

BASES DE DATOS

SQL: Ejercicios

SQL

1. Sobre el ejercicio de música responde a las siguientes cuestiones

CANCION(cod: d_can, título: d_tit, duración: d_dur)

COMPAÑIA(cod: d_comp, nombre: d_nom, dir: d_dir, fax: d_tel, tfno: d_tel)

DISCO(cod: d_dis, nombre: d_nom, fecha: d_fecha, cod_comp: d_comp, cod_gru: d_gru)

ESTA(can: d_can, cod: d_dis)

GRUPO(cod: d_gru, nombre: d_nom, fecha: d_fecha, pais: d_pais)

ARTISTA(dni: d_dni, nombre: d_nom)

CLUB (cod: d_club, nombre: d_nom, sede: d_dir, num: d_num, cod_gru: d_gru)

PERTENECE(dni: d_dni, cod: d_gru, funcion: f_fun)

- Obtén toda la información de los artistas
- Obtén el nombre de los artistas que se llamen Paco
- Obtén el nombre de los artistas cuyo nombre empieza por P y termina por O
- Obtén el nombre de los artistas cuyo nombre empieza por P y termina por O y cuya longitud sea 5
- Clubes que tengan 3,4 o 5 socios
- Los que tengan entre 20 y 30
- Obtén el nombre, fax y teléfono de todas las compañías
- Obtén los distintos nombres de artista que tenemos (sin repeticiones)
- Compañías para las que no hayamos guardado dirección
- Aquellas para las que si lo hayamos guardado
- El mismo resultado pero ordenado alfabéticamente por el nombre de manera ascendente
- El mismo resultado que el ejercicio j pero ordenado alfabéticamente por el nombre de manera descendente
- Número de tuplas en la relación grupo
- Número de discos de cada grupo

2. Sobre el ejercicio del mundial responde a las siguientes cuestiones

HOTEL	Código_H	Nombre	Dirección	Teléfono	Código_E	NºHabitaciones			
EQUIPO	Código_E	País	Camiseta	Total_Goles	ÁRBITRO	Código_E1	Código_E2	Fecha	Árbitro
JUGADOR	DNI	Nombre	FechaN	Goles	Código_E	CLUBES_AÑO	DNI	Club	
PARTIDO	Código_E1	Código_E2	Fecha	Resultado	CONTROL_ANTIDOPAJE	DNI	Fecha	Lugar	

- Buscando casualidades que se hayan producido en el mundial, deseamos obtener los nombres de aquellos jugadores (JUGADOR.Nombre contiene el nombre seguido del apellido) cuyo nombre de pila comience por “Carlos” y se hayan hospedado en el hotel del mismo nombre (“Carlos”), además de los nombres de aquellos jugadores que cumplieran años el día de la inauguración (el 31/05/2002).
- Mostrar, junto al nombre de cada país participante en el mundial, el número total de sus jugadores que hayan pasado al menos un control antidopaje. Téngase en cuenta que un mismo jugador ha podido pasar varios controles y deseamos contar el número de jugadores (NO el número de controles realizados a los jugadores de un país). Se desea que el resultado quede ordenado alfabéticamente por nombre de país de forma descendente.

- c. Para aquellos equipos tales que entre dos de sus jugadores hayan marcado más de la mitad de los goles totales del equipo, se desea obtener el código del equipo, los nombres de ambos jugadores, el total de goles del equipo y los goles que ha marcado cada uno de los dos jugadores. No queremos que una misma pareja de jugadores aparezca repetida en el resultado (el primer jugador deberá ser menor alfabéticamente que el segundo). Por ejemplo:

ESP, Gaizka Mendieta, Raul González, 25, 10, 5

FRA, Francois Republique, Zinedine Zidane, 24, 9, 9

- d. Escribe las sentencias de creación de tablas necesarias para generar la base de datos del mundial de fútbol. (incluyendo restricciones).
- e. Introduce tres hoteles (códigos h1, h2 y h3) , dos equipos (códigos e1,e2) y 5 jugadores (j1 a j5). ¿Es relevante el orden en el que introduces los datos?
- f. Por motivos de organización, se ha eliminado el hotel 3 y se ha incluido el 4 teniendo que asignar los equipos que estaban en dicho hotel al 4. Ejecuta las sentencias necesarias para ejecutar dichos cambios en la base de datos.
- g. En la información que se almacena sobre los hoteles se nos ha olvidado introducir la población. Modifica la tabla para introducir ese nuevo atributo. Luego introduce a cada hotel una población
- h. Elimina de la relación Hotel el atributo que acabas de incluir
- i. Obtén toda la información de la relación hotel (atributos que tiene y tipo de los mismos)
- j. Obtén los nombres de todos los jugadores del mundial
- k. Obtén los nombres de los jugadores del equipo e1
- l. Los nombres de jugador que empiecen por H.
- m. Los nombres de jugador que empiecen por H y sean de longitud 5
- n. Los jugadores que tengan el número de camiseta igual a 1,3 o 5
- o. Los jugadores con número de camiseta entre 1 y 15
- p. Obtén toda la información de los jugadores ordenado por su código de equipo, en orden descendente y luego en orden ascendente
- q. Obtén la información de los hoteles cuyo teléfono o nombre sean null. Luego los que no tengan null ni en uno ni en otro
- r. Obtén la cantidad de goles obtenidos por cada equipo del mundial ordenado por país descendente
- s. Obtén la media de goles de los jugadores de cada equipo
- t. Obtener el máximo, mínimo media de goles de jugadores del mundial
- u. ¿Cuántos jugadores han participado en el mundial? ¿cuántos jugadores han participado por cada equipo?

