Dirección Xeral de Educación, Formación Profesional e Innovación Educativa

Material para

a formación profesional inicial

A01. Consultas simples cunha táboa

|  |  |  |
| --- | --- | --- |
| 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 | UD05 | Consultas de datos |
| **Actividade** | **A01** | **Consultas simples cunha táboa** |
| 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\_V000501\_UD05\_A01\_ConsultasSimples.docx |
| © 2015 Xunta de Galicia.  Consellería de Cultura, Educación e Ordenación Universitaria.  Este traballo foi realizado durante unha licenza de formación retribuída pola Consellería de Cultura, Educación e Ordenación Universitaria e ten licenza CreativeCommons BY-NC-SA (recoñecemento - non comercial - compartir igual). Para ver unha copia desta licenza, visitar a ligazón http://creativecommons.org/licenses/by-nc-sa/3.0/es/. | | |

Índice

[1. Ficha técnica 5](#_Toc439853478)

[Contexto da actividade 5](#_Toc439853479)

[Título da actividade 6](#_Toc439853480)

[Resultados de aprendizaxe do currículo 6](#_Toc439853481)

[Obxectivos didácticos e título e descrición da actividade 6](#_Toc439853482)

[Criterios de avaliación 6](#_Toc439853483)

[Contidos 6](#_Toc439853484)

[AcActividades de ensino e aprendizaxe e de avaliación, métodos, recursos e instrumentos de avaliación 7](#_Toc439853485)

[2. A01. Consultas simples cunha táboa 8](#_Toc439853486)

[2.1 Introdución 8](#_Toc439853487)

[2.1.1 Obxectivos 8](#_Toc439853488)

[2.1.2 Software 8](#_Toc439853489)

[2.1.3 Bases de datos de traballo 9](#_Toc439853490)

[2.1.3.1 Base de datos tendaBD 9](#_Toc439853491)

[2.1.3.2 Base de datos practicas5 11](#_Toc439853492)

[2.1.3.3 Base de datos traballadores 12](#_Toc439853493)

[2.2 Actividade 13](#_Toc439853494)

[2.2.1 Manipulación de datos con SQL 13](#_Toc439853495)

[2.2.2 Ferramentas para realizar consultas 13](#_Toc439853496)

[2.2.3 Sentenza SELECT para consulta de datos 14](#_Toc439853497)

[2.2.3.1 Lista de selección 15](#_Toc439853498)

[2.2.3.2 Expresións 17](#_Toc439853499)

[Constantes 17](#_Toc439853500)

[Variables de usuario 18](#_Toc439853501)

[Operadores 19](#_Toc439853502)

[Operadores aritméticos 19](#_Toc439853503)

[Operadores relacionais 19](#_Toc439853504)

[Operadores lóxicos 20](#_Toc439853505)

[Precedencia dos operadores 20](#_Toc439853506)

[2.2.4 Consultas simples 21](#_Toc439853507)

[2.2.4.1 Cláusula FROM 21](#_Toc439853508)

[2.2.4.2 Cláusula WHERE 21](#_Toc439853509)

[Predicado BETWEEN 22](#_Toc439853510)

[Predicado IN 22](#_Toc439853511)

[Predicado LIKE 23](#_Toc439853512)

[Predicado REGEXP 23](#_Toc439853513)

[Predicado IS [ NOT ] NULL 25](#_Toc439853514)

[2.2.4.3 Cláusula ORDER BY 26](#_Toc439853515)

[2.2.4.4 Cláusula LIMIT 27](#_Toc439853516)

[2.2.5 Funcións incorporadas en MySQL 28](#_Toc439853517)

[2.2.5.1 Funcións de data e hora 28](#_Toc439853518)

[Información sobre data e hora do servidor 28](#_Toc439853519)

[Aritmética de datas 28](#_Toc439853520)

[Formato de saída para datos tipo data e hora 29](#_Toc439853521)

[2.2.5.2 Funcións de cadeas de carácteres 30](#_Toc439853522)

[2.2.5.3 Funcións numéricas 33](#_Toc439853523)

[2.2.5.4 Funcións de agrupamento ou de columna 34](#_Toc439853524)

[Valores NULL e as funciones de agrupamento: 36](#_Toc439853525)

[2.2.5.5 Outras funcións 36](#_Toc439853526)

[Funcións de control de fluxo 36](#_Toc439853527)

[Funcións de información do sistema 37](#_Toc439853528)

[Funcións de cifrado 37](#_Toc439853529)

[Funcións de conversión de tipos 38](#_Toc439853530)

[2.3 Tarefas 40](#_Toc439853531)

[2.3.1 Tarefa 1. Escribir e probar consultas simples 40](#_Toc439853532)

[Solución 42](#_Toc439853533)

[2.3.2 Tarefa 2. Escribir e probar consultas simples que utilizan funcións 48](#_Toc439853534)

[Solución 49](#_Toc439853535)

[3. Materiais 54](#_Toc439853536)

[3.1 Documentos de apoio ou referencia 54](#_Toc439853537)

[3.2 Recursos didácticos 54](#_Toc439853538)

[3.3 Material auxiliar 54](#_Toc439853539)

[4. Avaliación 55](#_Toc439853540)

[Criterios de avaliación 55](#_Toc439853541)

[Modelo de proba combinada para PE.1, TO.1 e TO.7 55](#_Toc439853542)

[Proba para CA5.1 sobre identificación de sentenzas e ferramentas 56](#_Toc439853543)

[Proba para CA5.2 sobre consultas simples sobre unha táboa 56](#_Toc439853544)

[Proba para CA5.8 sobre utilización de funcións MySQL nas consultas 56](#_Toc439853545)

[Exemplo de solución para entregar en papel 57](#_Toc439853546)

[Exemplo de lista de valoración para TO.1 59](#_Toc439853547)

[Exemplo de lista de valoración para TO.7 59](#_Toc439853548)

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\_V000500\_UD05\_Consultas.pdf

Título da actividade

|  |  |  |  |
| --- | --- | --- | --- |
| Nº | Título | Descrición | Duración |
| A01 | Consultas simples cunha táboa | Realizáronse consultas simples dos datos contidos nunha táboa e manexáronse as funcións incorporadas en MySQL, utilizando a linguaxe de manipulación de datos. | 12 |

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 | Non |

Obxectivos didácticos e título e descrición da actividade

| Obxectivos específicos | | Actividade | | Descrición básica | Duración |
| --- | --- | --- | --- | --- | --- |
| O1.1 | Realizar consultas simples nunha táboa da base de datos utilizando a sentenza SELECT coas cláusulas FROM, WHERE, ORDER BY e LIMIT. | A02 | Consultas simples cunha táboa | Realizáronse consultas simples dos datos contidos nunha táboa e manexáronse as funcións incorporadas en MySQL, utilizando a linguaxe de manipulación de datos. | 12 |
| O1.2 | Utilizar na consulta funcións integradas no xestor de base de datos. |

Criterios de avaliación

|  |
| --- |
| Criterios de avaliación |
| * CA5.1 Identificáronse as ferramentas e as sentenzas para realizar consultas. * CA5.2 Realizáronse consultas simples sobre unha táboa. * CA5.8 Realizáronse consultas utilizando funcións básicas integradas no SXBD. |

Contidos

| Contidos |
| --- |
| * Ferramentas gráficas proporcionadas polo sistema xestor para a realización de consultas. * Funcións básicas integradas no SXBD. * Sentenza SELECT. * Selección e ordenación de rexistros. Tratamento de valores nulos. * Operadores: de comparación e lóxicos. Precedencia de operadores. * Consultas calculadas. * Sinónimos |

AcActividades 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 |
| A01. Consultas simples cunha táboa   * Realizáronse consultas simples dos datos contidos nunha táboa e manexáronse as funcións incorporadas en MySQL, utilizando a linguaxe de manipulación de datos. | * **Tp1.1** Exposición de conceptos sobre sentenzas LMD (Linguaxe de manipulación de datos) que permiten realizar consultas simples sobre unha táboa. | **Ta1.1** 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. |  | 1 |
| * **Tp1.2** Demostración práctica de realización de consultas simples sobre unha táboa utilizando sentenzas LMD e funcións incorporadas en MySQL. | * **Ta1.2** Realización de consultas simples sobre unha táboa utilizando sentenzas LMD e funcións incorporadas en MySQL | * Guións de sentenzas LMD que solucionan os exercicios de consultas propostos. |  | 10.5 |
|  | * **Ta1.3** Tarefa de avaliación combinando os instrumentos PE.1, TO.1 e TO.7. | * Exame en papel e en formato dixital. | * Máquina virtual para exame que terá instalado MySQL e MySQL Workbench. * Manual de referencia de MySQL. | * **PE.1** - Cuestionario con preguntas de resposta breve sobre ferramentas e sentenzas para consultas. * **TO.1** - Documento de rexistro de realización de consultas simples cunha táboa. Escala de valores (observación indirecta). * **TO.7** - Documento de rexistro de realización de consultas que utilicen funcións básicas integradas no SXBD. Escala de valores (observación indirecta). | 0.5 |

1. A01. Consultas simples cunha táboa
   1. Introdución
      1. Obxectivos

Os obxectivos desta actividade son:

* Realizar consultas simples nunha táboa da base de datos utilizando a sentenza SELECT coas cláusulas FROM, WHERE, ORDER BY e LIMIT.
* Utilizar na consulta funcións integradas no xestor de base de datos.
  + 1. Software

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

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

Utilizarase MySQL Workbench 6.3 como ferramenta cliente gráfica 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.

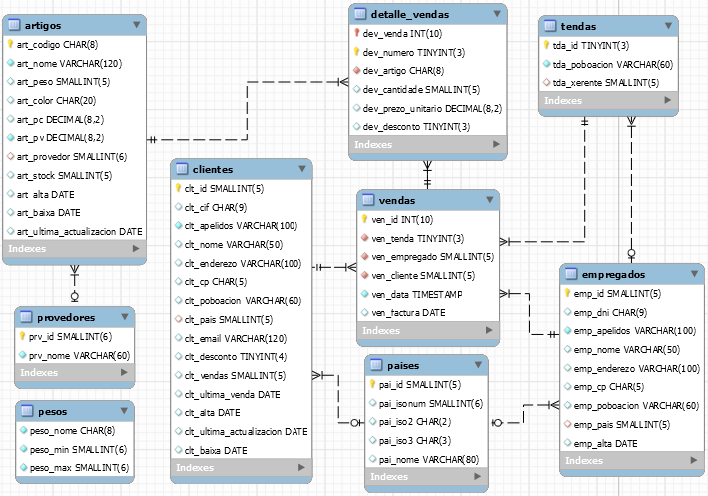
Normalmente, para a proba das consultas realizadas nesta actividade, mostrarase a zona de manipulación de datos de Workbench, coas filas que forman a táboa de resultados. Para completar a anterior información ou cando a consulta non pode mostrarse enteira xa que devolve moitas filas, mostrarase ademais a zona de saída (output) coa información do estado da execución da consulta e o número de filas que devolve.

* 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 *tendaBD*, *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 tendaBD

A base de datos *tendaBD* 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*

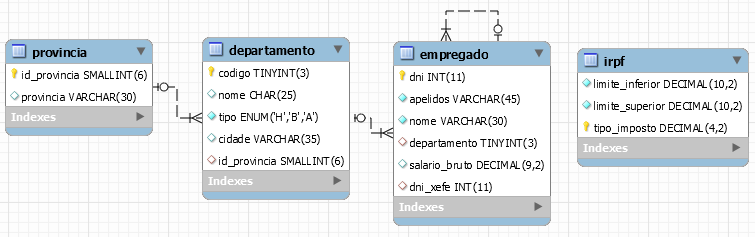
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome columna | Tipo | Null | Clave | Observacións |
| ven\_id | int unsigned | Non | Primaria | Identificador da venda. Numeraranse de 1 en adiante de forma automática. |
| ven\_tenda | tinyint unsigned | Non | Foránea | Identificador da tenda na que 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. Manipulación de datos con SQL

SQL corresponde ao acrónimo de *Structured Query Language* (Linguaxe Estruturado de Consultas). Aínda que nun principio foi creado para realizar consultas, utilízase para controlar todas as funcións que subministra un SXBDR aos seus usuarios, incluíndo todas as funcións propias das linguaxes deseñadas para o manexo de bases de datos: Linguaxe de Definición de Datos, Linguaxe de Manipulación de Datos e Linguaxe de Control de Datos.

A linguaxe de manipulación de datos ou LMD (en inglés *Data Management Language* ou *DML*), permite realizar as operación necesarias para manexar os datos almacenados nunha base de datos. Estas operacións son: inserir filas de datos (sentenza INSERT), modificar o contido das filas de datos (sentenza UPDATE), borrar filas de datos (sentenza DELETE), e consultar os datos contidos nas táboas da base de datos (sentenza SELECT).

* + 1. Ferramentas para realizar consultas

Para realizar consultas necesítase o código coas sentenzas SQL, un cliente conectado ao servidor que permita enviar o código e un servidor que execute ese código.

A escritura de guións de sentenzas SQL pode facerse con calquera editor de texto plano.

As aplicacións cliente cunha interface gráfica (GUI) incorporan un editor especializado na escritura de guións e posibilitan ademais de maneira sinxela e rápida a execución dos guións mediante un sistema de menús ou combinación de teclas.

As aplicacións cliente máis empregadas para conectase a un servidor MySQL e enviarlle as sentenzas SQL para que se executen, son as seguintes:

* MySQL Workbench

Cliente gráfico que de forma fácil e intuitiva permite establecer conexión cun servidor, mostrar información sobre a conexión, o estado do servidor, bases de datos e táboas; escribir sentenzas SQL mediante un editor; enviar sentenzas ao servidor para que sexan executadas; e dispón de axuda nas tarefas de edición e execución de sentenzas. Para máis información sobre o seu funcionamento, débese utilizar o manual de referencia ou ben a guía básica de funcionamento de MySQL Workbench incluído no material auxiliar da actividade.

* Consola modo texto mysql.exe

Cliente en modo texto que permite establecer a conexión co servidor e escribir e executar sentenzas SQL.

* No caso de ter instalado Wampserver, hai que abrir o menú de Wampserver, e seleccionar 'MySQL console' no submenú de MySQL. Ábrese unha consola en modo texto que solicita introducir o contrasinal do usuario *root*. Unha vez introducido o contrasinal, móstrase o prompt *mysql>* que indica que está establecida a conexión co servidor e se poden empezar a escribir sentenzas para que o servidor as execute.
* No caso de non ter instalado Wampserver, hai que abrir unha consola de ordes do sistema, e executar a utilidade cliente mysql.exe que está no directorio bin que se atopa no directorio de instalación de MySQL (en Windows é normalmente, c:\Program Files\MySQL\MySQL version).
* Aplicación web phpMyAdmin:

Cliente web que de forma gráfica permite establecer a conexión co servidor e enviarlle sentenzas SQL para que sexan executadas.

