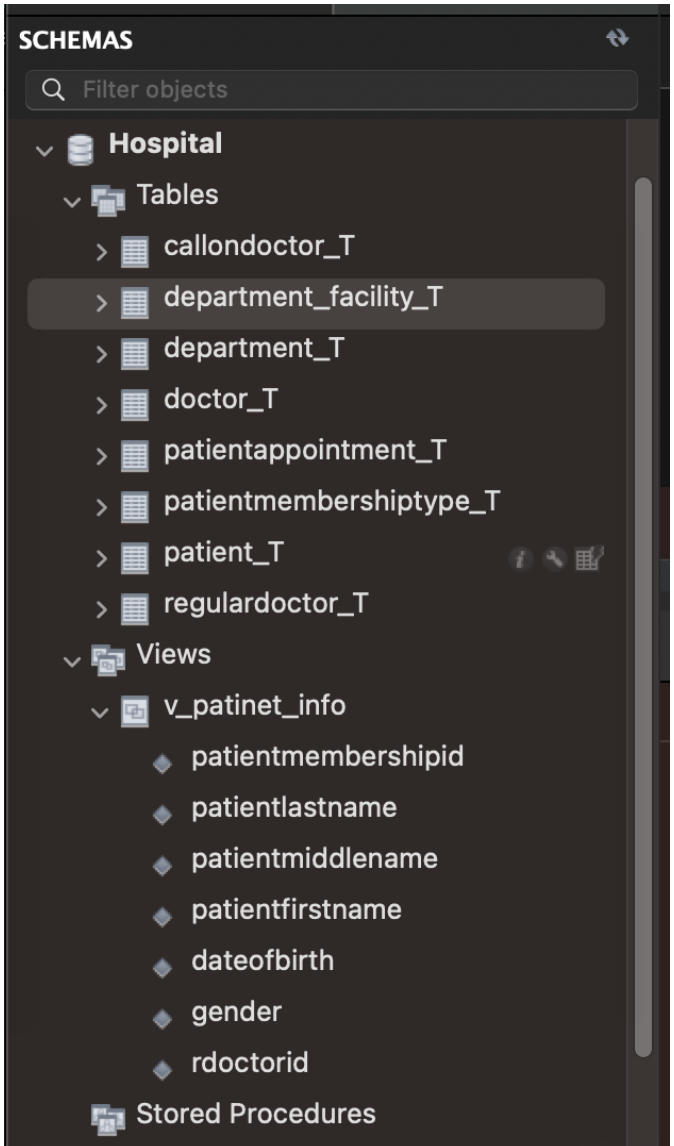
Assignment 4: SQL Implementation

Please use the Relational Data Model on the last page to implement a database.

1. Transform the relational model (attached) to SQL DDL, and implement in MySQL database, and generate a final screenshot of database and all the tables created under the schema;



2. Populate the tables with at least 5 rows per table. Note that rows in associative entity relations may not be 5 (could be more or fewer) since they are subject to the relationships between regular entity relations.

```
175    SELECT * FROM patientappointment_T;
176    SELECT * FROM patientmembershiptype_T;
```

75% 18:175

Result Grid | Filter Rows: Search | Edit: | Export/Import:

appointmentconfirm...	appointmentd...	appointmentlocati...	doctorname	rdoctor...	patientmembershi...
10	2022-09-01	20 Orchard Rd.	Tranquilli	203	1
20	2022-09-02	20 Orchard Rd.	Tranquilli	203	2
30	2022-09-03	20 Orchard Rd.	Ken Hurt	204	3
40	2022-09-04	20 Orchard Rd.	Ken Hurt	204	6
50	2022-09-05	20 Orchard Rd.	Johnathan Paine	205	7
NULL	NULL	NULL	NULL	NULL	NULL

```
175    SELECT * FROM patientappointment_T;
176    SELECT * FROM patientmembershiptype_T;
177    SELECT * FROM patient_T;
```

75% 28:176

Result Grid | Filter Rows: Search | Edit: | Export/Import:

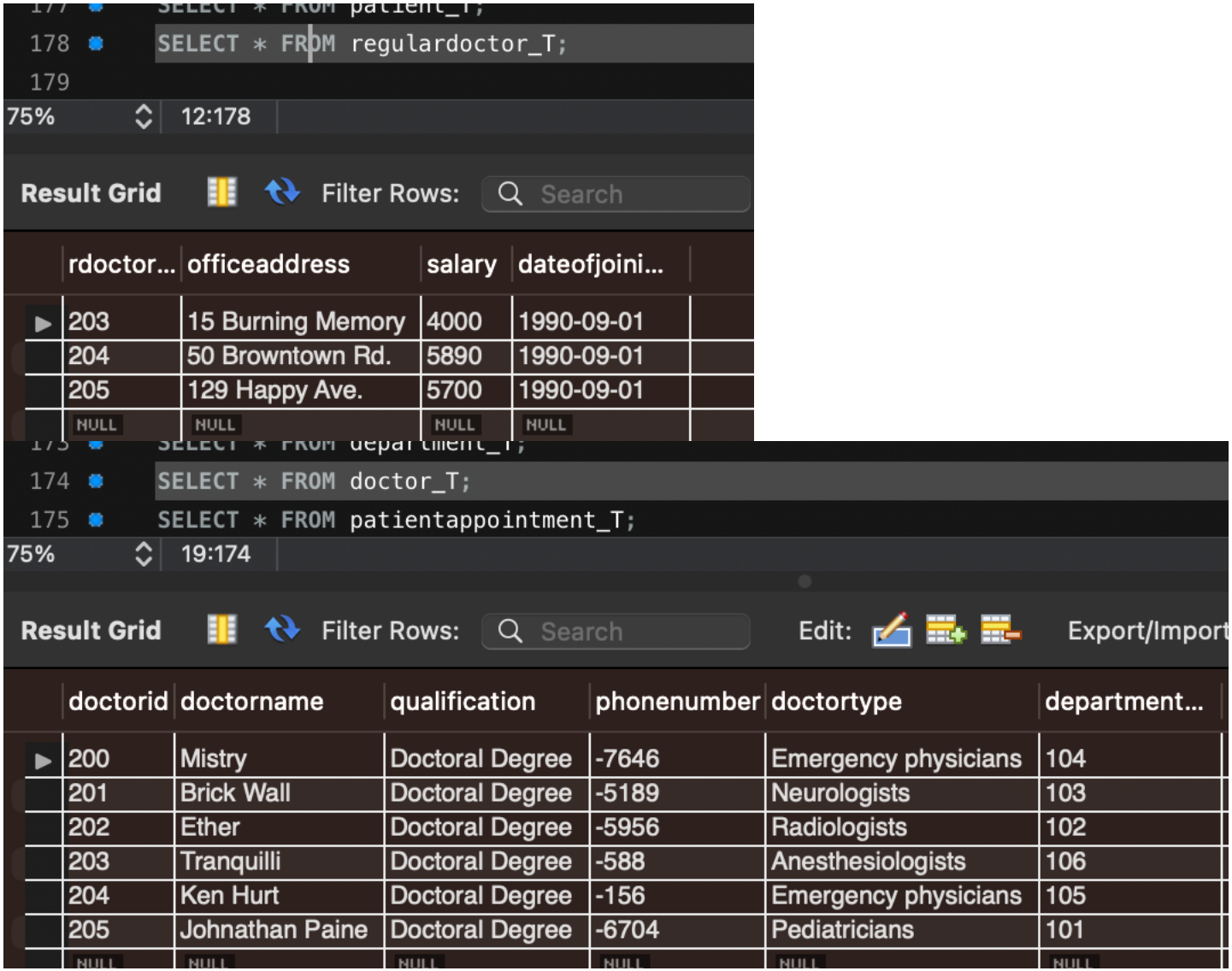
patientmembershipid	membershiptype
1	Gold
2	Bronze
3	Silver
4	Gold
5	Silver
6	Bronze

```
SELECT * FROM patientmembershiptype_T;
SELECT * FROM patient_T;
SELECT * FROM regulardoctor_T;
```

75% 19:177

Result Grid Filter Rows: Search Edit: Export/Import:

	patientmembershi...	patientlastna...	patientmiddlena...	patientfirstna...	dateofbirth	gender	rdoctor...
▶	1	Wesley	Aliana	Darwin	1997-09-01	Male	203
	2	Jayce	Allan	Bray	1997-02-01	Female	203
	3	Ether	Juana	Clarissa	1998-02-01	Male	203
	4	Tranquilli	Neeley	Rafferty	1998-03-01	Male	204
	5	Ken	Hasan	Bowen	1999-03-01	Female	204
	6	Pierce	Rosen	Lockwood	2000-03-01	Female	204
	7	Gardner	Greyson	Mercedez	2002-03-01	Male	205
	8	Bradshaw	Galindo	Benedict	2002-03-01	Male	205
	9	Fuller	Shivani	Garret	2002-03-01	Male	205
	10	Jasiah	Dupree	Stubblefield	1998-03-01	Male	204
	NULL	NULL	NULL	NULL	NULL	NULL	NULL

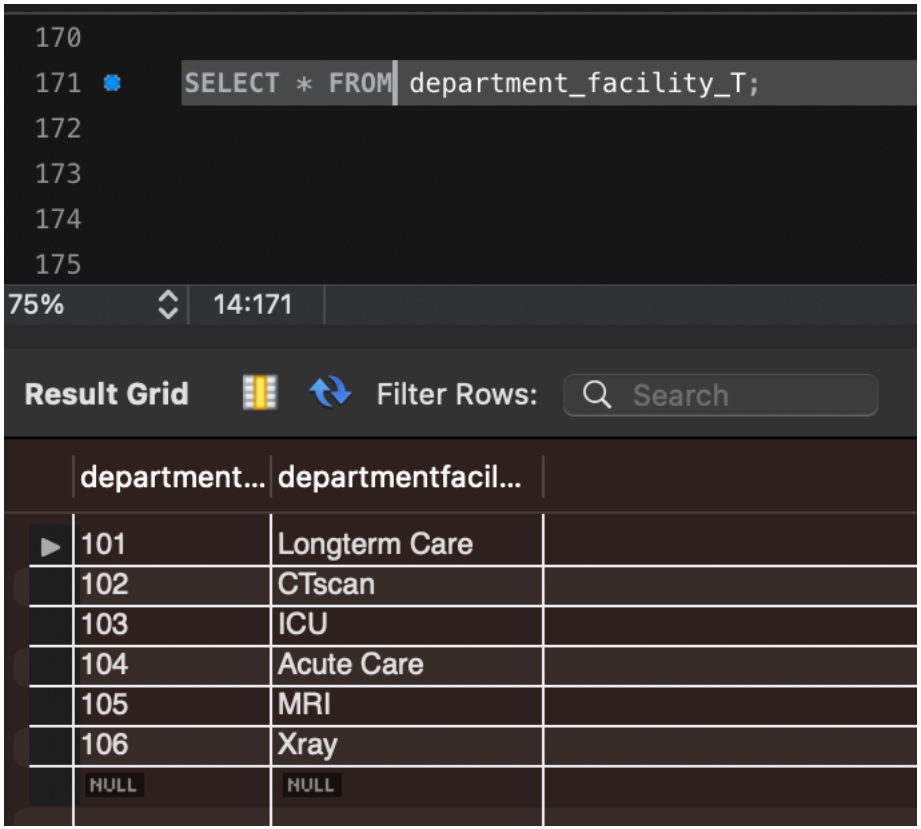

SELECT * FROM patient_T;
178 SELECT * FROM regulardoctor_T;
179
75%
12:178
Result Grid
Filter Rows:
Search
rdoctor... officeaddress salary dateofjoini...
203 15 Burning Memory 4000 1990-09-01
204 50 Browntown Rd. 5890 1990-09-01
205 129 Happy Ave. 5700 1990-09-01
NULL NULL NULL NULL
SELECT * FROM department_T;
174 SELECT * FROM doctor_T;
175 SELECT * FROM patientappointment_T;
75%
19:174
Result Grid
Filter Rows:
Search
Edit:
Export/Import
doctorid doctorname qualification phonenumber doctortype department...
200 Mistry Doctoral Degree -7646 Emergency physicians 104
201 Brick Wall Doctoral Degree -5189 Neurologists 103
202 Ether Doctoral Degree -5956 Radiologists 102
203 Tranquilli Doctoral Degree -588 Anesthesiologists 106
204 Ken Hurt Doctoral Degree -156 Emergency physicians 105
205 Johnathan Paine Doctoral Degree -6704 Pediatricians 101

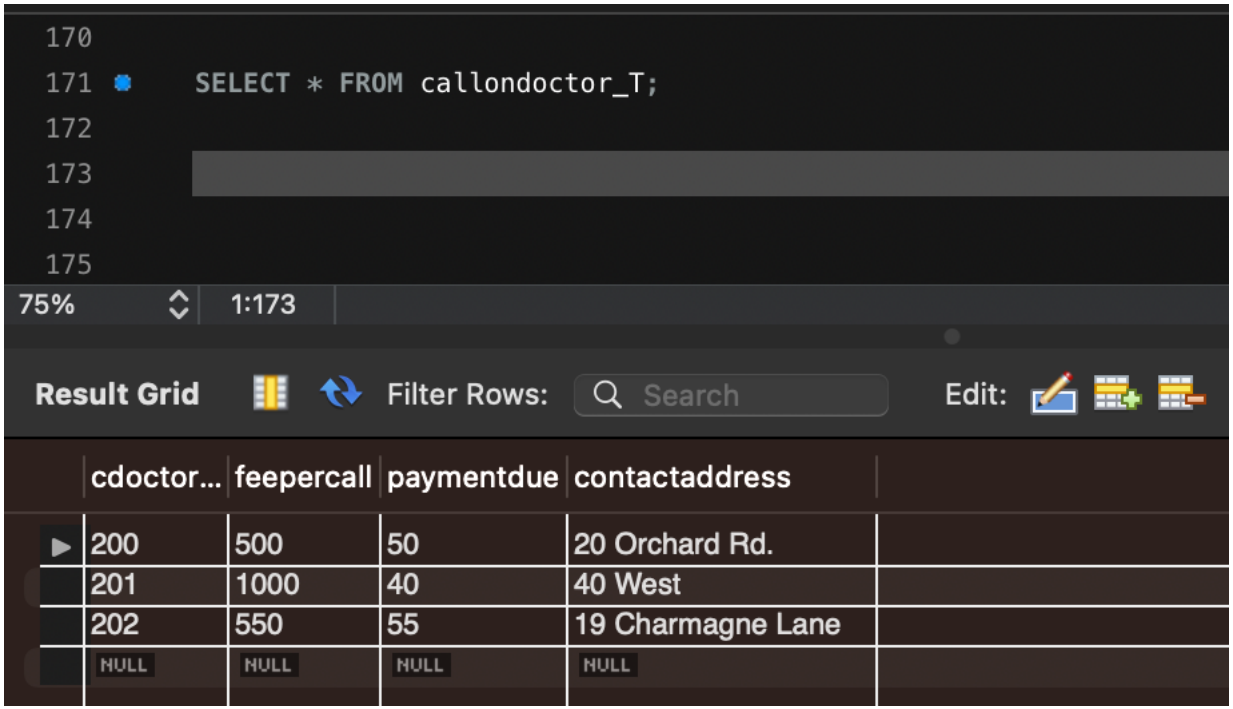
```
171    SELECT * FROM department_T;
172
```

75% 25:171

Result Grid Filter Rows: Search

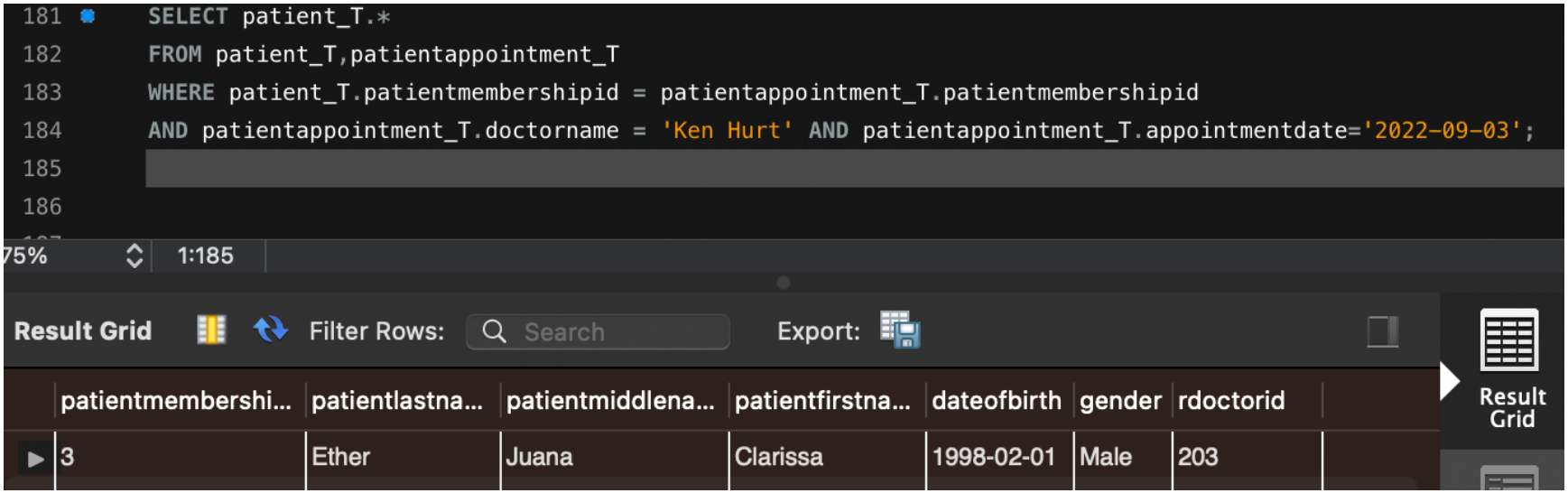
	department...	departmentna...	departmentlocation
▶	101	medicine	20 Orchard Rd.
	102	surgery	40 West
	103	gynaecology	19 Charmagne Lane
	104	obstetrics	302 Lakeland
	105	paediatrics	500 Coplin Ave.
	106	ENT	250 Rivendell Dr.


170
171
SELECT * FROM department_facility_T;
172
173
174
175
75%
14:171
Result Grid
Filter Rows:
Search
department...
departmentfacil...
101
Longterm Care
102
CTscan
103
ICU
104
Acute Care
105
MRI
106
Xray
NULL
NULL



3. List patient information for patients who have made appointment on an appointment date and with a particular doctor name (You decide on the values of any date and doctor name based on your inserted values). Use Equi-join.

```
SELECT patient_T.*
FROM patient_T,patientappointment_T
WHERE patient_T.patientmembershipid = patientappointment_T.patientmembershipid
AND patientappointment_T.doctorname = 'Ken Hurt' AND patientappointment_T.appointmentdate='2022-09-03';
```



4. List regular doctor information who's name starts with a letter 'J' and who work at a particular department name (You decide on the values of department name based on your inserted values). Use Inner Join.

SELECT doctor_T.*, regulardoctor_T.*
FROM regulardoctor_T INNER JOIN
doctor_T INNER JOIN department_T
ON regulardoctor_T.rdoctorid = doctor_T.doctorid
AND doctor_T.departmentid= department_T.departmentid
WHERE doctor_T.doctorname LIKE "J%" AND department_T.departmentname = "medicine" ;

```
186    SELECT doctor_T.*, regulardoctor_T.*
187    FROM regulardoctor_T INNER JOIN
188    doctor_T INNER JOIN department_T
189    ON regulardoctor_T.rdoctorid = doctor_T.doctorid
190    AND doctor_T.departmentid= department_T.departmentid
191    WHERE doctor_T.doctorname LIKE "J%" AND department_T.departmentname = "medicine" ;
192
193
194
195
196
```

75% 53:190

Result Grid Filter Rows: Search Export:

doctorid	doctorname	qualification	phonenumber	doctortype	department...	rdoctorid	officeaddress	salary	dateofjoining
205	Johnathan Paine	Doctoral Degree	-6704	Pediatricians	101	205	129 Happy Ave.	5700	1990-09-01

5. Do a LEFT JOIN on DEPARTMENT and DEPARTMENT_FACILITY tables.

```
SELECT *
FROM department_T LEFT JOIN department_facility_T
ON department_T.departmentid = department_facility_T.departmentid ;
```

```
193
194     /* Q5 */
195     SELECT *
196     FROM department_T LEFT JOIN department_facility_T
197     ON department_T.departmentid = department_facility_T.departmentid ;
198
199
200
```

