**Софийски универитет „Св. Климент Охидски“**

*Факултет по математика и информатика*

*Специалност: “Информационни системи ”*

*Курс: 2*

*Дисциплина: “Бази от данни-практикум ”*

***Курсов Проект***

*Тема №7: пицария „Донатело“*

***Изготвил:***

Красимира Станкова Ф.№71839

Даян Митев Ф.№71908

Станислав Арабов Ф.№71938

***Преподавател:***

доц. Радослава Христова

София

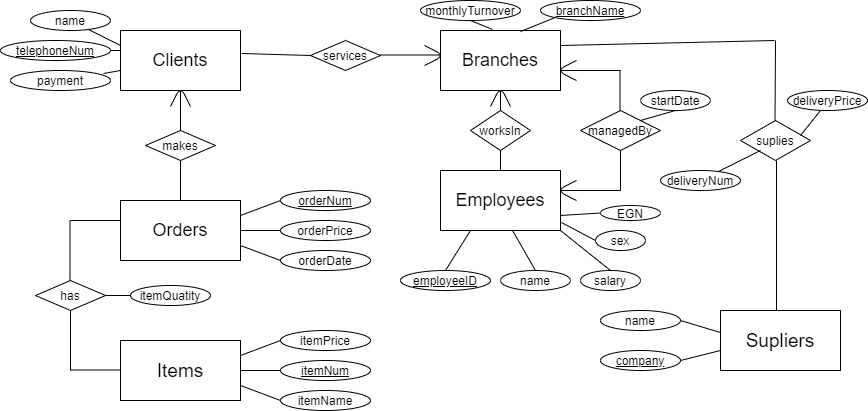
Летен семестър 2019/2020

# Описание на проекта

Базата от данни съхранява данни и информация за пица-ресторант „Донатело“. Разработена изключително просто и удобно за обикновения протребител, базата данни може лесно да се използва и от други ресторанти. Проекта се състои от 8 таблици, които съдържат както основна така и помощна информация за функционирането на системата. Разработени са функции и изгледи за бърза и лесна справка от служителите за менюто, дневния и месечния оборот и т.н. Също така са създадени тригери, с чиято помощ системата уведомява потребителя, под формата на различни съобщения, за различни извършени корекции в базата от данни.

# Модел на базата от данни

* 1. Модел „Същност- отношение“



* 1. Описание на множествата същности

Модела на базата от данни се състои от 6 множества същности:

* Clients – съдържа информация за клиентите на пицарията
* Branches – пази информация за различните отдели, работещи в ресторанта
* Employees – съдържа информация за служителите, работещи в различните отдели, и за техните началници
* Suppliers – тук е информацията за всички фирми, снабдяващи пицарията с продукти
* Orders – информация за направените поръчки
* Items – съдържа информация за артикулите, които се предлагат в пицарията
  1. Ограничения и атрибути
* Clients – telephoneNum, name, payment:

Тъй като може да има клиенти с еднакви имена, ключ, по-който ще се разграничават клиентите, е телефонният им номер. А в атрибута payment се държи информация за начина на плащане на клиента(с карта или в брой).

* Branches – branchName, monthlyTurnover:

Няма как да има два отдела с едно и също име, за това името на отдела е ключ на множеството същности , а monthlyTurnover се съдържа информация за месечния оборот на отделите.

* Employees – employeeID, EGN, name, salary, sex:

Служителите се разграничават един от друг по уникален номер, генериран автоматично от системата.

* Suppliers – company, name:

Тъй като в различните фирми- доставчици може да има служители с еднакви имена за ключ е определено името на компанията, тъй като възможността две фирми да са с едно и също име е минимална.

* Orders – orderNum, orderDate, orderPrice:

Поръчките се различават една от друга по техният уникален номер, отново генериран автоматично от системата. Атрибутите orderDate и orderPrice са добавени за по-лесното извеждане на справки от потребителите.

* Items – itemNum, itemPrice, itemName:

Както поръчките, артикулите също си имат уникален номер, генериран автоматично от системата. Името и цената на артикулите се включват в разработката на менюто и други справки.

