**PRACTICES (SQL)**

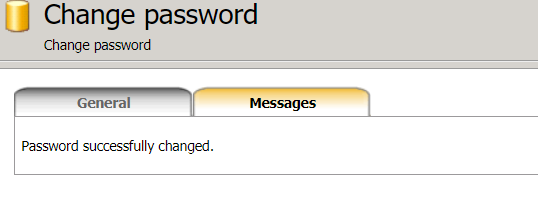
**Tutorial 1**

(no SQL questions)

**Tutorial 2**

**II. (1 mark) SQL – connect to the MyLittle SQL Server.**

1) Follow the instruction, connect to the Web based MyLittle SQL Server.SQL – connect to the MyLittle SQL Server. 2. Change the password under “tools” “change password”.

****

**III. SQL – table creation and deletion.**

**1)**

CREATE TABLE student (

studentNo numeric(8) NOT NULL,

firstName varchar(20) NOT NULL,

lastName VARCHAR(20) NOT NULL,

address varchar(30),

PRIMARY KEY (studentNo) );

CREATE TABLE staff (

staffNo numeric(8) NOT NULL,

firstName varchar(20) NOT NULL,

lastName VARCHAR(20) NOT NULL,

address varchar(30),

PRIMARY KEY (staffNo) );

CREATE TABLE unit (

unitNo numeric(8) NOT NULL,

unitName varchar(20) NOT NULL,

PRIMARY KEY (unitNo) );

**2)**

DROP TABLE student;

DROP TABLE staff;

DROP TABLE unit;

CREATE TABLE student (

studentNo numeric(8) NOT NULL,

firstName varchar(20) NOT NULL,

lastName VARCHAR(20) NOT NULL,

address varchar(30),

PRIMARY KEY (studentNo));

CREATE TABLE staff (

staffNo numeric(8) NOT NULL,

firstName varchar(20) NOT NULL,

lastName VARCHAR(20) NOT NULL,

address varchar(30),

PRIMARY KEY (staffNo));

CREATE TABLE unit (

unitNo numeric(8) NOT NULL,

unitName varchar(20) NOT NULL,

PRIMARY KEY (unitNo));

**Tutorial 3**

**II. (1 mark) SQL - Insert, delete, update, select.**

**a)**

create table room (

roomNo numeric(5) NOT NULL,

hotelNo varchar(30) NOT NULL,

type varchar(30) NOT NULL,

price numeric(7) NOT NULL,

guestNo numeric(8),

primary key (roomNo, hotelNo)

);

**b)**  Insert into room values (101,Null,'Double',100,10000);

Cannot insert the value NULL into column 'hotelNo'.

It is because the property of hotelNo is set as NOT NULL.

**c)** Result: Violation of PRIMARY KEY constraint, Cannot insert duplicate key.

It is because there is already a row having the same roomNo and hotelNo values.

**d)**  Cannot insert the value NULL into column 'hotelNo'.

It is because the property of hotelNo is set as NOT NULL.

**e)** The query is executed successfully and a new row is inserted.

**f)**  Row having roomNo. 202 is deleted.

**g)** Row having roomNo. 101 and hotelNo. Hilton is deleted.

**h)** Row having roomNo. 101 and hotelNo. Hilton is deleted.

**i)**  Truncate table room; (The table is truncated)

**j)**  update room set price = 500 where roomNo = 101 and hotelNo = 'Hilton';

**k)** update room set price = price \* 1.05;

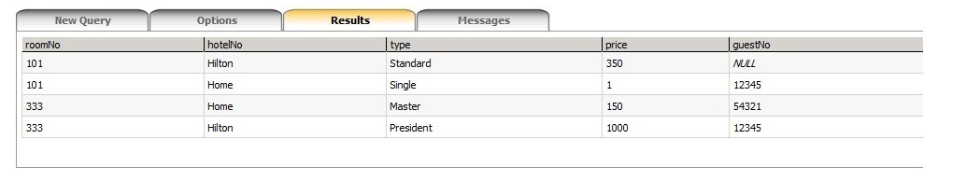
**l)**  select \* from room where hotelNo = 'Hilton';

