

Caso práctico: Almacén de datos para el análisis del parque de vehículos

PR2 – Sistema de bases de datos relacionales

Esta segunda práctica del curso (**PR2 - Sistema de bases de datos relacionales**) consta de dos partes, una primera parte destinada a la evaluación del contenido teórico y una segunda parte práctica destinada a realizar una serie de ejercicios prácticos.

La parte teórica tiene una ponderación del 20% sobre la nota total de la práctica. Se evalúa mediante un **cuestionario**, disponible en el apartado *Contenidos del aula*. Su objetivo es comprobar la correcta comprensión del segundo módulo de la asignatura. Debe realizarse después de haber estudiado el módulo 2 y de haber leído detenidamente las indicaciones de realización del mismo.

La parte práctica, cuya ponderación es del 80% de la nota total de la práctica, se lleva a cabo en la máquina virtual (VDI) mediante la realización de todos los ejercicios que se detallan en el apartado “Enunciado parte práctica” de este documento.

Se recomienda consultar la documentación disponible en el apartado *Recursos de Aprendizaje PR2 del espacio Contenidos del aula*. Dichos recursos pretenden facilitar la realización de los ejercicios y resolver posibles dudas. También se incluye la **documentación técnica**, imprescindible para trabajar en el entorno virtual (VDI), que está preparado con todas las herramientas necesarias, instaladas y configuradas. Recordad que ante cualquier duda y/o problema técnico relativo al funcionamiento del entorno o del software instalado debéis consultar al profesor del aula 76.596.- *Laboratorio de Soporte a las bases de datos analíticas*. **Únicamente se evaluarán los ejercicios cuya resolución esté acompañada de comentarios; las imágenes o gráficas sin ningún tipo de explicación no serán evaluadas.**

Parte práctica

En esta segunda práctica (PR2) y dentro del contexto del caso práctico “Almacén de datos para el análisis del parque de vehículos”, se utilizará una PostgreSQL como sistema de gestión de bases de datos y SQL como lenguaje de programación y acceso a la base de datos que almacena los datos de caso práctico en cuestión.

Contexto

Una vez realizada la PR1 y familiarizados con algunas de las herramientas necesarias para llevar a cabo el diseño y construcción del caso práctico, surge la necesidad de dar cobertura a la gestión de un parque de vehículos utilizando el sistema de bases de datos relacional PostgreSQL.

Un parque de vehículos consiste en el conjunto total de automóviles, motocicletas, camiones, autobuses y otros medios de transporte y maquinaria motorizada que están registrados y en circulación dentro de un territorio. El conocimiento y gestión de estos datos es fundamental para entender el desarrollo del sector transporte, así como su impacto en la economía, el medio ambiente y la planificación urbana.

El análisis del parque de vehículos permite conocer tendencias como el crecimiento del número de vehículos, la antigüedad del parque, el tipo de combustibles utilizados, y la distribución por las diferentes regiones geográficas. Es por ello por lo que se desea disponer de una base de datos al servicio de un software destinado a la gestión del parque de vehículos matriculados en un territorio, así como de sus propietarios y conductores.

En los siguientes ejercicios trabajaremos sobre el diseño de la base de datos y practicaremos el uso del lenguaje de consultas SQL para el análisis de datos.

Ejercicio 1. Diseño conceptual de la base de datos del parque de vehículos (35%)

Objetivo:

Comprender el proceso de diseño de una base de datos y sus etapas. Se trata de entender cómo representar los requisitos de la base de datos mediante un modelo conceptual expresado con diagramas E/R (entidad/relación) que represente la información y sus relaciones, el modelo lógico correspondiente suponiendo tecnología relacional y cómo se realiza la traducción de conceptual a lógico relacional.

Indicaciones para el desarrollo del ejercicio:

En base a la necesidad del desarrollo de un software destinado a la gestión del parque de vehículos perteneciente al sector de la automoción, se requiere diseñar un modelo de datos E/R como soporte a la aplicación de gestión de modelos de vehículos, vehículos y propietarios de los vehículos.

Partiendo de la base de que un vehículo a motor es aquel que se propulsa mediante un motor y tiene la capacidad de circular por las vías públicas (por ejemplo, turismos, motocicletas, camiones, autobuses, etc.), se requiere almacenar la información base de cada **vehículo** como un identificador único, la matrícula, la fecha de matriculación, la marca y el modelo, y si fue matriculado como nuevo o usado. Asimismo, deben registrarse los datos básicos de su **propietario**, incluyendo un identificador único, nombre y apellidos, el identificador del documento nacional de identidad o equivalente (DNI, NIE, pasaporte, etc), fecha de nacimiento, teléfono de contacto, dirección de residencia habitual y correo electrónico.

Debe existir un único propietario para cada vehículo, permitiendo que un propietario permanezca en la base de datos en caso de vender o dar de baja un vehículo, quedando el propietario sin ningún vehículo de propiedad registrado.

Adicionalmente, para cada vehículo se requiere poder conocer cuál es su residencia, siendo el **municipio** el dato base que se necesita almacenar. Dada la cantidad de municipios existentes, se requiere disponer de información geográfica adicional con el objetivo de poder agilizar la gestión del parque de vehículos.

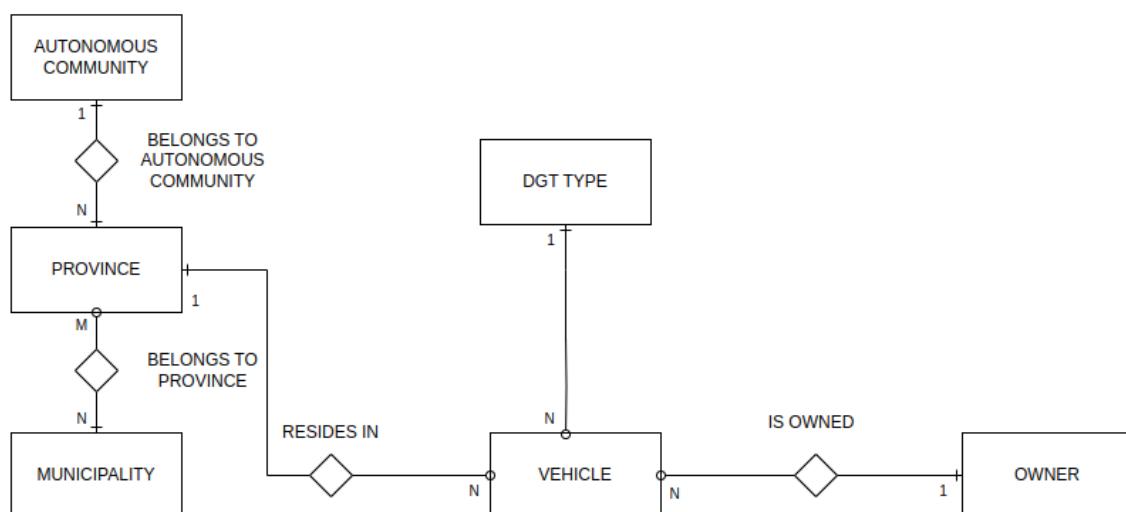
Es por ello que se requiere a modo de jerarquía disponer de la información de las **comunidades autónomas** (identificador y nombre de cada una de ellas), de las **provincias** que las constituyen (identificador y nombre de cada una de ellas) y de los **municipios**, disponiendo para cada uno de estos últimos, su identificador, su nombre, población, superficie, perímetro, sus datos de geolocalización y altitud.

Para facilitar la clasificación de los diferentes tipos de vehículos existentes, se requiere disponer de una categorización de la DGT (Dirección General de Tráfico), estructurada en dos niveles: tipo y subtipo de vehículo. Cada vehículo estará vinculado a un subtipo. Para simplificar el modelo conceptual, esta categorización se almacenará en una única entidad, en la cual coexisten el tipo y el subtipo.

1.1 Revisión del modelo conceptual E/R

La empresa subcontratada para llevar a cabo el diseño de la base de datos que permita incorporar funcionalidades para la nueva necesidad detectada propone el siguiente modelo conceptual.

Modelo conceptual propuesto:



Descripción de atributos de las entidades propuestas:

AUTONOMOUS COMMUNITY

aacc-id, aacc-name

PROVINCE

province-id, province-name, aacc-id

MUNICIPALITY

cod_geo, name_muni, population, surface, perimeter, longitude_etr89,
latitude_etr89, altitude

DGT TYPE

cod_subtype_dgt, des_subtype_dgt, cod_type_dgt, des_type_dgt

OWNER

name, first_name, last_name, birthdate, phone, address, email

VEHICLE

vehicle-id, registration_number, brand, model, registration_date, new_used,
cod_subtype_dgt, province-id

Se pide:

- 1) Realizar una revisión del modelo conceptual completo facilitado con el objetivo de indicar si la representación de información es correcta o no en base a los requerimientos descritos.
- 2) Para cada error u omisión detectados, modificar el diagrama del diseño conceptual para corregirlo.

En esta actividad se evaluará la adecuación del modelo conceptual propuesto con respecto a los requerimientos establecidos en el enunciado. Especialmente, se valorará la justificación de cada modificación, así como la elaboración de un nuevo diagrama que refleje dichas correcciones.

1.2 Evolución del modelo conceptual E/R

Una vez corregidos los errores del modelo conceptual, piden algunas modificaciones para incorporar nuevas funcionalidades a la aplicación.

Se desea ampliar el modelo actual, reflejando situaciones de multipropiedad y posibilitando **registrar uno o más propietarios por vehículo**.

Por otro lado, desea registrar los diferentes **conductores** que, utilizando los vehículos registrados en el parque de vehículos, son multados por las autoridades competentes.

De la **sanción** se desea registrar, el **vehículo**, la **persona que conducía** (en caso de identificarla), el **municipio** donde se ha cometido la infracción y la **fecha** en que tuvo lugar. En caso de no poder identificar al conductor del vehículo, la sanción no podrá reflejar al conductor en el registro.

Respecto a la **persona que conducía** se desea almacenar un identificador único, nombre y apellidos, el identificador del documento nacional de identidad u homólogo (DNI, NIE, pasaporte, etc.), fecha de nacimiento, fecha de obtención del carnet de conducir, fecha de caducidad del mismo, teléfono de contacto, dirección de residencia habitual y correo electrónico.

Por ello se solicita adaptar el modelo conceptual de datos para poder llevar a cabo esta nueva funcionalidad. En concreto, se solicita la modificación del diseño conceptual, justificando cada cambio a realizar, adaptando el diagrama del diseño conceptual.

En esta actividad se evaluará cada una de las adaptaciones del modelo conceptual, poniendo especial énfasis en la justificación de los cambios realizados y en su representación gráfica.

Ejercicio 2. Diseño lógico de la base de datos del parque de vehículos (15%)

Objetivo:

Comprender el proceso de diseño de bases de datos y en concreto, como se representa el modelo lógico-relacional a partir de un modelo conceptual dado.

Indicaciones para el desarrollo del ejercicio:

Este es el modelo lógico correspondiente al modelo conceptual del apartado **2.2.**

Ejercicio 1, que se muestra a modo de ejemplo.

AUTONOMOUS COMMUNITY (aacc-id, aacc-name)

PROVINCE (province-id, province-name, aacc-id)

MUNICIPALITY (cod_geo, name_muni, population, surface, perimeter, longitude, latitude, altitude)

DGT TYPE (cod_subtype_dgt, des_subtype_dgt, cod_type_dgt, des_type_dgt)

OWNER (name, first_name, last_name, birthdate, phone, address, email)

VEHICLE (vehicle-id, registration_number, brand, model, registration_date, new_used, cod_subtype_dgt, province-id)

A partir de las correcciones indicadas en el apartado **2.2. Ejercicio 1** y los cambios solicitados en el apartado **2.3. Ejercicio 2**, se pide:

- 1) Identificar las modificaciones a realizar en el modelo lógico propuesto para incorporar dichas correcciones y cambios.
- 2) Expresar el modelo lógico final utilizando la notación empleada en los módulos didácticos, indicando claves primarias, alternativas, si existen, y claves foráneas.

En esta actividad se evaluará la corrección del modelo lógico propuesto en base a los requerimientos fijados en el enunciado, así como en la presentación del modelo lógico final evolucionado. Especialmente, se valorará la representación del modelo lógico final **utilizando la notación** empleada en los módulos didácticos.

Ejercicio 3. Modelo físico de la base de datos para la gestión del parque de vehículos (30%)

Objetivo:

Conocer los conceptos básicos del lenguaje estándar SQL para sus diferentes usos:

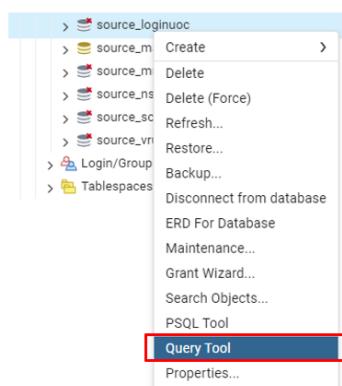
- al diseñar y construir una base de datos relacional, introducir, borrar y modificar datos
- realizar cualquier tipo de consulta a la base de datos

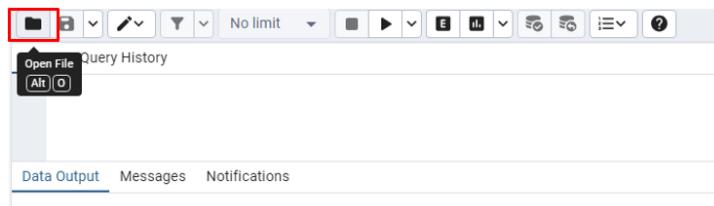
Indicaciones para el desarrollo del ejercicio: 3. 1 Creación de sentencias SQL

La primera tarea a realizar es llevar a cabo la creación de las tablas y carga de los datos en la base de datos *PostgreSQL*, necesarios que constituirán una pequeña parte del modelo del software de la aplicación de gestión del parque automovilístico. Para ello, accediendo primero la herramienta “Query Tool”, se debe abrir los ficheros siguientes ficheros facilitados junto al enunciado,

- dgt_type.sql
- municipalities.sql
- vehicles.sql

mediante el siguiente cuadro de navegación:





Y una vez cargado en el editor se debe ejecutar los scripts de carga. Para comprobar que se han creado e insertado correctamente los datos, consultaremos las tablas creadas en el esquema *dbo*. La carga del script así como la verificación de la creación de las tablas debe incorporarse como evidencias en la solución de la práctica.

The screenshot shows the Object Explorer pane of SSMS. The tree view displays the following structure:

- > Publications
- ▽ Schemas (2)
 - ▽ dbo
 - > Aggregates
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Operators
 - > Procedures
 - > Sequences
 - ▽ Tables (3)
 - > dgt_type
 - > municipalities
 - > vehicles

y ejecutaremos diferentes consultas SQL para verificar los datos cargados:

- Consultar el número de registros de la tabla *dgt_type*

```
SELECT
    COUNT(1) as total_subtypes
FROM
    dbo.dgt_type
```

```
1  SELECT
2      count(1)  as  total_subtypes
3  FROM
4      dbo.dgt_type
```

Data Output Messages Notifications

	total_subtypes
1	112

Mediante la función *COUNT()* obtendremos los registros almacenados en la tabla *dgt_type* que reside dentro del esquema *dbo*.

- Consultar el número de registros de la tabla *dbo.municipalities*

```
SELECT
    COUNT(1) as total_municipalities
FROM
    dbo.municipalities
```

```
1 SELECT
2     count(1) as total_municipalities
3 FROM
4     dbo.municipalities
```

Data Output Messages Notifications

The screenshot shows the results of a SQL query in a table format. The table has two columns: 'total_municipalities' and its data type 'bigint'. There is one row with the value '8132'. Below the table are standard SQL Server Management Studio toolbar icons.

	total_municipalities
1	8132

Mediante la función `COUNT()` obtendremos los registros almacenados en la tabla *municipalities* que reside dentro del esquema *dbo*.

- Consultar el número de registros de la tabla *dbo.vehicles*

```
SELECT
    COUNT(1) as total_vehicles
FROM
    dbo.vehicles
```

```
1 SELECT
2     count(1) as total_vehicles
3 FROM
4     dbo.vehicles
```

Data Output Messages Notifications

The screenshot shows the results of a SQL query in a table format. The table has two columns: 'total_vehicles' and its data type 'bigint'. There is one row with the value '1721704'. Below the table are standard SQL Server Management Studio toolbar icons.

	total_vehicles
1	1721704

Mediante la función `COUNT()` obtendremos los registros almacenados en la tabla *vehicles* que reside dentro del esquema *dbo*.

3.2 Ejercicios a realizar:

- 1) ¿Qué subtipos de vehículos DGT existen? Mostrar la lista de subtipos con su código y descripción, y ordenados por ambos campos.
- 2) Obtener el total de municipios cuya provincia sea “Sevilla”.
- 3) Obtener el listado de municipios (listar ordenados descendente por nombre de municipio) cuya provincia sea “Sevilla”, y cuyo nombre de municipio empiece por “A” o por “U”.
- 4) Insertar en la tabla dgt_type un nuevo registro con los valores en los siguientes campos:
 - cod_type_dgt: valor “99”
 - des_type_dgt: valor “UOC”
 - cod_subtype_dgt: valor “00”
 - des_subtype_dgt: vuestro ‘loginuoc’ asignado en el campus

Una vez insertado el registro, visualizarlo mediante una consulta filtrando por cod_type_dgt = “99” y cod_subtype_dgt = “00”, y ordenando ascendente por los códigos y subcódigos de los tipos DGT.

- 5) Actualizar el registro insertado en el punto anterior, para que tenga el campo des_subtype_dgt = “campusuoc”, y realizar una consulta para visualizar el cambio.
- 6) Eliminar el nuevo registro insertado en el punto 4) y actualizado en el punto 5).
- 7) Mostrar los datos de los vehículos categorizados como “TURISMOS” y el nombre del municipio de residencia, filtrados por los municipios de residencia con una altura superior a 950 metros y ordenados por la marca del vehículo.
- 8) Mostrar el número de vehículos agrupados y ordenados por el nombre del municipio de residencia.
- 9) Incorporar a la consulta anterior un filtro para que únicamente se muestren los municipios con menos de 15.000 vehículos.
- 10) Mostrar el número de vehículos agrupados y ordenados por el campo des_type_dgt y por si el vehículo está registrado como usado o no.

En esta actividad se evaluará la correcta creación de las consultas SQL que den respuesta a los requerimientos planteados, así como la presentación ordenada y coherente de los resultados. Además, especialmente se valorará la correcta aplicación de filtros adecuados y el uso de buenas prácticas en la creación de consultas, como es el uso de alias claros, sentencias con cláusulas ordenadas y criterios de eficiencia.

3.3 Corrección de sentencias SQL

Verificar que las siguientes consultas SQL están construidas y retornan los valores según los requerimientos definidos.

- 1) Se desea conocer la información de todos los vehículos cuya marca empiece por “Y” y cuya provincia de residencia sea “Barcelona”, ordenados por la marca del vehículo.

```
SELECT veh.*  
FROM  
    dbo.vehicles veh,  
    dbo.municipalities mun  
WHERE  
    veh.brand like 'Y%'  
    and mun.province = 'Barcelona'  
ORDER BY  
    veh.brand;
```

- 2) Se desea conocer el nombre de todos los municipios, ordenados por nombre, cuyo nombre termine en la letra ‘a’, y cuya población sea menor de 5 habitantes o mayor a un millón de habitantes.

```
SELECT mun.name_muni, population_muni  
FROM  
    dbo.municipalities mun  
WHERE  
    mun.name_muni like '%a'  
    and mun.population_muni > 1000000  
    or mun.population_muni < 5  
ORDER BY  
    mun.name_muni;
```

En esta actividad se evaluará la adecuada respuesta para cada consulta SQL propuesta, así como el razonamiento empleado para determinar la corrección de cada consulta.

Criterios de evaluación

Parte teórica (20%)

- Se detallan al inicio del cuestionario, accesible desde el apartado *Contenidos del aula, Entrega cuestionario PR2*.

Parte práctica (80%)

- Se valorará mediante un documento en formato Word o PDF que muestre evidencias de cumplimiento de los pasos indicados y aportados por el estudiante al resolver los ejercicios solicitados.
- En el aula está disponible la rúbrica de corrección.

Nota final:

La nota final se calculará de acuerdo con esta fórmula:

$$NF = \text{NotaCuestionario (20\%)} + \text{NotaEjerciciosPrácticos (80\%)}$$

dónde:

$$\text{NotaEjerciciosPrácticos} = \text{Ejercicio 1 (35\%)} + \text{Ejercicio 2 (15\%)} + \text{Ejercicio 3 (30\%)}$$

Formato y fecha de entrega

La solución de la PR2 se entregará, a través del espacio “**Contenidos del aula**”, en dos partes:

1. **Entrega Cuestionario PR2**
2. **Entrega práctica PR2.** Envío del documento de solución, en formato Word o PDF, a través del espacio “**Contenidos**” del aula. El nombre del fichero a entregar debe seguir el patrón:
BDA_PR2_Apellido1_Aellido2_Nombre.extensión.

IMPORTANTE: Tal y como indica el plan docente:

- Es responsabilidad única del estudiante asegurarse que envía el cuestionario y entrega el documento que pretende en el lugar del aula habilitado para ese cometido, de lo contrario no se evaluará el ejercicio.
- Entregas realizadas fuera de los canales indicados se considerarán como No presentadas.
- Entregas pasadas las 23:59 h de la fecha límite no serán aceptadas y por tanto, no podrán ser evaluadas.

La fecha máxima de entrega es el 19/11/2025 a las 23:59 horas.