Databases: First Semester Project

***Enterprise Management System 2020-2021***

**List of members (name and email):** Lluís Gumbau (lluis.gumbaui@students.salle.url.edu), Narcís Cisquella (narcis.cisquella@students.salle.url.edu), Joan Llobet (joan.llobet@students.salle.url.edu) i Marc Postils Bragado (marc.postils@students.salle.url.edu)

**Date of finalisation:** 19/03/2021

### Summary of tasks

|  |  |  |
| --- | --- | --- |
| **Task** | **Subtask** | **Status**  (completed, in progress, not started, 3/5 completed…) |
| Entity-relationship model | Design | completed |
| Documentation | completed |
| Relational model | Design | completed |
| Normalisation | completed |
| Documentation | completed |
| Physical model | Implementation | completed |
| Documentation | completed |
| Database population | Import Sales CSV | completed |
| Import Product CSVs | completed |
| Import Shipping CSV | completed |
| Import HR | completed |
| Scripts for populating the tables that have not been populated before. | completed |
| Documentation | completed |
| Database validation | Validation Sales queries | completed |
|  | Validation Product queries | completed |
|  | Validation Shipping queries | completed |
|  | Validation HR queries | completed |
|  | Validation cross queries | completed |
|  | Documentation | completed |
| Conclusions | Use of resources | completed |
|  | Lessons learned | completed |
|  | Future work and conclusions | completed |

**Index**

1. [INTRODUCTION (1 PAGE) 3](#_bookmark0)
2. [ENTITY-RELATIONSHIP MODEL 4](#_bookmark1)
   1. [FULL ENTITY-RELATIONSHIP MODEL 4](#_bookmark2)
   2. [SALES MANAGEMENT MODULE 5](#_bookmark3)
   3. [PRODUCT MANUFACTURING MODULE 6](#_bookmark4)
   4. [SHIPPING LOGISTICS MODULE 7](#_bookmark5)
   5. [HUMAN RESOURCES MODULE 8](#_bookmark6)
3. [RELATIONAL MODEL 10](#_bookmark7)
   1. [FULL RELATIONAL MODEL 10](#_bookmark8)
   2. [SALES MANAGEMENT MODULE 11](#_bookmark9)
   3. [PRODUCT MANUFACTURING MODULE 12](#_bookmark10)
   4. [SHIPPING LOGISTICS MODULE 13](#_bookmark11)
   5. [HUMAN RESOURCES MODULE 14](#_bookmark12)
4. [PHYSICAL MODEL 16](#_bookmark13)
   1. [SALES MANAGEMENT MODULE (1-2 PAGES) 16](#_bookmark14)
   2. [PRODUCT MANUFACTURING MODULE (1-2 PAGES) 21](#_bookmark15)
   3. [SHIPPING LOGISTICS MODULE (1-2 PAGES) 24](#_bookmark16)
   4. [HUMAN RESOURCES MODULE (1-2 PAGES) 27](#_bookmark17)
5. [DATABASE POPULATION 30](#_bookmark18)
   1. [SALES MANAGEMENT MODULE (1-2 PAGES) 30](#_bookmark19)
   2. [PRODUCT MANUFACTURING MODULE (1-2 PAGES) 30](#_bookmark20)
   3. [SHIPPING LOGISTICS MODULE (1-2 PAGES) 31](#_bookmark21)
   4. [HUMAN RESOURCES MODULE (1-2 PAGES) 31](#_bookmark22)
6. [DATABASE VALIDATION 33](#_bookmark23)
   1. [SALES MANAGEMENT QUERIES 33](#_bookmark24)
   2. [PRODUCT MANUFACTURING QUERIES 36](#_bookmark25)
   3. [SHIPPING LOGISTICS MODULE 41](#_bookmark26)
   4. [HUMAN RESOURCES MODULE 45](#_bookmark27)
   5. [CROSS-PROCESS ANALYSES 49](#_bookmark28)
7. [CONCLUSIONS 60](#_bookmark29)
   1. [USE OF RESOURCES 60](#_bookmark30)
   2. [LESSONS LEARNT (1 PAGE) 60](#_bookmark31)
   3. [FUTURE WORK AND CONCLUSIONS (1 PAGE) 61](#_bookmark32)

# Introduction (1 page)

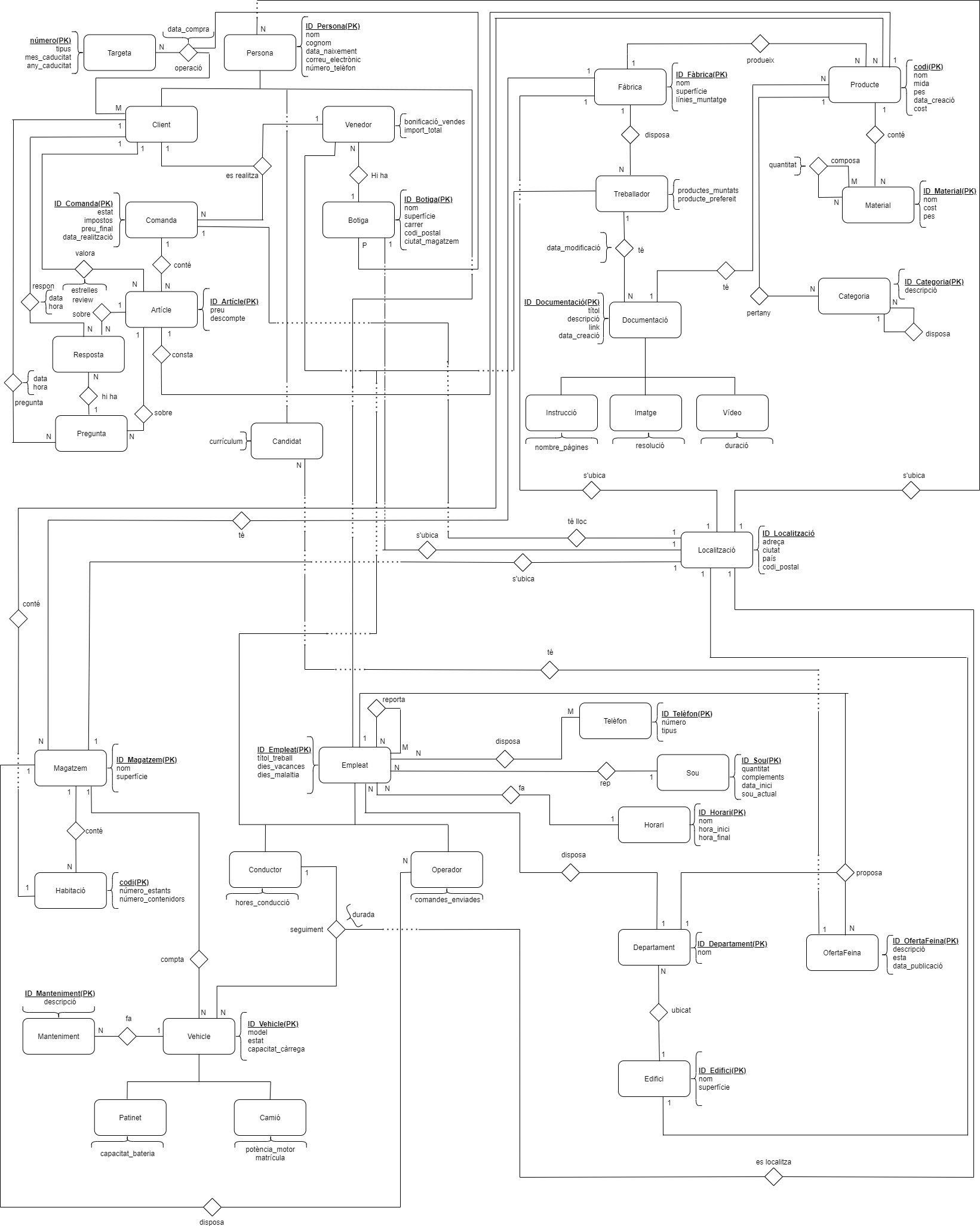
El principals objectius de la pràctica són els següents: dissenyar i implementar la base de dades, estructurar la bases de dades, i finalment demostrar que la base de dades està ben implementada i funciona correctament realitzant les quèries proposades.

Bàsicament, el context d’aquesta pràctica consisteix en la implementació interna de tots els factors que poden formar part d’una empresa. Primer de tot, tenim en compte les persones i les botigues, és a dir, el vincle entre el client i l’empresa. Després, tenint en compte les fàbriques i la producció dels diferents productes. Després, tenint en compte tots els empleats i transports per realitzar l’entrega de les comandes. I finalment, organitzar l’horari, sou, departament, etc. dels empleats.

El que vam fer per organitzar-nos amb el projecte va ser treballar conjuntament, tot i així hi va haver parts que si ens vam dividir la feina. Per als models conceptuals, cadascú dels membres del grup va fer un model conceptual diferent. Després, per passar del model conceptual al model relacional, si vam fer-ho tots junts i discutint com es posaven les taules correctament, decidint les PF i FK i, al mateix temps, anàvem decidint el tipus de dades dels atributs de les taules. Pel model físic, el Lluís i el Narcís van dedicar-hi més hores en dissenyar-lo i implementar-lo i en solucionar els problemes que anaven sorgint. Per altra banda, el Joan i el Marc es van centrar més en la realització de les quèries.

# Entity-relationship model

## Full entity-relationship model

* + 1. ***Diagram***

(Fent zoom a la imatge, es poden veure nítidament el nom de les entitats i els seus atributs)

* + 1. ***Justification (1-2 pages)***

L’organització de tot el model conceptual és “simple”. La part de dalt a l’esquerra pertany al primer mòdul (Sales management), la part de dalt a la dreta pertany al segon mòdul (Product manufacturing management), la part d’abaix a l’esquerra pertany al tercer mòdul (Shipping logistics) i la part d’abaix a la dreta pertany al últim mòdul (Human resources). Recalcar que l’entitat Candidat en la imatge es veu que està al primer mòdul però realment pertany al últim mòdul, l’hem posat al primer perquè no es creuin tantes línies. L’entitat Localització és clau també perquè el relacionem amb els quatre mòduls i l’hem posat per allà al mig. L’entitat Persona està ubicada al primer mòdul però els fills de Persona són Client (1r mòdul), Empleat (diversos mòduls) i Candidat (4t mòdul). Després, l’entitat Empleat té de fills: Conductor (3r mòdul), Operador (3r mòdul), Treballador (2n mòdul) i Venedor (1r mòdul). La interconnexió entre els quatre mòduls s’explicarà a continuació quan expliquem les relacions existent de cada mòdul i també la relació amb els altres mòduls.

## Sales management module

* + 1. ***Diagram***

Aquest diagrama es pot observar a la part de dalt a l’esquerra de la imatge del model conceptual mostrada anteriorment.

* + 1. ***Justification (1-2 pages)***

Primer de tot, hem creat la entitat Botiga i Venedor amb els seus atributs corresponents. Hem relacionat Botiga amb Venedor ja que ens diuen que a les botigues hi treballen venedors. A cada botiga (1) hi treballen diversos venedors (N), i un venedor sols pot treballar a una botiga. Després vam crear Client i Targeta amb els atributs corresponents. Després vam fer una ternària entre Targeta, Botiga i Client ja que ens diuen que els clients (N) poden comprar a diferents botigues (M), i poden utilitzar targetes de crèdits (P), que seran compartides entre els membres de la família. I l’atribut de la ternària hem posat data\_compra, que es la data en que el client va realitzar la compra a la botiga. Creem la entitat Comanda i fem una ternària entre Client, Comanda i Venedor. Perquè un client (1) pot fer diferents comandes (N) però només pot ser atès per un venedor (1). Considerem que un venedor atén a un sol client. Creem l’entitat Article i la relacionem amb Comanda i és 1:N ja que ens diuen que una comanda (1) conté una llista de articles (N). Relacionem Client amb Article i és 1:N ja que un client (1) pot escriure reviews i puntuar més d’un article (N), però aquesta review i puntuació del article sols pot estar feta per un client. I d’atributs d’aquesta relació hem posat les estrelles ja que la valoració que fa el client de l’article es guarda amb el número d’estrelles i també hem posat la review, que es en format text. Creem les entitats Pregunta i Resposta i cada una per separat la relacionem amb Client ja que cada client (1) pot redactar diferents preguntes (N) i cada client (1) pot

redactar diverses respostes (N). I aquella pregunta o resposta que ha fet el client, esta feta sols per ell. I en aquestes dues relacions li posem d’atributs data i hora perquè ens volem guardar la data i l’hora de quan aquell client ha realitzat aquella pregunta o resposta. Aleshores, també vam haver de relacionar Pregunta i Resposta amb Article ja que aquestes preguntes (N) i respostes (N) són sobre un article (1) en concret. També relacionem Pregunta amb Resposta ja que una pregunta (1) pot tenir més d’una resposta (N) i una resposta sols pertany a una pregunta. Hem decidit crear una entitat anomenada Localització ja que a moltes entitats futures d’altres mòduls teníem atributs repetits i per normalitzar-ho, crear la entitat Localització ha estat la millor opció. Aquesta entitat anirà relacionada amb moltes entitats dels mòduls següents. Aleshores relacionem Botiga amb Localització, i té una relació 1:1 ja que cada botiga té una única localització. Comanda també ho relacionem amb Localització i es 1:1 també perquè cada comanda té lloc a una localització en concret.

## Product manufacturing module

* + 1. ***Diagram***

Aquest diagrama es pot observar a la part de dalt a la dreta de la imatge del model conceptual mostrada anteriorment.

* + 1. ***Justification (1-2 pages)***

Primer vam crear les entitats Producte i Fàbrica amb els seus atributs corresponents. La seva relació és 1:N ja que en una fàbrica (1) es produeixen molts productes (N), i aquests productes en concret sols es poden haver produït en una fàbrica en concret. Hem creat la taula Materials i la hem relacionat amb Producte, la seva relació es 1:N ja que cada producte (1) està format per una sèrie de materials (N), però també en un inici vam considerar que la seva relació era N:M. Després vam fer una relació reflexiva N:M de Material, ja que els materials (N) estan composats d’altres materials (M), i aquests altres materials pertanyen a molts altres materials, i l’atribut d’aquesta relació hem posat quantitat volem saber la quantitat de cada material que està composat per un material. Hem creat l’entitat Categoria i l’hem relacionat amb Producte i la relació és 1:N ja que un producte (1) pertany a diferents categories (N). En un inici vam considerar que aquesta relació fos N:M. També hem fet una relació reflexiva a Categoria de relació 1:N perquè cada categoria pot tenir subcategories, però les subcategories també son categories i d’aquí la relació reflexiva. Hem creat l’entitat Documentació i la hem relacionat amb Producte, és 1:N ja que cada producte té una documentació en particular (1) i una documentació pot ser la mateixa per diferents productes (N), com productes del mateix tipus creats al mateix dia per exemple. Creem la taula Treballador, que són els treballadors de la fàbrica. Treballador ho relacionem amb Fàbrica i la relació és 1:N, ja que en una fàbrica (1) hi treballen

diversos treballadors (N) i cada treballador sols pot treballar en una fàbrica. Treballador també ho relacionem amb Documentació i és 1:N perquè cada treballador (1) modifica diversos documents (N) i cada document es modificat per un treballador en concret, hem posat un atribut en aquesta relació que és data\_modificació ja que volem saber quan el treballador ha modificat aquest document. De l’entitat Documentació hem fet una generalització ja que n’hi ha de tres tipus diferents que són Instrucció, Imatge i Vídeo, cadascuna amb els seus atributs corresponents. També relacionem Producte amb Article, que aquest últim pertany al primer mòdul i la relació és 1:1 ja que cada producte

(1) consta d’un article en concret (1), i aquest article fa referència a un producte. Cal relacionar també Fàbrica amb Localització i és 1:1 ja que cada fàbrica (1) s’ubica a una única localització en concret (1).

## Shipping logistics module

* + 1. ***Diagram***

Aquest diagrama es pot observar a la part de baix a l’esquerra de la imatge del model conceptual mostrada anteriorment.

* + 1. ***Justification (1-2 pages)***

Abans de tot, hem creat l’entitat Magatzem i Habitació i les relacionem amb una relació 1:N ja que ens diuen que cada magatzem (1) està composat per diverses habitacions (N) i aquestes sols poden pertànyer a un sol magatzem. En aquesta relació li hem posat l’atribut ID\_Producte ja que les habitacions contenen el producte i aquest només es pot ubicar en una habitació. Creem l’entitat Vehicle i la relacionem amb Magatzem ja que un magatzem (1) compta de diferents vehicles (N) i cada un d’aquests vehicles pertany a un magatzem. Aleshores creem les entitats Conductor i Operador i arribem a una conclusió, que molts atributs de diverses entitats estaven repetits i s’havia de normalitzar. Aleshores, vam crear la superentitat Empleat que està composta per l’Operador, definit al tercer mòdul, el Conductor, definit al tercer mòdul, el Venedor que hem definit anteriorment al primer mòdul i Treballador (de la fàbrica) que també la hem definit anteriorment al segon mòdul. Relacionem Conductor amb Vehicle (i amb una altre entitat que ara comentarem) ja que ens diuen que els vehicles són conduits pels conductors. Aleshores fem una ternària entre Vehicle, Conductor i Localització i la relació es 1:1:N ja que els conductors condueixen vehicles i volem guardar la adreça especificada pel client. Creem l’entitat Manteniment i la relacionem amb Vehicle ja que ens diuen que un vehicle (1) és mantingut diverses vegades a la vida (N). Fem una generalització de Vehicle que es divideix en Patinet i Camió amb els seus respectius atributs. El Magatzem el relacionem amb la taula Localització creada anteriorment, ja que ens demanen els atributs que té Localització pel Magatzem i la relació es 1:1 ja que un magatzem en concret (1) sols pot estar ubicat en una única localització (1). També hem relacionat Magatzem amb Fàbrica, que aquesta última pertany al segon

mòdul i la relació és 1:N ja que cada fàbrica (1) està composta per diversos magatzems (N) i cada un dels magatzems sols pertany a una fàbrica en concret. També relacionat Operador amb Magatzem ja que els operadors són operadors del magatzem i la seva relació és 1:N ja que en un magatzem (1) hi treballen diversos operadors (N) i un operador sols treballa en un magatzem en particular. També comentar que hem relacionat Habitació amb Producte ja que ens diuen que les habitacions contenen els productes (N) i que un producte sols pot estar ubicat en una habitació (1).

## Human resources module

* + 1. ***Diagram***

Aquest diagrama es pot observar a la part de baix a la dreta de la imatge del model conceptual mostrada anteriorment.

