Софийски университет ,,Св. Климент Охридски“

Факултет по математика и информатика

Курсов проект

На тема

Хотел&Ресторант

По СУБД

Изготвил:

Веселина Чотрева

ИС

3 курс

ФН: 71840

Описание на множествата същности

Проектът е продължение на курсовият проект, изготвен по курса Бази данни.

- MyRooms /Стаи/

Множеството същности описва стая в хотел, като негови атрибути са roomNumber /номер на стая/, roomType /тип/ и pricePerNight /цена/. Ключовият атрибут е roomNumber. Стаите могат да са единични, двойни и апартаменти (тройни). В базата от данни те са означени с цифрите 1 – единична, 2 – двойна, 3 – апартамент. И трите атрибута са цели положителни числа.

- Staff /Служители/

  Множеството от същности Staff характеризира служителите в хотела, като всички те имат следните атрибути: код на служител /ID/, име /name/, заплата /salary/, бонуси /bonuses/, дата на постъпване на работа /hireDate/, телефонен номер /phoneNumber/, имейл /email/. Заплатата и бонусите са положителни числа, датата на постъпване е от тип дата. ID е ключът и е представлява символен низ с точно 6 символа. Името е символен низ с дължина 50 символа. Телефонният номер и имейлът също са символни низове с дължини съответно 10 и 30 символа. Служителите могат да бъдат: Управители /MyManagers/, Рецепционисти/MyReceptionists/ и Камериери /MyAttendants/. Всеки от тях наследява множеството Staff, като добавя и свои собствени атрибути, които го отличават от останалите служители.

- MyManagers /Управители/

Множеството същности описва харакетистиките на управители в хотел. Един управител отговаря за определени служители в хотела – както за камериерите, така и за рецепционистите.

Мениджърът се отличава от останалите служители с фиксирана заплата, като също може да получава бонуси.

- MyReceptionists /Рецепционисти/

Множеството същности описва рецепционист в хотел. То наследява множеството същности Staff, като добавя и собствен атрибут – образование /education/, което е от тип символен низ. Означенията му в базата от данни са следните: A – средно, B – средно специално, C – висше.Рецепционистите отговарят за регистрирането на гостите и тяхното заплащане зависи от степента на образование, което притежават.

- MyAttendants /Камериери/

  Множеството същности MyAttendants описва служителите в хотела, които отговарят за чистотата и състоянието на стаите. Освен наследените атрибути са добавени и: изработени часове /worked hours/ и ставка /moneyPerHour/. И двата трябва да положителни числа, но ставката не е задължително да е цяло число. Всеки камериер отговаря за определени стаи. Заплащането му се формира като броят на изработените часове се умножава по определен коефициент (ставка).
- MyCooks /Готвачи/

  Множеството същности MyCooks описва служителите в хотела, които работят в ресторанта на хотела. То наследява множеството същности Staff, като добавя и собствен атрибут позиция - TYPE_OF_POSITION CHAR, от тип символен низ с дължина 10 символа.
- MyGuests /Гости/

