

## 一、题目描述

①数据库名字，数据库的简短说明

电商管理数据库，用于记录用户购买商品的操作以及商品的提供商等其他详细信息

②数据库将要存储的数据

用户 user	用户名	user_name
	用户身份证号	identity_card
	用户住址	address
	用户注册密码	user_password
	用户电话	telephone
	用户邮箱	email

订单 order_table	订单号	order_number
	订单状态	order_state

商品 product	商品编号	product_number
	商品价格	product_price
	商品名称	product_name
	商品描述	product_description

供应商 supplier	供应商编号	supplier_number
	供应商名称	supplier_name
	供应商联系人	supplier_linkman
	供应商电话	supplier_telephone
	供应商供应产品	supplier_goods

购物车 shopping_cart	购物车编号	shopping_cart_number
	用户名	user_name

其他	订单选择某种商品的数量	quantity
	供应商供应某种商品的数量	supply_number
	购物车选择某种商品的数量	quantity_of_product
	商品加入购物车的时间	join_date

③数据间存在的联系

order\_table 与 user（多对一）

user 与 shopping\_cart（一对一）

order\_table 与 product（多对多）

shopping\_cart 与 product（多对多）

supplier 与 product（多对多）

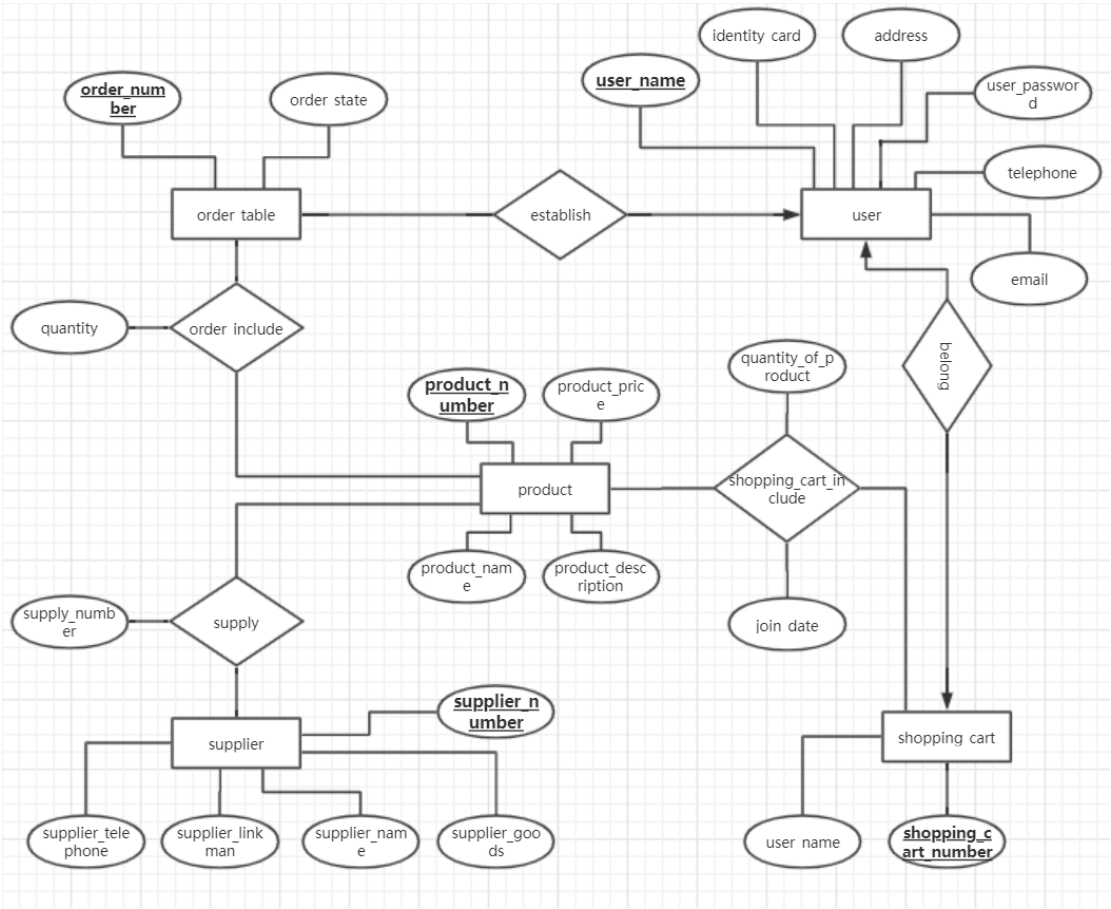
④可能经常进行的操作

1.查找某个用户所有的订单以及订单的详细内容（商品名称，商品价格，商品数量）

2.查看购物车的内容

3.查看某种商品的详细信息（包括该商品的供应商等）

二、数据库建模（ER 图）



三、表

user(user\_name,identity\_card,address,user\_password,telephone,email)  
order\_table(order\_number,order\_state)  
establish(order\_number, user\_name)  
shopping\_cart(shopping\_cart\_number,user\_name)  
product(product\_number,product\_price,product\_name,product\_description)  
order\_include(order\_number,product\_number,quantity)  
shopping\_cart\_include(shopping\_cart\_number,product\_number,quantity\_of\_product,join\_date)  
supplier(supplier\_number,supplier\_name,supplier\_linkman,supplier\_telephone,supplier\_goods)  
supply(supplier\_number,product\_number,supply\_number)

