



Universitat Oberta  
de Catalunya

---

**Disseny i Implementació de la Base de  
Dades per a una Aplicació de Gestió de  
Processos Informàtics**

---

**Treball Final de Grau en Enginyeria Informàtica**

Autor: Enric Garcia Sirera

Àmbit: Bases de Dades

Tutor: Jordi Ferrer Duran

Responsable Àrea: Xavier Baró Solé

Gener de 2022



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## FITXA DEL TREBALL FINAL DE GRAU

Títol	Disseny i Implementació de la Base de Dades per a una Aplicació de Gestió de Processos Informàtics
Autor	Enric Garcia Sirera
Tutor	Jordi Ferrer Duran
Responsable d'Àrea	Xavier Baró Solé
Àrea	Bases de Dades
Idioma	Català
Data	Febrer 2022
Paraules Clau	Bases de Dades, SQL, postgresql, ITSM, database, BBDD, Data Warehouse

Resum
<p>El present treball pretén resoldre les necessitats d'automatització i organització del control dels processos i dispositius informàtics d'una empresa.</p> <p>A partir de l'anàlisi dels requisits i condicionants del sistema, es dissenya una base de dades relacional, es tria el gestor més adient disponible en el mercat, s'implementa la solució i es genera tota la documentació associada. Finalment, es sotmet la solució a una càrrega de dades i una bateria de proves per validar-ne el funcionament.</p> <p>El resultat final es presenta en forma de memòria on es recull la metodologia seguida i les decisions de disseny, un script de creació de la base de dades i de les restriccions i funcions derivades del disseny, un script de càrrega de dades, un joc de proves amb els resultats obtinguts i un guió per reproduir el sistema.</p> <p>La elaboració d'aquest treball ha permès aprofundir en els aspectes teòrics i pràctics del disseny d'una base de dades, entrar al detall en les possibilitats i limitacions d'un dels gestors més populars del mercat (postgresql) i, també, posar en pràctica alguns aspectes de la gestió de projectes en TI.</p>

Abstract
<p>The purpose of this end of degree project is to solve the needs of automation and organization of the IT processes and assets of a company.</p> <p>Given a set of requirements and conditions, we design a relational database capable of hosting the necessary information and provide the answers to the users request. Along the project, we choose the most suitable management system in the market, deploy the solution, and generate all the associated documents. To finalize, a significant amount of data is loaded and set of tests are run to validate the performance.</p>

The result comprehends this memory, detailing the methodology and the design decisions, the database creation script, a script to introduce the designed restrictions and functions, the data loading script, a set of tests and their results, and a guideline to install and reproduce the system.

Working on this project has helped understanding the theory and practice associated with data base design, has allowed the author to learn the possibilities and limitations of one of the leader products in the area (postgresql) and, also, has provided an opportunity to try some relevant aspects of project management in IT.

## ÍNDIX

Introducció .....	7
Descripció del Projecte.....	8
Objectius .....	8
Descomposició en Tasques .....	8
Anàlisi de Riscos .....	9
Planificació Temporal.....	10
Seguiment de la Planificació .....	11
Primera entrega. 4/10/21 .....	11
Segona entrega. 8/11/21 .....	11
Tercera entrega. 9/12/21 .....	12
Entrega final. 7/1/22 .....	12
Requisits .....	13
Disseny Conceptual .....	15
Entitats .....	15
Relacions .....	17
Diagrama E/R del model.....	18
Decisions de Disseny .....	20
Restriccions .....	20
Decisions associades al repositori estadístic.....	20
Funcionalitats Addicionals.....	20
Disseny Lògic .....	21
Normalització .....	23
Primera Forma Normal.....	23
Segona Forma Normal.....	23
Tercera Forma Normal .....	23
Forma Normal de Boyce-Codd .....	23
Quarta Forma Normal .....	23
Cinquena Forma Normal .....	23
Disseny Físic.....	24
Tria del SGBD.....	24
Aspectes derivats del SGBD triat.....	24
Procediments ABM.....	25
Processos Temporitzats .....	32
Consultes .....	32
Optimització .....	33

Carrega de Dades Bàsiques i Proves Inicials.....	34
Proves dels disparadors .....	34
alta d'usuari.....	34
alta de device_type i department.....	34
alta de order_line .....	34
modificació de order_line .....	34
alta de acció .....	35
alta de incident.....	36
alta de process_check .....	37
alta de inventory_check.....	37
Proves de Volum .....	37
Eficàcia del disseny Data Warehouse.....	40
Desenvolupament Futur .....	42
Conclusions .....	42
Glossari.....	43
Bibliografia .....	44
Annexos.....	45
Annex 1. Script de Creació de la Base de Dades .....	45
Annex 2. Script de Programació .....	45
Procediments ABM.....	45
Procediments Temporitzats .....	45
Annex 3. Scripts de Càrrega de Dades i Proves.....	45
Càrrega manual .....	45
Càrrega automàtica.....	45
Annex 4. Script de Consultes.....	45
Annex 5. Guia d'Instal·lació .....	46

## Introducció

La empresa Next World SA, ha decidit implementar una eina ITSM (IT Service Management) amb la intenció específica de portar una gestió d'incidències relacionades amb el sistema informàtic (Incident Management) i, també, la gestió dels equips (Assets Management).

El sistema de gestió d'incidències ha de permetre registrar els problemes o queixes dels usuaris, guardar-ne els detalls més importants i assignar-hi un tècnic que resolgui el problema.

Pel que fa a la gestió dels actius, el objectiu es controlar el material assignat a cada usuari, el cost de compra i de manteniment, les incidències i les baixes.

A més, la gestió dels equips ha de portar incorporada la informació financera necessària per saber si es una despesa o una inversió, en cas d'inversió a quin projecte està associada i quin nivell d'amortització acumula. També cal enregistrar les incidències associades a cada equip i usuari.

Finalment el Asset Management, ha de permetre realitzar dos tipus d'auditoria, la financera (requisit legal) i la de processos.

L'auditoria financera ha de validar la inversió en material de manera periòdica, fent un inventari físic que garanteixi que allò que està detallat en el sistema existeixi realment. Per cada actiu caldrà doncs guardar la data en que s'ha fet la darrera comprovació

L'auditoria de processos ha de mostrejar incidències i equips regularment per garantir que es compleixen les directrius de la empresa al respecte. El sistema ha de guardar la informació de les incidències o actius que s'hagin revisat i les conclusions.

L'abast d'aquest treball serà la d'analitzar el problema descrit i definir la estructura de Base de Dades necessària per donar suport a la eina ITSM. En queda fora de l'abast el desenvolupament de la eina en si.

## Descripció del Projecte

El projecte consisteix en recollir els requisits de la empresa, fer-ne l'anàlisi per si cal complementar-los i, a partir d'aquí, definir la estructura de la Base de Dades necessària per donar-hi solució.

Es seguiran les fases de disseny d'una Base de Dades: conceptual, lògic i físic. Implementant, provant i optimitzant aquest darrer sobre el SGBD que seleccionem.

## Objectius

- Identificar els requisits del sistema ITSM
- Fer el disseny conceptual amb UML o similar.
- Elaborar el disseny lògic a partir del conceptual
- Decidir el SGBD a utilitzar
- Generar les crides SQL necessaris per implementar el disseny físic sobre el SGBD.
- Implementar les consultes i/o procediments per cobrir tots els requisits del sistema dins el SGBD
- Generar un joc de dades suficient per poder provar la bondat del sistema.
- Dissenyar un joc de proves exhaustiu per validar el sistema.

## Descomposició en Tasques

Planificació	Definició de les tasques i planificació temporal.
Requisits	Recollida, anàlisi i definició dels requisits del sistema.
Disseny Conceptual	Elaboració del diagrama UML basat en el model relacional.
Disseny Lògic	Transformar el model conceptual i normalitzar.
Tria del SGBD	Definir sobre quin sistema treballarem
Disseny Físic	Definir les taules i columnes per implementar el sistema.
Programació	Programació dels scripts necessaris per la creació del sistema i per complir amb els requisits definits.
Inicialització de dades	Scripts per incorporar les dades de prova.
Proves i Optimització	Scripts i operacions necessaris per verificar el correcte funcionament i les possibles millores sobre el disseny.
Documentació	Al llarg del projecte caldrà anar preparant la documentació de cada fase.
Revisió	Revisió de tot el sistema i la seva documentació
Presentació	Entrega del projecte.
Tribunal	Defensa del projecte



## Anàlisi de Riscos

Al llarg de la elaboració del projecte sorgiran, inexorablement, diversitat de problemes i circumstàncies que poden complicar la consecució del mateix. A continuació tenim un detall d'aquests riscos amb les mesures correctores previstes per afrontar-los:

Risc	Mesures
<b>Problemes d'agenda:</b> impossibilitat de dedicar el temps previst al projecte davant d'altres obligacions.	La dedicació prevista en la planificació de de 4h diàries en dies laborables. Quan en algun període això no es pugui complir, caldrà treballar en dies festius per compensar. En el calendari tenim 43 dies festius per 67 de laborables i, per tant, s'hauria de poder aplicar aquest sistema sense problemes.
<b>Problemes amb el hardware:</b> instal·lacions, equips, connexions.	Disposar de dues instal·lacions alternatives, a la feina i al domicili particular per poder fer front a imprevistos tècnics.
<b>Problemes amb el programari:</b> mal funcionament de les eines.	En aquest cas, a part del tutor, es pot recórrer tant a la documentació en línia com a diferents fòrums tècnics on, normalment, hi ha informació sobre tota la casuística que ens podem trobar.
<b>Pèrdues de codi</b> per algun dels problemes anteriors.	Tot el codi, scripts i documentació que es vagi generant es guardarà en equips locals amb còpies al núvol.
<b>Problemes personals:</b> malalties, temes familiars, etc.	La dedicació prevista en la planificació de de 4h diàries en dies laborables. Quan en algun període això no es pugui complir, caldrà treballar en dies festius per compensar.



## Seguiment de la Planificació

El seguiment de la planificació inicial no ha estat perfecte, tal com era d'esperar, però es pot dir que ha anat prou bé. En els apartats següents hi ha detallada la situació de les tasques en cada una de les entregues, així com els problemes particulars trobats en cada moment.

