Dirección Xeral de Educación, Formación Profesional e Innovación Educativa

Material para

a formación profesional inicial

A02. Vistas

|  |  |  |
| --- | --- | --- |
| 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 | UD06 | Tratamento de datos |
| **Actividade** | **A02** | **Vistas** |
| 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\_V000602\_UD06\_A02\_Vistas.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](#_Toc442188477)

[Contexto da actividade 4](#_Toc442188478)

[Título da actividade 5](#_Toc442188479)

[Resultados de aprendizaxe do currículo 5](#_Toc442188480)

[Obxectivos didácticos e título e descrición da actividade 5](#_Toc442188481)

[Criterios de avaliación 5](#_Toc442188482)

[Contidos 5](#_Toc442188483)

[Actividades de ensino e aprendizaxe e de avaliación, métodos, recursos e instrumentos de avaliación 6](#_Toc442188484)

[2. A02. Vistas 7](#_Toc442188485)

[2.1 Introdución 7](#_Toc442188486)

[2.1.1 Obxectivos 7](#_Toc442188487)

[2.1.2 Software 7](#_Toc442188488)

[2.1.3 Bases de datos de traballo 8](#_Toc442188489)

[2.1.3.1 Base de datos tendasbd 8](#_Toc442188490)

[2.1.3.2 Base de datos practicas5 10](#_Toc442188491)

[2.1.3.3 Base de datos traballadores 11](#_Toc442188492)

[2.2 Actividade 12](#_Toc442188493)

[2.2.1 As vistas 12](#_Toc442188494)

[2.2.1.1 Sentenza CREATE VIEW e utilización de vistas 13](#_Toc442188495)

[Exemplos 14](#_Toc442188496)

[2.2.1.2 Sentenza SHOW CREATE VIEW 18](#_Toc442188497)

[2.2.1.3 Sentenza ALTER VIEW 19](#_Toc442188498)

[2.2.1.4 Sentenza DROP VIEW 19](#_Toc442188499)

[2.2.2 Manipulación de datos a través das vistas 19](#_Toc442188500)

[2.2.3 Avantaxes do uso de vistas 20](#_Toc442188501)

[2.3 Tarefas 21](#_Toc442188502)

[2.3.1 Tarefa 1. Crear e utilizar vistas 21](#_Toc442188503)

[Solución 23](#_Toc442188504)

[3. Materiais 28](#_Toc442188505)

[3.1 Documentos de apoio ou referencia 28](#_Toc442188506)

[3.2 Recursos didácticos 28](#_Toc442188507)

[3.3 Material auxiliar 28](#_Toc442188508)

[4. Avaliación 29](#_Toc442188509)

[Criterios de avaliación 29](#_Toc442188510)

[Modelo de proba para TO.1 29](#_Toc442188511)

[Proba para CA5.10.1 sobre creación e utilización de vistas 30](#_Toc442188512)

[Exemplo de solución para entregar en papel 31](#_Toc442188513)

[Exemplo de lista de valoración para TO.1 33](#_Toc442188514)

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\_V000600\_UD06\_TratamentoDatos.pdf

Título da actividade

|  |  |  |  |
| --- | --- | --- | --- |
| Nº | Título | Descrición | Duración |
| A02 | Vistas. | Crearanse, modificaranse, eliminaranse e utilizaranse vistas para facer consultas e operacións de actualización de datos. | 5 |

Resultados de aprendizaxe do currículo

|  |  |
| --- | --- |
| Resultados de aprendizaxe do currículo | Completo |
| * RA5 - Consulta a información almacenada nunha base de datos empregando asistentes, ferramentas gráficas e a linguaxe de manipulación de datos. * RA6 -  Modifica a información almacenada na base de datos utilizando asistentes, ferramentas gráficas e a linguaxe de manipulación de datos. | Non  Si |

Obxectivos didácticos e título e descrición da actividade

| Obxectivos específicos | | Actividade | | Descrición básica | Duración |
| --- | --- | --- | --- | --- | --- |
| O2.1 | Crear, modificar, eliminar e utilizar vistas utilizando a linguaxe de manipulación de datos. | A02 | Vistas. | Crearanse, modificaranse, eliminaranse e utilizaranse vistas para facer consultas e operacións de actualización de datos. | 5 |

Criterios de avaliación

|  |
| --- |
| Criterios de avaliación |
| * CA5.10 - Creáronse vistas.   CA5.10.1 - Creáronse e utilizáronse vistas. |

Contidos

| Contidos |
| --- |
| * Vistas. |

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. Vistas.   * Crearanse, modificaranse, eliminaranse e utilizaranse vistas para facer consultas e operacións de actualización de datos. | * **Tp2.1** Exposición sobre a sintaxe da LMD (linguaxe de manipulación de datos) para crear e utilizar vistas. | **Ta2.1** Toma de notas e formulación de dúbidas. |  | * Material didáctico subministrado polo profesorado en papel e/ou formato dixital. * Proxector. * Ordenadores con conexión a Internet, que terán instalado MySQL e MySQL Workbench. * Manual de referencia de MySQL. |  | 0.5 |
| * **Tp2.2** Demostración práctica de creación e utilización de vistas para realizar consultas e operacións de actualización de datos mediante LMD. | * **Ta2.2** Creación e utilización de vistas para realizar consultas e operacións de actualización de datos mediante a linguaxe de manipulación de datos LMD. | * Guións de sentenzas SQL que solucionan os exercicios propostos. |  | 3.75 |
|  | * **Ta2.3** Tarefa de avaliación utilizando o instrumento de avaliación TO.1. | * Exame en papel e en formato dixital. | * Máquina virtual para exame que terá instalado MySQL e MySQL Workbench. * Manual de referencia de MySQL. | * **TO.1** - Documento de rexistro de creación e utilización de vistas. Escala de valores (observación indirecta). | 0.75 |

1. A02. Vistas
   1. Introdución
      1. Obxectivos

O obxectivo desta actividade é:

* Crear, modificar, eliminar e utilizar vistas coa LMD (linguaxe de manipulación 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 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.

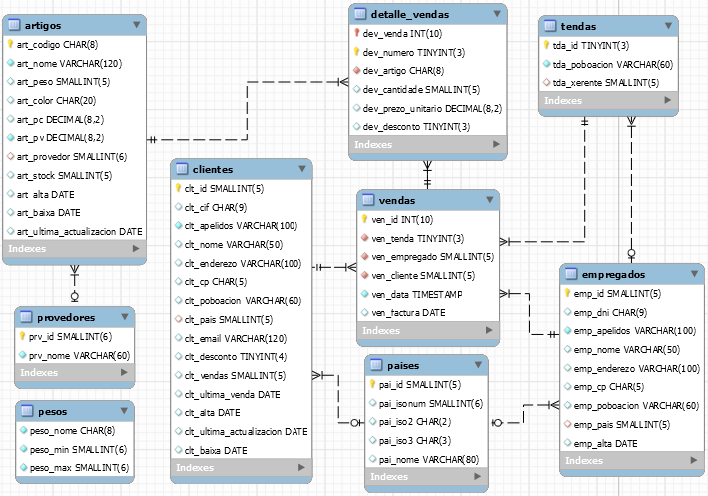
Normalmente, para ilustrar as probas realizadas nesta actividade, mostrarase unha imaxe da zona de manipulación de datos de Workbench co resultado da utilización da vista. Para completar a anterior información ou cando o resultado non poda mostrarse completo, mostrarase ademais unha imaxe con outra zona de Workbench que a complemente.

* 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](http://dev.mysql.com/doc/index-gui.html%20) pode descargarse o manual de MySQL Workbench.
* O material anexo a esta actividade inclúe unha guía básica de MySQL Workbench 6.3.
  + 1. Bases de datos de traballo

As bases de datos *tendasBD*, *traballadores* e *practicas5* 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 tendasbd

A base de datos *tendasBD* serve para controlar as vendas dunha cadea de tendas. Gárdanse nela os datos das vendas que se realizan, das tendas nas que se fan as vendas, dos artigos vendidos, e dos clientes. As táboas desta base de datos que se van a utilizar nesta actividade móstranse no seguinte diagrama entidade relación deseñado con Workbench e descríbense a continuación.



