**PUNTO X**

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

ALUMNO = (DNI, Legajo, Año\_Ingreso)

PROFESOR = (DNI, Matricula, Nro\_Expediente)

TITULO = (Cod\_Titulo, Nombre, Descripción)

TITULO-PROFESOR = (Cod\_Titulo, DNI, Fecha)

CURSO = (Cod\_Curso, Nombre, Descripción, Fecha\_Creacion, Duracion)

ALUMNO-CURSO = (DNI, Cod\_Curso, Año, Desempeño, Calificación)

PROFESOR-CURSO = (DNI, Cod\_Curso, Fecha\_Desde, Fecha\_Hasta)

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

SELECT a.DNI, a.legajo, p.nombre, p.apellido

FROM (Alumno a

NATURAL JOIN Persona p)

WHERE (a.Año\_Ingreso < 2014)

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

SELECT pr.DNI, pr.matricula, p.nombre, p.apellido

FROM (((Profesor pr

NATURAL JOIN Persona p)

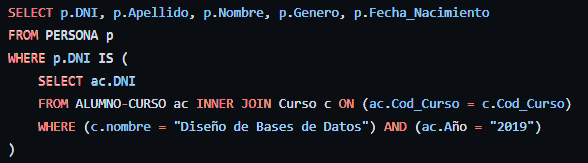
NATURAL JOIN Profesor-Curso pc)

INNER JOIN Curso c ON (pc.cod\_curso = c.cod\_curso))

WHERE (c.duración > 100)

ORDER BY pr.DNI, pr.matricula

**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 2019.**

SELECT a.DNI, p.nombre, p.apellido, p.genero, p.fecha\_nacimiento

FROM (((Alumno a

NATURAL JOIN Persona p)

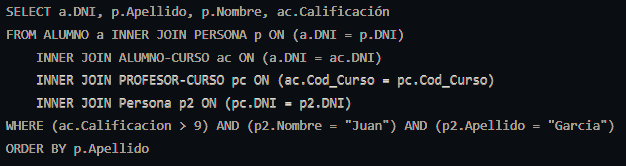
NATURAL JOIN Alumno-Curso ac)

INNER JOIN Curso c ON (c.cod\_curso = ac.cod\_curso))

WHERE (ac.año = 2019) and (c.nombre = “Diseño de Bases de Datos”)

**4. Listar el DNI, Apellido, Nombre y Calificación de aquellos alumnos que obtuvieron una calificación superior a 9 en los cursos que dicta el profesor “Juan Garcia”. Dicho listado deberá estar ordenado por Apellido.**

SELECT a.DNI, p.nombre, p.apellido, ac.calificación

FROM ((Alumno a

NATURAL JOIN Persona p)

NATURAL JOIN Alumno-Curso ac)

WHERE (ac.calificación > 9) (ac.cod\_curso IN (

SELECT pc.cod\_curso

FROM ((Profesor pr

NATURAL JOIN Persona p)

NATURAL JOIN Profesor-Curso pc)

WHERE (p.nombre = “Juan”) and (p.apellido = “García”)

)

ORDER BY p.apellido, p.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 pr.DNI, pr.matricula, p.nombre, p.apellido

FROM ((Profesor pr

NATURAL JOIN Persona p)

NATURAL JOIN Titulo-Profesor tp)

GROUP BY pr.DNI, pr.matricula, p.nombre, p.apellido

HAVING COUNT(\*) > 3

ORDER BY p.apellido, p.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 pr.DNI, p.nombre, p.apellido, SUM(c.duracion) as cantHoras, AVG(c.duracion) as promedio

FROM (((Profesor pr

NATURAL JOIN Persona p)

LEFT JOIN Profesor-Curso pc ON (pc.DNI = pr.DNI))

LEFT JOIN Curso c ON (c.cod\_curso = pc.cod\_curso))

GROUP BY pr.DNI, p.nombre, p.apellido

**7. Listar Nombre, Descripción del curso que posea más alumnos inscriptos y del que posea menos alumnos inscriptos durante 2019.**

(SELECT c.nombre, c.descripcion

FROM (Curso c

NATURAL JOIN Alumno-Curso ac)

WHERE (ac.año = 2019)

GROUP BY c.cod\_curso, c.nombre, c.descripcion -> si o si por el identificador

HAVING COUNT(\*) >= ALL (

SELECT COUNT(\*)

FROM (Curso c

NATURAL JOIN Alumno-Curso ac)

WHERE (ac.año = 2019)

GROUP BY c.cod\_curso

)

)

UNION

(SELECT c.nombre, c.descripcion

FROM (Curso c

NATURAL JOIN Alumno-Curso ac)

WHERE (ac.año = 2019)

GROUP BY c.cod\_curso, c.nombre, c.descripcion

HAVING COUNT(\*) <= ALL (

SELECT COUNT(\*)

FROM (Curso c

NATURAL JOIN Alumno-Curso ac)

WHERE (ac.año = 2019)

GROUP BY c.cod\_curso

)

)

**8. Listar el DNI, Apellido, Nombre, Legajo de alumnos que realizaron cursos con nombre conteniendo el string ‘BD’ durante 2018 pero no realizaron ningún curso durante 2019.**

SELECT a.DNI, a.legajo, p.nombre, p.apellido

FROM (((Alumno a

NATURAL JOIN Persona p)

NATURAL JOIN Alumno-Curso ac)

INNER JOIN Curso c ON (c.cod\_curso = ac.cod\_curso))

WHERE (ac.año = 2018) and (c.nombre LIKE “%BD%”) and (a.DNI NOT IN (

SELECT ac.DNI

FROM Alumno-Curso ac

WHERE (ac.año = 2019)

)

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

INSERT INTO Título (Cod\_Titulo, Nombre, Descripción)

VALUES (25, “Titulo”, “Yes”);

INSERT INTO Profesor (DNI, Matricula, Nro\_Expediente)

VALUES (44124642, 123, 123)

INSERT INTO Titulo-Profesor (Cod\_Titulo, DNI, Fecha)

VALUES (25, 44124642, "22/10/2022")

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

UPDATE Persona

SET estado\_civil = “Divorciado”

WHERE (p.DNI IN (

SELECT a.DNI

FROM Alumno a

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 estado inconsistente.**

DELETE FROM Alumno-Curso

WHERE DNI = 30568989

DELETE FROM Persona

WHERE DNI = 30568989

DELETE FROM Alumno

WHERE DNI = 30568989