DATABASE DESIGN
TEAM PROJECT

Design of a confectionery Factory DB

T E A M 5

FACTORY DB

INDEX

User Requirements

User Needs

■ HR department: this database will provide HR department with employee information to evaluate them.

■ Maintenance department: maintenance department will be allowed to get information of machine inspections and repairs.

■ Procurement and sales department: Procurement and sales employees are given price and inventory information of products a nd ingredients and buyer and supplier information.

■ Production department: information of orders from buyers goes to production department for inventory control and order processing.

■ R&D department: how many products are sold is passed on to R&D department to decide which product is enhanced or which product is developed.

■ All employees: commuter bus information is provided to all employees

a **confectionery** factory

- Employee
- Supplier
- Buyer
- Machine
- Product
- Sales
- Ingredient
- Purchase
- Commuter Bus
- Refund

Enumerated

Based on user requirement

a **confectionery** factory

Person

Emloyee

name, department, position etc ..

Inventory

Product

name, stock, expiration date,

Ingredient

ingredient, price, stock

Produce

Machine

machine inspection, repair, manager, operation sheet

Transaction

Purchase

seller, price, timeslot, quantity, oder etc.

Sales

buyer, shipping address, price, order etc.

Refund

sales order, buyer product, quantity, timeslot

Company

Supplier

company name, email, contact, address

Buyer

company name, email, contact, address

a **confectionery** factory

Person

Emloyee

name, department, position etc ..

Inventory

Product

name, stock, expiration date,

Ingredient

ingredient, price, stock

Produce

Machine

machine inspection, repair, manager, operation sheet

Transaction

Purchase

seller, price, timeslot, quantity, oder etc.

Sales

buyer, shipping address, price, order etc.

Refund

sales order, buyer product, quantity, timeslot

Company

Supplier

company name, email, contact, address

Buyer

company name, email, contact, address

Key Entities

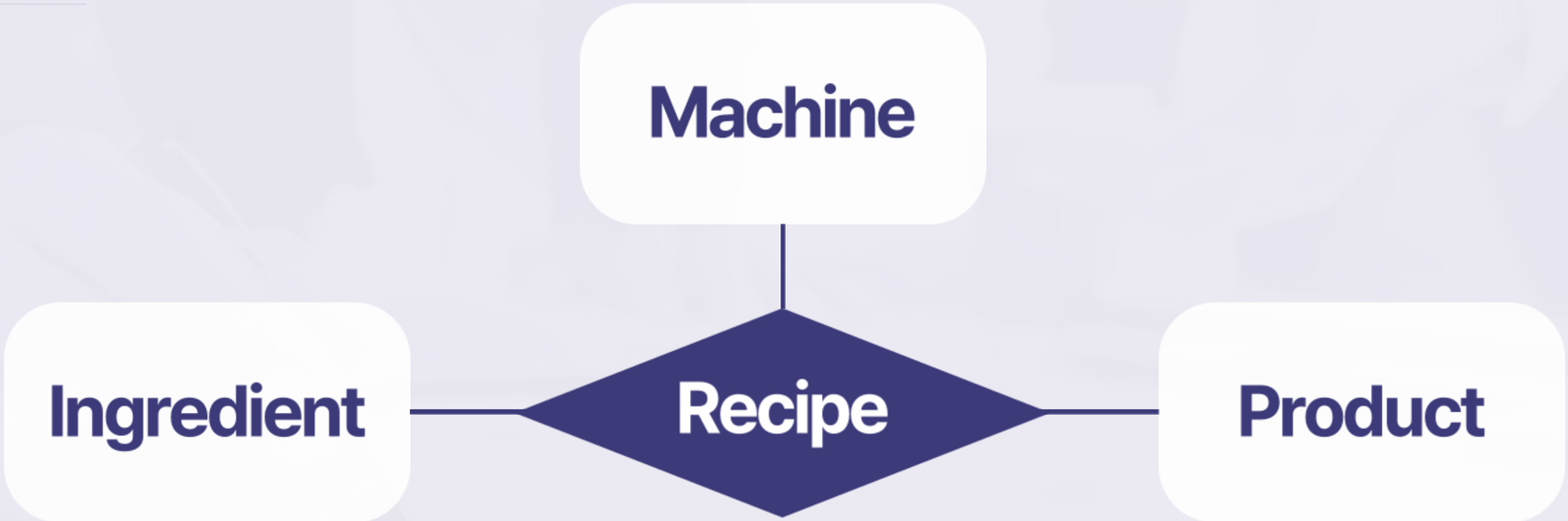
Sub entities are found from three tables

