Assignment 2

matriculation number: 200009834

Task 1: Translation

First analyze the customer. Since the customer has multiple phone numbers, another table needs to be created. The conversion is as follows:

Customer(customer_id, name, email, street, city, postcode, country)
Entity set: Customer
Primary key: customer_id
Multivalued attributes: phone
Composite attribute: address
General attribute: name

Customer_phone(customer_id, phone_type, phone_number)
Entity set: Customer_phone
Primary key: customer_id
Foreign key: customer_id
General attribute: phone_type, phone_number

Next, look at the order table on the left side of the ER diagram. The order needs to establish a relationship with the customer through Places, so my conversion is as follows:

Order(order_id, delivery_street, delivery_city, delivery_postcode, delivery_country, date_ordered, date_delivered)
Entity set: Order
Primary key: order_id
Composite attribute: delivery_address
General attribute: date_ordered, date_delivered

Order_Places(order_id, customer_id)
Entity set: Order_Places
Primary keys: order_id, customer_id
Foreign keys: order_id, customer_id

Immediately afterwards, continue to observe the right side of the ER diagram with the customer as the center. It is the book table. There are genre in which another table needs to be created because it is a multi-valued attribute. The details are as follows:

Book(book_id, title, author, publisher)
Entity set: Book
Primary key: book_id
Multivalued attributes: genre
General attribute: title, author, publisher

Book_genre(book_id, genre)
Entity set: Book_genre
Primary key: book_id
Foreign key: book_id
General key: genre

Then, the relationship between the customer and the book is reviews, so the constructed relationship is as follows:

Reviews(book_id, customer_id, rating)
Entity set: Reviews
Primary keys: book_id, customer_id
Foreign keys: book_id, customer_id
General attribute: rating

Next, observe the lower right part of the ER diagram, the first is the Edition, because, Edition is total participation of book. The relationship of construction is as follows:

Edition(book_id, edition, type, price, quantity_in_stock)
Entity set: Edition
Primary keys: book_id, edition, type
Foreign keys: book_id, edition
General attribute: price, quantity_in_stock

Contains(book_id, order_id, edition, type)
Entity set: Contains
Primary keys: book_id, order_id
Foreign keys: book_id, order_id
General attributes: edition, type

The last part is the supplier part, the relationship is as follows:

Supplier(supplier_id, name, account_no)
Entity set: Supplier
Primary key: supplier_id
Multivalued attributes: phone
General attributes: name, account_no

Supplier_phone(supplier_id, phone_number)
Entity set: Supplier
Primary key: supplier_id, phone_number
Foreign key: supplier_id

Supplies(supplier_id, book_id, supply_price)
Entity set: Supplies
Primary keys: supplier_id, book_id
Foreign keys: supplier_id, book_id
General attribute: supply_price

Task 2: SQL Data Definition

```sql
CREATE TABLE Customer (
customer_id CHAR(8)PRIMARY KEY,
name VARCHAR(20)NOT NULL,
email VARCHAR(30)NOT NULL,
street VARCHAR(20)NOT NULL,
city VARCHAR(20)NOT NULL,
postcode VARCHAR(10)NOT NULL,
country VARCHAR(10)NOT NULL);

CREATE TABLE Customer_phone (
customer_id CHAR(8),
customer_phone_type VARCHAR(20)NOT NULL,
customer_phone_number CHAR (13),
PRIMARY KEY(customer_id, customer_phone_number),
FOREIGN KEY(customer_id) REFERENCES Customer(customer_id)
);
```

```sql
INSERT INTO Customer VALUES
('00000001', 'Robert', 'robert@mail.com', '11 random street', 'Edinburgh', 'KY1
('00000002', 'Jack', 'jack@mail.com', '22 king street', 'Edinburgh', 'EB16 9XW'
('00000003', 'Bob', 'bob@mail.com', '33 happy street', 'London', 'LD1A 1EE', 'U
('00000004', 'Quentin', 'quentin@mail.com', '44 west street', 'Cardiff', 'CD23
('00000005', 'Leon', 'Leon@mail.com', '55 east street', 'Glasgow', 'GG23 DRE',
('00000006', 'Peter', 'peter@mail.com', '66 spider street', 'Manchester', 'MC66
('00000007', 'Tony', 'tony@mail.com', '77 iron street', 'Dundee', 'DD78 9XW', '
('00000008', 'Anna', 'anna@mail.com', '88 peace street', 'Southampton', 'SA89 8
('00000009', 'Ci', 'ci@mail.com', '99 universe street', 'Liverpool', 'LP09 78W
('00000010', 'Joe', 'joe@mail.com', '10 final street', 'Bristol', 'BT10 99X',

INSERT INTO Customer_phone VALUES
('00000001', 'mobile phone', '078 1111 6785'),
('00000002', 'telephone', '078 2222 6785'),
('00000003', 'telephone', '078 3333 6785'),
('00000004', 'mobile phone', '078 4444 6785'),
```

