Dirección Xeral de Educación, Formación Profesional e Innovación Educativa

Material para

a formación profesional inicial

A03. Verificación e probas

|  |  |  |
| --- | --- | --- |
| 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** | **A03** | **Verificación e probas** |
| 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\_V000403\_UD04\_A03\_ProbasBD.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 5](#_Toc443241553)

[Contexto da actividade 5](#_Toc443241554)

[Título da actividade 6](#_Toc443241555)

[Resultados de aprendizaxe do currículo 6](#_Toc443241556)

[Obxectivos didácticos e título e descrición da actividade 6](#_Toc443241557)

[Criterios de avaliación 6](#_Toc443241558)

[Contidos 6](#_Toc443241559)

[Actividades de ensino e aprendizaxe e de avaliación, métodos, recursos e instrumentos de avaliación 7](#_Toc443241560)

[2. A03. Verificación e probas de datos 8](#_Toc443241561)

[2.1 Introdución 8](#_Toc443241562)

[2.1.1 Obxectivos 8](#_Toc443241563)

[2.1.2 Software 8](#_Toc443241564)

[2.1.3 Bases de datos de traballo 8](#_Toc443241565)

[2.1.3.1 Base de datos eleccionModulos 9](#_Toc443241566)

[2.1.3.2 Base de datos gabinete 9](#_Toc443241567)

[2.2 Actividade 10](#_Toc443241568)

[2.2.1 Introdución 10](#_Toc443241569)

[2.2.2 Modo do servidor MySQL 11](#_Toc443241570)

[2.2.3 Edición de datos con MySQL Workbench 12](#_Toc443241571)

[2.2.4 Integridade e restricións 13](#_Toc443241572)

[2.2.4.1 Verificar restrición PRIMARY KEY 14](#_Toc443241573)

[2.2.4.2 Verificar restrición de integridade referencial 15](#_Toc443241574)

[2.2.4.3 Verificar restrición UNIQUE 17](#_Toc443241575)

[2.2.4.4 Verificar restrición DEFAULT 18](#_Toc443241576)

[2.2.4.5 Verificar restrición CHECK 18](#_Toc443241577)

[2.2.4.6 Verificar restrición implícitas nos tipos ENUM, e SET 19](#_Toc443241578)

[2.2.4.7 Verificar restricións asociadas ao valor NULL 20](#_Toc443241579)

[2.2.4.8 Verificar restricións asociadas ao tipo de dato e tamaño da columna 20](#_Toc443241580)

[2.2.5 Resumo de mensaxes de erro asociados á carga de datos en MySQL. 21](#_Toc443241581)

[2.3 Tarefas 22](#_Toc443241582)

[2.3.1 Tarefa 1. Establecer o modo estrito para a verificación de datos no servidor 22](#_Toc443241583)

[Solución 22](#_Toc443241584)

[2.3.2 Tarefa 2. Verificar restrición PRIMARY KEY 22](#_Toc443241585)

[Solución 23](#_Toc443241586)

[2.3.3 Tarefa 3. Verificar restrición de integridade referencial 24](#_Toc443241587)

[Solución 25](#_Toc443241588)

[2.3.4 Tarefa 4. Verificar restrición UNIQUE 26](#_Toc443241589)

[Solución 26](#_Toc443241590)

[2.3.5 Tarefa 5. Verificar restrición DEFAULT 26](#_Toc443241591)

[Solución 26](#_Toc443241592)

[2.3.6 Tarefa 6. Verificar restricións implícitas nos tipos ENUM, e SET 27](#_Toc443241593)

[Solución 27](#_Toc443241594)

[2.3.7 Tarefa 7. Tarefa integradora sobre verificación de restricións 27](#_Toc443241595)

[Solución 27](#_Toc443241596)

[3. Materiais 31](#_Toc443241597)

[3.1 Documentos de apoio ou referencia 31](#_Toc443241598)

[3.2 Recursos didácticos 31](#_Toc443241599)

[3.3 Material auxiliar 31](#_Toc443241600)

[4. Avaliación 32](#_Toc443241601)

[Criterios de avaliación 32](#_Toc443241602)

[Modelo de proba combinada para TO.6 32](#_Toc443241603)

[Proba para CA4.7 sobre o axuste da implementación ao modelo mediante un conxunto de datos de proba 33](#_Toc443241604)

[Exemplo de solución para entregar en papel 33](#_Toc443241605)

[Exemplo de lista de valoración para TO.6 37](#_Toc443241606)

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 |
| A03 | Verificación e probas con datos. | Verificarase que se cumpren as regras de definición e integridade da base de datos utilizando xogos de datos de proba. | 3 |

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 |
| --- | --- | --- | --- | --- | --- |
| O3.1 | Deseñar datos de proba para verificar o cumprimento das regras de definición e integridade. | A03 | Verificación e probas de datos. | * Verificarase que se cumpren as regras de definición e integridade da base de datos utilizando xogos de datos de proba. | 3 |
| O3.2 | Adaptar scripts para cargar os datos de proba. |
| O3.3 | Realizar a proba e interpretar os resultados. |

Criterios de avaliación

|  |
| --- |
| Criterios de avaliación |
| * CA4.7 - Verificouse o axuste da implementación ao modelo mediante un conxunto de datos de proba. |

Contidos

| Contidos |
| --- |
| Verificación e probas. |

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 |
| A03. Verificación e probas con datos   * Verificarase que se cumpren as regras de definición e integridade da base de datos utilizando xogos de datos de proba | * **Tp3.1** Explicación de conceptos relativos as restrición de integridade, verificación e probas de datos. | * **Ta3.1** Toma de notas e formulación de dúbidas. |  | * Ordenadores con conexión a Internet, que terán instalado o sistema xestor de bases de datos MySQL e o cliente MySQL Workbench. * Material didáctico subministrado polo profesorado en papel e/ou formato dixital. * Proxector. |  | 2 |
| * **Tp3.2** Demostración de: deseño de xogo de datos para a proba; carga de datos nas bases de datos utilizando scritps de sentenzas SQL e ferramentas gráficas; análise das mensaxes de erro que se deben ao incumprimento das restricións de definición e integridade. | * **Ta3.2** Deseño de xogo de datos para a proba. * **Ta3.3** Carga de datos de proba executando scripts de sentenzas SQL * **Ta3.4** Interpretación das mensaxes cos resultados da proba en especial as relacionadas co cumprimento das restricións de definición e integridade das bases de datos. | * Script de carga dos datos de proba. * Imaxes das mensaxes producidas pola carga dos datos de proba. * Base de datos cos datos de proba cargados verificando que cumpren as restricións de definición e integridade. |
|  | * **Ta3.5** Tarefa de avaliación utilizando o instrumento TO.8. | * Exame en papel e/ou en formato dixital. | * Máquina virtual especifica para exame que terá instalado o sistema xestor de bases de datos MySQL e o cliente MySQL Workbench. | * **TO.8** Documento de rexistro de proba documentado que inclúa definición dun conxunto de datos de proba, a carga de datos e a interpretación dos resultados.Escala de valores (observación indirecta). | 1 |

1. A03. Verificación e probas de datos
   1. Introdución
      1. Obxectivos

O obxectivo desta actividade é utilizar xogos de datos de proba para verificar que se cumpren as regras de definición e integridade da base de datos.

* + 1. Software

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) pódese obter información detallada sobre a ferramenta MySQL Workbench e descargar o software.
* En http[://dev.mysql.com/doc/index-gui.html](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.
  + 1. Bases de datos de traballo

As bases de datos *eleccionModulos* e *gabinete* utilizaranse para os exemplos e tarefas desta actividade. Antes de empezar a probar os exemplos ou realizar as tarefas, hai que executar os scripts de creación no servidor e poñer en uso a base de datos correspondente. Os scripts atópanse no cartafol anexo a esta actividade descrito no apartado '3.3 Material auxiliar'.

* + - 1. Base de datos eleccionModulos

A base de datos *eleccionModulos* serve para levar un control dos grupos, módulos, profesores e clases que se imparten nun centro de ensino no que se imparten Ciclos Formativos. Descrición das táboas e das relacións que hai entre elas:

u4a2_tarefa4_1_grafo.emf

* 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 |

* + - 1. Base de datos gabinete

A base de datos *gabinete* serve para levar un control dos clientes e dos asuntos que se levan nun gabinete de avogados. Descrición das táboas e das relacións que hai entre elas:

u4a3_gabinete.emf

* Táboa *cliente*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome | Tipo | Null | Clave | Observacións |
| dni | char(9) | Non | Primary |  |
| nome | varchar(100) | Non | Indice |  |
| enderezo | varchar(150) | Non |  |  |

* Táboa *telefono*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome | Tipo | Null | Clave | Observacións |
| dni | char(9) | Non | Primary | DNI do cliente |
| telefono | char(12) | Non | Teléfono do cliente |

* Táboa *asunto*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome | Tipo | Null | Clave | Observacións |
| expediente | integer | Non | Primary | Clave para identificar o grupo |
| dataInicio | date | Non |  | Descrición do nome do grupo |
| descricion | text | Non |  | Nº de horas de redución para o profesor titor |
| dataArquivo | date |  |  | Identificador do profesor que é titor do grupo |
| estado | enum | Non |  | Valores permitidos:  `recopilación’,’trámite’,xuizo’,’arquivado’ |
| dniCliente | char(9) | Non |  | DNI do cliente que contrata o asunto |

* Táboa *avogado*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome | Tipo | Null | Clave | Observacións |
| dni | char(9) | Non | Primary |  |
| nome | varchar(100) | Non |  |  |
| enderezo | varchar(150) | Non |  |  |

* Táboa *avogado\_asunto* (clave primaria composta)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome | Tipo | Null | Clave | Observacións |
| dniAbogado | char(9) | Non | Primary | Clave para identificar o avogado |
| expediente | integer | Non | Clave para identificar o asunto |

* 1. Actividade
     1. Introdución

Unha vez rematado o proceso de análise, deseño e implementación da base de datos, a seguinte fase no ciclo de vida dunha base de datos é a fase de probas. Nesta fase fanse probas do rendemento, de seguridade e de integridade dos datos, normalmente, en combinación coas aplicacións que se desenvolven para manexar esa base de datos.

