Universitat Autònoma de Barcelona

Facultat de Ciències

Pràctica MongoDB

Bases de dades no relacionals

Sergi Cantón Simó - 1569251

Bernat Espinet Torrescassana - 1564342

Marc Llopart Enajas - 1569054

Gerard Vinyes Sanchez - 1563545

1/4/2022

Índex

1 Repositori GitHub **2**

2 Creació de les col·leccions **2**

3 Creació del fitxer de python per generar la base de dades **3**

4 Joc de proves **4**

4.1 *Escàners diferents que hi ha a la BD. Mostra el device* 4

4.2 *Número total de nòduls que s'han utilitzat per l'entrenament (train=1) de l'experiment 1 del mètode "Method2"* 4

4.3 *Valor màxim, mínim i mitjà de BenignPrec agrupat per classificador (classifier). Mostra ID del mètode, MaxBenignPrec, MinBenignPrec, AvgBenignPrec.* 4

4.4 *Número total d'homes i dones. Mostra sexe i número total* 5

4.5 *Pacients amb més de dos nòduls. Mostra ID del Pacient, sexe, edat, diagnòstic del Pacient* 5

4.6 *Mostrar els 4 mètodes amb més repeticions de l'experiment. Mostra el ID del Mètode i número de repeticions de l'experiment* 5

4.7 *Per cada pacient els escàners (CTs) que s'ha fet. Mostra el ID del Pacient, device i la data del CT* 6

4.8 *Mostrar els pacients que tenen tots els seus nòduls amb diagnosis = "Benign" i el seu recompte* 6

4.9 *Modificar la ResolutionTV aumentant-la un 20 % dels escàners que es van realitzar amb DataCT = 18/11/2018* 7

5 Repartiment de tasques **7**

1 Repositori GitHub

Per fer la pràctica, hem creat un repositori GitHub per a emmagatzemar i anar actualitzant els fitxers que hem anat fer servir.

L'enllaç al repositori és https://github.com/BernatEspinet/MongoDB-BDNR.

En aquest repositori podem trobar cinc arxius. El primer, *Dades.xlsx*, conté les dades necessàries per dur a terme les consultes del projecte. El segon fitxer, *Informe_MongoDB_BNR.pdf* és aquest mateix informe, on es troben les explicacions i justificacions de la pràctica. En el tercer fitxer, *JocDeProves.js* hi ha el conjunt de consultes de MongoDB que se'ns demanaven. Al quart fitxer, *Mongodb-BDNR.py*, hi ha el codi que crea la base de dades, llegeix les dades i les organitza en les col·leccions. Finalment, *options.py*, és un fitxer auxiliar on es defineixen opcions i paràmetres necessaris per a crear la base de dades.

2 Creació de les col·leccions

Per a la nostra base de dades, vam considerar 5 col·leccions: Nodule, Patient, CTScanner, Experiment i Relation.

La col·lecció Nodule conté els atributs "PatientID", "NoduleID", "DiagnosisPatient", "DiagnosisNodule", "PositionX", "PositionY", "PositionZ". És a dir, tota la informació del propi nòdul (id, coordenades d'on es troba i si és benigne o maligne) així com la id per relacionar-lo amb el pacient que el conté. A continuació, la col·lecció Patient està composta dels atributs "PatientID", "Age", "Gender", "DiagnosisPatient". Aquí tenim les dades personals del pacient, incloent-hi si té algun nòdul maligne o tots són benignes a l'atribut DiagnosisPatient. La col·lecció CTScanner té els atributs ÇTID", "Device", "dataCT", "ResolutionTC", "ResolutionTV", "ResolutionT", "Diameter (mm)". Aquests són els paràmetres de l'aparell de de detecció de tumors, però també s'hi inclou la dada del diàmetre del tumor detectat amb l'aparell. Seguidament, la col·lecció Experiment té els atributs "PatientID","NodulID","MethodID", "ExperimentRepetition", "Train", "RadiomicsDiagnosis". ExperimentRepetition compta el nombre de repeticions que s'han fet de l'experiment, Train correspon al mètode d'aprenentatge computacional que s'ha fet servir, RadiomicsDiagnosis guarda si amb l'experiment s'ha detectat que el nòdul és maligne o benigne i, PatientID, NodulID, i MethodID són les id's d'allò amb el qual és necessari relacionar l'experiment. Finalment, la col·lecció Relation conté els atributs "PatientID", "NodulID", "MethodID", "ExperimentRepetition", "Train", "RadiomicsDiagnosis". Aquesta col·lecció és auxiliar, és a dir, no representa res físic, i serveix únicament per fer de pont entre les col·leccions Patient, Nodule, Method i Experiment.

