ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – ВАРНА

ФАКУЛТЕТ ПО ИЗЧИСЛИТЕЛНА ТЕХНИКА И АВТОМАТИЗАЦИЯ

Катедра „Софтуерно Инженерство “

A blue and white logo

Description automatically generated with medium confidence

**Курсова Работа по**

**УИС**

Управленски информационни системи

**Тема:**

Информационна система, подпомагаща управлението на музикален магазин.

**Разработил:**  **Проверил:**

Ивайло Пламенов Руменов  гл. ас. д-р Диян Динев

Фак. № 23651227

Съдържание

[Задание 2](#_Toc152195068)

[Увод 3](#_Toc152195069)

[Проектиране на спрямо бизнес процеси на заданието 3](#_Toc152195070)

[Архитектура на информационната система 4](#_Toc152195071)

[Проектиране на базата от данни 5](#_Toc152195072)

[Програмен код 10](#_Toc152195073)

[Потребителски интерфейс 10](#_Toc152195074)

[Заключение 11](#_Toc152195075)

[Положение 11](#_Toc152195076)

# Задание

Управленски информационни системи -Задание 12

Да се създаде информационна система, подпомагаща управлението на музикален магазин. БД да съхранява информация за стоки, клиенти, служители и продажби. В таблиците да се включат атрибути като номер на артикул, вид, година, наименувания, изпълнител, жанр, музикална компания, единична цена, наличност, номер на продажба, дата на продажбата, стоки, брой, име на клиент, адрес, телефон, име на служител, позиция, телефон, отдел и др. Базата от данни трябва да е нормализирана. При създаване на таблиците изберете подходящ тип данни и други свойства на полетата. Да се валидират данните. Ограничете броя и наличността да са само положителни числа, а датата на продажбата да е днешна. Наименуванията на вида, изпълнителя, жанра и музикалната компания да се избират от списъци. Изберете правилни ключови полета. Свържете таблиците с подходящи релации. В основните таблици да има въведени минимум 10 записа коректни данни. Системата да генерира справки на база информация съхранена в повече от една таблица. Справките да са минимум 5. Например: Да се генерира справка, която да показва последните 5 продажби на стоки издадени последната година. Създадената информационна система да позволява въвеждане, корекция, актуализация и търсене на данни. Направените справки да могат да се експортират в подходящ файлов формат за отпечатване.

# Увод

В курсовата работа е разгледано как се реализира информационна система подпомагаща музикален магазин. За реализацията на информационната система е разгледан как се декомпозира идеята за приложението на множество стъпки, като и реализацията на всяка стъпка. Започвайки от анализиране на нужната бизнес логика и тя как повлиява на използваните архитектури и програмни технологии за имплементация. Преминавайки към дълбоко разглеждане как е разработена базата от данни и нейните градивни елементи. Проекта включва и разглеждане на реализацията на бизнес логиката като програмен код. Както и как е тестван и застрахован от провал вътрешната логика на проекта. Проекта включва и разглеждане на как е реализиран потребителския интерфейс, както и как той комуникира с бизнес логика. И финализиране с оценка на трудностите, грешките и подобренията към проекта в негова цялост.

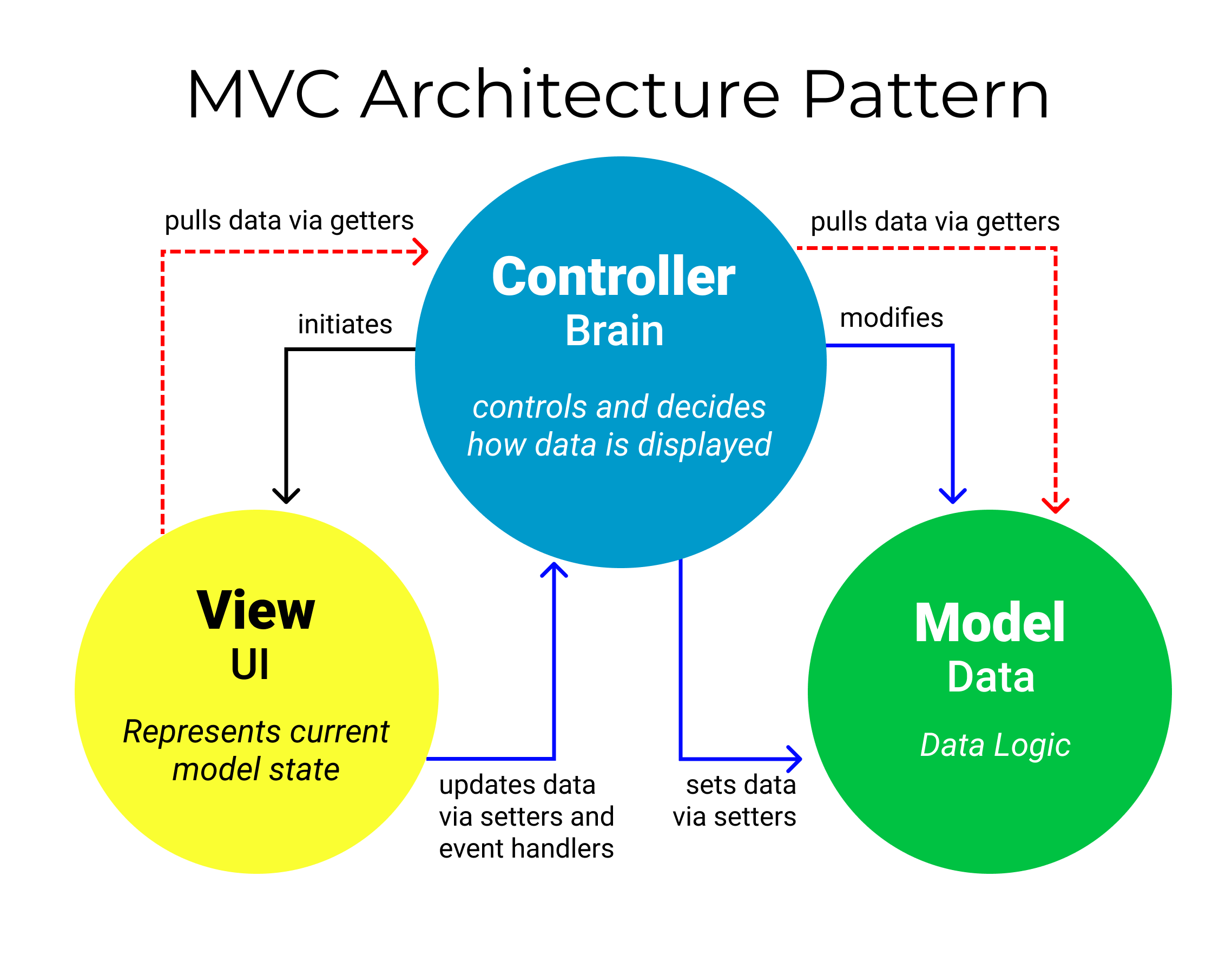
# Проектиране на спрямо бизнес процеси на заданието

При започване на задание трябва да се разчертае и цялостната структура спрямо бизнес логиката на информационната система. В този случай е предвидено информационната система да се използва като спомагателно средство на музикален магазин. Тоест цялостната цел на заданието е с комерсиална цел. Знаейки това се примамва към факта че съдържанието и менажирането на стоки и продукти ще е главната цел на информационната система. Както и при комерсиалните приложения за пазаруване се предполага че клиенти ще желаят множеството от артиколи да са сортирани и те самите при желание да сортират и търсят спрямо дадени критерий. Както при пазарува ще е нужно да се поддържа активна дадената количка на клиента. Тоест клиента да има свободата да добавя и премахва артикули в количката преди закупуване. И при закупуване на артикули информационната системата трябва да изведе дадените артикули от складова база с данни и да промени количество и да запише поръчката на клиента.

