

Grado en Ingeniería del Software
Grado en Ingeniería del Software y Matemática Computacional

Bases de Datos



Práctica 1 Modelo Relacional y Lenguaje SQL

Rafael Socas Gutiérrez
Amador Maho Etohá

Curso 2021/22



Datos de los Alumnos

#	Nombre y Apellidos	Curso
1	Gonzalo Fernández Suárez	2 C
2		
3		

Instrucciones

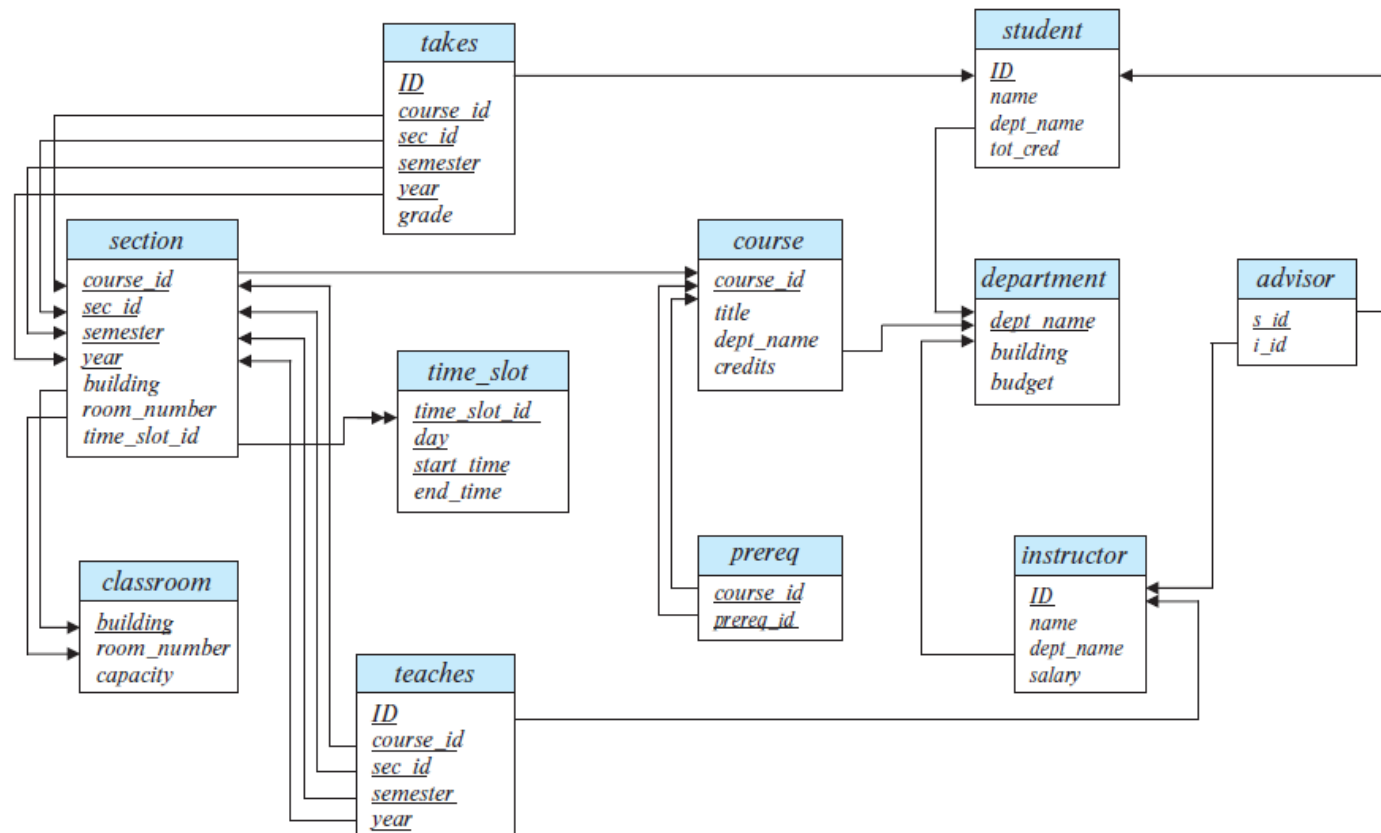
- Completa la práctica en este mismo power point rellenando las páginas en blanco o incluyendo más páginas si necesitas más espacio para los pantallazos y las explicaciones.
- Una vez completado el power point, guárdalo en formato pdf. **A la plataforma BB sube el pdf resultante.**
- **IMPORTANTE.** Adjunta también un script con todas las queries de Algebra Relacional y SQL que has utilizado para resolver las diferentes preguntas.
- Rellene el nombre/apellidos y el curso de los participantes del grupo.
- **Fecha máxima de entrega: Domingo 27 de Marzo del 2022 a las 23:59**