**m)**  select roomNo, price from room where hotelNo = 'Hilton';

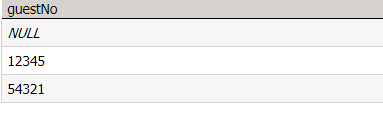
**Tutorial 4**

**II.**

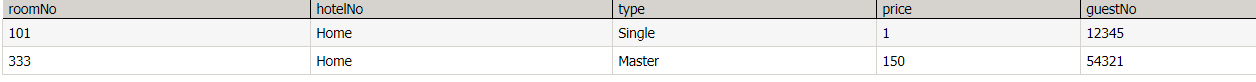
a) Again, work on the room table created in the last week with the records as follows:



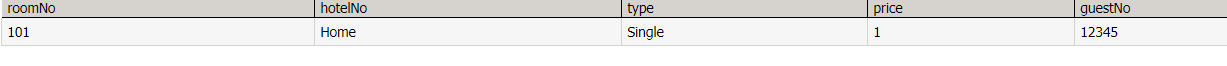
b) Select distinct guestNo from room;



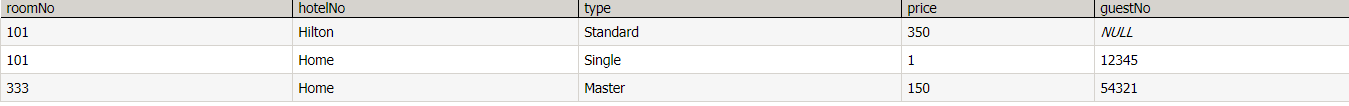
c) Select \* from room where hotelNo not like 'Hilton';



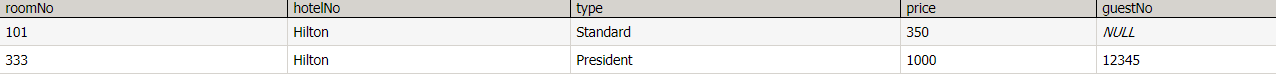
d) Select \* from room where hotelNo != 'Hilton' and roomNo != 333;



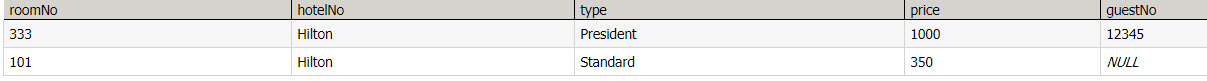
e) Select \* from room where hotelNo != 'Hilton' or roomNo != 333;



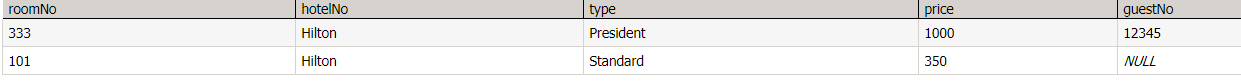
f) Select \* from room where hotelNo like 'Hilton' order by roomNo;



g) Select \* from room where hotelNo like 'Hilton' order by roomNo desc;



h) Select \* from room where price > 300 order by hotelNo, price desc;



i) alter table room add floor varchar(10);

j) alter table room drop column floor;

k) An error occured because hotelNo and roomNo together is a primary key. So, hotelNo has a

dependency with roomNo.

l) alter table room alter column price numeric(8,2);

m) create table guest (

guestNo integer,

guestname varchar(30),

address varchar(30)

);

n) alter table guest drop column guestNo;

alter table guest add guestNo integer;

o) Cannot define PRIMARY KEY constraint on nullable column in table 'guest'.

p) alter table guest alter column guestNo integer not null;

alter table guest add primary key(guestNo);

q) Cannot drop column because it is the primary key.

r) Error occured because the datatype to guestNo was not the same in both tables.

s) INSERT INTO guest VALUES (12345,'James Bond',' London');

INSERT INTO guest VALUES (54321,'Lady Gaga',null);

INSERT INTO guest VALUES (11111,'Super Man','Mars');

t) alter table room add foreign key(guestNo) references guest(guestNo);

u) Cannot delete due to referential integrity because guestNo is foreign key in the table room.

**Tutorial 5**

**II.**

**a)**

DROP table room;

create table room (

roomNo integer NOT NULL,

hotelNo varchar(30) NOT NULL,

hotelName varchar(30),

type varchar(30) NOT NULL default 'standard' check (type in('standard','single', 'double', 'master', 'president')),

price numeric(5) check(price between 0 and 99999),

guestNo integer

);

**b)**

ALTER table room add constraint PK\_room\_hotel PRIMARY KEY(roomNo, hotelNo);

**c)**

ALTER table room add Foreign Key (guestNo) REFERENCES guest(guestNo);

Invalid table; the table does not exist.

