VILNIAUS GEDIMINO TECHNICAL UNIVERSITY

FACULTY OF FUNDAMENTAL SCIENCES

DEPARTMENT OF INFORMATION TECHNOLOGIES

ELEKTRONIC STORES SHOPPING DATABASE

Author: ITfuc-19 gr. st. Abdul Hannan Ayubi

Checked by : lect. Andrej Tunevič

TABLE OF CONTENTS

1. THE TASK OF THE COURSE PROJECT

2. The topic of the course work may be freely chosen and should be approved by the lecturer.
3. The subject area must be described and presented in accordance with the requirements presented in this document.
4. DB must have at least 5 entities>tables.
5. The following diagrams should be presented
    - Conceptual model (each entity must have visible attribute names, descriptions, types…). All relations must have names according to created DB and subject area
    - Logical model.
    - Physical model.
6. Prepare documentation of created database (*Oracle SQL Developer Data Modeler*).
7. Design user interfaces (APEX) using various graphical elements such as graphs, calendars, etc..
8. Create database users.
9. The course project must be prepared according to VGTU requirements and uploaded to Moodle system until 8th of December, 2021.

2. ANALYSIS OF THE SUBJECT AREA

With this Database which I create I would like to archive my stores total revenue, customer information, employee information, products, producer, sold items, and get information about the producer which sold most items The category which sold most etc. This database is need for shop owners, customer who needs some specific information about the shops, products, producers, and employee of shops and producers who want to get information about in their products which sold most in and where, what type of product sold most etc. The worker of store who can easily deal with customers and orders. The second aim of creating this Database is to give some specific information for customers as well. This Database provides the location of shops, The name of shops, The total customer of the shops, category which sold most, producers who have products etc. This will help customer to give some specific decision about their orders with products. For example, as an owner of shops he/she can see the total revenue of him/her shops, Total employee of shops, Total salary which given to the employees, total products and name of products, total producers who he sells their products, total category of product which he / she has in the shops. This information can help the shop owner to buy new items in a specific area, recruitment new employee or not etc.

On the other hand, this database can help customers as well for example Adrian wants to buy a phone and he can get information about the products and the producers with created database, he can get information about phones category which product sold most in where and what is the price of products. This information will help Adrian to give some idea about his needs.

2.1. Description of the subject area, visual image

The area of the database is shopping about technology products, which in real world we can have many technology shops such is technotronika, mediamarkt, Electromarkts etc. As we can see in the pic-1 and pic-2

2.2 Identification of requirements for the designed DB

- Administrator should Add, Delete, and Update in each section except the main page.
- There should be three types of User Administrator, Developer (worker) and END User (reader).
- Developer (Worker) should be able to add new user , new order new producer , new category to the Database
- Administrator should select the Cities from LOV (List of Values) in Apex when adding the new stores in the list
- Grap, Calendar, Map and Detail description should be appeared in main page.
- In the customer page there should be a PIE graph to show the average of gender of customer for both User types (End- user) , Worker.
- In the Main Page there should be Dynamic section which should show the Total Revenue, Total Salary
- In apex there should be LOV for select cities in store section
- In Customer section there should be popup List of values for select the Customer gender and there should be one gender as a U (Unknown) for select as well.
- In the Store section in the Adress there should be only City list. 5 City only should be appeared. (Vilnius, Kaunas, Warsaw, Riga, and Tallinn).
- The employee menu must show the employee list for the Users with name, Email, and the store where they work.
- There should be a button in production the direct the category and producer menu for Workers user.

- In the main page there should be mapped to show the Cities of our shops. (Warsaw, Vilnius, Kaunas, Riga, Tallinn)
- The menus should be at the top of the page.
- The END user should sort in each page were lists available.

