

# SQL Exercise

Camilo Vega

## Contenido

|                     |          |
|---------------------|----------|
| <b>Punto A</b>      | <b>1</b> |
| Query . . . . .     | 1        |
| Resultado . . . . . | 2        |
| <b>Punto B</b>      | <b>3</b> |
| Query . . . . .     | 3        |
| Resultado . . . . . | 3        |
| <b>Punto C</b>      | <b>3</b> |
| Query . . . . .     | 3        |
| Resultado . . . . . | 4        |
| <b>Punto E</b>      | <b>5</b> |
| Query . . . . .     | 5        |
| Resultado . . . . . | 5        |

---

## Punto A

Listar todos los cursos ofrecidos en la carrera de Ingeniería de Sistemas. Para este caso se debe tener en cuenta que un mismo curso puede ser dictado por profesores diferentes, en salones diferentes y en horarios diferentes. Se deben obtener los siguientes datos: el nombre del curso, el nombre completo del profesor que dicta el curso, el salón en que se dicta el curso y la hora en la que se dicta del curso.

## Query

**NOTA:** Se utiliza la cláusula **DISTINCT** en el QUERY para obtener valores únicos, dado que en los datos actuales los cursos que se dictan en distintos días tienen el mismo horario. Si en el futuro esta situación cambia se recomienda incluir el campo **DIA** en la consulta, y eliminar la cláusula **DISTINCT**.”

```

SELECT DISTINCT
    CURSOS.NOMBRE AS CURSO,
    USUARIOS.NOMBRE || ' ' || APELLIDO AS PROFESOR,
    SALONES.NOMBRE AS SALON,
    HORA_INICIO || ' a ' || HORA_FIN AS HORARIO
FROM CALENDARIO_CURSOS
JOIN CURSOS
    ON CALENDARIO_CURSOS.ID_CURSO = CURSOS.ID_CURSO
JOIN CURSOS_CARRERAS
    ON CURSOS.ID_CURSO = CURSOS_CARRERAS.ID_CURSO
JOIN CARRERAS
    ON CURSOS_CARRERAS.ID_CARRERA = CARRERAS.ID_CARRERA
JOIN USUARIOS
    ON CALENDARIO_CURSOS.ID_PROFESOR = USUARIOS.ID_USUARIO
JOIN SALONES
    ON CALENDARIO_CURSOS.ID_SALON = SALONES.ID_SALON
WHERE CARRERAS.NOMBRE = 'Ingeniería de Sistemas'
ORDER BY CURSOS.NOMBRE;

```

## Resultado

| CURSO                                   | PROFESOR       | SALON | HORARIO             |
|---|----------------|-------|---------------------|
| Aprendizaje Automático                  | Robert Vance   | 1.1   | 04:00 PM a 06:00 PM |
| Arquitectura del Computador             | Grace Stone    | 2.6   | 04:00 PM a 06:00 PM |
| Cinemática y Dinámica                   | Saanvi Bahl    | 2.2   | 02:00 PM a 04:00 PM |
| Cálculo Diferencial                     | Grace Stone    | 2.5   | 08:00 AM a 10:00 AM |
| Cálculo Integral                        | Jared Vásquez  | 3.3   | 10:00 AM a 12:00 PM |
| Cálculo Multivariado                    | Jared Vásquez  | 2.1   | 02:00 PM a 04:00 PM |
| Desarrollo de Videuegos                 | Ben Stone      | 3.2   | 02:00 PM a 05:00 PM |
| Electricidad y Magnetismo               | Saanvi Bahl    | 1.5   | 04:00 PM a 06:00 PM |
| Estructuras de Datos                    | Ben Stone      | 3.4   | 10:00 AM a 12:00 PM |
| Introducción a la Programación          | Ben Stone      | 1.4   | 08:00 AM a 10:00 AM |
| Introducción a la Seguridad Informática | Robert Vance   | 3.5   | 04:00 PM a 06:00 PM |
| Lógica Digital y Lenguaje de Máquina    | Grace Stone    | 1.2   | 02:00 PM a 04:00 PM |
| Optimización Matemática                 | Jared Vásquez  | 2.2   | 02:00 PM a 04:00 PM |
| Optimización Matemática                 | Ulrich Nielsen | 1.3   | 08:00 AM a 10:00 AM |
| Programación Paralela                   | Ulrich Nielsen | 3.3   | 08:00 AM a 10:00 AM |
| Práctica Estudiantil                    | Jared Vásquez  | 1.6   | 02:00 PM a 04:00 PM |
| Sistemas Inteligentes                   | Robert Vance   | 2.2   | 10:00 AM a 12:00 PM |
| Teología                                | Zeke Landon    | 1.8   | 10:00 AM a 12:00 PM |
| Técnicas y Prácticas de Programación    | Ben Stone      | 3.4   | 04:00 PM a 06:00 PM |
| Álgebra Lineal                          | Grace Stone    | 3.1   | 10:00 AM a 12:00 PM |
| Ética                                   | Mikaela Stone  | 3.1   | 08:00 AM a 10:00 AM |
| Ética                                   | Jared Vásquez  | 3.7   | 08:00 AM a 10:00 AM |

