BASE DE DADES II

PRÀCTICA 1

2021-2022

Damià Escalas Roig – 43221397G

Lluís Albert González Peña – 45372522P

INDEX

INTRODUCCIÓ

En aquest document es presenta el model conceptual, la seva traducció a relacional i la posterior creació de la base de dades en SQL resultant d'un problema que es presenta en l'Escola Politècnica que vol gestionar els exàmens de l'estudi d'Ingenyeria Informàtica. A més, també es realitza un petit estudi sobre les bases de dades al núvol, alguns exemples d'aquestes i la possible aplicació al cas d'aquesta pràctica.

El que es vol aconseguir és una base de dades que emmagatzemi els examens que s'han fet i que s'han de fer, les preguntes que surten als examens o que poden ser emprades, juntament amb els professors que s'encarreguen d'aquestes entitats i, els alumnes matriculats i les notes que treuen cada pregunta de cada examen, juntament amb les assignatures de les que s'han matriculat, tenint en compte que poden repetir assignatures.

En el model conceptual s'identifiquen les classes, atributs i relacions que permeten la creació d'una base de dades que enmmagatzemi tota la informació que necessita l'Escola Politècnica.

Seguidament, cada entitat identificada en el model conceptual esdevé en una taula del relacional. S'identifiquen les claus identificatives de cada una d'elles i s'estudia la possibilitat de fusionar algunes de les taules/relacions resultants, donant pas a un conjunt de taules interrelacionades amb un mecanisme de claus foranes que permet referenciar les taules que siguin essàries i òptimes.

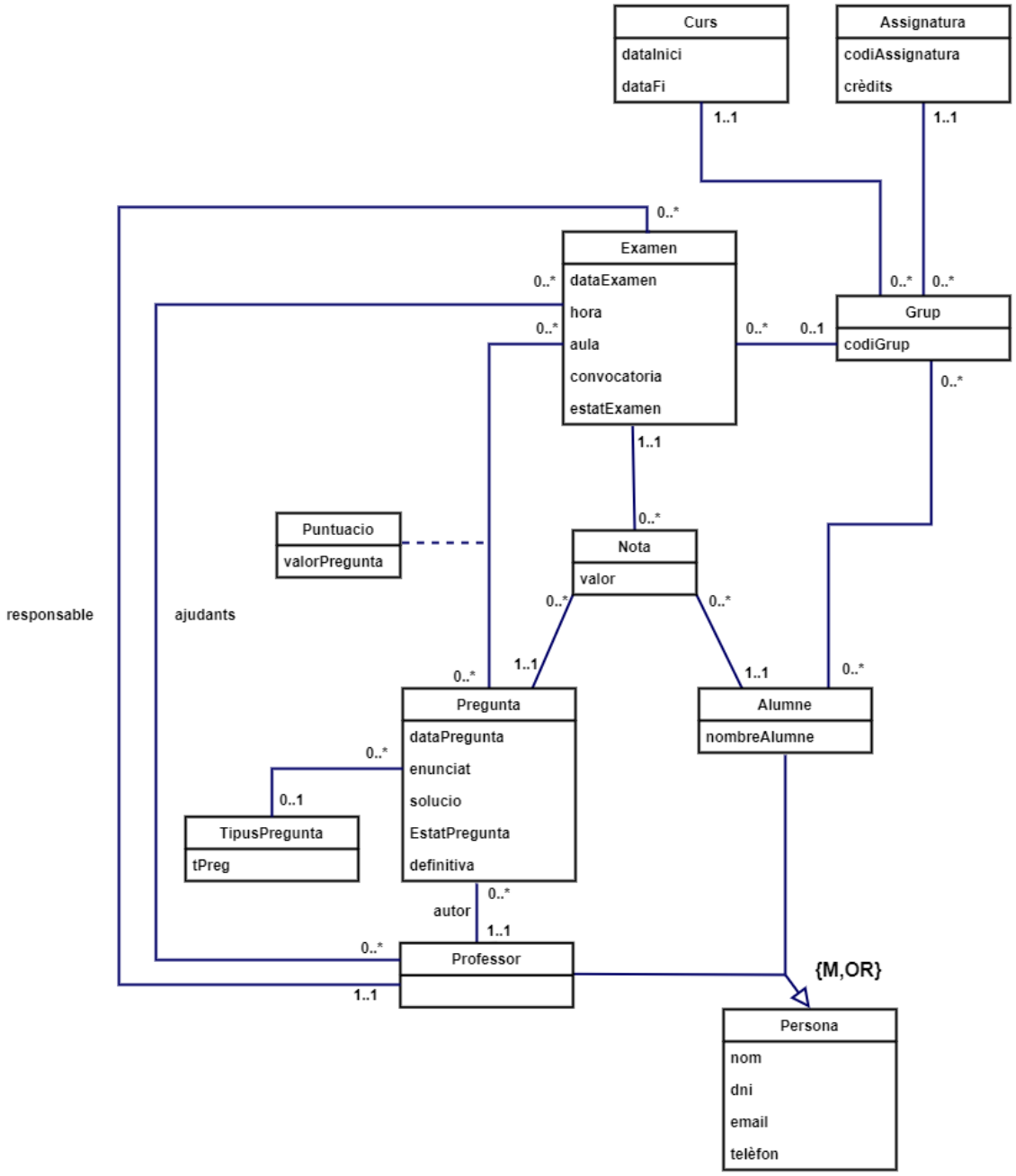
A més, tot aquest conjunt de taules serà interpetat en llenguatge sql, llenguatge adoptat per la majoria dels SGBDR (Sistemes de Gestió de Bases de Dades Relacionals), permetent la creació de la base de dades.

Finalment, ens trobam amb l'estudi de les bases de dades al núvol, on hem definit les característiques clau, els tipus que hi ha, els model de dades amb que treballa, els beneficis i desavantatges, la migració de bases de dades locals i alguns exemples d'aquestes bases de dades. També es valora la seva utilització en el cas d'aquesta pràctica.

CAPÍTOL 1 – MODEL CONCEPTUAL

Després d'un exhaustiu anàlisi del problema proposat es procedeix a construir un model conceptual que permeti enregistrar totes les dades personals.

El model conceptual resultant és el que es mostra a continuació:



En aquest diagrama s'hi distingueixen una sèrie de classes, els seus atributs i les relacions entre elles. Ara procedirem a explicar-les indicant les suposicions adoptades i els comentaris pertinents.

DOCUMENTACIÓ DEL MODEL CONCEPTUAL

Començarem per la classe persona (nom, dni, email i telèfon) de la qual n'hereten la classe Professor i la classe Alumne, ja que comparteixen els atributs de la classe mare. La classe Alumne tendrà un atribut més, nombreAlumne.

L'herència serà del tipus {M, OR}. La restricció de participació Mandatory es deu a que qualsevol persona que es registri a la base de dades haurà de ser, o bé un alumne, o bé un professor. I la restricció de conjunció OR es deu a la possibilitat què un professor hagi estat prèviament un alumne.

- *PERSONA (nom, dni, email, telefon)*
- *PROFESSOR ( )*
- *ALUMNE (nombreAlumne)*

Un cop explicat l'herència de les classes Professor i Alumne passam a identificar les classe pregunta i tipusPregunta.

La classe pregunta ens serveix per enregistrar totes les preguntes que poden ser incloses en un examen. Té els atributs data, enunciat, solució, estatPregunta (indicar si la pregunta està oberta i es pot emprar o tancada i no es pot emprar) i definitiva (indicar si l'autor ha donat la pregunta per definitiva o encara ha de realitzar algun canvi). La classe TipusPregunta ens indica el tipus de pregunta que és: desenvolupament, test, curta, …

- *PREGUNTA (dataPregunta, enunciat, solucio, EstatPregunta, definitiva)*
- *TIPUSPREGUNTA (tPreg)*

A continuació hi trobam la classe Examen amb els atributs dataExamen, hora, aula, convocatoria (extraordinaria o ordinaria) i estatExamen (planificant, bloquetjat, en curs, finalitzat, corregint, corregit, tancat…). Hem decidit que convocatòria sigui un atribut perquè lo únic que ens interessa de la convocatoria és dir de quin tipus és, pel que la clase només tendría un atribut que seria tipusConvocatoria, i ja que l'examen pertany sempre a la convocatoria ordinaria o extraordinaria, hem considerat que fer una clase no seria necessari.