# Архитектура на информационната система

A diagram of a diagram

Description automatically generatedСлед обзора на бизнес логиката можем се стига до факта че информационната система ще бъде от вид Модел Вю Контролер (Фиг. №1). Този модел е често срещан при програмни продукти от разновиден вид. Той представлява групирани на програмния ко по неговата роля. Така се схематизира всяка единица да е отговорна за едно работа, което също подкрепя един от законите на обектното ориентирано програмиране. При Модел Вю Контролер се изготвят три елемента независими едни от друг. Модела отговорен за бизнес логиката както и връзка с базата от данни. Вю-то което е отговорно за потребителския интерфейс на информационната система. Контролера е последната част от модела и отговаря за комуникационния слой помежду двата елемента. За целта на изпълнението на задачата Контролера е реализиран чрез REST метода на работа и json стандарта за предаване на данни помежду програмните слоеве. REST e метод на реализиране на HTTP заявки помежду два така наречени крайни връзки ( “Endpoints”). Често използван е метода REST за комуникация и стандартизиран като доверчив и бърз.



Фиг. №1 Графика ма архитектурата Модел Вю Контролер

# Проектиране на базата от данни

След приемане на метод на работа се преминава към едни от базовите елементи на модела Модел Вю Контролер и това е базата от данни. За целта на разработване на информационна система за музикален магазин, както и вземайки предвид предишен опит с работата по и с бази от данни, е избрано да се използва PosgreSql база от данни. Тя използва налог на sql наречен psql. Базата е разгъната локално на машината, на която се разработва проекта. Съществуват облачно базирани услуги за поддържане на база от данни, като neon, digitaOcian, cauchebase и много други. Удобството на облачно разгъната база от данни е че множество души могат да работят едновременно по базата но и съществуват проблеми при заключване на достъп до полета и таблици. Но за целта на задачата е предприето базата от данни да е разгъната на локалната машина. Поради липсата от нуждата да се споделя данни, като при екипи от разработчици. Но при имплементация на реална система е силно препоръчано тя да е разгъната на специализиран сървър за менажиране единствено и само за база от данни. Съществуват както и специализирани операционно системи само за менажиране и поддръжка на бази от данни, техните инстанции и мрежовата комуникация нужна за работа с базата.

След като е избрана базата за работа и тя къде ще бъде разгъната може да се премине към писане на разработка на базата от данни и това може да се сведе до няколко ключови стъпки:

* Определете целта на база данни. Това ще помогне да се подготвите за останалите стъпки.
* Намиране и организиране на необходимата информация. Събиране на всички типове информация, която може да запишете в базата данни, като например име на продукт и номер на поръчка.
* Разделете информацията в таблици. Разделяне на информацията на елементи на основни обекти или теми, като например продукти или поръчки. След това всеки предмет се превръща в таблица.
* Превръщане на информационните елементи в колони. Решава се каква информация ще се съхранявате във всяка таблица. Всеки елемент става поле и се показва като колона в таблицата. Например таблицата "Служители" може да включва полета като "Фамилно име" и "Дата на наемане".
* Задаване на първични ключове. Избиране първичния ключ на всяка таблица. Първичният ключ е колона, която се използва за еднозначно идентифициране на всеки ред. Пример за това може да бъде ИД на продукт или ИД на поръчка.
* Настройване на релациите между таблиците. Поглежда се всяка таблица и решете как данните в една таблица са свързани с данните в други таблици. Добавете полета към таблици или създайте нови таблици, за да изясните релациите, ако е необходимо.
* Прилагане на правилата за нормализиране. Прилага се правилата за нормализиране на данните, за да видите дали вашите таблици са структурирани правилно. Направете корекции в таблиците, ако е необходимо.

Това е само един от вариантите за разработка на база от данни. Но е преценено че за конкретната задача ще е нужен този метод на разработка на база от данни.

Първо е нужно да се определи целта на базата от данни. За проекта тази цел е се състои от това да се съхранява информация за дадените артикули за продан, да се съдържа информация за клиента и неговата количка, да се съхранява потребителска информация за клиента както и да се съхранява информация за работещия персонал. Тези елементи от базата са ключови за нейното опериране. За визуализиране на тези изисквания е представена чрез графика(Фиг. №2)

A diagram of a flowchart

Description automatically generated

Фиг. №2 Графика представляваща чернова на базата от данни

След като е представяне черновата и е определено целта на база от данни. Се преминава към отделяне на откритата информация в таблици. След гарирането на нужните елементи и атрибути, в базата от данни, се влиза в конкретиката на дадените атрибути. Осмисля се какви полета са нужни за таблицата и как те ще са обвързани с полета от други таблици. Относно условията на проекта, които гласят че базата от данни ще се употребявал като спомагателно средство на музикален магазин, се извежда факта че трябва да се инстанции устойчива система за следене състоянието на артикулите за продан. Тази функционалност е ядрото на всеки комерсиална информационна система. За това е решено да има таблица “Item” (Фиг. № 3), която е състоене от всичко сходни полета нужни на един артиколи да е годен за продан. От тази таблица се извежда две таблици едната за музикални инструменти „Instrument“ и за грамофонни плочи „Record“. Така системата може да се разпределя класовете на продукти в различни таблици с тяхното наименование, и когато бизнеса си реши да си разшири предметната дейност то може само да дови нова таблица с релация към таблицата с артиколи „Item“. Това от друга страна е недостатък към системата. Тъй като за всяко ново разделение ще е нужна нова таблица и може да се стигне до момент където множество таблици имат релация към таблицата с артиколи. Водейки към голяма опашка при заключване и отключване на достъпа към ресурси.

A screenshot of a computer

Description automatically generated

Фиг. № 3 Таблицата за Артиколи „Item“

Но за целите на проекта е преценено е че най подходящия вариант за предметната дейност на магазина, която зависи от голяма разновидност при един вид продукти. На пример таблицата за инструменти „Instruments“ (Фиг. № 4), сама дефиниция на какво е инструмент варира от духово до струене. И в тази вариация се специализира допълнително, на пример от какъв материал е направен инструмента, от кой е настроен, дали е електронно съвместен и много други. Точно тази дълбока квалификация на инструментите е причината за разделяне на абстрактния артиколи, като нещо което се продава, и инструмента, като нещо което се използва за свирене. Така лесно се обединяват двете абстракции чрез релация и се получава инструмент за продаване. При това разделение на две абстрактни елемента лесно се добавя и премахнат един от друг. Затова е и възможно да се довят и други абстракции като таблицата за грамофонни плочи „Record“. Както и лесно може да се добавят допълнителни таблици към артикулите. Като е направено за производителя на артиколи с таблицата „Company“. Таблицата „Company“ сама по себе си представлява компания но тази компания в информационното приложение може да се използва като таблица на производител или друг вид предназначение.