* No caso de ter instalado Wampserver, hai que abrir o menú e seleccionar a opción phpMyAdmin.
* No caso de non ter instalado Wampserver, hai que abrir un navegador e teclear a url coa identificación do servidor web no que está instalado phpMyAdmin. Exemplos de url: http://localhost/phpMyAdmin (no caso que phpMyAdmin estea instalado nun servidor web no equipo local), http://proveedor\_web.com/phpMyAdmin (no caso que phpMyAmin estea instalado no servidor proveedor\_web.com).
  + 1. Sentenza SELECT para consulta de datos

A sentenza SELECT permite realizar consultas e é a instrución máis potente e complexa de todas as instrucións SQL. A pesar da gran cantidade de opcións que ofrece, é posible empezar formulando consultas simples e ir construíndo consultas máis complexas a medida que se teña maior coñecemento da linguaxe.

A sentenza SELECT recupera datos das táboas dunha base de datos e devolve o resultado da consulta en forma de táboa, coas filas e columnas seleccionadas. Resumo da sintaxe da sentenza:

SELECT [ALL | DISTINCT | DISTINCTROW] lista\_de\_selección

[FROM lista\_referencias\_táboas

[WHERE expresión\_condicional]

[GROUP BY {lista\_columnas | expresión | posición}

[HAVING expresión\_condicional]

[ORDER BY {columna | expresión | posición} [ASC | DESC], ...]

[LIMIT intervalo]

[INTO OUTFILE 'nome\_arquivo']

* As opcións DISTINCT e ALL especifican se hai que mostrar as filas duplicadas ou non. A opción ALL é o valor predeterminado se non se indica ningunha das opcións, e significa que se mostran todas as filas, incluídas as duplicadas. A opción DISTINCT utilízase cando se queren eliminar as filas duplicadas do conxunto de resultados. A opción DISTINCTROW é un sinónimo de DISTINCT.
* A lista de selección (*lista\_de\_selección)* é a relación dos datos que se queren obter na consulta, separados por comas.
* A cláusula FROM utilízase para escribir a relación de táboas utilizadas na consulta e as relacións que hai entre elas.
* A cláusula WHERE utilízase para escribir as condicións que teñen que cumprir as filas para que se mostren no conxunto de resultados.
* A cláusula GROUP BY permite agrupar as filas do conxunto de resultados.
* A cláusula HAVING utilízase para escribir as condicións que teñen que cumprir os grupos resultantes de agrupar as filas do conxunto de resultados coa cláusula GROUP BY.
* A cláusula LIMIT utilízase para restrinxir o número de filas retornadas pola consulta.
* A cláusula INTO OUTFILE utilízase para enviar o conxunto de resultados a un arquivo.

Nas especificacións da sintaxe indícase a orde en que se teñen que escribir as cláusulas opcionais. No caso de escribir as cláusulas nun orden diferente do que figura na sintaxe, móstrase unha mensaxe de erro.

As sentenzas SELECT, igual que o resto de sentenzas de SQL rematan nun punto e coma, aínda que MySQL permite escribir as consultas sen o punto e coma final por tratarse dunha única sentenza. É unha boa práctica utilizar o punto e coma sempre, porque este é o xeito habitual de indicar o fin dunha instrución en programación.

* + - 1. Lista de selección

A *lista\_de\_selección* da sentenza SELECT, indica os datos que se queren obter coa consulta, separados por comas. Cada dato representa unha columna da táboa que contén o conxunto de resultados, e pode conter calquera das funcións e operadores que soporta MySQL. A sentenza permite recuperar datos de cálculos ou de información do servidor, sen facer referencia a ningunha táboa, como por exemplo:

**select** 2+2**,** **version(),** **user(),** **curdate();**



A lista de selección ten que cumprir a seguinte sintaxe:

expresión [ [ AS ] alias\_columna ] [, expresión [ [ AS ] alias\_columna ]] ...

* A *expresión* pode ser o nome dunha columna, unha expresión aritmética, unha expresión de data, unha constante, unha variable de usuario, unha función, ou unha combinación de todas esas cousas. Cando se fai referencia a un nome de columna, pode suceder que ese nome exista en máis dunha das táboas relacionadas na cláusula FROM; nese caso para diferencialas e que non existan nomes ambiguos na lista de selección, é obrigatorio utilizar nomes cualificados co formato *nome\_táboa.nome\_columna*.

Exemplo:

u501_lista_seleccion.emf

* Os *alias\_columna* son nomes que se lles dá ás columnas da táboa de resultados e son especialmente útiles no caso de columnas que conteñen unha expresión. Se non se asigna un alias, o nome da columna coincide coa expresión. Os alias poden ser utilizados nas cláusulas ORDER BY, GROUP BY e HAVING, pero non poden ser utilizados na cláusula WHERE xa que nesta cláusula establécense condicións para filtrar filas das táboas relacionadas na cláusula FROM, e os alias corresponden a nomes de columnas da táboa de resultados que no momento de resolver a cláusula WHERE aínda está en formación.

Como calquera nome de columna, os alias deben cumprir as regras do servidor referidas aos carácteres e tamaño permitidos, aínda que MySQL permite definir alias que non cumpran estas regras sempre que vaian pechados entre comiñas. Estes últimos só se deben utilizar para que o contido da columna se entenda mellor, pero poden dar problemas cando se utilizan noutras cláusulas.

Exemplos:

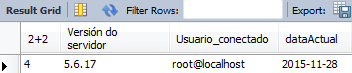
**select** 2+2**,**

**version()** **as** 'Versión do servidor'**,**

**user()** **as** Usuario\_conectado**,**

**curdate()** **as** dataActual**;**

A utilización de alias permite facilitar a lectura dos resultados, como se pode ver na saída que produce a consulta anterior.



Como a palabra AS é opcional, pode ocorrer un sutil problema no caso de esquecerse da coma entre dúas expresións da lista de selección: MySQL interpreta a segunda expresión como un nome de alias.

Exemplo:

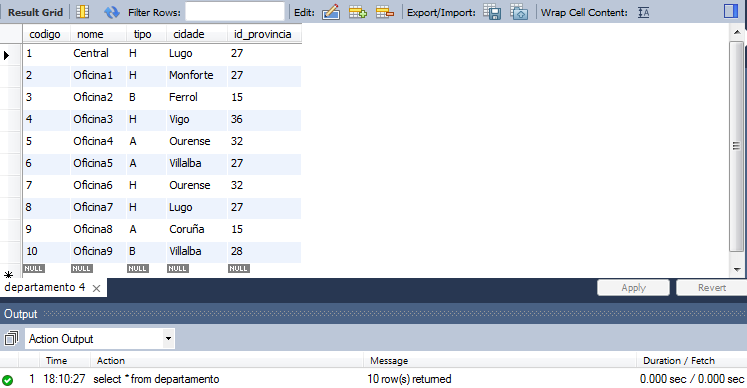
**select** apelidosnome**,** cidade **from** empregado;

Nesta consulta MySQL interpreta que *nome* é un alias de *apelidos*. Por esta razón é unha boa práctica poñer os alias de columnas empregando a palabra AS diante.

* Na lista de selección pódese empregar o símbolo \* para representar todas as columnas das táboas seleccionadas, ou empregar o formato *nome\_táboa*.\* para facer referencia a todas as columnas dunha táboa.

Exemplo:

**select** **\*** **from** departamento**;** /\* Mostra todas as columnas da táboa departamento\*/



Cando as consultas están embebidas no código dun programa nunca se debe utilizar o símbolo \*, xa que pode ocorrer que se modifique o esquema da táboa e isto afecte ao resultado que devolve a consulta. Por exemplo, a consulta anterior devolve 5 columnas; modifícase o esquema da táboa e engádese unha nova columna; despois do cambio, a consulta devolve 6 columnas; isto pode producir un erro na aplicación se é que o código da mesma só está preparado para manexar 5 columnas.

O \* é útil para facer probas de condicións, mostrando todas as columnas e seleccionando só as filas que cumpran a condición.

* + - 1. Expresións

Os elementos que poden conter as expresións son: nomes de columnas, constantes ou literais, variables de usuario, funcións e operadores.

Constantes

As constantes ou valores literais representan valores fixos. As máis utilizadas en MySQL son as de tipo cadeas de carácteres, números, data e hora, booleanas ou lóxicas, e o valor NULL.

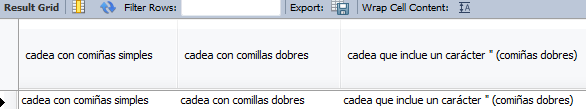
* Os valores constantes de tipo cadea de carácteres represéntanse por un conxunto de carácteres pechados entre comiñas simples ou dobres. A configuración por defecto do servidor MySQL, fai que non se diferencie entre maiúsculas e minúsculas, excepto que a cadea se defina como binaria (BINARY).

Exemplo:

**select** 'cadea con comiñas simples'**,**

"cadea con comiñas dobres"**,**

'cadea que inclúe un carácter " (comiñas dobres)'**;**



* Se o servidor ten habilitado o modo ANSI\_QUOTES, as cadeas só poden delimitarse con comiñas simples. Nese caso, unha cadea delimitada por comiñas dobres será interpretada como un identificador. Na instalación por defecto de MySQL este modo está deshabilitado.
* Os literais de cadea poden levar asociado un xogo de carácteres e un sistema de colación especificado cunha cláusula COLLATE. Isto pode ser importante cando se vai a comparar con outras cadeas. Sintaxe:

[\_nome\_xogo\_caracteres] 'cadea' [COLLATE tipo\_ordenación]

Exemplos:

**select** \_latin1 'cadea de carácteres'**;**

**select** \_latin1 'cadea de carácteres' **collate** latin1\_spanish\_ci**;**

Dentro das comiñas pódense utilizar algunhas secuencias de carácteres para representar carácteres especiais. Estas secuencias empezan cun carácter de escape, representado pola barra invertida (\). As secuencias de escape que recoñece MySQL pódense consultar no manual de referencia. Nesta táboa móstranse algúns dos máis empregados:

|  |  |
| --- | --- |
| \n | Carácter de salto de liña (LF). Código ASCII: 10 |
| \r | Carácter de retorno de carro. (CR). Código ASCII: 13 |
| \t | Carácter de tabulación |
| \\ | Carácter de barra invertida (\) |

Exemplos:

**select** 'exemplo \r\n cambio de liña'**;**

**select** 'A barra invertida non se mostra, \ pero pódese "escapar" \\ para mostrala'**;**

Resultado da execución:

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

* Os valores constantes de tipo numérico enteiro represéntanse como unha secuencia de díxitos; os de tipo decimal utilizan o carácter punto (.) como separador decimal. Calquera número pode ir precedido do signo menos (-) para indicar un valor negativo, ou do signo máis para indicar un valor positivo (+). Se non se pon ningún signo, considérase positivo. Exemplos:

**select** 258 **as** enteiro**,** 3658.25 **as** 'decimal'**,** **-**25 **as** negativo**;**



* Os valores constantes de tipo DATE (data), TIME (hora) e DATETIME (data e hora) representan valores pechados entre comiñas simples ou dobres cun determinado formato.
* No caso das datas, o formato establecido é *'aaaa-mm-dd'* ou *'aa-mm-dd'*, onde *aaaa* e *aa* representan o ano con catro díxitos ou dous díxitos, *mm* os dous díxitos que corresponden ao mes e *dd* os dous díxitos que corresponden ao día. MySQL admite calquera carácter como delimitador de ano, mes e día; e incluso admite que se omitan os delimitadores.

Exemplos:

**select** '2015-12-25'**,**"15-12-25"**,**'20151225'

* No caso das horas, o formato establecido é *'hh:mm:ss'* onde *hh* representa os dous díxitos que corresponden á hora, *mm* os dous díxitos que corresponden aos minutos e *ss* os dous díxitos que corresponden aos segundos.

Exemplos:

**select** '12:00:30'**,**"12:00:30"**,**'120030'

* Os valores constantes de tipo lóxico en MySQL son TRUE (verdadeiro) e FALSE (falso) e avalíanse como 1 e 0 respectivamente. Os nomes destas constantes pódense escribir en calquera combinación de maiúsculas e minúsculas.

Exemplos:

**select** **TRUE,** **true,** **FALSE,** **false,** 3**>**2**,** 5**=**6**;**



* O valor NULL significa 'non hai dato' ou 'valor descoñecido'. A palabra pódese escribir en calquera combinación de maiúsculas e minúsculas. Hai que ter en conta que o valor NULL non é o mesmo que o valor 0 para tipos numéricos ou á cadea valeira para tipos cadea de carácteres.

Exemplos:

**select** **NULL,** **null,** @**variable;**



Variables de usuario

As variables de usuario permiten almacenar un valor e facer referencia a el máis tarde; isto fai posible pasar valores dunha sentenza a outra. As variables de usuario son propias da conexión na que se crean e non poden ser vistas por outros clientes. A memoria utilizada polas variables de usuario é liberada no momento en que se pecha a conexión. Os nomes das variables de usuario levan como primeiro carácter un símbolo @.

Para asignar valores ás variables de usuario utilízase a sentenza SET, ou ben pódense asignar na propia sentenza SELECT co operador := .

Exemplos:

**set** @nome **=** 'Pepe'**;**

**select** 'Bos días, '**,** @nome**;**



**select** @numero**:=**5**;**



**select** @numero**:=**@numero**+**1**;**



Operadores

MySQL dispón de multitude de operadores diferentes para cada un dos tipos de dato. Os operadores permiten construír expresións que se utilizan na lista de selección, ou nas cláusulas das sentenzas de manipulación de datos, incluída a sentenza SELECT.

Operadores aritméticos

Permiten facer operacións aritméticas con datos de tipo numérico. Se algún dos operandos toma o valor NULL o resultado da operación sempre é o valor NULL.

|  |  |
| --- | --- |
| \* | Multiplicación |
| / DIV | División e División enteira |
| % MOD | Resto da división enteira |
| + | Suma |
| - | Resta |

Operadores relacionais

Permiten facer comparacións entre expresións devolvendo o valor 1 (*true = verdadeiro*) ou 0 (*false = falso*). Se un dos valores a comparar é o valor NULL, o resultado e NULL, excepto cando se utiliza o operador <=>, ou o operador IS NULL.

|  |  |
| --- | --- |
| = | Igual |
| != <> | Distinto |
| > | Maior |
| < | Menor |
| >= | Maior ou igual |
| <= | Menor ou igual |
| <=> | Igual, pero devolve 1 en lugar de NULL se ambos operandos son NULL, ou 0 en lugar de NULL se un deles é NULL |
| IS | Comparación co valor NULL. Explícase con máis detalle no apartado: Cláusula WHERE |
| LIKE | Comparación con patrón de busca. Explícase con máis detalle no apartado: Cláusula WHERE |
| REGEXP | Comparación con patrón de busca. Explícase con máis detalle no apartado: Cláusula WHERE |
| IN | Comparación cun conxunto de valores. Explícase con máis detalle no apartado: Cláusula WHERE |
| BETWEEN | Comparación cun intervalo de valores. Explícase con máis detalle no apartado: Cláusula WHERE |

Operadores lóxicos

Permiten formar condicións compostas por máis dunha condición, ou negar unha condición.

|  |  |
| --- | --- |
| NOT | NEGACIÓN. Devolve o valor verdadeiro cando non é verdadeira a condición |
| AND && | E. Devolve o valor verdadeiro cando son verdadeiras as dúas condicións |
| XOR | OU LÓXICO. Devolve o valor verdadeiro cando é verdadeira unha das dúas condicións, pero non as dúas |
| OR || | OU. Devolve o valor verdadeiro cando é verdadeira algunha das dúas condicións, ou se son verdadeiras as dúas |

Para comparar filas de valores, a expresión (a, b) = (x, y) é equivalente á expresión (a = x) and (b=y).