* Táboa *empregados*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome columna | Tipo | Null | Clave | Observacións |
| emp\_id | smallint unsigned | Non | Primaria | Identificador do empregado. Numéranse de 1 en adiante de forma automática. |
| emp\_dni | char(9) |  |  | DNI do empregado. |
| emp\_apelidos | varchar(100) | Non | Índice | Apelidos do empregado. |
| emp\_nome | varchar(50) |  | Nome do empregado. |
| emp\_enderezo | varchar(100) |  |  | Enderezo do empregado. |
| emp\_cp | char(5) |  |  | Código postal do empregado. |
| emp\_poboacion | varchar(60) |  |  | Poboación do empregado. |
| emp\_pais | smallint unsigned |  | Foránea | Código do país segundo a táboa de países. |
| emp\_alta | date |  |  | Data na que se deu de alta o empregado. |

* Táboa *pesos*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome columna | Tipo | Null | Clave | Observacións |
| peso\_nome | char(8) | Non |  | Nome que describe o tipo de peso. |
| peso\_min | smallint | Non |  | Peso mínimo para ese nome. |
| peso\_max | smallint | Non |  | Peso máximo para ese nome. |

* Táboa *clientes*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome columna | Tipo | Null | Clave | Observacións |
| clt\_id | smallint unsigned | Non | Primaria | Identificador do cliente. Numeraranse de 1 en adiante de forma automática. |
| clt\_cif | char(9) |  | Única |  |
| clt\_apelidos | varchar(100) | Non | Índice | Apelidos ou razón social do cliente. |
| clt\_nome | varchar(50) |  | Nome ou tipo de sociedade (SL, SA, ...) do cliente. |
| clt\_enderezo | varchar(100) |  |  |  |
| clt\_cp | char(5) |  |  | Código postal do cliente. |
| clt\_poboacion | varchar(60) |  |  |  |
| clt\_pais | smallint unsigned |  | Foránea | Código do país segundo a táboa de países. |
| clt\_email | varchar(120) |  |  |  |
| clt\_desconto | tinyint |  |  | Porcentaxe de desconto aplicable ao cliente. |
| clt\_vendas | smallint unsigned |  |  | Número de vendas feitas ao cliente. |
| clt\_ultima\_venda | date |  |  | Data da última venda feita ao cliente. |
| clt\_alta | date | Non |  | Data na que se deu de alta ao cliente. |
| clt\_ultima\_actualizacion | date |  |  | Data da última vez que se fixeron cambios nos datos do cliente. |
| clt\_baixa | date |  |  | Data na que se deu de baixa ao cliente. |

* Táboa *artigos*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome columna | Tipo | Null | Clave | Observacións |
| art\_codigo | char(8) | Non | Primaria | Toma valores entre 1 e 200.000. |
| art\_nome | varchar(120) | Non | Índice | Nome ou descrición do artigo. |
| art\_peso | smallint unsigned |  |  | Peso en gramos. Valor numérico enteiro. |
| art\_color | char(20) |  |  | Cor do artigo |
| art\_pc | decimal(8,2) |  |  | Prezo de compra do artigo. |
| art\_pv | decimal(8,2) | Non |  | Prezo de venda do artigo. |
| art\_provedor | smallint |  | Foránea | Identificador do provedor. |
| art\_stock | smallint unsigned |  |  | Número de unidades do artigo dispoñibles no almacén. |
| art\_alta | date | Non |  | Data na que se deu de alta o artigo. |
| art\_baixa | date |  |  | Data na que se deu de baixa o artigo. |
| art\_ultima\_actualizacion | date |  |  | Data da última vez que se fixeron cambios nos datos do artigo. |

* Táboa *paises*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome columna | Tipo | Null | Clave | Observacións |
| pai\_id | smallint unsigned | Non | Primaria | Identificador do país. Numeraranse de 1 en adiante de forma automática. |
| pai\_isonum | smallint |  |  | Número de país segundo a norma ISO 3166-1:2013.[[1]](#footnote-3) |
| pai\_iso2 | char(2) |  |  | Código de país de 2 carácteres segundo a norma ISO 3166-1:2013. |
| pai\_iso3 | char(3) |  |  | Código de país de 3 carácteres segundo a norma ISO 3166-1:2013. |
| pai\_nome | varchar(80) |  |  | Nome do país. |

* Táboa *provedores*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome columna | Tipo | Null | Clave | Observacións |
| prv\_id | smallint | Non | Primaria | Identificador do provedor. |
| prv\_nome | varchar(60) | Non |  | Nome do provedor. |

* Táboa *tendas*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome columna | Tipo | Null | Clave | Observacións |
| tda\_id | tinyint unsigned | Non | Primaria | Identificador da tenda. Numéranse do 1 en adiante de forma automática. |
| tda\_poboacion | varchar(60) | Non |  | Poboación na que está situada a tenda. |
| tda\_xerente | smallint unsigned |  | Foránea | Identificador do empregado que é xerente da tenda. |

* Táboa *vendas*

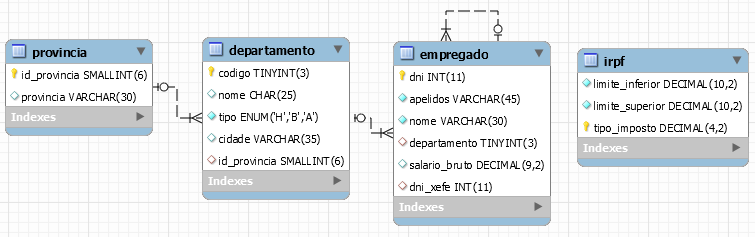
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome columna | Tipo | Null | Clave | Observacións |
| ven\_id | int unsigned | Non | Primaria | Identificador da venda. Numeraranse de 1 en adiante de forma automática. |
| ven\_tenda | tinyint unsigned | Non | Foránea | Identificador da tenda na que se fixo a venda. |
| ven\_empregado | smallint unsigned | Non | Foránea | Identificador do empregado que fixo a venda. |
| ven\_cliente | smallint unsigned | Non | Foránea | Identificador do cliente ao que se fixo a venda. |
| ven\_data | date | Non |  | Data e hora na que se fixo a venda. |
| ven\_factura | date |  |  | Data da factura na que se inclúe esta venda. |

* Táboa *detalle\_vendas*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nome columna | Tipo | Null | Clave | | Observacións |
| dev\_venda | int unsigned | Non | Primaria | Foránea | Identificador da venda á que corresponde a liña de detalle. |
| dev\_numero | tinyint unsigned | Non |  | Número da liña de detalle dentro da venda. |
| dev\_artigo | char(8) | Non | Foránea | | Identificador do artigo vendido. |
| dev\_cantidade | smallint unsigned | Non |  | | Número de unidades vendidas. |
| dev\_prezo\_unitario | decimal(8,2) unsigned | Non |  | | Prezo por cada unidade vendida. |
| dev\_desconto | tinyint unsigned | Non |  | | Porcentaxe de desconto aplicado. |

* + - 1. Base de datos practicas5

A base de datos *practicas5* está creada con fins didácticos para realizar os exemplos de consultas nesta unidade. Está formada por un grupo de táboas, relacionadas entre si, tal e como se mostra no seguinte diagrama entidade relación deseñado con Workbench e se describe a continuación.



* Táboa *empregado.*  A columna *departamento* é unha clave foránea que contén o código do departamento no que traballa o empregado, e fai referencia á columna *codigo* da táboa *departamento*. Os valores que toma a columna *departamento* teñen que coincidir cos que toma a columna *codigo* da táboa *departamento*, ou ser NULL no caso que o empregado non teña asignado ningún departamento. A columna *dni\_xefe* é outra clave foránea que contén o dni doutro empregado que sería o seu xefe, ou o valor NULL no caso que non tivera xefe.
* Táboa *departamento*. A columna *id\_provincia* é unha clave foránea que fai referencia á columna *id\_provincia* da táboa *provincia*.
* Táboa *irpf.* Contén a porcentaxe de imposto que hai que aplicarlle a cada empregado, en función do seu salario bruto, dependendo dos límites entre os que se atope. Esta táboa podería conter unha información similar a esta:

u5a1_taboairpf.emf

* + - 1. Base de datos traballadores

A base de datos *traballadores* serve para levar control dos empregados, departamentos e centros dunha empresa. Está formada por un grupo de táboas, relacionadas entre si, tal e como se mostra no seguinte grafo relacional e se describe a continuación. As táboas son *MyIsam* (non transaccionais) e por tanto non teñen definidas claves foráneas.

TRABALLADORES_GRAFO.emf

* Táboa centro

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome columna | Tipo | Null | Clave | Observacións |
| cenNumero | int | Non | Primaria | Número co que se identifica. |
| cenNome | char(30) |  | Índice | Nome. |
| cenEnderezo | char(30) |  |  | Enderezo. |

* Táboa empregado

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome columna | Tipo | Null | Clave | Observacións |
| empNumero | int | Non | Primaria | Número co que se identifica. |
| empDepartamento | int | Non | Índice | Número do departamento no que traballa. |
| empExtension | smallint | Non |  | Extensión telefónica para o empregado. Pode compartirse entre empregados de diferentes departamentos. |
| empDataNacemento | date |  |  | Data de nacemento. |
| empDataIngreso | date |  |  | Data de ingreso na empresa. |
| empSalario | decimal(6,2) |  |  | Salario mensual en euros. |
| empComision | decimal(6,2) |  |  | Comisión mensual. |
| empFillos | smallint |  |  | Número de fillos. |
| empNome | char(20) | Non | Índice | Nome do empregado coa forma: primeiro apelido, nome. |

* Táboa departamento

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome columna | Tipo | Null | Clave | Observacións |
| depNumero | int | Non | Primaria | Número co que se identifica. |
| depNome | char(20) |  | Índice | Nome. |
| depDirector | int | Non | Índice | Número do empregado director do departamento. |
| deptipoDirector | char(1) |  |  | Tipo de directo: P (en propiedade, é dicir, titular), F (en funcións). |
| depPresuposto | decimal(9,2) |  |  | Cantidade en euros de presuposto anual. |
| depDepende | int |  | Índice | Número do departamento do que depende. |
| depCentro | int |  | Índice | Número do centro ao que pertence. |
| depEmpregados | smallint unsigned |  |  | Número de empregados que traballan no departamento. |

* 1. Actividade
     1. As vistas

Unha vista é unha táboa virtual que non ten existencia física como unha táboa base, pero que os usuarios ven como unha táboa máis. Distintos usuarios poden ter unha percepción distinta do contido dunha mesma base de datos en función das vistas que utilizan para acceder aos datos.

