

Информационна система за управление на резервации в хотел

Изготвили:

Кристина Киримова 62246,
Силвия Стоянова 62365 ,
Димитър Колев 62295,
Ивана Тонева 62337

Таблица с разпределение на задачите

Задача	Фак. №
1. Създаване на схемите и релациите	62295
2. Добавяне на съдържание	62337
3. Примерни прости заявки и заявки върху две и повече релации	62295, 62337
4. Примери с подзаявки	62295
5. Примери със съединения	62337
6. Примери с групиране и агрегация	62365
7. Примери с ограничения	62365
8. Примери с изгледи и индекси	62246
9. Примери с тригери	62246
10. Оформяне на документацията на проекта и създаване на презентация за представянето му	62295,62337, 62365,62246

Съдържание

1. Описание на предмета и областта на проекта.....	3
2. Дефиниране на схемата на релациите.....	4
2.1. Създаване на релационни таблици.....	4
2.2. Схемата на релациите.....	5
3. Добавяне на примерно съдържание.....	6
4. Примерни заявки.....	11
4.1. Прости заявки.....	11
4.2. Заявки върху две и повече релации.....	13
4.3. Подзаявки.....	14
4.4. Съединения.....	16
4.5. Групиране и агрегации.....	17
5. Подобрения.....	20
5.1. Ограничения.....	20
5.2. Тригери.....	22
5.3. Изгледи и индекси.....	24

Описание на предмета и областта на проекта

Информационна система за управление на резервации в хотел

Основното предназначение на базата данни е менажирането на резервациите в хотелски комплекс, което ще помогне на служителите от отдели „Резервации“ и „Администрация“ да изпълняват безпроблемно ежедневните си задачи. Разработена е с идеята да бъде интегрирана към система с по-широк обхват, даваща възможност на служителите, които я използват, да съхраняват и манипулират данни както за клиентските резервации, така и за управление на персонала и други компоненти от хотелския мениджмънт.

Основни представени възможности на базата данни:

1. Съхранение и манипулиране на информация за наличните стаи

Базата данни пази информация за всички налични стаи в хотела. Въведени са техните основни характеристики, които позволяват на служителите да виждат ясно данните, които са им нужни, за да могат да създават нови резервации, да преценяват бързо и лесно къде да настанят нов клиент, да следят заетостта на хора и други.

2. Съхранение и манипулиране на информация за направените клиентски резервации

Базата данни пази информация за всяка направена клиентска резервация. Позволява лесно следене на предишни, настоящи и бъдещи резервации, както и записване на нови. Чрез създадените ограничения се цели избягването на грешки като дублирани резервации, неясна информация за това каква част от цената за коя услуга отговаря и други.

3. Съхранение и манипулиране на информация за клиентите

Базата данни пази информация за всеки клиент, направил резервация в хотела. Така служителите мога да съхраняват основните данни за всеки, живущ с сградата за определен период. Това е важна част от проекта, тъй като всеки хотел е длъжен да пази информация за гостите си, основно поради съображения за сигурност.

4. Съхранение и манипулиране на информация за различните предлагани пакети

Базата данни пази информация за предлаганите пакети изхранвания в хотела. Както следва, те са RO - стая без изхранване, BB – стая със закуска, HB – стая със закуска и вечеря, FB – стая със закуска, обяд и вечеря и AI – стая с всичко включено. Има възможност да се добавят още отличителни характеристики на всеки пакет според различни фактори като например – празници, събития, извънредни ситуации или сезон.

5. Съхранение и манипулиране на информация за служителите, заемащи хотелски стаи

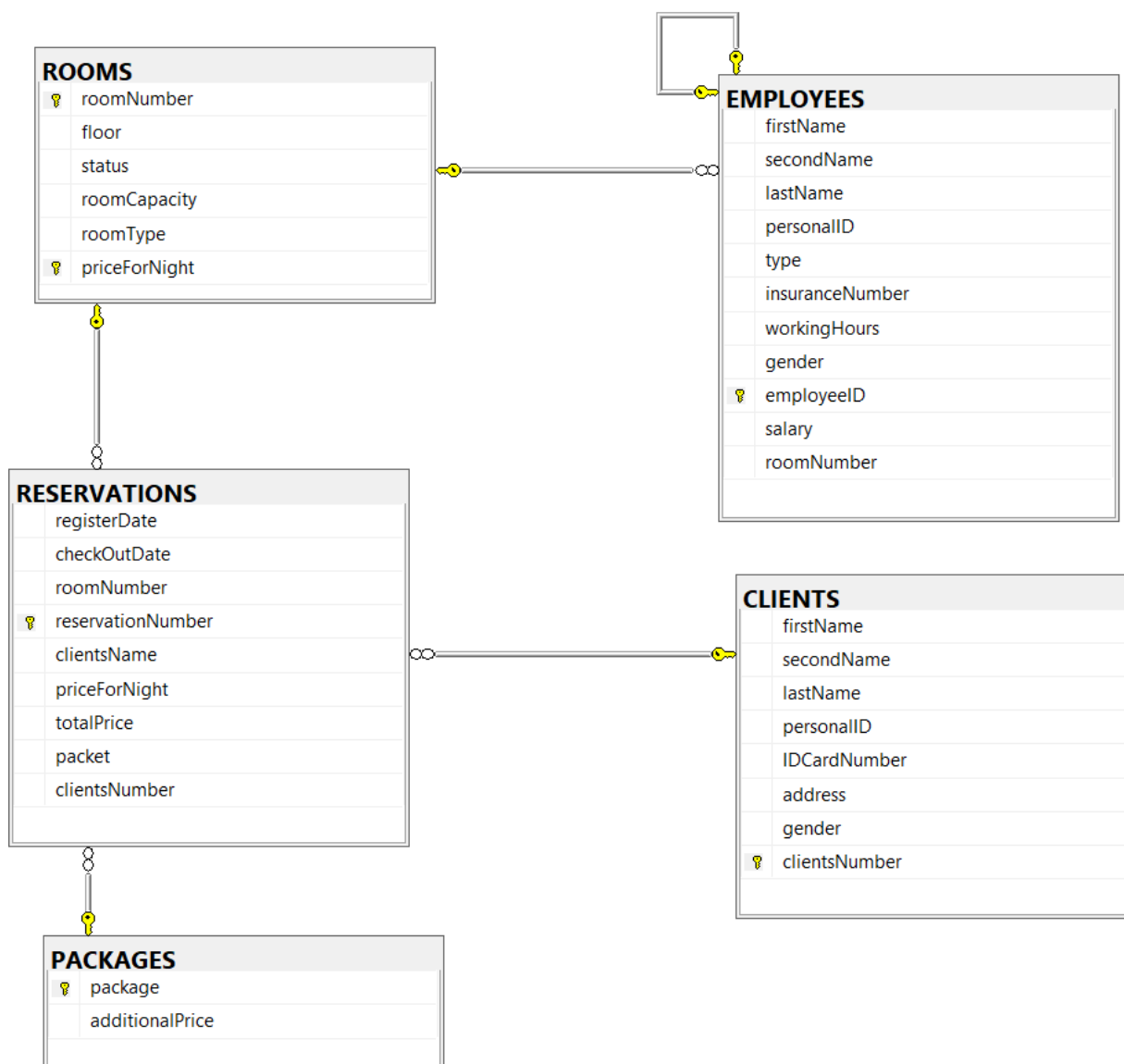
Има множество хотели, предлагащи настаняване на служителите си. За тази цел базата данни дава възможност да се следят работниците, настанени в хотела. Съдържа се информация за всички работници като въпросната таблица може да се използва и от други отдели на хотела като например - управление на служители или финанси.

