

# Consultas en SQL

Grupo 8

06/07/2024

## Inserción, modificación y borrado de datos

Una vez cargadas las tablas con datos, realizar a criterio del grupo:

1. Cinco consultas distintas para inserción de datos (distintos a los ya cargados en forma masiva).
2. Cinco consultas distintas para modificación de datos.
3. Cinco consultas SQL distintas para borrado de filas.

Las consultas pueden ser ejecutadas sobre una misma tabla o distintas, pero todas deben cumplir condiciones medianamente complejas para ejecutarse (por ej. buscar valores en otra tabla relacionada mediante una subconsulta).

### Inserción de datos

1. 

```
INSERT INTO Profesores(dni, tipo_domicilio, fecha_nacimiento, nombre, apellido,
↪ localidad, departamento, provincia, numero, piso, codigo_postal, calle,
↪ edad, cuil, inicio_actividad, estado_civil, sexo, barrio, legajo)
VALUES
↪ (23123125,'Particular','1987-02-12','Julio','Cortázar','París','Lingüística',
'Montmartre',1029,'1',3500,'Bordeaux',70,20231231253,'2020-03-12','Soltero',
↪ 'M','Parisiennes',45564);
```
2. 

```
INSERT INTO Familiar(apellido, parentesco, tipo_documento, nro_documento,
↪ nombre, fecha_nacimiento, idos, dni, idsv, porcentaje, domicilio)
VALUES ('Pizarnik', 'Hermana', 'DNI', 19555009, 'Alejandra', '1936-05-29', NULL,
↪ 23123125, NULL, 64.00, '1 Rue Victor Cousin 75005 Paris Francia');
```
3. 

```
INSERT INTO Datos_de_Contacto(email, tipo_email, telefono, tipo_telefono,
↪ id_contacto, dni)
SELECT 'jcort@proton.me', 'Personal', 3777892312, 'Personal', 1233982, p.dni
FROM Profesores AS p
WHERE p.nombre LIKE '%Julio%' AND p.apellido LIKE '%Cortázar%';
```
4. 

```
INSERT INTO Antecedentes_Profesionales(empresa_anterior, tipo_actividad, cargo,
↪ desde, hasta, dni)
SELECT 'Editorial Planeta', 'Escritor', 'Editor', '1989-11-09', NULL, p.dni
FROM Profesores AS p
WHERE p.dni = 23123125;
```

```

5.  INSERT INTO Participacion_Reuniones_Cientificas(fecha, participacion, titulo,
    ↪  dni)
    SELECT '2004-07-14', 'Orador', 'Lingüísitca según Chomsky', p.dni
    FROM Profesores AS p
    WHERE p.dni = 23123125;

```

## Modificación de datos

```

1.  UPDATE Profesores
    SET legajo = legajo + 10000000
    WHERE codigo_postal >= 3500;

2.  UPDATE Datos_de_Contacto
    SET id_contacto = id_contacto + 12034
    WHERE dni > 50000000 AND tipo_email LIKE 'Trabajo';

3.  UPDATE Datos_de_Contacto
    SET tipo_email = 'Trabajo'
    WHERE dni = 23123125 AND tipo_email LIKE 'Personal';

4.  UPDATE Dependencia
    SET observacion = 'Observación actualizada'
    WHERE idos IN (
        SELECT idos
        FROM Obra_Social
        WHERE dni = 23154067
    );

5.  UPDATE Profesores
    SET departamento = 'Literatura', codigo_postal = 3450
    WHERE dni = 23123125;

6.  UPDATE Profesores
    SET inicio_actividad = '2020-01-01'
    WHERE dni IN (
        SELECT dni
        FROM Datos_Trabajo
        WHERE presta_servicio_utn = 'True'
    );

7.  UPDATE Titulos AS t
    INNER JOIN Profesores AS p ON t.dni = p.dni
    SET t.institucion = 'Pierre-Auguste Renoir'
    WHERE p.fecha_nacimiento < '1970-01-01';

```

## Borrado de datos

```

1.  DELETE FROM Cursos_y_Conferencias

```

```
WHERE institucion LIKE 'Familia Guerra S.Coop.' AND desde >= '2005-01-01' AND
↳ hasta <= '2021-12-31';
```

2. 

```
DELETE FROM Antecedentes_Profesionales
WHERE dni IN (
    SELECT dni
    FROM Profesores
    WHERE edad > 60
)
AND empresa_anterior LIKE 'Editorial Planeta';
```

3. 

```
DELETE FROM Publicaciones
WHERE dni IN (SELECT dni FROM Profesores WHERE nombre LIKE '%Obdulia%')
AND titulo IN (SELECT titulo FROM Publicaciones WHERE autores LIKE '%PANCHO%');
```

4. 

```
DELETE FROM Familiar
WHERE domicilio LIKE '%Pasaje%' AND porcentaje < 60.00;
```

5. 

```
DELETE FROM Profesores
WHERE departamento LIKE '%Asturias%'
AND dni NOT IN (SELECT dni FROM Obra_Social)
AND dni NOT IN (SELECT dni FROM Seguro_de_Vida);
```

## Consultas SELECT

1. Listado de docentes que viven en una provincia distinta de aquella en la que trabajan.

```
SELECT *
FROM Profesores p INNER JOIN Declaracion_Jurada dj ON p.dni = dj.dni INNER JOIN
↳ Datos_de_Cargo dc ON dc.iddj = dj.iddj
WHERE p.provincia <> dc.provincia;
```

2. Listado de docentes que poseen títulos de posgrado y no realizan tareas de investigación.

```
SELECT *
FROM Profesores p
WHERE p.dni IN (
    SELECT DISTINCT t.dni
    FROM Titulos t
    WHERE t.nivel LIKE 'Maestría' OR t.nivel LIKE 'Doctorado'
)
AND p.dni NOT IN (
    SELECT DISTINCT a.dni
    FROM Actividad_e_Investigacion a
);
```

3. Informar promedio de edad de los docentes que poseen más de 10 años de antecedentes como docentes.

```
SELECT AVG(p.edad) AS promedio_edad
FROM Profesores p
WHERE p.dni IN (
    SELECT ap.dni
```

```

FROM Antecedentes_Docentes ap
WHERE DATEDIFF(CURDATE(), ap.desde) >= 3650 -- 10 años en días
      ↳ aproximadamente
);

```

4. Listar DNI y nombre de los docentes que presentaron más de un cargo docente en las declaraciones juradas de los últimos 3 años.

```

SELECT p.dni, p.nombre, p.apellido
FROM Profesores p
INNER JOIN Declaracion_Jurada dj ON p.dni = dj.dni
INNER JOIN Datos_de_Cargo ddc ON dj.iddj = ddc.iddj
WHERE YEAR(dj.fecha_dj) >= YEAR(CURDATE()) - 3
GROUP BY p.dni, p.nombre, p.apellido
HAVING COUNT(DISTINCT ddc.cargo) > 1;

```

5. Listado de docentes cuya carga horaria superan las 20 horas semanales, en función de la última declaración jurada presentada.

```

SELECT p.dni, p.nombre, p.apellido, SUM(ch.horas_clase) AS total_horas_semanales
FROM Profesores p INNER JOIN Declaracion_Jurada dj ON p.dni = dj.dni INNER JOIN
      ↳ Carga_Horaria ch ON dj.iddj = ch.iddj
WHERE dj.fecha_dj = (
      SELECT MAX(dj2.fecha_dj)
      FROM Declaracion_Jurada dj2
      WHERE dj2.dni = p.dni
      )
GROUP BY p.dni, p.nombre, p.apellido
HAVING SUM(ch.horas_clase) > 20;

```

6. Apellido y nombre de aquellos docentes que poseen la máxima cantidad de cargos docentes actualmente. (La cantidad de cargos surge de sumar todos los cargos docentes que se ejercen -suma de cargos docentes de la última declaración jurada-. Una vez que se sabe la cantidad de cargos por docente se puede averiguar cuál es la máxima cantidad y seguidamente los docentes que tienen esa máxima cantidad). No nos interesan las horas.

```

WITH Ultima_Declaracion AS (
      SELECT dj.dni, MAX(dj.fecha_dj) AS ultima_fecha
      FROM Declaracion_Jurada dj
      GROUP BY dj.dni
),
Cantidad_Cargos AS (
      SELECT p.dni, p.apellido, p.nombre, COUNT(ddc.cargo) AS cantidad_cargos
      FROM Profesores p INNER JOIN Declaracion_Jurada dj ON p.dni = dj.dni INNER
      ↳ JOIN Datos_de_Cargo ddc ON dj.iddj = ddc.iddj INNER JOIN
      ↳ Ultima_Declaracion ud ON dj.dni = ud.dni AND dj.fecha_dj =
      ↳ ud.ultima_fecha
      GROUP BY p.dni, p.apellido, p.nombre
),
Maxima_Cantidad_Cargos AS (
      SELECT MAX(cantidad_cargos) AS maxima_cantidad
      FROM Cantidad_Cargos
)
SELECT cc.apellido, cc.nombre
FROM Cantidad_Cargos cc INNER JOIN Maxima_Cantidad_Cargos mcc ON
      ↳ cc.cantidad_cargos = mcc.maxima_cantidad;

```

7. Listado de docentes solteros/as (sin esposa/o e/o hijos a cargo en la obra social).

```
-- Opción 1
SELECT *
FROM Profesores p
WHERE p.dni NOT IN (
    SELECT DISTINCT f.dni
    FROM Familiar f
    WHERE f.parentesco IN ('Esposa', 'Esposo', 'Hijo', 'Hija')
);

-- En nuestra tabla tenemos el campo "Estado Civil". Podemos resolver la
-- consulta consultando el mismo.

-- Opción 2
SELECT *
FROM Profesores p
WHERE p.estado_civil = 'Soltero';
```

8. Cantidad de docentes cuyos hijos a cargo son todos menores de 10 años.

```
SELECT COUNT(DISTINCT p.dni) AS cantidad_de_docentes
FROM Profesores p
WHERE p.dni IN (
    SELECT DISTINCT f.dni
    FROM Familiar f
    WHERE YEAR(f.fecha_nacimiento) > YEAR(CURDATE()) - 10 AND f.parentesco LIKE
        ↳ 'Hij%'
);
```

9. Informar aquellos docentes que posean alguna persona del grupo familiar a cargo en la obra social que no es beneficiario del seguro de vida obligatorio.

```
SELECT p.nombre, p.apellido
FROM Profesores p
WHERE p.dni IN (
    SELECT DISTINCT f.dni
    FROM Familiar f
    WHERE f.dni NOT IN (
        SELECT DISTINCT sv.dni
        FROM Seguro_de_Vida sv
    )
);
```

10. Informar Cantidad de individuos asegurados por provincia.

```
SELECT p.provincia, COUNT(*) AS cantidad_asegurados
FROM Profesores p INNER JOIN Seguro_de_Vida sv ON p.dni = sv.dni
GROUP BY p.provincia;
```