Dirección Xeral de Educación, Formación Profesional e Innovación Educativa

Material para

a formación profesional inicial

A02. Modificación da estrutura de bases de datos relacionais

|  |  |  |
| --- | --- | --- |
| Familia profesional | IFC | Informática e comunicacións |
| Ciclo formativo | CSIFC03  CSIFC02 | Desenvolvemento de aplicacións web  Desenvolvemento de aplicacións multiplataforma |
| Grao |  | Superior |
| Módulo profesional | MP0484 | Bases de datos |
| Unidade didáctica | UD04 | Deseño físico de bases de datos |
| **Actividade** | **A02** | **Modificación da estrutura de bases de datos relacionais** |
| Autores |  | Marta Fernández García  María del Carmen Fernández Lameiro  Miguel Fraga Vila  María Carmen Pato González  Andrés del Río Rodríguez |
| Nome do arquivo |  | CSIFC02\_ MP0484\_V000402\_UD04\_A02\_ModBD.docx |
| © 2015 Xunta de Galicia.  Consellería de Cultura, Educación e Ordenación Universitaria.  Este traballo foi realizado durante unha licenza de formación retribuída pola Consellería de Cultura, Educación e Ordenación Universitaria e ten licenza CreativeCommons BY-NC-SA (recoñecemento - non comercial - compartir igual). Para ver unha copia desta licenza, visitar a ligazón http://creativecommons.org/licenses/by-nc-sa/3.0/es/. | | |

Índice

[1. Ficha técnica 4](#_Toc443054175)

[Contexto da actividade 4](#_Toc443054176)

[Título da actividade 5](#_Toc443054177)

[Resultados de aprendizaxe do currículo 5](#_Toc443054178)

[Obxectivos didácticos e título e descrición da actividade 5](#_Toc443054179)

[Criterios de avaliación 5](#_Toc443054180)

[Contidos 5](#_Toc443054181)

[Actividades de ensino e aprendizaxe e de avaliación, métodos, recursos e instrumentos de avaliación 6](#_Toc443054182)

[2. A02. Modificación da estrutura de bases de datos relacionais 7](#_Toc443054183)

[2.1 Introdución 7](#_Toc443054184)

[2.1.1 Obxectivos 7](#_Toc443054185)

[2.1.2 Software e bases de datos 7](#_Toc443054186)

[2.2 Actividade 8](#_Toc443054187)

[2.2.1 Modificación dunha base de datos 8](#_Toc443054188)

[2.2.2 Modificación do esquema dunha táboa 8](#_Toc443054189)

[2.2.3 Engadir relacións e restricións de clave foránea 10](#_Toc443054190)

[2.2.4 Tarefa integradora sobre a LDD 10](#_Toc443054191)

[2.3 Tarefas 11](#_Toc443054192)

[2.3.1 Tarefa 1. Modificar bases de datos empregando a sentenza ALTER 11](#_Toc443054193)

[Solución 11](#_Toc443054194)

[2.3.2 Tarefa 2. Modificar táboas empregando a sentenza ALTER 11](#_Toc443054195)

[Solución 11](#_Toc443054196)

[2.3.3 Tarefa 3: Probar as posibilidades de creación de dúas táboas con interrelacións 1:N bidireccionais 12](#_Toc443054197)

[Solución 12](#_Toc443054198)

[2.3.4 Tarefa 4. Utilizar a sentenza ALTER TABLE para a creación das relacións entre táboas 14](#_Toc443054199)

[Solución 15](#_Toc443054200)

[2.3.5 Tarefa 5: Tarefa integradora sobre LDD 17](#_Toc443054201)

[Solución 18](#_Toc443054202)

[3. Materiais 22](#_Toc443054203)

[3.1 Documentos de apoio ou referencia 22](#_Toc443054204)

[3.2 Recursos didácticos 22](#_Toc443054205)

[3.3 Material auxiliar 22](#_Toc443054206)

[4. Avaliación 23](#_Toc443054207)

[Criterios de avaliación 23](#_Toc443054208)

[Modelo de proba combinada para TO.2 e TO.4 23](#_Toc443054209)

[Proba para CA4.2.2 sobre modificación de bases de datos 24](#_Toc443054210)

[Proba para CA4.3.2 sobre modificación de táboas 24](#_Toc443054211)

[Exemplo de solución para entregar en papel 24](#_Toc443054212)

[Exemplo de lista de valoración para TO.2 26](#_Toc443054213)

[Exemplo de lista de valoración para TO.4 26](#_Toc443054214)

1. Ficha técnica

Contexto da actividade

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Módulo | Duración  horas | Unidade didáctica. | Sesións 50´ | Actividades | Sesións 50´ |
| MP0484. Bases de datos | 187 | UD01. Bases de datos e sistemas de almacenamento da información | 12 | A01. Bases de datos e sistemas de almacenamento da información | 7 |
| A02. Introdución aos sistemas xestores de bases de datos | 5 |
| UD02. Deseño conceptual de bases de datos | 38 | A01. Fases do deseño de bases de datos | 3 |
| A02.Descrición e representación gráfica dos elementos do Modelo Entidade-Interrelación (MER) | 10 |
| A03. Descrición e representación gráfica dos elementos do Modelo Entidade-Interrelación Estendido(MERE) | 9 |
| A04. Construción e validación do modelo de datos | 11 |
| A05. Notación Martin e ferramentas CASE | 5 |
| UD03. Deseño lóxico de base de datos | 32 | A01. O deseño lóxico empregando o modelo relacional | 15 |
| A02. Normalización de relacións | 10 |
| A03. Operacións do MR: álxebra e cálculo | 7 |
| **UD04. Deseño físico de bases de datos** | **28** | A01. Creación da estrutura de bases de datos relacionais | 13 |
| **A02. Modificación da estrutura de bases de datos relacionais** | **9** |
| A03. Verificación e probas | 3 |
| A04. Utilización de ferramentas gráficas para a creación da estrutura de bases de datos relacionais | 3 |
| UD05. Consultas de datos | 39 | A01. Consultas simples cunha táboa. | 12 |
| A02. Consultas con datos de máis dunha táboa | 10 |
| A03. Consultas resumo con agrupamentos | 7 |
| A04. Consultas con subconsultas | 7 |
| A05. Optimización de consultas | 3 |
| UD06. Tratamento de datos | 24 | A01. Manipulación de datos | 12 |
| A02. Vistas | 5 |
| A03. Transaccións e bloqueos | 7 |
| UD07. Programación de bases de datos | 22 | A01. Introdución á programación con SQL | 4 |
| A02. Rutinas almacenadas | 9 |
| A03. Disparadores e eventos | 5 |
| A04. Cursores | 4 |
| UD08. Administración de bases de datos | 19 | A01. Copias de seguridade e intercambio de datos entre SXBDs | 10 |
| A02. Índices e xestión de usuarios | 9 |
| UD09. Bases de datos obxecto-relacionais | 10 | A01. Uso de bases de datos obxecto-relacionais | 10 |

NOTA: Esta actividade está vinculada á programación recollida no arquivo CSIFC02\_MP0487\_V000400\_UD04\_DesenoFisicoBD.pdf

Título da actividade

|  |  |  |  |
| --- | --- | --- | --- |
| Nº | Título | Descrición | Duración |
| A02 | Modificación da estrutura de bases de datos relacionais. | Modificarase a estrutura das bases de datos relacionais, táboas e relacións entre elas, utilizando a linguaxe de definición de datos. | 9 |