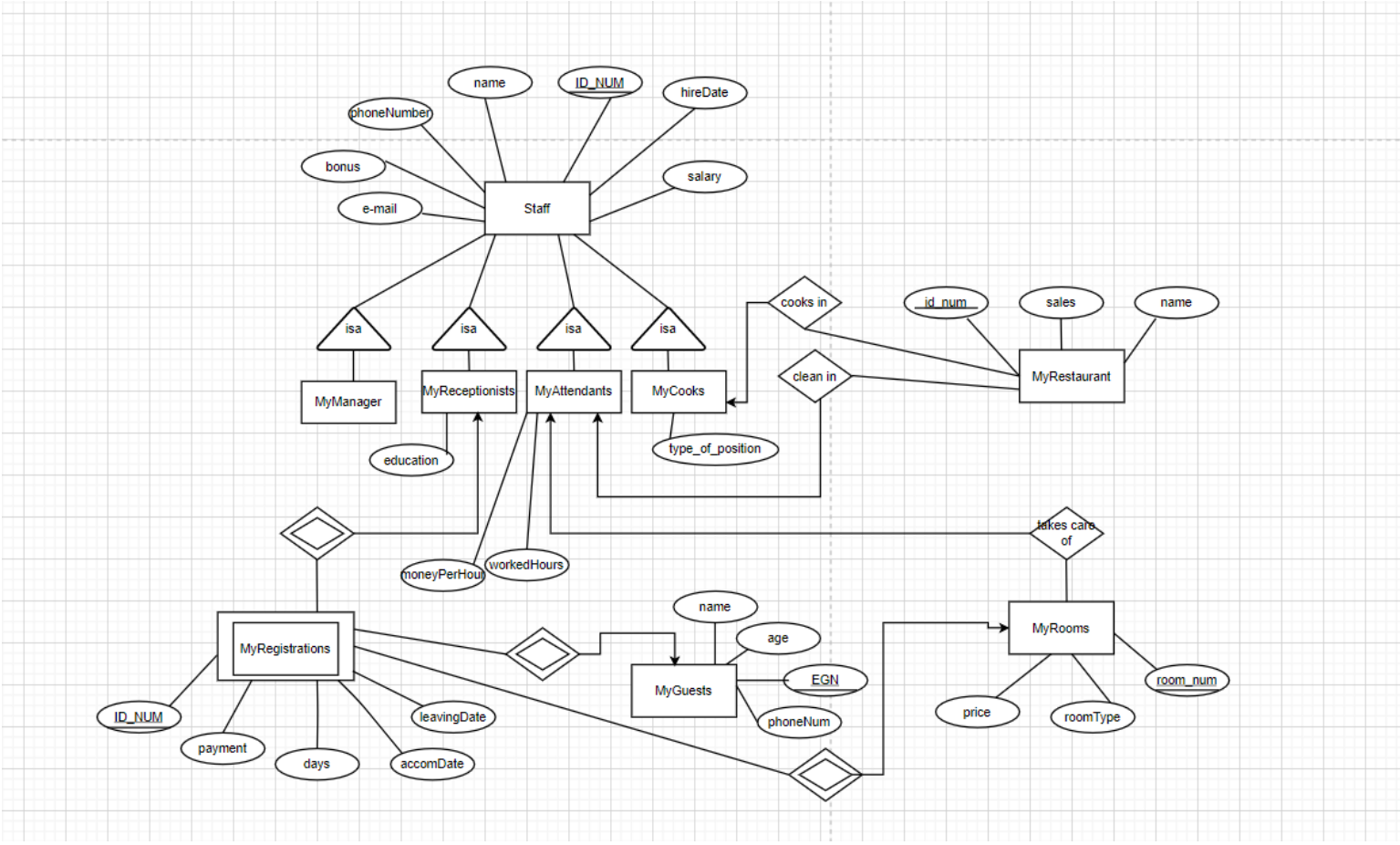
  Това множество същности описва гостите в хотела. Те имат ЕГН /EGN/, име /namee/, възраст /age/ и телефонен номер /phoneNumber/. ЕГН-то е символен низ с точно 10

символа. Това е и ключът. Името и телефонният номер също са символни низове.

- MyRestaurant

  Множеството същности MyRestaurant представя различни видове заведения, които могат да се съдържат в хотела, а в конкретния случай са три – Restaurant, Cafeteria и Night club. MyRestaurant има свои собствени атрибути /id_num/ номер на заведението, /NAMEE/ име, / SALES/ промоции.

E\R диаграма на модела на БД (Картинка на диаграмата)



Преобразуване от E\R модел към релационен модел

Staff (ID_num, name, salary, bonus, emplDate, phoneNumber, e-mail)

Staff_Manager (ID_num, name, salary, bonuses, emplDate, phoneNumber, e-mail)

Staff_Attendants (ID_num, name, salary, bonuses, emplDate, phoneNumber, e-mail, workedHours, moneyPerHour)

Staff_Receptionists (ID_num, name, salary, bonuses, emplDate, phoneNumber, e-mail, education)

Staff_Cooks(ID_Num, salary, bonuses, emplDate, phoneNumber, e-mail,type_of_position)

Staff_Receptionists_Managers(ID_num,name, salary, bonuses, emplDate, phoneNumber, email, education)

Staff_Receptionists_Attendants (ID_num, name, salary, bonuses, emplDate, phoneNumber, e-mail, education, workedHours, moneyPerHour)

Staff_Managers_Attendants (ID_num, name, salary, bonuses, emplDate, phoneNumber, e-mail, workedHours, moneyPerHour)

Staff_Cooks_Attendants(ID_num,name, salary, bonuses, emplDate, phoneNumber, email, workedHours, moneyPerHour, type_of_position)

Staff_Cooks_Receptionists(ID_num,name, salary, bonuses, emplDate, phoneNumber, email, education,type_of_position)