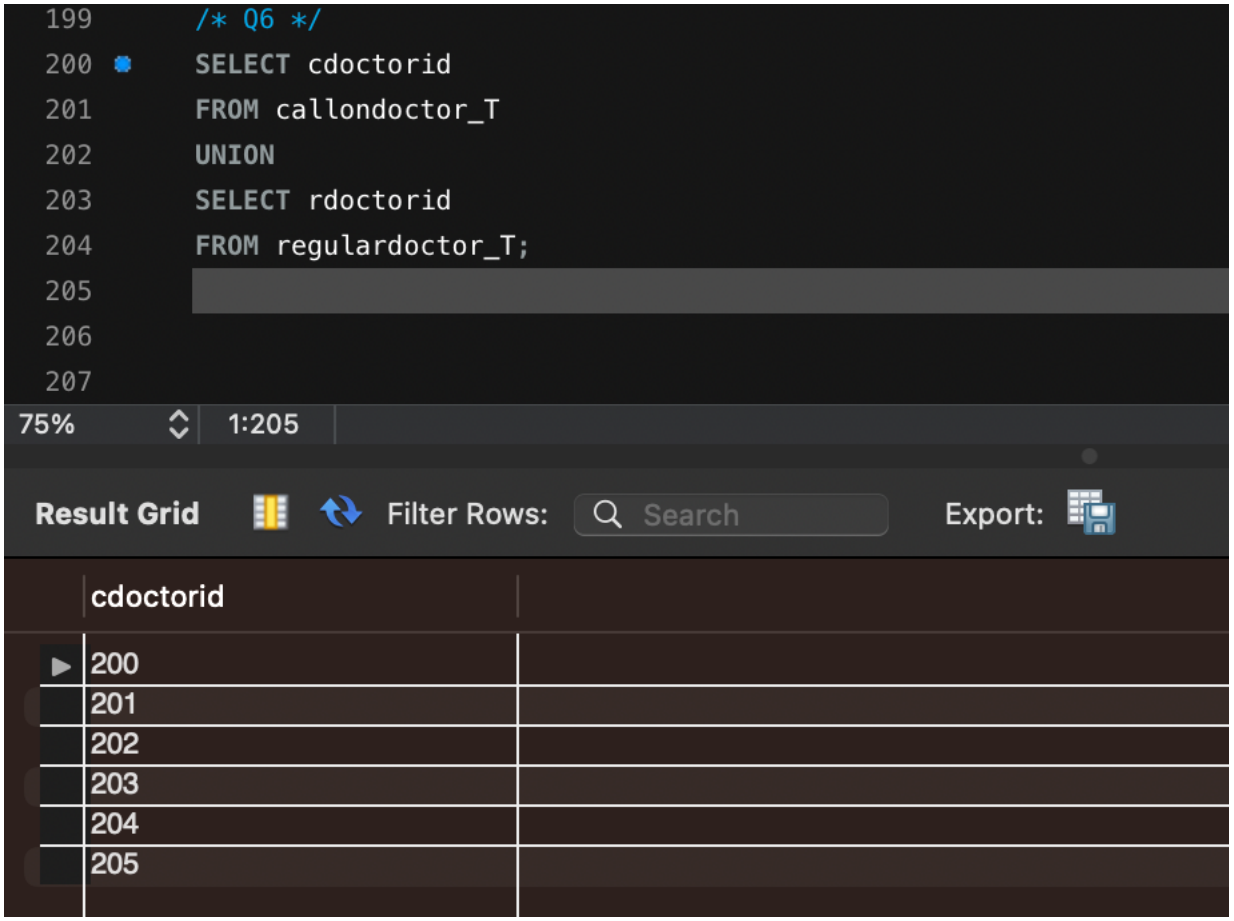
75% 49:197

Result Grid Filter Rows: Search Export:

departmentid	departmentname	departmentlocation	departmentid	departmentfacility
101	medicine	20 Orchard Rd.	101	Longterm Care
102	surgery	40 West	102	CTscan
103	gynaecology	19 Charmagne Lane	103	ICU
