



PRÀCTICA 1-FASE 1

Bases de Dades

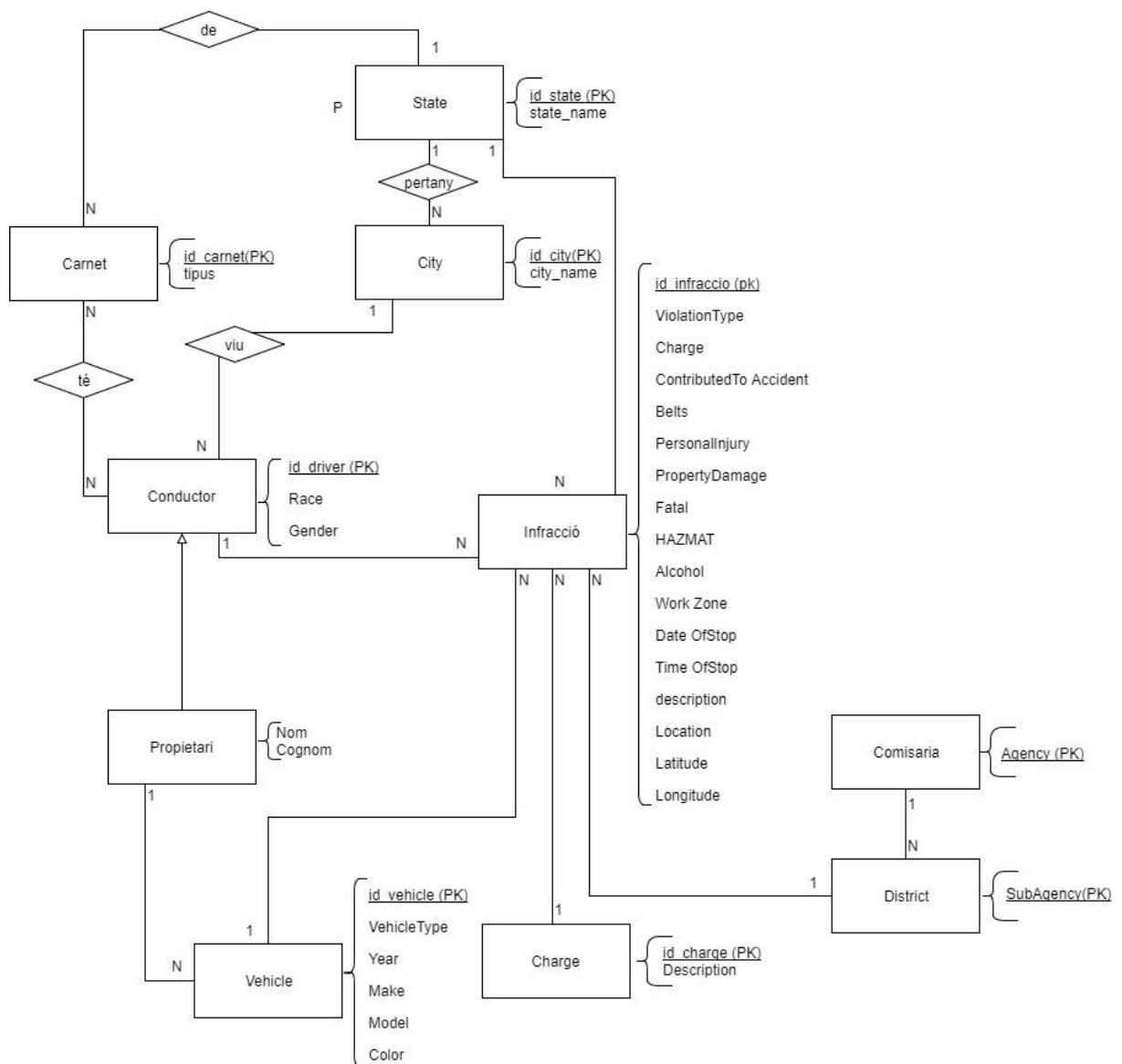
Marc Llort Maulion
Alex Almansa Casanoves

marc.llort.2016
alex.almansa.2016

ÍNDIX

Model Conceptual	2
Model Lògic	3
Estudi de Mercat de SGBD	4
Pregunta	5
Model Físic	6
Charset i Collate	7
Test del procés d'importació.....	10
Conclusions	12

