



Master in Smart Industry (MIC)

TRABAJO FIN DE MASTER

SMART SYSTEM TECHNOLOGIES APPLIED TO
THE FIELD OF CIRCULAR ECONOMICS TO
REGENERATE RAW MATERIALS FOR
PACKAGING INDUSTRIES COLLECTED AS
WASTE FROM ONLINE ORDERS DELIVERY.

Author: Ragul Manoharan

Director: Bernardo Villazan Gil

Madrid

August 2021

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título “SMART SYSTEM TECHNOLOGIES APPLIED TO THE FIELD OF CIRCULAR ECONOMICS TO REGENERATE RAW MATERIALS FOR PACKAGING INDUSTRIES COLLECTED AS WASTE FROM ONLINE ORDERS DELIVERY”

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2020-21 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.



Fdo.: (Ragul Manoharan)

Fecha: 26/August/21

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO



Fdo.: (Bernardo Villazán)

Fecha: 31/August/21

Abstract

This paper discusses about how we can apply Smart Industry technologies to circular economics. This research basically aims to reduce extraction of non-renewable resources continuously which leads to depletion of a resource and creates scarcity. The approach of how we dispose a product after its use have immensely contributed to huge CO₂ footprint, thus increasing greenhouse effects and global warming. When we apply the concept of circular economics joining hands with Smart Industry technology, we can easily deal in reducing CO₂ footprint. This is achieved by using principles of Circular Economics such as upcycling a product/goods after its use to serve a new purpose. This paper is basically designed to consider cardboard boxes which is used to fulfill online order deliveries, after fulfilling the purpose of delivering the ordered goods, the packages can either be upcycled, refurbished, biomimicry or can be used as raw material for other industries like packaging. Applying the concept of “How Smart systems can be applied to Industries” to the cradle-to-cradle thinking instead of cradle-to-grave approach will yield not only revenues and employments to but also helps us to create a more sustainable cities. This research in turn enables to achieve Sustainable development goals such as alleviating poverty, Industrial Innovation and Economic growth of an individual and country.

Keywords

Circular Economics, Smart systems, Packaging, Global Warming, Upcycling, CO₂ footprint, cradle-to-cradle, cradle-to-grave, Sustainability.

1. Introduction

In our day-to-day life we use everything in a linear economic way that is in simple words we Take - Use - dispose - Incinerate. India with a population 1.3 billion generates highest quantity of waste which in turn contributes to harmful gas emissions thus depleting ozone and the quality of life. These CO₂ emissions have doubled the death rates in 20 years due to Air pollution [20].

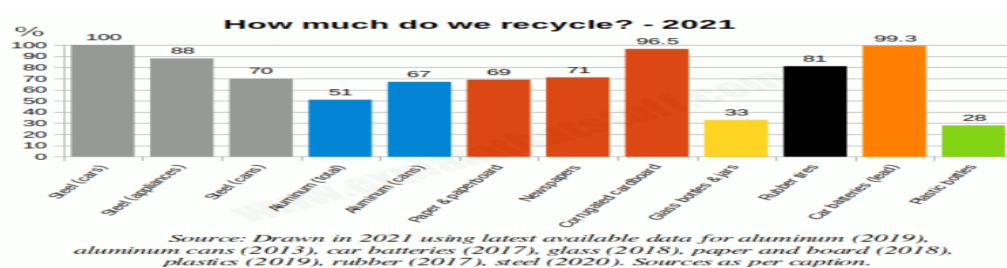
This causes a lot of energy spent as well as we leave a huge CO₂ footprint on the planet which eventually cause global warming and green house effects. The main motto of me choosing this topic as my project is to make all linear life cycle product into a circular way: Take - make - use - reuse [1].

This project is going to focus on making the packaging boxes (linear) into a circular economic process that is once the customer orders something from an online store (Basically a laptop), he has the choice of choosing custom cardboard boxes (large, medium, and small) which then deposited into the green bins from where it goes to a regeneration unit. From the regeneration unit the cardboard boxes can be used as a raw material in different industry like paper, plywood etc as well as repacked or refurbished again if it is not damaged much. For consideration and for easy understanding the number of companies, trucks, bin are kept as one.

This system relies on customer input in buying the product and selecting the package type, after that everything happens automatically. The Customer will deposit the package in a specified green bin (once the bin is full it intimates the collection truck then this truck will reach the green company from there it is sent to regeneration unit.

Technically this research have used MySQL database where DDL and DML commands are used (MySQL Workbench is used to design DDL and DML commands Entity relation design is obtained through reverse engineering, MySQL command line prompt) to monitor the customer shopping, updating and deleting the customer details and orders respectively, selecting packages once the customer deposits the package in the bin, the counter starts and then once the full capacity of the bin is filled then that is also monitored through DB.

The DB along with CPP is used to integrate into the system and to communicate to the physical world. The main pillar of strength consider in this project is that having our world in mind to reduce temperature rise and its associated causes through a technological way.



Recycling signifies incineration of cardboard boxes contributes to 96.5% of recycling India which in turn contributes to CO₂ emissions.

Emission Category (million metric tons CO ₂ e)	2018	2019	2020	YoY%
Emissions from Direct Operations (Scope 1)	4.98	5.76	9.62	+67%
Fossil fuels	4.70	5.57	9.37	+69%
Refrigerants	0.28	0.19	0.25	+32%
Emissions from Purchased Electricity (Scope 2)	4.71	5.50	5.27	-4%
Emissions from Indirect Sources (Scope 3)	34.71	39.91	45.75	+15%
Corporate purchases and Amazon-branded product emissions (e.g., operating expenses, business travel, and Amazon-branded product manufacturing, use phase, and end-of-life)	11.95	15.41	16.70	+8%
Capital goods (e.g. building construction, servers and other hardware, equipment, vehicles)	4.64	8.01	10.52	+31%
Other indirect emissions (e.g., third-party transportation, packaging, upstream energy related)	13.89	12.44	15.77	+27%
Lifecycle emissions from customer trips to Amazon's physical stores	4.23	4.05	2.77	-32%
Amazon's Total Footprint	44.40	51.17	60.64	+19%

CO₂ footprint data obtained from Amazon due to packaging, it contributed to 15.7% CO₂ in the year 2020 alone [7].

2. State of Art

2.1 Kalundborg symbiosis

Kalundborg Symbiosis is a partnership between twelve public and private companies in Kalundborg. In addition, working closely with project Biopro [14] [2] (A biotech company which partnered with other private in companies in Kalundborg to make a biotech production more sustainable and efficient which is achieved by reducing the consumption of energy and raw materials while improving yields). Together, since 1972, they have developed the world's first industrial symbiosis with a circular approach to production. The main principle is that a residual that is waste flow in one company becomes a resource in another, for the benefit of both the environment and the economy. In a local partnership, the resources can be shared to recycle resources, thus both achieving savings and minimising waste. The symbiosis creates community growth and supports the green transition.

This method can be applied to India because India has many MPEZ (Metropolitan Economic Zones) in urban areas, SEZ (Special Economic Zones) in sub-urban areas and SIPCOT (State Industries Promotion Corporation of Tamil Nadu) in Rural areas [6], all these regions consists of Industries of different fields and its layout is as same as Kalundborg that is why it is easy to apply the concept of circular economics in India as it have got the facilities already.

2.2 Biomimicry

It is an inherent part of Circular economics where an engineer copy ideas from mother nature and adapt it to our needs, this concept is being introduced here though not used it in my project could be a scope in future to copy from living organisms how effectively they re-use the waste as raw materials.

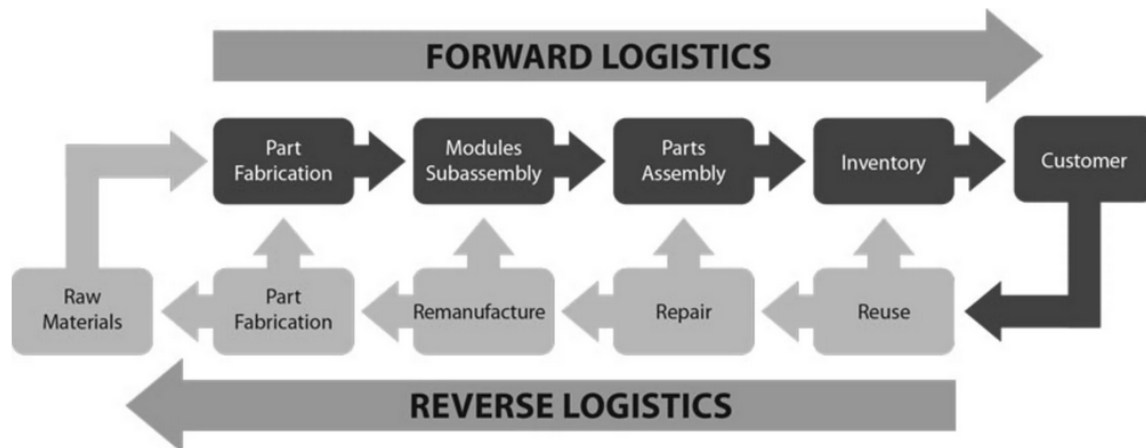
Biomimicry is an innovation process modelled on living things. It attempts to find solutions invented by nature to answer the problems of our society. It's a multidisciplinary approach that often starts with basic research, usually followed up by engineering and then popularized and marketed by entrepreneurship. It's a methodology that tends to link ecology and technology [9] [16].

Example: When a tree is fallen, the wood is upcycled for various purposes by microbes for their shelter, source of food and certain fungus breaks the complex chemicals into Lignin which is used as building blocks of other chemicals.

2.3 Enabling the Circular Economy transition: a sustainable lean manufacturing recipe for Industry 4.0

Over the past 10 years, the concepts and objectives of circular economy have been increasingly popular with detailed description and become strategic issues of international, European, and national policies. However, the transition towards circular production models continues to be affected by several barriers and critical factors that make the transition difficult to achieve.

My topic tries to design a relationship between sustainable production and lean production, highlighting the opportunity to invest in reverse-logistics [21] and how Industry 4.0 system represents a breeding ground for circular economy in a targeted application. The aim of the current study is to examine the relationships among sustainable production, lean production, and Industry 4.0 in order to evidence the need to adopt a lean methodology and Industry 4.0 technologies in a sustainable development perspective for companies [15].



Reverse-Logistics is the methods of implementing, controlling, and planning the cost-effective flow of finished goods, raw materials, and in-process inventory. The flow is from the point of consumption that is customer to the point of origin that is manufacturer, to properly dispose of these or to recapture value [21].

2.4 Butterfly Model

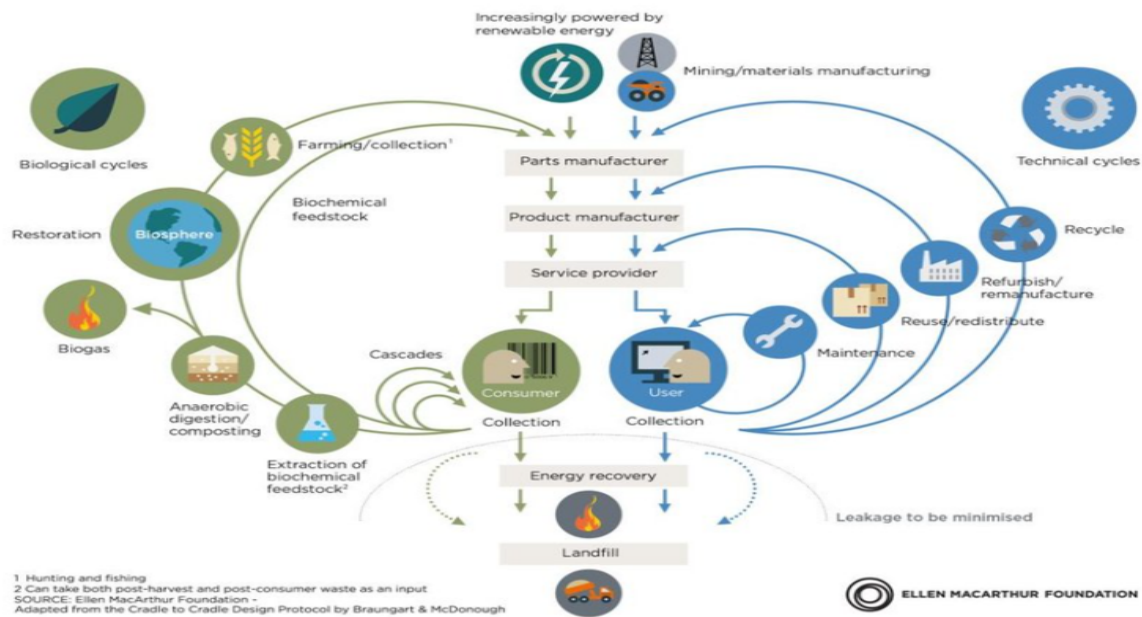
The butterfly diagram [11] is based on cradle-to-cradle thinking designed by Baumgart and McDonough (2009) [18]. They distinguished biological and technical nutrients, initially defined as materials required by the biosphere and the Technosphere (system of industrial processes) respectively. The diagram tries to capture the flow of materials, nutrients, components, and products, while adding an element of financial value. It builds on several schools of thought, but it is perhaps the most recognisably influenced by Cradle to Cradle's two material cycles.