Staff_Cooks_Managers(ID_num,name, salary, bonuses, emplDate, phoneNumber, email,type_of_position)

Staff_Manager_Attendants_Receptionists_Cooks (ID_num, name, salary, bonuses, emplDate, phoneNumber, e-mail, workedHours, moneyPerHour, education, type_of_position)

Остават само четири:

Staff_Manager (ID_num, name, salary, bonuses, emplDate, phoneNumber, e-mail)

Staff_Attendants (ID_num, name, salary, bonuses, emplDate, phoneNumber, e-mail, workedHours, moneyPerHour)

Staff_Receptionists (ID_num, name, salary, bonuses, emplDate, phoneNumber, e-mail, education)

Staff_Cooks(ID_Num, salary, bonuses, emplDate, phoneNumber, e-mail,type_of_position)

След оптимизация на отношенията takesCareOf, cleanIN и cooksIn цялостната релационна схема изглежда по следния начин:

Rooms (roomNumber, pricePerNight, roomType, attendantID)

Guests (EGN, name, age, phoneNumber)

Registrations (ID_num, payment, accomodationDate, days, leavingDate, roomNumber, guestEGN, guest2EGN , guest, EGN, receptionistID)

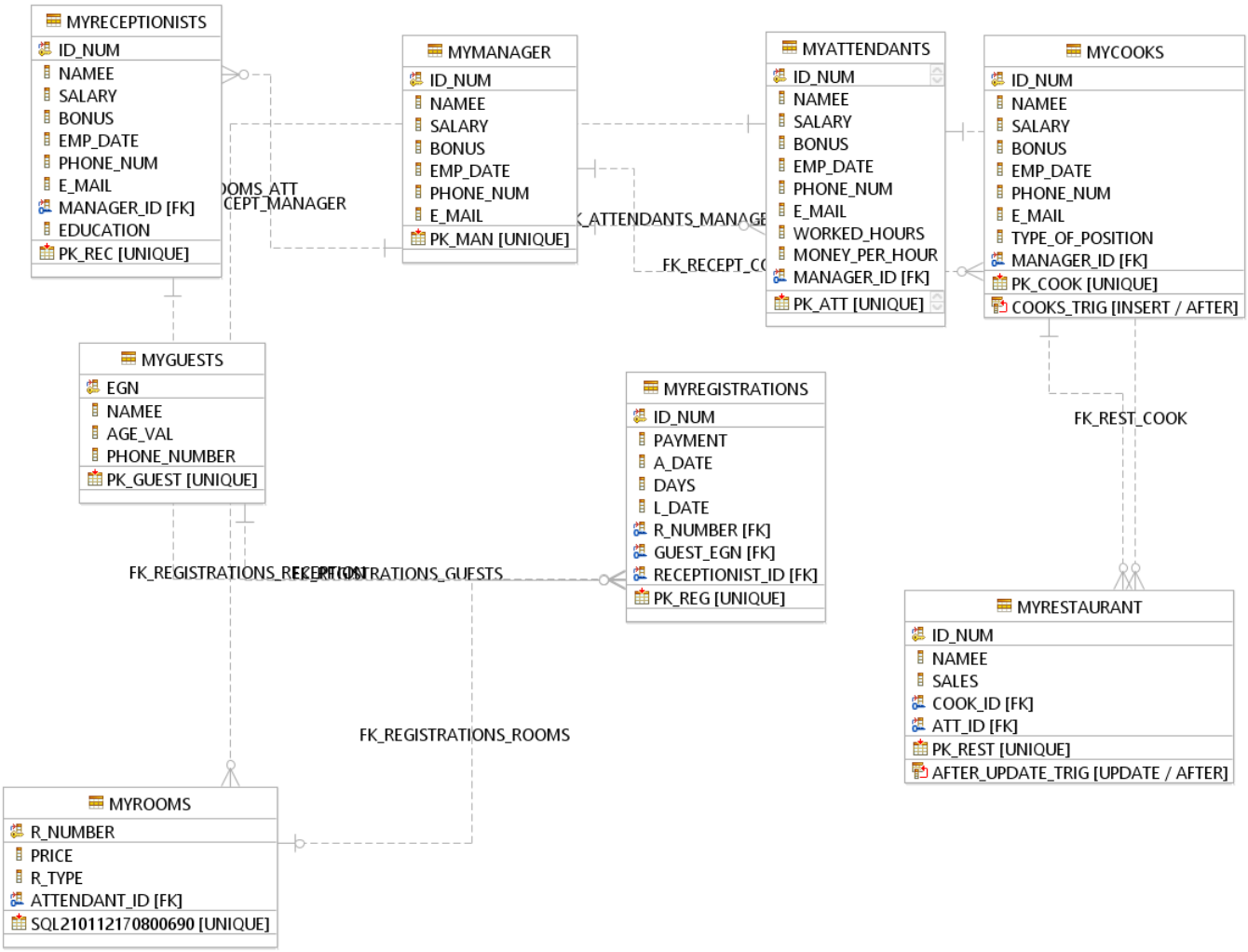
Manager (ID_num, name, salary, bonus, emplDate, phoneNumber, e-mail)

