



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

Laboratorio de:

Materia: Fundamentos de Bases de Datos

Práctica No.: LABORATORIO PRÁCTICO - TÓPICO 6

Tema: Consultas SQL en BD relacionales

SGBD: Oracle Database

TABLA DE CONTENIDOS

1. [Objetivos](#)
2. [Requisitos Previos](#)
3. [Fundamentos de SELECT](#)
4. [Cláusula WHERE y Operadores](#)
5. [ORDER BY - Ordenamiento](#)
6. [Funciones de Agregación](#)
7. [GROUP BY y HAVING](#)
8. [JOINS - Uniones de Tablas](#)
9. [Subconsultas \(Subqueries\)](#)
10. [Operadores de Conjuntos](#)
11. [Consultas Avanzadas](#)
12. [Optimización de Consultas](#)
13. [Ejercicios Prácticos](#)
14. [Casos de Prueba](#)
15. [Preguntas de Evaluación](#)

OBJETIVOS

Objetivo General

Dominar el lenguaje de consultas SQL para extraer, analizar y presentar información de bases de datos relacionales, utilizando consultas simples y



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

complejas con múltiples tablas.

Objetivos Específicos

1. **Consultar datos** usando SELECT con todas sus cláusulas
2. **Filtrar información** con WHERE y operadores lógicos
3. **Ordenar resultados** con ORDER BY
4. **Agregar datos** con funciones COUNT, SUM, AVG, MAX, MIN
5. **Agrupar información** con GROUP BY y HAVING
6. **Relacionar tablas** usando INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL OUTER JOIN
7. **Crear subconsultas** en SELECT, WHERE y FROM
8. **Usar operadores de conjuntos** UNION, INTERSECT, MINUS
9. **Optimizar consultas** para mejor rendimiento
10. **Resolver problemas complejos** con consultas SQL

REQUISITOS PREVIOS

Conocimientos Necesarios

- Laboratorios 4 (DDL) y 5 (DML) completados
- Datos cargados en las tablas del sistema académico
- Comprensión del modelo relacional
- Álgebra relacional básica

Verificación de Datos

-- Conectarse al usuario

CONN gestion_academica/EPN2024Secure;

-- Verificar que hay datos en las tablas

SELECT 'CARRERA' AS tabla, COUNT(*) AS registros FROM Carrera
UNION ALL



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
SELECT 'ESTUDIANTE', COUNT(*) FROM Estudiante
UNION ALL
SELECT 'ASIGNATURA', COUNT(*) FROM Asignatura
UNION ALL
SELECT 'DOCENTE', COUNT(*) FROM Docente
UNION ALL
SELECT 'MATRICULA', COUNT(*) FROM Matricula
UNION ALL
SELECT 'PRERREQUISITO', COUNT(*) FROM Prerrequisito;
```

-- Debería mostrar datos en todas las tablas

-- Si alguna tabla está vacía, ejecutar scripts de INSERT del Lab 5

Preparar Datos de Prueba Adicionales

-- Si necesita más datos para práctica, ejecutar:

-- Más estudiantes

```
INSERT ALL INTO Estudiante VALUES ('1750700001', 'Andrea', 'Alvarez Brito', 'andrea.alvarez@epn.edu.ec', '0991111111', TO_DATE('2003-01-10', 'YYYY-MM-DD'), 'F', 'ING-SIS', 'ACTIVO', 15, SYSDATE)
INSERT INTO Estudiante VALUES ('1750700002', 'Bryan', 'Bravo Cárdenas', 'bryan.bravo@epn.edu.ec', '0992222222', TO_DATE('2002-05-22', 'YYYY-MM-DD'), 'M', 'ING-SIS', 'ACTIVO', 20, SYSDATE)
INSERT INTO Estudiante VALUES ('1750700003', 'Carla', 'Cruz Delgado', 'carla.cruz@epn.edu.ec', '0993333333', TO_DATE('2003-08-15', 'YYYY-MM-DD'), 'F', 'ING-CIV', 'ACTIVO', 10, SYSDATE)
INSERT INTO Estudiante VALUES ('1750700004', 'David', 'Diaz Espinoza', 'david.diaz@epn.edu.ec', '0994444444', TO_DATE('2002-11-30', 'YYYY-MM-DD'), 'M', 'ING-ELE', 'ACTIVO', 25, SYSDATE)
INSERT INTO Estudiante VALUES ('1750700005', 'Elena', 'Estrella Flores', 'elena.estrella@epn.edu.ec', '0995555555', TO_DATE('2003-03-18', 'YYYY-MM-DD'), 'F', 'ING-SIS', 'ACTIVO', 30, SYSDATE)
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
INTO Estudiante VALUES ('1750700006', 'Fernando', 'Fuentes García', 'fernando.fuentes@epn.edu.ec', '0996666666', TO_DATE('2002-07-25', 'YYYY-MM-DD'), 'M', 'ING-SIS', 'INACTIVO', 5, SYSDATE)
INTO Estudiante VALUES ('1750700007', 'Gabriela', 'González Herrera', 'gabriela.gonzalez@epn.edu.ec', '0997777777', TO_DATE('2003-12-05', 'YYYY-MM-DD'), 'F', 'ING-CIV', 'ACTIVO', 35, SYSDATE)
INTO Estudiante VALUES ('1750700008', 'Hugo', 'Hidalgo Ibarra', 'hugo.hidalgo@epn.edu.ec', '0998888888', TO_DATE('2002-04-14', 'YYYY-MM-DD'), 'M', 'ING-ELE', 'GRADUADO', 180, SYSDATE) SELECT * FROM DUAL;
```

-- Más matrículas con calificaciones

```
INSERT ALL
INTO Matricula (cedula_estudiante, codigo_asignatura, cedula_docente, periodo, paralelo, nota_parcial1, nota_parcial2, nota_final, estado) VALUES ('1750700001', 'BD-501', '1712345678', '2024-2S', 'A', 8.5, 9.0, 8.75, 'APROBADO')
INTO Matricula (cedula_estudiante, codigo_asignatura, cedula_docente, periodo, paralelo, nota_parcial1, nota_parcial2, nota_final, estado) VALUES ('1750700002', 'BD-501', '1712345678', '2024-2S', 'A', 7.0, 7.5, 7.25, 'APROBADO')
INTO Matricula (cedula_estudiante, codigo_asignatura, cedula_docente, periodo, paralelo, nota_parcial1, nota_parcial2, nota_final, estado) VALUES ('1750700005', 'BD-501', '1712345678', '2024-2S', 'A', 9.5, 9.0, 9.25, 'APROBADO')
INTO Matricula (cedula_estudiante, codigo_asignatura, cedula_docente, periodo, paralelo, nota_parcial1, nota_parcial2, nota_final, estado) VALUES ('1750700001', 'PRG-201', '1734567890', '2024-2S', 'B', 6.5, 6.0, 6.25, 'REPROBADO')
INTO Matricula (cedula_estudiante, codigo_asignatura, cedula_docente, periodo, paralelo, nota_parcial1, nota_parcial2, nota_final, estado) VALUES ('1750700002', 'PRG-201', '1734567890', '2024-2S', 'B', 8.0, 8.5, 8.25, 'APROBADO')
SELECT * FROM DUAL;
```

COMMIT;

-- Verificar

```
SELECT COUNT(*) AS total_estudiantes FROM Estudiante;
SELECT COUNT(*) AS total_matriculas FROM Matricula;
```

FUNDAMENTOS DE SELECT

Sintaxis Básica



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
SELECT columna1, columna2, columna3, ...  
FROM nombre_tabla;
```

-- *O para todas las columnas*

```
SELECT * FROM nombre_tabla;
```

Ejemplo 1: SELECT Simple

-- *EJEMPLO 1: SELECT básico - todas las columnas*

-- =====

-- *Seleccionar todas las carreras*

```
SELECT * FROM Carrera;
```

-- *Seleccionar columnas específicas*

```
SELECT codigo, nombre, duracion_semestres  
FROM Carrera;
```

-- *Usar alias para columnas*

```
SELECT  
    codigo AS "Código",  
    nombre AS "Nombre de Carrera",  
    duracion_semestres AS "Duración (semestres)",  
    creditos_totales AS "Créditos"  
FROM Carrera;
```

Ejemplo 2: DISTINCT



**ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS**

-- EJEMPLO 2: Eliminar duplicados con *DISTINCT*

-- =====

-- Ver todos los estados de estudiantes (con duplicados)

SELECT estado **FROM** Estudiante;

-- Ver estados únicos

SELECT DISTINCT estado **FROM** Estudiante;

-- Combinaciones únicas de carrera y estado

SELECT DISTINCT codigo_carrera, estado

FROM Estudiante

ORDER BY codigo_carrera, estado;

-- Contar valores únicos

SELECT COUNT(**DISTINCT** codigo_carrera) **AS** total_carreras_con_estudiantes

FROM Estudiante;

Ejemplo 3: Expresiones y Cálculos

-- EJEMPLO 3: Cálculos en *SELECT*

-- =====

-- Calcular horas totales de cada asignatura



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
SELECT
    codigo,
    nombre,
    horas_teoría,
    horas_práctica,
    (horas_teoría + horas_práctica) AS horas_totales
FROM Asignatura;
```

-- Concatenar cadenas

```
SELECT
    cedula,
    nombres || ' ' || apellidos AS nombre_completo,
    email
FROM Estudiante;
```

-- Cálculos con NULL