四、各个表的结构

user 用户表		
列名	数据类型	描述
user_name	varchar(30)	主键
identity_card	char(18)	NOT NULL
user_password	char(6)	NOT NULL

telephone	char(11)	NOT NULL
email	varchar(60)	NULL
address	varchar(60)	NOT NULL

product 商品表		
列名	数据类型	描述
product_number	char(10)	主键
product_name	varchar(30)	NOT NULL
product_description	varchar(60)	NULL
product_price	unsigned tinyint	NOT NULL

order_table 订单表		
列名	数据类型	描述
order_number	char(10)	主键
order_state	boolean	NOT NULL, 默认为 0 未付款, 1 已付款

establish 创建表		
列名	数据类型	描述
order_number	char(10)	主键
user_name	varchar(30)	外键, NOT NULL

order_include 订单包含表		
列名	数据类型	描述
order_number	char(10)	主键, 外键
product_number	char(10)	主键, 外键
quantity	tinyint	null default 0

shopping_cart 购物车表		
列名	数据类型	描述
shopping_cart_number	char(10)	主键
user_name	varchar(30)	外键

shopping_cart_include 购物车包含表		
列名	数据类型	描述
shopping_cart_number	char(10)	主键, 外键
product_number	char(10)	主键, 外键
quantity_of_product	tinyint	NULL DEFAULT 0
join_date	datetime	NOT NULL

supplier 供应商表		
---------------	--	--

列名	数据类型	描述
supplier_number	char(5)	主键
supplier_name	varchar(30)	NOT NULL
supplier_linkman	varchar(30)	NULL
supplier_telephone	char(11)	NOT NULL
supplier_goods	varchar(30)	NOT NULL

supply 供应表		
列名	数据类型	描述
supplier_number	char(5)	主键
product_number	char(10)	主键
supply_number	smallint	NULL,默认是 0, 取值>=0

## 五、样例数据

### 1.user 表

```

5  -- -----
6  -- Table structure for user
7  -- -----
8  -- CREATE TABLE IF NOT EXISTS user(
9  --     user_name VARCHAR(30) NOT NULL,
10 --     identity_card CHAR(18) NOT NULL,
11 --     user_password char(6) NOT NULL,
12 --     telephone char(11) NOT NULL,
13 --     email VARCHAR(60) DEFAULT NULL,
14 --     user_address VARCHAR(60) NOT NULL,
15 --     PRIMARY KEY(user_name)
16 -- )ENGINE=InnoDB;
17
18 -- -----
19 -- Records of user
20 -- -----
21 -- INSERT INTO user VALUES('user1','100000000000000000','111111','111111111111','1111@qq.com','北京市朝阳区'),
22 -- ('user2','10000000000000000001','222222','222222222222','2222@qq.com','北京市海淀区'),
23 -- ('user3','10000000000000000002','333333','333333333333','3333@qq.com','北京市海淀区'),
24 -- ('user4','10000000000000000003','444444','444444444444','4444@qq.com','北京市通州区'),
25 -- ('user5','10000000000000000004','555555','555555555555','5555@qq.com','北京市顺义区'),
26 -- ('user6','10000000000000000005','666666','666666666666','6666@qq.com','北京市丰台区'),
27 -- ('user7','10000000000000000006','777777','777777777777','7777@qq.com','北京市大兴区'),
28 -- ('user8','10000000000000000007','888888','888888888888','8888@qq.com','北京市通州区'),
29 -- ('user9','10000000000000000008','999999','999999999999','9999@qq.com','北京市东城区'),
30 -- ('user10','10000000000000000009','000000','000000000000','0000@qq.com','北京市海淀区');
31
32 -- SELECT * FROM user ORDER BY identity_card;

```

user_name	identity_card	user_password	telephone	email	user_address
user1	100000000000000000	111111	11111111111	1111@qq.com	北京市朝阳区
user2	100000000000000001	222222	22222222222	2222@qq.com	北京市海淀区
user3	100000000000000002	333333	33333333333	3333@qq.com	北京市海淀区
user4	100000000000000003	444444	44444444444	4444@qq.com	北京市通州区
user5	100000000000000004	555555	55555555555	5555@qq.com	北京市顺义区
user6	100000000000000005	666666	66666666666	6666@qq.com	北京市丰台区
user7	100000000000000006	777777	77777777777	7777@qq.com	北京市大兴区
user8	100000000000000007	888888	88888888888	8888@qq.com	北京市通州区
user9	100000000000000008	999999	99999999999	9999@qq.com	北京市东城区
user10	100000000000000009	000000	00000000000	0000@qq.com	北京市海淀区

## 2.product 表

```

35  -- -----
36  -- Table structure for product
37  -- -----
38  -- CREATE TABLE IF NOT EXISTS product(
39  --     product_number char(10) NOT NULL,
40  --     product_name VARCHAR(30) NOT NULL,
41  --     product_description VARCHAR(60) DEFAULT NULL,
42  --     product_price TINYINT NOT NULL,
43  --     PRIMARY KEY(product_name)
44  -- )ENGINE=InnoDB;
45
46  -- -----
47  -- Records of product
48  -- -----
49  -- INSERT INTO product VALUES('0000000000','曲奇','脆就一个字，我只说一次',30),
50  -- ('0000000001','夹心饼干','',5),
51  -- ('0000000002','棒棒糖','混合口味的每袋是5个口味混合',26),
52  -- ('0000000003','大白兔奶糖','入口即化',20),
53  -- ('0000000004','乐事薯片','干脆可口',10),
54  -- ('0000000005','可口可乐','一箱24罐，冰冻后饮用口感更佳',50),
55  -- ('0000000006','红牛','一箱24罐，激发你的能量',125),
56  -- ('0000000007','椰树牌椰汁','植物蛋白饮料',85),
57  -- ('0000000008','豆本豆奶','早餐奶低脂低糖原味豆奶250ml*24',56),
58  -- ('0000000009','真果粒','牛奶+果粒，自然有乐趣',60);
59
60  -- SELECT * FROM product ORDER BY product_number;
```