3. Sobre la base de datos de viajes

Oferta (<u>Cod_via</u> , <u>Fecha_Salida</u> , Suplemento)
Viaje (<u>Cod_Via</u> , Precio, NDias)
Ciudad (<u>Cod_Ciu</u> , Nombre, Pais, Info_extra)
Ciudad_Viaje (<u>Cod_Via</u> , <u>Cod_ciu</u> , Orden, Kdias)
Reserva (<u>Cod_res</u> , NPersonas, Fecha_Reserv, Cod_Cli, Cod_Via, Fecha_Salida)

- a. Escribe las sentencias SQL necesarias para crear las tablas anteriores

Para todas las claves externas explica qué implica cada una de las opciones que se pueden poner tras on delete y on update. Indica también por qué pones not null en los atributos en los que lo pongas

- b. Oferta: añadirle un atributo descuento
 - c. Las ciudades por defecto serán Francesas
 - d. Quitarle a ciudad el valor por defecto
 - e. ¿En qué orden han de crearse las tablas?
 - f. Indica el orden de creación de tablas para todos los modelos relacionales hechos en clase
 - g. Escribe las sentencias insert para reflejar la siguiente información
 - i. Tenemos un viaje que pasa por Toulouse, Montauban y Moissac que tiene una duración de 10 días. Si salimos el día 10 de mayo costará 400 euros por persona y saliendo el día 17 costará 450 euros por persona
 - ii. Tenemos tres clientes. El primero reserva 2 plazas en el viaje del 10 de mayo, el segundo 3 en el del 17 y por último el tercero reservará 10 plazas en el primero y otras 10 en el segundo
 - iii. Quitarle al cliente 3 las 10 plazas del viaje del 17 de mayo y añadirle 3 en el del 10 de mayo
 - h. La oferta del diez de mayo pasa a costar 300 euros
 - i. La oferta del 10 de mayo pasa a costar 10 euros más
 - j. Las dos ofertas pasan a costar la media de ambas
4. Dada una base de datos con las relaciones que se muestran a continuación:

Oferta (<u>Cod_via</u> , <u>Fecha_Salida</u> , Suplemento)
Viaje (<u>Cod_Via</u> , Precio, NDias)
Ciudad (<u>Cod_Ciu</u> , Nombre, Pais, Info_extra)
Ciudad_Viaje (<u>Cod_Via</u> , <u>Cod_ciu</u> , Orden, Kdias)
Reserva (<u>Cod_res</u> , NPersonas, Fecha_Reserv, Cod_Cli, Cod_Via, Fecha_Salida)

Da las sentencias SQL que den respuesta a los ejercicios siguientes:

- a. Obtén el nombre de todas las ciudades
- b. Obtén los datos de todas las ciudades almacenadas en la BD ordenados alfabéticamente por el nombre de ciudad.
- c. Obtén la lista de todos los viajes que cuestan menos de 100.000 .
- d. Obtén la lista de los viajes que cuestan menos de 100.000 y que duren más de seis días.
- e. Obtén los datos de todas las reservas realizadas en los tres primeros meses de 1998..
- f. Obtén la lista de las ciudades que tienen más de una palabra.
- g. Obtén los datos de los viajes ofrecidos en verano de 1997 (julio, agosto, septiembre) ordenado ascendentemente por la fecha de salida.
- h. Obtén los nombres de las ciudades que se visitan en el viaje 2 ordenado por orden de visita.
- i. Muestra las ofertas de febrero de 1998 junto con su precio total (precio+suplemento)
- j. Consulta el número de ofertas existentes con fecha de salida mayor que el uno de enero de 1998
- k. Obtén la duración media de los viajes así como la duración máxima y la duración mínima

- l. Saca una lista en la que aparezcan los códigos de los viajes junto con la cantidad de ciudades que se visitan en cada.
 - m. Saca una lista en la que aparezcan los códigos de los viajes junto con la cantidad de ciudades diferentes que se visitan en cada.
 - n. Queremos obtener una lista de los códigos de viaje junto con su duración en palabras. Es decir, no 1 sino “1 día”, no 2 sino “ 2 días” etc. Para los de más de cuatro pondremos “más de cuatro días”
5. Sobre la base de datos viaje, escribe la sentencia SQL para obtener
- a. Obtén la información de los viajes cuya duración sea mayor que la del viaje con código 1
 - b. Obtén la información de las ofertas relacionadas con los viajes cuya duración sea mayor que la del viaje con código 1
 - c. La información de los viajes junto a las de sus ofertas. Si no tenemos ofertas para un viaje también queremos que aparezca
 - d. ¿cuántas ofertas tenemos relacionadas con los viajes cuya duración es mayor que la del viaje con código 1?
 - e. Número de ciudades en la base de datos
 - f. Cuenta el número de ofertas diferentes para cada viaje
 - g. Cuenta el número de ofertas diferentes para cada viaje. Si algún viaje no tiene ofertas relacionadas aparecerá en el resultado con un 0 como número de ofertas.
 - h. inserta una nueva tupla en ciudad, sin introducir ninguna información extra
 - i. Número de ciudades que tengan almacenada información extra
 - j. Número de ciudades que no tengan almacenada información extra
 - k. Nombres de los países diferentes para los que tenemos ciudades almacenadas
 - l. Número de países diferentes a los que podemos viajar
 - m. Número de ciudades por país
6. Teniendo en cuenta la base de datos con las relaciones que aparecen a continuación,

ALUMNO (AL, NOMBREA, FECHANCTO)

ASIGNATURAS (AS, NOMBREA SIG, CURSO)

MATRICULA_NOTA (AL, AS, AÑO , NOTAJ, NOTAS)

Escribe las sentencias SQL que den respuesta a las siguientes preguntas. Utilizando en la medida de lo posible INNER y NATURAL JOINS. Aquellas respuestas que incluyan cosas no vistas en clase se contarán como mal.