Key **Roles** of a Factory

- Produce
- Machine Operation
- Price
- Order

Produce

A product is produced with ingredients through a machine



Machine Operation

An employee operates and manages machines

Ingredient Price

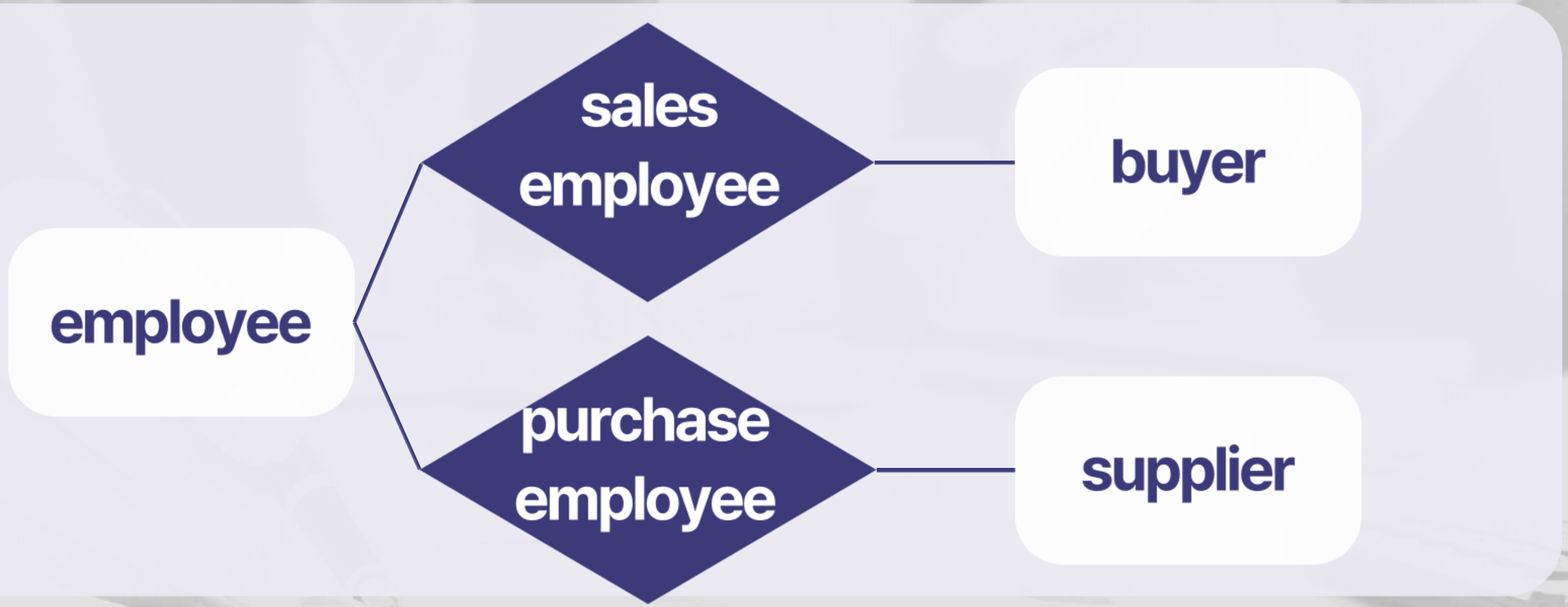
Ingredients are provided from a supplier