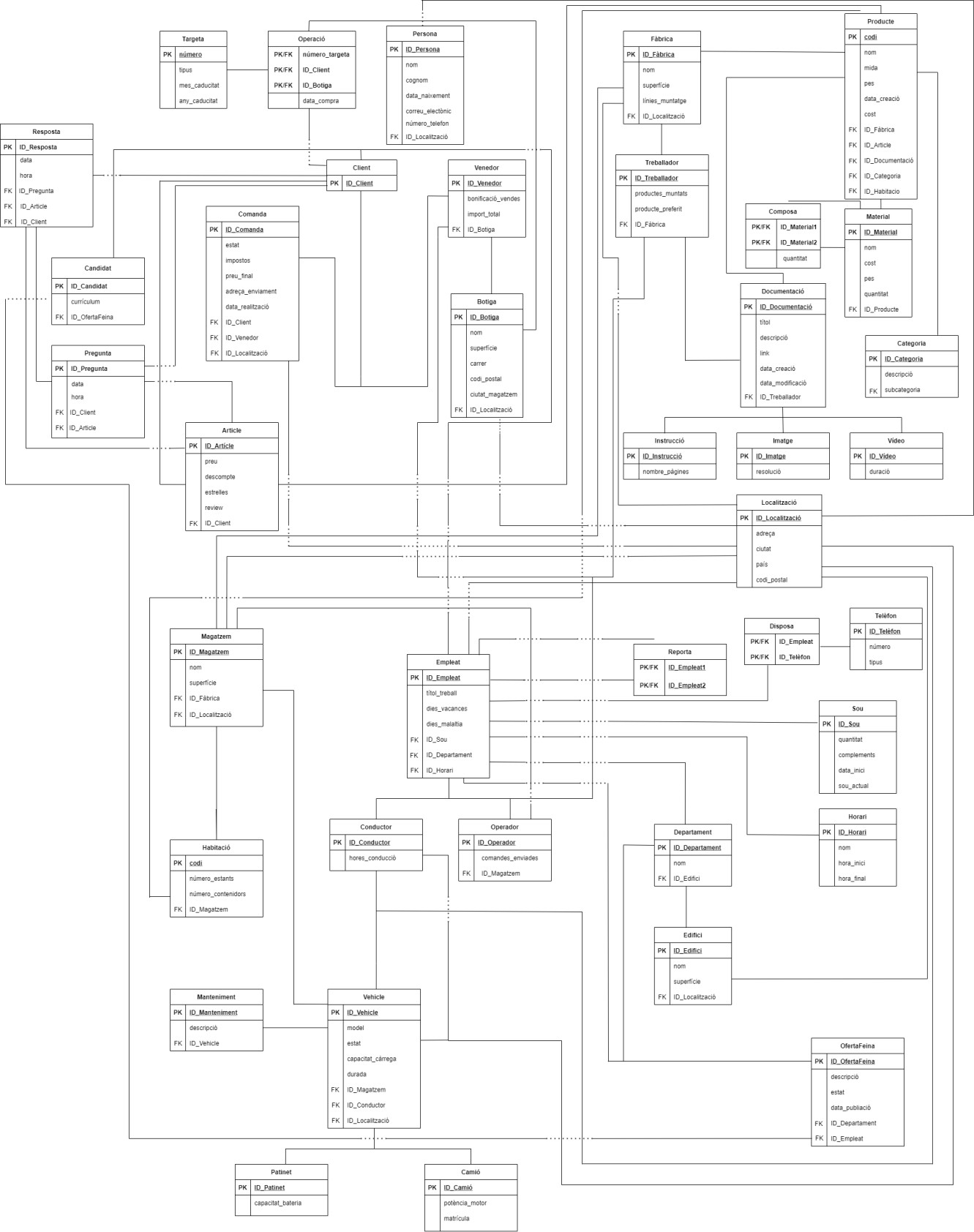
* + 1. ***Justification (1-2 pages)***

Primer vam crear la entitat Candidat. Aleshores, ens vam donar compte que havien atributs repetits entre Candidat, Client i Empleat i havíem de normalitzar-ho. Ho vam fer creant una superclasse anomenada Persona. I lo que vam fer és el següent: Persona té de fills: Client, Empleat i Candidat. I Empleat té de fills: Venedor, Treballador, Conductor i Operador. Aleshores vam relacionar Persona amb Localització ja que de cada persona necessitem saber els atributs de Localització i la relació és 1:N ja que cada persona pertany a una localització (1) en particular, però d’una localització hi poden haver-hi varies persones (N), per exemple membres de la mateixa família. Després vam crear una relació reflexiva a Empleat ja que ens diuen que un empleat realitza un informe sobre un altre empleat (ja sigui el cap o qualsevol altre). Lo més correcte seria que la relació fos la relació fos 1:N ja que un empleat reporta al seu cap (1) i el cap pot ser reportat per diversos empleats seus (N), però finalment vam posar N:M ja que vam considerar que un empleat podia reportar a altres persones responsables a part del cap. Per l’empleat necessitem guardar moltes coses. Primer de tot creem la entitat Horari i la relacionem amb Empleat i la relació és 1:N ja que un empleat sols pot tenir un horari (1) però un horari el poden complir/tenir diversos empleats (N). També volem saber el sou del empleat, creem la entitat Sou i la relacionem amb Empleat. La seva relació també és 1:N ja que cada empleat rep un sou en concret (1), però el mateix sou el poden cobrar diferents empleats. Després, creem l’entitat Telèfon i la relacionem amb Empleat en una relació de N:M, ja que ens diuen que els empleats poden tenir diversos mòbils (N) i que son compartits amb la resta d’empleats (M). Creem l’entitat Departament i la relacionem amb Empleat amb una relació de 1:N ja que ens diuen que cada empleat pertany únicament a un departament en concret (1), però en un departament hi ha treballa més d’un empleat (N). També creem una nova entitat anomenada Edifici i està relacionada amb Departament perquè ens diuen que els departaments (N) estan ubicats en un edifici (1), és a dir, a cada edifici hi ha diversos departaments. Relacionem també Edifici amb Localització amb una relació de 1:1 ja que de

cada edifici hem de saber la seva localització. Finalment, creem una última entitat anomenada OfertaFeina i creem una ternària entre OfertaFeina, Departament i Empleat de relació 1:1:N, ja que ens diuen que les ofertes de feina (N) són proposades per un departament (1) i per un empleat (1) que pertany a aquell departament en concret. Finalment, relacionem OfertaFeina amb Candidat ja que ens diuen que per una oferta de feina (1), poden haver-hi diversos candidats (N).

# Relational model

## Full relational model

* + 1. ***Diagram***

(Fent zoom a la imatge, es poden veure nítidament el nom de les entitats i els seus atributs)

* + 1. ***Justification (1 page)***

L’estructura del model relacional es la mateixa que la del conceptual. Pràcticament totes les taules estan ubicades al mateix lloc que en el model conceptual. La part de dalt a l’esquerra pertany al primer mòdul (Sales management), la part de dalt a la dreta pertany al segon mòdul (Product manufacturing management), la part d’abaix a l’esquerra pertany al tercer mòdul (Shipping logistics) i la part d’abaix a la dreta pertany al últim mòdul (Human reosurces). Cal matisar que la taula Candidat realment pertany al 4t mòdul però per un tema d’estètica i d’evitar incorporar creuar mes línies en el model, l’hem posat en el 1r mòdul. Les connexions entre tots els mòduls ja estan explicades al model conceptual, però més endavant ho tornem a explicar més per sobre per no repetir-se tota l’estona.

## Sales management module

* + 1. ***Diagram***

Aquest diagrama es pot observar a la part de dalt a l’esquerra de la imatge del model relacional mostrada anteriorment.

* + 1. ***Justification (1 page)***

Explicarem com hem passat del model conceptual al relacional posant exemples reals d’aquest mòdul.

Quan s’ha donat el cas que la relació entre dues entitats fos 1:1 com per exemple Botiga i Localització, el que s’ha de fer és crear una taula per cada entitat i posar-li els seus atributs corresponents. De PK li hem posat el seu identificador (ID\_Botiga i ID\_Localitzacio) que és únic, i no pot ser nul. Aleshores, per marcar la relació entre aquestes dues taules, hem de triar una de les dues taules que més ens convingui i més sentit tingui per posar-li de FK la PK de l’altra taula. Per exemple, a nosaltres de cada botiga ens interessava tenir com a FK el PK de la taula Localització, i no al inrevés perquè volem saber la localització de la botiga.

Quan s’ha donat el cas que la relació entre dues entitats fos 1:N com per exemple Botiga i Venedor, el que es fa es crear una taula per cada entitat, en aquest cas una taula Botiga i una altre Venedor amb els seus atributs corresponents i de PK el seu identificador. Un cop fet això, la taula que tingui la multiplicitat N, en aquest cas, Venedor, doncs absorbeix com a FK la PK de Botiga. Per tant, la taula Venedor té com a FK ID\_Botiga.

Quan s’ha donat el cas que la relació ha estat 1:N i que aquesta relació tingui un atribut, com per exemple podria ser la relació entre Client i Article. El que es fa en aquest cas és seguir el procediment que es fa quan hi ha una relació 1:N però afegint l’atribut de la relació a la taula que té multiplicitat

N. Per exemple, els atributs de la relació entre Client i Article són estrelles i review, i com l’entitat

que te multiplicitat N és Article, doncs a més dels atributs que ja té article, li afegim el de estrelles i review.

Quan s’ha donat el cas d’una ternària amb relació 1:1:N, com pot ser el cas de la relació entre Client, Venedor i Comanda s’ha de fer el següent. La taula que té multiplicitat N, en aquest cas Comanda, doncs absorbeix com a FK les PK de les altres dues taules amb multiplicitat 1. És a dir, Comanda tindrà de FK ID\_Client (PK de la taula Client) i ID\_Venedor (PK de la taula Venedor).

Quan s’ha donat el cas d’una ternària amb relació N:M:P, com pot ser la relació entre Targeta, Client i Botiga, el que s’ha de fer és el següent. Hem de crear una taula intermitja amb el nom del verb de la relació (pots posar un altre) i aquesta taula té com a PK/FK les PK de les tres taules. Per tant, el verb de la relació és operació i doncs la taula es dirà Operació i tindrà de PK/FK el número\_targeta (PK de la taula Targeta), ID\_Client (PK de la taula Client) i ID\_Botiga (PK de la taula Botiga). I com aquesta ternària té un atribut, doncs aquest atribut se li posa a la taula Operació.

Les connexions amb els altres mòduls són les mateixes que en el model conceptual, allà està explicat més concreta i extensament.

Aquestes són tots els tipus de relacions existents d’aquest primer mòdul.

## Product manufacturing module

* + 1. ***Diagram***

Aquest diagrama es pot observar a la part de dalt a la dreta de la imatge del model relacional mostrada anteriorment.

* + 1. ***Justification (1 page)***

Explicarem com hem passat del model conceptual al relacional posant exemples reals d’aquest mòdul.

Quan s’ha donat el cas que la relació entre dues entitats fos 1:1 com per exemple Fàbrica i Localització, el que s’ha de fer és crear una taula per cada entitat i posar-li els seus atributs corresponents. De PK li hem posat el seu identificador (ID\_Fàbrica i ID\_Localitzacio) que és únic, i no pot ser nul. Aleshores, per marcar la relació entre aquestes dues taules, hem de triar una de les dues taules que més ens convingui i més sentit tingui per posar-li de FK la PK de l’altra taula. Per exemple, a nosaltres de cada fàbrica ens interessava tenir com a FK el PK de la taula Localització, i no al inrevés perquè volem saber la localització de la fàbrica.

Quan s’ha donat el cas que la relació entre dues entitats fos 1:N com per exemple Fàbrica i Treballador, el que es fa es crear una taula per cada entitat, en aquest cas una taula Fàbrica i una altre Treballador amb els seus atributs corresponents i de PK el seu identificador. Un cop això, la taula que

tingui la multiplicitat N, en aquest cas, Treballador, doncs absorbeix com a FK la PK de Fàbrica. Per tant, la taula Treballador té com a FK ID\_Fàbrica.

Quan s’ha donat el cas que la relació ha estat 1:N i que aquesta relació tingui un atribut, com per exemple podria ser la relació entre Treballador i Documentació. El que es fa en aquest cas és seguir el procediment que es fa quan hi ha una relació 1:N però afegint l’atribut de la relació a la taula que té multiplicitat N. Per exemple, l’atribut de la relació entre Treballador i Documentació és data\_modificació, i com l’entitat que te multiplicitat N és Documentació, doncs a més dels atributs que ja té documentació, li afegim el de data\_modificació.

Quan s’ha donat el cas que hi ha hagut una relació reflexiva de relació 1:N, com per exemple en la entitat Categoria, s’ha de fer lo següent. Has de crear una taula amb el nom de la entitat, en aquest cas Categoria, amb el seu PK i atributs corresponents. I la PK es duplica com a FK però canviant-li el nom, en el nostre cas, subcategoria, ja que ens diuen que una categoria disposa de subcategories.

Quan s’ha donat el cas que hi ha hagut una relació reflexiva de relació N:M, com es el cas de Material, s’ha de fer lo següent. Hem de crear dues taules, una anomenada Material i l’altre amb el nom del verb, en el nostre cas Composa. La taula Material té de PK i atributs els que li correspongui però la taula Composa tindrà de PK/FK la PK de Material però amb noms diferents. En el nostre cas de PK/FK li hem posat ID\_Material1 i ID\_Material2. I com a la relació té un atribut anomenat quantitat, doncs aquest atribut el posem a la taula Composa. I finalment, les dues taules les hem de relacionar dues vegades (dues línies).

Quan s’ha donat el cas que tenim una generalització, com és Documentació, que té de fills Instrucció, Imatge i Vídeo, s’ha de fer el següent. En el model conceptual, les taules Instrucció, Imatge i Vídeo no tenen PK, però en el relacional si se l’hi ha de posar.

Les connexions amb els altres mòduls són les mateixes que en el model conceptual, allà està explicat més concreta i extensament.

Aquestes són tots els tipus de relacions existents d’aquest segon mòdul.

## Shipping logistics module

* + 1. ***Diagram***

Aquest diagrama es pot observar a la part de baix a l’esquerra de la imatge del model relacional mostrada anteriorment.

* + 1. ***Justification (1 page)***

Explicarem com hem passat del model conceptual al relacional posant exemples reals d’aquest mòdul.

Quan s’ha donat el cas que la relació entre dues entitats fos 1:1 com per exemple Magatzem i Localització, el que s’ha de fer és crear una taula per cada entitat i posar-li els seus atributs corresponents. De PK li hem posat el seu identificador (ID\_Magatzem i ID\_Localitzacio) que és únic, i no pot ser nul. Aleshores, per marcar la relació entre aquestes dues taules, hem de triar una de les dues taules que més ens convingui i més sentit tingui per posar-li de FK la PK de l’altra taula. Per exemple, a nosaltres de cada magatzem ens interessava tenir com a FK el PK de la taula Localització, i no al inrevés perquè volem saber la localització del magatzem.

Quan s’ha donat el cas que la relació entre dues entitats fos 1:N com per exemple Vehicle i Manteniment, el que es fa es crear una taula per cada entitat, en aquest cas una taula Vehicle i una altre Manteniment amb els seus atributs corresponents i de PK el seu identificador. Un cop això, la taula que tingui la multiplicitat N, en aquest cas, Manteniment, doncs absorbeix com a FK la PK de Vehicle. Per tant, la taula Manteniment té com a FK ID\_Vehicle.

Quan s’ha donat el cas que tenim una generalització, com pot ser Vehicle, que té de fills Patinet, i Camió, s’ha de fer el següent. En el model conceptual, les taules Patinet i Camió no tenen PK, però en el relacional si se l’hi ha de posar.

Quan s’ha donat el cas d’una ternària amb relació 1:1:N, com pot ser el cas de la relació entre Conductor, Localització i Vehicle s’ha de fer el següent. La taula que té multiplicitat N, en aquest cas Vehicle, doncs absorbeix com a FK les PK de les altres dues taules amb multiplicitat 1. És a dir, Vehicle tindrà de FK ID\_Conductor (PK de la taula Conductor) i ID\_Localització (PK de la taula Localització). L’atribut de la ternària, que en aquest cas es diu durada, es posa en la taula que té multiplicitat N, en aquest cas, Vehicle.

Les connexions amb els altres mòduls són les mateixes que en el model conceptual, allà està explicat més concreta i extensament.

Aquestes són tots els tipus de relacions existents d’aquest tercer mòdul.

## Human resources module

* + 1. ***Diagram***

Aquest diagrama es pot observar a la part de baix a la dreta de la imatge del model relacional mostrada anteriorment.

* + 1. ***Justification (1 page)***

Explicarem com hem passat del model conceptual al relacional posant exemples reals d’aquest mòdul.

Quan s’ha donat el cas que la relació entre dues entitats fos 1:1 com per exemple Edifici i Localització, el que s’ha de fer és crear una taula per cada entitat i posar-li els seus atributs corresponents. De PK li hem posat el seu identificador (ID\_Edifici i ID\_Localització) que és únic, i no pot ser nul. Aleshores, per marcar la relació entre aquestes dues taules, hem de triar una de les dues taules que més ens convingui i més sentit tingui per posar-li de FK la PK de l’altra taula. Per exemple, a nosaltres de cada edifici ens interessava tenir com a FK el PK de la taula Localització, i no al inrevés perquè volem saber la localització de cada edifici.

Quan s’ha donat el cas que la relació entre dues entitats fos 1:N com per exemple Edifici i Departament, el que es fa es crear una taula per cada entitat, en aquest cas una taula Edifici i una altre Departament amb els seus atributs corresponents i de PK el seu identificador. Un cop això, la taula que tingui la multiplicitat N, en aquest cas, Departament, doncs absorbeix com a FK la PK de Edifici. Per tant, la taula Departament té com a FK ID\_Edifici.

Quan s’ha donat el cas que la relació entre dues entitats fos N:M com per exemple Empleat i Telèfon, el que s’ha de fer es crear una taula intermitja amb el nom del verb de la relació (generalment) i que aquesta taula tingui de PK/FK les PK de les dues entitats. En aquest cas, creem la taula intermitja anomenada Disposa amb PK/FK ID\_Empleat (PK de la taula Empleat) i ID\_Telèfon (PK de la taula Telèfon).

Quan s’ha donat el cas d’una ternària amb relació 1:1:N, com pot ser el cas de la relació entre Empleat, Departament i OfertaFeina s’ha de fer el següent. La taula que té multiplicitat N, en aquest cas OfertaFeina, doncs absorbeix com a FK les PK de les altres dues taules amb multiplicitat 1. És a dir, OfertaFeina tindrà de FK ID\_Empleat (PK de la taula Empleat) i ID\_Departament (PK de la taula Departament).

Quan s’ha donat el cas que hi ha hagut una relació reflexiva de relació N:M, com es el cas de Empleat, s’ha de fer lo següent. Hem de crear dues taules, una anomenada Empleat i l’altre amb el nom del verb, en el nostre cas Reporta. La taula Empleat té de PK i atributs els que li correspongui però la taula Reporta tindrà de PK/FK la PK de Empleat però amb noms diferents. En el nostre cas de PK/FK li hem posat ID\_Empleat1 i ID\_Empleat2. I finalment, les dues taules les hem de relacionar dues vegades (dues línies).

Les connexions amb els altres mòduls són les mateixes que en el model conceptual, allà està explicat més concreta i extensament.

Aquestes són tots els tipus de relacions existents d’aquest quart mòdul.

