

24303 BASES DE DADES

LLENGUATGE DDL i DML

Pràctica 1 - UPF RIDERS

Objectius: En aquesta pràctica es cobreixen diversos aspectes del curs en particular:

- 1) Ser capaços d'escriure comandes de creació de taules (Data Definition Language - DDL).
- 2) Ser capaços de crear i inserir registres a cada taula de manera que l'estat de la base de dades sigui consistent i es puguin realitzar les consultes plantejades.
- 3) Ser capaços d'escriure consultes en SQL (Data Manipulation Language – DML) que corresponguin al seu enunciat en llenguatge natural.

Es realitzarà en grups de DUES o TRES persones. La pràctica **es desenvoluparà en dues sessions amb el vostre professor de pràctiques** que donaran les indicacions necessàries per al seu desenvolupament.

És possible que s'hagi d'utilitzar temps d'estudi addicional per poder-la dur a terme íntegrament. Es recomana als alumnes provar detalladament tot el material abans de lliurar-ho.

Nota: La universitat ha posat a disposició de l'alumnat el software necessari instal·lat degudament als ordinadors de laboratori de la facultat per a poder realitzar les pràctiques amb total garantia. Afegidament, s'han proporcionat guies i manuals d'instal·lació del programari per als que vulguin instal·lar-s'ho a les seves màquines personals.

Metodologia: L'enunciat de la pràctica cal anar llegint-lo en ordre, seguint pas a pas les instruccions que es van donant. A la primera sessió es proporcionarà tota la informació per crear i omplir la base de dades. L'estudiant haurà de crear un fitxer SQL different per la creació (**creates_upf_riders.sql**) i la càrrega de dades (**inserts_upf_riders.sql**) a la base de dades. A la segona sessió, es donarà la informació sobre totes les consultes que s'hauran d'implementar en un fitxer nou (**queries_upf_riders.sql**).

Instruccions d'entrega: és important seguir cada punt de les instruccions de lliurament. Qualsevol desviació d'aquestes instruccions podrà esdevenir en el fet que la pràctica no serà corregida, i avaluada amb un 0.

Recordeu, verificar que :

- tots els "scripts" funcionen correctament ABANS de lliurar.
- els noms de les taules i els camps han de ser els mateixos que s'han especificat al punt 1.
- els arxius han de trobar-se en codificació de caràcters UTF-8
- s'ha de respectar els noms dels arxius donats

S'haurà de lliurar els fitxers **creates_upf_riders.sql**, **inserts_upf_riders.sql**, i **queries_upf_riders.sql**, tots en un ZIP. El fitxer ZIP a lliurar s'haurà d'anomenar amb el NIA de cada membre del grup (per exemple NIA1_NIA2.ZIP per a grups de 2 persones). Hi haurà un link a l'aula global per fer el lliurament. **NOMÉS UN MEMBRE DEL GRUP ES FARÀ RESPONSABLE DE LLIURAR EL TREBALL.**

Data de lliurament: 23 de novembre de 2021 a les 23:59

UPF RIDERS - Part I - Primera Sessió de la Pràctica 1

1. Introducció

En aquesta pràctica implementarem una solució senzilla de base de dades en MySQL per a poder representar part de les dades d'una empresa fictícia de repartiment de productes a diferents negocis de la ciutat. En concret, en guardarem les dades de les rutes de repartiment.

L'empresa l'hem anomenat "UPF Riders" i en voldrem guardar les dades dels diferents locals de l'empresa de repartiment (*headquarters*), els vehicles dels quals disposen, els repartidors (*riders*) i els negocis on entregar les comandes.

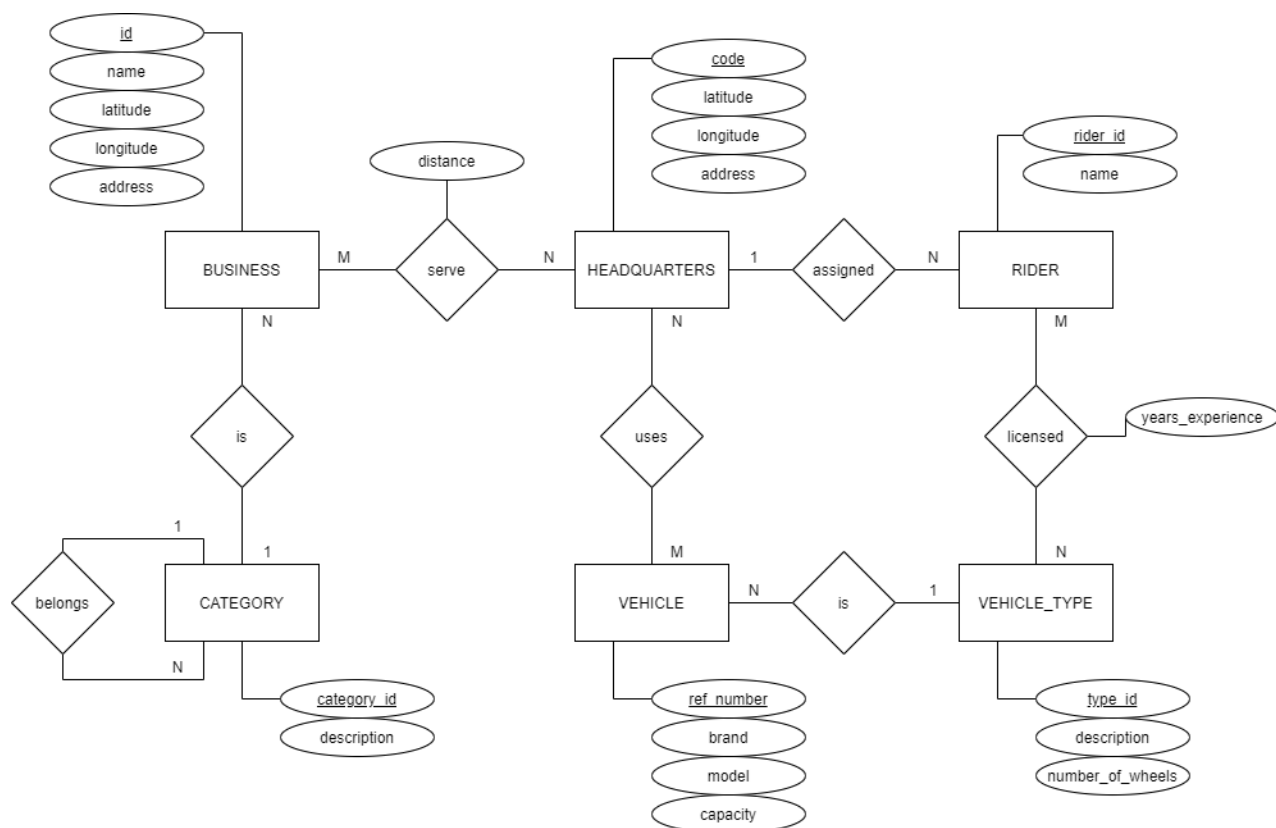
Per a cada local de l'empresa, en voldrem guardar la seva localització en forma de latitud-longitud i l'adreça postal física. Per tal d'identificar-los individualment, usarem un valor enter autoincremental. Cada local pot servir comandes a diferents negocis de la ciutat però al mateix temps, els diferents locals poden rebre comandes provinents de diferents locals de l'empresa repartidora en funció de necessitat. Per tal de programar les rutes de repartiment de forma òptima, ens interessa saber la distància en metres entre els diferents locals de l'empresa i els negocis de destí.

Els clients de l'empresa són tot tipus de negocis repartits per la ciutat, dels quals en guardarem un identificador únic, el nom, la latitud-longitud i l'adreça postal. Cada client el podem catalogar amb un tipus de negoci. Per aquest motiu en voldrem saber la descripció del tipus i els identificarem amb un nombre únic. Els tipus de negocis estan jerarquitzats de tal manera que d'un tipus de negoci en depenen altres tipus de negoci (un restaurant pot ser de categoria italià, rus... però els dos coincideixen que són negocis de tipus restaurant).

