

Contents

1. Background	3
2. Definitions of Entities and Relationships.....	3
2.1 Entities	3
2.2 Relations	3
3. Conceptual and Logical Database Design	4
3.1 Conceptual Database Design (EER Diagram).....	4
3.2 Logical Database Design (Relation Schema).....	5
3.3 Attributes Explanation	5
4. Physical Database Design	7
4.1 Denormalization.....	7
4.2 Data Type Definition	8
4.3 Index Design	9
5. DBMS Implement	12
5.1 Create Database	12
5.2 SQL Query Statement	15

1. Background

The outbreak of Covid-19 has become a hot emergency in the world. Research and vaccination are needed and scheduled. Therefore, our group choose this topic as our project theme, designing a database which shows the process from vaccine appointment to customers' vaccination. This will be a guide for some areas that have not yet started vaccinating residents.

2. Definitions of Entities and Relationships

2.1 Entities

- 1.1.1. **Product:** Different kind of vaccines.
- 1.1.2. **Customer:** Individuals who apply for the covid-19 vaccines.
- 1.1.3. **Application:** Vaccination order submitted by customers.
- 1.1.4. **Provider:** Employees in the hospitals who give customers injections.
- 1.1.5. **Clinic:** Institutions where the customers get inoculated.
- 1.1.6. **Suppliers:** Institutions who develop, manufacture and supply vaccines.
- 1.1.7. **Shipment:** Shipping order.
- 1.1.8. **Record:** Actual injection record.

2.2 Relations

2.2.1 Customer-Application

Cardinality: 1:1

Explanation: In a limited time, a customer can only submit one application. It is not allowed in policy for a customer to submit multiple applications at the same time.

2.2.2 Application-Clinics

Cardinality: N:1

Explanation: A clinic can accept many applications. On the application, there are many clinics for customers to choose. Customers can choose a clinic close to them to give priority to vaccination.

2.2.3 Application-Product

Cardinality: N:1

Explanation: An application can only specify one type of vaccine.
Customers can choose to be vaccinated against different types.

2.2.4 Product-Shipment

Cardinality: 1:N

Explanation: A type of vaccine can be delivered in different shipping orders.
A shipment order can only deliver one type of vaccines a time.

2.2.5 Shipment-Supplier

Cardinality: N:1

Explanation: A supplier can deliver its product in different shipping orders.
A shipment order can only deliver product from a certain supplier.

2.2.6 Clinics- Provider

Cardinality: 1:N

Explanation: A clinics may employ several providers.

2.2.7 Application- Record

Cardinality: 1:1

Explanation: A application will lead to be a injection record.