Resultados de aprendizaxe do currículo

|  |  |
| --- | --- |
| Resultados de aprendizaxe do currículo | Completo |
| * RA4 -  Crea bases de datos, e define a súa estrutura e as características dos seus elementos segundo o modelo relacional. | Si |

Obxectivos didácticos e título e descrición da actividade

| Obxectivos específicos | | Actividade | | Descrición básica | Duración |
| --- | --- | --- | --- | --- | --- |
| O2.1 | Modificar a estrutura de bases de datos e táboas seguindo as especificacións dos diagramas conceptuais e lóxicos ou atendendo a novos requirimentos, utilizando a LDD, asistentes e ferramentas gráficas. | A02 | Modificación da estrutura de bases de datos relacionais. | Modificarase a estrutura das bases de datos relacionais, táboas e relacións entre elas, utilizando a linguaxe de definición de datos. | 9 |

Criterios de avaliación

|  |
| --- |
| Criterios de avaliación |
| * CA4.2 -  Creáronse bases de datos.   CA4.2.2 - Modificáronse bases de datos.   * CA4.3 -  Creáronse as táboas e as relacións entre elas.   CA4.3.2 - Modificáronse táboas e relacións entre elas. |

Contidos

| Contidos |
| --- |
| * Creación, modificación e eliminación de bases de datos.   Modificación de bases de datos (CA 4.2.2).   * Creación, modificación e eliminación de táboas.   Modificación de táboas (CA 4.3.2). |

Actividades de ensino e aprendizaxe e de avaliación, métodos, recursos e instrumentos de avaliación

| Que e para que | Como | | | Con que | Como e con que se valora | Duración (sesións) |
| --- | --- | --- | --- | --- | --- | --- |
| Actividade (título e descrición) | Profesorado (en termos de tarefas) | Alumnado (tarefas) | Resultados  ou produtos | Recursos | Instrumentos e procedementos de avaliación |
| A02. Modificación da estrutura de bases de datos relacionais   * Modificarase a estrutura das bases de datos relacionais, táboas e relacións entre elas, utilizando a linguaxe de definición de datos. | * **Tp2.1** Explicación da sintaxe e utilización de sentenzas da linguaxe de definición de datos (LDD) para modificar as bases de datos, as táboas e as relacións entre elas. | * **Ta2.1** Modificación de estruturas de bases de datos relacionais, utilizando sentenzas da LDD. | * Guións de sentenzas da LDD coas solucións aos exercicios propostos. * Resultados da execución dos guións de sentenzas. | * Material didáctico subministrado polo profesorado en papel e/ou formato dixital. * Ordenadores con conexión a Internet, que terán instalado o sistema xestor de bases de datos MySQ L, o cliente MySQL Workbench e un programas de deseño de diagramas (tipo MS Visio ou Dia). * Proxector. * Manual de referencia de MySQL. |  | 8 |
|  | * **Ta2.2** Tarefa de avaliación utilizando os instrumentos de avaliación TO.2 e TO.4. | * Exame en papel e en formato dixital. * Resultado da execución dos guións de sentenzas. | * Máquina virtual específica para exame que terá instalado o sistema xestor de bases de datos MySQL e o cliente MySQL Workbench. * Manual de referencia de MySQL. | * **TO.2** - Documento de rexistro de modificación de bases de datos utilizando a LDD.Escala de valores (observación indirecta). * **TO.4 -** Documento de rexistro de modificación de táboas e das relacións entre elas utilizando a LDD.Escala de valores (observación indirecta). | 1 |

1. A02. Modificación da estrutura de bases de datos relacionais
   1. Introdución
      1. Obxectivos

O obxectivo desta actividade é modificar a estrutura de bases de datos e táboas seguindo as especificacións dos diagramas conceptuais e lóxicos ou atendendo a novos requirimentos, utilizando a linguaxe SQL de definición de datos (LDD ou DDL), e asistentes e ferramentas gráficas.

* + 1. Software e bases de datos

Utilizarase a plataforma WAMP (Windows-Apache-MySQL-PHP) WampServer 2.5 (última versión estable en outubro 2015), que inclúe MySQL Community Edition 5.6.17 como SXBDR (Sistema Xestor de Bases de Datos Relacional). As razóns de utilización deste software son que:

* É software libre, polo que o alumnado poderá descargalo de forma gratuíta e utilizalo legalmente na súa casa.
* É unha forma sinxela de facer a instalación do software necesario para desenvolver aplicacións web.
* Páxina oficial de  WampServer: <http://www.wampserver.com>
* Páxina oficial de MySQL: <https://www.mysql.com/>

Utilizarase MySQL Workbench 6.3 como ferramenta cliente gráfica para facer as prácticas asociadas a esta actividade, xa que é a recomendada por MySQL en outubro de 2015, aínda que tamén poderían utilizarse outras como phpMyAdmin, EMS MyManager, ou MySQL Query Browser.

* En [https://www.mysql.com/products/workbench/](https://www.mysql.com/products/workbench/%20) pode obterse información detallada sobre a ferramenta MySQL Workbench e descargar o software. En <http://dev.mysql.com/doc/index-gui.html> pode descargarse o manual de MySQL Workbench.
* O material auxiliar anexo a esta actividade inclúe unha Guía básica de MySQL Workbench 6.3.

As bases de datos sobre as que se farán as tarefas desta actividade, poden crearse utilizando os scripts que se subministran no material auxiliar anexo e correspóndense coas bases de datos creadas na actividade anterior 'A01 Creación da estrutura de bases de datos relacionais'.

* 1. Actividade
     1. Modificación dunha base de datos

A sintaxe da sentenza que permite cambiar as características globais dunha base de datos é:

ALTER {DATABASE | SCHEMA} nome\_da\_base [opcións\_a\_modificar] ...

Consideracións sobre a sintaxe anterior:

* As opcións para modificar son as mesmas que as opcións de creación:

[DEFAULT] CHARACTER SET [=] nome\_xogo\_carácteres

[DEFAULT] COLLATE [=] nome\_sistema\_ordenación

* O cambio do xogo de carácteres ou as opcións de cotexamento para unha base de datos na que xa existan táboas, non afecta a estas e só afecta ás que se creen no futuro.
* Tarefa 1. Modificar bases de datos empregando a sentenza ALTER.
  + 1. Modificación do esquema dunha táboa

Co paso do tempo é normal que se teñan que facer adaptacións no esquema das táboas das bases de datos, pola aparición de novos requirimentos, novas restricións, ou a desaparición dalgunhas das existentes. Algunhas veces, ter que facer cambios débese a non terlle dedicado o tempo suficiente ás fase de deseño conceptual e lóxico; é moi recomendable pararse a facer un bo deseño antes de empezar a escribir código para crear a base de datos.