Дефиниране на схемата на релациите

2.1. Създаване на релационни таблици:

```
CREATE TABLE ROOMS(  
    roomNumber INT,  
    floor INT,  
    status CHAR(1) DEFAULT 'F',  
    roomCapacity INT,  
    roomType VARCHAR(10),  
    priceForNight DECIMAL(8,2)  
);  
  
CREATE TABLE RESERVATIONS(  
    registerDate DATE,  
    checkOutDate DATE,  
    roomNumber INT,  
    reservationNumber INT IDENTITY(1,1),  
    clientsName VARCHAR(60),  
    priceForNight DECIMAL(8,2),  
    totalPrice AS (priceForNight * DATEDIFF(DAY,registerDate,checkOutDate)) PERSISTED,  
    clientsNumber INT,  
    packet CHAR(2)  
);  
  
CREATE TABLE CLIENTS(  
    firstName VARCHAR(20),  
    secondName VARCHAR(20),  
    lastName VARCHAR(20),  
    personalID CHAR(10),  
    IDCardNumber CHAR(9),  
    address VARCHAR(100),  
    gender CHAR(1),  
    clientsNumber INT IDENTITY(1,1)  
);  
  
CREATE TABLE EMPLOYEES(  
    firstName VARCHAR(20),  
    secondName VARCHAR(20),  
    lastName VARCHAR(20),  
    personalID CHAR(10),  
    type VARCHAR(15),  
    insuranceNumber VARCHAR(20),  
    workingHours INT,  
    gender CHAR(1),  
    employeeID INT IDENTITY(1,1),  
    salary DECIMAL(10,2),  
    roomNumber INT  
);  
  
CREATE TABLE PACKAGES(  
    package CHAR(2),  
    additionalPrice INT  
);
```

2.2.Схема на релациите:



	roomNumber	floor	status	roomCapacity	roomType	priceForNight		roomNumber	floor	status	roomCapacity	roomType	priceForNight
1	101	1	F	1	Room	75.00	51	301	3	F	3	Room	110.00
2	102	1	F	1	Room	75.00	52	302	3	F	3	Room	110.00
3	103	1	F	1	Room	75.00	53	303	3	F	3	Room	110.00
4	104	1	F	1	Room	75.00	54	304	3	F	3	Room	110.00
5	105	1	F	1	Room	75.00	55	305	3	F	3	Room	110.00
6	106	1	F	1	Room	75.00	56	306	3	F	3	Room	110.00
7	107	1	F	1	Room	75.00	57	307	3	F	3	Room	110.00
8	108	1	F	1	Room	75.00	58	308	3	F	3	Room	110.00
9	109	1	F	1	Room	75.00	59	309	3	F	3	Room	110.00
10	110	1	F	1	Room	75.00	60	310	3	F	3	Room	110.00
11	111	1	F	1	Room	75.00	61	311	3	F	3	Room	110.00
12	112	1	F	1	Room	75.00	62	312	3	F	3	Room	110.00
13	113	1	F	1	Room	75.00	63	313	3	F	3	Room	110.00
14	114	1	F	2	Room	90.00	64	314	3	F	3	Room	110.00
15	115	1	F	2	Room	90.00	65	315	3	F	3	Room	110.00
16	116	1	F	2	Room	90.00	66	316	3	F	3	Room	110.00
17	117	1	F	2	Room	90.00	67	317	3	F	3	Room	110.00
18	118	1	F	2	Room	90.00	68	318	3	F	3	Room	110.00
19	119	1	F	2	Room	90.00	69	319	3	F	3	Room	110.00
20	120	1	S	2	Room	90.00	70	320	3	F	3	Room	110.00
21	121	1	S	2	Room	90.00	71	321	3	F	3	Room	110.00
22	122	1	S	2	Room	90.00	72	322	3	F	3	Room	110.00
23	123	1	S	2	Room	90.00	73	323	3	F	3	Room	110.00
24	124	1	S	2	Room	90.00	74	324	3	F	3	Room	110.00
25	125	1	S	2	Room	90.00	75	325	3	F	3	Room	110.00
26	201	2	F	2	Family	100.00	76	401	4	F	3	Family	120.00
27	202	2	F	2	Family	100.00	77	402	4	F	3	Family	120.00
28	203	2	F	2	Family	100.00	78	403	4	F	3	Family	120.00
29	204	2	F	2	Family	100.00	79	404	4	F	3	Family	120.00
30	205	2	F	2	Family	100.00	80	405	4	F	3	Family	120.00
31	206	2	F	2	Family	100.00	81	406	4	F	3	Family	120.00
32	207	2	F	2	Family	100.00	82	407	4	F	3	Family	120.00
33	208	2	F	2	Family	100.00	83	408	4	F	3	Family	120.00
34	209	2	F	2	Family	100.00	84	409	4	F	3	Family	120.00
35	210	2	F	2	Family	100.00	85	410	4	F	3	Family	120.00
36	211	2	F	2	Family	100.00	86	411	4	F	3	Family	120.00
37	212	2	F	2	Family	100.00	87	412	4	F	3	Family	120.00
38	213	2	F	2	Family	100.00	88	413	4	F	3	Family	120.00
39	214	2	F	2	Family	100.00	89	414	4	F	3	Family	120.00
40	215	2	F	2	Family	100.00	90	415	4	F	3	Family	120.00
41	216	2	F	2	Family	100.00	91	416	4	F	3	Family	120.00
42	217	2	F	2	Family	100.00	92	417	4	F	3	Family	120.00
43	218	2	F	2	Family	100.00	93	418	4	F	3	Family	120.00
44	219	2	F	2	Family	100.00	94	419	4	F	3	Family	120.00
45	220	2	F	2	Family	100.00	95	420	4	F	3	Family	120.00
46	221	2	F	2	Family	100.00	96	421	4	F	3	Family	120.00
47	222	2	F	2	Family	100.00	97	422	4	F	3	Family	120.00
48	223	2	F	2	Family	100.00	98	423	4	F	3	Family	120.00
49	224	2	F	2	Family	100.00	99	424	4	F	3	Family	120.00
50	225	2	F	2	Family	100.00	100	425	4	F	3	Family	120.00

	roomNumber	floor	status	roomCapacity	roomType	priceForNight
101	501	5	F	5	Apartm...	180.00
102	502	5	F	5	Apartm...	180.00
103	503	5	F	5	Apartm...	180.00
104	504	5	F	5	Apartm...	180.00
105	505	5	F	5	Apartm...	180.00
106	506	5	F	5	Apartm...	180.00
107	507	5	F	5	Apartm...	180.00
108	508	5	F	5	Apartm...	180.00
109	509	5	F	5	Apartm...	180.00
110	510	5	F	5	Apartm...	180.00
111	511	5	F	5	Apartm...	180.00
112	512	5	F	5	Apartm...	180.00
113	513	5	F	5	Apartm...	180.00
114	514	5	F	5	Apartm...	180.00
115	515	5	F	5	Apartm...	180.00

