**Demonstrate Examples of the following queries within the context of your project**

* (10 points) Join query: Pick one query of this category, which joins at least two tables and performs a meaningful query, and provide an interface for the user to choose this query (eg. join the Customers and the Transactions table to find the phone numbers of all customers who has purchased a specific item).

***ALL Patient's appointments list so join Patient and Appointment.***

**https://troyforster76.uqcloud.net/apptlist.php**

SELECT

Appointment.AppDateTime,

Appointment.Length,

CONCAT(Patient.FirstName, ' ',Patient.LastName) AS 'Patient',

Patient.DOB,

Patient.Gender,

CONCAT(Clinician.FirstName,' ', Clinician.LastName) AS 'Clinician'

FROM Appointment LEFT JOIN Patient ON Appointment.PatientID=Patient.PatientID

LEFT JOIN Clinician ON Appointment.ClinicianID=Clinician.ClinicianID

ORDER BY AppDateTime

* (10 points) Division query: Pick one query of this category and provide an interface for the user to choose this query (eg. find all the customers who bought all the items). Prove that your division results would change based on the data in your database. You can do it either by inserting a new tuple into or deleting an existing tuple from the appropriate tables.

***Patient's appointments list using previous join and allow patient to be changed in the UI***

**https://troyforster76.uqcloud.net/showpatient.php**

SELECT

Appointment.AppDateTime,

Appointment.Length,

CONCAT(Patient.FirstName, ' ',Patient.LastName) AS 'Patient',

Patient.DOB,

Patient.Gender,

CONCAT(Clinician.FirstName,' ', Clinician.LastName) AS 'Clinician'

FROM Appointment LEFT JOIN Patient ON Appointment.PatientID=Patient.PatientID

LEFT JOIN Clinician ON Appointment.ClinicianID=Clinician.ClinicianID

WHERE Patient.PatientID=" . $varPID;

* (10 points) Aggregation query: Pick one query that requires the use of aggregation (min, max, average, or count are all fine). Rerun with at least one other example of aggregation.

***Count of Conditions for Patient***

[SELECT](https://troyforster76.uqcloud.net/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) DISTINCT Patient.PatientID, Patient.FirstName, Patient.LastName, [Count](https://troyforster76.uqcloud.net/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html)(\*) as 'No\_Conditions'

FROM Patient

[RIGHT](https://troyforster76.uqcloud.net/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-functions.html) JOIN Appointment ON Patient.PatientID=Appointment.PatientID

[RIGHT](https://troyforster76.uqcloud.net/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-functions.html) JOIN Diagnosed ON Appointment.AppointmentID=Diagnosed.AppointmentID

[LEFT](https://troyforster76.uqcloud.net/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-functions.html) JOIN Conditions ON Diagnosed.ConditionID= Conditions.ConditionID

GROUP BY Patient.PatientID, Patient.FirstName, Patient.LastName

**https://troyforster76.uqcloud.net/no\_conditions.php**

***Avg age of Patient***

SELECT

AVG(DATEDIFF(CURRENT\_DATE(),DOB)/365.25)

FROM

Patient

**https://troyforster76.uqcloud.net/avg\_age.php**

* (20 points) Nested aggregation with group-by: Pick one query that finds some aggregated value for each group (eg. the average number of items purchased per customer) and then return with a different aggregation. Provide an interface for the user to specify whether the average, min, max or count is requested.

***The sickest person I.e. Max of Count of Conditions for Patient***

[SELECT](https://troyforster76.uqcloud.net/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM ( [SELECT](https://troyforster76.uqcloud.net/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) DISTINCT Patient.PatientID, Patient.FirstName, Patient.LastName, [Count](https://troyforster76.uqcloud.net/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html)(\*) as 'No\_Conditions'

FROM Patient

[RIGHT](https://troyforster76.uqcloud.net/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-functions.html) JOIN Appointment ON Patient.PatientID=Appointment.PatientID

[RIGHT](https://troyforster76.uqcloud.net/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-functions.html) JOIN Diagnosed ON Appointment.AppointmentID=Diagnosed.AppointmentID

[LEFT](https://troyforster76.uqcloud.net/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-functions.html) JOIN Conditions ON Diagnosed.ConditionID= Conditions.ConditionID

GROUP BY Patient.PatientID, Patient.FirstName, Patient.LastName

) AS RESULTS WHERE RESULTS.No\_Conditions >=ALL ( [SELECT](https://troyforster76.uqcloud.net/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) DISTINCT [Count](https://troyforster76.uqcloud.net/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html)(\*) as 'No\_Conditions'

FROM Patient

[RIGHT](https://troyforster76.uqcloud.net/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-functions.html) JOIN Appointment ON Patient.PatientID=Appointment.PatientID

[RIGHT](https://troyforster76.uqcloud.net/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-functions.html) JOIN Diagnosed ON Appointment.AppointmentID=Diagnosed.AppointmentID

[LEFT](https://troyforster76.uqcloud.net/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-functions.html) JOIN Conditions ON Diagnosed.ConditionID= Conditions.ConditionID

GROUP BY Patient.PatientID, Patient.FirstName, Patient.LastName

)

**https://troyforster76.uqcloud.net/no\_conditions\_max.php**

* (10 points) Delete operation: Implement a cascade-on-delete situation. Provide an interface for the user to specify some input for the deletion operation. Some input values would fail the cascade specification but others would successfully follow the cascade specification. Provide an interface for the user to display the relation instance after the operation.

***Delete a patient record (and associated emergency contacts) but only if they have no appointment history.***

DELETE FROM Patient

WHERE PatientID = $varPID

[**https://troyforster76.uqcloud.net/delpatient.php**](https://troyforster76.uqcloud.net/delpatient.php)

* (10 points) Update operation: Implement a constraint using the check statement. Provide an interface for the user to specify some input for the update operation. Some input values would successfully satisfy a constraint while others would fail. Provide an interface for the user to display the relation after the operation. Note that MySQL 5.5 does not support the Check constraint. If you are using MySQL 5.5, keep the check in the SQL statement, but actually do the check at the application level (and comment it in the code).

***Update Last Name***

**https://troyforster76.uqcloud.net/updatepatient.php**