Os pasos a seguir na fase de probas son:

* Comprobación da integridade de datos mediante as restricións.
* Comprobación da seguridade para garantir que os usuarios dispoñen de acceso e só poden cambiar os datos que o administrador lles permite.
* Comprobación do rendemento. O rendemento próbase baixo diferentes condicións de carga para ver como procesa a base de datos varias conexións simultáneas e altos volumes de procesos de actualización e lectura. ¿A resposta ás peticións de datos e o suficientemente rápida?. Por exemplo, unha aplicación deseñada con táboas MyISAM pode resultar demasiado lenta no caso de que se fagan moitas actualizacións de datos de maneira simultánea, debido a que o motor de almacenamento MyISAM fai bloqueos a nivel de táboa cando se fan actualizacións nos datos; pode resultar necesario modificar o motor de almacenamento e utilizar InnoDB.
* Axuste dos parámetros a modificar no deseño lóxico e físico en resposta ao resultado das probas. Como resultado das probas, pode que resulte necesario modificar o deseño lóxico ou o deseño físico da base de datos. Quizais se necesiten novos índices (que o responsable das probas pode descubrir tras usar meticulosamente a instrución EXPLAIN de MySQL, que se verá nunha unidade posterior) ou desnormalizar determinadas táboas por razóns de rendemento.

O proceso de verificación e proba é de carácter interactivo, é dicir, durante o proceso lévanse a cabo varias probas e impleméntanse os cambios requiridos para mellorar o rendemento e a seguridade.

Nesta actividade estudaranse as probas de comprobación da integridade dos datos; o resto de pasos vense noutras unidades didácticas.

* + 1. Modo do servidor MySQL

O comportamento do servidor MySQL cando se producen erros asociados aos datos dependerá de se está traballando en modo SQL estrito ou non. No caso de non estar activado o modo estrito, os datos que non se axustan á definición da columna son engadidos á táboa pero móstrase unha mensaxe de alerta (*warning*), e no caso de estar activado o modo estrito, xa non se permite que os datos sexan engadidos e móstrase unha mensaxe de erro.

*MySQL 5.6. Reference Manual, apartado 5.1.7 Server SQL Modes*:

'MySQL server puede operar en distintos modos SQL, y puede aplicar estos modos de forma distinta a diferentes clientes. Esto permite que cada aplicación ajuste el modo de operación del servidor a sus propios requerimientos.

Los modos definen qué sintaxis SQL debe soportar MySQL y que clase de chequeos de validación de datos debe realizar. Esto hace más fácil de usar MySQL en distintos entornos y usar MySQL junto con otros servidores de bases de datos.'

Pódese consultar o contido da variable *sql\_mode* coa sentenza:

**select** @@sql\_mode**;**



O modo que se mostra, NO\_ENGINE\_SUBSTITUTION, evita a substitución automática do motor de almacenamento cando o motor desexado non está dispoñible ou compilado.

Cando o servidor está en funcionamento, pódese cambiar o modo SQL utilizando a sentenza SET e cambiando o valor da variable do sistema *sql\_mode*. Calquera cliente pode cambiar o valor de *sql\_mode* en calquera momento. Sintaxe:

SET [GLOBAL|SESSION] sql\_mode = 'modos';

* Cando se asigna a variable a nivel GLOBAL, afecta a todos os clientes que se conecten a partir dese momento. Para executala hai que ter o privilexio SUPER de administrador.
* Cando se asigna a variable a nivel SESSION afecta só ao cliente actual. Se non se indica explicitamente o nivel, por defecto, é SESSION.
* O contido de 'modos' é unha lista dos diferentes modos que se lle queren asignar á variable, separados por comas[[1]](#footnote-3). No que afecta á inserción e actualización de filas nas táboas, os modos máis importantes aos que fai referencia o manual son os seguintes:
* STRICT\_TRANS\_TABLES: se un valor non se pode inserir nunha columna dunha táboa transacional, abórtase a execución do comando e provoca unha mensaxe de erro.
* STRICT\_ALL\_TABLES: se un valor non se pode inserir nunha columna dunha táboa que utilice calquera motor de almacenamento, abórtase a execución do comando e provoca unha mensaxe de erro.

Cando o manual fai referencia ao Modo SQL estrito, significa que está habilitado un ou ambos dos anteriores STRIC\_TRANS\_TABLES ou STRIC\_ALL\_TABLES.

Exemplo: modificar o contido da variable *sql\_mode*, engadíndolle o modo PIPES\_AS\_CONCAT ao contido que tiña, para tratar os símbolos || como un operador de concatenación de cadeas (como a función concat()), en lugar de como sinónimo do operador lóxico OR.

**set** **session** sql\_mode **=** 'PIPES\_AS\_CONCAT,NO\_ENGINE\_SUBSTITUTION'**;**

**select** @@**session.**sql\_mode**,** @@**global.**sql\_mode**;**

sql_mode.PNG

* Tarefa 1. Establecer o modo *estrito* para a verificación de datos no servidor.
  + 1. Edición de datos con MySQL Workbench

A carga de datos de proba pódese facer executando un script coas sentenzas SQL adecuadas ou utilizando un cliente gráfico como MySQL Workbench. No primeiro caso, é posible gardar o script e volver a executalo as veces que faga falta pero o estudo das sentenzas SQL necesarias para a carga de datos explicarase en unidades didácticas posteriores.

MySQL Workbench proporciona un contorno gráfico de edición dos datos contidos nas táboas, que permite aos usuarios menos expertos inserir, modificar, borrar e consultar filas sen necesidade de escribir sentenzas SQL. Ao pasar o cursor sobre o nome dunha táboa móstranse tres iconas. A da dereita (Manipulación de datos) permite mostrar os datos na zona de edición e facer cambios neles.



Para inserir novas filas, hai que escribir os datos na última fila, en substitución dos valores NULL e a continuación seleccionar o botón 'Apply' para aplicar os cambios.

edicion_datos.emf

A modificación dos datos dunha fila pódese facer escribindo os novos datos sobre os datos actuais e a continuación seleccionar o botón 'Apply'.

Para borrar filas, hai que seleccionar as filas que se queren borrar, pulsar o botón dereito do rato e seleccionar a opción *Delete Row(s*).

Tamén se pode escoller o formato 'Form Editor' para ter unha vista da fila en forma de formulario.

formulario_datos.emf

* + 1. Integridade e restricións

O concepto de integridade ten unha gran importancia no deseño e implementación de bases de datos. Para manter a integridade dos datos, é necesario establecer un conxunto de regras ou restricións que garantan que a información almacenada na bases de datos é correcta. Unha vez rematado o proceso de implementación, hai que establecer un sistema de probas de datos, que permita verificar que a base de datos cumpre as regras e restricións. Hai algunhas restricións de usuario que se implementan facendo a validación coa axuda de disparadores (*triggers*), que se ven nunha unidade posterior.

No caso das bases de datos relacionais, pódese diferenciar entre as restricións inherentes ao modelo, que son un conxunto de regras que son esenciais para configurar a estrutura dos datos e manter a coherencia das relacións, e as restricións específicas que son propias de cada base de datos.

Entre as restricións inherentes ao modelo relacional, pódese destacar a restrición de integridade de entidade e a restrición de integridade referencial.

Unha vez creado o esquema dunha base de datos, pódese consultar información sobre as restricións consultando as táboas da base de datos INFORMATION\_SCHEMA: TABLE\_CONSTRAINTS e REFERENTIAL\_CONSTRAINTS.

Na fase de probas hai que comprobar os seguintes tipos de restricións de integridade:

* Restricións PRIMARY KEY.
* Restricións de integridade referencial.
* Restricións UNIQUE.
* Restricións DEFAULT.
* Restricións CHECK.
* Restricións implícitas nos tipos ENUM, e SET.
* Restricións de valor NULL.
* Restricións asociadas ao tipo de dato e tamaño da columna.

A fase de probas empeza, normalmente, coa carga dun conxunto de datos de proba que ten como propósito verificar que se cumpren as restricións de integridade da base de datos.

Para facer as probas que veñen a continuación debe estar habilitado o modo SQL estrito (STRIC\_ALL\_TABLES ou STRIC\_TRANS\_TABLES).

-- habilitando o modo SQL estrito

**set** **session** sql\_mode **=** 'STRICT\_ALL\_TABLES,NO\_ENGINE\_SUBSTITUTION'**;**

* + - 1. Verificar restrición PRIMARY KEY

Hai que comprobar que:

* Cada fila ten un valor distinto na columna definida como clave primaria. Non se permiten valores repetidos.
* Non se permiten valores nulos nesa columna.

Exemplo: para comprobar que a táboa *modulo* da base de datos *eleccionmodulos* cumpre as restricións de clave primaria, pódese executar o seguinte *script* de sentenzas SQL que empeza inserindo filas con claves distintas de null e sen repetir claves e termina inserindo unha fila con clave repetida e outra con clave null.

-- habilitando o modo SQL estrito

**set** **session** sql\_mode **=** 'STRICT\_ALL\_TABLES,NO\_ENGINE\_SUBSTITUTION'**;**

-- inserindo filas na táboa modulo

**use** eleccionmodulos**;**

**insert into** modulo **(**idModulo**,** horasSemanais**,** descricion**,** corpo**)** **values**

**(**'MP0484'**,** 7**,** 'Bases de datos'**,** 'S'**);**

**insert** **into** modulo **(**idModulo**,** horasSemanais**,** descricion**,** corpo**)** **values**

**(**'MP0485'**,** 9**,** 'Programación'**,** 'S'**);**

**insert** **into** modulo **(**idModulo**,** horasSemanais**,** descricion**,** corpo**)** **values**

**(**'MP0373'**,** 5**,** 'Linguaxes de marcas e sistemas de xestión de información'**,** 'S'**);**

**insert** **into** modulo **(**idModulo**,** horasSemanais**,** descricion**,** corpo**)** **values**

**(**'MP0487'**,** 4**,** 'Contornos de desenvolvemento'**,** 'S'**);**

**insert** **into** modulo **(**idModulo**,** horasSemanais**,** descricion**,** corpo**)** **values**

**(**'MP0615'**,** 5**,** 'Deseño de interfaces web'**,** 'T'**);**

**insert** **into** modulo **(**idModulo**,** horasSemanais**,** descricion**,** corpo**)** **values**

**(**'MP0615'**,** 5**,** 'Desenvolvemento web en contorno servidor'**,** 'T'**);**

**insert** **into** modulo **(**idModulo**,** horasSemanais**,** descricion**,** corpo**)** **values**

**(null,** 5**,** 'Desenvolvemento web en contorno servidor'**,** 'T'**);**

As primeiras 7 liñas execútanse sen problemas e para cada inserción móstrase unha mensaxe informando das filas inseridas co seguinte formato:



Cando se executa a liña número 8 móstrase unha mensaxe de erro porque xa existe un módulo que ten o mesmo valor na columna *idModulo*, definida como clave primaria.

-- a liña 8 contén o código:

**insert** **into** modulo **(**idModulo**,** horas\_semanais**,** descricion**,** corpo**)** **values**

**(**'MP0615'**,** 5**,** 'Desenvolvemento web en contorno servidor'**,** 'T'**);**



Cando se executa a liña número 9 móstrase unha mensaxe de erro informando que a columna *idModulo* non pode tomar o valor NULL por ser clave primaria.