Таблица за резервации

--Insert some example values in the RESERVATIONS table

```
INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsName,packet)
VALUES('2020-03-14','2020-03-16',125,'Ivana Petkova','RO')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsName,packet)
VALUES('2019-12-01','2019-12-05',404,'Hristo Velev','BB')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsName,packet)
VALUES('2019-07-28','2019-08-05',512,'Georgi Kovachev','AI')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsName,packet)
VALUES('2020-02-14','2020-02-15',103,'Petar Georgiev','BB')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsName,packet)
VALUES('2020-01-30','2020-02-01',222,'Daniela Cvetkova','HB')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsNumber,packet)
VALUES('2020-01-21','2020-01-23',320,5,'RO')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsNumber,packet)
VALUES('2020-03-10','2020-03-15',110,15,'RO')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsNumber,packet)
VALUES('2020-04-04','2020-04-10',502,1,'AI')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsNumber,packet)
VALUES('2019-09-26','2019-09-28',206,10,'HB')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsNumber,packet)
VALUES('2019-08-01','2019-08-10',414,3,'AI')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsNumber,packet)
VALUES('2020-03-06','2020-03-08',315,18,'BB')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsNumber,packet)
VALUES('2020-01-21','2020-01-23',120,2,'RO')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsName,packet)
VALUES('2019-07-28','2019-07-31',125,'Ivana Petkova','RO')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsName,packet)
VALUES('2019-07-10','2019-07-15',404,'Hristo Velev','BB')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsName,packet)
VALUES('2019-09-12','2019-09-15',512,'Georgi Kovachev','AI')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsName,packet)
VALUES('2019-07-15','2019-07-18',103,'Petar Georgiev','BB')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsName,packet)
VALUES('2019-07-01','2019-07-06',222,'Daniela Cvetkova','HB')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsNumber,packet)
VALUES('2019-09-10','2019-09-15',320,5,'RO')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsNumber,packet)
VALUES('2019-07-05','2019-07-12',110,15,'RO')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsNumber,packet)
VALUES('2019-07-03','2019-07-10',502,1,'AI')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsNumber,packet)
VALUES('2019-07-02','2019-07-06',206,10,'HB')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsNumber,packet)
VALUES('2019-07-12','2019-07-19',414,3,'AI')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsNumber,packet)
VALUES('2019-07-29','2019-08-02',315,18,'BB')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsNumber,packet)
VALUES('2019-06-29','2019-07-02',120,2,'RO')
```

```
INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsNumber,packet)
VALUES('2019-07-02','2019-07-06',206,10,'HB')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsNumber,packet)
VALUES('2019-07-12','2019-07-19',414,3,'AI')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsNumber,packet)
VALUES('2019-07-29','2019-08-02',315,18,'BB')

INSERT INTO RESERVATIONS(registerDate,checkOutDate,roomNumber,clientsNumber,packet)
VALUES('2019-06-29','2019-07-02',120,2,'RO')

UPDATE RESERVATIONS SET priceForNight =
(SELECT ROOMS.priceForNight FROM ROOMS WHERE RESERVATIONS.roomNumber = ROOMS.roomNumber) +
(SELECT PACKAGES.additionalPrice FROM PACKAGES WHERE RESERVATIONS.packet = PACKAGES.package)

UPDATE RESERVATIONS SET clientsName =
(SELECT CLIENTS.firstName + ' ' + CLIENTS.lastName FROM CLIENTS
WHERE RESERVATIONS.clientsNumber = CLIENTS.clientsNumber)
WHERE clientsNumber IS NOT NULL
```

	registerDate	checkOutDate	roomNumber	reservationNumber	clientsName	priceForNight	totalPrice	clientsNumber	packet
1	2020-03-14	2020-03-16	125	1	Ivana Petkova	90.00	180.00	NULL	RO
2	2019-12-01	2019-12-05	404	2	Hristo Velev	125.00	500.00	NULL	BB
3	2019-07-28	2019-08-05	512	3	Georgi Kovachev	200.00	1600.00	NULL	AI
4	2020-02-14	2020-02-15	103	4	Petar Georgiev	80.00	80.00	NULL	BB
5	2020-01-30	2020-02-01	222	5	Daniela Cvetkova	110.00	220.00	NULL	HB
6	2020-01-21	2020-01-23	320	6	Angelina Dimitrova	110.00	220.00	5	RO
7	2020-03-10	2020-03-15	110	7	Nikola Dimitrov	75.00	375.00	15	RO
8	2020-04-04	2020-04-10	502	8	Angel Dimitrov	200.00	1200.00	1	AI
9	2019-09-26	2019-09-28	206	9	Elena Kostova	110.00	220.00	10	HB
10	2019-08-01	2019-08-10	414	10	Victor Yanev	140.00	1260.00	3	AI
11	2020-03-06	2020-03-08	315	11	Radoslava Popo...	115.00	230.00	18	BB
12	2020-01-21	2020-01-23	120	12	Boris Tonev	90.00	180.00	2	RO
13	2019-07-28	2019-07-31	125	13	Ivana Petkova	90.00	270.00	NULL	RO
14	2019-07-10	2019-07-15	404	14	Hristo Velev	125.00	625.00	NULL	BB
15	2019-09-12	2019-09-15	512	15	Georgi Kovachev	200.00	600.00	NULL	AI
16	2019-07-15	2019-07-18	103	16	Petar Georgiev	80.00	240.00	NULL	BB
17	2019-07-01	2019-07-06	222	17	Daniela Cvetkova	110.00	550.00	NULL	HB
18	2019-09-10	2019-09-15	320	18	Angelina Dimitrova	110.00	550.00	5	RO
19	2019-07-05	2019-07-12	110	19	Nikola Dimitrov	75.00	525.00	15	RO
20	2019-07-03	2019-07-10	502	20	Angel Dimitrov	200.00	1400.00	1	AI
21	2019-07-02	2019-07-06	206	21	Elena Kostova	110.00	440.00	10	HB
22	2019-07-12	2019-07-19	414	22	Victor Yanev	140.00	980.00	3	AI
23	2019-07-29	2019-08-02	315	23	Radoslava Popo...	115.00	460.00	18	BB
24	2019-06-29	2019-07-02	120	24	Boris Tonev	90.00	270.00	2	RO

Таблица с видове пакетни услуги

--Insert some example values in the PACKAGES table

```
INSERT INTO PACKAGES VALUES ('RO',0);
INSERT INTO PACKAGES VALUES ('BB',5);
INSERT INTO PACKAGES VALUES ('HB',10);
INSERT INTO PACKAGES VALUES ('FB',15);
INSERT INTO PACKAGES VALUES ('AI',20);
```

