

Diseño Físico. Verificación e pruebas de datos

1.	Deseño Físico. Verificación e probas de datos	3
1.1	Bases de datos de traballo	3
1.1.1	Base de datos eleccionModulos.....	3
1.1.2	Base de datos gabinete.....	4
1.2	Verificación e probas de BD.....	5
1.2.1	Modo do servidor MySQL.....	6
1.2.2	Edición de datos con MySQL Workbench.....	7
1.2.3	Integridade e restricións.....	8
1.2.3.1	Verificar restrición PRIMARY KEY	9
1.2.3.2	Verificar restrición de integridade referencial	10
1.2.3.3	Verificar restrición UNIQUE	13
1.2.3.4	Verificar restrición DEFAULT	14
1.2.3.5	Verificar restrición CHECK	14
1.2.3.6	Verificar restrición implícitas nos tipos ENUM, e SET	14
1.2.3.7	Verificar restricións asociadas ao valor NULL	15
1.2.3.8	Verificar restricións asociadas ao tipo de dato e tamaño da columna.....	16
1.2.4	Resumo de mensaxes de erro asociados á carga de datos en MySQL.....	17

1. Deseño Físico. Verificación e probas de datos

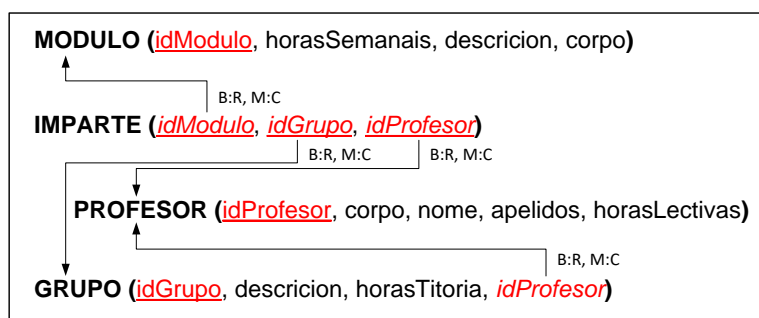
1.1 Bases de datos de traballo

Os exemplos deste tema empregan as bases de datos *eleccionModulos* e *gabinete*. Para poder probar os exemplos que se presentan no documento, débense executar os scripts de creación de base de datos no servidor e poñer en uso a base de datos que corresponda en cada caso.

1.1.1 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.

A descrición das táboas e das relacións que hai entre elas é como sigue:



■ 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	Índice	
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*

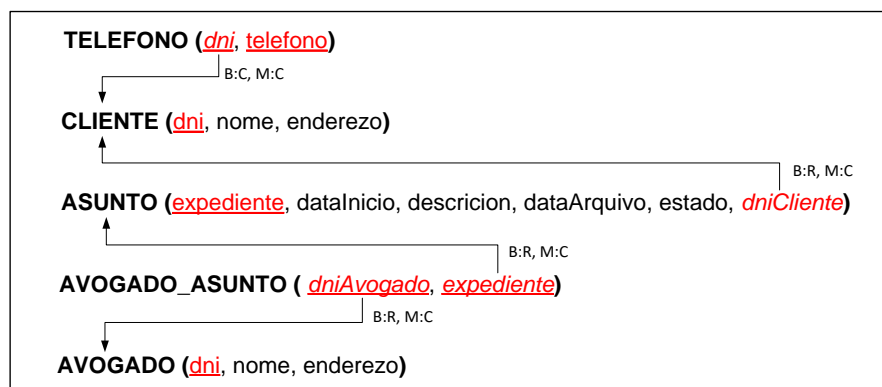
Nome	Tipo	Null	Clave	Observacións
idModulo	char(6)	Non	Primary	Clave para identificar o módulo
horasSemanais	tinyint (sen signo)			Descrición do nome do grupo
descripcion	varchar(100)			Nº de horas de redución para o profesor tutor
corpo	enum	Non		Corpo dos profesores que poden impartir o módulo. Valores 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