	customer_id	name	email	street	city	postcode	country
	过滤	过滤	过滤	过滤	过滤	过滤	过滤
1	00000001	Robert	robert@mail.com	11 random street	Edinburgh	KY16 9XW	UK
2	00000002	Jack	jack@mail.com	22 king street	Edinburgh	EB16 9XW	UK
3	00000003	Bob	bob@mail.com	33 happy street	London	LD1A 1EE	UK
4	00000004	Quentin	quentin@mail.com	44 west street	Cardiff	CD23 2DF	UK
5	00000005	Leon	Leon@mail.com	55 east street	Glasgow	GG23 DRE	UK
6	00000006	Peter	peter@mail.com	66 spider street	Manchester	MC66 89W	UK
7	00000007	Tony	tony@mail.com	77 iron street	Dundee	DD78 9XW	UK
8	00000008	Anna	anna@mail.com	88 peace street	Southampton	SA89 8TR	UK
9	00000009	Ci	ci@mail.com	99 universe street	Liverpool	LP09 78W	UK
10	00000010	Joe	joe@mail.com	10 final street	Bristol	BT10 99X	UK

```sql
CREATE TABLE Customer_phone (
customer_id CHAR(8),
customer_phone_type VARCHAR(20)NOT NULL,
customer_phone_number CHAR (13),
PRIMARY KEY(customer_id, customer_phone_number),
FOREIGN KEY(customer_id) REFERENCES Customer(customer_id)
);

CREATE TABLE Order_ (
order_id CHAR(10)PRIMARY KEY,
delivery_street VARCHAR(20)NOT NULL,
delivery_city VARCHAR(20)NOT NULL,
```

```sql
INSERT INTO Customer_phone VALUES
('00000001', 'mobile phone', '078 1111 6785'),
('00000002', 'telephone', '078 2222 6785'),
('00000003', 'telephone', '078 3333 6785'),
('00000004', 'mobile phone', '078 4444 6785'),
('00000005', 'telephone', '078 5555 6785'),
('00000006', 'telephone', '078 6666 6785'),
('00000007', 'mobile phone', '078 7777 6785'),
('00000008', 'telephone', '078 8888 6785'),
('00000009', 'telephone', '078 9999 6785'),
('00000010', 'mobile phone', '078 1010 6785');
```