Hi hagut endarreriments deguts, a la necessària corba d'aprenentatge en temes de programació, en concret els relacionats amb els procediments tem

### Primera entrega. 4/10/21

La primera entrega no va presentar problemes més enllà de fer les estimacions de temps que, sense la experiència corresponent, havien de ser per força un punt arbitraris.

Tasques vençudes segons la previsió	
tasca	situació
descripció del projecte	acabat
pla de treball	acabat

### Segona entrega. 8/11/21

Aquí estava concentrada la part més teòrica del projecte i, com que 'el paper ho aguanta tot', tampoc no hi va haver massa problemes en el compliment de les previsions. Tot i així, el fet de no haver arribat a fer el disseny físic, ja era un advertiment de que la pràctica, sovint, es força mes complexa que la teoria.

Tasques vençudes segons la previsió	
tasca	situació
descripció del projecte	acabat
pla de treball	acabat
disseny conceptual	acabat
disseny lògic	acabat
tria del SGBD	acabat
disseny físic	pendent

Tasques no vençudes segons la previsió	
tasca	situació
programació	pendent
càrrega de dades	pendent
proves i optimització	pendent
documentació	en procés
revisió	pendent
presentació	pendent

### Tercera entrega. 9/12/21

En aquesta entrega s'ha concentrat la feina de implementació i programació i ha estat aquí, precisament, on han aparegut els problemes. La instal·lació i programació del scheduler, la programació del triggers (tenint en compte que l'entorn de desenvolupament dins la BBDD no es ideal) i la gestió de les excepcions, han resultat força complicats i han requerit més temps del previst. A més, al dissenyar les funcions, ha calgut revisar detalls del disseny conceptual que no resultaven evidents a priori.

Tasques vençudes segons la previsió	
tasca	situació
descripció del projecte	acabat
pla de treball	acabat
disseny conceptual	acabat
disseny lògic	acabat
tria del SGBD	acabat
disseny físic	acabat
programació	en procés
càrrega de dades	pendent

Tasques no vençudes segons la previsió	
tasca	situació
proves i optimització	en procés
documentació	pendent
revisió	pendent
presentació	pendent

### Entrega final. 7/1/22

El plat fort aquí ha estat la elaboració de la documentació. Tot i que durant el projecte s'ha anat preparant la memòria. Calia ordenar-la, acabar-la i revisar-la.

S'ha dedicat força temps, no planificat, a la programació de les proves exhaustives per validar el funcionament del sistema amb volums 'grans' de dades.

Finalment, la producció del vídeo també ha resultat més feixuga del esperat.

Tasques vençudes segons la previsió	
tasca	situació
totes	acabat

Tasques no vençudes segons la previsió	
tasca	situació
presentació	pendent

## Requisits

La primera fase en el disseny d'una base de dades es la recollida de requisits. Es la etapa en la que cal una relació més activa amb els stakeholders (persones o entitats amb algun interès o relació amb el producte final), ja que cal definir les seves necessitats i expectatives sobre el sistema.

Es el període més important en el desenvolupament, ja que qualsevol malentès u omisió ens pot portar a productes no adequats que caldrà redissenyar a posteriori amb el consegüent cost en temps i en diners.

En el nostre cas, però, es una etapa senzilla, ja que els requisits estan expressats en el enunciat i, per tant, prendrem aquest document com a punt de partida de la nostra solució.

Tal com s'explica al manual [1], agruparem els requisits segons siguin de procés o de producte i, aquests darrers, els subdividirem entre funcionals i no funcionals. Prendrem com a referencia la plantilla Volere[2].

Requisits de Procés	
codi	descripció
RP01	El producte s'ha de lliurar el 7/1/2022
RP02	Cal fer una presentació del producte entre el 10 i el 14 de Gener del 2022
Requisits de Producte. No Funcionals	
codi	descripció
RNF01	La base de dades serà relacional
RNF02	S'utilitzarà el llenguatge SQL
RNF03	Tota la gestió i accés a la informació es farà mitjançant procediments de BD
RNF04	es farà un diagrama E/R per representar el model conceptual de la BD
RNF05	La BD ha de ser escalable
RNF06	Cal mantenir un log de les accions sobre la BD amb el resultat OK/ERROR
RNF07	Les consultes estadístiques específiques definides en els requisits funcionals, s'hauran de poder respondre en temps 1.

[1] Pradel Miquel, J., Raya Martos (2021), J. *Introducció a l'enginyeria de requisits*. UOC

[2] Robertson, J.; Robertson, S. *Volere Requirements Specification Template*.  
<https://www.volere.org/templates/volere-requirements-specification-template/> . Access  
desembre 2021

## Requisits de Producte. Funcionals

codi	descripció
RF01	Enmagatzemar les dades financeres corresponents a cada comanda i si es inversió o despesa
RF02	Registrar la recepció del material i controlar que l'encarregat es tècnic informàtic.
RF03	Registrar les accions (assignació, alta, baixa etc) sobre el material. Controlant que hi ha un tècnic responsable i registrant la data, el dispositiu i l'usuari afectats.
RF04	Registrar les incidències dels dispositius
RF05	Mantenir un registre de cada auditoria financera, guardant la llista dels dispositius verificats i la data de la validació
RF06	Mantenir un registre de cada auditoria de processos, guardant la llista de les incidències verificades i la data de la validació
RF07	Enmagatzemar la informació dels dispositius informàtics
RF08	Enmagatzemar la informació dels departaments de l'empresa
RF09	Enmagatzemar la informació dels usuaris
RF10	Enmagatzemar la informació dels proveïdors
RF11	Mantenir un repositori estadístic per donar resposta ràpida a les preguntes detallades en els sub-requisits següents
RF11-01	Departament amb un percentatge més alt d'incidències en el darrer any (actualitzat cada mes) en relació al nombre d'usuaris del departament
RF11-02	Número de dispositius comprats en un any determinat
RF11-03	Percentatge de portàtils que no han sofert cap averia des de la assignació a un usuari
RF11-04	Comandes lligades a una ordre d'inversió determinada en un any concret
RF11-05	Proveïdor amb un temps de lliurament més curt
RF11-06	Número de dispositius de cada tipus per un departament concret
RF11-07	Valor econòmic dels dispositius actius assignats a un departament
RF11-08	Número de dispositius amb data d'inventari físic superior a un any
RF11-09	Usuari que, en el any en curs, ha creat més incidències
RF11-10	Departament que, en un any concret, ha fet més reassignacions.
RF11-11	Percentatge d'usuaris sense dispositius assignats.
RF11-12	Per una auditoria financera concreta, registre dels dispositius auditats.
RF11-13	Per una auditoria de processos concreta, registre de les incidències auditades.
RF11-14	Departament amb més tipus de dispositius diferents i quants tipus són.
RF11-15	Número d'usuaris amb més de 5 dispositius assignats
RF11-16	Top10 d'usuaris per nombre d'incidències

## Disseny Conceptual

La fase de disseny conceptual consisteix a modelar els requisits en forma esquemàtica utilitzant diagrames UML (Unified Modeling Language) o ER (Entity Relation). En el nostre cas utilitzarem un diagrama ER expressant la cardinalitat amb números i detallant les claus principals i foranes en el mateix esquema.

Hi ha diferents eines al mercat que permeten fer aquest modelatge, tant de comercials (com MagicDraw), com de gratuïtes (com diagrams.net, abans draw.io) i, fins hi tot, alguns SGBD disposen d'eines pròpies. En aquest cas hem triat diagrams.net (<https://www.diagrams.net/>) perquè, sent de llicència lliure, es una eina prou sòlida, versàtil i còmode per un projecte d'aquesta complexitat.

### Entitats

La següent relació detalla les entitats que hem identificat com a necessàries per satisfer els requisits. Per evitar que la extensió d'aquesta llista sigui innecessàriament llarga, per a cada entitat detallarem, només, els atributs essencials pel funcionament del sistema.

D'altra banda, dividirem entitats i atributs entre aquells relacionats, estrictament, amb la informació del sistema i els que farem servir per garantir la funcionalitat de Datawarehouse expressada en el requisit RF11.

Entitats i Atributs relacionats amb Informació		
nom	descripció	atributs essencials
<b>usuaris</b>	informació dels usuaris	id usuari id departament
<b>departaments</b>	informació dels departaments	id departament
<b>comandes actius</b>	informació de les comandes relacionades amb la compra d'actius	id comanda actiu
<b>comandes</b>	informació de les comandes	id comanda id comanda actiu id proveïdor data
<b>linies de comanda</b>	informació de cada línia d'una comanda	id comanda id línia id tipus dispositiu id dispositiu preu data rebut
<b>proveïdors</b>	informació dels proveïdors	id proveïdor
<b>dispositius</b>	informació dels dispositius	id dispositiu id tipus dispositiu
<b>tipus dispositius</b>	enumeració dels tipus de dispositius	id tipus dispositiu

<b>accions</b>	informació de les accions sobre els dispositius	id dispositiu id usuari id tècnic id tipus acció data
<b>tipus acció</b>	enumeració dels tipus d'acció	id tipus acció
<b>incidències</b>	registre d'incidències dels dispositius	id incidència id dispositiu id tècnic data
<b>auditoria financera</b>	informació sobre les auditories financeres (físiques)	id auditoria financera
<b>control inventari</b>	informació sobre els controls d'inventari de cada auditoria financera	id auditoria financera id dispositiu data
<b>auditoria processos</b>	informació sobre les auditories de processos (incidències)	id auditoria proces
<b>control proces</b>	informació sobre els controls de cada auditoria de processos	id auditoria proces id incidència data
<b>assignacions</b>	dispositius assignats a cada usuari	id dispositiu id usuari

## Entitats i Atributs relacionats amb Datawarehouse

nom	descripció	atributs essencials
<b>usuaris</b>	informació dels usuaris	número d'incidències incidències any en curs número de dispositius
<b>departaments</b>	informació dels departaments	número d'usuaris incidències darrer any valor inventari assignacions del any
<b>dispositius per tipus</b>	dispositius per tipus i departament	id departament id tipus dispositiu unitats
<b>dispositius</b>	informació dels dispositius	darrer inventari id status
<b>status</b>	enumeració dels possibles status dels dispositius	id status
<b>auditoria financera</b>	informació sobre les auditories financeres (físiques)	número de controls
<b>auditoria processos</b>	informació sobre les auditories de processos (incidències)	número de controls