A sentenza ALTER TABLE permite facer modificacións no esquema dunha táboa que xa existe na base de datos. Sintaxe:

ALTER TABLE nome\_táboa

[especificación\_alter [, especificación\_alter] …]

A *especificación\_alter* pode ser:

opcións de táboa

| ADD [COLUMN] nome\_columna definición\_columna [FIRST | AFTER nome\_columna]

| ADD [COLUMN] (nome\_columna definición\_columna, …)

| CHANGE [COLUMN] nome\_columna nome\_novo nova\_definición\_columna

[FIRST|AFTER nome\_columna],

| MODIFY [COLUMN] nome\_columna <nova\_definición\_columna>] [FIRST | AFTER nome\_columna],

| ALTER [COLUMN] nome\_columna {SET DEFAULT valor | DROP DEFAULT},

| ADD {INDEX|KEY} [nome\_índice] (columnas\_índice),

| ADD [CONSTRAINT [nome\_restrición]] PRIMARY KEY (lista\_columnas),

| ADD [CONSTRAINT [nome\_restrición]] UNIQUE {INDEX|KEY} [nome\_indice](lista\_columnas),

| ADD [CONSTRAINT [nome\_restrición]] FOREIGN KEY [nome\_índice](lista\_columnas)

REFERENCES nome\_táboa (lista\_de\_columnas)[ON DELETE opción][ON UPDATE opción]

| DROP [COLUMN] nome\_columna,

| DROP {INDEX | KEY} nome\_índice,

| DROP PRIMARY KEY,

| DROP FOREIGN KEY nome\_restrición,

| RENAME [TO | AS] nome\_táboa\_nova

| CONVERT TO CHARACTER SET xogo\_carácteres [COLLATE sistema\_colación]

Consideracións sobre a sintaxe:

* As opcións de táboa afectan ás características da táboa e son as mesmas que se utilizan na sentenza CREATE TABLE. Algunhas opcións de táboa son:

[DATA DIRECTORY= 'directorio']

[INDEX DIRECTORY= 'directorio']

[{ENGINE | TYPE} = {ISAM, MyISAM, InnodDB, ...}]

[[DEFAULT] CHARACTER SET nome\_xogo\_carácteres] [COLLATE nome\_sistema\_colación]]

[AUTO\_INCREMENT = número]

* DATA DIRECTORY e INDEX DIRECTORY permiten cambiar as rutas absolutas nas que se almacenan os datos e os índices.
* ENGINE permite cambiar o motor de almacenamento asociado á táboa. Ver o apartado "Motores de almacenamento en MySQL" que está máis adiante
* CHARACTER SET e COLLATE permiten cambiar o conxunto de carácteres e o sistema de colación predeterminados para as columnas que se crean nesa táboa. Non afecta ás columna que xa están creadas. Para cambiar o contido das columnas tipo cadea de carácteres, hai que utilizar a cláusula CONVERT TO CHARACTER SET.
* AUTO\_INCREMENT permite cambiar o número de comezo para a columna de tipo autoincremental.
* En xeral, as cláusulas ADD permiten engadir novas propiedades e as cláusulas DROP permiten eliminalas.
* Se ao engadir unha columna (ADD) non se especifica a cláusula FIRST | AFTER, a nova columna engádese ao final da táboa.
* Non é posible engadir unha columna tipo autoincremento se a táboa non está baleira.
* Non se pode modificar NULL por NOT NULL se a táboa contén valores nulos para a columna a modificar; a operación inversa non presenta ningún problema.
* Para cambiar a definición dunha columna, hai que utilizar a cláusula MODIFY, pero se ademais da definición tamén se quere cambiar o nome, entón hai que utilizar a cláusula CHANGE. O editor da actual versión de Workbench non recoñece a cláusula MODIFY e marca a liña como un erro aínda que se executa correctamente.
* A cláusula ALTER permite modificar o valor por defecto para unha columna.
* A cláusula RENAME permite cambiar o nome da táboa.
* A cláusula CONVERT TO CHARACTER SET permite cambiar o xogo de carácteres e o sistema de colación por defecto, e ademais, cambiar os valores de todas as columnas tipo cadea de carácteres (CHAR, VARCHAR e TEXT). Hai que ter en conta que non todos os sistemas de colación utilizan o mesmo número de bytes polo que pode ser necesario cambiar antes o tipo ou tamaño da columna.

Pódese facer a conversión só en algunha columna e non en todas as tipo cadea de carácteres, nese caso pódese utilizar a cláusula MODIFY. Exemplo:

**alter** **table** proba

**modify** **column** texto\_latin1 **varchar(**200**)** **character set utf8,**

No manual pódense consultar todas as opcións que ten esta sentenza.

Exemplo de modificación da estrutura da táboa *fabricante* da base de datos *practicas1*:

**alter** **table** fabricante

**add** **column** pais **varchar(**60**)** **default** **null,**

**add** **column** enderezo **varchar(**200**)** **not** **null** **after** idFabricante**,**

**add** **index** idx\_fabricante\_nome **(**nome)**,**

**engine = Innodb;**

* Tarefa 2. Modificar táboas empregando a sentenza ALTER.
  + 1. Engadir relacións e restricións de clave foránea

Pódense engadir relacións e restricións de clave foránea de dúas formas:

* Crear ao mesmo tempo as táboas e as relacións, empregando a sentenza CREATE TABLE. Neste caso hai que ter en conta a orde en que se crean as táboas, xa que non se pode crear unha táboa que conteña unha clave foránea se aínda non está creada a táboa á que fai referencia. Isto pode representar un problema no caso de relacións bidireccionais.
* Crear primeiro as táboas e establecer as relacións despois empregando sentenzas ALTER TABLE. Recoméndase esta segunda opción.

Por exemplo, supóñase que a entidade *empregado* está relacionada coa entidade *departamento* cunha relación *traballa* de tipo N:1, e *departamento* está relacionada con *empregado* coa relación *dirixe*, de tipo 1:1, con cardinalidade mínima 0. Ao crear primeiro a táboa *empregado* e despois *departamento*, provocaríase un erro porque na orde de creación da táboa *empregado* se define unha clave foránea que fai referencia á táboa *departamento*, que aínda non existe. O mesmo ocorre se empeza creando *departamento*.

En MySQL pódese solucionar o problema anterior desactivando a verificación de claves foráneas, poñendo o valor 0 ou OFF na variable *foreign\_key\_checks*.

u4a2_tarefa3_interelacion.emf

* Tarefa 3: Probar as posibilidades de creación de dúas táboas con interrelacións 1:N bidireccionais.
* Tarefa 4. Utilizar a sentenza ALTER TABLE para a creación das relacións entre táboas.
  + 1. Tarefa integradora sobre a LDD

A existencia desta tarefa está xustificada pola necesidade de realizar consecutivamente varias operacións sobre a base de datos que integre a maior parte das operacións en LDD estudadas ata o momento nesta unidade didáctica, como son: creación, borrado e modificación de bases de datos; creación, borrado e modificación de táboas e aplicación de restricións.

* Tarefa 5. Tarefa integradora sobre LDD.
  1. Tarefas

As tarefas propostas son as seguintes:

* Tarefa 1. Modificar bases de datos empregando a sentenza ALTER.
* Tarefa 2. Modificar táboas empregando a sentenza ALTER.
* Tarefa 3. Probar as posibilidades de creación de dúas táboas con interrelacións 1:N bidireccionais.
* Tarefa 4. Utilizar a sentenza ALTER TABLE para a creación das relacións entre táboas.
* Tarefa 5. Tarefa integradora sobre LDD.
  + 1. Tarefa 1. Modificar bases de datos empregando a sentenza ALTER

A tarefa consiste en cambiar o xogo de carácteres para a base de datos *practicas1*. O novo xogo de carácteres será latin1, co sistema de colación ‘general’.