product_number	product_name	product_description	product_price
0000000000	曲奇	脆就一个字，我只说一次	30
0000000001	夹心饼干		5
0000000002	棒棒糖	混合口味的每袋是5个口味混合	26
0000000003	大白兔奶糖	入口即化	20
0000000004	乐事薯片	干脆可口	10
0000000005	可口可乐	一箱24罐，冰冻后饮用口感更佳	50
0000000006	红牛	一箱24罐，激发你的能量	125
0000000007	椰树牌椰汁	植物蛋白饮料	85
0000000008	豆本豆奶	早餐奶低脂低糖原味豆奶250ml*24	56
0000000009	真果粒	牛奶+果粒，自然有乐趣	60

### 3.order\_table 表

```

64  -- -----
65  -- Table structure for order_table
66  -- -----
67  -- CREATE TABLE IF NOT EXISTS order_table(
68  --     order_number char(10) NOT NULL,
69  --     order_state boolean DEFAULT 0 ,
70  --     PRIMARY KEY(order_number)
71  -- )ENGINE=InnoDB;
72
73  -- -----
74  -- Records of order_table
75  -- -----
76  -- INSERT INTO order_table VALUES('0000000000',0),
77  -- ('0000000001',0),
78  -- ('0000000002',0),
79  -- ('0000000003',0),
80  -- ('0000000004',0),
81  -- ('0000000005',0),
82  -- ('0000000006',0),
83  -- ('0000000007',0);
84  -- ('0000000008',0),
85  -- ('0000000009',0),
86  -- ('0000000010',0),
87  -- ('0000000011',0);
88
89  -- SELECT * FROM order_table;

```

order_number	order_state
0000000000	0
0000000001	0
0000000002	0
0000000003	0
0000000004	0
0000000005	0
0000000006	0
0000000007	0
0000000008	0
0000000009	0
0000000010	0
0000000011	0

#### 4.establish 表

```

91  -- -----
92  -- Table structure for establish
93  -- -----
94  -- CREATE TABLE IF NOT EXISTS establish(
95  --     order_number char(10) NOT NULL,
96  --     user_name VARCHAR(30) NOT NULL,
97  --     PRIMARY KEY(order_number),
98  --     CONSTRAINT fk_establish_user_name
99  --     FOREIGN KEY(user_name) REFERENCES user(user_name)
100 -- )ENGINE=InnoDB;
101
102 -- -----
103 -- Records of establish
104 -- -----
105 -- INSERT INTO establish VALUES('0000000000','user1'),
106 -- ('0000000001','user1'),
107 -- ('0000000002','user2'),
108 -- ('0000000003','user3'),
109 -- ('0000000004','user4'),
110 -- ('0000000005','user5'),
111 -- ('0000000006','user6'),
112 -- ('0000000007','user7'),
113 -- ('0000000008','user8'),
114 -- ('0000000009','user9'),
115 -- ('0000000010','user10'),
116 -- ('0000000011','user2');
117
118 -- SELECT * FROM establish ORDER BY order_number;

```

order_number	user_name
0000000000	user1
0000000001	user1
0000000002	user2
0000000003	user3
0000000004	user4
0000000005	user5
0000000006	user6
0000000007	user7
0000000008	user8
0000000009	user9
0000000010	user10
0000000011	user2

5.order\_include 表



```

120 -- -----
121 -- Table structure for order_include
122 -- -----
123 -- CREATE TABLE IF NOT EXISTS order_include(
124 --     order_number char(10) NOT NULL,
125 --     product_number char(10) NOT NULL,
126 --     quantity TINYINT NULL DEFAULT 0,
127 --     PRIMARY KEY(order_number,product_number)
128 -- )ENGINE=InnoDB;
129
130 -- -----
131 -- Records of order_include
132 -- -----
133 -- INSERT INTO order_include VALUES('000000000', '000000000',2),
134 -- ('000000000', '000000001',2),
135 -- ('000000000', '000000002',2),
136 -- ('000000001', '000000001',2),
137 -- ('000000001', '000000002',1),
138 -- ('000000001', '000000003',3),
139 -- ('000000002', '000000008',1),
140 -- ('000000002', '000000001',1),
141 -- ('000000003', '000000003',5),
142 -- ('000000003', '000000006',1),
143 -- ('000000003', '000000004',2),
144 -- ('000000004', '000000007',3),
145 -- ('000000004', '000000002',2),
146 -- ('000000005', '000000008',1),
147 -- ('000000005', '000000007',2),
148 -- ('000000005', '000000006',2),
149 -- ('000000006', '000000004',1),
150 -- ('000000006', '000000001',2),
151 -- ('000000007', '000000001',1),
152 -- ('000000007', '000000002',2),
153 -- ('000000008', '000000009',2),
154 -- ('000000008', '000000008',1),
155 -- ('000000008', '000000007',2),
156 -- ('000000009', '000000005',3),
157 -- ('000000009', '000000002',1),
158 -- ('000000009', '000000008',4),
159 -- ('000000010', '000000002',1),
160 -- ('000000010', '000000006',2),
161 -- ('000000011', '000000000',6),
162 -- ('000000011', '000000009',1);
163
164 -- SELECT * FROM order_include;

