

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Disseny de bases de dades	06.522	22/01/2020	18:30



06.522 22 01 20 EX

Enganxeu en aquest espai una etiqueta identificativa
amb el vostre codi personal
Examen

Fitxa tècnica de l'examen

- Comprova que el codi i el nom de l'assignatura corresponen a l'assignatura matriculada.
- Només has d'enganxar una etiqueta d'estudiant a l'espai corresponent d'aquest full.
- No es poden adjuntar fulls addicionals, ni realitzar l'examen en llapis o retolador gruixut.
- Temps total: **2 hores** Valor de cada pregunta: **Indicat a cada exercici**
- En cas que els estudiants puguin consultar algun material durant l'examen, quins són?
En cas de poder fer servir calculadora, de quin tipus? **CAP**
- Si hi ha preguntes tipus test: Descompten les respostes errònies? **NO** Quant?
- Indicacions específiques per a la realització d'aquest examen:

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Disseny de bases de dades	06.522	22/01/2020	18:30

Enunciats

Exercici 1 (3 punts)

L'empresa North-Park ens ha demanat una base de dades per controlar els aparcaments dels cotxes als seus pàrquings.

Cada pàrquing ve identificat per un nom diferent. A més, es vol informació de la seva adreça i número de places. L'empresa considera que, segons uns certs criteris que no interessen, alguns d'aquests pàrquings són equivalents, amb un cert nivell d'equivalència.

Com és natural, aquests pàrquings disposen de places d'aparcament. Cada plaça te un número diferent dintre de cada pàrquing. Així, el pàrquing "Park-et" te les places número 1, 2 i 3, i el pàrquing "Remei" te les seves places 1, 2 i 3. A més del número de plaça, també interessa conèixer els metres quadrats de la plaça.

Els cotxes (dels quals ens interessa la seva matrícula, marca i model) aparquen a places de pàrquing. A una certa hora d'un dia, només hi ha un sol cotxe aparcad a una determinada plaça de pàrquing. Això si, el mateix cotxe pot aparcar a la mateixa plaça en dies diferents, i fins i tot el mateix dia, a diferents hores. Es vol conèixer el número de minuts que el cotxe ha estat aparcad en una plaça de pàrquing.

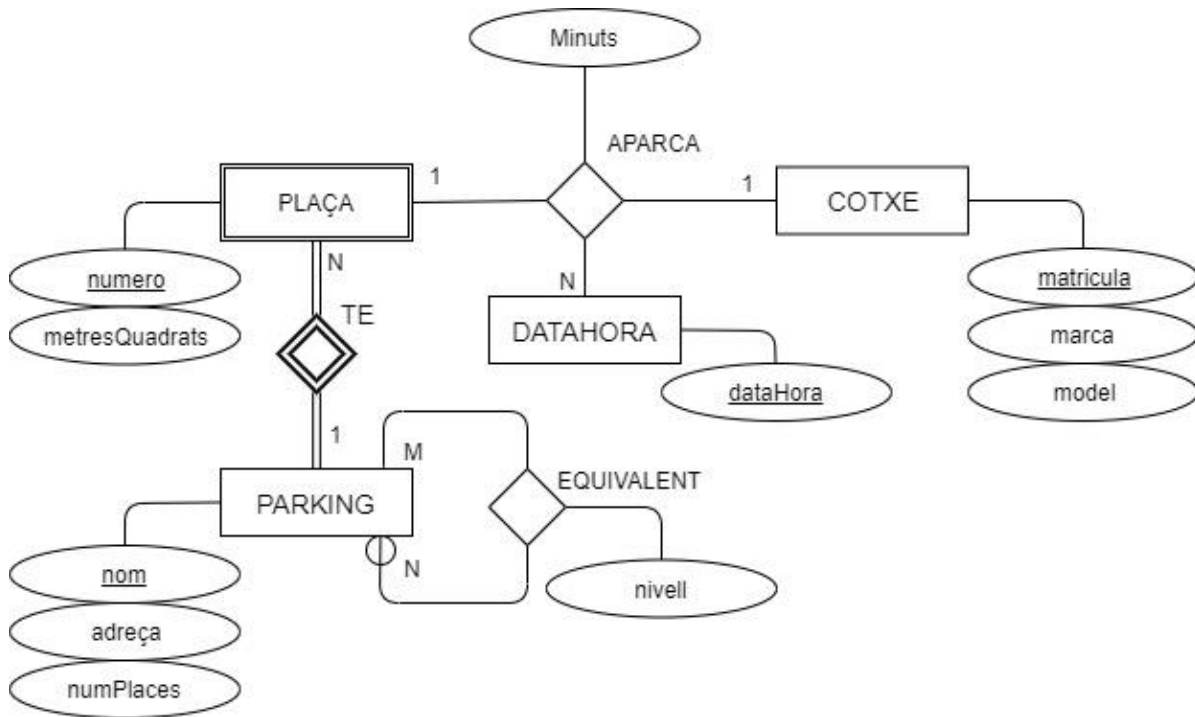
Es demana:

Feu el diagrama E-R que permeti representar la informació descrita per a donar resposta a les necessitats plantejades en el enunciat.

