

Resultats d'aprenentatge

RA2. Crea bases de dades definint-ne l'estructura i les característiques dels elements segons el model relacional.

Objectiu

Practicar els conceptes estudiats sobre el llenguatge SQL-DDL

Exercicis

Has de crear un script SQL amb les sentències que utilitzis.

Exercici 1

a) Crear la següent taula **FITXA** d'una base de dades d'una **AGENDA**. L'estructura és la següent:

Nom Camp	Tipus de dades	Comentari	Restricció
DNI	NUMERIC(10)	DNI de la persona	Camp clau
NOM	VARCHAR(30)	Nom de la persona	Ha de ser obligatori
COGNOMS	VARCHAR(70)	Cognoms de la persona	Ha de ser obligatori
ADREÇA	VARCHAR(60)	Adreça de la persona	
TELEFON	VARCHAR(11)	Telèfon de la persona	Ha de ser obligatori
PROVINCIA	VARCHAR(30)	Província on resideix la persona	
DATA_NAIX	DATE	Data de naixement de la persona	Per defecte ha de ser la d'avui

Important: Per establir la data actual utilitza la funció **DATE DEFAULT CURRENT_DATE**.

--Creem la base de dades, usuari i schema **agenda**

```
CREATE ROLE agenda
```

```
superuser
```

```
createdb
```

```
creatorole
```

```
inherit
```

```
login
```

```
replication;
```

```
ALTER USER agenda WITH PASSWORD 'agenda';
```

```
CREATE DATABASE agenda WITH OWNER agenda;
```

```
CREATE SCHEMA agenda;
```

```
GRANT ALL PRIVILEGES ON DATABASE agenda TO agenda;
```

```
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA agenda TO agenda;
```

--Ens connectem a la base de dades **agenda** des de l'usuari **postgres**

\c agenda

--O sortim amb \q i ens connectem a la base de dades **agenda** amb l'usuari **agenda**.

psql -U agenda -W -d agenda

--Creem la taula **Fitxa** a la base de dades **agenda**

```
CREATE TABLE IF NOT EXISTS Fitxa (  
  DNI NUMERIC(10) NOT NULL,  
  Nom VARCHAR (30) NOT NULL,  
  Cognoms VARCHAR (70) NOT NULL,  
  Adreça VARCHAR (60),  
  Telefon VARCHAR (11) NOT NULL,  
  Provincia VARCHAR (30),  
  Data_Naix DATE DEFAULT CURRENT_DATE,  
  CONSTRAINT PK_Fitxa PRIMARY KEY (DNI));
```

--Comprovem les taules creades a la base de dades agenda

\d

--Revisem al informació dels camps de la taula Fitxa

\d fitxa

b) Afegir un nou camp a la taula **Fitxa**, anomenat **cp** que serà el codi postal. Serà de tipus varchar(5).

```
ALTER TABLE Fitxa  
ADD cp VARCHAR(5);
```

c) Comprovar que s'ha creat correctament el camp a la taula.

\d fitxa

d) Canviar el nom del nou camp **cp**. S'ha de dir **Codi_Postal**.

--Si volem canviar el nom del camp a codiPostal fem:

```
ALTER TABLE fitxa RENAME COLUMN cp TO Codi_Postal;
```

--Comprovem el canvi

\d fitxa

e) Canviar el nom de la restricció anomenada **PK_Fitxa**. S'ha de dir **PrimKey_Fitxa**

```
ALTER TABLE fitxa RENAME CONSTRAINT PK_Fitxa TO PrimKey_Fitxa;
```

--Comprovem el canvi

\d fitxa

f) Modificar la longitud del tipus de dada del camp **Codi_Postal**. La nova longitud és VARCHAR(10).

```
ALTER TABLE fitxa ALTER COLUMN codiPostal SET DATA TYPE VARCHAR(10);
```

```
-- I comprovem que el canvi
```

```
\d fitxa
```

g) Modificar el tipus de dades del camp **Codi_Postal**. a El nou tipus de dades per aquest camp és NUMERIC(5). Si et salta l'error busca per internet com es podria fer.

```
ALTER TABLE fitxa ALTER COLUMN Codi_Postal SET DATA TYPE NUMERIC(5);
```

```
--Ens salta un error i no ens deixa canvir de tipus caràcter a numéric
```

```
--Per fer-ho fem:
```

```
ALTER TABLE fitxa ALTER COLUMN Codi_Postal TYPE NUMERIC USING Codi_Postal::NUMERIC(5);
```

Referència:

<https://www.todopostgresql.com/comandos-postgresql-alter/>

h) Afegeix una restricció amb el nom **CK_Upper_Prov** a la taula **FITXA** per verificar que els valors de camp **Provincia** estiguin en majúscules.

```
ALTER TABLE fitxa
```

```
ADD CONSTRAINT CK_Upper_Prov CHECK (Provincia = UPPER (Provincia));
```

```
-- I comprovem que el canvi
```

```
\d fitxa
```

i) Elimina la restricció **CK_Upper_Prov** de la taula **FITXA** i comprova el canvi.

```
ALTER TABLE fitxa
```

```
DROP CONSTRAINT CK_Upper_Prov;
```

```
-- I comprovem que el canvi
```

```
\d fitxa
```

j) Canvia el nom de la taula **FITXA**. S'ha de dir **ENTRADA** i comprova el canvi.

```
ALTER TABLE FITXA RENAME TO ENTRADA;
```

k) Elimina la taula **ENTRADA** i Comprovar que la taula està eliminada.

```
DROP TABLE ENTRADA;
```

```
\d
```

CFGS DAMv1

Mòdul 2 NF2

16/10/2024

Primera Convocatòria

Solució EA4. Modificació taules

Aquest EA correspon al RA2

S'avalua com apte/no apte i contribueix junt amb els altres EAs al 20% de la nota final del RA2

Formador/a: joan.colomer@itb.cat

Lliurament de l'activitat

Envia la solució de l'exercici en un fitxer **.sql** anomenat **M2NF2EA4_CognomNom.sql** que contindrà el script sql creat.