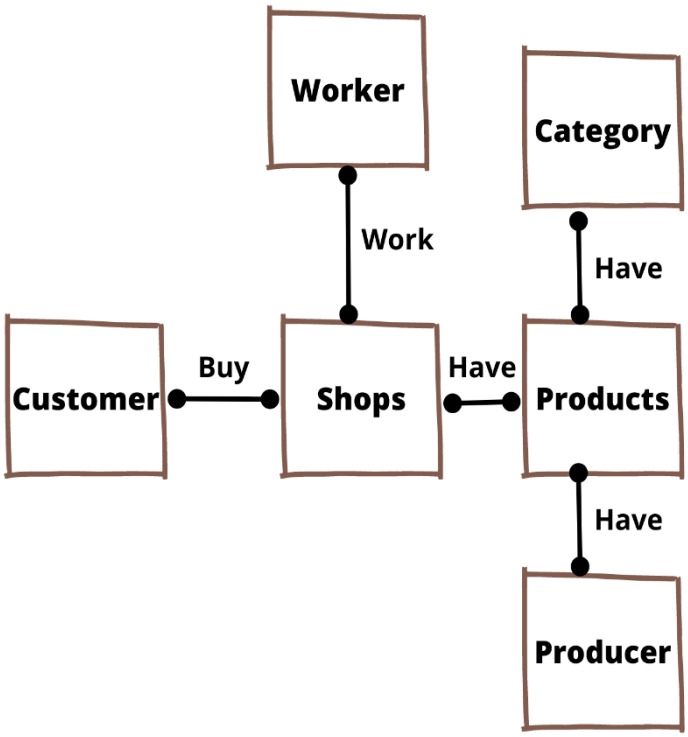
2.3 Functional Requirements

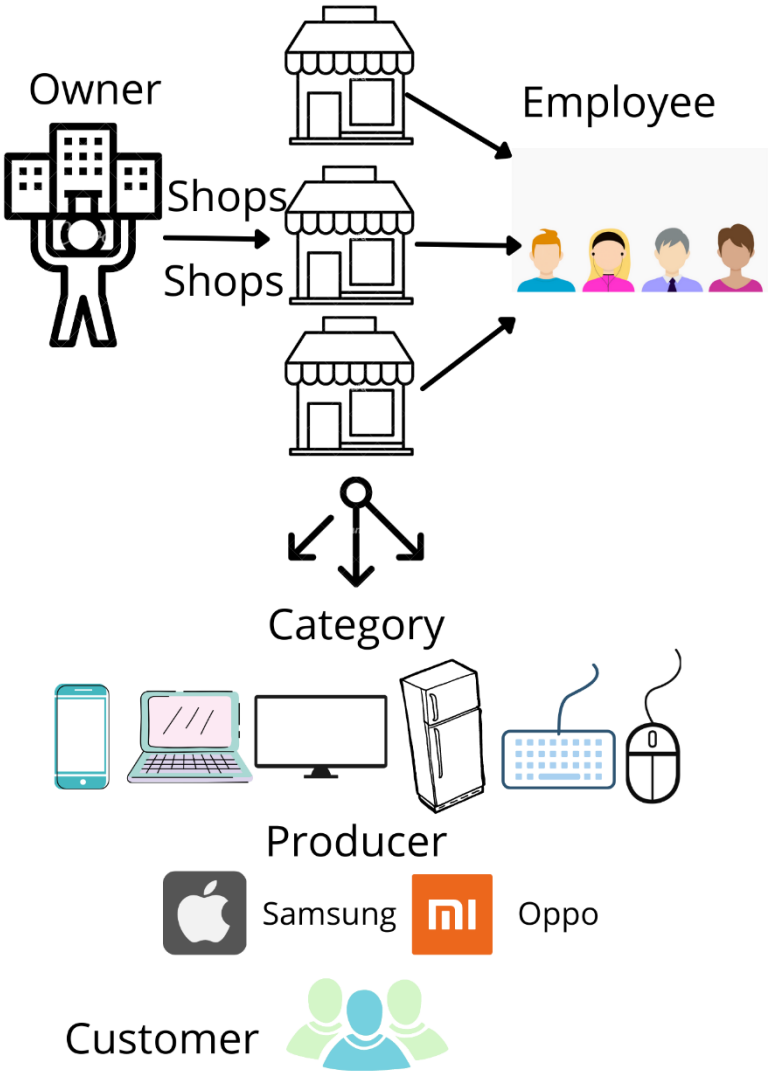
- The total price should be calculated by the selecting the product in the list of products and entering the Quantity and there should be Tax rate of 15% in each product equally.
- The ID of products should be generated automatically when adding by the Administrator and workers.
- Each graph should be Dynamic.
- Only Registered User must be able to be order the products by the Administrator and worker.
- END user should not add new Customer, Employee, Order, Product and Producer
- Each END user should have their own authentication ID and Password which should give by the Administrator in Apex
- The whole data should be saved in the apex database system.
- The total Revenue should be calculated according to the total cost of orders minus the total salary which is given to the employees
- In the Employee page there should be a link for the END user to direct them to the store page
- There should be a print section appear for the END user in the product menu.

3. MODELING STAGES

3.1 Conceptual data model

In this data model as shown in the pictures below there is 6 entities related to my designed database model. In conceptual data model the entities shown without any attributes. I have Worker, shops, Customer, Category, Products, Producers as my entities. Shops have relation between the worker, customer, and products. Workers have connection between shops. Product has connection between the category and producer. There is text next to the line to make the initial modeling phase easier to understand. For example, worker work in shops, customers buy from shop, shops have products. products have category and producer. In the second Image it's a bit clear and real-world example.