```

order_number	product_number	quantity
0000000000	0000000000	2
0000000000	0000000001	2
0000000000	0000000002	2
0000000001	0000000001	2
0000000001	0000000002	1
0000000001	0000000003	3
0000000002	0000000001	1
0000000002	0000000008	1
0000000003	0000000003	5
0000000003	0000000004	2
0000000003	0000000006	1
0000000004	0000000002	2
0000000004	0000000007	3
0000000005	0000000006	2
0000000005	0000000007	2
0000000005	0000000008	1
0000000006	0000000001	2
0000000006	0000000004	1
0000000007	0000000001	1
0000000007	0000000002	2
0000000008	0000000007	2
0000000008	0000000008	1
0000000008	0000000009	2
0000000009	0000000002	1
0000000009	0000000005	3
0000000009	0000000008	4
0000000010	0000000002	1
0000000010	0000000006	2
0000000011	0000000000	6
0000000011	0000000009	1

6.shopping\_cart 表

```

167  -- -----
168  -- Table structure for shopping_cart
169  -- -----
170  -- CREATE TABLE IF NOT EXISTS shopping_cart(
171  --     shopping_cart_number char(10) NOT NULL,
172  --     user_name VARCHAR(30) NOT NULL,
173  --     PRIMARY KEY(shopping_cart_number),
174  --     CONSTRAINT fk_shopping_cart_user_name
175  --     FOREIGN KEY(user_name) REFERENCES user(user_name)
176
177  -- )ENGINE=InnoDB;
178
179  -- -----
180  -- Records of shopping_cart
181  -- -----
182  -- INSERT INTO shopping_cart VALUES('0000000001','user1'),
183  -- ('0000000002','user2'),
184  -- ('0000000003','user3'),
185  -- ('0000000004','user4'),
186  -- ('0000000005','user5'),
187  -- ('0000000006','user6'),
188  -- ('0000000007','user7'),
189  -- ('0000000008','user8'),
190  -- ('0000000009','user9'),
191  -- ('0000000010','user10');
192
193  -- SELECT * FROM shopping_cart ORDER BY shopping_cart_number;

```

shopping_cart_number	user_name
0000000001	user1
0000000002	user2
0000000003	user3
0000000004	user4
0000000005	user5
0000000006	user6
0000000007	user7
0000000008	user8
0000000009	user9
0000000010	user10

7.shopping\_cart\_include 表

```

199  -- -----
200  -- Table structure for shopping_cart_include
201  -- -----
202  -- CREATE TABLE IF NOT EXISTS shopping_cart_include(
203  --     shopping_cart_number char(10) NOT NULL,
204  --     product_number char(10) NOT NULL,
205  --     join_date DATETIME NOT NULL,
206  --     quantity_of_product TINYINT NULL DEFAULT 0,
207  --     PRIMARY KEY(shopping_cart_number,product_number)
208  -- )ENGINE=InnoDB;
209
210  -- -----
211  -- Records of shopping_cart_include
212  -- -----
213  -- INSERT INTO shopping_cart_include VALUES('0000000001','0000000001',now(),1),
214  -- ('0000000001','0000000002',now(),2),
215  -- ('0000000001','0000000003',now(),1),
216  -- ('0000000002','0000000001',now(),2),
217  -- ('0000000003','0000000004',now(),1),
218  -- ('0000000003','0000000005',now(),2),
219  -- ('0000000003','0000000006',now(),3),
220  -- ('0000000004','0000000007',now(),2),
221  -- ('0000000004','0000000005',now(),1),
222  -- ('0000000004','0000000003',now(),2),
223  -- ('0000000005','0000000000',now(),1),
224  -- ('0000000005','0000000008',now(),2),
225  -- ('0000000005','0000000007',now(),1),
226  -- ('0000000006','0000000009',now(),2),
227  -- ('0000000006','0000000008',now(),1),
228  -- ('0000000006','0000000007',now(),1),
229  -- ('0000000007','0000000000',now(),2),
230  -- ('0000000007','0000000002',now(),3),
231  -- ('0000000007','0000000003',now(),1),
232  -- ('0000000008','0000000001',now(),2),
233  -- ('0000000008','0000000006',now(),3),
234  -- ('0000000008','0000000003',now(),2),
235  -- ('0000000009','0000000009',now(),1),
236  -- ('0000000009','0000000003',now(),2),
237  -- ('0000000009','0000000004',now(),1),
238  -- ('0000000010','0000000006',now(),2),
239  -- ('0000000010','0000000007',now(),1),
240  -- ('0000000010','0000000002',now(),2);
241
242  -- SELECT * FROM shopping_cart_include;