```
SELECT
    codigo,
    nombre,
    créditos,
    créditos * 16 AS horas_estimadas_estudio -- 16 horas por crédito
FROM Asignatura;
```

-- Formato de números

```
SELECT
    nombres,
    apellidos,
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
    creditos_aprobados,  
    ROUND(creditos_aprobados / 180 * 100, 2) AS porcentaje_avance  
FROM Estudiante  
WHERE creditos_aprobados > 0;
```

Ejemplo 4: Funciones de Cadena

```
-- EJEMPLO 4: Funciones de texto  
-- =====
```

```
-- Convertir a mayúsculas/minúsculas  
SELECT  
    UPPER(nombres) AS nombres_mayusculas,  
    LOWER(apellidos) AS apellidos_minusculas,  
    INITCAP(email) AS email_capitalizado  
FROM Estudiante  
WHERE ROWNUM <= 5;
```

```
-- Extraer subcadenas  
SELECT  
    codigo,  
    SUBSTR(codigo, 1, 3) AS prefijo_carrera,  
    SUBSTR(codigo, 5, 3) AS numero_asignatura  
FROM Asignatura;
```

```
-- Longitud de cadenas
```




ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
SELECT
    nombres,
    LENGTH(nombres) AS longitud_nombre
FROM Estudiante
ORDER BY LENGTH(nombres) DESC;
```

-- Buscar y reemplazar

```
SELECT
    email,
    REPLACE(email, '@epn.edu.ec', '@gmail.com') AS email_alternativo
FROM Estudiante
WHERE ROWNUM <= 3;
```

-- Eliminar espacios

```
SELECT
    ' ' || nombres || ' ' AS con_espacios,
    TRIM(' ' || nombres || ' ') AS sin_espacios
FROM Estudiante
WHERE ROWNUM <= 3;
```

SQL Todas las Filas Recuperadas: 5 en 0,023 segundos			
	NOMBRES_MAYUSCULAS	APELLIDOS_MINUSCULAS	EMAIL_CAPITALIZADO
1	ANDREA	alvarez brito	Andrea.Alvarez@Epn.Edu.Ec
2	BRYAN	bravo cárdenas	Bryan.Bravo@Epn.Edu.Ec
3	CARLA	cruz delgado	Carla.Cruz@Epn.Edu.Ec
4	DAVID	diaz espinoza	David.Diaz@Epn.Edu.Ec
5	ELENA	estrella flores	Elena.Estrella@Epn.Edu.Ec



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

Resultado de la Consulta x			Resultado de la Consulta 1 x			Resultado de la Consulta 2 x		
SQL			Todas las Filas Recuperadas: 10 en 0,004 seg					
	CODIGO	PREFIJO_CARRERA		NUMERO_ASIGNATURA				
1	BD-501	BD-		01				
2	BD-601	BD-		01				
3	CIV-101	CIV		101				
4	ELE-101	ELE		101				
5	EST-301	EST		301				
6	FIS-101	FIS		101				
7	MAT-101	MAT		101				
8	MAT-201	MAT		201				
9	PRG-101	PRG		101				
10	PRG-201	PRG		201				




ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

SQL Todas las Filas Recuperadas: 16 en C		
	NOMBRES	LONGITUD_NOMBRE
1	Carlos Andr,s	13
2	María Jos,	10
3	Pedro Luis	10
4	Juan Pablo	10
5	Ana Sofía	9
6	Fernando	8
7	Gabriela	8
8	Andrea	6
9	David	5
10	Bryan	5
11	Carla	5
12	Elena	5
13	Test	4
14	Otro	4
15	Test	4
16	Hugo	4



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

	NOMBRES	LONGITUD_NOMBRE
1	Carlos Andrés	13
2	María José	10
3	Pedro Luis	10
4	Juan Pablo	10
5	Ana Sofía	9
6	Fernando	8
7	Gabriela	8
8	Andrea	6
9	David	5
10	Bryan	5
11	Carla	5
12	Elena	5
13	Test	4
14	Otro	4
15	Test	4
16	Hugo	4

 SQL | Todas las Filas Recuperadas: 3 en 0,005 segundos

	EMAIL	EMAIL_ALTERNATIVO
1	andrea.alvarez@epn.edu.ec	andrea.alvarez@gmail.com
2	bryan.bravo@epn.edu.ec	bryan.bravo@gmail.com
3	carla.cruz@epn.edu.ec	carla.cruz@gmail.com

	N_ESPACIOS
1	Andrea
2	Bryan
3	Carla



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

Ejemplo 5: Funciones de Fecha

-- EJEMPLO 5: Trabajar con fechas

-- =====

-- Fecha actual

SELECT SYSDATE AS fecha_actual FROM DUAL;

SELECT CURRENT_DATE AS fecha_actual FROM DUAL;

SELECT SYSTIMESTAMP AS timestamp_actual FROM DUAL;

-- Calcular edad de estudiantes

```
SELECT
    nombres,
    apellidos,
    fecha_nacimiento,
    TRUNC((SYSDATE - fecha_nacimiento) / 365.25) AS edad_anios,
    TRUNC(MONTHS_BETWEEN(SYSDATE, fecha_nacimiento) / 12) AS edad_meses
FROM Estudiante;
```

-- Formatear fechas

```
SELECT
    nombres,
    fecha_nacimiento,
    TO_CHAR(fecha_nacimiento, 'DD/MM/YYYY') AS fecha_formato_1,
    TO_CHAR(fecha_nacimiento, 'Day, DD "de" Month "de" YYYY') AS fecha_formato_2,
    TO_CHAR(fecha_nacimiento, 'DD-Mon-YYYY') AS fecha_formato_3
FROM Estudiante
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
WHERE ROWNUM <= 5;
```

```
-- Extraer partes de fecha
```

```
SELECT
```

```
    nombres,  
    fecha_nacimiento,  
    EXTRACT(YEAR FROM fecha_nacimiento) AS anio_nacimiento,  
    EXTRACT(MONTH FROM fecha_nacimiento) AS mes_nacimiento,  
    EXTRACT(DAY FROM fecha_nacimiento) AS dia_nacimiento
```

```
FROM Estudiante
```

```
WHERE ROWNUM <= 5;
```

```
-- Operaciones con fechas
```

```
SELECT
```

```
    nombres,  
    fecha_ingreso,  
    fecha_ingreso + 30 AS fecha_30_dias_despues,  
    ADD_MONTHS(fecha_ingreso, 6) AS fecha_6_meses_despues,  
    TRUNC(fecha_ingreso, 'MONTH') AS primer_dia_mes
```

```
FROM Estudiante
```

```
WHERE ROWNUM <= 5;
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

Resultado de la Consulta x Resultado de la Consulta 1 x Resultado de la... x x x x x

SQL | Todas las Filas Recuperadas: 5 en 0,002 segundos

	NOMBRES	FECHA_INGRESO	FECHA_30_DIAS_DESPUES	FECHA_6_MESES_DESPUES	PRIMER_DIA_MES
1	Andrea	14/11/2025	14/12/2025	14/05/2026	01/11/2025
2	Bryan	14/11/2025	14/12/2025	14/05/2026	01/11/2025
3	Carla	14/11/2025	14/12/2025	14/05/2026	01/11/2025
4	David	14/11/2025	14/12/2025	14/05/2026	01/11/2025
5	Elena	14/11/2025	14/12/2025	14/05/2026	01/11/2025

SQL | Todas las Filas Recuperadas: 16 en 0,003 segundos

	NOMBRES	APELLIDOS	FECHA_NACIMIENTO	EDAD_ANIOS	EDAD_MESES
1	Andrea	Alvarez Brito	10/01/2003	22	22
2	Bryan	Bravo Cárdenas	22/05/2002	23	23
3	Carla	Cruz Delgado	15/08/2003	22	22
4	David	Diaz Espinoza	30/11/2002	22	22
5	Elena	Estrella Flores	18/03/2003	22	22
6	Fernando	Fuentes García	25/07/2002	23	23
7	Gabriela	González Herrera	05/12/2003	21	21
8	Hugo	Hidalgo Ibarra	14/04/2002	23	23
9	Juan Pablo	Andrade Morales	15/05/2003	22	22
10	María Jos,	Benítez Castro	20/08/2002	23	23
11	Carlos Andr,s	Castillo Díaz	10/11/2003	22	22
12	Ana Sofía	Dominguez Escobar	25/03/2002	23	23
13	Pedro Luis	Espinoza Flores	18/07/2003	22	22
14	Otro	Usuario	01/01/2000	25	25
15	Test	Usuario	15/05/2000	25	25
16	Test	Usuario	15/06/2003	22	22



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

Resultado de la Consulta 2 x | Resultado de la Consulta 3 x | Resultado de la Consulta 4 x | Resul... x

Todas las Filas Recuperadas: 5 en 0,002 segundos

	NOMBRES	FECHA_INGRESO	FECHA_30_DIAS_DESPUES	FECHA_6_MESES_DESPUES	PRIMER_DIA_MES
1	Andrea	14/11/2025	14/12/2025	14/05/2026	01/11/2025
2	Bryan	14/11/2025	14/12/2025	14/05/2026	01/11/2025
3	Carla	14/11/2025	14/12/2025	14/05/2026	01/11/2025
4	David	14/11/2025	14/12/2025	14/05/2026	01/11/2025
5	Elena	14/11/2025	14/12/2025	14/05/2026	01/11/2025

Todas las Filas Recuperadas: 5 en 0,003 segundos

	NOMBRES	FECHA_NACIMIENTO	FECHA_FORMATO_1	FECHA_FORMATO_2	F
1	Andrea	10/01/2003	10/01/2003	Viernes , 10 de Enero de 2003	10-
2	Bryan	22/05/2002	22/05/2002	Miércoles, 22 de Mayo de 2002	22-
3	Carla	15/08/2003	15/08/2003	Viernes , 15 de Agosto de 2003	15-
4	David	30/11/2002	30/11/2002	Sábado , 30 de Noviembre de 2002	30-
5	Elena	18/03/2003	18/03/2003	Martes , 18 de Marzo de 2003	18-

	NOMBRES	FECHA_NACIMIENTO	ANIO_NACIMIENTO	MES_NACIMIENTO	DIA_NACIMIENTO
1	Andrea	10/01/2003	2003	1	10
2	Bryan	22/05/2002	2002	5	22
3	Carla	15/08/2003	2003	8	15
4	David	30/11/2002	2002	11	30
5	Elena	18/03/2003	2003	3	18



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

	⚡ NOMBRES	⚡ FECHA_INGRESO	⚡ FECHA_30_DIAS_DESPUES	⚡ FECHA_6_MESES_DESPUES	⚡ PRIMER_DIA_MES
1	Andrea	14/11/2025	14/12/2025	14/05/2026	01/11/2025
2	Bryan	14/11/2025	14/12/2025	14/05/2026	01/11/2025
3	Carla	14/11/2025	14/12/2025	14/05/2026	01/11/2025
4	David	14/11/2025	14/12/2025	14/05/2026	01/11/2025
5	Elena	14/11/2025	14/12/2025	14/05/2026	01/11/2025

Ejemplo 6: Funciones de Conversión