Partint de la col·lecció Nodule, inicialment vam relacionar-la amb Patient afegint l'atribut PatientID dins de Nodule. A continuació, vam també referenciar CTScanner dins de Nodule col·locant l'atribut CTID dins de la col·lecció Nodule. A més, però, vam afegir també l'atribut Diameter, de la relació entre Nodule i CTScanner, i els atributs Device i DataCT de la classe CTScanner, dins de Nodule. Vam prendre aquesta decisió perquè vam observar que aquests tres atributs s'han de fer servir sovint a les consultes que posteriorment se'ns demanen. La classe Method i Experiment les vam relacionar encastant tots els atributs de Method dins d'Experiment perquè Method no és una col·lecció massa gran i, basant-nos en les consultes, ens era el més òptim organitzar-ho així. Seguidament, Nodule i Experiment els vam relacionar referenciant Nodule dins d'Experiment afegint l'atribut NoduleID dins d'Experiment. També, vam afegir els atributs de la relació entre Nodule i Experiment, Diagnosis i Train, dins d'Experiment. Finalment, Relation ens relaciona les claus primàries de Patient, Nodule, Method i, també, inclou els atributs ExperimentRepetition, Train i RadiomicsDiagnosis. Relation és una col·lecció artificial que ens facilita dur a terme les consultes.

3 Creació del fitxer de python per generar la base de dades

Dins l'arxiu de python, vam establir la connexió amb el servidor local, vam crear la base de dades amb les seves col·leccions i vam llegir i insertar-hi les dades.

La base de dades que vam crear s'anomena *MongoDB-BDNR*. Després de generar-la, ens va fer falta crear concretament les col·leccions que hem especificat a l'apartat 1. Per crear cada col·lecció, es fa un tractament de l'error per si la col·lecció ja està creada. En cas afirmatiu, s'esborra la col·lecció i es torna a crear. Això ens permet evitar que s'insereixin valors duplicats dins la base de dades si hi ha qualsevol error d'execució, la base dades es queda a mitja creació i s'ha de tornar a executar el fitxer de python.

A continuació, es genera un objecte de la classe *Options* que ens aporta les opcions necessàries per llegir el fitxer de dades, esborrar el contingut de les nostres bases de dades, etc.

Posteriorment, dins del fitxer de python es llegeixen les dades des de Excel fent servir una funció de la llibreria *Pandas* i s'afegeixen aquestes dades dins de les col·leccions que correspongui. En els casos on vam decidir referenciar, únicament ens va caler posar l'identificador d'una col·lecció dins de l'altre. Pels casos on vam decidir encastar, inicialment vam haver de crear el diccionari amb els atributs a encastar per, posteriorment, afegir aquest diccionari dins del diccionari corresponent a la col·lecció on volíem encastar.

Per acabar, vam modificar el fitxer *Options* per afegir la opció que esborri el contingut de les bases de dades al crear un objecte de la classe *Options*.

4 Joc de proves

L'última part de la pràctica va consistir en dur a terme una sèrie de consultes de la base de dades.

4.1 *Escàners diferents que hi ha a la BD. Mostra el device*

Resolem la consulta fent un *distinct* dels *Devices* de la base de dades a la col·lecció CTScanner. La consulta és

```
db.CTScanner.distinct("Device"),
```

i dona com a resultat

	value
1	Fujifilm Holdings
2	GE Healthcare
3	Siemens Sensation
4	Toshiba Aquilon One

Figura 1: Resultat de la primera consulta.