Solución

**alter database** practicas1

**character set** latin1

**collate** latin1\_general\_ci**;**

* + 1. Tarefa 2. Modificar táboas empregando a sentenza ALTER

A tarefa consiste en realizar modificacións en táboas empregando a sentenza ALTER en varios casos:

* Tarefa 2.1. Facer os seguintes cambios na táboa *fabricante* da base de datos *practicas1*:
* Engadir a columna de tamaño variable *email*, asignándolle como valor por defecto o valor nulo.
* Engadir a columna de tamaño fixo *cif*, que non admite valores nulos, e colocala na segunda posición, despois da columna *idFabricante*.
* Engadir un índice asociado á columna *cif*.
* Cambiar o motor de almacenamento que ten actualmente, por Innodb.
* Tarefa 2.2. Facer os seguintes cambios na táboa *fabricante* da base de datos *practicas1*:
* Cambiar a clave primaria da táboa *fabricante*. A nova clave primaria pasa a ser a columna *cif*.
* Eliminar a columna *idFabricante*.
* Eliminar o índice asociado á columna *nome*.
* Tarefa 2.3. Facer os seguintes cambios na táboa *grupo* da base de datos *practicas1*:
* Cambiarlle o nome a columna *cache*. O novo nome é *custo*.
* Asignarlle á columna dataFormación como valor por defecto o valor cero.

Solución

* Tarefa 2.1

/\*cambios na táboa de fabricante\*/

**alter** **table** fabricante

**add** **column** email **varchar(**150**)** **default** **null,**

**add** **column** cif **char(**9**)** **not** **null** **after** idFabricante**,**

**add** **index** idx\_fabicante\_cif **(**cif**),**

**engine** **=** Innodb**;**

* Tarefa 2.2

/\*cambios na táboa de fabricante\*/

**alter** **table** fabricante

**add** **primary** **key** **(**cif**),**

**drop** **column** idFabricante**,**

**drop** **index** idx\_fabricante\_nome**;**

* Tarefa 2.3

/\*cambios na táboa grupo\*/

**alter** **table** grupo

**change** cache custo **mediumint** unsigned**,**

**alter** dataFormacion **set** **default** 0**;**

* + 1. Tarefa 3: Probar as posibilidades de creación de dúas táboas con interrelacións 1:N bidireccionais

A tarefa consiste en crear na base de datos *practicas1*, as táboas *empregado* e *departamento* que están relacionadas como se indica no seguinte diagrama E/R:

u4a2_tarefa3_interelacion.emf

Baixo os seguintes supostos:

* Tarefa 3.1. Crear a táboa empregado só sentenzas CREATE TABLE.
* Tarefa 3.2. Crear primeiro as dúas táboas, sen claves foráneas, e despois engadir as claves foránea que establecen a relación entre elas coa axuda da sentenza ALTER TABLE. Borrar primeiro as táboas, se xa existiran.
* Tarefa 3.3. Desactivar a verificación de claves foráneas, coa variable *foreign\_key\_checks*, e despois facer a creación das táboas. Borrar primeiro as táboas, se xa existiran.

Solución

* Tarefa 3.1

Este primeiro suposto vai dar lugar a unha mensaxe de erro, debido a que cando se crea a táboa empregado aínda non existe a táboa departamento.

**use** practicas1**;**

**drop** **table** **if** **exists** empregado**;**

**drop** **table** **if** **exists** departamento**;**

**create** **table** empregado

**(**

dni **char(**9**)** **not** **null,**

nss **char(**10**)** **not** **null,**

nome **varchar(**80**)** **not** **null,**

dataNacemento **date,**

sexo enum**(**'h'**,**'m'**)** **comment** 'h = home, m = muller'**,**

salario **decimal(**11**,**2**)** unsigned**,**

**primary** **key** **(**dni**),**

**index** idx\_empregado\_nss **(**nss**),**

codigoDepartamento **smallint** unsigned**,**

**constraint** fk\_empregado\_departamento **foreign** **key** **(**codigoDepartamento**)**

**references** departamento**(**codigoDepartamento**)**

**on** **delete** **set** **null**

**on** **update** **cascade**

**);**

**create** **table** departamento

**(**

codigoDepartamento **smallint** unsigned auto\_increment **not** **null,**

nome **varchar(**60**)** **not** **null,**

localizacion **varchar(**40**),**

dniXefe **char(**9**),**

**primary** **key** **(**codigoDepartamento**),**

**constraint** fk\_departamento\_empregado **foreign** **key** **(**dniXefe**)** **references** empregado**(**dni**)**

**on** **delete** **set** **null**

**on** **update** **cascade**

**);**

Cando se executa a sentenza *create table empleado* prodúcese un erro porque a clave foránea *codigoDepartamento* fai referencia a unha táboa que aínda non existe. Se mostra a seguinte mensaxe de erro:

Error Code: 1215. No puede adicionar clave extranjera constraint

* Tarefa 3.2

Este segundo suposto solucionaría o problema anterior creando primeiro as táboas e despois engadindo as relacións coa axuda da sentenza ALTER.

**use** practicas1**;**

**drop** **table** **if** **exists** empregado**;**

**drop** **table** **if** **exists** departamento**;**

/\*creación das táboas\*/

**create** **table** empregado

**(**

dni **char(**9**)** **not** **null,**

nss **char(**10**)** **not** **null,**

nome **varchar(**80**)** **not** **null,**

dataNacemento **date,**

sexo enum**(**'h'**,**'m'**)** **comment** 'h = home, m = muller'**,**

salario **decimal(**11**,**2**)** unsigned**,**

codigoDepartamento **smallint** unsigned**,**

**primary** **key** **(**dni**),**

**index** idx\_empregado\_nss **(**nss**)**

**);**

**create** **table** departamento

**(**

codigoDepartamento **smallint** unsigned auto\_increment **not** **null,**

nome **varchar(**60**)** **not** **null,**

localizacion **varchar(**40**),**

dniXefe **char(**9**),**

**primary** **key** **(**codigoDepartamento**)**

**);**

/\*engadir as restriccións de clave foránea par reprentar as relacións \*/

**alter** **table** empregado

**add** **constraint** fk\_empregado\_departamento **foreign** **key** **(**codigoDepartamento**)**

**references** departamento**(**codigoDepartamento**)**

**on** **delete** **set** **null**

**on** **update** **cascade;**

**alter** **table** departamento

**add** **constraint** fk\_departamento\_empregado **foreign** **key** **(**dniXefe**)**

**references** empregado**(**dni**)**

**on** **delete** **set** **null**

**on** **update** **cascade;**

* Tarefa 3.3

Este terceiro suposto solucionaría o problema desactivando a verificación de claves foráneas antes da creación das táboas.

/\*desactivar a verificación de claves foráneas \*/

**set** FOREIGN\_KEY\_CHECKS **=** 0**;**

**use** practicas1**;**

**drop** **table** **if** **exists** empregado**;**

**drop** **table** **if** **exists** departamento**;**

**create** **table** empregado

**(**

dni **char(**9**)** **not** **null,**

nss **char(**10**)** **not** **null,**

nome **varchar(**80**)** **not** **null,**

dataNacemento **date,**

sexo enum**(**'h'**,**'m'**)** **comment** 'h = home, m = muller'**,**

salario **decimal(**11**,**2**)** unsigned**,**

**primary** **key** **(**dni**),**

**index** idx\_empregado\_nss **(**nss**),**

codigoDepartamento **smallint** unsigned**,**

**constraint** fk\_empregado\_departamento **foreign** **key** **(**codigoDepartamento**)**

**references** departamento**(**codigoDepartamento**)**

**on** **delete** **set** **null**

**on** **update** **cascade**

**);**

**create** **table** departamento

**(**

codigoDepartamento **smallint** unsigned auto\_increment **not** **null,**

nome **varchar(**60**)** **not** **null,**

localizacion **varchar(**40**),**

dniXefe **char(**9**),**

**primary** **key** **(**codigoDepartamento**),**

**constraint** fk\_departamento\_empregado **foreign** **key** **(**dniXefe**)**

**references** empregado**(**dni**)**

**on** **delete** **set** **null**

**on** **update** **cascade**

**);**

/\*activar a verificación de claves foráneas\*/

**set** FOREIGN\_KEY\_CHECKS **=** 1**;**

* + 1. Tarefa 4. Utilizar a sentenza ALTER TABLE para a creación das relacións entre táboas