Cooks(ID_num, name, salary, bonus, emplDate, phoneNumber, e-mail, type_of_position, managerID)

Receptionists (ID_num, name, salary, bonus, emplDate, phoneNumber, e-mail, education, managerID)

Restaurant(ID_num, name, sales, attendantID, cooksID)

Attendants (ID_num, name, salary, bonuses, emplDate, phoneNumber, e-mail, workedHours, moneyPerHour, managerID)



Създаване на таблиците и вмъкване на данните

```
INSERT INTO MYGUESTS VALUES ('9909100070', 'Ivan Ivanov', 22, '0887653498')@
INSERT INTO MYGUESTS VALUES ('9105120009', 'Simona Georgieva', 30, '0896754534')@
INSERT INTO MYGUESTS VALUES ('9101010065', 'John Calvin', 30, '0887653498')@
INSERT INTO MYGUESTS VALUES ('9909040088', 'Maria Nicolaeva', 22, '0887653498')@
INSERT INTO MYGUESTS VALUES ('9909100070', 'Lilly Trifanova', 22, '0887653498')@
INSERT INTO MYGUESTS VALUES ('8109104477', 'David Simeonov', 40, '0887653498')@
INSERT INTO MYGUESTS VALUES ('9309109080', 'Sara Johnson', 28, '0887653498')@
INSERT INTO MYGUESTS VALUES ('9409100070', 'Ivet Georgieva', 27, '0887653498')@
INSERT INTO MYGUESTS VALUES ('8109100070', 'Ivailo Kostadinov', 40, '0887653498')@
INSERT INTO MYGUESTS VALUES ('8109010087', 'Stefen Dimitrov', 40, '0887653498')@

INSERT INTO MYATTENDANTS VALUES ('ATT333','Kate Williams', 700, 100, '2019-01-01',
'0878908090', 'k.wl@gmail.com','8',700,'MNG101')@
INSERT INTO MYATTENDANTS VALUES('ATT133','Daria Hristova', 700, 100, '2019-02-01',
'0886458901', 'd.hr@gmail.com','8',700,'MNG102')@
INSERT INTO MYATTENDANTS VALUES('ATT233','Elena Pavlova', 700, 400, '2019-03-01',
'0888567686', 'e.pl@gmail.com','8',700,'MNG103')@
INSERT INTO MYATTENDANTS VALUES('ATT433','Qna Mitova', 700, 300, '2019-03-01',
'0878996677', 'q.mt@gmail.com','8',700,'MNG104')@


INSERT INTO MYCOOKS VALUES ('COO101','Sofia Borisova', 1000, 50, '2019-10-03',
'0878790989', 's.b@gmail.com','assistant','MNG102')@
INSERT INTO MYCOOKS VALUES ('COO102','Ivan Manchev', 1500, 50, '2019-09-05',
'0893456556', 'i.man@gmail.com', 'chef','MNG102')@
INSERT INTO MYCOOKS VALUES ('COO103','Viktor Angelov', 1550, 50, '2019-01-08',
'0878090909', 'v.ang@gmail.com', 'chef','MNG103')@
INSERT INTO MYCOOKS VALUES ('COO104','Milena Chobanova', 1100, 50, '2019-11-01',
'0897654512', 'v.ch@gmail.com', 'assistant', 'MNG104')@

INSERT INTO MYRECEPTIONISTS VALUES ('REC101','Tanya Hristova', 800, 110, '2020-01-
07', '0873232312', 't.hr@gmail.com','MNG105', 'A')@
INSERT INTO MYRECEPTIONISTS VALUES ('REC102','Hristina Hristova', 830, 200, '2018-
08-06', '0899121113', 'hr.hr@gmail.com','MNG101', 'B')@
INSERT INTO MYRECEPTIONISTS VALUES ('REC103','Kalin Spasov', 900, 300, '2017-07-
06', '0878434244', 'k.spas@gmail.com','MNG105','B')@
INSERT INTO MYRECEPTIONISTS VALUES ('REC104','George Stefanov', 700, 100, '2019-
06-03', '0887674532', 'g.steff@gmail.com','MNG102', 'C')@


INSERT INTO MYMANAGER VALUES ('MNG101','Gergana Dimitrova', 1900, 100, '2019-01-
01', '0878908090', 'g.d@gmail.com')@