<b>inversions anuals comanda inventari</b>	total invertit en un any per cada comanda d'inversió	id comanda inversio any total invertit
<b>estadístiques generals</b>	registre d'estadístiques generals del sistema	número usuaris número dispositius portatils sense incidència número de portàtils dispositius inventari vell usuaris sense dispositius usuaris més de 5 dispst. departament pitjor rati departament més tipus número de tipus temps d'entrega més curt proveïdor més rapid usuari amb mes incidents id laptop id departament IT
<b>estadístiques anuals</b>	registre d'estadístiques generals del sistema per any	any dispositius comprats departament més assignacions. Máximes assignacions
<b>log</b>	registre dels procediments sobre la BBDD	Id log data entrades RSP

## Relacions

Les relacions entre els entitats son les descrites a continuació:

### Relacions (sense exclusivament estadístiques)

entitat 1	entitat 2	tipus	descripcio
usuaris	departaments	<b>N a 1</b>	pertany a
departaments	usuaris	<b>1 a 0..1</b>	cap de
departaments	dispositius per tipus	<b>1 a N</b>	té
departaments	presupost	<b>1 a N</b>	té
comandes	comandes actius	<b>N a 1</b>	pertany a
comandes	proveïdors	<b>N a 1</b>	feta a
linies de comanda	comandes	<b>N a 1</b>	pertany a
linies de comanda	tipus dispositiu	<b>N a 1</b>	es de tipus

linies de comanda	dipositiu	<b>1 a 0..1</b>	correspon a
dipositiu	accio	<b>1 a N</b>	l'hi correspon
dipositiu	control inventari	<b>0..1 a N</b>	l'hi correspon
dipositiu	incident	<b>0..1 a N</b>	relacionat amb
dipositiu	tipus dipositiu	<b>N a 1</b>	es de tipus
dipositiu	status	<b>N a 1</b>	estat actual es
dipositiu	assignacions	<b>1 a 0..1</b>	assignat a
assignacions	usuari	<b>N a 1</b>	assignat a
accio	tipus accio	<b>N a 1</b>	es de tipus
accio	usuari	<b>N a 1</b>	relacionat amb
accio	usuari	<b>N a 1</b>	qui es el responsable
incident	usuari	<b>N a 1</b>	qui es el responsable
incident	control proces	<b>1 a N</b>	l'hi correspon
control proces	auditoria proces	<b>N a 1</b>	pertany a
control inventari	auditoria financera	<b>N a 1</b>	pertany a

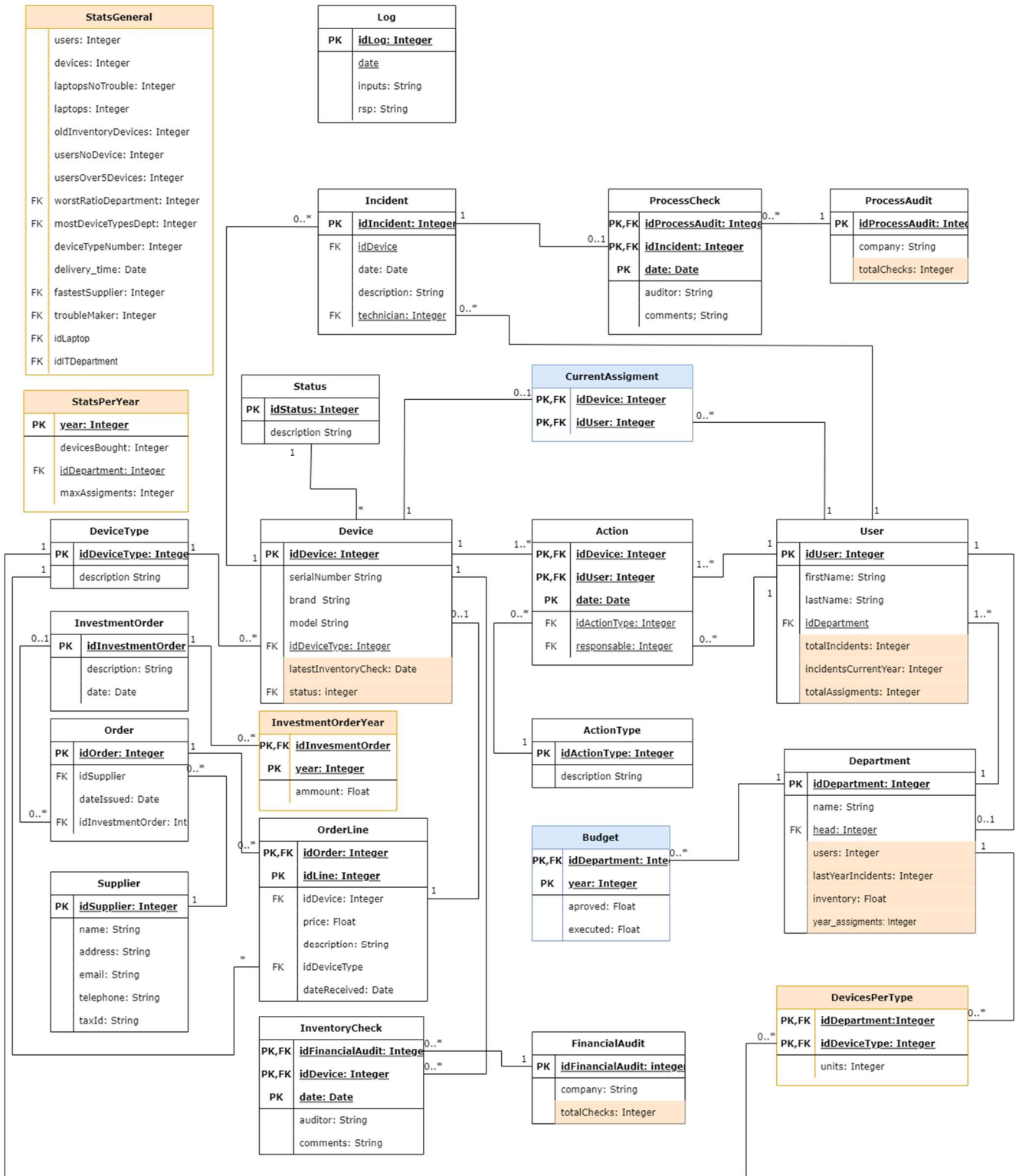
### Diagrama E/R del model

Per tal que el diagrama sigui el mes detallat possible respecte a la aplicació final del sistema hem adoptat les següents premisses:

- S'indicaran les claus primàries amb PK i, si son alhora claus foranes: PK,FK
- En el cas de claus compostes l'ordre serà de dalt a baix.
- El subratllat indica que un atribut no pot ser NULL
- El color taronja indica que una taula o atribut, son redundants i s'han incorporat per donar respostes ràpides a les consultes descrites en els requisits.
- El color blau indica taules associades a funcionalitats NO específiques dels requisits però que hem considerat interessants.

Hem començat per definir les taules principals, després les derivades de relacions i, finalment hem afegit les necessàries per poder donar resposta en temps 1 a les consultes detallades al enunciat.

El diagrama resultant el trobem a la pàgina següent



## Decisions de Disseny

Tant les taules principals, com les de relació i les cardinalitats queden clarament detallades al diagrama. Tanmateix, les taules i atributs destinats a respondre les consultes en temps 1 i algunes altres que hem incorporat, així com algunes restriccions, mereixen comentari a part.

### Restriccions

Els dispositius s'assignaran a usuaris individuals. En el cas de que un dispositiu s'utilitzi per diversos usuaris d'un departament, caldrà assignar-lo al cap del mateix.

Només els usuaris del departament d'informàtica poden ser responsables de les assignacions o els tècnics que solucionen les incidències.

S'ha considerat que la majoria de taules no admeten baixes per tal de mantenir la informació històrica disponible. Per exemple, si donéssim de baixa (de la taula) un dispositiu, caldria eliminar el seu registre d'incidències o les anotacions en les auditories que son informacions que poden ser útils en el futur. Òbviament es pot deixar un dispositiu de 'baixa' (amb l'atribut status) o un usuari sense departament ni dispositius, però en ambdós casos mantenint la informació històrica.

La contrapartida al punt anterior es que, al llarg del temps, es pot acumular molta informació 'desactivada' dins la base de dades. Per evitar això, caldrà decidir quant de temps es vol conservar aquesta informació i, periòdicament, fer buidats de la que hagi quedat obsoleta.

### Decisions associades al repositori estadístic

Com que ens demanen que el sistema pugui respondre de manera ràpida a un seguit de consultes amb independència del volum de dades, caldrà mantenir alguns atributs o taules addicionals que, malgrat ser redundants (perquè es podrien calcular sobre els propis registres de la BBDD) ens permetran complir amb aquest requisit.

Tots els atributs relacionats amb el repositori estadístic s'actualitzaran o calcularan via processos interns de la base de dades. A l'apartat on es descriuen els ABM es pot veure com i en quin moment s'actualitza cada un d'aquests atributs.

### Funcionalitats Addicionals

Hem afegit una taula *Budget* per guardar els pressupostos (en TI) aprovats anuals de cada departament y la quantitat executada. Això permetrà, per exemple, saber si un departament té o no pressupost disponible alhora de autoritzar una compra i, també, comprovar si les partides assignades s'acaben utilitzant o no.

Hem afegit la taula *CurrentAssignment* per poder consultar amb més facilitat a quin usuari està assignat cada dispositiu i quins dispositius té assignats un usuari.

Hem afegit un atribut *status* a la taula *Device* per poder consultar directament si està pendent d'alta, no assignat, assignat o de baixa sense haver de recórrer a cerques sobre *Acció*.

A *generalStats* guardarem l'identificador del departament tècnic i el del tipus laptop.

## Disseny Lògic

A continuació es detallen les entitats, relacions i atributs normalitzades per la posterior implantació en el SGBD.

El nom de cada relació, començant en majúscula, encapçala cada línia, seguit dels atributs entre parèntesi.

Les claus primàries estat subratllades en continu (també si son claus foranes), i les foranes amb punts.