1.1.2 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:



▪ Táboa *cliente*

Nome	Tipo	Null	Clave	Observacións
dni	char(9)	Non	Primary	
nome	varchar(100)	Non	Índice	
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
descripcion	text	Non		Nº de horas de redución para o profesor tutor
dataArquivo	date			Identificador do profesor que é tutor do grupo
estado	enum	Non		Valores permitidos: 'recopilación','trámite','xuízo','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		
endereço	varchar(150)	Non		

▪ Táboa *avogado_asunto* (clave primaria composta)

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

1.2 Verificación e probas de BD

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 é 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, 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.

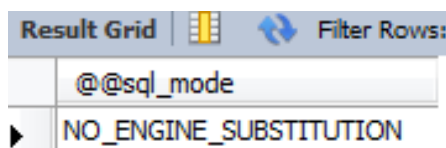
A continuación veremos as probas de comprobación da **integridade dos datos**.

1.2.1 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.

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 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. Pódese facer unha consulta á lista de modos no apartado *SQL Server Modes* do manual de referencia de MySQL. 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 transaccional, 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 **STRICT_TRANS_TABLES** ou **STRICT_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**.

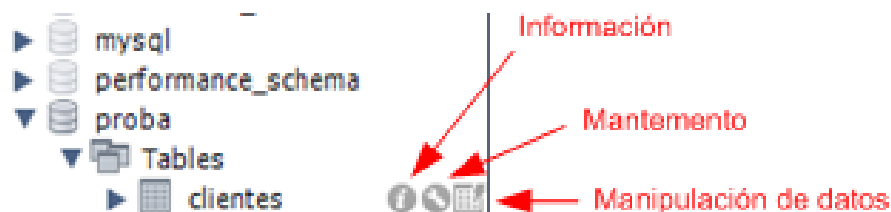
```
set session sql_mode = 'PIPES_AS_CONCAT,NO_ENGINE_SUBSTITUTION';
select @@session.sql_mode, @@global.sql_mode;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
@@session.sql_mode		@@global.sql_mode		
PIPES_AS_CONCAT,NO_ENGINE_SUBSTITUTION		NO_ENGINE_SUBSTITUTION		

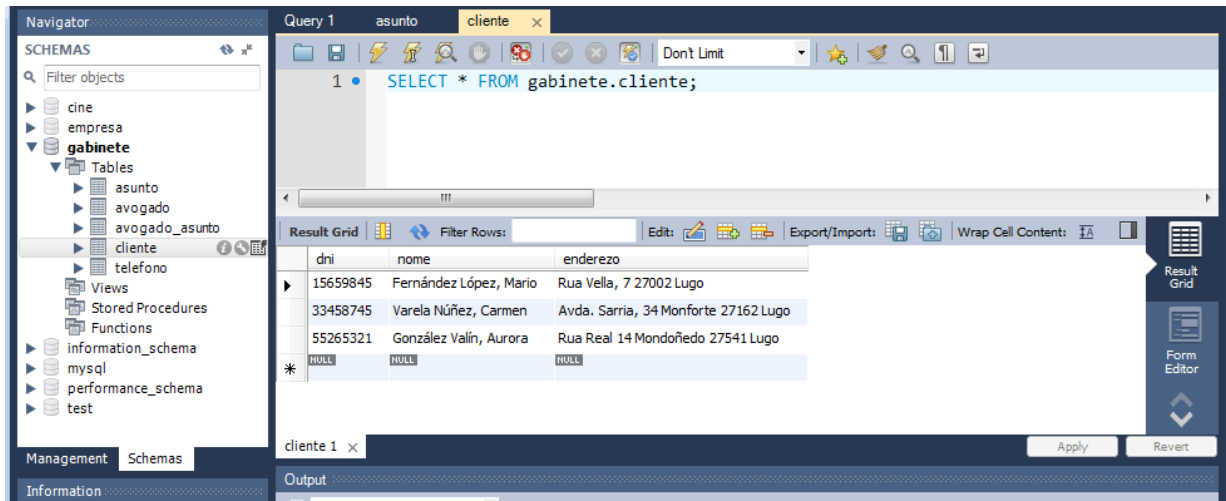
1.2.2 Edición de datos con MySQL Workbench

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 as sentenzas SQL necesarias para a carga de datos aínda non foron explicadas.

MySQL Workbench proporciona un contorno gráfico de edición dos datos contidos nas táboas, que permite aos usuarios 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.



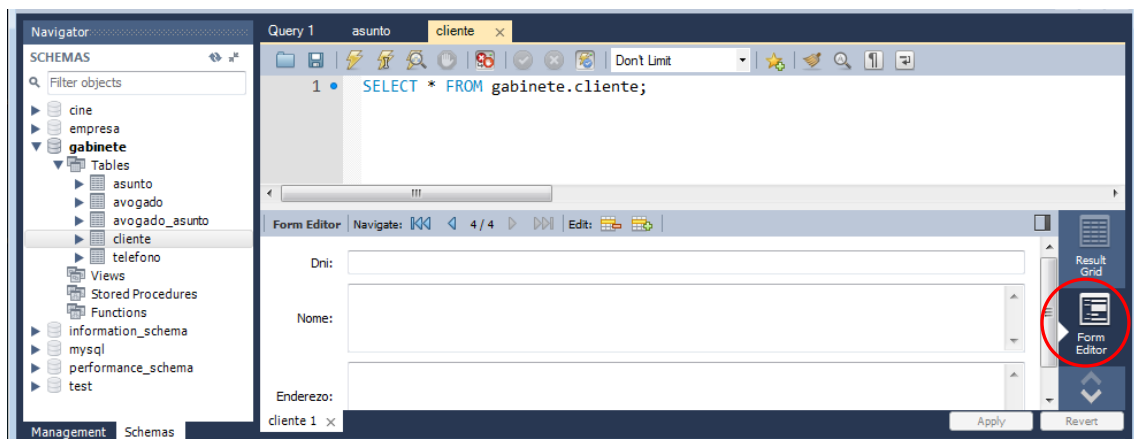
Para inserir novas filas, hai que escribir os datos na última fila, en substitución dos valores **NULL** e a continuación seleccionar o botón 'Apply' para aplicar os cambios.



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.



1.2.3 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 base 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*).

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 (clave primaria) 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 (STRICT_ALL_TABLES ou STRICT_TRANS_TABLES).