* 1. Преобразуване от модел „Същност- отношение“ в релационен модел

С преобразуването от модела „Същност- отношение“ в релационния модел се получават общо 8 таблици:

**Clients** (telephoneNumber, name, payment, branchName)

**Orders** (orderNumber, orderPrice, orderDate, clientTelephone)

**Branches** (branchName, montlyTurnover, mngrID, mngrStartDate)

**Employees** (employeeID, EGN, employeeName, salary, sex, branchName)

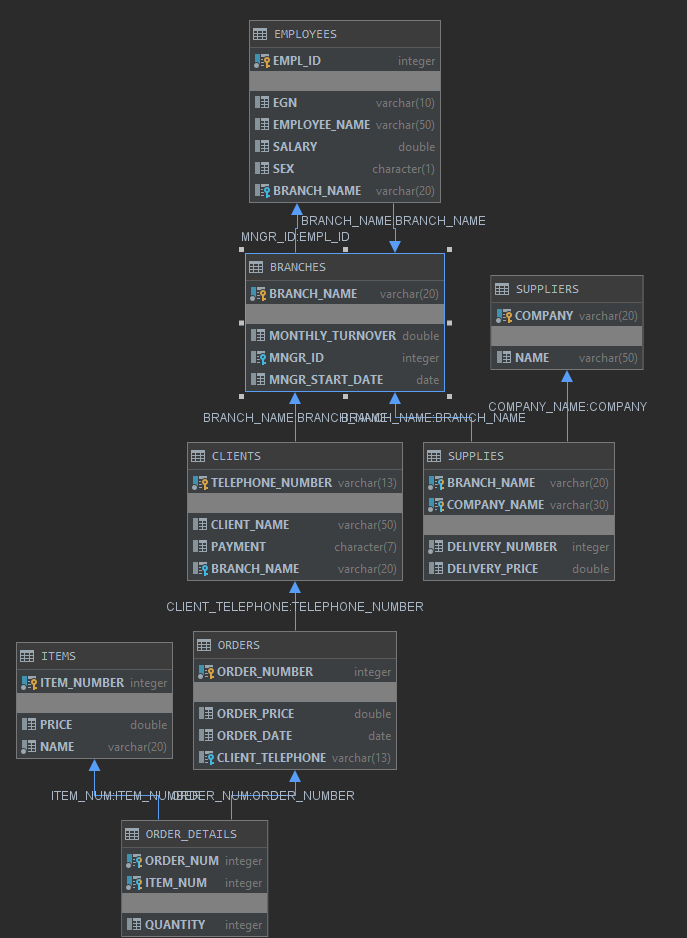
**Suppliers** (company, supplierName)

**Items** (itemNumber, itemPrice, itemName)

**Supplies** (branchName, companyName, deliveryNum, deliveryPrice)

**OrderDetails** (orderNumber, itemNumber, itemQuantity)

* 1. Картинка на релационния модел



# Описание на функциите

* 1. Функция, която извежда справка за днеяния оборот

**CREATE FUNCTION *DAILY\_TURNOVER*(CURRDATE DATE)  
RETURNS FLOAT  
SPECIFIC DAILY\_TURNOVER  
RETURN  
 SELECT *SUM*(ORDER\_PRICE)  
 FROM ORDERS  
 WHERE ORDER\_DATE = CURRDATE;**

* 1. Функция, която връща таблица с името на всеки клиент и срещу него начина му на плащане, за текущата дата.

**CREATE FUNCTION *PAYMENT\_TYPE\_CLIENTS*(CURRDATE DATE)  
RETURNS TABLE  
(  
 CLIENT\_NAME VARCHAR(50),  
 PAYMENT CHAR(7)  
)  
SPECIFIC PAYMENT\_TYPE\_CLIENTS  
RETURN  
 SELECT CLIENT\_NAME, PAYMENT  
 FROM CLIENTS CL, ORDERS O  
 WHERE (O.CLIENT\_TELEPHONE LIKE CL.TELEPHONE\_NUMBER)  
 AND (ORDER\_DATE = CURRDATE);**

* 1. Функция, която извежда търсен елемент, заедно с цената му

**CREATE FUNCTION *SEARCH\_FOR\_MEAL*(S\_NAME VARCHAR(60))  
RETURNS TABLE (NAME VARCHAR(60) , PRICE DOUBLE )  
BEGIN ATOMIC  
 DECLARE S\_ITEM VARCHAR(60);  
 SET S\_ITEM = ('%' || S\_NAME || '%');  
RETURN SELECT I.NAME , I.PRICE  
 FROM ITEMS I  
 WHERE I.NAME LIKE S\_ITEM;  
end;**

* 1. Функция, която извежда месечния оборот на всеки отдел

**CREATE FUNCTION *GET\_ALL\_BRANCHES*(branch VARCHAR(20))  
RETURNS TABLE  
(  
 BRANCH\_NAME varchar(20),  
 TURNOVER DOUBLE  
)  
RETURN  
 SELECT b.BRANCH\_NAME, b.MONTHLY\_TURNOVER  
 FROM BRANCHES b  
 WHERE BRANCH\_NAME != branch;**