A circular economic butterfly model is the key to seek and to rebuild capital, whether this is financial, manufactured, human, social or natural. This ensures enhanced flows of goods and services. The system diagram illustrates the continuous flow of technical and biological materials through the 'value circle'.

Biologically available nutrients can be organic or inorganic and are described as materials or products *“designed to return to the biological cycle by being consumed by micro-organisms in the soil and by other animals”*. Technical nutrients are designed to stay within the Technosphere only, but there is *“a confusion of flows”* in which materials that can be harmful to the environment (e.g., metal, and mineral-based products) enter organic matter and water flows. Despite this, Baumgart and McDonough argue to strive to keep flows separate, as indicated in the butterfly diagram, the biosphere wing is depicted as the source of organic resources, omitting the role of the biophysical environment in (re)circulating other materials

that are essential for society (e.g., water, aggregates, minerals, and metals). As indicated above, the bioeconomy community continued to build on this image and adopted the biosphere wing as being the ‘bioeconomy’ part of the circular economy (e.g. Carrez and van Leeuwen, 2016) [13]. Based on the prevalence of naturally occurring and designed material composites and mixtures, the consideration of separate flows of organic and inorganic is inappropriate, as noted by others (e.g. Carus and Dammer, 2018) [19]. The initial starting point of Baumgart and McDonough, distinguishing flows of materials within our industrial system and to and from the wider environment, may be a more fruitful one to visualize the circular economy anew.

CIRCULAR ECONOMY - an industrial system that is restorative by design



3. Motivation

3.1 Motto

The main motto behind choosing this subject is to show that even new concept of circular economics can be incorporated into Industry 4.0 framework and to have self-satisfaction of developing a simple project which can contribute to reduction carbon footprint.

The further motto is by using my company Onward technologies limited (Where I work) is to give an idea or proposal how we can suggest our clients like Ford Motors and Caterpillar to invest in circular economics using digitalization and Industry 4.0.

There are plans to develop this project further in future using AI (R-Language) for forecasting, understanding the customer buying patterns and package choosing patterns and give real time suggestions.

Further if my company supports, this research will be further developed into prototype of physical models, cloud integration and exploit the full power of IIOT.

In countries like India from needs awareness of circular economics and Industry 4.0 through one, we could also become a cleaner country.

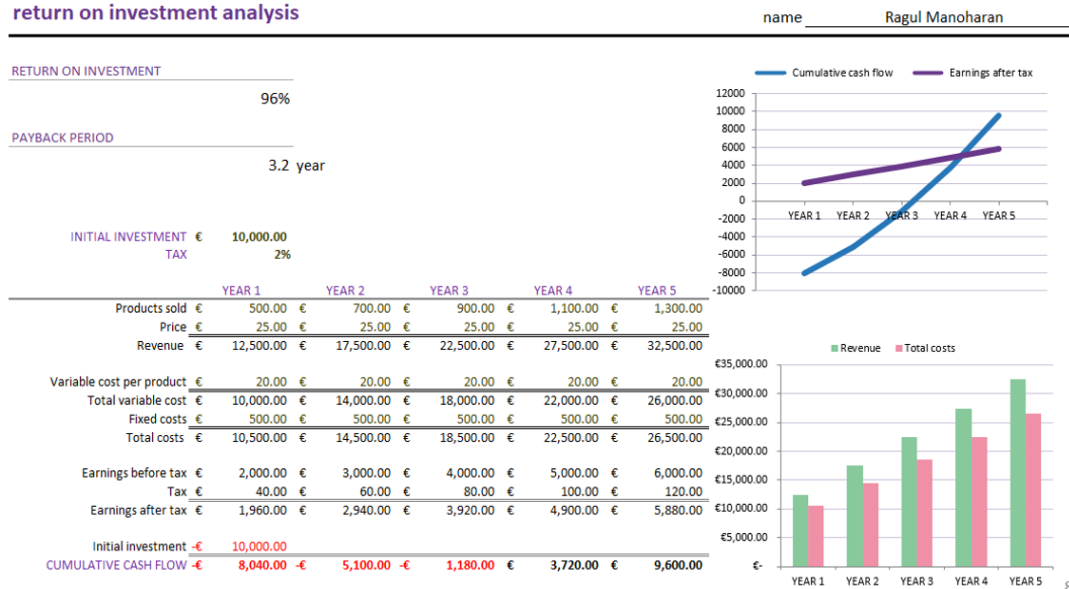
This project focused on cardboard packaging alone, but it can be applied to any product industry, and may incorporate a concept of leasing products which is also a coherent part in circular economics.

Example: Global recycling company Gemini Corporation has its sights set on turning India's mounting plastic waste problem into huge profits. The country's plastic upcycling industry is highly unorganized, with people scavenging rubbish dumps to gather material with no protective clothing, while health and safety standards are often non-existent at scrap dealer's facilities where workers sort through mountains of trash. Plastic upcycling in India is largely a 'grey market', and for Belgium-based Gemini, it sees a lucrative opportunity to turn the sector into a formal economy [10].

3.2 Return on Investment

We are living in a world where we keep on dumping garbage which can be turned into wealth. This project considers cardboard packages as waste which can be regenerated instead of recycling (recycling uses lot of incineration techniques which gives out harmful gases out to environment) to produce raw materials for packaging industries.

return on investment analysis



For whatever the investment we make, we achieve break even in 3rd year, after which it is only profit that we obtain from waste. The initial investment made is around 10000 Euros in order to procure the physical entities to venture out as a start-up company. Considering the tentative price of goods sold from 1st year to 5th year tend to increase by 200 euros. The variable cost is kept constant throughout this time period by optimal utilization financial, social, human resources of as we are handling only the waste that is left behind, the fixed cost for operation is kept as 500 euros. Earning before the Tax(EBIT) is considered to increase from year to year. The earning becomes larger as government of India tends to give more subsidies that is up to 50% [22] .

The costs are calculated in INR which converted to Euros for easy understanding. The source for investment comes from the “Green Company” which initially invest to kick start the business, as green company acts as the middlemen for the service. The customer has the

opportunity to choose right size of package that is needed to them and only help that is required from customer who shops online is to deposit the cardboard boxes in green bin. Once the order is delivered the customer is given few days of time to deposit the cardboard box in the nearest green bin. The value we obtain is from regeneration unit which utilizes the cardboard boxes as raw materials for their production. Green Company ships them the raw material at a much cheaper price than extracting it from nature.

4. Aim & Objective

The aim of the project is to turn linear product life cycle into circular life cycle.

- The system automatically helps customer to shop not only products but also enables to choose custom packages.
- These packages are finally put into Circular trash bins which automatically communicates to garbage truck for collection.
- After collecting the packages, the truck reaches Green Company from where it goes into a regeneration unit.
- Hence the packages (cardboard boxes) are either reused, refurbished, or sent to other industries as raw materials.
- The consumers interact in the system only during shopping rest all are smartly handled and regeneration is done through available engineering processes.
- This system keeps track of customer order along with product and packages, filling of trash bins, collection truck, green company, and regeneration unit.
- Interface is simple to use in choosing the product and customised packages.
- The database relies on MySQL
- The online shop and all necessary interfaces to work with DB are made in C++.

4.1 Specification

4.1.1 Characteristics to be selected by consumer

- Customer can choose Product (Laptops)
- Custom package as they shop.
- Nearest circular green bin.

4.1.2 Database

- ID.
- Execution of order.
- Trash bin filling.
- Truck dispatch and collection.

- Reaching Green company.
- Final destination regeneration unit.

4.1.3 Application

- The order is tracked.
- Filling of green bin is monitored.
- Collection truck and delivery trucks are monitored.
- To ensure safe regeneration, reuse and repair is taking place.

5. Sustainable Developmental Goals

Out of the seventeen Sustainable development goals laid out by United nation, in this research the goal that is chosen is number eleven that is “Sustainable Cities and Communities”. This is topic is chosen because due to huge urbanization in third world nation, this is a high time we implement the concept of Circular Economics, where we can regenerate value from waste instead of recycling and incinerating where we add energy, and the output of incineration are harmful gases which contributes to the global warming. This project helps destitute communities to harvest income from waste which in turn helps in development of the community in sustainable manner [3] [2] [4] [5] as well as the country.

5.1 Sustainable Cities and Communities

There is a strong link between the quality of life in cities and how cities draw on and manage the natural resources available to them. To date, the trend towards urbanization has been accompanied by increased pressure on the environment and accelerated demand for basic services, infrastructure, jobs, land, and affordable housing, particularly for the nearly 1 billion urban poor who live in informal settlements. Due to their high concentration of people, infrastructures, housing and economic activities, cities are particularly vulnerable to climate change and natural disasters impacts. Building urban resilience is crucial to avoid human, social and economic losses [3] [4] [5] while improving the sustainability of urbanization processes is needed to protect the environment and mitigate disaster risk and climate change. Resource efficient cities combine greater productivity and innovation with lower costs and reduced environmental impacts, while providing increased opportunities for consumer choices and sustainable lifestyles.

5.2 Circular Economic Business Model

- Environmental Impact: No incineration cause CO₂ emissions to reduce and demand to extract raw materials from nature is highly reduced.
- Circular Value Chain: The Customer who orders online, packaging & online shop and government play an important role in achieving the moto of the project.
- Unique Circular Value chain: The customer would get offers on next orders when the deposit the carboard boxes in green bins.

- Social Impact: if people really care about nature, they will go for refurbishment, reuse and repairs [8].

5.3 Top 10 methods of Circular Economics [17]

- **Renewability**
Capability of being replaced by natural cycles or by sound management of renewable resources.
- **Up cyclability**
The process of taking waste and creating something new in its current state without destruction.
- **Repairability**
It is measure of the degree of the degree to and ease with which a product can be repaired and maintained by end consumers without making the product obsolete.
- **Maintenance**
The process of being preserved at a condition.
- **Updatability**
To alter as to be up to date of a product with respect to time.
- **Reusability**
To use again, especially after salvaging or special treatment or processing [4].
- **Shareability**
The state or property of being able to be shared, for example awareness about public transport, carpooling techniques, which immensely helps in reducing the release of harmful gases.
- **Open Sourced**
Products or components that are commonly available or produced by common people instead of an organization or an individual holding pattern.
- **Dematerialization/Digitalization**
The process of converting physical non-living things into electronic formats. For example: certificates, books, etc.
- **Refurbishment**
The process of cleaning, decorating or providing it with additional features or functionalities.
- **Cascade Value**
Cascading is the process of maximising resource effectiveness by using biological material or technological material in products that creates most economic value over

multiple lifetimes or continuously extracting the residual value up to the final disposal.[12].

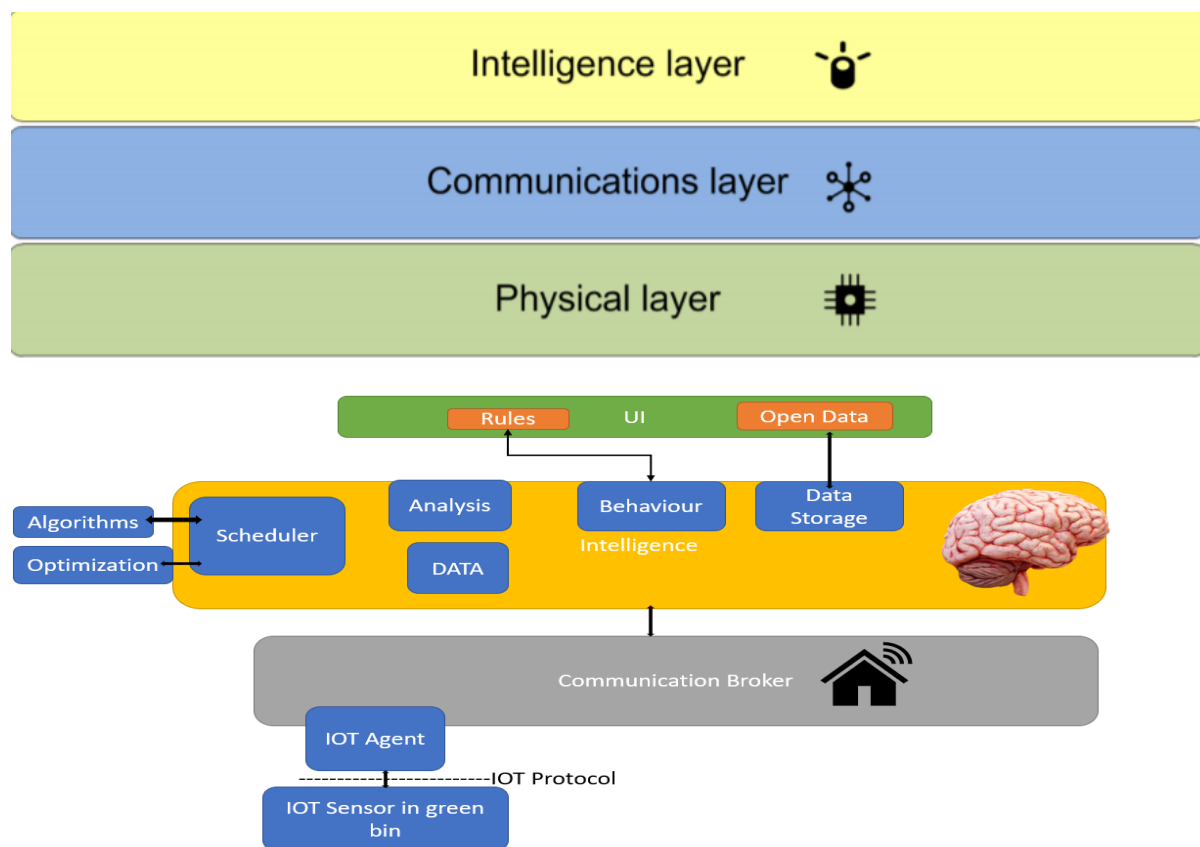
- **Servitization**

Industries using their products to sell “outcome as service” rather than one-off sale [8].

6. Method of Implementation

In this section we are going to talk about how the motto of my project is achieved using tools available in Industry 4.0.

6.1 Basic Working Structure and Architecture

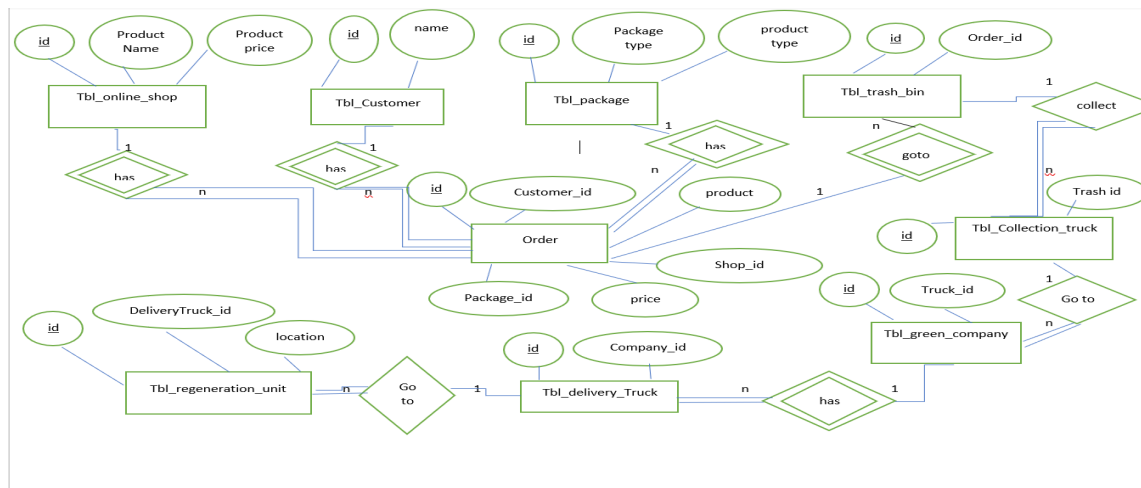


There is a physical layer help us to defines the electrical and physical specification of data connection, which establishes relationships with devices such as adapters, networking hub, trash bin, green company, regeneration unit, collection and delivery truck on which the physical entities relies on, the physical layer is not concerned with protocols, this physical layer defines the relationship between a device and transmission medium then there is a communication layer where IOT protocols would be used in order to dispatch the truck for collection of trash and the Intelligence layer consists of intelligence to know when the bin is full and to call of collection truck. Once the Green Bin where is filled IOT protocols are used to communicate with the garbage collection truck automatically, then the garbage truck

reaches the circular trash bin to collect the trash which is transported to Green Company at which the packages are sorted manually and from here the delivery truck picks up cardboard boxes and delivers to companies which follows the ten methods of circular economics.

6.2 Conceptual Model

The Customer can trash the cardboard boxes in any condition, if they leave it in OK condition we can repair and reuse it for another delivery or else it would be sent to regeneration unit.

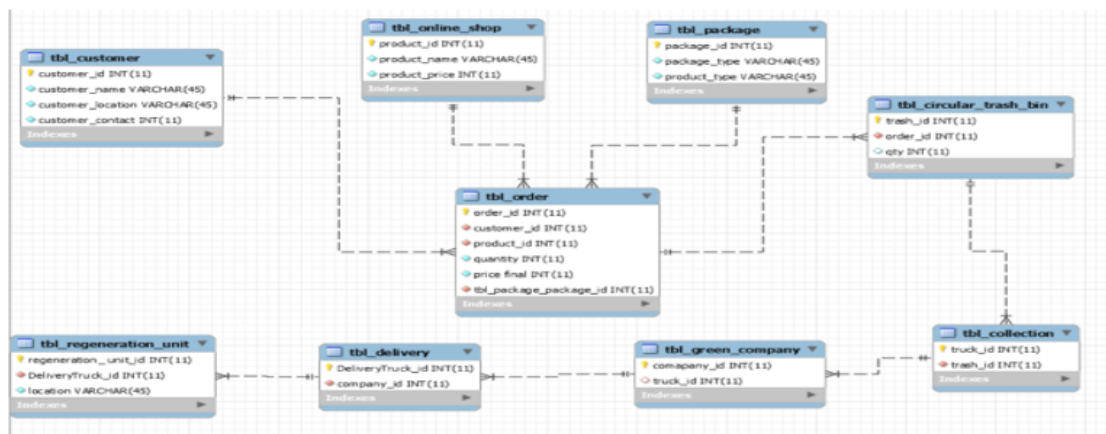


This is the basic conceptual idea where we can see the customer can enter into an online shop buy the required product and customise the package as required and the order is executed, once the order is executed the product is delivered to the customer and the customer deposits the cardboard boxes into the green bin (Trash Bin Size=10), once filled collection truck automatically dispatched to collect trash (IOT protocols would be used to alarm the collection truck that the bin is full) and reaches the green company where sorting of garbage takes place and the delivery truck delivers the cardboard boxes as raw material for required industry.

Each box contains a QR code through which we can trace each cardboard boxes. The conceptual model is a basic schema where once can understand the cardinality, attributes (given inside an ellipse) that are present inside an entity and the weak entity (inside a double rectangular box) and strong entities (present inside a single rectangular box) and their associated primary and foreign keys respectively.

The Primary Key (Present inside an elliptical box with an underline) of an entity helps us to uniquely identify a row in a table, it is usually present in a strong entity. The Foreign key in a column of a relates to a unique value in another table which is often a primary key to create a way of cross referencing the two tables. The conceptual design also displays the association (inside a diamond) between two entities. The conceptual design also tells the basic schema of existence dependency and identification dependencies and further total participation, which is nothing, but an existence dependency denoted by double lines and partial participation which is denoted by single line. From rough draft of conceptual design, more technical Entity-Relation Logical model can be created.

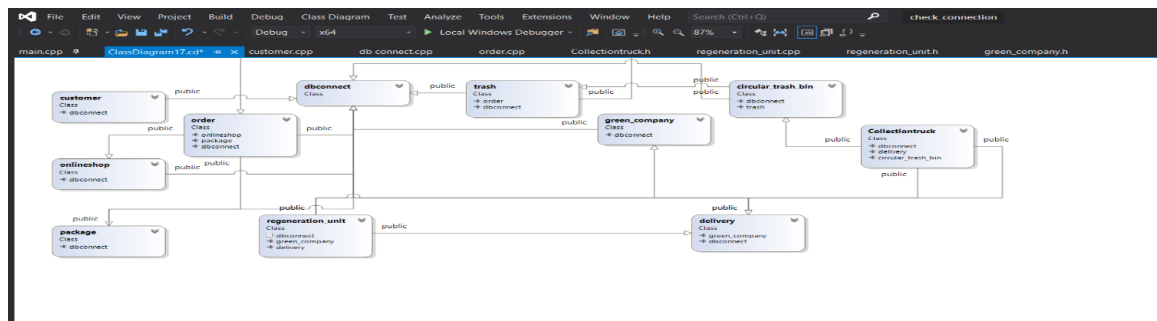
6.3 Logical Design



In the Logical Design, Primary keys, foreign keys, attributes, instance relationships (1:n), existence and identification relationships are given accordingly. As we see in the Logical Design each box is a strong entity which has their own primary keys and relates to other entity as foreign key. Each strong entity has its own attributes which is filled through Microsoft Visual Studio and stored. The database utilises DDL and DML commands to successfully execute the idea of project. Existence and Identification dependency is used while creating Logical Entity relationship.

6.4 UML Class Design

This diagram portrays how the MS Visual studio class entities interacts with DB.



In the UI, primarily first the database is connected to the Microsoft Visual Studio, once the connection is successful,

- The customer can register to an online shop as well as he can update and delete the details from the online shop.
- Then the customer is shown with list of products in my case it is laptops where he can choose the required laptop.
- After choosing the product he can select required cardboard box for his laptop and finally the order is executed and stored in database.
- The package then goes to the circular trash bin, once filled (bin size can hold up to 10 boxes) which is automatically counted the collection truck is automatically dispatched to collect the trash.

- Using Standard Template Library which counts the total number of trash being delivered to green company which acts as the middlemen for the collected trash.
- From the green company these trashes are sent to regeneration units where it is either repaired and sent out for another deliver or can be reused as raw material.
- In this manner the material goes in a circular path, as we know most of carboard boxes are made from wood which can lead to deforestation, my project is a simple idea through which we can avoid deforestation and thus continuously getting resources from nature.
- Multiple inheritance, Virtual inheritance are used in order to utilise function which is already created in some other class, data abstraction is used for security and dynamic memory allocations are used to allocate memory that the program needs in heap in order to have a flexibility in usage of memory in main, wherever it is required.

7. Resources Used

- MYSQL WORKBENCH
- Microsoft Visual Studio 2019
- Office 365- Word for Conceptual Model.

8. Conclusion

The increase in CO₂ footprint which results greenhouse effect and global warming , this paper deeply relies on Circular Economic principles incorporating industry 4.0 technologies to have a cradle-to-cradle approach by which waste especially cardboard boxes can be turned into economic resources and also prevent us from continuously depend on mother nature for raw materials.

Among the 17 Sustainable Development Goals, this paper considered mostly on answering the question of how to make Sustainable cities and Responsible Consumption and production, this is achieved by using Circular Economic principles to obtain value from waste instead of traditional method of disposing wastes.

Incorporating Industry 4.0 tools with circular economics we can upcycle, refurbish, and repair things that are considered to be a waste by consumers to give a new life or it can be used as a raw material through industrial symbiosis.

As discussed earlier in this paper CO₂ emission have contributed to double the number of deaths due to air pollution in last 20 years, this is mainly because of incinerating waste and the pollutants released into atmosphere. Principles of Circular economics would eventually reduce the number of deaths due to incineration of waste and also paves way for more employment opportunities for destitute population which moves to urban areas to find jobs through their economic standard will improve thus improving the economy of a country with sustainability. With this outcome this research, we cannot stop innovating, more and more non-biodegradable wastes must be brought into the hands of cradle-to-cradle approach and

step by step elimination of cradle-to-grave approach. Further, the research has to be physically implemented by incorporating IIOT technologies, Machine Learning principles to harvest all the fruitful benefits of Circular Economics and Industry 4.0.

Acknowledgement

This work is generously supported by Universidad Pontificia Comillas, Spain.

I would like to convey a special thanks to Dr. Juan Carlos Del Real, Dr. Jesus Jiménez, Dir. Bernardo Villazan Gil, Dr. Jaime Boal, Dr. Alvaro Jesus Lopez Lopez, Dr. Caleb Raj, and my Father Mr. Manoharan for supporting me and guiding me in each and every step to carry out this research.

9. References

- [1] Gallego-Schmid, A., Mendoza, J. M. F., & Azapagic, A. (2019). Environmental impacts of takeaway food containers. *Journal of Cleaner Production*, 211, 417-427.
- [2] Awasthi, M. K., Sarsaiya, S., Wainaina, S., Rajendran, K., Awasthi, S. K., Liu, T., Yumin, D., Jain, A., Raveendran, S., Parameshwaran, B., Ashok, P., Zenqiang, Z., & Taherzadeh, M. J. (2021). Techno-economics and life-cycle assessment of biological and thermochemical treatment of bio-waste. *Renewable and Sustainable Energy Reviews*, 144, 110837.
- [3] Grafström, J., & Aasma, S. (2021). Breaking circular economy barriers. *Journal of Cleaner Production*, 126002.
- [4] Sgroi, M., Vagliasindi, F. G., & Roccaro, P. (2018). Feasibility, sustainability and circular economy concepts in water reuse. *Current opinion in environmental Science & Health*, 2, 20-25.
- [5] Nikolaou, I. E., & Tsagarakis, K. P. (2021). An introduction to Circular Economy and Sustainability: Some existing lessons and future directions.
- [6] Niti Aayog on Circular Economy published Indian Government Order on 18th March 2021.
- [7] Sustainability.aboutamazon.com.
- [8] ellenmacarthurfoundation.org/assets/downloads/publications/Elle-MacArthur-Foundation-Towards-the-Circular-Economy-vol.1.pdf.
- [9] Vincent, J. F., Bogatyreva, O. A., Bogatyrev, N. R., Bowyer, A., & Pahl, A. K. (2006). Biomimetics: its practice and theory. *Journal of the Royal Society Interface*, 3(9), 471-482.
- [10] thenationalnews.com/business/economy/how-one-company-is-helping-india-progress-towards-a-circular-economy-1.1131469.

- [11] Velenturf, A. P., Archer, S. A., Gomes, H. I., Christgen, B., Lag-Brotons, A. J., & Purnell, P. (2019). Circular economy and the matter of integrated resources. *Science of the Total Environment*, 689, 963-969.
- [12] ceguide.org
- [13] Carrez, D., & Van Leeuwen, P. (2015). Bioeconomy: circular by nature. *The European Files*, 38, 34-35.
- [14] biopro.nu
- [15] ge.com/digital/blog/industry-40-changing-world-could-circular-economy-save-it.
- [16] asknature.com
- [17] ecologing.es
- [18] McDonough, W., & Braungart, M. (2010). *Cradle to cradle: Remaking the way we make things*. North point press.
- [19] Carus, M., & Dammer, L. (2018). The circular bioeconomy—concepts, opportunities, and limitations. *Industrial biotechnology*, 14(2), 83-91.
- [20] theprint.in/science/air-pollution-from-fossil-fuels-causes-over-30-deaths-in-india-every-year-harvard-study/601757/
- [21] c3controls.com/white-paper/what-is-reverse-logistics/
- [22] mnre.gov.in

Appendix. Code and DB Scripts

C++ Code:

Main:

```
#include<windows.h>
#include<mysql.h>
#include<iostream>
#include<string.h>
#include<sqlext.h>
#include<stdio.h>
#include<iomanip>
#include<vector>
#include "db_connect.h"
#include "Collectiontruck.h"
#include "circular_trash_bin.h"
#include "green_company.h"
#include "delivery.h"
#include "regeneration_unit.h"
#include "Collectiontruck.h"
#include "customer.h"
#include "onlineshop.h"
#include "package.h"
#include "order.h"
#include "trash.h"

using namespace std;
int main(char** args)
{
    cout << "Welcome to the Circular_Economics_Online Shop" << endl;
    cout << "You are one step ahead to make Linear_Product_Life_Cycle into
Circularr_life_cycle " << endl;

    dbconnect db;
    db.connect();

    MYSQL_RES* result;
    MYSQL_ROW row;
    MYSQL* connection, mysql;
    int state;
    int rc = 0;
    int row_num = 0;
    mysql_init(&mysql);
    connection = mysql_real_connect(&mysql, "localhost", "root", "root",
"jesus", 3306, 0, 0);

    // check for a connection error //
    if (connection == NULL) {
        // print the error message //
        cout << (mysql_error(&mysql));
        return 1;
    }
    customer c1;
    c1.displaycustomer();
    c1.updatecustomer();
    mysql_init(&mysql);
```

```

        result = mysql_store_result(connection);

// check for a connection error //
    if (connection == NULL) {
        // print the error message //
        cout << (mysql_error(&mysql));
        return 1;
    }
    onlineshop os;
    os.displayproducts();
    os.selectproduct();
    mysql_init(&mysql);
    result = mysql_store_result(connection);

// check for a connection error //
    if (connection == NULL) {
        // print the error message //
        cout << (mysql_error(&mysql));
        return 1;
    }

package pc;
    pc.displaypackage();
    pc.selectpackage();
    mysql_init(&mysql);
    result = mysql_store_result(connection);
    cout << "IF your a customer of ID=1(MANOHARAN), then you have choice
to select keys 1('dell' and 'small package') or 2('mac' and 'medium_package')
Or 3('lenevo and large_package')" << endl;
    cout << "IF your a customer of ID=2(RAGUL), then you have choice to
select keys 4('dell' and 'small package')or 5('mac' and 'medium_package') Or
6('lenevo and large_package')" << endl;
    cout<< "IF your a customer of ID=3(KALAIMATHI), then you have choice
to select keys 7('dell' and 'small package') or 8('mac' and 'medium_package')
or 9('lenevo and large_package')" << endl;

    mysql_init(&mysql);
    connection = mysql_real_connect(&mysql, "localhost", "root", "root",
"jesus", 3306, 0, 0);
    /* check for a connection error */
    if (connection == NULL) {
        /* print the error message */
        cout << (mysql_error(&mysql));
        return 1;
    }
    order o;
    o.Order();
    mysql_init(&mysql);
    result = mysql_store_result(connection);

//trash class & collection_truck dispatch//
    /*dynamic memory*/
    circular_trash_bin* tr = new circular_trash_bin[11];
    mysql_init(&mysql);
    connection = mysql_real_connect(&mysql, "localhost", "root",
"root", "jesus", 3306, 0, 0);
    // check for a connection error //
    if (connection == NULL) {
        // print the error message //
        cout << (mysql_error(&mysql));
        return 1;
    }
}

```

```

tr->bin();
tr->fillBin();
delete[] tr;
mysql_init(&mysql);
result = mysql_store_result(connection);
cout << "success" << endl;

// collection* clt;\

// clt->&del;//

//Dynamic memory alloc//
Collectiontruck *ctr=new Collectiontruck[11];
ctr->display();
ctr->truck();
delete[] ctr;

vector<int> vec;
int i;
cout << "vector size-" << vec.size() << endl;
for (i = 0; i <= 10; i++)
{
    vec.push_back(i);
}
cout << "extended vector size=" << vec.size() << endl;
for (i = 0; i <= 10; i++)
{
    cout << "values of vec[" << i << "]= " << vec[i] << endl;
}
vector<int>::iterator v = vec.begin();
while (v != vec.end()) {
    cout << "value of v=" << *v << endl;
    v++;
}
cout << "package is going to get delivered to green company" <<
endl;

//green company//
mysql_init(&mysql);
connection = mysql_real_connect(&mysql, "localhost", "root",
"root", "jesus", 3306, 0, 0);
// check for a connection error //
if (connection == NULL) {
    // print the error message //
    cout << (mysql_error(&mysql));
    return 1;
}
green_company gc;
mysql_init(&mysql);
result = mysql_store_result(connection);
cout << "success" << endl;
//regeneration unit
mysql_init(&mysql);
connection = mysql_real_connect(&mysql, "localhost", "root",
"root", "jesus", 3306, 0, 0);
// check for a connection error //
if (connection == NULL) {
    // print the error message //
    cout << (mysql_error(&mysql));
    return 1;
}

```

```

    }
    gc.display();
    gc.insertGreenCompany();
    mysql_init(&mysql);
    result = mysql_store_result(connection);
    cout << "success" << endl;
    //delivery_truck with virtual functionwith overriding or
overloading//no signature is initialised in dynamic functions//
    mysql_init(&mysql);
    connection = mysql_real_connect(&mysql, "localhost", "root",
"root", "jesus", 3306, 0, 0);
    // check for a connection error //
    if (connection == NULL) {
        // print the error message //
        cout << (mysql_error(&mysql));
        return 1;
    }
    /*Vitrul functions*/
    green_company* gc1;
    delivery del;
    gc1 = &del;
    gc1->display();
    del.truck();

    //regeneration_unit
    mysql_init(&mysql);
    connection = mysql_real_connect(&mysql, "localhost", "root",
"root", "jesus", 3306, 0, 0);
    // check for a connection error //
    if (connection == NULL) {
        // print the error message //
        cout << (mysql_error(&mysql));
        return 1;
    }
    regeneration_unit *reg=new regeneration_unit[11];
    reg->destiny();
    delete[] reg;
    mysql_init(&mysql);
    result = mysql_store_result(connection);
    cout << "success" << endl;

    cout << "Finally the your packaging is safely regenerated for
someother purpose" << endl;
    cout << "Be proud that we dint extract cardboard from trees like
before because of your small initiative" << endl;
    cout << "The software has reached it's end and destiny" << endl;
    //Deleting from trash bin, green company and regeneration unit//
    cout << "if you want to delete last entry into order, trashbin,
green_company and regeneration unit press '1', don't want to delete a record
press '2'" << endl;
    int d;
    cin >> d;
    if (d == 1) {
        mysql_init(&mysql);
        connection = mysql_real_connect(&mysql, "localhost", "root",
"root", "jesus", 3306, 0, 0);
        state = mysql_query(connection, "DELETE FROM
`jesus`.`tbl_regeneration_unit` WHERE (`regeneration_unit_id` = '1')");
        state = mysql_query(connection, "DELETE FROM
`jesus`.`tbl_delivery` WHERE (`DeliveryTruck_id` = '1')");
        state = mysql_query(connection, "DELETE FROM
`jesus`.`tbl_green_company` WHERE (`comapany_id` = '1')");

```

```

        state = mysql_query(connection, "DELETE FROM
`jesus`.`tbl_collection` WHERE (`truck_id` = '1')");
        state = mysql_query(connection, "DELETE FROM
`jesus`.`tbl_circular_trash_bin` WHERE (`trash_id` = '1')");
        state = mysql_query(connection, "DELETE FROM
`jesus`.`tbl_order` WHERE (`order_id` = '1')");
        cout << "desired entry is deleted" << endl;
    }
    else if (d == 2)
    {
        cout << "nothing is deleted" << endl;
    }

    if (state != 0)
    {
        cout << (mysql_error(connection));
        return 1;
    }
    /* free the result set */
    mysql_free_result(result);
    /* close the connection */
    mysql_close(connection);
    cout<<("Done.\n");
}

```

Classes

```

//DB Header//
#pragma once
#include<windows.h>
#include<mysql.h>
#include<iostream>
#include<string.h>
#include<sqlext.h>
#include<stdio.h>
#include<iomanip>
#include<windows.h>
#include<mysql.h>
#include<iostream>
#include<string.h>
#include<sqlext.h>
#include<stdio.h>
#include<iomanip>
#include<vector>

using namespace std;

class dbconnect
{
public:
    MYSQL_RES* result;
    MYSQL_ROW row;
    MYSQL* connection, mysql;
    int state;
    int rc;
    int row_num;
    int connect();
};
//DB CPP//
#include "db connect.h"
#include<windows.h>

```

```

#include<mysql.h>
#include<iostream>
#include<string.h>
#include<sqlext.h>
#include<stdio.h>
#include<iomanip>
#include<windows.h>
#include<mysql.h>
#include<iostream>
#include<string.h>
#include<sqlext.h>
#include<stdio.h>
#include<iomanip>
#include<vector>

using namespace std;

int dbconnect::connect()
{
    MYSQL_RES* result;
    MYSQL_ROW row;
    MYSQL* connection, mysql;
    int state;
    int rc = 0;
    int row_num = 0;
    mysql_init(&mysql);
    connection = mysql_real_connect(&mysql, "localhost", "root", "root",
"jesus", 3306, 0, 0);
    cout << "db connected from db class" << endl;
    // check for a connection error //
    if (connection == NULL) {
        // print the error message //
        cout << (mysql_error(&mysql));
        return 1;
    }
}

//Customer Header//
#include<iostream>
#include<string.h>
#include<windows.h>
#include<mysql.h>
#include<iostream>
#include<string.h>
#include<sqlext.h>
#include<stdio.h>
#include<iomanip>
#include"db connect.h"

using namespace std;

class customer:public dbconnect
{
public:

    int state;
    MYSQL* connection;
    int a;
    char update;
    int change;
    int customerid;

```

```

public:
    customer();
    void displaycustomer();
    void updatecustomer();
    ~customer();

};

//Customer CPP//
#include<iostream>
#include<string.h>
#include<windows.h>
#include<mysql.h>
#include<iostream>
#include<string.h>
#include<sqllex.h>
#include<stdio.h>
#include<iostream>
#include<string.h>
#include<iomanip>
#include "customer.h"

using namespace std;
customer::customer() {}
void customer::displaycustomer() {
    MYSQL_RES* result;
    MYSQL_ROW row;
    MYSQL* connection, mysql;
    int state;
    int rc = 0;
    int row_num = 0;

    mysql_init(&mysql);
    connection = mysql_real_connect(&mysql, "localhost", "root", "root",
"jesus", 3306, 0, 0);
    // check for a connection error //
    if (connection == NULL) {
        // print the error message //
        cout << (mysql_error(&mysql));
    }
    //display registered customers//
    cout << "Please press '1' if you are a registered customer" << endl;
    cout << "A:" << endl;
    cin >> a;
    if (a == 1)
    {
        cout << "The customer registered are displayed below!" << endl;
        state = mysql_query(connection, "select* from tbl_customer;");
        result = mysql_store_result(connection);
        cout << ("Rows: %d\n", mysql_num_rows(result));
        // process each row in the result set //
        while ((row = mysql_fetch_row(result)) != NULL) {
            cout << setw(18) << "ID" << setw(18) << "NAME" << setw(18) <<
"LOCATION" << setw(18) << "CONATCT" << endl;
            cout << setw(18) << row[0]
                << setw(18) << row[1]
                << setw(18) << row[2]
                << setw(18) << row[3] << endl;
        }
    }
}
}

```



```

void customer::updatecustomer() {
    MYSQL_RES* result;
    MYSQL_ROW row;
    MYSQL* connection, mysql;
    int state;
    int rc = 0;
    int row_num = 0;

    cout << "if you need to update customer contact details please press
'a'" << endl;
    cout << "if you don't want to update customer details after pressin
'a', press 4 to carryon with shopping" << endl;
    cin >> update;
    cout << "enter an int 1 to update Manoharan,2 to update ragul ,3 to
update kalaimathi,4 for no update" << endl;
    cin >> change;
    if (update == a) {
        cout << "to update the details" << endl;
    }
    else if (change == 1) {

        mysql_init(&mysql);
        connection = mysql_real_connect(&mysql, "localhost", "root",
"root", "jesus", 3306, 0, 0);
        state = mysql_query(connection, "UPDATE `jesus`.`tbl_customer` SET
`customer_contact` = '90304050' WHERE(`customer_id` = '1')");
        cout << "updated" << endl;
    }
    else if (change == 2) {
        mysql_init(&mysql);
        connection = mysql_real_connect(&mysql, "localhost", "root",
"root", "jesus", 3306, 0, 0);
        state = mysql_query(connection, "UPDATE `jesus`.`tbl_customer` SET
`customer_contact` = '10203040' WHERE(`customer_id` = '2')");
        cout << "updated" << endl;
    }
    else if (change == 3) {
        mysql_init(&mysql);
        connection = mysql_real_connect(&mysql, "localhost", "root",
"root", "jesus", 3306, 0, 0);
        state = mysql_query(connection, "UPDATE `jesus`.`tbl_customer` SET
`customer_contact` = '40506070' WHERE(`customer_id` = '3')");
        cout << "updated" << endl;
    }
    else if (change == 4) {
        cout << "no update needed" << endl;
    }
}

cout << "enter a charcater namely 1,2,3 according to your id's" <<
endl;
cin >> customerid;
switch (customerid) {
case 1:
    cout << "Welcome Mr.Manoharan to online shop" << endl;
    break;
case 2:
    cout << "Welcome Mr.Ragul to online shop" << endl;
    break;
case 3:
    cout << "Welcome Mrs.kalaimathee to online shop" << endl;
    break;
default:

```

```

        cout << "you are not a registered ciustomer, Contact customer help"
<< endl;
    }

}
customer::~customer() {}

//Online shop Header//

#include<iostream>
#include<string.h>
#include<windows.h>
#include<mysql.h>
#include<iostream>
#include<string.h>
#include<sqlext.h>
#include<stdio.h>
#include<iomanip>
#include"db connect.h"
class onlineshop:public dbconnect{
public:
    int state;
    MYSQL* connection;
    int b;
    int productid;
public:
    onlineshop();
    void displayproducts();
    void selectproduct();
    ~onlineshop();

};

//online Shop cpp//

#include<iostream>
#include<string.h>
#include<windows.h>
#include<mysql.h>
#include<iostream>
#include<string.h>
#include<sqlext.h>
#include<stdio.h>
#include<iostream>
#include<string.h>
#include<iomanip>
#include "onlineshop.h"
#include "db connect.h"
#include"db connect.h"
using namespace std;

onlineshop::onlineshop() {}
void onlineshop::displayproducts() {
    MYSQL_RES* result;
    MYSQL_ROW row;
    MYSQL* connection, mysql;
    int state;
    int rc = 0;
    int row_num = 0;
    mysql_init(&mysql);

```

```

    connection = mysql_real_connect(&mysql, "localhost", "root", "root",
"jesus", 3306, 0, 0);
    /* check for a connection error */
    if (connection == NULL) {
        /* print the error message */
        cout << (mysql_error(&mysql));

    }
    int b;
    cout << "Please press 2 to view the product in the online store" << endl;
    cout << "B:" << endl;
    cin >> b;
    if (b == 2)
    {
        cout << "Welcome to Online_Shop and below you will find all the
products that are currently available" << endl;
        state = mysql_query(connection, "select* from tbl_online_shop;");
    }
    //laptop selection//
    result = mysql_store_result(connection);
    cout << ("Rows: %d\n", mysql_num_rows(result));
    /* process each row in the result set */
    while ((row = mysql_fetch_row(result)) != NULL) {
        cout << setw(18) << "PRODUCT_ID" << setw(18) << "PRODUCT_NAME" <<
setw(18) << "PRODUCT_PRICE" << endl;
        cout << setw(18) << row[0]
            << setw(18) << row[1]
            << setw(18) << row[2]
            << setw(18) << row[3] << endl;
    }
}

void onlineshop::selectproduct() {
    cout << "enter an integer  namely 1,2,3 according to your product id to
add to cart" << endl;
    cin >> productid;
    switch (productid) {
    case 1:
        cout << "You have choosen dell with $2000" << endl;
        break;
    case 2:
        cout << "You have choose MAC with $1500" << endl;
        break;
    case 3:
        cout << "You have choosen Lenevo with $1000" << endl;
        break;
    default:
        cout << "your cart is empty" << endl;
    }
}

onlineshop::~onlineshop(){}

//package header//
#pragma once
#include<iostream>
#include<string.h>
#include<windows.h>
#include<mysql.h>
#include<iostream>
#include<string.h>
#include<sqlxext.h>
#include<stdio.h>
#include<iomanip>
#include"db connect.h"

```

```

class package:public dbconnect
{
public:
    int state;
    MYSQL* connection;
    int c;
    int packageid;
public:
    package();
    ~package();
    void displaypackage();
    void selectpackage();

};

//package cpp//
#include<iostream>
#include<string.h>
#include<windows.h>
#include<mysql.h>
#include<iostream>
#include<string.h>
#include<sqlext.h>
#include<stdio.h>
#include<iomanip>
#include "package.h"
#include"db connect.h"
#include"db connect.h"
using namespace std;
package::package(){}
void package::displaypackage() {
    MYSQL_RES* result;
    MYSQL_ROW row;
    MYSQL* connection, mysql;
    int state;
    int rc = 0;
    int row_num = 0;
    mysql_init(&mysql);
    connection = mysql_real_connect(&mysql, "localhost", "root", "root",
"jesus", 3306, 0, 0);
    // check for a connection error //
    if (connection == NULL) {
        // print the error message //
        cout << (mysql_error(&mysql));

    }
    cout << "press 3 to see the list of packages available to choose" << endl;
    cin >> c;
    if (c == 3)
    {
        cout << "All the packages are displayed below to choose according to
your product" << endl;
        state = mysql_query(connection, "select* from tbl_package;");

    }
    result = mysql_store_result(connection);
    cout << ("Rows: %d\n", mysql_num_rows(result));
    //process each row in the result set //
    while ((row = mysql_fetch_row(result)) != NULL) {
        cout << setw(18) << "PACKAGE_ID" << setw(18) << "PACKAGE_TYPE" <<
setw(18) << "PRODUCT_TYPE" << endl;
        cout << setw(18) << row[0]
            << setw(18) << row[1]

```

```

        << setw(18) << row[2]
        << setw(18) << row[3] << endl;
    }
}
void package::selectpackage() {
    cout << "choose package according to your need" << endl;
    cout << "Namely press 1,2,3 to choose your desired product id" << endl;
    cin >> packageid;
    switch (packageid) {
    case 1:
        cout << "You have choosen small package" << endl;
        cout << "once your ID, ProductID and PAcKage ID you have choosen, then
the particular order will be executed" << endl;
        cout << "kindly find the Circular_Green_Bin to deposit your package,
you will be most welcome if dont tamper the packages much" << endl;
        break;
    case 2:
        cout << "You have choose medium package" << endl;
        cout << "once your ID, ProductID and PAcKage ID you have choosen, then
the particular order will be executed" << endl;
        cout << "kindly find the Circular_Green_Bin to deposit your package,
you will be most welcome if dont tamper the packages much" << endl;
        break;
    case 3:
        cout << "You have choosen large package" << endl;
        cout << "once your ID, ProductID and PAcKage ID you have choosen, then
the particular order will be executed" << endl;
        cout << "kindly find the Circular_Green_Bin to deposit your package,
you will be most welcome if dont tamper the packages much" << endl;
        break;
    default:
        cout << "kindly choose any one eco-packages!" << endl;
    }
}
package::~~package() {}

//order header//
#pragma once
#include<iostream>
#include<string.h>
#include<windows.h>
#include<mysql.h>
#include<iostream>
#include<string.h>
#include<sqlext.h>
#include<stdio.h>
#include<iomanip>
#include"onlineshop.h"
#include"package.h"
#include "db connect.h"
using namespace std;
class order:public onlineshop,public package,public dbconnect
{
public:
    int state;
    MYSQL* connection;
    int sorder;
    int customer_id;
    int product_id;
    int package_id;
    int qty;
    int price;
public:

```

```

    order();
    int Order();
    ~order();

};

//order cpp//
#include<iostream>
#include<string.h>
#include<windows.h>
#include<mysql.h>
#include<iostream>
#include<string.h>
#include<sqlext.h>
#include<stdio.h>
#include<iomanip>
#include "order.h"
#include"onlineshop.h"
#include"package.h"
#include"db connect.h"

using namespace std;

order::order(){}

int order::Order()
{
    MYSQL_RES* result;
    MYSQL_ROW row;
    MYSQL* connection, mysql;
    int state;
    int rc = 0;
    int row_num = 0;
    mysql_init(&mysql);
    connection = mysql_real_connect(&mysql, "localhost", "root", "root",
"jesus", 3306, 0, 0);

    int sorder;
    cout << "enter order s_orderid" << endl;
    cin >> sorder;
    int customer_id;
    cout << "enter your customer_id" << endl;
    cin >> customer_id;
    int product_id;
    cout << "enter the product id" << endl;
    cin >> product_id;
    int package_id;
    cout << "enter the package id" << endl;
    cin >> package_id;
    int qty;
    cout << "enter qty" << endl;
    cin >> qty;
    cout << "Dell= $2000, mac = $1500, hp= $1000" << endl;
    int price;
    cout << "enter the price amount" << endl;
    cin >> price;

    mysql_init(&mysql);