# Physical model

## Sales management module (1-2 pages)

* + 1. ***Data type selection***

**Taula Localitzacio:** ID\_Localitzacio SERIAL, adreça VARCHAR(255), ciutat VARCHAR(255), pais VARCHAR(255),

codi\_postal VARCHAR(255),

**Taula Localitzacio\_extra:** adreça VARCHAR(255), ciutat VARCHAR(255), pais VARCHAR(255),

codi\_postal VARCHAR(255),

**Taula Data\_local:** data\_naixament DATE, ID\_Localització SERIAL,

**Taula OfertaFeina\_rand:** ID\_Persona SERIAL, data\_naixament DATE, numero\_telefon VARCHAR(255), ID\_Localització SERIAL, ID\_OfertaFeina SERIAL,

**Taula OfertaFeina\_persona:** ID\_Persona INTEGER, nom VARCHAR(255),

cognom VARCHAR(255), data\_naixament DATE, correu\_electronic VARCHAR(255), numero\_telefon VARCHAR(255), ID\_Localització SERIAL,

**Taula Persona\_aux:** ID\_Persona SERIAL, nom VARCHAR(255),

cognom VARCHAR(255), data\_naixament DATE, correu\_electronic VARCHAR(255), numero\_telefon VARCHAR(255), ID\_Localització SERIAL,

**Taula Persona:** ID\_Persona SERIAL, nom VARCHAR(255),

cognom VARCHAR(255), data\_naixement DATE, correu\_electronic VARCHAR(255), numero\_telefon VARCHAR(255), ID\_Localitzacio INTEGER,

### Taula Persona2:

nom VARCHAR(255),

cognom VARCHAR(255), data\_naixement DATE, correu\_electronic VARCHAR(255), numero\_telefon VARCHAR(255), ID\_Localitzacio INTEGER,

### Taula Client:

ID\_Client INTEGER,

**Taula Botiga:** ID\_Botiga SERIAL, nom VARCHAR(255),

superficie FLOAT, ID\_Localitzacio INTEGER,

### Taula Venedor:

ID\_Venedor INTEGER, bonificacio\_vendes INTEGER, import\_total FLOAT4, ID\_Botiga INTEGER,

### Taula Targeta:

ID\_targeta SERIAL, numero VARCHAR(255), tipus VARCHAR(255),

mes\_caducitat INTEGER, any\_caducitat INTEGER,

**Taula Comanda\_aux:** ID\_Comanda INTEGER, estat VARCHAR(255),

impostos FLOAT4, preu\_final FLOAT4, data\_realitzacio DATE, ID\_Client SERIAL, ID\_Venedor INTEGER, ID\_Localitzacio SERIAL,

**Taula Comanda:** ID\_Comanda INTEGER, estat VARCHAR(255),

impostos FLOAT4, preu\_final FLOAT4, data\_realitzacio DATE, ID\_Client SERIAL, ID\_Venedor INTEGER, ID\_Localitzacio SERIAL,

### Taula Operació:

numero\_targeta VARCHAR(255), ID\_Client INTEGER,

ID\_Botiga INTEGER, data\_compra DATE,

**Taula Article\_aux:** ID\_Article SERIAL, preu FLOAT4, descompte INTEGER,

**Taula Article:** ID\_Article SERIAL, preu FLOAT4, descompte INTEGER, ID\_Client INTEGER,

ID\_Comanda INTEGER, review TEXT,

estrelles VARCHAR(255), ID\_ArticleClient SERIAL,

**Taula Pregunta:** ID\_Pregunta SERIAL, data DATE,

hora TIME, ID\_Client INTEGER,

ID\_Article INTEGER,

**Taula Resposta\_aux:** ID\_Resposta SERIAL, data DATE,

hora TIME, ID\_Client INTEGER,

ID\_Pregunta INTEGER

**Taula Resposta:** ID\_Resposta SERIAL, data DATE,

hora TIME, ID\_Client INTEGER,

ID\_Article INTEGER,

ID\_Pregunta INTEGER,

* + 1. ***Creation script description***

Per fer el model físic vam utilitzar el programa PgAdmin, on primer de tot vam crear una Base de Dades on poder emmagatzemar tot el procés a mesura que avançàvem, un cop aquesta creada, vam crear un nou Script per poder crear les taules i poder treballar amb elles i així amb totes les parts del model físic.

## Product manufacturing module (1-2 pages)

* + 1. ***Data type selection***

**Taula Fàbrica:** ID\_Fabrica SERIAL, nom VARCHAR(255),

superficie FLOAT4, linies\_muntatge INTEGER, ID\_Localitzacio INTEGER,

**Taula Treballador:** ID\_Treballador SERIAL, productes\_muntats INTEGER, ID\_producte\_preferit INTEGER, ID\_Fabrica INTEGER,

**Taula Documentació\_aux:** ID\_Documentacio INTEGER, titol VARCHAR(255),

descripcio TEXT, link TEXT, data\_creacio DATE,

data\_modificacio DATE, ID\_Treballador INTEGER

**Taula Documentació2:** ID\_Documentacio INTEGER, data\_modificacio DATE, ID\_Treballador INTEGER

**Taula Documentació:** ID\_Documentacio SERIAL, titol VARCHAR(255),

descripcio TEXT, link TEXT, data\_creacio DATE,

data\_modificacio DATE, ID\_Treballador INTEGER,

**Taula Categoria1:** ID\_Categoria SERIAL, nom VARCHAR(255),

**Taula Categoria2:** ID\_Categoria2 SERIAL, nom2 VARCHAR(255),

ID\_Categoria INTEGER,

**Taula Categoria3:** ID\_Categoria3 SERIAL, nom3 VARCHAR(255),

ID\_Categoria2 INTEGER,

### Taula Producte:

codi SERIAL,

nom VARCHAR(255),

mida FLOAT4, pes FLOAT4,

data\_creacio DATE, cost FLOAT4, ID\_Fabrica INTEGER, ID\_Article INTEGER,

ID\_Categoria INTEGER, ID\_Documentacio INTEGER,

**Taula Material:** ID\_Material SERIAL, nom VARCHAR(255),

cost FLOAT4, pes FLOAT4,

quantitat FLOAT4, ID\_Producte INTEGER,

**Taula Composa:** ID\_Material1 INTEGER, ID\_Material2 INTEGER, quantitat FLOAT4,

**Taula Instrucció:** ID\_Instruccio INTEGER, nombre\_pagines INTEGER,

### Taula Imatge:

ID\_Imatge INTEGER, resolucio VARCHAR(255),

**Taula Video:** ID\_Video INTEGER, duracio INTEGER,

* + 1. ***Creation script description***

Per fer el model físic vam utilitzar el programa PgAdmin, on primer de tot vam crear una Base de Dades on poder emmagatzemar tot el procés a mesura que avançàvem, un cop aquesta creada, vam crear un nou Script per poder crear les taules i poder treballar amb elles i així amb totes les parts del model físic.

## Shipping logistics module (1-2 pages)

* + 1. ***Data type selection***

**Taula Magatzem:** ID\_Magatzem SERIAL, nom VARCHAR(255),

superficice FLOAT4, ID\_Fabrica INTEGER, ID\_Localitzacio INTEGER,

### Taula Habitació:

codi SERIAL, numero\_estants INTEGER,

numero\_contenidors INTEGER, ID\_Producte INTEGER, ID\_Magatzem INTEGER,

**Taula Conductor:** ID\_Conductor INTEGER, hores\_conduccio FLOAT4,

**Taula Operador\_aux:** ID\_Operador INTEGER, ID\_Magatzem INTEGER

### Taula Operador:

ID\_Operador INTEGER, comandes\_enviades INTEGER, ID\_Magatzem INTEGER,

**Taula Vehicle\_aux:** ID\_Vehicle SERIAL, ID\_Magatzem INTEGER, ID\_Localitzacio INTEGER

**Taula Vehicle2:** ID\_Vehicle SERIAL, model VARCHAR(255), estat VARCHAR(255),

capacitat\_carrega FLOAT4, ID\_Magatzem INTEGER, ID\_Conductor INTEGER, ID\_Localitzacio INTEGER

**Taula Vehicle:** ID\_Vehicle SERIAL, model VARCHAR(255), estat VARCHAR(255),

capacitat\_carrega FLOAT4, ID\_Magatzem INTEGER, ID\_Conductor INTEGER,

ID\_Localitzacio INTEGER,

**Taula Seguiment:** ID\_Conductor INTEGER, ID\_Vehicle INTEGER, ID\_Localitzacio INTEGER, durada INTEGER,

### Taula Patinet:

ID\_Patinet INTEGER, capacitat\_bateria INTEGER,

### Taula Camió:

ID\_Camio INTEGER, matricula VARCHAR(255), potencia\_motor INTEGER,

**Taula Manteniment:** ID\_Manteniment INTEGER, ID\_Vehicle INTEGER, descripcio TEXT, any\_manteniment INTEGER,

* + 1. ***Creation script description***

Per fer el model físic vam utilitzar el programa PgAdmin, on primer de tot vam crear una Base de Dades on poder emmagatzemar tot el procés a mesura que avançàvem, un cop aquesta creada, vam crear un nou Script per poder crear les taules i poder treballar amb elles i així amb totes les parts del model físic.

## Human resources module (1-2 pages)

* + 1. ***Data type selection***

### Taula Sou:

ID\_Sou SERIAL,

quantitat FLOAT4, complements FLOAT4, data\_inici DATE, sou\_actual BOOLEAN,

**Taula Horari:** ID\_Horari SERIAL, nom VARCHAR(255),

hora\_inici TIME, hora\_final TIME,

### Taula Edifci:

ID\_Edifici SERIAL, nom VARCHAR(255),

superficie FLOAT4, ID\_Localitzacio INTEGER,

**Taula Departament:** ID\_Departament SERIAL, nom VARCHAR(255),

ID\_Edifici SERIAL,

### Taula Empleat:

ID\_Empleat INTEGER, titol\_treball VARCHAR(255), dies\_vacances INTEGER,

dies\_malalties INTEGER, ID\_Sou INTEGER,

ID\_Departament INTEGER, ID\_Horari INTEGER,

**Taula OfertaFeina:** ID\_OfertaFeina SERIAL, departament VARCHAR(255), descripcio TEXT,

estat VARCHAR(255),

data\_publicacio DATE, ID\_Persona INTEGER,

**Taula Candidat:** ID\_Candidat INTEGER, curriculum TEXT, ID\_OfertaFeina INTEGER,

**Taula Reporta:** ID\_Empleat1 INTEGER, ID\_Empleat2 INTEGER,

**Taula Telefon\_aux:** ID\_Telefon INTEGER, tipus VARCHAR(255)

**Taula Telefon\_tv:** ID\_Telefon INTEGER, numero VARCHAR(255), tipus VARCHAR(255)

**Taula Telèfon2:** ID\_Telefon SERIAL, numero VARCHAR(255)

### Taula Telèfon:

ID\_Telefon SERIAL, numero VARCHAR(255), tipus VARCHAR(255),

**Taula Disposa:** ID\_Empleat INTEGER, ID\_Telefon INTEGER,

* + 1. ***Creation script description***

Per fer el model físic vam utilitzar el programa PgAdmin, on primer de tot vam crear una Base de Dades on poder emmagatzemar tot el procés a mesura que avançàvem, un cop aquesta creada, vam crear un nou Script per poder crear les taules i poder treballar amb elles i així amb totes les parts del model físic.

# Database population

## Sales management module (1-2 pages)

Per començar, vam crear taules buides, un cop creades, vam crear informació per cada una de les columnes de les taules més tard, amb el COPY, vam passar la informació dels CSVS a les taules de la nostra Base de Dades. Quan ja estaven totes les taules plenes de informació, vam crear les taules ja normalitzades corresponents als nostres models conceptual i relacional, i les vam omplir amb la informació de les taules que contenien tots els atributs dels CSVS, això ho vam fer amb l’INSERT INTO, on seleccionàvem la informació que volíem i la copiàvem, ja que per exemple en les taules d’importació podíem tenir 10 atributs i la taula destí només ens interessaven 4 dels 10 atributs d’origen.

Les taules que no ens vau donar ja fetes, les hem creat nosaltres amb un generador de dades aleatòries, en el nostre cas hem utilitzat el lloc web Mockaroo.

Un exemple del nostre model físic podria ser la taula Person\_Review, aquesta taula està plena d’informació d’uns dels CSVS, i amb un INSERT INTO, agafem les columnes que ens interessen i les importem a la nostra taula Persona.

## Product manufacturing module (1-2 pages)

Per començar, vam crear taules buides, un cop creades, vam crear informació per cada una de les columnes de les taules més tard, amb el COPY, vam passar la informació dels CSVS a les taules de la nostra Base de Dades. Quan ja estaven totes les taules plenes de informació, vam crear les taules ja normalitzades corresponents als nostres models conceptual i relacional, i les vam omplir amb la informació de les taules que contenien tots els atributs dels CSVS, això ho vam fer amb l’INSERT INTO, on seleccionàvem la informació que volíem i la copiàvem, ja que per exemple en les taules d’importació podíem tenir 10 atributs i la taula destí només ens interessaven 4 dels 10 atributs d’origen.

Les taules que no ens vau donar ja fetes, les hem creat nosaltres amb un generador de dades aleatòries, en el nostre cas hem utilitzat el lloc web Mockaroo.

Un exemple del nostre model físic és la taula Treballador\_aux, és una taula creada i omplerta per nosaltres, més tard creem la taula Fabrica(taula ja normalitzada del model relacional o conceptual) i la omplim amb la informació que ens interessa de Treballador\_aux, ho vam fer així per poder relacionar el productes cartesians de la taula Fàbrica amb les altres taules d’interès.

## Shipping logistics module (1-2 pages)

Per començar, vam crear taules buides, un cop creades, vam crear informació per cada una de les columnes de les taules més tard, amb el COPY, vam passar la informació dels CSVS a les taules de la nostra Base de Dades. Quan ja estaven totes les taules plenes de informació, vam crear les taules ja normalitzades corresponents als nostres models conceptual i relacional, i les vam omplir amb la informació de les taules que contenien tots els atributs dels CSVS, això ho vam fer amb l’INSERT INTO, on seleccionàvem la informació que volíem i la copiàvem, ja que per exemple en les taules d’importació podíem tenir 10 atributs i la taula destí només ens interessaven 4 dels 10 atributs d’origen.

Les taules que no ens vau donar ja fetes, les hem creat nosaltres amb un generador de dades aleatòries, en el nostre cas hem utilitzat el lloc web Mockaroo.

En aquest cas, un clar exemple és la taula Empleat\_aux, on primer omplim aquesta taula, després creem la taula Empelat ja normalitzada, i seleccionem i importem la informació que volem afegir amb el INSERT INTO. En aquest cas, com empleat és una persona, també havíem d’afegir els empleats de la taula Empleats a la taula persona, això ho vem fer amb un INSERT INTO, on agafàvem tota la informació que tenien en comú Persona i Empleat\_aux, i afegíem a les columnes a la taula Persona, al ID\_Persona ser un SERIAL, els empleats afegits a Persona s’afegien com una persona més i tenien el seu propi ID.

## Human resources module (1-2 pages)

Per començar, vam crear taules buides, un cop creades, vam crear informació per cada una de les columnes de les taules més tard, amb el COPY, vam passar la informació dels CSVS a les taules de la nostra Base de Dades. Quan ja estaven totes les taules plenes de informació, vam crear les taules ja normalitzades corresponents als nostres models conceptual i relacional, i les vam omplir amb la informació de les taules que contenien tots els atributs dels CSVS, això ho vam fer amb l’INSERT INTO, on seleccionàvem la informació que volíem i la copiàvem, ja que per exemple en les taules d’importació podíem tenir 10 atributs i la taula destí només ens interessaven 4 dels 10 atributs d’origen.

Les taules que no ens vau donar ja fetes, les hem creat nosaltres amb un generador de dades aleatòries, en el nostre cas hem utilitzat el lloc web Mockaroo.

En aquest cas, tenim la taula Telefon\_aux, on com igual que a casos anteriors, omplim la taula amb informació provinent d’un csv creat específicament per aquesta taula, això ho fem amb un “COPY”,

un cop aquesta plena, tenim la taula Telèfon, ja normalitzada, i l’omplim amb la informació de la taula Telefon\_aux.

En aquests últims 4 apartats, hem fet l’explicació d’una de moltes taules que tenim, la majoria de taules segueixen el mateix, o un molt semblant, procediment de creació i importació.

# Database validation

## Sales management queries

* + 1. ***Query 1***
       1. *Solution*

SELECT t.numero, t.tipus

FROM Targeta AS t, Botiga AS b, Operacio AS o, Venedor AS v, Persona AS p, Localitzacio AS l

WHERE o.numero\_targeta = t.numero

AND o.ID\_Botiga = b.ID\_Botiga

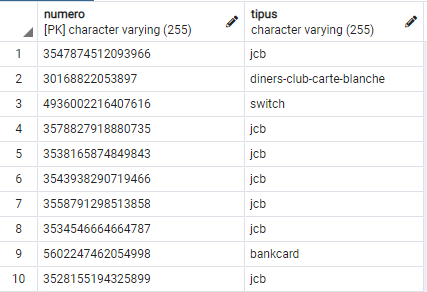
AND v.ID\_Botiga = b.ID\_Botiga

AND p.ID\_Persona = v.ID\_Venedor

AND p.ID\_Localitzacio = l.ID\_Localitzacio

AND l.pais = 'France'

GROUP BY t.numero, t.tipus HAVING COUNT(b.ID\_Botiga) > 3;



* + - 1. *Explanation*

En aquesta query mostrem el número i tipus de la targeta que s’han utilitzat més de tres vegades i que en una de les compres el venedor fos de nacionalitat francesa. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat les condicions necessàries. A més a més, hem fet un group by amb la condició que hagi utilitzat la targeta a més de tres botigues.

* + - 1. *Query validation*

En la primera query de validació mostrem el count i observem com la primera targeta té 5 botigues

Fem un insert a operació amb la mateixa targeta i a una altra botiga amb un dels seus venedors francès i observem el canvi.

SELECT t.numero, t.tipus,COUNT(b.ID\_Botiga)

FROM Targeta AS t, Botiga AS b, Operacio AS o, Venedor AS v, Persona AS p, Localitzacio AS l

WHERE o.numero\_targeta = t.numero

AND o.ID\_Botiga = b.ID\_Botiga

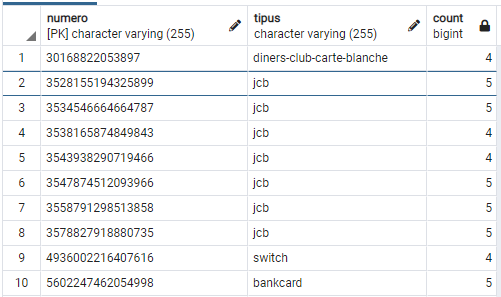
AND v.ID\_Botiga = b.ID\_Botiga

AND p.ID\_Persona = v.ID\_Venedor

AND p.ID\_Localitzacio = l.ID\_Localitzacio

AND l.pais = 'France'

GROUP BY t.numero, t.tipus HAVING COUNT(b.ID\_Botiga) > 3;



INSERT INTO Operacio(numero\_targeta, ID\_Client, ID\_Botiga, data\_compra)

VALUES (3547874512093966, 6, 1,'2019/02/08');

SELECT t.numero, t.tipus,COUNT(b.ID\_Botiga)

FROM Targeta AS t, Botiga AS b, Operacio AS o, Venedor AS v, Persona AS p, Localitzacio AS l

WHERE o.numero\_targeta = t.numero

AND o.ID\_Botiga = b.ID\_Botiga

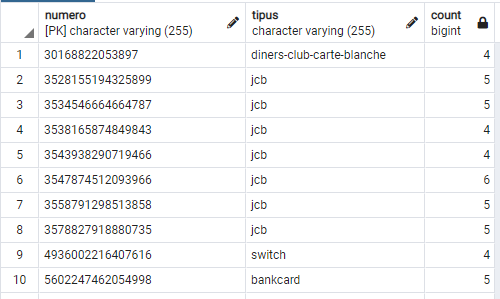
AND v.ID\_Botiga = b.ID\_Botiga

AND p.ID\_Persona = v.ID\_Venedor

AND p.ID\_Localitzacio = l.ID\_Localitzacio

AND l.pais = 'France'

GROUP BY t.numero, t.tipus HAVING COUNT(b.ID\_Botiga) > 3;



Després de fer l’insert podem veure com a la fila 6 s’ha afegit una botiga nova.

* + 1. ***Query 2***
       1. *Solution*
       2. *Explanation*

No l’hem sabut fer.

* + - 1. *Query validation*
    1. ***Query 3***
       1. *Solution*

SELECT l.pais

FROM Localitzacio AS l, Botiga AS b

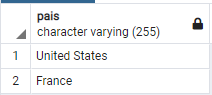
WHERE l.ID\_Localitzacio = b.ID\_Localitzacio

AND b.superficie > 1300

GROUP BY l.pais

ORDER BY COUNT(b.ID\_Botiga) DESC

LIMIT 2;



* + - 1. *Explanation*

En aquesta query mostrem el país que tenen més botigues amb una superfície superior a 1300. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat les condicions de que la superfície sigui major a 1300 i hem fet un group by per agrupar per països. Hem afegit un order by que ordena segons el nombre de botigues que té cada país, ho ordenem de manera descendent, així els primers que mostra son els països amb més botigues. El límit 2 l’hem posat perquè ens demanaven mostrar 2 països.

* + - 1. *Query validation*

Mostrem els mateixos atributs que a la query principal afegint el nombre de botigues que te cada país, així podem comprovar que el resultat de la query principal és correcte. En la validació, a diferència de la principal, mostrem tots els països amb el nombre de botigues (sense LIMIT), així podem veure tots el països i comprovar que realment mostrem els dos amb més botigues.

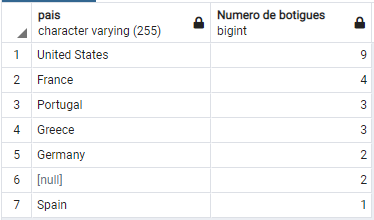
SELECT l.pais, COUNT(b.ID\_Botiga) AS "Numero de botigues"

FROM Localitzacio AS l, Botiga AS b

WHERE l.ID\_Localitzacio = b.ID\_Localitzacio

AND b.superficie > 1300

GROUP BY l.pais ORDER BY COUNT(b.ID\_Botiga) DESC;



A més a més tenim una segona query per poder comprovar el nombre de botigues, amb una superfície major a 1300, que hi ha en el país que li diem, per poder comprovar entre els 7 països que ens mostra la query anterior, on inicialment posa ‘United States’ canviem pels altres, i validem que el count funciona correctament i que realment seleccionem els països amb més botigues. També mostrem l’ID de les botigues per comprovar que no hi ha cap repetida, tanmateix afegim al select la superfície de la botiga per també validar que és superior a 1300.

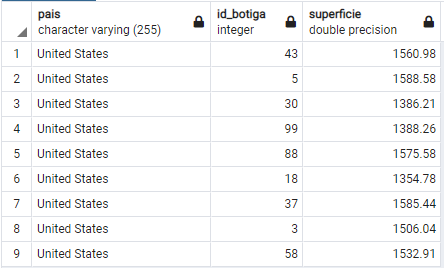
SELECT l.pais, b.ID\_Botiga, b.superficie

FROM Localitzacio AS l, Botiga AS b

WHERE l.ID\_Localitzacio = b.ID\_Localitzacio

AND l.pais = 'United States'

AND b.superficie > 1300;



* + 1. ***Query 4***
       1. *Solution*

SELECT p.cognom, p.nom, p.data\_naixement,

FROM Persona AS p, Client AS c, Article AS a

WHERE p.ID\_Persona = c.ID\_Client

AND a.ID\_Client = c.ID\_Client

AND a.review IS NOT NULL

GROUP BY p.cognom, p.nom, p.data\_naixement

ORDER BY COUNT(a.review) DESC

LIMIT 3;

* + - 1. *Explanation*

En aquesta query mostrem el cognom, el nom i la data de naixement dels tres clients que han fet més ressenyes (reviews). Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat la condició que els clients hagin escrit una ressenya. I hem fet el group by per a que no es repetís el mateix client, i l’order by l’hem posat ja que d’aquesta manera per que al primer ens mostri el client que més ressenyes ha fet. Finalment hem posat límit 3 ja que havíem de mostrar tres clients.

* + - 1. *Query validation*

En la primera query de validació mostrem el count i observem com la primera persona té 10 review la segona 2 i la tercera 1.

Fem un insert al client 100 per exemple perquè passi d'1 review a 2 i quan tornem a executar la query ens mostra com a canviat donant per bona la validació.

SELECT p.cognom, p.nom, p.data\_naixement, COUNT(a.review) AS "Numero de reviews",p.ID\_Persona

FROM Persona AS p, Client AS c, Article AS a

WHERE p.ID\_Persona = c.ID\_Client

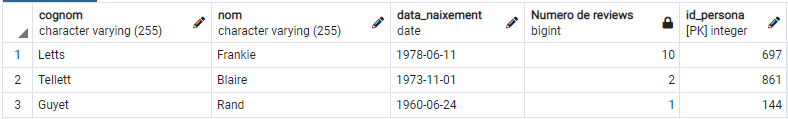
AND a.ID\_Client = c.ID\_Client

AND a.review IS NOT NULL

GROUP BY p.cognom, p.nom, p.data\_naixement, p.ID\_Persona, a.ID\_Client

ORDER BY COUNT(a.review) DESC

LIMIT 3;



INSERT INTO Article(ID\_Article, preu, descompte, ID\_Client, ID\_Comanda, review, estrelles, ID\_ArticleClient)

VALUES (26, 52.13, 12,100, null, 'Bon producte', 4, 647);

SELECT p.cognom, p.nom, p.data\_naixement, COUNT(a.review) AS "Numero de reviews",p.ID\_Persona

FROM Persona AS p, Client AS c, Article AS a

WHERE p.ID\_Persona = c.ID\_Client

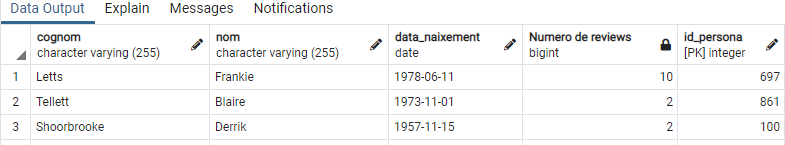
AND a.ID\_Client = c.ID\_Client

AND a.review IS NOT NULL

GROUP BY p.cognom, p.nom, p.data\_naixement, p.ID\_Persona, a.ID\_Client

ORDER BY COUNT(a.review) DESC

LIMIT 3;



A la query principal teniem 3 clients, amb 10, 2 i una ressenya, després de fer el insert, on afegim a un altre usuari una ressenya més entra dins del “top 3”. Així podem comprobar que realmente s’actualitza i mostrem els tres que realmente han fet més ressenyes.

* + 1. ***Query 5***
       1. *Solution*

SELECT c.ID\_Comanda, c.estat, c.preu\_final

FROM Comanda AS c, Localitzacio AS l, Persona AS p, Article AS a, Resposta AS r, Pregunta AS pr

WHERE a.ID\_Comanda = c.ID\_Comanda

AND pr.ID\_Article = a.ID\_Article

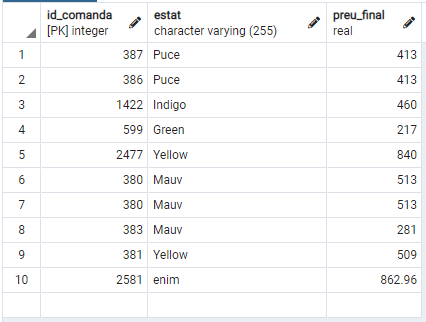
AND pr.ID\_Pregunta = r.ID\_Pregunta

AND r.data = '2020-03-14'

AND p.ID\_Localitzacio = l.ID\_Localitzacio

AND c.ID\_Client = p.ID\_Persona

AND l.pais = 'Italy';



* + - 1. *Explanation*

En aquesta query mostrem les comandes que ha fet un client italià que l’article té una pregunta i aquesta ha estat resposta al dia 14 d’abril del 2020. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat la condició de la data requerida i que el país origen del client fos Itàlia.

* + - 1. *Query validation*

Mostrem els mateixos atributs que a la query principal, el país del client i la data que es va publicar la resposta, així podem verificar que el client és Italià i que la data de la resposta és la que se'ns requereix, en aquest cas 14/03/2020. Finalment, veiem que els resultats coincideixen.

SELECT c.ID\_Comanda, c.estat, c.preu\_final, l.pais AS "País del client", r.data AS "Data publicacio resposta"

FROM Comanda AS c, Localitzacio AS l, Persona AS p, Article AS a, Resposta AS r, Pregunta AS pr

WHERE a.ID\_Comanda = c.ID\_Comanda

AND pr.ID\_Article = a.ID\_Article

AND pr.ID\_Pregunta = r.ID\_Pregunta

AND r.data = '2020-03-14'

AND p.ID\_Localitzacio = l.ID\_Localitzacio

AND c.ID\_Client = p.ID\_Persona

AND l.pais = 'Italy';



A més a més, us mostrem, amb la query a continuació, les comandes que han fet tots els Italians, sense tenir en compte la data, així comprovem que hi ha més dates i realment mostrem les que s’han fet el 14/03/2020, així que tot compila bé.

SELECT c.ID\_Comanda, c.estat, c.preu\_final, l.pais AS "País del client", r.data AS "Data publicacio resposta"

FROM Comanda AS c, Localitzacio AS l, Persona AS p, Article AS a, Resposta AS r, Pregunta AS pr

WHERE a.ID\_Comanda = c.ID\_Comanda

AND pr.ID\_Article = a.ID\_Article

AND pr.ID\_Pregunta = r.ID\_Pregunta

AND p.ID\_Localitzacio = l.ID\_Localitzacio

AND c.ID\_Client = p.ID\_Persona

AND l.pais = 'Italy';



## Product manufacturing queries

* + 1. ***Query 1***
       1. *Solution*

SELECT p.nom, p.correu\_electronic, t.productes\_muntats, p.nom, pr.cost

FROM Persona AS p, Treballador AS t, Producte AS pr, Fabrica AS f, Localitzacio AS l, Localitzacio AS l2

WHERE p.ID\_Persona = t.ID\_Treballador

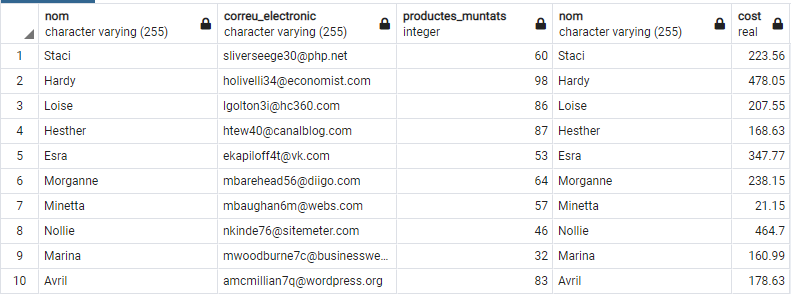
AND t.ID\_producte\_preferit = pr.codi

AND t.ID\_Fabrica = f.ID\_Fabrica

AND f.ID\_Localitzacio = l.ID\_Localitzacio

AND l2.ID\_Localitzacio = p.ID\_Localitzacio

AND l.ciutat = l2.ciutat;



* + - 1. *Explanation*

En aquesta query mostrem el nom, email, número de productes muntats, nom del seu producte preferit i el seu cost corresponent. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat la condició de que la fàbrica estigui a la mateixa ciutat ubicada que la ciutat de naixement del treballador.

* + - 1. *Query validation*

Mostrem els mateixos atributs que a la query principal, les ciutats on es troba la fàbrica i la ciutat on viu actualment el treballador. Com podem comprovar els resultats d'ambdues queries son els mateixos i les ciutats coindiceixen.

SELECT p.nom, p.correu\_electronic, t.productes\_muntats, p.nom, pr.cost, l.ciutat AS "Ciutat Fabrica", l2.ciutat AS "Ciutat Treballador"

FROM Persona AS p, Treballador AS t, Producte AS pr, Fabrica AS f, Localitzacio AS l, Localitzacio AS l2

WHERE p.ID\_Persona = t.ID\_Treballador

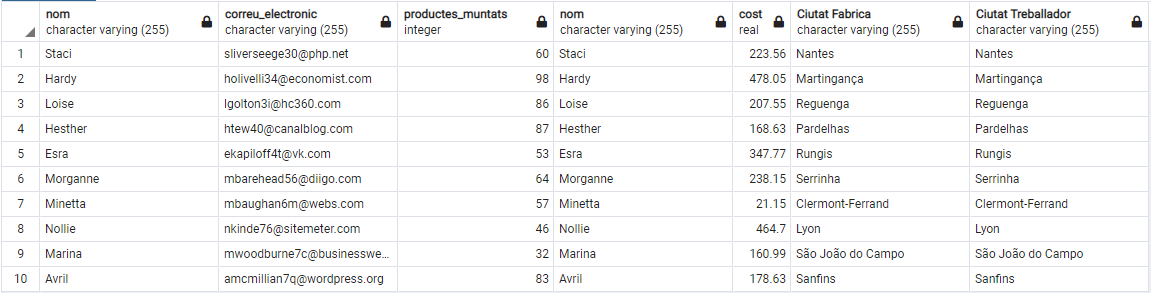
AND t.ID\_producte\_preferit = pr.codi

AND t.ID\_Fabrica = f.ID\_Fabrica

AND f.ID\_Localitzacio = l.ID\_Localitzacio

AND l2.ID\_Localitzacio = p.ID\_Localitzacio

AND l.ciutat = l2.ciutat;



* + 1. ***Query 2***
       1. *Solution*

SELECT p.nom, p.pes, p.mida

FROM Producte AS p, Localitzacio AS l, Material AS m, Fabrica AS f

WHERE f.ID\_Fabrica = p.ID\_Fabrica

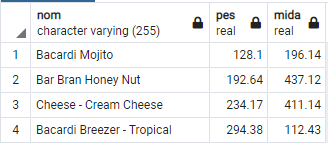
AND f.ID\_Localitzacio = l.ID\_Localitzacio

AND m.ID\_Producte = p.codi

AND l.pais = 'Spain'

GROUP BY p.nom, p.pes, p.mida

ORDER BY COUNT(m.ID\_Material) DESC

LIMIT 4;

* + - 1. *Explanation*

En aquesta query mostrem el nom, el pes i la mida dels productes que han estat fabricats a Espanya i que estan compostos pel major número de materials. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat la condició de que el producte hagi estat muntat a Espanya. Hem posat el group by per a que no es repetís cap producte. Hem posat el order by per a que ens mostri el productes que estan composats per més materials, ordenats de més a menys. Finalment, hem posat el límit 4 ja que ens ho requereix l’enunciat.

* + - 1. *Query validation*

Mostrem els mateixos atributs que a la query principal juntament amb un COUNT dels materials amb els que ha estat fet. A diferència de la query principal, aqui mostrem tots els productes fets a "Spain" així podem comprovar que realment els que mostrem son els 4 que estan fets amb més materials que els altres.

Inserts per afegir un altre material al nostre producte, així podem demostrar que realment mostrem els que més materials tenen. (Primer executar els inserts!)

INSERT INTO Material(ID\_Material, nom, cost, pes, quantitat, ID\_Producte)

VALUES(1010, 'Lemon', 3.04, 1.25, 21.08, 1000305);

INSERT INTO Material(ID\_Material, nom, cost, pes, quantitat, ID\_Producte)

VALUES(1011, 'Sugar', 2.56, 5.63, 188.25, 1000316);

INSERT INTO Material(ID\_Material, nom, cost, pes, quantitat, ID\_Producte)

VALUES(1012, 'Milk', 52.36, 6.35, 2.25, 1000597);

INSERT INTO Material(ID\_Material, nom, cost, pes, quantitat, ID\_Producte)

VALUES(1013, 'Ice', 50.56, 3.69, 3.58, 1000501);

SELECT p.nom, p.pes, p.mida, COUNT(m.ID\_Material)

FROM Producte AS p, Localitzacio AS l, Material AS m, Fabrica AS f

WHERE f.ID\_Fabrica = p.ID\_Fabrica

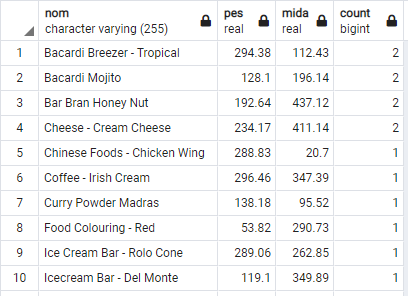
AND f.ID\_Localitzacio = l.ID\_Localitzacio

AND m.ID\_Producte = p.codi

AND l.pais = 'Spain'

GROUP BY p.nom, p.pes, p.mida

ORDER BY COUNT(m.ID\_Material) DESC;



Després de fer els inserts veiem que a la query principal mostrem les 4 que tenen més materials.

* + 1. ***Query 3***
       1. *Solution*

SELECT d.titol, d.data\_creacio

FROM Documentacio AS d, Treballador AS t, Producte AS p, Fabrica AS f

WHERE t.ID\_Fabrica = f.ID\_Fabrica

AND f.ID\_Fabrica = p.ID\_Fabrica

AND p.ID\_Documentacio = d.ID\_Documentacio

AND t.productes\_muntats > 1000

GROUP BY d.titol, d.data\_creacio;



* + - 1. *Explanation*

En aquesta query mostrem el títol i la data de creació dels documents i al quantitat de modificacions que han fet els treballadors de la fàbrica que han fabricat més de 1000 productes. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat la condició de que el els treballadors hagin muntat més de 1000 productes. Hem fet posat un group by per a que no es repeteixi cap document.

* + - 1. *Query validation*

Mostrem els mateixos atributs que a la query principal, però, a més a més mostrem els productes que ha muntat el treballador, d'aquesta manera demostrem que realment ha muntat mes de 1000 productes. Com podem comprovar els resultats d'ambdues queries son els mateixos.

L’insert el fem, perquè realment veieu que la informació que estem mostrant, és la correcte, en aquest cas afegim una nova modificació en la documentació anomenada “Smooth Brome”.

INSERT INTO Producte(codi, nom, mida, pes, data\_creacio, cost, ID\_Fabrica, ID\_Article, ID\_Categoria, ID\_Documentacio)

VALUES (1001014, 'Postils - Product', 8.23, 5.67, '1999-03-27', 2.34, 12, 1302 , 9, 82);

SELECT d.titol, d.data\_creacio, COUNT(d.data\_modificacio), t.productes\_muntats

FROM Documentacio AS d, Treballador AS t, Producte AS p, Fabrica AS f

WHERE t.ID\_Fabrica = f.ID\_Fabrica

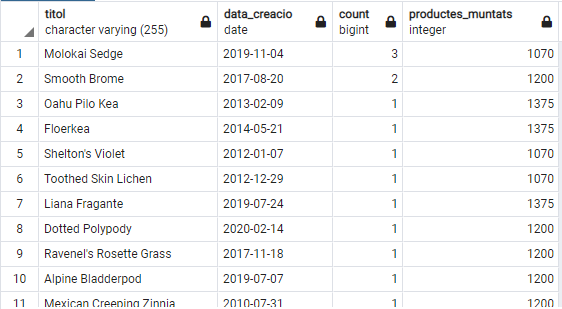
AND f.ID\_Fabrica = p.ID\_Fabrica

AND p.ID\_Documentacio = d.ID\_Documentacio

AND t.productes\_muntats > 1000

GROUP BY d.titol, d.data\_creacio, t.productes\_muntats

ORDER BY COUNT(d.data\_modificacio) DESC;



* + 1. ***Query 4***
       1. *Solution*

SELECT d.titol, d.link, d.data\_creacio

FROM Documentacio AS d, Video AS v, Material AS m, Categoria1 AS c1, Categoria2 AS c2, Producte AS p

WHERE d.ID\_Documentacio = v.ID\_Video

AND d.ID\_Documentacio = p.ID\_Documentacio

AND p.codi = m.ID\_Producte

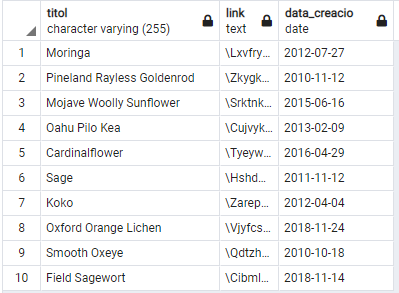
AND p.ID\_Categoria = c1.ID\_Categoria

AND c2.ID\_Categoria = c1.ID\_Categoria

AND v.duracio > 25

AND c2.nom2 = 'Printmaking'

AND m.nom LIKE 'Wood';



* + - 1. *Explanation*

En aquesta query mostrem el títol, el link i la data de de creació dels vídeos dels productes fets del material de fusta, que pertanyen a la subcategoria “Printmaking” i que els vídeos durin més de 25 minuts. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat la condició de que el vídeo duri més de 25 minuts i que la subcategoria es digui “Printmaking” i que el material es digui “Wood”, que és fusta.

* + - 1. *Query validation*

Mostrem els mateixos atributs que a la query principal, a més a més, mostrem que el material del producte realment és fusta, que té com a subcategoria "Printmaking" i que conté un video amb una duració superior a 25. Com podem comprovar els resultats d'ambdues queries son els mateixos. També, cal afegir, que vam haver d'afegir manualment amb INSERTs la informació necessària, perque hi haguessin productes amb aquestes dues categories. Aquests inserts els podem trobar en el model físic.