Precedencia dos operadores

As expresións avalíanse tendo en conta a precedencia dos operadores que a forman. A precedencia está establecida polo SXBDR, aínda que é posible cambiala introducindo parénteses nas expresións. O primeiro que se vai a avaliar é a expresión que está contida entre parénteses e dentro dos parénteses utilízase a precedencia preestablecida. A precedencia dos operadores máis empregados móstrase na seguinte táboa, na que os operadores que están dentro da mesma liña teñen a mesma precedencia e por tanto execútanse a medida que se aparecen na expresión dende esquerda á dereita.

|  |  |
| --- | --- |
| 1 | \*, /, DIV, %, MOD |
| 2 | -, + |
| 3 | = (comparación), <=>, >=, >, <=, <, <>, !=, IS, LIKE, REGEXP, IN |
| 4 | BETWEEN, CASE, WHEN, THEN, ELSE |
| 5 | NOT |
| 6 | AND, && |
| 7 | XOR |
| 8 | OR, || |

Exemplos:

**select** 1+2**\***3**,** **(**1+2**)\***3**,** 5**/**2**,** 5 **div** 2**,** 5 **mod** 2**,** 5**%**2**; # aritméticos**



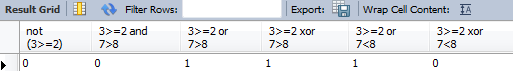
**select** 3**=**2**,** 3**<=>**2**,** **null** **=** **null,** **null** **<=>null,** 3 **=** **null,** 3 **<=>** **null;**



**select** 3**>=**2**,** 2**>**3**,** 2**<>**3**,** 3**!=**3**,** 3 **in** **(**2**,**3**,**4**);**



**select** **not (**3**>=**2**),** 3**>=**2 **and** 7**>**8**,** 3**>=**2 **or** 7**>**8**,** 3**>=**2 **xor** 7**>**8**,** 3**>=**2 **or** 7**<**8**,** 3**>=**2 **xor** 7**<**8



* + 1. Consultas simples

Son consultas que utilizan unicamente as cláusulas FROM, WHERE, ORDER BY, LIMIT e INTO OUTFILE, aínda que non teñen porque utilizalas todas. Nesta primeira actividade realízanse consultas deste tipo tomando os datos dunha soa táboa.

* + - 1. Cláusula FROM

Esta cláusula utilízase para escribir a relación das táboas nas que están os datos que se utilizan na consulta. Cada elemento da *lista\_referencias\_táboas* segue a seguinte sintaxe:

referencia\_táboa [[AS] alias\_táboa]

[[USE INDEX (key\_list)]

| [IGNORE INDEX (key\_list)]

| [FORCE INDEX (key\_list)]]

* A *referencia\_táboa* pode ser o nome dunha táboa ou unha expresión (posta entre parénteses) que dá como resultado unha táboa. O alias é un nome que pode ser empregado en lugar do nome da táboa. Se a referencia de táboa é unha expresión, hai que especificar obrigatoriamente un alias.

Exemplos:

* Utilizando o nome dunha táboa:

**select** dni,apelidos,nome

**from** empregado;

* Utilizando unha expresión que dá como resultado unha táboa:

**select** codigo,nome**,** cidade

**from**  (**select** \* **from** departamento **where** cidade = 'Lugo') as DepartamentosLugo;

Na segunda consulta, o resultado de executar a sentenza SELECT que vai pechada entre paréntese ten forma de táboa e asignáselle o nome *DepartamentosLugo*. A consulta principal colle os datos desa táboa.

O nome da táboa pode ter o formato *nome\_base\_datos.nome\_táboa* para indicar de forma explícita a base de datos á que pertence a táboa.

* As opcións USE INDEX, IGNORE INDEX, FORCE INDEX permiten dar pistas ao optimizador de consultas acerca de cómo escoller os índices para facer a consulta. Pode ser moi útil en consultas que utilizan máis dunha táboa.

Máis adiante verase a sintaxe da cláusula FROM cando hai que coller datos de máis dunha táboa.

* + - 1. Cláusula WHERE

Esta cláusula permite seleccionar as filas, mediante condicións do tipo:

expresión operador\_relacional expresión

expresión [NOT] BETWEEN expresión1 AND expresión2

expresión [NOT] IN (lista\_de\_valores)

nome\_columna [NOT] LIKE 'patrón \_de\_busca'

nome\_columna [NOT] REGEXP [BINARY] 'expresión\_regular'

nome\_columna IS [NOT] NULL

As condicións utilizadas na cláusula poden ser compostas, combinando máis dunha condición simple cos operadores lóxicos:

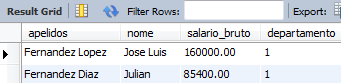
* AND ou && : Para que a condición composta devolva o valor verdadeiro todas as condicións teñen que ser verdadeiras.
* OR ou || : Para que a condición composta devolva o valor verdadeiro é suficiente con que sexa verdadeira algunha das condicións.
* XOR : Para que a condición composta devolva o valor verdadeiro ten que ser verdadeira algunha das condicións pero non as dúas.

Exemplo de condición composta: seleccionar os *apelidos*, *nome*, *salario\_bruto* e *departamento* dos empregados do departamento 1 que teñan un salario superior a 40000 euros.

**select** apelidos**,** nome**,** salario\_bruto**,** departamento

**from** empregado

**where** departamento **=** 1 **and** salario\_bruto **>** 40000**;**



Predicado BETWEEN

Permite comparar o valor da expresión situada á esquerda da palabra BETWEEN cos valores comprendidos no intervalo definido pola expresión1 e a expresión2, ambas incluídas. Sintaxe:

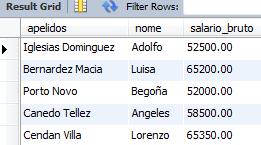
expresión [NOT] BETWEEN expresión1 AND expresión2

Exemplo: seleccionar *apelidos*, *nome* e *salario\_bruto* dos empregados que teñan un salario bruto entre 50000 e 70000 euros, ambos incluídos.

**select** apelidos**,** nome**,** salario\_bruto

**from** empregado

**where** salario\_bruto **between** 50000 **and** 70000**;**



Predicado IN

Permite comparar o valor da expresión situado á esquerda da palabra IN coa lista de valores pechados entre parénteses. Sintaxe:

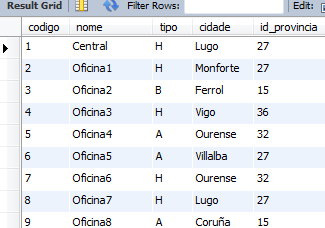
expresión [NOT] IN (lista\_de\_valores)

Exemplos:

* Seleccionar o *nome*, *cidade* e *id\_provincia* de todos os departamento situados en provincias galegas, e dicir, que a columna *id\_provincia* tome os valores 15, 27, 32 ou 36.

**select** **\*** **from** practicas5**.**departamento

**where** id\_provincia **in** **(**15**,**27**,**32**,**36**);**



* Seleccionar o *nome*, *cidade* e *id\_provincia* de todos os departamentos situados en provincias que non sexan galegas, e dicir, que a columna *id\_provincia* tome valores distintos a 15, 27, 32 ou 36.

**select** **\*** **from** practicas5**.**departamento

**where** id\_provincia **not** **in** **(**15**,**27**,**32**,**36**);**



Predicado LIKE

Permite realizar unha comparación de semellanza entre o valor dunha expresión e o dunha cadea de carácteres que representa un patrón de busca. Sintaxe:

nome\_columna [NOT] LIKE 'patrón \_de\_busca'

No patrón pódense empregar os seguintes carácteres 'comodín':

|  |  |
| --- | --- |
| %  (porcentaxe) | Substitúe a un grupo de carácteres. Exemplos:  'A%' representa un grupo de carácteres que empeza por 'A'  '%ez' representa un grupo de carácteres que acaba por 'ez'  '%a%' representa un grupo de carácteres que contén a letra 'a' |
| \_  (guión baixo) | Substitúe a un carácter. Exemplo:  '1\_ \_a' representa un grupo de carácteres que empeza por lo número 1, seguido de dous carácteres calquera e remata coa letra 'a' |

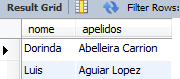
Exemplos:

* Seleccionar *apelidos*, *nome* e *salario\_bruto* de todos os empregados que teñen uns *apelidos* que empezan por 'A'.

**select** nome**,** apelidos

**from** empregado

**where** apelidos **like** 'A%'**;**

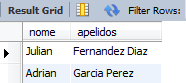


* Seleccionar o *nome* e *apelidos* de todos os empregados que teñen un *nome* formado por 6 letras, e acaba en 'AN'.

**select** nome**,** apelidos

**from** empregado

**where** nome **like** '\_\_\_\_an'**;**



Predicado REGEXP

Serve para atopar as coincidencias dunha expresión cunha expresión regular formada por unha cadea de carácteres que pode incluír carácteres especiais. Sintaxe:

expresión [NOT] REGEXP [BINARY] 'expresión\_regular'

* A opción BINARY permite diferenciar entre maiúsculas e minúsculas, aínda que a expresión non sexa de tipo binario.
* Unha expresión regular é un patrón no que se poden empregar carácteres especiais que teñen un significado predeterminado.

Permite buscas máis complexas que LIKE. Algún dos carácteres especiais que se poden utilizar nas expresións regulares:

|  |  |
| --- | --- |
| carácter | Detecta a coincidencia con ese carácter. Exemplo: 'a' detecta a aparición da letra 'a' en calquera lugar da expresión. |
| [expresión] | Detecta a coincidencia con calquera dos carácteres entre corchetes. O símbolo ^ ao inicio dos corchetes representa a negación. Exemplos:  '[ab]' detecta a letra 'a' ou 'b', maiúsculas ou minúsculas,  '[a-f]' detecta letras dende 'a' ata 'f',  '[1-3]' números do 1 ao 3, '[^ab]' letras distintas de 'a' e' b'. |
| ^ | Para o inicio da cadea. Exemplo: '^A' que empecen por 'A' ou 'a'. |
| $ | Para o final da cadea. Exemplo: 'A$' que acaben por 'A' ou 'a'. |
| carácter{n} | n carácteres ou expresións seguidas. Exemplos:  ''9{2}' representa dous noves seguidos,  '\\.9{2}$' representa unha cantidade que acaba en .99. |
| carácter{m,n} | Repetición dun carácter entre m e n (incluídos) veces. Exemplo: '[bc]{2,3}' representa a letra 'b' ou 'c' repetidas entre 2 e 3 veces, polo que detectaría 'bb', 'cc', 'bbb', 'ccc'. |
| .  (punto) | Detecta unha coincidencia individual incluíndo o retorno de carro e a nova liña. Utilízase normalmente en combinación con outros símbolos. Exemplos:  '.{5}' representa 5 carácteres,  '^.{5}$' representa exactamente 5 carácteres. |
| carácter \* | Detecta a secuencia de 0 ou máis aparicións do carácter ou expresión que o precede.Exemplos*:*  '^ba\*n' detectaría 'bn', 'ban', 'baaaaaaaan',  '^(ana)\*$' detectaría 'ana', 'anana', 'ananana'. |
| carácter + | Detecta a secuencia de 1 ou máis aparicións do carácter co precede. Exemplo: '^ba+n' detectaría 'ban', 'baaaaaaaan'. |
| carácter ? | Detecta a secuencia de 0 ou 1 aparicións do carácter co precede. Exemplo: '^ba?n' detectaría 'bn', 'ban'. |
| cadea1| cadea2 | Detecta a aparición de cadea1 ou cadea2. Exemplo: '^PABLO|PEDRO$' detectaría 'Pablo' ou 'Pedro'. |

Exemplos:

* Seleccionar *nome* e *apelidos* dos empregados que teñan o primeiro apelido que empece por unha letra entre 'F' e 'M':

**select** nome**,** apelidos

**from** empregado

**where** apelidos regexp '^[f-m]'**;**

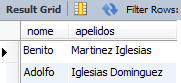


* Seleccionar *nome* e *apelidos* dos empregados que teñan un *nome* que conteña 6 letras, sendo a última a letra o minúscula:

**select** nome**,** apelidos

**from** empregado

**where** nome regexp **binary** '^.{5}o$'**;**



* Comparar as cadeas de carácteres 'anhna', 'anhnA', 'ANA' e 'anhna' co patrón '^.{4}A$' que representa unha cadea que empeza por catro carácteres calquera e remate cunha 'A'.

**select** 'anhna' regexp **binary** '^.{4}A$' **as** resultado**;**



**select** 'anhnA' regexp **binary** '^.{4}A$' **as** resultado**;**



**select** 'ANA' regexp **binary** '^.{4}A$' **as** resultado**;**



**select** 'anhna' regexp '^.{4}A$' **as** resultado**;**



Na última consulta devolve o valor verdadeiro porque non se utiliza a opción BINARY.

Predicado IS [ NOT ] NULL

Serve para comprobar se o contido dunha columna é un valor nulo ou non. En SQL, o valor nulo representa un valor descoñecido, diferente de 0 e dunha cadea baleira. Sintaxe:

nome\_columna IS [NOT] NULL

Exemplos:

* Seleccionar *nome*, *apelidos* e *dni\_xefe* dos empregados que non teñen xefe, ou non se coñece quen é o seu xefe, é dicir, os que teñen o valor NULL na columna *dni\_xefe*.

**select** nome**,** apelidos**,** dni\_xefe

**from** empregado

**where** dni\_xefe **is** **null;**

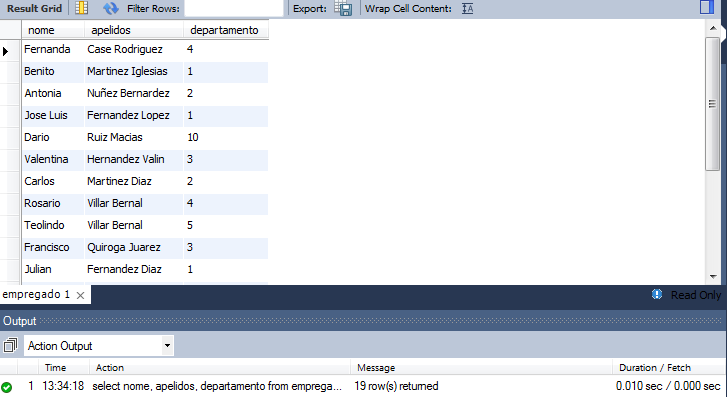


* Seleccionar *nome,* *apelidos* e *departamento* dos empregados que teñen asignado un departamento, e dicir, que teñen un valor distinto de NULL na columna *departamento*.

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

**from** empregado

**where** departamento **is** **not** **null;**



* + - 1. Cláusula ORDER BY

Esta cláusula permite ordenar os resultados da consulta tendo en conta o contido dunha ou máis columnas ou expresións. Sintaxe:

[ORDER BY {columna | expresión | posición} [ASC | DESC], ...]