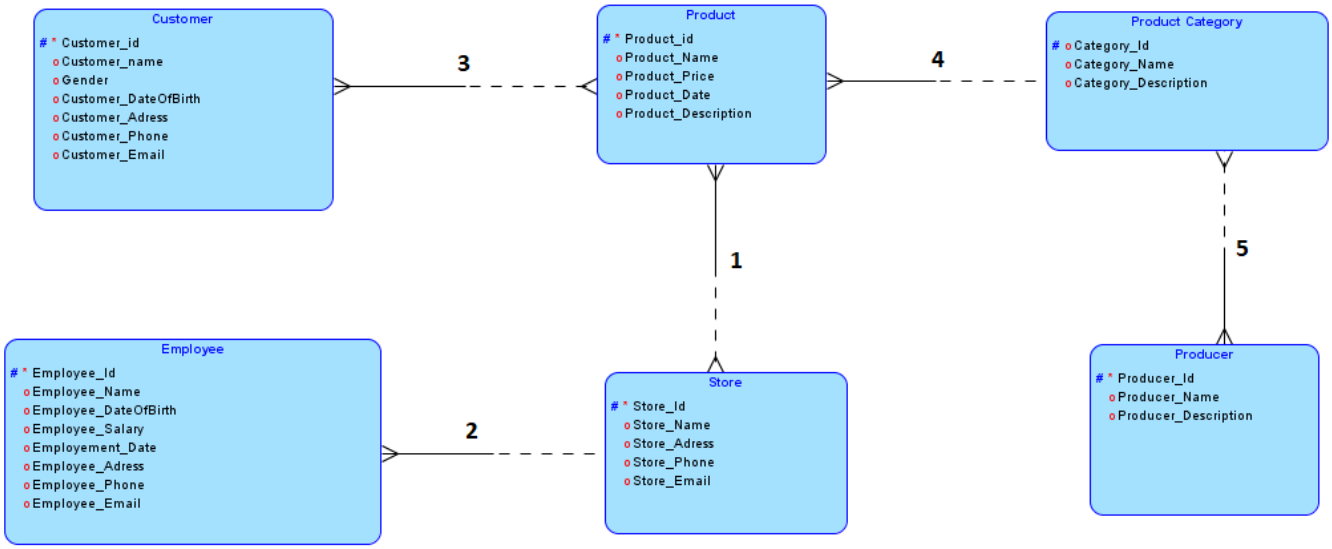




3.2 Logical data model

In this data model stage, it's clear to have relationships between the entities and have attributes in the entities. My main entity is my store entity. I have store entity with 5 attributes. The Store ID is primary key which is numeric value. [1] The relation between product entity is many to many because each store can have many products and each product can be in many stores. That's why we have many to many relationships. For example, there are 3 product which can be in 3 shops at the same time on the other side there 3 shop which can have 3 products at the same time. [2] The relationship between the store and employee is one to many because each employee can work only in one store, but one store can have many employees that's why we have one to many relationships. For Example, Store_101 can have Adraian and Matilda as an employee but Adraian and Matilda can't work in two stores at the same time. [3] The relationship between the product and customer is many to many because each customer can buy many products and each product can be salable to many customers. For example, Adrian can buy product_101 and product_102 and product_101 can be salable for Matilda as well that's why we have many to many relationships. [4] Each product can have one category, but each category can have many products.

For example, in my Database which is about technology products. Phone is a category, and it can have many products in it. iPhone 11 pro, Samsung S21, etc. But Samsung S21 can be only in one category which is phone that's why we have one to many relationships between these entities. [5] The relationship between category and producer is many to many. Each producer can have many category type products and each category can have many producers. For Example Apple inc. is a producer and it can have Phone, TV, Computers as a product category and Phone, TV, Computer are categories and it can have many producers such as Samsung, Xiomi, OPPO, Apple , Nokia and etc. That's why we have many to many relationships between entities.



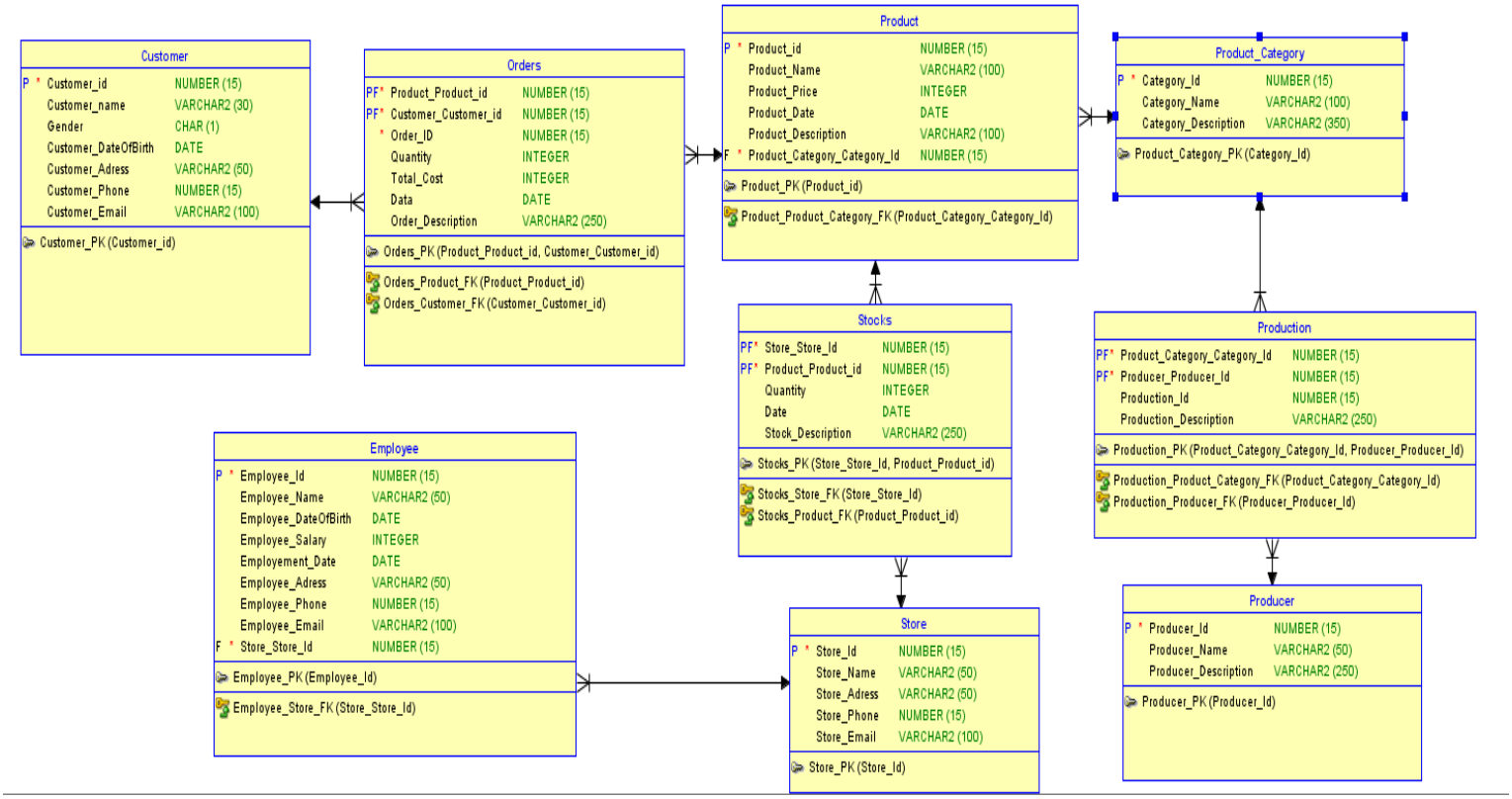
(Logical Data Model)

Oracle Data Modeler

3.3.Physical data model

In this model I have 9 tables with the relationships between the other tables. In previous model which is logical data model the many to many relationships became a table in this data model. Between the customer and product, I have a table called orders. This table consist of its own columns as well as foreign keys from customer and product. I used two foreign keys in my table because I used it in my Apex application. This model consists of columns, and it details as well. For example, Customer Table has customer Id as a primary key and in orders table we get this key and make it foreign keys. Each table has its primary key as well. The relation between store and Employee is one to many relationships and in the primary key of the store became a column in the employee table. With this way we can specify the workplace of employee and we can assign only one shop in employee table which we thought to design in the first phase.