	customer_id	customer_phone_type	customer_phone_number
	过滤	过滤	过滤
1	00000001	mobile phone	078 1111 6785
2	00000002	telephone	078 2222 6785
3	00000003	telephone	078 3333 6785
4	00000004	mobile phone	078 4444 6785
5	00000005	telephone	078 5555 6785
6	00000006	telephone	078 6666 6785
7	00000007	mobile phone	078 7777 6785
8	00000008	telephone	078 8888 6785
9	00000009	telephone	078 9999 6785
10	00000010	mobile phone	078 1010 6785

```
CREATE TABLE Order_ (
order_id CHAR(10)PRIMARY KEY,
delivery_street VARCHAR(20)NOT NULL,
delivery_city VARCHAR(20)NOT NULL,
delivery_postcode VARCHAR(10)NOT NULL,
delivery_country VARCHAR(10)NOT NULL,
date_ordered date,
date_delivered date);

CREATE TABLE Order_Places (
order_id CHAR(10),
customer_id CHAR(8),
PRIMARY KEY(order_id, customer_id),
```

```
INSERT INTO Order_ VALUES
('1234567891', '111 start street', 'Edinburgh', 'ED1', 'UK', 20_130_101, 20_130
('1234567892', '222 crown street', 'Edinburgh', 'ED2', 'UK', 20_140_202, 20_140
('1234567893', '333 park street', 'Manchester', 'MC3', 'UK', 20_150_303, 20_150
('1234567894', '444 Roman street', 'Bristol', 'BT4', 'UK', 20_160_404, 20_160_4
('1234567895', '555 King road', 'Southampton', 'SA5', 'UK', 20_170_505, 20_170_
('1234567896', '666 west road', 'Dundee', 'DD6', 'UK', 20_180_606, 20_180_607),
('1234567897', '777 east road', 'Glasgow', 'GG7', 'UK', 20_190_707, 20_190_708)
('1234567898', '888 north road', 'Edinburgh', 'ED8', 'UK', 20_100_808, 20_100_8
('1234567899', '999 south road', 'London', 'LD9', 'UK', 20_110_909, 20_110_910)
('1234567810', '100 final road', 'Edinburgh', 'ED10', 'UK', 20_201_010, 20_201_
('1234567811', '101 another final road', 'Edinburgh', 'ED11', 'UK', 20_201_111,
```

	order_id	delivery_street	delivery_city	delivery_postcode	delivery_country	date_ordered	date_delivered
	过滤	过滤	过滤	过滤	过滤	过滤	过滤
1	1234567891	111 start street	Edinburgh	ED1	UK	20130101	20130102
2	1234567892	222 crown street	Edinburgh	ED2	UK	20140202	20140203
3	1234567893	333 park street	Manchester	MC3	UK	20150303	20150304
4	1234567894	444 Roman street	Bristol	BT4	UK	20160404	20160405
5	1234567895	555 King road	Southampton	SA5	UK	20170505	20170506
6	1234567896	666 west road	Dundee	DD6	UK	20180606	20180607
7	1234567897	777 east road	Glasgow	GG7	UK	20190707	20190708
8	1234567898	888 north road	Edinburgh	ED8	UK	20100808	20100809
9	1234567899	999 south road	London	LD9	UK	20110909	20110910
10	1234567810	100 final road	Edinburgh	ED10	UK	20201010	20201011
11	1234567811	101 another final road	Edinburgh	ED11	UK	20201111	20201112

```
CREATE TABLE Order_Places (
order_id CHAR(10),
customer_id CHAR(8),
PRIMARY KEY(order_id, customer_id),
FOREIGN KEY(order_id) REFERENCES Order_(order_id),
FOREIGN KEY(customer_id) REFERENCES Customer(customer_id)
);

CREATE TABLE Book (
book_id CHAR(5)PRIMARY KEY,
```

```
INSERT INTO Order_Places VALUES
('1234567891', '00000002'),
('1234567892', '00000004'),
('1234567893', '00000006'),
('1234567894', '00000008'),
('1234567895', '00000010'),
('1234567896', '00000009'),
('1234567897', '00000007'),
('1234567898', '00000005'),
('1234567899', '00000003'),
('1234567810', '00000001');
```