SELECT d.titol, d.link, d.data\_creacio, m.nom AS "Material del producte" ,c2.nom2 AS "Subcategoria", v.duracio AS "Duracio del video"

FROM Documentacio AS d, Video AS v, Material AS m, Categoria1 AS c1, Categoria2 AS c2, Producte AS p

WHERE d.ID\_Documentacio = v.ID\_Video

AND d.ID\_Documentacio = p.ID\_Documentacio

AND p.codi = m.ID\_Producte

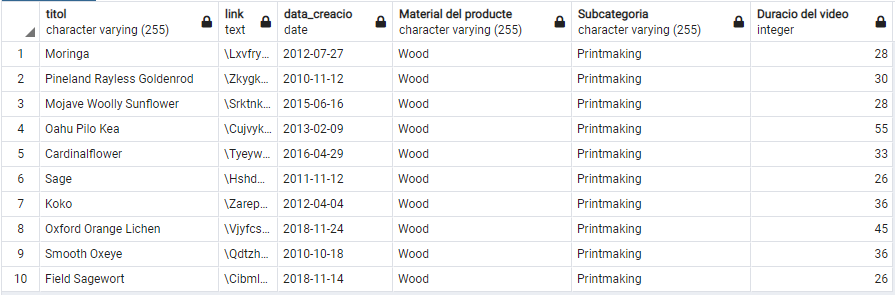
AND p.ID\_Categoria = c1.ID\_Categoria

AND c2.ID\_Categoria = c1.ID\_Categoria

AND v.duracio > 25

AND c2.nom2 = 'Printmaking'

AND m.nom LIKE 'Wood';



* + 1. ***Query 5***
       1. *Solution*

SELECT c3.nom3

FROM Producte AS p, Categoria1 AS c1, Categoria2 AS c2, Categoria3 AS c3

WHERE c3.ID\_Categoria2 = c2.ID\_Categoria2

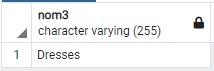
AND c2.ID\_Categoria = c1.ID\_Categoria

AND p.ID\_Categoria = c1.ID\_Categoria

AND p.cost > 111

AND c3.nom3 = 'Dresses'

GROUP BY c3.nom3;



* + - 1. *Explanation*

En aquesta query mostrem la subcategoria “Dresses”, que tenen més productes amb un preu superior a 111 euros. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat la condició de que el cost del producte sigui superior a 111 euros i que la subcategoria es digui “Dresses”. Hem posat un group by del nom de la categoria “Dresses” perquè no surti repetida.

* + - 1. *Query validation*

Mostrem els mateixos atributs que a la query principal, a més a més, mostrem el preu del producte, així podem comprovar que realment el preu és superior a 111, i les categories superiors a la de dresses. En aquest cas, no coincideixen resultats entre les dues queries, ja que en la principal ens demanen que ho agrupem per categories, i com podem observar a la subquery tots els productes que compleixen els requisits de la query principal, pertanyen a la categoria "Dresses".

SELECT c3.nom3, p.cost AS "Preu del producte", c1.nom, c2.nom2

FROM Producte AS p, Categoria1 AS c1, Categoria2 AS c2, Categoria3 AS c3

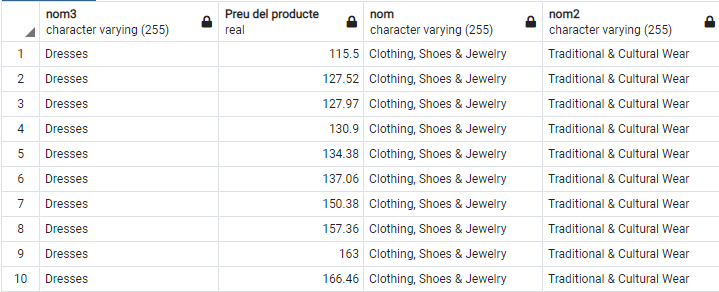
WHERE c3.ID\_Categoria2 = c2.ID\_Categoria2

AND c2.ID\_Categoria = c1.ID\_Categoria

AND p.ID\_Categoria = c1.ID\_Categoria

AND p.cost > 111

AND c3.nom3 = 'Dresses'

 GROUP BY c3.nom3, p.cost, c1.nom, c2.nom2;

A aquesta última query mostrem totes les categories sense codicions, així podem comprovar que hi ha moltes més categories i que el que mostrem està tot ben filtrat.

SELECT c3.nom3, p.cost AS "Preu del producte", c1.nom, c2.nom2

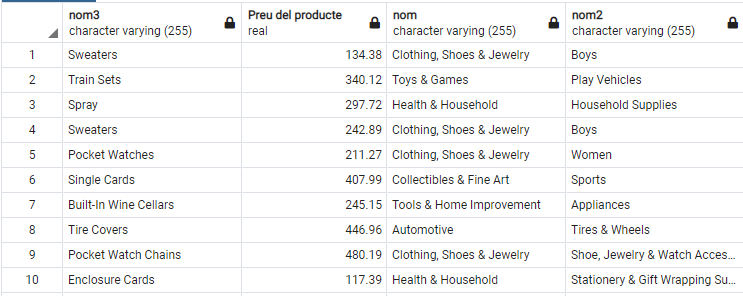
FROM Producte AS p, Categoria1 AS c1, Categoria2 AS c2, Categoria3 AS c3

WHERE c3.ID\_Categoria2 = c2.ID\_Categoria2

AND c2.ID\_Categoria = c1.ID\_Categoria

AND p.ID\_Categoria = c1.ID\_Categoria

GROUP BY c3.nom3, p.cost, c1.nom, c2.nom2;



## Shipping logistics module

* + 1. ***Query 1***
       1. *Solution*

SELECT p.nom, p.cognom, c.hores\_conduccio

FROM Persona AS p, Conductor AS c, Localitzacio AS l1, Seguiment AS s, Localitzacio AS l2

WHERE c.ID\_Conductor = p.ID\_Persona

AND p.ID\_localitzacio = l1.ID\_Localitzacio

AND l1.pais <> l2.pais

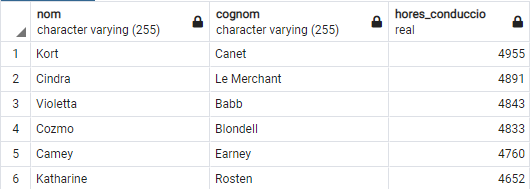
AND l2.ID\_Localitzacio = s.ID\_Localitzacio

AND s.ID\_Conductor = c.ID\_Conductor

GROUP BY p.nom, p.cognom, c.hores\_conduccio

ORDER BY c.hores\_conduccio DESC

LIMIT 6;



* + - 1. *Explanation*

En aquesta query mostrem el nom, cognom i les hores de conducció dels conductors que tenen més hores de conducció i han estat en països diferents del país d’on viuen. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat la condició de que els països on han estat siguin diferents del país d’on viuen. Posem un group by del conductor per a que no es repeteixin els conductors. Fem un order by de les hores de conducció i les ordenem de major a menor, ja que volem mostrar els que tenen més hores de conducció. Hem posat límit 6 ja que ens ho requereix l’enunciat.

* + - 1. *Query validation*

Mostrem els mateixos atributs que a la query principal juntament amb el pais on resideix el treballador i un dels països on ha estat de servei, així podem comprovar que el conductor ha estat realment en un país diferent a on viu. A diferència de la query principal mostrem tots els conductors, així tambe comprovem que estem ensenyant els 6 amb més hores de conducció i podem veure que el resultat és correcte.

SELECT p.nom, p.cognom, c.hores\_conduccio, l1.pais AS "Pais on viu el treballador", l2.pais AS "Pais on ha estat"

FROM Persona AS p, Conductor AS c, Localitzacio AS l1, Seguiment AS s, Localitzacio AS l2

WHERE c.ID\_Conductor = p.ID\_Persona

AND p.ID\_localitzacio = l1.ID\_Localitzacio

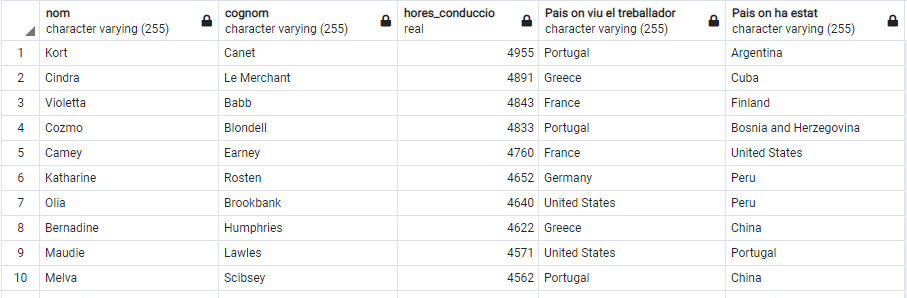
AND l1.pais <> l2.pais

AND l2.ID\_Localitzacio = s.ID\_Localitzacio

AND s.ID\_Conductor = c.ID\_Conductor

GROUP BY p.nom, p.cognom, c.hores\_conduccio, l1.pais, l2.pais

ORDER BY c.hores\_conduccio DESC;



* + 1. ***Query 2***
       1. *Solution*

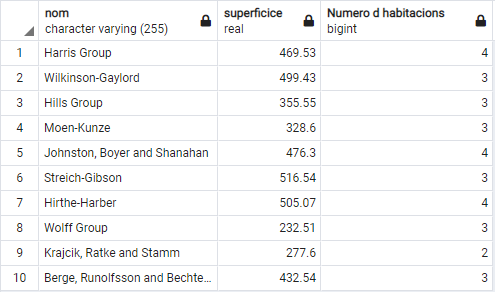
SELECT m.nom, m.superficice, COUNT(h.ID\_Magatzem) AS "Numero d habitacions"

FROM Magatzem AS m, Habitacio AS h

WHERE m.ID\_Magatzem = h.ID\_Magatzem

GROUP BY m.nom, m.superficice, h.ID\_Magatzem

HAVING COUNT(h.ID\_Magatzem) BETWEEN 2 AND 4;



* + - 1. *Explanation*

En aquesta query mostrem el nom, superfície i número d’habitacions dels magatzems que tenen entre 2 i 4 habitacions. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat el group by per a que no es repeteixi cap magatzem i també hem posat un having count del numero d’habitacions i que estigui entre 2 i 4.

* + - 1. *Query validation*

En aquest cas, mostrem els mateixos atributs que a la query principal pero sense la condició que sigui un magatzem que tingui de 2 a 4 habitacions, com podem veure hi ha 30, si afegim la condició, en canvi només ens mostra 10, els que tenen entre 2 i 4 magatzem.

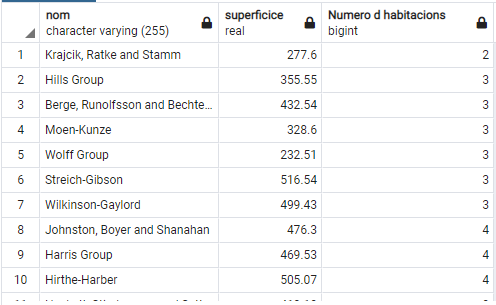
SELECT m.nom, m.superficice, COUNT(h.ID\_Magatzem) AS "Numero d habitacions"

FROM Magatzem AS m, Habitacio AS h

WHERE m.ID\_Magatzem = h.ID\_Magatzem

GROUP BY m.nom, m.superficice, h.ID\_Magatzem

ORDER BY COUNT(h.ID\_Magatzem) ASC;

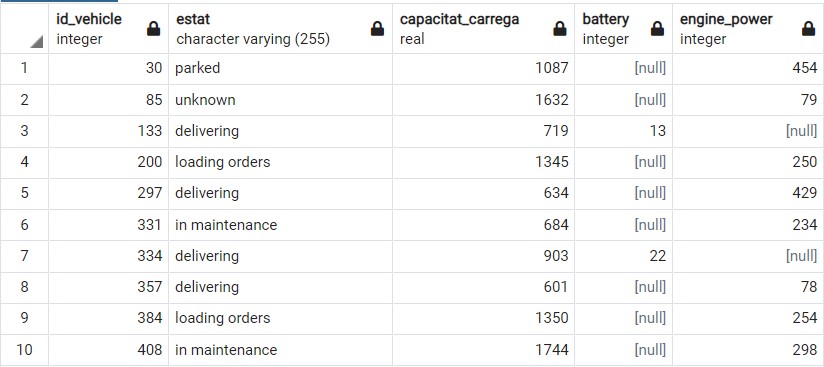


* + 1. ***Query 3***
       1. *Solution*

SELECT v.ID\_Vehicle, v.estat, v.capacitat\_carrega, vd.battery, vd.engine\_power FROM Vehicle AS v, Manteniment AS m, Vehicles\_Drivers AS vd, Persona AS p WHERE p.nom = vd.first\_name

AND p.cognom = vd.last\_name

AND vd.model\_status IS NOT NULL AND v.ID\_Conductor = p.ID\_Persona AND m.ID\_Vehicle = v.ID\_Vehicle AND m.any\_manteniment = '2020';



* + - 1. *Explanation*

En aquesta query mostrem el id, el estat, capacitat de càrrega, la bateria, la potència del motor dels vehicles que han tingut un manteniment aquest últim any. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat la condició de que l’any de manteniment sigui 2020 i que el tinguin model status.

* + - 1. *Query validation*

Com requereix l’enunciat, ens hem assegurat de que hi hagi almenys 10 resultats.

* + 1. ***Query 4***
       1. *Solution*

SELECT m.nom, m.superficice, ROUND(AVG(p.capacitat\_bateria)),l.pais FROM Magatzem AS M, Vehicle as v, Patinet as p, Localitzacio as l WHERE v.ID\_Vehicle = p.ID\_Patinet

AND V.ID\_Magatzem = m.ID\_Magatzem AND m.ID\_Localitzacio = l.ID\_Localitzacio AND l.pais = 'France'

GROUP BY m.nom, m.superficice, l.pais;



* + - 1. *Explanation*

Aquesta no es pot fer ja que ens demanen la mitja de la capacitat de bateria d’un patinet i després ens diuen que només considerem els patinets que tenen un 10% de càrrega, per tant, no té cap sentit.

* + - 1. *Query validation*

No es pot fer.

* + 1. ***Query 5***
       1. *Solution*

SELECT p.nom, p.cognom, e.titol\_treball, dies\_malalties

FROM Persona AS p, Magatzem AS M, Operador AS o, Empleat AS e

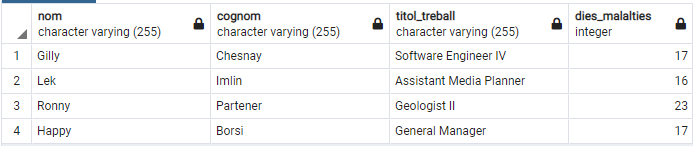
WHERE o.comandes\_enviades >= 10

AND o.ID\_Operador = e.ID\_Empleat

AND e.ID\_Empleat = p.ID\_Persona

AND o.ID\_Magatzem = m.ID\_Magatzem

AND m.superficice > 10000;



* + - 1. *Explanation*

En aquesta query mostrem el nom, cognom, títol de feina i dies malalties dels operadors que han enviat almenys 10 comandes i que treballen en un magatzem de més de 10000 metres quadrats. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat la condició que la superfície del magatzem sigui superior a 10000 i que el número de comandes enviades sigui major o igual a 10.

* + - 1. *Query validation*

Mostrem els mateixos atributs que en la query principal, hem afegit el número de comandes enviades i la superficie del magatzem on treballen, això podem fer les dues comprovacions, que treballi en un magatzem de més de 10000 metres quadrats, en aquest cas els 4 treballadors treballen en el mateix magatzem, i que tingui més de 10 comandes enviades. El resultat coincideix.

SELECT p.nom, p.cognom, e.titol\_treball, dies\_malalties, o.comandes\_enviades, m.superficice

FROM Persona AS p, Magatzem AS M, Operador AS o, Empleat AS e

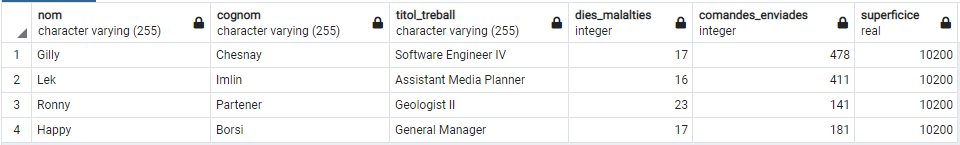
WHERE o.comandes\_enviades >= 10

AND o.ID\_Operador = e.ID\_Empleat

AND e.ID\_Empleat = p.ID\_Persona

AND o.ID\_Magatzem = m.ID\_Magatzem

AND m.superficice > 10000;



## Human resources module

* + 1. ***Query 1***
       1. *Solution*

SELECT p.ID\_Persona, p.nom, p.cognom, e.titol\_treball, e.dies\_malalties, s.quantitat

FROM Persona AS p, Empleat AS e, Sou AS s, Horari AS h

WHERE e.ID\_Empleat = p.ID\_Persona

AND e.ID\_Sou = s.ID\_Sou

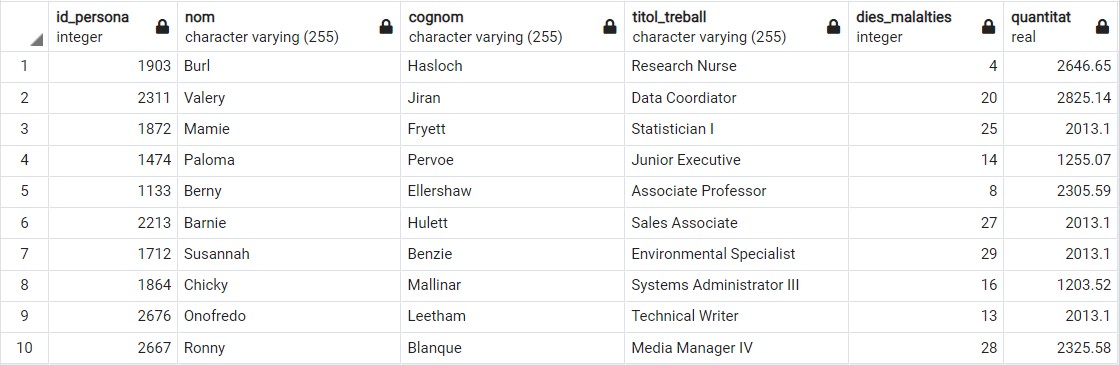
AND e.ID\_Horari = h.ID\_Horari

AND s.complements IS NULL

AND s.sou\_actual = 'true'

GROUP BY p.ID\_Persona, p.nom, p.cognom, e.titol\_treball, e.dies\_malalties, s.quantitat, h.hora\_final, h.hora\_inici

ORDER BY h.hora\_final - h.hora\_inici DESC

LIMIT 10;

* + - 1. *Explanation*

En aquesta query mostrem el id, nom, cognom, títol de feina, dies de malaltia i sou actual dels empleats que treballen més hores i no tenen cap complement en el seu salari. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat la condició de que l’empleat no tingui cap complement i que el seu sou sigui l’actual. També hem posat un group by per no repetir cap empleat . Hem posat un order by de les hores que treballen els empleats ordenats de més a menys hores. El límit és de 10 ja que sinó ens sortien molts empleats.

