

BASES DE DADES

DSIC. Curs 2000-2001

Pràctica Definició de Dades (SQL)

Objectius

- Presentar el llenguatge de definició de dades del SQL/92
 - Presentar la sintaxi de creació de taules
 - Presentar la sintaxi de la modificació de definició d'una taula
 - Presentar la sintaxi de la creació d'una vista
 - Presentar la sintaxi per a atorgar autoritzacions
 - Presentar la sintaxi per a la creació de regles d'activitat (disparadors)
- Tot açò per al sistema Oracle 8.x

Definició de Dades (SQL)

Instruccions del SQL per a poder definir esquemes relacionals:

- **create schema:** permet donar nom a un esquema relacional i declarar l'usuari que és el creador i propietari del dit esquema.
- **create domain:** permet definir un nou domini de dades.
- ORACLE • **create table:** defineix una taula, el seu esquema i les restriccions associades.
- ORACLE • **create view:** defineix una vista o relació derivada en l'esquema relacional.
- **create assertion:** permet definir restriccions d'integritat generals.
- ORACLE • **grant:** permet definir autorització d'operacions sobre objectes de la BD.

Totes aquestes instruccions tenen associada l'operació inversa (DROP / REVOKE) i modificació (ALTER).

Definició de l'Esquema (SQL)

```
create schema [esquema] [authorization usuari]  
[llista_element_esquema];
```

Un element d'esquema pot ser un dels següents:

- Definició de domini.
- Definició de taula.
- Definició de vista.
- Definició de restricció.
- Definició de privilegi.

Eliminació de la definició d'un esquema relacional:

```
drop schema esquema {restrict | cascade};
```

Definició de Dominis (SQL)

```
create domain domini [as] tipus_dada  
    [default {literal | funció_sistema | null }]  
    [definició_restricció_domini];
```

Funcions del sistema:

- user
- current_user
- session_user
- current_date
- current_time
- current_timestamp.

Definició de Dominis (SQL)

A un domini se li pot associar un conjunt de restriccions:

[constraint *restricció*]

check (expressió_condicional)

[not] deferrable

- *expressió_condicional* permet expressar qualsevol condició que ha de complir sempre el domini (ha de ser CERTA o INDEFINIDA)
- **deferrable** indica que el sistema ha de comprovar la restricció al finalitzar la transacció activa.
- **Not deferrable** indica que el sistema ha de comprovar la restricció després de cada operació d'actualització a la base de dades.

Definició de Dominis (SQL). Exemple

```
CREATE DOMAIN angle AS FLOAT  
    DEFAULT 0  
    CHECK (VALUE >= 0 AND VALUE < 360)  
    NOT DEFERRABLE;
```

Eliminació d'un Domini:

```
drop domain domini [restrict | cascade]
```

Definició de Taules (SQL).

```
CREATE TABLE taula  
    comallista_definició_columna  
    [comallista_definició_restricció_taula];
```

La definició d'una columna d'una taula es realitza com segueix:

```
columna { tipus_dada | domini }  
    [default { literal | funció_sistema | null }]  
    [llista_definició_restricció_columna]
```

Les restriccions que es poden definir sobre les columnes són les següents:

- not null: restricció de valor no nul.
- Definicions de restriccions de CP, UNI, CAI d'una sola columna.
- Definició de restriccions generals amb la clàusula check.

Definició de Taules (SQL).

La clàusula per a definir restriccions de taula és la següent:

[constraint *restricció*]

{ primary key (*comallista_columnna*)

| unique (*comallista_columnna*)

| foreign key (*comallista_columnna*)

references *taula*[(*comallista_columnna*)]

[match { full | partial }]

* NO ORACLE

[on update [cascade |

* NO ORACLE

set null | set default | no action]]

* NO ORACLE

[on delete [cascade |

set null | set default | no action]]

* NO ORACLE

| check *expressió_condicional* }

[comprovació_restricció]

- ha de ser CERTA o INDEFINIDA.

- no pot incloure subconsultes ni referències a altres taules.

Exemple: Proveedor-Piezas-Suministro

d_cod_pieza: tira(4)

d_cod_proy: tira(4)

d_dni: entero (positivo)

Proveedor(dni: d_dni, nombre: tira(40), dirección: tira(25), ciudad: tira(30))

CP: {dni}

VNN: {nombre}

Pieza(código: d_cod_pieza, desc: tira(40), color: tira(20), peso: real)

CP: {código}

Suministro(dni: d_dni, código: tira(4), precio: real)

CP: {dni, código}

CAj: {dni} \rightarrow Proveedor

CAj: {código} \rightarrow Pieza

Restriccions d'integritat:

R1) Px: Pieza $\forall Px (\text{Pieza}(Px) \wedge Px.\text{color} = \text{'rojo'} \rightarrow Px.\text{peso} > 100)$

R2) Px: Pieza, Sx: Suministro $\forall Px (\text{Pieza}(Px) \rightarrow \exists Sx (\text{Suministro}(Sx) \wedge Sx.\text{código} = Px.\text{código}))$

Example: Proveedor-Piezas-Suministro (SQL)

```
create schema Almacén
authorization pepe
create domain d_cod_pieza as char(4)
create domain d_cod_proy as char(4)
create domain d_dni as integer check value>0
create table Proveedor ( dni          d_dni    primary key,
                        nombre        varchar(40) not null,
                        dirección      char(25),
                        ciudad         char(30) )
create table Pieza ( código          d_cod_pieza primary key,
                   desc              varchar(40) not null,
                   color             char(20),
                   peso              float,
                   constraint r1 check (color<>'rojo' or peso>100))
create table Suministro ( dni          d_dni,
                        código         d_cod_pieza references Pieza,
                        precio         float,
                        primary key (dni, código),
                        foreign key (dni) references Proveedor(dni) );
```

← R1

¿i R2?

Definició de Taules (SQL). Clàusula MATCH

- completa (**match full**): en cada tupla de R la clau aliena CA té el valor nul o no el té, en cada una de les seues columnes. En el segon cas, ha d'existir una fila en la taula S tal que el seu valor en les columnes de CU siga idèntic.
- parcial (**match partial**): en cada tupla de R la clau aliena CA té el valor nul en cada una de les seues columnes, o ha d'existir una fila en la taula S , de forma que per a les columnes de la clau aliena CA que no tenen valor nul, el valor en les columnes corresponents de CU es idèntic.
- dèbil (**no s'inclou clàusula match**): en cada tupla de R si la clau aliena CA no té el valor nul, en cada una de les seues columnes, ha d'existir una fila en la taula S tal que el valor en les columnes corresponents de CU coincidisca.

ORACLE

Modificació de Definició de Taules (SQL).

Per a modificar la definició d'una taula:

`alter table taula_base`

`{ add [column] definició_columna`

`| alter [column] columna`

`{ set default { literal | funció_sistema | null }`

`| drop default }`

`| drop [column] columna { restrict | cascade } };`

En ORACLE canvien
algunes coses

Per a eliminar una taula de l'esquema relacional:

`drop table taula_base { restrict | cascade };`

En ORACLE és CASCADE
CONSTRAINTS

Definició de Vistes (SQL).

Les vistes són els objectes que proporciona el llenguatge SQL per a definir esquemes externs.

- Una vista és una taula virtual (no té una correspondència a nivell físic)
- Es pot manejar com una taula bàsica.
- Una vista es defineix en base a altres taules (bàsiques o virtuals).
- Les actualitzacions es transfereixen a la/es taula/es original/s (amb certes limitacions).

```
CREATE | REPLACE VIEW vista [(comallista_columna)]  
AS expressió_taula [with check option]
```

Definició de Vistes (SQL). Exemple.

```
CREATE VIEW PROVEEDORES_VAL  
AS    SELECT * FROM PROVEEDOR  
      WHERE ciudad = 'Valencia'  
  
WITH CHECK OPTION;
```

El Check Option impideix que jo puga afegir proveïdors que no siguen de València

Eliminació d'una Vista

```
DROP VIEW vista [restrict | cascade];
```

ULL EN EL SENTIT
DE CASCADE!

Vistes Actualitzables.

Motius pels quals una vista no és actualitzable:

- conté operadors conjuntistes (UNION, INTERSECT,...)
- conté l'operador DISTINCT
- conté funcions agregades (SUM, AVG, ..)
- conté la clàusula GROUP BY

Vistes Actualitzables.

Vista sobre una taula bàsica:

- el sistema traduirà l'actualització sobre la vista en una operació de actualització sobre la relació bàsica
- sempre que no es viole cap restricció d'integritat definida sobre la dita relació

Vistes Actualitzables.

Vista sobre una concatenació de relacions:

- l'actualització només pot modificar una de les taules bàsiques
- actualització modificarà la relació bàsica que complisca la propietat de *conservació de la clau* (aquella relació tal que la seua clau primària podria ser també clau de la vista)
- l'actualització no es realitzarà si viola alguna de les restriccions definides sobre la relació bàsica que es vol actualitzar

Definició de Restriccions (SQL)

```
create assertion restricció  
    check (expressió_condicional)  
    [comprovació_restricció];
```

La condició ha de ser CERTA.

Exemple: Proveedor-Piezas-Suministro (SQL)

La restricció R2 :

R2) Px: Pieza, Sx: Suministro $\forall Px (\text{Pieza}(Px) \rightarrow \exists Sx (\text{Suministro}(Sx) \wedge Sx.\text{código}=Px.\text{código}))$

es defineix mitjançant una restricció general:

create assertion R2 check

not exists(select * from Pieza P

where not exists(select *

from Suministro S

where P.código=S.código));

Eliminació d'una Restricció

DROP ASSERTION *restricció*

Definició de Privilegis (SQL).

Per a realitzar qualsevol operació sobre un objecte (taula o vista) s'ha de tindre el privilegi necessari. Les operacions amb privilegis són:

- update (especificant quines columnes)
- insert (especificant quines columnes)
- delete
- select
- create view: és necessari tindre el privilegi sobre l'expressió de taula de la definició de la vista (el SELECT de la vista).

Definició de Privilegis (SQL).

```
grant {all | select | insert[(comallista_columna)] |  
      delete | update[(comallista_columna)]}  
on objecte to {comallista_usuaris | public }  
[with grant option]
```

En ORACLE canvien
algunes coses

Eliminació d'un Privilegi

```
revoke [grant option for]  
      {all | select | insert[(comallista_columna)] |  
      delete | update[(comallista_columna)]}  
      on objecte to {comallista_usuaris | public }  
      {restrict | cascade}
```