The relation between product and shops became a new table which's stock. The Stock has its own columns as well Quantity column save the quantity of products related to shops. For example, Store_101 has 100 Quantity of product_101. The relation between the category (product_category) and producer is many to many that's why we had a new table here as well which is production table. This table has its own columns as well.



Oracle SQL Developer Data Modeler

DDL File Editor - Oracle Database 11g

Oracle Database 11g | Relational_1 | Generate | Clear

```
CREATE TABLE customer (
    customer_id          NUMBER(15) NOT NULL,
    customer_name        VARCHAR2(30),
    gender               CHAR(1),
    customer_dateofbirth DATE,
    customer_adress      VARCHAR2(50),
    customer_phone       NUMBER(15),
    customer_email       VARCHAR2(100)
);

ALTER TABLE customer ADD CONSTRAINT customer_pk PRIMARY KEY ( customer_id );

CREATE TABLE employee (
    employee_id          NUMBER(15) NOT NULL,
    employee_name        VARCHAR2(50),
    employee_dateofbirth DATE,
    employee_salary      INTEGER,
    employement_date     DATE,
    employee_adress      VARCHAR2(50),
    employee_phone       NUMBER(15),
    employee_email       VARCHAR2(100),
    store_store_id       NUMBER(15) NOT NULL
);

ALTER TABLE employee ADD CONSTRAINT employee_pk PRIMARY KEY ( employee_id );

CREATE TABLE orders (
    product_product_id   NUMBER(15) NOT NULL,
    customer_customer_id NUMBER(15) NOT NULL,
    order_id             NUMBER(15) NOT NULL,
    quantity             INTEGER,
    total_cost           INTEGER,
    data                 DATE,
    order_description    VARCHAR2(250)
);

ALTER TABLE orders ADD CONSTRAINT orders_pk PRIMARY KEY ( product_product_id,
                                                          customer_customer_id );

CREATE TABLE producer (
    producer_id          NUMBER(15) NOT NULL,
    producer_name        VARCHAR2(50),
    producer_description VARCHAR2(250)
);

ALTER TABLE producer ADD CONSTRAINT producer_pk PRIMARY KEY ( producer_id );

CREATE TABLE product (
    product_id                  NUMBER(15) NOT NULL,
    product_name                VARCHAR2(100),
    product_price               INTEGER,
    product_date                DATE,
    product_description         VARCHAR2(100),
    product_category_category_id NUMBER(15) NOT NULL
);
```

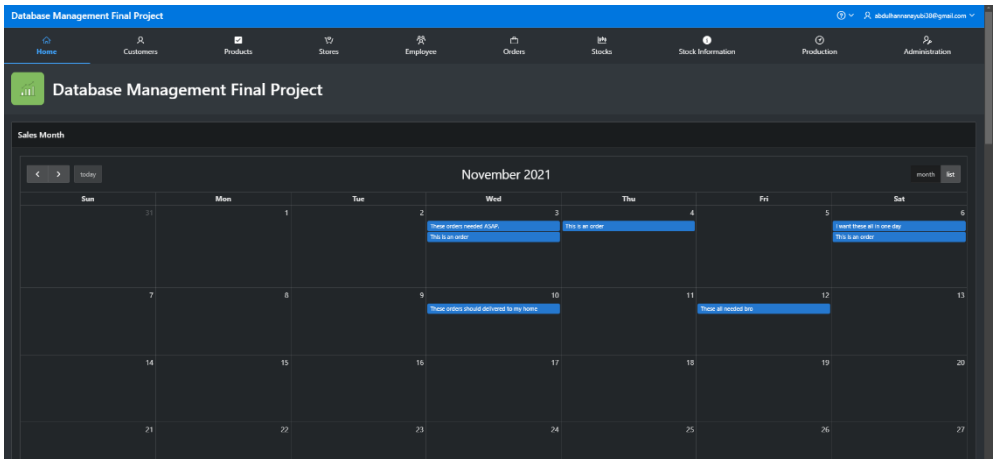
```
CREATE TABLE product_category (
    category_id          NUMBER(15) NOT NULL,
    category_name        VARCHAR2(100),
    category_description VARCHAR2(350)
);

ALTER TABLE product_category ADD CONSTRAINT product_category_pk PRIMARY KEY ( category_id );

CREATE TABLE production (
    product_category_category_id NUMBER(15) NOT NULL,
    producer_producer_id         NUMBER(15) NOT NULL,
    production_id                NUMBER(15),
    production_description       VARCHAR2(250)
);

ALTER TABLE production ADD CONSTRAINT production_pk PRIMARY KEY ( product_category_category_id,
                                                                  producer_producer_id );

CREATE TABLE stocks (
    store_store_id     NUMBER(15) NOT NULL,
    product_product_id NUMBER(15) NOT NULL,
    quantity           INTEGER,
    "Date"             DATE,
    stock_description  VARCHAR2(250)
);

ALTER TABLE stocks ADD CONSTRAINT stocks_pk PRIMARY KEY ( store_store_id,
                                                          product_product_id );

CREATE TABLE store (
    store_id     NUMBER(15) NOT NULL,
    store_name   VARCHAR2(50),
    store_adress VARCHAR2(50),
    store_phone  NUMBER(15),
    store_email  VARCHAR2(100)
);

ALTER TABLE store ADD CONSTRAINT store_pk PRIMARY KEY ( store_id );

ALTER TABLE employee
    ADD CONSTRAINT employee_store_fk FOREIGN KEY ( store_store_id )
        REFERENCES store ( store_id );

ALTER TABLE orders
    ADD CONSTRAINT orders_customer_fk FOREIGN KEY ( customer_customer_id )
        REFERENCES customer ( customer_id );

ALTER TABLE orders
    ADD CONSTRAINT orders_product_fk FOREIGN KEY ( product_product_id )
        REFERENCES product ( product_id );

ALTER TABLE product
    ADD CONSTRAINT product_product_category_fk FOREIGN KEY ( product_category_category_id )
        REFERENCES product_category ( category_id );
```