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Disseny de bases de dades	06.522	22/01/2020	18:30

SOLUCIÓ:

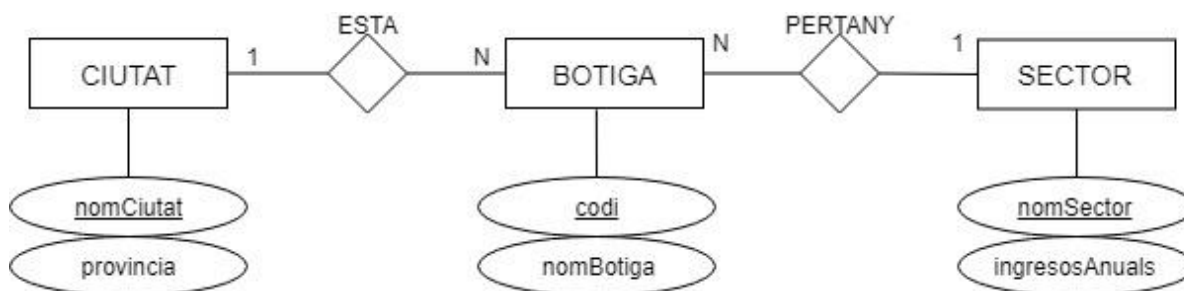


Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Disseny de bases de dades	06.522	22/01/2020	18:30

Exercici 2 (3.5 punts)

A partir del model conceptual descrit pel següent diagrama ER:



Suposant que disposem de MySQL amb el motor de dades Innodb **es demana** fer les instruccions SQL necessàries per a donar resposta a les següents qüestions:

1.- Creeu la taula que representa l'entitat BOTIGA utilitzant SQL. Per això caldrà abans que feu la traducció a model relacional del diagrama ER donat.

```

CREATE TABLE BOTIGA
(
    codi integer,
    nomBotiga varchar (50),
    nomSector varchar(50),
    nomCiutat varchar (50),
    PRIMARY KEY (codi),
    FOREIGN KEY (nomSector) REFERENCES SECTOR(nomSector),
    FOREIGN KEY (nomCiutat) REFERENCES CIUTAT(nomCiutat)
) ENGINE = INNODB;
  
```

2.- Modifiqueu la taula BOTIGA per afegir el camp "adreça" amb el tipus de dades adient.

```

ALTER TABLE BOTIGA add adreça varchar(50);
  
```

3.- Mostreu el nom de totes les botigues del sector 'tèxtil' de la província de Tarragona.

```

SELECT b.nomBotiga
FROM BOTIGA b INNER JOIN CIUTAT c ON b.nomCiutat = c.nomCiutat
WHERE b.nomSector = "Tèxtil" and c.provincia = "Tarragona";
  
```

4.- Mostreu el nom de les províncies que tenen més de 10.000 botigues i quina quantitat en tenen.

```

SELECT c.provincia, count(*)
FROM CIUTAT c inner join BOTIGA B on c.nomCiutat = b.nomCiutat
GROUP BY c.provincia
HAVING count(*) > 10000
  
```

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Disseny de bases de dades	06.522	22/01/2020	18:30

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Disseny de bases de dades	06.522	22/01/2020	18:30

Exercici 3 (2 punts)

Es disposa de la relació VOL amb les dades informades que es mostren a continuació:

VOL (nomPilot, codiAvio, data, DNIPilot, dadesAvio, duradaVol)

VOL					
<u>nomPilot</u>	<u>codiAvio</u>	<u>data</u>	DNIPilot	dadesAvio	duradaVol
Sara Marinas	AV001	01/10/2019	52111111A	Boeing 737, Boeing Industries	180
Sara Marinas	AV001	02/10/2019	52111111A	Boeing 737, Boeing Industries	210
Sara Marinas	AV002	03/10/2019	52111111A	DC10, McDonell Douglas	150
Andreu Sala	AV001	01/10/2019	52222222B	Boeing 737, Boeing Industries	150
Andreu Sala	AV002	02/10/2019	52222222B	DC10, McDonell Douglas	300

Per contextualitzar les dades cal dir que els pilots s'identifiquen pel seu nom, els avions pel seu codiAvio, que la durada del vol està expressada en minuts i que les dades de l'avió inclouen el tipus d'avió i l'empresa que el fabrica. Cal tenir en compte que es voldrà fer gestions a la base de dades per qualsevol de les informacions individuals dels avions que actualment estan emmagatzemades a la columna dadesAvio.