* + - 1. *Query validation*

Mostra les dades amb hora inici i final i la seva diferència horària i complements “Is Null” ordenats per major horari

SELECT p.ID\_Persona, p.nom, p.cognom, e.titol\_treball, e.dies\_malalties, s.quantitat,s.complements,h.hora\_final-h.hora\_inici as horas ,h.hora\_inici,h.hora\_final

FROM Persona AS p, Empleat AS e, Sou AS s, Horari AS h

WHERE e.ID\_Empleat = p.ID\_Persona

AND e.ID\_Sou = s.ID\_Sou

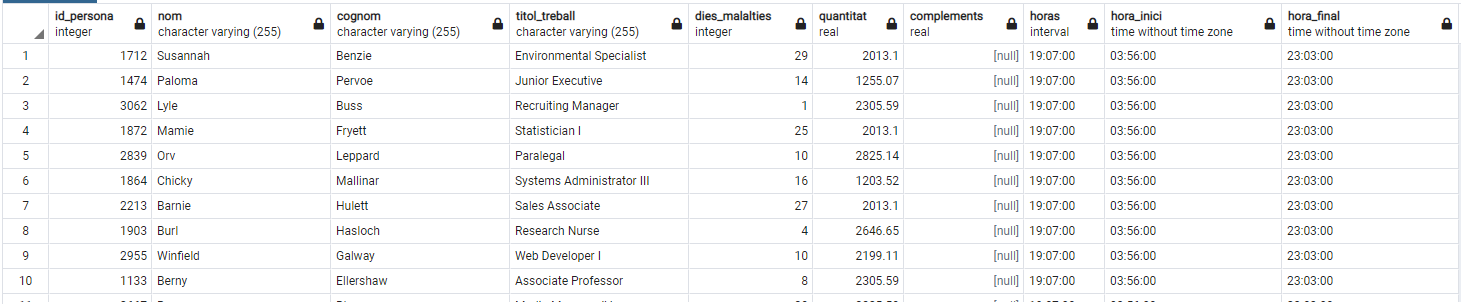
AND e.ID\_Horari = h.ID\_Horari

AND s.complements IS NULL

AND s.sou\_actual = 'true'

GROUP BY p.ID\_Persona, p.nom, p.cognom, e.titol\_treball, e.dies\_malalties, s.quantitat,s.complements, h.hora\_final, h.hora\_inici

ORDER BY h.hora\_final - h.hora\_inici DESC;



* + 1. ***Query 2***
       1. *Solution*

SELECT d.nom, COUNT(of.departament)

FROM Departament AS d, OfertaFeina AS of, Empleat AS e

WHERE e.ID\_Departament = d.ID\_Departament

AND e.ID\_Empleat = of.ID\_Persona

AND d.nom=of.departament

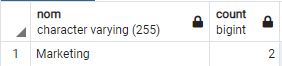
AND of.estat = 'open'

AND e.dies\_malalties < 10

GROUP BY d.nom

ORDER BY COUNT(of.departament) DESC

LIMIT 1;



* + - 1. *Explanation*

En aquesta query mostrem el nom del departament que té més ofertes de feina disponibles, és a dir, que l’estat està en obert (open) i que l’empleat que hagi publicat l’oferta de feina tingui menys de 10 dies de malaltia. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat la condició de que l’estat estigui en obert (open) i que els dies de malaltia de l’empleat sigui menor a 10. Hem posat un group by per a que no es repeteixi cap departament. També hem posat un order by per ordenar de major a menor el número de ofertes de feines. I el límit a 1 ja que ens diuen que mostrem un departament.

* + - 1. *Query validation*

La primera query mostra les ofertes de feina dels departaments amb un count per validar la query a realitzar i en la segona mostrem totes les ofertes de feina amb el seu estat i dies malaltia del treballador.

SELECT d.nom, COUNT(of.departament)

FROM Departament AS d, OfertaFeina AS of, Empleat AS e

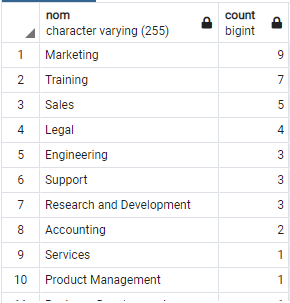
WHERE e.ID\_Departament = d.ID\_Departament

AND e.ID\_Empleat = of.ID\_Persona

AND d.nom=of.departament

GROUP BY d.nom

ORDER BY COUNT(of.departament) DESC;



En els resultats d'aquesta query, veiem que pel departament de Marketing, només hi ha 2 ofertes de feina que compleixen que l'estat sigui obert (open) i que l'empleat que ha publicat la oferta tingui menys de 10 dies de malaltia. El resultat coincideix amb el resultat de la query principal.

SELECT d.nom, e.ID\_Empleat,e.dies\_malalties,of.estat

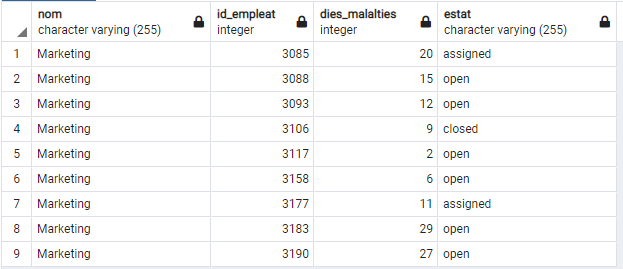
FROM Departament AS d, OfertaFeina AS of, Empleat AS e

WHERE e.ID\_Departament = d.ID\_Departament

AND e.ID\_Empleat = of.ID\_Persona

AND d.nom = 'Marketing'

AND d.nom = of.departament;



* + 1. ***Query 3***
       1. *Solution*

SELECT of.descripcio, of.estat, of.data\_publicacio

FROM OfertaFeina AS of, Candidat AS c

WHERE of.ID\_OfertaFeina = c.ID\_OfertaFeina

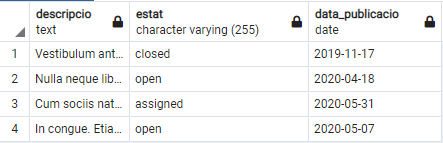
AND c.curriculum LIKE '%SQL%'

AND c.curriculum LIKE '%Java%'

GROUP BY of.ID\_OfertaFeina

ORDER BY COUNT(of.ID\_OfertaFeina) DESC

LIMIT 4;



* + - 1. *Explanation*

En aquesta query mostrem les quatre ofertes de feina, la seva descripció, l'estat i la data, que tenen més candidats els quals en el seu currículum consti que han treballat amb SQL i Java. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat les condicions de que el currículum contingui les paraules SQL i Java. Hem fet un group by per a que no es repeteixi cap oferta de feina. Hem fet un order by de major a menor per a que ens ho ordeni segons tingui els major número de candidats. El límit 4 ja que ens ho especifica l’enuciat.

* + - 1. *Query validation*

Aquesta query mostra totes les ofertes de feina sense SQL i JAVA en el seu currículum i les ordena de més a menys oferta de feina.

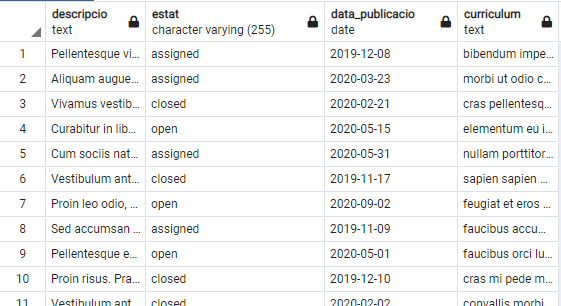
SELECT of.descripcio, of.estat, of.data\_publicacio, c.curriculum

FROM OfertaFeina AS of, Candidat AS c

WHERE of.ID\_OfertaFeina = c.ID\_OfertaFeina

GROUP BY of.descripcio, of.estat, of.data\_publicacio, c.ID\_Candidat

ORDER BY COUNT(of.ID\_OfertaFeina) DESC;



* + 1. ***Query 4***
       1. *Solution*

SELECT DISTINCT p.nom, p.cognom, p.correu\_electronic, e.titol\_treball, e.dies\_vacances, l1.ciutat

FROM Persona AS p, Empleat AS e, Localitzacio AS l1, Edifici AS ed, Localitzacio AS l2, Departament AS d

WHERE p.ID\_Persona = e.ID\_Empleat

AND p.ID\_Localitzacio = l1.ID\_Localitzacio

AND ed.ID\_Localitzacio = l2.ID\_Localitzacio

AND e.ID\_Departament = d.ID\_Departament

AND d.ID\_Edifici = ed.ID\_Edifici

AND l1.ciutat <> l2.ciutat

LIMIT 10;



* + - 1. *Explanation*

En aquesta query mostrem el nom, cognom, correu electrònic, títol treball, dies de vacances i la ciutat dels empleats que treballen en un edifici localitzat en una ciutat diferent a on viuen els mateixos empleats. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat la condició de que la ciutat on s’ubica l’edifici sigui diferent a la ciutat on viu l’empleat. Posem el group by per a que no ens surti cap empleat repetit i límit 10 perquè sinó ens sortien moltíssims empleats.

* + - 1. *Query validation*

Query que mostra les ciutats on viuen i on treballen, així podem comprovar que realment les ciutats son diferents.

SELECT DISTINCT p.nom, p.cognom, p.correu\_electronic, e.titol\_treball, e.dies\_vacances, l1.ciutat as "on viu", l2.ciutat as "ciutat edifici"

FROM Persona AS p, Empleat AS e, Localitzacio AS l1, Edifici AS ed, Localitzacio AS l2, Departament AS d

WHERE p.ID\_Persona = e.ID\_Empleat

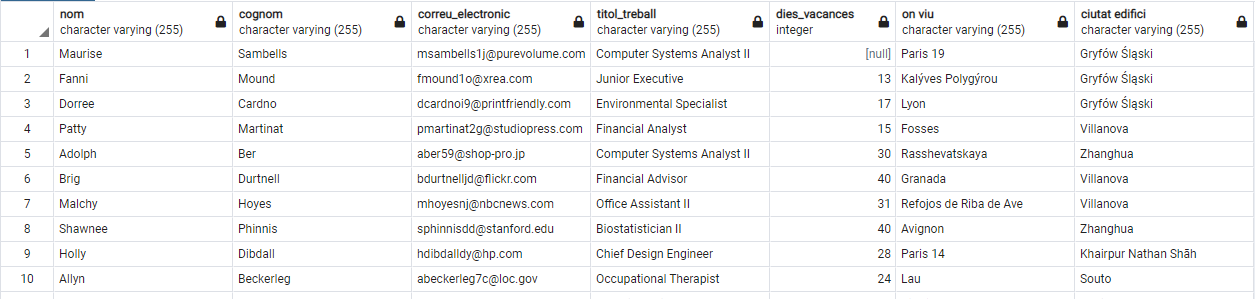
AND p.ID\_Localitzacio = l1.ID\_Localitzacio

AND ed.ID\_Localitzacio = l2.ID\_Localitzacio

AND e.ID\_Departament = d.ID\_Departament

AND d.ID\_Edifici = ed.ID\_Edifici

AND l1.ciutat <> l2.ciutat;



I la segona query mostra l'exemple d'un que viu i treballa a la mateixa, forçant així l’error. Com podem comprovar, el resultat d’aquesta segona query, no surt en la primera query de validació, ja que trenca la condició de que no visqui i treballi en la mateixa ciutat.

SELECT DISTINCT p.nom, p.cognom, p.correu\_electronic, e.titol\_treball, e.dies\_vacances, l1.ciutat as "on viu", l2.ciutat as "ciutat edifici"

FROM Persona AS p, Empleat AS e, Localitzacio AS l1, Edifici AS ed, Localitzacio AS l2, Departament AS d

WHERE p.ID\_Persona = e.ID\_Empleat

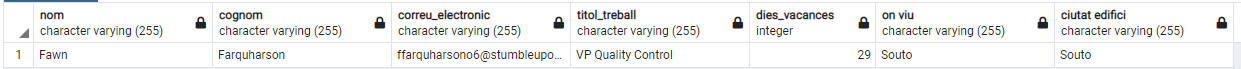
AND p.ID\_Localitzacio = l1.ID\_Localitzacio

AND ed.ID\_Localitzacio = l2.ID\_Localitzacio

AND e.ID\_Departament = d.ID\_Departament

AND d.ID\_Edifici = ed.ID\_Edifici

AND l1.ciutat = l2.ciutat;



* + 1. ***Query 5***
       1. *Solution*
       2. *Explanation*

No la sabem fer.

* + - 1. *Query validation*

## Cross-process analyses

* + 1. ***Query 1***
       1. *Solution*

SELECT DISTINCT p.nom, p.cognom, p.correu\_electronic

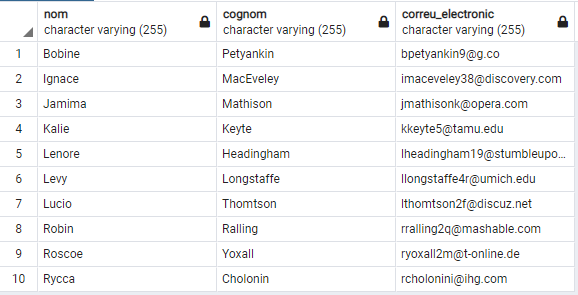
FROM Persona AS p, Candidat AS c, Article AS a, OfertaFeina AS of

WHERE c.ID\_OfertaFeina = of.ID\_OfertaFeina

AND of.ID\_Persona = p.ID\_Persona

AND a.ID\_Client = p.ID\_Persona

AND a.estrelles < '3';



* + - 1. *Explanation*

En aquesta query mostrem el nom, cognom i correu electrònic dels candidats que han escrit una ressenya negativa, és a dir que la puntuació d'aquesta sigui inferior a 3. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat la condició de que les estrelles del article valorat sigui menor a 3. Hem posat un group by per a que no ens surti repetit cap candidat.

* + - 1. *Query validation*

Primer fem un insert afegint una review d’un candidat on valora la seva compra, puntuant-la amb un 4, d’aquesta manera assegurem que tot hi havent-hi articles amb puntuacions superiors a tres, no les mostrem a la query principal.

INSERT INTO Article(ID\_Article, preu, descompte, ID\_Client, ID\_Comanda, review, estrelles, ID\_ArticleClient)

VALUES (1035, 56.32, 12, 3172, 4, 'All good', 4, 647);

En aquesta query mostrem totes les valoracions de tots el candidat, independentment de la puntuació, com podem veure apareix la fila introduida per l’insert.

SELECT p.nom, p.cognom, p.correu\_electronic, a.estrelles

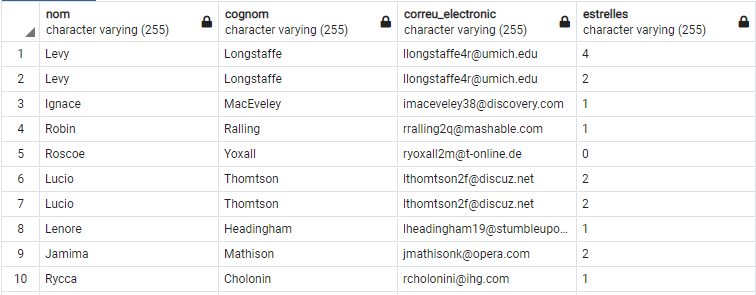
FROM Persona AS p, Candidat AS c, Article AS a, OfertaFeina AS of

WHERE c.ID\_OfertaFeina = of.ID\_OfertaFeina

AND of.ID\_Persona = p.ID\_Persona

AND a.ID\_Client = p.ID\_Persona

ORDER BY p.ID\_Persona DESC;



En aquesta, en canvi, mostrem el resultat de la query principal, afegint-hi les puntuacions, així podem veure que totes són menors a 3, i que l’usuari introduït amb l’insert no es mostra.

SELECT p.nom, p.cognom, p.correu\_electronic, a.estrelles

FROM Persona AS p, Candidat AS c, Article AS a, OfertaFeina AS of

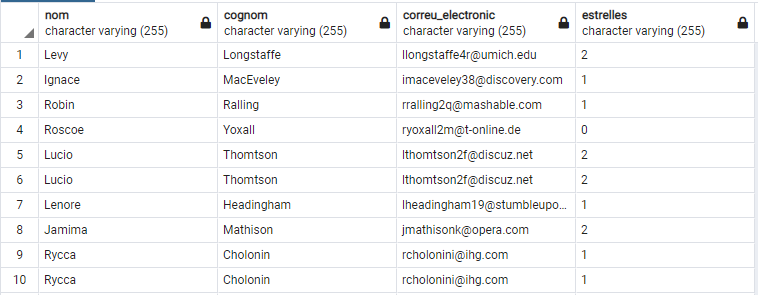
WHERE c.ID\_OfertaFeina = of.ID\_OfertaFeina

AND of.ID\_Persona = p.ID\_Persona

AND a.ID\_Client = p.ID\_Persona

AND a.estrelles < '3'

ORDER BY p.ID\_Persona DESC;



* + 1. ***Query 2***
       1. *Solution*

SELECT p.nom, p.cognom, p.correu\_electronic, e.titol\_treball, v.bonificacio\_vendes, s.quantitat

FROM Persona AS p, Venedor AS v, Empleat AS e, Sou AS s

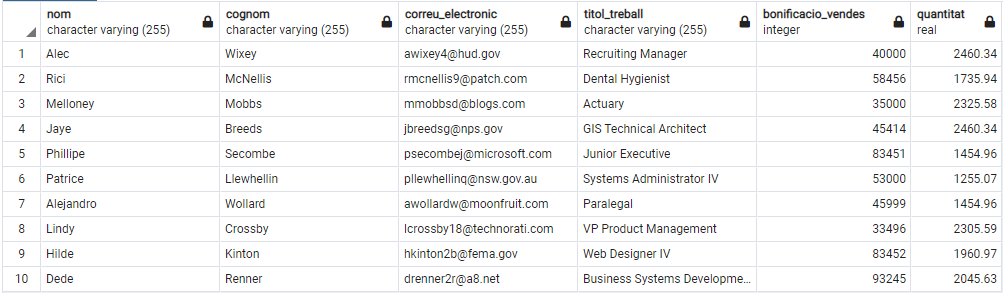
WHERE p.ID\_Persona = e.ID\_Empleat

AND e.ID\_Empleat = v.ID\_Venedor

AND e.ID\_Sou = s.ID\_Sou

AND v.bonificacio\_vendes > 10 \* s.quantitat

AND s.sou\_actual = 'true';



* + - 1. *Explanation*

En aquesta query mostrem el nom, cognom, correu electrònic, títol de treball, vendes i el salari actual dels venedors on els beneficis obtinguts a partir de les seves vendes és deu vegades superior al seu salari actual. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat la condició de que els beneficis obtinguts a partir de les seves vendes siguin 10 vegades superiors al seu salari actual. També que el sou del venedor sigui l’actual (true). Hem posat el group by per que no ens surti repetit cap venedor.

* + - 1. *Query validation*

Mostrem tots el venedors amb sou actual = true

I ordena amb el càlcul de v.bonificaio>10\*s.quantitat i podem observar el resultat de la query és l'esperat.

SELECT p.nom, p.cognom, p.correu\_electronic, e.titol\_treball, v.bonificacio\_vendes, s.quantitat,ROUND(v.bonificacio\_vendes/s.quantitat)

FROM Persona AS p, Venedor AS v, Empleat AS e, Sou AS s

WHERE p.ID\_Persona = e.ID\_Empleat

AND e.ID\_Empleat = v.ID\_Venedor

AND e.ID\_Sou = s.ID\_Sou

AND s.sou\_actual = 'true'

ORDER BY v.bonificacio\_vendes > 10 \* s.quantitat DESC;



* + 1. ***Query 3***
       1. *Solution*

SELECT p.nom, p.cognom, p.correu\_electronic, e.titol\_treball, c.hores\_conduccio, s.quantitat

FROM Persona AS p, Empleat AS e, Sou AS s, Conductor AS c, Horari AS h, Disposa as d, Telefon as t

WHERE p.ID\_Persona = e.ID\_Empleat

AND e.ID\_Empleat = c.ID\_Conductor

AND e.ID\_Sou = s.ID\_Sou

AND e.ID\_Horari = h.ID\_Horari

AND s.sou\_actual = 'true'

AND e.ID\_Empleat = d.ID\_Empleat

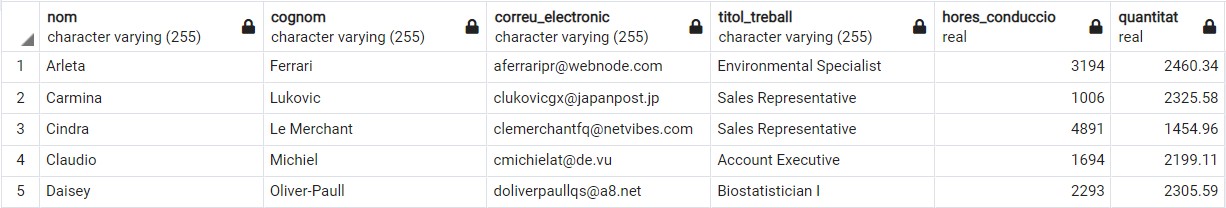
AND d.ID\_Telefon = t.ID\_Telefon

GROUP BY p.nom, p.cognom, p.correu\_electronic, e.titol\_treball, c.hores\_conduccio, s.quantitat, h.hora\_final, h.hora\_inici

HAVING h.hora\_final - h.hora\_inici = '07:00:00'

ORDER BY COUNT(d.ID\_Empleat)DESC

LIMIT 5;



* + - 1. *Explanation*

En aquesta query mostrem el nom, cognom, correu electrònic, títol de treball, vendes i el salari actual dels conductors els quals tenen un horari complet de set hores i hagin utilitzat més telèfons. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat la condició de que el conductor estigui cobrant el seu sou actual (true). Hem posat el group by perquè no ens surti repetit cap conductor. I en el having posem que el conductor treballi 7 hores. El límit 5 perquè ho especifica l’enunciat.