```
-- EJEMPLO 6: Conversión de tipos de datos
-- =====

-- TO_CHAR: Convertir a cadena
SELECT
    creditos_aprobados,
    TO_CHAR(creditos_aprobados) AS creditos_texto,
    TO_CHAR(creditos_aprobados, '999') AS creditos_formato
FROM Estudiante
WHERE ROWNUM <= 5;

-- TO_NUMBER: Convertir a número
SELECT
    codigo,
    TO_NUMBER(SUBSTR(codigo, 5, 3)) AS numero_asignatura
FROM Asignatura
WHERE REGEXP_LIKE(SUBSTR(codigo, 5, 3), '^\\d+$')
AND ROWNUM <= 5;
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

-- TO_DATE: Convertir a fecha

```
SELECT
    TO_DATE('2024-09-01', 'YYYY-MM-DD') AS fecha_inicio_semestre,
    TO_DATE('15/12/2024', 'DD/MM/YYYY') AS fecha_fin_semestre
FROM DUAL;
```

-- NVL: Manejar valores NULL

```
SELECT
    nombres,
    telefono,
    NVL(telefono, 'Sin teléfono') AS telefono_con_default
FROM Estudiante
WHERE ROWNUM <= 10;
```

-- NVL2: Valor si es NULL o no NULL

```
SELECT
    nombres,
    telefono,
    NVL2(telefono, 'Tiene teléfono', 'No tiene teléfono') AS tiene_telefono
FROM Estudiante
WHERE ROWNUM <= 10;
```

-- COALESCE: Primera expresión no nula

```
SELECT
    nombres,
    telefono,
    email,
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

COALESCE(teléfono, email, 'Sin contacto') AS contacto_disponible
FROM Estudiante
WHERE ROWNUM <= 5;

	NOMBRES	TELEFONO	EMAIL	CONTACTO_DISPONIBLE
1	Andrea	0991111111	andrea.alvarez@epn.edu.ec	0991111111
2	Bryan	0992222222	bryan.bravo@epn.edu.ec	0992222222
3	Carla	0993333333	carla.cruz@epn.edu.ec	0993333333
4	David	0994444444	david.diaz@epn.edu.ec	0994444444
5	Elena	0995555555	elena.estrella@epn.edu.ec	0995555555

	CREDITOS_APROBADOS	CREDITOS_TEXTO	CREDITOS_FORMATO
1	4 4	4	
2	9 9	9	
3	10 10	10	
4	25 25	25	
5	4 4	4	

	CODIGO	NUMERO_ASIGNATURA
1	BD-501	1
2	BD-601	1
3	CIV-101	101
4	ELE-101	101
5	EST-301	301

	FECHA_INICIO_SEMESTRE	FECHA_FIN_SEMESTRE
1	01/09/2024	15/12/2024



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

⚡ NOMBRES	⚡ TELEFONO	⚡ TELEFONO_CON_DEFAULT
1 Andrea	0991111111	0991111111
2 Bryan	0992222222	0992222222
3 Carla	0993333333	0993333333
4 David	0994444444	0994444444
5 Elena	0995555555	0995555555
6 Fernando	0996666666	0996666666
7 Gabriela	0997777777	0997777777
8 Hugo	0998888888	0998888888
9 Juan Pablo	0991111111	0991111111
10 María Jos,	0992345678	0992345678

⚡ NOMBRES	⚡ TELEFONO	⚡ TIENE_TELEFONO
1 Andrea	0991111111	Tiene teléfono
2 Bryan	0992222222	Tiene teléfono
3 Carla	0993333333	Tiene teléfono
4 David	0994444444	Tiene teléfono
5 Elena	0995555555	Tiene teléfono
6 Fernando	0996666666	Tiene teléfono
7 Gabriela	0997777777	Tiene teléfono
8 Hugo	0998888888	Tiene teléfono
9 Juan Pablo	0991111111	Tiene teléfono
10 María Jos,	0992345678	Tiene teléfono

CLÁUSULA WHERE Y OPERADORES

Operadores de Comparación

-- OPERADORES DE COMPARACIÓN

-- =====



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

-- Igual (=)

SELECT * FROM Estudiante WHERE estado = 'ACTIVO';

-- Diferente (!= o <>)

SELECT * FROM Estudiante WHERE genero != 'M';

SELECT * FROM Estudiante WHERE genero <> 'M';

-- Mayor que (>)

SELECT * FROM Estudiante WHERE creditos_aprobados > 20;

-- Mayor o igual (>=)

SELECT * FROM Asignatura WHERE creditos >= 5;

-- Menor que (<)

SELECT * FROM Estudiante WHERE creditos_aprobados < 10;

-- Menor o igual (<=)

SELECT * FROM Asignatura WHERE nivel <= 3;

	⚡ CEDULA	⚡ NOMBRES	⚡ APELLIDOS	⚡ EMAIL	⚡ TELEFONO	⚡ FE
1	1750700001	Andrea	Alvarez Brito	andrea.alvarez@epn.edu.ec	0991111111	10/0
2	1750700003	Carla	Cruz Delgado	carla.cruz@epn.edu.ec	0993333333	15/0
3	1750700005	Elena	Estrella Flores	elena.estrella@epn.edu.ec	0995555555	18/0
4	1750700007	Gabriela	González Herrera	gabriela.gonzalez@epn.edu.ec	0997777777	05/1
5	1750234567	Marija Jos,	Benitez Castro	maria.benitez@epn.edu.ec	0992345678	20/0
6	1750456789	Ana Sofía	Dominguez Escobar	ana.dominguez@epn.edu.ec	0994567890	25/0



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

Todas las Filas Recuperadas: 6 en 0,002 segundos

	CEDULA	NOMBRES	APELLIDOS	EMAIL	TELEFONO	FECHA
1	1750700001	Andrea	Alvarez Brito	andrea.alvarez@epn.edu.ec	0991111111	10/0
2	1750700003	Carla	Cruz Delgado	carla.cruz@epn.edu.ec	0993333333	15/0
3	1750700005	Elena	Estrella Flores	elena.estrella@epn.edu.ec	0995555555	18/0
4	1750700007	Gabriela	González Herrera	gabriela.gonzalez@epn.edu.ec	0997777777	05/1
5	1750234567	Maria Jos,	Benitez Castro	maria.benitez@epn.edu.ec	0992345678	20/0
6	1750456789	Ana Sofia	Dominguez Escobar	ana.dominguez@epn.edu.ec	0994567890	25/0

	CEDULA	NOMBRES	APELLIDOS	EMAIL	TELEFONO	FECHA
1	1750700004	David	Diaz Espinoza	david.diaz@epn.edu.ec	0994444444	30/1
2	1750700007	Gabriela	González Herrera	gabriela.gonzalez@epn.edu.ec	0997777777	05/1
3	1750700008	Hugo	Hidalgo Ibarra	hugo.hidalgo@epn.edu.ec	0998888888	14/0
4	1750123456	Juan Pablo	Andrade Morales	juan.andrade.nuevo@epn.edu.ec	0991111111	15/0

	CODIGO	NOMBRE	CREDITOS	NIVEL	CODIGO_CARRERA	HORAS_1
1	MAT-101	C lculo Diferencial	5	1	ING-SIS	
2	PRG-101	Fundamentos de Programaci n	5	1	ING-SIS	
3	MAT-201	C lculo Integral	5	2	ING-SIS	
4	PRG-201	Programaci n Orientada a Objetos	5	2	ING-SIS	
5	EST-301	Estructura de Datos	5	3	ING-SIS	
6	BD-601	Bases de Datos Avanzadas	5	6	ING-SIS	
7	ELE-101	Circuitos El,ctricos I	5	1	ING-ELE	



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

	⚡ CEDULA	⚡ NOMBRES	⚡ APELLIDOS	⚡ EMAIL	⚡ TELEFONO	⚡ I
1	1750700001	Andrea	Alvarez Brito	andrea.alvarez@epn.edu.ec	0991111111	10
2	1750700002	Bryan	Bravo Cárdenas	bryan.bravo@epn.edu.ec	0992222222	22
3	1750700005	Elena	Estrella Flores	elena.estrella@epn.edu.ec	0995555555	18
4	1750700006	Fernando	Fuentes García	fernando.fuentes@epn.edu.ec	0996666666	25
5	1750234567	María Jos,	Benítez Castro	maria.benitez@epn.edu.ec	0992345678	20
6	1750345678	Carlos Andr,s	Castillo Díaz	carlos.castillo@epn.edu.ec	0993456789	10
7	1750456789	Ana Sofía	Dominguez Escobar	ana.dominguez@epn.edu.ec	0994567890	25
8	1750567890	Pedro Luis	Espinoza Flores	pedro.espinoza@epn.edu.ec	0995678901	18
9	1750777777	Otro	Usuario	juan.andrade@epn.edu.ec	(null)	01
10	1750444444	Test	Usuario	test5@epn.edu.ec	(null)	15
11	1750333333	Test	Usuario	test6@epn.edu.ec	(null)	15

	⚡ CO...	⚡ NOMBRE	⚡ CREDITOS	⚡ NIVEL	⚡ CODIGO_CARRERA	⚡ HORAS_1
1	MAT-101	C lculo Diferencial	5	1	ING-SIS	
2	FIS-101	Física I	4	1	ING-SIS	
3	PRG-101	Fundamentos de Programaci	5	1	ING-SIS	
4	MAT-201	C lculo Integral	5	2	ING-SIS	
5	PRG-201	Programaci	5	2	ING-SIS	
6	EST-301	Estructura de Datos	5	3	ING-SIS	
7	CIV-101	Dibujo T,cnico	4	1	ING-CIV	
8	ELE-101	Circuitos El,ctricos I	5	1	ING-ELE	

Operadores Lógicos

-- OPERADORES LÓGICOS: AND, OR, NOT

-- =====

-- AND: Ambas condiciones deben cumplirse

SELECT nombres, apellidos, codigo_carrera, creditos_aprobados

FROM Estudiante



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
WHERE codigo_carrera = 'ING-SIS'  
AND creditos_aprobados >= 20;
```

-- OR: Al menos una condición debe cumplirse

```
SELECT nombres, apellidos, estado  
FROM Estudiante  
WHERE estado = 'GRADUADO'  
OR creditos_aprobados >= 180;
```

-- NOT: Negación

```
SELECT nombres, apellidos, genero  
FROM Estudiante  
WHERE NOT genero = 'M'; -- Equivalente a genero != 'M'
```

-- Combinación de operadores (usar paréntesis)

```
SELECT nombres, apellidos, codigo_carrera, creditos_aprobados  
FROM Estudiante  
WHERE (codigo_carrera = 'ING-SIS' OR codigo_carrera = 'ING-CIV')  
AND creditos_aprobados > 15  
AND estado = 'ACTIVO';
```

	NOMBRES	APELLIDOS	CODIGO_CARRERA	CREDITOS_APROBADOS
1	Juan Pablo	Andrade Morales	ING-SIS	28

	NO...	APELLIDOS	ESTADO
1	Hugo	Hidalgo Ibarra	INACTIVO



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

	NOMBRES	APELLIDOS	GENERO
1	Andrea	Alvarez Brito	F
2	Carla	Cruz Delgado	F
3	Elena	Estrella Flores	F
4	Gabriela	González Herrera	F
5	María Jos,	Benítez Castro	F
6	Ana Sofía	Domínguez Escobar	F

Operador BETWEEN

-- BETWEEN: Rango de valores

-- =====

-- Estudiantes con créditos entre 10 y 30

```
SELECT nombres, apellidos, creditos_aprobados
FROM Estudiante
WHERE creditos_aprobados BETWEEN 10 AND 30
ORDER BY creditos_aprobados;
```

-- Asignaturas de nivel 2 a 5

```
SELECT codigo, nombre, nivel
FROM Asignatura
WHERE nivel BETWEEN 2 AND 5
ORDER BY nivel, codigo;
```

-- Estudiantes nacidos en 2002

```
SELECT nombres, apellidos, fecha_nacimiento
FROM Estudiante
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
WHERE fecha_nacimiento BETWEEN TO_DATE('2002-01-01', 'YYYY-MM-DD')  
      AND TO_DATE('2002-12-31', 'YYYY-MM-DD')  
ORDER BY fecha_nacimiento;
```

-- NOT BETWEEN

```
SELECT nombres, creditos_aprobados  
FROM Estudiante  
WHERE creditos_aprobados NOT BETWEEN 10 AND 30;
```

	NOMBRES	APELLIDOS	FECHA_NACIMIENTO
1	Ana Sofía	Domínguez Escobar	25/03/2002
2	Hugo	Hidalgo Ibarra	14/04/2002
3	Bryan	Bravo Cárdenas	22/05/2002
4	Fernando	Fuentes García	25/07/2002
5	María Jos,	Benítez Castro	20/08/2002
6	David	Díaz Espinoza	30/11/2002



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

	NOMBRES	CREDITOS_APROBADOS
1	Andrea	4
2	Bryan	9
3	Elena	4
4	Fernando	5
5	Gabriela	35
6	Hugo	180
7	María Jos,	4
8	Carlos Andr,s	0
9	Ana Sofía	0
10	Pedro Luis	5
11	Otro	0
12	Test	0
13	Test	0

	NOMBRES	APELLIDOS	CREDITOS_APROBADOS
1	Carla	Cruz Delgado	10
2	David	Diaz Espinoza	25
3	Juan Pablo	Andrade Morales	28

	CO...	NOMBRE	NIVEL
1	MAT-201	C lculo Integral	2
2	PRG-201	Programaci en Orientada a Objetos	2
3	EST-301	Estructura de Datos	3
4	BD-501	Fundamentos de Bases de Datos	5



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

	NOMBRES	APELLIDOS	FECHA_NACIMIENTO
1	Ana Sofía	Domínguez Escobar	25/03/2002
2	Hugo	Hidalgo Ibarra	14/04/2002
3	Bryan	Bravo Cárdenas	22/05/2002
4	Fernando	Fuentes García	25/07/2002
5	María Jos,	Benítez Castro	20/08/2002
6	David	Díaz Espinoza	30/11/2002

	NOMBRES	CREDITOS_APROBADOS
1	Andrea	4
2	Bryan	9
3	Elena	4
4	Fernando	5
5	Gabriela	35
6	Hugo	180
7	María Jos,	4
8	Carlos Andr,s	0
9	Ana Sofía	0
10	Pedro Luis	5
11	Otro	0
12	Test	0
13	Test	0

Operador IN

-- IN: Lista de valores
-- =====

-- Estudiantes de carreras específicas

SELECT nombres, apellidos, codigo_carrera



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
FROM Estudiante
WHERE codigo_carrera IN ('ING-SIS', 'ING-CIV', 'ING-ELE')
ORDER BY codigo_carrera, apellidos;
```

-- Estados válidos

```
SELECT nombres, apellidos, estado
FROM Estudiante
WHERE estado IN ('ACTIVO', 'GRADUADO')
ORDER BY estado, apellidos;
```

-- NOT IN

```
SELECT nombres, apellidos, estado
FROM Estudiante
WHERE estado NOT IN ('ACTIVO', 'GRADUADO')
ORDER BY apellidos;
```

-- IN con subconsulta (ver más adelante)

```
SELECT codigo, nombre
FROM Asignatura
WHERE codigo_carrera IN (
    SELECT codigo FROM Carrera WHERE duracion_semestres = 10
);
```

-- LIKE: Búsqueda de patrones

-- =====



**ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS**

-- % = *Cualquier secuencia de caracteres (0 o más)*

-- _ = *Un solo carácter*

-- *Estudiantes cuyo apellido empieza con 'G'*

```
SELECT nombres, apellidos
FROM Estudiante
WHERE apellidos LIKE 'G%'
ORDER BY apellidos;
```

-- *Estudiantes con 'ana' en el nombre*

```
SELECT nombres, apellidos
FROM Estudiante
WHERE LOWER(nombres) LIKE '%ana%'
ORDER BY nombres;
```

-- *Emails que contienen 'garcia'*

```
SELECT nombres, email
FROM Estudiante
WHERE email LIKE '%garcia%';
```

-- *Cédulas que terminan en '01'*

```
SELECT cedula, nombres, apellidos
FROM Estudiante
WHERE cedula LIKE '%01';
```

-- *Códigos de asignatura con patrón específico (XXX-###)*