* Os nomes de columna aos que se fai referencia nesta cláusula son os da táboa do conxunto de resultados da consulta, polo tanto, se hai alias asociados ás columnas, poden ser utilizados. Tamén se poden utilizar expresións, ou o número de orde da columna na lista de selección. Non é recomendable utilizar o número de orde da columna porque no caso de facer cambios na lista de selección, cambiando a orde das columnas, ou engadindo ou eliminando algunha columna, habería que cambiar tamén a cláusula ORDER BY para que non varíe o resultado.
* As opcións [ASC | DESC] indican a orde na que se mostran os resultados (ASC = ascendente, DESC = descendente). Se non se indica nada, tómase ASC como valor por defecto.

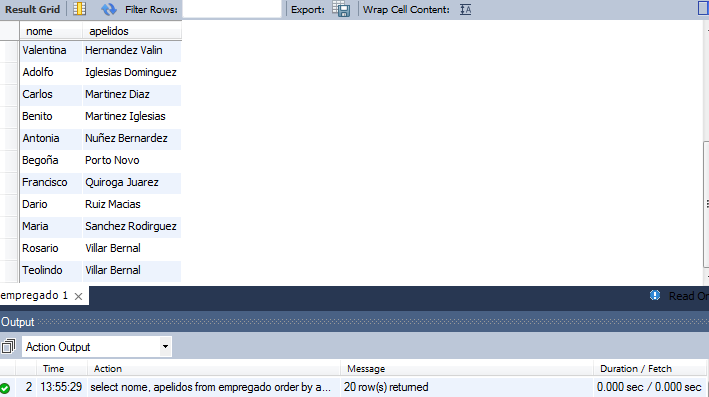
Exemplos:

* Seleccionar *nome*, e *apelidos* de todos os empregados ordenando o resultado alfabeticamente polos apelidos, e no caso de haber empregados que teñan os mesmos *apelidos*, ordénanse alfabeticamente polo *nome*.

**select** nome**,** apelidos

**from** empregado

**order** **by** apelidos**,** nome**;**

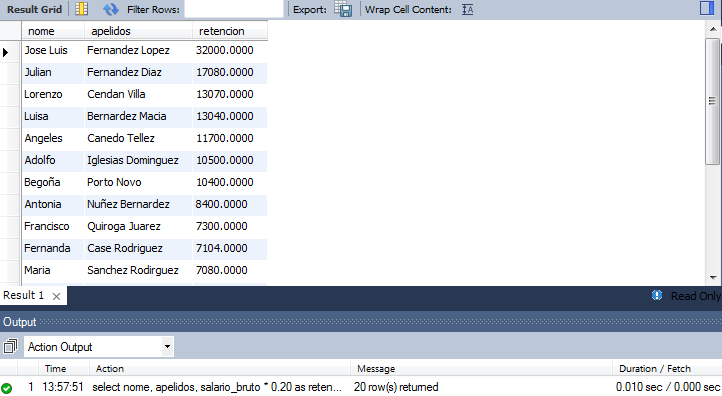


* Mostrar o nome, apelidos e o 20% do *salario\_bruto*, que representa a retención que se lle vai a facer, de todos os empregados ordenando o resultado pola retención, de tal maneira que se mostren en primeiro lugar os que teñen unha retención máis alta.

**select** nome**,** apelidos**,** salario\_bruto **\*** 0.20 **as** retencion

**from** empregado

**order** **by** retencion **desc;**



Calquera das seguintes solucións tamén serían válidas, aínda que a anterior é a máis recomendable:

-- utilizando unha expresión

**select** nome**,** apelidos**,** salario\_bruto **\*** 0.20 **as** retencion

**from** empregado

**order** **by** salario\_bruto **\*** 0.20 **desc;**

-- utilizando o número de orden da columna dentro da lista de selección

**select** nome**,** apelidos**,** salario\_bruto **\*** 0.20 **as** retencion

**from** empregado

**order** **by** 3 **desc;**

* + - 1. Cláusula LIMIT

Esta cláusula permite indicar o número de filas que ten que devolver a consulta e de forma optativa dende que posición se van a mostrar. Pode ter un ou dous argumentos numéricos, que deben ser enteiros positivos (incluíndo cero). No caso de utilizar un só argumento, este indica o número de filas que hai que mostrar; e no caso de utilizar dous argumentos, o primeiro indica o número de fila a partir da que hai que mostrar o conxunto de resultados, e o segundo representa o nº de filas.

Exemplos:

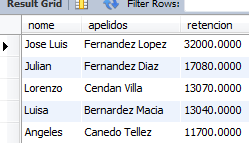
* Seleccionar o *nome*, *apelido* e retención dos 5 empregados coas retencións máis altas. A retención representa o 20% do *salario\_bruto*.

**select** nome**,** apelidos**,** salario\_bruto **\*** 0.20 **as** retencion

**from** empregado

**order** **by** retencion **desc**

**limit** 5**;**



A consulta mostra as 5 primeiras filas da táboa de resultados, despois de ordenar pola columna *retencion*.

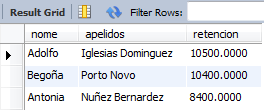
* Seleccionar o *nome*, *apelido* e *retención* dos 3 empregados seguintes aos mostrados no exemplo anterior, tendo en conta que están ordenados polas retencións, de maior a menor. A retención representa o 20% do salario\_bruto.

**select** nome**,** apelidos**,** salario\_bruto **\*** 0.20 **as** retencion

**from** empregado

**order** **by** retencion **desc**

**limit** 5**,**3**;**



A consulta mostra 3 filas da táboa de resultados, empezando pola 6ª, despois de ordenar pola columna *retencion*.

* Tarefa 1. Escribir e probar consultas simples.
  + 1. Funcións incorporadas en MySQL

Cando se necesita facer algunha operación complexa, débese de consultar o manual para saber se xa existe algunha función implícita no servidor para resolvela antes de poñerse a crear unha función nova de usuario. MySQL incorpora unha gran cantidade de funcións que poden ser utilizadas nas expresións contidas nas consultas. O manual de referencia de MySQL posúe información moi detallada das funcións; aquí móstrase un resumo das máis utilizadas.

* + - 1. Funcións de data e hora

Permiten obter información sobre a data e hora do sistema en distintos formatos, resolver os problemas de aritmética de datas (cálculos aritméticos feitos con datos tipo data e hora) e cambiar o formato de saída.

Información sobre data e hora do servidor

* CURRENT\_DATE() ou CURDATE()

Devolve a data actual do sistema, en formato 'aaaa-mm-dd'

* CURRENT\_TIME() ou CURTIME()

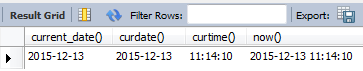
Devolve a hora actual do sistema, en formato 'hh:mm:ss'

* NOW()

Devolve a data e hora do sistema, en formato 'aaaa-mm-dd hh:mm:ss'

Exemplo considerando que agora son as 11:14:10 do 13/12/2015:

**select** **current\_date(),** **curdate(),** **curtime(),** **now();**



Aritmética de datas

Para facer operacións aritméticas con datas en MySQL, hai que utilizar as funcións que incorpora para tal fin. Algunhas destas funcións poden manexar intervalos de tempo facendo referencia ás palabras reservadas: MICROSECOND, SECOND, MINUTE, HOUR, DAY, WEEK, MONTH, QUARTER, ou YEAR.

* DATE\_ADD

Suma (*add*) a unha data que se pasa como primeiro parámetro o intervalo de tempo especificado como segundo parámetro. Sintaxe:

DATE\_ADD(data, INTERVAL expresión unidade\_intervalo)

* ADDDATE

Suma (*add*), a unha data que se pasa como primeiro parámetro, un número de días ou un intervalo de tempo, que se pasan como segundo parámetro. Sintaxe:

ADDDATE(data, {número\_días | INTERVAL expresión unidade\_intervalo )

Cando se utiliza coa forma INTERVAL no segundo parámetro, dá o mesmo resultado cá función DATE\_ADD.

* DATE\_SUB

Resta (*substract*) a unha data que se pasa como primeiro parámetro o intervalo de tempo especificado como segundo parámetro. Sintaxe:

DATE\_ADD(data, INTERVAL expresión unidade\_intervalo)

* SUBDATE

Resta (*substract*), a unha data que se pasa como primeiro parámetro, un número de días ou un intervalo de tempo especificado como segundo parámetro. Sintaxe:

SUBDATE(data, {número\_días|INTERVAL expresión unidade\_intervalo})

Cando se utiliza coa forma INTERVAL no segundo parámetro, dá o mesmo resultado cá función DATE\_SUB.

* DATEDIFF

Devolve o número de días transcorridos entre dúas datas que se pasan como parámetros. Sintaxe:

DATEDIFF(data\_maior, data\_menor)

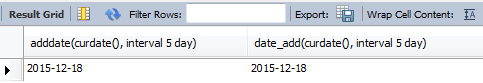
* TIMESTAMPDIFF

Devolve o número de intervalos de tempo (MICROSECOND, SECOND, MINUTE, HOUR, DAY, WEEK, MONTH, QUARTER, ou YEAR) que se pasa como primeiro parámetro, transcorridos entre dúas expresións tipo data e hora, ou tipo data. Sintaxe:

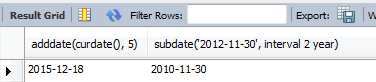
TIMESTAMPIFF(unidade\_intervalo, data\_hora\_inicio, data\_hora\_fin)

Exemplos considerando que hoxe é 2015/12/13 (según o formato de MySQL):

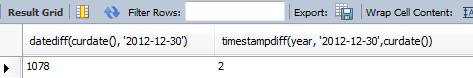
**select** **adddate(curdate(),** **interval** 5 **day), date\_add(curdate(),** **interval** 5 **day);**



**select** **adddate(curdate(),** 5**), subdate(**'2012-11-30'**,** **interval** 2 **year);**



**select** **datediff(curdate(),** '2012-12-30'**), timestampdiff(year,** '2012-12-30'**,curdate());**



Formato de saída para datos tipo data e hora

* DATE\_FORMAT

Aplica un formato de saída a unha expresión tipo data, ou tipo data e hora. Sintaxe:

DATE\_FORMAT(data, 'cadea\_de\_formato')

A cadea de formato representa a forma en que vai a mostrar a data. Dentro da cadea pódese poñer calquera carácter que será mostrado tal e como aparece na cadea de formato, e ademais, pode levar o carácter % que engade un significado especial a algúns carácteres. Algunhas combinacións de carácteres especiais permitidas son:

|  |  |
| --- | --- |
| %d | Día do mes, con dous díxitos (entre 01 e 31) |
| %a | Abreviatura do nome do día da semana (en inglés: Sun...Sat) |
| %W | Nome do día da semana (en inglés: Sunday...Saturday) |
| %w | Día da semana en cifras (0 = domingo ... 6 = sábado) |
| %m | Mes, con dous díxitos (entre 01 e 12) |
| %b | Abreviatura do nome do mes (en inglés: Jan...Dec) |
| %M | Nome do mes (en inglés: January...December) |
| %y | Ano, con dous díxitos |
| %Y | Ano, con catro díxitos |
| %h | Hora, con dous díxitos (entre 01 e 12) |
| %H | Hora, con dous díxitos (entre 00 e 23) |
| %i | Minutos, con dous díxitos (entre 00 e 59) |
| %s | Segundos, con dous díxitos (entre 00 e 59) |
| %r | Hora con formato 12 horas (hh:mm:ss), seguido de AM ou PM |
| %p | AM ou PM |
| %% | Mostra o literal % |

As sete últimas combinacións de carácteres desta lista tamén poden ser utilizados coa función TIME\_FORMAT.

* TIME\_FORMAT

Aplica un formato a unha expresión tipo hora. Funciona igual que DATE\_FORMAT, pero só pode levar as combinacións especiais de formato para horas, minutos, segundos e microsegundos. Sintaxe:

TIME\_FORMAT(hora, “cadea\_de\_formato”)

* DAY ou DAYOFMONTH

Devolve o día do mes da data que se pasa como parámetro, en número. Sintaxe:

DAY(data)

DAYOFMONTH(data)

* DAYOFWEEK

Devolve o número de día da semana da data que se pasa como parámetro. Mostra valores numéricos entre o 1 (para o domingo) e o 7 (para o sábado). Sintaxe:

DAYOFWEEK(data)

* MONTH

Devolve o mes da data especificada. Sintaxe:

MONTH(data)

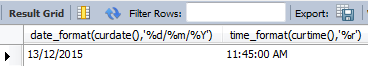
* YEAR

Devolve o ano da data especificada.Sintaxe:

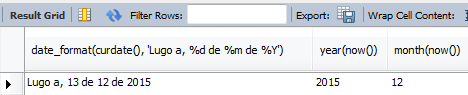
YEAR(data)

Exemplos considerando que agora son as 11:45:00 do 13/12/2015:

**select** **date\_format(curdate(),**'%d/%m/%Y'**),** **time\_format(curtime(),**'%r'**);**



**select** **date\_format(curdate(),** 'Lugo a, %d de %m de %Y'**), year(now()), month(now());**



* + - 1. Funcións de cadeas de carácteres
* ASCII

Devolve o código ASCII do primeiro carácter dunha cadea. Sintaxe:

ASCII(cadea)

* CHAR

Devolve os carácteres que se corresponden con cada número, segundo a táboa de códigos ASCII. Sintaxe:

CHAR(número1 [, número2] ...)

* HEX

Devolve a representación en hexadecimal dunha cadea ou dun valor numérico. Cando o parámetro é unha cadea de carácteres, devolve unha combinación de dous díxitos hexadecimais para cada carácter da cadea. Cando o parámetro é un número, devolve a representación do número en hexadecimal. Sintaxe:

HEX(cadea\_carácteres)

HEX(número)

* CONCAT

Concatena unha serie de cadeas e devolve unha cadea, ou o valor NULL se algún argumento é NULL. Sintaxe:

CONCAT(cadea1, cadea2 [,cadea3] ...)