A screenshot of a computer

Description automatically generated

Фиг. № 4 Таблицата за инструменти „Instrument“

Следващата ключова част е самата продажба на артикулите. След като сме запазили правилно артикулите и знаем е са устойчиви може да се помисли как те могат да се продава (Фиг. №5). За тази цел е реализирана така наречената количка за пазаруване, която запазва в себе си релация към артикулите желани от купувача и таблица събирайки информацията за тази неизпълнена покупка. И когато клиента реши да си изпълни поръчка се записва като изпълнена и на клиента се създава нова количка за пазаруване.

A screenshot of a computer

Description automatically generated

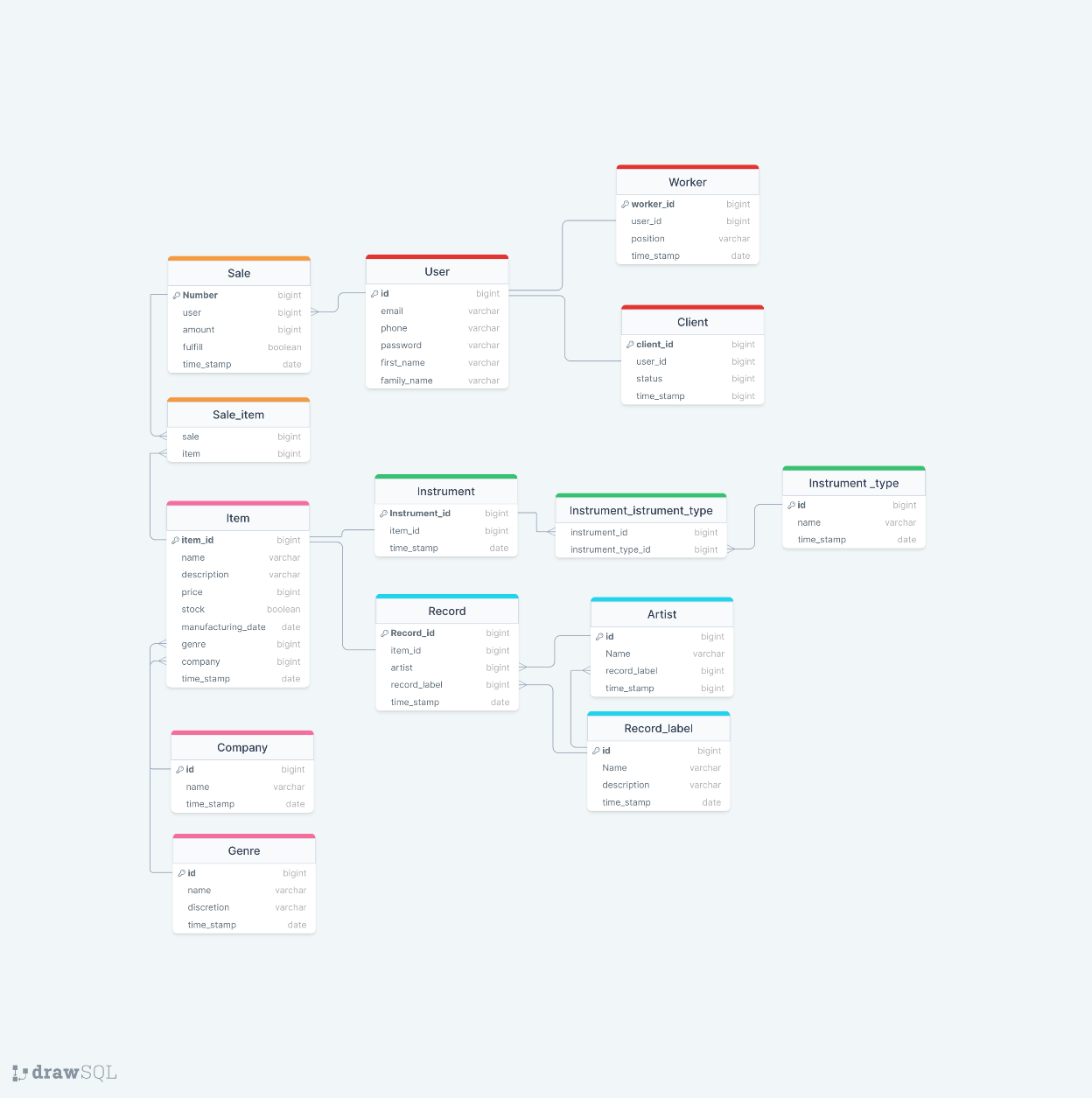
Фиг. №5 Таблиците за кошница за пазара „Sale\_item“ и таблицата за продажба „Sale“

Но за да има клиента количка той първо се нуждае от профил, в който ще се отбележи колко поръчки един клиент има и от какво са се състои дадената поръчка. Когато се имплементира модел за профил на потребите то лесно може да се добави профили за работници( Фиг. №6). Това е възможно когато се декомпозира какво е потребител и се раздели на купувач и работник. Те в същности си имат еднакви елементи.

A screenshot of a computer

Description automatically generated

Фиг. №6 Таблицата за потребител на системата „User“, таблицата за клиента на системата „Client“ и таблицата за работни в системата “Worker”



Фиг. №3 Графика на реализираните таблици в базата от данни и техните релации.

# Програмен код

След като е инициализиране базата на система може да се премини към вливането на програмния код. Тоест да се реализира бизнес процеси чрез вътрешни алгоритми на системата. За реализирането на това е избрано програмния език java, избран защото е отвратен на пазара, представлява устойчивост чрез силно типизиран обектно ориентиран модел на работа и лесен за имплементиране като модел. Но сам по себе си java е слаб затова е имплементиран чрез библиотечната средата Spring. Който позволява улеснен метод за работа с REST и бази от данни едновременно. Той предоставя и вграден начин на тестване на цялостната система от край до край.

Когато имаме бизнес процеси за имплементиране и модела на работа се започва от горе надуло. Тоест се запазва от външните точки (endpoints) на системата. Тези външни точки представляват http uri, който крие зад себе си логиката имплементиране външни. Затова на този крий и само на този слой се правят проверки за коректни данни на входа. След което се преминава към сервизната част на програмата, която представлява алгоритмичната част на програмата. Тук се реализира по някаква форма бизнес логиката на приложение съчета с вътрешно програмна. Вътрешно програмната реализации използва методи на езика за предотвратяване на повторение или други проблеми свързани с обектното ориентирано програмиране. Този слой може множествено пъти да вика набор от методи отговорни за достъп до базата от данни. Тези методи са в така наречен репозитори “repository” и в някой имплементации на модела се използва друг метод за достъп до базата наречен персистентност „persistence“. На за даденото задание е прието да се използват директни заявки към базата нарече queries. Този трислоен подход позволява за висока устойчивост на предложението.

# Потребителски интерфейс

Потребителския интерфейс е реализиран чрез уеб базиран метод. Той се гради на javascript езика отговарящ за логика на интерфейса и htlp съчетан със css. Това е често срещата комбинация за имплементиране на потребителски интерфейс в уеб пространството. Цели този пакет е имплементиран чрез скелета за имплементация наречен angular. Angular е скелет осланяващ процеса на работа в уеб пространството. Той е разработен от Google и е от ней-употребяваните на пазара.