As vistas teñen un nome asociado e a mesma estrutura cá unha táboa, composta de filas e columnas. A diferenza fundamental entre as vistas e as táboas base é que das vistas só se almacena a definición da súa estrutura, e os datos só se almacenan nas táboas base.

As vistas son como un filtro entre os usuarios e os datos almacenados nas táboas base. Este filtro está definido como unha selección de datos mediante unha sentenza SELECT sobre unha ou máis táboas, ou sobre outras vistas.

esquemavista.emf

Os permisos que teña o usuario limitan as operacións que pode facer nas bases de datos. Se un usuario pode consultar datos (SELECT), e realizar o resto de operacións de manipulación de datos (INSERT, UPDATE, DELETE), tamén terá permiso para facer as mesmas operacións utilizando vistas. Para crear ou modificar vistas, necesita ter permisos especiais tal e como se verá máis adiante.

A maioría dos SXBDR permiten a creación e manexo de vistas; MySQL, incorpora as vistas a partir da versión 5.0.1.

* + - 1. Sentenza CREATE VIEW e utilización de vistas

A sentenza CREATE VIEW permite crear novas vistas, ou substituír vistas que xa existen. Sintaxe:

CREATE [OR REPLACE]

[DEFINER = { usuario | CURRENT\_USER }]

[SQL SECURITY { DEFINER | INVOKER }]

VIEW nome\_vista [(lista\_columnas)]

AS sentenza\_select

[WITH [CASCADED | LOCAL] CHECK OPTION]

* A cláusula opcional OR REPLACE permite borrar unha vista que exista co mesmo nome e crear a nova vista. Se non se utiliza esta cláusula e xa existe unha vista co nome da que se vai a crear, prodúcese unha condición de erro e non se crea a nova vista.
* As cláusulas DEFINER e SQL SECURITY especifican o contexto de seguridade no momento da execución da vista. Con DEFINER pódese indicar o nome do usuario que vai ser considerado como o creador da vista. Se non se especifica nada, tómase CURRENT\_USER que fai referencia ao usuario actual que está creando a vista.
* A cláusula SQL\_SECURITY permite indicar se na execución da vista se utilizan privilexios do creador da vista (DEFINER - valor por defecto) ou do usuario que a chama (INVOKER).
* O nome da vista, *nome\_ vista,* ten que cumprir as mesmas condicións que calquera nome de táboa, en canto a carácteres e lonxitude permitidos. A continuación do nome da vista pódense poñer, pechados entre parénteses, os nomes que van a ter as columnas da nova vista separados por comas (*lista\_columnas*). Se non se especifican os nomes, tómanse os nomes das columnas da lista de selección da consulta empregada para crear a vista. O número de nomes da *lista\_columnas* ten que coincidir co número de columnas da lista de selección da consulta.
* *sentenza\_select* representa unha sentenza SELECT, que selecciona os datos de definición da vista. Os datos poden ser seleccionados de táboas base ou doutras vistas. A sentenza SELECT utilizada para crear unha vista ten unha serie de restricións:
* Non pode conter subsentenzas na cláusula FROM.
* Non pode conter referencias a variables do sistema ou de usuario.
* Non pode facer referencia a táboas de tipo temporal, e non se poden crear vistas temporais.
* A cláusula WITH CHECK OPTION utilízase para crear vistas actualizables, que poden ser utilizadas para inserir ou modificar datos das táboas. Cando se utiliza a cláusula, só permitirá inserir ou modificar filas que cumpran as condicións recollidas na cláusula WHERE da *sentenza\_select*. A cláusula WITH CHECK OPTION incorpórase na versión MySQL 5.0.2.
* As opcións LOCAL e CASCADED pódense utilizar cando na definición da vista se utilicen outras vistas. Permiten establecer a forma en que se comprobarán as condicións incluídas nas cláusulas WHERE das vistas. Cando se utiliza a palabra LOCAL, a cláusula CHECK OPTION só comproba as condicións da *sentenza\_select* da vista que se está creando. Cando se utiliza a palabra CASCADED, compróbanse as condicións de todas as vistas que interveñen na definición. Se non se especifica nada tómase, por defecto, CASCADED.

O usuario que cree as vistas ten que ter o privilexio CREATE VIEW e os privilexios adecuados sobre as columnas ás que se fai referencia na SELECT. No caso de utilizar a opción OR REPLACE, tamén é necesario que teña o privilexio DROP VIEW para borrar a vista.

As vistas utilízanse como se fosen táboas base xunto con SELECT, INSERT, DELETE OU UPDATE.

Exemplos

* Exemplo de creación dunha vista para manexar a información dos departamentos que pertencen á provincia de Lugo (*id\_provincia* = 27).

**create** **view** departamento\_lugo

**as**

**select** codigo**,** nome**,** tipo**,** cidade**,** id\_provincia

**from** departamento

**where** id\_provincia **=** 27**;**

* Exemplo de utilización da vista anterior para que os usuarios só poidan consultar os departamentos da súa provincia, e non as doutras provincias.

**select** **\*** **from** departamento\_lugo**;**

creacionvistas.emf

Cando se fai a consulta anterior a vista de exemplo como filtro, só se mostran os departamentos que teñan en *id\_provincia* o valor 27, e as columnas vense cos nomes que teñen na táboa *departamento*.

* Exemplo de creación dunha vista para inserir ou modificar a información dos departamentos que pertencen á provincia de Lugo.

No caso de querer utilizar a vista para actualizar datos sobre a táboa *departamento*, verificando a condición que a columna *id\_provincia* só pode tomar o valor 27, hai que engadirlle á vista a cláusula WITH CHECK OPTION. No código que ven a continuación incorpórase esa cláusula, e ademais, cámbiaselle o nome ás columnas poñendo a lista de columnas, pechadas entre parénteses a continuación do nome da vista, e utilízase a opción OR REPLACE, que fai que a vista creada antes co mesmo nome se borre, e créase unha vista coas novas especificacións.

**create** **or** **replace**

**view** departamento\_lugo **(**dl\_codigo**,** dl\_nome**,** dl\_tipo**,** dl\_cidade**,** dl\_provincia**)**

**as**

**select** codigo**,** nome**,** tipo**,** cidade**,** id\_provincia

**from** departamento

**where** id\_provincia **=** 27

**with** **check** **option;**

* Exemplo de utilización da vista anterior nunha consulta.

Mostra os mesmo datos que na consulta anterior, pero cambian os nomes das columnas.

**select** **\*** **from** departamento\_lugo**;**

creacionvistas2.emf

* Exemplo de utilización da vista anterior para unha inserción.

Ao levar a cláusula WITH CHECK OPTION, a vista anterior tamén pode ser utilizada para actualizar datos da táboa *departamento*. Cando se utiliza para inserir ou modificar filas, o sistema comproba que o valor almacenado na columna *dl\_provincia* tome o valor 27; se toma un valor distinto, non permite a operación e mostra unha mensaxe de erro informando que non se cumpre as condicións CHECK OPTION; se toma o valor 27, realiza a inserción.

* Exemplo de inserción de datos da provincia 15.

**insert** **into** departamento\_lugo

**values** **(**101**,**'proba local'**,**'H'**,**''**,**15**);**



* Exemplo de inserción de datos da provincia 27.

**insert** **into** departamento\_lugo

**values** **(**101**,**'proba local'**,**'H'**,**'proba'**,**27**);**



* Exemplo de utilización dunha vista creada con LOCAL en CHECK OPTION.

Crear unha vista para manexar a información dos departamentos que pertencen á provincia de Lugo (id\_provincia = 27), e que sexan de tipo 'H'.

Créase a vista sobre a vista *departamento\_lugo* que xa selecciona os departamentos de Lugo, e engádese a condición de que o contido da columna *dl\_tipo* tome o valor 'H'.

**create** **view** departamentoh\_lugo

**as**

**select** dl\_codigo**,** dl\_nome**,** dl\_tipo**,** dl\_cidade**,** dl\_provincia

**from** departamento\_lugo # créase a vista sobre unha vista que existe

**where** dl\_tipo **=** 'H'

**with** **local** **check** **option;**

* Cando se fai a consulta utilizando esta vista, móstranse os departamentos que sexan da provincia de Lugo (*dl\_provincia* = 27), e teñan na columna *dl\_tipo* o valor 'H'.

**select** **\*** **from** departamentoh\_lugo**;**

creacionvistas3.emf

* Cando se actualizan datos utilizando esta vista, o sistema só comproba que o valor que toma a columna dl\_tipo sexa 'H', pero non comproba a condición da vista *departamento\_lugo* (id\_provincia = 27) porque a cláusula WITH CHECK OPTION leva a opción LOCAL.