**d)**

CREATE TABLE guest (

guestNo integer PRIMARY KEY,

guestName varchar(30),

email varchar(100),

gender varchar(6) check (gender in ('male', 'female')),

nationality varchar(50),

Constraint CK\_gname\_email unique (guestName, email)

);

**e)** Table was altered successfully.

**f)**

Error occured because the check constraint was violated. The gender can be male or female.

**g)** The values were inserted.

**h)**

Violation of UNIQUE KEY constraint in column guestName and email. The duplicate key value is (James Bond, [jbond@security.gov.uk](mailto:jbond@security.gov.uk)).

This happened because the guestName and email has a unique constraint associated with them.

**i)** Cannot insert the value NULL into column 'guestNo' because its property is not null.

**j)** In the first insert statement email is null which is valid as the email doesn't have not null property.

In the second statement the gender and email are nullable so not inserting them does not violate any constraints.

**k)** Error inserting the value 1 50000.

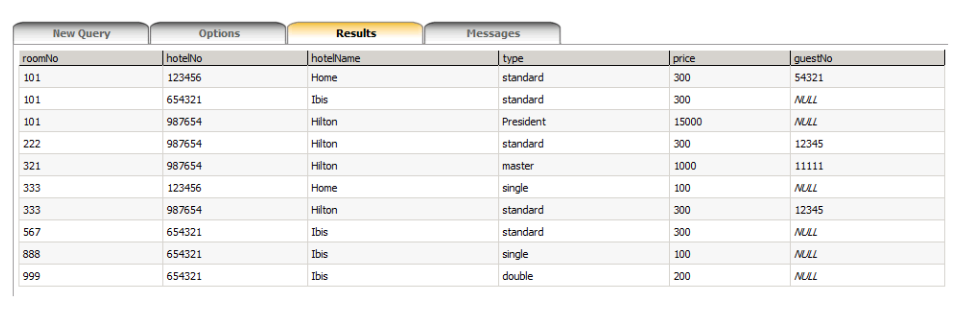
This error occured because there should not be a space in the integer value.

**l)** The INSERT statement conflicted with the FOREIGN KEY constraint of guestNo because there is no record with guestNo 12321 in the guest table.

**m)** The data is inserted because the guestNo column is nullable.

**n)** Default value of column type is inserted which is ‘standard’.

**o)**



INSERT INTO room (roomNo, hotelNo, hotelName, type, price, guestNo) VALUES (101, '123456', 'Home', 'standard', 300, 54321);

INSERT INTO room (roomNo, hotelNo, hotelName, type, price, guestNo) VALUES (101, '654321', 'Ibis', 'standard', 300, NULL);

INSERT INTO room (roomNo, hotelNo, hotelName, type, price, guestNo) VALUES (101, '987654', 'Hilton', 'President', 15000, NULL);

INSERT INTO room (roomNo, hotelNo, hotelName, type, price, guestNo) VALUES (222, '987654', 'Hilton', 'standard', 300, 12345)

INSERT INTO room (roomNo, hotelNo, hotelName, type, price, guestNo) VALUES (321, '987654', 'Hilton', 'master', 1000, 11111);

INSERT INTO room (roomNo, hotelNo, hotelName, type, price, guestNo) VALUES (333, '123456', 'Home', 'single', 100, NULL);

INSERT INTO room (roomNo, hotelNo, hotelName, type, price, guestNo) VALUES (333, '987654', 'Hilton', 'standard', 300, 12345);

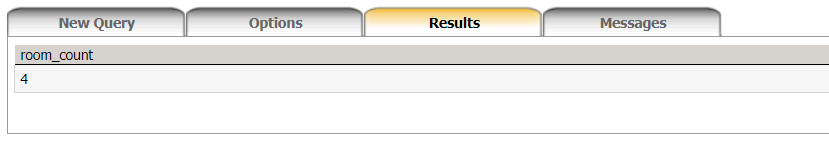
INSERT INTO room (roomNo, hotelNo, hotelName, type, price, guestNo) VALUES (567, '654321', 'Ibis', 'standard', 300, NULL);

INSERT INTO room (roomNo, hotelNo, hotelName, type, price, guestNo) VALUES (888, '654321', 'Ibis', 'single', 100, NULL);

INSERT INTO room (roomNo, hotelNo, hotelName, type, price, guestNo) VALUES (999, '654321', 'Ibis', 'double', 200, NULL);

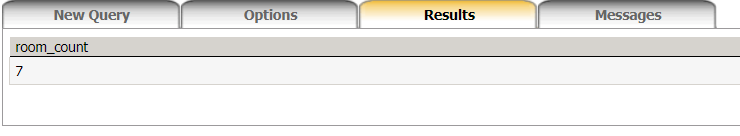
**p)**

select count(roomNo) as room\_count from room where hotelName = 'Hilton';



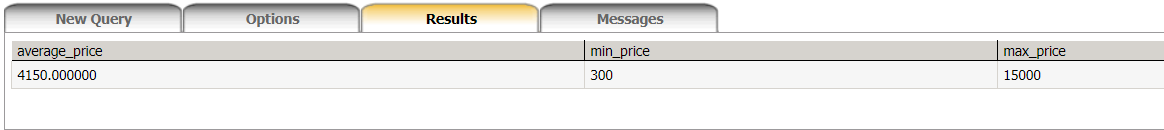
**q)**

select count (distinct roomNo) as room\_count from room;

****

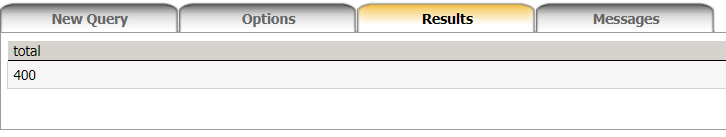
**r)**

select AVG(price) as average\_price, MIN(price) as min\_price, MAX(price) as max\_price from room where hotelName = 'Hilton';



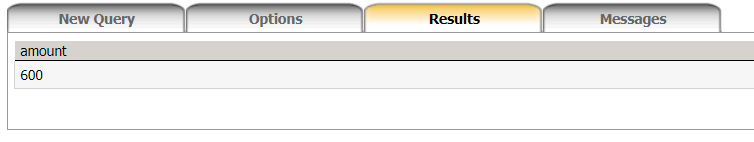
**s)**

select sum(price) as total from room where hotelName = 'Home';



**t)**

select sum(price) as amount from room where guestNo = 12345;

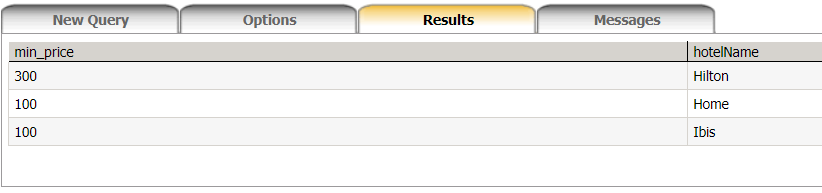


**u)**

No. Because the aggregate function cannot be used in where clause.

**v)**

select min(price) as min\_price, hotelName from room group by hotelName;

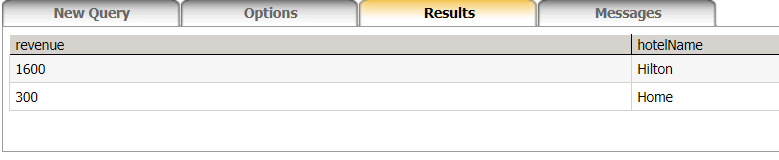


**w)**

select sum(price) as revenue, hotelName from room

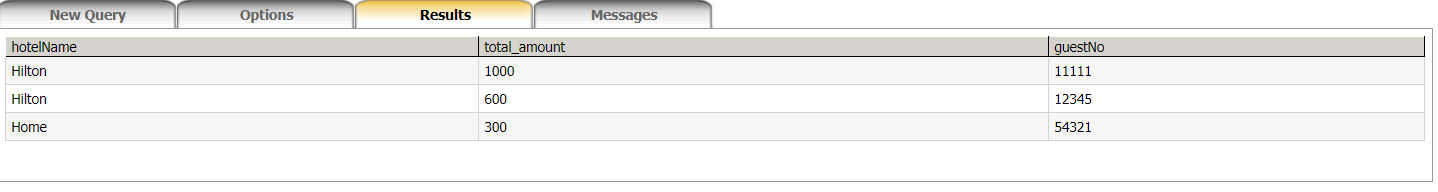
where guestNo is not null

group by hotelName;



