

clau primaria : Delegacio - nom , Ciutat - nom
una ciutat una sola delegació
una delegació ha d'existir en una ciutat

11.2 Conductors i clients

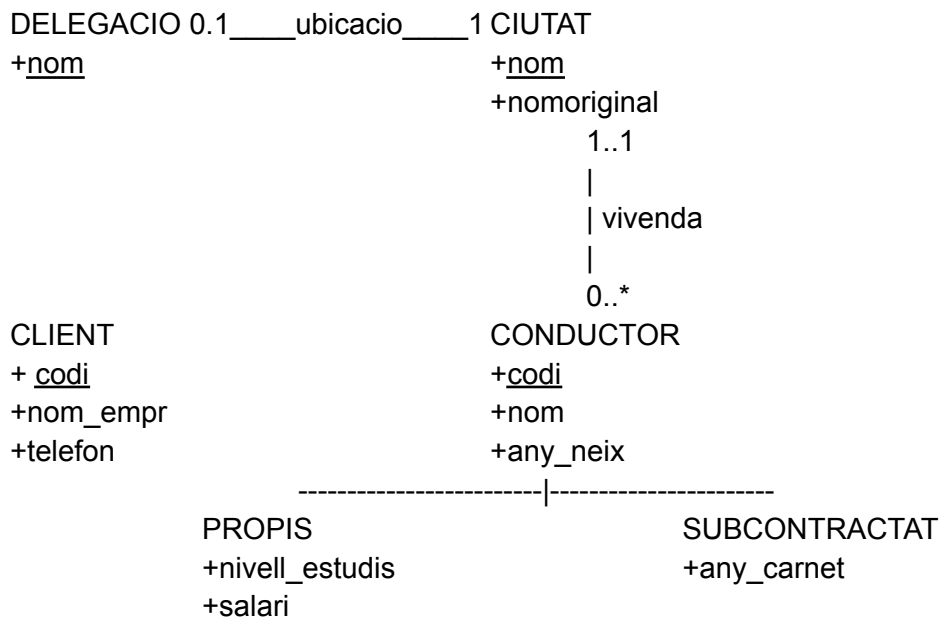
Dels conductors contractats per l'empresa es vol disposar de la informació següent: codi intern (és un nombre donat per l'empresa, amb el qual s'identifiquen els conductors), nom i any de naixement. Els conductors es divideixen en dos grans grups:

- Propis, dels quals interessa conèixer el seu nivell d'estudis (codi d'un caràcter) i el seu salari.
- Subcontractats, dels quals interessa l'any d'obtenció del carnet de conduir.

Els conductors no estan associats a cap delegació. Cada conductor viu en una ciutat.

Els clients solen ser grans empreses multinacionals. S'identifiquen per un codi de client.

De cada client es vol disposar del seu nom d'empresa i telèfon.



Clau primaria: Client - codi , Conductor - codi

Subcategories de Conductor: Propis , Subcontractat

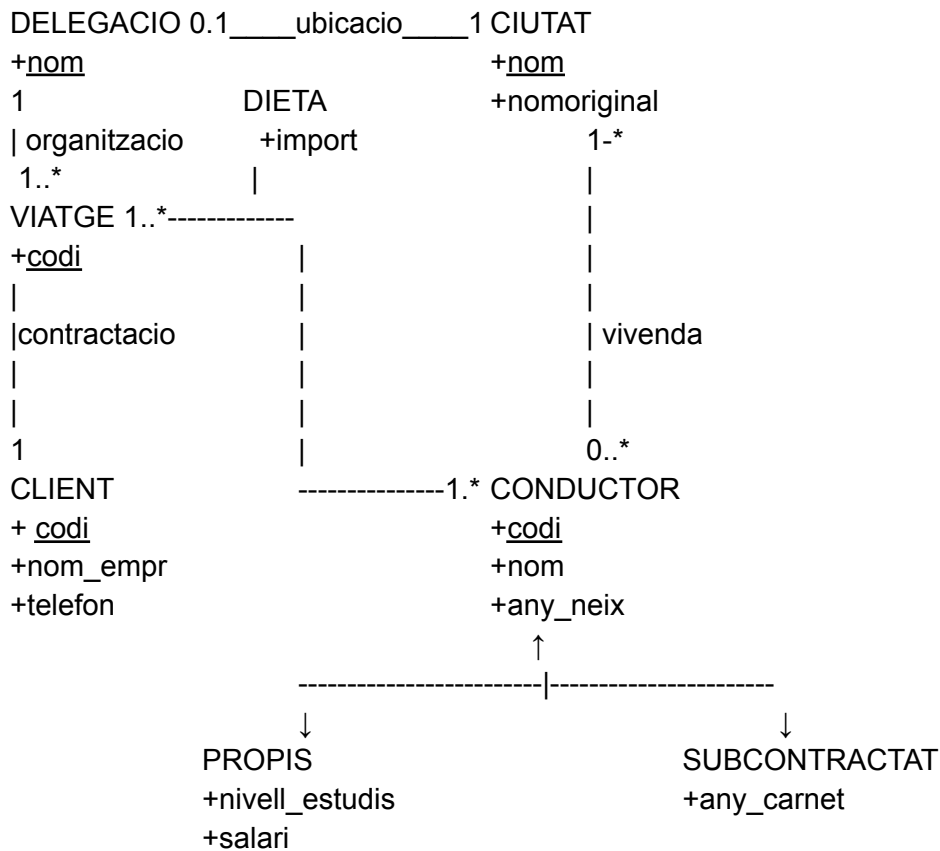
11.3 Activitat de transport

Els viatges que efectua l'empresa no estan estandarditzats (al contrari de les agències de viatges turístics). S'organitzen un a un segons les necessitats puntuals dels clients i són organitzats per les delegacions.

Cada delegació identifica cada viatge amb un codi numèric.

En un viatge hi poden participar més d'un conductor (la normativa legal exigeix que en els viatges llargs hi hagi més d'un conductor, i que aquests es vagin tornant).

Per a cada viatge es vol saber quin conductor o conductors l'han realitzat i les seves dietes. A cada conductor se li assigna una dieta per cada viatge concret (en un mateix viatge poden cobrar diferent import els diferents conductors i el mateix conductor pot tenir dietes diferents en diferents viatges).



Viatge- identitat feble- delegació identifica el viatge no es independent

Claus primaria: Viatge - codi - identitat feble organitzats per delegacions - PK

Esquema relacional:

CLIENT (codi INT AUTOINCREMENT PK,
nomemp TEXT NOT NULL,
telefon TEXT NOT NULL UNIQUE)

CIUTAT (nom TEXT PK,
nomoriginal TEXT NOT NULL)

DELEGACIO (nom TEXT PK,
ubicacio TEXT FK REFERENCES (CIUTAT.nom) NOT NULL)

CONDUCTOR (codi INT AUTOINCREMENT PK,
nom TEXT NOT NULL,
any_neixament DATE NOT NULL ,
vivenda TEXT FK REFERENCES (ciutat.nom) NOT NULL)

PROPIS (codi INT PK FK REFERENCES (conductor.codi),
nivell_estudis CHAR NOT NULL,
salari FLOAT NOT NULL)

SUBCONTRACTAT (codi INT PK FK REFERENCES (conductor.codi),
any_carnet DATE NOT NULL)

VIATGE (codi INT,
organitzacio TEXT FK REFERENCES (delegacio.nom), PK,
codiclient INT FK REFERENCES(client.codi) NOT NULL)

DIETA ({codiconductor INT FK REFERENCE(conductor.codi),
codiviatge INT FK REFERENCES (viatge.codi),
organitzacio TEXT FK REFERENCE (viatge.organitzacio) } } PK,
import FLOAT NOT NULL)

3 Comptes bancaris

Considerem un context d'activitat bancària representat mitjançant una única relació, amb els següents atributs:

- número d'un compte bancari; acc_id
- classe del compte (corrent, estalvis, llibreta...): type
- saldo; balance
- nom del titular del compte (un mateix compte pot tenir diversos titulars, que apareixen documentats en tuples diferents); owner
- adreça, telèfon i NIF del titular.

Alguns titulars apareixen amb noms diferents; sabem que es tracta del mateix titular quan el seu NIF coincideix.

El telèfon és fix, no mòbil; per tant, persones que visquin en la mateixa direcció tenen també el mateix telèfon. address, phone, owner_id

Taula antiga accounts:

accounts(acc_id,type,balance,owner,owner_id, address, phone)

Dependències funcionals:

owner_id -> address, phone (persona només viu en un lloc)

phone -> address

address->phone (telefon fixe i adreça fixe)

NO address, phone -> owner_id (mateixa adreça pot viure +d'una persona)

NO owner_id-> owner (poden tenir varis noms)

NO owner ->owner_id (un nom no determina el id, problema si noms iguals)

acc_id ->balance, type (determinats per la account)

acc_id, owner_id -> owner (dif account amb mateix un owner id tenen noms diferents)

TAULES:

R1 (owner_id,address) PK: owner_id FK: adress=R2.address

R2 (address,phone) PK: address, UNIQUEK:phone

R3 (acc_id, balance,type) PK: acc_id

R4 (acc_id, owner_id,owner) PK:acc_id, owner_id FK: acc_id=R3.acc_id,
owner_id=R1.owner_id

nom taules:

R1: titulars (owner_id,address) PK: owner_id FK: adress=Adreces.address

R2: adreces (address,phone) PK: address, UNIQUEK:phone

R3: comptes (acc_id, balance,type) PK: acc_id

R4: contractes (acc_id, owner_id,owner) PK:acc_id, owner_id FK: acc_id=Comptes.acc_id,
owner_id=titulars.owner_id

NORMALITZAT AMB BCNF - totes dependències en forma de CLAU

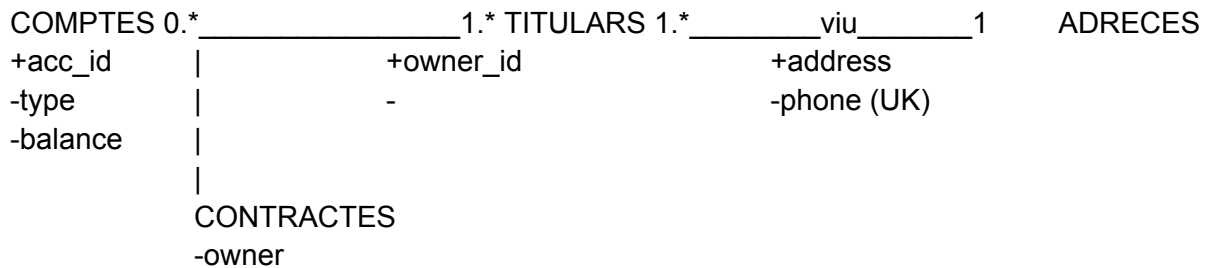
(owner_id,phone,address,acc_id)

i 3NF

(Clau forana es la clau primària de la taula destí)

(UNIQUE KEY-quan ja tens una PRIMARY KEY ja establerta pero l'atribut es pot comportar com una clau de la taula)

UML:



+ = Pk

Variants:

1. cada compte té un únic titular
2. cada persona només pot ser titular d'un compte
3. cada persona només pot ser titular d'un compte de cada classe
4. una persona no té una única direcció sinó potser diverses (Personal, laboral, segona residència...) i a més en cadascuna d'elles pot haver-hi diversos telèfons

2 Felicitacions d'aniversari

El nostre amic John ara viu a Barcelona, però ha viscut en molts països i coneix moltes llengües, i té una enorme llista d'amics repartits per tot el món. Té una col·lecció de missatges de 'feliç aniversari', curiosament redactats perquè qualsevol missatge serveixi per a qualsevol persona, i va afegint nous missatges de tant en tant per repetir menys. Té muntat un sistema automàtic que envia un e-mail de felicitació d'aniversari el dia adequat a la persona adequada, tots els anys, triant aleatòriament el text del missatge.

Missatge - M

Persona - P

Data aniversari - D

Any - A

Email - E

Països - PA

Idiomes - I

$P \rightarrow D$

$P \wedge A \rightarrow M$

$P \wedge A \rightarrow E$

$E \rightarrow P, A, M$

=>finals:

$P \rightarrow D$

$P A \rightarrow E$

$E \rightarrow P, A, M$

$M \rightarrow I$

R1(P, A) - PK:P

R2(A) - PK:A

R3(P, A, E, M) - PK:E , UK: P, A

R4(M) - PK:M

Variants:

1. John no vol repetir el mateix missatge mai a persones que visquin a la mateixa ciutat, ni tan sols en anys diferents.

Ciutat- C

R1(P, A, C) - PK:P

R2(E, M, P, A) - PK:E , UK: P, A

R3(M,I) - PK:M

R4(M,I) - PK:M

R5(PA)

R6(I)

==> Resultat

R1(P,D)- PK:P

R2(E,C,M,P,A)- PK:E , UK:C,M UK: P,A

Combinació: Ciutat missatge - únic - no cal fer-ho en software

Combinació: Persona any - únic -

R3(M,I)- PK:M

=>finals variant:

$P \rightarrow D, C$

$P A \rightarrow E$

$E \rightarrow P, A, M$

$M \rightarrow I$

$M, C \rightarrow E$

2 Felicitacions d'aniversari

El nostre amic John ara viu a Barcelona, però ha viscut en molts països i coneix moltes llengües, i té una enorme llista d'amics repartits per tot el món. Té una col·lecció de missatges de 'feliç aniversari', curiosament redactats perquè qualsevol missatge serveixi per a qualsevol persona, i va afegint nous missatges de tant en tant per repetir menys. Té muntat un sistema automàtic que envia un e-mail de felicitació d'aniversari el dia adequat a la persona adequada, tots els anys, triant aleatòriament el text del missatge.

Missatge - M

Persona - P

Data aniversari - D

Any - A

Email - E

$P \rightarrow D$

$P \wedge A \rightarrow M$

$P \wedge A \rightarrow E$

$E \rightarrow P, A, M$

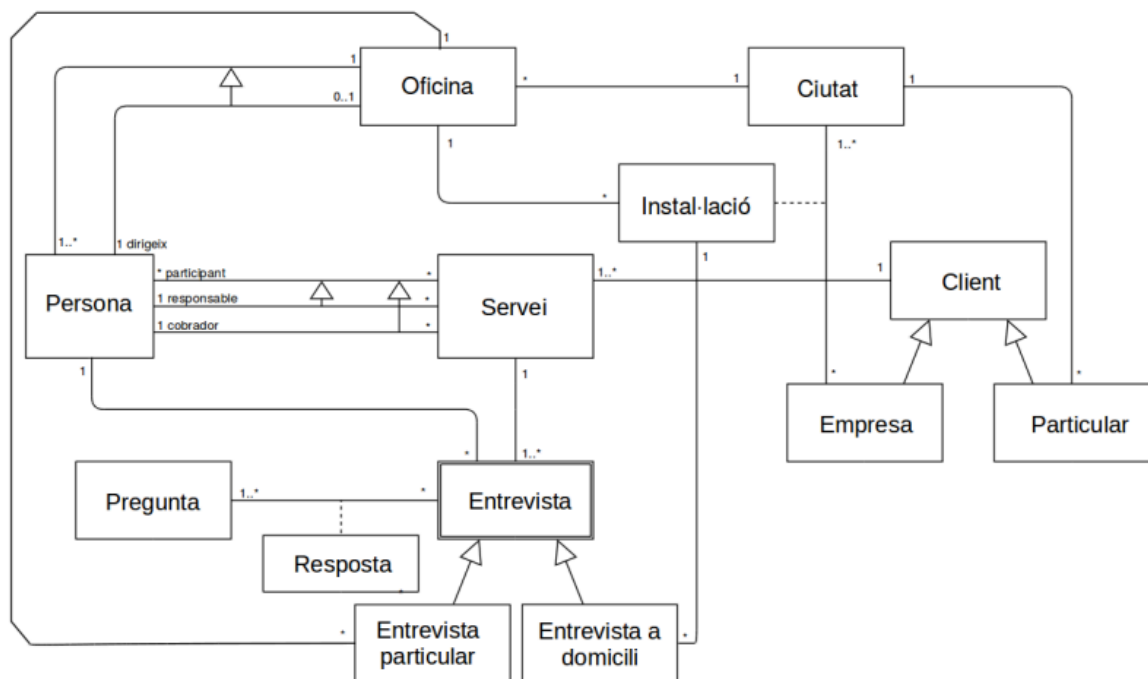
Variants:

1. John no vol repetir el mateix missatge mai a persones que visquin a la mateixa ciutat, ni tan sols en anys diferents.
2. Darrerament John va escàs de temps per preparar nous missatges, i no obstant això la seva llista d'amigues està creixent bastant (igual és per això que va escàs de temps). Com John s'ha adonat que, d'un any per l'altre, gairebé ningú es recorda del missatge, no li importa repetir el mateix missatge a persones diferents de la mateixa ciutat però només si és en anys diferents.
3. Cada amic de John coneix un o diversos idiomes; cada missatge pot tenir diverses versions, cada versió en un idioma. Per descomptat, John no enviarà un missatge escrit en una llengua a una persona que no coneix aquest idioma.

Exercici d'examen Control parcial 2018:

Problema 2 [7,5 punts; màxim: 75 minuts]

Considerem el diagrama següent:



- a) (3,5 punts) Descriu informalment les relacions adequades per transformar aquest diagrama en un esquema relacional, indicant, per a cada relació, a quina part del diagrama correspon, quins són els seus atributs més importants, les claus primàries que trias, les claus alternatives si n'hi ha, i les claus foranes que necessita

Ciutat(nom,país)

Client (NIE)

Particular(NIE FK Client, (nom, país) FK ciutat) -- especialització Client FK NOT NULLS

CLAU FORANA ÉS LA CLAU FORANA DE LA CLAU DESTÍ

- b) (1,5 punts) Per a cadascuna de les claus foranes, indica i justifica la política d'esborrat ON DELETE i la política de actualització ON UPDATE més convenientes

ON DELETE CASCADE// RESTRICCION depèn del problema

ON UPDATE CASCADE (habitual)

- c) (1,5 punts) Considera la relació que correspon a la classe "Instal·lació" i "Oficina", tenint en compte només els atributs Oficina-O, Director-D, Ciutat-C, Empresa-E i Instal·lació-I. Analitza les seves dependències funcionals, i indica si aquest esquema de relació amb els atributs (O,D,C,E,I) està o no en 3NF i si està o no en BCNF. Quan la resposta sigui negativa, proposa diferents formes de normalitzar i explica els avantatges i inconvenients d'aplicar cadascuna d'elles

Dins de la Classe Instal·lació i Oficina

Oficina-O, Director-D, Ciutat-C, Empresa-E i Instal·lació-I

R(O,D,C,E,I) - no està 3NF

Dependències

Oficina té un Director $O \rightarrow D, C$

diverses oficines mateixa ciutat $C \not\rightarrow O$

Empresa client pertany instal·lacions $I \rightarrow E$

Instal·lació determina ciutat $I \rightarrow C$

Instal·lació no determina empresa o ciutat- varies instal·lacions en ciutats $I \not\rightarrow E, C$

Empresa i Ciutat determinen instal·lacions $E, C \rightarrow I$

mateixa oficina tots serveis mateixa instal·lació $I \rightarrow O$