A sentenza INSERT seguinte intenta inserir unha fila na táboa departamento, utilizando como filtro a vista *departamentoh\_lugo*. O valor que toma a columna *dl\_tipo* non cumpre a condición LOCAL establecida na sentenza SELECT de creación da vista, polo que a fila non se insire e móstrase unha mensaxe de erro.

**insert** **into** departamentoh\_lugo

**values** **(**102**,**'proba local'**,**'A'**,**'proba'**,**15**);**



A sentenza INSERT seguinte execútase correctamente porque o valor da columna *dl\_tipo* cumpre a condición establecida na sentenza SELECT de creación da vista (*dl\_tipo* = 'H'), e non se ten en conta que non cumpra a condición establecida na sentenza SELECT de creación da vista *departamento\_lugo* (*id\_provincia* = 27), porque a cláusula WITH CHECK OPTION leva a opción LOCAL.

**insert** **into** departamentoh\_lugo

**values** **(**102**,**'proba local'**,**'H'**,**'proba'**,**15**);**



* Exemplo de utilización dunha vista creada con CASCADED en CHECK OPTION.

No código que ven a continuación vaise a substituír a vista *departamentoh\_lugo*, que utiliza a opción LOCAL na cláusula WITH CHECK OPTION, por outra vista co mesmo nome pero utilizando a opción CASCADED na cláusula WITH CHECK OPTION, para ver a diferenza de comportamento entre as dúas opcións.

**create** **or** **replace** **view** departamentoh\_lugo

**as**

**select** dl\_codigo**,** dl\_nome**,** dl\_tipo**,** dl\_cidade**,** dl\_provincia

**from** departamento\_lugo # créase a vista sobre unha vista que existe

**where** dl\_tipo **=** 'H'

**with** **cascaded** **check** **option;**

* Cando se fai a consulta utilizando esta vista, móstranse os departamentos que son da provincia de Lugo (*dl\_provincia* = 27), e teñen na columna *dl\_tipo* o valor 'H', igual que na vista anterior.

**select** **\*** **from** departamentoh\_lugo**;**

creacionvistas3.emf

* Cando se utiliza a vista para actualizar datos, é cando cambia o comportamento. Ao inserir ou modificar unha fila, o sistema comproba que o valor que toma a columna *dl\_tipo* sexa 'H', e ademais, comproba a condición da vista *departamento\_lugo* (id\_provincia = 27) porque a cláusula WITH CHECK OPTION leva a opción CASCADED.

A sentenza INSERT seguinte intenta inserir unha fila na táboa departamento, utilizando como filtro a vista *departamentoh\_lugo*. O valor que toma a columna *dl\_tipo* cumpre a condición establecida na sentenza SELECT de creación da vista, pero a columna *dl\_provincia* toma o valor 15 e non cumpre a condición establecida na sentenza SELECT de creación da vista *departamento\_lugo*, polo que a fila non se insire e móstrase unha mensaxe de erro.

**insert** **into** departamentoh\_lugo

**values** **(**103**,**'proba local'**,**'H'**,**'proba'**,**15**);**



A sentenza INSERT seguinte execútase correctamente porque cumpre as condicións contidas nas dúas vistas.

**insert** **into** departamentoh\_lugo

**values** **(**103**,**'proba local'**,**'H'**,**'proba'**,**27**);**



* Exemplo con limitación de columnas.

Crear unha vista que mostre o *dni*, *nome*, *apelidos*, e *departamento* de todos os empregados, ocultando os datos de salario bruto e xefe.

Nos exemplos feitos ata o momento faise unha limitación de filas da táboa *departamento*, introducindo unha cláusula WHERE na consulta de creación da vista que filtra as filas que cumpren a condición. Cando se quere facer unha limitación das columnas o único que hai que facer é poñen na lista de selección da consulta só as columnas ou expresións que se lle permiten ver aos usuarios que manexan a vista. Cando se executa unha sentenza INSERT sobre a vista, as columnas que non figuran na lista de selección toman o valor nulo (NULL).

**create** **view** vistaEmpregado

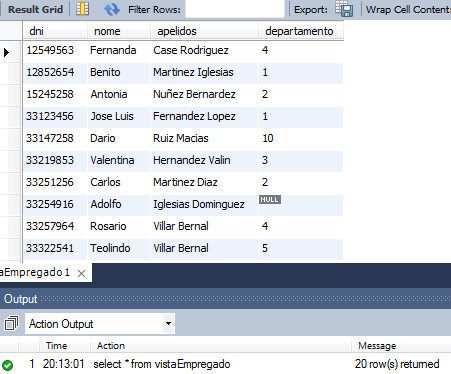
**as**

**select** dni**,** nome**,** apelidos**,** departamento

**from** empregado**;**

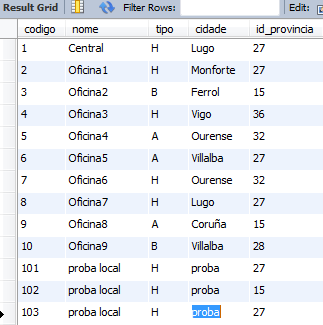
Esta vista permite ocultar a un grupo de usuarios a información sobre os salarios brutos dos seus compañeiros. Por exemplo, supoñamos que o xerente da empresa ten acceso á táboa *empregado*, pero o resto de traballadores non teñen permisos de acceso a esa táboa, pero se poden consultar datos utilizando a vista; cando executan unha consulta utilizando a vista, verán só as columnas contidas na lista de selección da consulta asociada á vista, pero eles poden ter a percepción de que son todos os datos que hai almacenados.

**select** **\*** **from** vistaEmpregado**;**



Comprobación do contido da táboa departamento despois das operacións feitas coas vistas.

**select** **\*** **from** departamento**;**



Unha vez finalizadas as probas de inserción e para non deixar os datos de proba na táboa, aconséllase borrar as filas que se inseriron de proba.

**delete** **from** departamento

**where** codigo **>=**100**;**

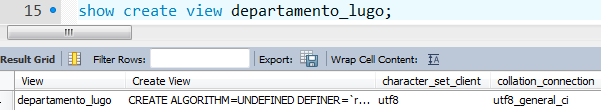
* + - 1. Sentenza SHOW CREATE VIEW

A información sobre as vistas creadas gárdase no dicionario de datos, igual có resto de obxectos das bases de datos. En MySQL, a información sobre as vistas pódese consultar en *information\_schema.views* mediante unha sentenza SELECT e dispón, ademais, da sentenza SHOW CREATE VIEW para consultar información sobre as vistas. Sintaxe:

SHOW CREATE VIEW nome\_vista

Exemplo: Mostrar información sobre a vista *departamento\_lugo*.

**show create view** departamento\_lugo



O resultado da sentenza anterior móstrase na zona *Result Grid* de Workbench. Móstranse catro columnas con información sobre o nome da vista, a sentenza de creación, o xogo de carácteres asociado e o sistema de colación. A información sobre a sentenza de creación non se pode ver porque é moi larga e non entra na pantalla. Para poder vela cun formato lexible pódense seguir estes pasos:

* Seleccionar o contido da columna *Create View* onde está a sentenza de creación.
* Facer clic co botón dereito do rato.
* Elixir a opción *Copy Field (unquoted).*
* Abrir unha nova pestana de consulta SQL e pegar nela o contido da columna.
* Pinchar na icona *Beautify/Reformat SQL script* e aparece o código na versión "bonita" de Workbench.

**CREATE**

ALGORITHM **=** UNDEFINED

DEFINER **=** `root`@`localhost`

**SQL** SECURITY DEFINER

**VIEW** `departamento\_lugo` **AS**

**select**

`departamento`**.**`codigo` **AS** `dl\_codigo`**,**

`departamento`**.**`nome` **AS** `dl\_nome`**,**

`departamento`**.**`tipo` **AS** `dl\_tipo`**,**

`departamento`**.**`cidade` **AS** `dl\_cidade`**,**

`departamento`**.**`id\_provincia` **AS** `dl\_provincia`

**from**

`departamento`

**where**

**(**`departamento`**.**`id\_provincia` **=** 27**)** **WITH** **CASCADED** **CHECK** **OPTION**

* + - 1. Sentenza ALTER VIEW

Permite cambiar a definición dunha vista que existe. A sintaxe e similar á da sentenza CREATE VIEW, e é o mesmo que utilizar a sentenza CREATE OR REPLACE VIEW. Sintaxe:

ALTER

[DEFINER = { usuario | CURRENT\_USER }]

[SQL SECURITY { DEFINER | INVOKER }]

VIEW nome\_vista [(lista\_columnas)]

AS sentenza\_select

[WITH [CASCADED | LOCAL] CHECK OPTION]

O usuario que modifique vistas ten que ter o privilexio CREATE VIEW, DROP VIEW e os privilexios adecuados sobre as columnas a que se fai referencia na SELECT.

* + - 1. Sentenza DROP VIEW

Para borrar vistas utilízase a sentenza DROP VIEW. Os usuarios que a utilicen teñen que ter o privilexio DROP VIEW. A sintaxe é a seguinte:

DROP VIEW [IF EXISTS] nome\_vista [, nome\_vista] ...