A tarefa consiste en crear o esquema dunha base de datos creando primeiro as táboas e definindo despois as relacións entre eles en dous supostos:

* Tarefa 4.1. Crear a base de datos *eleccionModulos* e crear nela as táboas *profesor*, *grupo*, *modulo* e *imparte*, segundo a seguinte descrición:
* Táboa profesor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome | Tipo | Null | Clave | Observacións |
| idProfesor | char(15) | Non | Primary | Nome curto do profesor |
| corpo | enum | Non |  | Corpo ao que pertence. Valores permitidos:  **S** = Profesorado Secundaria **T** = Profesorado Técnico |
| nrp | char(20) | Non | Única | Número de Rexistro Persoal |
| nome | varchar(30) | Non |  |  |
| apelidos | varchar(60) | Non | Indice |  |
| horasLectivas | smallint (sen signo) |  |  | Nº de horas de clase que imparte o profesor. Por defecto: 18 |

* Táboa *grupo*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome | Tipo | Null | Clave | Observacións |
| idGrupo | char(15) | Non | Primary | Clave para identificar o grupo |
| descricion | varchar(100) | Non |  | Descrición do nome do grupo |
| horasTitoria | smallint (sen signo) |  |  | Nº de horas de redución para o profesor titor |
| idProfesor | char(15) |  |  | Identificador do profesor que é titor do grupo |

* Táboa *modulo*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome | Tipo | Null | Clave | Observacións |
| idModulo | char(6) | Non | Primary | Clave para identificar o módulo |
| horasSemanais | tinyint (sen signo) |  |  | Descrición do nome do grupo |
| descricion | varchar(100) |  |  | Nº de horas de redución para o profesor titor |
| corpo | enum | Non |  | Corpo dos profesores que poden impartir o módulo. Valores permitidos: **S** = Profesorado Secundaria **T** = Profesorado Técnico |

* Táboa *imparte* (clave primaria composta)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome | Tipo | Null | Clave | Observacións |
| idModulo | char(6) | Non | Primary | Clave para identificar o módulo |
| idGrupo | char(15) | Non | Clave para identificar o grupo |
| idProfesor | char(15) | Non | Nome curto do profesor |

* Tarefa 4.2. Definir as relacións entre esas táboas tendo en conta o seguinte grafo relacional.

u4a2_tarefa4_1_grafo.emf

Solución

* Tarefa 4.1

/\*borrado da base de datos, se existe (só cando se están facendo probas)\*/

**drop** **database** **if** **exists** eleccionModulos**;**

/\*creación da base de datos se non existe\*/

**create** **database** **if** **not** **exists** eleccionModulos

**character** **set** utf8

**collate** utf8\_spanish\_ci**;**

/\* ou tamén \*/

**create** **schema** **if** **not** **exists** eleccionModulos

**character** **set** utf8

**collate** utf8\_spanish\_ci**;**

/\*activar vase de datos\*/

**use** eleccionModulos**;**

/\* TÁBOA profesor \*/

**create** **table** profesor **(**

idProfesor **char(**15**)** **not** **null,**

corpo enum**(**'S'**,**'T'**)** **not** **null,**

nrp **char(**20**)** **not** **null,**

nome **varchar(**30**)** **not** **null,**

apelidos **varchar(**60**)** **not** **null,**

horasLectivas **smallint** unsigned **default** 18**,**

**primary** **key** **(**idProfesor**),**

**unique** **key** **(**nrp**)**

**)engine=**InnoDB**;**

/\* TÁBOA grupo\*/

**create** **table** grupo **(**

idGrupo **char(**15**),**

descricion **varchar(**100**)** **not** **null,**

horasTitoria **smallint** unsigned**,**

idProfesor **char(**15**),**

**primary** **key** **(**idGrupo**)**

**)engine=**InnoDB**;**

/\* TÁBOA modulo\*/

**create** **table** modulo **(**

idModulo **char(**6**),**

horasSemanais **smallint** unsigned **,**

descricion **varchar(**100**)** **not** **null,**

corpo enum**(**'S'**,**'T'**),**

**primary** **key** **(**idModulo**)**

**)engine=**InnoDB**;**

/\* TÁBOA imparte \*/

**create** **table** imparte **(**

idModulo **char(**6**),**

idGrupo **char(**15**),**

idProfesor **char(**15**),**

**primary** **key** **(**idGrupo**,**idModulo**,**idProfesor**)**

**)engine=**InnoDB**;**

* Tarefa 4.2

/\* Definición das relacións entre as táboas: RESTRICIÓNS DE CLAVE FORÁNEA \*/

**alter** **table** grupo

**add** **constraint** fk\_grupo\_profesor **foreign** **key** **(**idprofesor**)**

**references** profesor **(**idprofesor**)**

**on** **delete** **restrict**

**on** **update** **cascade;**

**alter** **table** imparte

**add** **constraint** fk\_imparte\_modulo **foreign** **key** **(**idmodulo**)**

**references** modulo **(**idmodulo**)**

**on** **delete** **restrict**

**on** **update** **cascade,**

**add** **constraint** fk\_imparte\_profesor **foreign** **key** **(**idprofesor**)**

**references** profesor **(**idprofesor**)**

**on** **delete** **restrict**

**on** **update** **cascade,**

**add** **constraint** fk\_impate\_grupo **foreign** **key** **(**idgrupo**)**

**references** grupo **(**idgrupo**)**

**on** **delete** **restrict**

**on** **update** **cascade;**

* + 1. Tarefa 5: Tarefa integradora sobre LDD

A tarefa consiste en realizar as tarefas integradoras nos seguintes supostos:

* Tarefa 5.1. Na táboa *pelicula* da base de datos *practicas1* hai unha columna chamada *xenero* que garda o xénero da película. Decídese empregar un sistema de codificación de xéneros cinematográficos.

Se pide: crear unha táboa co sistema de codificación de xéneros, e modificar a columna *xenero* para utilizala como unha clave foránea que fai referencia a esa táboa.

A táboa de xéneros terá tres columnas, unha co código numérico (actualmente hai uns 35 códigos recoñecidos), outra cunha descrición curta, e outra cunha descrición máis detallada do xénero.

* Tarefa 5.2. Cambiar a clave primaria na táboa de *empregado* da base de datos *practicas1*. Actualmente é a columna *dni*, e se quere cambiar pola columna *nss*.

Hai que ter en conta que esa clave primaria é referenciada pola clave foránea *dniXefe* da táboa *departamento*, polo que será necesario facer unha serie de operacións previas ao cambio de clave primaria. Na táboa departamento xa non interesa ter o dni do xefe, senón o seu número de seguridade social (nss).