Device(idDevice, serialNumber, brand, model, idDeviceType, lastInventoryCheck, status)

On idDeviceType referencia DeviceType (idDeviceType)

On status referencia Status(idStatus)

DeviceType(idDeviceType, description)

User (idUser, firstName, LastName, idDepartment, totalIncidents, incidentsCurrentYear, totalAssignments)

On idDepartment referencia Department (idDepartment)

CurrentAssignment(idDevice, idUser)

On idDevice referencia Device(idDevice)

On idUser referencia User(idUser)

Department (idDepartment, name, head, users, lastYearIncidents, inventory)

On head referencia Users (idUser)

Budget (idDepartment, year, aproved, executed)

On idDepartment referencia Department (idDepartment)

DevicesPerType (idDepartmentType, idDeviceType, units)

On idDepartment referencia Department(idDepartment)

On idDeviceType referencia DeviceType(idDeviceType)

Action (idDevice, idUser, date, idActionType, responsible)

On idDevice referencia Device(idDevice)

On idUser referencia User(idUser)

On idActionType referencia ActionType (idActionType)

On responsible referencia User(idUser)

ActionType (idActionType, description)

Incident (idIncident, idDevice, technician, date, description)

On idDevice referencia Device(idDevice)

On technician referencia User(idUser)

InvestmentOrder (idInvestmentOrder, description, date)

Order (idOrder, idSupplier, idInvestmentOrder, dateIssued)

On idSupplier referencia Supplier (idSupplier)

On idInvestmentOrder referencia InvestmentOrder (idInvestmentOrder)

OrderLine (idOrder,idLine, idDevice, description, price, dateReceived)

On idOrder referencia Order (idOrder)

On idDevice referencia Device(idDevice)

InvestmentOrderYear (idInvestmentOrder, year, ammount)

On idInvestmentOrder referencia InvestmentOrder (idInvestmentOrder)

Supplier (idSupplier, name, address, email, telephone, taxId)

FinancialAudit (idFinacialAudit, company, totalChecks)

InventoryCheck(idDevice, idFinacialAudit,date, auditor, comments)

On idDevice referencia Device(idDevice)

On idFinacialAudit referencia FinancialAudit (idFinacialAudit)

ProcessAudit (idProcessAudit, company, totalChecks)

ProcessCheck (idIncident, idProcessAudit, date, auditor, comments)

On idIncident referencia Incident (idIncident)

On idProcessAudit referencia ProcessAudit (idProcessAudit)

StatsGeneral (id, users, devices, laptopsNoTrouble, oldInventoryDevices, usersNoDevice, usersOver5Devices, worstRatioDepartment, mostDeviceTypesDept, deviceTypeNumber, deliveryTime, fastestSupplier, troubleMaker, idLaptop, idITDepartment)

On worstRatioDeparment referencia Department(idDepartment)

On fastestSupplier referencia Supplier(idSupplier)

On troubleMaker referencia User(idUser)

On idLaptop referencia DeviceType(idDeviceType)

On idITDeparment referencia Department(idDepartment)

StatsPerYear (year, devicesBought, idDepartment, maxAssignments)

On idDepartment referencia Department (idDepartment)

## Normalització

Cal comprovar que el disseny està normalitzat i, per fer-ho, repassarem les condicions de cada forma normal.

### Primera Forma Normal

Els atributs han de ser atòmics. Es compleix en tots els casos

### Segona Forma Normal

Cal que tot atribut, que no formi part d'una clau candidata, depengui de totes les claus candidates de la relació. En el cas de claus formades per un sol atribut el compliment es automàtic. Per les claus formades per més d'un atribut, donat que en el disseny sempre corresponen a claus foranes o a una data concreta, també podem afirmar el compliment.

### Tercera Forma Normal

No tenim atributs que depenguin d'altres que no formin part de claus candidates. Es compleix.

### Forma Normal de Boyce-Codd

Com estem en 3FN i no tenim claus candidates compostes, que no siguin les pròpies claus. Podem assegurar que estem en FNBC.

### Quarta Forma Normal

Donat que no tenim atributs multivaluats i estem en FNBC, estarem en 4FN

### Cinquena Forma Normal

El nostre disseny no presenta dependències de projecció-combinació i està en 4FN, per tant està en 5FN.

Podem afirmar, doncs, que el nostre model està normalitzat.

## Disseny Físic

Aquest es el punt on cal implementar el model lògic dissenyat anteriorment sobre un suport físic. En el nostre cas, aquest suport serà un sistema gestor de bases de dades relacionals que ens ha de permetre emmagatzemar la informació del sistema, amb les seves funcions addicionals, i garantir-ne la integritat i disponibilitat.

### Tria del SGBD

Per començar s'han tingut en compte els requisits directament expressats pel client: cal emprar el llenguatge SQL, incloure tota la gestió mitjançant procediments de BD i, finalment, garantir que la aplicació ha de servir per a qualsevol volum de dades. Tanmateix es lògic afegir que ha de permetre una instal·lació remota amb accés concurrent de diferents usuaris.

Hi ha una oferta variada de gestors de bases de dades al mercat (Oracle, SQL Server, PostgreSQL), i no hi ha gaire diferència en les prestacions dels líders de mercat excepte en una cosa: el tipus de llicència. Tant Oracle com SQL Server són productes comercials que, en funció del número de llicències i prestacions que necessitem poden arribar a ser força cars. D'altra banda, PostgreSQL és Open Source i, per tant, gratuït.

Des de el punt de vista de les prestacions hi ha petits avantatges i inconvenients en cada uns dels sistemes, però el fet de ser un producte open source ja consolidat i contrastat, ens ha fet decantar per PostgreSQL ja que, a part del aspecte econòmic, ve acompanyat d'una comunitat de desenvolupadors i usuaris que n'estan accelerant les millores.

### Aspectes derivats del SGBD triat

La versió de PostgreSQL sobre la que s'ha fet el desenvolupament és la 14.1 instal·lada en local.

S'ha utilitzat la aplicació PgAdmin 4.6 per gestionar la creació de la base de dades, la programació de les funcions, la inserció de les dades i la execució de les proves, d'una manera més còmode que via consola.

S'ha instal·lat la extensió pgAgent que permet programar la execució automàtica de procediments en moments concrets (scheduler).

S'ha fet tot el desenvolupament sobre la base de dades per defecte (postgres), l'esquema públic, i l'usuari per defecte (postgres) per facilitar-ne la instal·lació posterior en el moment de la avaluació.

S'han modificat els noms de les taules i els atributs per excloure les majúscules, ja que el tractament d'aquestes complica de manera innecessària les sentències SQL.

També s'han canviat el nom de les taules *order* i *user*, per *orderx* i *usuari* respectivament, perquè són paraules reservades del SGBD.

Pel que fa als identificadors únics, si bé en un entorn real caldria decidir si en tenim prou amb enters de 2, 4 o 8 bytes o bé ens cal recórrer a uuid (identificador únic universal), pel nostre projecte utilitzarem el tipus serial (enters de 4 bytes auto incrementats).



## Procediments ABM

Els procediments ABM (alta, baixa i modificació) sobre cada una de les taules son els que ens permetran, convenientment estructurats, garantir el compliment dels requisits del sistema tant pel que fa a la integritat com a la funcionalitat.

Tanmateix, moltes de les taules no presenten complicacions específiques relacionades amb els ABM, perquè no afecten significativament a altres taules (tret de les baixes de registres relacionats que, simplement, no es permet) ni, tampoc, tenen conseqüències en les estadístiques o registres generals que hem dissenyat. Per tant, n'obviarem el detalls ja que consistiran a simples INSERTS o UPDATES.

Pel que fa a la resta, en detallarem a continuació els aspectes a tenir en compte en les ABM que ho requereixin així com el pseudocodi per il·lustrar el script. Aquests detalls seran la guia per la elaboració dels procediments que incorporem en el annex corresponent.

Per tal de no alentir el sistema, tenint en compte que cal que la BBDD sigui capaç de gestionar grans volums de dades, no s'han utilitzat funcions d'agregats en la programació dels procediments d'ABM.

order_line	
procediment	actuació
alta	comprovar el número de la darrera línia existent a la comanda i incrementar en una unitat per assignar-li al <i>id_line</i> de la nova (si no hi ha línies, assignar 1)
pseudocodi	BEFORE INSERT guardar paràmetres d'entrada llegir el numero de línia de la darrera línia de la comanda corresponent a la <i>order_line</i> . Si no hi a línies torna 0. incrementar el numero de línia i assignar-lo a la nova <i>order_line</i> escriure registre al log amb OK si alguna cosa falla: registrar l'error al log
modificació	donar d'alta el <i>device</i> amb status 'pendent' i relacionar-lo amb la <i>order_line</i> Incrementar <i>devices a general_stats</i> Incrementar <i>laptops a general_stats</i> si es un portatil Comprovar si el temps des de <i>date_issued</i> de <i>orderx</i> i el <i>date_received</i> es més petit que <i>delivery_time de general_stat</i> , si ho es, canviar <i>delivery_time i fastest_supplier</i> Si la <i>orderx</i> corresponent a <i>id_order</i> té una <i>id_investment_order</i> , incrementar <i>ammount</i> amb <i>price a investment_order_year</i> per l'any en curs

pseudocodi	<p>BEFORE UPDATE</p> <p>guardar paràmetres d'entrada</p> <p>inserta un nou device i assigna el seu id al id_device de la order_line.</p> <p>Llegir devices de general_stats</p> <p>incrementar el valor</p> <p>actualitzar general_stats amb el nou devices</p> <p>si el dispositiu es un portatil:</p> <p>    llegir laptops de general_stats</p> <p>    incrementar el valor llegit</p> <p>    actualitzar general_stats amb el nou valor</p> <p>llegir l'any de la data de la recepció</p> <p>de year_stats llegir devices_bought de l'any corresponent</p> <p>incrementar el valor llegit</p> <p>actualitzar year_stats amb el nou valor</p> <p>llegir date_issued de la comanda corresponent a la order_line</p> <p>calcular el temps que ha passat fins ara</p> <p>llegir delivery_time de general_stats</p> <p>comparar els dos temps</p> <p>si nou temps es mes petit que el que teniem:</p> <p>    llegir el supplier de la comanda corresponent a la order_line</p> <p>    actualitzar general_stats amb el nou fastest_supplier i delivery_time</p> <p>llegir el id de la investment_order lligada a la comanda</p> <p>si existeix:</p> <p>    comprova si existeix registre a investment_order_year per la comanda i l'any corresponent</p> <p>    si existeix:</p> <p>        llegeix ammount del registre.</p> <p>        incrementa el valor amb price.</p> <p>        actualitza el registre a investment_order_year</p> <p>    sino</p> <p>        el crea amb ammount = price.</p> <p>    fisi</p> <p>escriu el registre al log amb OK.</p> <p>si alguna cosa falla:</p> <p>    registrar l'error al log</p>
------------	--