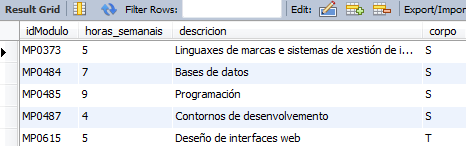
-- a liña 9 contén o código:

**insert** **into** modulo **(**idModulo**,** horas\_semanais**,** descricion**,** corpo**)** **values**

**(null,** 5**,** 'Desenvolvemento web en contorno servidor'**,** 'T'**);**



As filas correspondentes ás liñas números 8 e 9 non se gravan na táboa porque non verifican as restricións de integridade de clave primaria. O resto das filas graváronse na táboa e na seguinte imaxe móstranse os datos contidos na táboa.



Con isto queda probado que o modelo cumpre as restricións de clave primaria para a táboa *modulo*. Habería que facer o mesmo co resto de táboas.

* Tarefa 2. Verificar restrición PRIMARY KEY.
  + - 1. Verificar restrición de integridade referencial

Hai que comprobar que:

* Cando se insiren ou modifican datos, os valores da clave foránea deben coincidir cos valores da clave primaria á que fai referencia, ou tomar o valor nulo.
* Cando se modifican ou borran filas de datos, cúmprense as restricións de integridade referencial asociadas as opcións RESTRICT, NO ACTION, CASCASE e SET NULL.

Exemplos:

* Cando se creou o esquema da táboa *grupo* da base de datos *eleccionmodulos* definiuse unha clave foránea para a columna *idProfesor*, que fai referencia á clave primaria da táboa *profesor* (*idProfesor*). Mostra do código utilizado para definir a clave foránea:

/\* 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;**

* Para comprobar a restrición de integridade referencial na clave foránea *idProfesor* da táboa *grupo* da base de datos *eleccionmodulos,* pódese executar o seguinte *script* de sentenzas SQL*.*

NOTA: Antes de executar este script ten que estar feita a Tarefa 2.1, porque necesítanse datos que se insiren nesa tarefa.

-- habilitando o modo SQL estrito

**set** **session** sql\_mode **=** 'STRICT\_ALL\_TABLES,NO\_ENGINE\_SUBSTITUTION'**;**

-- inserindo filas na táboa grupo

**use** eleccionmodulos**;**

**insert** **into** grupo **(**idGrupo**,** descricion**,** horasTitoria**,** idProfesor**)** **values**

**(**'ADSIINRE-1A'**,**'1º Ordinario Administración de sistemas informáticos en rede A'**,**2**,** 'pepe'**);**

**insert** **into** grupo **(**idGrupo**,** descricion**,** horasTitoria**,** idProfesor**)** **values**

**(**'(A)ADSIINRE-A'**,**'Adultos - Administración de sistemas informáticos en rede A'**,**2**,** **null);**

**insert** **into** grupo **(**idGrupo**,** descricion**,** horasTitoria**,** idProfesor**)** **values**

**(**'(A)DAW-A'**,**'Adultos - Desenvolvemento de aplicacións web A'**,**2**,** 'bernardo'**);**

As dúas primeiras sentenzas INSERT execútanse sen problemas. A primeira toma o valor 'pepe' na columna *idProfesor*, que é un valor que existe na clave primaria da táboa *profesor*. A segunda toma o valor NULL na columna *idProfesor*, que admite o valor nulo. Mensaxes:



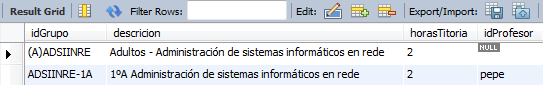
A terceira sentenza intenta inserir unha fila que toma o valor 'bernardo' na columna *idProfesor*, que é un valor que non existe na clave primaria de ningunha fila da táboa *profesor*. Isto da lugar a un erro, e o sistema mostra a seguinte mensaxe informando que non se pode inserir a fila e non se producen cambios no contido da táboa.



Texto completo da mensaxe:

|  |
| --- |
| Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`eleccionmodulos`.`grupo`, CONSTRAINT `fk\_grupo\_profesor` FOREIGN KEY (`idProfesor`) REFERENCES `profesor` (`idProfesor`) ON UPDATE CASCADE) |

Os datos que contén a táboa *grupo* móstranse na seguinte imaxe:



Con isto queda probado que o modelo cumpre as restricións de clave foránea para a inserción e modificación de filas na táboa *grupo*. Habería que facer o mesmo co resto de táboas.

* Para comprobar a integridade referencial cando se borran filas na táboa *profesor* da base de datos *eleccionmodulos* á que fai referencia a clave foránea *idProfesor* da táboa *grupo*, pódese executar o seguinte *script* de sentenzas SQL*.*

-- habilitando o modo SQL estrito

**set** **session** sql\_mode **=** 'STRICT\_ALL\_TABLES,NO\_ENGINE\_SUBSTITUTION'**;**

-- borrar o profesor con clave primaria 'pepe'

**use** eleccionmodulos**;**

**delete** **from** profesor

**where** idProfesor**=**'pepe'**;**

Cando se executa a sentenza para borrar (*delete*) a fila correspondente ao profesor que ten o valor 'pepe' na columna *idProfesor*, definida como clave primaria, o sistema detecta un erro nas restricións de clave foránea e mostra a seguinte mensaxe:



A mensaxe informa que non se pode borrar ou modificar unha fila 'pai' asociada a unha clave foránea. Isto significa que non se pode borrar a fila do profesor 'pepe' porque hai unha fila na táboa *grupo* que ten un valor na clave foránea que fai referencia a esa fila, e cando se creou a clave foránea engadiuse a opción *'on delete restrict'*. Non hai cambios no contido da táboa.

* Para comprobar a integridade referencial cando se modifican filas na táboa *profesor* da base de datos *eleccionmodulos* á que fai referencia a clave foránea *idProfesor* da táboa *grupo*, pódese executar o seguinte *script* de sentenzas SQL*.*

-- habilitando o modo SQL estrito

**set** **session** sql\_mode **=** 'STRICT\_ALL\_TABLES,NO\_ENGINE\_SUBSTITUTION'**;**

-- modificar o valor da clave primaria para o profesor con clave primaria 'pepe'

**use** eleccionmodulos**;**

**update** profesor

**set** idProfesor **=** 'jose'

**where** idProfesor**=**'pepe'**;**



Cando se executa a sentenza para modificar (*update*) o valor da clave primaria do profesor 'pepe', móstrase unha mensaxe informando que a fila foi modificada con éxito.

Se temos en conta que cando se creou a clave foránea engadiuse a opción *'on update cascade',* o cambio de valor da clave primaria do profesor 'pepe' por 'jose' provoca un cambio 'en cascada' dos valores da columna definida como clave foránea nas filas que fan referencia a ese profesor.

cascada.emf

Con estes dous últimos exemplos queda probado que o modelo cumpre as restricións de integridade referencial para o borrado e modificación de filas na táboa á que fai referencia a clave foránea da táboa *grupo*, que é a táboa *profesor*. Habería que facer o mesmo co resto de táboas.

* Tarefa 3. Verificar restrición FOREIGN KEY e integridade referencial.
  + - 1. Verificar restrición UNIQUE

Hai que comprobar que:

* Cada fila ten un valor distinto na columna que ten asociado un índice de tipo UNIQUE. Non se permiten valores repetidos.

Exemplo: para verificar a restrición de clave única na columna *nrp* da táboa *profesor* da base de datos *eleccionmodulos,* cando se insiren filas, pódese executar o seguinte *script* de sentenzas SQL.

-- habilitando o modo SQL estrito

**set** **session** sql\_mode **=** 'STRICT\_ALL\_TABLES,NO\_ENGINE\_SUBSTITUTION'**;**

-- inserción dunha fila cun valor duplicado nunha clave única

**use** eleccionmodulos**;**

**insert** **into** profesor **(**idProfesor**,** corpo**,** nrp**,** nome**,** apelidos**,** horasLectivas**)** **values**

**(**'andres'**,** 'S'**,** '152365842.37'**,** 'Andrés'**,** 'Villar Silva'**,**17**);**

Cando se executa a sentenza para inserir un novo profesor, móstrase unha mensaxe de erro informando que xa existe unha fila que contén o mesmo valor na columna *nrp* que é unha clave única. Non hai cambios no contido da táboa.



Con este exemplo queda probado que o modelo cumpre a restricións de clave única para a columna *nrp* da táboa *profesor*. Habería que facer o mesmo co resto de columnas definidas como clave única, no caso de que as houbera.

* Tarefa 4. Verificar restrición UNIQUE.
  + - 1. Verificar restrición DEFAULT

Hai que comprobar que:

* Cando se insiren datos nunha fila e non se asigna un valor ás columnas que teñen o atributo DEFAULT, estas columnas tomarán o valor especificado por defecto.

Exemplo: para comprobar a restrición DEFAULT para a columna *horasLectivas* da táboa *profesor* da base de datos *eleccionmodulos,* ao inserir filas, pódese executar o seguinte *script* de sentenzas SQL.

-- habilitando o modo SQL estrito

**set** **session** sql\_mode **=** 'STRICT\_ALL\_TABLES,NO\_ENGINE\_SUBSTITUTION'**;**

-- columna con valor por defecto: horasLectivas

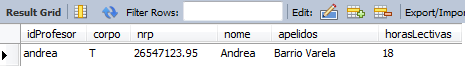
**use** eleccionmodulos**;**

**insert** **into** profesor **(**idProfesor**,** corpo**,** nrp**,** nome**,** apelidos**)** **values**

**(**'andrea'**,** 'T'**,** '26547123.95'**,** 'Andrea'**,** 'Barrio Varela'**);**



A mensaxe que se mostra cando se executa a sentenza de inserción informa que foi inserida unha fila con éxito. Como non se deu ningún valor para a columna *horasLectivas* e esta ten definido como valor por defecto o valor 18 (*default 18*), ese será o valor que terá a columna para a fila inserida.



Con este exemplo queda probado que o modelo cumpre a restrición DEFAULT para a columna *horasLectivas* da táboa *profesor*. Habería que facer o mesmo co resto de columnas definidas con valores por defecto, no caso de que as houbera.

* Tarefa 5. Verificar restrición DEFAULT.
  + - 1. Verificar restrición CHECK

Hai que comprobar que:

* Cando se insiren ou modifican datos nunha columna que ten o atributo CHECK, se verifiquen as condicións establecidas para esa columna.

MySQL non implementa esta restrición. Permite escribir a cláusula CHECK na creación da estrutura da táboa sen dar erro de sintaxe, pero non fai as comprobacións de valor.