* + - 1. *Query validation*

Nosaltres havíem assignat un telèfon per cada persona. Per aquesta raó per fer la comprovació inserim un telèfon més en un dels treballadors que té l'horari de 7 hores i en la query de validació es mostra com s'ordena per mòbils utilitzats

INSERT INTO DISPOSA(ID\_Empleat,ID\_Telefon)VALUES (1534,1);

SELECT p.nom, p.cognom, p.correu\_electronic, e.titol\_treball, c.hores\_conduccio, s.quantitat, COUNT(d.ID\_Empleat),d.ID\_Empleat

FROM Persona AS p, Empleat AS e, Sou AS s, Conductor AS c, Horari AS h, Disposa as d, Telefon as t

WHERE p.ID\_Persona = e.ID\_Empleat

AND e.ID\_Empleat = c.ID\_Conductor

AND e.ID\_Sou = s.ID\_Sou

AND e.ID\_Horari = h.ID\_Horari

AND s.sou\_actual = 'true'

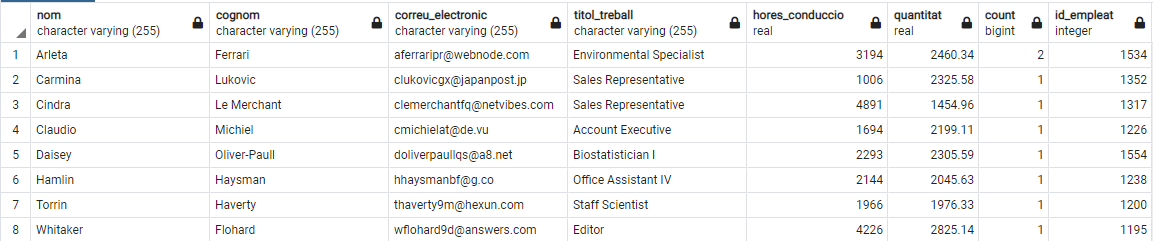
AND e.ID\_Empleat = d.ID\_Empleat

AND d.ID\_Telefon = t.ID\_Telefon

GROUP BY p.nom, p.cognom, p.correu\_electronic, e.titol\_treball, c.hores\_conduccio, s.quantitat, h.hora\_final, h.hora\_inici,d.ID\_Empleat

HAVING h.hora\_final - h.hora\_inici = '07:00:00'

ORDER BY COUNT(d.ID\_Empleat)DESC



* + 1. ***Query 4***
       1. *Solution*

SELECT DISTINCT e.nom, l.ciutat

FROM Edifici AS e, Localitzacio AS l, Departament AS d, Magatzem AS m,localitzacio AS l2

WHERE e.ID\_Edifici = d.ID\_Edifici

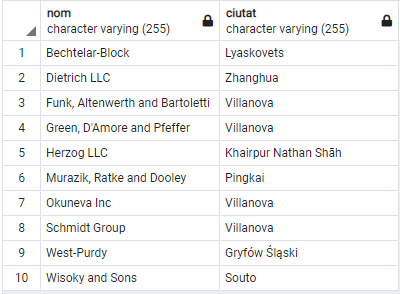
AND e.ID\_Localitzacio = l.ID\_Localitzacio

AND m.ID\_Localitzacio = l2.ID\_Localitzacio

AND l.ciutat = l2.ciutat

AND d.nom = 'Human Resources'

GROUP BY e.nom, l.ciutat, m.superficice HAVING (m.superficice) < 300;



* + - 1. *Explanation*

En aquesta query mostrem el nom i la ciutat dels edificis que tenen el departament de Recursos Humans ubicat en una ciutat on també puguem trobar un magatzem amb una superfície inferior a 300 metres quadrats. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat les condicions de que la ciutat del edifici sigui la mateixa que la del magatzem i que el nom del departament sigui el de recursos humans (Human Resources). Posem el group by perquè no surti repetit cap edifici.

* + - 1. *Query validation*

La primera query ens mostra els edificis que tinguin un magatzem a la mateixa ciutat

SELECT DISTINCT e.nom, l.ciutat,l2.ciutat,d.nom

FROM Edifici AS e, Localitzacio AS l, Departament AS d, Magatzem AS m,localitzacio AS l2

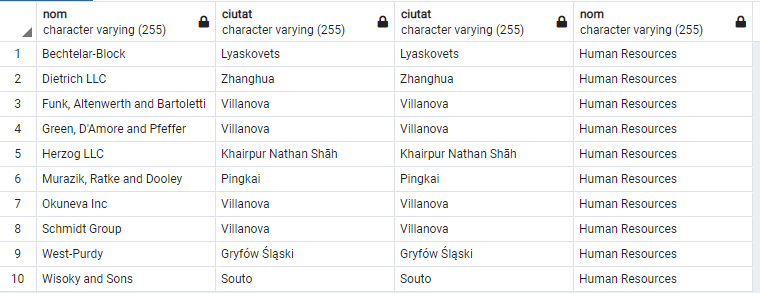
WHERE e.ID\_Edifici = d.ID\_Edifici

AND e.ID\_Localitzacio = l.ID\_Localitzacio

AND m.ID\_Localitzacio = l2.ID\_Localitzacio

AND l.ciutat = l2.ciutat

AND d.nom = 'Human Resources';



Ens mostra tots els edificis de Human Reosurces i observem com ID\_Edifici 12 apareix aqui i no en l'anterior això vol dir que no te un magatzem en la mateixa ciutat

SELECT DISTINCT e.nom, l.ciutat,d.nom

FROM Edifici AS e, Localitzacio AS l, Departament AS d, Magatzem AS m,localitzacio AS l2

WHERE e.ID\_Edifici = d.ID\_Edifici

AND e.ID\_Localitzacio = l.ID\_Localitzacio

AND d.nom = 'Human Resources';



Observem tots els magatzems més petits de 300 metres que estan a la mateixa ciutat que un edifici. Un edifici pot tenir més d'un magatzem

SELECT DISTINCT e.nom, l.ciutat,l2.ciutat,m.superficice

FROM Edifici AS e, Localitzacio AS l, Departament AS d, Magatzem AS m,localitzacio AS l2

WHERE e.ID\_Edifici = d.ID\_Edifici

AND e.ID\_Localitzacio = l.ID\_Localitzacio

AND m.ID\_Localitzacio = l2.ID\_Localitzacio

AND l.ciutat = l2.ciutat

AND d.nom = 'Human Resources'

GROUP BY e.nom, l.ciutat, m.superficice,l2.ciutat HAVING (m.superficice) < 300;



Aquesta podem observar com 4 dels edificis també tenen diversos magatzems amb més de 300 m

SELECT DISTINCT e.nom, l.ciutat,l2.ciutat,m.superficice

FROM Edifici AS e, Localitzacio AS l, Departament AS d, Magatzem AS m,localitzacio AS l2

WHERE e.ID\_Edifici = d.ID\_Edifici

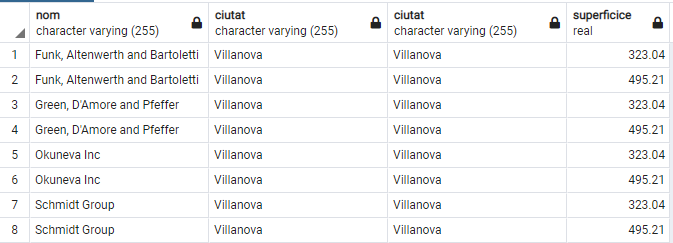
AND e.ID\_Localitzacio = l.ID\_Localitzacio

AND m.ID\_Localitzacio = l2.ID\_Localitzacio

AND l.ciutat = l2.ciutat

AND d.nom = 'Human Resources'

GROUP BY e.nom, l.ciutat, m.superficice,l2.ciutat HAVING (m.superficice) > 300;



* + 1. ***Query 5***
       1. *Solution*

SELECT p.nom, p.cognom, p.correu\_electronic, e.titol\_treball, t.productes\_muntats, s.quantitat

FROM Persona AS p, Empleat AS e, Treballador AS t, Sou AS s, Horari AS h, Fabrica AS f, Producte AS pr

WHERE p.ID\_Persona = e.ID\_Empleat

AND p.ID\_Persona = t.ID\_Treballador

AND t.ID\_Fabrica = f.ID\_Fabrica

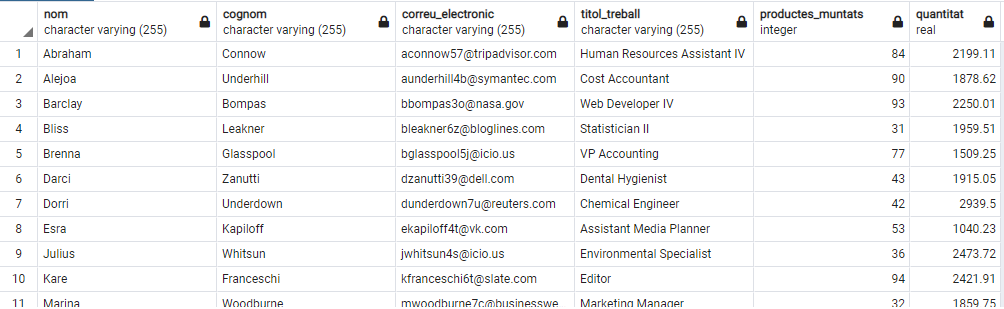
AND e.ID\_Sou = s.ID\_Sou

AND t.ID\_producte\_preferit = pr.codi

AND e.ID\_Horari = h.ID\_Horari

AND f.linies\_muntatge = 3

AND h.hora\_final - h.hora\_inici = '07:00:00'

GROUP BY p.nom, p.cognom, p.correu\_electronic, e.titol\_treball, t.productes\_muntats, s.quantitat;

* + - 1. *Explanation*

En aquesta query mostrem el nom, cognom, correu electrònic, títol de treball, productes muntats i sou actual dels treballadors que tinguin un horari de set hores i que el seu producte preferit hagi estat muntat en una fàbrica que tingui 3 línies de muntatge en total. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat la condició de que la fabrica tingui 3 línies de muntatge. Hem fet un group by per a que no surti repetit cap treballador. En el having posem que el treballador tingui un horari de 7 hores.

* + - 1. *Query validation*

La primera query mostra tots els treballadors amb les seves hores de inici i final amb lines muntatge = 3

SELECT p.nom, p.cognom, p.correu\_electronic, e.titol\_treball, t.productes\_muntats, s.quantitat,h.hora\_inici , h.hora\_final

FROM Persona AS p, Empleat AS e, Treballador AS t, Sou AS s, Horari AS h, Fabrica AS f, Producte AS pr

WHERE p.ID\_Persona = e.ID\_Empleat

AND p.ID\_Persona = t.ID\_Treballador

AND t.ID\_Fabrica = f.ID\_Fabrica

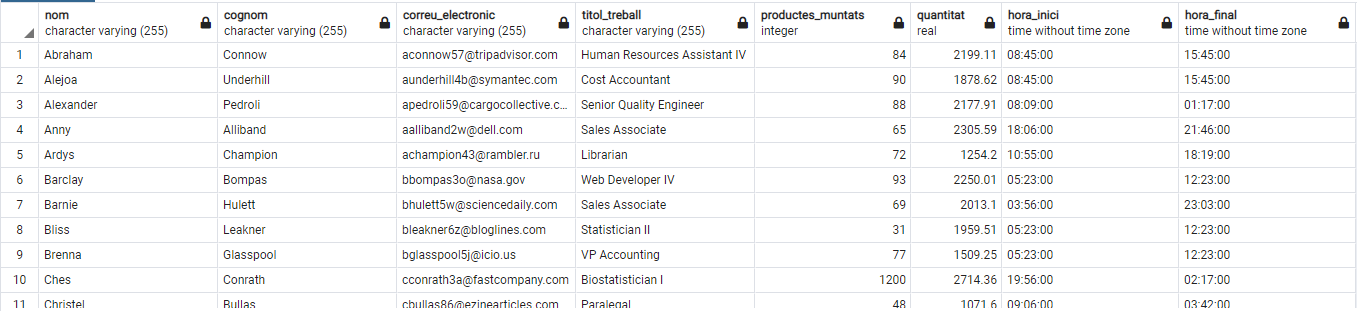
AND e.ID\_Sou = s.ID\_Sou

AND t.ID\_producte\_preferit = pr.codi

AND e.ID\_Horari = h.ID\_Horari

AND f.linies\_muntatge = 3

GROUP BY p.nom, p.cognom, p.correu\_electronic, e.titol\_treball, t.productes\_muntats, s.quantitat, h.hora\_final, h.hora\_inici;



La segona query mostra els treballadors que treballen 7 hores i les línies de muntatge del seu producte preferit i observem que la nostra query funciona correctament, ja que no agafa aquest treballador amb nombre diferent de línies de muntatge.

SELECT p.nom, p.cognom, p.correu\_electronic, e.titol\_treball, t.productes\_muntats, s.quantitat,f.linies\_muntatge

FROM Persona AS p, Empleat AS e, Treballador AS t, Sou AS s, Horari AS h, Fabrica AS f, Producte AS pr

WHERE p.ID\_Persona = e.ID\_Empleat

AND p.ID\_Persona = t.ID\_Treballador

AND t.ID\_Fabrica = f.ID\_Fabrica

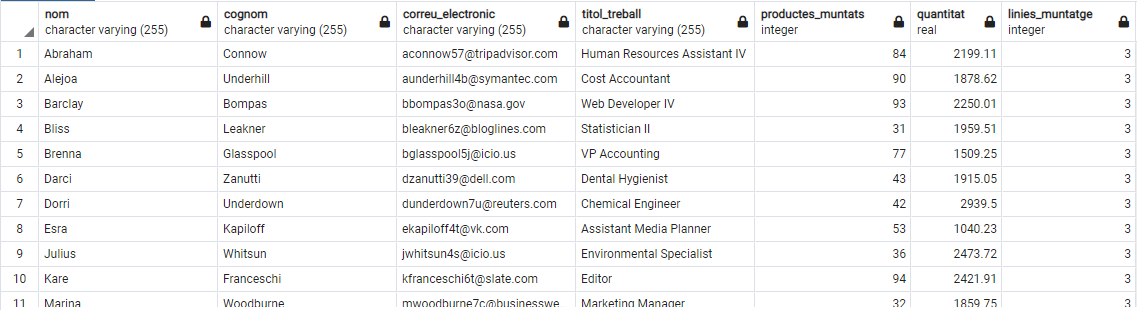
AND e.ID\_Sou = s.ID\_Sou

AND t.ID\_producte\_preferit = pr.codi

AND e.ID\_Horari = h.ID\_Horari

AND h.hora\_final - h.hora\_inici = '07:00:00'

GROUP BY p.nom, p.cognom, p.correu\_electronic, e.titol\_treball, t.productes\_muntats, s.quantitat, f.linies\_muntatge;



* + 1. ***Query 6***
       1. *Solution*
       2. *Explanation*

No la sabem fer.

* + - 1. *Query validation*
    1. ***Query 7***
       1. *Solution*

SELECT p.nom, p.mida, p.pes, p.cost

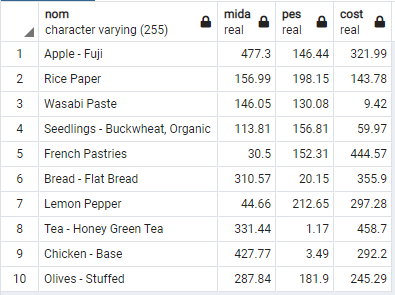
FROM Producte AS p, Article AS a, Categoria1 AS c1

WHERE p.ID\_Categoria = c1.ID\_Categoria

AND a.ID\_Article = p.ID\_Article

AND c1.nom = 'Slot Cars'

AND a.estrelles < '3';



* + - 1. *Explanation*

En aquesta query mostrem el nom, mida, pes i cost total dels productes que tenen una valoració negativa, és a dir, menys de tres estrelles en total, i a més a més el producte pertanyi a la categoria de "Slot Cars". Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat les condicions que la categoria del producte pertanyi a “Slot Cars” i que la valoració del producte sigui inferior a 3 estrelles.

* + - 1. *Query validation*

Només hi ha 10 opinions de productes Slot Cars per aquesta raó per comprovar que funciona bé hem fet una query que mostri valoració Estrelles inferiors a 3.

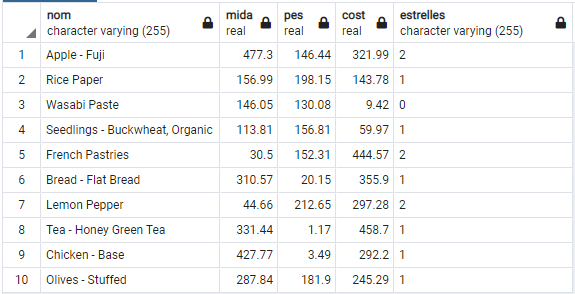
SELECT p.nom, p.mida, p.pes, p.cost,a.estrelles

FROM Producte AS p, Article AS a, Categoria1 AS c1

WHERE p.ID\_Categoria = c1.ID\_Categoria

AND a.ID\_Article = p.ID\_Article

AND c1.nom = 'Slot Cars';



On observem que la comprovació de les estrelles està ben feta

SELECT p.nom, p.mida, p.pes, p.cost,a.estrelles

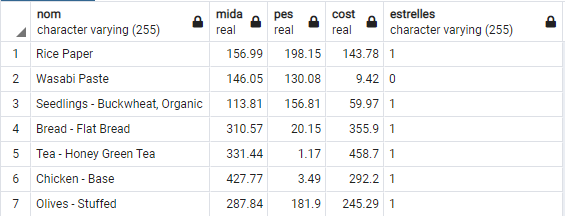
FROM Producte AS p, Article AS a, Categoria1 AS c1

WHERE p.ID\_Categoria = c1.ID\_Categoria

AND a.ID\_Article = p.ID\_Article

AND c1.nom = 'Slot Cars'

AND a.estrelles < '2';



Query amb valoració inferior a 3 però sense pertànyer exclusivament a Slot Cars.

