

Actividad 2 Base de datos Oracle

Gestión de datos, Maestría en ciencia de datos

Universidad Javeriana de Cali

Equipo 2:

Carlos Piñeros Castro

Juan José Restrepo Rosero

Joel Doria Atencia

Contexto

Imaginen que determinada Universidad desarrolló una plataforma educativa que contiene un gran número de funcionalidades, entre las que se encuentra un módulo de registro académico. Gracias a este módulo la Universidad ha podido almacenar gran cantidad de datos relacionados con carreras, cursos, profesores y estudiantes. Actualmente los directivos se encuentran interesados en llevar a cabo procesos de análisis que les permita visualizar ciertas tendencias y cuestiones académicas, por lo que requieren del conocimiento de un científico de datos para realizar esta labor.

Consultas

Con el objetivo de seleccionar la información pertinente para los procesos de análisis que requiere la Universidad los directivos nos piden efectuar las siguientes consultas en SQL sobre los datos del módulo de registro académico:

- 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.

Sentencia SQL:

```
SELECT
  ca.nombre AS nombre_carrera,
  cc.semestre,
  cal.ID_CURSO,
  c.nombre AS nombre_curso,
  cal.dia AS dia,
  cal.hora_inicio,
  cal.hora_fin,
  u.nombre || ' ' || u.apellido AS nombre_completo_profesor,
  s.nombre AS salon
```

```

FROM
  Cursos c
INNER JOIN Cursos_Carreras cc ON c.id_curso = cc.id_curso
INNER JOIN Calendario_Cursos cal ON cc.id_curso = cal.id_curso
INNER JOIN Salones s ON cal.id_salon = s.id_salon
INNER JOIN Usuarios u ON cal.id_profesor = u.id_usuario
INNER JOIN Carreras ca ON cc.ID_CARRERA = ca.ID_CARRERA
WHERE
  -- cc.id_carrera = (SELECT cc.ID_CARRERA FROM Carreras WHERE nombre = 'Ingenieria de Sistemas')
  cc.id_carrera = 1
ORDER BY
  ca.nombre, c.nombre, cal.hora_inicio;

```

RESULTADO:

NOMBRE_CARRERA	SEMESTRE	ID_CURSO	NOMBRE_CURSO	DIA	HORA_INICIO	HORA_FIN	NOMBRE_COMPLETO_PROFESOR	SALON
Ingeniería de Sistemas	7	20	Aprendizaje Automático	Martes	4:00 p. m.	6:00 p. m.	Robert Vance	1.1
Ingeniería de Sistemas	7	20	Aprendizaje Automático	Jueves	4:00 p. m.	6:00 p. m.	Robert Vance	1.1
Ingeniería de Sistemas	4	10	Arquitectura del Computador	Lunes	4:00 p. m.	6:00 p. m.	Grace Stone	2.6
Ingeniería de Sistemas	4	10	Arquitectura del Computador	Viernes	4:00 p. m.	6:00 p. m.	Grace Stone	2.6
Ingeniería de Sistemas	4	28	Cinemática y Dinámica	Miércoles	2:00 p. m.	4:00 p. m.	Saanvi Bahl	2.2
Ingeniería de Sistemas	4	28	Cinemática y Dinámica	Lunes	2:00 p. m.	4:00 p. m.	Saanvi Bahl	2.2
Ingeniería de Sistemas	4	28	Cinemática y Dinámica	Viernes	2:00 p. m.	4:00 p. m.	Saanvi Bahl	2.2
Ingeniería de Sistemas	1	44	Cálculo Diferencial	Jueves	8:00 a. m.	10:00 a. m.	Grace Stone	2.5
Ingeniería de Sistemas	1	44	Cálculo Diferencial	Martes	8:00 a. m.	10:00 a. m.	Grace Stone	2.5
Ingeniería de Sistemas	2	47	Cálculo Integral	Martes	10:00 a. m.	12:00 p. m.	Jared Viñesquez	3.3
Ingeniería de Sistemas	2	47	Cálculo Integral	Jueves	10:00 a. m.	12:00 p. m.	Jared Viñesquez	3.3
Ingeniería de Sistemas	3	48	Cálculo Multivariado	Jueves	2:00 p. m.	4:00 p. m.	Jared Viñesquez	2.1
Ingeniería de Sistemas	3	48	Cálculo Multivariado	Martes	2:00 p. m.	4:00 p. m.	Jared Viñesquez	2.1
Ingeniería de Sistemas	9	26	Desarrollo de Videuegos	Martes	2:00 p. m.	5:00 p. m.	Ben Stone	3.2
Ingeniería de Sistemas	9	26	Desarrollo de Videuegos	Viernes	2:00 p. m.	5:00 p. m.	Ben Stone	3.2
Ingeniería de Sistemas	5	34	Electricidad y Magnetismo	Miércoles	4:00 p. m.	6:00 p. m.	Saanvi Bahl	1.5
Ingeniería de Sistemas	5	34	Electricidad y Magnetismo	Lunes	4:00 p. m.	6:00 p. m.	Saanvi Bahl	1.5
Ingeniería de Sistemas	5	34	Electricidad y Magnetismo	Viernes	4:00 p. m.	6:00 p. m.	Saanvi Bahl	1.5
Ingeniería de Sistemas	2	3	Estructuras de Datos	Jueves	10:00 a. m.	12:00 p. m.	Ben Stone	3.4
Ingeniería de Sistemas	2	3	Estructuras de Datos	Martes	10:00 a. m.	12:00 p. m.	Ben Stone	3.4
Ingeniería de Sistemas	1	1	Introducción a la Programación	Martes	8:00 a. m.	10:00 a. m.	Ben Stone	1.4

