

Proyecto Base de Datos Academia Atentos



Andrea Vieira Hernández
DAW1 - Mañana
IES Alixar
Curso 2021-2022

Índice de contenidos

Introducción del proyecto	3
Presentación de la situación	4
Diagrama Entidad-Relación	5
Diagrama Relacional	6
Creación de la base de datos	7
Carga masiva de datos	9
Consultas	12
Consulta I	12
Consulta II	13
Consulta III	14
Consulta IV	15
Consulta V	16
Consulta VI	17
Vistas	18
Vista Jefe	18
Vista Poblaciones frecuentes	19
Vista Profesor más antiguo	20

Introducción del proyecto

En el proyecto que se presenta a continuación, queremos representar una situación lo más cercana posible a la realidad, relacionada con el análisis de datos que se desea almacenar de una empresa para su posterior incorporación a una interfaz gráfica con funcionalidades, que permita interactuar con el sistema y tener acceso a dichos datos.

En este caso particular, se analizan las necesidades de una academia de apoyo escolar, con unas necesidades particulares.

Para la elaboración de este proyecto utilizaremos diversas herramientas según la fase en la que nos encontremos. En primer lugar, para el diseño entidad-relación, utilizaremos la aplicación web de Diagrams.net, pues nos permite crear de manera muy sencilla y gráfica el dibujo de las relaciones.

Por otro lado, para pasar este formato de dibujo al esquema de tablas, es decir, al modelo relacional, utilizaremos la herramienta Workbench, por su capacidad para hacer el diseño relacional con los distintos tipos de relaciones, y que se generen de manera automática el traslado de las claves foráneas y la creación de las tablas que surgen por las relaciones N:M y la definición de los tipos de campos; además, nos permite generar archivos csv y scripts.

Finalmente, para la carga de datos y las consultas utilizaremos DBeaver, un gestor gráfico de base de datos. Cabe resaltar que se trata de un proyecto evolutivo, es decir, que va a evolucionar a medida que avancemos en el proceso de creación. Partiremos de los bocetos creados hasta que, en el punto final, la creación de consultas y ver cómo se relacionan las tablas, hasta dar por cerrada toda la macroestructura de dicho proyecto.

Presentación de la situación

La Academia desea tener una base de datos donde almacenar los datos relacionados con sus alumnos, sus empleados, los padres de los alumnos, la facturación y los horarios. De todas las personas que están en el centro se desea guardar el nombre y los apellidos, el DNI, el número de teléfono, el correo electrónico y el identificador que asignará el centro. Todas las personas tienen también una dirección de la cual se quiere destacar la población; de cada población se desea conocer el código postal, el nombre y la provincia.

Cada alumno, del que se debe indicar el colegio del que viene, su curso, si es repetidor y la cantidad de asignaturas suspensas, se matricula, en una fecha para un inicio específico, a una de las modalidades de las que dispone la academia y paga los recibos mensualmente. De las modalidades se almacena el tipo de modalidad (general, específica o idiomas), el tipo de asistencia (presencial, online o mixta) y la tarifa que se le aplica. De los recibos se desea saber la cantidad, la fecha en la que se hace, el modo (efectivo, tarjeta o transferencia) y una descripción. Cada alumno tiene un tutor con el que mantiene contacto la academia, de esos tutores se necesita saber si es la madre, el padre u otro familiar, y la forma de contacto que prefieren. Los alumnos se ubican en aulas, que tiene una capacidad determinada y dispone de unos medios, en día y a una hora en concreto, en ese aula tienen a un profesor que es quien imparte las clases.

La academia tiene distintas clases de empleados (profesores, auxiliares, administrativos y coordinadores), los coordinadores pueden realizar cualquiera de las funciones en la empresa, y el resto tiene unas tareas específicas. De todos estos empleados se desea guardar el puesto que tienen, la titulación, la fecha de alta y la fecha de baja, que aparecerá solo si no continúan en la empresa. Cabe destacar, por último, que todos los profesores, auxiliares y administrativos tienen un coordinador.

Estos serían los requisitos iniciales para la elaboración de los bocetos, teniendo en cuenta que luego pueden surgir modificaciones.

Diagrama Entidad-Relación

Con el modelo que se presenta a continuación se pretende ilustrar la organización de las entidades y qué relación existe entre ellas. Destacamos la entidad ALUMNOS, pues es sobre la que caerá la mayor parte de la información fundamental que pueda necesitar la empresa.