```

```

        connection = mysql_real_connect(&mysql, "localhost", "root", "root",
"jesus", 3306, 0, 0);
        // check for a connection error //
        if (connection == NULL) {
            // print the error message //
            cout << (mysql_error(&mysql));
            return 1;
        }

        switch (sorder)
        {
        case 1:
            if (customer_id == 1) {
                cout << "passing on" << endl;
                if (product_id == 1 && package_id == 1)
                {
                    cout << "passing on" << endl;
                }
                if (qty == 1 && price == 2000) {

                    mysql_init(&mysql);
                    connection = mysql_real_connect(&mysql, "localhost", "root",
"root", "jesus", 3306, 0, 0);
                    state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_order`VALUES( 1 , 1 , 1 , 1, 1,2000 )");
                    mysql_init(&mysql);
                    mysql_init(&mysql);
                    result = mysql_store_result(connection);
                    cout << "insertion successfull" << endl;
                    mysql_init(&mysql);
                    connection = mysql_real_connect(&mysql, "localhost", "root",
"root", "jesus", 3306, 0, 0);
                    cout << "inserting into circular_tbl_trash" << endl;
                    state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_circular_trash_bin` (`trash_id`, `order_id`, `qty`) VALUES ('1',
'1', '10')");
                    mysql_init(&mysql);
                    result = mysql_store_result(connection);
                    cout << "insertion successfull" << endl;
                    cout << "sum" << endl;
                    mysql_init(&mysql);
                    connection = mysql_real_connect(&mysql, "localhost", "root",
"root", "jesus", 3306, 0, 0);
                    state = mysql_query(connection, "select SUM(qty) from
tbl_circular_trash_bin;");
                    mysql_init(&mysql);
                    result = mysql_store_result(connection);

                }
            }
        }

        mysql_init(&mysql);
        connection = mysql_real_connect(&mysql, "localhost", "root", "root",
"jesus", 3306, 0, 0);
        /* check for a connection error */

```

```

if (connection == NULL) {
    /* print the error message */
    cout << (mysql_error(&mysql));
    return 1;
}
switch (sorder)
{
case 2:
    if (customer_id == 1) {
        cout << "pass on" << endl;

        if (product_id == 2 && package_id == 2)
        {
            cout << "pass on" << endl;
        }
        if (qty == 1 && price == 1500)
        {
            mysql_init(&mysql);
            connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
            state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_order`VALUES( 2 , 1 , 2 , 2, 1,1500)");
            result = mysql_store_result(connection);
            cout << "insertion successfull" << endl;
            mysql_init(&mysql);
            mysql_init(&mysql);
            connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
            cout << "inserting into circular_tbl_trash" << endl;
            state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_circular_trash_bin` (`trash_id`, `order_id`, `qty`) VALUES ('2',
'1', '10')");
            mysql_init(&mysql);
            result = mysql_store_result(connection);
            cout << "insertion successfull" << endl;
            mysql_init(&mysql);
            connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
            state = mysql_query(connection, "select SUM(qty) from
tbl_circular_trash_bin;");
            mysql_init(&mysql);
            result = mysql_store_result(connection);
            mysql_init(&mysql);
            connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
            state = mysql_query(connection, "select SUM(qty) from
tbl_circular_trash_bin;");
            mysql_init(&mysql);
            result = mysql_store_result(connection);

        }
    }
}

mysql_init(&mysql);
connection = mysql_real_connect(&mysql, "localhost", "root", "root",
"jesus", 3306, 0, 0);
/* check for a connection error */
if (connection == NULL) {
    /* print the error message */
    cout << (mysql_error(&mysql));
    return 1;
}

```



```

    }
    switch (sorder)
    {
    case 3:
        if (customer_id == 1) {
            cout << "pass on" << endl;

            if (product_id == 3 && package_id == 3)
            {
                cout << "pass on" << endl;
            }
            if (qty == 1 && price == 1000)
            {
                mysql_init(&mysql);
                connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
                state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_order`VALUES( 3 , 1 , 3 , 3, 1,1000)");
                result = mysql_store_result(connection);
                cout << "insertion successfull" << endl;
                mysql_init(&mysql);
                mysql_init(&mysql);
                connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
                cout << "inserting into circular_tbl_trash" << endl;
                state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_circular_trash_bin` (`trash_id`, `order_id`, `qty`) VALUES ('3',
'1', '10')");

                mysql_init(&mysql);
                result = mysql_store_result(connection);
                cout << "insertion successfull" << endl;
                mysql_init(&mysql);
                connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
                state = mysql_query(connection, "select SUM(qty) from
tbl_circular_trash_bin;");
                mysql_init(&mysql);
                result = mysql_store_result(connection);

            }
        }
    }
    mysql_init(&mysql);
    connection = mysql_real_connect(&mysql, "localhost", "root", "root",
"jesus", 3306, 0, 0);
    // check for a connection error //
    if (connection == NULL) {
        // print the error message //
        cout << (mysql_error(&mysql));
        return 1;
    }

    switch (sorder)
    {
    case 4:
        if ( customer_id == 2) {
            cout << "passing on" << endl;
            if (product_id == 1 && package_id == 1)
            {
                cout << "passing on" << endl;
            }
            if (qty == 1 && price == 2000) {

```

```

        mysql_init(&mysql);
        connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
        state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_order`VALUES( 4 , 2 , 1 , 1, 1,2000 )");
        result = mysql_store_result(connection);
        cout << "insertion successfull" << endl;
        mysql_init(&mysql);
        mysql_init(&mysql);
        connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
        cout << "inserting into circular_tbl_trash" << endl;
        state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_circular_trash_bin` ( `trash_id`, `order_id`, `qty`) VALUES ('1',
'2', '10')");

        mysql_init(&mysql);
        result = mysql_store_result(connection);
        cout << "insertion successfull" << endl;
        mysql_init(&mysql);
        connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
        state = mysql_query(connection, "select SUM(qty) from
tbl_circular_trash_bin;");
        mysql_init(&mysql);
        result = mysql_store_result(connection);

    }
}
}

```

```

mysql_init(&mysql);
connection = mysql_real_connect(&mysql, "localhost", "root", "root",
"jesus", 3306, 0, 0);
/* check for a connection error */
if (connection == NULL) {
    /* print the error message */
    cout << (mysql_error(&mysql));
    return 1;
}
switch (sorder)
{
case 5:
    if (customer_id == 2) {
        cout << "pass on" << endl;

        if (product_id == 2 && package_id == 2)
        {
            cout << "pass on" << endl;
        }
        if (qty == 1 && price == 1500)
        {
            mysql_init(&mysql);
            connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
            state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_order`VALUES( 5 , 2 , 2 , 2, 1,1500)");
            result = mysql_store_result(connection);
            cout << "insertion successfull" << endl;
            mysql_init(&mysql);

```

```

        connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
        cout << "inserting into circular_tbl_trash" << endl;
        state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_circular_trash_bin` (`trash_id`, `order_id`, `qty`) VALUES ('2',
'2', '10')");
        mysql_init(&mysql);
        mysql_init(&mysql);
        result = mysql_store_result(connection);
        cout << "insertion successfull" << endl;
        mysql_init(&mysql);
        connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
        state = mysql_query(connection, "select SUM(qty) from
tbl_circular_trash_bin;");
        mysql_init(&mysql);
        result = mysql_store_result(connection);

    }
}

mysql_init(&mysql);
connection = mysql_real_connect(&mysql, "localhost", "root",
"root", "jesus", 3306, 0, 0);
// check for a connection error //
if (connection == NULL) {
    // print the error message //
    cout << (mysql_error(&mysql));
    return 1;
}

switch (sorder)
{
case 6:
    if (customer_id == 2) {
        cout << "pass on" << endl;

        if (product_id == 3 && package_id == 3)
        {
            cout << "pass on" << endl;
        }
        if (qty == 1 && price == 1000)
        {
            mysql_init(&mysql);
            connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
            state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_order`VALUES( 6 , 2 , 3 , 3, 1,1000)");
            result = mysql_store_result(connection);
            cout << "insertion successfull" << endl;
            mysql_init(&mysql);
            mysql_init(&mysql);
            connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
            cout << "inserting into circular_tbl_trash" << endl;
            state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_circular_trash_bin` (`trash_id`, `order_id`, `qty`) VALUES ('3',
'2', '10')");

            mysql_init(&mysql);
            mysql_init(&mysql);
            result = mysql_store_result(connection);

```

```

        cout << "insertion successfull" << endl;
        mysql_init(&mysql);
        connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
        state = mysql_query(connection, "select SUM(qty) from
tbl_circular_trash_bin;");
        mysql_init(&mysql);
        result = mysql_store_result(connection);

    }
}
mysql_init(&mysql);
connection = mysql_real_connect(&mysql, "localhost", "root",
"root", "jesus", 3306, 0, 0);
// check for a connection error //
if (connection == NULL) {
    // print the error message //
    cout << (mysql_error(&mysql));
    return 1;
}
switch (sorder)
{
case 7:
    if (customer_id == 3) {
        cout << "passing on" << endl;
        if (product_id == 1 && package_id == 1)
        {
            cout << "passing on" << endl;
        }
        if (qty == 1 && price == 2000) {

            mysql_init(&mysql);
            connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
            state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_order`VALUES( 7 , 3 , 1 , 1, 1,2000 )");
            result = mysql_store_result(connection);
            cout << "insertion successfull" << endl;
            mysql_init(&mysql);
            connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
            cout << "inserting into circular_tbl_trash" << endl;
            state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_circular_trash_bin` (`trash_id`, `order_id`, `qty`) VALUES ('1',
'3', '10')");

            mysql_init(&mysql);
            mysql_init(&mysql);
            result = mysql_store_result(connection);
            cout << "insertion successfull" << endl;
            mysql_init(&mysql);
            connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
            state = mysql_query(connection, "select SUM(qty) from
tbl_circular_trash_bin;");
            mysql_init(&mysql);
            result = mysql_store_result(connection);

        }
    }
}
mysql_init(&mysql);

```

```

        connection = mysql_real_connect(&mysql, "localhost", "root",
"root", "jesus", 3306, 0, 0);
        /* check for a connection error */
        if (connection == NULL) {
            /* print the error message */
            cout << (mysql_error(&mysql));
            return 1;
        }
        switch (sorder)
        {
        case 8:
            if (customer_id == 2) {
                cout << "pass on" << endl;

                if (product_id == 2 && package_id == 2)
                {
                    cout << "pass on" << endl;
                }
                if (qty == 1 && price == 1500)
                {
                    mysql_init(&mysql);
                    connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
                    state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_order`VALUES( 8 , 2 , 2 , 2, 1,1500)");
                    result = mysql_store_result(connection);
                    cout << "insertion successfull" << endl;
                    mysql_init(&mysql);
                    connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
                    cout << "inserting into circular_tbl_trash" << endl;
                    state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_circular_trash_bin` (`trash_id`, `order_id`, `qty`) VALUES ('2',
'3', '10')");

                    mysql_init(&mysql);
                    mysql_init(&mysql);
                    result = mysql_store_result(connection);
                    cout << "insertion successfull" << endl;
                    mysql_init(&mysql);
                    connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
                    state = mysql_query(connection, "select SUM(qty) from
tbl_circular_trash_bin;");
                    mysql_init(&mysql);
                    result = mysql_store_result(connection);

                }
            }
        }

        mysql_init(&mysql);
        connection = mysql_real_connect(&mysql, "localhost", "root",
"root", "jesus", 3306, 0, 0);
        /* check for a connection error */
        if (connection == NULL) {
            /* print the error message */
            cout << (mysql_error(&mysql));
            return 1;
        }
        switch (sorder)
        {
        case 9:

```

```

        if (customer_id == 3) {
            cout << "pass on" << endl;

            if (product_id == 3 && package_id == 3)
            {
                cout << "pass on" << endl;
            }
            if (qty == 1 && price == 1000)
            {
                mysql_init(&mysql);
                connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
                state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_order`VALUES( 9 , 2 , 3 , 3 , 1,1000)");
                result = mysql_store_result(connection);
                cout << "insertion successfull" << endl;
                mysql_init(&mysql);
                connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
                state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_circular_trash_bin` (`trash_id`, `order_id`, `qty`) VALUES ('3',
'3', '10')");

                cout << "inserting into circular_tbl_trash" << endl;
                mysql_init(&mysql);
                mysql_init(&mysql);
                result = mysql_store_result(connection);
                cout << "insertion successfull" << endl;
                mysql_init(&mysql);
                connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
                state = mysql_query(connection, "select SUM(qty) from
tbl_circular_trash_bin;");
                mysql_init(&mysql);
                result = mysql_store_result(connection);

            }
        }
    }
}

order::~~order() {}

//trash header//
#pragma once
#include<iostream>
#include<string.h>
#include<windows.h>
#include<mysql.h>
#include<iostream>
#include<string.h>
#include<sqlext.h>
#include<stdio.h>
#include<iomanip>
#include "order.h"
#include"db connect.h"
class trash :public order,public dbconnect
{
public:
    int state;
    MYSQL* connection;
public:
    void Order();
};