**x)**

select hotelName, sum(price) as total\_amount, guestNo from room where guestNo is not null group by hotelName, guestNo;

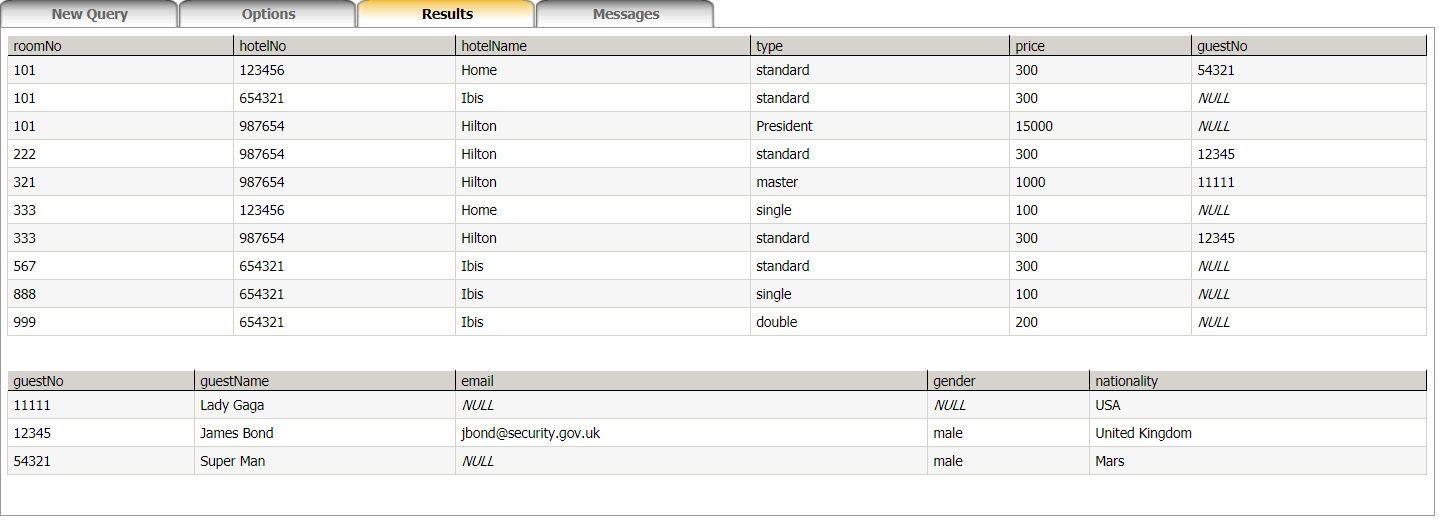


**Tutorial 6**

**II.**

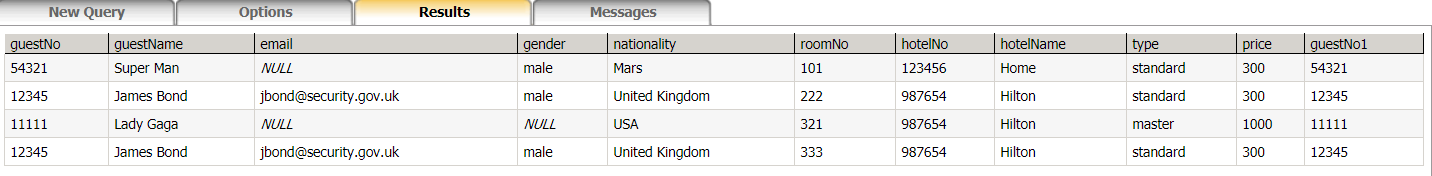
**a)**

Answer:



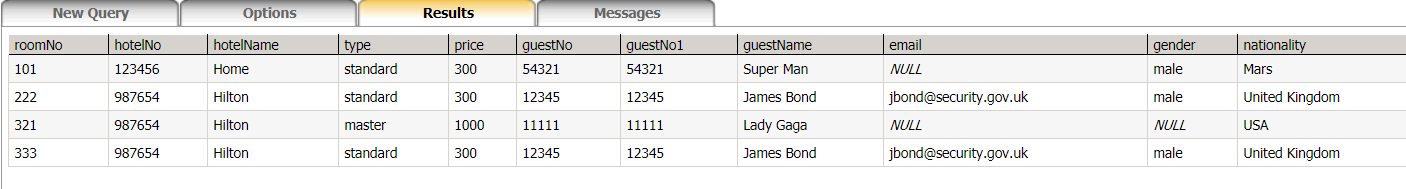
**b)**

SELECT \* FROM guest, room WHERE guest.guestNo = room.guestNo;



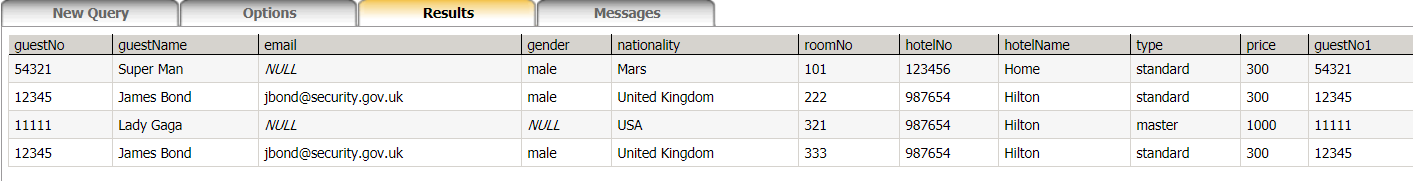
**c)**

SELECT \* FROM room r inner join guest g on r.guestNo = g.guestNo;

****

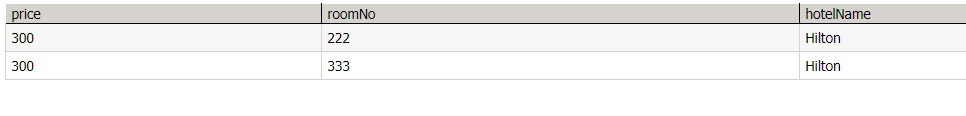
**d)**

SELECT \* FROM guest g, room r WHERE g.guestNo = r.guestNo;



**e)**

SELECT price, roomNo, hotelName FROM guest, room WHERE guest.guestNo = room.guestNo and guest.guestName = 'James Bond';

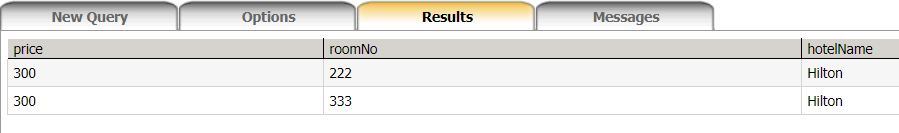


**f)**

SELECT r.price, r.roomNo, r.hotelName from room r

join guest g

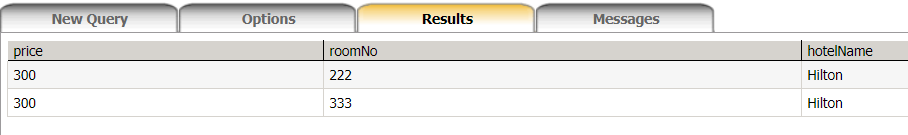
on r.guestNo = g.guestNo and g.guestName = 'James Bond';



**g)**

SELECT r.price FROM guest g, room r

WHERE g.guestNo = r.guestNo and g.guestName = 'James Bond';



**h)**

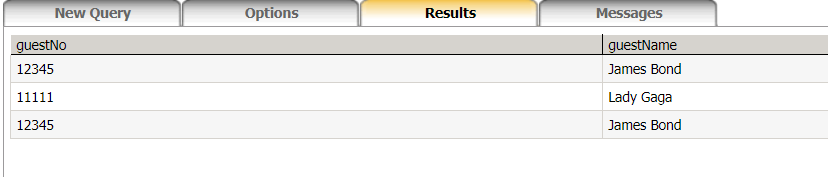
It gives an error saying “Ambiguous column name 'guestNo'.” It is because the column guestNo is present in both room and guest and it is not specified from which table the guestNo column is to be selected.

