

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

ΟΜΑΔΙΚΗ ΕΡΓΑΣΙΑ

ΕΞΑΜΗΝΟ 4^ο

ΟΝΟΜΑ	ΑΜ
ΥΦΑΝΤΗΣ ΓΕΩΡΓΙΟΣ	Π18163
ΦΩΤΟΠΟΥΛΟΣ ΙΩΑΝΝΗΣ	Π18238
ΚΑΤΕΒΑΣ ΧΡΗΣΤΟΣ	Π18068

ΠΑΡΟΥΣΙΑΣΗ ΕΡΓΑΣΙΑΣ

ΕΡΩΤΗΜΑ 1^ο

Στο πρώτο ερώτημα δεν περιέχει τίποτα περισσότερο από την δημιουργία πινάκων ,τύπων και η εισαγωγή δεδομένων στους πίνακες.

--THE DATABASE IS IN 3 NORMAL FORM

```
CREATE TYPE Room AS ENUM('DBL BUDGET','DBL STD','DBL DLX',
                          'TPL BUDGET','TPL STD','TPL DLX',
                          'QUAD BUDGET','QUAD STD','QUAD DLX'); --ALL THE ROOM TYPES
```

```
CREATE TABLE RESERVATIONS (
  ReservationID int NOT NULL,
  NameOfReservation name NOT NULL,
  DateOfReservation timestamp with time zone NOT NULL,
  Pax int NOT NULL,
  NumberOfRooms int NOT NULL,
  RoomID int NOT NULL,
  ClientID int NOT NULL,
  Check_In date NOT NULL,
  Check_Out date NOT NULL,
  Stay int NOT NULL,
  PaymentID int NOT NULL,
  PRIMARY KEY( ReservationID )
);
```

```
CREATE TABLE RESERVATION_PAYMENT (
  PaymentID int NOT NULL,
  ReservationID int NOT NULL,
  Bill int NOT NULL,
  Paid boolean NOT NULL,
  Deadline date NOT NULL,
  PRIMARY KEY( PaymentID )
);
```

```
CREATE EXTENSION citext;
CREATE DOMAIN emailDomain AS citext
CHECK ( value ~* '^[a-zA-Z0-9.!#$%&'"+/=?^_`{}~]*@[a-zA-Z0-9]({:[a-zA-Z0-9-]({0,61}[a-zA-Z0-9])?({:[a-zA-Z0-9]({:[a-zA-Z0-9-]({0,61}[a-zA-Z0-9])?)*$' );

CREATE TYPE titles AS ENUM('Mr','Ms'); --ALL THE TITLES
```

```
CREATE TABLE CLIENTS (
    ClientID int NOT NULL,
    Title text NOT NULL,
    Name text NOT NULL,
    Age int NOT NULL,
    HomeID int NOT NULL,
    PhoneNumber numeric(10,0) NOT NULL UNIQUE,
    Email emailDomain NOT NULL UNIQUE,
    CardNumber numeric(16,0) NOT NULL UNIQUE,
    PRIMARY KEY( ClientID )
);

CREATE TABLE CLIENT_HOME(
    HomeID int NOT NULL,
    ClientID int NOT NULL,
    Country text NOT NULL,
    Town text NOT NULL,
    Street text NOT NULL,
    Street_Num int NOT NULL,
    Postcode text NOT NULL,
    PRIMARY KEY( HomeID )
);

CREATE TABLE ROOMS (
    RoomID int NOT NULL,
    FacilitiesID int NOT NULL,
    RoomCapacity int NOT NULL, -- DOUBLE-TWIN(2),TRIPLE(3),QUAD(4)
    RoomType text NOT NULL, -- COST ZONE (BUDGET,STANDARD (STD) ,DELUXE (DLX))
    RoomFullName room NOT NULL, --DBL BUDGET,DBL STD, ETC.
    Cost int NOT NULL,
    PRIMARY KEY( RoomID )
);

CREATE TABLE TRANSACTIONS (
    TransactionID int NOT NULL,
    ReservationID int NOT NULL,
    ClientID int NOT NULL,
    CardNumber numeric(16,0) NOT NULL,
    Cost int NOT NULL,
    PRIMARY KEY( TransactionID )
);

CREATE TABLE ROOM_FACILITIES (
    FacilityID int NOT NULL,
    RoomID int NOT NULL,
    Breakfast boolean NOT NULL,
    StreetView boolean NOT NULL,
    SeaView boolean NOT NULL,
    MountainView boolean NOT NULL,
    Spa boolean NOT NULL,
    Bathtub boolean NOT NULL,
    Shower boolean NOT NULL,
    TV boolean NOT NULL,
    PRIMARY KEY( FacilityID )
);
```

```
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (1, 'Somerset McCaw', '2019-07-11 03:17:57', 2, 17, 18,1, '2019-04-03', '2019-04-05', 2, 1);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (2, 'George Francis', '2019-07-12 17:40:28', 1, 18, 18,2, '2020-07-01', '2020-07-16', 4, 2);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (3, 'Janelle Poon', '2019-12-14 05:00:00', 3, 13, 3,3, '2020-07-07', '2020-07-27', 15, 3);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (4, 'Giannis Fotopoulos', '2019-09-20 07:30:18', 3, 6, 3,4, '2020-08-11', '2020-09-19', 20, 4);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (5, 'Giannis Fotopoulos', '2019-10-00 04:14:51', 3, 16, 5, '2020-03-14', '2020-12-28', 19, 5);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (6, 'Giannis Fotopoulos', '2019-11-17 12:12:52', 1, 13, 10,6, '2020-12-16', '2020-12-24', 8, 6);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (7, 'George Francis', '2020-06-06 03:11:27', 2, 9, 8, '2019-12-18', '2019-12-18', 11, 7);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (8, 'Janelle Poon', '2019-09-27 23:16:48', 1, 1, 2,8, '2020-04-26', '2020-05-01', 6, 8);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (9, 'Somerset McCaw', '2019-10-13 01:21:01', 2, 14, 14, '2019-08-01', '2019-08-01', 12, 9);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (10, 'George Francis', '2019-05-18 21:33:13', 2, 18, 14,10, '2019-02-24', '2019-02-26', 3, 10);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (11, 'George Francis', '2019-07-11 03:17:57', 2, 18, 14,11, '2019-07-01', '2019-07-01', 12, 11);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (12, 'Somerset McCaw', '2020-01-22 11:48:51', 2, 9, 8, '2020-06-23', '2020-06-24', 1, 12);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (13, 'George Francis', '2019-10-13 01:21:01', 2, 14, 14, '2019-08-01', '2019-08-01', 12, 13);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (14, 'Angelica Tesco', '2019-06-08 01:01:01', 2, 5, 17, '2020-12-01', '2020-12-10', 7, 14);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (15, 'George Francis', '2019-05-18 21:33:13', 2, 18, 14,15, '2019-08-01', '2019-08-01', 12, 15);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (16, 'George Francis', '2019-05-18 21:33:13', 2, 18, 14,16, '2019-08-01', '2019-08-01', 12, 16);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (17, 'Christa Kateva', '2020-03-27 01:24:01', 2, 18, 17, '2020-02-04', '2020-02-06', 2, 17);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (18, 'Janelle Poon', '2019-11-01 06:00:00', 4, 16, 13,18, '2020-03-20', '2020-03-20', 5, 18);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (19, 'Thatcher Murre', '2019-10-17 01:18:13', 1, 4, 3,19, '2019-10-11', '2019-10-18', 7, 19);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (20, 'Somerset McCaw', '2019-08-04 00:44:08', 4, 6, 15,20, '2020-05-15', '2020-05-18', 3, 20);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (21, 'Carree Groome', '2020-04-21 13:25:38', 1, 6, 5,21, '2019-10-03', '2019-10-12', 12, 21);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (22, 'Christa Kateva', '2019-07-08 20:16:45', 1, 7, 12,22, '2020-07-05', '2020-07-24', 19, 22);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (23, 'Giannis Fotopoulos', '2020-05-03 18:59:40', 1, 7, 12,23, '2020-06-24', '2020-06-24', 6, 23);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (24, 'Christa Kateva', '2019-07-08 20:16:45', 1, 7, 12,24, '2020-07-24', '2020-07-24', 19, 24);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (25, 'Angelica Tesco', '2019-10-29 16:45:00', 2, 9, 2,25, '2020-07-06', '2020-07-22', 16, 25);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (26, 'Somerset McCaw', '2019-08-04 00:44:08', 4, 6, 15,26, '2020-05-15', '2020-05-18', 3, 26);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (27, 'Carree Groome', '2019-10-25 11:02:08', 4, 1, 2,27, '2020-08-13', '2020-08-17', 5, 27);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (28, 'George Francis', '2019-05-18 21:33:13', 2, 18, 4,28, '2020-05-13', '2020-05-13', 3, 28);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (29, 'Giannis Fotopoulos', '2020-05-05 14:04:12', 2, 14, 17,29, '2020-01-01', '2020-01-17', 16, 29);
insert into Reservations (reservationid, nameofreservation, dateofreservation, pa, numberofrooms, roomid,ClientID, check_in, check_out, stay, paymentid) values (30, 'Giannis Fotopoulos', '2019-06-29 18:15:09', 4, 18, 1,30, '2020-05-17', '2020-05-17', 3, 30);

insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (1, 1, 126, true, '2020-12-08-19');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (2, 2, 87, false, '2020-03-05');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (3, 3, 103, false, '2020-03-05');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (4, 4, 183, false, '2020-06-20');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (5, 5, 183, false, '2020-06-20');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (6, 6, 139, true, '2020-08-03');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (7, 7, 96, false, '2020-08-03');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (8, 8, 146, false, '2020-11-07');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (9, 9, 138, false, '2020-11-07');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (10, 10, 188, false, '2020-11-17');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (11, 11, 173, true, '2020-11-17');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (12, 12, 55, false, '2020-06-23');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (13, 13, 133, false, '2020-05-05');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (14, 14, 176, false, '2020-05-05');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (15, 15, 156, false, '2020-05-04');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (16, 16, 149, false, '2020-05-04');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (17, 17, 174, false, '2020-06-05');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (18, 18, 53, false, '2020-05-30');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (19, 19, 149, false, '2020-04-29');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (20, 20, 162, false, '2020-12-20');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (21, 21, 75, false, '2020-10-23');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (22, 22, 146, false, '2020-10-23');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (23, 23, 136, true, '2020-08-21');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (24, 24, 144, false, '2020-04-08');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (25, 25, 56, false, '2020-11-15');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (26, 26, 178, false, '2020-07-08');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (27, 27, 77, true, '2020-05-25');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (28, 28, 175, true, '2020-07-08');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (29, 29, 145, true, '2020-07-08');
insert into Reservation_Payment (paymentid, reservationid, bill, paid, deadline) values (30, 30, 217, true, '2020-06-18');

insert into Clients (Clientid, Title, Name, Age,HomeID, PhoneNumber, Email, CardNumber) values (1, 'Mr', 'Luigi Silbray', '11', '4248317608', 'lsilbray@hhs.us', '35320073084517');
insert into Clients (Clientid, Title, Name, Age,HomeID, PhoneNumber, Email, CardNumber) values (2, 'Mr', 'New McKeel', '12', '65318421', 'newmckeel@hhs.us', '3812055482815');
insert into Clients (Clientid, Title, Name, Age,HomeID, PhoneNumber, Email, CardNumber) values (3, 'Mr', 'Natalie Baldwin', '3', '5715815', 'natalieb@hhs.us', '3540057240444');
insert into Clients (Clientid, Title, Name, Age,HomeID, PhoneNumber, Email, CardNumber) values (4, 'Mr', 'Natalie Baldwin', '3', '5715815', 'natalieb@hhs.us', '3540057240444');
insert into Clients (Clientid, Title, Name, Age,HomeID, PhoneNumber, Email, CardNumber) values (5, 'Ms', 'Narene Agilione', '5', '581980895', 'nagilione@hhs.us', '3716141610789');
insert into Clients (Clientid, Title, Name, Age,HomeID, PhoneNumber, Email, CardNumber) values (6, 'Ms', 'Holly Miller', '6', '59553054', 'holliem@hhs.us', '4004751545412');
insert into Clients (Clientid, Title, Name, Age,HomeID, PhoneNumber, Email, CardNumber) values (7, 'Mr', 'Harley Kial', '7', '775937884', 'haliokial@hhs.us', '3532151344665');
insert into Clients (Clientid, Title, Name, Age,HomeID, PhoneNumber, Email, CardNumber) values (8, 'Ms', 'Natalie Baldwin', '3', '5715815', 'natalieb@hhs.us', '3540057240444');
insert into Clients (Clientid, Title, Name, Age,HomeID, PhoneNumber, Email, CardNumber) values (9, 'Ms', 'Ruggiero Kelings', '9', '50422978', 'ruggiero@hhs.us', '5002234674591');
insert into Clients (Clientid, Title, Name, Age,HomeID, PhoneNumber, Email, CardNumber) values (10, 'Mr', 'Fred Ciala', '10,18', '403518138', 'fciala@hhs.us', '35500502209448');
insert into Clients (Clientid, Title, Name, Age,HomeID, PhoneNumber, Email, CardNumber) values (11, 'Mr', 'Moris Ochwell', '11,11', '860421231', 'mochwell@hhs.us', '12017338806125');
insert into Clients (Clientid, Title, Name, Age,HomeID, PhoneNumber, Email, CardNumber) values (12, 'Mr', 'Gordon Docker', '12,13', '169184378', 'cdocker@hhs.us', '357912373228793');
insert into Clients (Clientid, Title, Name, Age,HomeID, PhoneNumber, Email, CardNumber) values (13, 'Mr', 'Gertrude Dyer', '13,16', '19178456', 'gdyer@hhs.us', '357912373228793');
insert into Clients (Clientid, Title, Name, Age,HomeID, PhoneNumber, Email, CardNumber) values (14, 'Mr', 'Gail Hervo', '14,15', '79570884', 'ghervo@hhs.us', '3540057240444');
insert into Clients (Clientid, Title, Name, Age,HomeID, PhoneNumber, Email, CardNumber) values (15, 'Ms', 'Holly Miller', '15,16', '59553054', 'holliem@hhs.us', '4004751545412');
insert into Clients (Clientid, Title, Name, Age,HomeID, PhoneNumber, Email, CardNumber) values (16, 'Ms', 'Holly Miller', '16,17', '59553054', 'holliem@hhs.us', '4004751545412');
insert into Clients (Clientid, Title, Name, Age,HomeID, PhoneNumber, Email, CardNumber) values (17, 'Ms', 'Hagan McCaghan', '17,17', '7319071782', 'hmcaghan@hhs.us', '35891254182008');
insert into Clients (Clientid, Title, Name, Age,HomeID, PhoneNumber, Email, CardNumber) values (18, 'Ms', 'Carree Maud', '18,18', '597648866', 'cmaud@hhs.us', '542367244841181');
insert into Clients (Clientid, Title, Name, Age,HomeID, PhoneNumber, Email, CardNumber) values (19, 'Mr', 'Farris Furumston', '19,19', '368088074', 'ffurumston@hhs.us', '6180195212335857');
insert into Clients (Clientid, Title, Name, Age,HomeID, PhoneNumber, Email, CardNumber) values (20, 'Mr', 'Gwen Wallicker', '20,20', '1726706410', 'gwallicker@hhs.us', '3589803685516494');

insert into Client_Home (HomeID,Clientid,Country, Town, Street, Street_Num, PostCode) values (1, 'Bulgaria', 'Athens', 'Ermoou', '1', '2630');
insert into Client_Home (HomeID,Clientid,Country, Town, Street, Street_Num, PostCode) values (2, '2, 'France', 'Athens', 'Ermoou', '2', '94264 CEDEX');
insert into Client_Home (HomeID,Clientid,Country, Town, Street, Street_Num, PostCode) values (3, '3, 'Venezuela', 'Athens', 'Ermoou', '3', '921264 CEDEX');
insert into Client_Home (HomeID,Clientid,Country, Town, Street, Street_Num, PostCode) values (4, '4, 'Belarus', 'Athens', 'Ermoou', '4', '969-7208');
insert into Client_Home (HomeID,Clientid,Country, Town, Street, Street_Num, PostCode) values (5, '5, 'Japan', 'Athens', 'Ermoou', '5', '969-7208');
insert into Client_Home (HomeID,Clientid,Country, Town, Street, Street_Num, PostCode) values (6, '6, 'China', 'Athens', 'Ermoou', '6', '115543');
insert into Client_Home (HomeID,Clientid,Country, Town, Street, Street_Num, PostCode) values (7, '7, 'Indonesia', 'Athens', 'Ermoou', '7', '15335');
insert into Client_Home (HomeID,Clientid,Country, Town, Street, Street_Num, PostCode) values (8, '8, 'China', 'Athens', 'Ermoou', '8', '115543');
insert into Client_Home (HomeID,Clientid,Country, Town, Street, Street_Num, PostCode) values (9, '9, 'Czech Republic', 'Athens', 'Ermoou', '9', '468 46');
insert into Client_Home (HomeID,Clientid,Country, Town, Street, Street_Num, PostCode) values (10, '10, 'Indonesia', 'Athens', 'Ermoou', '10, '555445');
insert into Client_Home (HomeID,Clientid,Country, Town, Street, Street_Num, PostCode) values (11, '11, 'Brazil', 'Athens', 'Ermoou', '11, '555445');
insert into Client_Home (HomeID,Clientid,Country, Town, Street, Street_Num, PostCode) values (12, '12, 'Indonesia', 'Athens', 'Ermoou', '12, '54-515');
insert into Client_Home (HomeID,Clientid,Country, Town, Street, Street_Num, PostCode) values (13, '13, 'Poland', 'Athens', 'Ermoou', '13, '05-288');
insert into Client_Home (HomeID,Clientid,Country, Town, Street, Street_Num, PostCode) values (14, '14, 'Indonesia', 'Athens', 'Ermoou', '14, '555445');
insert into Client_Home (HomeID,Clientid,Country, Town, Street, Street_Num, PostCode) values (15, '15, 'Indonesia', 'Athens', 'Ermoou', '15, '54654');
insert into Client_Home (HomeID,Clientid,Country, Town, Street, Street_Num, PostCode) values (16, '16, 'United States', 'Athens', 'Ermoou', '16, '48232');
insert into Client_Home (HomeID,Clientid,Country, Town, Street, Street_Num, PostCode) values (17, '17, 'China', 'Athens', 'Ermoou', '17, 'ds45');
insert into Client_Home (HomeID,Clientid,Country, Town, Street, Street_Num, PostCode) values (18, '18, 'Nicaragua', 'Athens', 'Ermoou', '18, 'ds454');
insert into Client_Home (HomeID,Clientid,Country, Town, Street, Street_Num, PostCode) values (19, '19, 'Sweden', 'Athens', 'Ermoou', '19, '121 63');
insert into Client_Home (HomeID,Clientid,Country, Town, Street, Street_Num, PostCode) values (20, '20, 'China', 'Athens', 'Ermoou', '20, '1545');

insert into Rooms (RoomID,FacilitiesID, RoomCapacity, RoomType,RoomFullName, Cost) values (1,1, 1, 'STD', 'DBL STD', 20);
insert into Rooms (RoomID,FacilitiesID, RoomCapacity, RoomType,RoomFullName, Cost) values (2,2, 2, 'STD', 'TPL STD', 40);