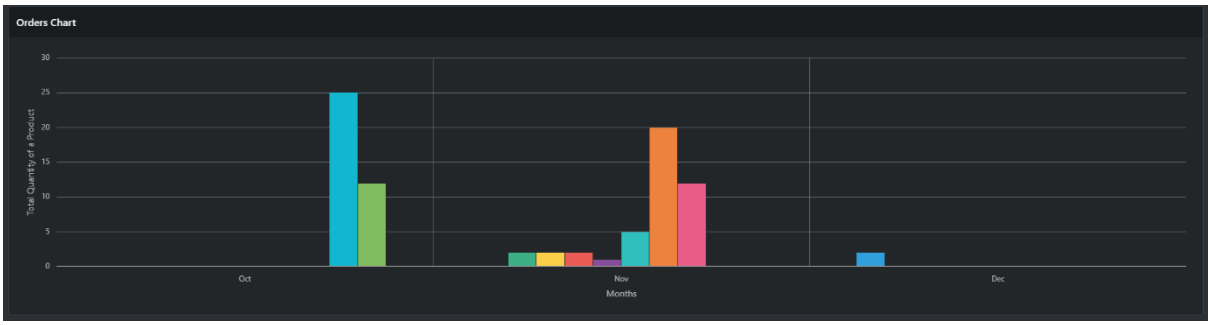
```
ALTER TABLE product
    ADD CONSTRAINT product_product_category_fk FOREIGN KEY ( product_category_category_id )
        REFERENCES product_category ( category_id );

ALTER TABLE production
    ADD CONSTRAINT production_producer_fk FOREIGN KEY ( producer_producer_id )
        REFERENCES producer ( producer_id );

ALTER TABLE production
    ADD CONSTRAINT production_product_category_fk FOREIGN KEY ( product_category_category_id )
        REFERENCES product_category ( category_id );

ALTER TABLE stocks
    ADD CONSTRAINT stocks_product_fk FOREIGN KEY ( product_product_id )
        REFERENCES product ( product_id );

ALTER TABLE stocks
    ADD CONSTRAINT stocks_store_fk FOREIGN KEY ( store_store_id )
        REFERENCES store ( store_id );
```

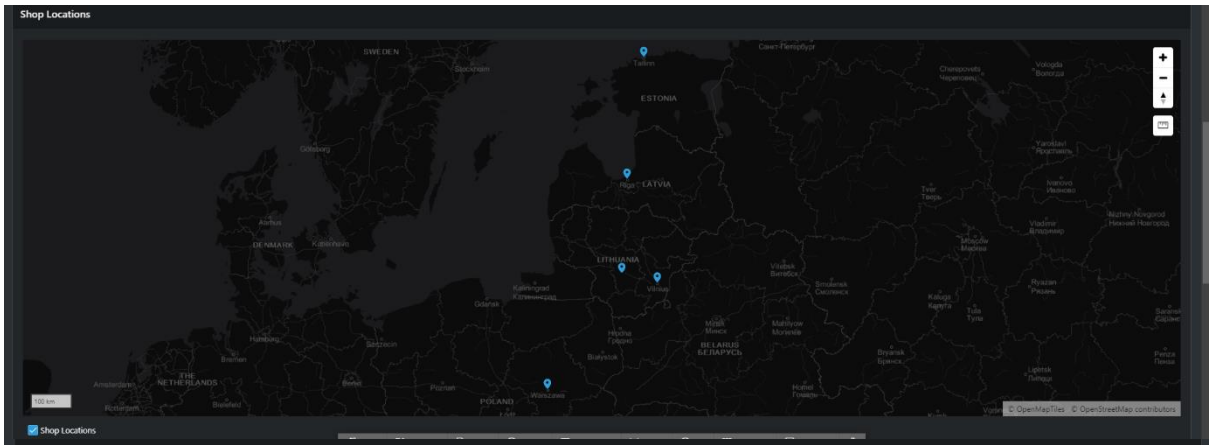
```
-- Oracle SQL Developer Data Modeler Summary Report:
--
-- CREATE TABLE                             9
-- CREATE INDEX                             0
-- ALTER TABLE                             17
-- CREATE VIEW                              0
-- ALTER VIEW                               0
-- CREATE PACKAGE                           0
-- CREATE PACKAGE BODY                      0
-- CREATE PROCEDURE                         0
-- CREATE FUNCTION                          0
-- CREATE TRIGGER                           0
-- ALTER TRIGGER                            0
-- CREATE COLLECTION TYPE                   0
-- CREATE STRUCTURED TYPE                   0
-- CREATE STRUCTURED TYPE BODY              0
-- CREATE CLUSTER                           0
-- CREATE CONTEXT                           0
-- CREATE DATABASE                          0
-- CREATE DIMENSION                         0
-- CREATE DIRECTORY                         0
-- CREATE DISK GROUP                        0

-- ORDS DROP SCHEMA                         0
-- ORDS ENABLE SCHEMA                       0
-- ORDS ENABLE OBJECT                       0
--
-- ERRORS                                   0
-- WARNINGS                                 0
```

These codes generated by the oracle SQL Developer. In this photo the generated code corresponding the tables in the design and "Alter table employee add constraint employee_store_fk Foreign Key (Store_store_id) references store (store_id)" this code connect two table with each other's. In this example Employee table with store table. Employee has a store_id foreign key as a column in the employee table.