Product Price

Product price depends on a type of buyer

Account Management

some employees must be responsible of buyers and suppliers

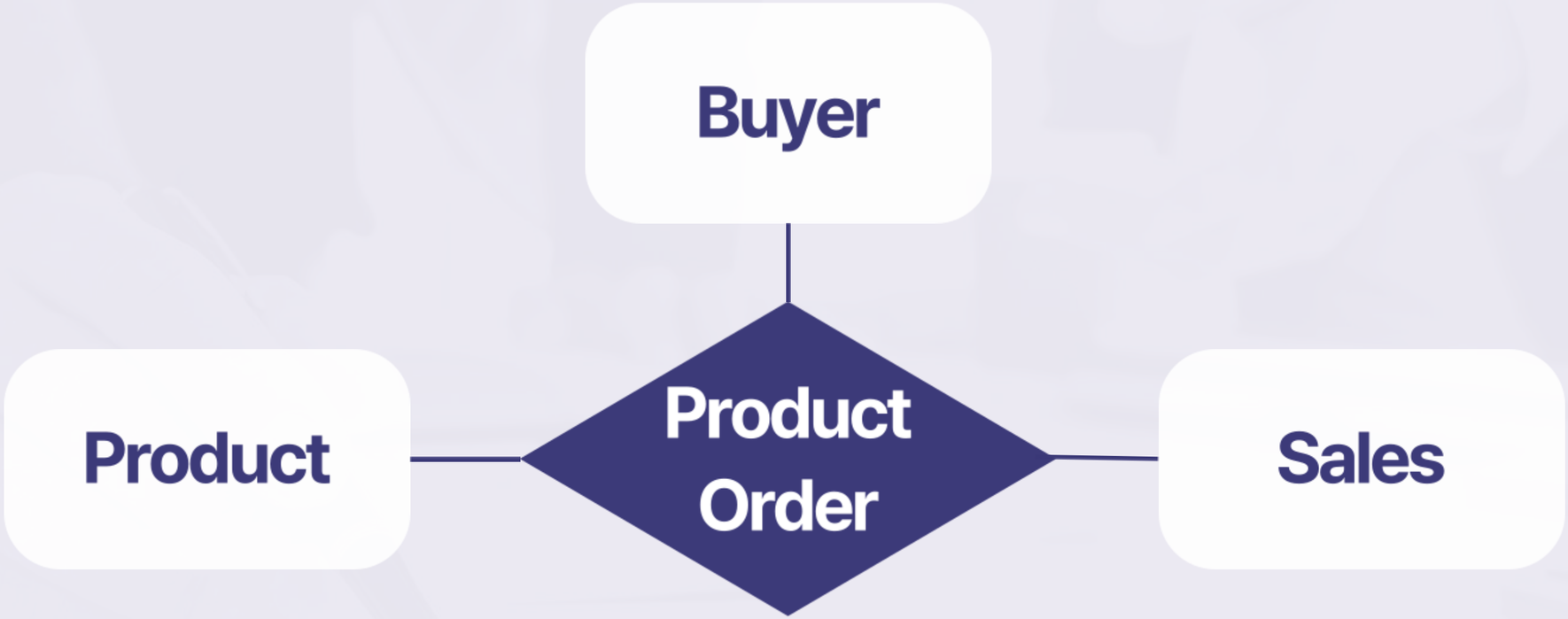


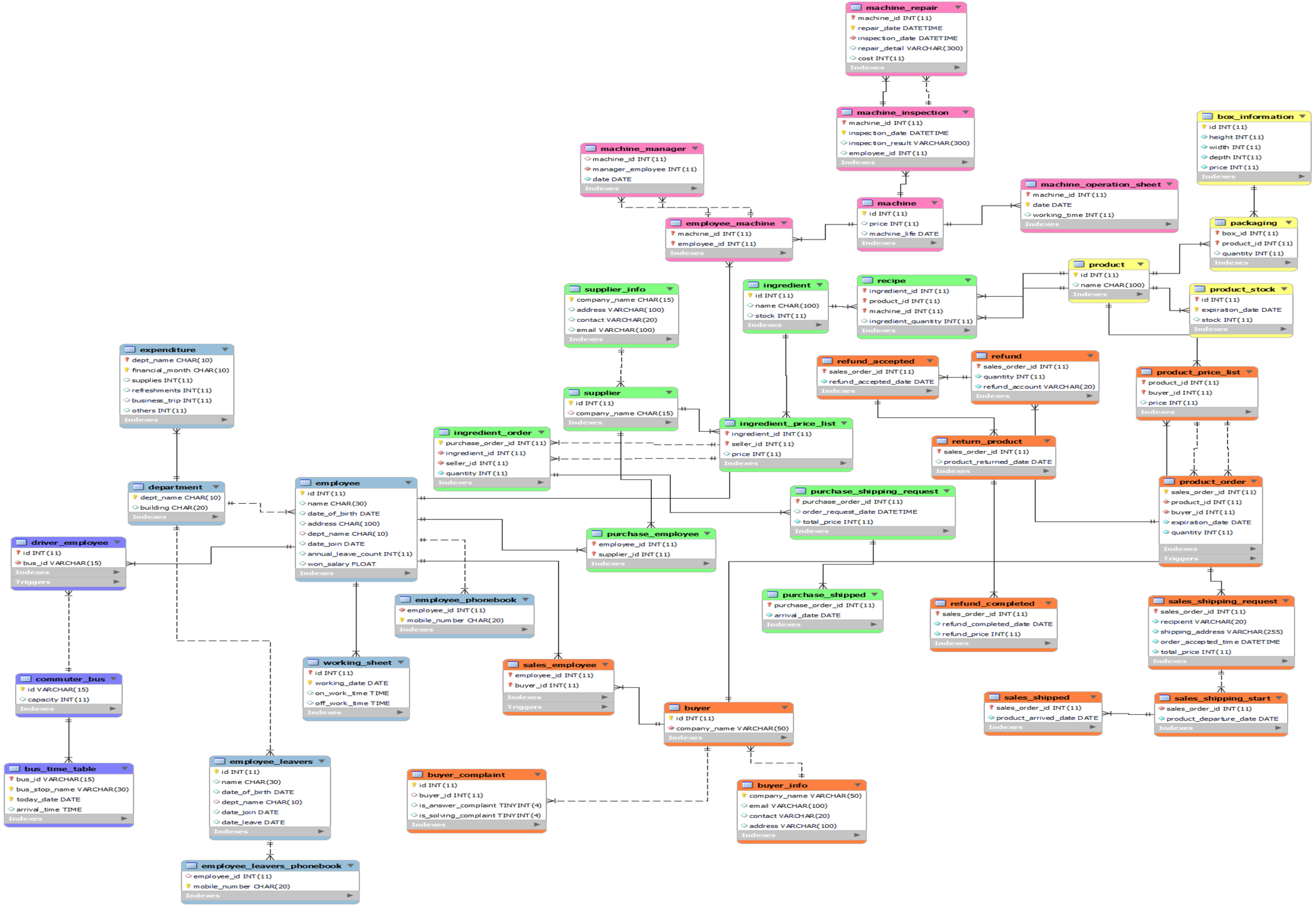
Order

It is important to output financial results

Order

It is important to output financial results




machine_repair
machine_id INT(11)
repair_date DATETIME
inspection_date DATETIME
repair_detail VARCHAR(300)
cost INT(11)
Indexes
machine_inspection
machine_id INT(11)
inspection_date DATETIME
inspection_result VARCHAR(300)
employee_id INT(11)
Indexes
box_information
id INT(11)
height INT(11)
width INT(11)
depth INT(11)
price INT(11)
Indexes
machine_manager
machine_id INT(11)
manager_employee INT(11)
date DATE
Indexes
machine
id INT(11)
price INT(11)
machine_life DATE
Indexes
machine_operation_sheet
machine_id INT(11)
date DATE
working_time INT(11)
Indexes
employee_machine
machine_id INT(11)
employee_id INT(11)
Indexes
packaging
box_id INT(11)
product_id INT(11)
quantity INT(11)
Indexes
product
id INT(11)
name CHAR(100)
Indexes
supplier_info
company_name CHAR(15)
address VARCHAR(100)
contact VARCHAR(20)
email VARCHAR(100)
Indexes
ingredient
id INT(11)
name CHAR(100)
stock INT(11)
Indexes
recipe
ingredient_id INT(11)
product_id INT(11)
machine_id INT(11)
ingredient_quantity INT(11)
Indexes
product_stock
id INT(11)
expiration_date DATE
stock INT(11)
Indexes
expenditure
dept_name CHAR(10)
financial_month CHAR(10)
supplies INT(11)
refreshments INT(11)
business_trip INT(11)
others INT(11)
Indexes
refund_accepted
sales_order_id INT(11)
refund_accepted_date DATE
Indexes
refund
sales_order_id INT(11)
quantity INT(11)
refund_account VARCHAR(20)
Indexes
product_price_list