Model Conceptual

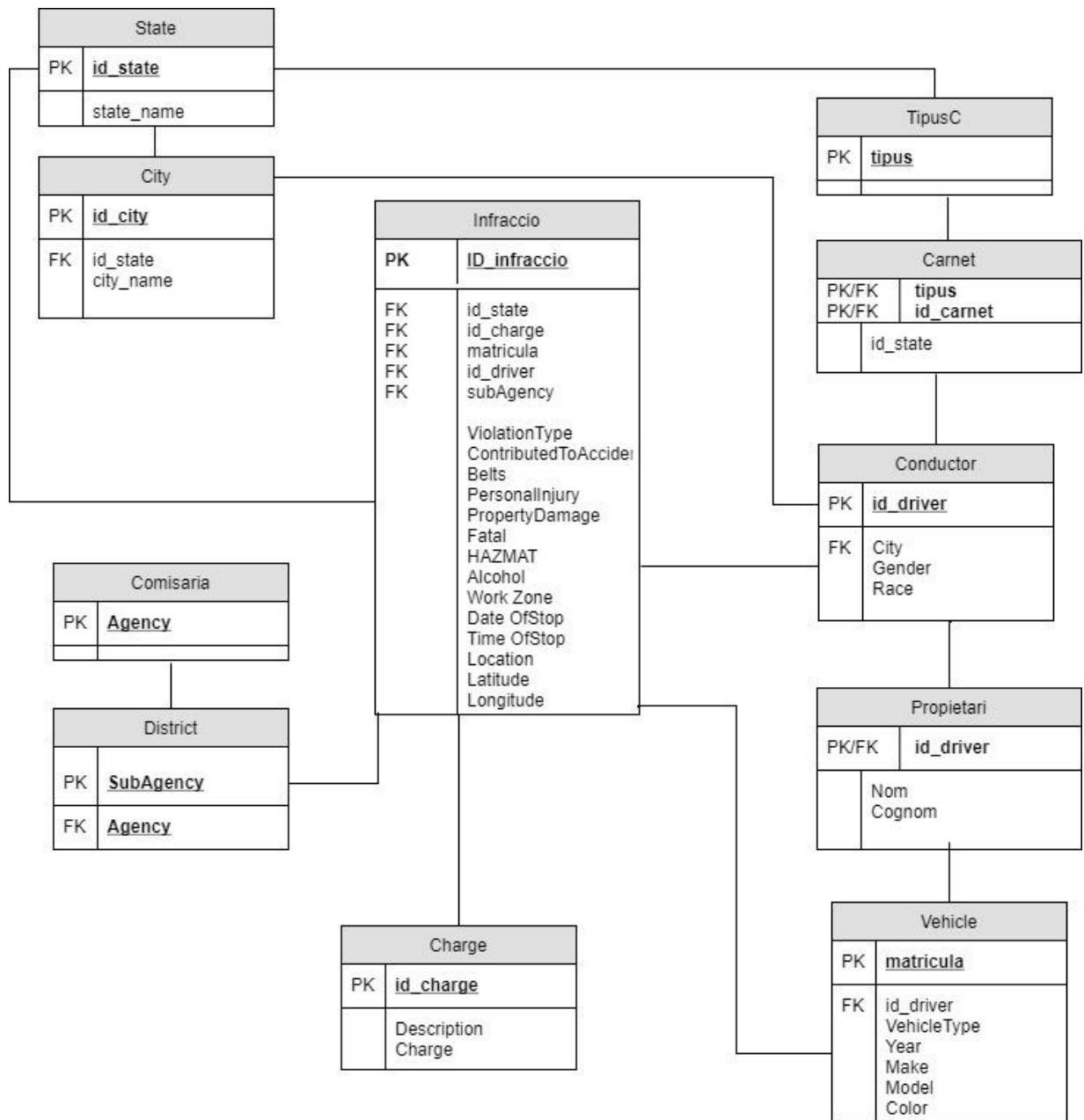


Alhora de realitzar els models conceptual i lògic, hem decidit ajuntar tot a Infracció, ja que son la resta de entitats els que formen Infracció.