104	obstetrics	302 Lakeland	104	Acute Care
105	paediatrics	500 Coplin Ave.	105	MRI
106	ENT	250 Rivendell Dr.	106	Xray

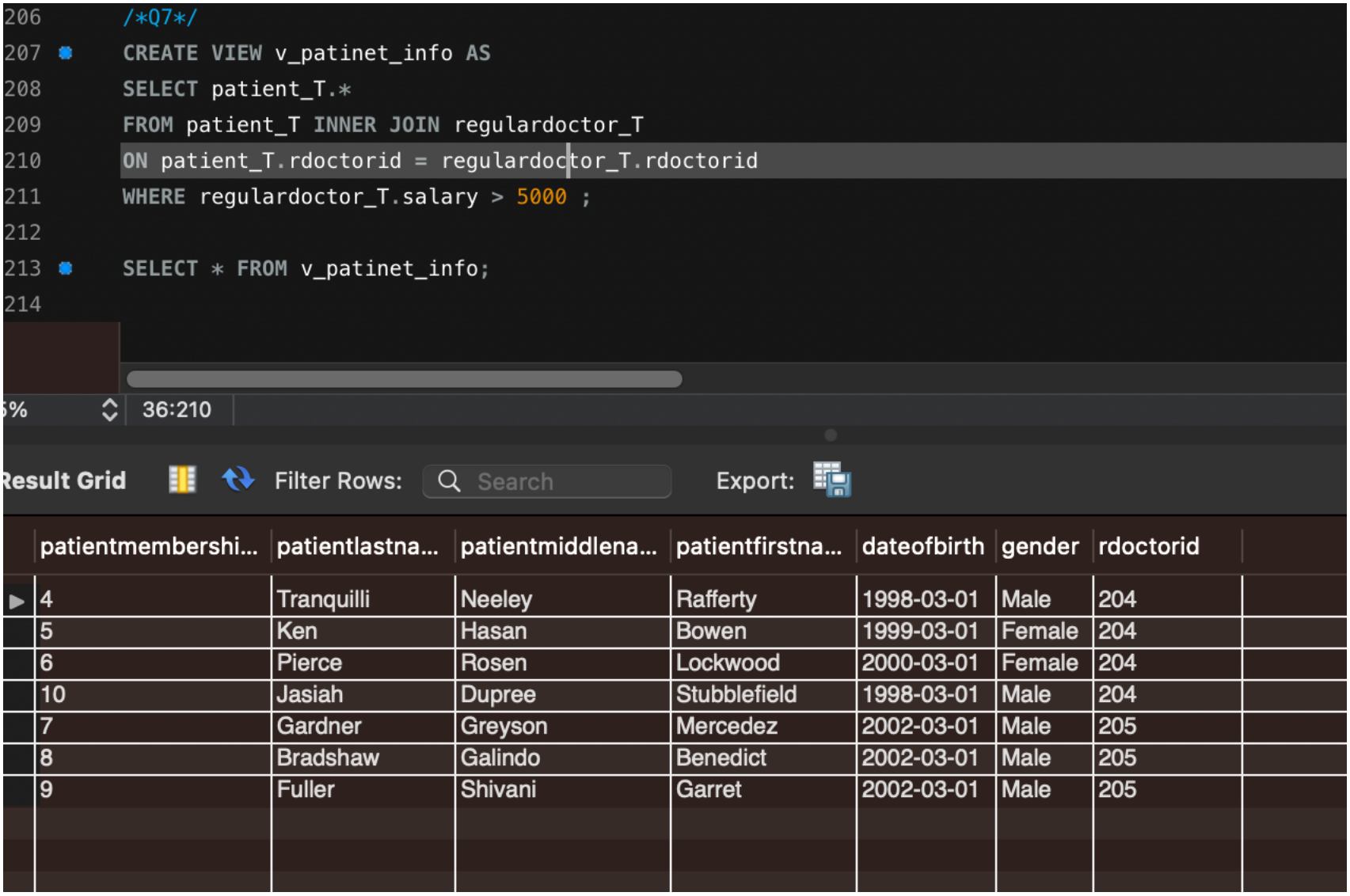
6. List doctorid from CALL_ON_DOCTOR and REGULAR_DOCTOR tables using UNION.

```
SELECT cdoctorid
FROM callondoctor_T
UNION
SELECT rdoctorid
FROM regulardoctor_T;
```