* A opción IF EXISTS permite evitar que se produza un erro cando a vista que se intenta borrar non existe. Ten utilidade cando a sentenza vai incluída nun script.
* Pódense borrar varias vista coa mesma sentenza poñendo os nomes separados por comas.
  + 1. Manipulación de datos a través das vistas

As operacións de manipulación de datos autorizadas sobre as vistas son as mesmas que sobre as táboas (consultas, inserción, modificación e supresión). Como se comentou anteriormente, todas as vistas poden ser utilizadas para consultar datos, pero non todas as vistas poden ser utilizadas para actualizar datos sobre as táboas.

Chámase vista 'actualizable' (*updatable*) aquela que pode ser utilizada para manipular os datos das táboas coas que está relacionada mediante INSERT, UPDATE ou DELETE.

O manual de referencia de MySQL expón detalladamente as condicións que ten que cumprir unha vista para que sexa 'actualizable'. Resumo desas condicións:

* Debe haber unha relación 'un a un' entre as filas da vista e as filas da táboa coa que está relacionada.
* Non contén:
* Funcións de agrupamento (AVG(), SUM(), COUNT(), ...).
* Cláusulas: DISTINCT, GROUP BY, HAVING, UNION ou UNION ALL.
* Subsentenzas na lista de selección.
* Certas combinacións con JOIN.
* Referencias a vistas non 'actualizables' na cláusula FROM.
* Para inserir filas utilizando unha vista, esta debe incorporar na lista de selección todas as columnas que non admiten valores nulos (propiedade NOT NULL), e non pode conter expresións; só nomes de columnas.

Cando se crea unha vista, o servidor engádelle un indicador que toma o valor verdadeiro (*true)* cando a vista é 'actualizable' e o valor falso (*false*) se non o é. Esta información pódese consultar na columna IS\_UPDATEBLE da táboa VIEW da base de datos INFORMATION\_SCHEMA, aínda que nalgúns casos pode ter o valor verdadeiro e non pode ser utilizada para calquera operación de actualización de datos. Isto significa que cando se executa unha sentenza INSERT, UPDATE ou DELETE sobre unha vista, o servidor sabe se están permitidas esas operacións sobre a vista, e en caso de que non estean mostra a mensaxe de erro correspondente.

Exemplos dalgunhas sentenzas SELECT que non se poderían utilizar para crear unha vista 'actualizable':

|  |  |
| --- | --- |
| **select** dni,apelidos**,**nome**,**salario\_bruto \*20/100  **from** empregado**;** | Porque contén unha expresión e non pode ser utilizada para inserir filas, aínda que podería ser utilizada para facer modificacións e borrado de filas |
| **select** id\_provincia**,** cidade **,** **count(\*)**  **from** departamento  **group** **by** id\_provincia**;** | Porque utiliza a cláusula GROUP BY, e ademais a función de agrupamento COUNT(\*) |
| **select** dni,apelidos**,** nome**,** salario\_bruto,  (**select avg()** **from** empregado) **as** media\_salario  **from** empregado  **where** salario\_bruto **>=** 50000**;** | Porque contén unha subconsulta na lista de selección. |
| **select** apelidos**,** nome**,** salario\_bruto  **from** empregado  **where** salario\_bruto **between** 50000 **and** 70000**;** | Non pode ser utilizada para inserir filas, porque non inclúe a columna dni que non admite valores null. |

* + 1. Avantaxes do uso de vistas

De forma resumida os beneficios máis importantes que aportan as vistas son os seguintes:

* Almacénanse no servidor, polo que o consumo de recursos e eficacia sempre serán óptimos. Cando se crea unha vista faise unha compilación da sentenza *SELECT* que contén, e gárdase asociada ao nome da vista. Desta maneira o tempo de execución dunha vista é menor que se se executa a sentenza *SELECT* directamente, xa que esta ten que pasar por o proceso de compilación.
* Unha vista é un camiño fácil para gardar consultas complexas na propia base de datos. Ocorre con frecuencia que os usuarios solicitan información que require realizar consultas complexas que poden levar moito tempo deseñar e probar, e que poden ser utilizadas noutro momento. Se estas consultas se fan con vistas, quedarán almacenadas no servidor e estarán accesibles cando se necesiten.

Tamén facilita o traballo aos desenvolvedores de software que teñan pouca experiencia traballando con bases de datos e dificultades para facer consultas complexas, dándolles a opción de chamar á vista almacenada na base de datos para poder obter os datos.

* Proporcionan flexibilidade na formulación de consultas moi complexas. Unha vista pode ser creada coa idea de facilitar a formulación de sentenzas complexas que non son soportadas nunha única sentenza SELECT.
* Son unha ferramenta importante para manter a confidencialidade dos datos.

Poden ser utilizadas como filtro entre os usuarios e as táboas base, de maneira que os usuarios non acceden directamente ás táboas e non teñen porque ver todos os datos da táboas se non que só ven os datos que foron seleccionados na consulta coa que se creou a vista.

Pódese limitar o acceso aos usuarios a determinadas filas ou columnas dunha ou máis táboas, sen ter que conceder privilexios aos usuarios sobre as táboas, sendo suficiente conceder privilexios sobre as vistas.

Aos desenvolvedores de software facilítalles o acceso ás vistas cos datos que necesitan, pero non se lles dá acceso as táboas.

* Axudan a manter a integridade referencial. Permiten controlar a integridade referencial para táboas non transaccionais nas operacións de inserción de filas utilizando a cláusula WITH CHECK OPTION.
* Axudan a manter a integridade de dominio. Unha vista pode restrinxir o acceso ás filas en función do contido dunha ou máis columnas cando inclúen condicións na cláusula WHERE na consulta utilizada na súa creación. Tamén, se pode utilizar para comprobar se os valores gardados nesas columnas cumpren as condicións establecidas, cando se insiren ou modifican filas. Por exemplo, podemos crear unha vista asociada á táboa clientes que comprobe que os datos almacenados na columna localidade pertenzan ao dominio {'Lugo','Sarria','Ourense','Ferrol','Vigo'}.

* Tarefa 1. Crear e utilizar vistas.
  1. Tarefas

A realización da tarefa proposta farase nun servidor que non ten habilitado o modo SQL estrito. A tarefa proposta é:

* Tarefa 1. Crear e utilizar vistas.
  + 1. Tarefa 1. Crear e utilizar vistas

A tarefa consiste en crear e utilizar vistas utilizando a LMD.

Sobre a base de datos traballadores

As táboas desta base de datos son *MyIsam* (non transaccionais) e por tanto non se comproba a integridade referencial.

* Tarefa 1.1. Crear unha vista sen CHECK OPTION para os empregados da empresa cos datos: número de empregado, nome, extensión telefónica, número e nome do departamento no que traballa. Utilizar a vista para:
* Inserir os datos dun empregado no departamento 110.
* Inserir os datos dun empregado no departamento 5.
* Inserir os datos dun empregado co nome NULL.
* Inserir os datos dun empregado sen nome.
* Consultar os datos ordenados alfabeticamente polo nome do empregado.
* Tarefa 1.2. Crear unha vista con CHECK OPTION asociada á táboa *departamentos*, que realiza as seguintes comprobacións:
* Existe o empregado que é director.
* Existe o centro do que depende.
* O presuposto está entre 100000 e 2000000 euros.
* O tipo de director só admite os valores P ou F.

Utilizar a vista para consultar os datos e comprobar se é posible utilizala con INSERT, DELETE ou UPDATE.

* Tarefa 1.3. Crear unha vista con CHECK OPTION que permita ver os datos dos empregados que traballan no departamento 111. Utilizar a vista para:
* Modificar o número de fillos do empregado 240 que existe na táboa creada pola vista, poñendo 5 fillos.
* Modificar o departamento do empregado 240 que existe na táboa creada pola vista, poñendo o departamento 5.
* Modificar o número de fillos do empregado 900 que non existe na táboa creada pola vista nin na táboa *empregado*, poñendo 5 fillos.
* Tarefa 1.4. Crear unha vista sen CHECK OPTION que permita mostrar o número, nome e departamento no que traballan os empregados que levan máis de 20 anos traballando na empresa e teñen máis de 2 fillos. Utilizar a vista para:
* Eliminar os datos do empregado 120 que existe na táboa *empregado* pero non existe na táboa creada pola vista.
* Eliminar os datos do empregado 900 que non existe na táboa *empregado* nin tampouco na táboa creada pola vista.
* Eliminar os datos do empregado 260 que existe na táboa creada pola vista.

Sobre a base de datos tendasBD

As táboas desta base de datos son *InnoDB* (transaccionais) e por tanto compróbase a integridade referencial.

* Tarefa 1.5. Crear unha vista sen CHECK OPTION cos seguintes datos, referidos aos artigos: código e nome do artigo e nome de provedor que o subministra.Utilizar a vista para:
* Inserir os datos do artigo '4065091' que existe na táboa artigos, subministrado polo provedor 1 que existe na táboa de provedores.
* Inserir os datos do artigo '0001122' que non existe na táboa artigos, subministrado polo provedor 100 que non existe na táboa de provedores.
* Inserir os datos do artigo '0001122' que non existe na táboa artigos, subministrado polo provedor 1 que existe na táboa de provedores.
* Consultar os datos ordenados alfabeticamente polo nome do artigo.
* Tarefa 1.6. Crear unha vista sen CHECK OPTION asociada á táboa *detalle\_vendas* que mostra código de artigo e número total de unidades vendidas ordenados por código.