# Заключение

Заданието за представляващ спомагателна информационна система за музикален магазин бе предизвикателна и широко разширяваща множествено знания за продукти тяхната логика и как работят едно с друго. Базата от данни използвана е PosgreSql достъпна чрез String rest application и Angular потребителски интерфейс възпроизведен в уеб браузър. За бъдеще може да се помисли и за добавяне допълнителни абстракции към проекта. Като например опцията за вземане продукт на изплащане. Това лесно може да се постигне тъй като имаме таблица за артикули „Item“, към която може да се добави таблица за период на изплащане който е вързан и към клиента. Друга иновация за процеса на работа, който е лесно за разработване благодарение на модела на имплементация на базата, е отбелязването на служител който да проследява поръчката на клиента. Така всеки клиент ще знае към кой служител може да се обърне за проблеми с поръчката или проблеми с продукта.

# Положение

Приложени скриптове за базата от данни

За създаване на базата е използван скрипта

Create.sql

CREATE TABLE Ape\_user(

                       id BIGINT NOT NULL,

                       email VARCHAR(255) NOT NULL,

                       phone VARCHAR(255) NOT NULL,

                       password VARCHAR(255) NOT NULL,

                       first\_name VARCHAR(255) NOT NULL,

                       family\_name VARCHAR(255) NOT NULL

);

ALTER TABLE

    Ape\_user ADD PRIMARY KEY(id);

CREATE TABLE Instrument(

                             Instrument\_id BIGINT NOT NULL,

                             item\_id BIGINT NOT NULL,

                             type BIGINT NOT NULL,

                             time\_stamp DATE NOT NULL

);

ALTER TABLE

    Instrument ADD PRIMARY KEY(Instrument\_id);

CREATE TABLE Item(

                       item\_id BIGINT NOT NULL,

                       stock BOOLEAN NOT NULL,

                       price BIGINT NOT NULL,

                       discrimination VARCHAR(255) NULL,

                       manufacturing\_date DATE NOT NULL,

                       name VARCHAR(255) NOT NULL,

                       description VARCHAR(255) NOT NULL,

                       genre BIGINT NOT NULL,

                       company BIGINT NOT NULL,

                       time\_stamp DATE NOT NULL

);

ALTER TABLE

    Item ADD PRIMARY KEY(item\_id);

CREATE TABLE Sale\_item(

                            sale BIGINT NOT NULL,

                            item BIGINT NOT NULL

);

CREATE TABLE Instrument\_istrument\_type(

                                            instrument\_id BIGINT NOT NULL,

                                            instrument\_type\_id BIGINT NOT NULL

);

CREATE TABLE Sale(

                       Number BIGINT NOT NULL,

                       ape\_user BIGINT NOT NULL,

                       amount BIGINT NOT NULL,

                       fulfill BOOLEAN NOT NULL,

                       time\_stamp DATE NOT NULL

);

ALTER TABLE

    Sale ADD PRIMARY KEY(Number);

CREATE TABLE Genre(

                        id BIGINT NOT NULL,

                        name VARCHAR(255) NOT NULL,

                        discretion VARCHAR(255) NOT NULL,

                        time\_stamp DATE NOT NULL

);

ALTER TABLE

    Genre ADD PRIMARY KEY(id);

CREATE TABLE Company(

                          id BIGINT NOT NULL,

                          name VARCHAR(255) NOT NULL,

                          time\_stamp DATE NOT NULL

);

ALTER TABLE

    Company ADD PRIMARY KEY(id);

CREATE TABLE Artist(

                         id BIGINT NOT NULL,

                         Name VARCHAR(255) NOT NULL,

                         record\_label BIGINT NOT NULL,

                         time\_stamp DATE NOT NULL

);

ALTER TABLE

    Artist ADD PRIMARY KEY(id);

CREATE TABLE Record(

                         Record\_id BIGINT NOT NULL,

                         item\_id BIGINT NOT NULL,

                         artist BIGINT NOT NULL,

                         record\_label BIGINT NOT NULL,

                         time\_stamp DATE NOT NULL

);

ALTER TABLE

    Record ADD PRIMARY KEY(Record\_id);

CREATE TABLE Record\_label(

                               id BIGINT NOT NULL,

                               Name VARCHAR(255) NOT NULL,

                               description VARCHAR(255) NOT NULL,

                               time\_stamp DATE NOT NULL

);

ALTER TABLE

    Record\_label ADD PRIMARY KEY(id);

CREATE TABLE Worker(

                         worker\_id BIGINT NOT NULL,

                         ape\_user BIGINT NOT NULL,

                         position VARCHAR(255) NOT NULL,

                         time\_stamp DATE NOT NULL

);

ALTER TABLE

    Worker ADD PRIMARY KEY(worker\_id);

CREATE TABLE Client(

                         client\_id BIGINT NOT NULL,

                         ape\_user BIGINT NOT NULL,

                         status BIGINT NOT NULL,

                         time\_stamp BIGINT NOT NULL

);

ALTER TABLE

    Client ADD PRIMARY KEY(client\_id);

CREATE TABLE Instrument\_type(

                                   id BIGINT NOT NULL,

                                   instrument\_type\_name VARCHAR(255) NOT NULL,

                                   time\_stamp DATE NOT NULL

);

ALTER TABLE

    Instrument\_type ADD PRIMARY KEY(id);

ALTER TABLE

    Instrument\_istrument\_type ADD CONSTRAINT instrument\_istrument\_type\_instrument\_type\_id\_foreign FOREIGN KEY(instrument\_type\_id) REFERENCES Instrument\_type(id);

ALTER TABLE

    Item ADD CONSTRAINT item\_company\_foreign FOREIGN KEY(company) REFERENCES Company(id);

ALTER TABLE

    Record ADD CONSTRAINT record\_item\_id\_foreign FOREIGN KEY(item\_id) REFERENCES Item(item\_id);

ALTER TABLE

    Sale\_item ADD CONSTRAINT sale\_item\_sale\_foreign FOREIGN KEY(sale) REFERENCES Sale(Number);

ALTER TABLE

    Record ADD CONSTRAINT record\_artist\_foreign FOREIGN KEY(artist) REFERENCES Artist(id);

ALTER TABLE

    Instrument ADD CONSTRAINT instrument\_item\_id\_foreign FOREIGN KEY(item\_id) REFERENCES Item(item\_id);

ALTER TABLE

    Artist ADD CONSTRAINT artist\_record\_label\_foreign FOREIGN KEY(record\_label) REFERENCES Record\_label(id);

ALTER TABLE

    Worker ADD CONSTRAINT worker\_user\_id\_foreign FOREIGN KEY(ape\_user) REFERENCES Ape\_user(id);

ALTER TABLE

    Sale\_item ADD CONSTRAINT sale\_item\_item\_foreign FOREIGN KEY(item) REFERENCES Item(item\_id);

ALTER TABLE

    Record ADD CONSTRAINT record\_record\_label\_foreign FOREIGN KEY(record\_label) REFERENCES Record\_label(id);

ALTER TABLE

    Client ADD CONSTRAINT client\_user\_id\_foreign FOREIGN KEY(ape\_user) REFERENCES Ape\_user(id);