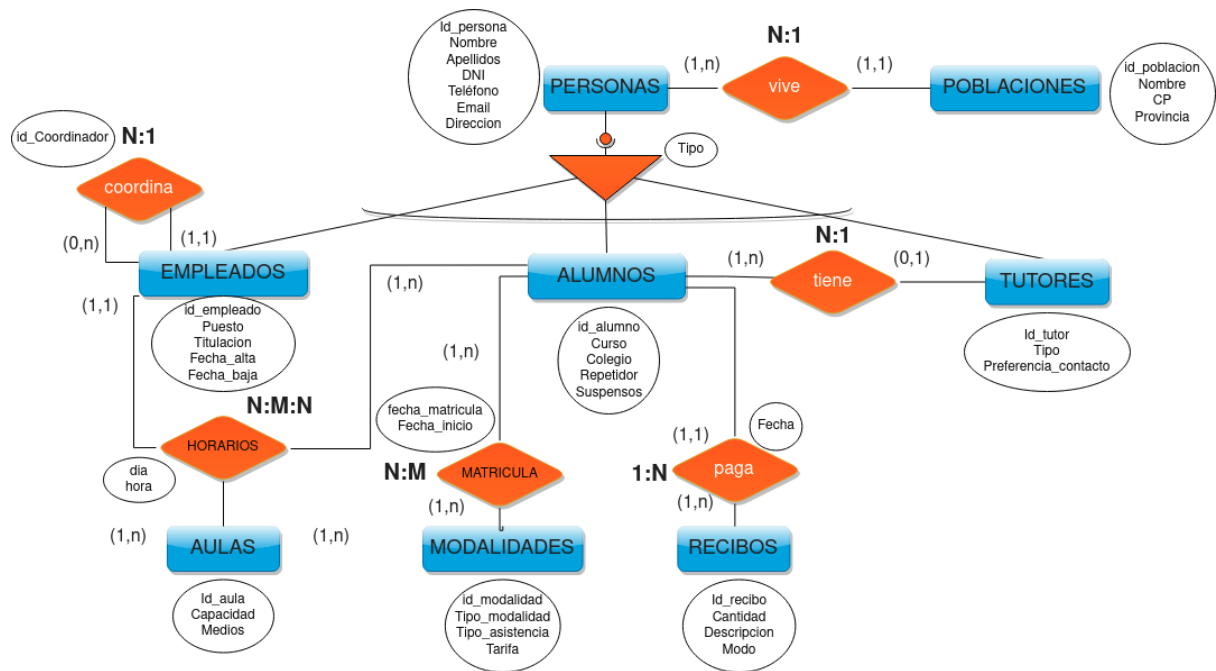
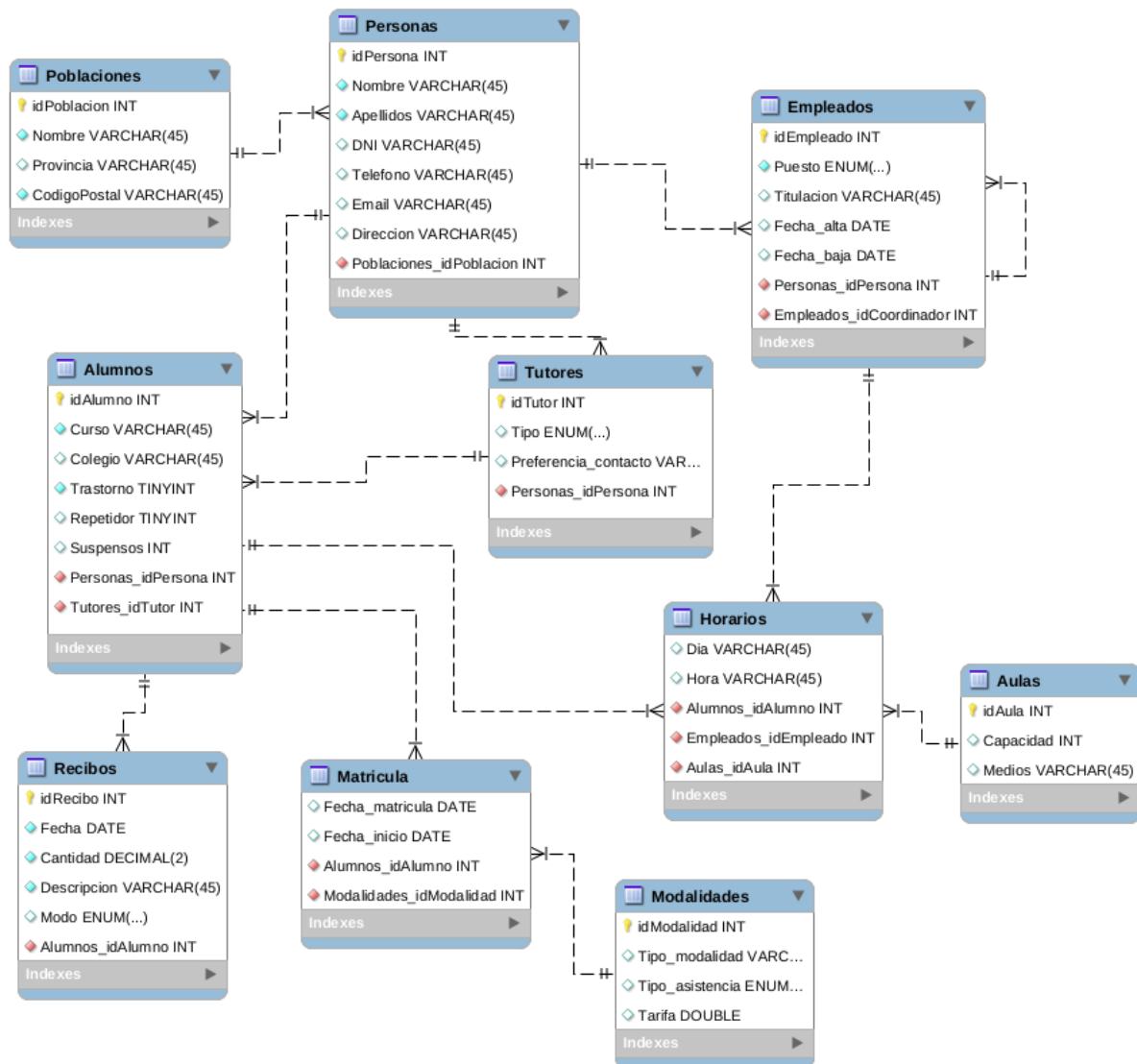


Diagrama Relacional

En este modelo, elaborado con Workbench, podemos ver en disposición tabla los elementos que compondrán la base de datos, junto con sus campos y el tipo de valor que se registra en cada uno de ellos. Gracias a este esquema podemos observar el punto de conexión entre las tablas, fundamental para realizar consultas.



