Final Documentation – Group3

60181866 김민기, 60171938 차유상, 60171917 이승학, 60181945 조수빈

목차

1. Group members and their roles
2. Project proposal

Ⅰ. General description of the company

1. Business goal of the company

2. Products / services of the company

Ⅱ. Purpose of the database system

1. Product inventory management

2. Customer management

3. Client management

1. User requirements
2. Conceptual Design
3. Logical design
4. Physical Design
5. DDL
6. DML(query)
7. System manual
8. Group members and their roles

60181866 김민기 :

60171938 차유상 :

60171917 이승학 :

60181945 조수빈 :

project proposal 작성, User requirements 작성, conceptual design, logical design, physical design, DDL 작성, DML 작성, final report 작성

1. Project Proposal

**Ⅰ. General description of the company**

**1.Business goal of the company**

현재 pc방 산업은 단순히 사양 높은 pc만 배치하면 창업이 가능하여 수많은 pc방들이 생기고 있다. 이로 인해 pc방들은 자신들만의 방법으로 경쟁력을 높여 고객들을 유지하고 있다. 스타덤 pc방 서창점은 경쟁력을 높이기 위해 최고급 pc 사양은 기본이며 각종 이벤트를 진행하고 좋은 질의 음식 및 음료를 제공하는 서비스를 제공하고 있다. 이벤트는 고객들을 유지 및 확장하기 위해 진행하며 pc방의 손해를 감수하면서도 진행을 하고 있다. 음식은 현재 일반 음식점에서 판매하는 음식의 수준으로 제공하고 있으며 음료 또한 일반 카페에서 제공하는 음료의 수준으로 제공하고 있다. 이로 인해 고객은 식사 또한 해결할 수 있어 이용하고자 하는 시간 동안 pc방을 나가는 불편함을 겪지 않아도 된다. 고객의 음식 및 음료 주문이 많아지면서 스타덤 pc방 서창점은 이익 구조는 약 70%가 제공하는 음식 및 음료의 값에서 나오고 있으며 자연스럽게 고객의 pc 사용료 보다 제공하는 음식 및 음료의 값으로 매출 초점이 바뀌었다. 이를 살펴봤을 때 스타덤 pc방의 비즈니스 목표는 기존 고객 유지 및 신규 고객 확보 그리고 고객에게 좋은 질의 음식 및 음료 서비스를 차질 없이 제공하여 매출액(pc 사용료 + 음식 및 음료 제공 값) 확보를 하는 것이다.

**2.Products / services of the company**

스타덤 pc방 서창점에서 제공하는 제품은 pc, 음식 및 음료가 있으며 서비스는 프린트기 및 와이파이 기기 사용, 각종 이벤트 진행이 있다. 먼저 제품에서 pc는 고사양의 그래픽 카드를 사용하고 있으며 32인치 모니터를 제공하고 있다. 음식의 종류로는 면 종류((컵) 라면, 우동, 냉모밀 등), 떡볶이 종류(떡볶이, 라볶이), 밥 종류(볶음밥, 국밥, 덮밥, 돈가스), 핫도그 종류, 스낵 종류(소떡, 치킨꼬치, 치즈스틱 등)를 제공하고 있다. 음료의 종류로는 캔음료를 기본으로 일반 자영업의 카페에서 판매하는 음료의 레시피를 가져와 커피 종류(아메리카노, 라떼 등)와 에이드 종류(아이스티, 청포도, 레몬, 자몽 등), 프라페 종류(초코, 녹차, 요거트 등)를 제공하고 있다. 프린트기는 전좌석 장 당 금액을 지불하면 프린트를 할수 있으며 와이파이 서비스는 무료로 이용할 수 있다. 이벤트 진행은 두달에 한번씩 진행을 한다. 이벤트의 종류로는 SNS에 pc방 홍보를 하면 음료를 제공하거나 각종 게임 대회 개최 및 특정 시간을 정하여 번호를 뽑아 해당 pc에 앉아 있는 고객에게 상품을 제공 등 다양한 이벤트를 제공하고 있다.

**Ⅱ. Purpose of the database system**

**1.Product inventory management**

스타덤 pc방 서창점에서 음식 및 음료의 재고는 직원 및 사장이 눈으로 직접 재고를 파악하여 부족한 것은 엑셀(스프레드시트)에 적어 주문하는 방식으로 채우고 있다. 눈으로 직접 파악하는 방식은 직원이 있는 재고를 없다고 생각할 수 있으며 3파트(오전, 오후, 야간)로 나눠 일하기 때문에 직원끼리 인수인가 잘못됐을 시 재고 주문이 중복되는 경우도 있다. 이에 중복된 재고로 유통기한 관련 문제가 발생할 수 있으며 있다고 생각한 재고가 없어 고객에게 음식 및 음료를 제공하지 못하여 고객에게 불편을 초래하는 문제가 발생한다. 이로 인해 발생하는 손해는 스타덤 pc방 서창점에게 부담이 되고 있다. pc 부품 재고도 음식 및 음료 재고와 같은 문제가 발생한다. 고장 난 부품을 수리 보내야 할 때 대체할 부품이 있는지 직원이 일일이 찾아봐야 한다. 또 이벤트를 진행할 때 pc 부품과 음식 및 음료를 제 공고 있는데 이때 정확한 재고를 몰라 재고가 없을 경우 고객에게 양해를 구한 후 쿠폰으로 대체하여 나중에 지급한 경우도 있다. 이에 스타덤 pc방 전 직원은 음식 및 음료, pc 부품의 재고를 부족한 재고가 무엇인지 또 있는 재고가 무엇인지 손쉽고 정확하게 재고 파악을 하여 손해를 줄이고자 한다.

**2.Customer management**

스타덤 pc방 서창점은 pc 부품과 음식 및 음료의 재고를 주문 및 pc 서버를 관리해 주는 거래처가 있다. 그러나 각 거래처의 명단은 메모장(txt 파일)로 거래처마다 따로 관리되고 있다. 이에 거래처에 발주 및 문의를 할 때 거래처의 정보를 찾는데 많은 시간이 소요되며 또 거래처의 갱신이 안 되어 있을 시 이전 거래처에 발주 및 문의를 하는 문제가 발생한다. 이러한 문제는 직원의 시간 낭비와 음식 및 음료의 경우 제품이 달라져 고객에게 불편함을 초래한다. 이에 각 거래처를 한눈에 확인할 수 있으며 갱신 시 쉽게 가능했으면 한다.

**3.Client management**

스타덤 pc방에서 고객은 회원가입을 통해 pc를 사용할 수 있으며 회원가입 한 고객의 정보는 pc방 클라이언트를 통해 저장된다. 회원가입은 고객이 ID, PW, 전화번호, 생년월일 만 입력하게 되면 가입되는 방식을 사용한다. 이는 한 고객이 회원가입을 여러 번 할 수 있는 방식으로 고객의 정보가 중복되는 문제가 발생하기도 한다. 이로 인해 고객이 ID를 까먹어 찾을 경우 많은 ID가 검색되는 문제가 발생하여 무슨 ID를 사용했는지 못 찾는 문제가 발생한다. 또한 성인 고객인 경우 생년월일을 조작하여 학생 ID와 성인 ID를 두 개씩 사용하여 학생 ID로 pc 사용료를 덜 내어 사용하는 문제도 발생한다. 제일 큰 문제는 이벤트 진행 시 중복된 아이디로 한 사람이 중복되게 상품을 지급받는 문제가 발생하여 다른 고객에게 불편함을 초래한다. 이에 고객이 회원 가입을 할 때 중복되지 않게 관리할 수 있으면 하고 고객의 정보 또한 쉽게 관리할 수 있었으면 한다.

1. user requirements

스타덤 PC방 서창점은 최고급 PC 사양과 각종 이벤트, 좋은 질의 음식 및 음료를 제공하여 경쟁력을 높이기 위해 노력하고 있다.

PC는 항상 고사양을 유지하려고 하며 그러기 위해서 Parts (CPU, Memory Size, Graphic Card)를 저장하여 더 높은 사양의 부품이 나올 때마다 교체를 해서 최고사양을 유지하고 있다. 또한 주변 기기로 I/O Device (Keyboard, Mouse, Headset)를 제공하며 듀얼 Monitor를 사용하는 좌석도 있다. PC에 대한 정보는 좌석 번호 Computer\_Num(identifier attribute)을 함께 저장하고자 한다.

컴퓨터와 음식의 재고는 각각의 Supplier를 통해 공급 받아 item에 저장되고, 각각 요리에 필요한 음식은 Food item로부터, 필요한 부품의 교체는 Computer item로부터 이루어진다.

item에 대한 정보는 Item\_Num (identifier attribute), name, Price, Manufacturer, location이 저장된다. 또한 supplier에게서 공급 받을 때 Date, Quantity, TotalPrice를 저장하여 각 재고가 언제, 얼마나 주문되었는지와 총 가격을 저장하고자 한다. 또한 item의 종류로는 Food item과 Computer item이 있는데, Food item에서는 각 음식에 대한 Shelf Life을 저장한다. Computer item에서는 각 컴퓨터의 Serial Number를 저장하며 컴퓨터나 주변기기의 교체가 일어났을 때는 교체한 날을 저장해야 한다. 또한 Item에는 Food와 Computer 둘 중 하나의 정보만 저장해야 한다.

Supplier에 대한 정보는 Vendor Num (identifier attribute), Name, Tel, Address(City, Road\_Name, Detail), Account(계좌)가 저장되며 여러 supplier에게 재고를 공급 받을 수 있고, 여러 supplier로부터 공급받은 재고는 공급 날짜와 함께 종류에 따라 여러 inventory에 저장될 수 있다. 모든 재고는 supplier를 통해서만 주문할 수 있다. 또한 한 번도 공급한 적이 없는 supplier여도 신규 계약이 되어 있다면 정보를 저장할 수 있다.

음식은 각 음식의 Food\_Num(identifier attribute)와 함께 Name, Category, Price를 저장한다. 또한 여러 음식에 대한 여러 개의 Set Menu가 구성될 수 있는데, Set Menu는 어떠한 음식에 추가되는 여러개의 Additional\_Menu과 함께 Price와 name을 함께 저장한다. 음식을 만들기 위해 필요한 재료들을 Food item에서 꺼내올 때 꺼낸 Date와 Quantity를 저장하여 재고 관리를 편히 하려고 한다. 여러 재료들로 여러 음식을 만들 수 있다.

스타덤 PC방 서창점은 근무자와 고객에 대한 정보 또한 저장하려고 한다. 먼저 공통적인 데이터로 Address(City, Road\_Name, Detail)와 SSN(identifier attribute), Name(First, Last), Phone Number, Birth를 저장한다.

PC방에 있는 사람의 정보는 근무자나 고객의 정보만 저장하며 근무자가 근무 외에 고객이 되어 PC를 사용할 수 있다. 고객이 컴퓨터를 사용할 때 날짜에 따라 사용 시간과 사용 금액을 저장하여 매출을 계산할 수 있게 한다.

근무자들은 Working\_hour과 Part-time을 추가로 저장하고 고객에는 Customer\_ID, Password, Total\_Use Time과 Remaining Time을 추가로 저장하려고 한다. 다수의 근무자가 다수의 고객을 관리할 수 있으며 일한 날짜 또한 함께 저장된다. 고객은 근무자가 없이는 PC방을 이용할 수 없으며 항상 근무자의 관리를 받아야 한다.

여러 명의 고객은 여러 개의 음식을 주문할 수 있다. 고객이 음식을 주문할 때, 주문한 Quantity와 Date, Total\_Price를 저장하여 매출을 계산할 수 있게 한다.

또한 근무자에 따라 급여가 결정되는데 이 급여 정보를 따로 저장하고자 한다. 근무자의 Account\_Num(partial key)에 gross pay를 계산하여 하나의 근무자에 대한 급여를 지급하려고 한다.