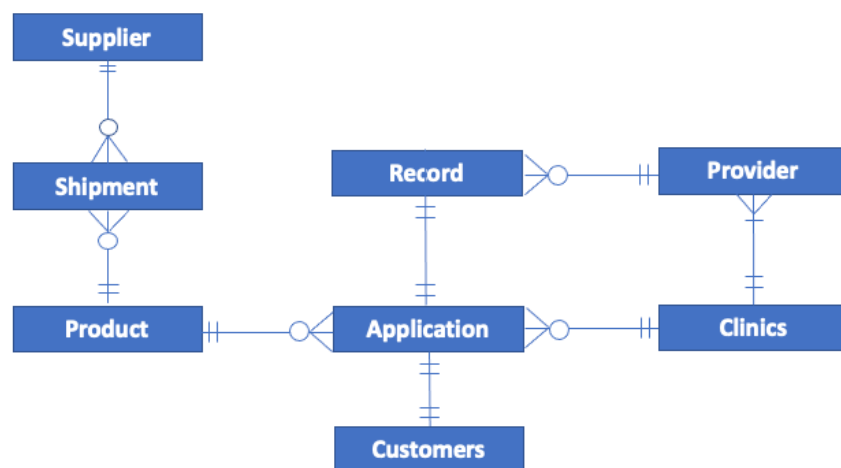
2.2.8 Record- Provider

Cardinality: N:1

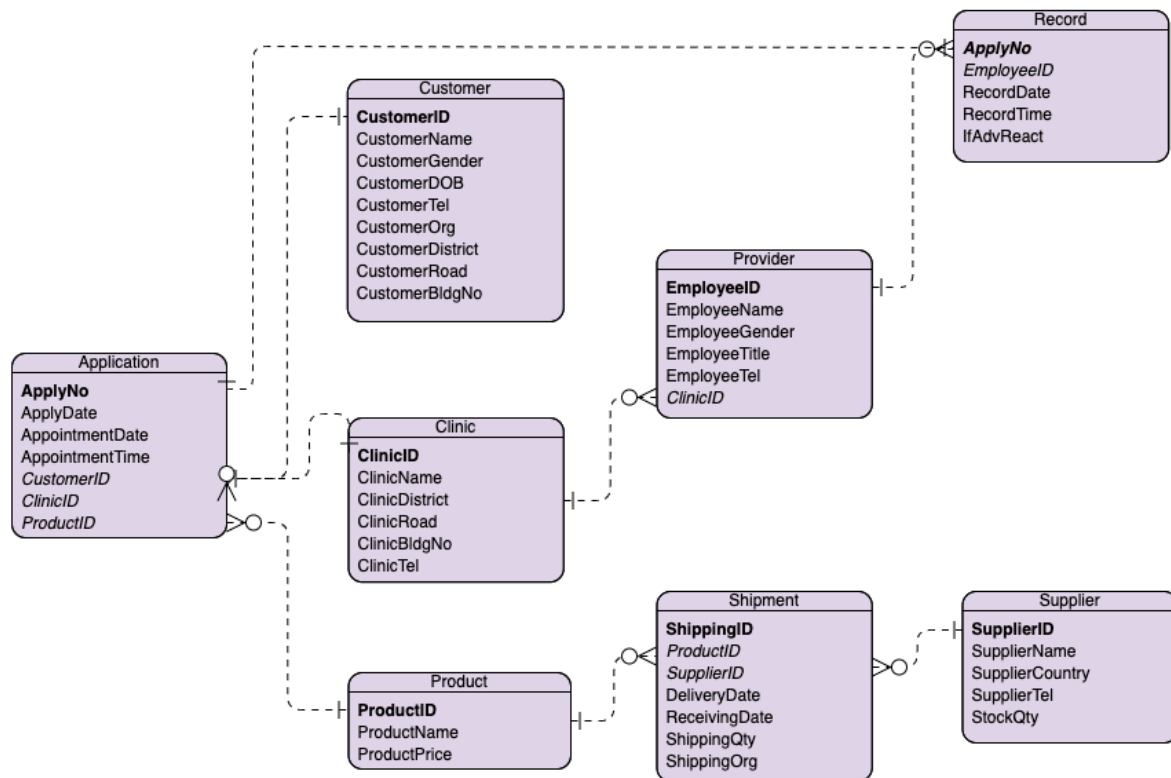
Explanation: A provider (a nurse or doctor) can give injection to many applicants, but a record only has one actual vaccine provider.

3. Conceptual and Logical Database Design

3.1 Conceptual Database Design (EER Diagram)



3.2 Logical Database Design (Relation Schema)



3.3 Attributes Explanation

Entity	Attribute	PK	FK	Explanation
Application	ApplyNo	Yes	No	The unique number of the application.
	ApplyDate	No	No	The date that the application is made.
	AppointmentDate	No	No	The date that appointed by the applicant for injection.
	AppointmentTime	No	No	The time that appointed by the applicant for injection.
	CustomerID	No	Yes	The unique number of the applicant.
	ClinicID	No	Yes	The unique number of the clinic.
	ProductID	No	Yes	The unique number of the vaccine.
Customer	CustomerID	Yes	No	The unique number of the applicant.
	CustomerName	No	No	The name of the applicant.
	CustomerGender	No	No	The gender of the applicant.
	CustomerDOB	No	No	The date of birth of the applicant.
	CustomerTel	No	No	The contact telephone numbers of the applicant.
	CustomerOrg	No	No	The organization that the applicant works for or belongs to.
	CustomerDistrict	No	No	The district of the applicant's living address.

