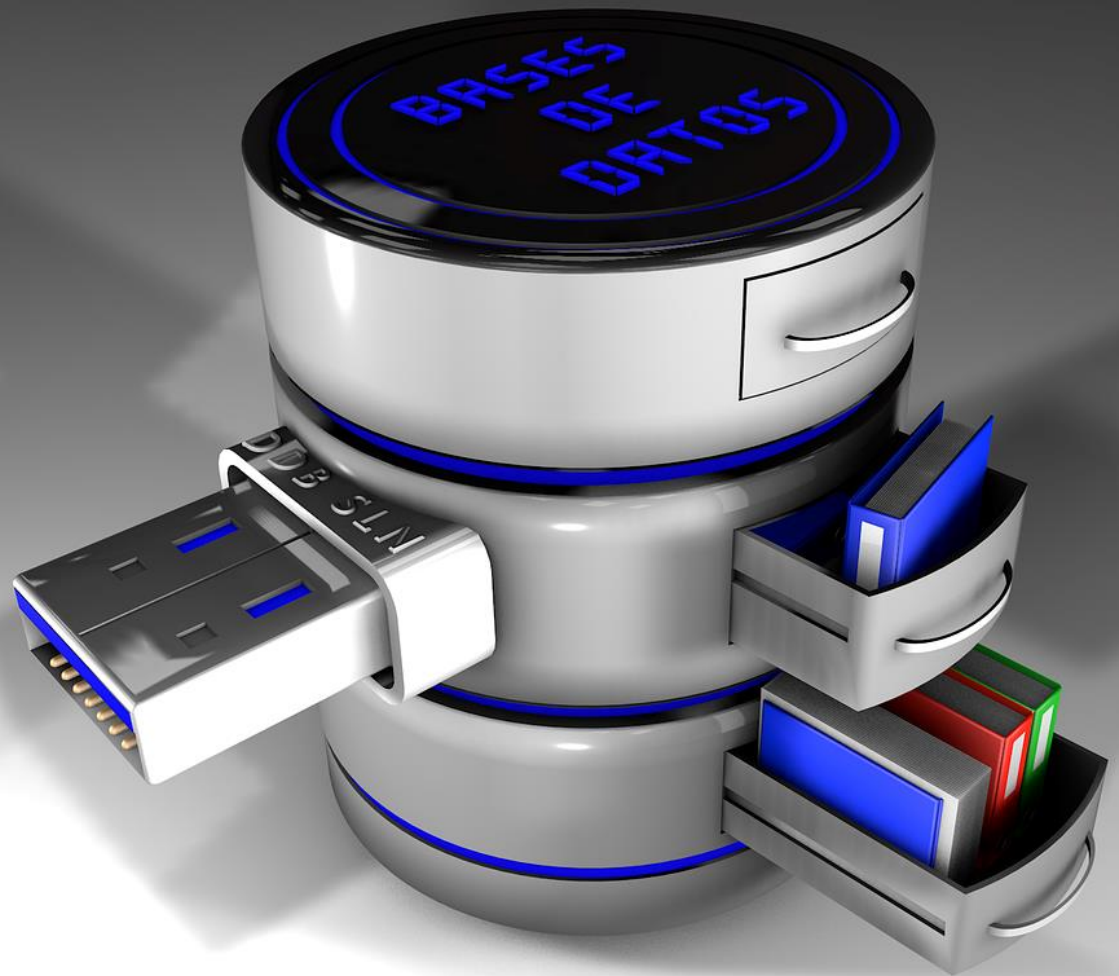


UNITAT DIDÀCTICA 11

SQL. Llenguatge DDL



Mòdul: Bases de Dades

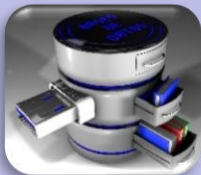
CFGs: Desenvolupament d'Aplicacions Multiplataforma

IES Serra Perenxisa (46019015)



Índex

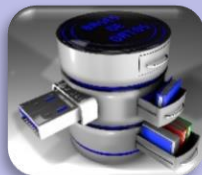
1. Nomenclatura de les taules
2. Sentència CREATE TABLE
3. Tipus de dades
4. Restriccions
5. Creació de taules mitjançant subconsultas
6. Sentència ALTER TABLE
7. Sentència DROP TABLE



Nomenclatura de les taules

Les taules i columnes de la base de dades s'han de nomenar segons les regles estàndard de nomenclatura d'objectes de Oracle Database:

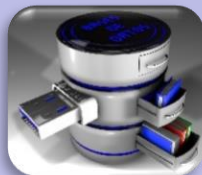
- Els noms de taula i de columna han de començar per una lletra i tindre entre 1 i 30 caràcters.
- Els noms només han de contindre els caràcters A–Z, a–z, 0–9, _ (subratllat), \$ i # (caràcters vàlids, però l'ús dels quals no és aconsellable).
- Els noms no han de ser un duplicat d'un altre nom d'objecte propietat del mateix usuari del servidor d'Oracle.
- Els noms no han de ser una paraula reservada del servidor d'Oracle.



Sentència CREATE TABLE

Pot crear taules per a emmagatzemar dades executant la sentència SQL CREATE TABLE. Aquesta sentència és una de les sentències DDL, que són un subjuego de sentències SQL que s'utilitzen per a crear, modificar o eliminar estructures d'Oracle Database. Aquestes sentències tenen un efecte immediat en la base de dades i registren informació en el diccionari de dades.

Per a crear una taula, un usuari ha de tindre el privilegi CREATE TABLE i una àrea d'emmagatzematge en la qual crear els objectes. L'administrador de la base de dades (DBA) utilitza sentències de llenguatge de control de dades (DCL) per a atorgar privilegis als usuaris.



Sentència CREATE TABLE

```
CREATE TABLE [schema.] table  
    (column datatype [DEFAULT expr] [, ...]);
```

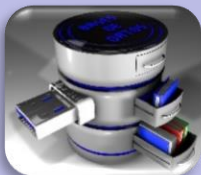
Schema és el mateix nom que el del propietari

Table és el nom de la taula

Column és el nom de la columna

Datatype és el tipus de dada i la longitud de la columna

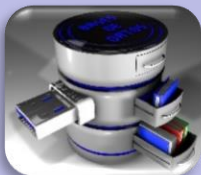
DEFAULT expr especifica un valor per defecte si s'omet un valor en la sentència INSERT



Sentència CREATE TABLE

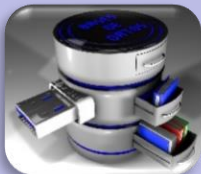
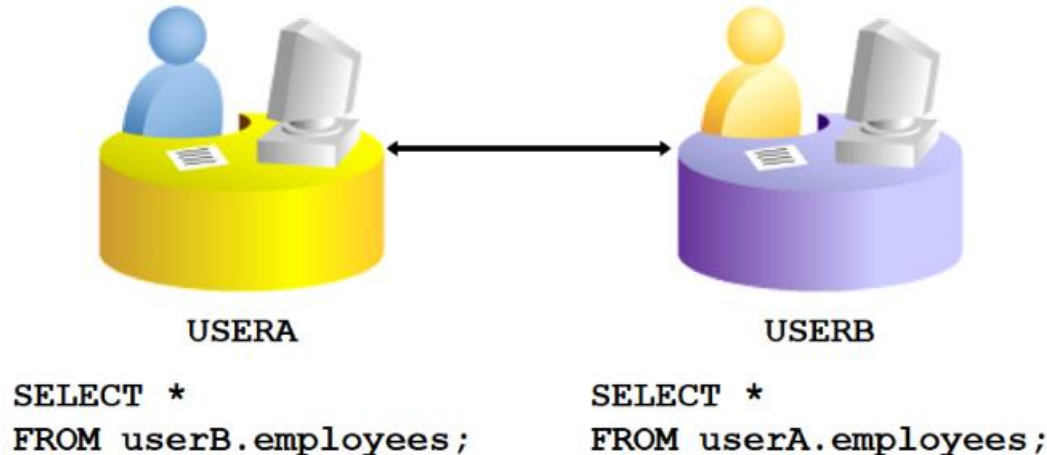
Un esquema és una recopilació d'estructures lògiques de dades o objectes d'esquema. Un esquema és propietat d'un usuari de base de dades i té el mateix nom que l'usuari. Cada usuari és propietari d'un únic esquema.

Els objectes d'esquema es poden crear i manipular amb SQL i inclouen taules, vistes, sinònims, seqüències, procediments emmagatzemats, índexs, clústers i enllaços de base de dades.



Sentència CREATE TABLE

Si una taula no pertany a l'usuari, el nom del propietari s'ha d'anteposar a la taula. Per exemple, donats els esquemes USERA i USERB i tots dos contenen la taula EMPLOYEES, si USERA desitja accedir a la taula EMPLOYEES que pertany a USERB, USERA ha d'anteposar el nom de l'esquema al nom de la taula



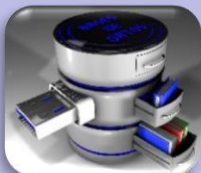
Sentència CREATE TABLE

Opció DEFAULT

Mitjançant l'opció DEFAULT proporcionem un valor per defecte a una columna.

Aquesta opció evita que s'introduïsquen valors nuls en les columnes si s'insereix una fila sense un valor per a la columna.

El valor per defecte pot ser un literal, una expressió o una funció SQL (com SYSDATE), però el valor no pot ser el nom d'una altra columna.



Sentència CREATE TABLE

Opció DEFAULT

Creació d'ua taula

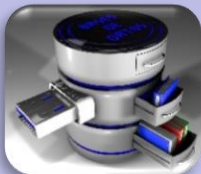
```
CREATE TABLE hire_dates(  
  id NUMBER(8),  
  hire_date DATE DEFAULT SYSDATE);
```

Table HIRE_DATES creado.

Confirmació de la creació de la taula

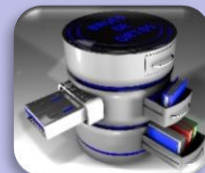
```
DESCRIBE hire_dates;
```

Nombre	Nulo	Tipo
-----	----	-----
ID		NUMBER (8)
HIRE_DATE		DATE



Tipus de dades

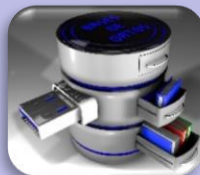
Tipo de Dato	Descripción
<code>VARCHAR2 (size)</code>	Datos de caracteres de longitud variable
<code>CHAR (size)</code>	Datos de caracteres de longitud fija
<code>NUMBER (p, s)</code>	Datos numéricos de longitud variable
<code>DATE</code>	Valores de fecha y hora
<code>LONG</code>	Datos de caracteres de longitud variable (hasta 2 GB).
<code>CLOB</code>	Datos binarios (hasta 4 GB).
<code>RAW and LONG RAW</code>	Datos binarios raw
<code>BLOB</code>	Datos binarios (hasta 4 GB)
<code>BFILE</code>	Datos binarios almacenados en un archivo externo (hasta 4 GB)
<code>ROWID</code>	Sistema numérico de base -64 que representa la dirección única de una fila en su tabla correspondiente.



Tipus de dades

Tipus de dades de data i hora

Tipo de Dato	Descripción
TIMESTAMP	Fecha con segundos fraccionarios
INTERVAL YEAR TO MONTH	Almacenados como un intervalo de años y meses
INTERVAL DAY TO SECOND	Almacenado como un intervalo de días, horas, minutos y segundos

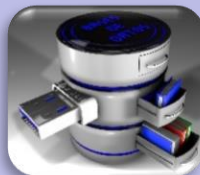


Restriccions

Oracle utilitza restriccions per a evitar la introducció de dades no vàlides en les taules.

Pot utilitzar restriccions per a realitzar el següent:

- ✓ Aplicar regles a les dades de la taula quan s'insereix, actualitza o suprimeix una fila d'aquesta. La restricció s'ha de complir perquè l'operació siga correcta.
- ✓ Evitar la supressió d'una taula si hi ha dependències d'altres taules.

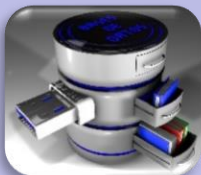


Restriccions

Totes les restriccions s'emmagatzemen en el diccionari de dades. És fàcil fer referència a les restriccions si els assigna un nom significatiu

El nom de la restricció no pot ser el mateix que el d'un altre objecte propietat del mateix usuari.

Si no assigna cap nom a la restricció, el servidor d'Oracle genera un nom amb el format SYS_Cn, on n és un enter perquè el nom de la restricció siga únic.



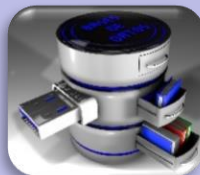
Restriccions

Les restriccions es poden definir en el moment de la creació de la taula o després d'haver-la creat.

Pot definir una restricció a **nivell de columna** o **de taula**. Funcionalment, una restricció a nivell de taula és la mateixa que una restricció a nivell de columna.

Les restriccions son:

- Primary key
- Foreign key
- Not null
- Unique
- Check



Restriccions

Sintaxis

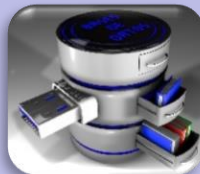
```
CREATE TABLE [schema.] table  
    (column datatype [DEFAULT expr]  
      [column_constraint],  
      ...  
      [table_constraint][, ...]);
```

Sintaxis a nivel de columna

```
column [CONSTRAINT constraint_name] constraint_type,
```

Sintaxis a nivel de taula

```
columna, ...  
    [CONSTRAINT constraint_name] constraint_type  
    (column, ...),
```



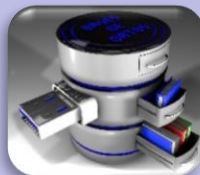
Restriccions

Les restriccions definides a **nivell de columna** s'inclouen en definir la columna.

Les restriccions a **nivell de taula** es defineixen al final de la definició de taula i ha de fer referència a la columna o les columnes a les quals pertany la restricció en un joc de parèntesi.

Es tracta principalment de la sintaxi que diferencia a les dues; en cas contrari, funcionalment, una restricció a nivell de columna és el mateix que una restricció a nivell de taula.

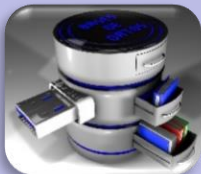
Les restriccions es creguen normalment al mateix temps que la taula. Encara que es poden agregar a una taula després de la seua creació i es poden desactivar temporalment.



Restriccions

Les restriccions NOT NULL s'han de definir a nivell de columna.

Las restricciones que se aplican a más de una columna se deben definir a nivel de tabla.



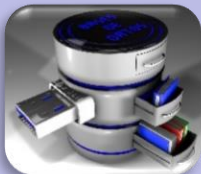
Restriccions

Exemple a nivell de columna

```
CREATE TABLE employees(  
    employee_id NUMBER(6)  
        CONSTRAINT emp_emp_id_pk PRIMARY KEY,  
    first_name VARCHAR2(20),  
    ...);
```

Exemple a nivell de taula

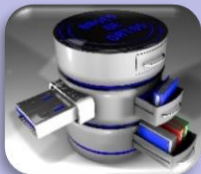
```
CREATE TABLE employees(  
    employee_id NUMBER(6),  
    first_name VARCHAR2(20),  
    ...  
    job_id VARCHAR2(10) NOT NULL,  
    CONSTRAINT emp_emp_id_pk  
        PRIMARY KEY (EMPLOYEE_ID));
```



Restriccions

NOT NULL

La restricció NOT NULL garanteix que la columna no conté cap valor nul. Les columnes sense la restricció NOT NULL poden contindre valors nuls per defecte. Les restriccions NOT NULL s'han de definir **a nivell de columna SEMPRE**.

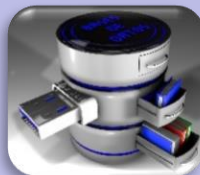


Restriccions

UNIQUE

Una restricció d'integritat UNIQUE necessita que tots els valors d'una columna o joc de columnes (clau) siguin únics; és a dir, que dues files d'una taula no poden tindre valors duplicats en una columna o joc de columnes concret. La columna (o joc de columnes) inclosa en la definició de la restricció de clau UNIQUE es denomina clau única. Si la restricció UNIQUE comprén més d'una columna, el grup de columnes es denomina **clau única composta**.

Les restriccions UNIQUE permeten l'entrada de valors nuls llevat que definisca també les restriccions NOT NULL per a les mateixes columnes. Els valors nuls no es consideren iguals a cap altre element



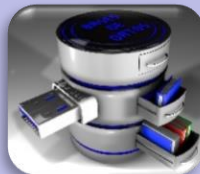
Restriccions

PRIMARY KEY

La restricció PRIMARY KEY crea una clau primària per a la taula. Només es pot crear una clau primària per a cada taula.

La restricció PRIMARY KEY és una columna o joc de columnes que identifica de manera única cada fila de taula.

Aquesta restricció **aplica** la **unicitat** de la columna o combinació de columnes i garanteix que **cap** columna que forme part de la clau continga un **valor nul**.



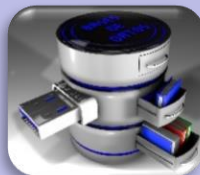
Restriccions

FOREIGN KEY

La restricció FOREIGN KEY (o integritat referencial) designa una columna o combinació de columnes com a clau aliena i estableix una relació entre una clau primària o única en la mateixa taula o en una altra diferent.

El valor de clau aliena ha de coincidir amb un valor existent de la taula principal o ser un valor NULL.

La clau aliena es defineix en la taula secundària i la taula que conté la columna a la qual es fa referència és la taula principal

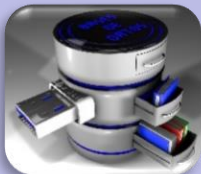


Restriccions

FOREIGN KEY

La clau aliena es defineix mitjançant una combinació de les següents paraules clau:

- FOREIGN KEY: s'utilitza per a definir la columna en la taula secundària a nivell de restricció de taula.
- REFERENCES identifica la taula i la columna en la taula principal.
- ON DELETE CASCADE indica que en suprimir una fila de la taula principal, també se suprimeixen les files dependents de la taula secundària.
- ON DELETE SET NULL indica que en suprimir una fila de la taula principal, els valors de clau aliena es defineixen com a nuls.

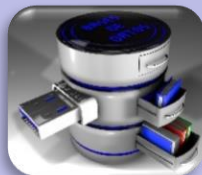


Restriccions

FOREIGN KEY

El comportament per defecte no permet l'actualització o supressió de les dades als quals es fa referència.

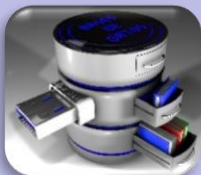
Sense les opcions ON DELETE CASCADE o ON DELETE SET NULL, la fila de la taula principal no es pot suprimir si fa referència a la mateixa en la taula secundària.



Restriccions

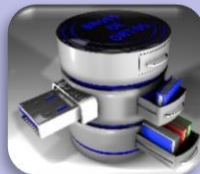
CHECK

La restricció CHECK defineix una condició que ha de complir cada fila.



Restriccions

```
CREATE TABLE employees
( employee_id      NUMBER(6)
  CONSTRAINT emp_employee_id PRIMARY KEY
, first_name      VARCHAR2(20)
, last_name       VARCHAR2(25)
  CONSTRAINT emp_last_name_nn NOT NULL
, email           VARCHAR2(25)
  CONSTRAINT emp_email_nn    NOT NULL
  CONSTRAINT emp_email_uk    UNIQUE
, phone_number    VARCHAR2(20)
, hire_date       DATE
  CONSTRAINT emp_hire_date_nn NOT NULL
, job_id          VARCHAR2(10)
  CONSTRAINT emp_job_nn      NOT NULL
, salary          NUMBER(8,2)
  CONSTRAINT emp_salary_ck   CHECK (salary>0)
, commission_pct  NUMBER(2,2)
, manager_id      NUMBER(6)
  CONSTRAINT emp_manager_fk REFERENCES
    employees (employee_id)
, department_id   NUMBER(4)
  CONSTRAINT emp_dept_fk     REFERENCES
    departments (department_id));
```



Activitat



Crea les següents taules

Curso (codigo, nom, hores, data_inici, comentaris)

CP{código}

VNN{nom}

Data_inici (valor por defecto la fecha del sistema)

PERSONA (dni, nom, cognom1, cognom2, email, cod_curso)

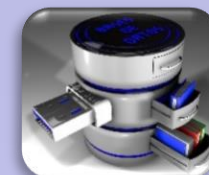
CP{dni}

VNN{nom}

VNN{cognom1}

UNI {email}

CAj{cod_curso} Referència a Curso (codigo)



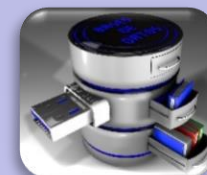
Activitat



Primer es creen les taules que no tenen claus alienes, ja que les claus alienes fan referència a una altra taula i si aquesta no existeix, no es pot indicar la clau aliena.

```
CREATE TABLE curso(  
codigo varchar2(4) CONSTRAINT cod_curso_pk PRIMARY KEY,  
nom varchar2(20) CONSTRAINT nom_curso_vnn NOT NULL,  
hores number(3),  
data_inici date DEFAULT sysdate,  
comentaris varchar2(150));
```

Per a comprovar que s'ha creat utilitzarem el comando DESCRIBE



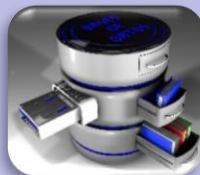
Activitat



El tipus de dades de la clau aliena ha de ser el mateix que la clau principal a la qual fa referència

```
CREATE TABLE persona(  
dni varchar2(9),  
nom varchar2(20) CONSTRAINT nom_persona_vnn NOT NULL,  
cognom1 varchar2(20) CONSTRAINT cognom_persona_vnn NOT NULL,  
email varchar2(40),  
cod_curso varchar2(4),  
CONSTRAINT dni_per_pk PRIMARY KEY (dni),  
CONSTRAINT email_per_uni UNIQUE (email),  
CONSTRAINT cod_curso_codigo_curso FOREIGN KEY (cod_curso) REFERENCES curso (codigo));
```

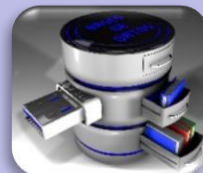
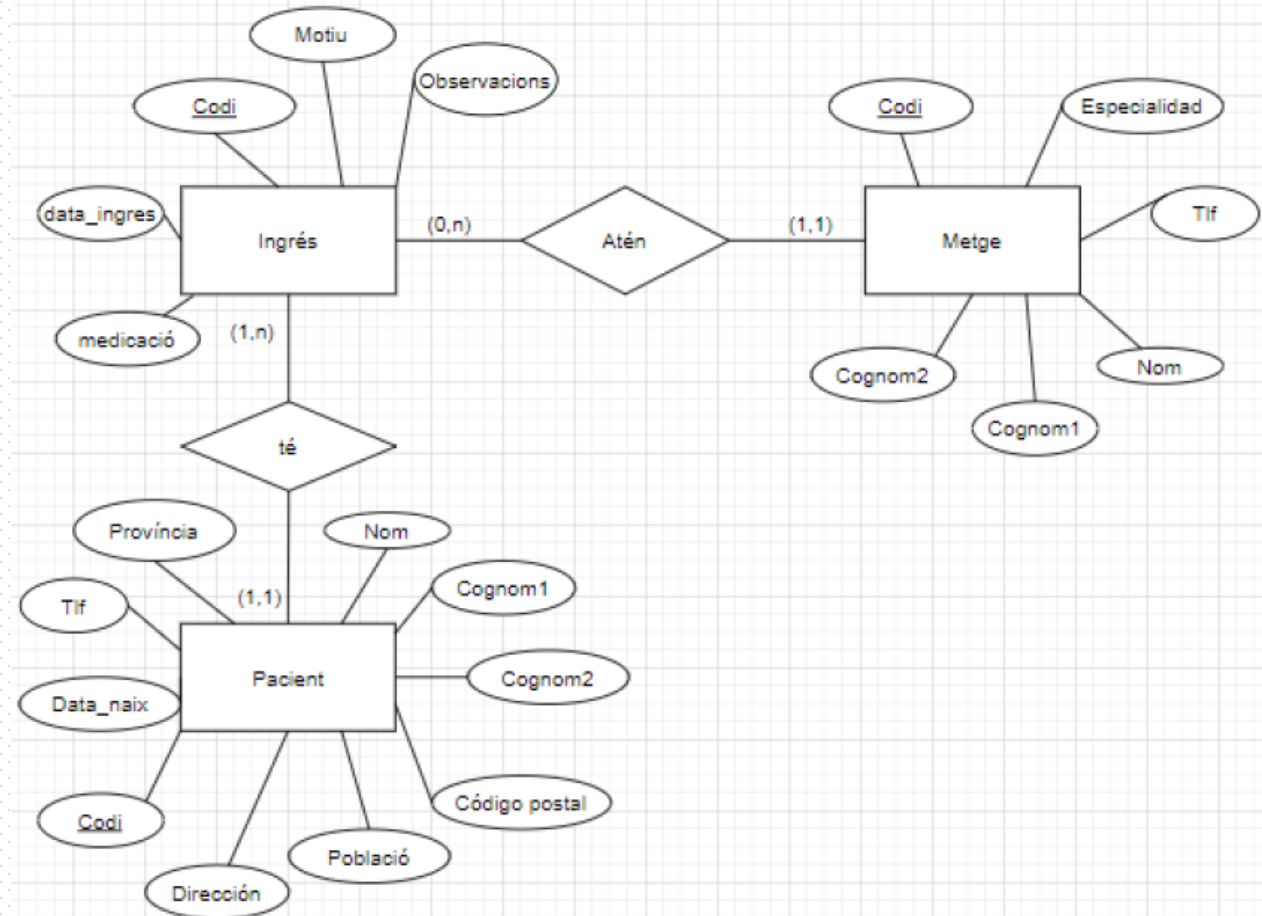
Per a comprovar que s'ha creat utilitzarem el comando DESCRIBE



Activitat



Donat este model E/R



Activitat



I la corresponent conversió

Ingrés (codi, data_ingrés, medicació, motiu, observacions, cod_m, cod_p)

CP{codi}

CAj{cod_m} Referència a Metge (codi)

CAJ{cod_p} Referència a Pacient (codi)

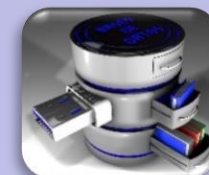
Metge (codi, nom, cognom1, cognom2, especialitat, tlf)

CP{codi}

Pacient (codi, nom, cognom1, cognom2, codi_postal, població, direcció, data_naix, tlf, provincia)

CP{codi}

Crea les taules corresponents



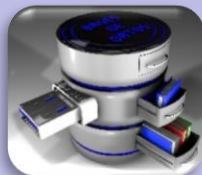
Creació de taules mitjançant subconsultas

Un segon mètode per a crear taules és aplicar la clàusula AS subquery, que crea la taula i inserida les files retornades de la subconsulta.

```
CREATE TABLE table  
            [(column, column...)]  
AS subquery;
```

Cal tindre en compte que:

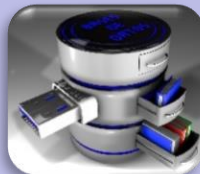
- La definició de columna només pot contindre el nom de la columna i el valor per defecte.
- Si es proporcionen especificacions de columna, el nombre de columnes ha de ser igual al nombre de columnes de la llista SELECT de la subconsulta.
- Si no es proporcionen especificacions de columna, els noms de columna de la taula són els mateixos que els de la subconsulta.



Creació de taules mitjançant subconsultas

```
CREATE TABLE table  
      [(column, column...)]  
AS subquery;
```

- Les definicions de tipus de dada de columna i de la restricció NOT NULL es transfereixen a la nova taula. Tinga en compte que només la restricció explícita NOT NULL s'heretarà. La columna PRIMARY KEY no transferirà la funció NOT NULL a la nova columna. Les altres regles de restricció no es transfereixen a la nova taula. No obstant això, pot agregar restriccions en la definició de columna



Creació de taules mitjançant subconsultas

```
CREATE TABLE dept80
AS
  SELECT last_name, first_name, salary, salary*12 ansal
  FROM employees
  WHERE department_id = 80;
```

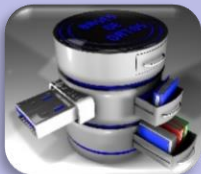
Table DEPT80 creado.

Comprovem la taula amb DESCRIBE

```
DESCRIBE dept80;
```

Nombre	Nulo	Tipo
LAST_NAME	NOT NULL	VARCHAR2(25)
FIRST_NAME		VARCHAR2(20)
SALARY		NUMBER(8,2)
ANSAL		NUMBER

Es molt important definir un alias

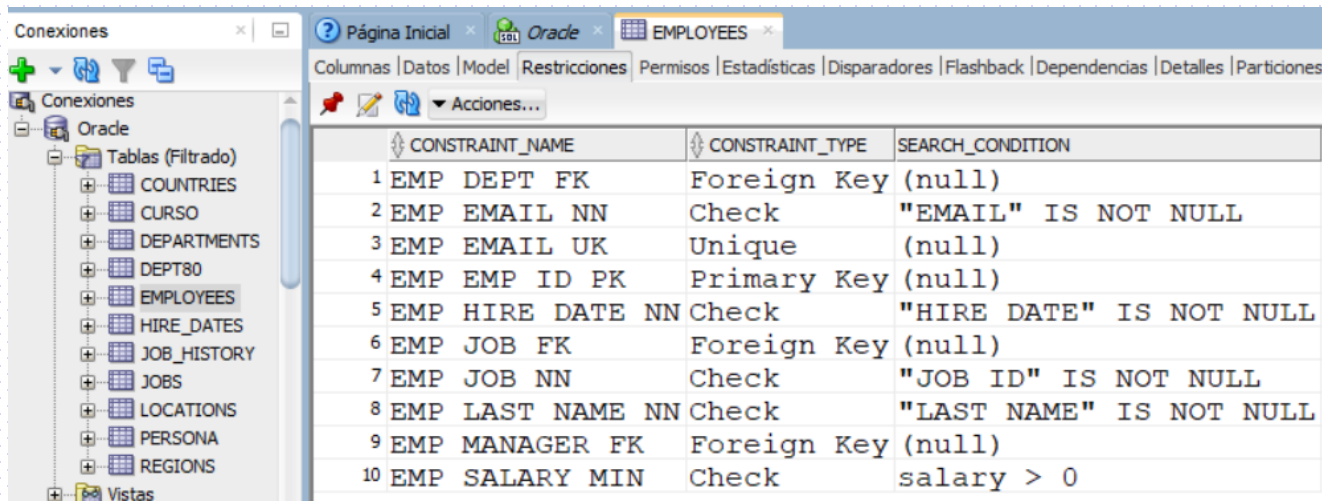


Creació de taules mitjançant subconsultas

```
CREATE TABLE emp_copy
AS

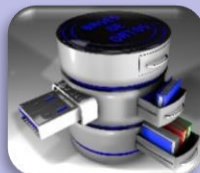
SELECT *
FROM employees;
```

Comprovem les restriccions que s'han heretat



	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
1	EMP DEPT FK	Foreign Key	(null)
2	EMP EMAIL NN	Check	"EMAIL" IS NOT NULL
3	EMP EMAIL UK	Unique	(null)
4	EMP EMP ID PK	Primary Key	(null)
5	EMP HIRE DATE NN	Check	"HIRE DATE" IS NOT NULL
6	EMP JOB FK	Foreign Key	(null)
7	EMP JOB NN	Check	"JOB ID" IS NOT NULL
8	EMP LAST NAME NN	Check	"LAST NAME" IS NOT NULL
9	EMP MANAGER FK	Foreign Key	(null)
10	EMP SALARY MIN	Check	salary > 0

Restriccions de
la taula
employees

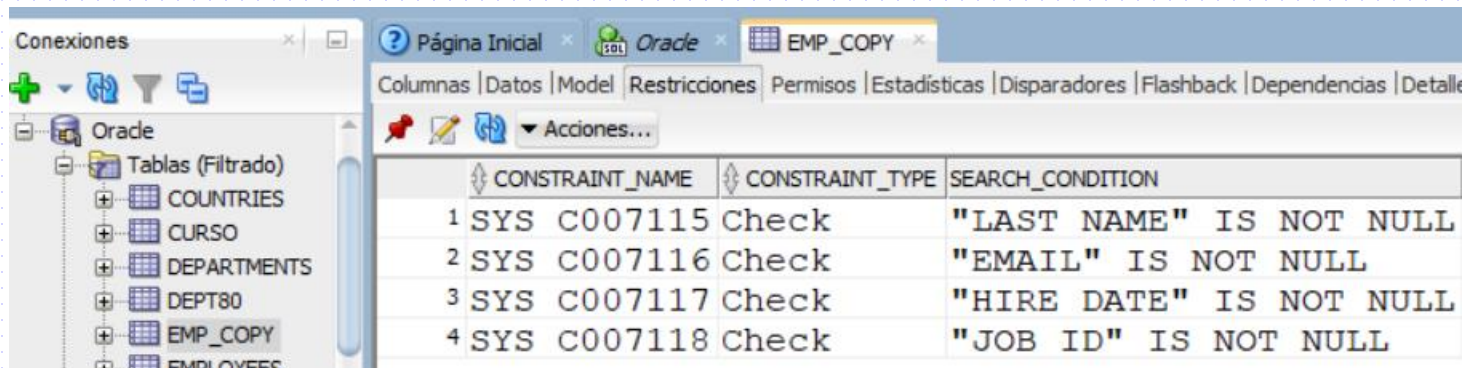


Creació de taules mitjançant subconsultas

```
CREATE TABLE emp_copy
AS

SELECT *
FROM employees;
```

Comprovem les restriccions que s'han heretat

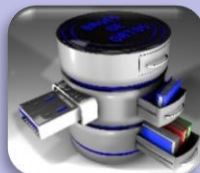


The screenshot shows the Oracle SQL Developer interface. On the left, the 'Conexiones' pane shows the 'Orade' database with a tree of tables including COUNTRIES, CURSO, DEPARTMENTS, DEPT80, EMP_COPY, and EMPLOYEES. The main window displays the 'EMP_COPY' table with the 'Restricciones' tab selected. The table has four constraints, all of type 'Check'.

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
1	SYS C007115	Check	"LAST NAME" IS NOT NULL
2	SYS C007116	Check	"EMAIL" IS NOT NULL
3	SYS C007117	Check	"HIRE DATE" IS NOT NULL
4	SYS C007118	Check	"JOB ID" IS NOT NULL

Restriccions de la taula emp_copy

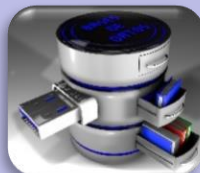
Recorda que només s'hereten les restriccions NOT NULL



Creació de taules mitjançant subconsultas

```
SELECT *  
FROM emp_copy;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SAL
100	Steven	King	SKING	515.123.4567	17/06/03	AD PRES	24000
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21/09/05	AD VP	17000
102	Lex	De Haan	LDEHAAN	515.123.4569	13/01/01	AD VP	17000
103	Alexander	Hunold	AHUNOLD	590.423.4567	03/01/06	IT PROG	9000
104	Bruce	Ernst	BERNST	590.423.4568	21/05/07	IT PROG	6000
105	David	Austin	DAUSTIN	590.423.4569	25/06/05	IT PROG	4800
106	Valli	Pataballa	VPATABAL	590.423.4560	05/02/06	IT PROG	4800
107	Diana	Lorentz	DLORENTZ	590.423.5567	07/02/07	IT PROG	4200
108	Nancy	Greenberg	NGREENBE	515.124.4569	17/08/02	FI MGR	12000
109	Daniel	Faviet	DFAVIET	515.124.4169	16/08/02	FI ACCOUNT	9000
110	John	Chen	JCHEN	515.124.4269	28/09/05	FI ACCOUNT	8200
111	Ismael	Sciarra	ISCIARRA	515.124.4369	30/09/05	FI ACCOUNT	7700
112	Jose Manuel	Urman	JMURMAN	515.124.4469	07/03/06	FI ACCOUNT	7800
113	Luis	Popp	LPOPP	515.124.4567	07/12/07	FI ACCOUNT	6900
114	Den	Raphaely	DRAPHEAT	515.127.4561	07/12/02	BIT MAN	11000

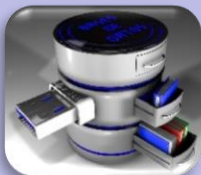


Sentència ALTER TABLE

Després de crear una taula, pot ser que necessite canviar l'estructura de la taula perquè ha omés una columna, s'ha de canviar la definició de la columna o necessita eliminar columnes.

Per a això, pot utilitzar la sentència ALTER TABLE.

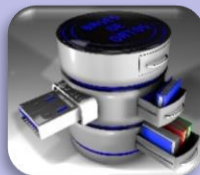
Per tant utilitzarem ALTER TABLE per a modificar l'estructura d'una taula



Sentència ALTER TABLE

Agregar, modificar i esborrar columnes

És molt probable que necessitem afegir/modificar/esborrar una columna en una taula ja existent. En totes les ocasions utilitzarem ALTER TABLE, encara que la sintaxi variarà en funció de l'operació que vulguem realitzar.



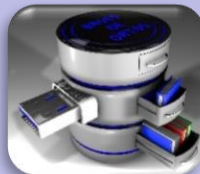
Sentència ALTER TABLE

Agregar, modificar i esborrar columnes

```
ALTER TABLE table
ADD          (column datatype [DEFAULT expr]
              [, column datatype]...);
```

```
ALTER TABLE table
MODIFY       (column datatype [DEFAULT expr]
              [, column datatype]...);
```

```
ALTER TABLE table
DROP (column [, column] ...);
```



Sentència ALTER TABLE

Agregar, modificar i esborrar columnes

Suposem la taula dept80 amb la següent estructura. `DESCRIBE dept80;`

Es pretén afegir una nova columna job_id

```
ALTER TABLE dept80  
ADD (job_id VARCHAR2(9));
```

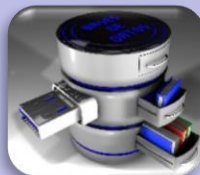
Table DEPT80 alterado.

```
DESCRIBE dept80;
```

Nombre	Nulo	Tipo
LAST_NAME	NOT NULL	VARCHAR2(25)
FIRST_NAME		VARCHAR2(20)
SALARY		NUMBER(8,2)
ANSAL		NUMBER
JOB_ID		VARCHAR2(9)

La nova columna està en última posició

Nombre	Nulo	Tipo
LAST_NAME	NOT NULL	VARCHAR2(25)
FIRST_NAME		VARCHAR2(20)
SALARY		NUMBER(8,2)
ANSAL		NUMBER

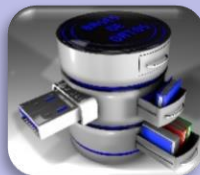


Sentència ALTER TABLE

Agregar, modificar i esborrar columnes

```
SELECT *  
FROM dept80;
```

LAST_NAME	FIRST_NAME	SALARY	ANSAL	JOB_ID
Russell	John	14000	168000	(null)
Partners	Karen	13500	162000	(null)
Errazuriz	Alberto	12000	144000	(null)
Cambrault	Gerald	11000	132000	(null)
Zlotkey	Eleni	10500	126000	(null)
Tucker	Peter	10000	120000	(null)
Bernstein	David	9500	114000	(null)
Hall	Peter	9000	108000	(null)
Olsen	Christopher	8000	96000	(null)
Cambrault	Marlette	7500	90000	(null)



Sentència ALTER TABLE

Agregar, modificar i esborrar columnes

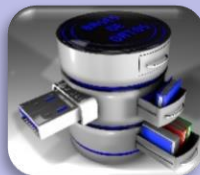
La modificació de la columna pot incloure canvis en la mena de dada, grandària i valor per defecte d'una columna.

```
ALTER TABLE dept80  
MODIFY (last_name VARCHAR2(30));
```

Table DEPT80 alterado.

```
DESCRIBE dept80;
```

Nombre	Nulo	Tipo
-----	-----	-----
LAST_NAME	NOT NULL	VARCHAR2 (30)
FIRST_NAME		VARCHAR2 (20)
SALARY		NUMBER (8,2)
ANSAL		NUMBER
JOB_ID		VARCHAR2 (9)

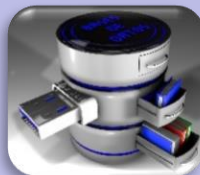


Sentència ALTER TABLE

Agregar, modificar i esborrar columnes

Instruccions

- ✓ Pot augmentar l'ample o la precisió d'una columna numèrica.
- ✓ Pot augmentar l'ample de les columnes de caràcters.
- ✓ Pot reduir l'ample d'una columna si:
 - La columna només conté valors nuls
 - La taula no té files
 - La disminució de l'ample de columna no és inferior als valors existents d'aquesta columna



Sentència ALTER TABLE

Agregar, modificar i esborrar columnes

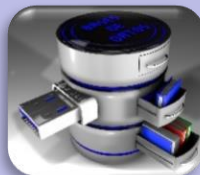
Per a esborrar una columna de la taula, utilitzarem la sentència ALTER TABLE amb la clàusula DROP COLUMN.

```
ALTER TABLE dept80  
DROP COLUMN job_id;
```

Table DEPT80 alterado.

```
DESCRIBE dept80;
```

Nombre	Nulo	Tipo
LAST_NAME	NOT NULL	VARCHAR2 (30)
FIRST_NAME		VARCHAR2 (20)
SALARY		NUMBER (8,2)
ANSAL		NUMBER



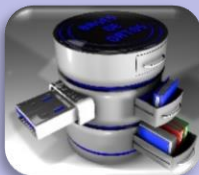
Sentència ALTER TABLE

Gestió de restriccions

Amb la instrucció ALTER TABLE es poden gestionar les restriccions de la següent manera:

- Agregar o esborrar una restricció, però sense modificar la seua estructura
- Activar o desactivar restriccions
- Agregar una restricció NOT NULL mitjançant la clàusula MODIFY

```
ALTER TABLE <table_name>  
ADD [CONSTRAINT <constraint_name>]  
type (<column_name>);
```

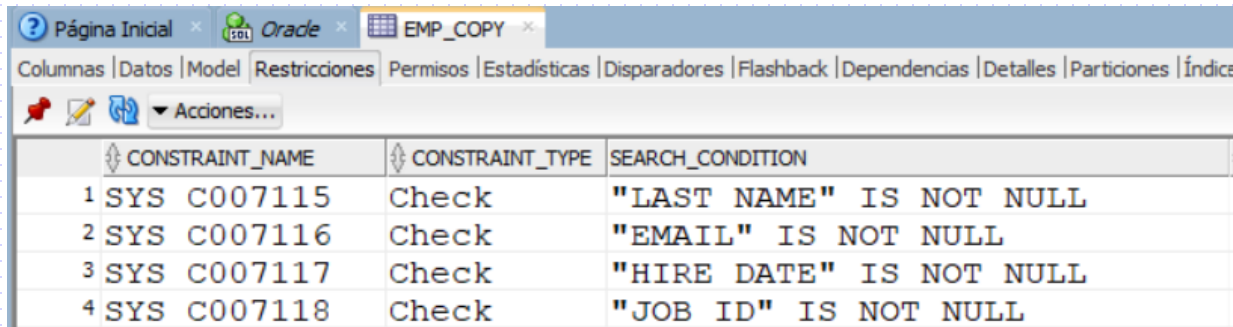


Sentència ALTER TABLE

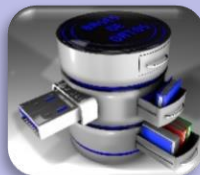
Gestió de restriccions - ADD

Tenim una copia de la taula employees, anomenada emp_copy, amb les restriccions son les següents

Nombre	Nulo	Tipo
EMPLOYEE_ID		NUMBER (6)
FIRST_NAME		VARCHAR2 (20)



	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
1	SYS_C007115	Check	"LAST NAME" IS NOT NULL
2	SYS_C007116	Check	"EMAIL" IS NOT NULL
3	SYS_C007117	Check	"HIRE DATE" IS NOT NULL
4	SYS_C007118	Check	"JOB ID" IS NOT NULL



Sentència ALTER TABLE

Gestió de restriccions - ADD

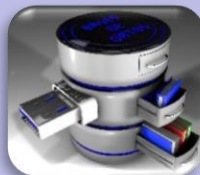
Per a afegir restriccions. En este cas una restricció de clau primària.

```
ALTER TABLE emp_copy  
ADD CONSTRAINT empc_empid_pk PRIMARY KEY (employee_id);
```

```
DESC emp_copy;
```

Nombre	Nulo	Tipo
-----	-----	-----
EMPLOYEE_ID	NOT NULL	NUMBER (6)
FIRST_NAME		VARCHAR2 (20)
LAST NAME	NOT NULL	VARCHAR2 (25)

Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Pa			
Acciones...			
CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	
1 EMPC EMPID PK	Primary Key	(null)	
2 SYS C007115	Check	"LAST NAME" IS NOT	



Sentència ALTER TABLE

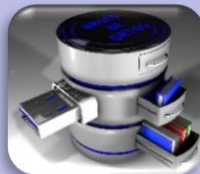
Gestió de restriccions - MODIFY

Només per a afegir una restricció de NOT NULL.

```
ALTER TABLE emp_copy  
MODIFY first name NOT NULL ;
```

```
DESC emp_copy;
```

Nombre	Nulo	Tipo
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME	NOT NULL	VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)



Sentència ALTER TABLE

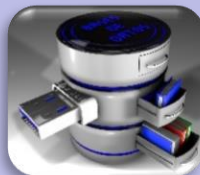
Gestió de restriccions - DROP

Per a eliminar restriccions, no la columna.

```
ALTER TABLE emp_copy  
DROP CONSTRAINT empc_empid_pk;
```

```
DESC emp_copy;
```

Nombre	Nulo	Tipo
EMPLOYEE_ID		NUMBER (6)
FIRST_NAME	NOT NULL	VARCHAR2 (20)
LAST_NAME	NOT NULL	VARCHAR2 (25)

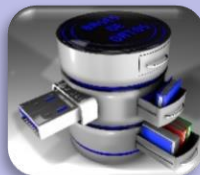


Sentència ALTER TABLE

Gestió de restriccions - DROP

També podria utilitzar-se la següent instrucció

```
ALTER TABLE emp_copy  
DROP PRIMARY KEY;
```



Sentència ALTER TABLE

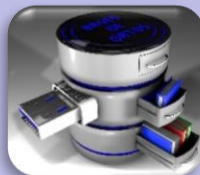
Canviar noms de restriccions o columnes

En canviar el nom d'una columna de la taula, el nou nom no ha d'entrar en conflicte amb el nom de qualsevol columna existent en la taula. Pot utilitzar-se ALTER TABLE juntament amb la clàusula RENAME COLUMN.

```
ALTER TABLE emp_copy  
RENAME COLUMN employee_id TO cod_employee;
```

```
DESC emp_copy;
```

Nombre	Nulo	Tipo
-----	-----	-----
COD_EMPLOYEE		NUMBER(6)
FIRST_NAME	NOT NULL	VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)

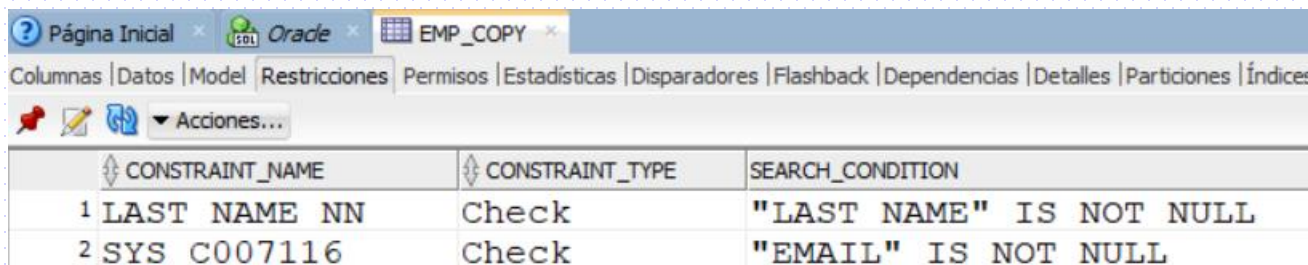


Sentència ALTER TABLE

Canviar noms de restriccions o columnes

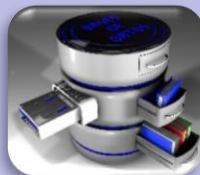
En canviar el nom d'una restricció existent per a una taula, el nou nom no ha d'entrar en conflicte amb cap dels noms de les restriccions existents (en qualsevol taula). Pot utilitzar la clàusula RENAME CONSTRAINT per a canviar el nom de les restriccions generades pel sistema.

```
ALTER TABLE emp_copy  
RENAME CONSTRAINT SYS_C007115 TO last_name_nn;
```



The screenshot shows the Oracle SQL Developer interface with the 'EMP_COPY' table selected. The 'Restricciones' (Constraints) tab is active, displaying a table of constraints for the selected table.

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
1	LAST NAME NN	Check	"LAST NAME" IS NOT NULL
2	SYS C007116	Check	"EMAIL" IS NOT NULL



Sentència DROP TABLE

La sentència DROP TABLE mou una taula a la paperera de reciclatge o elimina la taula i totes les seues dades de la base de dades completament. Llevat que especifique la clàusula PURGE, la sentència DROP TABLE no torna a alliberar espai en els tablespaces perquè ho utilitzen altres objectes i l'espai continua comptant en la quota d'espai de l'usuari. L'esborrat d'una taula invalida objectes dependents i elimina privilegis d'objecte en la taula.

En esborrar una taula, la base de dades perd totes les dades de la taula i els índexs associats a aquests.

```
DROP TABLE emp_copy;
```

```
DROP TABLE emp_copy PURGE;
```