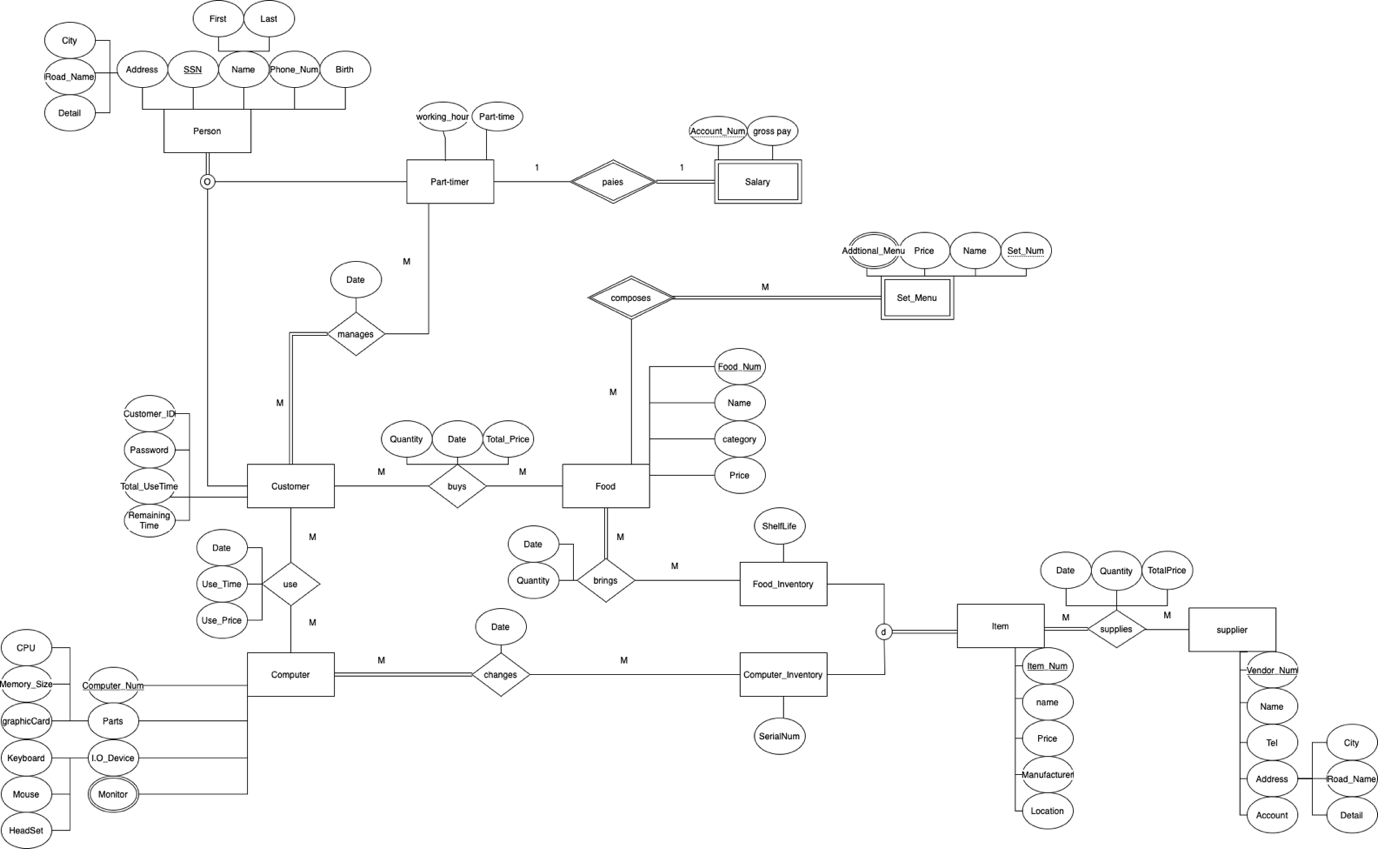
21.12.14 User requirements 수정사항 반영

* Supplier와 Inventory 간의 Cardinality constrain, relationship attribute 추가
  + 여러 supplier에게 재고를 공급 받을 수 있고, 여러 supplier로부터 공급받은 재고는 공급 날짜와 함께 종류에 따라 여러 inventory에 저장될 수 있다.
* Food와 Set\_Menu 간의 cardinality constraint 추가
  + 또한 여러 음식에 대한 여러 개의 Set Menu가 구성될 수 있는데, Set Menu는 어떠한 음식에 추가되는 여러개의 Additional\_Menu과 함께 Price와 name을 함께 저장한다.
* Customer와 Part-timer 간의 total/partial participation 추가
  + 고객은 근무자가 없이는 PC방을 이용할 수 없으며 항상 근무자의 관리를 받아야 한다.
* Unary 삭제 -> inventory\_num이 inventory\_num을 버릴 수 없음.
  + Shelf Life가 지난 여러 음식의 재고를 따로 저장해놓고 버릴 때 사용한다.

21.12.17 User requirements 수정사항 반영

* Inventory -> item으로 이름 변경하고 subtype의 이름도 Food\_item과 Computer\_item으로 변경
* Item과 Supplier 사이 supplier 쪽 partial participation으로 변경
  + 신규 계약했지만 아직 공급하지 않은 supplier의 정보가 저장될 수 있음.
  + 또한 한 번도 공급한 적이 없는 supplier여도 신규 계약이 되어 있다면 정보를 저장할 수 있다.
* Item과 supplier relationship에 quantity와 total price 추가하여 각 item에 대한 공급 날짜, 수량, 총 금액을 저장하게 하고, item에 세세한 항목들(item\_Num, name, price, manufacturer, location) 정의함.
* Customer와 Computer 사이 relationship에 사용시간과 사용금액 저장하여 하루 동안 쓴 금액 구할 수 있게 함.
  + 고객이 컴퓨터를 사용할 때 날짜에 따라 사용 시간과 사용 금액을 저장하여 매출을 계산할 수 있게 한다.
* Customer에 있던 use\_time -> total\_use\_time으로 변경하여 헷갈리지 않게 함.

1. conceptual design



1. Logical design

* 모든 inventory -> item으로 이름 변경

Person(SSN, City, Road\_Name, Detail, First, Last, Phone\_Num, Birth)

Part-timer(P\_SSN, working\_hour, Prat\_time)

Foreign Key(P\_SSN) references Person(SSN)

Salary(P\_SSN, Account\_Num, Gross\_pay)

Foreign Key(P\_SSN) references Part-timer(P\_SSN)

Customer(C\_SSN, Customer\_ID, Password, Total\_Use\_time, Remaining\_Time)

Foreign Key(C\_SSN) references Person(SSN)

* Use time -> toal\_use\_time으로 변경

Manage\_Customer(P\_SSN, C\_SSN, Date)

Foreign Key(P\_SSN) references Part-timer(P\_SSN)

Foreign Key(C\_SSN) references Customer(C\_SSN)

Computer(Computer\_Num, CPU, Memory\_Size, Graphiccard, Keyboard, Mouse, Headset)

Computer\_Monitor(Computer\_Num, Monitor)

Foreign Key(Computer\_Num) references Computer(Computer\_Num)

Use\_Computer(C\_SSN, Computer\_Num, Date, Use\_Time, Use\_Price)

Foreign Key(C\_SSN) references Customer(C\_SSN)

Foreign Key(Computer\_Num) references Computer(Computer\_Num)

* Table명 use -> use\_computer로 변경, Use\_time, use\_price 추가

Food(Food\_Num, Name, Category, Price)

Set\_Menu(Food\_Num, Set\_Num, Price, Name)

Foreign Key(Food\_Num) references Food(Food\_Num)

Addtional\_Menu(Food\_Num, Set\_Num, Additional\_Menu)

Foregin Key(Food\_Num) references Set\_Menu(Food\_Num)

Foregin Key(Set\_Num) references Set\_Menu(Set\_Num)

Buy\_Food(C\_SSN, Food\_Num, Quantity, Date, Total\_Price)

Foreign Key(C\_SSN) references Customer(C\_SSN)

Foreign Key(Food\_Num) references Food(Food\_Num)

Supplier(Vendor\_Num, Name, Tel, City, Road\_Name, Detail, Account)

Item(Item\_Num, name, Price, Manufacturer, location)

Item\_Order(Vendor\_Num, Item\_Num, Date, Quantity, TotalPrice)

Foreign Key(Vendor\_Num) references Supplier(Vendor\_Num)

Foreign Key(Inventory\_Num) references Item(Item\_Num)

* Order -> item\_Order로 변경, Quantity, TotalPrice 추가

Food\_Item(F\_Item\_Num, ShelfLife)

Foreign Key(F\_Item\_Num) references Item(Item\_Num)

Bring\_Food(Food\_Num, F\_Item\_Num, Date, Quantity)

Foregin Key(Food\_Num) references Food(Food\_Num)

Foreign Key(F\_Item\_Num) references Food\_Item(F\_Item\_Num)

Computer\_Item(C\_Item\_Num, SerialNum)

Foreign Key(C\_Item\_Num) references Item(Item\_Num)

Change\_Computer(Computer\_Num, C\_Item\_Num, Date)

Foregin Key(Computer\_Num) references Computer(Computer\_Num)

Foreign Key(C\_Item\_Num) references Computer\_Item (C\_Item\_Num)

*Why it is in the Thrid Normal Form*

모든 attribute가 amotic한 상태이므로 1NF에 해당된다.

모든 non-primary key attribute value가 primary key에 Fully functional dependency하고있으므로 2NF에 해당된다.

모든 Entity의 non-primary-key attribute가 primary key에 의존적이므로 3NF를 충족한다.