```

shopping_cart_number	product_number	join_date	quantity_of_product
0000000001	0000000001	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	1
0000000001	0000000002	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	2
0000000001	0000000003	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	1
0000000002	0000000001	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	2
0000000003	0000000004	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	1
0000000003	0000000005	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	2
0000000003	0000000006	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	3
0000000004	0000000003	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	2
0000000004	0000000005	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	1
0000000004	0000000007	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	2
0000000005	0000000000	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	1
0000000005	0000000007	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	1
0000000005	0000000008	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	2
0000000006	0000000007	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	1
0000000006	0000000008	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	1
0000000006	0000000009	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	2
0000000007	0000000000	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	2
0000000007	0000000002	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	3
0000000007	0000000003	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	1
0000000008	0000000001	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	2
0000000008	0000000003	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	2
0000000008	0000000006	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	3
0000000009	0000000003	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	2
0000000009	0000000004	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	1
0000000009	0000000009	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	1
0000000010	0000000002	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	2
0000000010	0000000006	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	2
0000000010	0000000007	Wed Nov 27 2019 11:05:34 GMT+0800 (中国标准时间)	1

8.supplier 表

```

245 -- -----
246 -- Table structure for supplier
247 -- -----
248 -- CREATE TABLE IF NOT EXISTS supplier(
249 --     supplier_number char(5) NOT NULL,
250 --     supplier_name varchar(30) NOT NULL,
251 --     supplier_linkman varchar(30) NULL,
252 --     supplier_telephone char(11) NOT NULL,
253 --     supplier_goods varchar(30) NOT NULL,
254 --     PRIMARY KEY(supplier_number)
255 -- )ENGINE=InnoDB;
256
257 -- -----
258 -- Records of supplier
259 -- -----
260 -- INSERT INTO supplier VALUES('00000','全部生产有限公司','全部联系','12345670000','全部'),
261 -- ('00001','豆奶生产有限公司','豆奶联系','12345671111','豆奶'),
262 -- ('00002','椰汁生产有限公司','椰汁联系','12345672222','椰汁'),
263 -- ('00003','红牛生产有限公司','红牛联系','12345673333','红牛'),
264 -- ('00004','可口可乐生产有限公司','可口可乐联系','12345674444','可口可乐'),
265 -- ('00005','薯片生产有限公司','薯片联系','12345675555','薯片'),
266 -- ('00006','大白兔奶糖生产有限公司','大白兔奶糖联系','12345676666','大白兔奶糖'),
267 -- ('00007','棒棒糖生产有限公司','棒棒糖联系','12345677777','棒棒糖'),
268 -- ('00008','夹心饼干生产有限公司','夹心饼干联系','12345678888','夹心饼干'),
269 -- ('00009','曲奇生产有限公司','曲奇联系','12345679999','曲奇'),
270 -- ('00010','真果粒生产有限公司','真果粒联系','12345680000','真果粒');
271
272 -- SELECT * FROM supplier;

```

supplier_number	supplier_name	supplier_linkman	supplier_telephone	supplier_goods
00000	全部生产有限公司	全部联系	12345670000	全部
00001	豆奶生产有限公司	豆奶联系	12345671111	豆奶
00002	椰汁生产有限公司	椰汁联系	12345672222	椰汁
00003	红牛生产有限公司	红牛联系	12345673333	红牛
00004	可口可乐生产有限公司	可口可乐联系	12345674444	可口可乐
00005	薯片生产有限公司	薯片联系	12345675555	薯片
00006	大白兔奶糖生产有限公司	大白兔奶糖联系	12345676666	大白兔奶糖
00007	棒棒糖生产有限公司	棒棒糖联系	12345677777	棒棒糖
00008	夹心饼干生产有限公司	夹心饼干联系	12345678888	夹心饼干
00009	曲奇生产有限公司	曲奇联系	12345679999	曲奇
00010	真果粒生产有限公司	真果粒联系	12345680000	真果粒

9.supply 表

```

274  -- -----
275  -- Table structure for supply
276  -- -----
277  -- CREATE TABLE IF NOT EXISTS supply(
278  --     supplier_number char(5) NOT NULL,
279  --     product_number char(10) NOT NULL,
280  --     supply_number SMALLINT NULL DEFAULT 0,
281  --     CHECK(supply_number>=0),
282  --     PRIMARY KEY(supplier_number,product_number)
283  -- )ENGINE=InnoDB;
284
285  -- -----
286  -- Records of supply
287  -- -----
288  -- INSERT INTO supply VALUES('00000','0000000000',1000),
289  -- ('00000','0000000001',1000),
290  -- ('00000','0000000002',1000),
291  -- ('00000','0000000003',1000),
292  -- ('00000','0000000004',1000),
293  -- ('00000','0000000005',1000),
294  -- ('00000','0000000006',1000),
295  -- ('00000','0000000007',1000),
296  -- ('00000','0000000008',1000),
297  -- ('00000','0000000009',1000),
298  -- ('00001','0000000008',1000),
299  -- ('00002','0000000007',1000),
300  -- ('00003','0000000006',1000),
301  -- ('00004','0000000005',1000),
302  -- ('00005','0000000004',1000),
303  -- ('00006','0000000003',1000),
304  -- ('00007','0000000002',1000),
305  -- ('00008','0000000001',1000),
306  -- ('00009','0000000000',1000),
307  -- ('00010','0000000009',1000);
308
309  -- SELECT * FROM supply;

```

supplier_number	product_number	supply_number
00000	0000000000	1000
00000	0000000001	1000
00000	0000000002	1000
00000	0000000003	1000
00000	0000000004	1000
00000	0000000005	1000
00000	0000000006	1000
00000	0000000007	1000
00000	0000000008	1000
00000	0000000009	1000
00001	0000000008	1000
00002	0000000007	1000
00003	0000000006	1000
00004	0000000005	1000
00005	0000000004	1000
00006	0000000003	1000
00007	0000000002	1000
00008	0000000001	1000
00009	0000000000	1000
00010	0000000009	1000