Enunciado

Se dispone de la BBDD **university** de la Universidad de Yale (<https://www.yale.edu/>) cuyo **Esquema** se presenta a continuación:

classroom(building, room_number, capacity)
department(dept_name, building, budget)
course(course_id, title, dept_name, credits)
instructor(ID, name, dept_name, salary)
section(course_id, sec_id, semester, year, building, room_number, time_slot_id)
teaches(ID, course_id, sec_id, semester, year)
student(ID, name, dept_name, tot_cred)
takes(ID, course_id, sec_id, semester, year, grade)
advisor(s_ID, i_ID)
time_slot(time_slot_id, day, start_time, end_time)
prereq(course_id, prereq_id)

**Cargue esta BBDD en MySQL usando los
ficheros `DDL.sql` y `smallRelationsInsertFile.sql`
que se adjuntan**

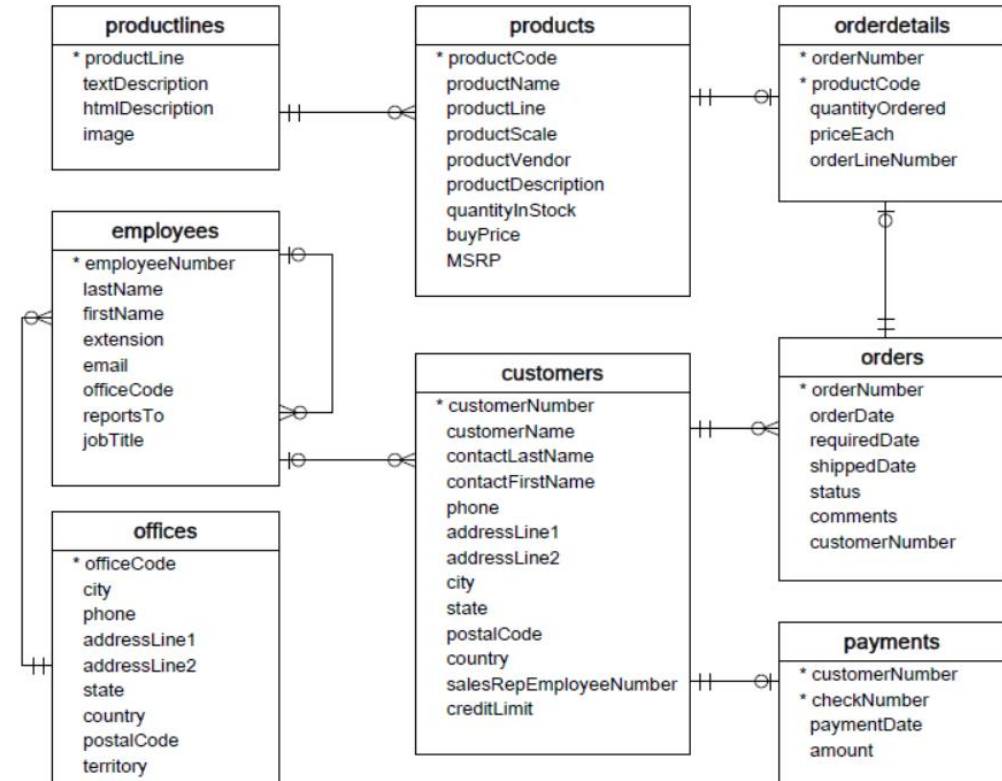


Enunciado

Se dispone también de la BBDD **classicmodels** que es la BBDD de ejemplo del tutorial de MySQL (<https://www.mysqltutorial.org/mysql-sample-database.aspx>) cuyo **Esquema** se presenta a continuación:

- **Customers:** stores customer's data.
- **Products:** stores a list of scale model cars.
- **ProductLines:** stores a list of product line categories.
- **Orders:** stores sales orders placed by customers.
- **OrderDetails:** stores sales order line items for each sales order.
- **Payments:** stores payments made by customers based on their accounts.
- **Employees:** stores all employee information as well as the organization structure such as who reports to whom.
- **Offices:** stores sales office data.