1. Physical Design

**Table Name : Person**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data Type** | **Constraint** |
| SSN | Social security number | Number(13,0) | Not null |
| City | City | Varchar2(20) |  |
| Road\_name | Road name | Varchar2(30) |  |
| Detail | Detail address | Varchar2(20) |  |
| First\_name | First name | Varchar2(10) | Not null |
| Last\_name | Last name | Varchar2(10) | Not null |
| Phone\_num | Phone number | Varchar2(11) | Not null |
| Birth | Birth | Date |  |

**Table Name : Part\_Timer**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data Type** | **Constraint** |
| P\_SSN | Part-timer’s  Social security number | Number(13,0) | Not null,  References  Person(SSN) |
| Working\_hour | Total working hour  of month | Number(10,0) |  |
| Part\_time | Part-time | Varchar2(20) | Not null |

**Table Name : Salary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data Type** | **Constraint** |
| P\_SSN | Part-timer’s  Social security number | Number(13,0) | Not null,  References  Part\_Timer(P\_SSN) |
| Account\_num | Part-timer’s  Account number | Varchar2(15) | Not null |
| Gross\_pay | Gross pay of month | number(8,0) | Not null |

**Table Name : Customer**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data Type** | **Constraint** |
| C\_SSN | Customer’s  Social security number | Number(13,0) | Not null,  References  Person(SSN) |
| Customer\_id | Customer’s login id | Varchar2(20) | Not null |
| Customer\_password | Customer’s  login password | Varchar2(20) | Not null |
| Total\_use\_time | Total using computer time | Varchar2(10) |  |
| Remaining\_time | Remaining computer time | Varchar2(10) |  |

**Table Name : Manage\_Customer**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data Type** | **Constraint** |
| P\_SSN | Part-timer’s  Social security number | Number(13,0) | Not null,  References  Part\_Timer(P\_SSN) |
| C\_SSN | Customer’s  Social security number | Number(13,0) | Not null,  References  Customer(C\_SSN) |
| Manage\_Date | Each Date of part-timer’s managing customer | date | Not null |

**Table Name : Computer**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data Type** | **Constraint** |
| Computer\_Num | Computer number | Number(2,0) | Not null |
| CPU | CPU of computer | Varchar2(20) | Not null |
| Memory\_size | Memory size of computer | Number(2,0) | Not null |
| Graphic\_card | Graphic card of computer | Varchar2(20) | Not null |
| Keyboard | Keyboard of computer | Varchar2(20) | Not null |
| Mouse | Mouse of computer | Varcher2(20) | Not null |
| Headset | Headset of computer | Varchar2(20) | Not null |

**Table Name : Computer\_Monitor**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data Type** | **Constraint** |
| Computer\_Num | Computer number | Number(2,0) | Not null,  References Computer(Computer\_Num) |
| Monitor | Monitor of computer | Varchar(20) | Not null |

**Table Name : Use\_Computer**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data Type** | **Constraint** |
| C\_SSN | Customer’s  Social security number | Number(13,0) | Not null,  References  Customer(C\_SSN) |
| Computer\_Num | Computer number | Number(2,0) | Not null,  References Computer(Computer\_Num) |
| Use\_date | Date of Using computer | Date | Not null |
| Use\_time | Time of Using computer | Number (10,0) | Not null |
| Use\_price | Price of Using computer | Number(10,0) |  |

**Table Name : Food**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data Type** | **Constraint** |
| Food\_Num | Food number | Number(3,0) | Not null |
| Name | Food name | Varchar2(20) | Not null |
| Category | Category of food | Varchar2(20) | Not null |
| Price | Price of food | Number(10,0) | Not null |

**Table Name : Set\_Menu**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data Type** | **Constraint** |
| Set\_Num | Set number about food | Number(3,0) | Not null |
| Food\_Num | Food number | Number(3,0) | Not null  References Food(Food\_Num) |
| Price | Price of set menu | Number(10,0) | Not null |
| Name | Name of set menu | Varchar2(50) | Not null |

**Table Name : Additional\_Menu**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data Type** | **Constraint** |
| Set\_Num | Set number | Number(3,0) | Not null |
| Food\_Num | Food number | Number(3,0) | Not null  References Food(Food\_Num) |
| Additional\_menu | Additional menu (not exists for food, just special menu for set menu) | Varchar2(50) | Not null |

**Table Name : Buy\_Food**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data Type** | **Constraint** |
| C\_SSN | Customer’s  Social security number | Number(13,0) | Not null,  References  Customer(C\_SSN) |
| Food\_Num | Food number | Number(3,0) | Not null  References Food(Food\_Num) |
| Quantity | Quantity of customer’s buying food | Number(3,0) | Not null |
| Buy\_date | Date of customer’s buying food | Date | Not null |
| Total\_Price | Total Price of customer’s buying food | Number(10,0) |  |

**Table Name : Supplier**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data Type** | **Constraint** |
| Vendor\_Num | Vendor number | Number(11) | Not null |
| Name | Supplier name | Varchar2(20) | Not null |
| Tel | Tel of supplier | Varchar2(15) | Not null |
| City | City of supplier location | Varchar2(15) |  |
| Road\_Name | Road\_Name of supplier location | Varchar2(30) |  |
| Detail | Detail address of supplier location | Varcher2(50) |  |
| Account | Supplier account | Varchar2(20) | Not null |

**Table Name : Item**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data Type** | **Constraint** |
| Item\_Num | Item is Food or computer’s part,  Such Item\_Num | Number(11) | Not null |
| Name | Item name | Varchar2(20) | Not null |
| Price | Price of item | Number (8,0) | Not null |
| Manufacturer | Manufacturer of item | Varchar2(20) | Not null |
| Location | Item location | Varchar2(20) | Not null |

**Table Name : Item\_Order**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data Type** | **Constraint** |
| Vendor\_Num | Vendor number | Number(11) | Not null  References Supplier(Vendor\_Num) |
| Item\_Num | Item is Food’s part,  Such Item\_Num | Number(11) | Not null  References Item(Item\_Num) |
| Order\_date | Order date of item order | date | Not null |
| Quantity | quantity of item | Number(3,0) | Not null |
| TotalPrice | Total price of item order | Number(10,0) |  |

**Table Name : Food\_Item**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data Type** | **Constraint** |
| F\_Item\_Num | Item is Food’s part,  Such Item\_Num | Number(11) | Not null  References Item(Item\_Num) |
| ShelfLife | Shelflife of food | date |  |

**Table Name : Bring\_Food**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data Type** | **Constraint** |
| Food\_Num | Food number | Number(3,0) | Not null  References Food(Food\_Num) |
| F\_Item\_Num | Item is Food’s part,  Such Item\_Num | Number(11) | Not null  References Item(Item\_Num) |
| Bring\_date | bring date of item for food | date | Not null |
| Quantity | quantity of item | Number(3,0) | Not null |

**Table Name : Computer\_Item**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data Type** | **Constraint** |
| C\_Item\_Num | Item is Computer’s part,  Such Item\_Num | Number(11) | Not null  References Item(Item\_Num) |
| SerialNum | Shelflife of food | Varchar2(10) | Not null |

**Table Name : Change\_Computer**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data Type** | **Constraint** |
| Computer\_Num | Computer number | Number(2,0) | Not null,  References Computer(Computer\_Num) |
| C\_Item\_Num | Item is Computer’s part,  Such Item\_Num | Number(11) | Not null  References Item(Item\_Num) |
| Change\_date | bring date of item for computer | date | Not null |

