



Creació i Gestió de taules

NF2 - Llenguatge SQL: DDL

UF2 – Llenguatge SQL: DML i DDL

Que veurem?

1. Creació de taules
 - I. Consideracions noms de les taules
 - II. Tipus de dades
 - III. Resum
 - IV. Exercici
 - V. Creació d'una taula amb dades recuperades d'una consulta
2. CONSULTAR TAULA D'USUARI
 - I. Activitats proposades
3. INTEGRITAT
 - I. Integritat de dades
 - II. Integritat Referencial
4. RESTRICCIONS
 - I. Cas Pràctic (1)
 - II. Clau primària. Restricció PRIMARY KEY
 - III. Cas Pràctic (2)
 - IV. Clau forana. Restricció FOREIGN KEY
 - V. Cas Pràctic (3)

Que veurem?

4. RESTRICCIONS

VI. Resum - On Delete

VII. Restricció NOT NULL

VIII. Valors per defecte. L'especificació DEFAULT

IX. Cas pràctic (4)

X. Verificació de condicions. La restricció CHECK

XI. Cas Pràctic (5)

XII. Activitat

XIII. Restricció UNIQUE

XIV. Vistes del diccionari de dades per a les restriccions

XV. Desactivar restriccions

XVI. Activar restriccions

XVII. Canviar els noms a les restriccions

5. MODIFICAR TAULES

I. Canviar noms

II. Esborrar contingut de taules

III. Modificar taules

6. ESBORRAR TAULES



Que veurem? (II)

8. Restriccions

VI. Afegir restriccions

VII. Esborrar restriccions

VIII. Desactivar restriccions

IX. Activar restriccions

9. Resum



1. CREACIÓ DE TAULES

- ❑ Per crear taules s'usa la sentència **CREATE TABLE**.

- ❑ **SINTAXIS**

```
CREATE TABLE nom_taula (  
    DEFINIC_CAMP1 nom de la columna1 tipus de dada [, ...] [restriction1,  
    [, restriction2, ...[, restricctionM] ..]);
```

- ❑ Existeixen dos tipus d'elements: els camps i les restriccions, encara que els únics que són obligatoris són els primers.
- ❑ La definició del camp:

```
Nom_camp tipus [(dim [, mida])] [null | not null] [ default valor]
```

- **Obligatori**, el nom de la taula és obligatori hi ha de ser únic a tota la taula.
- **Opcional**, indicar si accepta nuls o no, i si tindrà valor per defecte quan és realitzi una inserció d'una fila.
 - **Si no s'especifica el contrari el camp acceptarà nuls i el valor per defecte serà nul.**

```
CREATE TABLE EDITORIAL(  
    ED_ID NUMBER,  
    ED_NAME VARCHAR2(50),  
    PRIMARY KEY (ED_ID));
```



Consideracions noms de les taules

- ❑ Les taules han de complir les següents regles:
 - Han de començar amb una lletra
 - No poden tenir més de 30 caràcters
 - Només es permet utilitzar lletres de l'alfabet anglès, nombres o el signe de subratllat (també el signe \$ i #, però aquests només es fan servir de manera especial pel qual no es recomana el seu ús)
 - No pot haver dues taules amb el mateix nom per al mateix usuari (poden coincidir els noms si hi són en distints esquemes)



Tipus de dades

Tipus de dada	Descripció
CHAR	Per textos de longitud fixa fins 2000 caràcters.
VARCHAR2(n)	Per textos de longitud variable fins 4000 caràcters.
BOOLEAN	Possibles valors true o false
DATE	Permet emmagatzemar dates. Les dates es poden escriure en format dia, mes i any entre cometes. El separador pot ser una barra de dividir, un guió i gairebé qualsevol símbol. Per emmagatzemar la data actual n'hi ha prou amb utilitzar la funció SYSDATE que retorna aquesta data
TIMESTAMP	És una extensió de l'anterior, emmagatzema valors de dia, mes i any, juntament amb hora, minut i segons (fins i tot amb decimals). Amb el que representa un instant concret en el temps. Un exemple de TIMESTAMP seria '2/2/2004 18:34:23,34521'. En aquest cas si el format de data i hora del sistema està pensat per a idioma espanyol, el separador decimal serà la coma (i no el punt).
NUMBER	El tipus NUMBER és un format versàtil que permet representar tot tipus de nombres. Els nombres decimals (nombres de coma fixa) s'indiquen amb NUMBER(p,s) , on p és la precisió màxima i s és l'escala (nombre de decimals a la dreta de la coma). Per exemple, NUMBER (8,3) indica que es representen nombres de vuit xifres de precisió i tres decimals. Els decimals en Oracle es presenta amb el punt i no amb la coma.



Resum

- ☐ Per crear una taula farem servir l'ordre CREATE TABLE.
- ☐ El seu format simple és:

```
CREATE TABLE nom taula (  
    Columna1 tipus_dada [NOT NULL],  
    Columna2 tipus_dada [NOT NULL],  
    .....  
) [TABLESPACE espai_de_taula];
```

- ☐ On:
 - ☐ ***Columna1, columna2*** són els noms de les columnes que contindran cada fila.
 - ☐ ***Tipus_dada***, indica el tipus de dada (VARCHAR2, NUMBER, etc) de cadascuna de les columnes.
 - ☐ ***TABLESPACE***, indica el TABLESPACE per emmagatzemar la taula. (schema)
 - ☐ ***NOT NULL***, indica que la columna ha de contenir alguna informació; mai pot ser nul·la.



Exercici

- ☐ Crea una connexió nova al teu schema Oracle que s'anomeni prova.
- ☐ Dins de prova crearem una taula nova amb el següent format:

Nom Taula		Alumne	
Columna	Nul?	Tipus Dades	Comentaris
Nom_matricula	NUMBER (6)	NOT NULL	
Nom	VARCHAR2 (15)	NOT NULL	
Data_naixament	DATE		
Adreça	VARCHAR2 (20)		
Localitat	VARCHAR2 (15)		



Exercici – Solució

```
CREATE TABLE ALUMNE (  
  num_matricula number(6) not null,  
  nom varchar2(15) not null,  
  data_naixement date,  
  adreça varchar2(30),  
  LOCALITAT varchar2(15)  
);
```



Creació d'una taula amb dades recuperades d'una consulta

- ☐ La sentència CREATE TABLE permet crear una taula a partir de la consulta d'una altra taula ja existent.
- ☐ La nova taula contindrà les dades obtingudes en la consulta.
- ☐ Es duu a terme aquesta acció amb la clàusula AS col·locada al final de l'ordre CREATE TABLE.

- ☐ **Sintaxi:**

```
CREATE TABLE nomTaula (  
  
Columna1 TIPUS_DE_DADA [NOT NULL],  
Columna2 TIPUS_DE_DADA[NOT NULL],  
Columna3 TIPUS_DE_DADA[NOT NULL],  
) [TABLESPACE espai_de_taula]  
AS consulta;
```

- ☐ **Exemple :**

```
CREATE TABLE EMPLOYEE2  
AS SELECT E.EMPLOYEE_ID, E.LAST_NAME, D.DEPARTMENT_NAME  
FROM EMPLOYEE E, DEPARTMENT D WHERE E.DEPARTMENT_ID= D.DEPARTMENT_ID;
```



2. CONSULTAR TAULA D'USUARI

- ☐ Els usuaris poden consultar les taules creades mitjançant la vista **USER_TABLES**
- ☐ Aquesta vista conté informació de les taules:
 - Nom de la taula
 - Nom del tablespace
 - Nom de files
 - Informació d'emmagatzematge
 - etc
- ☐ Per visualitzar el nom de les taules creades podem fer servir l'ordre:

```
SELECT TABLE_NAME FROM USER_TABLES;
```

- ☐ Existeixen altres dos vistes que permeten obtenir informació dels objectes que són propietat de l'usuari:
 - ☐ **USER_OBJECTS** → Objectes que són propietat de l'usuari.
 - ☐ **USER_CATALOG** → taules, vistes, sinònims i seqüències propietat de l'usuari.



Activitats proposades

- ☐ Investiga les columnes **USER_TABLES** i fer-les servir per consultar el nom de les taules que tens i el nom de files que té cadascuna.

- ☐ Investiga les columnes **USER_OBJECTS** i **USER_CATALOG** i utilitza les vistes per consultar les taules que tens.



Activitats proposades - Solució

- ❑ Investiga les columnes **USER_TABLES** i fer-les servir per consultar el nom de les taules que tens i el nom de files que té cadascuna.

*select * from user_tables;*

DESCRIBE USER_TABLES;

- ❑ Investiga les columnes **USER_OBJECTS** i **USER_CATALOG** i utilitza les vistes per consultar les taules que tens.

*select * from user_OBJECTS;*

*select * from user_catalog;*

*SELECT * FROM EMP_DETAILS_VIEW;*



3. INTEGRITAT

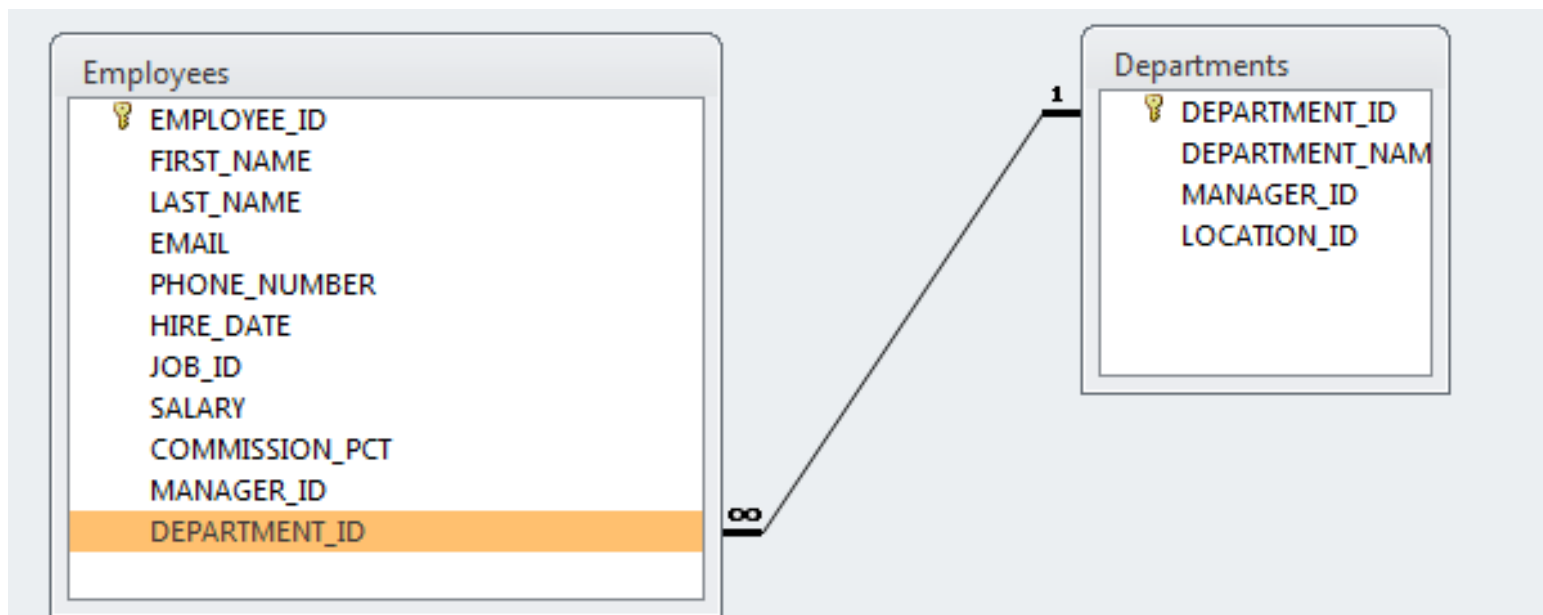
Integritat de dades

- ☐ Quan emmagatzem dades a les nostres taules, s'ajusten a una sèrie de restriccions pre-establertes.
- ☐ Per exemple, que una columna no pugui tenir valors negatius, que una cadena de caràcters s'emmagatzemi en minúscules o que una columna no pugui valer 0.
- ☐ La **integritat** fa referència al fet que les dades d'una BD han d'ajustar-se a restriccions abans d'emmagatzemar-se dins d'aquesta.
- ☐ Una **restricció d'integritat** és una regla que restringeix el rang de valors per una o més columnes dins una taula.
- ☐ Si es produeix qualsevol errada mentre un usuari està fent un canvi a les dades de la BD, aquesta té la capacitat de desfer o cancel·lar qualsevol acció sospitosa.



Integritat Referencial

- ❑ La **integritat referencial** garanteix que els valors d'una columna (o columnes) d'una taula (**clau forana**) depenguin dels valors d'altra columna (o columnes) d'altre taula (**clau primària**).
- ❑ Prenen com exemple l'schema HR.
 - La taula employees té una relació amb la taula Departments. Si definim integració referencial per aquestes taules, mai és donarà la situació d'inserir un empleat amb un department que no existeixi dins la taula departments.





4. RESTRICCIONS

- ❑ L'**objectiu d'una restricció** és que les aplicacions o els usuaris que manipulen les dades tinguin menys treball, i que sigui Oracle el que realitzi la major part de les tasques de manteniment de la integritat de la BD.

- ❑ Una **restricció d'integritat** és una regla que restringeix el rang de valors per una o més columnes dins una taula.

- ❑ L'ordre **CREATE TABLE** permet definir diferents tipus de restriccions dins una taula:
 - Claus primàries
 - Claus foranes
 - Obligatorietat
 - Valors per defecte
 - Verificació de condicions.



4. RESTRICCIONS

- ☐ Per definir les **restriccions** dins l'ordre **CREATE TABLE** fem servir la **clàusula CONSTRAINT**.
- ☐ Amb aquesta clàusula podem restringir una columna (restricció columna) o un grup de columnes (restricció de taula).
- ☐ Podem especificar les restriccions:
 - **Restricció d'una columna** → part d'una definició d'una columna.
 - **Restricció d'una taula** → una vegada especificades totes les columnes.
- ☐ **Sintaxis:**

```
CREATE TABLE nomTaula (
```

```
Columna1 TIPUS_DADA [CONSTRAINT nom_restricció] [NOT NULL] [UNIQUE] [PRIMARY KEY] [DEFAULT valor] [REFERENCES nom_taula [(columna [, columna])] [ON DELETE CASCADE]] [CHECK (condició)],
```

```
Columna2 TIPUS_DADA [CONSTRAINT nom_restricció] [NOT NULL] [UNIQUE] [PRIMARY KEY] [DEFAULT valor] [REFERENCES nom_taula [(columna [, columna])] [ON DELETE CASCADE]] [CHECK (condició)],
```

```
...
```

```
) [TABLESPACE espai_de_taula];
```

4. RESTRICCIONS

❑ Exemple:

```
CREATE TABLE EMPLEAT (  
    NOM VARCHAR2(25) PRIMARY KEY,  
    EDAT NUMBER CHECK (EDAD BETWEEN 18 AND 35),  
    COD_PROVINCIA NUMBER(2) REFERENCES PROVINCIAS ON DELETE CASCADE);
```

Restriccions columnes

- ❑ A l'anterior exemple es defineixen les següents restriccions:
 - Clau primària → NOM
 - Clau forana → COD_PROVINCIA, que fa referència a la taula PROVINCIAS
 - Verificació de condició (CHECK) → l'edat ha d'estar compresa entre 18 i 35.



4. RESTRICCIONS

- ❑ Les restriccions dins la ordre CREATE TABLE que apareixen al final de la definició de les columnes (o d'una taula) es diferencien de l'anterior en que es poden fer referència a varies columnes en una única restricció

➤ *Podem declarar dos columnes com claus primàries o foranes.*

- ❑ **Sintaxi:**

```
CREATE TABLE nomTaula (  
  
Columna1 TIPUS_DE_DADA[NOT NULL],  
Columna2 TIPUS_DE_DADA[NOT NULL],  
Columna3 TIPUS_DE_DADA[NOT NULL],  
...  
[CONSTRAINT nom_restricció] {[UNIQUE] | [PRIMARY KEY] (columna[,columna])},  
[CONSTRAINT nom_restricció] [FOREIGN KEY (columna[,columna]) REFERENCES nom_taula  
[(columna[,columna])] [ON DELETE CASCADE]],  
[CONSTRAINT nom_restricció] [CHECK (condició)],  
...  
) [TABLESPACE espai_de_taula];
```

4. RESTRICCIONS

Cas Pràctic (1)

- ❑ Crearem les taules PROVINCIA i la taula EMPLEAT. Primer crearem la taula PROVINCIA ja que fa referència a la taula empleat:

```
CREATE TABLE PROVINCIA (  
COD NUMBER(2) PRIMARY KEY,  
NOM VARCHAR2(25) );
```

Restriccions columnes

```
CREATE TABLE EMPLEAT (  
NOM VARCHAR2(25) ,  
EDAT NUMBER,  
COD_PROVINCIA NUMBER(2) ,  
CONSTRAINT PK_EMPLEADO PRIMARY KEY (NOM) ,  
CONSTRAINT CK_EDAT CHECK(EDAT BETWEEN 18 AND 35) ,  
CONSTRAINT FK_EMPLEADO FOREIGN KEY (COD_PROVINCIA) REFERENCES PROVINCIA ON DELETE CASCADE);
```

Restriccions d'una taula

- ❑ El nom de les constraints són opcionals, però està bé documentar-les perquè així les podem trobar dins el diccionari de la BD.



4. RESTRICCIONS

Clau primària. Restricció PRIMARY KEY

- ❑ Una **clau primària** dins d'una taula és una columna o un conjunt de columnes que identifiquen unívocament a cada fila.
- ❑ Ha de ser **única, no nul·la i obligatòria**.
- ❑ Aquesta clau es pot referenciar per una columna o columnes d'una altra taula.
- ❑ Anomenarem **clau forana** a aquesta columna o columnes.
- ❑ Quan es crea una clau primària, automàticament es crea un índex que facilita l'accés a la taula.
- ❑ Per definir una clau primària en una taula usem la restricció **PRIMARY KEY**.
- ❑ **Sintaxi:**

Sintaxi restricció columna	Sintaxi restricció taula
<pre>CREATE TABLE nom_taula (Columna1 TIPUS_DADA [CONSTRAINT nomrestricció] PRIMARY KEY, Columna2 TIPUS_DADA, ...) [TABLESPACE espai_de_taula];</pre>	<pre>CREATE TABLE nom_taula (Columna1 TIPUS_DADA, Columna2 TIPUS_DADA, ... [CONSTRAINT nom_restricció] PRIMARY KEY (columna [,columna]), ...) [TABLESPACE espai_de_taula];</pre>



4. RESTRICCIONS

Cas Pràctic (2)

- ❑ Crearem la taula BLOCPISOS. Les columnes són les següents:

Nom Columna	Comentari	Tipus
CARRER	Carrer on està el bloc	VARCHAR2(30)
NOMBRE	Nombre on està el bloc	NUMBER(3)
PIS	Nombre de planta	NUMBER(2)
PORTA	Porta	CHAR(1)
CP	Codi Postal	NUMBER(5)
METRES	Metres de l'habitatge	NUMBER(5)
COMENTARIS	Altres dades de l'habitatge	VARCHAR2(60)
COD_ZONA	Codi zona on està l'habitatge	NUMBER(2)
DNI	DNI propietari	VARCHAR2(10)



4. RESTRICCIONS

Cas Pràctic (II)

- ❑ La clau primària està formada per les columnes CARRER, NOMBRE, PIS i PORTA, això vol dir que no pot contenir valors nuls. Podem crear la taula de la següent manera:

```
CREATE TABLE BLOCPISOS (  
  CARRER VARCHAR2(30) NOT NULL,  
  NOMBRE NUMBER(3) NOT NULL,  
  PIS NUMBER(2) NOT NULL,  
  PORTA CHAR(1) NOT NULL,  
  CP NUMBER(5),  
  METRES NUMBER(5),  
  COMENTARIS VARCHAR2(60),  
  COD_ZONA NUMBER(2),  
  DNI VARCHAR2(10),  
  CONSTRAINT PK_HAB PRIMARY KEY (CARRER, NOMBRE, PIS, PORTA));
```

- ❑ La clau primària d'aquesta taula és la combinació de les columnes carrer, nombre, pis i porta, que estan identificades com NOT NULL.
- ❑ Si definim una columna como PK i no identifiquem aquesta columna com NOT NULL, Oracle automàticament col·loca NOT NULL en aquest atribut.



4. RESTRICCIONS

Cas Pràctic (III)

- ❑ Creem ara, la taula ZONA. Les columnes de la taula són:

Nom Columna	Comentari	Tipus
COD_ZONA	Codi Zona	NUMBER(2)
NOM_ZONA	Nom de la zona	VARCHAR2(20)
AA_ZONA	Altres dades de la zona	VARCHAR2(50)

- ❑ La PK és el codi de la zona (COD_ZONA) i la definim formant part de la columna (restricció columna):

```
CREATE TABLE ZONA (  
  COD_ZONA NUMBER(2) PRIMARY KEY,  
  NOM_ZONA VARCHAR2(15) NOT NULL,  
  AA_ZONA VARCHAR2(60) );
```

- ❑ Quan fem servir el format de restricció de columna, la clàusula PRIMARY KEY només pot aparèixer una vegada dins la definició de la taula.



4. RESTRICCIONS

Clau forana. Restricció FOREIGN KEY

- ❑ Una **clau forana** està formada per una o diverses columnes que estan associades a una clau primària d'una altra o de la mateixa taula.
- ❑ Es poden definir tantes claus foranes com calgui, i poden estar o no en la mateixa taula que la clau primària.
- ❑ El valor de la columna o columnes que són claus foranes ha de ser NULL o igual a un valor de la clau referenciada (**regla d'integritat referencial**).
- ❑ **Sintaxi:**

Sintaxi restricció columna	Sintaxi restricció taula
<pre>CREATE TABLE nom_taula (Columna1 TIPUS_DADA [CONSTRAINT nomrestricció] REFERENCES nom_taula [(columna)] [ON DELETE CASCADE], Columna2 TIPUS_DADA, ...) [TABLESPACE espai_de_taula];</pre>	<pre>CREATE TABLE nom_taula (Columna1 TIPUS_DADA, Columna2 TIPUS_DADA, ... [CONSTRAINT nom_restricció] FOREIGN KEY (columna [,columna]), REFERENCES nomtaula [(columna [,columna]))[ON DELETE CASCADE], ...) [TABLESPACE espai_de_taula];</pre>



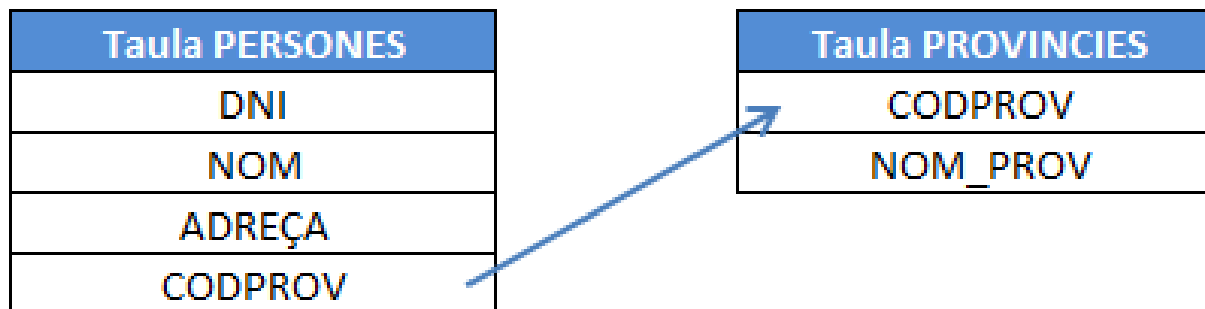
4. RESTRICCIONS

Clau forana. Restricció FOREIGN KEY (II)

- ☐ En la clàusula **REFERENCES** indiquem la taula a la qual remet la clau forana.
- ☐ La dreta de **FOREIGN KEY** i, entre parèntesi, indiquem la columna o columnes que formen part de la clau forana.
- ☐ La clàusula **ON DELETE CASCADE** o esborrat en cascada es defineix quan en esborrar les files associades amb claus primàries desitgem que s'eliminïn automàticament les files amb claus alienes que referenciïn a aquestes claus.

Cas Pràctic (3)

- ☐ Siguin les taules PERSONES i PROVÍNCIES. La taula PERSONES conté dades sobre les persones d'una comunitat, mentre que la taula PROVÍNCIES conté el codi i nom de cada província, es relacionen per l'atribut COD_PROVIN:





4. RESTRICCIONS

Cas Pràctic

☐ On:

- DNI és la clau primària de la taula PERSONES.
- CODPROV de la taula PROVÍNCIES és clau primària d'aquesta taula.
- CODPROV de la taula PERSONES és clau forana, perquè es relaciona amb la clau primària de la taula PROVÍNCIES.
 - ✓ Els valors que s'emmagatzemin en aquesta columna han de coincidir amb la clau primària de la taula PROVÍNCIES. Es pot afirmar que PROVÍNCIES és la taula mestra i PERSONES és la taula detall.

☐ Hem de crear, en primer lloc, la taula PROVÍNCIES i, després, la taula PERSONES, ja que PERSONES referència a PROVÍNCIES.

☐ Si creem primer la taula PERSONES i la taula PROVÍNCIES no està creada, Oracle emetrà un missatge d'error.



4. RESTRICCIONS

Cas Pràctic

```
CREATE TABLE PROVINCIAS (  
  CODPROV NUMBER(2) PRIMARY KEY,  
  NOM_PROV VARCHAR2(15)) ;
```

```
CREATE TABLE PERSONES (  
  DNI NUMBER(8) PRIMARY KEY,  
  NOM VARCHAR2(15) ,  
  ADREÇA VARCHAR2(25) ,  
  POBLACIO VARCHAR2(20) ,  
  CODPROV NUMBER(2) NOT NULL REFERENCES PROVINCIAS) ;
```

- ☐ La clau forana s'ha definit usant la clàusula REFERENCES, encara que també es pot definir usant la clàusula FOREIGN KEY de la següent manera:

```
CREATE TABLE PERSONES (  
  DNI NUMBER(8) PRIMARY KEY,  
  NOM VARCHAR2(15) ,  
  ADREÇA VARCHAR2(25) ,  
  POBLACIO VARCHAR2(20) ,  
  CODPROV NUMBER(2) NOT NULL,  
  FOREIGN KEY (CODPROVIN) REFERENCES PROVINCIAS) ;
```



4. RESTRICCIONS

Cas Pràctic

- ☐ Imagina el cas anterior. Partim de la situació de que les dos taules tenen dades. Volem esborrar totes les files de la taula PROVINCIA.

```
DELETE PROVINCIAS;  
*  
ERROR en línea 1:  
ORA-02292: restricción de integridad (SCOTT.SYS_C005389) violada - registro secun-  
dario encontrado
```

- ☐ Es produeix un error: no podem esborrar files en la taula mestra (PROVINCIES) si hi ha files en la taula detall (PERSONES) que les estiguin referenciant.
- ☐ És a dir, una fila no es pot esborrar si és referenciada per alguna clau forana.
- ☐ Si s'afegeix la clàusula **ON DELETE CASCADE** en l'opció REFERENCES de la taula detall (PERSONES) es podran eliminar les files de la taula mestra (PROVINCIES) i les files corresponents en la taula detall (PERSONES) amb aquesta província seran eliminades.
- ☐ Per modificar la taula, esborren la taula PERSONES de la següent manera:

```
DROP TABLE PERSONES;
```



4. RESTRICCIONS

Cas Pràctic

- ☐ Tornem a crear la taula PERSONES:

```
CREATE TABLE PERSONES (  
  DNI NUMBER(8) PRIMARY KEY,  
  NOM VARCHAR2(15),  
  ADREÇA VARCHAR2(25),  
  POBLACIO VARCHAR2(20),  
  CODPROV NUMBER(2) NOT NULL,  
  FOREIGN KEY (CODPROVIN) REFERENCES PROVINCIAS(CODPROV) ON DELETE CASCADE;
```

- ☐ Una vegada creada la taula inserim files i esborrem una fila de la taula mestra (PROVINCIES).
- ☐ Automàticament s'esborraran les files de la taula detall que es corresponguin amb les files de la taula mestra.
- ☐ Aquesta acció manté automàticament la integritat referencial.



8. RESTRICCIONS

Resum - On Delete

Comanda	Descripció	Opcions
ON DELETE	Indica que ocurrerà quan s'intenti esborrar en la taula referida una fila la clau primària de la qual apareix com valor de clau forana en alguna fila de la taula actual. Defineix cinc opcions.	<ol style="list-style-type: none">1) No action, no fer res a la taula actual, però esborra la fila de la clau forana, el que pot provocar problemes de falta d'integritat a les dades.2) Set null, , s'assigna el valor null totes les claus foranes relacionades amb la clau primària que es desitja esborrar.3) Set default, s'assigna el valor per defecte als camps que formen la clau forana relacionada amb la clau primària que es desitja esborrar i posteriorment s'esborra aquesta fila.4) Cascade, s'esborren tots els registres per la qual la clau forana és igual que la clau primària del registre esborrat.5) Restrict, si existeix alguna fila que té com a clau forana el valor de la clau primària que es desitja esborrar a la taula referida, la fila associada no s'esborra.

- ☐ Si no s'indiquen aquestes clàusules no es poden ni actualitzar ni esborrar el registres relacionats.



4. RESTRICCIONS

Restricció NOT NULL.

- ☐ Aquesta restricció associada a una columna significa que no pot tenir valors nuls, és a dir, que ha de tenir obligatòriament un valor.
- ☐ En cas contrari, causa una excepció. En exemples anteriors ens hem ocupat de com es defineix una columna amb la restricció NOT NULL.

Valors per defecte. L'especificació DEFAULT.

- ☐ Al moment de crear una taula podem assignar valors per defecte a les columnes.
- ☐ Si especifiquem la clàusula DEFAULT a una columna, li proporcionem un valor per omissió quan el valor de la columna no s'especifica en la clàusula INSERT.
- ☐ En l'especificació DEFAULT és possible incloure diverses expressions: constants, funcions SQL i variables UID i SYSDATE.
- ☐ No es pot fer referències a columnes o a funcions PL/SQL.



4. RESTRICCIONS

Cas pràctic (4)

- ❑ Crearem la taula Exemple01 i assignarem a la columna DATA, la data del sistema.

```
CREATE TABLE EXEMPLE01 (  
  DNI VARCHAR2(10) NOT NULL,  
  NOM VARCHAR(30) ,  
  DATA DATE DEFAULT SYSDATE) ;
```

- ❑ S'insereix una fila en la taula donant valors a totes les columnes, excepte a la columna DATA:
INSERT INTO EXEMPLE01(DNI, NOM) VALUES('1234','PEPA');
- ❑ En visualitzar el contingut de la taula, en la columna DATA s'emmagatzemarà la data del sistema, ja que no es va donar valor.



4. RESTRICCIONS

Verificació de condicions. La restricció CHECK

- ❑ Moltes columnes de taules requereixen valors limitats dins d'un rang o el compliment de certes condicions.
- ❑ Amb una restricció de verificació de condicions es pot expressar una condició que ha de complir-se para totes i cadascuna de les files de la taula.
- ❑ La restricció CHECK actua com una clàusula WHERE. Pot fer referència a una o a més columnes, però no a valors d'altres files.
- ❑ En una clàusula CHECK no cal incloure subconsultes ni les pseudocolumnes SYSDATE, UID i USER.
- ❑ **Sintaxis:**

Sintaxi restricció columna	Sintaxi restricció taula
<pre>CREATE TABLE nom_taula (Columna1 TIPUS_DADA [CONSTRAINT nomrestricció] CHECK (condició), Columna2 TIPUS_DADA, ...) [TABLESPACE espai_de_taula];</pre>	<pre>CREATE TABLE nom_taula (Columna1 TIPUS_DADA, Columna2 TIPUS_DADA, ... [CONSTRAINT nom_restricció] CHECK (condició), ...) [TABLESPACE espai_de_taula];</pre>



4. RESTRICCIONS

Cas Pràctic (5)

- ☐ Es crea la taula EXEMPL02. Les columnes són: DNI VARCHAR2(10), NOM VARCHAR2(30), EDAT NUMBER(2), CURS NUMBER; i les restriccions:
 - El DNI no pot ser nul.
 - La clau primària és el DNI.
 - El NOM no pot ser nul.
 - L'EDAT ha d'estar compresa entre 5 i 20 anys.
 - El NOM ha d'estar en majúscules.
 - El CURS només pot emmagatzemar 1, 2 o 3.
- ☐ És possible crear la taula de diverses maneres: es pot donar nom o no a les restriccions, definir les restriccions en la descripció de la columna o al final, combinant ambdues, etcètera:

```
CREATE TABLE EXEMPLE02 (  
  DNI VARCHAR2(10) NOT NULL,  
  NOM VARCHAR2(30) NOT NULL,  
  EDAT NUMBER(2) ,  
  CURS NUMBER,  
  CONSTRAINT CLAU_P PRIMARY KEY(DNI) ,  
  CONSTRAINT COMP_EDAT CHECK (EDAT BETWEEN 5 AND 20) ,  
  CONSTRAINT NOM_MAJUS CHECK (NOM = UPPER(NOM)) ,  
  CONSTRAINT COMP_CURS CHECK (CURS IN(1, 2, 3))  
);
```



4. RESTRICCIONS

Cas Pràctic

- ❑ Sense donar nom a les restriccions i definint-les en la descripció de les columnes, tenim:

```
CREATE TABLE EXEMPLE02 (  
  DNI VARCHAR2(10) NOT NULL PRIMARY KEY,  
  NOM VARCHAR2(30) NOT NULL CHECK (NOM = UPPER(NOM)) ,  
  EDAT NUMBER(2) CHECK (EDAT BETWEEN 5 AND 20) ,  
  CURS NUMBER CHECK (CURS IN(1, 2, 3))  
);
```



4. RESTRICCIONS

Activitat

❑ Crea les següents taules:

Taula FABRICANT	
COD_FABRICANT	NUMBER(3)
NOM	VARCHAR2(15)
PAIS	VARCHAR2(15)

Taula ARTICLES	
ARTICLE	VARCHAR2(20)
COD_FABRICANT	NUMBER(3)
PES	NUMBER(3)
CATEGORIA	VARCHAR2(10)
PREU_VENDA	NUMBER(6,2)
PREU_COST	NUMBER(6,2)
STOCK	NUMBER(5)

❑ Restriccions per a la taula FABRICANTS:

- La clau primària és COD_FABRICANT.
- Les columnes NOM i PAIS han d'emmagatzemar-se en majúscula.

❑ Restriccions per a la taula ARTICLES:

- La clau primària està formada per les columnes: ARTICLE, COD_FABRICANT, PES I CATEGORIA.
- COD_FABRICANT és clau forana que referencia a la taula FABRICANT.
- PREU_VENDA, PREU_COST i PES han de ser > 0.
- CATEGORIA ha de ser 'Primera', 'Segona' o 'Tercera'.



4. RESTRICCIONS

Restricció UNIQUE

- ❑ Evita valors repetits en la mateixa columna. Pot contenir una o diverses columnes.
- ❑ És similar a la restricció PRIMARY KEY, tret que són possibles diverses columnes UNIQUE definides en una taula.
- ❑ Admet valors NULL.
- ❑ Igual que en PRIMARY KEY, quan es defineix una restricció UNIQUE es crea un índex automàticament.
- ❑ **Sintaxi:**

Sintaxi restricció columna	Sintaxi restricció taula
<pre>CREATE TABLE nom_taula (Columna1 TIPUS_DADA [CONSTRAINT nomrestricció] UNIQUE, Columna2 TIPUS_DADA, ...) [TABLESPACE espai_de_taula];</pre>	<pre>CREATE TABLE nom_taula (Columna1 TIPUS_DADA, Columna2 TIPUS_DADA, ... CONSTRAINT nom_restricció] UNIQUE (columna [, columna]), ...) [TABLESPACE espai_de_taula];</pre>



4. RESTRICCIONS

Vistes del diccionari de dades per a les restriccions

- ❑ Existeix una sèrie de vistes creades per Oracle que contenen informació referent a les restriccions definides en les taules. Contenen informació general les següents:
 - USER_CONSTRAINTS: definicions de restriccions de taules propietat de l'usuari.
 - ALL_CONSTRAINTS: definicions de restriccions sobre taules a les quals pot accedir l'usuari.
 - DBA_CONSTRAINTS: totes les definicions de restriccions sobre totes les taules.

- ❑ Per a informació sobre restriccions en les columnes tenim:
 - USER_CONS_COLUMNS: informació sobre les restriccions de columnes en taules de l'usuari.
 - ALL_CONS_COLUMNS: informació sobre les restriccions de columnes en taules a les quals pot accedir l'usuari.
 - DBA_CONS_COLUMNS: informació sobre totes les restriccions de columnes.



4. RESTRICCIONS

Vistes del diccionari de dades per a les restriccions

- ❑ Existeix una sèrie de vistes creades per Oracle que contenen informació referent a les restriccions definides en les taules. Contenen informació general les següents:
 - USER_CONSTRAINTS: definicions de restriccions de taules propietat de l'usuari.
 - ALL_CONSTRAINTS: definicions de restriccions sobre taules a les quals pot accedir l'usuari.
 - DBA_CONSTRAINTS: totes les definicions de restriccions sobre totes les taules.

- ❑ Per a informació sobre restriccions en les columnes tenim:
 - USER_CONS_COLUMNS: informació sobre les restriccions de columnes en taules de l'usuari.
 - ALL_CONS_COLUMNS: informació sobre les restriccions de columnes en taules a les quals pot accedir l'usuari.
 - DBA_CONS_COLUMNS: informació sobre totes les restriccions de columnes.



4. RESTRICCIONS

Desactivar restriccions

- ☐ A vegades necessitem desactivar restriccions de manera temporal per saltar-se les regles que imposa.

- ☐ **Sintaxis:**

```
ALTER TABLE taula  
DISABLE CONSTRAINT nom [CASCADE]
```

- ☐ L'opció CASCADE fa que es desactivin també les restriccions defenents de la que és va desactivar.

Activar restriccions

- ☐ Anul·la la desactivació.

- ☐ **Sintaxis:**

```
ALTER TABLE taula  
ENABLE CONSTRAINT nom [CASCADE]
```

- ☐ Només es permet tornar a activar si els valors de la taula compleixen la restricció que s'activa.
- ☐ Si es va desactivar en cascada, caldrà activar cada restricció individualment.



4. RESTRICCIONS

Canviar els noms a les restriccions

☐ Sintaxis:

```
ALTER TABLE taula RENAME CONSTRAINT nomvell TO nomnou;
```

Modificar/afegir/Esborrar restriccions

☐ Sintaxis:

```
ALTER TABLE nomtaula  
ADD CONSTRAINT nomconstraint CONSTRAINT clause { PRIMARY KEY | FOREIGN KEY |  
UNIQUE | CHECK } depenen del tipus de restricció farem servir el mateix que a les restriccions  
de taula.  
...  
ADD CONSTRAINT nomconstraint CONSTRAINT clause { PRIMARY KEY | FOREIGN KEY |  
UNIQUE | CHECK } depenen del tipus de restricció farem servir el mateix que a les restriccions  
de taula.  
;
```



5. MODIFICAR TAULES

Canviar noms

- ☐ L'ordre **RENAME** permet el canvi de nom de qualsevol objecte.

```
RENAME nom_antig TO nom_nou;
```

Esborrar continguts de taules

- ☐ L'ordre **TRUNCATE TABLE** seguida del nom d'una taula, fa que s'elimini el contingut de la taula, però no la taula en si. Fins i tot esborra de l'arxiu de dades l'espai ocupat per la taula.
- ☐ Aquesta ordre no pot anul·lar-se amb un **ROLLBACK**



5. MODIFICAR TAULES

Modificar taules

- ❑ L'ordre **ALTER TABLE** permet fer canvis a l'estructura d'una taula.

Comanda	Descripció	Sintaxis
Afegir columnes	Permet afegir columnes a la taula. S'ha d'indicar el tipus de dada i les seves propietats si és necessari. Les noves columnes s'afegeixen al final, no es pot indicar la posició.	ALTER TABLE <i>nomdeTaula</i> ADD (<i>nomColumna TipusDada</i> [<i>Propietats</i>] [, <i>columnaSegüent tipusDades</i> [<i>propietats</i>]...)
Esborrar columnes	Elimina la columna indicada de manera irreversible i incloent les dades que contenia. No es pot eliminar l'última columna (caldrà usar DROP TABLE per esborrar tota la taula).	ALTER TABLE <i>nomTaula</i> DROP (<i>columna</i>);
Modificar columnes	Permet canviar el tipus de dades i propietats d'una determinada columna. Els canvis permesos són: <ul style="list-style-type: none">•Incrementar precisió o amplària dels tipus de dades.•Només es pot reduir l'amplària, si aquesta columna posseeix nuls en tots els registres, o no hi ha registres.•Es pot passar de CHAR a VARCHAR2 i viceversa (si no es modifica l'amplària).•Es pot passar de DATE a TIMESTAMP i viceversa	ALTER TABLE <i>nomTaula</i> MODIFY (<i>columna</i> <i>tipus</i> [<i>propietats</i>] [, <i>columnaSegüent tipus</i> [<i>propietats</i>] ...])



5. MODIFICAR TAULES

Modificar taules

Comanda	Descripció	Sintaxis
Inhabilitar columnes	<p>Aquesta opció es fa servir per macar una o més columnes com no utilitzades.</p> <ul style="list-style-type: none">•S'ha de tenir en compte que la columna no s'elimina físicament.•Una consulta SELECT * no recupera les dades.•Amb la comanda DESCRIBE tampoc apareix les columnes no utilitzades.•Si es fa servir aquesta comanda no podrem tornar activar aquesta columna. Lo millor es crear una copia de la taula i desactivar les columnes per aquesta nova taula.	<pre>ALTER TABLE <i>nomTaula</i> SET UNUSED (<i>columna</i>); O ALTER TABLE <i>nomTaula</i> SET UNUSED COLUM (<i>columna</i>);</pre>



5. MODIFICAR TAULES

Modificar taules - exemples

- ☐ Afegir una columna nova a una taula
- ☐ Afegir varies columnes a la taula

```
ALTER TABLE employees  
ADD last_name VARCHAR(50);
```

```
ALTER TABLE MY_EMPLOYEE  
ADD INFORMACIO VARCHAR2(255)  
ADD DESCRIPCIO VARCHAR2(10);
```

- ☐ Modificar una columna:

```
ALTER TABLE MY_EMPLOYEE  
MODIFY INFORMACIO NUMBER(5,7) DEFAULT 0.0  
MODIFY DESCRIPCIO NUMBER (5,1) UNIQUE;
```

- ☐ Esborrar una columna:

```
ALTER TABLE MY_EMPLOYEE  
DROP (INFORMACIO, DESCRIPCIO);
```



5. MODIFICAR TAULES

Afegir comentaris a les taules

- ☐ Se li poden posar comentaris a les taules i les columnes.
- ☐ Un comentari és un text descriptiu utilitzat per documentar la taula.
- ☐ Sintaxis:

```
COMMENT ON { TABLE NomTaula / COLUMN taula.nomColumna }  
IS 'Comentari'
```

- ☐ Per mostrar els comentaris posats s'usen les següents vistes del diccionari de dades mitjançant la instrucció SELECT:
 - ☐ **USER_TAB_COMMENTS**, comentaris de les taules de l'usuari actual.
 - ☐ **USER_COL_COMMENTS**, comentaris de les columnes de l'usuari actual.
 - ☐ **ALL_TAB_COMMENTS**, comentaris de les taules de tots els usuaris (només l'administrador)
 - ☐ **ALL_COL_COMMENTS**, comentaris de les columnes de tots els usuaris (només l'administrador)



6. ESBORRAR TAULES

- ☐ L'ordre **DROP TABLES** seguit del nom de la taula, permet eliminar la taula.
- ☐ A l'hora d'esborrar taules s'ha de tenir en compte:
 - Desapareixen totes les dades.
 - Qualsevol vista i sinònim referent a la taula seguiran existint, però ja no funcionaran (convé eliminar-los).
 - Només és possible realitzar aquesta operació si s'és el propietari de la taula o es posseeix el privilegi **DROP ANY TABLE**.
- ☐ **L'esborrat d'una taula és irreversible**, i no hi ha cap petició de confirmació, per la qual cosa convé ser molt acurat amb aquesta operació.



Preguntes!!!!