```
SELECT codigo, nombre
```



**ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS**

FROM Asignatura

WHERE codigo **LIKE** '____-____'; -- 3 caracteres, guion, 3 caracteres

-- *NOT LIKE*

SELECT nombres, email

FROM Estudiante

WHERE email **NOT LIKE** '%@epn.edu.ec';

-- *LIKE con ESCAPE (para buscar % o _)*

-- *Si necesita buscar el carácter % literal*

SELECT * FROM Tabla **WHERE** columna **LIKE** '%\%%%' **ESCAPE** '\';



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

NOMBRES	APELLIDOS	ESTADO
1 Andrea	Alvarez Brito	INACTIVO
2 Juan Pablo	Andrade Morales	INACTIVO
3 María Jos,	Benítez Castro	INACTIVO
4 Bryan	Bravo Cárdenas	INACTIVO
5 Carlos Andr,s	Castillo Díaz	INACTIVO
6 Carla	Cruz Delgado	INACTIVO
7 David	Diaz Espinoza	INACTIVO
8 Ana Sofía	Domínguez Escobar	INACTIVO
9 Pedro Luis	Espinoza Flores	INACTIVO
10 Elena	Estrella Flores	INACTIVO
11 Fernando	Fuentes García	INACTIVO
12 Gabriela	González Herrera	INACTIVO
13 Hugo	Hidalgo Ibarra	INACTIVO
14 Otro	Usuario	INACTIVO
15 Test	Usuario	INACTIVO
16 Test	Usuario	INACTIVO

CODIGO	NOMBRE
1 MAT-101	C lculo Diferencial
2 FIS-101	Física I
3 PRG-101	Fundamentos de Programaci n
4 MAT-201	C lculo Integral
5 PRG-201	Programaci n Orientada a Objetos
6 EST-301	Estructura de Datos
7 BD-501	Fundamentos de Bases de Datos
8 BD-601	Bases de Datos Avanzadas
9 CIV-101	Dibujo T,cnico
10 ELE-101	Circuitos El,ctricos I



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

NOMBRES	APELLIDOS
1 Gabriela	González Herrera

NOMBRES	APELLIDOS
1 Ana Sofía	Dominguez Escobar

CODIGO	NOMBRE
--------	--------

Operador IS NULL

```
-- IS NULL / IS NOT NULL
--

-- Estudiantes sin teléfono registrado
SELECT nombres, apellidos, telefono
FROM Estudiante
WHERE telefono IS NULL;

-- Estudiantes con teléfono registrado
SELECT nombres, apellidos, telefono
FROM Estudiante
WHERE telefono IS NOT NULL;

-- Matrículas sin calificaciones
SELECT
  m.id_matricula,
  e.nombres,
  a.nombre AS asignatura
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
FROM Matricula m
JOIN Estudiante e ON m.cedula_estudiante = e.cedula
JOIN Asignatura a ON m.codigo_asignatura = a.codigo
WHERE m.nota_final IS NULL;
```

-- IMPORTANTE: No usar = NULL o != NULL

-- ✗ INCORRECTO

SELECT * FROM Estudiante WHERE telefono = NULL; -- No funciona

-- ✓ CORRECTO

SELECT * FROM Estudiante WHERE telefono IS NULL;

	NOMBRES	APELLIDOS	TELEFONO
1	Otro	Usuario	(null)
2	Test	Usuario	(null)
3	Test	Usuario	(null)

	NOMBRES	APELLIDOS	TELEFONO
1	Andrea	Alvarez Brito	0991111111
2	Bryan	Bravo Cárdenas	0992222222
3	Carla	Cruz Delgado	0993333333
4	David	Diaz Espinoza	0994444444
5	Elena	Estrella Flores	0995555555
6	Fernando	Fuentes García	0996666666
7	Gabriela	González Herrera	0997777777
8	Hugo	Hidalgo Ibarra	0998888888
9	Juan Pablo	Andrade Morales	0991111111
10	María Jos,	Benítez Castro	0992345678
11	Carlos Andr,s	Castillo Díaz	0993456789
12	Ana Sofía	Domínguez Escobar	0994567890
13	Pedro Luis	Espinoza Flores	0995678901



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

	ID_MATRICULA	NOMBRES	ASIGNATURA
1	1	Juan Pablo	Fundamentos de Bases de Datos
2	4	Carlos Andrés	Dibujo Técnico
3	5	Ana Sofía	Circuitos Eléctricos I
4	3	Pedro Luis	Fundamentos de Bases de Datos

Operador exists

-- EXISTS: Verificar existencia en subconsulta

-- =====

-- Estudiantes que tienen al menos una matrícula

```
SELECT e.cedula, e.nombres, e.apellidos
FROM Estudiante e
WHERE EXISTS (
    SELECT 1 FROM Matricula m
    WHERE m.cedula_estudiante = e.cedula
);
```

-- Carreras que tienen estudiantes matriculados

```
SELECT c.codigo, c.nombre
FROM Carrera c
WHERE EXISTS (
    SELECT 1 FROM Estudiante e
    WHERE e.codigo_carrera = c.codigo
);
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

-- NOT EXISTS: Estudiantes sin matrículas

```
SELECT e.cedula, e.nombres, e.apellidos
FROM Estudiante e
WHERE NOT EXISTS (
    SELECT 1 FROM Matricula m
    WHERE m.cedula_estudiante = e.cedula
);
```

	⚙ CEDULA	⚙ NOMBRES	⚙ APELLIDOS	
1	1750700003	Carla	Cruz Delgado	
2	1750700004	David	Diaz Espinoza	
3	1750700006	Fernando	Fuentes García	
4	1750700007	Gabriela	González Herrera	
5	1750700008	Hugo	Hidalgo Ibarra	
6	1750777777	Otro	Usuario	
7	1750444444	Test	Usuario	
8	1750333333	Test	Usuario	

	⚙ CEDULA	⚙ NOMBRES	⚙ APELLIDOS	
1	1750700003	Carla	Cruz Delgado	
2	1750700004	David	Diaz Espinoza	
3	1750700006	Fernando	Fuentes García	
4	1750700007	Gabriela	González Herrera	
5	1750700008	Hugo	Hidalgo Ibarra	
6	1750777777	Otro	Usuario	
7	1750444444	Test	Usuario	
8	1750333333	Test	Usuario	



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

	⚡ CEDULA	⚡ NOMBRES	⚡ APELLIDOS
1	1750700003	Carla	Cruz Delgado
2	1750700004	David	Diaz Espinoza
3	1750700006	Fernando	Fuentes García
4	1750700007	Gabriela	González Herrera
5	1750700008	Hugo	Hidalgo Ibarra
6	1750777777	Otro	Usuario
7	1750444444	Test	Usuario
8	1750333333	Test	Usuario

Ejemplo Completo: Múltiples Condiciones

-- EJEMPLO COMPLETO: Combinación de operadores

-- =====

-- Consulta compleja con múltiples condiciones

SELECT

e.cedula,

e.nombres || ' ' || e.apellidos AS nombre_completo,

e.codigo_carrera,

e.creditos_aprobados,

e.estado,

TRUNC(MONTHS_BETWEEN(SYSDATE, e.fecha_nacimiento) / 12) AS edad

FROM Estudiante e

WHERE

-- Condiciones con AND/OR

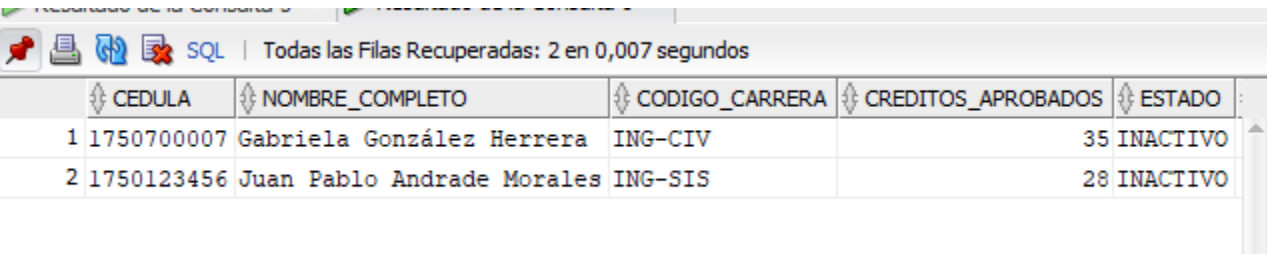
(e.codigo_carrera IN ('ING-SIS', 'ING-CIV'))



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
OR e.creditos_aprobados >= 30)
-- Rango de créditos
AND e.creditos_aprobados BETWEEN 10 AND 100
-- Estado activo
AND e.estado = 'ACTIVO'
-- Apellido empieza con letra específica
AND e.apellidos LIKE 'A%' OR e.apellidos LIKE 'G%'
-- Tiene teléfono
AND e.telefono IS NOT NULL
-- Mayor de edad
AND MONTHS_BETWEEN(SYSDATE, e.fecha_nacimiento) / 12 >= 18
ORDER BY
e.codigo_carrera,
e.creditos_aprobados DESC,
e.apellidos;
```

Se tuvo que hacer un cambio en la parte de activo porque resultaba en una conslta vacia



Todas las Filas Recuperadas: 2 en 0,007 segundos

	CEDULA	NOMBRE_COMPLETO	CODIGO_CARRERA	CREDITOS_APROBADOS	ESTADO
1	1750700007	Gabriela González Herrera	ING-CIV	35	INACTIVO
2	1750123456	Juan Pablo Andrade Morales	ING-SIS	28	INACTIVO

ORDER BY - ORDENAMIENTO



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

Sintaxis y Ejemplos

-- *ORDER BY: Ordenar resultados*

-- =====

-- *Orden ascendente (ASC) - por defecto*

```
SELECT nombres, apellidos, creditos_aprobados
FROM Estudiante
ORDER BY apellidos ASC;
```

-- *Orden descendente (DESC)*

```
SELECT nombres, apellidos, creditos_aprobados
FROM Estudiante
ORDER BY creditos_aprobados DESC;
```

-- *Ordenar por múltiples columnas*