4. USER INTERFACE DESIGN







Revenue Information

Total Order
10

Total Customer
8

Total Cost Of Orders ($)
30989

Total Salary Of Emlpoyee
7250

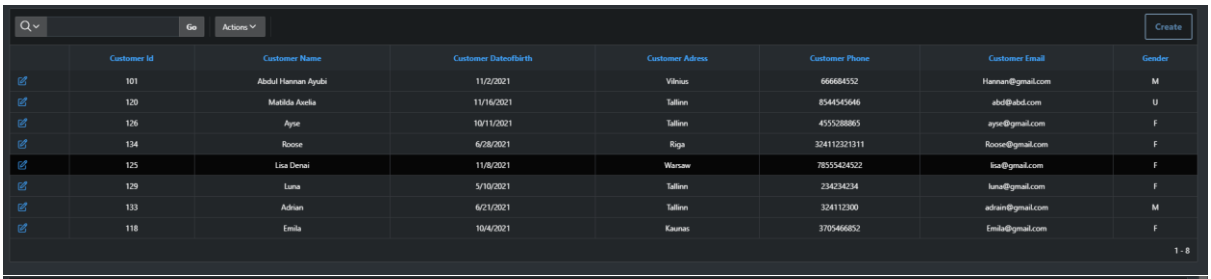
Total Revenue ($)
23739

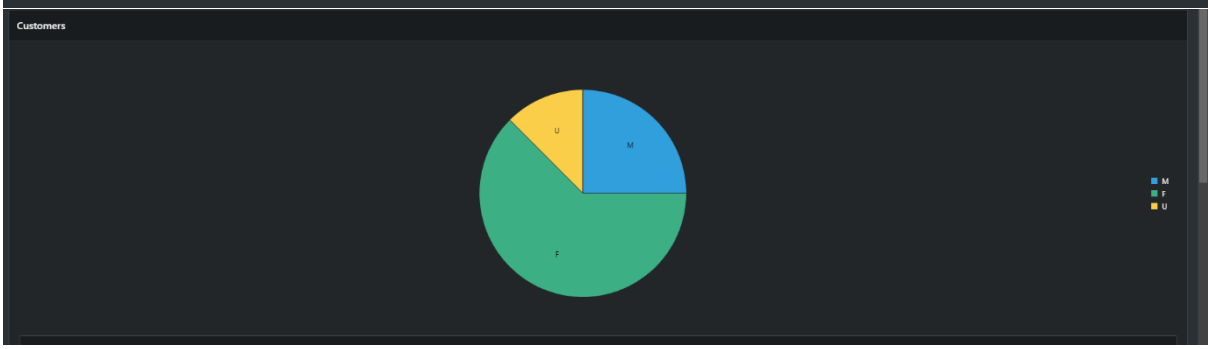
Companies & Products

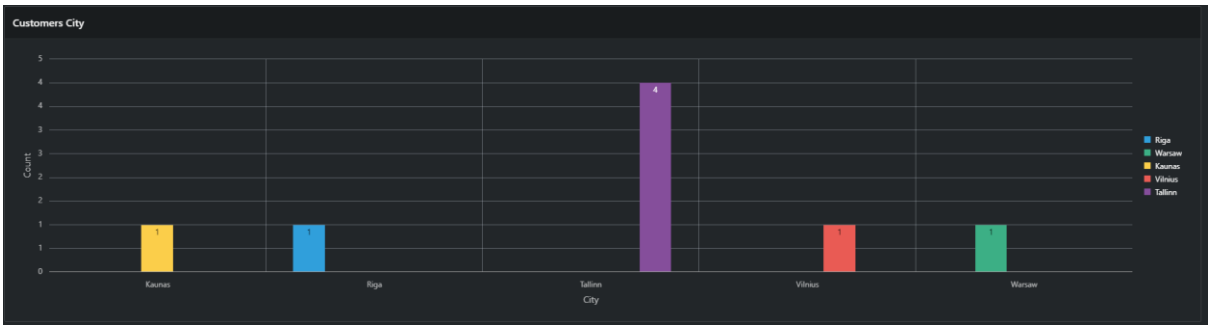
Total Product
Xiomi X20:Abra A5:IphoneSE:Samsung S21:Iphone 7:Samsung A9 (2018):Iphone X:Dell Monitor:Iphone 8:Samsung A51:Apple M5

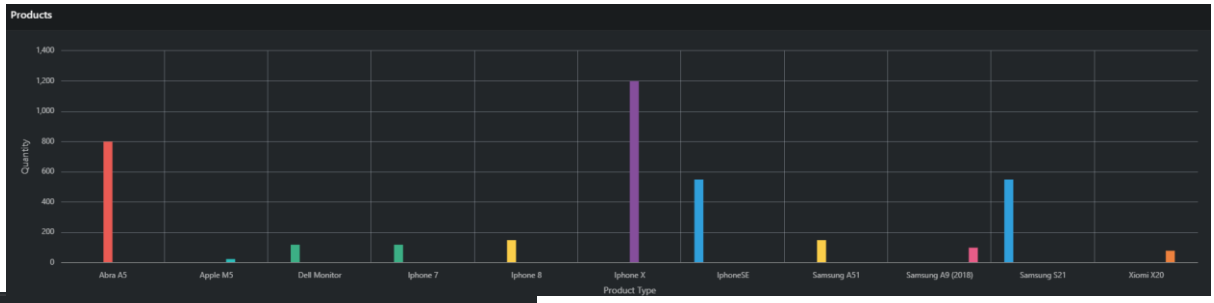
Total Product Type
Mouse:Headphone:Phones:Computers:Tablets:Cameras:Keyboards:Monitor