incident	
procediment	actuació
alta	<p>Verifica que el tècnic es del departament de IT</p> <p>Incrementar <i>total_incidents</i> i <i>incidents_current_year</i> al usuari</p> <p>Actualitzar <i>trouble_maker a general_stats</i> si cal</p> <p>Decrementar <i>laptops_no_trouble a general_stats</i> si es un portatil</p>
pseudocodi	<p>AFTER INSERT</p> <p>guardar parametres d'entrada</p> <p>llegir el <i>id_department</i> del responsable</p> <p>llegir el <i>IT_department</i> de <i>general_stats</i></p> <p>si no son iguals llença un error</p> <p>llegir els <i>total_incidents</i>, <i>incidents_current_year</i> i el id del usuari que te el dispositiu assignat.</p> <p>Incrementar <i>total_incidents</i> i <i>incidents_current_year</i> i actualitzar l'usuari</p> <p>llegir els <i>total_incidents</i> del usuari <i>trouble_maker</i> (de <i>general_stats</i>)</p> <p>si l'usuari que té el dispositiu assignat supera els <i>total_incidents</i> del <i>trouble_maker</i>, actualitza <i>trouble_maker</i> amb el nou usuari.</p> <p>llegir el <i>device_type</i> del dispositiu</p> <p>si el <i>device_type</i> es un laptop:</p> <p>    llegir el <i>laptops_no_trouble</i> de <i>general_stats</i></p> <p>    decrementar el valor llegit i actualitzar <i>general_stats</i></p> <p>escriure registre al log</p> <p>si alguna cosa falla:</p> <p>    registrar l'error al log</p>

inventory_check	
procediment	actuació
alta	<p>Incrementar <i>total_checks</i> a <i>finacial_audit</i></p> <p>Actualitzar <i>latest_inventory_check a device</i></p>
pseudocodi	<p>AFTER INSERT</p> <p>guardar paràmetres d'entrada</p> <p>llegir els <i>total_checks</i> de la auditoria corresponent al dispositiu</p> <p>incrementar el valor llegit i actualitzar el registre de la auditoria</p> <p>actualitzar <i>last_inventory_check</i> al dispositiu amb la data actual</p> <p>escriure registre al log amb OK</p> <p>si alguna cosa falla:</p> <p>    registrar l'error al log</p>

<b>process_check</b>	
<b>procediment</b>	<b>actuació</b>
<b>alta</b>	Incrementar <i>total_checks</i> a <i>process_audit</i>
pseudocodi	AFTER INSERT guardar paràmetres d'entrada llegir els <i>total_checks</i> de la auditoria corresponent al procediment incrementar el valor llegit i actualitzar el registre de la auditoria escriure registre al log amb OK si alguna cosa falla: registrat l'error al log

<b>department</b>	
<b>procediment</b>	<b>actuació</b>
<b>alta</b>	Crear un registre a <i>devices_per_type</i> per cada <i>device_type</i>
pseudocodi	AFTER INSERT guardar paràmetres d'entrada si no existeix crear un registre a <i>devices_per_type</i> per amb el id del departament per cada <i>device_type</i> escriure registre al log amb OK si alguna cosa falla: registrat l'error al log

<b>device_type</b>	
<b>procediment</b>	<b>actuació</b>
<b>alta</b>	Crear un registre a <i>devices_per_type</i> per cada <i>department</i>
pseudocodi	AFTER INSERT guardar paràmetres d'entrada si no existeix crear un registre a <i>devices_per_type</i> per amb el id del <i>device_type</i> per a cada <i>department</i> escriure registre al log amb OK si alguna cosa falla: registrat l'error al log

accio	
procediment	actuació
alta	verificar que el responsable pertany al departament IT
pseudocodi	<p>BEFORE INSERT</p> <p>guardar parametres d'entrada</p> <p>llegir el id_department del responsable</p> <p>llegir el IT_department de general_stats</p> <p>si no son iguals llença un error</p> <p>Executar el codi corresponent al tipus d'accio</p> <p>escriure registre al log amb OK</p> <p>si alguna cosa falla:</p> <p style="padding-left: 20px;">registrar l'error al log</p>
	en funció del tipus d'acció cal:
alta	canviar <i>status</i> al <i>device</i> a <i>no-assignat</i>
pseudocodi	actualitzar l' <i>status</i> al <i>device</i> corresponent a la acció
assignació	<p>Posar <i>status no-assignat</i> al <i>device</i> a <i>assignat</i></p> <p>Afegir registre a <i>current_assignment</i></p> <p>Incrementar <i>total_assignments</i> al usuari</p> <p>Incrementar <i>users_over5</i> a <i>general_stats</i> si cal</p> <p>Decrementar <i>users_no_device</i> a <i>general_stats</i> si cal</p> <p>Incrementar <i>laptops_no_trouble</i> a <i>general_stats</i> si es un laptop</p> <p>Incrementar <i>Inventory</i> al departament del usuari</p> <p>Incrementar <i>units</i> a <i>devices_per_type</i> del <i>department</i> i <i>device_type</i></p> <p>Actualitzar <i>max_assignments</i> i <i>id_department</i> a <i>year_stats</i> si cal</p> <p>Actualitzar <i>most_devices_types_dept</i> i <i>device_type_number</i> si cal</p>
pseudocodi	<p><u>posar status assignat al device a assignat</u></p> <p><u>insertar un registre a current_assignment amb el usuari i dispositiu</u></p> <p>llegir <i>total_assignments</i> del usuari corresponent de <i>user</i></p> <p>incrementar el valor llegit</p> <p><u>actualitzar l'usuari amb el nou valor</u></p> <p>si el nou valor es 6:</p> <p style="padding-left: 20px;">llegir <i>users_over5_devices</i> de <i>general_stats</i></p> <p style="padding-left: 20px;">incrementar el valor llegit</p> <p><u>actualitzar general_stats</u></p> <p>si el nou valor es 1:</p> <p style="padding-left: 20px;">llegir <i>users_no_device</i> de <i>general_stats</i></p> <p style="padding-left: 20px;">decrementar el valor</p> <p><u>actualitzar general_stats</u></p> <p>llegir el <i>device_type</i> del <i>device</i> corresponent</p> <p>si es un laptop:</p> <p style="padding-left: 20px;">llegir <i>laptops_no_trouble</i> from <i>general_stats</i></p> <p style="padding-left: 20px;">incrementar el valor llegit</p>

	<p><u>actualitzar general_stats</u>  llegir el id_department del usuari corresponent  llegir el inventory del departament  llegir el price de la order_line corresponent al dispositiu  incrementar inventory amb price  <u>actualitzar el departament amb el nou inventory</u>  llegir units de devices_per_type del departament i tipus corresponents  incrementar units  <u>actualitzar devices_per_type</u>  llegir device_type_number de general_stats  contar quants registres hi ha a devices_per_type del departament  si hi ha mes registres que device_type_number:  actualitzar general_stats amb nou device_type_number i  most_devices_types_dept nous</p>
alliberament	<p>Canviar <i>status</i> al <i>device</i> a <i>no-assignat</i>  Eliminar registre a <i>current_assignment</i>  Decrementar <i>total_assignments</i> al usuari  Decrementar <i>users_over5</i> a <i>general_stats</i> si cal  Incrementar <i>users_no_device</i> a <i>general_stats</i> si cal  Decrementar <i>laptops_no_trouble</i> a <i>general_stats</i> si es un laptop  Decrementar <i>Inventory</i> al departament del usuari  Decrementar <i>units</i> a <i>devices_per_type</i> del <i>department</i> i <i>device_type</i>  Actualitzar <i>max_assignments</i> i <i>id_department</i> a <i>year_stats</i> si cal  Actualitzar <i>most_devices_types_dept</i> i <i>device_type_number</i> si cal</p>
pseudocodi	<p><u>posar status assignat al device a no-assignat</u>  seguir amb el codi de baixa</p>
baixa	<p>Canviar <i>status</i> al <i>device</i> a <i>baixa</i>  si el old <i>status</i> NO era <i>pendent</i>:  Decrementar <i>devices</i> a <i>general_stats</i>  Decrementar <i>laptops</i> a <i>general_stats</i> si es un portatil  si el old <i>status</i> era <i>assignat</i> fer el mateix que amb <i>alliberament</i></p>
pseudocodi	<p><u>eliminar el registre a current_assignment corresponent al dispositiu</u>  llegir <i>total_assignments</i> del usuari corresponent de user  decrementar el valor llegit  <u>actualitzar l'usuari amb el nou valor</u>  si el nou valor es 5:  llegir <i>users_over5_devices</i> de <i>general_stats</i>  decrementar el valor llegit  <u>actualitzar general_stats</u>  si el nou valor es 0:  llegir <i>users_no_device</i> de <i>general_stats</i>  incrementar el valor</p>

	<p><u>actualitzar general_stats</u>  llegir el device_type del device corresponent  si es un laptop:  llegir laptops_no_trouble from general_stats  decrementar el valor llegit  <u>actualitzar general_stats</u>  llegir el id_department del usuari corresponent  llegir el inventory del departament  llegir el price de la order_line corresponent al dispositiu  decrementar inventory amb price  <u>actualitzar el departament amb el nou inventory</u>  llegir units de devices_per_type del departament i tipus corresponents  decrementar units  <u>actualitzar devices_per_type</u>  llegir most_device_type_dept de general_stats  si el departament era e most_device_type_dept:  contar quants registres hi ha a devices_per_type del departament  que  en tingui mes.  actualitzar general_stats amb nou device_type_number i  most_devices_types_dept de la consulta anterior</p>
--	--

usuari	
procediment	actuació
<b>alta</b>	incrementar <i>users</i> i <i>users_no_device</i> a <i>general_stats</i> incrementar users al departament corresponent
pseudocodi	AFTER INSERT guardar parametres d'entrada llegir <i>users</i> i <i>users_no_device</i> de <i>general_stats</i> incrementar els dos valors actualitzar <i>general_stats</i> amb els nous valors llegir <i>users</i> del departament corresponent incrementar el valor llegit actualitzar <i>department</i> amb el nou valor escriure registre al log amb OK si alguna cosa falla: registrat l'error al log

## Processos Temporitzats

Hi ha una sèrie de dades, derivades dels requisits d'informació estadística que ens demanen tenir disponibles de manera immediata, que caldrà actualitzar amb processos auto-executats en moments determinats. No son, per tant, processos ABM, sinó consultes (o càlculs sobre consultes) els resultats de les quals volem tenir disponibles sense haver d'esperar a que s'executin.