4.2 *Número total de nòduls que s'han utilitzat per l'entrenament (train=1) de l'experiment 1 del mètode "Method2"*

A la col·lecció Relation, apliquem una pipeline (*aggregate*), dins de la qual seleccionem els atributs que ens interessen i fem un *count* del nombre de nòduls. La consulta és

```
db.Relation.aggregate({$match: {Train: 1, ExperimentRepetition:1, "MethodID":"Method2"}},
{$count: "NodulID"}) ,
```

i dona com a resultat

	NodulID
1	70

Figura 2: Resultat de la segona consulta.

4.3 *Valor màxim, mínim i mitjà de BenignPrec agrupat per classificador (classifier). Mostra ID del mètode, MaxBenignPrec, MinBenignPrec, AvgBenignPrec.*

A la col·lecció Experiment, apliquem una pipeline (*aggregate*), dins de la qual separem l'array on guardem les dades del mètode. La consulta és

```
db.Experiment.aggregate([ {$unwind: "$Method"}, {$group: {_id: "$Method.Classifier",
MaxBenignPrec: {$max: "$BenignPrec"}, MinBeningPrec: {$min: "$BenignPrec"}, AvgBeningPrec:
{$avg: "$BenignPrec"}}} ]) ,
```

i dona com a resultat

	_id	MaxBenignPrec	MinBeningPrec	AvgBeningPrec
1	NN	44,4444	30,1887	35,5997
2	SVM	66,6667	27,5	38,2581
3	Logit	39,4737	0	31,521

Figura 3: Resultat de la tercera consulta.

4.4 *Número total d'homes i dones. Mostra sexe i número total*

A la col·lecció Patient, apliquem una pipeline (*aggregate*), dins de la qual agrupem per gènere i fem el recompte. La consulta és

```
db.Patient.aggregate({$group: {_id:"$Gender", count:{$sum:1}}}),
```

i dona com a resultat

	_id	count
1	Man	63
2	Woman	38

Figura 4: Resultat de la quarta consulta.

4.5 *Pacients amb més de dos nòduls. Mostra ID del Pacient, sexe, edat, diagnòstic del Pacient*

A la col·lecció Nodule, apliquem una pipeline (*aggregate*), dins de la qual fem un left join agrupant per la id del pacient i projectem allò que se'ns demana. A continuació, separem pel camp *info_pacient*, que ens hem creat nosaltres, i projectem per obtenir els atributs que ens interessen. La consulta és

```
db.Nodule.aggregate({$group:{_id:"$PatientID", count:{$sum:1}}}, {$match: {count:{$gt:
2}}}, {$lookup:{ from:"Patient", localField: "_id", foreignField: "_id", pipeline:[{$project:
{Gender:1, Age:1, DiagnosisPatient:1,_id:0}}], as: ïnfo_patient"}}, {$unwind: "$info_patient"},{$p
ïnfo_patient.Gender":1, ïnfo_patient.Age":1, ïnfo_patient.DiagnosisPatient":1}}),
```

i dona com a resultat

	_id	info_patient		
		Age	Gender	DiagnosisPatient
1	LIDC-IDRI-0194	60	Man	Malign
2	LIDC-IDRI-1011	69	Woman	Malign
3	LIDC-IDRI-0174	75	Man	Malign

Figura 5: Resultat de la cinquena consulta.

4.6 *Mostrar els 4 mètodes amb més repeticions de l'experiment. Mostra el ID del Mètode i número de repeticions de l'experiment*

A la col·lecció Experiment, apliquem una pipeline (*aggregate*), dins de la qual separem el camp *Method*, agrupem per id i comptem quants n'hi ha. Finalment, mostrem només aquells 4 que es repeteixen més cops. La consulta és

```
db.Experiment.aggregate({$unwind: "$Method"},{$group:{_id: "$Method.MethodID", count:{$sum:1}}},{
{$limit: 4}) ,
```

i dona com a resultat

	_id	count
1	Method2	5
2	Method16	5
3	Method13	5
4	Method9	5

Figura 6: Resultat de la sisena consulta.