* 1. Функция, която извежда всички артикули в менюто с техните цени

**CREATE FUNCTION *ITEMS\_INFO*(I\_PRICE DOUBLE)  
RETURNS TABLE (NAME VARCHAR(60) , PRICE DOUBLE)  
RETURN  
 SELECT I.NAME , I.PRICE  
 FROM ITEMS I  
 WHERE I.PRICE < I\_PRICE;**

# Описание на тригерите

* 1. Тригер, който записва в таблица промените в заплащането на служителите

**CREATE TABLE AUDIT(CTIME TIMESTAMP, TEXT VARCHAR(200));  
  
CREATE TRIGGER TRIG\_UPD\_EMP  
 AFTER UPDATE OF SALARY ON EMPLOYEES  
 REFERENCING OLD AS O NEW AS N  
 FOR EACH ROW  
 WHEN (O.SALARY != N.SALARY)  
 BEGIN  
 DECLARE V\_TEXT VARCHAR(200);  
 SET V\_TEXT = USER || ' EMPL\_ID = ' || O.EMPL\_ID  
 || ' OLD SALARY = ' || *CHAR*(O.SALARY)  
 || 'NEW SALARY = ' || *CHAR*(N.SALARY);  
 INSERT INTO AUDIT VALUES(CURRENT\_TIMESTAMP, V\_TEXT);  
 END;**

* 1. Тригер, който извежда съобщение при въвеждането на нова поръчка

**CREATE TABLE TRIG\_INSERTION(  
 MESSAGE VARCHAR(100)  
);  
CREATE TRIGGER TRIG\_INS\_ORDER  
 BEFORE INSERT ON ORDERS  
 FOR EACH ROW BEGIN  
 INSERT INTO TRIG\_INSERTION VALUES ('added new order');  
END;**

* 1. Тригер, който пази в таблица информация за направените доставки на артикули към пицарията, като пази информация за фирмата, направила доставката, и за отдела, към който е направена доставката

**CREATE TABLE NEW\_SUPPLIES(  
 SUPP\_MASSAGE VARCHAR(250)  
);  
  
CREATE TRIGGER TRIG\_NEW\_SUPPLIES  
 AFTER INSERT ON SUPPLIES  
 REFERENCING NEW AS N  
 FOR EACH ROW  
 INSERT INTO NEW\_SUPPLIES(SUPP\_MASSAGE)  
 VALUES ('New delivery was made to branch: ' || N.BRANCH\_NAME ||  
 ' , By company: ' || N.COMPANY\_NAME || '. Cost:' || *CHAR*(N.DELIVERY\_PRICE));**

* 1. Тригер, който извежда информация, когато служител бъде премахната от базата данни

**CREATE TABLE trigger\_test(  
 message VARCHAR(100)  
);  
  
CREATE TRIGGER TRIG\_INSERT\_EMP  
 AFTER DELETE ON EMPLOYEES  
 FOR EACH ROW BEGIN  
 INSERT INTO trigger\_test VALUES ('employee has been deleted');  
END;**

# Описание на изгледите

* 1. Изглед, който извежда плащанията в брой на текущата дата

**CREATE VIEW PAYMENT\_REFERENCE  
AS  
 SELECT PAYMENT, ORDER\_DATE  
 FROM CLIENTS, ORDERS  
 WHERE ORDER\_DATE = CURRENT\_DATE AND PAYMENT= 'IN CASH';**

* 1. Изгледи, които извеждат броя на служителите в даден отдел и мениджъра на този отдел

**CREATE VIEW V\_CNT\_EMP\_SERV  
AS  
 SELECT *COUNT*(\*) AS CNT\_EMP  
 FROM EMPLOYEES  
 WHERE EMPLOYEES.BRANCH\_NAME = 'Service';**

**CREATE VIEW V\_CNT\_EMP\_BAR  
AS  
 SELECT *COUNT*(\*) AS CNT\_EMP  
 FROM EMPLOYEES  
 WHERE EMPLOYEES.BRANCH\_NAME = 'Bartenders';**

**CREATE VIEW V\_CNT\_EMP\_CHEF  
AS  
 SELECT *COUNT*(\*) AS CNT\_EMP  
 FROM EMPLOYEES  
 WHERE EMPLOYEES.BRANCH\_NAME = 'Chef';**

* 1. Изглед, който извежда имената на всички артикули с цена над 12лв.

**CREATE VIEW V\_ITEMS  
AS  
 SELECT NAME , PRICE  
 FROM ITEMS  
 WHERE PRICE > 12;**

# Заключение

В бъдеще базата от данни може да се развие изключително много, като се добавят възможности за извеждане на още много и разнообразни справки и отчети. Като може да се запази простият дизайн и удобнят потребителски интерфейс на системата.