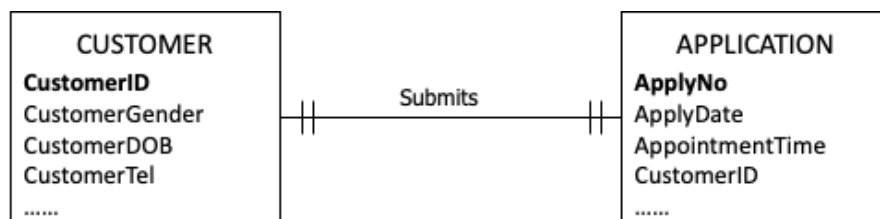
	CustomerRoad	No	No	The road of the applicant's living address.
	CustomerBldgNo	No	No	The house number of the applicant's living address.
Clinic	ClinicID	Yes	No	The unique number of the clinic which provides vaccines.
	ClinicName	No	No	The name of the clinic which provides vaccines.
	ClinicDistrict	No	No	The district of the clinic's address.
	ClinicRoad	No	No	The road of the clinic's address.
	ClinicBldgNo	No	No	The house number of the clinic's address.
	ClinicTel	No	No	The telephone number of the clinic.
Provider	EmployeeID	Yes	No	The unique number of the employee.
	EmployeeName	No	No	The name of the employee.
	EmployeeGender	No	No	The gender of the employee.
	EmployeeTitle	No	No	The job title of the employee.
	EmployeeTel	No	No	The telephone number of the employee.
	ClinicID	No	Yes	The unique number of the clinic which provides vaccines.
Product	ProductID	Yes	No	The unique number of each type of vaccine.
	ProductName	No	No	The name of each type of vaccine.
	ProductPrice	No	No	The price of each type of vaccine.
Shipment	ShippingID	Yes	No	The unique number of the application.
	ProductID	No	Yes	The unique number of the product.
	SupplierID	No	Yes	The unique number of the supplier.
	DeliveryDate	No	No	The date when the products were sent out.
	ReceivingDate	No	No	The date when the clinics received the product.
	ShippingQty	No	No	The quantity of the shipping.
	ShippingOrg	No	No	The organization of the shipping.
Supplier	SupplierID	Yes	No	The unique number of the supplier.
	SupplierName	No	No	The name of the supplier.
	SupplierCountry	No	No	The country that the supplier belongs to.
	SupplierTel	No	No	The telephone number of the supplier.
	StockQty	No	No	The quantity of stock this supplier holds.
Record	ApplyNo	Yes	Yes	The unique number of the application.
	RecordDate	No	No	The actual date of the injection.
	RecordTime	No	No	The actual time of the injection.
	EmployeeID	No	Yes	The ID of the employee that actual give the rejection.
	IfAdvReact	No	No	If the applicant has any adverse reaction after the injection.

4. Physical Database Design

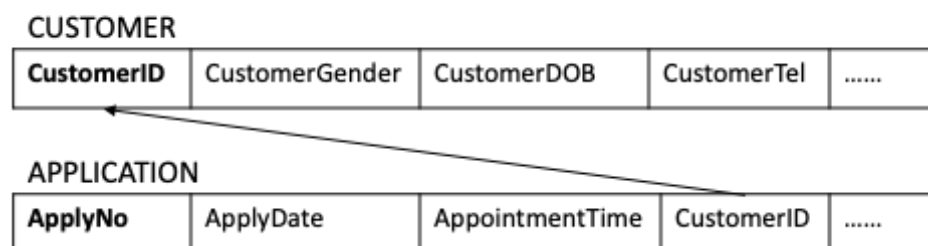
4.1 Denormalization

To improve the efficiency for key functions in this vaccine application, the logical designed relation database system is supposed to be denormalized on a physical level. The one-to-one relationship seems to be a proper candidate for denormalization in this case, which on one hand can improve the system performance, on the other costs little in wasting storage space and data integrity or consistency threats.

