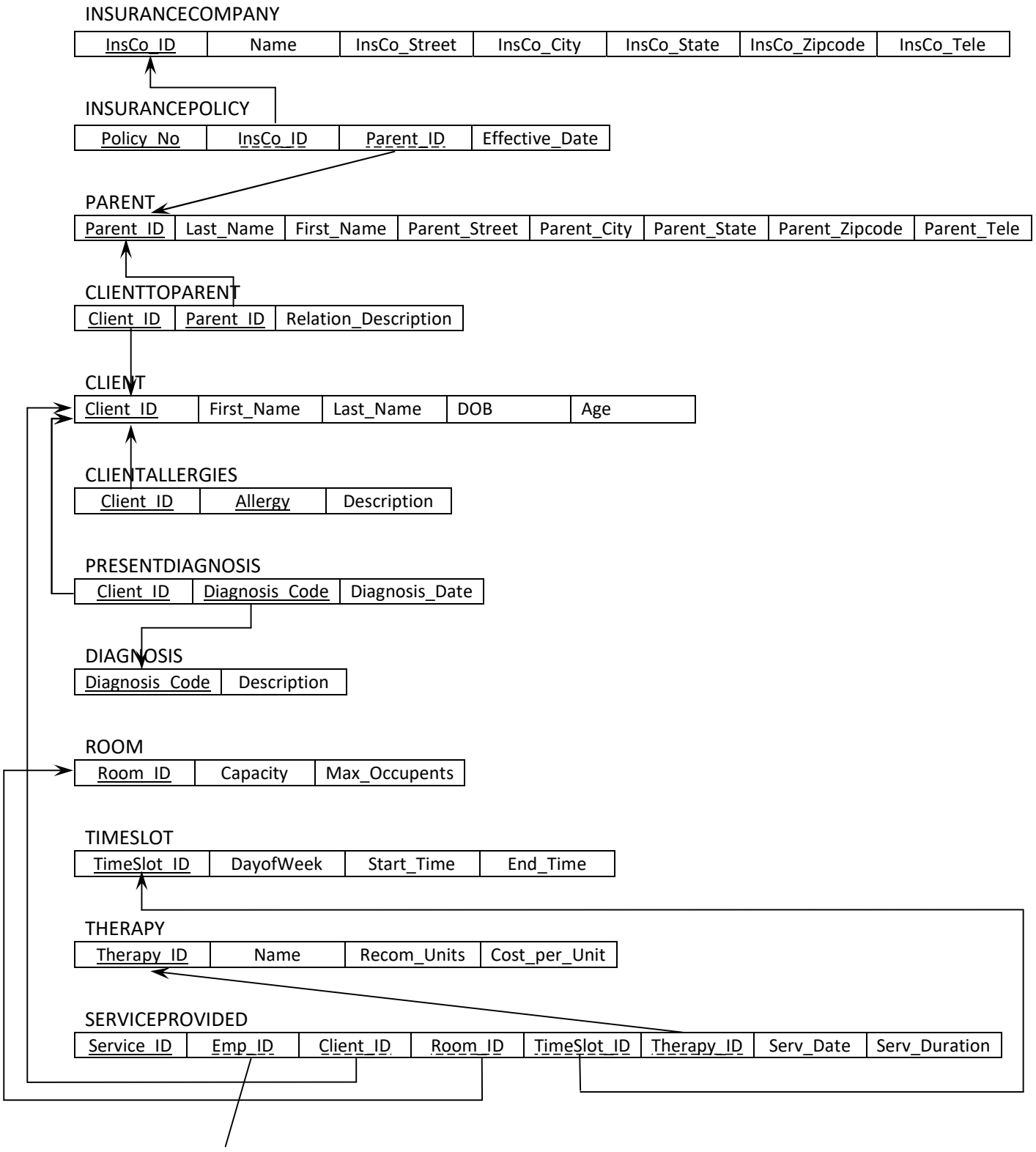
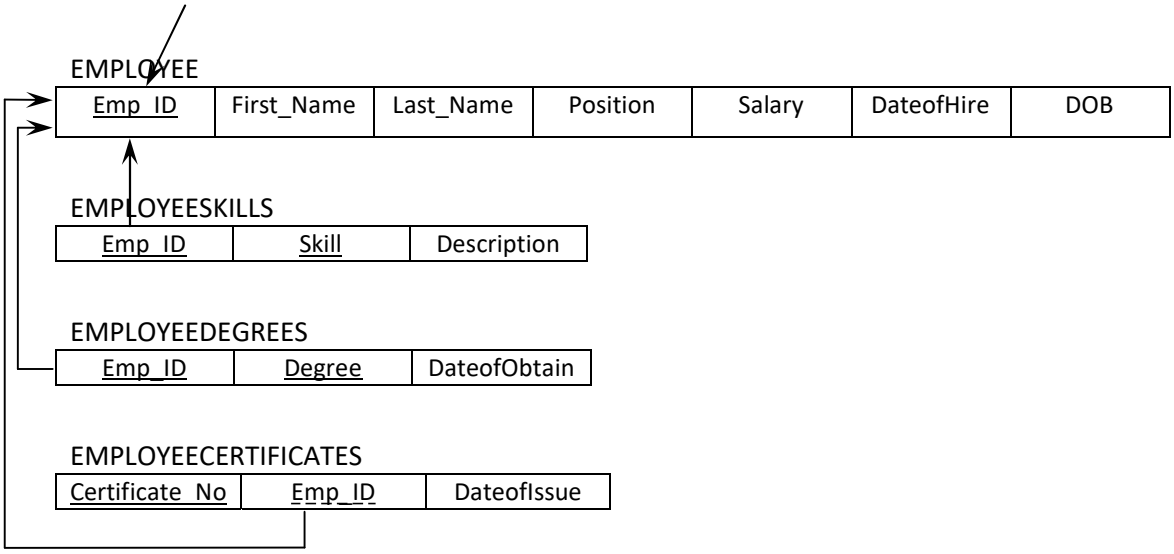
Final Exam of Database Design
Student: Lina Mi @377283

Part I: Database Design:

1) The normalized relations include the following:





2) Scripts that create the above tables are shown as following:

```
create table insurancecompany
(InsCo_ID varchar(8) primary key,
name varchar(30) not null,
InsCo_Street varchar(30),
InsCo_City varchar(20),
InsCo_State varchar(2),
InsCo_Zipcode varchar(10),
InsCo_Tele varchar(13));

 create table parent
(Parent_ID varchar(5) primary key,
Last_Name varchar(25) not null,
First_Name varchar(25) not null,
Parent_Street varchar(30),
Parent_City varchar(20),
Parent_State varchar(2),
Parent_Zipcode varchar(10),
Parent_Tele varchar(13));


create table insurancepolicy
(Policy_No varchar(5) primary key,
InsCo_ID varchar(8) references insurancecompany(InsCo_ID),
Parent_ID varchar(5) references parent(Parent_ID),
Effective_Date date);


create table client
(Client_ID varchar(5) primary key,
First_Name varchar(25) not null,
Last_Name varchar(25) not null,
DOB date,
Age numeric(3,1));
```

```
create table clienttoparent
(Client_ID varchar(5) references client(Client_ID),
Parent_ID varchar(5) references parent(Parent_ID),
Relation_Description varchar(40),
primary key(Client_ID, Parent_ID));


create table clientallergies
(Client_ID varchar(5) references client(Client_ID),
Allergy varchar(20),
Description varchar(40),
primary key(Client_ID, Allergy));


create table diagnosis
(Diagnosis_Code varchar(8) primary key,
Description varchar(40));


create table presentdiagnosis
(Client_ID varchar(5) references client(Client_ID),
 Diagnosis_Code varchar(8) references diagnosis(Diagnosis_Code),
 Diagnosis_Date date,
 primary key(Client_ID, Diagnosis_Code));

create table room
(Room_ID varchar(5) primary key,
Capacity numeric(4.2),
Max_Occupents int(2));


create table timeslot
(TimeSlot_ID varchar(5) primary key,
Datofweek varchar(4) check(Dateofweek in ('Mon','Tue','Wed','Thu','Fri', 'Sat','Sun')),
Start_Time varchar(10),
End_Time varchar(10));


create table therapy
(Therapy_ID varchar(5) primary key,
Name varchar(20) not null,
Recom_Unit int(2),
Cost_per_Unit numeric(6,2));


create table serviceprovided
(Service_ID varchar(5) primary key,
Emp_ID varchar(5) references employee(Emp_ID),
 Client_ID varchar(5) references client(Client_ID),
 Room_ID varchar(5) references room(Room_ID),
 TimeSlot_ID varchar(5) references timeslot(TimeSlot_ID),
```