insert into Rooms (RoomID,FacilitiesID, RoomCapacity, RoomType,RoomFullName, Cost) values (3,3, 3, 'STD', 'DBL DLX', 40);
insert into Rooms (RoomID,FacilitiesID, RoomCapacity, RoomType,RoomFullName, Cost) values (4,4, 4, 'STD', 'QUAD STD', 40);
insert into Rooms (RoomID,FacilitiesID, RoomCapacity, RoomType,RoomFullName, Cost) values (5,5, 5, 'STD', 'DBL STD', 20);
insert into Rooms (RoomID,FacilitiesID, RoomCapacity, RoomType,RoomFullName, Cost) values (6,6, 6, 'STD', 'DBL BUDGET', 18);
insert into Rooms (RoomID,FacilitiesID, RoomCapacity, RoomType,RoomFullName, Cost) values (7,7, 7, 'STD', 'DBL STD', 20);
insert into Rooms (RoomID,FacilitiesID, RoomCapacity, RoomType,RoomFullName, Cost) values (8,8, 8, 'STD', 'TPL STD', 40);
insert into Rooms (RoomID,FacilitiesID, RoomCapacity, RoomType,RoomFullName, Cost) values (9,9, 9, 'STD', 'DBL STD', 20);
insert into Rooms (RoomID,FacilitiesID, RoomCapacity, RoomType,RoomFullName, Cost) values (10,10, 10, 'STD', 'QUAD STD', 40);
insert into Rooms (RoomID,FacilitiesID, RoomCapacity, RoomType,RoomFullName, Cost) values (11,11, 11, 'STD', 'QUAD DLX', 80);
insert into Rooms (RoomID,FacilitiesID, RoomCapacity, RoomType,RoomFullName, Cost) values (12,12, 12, 'STD', 'QUAD DLX', 80);
insert into Rooms (RoomID,FacilitiesID, RoomCapacity, RoomType,RoomFullName, Cost) values (13,13, 13, 'STD', 'TPL STD', 40);
insert into Rooms (RoomID,FacilitiesID, RoomCapacity, RoomType,RoomFullName, Cost) values (14,14, 14, 'STD', 'TPL STD', 40);
insert into Rooms (RoomID,FacilitiesID, RoomCapacity, RoomType,RoomFullName, Cost) values (15,15, 15, 'STD', 'TPL BUDGET', 20);
insert into Rooms (RoomID,FacilitiesID, RoomCapacity, RoomType,RoomFullName, Cost) values (16,16, 16, 'STD', 'QUAD BUDGET', 40);
insert into Rooms (RoomID,FacilitiesID, RoomCapacity, RoomType,RoomFullName, Cost) values (17,17, 17, 'STD', 'QUAD STD', 40);
insert into Rooms (RoomID,FacilitiesID, RoomCapacity, RoomType,RoomFullName, Cost) values (18,18, 18, 'STD', 'DBL STD', 20);
insert into Rooms (RoomID,FacilitiesID, RoomCapacity, RoomType,RoomFullName, Cost) values (19,19, 19, 'STD', 'TPL STD', 40);
insert into Rooms (RoomID,FacilitiesID, RoomCapacity, RoomType,RoomFullName, Cost) values (20,20, 20, 'STD', 'QUAD STD', 40);

insert into room_facilities (facilityID, roomID, breakfast, streetview, seaview, mountainview, spa, bathtub, shower, tv) values (1, 1, true, false, false, false, true, true, true);
insert into room_facilities (facilityID, roomID, breakfast, streetview, seaview, mountainview, spa, bathtub, shower, tv) values (2, 2, false, false, true, true, false, false, false);
insert into room_facilities (facilityID, roomID, breakfast, streetview, seaview, mountainview, spa, bathtub, shower, tv) values (3, 3, false, true, false, true, false, false, true);
insert into room_facilities (facilityID, roomID, breakfast, streetview, seaview, mountainview, spa, bathtub, shower, tv) values (4, 4, false, true, true, false, true, true, false);
insert into room_facilities (facilityID, roomID, breakfast, streetview, seaview, mountainview, spa, bathtub, shower, tv) values (5, 5, true, true, false, true, true, true, true);
insert into room_facilities (facilityID, roomID, breakfast, streetview, seaview, mountainview, spa, bathtub, shower, tv) values (6, 6, true, true, false, true, true, true, true);
insert into room_facilities (facilityID, roomID, breakfast, streetview, seaview, mountainview, spa, bathtub, shower, tv) values (7, 7, false, true, false, true, true, true, true);
insert into room_facilities (facilityID, roomID, breakfast, streetview, seaview, mountainview, spa, bathtub, shower, tv) values (8, 8, false, true, false, true, false, false, true);
insert into room_facilities (facilityID, roomID, breakfast, streetview, seaview, mountainview, spa, bathtub, shower, tv) values (9, 9, false, true, false, true, false, false, true);
insert into room_facilities (facilityID, roomID, breakfast, streetview, seaview, mountainview, spa, bathtub, shower, tv) values (10, 10, false, true, false, false, false, false, true);
insert into room_facilities (facilityID, roomID, breakfast, streetview, seaview, mountainview, spa, bathtub, shower, tv) values (11, 11, false, true, false, false, true, false, true);
insert into room_facilities (facilityID, roomID, breakfast, streetview, seaview, mountainview, spa, bathtub, shower, tv) values (12, 12, true, true, false, true, false, false, true);
insert into room_facilities (facilityID, roomID, breakfast, streetview, seaview, mountainview, spa, bathtub, shower, tv) values (13, 13, false, false, true, true, false, false, true);
insert into room_facilities (facilityID, roomID, breakfast, streetview, seaview, mountainview, spa, bathtub, shower, tv) values (14, 14, false, true, true, true, false, true, true);
insert into room_facilities (facilityID, roomID, breakfast, streetview, seaview, mountainview, spa, bathtub, shower, tv) values (15, 15, true, true, false, false, false, false, true);
insert into room_facilities (facilityID, roomID, breakfast, streetview, seaview, mountainview, spa, bathtub, shower, tv) values (16, 16, false, false, true, true, false, false, true);
insert into room_facilities (facilityID, roomID, breakfast, streetview, seaview, mountainview, spa, bathtub, shower, tv) values (17, 17, false, false, false, true, true, true, true);
insert into room_facilities (facilityID, roomID, breakfast, streetview, seaview, mountainview, spa, bathtub, shower, tv) values (18, 18, true, false, true, false, false, false, true);
insert into room_facilities (facilityID, roomID, breakfast, streetview, seaview, mountainview, spa, bathtub, shower, tv) values (19, 19, false, true, false, false, true, false, true);
insert into room_facilities (facilityID, roomID, breakfast, streetview, seaview, mountainview, spa, bathtub, shower, tv) values (20, 20, true, false, false, true, false, false, true);

insert into Transactions (TransactionID, ReservationID, ClientID, CardNumber, Cost) values (1, 1, 1, '3008361807444715', 1);
insert into Transactions (TransactionID, ReservationID, ClientID, CardNumber, Cost) values (2, 2, 2, '561899378580695', 2);
insert into Transactions (TransactionID, ReservationID, ClientID, CardNumber, Cost) values (3, 3, 3, '5641821323833770', 3);
insert into Transactions (TransactionID, ReservationID, ClientID, CardNumber, Cost) values (4, 4, 4, '5002350743086692', 4);
insert into Transactions (TransactionID, ReservationID, ClientID, CardNumber, Cost) values (5, 5, 5, '384726252074215', 5);
insert into Transactions (TransactionID, ReservationID, ClientID, CardNumber, Cost) values (6, 6, '67547651759945', 6);
insert into Transactions (TransactionID, ReservationID, ClientID, CardNumber, Cost) values (7, 7, '357319733627328', 7);
insert into Transactions (TransactionID, ReservationID, ClientID, CardNumber, Cost) values (8, 8, '357891716006526', 8);
insert into Transactions (TransactionID, ReservationID, ClientID, CardNumber, Cost) values (9, 9, '560227098517931', 9);
insert into Transactions (TransactionID, ReservationID, ClientID, CardNumber, Cost) values (10, 10, '35515913464282', 10);
insert into Transactions (TransactionID, ReservationID, ClientID, CardNumber, Cost) values (11, 11, '4041595938114857', 11);
insert into Transactions (TransactionID, ReservationID, ClientID, CardNumber, Cost) values (12, 12, '357599073796522', 12);
insert into Transactions (TransactionID, ReservationID, ClientID, CardNumber, Cost) values (13, 13, '67861344839888', 13);
insert into Transactions (TransactionID, ReservationID, ClientID, CardNumber, Cost) values (14, 14, '3572081472241688', 14);
insert into Transactions (TransactionID, ReservationID, ClientID, CardNumber, Cost) values (15, 15, '3538476072728469', 15);
insert into Transactions (TransactionID, ReservationID, ClientID, CardNumber, Cost) values (16, 16, '450842767360353', 16);
insert into Transactions (TransactionID, ReservationID, ClientID, CardNumber, Cost) values (17, 17, '354008213829562', 17);
insert into Transactions (TransactionID, ReservationID, ClientID, CardNumber, Cost) values (18, 18, '367339173673143', 18);
insert into Transactions (TransactionID, ReservationID, ClientID, CardNumber, Cost) values (19, 19, '5602211487860979', 19);
insert into Transactions (TransactionID, ReservationID, ClientID, CardNumber, Cost) values (20, 20, '6761590681912474', 20);
```