- a. Media de la diferencia de las notas de junio y septiembre para aquellos alumnos que hayan realizado el examen en ambas convocatorias en el año 2003.
 - b. ¿Cómo modificarías la sentencia anterior para tener únicamente en cuenta los alumnos cuyo nombre empiece por I y acabe por n?
 - c. ¿Cómo modificarías la sentencia a para sacar lo mismo para los años 2001, 2002, 2003 ó 2004? Escribe dos opciones posibles.
 - d. Queremos obtener la lista de alumnos ordenados por su Nombre junto con las notas obtenidas en junio de 2003 en cada asignatura, pero la nota la queremos en letras. Es decir, por ejemplo, si tiene un 5 aprobado etc.
 - e. Indica cada alumno cuantas veces se ha matriculado en cada asignatura
 - f. Alumnos cuya nota en junio de 2003 en metodología sea igual a un tercio de su edad, ordenado por fecha de nacimiento creciente.(Fecha es el año de nacimiento)
7. Dadas las siguientes relaciones de una base de datos en las que se representan un conjunto de medicamentos, laboratorios que los fabrican y componentes de los mismos (cada medicamento puede tener muchos componentes y al revés)
- Medicamento (cod, nombre, laboratorio, año_ideado)
 Laboratorio(NIF, nombre, tel)
 Componente (cod, nombre)
 med_comp.(codmed, codcomp, cantidad)

Escribe las sentencias SQL que den respuesta a las siguientes preguntas:

- a. Todos los componentes que se usan en el laboratorio de nombre BAYER ordenados por la cantidad de componente necesario descendente
- b. Haz una lista en la que aparezca para cada uno de los laboratorios qué productos han ideado después de 1977 y cada uno de los productos qué cantidad de componentes necesita.
- c. Medicamentos ideados antes que la aspirina
- d. Cambiar la sentencia anterior para obtener los medicamentos que empiezan por A y acaban en A con un máximo de 5 caracteres o aquellos que acaban en B

8. Sobre el ejercicio de la cooperativa de los taxistas

Taxi(matrícula, tfno, km_rev)

Taxista(dni, nombre, tfno, dirección, edad)

Taller(codt, nombre, tfno, responsable)

Conduce(dni, matrícula, fecha, h_inicio, h_fin)

Revisión(codt, matrícula, fecha)

donde los atributos tienen el siguiente significado:

Taxi :	<i>matrícula</i> : matrícula del coche	<i>tfno</i> : teléfono del coche
	<i>km_rev</i> : kilómetros a los que se tiene que hacer la próxima revisión	
Taxista:	<i>dni</i> : D.N.I. del taxista	<i>nombre</i> : nombre del taxista
	<i>edad</i> : del taxista	<i>tfno</i> : teléfono particular del taxista
	<i>dirección</i> : dirección del taxista	
Taller:	<i>codt</i> : código del taller	<i>nombre</i> : nombre del taller
	<i>tfno</i> : teléfono del taller	<i>responsable</i> : persona responsable del taller
Conduce	el taxista de DNI <i>dni</i> condujo (o conduce) el taxi de matrícula <i>matrícula</i> el día <i>fecha</i> desde las <i>h_inicio</i> hasta las <i>h_fin</i>	
Revisión	el taller de código <i>codt</i> revisó el taxi de matrícula <i>matrícula</i> el día <i>fecha</i>	

- Obtén para cada taxista los taxis que ha conducido ordenado por nombre de taxista ascendente
- Número de taxis conducidos por cada taxista..
- Obtén los nombres de los talleres que han revisado algún coche cuya matrícula esté entre 1256 y 1356
- Nombres de los taxistas que no tienen teléfono almacenado
- Escribe en SQL las sentencias necesarias para crear las relaciones conduce y taxista (tal y como lo harías en SQL)

9. Sobre el modelo relacional de congresos, responde a las siguientes cuestiones en SQL

Moderador(dni, nombre, especialidad, pais)

Autor(dni, nombre, pais, edad)

Sesión (cod_sesion, fecha, hora_inicio, hora_fin, dni)

Clave ajena dni → Moderador.dni

Ponencia(cod_ponencia, titulo, tematica, cod_sesion, posición)

Clave ajena cod_session → sesion.cod_sesion

Ha_escrito(dni, cod_ponencia)

Clave ajema cod_ponencia → ponencia, dni → autor

donde los atributos tienen el siguiente significado:

Moderador:	<i>dni</i> : D.N.I. del moderador <i>especialidad</i> : del moderador	<i>nombre</i> : del moderador <i>país</i> : del moderador
Autor:	<i>dni</i> : D.N.I. del autor <i>edad</i> : del autor	<i>nombre</i> : del autor <i>país</i> : del autor
Sesión:	<i>cod_sesión</i> : código de la sesión <i>hora_fin</i> : cuándo termina la sesión	<i>hora_inicio</i> : cuándo empieza la sesión <i>dni</i> : D.N.I del moderador de la sesión
Ponencia:	<i>cod_ponencia</i> : código de la ponencia <i>temática</i> : de qué trata la ponencia <i>posición</i> : en qué orden se presenta (1º, 2º,...)	<i>título</i> : de la ponencia <i>cod_sesión</i> : en qué sesión se presenta
Ha _escrito	el autor de D.N.I. <i>dni</i> ha escrito la ponencia de código <i>cod_ponencia</i>	

- Número de ponencias escritas por cada autor (0,1,2,...)
- Pares de autores del mismo país
- Autores cuyo nombre empiece por D o termine en F y que sea mayor de edad pero que no tenga más de 30 años.
- Dni de los autores y de los moderadores franceses
- Moderadores cuya especialidad esté formada por dos palabras que empiecen por T y no tengan nombre almacenado junto con aquellos cuyo nombre es de 5 caracteres
- Nombre de moderador junto con el número de sesiones moderadas ordenador por su dni de manera descendente
- Número de países diferentes para los que tenemos algún moderador
- Ya no nos interesa almacenar la hora en la que termina cada sesión
- Eliminar las tuplas de ha_escrito del escritor con DNI '0909'
- Todas las sesiones moderadas por Paul pasan a ser moderadas por Izaro
- Necesitamos poder almacenar el año en el que se ha escrito cada ponencia por cada autor
- El autor con dni '1989' ha escrito una nueva ponencia . Supón que la bd está vacía. Indica todas las instrucciones necesarias y en qué orden las ejecutarías
- Borra las sesiones moderadas por moderadores Franceses.