Cargue esta BBDD en MySQL usando el fichero **mysqlsampledatabse.sql** que se adjunta.



- 1) **PREGUNTA (1 punto):** Dada la tabla *office*, escriba en álgebra relacional y presentando la tabla resultante de cada apartado:
1. El código para presentar solamente las tuplas donde el territorio sea EMEA y la ciudad no sea London.
 2. Sobre la consulta anterior, escribe el código en álgebra relacional para obtener una relación donde solo estén: el nombre de la ciudad (city), el país (country) y el officeCode.
 3. Renombra la relación del ejercicio anterior como "EUoffice" ("oficina de la Unión Europea").

office

officeCode	city	phone	addressLine1	country	postalCode	territory
1	San_Francisco	16502194782	100 Market Street	USA	94080	NA
2	Boston	12158370825	1550 Court Place	USA	2107	NA
3	NYC	12125553000	523 East 53rd Street	USA	10022	NA
4	Paris	33147234404	43 Rue Jouffroy D'abbans	France	75017	EMEA
5	Tokyo	81332245000	4-1 Kioicho	Japan	1028578	Japan
6	Sydney	6129264451	5-11 Wentworth Avenue	Australia	2010	APAC
7	London	442078772041	25 Old Broad Street	UK	175	EMEA

$\sigma_{\text{territory} \neq \text{'EMEA'} \text{ and city} \neq \text{'London'}} \text{ office}$

office.officeCode	office.city	office.phone	office.addressLine1	office.country
1	' San_Francisco'	'16502194782'	' 100 Market Street'	' USA'
2	' Boston'	'12158370825'	' 1550 Court Place'	' USA'
3	' NYC'	'12125553000'	' 523 East 53rd Street'	' USA'
5	' Tokyo'	'81332245000'	' 4-1 Kioicho'	' Japan'
6	' Sydney'	'6129264451'	' 5-11 Wentworth Avenue'	' Australia'
7	' London'	'442078772041'	' 25 Old Broad Street'	' UK'

1

$\rho_{\text{EUoffice}} \pi_{\text{city, country, officeCode}} \sigma_{\text{territory} \neq \text{'EMEA'} \text{ and city} \neq \text{'London'}} \text{ office}$

EUoffice.city	EUoffice.country	EUoffice.officeCode
' San_Francisco'	' USA'	1
' Boston'	' USA'	2
' NYC'	' USA'	3
' Tokyo'	' Japan'	5
' Sydney'	' Australia'	6
' London'	' UK'	7

3

2

$\pi_{\text{city, country, officeCode}} \sigma_{\text{territory} \neq \text{'EMEA'} \text{ and city} \neq \text{'London'}} \text{ office}$

office.city	office.country	office.officeCode
' San_Francisco'	' USA'	1
' Boston'	' USA'	2
' NYC'	' USA'	3
' Tokyo'	' Japan'	5
' Sydney'	' Australia'	6
' London'	' UK'	7

2) PREGUNTA (1 punto): Indica e interpreta lo que se obtiene después de ejecutar las siguientes expresiones de Álgebra relacional sobre las dos relaciones dadas :

Estudiante (dni_est, nomb_est)

Profesor (dni_prof, nomb_prof).

dni_est	nomb_est
12345	Perez
968574	Valdez
192837	Juarez

dni_prof	nomb_prof
654789	Perez
852741	Fernandez
192837	Juarez

1. Estudiante \cup Profesor
2. Estudiante \cap Profesor
3. Profesor $-$ Estudiante

¿De cuántas tuplas consta cada relación resultante?