Cal comentar que, per tal de que no hi hagi col·lisions entre els processos temporitzats, en el cas de que puguin coincidir, al fer la programació s'han separat amb un marge d'una hora.

A continuació, en funció del moment en que s'executaran, en detallem les seves funcions:

Procediments temporitzats	
moment	actuació
<b>diari</b>	Actualitzar <i>old_inventory_devices</i> a <i>general_stats</i> .
pseudocodi	contar els dispositius amb data inventari de fa més d'un any actualitzar <i>old_inventory_devices</i> a <i>general_stats</i>
<b>mensual</b>	Actualitzar el <i>last_year_incidents</i> per a cada <i>department</i> Actualitzar els <i>worst_ratio_department</i> a <i>general_stats</i>
pseudocodi	contar els la suma dels incidents del usuaris de cada departament en els darrers 12 mesos <u>actualitza last_year_incidents a cada departament</u> busca el departament amb last_year_incidents mes gran <u>actualitza worst_ratio_department a general_stats</u>
<b>anual</b>	Crear nou registre a <i>year_stats</i> Posar a zero el <i>incidents_current_year</i> de tots els usuaris.
pseudocodi	crear un registre a <i>year_stats</i> amb el nou any i <i>incidents_current_year</i> a zero

## Consultes

Una de les conseqüències del requisit explícit de que la resposta de les consultes es pugui obtenir en temps 1, es que tota la complexitat es trasllada als processos ABM i els temporitzats. Per tant, les consultes demanades es resolen amb SELECTS directes sobre alguna taula o, com a molt, contra un INNER JOIN entre dues taules tal com es pot comprovar en el script de consultes del annex corresponent.

Tanmateix, a part dels requisits, s'ha considerat interessant afegir un parell de consultes addicionals que també es troben al mateix script, a saber:

- Dispositius assignats a un usuari concret
- Percentatge del pressupost executat per un departament en un any determinat.



## Optimització

El model que hem implementat no presenta grans complicacions més enllà de les derivades del gran volum de dades que ha de suportar. Tanmateix, aquest aspecte ja ha estat tingut en compte a l'hora de crear el repositori estadístic que s'actualitza en moments poc crítics (altes, modificacions, o moments de baixa exigència) per ser molt ràpid a l'hora de les consultes.

Tot i així, per agilitzar alguns processos s'han seguit les indicacions de Winand (2012) [3] pel que fa al ordre de les claus compostes i als índex addicionals.

En els casos de taules amb claus primàries compostes, s'ha posat en primer lloc la clau que s'ha considerat més rellevant a l'hora de fer consultes. Per exemple a la taula *process\_check* la clau esta formada per *id\_process\_audit* + *id\_incident* ja que, el mes normal es que fem consultes de una auditoria determinada.

En les taules amb clau combinada en les que ens interessa fer consultes 'des de' diferents claus, s'ha afegit un índex addicional per accelerar les consultes que no entren en l'ordre de la combinació de camps. Per exemple a *current\_assignment* on la clau es *id\_device+id\_user*, s'ha incorporat un índex per usuari.

També hi algun cas en que indexar per un camp no associat a cap clau ens pot ajudar. Per exemple, a la taula *order\_line* ens anirà bé un index per *id\_device* per quan vulguem consultar el preu de compra d'un dispositiu.

[3] Winand, M. (2012). *SQL Performance Explained*. <https://winand.at/>

## Carrega de Dades Bàsiques i Proves Inicials

En primer lloc carregarem les dades bàsiques, per poder començar a treballar, amg d'un script SQL que es pot veure al annex. Seran les corresponents a les taules d'enumeració (status i action\_type), les independents amb poques entrades (device\_type, department) i les corresponents a les inicialitzacions dels repositoris estadístics.

Tot seguit, poblarem les taules de proveïdors i usuaris. Per fer-ho, busquem a internet els noms de subministradors de hardware, així com els de noms i cognoms més habituals. Amb aquestes dades, i via un script en Python, generem les dades desitjades.

En total ens quedaran les taules amb els següents registres:

Dades bàsiques	
taula	registres
status	4
action_type	4
device_type	16
department	9
supplier	100
usuari	10.000
general_stats	1
year_stats	1

### Proves dels disparadors

A partir d'aquestes dades, i entrant manualment els registres que necessitem en cada cas, fem les proves per la validació inicial de cada un dels triggers programats.

#### alta d'usuari

verifiquem que, després de la càrrega de dades inicials, el registre *users* i *users\_no\_device* de *general\_stats* indiquen 10.000

#### alta de device\_type i department

verifiquem que, després de la càrrega de dades inicials, la taula *devices\_per\_type* conté 144 registres (9 departaments x 16 tipus de dispositiu) amb *units* a zero.

#### alta de order\_line

per poder provar les altes a *order\_line*, afegirem primer un registre a *investment\_order*, i dues orderx corresponent a dos proveïdors diferents. Ens caldrà verificar els següents casos:

alta order_line	
cas	resultat
primera línia d'una order	ok
segona línia d'una order	ok

#### modificació de order\_line

modifiquem les línies de comanda per comprovar els casos següents:

<b>modificació order_line</b>	
<b>cas</b>	<b>resultat</b>
no és un portatil	ok
és un portatil	ok
entrega més ràpida que el supplier ràpid	ok
entrega més lenta que el supplier lent	ok
correspon a investment order	ok
no correspon a investment order	ok

alta de acció

en les altes a accio cal, primer, verificar que el responsable es un tècnic.

<b>alta accio (tipus: tots)</b>	
<b>cas</b>	<b>resultat</b>
el responsable no és tecnic	ok
el responsable és tecnic	ok

Després, en funció del tipus d'acció, cal fer diferents proves:

<b>alta accio (tipus: assignació)</b>	
<b>cas</b>	<b>resultat</b>
l'usuari no tenia dispositius	ok
l'usuari té menys de 5 dispositius	ok
l'usuari té més de 5 dispositius	ok
el dispositiu es un portatil	ok
el dispositiu no és un portatil	ok
el departament es el que te més assignacions	ok
el departament no és el que te més assignacions	ok
el departament és el que te mes tipus	ok
el departament no és el que te mes tipus	ok

<b>alta accio (tipus: alliberament)</b>	
<b>cas</b>	<b>resultat</b>
l'usuari tenia 5 dispositius o menys	ok
l'usuari tenia més de 5 dispositius	ok
el dispositiu es un portatil	ok
el dispositiu no és un portatil	ok
el departament es el que tenia més assignacions	ok
el departament no és el que tenia més assignacions	ok
el departament és el que tenia mes tipus	ok
el departament no és el que tenia mes tipus	ok

<b>alta accio (tipus: baixa)</b>	
<b>cas</b>	<b>resultat</b>
el dispositiu estava assignat	ok
el dispositiu no estava assignat	ok
l'usuari tenia 5 dispositius o menys	ok
l'usuari tenia més de 5 dispositius	ok
el dispositiu es un portatil	ok
el dispositiu no és un portatil	ok
el departament es el que tenia més assignacions	ok
el departament no és el que tenia més assignacions	ok
el departament és el que tenia mes tipus	ok
el departament no és el que tenia mes tipus	ok

alta de incident

<b>alta incident</b>	
<b>cas</b>	<b>resultat</b>
el responsable no és tecnic	ok
el responsable és tecnic	ok
l'usuari es converteix en trouble_maker	ok
l'usuari no es converteix en trouble_maker	ok
el dispositiu és un portatil	ok
el dispositiu no és un portatil	ok

alta de process\_check

alta process_check	
cas	resultat
és el primer de la auditoria	ok
no és el primer de la auditoria	ok

alta de inventory\_check

alta inventory_check	
cas	resultat
és el primer de la auditoria	ok
no és el primer de la auditoria	ok

## Proves de Volum

Amb les proves del apartat anterior hem pogut verificar que els procediments ABM funcionen en els casos que hem previst. Tot i així. El sistema que hem preparat té, com a requisit principal, funcionar com a Data Warehouse i es per això que hem preparat el repositori estadístic actualitzat via triggers i procediments temporitzats, però amb aquestes proves de petita escala no podem garantir el correcte funcionament.

El principal problema de validar el disseny com Data Warehouse es que, per fer-ho, ens cal un volum gran de dades generades, a més, al llarg del temps per garantir que funcionen tant els disparadors associats als ABM com, també, els procediments temporitzats.

