

# Ejercicio 4

Persona = (DNI, Apellido, Nombre, Fecha\_Nacimiento, Estado\_Civil, Genero)

Alumno = (DNI (fk), Legajo, Anio\_Ingreso)

Profesor = (DNI (fk), Matricula, Nro\_Expediente)

Titulo = (Cod\_Titulo, Nombre, Descripcion)

Titulo-Profesor = (Cod\_Titulo (fk), DNI (fk), Fecha)

Curso = (Cod\_Curso, Nombre, Descripcion, Fecha\_Creacion, Duracion)

Alumno-Curso = (DNI (fk), Cod\_Curso (fk), Anio, Desempenio, Calificacion)

Profesor-Curso = (DNI (fk), Cod\_Curso (fk), Fecha\_Desde, Fecha\_Hasta?)

1. Listar DNI, legajo y apellido y nombre de todos los alumnos que tengan año de ingreso inferior a 2014

```
SELECT DNI, Legajo, Apellido, Nombre  
FROM Alumno NATURAL JOIN Persona  
WHERE Anio_Ingreso < 2014
```

2. Listar DNI, matrícula, apellido y nombre de los profesores que dictan cursos que tengan más de 100 horas de duración. Ordenar por DNI.

```
SELECT DISTINCT DNI, Matricula, Apellido, Nombre  
FROM Profesor NATURAL JOIN Persona NATURAL JOIN Profesor_Curso N  
ATURAL JOIN (SELECT Cod_Curso FROM Curso WHERE Duracion > 100) A  
S C  
ORDER BY DNI
```

3. Listar el DNI, Apellido, Nombre, Género y Fecha de nacimiento de los alumnos inscriptos al curso con nombre "Diseño de Bases de Datos" en 2023.

```
SELECT DNI, Apellido, Nombre, Genero, Fecha_Nacimiento  
FROM Alumno NATURAL JOIN Alumno_Curso NATURAL JOIN Persona NAT  
URAL JOIN  
(SELECT Cod_Curso FROM Curso WHERE Nombre = 'Diseño de Bases De
```

```
Datos') AS C  
WHERE Anio = 2023
```

4. Listar el DNI, Apellido, Nombre y Calificación de aquellos alumnos que obtuvieron una calificación superior a 8 en algún curso que dicta el profesor "Juan Garcia". Dicho listado deberá estar ordenado por Apellido y nombre.

```
SELECT DNI, Apellido, Nombre, Calificacion  
FROM Alumno NATURAL JOIN Alumno_Curso NATURAL JOIN Persona NAT  
URAL JOIN  
(SELECT Cod_Curso  
FROM Profesor_Curso NATURAL JOIN Persona  
WHERE Nombre = "Juan" and Apellido = "Garcia") AS C  
ORDER BY Apellido, Nombre
```

5. Listar el DNI, Apellido, Nombre y Matrícula de aquellos profesores que posean más de 3 títulos. Dicho listado deberá estar ordenado por Apellido y Nombre.

```
SELECT DNI, Apellido, Nombre, Matricula  
FROM Persona NATURAL JOIN Profesor NATURAL JOIN Titulo_Profesor  
GROUP BY DNI, Apellido, Nombre, Matricula  
HAVING COUNT(Cod_Titulo) > 3  
ORDER BY Apellido, Nombre
```

6. Listar el DNI, Apellido, Nombre, Cantidad de horas y Promedio de horas que dicta cada profesor. La cantidad de horas se calcula como la suma de la duración de todos los cursos que dicta.

```
SELECT DNI, Apellido, Nombre, SUM(Duracion) as CantidadDeHoras, AVG  
(Duracion) as PromedioDeHoras  
FROM Persona NATURAL JOIN Profesor NATURAL JOIN Profesor_Curso N  
ATURAL JOIN  
(SELECT Cod_Curso, Duracion FROM Curso) as C  
GROUP BY DNI, Apellido, Nombre
```

7. Listar Nombre y Descripción del curso que posea más alumnos inscriptos y del que posea menos alumnos inscriptos durante 2024.

```
(SELECT Nombre, Descripcion  
FROM Curso NATURAL JOIN Alumno_Curso  
WHERE Anio = 2024  
GROUP BY Cod_Curso, Nombre, Descripcion  
HAVING COUNT(DNI) >= all (SELECT COUNT(DNI) FROM Curso NATURAL  
JOIN Alumno_Curso WHERE Anio = 2024 GROUP BY Cod_Curso)  
  
UNION  
  
SELECT Nombre, Descripcion  
FROM Curso NATURAL JOIN Alumno_Curso  
WHERE Anio = 2024  
GROUP BY Cod_Curso, Nombre, Descripcion  
HAVING COUNT(DNI) <= all(SELECT COUNT(DNI) FROM Curso NATURAL  
JOIN Alumno_Curso WHERE Anio = 2024 GROUP BY Cod_Curso))
```

```
SELECT Nombre, Descripcion  
FROM Curso NATURAL JOIN Alumno_Curso  
WHERE Anio = 2024  
GROUP BY Cod_Curso, Nombre, Descripcion  
HAVING COUNT(*) >= ALL (SELECT COUNT(*) FROM Alumno_Curso WHERE Anio = 2024 GROUP BY Cod_Curso) or COUNT(*) <= ALL (SELECT COUNT(*) FROM Alumno_Curso WHERE Anio = 2024 GROUP BY Cod_Curso)
```

8. Listar el DNI, Apellido, Nombre y Legajo de alumnos que realizaron cursos con nombre conteniendo el string 'BD' durante 2022 pero no realizaron ningún curso durante 2023.

```
SELECT alu.DNI, Apellido, p.Nombre, Legajo  
FROM Persona p NATURAL JOIN Alumno alu NATURAL JOIN Alumno_Curso a INNER JOIN Curso c ON (a.Cod_Curso = c.Cod_Curso)  
WHERE c.Nombre like '%BD%' and Anio = 2022 and alu.DNI NOT IN (SELECT DNI
```

```
FROM Alumno_Curso  
WHERE Anio = 2023)
```

9. Agregar un profesor con los datos que prefiera y agregarle el título con código: 25.

```
INSERT INTO Persona (DNI, Apellido, Nombre, Fecha_Nacimiento, Estado_Civil, Genero) VALUES (111, "a", "a", '2025-11-10', "Soltero", "F");  
INSERT INTO Profesor (DNI, Matricula, Nro_Expediente) VALUES (111, 111, 111);  
INSERT INTO Titulo_Profesor (Cod_Titulo, DNI, Fecha) VALUES (25, 111, '2025-11-10')
```

10. Modificar el estado civil del alumno cuyo legajo es '2020/09', el nuevo estado civil es divorciado.

```
UPDATE Persona NATURAL JOIN Alumno SET Estado_Civil = 'Divorciado'  
WHERE legajo = '2020/09'
```

```
UPDATE Persona SET Estado_Civil = 'Divorciado' WHERE DNI = (SELECT DNI FROM Alumno WHERE Legajo = '2020/09');
```

11. Dar de baja el alumno con DNI 30568989. Realizar todas las bajas necesarias para no dejar el conjunto de relaciones en un estado inconsistente.

```
// prestar atención al orden!!!!  
DELETE FROM Alumno_Curso WHERE DNI = 30568989;  
DELETE FROM Alumno WHERE DNI = 30568989;  
DELETE FROM Persona WHERE DNI = 30568989
```