```
SELECT codigo_carrera, apellidos, nombres, creditos_aprobados
FROM Estudiante
ORDER BY codigo_carrera ASC, creditos_aprobados DESC, apellidos ASC;
```

-- *Ordenar por posición de columna (no recomendado)*

```
SELECT nombres, apellidos, creditos_aprobados
FROM Estudiante
ORDER BY 3 DESC, 2 ASC; -- 3=creditos_aprobados, 2=apellidos
```

-- *Ordenar por expresión calculada*

```
SELECT
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
nombres,  
apellidos,  
creditos_aprobados,  
(creditos_aprobados / 180 * 100) AS porcentaje_avance
```

```
FROM Estudiante
```

```
WHERE creditos_aprobados > 0
```

```
ORDER BY (creditos_aprobados / 180 * 100) DESC;
```

```
-- Ordenar por alias
```

```
SELECT
```

```
nombres || ' ' || apellidos AS nombre_completo,  
creditos_aprobados
```

```
FROM Estudiante
```

```
ORDER BY nombre_completo;
```

```
-- Ordenar con NULL
```

```
-- NULLS FIRST: NULL al inicio
```

```
SELECT nombres, telefono
```

```
FROM Estudiante
```

```
ORDER BY telefono NULLS FIRST;
```

```
-- NULLS LAST: NULL al final (por defecto en ASC)
```

```
SELECT nombres, telefono
```

```
FROM Estudiante
```

```
ORDER BY telefono NULLS LAST;
```

```
-- Ordenar por fecha
```




**ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS**

```
SELECT nombres, apellidos, fecha_nacimiento  
FROM Estudiante  
ORDER BY fecha_nacimiento DESC; -- Más jóvenes primero
```

-- Ordenar resultados de JOIN

```
SELECT  
    e.apellidos,  
    e.nombres,  
    c.nombre AS carrera  
FROM Estudiante e  
JOIN Carrera c ON e.codigo_carrera = c.codigo  
ORDER BY c.nombre, e.apellidos;
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

	↕ NOMBRES	↕ APELLIDOS	↕ CREDITOS_APROBADOS
1	Hugo	Hidalgo Ibarra	180
2	Gabriela	González Herrera	35
3	Juan Pablo	Andrade Morales	28
4	David	Diaz Espinoza	25
5	Carla	Cruz Delgado	10
6	Bryan	Bravo Cárdenas	9
7	Pedro Luis	Espinoza Flores	5
8	Fernando	Fuentes García	5
9	María Jos,	Benítez Castro	4
10	Andrea	Alvarez Brito	4
11	Elena	Estrella Flores	4
12	Test	Usuario	0
13	Otro	Usuario	0
14	Ana Sofía	Domínguez Escobar	0
15	Test	Usuario	0
16	Carlos Andrés	Castillo Díaz	0



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

	❖ CODIGO_CARRERA	❖ APELLIDOS	❖ NOMBRES	❖ CREDITOS_APROBADOS	
1	ING-CIV	González Herrera	Gabriela	35	
2	ING-CIV	Cruz Delgado	Carla	10	
3	ING-CIV	Castillo Díaz	Carlos Andr,s	0	
4	ING-ELE	Hidalgo Ibarra	Hugo	180	
5	ING-ELE	Díaz Espinoza	David	25	
6	ING-ELE	Domínguez Escobar	Ana Sofía	0	
7	ING-SIS	Andrade Morales	Juan Pablo	28	
8	ING-SIS	Bravo Cárdenas	Bryan	9	
9	ING-SIS	Espinoza Flores	Pedro Luis	5	
10	ING-SIS	Fuentes García	Fernando	5	
11	ING-SIS	Alvarez Brito	Andrea	4	
12	ING-SIS	Benítez Castro	María Jos,	4	
13	ING-SIS	Estrella Flores	Elena	4	
14	ING-SIS	Usuario	Test	0	
15	ING-SIS	Usuario	Otro	0	
16	ING-SIS	Usuario	Test	0	



	NOMB...	APELLIDOS	CREDITOS_APROBADOS	PORCENTAJE_AVANCE
1	Hugo	Hidalgo Ibarra	180	
2	Gabriela	González Herrera	35	19,444444444444444444444444444444
3	Juan Pablo	Andrade Morales	28	15,5555555555555555555555555555
4	David	Diaz Espinoza	25	13,888888888888888888888888888888
5	Carla	Cruz Delgado	10	5,555555555555555555555555555555
6	Bryan	Bravo Cárdenas	9	
7	Pedro Luis	Espinoza Flores	5	2,777777777777777777777777777777
8	Fernando	Fuentes García	5	2,777777777777777777777777777777
9	Andrea	Alvarez Brito	4	2,222222222222222222222222222222
10	María Jos,	Benítez Castro	4	2,222222222222222222222222222222
11	Elena	Estrella Flores	4	2,222222222222222222222222222222



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

	◊ NOMBRE_COMPLETO	◊ CREDITOS_APROBADOS
1	Ana Sofía Domínguez Escobar	0
2	Andrea Alvarez Brito	4
3	Bryan Bravo Cárdenas	9
4	Carla Cruz Delgado	10
5	Carlos Andrés Castillo Díaz	0
6	David Diaz Espinoza	25
7	Elena Estrella Flores	4
8	Fernando Fuentes García	5
9	Gabriela González Herrera	35
10	Hugo Hidalgo Ibarra	180
11	Juan Pablo Andrade Morales	28
12	María José Benítez Castro	4
13	Otro Usuario	0
14	Pedro Luis Espinoza Flores	5
15	Test Usuario	0
16	Test Usuario	0



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

	NOMBRES	TELEFONO
1	Andrea	0991111111
2	Juan Pablo	0991111111
3	Bryan	0992222222
4	María Jos,	0992345678
5	Carla	0993333333
6	Carlos Andr,s	0993456789
7	David	0994444444
8	Ana Sofía	0994567890
9	Elena	0995555555
10	Pedro Luis	0995678901
11	Fernando	0996666666
12	Gabriela	0997777777
13	Hugo	0998888888
14	Otro	(null)
15	Test	(null)
16	Test	(null)



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

NOMBRES	APELLIDOS	FECHA_NACIMIENTO
1 Gabriela	González Herrera	05/12/2003
2 Carlos Andr,s	Castillo Díaz	10/11/2003
3 Carla	Cruz Delgado	15/08/2003
4 Pedro Luis	Espinoza Flores	18/07/2003
5 Test	Usuario	15/06/2003
6 Juan Pablo	Andrade Morales	15/05/2003
7 Elena	Estrella Flores	18/03/2003
8 Andrea	Alvarez Brito	10/01/2003
9 David	Diaz Espinoza	30/11/2002
10 María Jos,	Benítez Castro	20/08/2002
11 Fernando	Fuentes García	25/07/2002
12 Bryan	Bravo Cárdenas	22/05/2002
13 Hugo	Hidalgo Ibarra	14/04/2002
14 Ana Sofía	Domínguez Escobar	25/03/2002
15 Test	Usuario	15/05/2000
16 Otro	Usuario	01/01/2000



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

⬇	⬇ APELLIDOS	⬇ NOMBRES	⬇ CARRERA	
1	Castillo Díaz	Carlos Andrés	Ingeniería Civil	
2	Cruz Delgado	Carla	Ingeniería Civil	
3	González Herrera	Gabriela	Ingeniería Civil	
4	Díaz Espinoza	David	Ingeniería Eléctrica	
5	Domínguez Escobar	Ana Sofía	Ingeniería Eléctrica	
6	Hidalgo Ibarra	Hugo	Ingeniería Eléctrica	
7	Alvarez Brito	Andrea	Ingeniería en Sistemas Informáticos	
8	Andrade Morales	Juan Pablo	Ingeniería en Sistemas Informáticos	
9	Benítez Castro	María José	Ingeniería en Sistemas Informáticos	
10	Bravo Cárdenas	Bryan	Ingeniería en Sistemas Informáticos	
11	Espinoza Flores	Pedro Luis	Ingeniería en Sistemas Informáticos	
12	Estrella Flores	Elena	Ingeniería en Sistemas Informáticos	
13	Fuentes García	Fernando	Ingeniería en Sistemas Informáticos	
14	Usuario	Test	Ingeniería en Sistemas Informáticos	
15	Usuario	Otro	Ingeniería en Sistemas Informáticos	
16	Usuario	Test	Ingeniería en Sistemas Informáticos	

FUNCIONES DE AGREGACIÓN

Funciones Básicas

-- FUNCIONES DE AGREGACIÓN

-- =====

-- COUNT: Contar registros

SELECT COUNT(*) AS total_estudiantes FROM Estudiante;

-- COUNT con columna específica (no cuenta NULL)



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
SELECT COUNT(telefono) AS estudiantes_con_telefono FROM Estudiante;
```

-- COUNT DISTINCT: Contar valores únicos

```
SELECT COUNT(DISTINCT codigo_carrera) AS total_carreras FROM Estudiante;
```

-- SUM: Suma total

```
SELECT SUM(creditos) AS total_creditos FROM Asignatura;
```

```
SELECT SUM(creditos_aprobados) AS total_creditos_todos_estudiantes FROM Estudiante;
```

-- AVG: Promedio

```
SELECT AVG(creditos_aprobados) AS promedio_creditos FROM Estudiante;
```

```
SELECT AVG(nota_final) AS promedio_nota_final FROM Matricula WHERE nota_final IS NOT NULL;
```

-- MAX: Valor máximo