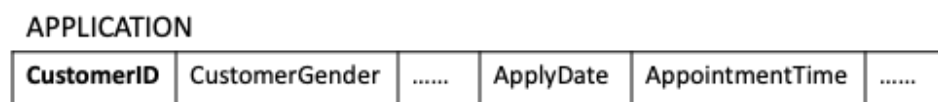
As shown below, **Customer** and **Application** which follow a one-to-one relationship is joint into a table. Note that each of the customer has a certain application, so that the application date and other information of the denormalized relation may all be applied with no wasting space. Moreover, data integrity is maintained robustly because there is no multiple records with the same customer ID and related details in this relationship.



Normalized relation:



Denormalized relation:



In this way, the physical deployment can be conducted, with balance between query performance and cost of data duplication.

4.2 Data Type Definition

Entity	Attribute	PK	FK	Data Type
Application	CustomerID	Yes	No	VARCHAR (11)
	CustomerName	No	No	VARCHAR(50)
	CustomerGender	No	No	VARCHAR(11)
	CustomerDOB	No	No	DATE
	CustomerTel	No	No	VARCHAR(11)
	CustomerOrg	No	No	VARCHAR(50)
	CustomerDistrict	No	No	VARCHAR(50)
	CustomerRoad	No	No	VARCHAR(50)
	CustomerBldgNo	No	No	VARCHAR(50)
	ApplyDate	No	No	DATE
	AppointmentDate	No	No	DATE
	AppointmentTime	No	No	TIME
	ClinicID	No	Yes	VARCHAR (11)
	ProductID	No	Yes	VARCHAR (11)
Clinic	ClinicID	Yes	No	VARCHAR (11)
	ClinicName	No	No	VARCHAR (50)
	ClinicDistrict	No	No	VARCHAR (50)
	ClinicRoad	No	No	VARCHAR (50)
	ClinicBldgNo	No	No	VARCHAR (50)
	ClinicTel	No	No	VARCHAR(11)
Provider	EmployeeID	Yes	No	VARCHAR (11)
	EmployeeName	No	No	VARCHAR (50)
	EmployeeGender	No	No	VARCHAR(11)
	EmployeeTitle	No	No	VARCHAR (50)
	EmployeeTel	No	No	VARCHAR(11)
	ClinicID	No	Yes	VARCHAR (11)
Product	ProductID	Yes	No	VARCHAR (11)
	ProductName	No	No	VARCHAR (50)
	ProductPrice	No	No	DECIMAL (9,2)
Shipment	ShippingID	Yes	No	VARCHAR (11)
	ProductID	No	Yes	VARCHAR (11)
	SupplierID	No	Yes	VARCHAR (11)
	DeliveryDate	No	No	DATE
	ReceivingDate	No	No	DATE
	ShippingQty	No	No	INT (11)
	ShippingOrg	No	No	VARCHAR (50)
Supplier	SupplierID	Yes	No	VARCHAR (11)

	SupplierName	No	No	VARCHAR (50)
	SupplierCountry	No	No	VARCHAR (50)
	SupplierTel	No	No	VARCHAR (11)
	StockQty	No	No	INT (11)
Record	CustomerID	Yes	Yes	VARCHAR (11)
	RecordDate	No	No	DATE
	RecordTime	No	No	TIME
	EmployeeID	No	Yes	VARCHAR (11)
	IfAdvReact	No	No	VARCHAR (11)

4.3 Index Design

To make this database more users-friendly, we design a join index to make the search speed faster, based on the tables below.

a. Customer and Application (Denormalized)

We design a join index based on two tables: Customer table and denormalized Application table. According to the rules of setting indexes, 'CustomerID' are the primary key of both Customer table and denormalized Application table, so the primary key 'CustomerID' will become an index automatically.