10. (2009) Sobre el modelo relacional de **olimpiadas**, responde a las siguientes cuestiones en SQL

- ¿En qué orden han de crearse?
- Introduce algunos datos en las tablas que has creado
- Añade a la tabla deportista la fecha de nacimiento del mismo
- Elimina el atributo que acabas de crear
- Modifica el atributo procedencia de voluntario y haz que por defecto tome el valor 'CHINA'
- Indica las restricciones de integridad que pueden violarse ante cada tipo de operación sobre las tablas. Pon ejemplos para cada una de ellas.
- Elimina todas las actividades desarrolladas en Shanghai
- Todas las actividades programadas en Quingdao se desarrollarán en Shanghai

- i. Diferentes ciudades para las que hay programada alguna actividad
 - j. Número de ciudades diferentes para las que hay programada alguna actividad
 - k. Para cada ciudad el número de actividades diferentes que tiene programada
 - l. Nombre de deportista junto con el número de récords diferentes obtenidos, los que no tienen ninguno también han de aparecer en el resultado
 - m. Nombre de deportistas que empiecen por r y contengan una l
 - n. Nombre de deportistas de 6 caracteres que empiece por m
 - o. Deportistas para los que no se haya almacenado la dirección
 - p. Deportistas para los que se haya almacenado la dirección
 - q. Deportistas del equipo del deporte con código 1 de Francia ordenados por nombre ascendente y apellido descendente
 - r. Pares de deportistas con el mismo nombre, sin repeticiones
 - s. Pares de deportistas cuyo primer apellido empiece por p, sin repeticiones
 - t. Nombres de deportistas que no aparezcan entre los nombres de los voluntarios(no funciona en MySQL)
 - u. Nombres de deportistas que aparezcan entre los nombres de los voluntarios
 - v. Los nombres de los deportistas y los de los voluntarios
11. Tomando como punto de partida el modelo relacional para la bd música, modifica las tablas de la base de datos música tal y como se describe a continuación
 - a. Añadir a los artistas su fecha de nacimiento
 - b. Queremos almacenar información sobre desde cuándo pertenece cada artista a cada grupo
 - c. La duración de las canciones por defecto será de un minuto
 - d. Los nombres de los grupos no se pueden repetir
 - e. Quitar a las canciones el valor por defecto de la duración
 - f. Añadir un atributo salario a los grupos que por defecto tomará el valor 0
 - g. Quitar el atributo que acabas de añadir
 12. Tomando como punto de partida el modelo relacional para la bd música, ,inserta en la base de datos la siguiente información
 - h. El grupo *grupo4* ha sacado un nuevo disco con la compañía *compañía2* que tiene dos canciones *cancion1* y *cancion2* de 2 minutos cada una
 - i. Crea un nuevo grupo de música con dos participantes y dos discos cada uno con tres canciones
 13. Sobre la base de datos de viajes

Oferta (<u>Cod_via</u> , <u>Fecha_Salida</u> , Suplemento)
Viaje (<u>Cod_Via</u> , Precio, NDias)
Ciudad (<u>Cod_Ciu</u> , Nombre, Pais, Info_extra)
Ciudad_Viaje (<u>Cod_Via</u> , <u>Cod_ciu</u> , Orden, Kdias)
Reserva (<u>Cod_res</u> , NPersonas, Fecha_Reserv, Cod_Cli, Cod_Via, Fecha_Salida)

Da las sentencias SQL que den respuesta a los ejercicios siguientes:

- a. Borra las ciudades que no se visitan en ningún viaje
- b. Inserta un nuevo viaje de tres días para el que desconocemos el precio
- c. Asigna las ofertas del viaje 20 al que acabas de crear
- d. Borra el viaje de código 10
- e. El viaje de código 2 incrementa su precio en un 10%
- f. El viaje de código 10 tendrá el mismo precio que el de código 2
- g. Borra de todos los viajes las etapas que realizan en Madrid

- h. Añade un nuevo atributo en viaje que se llame netapas que representará el número de etapas que realiza. Actualiza el valor de dicho atributo teniendo en cuenta la información almacenada en la base de datos
- i. El viaje 2 pasa por Madrid en sexto lugar, y estará allí 4 días
- j. Borra todas las ofertas de los viajes que pasen por Madrid
- k. Crea una nueva tabla viaje2 con dos atributos: cod_via y numciudades. Donde vamos a almacenar para cada viaje el número de ciudades por los que pasa. Inicializa el contenido de la tabla con la información de la que disponemos en la base de datos

14. Dado el siguiente modelo relacional de una Base de Datos sobre pintores, escribe las sentencias sql que den respuesta a las siguientes preguntas:

PINTOR (CodP, nombre, fecha_nacimiento)

ESTILOS(CodE, descripción)

TIENEESTILOS(CodP, CodE)

CUADROS(CodC, titulo, fecha, CodP, CodE)

Con las claves externas

TIENEESTILOS.CodP → PINTOR.CodP

TIENEESTILOS.CodE → ESTILOS.CodE

CUADROS.CodP → PINTOR.CodP

CUADROS.CodE → ESTILOS.CodE

- a. Obtener los códigos de los pintores que tengan el estilo 'cubismo' pero que nunca hayan tenido el estilo 'impresionismo'
- b. Nombre y fecha de nacimiento del pintor que más cuadros tiene
- c. Nombre de los pintores que tengan un solo estilo (con exists)
- d. Obtener los nombres de los pintores que hayan hecho uso de todos los estilos
- e. (1) Obtener los códigos de los pintores que tengan los estilos impresionismo y realismo (con in)
- f. Obtener el código de los pintores cuyo único estilo sea impresionismo (con in y con exists)
- g. Estilo para el que menos cuadros tenemos
- h. Nombre del pintor que ha pintado cuadros de más estilos diferentes
- i. Pintores que tienen el estilo impresionista pero no han pintado ningún cuadro de dicho estilo (con in y con exists)
- j. Estilos que no hayan sido utilizados por ningún pintor (con exists)

15. Sobre la base de datos de Centro de salud, expresa las siguientes consultas en SQL

CENTRO_SALUD

<u>Barrio</u>	<u>Localidad</u>	Responsable
---------------	------------------	-------------

LOCALIDAD

<u>Nombre</u>	CodPostal	Concejal
---------------	-----------	----------

MEDICO

<u>Codigo</u>	Nombre	Barrio	Localidad
---------------	--------	--------	-----------

TELEFONOS

Telefono	codMedico
----------	-----------

TRABAJA

<u>CodMedico</u>	<u>Barrio</u>	<u>Localidad</u>
------------------	---------------	------------------

ATIENDE

<u>Localidad</u>	<u>Barrio</u>	<u>LocCentro</u>
------------------	---------------	------------------

ASEGURADO

<u>CodOsakidetza</u>	Nombre	Ap1	Ap2	LocEmpad	Medico	Barrio	LocCentro
----------------------	--------	-----	-----	----------	--------	--------	-----------

- Escribe las sentencias necesarias para crear las tablas. ¿En qué orden han de crearse ?
- Queremos añadir los siguientes atributos en las tablas
 - Fecha de nacimiento de los asegurados y de los médicos
 - Fecha desde la que está trabajando en cada centro de salud
 - Horario de apertura de los centros de salud (hora de inicio y hora de fin)
 - Salario de los médicos
- Diferentes localidades en las que hay algún centro de salud
- Número de localidades diferentes en las que hay algún centro de salud
- Número de localidades existentes en la base de datos
- Localidades con nombre de 5 letras
- Localidades con nombre compuesto de al menos dos palabras
- Número de localidades cuyo nombre empiece por A
- Número de pacientes asignados a cada médico (0,1,2...)
- Localidades para las que no hayamos guardado la información de su concejal
- Pares de asegurados cuyo primer apellido empiece por p, sin repeticiones
- Nombre de los asegurados ordenados por el primer apellido de manera ascendente
- Media de salario de los médicos
- Media de salario de los médicos de Vitoria
- Diferentes salarios almacenados
- Nombre de localidades junto con los barrios en los que tiene centro de salud, los que no tienen centro de salud también han de aparecer
- Crea una tabla con nombre de localidad y barrio junto con número de asegurados para cada barrio en el que hay centro de salud. Rellena la información de esta tabla con la existente en la base de datos
- Elimina las localidades sin concejal asignado
- Nombre de los asegurados ordenados por el primer apellido de manera descendente
- Algunos médicos tienen asignados asegurados que están empadronados en una localidad que no es la del centro de salud al que el médico está adscrito. Deseamos conocer las tripletas formadas por código y nombre de esos médicos y localidad de empadronamiento del asegurado. El resultado no deberá contener filas repetidas.

- u) De cada centro de salud deseamos conocer: Nombre del responsable, localidad en la que se ubica, número de asegurados y número de médicos adscritos. El resultado no deberá contener filas repetidas.

16. Dada la siguiente base de datos sobre centros de salud

CENTRO_SALUD				LOCALIDAD			
<u>Barrio</u>	<u>Localidad</u>	Responsable		<u>Nombre</u>	CodPostal	Concejal	
MEDICO				TELEFONOS			
<u>Codigo</u>	Nombre	Barrio	Localidad	Telefono	codMedico		
TRABAJA				ATIENDE			
<u>CodMedico</u>	<u>Barrio</u>	<u>Localidad</u>		<u>Localidad</u>	<u>Barrio</u>	<u>LocCentro</u>	
ASEGURADO							
CodOsakidetza	Nombre	Ap1	Ap2	LocEmpad	Medico	Barrio	LocCentro

- Todos los nombres existentes en la base de datos (los de médicos, localidades y asegurados)
- Los nombres de médico diferentes a todos los nombres de los asegurados
- Nombres de médico cuyo sea igual al de algún asegurado almacenado en la base de datos
- Nombres de médico cuyo nombre sea igual al de alguno de sus pacientes
- La médico con nombre 'Leire' se jubila y queremos asignar sus pacientes a Karmen
- Nombre de localidad junto con el nombre de su concejal, si no lo tiene mostrar 'no tiene concejal almacenado'. Este ejercicio hay que hacerlo de tres maneras diferentes.
- Nombre de las localidades cuyos centros de salud atienden únicamente a asegurados empadronados en la propia localidad.
- Médicos sin pacientes asignados
- Nombre de los médicos de cabecera con más pacientes asignados
- Todos los nombres existentes en la base de datos(los de médicos, localidades y asegurados)
- Los diferentes nombres de médico que sean diferentes de todos los nombres de asegurados
- Nombres de médico cuyo nombre sea igual al de algún asegurado almacenado en la base de datos
- Nombres de médico cuyo nombre sea igual al de alguno de sus pacientes
- Información de los centros de salud que tengan 20 o 30 asegurados
- Información de los centros de salud que tengan entre 20 y 30 asegurados
- Médicos con 20 pacientes asignados
- todos los médicos de cabecera pasan a cobrar 100 euros más que el médico med1
- La edad de todos los asegurados
- Médicos que cumplan años el día que ejecutes la sentencia

- t. Médicos que cumplan años el mes que ejecutes la sentencia
- u. Tiempo que está abierto cada centro de salud
- v. Médicos que empezaron a trabajar en algún centro de salud cuando tenían más de veinte años
- w. Centro de salud que esté ahora abierto

SQL y Álgebra relacional examen

1. Tenemos una base de datos con las siguientes relaciones. Las claves primarias están subrayadas. En usos el atributo U hace referencia al atributo de mismo nombre de usuarios, el p al del mismo nombre de programas y el O al mismo atributo de ordenadores.
PROGRAMAS(P, Memoria, SO, Distribuidor)
USUARIOS (U, EDAD, SEXO)
ORDENADORES (O, MODELO, SO, CAPACIDAD)
USOS(U, P, O, TIEMPO) --el tiempo se da en horas
 - a. Expresar en términos de álgebra relacional la secuencia de operaciones para efectuar las siguientes consultas:
 - *Obtener los usuarios que usan al menos todos los programas del distribuidor D1
 - *Obtener los programas que sólo son usados por el usuario U5
 - *Obtener los modelos de los ordenadores que son usados por personas mayores de 30 años durante más de 3 horas
 - * Indica los problemas que pueden surgir al introducir, borrar o modificar tuplas de las tablas usos y programas del modelo relacional inicial
 - b. Dar las sentencias SQL que den respuesta a las siguientes consultas
 - *Número de programas diferentes usados por cada usuario (0,1,2)
 - *Usuarios que no hayan usado programas de los distribuidores que empiezan por D
2. Tenemos una base de datos con las siguientes relaciones. Las claves primarias tienen una almohadilla al lado. En suministro E# hace referencia al atributo E# de editorial, L# al L# de libro y P# a P# de papelería
EDITORIAL (E#, NOMBRE, CIUDAD)
LIBRO (L#, TITULO, AUTOR, AÑO)
PAPELERIA (P#, PNOMBRE, CIUDAD)
SUMINISTRO (E#, L#, P#, CANTIDAD)
 - a. Da respuesta algebraica a las siguientes consultas
 - *Obtener los nombres de las papelerías que son suministradas por cualquier editorial de Madrid
 - *Obtener las editoriales que suministran a las papelerías P1 y P2 libros publicados en el año 1978
 - *Obtener los nombres de las papelerías suministradas completamente por la editorial E1
 - *Obtener los títulos de los libros que no han sido suministrados a ninguna papelería
 - b. (1)Escribe la sentencia SQL estándar para crear la tabla suministro. Explica las diferentes opciones que tienes al definir las claves externas
 - c. (1) Crea una vista que contenga los títulos y autores de los libros. Inserta nuevas tuplas usando únicamente dicha vista
 - d. (2.5) Dar respuesta en SQL a las siguientes consultas. No se puede usar nada que no se haya visto en clase, especialmente las select imbricadas.
 - *Nombre de papelería junto con el número de editoriales que la suministran (0,1,2....) ordenados de manera descendente por nombre de papelería
 - *Introduce el libro la sombra del viento de Carlos Ruiz Zafón
 - *En la tabla papelería introduce un atributo descuento con 0 como valor por defecto
 - *Editoriales que no tengan el mismo nombre que ningún autor y que empiece por p

Ejercicios SQL de exámenes

1. El siguiente esquema relacional representa el sistema de información de una cooperativa de taxis:

Taxi(matrícula, tfno, km_rev)

Taxista(dni, nombre, tfno, dirección, edad)

Taller(codt, nombre, tfno, responsable)

Conduce(dni, matrícula, fecha, h_inicio, h_fin)

Revisión(codt, matrícula, fecha)

donde los atributos tienen el siguiente significado:

Taxi :	<i>matrícula</i> : matrícula del coche	<i>tfno</i> : teléfono del coche
	<i>km_rev</i> : kilómetros a los que se tiene que hacer la próxima revisión	
Taxista:	<i>dni</i> : D.N.I. del taxista	<i>nombre</i> : nombre del taxista
	<i>edad</i> : del taxista	<i>tfno</i> : teléfono particular del taxista
	<i>dirección</i> : dirección del taxista	
Taller:	<i>codt</i> : código del taller	<i>nombre</i> : nombre del taller
	<i>tfno</i> : teléfono del taller	<i>responsable</i> : persona responsable del taller
Conduce	el taxista de DNI <i>dni</i> condujo (o conduce) el taxi de matrícula <i>matrícula</i> el día <i>fecha</i> desde las <i>h_inicio</i> hasta las <i>h_fin</i>	
Revisión	el taller de código <i>codt</i> revisó el taxi de matrícula <i>matrícula</i> el día <i>fecha</i>	

¿Qué condiciones cumplen los taxis cuyos datos se obtienen como resultado de la siguiente sentencia SELECT?

```
SELECT MATRICULA FROM CONDUCE
INTERSECT
SELECT MATRICULA FROM REVISION.
```

Escriba en SQL las siguientes consultas

- 1.1. Obtener cuántos taxis que han sido conducidos por algún taxista de menos de 30 años no han sido revisados nunca.
- 1.2. Escribir una consulta que permita saber cuántos taxis ha revisado cada taller cada día que haya revisado más de 5 taxis. Del taller especificar el código y el nombre, del día especificar la fecha.
- 1.3. Obtener la matrícula y el teléfono de los taxis que hayan sido conducido por algún conductor con edad superior a la edad media de todos los taxistas.
- 1.4. Obtener la matrícula y el teléfono de los taxis que hayan sido revisados al menos en dos talleres diferentes.

2. Sea el siguiente esquema relacional de una base de datos para la gestión de las misiones espaciales a nivel mundial

Astronauta (cod_astro, nombre, dirección, teléfono, país, num_viajes);

Científico(cod_cien, especialidad)

Nave (cod_nav, nombre, coste)

Base(cod_base, dirección, país)

Viaje(cod_via, cod_nav, cod_base, fecha_salida, fecha_llegada, nivel_éxito)

Tripulante(cod_astro, cod_via)

donde los atributos tienen el siguiente significado:

Astronauta:

cod_astro: código identificador del astronauta

nombre: nombre del astronauta *dirección*: domicilio del astronauta

teléfono: teléfono de localización del astronauta

país: país de residencia del astronauta

num_viajes: número total de misiones realizadas por el astronauta

Científico:

cod_cien: código identificador del científico

especialidad: especialidad en la que destaca el científico

Nave:

cod_nav: código identificador de la nave

nombre: nombre de la nave *coste*: valor económico de la nave

Base:

cod_base: código identificador de la base de lanzamiento

dirección: localización de la base de lanzamiento

país: país en el que se encuentra ubicada la base de lanzamiento

Viaje:

cod_via: código identificador del viaje.

cod_nav: código identificador de la nave que va a ser lanzada

cod_base: código identificador de la base de lanzamiento

fecha_salida: fecha de lanzamiento *fecha_llegada*: fecha de regreso

nivel_éxito: valor entre 0 (misión fracasada) y 5 (misión exitosa)

Tripulante:

cod_astro: código identificador del astronauta

cod_via: código identificador del viaje en el que va a ser lanzado

Escribe en SQL las siguientes consultas

2.1. Obtener el número total de astronautas que no son científicos.

2.2. Obtener para cada nave cuántos viajes ha hecho. Incluir en el resultado el nombre de la nave y las naves que no han hecho ningún viaje.

