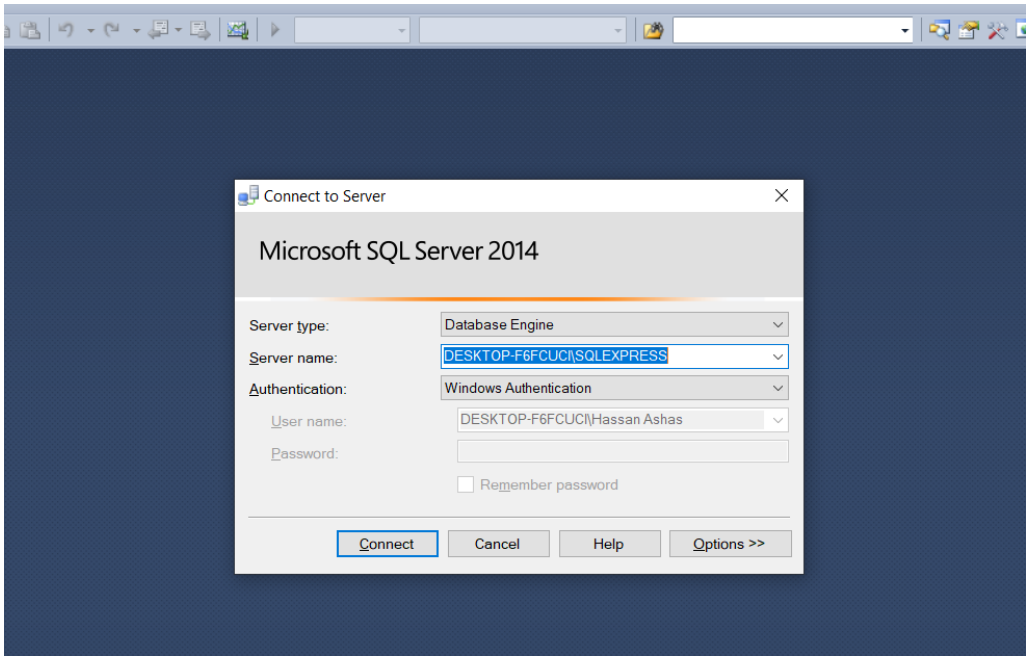
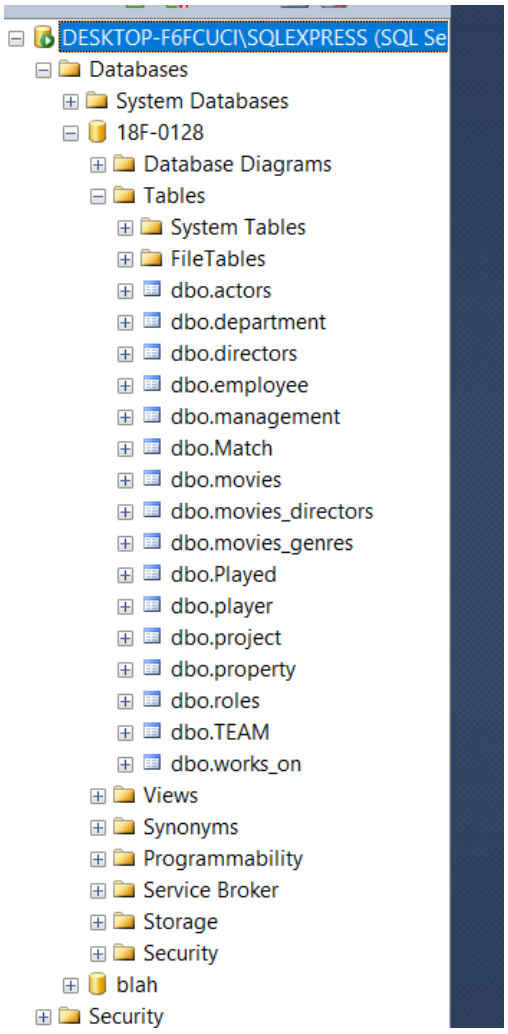
SQL Server and Database Screenshots

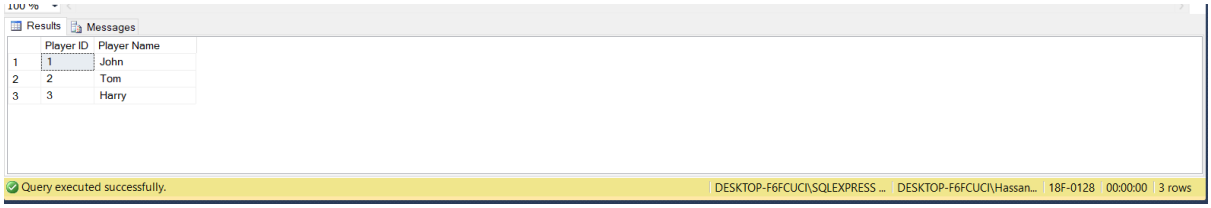




Question 1

SQL Query

```
select playerid 'Player ID', Nname 'Player Name'
from player where playerid
 = all (select playerid from played where Matchid
 = all (select matchid from Match where player.Team=match.Team1 or
player.Team=match.Team2))
```



Question 2

➔ Datatype of “yearlyProfit” is changed from varchar to integer, because avg command can not be used on varchar.