```
-- habilitando o modo SQL estrito
set session sql_mode = 'STRICT_ALL_TABLES,NO_ENGINE_SUBSTITUTION';
```

1.2.3.1 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, descripcion, corpo) values
('MP0484', 7, 'Bases de datos', 'S');
insert into modulo (idModulo, horasSemanais, descripcion, corpo) values
('MP0485', 9, 'Programación', 'S');
```

```

insert into modulo (idModulo, horasSemanais, descripcion, corpo) values
('MP0373', 5, 'Linguaxes de marcas e sistemas de xestión de información', 'S');
insert into modulo (idModulo, horasSemanais, descripcion, corpo) values
('MP0487', 4, 'Contornos de desenvolvemento', 'S');
insert into modulo (idModulo, horasSemanais, descripcion, corpo) values
('MP0615', 5, 'Deseño de interfaces web', 'T');
insert into modulo (idModulo, horasSemanais, descripcion, corpo) values
('MP0615', 5, 'Desenvolvemento web en contorno servidor', 'T');
insert into modulo (idModulo, horasSemanais, descripcion, 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:

	Time	Action	Message	Duration / Fetch
✓	1 13:01:45	insert into modulo (idModulo, horas_semanais, descripcion, corpo) values ('MP0484', 7, 'Bases de datos', 'S')	1 row(s) affected	0.020 sec

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, descripcion, corpo) values
('MP0615', 5, 'Desenvolvemento web en contorno servidor', 'T');