2.3. Obtener el código de los viajes en los que todos los astronautas son científicos.

2.4. Obtener el nombre de las naves que sólo han partido desde una base.

2.5. Obtener el país desde cuyas bases ha habido más lanzamientos.

3. Dada la base de datos que se describe a continuación

Actividad (**cod_act**: d_act, nombre: d_nom2, descripción: d_des) Uni: { nombre }
Cooperante(**dni**:d_dni, nombre: d_nom, edad:d_edad, enseña, d_act, aprende: d_act)
CAj: { enseña } ® Actividad f(enseña) = cod_act
CAj: { aprende } ® Actividad f(aprende) = cod_act
País(**cod_país**: d_país, nombre: d_nom1, problemas: d_prob)
Necesita(**cod_país**: d_país, **cod_act**: d_act, urgencia: d_urg)
CAj: { cod_país } ® País CAj: { cod_act } ® Actividad
Viajar(**dni**: d_dni, **cod_país**: d_país, fecha: d_fecha)
CAj: { dni } ® Cooperante CAj: { cod_país } ® País

Donde los atributos tienen el siguiente significado:

Cooperante: *dni*: D.N.I. del cooperante *nombre*: nombre del cooperante
edad: edad del cooperante *enseña*: actividad que enseña
Actividad: *cod_act*: código de la actividad *nombre*: nombre de la actividad
descripción: descripción de la actividad
País: *cod_país*: código del país *nombre*: nombre del país
problemas: situación política del país
Necesita: el país de código *cod_pais* necesita la actividad de código *and_act* con grado de prioridad *urgencia*
Viajar: el cooperante con D.N.I. *dni* se ha apuntado para viajar al país de código *cod_pais* en la fecha *fecha*

- Obtener el nombre de los países que necesitan una o más actividades, todas ellas con urgencia 'Máxima'.
- Obtén el número de actividades con urgencia máxima que necesita cada país. Queremos que aparezcan los nombres de todos los países almacenados en la base de datos
- Para cada país que reciba más de 10 cooperantes, indica su nombre y cuántos cooperantes diferentes recibe
- Borra los países que necesitan más de tres actividades

4. El Siguiente esquema relacional representa la información que se maneja en una inmobiliaria

Persona(nif, nombre, dir)

Propietario(nif, categoría)

donde la categoría es un valor numérico entre 1 y 4 y el nif es una clave extranjera que referencia al nif de persona

Alquilado(nif, moroso)

donde moroso es un valor booleano y el nif es una clave extranjera que referencia al nif de persona

Inmueble (código, dir, tipo, prop, alq, fecha_alta)

Donde prop es una clave extranjera que hace referencia al nif de propietario y alq es una clave extranjera que referencia al nif de alquilado. Los tipos de inmueble que tenemos almacenados son: comercial, piso, adosado, chalet. La fecha_alta indica cuando ocupó el alquilado el inmueble

Reserva(número, código, nif, fecha_ini, fecha_fin)

Donde código referencia al código del inmueble, el nif hace referencia al nif de persona.

Usando el modelo entidad-relación anterior responde usando SQL:

- 4.1. Obtener para cada inmueble (indicando su código y dirección) el número de personas diferentes para las que existe alguna reserva del mismo en la base de datos

- 4.2. Obtener el nif y el nombre de los propietarios tales que todos sus inmuebles tengan tres o más reservas en la base de datos

NOTA IMPORTANTE: se piden los propietarios que tengan al menos un inmueble.

- 4.3. Obtener el nombre del propietario y el código de los inmuebles sin alquilar (que no tienen alquilado) para los que no hay ninguna reserva en la base de datos

- 4.4. Obtén los nombres de las personas con mayor número de reservas.

5. Tenemos el siguiente modelo relacional que representa la información sobre un casting para un concurso de actores:

Actor(dni_act, nombre, fecha_ncto)

Miembro_Jurado(dni_jur, nombre, especialidad)

Papel(cod_pap, descripción, duración)

Actuación(cod_pap, dni_act, fecha)

Donde cod_pap es clave externa que hace referencia a cod_pap de papel y dni_Act es una clave externa que hace referencia a dni_act de actor.

El actor de dni dni_act ha interpretado el papel de código cod_pap el día fecha

Ficha_puntuación(cod_ficha, nota, cod_pap, dni_act, dni_jur)

Donde dni_jur es una clave externa que hace referencia a dni_jur de miembro jurado y cod_pap y dni_act hacen referencia a la tabla actuación.

Hay una ficha de código cod_ficha que rellena el miembro_jurado dni_jur y en la que se otorga nota puntos a la actuación de actor dni_Act en el papel de cod_pap

Ejemplo:

Sea BD la siguiente base de datos del esquema anterior:

Actor			Miembro_Jurado			Papel		
Dni_act	Nombre	edad	dni_jur	Nombre	especialidad	Cod_pap	Descripción	Duración
456	Luisa	18	222	Marta		1	Chica	20
678	María	21	333	Pablo		2	Chico	17
123	Juan	32	444	Aurora		3	Malo	7
						4	Amiga	3

Actuación			Ficha_Puntuación				
Cod_pap	Dni_act	Fecha	cod_ficha	nota	cod_pap	dni_act	dni_jur
1	456	3-3-02	1	3	1	678	222
4	456	3-3-02	2	5	4	456	333
1	678	3-3-02	3	7	-	456	333
2	123	3-3-02	4	6	2	123	444
			5	8	1	678	222
			6	6	3	678	444
			7	5	-	-	333