La solució que es planteja es basa en un programa fet en Python, en concret sobre l'entorn interactiu Jupyter Notebook (a l'annex corresponent es troba una versió en html i el notebook en si) per poder interactuar i canviar els paràmetres segons ens convingui, que ens genera el volum de dades que vulguem al llarg d'un període de temps. L'algoritme es el següent:

---

<b>Algoritme</b>	Generació de dades massives
<b>Entrada:</b>	dades de connexió amb la BBDD dades de inicialització de la BBDD data d'inici de les dades generades numero de cicles (operacions sobre la BBDD) rang d'increment de temps per cicle
<b>Sortida:</b>	Volum de dades, en funció dels cicles (de tipus aleatori)

connecta amb la base de dades  
esborra els registres de totes les taules  
inicialitza a 1 totes les seqüències de identificadors autoincrementats  
entra els valors de les taules basiques: action\_type, status, device\_type  
departments i general\_stats

entra els valors de la taula suppliers a partir d'un arxiu txt  
genera 10.000 usuaris diferents a partir de tres arxius amb noms de dona,  
home i cognoms. Assignant un departament aleatòriament  
data = data inicial  
**per** i des de 1 a numero de cicles **fer** :  
data = data + aleatori (entre 5 i 120 segons)  
**si** ha canviat l'any **llavors**:  
executar procediment anual  
**fi si**  
**si** ha canviat el mes **llavors**:  
executar procediment mensual  
**fi si**  
**si** ha canviat el dia **llavors**:  
executar procediment mensual  
**si** es final de trimestre **llavors**:  
fer inventari físic d'un número aleatori de dispositius  
eliminar les comandes d'inversió de més de 6  
mesos  
**fi si**  
**si** es final de semestre **llavors**:  
fer auditoria de processos sobre un nombre aleatori  
d'incidents  
**fi si**  
**fi si**  
triari un procés aleatòriament ponderant la freqüència  
**si** és una comanda **llavors**:  
crear una comanda nova  
associar-la, o no, a una ordre d'inversió aleatòriament  
associar-la, aleatòriament, a un proveïdor  
assignar un numero de línies aleatori (entre 1 i 10)  
**per** j des de 1 a numero de línies **fer**:  
crear una línia de comanda  
assignar aleatòriament el tipus de  
dispositiu  
assignar aleatòriament el preu (entre 50 i 5000)  
**fi per**  
**fi si**  
**si** és una recepció **llavors**:  
triari, aleatòriament, una línia de comanda no rebuda  
modificar la línia amb la data de recepció actual  
**fi si**  
**si** és una acció **llavors**:  
triari aleatòriament el tipus d'acció  
triari aleatòriament el tècnic responsable  
**si** és una alta **llavors**:  
triari, aleatòriament, un dispositiu pendent d'alta

modificar el seu status a pendent  
d'assignar

**fi si**

**si** és una assignació **llavors:**

triar, aleatòriament, un dispositiu pendent d'assignar  
triar, aleatòriament, un  
usuari  
assignar el dispositiu al  
usuari

**fi si**

**si** és un alliberament **llavors:**

triar, aleatòriament, un dispositiu  
assignat  
modificar el status a pendent d'assignar.

**fi si**

**si** és una baixa **llavors:**

triar aleatòriament si serà dispositiu assignat o  
no

**si** és assignat

**llavors:**

triar, aleatòriament, un dispositiu  
assignat  
modificar el status a  
baixa.

**si no**

**llavors:**

triar, aleatòriament, un dispositiu pendent  
d'assignar  
modificar el status a  
baixa.

**fi si**

**fi si**

**fi si**

**si** és un incident **llavors:**

tria, aleatòriament, un dispositiu assignat  
triar, aleatòriament, un tècnic  
dona d'alta el incident amb la data actual.

**fi si**

**fi per**

---

Un cop llegit el script anterior, s'ha pogut comprovar, amb diferents número de cicles (fins a 3 milions) i rangs temporals, que tots el valors del repositori estadístic es calculaven correctament.

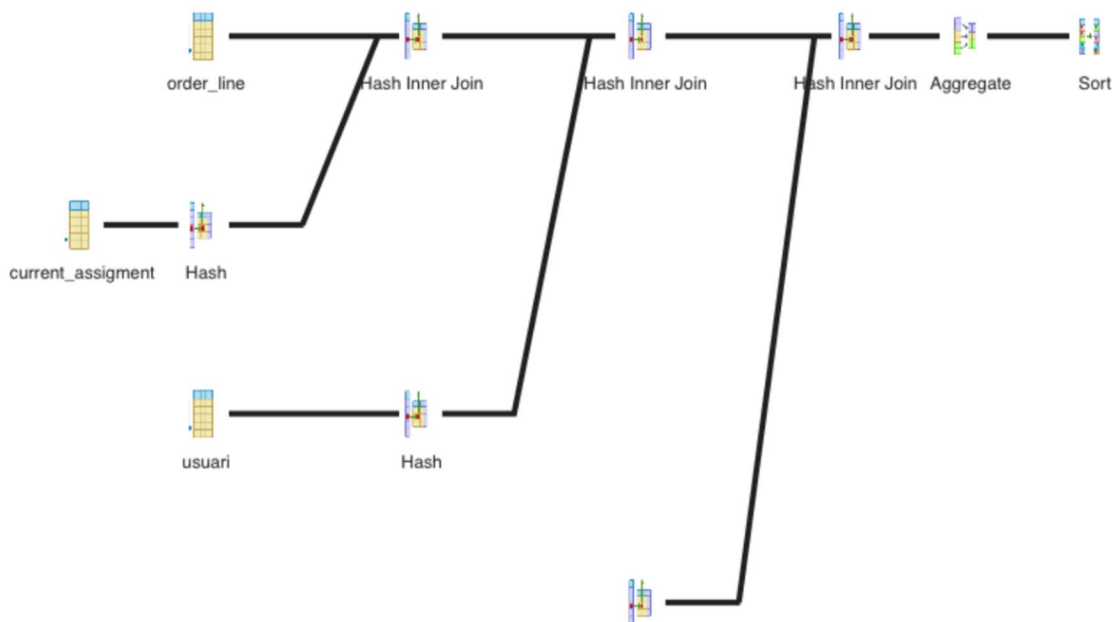
## Eficàcia del disseny Data Warehouse

Més enllà de la validació, amb gran volum de dades, del funcionament de la BBDD, es també interessant comprovar les avantatges del disseny data warehouse. Per fer-ho prendrem com a referència el requisit RF11-07 que ens demanava poder consultar el valor econòmic dels dispositius, actius, associats a un departament.

Per obtenir aquesta informació, sense el repositori estadístic, ens caldria fer la consulta SQL següent:

```
SELECT department.id_department, SUM(price) FROM order_line
JOIN current_assignment ON order_line.id_device =
current_assignment.id_device
JOIN usuari ON current_assignment.id_user = usuari.id_user
JOIN department ON usuari.id_department = department.id_department
GROUP BY department.id_department ORDER BY department.id_department;
```

Utilitzant la eina EXPLAIN de Postgres via el pgAdmin, podem veure, gràficament, quin es el recorregut que necessita executar el SGBD per resoldre la consulta:



Podem, també consultar el temps necessari per donar la resposta

Statistics per Node Type

Node type	Count	Time spent	% of query
Aggregate	1	8.68 ms	12.05%
Hash	3	5.466 ms	7.59%
Hash Inner Join	3	46.947 ms	65.13%
Seq Scan	4	10.96 ms	15.21%
Sort	1	0.035 ms	0.05%

en total son 48,92 ms , considerant que les principals taules implicades (order\_line, current\_assignment i usuari) tenen 74.000, 40.000 i 10.000 registres (aproximat) respectivament.



En el nostre cas, però, disposem del repositori estadístic per obtenir la mateixa informació, així que podem comparar.

En primer lloc el pla de consulta:



El canvi es evident ja que, tal com ens exigien en els requisits, es una cerca directa en temps constant (només dependrà del número de departaments).

Si consultem el temps invertit:

**Statistics per Node Type**

Node type	Count	Time spent	% of query
Seq Scan	1	0.019 ms	59.38%
Sort	1	0.014 ms	43.75%

Podem veure que, en total, son 0,033 ms, es a dir, per sobre de 2.000 vegades més ràpid que la consulta SQL sobre les taules. A més, aquesta diferencia s'incrementarà amb el número de registres ja que, la consulta al repositori sempre consumirà el mateix temps.

Podem comprovar que els resultats son els mateixos:

id_department [PK] integer	inventory double precision
1	11356971
2	11233831
3	11031743
4	11032486
5	11911967
6	11747036
7	11466232
8	11104573
9	11736228

Consulta Estadística

id_department [PK] integer	sum double precision
1	11356971
2	11233831
3	11031743
4	11032486
5	11911967
6	11747036
7	11466232
8	11104573
9	11736228

Consulta SQL

No totes les consultes replicaran aquest resultats, però l'exemple es suficient per determinar la conveniència del repositori estadístic per aquest tipus de sistema.

## Desenvolupament Futur

De cara a una futura ampliació del projecte considero que seria interessant explorar les següents possibilitats:

- Implementació del sistema en un servidor al núvol.
- Proves de accés simultani, i massiu, des de diferents clients.
- Afegir requisits de tipus NoSQL .
- Requerir atributs de tipus no convencional: geomètric, geogràfic, cadenes de bytes etc.

## Conclusions

Des de un punt de vista pràctic, aquest projecte m'ha permès aprofundir en l'aprenentatge de la programació en PLSQL, descobrir el funcionament i les possibilitats dels processos temporitzats i, en definitiva, les capacitats de control i gestió de bases de dades des de el propi SGBD.

A nivell més conceptual, haver de satisfer els requisits derivats del datawarehouse i la escalabilitat, m'han obligat a encarar i solucionar problemes que, en els projectes en que havia treballat fins ara, no s'havien presentat.

També ha estat un repte muntar un sistema que permetés, amb un mínim de garanties, provar un sistema d'aquestes característiques i, tot i que amb mancances, crec que el resultat es prou bo.

Finalment, la generació de una documentació complerta i estructurada, així com el seguiment d'una planificació estricta, han estat, també, novetats a nivell personal i, amb alguns problemes, m'han acabat semblant molt interessants.

No vull acabar aquest document sense un agraïment explícit al meu tutor, en Jordi Ferrer Duran, que ha estat en tot moment ràpid i concret en respondre els dubtes que em sorgien i, alhora, exigent i motivador quan ha fet falta.

## Glossari

**ABM:** Alta, baixa i modificació. Referit a les accions possibles sobre una taula. En anglès s'utilitza CRUD (Create, Read, Update, Delete).

**BBDD:** Base de dades.

**Data Warehouse:** Magatzem de dades. Referit a sistemes de emmagatzematge de grans volums de dades.

**ERD:** Diagrama d'Entitat-Relació. Referit a un dels tipus de diagrama habituals per representar una base de dades.

**ITSM:** Information Technology Service Management. Aplicacions que permeten gestionar els tots processos informàtics, com la gestió d'incidències, el inventari, etc.