A partir de les dades que ens dona el arxiu de tipus, hem anat afegint els atributs necessaris a cada entitat.

Per solucionar el problema de que no sempre el conductor era el propietari del vehicle, hem optat per fer una generalització de propietari, que en el cas de que ho sigui haurà de donar els noms i cognoms.

Model Lògic



En el cas del model lògic hem seguit les regles de canvi de model per canviar de conceptual a lògic.

Estudi de Mercat de SGBD

Després de indagar sobre quins són els sistemes gestors de bases de dades que més s'utilitzen i podrien haver sigut utilitzats per crear la nostra base de dades de la policia, hem trobat que són:

- MySQL
- Microsoft SQL Server
- Oracle
- PostgreSQL
- DB2
- Microsoft Access

Cadascun té les seves particularitats, el que fa que alguns siguin més idonis per unes tasques que per d'altres. Gairebé tots (menys DB2) utilitzen el model relacional.

En el cas de MySQL les seves característiques principals són que es multiusuari i que es el sgbd Open Source més popular del mon. Es molt utilitzada a webs.

Els seus principals avantatges són que és molt ràpid alhora de realitzar operacions, no necessita gaires recursos per generar les bases de dades, i és fàcil de configurar. Com a desavantatge podríem dir que no es gaire intuïtiu, ja que hi ha bastantes utilitats de MySQL no documentades.

Microsoft SQL Server es un sgbd basat en Transact-SQL que permet posar a disposició de molts usuaris, gran quantitat de dades de forma simultània. Pertany a Microsoft.

Els seus principals avantatges són que es molt escalable, estable i segur, suporta transaccions, té interfície gràfica que permet fer us de codis DDL y DML. També permet treballar directament des de el servidor, on el usuari, només accedeix a les dades del servidor però no les té "físicament".

Com a desavantatge és que fa us de molta memòria RAM, té moltes restriccions, és molt car per el que ofereix (en comparació amb Oracle).

Oracle tradicionalment ha sigut el sgbd més important gràcies a ser el més complert i robust.

Destaca per ser molt escalable, suportar transaccions, és molt estable i multiplataforma. Per altra banda, cal dir que es bastant car, el que ha evitat que sigui un estàndard. Si no esta ben configurat pot arribar a ser molt lent, i no hi ha molta informació disponible de com administrar o instal·lar característiques a oracle.

PostgreSQL és un sgbd lliure orientat a objectes que pretén ser la alternativa a MySQL, oracle...

Els seus principals avantatges són que té alta concurrència, per tant un usuari pot modificar una taula mentre un altre hi excedeix a la mateixa, sense bloquejar-se mútuament, té més tipus natius que SQL, és molt estable i bastant ràpid en bases de dades de grans tamanys. Al ser open Source, pots modificar el codi font del programa per adaptar-lo millor a les teves necessitats.

Els desavantatges es que en insercions i actualitzacions és més lent que MySQL, no trobes gaire suport a internet (nomes “foros”) ja que és un projecte de la comunitat, consumeix més recursos que altres alternatives i hi ha algunes sintaxis que no són gaire intuïtives.

DB2, l'únic sistema de la llista no relacional, propietat de IBM. Utilitza XML com a motor, i ens permet emmagatzemar document complets dins el tipus de dada XML per realitzar operacions i busques.

La part positiva d'utilitzar aquest motor és que permet tractar objectes mes grossos, el usuari pot definir dades i funcions, suporta arxius multimèdia.

Com a desavantatge és que és molt car, ja que esta orientat a grans empreses que necessiten molt emmagatzematge, és lent alhora de crear i executar consultes, utilitza molts recursos (RAM).

Pregunta

Han fet bé els policies en sol·licitar l'estructuració de les seves dades en el programari emprat?

La elecció de fer servir PostgreSQL, juntament amb la de MySQL, serien les més acertades.

Això és degut a que ambdues són lliures, i per tant no cal que es gastin els diners dels contribuents, són multiusuari, per tant poden haver molts treballadors de la policia accedint a la base de dades al mateix moment sense que hi hagi problemes.

També és molt important que els dos sistemes són molt escalables, ja que tractant-se d'una comissaria on cada dia hi ha infraccions, és molt important que pugin afegir informació a la base de dades fàcilment.

Model Físic

```
81
82 --Creem taula comisaria
83 DROP TABLE IF EXISTS Comisaria CASCADE;
84 CREATE TABLE Comisaria (
85
86     agency CHAR(4),
87     PRIMARY KEY (agency)
88
89 );
90
91 --Creem taula districte
92 DROP TABLE IF EXISTS District CASCADE;
93 CREATE TABLE District (
94
95     sub_agency VARCHAR(255),
96     agency CHAR(4),
97     FOREIGN KEY (agency) REFERENCES Comisaria(agency),
98     PRIMARY KEY (sub_agency)
99 );
100
101 --Creem la taula State
102 DROP TABLE IF EXISTS State CASCADE;
103 CREATE TABLE State (
104
105     state_name VARCHAR(255),
106     id_state serial,
107     PRIMARY KEY (id_state)
108 );
109
110
111 --Creem la taula City
112
113 DROP TABLE IF EXISTS City CASCADE;
114 CREATE TABLE City (
115
116     city_name VARCHAR(255),
117     id_city serial,
118     id_state integer,
119     FOREIGN KEY (id_state) REFERENCES State(id_state),
120     PRIMARY KEY (id_city)
121 );
122
123
124
125 --Creem taula conductor
126 DROP TABLE IF EXISTS Conductor CASCADE;
127 CREATE TABLE Conductor (
128
129     id_driver bigint,
130     race VARCHAR(255),
131     gender CHAR(3),
132     city integer,
133     PRIMARY KEY (id_driver),
134     FOREIGN KEY (city) REFERENCES City(id_city)
135
136 );
137
138 --Creem la taula Tipus Carnet
139 DROP TABLE IF EXISTS TipusC CASCADE;
140 CREATE TABLE TipusC (
141
142     tipus CHAR(3),
143     PRIMARY KEY (tipus)
144
145 );
146
147
148 --creem la taula Carnet
149 DROP TABLE IF EXISTS Carnet CASCADE;
150 CREATE TABLE Carnet (
151
152     id_driver bigint,
153     id_state int,
154     tipus CHAR(3),
155     FOREIGN KEY (id_driver) REFERENCES Conductor(id_driver),
156     FOREIGN KEY (id_state) REFERENCES State(id_state),
157     FOREIGN KEY (tipus) REFERENCES TipusC(tipus),
158     PRIMARY KEY(id_driver,tipus)
159 );
160
161 --Creem taula propietari
162
163 DROP TABLE IF EXISTS Propietari CASCADE;
164 CREATE TABLE Propietari (
165
166     id_driver bigint,
167     nom VARCHAR(255),
168     FOREIGN KEY (id_driver) REFERENCES Conductor(id_driver),
169     PRIMARY KEY (id_driver)
170 );
171
```

```

172 --Creem taula vehicle
173
174 DROP TABLE IF EXISTS Vehicle CASCADE;
175 CREATE TABLE Vehicle (
176
177     matricula VARCHAR(255),
178     id_prop bigint,
179     vehicle_type VARCHAR(255),
180     year int,
181     make VARCHAR(255),
182     model VARCHAR(255),
183     color VARCHAR(255),
184     FOREIGN KEY (id_prop) REFERENCES Propietari(id_driver),
185     PRIMARY KEY (matricula)
186 );
187
188
189 --Creem la taula Charge
190
191 DROP TABLE IF EXISTS Charge CASCADE;
192 CREATE TABLE Charge (
193
194     charge VARCHAR(255),
195     description VARCHAR(255),
196     PRIMARY KEY (charge)
197 );
198
199
200 --Creem la taula Infraccio
201
202 DROP TABLE IF EXISTS Infraccio;
203 CREATE TABLE Infraccio (
204
205     id_Infraccio serial,
206     charge VARCHAR(255),
207     id_vehicle VARCHAR(255),
208     subAgency VARCHAR(255),
209     id_driver bigint,
210     id_state int,
211     violation_type VARCHAR(255),
212     ContributedToAccident boolean,
213     Belts boolean,
214     PersonalInjury boolean,
215     PropertyDamage boolean,
216     Fatal boolean,
217     HAZMAT boolean,
218     Alcohol boolean,
219     Work_Zone boolean,
220     Date_OfStop VARCHAR(255),
221     Time_OfStop time,
222     Location VARCHAR(255),
223     Latitude VARCHAR(255),
224     Longitude VARCHAR(255),
225     FOREIGN KEY (id_vehicle) REFERENCES Vehicle(matricula),
226     FOREIGN KEY (subAgency) REFERENCES District(sub_agency),
227     FOREIGN KEY (id_driver) REFERENCES Conductor(id_driver),
228     FOREIGN KEY (id_state) REFERENCES State(id_state),
229     PRIMARY KEY (id_Infraccio)
230 );
231

```

A aquestes imatges es pot observar la creació del nostre model físic amb SQL.


```

232
233 --Inserir la informació a la taula Comissaria
234 INSERT INTO Comissaria (agency) SELECT DISTINCT agency FROM TaulaITraffic;
235 SELECT * FROM Comissaria;
236
237 --Inserir la informació a la taula District
238 INSERT INTO District (sub_agency, agency)
239 SELECT DISTINCT ON (sub_agency) TaulaITraffic.sub_agency, Comissaria.agency FROM TaulaITraffic, Comissaria
240 WHERE TaulaITraffic.agency = Comissaria.agency;
241 SELECT * FROM District;
242
243 --Inserir la informació a la taula State
244 INSERT INTO State (state_name) SELECT DISTINCT TaulaITraffic.State FROM TaulaITraffic --CAL FER SUBQUERY??
245 WHERE TaulaITraffic.state IS NOT NULL;
246 INSERT INTO State (state_name) SELECT DISTINCT driver_state FROM TaulaITraffic
247 WHERE NOT EXISTS (SELECT TaulaITraffic FROM State WHERE state.state_name = TaulaITraffic.driver_state) AND TaulaITraffic.driver_state IS NOT NULL;
248 INSERT INTO State (state_name) SELECT DISTINCT dl_state FROM TaulaITraffic
249 WHERE NOT EXISTS (SELECT TaulaITraffic FROM State WHERE state.state_name = TaulaITraffic.dl_state) AND TaulaITraffic.driver_state IS NOT NULL;
250 SELECT * FROM State;
251 --Inserir la informació a la taula City
252 INSERT INTO City(city_name, id_state)
253 SELECT DISTINCT ON (driver_city) TaulaITraffic.driver_city, State.id_state FROM TaulaITraffic, State
254 WHERE TaulaITraffic.driver_state = State.state_name AND TaulaITraffic.driver_city IS NOT NULL;
255 SELECT * FROM City;
256
257 --Inserir la informació a la taula Conductor
258 INSERT INTO Conductor (id_driver, race, gender, city)
259 SELECT DISTINCT ON (id_person) t.id_person, t.race, t.gender, c.id_city
260 FROM City AS c, TaulaITraffic AS t
261 WHERE t.driver_city = c.city_name;
262 SELECT * FROM Conductor;
263
264 --Inserir la informació a la taula TipusC
265 INSERT INTO TipusC (tipus)
266 SELECT DISTINCT ON (TaulaITraffic.driving_license) TaulaITraffic.driving_license
267 FROM TaulaITraffic ;
268 SELECT * FROM TipusC;
269
270
271 --Inserir la informació a la taula TipusC
272 INSERT INTO TipusC (tipus)
273 SELECT DISTINCT ON (TaulaITraffic.driving_license) TaulaITraffic.driving_license
274 FROM TaulaITraffic ;
275 SELECT * FROM TipusC;
276
277 --Inserir la informació a la taula Carnet
278 INSERT INTO Carnet (id_driver, id_state, tipus) SELECT DISTINCT ON (d.id_driver, t.driving_license)d.id_driver, s.id_state, t.driving_license
279 FROM Conductor AS d, TaulaITraffic AS t, State AS s, tipusc AS t1
280 WHERE d.id_driver = t.id_person AND s.state_name = t.dl_state AND t1.tipus = t.driving_license;
281 SELECT * FROM Carnet;
282
283 --Inserir la informació a la taula Propietari
284 INSERT INTO Propietari (id_driver, nom)
285 SELECT DISTINCT ON (d.id_driver) d.id_driver, t.nom
286 FROM TaulaITraffic AS t, conductor AS d
287 WHERE t.id_prop = d.id_driver ;
288 SELECT * FROM Propietari;
289
290 --Inserir la informació a la taula Vehicle
291 INSERT INTO Vehicle (matricula,id_prop, vehicle_type, year, make, model, color)
292 SELECT DISTINCT ON (t.license) t.license,p.id_driver, t.vehicle_type, t-year, t.make, t.model, t.color
293 FROM TaulaITraffic AS t, propietari AS p, TaulaITraffic AS tp
294 WHERE license IS NOT NULL AND p.id_driver = tp.id_prop AND tp.id_prop = t.id_person ;
295 SELECT * FROM Vehicle;
296
297 --Inserir la informació a la taula Charge
298 INSERT INTO Charge (charge, description) SELECT DISTINCT ON (charge) charge, description
299 FROM TaulaITraffic AS t;
300 SELECT * FROM Charge;
301
302 --Inserir la informació a la taula Infraccio
303 INSERT INTO Infraccio( charge, id_vehicle, subAgency, id_driver,id_state,violation_type, ContributedtoAccident, Belts, PersonalInjury, PropertyDamage, Fatal, HAZMAT, Alcohol, Work_Zone, Date_OfStop, Time_OfStop)
304 SELECT c.charge, v.matricula, d.sub_agency, co.id_driver,s.id_state,t.violation_type, t.Contributed, t.Belts, t.Personal_Injury, t.Property_Damage, t.Fatal, t.HAZMAT, t.Alcohol, t.Work_Zone, t.Date_Of_Stop, t.I
305 FROM State AS s, Charge AS c, vehicle AS v, District AS d, TaulaITraffic AS t, Conductor AS co
306 WHERE c.charge = t.charge AND v.matricula = t.license AND d.sub_agency = t.sub_agency AND co.id_driver = t.id_person AND s.state_name = t.state;
307 SELECT * FROM Infraccio;
308

```

A aquestes imatges hi ha tot el codi que hem fet servir per inserir la informació de la taula d'importació a totes les diferents taules creades amb el model físic i hem lincat totes les taules mitjançant fk i condicions de lincatge.

Charset i Collate

Com s'han solucionat les incoherències de dades?

En el nostre cas, vam tenir un problema amb el charset ja que inicialment teníem seleccionat el que ve per defecte i llavors ens donava problemes amb el tipus date ja que el format estatunidenc es mes/dia/any i a nosaltres ens donaven dia/mes/any. Això ho vam solucionar al canviar el charset al espanyol. En el cas del colate, no hem tingut cap problema ja que aquest es selecciona automàticament en funció del charset seleccionat.