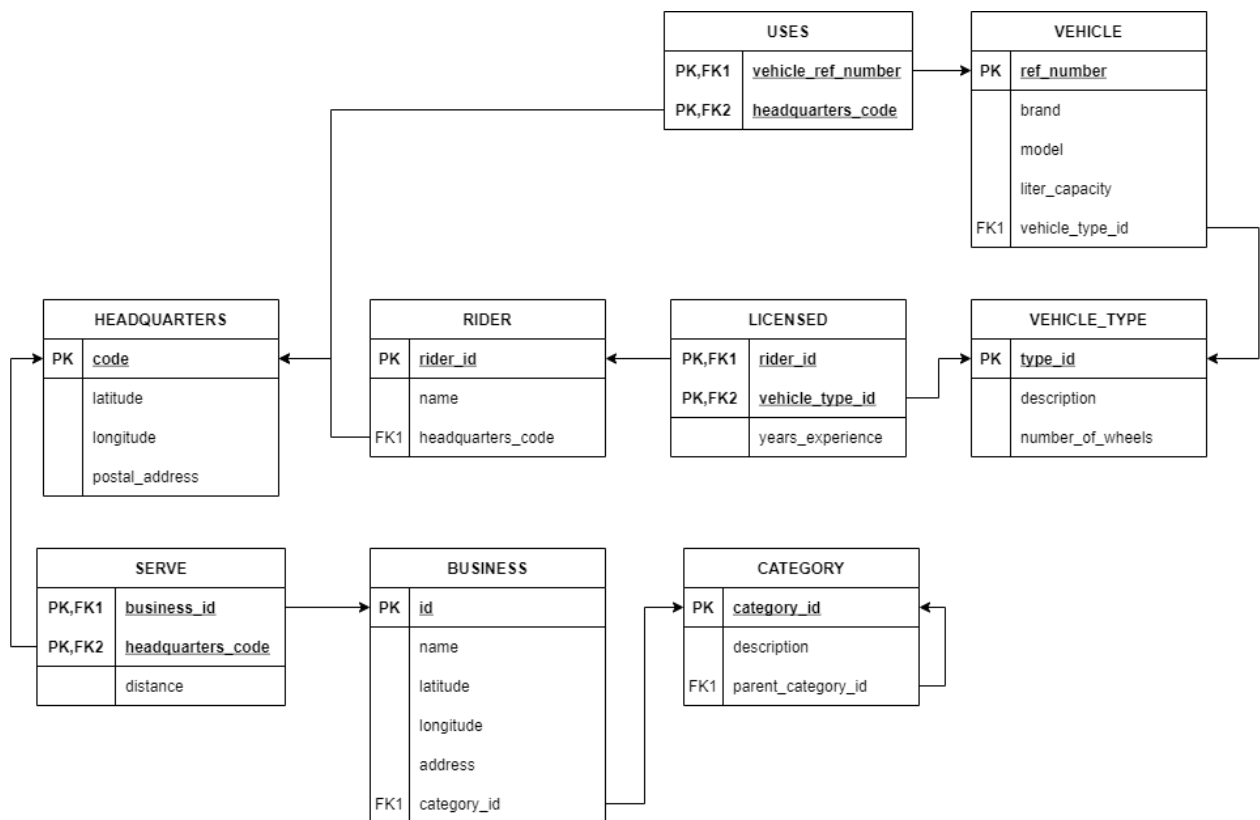
L'empresa disposa d'una flota de vehicles de diferents tipus que identificarem amb un valor enter únic per a cada un i pels quals en guardarem la marca, el model i la capacitat en litres del seu malater. A més a més, tal com dèiem, cada vehicle és d'un tipus concret (moto, coche, furgoneta, etc.) i per això voldrem catalogar aquests tipus guardant-ne la descripció i el número de rodes. Cada tipus serà identificat per un nombre enter únic.

Cada rider està assignat a un únic local i està acreditat per a conduir diferents tipus de vehicle; per a cadascun d'ells, en voldrem saber quina experiència de conducció en té. Afegidament, voldrem guardar el nom del rider i els identificarem amb ids enters que no es repetiran.

A continuació podem veure el model conceptual Entitat – Relació que es deriva de l'enunciat anterior:



Del Model Conceptual anterior, en podem deduir el següent Model Relacional amb taules:



L'equivalent Model Relacional en format Schema és el següent:

1. **headquarters** (code, latitude, longitude, postal_address)
2. **rider** (rider_id, name, headquarters_code (FK1))
3. **vehicle_type** (type_id, description, number_of_wheels)
4. **vehicle** (ref_number, brand, model, liter_capacity, vehicle_type_id (FK1))
5. **category** (category_id, description, parent_category_id (FK1))
6. **business** (id, name, latitude, longitude, address, category_id (FK1))
7. **serve** (business_id (FK1), headquarters_code (FK2), distance)
8. **licensed** (rider_id (FK1), vehicle_type_id (FK2), years_experience)
9. **uses** (vehicle_ref_number (FK1), headquarters_code (FK2))

2. Creació de la base de dades (20 punts)

A partir de l'estructura de taules proposada, haureu de decidir els tipus de dades i els dominis de valors que han de tenir atributs, les claus primàries (PK) i les claus foranes (FK) de les taules. Tot seguit, haureu d'escriure les ordres SQL (DDL) per crear la base de dades i les taules. Comprova la sintaxi SQL de creació de taules a l'enunciat de la pràctica prèvia (P00).