1. DDL

create table person(

SSN number(13,0) not null,

city varchar2(20),

road\_name varchar2(30),

detail varchar2(20),

first\_name varchar2(10) not null,

last\_name varchar2(10) not null,

phone\_num varchar2(11) not null,

birth date,

CONSTRAINT person\_pk PRIMARY KEY (SSN));

create table part\_timer(

P\_SSN number(13,0) not null,

working\_hour number(10,0),

part\_time varchar2(20) not null,

CONSTRAINT part\_timer\_pk PRIMARY KEY (P\_SSN),

CONSTRAINT part\_timer\_fk FOREIGN KEY (P\_SSN) REFERENCES person(SSN));

create table salary(

P\_SSN number(13,0) not null,

account\_num varchar2(15) not null,

gross\_pay number(8,0) not null,

CONSTRAINT salary\_pk PRIMARY KEY (P\_SSN, account\_num),

CONSTRAINT salary\_fk FOREIGN KEY (P\_SSN) REFERENCES part\_timer(P\_SSN));

create table customer(

C\_SSN number(13,0) not null,

customer\_id varchar2(20) not null,

customer\_password varchar2(20) not null,

total\_use\_time varchar2(10),

remaining\_time varchar2(10),

CONSTRAINT customer\_pk PRIMARY KEY (C\_SSN),

CONSTRAINT customer\_fk FOREIGN KEY (C\_SSN) REFERENCES person(SSN));

create table manage\_customer(

P\_SSN number(13,0) not null,

C\_SSN number(13,0) not null,

Manage\_Date date not null,

CONSTRAINT manage\_customer\_pk PRIMARY KEY (P\_SSN, C\_SSN),

CONSTRAINT manage\_customer\_fk1 FOREIGN KEY (P\_SSN) REFERENCES part\_timer

(P\_SSN),

CONSTRAINT manage\_customer\_fk2 FOREIGN KEY (C\_SSN) REFERENCES customer

(C\_SSN));

create table computer(

computer\_num number(2,0) not null,

cpu varchar2(20) not null,

memory\_size number(2,0) not null,

graphic\_card varchar2(20) not null,

keyboard varchar2(20) not null,

mouse varchar2(20) not null,

headset varchar2(20) not null,

CONSTRAINT computer\_pk PRIMARY KEY (computer\_num));

create table computer\_monitor(

computer\_num number(2,0) not null,

monitor varchar(20) not null,

CONSTRAINT computer\_monitor\_pk PRIMARY KEY (computer\_num, monitor),

CONSTRAINT computer\_monitor\_fk FOREIGN KEY (computer\_num) REFERENCES computer(computer\_num));

create table use\_computer(

C\_SSN number(13,0) not null,

computer\_num number(2,0) not null,

use\_date date not null,

use\_time number(10,0) not null,

use\_price number(10,0),

CONSTRAINT use\_computer\_pk PRIMARY KEY (computer\_num, C\_SSN),

CONSTRAINT use\_computer\_fk1 FOREIGN KEY (computer\_num) REFERENCES computer(computer\_num),

CONSTRAINT use\_computer\_fk2 FOREIGN KEY (C\_SSN) REFERENCES customer(C\_SSN));

create table food(

food\_num number(3,0) not null,

name varchar2(20) not null,

category varchar2(20) not null,

price number(10,0) not null,

CONSTRAINT food\_pk PRIMARY KEY (food\_num));

create table set\_menu(

food\_num number(3,0) not null,

set\_num number(3,0) not null,

price number(10,0) not null,

name varchar2(50) not null,

CONSTRAINT set\_menu\_pk PRIMARY KEY (food\_num, set\_num),

CONSTRAINT set\_menu\_fk FOREIGN KEY (food\_num) REFERENCES food(food\_num));

create table additional\_menu(

food\_num number(3,0) not null,

set\_num number(3,0) not null,

additional\_menu varchar2(50) not null,

CONSTRAINT additional\_menu\_pk PRIMARY KEY (food\_num, set\_num, additional\_menu),

CONSTRAINT additional\_menu\_fk FOREIGN KEY (food\_num, set\_num) REFERENCES set\_menu(food\_num, set\_num));

create table buy\_food(

C\_SSN number(13,0) not null,

food\_num number(3,0) not null,

quantity number(3,0) not null,

buy\_date date not null,

total\_price number(10,0),

CONSTRAINT buy\_food\_pk PRIMARY KEY (C\_SSN, food\_num),

CONSTRAINT buy\_food\_fk1 FOREIGN KEY (C\_SSN) REFERENCES Customer(C\_SSN),

CONSTRAINT buy\_food\_fk2 FOREIGN KEY (food\_num) REFERENCES Food(food\_num));

create table Supplier (

Vendor\_Num number(11) not null,

Name varchar2(20) not null,

Tel varchar2(15) not null,

City varchar2(15),

Road\_Name varchar2(30),

Detail varchar2(50),

Account varchar2(20) not null,

constraint Supplier\_pk primary key (Vendor\_Num)

);

create table Item (

Item\_Num number(11) not null,

Name varchar2(20) not null,

Price number(8,0) not null,

Manufacturer varchar2(20) not null,

Location varchar2(20) not null,

constraint Item\_pk primary key (Item\_Num)

);

create table Item\_order (

Vendor\_Num number(11) not null,

Item\_Num number(11) not null,

order\_date date not null,

Quantity number(3,0) not null,

TotalPrice number(10,0),

constraint Item\_order\_pk primary key (Vendor\_Num, Item\_Num),

constraint Item\_order\_fk1 foreign key (Vendor\_Num) references Supplier (Vendor\_Num),

constraint Item\_order\_fk2 foreign key (Item\_Num) references Item (Item\_Num)

);

create table Food\_Item (

F\_Item\_Num number(11) not null,

ShelfLife date,

constraint Food\_Item\_pk primary key (F\_Item\_Num),

constraint Food\_Item\_fk foreign key (F\_Item\_Num) references Item (Item\_Num)

);

create table Bring\_Food (

Food\_Num number(3,0) not null,

F\_Item\_Num number(11) not null,

bring\_date date not null,

Quantity number(3,0) not null,

constraint Bring\_Food\_pk primary key (Food\_Num, F\_Item\_Num),

constraint Bring\_Food\_fk1 foreign key (F\_Item\_Num) references Food\_Item (F\_Item\_Num),

constraint Bring\_Food\_fk2 foreign key (Food\_Num) references Food (Food\_Num)

);

create table computer\_Item (

C\_Item\_Num number(11) not null,

SerialNum varchar2(10) not null,

constraint c\_Item\_pk primary key (C\_Item\_Num),

constraint c\_Item\_fk foreign key (C\_Item\_Num) references Item (Item\_Num)

);

create table Change\_Computer(

Computer\_Num number(2,0) not null,

C\_Item\_Num number(11) not null,

change\_date date,

constraint change\_computer\_pk primary key (Computer\_Num, C\_Item\_Num),

constraint change\_computer\_fk foreign key (C\_Item\_Num) references Computer\_Item (C\_Item\_Num));