ΕΡΩΤΗΜΑ 2^ο

(a)

Επιλέγουμε την στήλη **ROOMFULLNAME** (το είδος δωματίου ανάλογα με την τιμή και τον τύπο) και τις φορές που εμφανίζεται ο κάθε τύπος δωματίου από τον πίνακα **ROOMS**. Στην 3^η γραμμή του κώδικα αντιστοιχούμε τις εγγραφές έτσι ώστε να γίνει σωστά η σύνδεση των πινάκων. Επίσης κάνουμε **GROUP BY** βάσει τον τύπο του δωματίου.

Query Editor Query History

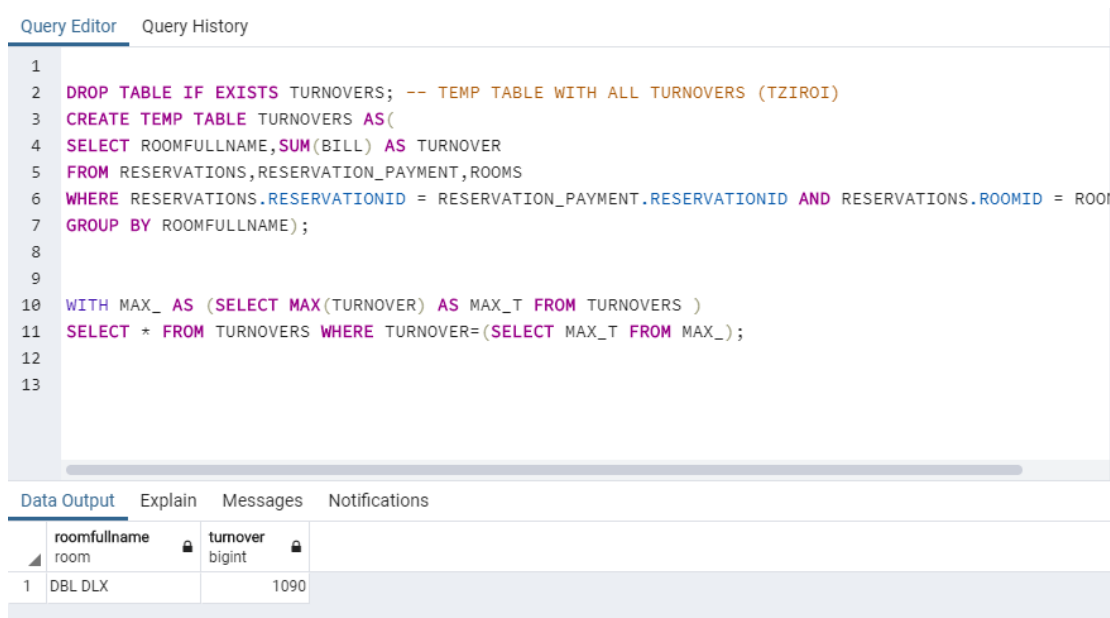
```
1 SELECT ROOMFULLNAME, COUNT (ROOMFULLNAME)
2 FROM RESERVATIONS, ROOMS
3 WHERE RESERVATIONS.ROOMID = ROOMS.ROOMID
4 GROUP BY ROOMFULLNAME;
5
```

Data Output Explain Messages Notifications

	roomfullname room	count bigint	
1	QUAD DLX	1	
2	TPL STD	4	
3	TPL DLX	4	
4	DBL DLX	7	
5	DBL STD	7	
6	QUAD STD	6	
7	TPL BUDGET	1	

(b)

Η εντολή **DROP TABLE IF EXISTS** μας επιτρέπει να διαγράψουμε τον πίνακα εάν υπάρχει ήδη. Στη συνέχεια δημιουργούμε έναν νέο προσωρινό πίνακα με όνομα **TURNOVERS** και σε συνδυασμό με την εντολή **AS** δημιουργείται βάση των εμπεριεχομένων της παρένθεσης που ακολουθεί. Η χρήση των εντολών **SELECT** και **FROM** μας επιτρέπει από ποιους πίνακες (**RESERVATIONS, RESERVATION_PAYMENT, ROOMS**) να διαλέξουμε ποιες στήλες(**ROOMFULLNAME**) βάση των περιορισμών της εντολής **WHERE**. Με την εντολή **WITH** δημιουργούμε έναν προσωρινό πίνακα **MAX_** ο οποίος έχει τη στήλη **MAX_T** που περιέχει τον μέγιστο τζίρο και χρησιμοποιείται για την σύγκριση στο τελευταίο **SELECT**



The screenshot shows a SQL Query Editor with a query that creates a temporary table **TURNOVERS** and uses a **WITH** clause to find the maximum turnover.

```
1
2 DROP TABLE IF EXISTS TURNOVERS; -- TEMP TABLE WITH ALL TURNOVERS (TZIROI)
3 CREATE TEMP TABLE TURNOVERS AS(
4 SELECT ROOMFULLNAME,SUM(BILL) AS TURNOVER
5 FROM RESERVATIONS,RESERVATION_PAYMENT,ROOMS
6 WHERE RESERVATIONS.RESERVATIONID = RESERVATION_PAYMENT.RESERVATIONID AND RESERVATIONS.ROOMID = ROOMS.ROOMID
7 GROUP BY ROOMFULLNAME);
8
9
10 WITH MAX_ AS (SELECT MAX(TURNOVER) AS MAX_T FROM TURNOVERS )
11 SELECT * FROM TURNOVERS WHERE TURNOVER=(SELECT MAX_T FROM MAX_);
12
13
```

Below the query editor, the **Data Output** tab is active, showing the results of the query. The output is a table with two columns: **roomfullname** and **turnover**.

	roomfullname	turnover
1	DBL DLX	1090

(c)

Ομοίως εδώ δημιουργούμε έναν προσωρινό πίνακα **ENGAGED** βάσει της παρένθεσης που ακολουθεί μετά την εντολή **AS**. Επιλέγουμε την στήλη **ROOMID** από τον πίνακα **RESERVATION** που έχουν **CHECK_IN** (δεδομένα της στήλης) μικρότερα ή ίσα του αποτελέσματος της συνάρτησης **CURRENT_DATE** και **CHECK_OUT**(δεδομένα της στήλης) μεγαλύτερα. Επιλέγουμε όλες της εγγραφές που υπάρχουν στον πίνακα **ROOMS** και δεν υπάρχουν στον πίνακα **ENGAGED**.

1	--(c) Πόσα δωμάτια είναι προς το παρόν διαθέσιμα προς κράτηση.
2	--θα εμφανίζει ποσα δωματια είναι διαθεσιμα προς κρατηση σημερα.
3	
4	WITH ENGAGED AS(SELECT ROOMID FROM RESERVATIONS WHERE CHECK_IN <= CURRENT_DATE AND CHECK_OUT > CURRENT_DATE)
5	
6	
7	SELECT 'AVAILABLE ROOMS',COUNT(*)
8	FROM ROOMS
9	LEFT JOIN ENGAGED USING(ROOMID)
10	WHERE ENGAGED.ROOMID IS NULL;
11	

Data Output	Explain	Messages	Notifications												
<table> <tr> <th>?</th> <th>column?</th> <th></th> <th>count</th> </tr> <tr> <th></th> <th>text</th> <th></th> <th>bigint</th> </tr> <tr> <td>1</td> <td>AVAILABLER...</td> <td></td> <td>20</td> </tr> </table>	?	column?		count		text		bigint	1	AVAILABLER...		20			
?	column?		count												
	text		bigint												
1	AVAILABLER...		20												

(d)

ΣΗΜΕΙΩΣΗ

Η ΠΡΩΤΗ ΓΡΑΜΜΗ ΚΩΔΙΚΑ ΕΙΝΑΙ Η ΕΞΗΣ:

```
SELECT
ROOMS.ROOMID,ROOMFULLNAME,ROOM_FACILITIES.BreakFast,
ROOM_FACILITIES.StreetView, ROOM_FACILITIES.SeaView,
ROOM_FACILITIES.MountainView, ROOM_FACILITIES.Spa,
ROOM_FACILITIES.Bathtub, ROOM_FACILITIES.Shower,
ROOM_FACILITIES.TV
```

Επιλέγουμε τον αριθμό δωματίου καθώς και τον τύπο του και τις διάφορες παροχές που αντιστοιχούν στα δωμάτια. Οι παροχές είναι τύπου BOOLEAN


```

1  --(d) Ποιες παροχές (facilities) διατίθενται σε συγκεκριμένα δωμάτια
2
3  SELECT ROOMS.ROOMID,ROOMFULLNAME,ROOM_FACILITIES.BreakFast, ROOM_FACILITIES.StreetView, ROOM_FACILITIES.SeaView,
4  FROM ROOMS,ROOM_FACILITIES
5  WHERE ROOMS.ROOMID = ROOM_FACILITIES.ROOMID;
6

```

Data Output		Explain	Messages	Notifications							
	roomid integer	roomfullname room	breakfast boolean	streetview boolean	seaview boolean	mountainview boolean	spa boolean	bathtub boolean	shower boolean	tv boolean	
1	1	DBL STD	true	false	false	false	true	true	true	true	
2	2	TPL STD	false	false	true	true	false	false	false	false	
3	3	DBL DLX	false	true	false	true	false	false	false	true	
4	4	QUAD STD	false	true	true	false	false	true	true	false	
5	5	DBL STD	true	true	false	true	true	true	false	true	
6	6	DBL BUDGET	true	true	true	true	true	false	false	true	
7	7	DBL STD	false	true	false	true	false	true	true	true	
8	8	TPL STD	false	true	false	true	false	false	false	false	
9	9	DBL STD	false	true	false	true	false	true	true	false	
10	10	QUAD STD	false	true	false	false	false	false	true	false	
11	11	DBL DLX	false	true	false	false	true	false	false	true	
12	12	QUAD DLX	true	true	false	true	true	false	false	false	
13	13	TPL STD	false	false	true	true	false	true	false	true	
14	14	TPL DLX	false	true	true	true	false	true	true	false	
15	15	TPL BUDGET	true	true	false	false	false	false	false	true	
16	16	QUAD BUDGET	false	false	false	true	true	false	false	true	
17	17	QUAD STD	false	false	false	true	true	true	false	false	
18	18	DBL STD	true	false	true	false	false	true	false	true	
19	19	TPL STD	false	true	false	false	true	false	true	true	
20	20	QUAD STD	true	false	false	true	false	false	false	false	

(e)

Επιλέγουμε τις στήλες **RESERVATIONID**, **NAMEOFRESERVATION**, **CHECK_IN**, **CHECK_OUT**, **ROOMID** από τον πίνακα **RESERVATIONS** έτσι ώστε να ελέγξουμε ποιοι επισκέπτες έχουν κάνει κράτηση αυτόν τον μήνα.

Query Editor

Query History

```
1 SELECT RESERVATIONID,NAMEOFRESERVATION,CHECK_IN,CHECK_OUT,ROOMID
2 FROM RESERVATIONS
3 WHERE ((EXTRACT(MONTH FROM CHECK_IN) = EXTRACT(MONTH FROM LOCALTIMESTAMP)) OR (EXTRACT(MONTH FROM CHECK_OUT) = EXTRACT(MONTH FROM LOCALTIMESTAMP)))
4 AND ((EXTRACT(YEAR FROM CHECK_IN) = EXTRACT(YEAR FROM LOCALTIMESTAMP)) OR (EXTRACT(YEAR FROM CHECK_OUT) = EXTRACT(YEAR FROM LOCALTIMESTAMP)));
```

Data Output

Explain

Messages

Notifications

	reservationid [PK] integer	nameofreservation name	check_in date	check_out date	roomid integer
1	12	Somerset McCaw	2020-06-23	2020-06-24	8
2	23	Giannis Fotopoulos	2020-06-18	2020-06-24	12

(f)

ΣΗΜΕΙΩΣΗ

Η ΔΕΥΤΕΡΗ ΓΡΑΜΜΗ ΚΩΔΙΚΑ ΕΙΝΑΙ Η ΕΞΗΣ:

WHERE RESERVATIONS.PAYMENTID =
RESERVATION_PAYMENT.PAYMENTID AND (EXTRACT(MONTH FROM

CHECK_IN) >= 6) AND (EXTRACT(MONTH FROM CHECK_IN)<=9) AND
RESERVATIONS.ROOMID = ROOMS.ROOMID

Επιλέγουμε την στήλη **ROOMFULLNAME**. Με την συνάρτηση **ROUND** και **AVG** βρίσκουμε τον ΜΟ των στοιχείων της στήλης **BILL** και τα
στρογγυλοποιεί μέχρι 2 δεκαδικά ψηφία. Με το **LOCALTIMESTAMP**
χρησιμοποιούμε την σημερινή ημερομηνία.

```

1 SELECT ROOMFULLNAME,ROUND(AVG(BILL),2)
2 FROM RESERVATIONS,RESERVATION_PAYMENT,ROOMS
3 WHERE RESERVATIONS.PAYMENTID = RESERVATION_PAYMENT.PAYMENTID AND (EXTRACT(MONTH FROM CHECK_IN) >= 6) AND (EXTR
4 AND (EXTRACT(YEAR FROM CHECK_IN) = EXTRACT(YEAR FROM LOCALTIMESTAMP))
5 GROUP BY RESERVATIONS.ROOMID,ROOMFULLNAME
6

```

	roomfullname room	round numeric
1	DBL STD	161.00
2	TPL STD	56.00
3	DBL DLX	151.50
4	TPL STD	55.00
5	QUAD STD	87.00
6	QUAD DLX	136.00
7	TPL DLX	77.00

(g)

Χρησιμοποιούμε το **WITH** για να δημιουργήσουμε δυο προσωρινούς
πίνακες **SDA** και **MAX_SDA**. Ομοίως με τα παραπάνω ερωτήματα
χρησιμοποιούνται οι εντολές **AS()**, **SELECT**, **FROM**, **WHERE GROUP BY**.

```

1 WITH SDA AS (
2 SELECT NAMEOFRESERVATION,ROOMFULLNAME,SUM(NUMBEROFFROOMS) AS SUMA
3 FROM RESERVATIONS,ROOMS
4 WHERE RESERVATIONS.ROOMID = ROOMS.ROOMID
5 GROUP BY ROOMFULLNAME,nameofreservation
6 ORDER BY ROOMFULLNAME)
7 ,
8 MAX_SDA AS (
9 SELECT ROOMFULLNAME,MAX(SUMA) AS MAX_SUMA
10 FROM SDA
11 GROUP BY ROOMFULLNAME)
12
13 SELECT NAMEOFRESERVATION,MAX_SDA.ROOMFULLNAME,MAX_SUMA
14 FROM SDA,MAX_SDA
15 WHERE SUMA = MAX_SUMA AND MAX_SDA.ROOMFULLNAME = SDA.ROOMFULLNAME;






```

	nameofreservation name	roomfullname room	max_suma bigint
1	Somerset McCaw	DBL STD	33
2	George Ifantis	DBL DLX	30
3	George Ifantis	TPL BUDGET	6
4	Jecho Wailes	TPL STD	16
5	George Ifantis	TPL DLX	33
6	Giannis Fotopoulos	QUAD STD	32
7	Giannis Fotopoulos	QUAD DLX	7

(h)

Τα δωμάτια που ενοικιάστηκαν από την ημερομηνία χ(συγκεκριμένα στο παράδειγμα 01-01-2020) τα βρίσκουμε με την ένωση των πινάκων **RESERVATIONS**, **ROOMS** με την συνθήκη ότι έχουν το ίδιο **ROOMID**.

```
1 SELECT NAMEOFRESERVATION, RESERVATIONS.ROOMID, ROOMFULLNAME, CHECK_IN, CHECK_OUT
2 FROM RESERVATIONS, ROOMS
3 WHERE CHECK_IN >= '2020-01-01'::DATE AND CHECK_IN <= CURRENT_DATE AND RESERVATIONS.ROOMID = ROOMS.ROOMID
4 ORDER BY CHECK_IN;
```

Data Output		Explain	Messages	Notifications	
	 nameofreservation name	 roomid integer	 roomfullname room	 check_in date	 check_out date
1	Giannis Fotopoulos	17	QUAD STD	2020-01-01	2020-01-17
2	Somerset McCaw	7	DBL STD	2020-01-15	2020-01-17
3	Christos Katevas	11	DBL DLX	2020-02-04	2020-02-06
4	Jecho Wailles	13	TPL STD	2020-03-20	2020-03-25
5	Christos Katevas	9	DBL STD	2020-03-21	2020-03-23
6	Janelle Poon	2	TPL STD	2020-04-26	2020-05-01
7	George Ifantis	3	DBL DLX	2020-05-04	2020-05-18
8	George Ifantis	15	TPL BUDGET	2020-05-11	2020-05-30
9	Giannis Fotopoulos	12	QUAD DLX	2020-06-18	2020-06-24
10	Somerset McCaw	8	TPL STD	2020-06-23	2020-06-24

ΕΡΩΤΗΜΑ 3^ο

(a)

Το trigger καλεί το function μετά από κάθε εγγραφή, για κάθε γραμμή του πίνακα **RESERVATION_PAYMENT**.

Το trigger function μετά από κάθε εγγραφή ελέγχει αν το δωμάτιο που έχει κλείσει ο πελάτης είναι ελεύθερο ή κρατημένο. Αν το δωμάτιο είναι ελεύθερο τότε η κράτηση γίνεται κανονικά. Σε περίπτωση που είναι κρατημένο, ψάχνει για άλλο δωμάτιο ίδιου τύπου και αν υπάρχει αλλάζει ο αριθμός του δωματίου στην κράτηση, διαφορετικά ψάχνει για δωμάτιο διαφορετικού τύπου και διαλέγει τυχαία απ' τα διαθέσιμα δωμάτια. Η τιμή προσαρμόζεται ανάλογα τον τύπο του δωματίου. Αν δεν βρει κανένα διαθέσιμο δωμάτιο τότε εμφανίζει αντίστοιχο μήνυμα.

```
-- Trigger: CHANGE_ROOM_TRIG
```

```
-- DROP TRIGGER "CHANGE_ROOM_TRIG" ON public.reservation_payment;
```

```
CREATE TRIGGER "CHANGE_ROOM_TRIG"
  AFTER INSERT
  ON public.reservation_payment
  FOR EACH ROW
  EXECUTE PROCEDURE public."CHANGE_ROOM"();
```

```
-- FUNCTION: public."CHANGE_ROOM"()
-- DROP FUNCTION public."CHANGE_ROOM"();

CREATE FUNCTION public."CHANGE_ROOM"()
  RETURNS trigger
  LANGUAGE 'plpgsql'
  COST 100
  VOLATILE NOT LEAKPROOF
AS $BODY$DECLARE
n_c integer; -- THE NUMBER OF RESERVATIONS IN THE SAME ROOM AT THE SAME TIME
n_c1 integer; -- THE NUMBER OF AVAILABLE ROOMS OF THE SAME TYPE
n_c2 integer; -- THE NUMBER OF AVAILABLE ROOMS OF A DIFFERENT TYPE

OLDROOMCOST integer; -- THE COST OF THE CURRENT ROOM
NEWROOMCOST integer; -- THE COST OF THE NEW ROOM

DISCOUNTPERCENT numeric; -- THE PERCENTAGE OF THE DISCOUNT

NEWROOMID integer; -- THE ID OF THE NEW ROOM
NEWBILL INTEGER; -- THE NEW BILL AFTER THE DISCOUNT

ROOM ROOM; -- THE CURRENT ROOM TYPE

BEGIN

DROP TABLE IF EXISTS reservation ; -- STORE THE NEW RESERVATION
CREATE TEMP TABLE reservation AS(
  select * from RESERVATIONS where RESERVATIONS.reservationid = new.reservationid);

SELECT COUNT(*) -- COUNT THE NUMBER OF RESERVATIONS IN THE SAME ROOM AT THE SAME TIME
INTO n_c
FROM RESERVATIONS,reservation
WHERE reservation.ROOMID = RESERVATIONS.ROOMID AND (reservation.CHECK_IN = RESERVATIONS.CHECK_IN OR (reservation.CHECK_IN >RESERVATIONS.CHECK_IN AND reservation.CHECK_IN <RESERVATIONS.CHECK_OUT));

IF n_c > 0 --IF THERE IS MORE THAN ONE
THEN
  SELECT ROOMFULLNAME
  INTO ROOM
  FROM ROOMS,reservation
  WHERE ROOMS.ROOMID = reservation.ROOMID;

DROP TABLE IF EXISTS SAME_ROOM_TYPE_ENGAGED ; --CREATE A TABLE WITH ALL THE ENGAGED ROOMS OF THE SAME TYPE
CREATE TEMP TABLE SAME_ROOM_TYPE_ENGAGED AS(
  SELECT rooms.roomid AS ROOMID,COALESCE(count(reservations.*))
  FROM rooms,reservations,reservation
  WHERE rooms.roomid = reservations.roomid and roomfullname = ROOM and reservations.check_in = reservation.CHECK_IN
  GROUP BY rooms.roomid);
```

```

DROP TABLE IF EXISTS SAME_ROOM_TYPE_AVAIL; --CREATE A TABLE WITH ALL THE AVAILABLE ROOMS OF THE SAME TYPE
CREATE TEMP TABLE SAME_ROOM_TYPE_AVAIL AS(
SELECT rooms.roomid AS ROOMID
FROM ROOMS,SAME_ROOM_TYPE_ENGAGED
WHERE ROOMS.ROOMID NOT IN(SELECT ROOMID FROM SAME_ROOM_TYPE_ENGAGED) and roomfullname = ROOM);

SELECT COUNT(*) -- COUNT THE AVAILABLE ONES
INTO n_c1
FROM SAME_ROOM_TYPE_AVAIL;

IF(n_c1 > 0) -- IF THERE IS AT LEAST ONE SELECT A RANDOM FROM THE AVAILABLE ONES AND MAKE IT THE NEW ROOM
THEN
UPDATE RESERVATIONS
SET ROOMID = (SELECT ROOMID FROM SAME_ROOM_TYPE_AVAIL ORDER BY RANDOM() LIMIT 1)
WHERE RESERVATIONS.RESERVATIONID = NEW.RESERVATIONID;

ELSE -- ELSE CHECK FOR A DIFFERENT TYPE

DROP TABLE IF EXISTS DIFF_ROOM_TYPE_ENGAGED ;
CREATE TEMP TABLE DIFF_ROOM_TYPE_ENGAGED AS(
select rooms.roomid,COALESCE(count(reservations.*))
from rooms,reservations,reservation
where roomfullname != ROOM and reservations.check_in = reservation.CHECK_IN
group by rooms.roomid);

DROP TABLE IF EXISTS DIFF_ROOM_TYPE_AVAIL;
CREATE TEMP TABLE DIFF_ROOM_TYPE_AVAIL AS(
SELECT rooms.roomid AS ROOMID
FROM ROOMS,DIFF_ROOM_TYPE_ENGAGED
WHERE ROOMS.ROOMID != DIFF_ROOM_TYPE_ENGAGED.ROOMID and roomfullname != ROOM);

SELECT COUNT(*) -- COUNT THE AVAILABLE ONES
INTO n_c2
FROM DIFF_ROOM_TYPE_AVAIL;

IF(n_c2 > 0)
THEN
SELECT ROOMID
INTO NEWROOMID
FROM DIFF_ROOM_TYPE_AVAIL
ORDER BY RANDOM() LIMIT 1;

SELECT COST
INTO OLDROOMCOST
FROM ROOMS
WHERE ROOMFULLNAME = ROOM;

SELECT COST
INTO NEWROOMCOST
FROM ROOMS
WHERE ROOMID = NEWROOMID;

IF(NEWROOMCOST < OLDROOMCOST)
THEN
SELECT (oldroomcost - newroomcost) * 100.0/oldroomcost - 5
INTO DISCOUNTPERCENT;

UPDATE RESERVATIONS
SET ROOMID = NEWROOMID
WHERE RESERVATIONS.RESERVATIONID = NEW.RESERVATIONID;

SELECT BILL INTO NEWBILL FROM RESERVATION_PAYMENT WHERE RESERVATIONID = NEW.RESERVATIONID;

UPDATE RESERVATION_PAYMENT
SET BILL = BILL - round((DISCOUNTPERCENT * BILL )/ 100)
WHERE RESERVATION_PAYMENT.RESERVATIONID = NEW.RESERVATIONID;

SELECT BILL INTO NEWBILL FROM RESERVATION_PAYMENT WHERE RESERVATIONID = NEW.RESERVATIONID;

ELSE
UPDATE RESERVATIONS
SET ROOMID = NEWROOMID
WHERE RESERVATIONS.RESERVATIONID = NEW.RESERVATIONID;

END IF;

ELSE
RAISE NOTICE'THERE IS NOT AN AVAILABLE ROOM';
DELETE FROM RESERVATIONS WHERE RESERVATIONS.RESERVATIONID = NEW.RESERVATIONID;
DELETE FROM ROOMS WHERE RESERVATIONS.RESERVATIONID = NEW.RESERVATIONID;

END IF;

END IF;

END IF;
RETURN NEW;
END;
$body$;

ALTER FUNCTION public."CHANGE_ROOM"()
OWNER TO postgres;

```

(b)

Εκτελείται το ερώτημα 2(α) με την χρήση cursor

```
CREATE OR REPLACE FUNCTION my_funct(roomid1 integer)
RETURNS TABLE(room text) as $$
DECLARE
    rec_rooms RECORD;

    cur_rooms CURSOR(roomid1 integer) FOR
    SELECT ROOMFULLNAME
    FROM ROOMS
    WHERE ROOMS.ROOMID = roomid1;

BEGIN

    OPEN cur_rooms(roomid1);

LOOP

    FETCH cur_rooms INTO rec_rooms;

    EXIT WHEN NOT FOUND;

    room := rec_rooms.ROOMFULLNAME;

    RETURN NEXT;
END LOOP;
CLOSE cur_rooms;
END;$$
LANGUAGE plpgsql;

SELECT F.ROOM,COUNT(F.ROOM) FROM RESERVATIONS R,my_funct(R.ROOMID) F GROUP BY F.ROOM;
```