	package	additionalPrice
1	AI	20
2	BB	5
3	FB	15
4	HB	10
5	RO	0

Таблица с клиенти

--Insert some example values in the CLIENTS table

```
INSERT INTO CLIENTS (firstName,secondName,lastName,personalID,IDCardNumber,address,gender)
VALUES ('Angel','Georgiev','Dimitrov','8706234402','731577245','Petko Voivoda str. 10 Qmbol','M');

INSERT INTO CLIENTS (firstName,secondName,lastName,personalID,IDCardNumber,address,gender)
VALUES ('Boris','Ivanov','Tonev','8004037820','254959893','Hristo Botev str. 25 Sliven','M');

INSERT INTO CLIENTS (firstName,secondName,lastName,personalID,IDCardNumber,address,gender)
VALUES ('Victor','Asenov','Yanev','9005163015','323967738','Ivan Vazov str. 56 Sofia','M');

INSERT INTO CLIENTS (firstName,secondName,lastName,personalID,IDCardNumber,address,gender)
VALUES ('Georgi','Borisov','Georgiev','7711065549','427857598','Hristo Botev str. 5 Smolqn','M');

INSERT INTO CLIENTS (firstName,secondName,lastName,personalID,IDCardNumber,address,gender)
VALUES ('Angelina','Ivanova','Dimitrova','7901157751','224062708','Georgi Rakovski bul. 10 Haskovo','F');

INSERT INTO CLIENTS (firstName,secondName,lastName,personalID,IDCardNumber,address,gender)
VALUES ('Borqna','Nikolova','Ivanova','8403123112','828334620','Petko R. Slaveikov str. 15 Blagoevgrad','F');

INSERT INTO CLIENTS (firstName,secondName,lastName,personalID,IDCardNumber,address,gender)
VALUES ('Vasilena','Pavlova','Goranova','9712083802','181320927','Hristo Smirnenski str. 3 Gorna Orqhovica','F');

INSERT INTO CLIENTS (firstName,secondName,lastName,personalID,IDCardNumber,address,gender)
VALUES ('Gergana','Atanasova','Yankova','7005193817','644724170','Ivan Vazov str. 18 Silistra','F');

INSERT INTO CLIENTS (firstName,secondName,lastName,personalID,IDCardNumber,address,gender)
VALUES ('Dimo','Venzislavov','Yordanov','8806203871','341831589','Hristo Botev str. 12 Sofia','M');

INSERT INTO CLIENTS (firstName,secondName,lastName,personalID,IDCardNumber,address,gender)
VALUES ('Elena','Boqnova','Kostova','6110022584','714676093','Geo Milev str. 40 Sofia','F');

INSERT INTO CLIENTS (firstName,secondName,lastName,personalID,IDCardNumber,address,gender)
VALUES ('Zdravko','Atanasov','Popyordanov','6508153210','796177560','Atanas Ishirkov str. 12','M');

INSERT INTO CLIENTS (firstName,secondName,lastName,personalID,IDCardNumber,address,gender)
VALUES ('Ivanka','Nikolaeva','Apostolova','7003034401','879017750','Vasil Levski str. 13 Pernik','F');

INSERT INTO CLIENTS (firstName,secondName,lastName,personalID,IDCardNumber,address,gender)
VALUES ('Kamen','Georgiev','Todorov','7904298031','515448687','Luben Karavelov str. 5 Qmbol','M');

INSERT INTO CLIENTS (firstName,secondName,lastName,personalID,IDCardNumber,address,gender)
VALUES ('Mariana','Stefanova','Popova','9411112115','773245521','Georgi Rakovski str. 55 Burgas','F');

INSERT INTO CLIENTS (firstName,secondName,lastName,personalID,IDCardNumber,address,gender)
VALUES ('Nikola','Ivanov','Dimitrov','8908139150','386316542','Nikola Vapcarov str. 16 Varna','M');

INSERT INTO CLIENTS (firstName,secondName,lastName,personalID,IDCardNumber,address,gender)
VALUES ('Ognqn','Nikolov','Hristov','5802147104','263262817','6-ti Septemvri str. 15 Kazanluk','M');

INSERT INTO CLIENTS (firstName,secondName,lastName,personalID,IDCardNumber,address,gender)
VALUES ('Petq','Kaloqnova','Koleva','8805302131','163711092','Hristo Smirnenski str. 40 Blagoevgrad','F');

INSERT INTO CLIENTS (firstName,secondName,lastName,personalID,IDCardNumber,address,gender)
VALUES ('Radoslava','Kirilova','Popova','9311285513','367169740','Ivan Vazov str. 50 Smolqn','F');

INSERT INTO CLIENTS (firstName,secondName,lastName,personalID,IDCardNumber,address,gender)
VALUES ('Stoqn','Dimitrov','Raichev','7807103541','713714202','Hristo Botev str. 18 Pernik','M');

INSERT INTO CLIENTS (firstName,secondName,lastName,personalID,IDCardNumber,address,gender)
VALUES ('Tanq','Nikolaeva','Konstantinova','7408195674','206592593','Hristo Smirnenski str. 60 Varna','F');
```

	firstName	secondName	lastName	personalID	IDCardNumber	address	gender	clientsNumber
1	Angel	Georgiev	Dimitrov	8706234402	731577245	Petko Voivoda str. 10 Qmbol	M	1
2	Boris	Ivanov	Tonev	8004037820	254959893	Hristo Botev str. 25 Sliven	M	2
3	Victor	Asenov	Yanev	9005163015	323967738	Ivan Vazov str. 56 Sofia	M	3
4	Georgi	Borisov	Georgiev	7711065549	427857598	Hristo Botev str. 5 Smolqn	M	4
5	Angelina	Ivanova	Dimitrova	7901157751	224062708	Georgi Rakovski bul. 10 Haskovo	F	5
6	Borqna	Nikolova	Ivanova	8403123112	828334620	Petko R. Slaveikov str. 15 Blagoevgrad	F	6
7	Vasilena	Pavlova	Goranova	9712083802	181320927	Hristo Smirnenki str. 3 Gorna Orqhovica	F	7
8	Gergana	Atanasova	Yankova	7005193817	644724170	Ivan Vazov str. 18 Silistra	F	8
9	Dimo	Venzislavov	Yordanov	8806203871	341831589	Hristo Botev str. 12 Sofia	M	9
10	Elena	Boqnova	Kostova	6110022584	714676093	Geo Milev str. 40 Sofia	F	10
11	Zdravko	Atanasov	Popyordanov	6508153210	796177560	Atanas Ishirkov str. 12	M	11
12	Ivanka	Nikolaeva	Apostolova	7003034401	879017750	Vasil Levski str. 13 Pernik	F	12
13	Kamen	Georgiev	Todorov	7904298031	515448687	Luben Karavelov str. 5 Qmbol	M	13
14	Mariana	Stefanova	Popova	9411112115	773245521	Georgi Rakovski str. 55 Burgas	F	14
15	Nikola	Ivanov	Dimitrov	8908139150	386316542	Nikola Vapcarov str. 16 Varna	M	15
16	Ognqn	Nikolov	Hristov	5802147104	263262817	6-ti Septemvri str. 15 Kazanluk	M	16
17	Petq	Kaloqnova	Koleva	8805302131	163711092	Hristo Smirnenki str. 40 Blagoevgrad	F	17
18	Radoslava	Kirilova	Popova	9311285513	367169740	Ivan Vazov str. 50 Smolqn	F	18
19	Stoqn	Dimitrov	Raichev	7807103541	713714202	Hristo Botev str. 18 Pernik	M	19
20	Tanq	Nikolaeva	Konstantinova	7408195674	206592593	Hristo Smirnenki str. 60 Varna	F	20

Таблица със служители

--Insert some example values in the EMPLOYEES table

```

INSERT INTO EMPLOYEES (firstName,secondName,lastName,personalID,type,insuranceNumber,workingHours,salary,gender, roomNumber)
VALUES('Georgi','Kaloqnov','Petrov','8010288531','piccolo','1089800023',8,900,'M',125)

INSERT INTO EMPLOYEES (firstName,secondName,lastName,personalID,type,insuranceNumber,workingHours,salary,gender, roomNumber)
VALUES('Nikola','Ivanov','Ivanov','9005147956','piccolo','1190454666',8,900,'M',125)

INSERT INTO EMPLOYEES (firstName,secondName,lastName,personalID,type,insuranceNumber,workingHours,salary,gender, roomNumber)
VALUES('Kalina','Petrova','Dinkova','6502185863','maid','2484636141',12,1200,'F',124)

INSERT INTO EMPLOYEES (firstName,secondName,lastName,personalID,type,insuranceNumber,workingHours,salary,gender, roomNumber)
VALUES('Borislava','Kalinova','Petrova','7703269817','maid','1430622368',12,1100,'F',124)

INSERT INTO EMPLOYEES (firstName,secondName,lastName,personalID,type,insuranceNumber,workingHours,salary,gender, roomNumber)
VALUES('Hristina','Todorova','Vladeva','9609081269','maid','7069411501',12,950,'F',123)

INSERT INTO EMPLOYEES (firstName,secondName,lastName,personalID,type,insuranceNumber,workingHours,salary,gender, roomNumber)
VALUES('Hristo','Dimitrov','Kolev','6402261370','doorsman','4543769705',10,800,'M',122)

INSERT INTO EMPLOYEES (firstName,secondName,lastName,personalID,type,insuranceNumber,workingHours,salary,gender, roomNumber)
VALUES('Teodor','Kirilov','Kapitanov','8001208536','doorsman','7802942555',10,850,'M',122)

INSERT INTO EMPLOYEES (firstName,secondName,lastName,personalID,type,insuranceNumber,workingHours,salary,gender, roomNumber)
VALUES('Mariq','Yordanova','Kostova','8905175210','receptionist','8738472850',12,1500,'F',123)

INSERT INTO EMPLOYEES (firstName,secondName,lastName,personalID,type,insuranceNumber,workingHours,salary,gender, roomNumber)
VALUES('Anna','Borisova','Karaivanova','7807256398','receptionist','8733359810',12,1500,'F',121)

INSERT INTO EMPLOYEES (firstName,secondName,lastName,personalID,type,insuranceNumber,workingHours,salary,gender, roomNumber)
VALUES('Kiril','Vladislavov','Boikov','8512045219','manager','6125811490',8,2350,'M',120)