* POSITION

Busca unha subcadea dentro dunha cadea e devolve a posición na que se atopa a primeira aparición. Sintaxe:

POSITION(subcadea IN cadea)

* LOCATE

Busca unha subcadea dentro dunha cadea e devolve a posición na que se atopa a primeira aparición. Pódese pasar un terceiro parámetro indicando a posición a partir da cal se fai a busca. Sintaxe:

LOCATE(subcadea, cadea [, posición])

* LEFT

Devolve os n primeiros carácteres da cadea, empezando a ler pola esquerda. Sintaxe:

LEFT(cadea, n)

* RIGHT

Devolve os n primeiros carácteres da cadea, empezando a ler pola dereita.Sintaxe:

RIGHT(cadea, n)

* LENGTH

Devolve a lonxitude da cadea en carácteres. Sintaxe:

LENGTH(cadea)

* LOWER

Pasa a minúsculas todos os carácteres da cadea. Sintaxe:

LOWER(cadea)

* UPPER

Pasa a maiúsculas todos os carácteres da cadea. Sintaxe:

UPPER(cadea)

* LTRIM

Elimina os espazos situados á esquerda na cadea. Sintaxe:

LTRIM(cadea)

* RTRIM

Elimina os espazos situados á dereita na cadea. Sintaxe:

RTRIM(cadea)

* TRIM

Elimina os espazos situados á esquerda e á dereita na cadea. Sintaxe:

TRIM(cadea)

* REPEAT

Repite unha cadea n veces. Sintaxe:

REPEAT(cadea, n)

* REPLACE

Substitúe unha subadea por outra dentro dunha cadea. Sintaxe:

REPLACE(cadea, subcadea\_inicial, subcadea\_final)

* SPACE

Devolve unha cadea composta de n espazos en branco. Sintaxe:

SPACE(n)

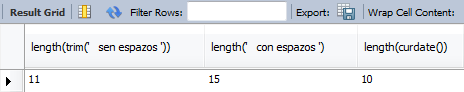
* SUBSTRING

Devolve unha subcadea de lonxitude especificada, empezando dende a posición elixida. Se non se indica lonxitude, se extrae dende a posición indicada ata o final. Sintaxe:

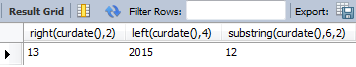
SUBSTRING(cadea, posición, [lonxitude])

Exemplos considerando que hoxe é 13/12/2015:

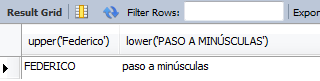
**select** **length(trim(**' sen espazos '**)),** **length(**' con espazos '**), length(curdate());**



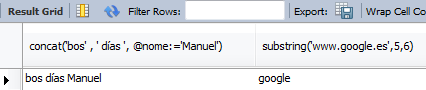
**select right(curdate(),**2**),** **left(curdate(),**4**),** **substring(curdate(),**6**,**2**);**



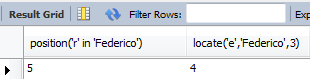
**select** **upper(**'Federico'**),** **lower(**'PASO A MINÚSCULAS'**)** **;**



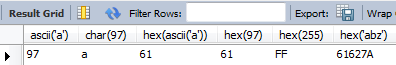
**select** **concat(**'bos' **,** ' días '**,** @nome**:=**'Manuel'**), substring(**'www.google.es'**,**5**,**6**);**



**select** **position(**'r' **in** 'Federico'**),** **locate(**'e'**,**'Federico'**,**3**);**



**select** **ascii(**'a'**),char(**97**),hex(ascii(**'a'**)),hex(**97**),hex(**255**),** **hex(**'abz'**);**



* + - 1. Funcións numéricas
* ABS

Obtén o valor absoluto dun número. Cando se pasa como parámetro un valor que non é numérico devolve o valor 0. Sintaxe:

ABS(número)

* SIGN

Devolve o valor -1 se o número que se pasa como parámetro é negativo, 1 se é positivo e 0 se o parámetro non é un número, ou é o número 0. Sintaxe:

SIGN(número)

* SQRT

Devolve a raíz cadrada do número. Sintaxe:

SQRT(número)

* POWER

Calcula potencias de número. Devolve o resultado de elevar o *número1* á potencia de *número2*. Sintaxe:

POWER(número1, número2)

* MOD

Devolve o resto da división enteira de dous números que se pasan como parámetros. Sintaxe:

MOD(número1, número2)

Outros operadores para obter o resto da división enteira: número1%número2.

* ROUND

Redondea un número decimal ao enteiro máis próximo. Pódese utilizar un segundo argumento para indicar o número de decimais cos que debe facer o redondeo. Cando *decimais* ten un valor negativo, converte en ceros ese número de díxitos contando dende o punto decimal á esquerda e fai o redondeo no seguinte díxito á esquerda. Sintaxe:

ROUND(número [,decimais])

* CEILING ou CEIL

Devolve o menor valor enteiro, non menor có número que se pasa como parámetro. Sintaxe:

CEIL(número)

CEILING(número)

* TRUNCATE

Retorna o número truncado co número de decimais que se pasan como parámetro. Cando *decimais* ten un valor negativo, converte en ceros ese número de díxitos contando dende o punto decimal á esquerda. Sintaxe:

TRUNCATE(número, decimais)

* FLOOR

Devolve o número enteiro máis grande que sexa maior có número que se pasa como parámetro. Sintaxe:

FLOOR(número)

* RAND

devolve un valor de coma flotante aleatorio no rango 0 <= valor <= 1.0. Sintaxe:

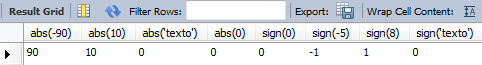
RAND()

RAND(número)

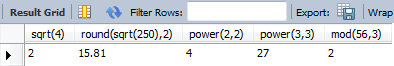
Cando se pasa un número como parámetro produce unha secuencia repetible sempre que se pase ese mesmo número como parámetro.

Exemplos:

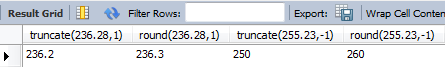
**select** **abs(-**90**),** **abs(**10**),** **abs(**'texto'**),abs(**0**),** **sign(**0**),sign(-**5**),sign(**8**),sign(**'texto'**);**



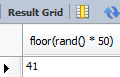
**select** **sqrt(**4**),** **round(sqrt(**250**),**2**),power(**2**,**2**),power(**3**,**3**),** **mod(**56**,**3**);**



**select** **truncate(**236.28**,**1**),** **round(**236.28**,**1**),truncate(**255.23**,-**1**),** **round(**255.23**,-**1**);**



**select** **floor(rand()** **\*** 50**); # xera un número de 0 a 49.**



**select** dni**,rand(**5**)** **from** empregado**;**

u501_lista_execucionRAND5.emf

**select** dni**,rand()** **from** empregado**;**

u501_lista_execucionRAND.emf

* + - 1. Funcións de agrupamento ou de columna

Permiten facer cálculos coas columnas da táboa de resultados dunha consulta. Cando se utilizan na lista de selección, non poden ir acompañadas de ningunha outra expresión que non conteña unha función deste tipo. Só poden ir acompañadas de nomes de columnas ou de expresións que conteñan nomes de columnas cando se agrupan filas utilizando a cláusula GROUP BY, que se verá nunha actividade posterior.

* COUNT

Conta o número de liñas resultantes da consulta, ou o número de valores distintos que toma unha expresión (normalmente, unha columna). Sintaxe:

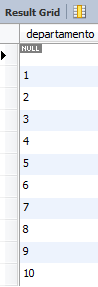
COUNT(\*)

COUNT([DISTINCT] expresión)

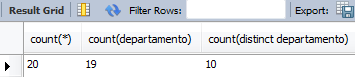
* Cando se utiliza o símbolo asterisco ('\*') como parámetro, devolve o número de filas que ten a táboa de resultados.
* Cando se pon como parámetro 'DISTINCT expresión', devolve o número de valores distintos que toma a expresión nas filas da táboa de resultados, sen ter en conta as que toman o valor NULL.
* Cando se pon como parámetro só unha 'expresión', devolve o número de filas da táboa de resultados en que a expresión toma un valor distinto de NULL.

Exemplos:

**select** **distinct** departamento **from** empregado**;**



**select** **count(\*), count(**departamento**), count(distinct** departamento**)** **from** empregado**;**



O resultados destas consultas informan que hai 20 empregados, 19 empregados cun valor distinto de NULL na columna departamento (hai un empregado que non ten departamento asignado) e hai empregados en 10 departamentos distintos numerados do 1 ao 10.

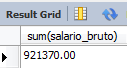
* SUM

Suma os valores que toma a expresión (normalmente, unha columna) especificada, para todas a filas resultantes da consulta. A opción DISTINCT non ten en conta os valores repetidos da columna. Sintaxe:

SUM ([DISTINCT] expresión)

Exemplo: Calcular o que se gasta en salarios. Suma dos salarios brutos.

**select** **sum(**salario\_bruto**)** **from** empregado**;**



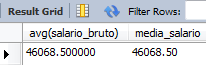
* AVG

Calcula a media dos valores que toma a expresión nas filas da táboa de resultados. A opción DISTINCT non ten en conta os valores repetidos da columna. Sintaxe:

AVG ([DISTINCT] expresión)

Exemplo: Calcular a media dos salarios brutos dos empregados.

**select** **avg(**salario\_bruto**),** **round(avg(**salario\_bruto**),**2**)** **as** media\_salario **from** empregado**;**



* MAX

Devolve o valor máis alto que toma a expresión nas filas da táboa de resultados. Sintaxe:

MAX (expresión)

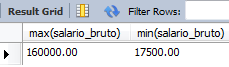
* MIN

Devolve o valor máis baixo da expresión nas filas da táboa de resultados. Sintaxe:

MIN (expresión)

Exemplo: Seleccionar o salario máis alto e o máis baixo de todos os empregados.

**select** **max(**salario\_bruto**),** **min(**salario\_bruto**)** **from** empregado**;**



Valores NULL e as funciones de agrupamento:

O estándar ANSI/ISO establece unhas regras para o manexo de valores NULL nas funcións de columna:

* Se algún dos valores dunha columna é NULL, non se ten en conta ao facer os cálculos nunha función de agrupamento.
* Se o valor de todas as columnas é NULL, as funcións SUM, AVG, MAX e MIN devolven o valor NULL, e a función COUNT o valor cero.
* COUNT(\*) conta filas e non depende da presenza de valores NULL nas columnas. Cando se quere ter en conta os valores NULL dunha columna, pódese utilizar a fórmula COUNT(expresión) que conta as filas nas que expresión non é NULL.
  + - 1. Outras funcións

A continuación móstranse de forma moi resumida grupos de funcións que MySQL incorpora que son utilizados con menos frecuencia pero que poden ser de utilidade.

Funcións de control de fluxo

* IF

Examina a *expresión1*; e se é verdadeira (*expresión1*<>0 e *expresión1*<>NULL) entón retorna *expresión2*; se é falsa, retorna *expresión3*. É posible aniñar condicións, de modo que *expresión2* e *expresión3* poden conter funcións IF. Sintaxe:

IF(expresión1, expresión2, expresión3)

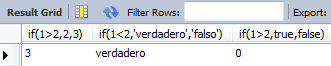
* IFNULL

Cando a expresión1 toma o valor NULL, a función devolve o contido de expresión2, noutro caso devolve o contido de expresión1. Sintaxe:

IFNULL(expresión1, expresión2)

Exemplos:

**select** **if(**1**>**2**,**2**,**3**),** **if(**1**<**2**,**'verdadero'**,**'falso'**),if(**1**>**2**,true,false);**



**select** apelidos**,** nome**,** **ifnull(**departamento**,**'Sen departamento'**)**

**from** empregado

**order** **by** departamento**;**





Funcións de información do sistema

* USER

Mostra información do usuario que fixo a conexión (nome e host). Sintaxe:

USER()

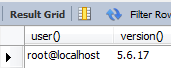
* VERSION

Mostra información da versión do servidor MySQL. Sintaxe:

VERSION()

Exemplo:

**select** **user(),** **version();**



Funcións de cifrado

As funcións de cifrado ou funcións *'hash'* utilízanse para enmascarar información que se desexa ocultar. Unha función *'hash'* é un algoritmo que transforma un conxunto de datos, como pode ser un ficheiro de texto ou unha cadea de carácteres, nun único valor de lonxitude fixa (*'hash'*). O valor *'hash'* calculado pode ser utilizado para verificar a integridade de copias dun dato orixinal sen necesidade de facilitar o dato orixinal. O proceso de cifrado é practicamente irreversible, polo que un valor *'hash'* pode ser libremente distribuído ou almacenado e só se utiliza para fins de comparación. Traballando con bases de datos, o uso máis corrente é para gardar contrasinais.

* PASSWORD

É a función que usa MySQL para cifrar as contrasinais dos usuarios que acceden ao servidor e que se almacenan na táboa *user* da base de datos *mysql*. Devolve unha cadea de 41 díxitos en hexadecimal. Sintaxe:

PASSWORD(cadea)

* MD5

Calcula unha suma de verificación (checksum) MD5 de 128-bit para unha cadea. Devolve unha cadea de 32 díxitos en hexadecimal.Sintaxe:

MD5(cadea)

* SHA1 (Secure Hash Algorithm)

Calcula unha suma de verificación (checksum) SHA1 de 160-bit para a cadea. O valor se devolve como unha cadea de 40 díxitos en hexadecimal, ou NULL, no caso de que o parámetro que se pasa tivera o valor NULL. Pode considerarse un equivalente a MD5(), criptográficamente máis seguro. Sintaxe:

SHA1(cadea)

* SHA2

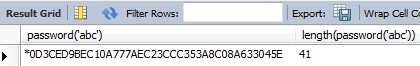
É un conxunto de funcións *'hash'* criptográficas (SHA-224, SHA-256, SHA-384, SHA-512) deseñadas pola 'Agencia de Seguridad Nacional' (NSA). SHA-2 inclúe un importante número de cambios respecto a súa predecesora, SHA-1, que a fai máis segura. Devolve unha cadea en hexadecimal de lonxitude 56, 64, 96, ou 128 carácteres, dependendo da función utilizada. Sintaxe:

SHA2(cadea, lonxitude\_resultado)

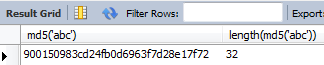
O primeiro parámetro que se pasa é unha cadea de texto plano, e o segundo indica o tipo de función que se vai a aplicar, e pode tomar os valores 224, 256, 384, 512, ou 0 (que é equivalente a 256). Se algún dos parámetros que se lle pasa toma o valor NULL ou o valor que se pasa como segundo parámetro non é un dos valores permitidos, devolve o valor NULL.

Exemplos:

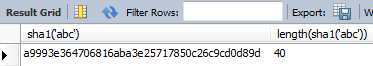
**select** **password(**'abc'**),** **length(password(**'abc'**));**



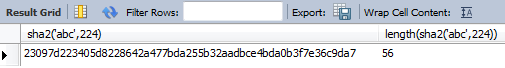
**select** **md5(**'abc'**),** **length(md5(**'abc'**));**



**select** **sha1(**'abc'**),** **length(sha1(**'abc'**));**



**select** **sha2(**'abc'**,**224**),length(sha2(**'abc'**,**224**));**



**select** **sha2(**'abc'**,**384**),length(sha2(**'abc'**,**384**));**

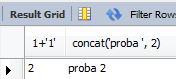


Funcións de conversión de tipos

Unha expresión ou unha comparación, poden ter operandos de distintos tipos que terán que converterse en tipos compatibles para que se resolva. Algunhas conversións fanse de forma implícita, como por exemplo, a conversión automática de números en cadeas, e viceversa, que fai MySQL.

Exemplo:

**select** 1+'1', **concat(**'proba '**,** 2**);**



As funcións CAST e CONVERT permiten facer conversións explícitas dun tipo de dato a outro, indicando o tipo de dato no que se quere converter. Con estas funcións soluciónase de forma fácil un problema que pode chegar a ser un quebracabezas, sobre todo nas comparacións. Os tipos de datos que se poden utilizar para a conversión son:

* BINARY[(tamaño)]
* CHAR[(tamaño)]
* DATE
* DATETIME
* DECIMAL[(tamaño[,decimais])]
* SIGNED [INTEGER]
* TIME
* UNSIGNED [INTEGER]

As funcións CAST e CONVERT que incorpora MySQL, utilizan a sintaxe SQL estándar.