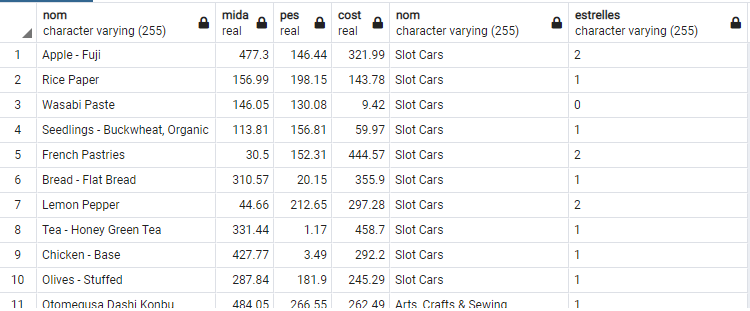
SELECT p.nom, p.mida, p.pes, p.cost,c1.nom,a.estrelles

FROM Producte AS p, Article AS a, Categoria1 AS c1

WHERE p.ID\_Categoria = c1.ID\_Categoria

AND a.ID\_Article = p.ID\_Article

AND a.estrelles < '3';



* + 1. ***Query 8***
       1. *Solution*

SELECT p.nom, p.cognom, p.correu\_electronic, p.numero\_telefon, l.ciutat

FROM Persona AS p, Localitzacio AS l, Producte AS po, Fabrica AS f, Localitzacio AS l1, Article AS a

WHERE po.ID\_fabrica = f.ID\_fabrica

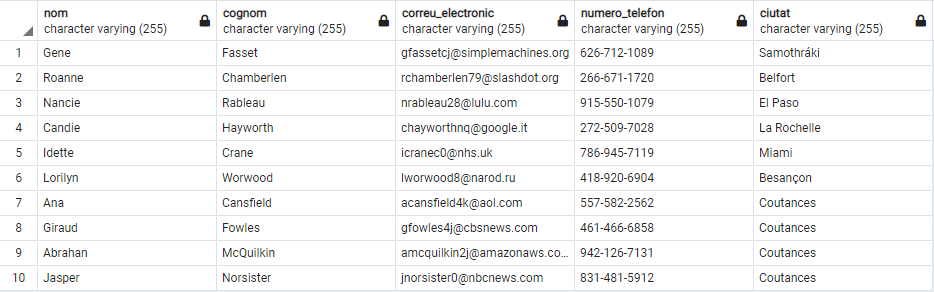
AND f.ID\_Localitzacio = l1.ID\_Localitzacio

AND po.ID\_Article = a.ID\_Article

AND a.ID\_Client = p.ID\_Persona

AND p.ID\_Localitzacio = l.ID\_Localitzacio

AND l.ciutat = l1.ciutat;



* + - 1. *Explanation*

En aquesta query mostrem el nom. cognom, correu electrònic, número de telèfon i ciutat dels clients que hagin comprat un producte i que aquest hagi estat muntat en una fàbrica situada en la mateixa ciutat on viu el mateix client. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat la condició de que la ciutat on viu del client sigui la mateixa que la ciutat de la fàbrica on ha comprat un producte.

* + - 1. *Query validation*

En aquesta QUERY mostrem els productes comprats pels clients, el lloc on estan fabricats i el lloc on es compra siguin iguals.

SELECT p.nom,p.ID\_persona,l.ciutat,l1.ciutat

FROM Persona AS p, Localitzacio AS l, Producte AS po, Fabrica AS f, Localitzacio AS l1, Article AS a

WHERE po.ID\_fabrica = f.ID\_fabrica

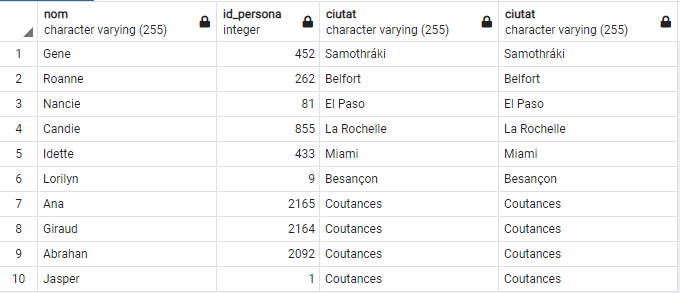
AND f.ID\_Localitzacio = l1.ID\_Localitzacio

AND po.ID\_Article = a.ID\_Article

AND a.ID\_Client = p.ID\_Persona

AND p.ID\_Localitzacio = l.ID\_Localitzacio

AND l.ciutat = l1.ciutat;



En aquesta QUERY mostrem els productes comprats pels clients, el lloc on estan fabricats i on es compren siguin iguals o diferents

SELECT p.nom,p.ID\_persona,l1.ciutat as "Ciutat fabricacio",l.ciutat as "Ciutat Compra producte"

FROM Persona AS p, Localitzacio AS l, Producte AS po, Fabrica AS f, Localitzacio AS l1, Article AS a

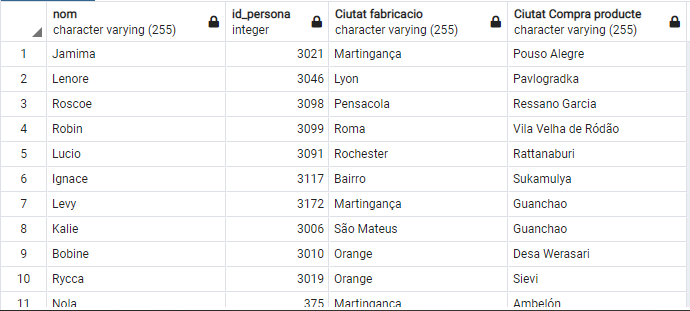
WHERE po.ID\_fabrica = f.ID\_fabrica

AND f.ID\_Localitzacio = l1.ID\_Localitzacio

AND po.ID\_Article = a.ID\_Article

AND a.ID\_Client = p.ID\_Persona

AND p.ID\_Localitzacio = l.ID\_Localitzacio;



* + 1. ***Query 9***
       1. *Solution*
       2. *Explanation*

No la sabem fer.

* + - 1. *Query validation*
    1. ***Query 10***
       1. *Solution*

SELECT t.ID\_Treballador, p.nom, p.cognom, pr.nom

FROM Treballador as t, Producte as pr, Persona as p, Localitzacio as l, Fabrica as f, Magatzem AS m

WHERE t.ID\_producte\_preferit = pr.codi

ANd t.ID\_Treballador = p.ID\_Persona

AND t.ID\_Fabrica = f.ID\_Fabrica

AND f.ID\_Localitzacio = l.ID\_Localitzacio

AND l.pais = 'Italy'

AND m.ID\_Fabrica = f.ID\_Fabrica

AND m.superficice > 1100;



* + - 1. *Explanation*

En aquesta query mostrem els productes preferits dels treballadors que treballen un magatzem localitzat a Itàlia, el magatzem ha de tenir 1100 metres quadrats. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat les condicions de que el país està localitzat a Itàlia i que la superfície del magatzem sigui superior a 1100 metres quadrats.

* + - 1. *Query validation*

En aquesta query mostrem tots els treballadors que treballen a Itàlia i es pot veure la superfície del magatzem on treballen. Com podem veure, hi ha treballadors que treballen a 2 magatzems diferents però, els treballadors, Daveta i Willow al no treballar en un magatzem de més de 1100 no apareixen en la query a realitzar.

SELECT t.ID\_Treballador, p.nom, p.cognom, pr.nom,m.superficice,l.pais,m.ID\_Magatzem

FROM Treballador as t, Producte as pr, Persona as p, Localitzacio as l, Fabrica as f, Magatzem AS m

WHERE t.ID\_producte\_preferit = pr.codi

ANd t.ID\_Treballador = p.ID\_Persona

AND t.ID\_Fabrica = f.ID\_Fabrica

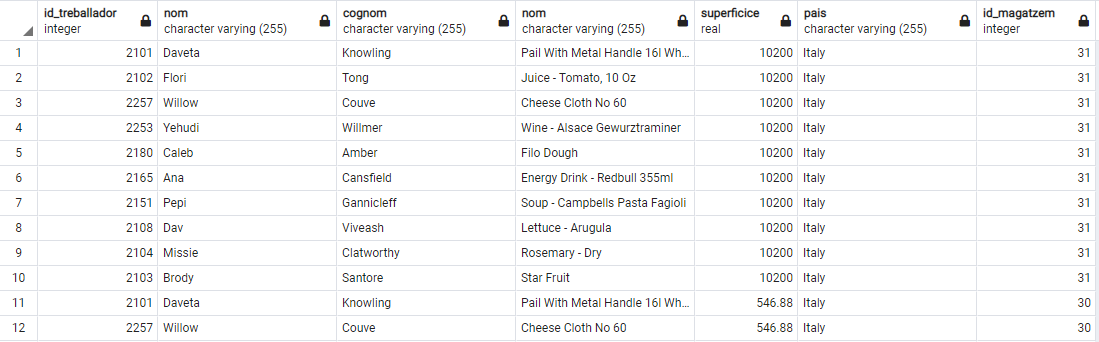
AND f.ID\_Localitzacio = l.ID\_Localitzacio

AND l.pais = 'Italy'

AND m.ID\_Fabrica = f.ID\_Fabrica

ORDER BY m.superficice DESC;

Per fer aquesta validació, vam fer uns inserts en el seu moment, tots els inserts es troben en l’script de Model Físic.



* + 1. ***Query 11***
       1. *Solution*

SELECT DISTINCT p.nom, p.cognom

FROM Persona AS p, Vehicle AS v, Camio AS c, Seguiment AS s, Localitzacio AS l, Comanda AS co

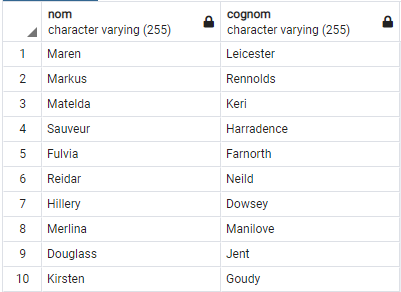
WHERE p.ID\_Persona=co.ID\_Client

AND s.ID\_Localitzacio=co.ID\_localitzacio

AND v.ID\_Vehicle=s.ID\_Vehicle

AND v.ID\_Vehicle=c.ID\_Camio

AND c.matricula LIKE '%79';



* + - 1. *Explanation*

En aquesta query mostrem el nom els clients que han fet una comanda i aquest ha sigut transportat per un camió que la seva matrícula finalitza amb 79. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat la condició que la matrícula acabi amb 79. Posem un group by al final de nom el client, ja que ens interessa només el nom i no si ha fet diverses comandes, transportades amb el mateix camió.

* + - 1. *Query validation*

Per començar, fem aquest 6 insert, els tres primer introdueixen un nou camió amb la matricula acabada en 79, els altres tres, en canvi, inserten un camió amb una matricula que no acaba en 79, així podem comprovar que tot funciona bé.

INSERT INTO Vehicle(ID\_Vehicle, model, estat, capacitat\_carrega, ID\_Magatzem, ID\_Conductor, ID\_Localitzacio)

VALUES(577, 'Volkswagen', 'Delivering', 2500, 25, 1417, 2556);

INSERT INTO Camio(ID\_Camio, matricula, potencia\_motor)

VALUES(577, 'LGS 1179', 220);

INSERT INTO Seguiment(ID\_Conductor, ID\_Vehicle, ID\_Localitzacio, durada)

VALUES(1417, 577, 56, 56);

--Camio que la matricula no acaba en 79

INSERT INTO Vehicle(ID\_Vehicle, model, estat, capacitat\_carrega, ID\_Magatzem, ID\_Conductor, ID\_Localitzacio)

VALUES(578, 'Volkswagen', 'Delivering', 2500, 25, 1417, 2556);

INSERT INTO Camio(ID\_Camio, matricula, potencia\_motor)

VALUES(578, 'MPB 1180', 220);

INSERT INTO Seguiment(ID\_Conductor, ID\_Vehicle, ID\_Localitzacio, durada)

VALUES(1417, 578, 56, 56);

En aquesta query seleccionem tot els camions, sense mirar que la matricula acabi en 79, així podem comprobar que els inserts anteriors realment han funcionat.

SELECT DISTINCT p.nom, c.matricula

FROM Persona AS p, Vehicle AS v, Camio AS c, Seguiment AS s, Localitzacio AS l, Comanda AS co

WHERE p.ID\_Persona = co.ID\_Client

AND s.ID\_Localitzacio = co.ID\_localitzacio

AND v.ID\_Vehicle = s.ID\_Vehicle

AND v.ID\_Vehicle = c.ID\_Camio;

I en aquesta seleccionem els camions amb una matricula acabada en 79, i podem verificar que la query funciona a la perfecció. Aquesta és la mateixa que la principal, però fent un DISTICNT de la matricula així podem veure tots els camioners tot i que es diguin igual, i en la principal no, ja que no mostrem la matricula.

SELECT DISTINCT p.nom, c.matricula

FROM Persona AS p, Vehicle AS v, Camio AS c, Seguiment AS s, Localitzacio AS l, Comanda AS co

WHERE p.ID\_Persona = co.ID\_Client

AND s.ID\_Localitzacio = co.ID\_localitzacio

AND v.ID\_Vehicle = s.ID\_Vehicle

AND v.ID\_Vehicle = c.ID\_Camio

AND c.matricula LIKE '%79';

* + 1. ***Query 12***
       1. *Solution*

SELECT p.ID\_Patinet

FROM magatzem AS m, patinet AS p,vehicle AS v, localitzacio AS l,comanda AS c,seguiment AS s

WHERE p.ID\_Patinet = v.ID\_Vehicle

AND v.ID\_Magatzem = m.ID\_Magatzem

AND m.ID\_Localitzacio = l.ID\_Localitzacio

AND l.pais = 'Italy'

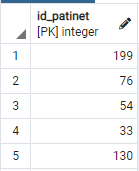
AND s.ID\_Vehicle = v.ID\_Vehicle

AND s.ID\_Localitzacio = c.ID\_Localitzacio

AND c.preu\_final > 1000

ORDER BY c.preu\_final DESC

LIMIT 5;



* + - 1. *Explanation*

En aquesta query mostrem el id dels 5 patinets que han entregat un comanda més cara i que estan ubicats en un magatzem Itàlia. Hem començat seleccionant les taules que necessitàvem, seguidament hem igualat els seus productes cartesians i finalment hem posat la condició que el país del magatzem sigui 'Italy' i que el preu final de la comanda sigui superior a 1000 tot i això ho hem ordenat perquè surtin en ordre de preu final més car. I hem posat límit a 5, ja que només ens demanaven 5.

* + - 1. *Query validation*

En aquesta query mostrem tots els patinets que han entregat una comanda i en la query a realitzar volem que només siguin d'Itàlia per aquesta raó tant el patinet 21 com el 24 no estan seleccionades i desprès ordenat pel preu\_final dels productes entregats com podem comprobar en aquesta query de validació per tant la nostra query queda ben validada.

SELECT p.ID\_Patinet,c.preu\_final,l.pais

FROM magatzem AS m, patinet AS p,vehicle AS v, localitzacio AS l,comanda AS c,seguiment AS s

WHERE p.ID\_Patinet = v.ID\_Vehicle

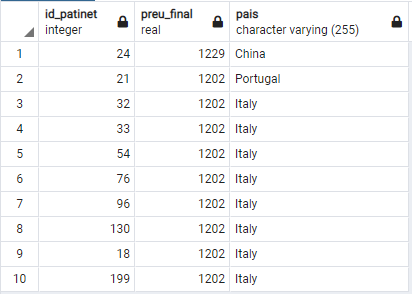
AND v.ID\_Magatzem = m.ID\_Magatzem

AND m.ID\_Localitzacio = l.ID\_Localitzacio

AND s.ID\_Vehicle = v.ID\_Vehicle

AND s.ID\_Localitzacio = c.ID\_Localitzacio

ORDER BY c.preu\_final DESC;



# Conclusions

## Use of resources

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stage** | **Marc** | **Joan** | **Narcís** | **Lluís** | **Total** |
| **ER model** | 4 | 2 | 2 | 5 | 13 |
| **Relational model** | 6 | 2 | 2 | 4 | 14 |
| **Physical model** | 8 | 4 | 15 | 20 | 47 |
| **DB population** | 2 | 2 | 5 | 8 | 17 |
| **DB validation** | 5 | 8 | 8 | 2 | 23 |
| **Documentation** | 7 | 2 | 2 | 3 | 14 |
| **Total:** | 32 | 20 | 34 | 42 | 128 |

La taula és una aproximació de les hores que hem empleat, ho hem intentat representar de la manera més adient, tot i així no creiem que sigui del tot correcte, ja que realment no hem calculat el temps que hi hem dedicat, el que podem garantir és que molt més del que teníem pensat.

**ER model**: el vam fer entre tots, la diferencia entre els membres és que els que tenen algo més són els que van ajuntar tots els models entre si, i deixar-ho tot polit.

**Relational model**: Exactament igual que el ER model.

**Physical model**: Com hem dit abans, ens vam dividir per igual la feina, però aquest apartat en concret és on han hagut més problemes i és on més hores hem empleat, els hi va tocar al Narcís i al Lluís, per això hi ha aquesta diferència de hores. També hem de dir que tot i que sembli exagerat, ho hem parlat i creiem que son més inclús...

**DB population**: Tota la importació de dades, crear tots els mockaroos i crear i importar les taules amb els csv que ens vau adjuntar.

**Documentació**: Aproximadament unes 13 hores, hi havia molta informació que afegir.

El total és exagerat, tot i que repetim, creiem que falten hores ,però hem de dir que moltes de les hores estan en conjunt, no per separat, per exemple quan el Narcís i el Lluís feien el model físic ho feien els dos junts i no per separat.

## Lessons learnt (1 page)

Per començar hem après a fer tots els models, a normalitzar-los i relacionar-los adequadament, i en certa manera, tot i que no es vegi gaire reflexat en la nostra pràctica, la manera d’endreçar-ho tot. A més a més, creiem que on més hem aprés és en la part del model físic, ja que en tot moment hem tingut problemes i ens hem hagut d’afanyar per solucionar-ho tot, que és el que ens pot passar en un futur laboral. Hem posat a prova tot el que hem treballat durant el curs, tant a classe com als labs de l’Avaluació Contínua, tot i que el concepte era el mateix, aquí ho hem portat a un nivell més elevat, ja que depenia únicament de la nostra feina. També hem après a utilitzar a la perfecció creadors de informació random web, en el nostre cas el Mockaroo, el 95% de les taules que hi ha en la nostra Base de Dades, han estat omplertes amb aquesta web.

## Future work and conclusions (1 page)

Fent la pràctica hem posat en prova tots els aspectes que hem anant aprenent al llarg de tot el primer semestre. No obstant, creiem que hagués estat molt millor que ens adjuntéssiu totes la taules que havíem de fer servir vosaltres plenes des del primer moment, ja que de la manera que està plantejada la pràctica hem hagut de crear molts csv randoms i després per fer que tot funcionés hem hagut de fer meravelles. En resum, creiem que la decisió de fer-nos crear tantes taules a part no ha estat encertada, i ja no només per nosaltres, ja que vosaltres a l’hora de corregir, cap grup tindrà el mateix resultat i haureu de mirar-ho pas a pas. I per la nostra part, creant tantes taules no hem fet servir cap dels

coneixements de l’assignatura, només hem aprés a fer servir el Mockaroo. Hem empleat més temps en fer csv que es fer les taules, i això no és el que es busca aprendre quan fem una pràctica d’aquesta assignatura. En resum, per futures pràctiques creiem que en comptes de donar-nos 5 taules fetes per vosaltres i nosaltres haver de fer 30, hauria de ser a la inversa, vosaltres ens doneu 30 i nosaltres fem

5. També creiem que la documentació que hem d’adjuntar en la memòria és exagerada, per exemple està bé que haguem d’explicar com hem plantejat i com funcionen els models, però tant específicament i tant densament ho trobem excessiu. Un altre exemple és que us haguem de raonar tots el tipus dels atributs, ja que potser en tenim 300. Al mateix temps volem agrair als becaris per la ajuda que ens han donat i la predisposició que han tingut en tot moment, més concretament a l’Anna, que sempre que l’hem escrit ha respòs de manera quasi immediata, fins el punt de fer vàries videotrucades si ho veia necessari, incloent les setmanes de vacances. En aquest aspecte estem molt agraïts amb tots el becaris, hem posat a l’Anna ja que és a la primera que vam contactar, estem segurs que tots els becaris ens haguessin ajudat igual que ella...