- *EXAMEN (dataExamen, hora, aula, convocatoria, estatExamen)*

El valor d'una pregunta de cada pregunta pot variar a cada un dels exàmens on s'hagi posat, per això tenim la classe associativa Puntuacio (valorPregunta).

- *PUNTUACIO (valorPregunta)*

Per indicar la nota que ha tret un alumne concret en una pregunta específica en un exàmen necessitam la classe Nota (valor).

- *NOTA (valor)*

Necessitam saber els alumnes que s'han matricular ara i sempre d'una assignatura (es pot repetir), per això necessitam les classe Curs, Assignatura i Grup.

Sobre un curs només hem de saber la data d'inici i la data de fi i d'una assignatura el seu codi i el nombre de crèdits que suposa.

La classe grup ens serveix per poder identificar els alumnes matriculats i els exàmens fets en qualsevol assignatura en un curs concret. Cada grup es trobarà identificat amb un codi únic (codiGrup).

- *CURS (dataInici, dataFi)*
- *ASSIGNATURA (codiAssignatura, credits)*
- *GRUP (codiGrup)*

RELACIONS ENTRE CLASSES

Un cop hem explicat el perquè de cada una de les classes procedirem a explicar el perquè de les relacions entre elles.

- R_PROFESSOR_PREGUNTA

Aquesta relació ens permet registrar qui és l'autor d'una pregunta i quines preguntes ha redactat un professor. Podria haver-hi professors que no han redactat cap pregunta (0..* a 1..1).

- R_PROFESSOR_EXAMEN_RESPONSABLE

Al igual que en la relació anterior, permet indicar qui és el professor responsable d'un examen i de quins exàmens ha estat responsable un professor. Podria haver-hi professors que no han estat responsable de cap exàmen (0..* a 1..1).

- R_PROFESSOR_EXAMEN_AJUDANTS

Permet saber quins professors intervenen en un examen (sense ser el responsable) i a quins exàmens ha intervingut un professor. Podria haver-hi professors que no han intervingut en cap examen i exàmens que no han necessitat la intervenció de més professors a part del responsable (0..* a 0..*).

- R_PREGUNTA_TIPUSPREGUNTA

Cada pregunta només pot ser d'un tipus concret i hi pot haver moltes preguntes d'un mateix tipus. Quan es registri un tipus de pregunta per primera vegada no haurà estat assignada a cap pregunta. A més, primer es crearà la pregunta i després se li assignarà el tipus que és (0..* a 0..1).

- R_PREGUNTA_EXAMEN

Relació associativa entre pregunta i examen, aquesta relació es dona ja que és necessari enregistrar un valor diferent per a una pregunta segons l'examen en que surti. Una pregunta pot no aparèixer a cap exàmen i un exàmen quan es crea encara no tenir cap pregunta assignada (0..* a 0..*).

- R_NOTA_PREGUNTA

Una nota es correspon amb una pregunta en concret i una pregunta tendrà moltes notes distints, segons l'examen i els alumnes. Una pregunta pot no tenir cap nota si no ha entrat a cap examen o si l'examen en què ha sortit encara no s'ha corregit (0..* a 1..1).

- R_NOTA_EXAMEN

Una nota pertany a un sol examen i un examen tendrà moltes notes distints, segons les preguntes i els alumnes. Un examen pot no tenir cap si encara no s'ha corregit (0..* a 1..1).

- R_NOTA_ALUMNE

Una nota es correspon amb una pregunta en concret i una pregunta tendrà moltes notes distints, segons l'examen i els alumnes. Un alumne pot no tenir cap nota si encara no ha realitzat cap examen o si els que ha realitzat encara no s'han corregit (0..* a 1..1).

- R_GRUP_ALUMNE

Un grup tindrà enregistrats molts d'alumnes i un alumne pot estar enregistrat a molts de grups. Quan es registra el grup no tendrà cap alumne apuntat i el mateix amb un alumne recent registrat (0..* a 0..*).

- R_GRUP_EXAMEN

Un examen només podrà pertànyer a un grup en concret, ja que tot i que es pogués emprar un mateix exàmen en anys distints, la fecha, hora i aula no seria la mateixa. I un grup realitzarà varis exàmens durant el curs. A més, s'ha considerat que primer es crearia l'examen i després se li assignaria el grup al que pertoca (0..* a 0..1).

- R_GRUP_CURS

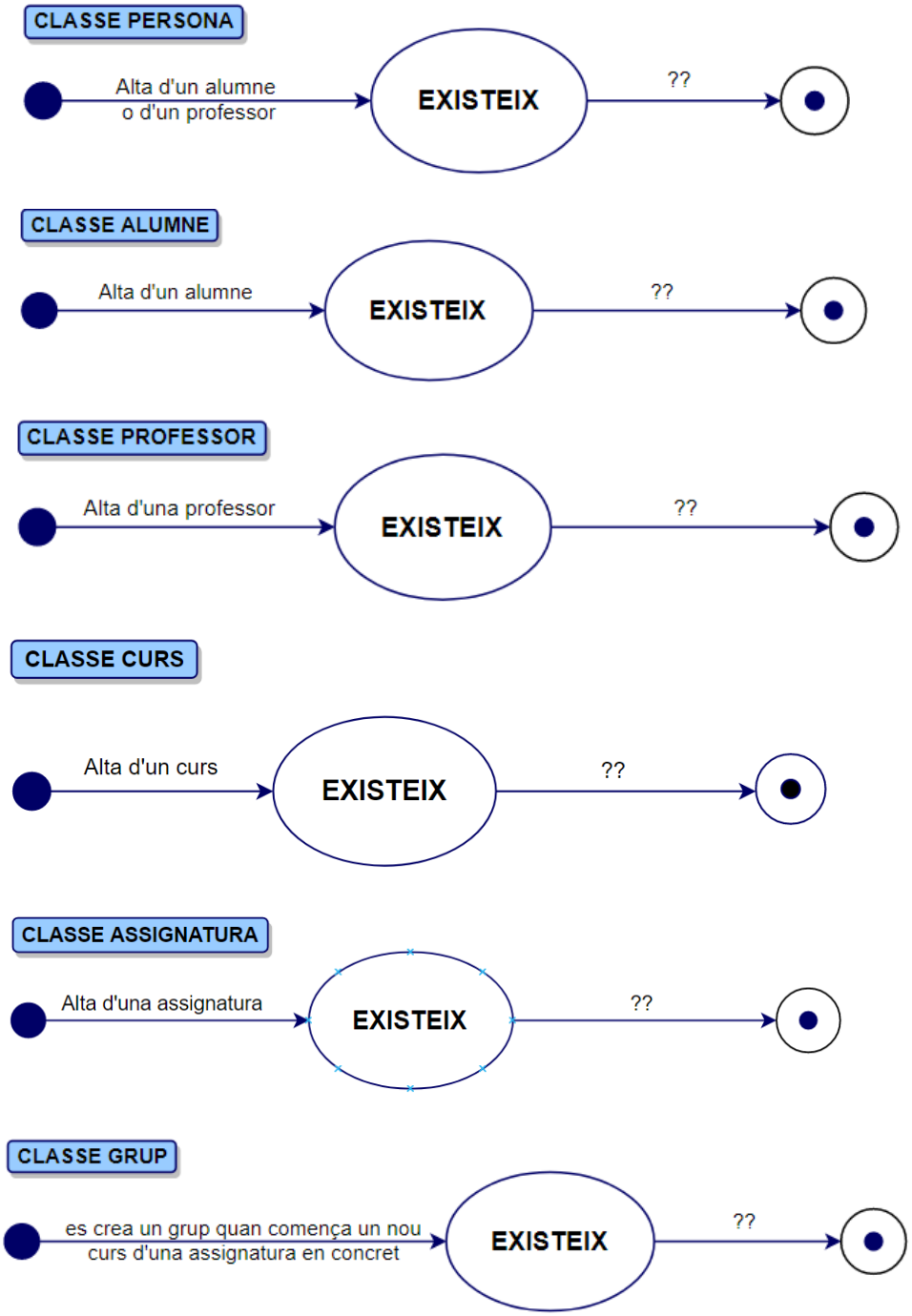
Un grup pertanyirà a un únic curs i un curs tindrà molts de grups, un per assignatura existent. Els grups sempre estaran identificats amb un curs (únic) però quan es registra un nou curs pot no tenir grups registrats (0..* a 1..1).