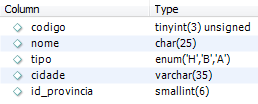
* CAST

Devolve o valor que se pasa como primeiro parámetro convertido ao tipo que se indica na función. Sintaxe:

CAST(expresión AS tipo\_dato)

As funcións de conversión son útiles, por exemplo, para ordenar os resultados por columnas de tipo ENUM por orden alfabético. Normalmente cando se ordena por unha columna ENUM, utilízanse os valores numéricos do orden interno. Facendo unha conversión a tipo CHAR pódese facer a ordenación por orden alfabético.

Exemplo: seleccionar *codigo*, *nome* e *tipo* de todos os departamentos, ordenando o resultado alfabeticamente pola columna *tipo*, que é de tipo ENUM como se pode ver na descrición do esquema da táboa.

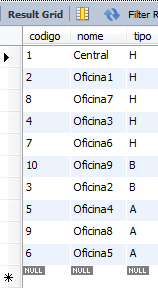


A primeira solución ordena polo tipo seguindo o número de orde dos valores válidos (H, B, A).

**select** codigo**,** nome**,** tipo

**from** departamento

**order** **by** tipo**;**

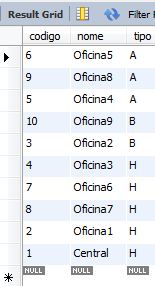


A segunda solución ordena alfabeticamente polo valor do tipo transformado en carácter (A, B, H).

**select** codigo**,** nome**,** tipo

**from** departamento

**order** **by** **cast(**tipo **as** **char);**



* CONVERT

Transforma unha expresión nun tipo de dato indicado como parámetro. Sintaxe:

CONVERT(expresión, tipo)

CONVERT(expresión USING nome\_xogo\_carácteres)

* Cando se utiliza a primeira forma da sintaxe, funciona igual que a función CAST, converte a expresión ao tipo de datos que se pasa como segundo parámetro.
* A segunda forma da sintaxe, con USING, utilízase para converter datos entre diferentes xogos de carácteres.

Exemplo: Converter a cadea 'abc' ao xogo de carácteres utf8:

**select** **convert(**'abc' **using** utf8**);**

* Tarefa 2. Escribir e probar consultas simples que utilizan funcións.
  1. Tarefas

As tarefas propostas son as seguintes:

* Tarefa 1. Escribir e probar consultas simples.
* Tarefa 2. Escribir e probar consultas simples que utilizan funcións.
  + 1. Tarefa 1. Escribir e probar consultas simples

A tarefa consiste en realizar as seguintes consultas simples.

Sobre a base de datos tendaBD

* Tarefa 1.1. Mostrar os datos de todas as tendas.
* Tarefa 1.2. Mostrar os nomes de todos os provedores.
* Tarefa 1.3. Obter a lista das poboacións nas que existen clientes.
* Tarefa 1.4. Mostrar o prezo de venda de todos os artigos e o prezo que resultaría despois de aplicarlles un incremento do 10%.
* Tarefa 1.5. Mostrar o número de cliente, apelidos e nome de todos os clientes de Madrid.
* Tarefa 1.6. Seleccionar o código, descrición e peso dos artigos que pesen máis de 500 gramos.
* Tarefa 1.7. Seleccionar todos os artigos que teñan prezo de venda superior ou igual ao dobre do prezo de compra.
* Tarefa 1.8. Seleccionar apelidos, nome, poboación e desconto, de todos clientes de Asturias ou Valencia que teñan un desconto superior ao 2% ou que non teñan desconto.
* Tarefa 1.9. Seleccionar todos os artigos de cor negra que pesen máis de 5000 gramos.
* Tarefa 1.10. Obter todos os artigos que non son de cor negra ou que teñan un peso menor ou igual de 5000 gramos, é dicir, obter o resultado complementario da consulta anterior.
* Tarefa 1.11. Seleccionar os artigos que son de cor negra e pesan máis de 100 gramos, ou ben son de cor cyan.
* Tarefa 1.12. Facer unha lista dos artigos que teñan un prezo de compra entre 12 e 18 euros, ambos prezos incluídos.
* Tarefa 1.13. Mostrar unha lista de artigos de cor negra ou de cor *cyan*.
* Tarefa 1.14. Buscar un cliente do que se descoñece o apelido exacto, pero se sabe que as dúas primeiras letras son 'RO'.
* Tarefa 1.15. Buscar clientes que teñan o nome de 5 letras, empezando por 'B' e terminando por 'A'.
* Tarefa 1.16. Buscar todos os artigos para os que non se gravou o seu color.
* Tarefa 1.17. Clasificar os artigos tendo en conta o seu peso, por orden decrecente.
* Tarefa 1.18. Mostrar código de artigo, nome, prezo de compra, prezo de venda e marxe de beneficio (prezo de venda – prezo de compra) dos artigos que teñen un prezo de compra superior a 3000 euros, ordenados pola marxe.
* Tarefa 1.19. Clasificar nome, provedor, stock e peso dos artigos que teñen un peso menor ou igual de 1000 gramos, por orden crecente do provedor. Cando os provedores coincidan, deben clasificarse polo stock en orden decrecente.
* Tarefa 1.20. Seleccionar nome e apelidos dos clientes que teñan un apelido que empece por 'F' e remate por 'Z'.
* Tarefa 1.21. Seleccionar todos os artigos que leven a palabra LED, en maiúsculas, na súa descrición.
* Tarefa 1.22. Seleccionar todos os artigos que teñan unha descrición que empece por 'CABI', sen diferenciar maiúsculas de minúsculas.
* Tarefa 1.23. Comprobar que un número é un valor enteiro, que pode empezar polos símbolos + ou -.
* Tarefa 1.24. Seleccionar os clientes que teñan un apelido que empece pola letra 'a' ou pola letra 'f'.
* Tarefa 1.25. Seleccionar os clientes que teñan un apelido que non empece por 'a','b','c', ou 'd'.
* Tarefa 1.26. Seleccionar os artigos que teñan un prezo de venta que remata en .00.
* Tarefa 1.27. Seleccionar os clientes que teñen un nome que teña exactamente 5 carácteres.

Sobre a base de datos traballadores

* Tarefa 1.28. Unha nova normativa non permite que as comisións superen o 10% do salario. Mostrar información ordenada por número de departamento, dos departamentos nos que exista algún empregado que incumpra esta normativa, nos seguintes casos:
* Mostrar número de departamento.
* Mostrar ademais nome de departamento utilizando *join* e sen utilizar *join*.
* Mostrar ademais nome do xefe de departamento.
* Tarefa 1.29. A campaña de axuda familiar posta en marcha pola empresa, establece que os empregados que teñan máis de 3 fillos, cobrarán unha paga extra de 30 euros por fillo a partir do terceiro e incluíndo este. Mostrar nome, salario, comisión, número de fillos, importe da paga extra e salario mensual final dos empregados, ordenados alfabeticamente polo nome, aplicando a axuda familiar.
* Tarefa 1.30. Mostrar os nomes dos empregados con apelido 'MORA' ou que empece por 'MORA' ordenados alfabeticamente.
* Tarefa 1.31. A empresa vai organizar un espectáculo para os fillos dos empregados que durará dous días. O primeiro día invitarase aos empregados con apelido que empece polas letras dende a 'A' ata a 'L', ambas inclusive. O segundo día invitarase ao resto dos empregados. Cada empregado recibirá unha invitación por fillo e dúas máis. Cada fillo recibirá un regalo durante o espectáculo. Mostrar unha lista ordenada alfabeticamente polo nome do empregado na que aparezan os nomes dos empregados que se invitarán no primeiro día, o número de invitación que lle corresponden e o número de regalos que hai que preparar para el.
* Tarefa 1.32. Mostrar os nomes e salarios dos empregados que cumpran algunha das seguintes condicións:
* Non teñen fillos e gañan máis de 1200 euros.
* Teñen fillos e gañan menos de 1800 euros.
* Tarefa 1.33. Mostrar nome e salario base dos empregados que non teñen fillos ordenados de maior a menor polo salario base.
* Tarefa 1.34. Mostrar por orden alfabético, os nomes e salarios base dos empregados que traballen no departamento 111 e teñan unha comisión que supere o 15% do seu salario base.
* Tarefa 1.35. A empresa decide aumentar a comisión nun 15% aos empregados que teñan máis de 2 fillos. Mostrar ordenados alfabeticamente polo nome de empregado: nome do empregado, número de fillos, importe da comisión antes do aumento e importe da comisión despois do aumento.

Solución

* Tarefa 1.1

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Mostrar os datos de todas as tendas

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** **\***

**from** tendas**;**

/\* Outra solución poñendo a relación de columnas na lista de selección \*/

**select** tda\_id **as** Identificador**,**

tda\_poboacion **as** Poboacion**,**

tda\_xerente **as** Xerente

**from** tendas**;**

* Tarefa 1.2

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Mostrar os nomes de todos os provedores

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** prv\_nome **as** Nome

**from** provedores**;**

* Tarefa 1.3

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Obter a lista das poboacións nas que existen clientes

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** **distinct** clt\_poboacion

**from** clientes**;**

* Tarefa 1.4

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Mostrar o prezo de venta de todos os artigos e o prezo que resultaría despois de

aplicarlles un incremento do 10%

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** art\_nome **as** Nome**,**

art\_pv **as** Prezo\_venta**,**

art\_pv**\***1.1 **as** Prezo\_final

**from** artigos**;**

* Tarefa 1.5

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Mostrar o número de cliente, apelidos e nome de todos os clientes de Madrid

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** clt\_id**,** clt\_apelidos**,** clt\_nome

**from** clientes

**where** clt\_poboacion**=**'MADRID'**;**

* Tarefa 1.6

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Seleccionar o código, descrición e peso dos artigos que pesen máis de 500 gramos

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** art\_codigo**,** art\_nome**,** art\_peso

**from** artigos

**where** art\_peso**>**500**;**

* Tarefa 1.7

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Seleccionar todos os artigos que teñan prezo de venta superior ou igual

ao dobre do prezo de compra

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** **\***

**from** artigos

**where** art\_pv**>=**2**\***art\_pc**;**

* Tarefa 1.8

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Seleccionar apelidos, nome, poboación e desconto, de todos clientes de Asturias ou

Valencia que teñan un desconto superior ao 2% ou que non teñan desconto

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** clt\_apelidos **as** Apelidos**,**

clt\_nome **as** Nome**,**

clt\_poboacion **as** Poboacion**,**

clt\_desconto **as** Desconto

**from** clientes

**where** clt\_poboacion in ('Asturias', 'Valencia')

**and** (clt\_desconto = 0 **or** clt\_desconto**>**2)**;**

* Tarefa 1.9

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Seleccionar todos os artigos de cor negra que pesen máis de 5000 gramos

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** art\_codigo **as** Codigo**,**

art\_nome **as** Nome**,**

art\_peso **as** Peso**,**

art\_color **as** Color

**from** artigos

**where** art\_color**=**'NEGRO' **and** art\_peso**>**5000**;**

* Tarefa 1.10

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Obter todos os artigos que non son de cor negra ou que teñan un peso menor ou igual de

5000 gramos, é dicir, obter o resultado complementario da consulta anterior

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** art\_codigo **as** Codigo**,**

art\_nome **as** Nome**,**

art\_peso **as** Peso**,**

art\_color **as** Color

**from** artigos

**where** **(**art\_color!**=**'negro' **or** art\_peso**<=**5000**)**

**or** art\_color **is** **null;** # ollo cos valores null

/\* Outra solución coa negación nunha condición:\*/

**select** art\_codigo **as** Codigo**,**

art\_nome **as** Nome**,**

art\_peso **as** Peso**,**

art\_color **as** Color

**from** artigos

**where** **not** **(**art\_color**=**'negro' **and** art\_peso**>**5000**)**

**or** art\_color **is** **null;** # ollo cos valores null

* Tarefa 1.11

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Seleccionar os artigos que son de cor negra e pesan máis de 100 gramos,

ou ben son de color cyan

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** art\_codigo **as** Codigo**,**

art\_nome **as** Nome**,**

art\_peso **as** Peso**,**

art\_color **as** Color

**from** artigos

**where** **(**art\_color**=**'negro' **and** art\_peso**>**100**)** **or** art\_color**=**'cyan'**;**

* Tarefa 1.12

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Facer unha lista dos artigos que teñan un prezo de compra entre 12 e 18 euros,

ambos prezos incluídos

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** art\_codigo **as** Codigo**,**

art\_nome **as** Nome**,**

art\_pc **as** Prezo\_compra

**from** artigos

**where** art\_pc **between** 12 **and** 18**;**

* Tarefa 1.13

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Mostrar unha lista de artigos de cor negra ou de cor cyan

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** art\_codigo **as** Codigo**,**

art\_nome **as** Nome**,**

art\_color **as** Color

**from** artigos

**where** art\_color **in** **(**'negro'**,**'cyan'**);**

* Tarefa 1.14

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Buscar un cliente do que se descoñece o apelido exacto, pero se sabe que as

dúas primeiras letras son 'RO'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** **\***

**from** clientes

**where** clt\_apelidos **like** 'RO%'**;**

* Tarefa 1.15

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Buscar clientes que teñan un nome de 5 letras, empezando por 'B' e terminando por 'A'.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** clt\_nome**,**

clt\_apelidos

**from** clientes

**where** clt\_nome **like** 'B\_\_\_A'**;**

* Tarefa 1.16

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Buscar todos os artigos para os que non se gravou o seu color

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** **\***

**from** artigos

**where** art\_color **is** **null;**

* Tarefa 1.17

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Clasificar os artigos tendo en conta o seu peso, por orden decrecente

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** art\_codigo **as** Numero**,**

art\_nome **as** Artigo**,**

art\_peso **as** Peso

**from** artigos

**order** **by** peso **desc;**

* Tarefa 1.18

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Mostrar código de artigo, nome, prezo de compra, prezo de venda e marxe de

beneficio (prezo de venda – prezo de compra) dos artigos que teñen un prezo de

compra superior a 3000 euros, ordenados pola marxe.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** art\_codigo as Codigo**,**

art\_nome **as** Nome**,**

art\_pc **as** PrezoCompra**,**

art\_pv **as** PrezoVenta**,**

art\_pv**-**art\_pc **as** Marxe

**from** artigos

**where** art\_pc**>**3000

**order** **by** Marxe**;**

* Tarefa 1.19

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Clasificar nome, provedor, stock e peso dos artigos que teñen un peso menor ou igual

de 1000 gramos, por orden crecente do provedor. Cando os provedores coincidan, deben

clasificarse polo stock en orden decrecente

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** art\_nome**,** art\_provedor**,** art\_stock**,** art\_peso

**from** artigos

**where** art\_peso**<=**1000

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

* Tarefa 1.20

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Seleccionar nome e apelidos dos clientes que teñan un apelido que

empece por 'F' e remate por 'Z'.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** **\***

**from** clientes

**where** clt\_apelidos regexp '^F.\*Z$'**;**

* Tarefa 1.21

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Seleccionar todos os artigos que leven a palabra LED, en maiúsculas, na súa

descrición.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** **\***

**from** artigos

**where** art\_nome REGEXP **BINARY** 'LED'**;**

* Tarefa 1.22

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Seleccionar todos os artigos que teñan unha descrición que empece por 'CABI',

sen diferenciar maiúsculas de minúsculas.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** **\***