Ingeniería de Sistemas	1	1	Introducción a la Programación	Jueves	8:00 a. m.	10:00 a. m.	Ben Stone	1.4
Ingeniería de Sistemas	8	22	Introducción a la Seguridad Informática	Miércoles	4:00 p. m.	6:00 p. m.	Robert Vance	3.5
Ingeniería de Sistemas	8	22	Introducción a la Seguridad Informática	Viernes	4:00 p. m.	6:00 p. m.	Robert Vance	3.5
Ingeniería de Sistemas	3	6	Lógica Digital y Lenguaje de Máquina	Viernes	2:00 p. m.	4:00 p. m.	Grace Stone	1.2
Ingeniería de Sistemas	3	6	Lógica Digital y Lenguaje de Máquina	Miércoles	2:00 p. m.	4:00 p. m.	Grace Stone	1.2
Ingeniería de Sistemas	1	53	Optimización Matemática	Miércoles	2:00 p. m.	4:00 p. m.	Jared Viñesquez	2.2
Ingeniería de Sistemas	1	53	Optimización Matemática	Lunes	2:00 p. m.	4:00 p. m.	Jared Viñesquez	2.2
Ingeniería de Sistemas	1	53	Optimización Matemática	Miércoles	8:00 a. m.	10:00 a. m.	Ulrich Nielsen	1.3
Ingeniería de Sistemas	1	53	Optimización Matemática	Viernes	8:00 a. m.	10:00 a. m.	Ulrich Nielsen	1.3
Ingeniería de Sistemas	7	19	Programación Paralela	Jueves	8:00 a. m.	10:00 a. m.	Ulrich Nielsen	3.3
Ingeniería de Sistemas	7	19	Programación Paralela	Martes	8:00 a. m.	10:00 a. m.	Ulrich Nielsen	3.3
Ingeniería de Sistemas	10	71	Práctica Estudiantil	Viernes	2:00 p. m.	4:00 p. m.	Jared Viñesquez	1.6
Ingeniería de Sistemas	8	24	Sistemas Inteligentes	Lunes	10:00 a. m.	12:00 p. m.	Robert Vance	2.2
Ingeniería de Sistemas	8	24	Sistemas Inteligentes	Miércoles	10:00 a. m.	12:00 p. m.	Robert Vance	2.2
Ingeniería de Sistemas	2	57	Teología	Lunes	10:00 a. m.	12:00 p. m.	Zeke Landon	1.8
Ingeniería de Sistemas	2	4	Técnicas y Prácticas de Programación	Miércoles	4:00 p. m.	6:00 p. m.	Ben Stone	3.4
Ingeniería de Sistemas	2	4	Técnicas y Prácticas de Programación	Lunes	4:00 p. m.	6:00 p. m.	Ben Stone	3.4
Ingeniería de Sistemas	2	45	Álgebra Lineal	Jueves	10:00 a. m.	12:00 p. m.	Grace Stone	3.1
Ingeniería de Sistemas	2	45	Álgebra Lineal	Martes	10:00 a. m.	12:00 p. m.	Grace Stone	3.1
Ingeniería de Sistemas	9	67	Ética	Miércoles	8:00 a. m.	10:00 a. m.	Mikaela Stone	3.1
Ingeniería de Sistemas	9	67	Ética	Viernes	8:00 a. m.	10:00 a. m.	Jared Viñesquez	3.7

- 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.