Normally, when using a Vaccine Application System, most of the customers searching for a record by entering their names, and they would like to know about the application date and appointment time reserved for them. Therefore, we make CustomerID the join index for non-key columns 'CustomerName', 'ApplyDate', and 'AppointmentTime' together.

Customer

CustomerID (PK)	Customer Name	Customer Gender	Customer DOB	Customer Tel	Customer Org	Customer District	Customer Road	Customer BldgNo

Application (Denormalized)

CustomerID (PK)	CustomerGender	ApplyDate	AppointmentTime
-----------------	----------------	-------	-----------	-----------------	-------

Join index

CustomerID	CustomerName	ApplyDate	AppointmentTime

b. Clinic and Product and Provider

We design the second join index based on another three tables: Clinic table, Product table and Provider table. According to the rules of setting indexes, ‘ClinicID’, ‘ProductID’, and ‘Provider ID’ are primary keys, so they will primary become the indexes automatically.

As the places where customers go to get injections of vaccines, Clinics need to prepare different products of vaccines, and hire doctors and nurses as providers. Therefore, when searching for data of clinics, usually, people would like to know which kind of vaccine they can provide, and who will help them to get injections. Therefore, we set ‘ClinicName’, ‘ProductName’, ‘EmployeeName’, and ‘EmployeeTitle’ to be join indexes.

Clinic

ClinicID (PK)	ClinicName	ClinicDistrict	ClinicRoad	ClinicBidgNo	ClinicTel

Product

ProductID (PK)	ProductName	ProductPrice

Provider

EmployeeID(PK)	EmployeeName	EmployeeGender	EmployeeTitle	EmployeeTel	ClinicID (FK)

Join index

ClinicID	ProductID	EmployeeID	ClinicName	ProductName	EmployeeName	EmployeeTitle

c. Product, Shipment, and Supplier

We design the third join index based on another three tables: Product table, Shipment table and Supplier table. According to the rules of setting indexes, 'ProductID', 'ShippingID', and 'SupplierID' are primary keys, so they will become the indexes automatically.

Usually, after clinics ordered certain brand of vaccines, suppliers will deliver vaccines via some shipping organizations. At that time, clinics who ordered vaccines may be eager to know when they can receive vaccines, and they can take delivery of vaccines from which shipping organizations. Thus, we set 'ProductName', 'SupplierName', 'Delivery Date', 'ReceivingDate', 'ShippingQty', 'ShippingOrg' to be join indexes, so as to save time when staff in clinics search for their orders.

Product

ProductID (PK)	ProductName	ProductPrice

Shipment

ShippingID(PK)	ProductID(FK)	SupplierID(FK)	DeliveryDate	ReceivingDate	ShippingQty	ShippingOrg