	order_id	customer_id
	过滤	过滤
1	1234567891	00000002
2	1234567892	00000004
3	1234567893	00000006
4	1234567894	00000008
5	1234567895	00000010
6	1234567896	00000009
7	1234567897	00000007
8	1234567898	00000005
9	1234567899	00000003
10	1234567810	00000001

```
CREATE TABLE Book (
book_id CHAR(5)PRIMARY KEY,
title VARCHAR(30)NOT NULL,
author VARCHAR(30)NOT NULL,
publisher VARCHAR(30)NOT NULL
);

CREATE TABLE Book_genre (
book_id CHAR(5)PRIMARY KEY,
genre VARCHAR(20)NOT NULL,
FOREIGN KEY(book_id) REFERENCES Book(book_id)
```

```
INSERT INTO Book VALUES
('00001', 'Harry Potter', 'J K Rowling', 'Ultimate Books'),
('00002', 'Les Enfants du capitaine Grant', 'Jules Gabriel Verne', 'Ultimate Bo
('00003', 'The Hunger Games', 'Suzanne Collins ', 'Tom media'),
('00004', 'Brave New World', 'Aldous Huxley', 'Recorded Books'),
('00005', 'Angels & Demons', 'Dan Brown', 'HighBridge Audio'),
('00006', 'The Lord of the Rings', 'J. R. R. Tolkien', 'HighBridge Audio'),
('00007', 'The Da Vinci Code', 'Dan Brown', 'Solis Press'),
('00008', 'I, Robot', 'Isaac Asimov', 'Ultimate Books'),
('00009', 'This is life', 'Elliot K Kirby', 'RealLifeBooks'),
('00010', 'The Foundation Series', 'Isaac Asimov', 'Ultimate Books');
```

	book_id	title	author	publisher
	过滤	过滤	过滤	过滤
1	00001	Harry Potter	J K Rowling	Ultimate Books
2	00002	Les Enfants du capitaine Grant	Jules Gabriel Verne	Ultimate Books
3	00003	The Hunger Games	Suzanne Collins	Tom media
4	00004	Brave New World	Aldous Huxley	Recorded Books
5	00005	Angels & Demons	Dan Brown	HighBridge Audio
6	00006	The Lord of the Rings	J. R. R. Tolkien	HighBridge Audio
7	00007	The Da Vinci Code	Dan Brown	Solis Press
8	00008	I, Robot	Isaac Asimov	Ultimate Books
9	00009	This is life	Elliot K Kirby	RealLifeBooks
10	00010	The Foundation Series	Isaac Asimov	Ultimate Books

```
CREATE TABLE Book_genre (
book_id CHAR(5)PRIMARY KEY,
genre VARCHAR(20)NOT NULL,
FOREIGN KEY(book_id) REFERENCES Book(book_id)
);

CREATE TABLE Reviews (
book_id CHAR(5),
customer_id CHAR(8),
rating CHAR(1),
```

```
INSERT INTO Book_genre VALUES
('00001', 'fiction'),
('00002', 'Science and Technology'),
('00003', 'adventure'),
('00004', 'classic'),
('00005', 'adventure'),
('00006', 'fiction'),
('00007', 'adventure'),
('00008', 'Science and Technology'),
('00009', 'classic'),
('00010', 'Science and Technology');
```

	book_id	genre
	过滤	过滤
1	00001	fiction
2	00002	Science and Technology
3	00003	adventure
4	00004	classic
5	00005	adventure
6	00006	fiction
7	00007	adventure
8	00008	Science and Technology
9	00009	classic
10	00010	Science and Technology

```
CREATE TABLE Reviews (
  book_id CHAR(5),
  customer_id CHAR(8),
  rating CHAR(1),
  PRIMARY KEY(book_id, customer_id),
  FOREIGN KEY(book_id) REFERENCES Book(book_id),
  FOREIGN KEY(customer_id) REFERENCES Customer(customer_id)
);

CREATE TABLE Edition (
```

```
INSERT INTO Reviews VALUES
('00001', '00000001', '9'),
('00002', '00000002', '8'),
('00003', '00000003', '7'),
('00004', '00000004', '6'),
('00005', '00000005', '5'),
('00006', '00000006', '4'),
('00007', '00000007', '4'),
('00008', '00000008', '3'),
('00009', '00000009', '1'),
('00010', '00000010', '1');
```