La seqüència de les ordres ha de ser correcta respectant les restriccions imposades per les claus foranes (FK). És a dir, l'ordre de creació de les taules és rellevant.

Les instruccions de creació de la base de dades s'hauran de lliurar en un fitxer (script) anomenat **creates_upf_riders.sql**. Aquest fitxer haurà de començar, obligatòriament, amb les següents tres instruccions:

```
DROP SCHEMA IF EXISTS upf_riders;  
CREATE DATABASE IF NOT EXISTS upf_riders;  
USE upf_riders;
```

Aquest fitxer, al ser executat, no ha de produir cap error, ja que si conté errors no serà possible corregir la resta de la pràctica. Recomanem afegir la sintaxi IF NOT EXISTS a cada CREATE TABLE per evitar que l'script falli en cas de que la taula a crear ja existeixi. Després de provar una estona, és normal tenir part de les taules de la BD creades i d'altres sense crear. Aleshores serà necessari fer un DROP de la base de dades abans d'intentar crear-la de nou.

El nom de la base de dades a crear haurà de ser **upf_riders**.

Cal indicar les ordres de creació de taules necessàries (separades per “;”).

2.1. Bestreta de correcció

Cada taula creada correctament aporta dos (2) punts a la nota, havent-hi 9 taules en total a crear, la creació de les 9 taules aporta divuit (18) punts.

Els dos (2) punts restants els aporta la reutilització (el fitxer es pot executar múltiples vegades sense generar errors) i la qualitat del fitxer (script).

Casos que poden penalitzar:

- Nom de taula erroni
- Nom d'atribut erroni

- Manca d'un atribut en una taula
- Manca de clau primària
- Manca de clau forana

Tots aquests casos penalitzen amb un màxim de dos (2) punts per taula, és a dir, si a una taula manca una clau forana, aquella taula no puntua, tot i que la resta estigui bé.

Quant a la reutilització del fitxer (script) és crucial entendre bé l'ordre de creació de les taules, per tal de resoldre les dependències i evitar errades.

3. Càrrega de la base de dades (20 punts)

A l'aula trobareu un fitxer anomenat **upf_riders.xlsx**, on trobareu les dades que ha de contenir la base de dades.

A partir d'aquesta informació, s'hauran de crear les comandes INSERT necessàries per tal d'inserir la informació continguda en aquest fitxer dins de les taules.

Podeu generar les sentències de INSERT manualment o auto-generar-les usant formulació d'excel (veure funció CONCATENAR). Trobareu un exemple de com fer-ho dins de l'excel a la pestanya per a la taula *headquarters*.

Nota: No es poden modificar les dades del fitxer excel proporcionat sota cap concepte.

Les instruccions d'inserció de dades s'hauran de lliurar en un fitxer (script) anomenat **inserts_upf_riders.sql**.

Aquest fitxer, al ser executat, no ha de produir cap error, ja que si conté errors no serà possible corregir la resta de la pràctica.

Molt important: Les regles d'integritat referencial que hem definit amb les PK i FK, ens obliguen a inserir les dades a les taules en un determinat ordre per tal de què no es produeixin errors de claus primàries, foranes o restriccions de domini.

Per exemple, no té sentit inserir primer les dades de la taula *vehicle* si prèviament no hem donat d'alta les dades dins de la taula *vehicle_type*. Com que cada vehicle té un tipus de vehicle associat, hem d'informar els diferents tipus de vehicle abans d'inserir els vehicles.

3.1. Bestreta de correcció

Cada taula que contingui totes els registres aporta dos (2) punts a la nota, havent-hi 9 taules en total a crear, la creació de les 9 taules aporta divuit (18) punts.

Els dos (2) punts restants els aporta la reutilització i la qualitat del fitxer (script).

Casos que poden penalitzar:

- Nom de taula erroni
- Nom d'atribut erroni
- Manca d'un atribut en una taula
- Modificació de les dades originals

Tots aquests casos penalitzen amb un màxim de dos (2) punts per taula, és a dir, si tots els registres d'una taula tenen les dades bé, però s'ha alterat un registre, no puntua la taula sencera.

Quant a la reutilització del fitxer (script) és crucial entendre bé l'ordre d'inserció de les taules, per tal de resoldre les dependències i evitar errades.

UPF RIDERS - Part II - Segona Sessió de la Pràctica 1

4. Consultes (60 punts)

(Properament)