SENTENCIA SQL:

```
SELECT
f.NOMBRE AS Facultad,
c.nombre AS nombre_curso,
u.nombre || ' ' || u.apellido AS nombre_completo_profesor
```

```

FROM
    Usuarios u
INNER JOIN Calendario_Cursos cal ON u.ID_USUARIO = cal.id_profesor
INNER JOIN Cursos c ON cal.id_curso = c.ID_CURSO
INNER JOIN Facultades f ON c.id_facultad = f.id_facultad
WHERE
    f.ID_FACULTAD=2
    -- f.nombre LIKE '%Humanidades%'
ORDER BY
    u.nombre, u.apellido, c.nombre;

```

RESULTADO:

FACULTAD	NOMBRE_CURSO	NOMBRE_COMPLETO_PROFESOR
Humanidades y Ciencias Sociales	Ética	Jared Vásquez
Humanidades y Ciencias Sociales	Ética	Mikaela Stone
Humanidades y Ciencias Sociales	Antropología Filosófica	Zeke Landon
Humanidades y Ciencias Sociales	Filosofía Antigua	Zeke Landon
Humanidades y Ciencias Sociales	Filosofía de la Ciencia	Zeke Landon
Humanidades y Ciencias Sociales	Teología	Zeke Landon

- 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.

```

SENTENCIA SQL:
WITH Profesores_Multiples_Carreras AS (
    SELECT
        u.ID_USUARIO AS id_profesor,
        u.nombre || ' ' || u.apellido AS nombre_completo_profesor
    FROM
        Usuarios u
    INNER JOIN Calendario_Cursos cal ON u.ID_USUARIO = cal.id_profesor
    INNER JOIN Cursos_Carreras cc ON cal.id_curso = cc.id_curso
    GROUP BY
        u.ID_USUARIO, u.nombre, u.apellido
    HAVING
        COUNT(DISTINCT cc.id_carrera) = 2
)
SELECT DISTINCT
    pmc.nombre_completo_profesor,
    c.ID_CURSO AS id_curso,
    c.nombre AS nombre_curso,
    ca.nombre AS nombre_carrera
FROM
    Profesores_Multiples_Carreras pmc
    INNER JOIN Calendario_Cursos cal ON pmc.id_profesor = cal.id_profesor
    INNER JOIN Cursos c ON cal.id_curso = c.id_curso
    INNER JOIN Cursos_Carreras cc ON cal.id_curso = cc.id_curso
    INNER JOIN Carreras ca ON cc.id_carrera = ca.ID_CARRERA
ORDER BY
    pmc.nombre_completo_profesor, id_curso;

```

RESULTADO:

NOMBRE_COMPLETO_PROFESOR	ID_CURSO	NOMBRE_CURSO	NOMBRE_CARRERA
Ben Stone	1	Introducción a la Programación	Ingeniería de Sistemas
Ben Stone	3	Estructuras de Datos	Ingeniería de Sistemas
Ben Stone	3	Estructuras de Datos	Matemáticas Aplicadas
Ben Stone	4	Técnicas y Prácticas de Programación	Ingeniería de Sistemas
Ben Stone	26	Desarrollo de Videuegos	Ingeniería de Sistemas
Claudia Tiedemann	29	Química y Ciencia de Materiales	Ingeniería Mecánica
Claudia Tiedemann	32	Propiedades de los Materiales	Ingeniería Mecánica
Claudia Tiedemann	39	Diseño Mecánico	Ingeniería Mecánica
Claudia Tiedemann	40	Máquinas Térmicas e Hidráulicas	Ingeniería Mecánica
Claudia Tiedemann	49	Álgebra Moderna	Matemáticas Aplicadas
Ulrich Nielsen	19	Programación Paralela	Ingeniería de Sistemas
Ulrich Nielsen	53	Optimización Matemática	Ingeniería de Sistemas
Ulrich Nielsen	53	Optimización Matemática	Matemáticas Aplicadas