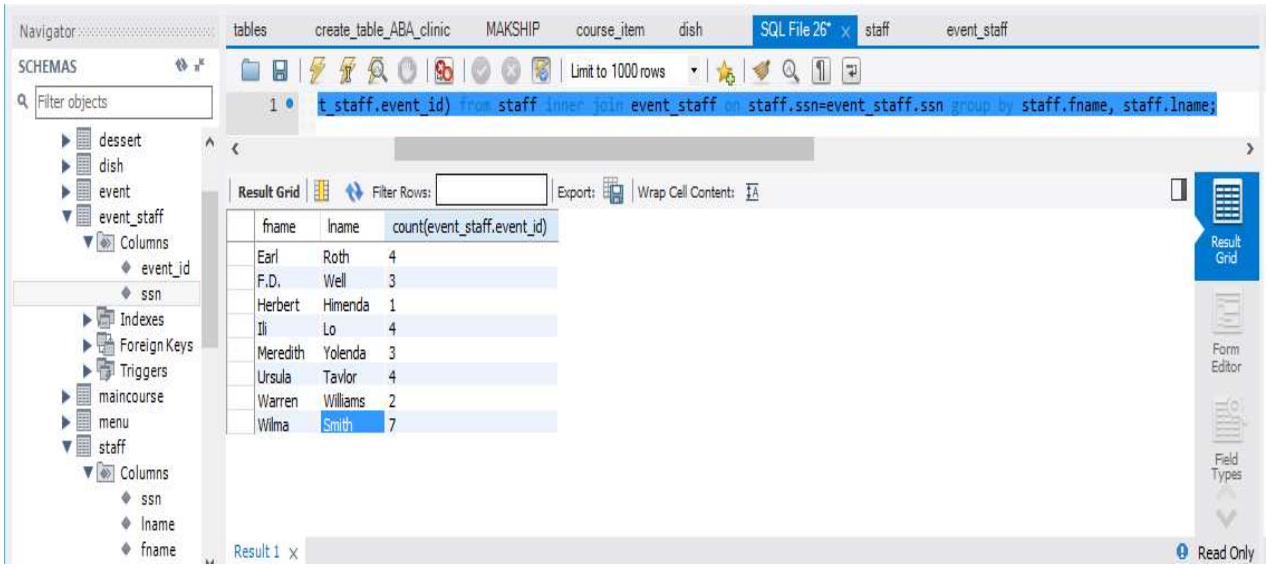
```
Therapy_ID varchar (5) references therapy(Therapy_ID),
Serv_Date date,
Serv_Duration int(2));


create table employee
(Emp_ID varchar(5) primary key,
 First_Name varchar(20) not null,
 Last_Name varchar(20) not null,
 Position varchar(8),
 Salary   numeric(10,2),
 DateofHire date,
 DOB date);


 create table empolyeeskills
 (Emp_ID varchar(5) references employee(Emp_ID),
  Skill  varchar(20),
  description varchar(40),
  primary key(Emp_ID, Skill));


  create table employdegree
  (Emp_ID varchar(5) references employee(Emp_ID),
  Degree varchar(8),
  DateofObtain date,
  primary key(Emp_ID, Degree));


  create table employcertificate
  (Emp_ID varchar(5) references employee(Emp_ID),
  Certificate_No varchar(8)   primary key,
  DateofIssue date);
```

Part II SQL – 50 points:

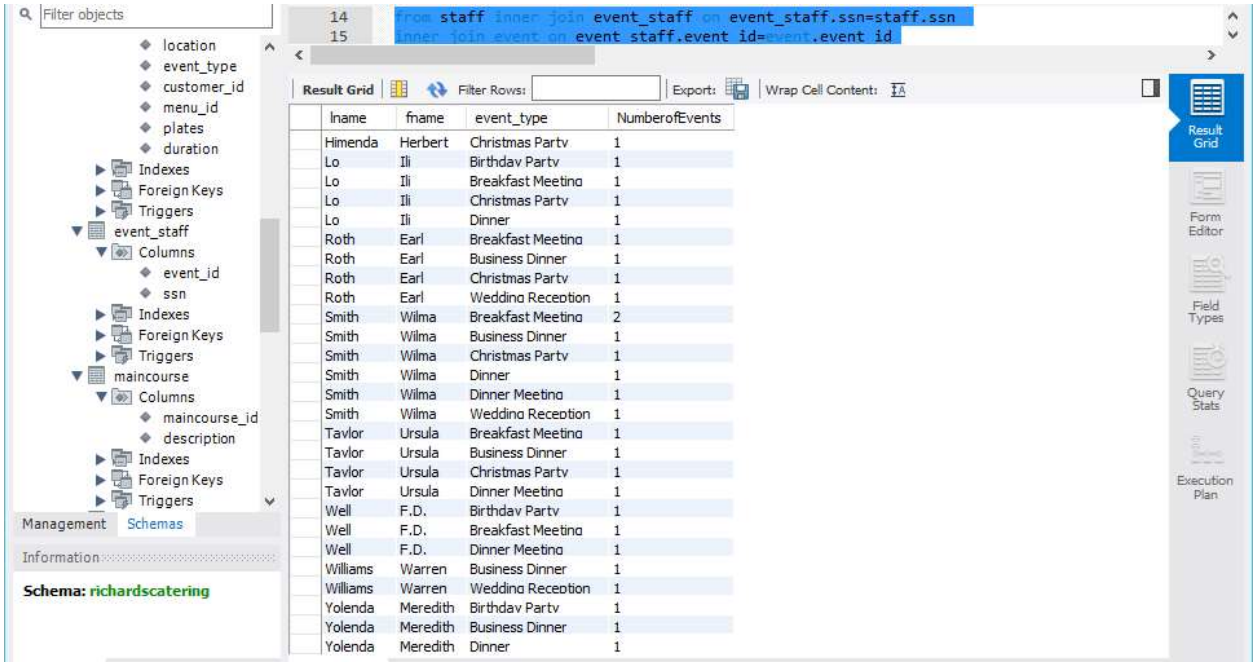
1) For each staff member, list the number of events that he or she has worked. Your output should include the last name, first name and the number of events.

   Script:
   ```
   select staff.fname, staff.lname, count(event_staff.event_id)
       from staff inner join event_staff
       on staff.ssn=event_staff.ssn
       group by staff.fname, staff.lname;
   ```

   Result:



2) For each staff member, list the number of each type of event that he or she has worked.your output should include the last name, first name, event_type and the number of events.
Script: *select staff.lname, staff.fname, event.event_type, count(*) as NumberofEvents*
from staff inner join event_staff on event_staff.ssn=staff.ssn
inner join event on event_staff.event_id=event.event_id
group by staff.lname, staff.fname, event.event_type
Result:

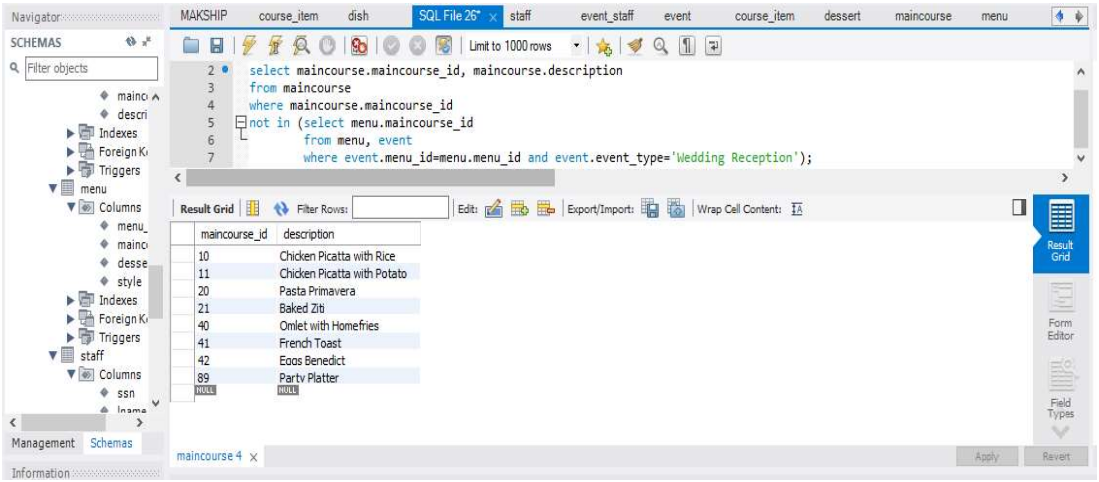


3) Find all maincourses that not been served at wedding receptions. List out the description
Script: *select maincourse.maincourse_id, maincourse.description*
from maincourse
where maincourse.maincourse_id
not in (select menu.maincourse_id

from menu, event

where event.menu_id=menu.menu_id and event.event_type='Wedding Reception');

Result:



4) List all dishes that have only been served at night (i.e. start hour >= 6) This the dish description.

Script: *select distinct dish.dish_id,dish.description*

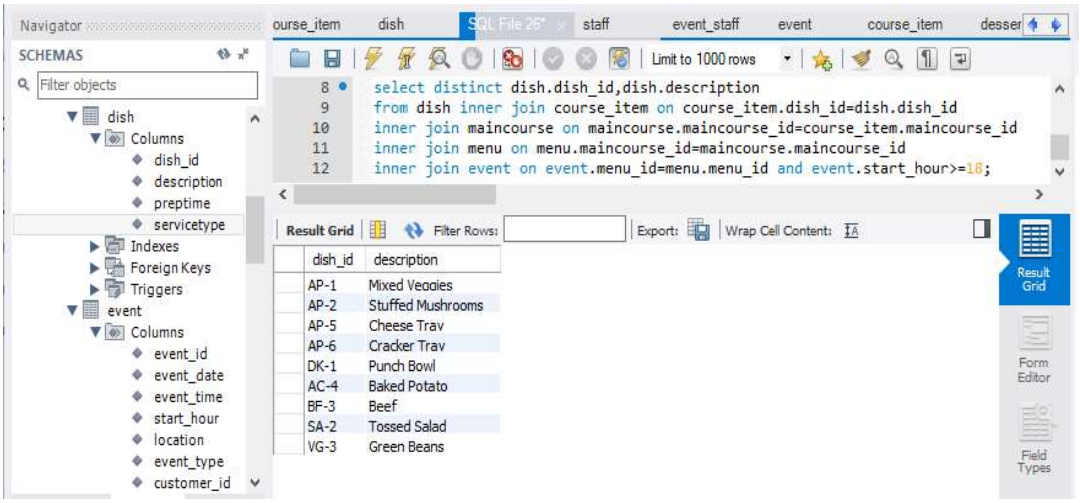
from dish inner join course_item on course_item.dish_id=dish.dish_id

inner join maincourse on maincourse.maincourse_id=course_item.maincourse_id

inner join menu on menu.maincourse_id=maincourse.maincourse_id

inner join event *on event.menu_id=menu.menu_id and event.start_hour>=18;*

Result:



5) Extra Credit: Calculate the total cost for each event

Script: *select event.event_id, event.event_type, sum(staff.hourly_rate*event.duration) as cost*

from event inner join event_staff on event.event_id=event_staff.event_id

inner join staff on staff.ssn=event_staff.ssn

group by event.event_id;

Result: