

Enunciados y soluciones

Ejercicios SQL

Convenciones

En las soluciones a los ejercicios propuestos se siguen las siguientes convenciones:

- Las soluciones se encontrarán en el tipo de letra Courier-New
- Las cláusulas SQL se encontrarán en letra negrita y mayúsculas: **SELECT**
- Los nombres de campos de tablas en letra Courier-New y precedidos del nombre de la tabla a que pertenecen separando ambos con un punto: `empleado.Salario`
- Los nombres de las tablas en una sentencia SQL, se escribirán en minúscula y tipo de letra Courier-New.
- Los nombres de campos empiezan siempre por letra mayúscula y el resto en minúscula, sin acentos. Los nombres de tablas, cuando se utilizan en una sentencia SQL se escriben en minúsculas.
- Las constantes se escribirán en tipo Courier-New, cursiva y negrita.
- Cuando el nombre de un campo, esté compuesto de más de una palabra, se utilizará para describirlo la notación SmallTalk, es decir se unirán ambas palabras pero la primera letra de cada una de ellas será en mayúsculas: `NombreEmpleado`
- Los tipos de datos para los diferentes campos, se asignarán de forma implícita de acuerdo con el contenido que se espera de ellos.

Ejercicios Consultas SQL

Ejercicio 1: Seguros

Consideremos la siguiente estructura de tablas:

EMPLEADO (DNI, NombreEmpleado, Salario)

VENDEDOR (NºVendedor, Zona, *DNI*)

POLIZA (NºPoliza, Importe, Beneficiario, *NºVendedor*, Fecha)

EMPLEADO-JEFE (*DNI*, *DNIJefe*)

Construir las sentencias de consulta SQL siguientes:

a) Nombre de los vendedores de la zona Norte

```
SELECT NombreEmpleado
FROM vendedor, empleado
WHERE vendedor.DNI = empleado.DNI
AND Zona = "Norte"
```

b) Nombre del jefe del vendedor con número de vendedor 113.654

```
SELECT NombreEmpleado
FROM vendedor, empleado, empleado-jefe
WHERE vendedor.DNI = empleado-jefe.DNI
AND vendedor.NºVendedor = 113.654
AND empleado-jefe.DNIjefe = empleado.DNI
```

En esta consulta se aprovecha que vendedor es una ISA de empleado para establecer una relación entre la clave foránea de vendedor hacia jefe y la clave foránea de empleado-jefe hacia empleado a través de la clave principal de empleado: DNI

Si queremos establecer un recorrido paso por paso a través de las conexiones entre clave principal de una con la foránea de la otra, debemos utilizar dos veces la tabla empleado, una jugando el papel de vendedor empleado y otra jugando el papel de jefe. Debemos por tanto renombrar la tabla.

```
SELECT jefe.NombreEmpleado
FROM vendedor, empleado AS subordinado, empleado AS jefe,
empleado-jefe
WHERE jefe.DNI = empleado-jefe.DNIjefe
AND empleado-jefe.DNI = subordinado.DNI
AND subordinado.DNI = vendedor.DNI
AND vendedor.NºVendedor = 113.654
```

c) Nº de póliza e importe de póliza de las ventas este año por los vendedores cuyo jefe es Pablo Collado.

```
SELECT NºPoliza, Importe
FROM empleado, empleado-jefe, vendedor, poliza
WHERE empleado.nombre = "Pablo Collado"
AND empleado.DNI = empleado-jefe.DNIjefe
AND empleado-jefe.DNI = vendedor.DNI
AND vendedor.NºVendedor = poliza.NºVendedor
```

Ejercicio 2: Empresa2

Consideremos la siguiente estructura de tablas:

DEPARTAMENTO (NombreDepartamento, N°Empleados)

EDIFICIO (IdEdificio, Nombre, N°Despachos, *IdPoligono*)

POLÍGONO (IdPoligono, Nombre, Ciudad)

DEPARTAMENTO-EDIFICIO (NombreDepartamento, IdEdificio, N°Despachos)

Construir las sentencias de consulta SQL siguientes:

- a) Nombre de los edificios, en los que el departamento de “Contabilidad” tiene más de 5 despachos asignados.

```
SELECT Nombre
FROM departamento-edificio, edificio
WHERE departamento-edificio.IdEdificio = edificio.IdEdificio
AND departamento-edificio.NombreDepartamento = "Contabilidad"
AND departamento-edificio.N°Despachos > 5
```

- b) Nombre de los departamentos y número de despachos de cada uno que estén ubicados en el edificio “La Campana”.

```
SELECT departamento-edificio.NombreDepartamento, N°despachos
FROM departamento-edificio, edificio
WHERE departamento-edificio.IdEdificio = edificio.IdEdificio
AND edificio.Nombre = "La Campana"
```

- c) Nombre de los departamentos con representación en “Logroño”.

```
SELECT departamento-edificio.NombreDepartamento
FROM departamento-edificio, edificio, poligono
WHERE departamento-edificio.IdEdificio = edificio.IdEdificio
AND edificio.IdPoligono = poligono.IdPoligono
AND poligono.Ciudad = "Logroño"
```

- d) Nombre edificio y número de despachos de los edificios de los polígonos “Cantabria”, y “La Portalada”.

```
SELECT edificio.nombre, edificio.N°Despachos
FROM edificio, poligono
WHERE edificio.IdPoligono = poligono.IdPoligono
AND poligono.Nombre IN ("Cantabria", "La Portalada")
```

Ejercicio 3: Ríos de España

Consideremos la siguiente estructura de tablas:

COMUNIDADAUTONOMA(NombreAutonomia, N°Habitantes)

CIUDAD (IdCiudad, Nombre, N°Habitantes, *NombreAutonomia*)

RIO (NombreRio, Caudal, Longitud)

COMUNIDADAUTONOMA-RIO (NombreAutonomia, NombreRio, N°Kilometros)

RIO-CIUDAD (NombreRio, IdCiudad)

Construir las sentencias de consulta SQL siguientes:

a) **Nombre y número de habitantes de las ciudades de La Rioja.**

```
SELECT ciudad.Nombre, ciudad.NºHabitantes
FROM ciudad
WHERE ciudad.nombreAutonomia = "La Rioja"
```

b) **Nombre y longitud de los ríos que pasan por la Comunidad de Aragón.**

```
SELECT rio.NombreRio, rio.Longitud
FROM rio, comunidadautonoma-rio
WHERE rio.NombreRio = comunidadautonoma-rio.NombreRio
AND comunidadautonoma-rio.NombreAutonomia = "Aragón"
```

c) **Nombre de los ríos que pasan parte por Castilla-La Mancha y parte por Andalucía.**

```
SELECT NombreRio
FROM comunidadautonoma-rio
WHERE NombreRio IN (SELECT NombreRio
                     FROM comunidadautonoma-rio
                     WHERE NombreAutonomia = "Andalucía")
AND NombreRio IN (SELECT NombreRio
                  FROM comunidadautonoma-rio
                  WHERE NombreAutonomia = "Castilla La Mancha")
```

Esta consulta, puede resolverse de otra forma, según se detalla a continuación:

```
SELECT NombreRio
FROM comunidadautonoma-rio
WHERE NombreAutonomia = "Andalucía"
OR NombreAutonomia = "Castilla La Mancha"
GROUP BY NombreRio
HAVING COUNT(NombreAutonomia) = 2
```

d) **Suma de los caudales de los ríos que pasan por Navarra.**

```
SELECT sum(Caudal) AS SumaCaudalNavarra
FROM comunidadautonoma-rio, rio
WHERE rio.NombreRio = comunidadautonoma-rio.NombreRio
AND comunidadautonoma-rio.NombreAutonomia = "Navarra"
```

En la anterior consulta hemos dado un nombre al campo que calculamos que refleja qué estamos obteniendo. Podemos expresar el nombre de la Comunidad Autónoma a través del campo en el que se encuentra ubicado dicho nombre a través de la siguiente consulta. Observar en la misma que debemos agrupar por NombreAutonomia al aparecer éste en la cláusula **SELECT**

```
SELECT comunidadautonoma-rio.NombreAutonomia, sum(Caudal) AS
SumaCaudal
FROM comunidadautonoma-rio, rio
WHERE rio.NombreRio = comunidadautonoma-rio.NombreRio
AND comunidadautonoma-rio.NombreAutonomia = "Navarra"
GROUP BY comunidadautonoma-rio.NombreAutonomia
```

Ejercicio 4: Vehículos

Consideremos la siguiente estructura de tablas:

CONDUCTOR (DNI, Nonbre, Apellido, N°Años)

VEHICULO (Matricula, Marca)

COCHE (Matricula, N° Puertas)

MOTO (Matricula)

CAMION (Matricula, Tara)

CONDUCTOR-VEHICULO (DNI, Matricula)

Construir las sentencias de consulta SQL siguientes:

a) DNI de los conductores de motos

```
SELECT conductor-vehiculo.DNI
FROM conductor-vehiculo, moto
WHERE conductor-vehiculo.Matricula = moto.Matricula
```

b) Marca de los vehículos que conduce Pepe Pérez.

```
SELECT vehiculo.Marca
FROM conductor-vehiculo, conductor, vehiculo
WHERE conductor.Nombre = "Pepe"
AND conductor.Apellido = "Pérez"
AND conductor.DNI = conductor-vehiculo.DNI
AND conductor-vehiculo.Matricula = vehiculo.Matricula
```

c) Marca y tara del camión conducido por la persona cuyo DNI sea "11111111".
Incluir también el número de años con el carnet de conducir.

```
SELECT vehiculo.Marca, camion.Tara, conductor.n°años
FROM conductor-vehiculo, conductor, vehiculo, camion
WHERE conductor.DNI = "11111111"
AND conductor-vehiculo.DNI = conductor.DNI
AND conductor-vehiculo.Matricula = vehiculo.Matricula
AND vehiculo.Matricula = camion.Matricula
```

Ejercicio 5: Autobuses

Consideremos la siguiente estructura de tablas:

AUTOBUS (N°Linea, N°Asientos)

CONDUCTOR (DNI, Nombre, Apellido, N°Años, *N°Linea*)

PARADA (IDParada, Calle, Numero, Marquesina?)

AUTOBUS-PARADA (*N°Linea*, IDParada, 1ªHoraLlegada, Frecuencia)

Construir las sentencias de consulta SQL siguientes:

a) Lineas que tienen parada en Vara de Rey

```
SELECT autobus-parada.N°Linea
FROM autobus-parada, parada
WHERE autobus-parada.IDParada = parada.IDParada
AND parada.Direccion = 'Vara de Rey'
```

b) Nº asientos del autobús que conduce Pepe Pérez.

```
SELECT autobus.NºAsientos
FROM conductor, autobus
WHERE conductor.Nombre = "Pepe"
AND conductor.Apellido = "Pérez"
AND conductor.NºLinea = autobus.NºLinea
```

c) ¿Todas las paradas de la línea 30 poseen marquesina?.

Para responder a la pregunta se decide contar las paradas de la línea 30 que no tengan marquesina. Si ese número es $< > 0$, entonces la respuesta a la pregunta es NO, en otro caso será SI.

```
SELECT count(*)
FROM autobus-parada, parada
WHERE autobus-parada.NºLinea = 30
AND autobus-parada.IDParada = parada.IDParada
AND parada.Marquesina? = No
```

d) Líneas que tienen parada en Vara de Rey 2 y Vara de Rey 40

```
SELECT NºLinea
FROM autobus-parada
WHERE autobus-parada.IDParada IN (SELECT IDParada
FROM parada
WHERE Calle = "Vara de Rey"
AND Numero = 2)
AND autobus-parada.IDParada IN (SELECT IDParada
FROM parada
WHERE Calle = "Vara de Rey"
AND Numero = 40)
```

Ejercicio 6: Seguros2

Consideremos la siguiente estructura de tablas:

PERSONA (DNI, Nombre, Apellido1, Apellido2)

PISO (IDPiso, Calle, Numero, Ciudad, N°m2, *DNIPropietario*)

SEGURO (N°Poliza, Tipo, CuotaBasica, *DNIAsegurado*, *IDPiso*, CantidadAsegurada)

Construir las sentencias de consulta SQL siguientes:

a) Datos personales de aquellas personas cuyo primer apellido es García.

```
SELECT Nombre, Apellido1, Apellido2
FROM persona
WHERE Apellido1 = "García"
```

b) Dirección completa de los pisos de más de 90 metros cuadrados.

```
SELECT Calle, Numero, Ciudad
FROM piso
WHERE N°m2 > 90
```

c) Nº de las pólizas contratadas por Pepe Pérez.

```
SELECT N°Poliza
FROM seguro, persona
WHERE seguro.DNIAsegurado = persona.DNI
AND persona.Nombre = "Pepe"
AND persona.Apellido1 = "Pérez"
```

- d) N° de póliza y cantidad asegurada, de los seguros de los pisos cuyo propietario es Pepe Pérez.

```
SELECT N°Poliza, CantidadAsegurada
FROM seguro, piso, persona
WHERE seguro.IDPiso = piso.IDPiso
AND piso.DNIPropietario = persona.DNI
AND persona.Nombre = "Pepe"
AND persona.Apellido1 = "Pérez"
```

- e) N° de póliza y cantidad asegurada, de los seguros de hogar en los que el contratante del seguro coincide con el propietario del piso.

```
SELECT N°Poliza, CantidadAsegurada
FROM seguro, piso
WHERE seguro.IDPiso = piso.IDPiso
AND DNIPropietario = DNIAsegurado
AND tipo = "Hogar"
```

Ejercicio 7: Horarios

Consideremos la siguiente estructura de tablas:

ASIGNATURA (Nombre, Cuatrimestre, N°Aula)

PROFESOR (Nombre, Departamento)

AULA (N°Aula, Capacidad)

PIZARRA (N°Aula, Ancho, Alto)

AULA-NO-INFORMATICA (N°Aula)

AULA-DE-INFORMATICA (N°Aula, N°Ordenadores)

ASIGNATURA-PROFESOR (NombreAsignatura, NombreProfesor)

Construir las sentencias de consulta SQL siguientes:

- a) Nombre de las asignaturas que se imparten en el aula 108 durante el 1er Cuatrimestre.

Nótese que se asume que el tipo de datos del campo "Cuatrimestre" en la tabla "Asignatura" es un campo de tipo numérico, de ahí que se convierte la información proporcionada en el enunciado al tipo de datos tal y como se almacena en la tabla.

```
SELECT asignatura.Nombre
FROM asignatura
WHERE asignatura.N°Aula = 108
AND asignatura.Cuatrimestre = 1
```

- b) Nombre y cuatrimestre de las asignaturas que se imparten en aulas de informática con capacidad mayor de una veintena de alumnos

```
SELECT asignatura.Nombre, asignatura.Cuatrimestre
FROM asignatura, aula, aula-de-informatica
WHERE asignatura.N°Aula = aula.N°Aula
AND aula.Capacidad > 20
AND aula.N°Aula = aula-de-informatica.N°Aula
```

- c) Numero y capacidad de las aulas sin pizarra

```
SELECT aula.N°Aula, aula.Capacidad
FROM aula
WHERE aula.N°Aula NOT IN (SELECT pizarra.N°Aula
```

FROM pizarra)

- d) Nombres de los profesores del departamento de Matemáticas y Computación que den clase en aulas de informática. Para cada uno de ellos, incluir el nombre de la asignatura, el número de ordenadores y la capacidad del aula.

```
SELECT profesor.Nombre, asignatura.nombre, aula.capacidad, aula-
de-informatica.NºOrdenadores
FROM asignatura-profesor, asignatura, aula, aula-de-informatica
WHERE asignatura.Nombre = asignatura-profesor.NombreAsignatura
AND asignatura-profesor.NombreProfesor = profesor.Nombre
AND profesor.Departamento = "Matemáticas y Computación"
AND asignatura.NºAula = aula.NºAula
AND aula.NºAula = aula-de-informatica.NºAula
```

Ejercicio 8: Prensa

Consideremos la siguiente estructura de tablas:

PERIODICO(nombre)

ARTICULO(id-articulo, tipo, titulo, texto completo, resumen)

AUTOR(id-autor, nombre)

PERIODISTA(id-autor)

NOPERIODISTA(id-autor, profesion)

PERIODICO-ARTICULO(nombre-periodico, id-articulo, fecha, seccion)

ARTICULO-AUTOR(id-articulo, id-autor)

PERIODISTA-PERIODICO(id-autor, nombre-periodico, cargo)

- a) Listado de artículos (mostrando título y tipo) publicados el 13 de marzo de 2001 en el diario "La Rioja".

```
SELECT articulo.Titulo, articulo.Tipo
FROM articulo, periodico-articulo
WHERE articulo.Id-articulo = periodico-articulo.Id-articulo
AND fecha = #13/03/2001#
AND periodico-articulo.Nombre-periodico = "La Rioja"
```

- b) Mostrar por orden alfabético los nombres de periódicos en los que algún empresario haya publicado artículos (1 o más) en la sección de opinión.

```
SELECT DISTINCT periodico-articulo.Nombre-periódico
FROM periodico-articulo, articulo, artículo-autor, autor,
noperiodista
WHERE noperiodista.Profesion = "Empresario"
AND periodico-articulo.Seccion = "Opinión"
AND periodico-articulo.Id-articulo = articulo.Id-articulo
AND articulo.Id-articulo = articulo-autor.Id-articulo
AND articulo-autor.Id-autor = autor.Id-autor
AND autor.Id-autor = noperiodista.Id-autor
ORDER BY periodico-articulo.Nombre-periodico
```

- c) Verificar si todos los autores son periodistas o no periodistas.

```
SELECT autor.Id-autor
FROM autor
WHERE autor.Id-autor NOT IN (SELECT periodista.Id-autor
FROM periodista
UNION
SELECT noperiodista. Id-autor
FROM periodista)
```


- d) Nombres de periódicos en los que haya al menos 6 empleados. Mostrar también el número de empleados.

```
SELECT Nombre-periodico, COUNT(Id-autor) AS NumeroEmpleados
FROM periodista-periodico
GROUP BY nombre-periodico
HAVING COUNT(Id-autor) >= 6
```

- e) Mostrar los títulos de los artículos que hayan sido publicados simultáneamente (en la misma fecha) en “La Rioja” y “El Correo”.

```
SELECT articulo.Titulo
FROM articulo, periodico-articulo
WHERE periodico-articulo.nombre-periodico = "La Rioja"
OR periodico-articulo.nombre-periodico = "El Correo"
AND articulo.Id-articulo = periodico-articulo.Id-articulo
GROUP BY periodico-articulo.Fecha, articulo.Titulo
HAVING COUNT(periodico-articulo.Nombre-periodico) >= 2
```

Ejercicio 9: Vinos

Consideremos la siguiente estructura de tablas:

VINO(id-vino, marca, tipo-vino, año, graduacion, *nombre-bodega*)

BODEGA(nombre-bodega)

EMPRESA(nombre propietario, *nombre-bodega*)

COOPERATIVA(nº cooperativistas, *nombre-bodega*)

CATADOR(nombre-catador, medio)

UVA(tipo-uva)

VINO-UVA(id-vino, tipo-uva)

VINO-CATADOR(id-vino, nombre-catador, calificacion)

VINO-BODEGA(id-vino, nombre-bodega)

- a) Nombres de las bodegas de particulares.

```
SELECT bodega.Nombre-bodega
FROM bodega
WHERE bodega.Nombre-bodega NOT IN (SELECT empresa.Nombre-bodega
FROM empresa
UNION
SELECT cooperativa.Nombre-bodega
FROM cooperativa)
```

- b) Marca y graduación de vinos del año que se hayan elaborado con uva garnacha o tempranillo.

```
SELECT vino.Marca, vino.Graducion
FROM vino, uva, vino-uva
WHERE vino.Año = 2002
AND uva.Tipo = "garnacha"
OR uva.Tipo = "tempranillo"
AND vino.Id-vino = vino-uva.Id-vino
AND vino-uva.Tipo-uva = uva.Tipo-uva
```

- c) **Marca, tipo de vino, año de cosecha, graduación y nombre del catador de los vinos de la bodega “Magnífica”.**

```
SELECT vino.Marca, vino.Tipo-vino, vino-Año, vino.Graducion,
vino-catador.Nombre
FROM vino, vino-catador, vino-bodega
WHERE vino-bodega.Nombre = "Magnífica"
AND vino.Id-vino = vino-catador.Id-vino
AND vino.Id-vino = vino-bodega.Id-vino
```

- d) **Marcas de vinos (marca, tipo y año de cosecha) en los que para su elaboración se hayan utilizado uva garnacha juntamente con uva tempranillo (las dos).**

```
SELECT Marca, Tipo-vino, Año
FROM vino
WHERE Id-vino IN (SELECT Id-vino
                  FROM vino-uva
                  WHERE tipo-uva = "tempranillo")
AND Id-vino IN (SELECT Id-vino
                FROM vino-uva
                WHERE tipo-uva = "garnacha")
```

Ejercicio 10: Museos

Consideremos la siguiente estructura de tablas:

AUTOR(id autor, nombre-autor, nacionalidad)

OBRADARTE(titulo, *nombre-museo*, *ciudad*)

PINTURA(ancho, alto, tipo, *titulo*)

ESCULTURA(material, estilo, *titulo*)

SALA(id-sala, nombre-sala, *nombre-museo*, *ciudad*)

MUSEO(*nombre-museo*, *calle*, *nº*, *ciudad*, *pais*)

OBRADARTE-AUTOR(*titulo*, *id-autor*)

OBRADARTE-SALA(*titulo*, *id-sala*)

- a) **Relación de los títulos de las pinturas junto con el nombre del museo donde se localizan.**

```
SELECT pintura.Titulo, museo.Nombre-museo, museo.Ciudad
FROM pintura, obradarte, museo
WHERE pintura.Titulo = obradarte.Titulo
AND museo.Nombre-museo = obradarte.Nombre-museo
AND museo.ciudad = obradarte.ciudad
```

- b) **Nombres de los autores que hayan realizado al menos una pintura y una escultura.**

```
SELECT DISTINCT autor.nombre-autor
FROM autor, obradarte-autor, obradarte, pintura
WHERE autor.Id-autor = obradarte-autor.Id-autor
AND obradarte-autor.Titulo = obradarte.Titulo
AND obradarte.Titulo = pintura.Titulo
AND autor.nombre-autor IN
(SELECT DISTINCT autor.nombre-autor
 FROM autor, obradarte-autor, obradarte, escultura
 WHERE autor.Id-autor = obradarte-autor.Id-autor
 AND obradarte-autor.Titulo = obradarte.Titulo)
```

AND obradearte.Titulo = escultura.Titulo)

- c) **Títulos de las obras de arte registradas en España junto con el museo en el que se encuentran.**

```
SELECT obradearte.Titulo, obradearte.Nombre-museo
FROM obradearte
WHERE obradearte.Ciudad = "España"
```

- d) **Nombre del autor y nombre del museo en que se encuentran las pinturas al óleo.**

```
SELECT autor.nombre-autor, obradearte.Nombre-museo
FROM autor, obradearte-autor, obradearte, pintura
WHERE autor.Id-autor = obradearte-autor.Id-autor
AND obradearte-autor.Titulo = obradearte.Titulo
AND obradearte.Titulo = pintura.Titulo
AND pintura.tipo = "óleo"
```

- e) **Título, autor, nombre de la sala y museo de las obras de arte que se encuentran en Madrid.**

```
SELECT obradearte.titulo, autor.nombre-autor, obradearte-
sala.nombre-sala, obradearte.Nombre-museo, obradearte.Ciudad,
sala.nombre-sala
FROM autor, obradearte-autor, obradearte, obradearte-sala, sala
WHERE obradearte.Ciudad = "Madrid"
AND autor.Id-autor = obradearte-autor.Id-autor
AND obradearte-autor.Titulo = obradearte.Titulo
AND obradearte.Titulo = obradearte-sala.Titulo
AND obradearte-sala.Id-sala = sala.Id-sala
```

- f) **Relación de las salas que tienen el mismo nombre en el museo del Prado y en el Louvre.**

```
SELECT id-sala, nombre-sala
FROM sala
WHERE nombre-sala IN (SELECT nombre-sala
FROM sala
WHERE nombre-museo = "Prado")
AND nombre-sala IN (SELECT nombre-sala
FROM sala
WHERE nombre-museo = "Louvre")
```

Ejercicio 11: Publicidad

Consideremos la siguiente estructura de tablas:

CAMPAÑAPUBLICITARIA(tema, presupuesto, *id-anunciante*)

ANUNCIO(slogan, *tema-campaña*)

ANUNCIOTELEVISIVO(minutos, *slogan*)

CUÑARADIOFONICA(minutos, *slogan*)

ANUNCIOIMPRESO(ancho, alto, imagenes?, *slogan*)

ANUNCIANTE(*id-anunciante*, nombre)

INSTITUCION(*id-anunciante*)

EMPRESA(*id-anunciante*)

MEDIOAUDIOVISUAL(nombre)

CADENATELEVISIVA(*nombre*)

EMISORADERADIO(*nombre*)

PUBLICACION(nombre, tirada)

PERIODICO(nombre)

REVISTA(nombre, periodicidad)

MEDIOAUDIOVISUAL-EMPRESA(nombre medio audiovisual, id-anunciante)

ANUNCIO-CADENATELEVISIVA(slogan, nombre-cadena, nº veces)

CUÑA-EMISORARADIO(slogan, nombre-emisora, nº veces)

ANUNCIO-PUBLICACION(slogan, nombre-publicacion, nº pagina)

- a) Slogan, tema y minutos de duración de los anuncios de televisión y las cuñas radiofónicas.

```
SELECT anuncio.Tema-campaña, anuncio.Slogan,
anunciotelevisivo.minutos
FROM anuncio, anunciotelevisivo
WHERE anuncio.Slogan = anunciotelevisivo.Slogan
UNION
SELECT anuncio.Tema-campaña, anuncio.Slogan,
cuñaradiofonica.minutos
FROM anuncio, cuñaradiofonica
WHERE anuncio.Slogan = cuñaradiofonica.Slogan
```

- b) Slogan y formato de los anuncios sin imágenes publicados en "La Rioja" o "El Correo"

```
SELECT anuncioimpreso.Slogan, Ancho, Alto
FROM anuncioimpreso, anuncio-publicacion
WHERE imágenes? = No
AND nombre-publicación = "La Rioja"
OR nombre-publicación = "El Correo"
AND anuncioimpreso.Slogan = anuncio-publicacion.Slogan
```

- c) Nombre de las emisoras de radio que no se anuncien.

```
SELECT Nombre
FROM emisoraderadio
WHERE Nombre NOT IN
(SELECT DISTINCT emisoraderadio.Nombre
FROM emisoraderadio, medioaudiovisual, medioaudiovisual-
empresa, empresa, anunciante, campañapublicitaria
WHERE emisoraderadio.Nombre = medioaudiovisual.Nombre
AND medioaudiovisual.Nombre = medioaudiovisual-empresa.Nombre-
medioaudiovisual
AND medioaudiovisual-empresa.Id-anunciante = empresa.Id-
anunciante
AND empresa.Id-anunciante = anunciante.Id-anunciante
AND anunciante.Id-anunciante = campañapublicitaria.Id-
anunciante)
```

- d) Nombre de anunciantes con anuncios de televisión de más de 2 minutos de duración.

```
SELECT anunciante.Nombre
FROM anunciante, campañapublicitaria, anuncio, anunciotelevisivo
WHERE minutos > 2
AND anunciante.Id-anunciante = campañapublicitaria.Id-anunciante
AND anuncio.Temacampaña = campañapublicitaria.Temacampaña
AND anuncio.Slogan = anunciotelevisivo.Slogan
```

- e) Dar una consulta que permita verificar si todo anunciante es o bien institución o bien empresa.

```
SELECT Id-anunciante
FROM anunciante
WHERE Id-anunciante NOT IN (SELECT Id-anunciante
                             FROM institucion
                             UNION
                             SELECT Id-anunciante
                             FROM empresa)
```

- f) Mayores tiradas de revistas, especificando su periodicidad.

```
SELECT MAX(tirada) AS tiradaMaximaRevista, periodicidad
FROM publicacion, revista
WHERE publicacion.Nombre = revista.Nombre
GROUP BY periodicidad
```

- g) Slogan y minutos de duración de los anuncios de televisión que hayan sido emitidos el mismo número de veces por todas las cadenas de televisión.

```
SELECT anunciotelevisivo.Slogan, anunciotelevisivo.Minutos
FROM anunciotelevisivo, anuncio-cadenatelevisiva
WHERE anunciotelevisivo.Slogan, anuncio-cadenatelevisiva.Slogan
GROUP BY anuncio-cadenatelevisiva.Slogan, anuncio-
cadenatelevisiva.Nºveces
HAVING COUNT(anuncio-cadenatelevisiva.Nombre-cadena)=
        (SELECT COUNT(nombre)
         FROM cadenatelevisiva)
```

Ejercicio 12: Aeropuerto

Consideremos la siguiente estructura de tablas:

PASAJERO(DNI, Nombre, Apellidos, Edad, *Id-vuelo*, N° asiento)

VUELO(TipoAvión, Compañía, *Id-vuelo*)

VUELO-DE-SALIDA(*HoraSalida*, *PuertaEmbarque*, *Id-vuelo*, *Ciudad*, *Aeropuerto*)

VUELO-DE-LLEGADA(*HoraLlegada*, *PuertaSalida*, *Id-vuelo*, *Ciudad*, *Aeropuerto*)

LUGAR(*Ciudad*, Pais, *Aeropuerto*)

- a) Hora de salida y puerta de embarque del pasajero Andrés Pérez García.

```
SELECT HoraSalida, PuertaEmbarque
FROM pasajero, vuelo, vuelo-de-salida
WHERE nombre = "Andrés"
AND apellidos = "Pérez López"
AND pasajero.Id.vuelo = vuelo.Id-vuelo
AND vuelo.Id-vuelo = vuelo-de-salida.Id-vuelo
```

- b) Número de vuelos realizados por la compañía Iberia.

```
SELECT compañía, COUNT(Id-vuelo) AS NumeroVuelos
FROM vuelo
WHERE compañía = "Iberia"
GROUP BY compañía
```

- c) Relación de compañías de aviones que utilizan el aeropuerto junto con el número de vuelos de salida y el número de vuelos de llegada que han realizado.

```
SELECT compañía, COUNT(vuelo-de-salida.Id-vuelo)
FROM vuelo, vuelo-de-salida
WHERE vuelo.Id-vuelo = vuelo-de-salida.Id-vuelo
GROUP BY compañía
UNION
SELECT compañía, COUNT(vuelo-de-llegada.Id-vuelo)
FROM vuelo, vuelo-de-llegada
WHERE vuelo.Id-vuelo = vuelo-de-llegada.Id-vuelo
GROUP BY compañía
```

- d) Nombre y apellidos de los pasajeros que viajan a París ordenados alfabéticamente.

```
SELECT nombre, apellidos
FROM pasajero, vuelo, vuelo-de-llegada
WHERE Ciudad = "París"
AND pasajero.Id.vuelo = vuelo.Id-vuelo
AND vuelo.Id-vuelo = vuelo-de-llegada.Id-vuelo
ORDER BY apellidos, nombre
```

- e) Compañías que hayan realizado vuelos con mas de 100 pasajeros.

```
SELECT DISTINCT Compañia
FROM pasajero, vuelo
WHERE pasajero.Id.vuelo = vuelo.Id-vuelo
GROUP BY Id-vuelo, Compañia
HAVING COUNT(DNI) > 100
```

Ejercicio 13: Ciclismo

Consideremos la siguiente estructura de tablas:

CICLISTA (IDCiclista, Nombre, Edad, Altura, Peso, Pulsaciones, *NombreSponsor*)

EQUIPO (*NombreSponsor*)

SPONSOR (*NombreSponsor*)

PRUEBA (NombrePrueba, Año1Edicion)

CLASICA (NombrePrueba, KmsRecorrido)

ETAPAS (NombrePrueba)

ORGANIZADOR (NombreOrganizador)

CICLISTA-PRUEBA (IDCiclista, NombrePrueba, N°Veces)

EQUIPO-ETAPA (NombreSponsor, NombrePrueba, N°Veces)

PRUEBA-ORGANIZADOR (NombrePrueba, NombreOrganizador)

SPONSOR-ORGANIZADOR (NombreSponsor, NombreOrganizador)

Construir las sentencias de consulta SQL siguientes:

- a) Nombre y edad de los ciclistas ganadoresdel Tour de Francia más de una vez.

```
SELECT ciclista.Nombre, ciclista.Edad
FROM ciclista, ciclista-prueba
WHERE ciclista.IDCiclista = ciclista-prueba.IDCiclista
AND ciclista-prueba.NombrePrueba = "Tour de Francia"
AND ciclista-prueba.N°Veces > 1
```

b) Nombre de los ciclistas del grupo Kelme que han ganado alguna clásica

```
SELECT ciclista.Nombre
FROM ciclista, ciclista-prueba, clasica
WHERE ciclista-prueba.IDCiclista = ciclista.IDCiclista
AND ciclista.NombreSponsor = "Kelme"
AND ciclista-prueba.NombrePrueba = clasica.NombrePrueba
```

c) Nombre de los ciclistas que han ganado el Tour de Francia y Vuelta a España.

```
SELECT ciclista.Nombre
FROM ciclista
WHERE ciclista.IDCiclista IN (SELECT ciclista-prueba.IDCiclista
                              FROM ciclista-prueba
                              WHERE ciclista-prueba.NombrePrueba
                                = "Tour de Francia")
AND ciclista.IDCiclista IN (SELECT ciclista-prueba.IDCiclista
                              FROM ciclista-prueba
                              WHERE ciclista-prueba.NombrePrueba
                                = "Vuelta a España")
```

d) Nombre de los sponsor de los equipos que han ganado (clasificación general por equipos) alguna prueba por etapas organizadas por dichos sponsor.

Nótese que al construir la sentencia, se asume que un equipo puede ganar tanto pruebas clásicas como pruebas por etapas. Del modelo relacional no se concluye por los nombres de campos utilizados lo que del diagrama E-R, donde se indica que un equipo solo puede vencer en pruebas por etapas. Si se asume esta misma consideración no sería necesaria ni la tabla ETAPAS ni la primera condición de la cláusula WHERE. Esta condición se incluye únicamente para asegurar que la prueba vencida es una prueba por etapas.

```
SELECT DISTINCT equipo-etapas.NombreSponsor
FROM equipo-etapas,
     etapas,
     prueba-organizador,
     sponsor-organizador
WHERE equipo-etapas.NombrePrueba = etapas.NombrePrueba
AND equipo-etapas.NombreSponsor =
  sponsor-organizador.NombreSponsor
AND equipo-etapas.NombrePrueba = prueba-organizador.NombrePrueba
AND prueba-organizador.NombreOrganizador =
  sponsor-organizador.NombreOrganizador
```

Ejercicio 14: Cine

Consideremos la siguiente estructura de tablas:

PELICULA (IDPelicula, Titulo, Año, Nacionalidad, OscarPelicula?, NombreDirector, OscarDirector?, *TituloGuion*, OscarGuion?)

GUION (Titulo)

PERSONA (Nombre, Edad, Nacionalidad)

GUIONISTA (*Nombre*)

DIRECTOR (*Nombre*)

INTERPRETE (*Nombre*)

REMAKE (*IDPeliculaOriginal*, *IDPeliculaRemake*)

PELICULA-INTERPRETE (*IDPelicula*, *NombreInterprete*, Oscar?)

ESCRITOR-GUION (*NombreGuionista*, *TituloGuion*)

Construir las sentencias de consulta SQL siguientes:

a) **Id y Título de las películas interpretadas por Ingrid Bergman.**

```
SELECT película.IDPelícula, película.Título
FROM película, película-interprete
WHERE película-interprete.IDPelícula = película.IDPelícula
AND película-interprete.NombreInterprete = "Ingrid Bergman"
```

b) **Título y Nombre del Director, de las películas en cuyo reparto se incluye a éste último**

```
SELECT película.Título, película.NombreDirector
FROM película, película-interprete
WHERE película.IDPelícula = películaInterprete.IDPelícula
AND película.NombreDirector =
    película-interprete.NombreInterprete
```

c) **Título de las películas que ganaron el Oscar a la mejor película y al mejor guion.**

```
SELECT Título
FROM película
WHERE OscarPelícula? = Sí
AND OscarGuion? = Sí
```

d) **Título de las películas interpretadas por Spencer Tracy y Catherine Hepburn.**

```
SELECT Título
FROM película
WHERE IDPelícula IN (SELECT IDPelícula
                     FROM película-interprete
                     WHERE NombreInterprete = "Spencer Tracy")
AND IDPelícula IN (SELECT IDPelícula
                   FROM película-interprete
                   WHERE NombreInterprete = "Catherine Hepburn")
```

e) **Título de las películas y sus remakes que hayan sido dirigidas por el mismo director. Incluye el nombre del director**

```
SELECT películaOriginal.Título,
       películaRemake.Título,
       películaOriginal.NombreDirector
FROM película AS películaOriginal,
     película AS películaRemake,
     remake
WHERE remake.IDPelículaOriginal = películaOriginal.IDPelícula
AND remake.IDPelículaRemake = películaRemake.IDPelícula
AND películaRemake.NombreDirector =
    películaOriginal.NombreDirector
```

Ejercicio 15: Grupos Musicales

Consideremos la siguiente estructura de tablas:

MUSICO (Nombre, Sexo, Edad, *NombreGrupo*, AñoIncorporacion)

LETRISTA (*Nombre*)

INSTRUMENTISTA (*Nombre*, Instrumento)

GRUPO (Nombre, AñoFundacion)

CANCIÓN (IDCancion, Título, Duración, *NombreCompañía*)

COMPAÑÍA (NombreCompañía, *NombreGrupoEstrella*)

GRUPO-CANCION (*NombreGrupo*, IDCancion, Superventas?)

LETRISTA-CANCION (*NombreLetrista*, *IDCancion*)

Construir las sentencias de consulta SQL siguientes:

- a) Relación de los nombres de los componentes de los grupos musicales surgidos en los años 80.

```
SELECT musico.Nombre
FROM musico, grupo
WHERE grupo.Nombre = musico.NombreGrupo
AND grupo.AñoFundacion BETWEEN 80 AND 89
```

- b) Nombre del bajo del grupo estrella de la EMI

```
SELECT musico.Nombre
FROM musico, instrumentista, compañía
WHERE instrumentista.Instrumento = "Bajo"
AND instrumentista.Nombre = musico.Nombre
AND musico.NombreGrupo = compañía.NombreGrupoEstrella
AND compañía.NombreCompañía = "EMI"
```

- c) Nombre de la compañía con los derechos de las canciones que han sido superventas, tocadas por el grupo Oasis.

```
SELECT canción.NombreCompañía
FROM canción, grupo-cancion
WHERE canción.IDCancion = grupo-cancion.IDCancion
AND grupo-cancion.Superventas? = Sí
AND grupo-cancion.NombreGrupo = "Oasis"
```

- d) Nombre de los músicos fundadores de los Rolling Stones.

```
SELECT musico.Nombre
FROM musico, grupo
WHERE musico.NombreGrupo = grupo.Nombre
AND musico.AñoIncorporacion = grupo.AñoFundacion
AND grupo.Nombre = "Rolling Stones"
```

- e) Título de las canciones superventas de The Beatles escritas por George Harrison

```
SELECT canción.Titulo
FROM cancion, grupo-cancion, letrista-cancion
WHERE cancion.IDCancion = grupo-cancion.IDCancion
AND canción.IDCancion = letrista-cancion.IDCancion
AND grupo-cancion.Superventas? = Sí
AND grupo-cancion.NombreGrupo = "The Beatles"
AND letrista-cancion.NombreLetrista = "George Harrison"
```

- f) Título y grupo de las canciones superventas escritas por algún componente de dicho grupo (The Beatles)

```
SELECT canción.Titulo, grupo-cancion.NombreGrupo
FROM canción, grupo-cancion, letrista-cancion
WHERE grupo-cancion.Superventas? = Sí
AND cancion.IDCancion = grupo-cancion.IDCancion
AND canción.IDCancion = letrista-cancion.IDCancion
AND letrista-cancion.NombreLetrista IN (SELECT Nombre
FROM musico
WHERE NombreGrupo = "The Beatles")
```

- g) Identificador de las canciones tocadas por los Beatles y los Rolling Stones

```
SELECT IDCancion
```

```
FROM grupo-cancion  
WHERE NombreGrupo IN ("The Beatles", "Rolling Stones")
```

Índice

CONVENCIONES	1
EJERCICIOS CONSULTAS SQL	2
Ejercicio 1: Seguros	2
Ejercicio 2: Empresa2.....	3
Ejercicio 3: Ríos de España	3
Ejercicio 4: Vehículos	5
Ejercicio 5: Autobuses	5
Ejercicio 6: Seguros2.....	6
Ejercicio 7: Horarios	7
Ejercicio 8: Prensa.....	8
Ejercicio 9: Vinos	9
Ejercicio 10: Museos	10
Ejercicio 11: Publicidad.....	11
Ejercicio 12: Aeropuerto.....	13
Ejercicio 13: Ciclismo	14
Ejercicio 14: Cine.....	15
Ejercicio 15: Grupos Musicales	16