/\* person \*/

insert into person values (9912132000000, 'seoul', 'milkyro 23', '502', 'subin', 'cho', '01019991213', '1999-12-13');

insert into person values (9510121000000, 'seoul', 'dbro 7', '302', 'gildong', 'hong', '01023112413', '1995-10-12');

insert into person values (9711102000000, 'seoul', 'milkyro 3', '107', 'sua', 'kim', '01091238102', '1997-11-10');

insert into person values (9810131000000, 'seoul', 'milkyro 20', '101', 'hodong', 'kim', '01023129101', '1998-10-13');

insert into person values (9307091000000, 'seoul', 'dbro 1', '105', 'hoyeol', 'han', '01033581923', '1993-07-09');

insert into person values (9001111000000, 'seoul', 'milkyro 4', '301', 'sinyeoung', 'lee', '01099885784', '1990-01-11');

insert into person values (9303091000000, 'seoul', 'milkyro 8', '508', 'cheho', 'seo', '01010045812', '1993-03-09');

insert into person values (9605031000000, 'seoul', 'dbro 22', '601', 'hojin', 'cho', '01028946309', '1996-05-03');

insert into person values (9708221000000, 'seoul', 'dbro 23', '701', 'minho', 'kim', '01077520101', '1997-08-22');

insert into person values (9406292000000, 'seoul', 'milkyro 3', '202', 'jihye', 'lee', '01012679912', '1994-06-29');

insert into person values (9103171000000, 'seoul', 'milkyro 15', '108', 'hansu', 'kim', '01088283912', '1991-03-17');

insert into person values (9605052000000, 'seoul', 'dbro 19', '301', 'suyeon', 'han', '01074178213', '1996-05-05');

insert into person values (9808092000000, 'seoul', 'milkyro 19', '1203', 'hyejin', 'kim', '01024124133', '1998-08-09');

insert into person values (9712201000000, 'seoul', 'udongro 23', '1106', 'mujin', 'lee', '01023125885', '1997-12-20');

insert into person values (9912251000000, 'seoul', 'udongro 10', '511', 'lohan', 'lee', '01078991320', '1999-12-25');

insert into person values (9502181000000, 'seoul', 'milkyro 20', '309', 'hosu', 'bae', '01030041241', '1995-02-18');

insert into person values (9809272000000, 'seoul', 'milkyro 3', '708', 'minhye', 'jang', '01094104628', '1998-09-27');

insert into person values (9810301000000, 'seoul', 'dbro 7', '312', 'jinho', 'lee', '01014119125', '1998-10-30');

select \* from person;

/\* part-timer \*/

insert into part\_timer values (9912132000000, 58, 'weekend morning');

insert into part\_timer values (9510121000000, 80, 'weekend afternoon');

insert into part\_timer values (9711102000000, 120, 'weekday morning');

insert into part\_timer values (9810131000000, 150, 'weekday afternoon');

select \* from part\_timer;

/\* salary \*/

insert into salary values (9912132000000, 44150201289993, 580000);

insert into salary values (9510121000000, 10239481920301, 880000);

insert into salary values (9711102000000, 12391238101923, 1300000);

insert into salary values (9810131000000, 78988410123912, 1700000);

select \* from salary;

/\* customer \*/

insert into customer values (9307091000000, 'sdwals28', 'ghdne9194', '56-34m', '2h-11m');

insert into customer values (9001111000000, 'dhewa1948', 'dsceu019', '12h-14m', '0h-12m');

insert into customer values (9303091000000, 'ubvknn5593', 'mbjfs3939', '06h-04m', '22h-56m');

insert into customer values (9605031000000, 'dkfnb9103', 'flvmf159', '16h-53m', '5h-55m');

insert into customer values (9708221000000, 'rlaal192', 'gmsm88882', '47h-33m', '8h-23m');

insert into customer values (9406292000000, 'dkssums23', 'gkdlrs990', '102h-19m', '50h-51m');

insert into customer values (9103171000000, 'wlsfl389', 'dlzk1908', '89h-32m', '00h-53m');

insert into customer values (9605052000000, 'sdwa3838', 'ghne9194', '78h-04m', '01h-00m');

insert into customer values (9808092000000, 'dksl475', 'rmaksg168', '12h-34m', '01h-44m');

insert into customer values (9712201000000, 'djwi5746', 'jghru7774', '08h-54m', '2h-16m');

insert into customer values (9912251000000, 'tptld593', 'sdwikj12', '06h-03m', '2h-11m');

insert into customer values (9502181000000, 'dgkrdp54', 'dkslgo57', '05h-12m', '12h-35m');

insert into customer values (9809272000000, 'lekgod6', 'rjdmln98', '00h-34m', '3h-59m');

insert into customer values (9810301000000, 'rmxl345', 'gkem5934', '88h-35m', '23h-53m');

select \* from customer;

/\* food \*/

insert into FOOD values (1, '치킨마요덮밥', '밥류', 5500);

insert into FOOD values (2, '육개장', '컵라면', 1200);

insert into FOOD values (3, '미숫가루', '음료수', 3500);

insert into FOOD values (4, '코카콜라', '음료수', 1000);

insert into FOOD values (5, '초코스타치노', '음료수', 3000);

insert into FOOD values (6, '안성탕면', '라면', 3000);

insert into FOOD values (7, '참깨라면', '컴라면', 1800);

insert into FOOD values (8, '튀김우동', '컵라면', 1800);

insert into FOOD values (9, '블랙보리', '음료수', 2000);

insert into FOOD values (10, '포카리스웨트', '음료수', 1500);

insert into FOOD values (11, '딸기요거트', '음료수', 3500);

insert into FOOD values (12, '조지아', '음료수', 1000);

insert into FOOD values (13, '2%부족할때', '음료수', 1000);

insert into FOOD values (14, '구운감자', '과자류', 1000);

insert into FOOD values (15, '숏다리', '과자류', 1500);

select \* from food;

/\* buy food \*/

insert into BUY\_FOOD values (9712201000000, 2, 1, '2021-11-17', 1200);

insert into BUY\_FOOD values (9810301000000, 3, 2, '2021-12-10', 7000);

insert into BUY\_FOOD values (9303091000000, 8, 1, '2021-09-01', 1800);

insert into BUY\_FOOD values (9605031000000, 10, 1, '2021-04-23', 1500);

insert into BUY\_FOOD values (9708221000000, 1, 1, '2021-11-22', 5500);

insert into BUY\_FOOD values (9406292000000, 5, 1, '2021-11-22', 3000);

insert into BUY\_FOOD values (9103171000000, 7, 1, '2021-12-10', 1800);

insert into BUY\_FOOD values (9605052000000, 8, 1, '2021-11-18', 1800);

insert into BUY\_FOOD values (9808092000000, 4, 1, '2021-07-01', 1000);

insert into BUY\_FOOD values (9712201000000, 6, 1, '2021-01-10', 3000);

select \* from buy\_food;

/\* set\_menu \*/

insert into SET\_MENU values (11, 1, 4700, '딸기요거트+마약핫도그');

insert into SET\_MENU values (1, 2, 7500, '치킨마요덮밥+매콤달달라볶이');

insert into SET\_MENU values (4, 3, 4000, '짜파게티+코카콜라');

insert into SET\_MENU values (14, 4, 3000, '구운감자+복숭아아이스티');

insert into SET\_MENU values (7, 5, 2500, '참께라면+볶음김치');

select \* from set\_menu;

/\* additional\_menu \*/

insert into ADDITIONAL\_MENU values (11, 1, '마약핫도그');

insert into ADDITIONAL\_MENU values (1, 2, '매콤달달 라볶이');

insert into ADDITIONAL\_MENU values (4, 3, '짜파게티');

insert into ADDITIONAL\_MENU values (14, 4, '복숭아아이스티');

insert into ADDITIONAL\_MENU values (7, 5, '볶음김치');