ΕΡΩΤΗΜΑ 4^ο

Το πρόγραμμα έχει πραγματοποιηθεί σε java console application και ζητάει από τον χρήστη να πληκτρολογήσει ένα γράμμα από a – h που αντιστοιχεί στο query από το ΕΡΩΤΗΜΑ 2 και επιστρέφει το αντίστοιχο αποτέλεσμα του query. Για το ερώτημα 2 (β) εκτελούνται 2 queries καθώς το πρώτο που έχει να κάνει με DROP και CREATE TABLE δεν επιστρέφει κάτι στον χρήστη και πρέπει να εκτελεστεί με statement.execute

Ακολουθούν screenshot από την εκτέλεση του προγράμματος και την εκτέλεση κάθε query.

```
↑ "C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:D:\IntelliJ IDEA 2020.1.2\lib\ide
↓ Welcome!!
⏸ Please Select which query you want to run:
⚙ (a) Πόσες κρατήσεις αντιστοιχούν σε κάθε κατηγορία δωματίων.
⚙ (b) Ποια κατηγορία δωματίων παρουσιάζει βάσει των κρατήσεων τον μεγαλύτερο τζίρο.
⚙ (c) Πόσα δωμάτια είναι προς το παρόν διαθέσιμα προς κράτηση.
⚙ (d) Ποιες παροχές (facilities) διατίθενται σε συγκεκριμένα δωμάτια.
⚙ (e) Ποιοι επισκέπτες έχουν κράτηση αυτό το μήνα.
⚙ (f) Ποιο είναι το μέσο κέρδος ανά τύπο/κατηγορία δωματίου για τη θερινή σεζόν.
⚙ (g) Ποιος είναι ο πελάτης με τις περισσότερες κρατήσεις ανά εύρος ζώνης τιμής δωματίου.
⚙ (h) Ποια δωμάτια ενοικιάστηκαν από την ημερομηνία "X" έως και σήμερα.
⚙ OR WRITE 0 (NUMBER) TO EXIT!
```

```
Please Select which query you want to run:
(a) Πόσες κρατήσεις αντιστοιχούν σε κάθε κατηγορία δωματίων.
(b) Ποια κατηγορία δωματίων παρουσιάζει βάσει των κρατήσεων τον μεγαλύτερο τζίρο.
(c) Πόσα δωμάτια είναι προς το παρόν διαθέσιμα προς κράτηση.
(d) Ποιες παροχές (facilities) διατίθενται σε συγκεκριμένα δωμάτια.
(e) Ποιοι επισκέπτες έχουν κράτηση αυτό το μήνα.
(f) Ποιο είναι το μέσο κέρδος ανά τύπο/κατηγορία δωματίου για τη θερινή σεζόν.
(g) Ποιος είναι ο πελάτης με τις περισσότερες κρατήσεις ανά εύρος ζώνης τιμής δωματίου.
(h) Ποια δωμάτια ενοικιάστηκαν από την ημερομηνία "X" έως και σήμερα.
OR WRITE 0 (NUMBER) TO EXIT!

0
ROOMFULLNAME || NUM
=====
QUAD STD || 7

TPL DLX || 4

TPL BUDGET || 1

DBL STD || 7

DBL DLX || 6

QUAD DLX || 1

TPL STD || 4

Please Select which query you want to run:
(a) Πόσες κρατήσεις αντιστοιχούν σε κάθε κατηγορία δωματίων.
(b) Ποια κατηγορία δωματίων παρουσιάζει βάσει των κρατήσεων τον μεγαλύτερο τζίρο.
(c) Πόσα δωμάτια είναι προς το παρόν διαθέσιμα προς κράτηση.
(d) Ποιες παροχές (facilities) διατίθενται σε συγκεκριμένα δωμάτια.
(e) Ποιοι επισκέπτες έχουν κράτηση αυτό το μήνα.
(f) Ποιο είναι το μέσο κέρδος ανά τύπο/κατηγορία δωματίου για τη θερινή σεζόν.
(g) Ποιος είναι ο πελάτης με τις περισσότερες κρατήσεις ανά εύρος ζώνης τιμής δωματίου.
(h) Ποια δωμάτια ενοικιάστηκαν από την ημερομηνία "X" έως και σήμερα.
OR WRITE 0 (NUMBER) TO EXIT!

0
ROOMFULLNAME || TURNOVER
=====
QUAD STD || 987

Please Select which query you want to run:
(a) Πόσες κρατήσεις αντιστοιχούν σε κάθε κατηγορία δωματίων.
(b) Ποια κατηγορία δωματίων παρουσιάζει βάσει των κρατήσεων τον μεγαλύτερο τζίρο.
(c) Πόσα δωμάτια είναι προς το παρόν διαθέσιμα προς κράτηση.
(d) Ποιες παροχές (facilities) διατίθενται σε συγκεκριμένα δωμάτια.
(e) Ποιοι επισκέπτες έχουν κράτηση αυτό το μήνα.
(f) Ποιο είναι το μέσο κέρδος ανά τύπο/κατηγορία δωματίου για τη θερινή σεζόν.
(g) Ποιος είναι ο πελάτης με τις περισσότερες κρατήσεις ανά εύρος ζώνης τιμής δωματίου.
(h) Ποια δωμάτια ενοικιάστηκαν από την ημερομηνία "X" έως και σήμερα.
OR WRITE 0 (NUMBER) TO EXIT!
```

Please Select which query you want to run:

- (a) Πόσες κρατήσεις αντιστοιχούν σε κάθε κατηγορία δωματίων.
 - (b) Ποια κατηγορία δωματίων παρουσιάζει βάσει των κρατήσεων τον μεγαλύτερο τζίρο.
 - (c) Πόσα δωμάτια είναι προς το παρόν διαθέσιμα προς κράτηση.
 - (d) Ποιες παροχές (facilities) διατίθενται σε συγκεκριμένα δωμάτια.
 - (e) Ποιοι επισκέπτες έχουν κράτηση αυτό το μήνα.
 - (f) Ποιο είναι το μέσο κέρδος ανά τύπο/κατηγορία δωματίου για τη θερινή σεζόν.
 - (g) Ποιος είναι ο πελάτης με τις περισσότερες κρατήσεις ανά εύρος ζώνης τιμής δωματίου.
 - (h) Ποια δωμάτια ενοικιάστηκαν από την ημερομηνία "X" έως και σήμερα.
- OR WRITE 0 (NUMBER) TO EXIT!

AVAILABLE ROOMS || 20

=====

Please Select which query you want to run:

- (a) Πόσες κρατήσεις αντιστοιχούν σε κάθε κατηγορία δωματίων.
 - (b) Ποια κατηγορία δωματίων παρουσιάζει βάσει των κρατήσεων τον μεγαλύτερο τζίρο.
 - (c) Πόσα δωμάτια είναι προς το παρόν διαθέσιμα προς κράτηση.
 - (d) Ποιες παροχές (facilities) διατίθενται σε συγκεκριμένα δωμάτια.
 - (e) Ποιοι επισκέπτες έχουν κράτηση αυτό το μήνα.
 - (f) Ποιο είναι το μέσο κέρδος ανά τύπο/κατηγορία δωματίου για τη θερινή σεζόν.
 - (g) Ποιος είναι ο πελάτης με τις περισσότερες κρατήσεις ανά εύρος ζώνης τιμής δωματίου.
 - (h) Ποια δωμάτια ενοικιάστηκαν από την ημερομηνία "X" έως και σήμερα.
- OR WRITE 0 (NUMBER) TO EXIT!

ROOMID	ROOMFULLNAME	BREAKFAST	STREET VIEW	SEA VIEW	MOUNTAIN VIEW	SPA	BATHTUB	SHOWER	TV
1	DBL STD	t	f	f	f	t	t	t	t
2	TPL STD	f	f	t	t	f	f	f	f
3	DBL DLX	f	t	f	t	f	f	f	t
4	QUAD STD	f	t	t	f	f	t	t	f
5	DBL STD	t	t	f	t	t	t	f	t
6	DBL BUDGET	t	t	t	t	t	f	f	t
7	DBL STD	f	t	f	t	f	t	t	t
8	TPL STD	f	t	f	t	f	f	f	f
9	DBL STD	f	t	f	t	f	t	t	f
10	QUAD STD	f	t	f	f	f	f	t	f

Please Select which query you want to run:

- (a) Πόσες κρατήσεις αντιστοιχούν σε κάθε κατηγορία δωματίων.
 - (b) Ποια κατηγορία δωματίων παρουσιάζει βάσει των κρατήσεων τον μεγαλύτερο τζίρο.
 - (c) Πόσα δωμάτια είναι προς το παρόν διαθέσιμα προς κράτηση.
 - (d) Ποιες παροχές (facilities) διατίθενται σε συγκεκριμένα δωμάτια.
 - (e) Ποιοι επισκέπτες έχουν κράτηση αυτό το μήνα.
 - (f) Ποιο είναι το μέσο κέρδος ανά τύπο/κατηγορία δωματίου για τη θερινή σεζόν.
 - (g) Ποιος είναι ο πελάτης με τις περισσότερες κρατήσεις ανά εύρος ζώνης τιμής δωματίου.
 - (h) Ποια δωμάτια ενοικιάστηκαν από την ημερομηνία "X" έως και σήμερα.
- OR WRITE 0 (NUMBER) TO EXIT!

