

---

# Consultas JOIN

Unidad 7  
Bases de Datos - 1º DAW

---

# Contenidos

- 1. Introducción**
- 2. Nombres cualificados**
- 3. CROSS JOIN**
- 4. INNER JOIN**
- 5. NATURAL JOIN**
- 6. OUTER JOIN**

# 1. Introducción

U7. Consultas JOIN

**Bases de Datos**  
**1º DAW**

Francisco Javier Castillo Jiménez

- 1.1. Introducción
- 1.2. Ejemplo de Base de Datos
- 1.3. SELECT de datos de diferentes tablas

# 1.1 Introducción

Las consultas sobre varias tablas van a permitir mostrar columnas de más de una tabla, especificando si se realiza un JOIN de tipo cruzado, interno o externo.

Se especificará, en la cláusula FROM, cuáles son las tablas a usar y cómo se relacionan entre sí (CROSS JOIN, INNER JOIN, OUTER JOIN)

Existen dos formas:

- **SQL1**, consistente en realizar el producto cartesiano de las tablas y, mediante un filtro, relacionar los datos que tienen en común (PK=FK)
- **SQL2**, incluyendo cláusulas de tipo JOIN.

## 1.2 Ejemplo Base de Datos (DDL)

```
CREATE                                DATABASE                                iespuertasdelcampo;  
USE iespuertasdelcampo;
```

```
CREATE TABLE turno (  
    idTurno INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    nombreTurno VARCHAR(100) NOT NULL,  
    horalnicio TIME );
```

```
CREATE TABLE grupo (  
    idGrupo CHAR(4) PRIMARY KEY,  
    nombreGrupo VARCHAR(100) NOT NULL,  
    idTurno INT UNSIGNED NOT NULL,  
    maxEstudiantes INT NOT NULL,  
    FOREIGN KEY (idTurno) REFERENCES turno(idTurno) ON UPDATE CASCADE );
```

## 1.2 Ejemplo Base de Datos (DDL)

```
CREATE TABLE alumnado (  
  expedienteA INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
  nombreA VARCHAR(100) NOT NULL,  
  ap1 VARCHAR(99) NOT NULL,  
  ap2 VARCHAR(99),  
  fechaNacA DATE NOT NULL,  
  repetidor ENUM('sí', 'no') NOT NULL,  
  email VARCHAR(99),  
  idGrupo CHAR(4),  
  FOREIGN KEY (idGrupo) REFERENCES grupo(idGrupo) ON UPDATE CASCADE  
);
```

## 1.2 Ejemplo Base de Datos DML (INSERT)

**INSERT INTO** turno (nombreTurno, HoralInicio) **VALUES**  
("Mañana", "08:00:00"), ("Tarde", "15:30:00");

**INSERT INTO** grupo **VALUES** ('1DAW', "1º Desarrollo Aplicaciones Web", 1, 25), ('2DAW', "2º Desarrollo Aplicaciones Web", 1, 20), ('1SMM', "1º Sistemas Microinformáticos y Redes", 1, 30), ('2SMM', "2º Sistemas Microinformáticos y Redes", 1, 25), ('1SMT', "1º Sistemas Microinformáticos y Redes", 2, 30), ('2SMT', "2º Sistemas Microinformáticos y Redes", 2, 25), ('1ASI', "1º Administración de Sistemas Informáticos en Red", 2, 25), ('2ASI', "2º Administración de Sistemas Informáticos en Red", 2, 20);

## 1.2 Ejemplo Base de Datos DML (INSERT)

```
INSERT INTO alumnado VALUES (1, 'María', 'Sánchez', 'Pérez', '1990/12/01',  
'no', "maria@ipc.es", "1DAW");
```

```
INSERT INTO alumnado VALUES (2, 'Juan', 'Sáez', 'Vega', '1998/04/02', 'no',  
"juan@ipc.es", "2DAW");
```

```
INSERT INTO alumnado VALUES (3, 'Pepe', 'Ramírez', 'Gea', '1988/01/03',  
'no', NULL, "1DAW");
```

```
INSERT INTO alumnado VALUES (4, 'Lucía', 'Sánchez', 'Ortega',  
'1993/06/13', 'sí', "lucia@ipc.es", "1SMM");
```

```
INSERT INTO alumnado VALUES (5, 'Paco', 'Martínez', 'López', '1995/11/24',  
'no', "paco@ipc.es", "2SMM");
```

```
INSERT INTO alumnado VALUES (6, 'Irene', 'Gutiérrez', 'Sánchez',  
'1991/03/28', 'sí', NULL, "1SMM");
```



## 1.2 Ejemplo Base de Datos DML (INSERT)

```
INSERT INTO alumnado VALUES (7, 'Cristina', 'Fernández',  
'Ramírez', '1996/09/17', 'no', "cris@ipc.es", "1ASI");
```

```
INSERT INTO alumnado VALUES (8, 'Antonio', 'Carretero',  
'Ortega', '1994/05/20', 'sí', "antonio@ipc.es", "2ASI");
```

```
INSERT INTO alumnado VALUES (9, 'Manuel', 'Domínguez',  
'Hernández', '1999/07/08', 'no', NULL, "1SMT");
```

```
INSERT INTO alumnado VALUES (10, 'Daniel', 'Moreno',  
'Ruiz', '1998/02/03', 'no', NULL, "2SMT");
```

# 1.2 Ejemplo Base de Datos DML (SELECT)

1 • `SELECT * FROM iespuertasdelcampo.turno;`

#	id	nombreTurno	horaInicio
1	1	Mañana	08:00:00
2	2	Tarde	15:30:00
*	NULL	NULL	NULL

1 • `SELECT * FROM iespuertasdelcampo.alumnado;`

#	expedienteA	nombreA	ap1	ap2	fechaNacA	repetido	email	grupo
1	1	María	Sánchez	Pérez	1990-12-01	no	maria@ipc.es	1DAW
2	2	Juan	Sáez	Vega	1998-04-02	no	juan@ipc.es	2DAW
3	3	Pepe	Ramírez	Gea	1988-01-03	no	NULL	1DAW
4	4	Lucía	Sánchez	Ortega	1993-06-13	sí	lucia@ipc.es	1SMM
5	5	Paco	Martínez	López	1995-11-24	no	paco@ipc.es	2SMM
6	6	Irene	Gutiérrez	Sánchez	1991-03-28	sí	NULL	1SMM
7	7	Cristina	Fernández	Ramírez	1996-09-17	no	cris@ipc.es	1ASI
8	8	Antonio	Carretero	Ortega	1994-05-20	sí	antonio@ipc.es	2ASI
9	9	Manuel	Dominguez	Hernández	1999-07-08	no	NULL	1SMT
10	10	Daniel	Moreno	Ruiz	1998-02-03	no	NULL	2SMT
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

1 • `SELECT * FROM iespuertasdelcampo.grupo;`

#	id	nombreGrupo	turno	maxEstudiante
1	1ASI	1º Administración de Sistemas Infor...	2	25
2	1DAW	1º Desarrollo Aplicaciones Web	1	25
3	1SMM	1º Sistemas Microinformáticos y Re...	1	30
4	1SMT	1º Sistemas Microinformáticos y Re...	2	30
5	2ASI	2º Administración de Sistemas Infor...	2	20
6	2DAW	2º Desarrollo Aplicaciones Web	1	20
7	2SMM	2º Sistemas Microinformáticos y Re...	1	25
8	2SMT	2º Sistemas Microinformáticos y Re...	2	25
*	NULL	NULL	NULL	NULL

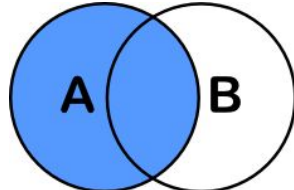
## 1.3 SELECT de datos de tablas diferentes

Pero, ¿y si quisiera consultar el nombre y apellidos del alumnado además del nombre del grupo y nombre del turno en el que se encuentran?

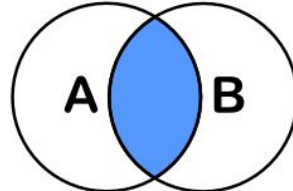
Realizando una SELECT con una tabla no podría acceder a toda esa información, por lo que se hace necesario añadir más de una tabla a la consulta en la cláusula FROM.

En función de cómo queramos presentar la información, se pueden realizar diferentes tipos de consultas entre varias tablas.

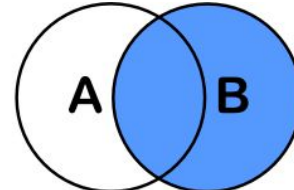
# 1.4 Formas de seleccionar datos de dos tablas



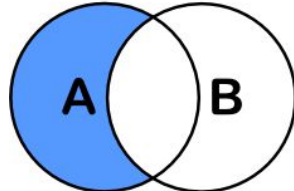
SELECT <auswahl>  
FROM tabelleA A  
LEFT JOIN tabelleB B  
ON A.key = B.key



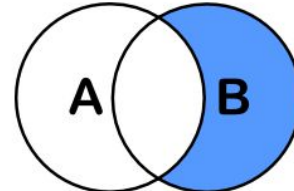
SELECT <auswahl>  
FROM tabelleA A  
INNER JOIN tabelleB B  
ON A.key = B.key



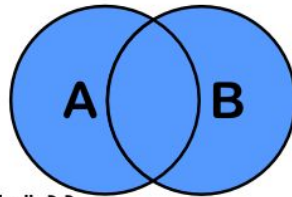
SELECT <auswahl>  
FROM tabelleA A  
RIGHT JOIN tabelleB B  
ON A.key = B.key



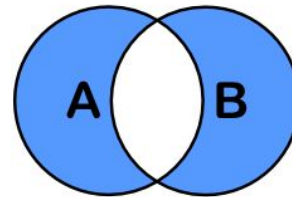
SELECT <auswahl>  
FROM tabelleA A  
LEFT JOIN tabelleB B  
ON A.key = B.key  
WHERE B.key IS NULL



SELECT <auswahl>  
FROM tabelleA A  
RIGHT JOIN tabelleB B  
ON A.key = B.key  
WHERE A.key IS NULL



SELECT <auswahl>  
FROM tabelleA A  
FULL OUTER JOIN tabelleB B  
ON A.key = B.key



SELECT <auswahl>  
FROM tabelleA A  
FULL OUTER JOIN tabelleB B  
ON A.key = B.key  
WHERE A.key IS NULL  
OR B.key IS NULL