ALTER TABLE

    Item ADD CONSTRAINT item\_genre\_foreign FOREIGN KEY(genre) REFERENCES Genre(id);

ALTER TABLE

    Sale ADD CONSTRAINT sale\_user\_foreign FOREIGN KEY(ape\_user) REFERENCES Ape\_user(id);

ALTER TABLE

    Instrument\_istrument\_type ADD CONSTRAINT instrument\_istrument\_type\_instrument\_id\_foreign FOREIGN KEY(instrument\_id) REFERENCES Instrument(Instrument\_id);

За вмъкване на тестова информация е използван скрита:

Isert.sql

/\*

  Creating 5 Users

\*/

INSERT INTO Ape\_user(id, email, phone, password, first\_name, family\_name)

VALUES (1, 'ivo@mail.com', '089671253', 'ivo12345678', 'Ivaylo', 'Rumenov');

INSERT INTO Ape\_user(id, email, phone, password, first\_name, family\_name)

VALUES (2, 'asen@mail.com', '089412425', 'asen123asen', 'Asen', 'Hristove');

INSERT INTO Ape\_user(id, email, phone, password, first\_name, family\_name)

VALUES (3, 'damqn@mail.com', '09984932', 'damqnShefa', 'Damqn', 'Mihailov');

INSERT INTO Ape\_user(id, email, phone, password, first\_name, family\_name)

VALUES (4, 'hasan@mail.com', '08931265', 'hasendej', 'Hasan', 'Hassanov');

INSERT INTO Ape\_user(id, email, phone, password, first\_name, family\_name)

VALUES (5, 'user@mail.com', '08999999', '1234', 'User1', 'User1');

INSERT INTO Ape\_user(id, email, phone, password, first\_name, family\_name)

VALUES (6, 'muamed@mail.com', '0894234', 'muhadqnina123', 'Muhamed', 'Imal');

INSERT INTO Ape\_user(id, email, phone, password, first\_name, family\_name)

VALUES (7, 'IvanZvazdev@gmail.com', '0347573648', '112323231', 'Ivan', 'Zvezdev');

INSERT INTO Ape\_user(id, email, phone, password, first\_name, family\_name)

VALUES (8, 'KokoNikolov@mail.com', '08123123543', 'kdopasaoi', 'Koko', 'Nikolov');

INSERT INTO Ape\_user(id, email, phone, password, first\_name, family\_name)

VALUES (9, 'CecaIvanova@hotmail.com', '0892342176', 'Mamboamericano', 'Ceca', 'Ivanova');

INSERT INTO Ape\_user(id, email, phone, password, first\_name, family\_name)

VALUES (10, 'DamqnHristov@abv.bg', '089123789', 'Carqnakopona', 'Damqn', 'Hristov');

INSERT INTO Ape\_user(id, email, phone, password, first\_name, family\_name)

VALUES (11, 'Marakew213@outloock.com', '08398345', 'Dubail123', 'Samoil', 'Kovachev');

/\*

  Creating 3 Workers

\*/

INSERT INTO Worker(worker\_id,ape\_user,position,time\_stamp)

VALUES (1,1,'Meneger','2023-10-28');

INSERT INTO Worker(worker\_id,ape\_user,position,time\_stamp)

VALUES (2,2,'HR','2023-10-28');

INSERT INTO Worker(worker\_id,ape\_user,position,time\_stamp)

VALUES (3,3,'Support','2023-10-28');

/\*

  Creating 2 Clients

\*/

INSERT INTO Client(client\_id,ape\_user,status,time\_stamp)

VALUES (1,4,1,1);

INSERT INTO Client(client\_id,ape\_user,status,time\_stamp)

VALUES (2,5,2,1);

INSERT INTO Client(client\_id,ape\_user,status,time\_stamp)

VALUES (3,6,1,1);

INSERT INTO Client(client\_id,ape\_user,status,time\_stamp)

VALUES (4,7,2,1);

INSERT INTO Client(client\_id,ape\_user,status,time\_stamp)

VALUES (5,8,1,1);

INSERT INTO Client(client\_id,ape\_user,status,time\_stamp)

VALUES (6,9,2,1);

INSERT INTO Client(client\_id,ape\_user,status,time\_stamp)

VALUES (7,10,2,1);

INSERT INTO Client(client\_id,ape\_user,status,time\_stamp)

VALUES (8,11,2,1);

/\*

  Creating 3 Instrument types

\*/

INSERT INTO Instrument\_type(id,instrument\_type\_name,time\_stamp)

values (1,'Gittar','2023-10-28');

INSERT INTO Instrument\_type(id,instrument\_type\_name,time\_stamp)

values (2,'Drums','2023-10-28');

INSERT INTO Instrument\_type(id,instrument\_type\_name,time\_stamp)

values (3,'Brass','2023-10-28');

/\*

  Creating 3 Genres

\*/

INSERT INTO Genre(id,name,discretion,time\_stamp)

values (1,'Rock','you know','2023-10-28');

INSERT INTO Genre(id,name,discretion,time\_stamp)

values (2,'Jazz','you know','2023-10-28');

INSERT INTO Genre(id,name,discretion,time\_stamp)

values (3,'Pop','you know','2023-10-28');

INSERT INTO Genre(id,name,discretion,time\_stamp)

values (4,'Reggae','you know','2023-10-28');

INSERT INTO Genre(id,name,discretion,time\_stamp)

values (5,'Mettal','you know','2023-10-28');

/\*

  Creating 3 companies

\*/

INSERT INTO Company(id,name,time\_stamp)

values (1,'Febder','2023-10-28');

INSERT INTO Company(id,name,time\_stamp)

values (2,'LPD','2023-10-28');

INSERT INTO Company(id,name,time\_stamp)

values (3,'Yamaha','2023-10-28');

INSERT INTO Company(id,name,time\_stamp)

values (4,'Perl','2023-10-28');

INSERT INTO Company(id,name,time\_stamp)

values (5,'Marpex','2023-10-28');

INSERT INTO Company(id,name,time\_stamp)

values (6,'Thumb','2023-10-28');

/\*

  Creating 2 guitars and one bass Instruments.