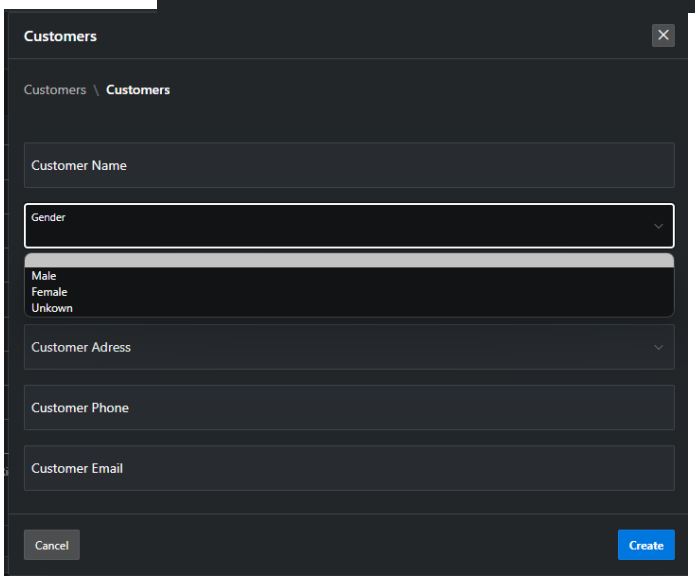
Total Companies
Oppo:Samsung Company:Xiomi:Sony:Canon Inc.:Dell Technologies:Apple Inc.:HP:Microsoft:RedMi:Google:IBM

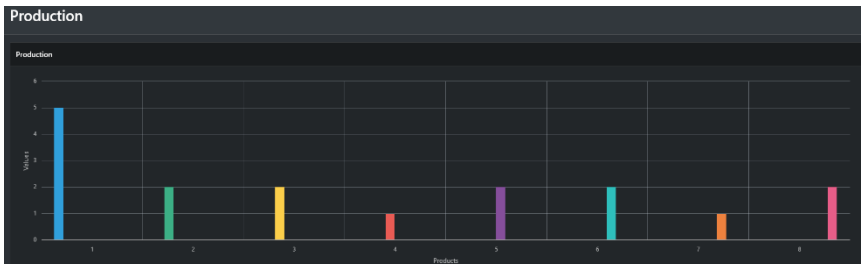

Go
Actions
Create
Customer id
Customer Name
Customer Dateofbirth
Customer Adress
Customer Phone
Customer Email
Gender
101 Abdul Hannan Ayubi 11/2/2021 Vilnius 666684552 Hannan@gmail.com M
120 Matilda Axelia 11/16/2021 Tallinn 8544545646 abd@abd.com U
126 Ayse 10/11/2021 Tallinn 4555288865 ayse@gmail.com F
134 Roose 6/28/2021 Riga 324112321311 Roose@gmail.com F
125 Lisa Denai 11/8/2021 Warsaw 78555424522 lisa@gmail.com F
129 Luna 5/10/2021 Tallinn 234234234 luna@gmail.com F
133 Adrian 6/21/2021 Tallinn 324112300 adrain@gmail.com M
118 Emila 10/4/2021 Kaunas 3705466852 Emila@gmail.com F
1 - 8


Customers
M
F
U


Customers City
Count
City
Kaunas
Riga
Tallinn
Vilnius
Warsaw
Riga
Warsaw
Kaunas
Vilnius
Tallinn


Products
Quantity
Product Type
Abra A5
Apple M5
Dell Monitor
Iphone 7
Iphone 8
Iphone X
IphoneSE
Samsung A51
Samsung A9 (2018)
Samsung S21
Xiomi X20


Customers
Customers \ Customers
Customer Name
Gender
Male
Female
Unkown
Customer Adress
Customer Phone
Customer Email
Cancel
Create


Production
Production
Values
Products

Stores

Go | Actions | Create

Store Id	Store Name	Store Adress	Store Phone	Store Email
105	Riga Technology Shop	Riga	5488785555	Riga@technologyshop.lv
115	Riga Technology_2 Shop	Riga	2131234422	Riga@tech.lt
101	Vilnius Tech_1 Store	Vilnius	370546531158	vilniustech@vilniustech.com
103	Kaunas Tech_191	Kaunas	6657780554	Kaunas@tech191.com
107	Tallinn Tech_2 Store	Tallinn	775856215566	tallinn@estonia.es
109	Warsaw Tech Shop	Warsaw	58452225485	Warsaw@tech.pl
113	Warsaw Tech_2 Shop	Warsaw	2211155588	Warsaw@tech2.pl
111	Vilnius Tech_2 Store	Vilnius	45885566	vilnius@tech2.com
117	Kaunas Tech_221	Kaunas	3705465349167	Kaunas@tech.lt

1 - 9

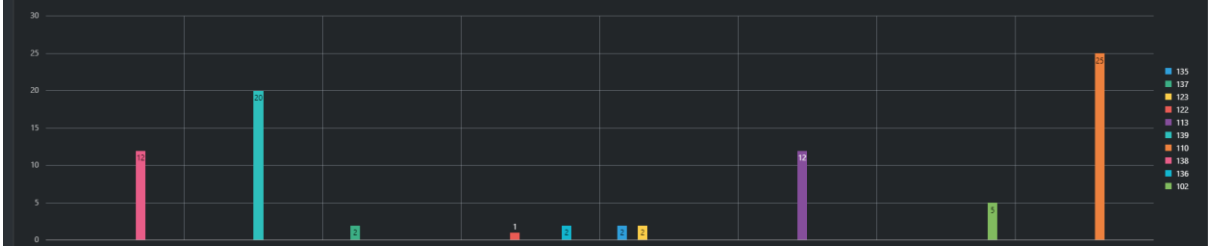
Employer List

Columns...

