

Estàndards de desenvolupament  
d'aplicacions del GOIB.  
**Base de dades**



G CONSELLERIA  
O ADMINISTRACIONS  
I PÚBLIQUES I  
B MODERNITZACIÓ  
/ DIRECCIÓ GENERAL  
MODERNITZACIÓ I  
ADMINISTRACIÓ DIGITAL

Palma, desembre de 2020

## Índex

<b>HISTORIAL DE VERSIONS.....</b>	<b>3</b>
<b>1. INTRODUCCIÓ.....</b>	<b>4</b>
<b>2. BASE DE DADES I VERSIONS SUPORTADES.....</b>	<b>5</b>
<b>3. NOMENCLATURA.....</b>	<b>7</b>
3.1. APLICACIONS.....	7
3.2. Objectes de bases de dades.....	7
3.3. Tablespaces.....	7
3.4. Taules i vistes.....	8
3.5. Seqüències.....	9
3.6. Restriccions (constraints).....	10
3.7. Altres restriccions.....	10
3.8. Índexs.....	11
3.9. Rols.....	11
3.10. Procediments, funcions, paquets i resta d'objectes.....	11
3.11. Sinònims.....	12
<b>4. RESTRICCIONS I RECOMANACIONS.....</b>	<b>13</b>
4.1. Accessos a les bases de dades.....	13
4.2. Tipus de dades.....	13
4.3. Codificació.....	13
4.4. Deadlocks.....	13
4.5. Particionament de taules.....	14
<b>5. REPRESENTACIÓ DE LA LÒGICA DE NEGOCI.....</b>	<b>15</b>
5.1. Database Links.....	15
5.2. Jobs.....	15
5.3. Sinònims.....	15
5.4. Disparadors.....	16
<b>6. PROBLEMES FREQUENTS.....</b>	<b>17</b>
6.1. Nomenclatura d'objectes.....	17
6.2. Objectes de tablespaces incorrectes.....	17
6.3. Ús de camps byte en lloc de char.....	17
6.4. Assignació de privilegis incorrectes.....	18
6.5. Objectes amb codi dins del script SQL.....	18
<b>7. ENVIAMENT DEL QUADERN DE CÀRREGA I EXECUCIÓ DE SCRIPTS.....</b>	<b>19</b>

## Historial de versions

Data	Versió	Descripció	Autor
05/04/16	7.0	Revisió d'estàndards	DGDT
28/05/19	7.5	Revisió d'estàndards	DGMAD
17/01/20	9.0	Revisió d'estàndards	DGMAD
03/12/20	9.5	Traducció al català i aclariments	DGMAD

## 1. INTRODUCCIÓ

Aquest document descriu l'estàndard de base de dades que s'ha de seguir per al desenvolupament d'aplicacions que s'instal·laran en els servidors de la *Direcció general de Modernització i Administració Digital (DGMAD)*.

Les principals novetats respecte de la versió anterior són les següents:

1. Actualització de les bases de dades suportades.
2. Nova secció de representació de la lògica de negoci.
3. Nova secció de problemes freqüents.

Per més informació sobre el procediments a seguir per enviar peticions (CAIs) consultar el document «Estàndard d'implantació d'aplicacions».

## 2. Base de dades i versions suportades

La aplicació haurà d'estar preparada per funcionar sobre una base de dades **Oracle 12.2**, concretament sobre la versió 19c (*release 12.2.0.3*) que serà la utilitzada per totes les noves aplicacions.

Millores de la versió 12.2:

- Suport a llarg termini per part del proveïdor (fins a 2026).
- Actualització directa des de les versions 11<sup>g</sup> (*release 2*), 12<sup>c</sup> i 18<sup>c</sup>.
- Indexació automàtica.
- Redireccionament DML d'Activi Data Guard.
- Taules amb particions híbrides.
- Quarantena de consultes.
- Compatibilitat amb JSON.

Quant a les aplicacions ja existents, des de la DGMAD s'està treballant en la migració a versions superiors lloc que, algunes d'elles, com les versions 9i o la 10g, van deixar de tenir suport amb el consegüent risc de seguretat i estabilitat que això comporta.

A continuació es mostra una taula de controladors JDBC i versions JDK que haurà de ser tinguda en compte per a les migracions a versions superiors.

BD Oracle	Versions JDBC i JDK
19c	ojdbc10.jar amb JDK10, JDK11 ojdbc8.jar amb JDK8, JDK9 i JDK11
18c	ojdbc8.jar amb JDK8, JDK9, JDK10 i JDK11
12c Release 2 (12.2)	ojdbc8.jar amb JDK8
12c Release 1 (12.1)	ojdbc7.jar amb JDK7 i JDK8 ojdbc6.jar amb JDK6
11g Release 2 (11.2)	ojdbc6.jar amb JDK6, JDK7 i JDK8 (Nota: JDK7 i JDK8 només estan suportades en 11.2.0.3 i 11.2.0.4) ojdbc5.jar amb JDK5



La necessitat d'utilització d'altres bases de dades (com PostgreSQL) haurà de venir motivada i ser aprovada per part del responsable de l'Àrea d'Administració de Bases de Dades a causa de les limitacions de suport i alta disponibilitat.

## 3. Nomenclatura

### 3.1. APLICACIONS

Totes les aplicacions han de disposar d'un codi d'aplicació i un prefix que hauran estat facilitats prèviament pel *Centre de Processament de Dades* de la DGMAD (veure document «Estàndards d'implantació d'aplicacions», Capítol 2. «Sol·licitud de codi d'aplicació»).

Tots els objectes de base de dades d'una aplicació seran propietat d'un mateix usuari de base de dades que haurà de coincidir amb el codi d'aplicació prèviament assignat per la DGMAD.

Així mateix, tots els objectes de base de dades començaran per un prefix de tres lletres, també assignat per la DGMAD, seguits d'un guió baix «\_».

En els exemples inclosos en aquest document, cada vegada que es faci referència al codi d'aplicació s'utilitzarà com a exemple el literal «APLICACIO», i com a prefix el literal «APL».

### 3.2. Objectes de bases de dades

El nom dels objectes serà com a màxim de 30 caràcters, i només podran incloure els caràcters A-Z, a-z, 0-9 i guió baix (\_).

La creació dels objectes no pot incloure cometes en la definició del nom de l'objecte.

### 3.3. *Tablespaces*<sup>1</sup>

En general, tots els objectes de l'usuari de base de dades hauran de crear-se en el *tablespace* de nom igual que l'usuari.

Si l'aplicació fa un ús intensiu d'índexs, és recomanable utilitzar *tablespaces* separats per a dades i índexs.

Si l'aplicació conté LOBs<sup>2</sup> és obligatori emmagatzemar-los en un *tablespace* separat de la resta d'objectes per motius de rendiment i administració.

---

1 Oracle divideix l'espai disponible en unitats lògiques denominades tablespaces, que tenen un nom identificatiu i un espai reservat per emmagatzemar elements d'una BBDD.

2 Els objectes grans (Large Objects - LOB) són un conjunt de tipus de dades que estan dissenyats per a contenir grans quantitats de dades.

En el moment de sol·licitar el codi d'aplicació a través del document d'inici de projecte (veure «Estàndards d'implantació d'aplicacions») s'ha d'indicar si l'aplicació utilitza LOBs i si es volen tenir dades i índexs separats.

En el cas de tenir els objectes en dos o més *tablespaces* (dades, índexs i LOBs) els noms dependran del tipus d'aplicació, com s'indica en la taula següent:

Tipus d'aplicació	Tablespace de dades	Tablespace d'índexs	Tablespace de LOB
Aplicació de grandària reduïda	APLICACIO	APLICACIO	No permesos
Aplicació de grandària reduïda amb LOB	APLICACIO	APLICACIO	APLICACIO_LOB
Aplicació amb dades i índexs separats	APLICACIO_DADES	APLICACIO_INDEX	No permesos
Aplicació amb dades i índexs separats i LOB	APLICACIO_DADES	APLICACIO_INDEX	APLICACIO_LOB

Per separar els objectes en un major nombre de tablespaces s'haurà de consultar amb l'Àrea d'Administració de Bases de Dades de la DGMAD.

### 3.4. Taules i vistes

Les taules (TABLE) i vistes (VIEW) seguiran el patró **APL\_XXX**, on XXX és el nom representatiu de l'entitat.

Exemples:

- APL\_CLIENT
- APL\_FACTURA

En les taules resultants d'una relació N:M, el nom de la taula contindrà el nom (o part d'ell) de cadascuna de les taules separades amb guió baix.

Exemple: taula resultant d'una relació N:M entre APL\_CLIENT i APL\_FACTURA:

- APL\_CLIENT\_FACTURA



### 3.4.1. Taules amb camps LOB

Per crear taules que tinguin algun camp d'objectes grans de tipus LOB (BLOB, CLOB o NCLOB) se seguirà el següent format (en l'exemple es crea la taula APL\_TAULA):

```
CREATE TABLE APL_TAULA
(
  camp1 tipus_camp1,
  camp2 CLOB,
  camp3 BLOB)
TABLESPACE APLICACIO_DADES
LOB (camp2) STORE AS APL_TAULA_camp2_LOB
(TABLESPACE APLICACIO_LOB
INDEX APL_TAULA_camp2_LOB_I)
LOB (camp3) STORE AS APL_TAULA_camp3_LOB
(TABLESPACE APLICACIO_LOB
INDEX APL_TAULA_camp3_LOB_I);
```

on:

- APL\_TAULA representa el nom de la taula.
- camp2 representa un camp de tipus CLOB.
- camp3 representa un camp de tipus BLOB.
- APLICACIO\_DADES representa el nom del *tablespace* de DADES.
- APLICACIO\_LOB representa el nom del *tablespace* de LOB.

El nom de l'objecte LOB (clàusula STORE AS) i del seu índex associat es forma afegint al nom de la taula el nom del camp LOB, seguit del sufix \_LOB (o \_LOB\_I per a l'índex del LOB). Si el nom resultant supera els 30 caràcters, s'ha de reduir el nom del camp o de la taula, però sempre mantenint el prefix de l'aplicació (APL\_) i el sufix (\_LOB o \_LOB\_I).

### 3.4.2. Columnes

Els noms dels camps o columnes de les taules són lliures; no és necessari que comencin per les tres lletres identificatives del nom de la taula. L'única restricció és la limitació de la grandària màxima a 30 caràcters.

## 3.5. Seqüències

Les seqüències (SEQUENCE) seguiran el patró **APL\_XXX\_SEQ**, on XXX és un nom representatiu de la taula o camp per la qual es crea la seqüència.

Exemple:

- APL\_CLIENT\_SEQ: per a la seqüència del codi de la taula APL\_CLIENT.

### 3.6. Restriccions (constraints)

#### 3.6.1. Clau Primària

Les claus primàries seguiran el patró **APL\_XXX\_PK**, on XXX indica el nom de la taula per la qual es crea la clau primària.

Exemple:

- APL\_CLIENT\_PK

#### 3.6.2. Clau Estrangera

Les claus estrangeres seguiran el patró **APL\_XXX\_YYY\_FK**, on XXX indica el nom de la taula d'origen i YYY indica el nom de la taula referenciada.

Exemple: clau estrangera de la taula APL\_CLIENT cap a la taula APL\_ILLA

- APL\_CLIENT\_ILLA\_FK

Si en la taula d'origen apareixen varis camps que han d'apuntar a la clau primària de la mateixa taula referenciada, s'afegirà el nom del camp (ZZZ) segons el patró **APL\_XXX\_ZZZ\_YYY\_FK**. Si es nom és massa llarg es podran utilitzar abreviatures per reduir la llargària.

Exemple: claus estrangeres de la taula APL\_CLIENT pels camps TANTERIOR i TACTUAL cap a la taula APL\_TIPUS

- APL\_CLIENT\_TANTERIOR\_TIPUS\_FK
- APL\_CLIENT\_TACTUAL\_TIPUS\_FK

### 3.7. Altres restriccions

La resta de restriccions seguiran el patró **APL\_XXX\_YYY\_ZZ**, on XXX indica el nom de la taula, YYY és el nom representatiu del camp o camps afectats, i ZZ és el sufix que indica el que fa la restricció que pot prendre un d'aquests valors:

- \_UK: per a claus úniques (UNIQUE).
- \_CK: per a restriccions de comprovació (CHECK).
- \_NN: per a restriccions no nulles (NOT NULL).

Per a les restriccions NOT NULL no és necessari definir el nom si es fan en línia (quan es defineix el nom del camp en la sentència de creació de taula). Per a la resta de restriccions s'ha de definir el nom seguint el patró descrit.

Exemples:

- APL\_CLIENT\_NIF\_UK: clau única pel camp NIF de la taula APL\_CLIENT
- APL\_CLIENT\_SEXE\_CK: comprovació del camp SEXE de la taula APL\_CLIENT
- APL\_CLIENT\_EDAT\_NN: el camp EDAT de la taula APL\_CLIENT no pot ser nul

### 3.8. Índexs

Els índex (INDEX) seguiran el patró **APL\_XXX\_YYY\_I**

En general, els índexs segueixen la mateixa nomenclatura que la restricció (*constraint*) corresponent, seguida del sufix '**I**'. Per a la resta d'índexs, XXX indica el nom de la taula i YYY el nom del camp a indexar.

Exemples:

- APL\_CLIENT\_PK\_I: índex per a la clau primària
- APL\_CLIENT\_ILLA\_FK\_I: índex per a la clau estrangera
- APL\_CLIENT\_NUM\_SS\_I: índex pel camp NUM\_SS

### 3.9. Rols

Els rols seguiran el patró **APL\_XXX**, on XXX és el nom representatiu del rol.

Exemples:

- APL\_CONSULTA
- APL\_USUARI
- APL\_ADMIN

### 3.10. Procediments, funcions, paquets i resta d'objectes

Generalment, no es permet l'ús de procediments (*procedure*), funcions (*function*), ni paquets (*packages*),

En els casos que el seu ús estigui justificat, la nomenclatura és més lliure; sempre que se segueixi la norma de començar cada nom pel prefix de l'aplicació, i que el nom de l'objecte sigui el més simple i representatiu possible.

Encara que no és obligatori, es recomana utilitzar un sufix per a cada tipus d'objecte a fi d'identificar-los ràpidament. Es proposen els següents:

- Procediments: APL\_XXX\_PR
- Funcions: APL\_XXX\_FN
- Paquets: APL\_XXX\_PQ

### **3.11. Sinònims**

No està permesa la creació de sinònims públics. Els sinònims han de quedar associats a l'esquema de l'usuari propietari de l'aplicació a partir del com hauran de crear-se. Si s'ha de substituir un sinònim, s'ha de fer abans el DROP SYNONYM i després tornar-ho a crear. No està permès utilitzar l'opció de CREATE OR REPLACE.

En qualsevol cas, deixa de ser necessari la creació de sinònims per poder referenciar els objectes de l'usuari APLICACIO des de l'usuari WWW\_APLICACIÓ. En el seu lloc, a més de concedir els permisos (grants) des de l'usuari APLICACIO a l'usuari WWW\_APLICACIÓ, es definirà l'esquema per defecte al fitxer datasource «APLICACIO-ds.xml»:

```
<new-connection-sql>  
  BEGIN  
    EXECUTE IMMEDIATE 'ALTER SESSION SET CURRENT_SCHEMA=APLICACIO';  
  END;  
</new-connection-sql>
```

## 4. RESTRICCIONS i recomanacions

### 4.1. Accessos a les bases de dades

Per a l'accés a la base de dades haurà de definir-se un pool de connexions. L'usuari del pool de connexions seguirà la nomenclatura **WWW\_APLICACIO**, on APLICACIO coincideix amb el codi d'aplicació assignat per la DGMAD.

Para que el usuari WWW\_APLICACIO pugui utilitzar els objectes de l'usuari propietari (usuari APLICACIO) serà necessari donar els privilegis d'accés (grants) adequats. Els permisos no es donaran directament a usuaris finals; es donaran només a rols o a usuaris tipus WWW\_APLICACIO.

Els únics permisos que es permeten assignar a rols o usuaris són: SELECT, INSERT, UPDATE, DELETE, EXECUTE.

Exemple: les sentències GRANT relatives a la taula APL\_CLIENT de l'aplicació APLICACIO podrien ser:

```
GRANT SELECT, INSERT, UPDATE, DELETE ON APL_CLIENT TO WWW_APLICACIO;
```

### 4.2. Tipus de dades

- No es permetrà la utilització de camps de tipus LONG.
- En el cas de crear camps de tipus «VARCHAR2» o «CHAR», en indicar el nombre de caràcters es posarà esment al fet que no posi «BYTE».

### 4.3. Codificació

- El joc de caràcters de les bases de dades serà «UTF8» (NLS\_CHARACTERSET = UTF8).
- El «national character set» serà «AI16UTF16» (NLS\_NCHAR\_CHARACTERSET = AI16UTF16).
- Les bases de dades tindrà el valor: «NLS\_LENGTH\_SEMANTICS = CHA».

### 4.4. Deadlocks

Per evitar les abraçades mortals (deadlocks) que es produeixen en esborrar registres d'una taula que té taules relacionades (taules filles), es recomana crear un índex per a les claus estrangeres de la taula filla.

## 4.5. Particionament de taules

Per a taules amb un volum de dades considerable i per les quals es realitzin moltes consultes per data, es recomana el particionament de les mateixes. Consultar abans amb els administradors de base de dades de la DGMAD per veure si això és possible i/o aplica.

## 5. Representació de la lògica de negoci

La lògica de negoci s'ha d'implementar en la capa de negoci de la aplicació, on resideixen els programes que s'executen, i no en la capa d'emmagatzematge. D'aquesta forma, la base de dades únicament exerceix la funció d'emmagatzematge de les dades. Consultar el document «Estàndards Jakarta EE» para més informació.

Per tant, no es permet l'ús de procediments emmagatzemats, paquets (packages), database links, jobs, disparadors (triggers),... en la base de dades. El seu ús únicament està justificat per compatibilitat amb bases de dades antigues que ja disposen d'ells i per necessitat molt motivada que haurà de ser justificada per escrit.

### 5.1. Database Links

Els database links no estan permesos excepte en els següents casos:

- Compatibilitat amb versions antigues que ja disposen d'ells. En tal cas, per temes de seguretat, és obligatori que el database link sigui anònim i no tingui credencials configurades. S'utilitzen les credencials pròpies de l'usuari que està utilitzant el DB Link.
- Aplicacions que utilitzen Pentaho.

### 5.2. Jobs

Els jobs no estan permesos excepte en el cas de bases de dades antigues que ja disposin d'ells. No obstant això, s'evitarà en la mesura dels possible la implementació de nous jobs.

En el seu lloc, en cas que aquesta lògica hagi de ser implementada en la base de dades, es recomana la utilització de crontab. Contactar amb els administradors de base de dades de la DGMAD per a més informació.

### 5.3. Sinònims

Els sinònims públics no estan permesos i la seva necessitat d'implementació ha d'estar molt justificat. Deixa de ser necessària la creació de sinònims per referenciar els objectes de l'usuari APLICACIO des de WWW\_APLICACIO. Consultar l'apartat «Nomenclatura – Sinònims» per a mes informació.

## **5.4. Disparadors**

Els disparadors (triggers) no estan permesos i la seva necessitat d'implementació en base de dades ha d'estar molt justificat.



## 6. Problemes freqüents

A continuació es mostra un llistat dels problemes més comuns detectats durant la validació d'estàndards de bases de dades, i que seria convenient revisar abans de demanar una instal·lació d'aplicacions en els servidors de la DGMAD.

### 6.1. *Nomenclatura d'objectes*

El principal problema trobat és el referent a la nomenclatura d'objectes. En general, se segueix l'estàndard de nomenclatura pel nom de les taules, però per a la resta d'objectes no es compleix (índexs, claus primàries i estrangeres, etc.).

### 6.2. *Objectes de tablespaces incorrectes*

Quan es crea un usuari amb dos o més tablespaces, el tablespace per defecte de l'usuari és el de dades. Moltes vegades es defineixen els índexs sense indicar el nom del tablespace d'índexs, per la qual cosa es crea en el tablespace per defecte de l'usuari. A l'hora de crear un índex o un LOB, s'ha d'indicar expressament que es creu en el tablespace corresponent.

Exemple:

```
CREATE INDEX APL_CLIENT_NIF_I ON APL_CLIENT(NIF) TABLESPACE
APLICACIO_INDEX;
```

### 6.3. *Ús de camps byte en lloc de char*

Les bases de dades de la DGMAD tenen definit el NLS\_LENGTH\_SEMANTICS a CHAR. L'ús de BYTE no està permès. La definició de columnes VARCHAR2 s'ha d'enviar en CHAR o sense posar res.

Exemple:

- Incorrecte 

```
CREATE TABLE APL_CLIENT
(camp1 VARCHAR2(100 BYTE));
```
- Correcte 

```
CREATE TABLE APL_CLIENT
(camp1 VARCHAR2(100 CHAR));
```

#### **6.4. Assignació de privilegis incorrectes**

Els únics privilegis permesos són: SELECT, INSERT, UPDATE, DELETE i EXECUTE. No pot haver-hi privilegis ALTER ni REFERENCES (en cas d'haver d'utilitzar-los s'ha de consultar abans amb els administradors de base de dades de la DGMAD).

És habitual trobar en els scripts de permisos ALTER a seqüències. Si s'ha de modificar una seqüència, s'ha de fer enviant el corresponent quadern de càrrega a la DGMAD amb la sentència DDL corresponent.

#### **6.5. Objectes amb codi dins del script SQL**

Els scripts de base de dades s'executen en SQL\*Plus. És habitual que els desenvolupadors utilitzin TOAD o altres programes per desenvolupar, però en enviar els scripts de base de dades aquests no funcionen en el SQL\*Plus.

L'error més habitual és no posar la barra (/) al final dels objectes que tenen codi (disparadors, package/procedure/function amb loop/declari/begin-end), ja que aquests no es compilen ni executen. **La barra (/) s'ha de posar després de cada objecte que conté codi.** Per exemple, si en un mateix arxiu s'envien cinc disparadores, hi ha d'haver una barra (/) després de la definició de cada disparador.

## 7. Enviament del quadern de càrrega i execució de scripts

L'execució dels scripts de base de dades ha de seguir el procediment especificat en el document "Estàndards d'implantació d'aplicacions", Capítol «3.5.1. Scripts de generació dels objectes de base de dades».

S'ha de prestar especial atenció al següent:

- S'han d'enumerar els arxius (en el cas d'instal·lacions de molts arxius és difícil trobar l'arxiu a executar) amb el nombre corresponent a l'ordre d'execució de la instal·lació (exemple: 01-taules.sql, 02-vistes.sql, 03-inserts.sql, ...).
- Els noms d'arxiu no han d'incloure espais en blanc ni caràcters especials.
- No cal barrejar sentències DDL i DML en el mateix arxiu.
- Evitar enviar arxius molt extensos, ja que si falla algun punt de l'arxiu, aquest es continua executant fins al final, i pot fer que la volta enrere de la instal·lació sigui molt complicada.
- En el cas d'haver d'executar scripts en lloc de sentències DML independents, s'haurà d'enviar cada script en un arxiu a part.
- **No enviar els scripts DML amb sentències commit.** Si l'execució del quadern de càrrega és correcta, els administradors de la base de dades ja faran el commit corresponent. Això pot evitar molts problemes ja que si s'envien cinc scripts i tots fan commit, però l'últim d'ells falla, en haver-se fet commit en els scripts anteriors no és possible tornar enrere la instal·lació completa fent un rollback.
- Per contra, **si el script s'ha de programar si cal enviar commit i exit al final del script.**
- Quant a la codificació dels fitxers, **tots els scripts de bases de dades hauran d'estar codificats en Unicode (UTF-8), Occidental (ISO-8859-1) o ASCII. En cas d'enviar diversos scripts no es poden barrejar fitxers codificats en UTF-8 amb fitxers codificats en ISO. No s'accepta fitxers amb BOM.**
- Per a més informació consultar el manual «Estàndards d'implantació d'aplicacions».