Utilizar a vista para consultar os datos e comprobar se é posible utilizala con INSERT, DELETE ou UPDATE.

* Tarefa 1.7. Crear unha vista con CHECK OPTION que permita ver os datos das vendas realizadas na tenda 30. Utilizar a vista para:
* Modificar o cliente da venda 98 que existe na táboa creada pola vista poñendo o cliente 101 que non existe na táboa de clientes.
* Utilizar a vista para modificar o cliente da venda 200 que non existe na táboa creada pola vista poñendo o cliente 100 que existe na táboa de clientes.
* Modificar o cliente da venda 98 que existe na táboa creada pola vista poñendo o cliente 100 que existe na táboa de clientes.
* Tarefa 1.8. Crear unha vista sen CHECK OPTION que permita mostrar código, nome, stock, data de alta e provedor dos artigos dados de alta antes do 1-1-2010, ordenados por provedor, stock (de forma descendente) e código. Utilizar a vista para:
* Eliminar o artigo de código '00112233' que non existe na táboa que crea a vista.
* Eliminar o artigo de código '00112233' que non existe na táboa que crea a vista e está rexistrada nalgunha venda.

Solución

* Tarefa 1.1

/\*Crear unha vista sen CHECK OPTION cos seguintes datos, referidos aos empregados da

empresa: número de empregado, nome, extensión telefónica, número e nome do departamento

no que traballa \*/

**create** **or** **replace** **view** tarefa11\_empregados **as**

**select** em**.**empNumero**,**

em**.**empNome**,**

em**.**empExtension**,**

em**.**empDepartamento**,**

de**.**depNome

**from** empregado **as** em **join** departamento **as** de **on** **(**em**.**empDepartamento **=** de**.**depNumero**);**

/\* Utilizar a vista para inserir os datos dun empregado no departamento 110 que existe

na táboa departamento \*/

**insert** **into** tarefa11\_empregados **(**empNome**,** empExtension**,** empDepartamento**)**

**values** **(**'proba1'**,**'500'**,**'110'**);**

A fila insírese colocando null, o valor por defecto ou o valor autoincrementado que corresponda ás columnas ás que non se adxudicou valor.

/\* Utilizar a vista para inserir os datos dun empregado no departamento que non existe

na táboa departamento\*/

**insert** **into** tarefa11\_empregados **(**empNome**,** empExtension**,** empDepartamento**)**

**values** **(**'proba2'**,**'500'**,**'5'**);**

Coidado con esta situación. A fila insírese colocando null, o valor por defecto ou o valor autoincrementado que corresponda para as columnas ás que non se adxudicou valor. Incluso coloca 5 no departamento aínda que non exista na táboa *departamento* xa que as táboas son non transacionais e por tanto non se comproba a integridade referencial.

/\* Utilizar a vista para inserir os datos dun empregado co nome NULL \*/

**insert** **into** tarefa11\_empregados **(**empNome**,** empExtension**,** empDepartamento**)**

**values** **(null,**'500'**,**'110'**);**

Non se insire a fila e móstrase unha mensaxe de erro xa que existe un valor null para unha columna que non o admite.

/\* Utilizar a vista para inserir os datos dun empregado sen nome \*/

**insert** **into** tarefa11\_empregados **(**empExtension**,** empDepartamento**)**

**values** **(**'500'**,**'110'**);**

Coidado con esta situación: a fila insírese aínda que se mostra unha advertencia avisando de que o campo *empNome* non ten definido un valor por defecto e por tanto tomará o valor por defecto implícito para as columnas de tipo cadea de carácteres que é unha cadea baleira ('').

/\* Utilizar a vista para consultar os datos ordenados alfabeticamente

polo nome do empregado\*/

**select** **\*** **from** tarefa11\_empregados

**order** **by** empNome**;**

* Tarefa 1.2

/\* Crear unha vista con CHECK OPTION asociada á táboa departamentos, que realiza as

seguintes comprobacións:

– Existe o empregado que é director.

– Existe o centro do que depende.

– O presuposto está entre 100000 e 2000000 euros.

– O tipo de director só admite os valores P ou F.

\*/

**create** **or** **replace** **view** tarefa12\_departamentos **as**

**select** **\***

**from** departamento

**where** depDirector **in** **(select** empNumero **from** empregado**)**

**and** depCentro **in** **(select** cenNumero **from** centro**)**

**and** depPresuposto **between** 100000 **and** 2000000

**and** deptipoDirector **in** **(**'P'**,**'F'**)**

**with** **check** **option;**

/\* Utilizar a vista para consultar os datos \*/

**select** **\*** **from** tarefa12\_departamentos**;**

/\* Comprobar se é posible utilizar a vista con INSERT, DELETE ou UPDATE \*/

**select** **\*** **from** information\_schema**.**VIEWS**;**

A columna IS\_UPDATABLE ten o valor YES , polo que é posible utilizar a vista con INSERT, DELETE e UPDATE.

* Tarefa 1.3

/\*Crear unha vista con CHECK OPTION que permita ver os datos dos empregados que

traballan no departamento 111 \*/

**create** **or** **replace** **view** tarefa13\_empregados **as**

**select** **\***

**from** empregado

**where** empDepartamento **=** 111

**with** **check** **option;**

/\* Utilizar a vista para modificar o número de fillos do empregado 240 que existe na

táboa creada pola vista, poñendo 5 fillos\*/

**update** tarefa13\_empregados **set** empfillos**=**5

**where** empNumero**=**240**;**

Realízase a modificación xa que o empregado existe na táboa que crea a vista.

/\* Utilizar a vista para modificar o departamento do empregado 240 que existe na

táboa creada pola vista, poñendo o departamento 5.\*/

**update** tarefa13\_empregados **set** empDepartamento**=**5

**where** empNumero**=**240**;**

Provoca un erro. Non se realiza a modificación porque non se cumpre a condición WHERE da vista.

/\* Modificar o número de fillos do empregado 900 que non existe na táboa creada pola

vista nin na táboa empregado, poñendo 5 fillos.\*/

**update** tarefa13\_empregados **set** empfillos**=**5

**where** empNumero**=**900**;**

Non provoca erro pero non afecta a ningunha fila.

/\* Utilizar a vista para consultar \*/

**select** **\*** **from** tarefa13\_empregados**;**

* Tarefa 1.4

/\* Crear unha vista sen CHECK OPTION que permita mostrar o número, nome e departamento

no que traballan os empregados que levan máis de 20 anos traballando na empresa e

teñen máis de 2 fillos\*/

**create** **or** **replace** **view** tarefa14\_empregados **as**

**select** empNumero**,** empNome**,** empDepartamento**,** empDataIngreso**,** empFillos

**from** empregado

**where** empFillos **>** 2

**and** timestampdiff**(year,** empDataIngreso**,** curdate**())** **>**20**;**

/\*Utilizar a vista para eliminar os datos do empregado 120 que existe na táboa

empregado pero non existe na táboa creada pola vista. \*/

**delete** **from** tarefa14\_empregados **where** empNumero**=**120**;**

Non provoca erro pero non afecta a ningunha fila.

/\*Utilizar a vista para eliminar os datos do empregado 900 que non existe na táboa

empregado nin tampouco na táboa creada pola vista. \*/

**delete** **from** tarefa14\_empregados **where** empNumero**=**900**;**

Non provoca erro pero non afecta a ningunha fila.

/\*Utilizar a vista para eliminar os datos do empregado 260 que existe na táboa creada

pola vista.\*/

**delete** **from** tarefa14\_empregados **where** empNumero**=**260**;**

Realízase o borrado xa que o empregado existe e cumpre o filtro da vista.

**select** **\*** **from** tarefa4\_empregados**;**

* Tarefa 1.5

/\*Crear unha vista sen CHECK OPTION cos seguintes datos, referidos aos artigos:

código e nome do artigo e nome de provedor que o subministra. \*/

**create** **or** **replace** **view** tarefa15\_artigos **as**

**select** ar**.**art\_codigo**,**

ar**.**art\_nome**,**

ar**.**art\_provedor

**from** artigos **as** ar **join** provedores **as** pr **on** **(**ar**.**art\_provedor **=** pr**.**prv\_id**);**

/\* Utilizar a vista para inserir os datos do artigo '4065091' que existe na táboa

artigos, subministrado polo provedor 1 que existe na táboa de provedores. \*/

**insert** **into** tarefa15\_artigos **(**art\_codigo**,** art\_nome**,** art\_provedor**)**

**values** **(**'4065091'**,**'artigo de proba 1'**,**1**);**

Non se realiza a inserción porque se detecta que o artigo xa existe.

/\* Utilizar a vista para inserir os datos do artigo '0001122' que non existe na táboa

artigos, subministrado polo provedor 100 que non existe na táboa de provedores. \*/

**insert** **into** tarefa15\_artigos **(**art\_codigo**,** art\_nome**,** art\_provedor**)**

**values** **(**'0001122'**,**'artigo de proba 1'**,**100**);**

Non se realiza a inserción porque se detecta que o provedor non existe na táboa de provedores.