```

```

//trash cpp//
#include<iostream>
#include<string.h>
#include<windows.h>
#include<mysql.h>
#include<iostream>
#include<string.h>
#include<sqlext.h>
#include<stdio.h>
#include<iomanip>
#include "order.h"
#include "trash.h"
#include"db connect.h"
#include"db connect.h"
using namespace std;
void trash::Order() {
    MYSQL_RES* result;
    MYSQL_ROW row;
    MYSQL* connection, mysql;
    int state;
    int rc = 0;
    int row_num = 0;
    mysql_init(&mysql);
    connection = mysql_real_connect(&mysql, "localhost", "root", "root",
"jesus", 3306, 0, 0);
    // check for a connection error //
    if (connection == NULL) {
        // print the error message //
        cout << (mysql_error(&mysql));
switch (sorder)
    {
        case 1:
            if (customer_id == 1) {
                cout << "passing on" << endl;
                if (product_id == 1 && package_id == 1)
                {
                    cout << "passing on" << endl;
                }
                if (qty == 1 && price == 2000) {
                    mysql_init(&mysql);
                    connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
                    cout << "inserting into circular_tbl_trash" << endl;

                    state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_circular_trash_bin` (`trash_id`, `order_id`, `qty`) VALUES ('1',
'1', '10')");
                    /* state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_circular_trash_bin` (`trash_id`, `order_id`, `qty`) VALUES ('2',
'2', '10')");
                    state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_circular_trash_bin` (`trash_id`, `order_id`, `qty`) VALUES ('3',
'3', '10')");*/

                    mysql_init(&mysql);
                    result = mysql_store_result(connection);
                    cout << "insertion successfull" << endl;
                    cout << "sum" << endl;
                    mysql_init(&mysql);

```

```

        connection = mysql_real_connect(&mysql, "localhost",
"root", "root", "jesus", 3306, 0, 0);
        state = mysql_query(connection, "select SUM(qty) from
tbl_circular_trash_bin;");
        mysql_init(&mysql);
        result = mysql_store_result(connection);
switch (sorder)
{
case 2:
    if (customer_id == 1) {
        cout << "pass on" << endl;

        if (product_id == 2 && package_id == 2)
        {
            cout << "pass on" << endl;
        }
        if (qty == 1 && price == 1500)
        {
            mysql_init(&mysql);
            mysql_init(&mysql);
            connection = mysql_real_connect(&mysql,
"localhost", "root", "root", "jesus", 3306, 0, 0);
            cout << "inserting into circular_tbl_trash" <<
endl;

            state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_circular_trash_bin` (`trash_id`, `order_id`, `qty`) VALUES ('2',
'1', '10')");

            mysql_init(&mysql);
            result = mysql_store_result(connection);
            cout << "insertion successfull" << endl;
            mysql_init(&mysql);
            connection = mysql_real_connect(&mysql,
"localhost", "root", "root", "jesus", 3306, 0, 0);
            state = mysql_query(connection, "select
SUM(qty) from tbl_circular_trash_bin;");
            mysql_init(&mysql);
            result = mysql_store_result(connection);
            mysql_init(&mysql);
            connection = mysql_real_connect(&mysql,
"localhost", "root", "root", "jesus", 3306, 0, 0);
            state = mysql_query(connection, "select
SUM(qty) from tbl_circular_trash_bin;");
            mysql_init(&mysql);
            result = mysql_store_result(connection);
            switch (sorder)
            {
                case 3:
                    if (customer_id == 1) {
                        cout << "pass on" << endl;

                        if (product_id == 3 && package_id == 3)
                        {
                            cout << "pass on" << endl;
                        }
                        if (qty == 1 && price == 1000)
                        {
                            mysql_init(&mysql);
                            mysql_init(&mysql);
                            connection =
mysql_real_connect(&mysql, "localhost", "root", "root", "jesus", 3306, 0, 0);

                            cout << "inserting into circular_tbl_trash" << endl;

```



```

state = mysql_query(connection, "INSERT INTO `jesus`.`tbl_circular_trash_bin`
(`trash_id`, `order_id`, `qty`) VALUES ('3', '1', '10')");
mysql_init(&mysql);
result =
mysql_store_result(connection);
endl;
cout << "insertion successfull" <<
mysql_init(&mysql);

connection = mysql_real_connect(&mysql, "localhost", "root", "root", "jesus",
3306, 0, 0);

state = mysql_query(connection, "select SUM(qty) from
tbl_circular_trash_bin;");
mysql_init(&mysql);
result =
mysql_store_result(connection);

switch (sorder)
{
case 4:
if (customer_id == 2) {
cout << "pass on" << endl;

if (product_id == 1 &&
package_id == 1)
{
cout << "pass on" <<
}
if (qty == 1 && price ==
1500)
{
mysql_init(&mysql);
mysql_init(&mysql);
connection =
mysql_real_connect(&mysql, "localhost", "root", "root", "jesus", 3306, 0, 0);

cout << "inserting into circular_tbl_trash" << endl;

state = mysql_query(connection, "INSERT INTO `jesus`.`tbl_circular_trash_bin`
(`trash_id`, `order_id`, `qty`) VALUES ('1', '2', '10')");
mysql_init(&mysql);

result = mysql_store_result(connection);

cout << "insertion successfull" << endl;

mysql_init(&mysql);

connection = mysql_real_connect(&mysql, "localhost", "root", "root", "jesus",
3306, 0, 0);

state = mysql_query(connection, "select SUM(qty) from
tbl_circular_trash_bin;");

mysql_init(&mysql);

result = mysql_store_result(connection);

switch (sorder)
{
case 5:

```

```

    if (customer_id == 2) {
cout << "pass on" << endl;

if (product_id == 2 && package_id == 2)
{
cout << "pass on" << endl;
}
if (qty == 1 && price == 1000)
{
mysql_init(&mysql);

connection = mysql_real_connect(&mysql, "localhost", "root", "root", "jesus",
3306, 0, 0);

cout << "inserting into circular_tbl_trash" << endl;

state = mysql_query(connection, "INSERT INTO `jesus`.`tbl_circular_trash_bin`
(`trash_id`, `order_id`, `qty`) VALUES ('2', '2', '10')");

mysql_init(&mysql);
mysql_init(&mysql);

result = mysql_store_result(connection);

cout << "insertion successfull" << endl;

mysql_init(&mysql);

connection = mysql_real_connect(&mysql, "localhost", "root", "root", "jesus",
3306, 0, 0);

mysql_query(connection, "select SUM(qty) from tbl_circular_trash_bin;");
state =
mysql_init(&mysql);

result = mysql_store_result(connection);

switch (sorder) {

case 6:

if (customer_id == 2) {

cout << "passing on" << endl;

if (product_id == 3 && package_id == 3)
{

cout << "passing on" << endl;

}

}
}

```

```

    if (qty == 1 && price == 2000) {
mysql_init(&mysql);

mysql_init(&mysql);

connection = mysql_real_connect(&mysql, "localhost", "root", "root", "jesus",
3306, 0, 0);

cout << "inserting into circular_tbl_trash" << endl;

    state = mysql_query(connection, "INSERT INTO `jesus`.`tbl_circular_trash_bin`
(`trash_id`, `order_id`, `qty`) VALUES ('3', '2', '10')");

mysql_init(&mysql);

mysql_init(&mysql);

result = mysql_store_result(connection);

    cout << "insertion successfull" << endl;

mysql_init(&mysql);

connection = mysql_real_connect(&mysql, "localhost", "root", "root", "jesus",
3306, 0, 0);

    state = mysql_query(connection, "select SUM(qty) from
tbl_circular_trash_bin;");

mysql_init(&mysql);

    result = mysql_store_result(connection);

switch (sorder)
{case 7:

    if (customer_id == 3) {

cout << "passing on" << endl;

if (product_id == 1 && package_id == 1)

{

cout << "passing on" << endl;

}

if (qty == 1 && price == 2000) {

mysql_init(&mysql);

connection = mysql_real_connect(&mysql, "localhost", "root", "root", "jesus",
3306, 0, 0);

cout << "inserting into circular_tbl_trash" << endl;

state = mysql_query(connection, "INSERT INTO `jesus`.`tbl_circular_trash_bin`
(`trash_id`, `order_id`, `qty`) VALUES ('1', '3', '10')");

mysql_init(&mysql);

```

```

mysql_init(&mysql);

result = mysql_store_result(connection);

cout << "insertion successfull" << endl;

mysql_init(&mysql);

connection = mysql_real_connect(&mysql, "localhost", "root", "root", "jesus",
3306, 0, 0);

state = mysql_query(connection, "select SUM(qty) from
tbl_circular_trash_bin;");

mysql_init(&mysql);

result = mysql_store_result(connection);

switch (sorder)
{
case 8:
if (customer_id == 2) {
cout << "pass on" << endl;

if (product_id == 2 && package_id == 2)
{
cout << "pass on" << endl;
}

if (qty == 1 && price == 1500)
{
mysql_init(&mysql);

connection = mysql_real_connect(&mysql, "localhost", "root", "root", "jesus",
3306, 0, 0);

cout << "inserting into circular_tbl_trash" << endl;

state = mysql_query(connection, "INSERT INTO `jesus`.`tbl_circular_trash_bin`
(`trash_id`, `order_id`, `qty`) VALUES ('2', '3', '10')");

mysql_init(&mysql);

mysql_init(&mysql);

result = mysql_store_result(connection);

cout << "insertion successfull" << endl;

mysql_init(&mysql);

connection = mysql_real_connect(&mysql, "localhost", "root", "root", "jesus",
3306, 0, 0);

```

```

state = mysql_query(connection, "select SUM(qty) from
tbl_circular_trash_bin;");

mysql_init(&mysql);

result = mysql_store_result(connection);

switch (sorder)

{case 9:

if (customer_id == 3) {

cout << "pass on" << endl;

if (product_id == 3 && package_id == 3)

{

cout << "pass on" << endl;

}

if (qty == 1 && price == 1000)

{

mysql_init(&mysql);

connection = mysql_real_connect(&mysql, "localhost", "root", "root", "jesus",
3306, 0, 0);

state = mysql_query(connection, "INSERT INTO `jesus`.`tbl_circular_trash_bin`
(`trash_id`, `order_id`, `qty`) VALUES ('3', '3', '10')");

cout << "inserting into circular_tbl_trash" << endl;

mysql_init(&mysql);

mysql_init(&mysql);

result = mysql_store_result(connection);

cout << "insertion successfull" << endl;

mysql_init(&mysql);

connection = mysql_real_connect(&mysql, "localhost", "root", "root", "jesus",
3306, 0, 0);

state = mysql_query(connection, "select SUM(qty) from
tbl_circular_trash_bin;");

mysql_init(&mysql);

result = mysql_store_result(connection);

}

}

```

```

}
}
}
}
}
}
}

```

```

}
}
}
}

```

```

//collection truck header//
#pragma once
#include<iostream>
#include<string.h>
#include "delivery.h"
#include"db connect.h"
#include"trash.h"
#include"delivery.h"
#include"Collectiontruck.h"
#include"Circular_trash_bin.h"

using namespace std;

class Collectiontruck :public dbconnect,public delivery,public
circular_trash_bin
{
public:
    void display();
    void truck();
public:
    int state;
    MYSQL* connection;
};

//collection truck cpp//
#include<iostream>
#include<string.h>
#include"delivery.h"
#include "db connect.h"
#include"Collectiontruck.h"
#include"trash.h"
#include"db connect.h"
#include"delivery.h"
#include"Circular_trash_bin.h"
using namespace std;

void Collectiontruck::display() {
    cout << "the collection truck on the way to collect trash" << endl;
}
void Collectiontruck::truck() {
    cout << "truck on the way to reach green company" << endl;
}