```

	firstName	secondName	lastName	personalID	type	insuranceNumber	workingHours	gender	employeeID	salary	roomNumber
1	Georgi	Kaloqnov	Petrov	8010288531	piccolo	1089800023	8	M	1	900.00	125
2	Nikola	Ivanov	Ivanov	9005147956	piccolo	1190454666	8	M	2	900.00	125
3	Kalina	Petrova	Dinkova	6502185863	maid	2484636141	12	F	3	1200.00	124
4	Borislava	Kalinova	Petrova	7703269817	maid	1430622368	12	F	4	1100.00	124
5	Hristina	Todorova	Vladeva	9609081269	maid	7069411501	12	F	5	950.00	123
6	Hristo	Dimitrov	Kolev	6402261370	doorsman	4543769705	10	M	6	800.00	122
7	Teodor	Kirilov	Kapitanov	8001208536	doorsman	7802942555	10	M	7	850.00	122
8	Mariq	Yordanova	Kostova	8905175210	receptionist	8738472850	12	F	8	1500.00	123
9	Anna	Borisova	Karaivanova	7807256398	receptionist	8733359810	12	F	9	1500.00	121
10	Kiril	Vladislavov	Boikov	8512045219	manager	6125811490	8	M	10	2350.00	120

Примерни заявки

4.1.Прости заявки

--Да се изведат имената и адресът всички редовни клиенти, живеещи в София.

```
SELECT firstName + ' ' + lastName as fullName, address FROM CLIENTS WHERE address LIKE '%Sofia';
```

	fullName	address
1	Victor Yanev	Ivan Vazov str. 56 Sofia
2	Dimo Yordanov	Hristo Botev str. 12 Sofia
3	Elena Kostova	Geo Milev str. 40 Sofia

--Да се изведат имената, длъжността и заплатата на всички служители със заплата под 1000 лв.

```
SELECT firstName + ' ' + lastName as fullName,type,salary FROM EMPLOYEES WHERE salary < 1000;
```

	fullName	type	salary
1	Georgi Petrov	piccolo	900.00
2	Nikola Ivanov	piccolo	900.00
3	Hristina Vladeva	maid	950.00
4	Hristo Kolev	doorsman	800.00
5	Teodor Kapitanov	doorsman	850.00

--Да се изведат имената на всички клиенти направили резервация за месец Юли 2019 година

```
SELECT clientsName FROM RESERVATIONS WHERE registerDate LIKE '2019-07%'
```

	clientsName
1	Georgi Kovachev
2	Ivana Petkova
3	Hristo Velez
4	Petar Georgiev
5	Daniela Cvetkova
6	Nikola Dimitrov
7	Angel Dimitrov
8	Elena Kostova
9	Victor Yanev
10	Radoslava Popova

--Да се изведат имената на всички служители работещи над 8 часа и заплата 1000 или повече лева.

```
SELECT firstName + ' ' + lastName as fullName FROM EMPLOYEES WHERE workingHours > 8 AND salary >= 1000;
```

	fullName
1	Kalina Dinkova
2	Borislava Petrova
3	Mariq Kostova
4	Anna Karaivanova

--Да се изведат имената на всички жени служители с длъжност пиколо.

```
SELECT firstName + ' ' + lastName as fullName FROM EMPLOYEES WHERE type LIKE 'piccolo' and gender LIKE 'F';
```

	fullName
--	----------

--Да се изведат номерата на всички фамилни стаи или апартаменти с капацитет над двама души.

```
SELECT roomNumber FROM ROOMS WHERE roomCapacity > 2 AND roomType LIKE 'Family' OR roomType LIKE 'Apartment'
```

	roomNumber
1	401
2	402
3	403
4	404
5	405
6	406
7	407
8	408
9	409
10	410
11	411
12	412
13	413
14	414
15	415
16	416
17	417
18	418
19	419
20	420
21	421
22	422
23	423
24	424
25	425
26	501
27	502
28	503
29	504
30	505
31	506
32	507
33	508
34	509
35	510
36	511
37	512
38	513
39	514
40	515

4.2. Заявки върху две и повече релации

```
--Да се изведат имената на всички служители и редовни клиенти родени през 80-те.  
SELECT firstName + ' ' + lastName as fullName FROM EMPLOYEES WHERE personalID LIKE '8%'  
UNION  
SELECT firstName + ' ' + lastName as fullName FROM CLIENTS WHERE personalID LIKE '8%'
```

	fullName
1	Angel Dimitrov
2	Boris Tonev
3	Borqna Ivanova
4	Dimo Yordanov
5	Georgi Petrov
6	Kiril Boikov
7	Mariq Kostova
8	Nikola Dimitrov
9	Petq Koleva
10	Teodor Kapitanov

```
--Да се изведат номерата на всички семейни стаи, които не с капацитет двама или по-малко души.  
SELECT roomNumber FROM ROOMS WHERE roomType LIKE 'Family'  
EXCEPT  
SELECT roomNumber FROM ROOMS WHERE roomCapacity <= 2
```

	roomNumber
1	401
2	402
3	403
4	404
5	405
6	406
7	407
8	408
9	409
10	410
11	411
12	412
13	413
14	414
15	415
16	416
17	417
18	418
19	419
20	420
21	421
22	422
23	423
24	424
25	425

```
--Да се изведат номерата на всички стаи резервирани през Юли 2019 с изключение на тези направени не от редовни клиенти.
SELECT roomNumber FROM RESERVATIONS WHERE registerDate LIKE '2019-07%'
EXCEPT
SELECT roomNumber FROM RESERVATIONS WHERE clientsNumber IS NULL
```

	roomNumber
1	110
2	206
3	315
4	414
5	502

```
--Да се изведат имената на всички редовни клиенти, които не живеят в София и са жени.
SELECT firstName + ' ' + lastName as fullName FROM CLIENTS WHERE address NOT LIKE '%Sofia'
INTERSECT
SELECT firstName + ' ' + lastName as fullName FROM CLIENTS WHERE gender LIKE 'F'
```

	fullName
1	Angelina Dimitrova
2	Borqna Ivanova
3	Gergana Yankova
4	Ivanka Apostolova
5	Mariana Popova
6	Petq Koleva
7	Radoslava Popova
8	Tanq Konstantinova
9	Vasilena Goranova

```
--Да се изведат ЕГН-тата на всички служители мъже и на всички служители работещи под 10 часа
SELECT personalID FROM EMPLOYEES WHERE gender LIKE 'M'
INTERSECT
SELECT personalID FROM EMPLOYEES WHERE workingHours < 10
```

	personalID
1	8010288531
2	8512045219
3	9005147956

4.3.Подзаявки

```
--Да се изведат ЕГН-тата на редовните клиенти направили резервация през 2020 година.
SELECT personalID FROM CLIENTS
WHERE clientsNumber IN ( SELECT clientsNumber FROM RESERVATIONS WHERE registerDate LIKE '2020%')
```

	personalID
1	8706234402
2	8004037820
3	7901157751
4	8908139150
5	9311285513

--Да се изведат номерата на стаите, в които са били настанени хора през месец Юли 2019 година.

```
SELECT roomNumber FROM ROOMS WHERE  
roomNumber IN (SELECT roomNumber FROM RESERVATIONS WHERE registerDate LIKE '2019-07%')
```

	roomNumber
1	103
2	110
3	125
4	206
5	222
6	315
7	404
8	414
9	502
10	512

--Да се изведат адресите на всички редовни клиенти направили резервация.

```
SELECT address FROM CLIENTS WHERE  
clientsNumber IN (SELECT clientsNumber FROM RESERVATIONS WHERE clientsNumber IS NOT NULL)
```

	address
1	Petko Voivoda str. 10 Qmbol
2	Hristo Botev str. 25 Sliven
3	Ivan Vazov str. 56 Sofia
4	Georgi Rakovski bul. 10 Haskovo
5	Geo Milev str. 40 Sofia
6	Nikola Vapcarov str. 16 Varna
7	Ivan Vazov str. 50 Smolqn