Supplier

SupplierID(PK)	SupplierName	SupplierCountry	SupplierTel	StockQty

5. DBMS Implement

5.1 Create Database

```
CREATE TABLE Clinic (  
  ClinicID VARCHAR(11) NOT NULL,  
  ClinicName varchar(50) NOT NULL,  
  ClinicDistrict varchar(50) NOT NULL,  
  ClinicRoad varchar(50) NOT NULL,  
  ClinicBidgNo varchar(50) NOT NULL,  
  ClinicTel varchar(50) DEFAULT NULL,  
  PRIMARY KEY (ClinicID)  
)  
  
INSERT INTO Clinic VALUES (1,'HKU Shenzhen Hospital','Futian','Haiyuan 1st  
Road','Number 1','+86-755-8691-3333');  
INSERT INTO Clinic VALUES (2,'PKU Shenzhen Hospital','Futian','Lianhua  
Road','Number 1120','+86-755-8392-3333');  
INSERT INTO Clinic VALUES (3,'Shenzhen People Hospital','Luohu','Dongmen North  
Road','Number 1017','+86-755-2553-3018');  
INSERT INTO Clinic VALUES (4,'Shenzhen Second People Hospital','Futian','Sungang  
West Road','Number 3002','+86-755-8336-6388');  
INSERT INTO Clinic VALUES (5,'Shenzhen Hang Seng Hospital of  
SMU','Baoan','Syracuse Road','Number 315','+86-755-252-7305');  
INSERT INTO Clinic VALUES (6,'Union Shenzhen Hospital of HUST','Nanshan','Garden  
Road','Number 415','+86-715-252-7305');  
INSERT INTO Clinic VALUES (7,'Shenzhen Hospital of GUCM','Futian','Beihuan  
Avenue','Number 515','+86-755-8354-8611');  
  
CREATE TABLE Product (  
  ProductID VARCHAR(11) NOT NULL,  
  ProductName varchar(50) NOT NULL,  
  ProductPrice decimal(9,2) NOT NULL,  
  PRIMARY KEY (ProductID)  
)  
  
INSERT INTO Product VALUES (1,'AZD1222','74.55');  
INSERT INTO Product VALUES (2,'mRNA-1273','122.45');  
INSERT INTO Product VALUES (3,'JNJ-78436735','64.55');  
INSERT INTO Product VALUES (4,'BNT162b2','87.45');  
INSERT INTO Product VALUES (5,'NVX-CoV2373','134.55');  
INSERT INTO Product VALUES (6,'Sputnik V','123.45');  
INSERT INTO Product VALUES (7,'CoronaVac','67.55');  
INSERT INTO Product VALUES (8,'GSK','55.45');  
  
CREATE TABLE Application (  
  CustomerID VARCHAR(11) NOT NULL ,  
  CustomerName varchar(50) NOT NULL,
```

```

CustomerGender varchar(6) NOT NULL,
CustomerDOB date NOT NULL,
CustomerTel varchar(50) NOT NULL,
CustomerOrg varchar(50) NOT NULL,
CustomerDistrict varchar(50) NOT NULL,
CustomerRoad varchar(50) NOT NULL,
CustomerBldgNo varchar(50) NOT NULL,
ApplyDate date NOT NULL,
AppointmentDate date NOT NULL,
AppointmentTime time NOT NULL,
ClinicID VARCHAR(11) NOT NULL,
ProductID VARCHAR(11) NOT NULL,
PRIMARY KEY (CustomerID),

CONSTRAINT fk_Application_ClinicID FOREIGN KEY (ClinicID) REFERENCES Clinic
(ClinicID) ON UPDATE CASCADE,
CONSTRAINT fk_Application_ProductID FOREIGN KEY (ProductID) REFERENCES Product
(ProductID) ON UPDATE CASCADE
)
INSERT INTO Application VALUES(1,'Babara','Female','1986-03-28','781-932-
9754','Hospital','Futian','Garden Road','266','2020-01-01','2020-02-
09','17:15:00',1,2);
INSERT INTO Application VALUES(2,'Ines','Male','1986-04-13','804-427-
9456','Government','Futian','Lillian Crossing','1031','2020-01-02','2020-02-
09','19:15:23',4,6);
INSERT INTO Application VALUES(3,'Freddi','Female','1985-02-07','719-724-
7869','Company','Nanshan','Mosinee Center','255','2020-01-03','2020-02-
10','21:15:15',7,4);
INSERT INTO Application VALUES(4,'Ambur','Male','1974-04-14','407-231-
8017','Government','Nanshan','Ohio Trail','3101','2020-01-04','2020-02-
10','22:15:07',7,8);
INSERT INTO Application VALUES(5,'Clemmie','Female','1973-11-07','406-231-
8217','Hospital','Futian','Lawn Avenue','2','2020-01-05','2020-02-
10','17:15:22',6,1);
INSERT INTO Application VALUES(6,'Elka','Male','1991-09-04','781-932-
9754','Company','Baoan','Spohn Circle','1','2020-01-06','2020-02-
12','18:15:44',3,6);
INSERT INTO Application VALUES(7,'Ilene','Female','1964-08-30','312-480-
8498','Hospital','Baoan','Arapahoe Terrace','233','2020-01-07','2020-02-
13','08:15:45',2,5);

CREATE TABLE Provider (
EmployeeID VARCHAR(11) NOT NULL,
EmployeeName varchar(50) NOT NULL,
EmployeeGender varchar(50) NOT NULL,
EmployeeTitle varchar(50) NOT NULL,
EmployeeTel varchar(50) NOT NULL,
ClinicID varchar(11) NOT NULL,
PRIMARY KEY (EmployeeID),