Correct statement is:

Select guest.guestNo, guestName From room, guest Where room.guestNo=guest.guestNo And hotelName='Hilton';

Or,

Select room.guestNo, guestName From room, guest Where room.guestNo=guest.guestNo And hotelName='Hilton';

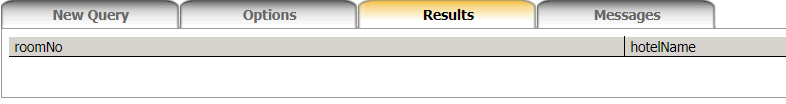


**i)**

select r.roomNo, r.hotelName from room r, guest g

where r.guestNo = g.guestNo and g.gender = 'male' and r.price > 500

order by r.price;



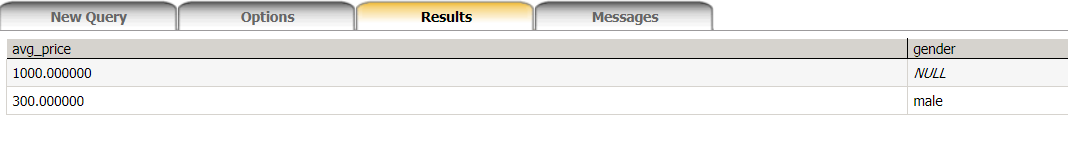
The result is empty because there are no rooms satisfying such conditions.

**j)**

SELECT avg(r.price) as avg\_price, g.gender from room r

join guest g on r.guestNo = g.guestNo

group by gender;

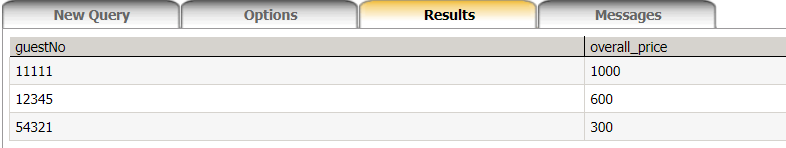


**k)**

select guestNo, sum(price) as overall\_price from room

where guestNo is not null

group by guestNo;

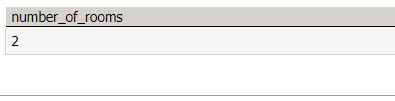


**l)**

select count(r.guestNo) as number\_of\_rooms from room r

join guest g on r.guestNo = g.guestNo

where g.guestName = 'James Bond';

****

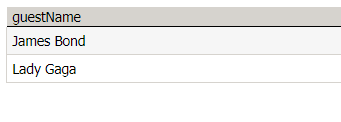
**m)**

select g.guestName from room r

join guest g on r.guestNo = g.guestNo

group by guestName

having sum(r.price) > 500;

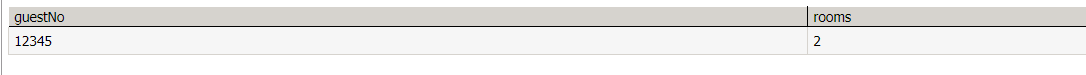


**n)**

select guestNo, count(guestNo) as rooms from room

group by guestNo

having count(guestNo) >=2;



**TUTORIAL 7**

**a)**

Select g.guestName, r.price from room r full outer join (select guestName, guestNo from guest ) g on (g.guestNo = r.guestNo) where r.guestNo is not null;

**b)**

Select g.guestName, sum(r.price)

from room r

full outer join guest g

on (g.guestNo = r.guestNo)

where r.guestNo is not null

group by guestName;

**c)**

create table staff (

staffNo int not null primary key,

staffName varchar(50) not null,

email varchar(50),

address varchar(50),

salary numeric(6)

);

**d)**

insert into staff (staffNo, staffName) values (11114567, 'Super Man');

insert into staff (staffNo, staffName) values (34533333, 'Super Man');

insert into staff (staffNo, staffName) values (98012456, 'Iron Man');

**e)**

select staffName as Name from staff

union

select guestName from guest;

**f)**

select staffName as Name from staff

intersect

select guestName from guest;

**g)**

select staffName from staff

except

select guestName from guest;

**h)**

select \* from room where price < (select max(price)/2 from room);

**i)**

select guest.guestName from room, guest

where guest.guestNo = room.guestNo

and price > (select avg(price) from room);

**j)**

SELECT guestName FROM guest,

(SELECT count(\*) AS rooms, guestNo FROM room GROUP BY guestNo) count

WHERE guest.guestNo = count.guestNo AND rooms > 3;