* + - 1. Verificar restrición implícitas nos tipos ENUM, e SET

Estes tipos de datos son propios de MySQL e permiten asociar unha restrición de valor a unha columna poñendo entre parénteses o conxunto de valores válidos para a columna.

Hai que comprobar que:

* Cando se insiren ou modifican datos nunha columna do tipo ENUM ou SET, os valores que toma a columna estean na lista de valores permitidos.

Exemplo: para comprobar a restrición ENUM da columna *corpo* da táboa *modulo* da base de datos *eleccionmodulos,* ao inserir filas, pódese executar o seguinte *script* de sentenzas SQL.

-- habilitando o modo SQL estrito

**set** **session** sql\_mode **=** 'STRICT\_ALL\_TABLES,NO\_ENGINE\_SUBSTITUTION'**;**

-- probas para columna e tipo ENUM. Valor non válido

**use** eleccionmodulos**;**

**insert** **into** modulo **(**idModulo**,** horasSemanais**,** descricion**,** corpo**)** **values**

**(**'MP0613'**,** 7**,** 'Desenvolvemento web en contorno servidor'**,** 'P'**);**

-- probas para columna e tipo ENUM. Valores válidos

**insert** **into** modulo **(**idModulo**,** horasSemanais**,** descricion**,** corpo**)** **values**

**(**'MP0613'**,** 7**,** 'Desenvolvemento web en contorno servidor'**,** 'S'**);**

**insert** **into** modulo **(**idModulo**,** horasSemanais**,** descricion**,** corpo**)** **values**

**(**'MP0612'**,** 6**,** 'Desenvolvemento web en contorno cliente'**,** **null);**

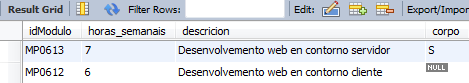
Cando se executa a primeira sentenza de inserción, móstrase unha mensaxe de erro que informa que o valor asignado á columna *corpo* non é válido. Isto é debido a que o tipo de dato da columna é *enum('S','T')* que significa que os únicos valores válidos para a columna son as letras 'S' ou 'T', ou ben, o valor nulo. Non hai cambios no contido da táboa.



As dúas sentenzas seguintes execútanse correctamente porque toman os valores 'S' e NULL que son valores permitidos.



Os datos inseridos na táboa *modulo* despois de executar o *script* son os seguintes:



Con este exemplo queda probado que o modelo cumpre as restricións de tipo ENUM para a columna *corpo* da táboa *modulo*. Habería que facer o mesmo co resto de columnas definidas de tipo ENUM, no caso de que as houbera.

* Tarefa 6. Verificar restrición implícitas nos tipos ENUM, e SET.
  + - 1. Verificar restricións asociadas ao valor NULL

As restricións de valor NULL serven para obrigar a cargar un dato nunha columna que se define coa propiedade NOT NULL.

Hai que comprobar que:

* Non se asigna o valor nulo a unha columna definida coa propiedade NOT NULL.

Exemplo: para comprobar a restrición NULL na columna *descricion* da táboa *modulo* da base de datos *eleccionmodulos,* pódese executar o seguinte *script* de sentenzas SQL para inserir filas.

-- habilitando o modo SQL estrito

**set** **session** sql\_mode **=** 'STRICT\_ALL\_TABLES,NO\_ENGINE\_SUBSTITUTION'**;**

-- probas para columna coa propiedade NOT NULL

**use** eleccionmodulos**;**

**insert** **into** modulo **(**idModulo**,** horasSemanais**,** descricion**,** corpo**)**

**values (**'MP0483'**,** 6**,** **null,** **null);**

Cando se executa a sentenza de inserción, móstrase unha mensaxe de erro porque a columna *descricion* está definida coa propiedade NOT NULL e inténtase asignarlle o valor nulo



* + - 1. Verificar restricións asociadas ao tipo de dato e tamaño da columna

En xeral, para todas as columnas, débese comprobar mediante a carga de datos reais, que o tipo de dato e o tamaño asignados son os adecuados.

Exemplo: para verificar as restricións de tipo de dato e tamaño de columna para a táboa *grupo* da base de datos *eleccionmodulos,* pódese executar o seguinte *script* de sentenzas SQL para inserir filas.

-- habilitando o modo SQL estrito

**set** **session** sql\_mode **=** 'STRICT\_ALL\_TABLES,NO\_ENGINE\_SUBSTITUTION'**;**

-- inserindo filas na táboa grupo

**use** eleccionmodulos**;**

**insert** **into** grupo **(**idGrupo**,** descricion**,** horasTitoria**,** idProfesor**)** **values**

**(**'DAM-1A'**,**'1ºA - Desenvolvemento de aplicacións multiplataforma'**,**'muchas'**,** 'angel'**);**

**insert** **into** profesor **(**idProfesor**,** corpo**,** nrp**,** nome**,** apelidos**)** **values**

**(**'federico'**,** 'S'**,** '22369741.27'**,** 'Federico Fermín de los Santos Inocentes'**,** 'Valencia Bargas'**);**



A primeira sentenza de inserción produce un erro debido a que se trata de asignar o valor 'muchas' á columna *horasTitoria* definida de tipo numérico enteiro (*smallint*). No caso de asignar valores de tipo numérico a unha columna definida de tipo alfanumérica, MySQL fai a conversión de tipos de dato automaticamente e non se produce erro.

O erro que produce a seguinte sentenza informa que o valor asignado á columna *nome* da táboa *profesor* é demasiado longo. No caso de necesitar almacenar os nomes completos habería que cambiar a definición da columna, dándolle un tamaño maior. Exemplo de código para cambiar o tamaño da columna:

**alter** **table** profesor

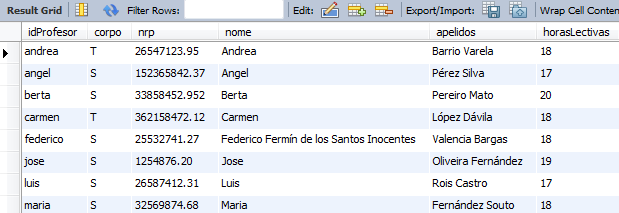
**modify** nome **varchar(**50**)** **not** **null;**



Unha vez feitos os cambios no esquema da táboa, pódese executar outra vez a última sentenza de inserción. A mensaxe que se mostra agora é de que a inserción se fixo correctamente.



Datos inseridos na táboa *grupo* despois de facer os cambios na estrutura da táboa e de executar o *script*:



Con este exemplo trátase de mostrar que hai veces que como resultado da fase de probas é necesario facer cambios na estrutura das táboas.

* + 1. Resumo de mensaxes de erro asociados á carga de datos en MySQL.

Na seguinte relación móstrase un resumo das mensaxes de erro que se produciron ao executar as tarefas de verificación de integridade propostas.

* Clave primaria. Son os erros que se producen cando se insire unha fila na que a clave primaria toma un valor duplicado, ou cando toma un valor NULL.

|  |
| --- |
| a) Error Code: 1062. Entrada duplicada '15659845' para la clave 'PRIMARY'  b) Error Code: 1048. La columna 'dni' no puede ser nula |

* Integridade referencial (inserción ou modificación de filas na táboa que ten a clave foránea). Son os erros que se producen cando nunha operación de inserción ou modificación o valor da clave foránea toma un valor diferente de NULL e dos valores que toma a clave primaria á que fai referencia.

|  |
| --- |
| Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`gabinete`.`asunto`, CONSTRAINT `fk\_asunto\_cliente` FOREIGN KEY (`dniCliente`) REFERENCES `cliente` (`dni`) ON UPDATE CASCADE) |

* Integridade referencial (modificación e borrado de filas na táboa á que fai referencia a clave foránea coa opción REFERENCES). Son os erros que se producen cando se intenta modificar ou borrar unha fila na táboa á que fai referencia unha clave foránea, cando ten asociada acción RESTRICT ou NO ACTION (que son sinónimos en MySQL) a algunha desas operacións.

|  |
| --- |
| Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`gabinete`.`asunto`, CONSTRAINT `asunto\_ibfk\_1` FOREIGN KEY (`dniCliente`) REFERENCES `cliente` (`dni`) ON UPDATE CASCADE) |

* Tipos de dato ENUM e SET. Son os erros que se producen cando o valor asignado toma un valor diferente dos relacionados na definición da columna.

|  |
| --- |
| Error Code: 1265. Datos truncados para columna 'estado' en la línea 1 |

* Tarefa 7. Tarefa integradora sobre verificación de restricións.
  1. Tarefas

As tarefas propostas son as seguintes:

* Tarefa 1. Establecer o modo estrito para a verificación de datos no servidor.
* Tarefa 2. Verificar restrición PRIMARY KEY.
* Tarefa 3. Verificar restrición de integridade referencial.
* Tarefa 4. Verificar restrición UNIQUE.
* Tarefa 5. Verificar restrición DEFAULT.
* Tarefa 6. Verificar restrición implícitas nos tipos ENUM, e SET.
* Tarefa 7. Tarefa integradora sobre verificación de restricións.
  + 1. Tarefa 1. Establecer o modo estrito para a verificación de datos no servidor

Modificar o contido da variable do sistema *sql\_mode*, para habilitar o modo SQL estrito.

Solución

Compróbase o contido da variable *sql\_mode*:

**select** @@sql\_mode**;**



Modifícase o contido da variable *sql\_mode*, engadíndolle ao contido que tiña, o modo STRICT\_ALL\_TABLES para que afecte a todas as táboas:

**set** **session** sql\_mode **=** 'STRICT\_ALL\_TABLES,NO\_ENGINE\_SUBSTITUTION'**;**

* + 1. Tarefa 2. Verificar restrición PRIMARY KEY
* Tarefa 2.1. Executar o seguinte *script* de sentenzas SQL para inserir filas na táboa *profesor* na base de datos *eleccionmodulos.* Comprobar que se cumpren as restricións de clave primaria.

-- habilitando o modo SQL estrito

**set** **session** sql\_mode **=** 'STRICT\_ALL\_TABLES,NO\_ENGINE\_SUBSTITUTION'**;**

-- inserindo filas na táboa profesor

**use** eleccionmodulos**;**

**insert** **into** profesor **(**idProfesor**,** corpo**,** nrp**,** nome**,** apelidos**,** horasLectivas**)** **values**

**(**'berta'**,** 'S'**,** '33858452.952'**,** 'Berta'**,** 'Pereiro Mato'**,**20**);**

**insert** **into** profesor **(**idProfesor**,** corpo**,** nrp**,** nome**,** apelidos**,** horasLectivas**)** **values**

**(**'pepe'**,** 'S'**,** '1254876.20'**,** 'Jose'**,** 'Oliveira Fernández'**,**19**);**

**insert** **into** profesor **(**idProfesor**,** corpo**,** nrp**,** nome**,** apelidos**,** horasLectivas**)** **values**

**(**'maria'**,** 'S'**,** '32569874.68'**,** 'Maria'**,** 'Fernández Souto'**,**18**);**

**insert** **into** profesor **(**idProfesor**,** corpo**,** nrp**,** nome**,** apelidos**,** horasLectivas**)** **values**

**(**'angel'**,** 'S'**,** '152365842.37'**,** 'Angel'**,** 'Pérez Silva'**,**17**);**

**insert** **into** profesor **(**idProfesor**,** corpo**,** nrp**,** nome**,** apelidos**,** horasLectivas**)** **values**

**(**'luis'**,** 'S'**,** '26587412.31'**,** 'Luis'**,** 'Rois Castro'**,**17**);**

**insert** **into** profesor **(**idProfesor**,** corpo**,** nrp**,** nome**,** apelidos**,** horasLectivas**)** **values**

**(**'carmen'**,** 'T'**,** '362158472.12'**,** 'Carmen'**,** 'López Dávila'**,**18**);**

**insert** **into** profesor **(**idProfesor**,** corpo**,** nrp**,** nome**,** apelidos**,** horasLectivas**)** **values**

**(**'maria'**,** 'S'**,** '33652481.33'**,** 'Maria Luisa'**,** 'Suárez Barrios'**,**18**);**

**insert** **into** profesor **(**idProfesor**,** corpo**,** nrp**,** nome**,** apelidos**,** horasLectivas**)** **values**

**(null,** 'S'**,** '33652481.33'**,** 'Maria Luisa'**,** 'Suárez Barrios'**,**18**);**

* Tarefa 2.2. Na base de datos *gabinete*, inserir as seguintes filas de datos na táboa *cliente*, e comprobar que se cumpren as restricións de clave primaria.

|  |  |  |  |
| --- | --- | --- | --- |
| clientes | dni | nome | enderezo |
|  | 15659845 | Fernández López, Mario | Rua Vella, 7 27002 Lugo |
|  | 32022100 | González Valín, Aurora | Rua Real 14 Mondoñedo 27541 Lugo |
|  | 33458745 | Varela Núñez, Carmen | Avda. Sarria, 34 Monforte 27162 Lugo |
|  | 15659845 | Rios Castro, Manuel | 'Plza. Maior, 2 32001 Ourense |

Para inserir os datos pódese utilizar o contorno gráfico de MySQ Workbench, ou ben executar o seguinte script de sentenzas SQL:

-- habilitando o modo SQL estrito

**set** **session** sql\_mode **=** 'STRICT\_ALL\_TABLES,NO\_ENGINE\_SUBSTITUTION'**;**

-- inserindo filas na táboa cliente

**use** gabinete**;**

**insert** **into** cliente

**values** **(**'15659845'**,**'Fernández López, Mario'**,**'Rua Vella, 7 27002 Lugo'**);**

**insert** **into** cliente

**values** **(**'32022100'**,**'González Valín, Aurora'**,**'Rua Real 14 Mondoñedo 27541 Lugo'**);**

**insert** **into** cliente

**values** **(**'33458745'**,**'Varela Núñez, Carmen'**,**'Avda. Sarria, 34 Monforte 27162 Lugo'**);**

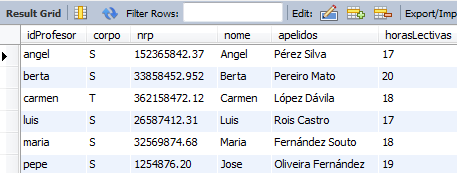
**insert** **into** cliente

**values** **(**'15659845'**,**'Rios Castro, Manuel'**,**'Plza. Maior, 2 32001 Ourense'**);**

Solución

* Tarefa 2.1

A execución das 15 primeiras liñas do *script* non darían problemas e os datos quedarían gravados como se mostra na seguinte imaxe.



Cando se executa a liña número 16, móstrase unha mensaxe de erro porque xa existe un profesor que ten o mesmo valor na columna *idProfesor*, definida como clave primaria. Cando se executa a liña número 18, móstrase unha mensaxe de erro informando que a columna *idProfesor* non pode tomar o valor NULL por ser clave primaria.

Código das liñas 16 e 18, e mensaxes de erro mostrados:

**insert** **into** profesor **(**idProfesor**,** corpo**,** nrp**,** nome**,** apelidos**,** horasLectivas**)** **values**

**(**'maria'**,** 'S'**,** '33652481.33'**,** 'Maria Luisa'**,** 'Suárez Barrios'**,**18**);**

**insert** **into** profesor **(**idProfesor**,** corpo**,** nrp**,** nome**,** apelidos**,** horasLectivas**)** **values**

**(null,** 'S'**,** '33652481.33'**,** 'Maria Luisa'**,** 'Suárez Barrios'**,**18**);**

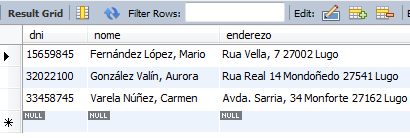


As filas correspondentes ás liñas números 16 e 18 non se gravan na táboa porque non verifican as restricións de integridade de clave primaria. O resto das filas graváronse na táboa correctamente.

Con isto queda probado que o modelo cumpre as restricións de clave primaria para a táboa *profesor*.

* Tarefa 2.2

A execución das tres primeiras instrucións non darían problemas, e os datos quedarían gravados como se mostra:



Ao intentar inserir os datos do cuarto cliente prodúcese un erro debido a que xa existe un cliente co mesmo DNI; a columna *dni* está definida como clave primaria e non pode ter valores duplicados.

A mensaxe de erro é a seguinte:



Con isto queda probado que o modelo cumpre as restricións de clave primaria para a táboa *cliente*.

* + 1. Tarefa 3. Verificar restrición de integridade referencial
* Tarefa 3.1. Executar o seguinte *script* de sentenzas SQL para inserir filas na táboa *telefono* da base de datos *gabinete*, e comprobar que se cumpren as restricións de integridade referencial para a inserción de filas na táboa que contén a clave foránea.

-- habilitando o modo SQL estrito

**set** **session** sql\_mode **=** 'STRICT\_ALL\_TABLES,NO\_ENGINE\_SUBSTITUTION'**;**

-- inserindo filas na táboa telefono

**use** gabinete**;**

**insert** **into** telefono **values**

**(**'15659845'**,**'625332255'**);**

**insert** **into** telefono **values**

**(**'15659845'**,**'698111222'**);**

**insert** **into** telefono **values**

**(**'32568925'**,**'982240103'**);**

* Tarefa 3.2. Executar o seguinte *script* de sentenzas SQL para modificar unha fila na táboa *cliente* da base de datos *gabinete*, e comprobar que se cumpren as restricións de integridade referencial para a modificación de filas na táboa á que fai referencia a clave foránea. (*'on update cascade'*).

-- habilitando o modo SQL estrito

**set** **session** sql\_mode **=** 'STRICT\_ALL\_TABLES,NO\_ENGINE\_SUBSTITUTION'**;**

-- modificación da clave primaria dun cliente

**use** gabinete**;**

**update** cliente

**set** dni **=** '28333845'

**where** dni **=** '15659845'**;**

* Tarefa 3.3. Executar o seguinte *script* de sentenzas SQL para borrar unha fila na táboa *cliente* da base de datos *gabinete*, e comprobar que se cumpren as restricións de integridade referencial para borrado de filas na táboa á que fai referencia a clave foránea. (*'on delete cascade'*).

-- habilitando o modo SQL estrito

**set** **session** sql\_mode **=** 'STRICT\_ALL\_TABLES,NO\_ENGINE\_SUBSTITUTION'**;**

-- borrado da clave primaria dun cliente

**use** gabinete**;**

**delete** **from** cliente

**where** dni **=** '28333845'**;**

Solución

* Tarefa 3.1

As dúas primeiras insercións execútanse sen problemas e engaden na táboa *telefono* dúas filas cos teléfonos para o cliente con dni '15659845', que corresponde a un cliente que existe na táboa *cliente*. A terceira inserción non se pode executar porque non existe ningún cliente co dni ' 32568925' na táboa cliente.



Con isto queda probado que o modelo cumpre as restricións de integridade referencial para a inserción de filas na táboa *telefono*.

* Tarefa 3.2

Cando se executa o *script,* modifícase o valor da clave primaria da táboa cliente para o cliente que ten o dni 15659845. O novo valor para a clave primaria é 28333845. Debido a que a clave foránea ten asociada a restrición *'on update cascade'*, faise unha modificación 'en cascada' de todas as filas da táboa telefono nas que a clave foránea fai referencia a ese cliente, cambiando o valor que teñen polo novo valor que toma a clave primaria.



u4a3tarefa0301.emf

Con isto queda probado que o modelo cumpre as restricións de integridade referencial para a modificación de filas na táboa á que fai referencia a clave foránea. (*'on update cascade'*).

* Tarefa 3.3

Cando se executa o *script,* bórrase a fila correspondente ao cliente que ten o dni 28333845. Debido a que a clave foránea ten asociada a restrición *'on delete cascade'*, faise un borrado 'en cascada' de todas as filas que corresponden a ese cliente da táboa *telefono*.



u4a3tarefa0302.emf

Con isto queda probado que o modelo cumpre as restricións de integridade referencial para o borrado de filas na táboa á que fai referencia a clave foránea. (*'on delete cascade'*).

* + 1. Tarefa 4. Verificar restrición UNIQUE

Executar o seguinte *script* de sentenzas SQL para inserir unha fila na táboa *profesor* da base de datos *eleccionmodulos.*

-- habilitando o modo SQL estrito

**set** **session** sql\_mode **=** 'STRICT\_ALL\_TABLES,NO\_ENGINE\_SUBSTITUTION'**;**

-- inserción dunha fila cun valor duplicado nunha clave única

**use** eleccionmodulos**;**

**insert** **into** profesor **(**idProfesor**,** corpo**,** nrp**,** nome**,** apelidos**,** horasLectivas**)** **values**

**(**'fermin'**,** 'T'**,** '33858452.952'**,** 'Fermín'**,** 'Rois Varela'**,** 17**);**

Solución

Cando se executa a sentenza para inserir un novo profesor, móstrase unha mensaxe de erro informando que xa existe unha fila que contén o mesmo valor na columna *nrp* que é unha clave única. Non hai cambios no contido da táboa.



Con este exemplo queda probado que o modelo cumpre as restricións de clave única para a columna *nrp* da táboa *profesor*.

* + 1. Tarefa 5. Verificar restrición DEFAULT

Executar o seguinte *script* de sentenzas SQL para inserir unha fila na táboa *asunto* da base de datos *gabinete.*

-- habilitando o modo SQL estrito

**set** **session** sql\_mode **=** 'STRICT\_ALL\_TABLES,NO\_ENGINE\_SUBSTITUTION'**;**

-- columnas con valor por defecto: dataInicio e dataArquivo

-- columna autoincremental: expediente

**use** gabinete**;**

**insert** **into** asunto **(**descricion**,** estado**,** dniCliente**) values**

**(**'Demanda a compañía de telefonía móbil por incumprimento'**,** 'recopilación'**,**'32022100'**);**

Solución

Cando se executa a sentenza de inserción, a mensaxe que mostra o servidor informa que foi inserida unha fila con éxito.



Na sentenza de inserción só se asignan valores para tres columnas (*descricion*, *estado* e *dniCliente*), as columnas *dataInicio* e *dataArquivo* toman o valor definido como valor por defecto ('*default current\_timestamp'* para a primeira e '*default null*' para a segunda), e no caso da columna *expediente,* o sistema ten que calcular o valor que lle corresponde por que está definida coa propiedade AUTO\_INCREMENT.



Con este exemplo queda probado que o modelo cumpre a restrición DEFAULT para as columnas *dataInicio* e *dataArquivo* da táboa *asunto*.

* + 1. Tarefa 6. Verificar restricións implícitas nos tipos ENUM, e SET

Executar o seguinte *script* de sentenzas SQL para inserir filas na táboa *profesor* da base de datos *eleccionmodulos*.

-- habilitando o modo SQL estrito

**set** **session** sql\_mode **=** 'STRICT\_ALL\_TABLES,NO\_ENGINE\_SUBSTITUTION'**;**

-- probas para columna e tipo ENUM. Valor non válido

**use** eleccionmodulos**;**

**insert** **into** profesor **(**idProfesor**,** corpo**,** nrp**,** nome**,** apelidos**)** **values**

**(**'loli'**,** 'B'**,** '01258450.40'**,** 'Dolores'**,** 'Medina Corredoira'**);**

-- probas para columna e tipo ENUM. Valor válido

**insert** **into** profesor **(**idProfesor**,** corpo**,** nrp**,** nome**,** apelidos**)** **values**

**(**'juan'**,** 'S'**,** '22369741.27'**,** 'Juan Antonio'**,** 'Zamora Barragán'**);**

Solución

Cando se executa a primeira sentenza de inserción, móstrase unha mensaxe de erro que informa que o valor asignado á columna *corpo* non é válido. Isto é debido a que o tipo de dato da columna é *enum('S','T'),* que significa que os únicos valores válidos para a columna son as letras 'S' ou 'T', e ademais, non permite o valor nulo.



A sentenza seguinte execútase correctamente porque toma o valor 'S' que é un valor permitido.



Datos inseridos na táboa *profesor* despois de executar o *script*:



Con este exemplo queda probado que o modelo cumpre a restricións de tipo ENUM para a columna *corpo* da táboa *profesor*.

* + 1. Tarefa 7. Tarefa integradora sobre verificación de restricións

Deseñar un xogo de datos de proba para verificar que se cumpren as restricións do modelo para á táboa *asunto* da base de datos *gabinete*.

Solución

-- habilitando o modo SQL estrito

**set** **session** sql\_mode **=** 'STRICT\_ALL\_TABLES,NO\_ENGINE\_SUBSTITUTION'**;**

-- inserindo filas na táboa asunto

**use** gabinete**;**

**insert** **into** asunto **(**descricion**,**estado**,**dniCliente**)**

**values** **(**'Licenza de obras para comunidade de vecinos'**,**'tramite'**,**'33458745'**);**

**insert** **into** asunto **(**dataInicio**,**descricion**,**dataArquivo**,**estado**,**dniCliente**)**

**values** **(**now**(),**'Reclamación cuota comunidad'**,null,**'tramite'**,**'33458745'**);**

-- restrición de clave primaria

**insert** **into** asunto **(**expediente**,**dataInicio**,**descricion**,**dataArquivo**,**estado**,**dniCliente**)**

**values** **(**2**,**now**(),**'Xestión de alquileres'**,null,**'recopilación'**,**'33458745'**);**

**insert** **into** asunto **(**expediente**,**dataInicio**,**descricion**,**dataArquivo**,**estado**,**dniCliente**)**

**values** **(null,**now**(),**'Descrición do asunto'**,null,**'recopilación'**,**'33458745'**);**

-- restrición de clave foránea

-- inserción (fila filla)

**insert** **into** asunto **(**dataInicio**,**descricion**,**dataArquivo**,**estado**,**dniCliente**)**

**values** **(**now**(),**'Xestión de alquileres'**,null,**'recopilación'**,**'33825123'**);**

-- modificación (fila pai) opción on update cascade

**update** cliente

**set** dni **=** '33200100'

**where** dni **=** '32022100'**;**

-- borrado (fila pai) opción on delete restrict

**delete** **from** cliente

**where** dni **=** '33458745'**;**

-- restrición DEFAULT

**insert** **into** asunto **(**descricion**,**estado**,**dniCliente**)**

**values** **(**'Xestión de alquileres'**,**'recopilación'**,**'33458745'**);**

-- restrición de tipo ENUM

**insert** **into** asunto **(**descricion**,**estado**,**dniCliente**)**

**values** **(**'Tramitación subvención ascensor'**,**'pendiente contrato'**,**'33458745'**);**

-- restrición de valor NULL

**insert** **into** asunto **(**descricion**,**estado**,**dniCliente**)**

**values** **(**'Tramitación subvención ascensor'**,null,**'33458745'**);**

-- restrición asociadas ao tipo e tamaño

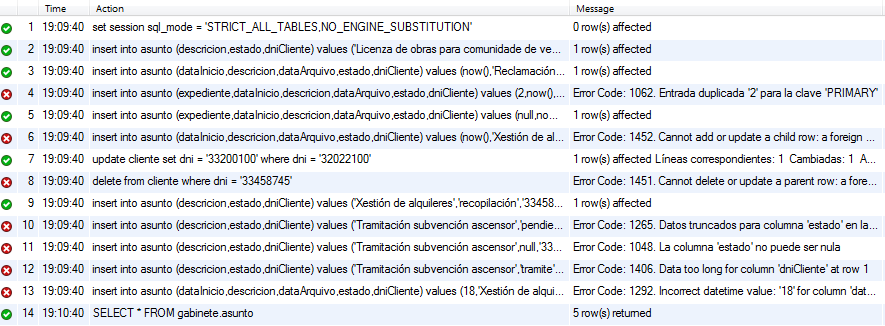
**insert** **into** asunto **(**descricion**,**estado**,**dniCliente**)**

**values** **(**'Tramitación subvención ascensor'**,**'tramite'**,**'27COM33458745'**);**

**insert** **into** asunto **(**dataInicio**,**descricion**,**dataArquivo**,**estado**,**dniCliente**)**

**values** **(**18**,**'Xestión de alquileres'**,null,**'recopilación'**,**'33458745'**);**

Copia de pantalla das mensaxes obtidas ao executar o *script*:



* A mensaxe 1 informa do cambio feito na variable *sql\_mode* para habilitar o modo estrito. Corresponde á sentenza:

**set** **session** sql\_mode **=** 'STRICT\_ALL\_TABLES,NO\_ENGINE\_SUBSTITUTION'**;**

* As mensaxes 2 e 3 informan que as dúas primeiras insercións executáronse correctamente. Corresponde ás sentenzas:

**insert** **into** asunto **(**descricion**,**estado**,**dniCliente**)**

**values** **(**'Licenza de obras para comunidade de vecinos'**,**'tramite'**,**'33458745'**);**

**insert** **into** asunto **(**dataInicio**,**descricion**,**dataArquivo**,**estado**,**dniCliente**)**

**values** **(**now**(),**'Reclamación cuota comunidad'**,null,**'tramite'**,**'33458745'**);**

As filas inseridas cumpren as restricións asociadas ás columnas da táboa.

* A mensaxe 4 mostra un erro producido ao intentar inserir unha fila na táboa asunto cun valor na clave primaria (2) que xa existe. Corresponde á sentenza:

**insert** **into** asunto **(**expediente**,**dataInicio**,**descricion**,**dataArquivo**,**estado**,**dniCliente**)**

**values** **(**2**,**now**(),**'Xestión de alquileres'**,null,**'recopilación'**,**'33458745'**);**

Con este exemplo queda probado que o modelo cumpre a restrición de clave primaria para a táboa *asunto*.

* A mensaxe 5 informa que a inserción executouse correctamente, a pesar de asignar o valor nulo para a clave primaria. Isto sucede porque a columna *expediente* está definida coa propiedade AUTO\_INCREMENT, e no caso de non asignar ningún valor á columna, ou o valor nulo, o sistema asígnalle o valor que corresponde ao seguinte número enteiro ao asignado á última fila inserida. Corresponde á sentenza:

**insert** **into** asunto **(**expediente**,**dataInicio**,**descricion**,**dataArquivo**,**estado**,**dniCliente**)**

**values** **(null,**now**(),**'Descrición do asunto'**,null,**'recopilación'**,**'33458745'**);**

* A mensaxe 6 mostra unha mensaxe de erro informando que non se pode inserir unha fila na táboa *asunto* (fila filla) se o valor da clave primaria non é igual a algún dos valores que toma a clave primaria da táboa *cliente*. Corresponde á sentenza:

**insert** **into** asunto **(**dataInicio**,**descricion**,**dataArquivo**,**estado**,**dniCliente**)**

**values** **(**now**(),**'Xestión de alquileres'**,null,**'recopilación'**,**'33825123'**);**

Con isto queda probado que o modelo cumpre as restricións de integridade referencial para a inserción de filas na táboa *telefono*.

* A mensaxe 7 informa que se fixo a modificación da columna dni, que está definida como clave primaria, para o cliente con dni número '32022100'. O novo valor para a columna dni é '33200100'. Tendo en conta que ao crear a clave foránea dniCliente na táboa *asunto* utilizouse a opción 'on update cascade', faise unha modificación 'en cascada' de todas as filas da táboa *asunto* nas que a clave foránea fai referencia a ese cliente, cambiando o valor que teñen polo novo valor que toma a clave primaria. Corresponde á sentenza:

**update** cliente

**set** dni **=** '33200100'

**where** dni **=** '32022100'**;**

Con este exemplo queda probado que o modelo cumpre as restricións de integridade referencial para a modificación de filas na táboa á que fai referencia a clave foránea. (*'on update cascade'*).

* A mensaxe 8 mostra un erro informando que non se pode borrar unha fila da táboa *cliente* (fila pai) se hai algunha clave foránea que ten o valor igual ao valor da clave primaria da fila que se intenta borrar. Corresponde á sentenza:

**delete** **from** cliente

**where** dni **=** '33458745'**;**

Con este exemplo queda probado que o modelo cumpre as restricións de integridade referencial para o borrado de filas na táboa á que fai referencia a clave foránea. (*'on delete restrict*).

* A mensaxe 9 informa que foi inserida unha fila con éxito. Na sentenza de inserción só se asignan valores para tres columnas (*descricion*, *estado* e *dniCliente*), as columnas *dataInicio* e *dataArquivo* toman o valor definido como valor por defecto ('*default current\_timestamp'* para a primeria e '*default null*' para a segunda), e no caso da columna *expediente* o sistema ten que calcular o valor que lle corresponde porque está definida coa propiedade AUTO\_INCREMENT.



Corresponde á sentenza:

**insert** **into** asunto **(**descricion**,**estado**,**dniCliente**)**

**values** **(**'Xestión de alquileres'**,**'recopilación'**,**'33458745'**);**

Con este exemplo queda probado que o modelo cumpre a restrición DEFAULT para as columnas *dataInicio* e *dataArquivo* da táboa *asunto*.

* A mensaxe 10 mostra unha mensaxe de erro porque o valor asignado á columna *estado* é distinto dos valores permitidos ('recopilación','tramite','xuizo','arquivado). Corresponde á sentenza:

**insert** **into** asunto **(**descricion**,**estado**,**dniCliente**)**

**values** **(**'Tramitación subvención ascensor'**,**'pendiente contrato'**,**'33458745'**);**

Con este exemplo queda probado que o modelo cumpre as restricións de tipo ENUM para a columna *estado* da táboa *asunto*.

* A mensaxe 11 mostra unha mensaxe de erro porque se asigna o valor nulo á columna estado e esta foi definida coa propiedade NOT NULL. Corresponde á sentenza:

**insert** **into** asunto **(**descricion**,**estado**,**dniCliente**)**

**values** **(**'Tramitación subvención ascensor'**,null,**'33458745'**);**

Con este exemplo queda probado que o modelo cumpre as restricións de valor NULL para a columna *estado* da táboa *asunto*.

* A mensaxe 12 mostra unha mensaxe de erro porque o tamaño do valor asignado á columna *dniCliente* é maior que o tamaño asignado a esa columna. Corresponde á sentenza:

**insert** **into** asunto **(**descricion**,**estado**,**dniCliente**)**

**values** **(**'Tramitación subvención ascensor'**,**'tramite'**,**'27COM33458745'**);**

Con este exemplo queda probado que o modelo cumpre as restricións de tamaño para a columna *dniCliente* da táboa *asunto*.

* A mensaxe 13 mostra unha mensaxe de erro porque o valor asignado á columna *dataInicio* non é un valor de tipo *datetime*. Corresponde á sentenza:

**insert** **into** asunto **(**dataInicio**,**descricion**,**dataArquivo**,**estado**,**dniCliente**)**

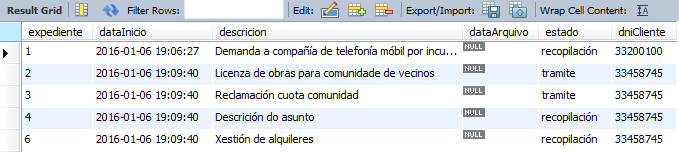
**values** **(**18**,**'Xestión de alquileres'**,null,**'recopilación'**,**'33458745'**);**

Con este exemplo queda probado que o modelo cumpre as restricións de tipo de dato para a columna *dataInicio* da táboa *asunto*.

* A mensaxe 14 informa que foron mostradas 5 filas ao executar a consulta. Corresponde á sentenza:

**select** \* **from** gabinete.asunto

As filas que se mostran son:



1. Materiais
   1. Documentos de apoio ou referencia

* DUBOIS, Paul. *La biblia de MySQL*. Anaya Multimedia, 2009.
* PONS, Olga e outros. *Introdución a las Bases de datos*.
* *Manual de referencia de MySQL 5.6*. <http://dev.mysql.com/doc/>
* *Manual de referencia de MySQL Workbench 6.3*. <http://dev.mysql.com/doc/index-gui.html>
  1. Recursos didácticos
* Material didáctico subministrado polo profesorado en papel e/ou formato dixital.
* Proxector.
* 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 software estritamente necesario para a realización da tarefa de avaliación mediante probas escritas prácticas.
  1. Material auxiliar

O material auxiliar anexo a esta actividade está almacenado na carpeta CSIFC02\_MP0484\_V000403\_ProbasBD\_Anexos que contén:

* O arquivo V000403\_scriptsSQL.zip cos scripts SQL, utilizadas como base para esta actividade.
* O arquivo GuiaWorkbench.docx que contén unha guía breve de manexo 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 |
| CA4.7 Verificouse o axuste da implementación ao modelo mediante un conxunto de datos de proba. | * Exame en papel e en formato dixital. | * **TO.8** Documento de rexistro de proba documentado que inclúa definición dun conxunto de datos de proba, a carga de datos e a interpretación dos resultados.Escala de valores (observación indirecta). | 5 |

Modelo de proba combinada para TO.6

Proponse unha proba práctica para avaliar CA4.7 mediante observación indirecta sobre un documento de rexistro da práctica. A solución da práctica deberá ser entregada polo alumnado nun documento escrito e en dixital.

* O documento escrito debe conter:
* Xogo de datos de proba para verificar que se cumpren as restricións de integridade solicitadas e bloque de sentenzas que cargan o xogo de datos de proba.
* Imaxes capturadas coas mensaxes que devolve o servidor ao executar o bloque de sentenzas dende o cliente Workbench e texto escrito polo alumno coa interpretación das mensaxes que devolve o servidor.
* Os arquivos dixitais deben conter:
* Arquivo .sql co código correspondente ao xogo de probas.
* Capturas das pantallas coas mensaxes que devolve o servidor ao executar o bloque de sentenzas dende o cliente Workbench e texto coa interpretación das mensaxes que devolve o servidor.

A práctica realizarase na base de datos *exame\_probas*. Subminístrase o arquivo *exame\_proba.sql* co conxunto de sentenzas SQL que permiten crear esa base de datos e engadir algunhas filas nas táboas. Contido do *script* que *exame\_proba.sql*:

/\*creacion da base de datos exame\_probas \*/

**create** **database** exame\_probas

**default** **character** **set** utf8

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

**use** exame\_probas**;**

**create** **table** hotel **(**

codigo **integer** unsigned **not** **null** auto\_increment**,**

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

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

email **varchar(**100**)** **default** **null,**

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

**unique** **(**cif**)**

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

**create** **table** habitacion **(**

hotel **integer** unsigned **not** **null,**

numero **smallint(**6**)** **not** **null,**

tipo **enum(**'M'**,**'D'**,**'I'**,**'S'**)** **not** **null,**

supletoria **enum(**'S'**,**'N'**)** **not** **null** **default** 'N'**,**

prezo **decimal(**6**,**2**)** unsigned **not** **null,**

**primary** **key** **(**hotel**,**numero**)**

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

/\* Definición das relacións entre as táboas\*/

**alter** **table** habitacion

**add** **constraint** fk\_habitacion\_hotel **foreign** **key** **(**hotel**)** **references** hotel **(**codigo**)**

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

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

/\* Inserción de datos\*/

**insert** **into** hotel **(**cif**,**nome**,**email**)** **values**

**(**'A15002545'**,**'Hotel Almirante'**,**'info@hotelalmirante.com'**),**

**(**'A32500124'**,**'Hotel San Martín'**,**'sanmartin@sanmartin.es'**),**

**(**'A36325889'**,**'Hotel Peregrina'**,**'info@peregrina.es'**),**

**(**'B27001200'**,**'Hotel Lugo'**,**'reservas@hotellugo.ga'**);**

**insert** **into** habitacion **(**hotel**,**numero**,**tipo**,**supletoria**,**prezo**)** **values**

**(**1**,**101**,**'M'**,**'N'**,**80**),(**1**,**102**,**'I'**,**'N'**,**50**),(**2**,**1**,**'S'**,**'N'**,**120**),(**3**,**11**,**'M'**,**'N'**,**75**),**

**(**4**,**101**,**'D'**,**'S'**,**85**),(**4**,**102**,**'D'**,**'N'**,**80**),(**4**,**103**,**'M'**,**'N'**,**90**);**

Capturas de pantalla dos datos que hai nas táboas despois de executar o *script*:

|  |  |
| --- | --- |
|  |  |

Proba para CA4.7 sobre o axuste da implementación ao modelo mediante un conxunto de datos de proba

Cear un xogo de datos de proba para verificar que se cumpren as seguintes restricións de integridade na base de datos *exame\_probas,* executar as operacións de carga dos datos de proba utilizando o cliente MySQL Workbench e interpretar as mensaxes que devolve o servidor:

* 1.- PRIMARY KEY na táboa *hotel.*
* 2*.-* UNIQUE na táboa *hotel.*
* 3.- NOT NULL na táboa *hotel*.
* 4.- PRIMARY KEY na táboa *habitacion.*
* 5.- Integridade referencial nas táboas *hotel* e *habitacion*.
* 6.- DEFAULT na táboa *habitacion.*
* 7.- ENUM na táboa *habitacion.*
* 8.- Tipo e lonxitude das columnas *número* e *prezo* na táboa *habitacion.*

Exemplo de solución para entregar en papel

* 1.- Comprobación das restricións de integridade PRIMARY KEY na táboa *hotel.*

-- primary key

**insert** **into** hotel **(**codigo**,**cif**,**nome**,**email**)** **values**

**(**1**,**'B27098890'**,**'Hotel Lugo Palace'**,null);**



A mensaxe de erro informa que xa existe un hotel con ese código. Ao ser de tipo autoincremental xa non vai admitir valores nulos, pois é o sistema o que calcula o valor que toma a columna no caso de non asignarlle ningún valor, ou asignarlle o valor NULL.

Queda probado que se cumpren as restricións PRIMARY KEY na táboa *hotel*, porque a clave primaria non admite valores nulos nin valores duplicados.

* 2.- Comprobación das restricións de integridadeUNIQUE na táboa *hotel.*

-- unique key

**insert** **into** hotel **(**cif**,**nome**,**email**)** **values**

**(**'A58818501'**,**'Hotel Primavera'**,**'info@hotelprimavera.es'**);**

**insert** **into** hotel **(**cif**,**nome**,**email**)** **values**

**(**'A58818501'**,**'Hotel Lugo Palace'**,null);**



A primeira mensaxe informa que o hotel con *cif* 'A58818501' foi engadido correctamente. A segunda mensaxe informa dun erro debido a que xa existe un hotel con ese cif e os datos do hotel non son engadidos.