SQL Query

```
Select p_ID 'Property ID', p_owner 'Property Owner', p_location 'Property Location',
p_price 'Property Price'
from property where p_price > (select avg(yearlyProfit) from management);
```

Question 3

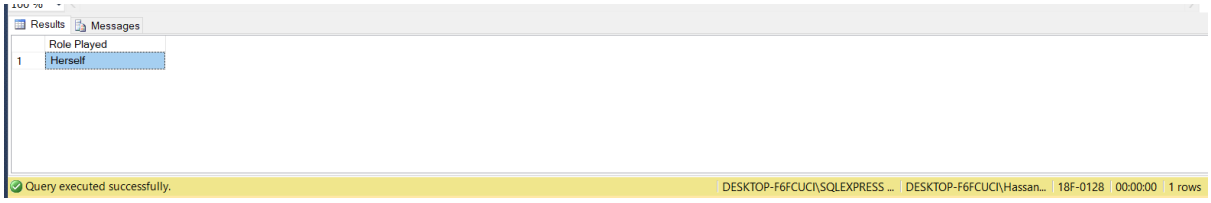
Table Creation and Insertion Queries

```
create table movies
(
id int NOT NULL Primary key,
name char(50),
myear int,
mrank int
);

insert into movies values(112290, 'Fight Club', 1999, 8.5);
insert into movies values(209658, 'Meet the Parents', 2000, 7);
insert into movies values(210511, ‘Pi’, 2000, 8.7);  -> I CHANGED THE NAME FROM
MOMENTO TO ‘PI’ FOR THIS PARTICULAR QUESTION.


create table actors
(
id int NOT NULL Primary key,
first_name char(50),
last_name char(50),
gender char,
film_count int
);

insert into actors values(433259, 'William', 'Shatner', 'M', 162);
insert ddinto actors values(797926, 'Britney', 'Spears', 'F', 65);
insert into actors values(831289, 'Sigourney', 'Weaver', 'F', 72);



create table directors
(
id int NOT NULL Primary key,
first_name char(50),
last_name char(50)
);

insert into directors values(24758Pi, 'Fincher');
insert into directors values(66965, 'Jay', 'Roach');
insert into directors values(72723, 'William', 'Shatner');


create table roles
(
actor_id int foreign key references actors(id),
movie_id int foreign key references movies(id),
role char(100)
);

insert into roles values (433259, 112290, 'Capt. James T. Kirk');
insert into roles values (433259, 209658, 'Sgt. T.J.Hooker');
insert into roles values (831289, 210511, 'Herself');



create table movies_directors
(
director_id int foreign key references directors(id),
movie_id int foreign key references movies(id)
```

```
);

insert into movies_directors values (24758, 112290);
insert into movies_directors values (66965, 209658);
insert into movies_directors values (72723, 210511);



create table movies_genres
(
movie_id int foreign key references movies(id),
genre char(30)
);

insert into movies_genres values (209658, 'Comedy');
insert into movies_genres values (210511, 'Action');
insert into movies_genres values (210511, 'Sci-Fi');
```

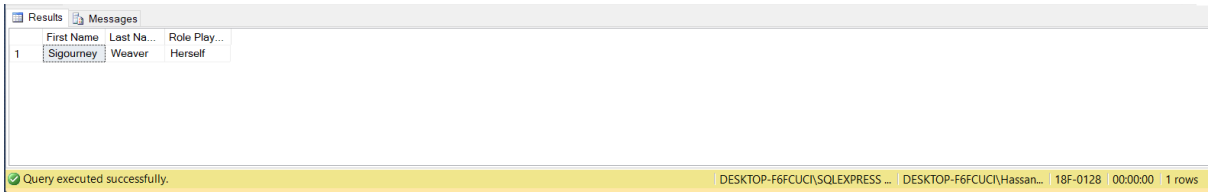
Roles in PI

```
SELECT role as 'Role Played'
FROM roles r, movies m
WHERE r.movie_id = m.id AND m.name = 'Pi';
```



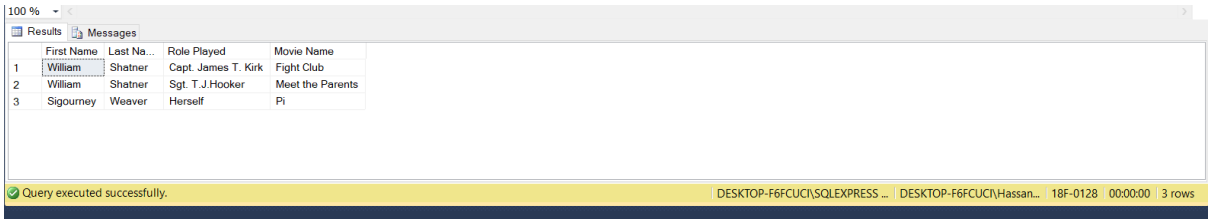
Actors in PI

```
SELECT first_name 'First Name', last_name 'Last Name', role 'Role Played'
FROM roles r, actors a, movies m
WHERE r.movie_id = m.id
AND r.actor_id = a.id
AND m.name = 'Pi';
```



Actors and Genres

```
SELECT DISTINCT a.first_name 'First Name', a.last_name 'Last Name', r.role 'Role
Played', m.name 'Movie Name'
FROM roles r, actors a, movies m, movies_genres g
WHERE r.movie_id = m.id
AND r.actor_id = a.id
AND (g.genre = 'Sci-Fi' OR g.genre = 'Horror')
ORDER BY a.last_name, a.first_name, m.name;
```



Question 4 – (A)

```
create trigger One_Row_Condition
after insert
on table_name
for each row
declare
total_count int;
begin
select count(*) into total_count from table_name;
if total_count > 1 then
delete from table_name where id = :new.id;
end if;
end;
```