- d. Para un estudiante en particular, obtener el listado de cursos que puede matricular. Para este caso particular se debe tener en cuenta que los cursos se dictan en semestres diferentes, pertenecen a carreras diferentes y un estudiante no puede matricular un curso que ya se encuentre matriculado. Se deben obtener los siguientes datos: el nombre del curso, el nombre de la carrera a la que pertenece y el semestre en el que se ubica. Pista: averigua el uso de la sentencia NOT EXISTS.

SENTENCIA SQL:

```
WITH CarrerasEstudiante AS (  
    SELECT id_carrera,id_usuario  
    FROM Carreras_Estudiantes  
    WHERE id_usuario = :id_estudiante  
)  
,  
CursosMatriculados AS (  
    SELECT cal.id_curso  
    FROM Cursos_Estudiantes ce  
    INNER JOIN Calendario_Cursos cal ON ce.ID_CALENDARIO = cal.ID_CALENDARIO  
    WHERE ce.id_usuario = :id_estudiante  
)  
SELECT  
    u.nombre || ' ' || u.apellido AS nombre_estudiante,  
    cc.ID_CURSO,  
    c.nombre AS nombre_curso,  
    ca.nombre AS nombre_carrera,  
    cc.semestre  
FROM  
    usuarios u  
INNER JOIN CarrerasEstudiante ce ON u.id_usuario = ce.id_usuario  
INNER JOIN Carreras ca ON ce.id_carrera = ca.id_carrera  
INNER JOIN Cursos_Carreras cc ON ca.id_carrera = cc.id_carrera  
INNER JOIN Cursos c ON cc.id_curso = c.id_curso
```

WHERE
NOT EXISTS (
SELECT 1 FROM CursosMatriculados cm WHERE c.id_curso = cm.id_curso
)
AND u.id_usuario = :id_estudiante
order by cc.SEMESTRE;

RESULTADO:

ID_ESTUDIANT E	NOMBRE_ESTUDIANT E	ID_CURS O	NOMBRE_CURSO	NOMBRE_CARRERA	SEMESTR E
17	Franziska Doppler	2	Introducción al Modelado de Sistemas	Ingeniería de Sistemas	1
17	Franziska Doppler	53	Optimización Matemática	Ingeniería de Sistemas	1
17	Franziska Doppler	57	Teología	Ingeniería de Sistemas	2
17	Franziska Doppler	7	Programación Funcional	Ingeniería de Sistemas	3
17	Franziska Doppler	5	Programación Orientada a Objetos	Ingeniería de Sistemas	3
17	Franziska Doppler	8	Lógica para Ciencias de la Computación	Ingeniería de Sistemas	4
17	Franziska Doppler	9	Arboles y Grafos	Ingeniería de Sistemas	4
17	Franziska Doppler	11	Diseño de Interfaces Humano- Computador	Ingeniería de Sistemas	4
17	Franziska Doppler	12	Computabilidad y Complejidad	Ingeniería de Sistemas	5
17	Franziska Doppler	34	Electricidad y Magnetismo	Ingeniería de Sistemas	5
17	Franziska Doppler	14	Comunicación de Datos	Ingeniería de Sistemas	5
17	Franziska Doppler	13	Computación Gráfica	Ingeniería de Sistemas	5
17	Franziska Doppler	16	Análisis y Diseño de Algoritmos	Ingeniería de Sistemas	6
17	Franziska Doppler	18	Sistemas Operativos	Ingeniería de Sistemas	6
17	Franziska Doppler	15	Desarrollo Formal de Sistemas	Ingeniería de Sistemas	6
17	Franziska Doppler	17	Gestión y Modelación de Datos	Ingeniería de Sistemas	6
17	Franziska Doppler	21	Animación y Simulación	Ingeniería de Sistemas	7
17	Franziska Doppler	19	Programación Paralela	Ingeniería de Sistemas	7
17	Franziska Doppler	20	Aprendizaje Automático	Ingeniería de Sistemas	7
17	Franziska Doppler	22	Introducción a la Seguridad Informática	Ingeniería de Sistemas	8
17	Franziska Doppler	24	Sistemas Inteligentes	Ingeniería de Sistemas	8
17	Franziska Doppler	23	Construcción de Software y Pruebas	Ingeniería de Sistemas	8
17	Franziska Doppler	26	Desarrollo de Videouuegos	Ingeniería de Sistemas	9
17	Franziska Doppler	70	Trabajo de Grado	Ingeniería de Sistemas	9
17	Franziska Doppler	25	Computación en la Nube	Ingeniería de Sistemas	9

17	Franziska Doppler	67	Ética	Ingeniería de Sistemas	9
17	Franziska Doppler	71	Práctica Estudiantil	Ingeniería de Sistemas	10

- 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.

SENTENCIA SQL:

```
SELECT distinct
cc.id_curso,
c.nombre AS nombre_curso,
u.nombre || ' ' || u.apellido AS nombre_estudiante
FROM
USUARIOS u
INNER JOIN Cursos_Estudiantes ce ON u.id_usuario = ce.id_usuario
INNER JOIN Calendario_Cursos cc ON ce.id_calendario = cc.id_calendario
INNER JOIN Cursos c ON cc.id_curso = c.id_curso
WHERE
c.id_curso= :idcurso
order by nombre_estudiante;
```

RESULTADO:

ID_CURSO	NOMBRE_CURSO	NOMBRE_ESTUDIANTE
57	Teología	Angelina Meyer
57	Teología	Bartosz Tiedemann
57	Teología	Jonas Kahnwald
57	Teología	Magnus Nielsen
57	Teología	Martha Nielsen
57	Teología	Olive Stone

Conclusiones:

Las consultas SQL son una herramienta fundamental en el mundo de las bases de datos relacionales. Su sintaxis, aunque pueda parecer compleja al principio, permite a los usuarios interactuar de manera eficiente y precisa con la información almacenada en bases de datos.

Ventajas de las consultas SQL:

Flexibilidad: Permiten realizar una amplia variedad de operaciones, desde simples selecciones hasta complejas uniones y agrupaciones.

Potencia: Pueden manipular grandes volúmenes de datos de forma rápida y eficiente.

Estandarización: El lenguaje SQL es un estándar ampliamente adoptado, lo que facilita su uso en diferentes sistemas de gestión de bases de datos.

Facilidad de aprendizaje: Aunque tiene su propia sintaxis, SQL es relativamente fácil de aprender y su lógica es intuitiva para quienes están familiarizados con la programación.

A lo largo de esta actividad, se ha demostrado la capacidad de SQL para gestionar bases de datos académicas complejas, proporcionando resultados eficientes y adaptados a las necesidades de análisis de los directivos universitarios. El uso de consultas con múltiples tablas y la implementación de cláusulas como WITH no solo han facilitado la legibilidad del código, sino que también han optimizado la eficiencia de las consultas.

Uso de WITH para simplificación: En las consultas que requieren manipulación de datos complejos, la utilización de subconsultas comunes mediante la sentencia WITH permite estructurar de forma más clara los pasos intermedios. Esto fue especialmente útil en la consulta que buscaba identificar profesores que imparten clases en diferentes carreras, permitiendo agrupar y filtrar los datos de manera eficaz.

Consultas con múltiples JOIN: El uso de combinaciones de tablas (JOIN) es esencial en una base de datos relacional como la de este proyecto. En particular, las consultas que involucraban la relación entre cursos, profesores y carreras dependían de múltiples uniones de tablas para proporcionar resultados útiles y completos, como se observó en el caso de la consulta sobre los cursos de la carrera de **Ingeniería de Sistemas**.

Condiciones avanzadas como NOT EXISTS: Esta cláusula ayudó a filtrar cursos que un estudiante aún no había matriculado, asegurando que los resultados sean precisos y sin duplicados. Esto resultó crítico en consultas donde era necesario eliminar posibilidades que ya habían sido registradas.

En resumen, las consultas SQL no solo son una herramienta potente para la gestión y análisis de datos, sino que también son altamente adaptables a las necesidades del negocio o del análisis académico. A través del uso de cláusulas avanzadas como WITH y NOT EXISTS, además de la capacidad de combinar datos de múltiples tablas mediante JOIN, se puede optimizar el rendimiento de las consultas y proporcionar información de alta calidad a quienes toman decisiones.