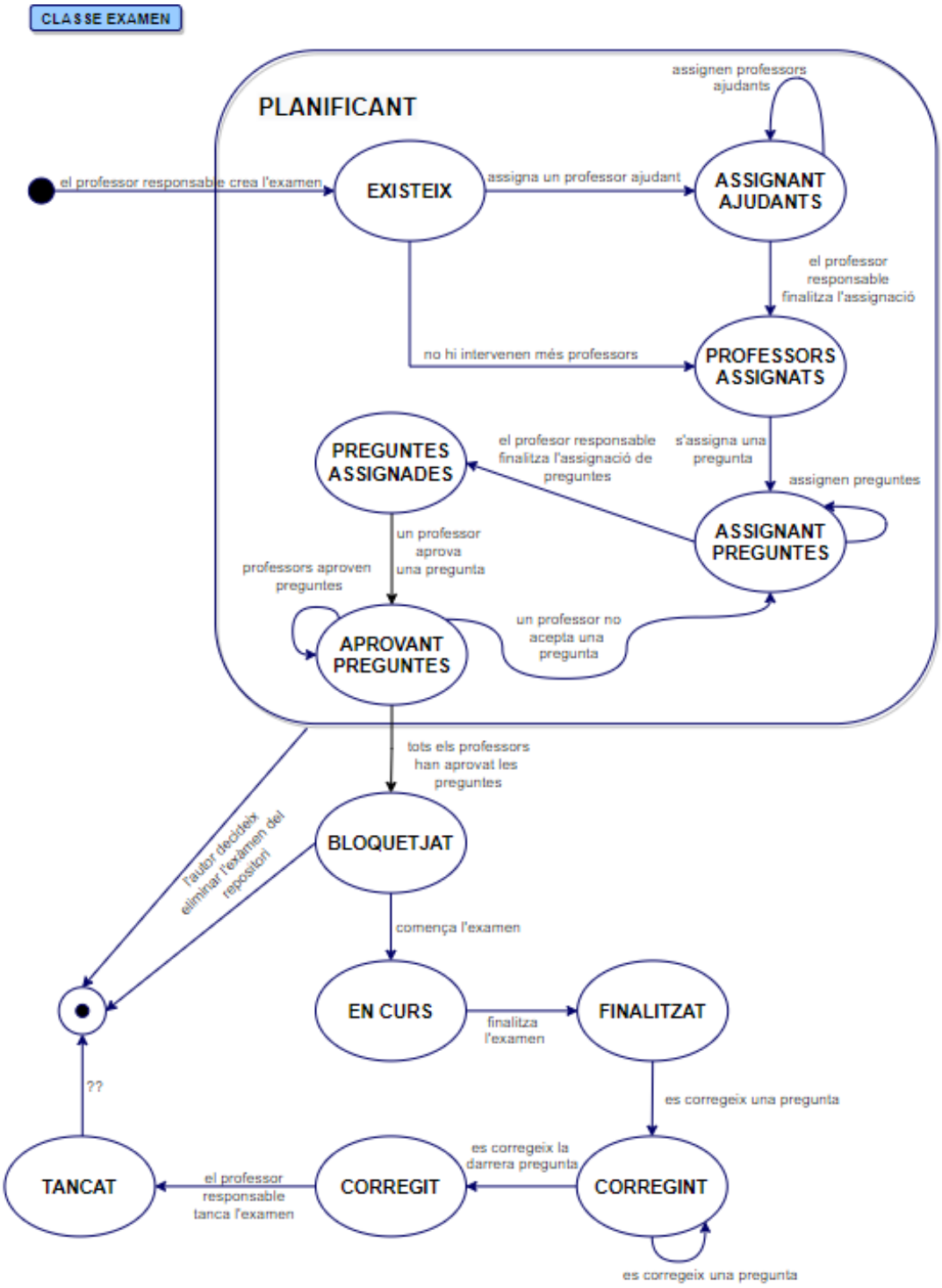
- R_GRUP_ASSIGNATURA

De mateixa manera, un grup es correspondrà amb una única assignatura i una assignatura tindrà molts de grups, un per cada curs que es registri. Al igual que en la relació anterior, els grups sempre estaran identificats amb una assignatura (única) però quan es registra una nova assignatura pot no tenir grups registrats (0..* a 1..1).

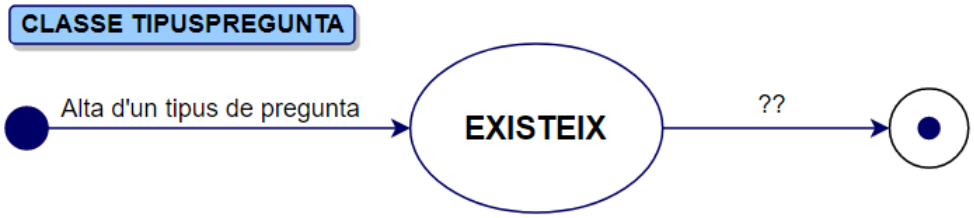
CAPÍTOL 2 - DIAGRAMES D'ESTAT DE LES CLASSES

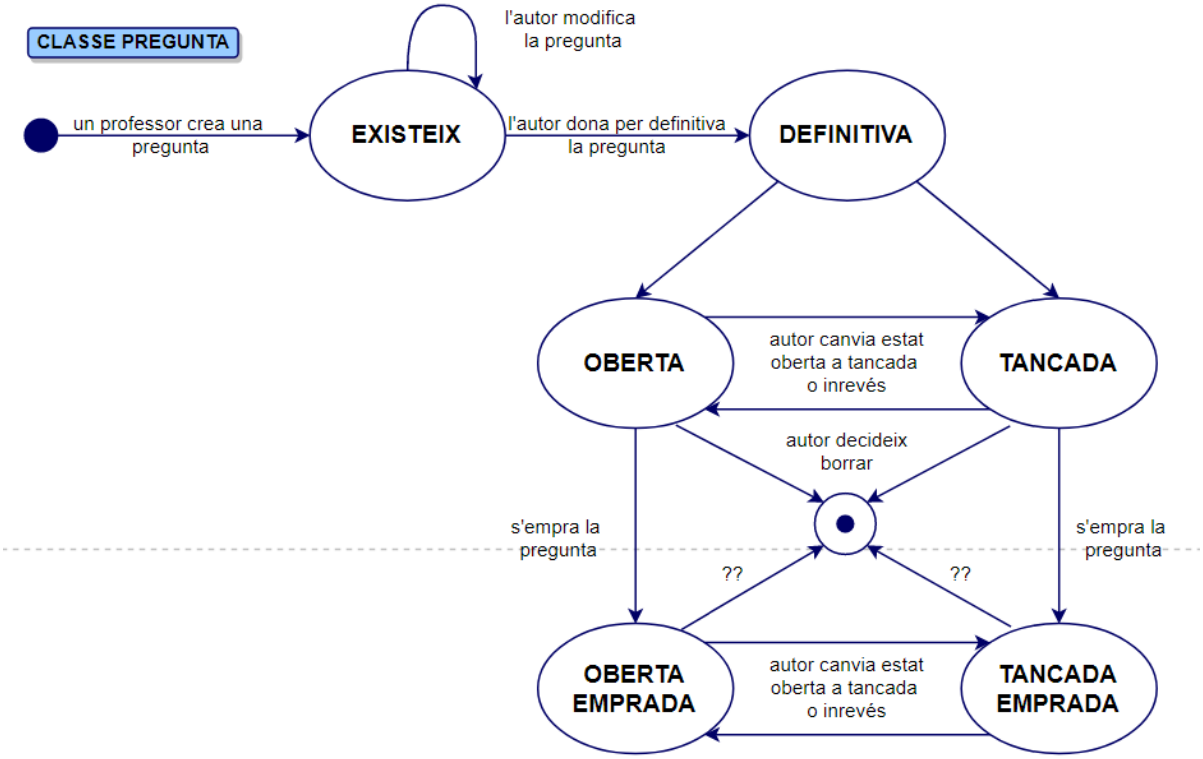
Una vegada hem identificat totes les classes necessàries i hem dissenyat el model conceptual de la base de dades procedirem a definir els diagrames d'estat de cada una de les classes.



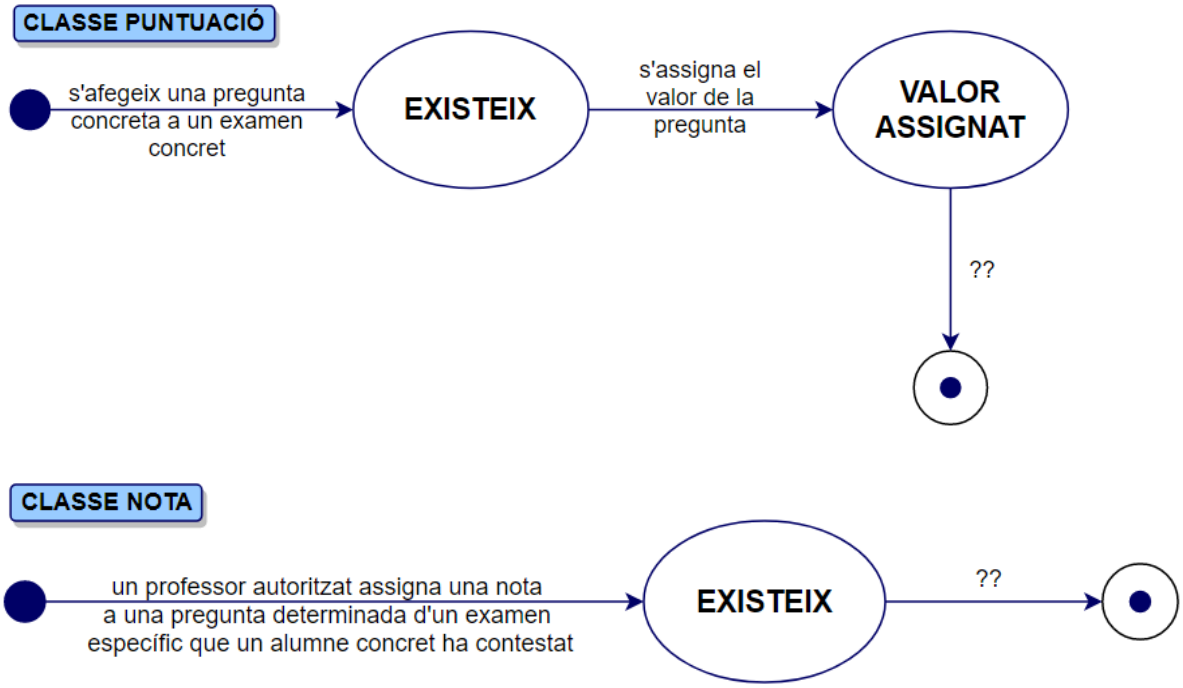


El diagrama d'estats de la classe EXAMEN és el més llarg degut a que un examen es trobarà en molts d'estats. Primer de tot, en un superestat, PLANIFICANT, que englobarà altres estats més concrets com ASSIGNANT PROFESSORS, PROFESSORS ASSIGNATS, ASSIGNANT PREGUNTES, PREGUNTES ASSIGNADES i APROVANT PREGUNTES. Si mentre s'estan aprovant preguntes s'en rebutja una, es tornarà a l'estat ASSIGNANT PREGUNTES. Un cop planificat i s'hagin aprovat totes les preguntes l'examen es bloquejarà. Posteriorment, l'examen es trobarà EN CURS, FINALITZAT, CORREGINT les preguntes i CORREGIT. Un cop estiguin corregits, el professor responsable podrà tancar l'examen.





L'altre diagrama d'estats una mica distint i dens és el de pregunta. Primer de tot es crearà la pregunta sense informació, la qual s'anirà introduint, fins que l'autor doni la pregunta per definitiva. La pregunta es podrà trobar OBERTA (pugui ser emprada per qualsevol professor) o TANCADA (si no es pot emprar), estats que l'autor podrà canviar quan vulgui. Quan s'empra una pregunta pasarà a estar OBERTA EMPRADA o TANCADA EMPRADA, segons el que decideixi l'autor. Aquesta diferenciació és necessària degut a que l'autor podrà borrar una pregunta sempre i quant no hagi estat emprada en cap examen.



Comentaris adients:

En tots els casos, s'ha incorporat el símbol **??** en la transició cap a l'estat final. Això és degut a que no sabem de quina manera ni quan s'eliminarà la informació de la base de dades. Exceptuant el casos específics, a la classe Pregunta, que ens indica que l'autor la podrà eliminar sempre i quant la pregunta no s'hagi emprat, i a la classe Examen en què l'autor el podrà borrar sempre i quant encara no s'hagi realitzat. Per aquest motiu, s'ha incorporat en el diagrama d'estats de la classe Examen una transició del superestat PLANIFICANT a l'estat final indicant que l'autor podrà borrar l'examen del repositori. De mateixa manera, s'ha indicat en el diagrama d'estats de Pregunta que, mentre la pregunta no s'hagi emprat, l'autor podrà borrar-la.

CAPÍTOL 3- TRADUCCIÓ A MODEL RELACIONAL

Per a traduir el model conceptual a relacional cada entitat del model conceptual esdevé una taula del relacional.

- PERSONA (nom, dni, email, telèfon)
- PROFESSOR ()
- ALUMNE (nombreAlumne)
- EXAMEN (data, hora, aula, convocatoria, estatExamen)
- PUNTUACIÓ (valorPreg)
- PREGUNTA (dataPregunta, enunciar, solucio, estat, definitiva)
- TIPUSPREGUNTA (tpreg)
- NOTA (notaPreg)
- CURS (dataInici, dataFi)
- ASSIGNATURA (codiAssignatura, crèdits)
- GRUP (codiGrup)

Un cop s'han plasmat totes les taules del model relacional s'escull una clau principal per a cada relació. La clau primària que s'esculli per a una relació no ha de tenir valors repetits, els seus atributs no poden tenir valors nuls, i no pot esser un atribut que pugui canviar al cap d'un cert temps.

- PERSONA (nom, **dni**, email, telèfon)

En la classe persona tenim un atribut clarament identificatiu, que no tendrá valors repetits ni nuls i mai canviarà. Aquest atribut és **dni**. De mateixa manera, les clases Professor i Alumne que hereten de Persona tendran la mateixa clau primària, la qual renombrarem en **dniProfessor** i **dniAlumne**, respectivament.

- PROFESSOR (**dniProfessor**)
- ALUMNE (nombreAlumne, **dniAlumne**)

- EXAMEN (data, hora, aula, convocatòria, estatExamen, **idExamen**)

Cap dels atributs de la classe Examen ens serveix. Tots ells podrien tenir valors repetits. Per tant, establim un **idExamen** com a clau primària.

- PUNTUACIÓ (valorPreg, **idPuntuació**)

El mateix succeeix amb la classe Puntuació, l'atribut valorPreg tendrá molts valors repetits. Per tant, afegim **idPuntuació** com la clau primària de la classe.

- PREGUNTA (dataPregunta, enunciar, solucio, estat, definitiva, **idPregunta**)

Com en les classes anteriors, la classe Pregunta no té un atribut que identifiqui de manera unequívoca cada una de les preguntes. Per això, introduim un **idPregunta** com a clau primària.

- TIPUSPREGUNTA (**tPreg**)

En la classe tipusPregunta sí podem agafar l'atribut **tPreg** com a clau primària, ja que identifica unequívocament cada un dels registres. No trobarem mai dos tPreg iguals, ni amb valors nuls.

- NOTA (notaPreg, **idNota**)

En aquest cas, tornam a necessitar la introducció d'un nou atribut, **idNota**, per a que sigui la clau primària, ja que notaPreg no serveix perquè hi haurà moltes preguntes amb una mateixa nota.

- CURS (dataInici, dataFi, **idCurs**)

Les dates d'inici i fi del curs no poden ser claus primàries ja que diferents cursos podrien començar i/o acabar el mateix dia. Per tant, emprarem un nou atribut, **idCurs**, com a clau primària.

- ASSIGNATURA (**codiAssignatura**, crèdits)

Aquí sí podem emprar **codiAssignatura** com a clau primària, ja que és un codi únic identificatiu per a cada asignatura. No hi haurà dues assignatures amb el mateix codi.

- GRUP (**codiGrup**)

En la classe grup podem emprar l'atribut que hem definit, **codiGrup,** per a què sigui la clau primària, ja que és únic per a cada grup.

Quan ja s'han elegit les claus principals per a totes les classes és el moment de les relacions. Cada relació esdevé una taula amb les claus de les entitats que relaciona i es selecciona una clau principal per a cada una d'elles.

- R_CURS_GRUP (idCurs, **codiGrup**)

En aquesta relació hem escollit codiGrup com a clau principal ja que idCurs no compleix amb la unicitat de la clau primària, té valors repetits per a una relació. Hi pot haver molts de grups en un curs, però un grup només pot pertànyer a un curs.

- R_ASSIGNATURA_GRUP (codiAssignatura, **codiGrup**)

Pel mateix motiu que en la relació anterior, codiGrup será la clau principal d'aquesta relació. Hi pot haver molts de grups d'una assignatura, però un grup només pot pertànyer a una asignatura.

- R_EXAMEN_GRUP (**idExamen**, codiGrup)

Repetim el mateix motiu. Un grup pot fer varis exàmens però un exàmen només pot ser realitzat per un grup. Per això, idExamen és la clau primària.

- R_ALUMNE_GRUP (**DNIAlumne, codiGrup**)

En aquesta relació les dues claus candidates actúen com a una clau primària composta, ja que és una relació de molts a molts. Cap de les dues pot actura com a clau primària única perquè no es compleix la unicitat de la clau primària.

- R_TIPUSPREGUNTA_PREGUNTA (tPreg, **idPregunta**)

Hem escollit idPregunta com a clau principal ja que tPreg no compleix amb la unicitat de la clau primària, té valors repetits per a una relació. Hi pot haver moltes preguntes d'un mateix tipus, però una pregunta sols pot ser d'un grup.

- R_PROFESSOR_PREGUNTA (dniProfessor, **idPregunta**)

Igual que en l'anterior relació. Una pregunta sols ha estat redactada per un professor, però un professor pot haver redactat moltes preguntes. Per tant, idPregunta será la clau principal.

- R_PROFESSOR_EXAMEN_RESPONSABLE (dniProfessorResponsable, **idExamen**)

Procedint de la mateixa manera; una examen sols té un professor responsable, però un professor pot ser/haver estat responsable de molts exàmens. Per tant, idExamen será la clau principal.

- R_PROFESSOR_EXAMEN_AJUDANT (**dniProfessorAjudant, idExamen**)

En aquesta relació les dues claus candidates actúen com a una clau primària composta, ja que és una relació de molts a molts. Cap de les dues pot actuar com a clau primària única perquè no es compleix la unicitat de la clau primària.

- R_PREGUNTA_EXAMEN (**idPregunta, idExamen**)

Aquí també actúen les dues claus candidates com a una clau primària composta, ja que és una relació de molts a molts. Cap de les dues pot actuar com a clau primària única perquè no es compleix la unicitat de la clau primària.

- R_PREGUNTA_NOTA (idPregunta, **idNota**)

Una pregunta pot tenir moltes notes, una per cada alumne que contesti aquesta pregunta, però una nota només pot ser d'una pregunta. Per això, idNota serà la clau principal.

- R_ALUMNE_NOTA (dniAlumne, **idNota**)

Un alumne pot tenir moltes notes, una per cada pregunta que contesti, però una nota només pot ser d'un examen. Per tant, idNota serà la clau principal.

- R_EXAMEN_NOTA (idExamen, **idNota**)

Un examen pot tenir moltes notes, una per cada pregunta que contesti cada alumne, però una nota només pot ser d'un examen. Per això, idNota serà la clau principal.

Un cop s'ha triat la clau principal de cada relació es procedeix a realitzar el darrer pas, estudiar la possibilitat de "fusionar" algunes de les taules/relacions resultants. Les candidates són les que tenen la mateixa clau però, cal tenir en compte el seu significat i la conveniència.

El primer que farem serà estudiar la possibilitat de fusionar l'herència de Persona. L'herència, com ja hem explicat abans, és del tipus {M, OR}, fent impossible fusionar la classe mare amb les filles. Per tant, es quedaran les tres classes independents.

- PERSONA (nom, **dni**, email, telèfon)
- PROFESSOR (**dniProfessor**)
- ALUMNE (**dniAlumne**)

Un cop s'ha analitzat la possibilitat de fusionar l'herència estudiarem fusionar les clases i les relacions restants:

- EXAMEN (dataExamen, hora, aula, convocatoria, estatExamen, **idExamen**)
- PUNTUACIO (valorPreg, **idPuntuacio**)
- PREGUNTA (dataPregunta, enuncita, solucio, estat, definitiva, **idPregunta**)
- TIPUSPREGUNTA (**tPreg**)
- NOTA (notaPreg, **idNota**)
- CURS (dataInici, dataFi, **idCurs**)
- ASSIGNATURA (**codiAssignatura**, crèdits)
- GRUP (**codiGrup**)

    - R_CURS_GRUP (idCurs, **codiGrup**)
    - R_ASSIGNATURA_GRUP (codiAssignatura, **codiGrup**)
    - R_EXAMEN_GRUP (**idExamen**, codiGrup)
    - R_ALUMNE_GRUP (**dniAlumne, codiGrup**)
    - R_TIPUSPREGUNTA_PREGUNTA (tPreg, **idPregunta**)
    - R_PROFESSOR_PREGUNTA (dniProfessor, **idPregunta**)
    - R_PROFESSOR_EXAMEN_RESPONSABLE (DNIProfessorResponsable, **idExamen**)
    - R_PROFESSOR_EXAMEN_AJUDANT (**dniProfessorAjudant, idExamen**)
    - R_PREGUNTA_EXAMEN (**idPregunta, idExamen**)
    - R_PREGUNTA_NOTA (idPregunta, **idNota**)
    - R_ALUMNE_NOTA (dniAlumne, **idNota**)
    - R_EXAMEN_NOTA (idExamen, **idNota**)

Podem fusionar la classe NOTA amb les relacions R_PREGUNTA_NOTA, R_ALUMNE_NOTA, R_EXAMEN_NOTA.

- NOTA (notaPreg, **idNota**, *idPregunta*, *idExamen*, *DNIAlumne*)

Per tant, la taula Nota té tres claus foranes *idPregunta* a Pregunta, *idExamen* a Examen i *dniAlumne* a Alumne.

La següent fusió que podem fer és la classe GRUP amb les relacions R_CURS_GRUP, R_ASSIGNATURA_GRUP.

- GRUP (**codiGrup**, acceptat, *idCurs*, *codiAssignatura*)

Ens queda així que la taula Grup té dues claus foranes *idCurs* a Curs i *codiAssignatura* a Assignatura.

També podem fusionar la classe EXAMEN amb les relacions R_PROFESSOR_EXAMEN_RESPONSABLE, R_EXAMEN_GRUP.

- EXAMEN (data, hora, aula, convocatòria, estatExamen, **idExamen**, *codiGrup*, *dniProfessorResponsable*)

Com a claus foranes tenim codiGrup a Grup i dniProfessorResponsable a Professor.

Podem fusionar la classe PREGUNTA amb les relacions R_TIPUSPREGUNTA_PREGUNTA, R_PROFESSOR_PREGUNTA.

- PREGUNTA (dataPregunta, enunciar, solucio, estat, definitiva, **idPregunta**, *tPreg*, *dniProfessor*)

Tenint com a claus foranes tPreg a TipusPregunta i dniProfessor a Professor.

Finalment, la darrera fusió que podem fer és entre la classe PUNTUACIÓ i la relació R_PREGUNTA_EXAMEN.

Quan una relació té una taula associativa, aquesta s'ha de fusionar amb la taula resultant de la relació.

Aquesta nova taula té dues claus candidates, la aue s'havia definit inicialment, idPuntuació, i l'agrupació de les dues claus foranes provinents de les taules que relacionava la relació original – idPregunta, idExamen -. Així, tenim la possibilitat d'escollir entre aquestes dues. Com que la clau candidata que prové de la taula associativa és creada per l'ocasió, la podem eliminar i emprar l'altra.

- PUNTUACIÓ (valorPreg, **idPregunta, idExamen**)

Cada clau forana referenciarà a la taula d'on prové; idPregunta a la taula Pregunta i idExamen a la taula Examen.

La resta de classes i relacions no es poden fusionar.

- TIPUSPREGUNTA (**tpreg**)
- CURS (dataInici, dataFi, **idCurs**)
- ASSIGNATURA (**codiAssignatura**, crèdits)
- R_ALUMNE_GRUP (**dniAlumne, codiGrup**)
- R_PROFESSOR_EXAMEN_AJUDANT (**dniProfessorAjudant, idExamen**)

ESTRUCTURA FINAL MODEL RELACIONAL

- PERSONA (nom, **dni**, email, telèfon)
- PROFESSOR (**dniProfessor**)
  PERSONA
- ALUMNE (**dniAlumne**)
  PERSONA
- EXAMEN (data, hora, aula, convocatòria, estatExamen, **idExamen**, *codiGrup*, *dniProfessorResponsable*)
  GRUP *PROFESSOR*
- PUNTUACIÓ (valorPreg, **idPregunta, idExamen**)
  PREGUNTA *EXAMEN*
- PREGUNTA (dataPregunta, enunciat, solucio, estat, definitiva, **idPregunta**, *tPreg*, *dniProfessor*)
  TIPUSPREGUNTA *PROFESSOR*
- TIPUSPREGUNTA (**tpreg**)
- NOTA (notaPreg, **idNota**, *idPregunta*, *idExamen*, *dniAlumne*)
  PREGUNTA *EXAMEN* *ALUMNE*
- CURS (dataInici, dataFi, **idCurs**)
- ASSIGNATURA (**codiAssignatura**, crèdits)
- GRUP (**codiGrup**, acceptat, *idCurs*, *codiAssignatura*)
  CURS *ASSIGNATURA*
- R_ALUMNE_GRUP (**dniAlumne, codiGrup**)
  EXAMEN *GRUP*
- R_PROFESSOR_EXAMEN_AJUDANT (**dniProfessorAjudant, idExamen**)
  PROFESSOR *EXAMEN*

NORMALITZACIÓ

Com podem observar, tota relació es troba en 3FN i en FNBC, ja que no existeixen dependències transitives i cada determinant és una clau candidata.

CAPÍTOL 4 – CODI MySQL DE CREACIÓ DE LA BASE DE DADES

```
CREATE DATABASE Practica1;

-- Es crearan les taules anant en compte a referenciar taules que encara
-- no hagin estat creades, per això començam amb la taula Persona que no
-- té cap clau forana.


CREATE TABLE Persona(
      dni CHAR(9),
      nom VARCHAR(40) NOT NULL,
      telefon INT(9) NOT NULL,
      adreça VARCHAR(255) NOT NULL,
      PRIMARY KEY (dni)
);

CREATE TABLE Professor(
      dniProfessor CHAR(9),
      PRIMARY KEY (dniProfessor),
      FOREIGN KEY (dniProfessor) REFERENCES Persona(dni)
);

CREATE TABLE Alumne(
      dniAlumne CHAR(9),
      nombreAlumne INT(6) NOT NULL,
      PRIMARY KEY (dniAlumne),
      FOREIGN KEY (dniAlumne) REFERENCES Persona(dni)
);

CREATE TABLE Curs(
      idCurs INT(3) AUTO_INCREMENT,
      dataInici DATE NOT NULL,
      dataFi DATE NOT NULL,
      PRIMARY KEY (idCurs)
);

CREATE TABLE Assignatura(
      codiAssignatura INT(6),
      credits INT(2) NOT NULL,
      PRIMARY KEY (codiAssignatura)
);
```

```sql
CREATE TABLE Grup(
      codiGrup INT(6),
      idCurs INT(3) NOT NULL,
      codiAssignatura INT(6) NOT NULL,
      PRIMARY KEY (codiGrup),
      FOREIGN KEY (idCurs) REFERENCES Curs(idCurs),
      FOREIGN KEY (codiAssignatura) REFERENCES
    Assignatura(codiAssignatura)
);

CREATE TABLE Examen(
      idExamen INT(7) AUTO_INCREMENT,
      dataExamen DATE NOT NULL,
      hora TIME NOT NULL,
      aula VARCHAR(9) NOT NULL,
      convocatoria VARCHAR(30) NOT NULL,
      estatExamen VARCHAR(30) NOT NULL,
      codiGrup INT(6),
      dniProfessorResponsable CHAR(9) NOT NULL,
      PRIMARY KEY (idExamen),
      FOREIGN KEY (codiGrup) REFERENCES Grup(codiGrup),
      FOREIGN KEY (dniProfessorResponsable) REFERENCES
    Professor(dniProfessor)
);

CREATE TABLE TipusPregunta(
      tPreg VARCHAR(30),
      PRIMARY KEY (tPreg)
);

CREATE TABLE Pregunta(
      idPregunta INT(8) AUTO_INCREMENT,
      dataPregunta DATE NOT NULL,
      solucio VARCHAR(255),
      definitiva ENUM('Si', 'No') NOT NULL,
      estatPregunta ENUM('Oberta', 'Tancada'),
      tPreg VARCHAR(30),
      dniProfessorAutor CHAR(9) NOT NULL,
      PRIMARY KEY (idPregunta),
      FOREIGN KEY (tPreg) REFERENCES TipusPregunta(tPreg),
      FOREIGN KEY (dniProfessorAutor) REFERENCES
    Professor(dniProfessor)
);
```

```
CREATE TABLE Puntuacio(
      valorPreg INT(2),
      idPregunta INT(8),
      idExamen INT(7),
      PRIMARY KEY (idPregunta, idExamen),
      FOREIGN KEY (idPregunta) REFERENCES Pregunta(idPregunta),
      FOREIGN KEY (idExamen) REFERENCES Examen(idExamen)
);

CREATE TABLE Nota(
      notaPreg INT(2),
      idNota INT(9) AUTO_INCREMENT,
      idPregunta INT(8) NOT NULL,
      idExamen INT(7) NOT NULL,
      dniAlumne CHAR(9) NOT NULL,
      PRIMARY KEY (idNota),
      FOREIGN KEY (idPregunta) REFERENCES Pregunta(idPregunta),
      FOREIGN KEY (idExamen) REFERENCES Examen(idExamen),
      FOREIGN KEY (dniAlumne) REFERENCES Alumne(dniAlumne)
);

CREATE TABLE R_Alumne_Grup(
      dniAlumne CHAR(9),
      codiGrup INT(6),
      PRIMARY KEY (dniAlumne, codiGrup),
      FOREIGN KEY (dniAlumne) REFERENCES Alumne(dniAlumne),
      FOREIGN KEY (codiGrup) REFERENCES Grup(codiGrup)
);

CREATE TABLE R_Professors_Ajudants_Examen(
      dniProfessorAjudant CHAR(9),
      idExamen INT(7),
      PRIMARY KEY (dniProfessorAjudant, idExamen),
      FOREIGN KEY (dniProfessorAjudant) REFERENCES
    Professor(dniProfessor),
      FOREIGN KEY (idExamen) REFERENCES Examen(idExamen)
);
```

CAPÍTOL 5- CLOUD DATABASE

Què entenem per cloud database?

Una "cloud database" o base de dades al núvol és un servei de base de dades creat i al qual s'accedeix a través d'una plataforma en el núvol. Des d'una perspectiva estructural i de disseny, una base de dades al núvol no és diferent d'una que opera en els propis servidors locals d'una empresa. La diferència fonamental recau en el lloc on resideix la base de dades.

Mentre una base de dades local està connectada a usuaris locals a través de la xarxa d'àrea local interna (LAN) d'una corporació, una base de dades al núvol resideix en servidors i emmagatzematge proporcionats per un proveïdor de núvol o base de dades com a servei (DbaaS) i s'accedeix únicament a través de l'Internet. Per a una aplicació de software, per exemple, una base de dades SQL que es troba en les instal·lacions o en el núvol hauria de semblar idèntica.

El comportament de la base de dades ha de ser el mateix ja sigui que s'accedeixi a través de consultes directes, com declaracions SQL, o mitjançant cridades a la API. No obstant això, és possible identificar petites diferències en el temps de resposta. És probable que una base de dades local, a la que s'hi accedeix amb una LAN, proporcioni una resposta un poc més ràpida que una "cloud database", que requereix un viatge d'anada i tornada a Internet per a cada interacció amb la base de dades.

Característiques clau de les "cloud database":

- És un servei de base de dades creat i al qual s'hi accedeix a través d'una plataforma en el núvol.
- Permet als usuaris empreserials allotjar bases de dades sense la necessitat de comprar hardware dedicat.
- Pot ser administrada per l'usuari o ofert com un servei i administrat per un proveïdor.
- Pot admitir bases de dades relacionals i bases de dades NoSQL.
- S'hi accedeix a través d'una interfície web o una API proporcionada pel proveïdor.

Tipus de bases de dades al núvol

Existeixen dos models d'entorn de bases de dades al núvol: tradicional i bases de dades com a servei (DbaaS).

En un model de núvol tradicional, una base de dades s'executa a la infraestructura d'un departament TI amb una màquina virtual. Les tasques de supervisió i gestió de bases de dades corresponen al personal informàtic de l'organització.

El model DbaaS és un servei de subscripció basat en tarifes en què la base de dades s'executa a la infraestructura física del proveïdor de serveis. Normalment hi ha diferents nivells de serveis disponibles. En una disposició DbaaS clàssica, el proveïdor manté la infraestructura física i la base de dades, deixant al client gestionar el contingut i el

funcionament de la base de dades. Alternativament, un client pot configurar un acord d'allotjament gestionat, en el qual el proveïdor gestiona el mantenimnet i la gestió de la base de dades.

Model de dades

És important diferenciar entre bases de dades en el núvol que són relacionals en oposició a las no relacionals (NoSQL):

- **Bases de dades SQL**: són un tipus de bases de dades que es poden executar en el núvol (ja sigui com una imatge de màquina virtual o com un servei, depenent del proveïdor). Les bases de dades relacionals tenen baixa escalabilitat, ja que no foren nativament dissenyades per entorns en el núvol, tot i que els serveis en el núvol basats en SQL estan tractant de fer front a aquest problema.

  Aquestes bases de dades s'organitzen en tables que, a la vegada, estan compostes per files (on figura cada un dels registres de la taula i columnes (on es posen els camps que defineixen els registres). Aquest tipus de base de dades es solen emprar per a l'emmagatzematge d'informació estructurada o amb un format coherent.

- **Bases de dades NoSQL**: són un altre tipus de bases de dades que poden executarse en el núvol. Les bases de dades NoSQL estan dissenyades per a servir càrregues de lectura/escriptura pesades i poden escalar fàcilment amunt i avall i, per tant, són més adequades de manera nativa per executar-se en el núvol. No obstant això, la majoria d'aplicacions actuals es construeixen al voltant d'un model de dades SQL, de manera que treballar amb bases de dades NoSQL sovint requereix una reescriptura completa del codi de l'aplicació.

  Aquestes bases de dades no empren el model de taula de les anteriors. S'empren per emmagatzemar informació semiestructurada o directament no estructura. Permeten emmagatzemar volums molt més grans d'informació.

Beneficis

Els principals avantatges de les bases de dades al núvol són:

- Són escalables. Els proveïdors de serveis poden augmentar els recursos o proporcionar més espai d'emmagatzematge en funció de les necessitats del client.
- Mantenen la integritat de la informació. Aquest tipus d'emmagatzematge d'informació pot replicar instàncies de les bases de dades, permeten així l'accés concurrent d'usuaris sense que les dades perdin integritat.
- Estalvi d'espai físic. Les empreses que contracten aquest tipus de servei no necessiten instal·lar infraestructurers en la seva empresa, tot queda emmagatzemat en els servidors del proveïdor.
- Estalvi de costos. Mitjançant l'eliminació d'una infraestructura física propietat i operada per un departament d'informàtica, es poden aconseguir estalvis

significatius amb la reducció de despeses de capital, menys personal, disminució dels costos d'operació elèctrics i una menor quantitat d'espai físic necessari.
- Eviten problemes. Alguns d'ells molt típics en les bases de dades tradicionals, com la impossibilitat d'accedir a l'informació si es cauen els servidors locals.
- Augmenten la seguretat. Normalment els proveïdors d'emmagatzematge al núvol són empreses de reputació provada en el món digital i incorporen solucions per evitar bretxes de seguretat o pèrdues d'informació.
- Recuperació de desastres. En cas d'un desastre natural, una fallada de l'equip o un tall de coreent, les dades es mantenen segures mitjançant còpies de seguretat en servidors remots.
- Finalment, es pot accedir a les bases de dades des de qualsevol lloc i amb qualsevol dispositiu, sempre que es tingui connexió a Internet i les claus d'accés.

Desavantatges de les bases de dades al núvol

- Un risc que cal tenir en compte és la Confidencialitat i el Compliment: des del país on s'allotjaran les vostres dades (i, per tant, quines normatives de protecció de dades hi ha) fins a confiar en el sistema de seguretat d'una altra persona, les empreses poden perdre el control de les seves pròpies dades.
- Pot semblar obvi, però per accedir a una base de dades al núvol és imprescindible una connexió a Internet. Si passa alguna cosa i una empresa perd la connexió a Internet, també perd l'accés a les seves dades.

Migració de bases de dades locals al núvol

Una base de dades local pot migrar a una implementació al núvol. Existeixen nombroses raons per fer-ho, entre les quals destaquen les següents:

- Es poden eliminar les infraestructures físiques locals necessàries per a l'emmagatzematge d'informació.
- És possible tenir una base de dades estructurada i organitzada tot i que el departament IT de la companyia no tengui l'experiència adequada.
- Millora l'eficiència del processament, especialment quan les aplicacions i les analítiques que accedeixen a les dades també resideixen en el núvol.
- Es poden reduir els costs relatius al manteniment de bases de dades en local:
    - Reducció de personal informàtic intern.
    - Eliminació d'equips físics.
    - Els preus del servei al núvol disminueixen contínuament.
    - Només es paga per els recursos consumits, conegut com a preus de pagament per ús.

La reubicació d'una base de dades al núvol pot ser una manera eficaç d'habilitar encara més el rendiment de les aplicacions empresarials com a part d'una implementació més àmplia de software com a servei. En fer-ho, es simplifica els processos necessaris per fer que la informació estigui disponible a través de connexions a Internet. La consolidació d'emmagatzematge també pot ser un avantatge de traslladar les bases de dades d'una empresa al núvol. Les bases de dades de diversos departaments d'una gran empresa, per

exemple, es poden combinar al núvol en un únic sistema de gestió de bases de dades allotjats.

Base de dades: local o en el núvol?

Cada vegada hi ha més proveïdors que ofereixen l'opció de tenir la base de dades en el núvol. Aixo es deu al augment en la demanda. Tot i això, encara hi ha empreses que decideixen tenir la base de dades en local.

Les raons per les que es decideix tenir la base de dades en local poden ser: perquè els hi compensa, ja que tenen servidors propis on allotjar-les sense que els suposi un cost extra; perquè els permet tenir-ho tot baix control; i altres, perquè pensen que tenint les dades en la seva pròpia empresa tindran una major seguretat. Això darrer és molt relatiu, ja que si no tenen el hardware ben protegit la seguretat es podria veure compromesa.

EXEMPLES DE BASES DE DADES AL NÚVOL

1- **Amazon Web Service (AWS)**

Amazon s'ha convertit en el líder del mercat en l'espai DBaaS. Ofereix serveis addicionals de gestió de dades com Redshift, un magatzem de dades i Data Pipeline, que és un servei d'integració de dades per facilitar la gestió de dades. Les ofertes actuals d'Amazon inclouen:

- **Amazon RDS:** el servei de base de dades relacional d'Amazon s'executa en instàncies de servidor Oracle, SQL o MySQL.
- **Amazon SimpleDB:** es tracta principalment d'una base de dades sense esquemes que està destinada a gestionar càrregues de treball més petites.
- **Amazon DynamoDB:** correspon a les bases de dades NoSQL (SSD), capaços de replicar automàticament les càrregues de treball en tres zones de disponibilitat.

Punts forts: moltes característiques, fàcil d'utilitzar, no suportn i documentació.
Punts febles: no molt personalitzable, temps d'inactivitat segons la programació d'Amazon.

2- **Oracle Database**

Oracle Database ofereix a les empreses tecnologia de bases de dades a escala empresarial emmagatzemada al núvol. Tot i que la seva primera oferta és bastant completa, l'oferta de la generació 2 té un rendiment conseqüentment superior amb controls de seguretat i govern amplis.

La migració de dades també es cobreix amb una solució dedicada i un servei d'atenció al client estricte en cas que sorgeixi algun problema tècnic o pregunta.

Punts forts: interfície intuïtiva, fàcil d'utilitzar, assistència al client sòlida.
Punts febles: sense versió gratuïta, sense accés mòbil, car per a petites empreses.

3- Microsoft Azure

Azure és una plataforma de computació en núvol per a la creació de VM, la construcció i execució d'aplicacions web, aplicacions de client intel·ligents i serveis web XML. Actualment compta amb la infraestructura global més gran i forta, amb 55 regions, més que qualsevol altre proveïdor de núvol.

Un punt important que cal tenir en compte és que Microsoft ofereix la gamma més àmplia de software que necessita una empresa moderna avui en dia. Aixo pot permetre crear un ecosistema enorme que tingui les mateixes arrels, amb un sol lloc on anar amb les preguntes o problemes que hi pugui haver.

Punts forts: solució integral, bona seguretat i un ecosistema fort.
Punts febles: servei al client d'Iffy i no és fàcil d'emprar.

SERIA ADEQUAT PER AL CAS D'AQUESTA PRÀCTICA?

Tenint en compte els beneficis que suposa una base de dades al núvol seria una bona opció emprar-la en el cas d'aquesta pràctica.

S'evitarien problemes com la caiguda dels servidors locals, s'estalviaria l'espai físic que suposen aquests servidors i es reduirien els costs d'operacions elèctriques i del personal de manteniment.

A més, aquestes bases de dades són escalables i permeten replicar instàncies de les bases de dades, permetent així l'accés concurrent dels usuaris sense que les dades perdin integritat. També inclouen una bona recuperació davant de desastres. En cas d'un desastre natural, una fallada de l'equip o un tall de coreent, les dades es mantenen segures mitjançant còpies de seguretat en servidors remots.

Respecte al nivell de seguretat, normalment els proveïdors d'emmagatzematge al núvol són empreses de reputació provada en el món digital i incorporen solucions per evitar bretxes de seguretat o pèrdues d'informació.

Tot i això, cal tenir en compte si l'Escola Politècnica ja té els servidors locals. En aquest cas, s'hauria d'estudiar si és rentable dur a terme la migració de la base de dades local al núvol o si és convenient mantenir-la en local. Per fer la decisió s'ha de valorar el cost que suposa mantenir aquests servidors, el temps que fa que estan en funcionament i si el seu funcionament és òptim. Si els servidors s'han instal·lat fa poc, a nivell econòmic probablement no sigui rentable fer el canvi, però si els servidors ja tenen uns anys i s'està considerant renovar-los o fer alguns canvis pentura és el moment de considerar dur a terme una migració a una base de dades al núvol.

REFERÈNCIES

https://ayudaleyprotecciondatos.es/bases-de-datos/en-la-nube/

https://en.wikipedia.org/wiki/Cloud_database

https://searchcloudcomputing.techtarget.com/definition/cloud-database

https://www.ibm.com/cloud/learn/what-is-cloud-database

https://www.ticportal.es/glosario-tic/base-datos-database#:~:text=Las%20razones%20por%20las%20que,tendr%C3%A1n%20una%20mayor%20seguridad%20TIC

CONCLUSIÓ

Durant el desenvolupament d'aquesta pràctica no hem trobat gairebé cap problema. En el model conceptual hem tengut alguna dubte en les relacions entre ALUMNE, GRUP, CURS, ASSIGNATURA i EXAMEN, especialment degut a la convocatoria d'un examen. No sabiem si introduir una classe CONVOCATORIA entre GRUP i EXAMEN però hem decidit que era millor deixar-ho com a un atribut d'examen.

Respecte al model relacional i als diagrames d'estat tampoc hem tengut quasi cap problema. La majoria dels diagrames d'estat eren simples i el d'examen, tot i ser llarg bastant intuitiu. El que més dubtes ens ha generat ha estat el diagrama d'estats de Pregunta i l'estat DEFINITIVA que no sabíem si posar-lo o no, al final l'hem deixat.

Respecte a SQL no ha hagut cap problema i sobre l'estudi pràcticament tampoc. L'únic punt amb algun dubte ha estat en la valoració de si la base de dades al núvol seria adequada per el cas d'aquesta pràctica o no.