```

```

    CONSTRAINT fk_ClinicID FOREIGN KEY (ClinicID) REFERENCES Clinic (ClinicID) ON
UPDATE CASCADE
)
INSERT INTO Provider VALUES (1,'Brushfield','Female','Executive Secretary','804-
427-9456',1);
INSERT INTO Provider VALUES (2,'MacCaffrey','Male','Executive Secretary','781-
932-9754',2);
INSERT INTO Provider VALUES (3,'Boagey','Female','Statistician III','719-724-
7869',4);
INSERT INTO Provider VALUES (4,'Roseburgh','Male','Assistant Professor','407-231-
8017',3);
INSERT INTO Provider VALUES (5,'Betchley','Female','Social Worker','312-480-
8498',6);
INSERT INTO Provider VALUES (6,'Twiddell','Male','Financial Advisor','615-641-
4759',6);
INSERT INTO Provider VALUES (7,'Naseby','Female','Computer Systems Analyst
IV','559-181-3744',7);

CREATE TABLE Supplier (
    SupplierID varchar(11) NOT NULL,
    SupplierName varchar(50) NOT NULL,
    SupplierCountry varchar(50) NOT NULL,
    SupplierTel varchar(50) NOT NULL,
    StockQty varchar(11) NOT NULL,
    PRIMARY KEY (SupplierID)
)
INSERT INTO Supplier VALUES (1,'Kexing','China','804-427-9456',12336);
INSERT INTO Supplier VALUES (2,'WuhanShengwu','China','312-480-8498',32455);
INSERT INTO Supplier VALUES (3,'Kexing','China','615-641-4759',36745);
INSERT INTO Supplier VALUES (4,'WuhanShengwu','China','941-527-3977',34335);
INSERT INTO Supplier VALUES (5,'KangXiNuo','China','804-427-9456',23325);
INSERT INTO Supplier VALUES (6,'Kexing','China','559-181-3744',34515);
INSERT INTO Supplier VALUES (7,'Kexing','China','404-246-3370',30095);

CREATE TABLE Record (
    CustomerID varchar(11) NOT NULL,
    RecordDate date NOT NULL,
    RecordTime TIME NOT NULL,
    EmployeeID varchar(11) NOT NULL foreign key references Provider (EmployeeID),
    IfAdvReact char(3) NOT NULL,
    PRIMARY KEY (CustomerID),
    CONSTRAINT fk_CustomerID FOREIGN KEY (CustomerID) REFERENCES Application
(CustomerID) ON UPDATE CASCADE,
)
INSERT INTO Record VALUES (1,'2020-2-14','17:15:00',2,'No');
INSERT INTO Record VALUES (2,'2020-2-14','20:25:36',3,'No');
INSERT INTO Record VALUES (3,'2020-2-15','22:25:35',6,'No');