\*/

INSERT INTO Item(item\_id,stock,price,discrimination,manufacturing\_date,name,description,genre,company,time\_stamp)

values (1,TRUE,32,'Fender\_Squier\_affinity\_Series\_Jazz\_Bass.jpg','2013-10-28','Fender\_Squier\_Classic','Some diescriopton',1,1,'2023-10-28');

INSERT INTO Item(item\_id,stock,price,discrimination,manufacturing\_date,name,description,genre,company,time\_stamp)

values (2,TRUE,54,'Fender\_Squier\_Affinity.jpg','2013-10-28','Fender\_Squier\_Affinity','Some diescriopton',1,1,'2023-10-28');

INSERT INTO Item(item\_id,stock,price,discrimination,manufacturing\_date,name,description,genre,company,time\_stamp)

values (3,TRUE,65,'Fender\_Squier\_Bass.jpg','2013-10-28','Yamaha\_YAS','Some diescriopton',2,3,'2023-10-28');

/\*

    Creating 3 Records

\*/

INSERT INTO Item(item\_id,stock,price,discrimination,manufacturing\_date,name,description,genre,company,time\_stamp)

values (4,TRUE,32,'Guns\_N\_Rosses\_GreatestHits.jpg','2013-10-28','Guns\_N\_Rosses\_GreatestHits','Some diescriopton',2,3,'2023-10-28');

INSERT INTO Item(item\_id,stock,price,discrimination,manufacturing\_date,name,description,genre,company,time\_stamp)

values (5,TRUE,43,'Nirvana\_Nevermind.JPG','2013-10-28','Nirvana\_Nevermind','Some diescriopton',2,3,'2023-10-28');

INSERT INTO Item(item\_id,stock,price,discrimination,manufacturing\_date,name,description,genre,company,time\_stamp)

values (6,TRUE,15,'Queen\_Greatest\_Hits\_1.jpg','2013-10-28','Queen\_Greatest\_Hits\_1','Some diescriopton',2,3,'2023-10-28');

INSERT INTO Item(item\_id,stock,price,discrimination,manufacturing\_date,name,description,genre,company,time\_stamp)

values (7,TRUE,65,'Yamaha\_YAS\_280\_S.jpg','2013-10-28','Yamaha\_YAS','Some diescriopton',2,3,'2023-10-28');

INSERT INTO Item(item\_id,stock,price,discrimination,manufacturing\_date,name,description,genre,company,time\_stamp)

values (8,TRUE,45,'Yamaha\_YAS\_280.jpg','2013-10-28','Yamaha\_YAS','Some diescriopton',2,3,'2023-10-28');

INSERT INTO Item(item\_id,stock,price,discrimination,manufacturing\_date,name,description,genre,company,time\_stamp)

values (9,TRUE,34,'Yamaha\_YEP\_642\_TS.jpg','2013-10-28','Yamaha\_YAS','Some diescriopton',2,3,'2023-10-28');

INSERT INTO Item(item\_id,stock,price,discrimination,manufacturing\_date,name,description,genre,company,time\_stamp)

values (10,TRUE,47,'Yamaha\_YEP\_642.jpg','2013-10-28','Yamaha\_YAS','Some diescriopton',2,3,'2023-10-28');

/\* Drum items \*/

INSERT INTO Item(item\_id,stock,price,discrimination,manufacturing\_date,name,description,genre,company,time\_stamp)

values (11,TRUE,500,'thumb\_base\_fdb9291a.jpg','2013-10-28','Tama MBSS55-SKA Starclassic','Some diescriopton',1,6,'2023-10-28');

INSERT INTO Item(item\_id,stock,price,discrimination,manufacturing\_date,name,description,genre,company,time\_stamp)

values (12,TRUE,1340,'Ymapex-armory-ocean-sunset-ar628sfujg.jpg','2013-10-28','Ymapex-armory','Some diescriopton',1,5,'2023-10-28');

INSERT INTO Item(item\_id,stock,price,discrimination,manufacturing\_date,name,description,genre,company,time\_stamp)

values (13,TRUE,1120,'EXL725S-C246-large.jpg','2013-10-28','Perl','Some diescriopton',1,4,'2023-10-28');

/\* Second set of records\*/

INSERT INTO Item(item\_id,stock,price,discrimination,manufacturing\_date,name,description,genre,company,time\_stamp)

values (14,TRUE,123,'Tyler\_The\_Creator\_Igor.jpg','2013-10-28','Tyler The Creator Igor','Some diescriopton',3,6,'2023-10-28');

INSERT INTO Item(item\_id,stock,price,discrimination,manufacturing\_date,name,description,genre,company,time\_stamp)

values (15,TRUE,32,'Tama\_Impala\_Currents.jpg','2013-10-28','Tama Impala Currents','Some diescriopton',1,5,'2023-10-28');

INSERT INTO Item(item\_id,stock,price,discrimination,manufacturing\_date,name,description,genre,company,time\_stamp)

values (16,TRUE,54,'The\_Bearles\_Abbey\_Read.JPG','2013-10-28','The Bearles Abbey Read','Some diescriopton',1,4,'2023-10-28');

/\* Metalica\*/

INSERT INTO Item(item\_id,stock,price,discrimination,manufacturing\_date,name,description,genre,company,time\_stamp)

values (17,TRUE,44,'Metakkica\_Garage\_Inc.jpg','2013-10-28','Metakkica Garage Inc','Some diescriopton',5,6,'2023-10-28');

INSERT INTO Item(item\_id,stock,price,discrimination,manufacturing\_date,name,description,genre,company,time\_stamp)

values (18,TRUE,35,'Metallica.JPG','2013-10-28','Kill dem all','Some diescriopton',5,5,'2023-10-28');

INSERT INTO Item(item\_id,stock,price,discrimination,manufacturing\_date,name,description,genre,company,time\_stamp)

values (19,TRUE,56,'Metallica\_Hardwired\_To\_Self\_Destruct.JPG','2013-10-28','Hardwired To Self Destruct','Some diescriopton',5,4,'2023-10-28');

/\* Bob marly \*/

INSERT INTO Item(item\_id,stock,price,discrimination,manufacturing\_date,name,description,genre,company,time\_stamp)

values (20,TRUE,46,'Bob\_Marley\_Natty\_Dread\_Tour\_75.jpg','2013-10-28','Natty Dread Tour','Some diescriopton',4,6,'2023-10-28');

INSERT INTO Item(item\_id,stock,price,discrimination,manufacturing\_date,name,description,genre,company,time\_stamp)

values (21,TRUE,75,'Bob\_Marley\_Uprising\_Live.jpg','2013-10-28','Uprising Live','Some diescriopton',4,5,'2023-10-28');

INSERT INTO Item(item\_id,stock,price,discrimination,manufacturing\_date,name,description,genre,company,time\_stamp)

values (22,TRUE,45,'Bob+Marley\_Songs\_Of\_Freedom.jpg','2013-10-28','Songs Of Freedom','Some diescriopton',4,4,'2023-10-28');

/\*

  Creating 3 Instruments

\*/

INSERT INTO Instrument(Instrument\_id,item\_id,type,time\_stamp)

VALUES (1,1,1,'2023-10-28');

INSERT INTO Instrument(Instrument\_id,item\_id,type,time\_stamp)

VALUES (2,2,2,'2023-10-28');

INSERT INTO Instrument(Instrument\_id,item\_id,type,time\_stamp)

VALUES (3,3,1,'2023-10-28');

INSERT INTO Instrument(Instrument\_id,item\_id,type,time\_stamp)

VALUES (4,7,1,'2023-10-28');

INSERT INTO Instrument(Instrument\_id,item\_id,type,time\_stamp)

VALUES (5,8,1,'2023-10-28');

INSERT INTO Instrument(Instrument\_id,item\_id,type,time\_stamp)

VALUES (6,9,1,'2023-10-28');

INSERT INTO Instrument(Instrument\_id,item\_id,type,time\_stamp)

VALUES (7,10,1,'2023-10-28');

INSERT INTO Instrument(Instrument\_id,item\_id,type,time\_stamp)

VALUES (8,11,2,'2023-10-28');

INSERT INTO Instrument(Instrument\_id,item\_id,type,time\_stamp)

VALUES (9,12,2,'2023-10-28');

INSERT INTO Instrument(Instrument\_id,item\_id,type,time\_stamp)