* Tarefa 5.3. Modificar a estrutura da base de datos *proxectos* (creada na actividade anterior), engadindo as columnas, e as restricións de integridade referencial necesarias, tendo en conta o seguinte grafo relacional:

exercico proxecto_MR.emf

Solución

* Tarefa 5.1

**use** practicas1**;**

**drop** **table** **if** **exists** xenero**;**

/\*creacion da táboa xenero\*/

**create** **table** xenero

**(**

codXenero **tinyint** unsigned zerofill **not** **null,**

descricionBreve **char(**15**)** **not** **null,**

descricionLarga **varchar(**150**)** **not** **null,**

**primary** **key** **(**codXenero**)**

**);**

/\*facer as modificacións necesarias na táboa pelicula\*/

**alter** **table** pelicula

**modify** **column** xenero **tinyint** unsigned zerofill**,**

**add** **constraint** fk\_pelicula\_xenero **foreign** **key** **(**xenero**)**

**references** xenero**(**codXenero**)**

**on** **delete** **restrict**

**on** **update** **cascade;**

* Tarefa 5.2

/\*activación da base de datos practicas1\*/

**use** practicas1**;**

/\*eliminar as claves foránes que fan referencia á clave primaria que se vai cambiar\*/

**alter** **table** departamento

**drop** **foreign** **key** fk\_departamento\_empregado**;**

/\*cambiar a clave primaria na táboa empregado\*/

**alter** **table** empregado

**drop** **primary** **key,**

**add** **primary** **key** **(**nss**);**

/\*cambiar a definición da clave foránea que fai ferencia á nova clave primaria

para que pertenezan ao mesmo dominio, e volver a crear a clave foranea\*/

**alter** **table** departamento

**change** dniXefe nssXefe **char(**10**),**

**add** **constraint** fk\_departamento\_empregado **foreign** **key** **(**nssXefe**)**

**references** empregado**(**nss**)**

**on** **delete** **set** **null**

**on** **update** **cascade;**

No caso de intentar borrar a clave primaria na táboa de empregado antes de borrar as claves foráneas que fan referencia a ela, móstrase a seguinte mensaxe de erro:

/\*Error Code: 1025. Error en el renombrado de '.\practicas1\#sql-e80\_2' a '.\practicas1\empregado' (Error: 150 - Foreign key constraint is incorrectly formed)

* Tarefa 5.3

**use** proxectos**;**

-- RESTRICIÓNS DE INTEGRIDADE REFERENCIAL. ATRIBUTOS MULTIVALUADOS

-- Atributo multivaluado telefono na entidade persoal

**alter** **table** telefono

**add** **constraint** fk\_telefono\_persoal **foreign** **key** **(**dni**)**

**references** persoal **(**dni**)**

**on** **delete** **cascade**

**on** **update** **cascade;**

-- Atributo multivaluado localizacion na entidade departamento

**alter** **table** localizacion

**add** **constraint** fk\_localizacion\_departamento **foreign** **key** **(**codDepartamento**)**

**references** departamento **(**codDepartamento**)**

**on** **delete** **cascade**

**on** **update** **cascade;**

-- RESTRICIÓNS DE INTEGRIDADE REFERENCIAL. INTERRELACIÓNS 1:N

-- Relación traballa (1:N). Creación da columna e definción de clave foránea

**alter** **table** empregado

**add** **column** deptoTraballa **smallint** unsigned**,**

**add** **constraint** fk\_empregado\_departamento **foreign** **key** **(**deptoTraballa**)**

**references** departamento **(**codDepartamento**)**

**on** **delete** **restrict**

**on** **update** **restrict;**

-- Relación organiza (1:N). Creación da columna e definción de clave foránea

-- edicion é entidade débil: Defición de clave foránea coa entidade forte

**alter** **table** edicion

**add** **column** deptoOrganizador **smallint** unsigned **not** **null,**

**add** **constraint** fk\_edicion\_curso **foreign** **key** **(**codCurso**)**

**references** curso **(**codCurso**)**

**on** **delete** **cascade**

**on** **update** **cascade,**

**add** **constraint** fk\_edicion\_departamento **foreign** **key** **(**deptoOrganizador**)**

**references** departamento **(**codDepartamento**)**

**on** **delete** **restrict**

**on** **update** **cascade;**

-- Relación controla (1:N). Creación da columna e definción de clave foránea

-- Relación componse (1:N). Creación da columna e definción de clave foránea

**alter** **table** proxecto

**add** **column** deptoControl **smallint** unsigned **not** **null,**

**add** **column** componseSubproxecto **smallint** unsigned **not** **null,**

**add** **constraint** fk\_proxecto\_departamento **foreign** **key** **(**deptoControl**)**

**references** departamento **(**codDepartamento**)**

**on** **delete** **restrict**

**on** **update** **restrict,**

**add** **constraint** fk\_proxecto\_proxecto **foreign** **key** **(**componseSubproxecto**)**

**references** proxecto **(**codProxecto**)**

**on** **delete** **restrict**

**on** **update** **cascade;**

-- RESTRICIÓNS DE INTEGRIDADE REFERENCIAL. INTERRELACIÓNS 1:1

-- Relación dirixe (1:1). Creación da columna e definción de clave foránea

**alter** **table** departamento

**add** **column** dniDirixe **integer(**8**)** unsigned **not** **null,**

**add** **column** dataDirixe **date** **not** **null,**

**add** **constraint** fk\_departamento\_empregado **foreign** **key** **(**dniDirixe**)**

**references** empregado **(**dni**)**

**on** **delete** **restrict**

**on** **update** **restrict;**

-- RESTRICIÓNS DE INTEGRIDADE REFERENCIAL. INTERRELACIÓNS N:M

-- Relación asiste (N:M). Definción de claves foráneas

**alter** **table** asiste

**add** **constraint** fk\_asiste\_persoal **foreign** **key** **(**dniPersoal**)**

**references** persoal **(**dni**)**

**on** **delete** **restrict**

**on** **update** **cascade,**

**add** **constraint** fk\_asiste\_edición **foreign** **key** **(**codCurso**,**codEdicion**)**

**references** edicion **(**codCurso**,**codEdicion**)**

**on** **delete** **restrict**

**on** **update** **cascade;**

-- Relación participa (N:M). Definción de claves foráneas

**alter** **table** participa

**add** **constraint** fk\_participa\_proxecto **foreign** **key** **(**codProxecto**)**

**references** proxecto **(**codProxecto**)**

**on** **delete** **restrict**

**on** **update** **restrict,**

**add** **constraint** fk\_participa\_empregado **foreign** **key** **(**dniEmpregado**)**

**references** empregado **(**dni**)**

**on** **delete** **restrict**

**on** **update** **restrict;**

-- Relación documenta (N:M). Definción de claves foráneas

**alter** **table** documenta

**add** **constraint** fk\_documenta\_proxecto **foreign** **key** **(**codProxecto**)**

**references** proxecto **(**codProxecto**)**

**on** **delete** **restrict**

**on** **update** **cascade,**

**add** **constraint** fk\_documenta\_persoal **foreign** **key** **(**dniPracticas**)**

**references** persoal **(**dni**)**

**on** **delete** **restrict**

**on** **update** **cascade;**

-- Relación podesupervisar (N:M). Definción de claves foráneas

**alter** **table** podesupervisar

**add** **constraint** fk\_podesupervisar\_proxecto **foreign** **key** **(**codProxecto**)**

**references** proxecto **(**codProxecto**)**

**on** **delete** **restrict**

**on** **update** **cascade,**

**add** **constraint** fk\_podesupervisar\_empregado **foreign** **key** **(**dniEmpregado**)**

**references** empregado **(**dni**)**

**on** **delete** **restrict**

**on** **update** **cascade;**

-- RESTRICIÓNS DE INTEGRIDADE REFERENCIAL. INTERRELACIÓNS 1:N:M

-- Relación supervisa (1:N:M). Definción de claves foráneas

**alter** **table** supervisa

**add** **constraint** fk\_supervisa\_proxecto **foreign** **key** **(**codProxecto**)**

**references** proxecto **(**codProxecto**)**

**on** **delete** **restrict**

**on** **update** **cascade,**

**add** **constraint** fk\_supervisa\_empregado\_supervisor **foreign** **key** **(**dniSupervisor**)**

**references** empregado **(**dni**)**

**on** **delete** **restrict**

**on** **update** **cascade,**

**add** **constraint** fk\_supervisa\_empregado\_supervisado **foreign** **key** **(**dniSupervisado**)**

**references** empregado **(**dni**)**

**on** **delete** **restrict**

**on** **update** **cascade;**

-- ESPECIALIZACIÓN:

**alter** **table** persoal

**add** **column** tipo enum**(**'practicas'**,**'empregado'**)** **not** **null;**

**alter** **table** empregado

**add** **constraint** fk\_empregado\_persoal **foreign** **key** **(**dni**)**

**references** persoal **(**dni**)**

**on** **delete** **restrict**

**on** **update** **restrict;**

/\* INCLUSION. Para esta restrición pódense utilizar disparadores (triggers), asercións, restriccións CHECK,ou definir claves foráneas. Como MySQL non ten implementada a cláusula CHECK, e os disparadores e asercións estúdianse mais adiante, nesta actividade utilízase a última das opcións (claves foráneas)\*/

-- Relacións dirixe e traballa (definindo claves foráneas)

-- Comprobar que o empregado que dirixe o departamento traballa nese departamento.

**alter** **table** empregado

**add** **index** idx\_departamento\_dni\_codDepartamento **(**dni**,**deptoTraballa**);**

**alter** **table** departamento

**add** **constraint** fk\_departamento\_empregado\_traballa

**foreign** **key** **(**dniDirixe**,**codDepartamento**)**

**references** empregado **(**dni**,**deptoTraballa**)**

**on** **delete** **restrict**

**on** **update** **restrict;**

-- Relacións supervisa e podesupervisar (definindo claves foráneas)

-- Comprobar que o empregado que supervisa o proxecto pode supervisalo.

**alter** **table** podesupervisar

**add** **index** idx\_podesupervisar\_codProxecto\_dniEmpregado **(**codProxecto**,**dniEmpregado**);**

**alter** **table** supervisa

**add** **constraint** fk\_supervisa\_podesupervisar **foreign** **key** **(**codProxecto**,**dniSupervisor**)**

**references** podesupervisar **(**codProxecto**,**dniEmpregado**)**

**on** **delete** **restrict**

**on** **update** **restrict;**

1. Materiais
   1. Documentos de apoio ou referencia

* DUBOIS, Paul. *La biblia de MySQL*. Anaya Multimedia, 2009.
* GROFF, James.*SQL.* McGraw-Hill.
* HUESO IBAÑEZ, Luis. *Bases de datos*. Ciclos Formativos Ra-Ma, 2012.
* *Manual de referencia de MySQL 5.6*. <http://dev.mysql.com/doc/>
* CAMPS PARÉ, Rafael, CASILLAS SANTILLÁN, Luís Alberto, COSTAL COSTA, Dolors, GILBERT GINESTÁ, Marc, MARTÍN ESCOFET, Carme, PÉREZ MORA, Óscar. *Bases de datos.* UOC, 2007.

<http://ocw.uoc.edu/informatica-tecnologia-y-multimedia/bases-de-datos/Course_listing>

* POZO CORONADO, Salvador. *MySQL con Clase.*

[http://mysql.conclase.net/curso/?cap=000#](http://mysql.conclase.net/curso/?cap=000)

* 1. Recursos didácticos
* Material didáctico subministrado polo profesorado en papel e/ou formato dixital.
* Ordenadores con conexión a Internet, que terán instalado o sistema xestor de bases de datos MySQL, o cliente MySQL Workbench e un programas de deseño de diagramas (tipo MS Visio ou Dia).
* Máquina virtual específica para exame que terá instalado o sistema xestor de bases de datos MySQL e o cliente MySQL Workbench.
* Proxector.
* Manual de referencia de MySQL.
  1. Material auxiliar

O material auxiliar anexo a esta actividade está almacenado na carpeta CSIFC02\_MP0484\_V000402\_ModBD\_Anexos que contén:

* O arquivo V000402\_scriptsSQL.zip. cos scripts SQL para a creación das bases de datos e as táboas das tarefas realizadas na actividade 'A01 Creación da estrutura de bases de datos relacionais', utilizadas como base para esta actividade.
* O arquivo GuiaWorkbench.docx que é unha guía básica de MySQL Workbench 6.3.

1. Avaliación

Criterios de avaliación

|  |  |  |  |
| --- | --- | --- | --- |
| Criterios de avaliación seleccionados  para esta actividade | Tipo de  evidencia de aprendizaxe | Instrumento de avaliación | Peso na cualificación  da UD |
| * CA4.2 -  Creáronse bases de datos.   CA4.2.2 - Modificáronse bases de datos. | * Exame en formato dixital e en papel. | * **TO.2** - Documento de rexistro de modificación de bases de datos utilizando a LDD. | 6 |
| * CA4.3 -  Creáronse as táboas e as relacións entre elas.   CA4.3.2 -  Modificáronse táboas e relacións entre elas. | * **TO.4 -** Documento de rexistro de modificación de táboas e das relacións entre elas utilizando a LDD. | 24 |

Modelo de proba combinada para TO.2 e TO.4

Proponse unha proba práctica para avaliar CA4.2.2 e CA4.3.2 mediante observación indirecta sobre un documento de rexistro da práctica. A solución deberá ser entregada polo alumnado nun documento escrito e en dixital.

* O documento escrito debe conter:
* Bloque de sentenzas solicitado.
* Imaxe capturada co resultado da execución do bloque dende o cliente Workbench.
* O arquivo dixital debe conter:
* Arquivos .sql co código correspondente ao bloque de sentenzas.

A proba consistirá en escribir guións de sentenzas SQL que permitan realizar modificacións sobre a base de datos *musica.* Subminístrase un script sql que permite a creación da base de datos e a seguinte información auxiliar sobre as táboas da base de datos.

* Táboa CANCION con motor de almacenamento innodb:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome | Tipo | Null | Clave | Observacións |
| id\_disco | smallint (sin signo) | N | P | Código identificativo do disco |
| numero\_pista | tinyint (sin signo) | N |  |
| titulo | varchar(90) | N |  |  |
| duración | decimal(5,2) |  |  | Duración da canción en minutos e segundos |

* Táboa DISCO con motor de almacenamento innodb:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome | Tipo | Null | Clave | Observacións |
| id | smallint (sin signo) | N | P | Código que identifica ao disco |
| titulo | varchar(80) | N |  |  |
| autor | char(3) |  |  | Código do autor do disco |
| id\_discografica | char(5) | N |  | Código que identifica á discográfica que edita o disco |

* Táboa DISCOGRAFICA con motor de almacenamento innodb:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome | Tipo | Null | Clave | Observacións |
| id | char(5) | N | P | Código da discográfica |
| nombre | varchar(80) | N |  |  |

Proba para CA4.2.2 sobre modificación de bases de datos

* 1.- Cambiar o xogo de carácteres por defecto (latin1) e o sistema de colación (latin1 general) para a base de datos.