En el cas de les incoherències de dades, les hem tractat de forma diferent en funció de quina dada es tracti. En el cas de la taula persona, on tenim clar que només existeix una persona fem que encara que hi hagi dos gèneros diferents, només es quedi amb un dels dos. En canvi amb el cas de la taula carnet, on una persona pot tenir n carnets, fem que agafi tots els tipus de carnet que hi hagi a la taula d'importació, encara que la dada sigui errònia, com per exemple en el cas del tipus de carnet que hi ha un tipus que es. Això ho fem perquè es vegi a la informació que hi ha un tipus de carnet d'aquesta persona que no ha estat correctament introduït.

Test del procés d'importació

Incloure imatges descriptives i codi

```
1      --Creem la taula d'importacio dels propietaris
2      DROP TABLE IF EXISTS TaulaIProp;
3      CREATE TABLE TaulaIProp (
4
5          id_prop bigint,
6          matricula CHAR(13),
7          nom VARCHAR(255)
8      );
9      --Inserim la informació a la taula
10     COPY TaulaIProp FROM 'C:\Program Files\BDD\propietaris.csv' DELIMITER ',';
11     SELECT * FROM TaulaIProp;
12
```

Creem la taula d'importació on hi ha tots els propietaris i els importem del fitxer propietaris.csv.

```
--
13
14 -- Creem la taula d'importacio de Infraccions de trafic
15 DROP TABLE IF EXISTS TaulaITraffic;
16 CREATE TABLE TaulaITraffic (
17     date_of_stop date,
18     time_of_stop time,
19     agency CHAR(4),
20     sub_agency VARCHAR (255),
21     description VARCHAR (255),
22     location VARCHAR (255),
23     latitude real,
24     longitude real,
25     accident boolean,
26     belts boolean,
27     personal_injury boolean,
28     property_damage boolean,
29     fatal boolean,
30     hazmat boolean,
31     alcohol boolean,
32     work_zone boolean,
33     state VARCHAR(255),
34     license VARCHAR(255),
35     vehicle_type VARCHAR(255),
36     driving_license CHAR(3),
37     year int,
38     make VARCHAR(255),
39     model VARCHAR(255),
40     color VARCHAR(255),
41     violation_type VARCHAR(255),
42     charge VARCHAR(255),
43     contributed boolean,
44     race VARCHAR(255),
45     gender CHAR(3),
46     driver_city VARCHAR(255),
47     driver_state CHAR(5),
48     dl_state CHAR(5),
49     arrest_type VARCHAR(255),
50     id_person bigint
51 );
52
```