product_id INT(11)
buyer_id INT(11)
price INT(11)
Indexes
supplier
id INT(11)
company_name CHAR(15)
Indexes
ingredient_order
purchase_order_id INT(11)
ingredient_id INT(11)
seller_id INT(11)
quantity INT(11)
Indexes
ingredient_price_list
ingredient_id INT(11)
seller_id INT(11)
price INT(11)
Indexes
return_product
sales_order_id INT(11)
product_returned_date DATE
Indexes
department
dept_name CHAR(10)
building CHAR(20)
Indexes
employee
id INT(11)
name CHAR(30)
date_of_birth DATE
address CHAR(100)
dept_name CHAR(10)
date_join DATE
annual_leave_count INT(11)
won_salary FLOAT
Indexes
purchase_shipping_request
purchase_order_id INT(11)
order_request_date DATETIME
total_price INT(11)
Indexes
product_order
sales_order_id INT(11)
product_id INT(11)
buyer_id INT(11)
expiration_date DATE
quantity INT(11)
Indexes
Triggers
driver_employee
id INT(11)
bus_id VARCHAR(15)
Indexes
Triggers
purchase_employee
employee_id INT(11)
supplier_id INT(11)
Indexes
employee_phonebook
employee_id INT(11)
mobile_number CHAR(20)
Indexes
purchase_shipped
purchase_order_id INT(11)
arrival_date DATE
Indexes
refund_completed
sales_order_id INT(11)
refund_completed_date DATE
refund_price INT(11)
Indexes
sales_shipping_request
sales_order_id INT(11)
recipient VARCHAR(20)
shipping_address VARCHAR(255)
order_accepted_time DATETIME
total_price INT(11)
Indexes
working_sheet
id INT(11)
working_date DATE
on_work_time TIME
off_work_time TIME
Indexes
sales_employee
employee_id INT(11)
buyer_id INT(11)
Indexes
Triggers
commuter_bus
id VARCHAR(15)
capacity INT(11)
Indexes
buyer
id INT(11)
company_name VARCHAR(50)
Indexes
sales_shipped
sales_order_id INT(11)
product_arrived_date DATE
Indexes
sales_shipping_start
sales_order_id INT(11)