¿Qué condición deben de cumplir ambas relaciones Estudiante y Profesor entre sí para poder realizarse las operaciones de álgebra relacional indicadas?

agregar nueva relación

```
1 group: university
2
3 Estudiante = {
4 dni_est, nomb_est
5 12345, 'Perez'
6 968574, 'Valdez'
7 192837, 'Juarez'
8 }
9
10 Profesor = {
11 dni_prof, nomb_prof
12 654789, 'Perez'
13 852741, 'Fernandez'
14 192837, 'Juarez'
15 }
```

university [utilizar Grupo en editor](#)

- Estudiante

Estudiante.dni_est	Estudiante.nomb_est
12345	'Perez'
968574	'Valdez'
192837	'Juarez'
- Profesor

Profesor.dni_prof	Profesor.nomb_prof
654789	'Perez'
852741	'Fernandez'
192837	'Juarez'

Estudiante \cup Profesor

Estudiante.dni_est	Estudiante.nomb_est
12345	'Perez'
968574	'Valdez'
192837	'Juarez'
654789	'Perez'
852741	'Fernandez'

Estudiante UNIÓN Profesor

Muestra todos los valores DISTINTOS almacenados en las tablas Estudiante y Profesor

Estudiante \cap Profesor

Estudiante.dni_est	Estudiante.nomb_est
192837	'Juarez'

Estudiante INTERSECCIÓN Profesor

Muestra los valores coincidentes entre las tablas Estudiante y Profesor

Profesor - Estudiante

Profesor.dni_prof	Profesor.nomb_prof
654789	'Perez'
852741	'Fernandez'

Profesor NO Estudiante

Muestra aquellos valores de la tabla Profesor que no se encuentran en Estudiante

3) PREGUNTA (1 punto): De la base de datos **university**, se quiere obtener en una sola consulta, el nombre del departamento (*dept_name*) que tiene el mayor número de profesores.

```
SELECT COUNT(dept_name) AS numeroProfes, dept_name FROM `instructor`  
GROUP BY dept_name ORDER BY COUNT(dept_name) DESC LIMIT 1;
```

numeroProfes	dept_name
3	Comp. Sci.

4) PREGUNTA (1 punto): Responda la siguiente pregunta: ¿Cuál es la clase dónde se impartieron más asignaturas en el año 2018? (BBDD **university**)

Condiciones: para responder la pregunta debes conseguir en una sola consulta, mostrar una tabla con la información solicitada.

```
SELECT COUNT(room_number) AS asignaturas, room_number FROM `section` WHERE year = 2018 GROUP BY room_number ORDER BY asignaturas DESC LIMIT 1;
```

asignaturas	room_number
3	101

5) PREGUNTA (1 punto): En la base de datos **university**, dentro de la tabla *course* hay un curso que no está vinculado a ninguna oferta o *section* y queremos asignarlo al aula menos utilizada del año 2018 para el siguiente año. En una sola query muestra una tabla donde se vea el curso que no está en ninguna *section*. ¿Cuál es el nombre de la asignatura? ¿A qué departamento pertenece este curso? ¿Y cuál es el aula menos utilizada del año 2018?

```
SELECT course_id, title, dept_name FROM course WHERE course_id
NOT IN (SELECT course_id FROM section);
```

course_id	title	dept_name
BIO-399	Computational Biology	Biology

Aula menos utilizada en 2018

```
SELECT COUNT(room_number) AS n, room_number FROM section WHERE year = 2018 GROUP BY
room_number ORDER BY n ASC LIMIT 1;
```

n	room_number
1	3128

6) PREGUNTA (1 punto): Crea una consulta donde aparezcan los profesores y el número de asignaturas que imparten cada uno ordenados de mayor a menor número de asignaturas.

```
SELECT instructor.name, COUNT(course.course_id) AS numAsignaturas FROM instructor, c
ourse WHERE course.dept_name = instructor.dept_name GROUP BY instructor.name ORDER
BY numAsignaturas DESC;
```

name	numAsignaturas ▼ 1
Srinivasan	5
Katz	5
Brandt	5
Crick	3
Wu	1
Mozart	1
Einstein	1
El Said	1
Gold	1
Califieri	1
Singh	1
Kim	1