```

```

INSERT INTO Record VALUES (4, '2020-2-16', '11:34:15', 6, 'No');
INSERT INTO Record VALUES (5, '2020-2-16', '09:08:13', 4, 'No');
INSERT INTO Record VALUES (6, '2020-2-16', '23:10:45', 5, 'No');
INSERT INTO Record VALUES (7, '2020-2-17', '12:00:00', 6, 'No');

CREATE TABLE Shipment (
  ShippingID varchar(11) NOT NULL,
  ProductID varchar(11) NOT NULL,
  SupplierID varchar(11) NOT NULL,
  DeliveryDate date NOT NULL,
  ReceivingDate date NOT NULL,
  ShippingQty int NOT NULL,
  ShippingOrg varchar(50) NOT NULL,
  PRIMARY KEY (ShippingID),
  CONSTRAINT fk_ProductID FOREIGN KEY (ProductID) REFERENCES Product (ProductID)
ON UPDATE CASCADE,
  CONSTRAINT fk_SupplierID FOREIGN KEY (SupplierID) REFERENCES Supplier
(SupplierID) ON UPDATE CASCADE
)
INSERT INTO Shipment VALUES (1, 1, 1, '2020-01-06', '2021-02-22', 2332, 'SF');
INSERT INTO Shipment VALUES (2, 6, 2, '2020-01-08', '2021-02-22', 4222, 'YT');
INSERT INTO Shipment VALUES (3, 3, 2, '2020-01-12', '2021-02-22', 2452, 'EMS');
INSERT INTO Shipment VALUES (4, 2, 1, '2020-01-14', '2021-02-22', 4452, 'JD');
INSERT INTO Shipment VALUES (5, 3, 5, '2020-01-16', '2021-02-22', 2562, 'YT');
INSERT INTO Shipment VALUES (6, 1, 1, '2020-02-18', '2021-02-22', 2352, 'EMS');
INSERT INTO Shipment VALUES (7, 1, 1, '2020-02-19', '2021-02-22', 2453, 'SF');

```

5.2 SQL Query Statement

```

/*1 2020 年 2 月 15 日之前, 福田区接种疫苗的信息
The information on vaccination in Futian District before 2020/02/15 */
select* from Record r
left join Provider p on r.EmployeeID=p.EmployeeID
left join Clinic c on p.ClinicID=c.ClinicID
where RecordDate <'2020-02-15' and ClinicDistrict= 'Futian'

/*2 接种人数增长 (减少) 率 (%)
Daily growth rate of vaccinated people */
select t2.RecordDate, 100*(t2.number-t1.number)/t1.number as rate
from (select RecordDate ,count(*) as number
from Record
group by RecordDate) t1,
(select RecordDate ,count(*) as number
from Record
group by RecordDate) t2
where datediff(dd, t1.RecordDate, t2.RecordDate)=1

```

```

/* 3 某用户的预约信息
Appointment information named Ines */
select*
from Application
where CustomerName='Ines'

/* 4 疫苗品牌的接种排行榜
Ranking of the number of people vaccinated by different vaccine brands */
select ProductName, count(*) as Number
from Application a left join Product p on a.ProductID=p.ProductID
group by ProductName
order by 2 desc

/* 5 疫苗地区接种人次排行
Ranking of vaccinations of different regions */
select CustomerDistrict, count(*) as Number
from Application
group by CustomerDistrict
order by 2 desc

/* 6 雇员最多的诊所的信息
Informantion on the clinic with the most employees */
select top 1 c.ClinicName, count(c.ClinicID) as number
from Provider p left join Clinic c on p.ClinicID=c.ClinicID
group by c.ClinicID, c.ClinicName
order by 2 desc

/* 7 从预约到接种的时间间隔排序
Sort by time interval from appointment to vaccination */
select a.CustomerID,datediff(d, AppointmentDate,RecordDate) as 'interval'
from Application a left join Record r on a.CustomerID=r.CustomerID
order by 2 desc

/* 8 职位为行政秘书的雇员的姓名和电话
Name and telephone Number of the employee whose title is Executive Secretary */
select EmployeeName, EmployeeTel
from Provider
where EmployeeTitle='Executive Secretary'

/* 9 快递公司被厂商选择的次数 从高到低排行
Ranking of the number of courier companies cooperating with vaccine manufacturers */
select t.SupplierName,count(*) as NumberOfExpressCompany
from
(select distinct st.ShippingOrg,sr.SupplierName
from Shipment st
left join Supplier sr
on st.SupplierID=sr.SupplierID) t
group by t.SupplierName

```

```
order by 2 desc

/* 10 大于 40 岁的女性的接种人数
Number of vaccinated female over 40 years old */
select count(CustomerID) as NumberOfPeople
from
(select *
from Application
where datediff(d, CustomerDOB, getdate())/365 >= 40 and CustomerGender = 'Female') t;
```