product_departure_date DATE
Indexes
bus_time_table
bus_id VARCHAR(15)
bus_stop_name VARCHAR(30)
today_date DATE
arrival_time TIME
Indexes
employee_leavers
id INT(11)
name CHAR(30)
date_of_birth DATE
dept_name CHAR(10)
date_join DATE
date_leave DATE
Indexes
buyer_complaint
id INT(11)
buyer_id INT(11)
is_answer_complaint TINYINT(4)
is_solving_complaint TINYINT(4)
Indexes
buyer_info
company_name VARCHAR(50)
email VARCHAR(100)
contact VARCHAR(20)
address VARCHAR(100)
Indexes
employee_leavers_phonebook
employee_id INT(11)
mobile_number CHAR(20)
Indexes

Database **Functions**

function1

average_working_time()

Function that returns
the average working hours of
the entire employee

function2

department_average_salary
(DEPT_NAME)

Function that returns
the average wage
for the entered department

function3

return_ingredient_stock
(INGREDIENT)

Function that returns
the number of inventory of
the ingredient entered

function4

calculate_refund_price
(sales_order_id)

Function that calculates and returns
the refund amount by collecting
quantity information and
the price information purchased,
assuming that the return is applied at the
corresponding order number

Database **Procedures**

Procedure1

findComplaintToProcess( )

Show everything
about complaints that
have not been processed yet

Procedure2

getIngredientsLessThanN
(N INT)

Show the name and number
of inventories for materials
with fewer inventories than N

Procedure3

UpdateProductStockSales
(sales_order_id INT);

In the case of a sale,
update the contents
of the inventory

Database **Triggers**

trigger1

product_order

If the number of products you want to order is more than the stock of the product, print an error message