**from** artigos

**where** art\_nomE REGEXP '^CABI'**;**

* Tarefa 1.23

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Comprobar que un número é un valor enteiro, que pode empezar polos símbolos + ou -.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** **-**125

regexp '^[+-]?[0-9]+'**;**

* Tarefa 1.24

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Seleccionar os clientes que teñan un apelido que empece pola letra 'a' ou 'f'.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** **\***

**from** clientes

**where** clt\_apelidos REGEXP '^[af]'**;**

* Tarefa 1.25

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Seleccionar os clientes que teñan un apelido que non empece por 'a','b','c', ou 'd'.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** **\***

**from** clientes

**where** clt\_apelidos REGEXP '^[^a-d]'**;**

* Tarefa 1.26

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Seleccionar os artigos que teñan un prezo de venta que remata en .00.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** **\***

**from** artigos

**where** art\_pv regexp '\\.0{2}$'**;**

# ou ben:

**select** **\***

**from** artigos

**where** art\_pv regexp '^[+-]?[0-9]+(\\.0{2})$'**;**

* Tarefa 1.27

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Seleccionar os clientes que teñen un nome que teña exactamente 5 carácteres.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** **\***

**from** clientes

**where** clt\_nome regexp '^.{5}$'**;** # o nome ten 5 carácteres, nin un máis nin un menos

* Tarefa 1.28

/\* Unha nova normativa non permite que as comisións superen o 10% do salario. Mostrar

os códigos dos departamentos nos que exista algún empregado que incumpra esta

normativa.\*/

**select** **distinct** empDepartamento **as** Departamento

**from** empregado

**where** empComision**>**0.10**\***empSalario

**order** **by** empDepartamento **;**

* Tarefa 1.29

/\* A campaña de axuda familiar posta en marcha pola empresa, establece que os

empregados que teñan máis de 3 fillos, cobrarán unha paga extra de 30 euros por fillo

a partir do terceiro e incluíndo este. Mostrar nome, salario, comisión, número de

fillos, importe da paga extra e salario mensual final dos empregados, ordenados

alfabeticamente polo nome, aplicando a axuda familiar. \*/

**select** empNome **as** Nome**,**

mpComision **as** Comision**,**

empSalario **as** Salario**,**

empFillos **as** Fillos**,**

**(**empFillos**-**2**)\***30 **as** Extra**,**

empSalario**+ifnull(**empComision**,**0**)+(**empFillos**-**2**)\***30 **as** Total

**from** empregado

**where** empFillos**>=**3

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

* Tarefa 1.30

/\*Mostrar os nomes dos empregados con apelido 'MORA' ou que empece por 'MORA' ordenados

alfabeticamente.\*/

**select** empNome **as** Empregado

**from** empregado

**where** empNome **like** 'MORA%'

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

* Tarefa 1.31

/\*Mostrar unha lista ordenada alfabeticamente polo nome do empregado na que aparezan os

nomes dos empregados que se invitarán no primeiro día, o número de invitación que lle

corresponden e o número de regalos que hai que preparar para el.\*/

**select** empNome **as** Empregado**,**

empFillos**+**2 **as** Invitacions**,**

empFillos **as** Regalos

**from** empregado

**where** empFillos**>**0 **and** empNome **between** 'A' **and** 'M'

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

/\* outras alternativas para poñer a condición que afecta ao nome:

a)substring(empNome,1,1) between 'a' and 'l'

b)left(empNome,1) between 'a' and 'l'

c)empNome regexp '^[a-l]' \*/

* Tarefa 1.32

/\*Mostrar os nomes e salarios dos empregados que cumpran algunha das seguintes

condicións:

– Non teñen fillos e gañan máis de 1200 euros.

– Teñen fillos e gañan menos de 1800 euros.

\*/

**select** empNome **as** Empregado**,**

empFillos **as** Fillos**,**

empSalario **as** Salario

**from** empregado

**where** **(**empFillos**=**0 **and** empSalario**>**1200**) or** **(**empFillos**>**0 **and** empSalario**<**1800**)**

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

* Tarefa 1.33

/\*Mostrar nome e salario base dos empregados que non teñen fillos, ordenados de maior

a menor polo salario base.\*/

**select** empNome **as** Empregado**,**

empSalario **as** Salario

**from** empregado

**where** empFillos **=** 0

**order** **by** empSalario **desc;**

* Tarefa 1.34

/\*Mostrar por orden alfabético, os nomes e salarios dos empregados que traballen no

departamento 111 e teñan unha comisión que supere o 15% do seu salario base.\*/

**select** empNome **as** Empregado**,**

empSalario **as** Salario**,**

empComision **as** Comision

**from** empregado

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

**and** empComision **is** **not** **null**

**and** empComision**>**empSalario**\***0.15

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

* Tarefa 1.35

/\*A empresa decide aumentar a comisión nun 15% aos empregados que teñan máis de 2

fillos. Mostrar ordenados alfabeticamente polo nome de empregado: nome do empregado,

número de fillos, importe da comisión antes do aumento e importe da comisión despois do

aumento.\*/

**select** empNome **as** Empregado**,**

empFillos **as** fillos**,**

empComision **as** Comision\_inicial**,**

empComision**\***1.15 **as** Comision\_final

**from** empregado

**where** empFillos**>**2 **and** empComision **is** **not** **null**

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

* + 1. Tarefa 2. Escribir e probar consultas simples que utilizan funcións

A tarefa consiste en realizar as seguintes consultas utilizando funcións MySQL.

Sobre a base de datos tendaBD

* Tarefa 2.1. Mostrar apelidos e nome nunha mesma columna separados por unha coma, e o número de letras que ten o nome.
* Tarefa 2.2. Mostrar nomes e apelidos dos clientes en minúscula.
* Tarefa 2.3. Mostrar a idade dunha persoa nacida o 22/03/1981 no momento actual. Suponse que a data do sistema é a correcta.
* Tarefa 2.4. Mostrar tódalas vendas do mes anterior ao mes actual en dous supostos:
* Na táboa gárdanse datos das vendas dun ano.
* Na táboa gárdanse datos das vendas de varios anos.

Pódense inserir vendas para poder comprobar o resultado. Se o data actual é 13/12/2015, pódense inserir ventas no mes de novembro utilizando o seguinte guión SQL:

/\* inserir vendas no mes anterior tendo en conta que a data actual é 13/12/2015 \*/

**insert** **into** vendas **(**ven\_tenda**,** ven\_empregado**,** ven\_cliente**,** ven\_data**)** **values**

**(**7**,**2**,**1**,**'2015-11-01'**),**

**(**2**,**14**,**30**,**'2015-11-15'**),**

**(**2**,**14**,**60**,**'2015-11-30'**),**

**(**2**,**14**,**60**,**'2014-11-30'**);**

* Tarefa 2.5. Mostrar número, nome e prezo de venda (redondeado, sen decimais) dos artigos de cor negra.
* Tarefa 2.6. Calcular a media dos pesos de todos os artigos.
* Tarefa 2.7. Calcular a media do peso, o marxe máximo ( máxima diferenza entre o prezo de venda e o prezo de compra) e a diferenza que se dá entre o maior prezo de venda e o menor prezo de compra. Estes cálculos terán que facerse para aqueles artigos que teñan descrito a cor cun valor distinto do NULL.
* Tarefa 2.8. Contar o número de cores distintos que existen na táboa de artigos.
* Tarefa 2.9. Mostrar nome e cor dos artigos. Se a cor é descoñecida, débese mostrar o texto ‘DESCOÑECIDO’.

Sobre a base de datos traballadores

* Tarefa 2.10. A xubilación na empresa está establecida aos 60 anos. O empregado xubilado ten dereito a unha liquidación que equivale ao salario dun mes por cada ano de servizo na empresa. Mostrar nome, data de nacemento, salario mensual base, antigüidade (número de anos dende que entrou a traballar na empresa ata a data de xubilación) e importe da liquidación que lle corresponde aos empregados que se xubilarán no ano actual.
* Tarefa 2.11. Mostrar nome, día e mes do aniversario dos empregados dos departamentos 110 e 111.
* Tarefa 2.12. Mostrar o número de empregados que este ano cumpren 20 anos traballando na empresa e o salario medio de todos eles.
* Tarefa 2.13. Mostrar o importe anual (14 pagas) correspondente ao soldos dos empregados (soldo máis comisións).
* Tarefa 2.14. Mostrar o número de departamentos que existen e o presuposto anual medio de todos eles.
* Tarefa 2.15. Mostrar o importe total das comisións dos empregados.
* Tarefa 2.16. Mostrar nome, data de nacemento e idade dos empregados que teñan actualmente 50 anos ou máis. Ordenar o resultado pola idade de maior a menor.
* Tarefa 2.17. Mostrar nome de empregado, data de entrada na empresa con formato dd/mm/aaaa e número de trienios completos que levan traballados os empregados que cumpren 57 anos no ano actual. Ordenar de forma descendente por número de trienios.
* Tarefa 2.18.Mostrar a media de idade á que os empregados entran a traballar na empresa.
* Tarefa 2.19.Mostrar nome, data de entrada na empresa con formato dd/mm/aaaa e o número de anos completos que leva traballando na empresa, para os empregados que cumpren anos no mes actual.
* Tarefa 2.20.Mostrar a diferenza de días traballados entre o empregado máis antigo e o máis recente, indicando a data de ingreso de ambos traballadores.
* Tarefa 2.21. Mostrar ordenados por número de empregado, o número de empregado, nome e salario mensual total (salario+comisión) dos empregados cun salario mensual total de máis de 1800 euros.

Solución

* Tarefa 2.1

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Mostrar apelidos e nome nunha mesma columna separados por unha coma, e o número

de letras que ten o nome

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** **concat(**clt\_apelidos**,**', '**,**clt\_nome**)** **as** nome**,**

**length(**clt\_nome**)** **as** tamaño

**from** clientes**;**

* Tarefa 2.2

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Mostrar nomes e apelidos dos clientes en minúscula.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** **lower(**clt\_apelidos**)** **as** Apelidos**,**

**lower(**clt\_nome**)** **as** Nome

**from** clientes**;**

* Tarefa 2.3

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Mostrar a idade dunha persoa nacida o 22/03/1981 no momento actual.

Suponse que a data do sistema é a correcta.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** timestampdiff**(year,**'1981-03-22'**,**curdate**());**

/\* outra forma de solucionalo, empregando función de cadea de carácteres: \*/

**select** **year(**curdate**())-year(**'1981-03-22'**)-(right(**'1981-03-2'**,**5**)>right(**curdate**(),**5**));**

/\* outra forma de solucionalo, empregando variables: \*/

**select** @**data:=**'1981-03-22' **as** fecha\_nac**,**

curdate**()** **as** fecha\_act**,**

**year(**curdate**())-year(**@**data)-(right(**@**data,**5**)>right(**curdate**(),**5**))** **as** Idade**;**

* Tarefa 2.4

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Mostrar tódalas vendas do mes anterior ao mes actual en dous supostos:

- Na táboa gárdanse datos das vendas dun ano.

- Na táboa gárdanse datos das vendas de varios anos.

Débense inserir novas vendas se é necesario para poder comprobar o resultado.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* solución considerando que na táboa gárdanse datos das vendas dun ano \*/

**select** **\***

**from** vendas

**where** **month(subdate(**curdate**(),interval** 1 **month))** **=** **month(**ven\_data**);**

/\* solución considerando que na táboa gárdanse datos das vendas de varios anos \*/

**select** **\***

**from** vendas

**where** **month(subdate(**curdate**(),interval** 1 **month))** **=** **month(**ven\_data**)**

**and** **year(subdate(**curdate**(),interval** 1 **month))** **=** **year(**ven\_data**);**

* Tarefa 2.5

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Mostrar número, nome e prezo de venda (redondeado, sen decimais)

dos artigos de cor negra.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** art\_codigo**,** art\_color**,** art\_pv**,** **round(**art\_pv**)** **as** Prezo\_venta

**from** artigos

**where** art\_color**=**'negro'**;**

/\* outra forma de solucinalo \*/

**select** art\_codigo**,** art\_color**,** art\_pv**,** **format(**art\_pv**,**0**)** **as** Prezo\_venta

**from** artigos

**where** art\_color**=**'negro'**;**

* Tarefa 2.6

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Calcular a media dos pesos de todos os artigos

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** **round(avg(**art\_peso**),**2**)** **as** Media

**from** artigos**;**

/\* con máis información \*/

**select** **count(\*)** **as** Unidades**,**

**sum(**art\_peso**)** **as** Suma\_pesos**,**

**round(avg(**art\_peso**),**2**)** **as** Media

**from** artigos**;**

* Tarefa 2.7

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Calcular a media do peso, o marxe máximo ( máxima diferencia entre o prezo de

venta e o prezo de compra) e a diferencia que se da entre o maior prezo de venta e

o menor prezo de compra. Estes cálculos terán que facerse para aqueles artigos que

teñan na columna art\_col algún valor distinto do NULL.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** **round(avg(**art\_peso**),**2**)** **as** Media**,**

**max(**art\_pv**-**art\_pc**)** **as** Marxe\_maximo**,**

**max(**art\_pv**)-min(**art\_pc**)**

**from** artigos

**where** art\_color **is** **not** **null;**

* Tarefa 2.8

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Contar o número de cores distintas que existen na táboa de artigos

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** **count(distinct** art\_color**)**

**from** artigos**;**

* Tarefa 2.9

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Mostrar nome e cor dos artigos. Se a cor é descoñecida, débese mostrar o texto

‘DESCOÑECIDO’.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**select** art\_nome**,**

**ifnull(**art\_color**,** 'DESCOÑECIDO'**)** **as** color

**from** artigos**;**

* Tarefa 2.10

/\*Mostrar nome, data de nacemento, salario mensual, antigüidade (número de anos dende

que entrou a traballar na empresa ata a data de xubilación) e importe da liquidación

que lle corresponde aos empregados que se xubilan este ano, con 60 anos.\*/

**select** empNome **as** Nome**,**

empDataNacemento **as** Nacemento**,**

empSalario **as** Salario**,**

**year(**curdate**())** **-** **year(**empDataIngreso**)** **as** Antigüidade**,**

empSalario**\*(year(**curdate**())** **-** **year(**empDataIngreso**))** **as** Liquidacion

**from** empregado

**where** **year(**curdate**())** **-** **year(**empDataNacemento**)=**60

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

* Tarefa 2.11

/\*Mostrar nome, día e mes do aniversario dos empregados dos departamentos 110 e 111\*/

