
199

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	배송업체명	주문번호		
▶	한전택배2	1		
	한전택배1	2		
	한전택배3	3		
	대한택배1	4		
	대한택배2	5		
	대한택배3	6		
	대한택배4	7		
	한전통운1	8		
	한전통운2	9		
	한전통운3	10		

8) 고객이 주문한 상품의 총 판매액, 평균값, 최저가, 최고가를 구하시오.

```

200      -- (8) 고객이 주문한 상품의 총 판매액, 평균값, 최저가, 최고가를 구하시오.
201 •    select sum(orderprice) as '총 판매액', avg(orderprice) as '평균값',
202           min(orderprice) as '최저가', max(orderprice) as '최고가'
203      from orders;
204

```

Result Grid				
Filter Rows: <input type="text"/>				
Export:  Wrap Cell Content:				
	총 판매액	평균값	최저가	최고가
▶	212000	21200.0000	10000	36000

9) 판매되고 있는 상품의 금액의 평균값, 최저가, 최고가를 구하시오.

```

205      -- (9) 판매되고 있는 상품의 금액의 평균값, 최저가, 최고가를 구하시오.
206 •    select avg(proprice) as '평균값',
207           min(proprice) as '최저가', max(proprice) as '최고가'
208      from product;

```

Result Grid			
Filter Rows: <input type="text"/>			
Export:  Wrap Cell Content:			
	평균값	최저가	최고가
▶	25110.0000	10000	36000

10) 가장 비싼 상품의 상품명과 상품상세정보를 보이시오.

```

210      -- (10) 가장 비싼 상품의 상품명과 상품상세정보를 보이시오.
211 •    select proname, prodetail
212      from product
213     where proprice = (select max(proprice) from product);
214

```

Result Grid	
Filter Rows: <input type="text"/>	
Edit:    Export/Import:   Wrap Cell Content:	
proname	prodetail
▶ 롱스커트 쉬폰	폴리에스터 함유율: 100%, 제조국: 중국
롱스커트 실크	실크 함유율: 100%, 제조국: 몽골
* NULL	NULL