**NoSQL:** Not Only SQL (no només SQL). Referit a SGBD que permeten definir funcionaments no relacionals.

**PL/SQL:** Procedural Language/Structured Query Language es una ampliació del llenguatge SQL (declaratiu) que permet utilitzar eines de tipus procedimentals dins la base de dades. Tot i que es propietat d'Oracle està també implementat en altres gestors, com ara PostgreSQL

**SQL:** Structured Query Language, es un llenguatge declaratiu per consultar bases de dades relacionals.

**SGBD:** Sistema Gestor de Bases de Dades. En anglès DBMS (Data Base Management System)

**TI:** Tecnologies de la Informació

**Trigger:** Disparador. Procediments programats que s'executen quan es produeix una determinada ABM sobre la base de dades.

**UML:** Unified Modeling Language o llenguatge unificat de modelització, es un llenguatge gràfic universal per modelitzar software.

**uuid:** universal unique identifier (identificador únic universal). Identificadors de 128 bits (16 bytes) generats segons un mètode estandarditzat que -gairebé- garanteixen que no es poden produir duplicats.

## Bibliografia

- [1] Pradel Miquel, J., Raya Martos (2021), J. *Introducció a l'enginyeria de requisits*. UOC
- [2] Robertson, J.; Robertson, S. *Volere Requirements Specification Template*.  
<https://www.volere.org/templates/volere-requirements-specification-template/> . Access  
desembre 2021
- [3] Winand, M. (2012). *SQL Performance Explained*. <https://winand.at/>. Access desembre 2021
- [4] Barrett, Zoe (2016). *Dive into Postgres*. Munich: BookkRix GmbH & Co. KG
- [4] The PostgreSQL Development Group (2021). *PostgreSQL 14.1 Documentation*.  
<https://www.postgresql.org/docs/current/>. Access desembre 2021

## Annexos

### Annex 1. Script de Creació de la Base de Dades

Veure arxiu: *creacio\_BBDD.sql*

### Annex 2. Script de Programació

Procediments ABM

Veure arxiu: *funcions\_triggers.sql*

Procediments Temporitzats

Veure arxiu: *funcions\_triggers.sql*

### Annex 3. Scripts de Càrrega de Dades i Proves

Càrrega manual

Veure arxiu: *dades\_basiques.sql*

Càrrega automàtica

Veure arxiu: *dades\_masives.ipynb* i *dades\_masives.py*, o *dades\_masives.html*

### Annex 4. Script de Consultes

Veure arxiu: *consultes.sql*

## Annex 5. Guia d'Instal·lació

Amb la finalitat de simplificar la possibilitat de reproduir el sistema del projecte, validar les consultes, i experimentar amb el generador de dades, s'adjunta aquesta guia d'instal·lació amb les passes que calen per arribar a tenir un producte operatiu.

### 1. Instal·lar Postgres (millor si es la versió 14 o superior).

Cal tenir en compte que, a més del SGBD instal·larem la eina de control pgAdmin i el scheduler pgAgent.

En el cas de Windows, el més senzill és utilitzar l'instal·lador de EDB:

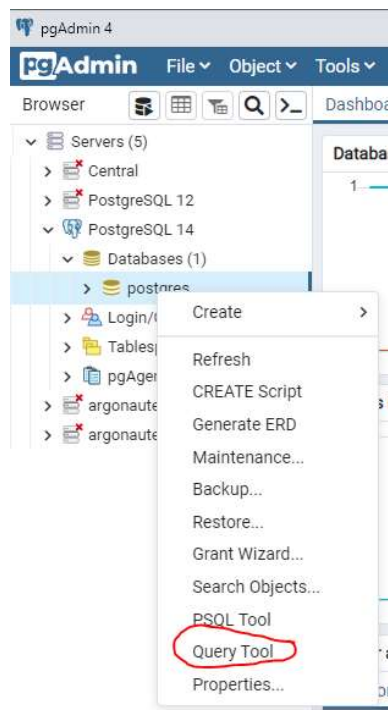
<https://www.enterprisedb.com/downloads/postgres-postgresql-downloads> que ja ens permet muntar els tres elements.

Per Mac OS o Linux, podem, també, utilitzar l'instal·lador de EDB per Postgres i pgAdmin o el oficial <https://www.postgresql.org/download/> però, caldrà després, instal·lar el pgAgent seguint les instruccions:

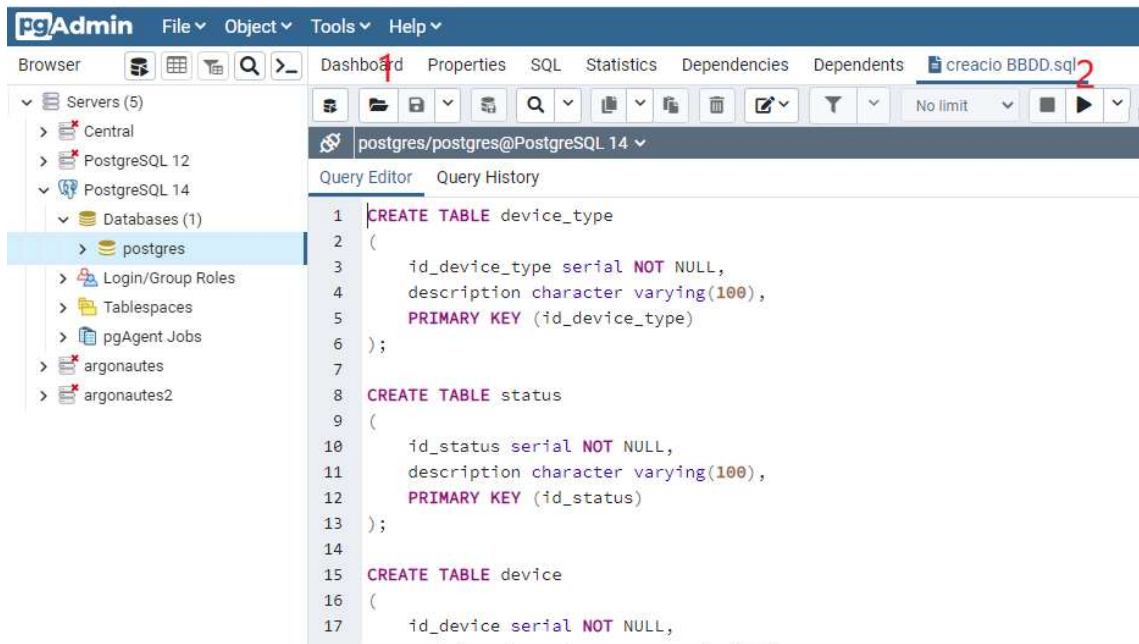
[https://www.pgadmin.org/docs/pgadmin4/development/pgagent\\_install.html](https://www.pgadmin.org/docs/pgadmin4/development/pgagent_install.html)

### 2. Crear la Base de Dades. Per simplificar el procés, ho farem des de pgAdmin i, quan ens demani nom i password podem utilitzar <postgres> en ambdós casos.

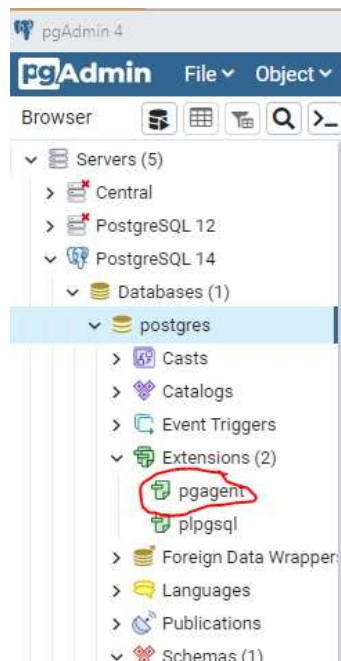
També per simplificar, utilitzarem la base de dades que es crea per defecte i que es diu (com no) *postgres*. Caldrà localitzar-la en el menú de la esquerra, obrir el menú desplegable i triar la opció *Query Tool*.



Ens obrirà una finestra amb el Query Editor i cal, primer, utilitzar el selector d'arxius per carregar l'script *creacio\_BBDD.sql* i executar-lo



Cal verificar que tenim activat el pgAgent i, si no, activar-lo ara afegint la extensió corresponent:



3. **Càrrega Funcions i Triggers.** Des del Query Editor (igual que en el pas anterior) carreguem i executem l'arxiu *funcions\_triggers.sql*
4. **Càrrega de dades bàsiques.** Per poder fer proves senzilles, amb pocs registres, podem carregar l'arxiu *dades\_basiques.sql* (amb el mateix procediment). Això buida les taules si hi ha algun registre i omple les taules d'enumeracions, els departaments i els tipus de dispositius per poder començar a entrar registres.
5. **Carregar de dades massives.** Per poder fer proves 'reals' sobre el sistema ens caldrà un volum de dades massa gran per entrar-les a mà. Per fer-ho, utilitzarem un script en Python que, a partir d'una data inicial i un numero de cicles determinat, poblarà les taules amb suficients registres per provar totes les funcionalitats. Disposem de l'script en python 'convencional' *dades\_masives.py* o be del jupyter notebook *dades\_masives.ipynb*. En ambdós casos cal comprovar que les constants definides per connectar amb la BBDD i per fixar el número de cicles i la data inicial, son les que ens interessen. Cal assegurar-se que, en el mateix directori on hi tinguem el notebook o l'script convencional, hi hagi els arxius: *suppliers.txt*, *men.txt*, *women.txt* i *last\_name.txt*

```
# Constants
BBDD = 'postgres'
HOST = 'localhost'
USER = 'postgres'
PASW = 'postgres'
PORT = '5432'

INICI = datetime(2022,1,15, 0,0,0)
CICLES = 100000
MIN_S = 5           # mínim numero de segons que el temps avança per iteració
MAX_S = 120        # màxim numero de segons que el temps avança per iteració
```

Com a referencia, en el sistema que hem provat, 250.000 iteracions triguen 6 minuts.

6. **Proves.** Un cop disposem de les dades, a l'arxiu *consultes.sql* hi trobarem totes les consultes que permeten satisfer els requisits d'informació estadística.