```
SELECT MAX(creditos_aprobados) AS maximo_creditos FROM Estudiante;
```

```
SELECT MAX(nota_final) AS nota_maxima FROM Matricula WHERE nota_final IS NOT NULL;
```

-- MIN: Valor mínimo

```
SELECT MIN(creditos_aprobados) AS minimo_creditos FROM Estudiante WHERE creditos_aprobados > 0;
```

```
SELECT MIN(nota_final) AS nota_minima FROM Matricula WHERE nota_final IS NOT NULL;
```

-- Múltiples agregaciones en una consulta

```
SELECT  
  
    COUNT(*) AS total_estudiantes,  
    COUNT(DISTINCT codigo_carrera) AS total_carreras,  
    AVG(creditos_aprobados) AS promedio_creditos,  
    MAX(creditos_aprobados) AS maximo_creditos,
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

MIN(creditos_aprobados) AS minimo_creditos,
SUM(creditos_aprobados) AS suma_total_creditos
FROM Estudiante;

TOTAL_ESTUDIANTES	
1	16

ESTUDIANTES_CON_TELEFONO	
1	13

TOTAL_CARRERAS	
1	3

TOTAL_CREDITOS	
1	47

TOTAL_CREDITOS_TODOS_ESTUDIANTES	
1	309

PROMEDIO_CREDITOS	
1	19,3125



	TOTAL_ESTUDI...	TOTAL_CARRERAS	PROMEDIO_CREDITOS	MAXIMO_CREDITOS	MINIMO_CREDIT
1	16	3	19,3125	180	

Ejemplos Prácticos



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

-- EJEMPLOS PRÁCTICOS DE AGREGACIÓN

-- =====

-- Estadísticas de calificaciones

```
SELECT
  COUNT(*) AS total_matriculas,
  COUNT(nota_final) AS matriculas_con_nota,
  ROUND(AVG(nota_final), 2) AS promedio_nota,
  MAX(nota_final) AS nota_maxima,
  MIN(nota_final) AS nota_minima,
  COUNT(CASE WHEN estado = 'APROBADO' THEN 1 END) AS total_aprobados,
  COUNT(CASE WHEN estado = 'REPROBADO' THEN 1 END) AS total_reprobados
FROM Matricula
WHERE nota_final IS NOT NULL;
```

-- Estadísticas de asignaturas

```
SELECT
  COUNT(*) AS total_asignaturas,
  AVG(creditos) AS promedio_creditos,
  SUM(creditos) AS total_creditos,
  AVG(horas_teoría + horas_práctica) AS promedio_horas_totales
FROM Asignatura;
```

-- Edad promedio de estudiantes

```
SELECT
  ROUND(AVG(TRUNC(MONTHS_BETWEEN(SYSDATE, fecha_nacimiento) / 12)), 1) AS edad_promedio,
  MIN(TRUNC(MONTHS_BETWEEN(SYSDATE, fecha_nacimiento) / 12)) AS edad_minima,
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
MAX(TRUNC(MONTHS_BETWEEN(SYSDATE, fecha_nacimiento) / 12)) AS edad_maxima  
FROM Estudiante;
```

-- Distribución de género (usando COUNT con CASE)

```
SELECT  
COUNT(CASE WHEN genero = 'M' THEN 1 END) AS total_masculino,  
COUNT(CASE WHEN genero = 'F' THEN 1 END) AS total_femenino,  
COUNT(CASE WHEN genero = 'O' THEN 1 END) AS total_otro,  
COUNT(*) AS total_general  
FROM Estudiante;
```

	TOTAL_MATRICULAS	MATRICULAS_CON_NOTA	PROMEDIO_NOTA	NOTA_MAXIMA	NOTA_MINIMA	Ti
1	6	6	7,92	9,25	6,25	

	TOTAL_ASIGNATURAS	PROMEDIO_CREDITOS	TOTAL_CREDITOS	PROMEDIO_HORAS_TOTALES
1	10	4,7	47	6,2

	EDAD_PROMEDIO	EDAD_MINIMA	EDAD_MAXIMA
1	22,6	21	25



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

Todas las filas recuperadas: 1 en 0,001 segundos				
	TOTAL_MASCULINO	TOTAL_FEMENINO	TOTAL_OTRO	TOTAL_GENERAL
1	10	6	0	16

GROUP BY Y HAVING

GROUP BY Básico

```
-- GROUP BY: Agrupar datos
-- =====

-- Contar estudiantes por carrera
SELECT
    codigo_carrera,
    COUNT(*) AS total_estudiantes
FROM Estudiante
GROUP BY codigo_carrera
ORDER BY total_estudiantes DESC;

-- Estudiantes por estado
SELECT
    estado,
    COUNT(*) AS total_estudiantes
FROM Estudiante
GROUP BY estado
ORDER BY total_estudiantes DESC;

-- Promedio de créditos por carrera
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
SELECT
    codigo_carrera,
    AVG(creditos_aprobados) AS promedio_creditos,
    MAX(creditos_aprobados) AS maximo_creditos,
    MIN(creditos_aprobados) AS minimo_creditos
FROM Estudiante
GROUP BY codigo_carrera
ORDER BY promedio_creditos DESC;
```

-- Asignaturas por nivel

```
SELECT
    nivel,
    COUNT(*) AS total_asignaturas,
    AVG(creditos) AS promedio_creditos
FROM Asignatura
GROUP BY nivel
ORDER BY nivel;
```

	NIVEL	TOTAL_ASIGNATURAS	PROMEDIO_CREDITOS
1	1	5	4,6
2	2	2	5
3	3	1	5
4	5	1	4
5	6	1	5



	ESTADO	TOTAL_ESTUDIANTES
1	INACTIVO	16

GROUP BY con Múltiples Columnas

```
SELECT
    codigo_carrera,
```




ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
estado,  
COUNT(*) AS total_estudiantes  
FROM Estudiante  
GROUP BY codigo_carrera, estado  
ORDER BY codigo_carrera, estado;
```

-- Matrículas por periodo y paralelo

```
SELECT  
    periodo,  
    paralelo,  
    COUNT(*) AS total_matriculas,  
    AVG(nota_final) AS promedio_notas  
FROM Matricula  
WHERE nota_final IS NOT NULL  
GROUP BY periodo, paralelo  
ORDER BY periodo, paralelo;
```

-- Estudiantes por carrera y género

```
SELECT  
    codigo_carrera,  
    genero,  
    COUNT(*) AS total,  
    ROUND(AVG(creditos_aprobados), 2) AS promedio_creditos  
FROM Estudiante  
GROUP BY codigo_carrera, genero  
ORDER BY codigo_carrera, genero;
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

Todas las Filas Recuperadas: 6 en 0,002 segundos

	CODIGO_CARRERA	GENERO	TOTAL	PROMEDIO_CREDITOS
1	ING-CIV	F	2	22,5
2	ING-CIV	M	1	0
3	ING-ELE	F	1	0
4	ING-ELE	M	2	102,5
5	ING-SIS	F	3	4
6	ING-SIS	M	7	6,71

Todas las Filas Recuperadas: 6 en 0,002 segundos

	CODIGO_CARRERA	GENERO	TOTAL	PROMEDIO_CREDITOS
1	ING-CIV	F	2	22,5
2	ING-CIV	M	1	0
3	ING-ELE	F	1	0
4	ING-ELE	M	2	102,5
5	ING-SIS	F	3	4
6	ING-SIS	M	7	6,71

Todas las Filas Recuperadas: 6 en 0,002 segundos

	CODIGO_CARRERA	GENERO	TOTAL	PROMEDIO_CREDITOS
1	ING-CIV	F	2	22,5
2	ING-CIV	M	1	0
3	ING-ELE	F	1	0
4	ING-ELE	M	2	102,5
5	ING-SIS	F	3	4
6	ING-SIS	M	7	6,71



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

HAVING - Filtrar Grupos

-- HAVING: Filtrar después de agrupar

-- =====

-- Carreras con más de 5 estudiantes

```
SELECT
    codigo_carrera,
    COUNT(*) AS total_estudiantes
FROM Estudiante
GROUP BY codigo_carrera
HAVING COUNT(*) > 5
ORDER BY total_estudiantes DESC;
```

-- Asignaturas por nivel con promedio de créditos ≥ 4

```
SELECT
    nivel,
    COUNT(*) AS total_asignaturas,
    AVG(creditos) AS promedio_creditos
FROM Asignatura
GROUP BY nivel
HAVING AVG(creditos)  $\geq 4$ 
ORDER BY nivel;
```

-- Estudiantes con promedio de notas ≥ 8.0

```
SELECT
    m.cedula_estudiante,
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
e.nombres || ' ' || e.apellidos AS nombre_completo,  
COUNT(*) AS total_materias,  
ROUND(AVG(m.nota_final), 2) AS promedio_notas  
FROM Matricula m  
JOIN Estudiante e ON m.cedula_estudiante = e.cedula  
WHERE m.nota_final IS NOT NULL  
GROUP BY m.cedula_estudiante, e.nombres, e.apellidos  
HAVING AVG(m.nota_final) >= 8.0  
ORDER BY promedio_notas DESC;
```

-- Diferencia WHERE vs HAVING

-- WHERE filtra ANTES de agrupar, HAVING filtra DESPUÉS de agrupar

-- Estudiantes activos, agrupados por carrera, con más de 3 estudiantes

```
SELECT  
    codigo_carrera,  
    COUNT(*) AS total_estudiantes  
FROM Estudiante  
WHERE estado = 'ACTIVO' -- WHERE filtra antes de GROUP BY  
GROUP BY codigo_carrera  
HAVING COUNT(*) > 3 -- HAVING filtra después de GROUP BY  
ORDER BY total_estudiantes DESC;
```