```

✗	6 13:07:00	insert into modulo (idModulo, horas_semanais, descripcion, corpo) values ('MP0615', 5, 'Desenvolvemento w...	Error Code: 1062. Entrada duplicada 'MP0615' para l...
---	------------	--	--

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, descripcion, corpo) values
(null, 5, 'Desenvolvemento web en contorno servidor', 'T');

```

✗	7 13:10:18	insert into modulo (idModulo, horas_semanais, descripcion, corpo) values (null, 5, 'Desenvolvemento ...	Error Code: 1048. La columna 'idModulo' no puede ser nula
---	------------	---	---

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.

Result Grid					Filter Rows:	Edit:	Export/Import
	idModulo	horas_semanais	descripcion	corpo			
▶	MP0373	5	Linguaxes de marcas e sistemas de xestión de i...	S			
	MP0484	7	Bases de datos	S			
	MP0485	9	Programación	S			
	MP0487	4	Contornos de desenvolvemento	S			
	MP0615	5	Deseño de interfaces web	T			

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.

1.2.3.2 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, CASCADE 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: RESTRICIÓN 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: Executar primeiro este script.

```
-- 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', 'Roís 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);
```

Script para comprobación da integridade referencial:

```
insert into grupo (idGrupo, descripcion, horasTitoria, idProfesor) values
('ADSIINRE-1A','1º Ordinario Administración de sistemas informáticos en rede A',2, 'pepe');
insert into grupo (idGrupo, descripcion, horasTitoria, idProfesor) values
('(A)ADSIINRE-A','Adultos - Administración de sistemas informáticos en rede A',2, null);
insert into grupo (idGrupo, descripcion, 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:

```

✓ 1 17:39:05 insert into grupo (idGrupo, descripcion, horasTitoria, idProfesor) values ('ADSIINRE-1A','1ª Administración d... 1 row(s) affected
✓ 2 17:39:05 insert into grupo (idGrupo, descripcion, horasTitoria, idProfesor) values (('A)ADSIINRE','Adultos - Administraci... 1 row(s) affected

```

A terceira sentença 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.

```

✗ 3 17:22:40 insert into grupo (idGrupo, descripcion, horasTitoria, idProfesor) values (('A)DAW-A','Adultos- Desenvolvemen... Error Code: 1452. Cannot add or update a child row: a f...

```

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:

idGrupo	descripcion	horasTitoria	idProfesor
(A)ADSIINRE	Adultos - Administración de sistemas informáticos en rede	2	NULL
ADSIINRE-1A	1ª Administración de sistemas informáticos en rede	2	pepe

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 sentença 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:

```

✗ 1 18:00:17 delete from profesor where idProfesor='pepe' Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails ('eleccionmodulos'.`grupo`, CON...

```

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';
```

1 18:22:13 update profesor set idProfesor = 'jose' where idProfesor='pepe' 1 row(s) affected Líneas correspondientes: 1 Cambiadas: 1 Avisos: 0

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.

Result Grid				
Filter Rows:				
Edit:				
Export/Import:				
	idGrupo	descripcion	horasTitoria	idProfesor
▶	(A)ADSIINRE	Adultos - Administración de sistemas informáticos en rede	2	NULL
	ADSIINRE-1A	1ªA Administración de sistemas informáticos en rede	2	jose

Cambio 'en cascada' ←

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.

1.2.3.3 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, apellidos, 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 restrición 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.

1.2.3.4 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, apellidos) values
('andrea', 'T', '26547123.95', 'Andrea', 'Barrio Varela');
```

✔ 4 19:01:51 insert into profesor (idProfesor, corpo, nrp, nome, apellidos) values ('andrea', 'T', '26547123.95', 'Andrea', 'Barrio Varela') 1 row(s) affected

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.