Proba para CA4.3.2 sobre modificación de táboas

* 2.- Facer as modificacións necesarias nas táboas da base de datos para representar as relacións entre elas segundo o que se representa no seguinte grafo relacional.

u4a2_grafo_exame.emf

* 3.- Crear a táboa *autor* con motor de almacenamento MyISAM, e os seguintes atributos:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome | Tipo | Null | Clave | Observacións |
| codigo | char(3) | N | Primary | Código do autor |
| nome | varchar(70) | N |  | Nome do autor |

A creación de táboas non se avalía nesta actividade pero é necesaria para desenvolver o resto da proba, polo que o profesorado poderá axudar na escritura do script ao alumnado que o precise. Despois da creación, deberase cambiar o motor de almacenamento da táboa *autor* e poñerlle o motor Innodb.

* 4.- Engadir a restrición de clave foránea na táboa *disco* para que a columna do código do autor da táboa *disco* faga referencia á táboa *autor* que se acaba de crear.
* 5.- Engadir unha columna na táboa *cancion* co código do autor da canción, colocándoa entre as columnas *titulo* e *duracion*. Define esa nova columna como unha clave foránea que fai referencia á táboa *autor*.
* 6.- Eliminar a columna *id* na táboa *disco*, que é clave primaria, e engadir a columna *isrc* (*International Standard Recording Code*, que é un código, de tipo char(12), estándar de carácter internacional para identificar de forma única gravacións de audio e vídeos musicais). A columna *isrc* será a partir de este momento a clave primaria.
* 7.- Cambiar o tamaño da columna *titulo* na táboa *cancion*, a un máximo de 80 en lugar dos 90 que tiña, e borrar a columna *duración*.

Exemplo de solución para entregar en papel

Proponse a seguinte solución:

* 1.- Cambiar o xogo de carácteres por defecto (latin1) e o sistema de colación (latin1 general) para a base de datos.

**alter** **database** musica

**default character set** latin1 **collate** latin1\_general\_ci;

* 2.- Facer as modificacións necesarias nas táboas da base de datos para representar as relacións entre elas segundo o que se representa no seguinte grafo relacional.

**use** musica**;**

-- crear claves foráneas

**alter** **table** disco

**add** **constraint** fk\_disco\_id\_discografica **foreign** **key** **(**id\_discografica**)**

**references** discografica **(**id**)**

**on** **delete** **restrict**

**on** **update** **cascade;**

**alter** **table** cancion

**add** **constraint** fk\_cancion\_id\_disco **foreign** **key** **(**id\_disco**)** **references** disco **(**id**)**

**on** **delete** **restrict**

**on** **update** **cascade;**

3.- Crear a táboa *autor* con motor de almacenamento MyISAM e despois cambiar o motor de almacenamento da táboa *autor* e poñerlle o motor Innodb.

-- crear táboa autor

**create** **table** autor

**(**

codigo **char(**3**)** **not** **null,**

nome **varchar(**70**)** **not** **null,**

**primary** **key** **(**codigo**)**

**)** **engine** **=** myisam**;**

-- cambiar motor de almacenamento

**alter** **table** autor

**engine** **=** innodb**;**

* 4.- Engadir a restrición de clave foránea na táboa *disco* para que a columna do código do autor da táboa *disco* faga referencia á táboa *autor* que se acaba de crear.

-- engadir restricción de clave foránea

**alter** **table** disco

**add** **constraint** fk\_disco\_autor **foreign** **key** **(**autor**)** **references** autor **(**codigo**)**

**on** **delete** **restrict**

**on** **update** **cascade;**

* 5.- Engadir unha columna na táboa *cancion* co código do autor da canción, colocándoa entre as columnas *titulo* e *duracion*. Define esa nova columna como unha clave foránea que fai referencia á táboa *autor*.

-- engadir columna, e definila como clave foranea

**alter** **table** cancion

**add** **column** autor **char(**3**)** **after** titulo**;**

**alter** **table** cancion

**add** **constraint** fk\_cancion\_autor **foreign** **key** **(**autor**)** **references** autor **(**codigo**)**

**on** **delete** **restrict**

**on** **update** **cascade;**

* 6.- Eliminar a columna *id* na táboa *disco*, que é clave primaria, e engadir a columna *isrc* (*International Standard Recording Code*, que é un código, de tipo char(12), estándar de carácter internacional para identificar de forma única gravacións de audio e vídeos musicais). A columna *isrc* será a partir de este momento a clave primaria.

-- cambiar clave primaria referenciada por outras táboas

**alter** **table** cancion

**drop** **primary** **key,**

**drop** **foreign** **key** fk\_cancion\_id\_disco**,**

**drop** **column** id\_disco**;**

**alter** **table** disco

**drop** **primary** **key,**

**drop** **column** id**;**

**alter** **table** disco

**add** **column** isrc **char(**12**)** **not** **null,**

**add** **primary** **key** **(**isrc**);**

**alter** **table** cancion

**add** **column** isrc **char(**14**),**

**add** **primary** **key** **(**isrc**,** numero\_pista**),**

**add** **constraint** fk\_cancion\_isrc **foreign** **key** **(**isrc**)** **references** disco **(**isrc**)**

**on** **delete** **restrict**

**on** **update** **cascade;**

* 7.- Cambiar o tamaño da columna *titulo* na táboa *cancion*, a un máximo de 80 en lugar dos 90 que tiña, e borrar a columna *duración*.

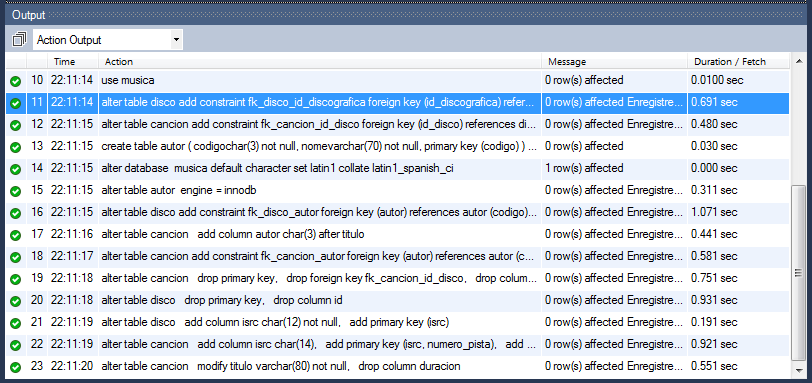
-- cambiar tamaño columna

**alter** **table** cancion

**modify** titulo **varchar(**80**)** **not** **null,**

**drop** **column** duracion**;**

A imaxe capturada dende a zona de saída (Output) do cliente Workbench co resultado da execución do bloque de sentenzas é:



Exemplo de lista de valoración para TO.2

Proponse a seguinte lista de valoración para o instrumento de avaliación TO.2 (EV) (observación indirecta: documento de rexistro de modificación de bases de datos utilizando a LDD).

| Nome | Data | |
| --- | --- | --- |
| Indicadores para  CA4.2.2 - Modificáronse bases de datos. | Valoración máxima  6 | Cualificación |
| Modificáronse as bases de datos coas novas especificacións. | 6 |  |

Exemplo de lista de valoración para TO.4

Proponse a seguinte lista de valoración para o instrumento de avaliación TO.4 (EV) (observación indirecta: documento de rexistro de modificación de táboas e das relacións entre elas utilizando a LDD).

| Nome | Data | |
| --- | --- | --- |
| Indicadores para  CA4.3.2 -  Modificáronse táboas e relacións entre elas. | Valoración máxima  24 | Cualificación |
| Engadíronse táboas. | 6 |  |
| Borráronse táboas. | 6 |  |
| Modificáronse táboas. | 6 |  |
| Aplicáronse novas restricións. | 6 |  |