RESERVATIONID	NAME OF RESERVATION	CHECK IN	CHECK OUT	ROOMID
12	Somerset McCaw	2020-06-23	2020-06-24	8
23	Giannis Fotopoulos	2020-06-18	2020-06-24	12

Please Select which query you want to run:

- (a) Πόσες κρατήσεις αντιστοιχούν σε κάθε κατηγορία δωματίων.
 - (b) Ποια κατηγορία δωματίων παρουσιάζει βάσει των κρατήσεων τον μεγαλύτερο τζίρο.
 - (c) Πόσα δωμάτια είναι προς το παρόν διαθέσιμα προς κράτηση.
 - (d) Ποιες παροχές (facilities) διατίθενται σε συγκεκριμένα δωμάτια.
 - (e) Ποιοι επισκέπτες έχουν κράτηση αυτό το μήνα.
 - (f) Ποιο είναι το μέσο κέρδος ανά τύπο/κατηγορία δωματίου για τη θερινή σεζόν.
 - (g) Ποιος είναι ο πελάτης με τις περισσότερες κρατήσεις ανά εύρος ζώνης τιμής δωματίου.
 - (h) Ποια δωμάτια ενοικιάστηκαν από την ημερομηνία "X" έως και σήμερα.
- OR WRITE 0 (NUMBER) TO EXIT!


```

f
ROOMFULLNAME  ||  BILL PER ROOM
=====
DBL STD  ||  161.00

TPL STD  ||  56.00

DBL DLX  ||  151.50

TPL STD  ||  55.00

QUAD STD  ||  87.00

QUAD DLX  ||  136.00

TPL DLX  ||  77.00

```

Please Select which query you want to run:

- (a) Πόσες κρατήσεις αντιστοιχούν σε κάθε κατηγορία δωματίων.
 - (b) Ποια κατηγορία δωματίων παρουσιάζει βάσει των κρατήσεων τον μεγαλύτερο τζίρο.
 - (c) Πόσα δωμάτια είναι προς το παρόν διαθέσιμα προς κράτηση.
 - (d) Ποιες παροχές (facilities) διατίθενται σε συγκεκριμένα δωμάτια.
 - (e) Ποιοι επισκέπτες έχουν κράτηση αυτό το μήνα.
 - (f) Ποιο είναι το μέσο κέρδος ανά τύπο/κατηγορία δωματίου για τη θερινή σεζόν.
 - (g) Ποιος είναι ο πελάτης με τις περισσότερες κρατήσεις ανά εύρος ζώνης τιμής δωματίου.
 - (h) Ποια δωμάτια ενοικιάστηκαν από την ημερομηνία "X" έως και σήμερα.
- OR WRITE 0 (NUMBER) TO EXIT!

Main x

```

(f) Ποιο είναι το μέσο κέρδος ανά τύπο/κατηγορία δωματίου για τη θερινή σεζόν.
(g) Ποιος είναι ο πελάτης με τις περισσότερες κρατήσεις ανά εύρος ζώνης τιμής δωματίου.
(h) Ποια δωμάτια ενοικιάστηκαν από την ημερομηνία "X" έως και σήμερα.
OR WRITE 0 (NUMBER) TO EXIT!

```

```

g
NAME OF RESERVATION  ||  ROOMFULLNAME  ||  MAX_SUMA
=====
Somerset McCaw  ||  DBL STD  ||  33

George Ifantis  ||  DBL DLX  ||  30

George Ifantis  ||  TPL BUDGET  ||  6

Jecho Wailes  ||  TPL STD  ||  16

George Ifantis  ||  TPL DLX  ||  33

Giannis Fotopoulos  ||  QUAD STD  ||  32

Giannis Fotopoulos  ||  QUAD DLX  ||  7

```

Please Select which query you want to run:

- (a) Πόσες κρατήσεις αντιστοιχούν σε κάθε κατηγορία δωματίων.
 - (b) Ποια κατηγορία δωματίων παρουσιάζει βάσει των κρατήσεων τον μεγαλύτερο τζίρο.
 - (c) Πόσα δωμάτια είναι προς το παρόν διαθέσιμα προς κράτηση.
 - (d) Ποιες παροχές (facilities) διατίθενται σε συγκεκριμένα δωμάτια.
 - (e) Ποιοι επισκέπτες έχουν κράτηση αυτό το μήνα.
 - (f) Ποιο είναι το μέσο κέρδος ανά τύπο/κατηγορία δωματίου για τη θερινή σεζόν.
 - (g) Ποιος είναι ο πελάτης με τις περισσότερες κρατήσεις ανά εύρος ζώνης τιμής δωματίου.
 - (h) Ποια δωμάτια ενοικιάστηκαν από την ημερομηνία "X" έως και σήμερα.
- OR WRITE 0 (NUMBER) TO EXIT!

```

Main x
(c) Πόσα δωμάτια είναι προς το παρόν διαθέσιμα προς κράτηση.
(d) Ποιες παροχές (facilities) διατίθενται σε συγκεκριμένα δωμάτια.
(e) Ποιοι επισκέπτες έχουν κράτηση αυτό το μήνα.
(f) Ποιο είναι το μέσο κέρδος ανά τύπο/κατηγορία δωματίου για τη θερινή σεζόν.
(g) Ποιος είναι ο πελάτης με τις περισσότερες κρατήσεις ανά εύρος ζώνης τιμής δωματίου.
(h) Ποια δωμάτια ανοικιάστηκαν από την ημερομηνία "X" έως και σήμερα.
OR WRITE 0 (NUMBER) TO EXIT!

0
NAME OF RESERVATION || ROOMID || ROOMFULLNAME || || CHECK IN || CHECK OUT
=====
Giannis Fotopoulos || 17 || QUAD STD || 2020-01-01 || 2020-01-17

Somerset McCaw || 7 || DBL STD || 2020-01-15 || 2020-01-17

Christos Katevas || 17 || QUAD STD || 2020-02-04 || 2020-02-06

Jecho Wailles || 13 || TPL STD || 2020-03-20 || 2020-03-25

Christos Katevas || 9 || DBL STD || 2020-03-21 || 2020-03-23

Janelle Poon || 2 || TPL STD || 2020-04-26 || 2020-05-01

George Ifantis || 3 || DBL DLX || 2020-05-04 || 2020-05-18

George Ifantis || 15 || TPL BUDGET || 2020-05-11 || 2020-05-30

Giannis Fotopoulos || 12 || QUAD DLX || 2020-06-18 || 2020-06-24

Somerset McCaw || 8 || TPL STD || 2020-06-23 || 2020-06-24

Please Select which query you want to run:
(a) Πόσες κρατήσεις αντιστοιχούν σε κάθε κατηγορία δωματίων.
(b) Ποια κατηγορία δωματίων παρουσιάζει βάσει των κρατήσεων τον μεγαλύτερο τζίρο.
(c) Πόσα δωμάτια είναι προς το παρόν διαθέσιμα προς κράτηση.

Main x

Somerset McCaw || 7 || DBL STD || 2020-01-15 || 2020-01-17

Christos Katevas || 17 || QUAD STD || 2020-02-04 || 2020-02-06

Jecho Wailles || 13 || TPL STD || 2020-03-20 || 2020-03-25

Christos Katevas || 9 || DBL STD || 2020-03-21 || 2020-03-23

Janelle Poon || 2 || TPL STD || 2020-04-26 || 2020-05-01

George Ifantis || 3 || DBL DLX || 2020-05-04 || 2020-05-18

George Ifantis || 15 || TPL BUDGET || 2020-05-11 || 2020-05-30

Giannis Fotopoulos || 12 || QUAD DLX || 2020-06-18 || 2020-06-24

Somerset McCaw || 8 || TPL STD || 2020-06-23 || 2020-06-24

Please Select which query you want to run:
(a) Πόσες κρατήσεις αντιστοιχούν σε κάθε κατηγορία δωματίων.
(b) Ποια κατηγορία δωματίων παρουσιάζει βάσει των κρατήσεων τον μεγαλύτερο τζίρο.
(c) Πόσα δωμάτια είναι προς το παρόν διαθέσιμα προς κράτηση.
(d) Ποιες παροχές (facilities) διατίθενται σε συγκεκριμένα δωμάτια.
(e) Ποιοι επισκέπτες έχουν κράτηση αυτό το μήνα.
(f) Ποιο είναι το μέσο κέρδος ανά τύπο/κατηγορία δωματίου για τη θερινή σεζόν.
(g) Ποιος είναι ο πελάτης με τις περισσότερες κρατήσεις ανά εύρος ζώνης τιμής δωματίου.
(h) Ποια δωμάτια ανοικιάστηκαν από την ημερομηνία "X" έως και σήμερα.
OR WRITE 0 (NUMBER) TO EXIT!

0

Process finished with exit code 0

```

ΒΙΒΛΙΟΓΡΑΦΙΑ

Λήφθηκε βοήθεια και χρησιμοποιήθηκαν εργαλεία για την ολοκλήρωση της εργασίας από τους εξής συνδέσμους:

<https://www.mockaroo.com>

<https://www.postgresqltutorial.com/>

<https://stackoverflow.com/>

<https://www.postgresql.org/>