**select** empNome **as** Nome**,**

**month(**empDataNacemento**)** **as** Mes**,**

**day(**empDataNacemento**)** **as** Dia

**from** empregado

**where** empDepartamento**=**110 **or** empDepartamento**=**111

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

* Tarefa 2.12

/\*Mostrar o número de empregados que este ano cumpren 20 anos traballando na empresa e

o salario medio de todos eles.\*/

**select** **count(\*)** **as** Empregados**,**

**avg(**empSalario**)** **as** SalarioMedio

**from** empregado

**where** **year(**curdate**())-year(**empDataIngreso**)=**20**;**

* Tarefa 2.13

/\*Mostrar o importe anual (14 pagas) correspondente ao soldos dos empregados

(soldo + comisións).\*/

**select** **sum(**empSalario**+ifnull(**empComision**,**0**))\***14 **as** Importe

**from** empregado**;**

* Tarefa 2.14

/\*Mostrar o número de departamentos que existen e o presuposto anual medio de todos

eles.\*/

**select** **count(\*)** **as** Numero**,**

**avg(**depPresuposto**)** **as** Presuposto\_medio

**from** departamento**;**

* Tarefa 2.15

/\*Mostrar o importe total das comisións dos empregados\*/

**select** **sum(**empComision**)** **as** Comisions

**from** empregado**;**

* Tarefa 2.16

/\*Mostrar nome, data de nacemento e idade dos empregados que teñan actualmente 50 anos

ou máis. Ordenar o resultado pola idade de maior a menor.\*/

**select** empNome **as** Nome**,**

**date\_format(**empDataNacemento**,** '%d-%m-%Y'**)** **as** Nacemento**,**

**date\_format(**curdate**(),** '%d-%m-%Y'**)** **as** Hoxe**,**

timestampdiff**(year,**empDataNacemento**,**curdate**())** **as** Idade

**from** empregado

**where** timestampdiff**(year,**empDataNacemento**,**curdate**())** **>=** 50

**order** **by** Idade **desc;**

#Outra forma de solucionalo

**select** empNome **as** Nome**,**

**date\_format(**empDataNacemento**,** '%d-%m-%Y'**)** **as** Nacemento**,**

**date\_format(**curdate**(),** '%d-%m-%Y'**)** **as** Hoxe**,**

**(year(**curdate**())**-**year(**empDataNacemento**))-(right(**curdate**(),**5**)**

**<** r**ight(**empDataNacemento**,**5**))** **as** Idade

**from** empregado

**where** **(year(**curdate**())-year(**empDataNacemento**))-(right(**curdate**(),**5**)**

**<** r**ight(**empDataNacemento**,**5**))** **>=** 50

**order** **by** Idade **desc;**

* Tarefa 2.17

/\*Mostrar nome de empregado, data de entrada na empresa con formato dd/mm/aaaa e

número de trienios completos que levan traballados os empregados que cumpren 57 anos no

ano actual. Ordenar de forma descendente por número de trienios.\*/

**select** empNome **as** Empregado**,**

**date\_format(**empDataIngreso**,**'%d/%m/%Y'**)** **as** 'Data ingreso'**,**

**truncate((timestampdiff(year**,empDataIngreso,curdate**())/**3**,**0) **as** Trienios

**from** empregado

**where** **year(**curdate**())-year(**empDataNacemento**)** **=** 57

**order** **by** Trienios **desc;**

* Tarefa 2.18

/\*Mostrar a media de idade á que os empregados entran a traballar na empresa.\*/

**select** **round(avg(timestampdiff(year,**empDataNacemento,empDataIngreso**)),**0**) as** Media

**from** empregado**;**

/\* Outra solución utilizando funcións de cadeas de carácteres\*/

**select** **round(avg(year(**empDataIngreso**)-year(**empDataNacemento**)-**

**(right(**empDataIngreso**,**5**)<right(**empDataNacemento**,**5**))))** **as** Media

**from** empregado**;**

* Tarefa 2.19

/\*Mostrar nome, data de entrada na empresa con formato dd/mm/aaaa e o número de anos

completos que leva traballando na empresa, para os empregados que cumpren anos no mes

actual.\*/

**select** empNome **as** Empregado**,**

**date\_format(**empDataIngreso**,**'%d/%m/%Y'**)** **as** Ingreso**,**

**timestampdiff(year**,empDataIngreso,curdate**())as** Antigüidade

**from** empregado

**where** **month(**empDataNacemento**)=month(**curdate**());**

* Tarefa 2.20

/\*Mostrar a diferenza de días traballados entre o empregado máis antigo e o máis

recente, indicando a data de ingreso de ambos traballadores.\*/

**select** **min(**empDataIngreso **)** **as** Antigo**,**

**max(**empDataIngreso**)** **as** Recente**,**

**datediff(max(**empDataIngreso**),min(**empDataIngreso**))** **as** Diferenza

**from** empregado**;**

* Tarefa 2.21

/\*Mostrar ordenados por número de empregado, o número de empregado, nome e salario

mensual total (salario+comisión) dos empregados cun salario mensual total de máis de

1800 euros.\*/

**select** empNumero **as** Numero**,**

empNome **as** Nome**,**

empSalario **as** Salario\_base**,**

empComision **as** Comision**,**

empSalario**+ifnull(**empComision**,**0**)** **as** Salario\_total

**from** empregado

**where** empSalario**+ifnull(**empComision**,**0**)** **>** 1800

**order** **by** empNumero**;**

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\_V000501\_UD05\_A01\_ConsultasSimples\_Anexos que contén:

* O arquivo GuiaWorkbench.docx que é unha guía básica de MySQL Workbench 6.3.
* O arquivo V000501\_scriptsSQL.zip cos scripts necesarios para crear as bases de datos empregadas nos exemplos, tarefas e probas de avaliación.

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 |
| * CA5.1 -  Identificáronse as ferramentas e as sentenzas para realizar consultas. | * Exame en papel. | * PE.1 - Cuestionario con preguntas de resposta breve sobre ferramentas e sentenzas para consultas. | 2 |
| * CA5.2 - Realizáronse consultas simples sobre unha táboa. | * Exame en papel e en formato dixital. | * TO.1 - Documento de rexistro de realización de consultas simples cunha táboa. Escala de valores (observación indirecta). | 20 |
| * CA5.8 - Realizáronse consultas utilizando funcións básicas integradas no SXBD. | * TO.7 - Documento de rexistro de realización de consultas que utilicen funcións básicas integradas no SXBD. Escala de valores (observación indirecta). | 10 |

Modelo de proba combinada para PE.1, TO.1 e TO.7

Proponse un cuestionario con preguntas de resposta breve para avaliar CA5.1 e unha práctica para avaliar CA5.2 e CA5.8 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 de consulta solicitado.
* Imaxe capturada co resultado da execución do bloque dende o cliente Workbench.
* O arquivo dixital debe conter:
* Arquivos .sql co código correspondente ao bloque de sentencias.

A parte práctica realizarase na base de datos *equipos*. Subminístrase o arquivo *equipos.sql* co script que permite crear esa base de datos. A base de datos está formada 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.

* Táboa *equipo*

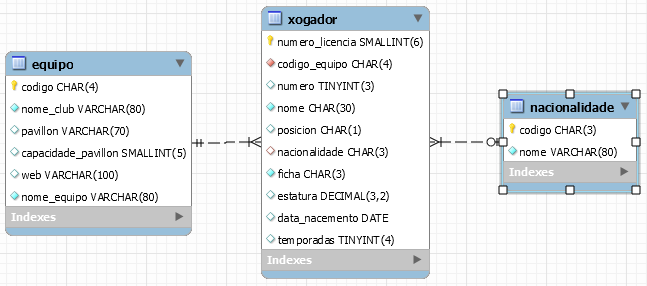
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome columna | Tipo | Null | Clave | Observacións |
| codigo | char(4) | non | Primaria | Código do equipo. Para enlazar con outras táboas |
| nome\_club | varchar(80) | non |  | Nome do club ao que pertence o equipo |
| pavillon | varchar(70) |  | 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

* Táboa *nacionalidade*

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

* 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 |  |  | Número de dorsal do xogador |
| nome | char(30) | non |  | Nome do xogador |
| posicion | char(1) |  |  | Posición na que xoga: B = base, P = pivot, A = alero, ...... |
| 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) |  |  | Estatura do xogador en metros |
| data\_nacemento | date |  |  | Data de nacemento |
| temporadas | tinyint |  |  | Número de tempadas que leva no equipo |



Proba para CA5.1 sobre identificación de sentenzas e ferramentas

Responder brevemente a seguinte cuestión:

* 1.- Indicar o nome de tres ferramentas e unha sentenza SQL que sirvan para realizar consultas. Explicar brevemente as características comúns que teñen as tres ferramentas.

Proba para CA5.2 sobre consultas simples sobre unha táboa

Escribir as sentenzas SQL que permitan realizar as seguintes consultas e capturar as imaxes que mostren o resultado dende o cliente. Se o resultado é moi longo, mostrar só un anaco do resultado e capturar a imaxe que identifique o número de liñas que devolve a consulta.

* 2.- Consultar o nome do club e nome e capacidade do pavillón no que xoga, para os equipos dos que se coñece a capacidade do seu pavillón.
* 3.- Consultar o código do equipo, nome, nacionalidade e estatura dos xogadores do equipo que ten como código 'SEVI', que teñen nacionalidade de Estados Unidos de América ('USA') ou Serbia ('SRB') e teñan unha estatura entre 2.03 e 2.15 (ambas incluídas).
* 4.- Consultar o nome dos clubs cun nome que empece por 'C.B' e que sexan Sociedade Anónima Deportiva, é dicir, que o nome do club finalice por 'S.A.D'.
* 5.- Consultar o nome, nacionalidade e estatura dos 5 xogadores máis altos.

Proba para CA5.8 sobre utilización de funcións MySQL nas consultas

Escribir as sentenzas SQL que permitan realizar as seguintes consultas e capturar as imaxes que mostren o resultado dende o cliente. Se o resultado é moi longo, mostrar só un anaco do resultado e capturar a imaxe que identifique o número de liñas que devolve a consulta.

* 6.- Calcular a media das estaturas dos xogadores que non son españois (os españois teñen na columna nacionalidade o valor 'ESP'). Redondear o resultado con dous decimais.
* 7.- Calcular o número de nacionalidades distintas que teñen os xogadores.

Exemplo de solución para entregar en papel

Proponse a seguinte solución:

* 1.- Indicar o nome de tres ferramentas e unha sentenza SQL que sirvan para realizar consultas. Explicar brevemente as características comúns que teñen as tres ferramentas.

O cliente gráfico Workbench, o cliente web PHPMyAdmin e a consola de texto mysql.exe. As tres teñen en común que poden establecer conexión co servidor, permitir escribir sentenzas de consulta e permitir enviar as consultas ao servidor para a súa execución.

A sentenza SQL que serve para facer consultas é SELECT.

* 2.- Consultar o nome do club e nome e capacidade do pavillón no que xoga, para os equipos dos que se coñece a capacidade do seu pavillón.

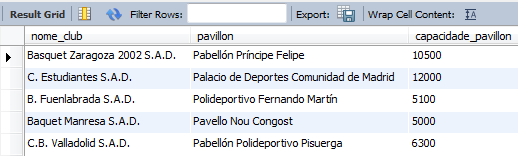
/\*Consultar o nome do club e o nome e capacidade do pavillón no que xoga,

para os equipos dos que se coñece a capacidade do seu pavillón\*/

**select** nome\_club**,** pavillon**,** capacidade\_pavillon

**from** equipo

**where** capacidade\_pavillon **is** **not** **null;**



* 3.- Consultar o código do equipo, nome, nacionalidade e estatura dos xogadores do equipo que ten como código 'SEVI', que teñen nacionalidade de Estados Unidos de América ('USA') ou Serbia ('SRB') e teñan unha estatura entre 2.03 e 2.15 (ambas incluídas).

/\* Consultar o código do equipo, nome, nacionalidade e estatura dos xogadores do equipo que ten como código 'SEVI', que teñen nacionalidade de Estados Unidos de América ('USA') ou Serbia ('SRB') e teñan unha estatura entre 2.03 e 2.15 (ambas incluídas).\*/

**select** codigo\_equipo**,** nome**,** nacionalidade**,** estatura

**from** xogador

**where** codigo\_equipo**=**'SEVI' **and** **(**nacionalidade**=**'USA' **or** nacionalidade**=**'SRB'**)**

**and** estatura**>=**2.03 **and** estatura**<=**2.15**;**

# Outra solución empregando o predicado BETWEEN

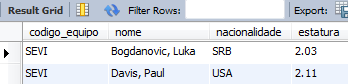
**select** codigo\_equipo**,** nome**,** nacionalidade**,** estatura

**from** xogador

**where** nacionalidade **in** **(**'USA'**,**'SRB'**)**

**and** codigo\_equipo **=** 'SEVI'

**and** estatura **between** 2.03 **and** 2.15**;**



* 4.- Consultar o nome dos clubs cun nome que empece por 'C.B' e que sexan Sociedade Anónima Deportiva, é dicir, que o nome do club finalice por 'S.A.D'.

/\*Consultar o nome dos clubs cun nome que empece por 'C.B' e que sexan

Sociedade Anónima Deportiva, é dicir, que o nome do club finalice por

'S.A.D'. \*/

**select** nome\_club

**from** equipo

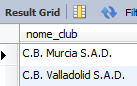
**where** nome\_club **like** 'C.B%S.A.D.'**;**

#Outra solución

**select** nome\_club

**from** equipo

**where** nome\_club regexp '^(c\\.b).\*(s\\.a\\.d\\.)$'**;**



* 5.- Consultar o nome, nacionalidade e estatura dos 5 xogadores máis altos.

/\* Consultar o nome, nacionalidade e estatura dos 5 xogadores máis altos \*/

**select** nome**,** nacionalidade**,** estatura

**from** xogador

**order** **by** estatura **desc** **limit** 5**;**



* 6.- Calcular a media das estaturas dos xogadores que non son españois (os españois teñen na columna nacionalidade o valor 'ESP'). Redondear o resultado con dous decimais.

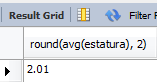
/\* Calcular a media das estaturas dos xogadores que non son españois ('ESP').

Redondear o resultado con dous decimais\*/

**select** **round(avg(**estatura**),** 2**)**

**from** xogador

**where** nacionalidade **!=** 'ESP'**;**

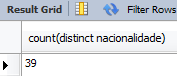


* 7.- Calcular o número de nacionalidades distintas que teñen os xogadores.

/\* Calcular o número de nacionalidades distintas que teñen os xogadores \*/

**select** **count(distinct** nacionalidade**)**

**from** xogador**;**



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 realización de consultas simples cunha táboa).

| Nome | Data | |
| --- | --- | --- |
| Indicadores para  CA5.2 - Realizáronse consultas simples sobre unha táboa. | Valoración máxima  20 | Cualificación |
| As sentenzas de consulta non teñen erros sintácticos e cumpren as condicións pedidas. | 18 |  |
| As sentenzas de consulta contidas no arquivo .sql correspondente, execútanse correctamente e resolven a consulta solicitada. | 1 |  |
| As imaxes capturadas dende o cliente Workbench reflicten os resultados pedidos e correspóndense co arquivo .sql. | 1 |  |

Exemplo de lista de valoración para TO.7

Proponse a seguinte lista de valoración para o instrumento de avaliación TO.7 (EV) (observación indirecta: documento de rexistro de realización de consultas que utilicen funcións básicas integradas no SXBD).

| Nome | Data | |
| --- | --- | --- |
| Indicadores para  CA5.8 - Realizáronse consultas utilizando funcións básicas integradas no SXBD. | Valoración máxima  10 | Cualificación |
| As sentenzas de consulta non teñen erros sintácticos e cumpren as condicións pedidas. | 8 |  |
| As sentenzas de consulta contidas no arquivo .sql correspondente, execútanse correctamente e resolven a consulta solicitada. | 1 |  |
| As imaxes capturadas dende o cliente Workbench reflicten os resultados pedidos e correspóndense co arquivo .sql. | 1 |  |

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