Brandon Busquets, Benjamin Fuller, Gregory Pierot
CPSC 332 Project
Doctor's Office Database

---

1. One SCRIPT to create this database (call it DocOffice) with MySQL server.

*** DocOffice is included in a .txt file with this project folder. ***

---

2. Doctor Robert Stevens is retiring. We need to inform all his patients, and ask them to select a new doctor. For this purpose, Create a VIEW that finds the names and Phone numbers of all of Robert's patients.

```
CREATE
VIEW `StevensPatients` AS
    SELECT DISTINCT
        `pe2`.`FirstName` AS `firstname`,
        `pe2`.`LastName` AS `lastname`,
        `pe2`.`PhoneNumber` AS `PhoneNumber`
    FROM
        ((((`Person` `pe`
        JOIN `Doctor` `d` ON ((`pe`.`PersonID` = `d`.`PersonID`)))
        JOIN `PatientVisit` `pv` ON ((`d`.`DoctorID` = `pv`.`DoctorID`)))
        JOIN `Patient` `pa` ON ((`pa`.`PatientID` = `pv`.`PatientID`)))
        JOIN `Person` `pe2` ON ((`pe2`.`PersonID` = `pa`.`PersonID`)))
    WHERE
        ((`pe`.`FirstName` = 'Robert')
            AND (`pe`.`LastName` = 'Stevens'))
```

---

3. Create a view which has First Names, Last Names of all doctors who gave out prescriptions for Vicodin.

```
CREATE VIEW ViewVicodin AS SELECT
    p.FirstName,
    p.LastName
```

```
FROM
   Person AS p,
   Doctor AS d,
   PatientVisit AS pv,
   PVisitPrescription AS pvd,
   Prescription AS pr
WHERE
   p.PersonID = d.PersonID
   AND d.DoctorID = pv.DoctorID
   AND pv.VisitID = pvd.VisitID
   AND pvd.PrescriptionID = pr.PrescriptionID
   AND pr.PrescriptionName = 'Vicodin';
```

---

4. Create a view which shows the First Name and Last name of all doctors and their specialty's.

```
CREATE VIEW ViewDoctors AS
SELECT
   p.FirstName,
   p.LastName,
   s.SpecialtyName
FROM
   Person AS p,
   Doctor AS d,
   Specialty AS s,
   DoctorSpecialty AS ds
WHERE
   p.PersonID = d.PersonID
   AND d.DoctorID = ds.DoctorID
   AND ds.SpecialtyID = s.SpecialtyID;
```

---

5. Modify the view created in Q4 to show the First Name and Last name of all doctors and their specialties ALSO include doctors who DO NOT have any specialty.

```
ALTER VIEW
   ViewDoctors AS SELECT
   p.FirstName,
   p.LastName,
```

```
    s.SpecialtyName
FROM
    Person AS p
JOIN Doctor AS d ON
    p.PersonID = d.PersonID
LEFT JOIN DoctorSpecialty AS ds ON
    d.DoctorID = ds.DoctorID
JOIN Specialty AS s ON
    ds.SpecialtyID = s.SpecialtyID
```

---

6. Create trigger on the DoctorSpeciality so that every time a doctor specialty is updated or added, a new entry is made in the audit table. The audit table will have the following:

- **Doctor's FirstName**
- **Action(indicate update or added)**
- **Specialty**
- **Date of modification**

```
CREATE TRIGGER Uaudit
AFTER UPDATE
 ON DoctorSpecialty
 FOR EACH ROW
 INSERT INTO `docoffice`.`audit`
(`FirstName`,
`Act`,
`SpecialtyName`,
`dat`)
SELECT
pe.firstname as DoctorFirstName,
"UPDATE" as Action,
s.specialtyName as Specialty,
curdate() as Date_Modified
from Person as pe
join doctor d on pe.personID = d.personID
left join DoctorSpecialty ds on d.doctorID = ds.doctorID
left join specialty s on s.specialtyID = ds.specialtyID
```

```
CREATE TRIGGER laudit
AFTER INSERT
 ON DoctorSpecialty
 FOR EACH ROW
 INSERT INTO `docoffice`.`audit`
(`FirstName`,
`Act`,
`SpecialtyName`,
`dat`)
SELECT
pe.firstname as DoctorFirstName,
"Insert" as Action,
s.specialtyName as Specialty,
curdate() as Date_Modified
from Person as pe
join doctor d on pe.personID = d.personID
left join DoctorSpecialty ds on d.doctorID = ds.doctorID
left join specialty s on s.specialtyID = ds.specialtyID
```

---

7. Create a script to do the following (Write the script for this) backups.

a. **If first time backup take backup of all the tables**
b. **If not the first time remove the previous backup tables and take new**
c. **Extra credit: Create a stored procedure that gives Prescription name and the number of patients from the city of Fullerton with that prescription.**

You can write a python file that access the command prompt/line and back up a database. Here is the python script code:

```
Import os
os.system(‘cmd /k “mysqldump --column-statistics=0 -h 127.0.0.1 -P 3306 -u root -p docoffice > docofficeBackup.sql”’)
```

Depending on where your mysqldump.exe is located you will need admin privileges. At the end of the command you can specify where you would like the backup file to be created by adding the file path. You can also run the command prompt from the directory that has mysqldump.exe. “column-statistics=0” was used because sometimes

it would crash and not make a backup of the database. This ensures it makes a backup and replaces the old one every time.

Extra credit:

```
Select concat(count(*), '||', Prescription.PrescriptionName)
AS "" FROM pvisitprescription inner join patientvisit
ON patientvisit.VisitID = pvisitprescription.VisitID
inner join prescription
ON prescription.PrescriptionID = pvisitprescription.PrescriptionID
inner join patient ON
patientvisit.PatientID = patient.PatientID inner join doctor
ON patientvisit.DoctorID = doctor.DoctorID
WHERE patient.PersonID in (SELECT PersonID FROM Person WHERE
city = "Fullerton") GROUP BY prescription.PrescriptionName;
```