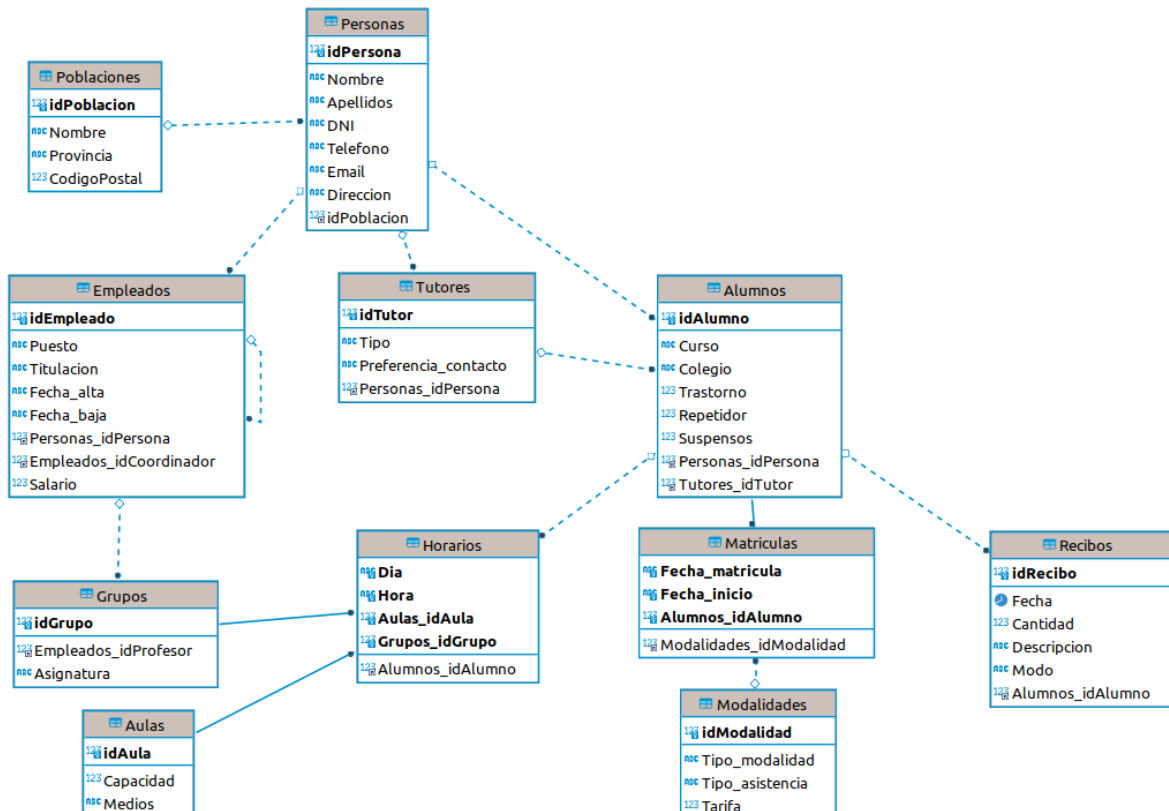
Creación de la base de datos

Dado el esquema que teníamos previamente planteado, pasamos a poner en marcha realmente el proyecto y, en este paso, se analiza nuevamente cada una de las tablas necesarias para la recopilación de datos y la interrelación entre ellos. En esta fase, hemos añadido una tabla Grupo, pues nos permite relacionar de una forma más óptima la distribución de Alumnos, Profesores y Horarios, de manera que se pueda extraer una organización diaria de los grupos para el centro.

En este punto, hemos utilizado la herramienta DBeaver para la generación de las tablas que vemos a continuación, las cuales se crean automáticamente al subir los datos, proceso que se explica en el siguiente apartado. Para hacer las modificaciones de datos y crear las claves primarias y foráneas, hemos alterado las tablas y hemos añadido las restricciones necesarias. Ejemplo:

```
ALTER TABLE `academia-atentos-db`.Horarios ADD CONSTRAINT Horarios_PK PRIMARY KEY (Dia,Hora,Aulas_idAula,Grupos_idGrupo);
```

```
ALTER TABLE `academia-atentos-db`.Matriculas ADD CONSTRAINT Matriculas_PK PRIMARY KEY (Fecha_matricula,Fecha_inicio,Alumnos_idAlumno);
```





academia-atentos-db	
Tables	
> Alumnos	80K
> Aulas	64K
> Empleados	80K
> Grupos	64K
> Horarios	224K
> Matriculas	288K
> Modalidades	128K
> Personas	160K
> Poblaciones	64K
> Recibos	304K
> Tutores	32K

En la imagen anterior, se muestran todas las tablas creadas, y en el diagrama se puede observar en negritas las claves primarias que hemos asignados y con una flechita y las líneas discontinuas, el punto de relación. En el gráfico se puede apreciar que la tabla más importante es la de Alumnos, pues es el foco de las relaciones.

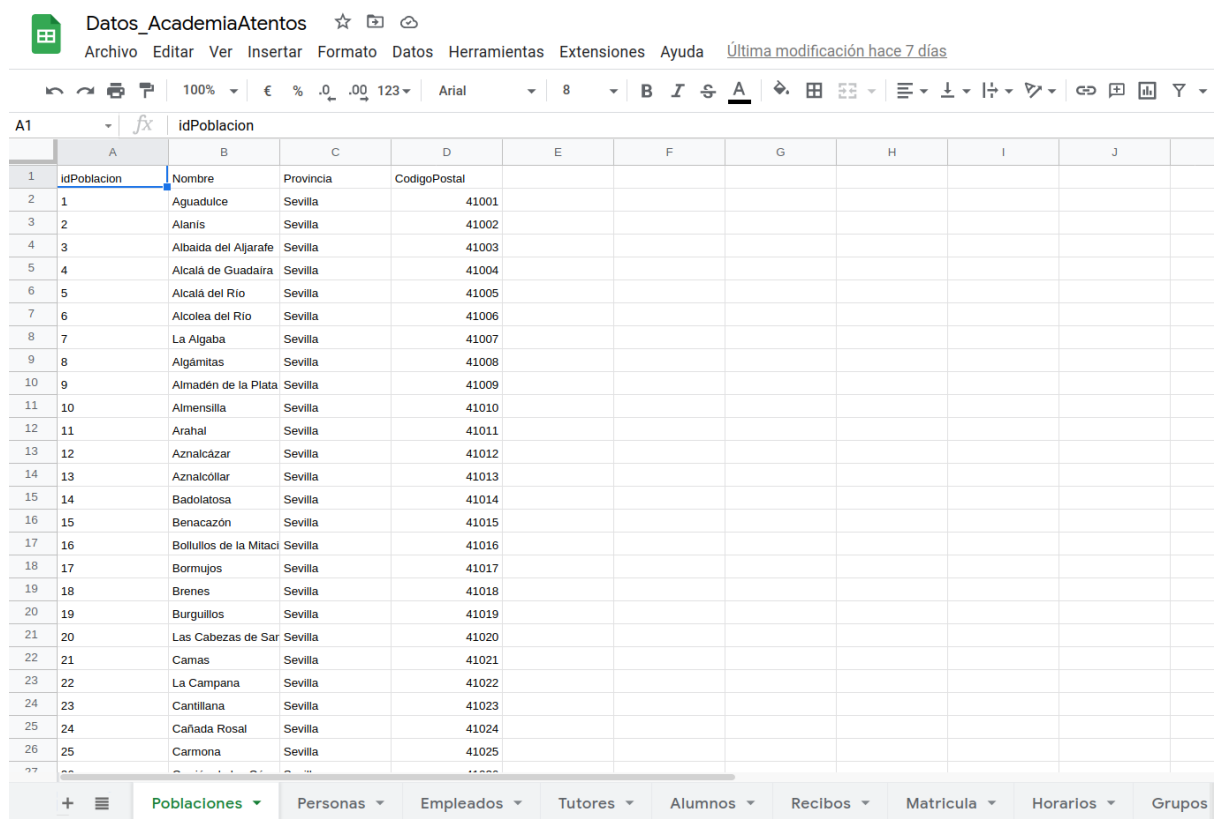
Carga masiva de datos

Como el proceso lo hemos hecho cargando las tablas directamente a partir de los archivos csv, es importante explicar cómo se ha realizado la carga de datos masiva.

Para la generación de los datos nos hemos servido principalmente de dos herramientas: un generador automático de datos, Mockaroo (<https://www.mockaroo.com/>) y las hojas de cálculo de Google Drive.

La primera herramienta, Mockaroo, consiste en una plataforma que nos permite configurar el tipo de datos que podemos generar, siendo bastante configurable con el uso de expresiones regulares en los campos introducidos. Cuando hemos seleccionado todos los campos que deseamos y el tipo de dato que contendrán, descargamos el documento en formato csv con los 1000 registros que genera de forma automática en su versión gratuita.

Como algunos de los datos necesitan unas características más específicas, y los campos de las claves foráneas tenían que corresponder con el elemento al que se refieren, hemos tenido que hacer un proceso más "manual" con el uso de las hojas de cálculo, pero que con la propia herramienta de arrastre y copia, hemos conseguido resultados muy rápidos. En los ejemplos que se muestran a continuación, se puede apreciar un ejemplo de los datos creados en hojas de cálculo, las extensiones de los archivos generados y el proceso de importación de dichos archivos al programa DBeaver y nuestra base de datos.



Datos_AcademiaAtentos										
Archivo Editar Ver Insertar Formato Datos Herramientas Extensiones Ayuda Última modificación hace 7 días										
100% € % 0.00 123 Arial 8 B I S A										
A1	idPoblacion									
	A	B	C	D	E	F	G	H	I	J
1	idPoblacion	Nombre	Provincia	CodigoPostal						
2	1	Aguadulce	Sevilla	41001						
3	2	Alanis	Sevilla	41002						
4	3	Albaida del Aljarafe	Sevilla	41003						
5	4	Alcalá de Guadaira	Sevilla	41004						
6	5	Alcalá del Río	Sevilla	41005						
7	6	Alcolea del Río	Sevilla	41006						
8	7	La Algaba	Sevilla	41007						
9	8	Algámitas	Sevilla	41008						
10	9	Almadén de la Plata	Sevilla	41009						
11	10	Almensilla	Sevilla	41010						
12	11	Arahal	Sevilla	41011						
13	12	Aznalcázar	Sevilla	41012						
14	13	Aznalcóllar	Sevilla	41013						
15	14	Badolatosa	Sevilla	41014						
16	15	Benacazón	Sevilla	41015						
17	16	Bollullos de la Mitad	Sevilla	41016						
18	17	Bormujos	Sevilla	41017						
19	18	Brenes	Sevilla	41018						
20	19	Burguillos	Sevilla	41019						
21	20	Las Cabezas de San Juan	Sevilla	41020						
22	21	Camas	Sevilla	41021						
23	22	La Campana	Sevilla	41022						
24	23	Cantillana	Sevilla	41023						
25	24	Cañada Rosal	Sevilla	41024						
26	25	Carmona	Sevilla	41025						
27	26	Castellana de Guadaj	Sevilla	41026						

Poblaciones

Personas

Empleados

Tutores

Alumnos





Recibos

Matricula

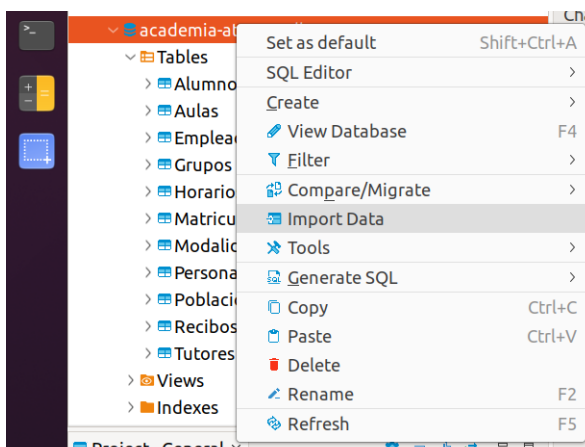
Horarios

Grupos

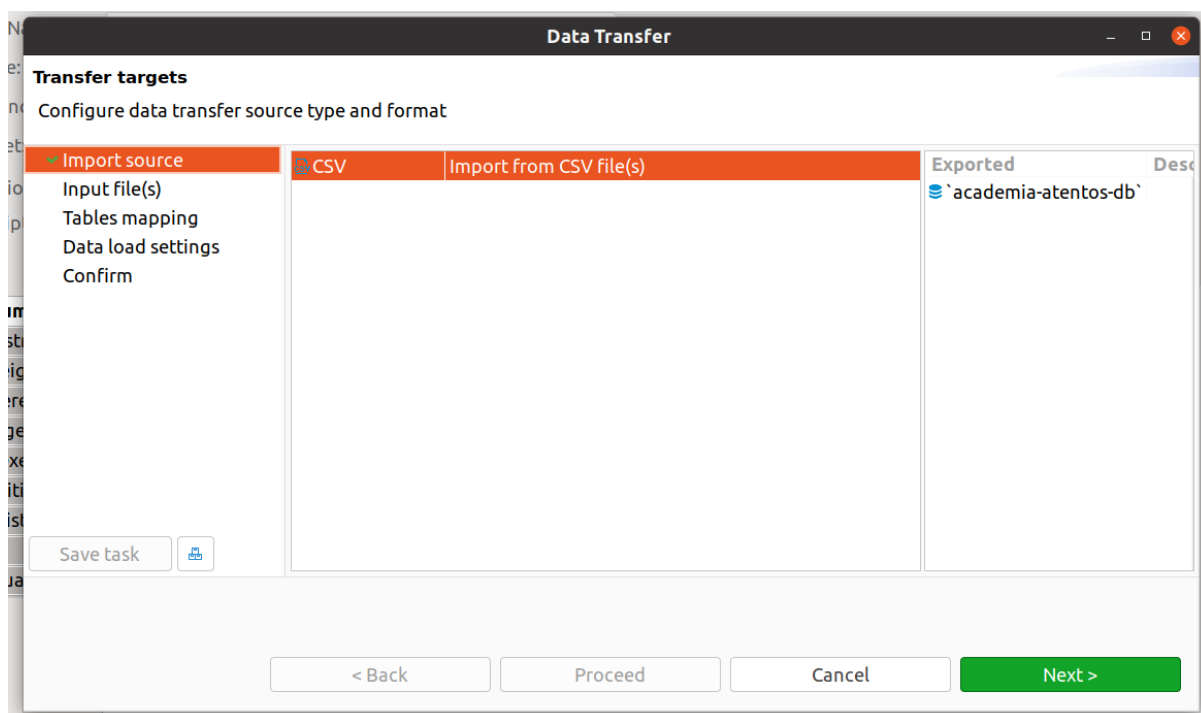
Aquí se puede apreciar un ejemplo de los datos generados en las hojas de cálculo.

	Datos_AcademiaAtentos - Recibos.csv	175,1 kB
	Datos_AcademiaAtentos - Tutores.csv	8,7 kB
	Datos_AcademiaAtentos - Alumnos.csv	16,0 kB
	Datos_AcademiaAtentos - Empleados.csv	15,6 kB

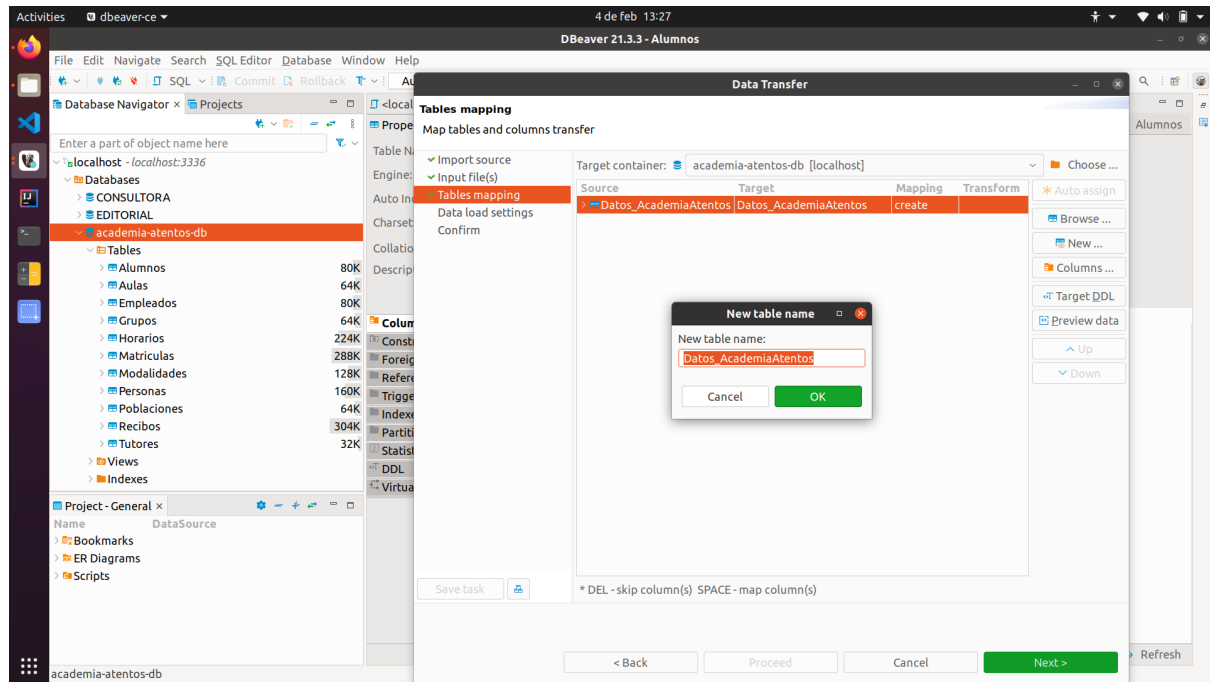
Archivos generados con extensión .csv



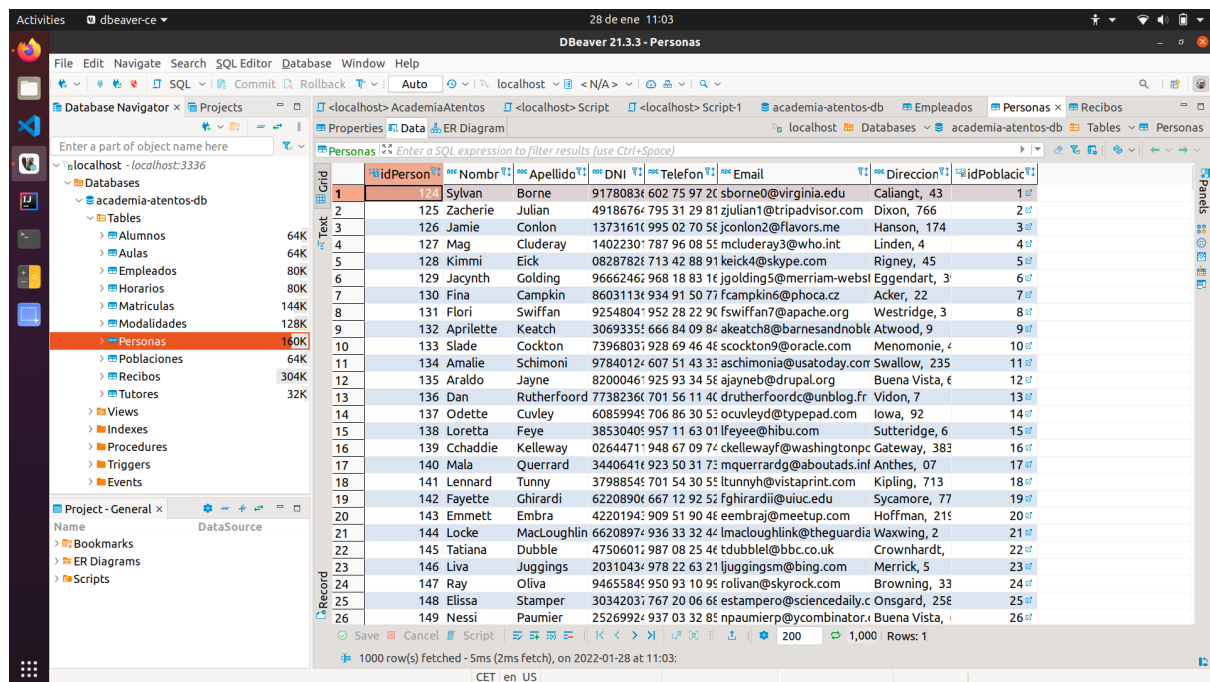
Importación de los archivos .csv al programa DBeaver dentro de nuestra base de datos.



Se indica que se trata de un archivo csv.



Le asignamos el nombre que debe aparecer en nuestra tabla. Cogera por defecto el nombre del archivo.



Observamos cómo se han incorporado los datos los datos a nuestra tabla.

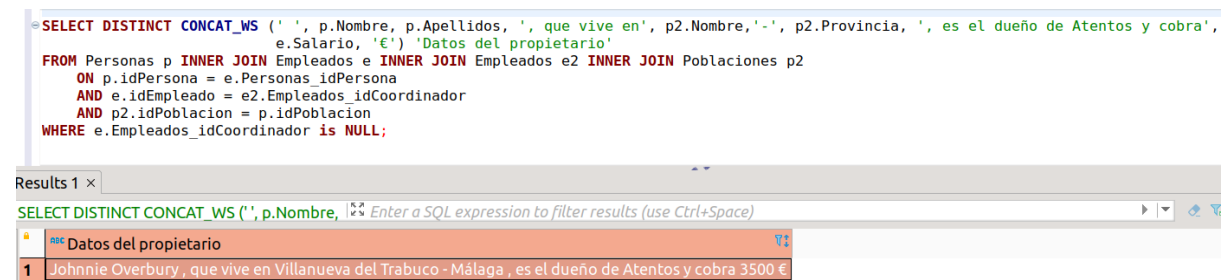
Consultas

Hemos realizado 6 consultas donde se puede apreciar la comunicación entre las tablas y el tipo de información interesante para la empresa.

Consulta I:

El empleado que sea coordinador de todos, porque es el dueño de la academia, debe aparecer toda la información en una sola columna, donde se recoja el nombre completo, la ciudad y provincia donde vive y cuánto cobra.

```
SELECT DISTINCT CONCAT_WS (' ', p.Nombre, p.Apellidos, ', que vive en', p2.Nombre, '-', p2.Provincia, ', es el dueño de Atentos y cobra',  
e.Salario, '€') 'Datos del propietario'  
FROM Personas p INNER JOIN Empleados e INNER JOIN Empleados e2 INNER JOIN Poblaciones p2  
ON p.idPersona = e.Personas_idPersona  
AND e.idEmpleado = e2.Empleados_idCoordinador  
AND p2.idPoblacion = p.idPoblacion  
WHERE e.Empleados_idCoordinador is NULL;
```



```
SELECT DISTINCT CONCAT_WS (' ', p.Nombre, p.Apellidos, ', que vive en', p2.Nombre, '-', p2.Provincia, ', es el dueño de Atentos y cobra',  
e.Salario, '€') 'Datos del propietario'  
FROM Personas p INNER JOIN Empleados e INNER JOIN Empleados e2 INNER JOIN Poblaciones p2  
ON p.idPersona = e.Personas_idPersona  
AND e.idEmpleado = e2.Empleados_idCoordinador  
AND p2.idPoblacion = p.idPoblacion  
WHERE e.Empleados_idCoordinador is NULL;
```

Results 1 x	
	SELECT DISTINCT CONCAT_WS (' ', p.Nombre, p.Apellidos, ', que vive en', p2.Nombre, '-', p2.Provincia, ', es el dueño de Atentos y cobra', e.Salario, '€') 'Datos del propietario'
1	Johnnie Overbury , que vive en Villanueva del Trabuco - Málaga , es el dueño de Atentos y cobra 3500 €

Consulta II:

Los alumnos de Castilleja de la Cuesta matriculados en Apoyo Escolar que asisten de manera presencial

```
SELECT p.Nombre, p.Apellidos
FROM Personas p INNER JOIN Alumnos a INNER JOIN Matriculas m INNER JOIN Modalidades m2 INNER JOIN Poblaciones p2
    ON p.idPersona = a.Personas_idPersona
    AND a.idAlumno = m.Alumnos_idAlumno
    AND m.Modalidades_idModalidad = m2.idModalidad
    AND p.idPoblacion = p2.idPoblacion
WHERE m2.Tipo_modalidad = 'Apoyo escolar'
    AND m2.Tipo_asistencia = 'Presencial'
    AND p2.Nombre = 'Castilleja de la Cuesta'
ORDER BY p.Nombre, p.Apellidos;
```

```
SELECT p.Nombre, p.Apellidos
FROM Personas p INNER JOIN Alumnos a INNER JOIN Matriculas m INNER JOIN Modalidades m2 INNER JOIN Poblaciones p2
    ON p.idPersona = a.Personas_idPersona
    AND a.idAlumno = m.Alumnos_idAlumno
    AND m.Modalidades_idModalidad = m2.idModalidad
    AND p.idPoblacion = p2.idPoblacion
WHERE m2.Tipo_modalidad = 'Apoyo escolar'
    AND m2.Tipo_asistencia = 'Presencial'
    AND p2.Nombre = 'Castilleja de la Cuesta'
ORDER BY p.Nombre, p.Apellidos;
```

Personas 1 x

SELECT p.Nombre, p.Apellidos FROM Persona *Enter a SQL expression to filter results (use Ctrl+Space)*

	Nombre	Apellidos
1	Alane	Clatworthy
2	Alisun	Cullin
3	Alvy	Romeo
4	Angus	Humbee
5	Aviva	Shearer
6	Batholomew	McFadyen
7	Blake	Downie
8	Carl	Hourican
9	Carrissa	Baynon
10	Chere	Bolino
11	Chere	Bolino
12	Clotilda	Dombrell
13	Coral	Camilli
14	Currie	Grinaugh
15	Dalenna	Derr
16	Darleen	Nesey
17	Darline	Grassick
18	Dino	Gozney
19	Dino	Gozney
20	Dugald	Goulborn

Save Cancel Script 200 54 Rows: 1 54 row(s) fetched - 11r

Consulta III:

Los alumnos que tienen clase los lunes a las 20:00 en cualquier aula y de cualquier asignatura

```
SELECT CONCAT_WS(' ',p.Nombre, p.Apellidos) Alumno, g.Asignatura, CONCAT('Aula: ', a2.idAula) Aula
FROM Personas p INNER JOIN Alumnos a INNER JOIN Horarios h INNER JOIN Aulas a2 INNER JOIN Grupos g
    ON p.idPersona = a.Personas_idPersona
    AND h.Alumnos_idAlumno = a.idAlumno
    AND h.Aulas_idAula = a2.idAula
    AND h.Grupos_idGrupo = g.idGrupo
WHERE h.Dia = 'Lunes' AND h.Hora = '20:00'
ORDER BY a2.idAula;
```

```
SELECT CONCAT_WS(' ',p.Nombre, p.Apellidos) Alumno, g.Asignatura, CONCAT('Aula: ', a2.idAula) Aula
FROM Personas p INNER JOIN Alumnos a INNER JOIN Horarios h INNER JOIN Aulas a2 INNER JOIN Grupos g
    ON p.idPersona = a.Personas_idPersona
    AND h.Alumnos_idAlumno = a.idAlumno
    AND h.Aulas_idAula = a2.idAula
    AND h.Grupos_idGrupo = g.idGrupo
WHERE h.Dia = 'Lunes' AND h.Hora = '20:00'
ORDER BY a2.idAula;
```

Grupos 1 x

SELECT CONCAT_WS(' ',p.Nombre, p.Apell

	Alumno	Asignatura	Aula
6	Lothario Filan	Geología	Aula: 41
7	Lamar Sambals	Naturales	Aula: 42
8	Erhart Aldred	Música	Aula: 46
9	Clotilda Dombrell	Música	Aula: 52
10	Sutherland Pechell	Educación física	Aula: 53
11	Ernaline Paslow	Biología	Aula: 62
12	Tabbatha Abrahms	Latín	Aula: 64
13	Dyna Sandeson	Filosofía	Aula: 75
14	Alvin Albisser	Física	Aula: 83
15	Currie Grinaugh	Naturales	Aula: 96
16	Gerhard Wilden	Geología	Aula: 98
17	Alejandrina McLae	Educación audiovi	Aula: 100
18	Reginauld Etching	Música	Aula: 101
19	Jesselyn Burrass	Catalán	Aula: 103
20	Philly Baker	Catalán	Aula: 104
21	Clyde Pisco	Sociales	Aula: 105
22	Lishe Braidon	Música	Aula: 107
23	Rik Oller	Matemáticas aca	Aula: 113
24	Lyn Hansen	Francés	Aula: 118
25	Berte Sawl	Cultura clásica	Aula: 121

Save Cancel Script 200 38 Rows: 1

38 row(s) fetched - 5ms, on 2022-02-06 at 20:48:29


Consulta IV:


Poblaciones de Sevilla con más de una persona relacionada con la academia

```
SELECT p2.Nombre, COUNT(*) Cantidad
FROM Personas p INNER JOIN Poblaciones p2
    ON p.IdPoblacion = p2.IdPoblacion
WHERE p2.Provincia = 'Sevilla'
GROUP BY p2.Nombre
HAVING COUNT(Cantidad) > 1;
```

```
SELECT p2.Nombre, COUNT(*) Cantidad
FROM Personas p INNER JOIN Poblaciones p2
    ON p.IdPoblacion = p2.IdPoblacion
WHERE p2.Provincia = 'Sevilla'
GROUP BY p2.Nombre
HAVING COUNT(Cantidad) > 1;
```

Personas 1

 Poblaciones 3

 Poblaciones 3 ×

SELECT p2.Nombre, COUNT(*) Cantidad FI

Enter a SQL ex

	ABC Nombre	123 Cantidad
1	Bormujos	65
2	Castilleja de la Cuesta	105
3	Tomares	281

Consulta V:

Nombre y apellidos de los alumnos en una misma columna que pagaron más que la media durante el año 2020 en Tomares. Se debe indicar la cantidad pagada y la fecha del recibo

```
SELECT CONCAT_WS(' ', p.Nombre, p.Apellidos) 'Alumno', DATE_FORMAT(r.Fecha, "%d %M %Y") 'Fecha de Pago', CONCAT(r.Cantidad, '€')
Cantidad
FROM Personas p INNER JOIN Alumnos a INNER JOIN Recibos r INNER JOIN Poblaciones p2
ON p.idPersona = a.Personas_idPersona
AND r.Alumnos_idAlumno = a.idAlumno
AND p2.idPoblacion = p.idPoblacion
WHERE r.Cantidad > (SELECT AVG(r2.Cantidad) FROM Recibos r2)
AND YEAR(r.Fecha) = 2020
AND p2.Nombre = 'Tomares'
ORDER BY p.Nombre;
```

```
SELECT CONCAT_WS(' ', p.Nombre, p.Apellidos) 'Alumno', DATE_FORMAT(r.Fecha, "%d %M %Y") 'Fecha de Pago', CONCAT(r.Cantidad, '€') Cantidad
FROM Personas p INNER JOIN Alumnos a INNER JOIN Recibos r INNER JOIN Poblaciones p2
ON p.idPersona = a.Personas_idPersona
AND r.Alumnos_idAlumno = a.idAlumno
AND p2.idPoblacion = p.idPoblacion
WHERE r.Cantidad > (SELECT AVG(r2.Cantidad) FROM Recibos r2)
AND YEAR(r.Fecha) = 2020
AND p2.Nombre = 'Tomares'
ORDER BY p.Nombre;
```

	Alumno	Fecha de Pago	Cantidad
4	Arlena Calwell	02 May 2020	199€
5	Arlena Calwell	24 November 2020	293€
6	Art Iwaszkiewicz	07 September 2020	297€
7	Art Iwaszkiewicz	22 June 2020	225€
8	Basilio Yewdall	12 October 2020	221€
9	Basilio Langelay	10 April 2020	228€
10	Berte Sawl	12 November 2020	327€
11	Clarine Edie	10 May 2020	266€
12	Danyelle McKeemar	14 June 2020	267€
13	Darell Yanele	16 March 2020	206€
14	Denice Minot	03 April 2020	201€
15	Ernaline Paslow	25 January 2020	204€
16	Franklin Portwain	05 August 2020	295€
17	Gilbert Treffrey	03 May 2020	251€
18	Ginger Palphreyman	15 August 2020	312€
19	Giselle Olivey	15 March 2020	278€
20	Harwell Rappaport	27 July 2020	339€
21	HelenElizabeth Starl	31 August 2020	191€
22	Hodge Overington	25 January 2020	290€
23	Holly Aitchison	11 April 2020	207€
24	Ingunna Bullier	28 September 2020	279€

Results 1 x
 SELECT CONCAT_WS(' ', p.Nombre, p.Apellidos) 'Alumno', DATE_FORMAT(r.Fecha, "%d %M %Y") 'Fecha de Pago', CONCAT(r.Cantidad, '€') Cantidad
 FROM Personas p INNER JOIN Alumnos a INNER JOIN Recibos r INNER JOIN Poblaciones p2
 ON p.idPersona = a.Personas_idPersona
 AND r.Alumnos_idAlumno = a.idAlumno
 AND p2.idPoblacion = p.idPoblacion
 WHERE r.Cantidad > (SELECT AVG(r2.Cantidad) FROM Recibos r2)
 AND YEAR(r.Fecha) = 2020
 AND p2.Nombre = 'Tomares'
 ORDER BY p.Nombre;

Save Cancel Script 200 56 Rows: 1 56 row(s) fetched - 11ms, on 2022-02-06 at 11:46:41

Consulta VI:

El profesor que haya estado en activo durante más tiempo, pero que ya no trabaje en la academia

```
SELECT CONCAT_WS(' ', p.Nombre, p.Apellidos) Profesor, e.Fecha_alta, e.Fecha_baja, DATEDIFF(e.Fecha_baja, e.Fecha_alta)
Dias_trabajados
FROM Personas p INNER JOIN Empleados e
ON p.idPersona = e.Personas_idPersona
WHERE e.Puesto = 'Profesor'
AND DATEDIFF(e.Fecha_alta, e.Fecha_baja) =
(SELECT MIN(DATEDIFF(e2.Fecha_alta, e2.Fecha_baja))
FROM Empleados e2
WHERE e2.Puesto = 'Profesor');
```

```
= SELECT CONCAT_WS(' ', p.Nombre, p.Apellidos) Profesor, e.Fecha_alta, e.Fecha_baja, DATEDIFF(e.Fecha_baja, e.Fecha_alta) Dias_trabajados
FROM Personas p INNER JOIN Empleados e
ON p.idPersona = e.Personas_idPersona
WHERE e.Puesto = 'Profesor'
AND DATEDIFF(e.Fecha_alta, e.Fecha_baja) = (SELECT MIN(DATEDIFF(e2.Fecha_alta, e2.Fecha_baja))
FROM Empleados e2
WHERE e2.Puesto = 'Profesor');
```

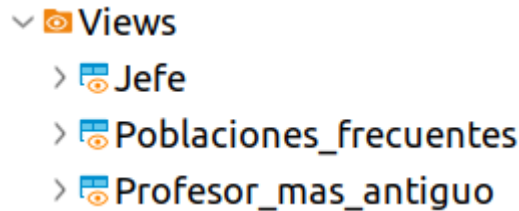
Empleados 1 x

SELECT CONCAT_WS(' ', p.Nombre, p.Apellidos) Profesor, e.Fecha_alta, e.Fecha_baja, DATEDIFF(e.Fecha_baja, e.Fecha_alta) Dias_trabajados

	Profesor	Fecha_alta	Fecha_baja	Dias_trabajados
1	Tiertza Tillett	2011-07-07	2020-01-06	3,105

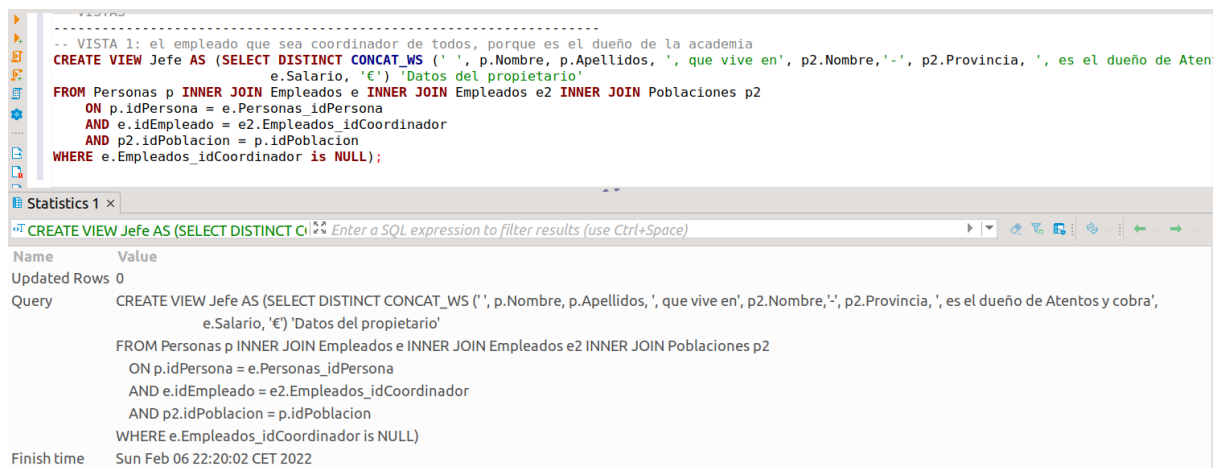
Vistas

A partir de las consultas que hemos creado en el apartado anterior, creamos las vistas. Se han creado 3 vistas, una para los diferentes tipos de consultas que se han planteado.



1. Vista Jefe: el empleado que sea coordinador de todos, porque es el dueño de la academia

```
CREATE VIEW Jefe AS (SELECT DISTINCT CONCAT_WS (' ', p.Nombre, p.Apellidos, ', que vive en', p2.Nombre, '-', p2.Provincia, ', es el dueño de Atentos y cobra', e.Salario, '€') 'Datos del propietario'
FROM Personas p INNER JOIN Empleados e INNER JOIN Empleados e2 INNER JOIN Poblaciones p2
ON p.idPersona = e.Personas_idPersona
AND e.idEmpleado = e2.Empleados_idCoordinador
AND p2.idPoblacion = p.idPoblacion
WHERE e.Empleados_idCoordinador is NULL);
```



2. Vista Poblaciones frecuentes: poblaciones de Sevilla con más cantidad de personas relacionadas con la academia

```
CREATE VIEW Poblaciones_frecuentes AS
```

```
(SELECT p2.Nombre, COUNT(*) Cantidad  
FROM Personas p INNER JOIN Poblaciones p2  
ON p.IdPoblacion = p2.IdPoblacion  
WHERE p2.Provincia = 'Sevilla'  
GROUP BY p2.Nombre  
HAVING COUNT(Cantidad) > 1);
```

The screenshot shows a SQL IDE interface. The top pane displays the SQL code for creating a view named 'Poblaciones_frecuentes'. The code is as follows:

```
-- VISTA 2: Poblaciones de Sevilla con más cantidad de personas relacionadas con la academia  
CREATE VIEW Poblaciones_frecuentes AS (SELECT p2.Nombre, COUNT(*) Cantidad  
FROM Personas p INNER JOIN Poblaciones p2  
ON p.IdPoblacion = p2.IdPoblacion  
WHERE p2.Provincia = 'Sevilla'  
GROUP BY p2.Nombre  
HAVING COUNT(Cantidad) > 1);
```

The bottom pane shows the 'Statistics 1' window. It contains the following information:

Name	Value
Updated Rows	0
Query	-- VISTA 2: Poblaciones de Sevilla con más cantidad de personas relacionadas con la academia CREATE VIEW Poblaciones_frecuentes AS (SELECT p2.Nombre, COUNT(*) Cantidad FROM Personas p INNER JOIN Poblaciones p2 ON p.IdPoblacion = p2.IdPoblacion WHERE p2.Provincia = 'Sevilla' GROUP BY p2.Nombre HAVING COUNT(Cantidad) > 1)
Finish time	Sun Feb 06 22:25:46 CET 2022

3. Vista Profesor más antiguo: el profesor que haya estado en activo durante más tiempo, pero que ya no trabaje en la academia

```
CREATE VIEW Profesor_mas_antiguo AS
```

```
(SELECT CONCAT_WS(' ', p.Nombre, p.Apellidos) Profesor, e.Fecha_alta, e.Fecha_baja,
    DATEDIFF(e.Fecha_baja, e.Fecha_alta) Dias_trabajados
FROM Personas p INNER JOIN Empleados e
    ON p.idPersona = e.Personas_idPersona
WHERE e.Puesto = 'Profesor'
AND DATEDIFF(e.Fecha_alta, e.Fecha_baja) =
    (SELECT MIN(DATEDIFF(e2.Fecha_alta, e2.Fecha_baja))
    FROM Empleados e2
    WHERE e2.Puesto = 'Profesor'));
```

The screenshot shows a SQL IDE interface. The top pane displays the SQL code for creating the view 'Profesor_mas_antiguo'. The bottom pane shows the 'Statistics 1' window, which includes a table with the following data:

Name	Value
Updated Rows	0
Query	<p>– VISTA 3: el profesor que haya estado en activo durante más tiempo, pero que ya no trabaje en la academia</p> <pre>CREATE VIEW Profesor_mas_antiguo AS (SELECT CONCAT_WS(' ', p.Nombre, p.Apellidos) Profesor, e.Fecha_alta, e.Fecha_baja, DATEDIFF(e.Fecha_baja, e.Fecha_alta) Dias_trabajados FROM Personas p INNER JOIN Empleados e ON p.idPersona = e.Personas_idPersona WHERE e.Puesto = 'Profesor' AND DATEDIFF(e.Fecha_alta, e.Fecha_baja) = (SELECT MIN(DATEDIFF(e2.Fecha_alta, e2.Fecha_baja)) FROM Empleados e2 WHERE e2.Puesto = 'Profesor'));</pre>
Finish time	Sun Feb 06 22:31:19 CET 2022