Teniendo en cuenta el modelo anterior, responde a los siguientes ejercicios con sentencias SQL

- 5.1. ¿Cuántas consultas hay que no hayan sido bien resueltas por nadie?
 - 5.2. Obtener el DNI y el nombre de los que tengan la nota de base de datos suspendida pero hayan resuelto bien al menos tres consultas
 - 5.3. Obtener el nombre y el grado de las relaciones que son necesitadas más de 25 veces entre todas las consultas que las necesitan indicando también ese número de veces
 - 5.4. Obtener de los alumnos que hayan resuelto al menos una consulta, el dni el nombre de los que han resuelto bien todas las consultas que han resuelto
 - 5.5. Obtener el DNI y el nombre del alumno (o alumnos) que hayan resuelto bien más consultas
6. El siguiente modelo relacional almacena información relativa a unos ejercicios realizados entre alumnos de la asignatura de bases de datos para medir su pericia en la resolución de consultas SQL. Las relaciones son las siguientes.

Consulta(cod_con, enunciado, dificultad)

Relación(nom_rel, grado, cardinalidad)

Alumno(dni, nombre, edad, nota_bda)

Necesita(cod_con, nom_rel, veces)

Cod_con es una clave externa que referencia a consulta y nom_rel es una clave externa que hace referencia a relación.

Resuelve(cod_con, dni, tiempo, correcta)

Cod_con es una clave externa que referencia a consulta y dni es una clave externa que referencia a alumno.

Donde los atributos tienen los siguientes significados:

Consulta: cod_con: código de la consulta enunciado: texto enunciado
Dificultad: baja, media, alta
Relación: nom_rel: nombre de la relación grado: número atributos
Cardinalidad: número de tuplas
Alumnos: dni:dni del alumno nota_bda: obtenida en el último examen
Necesita: La consulta de código cod_con necesita para su resolución la relación de nombre nom_rel al menos un número veces de veces
Resuelve: El alumno de DNI dni ha resuelto la consulta de código cod_con en tiempo segundos. La solución dada es buena si correcta toma el valor sí. En caso contrario tomará el valor no

Sobre la base de datos anterior, responde usando SQL

- 6.1. ¿Cuántas consultas hay que no hayan sido bien resueltas por nadie?
- 6.2. Obtener el DNI y el nombre de los que tengan la nota de bases de datos suspendida pero hayan resuelto bien al menos tres consultas
- 6.3. Obtener el nombre y el grado de las relaciones que son necesitadas más de 25 veces entre todas las consultas que las necesitan indicando también ese número de veces
- 6.4. Obtener, de los alumnos que hayan resuelto al menos una consulta, el dni y el nombre de los que han resuelto bien todas las consultas que han resuelto

7. Dado el siguiente modelo relacional



Donde tenemos las claves externas que se mencionan a continuación:

HotelViaje → destinoV, fecha salida referencian a la tabla viaje

Idhotel referencia a la tabla hotel

Excursión opcional → destinoV, fecha salida referencian a la tabla viaje

Idioma → dni referencia a la tabla guía

HotelViajeCliente → destinoV, fecha salida, idhotel referencian a la tabla hotelviaje

Excur_opc_cliente → destinoV, fecha salida, numero referencian a la tabla

excursiónOpcional

dni regencia a la tabla cliente

Se pide realizar las siguientes consultas en SQL

- a. Los nombres de guías que aparecen también entre los clientes
- b. Clientes que se hayan apuntado a todas las excursiones opcionales de algún viaje
- c. Clientes que no se hayan apuntado aún a ningún viaje (hazlo de tres maneras diferentes. Una en la que aparezca exists otra con in y finalmente una con all)

8. Tenemos una base de datos de viajes reflejada por las siguientes tablas:

VIAJE	DestinoV	FechaSalida	Días	PrecioDía	CiudadSalida	DNI	HOTEL	Id_Hotel	Nombre	Localidad	Capacidad	
HOTEL VIAJE	DestinoV	FechaSalida	Id_Hotel	CLIENTE	DNI	Nombre	Dirección	N_Tfno	GUÍA	DNI	Nombre	N_Tfno
EXCURSIÓN OPCIONAL	DestinoV	FechaSalida	Número	DestinoExcursión	HoraSalida	LugarSalida	Precio	IDIOMA	DNI	Idioma		
HOTEL VIAJE CLIENTE	DestinoV	FechaSalida	IdHotel	DNI	NúmeroNoches		EXCUR OPC CLIENTE	DestinoV	FechaSalida	Número	DNI	

Donde:

Se pide realizar las siguientes consultas en SQL

- Viajes (destino,fechasalida,días) que no tengan ninguna excursión opcional asignada con exists , con in y con all
 - Los nombres de los clientes y de los guías
 - Hoteles por los que pasan clientes de más de 4 viajes diferentes
9. Tenemos una base de datos de copisterías reflejada por las siguientes tablas:

COPISTERÍA	<u>NIF</u>	Dirección	Precio	ASIGNATURA	<u>IdAsignatura</u>	Curso	Carrera	NIF		
ALUMNO	<u>DNI</u>	Clave	Dirección	APUNTES	<u>IdAsignatura</u>	<u>Tema</u>	NúmeroHojas			
CURSA	<u>DNI</u>	<u>IdAsignatura</u>		FICHA	<u>DNI</u>	<u>IdAsignatura</u>	<u>Tema</u>	<u>FechaPetición</u>	NúmeroCopias	FechaRecogida

Donde tenemos las siguientes claves externas:

Apuntes → Idasignatura referencia a la tabla asignatura

Cursa → dni a la tabla alumno

Idasignatura referencia a la tabla asignatura

Ficha → dni a la tabla alumno

Idasignatura, tema a la tabla apuntes

Se pide realizar las siguientes consultas en SQL

- Alumnos que cursan todas las asignaturas de Segundo
- Alumnos que no cursan ninguna asignatura de Segundo(de cuatro maneras diferentes. En cada una, una de las siguientes opciones: exists, in,all y una operación de conjuntos)