INSERT INTO MYMANAGER VALUES ('MNG102','Marina Kostova', 2000, 150, '2019-01-01',
'0878908090', 'm.k@gmail.com')@
INSERT INTO MYMANAGER VALUES ('MNG103','Kostadin Kostadinov', 2300, 200, '2019-01-
01', '0878908090', 'k.k@gmail.com')@
INSERT INTO MYMANAGER VALUES ('MNG104','Sally Johnson', 2200, 130, '2019-01-01',
'0878908090', 's.j@gmail.com')@
INSERT INTO MYMANAGER VALUES ('MNG105','Grigor Krystev', 1500, 160, '2019-01-01',
'0878908090', 'g.k@gmail.com')@


INSERT INTO MYREGISTRATIONS(PAYMENT,A_DATE, DAYS,L_DATE, R_NUMBER, GUEST_EGN,
RECEPTIONIST_ID)
VALUES (1,'2020-12-09',3 ,'2020-12-12', 305, '9909100070', 'REC101' )@
```

```
INSERT INTO MYREGISTRATIONS (PAYMENT,A_DATE, DAYS,L_DATE, R_NUMBER, GUEST_EGN,
RECEPTIONIST_ID)
VALUES(1,'2019-12-03', 2 ,'2020-12-05', 205, '9105120009',  'REC103' )@
INSERT INTO MYREGISTRATIONS (PAYMENT,A_DATE, DAYS,L_DATE, R_NUMBER, GUEST_EGN,
RECEPTIONIST_ID)
VALUES(1,'2019-12-03', 2,'2020-12-05', 105, '9101010065',  'REC104' )@
INSERT INTO MYREGISTRATIONS (PAYMENT,A_DATE, DAYS,L_DATE, R_NUMBER, GUEST_EGN,
RECEPTIONIST_ID)
VALUES(1,'2019-12-02', 7,'2020-12-09', 306, '9909040088',  'REC101' )@
INSERT INTO MYREGISTRATIONS(PAYMENT,A_DATE, DAYS,L_DATE, R_NUMBER, GUEST_EGN,
RECEPTIONIST_ID)
VALUES(1,'2020-12-07',7 ,'2020-12-09', 306, '9909100070',  'REC102' )@
INSERT INTO MYREGISTRATIONS (PAYMENT,A_DATE, DAYS,L_DATE, R_NUMBER, GUEST_EGN,
RECEPTIONIST_ID)
VALUES(1,'2015-12-02',7 ,'2015-12-09', 306, '8109104477',  'REC101' )@
INSERT INTO MYREGISTRATIONS (PAYMENT,A_DATE, DAYS,L_DATE, R_NUMBER, GUEST_EGN,
RECEPTIONIST_ID)
VALUES(1,'2016-12-02',7 ,'2016-12-09', 306, '9309109080',  'REC102' )@
INSERT INTO MYREGISTRATIONS (PAYMENT,A_DATE, DAYS,L_DATE, R_NUMBER, GUEST_EGN,
RECEPTIONIST_ID)
VALUES(1,'2016-12-02', 7,'2016-12-09', 306, '9409100070',  'REC104' )@


INSERT INTO MYROOMS VALUES (101,60, 1,'ATT333')@
INSERT INTO MYROOMS VALUES (102,60,1,'ATT433')@
INSERT INTO MYROOMS VALUES (103,60,1,'ATT133')@
INSERT INTO MYROOMS VALUES (104,60,1,'ATT233')@
INSERT INTO MYROOMS VALUES (105,60,1,'ATT333')@
INSERT INTO MYROOMS VALUES (106,60,1,'ATT433')@
INSERT INTO MYROOMS VALUES (201,80,2,'ATT133')@
INSERT INTO MYROOMS VALUES (202,80,2,'ATT133')@
INSERT INTO MYROOMS VALUES (203,80,2,'ATT433')@
INSERT INTO MYROOMS VALUES (204,80,2,'ATT433')@
INSERT INTO MYROOMS VALUES (205,80,2,'ATT233')@
INSERT INTO MYROOMS VALUES (206,80,2,'ATT233')@
INSERT INTO MYROOMS VALUES (301,100,3,'ATT333')@
INSERT INTO MYROOMS VALUES (302,100,3,'ATT433')@
INSERT INTO MYROOMS VALUES (303,100,3,'ATT333')@
INSERT INTO MYROOMS VALUES (304,100,3,'ATT133')@
INSERT INTO MYROOMS VALUES (305,100,3,'ATT133')@
INSERT INTO MYROOMS VALUES (306,100,3,'ATT333')@


INSERT INTO MYRESTAURANT VALUES ('01', 'Night club', 'Wine', 'COO101','ATT133')@
INSERT INTO MYRESTAURANT VALUES ('02', 'Restaurant', 'None', 'COO102','ATT333' )@
INSERT INTO MYRESTAURANT VALUES ('03', 'Cafeteria', 'None',  'COO103','ATT133')@
```