```

```

MYSQL_RES* result;
MYSQL_ROW row;
MYSQL* connection, mysql;
int state;
int rc = 0;
int row_num = 0;

// connect to the MySQL database//
mysql_init(&mysql);
connection = mysql_real_connect(&mysql, "localhost", "root", "root",
"jesus", 3306, 0, 0);
// check for a connection error //
if (connection == NULL) {
    // print the error message //
    cout << (mysql_error(&mysql));
    //return 0;//
}
mysql_init(&mysql);
connection = mysql_real_connect(&mysql, "localhost", "root", "root",
"jesus", 3306, 0, 0);

state = mysql_query(connection, "INSERT INTO `jesus`.`tbl_collection`
(`truck_id`, `trash_id`) VALUES ('1', '1');");
}

//green company header//
#pragma once
#include<windows.h>
#include<mysql.h>
#include<iostream>
#include<string.h>
#include<sqlext.h>
#include<stdio.h>
#include<iostream>
#include<string.h>
#include"db connect.h"

using namespace std;

class green_company:public dbconnect{
public:
    green_company();
    int state;
    MYSQL* connection;
    virtual void display();
public:
    void insertGreenCompany();
    ~green_company();
};

//green company cpp//
#include<windows.h>
#include<mysql.h>
#include<iostream>
#include<string.h>
#include<sqlext.h>
#include<stdio.h>
#include<iostream>

```

```

#include<string.h>
#include "green_company.h"
#include"db connect.h"

using namespace std;

green_company::green_company() {};
void green_company::display() {
    cout << "the collection truck has reached the green company" << endl;
}
void green_company::insertGreenCompany() {
    MYSQL_RES* result;
    MYSQL_ROW row;
    MYSQL* connection, mysql;
    int state;
    int rc = 0;
    int row_num = 0;

    // connect to the mySQL database//
    mysql_init(&mysql);
    connection = mysql_real_connect(&mysql, "localhost", "root", "root",
"jesus", 3306, 0, 0);
    // check for a connection error //
    if (connection == NULL) {
        // print the error message //
        cout << (mysql_error(&mysql));
        //return 0;//
    }
    mysql_init(&mysql);
    connection = mysql_real_connect(&mysql, "localhost", "root", "root",
"jesus", 3306, 0, 0);

    state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_green_company` (`comapany_id`, `truck_id`) VALUES ('1', '1')");

    cout << "success from green company" << endl;
}
green_company::~green_company() {};

//delivery header//
#pragma once
#include<iostream>
#include<string.h>
#include"green_company.h"
#include"db connect.h"
using namespace std;
class delivery :public green_company,public dbconnect {
public:
    void display();
    void truck();
    delivery();
    ~delivery();
};

//deliver cpp//
#include<iostream>
#include<string.h>
#include"green_company.h"

```



```

#include"delivery.h"
#include "db connect.h"

using namespace std;

delivery::delivery(){}
void delivery::display() {
    cout << "the delivery truck on the way" << endl;
}
void delivery::truck() {
    cout << "truck on the way to reach regeneration unit" << endl;
    MYSQL_RES* result;
    MYSQL_ROW row;
    MYSQL* connection, mysql;
    int state;
    int rc = 0;
    int row_num = 0;

    // connect to the mySQL database//
    mysql_init(&mysql);
    connection = mysql_real_connect(&mysql, "localhost", "root", "root",
"jesus", 3306, 0, 0);
    // check for a connection error //
    if (connection == NULL) {
        // print the error message //
        cout << (mysql_error(&mysql));
        //return 0;//
    }
    mysql_init(&mysql);
    connection = mysql_real_connect(&mysql, "localhost", "root",
"root", "jesus", 3306, 0, 0);

    state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_delivery` (`DeliveryTruck_id`, `company_id`) VALUES ('1', '1');");
}
delivery::~~delivery() {}

// regeneration unit header//
#pragma once
#include<windows.h>
#include<mysql.h>
#include<iostream>
#include<string.h>
#include<sqlext.h>
#include<stdio.h>
#include<iostream>
#include<string.h>
#include"db connect.h"
#include"green_company.h"
#include"delivery.h"

using namespace std;
class regeneration_unit:public dbconnect,public green_company,public delivery{
public:
    regeneration_unit();
    int state;
    MYSQL* connection;
public:
    void destiny();

```

```

        ~regeneration_unit();
};

//regeneration unit cpp//
#include<windows.h>
#include<mysql.h>
#include<iostream>
#include<string.h>
#include<sqlext.h>
#include<stdio.h>
#include<iostream>
#include<string.h>
#include "regeneration_unit.h"
#include"green_company.h"
#include"delivery.h"
#include"db connect.h"
#include"green_company.h"
#include"delivery.h"
using namespace std;

regeneration_unit::regeneration_unit() {};
void regeneration_unit::destiny() {
    MYSQL_RES* result;
    MYSQL_ROW row;
    MYSQL* connection, mysql;
    int state;
    int rc = 0;
    int row_num = 0;

    // connect to the mySQL database//
    mysql_init(&mysql);
    connection = mysql_real_connect(&mysql, "localhost", "root", "root",
"jesus", 3306, 0, 0);
    // check for a connection error //
    if (connection == NULL) {
        // print the error message //
        cout << (mysql_error(&mysql));
        //return 0;//
    }
    mysql_init(&mysql);
    connection = mysql_real_connect(&mysql, "localhost", "root", "root",
"jesus", 3306, 0, 0);
    /*state = mysql_query(connection, "INSERT INTO
`jesus`.`tbl_regeneration_unit`VALUES( 1,1,'chennai')");*/

state = mysql_query(connection, "INSERT INTO `jesus`.`tbl_regeneration_unit`
(`regeneration_unit_id`, `DeliveryTruck_id`, `location`) VALUES ('1', '1',
'chennai')");

cout << "Reg unit success!" << endl;
}
regeneration_unit::~~regeneration_unit() {};

//MY SQL Scrip//

-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;

```

```

SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,E
RROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

```

```

-----
-- Schema mydb
-----
-----
-- Schema jesus
-----

```

```

-----
-- Schema jesus
-----

```

```

CREATE SCHEMA IF NOT EXISTS `jesus` DEFAULT CHARACTER SET latin1 ;
USE `jesus` ;

```

```

-----
-- Table `jesus`.`tbl_customer`
-----

```

```

CREATE TABLE IF NOT EXISTS `jesus`.`tbl_customer` (
  `customer_id` INT(11) NOT NULL AUTO_INCREMENT,
  `customer_name` VARCHAR(45) NOT NULL,
  `customer_location` VARCHAR(45) NOT NULL,
  `customer_contact` INT(11) NOT NULL,
  PRIMARY KEY (`customer_id`),
  UNIQUE INDEX `customer_contact_UNIQUE` (`customer_contact` ASC) ,
  UNIQUE INDEX `customer_id_UNIQUE` (`customer_id` ASC) )
ENGINE = InnoDB
AUTO_INCREMENT = 4
DEFAULT CHARACTER SET = latin1;

```

```

-----
-- Table `jesus`.`tbl_package`
-----

```

```

CREATE TABLE IF NOT EXISTS `jesus`.`tbl_package` (
  `package_id` INT(11) NOT NULL AUTO_INCREMENT,
  `package_type` VARCHAR(45) NOT NULL,
  `product_type` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`package_id`),
  UNIQUE INDEX `product_id` (`package_id` ASC) )
ENGINE = InnoDB
AUTO_INCREMENT = 4
DEFAULT CHARACTER SET = latin1;

```

```

-----
-- Table `jesus`.`tbl_online_shop`
-----

```

```

CREATE TABLE IF NOT EXISTS `jesus`.`tbl_online_shop` (
  `product_id` INT(11) NOT NULL AUTO_INCREMENT,
  `product_name` VARCHAR(45) NOT NULL,
  `product_price` INT(11) NOT NULL,
  PRIMARY KEY (`product_id`),
  UNIQUE INDEX `product_id_UNIQUE` (`product_id` ASC) )
ENGINE = InnoDB
AUTO_INCREMENT = 4
DEFAULT CHARACTER SET = latin1;

```

```

-----
-- Table `jesus`.`tbl_order`
-----

```

```

-----
CREATE TABLE IF NOT EXISTS `jesus`.`tbl_order` (
  `order_id` INT(11) NOT NULL AUTO_INCREMENT,
  `customer_id` INT(11) NOT NULL,
  `product_id` INT(11) NOT NULL,
  `package_id` INT(11) NOT NULL,
  `quantity` INT(11) NOT NULL,
  `price final` INT(11) NOT NULL,
  PRIMARY KEY (`order_id`),
  UNIQUE INDEX `order_id_UNIQUE` (`order_id` ASC) ,
  INDEX `customer_id_idx` (`customer_id` ASC) ,
  INDEX `product_id_idx` (`product_id` ASC) ,
  INDEX `package_id` (`package_id` ASC) ,
  CONSTRAINT `customer_id`
    FOREIGN KEY (`customer_id`)
    REFERENCES `jesus`.`tbl_customer` (`customer_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `package_id`
    FOREIGN KEY (`package_id`)
    REFERENCES `jesus`.`tbl_package` (`package_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `product_id`
    FOREIGN KEY (`product_id`)
    REFERENCES `jesus`.`tbl_online_shop` (`product_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 2
DEFAULT CHARACTER SET = latin1;

```

```

-----
-- Table `jesus`.`tbl_circular_trash_bin`
-----

```

```

CREATE TABLE IF NOT EXISTS `jesus`.`tbl_circular_trash_bin` (
  `trash_id` INT(11) NOT NULL AUTO_INCREMENT,
  `order_id` INT(11) NOT NULL,
  `qty` INT(11) NULL DEFAULT NULL,
  PRIMARY KEY (`trash_id`),
  INDEX `order_id_idx` (`order_id` ASC) ,
  CONSTRAINT `order_id`
    FOREIGN KEY (`order_id`)
    REFERENCES `jesus`.`tbl_order` (`order_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 2
DEFAULT CHARACTER SET = latin1;

```

```

-----
-- Table `jesus`.`tbl_collection`
-----

```

```

CREATE TABLE IF NOT EXISTS `jesus`.`tbl_collection` (
  `truck_id` INT(11) NOT NULL,
  `trash_id` INT(11) NOT NULL,
  PRIMARY KEY (`truck_id`),
  UNIQUE INDEX `truck_id_UNIQUE` (`truck_id` ASC) ,
  INDEX `trash_id_idx` (`trash_id` ASC) ,
  CONSTRAINT `trash_id`
    FOREIGN KEY (`trash_id`)

```

```

REFERENCES `jesus`.`tbl_circular_trash_bin` (`trash_id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;

```

```

-----
-- Table `jesus`.`tbl_green_company`
-----

```

```

CREATE TABLE IF NOT EXISTS `jesus`.`tbl_green_company` (
  `comapany_id` INT(11) NOT NULL AUTO_INCREMENT,
  `truck_id` INT(11) NULL DEFAULT NULL,
  PRIMARY KEY (`comapany_id`),
  UNIQUE INDEX `comapany_id_UNIQUE` (`comapany_id` ASC) ,
  INDEX `truck_id_idx` (`truck_id` ASC) ,
  CONSTRAINT `truck_id`
    FOREIGN KEY (`truck_id`)
    REFERENCES `jesus`.`tbl_collection` (`truck_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 2
DEFAULT CHARACTER SET = latin1;

```

```

-----
-- Table `jesus`.`tbl_delivery`
-----

```

```

CREATE TABLE IF NOT EXISTS `jesus`.`tbl_delivery` (
  `DeliveryTruck_id` INT(11) NOT NULL AUTO_INCREMENT,
  `company_id` INT(11) NOT NULL,
  PRIMARY KEY (`DeliveryTruck_id`),
  UNIQUE INDEX `DeliveryTruck_id_UNIQUE` (`DeliveryTruck_id` ASC) ,
  INDEX `comapny_id_idx` (`company_id` ASC) ,
  CONSTRAINT `comapny_id`
    FOREIGN KEY (`company_id`)
    REFERENCES `jesus`.`tbl_green_company` (`comapany_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 2
DEFAULT CHARACTER SET = latin1;

```

```

-----
-- Table `jesus`.`tbl_regeneration_unit`
-----

```

```

CREATE TABLE IF NOT EXISTS `jesus`.`tbl_regeneration_unit` (
  `regeneration_unit_id` INT(11) NOT NULL AUTO_INCREMENT,
  `DeliveryTruck_id` INT(11) NOT NULL,
  `location` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`regeneration_unit_id`),
  UNIQUE INDEX `regeneration_unit_id_UNIQUE` (`regeneration_unit_id` ASC) ,
  INDEX `DeliveryTruck_id_idx` (`DeliveryTruck_id` ASC) ,
  CONSTRAINT `DeliveryTruck_id`
    FOREIGN KEY (`DeliveryTruck_id`)
    REFERENCES `jesus`.`tbl_delivery` (`DeliveryTruck_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 2
DEFAULT CHARACTER SET = latin1;

```

```
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```