/\* computer \*/

insert into computer values(1, 'i7-7700', 16, 'gtx-2060', 'k640', 'g10', 'n500');

insert into computer values(2, 'i7-7700', 16, 'gtx-2060', 'k640', 'g10', 'n500');

insert into computer values(3, 'i7-7700', 16, 'gtx-2060', 'k640', 'g10', 'n500');

insert into computer values(4, 'i7-7700', 16, 'gtx-2060', 'k640', 'g10', 'n500');

insert into computer values(5, 'i7-7700', 16, 'gtx-2060', 'k640', 'g10', 'n500');

insert into computer values(6, 'i7-7700', 16, 'gtx-2060', 'k640', 'g10', 'n500');

insert into computer values(7, 'i7-7700', 16, 'gtx-2060', 'k640', 'g10', 'n500');

insert into computer values(8, 'i7-7700', 16, 'gtx-2060', 'k640', 'g10', 'n500');

insert into computer values(9, 'i7-7700', 16, 'gtx-2060', 'k640', 'g10', 'n500');

insert into computer values(10, 'i7-7700', 16, 'gtx-2060', 'k640', 'g10', 'n500');

insert into computer values(11, 'i7-7700', 16, 'gtx-2080', 'k640', 'g10', 'n500');

insert into computer values(12, 'i7-7700', 16, 'gtx-2080', 'k640', 'g10', 'n500');

insert into computer values(13, 'i7-7700', 16, 'gtx-2080', 'k640', 'g10', 'n500');

insert into computer values(14, 'i7-7700', 16, 'gtx-2080', 'k640', 'g10', 'n500');

insert into computer values(15, 'i7-7700', 16, 'gtx-2080', 'k640', 'g10', 'n500');

insert into computer values(16, 'i7-7700', 16, 'gtx-2080', 'k640', 'g10', 'n500');

insert into computer values(17, 'i7-7700', 16, 'gtx-2080', 'k640', 'g10', 'n500');

insert into computer values(18, 'i7-7700', 16, 'gtx-2080', 'k640', 'g10', 'n500');

insert into computer values(19, 'i7-7700', 16, 'gtx-2080', 'k640', 'g10', 'n500');

insert into computer values(20, 'i7-7700', 16, 'gtx-2080', 'k640', 'g10', 'n500');

select \* from computer;

/\* computer\_monitor \*/

insert into computer\_monitor values(1, 'TFG27F07V 1500R');

insert into computer\_monitor values(2, 'TFG27F07V 1500R');

insert into computer\_monitor values(3, 'TFG27F07V 1500R');

insert into computer\_monitor values(4, 'TFG27F07V 1500R');

insert into computer\_monitor values(5, 'TFG27F07V 1500R');

insert into computer\_monitor values(6, 'TFG27F07V 1500R');

insert into computer\_monitor values(7, 'TFG27F07V 1500R');

insert into computer\_monitor values(8, 'TFG27F07V 1500R');

insert into computer\_monitor values(9, 'TFG27F07V 1500R');

insert into computer\_monitor values(10, 'TFG27F07V 1500R');

insert into computer\_monitor values(11, 'X27F-240');

insert into computer\_monitor values(12, 'X27F-240');

insert into computer\_monitor values(13, 'X27F-240');

insert into computer\_monitor values(14, 'X27F-240');

insert into computer\_monitor values(15, 'X27F-240');

insert into computer\_monitor values(16, 'X27F-240');

insert into computer\_monitor values(17, 'X27F-240');

insert into computer\_monitor values(18, 'X27F-240');

insert into computer\_monitor values(19, 'X27F-240');

insert into computer\_monitor values(20, 'X27F-240');

insert into computer\_monitor values(11, 'V28UE');

insert into computer\_monitor values(12, 'V28UE');

insert into computer\_monitor values(13, 'V28UE');

insert into computer\_monitor values(14, 'V28UE');

insert into computer\_monitor values(15, 'V28UE');

insert into computer\_monitor values(16, 'V28UE');

insert into computer\_monitor values(17, 'V28UE');

insert into computer\_monitor values(18, 'V28UE');

insert into computer\_monitor values(19, 'V28UE');

insert into computer\_monitor values(20, 'V28UE');

select \* from computer\_monitor;

/\* use\_computer \*/

insert into use\_computer values (9001111000000, 11 , '2021-11-06', 5, 5000);

insert into use\_computer values (9001111000000, 10 , '2021-11-06', 20, 20000);

insert into use\_computer values (9708221000000, 15 , '2021-11-07', 10, 10000);

insert into use\_computer values (9708221000000, 3 , '2021-11-08', 3, 3000);

insert into use\_computer values (9708221000000, 7 , '2021-11-09', 3, 3000);

insert into use\_computer values (9605052000000, 9 , '2021-11-10', 4, 4000);

insert into use\_computer values (9605052000000, 10 , '2021-11-11', 8, 8000);

insert into use\_computer values (9808092000000, 11, '2021-11-11', 7, 7000);

insert into use\_computer values (9103171000000, 12, '2021-11-12', 10, 10000);

insert into use\_computer values (9502181000000, 20, '2021-11-12', 5, 5000);

insert into use\_computer values (9810301000000, 7, '2021-11-13', 7, 7000);

insert into use\_computer values (9809272000000, 9 , '2021-11-13', 8, 8000);

insert into use\_computer values (9406292000000, 1 , '2021-11-13', 3, 3000);

insert into use\_computer values (9303091000000, 14 , '2021-11-14', 2, 2000);

insert into use\_computer values (9001111000000, 17, '2021-11-15', 1, 1000);

insert into use\_computer values (9001111000000, 19, '2021-11-16', 5, 5000);

insert into use\_computer values (9406292000000, 7 , '2021-11-22', 8, 8000);

insert into use\_computer values (9912251000000, 8 , '2021-11-25', 10, 10000);

insert into use\_computer values (9502181000000, 10 , '2021-11-26', 20, 20000);

insert into use\_computer values (9809272000000, 11 , '2021-11-26', 9, 9000);

insert into use\_computer values (9809272000000, 15 , '2021-11-26', 7, 7000);

insert into use\_computer values (9810301000000, 19 , '2021-11-26', 4, 4000);

insert into use\_computer values (9810301000000, 20 , '2021-11-27', 6, 6000);

insert into use\_computer values (9303091000000, 9 , '2021-11-28', 1, 1000);

insert into use\_computer values (9303091000000, 7 , '2021-11-29', 2, 2000);

insert into use\_computer values (9605031000000, 12 , '2021-11-29', 5, 5000);

insert into use\_computer values (9001111000000, 13, '2021-11-30', 5, 5000);

insert into use\_computer values (9001111000000, 14, '2021-11-30', 10, 10000);

insert into use\_computer values (9303091000000, 19 , '2021-12-6', 1, 1000);

insert into use\_computer values (9303091000000, 17 , '2021-12-10', 2, 2000);

insert into use\_computer values (9605031000000, 2 , '2021-12-12', 5, 5000);

insert into use\_computer values (9001111000000, 3, '2021-12-13', 5, 5000);

insert into use\_computer values (9001111000000, 4, '2021-12-14', 10, 10000);

select \* from use\_computer;

/\* supplier \*/

insert into supplier values (1, '커피몰', '1544-0807', 'Asan','Eumbongro 681', '28', '483051-00-122220');

insert into supplier values (2, '건우유통', '1899-3400', 'Siheung','Beomanro 390', '38-2', '100-029-332220');

insert into supplier values (3, '피카', '1544-2552', 'Seoul','Gasan Digital1ro 149','신한이노플렉스 13층', '206441-00-111111');

insert into supplier values (4, '인텍앤컴퍼니', '1544-7824', 'Seoul','Hackdongro 155', '원영빌딩 5층', '1002-155-1222250');

select \* from supplier;