	book_id	customer_id	rating
	过滤	过滤	过滤
1	00001	00000001	9
2	00002	00000002	8
3	00003	00000003	7
4	00004	00000004	6
5	00005	00000005	5
6	00006	00000006	4
7	00007	00000007	4
8	00008	00000008	3
9	00009	00000009	1
10	00010	00000010	1

```sql
CREATE TABLE Edition (
book_id CHAR(5),
book_edition VARCHAR(5),
type VARCHAR(15)NOT NULL,
price NUMERIC(5,2)NOT NULL,
quantity_in_stock VARCHAR(5)NOT NULL,
PRIMARY KEY(book_id, book_edition, type),
FOREIGN KEY(book_id) REFERENCES Book(book_id)
);

CREATE TABLE Contains_ (
book_id CHAR(5),
order_id CHAR(10),
book_edition VARCHAR(5)NOT NULL,
type VARCHAR(15)NOT NULL,
PRIMARY KEY(order_id, book_id, book_edition, type),
```

```sql
INSERT INTO Edition VALUES
('00001', '8', 'audiobook', '010.10', '100'),
('00001', '4', 'hardcover', '010.10', '1'),
('00001', '5', 'paperback', '010.10', '2'),
('00001', '6', 'audiobook', '010.10', '3'),
('00002', '7', 'hardcover', '020.20', '200'),
('00003', '6', 'audiobook', '030.30', '300'),
('00004', '5', 'hardcover', '040.40', '400'),
('00005', '4', 'paperback', '050.50', '500'),
('00006', '3', 'audiobook', '060.60', '600'),
('00007', '2', 'paperback', '070.70', '700'),
('00007', '3', 'paperback', '070.70', '4'),
('00007', '4', 'audiobook', '070.70', '3'),
('00007', '6', 'hardcover', '070.70', '2'),
('00008', '1', 'paperback', '080.80', '800'),
('00009', '9', 'audiobook', '090.90', '900'),
('00010', '5', 'hardcover', '100.10', '1000');
```

	book_id	book_edition	type	price	quantity_in_stock
	过滤	过滤	过滤	过滤	过滤
1	00001	8	audiobook	10.1	100
2	00001	4	hardcover	10.1	1
3	00001	5	paperback	10.1	2
4	00001	6	audiobook	10.1	3
5	00002	7	hardcover	20.2	200
6	00003	6	audiobook	30.3	300
7	00004	5	hardcover	40.4	400
8	00005	4	paperback	50.5	500
9	00006	3	audiobook	60.6	600
10	00007	2	paperback	70.7	700
11	00007	3	paperback	70.7	4
12	00007	4	audiobook	70.7	3
13	00007	6	hardcover	70.7	2
14	00008	1	paperback	80.8	800
15	00009	9	audiobook	90.9	900
16	00010	5	hardcover	100.1	1000

```sql
);

CREATE TABLE Contains_ (
book_id CHAR(5),
order_id CHAR(10),
book_edition VARCHAR(5)NOT NULL,
type VARCHAR(15)NOT NULL,
PRIMARY KEY(order_id, book_id, book_edition, type),
FOREIGN KEY(order_id) REFERENCES Order_(order_id),
FOREIGN KEY(book_id) REFERENCES Book(book_id)
);
```

```sql
INSERT INTO Contains_ VALUES
('00001', '1234567891', '8', 'audiobook'),
('00002', '1234567892', '7', 'hardcover'),
('00003', '1234567893', '6', 'audiobook'),
('00004', '1234567894', '5', 'hardcover'),
('00005', '1234567895', '4', 'paperback'),
('00006', '1234567896', '3', 'audiobook'),
('00007', '1234567897', '2', 'paperback'),
('00008', '1234567898', '1', 'paperback'),
('00009', '1234567899', '9', 'audiobook'),
('00010', '1234567810', '5', 'hardcover');
```