/\* Utilizar a vista para inserir os datos do artigo '0001122' que non existe na táboa

artigos, subministrado polo provedor 1 que existe na táboa de provedores. \*/

**insert** **into** tarefa15\_artigos **(**art\_codigo**,** art\_nome**,** art\_provedor**)**

**values** **(**'0001122'**,**'artigo de proba 1'**,**1**);**

Realízase a inserción pero fai unha advertencia xa que hai columnas que non teñen valor por defecto na táboa *artigos* e esta inserción non contempla valor para tódalas columnas.

/\* Utilizar a vista para consultar os datos ordenados alfabeticamente

polo nome do artigo.\*/

**select** **\*** **from** tarefa15\_artigos

**order** **by** art\_nome**;**

* Tarefa 1.6

/\* Crear unha vista sen CHECK OPTION asociada á táboa detalle\_vendas que mostra

código de artigo e número total de unidades vendidas ordenados por código.

\*/

**create** **or** **replace** **view** tarefa16\_artigos **as**

**select** dv**.**dev\_artigo**,** **count(\*)**

**from** artigos **as** ar **join** detalle\_vendas **as** dv **on** dv**.**dev\_artigo **=** ar**.**art\_codigo

**group** **by** dv**.**dev\_artigo**;**

/\* Utilizar a vista para consultar os datos \*/

**select** **\*** **from** tarefa16\_artigos**;**

/\* Comprobar se é posible utilizar a vista con INSERT, DELETE ou UPDATE \*/

**select** **\*** **from** information\_schema**.**VIEWS**;**

Na zona de resultados de Workbench, na columna IS\_UPDATABLES aparece que esta vista non é UPDATABLE e por tanto non se pode utilizar con INSERT, DELETE e UPDATE.

* Tarefa 1.7

/\*Crear unha vista con CHECK OPTION que permita ver os datos das vendas realizadas

na tenda 30 \*/

**create** **or** **replace** **view** tarefa17\_vendas **as**

**select** **\***

**from** vendas

**where** ven\_tenda **=** 30

**with** **check** **option;**

/\* Utilizar a vista para modificar o cliente da venda 98 que existe na táboa creada

pola vista poñendo o cliente 101 que non existe na táboa de clientes.\*/

**update** tarefa17\_vendas **set** ven\_cliente **=** 101

**where** ven\_id**=**98**;**

Provoca un erro por non existir o cliente 101.

/\* Utilizar a vista para modificar o cliente da venda 200 que non existe na táboa

creada pola vista poñendo o cliente 100 que existe na táboa de clientes.\*/

**update** tarefa17\_vendas **set** ven\_cliente **=** 100

**where** ven\_id**=**200**;**

Non se fai ningunha modificación, é dicir, a actualización non afecta a ningunha fila.

/\* Utilizar a vista para modificar o cliente da venda 98 que existe na táboa creada

pola vista poñendo o cliente 100 que existe na táboa de clientes.\*/

**update** tarefa17\_vendas **set** ven\_cliente **=** 100

**where** ven\_id**=**98**;**

Realízanse os cambios na fila correspondente.

* Tarefa 1.8

/\* Crear unha vista sen CHECK OPTION que permita mostrar código, nome, stock,

fecha de alta e provedor dos artigos dados de alta antes do 1-1-2010 ou que non

teñen fecha de alta,ordenados por provedor, stock (de forma descendente) e código\*/

**create** **or** **replace** **view** tarefa18\_artigos **as**

**select** art\_codigo**,**

art\_nome**,**

art\_stock**,**

art\_alta**,**

art\_provedor

**from** artigos

**where** art\_alta**<**'2010-01-01'**or** art\_alta **is null**

**order** **by** art\_provedor**,** art\_stock **desc,** art\_codigo**;**

/\*Utilizar a vista para eliminar o artigo de código '00112233' que non

existe na táboa que crea a vista. \*/

**delete** **from** tarefa18\_artigos **where** art\_codigo**=** '00112233'**;**

Non se provoca ningún erro pero non fai ningún borrado.

/\*Utilizar a vista para eliminar o artigo de código '07735Y4'que existe na táboa

que crea a vista e está rexistrada nalgunha venda.\*/

**delete** **from** tarefa18\_artigos **where** art\_codigo**=** '07735Y4'**;**

Provoca un erro e non permite o borrado e así presérvase a integridade referencial.

/\*Utilizar a vista para eliminar o artigo de código '0001122'que existe na táboa

que crea a vista e non ten rexistrada ningunha venda.\*/

**delete** **from** tarefa18\_artigos **where** art\_codigo**=** '0001122'**;**

Bórrase o artigo.

1. Materiais
   1. Documentos de apoio ou referencia

* HUESO IBAÑEZ, Luis. *Bases de datos*. Ciclos Formativos Ra-Ma, 2012.
* SANCHEZ, Jorge. *Manual de Gestión e bases de datos*.

<http://www.jorgesanchez.net/bd/sgbd.html>

* 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>

* *Manual de referencia de MySQL 5.6.* <http://dev.mysql.com/doc/>
  1. Recursos didácticos
* Ordenadores con conexión a Internet, que terán instalado o sistema xestor de bases de datos MySQL e o cliente MySQL Workbench.
* Proxector.
* Material didáctico subministrado polo profesorado en papel e/ou formato dixital.
* Máquina virtual para exame que terá instalado o software estritamente necesario para a realización da tarefa de avaliación mediante as probas escritas prácticas.
* Manual de referencia de MySQL.
  1. Material auxiliar

O material auxiliar anexo a esta actividade está almacenado na carpeta CSIFC02\_MP0484\_V000602\_UD06\_A02\_Vistas\_Anexos que contén:

* O arquivo GuiaWorkbench.docx que é unha guía básica de MySQL Workbench 6.3.
* O arquivo V000602\_scriptsSQL.zip cos scripts necesarios para crear as bases de datos empregadas nos exemplos, tarefas e proba de avaliación da actividade.

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 |
| * CA5.10 - Creáronse vistas.   CA5.10.1 - Creáronse e utilizáronse vistas. | * Exame en papel e en formato dixital. | * **TO.1** - Documento de rexistro de creación e utilización de vistas. Escala de valores (observación indirecta). | 20 |

Modelo de proba para TO.1

Proponse unha proba práctica para avaliar CA5.10.1 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:
* Bloque de sentenzas solicitadas.
* Imaxes solicitadas e capturadas dende o cliente Workbench.
* O arquivo dixital debe conter:
* Arquivos .sql co código correspondente ao bloque de sentenzas.

A parte práctica realizarase na base de datos *equiposMyISAM* e *equiposInnoDB.* Subminístranse os arquivos *equiposMyISAM.sql* e *equiposInnoDB.sql* cos scripts que permiten crear as bases de datos. As dúas bases están formadas polas táboas que se describen a continuación e a relación entre táboas móstrase no diagrama E/R deseñado con Workbench que aparece despois da descrición. A diferenza entre as dúas bases de datos é que tal e como vai implícito no nome, as táboas dunha son MyISAM e as da outra son InnoDB.

* Táboa *equipo*

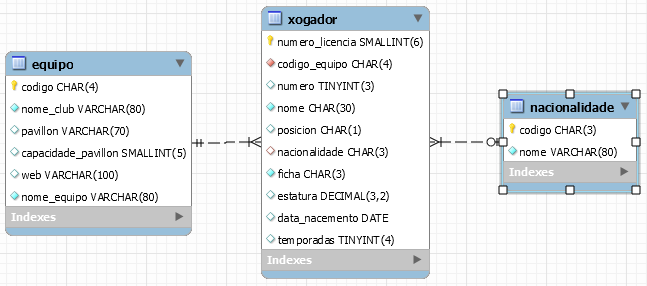
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome columna | Tipo | Null | Clave | Observacións |
| codigo | char(4) | non | Primaria | Código do equipo. Para enlazar con outras táboas |
| nome\_club | varchar(80) | non |  | Nome do club ao que pertence o equipo |
| pavillon | varchar(70) |  |  | Nome do pavillón no que xogan os partidos como local |
| capacidade\_pavillon | smallint unsigned |  |  | Número de espectadores que entran no seu pavillón |
| web | varchar(100) |  |  | Url da web |
| nome\_equipo | varchar(80) | non | Única | Nome oficial do equipo na liga |

* Táboa *xogador*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome columna | Tipo | Null | Clave | Observacións |
| numero\_licencia | smallint | non | Primaria | Código do equipo no que xoga o xogador. Autoincrementado |
| codigo\_equipo | char(4) | non | Foránea | Código do equipo no que xoga o xogador |
| numero | tinyint unsigned |  |  | Número de dorsal do xogador |
| nome | char(30) | non |  | Nome do xogador |
| posicion | char(1) |  |  | Posición na que xoga: B = base, P = pivot, A = aleiro, ...... |
| nacionalidade | char(3) | non | Foránea | Nacionalidade do xogador |
| ficha | char(3) | non |  | Tipo de ficha: EXT=estranxeiro EUR=europeo JFL=nacional |
| estatura | Decimal(3,2) unsigned |  |  | Estatura do xogador en metros |
| data\_nacemento | date |  |  | Data de nacemento |
| temporadas | tinyint unsigned |  |  | Número de tempadas que leva no equipo |

* Táboa *nacionalidade*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome columna | Tipo | Null | Clave | Observacións |
| codigo | char(3) | non | Primaria | Código do país |
| nome | varchar(80) | non |  | Nome do país |



Proba para CA5.10.1 sobre creación e utilización de vistas

Escribir as sentenzas SQL que permitan realizar as seguintes operacións de creación e utilización de vistas e capturar as imaxes que identifiquen os resultados pedidos dende o cliente Workbench. Se o resultado é moi longo, débese mostrar un anaco do resultado e capturar outra imaxe que o complemente.