/\* item \*/

insert into item values (1,'치킨패티', 3250, '새롬에프에스', '냉동실1');

insert into item values (2,'육개장', 850, '농심', '진열대1');

insert into item values (3,'미숫가루', 4120, '배대감', '파우더선반');

insert into item values (4,'우유', 1250, '매일', '냉장고1');

insert into item values (5,'초코파우더', 8000, '네이쳐티', '파우더선반');

insert into item values (6,'바닐라파우더', 8800, '까로망', '파우더선반');

insert into item values (7,'코카콜라', 1200, '코카콜라', '냉장고2');

insert into item values (8,'안성탕면', 480, '농심', '진열대2');

insert into item values (9,'참깨라면', 900, '오뚜기', '진열대1');

insert into item values (10,'튀김우동', 910, '농심', '진열대1');

insert into item values (11,'블랙보리', 750, '하이트진로', '냉장고2');

insert into item values (12,'포카리스웨트', 580, '동아오츠카', '냉장고2');

insert into item values (13,'햇반', 1000, 'CJ', '진열대2');

insert into item values (14,'k640', 30000, '해커', '부품창고');

insert into item values (15,'g102', 12000, '로지텍', '부품창고');

insert into item values (16,'i7-7700', 400000, '인텔', '부품창고');

insert into item values (17,'gtx-2060', 1500000, '기가바이트', '부품창고');

insert into item values (18,'gtx-2080', 2000000, '기가바이트', '부품창고');

select \* from item;

/\* item\_order \*/

insert into item\_order values (2, 1, '2021-11-01', 5, 16250);

insert into item\_order values (2, 3, '2021-11-06', 2, 8240);

insert into item\_order values (1, 4, '2021-11-08', 1, 1250);

insert into item\_order values (1, 5, '2021-11-12', 2, 16000);

insert into item\_order values (1, 6, '2021-11-18', 2, 17600);

insert into item\_order values (2, 9, '2021-11-20', 12, 10800);

insert into item\_order values (2, 10, '2021-11-25', 12, 10920);

insert into item\_order values (4, 18, '2021-12-01', 1, 2000000);

select \* from item\_order;

/\* food\_item \*/

insert into food\_item values (1, '2022-09-12');

insert into food\_item values (2, '2022-01-10');

insert into food\_item values (3, '2022-10-21');

insert into food\_item values (4, '2021-12-15');

insert into food\_item values (5, '2022-12-22');

insert into food\_item values (6, '2022-12-30');

insert into food\_item values (7, '2022-07-01');

insert into food\_item values (8, '2022-01-30');

insert into food\_item values (9, '2022-02-22');

insert into food\_item values (10, '2022-02-04');

insert into food\_item values (11, '2022-09-18');

insert into food\_item values (12, '2022-06-17');

insert into food\_item values (13, '2022-01-30');

select \* from food\_item;

/\* bring\_food \*/

insert into bring\_food values (1, 1, '2021-11-10', 5);

insert into bring\_food values (2, 2, '2021-11-14', 6);

insert into bring\_food values (3, 4, '2021-12-22', 2);

insert into bring\_food values (4, 7, '2021-10-30', 1);

insert into bring\_food values (5, 6, '2021-09-01', 2);

insert into bring\_food values (6, 8, '2021-11-11', 10);

insert into bring\_food values (7, 9, '2021-10-26', 5);

insert into bring\_food values (8, 10, '2021-11-17', 2);

insert into bring\_food values (9, 11, '2021-12-04', 5);

insert into bring\_food values (10, 12, '2021-10-31', 8);

select \* from bring\_food;

1. DML(query)

/\*

1. Food item에서 유통기한 지난 재고 구하기

🡪 유통기한 지난 재고를 분류해서 버리기 위한 쿼리

\*/

select \*

from item, food\_item

where item.item\_num = food\_item.f\_item\_num

and shelfLife < sysdate;

/\*

1. Food 주문량에 따른 인기 음식 5개 선정하기

🡪 인기있는 음식의 재고를 더 많이 주문하기 위한 쿼리

\*/

select name

from food

where food\_num in (

select food\_num

from(

select food\_num

from buy\_food

group by food\_num

order by sum(quantity) desc

)

where rownum <=5);

/\*

1. 지난 달 가장 출석을 많이 한 고달객 5명의 정보 출력

🡪 피시방을 자주 이용하는 고객에 대한 감사이벤트

\*/

select first\_name, last\_name

from person

where SSN in (select C\_SSN

from(

select C\_SSN

from use\_computer

where use\_date < '2021-12-01'

and use\_date >= '2021-11-01'

group by C\_SSN

order by count(use\_date) desc

)where rownum <=5);

/\*

1. 해당일자에 팔린 음식의 총 판매금액

🡪 평일, 주말, 공휴일에 따른 음식 판매 매출을 분석하기 위한 쿼리

\*/

select sum(Total\_Price)

from buy\_food

group by buy\_date

having buy\_date = '2021-11-22';

/\*

1. 월 60시간 이상(주휴 시간) 근무한 part-timer 구하기

🡪 월 60시간 이상 근무한 part-timer에게 주휴수당을 지급하기 위한 쿼리

\*/

select first\_name, last\_name

from person, part\_timer

where person.SSN = part\_timer.P\_SSN

and working\_hour >= 60;

/\*

1. 듀얼 모니터인 컴퓨터 번호

🡪 듀얼모니터를 사용하는 컴퓨터를 관리하기 위한 쿼리, 듀얼모니터가 아닌 컴퓨터에 모니터를 추가하기 위한 쿼리

\*/

select computer\_num

from computer\_monitor

group by computer\_num

having count(monitor) = 2;

/\*

1. 재고가 5개 미만인 item 구하기

🡪 재고가 5개 미만인 item의 재고를 미리 주문하기 위한 쿼리

\*/

select name

from item, item\_order

where item.item\_num = item\_order.item\_num

and quantity <5;

/\*

1. 성인이 주문한 음식의 이름, 총 개수, 음식 금액 구하기

🡪 피시방을 이용하는 성인의 음식 소비 패턴 분석을 위한 쿼리

\*/

select food.name, sum(quantity) as total\_quantity, food.price

from buy\_food, food

where C\_SSN in (

select C\_SSN

from customer, person

where customer.C\_SSN = person.SSN

and birth < '2004-01-01')

and buy\_food.food\_num = food.food\_num

group by food.name, food.price

order by total\_quantity desc;

/\*

1. 이번달에 생일이고 PC방 사용내역이 있는 customer 구하기

🡪 해당 달에 사용내역이 있고 생일인 고객에게 생일 이벤트 상품을 증정하기 위한 쿼리

\*/

select first\_name, last\_name

from person

where SSN like '\_\_12\_\_\_\_\_\_\_\_\_'

and SSN in(

select customer.C\_SSN

from customer, use\_computer

where customer.C\_SSN = use\_computer.C\_SSN);

/\*

1. 지정 달 총 음식 주문 금액이 가장 많은 customer 구하기

🡪 이번 달 음식왕에게 아메리카노를 제공하기위한 쿼리

\*/

select buy\_food.C\_SSN, person.first\_name, person.last\_name , sum(total\_price) as total\_price

from buy\_food, person

where buy\_food.C\_SSN = person.SSN

and buy\_date < '2021-12-01'

and buy\_date >= '2021-11-01'

group by buy\_food.C\_SSN, person.first\_name, person.last\_name

order by total\_price desc;

1. System manual

(1). part-timer의 working hour과 Salary의 gross pay는 해당 달에 대한 정보만 저장한다.