--Да се изведат видовете пакети на резервациите настанени в апартамент.

```
SELECT packet FROM RESERVATIONS WHERE  
roomNumber IN (SELECT roomNumber FROM ROOMS WHERE roomType LIKE 'Apartment')
```

	packet
1	AI
2	AI
3	AI
4	AI

--Да се изведат без да се повтарят цената за нощувка на всички стаи от първия етаж.

```
SELECT DISTINCT priceForNight FROM RESERVATIONS WHERE  
roomNumber IN (SELECT roomNumber FROM ROOMS WHERE floor LIKE 1)
```

	priceForNight
1	75.00
2	80.00
3	90.00

4.4.Съединения

--Да се изведат датите на освобождаване на фамилните стаи.

```
SELECT RS.checkOutDate FROM RESERVATIONS RS JOIN ROOMS R ON RS.roomNumber = R.roomNumber WHERE R.roomType LIKE 'Family';
```

	checkOutDate
1	2019-12-05
2	2020-02-01
3	2019-09-28
4	2019-08-10
5	2019-07-15
6	2019-07-06
7	2019-07-06
8	2019-07-19

--Да се изведат номерата на всички стаи, за които няма направена резервация.

```
SELECT DISTINCT R.roomNumber FROM ROOMS R LEFT JOIN RESERVATIONS RS ON RS.roomNumber = R.roomNumber WHERE RS.registerDate IS NULL
```

	roomNumber
1	101
2	102
3	104
4	105
5	106
6	107
7	108
8	109
9	111
10	112
11	113
12	114
13	115
14	116
15	117
16	118
17	119
18	121
19	122
20	123
21	124
22	201
23	202
24	203
25	204
26	205
27	207
28	208
29	209
30	210
31	211
32	212
33	213
34	214

	roomNumber
35	215
36	216
37	217
38	218
39	219
40	220
41	221
42	223
43	224
44	225
45	301
46	302
47	303
48	304
49	305
50	306
51	307
52	308
53	309
54	310
55	311
56	312
57	313
58	314
59	316
60	317
61	318
62	319
63	321
64	322
65	323
66	324
67	325
68	401

	roomNumber
69	402
70	403
71	405
72	406
73	407
74	408
75	409
76	410
77	411
78	412
79	413
80	415
81	416
82	417
83	418
84	419
85	420
86	421
87	422
88	423
89	424
90	425
91	501
92	503
93	504
94	505
95	506
96	507
97	508
98	509
99	510
100	511
101	513
102	514
103	515


```
--Да се изведе средната цена по етаж, закръглена до два символа след запетаята, на стаите с единични легла.
SELECT floor, FORMAT(AVG(RS.priceForNight),'N2') AS avgPrice
FROM ROOMS R JOIN RESERVATIONS RS ON RS.roomNumber = R.roomNumber
WHERE roomType LIKE 'Room' GROUP BY floor
```

	floor	avgPrice
1	1	83.75
2	3	112.50

```
--Да се изведат имената на редовните клиенти и броят направени от тях резервации през 2020г.
```

```
--(Да не се извеждат тези, които са без резервации)
```

```
SELECT firstName + ' ' + lastName as fullName, COUNT(C.clientsNumber) as NumberOfReservations
FROM CLIENTS C JOIN RESERVATIONS R ON C.clientsNumber = r.clientsNumber
WHERE R.registerDate LIKE '2020%' GROUP BY firstName,lastName
```

	fullName	NumberOfReservations
1	Angel Dimitrov	1
2	Nikola Dimitrov	1
3	Angelina Dim...	1
4	Radoslava P...	1
5	Boris Tonev	1

```
--Да се изведе броят на резервации с включена закуска за клиентите настанени на третия етаж.
```

```
SELECT COUNT(C.clientsNumber) AS clientsWithBreakfast
FROM CLIENTS C JOIN RESERVATIONS R ON C.clientsNumber = r.clientsNumber JOIN ROOMS RM ON RM.roomNumber = R.roomNumber
WHERE RM.floor = 3 AND R.packet LIKE 'BB'
```

	clientsWithBreakfast
1	2

4.5.Групиране и Агрегации

```
--Prints the average total price for a reservation
```

```
SELECT AVG(totalPrice) AS AverageReservationPrice
FROM RESERVATIONS
```

	AverageReservationPrice
1	548.958333

```
--Prints the average salary of female employees
```

```
SELECT AVG(salary) AS AverageFemaleSalary
FROM EMPLOYEES
WHERE gender = 'F'
```

	AverageFemaleSalary
1	1250.000000

```
--Prints the maximum salary of a maid
SELECT firstName, lastName, type, salary
FROM EMPLOYEES
WHERE salary = (SELECT MAX(salary) FROM EMPLOYEES WHERE type = 'maid')
```

	firstName	lastName	type	salary
1	Kalina	Dinkova	maid	1200.00

```
--Prints the number of employees by type
SELECT type, COUNT(employeeID) as EmployeesCount
FROM EMPLOYEES
GROUP BY type
```

	type	EmployeesCount
1	doorsman	2
2	maid	3
3	manager	1
4	piccolo	2
5	receptio...	2

```
--Prints the nuber of reservations by package type
SELECT packet, COUNT(*) as ReservationsCount
FROM RESERVATIONS
GROUP BY packet
```

	packet	ReservationsCount
1	AI	6
2	BB	6
3	HB	4
4	RO	8

```
--Prints the hotel income from all inclusive reservations
SELECT SUM(totalPrice) AS TotalIncome
FROM RESERVATIONS
WHERE packet = 'AI'
```

	TotalIncome
1	7040.00

```
--Prints the income of all the apartments for a week considering they are all booked
SELECT SUM(priceForNight) AS TotalIncome
FROM ROOMS
WHERE roomType = 'Apartment'
```

	TotalIncome
1	2700.00

--Prints the cheapest available regular rooms ordered by number

```
SELECT *
FROM ROOMS
WHERE priceForNight = (SELECT MIN(priceForNight) FROM ROOMS WHERE roomType = 'Room')
AND status = 'F'
AND roomType = 'Room'
ORDER BY roomNumber
```

	roomNumber	floor	status	roomCapacity	roomType	priceForNight
1	101	1	F	1	Room	75.00
2	102	1	F	1	Room	75.00
3	103	1	F	1	Room	75.00
4	104	1	F	1	Room	75.00
5	105	1	F	1	Room	75.00
6	106	1	F	1	Room	75.00
7	107	1	F	1	Room	75.00
8	108	1	F	1	Room	75.00
9	109	1	F	1	Room	75.00
10	110	1	F	1	Room	75.00
11	111	1	F	1	Room	75.00
12	112	1	F	1	Room	75.00
13	113	1	F	1	Room	75.00

