**Nama : Agung Wicaksono**

**Kode Peserta : FSDO002ONL012**

**Tugas : Assignment 2**

1. Customers => stores data customer

Query :

create database db\_bank;

use db\_bank;

create table tb\_customers(

customerNumber int primary key identity(1,1),

customerName varchar(50) not null,

contactFirstName varchar(50) not null,

contactLastName varchar(50) not null,

phone varchar(15) not null,

addressLine1 varchar(200) not null,

addressLine2 varchar(200) not null,

city varchar(30) not null,

state varchar(30) not null,

postalCode varchar(10) not null,

country varchar(20) not null,

salesRepEmployeeNumber int not null,

creditLimit int not null,

);

insert into tb\_customers(customerName, contactFirstName, contactLastName, phone, addressLine1, addressLine2, city, state, postalCode, country, salesRepEmployeeNumber, creditLimit)

values('Agung', 'Agung', 'Wicaksono', '088812345678', 'Jl. Keputih', 'Jl. Mulyorejo', 'Surabaya', 'Jawa Timur', '60111', 'Indonesia', 1, 1000000),

('Faishal', 'Achmad', 'Rizal', '088812345678', 'Jl. Keputih', 'Jl. Mulyorejo', 'Sidoarjo', 'Jawa Timur', '61111', 'Indonesia', 2, 5000000),

('Zakiy', 'Achmad', 'Mulyanto', '088812345678', 'Jl. Keputih', 'Jl. Mulyorejo', 'Gresik', 'Jawa Timur', '62111', 'Indonesia', 1, 8000000),

('Syiham', 'Achmad', 'Akbar', '088812345678', 'Jl. Keputih', 'Jl. Mulyorejo', 'Surabaya', 'Jawa Timur', '60111', 'Indonesia', 1, 5000000),

('Hafizh', 'Hafizh', 'Fauzan', '088812345678', 'Jl. Sukajaya', 'Jl. Majumapan', 'Bandung', 'Jawa Barat', '40111', 'Indonesia', 4, 10000000),

('Grace', 'Elva', 'Alverina', '088812345678', 'Jl. Sukajaya', 'Jl. Majumapan', 'Bogor', 'Jawa Barat', '41111', 'Indonesia', 3, 10000000),

('Fikar', 'Muhammad', 'Fikar', '088812345678', 'Jl. Sukajaya', 'Jl. Majumapan', 'Cilegon', 'Banten', '42111', 'Indonesia', 7, 1000000),

('Fradipta', 'Fradipta', 'Alqoyum', '088812345678', 'Jl. Sukasuka', 'Jl. Kepanjangan', 'Bukittinggi', 'Sumatera Barat', '20111', 'Indonesia', 5, 10000000),

('Mpu', 'Hambyah', 'Bagaskara', '088812345678', 'Jl. Keputih', 'Jl. Mulyorejo', 'Klaten', 'Jawa Tengah', '50111', 'Indonesia', 6, 2500000),

('Pujik', 'Syaidatul', 'Fauziah', '088812345678', 'Jl. Keputih', 'Jl. Mulyorejo', 'Bukittinggi', 'Sumatera Barat', '20111', 'Indonesia', 4, 7000000),

('Affan', 'Ahsanul', 'Habib', '088812345678', 'Jl. Keputih', 'Jl. Mulyorejo', 'Madiun', 'Jawa Timur', '63173', 'Indonesia', 2, 10000000);

select \* from tb\_customers;

A screenshot of a computer

Description automatically generated

1. Products => stores daftar/list model product ( dalam hal ini bisa mobil/motor dll )

Query:

create table tb\_product(

productCode int primary key identity(1,1),

productName varchar(50) not null,

productLine int not null,

productScale int not null,

productVendor varchar(50) not null,

productDescription text not null,

quantityStock int not null,

buyPrice int not null,

MSRP int not null,

foreign key (productLine) references tb\_productlines(productLine)

);

insert into tb\_product(productName, productLine, productScale, productVendor, productDescription, quantityStock, buyPrice, MSRP)

values('tabungan siswa', 2, 30, 'bank1', 'tabungan untuk siswa', 1000, 50000, 40000),

('tabungan mahasiswa', 2, 10, 'bank2', 'tabungan untuk mahasiswa', 500, 100000, 90000),

('tabungan orang tua', 5, 30, 'bank10', 'tabungan untuk persiapan pensiun', 10000, 1000000, 500000),

('kredit hp', 9, 70, 'hp2', 'cicilan membeli hp', 10000, 1000000, 1000000),

('kredit rumah', 7, 5, 'perumahan1', 'cicilan perumahan', 1000, 100000000, 9000000),

('kredit motor', 8, 7, 'motor1', 'cicilan membeli motor', 100, 15000000, 16000000),

('kredit mobil', 8, 2, 'mobil1', 'cicialan untuk membeli mobil', 10, 100000000, 200000000),

('sms banking', 10, 20, 'bank11', 'layanan bank melalui sms', 100, 500000, 400000),

('pinjaman online', 6, 10, 'bank100', 'pinjaman online', 10, 500000, 400000),

('dompet digital', 5, 15, 'ewallet1', 'uang digital', 1000, 10000, 5000);

select \* from tb\_product;

A screenshot of a computer

Description automatically generated

1. Productlines => stores daftar/list kategori product

Query:

create table tb\_productlines(

productLine int primary key identity(1,1),

textDescription text not null,

htmlDescription text not null,

image varchar(200) not null

);

insert into tb\_productlines(textDescription, htmlDescription, image)

values('deposito', 'deposito.html', 'deposito.jpg'),

('tabunganmuda', 'tabunganmuda.html', 'tabunganmuda.jpg'),

('tabunganpekerja', 'tabunganpekerja.html', 'tabunganpekerja.jpg'),

('tabunganmenikah', 'tabunganmenikah.html', 'tabunganmenikah.jpg'),

('tabunganpensiun', 'tabunganpensiun.html', 'tabunganpensiun.jpg'),

('investasisaham', 'investasisaham.html', 'investasisaham.jpg'),

('kprrumah', 'kprrumah.html', 'kprrumah.jpg'),

('pinjaman', 'pinjaman.html', 'pinjaman.jpg'),

('asuransi', 'asuransi.html', 'asuransi.jpg'),

('ebanking', 'ebanking.html', 'ebanking.jpg'),

('mbanking', 'mbanking.html', 'mbanking.jpg');

select \* from tb\_productlines;

Graphical user interface, text, application

Description automatically generated

1. Orders => store Order Sales oleh customer

Query:

create table tb\_orders(

orderNumber int primary key identity(1,1),

orderDate date not null,

requiredDate date not null,

shippedDate date not null,

status varchar(50) not null,

comments text not null,

customerNumber int not null,

foreign key (customerNumber) references tb\_customers(customerNumber)

);

insert into tb\_orders(orderDate, requiredDate, shippedDate, status, comments, customerNumber)

values('2021-09-10', '2021-09-11', '2021-09-12', 'Valid', 'no comment', 1),

('2021-09-13', '2021-09-14', '2021-09-15', 'Diproses', 'no comment', 2),

('2021-09-10', '2021-09-11', '2021-09-12', 'Dikirim', 'no comment', 2),

('2021-09-10', '2021-09-11', '2021-09-12', 'Diterima', 'no comment', 10),

('2021-09-10', '2021-09-11', '2021-09-12', 'Selesai', 'no comment', 6),

('2021-09-10', '2021-09-11', '2021-09-12', 'Valid', 'no comment',5),

('2021-09-10', '2021-09-11', '2021-09-12', 'Selesai', 'no comment', 7),

('2021-09-10', '2021-09-11', '2021-09-12', 'Valid', 'no comment', 3),

('2021-09-10', '2021-09-11', '2021-09-12', 'Diproses', 'no comment', 3),

('2021-09-10', '2021-09-11', '2021-09-12', 'Diterima', 'no comment', 8);

select \* from tb\_orders;

Graphical user interface, application, table

Description automatically generated

1. OrderDetails => store Item Order sales dalam setiap order sales

Query:

create table tb\_orderdetails(

orderNumber int not null foreign key references tb\_orders(orderNumber),

productCode int not null foreign key references tb\_product(productCode),

constraint pk\_product\_orders primary key (orderNumber, productCode),

quantityOrdered int not null,

priceEach int not null,

orderLineNumber int not null,

);

insert into tb\_orderdetails(orderNumber, productCode, quantityOrdered, priceEach, orderLineNumber)

values(1, 2, 5, 20000, 1),

(1, 9, 1, 50000, 1),

(1, 3, 2, 1000, 2),

(1, 6, 1, 10000, 3),

(2, 1, 1, 200000, 5),

(5, 2, 5, 1000, 1),

(5, 7, 1, 50000, 1),

(8, 2, 2, 20000, 1),

(4, 1, 2, 1000, 5),

(3, 8, 1, 200000, 1),

(9, 5, 10, 5000, 1);

select \* from tb\_orderdetails;

Graphical user interface, application

Description automatically generated

1. Payments => store Pembayaran oleh customer sesuai dengan akun pembayaran

Query:

create table tb\_payments (

customerNumber int not null,

checkNumber int primary key identity(1,1),

paymentDate date not null,

amount int not null,

foreign key (customerNumber) references tb\_customers(customerNumber)

);

insert into tb\_payments(customerNumber, paymentDate, amount)

values(1, '2021-09-10', 100000),

(2, '2021-09-10', 200000),

(5, '2021-09-10', 2000000),

(1, '2021-09-10', 20000),

(3, '2021-09-10', 10000),

(2, '2021-09-10', 200000),

(9, '2021-09-10', 350000),

(3, '2021-09-10', 20000),

(1, '2021-09-10', 200000),

(8, '2021-09-10', 970000);

select \* from tb\_payments;

Graphical user interface, text, application

Description automatically generated

1. Employee => store informasi karyawan dalam sebuah organisasi struktur

Query:

create table tb\_employees (

employeeNumber int primary key identity(1,1),

lastName varchar(50) not null,

firstName varchar(50) not null,

extension varchar(10) not null,

email varchar(50) not null,

officeCode int not null,

reportsTo int not null,

jobTitle varchar(50) not null,

foreign key (officeCode) references tb\_offices(officeCode),

foreign key (reportsTo) references tb\_employees(employeeNumber)

);

insert into tb\_employees(lastName, firstName, extension, email, officeCode, reportsTo, jobTitle)

values('Wicaksono', 'Agung', 'S.T.', 'email@gmail.com', 1, 1, 'Head'),

('Habib', 'Affan', 'S.T.', 'email@gmail.com', 1, 1, 'Asistence'),

('Ahsan', 'Muhammad', 'S.Kom.', 'email@gmail.com', 2, 1, 'Asistence'),

('Prasetyo', 'Yogi', 'Amd.', 'email@gmail.com', 2, 1, 'Asistence'),

('Erucokro', 'Arimurti', 'S.Pet.', 'email@gmail.com', 3, 1, 'Asistence'),

('Haryanto', 'Pebri', 'S.T.', 'email@gmail.com', 5, 1, 'Asistence'),

('Arifandi', 'Urvan', 'S.T.', 'email@gmail.com', 1, 1, 'Asistence'),

('Adwitian', 'Emeraldin', 'S.T.', 'email@gmail.com', 7, 2, 'Asistence'),

('Deny', 'Achmad', 'S.T.', 'email@gmail.com', 2, 3, 'Asistence'),

('Herlambang', 'Dane', 'Amd.', 'email@gmail.com', 5, 5, 'Asistence');

select \* From tb\_employees;

alter table tb\_customers ADD foreign key (salesRepEmployeeNumber) references tb\_employees(employeeNumber);

Graphical user interface, table

Description automatically generated

1. Offices => store data sales office

Query:

create table tb\_offices(

officeCode int primary key identity(1,1),

city varchar(50) not null,

phone varchar(15) not null,

addressLine1 varchar(200) not null,

addressLine2 varchar(200) not null,

state varchar(50) not null,

country varchar(50) not null,

postalCode varchar(10) not null,

territory varchar(50) not null,

);

insert into tb\_offices(city, phone, addressLine1, addressLine2, state, country, postalCode, territory)

values('Jakarta Pusat', '088812345678', 'Jl. Soekarno-Hatta', 'Jl. Pahlawan', 'DKI Jakarta', 'Indonesia', '10110', 'Indonesia Barat'),

('Jakarta Selatan', '088812345678', 'Jl. Soekarno-Hatta', 'Jl. Pahlawan', 'DKI Jakarta', 'Indonesia', '10111', 'Indonesia Barat'),

('Jakarta Timur', '088812345678', 'Jl. Soekarno-Hatta', 'Jl. Pahlawan', 'DKI Jakarta', 'Indonesia', '10112', 'Indonesia Barat'),

('Jakarta Utara', '088812345678', 'Jl. Soekarno-Hatta', 'Jl. Pahlawan', 'DKI Jakarta', 'Indonesia', '10113', 'Indonesia Barat'),

('Jakarta Barat', '088812345678', 'Jl. Soekarno-Hatta', 'Jl. Pahlawan', 'DKI Jakarta', 'Indonesia', '10114', 'Indonesia Barat'),

('Bandung', '088812345678', 'Jl. Soekarno-Hatta', 'Jl. Pahlawan', 'Jawa Barat', 'Indonesia', '20111', 'Indonesia Barat'),

('Bekasi', '088812345678', 'Jl. Soekarno-Hatta', 'Jl. Pahlawan', 'Jawa Barat', 'Indonesia', '20112', 'Indonesia Barat'),

('Surabaya', '088812345678', 'Jl. Soekarno-Hatta', 'Jl. Pahlawan', 'Jawa Timur', 'Indonesia', '60111', 'Indonesia Timur'),

('Makassar', '088812345678', 'Jl. Soekarno-Hatta', 'Jl. Pahlawan', 'Sulawesi Selatan', 'Indonesia', '110111', 'Indonesia Timur');

select \* from tb\_offices;

A screenshot of a computer

Description automatically generated

Berikut adalah hasil ERD Diagram dari database yang sudah dibuat :

Diagram

Description automatically generated