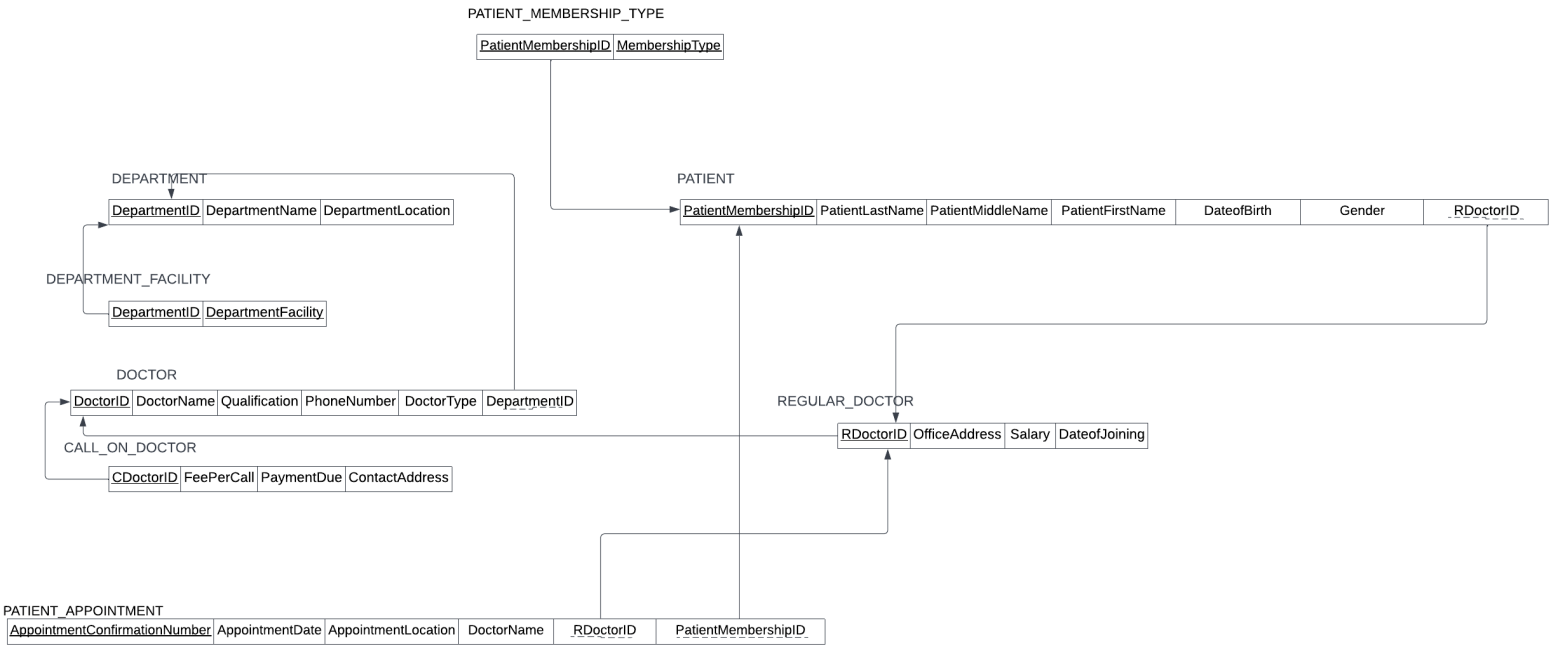


7. Create a dynamic VIEW for patient information whose caring doctor's salary is over $5000.

```
CREATE VIEW v_patinet_info AS
SELECT patient_T.*
FROM patient_T INNER JOIN regulardoctor_T
ON patient_T.rdoctorid = regulardoctor_T.rdoctorid
WHERE regulardoctor_T.salary > 5000 ;
```


206 /*Q7*/
207 CREATE VIEW v_patinet_info AS
208 SELECT patient_T.*
209 FROM patient_T INNER JOIN regulardoctor_T
210 ON patient_T.rdoctorid = regulardoctor_T.rdoctorid
211 WHERE regulardoctor_T.salary > 5000 ;
212
213 SELECT * FROM v_patinet_info;
214
36:210
Result Grid
Filter Rows:
Search
Export:
patientmembershi... patientlastna... patientmiddlena... patientfirstna... dateofbirth gender rdoctorid
4 Tranquilli Neeley Rafferty 1998-03-01 Male 204
5 Ken Hasan Bowen 1999-03-01 Female 204
6 Pierce Rosen Lockwood 2000-03-01 Female 204
10 Jasiah Dupree Stubblefield 1998-03-01 Male 204
7 Gardner Greyson Mercedez 2002-03-01 Male 205
8 Bradshaw Galindo Benedict 2002-03-01 Male 205
9 Fuller Shivani Garret 2002-03-01 Male 205

Submit both the .sql file and your word document file containing the working SQL DDL and DML queries output screenshots to Blackboard. Do note down any assumptions that you make if any such information is missing in the homework instructions.



DEPARTMENT

Attribute	Data Type	Size	AUTO	Null	Constraint
DepartmentID	Int	11	Yes	No	Primary
DepartmentName	varchar	30	No	No	
DepartmentLocation	Varchar	30	No	No	

DEPARTMENT_FACILTY

Attribute	Data Type	Size	AUTO	Null	Constraint

DepartmentID	Int	11			Part of Primary, Foreign
DepartmentFacility	varchar	30	No	No	Part of Primary

DOCTOR

Attribute	Data Type	Size	AUTO	Null	Constraint
DoctorID	Int	11	Yes	No	Primary
DoctorName	Varchar	30	No	No	
Qualification	Varchar	30	No	No	
PhoneNumber	Int	11	No	No	
DoctorType	Varchar	30	No	No	
DepartmentID	Int	11			Foreign

CALL_ON_DOCTOR

Attribute	Data Type	Size	AUTO	Null	Constraint
CDoctorID	Int	11	No	No	Primary, Foreign
FeePerCall	Double		No	No	
PaymentDue	Int	11	No	Yes	
ContactAddress	Varchar	30	No	Yes	

REGULAR_DOCTOR

Attribute	Data Type	Size	AUTO	Null	Constraint
RDoctorID	Int	11	No	No	Primary, Foreign
OfficeAddress	Varchar	30	No	Yes	
Salary	Double		No	No	
DateofJoining	Date		No	No	

PATIENT

Attribute	Data Type	Size	AUTO	Null	Constraint
PatientMembershipID	Int	11	Yes	No	Primary
PatientLastName	Varchar	30	No	Yes	
PateintMiddleName	Varchar	30	No	Yes	
PatientFirstName	Varchar	30	No	No	
DateofBirth	Date		No	No	
Gender	Varchar	11	No	Yes	
RDoctorid	Int	11			Foreign

PATIENT_MEMBERSHIP_TYPE

Attribute	Data Type	Size	AUTO	Null	Constraint
PatientMembershipID	Int	11			Part of Primary, Foreign
MembershipType	varchar	30	No	No	Part of Primary

PATIENT_APPOINTMENT

Attribute	Data Type	Size	AUTO	Null	Constraint
AppointmentConfirmationNumber	Int	11	Yes	No	Primary
AppointmentDate	Date		No	No	
AppointmentLocation	Varchar	30	No	No	
DoctorName	Varchar	30	No	No	
RDoctorid	Int	11			Foreign
PatientMembershipID	Int	11			Foreign