## 六、创建和删除数据库

1.创建 XXX 的数据库

2.删除 XXX 的数据库

DROP DATABASE XXX 的数据库;

## 七、创建和删除基本表

1.user 表



```

-- -----
-- Table structure for user
-- -----
-- CREATE TABLE IF NOT EXISTS user(
--     user_name VARCHAR(30) NOT NULL,
--     identity_card CHAR(18) NOT NULL,
--     user_password char(6) NOT NULL,
--     telephone char(11) NOT NULL,
--     email VARCHAR(60) DEFAULT NULL,
--     user_address VARCHAR(60) NOT NULL,
--     PRIMARY KEY(user_name)
-- )ENGINE=InnoDB;

```

## 2.product 表

```

-- -----
-- Table structure for product
-- -----
-- CREATE TABLE IF NOT EXISTS product(
--     product_number char(10) NOT NULL,
--     product_name VARCHAR(30) NOT NULL,
--     product_description VARCHAR(60) DEFAULT NULL,
--     product_price TINYINT NOT NULL,
--     PRIMARY KEY(product_name)
-- )ENGINE=InnoDB;

```

## 3.order\_table 表

```

-- -----
-- Table structure for order_table
-- -----
-- CREATE TABLE IF NOT EXISTS order_table(
--     order_number char(10) NOT NULL,
--     order_state boolean DEFAULT 0 ,
--     PRIMARY KEY(order_number)
-- )ENGINE=InnoDB;

```

## 4.establish 表

```

-- -----
-- Table structure for establish
-- -----
-- CREATE TABLE IF NOT EXISTS establish(
--     order_number char(10) NOT NULL,
--     user_name VARCHAR(30) NOT NULL,
--     PRIMARY KEY(order_number),
--     CONSTRAINT fk_establish_user_name
--     FOREIGN KEY(user_name) REFERENCES user(user_name)
-- )ENGINE=InnoDB;

```

## 5.order\_include 表

```
-- -----
-- Table structure for order_include
-- -----
-- CREATE TABLE IF NOT EXISTS order_include(
--     order_number char(10) NOT NULL,
--     product_number char(10) NOT NULL,
--     quantity TINYINT NULL DEFAULT 0,
--     PRIMARY KEY(order_number,product_number)
-- )ENGINE=InnoDB;
```

#### 6.shopping\_cart 表

```
-- -----
-- Table structure for shopping_cart
-- -----
-- CREATE TABLE IF NOT EXISTS shopping_cart(
--     shopping_cart_number char(10) NOT NULL,
--     user_name VARCHAR(30) NOT NULL,
--     PRIMARY KEY(shopping_cart_number),
--     CONSTRAINT fk_shopping_cart_user_name
--     FOREIGN KEY(user_name) REFERENCES user(user_name)

-- )ENGINE=InnoDB;
```

#### 7.shopping\_cart\_include 表

```
-- -----
-- Table structure for shopping_cart_include
-- -----
-- CREATE TABLE IF NOT EXISTS shopping_cart_include(
--     shopping_cart_number char(10) NOT NULL,
--     product_number char(10) NOT NULL,
--     join_date DATETIME NOT NULL,
--     quantity_of_product TINYINT NULL DEFAULT 0,
--     PRIMARY KEY(shopping_cart_number,product_number)
-- )ENGINE=InnoDB;
```

#### 8.supplier 表

```
-- -----
-- Table structure for supplier
-- -----
-- CREATE TABLE IF NOT EXISTS supplier(
--     supplier_number char(5) NOT NULL,
--     supplier_name varchar(30) NOT NULL,
--     supplier_linkman varchar(30) NULL,
--     supplier_telephone char(11) NOT NULL,
--     supplier_goods varchar(30) NOT NULL,
--     PRIMARY KEY(supplier_number)
-- )ENGINE=InnoDB;
```

## 9.supply 表

```
-- -----  
-- Table structure for supply  
-- -----  
-- CREATE TABLE IF NOT EXISTS supply(  
--     supplier_number char(5) NOT NULL,  
--     product_number char(10) NOT NULL,  
--     supply_number SMALLINT NULL DEFAULT 0,  
--     CHECK(supply_number>=0),  
--     PRIMARY KEY(supplier_number,product_number)  
-- )ENGINE=InnoDB;
```

## 八、查询

### 1.选择表中的若干列

例一：

```
SELECT user_name,user_password  
FROM user;
```

user_name	user_password
user1	111111
user10	000000
user2	222222
user3	333333
user4	444444
user5	555555
user6	666666
user7	777777
user8	888888
user9	999999

例二：

```
SELECT *  
FROM establish;
```

order_number	user_name
0000000000	user1
0000000001	user1
0000000010	user10
0000000002	user2
0000000011	user2
0000000003	user3
0000000004	user4
0000000005	user5
0000000006	user6
0000000007	user7
0000000008	user8
0000000009	user9

## 2.选择表中的若干元组

例一：

```
1  SELECT *
2  FROM product
3  WHERE product_number='0000000002';
```

product_number	product_name	product_description	product_price
0000000002	棒棒糖	混合口味的每袋是5个口味混合	26

例二：

```
SELECT *
FROM establish
WHERE order_number='0000000002';
```

order_number	user_name
0000000002	user2

## 3.录入表数据

例一：

```
INSERT INTO product VALUES('0000000000','曲奇','脆就一个字，我只说一次',30),
('0000000001','夹心饼干','',5),
('0000000002','棒棒糖','混合口味的每袋是5个口味混合',26),
('0000000003','大白兔奶糖','入口即化',20),
('0000000004','乐事薯片','干脆可口',10),
('0000000005','可口可乐','一箱24罐，冰冻后饮用口感更佳',50),
('0000000006','红牛','一箱24罐，激发你的能量',125),
('0000000007','椰树牌椰汁','植物蛋白饮料',85),
('0000000008','豆本豆奶','早餐奶低脂低糖原味豆奶250ml*24',56),
('0000000009','真果粒','牛奶+果粒，自然有乐趣',60);
```

product_number	product_name	product_description	product_price
0000000004	乐事薯片	干脆可口	10
0000000005	可口可乐	一箱24罐，冰冻后饮用口感更佳	50
0000000003	大白兔奶糖	入口即化	20
0000000001	夹心饼干		5
0000000000	曲奇	脆就一个字，我只说一次	30
0000000002	棒棒糖	混合口味的每袋是5个口味混合	26
0000000007	椰树牌椰汁	植物蛋白饮料	85
0000000009	真果粒	牛奶+果粒，自然有乐趣	60
0000000006	红牛	一箱24罐，激发你的能量	125
0000000008	豆本豆奶	早餐奶低脂低糖原味豆奶250ml*24	56

例二：

```
-- INSERT INTO supplier VALUES('00000','全部生产有限公司','全部联系','12345670000','全部'),
-- ('00001','豆奶生产有限公司','豆奶联系','12345671111','豆奶'),
-- ('00002','椰汁生产有限公司','椰汁联系','12345672222','椰汁'),
-- ('00003','红牛生产有限公司','红牛联系','12345673333','红牛'),
-- ('00004','可口可乐生产有限公司','可口可乐联系','12345674444','可口可乐'),
-- ('00005','薯片生产有限公司','薯片联系','12345675555','薯片'),
-- ('00006','大白兔奶糖生产有限公司','大白兔奶糖联系','12345676666','大白兔奶糖'),
-- ('00007','棒棒糖生产有限公司','棒棒糖联系','12345677777','棒棒糖'),
-- ('00008','夹心饼干生产有限公司','夹心饼干联系','12345678888','夹心饼干'),
-- ('00009','曲奇生产有限公司','曲奇联系','12345679999','曲奇'),
-- ('00010','真果粒生产有限公司','真果粒联系','12345680000','真果粒');
```

supplier_number	supplier_name	supplier_linkman	supplier_telephone	supplier_goods
00000	全部生产有限公司	全部联系	12345670000	全部
00001	豆奶生产有限公司	豆奶联系	12345671111	豆奶
00002	椰汁生产有限公司	椰汁联系	12345672222	椰汁
00003	红牛生产有限公司	红牛联系	12345673333	红牛
00004	可口可乐生产有限公司	可口可乐联系	12345674444	可口可乐
00005	薯片生产有限公司	薯片联系	12345675555	薯片
00006	大白兔奶糖生产有限公司	大白兔奶糖联系	12345676666	大白兔奶糖
00007	棒棒糖生产有限公司	棒棒糖联系	12345677777	棒棒糖
00008	夹心饼干生产有限公司	夹心饼干联系	12345678888	夹心饼干
00009	曲奇生产有限公司	曲奇联系	12345679999	曲奇
00010	真果粒生产有限公司	真果粒联系	12345680000	真果粒

#### 4.使用计算函数汇总数据

例一：

```
SELECT SUM(total_price)
FROM user1;
```

SUM(total\_price)

218

例二：

```
SELECT COUNT(*)  
FROM establish;
```

COUNT(\*)

12

5.对查询结果进行分组计算

例一：

```
-- SELECT user.user_name,telephone,email,COUNT(establish.order_number)AS order_count,SUM(product.product_price*order_include.quantity) AS all_total_price  
-- FROM user  
-- INNER JOIN establish ON user.user_name=establish.user_name  
-- INNER JOIN order_table ON establish.order_number=order_table.order_number  
-- INNER JOIN order_include ON order_table.order_number=order_include.order_number  
-- INNER JOIN product ON order_include.product_number=product.product_number  
-- GROUP BY user.user_name  
-- ORDER BY establish.order_number,all_total_price;
```

user_name	telephone	email	order_count	all_total_price
user1	11111111111	1111@qq.com	6	218
user2	22222222222	2222@qq.com	4	301
user3	33333333333	3333@qq.com	3	245
user4	44444444444	4444@qq.com	2	307
user5	55555555555	5555@qq.com	3	476
user6	66666666666	6666@qq.com	2	20
user7	77777777777	7777@qq.com	2	57
user8	88888888888	8888@qq.com	3	346
user9	99999999999	9999@qq.com	3	400
user10	00000000000	0000@qq.com	2	276

例二：

```
SELECT COUNT(order_number)  
FROM establish  
GROUP BY user_name;
```

COUNT(order_number)
2
1
2
1
1
1
1
1
1
1
1

## 6.多表连接查询

例一：

```
-- SELECT user.user_name,telephone,shopping_cart.shopping_cart_number,product_name
-- FROM user
-- INNER JOIN shopping_cart ON user.user_name=shopping_cart.user_name
-- INNER JOIN shopping_cart_include ON shopping_cart_include.shopping_cart_number=shopping_cart.shopping_cart_number
-- INNER JOIN product ON product.product_number=shopping_cart_include.product_number
-- ORDER BY shopping_cart.shopping_cart_number;
```

---

user_name	telephone	shopping_cart_number	product_name
user1	11111111111	0000000001	夹心饼干
user1	11111111111	0000000001	棒棒糖
user1	11111111111	0000000001	大白兔奶糖
user2	22222222222	0000000002	夹心饼干
user3	33333333333	0000000003	可口可乐
user3	33333333333	0000000003	红牛
user3	33333333333	0000000003	乐事薯片
user4	44444444444	0000000004	大白兔奶糖
user4	44444444444	0000000004	可口可乐
user4	44444444444	0000000004	椰树牌椰汁
user5	55555555555	0000000005	曲奇
user5	55555555555	0000000005	椰树牌椰汁
user5	55555555555	0000000005	豆本豆奶
user6	66666666666	0000000006	椰树牌椰汁
user6	66666666666	0000000006	豆本豆奶
user6	66666666666	0000000006	真果粒
user7	77777777777	0000000007	曲奇
user7	77777777777	0000000007	棒棒糖
user7	77777777777	0000000007	大白兔奶糖
user8	88888888888	0000000008	红牛
user8	88888888888	0000000008	夹心饼干
user8	88888888888	0000000008	大白兔奶糖
user9	99999999999	0000000009	大白兔奶糖
user9	99999999999	0000000009	乐事薯片
user9	99999999999	0000000009	真果粒
user10	00000000000	0000000010	棒棒糖
user10	00000000000	0000000010	红牛
user10	00000000000	0000000010	椰树牌椰汁

例二：

```
SELECT supplier_name,supplier_goods,product.product_number,product.product_name
FROM supplier,supply,product
WHERE supplier.supplier_number=supply.supplier_number
AND supply.product_number=product.product_number;
```



supplier_name	supplier_goods	product_number	product_name
全部生产有限公司	全部	0000000000	曲奇
全部生产有限公司	全部	0000000001	夹心饼干
全部生产有限公司	全部	0000000002	棒棒糖
全部生产有限公司	全部	0000000003	大白兔奶糖
全部生产有限公司	全部	0000000004	乐事薯片
全部生产有限公司	全部	0000000005	可口可乐
全部生产有限公司	全部	0000000006	红牛
全部生产有限公司	全部	0000000007	椰树牌椰汁
全部生产有限公司	全部	0000000008	豆本豆奶
全部生产有限公司	全部	0000000009	真果粒
豆奶生产有限公司	豆奶	0000000008	豆本豆奶
椰汁生产有限公司	椰汁	0000000007	椰树牌椰汁
红牛生产有限公司	红牛	0000000006	红牛
可口可乐生产有限公司	可口可乐	0000000005	可口可乐
薯片生产有限公司	薯片	0000000004	乐事薯片
大白兔奶糖生产有限公司	大白兔奶糖	0000000003	大白兔奶糖
棒棒糖生产有限公司	棒棒糖	0000000002	棒棒糖
夹心饼干生产有限公司	夹心饼干	0000000001	夹心饼干
曲奇生产有限公司	曲奇	0000000000	曲奇
真果粒生产有限公司	真果粒	0000000009	真果粒

## 九、数据增删改

### 1.向订单表中新增一条用户订单信息

```
-- INSERT INTO establish VALUES('0000000012','user1');
-- INSERT INTO order_table VALUES('0000000012',0);
-- SELECT * FROM establish;
-- SELECT * FROM order_table;
```

order_number	user_name
0000000000	user1
0000000001	user1
0000000012	user1
0000000010	user10
0000000002	user2
0000000011	user2
0000000003	user3
0000000004	user4
0000000005	user5
0000000006	user6
0000000007	user7
0000000008	user8
0000000009	user9

order_number	order_state
0000000000	0
0000000001	0
0000000002	0
0000000003	0
0000000004	0
0000000005	0
0000000006	0
0000000007	0
0000000008	0
0000000009	0
0000000010	0
0000000011	0
0000000012	0

2.修改某个用户的某个订单的所选商品信息

```

UPDATE order_include
SET quantity=5
WHERE order_number='0000000000' AND product_number='0000000001';
SELECT order_number,product_number,quantity
FROM order_include
WHERE order_number='0000000000' AND product_number='0000000001';

```

order_number	product_number	quantity
0000000000	0000000001	5

### 3.删除某个用户的某条订单信息

有条件删除

```

-- DELETE FROM establish
-- WHERE order_number='0000000012';
-- DELETE FROM order_table
-- WHERE order_number='0000000012';
-- SELECT * FROM establish;
-- SELECT * FROM order_table;

```

order_number	user_name
0000000000	user1
0000000001	user1
0000000010	user10
0000000002	user2
0000000011	user2
0000000003	user3
0000000004	user4
0000000005	user5
0000000006	user6
0000000007	user7
0000000008	user8
0000000009	user9

order_number	order_state
0000000000	0
0000000001	0
0000000002	0
0000000003	0
0000000004	0
0000000005	0
0000000006	0
0000000007	0
0000000008	0
0000000009	0
0000000010	0
0000000011	0

无条件删除 establish 表的内容

```
DELETE FROM establish;  
SELECT * FROM establish;
```

No data