	book_id	order_id	book_edition	type
	过滤	过滤	过滤	过滤
1	00001	1234567891	8	audiobook
2	00002	1234567892	7	hardcover
3	00003	1234567893	6	audiobook
4	00004	1234567894	5	hardcover
5	00005	1234567895	4	paperback
6	00006	1234567896	3	audiobook
7	00007	1234567897	2	paperback
8	00008	1234567898	1	paperback
9	00009	1234567899	9	audiobook
10	00010	1234567810	5	hardcover

```sql
CREATE TABLE Supplier (
supplier_id CHAR(5)PRIMARY KEY,
name VARCHAR(30)NOT NULL,
account_no CHAR(10)NOT NULL
);

CREATE TABLE Supplier_phone (
supplier_id CHAR(5),
supplier_phone_number CHAR(13),
PRIMARY KEY(supplier_id, supplier_phone_number),
FOREIGN KEY(supplier_id) REFERENCES Supplier(supplier_id)
);
```

```sql
INSERT INTO Supplier VALUES
('11111', 'The Worker', '1111100000'),
('22222', 'The Student', '2222200000'),
('33333', 'The Book', '3333300000'),
('44444', 'The Teacher', '4444400000'),
('55555', 'The Learner', '5555500000'),
('66666', 'The Listener', '6666600000'),
('77777', 'The Supplier', '7777700000'),
('88888', 'The One', '8888800000'),
('99999', 'The What?', '9999900000'),
('10101', 'The NeXt', '1010100000');
```

	supplier_id	name	account_no
	过滤	过滤	过滤
1	11111	The Worker	1111100000
2	22222	The Student	2222200000
3	33333	The Book	3333300000
4	44444	The Teacher	4444400000
5	55555	The Learner	5555500000
6	66666	The Listener	6666600000
7	77777	The Supplier	7777700000
8	88888	The One	8888800000
9	99999	The What?	9999900000
10	10101	The NeXt	1010100000

```sql
CREATE TABLE Supplier_phone (
 supplier_id CHAR(5),
 supplier_phone_number CHAR(13),
 PRIMARY KEY(supplier_id, supplier_phone_number),
 FOREIGN KEY(supplier_id) REFERENCES Supplier(supplier_id)
);

CREATE TABLE Supplies (
 supplier_id CHAR(5),
 book_id CHAR(5),
 supply_price NUMERIC(5,2)NOT NULL,
 PRIMARY KEY(supplier_id, book_id),
```

```sql
INSERT INTO Supplier_phone VALUES
 ('11111', '088 0631 1111'),
 ('22222', '088 0631 2222'),
 ('33333', '088 0631 3333'),
 ('44444', '088 0631 4444'),
 ('55555', '088 0631 5555'),
 ('66666', '088 0631 6666'),
 ('77777', '088 0631 7777'),
 ('88888', '088 0631 8888'),
 ('99999', '088 0631 9999'),
 ('10101', '088 0631 1010');
```

	supplier_id	supplier_phone_number
	过滤	过滤
1	11111	088 0631 1111
2	22222	088 0631 2222
3	33333	088 0631 3333
4	44444	088 0631 4444
5	55555	088 0631 5555
6	66666	088 0631 6666
7	77777	088 0631 7777
8	88888	088 0631 8888
9	99999	088 0631 9999
10	10101	088 0631 1010

```sql
CREATE TABLE Supplies (
 supplier_id CHAR(5),
 book_id CHAR(5),
 supply_price NUMERIC(5,2)NOT NULL,
 PRIMARY KEY(supplier_id, book_id),
 FOREIGN KEY(supplier_id) REFERENCES Supplier(supplier_id),
 FOREIGN KEY(book_id) REFERENCES Book(book_id)
);

-------------------------------------------------------------
-- Insert data
-------------------------------------------------------------
/* All information is fake. Just for testing. */
```

```sql
INSERT INTO Supplies VALUES
 ('11111', '00001', '001.10'),
 ('22222', '00002', '002.20'),
 ('33333', '00003', '003.30'),
 ('44444', '00004', '004.40'),
 ('55555', '00005', '005.50'),
 ('66666', '00006', '006.60'),
 ('77777', '00007', '007.70'),
 ('88888', '00008', '008.80'),
 ('99999', '00009', '009.90'),
 ('10101', '00010', '010.10');
```

	supplier_id	book_id	supply_price
	过滤	过滤	过滤
1	11111	00001	1.1
2	22222	00002	2.2
3	33333	00003	3.3
4	44444	00004	4.4
5	55555	00005	5.5
6	66666	00006	6.6
7	77777	00007	7.7
8	88888	00008	8.8
9	99999	00009	9.9
10	10101	00010	10.1

Task3:

List all books published by “Ultimate Books” which are in the “Science and Technology” genre;

```
1   SELECT Book. book_id, title, author, publisher, genre
2   FROM Book, Book_genre
3   WHERE Book.book_id = Book_genre.book_id
4   AND publisher = 'Ultimate Books'
5   AND genre = 'Science and Technology';
```

	book_id	title	author	publisher	genre
1	00002	Les Enfants du capitaine Grant	Jules Gabriel Verne	Ultimate Books	Science and Technology
2	00008	I, Robot	Isaac Asimov	Ultimate Books	Science and Technology
3	00010	The Foundation Series	Isaac Asimov	Ultimate Books	Science and Technology

```
执行完成。
结果: 3 行返回，耗时 17ms
在行 1:
SELECT Book. book_id, title, author, publisher, genre
FROM Book, Book_genre
WHERE Book.book_id = Book_genre.book_id
AND publisher = 'Ultimate Books'
AND genre = 'Science and Technology';
```

List all orders placed by customers with addresses in the city of Edinburgh, since 2016, in order of date, latest first;

```
SELECT Order_. order_id, delivery_city, date_ordered
FROM Order_
WHERE Order_.delivery_city = 'Edinburgh'
AND Order_.date_ordered >= 20160000
ORDER BY date_ordered DESC;
```

	order_id	delivery_city	date_ordered
1	1234567811	Edinburgh	20201111
2	1234567810	Edinburgh	20201010

```
执行完成。
结果：2 行返回，耗时 7ms
在行 1:
SELECT Order_. order_id, delivery_city, date_ordered
FROM Order_
WHERE Order_.delivery_city = 'Edinburgh'
AND Order_.date_ordered >= 20160000
ORDER BY date_ordered DESC;
```

List all book editions which have less than 5 items in stock, together with the name, account number and supply price of the minimum priced supplier for that edition.

```
SELECT book_id, title, book_edition, quantity_in_stock, name, account_no, min(supply_price)
FROM Book NATURAL JOIN Edition NATURAL JOIN Supplier NATURAL JOIN Supplies
WHERE quantity_in_stock < 5
GROUP BY book_edition;
```

	book_id	title	book_edition	quantity_in_stock	name	account_no	min(supply_price)
1	00007	The Da Vinci Code	3	4	The Supplier	7777700000	7.7
2	00001	Harry Potter	4	1	The Worker	1111100000	1.1
3	00001	Harry Potter	5	2	The Worker	1111100000	1.1
4	00001	Harry Potter	6	3	The Worker	1111100000	1.1
5	00002	Les Enfants du capitaine Grant	7	200	The Student	2222200000	2.2
6	00001	Harry Potter	8	100	The Worker	1111100000	1.1

```
执行完成。
结果：6 行返回，耗时 9ms
在行 1:
SELECT book_id, title, book_edition, quantity_in_stock, name, account_no, min(supply_price)
FROM Book NATURAL JOIN Edition NATURAL JOIN Supplier NATURAL JOIN Supplies
WHERE quantity_in_stock < 5
GROUP BY book_edition;
```

List all books published more than 5th edition, together with the book's name, author and publisher.

```sql
SELECT Book. book_id, title, author, book_edition
FROM Book, Edition
WHERE Book.book_id = Edition.book_id
AND book_edition > 5;
```

	book_id	title	author	book_edition
1	00001	Harry Potter	J K Rowling	6
2	00001	Harry Potter	J K Rowling	8
3	00002	Les Enfants du capitaine Grant	Jules Gabriel Verne	7
4	00003	The Hunger Games	Suzanne Collins	6

```
执行完成。
结果: 6 行返回，耗时 8ms
在行 1:
SELECT Book. book_id, title, author, book_edition
FROM Book, Edition
WHERE Book.book_id = Edition.book_id
AND book_edition > 5;
```