4.7 *Per cada pacient els escàners (CTs) que s'ha fet. Mostra el ID del Pacient, device i la data del CT*

A la col·lecció Nodule, apliquem una pipeline (*aggregate*), dins de la qual fem un left join de Nodule amb CTScanner, per tenir les dades del scanner als documents dels nòduls. A continuació, separem els arrays de la infomació que acabem d'afegir amb el join i agrupem per aquella informació que se'ns demana. La consulta és

```
db.Nodule.aggregate({$lookup:{ from:ÇTScanner", localField: ÇTScanner.CTID", foreignField:
ÇTID", pipeline:[{$project:{Device:1, dataCT:1, _id:0}}], as: ïnfo_ct"}}, {$unwind:
"$info_ct"}, {$group: {_id:"$PatientID", device:{$addToSet:"$info_ct.Device"}, dataCT:
{$addToSet: "$info_ct.dataCT"}}}, {$unwind:"$device"}, {$unwind:"$dataCT"}),
```

i dona com a resultat

	_id	device	dataCT
1	LIDC-IDRI-0286	Fujifilm Holdings	8/5/2018 2:00:00
2	LIDC-IDRI-0580	Fujifilm Holdings	23/12/2018 1:00:00
3	LIDC-IDRI-0169	Fujifilm Holdings	30/11/2018 1:00:00
4	LIDC-IDRI-0255	Fujifilm Holdings	13/3/2018 1:00:00
5	LIDC-IDRI-0137	Fujifilm Holdings	7/8/2018 2:00:00
6	LIDC-IDRI-0180	Toshiba Aquilon One	5/6/2018 2:00:00
7	LIDC-IDRI-0242	Toshiba Aquilon One	3/8/2018 2:00:00
8	LIDC-IDRI-0071	Siemens Sensation	15/7/2018 2:00:00
9	LIDC-IDRI-0256	Siemens Sensation	26/11/2018 1:00:00

Figura 7: Resultat de la setena consulta.

4.8 *Mostrar els pacients que tenen tots els seus nòduls amb diagnosis = "Benign" i el seu recompte*

A la col·lecció Nodule, apliquem una pipeline (*aggregate*), dins de la qual filtrem per la condició que el nòdul sigui benigne i comptem quants n'hi ha. La consulta és

```
db.Nodule.aggregate({$match: {"DiagnosisPatient":"Benign", "DiagnosisNodul": "Benign"}},
{$group: {_id:"$PatientID", recompte:{$sum:1}}}) ,
```

i dona com a resultat

	_id	recompte
1	LIDC-IDRI-0233	1
2	LIDC-IDRI-0185	1
3	LIDC-IDRI-0234	2
4	LIDC-IDRI-0212	1
5	LIDC-IDRI-0183	1
6	LIDC-IDRI-0253	1
7	LIDC-IDRI-0187	1
8	LIDC-IDRI-0275	1
9	LIDC-IDRI-0167	1

Figura 8: Resultat de la vuitena consulta.

4.9 *Modificar la ResolutionTV aumentant-la un 20 % dels escàners que es van realitzar amb DataCT = 18/11/2018*

A la col·lecció CTScanner, fem un *update* per actualitzar la resolució multiplicant-la per 1.2 únicament en els escàners amb la data que se'ns demana. La consulta és

```
db.CTScanner.updateMany({dataCT: ISODate("2018-11-18")},{$mul:  {ResolutionTV: 1.2}})
db.CTScanner.find().
```

Amb aquesta consulta es modifica la base de dades tal i com s'especifica.

5 Repartiment de tasques

A l'iniciar la pràctica, vam repartir les tasques equitativament. El script de python el vam fer principalment el Bernat i el Sergi i les consultes de MongoDB les vam fer majoritàriament el Marc i el Gerard. Respecte l'informe, cadascú va escriure allò referent a la part de la pràctica que havia dut a terme.

No obstant, tot i haver repartit les parts de la pràctica, tots hem estat al cas també de les parts que inicialment no ens corresponien i hem ajudat en allò que ha calgut, independentment de si se'ns havia assignat o no.