	Employee Name	Employee Dateofbirth	Employee Salary	Employement Date	Employee Adress	Employee Phone	Employee Email	Store ID
101	Ali	11/2/2021	1200	10/10/2021	Vilnius	370546665222	ali@gmail.com	101
113	Adrain	11/1/1995	1200	11/2/2021	Riga	456888225	Adrian@gmail.com	105
103	Emily Johansin	11/1/2021	1250	11/15/2021	Kaunas	666548222	Emily@hotmail.com	103
1006	Ali	2/2/2005	650	11/1/2021	Riga	45220548	Ali@gmail.com	105
2	Adrian	11/2/2020	1500	10/10/2021	Kaunas	35488888888	abd@abd.com	101
1004	Lisa	7/4/1996	1450	11/2/2021	Vilnius	4558872221	lisa@gmail.com	101

Orders

Orders Chart



Stocks

Go | Actions | Create

Store Name	Product Name	Quantity	Date	Stock Description	Stock Id
Kaunas Tech_191	Samsung S21	20	11/23/2021	This product has in this stores	1
Kaunas Tech_191	Xiomi X20	50	11/1/2021	These all products available.	21
Vilnius Tech_1 Store	IphoneSE	12	10/7/2021	These all available after specified date in specified Store	22
Kaunas Tech_191	Iphone 7	12	10/3/2021	These products are available in this shops.	23
Vilnius Tech_1 Store	Dell Monitor	45	10/6/2021	These products available in the stock.	41
Vilnius Tech_1 Store	Xiomi X20	25	11/23/2021	These product are available in this shop.	2
Warsaw Tech Shop	Dell Monitor	45	11/15/2021	These products are available.	61

1 - 7

Stock Information

Search: All Text Columns | Go | Actions | Edit | Add Row | Reset

Store ID	Product ID	Quantity	Date	Stock Description
103	2	20	11/23/2021	This product has in this stores
103	7	50	11/1/2021	These all products available.
101	1	12	10/7/2021	These all available after specified date in specified Store
103	3	12	10/3/2021	These products are available in this shops.
101	9	45	10/6/2021	These products available in the stock.
101	7	25	11/23/2021	These product are available in this shop.
109	9	45	11/15/2021	These products are available.

Total 7

Search: All Text Columns | Go | Actions | Edit | Add Row | Reset

Store Name	Store Adress	Store Phone	Store Email
Kaunas Tech_191	Kaunas	6657780554	Kaunas@tech191.com

Total 1

Search: All Text Columns | Go | Actions | Reset

Product Name	Product Price	Product Date	Product Description	Category Type
Samsung S21	550	11/8/2021	Display 6.20-inch (1080x2400) Processor Samsung Exynos 2100. F...	1

Create a Database Users

I create three users for my database system. The first use is the administrator who can do anything in the application like adding new employee, delete existing employee, or changing the employee information.

The next user is worker who can do some specific tasks which I assigned as a role for the workers. For example, worker can add new product orders customers etc. The worker can perform two roles which is creating a new employee or adding new stores. These are the roles of the administrator or the owner of the shop.

The last users only can see the specific things for example reader can see the orders month or the months which has the most orders. He can see some graphics he can see the products, stores, some information of the employees' orders which product ordered most, which producer and the category of the products. The end use only gets information but not adding or deleting or modifying the database.

Link:https://apex.oracle.com/pls/apex/database_management_course/r/database-management-final-project/login?session=8605249530816

Username	Password	Role
Admin	**admin12345**	**Administrator (Owner)**
Worker	**worker123**	**Worker (Employer)**
Reader	**reader123**	**End User**

5. SUMMARY OF WORK RESULT

In this project I implemented a simple stores database which has three main roles administrator, worker, and end – user. The main goal of implementing this project to make some easiness for the above user. The main user who is administrator can easily deal with his / her shops information, calculate the total revenue of their stores, make easily add or delete a new employee calculate the total employee salary etc. The second main role is for workers of stores who can easily add customer and sell some products to them. Record customer information and analyses the information. Example, Which product has sold most and in which city and shop and how many customers do they have and sort it with their genders etc. The last idea of the creating this project is for the end users. The end user can easily understand which month have most orders in which product, by this he / she can decide to the get new item at those dates. He can see the product and producers which producer have what type of product category and how much these shops have sold the product etc. by this information the end use can easily decide to buy what kind of the product he can get and what is the price of the products. By making this project I understand queries like select, insert, create, count, where, the conditions (=,< , >) etc. This project makes me understand the relations between the tables and the columns.

6. DEFENCE OF COURSE WORK

This the link of my whole project which you can access from here and download every necessary information from here.

https://drive.google.com/drive/folders/1aI_Tz-6I1yzlwpgkBoTvLWn_qxDfMedC?usp=sharing

7. LITERATURE

- **https://www.w3schools.com/sql/**
- https://thumbs.dreamstime.com/b/elektromarkt-consumer-electronics-store-vilnius-lithuania-september-akropolis-hyper-market-discounts-bonuses-offered-80011166.jpg (Pic – 1)
- https://thumbs.dreamstime.com/b/elektromarkt-electronics-store-lithuania-vilnius-september-consumer-akropolis-hyper-market-discounts-bonuses-offered-80011114.jpg (pic - 2)
- https://www.youtube.com/watch?v=ahlW-Jkumto&list=PL_c9BZzLwBRJXk2hEP-U_OUMJAf_TYEiZ&ab_channel=CalebCurry
- https://www.youtube.com/watch?v=Z5Fwx8LygW0&list=PL1MJdy9N8XJLLs-4IGpEPet-7MJf4HUic&ab_channel=JKreieDatabase