List all books have at least 6 grade rating, recording the book's title, id and genre.

```sql
SELECT Book.book_id, title, author, rating
FROM Book, Reviews
WHERE Book.book_id = Reviews.book_id
AND rating >=6;
```

	book_id	title	author	rating
1	00001	Harry Potter	J K Rowling	9
2	00002	Les Enfants du capitaine Grant	Jules Gabriel Verne	8
3	00003	The Hunger Games	Suzanne Collins	7
4	00004	Brave New World	Aldous Huxley	6

```
执行完成。
结果: 4 行返回，耗时 9ms
在行 1:
SELECT Book.book_id, title, author, rating
FROM Book, Reviews
WHERE Book.book_id = Reviews.book_id
AND rating >=6;
```

List the customers who ordered after 2016, together with the book's title and the customers name and email.

```
SELECT Customer.name, email, title, author, date_ordered
FROM Customer NATURAL JOIN Book NATURAL JOIN Order_ NATURAL JOIN Contains_ NATURAL JOIN Order_Places
WHERE date_ordered <= 20160000;
```

	name	email	title	author	date_ordered
1	Jack	jack@mail.com	Harry Potter	J K Rowling	20130101
2	Quentin	quentin@mail.com	Les Enfants du capitaine Grant	Jules Gabriel Verne	20140202
3	Peter	peter@mail.com	The Hunger Games	Suzanne Collins	20150303
4	Leon	Leon@mail.com	I, Robot	Isaac Asimov	20100808
5	Bob	bob@mail.com	This is life	Elliot K Kirby	20110909

```
执行完成。
结果：5 行返回，耗时 0ms
在行 1:
SELECT Customer.name, email, title, author, date_ordered
```

A customer wants to find basic information about all Science and Technology books.

```
CREATE VIEW Science_and_Technology_Book AS
SELECT genre, title, author, publisher, price
FROM Book NATURAL JOIN Edition NATURAL JOIN Book_genre
WHERE Book.book_id = Edition.book_id
AND genre = 'Science and Technology';
```

	genre	title	author	publisher	price
	过滤	过滤	过滤	过滤	过滤
1	Science and Technology	Les Enfants du capitaine Grant	Jules Gabriel Verne	Ultimate Books	20.2
2	Science and Technology	I, Robot	Isaac Asimov	Ultimate Books	80.8
3	Science and Technology	The Foundation Series	Isaac Asimov	Ultimate Books	100.1

Bookstore employees hope to find all books with less than 100 books in stock.

```
CREATE VIEW Book_Quantity AS
SELECT quantity_in_stock, title, author, publisher, price, type, book_edition
FROM Book NATURAL JOIN Edition
WHERE Book.book_id = Edition.book_id
AND quantity_in_stock <= 100;
```

	quantity_in_stock	title	author	publisher	price	type	book_edition
	过滤	过滤	过滤	过滤	过滤	过滤	过滤
1	100	Harry Potter	J K Rowling	Ultimate Books	10.1	audiobook	8
2	1	Harry Potter	J K Rowling	Ultimate Books	10.1	hardcover	4

Task4:

In general, if you can take the class seriously and read more related books provided by the teacher before the next class, then it is not difficult to complete this task. The difficulty for me is the choice of data. When I make a table, I usually encounter various small errors, such as semantics, spelling, etc., which will reduce efficiency and often forget attributes when building tables and queries and name. In addition, when the corresponding SQL query was produced, a lot of repetitions were made because natural connections were not used. And the view part is still not perfect, and there are loopholes in many places. Through this project, I understand that there are many problems in building the database together. Contacting the first project together, I feel that I have grown a lot.