--Prints the number of double rooms grouped by type

```
SELECT roomType, COUNT(roomNumber) AS TotalRooms
FROM ROOMS
WHERE roomCapacity = 2
GROUP BY roomType
```

	roomType	TotalRooms
1	Family	25
2	Room	12

--Prints the lowest paid employees working on 8 hour shifts

```
SELECT *
FROM EMPLOYEES
WHERE salary = (SELECT MIN(salary) FROM EMPLOYEES WHERE workingHours = 8)
AND workingHours = 8
```

	firstName	secondName	lastName	personalID	type	insuranceNumber	workingHours	gender	employeeID	salary	roomNumber
1	Georgi	Kaloqnov	Petrov	8010288531	piccolo	1089800023	8	M	1	900.00	125
2	Nikola	Ivanov	Ivanov	9005147956	piccolo	1190454666	8	M	2	900.00	125

Подобрения

5.1.Ограничения

```
--NOT NULL CONSTRAINTS

--ROOMS
UPDATE ROOMS SET roomNumber = -1 WHERE roomNumber IS NULL;
ALTER TABLE ROOMS ALTER COLUMN roomNumber INT NOT NULL;

UPDATE ROOMS SET priceForNight = 0.0 WHERE priceForNight IS NULL;
ALTER TABLE ROOMS ALTER COLUMN priceForNight DECIMAL(8,2) NOT NULL;

--CLIENTS
UPDATE CLIENTS SET firstName = ' ' WHERE firstName IS NULL;
ALTER TABLE CLIENTS ALTER COLUMN firstName VARCHAR(20) NOT NULL;

UPDATE CLIENTS SET lastName = ' ' WHERE lastName IS NULL;
ALTER TABLE CLIENTS ALTER COLUMN lastName VARCHAR(20) NOT NULL;

UPDATE CLIENTS SET personalID = ' ' WHERE personalID IS NULL;
ALTER TABLE CLIENTS ALTER COLUMN personalID CHAR(10) NOT NULL;

UPDATE CLIENTS SET IDCardNumber = ' ' WHERE IDCardNumber IS NULL;
ALTER TABLE CLIENTS ALTER COLUMN IDCardNumber CHAR(9) NOT NULL;

--RESERVATIONS
UPDATE RESERVATIONS SET roomNumber = -1 WHERE roomNumber IS NULL;
ALTER TABLE RESERVATIONS ALTER COLUMN roomNumber INT NOT NULL;

UPDATE RESERVATIONS SET clientsNumber = -1 WHERE clientsNumber IS NULL;
ALTER TABLE RESERVATIONS ALTER COLUMN clientsNumber INT NOT NULL;

UPDATE RESERVATIONS SET packet = ' ' WHERE packet IS NULL;
ALTER TABLE RESERVATIONS ALTER COLUMN packet CHAR(2) NOT NULL;

--EMPLOYEES
UPDATE EMPLOYEES SET firstName = ' ' WHERE firstName IS NULL;
ALTER TABLE EMPLOYEES ALTER COLUMN firstName VARCHAR(20) NOT NULL;

UPDATE EMPLOYEES SET lastName = ' ' WHERE lastName IS NULL;
ALTER TABLE EMPLOYEES ALTER COLUMN lastName VARCHAR(20) NOT NULL;

UPDATE EMPLOYEES SET personalID = ' ' WHERE personalID IS NULL;
ALTER TABLE EMPLOYEES ALTER COLUMN personalID CHAR(10) NOT NULL;

UPDATE EMPLOYEES SET type = ' ' WHERE type IS NULL;
ALTER TABLE EMPLOYEES ALTER COLUMN type VARCHAR(15) NOT NULL;

UPDATE EMPLOYEES SET insuranceNumber = ' ' WHERE insuranceNumber IS NULL;
ALTER TABLE EMPLOYEES ALTER COLUMN insuranceNumber VARCHAR(20) NOT NULL;

UPDATE EMPLOYEES SET roomNumber = 0 WHERE roomNumber IS NULL;
ALTER TABLE EMPLOYEES ALTER COLUMN roomNumber INT NOT NULL;
```

```

--PACKAGES
UPDATE PACKAGES SET package = ' ' WHERE package IS NULL;
ALTER TABLE PACKAGES ALTER COLUMN package CHAR(2) NOT NULL;

UPDATE PACKAGES SET additionalPrice = 0 WHERE additionalPrice IS NULL;
ALTER TABLE PACKAGES ALTER COLUMN additionalPrice INT NOT NULL;

|--DEFAULT

|--CLIENTS
ALTER TABLE CLIENTS ADD CONSTRAINT DF_CLIENTSECONDDNAME DEFAULT 'N/A' FOR secondName;

ALTER TABLE CLIENTS ADD CONSTRAINT DF_CLIENTGENDER DEFAULT 'N/A' FOR gender;

ALTER TABLE CLIENTS ADD CONSTRAINT DF_CLIENTADDRES DEFAULT 'N/A' FOR address;

--EMPLOYEES
ALTER TABLE EMPLOYEES ADD CONSTRAINT DF_EMPLOYEESECONDDNAME DEFAULT 'N/A' FOR secondName;

ALTER TABLE EMPLOYEES ADD CONSTRAINT DF_EMPLOYEEGENDER DEFAULT 'N/A' FOR gender;

|--UNIQUE

--CLIENTS
ALTER TABLE CLIENTS ADD CONSTRAINT UQ_CLIENTS_PERSONALID UNIQUE(personalID);

ALTER TABLE CLIENTS ADD CONSTRAINT UQ_CLIENTS_IDcardNumber UNIQUE(IDcardNumber);

ALTER TABLE CLIENTS ADD CONSTRAINT UQ_CLIENTS_NUMBER UNIQUE(clientsNumber);

--EMPLOYEES
ALTER TABLE EMPLOYEES ADD CONSTRAINT UQ_EMPLOYEES_PERSONALID UNIQUE(personalID);

ALTER TABLE EMPLOYEES ADD CONSTRAINT UQ_EMPLOYEES_INSURANCENUMBER UNIQUE(insuranceNumber);

ALTER TABLE EMPLOYEES ADD CONSTRAINT UQ_EMPLOYEES_EMPLOYEEID UNIQUE(employeeID);

--RESERVATIONS
ALTER TABLE RESERVATIONS ADD CONSTRAINT UQ_RESERVATIONS_RESERVATIONNUMBER UNIQUE(reservationNumber);

--ROOMS
ALTER TABLE ROOMS ADD CONSTRAINT UQ_ROOMS_ROOMNUMBER UNIQUE(roomNumber);

--PACKAGES
ALTER TABLE PACKAGES ADD CONSTRAINT UQ_PACKAGES_PACKAGE UNIQUE(package);

|--CHECK

--CLIENTS
ALTER TABLE CLIENTS ADD CONSTRAINT CLIENTS_GENDER_CK CHECK(gender IN('M','F'));

--EMPLOYEES
ALTER TABLE EMPLOYEES ADD CONSTRAINT EMPLOYEES_GENDER_CK CHECK(gender IN('M','F'));

--There could be included more employee roles depending on the hotel management decisions
ALTER TABLE EMPLOYEES ADD CONSTRAINT EMPLOYEES_TYPE_CK CHECK(type IN('manager', 'doorsman', 'maid', 'receptionist', 'piccolo'));