7) PREGUNTA (1 punto): Inspecciona la tabla *employees* de la base de datos *classicmodels*. Crea una consulta donde haya una columna con el nombre y apellido (concatenados) de los empleados y otra columna con el nombre y apellido (concatenada también) donde aparezca el jefe a quién reporta.

```
SELECT CONCAT(e.firstName, ' ', e.lastName) AS nombreEmpleado, CONCAT(e1.firstName, ' ', e1.lastName) AS nombreJefe FROM employees AS e JOIN employees AS e1 ON e.reportsTo = e1.employeeNumber;
```

nombreEmpleado	nombreJefe
Mary Patterson	Diane Murphy
Jeff Firrelli	Diane Murphy
William Patterson	Mary Patterson
Gerard Bondur	Mary Patterson
Anthony Bow	Mary Patterson
Leslie Jennings	Anthony Bow
Leslie Thompson	Anthony Bow
Julie Firrelli	Anthony Bow
Steve Patterson	Anthony Bow
Foon Yue Tseng	Anthony Bow
George Vanauf	Anthony Bow
Loui Bondur	Gerard Bondur
Gerard Hernandez	Gerard Bondur
Pamela Castillo	Gerard Bondur
Larry Bott	Gerard Bondur
Barry Jones	Gerard Bondur
Andy Fixter	William Patterson
Peter Marsh	William Patterson
Tom King	William Patterson

8) PREGUNTA (1 punto): Ya conocemos algunas formas de importar datos en una tabla dentro de una base de datos a través de herramientas y funcionalidades del *dump*, basadas generalmente en sentencias SQL INSERT.

Se pide investigar una forma alternativa de realizar esta misma operación, es decir, importar datos de una tabla a una base de datos, pero en este caso siendo una tabla en formato .csv sin utilizar código SQL INSERT.

Dada la siguiente tabla ***asignatura*** con los datos que se especifican, transfórmala a un fichero en formato .csv (*asignatura.csv*) y crea una tabla (CREATE TABLE) llamada Asignatura en cualquier base de datos de tu sistema e importa los datos de la tabla *asignatura.csv* a la tabla Asignatura recién creada en tu SGBD.

Explica detalladamente todos los pasos seguidos para realizar este ejercicio y demostrando con ficheros, pantallazos, etc., todas las evidencias que justifiquen que has seguido el proceso de forma válida. **Prueba con una consulta SQL** que los datos de la tabla .csv se han importado correctamente a la correspondiente tabla de la base de datos en tu SGBD.

Ayuda: puedes utilizar el tutorial <https://www.mysqltutorial.org/import-csv-file-mysql-table/> de MySQL para realizar la investigación.

asignatura

ID	asignatura	semestre	creditos
1	Int BBDD	2	6
2	Sist. Oper.	1	6
3	Lab. Programacion	2	4,5

1

```
CREATE TABLE asignatura (  
id INT NOT NULL AUTO_INCREMENT,  
asignatura VARCHAR(255) NOT NULL,  
semestre INT NOT NULL,  
creditos DECIMAL(10,9) NOT NULL,  
PRIMARY KEY(id));
```

```
1 ID;asignatura;semestre;creditos  
2 1;Int BBDD;2;6  
3 2;Sist. Oper.;1;6  
4 3;Lab. Programacion;2;4.5  
5
```

2

```
LOAD DATA INFILE 'C:/Users/gonef/Desktop/U-tad/2º/2º cuatri/Bases de Datos/2022 BASES DE  
DATOS/PRACTICA 1/asignaturas.csv'  
INTO TABLE asignatura  
FIELDS TERMINATED BY ';'   
ENCLOSED BY ""  
LINES TERMINATED BY '\n'  
IGNORE 1 ROWS;
```

3

```
SELECT * FROM `asignatura`;
```

id	asignatura	semestre	creditos
1	Int BBDD	2	6.000000000
2	Sist. Oper.	1	6.000000000
3	Lab. Programacion	2	4.500000000

9) PREGUNTA (1 punto): Se pide mostrar en una misma tabla, y a través de una sola consulta, el top 5 de clientes que han realizado más pagos. En la tabla tendrán que estar el número de cliente, su nombre y el número de pagos que ha realizado cada uno (BBDD **classicmodels**)

Ayuda: hay que trabajar con las tablas *payments* y *customers*.