Queda probado que se cumpre a restrición UNIQUE para a columna *cif* da táboa *hotel*.

* 3.- Comprobación das restricións de integridade NOT NULL na táboa *hotel*.

-- valor NULL

**insert** **into** hotel **(**cif**,**nome**,**email**)** **values**

**(null,**'Hotel Lugo Palace'**,null);**



A mensaxe informa dun erro debido a que a columna *cif* non pode tomar o valor nulo, e os datos do hotel non son engadidos.

Queda probado que se cumpre a restrición NOT NULL para a columna *cif*.

* 4.- Comprobación das restricións de integridade PRIMARY KEY na táboa *habitacion.*

-- primary key

**insert** **into** habitacion **(**hotel**,**numero**,**tipo**,**supletoria**,**prezo**)** **values**

**(**1**,**101**,**'M'**,**'N'**,**65**);**

**insert** **into** habitacion **(**hotel**,**numero**,**tipo**,**supletoria**,**prezo**)** **values**

**(**1**,null,**'M'**,**'N'**,**80**);**



A primeira mensaxe de erro informa que xa existe unha fila na táboa *habitacion* co mesmo valor na clave primaria. A segunda mensaxe erro informa que a columna *numero* non pode tomar o valor nulo, debido a que forma parte da clave primaria.

Queda probado que se cumpren as restricións PRIMARY KEY na táboa *habitacion*, porque a clave primaria non admite valores duplicados, nin o valor NULL.

* 5.- Comprobación das restricións de integridade referencial nas táboas *hotel* e *habitacion*.

-- Integridade referencial inserción en habitacion

**insert** **into** habitacion **(**hotel**,**numero**,**tipo**,**supletoria**,**prezo**)** **values**

**(**10**,**1**,**'M'**,**'N'**,**55**);**

**insert** **into** habitacion **(**hotel**,**numero**,**tipo**,**supletoria**,**prezo**)** **values**

**(**5**,**1**,**'M'**,**'N'**,**55**);**



A primeira mensaxe de erro informa que non se pode engadir a fila na táboa porque o valor da clave foránea non cumpre as restricións. Non existe ningún hotel co código 10. O texto completo da mensaxe é:

|  |
| --- |
| Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`exame\_probas`.`habitacion`, CONSTRAINT `fk\_habitacion\_hotel` FOREIGN KEY (`hotel`) REFERENCES `hotel` (`codigo`) ON UPDATE CASCADE) |

A segunda mensaxe informa que a fila foi inserida correctamente. O valor da columna *hotel* (5) coincide co código dun hotel que existe na táboa *hotel*. Despois de facer as insercións, as táboas teñen estes datos:

|  |  |
| --- | --- |
|  |  |

-- Integridade referencial modificación en hotel

**update** hotel

**set** codigo **=** 6

**where** codigo **=** 1**;**



A mensaxe informa que a fila foi modificada correctamente. Asígnase o valor 6 á columna *codigo* para a fila da táboa *hotel* que ten o valor 1 na columna *codigo*. Cando se definiu a clave foránea utilizouse a opción *'on update cascade'*, polo que agora faise unha modificación 'en cascada' do valor da clave foránea para todas as filas da táboa *habitacion* nas que ten o valor 1. A clave foránea para esas filas pasaría a tomar o valor 6. Despois de facer a modificación, as táboas teñen estes datos:

u4a3exame2.emf

-- Integridade referencial borrado en hotel

**delete** **from** hotel

**where** codigo**=**2**;**



A mensaxe de erro informa que non se pode engadir a fila na táboa porque o valor da clave foránea non cumpre as restricións. Cando se definiu a clave foránea utilizouse a opción *'on delete restrict',* polo que non está permitido borrar unha fila da táboa *hotel* cando hai filas na táboa *habitacion* relacionadas con el. Texto completo da mensaxe:

|  |
| --- |
| Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`exame\_probas`.`habitacion`, CONSTRAINT `fk\_habitacion\_hotel` FOREIGN KEY (`hotel`) REFERENCES `hotel` (`codigo`) ON UPDATE CASCADE) |

Queda probado que se verifican as restricións de integridade referencial.

* 6.- Comprobación das restricións de integridade DEFAULT na táboa *habitacion.*

-- DEFAULT

**insert** **into** habitacion **(**hotel**,**numero**,**tipo**,**prezo**)** **values**

**(**4**,**201**,**'M'**,**65**);**



A mensaxe informa que a fila foi inserida correctamente. Como non se asignou ningún valor á columna *supletoria*, esta tomará o valor por defecto asignado de forma explícita ao crear a táboa. Captura de pantalla co contido da fila:

u4a3exame1.emf

Queda probado que se verifica a restrición DEFAULT para a columna *supletoria*.

* 7.- Comprobación das restricións de integridade ENUM na táboa *habitacion.*

-- ENUM

**insert** **into** habitacion **(**hotel**,**numero**,**tipo**,**supletoria**,**prezo**)** **values**

**(**4**,**202**,**'M'**,**'N'**,**80**);**

**insert** **into** habitacion **(**hotel**,**numero**,**tipo**,**supletoria**,**prezo**)** **values**

**(**4**,**202**,**'J'**,**'N'**,**80**);**



A primeira mensaxe informa que a fila foi inserida correctamente, porque os valores asignados ás columnas de tipo ENUM están dentro da lista de valores permitidos. A segunda mensaxe advirte dun erro que se produce porque a columna *tipo* toma un valor que non está permitido.

Queda probado que se verifican as restricións implícitas no tipo de dato ENUM.

* 8.- Comprobación das restricións de integridade tipo e lonxitude das columnas *número* e *prezo* na táboa *habitacion.*

-- Tipo de dato e tamaño para as columnas numero e

**insert** **into** habitacion **(**hotel**,**numero**,**tipo**,**supletoria**,**prezo**)** **values**

**(**4**,**'Picasso'**,**'D'**,**'N'**,**80**);**

**insert** **into** habitacion **(**hotel**,**numero**,**tipo**,**supletoria**,**prezo**)** **values**

**(**4**,**203**,**'D'**,**'N'**,**10000**);**



A primeira mensaxe de erro informa que o valor asignado á columna *numero* non é de tipo enteiro (*integer*). A segunda mensaxe informa dun erro que se produce porque o valor asignado á columna *prezo* está fora do rango de valores permitidos, que van dende 0 a 9999.99 por estar definido como *decimal(6,2) unsigned*.

Queda probado que se cumpren as restricións asociadas ao tipo de dato e tamaño das columnas *numero* e *prezo* da táboa *habitación*.

O *script* completo que realiza a carga dos datos de proba para comprobar as restricións de integridade pedidas é o seguinte:

-- habilitando o modo SQL estrito

**set** **session** sql\_mode **=** 'STRICT\_ALL\_TABLES,NO\_ENGINE\_SUBSTITUTION'**;**

-- TÁBOA: hotel

-- primary key

**insert** **into** hotel **(**codigo**,**cif**,**nome**,**email**)** **values**

**(**1**,**'B27098890'**,**'Hotel Lugo Palace'**,null);**

-- unique key

**insert** **into** hotel **(**cif**,**nome**,**email**)** **values**

**(**'A58818501'**,**'Hotel Primavera'**,**'info@hotelprimavera.es'**);**

**insert** **into** hotel **(**cif**,**nome**,**email**)** **values**

**(**'A58818501'**,**'Hotel Lugo Palace'**,null);**

-- valor NULL

**insert** **into** hotel **(**cif**,**nome**,**email**)** **values**

**(null,**'Hotel Lugo Palace'**,null);**

-- TÁBOA: habitacion

-- primary key

**insert** **into** habitacion **(**hotel**,**numero**,**tipo**,**supletoria**,**prezo**)** **values**

**(**1**,**101**,**'M'**,**'N'**,**65**);**

**insert** **into** habitacion **(**hotel**,**numero**,**tipo**,**supletoria**,**prezo**)** **values**

**(**1**,null,**'M'**,**'N'**,**80**);**

-- foreign key inserción

**insert** **into** habitacion **(**hotel**,**numero**,**tipo**,**supletoria**,**prezo**)** **values**

**(**10**,**1**,**'M'**,**'N'**,**55**);**

**insert** **into** habitacion **(**hotel**,**numero**,**tipo**,**supletoria**,**prezo**)** **values**

**(**5**,**1**,**'M'**,**'N'**,**55**);**

-- foreign key modificación

**update** hotel

**set** codigo **=** 6

**where** codigo **=** 1**;**

-- foreign key borrado

**delete** **from** hotel

**where** codigo**=**2**;**

-- DEFAULT

**insert** **into** habitacion **(**hotel**,**numero**,**tipo**,**prezo**)** **values**

**(**4**,**201**,**'M'**,**65**);**

-- ENUM

**insert** **into** habitacion **(**hotel**,**numero**,**tipo**,**supletoria**,**prezo**)** **values**

**(**4**,**202**,**'M'**,**'N'**,**80**);**

**insert** **into** habitacion **(**hotel**,**numero**,**tipo**,**supletoria**,**prezo**)** **values**

**(**4**,**202**,**'J'**,**'N'**,**80**);**

-- Tipo de dato e tamaño para as columnas numero e

**insert** **into** habitacion **(**hotel**,**numero**,**tipo**,**supletoria**,**prezo**)** **values**

**(**4**,**'Picasso'**,**'D'**,**'N'**,**80**);**

**insert** **into** habitacion **(**hotel**,**numero**,**tipo**,**supletoria**,**prezo**)** **values**

**(**4**,**203**,**'D'**,**'N'**,**10000**);**

Exemplo de lista de valoración para TO.6

Proponse a seguinte lista de valoración para o instrumento de avaliación TO.6 (observación indirecta: documento de rexistro do deseño dun xogo de datos de proba para validar as restricións do modelo e a interpretación das mensaxes resultantes da execución da proba).

| Nome | Data | |
| --- | --- | --- |
| Indicadores para  CA4.7 Verificouse o axuste da implementación ao modelo mediante un conxunto de datos de proba. | Valoración máxima  5 | Cualificación |
| O conxunto de datos de proba permite comprobar as restriccións propostas. | 2 |  |
| As imaxes capturadas reflicten o resultado da execución das operación de manexo de datos executadas dende o cliente Workbench. | 1 |  |
| As mensaxes que resultan da execución do xogo de probas son interpretadas correctamente. | 2 |  |

1. Ver lista de modos e o seu significado no apartado 5.1.7 Server SQL Modes do manual de referencia de MySQL 5.6. [↑](#footnote-ref-3)