Описание на функциите

- Скаларна функция GET_SALES_RESTAURANT_FUNC

```
CREATE FUNCTION GET_SALES_RESTAURANT_FUNC(VAL_ID CHAR(6))
    RETURNS CHAR(20)
    RETURN
      SELECT SALES
      FROM MYRESTAURANT
```

```
      WHERE ID_NUM = VAL_ID@

   --CALL FUNCTION
    VALUES FN71840.GET_SALES_RESTAURANT_FUNC('01')@

    SELECT FN71840.GET_SALES_RESTAURANT_FUNC(ID_NUM) AS IDENTIFICATION
            FROM MYRESTAURANT@
```

Въпросната функция приема подаден номер на ресторант и има идеята да върне съответно промоциите, които има съответния ресторант.
Извикваме функцията като подаваме номер, както е в случая '01' и след извикването й трябва да ни изведе 'Wine'.

- Таблична функция

```
CREATE FUNCTION ATTENDANTS_FUNC1(VAL_ID CHAR(6))
 RETURNS TABLE (V_NAMEE CHAR(50), V_PHONE_NUM CHAR(10), V_E_MAIL VARCHAR(30))
 RETURN
      SELECT NAMEE, PHONE_NUM, E_MAIL
      FROM MYATTENDANTS
      WHERE ID_NUM = VAL_ID@

SELECT *
         FROM TABLE (FN71840.ATTENDANTS_FUNC1('ATT333')) V@
```

Функцията приема номер на камериер от таблицата MyAttendants

И целта й е да изведе таблица с името му, телефонният му номер, имейла му по този подаден номер, както е в нашия случай 'ATT333' и в нашия случай резултатът след извикването ще бъде

```
V_NAMEE                                            V_PHONE_NUM V_E_MAIL
 -------------------------------------------------- ----------- -----------
---
          Kate Williams                                         0878908090
         k.wl@gmail.com
```

Описание на тригерите

- AFTER_UPDATE_TRIG1

```
CREATE TRIGGER AFTER_UPDATE_TRIG1
   AFTER UPDATE OF sales ON  MYRESTAURANT
   referencing OLD AS O NEW AS N
   FOR EACH ROW
         WHEN(O.ID_NUM != N.ID_NUM)
         UPDATE MYRESTAURANT
         SET ATT_ID  = 'ATT433'@


UPDATE MYRESTAURANT
   SET SALES = 'Fruit juice'
   WHERE ID_NUM = '01'@
```

Този тригер се нарича after тригер и се задейства след като сме подновили, тоест сме добавили, както е в нашия случай промоция на ресторант с ID_NUM = '01', като след промяната на промоцията в съответния ресторант си създаме тригера и номера на ATT_ID се променя на 'ATT433'.

- dataFromCooks

```
CREATE PROCEDURE dataFromCooks(IN v_id CHAR(6))
DYNAMIC RESULT SETS 1
P1: BEGIN
DECLARE c1 CURSOR WITH RETURN FOR SELECT * FROM MYCOOKS WHERE
ID_NUM = v_id;
OPEN c1;
END P1@

CALL FN71840.dataFromCooks('COO101')@

CREATE TRIGGER COOKS_TRIG
   AFTER INSERT ON MYCOOKS
   REFERENCING NEW AS N
   FOR EACH ROW
   BEGIN ATOMIC
         UPDATE MYCOOKS
         SET TYPE_OF_POSITION = 'Chef';
         CALL FN71840.dataFromCooks('COO105');
         END@

 INSERT INTO
MYCOOKS(ID_NUM,NAMEE,SALARY,BONUS,EMP_DATE,PHONE_NUM,E_MAIL,TYPE_O
F_POSITION,MANAGER_ID)
 VALUES('COO105','Toni Bakalov', 1200, 100, '2020-01-04',
'0889080866', 'toni.bak@gmail.com','assistant','MNG102')@
```