trigger2

sales_employee

Can be added if employee is sales team, error if not sales team

trigger3

driver_employee

Can be added if employee is driver team, error if not driver team

*Fuc_StockCheck.java

```java
package Connection;

import java.sql.*;

public class Fuc_StockCheck {
    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null; //SQL문을 데이터베이스에 보내기 위한 객체
        ResultSet rs = null; //결과 리턴하기 위한 객체

            String className = "com.mysql.cj.jdbc.Driver";
            String url = "jdbc:mysql://43.200.203.69:57751/factory"; //mysql 접근 위한 호스트와 port번호, 접근 스키마 주소
            String user = "admin"; //user id
            String passwd = "pass"; //user password
            String sql = "SELECT name, stock FROM ingredient"; //실행 sql문

            try {
                Class.forName(className); //JDBC 드라이버 로딩

                conn = DriverManager.getConnection(url, user, passwd); //Connection 객체 생성
                System.out.println("Successfully Connected!");
                stmt = conn.createStatement(); // Statement객체 생성
                rs = stmt.executeQuery(sql); // 어떠한 결과를 받아오는 ResultSet 타입의 rs 변수에 쿼리문을 실행한 결과 넣어줌

                while (rs.next()) {
                    String name = rs.getString("name");
                    String stock = rs.getString("stock");
                    int st = Integer.parseInt(stock);
                    if(st<1) System.out.println(name + "의 재고가 없습니다. 재고 확인 후 주문바랍니다.");
                    else System.out.println(name + "의 재고가 존재합니다.");
                }

            }

            catch(ClassNotFoundException e){
                System.out.println("[JDBC Connector Driver 오류 : " + e.getMessage() + "]");
            }
            catch(SQLException e){
```

Console

```
<terminated> ProgramFunct [J
Successfully Connected!
밀가루의 재고가 존재합니다.
초콜릿의 재고가 존재합니다.
소금의 재고가 존재합니다.
설탕의 재고가 존재합니다.
바나나의 재고가 존재합니다.
파슬리의 재고가 존재합니다.
꿀의 재고가 존재합니다.
버터의 재고가 존재합니다.
아몬드의 재고가 존재합니다.
크림의 재고가 존재합니다.
새우의 재고가 존재합니다.
조미료의 재고가 존재합니다.
옥수수의 재고가 존재합니다.
감자의 재고가 존재합니다.
후추의 재고가 존재합니다.
반죽의 재고가 존재합니다.
```

Navigator

SCHEMAS

Filter objects

- employee_leavers_phoneboo
- employee_machine
- employee_phonebook
- expenditure
- ingredient
  - Columns
  - Indexes
  - Foreign Keys
  - Triggers
- ingredient_order
- ingredient_price_list
- machine
- machine_inspection
- machine_manager
- machine_operation_sheet
- machine_repair
- packaging
- product
- product_order
- product_price_list

Administration | Schemas

Information

Table: ingredient

Columns:

id	int(11) PK
name	char(100)
stock	int(11)

recipe | ingredient

```
1 • SELECT * FROM
```

Result Grid | Filter Ro

id	name	stock
1	밀가루	200
2	초콜릿	100
3	소금	50
4	설탕	50
5	바나나	20
6	파슬리	5
7	꿀	50
8	버터	20
9	아몬드	100
10	크림	80
11	새우	150
12	조미료	200
13	옥수수	50
14	감자	100
15	후추	50
16	반죽	100
NULL	NULL	NULL

ingredient 1

Fun_ProductMade.java

```java
package Connection;
import java.sql.*;

public class Fun_ProductMade {
    public static void main(String[] args) {
        Connection conn = null;
        PreparedStatement pstmt = null; //SQL문을 데이터베이스에 보내기 위한 객체
        ResultSet rs = null; //결과 리턴하기 위한 객체

            String className = "com.mysql.cj.jdbc.Driver";
            String url = "jdbc:mysql://43.200.203.69:57751/factory"; //mysql 접근 위한 호스트와 port번호, 접근 스키마 주소
            String user = "admin"; //user id
            String passwd = "pass"; //user password

            try {
                Class.forName(className); //JDBC 드라이버 로딩
                conn = DriverManager.getConnection(url, user, passwd); //Connection 객체 생성
                System.out.println("Successfully Connected!");

                StringBuilder sb = new StringBuilder();
                sb.append("SELECT recipe.ingredient_id, recipe.product_id, recipe.machine_id, product.name, product_stock.stock ");
                sb.append("FROM recipe join product join product_stock ");
                sb.append("WHERE recipe.product_id = product.id && recipe.product_id = product_stock.id");
                pstmt = conn.prepareStatement(sb.toString());

                rs = pstmt.executeQuery(); // 어떠한 결과를 받아오는 ResultSet 타입의 rs 변수에 쿼리문을 실행한 결과 넣어줌
                while (rs.next()) {
                    String ing = rs.getString("recipe.ingredient_id");
                    String id = rs.getString("recipe.product_id");
                    String machine = rs.getString("recipe.machine_id");
                    String name = rs.getString("product.name");
                    String stock = rs.getString("product_stock.stock");
                    int st = Integer.parseInt(stock);
                    if(st<1) {
                        System.out.println(name + "의 재고가 없습니다. ingredient " + ing + "로 기계" + machine + "에게 생산을 요청합니다.");
                        continue;
                    }
                }
            }

            catch(ClassNotFoundException e){
                System.out.println("[JDBC Connector Driver 오류 : " + e.getMessage() + "]");
            }
```