Creem la taula que servirà per importar tota la informació

```

43     contributed boolean,
44     race VARCHAR(255),
45     gender CHAR(3),
46     driver_city VARCHAR(255),
47     driver_state CHAR(5),
48     dl_state CHAR(5),
49     arrest_type VARCHAR(255),
50     id_person bigint
51 );
52
53     --Inserim la informació a la taula
54
55 COPY TaulaITraffic FROM 'C:\Program Files\BDD\traffic_violations_1.csv' DELIMITER ';' CSV HEADER;
56 COPY TaulaITraffic FROM '/Applications/BDD/traffic_violations_2.csv' DELIMITER ';' CSV HEADER;
57 COPY TaulaITraffic FROM '/Applications/BDD/traffic_violations_3.csv' DELIMITER ';' CSV HEADER;
58 COPY TaulaITraffic FROM '/Applications/BDD/traffic_violations_4.csv' DELIMITER ';' CSV HEADER;
59 COPY TaulaITraffic FROM '/Applications/BDD/traffic_violations_5.csv' DELIMITER ';' CSV HEADER;
60 COPY TaulaITraffic FROM '/Applications/BDD/traffic_violations_6.csv' DELIMITER ';' CSV HEADER;
61 COPY TaulaITraffic FROM '/Applications/BDD/traffic_violations_7.csv' DELIMITER ';' CSV HEADER;
62 COPY TaulaITraffic FROM '/Applications/BDD/traffic_violations_8.csv' DELIMITER ';' CSV HEADER;
63 COPY TaulaITraffic FROM '/Applications/BDD/traffic_violations_9.csv' DELIMITER ';' CSV HEADER;
64 COPY TaulaITraffic FROM '/Applications/BDD/traffic_violations_10.csv' DELIMITER ';' CSV HEADER;
65 COPY TaulaITraffic FROM '/Applications/BDD/traffic_violations_11.csv' DELIMITER ';' CSV HEADER;
66 COPY TaulaITraffic FROM '/Applications/BDD/traffic_violations_12.csv' DELIMITER ';' CSV HEADER;
67 COPY TaulaITraffic FROM '/Applications/BDD/traffic_violations_13.csv' DELIMITER ';' CSV HEADER;
68 COPY TaulaITraffic FROM '/Applications/BDD/traffic_violations_14.csv' DELIMITER ';' CSV HEADER;
69 COPY TaulaITraffic FROM '/Applications/BDD/traffic_violations_15.csv' DELIMITER ';' CSV HEADER;
70 COPY TaulaITraffic FROM '/Applications/BDD/traffic_violations_16.csv' DELIMITER ';' CSV HEADER;
71 COPY TaulaITraffic FROM '/Applications/BDD/traffic_violations_17.csv' DELIMITER ';' CSV HEADER;
72 COPY TaulaITraffic FROM '/Applications/BDD/traffic_violations_18.csv' DELIMITER ';' CSV HEADER;
73 COPY TaulaITraffic FROM '/Applications/BDD/traffic_violations_19.csv' DELIMITER ';' CSV HEADER;
74 COPY TaulaITraffic FROM '/Applications/BDD/traffic_violations_20.csv' DELIMITER ';' CSV HEADER;
75 COPY TaulaITraffic FROM '/Applications/BDD/traffic_violations_21.csv' DELIMITER ';' CSV HEADER;
76 COPY TaulaITraffic FROM '/Applications/BDD/traffic_violations_22.csv' DELIMITER ';' CSV HEADER;
77 COPY TaulaITraffic FROM '/Applications/BDD/traffic_violations_23.csv' DELIMITER ';' CSV HEADER;
78 select * from TaulaITraffic;
79
80

```

Finalment, importem la informació de tots als fitxers a la taula d'importació

Conclusions

Alhora de realitzar la fase 1, tot hi que hem hagut de anar enrederint la realització de la pràctica a causa de que havien varies conceptes que encara no havien sigut explicats a classe, hem aconseguit anar aprenent sense gaire dificultat com realitzar cada part de la pràctica.

Ens ha sigut de molta utilitat haver acudit i fet tots els laboratoris, ja que cada sessió la podíem extrapol·lar a diferents parts de la pràctica, alhora que ens servien d'exemple a seguir en el cas de que tinguéssim algun problema.

On hem tingut més problemes, va ser alhora de començar la pràctica, ja que no aconseguíem relacionar totes les diferents entitats fins que vam veure que era necessari crear una entitat Infracció que les relacionés totes. Un cop vam tenir el model conceptual ja vam poder avançar al model lògic i físic.

El segon problema més important a sigut quan ens va caldre ajuntar el codi d'importació amb el model físic, ja que no havíem realitzat abans mai res similar. En aquest moment va ser quan vam adonar-nos de les moltes incoherències que tenia el nostre model físic alhora de voler importar les diferents informacions dels fitxers, per tant vam tenir que reestructurar una part important tan del codi, com del model conceptual i lògic, per corregir els errors que ens frenaven d'aconseguir el correcte funcionament de la pràctica.

Posteriorment realitzant el estudi dels diferents sistemes gestors de bases de dades, ens hem adonat de les moltes solucions que hi ha per un mateix propòsit, i que és molt important saber quins són els punts forts i els punts dèbils de cada sistema per saber quin és el idoni per cada situació.