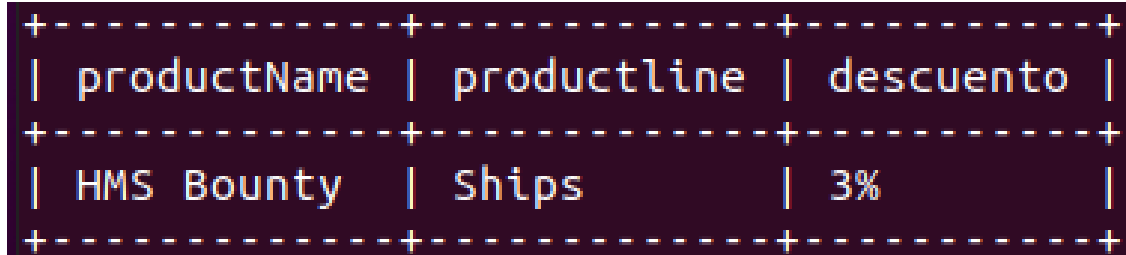
```
SELECT COUNT(payments.customerNumber) AS nPagos, c.customerNumber,  
CONCAT(c.contactFirstName, ' ', c.contactLastName)  
FROM `customers` AS c JOIN payments ON c.customerNumber = payments.customerNumber  
GROUP BY payments.customerNumber ORDER BY nPagos DESC LIMIT 5;
```

nPagos ▼ 1	customerNumber	CONCAT(c.contactFirstName,' ',c.contactLastName)
13	141	Diego Freyre
9	124	Susan Nelson
4	114	Peter Ferguson
4	128	Roland Keitel
4	121	Jonas Bergulfsen

10) PREGUNTA (1 punto) De cara al próximo cuatrimestre, la empresa ClassicModels pretende aplicar una serie de descuentos extra para sus clientes más fieles. La cuantía de los descuentos se establece en función de las distintas líneas de producto, de manera que para cada tipo se aplicará el porcentaje de descuento siguiente: coches (Classic Cars → '5%'), aeroplanos (Planes → '2%'), motocicletas (Motorcycles → '4%'), embarcaciones (Ships → '3%') y resto un 6%.

Crea **una** sola consulta, en la que a partir del nombre de un producto se muestre una sola tabla, su línea de producto y el descuento correspondiente, por ejemplo como en la **Figura10**, es decir, para el productName HMS Bounty (BBDD **clasicmodels**).

Ayuda: tienes que trabajar con la tabla **products**.




productName	productline	descuento
HMS Bounty	Ships	3%

Figura10


```
SELECT productName, productLine,  
CASE  
    WHEN productLine = 'Classic Cars' THEN '5%'  
    WHEN productLine = 'Planes' THEN '2%'  
    WHEN productLine = 'Motorcycles' THEN '4%'  
    WHEN productLine = 'Ships' THEN '3%'  
ELSE '6%'  
END AS descuento FROM products;
```

productName	productLine	descuento
1969 Harley Davidson Ultimate Chopper	Motorcycles	4%
1952 Alpine Renault 1300	Classic Cars	5%
1996 Moto Guzzi 1100i	Motorcycles	4%
2003 Harley-Davidson Eagle Drag Bike	Motorcycles	4%
1972 Alfa Romeo GTA	Classic Cars	5%
1962 LanciaA Delta 16V	Classic Cars	5%
1968 Ford Mustang	Classic Cars	5%
2001 Ferrari Enzo	Classic Cars	5%
1958 Setra Bus	Trucks and Buses	6%
2002 Suzuki XREO	Motorcycles	4%
1969 Corvair Monza	Classic Cars	5%
1968 Dodge Charger	Classic Cars	5%
1969 Ford Falcon	Classic Cars	5%
1970 Plymouth Hemi Cuda	Classic Cars	5%
1957 Chevy Pickup	Trucks and Buses	6%
1969 Dodge Charger	Classic Cars	5%
1940 Ford Pickup Truck	Trucks and Buses	6%
1993 Mazda RX-7	Classic Cars	5%
1937 Lincoln Berline	Vintage Cars	6%
1936 Mercedes-Benz 500K Special Roadster	Vintage Cars	6%
1965 Aston Martin DB5	Classic Cars	5%
1980s Black Hawk Helicopter	Planes	2%
1917 Grand Touring Sedan	Vintage Cars	6%
1948 Porsche 356-A Roadster	Classic Cars	5%
1995 Honda Civic	Classic Cars	5%



 Calle Playa de Liencres, 2 bis
(entrada por calle Rozabella)
Parque Europa Empresarial
Edificio Madrid
28290 Las Rozas, Madrid

 900 373 379  info@u-tad.com

 [SOLICITA MÁS INFORMACIÓN](#)



CENTRO ADSCRITO A:



PROYECTO COFINANCIADO POR:

