**CSD 2204 Database Design and SQL - Case Study/ Term Project**

**Case Study: Design and Implementation of a Doctors’ Clinic Database**

**Business Rules**:

1. The clinic has several doctors
2. Each doctor has a speciality (GP, Cardiology, Pediatrics, Children, etc...)
3. Each patient has a primary doctor
4. Each patient may have several prescriptions (may take several medications)
5. If the patient's primary is not available, then any available doctor might write the prescription to this patient
6. For the Term Project, add the pharmacy's functionality as follow:
   1. The clinic has only one pharmacy
   2. The pharmacy may has one and only one pharmacist
   3. The patient may purchase the medication(s) given in the doctor's prescription(s) from the clinic's pharmacy.

Specify any assumptions you may need to take if you need to

**Requirements**

Stage 1:

1. Identify Entities (Doctor, Specialty, Patient, Prescription, Medication, etc….)
2. Provide the adequate attributes in each entity
3. Clearly identify the relationships between entities, specifying each relationship name
4. Clearly specify the maximum and minimum cardinalities in each relationship
5. All entities and relationships must be at least in the 3rd Normal Form

Stage 2:

1. Build the relational database model for the finalized E-R diagram
2. Using Ms-Access transfer the relation database model into a physical database, this would include:
   1. The specification and selection the appropriate data type and size for each field
   2. The implementation of the primary keys, foreign keys
   3. The creation of the appropriate indexes (if necessary)
3. Populate each table with an adequate number of records (entity instances). At least 25 records in each table. You may use a less number of records in the speciality table not below 7 records
4. Using SQL build the following views (or reports):
   1. The clinic prints and distributes newsletters for its patients who are actively visiting the clinic. Active patients are identified by those who visit the clinic at least twice in the past year

SELECT p.id, p.name, p.phone, p.address, Count(p.id) as counter\_visits

FROM patient\_profile as pp INNER JOIN appointment as ap ON pp.appointment\_id = ap.id INNER JOIN patient as p ON ap.patient\_id = p.id

where pp.date\_time >= DATE\_SUB(NOW(), INTERVAL 1 YEAR) GROUP BY p.id HAVING counter\_visits >= 2;

* 1. List the number of patients in each medical speciality
     + Number of doctors in each medical speciality:

SELECT spec.name, COUNT(spec.id) as counter\_doctor from doctor as doc INNER JOIN speciality as spec ON doc.speciality\_id = spec.id GROUP BY spec.id

* 1. List the doctors who work in the clinic sorted in descending order by the number of visits from patients in each month

SELECT doc.id as doctor\_id, doc.name as doctor\_name, COUNT(app.doctor\_id) as patient\_counter, MONTH(app.date\_time) as month, MONTHNAME(app.date\_time) as month\_name FROM appointment as app INNER JOIN doctor as doc ON doc.id = app.doctor\_id GROUP BY app.doctor\_id, month ORDER BY month ASC, patient\_counter DESC

* 1. In a monthly basis, the clinic needs to provide a list of doctors showing how many times a doctor examined a patient during the last month, sorted in descending order by the number of exams

SELECT doc.id as doctor\_id, doc.name as doctor\_name, COUNT(app.doctor\_id) as patient\_counter, MONTH(app.date\_time) as month, MONTHNAME(app.date\_time) as month\_name

FROM appointment as app INNER JOIN doctor as doc ON doc.id = app.doctor\_id

where MONTH(app.date\_time) = 1

GROUP BY app.doctor\_id

ORDER BY patient\_counter DESC

* 1. In an annual basis, the clinic needs to provide a list of doctors showing how many times a doctor examined a patient during the last year, sorted in descending order by the number of exams

SELECT doc.id as doctor\_id, doc.name as doctor\_name, COUNT(app.doctor\_id) as appointment\_counter, YEAR(app.date\_time) as year

FROM appointment as app INNER JOIN doctor as doc ON doc.id = app.doctor\_id

where YEAR(app.date\_time) = 2017

GROUP BY app.doctor\_id

ORDER BY appointment\_counter DESC

* 1. In a monthly basis, the clinic needs to provide a list of doctors showing how many times a doctor worked in behalf of another doctor during the last month. Sort the list in descending order by the number of times

SELECT doc.id as doctor\_id, doc.name as doctor\_name, COUNT(app.doctor\_id) as patient\_counter, MONTH(app.date\_time) as month, MONTHNAME(app.date\_time) as month\_name

FROM appointment as app

INNER JOIN doctor as doc ON doc.id = app.doctor\_id

INNER JOIN patient as pat ON pat.id = app.patient\_id

where MONTH(app.date\_time) = 1 AND pat.family\_doctor\_id != doc.id

GROUP BY app.doctor\_id

ORDER BY patient\_counter DESC

* 1. In an annual basis, the clinic needs to provide a list of doctors showing how many times a doctor worked in behalf of another doctor during the last year. Sort the list in descending order by the number of times
  2. In a monthly basis, the pharmacy needs to get a list of the Medication names sorted in descending order by the number of times it is prescribed by a doctor during the last month
  3. In an annual basis, the pharmacy needs to get a list of the Medication names sorted in descending order by the number of times it is prescribed by a doctor during the last year
  4. In a monthly basis the pharmacy needs to get a list of all medications that are not prescribed during the last month
  5. In an annual basis the pharmacy needs to get a list of all medications that are not prescribed during the last year
  6. Build a list showing how many prescriptions received by the pharmacy for each month starting with the current month and back to the same month last year. i.e. 13 months history

1. Building the Data Entry forms for each entity. Each form should allow insert a new record, update/delete of desired records
2. Build well formatted reports to incorporate the results of all the SQL queries listed above in No 4 above