* 1.- Vista *vistaequipoInnoDB* en *equiposInnoDB* para manexar os datos da táboa *equipo* coas seguintes columnas: código, nome e web do equipo.
* Escribir o código da vista.
* Escribir e executar unha sentenza SELECT que a utilice.
* Xustificar se pode ser utilizada para editar datos coas instrucións INSERT, UPDATE e DELETE.
* 2.- Vista *vistaxogadorInnoDB* en *equiposInnoDB* para manexar os datos da táboa *xogador*, que conteña tódalas columnas e contemple as seguintes restricións: a columna *ficha* so admita JFL, EUR ou EXT, e a columna *posicion* só admita B, E, A, P, F ou NULL.
* Escribir o código da vista.
* ¿As seguintes filas poderían inserirse utilizando a vista?. Facer e xustificar os cambios que se estimen necesarios nos datos para que a sentenza INSERT non produza erros.

'ALI',55,'Pérez, Fernando','A','ESP','JFL',1.92,'1989-02-12',1

'ALI',59,'Pérez, Jesús,'A','ESP','XXX',1.92,'1989-02-12',1

* 3.- Vista *vistaInnoDB* en *equiposInnoDB* que permita ver os nomes dos equipos e o número de xogadores que teñen en plantilla.
* Escribir o código da vista.
* Escribir e executar unha sentenza SELECT que a utilice.
* Xustificar se pode ser utilizada para editar datos coas instrucións INSERT, UPDATE e DELETE.
* 4.- ¿A vista *vistaxogadorInnoDB* poderíase crear na base de datos *equiposMyISAM* e utilizar do mesmo modo coas mesmas instrucións?. Escribir o novo código no caso de ser necesario.

Exemplo de solución para entregar en papel

Proponse a seguinte solución:

* 1.- Vista *vistaequipoInnoDB* en *equiposInnoDB* para manexar os datos da táboa *equipo* coas seguintes columnas: código, nome e web do equipo.
* Escribir o código da vista.

/\*Vista vistaequipoInnoDB en equiposInnoDB para manexar os datos da táboa equipo coas

seguintes columnas: código, nome e web do equipo\*/

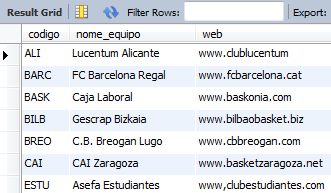
**create** **view** vistaequipoInnoDB **as**

**select** codigo**,** nome\_equipo**,** web

**from** equipo**;**

* Escribir e executar unha sentenza SELECT que a utilice.

**select** **\*** **from** vistaequipoInnoDB**;**





* Xustificar se pode ser utilizada para editar datos coas instrucións INSERT, UPDATE e DELETE.

Non pode ser utilizada para inserir nin actualizar porque non contén a columna *nome\_club* que por definición da columna non admite valores nulos.

* 2.- Vista *vistaxogadorInnoDB* en *equiposInnoDB* para manexar os datos da táboa *xogador*, que conteña tódalas columnas e contemple as seguintes restricións: a columna *ficha* so admita JFL, EUR ou EXT, e a columna *posicion* só admita B, E, A, P, F ou NULL.
* Escribir o código da vista.

/\*Vista vistaxogadorInnoDB en equiposInnoDB para manexar os datos da táboa xogador,

que conteña tódalas columnas, e contemple as seguintes restricións:

– A columna ficha so admite os seguintes valores: JFL, EUR, EXT.

– A columna posicion só admite os seguintes valores: B, E, A, P, F ou NULL.

\*/

**create** **view** vistaxogadorInnoDB **as**

**select** **\***

**from** xogador

**where** posicion **in** **(**'B'**,**'E'**,**'A'**,**'P'**,**'F'**,** **null)** **and** ficha **in** **(**'JFL'**,**'EUR'**,**'EXT'**)**

**with** **check** **option;**

* ¿As seguintes filas poderían inserirse utilizando a vista?. Facer e xustificar os cambios que se estimen necesarios nos datos para que a sentenza INSERT non produza erros.

'ALI',55,'Pérez, Fernando','A','ESP','JFL',1.92,'1989-02-12',1

'ALI',59,'Pérez, Jesús,'A','ESP','XXX',1.92,'1989-02-12',1

A segunda fila non podería inserirse coa vista por non cumprir a segunda condición da mesma. Debería de cambiarse a posición 'XXX' por outra válida como 'EUR'.

**insert** **into** vistaxogadorInnoDB **(**codigo\_equipo**,** numero**,** nome**,** posicion**,** nacionalidade**,** ficha**,** estatura**,** data\_nacemento**,** temporadas**)**

**values**

**(**'ALI'**,**55**,**'Pérez, Fernando'**,**'A'**,**'ESP'**,**'JFL'**,**1.92**,**'1989-02-12'**,**1**),**

**(**'ALI'**,**59**,**'Pérez, Jesús'**,**'A'**,**'ESP'**,**'EUR'**,**1.92**,**'1989-02-12'**,**1**);**

* 3.- Vista *vistaInnoDB* en *equiposInnoDB* que permita ver os nomes dos equipos e o número de xogadores que teñen en plantilla.
* Escribir o código da vista.

/\*vista vistaInnoDB que permita ver os nomes dos equipos e o número de xogadores

que teñen en plantilla\*/

**create** **view** vistaInnoDB **as**

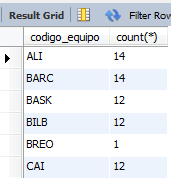
**select** codigo\_equipo**,** **count(\*)**

**from** xogador

**group** **by** codigo\_equipo**;**

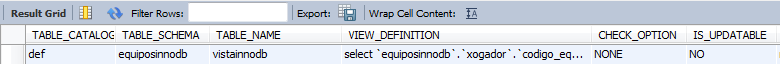
* Escribir e executar unha sentenza SELECT que a utilice.

**select** **\*** **from** vistaInnoDB**;**





* Xustificar se pode ser utilizada para editar datos coas instrucións INSERT, UPDATE e DELETE.



Non pode ser utilizada para edición porque contén unha columna cunha función de columna.

* ¿A vista *vistaxogadorInnoDB* poderíase crear na base de datos *equiposMyISAM* e utilizar do mesmo modo coas mesmas instrucións?. Escribir o novo código no caso de ser necesario.

Non valería o mesmo código xa que por tratarse de táboas MyISAM é necesario engadir as dúas seguintes restricións:

* A columna *codigo\_equipo* ten que coincidir con algún código da táboa equipo.
* A columna *nacionalidade* ten que coincidir con algún código da táboa nacionalidade.

O novo código sería:

/\*Vista vistaxogadorMyISAM en equiposMyISAM para manexar os datos da táboa xogador, que conteña tódalas columnas, e contemple as seguintes restricións:

– A columna codigo\_equipo ten que coincidir con algún código da táboa equipo

– A columna nacionalidade ten que coincidir con algún código da táboa nacionalidade

– A columna ficha so admite os seguintes valores: JFL, EUR, EXT, COT

– A columna posicion só admite os seguintes valores: B, E, A, P, F

\*/

**create** **view** vistaxogadorMyISAM **as**

**select** **\***

**from** xogador

**where** codigo\_equipo **in** **(select** codigo **from** equipo**)**

**and** nacionalidade **in** **(select** codigo **from** nacionalidade**)**

**and** posicion **in** **(**'B'**,**'E'**,**'A'**,**'P'**,**'F'**,** **null)**

**and** ficha **in** **(**'JFL'**,**'EUR'**,**'EXT'**)**

**with** **check** **option;**

Exemplo de lista de valoración para TO.1

Proponse a seguinte lista de valoración para o instrumento de avaliación TO.1 (EV) (observación indirecta: documento de rexistro de creación e utilización de vistas).

| Nome | Data | |
| --- | --- | --- |
| Indicadores para  CA5.10.1 - Creáronse e utilizáronse vistas. | Valoración máxima  20 | Cualificación |
| As sentenzas de creación de vistas non teñen erros sintácticos e cumpren as condicións pedidas. | 12 |  |
| As sentenzas de utilización das vistas funcionan correctamente. | 3 |  |
| As xustificacións requiridas da utilización das vistas son acertadas.. | 4 |  |
| As imaxes capturadas dende o cliente Workbench reflicten claramente o resultado da execución das sentenzas contidas nos arquivos .sql correspondentes. | 1 |  |

1. Máis información sobre a norma ISO 3166-1:2013 en <https://es.wikipedia.org/wiki/ISO_3166-1> [↑](#footnote-ref-3)