1. Indiqueu en quina forma normal està aquesta relació i perquè.
2. Normalitzeu aquesta relació fins a 3FN, passant per cadascuna de les formes normals intermèdies tot explicant el procés de normalització que feu. A cada pas, cal mostrar totes les relacions resultants de la normalització.

- La relació està en **0FN** perquè té un atribut que no és atòmic (dadesAvio)

- Per tal de normalitzar la relació a **1FN** cal descomposar l'atribut "dadesAvio" en atributs atòmics:

VOL (nomPilot, codiAvio, data, DNIPilot, **tipus**, **fabricant**, duradaVol)

- Per a normalitzar aquesta relació a **2FN** distingir les dependències funcionals plenes i no plenes dels atributs que no formen part de la clau primària amb els que sí que en formen part:

"DNIPilot" té una dependència funcional amb "nomPilot" però es pot considerar que DNIPilot és clau candidata, perquè cada pilot té un "DNIPilot" diferent que l'identifica.

"tipus": té una dependència funcional amb "codiAvio".

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Disseny de bases de dades	06.522	22/01/2020	18:30

“fabricant”: te una dependència funcional amb “tipus” i per transitivitat, te dependència funcional amb “codiAvio”.

“duradaVol”: te una dependència funcional plena.

Així, cal separar en altres relacions els atributs que no tinguin una dependència funcional plena:

VOL (nomPilot, codiAvio, data, DNIPilot, duradaVol)

On {codiAvio} referencia AVIO{codiAvio}

AVIO (codiAvio, tipus, fabricant)

- Per a obtenir la **3FN** a partir de les relacions resultants cal comprovar que cap atribut que no forma part de la clau primària te cap dependència funcional amb cap altre atribut que no forma part de la clau primària.

A la relació VOL no hi ha cap dependència funcional entre “DNIPilot” i “durada Vol” amb el que ja està normalitzada fins a 3FN.

A la relació AVIO tenim la següent dependència funcional:

“fabricant” te una dependència funcional amb “tipus”

Així, cal separar aquest atribut en una altre relació:

VOL (nomPilot, codiAvio, data, DNIPilot, duradaVol)

On {codiAvio} referencia AVIO{codiAvio}

AVIO (codiAvio, tipus)

On {tipus} referencia TIPUSAVIO {tipus}

TIPUSAVIO (tipus, fabricant)

- Per comprovar si alguna relació no està en **BCNF**, cal comprovar que tot determinant també és clau alternativa. En aquest cas, l'atribut “DNIPilot” es pot considerar clau candidata. Cal fer una llista de determinats i claus candidates:

DETERMINANTS:

{nomPilot, codiAvio, data} determina el valor de “DNIPilot” i de “duradaVol”

{DNIPilot, codiAvio, data} determina el valor de “nomPilot” i de “duradaVol”

{nomPilot} determina el valor de “DNIPilot”

{DNIPilot} determina el valor de “nomPilot”

CLAUS CANDIDATES:

{nomPilot, codiAvio, data}

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Disseny de bases de dades	06.522	22/01/2020	18:30

{DNIPilot, codiAvio, data}

Com es pot veure, la llista de determinants i de claus candidates no és idèntica. Així, aquesta relació no està en BCNF. Cal separar aquest atribut en una altre relació:

VOL (nomPilot, codiAvio, data, duradaVol)

On {codiAvio} referencia AVIO{codiAvio}

On {nomPilot} referencia PILOT {nomPilot}

PILOT (nomPilot, DNIPilot)

AVIO (codiAvio, tipus)

On {tipus} referencia TIPUSAVIO {tipus}

TIPUSAVIO (tipus, nomFabricant)

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Disseny de bases de dades	06.522	22/01/2020	18:30

Exercici 4 (1,5 punts)

Respongueu a les següents preguntes:

- a) Per què és necessari el procés de normalització?

Perquè cada informació estigui representada a la taula que li correspon per la seva pròpia semàntica, de forma que no es produeixin problemes d'actualització que deixin la base de dades inconsistent.

- b) Què és una restricció de columna o d'atribut? Posa un exemple d'aquest tipus de restricció.

És una regla lògica que afecta només a un atribut d'una taula. Per exemple:

DNIEstudiant char(9) **not null** vol dir que la columna DNIEstudiant sempre ha d'estar informada.

- c) A l'hora d'eliminar diverses taules d'una base de dades, quin ordre cal seguir? Per què?

Cal eliminar primer les taules que tinguin alguna clau forània cap a una altra taula o taules, i després les que no en tenen cap

- d) Quina diferència hi ha entre una clau alternativa i una clau candidata?

- e) La clau candidata és una clau que podria haver-se triat com a clau primària de la relació però que no ho ha estat, per això es diu alternativa.