Console

```
<terminated> Fun_ProductMade [Java Application] C:₩Users₩USER₩.p2₩p
Successfully Connected!
알새우칩의 재고가 없습니다. ingredient 1로 기계2에게 생산을 요청합니다.
알새우칩의 재고가 없습니다. ingredient 3로 기계2에게 생산을 요청합니다.
알새우칩의 재고가 없습니다. ingredient 11로 기계3에게 생산을 요청합니다.
알새우칩의 재고가 없습니다. ingredient 13로 기계2에게 생산을 요청합니다.
```

	id	expiration_date	stock
▶	1	2023-12-04	80
	1	2023-12-14	400
	2	2023-05-04	20
	2	2023-07-04	150
	3	2024-02-01	5
	4	2021-11-30	5
	4	2023-02-27	100
	5	2024-08-20	500
	6	2022-12-25	10
	7	2020-07-20	20
	8	2020-12-13	30
	9	2022-04-08	0
	10	2023-02-08	40
	10	2023-03-03	100
	10	2023-05-24	300
*	NULL	NULL	NULL

Fun_Prod.java

```
8         Connection conn = null;
9         PreparedStatement pstmt = null; //SQL문을 데이터베이스에 보내기 위한 객체
10        Statement stmt = null;
11        ResultSet rs = null; //결과 리턴하기 위한 객체
12
13            String className = "com.mysql.cj.jdbc.Driver";
14            String url = "jdbc:mysql://43.200.203.69:57751/factory"; //mysql 접근 위한 호스트와 port번호, 접근 스키마 주소
15            String user = "admin"; //user id
16            String passwd = "pass"; //user password
17
18            try {
19                Class.forName(className); //JDBC 드라이버 로딩
20                conn = DriverManager.getConnection(url, user, passwd); //Connection 객체 생성
21                System.out.println("Successfully Connected!");
22
23                System.out.println("어떤 product를 만드시겠습니까?(id로 답변) ");
24                int pid = sc.nextInt();
25                System.out.println("최종 재고량을 얼마로 하시겠습니까?(기존개수를 꼭 더해주세요) ");
26                int pcnt = sc.nextInt();
27                String sql1 = "update product_stock set stock=? where id=?";
28
29                pstmt = conn.prepareStatement(sql1);
30
31                pstmt.setInt(1, pcnt);
32                pstmt.setInt(2, pid);
33
34                int r = pstmt.executeUpdate();
35                System.out.println("변경된 row : " + r);
36
37                String sql2 = "select * from product_stock";
38                stmt = conn.createStatement();
39                rs = stmt.executeQuery(sql2);
40                System.out.println("<product_stock table>");
41                while (rs.next()) {
42
43                    String id = rs.getString("id");
44                    String ex_date = rs.getString("expiration_date");
45                    String stock= rs.getString("stock");
46
47                    System.out.println(id + " " + ex_date + " " + stock);
48                }
```

Console

```
<terminated> Fun_Prod [Java Application] C:₩Users₩US
Successfully Connected!
어떤 product를 만드시겠습니까?(id로 답변)
6
최종 재고량을 얼마로 하시겠습니까?(기존개수를 꼭 더해주세요)
120
변경된 row : 1
<product_stock table>
1 2023-12-04 80
1 2023-12-14 400
2 2023-05-04 20
2 2023-07-04 150
3 2024-02-01 5
4 2021-11-30 5
4 2023-02-27 100
5 2024-08-20 500
6 2022-12-25 120
7 2020-07-20 20
8 2020-12-13 30
9 2022-04-08 0
10 2023-02-08 40
10 2023-03-03 100
10 2023-05-24 300
```

Thank you

TEAM 5

김가연(20206319)
김벼리(20191397)
박영재(20196687)
이승아(20203189)
장혜정(20203524)