Result Grid						
Filter Rows:						
Edit:						
Export/Import						
	idProfesor	corpo	nrp	nome	apellidos	horasLectivas
▶	andrea	T	26547123.95	Andrea	Barrio Varela	18

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.

1.2.3.5 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 implementa esta validación a partir da versión 8.0. Nas versións 5.X, o servidor non levaba a cabo esta comprobación.

1.2.3.6 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, descripcion, corpo) values
('MP0613', 7, 'Desenvolvemento web en contorno servidor', 'P');
-- probas para columna e tipo ENUM. Valores válidos
insert into modulo (idModulo, horasSemanais, descripcion, corpo) values
('MP0613', 7, 'Desenvolvemento web en contorno servidor', 'S');
insert into modulo (idModulo, horasSemanais, descripcion, 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.

✖ 6 19:13:35 insert into modulo (idModulo, horas_semanais, descripcion, corpo) values ('MP0613', 7, 'Desenvolv... Error Code: 1265. Datos truncados para columna 'corpo' en ...

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

✔ 7 19:13:44 insert into modulo (idModulo, horas_semanais, descripcion, corpo) values ('MP0613', 7, 'Desenvolv... 1 row(s) affected
✔ 8 19:13:44 insert into modulo (idModulo, horas_semanais, descripcion, corpo) values ('MP0612', 6, 'Desenvolv... 1 row(s) affected

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

Result Grid		Filter Rows:		Edit:	Export/Import	
	idModulo	horas_semanais	descripcion	corpo		
	MP0613	7	Desenvolvimento web en contorno servidor	S		
	MP0612	6	Desenvolvimento web en contorno cliente	NULL		

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.

1.2.3.7 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 *descripcion* 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, descripcion, corpo)
values ('MP0483', 6, null, null);
```

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

```
✖ 13 15:00:28 insert into modulo (idModulo, horas_semanais, descripcion, corpo) values ('MP0483', 6, null, null) Error Code: 1048. La columna 'descripcion' no puede ser nula
```

1.2.3.8 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, descripcion, horasTitoria, idProfesor) values
('DAM-1A', '1ºA - Desenvolvemento de aplicacións multiplataforma', 'muchas', 'angel');
insert into profesor (idProfesor, corpo, nrp, nome, apellidos) values
('federico', 'S', '22369741.27', 'Federico Fermín de los Santos Inocentes', 'Valencia Bargas');
```

```
✖ 3 13:54:03 insert into grupo (idGrupo, descripcion, horasTitoria, idProfesor) values ('DAM-1A', '1ºA - ... Error Code: 1366. Incorrect integer value: 'muchas' for column 'horasTitoria' ...
✖ 4 13:54:03 insert into profesor (idProfesor, corpo, nrp, nome, apellidos) values ('federico', 'S', '25532... Error Code: 1406. Data too long for column 'nome' at row 1
```

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;
```

```
✔ 5 13:54:03 alter table profesor modify nome varchar(50) not null 7 row(s) affected Registros: 7 Duplicados: 0 Peligros: 0
```


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.

6 13:54:03 insert into profesor (idProfesor, corpo, nrp, nome, apelidos) values ('federico', 'S', '25532...' 1 row(s) affected

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

idProfesor	corpo	nrp	nome	apelidos	horasLectivas
andrea	T	26547123.95	Andrea	Barrio Varela	18
angel	S	152365842.37	Angel	Pérez Silva	17
berta	S	33858452.952	Berta	Pereiro Mato	20
carmen	T	362158472.12	Carmen	López Dávila	18
federico	S	25532741.27	Federico Fermín de los Santos Inocentes	Valencia Bargas	18
jose	S	1254876.20	Jose	Oliveira Fernández	19
luis	S	26587412.31	Luis	Rois Castro	17
maria	S	32569874.68	Maria	Fernández Souto	18

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

1.2.4 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