## Punto B

Obtener la lista de profesores que dictan cursos pertenecientes a la facultad de Humanidades. Se deben obtener los siguientes datos: el nombre completo del profesor y el nombre del curso que dicta.

### Query

```
SELECT DISTINCT
    USUARIOS.NOMBRE || ' ' || APELLIDO AS PROFESOR,
    CURSOS.NOMBRE AS CURSO
FROM CALENDARIO_CURSOS
JOIN CURSOS
    ON CALENDARIO_CURSOS.ID_CURSO = CURSOS.ID_CURSO
JOIN FACULTADES
    ON CURSOS.ID_FACULTAD = FACULTADES.ID_FACULTAD
JOIN USUARIOS
    ON CALENDARIO_CURSOS.ID_PROFESOR = USUARIOS.ID_USUARIO
WHERE FACULTADES.NOMBRE = 'Humanidades y Ciencias Sociales'
ORDER BY PROFESOR;
```

### Resultado

| PROFESOR      | CURSO                   |
|---------------|-------------------------|
| Jared Vásquez | Ética                   |
| Mikaela Stone | Ética                   |
| Zeke Landon   | Teología                |
| Zeke Landon   | Filosofía Antigua       |
| Zeke Landon   | Antropología Filosófica |
| Zeke Landon   | Filosofía de la Ciencia |

## Punto C

Obtener la lista de profesores que dictan cursos en **dos** carreras diferentes. Se deben obtener los siguientes datos: el nombre completo del profesor, el nombre del curso que dicta y el nombre de la carrera a la que pertenece el curso. Pista: averigua cómo funciona la sentencia **WITH**.

### Query

```
WITH p2c AS(
    SELECT DISTINCT
        USUARIOS.ID_USUARIO,
```

```

    USUARIOS.NOMBRE,
    APELLIDO
FROM CALENDARIO_CURSOS
JOIN CURSOS
    ON CALENDARIO_CURSOS.ID_CURSO = CURSOS.ID_CURSO
JOIN CURSOS_CARRERAS
    ON CALENDARIO_CURSOS.ID_CURSO = CURSOS_CARRERAS.ID_CURSO
JOIN CARRERAS
    ON CURSOS_CARRERAS.ID_CARRERA = CARRERAS.ID_CARRERA
JOIN USUARIOS
    ON CALENDARIO_CURSOS.ID_PROFESOR = USUARIOS.ID_USUARIO
GROUP BY USUARIOS.ID_USUARIO, USUARIOS.NOMBRE, APELLIDO
    HAVING COUNT(DISTINCT CARRERAS.ID_CARRERA) = 2
)
SELECT DISTINCT
    p2c.NOMBRE || ' ' || p2c.APELLIDO AS PROFESOR,
    CURSOS.NOMBRE AS CURSO,
    CARRERAS.NOMBRE AS CARRERA
FROM p2c
JOIN CALENDARIO_CURSOS
    ON p2c.ID_USUARIO = CALENDARIO_CURSOS.ID_PROFESOR
JOIN CURSOS
    ON CALENDARIO_CURSOS.ID_CURSO = CURSOS.ID_CURSO
JOIN CURSOS_CARRERAS
    ON CURSOS.ID_CURSO = CURSOS_CARRERAS.ID_CURSO
JOIN CARRERAS
    ON CURSOS_CARRERAS.ID_CARRERA = CARRERAS.ID_CARRERA
ORDER by PROFESOR, CARRERA;

```

## Resultado

| PROFESOR          | CURSO                                | CARRERA                |
|-------------------|--------------------------------------|------------------------|
| Ben Stone         | Introducción a la Programación       | Ingeniería de Sistemas |
| Ben Stone         | Estructuras de Datos                 | Ingeniería de Sistemas |
| Ben Stone         | Técnicas y Prácticas de Programación | Ingeniería de Sistemas |
| Ben Stone         | Desarrollo de Videouegos             | Ingeniería de Sistemas |
| Ben Stone         | Estructuras de Datos                 | Matemáticas Aplicadas  |
| Claudia Tiedemann | Química y Ciencia de Materiales      | Ingeniería Mecánica    |
| Claudia Tiedemann | Propiedades de los Materiales        | Ingeniería Mecánica    |
| Claudia Tiedemann | Diseño Mecánico                      | Ingeniería Mecánica    |
| Claudia Tiedemann | Máquinas Térmicas e Hidráulicas      | Ingeniería Mecánica    |
| Claudia Tiedemann | Álgebra Moderna                      | Matemáticas Aplicadas  |
| Ulrich Nielsen    | Programación Paralela                | Ingeniería de Sistemas |
| Ulrich Nielsen    | Optimización Matemática              | Ingeniería de Sistemas |
| Ulrich Nielsen    | Optimización Matemática              | Matemáticas Aplicadas  |

## Punto E

Listar los estudiantes que se han inscrito a un curso determinado. Se deben obtener los siguientes datos: el nombre completo del estudiante y el nombre del curso.

### Query

**NOTA:** Es posible sustituir el nombre del curso deseado en la cláusula **WHERE** para encontrar a los estudiantes que se hayan matriculado en él.

```
SELECT DISTINCT
    USUARIOS.NOMBRE || ' ' || APELLIDO AS ESTUDIANTE,
    CURSOS.NOMBRE AS CURSO
FROM CURSOS_ESTUDIANTES
JOIN CALENDARIO_CURSOS
    ON CURSOS_ESTUDIANTES.ID_CALENDARIO = CALENDARIO_CURSOS.ID_CALENDARIO
JOIN CURSOS
    ON CALENDARIO_CURSOS.ID_CURSO = CURSOS.ID_CURSO
JOIN USUARIOS
    ON CURSOS_ESTUDIANTES.ID_USUARIO = USUARIOS.ID_USUARIO
WHERE CURSOS.NOMBRE = 'Cálculo Diferencial'
ORDER BY ESTUDIANTE;
```

### Resultado

| ESTUDIANTE        | CURSO               |
|-------------------|---------------------|
| Bartosz Tiedemann | Cálculo Diferencial |
| Franziska Doppler | Cálculo Diferencial |
| Jonas Kahnwald    | Cálculo Diferencial |
| Mads Nielsen      | Cálculo Diferencial |
| Magnus Nielsen    | Cálculo Diferencial |
| Martha Nielsen    | Cálculo Diferencial |
| Olive Stone       | Cálculo Diferencial |