VALUES (10,13,2,'2023-10-28');

/\*

  Creating 3 Instruments

\*/

INSERT INTO Instrument\_istrument\_type(instrument\_id,instrument\_type\_id)

VALUES (1,1);

INSERT INTO Instrument\_istrument\_type(instrument\_id,instrument\_type\_id)

VALUES (2,1);

INSERT INTO Instrument\_istrument\_type(instrument\_id,instrument\_type\_id)

VALUES (3,3);

INSERT INTO Instrument\_istrument\_type(instrument\_id,instrument\_type\_id)

VALUES (4,3);

INSERT INTO Instrument\_istrument\_type(instrument\_id,instrument\_type\_id)

VALUES (5,3);

INSERT INTO Instrument\_istrument\_type(instrument\_id,instrument\_type\_id)

VALUES (6,3);

INSERT INTO Instrument\_istrument\_type(instrument\_id,instrument\_type\_id)

VALUES (7,3);

INSERT INTO Instrument\_istrument\_type(instrument\_id,instrument\_type\_id)

VALUES (8,2);

INSERT INTO Instrument\_istrument\_type(instrument\_id,instrument\_type\_id)

VALUES (9,2);

INSERT INTO Instrument\_istrument\_type(instrument\_id,instrument\_type\_id)

VALUES (10,2);

/\*

  Creating 3 Record label

\*/

INSERT INTO Record\_label(id,Name,description,time\_stamp)

VALUES (1,'Wiener Records','Some diescriopton','2023-10-28');

INSERT INTO Record\_label(id,Name,description,time\_stamp)

VALUES (2,'Kanine Records','Some diescriopton','2023-10-28');

INSERT INTO Record\_label(id,Name,description,time\_stamp)

VALUES (3,'Aftermath Entertainment','Some diescriopton','2023-10-28');

/\*

  Creating 3 Artists

\*/

INSERT INTO Artist(id,Name,record\_label,time\_stamp)

VALUES (1,'GunsNRoses',1,'2023-10-28');

INSERT INTO Artist(id,Name,record\_label,time\_stamp)

VALUES (2,'Nirvana',1,'2023-10-28');

INSERT INTO Artist(id,Name,record\_label,time\_stamp)

VALUES (3,'Queen',1,'2023-10-28');

INSERT INTO Artist(id,Name,record\_label,time\_stamp)

VALUES (4,'Tyler The Creator',1,'2023-10-28');

INSERT INTO Artist(id,Name,record\_label,time\_stamp)

VALUES (5,'Tema',1,'2023-10-28');

INSERT INTO Artist(id,Name,record\_label,time\_stamp)

VALUES (6,'The Bearles',1,'2023-10-28');

INSERT INTO Artist(id,Name,record\_label,time\_stamp)

VALUES (7,'Metalica',1,'2023-10-28');

INSERT INTO Artist(id,Name,record\_label,time\_stamp)

VALUES (8,'Bob Marley',1,'2023-10-28');

/\*

  Creating 3 Records

\*/

INSERT INTO Record(Record\_id,item\_id,artist,record\_label,time\_stamp)

VALUES (1,4,1,1,'2023-10-28');

INSERT INTO Record(Record\_id,item\_id,artist,record\_label,time\_stamp)

VALUES (2,5,2,1,'2023-10-28');

INSERT INTO Record(Record\_id,item\_id,artist,record\_label,time\_stamp)

VALUES (3,6,3,3,'2023-10-28');

INSERT INTO Record(Record\_id,item\_id,artist,record\_label,time\_stamp)

VALUES (4,14,4,1,'2023-10-28');

INSERT INTO Record(Record\_id,item\_id,artist,record\_label,time\_stamp)

VALUES (5,15,5,1,'2023-10-28');

INSERT INTO Record(Record\_id,item\_id,artist,record\_label,time\_stamp)

VALUES (6,16,6,3,'2023-10-28');

INSERT INTO Record(Record\_id,item\_id,artist,record\_label,time\_stamp)

VALUES (7,17,7,2,'2023-10-28');

INSERT INTO Record(Record\_id,item\_id,artist,record\_label,time\_stamp)

VALUES (8,18,7,2,'2023-10-28');

INSERT INTO Record(Record\_id,item\_id,artist,record\_label,time\_stamp)

VALUES (9,19,7,2,'2023-10-28');

INSERT INTO Record(Record\_id,item\_id,artist,record\_label,time\_stamp)

VALUES (10,20,8,1,'2023-10-28');

INSERT INTO Record(Record\_id,item\_id,artist,record\_label,time\_stamp)

VALUES (11,21,8,1,'2023-10-28');

INSERT INTO Record(Record\_id,item\_id,artist,record\_label,time\_stamp)

VALUES (12,22,8,1,'2023-10-28');

/\*

  Creating 10 Sale

\*/

INSERT INTO Sale(Number,ape\_user,amount,fulfill,time\_stamp)

values (1,4,44,TRUE,'2023-10-28');

INSERT INTO Sale(Number,ape\_user,amount,fulfill,time\_stamp)

values (2,4,44,TRUE,'2023-10-28');

INSERT INTO Sale(Number,ape\_user,amount,fulfill,time\_stamp)

values (3,4,123,TRUE,'2023-10-28');

INSERT INTO Sale(Number,ape\_user,amount,fulfill,time\_stamp)

values (4,4,432,TRUE,'2023-10-28');

INSERT INTO Sale(Number,ape\_user,amount,fulfill,time\_stamp)

values (5,4,425,TRUE,'2023-10-28');

INSERT INTO Sale(Number,ape\_user,amount,fulfill,time\_stamp)

values (6,4,52,TRUE,'2023-10-28');

INSERT INTO Sale(Number,ape\_user,amount,fulfill,time\_stamp)

values (7,4,36,TRUE,'2023-10-28');

INSERT INTO Sale(Number,ape\_user,amount,fulfill,time\_stamp)

values (8,4,16,TRUE,'2023-10-28');

INSERT INTO Sale(Number,ape\_user,amount,fulfill,time\_stamp)

values (9,4,134,TRUE,'2023-10-28');

INSERT INTO Sale(Number,ape\_user,amount,fulfill,time\_stamp)

values (10,4,543,TRUE,'2023-10-28');

INSERT INTO Sale(Number,ape\_user,amount,fulfill,time\_stamp)

values (12,5,543,TRUE,'2023-10-28');

/\*

  Creating 3 Sale\_item

\*/

INSERT INTO Sale\_item(sale,item)

values (1,1);

INSERT INTO Sale\_item(sale,item)

values (1,2);

INSERT INTO Sale\_item(sale,item)

values (1,3);

INSERT INTO Sale\_item(sale,item)

values (2,1);

INSERT INTO Sale\_item(sale,item)

values (2,3);

INSERT INTO Sale\_item(sale,item)

values (4,2);

INSERT INTO Sale\_item(sale,item)

values (5,6);

INSERT INTO Sale\_item(sale,item)

values (6,7);

INSERT INTO Sale\_item(sale,item)

values (7,7);

INSERT INTO Sale\_item(sale,item)

values (8,8);

INSERT INTO Sale\_item(sale,item)

values (9,9);

INSERT INTO Sale\_item(sale,item)

values (10,10);