Тригерът dataFromCooks, за да бъде задействан, първо трябва да бъде създадена процедура, тъй като този процедура е вградена в него и следва да бъде извикана.
Самата процедура dataFromCooks приема дадено id на готвач и следва да изведе множество от всички данни на готвача, чийто id сме подали.

След създаване на процедурата се прехвърляме към тригера, но за да се задейства е необходимо да сме добавили нов готвач във нашата таблица, като му въведем нови данни. След което при задействане н тригера в неговото тяло ще извикаме процедурата, подайки и id-то на новия служител и като резултат ще видим множеството същности, тоест всички данни, които преди малко сме добавили на служителя, освен това другото, което ще се update-не с тригера е позията, която ще бъде 'Chef'.

Описание на изгледите

- ATTENDENTS_CLEAN_ROOMS_VIEW

```
CREATE VIEW ATTENDENTS_CLEAN_ROOMS_VIEW
AS
      SELECT V2.ID_NUM, V2.NAMEE,V2.SALARY ,V1.R_NUMBER
      FROM MYROOMS V1, MYATTENDANTS V2
      WHERE  V2.ID_NUM = V1.ATTENDANT_ID AND V2.SALARY > 300@
--CALL 1 VIEW
SELECT * FROM ATTENDENTS_CLEAN_ROOMS_VIEW@
```

Следният изглед свързва две таблици MYROOMS , MYATTENDANTS, като взема идентификационният номер на дадени камериери, името му, заплатата им и ги извежда само на тези, които получават по-голяма заплата от 300.

- GUEST_STAY_VIEW

```
CREATE VIEW GUEST_STAY_VIEW
AS
      SELECT V1.NAMEE, V1.EGN, V1.AGE_VAL, V2.R_NUMBER
      FROM MYGUESTS V1, MYREGISTRATIONS V2
      WHERE  V1.EGN = V2.GUEST_EGN @

--CALL 2 VIEW
SELECT* FROM GUEST_STAY_VIEW@
```

При този изглед отново свързва две таблици MYGUESTS и MYREGISTRATIONS, които са свързани чрез егн-то на госта и след извикване на изгледа, трябва да ни изкара името, егн и възраст на служителие, които на гостите, които са геистрирани в ххотела(тоест от таблицата MYREGISTRATIONS).

- MYREGISTRATIONS_INFO

```
CREATE VIEW MYREGISTRATIONS_INFO
AS
   SELECT *
   FROM MYREGISTRATIONS
   WHERE RECEPTIONIST_ID = 'REC101' AND GUEST_EGN = '8%'@

SELECT * FROM  MYREGISTRATIONS_INFO@
```

Последният изглед просто се вземат всички данни от таблицата на тезо служител, чийто номер е 'REC101' и в конкретния случай и егн-то му започва с “8”.

Описание на процедурите

- процедура с курсор и входни и изходни параметри

```
CREATE procedure calculateBonusSalaryRandomNumProc(IN randomNum INTEGER, OUT res
DOUBLE)
 RESULT SETS 1
 BEGIN
 DECLARE counter double;
       FOR v1 AS c1 CURSOR FOR SELECT id_num, salary, bonus FROM MYRECEPTIONISTS
              DO
              IF (ID_NUM = 'REC101') THEN
                     SET counter = salary + randomNum + bonus;
              ELSE
              SET counter = (salary + bonus) - randomNum;
              END IF;
       END FOR;
       SET res = counter;
END@

CALL FN71840.calculateBonusSalaryRandomNumProc(100,?)@
```

Следната процедура приема като входен параметър някакво число, което след това в един цикъл чрез проверка ще искаме да го използваме, като го добавим на конкретен номер ID_NUM = 'REC101' на рецепционист и на всички останали като извадим това число и след това просто ще го изведем.

- процедура с прихващане на изключение

```
CREATE PROCEDURE RAISE_ERROR( ID_Num char(2), OUT O_SQLSTATE CHAR(5), OUT
O_SQLCODE INTEGER)
LANGUAGE SQL
SPECIFIC changeSalesProc
BEGIN
DECLARE SQLCODE INTEGER DEFAULT 0;
DECLARE SQLSTATE CHAR (5) DEFAULT '00000';
DECLARE EXIT HANDLER FOR SQLEXCEPTION SELECT SQLCODE, SQLSTATE INTO O_SQLCODE,
O_SQLSTATE FROM SYSIBM.SYSDUMMY1;

UPDATE MYRESTAURANT
       SET sales = (CASE WHEN namee = 'Night club' THEN 'Champagne'
                                    WHEN namee = 'Restaurant' THEN  'Salad Cesar'
else RAISE_ERROR('70001', 'Incorrect!') END); --СЪОБЩЕНИЕ ЗА ГРЕШКА

END@
```

При следната процедура приемаме номер на ресторант от таблицата и искаме да променим промоциите в зависимост от името. Съответно че изкара съобщение за грешка, ако възникне такава, без значение каква е тя.

- Втора процедура с прихващане на изключение

```
CREATE PROCEDURE handlingNotFound11(out res_payment INT,out res_recept_id char(6))
LANGUAGE SQL
BEGIN
DECLARE flag INTEGER DEFAULT 0;
DECLARE N INTEGER DEFAULT 0;
DECLARE v_id_num INT;
DECLARE v_payment INT;
DECLARE v_r_number INT;
DECLARE v_receptionist_id char(6) DEFAULT ' ';
DECLARE c1 CURSOR FOR select  ID_NUM, PAYMENT, R_NUMBER,RECEPTIONIST_ID from
MYREGISTRATIONS;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET flag = 1;
OPEN c1;

WHILE(flag = 1) DO
FETCH c1 INTO v_id_num, v_payment, v_r_number,v_receptionist_id;
      IF (v_payment = 1 and v_receptionist_id = 'REC101' )
THEN  SET v_payment = 0;
        SET v_receptionist_id = 'REC104' ;
   else
       SET v_payment = 2;
       SET v_receptionist_id = 'REC104';
END IF;
END WHILE;

set res_payment = v_payment;
set res_recept_id = v_receptionist_id;

CLOSE c1;
END@

CALL FN71840.handlingNotFound11(?,?)@
```

Следната процедура съдържа цикъл, който минава през всички редове на таблицата и има идеята да промени там, където номер на рецепциониста, както е в случая 'REC101' и да промени заплащането да бъде с карта или по данков път, тоест v_payment = 0, както и ще се ромени номера на рецепциониста. В цикъла искаме да направим проверка чрез CONTINUE HANDLER FOR NOT FOUND, тоест, когато стигнем до края на курсора да а подаде грешка, но тъй като е continue, той директно ще мине на следващия statement и не би трябало да създаде проблем. Накрая процедурата ще върне съответно номера на рецепциониста и заплащането в карта или по банков път.

- процедура с курсор и while цикъл

```
CREATE PROCEDURE changeSalaryOfCookProcedure(IN ID_VALUE ANCHOR MYCOOKS.ID_NUM,
OUT new_salary DOUBLE)
LANGUAGE SQL
SPECIFIC changeSalaryOfCookProcedure
DYNAMIC RESULT SETS 1
BEGIN
DECLARE v_salary DOUBLE;
DECLARE v_name CHAR(50);
DECLARE v_TYPE_OF_POSITION CHAR(10) DEFAULT ' ';
DECLARE flag integer default 0;

DECLARE c1 CURSOR FOR SELECT NAMEE, SALARY, TYPE_OF_POSITION FROM MYCOOKS;
OPEN c1;
FETCH c1 INTO v_name, v_salary, v_TYPE_OF_POSITION ;
set flag = 0;
WHILE (flag = 1) DO
 IF (v_TYPE_OF_POSITION = 'assistant' AND v_name = 'S%')
       THEN SET v_salary = v_salary - 100;
              set flag = 1;
       ELSE SET v_salary = v_salary + 100;
              set flag = 1;
  END IF;
END WHILE;
 set new_salary=v_salary;
CLOSE c1;
END@

CALL FN71840.changeSalaryOfCookProcedure('COO101', ?)@
```

Процедурата приема като параметър идентификационен номер на служител готвач и връща неговата нова променена заплата. Ако позицията е асистент и името му започва със ‚S‘, тогава ще се промени заплатата му, като я намали със 100. Ако не отговаря на условието ще влезе във вторият statement тогава ще се промени заплатата му, като я увеличи със 100.