```

```

--According to the bulgarian ministry of labour and social policy it is prohibited to take daily shifts longer than 12 hours
ALTER TABLE EMPLOYEES ADD CONSTRAINT EMPLOYEES_WORKINGHOURS_CK CHECK(workingHours <= 12);

--RESERVATIONS
ALTER TABLE RESERVATIONS ADD CONSTRAINT RESERVATIONS_PACKET_CK CHECK(packet IN('RO', 'BB', 'FB', 'HB', 'AI'));

]--ROOMS

--Suppose we have 5 floors hotel, since they are only five, we can use IN instead of checking if floor is less than 6
ALTER TABLE ROOMS ADD CONSTRAINT ROOMS_FLOOR_CK CHECK(floor IN(1,2,3,4,5));

ALTER TABLE ROOMS ADD CONSTRAINT ROOMS_CAPACITY_CK CHECK(roomCapacity IN(1,2,3,4,5));

ALTER TABLE ROOMS ADD CONSTRAINT ROOMS_STATUS_CK CHECK(status IN('F', 'T', 'S'));

--Could be included more types
ALTER TABLE ROOMS ADD CONSTRAINT ROOMS_TYPE_CK CHECK(roomType IN('Room', 'Family', 'Apartment'));

--PACKAGES
ALTER TABLE PACKAGES ADD CONSTRAINT PACKAGES_PACKAGE_CK CHECK(package IN('RO', 'BB', 'FB', 'HB', 'AI'));

--PRIMARY KEYS
ALTER TABLE CLIENTS ADD CONSTRAINT PK_CLIENTS PRIMARY KEY(clientsNumber);

ALTER TABLE EMPLOYEES ADD CONSTRAINT PK_EMPLOYEES PRIMARY KEY(employeeID);

ALTER TABLE RESERVATIONS ADD CONSTRAINT PK_RESERVATIONS PRIMARY KEY(reservationNumber);

ALTER TABLE ROOMS ADD CONSTRAINT PK_ROOM PRIMARY KEY(roomNumber, priceForNight);

ALTER TABLE PACKAGES ADD CONSTRAINT PK_PACKAGES PRIMARY KEY(package);

--FOREIGN KEYS
ALTER TABLE RESERVATIONS ADD CONSTRAINT FK_RESERVATIONS_CLIENTS FOREIGN KEY(clientsNumber) REFERENCES CLIENTS(clientsNumber);

ALTER TABLE RESERVATIONS ADD CONSTRAINT FK_RESERVATIONS_PACKAGES FOREIGN KEY(packet) REFERENCES PACKAGES(package);

ALTER TABLE RESERVATIONS ADD CONSTRAINT FK_RESERVATIONS_ROOMS FOREIGN KEY(roomNumber) REFERENCES ROOMS(roomNumber);

ALTER TABLE EMPLOYEES ADD CONSTRAINT FK_EMPLOYEES_ROOMS FOREIGN KEY(roomNumber) REFERENCES ROOMS(roomNumber);

```

5.2. Тригери

```

--Trigger for time of employment
--1.
ALTER TABLE EMPLOYEES ADD employed_since DATETIME
--2.
CREATE TRIGGER employment_trigger ON employees
AFTER INSERT
AS
UPDATE employees
SET
employed_since = GETDATE()
FROM inserted
WHERE employees.employeeID= inserted.employeeID;

--Triger to block the delete operation
CREATE TRIGGER notdeleted_clients_trigger ON clients
INSTEAD OF DELETE
AS
RAISERROR ('Sorry, you can not delete clients!',16,10);

```

```

| -- Trigger INSTEAD OF DELETE on rooms
| --1.
| ALTER TABLE rooms ADD usable CHAR(1) NOT NULL DEFAULT 'T' ;
| --2.
| ALTER TABLE rooms ADD CONSTRAINT usable_as_bool CHECK (usable IN ('T','F'));
| --3.
| CREATE TRIGGER notdeleted_rooms_trigger ON rooms
| INSTEAD OF DELETE
| AS
| UPDATE rooms
| SET
| usable = 'F'
| FROM deleted
| WHERE rooms.roomNumber= deleted.roomNumber;

-- History of packages price
--1.
CREATE TABLE package_audit(
package CHAR(2),
oldPrice INT,
newPrice INT,
updated_at DATETIME
)
--2.
CREATE TRIGGER price_audit_trigger ON packages
AFTER UPDATE
AS
INSERT INTO package_audit (package, oldPrice, newPrice, updated_at)
SELECT inserted.package, deleted.additionalPrice, inserted.additionalPrice , GETDATE()
FROM inserted, deleted;
--3.
CREATE TRIGGER price_delete_trigger ON packages
AFTER DELETE
AS
INSERT INTO package_audit (package, oldPrice, updated_at)
SELECT deleted.package, deleted.additionalPrice, GETDATE()
FROM deleted;

--Trigger for employees who are laid off
--1.
CREATE TABLE former_employees (
firstName VARCHAR(20),
secondName VARCHAR(20),
lastName VARCHAR(20),
personalID CHAR(10),
type VARCHAR(15),
insuranceNumber VARCHAR(20),
workingHours INT,
salary DECIMAL(10,2),
employed_sinse DATETIME,
unemployed DATETIME
)
--2.
CREATE TRIGGER exemployees_trigger ON employees
AFTER DELETE
AS
INSERT INTO former_employees (firstName, secondName, lastName, personalID, type,
insuranceNumber,workingHours, salary, employed_sinse, unemployed)
SELECT
d.firstName, d.secondName, d.lastName, d.personalID, d.type, d.insuranceNumber, d.workingHours,
d.salary, d.employed_sinse, GETDATE()
FROM deleted d;

```

5.3.Изгледы и индекси

--INDEXES-----

```
CREATE INDEX idx_room
ON rooms (roomNumber, priceForNight);

CREATE UNIQUE INDEX idx_exemployees
ON former_employees (personalID);

CREATE UNIQUE INDEX idx_employees
ON employees (personalID);

CREATE UNIQUE INDEX idx_clients
ON clients (personalID);

CREATE CLUSTERED INDEX idx_package_changes
ON package_audit (package);
```

| --VIEWS-----

--1.

```
CREATE VIEW reservations_with_breakfast
AS
SELECT reservationNumber, registerDate, checkOutDate
FROM reservations
WHERE packet IN ('BB', 'HB', 'FB', 'AI');
```

--2.

```
SELECT reservationNumber, DATEDIFF(day, registerDate, checkOutDate) AS days
FROM reservations_with_breakfast;
--WHERE reservationNumber = 10
```

	reservationNumber	days
1	2	4
2	3	8
3	4	1
4	5	2
5	8	6
6	9	2
7	10	9
8	11	2
9	14	5
10	15	3
11	16	3
12	17	5
13	20	7
14	21	4
15	22	7
16	23	4


```
--1.
CREATE VIEW number_usable_rooms_view
AS
SELECT roomType, COUNT(roomType) AS TotalRooms, usable
FROM ROOMS
GROUP BY roomType, usable;
```

```
--2.
SELECT *
FROM number_usable_rooms_view
WHERE roomType LIKE 'Room'
```

	roomType	TotalRooms	usable
1	Room	50	T

```
--1.
CREATE VIEW employee_position_view
AS
SELECT type, COUNT(type) AS number,salary
FROM employees
GROUP BY type,salary;
```

```
--2.
SELECT *
FROM employee_position_view
WHERE type LIKE 'maid';
```

	type	number	salary
1	maid	1	950.00
2	maid	1	1100.00
3	maid	1	1200.00

```
--1.
CREATE VIEW man_employee
AS
SELECT *
FROM EMPLOYEES
WHERE gender LIKE 'M'
WITH CHECK OPTION;
```

```
--2.
SELECT *
FROM man_employee
```

```
--3.1.
INSERT INTO man_employee (firstName, secondName, lastName, personalID,
                           type,insuranceNumber, workingHours,gender, salary)
VALUES ('Dimityr', 'Ivanov', 'Boykov', '8110304594',
        'piccolo','1488369752', 8,'M', 950)
```

```
--3.2.Error
INSERT INTO man_employee (firstName, secondName, lastName, personalID,
                           type,insuranceNumber, workingHours,gender, salary)
VALUES ('Mariq', 'Ivanov', 'Boykova', '9310304594', 'maid',
        '1487369752', 8,'F', 950)
```