INSERT INTO Sale\_item(sale,item)

values (10,1);

INSERT INTO Sale\_item(sale,item)

values (10,1);

За бизнес логита са използвани следните inset, create, update, delete:

Select.sql

/\*

    Get all Instruments

\*/

SELECT

    it.instrument\_type\_name,

    i3.name as item\_name,c.name,i3.stock,i3.price,i3.manufacturing\_date, i3.discrimination

    ,g.name as gener\_name

FROM instrument

join public.item i3 on i3.item\_id = instrument.item\_id

join public.genre g on g.id = i3.genre

join public.instrument\_istrument\_type iit on instrument.instrument\_id = iit.instrument\_id

join public.instrument\_type it on it.id = iit.instrument\_type\_id

join public.company c on c.id = i3.company;

insert into instrument\_istrument\_type(instrument\_id, instrument\_type\_id)

VALUES(10,1);

select \* from genre;

/\*

    Get all records

\*/

SELECT i.name as item\_name ,i.price,i.stock,a.name as artist\_name,rl.name as record\_label\_name from record

join public.artist a on a.id = record.artist

join public.record\_label rl on rl.id = record.record\_label

join public.item i on i.item\_id = record.item\_id;

/\*

    Get all artist records

\*/

SELECT artist.name as artist\_name, rl.name as record\_label\_name from artist

    join public.record r on artist.id = r.artist

join public.record\_label rl on artist.record\_label = rl.id;

/\*

    Get artist by name

 \*/

SELECT artist.name as artist\_name, rl.name as record\_label\_name from artist

join public.record r on artist.id = r.artist

join public.record\_label rl on artist.record\_label = rl.id

where artist.name='Queen'

Limit 1;

/\*

    todo

    sell item

        1-add to sale\_item

        2-add to sale

        3-finish

    see all sales and user to dem

    WEB

    add drop down to admin meny

    add id to the tables & order them by id

    add user profile

\*/

/\*

    All sales for a user

\*/

SELECT sale.number, au.email from sale

join public.ape\_user au on au.id = sale.ape\_user

where au.id=4

group by au.email, sale.number;

SELECT sale.number , sale.amount, sale.time\_stamp, sale.fulfill from sale

join public.ape\_user au on au.id = sale.ape\_user

where au.id=4

group by au.email, sale.number;

/\*

    Update user status

\*/

    update client set status=2 where ape\_user=1;

/\*

 Most sold item

 \*/

SELECT count(sale\_item.item) as count\_,i.name, i.discrimination as img\_url from sale\_item

    join public.item i on sale\_item.item=i.item\_id

group by i.name, i.discrimination

order by count\_ desc

LIMIT 1;

/\*

    Sale a item

\*/

/\* Step 0: get users curent shoping cart if he hsae one\*/

SELECT s.number from ape\_user

join public.sale s on ape\_user.id = s.ape\_user

where s.fulfill=true and s.ape\_user=4;

    SELECT client.status from client

        where client\_id =1;

/\* Step 1: make a Sale and add a item\*/

insert into sale(number, ape\_user, amount, fulfill, time\_stamp)

values (11,2,0,false, CURRENT\_DATE);

/\* Stem 2: Persist clients shopping cart id as sale.number

   and add new items as the clients adds them to the cart

   where 1 is the new item

   where 11 is the users shopping cart\*/

insert into sale\_item(sale, item)

values (11,1);

/\*delete item from shopping cart\*/

delete from Sale\_item where sale\_item.sale=? && sale\_item.item=?;

/\* Step 3: Finish the order and set to user\*/

update sale set fulfill=true where number=?;

/\* Get users shpoing card whit items by user id\*/

select si.item from sale

join public.sale\_item si on sale.number = si.sale

join public.ape\_user au on sale.ape\_user = au.id

where au.id=4 and sale.fulfill=false;

/\*

    Get by Instrument Type

\*/

/\* Select all instrument types for drop down\*/

SELECT instrument\_type\_name from instrument\_type;

/\* Get all Brass instruments\*/

SELECT

    it.instrument\_type\_name,

    i3.name as item\_name,c.name,i3.stock,i3.price,i3.manufacturing\_date, i3.discrimination

        ,g.name as gener\_name

FROM instrument

         join public.item i3 on i3.item\_id = instrument.item\_id

         join public.genre g on g.id = i3.genre

         join public.instrument\_istrument\_type iit on instrument.instrument\_id = iit.instrument\_id

         join public.instrument\_type it on it.id = iit.instrument\_type\_id

         join public.company c on c.id = i3.company

WHERE it.id=3;

/\*

    Get by Records Genre

\*/

/\* Select all genres for drop down\*/

SELECT name from genre;

SELECT i.name as item\_name ,i.price,i.stock,

       a.name as artist\_name,rl.name as record\_label\_name from record

           join public.artist a on a.id = record.artist

           join public.record\_label rl on rl.id = record.record\_label

           join public.item i on i.item\_id = record.item\_id

where i.genre=2;

/\*

    See all used sales

\*/

select sale.amount,sale.time\_stamp as date,au.email as user from sale

join public.ape\_user au on au.id = sale.ape\_user

join public.client c on au.id = c.ape\_user

where sale.fulfill=true;

/\*

    Drop down selects

\*/

select artist.name from artist;

select company.name from company;

SELECT name from genre;

SELECT instrument\_type\_name from instrument\_type;

select rl."name"  from record\_label rl;

/\* All record names\*/

select i."name"  from record r

join public.item i on i.item\_id = r.item\_id ;

/\* All instrument names\*/

select i."name"  from instrument i2

join public.item i on i.item\_id = i2.item\_id ;

/\*

    user profile

\*/

SELECT \* from ape\_user

join public.client cl on cl.ape\_user=ape\_user.id

where ape\_user.id=4;

/\*

    All DELETES

\*/

/\* Delete Intrument \*/

delete from instrument\_istrument\_type where instrument\_istrument\_type.instrument\_id=8;

delete from instrument where instrument.instrument\_id=8;

delete from item where item.item\_id=11;

/\*delete item from shopping cart\*/

delete from Sale\_item where sale\_item.sale=? && sale\_item.item=?;

/\*

    Delete record

\*/

delete from record where record.Record\_id=12;

delete from item where item\_id=22;

select i.item\_id from record

join public.item i on i.item\_id = record.item\_id

where record\_id=11;

select record\_id from record

join public.item i on i.item\_id = record.item\_id

where i.item\_id=18;

select instrument\_id from instrument

join public.item i on i.item\_id = instrument.item\_id

where i.item\_id=2;

delete from instrument\_istrument\_type where instrument\_id=1;

delete from instrument where instrument\_id=1;

delete from sale\_item where item=1;

delete from item where item\_id=1;

За затриване на базата от данни и техните полета е използван следния скрипт

Drop.sql

DROP TABLE Client;

DROP TABLE Worker;

DROP TABLE Sale\_item;

DROP TABLE Sale;

DROP TABLE User;

DROP TABLE Instrument\_istrument\_type;

DROP TABLE Record;

DROP TABLE Artist;

DROP TABLE Record\_label;

DROP TABLE Instrument\_type;

Drop TABLE Instrument;

DROP TABLE Item;

DROP TABLE Genre;

DROP TABLE Company;