	CODIGO_CARRERA	TOTAL_ESTUDIANTES
1	ING-SIS	10



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

	NIVEL	TOTAL_ASIGNATURAS	PROMEDIO_CREDITOS
1	1	5	4,6
2	2	2	5
3	3	1	5
4	5	1	4
5	6	1	5

CEDULA_ESTUDIANTE	NOMBRE_COMPLETO	TOTAL_MATERIAS	PROMEDIO_NOTAS
1 1750700005	Elena Estrella Flores	1	9,25

SQL

Todas las Filas Recuperadas: 0 en 0,003 segundos

CODIGO_CARRERA	TOTAL_ESTUDIANTES
----------------	-------------------

Ejemplos Avanzados GROUP BY

-- EJEMPLOS AVANZADOS CON GROUP BY

-- =====

-- Top 5 asignaturas con más matrículas

```
SELECT
  a.codigo,
  a.nombre,
  COUNT(m.id_matricula) AS total_matriculas
FROM Asignatura a
LEFT JOIN Matricula m ON a.codigo = m.codigo_asignatura
GROUP BY a.codigo, a.nombre
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
HAVING COUNT(m.id_matricula) > 0  
ORDER BY total_matriculas DESC  
FETCH FIRST 5 ROWS ONLY;
```

-- Docentes con su carga académica

```
SELECT  
    d.cedula,  
    d.nombres || ' ' || d.apellidos AS nombre_docente,  
    d.tipo_contrato,  
    COUNT(DISTINCT m.codigo_asignatura) AS total_asignaturas_distintas,  
    COUNT(m.id_matricula) AS total_estudiantes_atendidos,  
    ROUND(AVG(m.nota_final), 2) AS promedio_notas_estudiantes  
FROM Docente d  
LEFT JOIN Matricula m ON d.cedula = m.cedula_docente  
GROUP BY d.cedula, d.nombres, d.apellidos, d.tipo_contrato  
ORDER BY total_estudiantes_atendidos DESC;
```

-- Análisis por periodo académico

```
SELECT  
    periodo,  
    COUNT(DISTINCT cedula_estudiante) AS estudiantes_unicos,  
    COUNT(DISTINCT codigo_asignatura) AS asignaturas_ofertadas,  
    COUNT(*) AS total_matriculas,  
    COUNT(CASE WHEN estado = 'APROBADO' THEN 1 END) AS aprobados,  
    COUNT(CASE WHEN estado = 'REPROBADO' THEN 1 END) AS reprobados,  
    ROUND(COUNT(CASE WHEN estado = 'APROBADO' THEN 1 END) * 100.0 /  
        NULLIF(COUNT(CASE WHEN estado IN ('APROBADO', 'REPROBADO') THEN 1 END), 0), 2) AS porcentaje_aprobacion
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

FROM Matricula
WHERE estado IN ('APROBADO', 'REPROBADO')
GROUP BY periodo
ORDER BY periodo DESC;

	CODIGO	NOMBRE	TOTAL_MATRICULAS
1	BD-501	Fundamentos de Bases de Datos	6
2	PRG-201	Programación Orientada a Objetos	2
3	ELE-101	Circuitos Eléctricos I	1
4	CIV-101	Dibujo Técnico	1

	CEDULA	NOMBRE_DOCENTE	TIPO_CONTRATO	TOTAL_ASIGNATURAS_DISTINTAS
1	1712345678	Carlos Alberto Pérez González	TIEMPO_COMPLETO	1
2	1734567890	Pedro José Martínez Torres	TIEMPO_COMPLETO	2
3	1745678901	Ana Lucía García Ramírez	TIEMPO_COMPLETO	1
4	1723456789	María Fernanda López Sánchez	MEDIO_TIEMPO	0
5	1767890123	Luis Vega	HORA_CLASE	0
6	1756789012	Roberto Carlos Sánchez Vera	HORA_CLASE	0

	PERIODO		APROBADOS	REPROBADOS	PORCENTAJE_APROBACION
1	2024-25	4 2 6	5	1	83,33

JOINS - UNIONES DE TABLAS

INNER JOIN



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

-- *INNER JOIN: Registros que coinciden en ambas tablas*

-- =====

-- *Estudiantes con su carrera*

```
SELECT
    e.cedula,
    e.nombres,
    e.apellidos,
    c.codigo AS codigo_carrera,
    c.nombre AS nombre_carrera
FROM Estudiante e
INNER JOIN Carrera c ON e.codigo_carrera = c.codigo
ORDER BY c.nombre, e.apellidos;
```

```
SELECT
    e.cedula,
    e.nombres,
    e.apellidos,
    c.codigo AS codigo_carrera,
    c.nombre AS nombre_carrera
FROM Estudiante e, carrera c
Where = c.codigo ORDER BY c.nombre, e.apellidos;
```

-- *Matrículas con información completa*

```
SELECT
    m.id_matricula,
    e.nombres || ' ' || e.apellidos AS estudiante,
```




ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

a.nombre AS asignatura,
d.nombres || ' ' || d.apellidos AS docente,
m.periodo,
m.nota_final,
m.estado

FROM Matricula m

INNER JOIN Estudiante e ON m.cedula_estudiante = e.cedula

INNER JOIN Asignatura a ON m.codigo_asignatura = a.codigo

INNER JOIN Docente d ON m.cedula_docente = d.cedula

WHERE m.periodo = '2024-2S'

ORDER BY m.id_matricula;

SELECT

m.id_matricula,
e.nombres || ' ' || e.apellidos AS estudiante, a.nombre AS asignatura,
d.nombres || ' ' || d.apellidos AS docente, m.periodo,
m.nota_final, m.estado

FROM Matricula m, Estudiante e, Asignatura a, Docente d

where m.cedula_estudiante = e.cedula

and m.codigo_asignatura = a.codigo

and m.cedula_docente = d.cedula

and m.periodo = '2024-2S'

ORDER BY m.id_matricula;

-- Asignaturas con sus prerrequisitos

SELECT

a1.codigo AS asignatura_codigo,
a1.nombre AS asignatura_nombre,
a2.codigo AS prerrequisito_codigo,
a2.nombre AS prerrequisito_nombre

FROM Prerrequisito p



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
INNER JOIN Asignatura a1 ON p.codigo_asignatura = a1.codigo  
INNER JOIN Asignatura a2 ON p.codigo_prerrequisito = a2.codigo  
ORDER BY a1.codigo;
```

-- JOIN con condiciones adicionales

```
SELECT  
    e.nombres,  
    e.apellidos,  
    c.nombre AS carrera,  
    e.creditos_aprobados  
FROM Estudiante e  
INNER JOIN Carrera c ON e.codigo_carrera = c.codigo  
WHERE e.creditos_aprobados >= 30  
AND e.estado = 'ACTIVO'  
ORDER BY e.creditos_aprobados DESC;
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

	⚡ CEDULA	⚡ NOMBRES	⚡ APELLIDOS	⚡ CODIGO_CARRERA	⚡ NOMBRE_CARRERA
1	1750345678	Carlos Andr,s	Castillo D;az	ING-CIV	Ingenier;a Civil
2	1750700003	Carla	Cruz Delgado	ING-CIV	Ingenier;a Civil
3	1750700007	Gabriela	González Herrera	ING-CIV	Ingenier;a Civil
4	1750700004	David	Diaz Espinoza	ING-ELE	Ingenier;a El,ctrice
5	1750456789	Ana Sof;a	Dom;nguez Escobar	ING-ELE	Ingenier;a El,ctrice
6	1750700008	Hugo	Hidalgo Ibarra	ING-ELE	Ingenier;a El,ctrice
7	1750700001	Andrea	Alvarez Brito	ING-SIS	Ingenier;a en Sistemas I
8	1750123456	Juan Pablo	Andrade Morales	ING-SIS	Ingenier;a en Sistemas I
9	1750234567	Mar;a Jos,	Ben;tez Castro	ING-SIS	Ingenier;a en Sistemas I
10	1750700002	Bryan	Bravo Cárdenas	ING-SIS	Ingenier;a en Sistemas I
11	1750567890	Pedro Luis	Espinoza Flores	ING-SIS	Ingenier;a en Sistemas I
12	1750700005	Elena	Estrella Flores	ING-SIS	Ingenier;a en Sistemas I
13	1750700006	Fernando	Fuentes García	ING-SIS	Ingenier;a en Sistemas I
14	1750444444	Test	Usuario	ING-SIS	Ingenier;a en Sistemas I
15	1750777777	Otro	Usuario	ING-SIS	Ingenier;a en Sistemas I
16	1750333333	Test	Usuario	ING-SIS	Ingenier;a en Sistemas I



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

❖	CEDULA	❖	NOMBRES	❖	APELLIDOS	❖	CODIGO_CARRERA	❖	NOMBRE_CARRERA
1	1750345678		Carlos Andr,s		Castillo D;az		ING-CIV		Ingenier;a Civil
2	1750700003		Carla		Cruz Delgado		ING-CIV		Ingenier;a Civil
3	1750700007		Gabriela		González Herrera		ING-CIV		Ingenier;a Civil
4	1750700004		David		Diaz Espinoza		ING-ELE		Ingenier;a El,ctrice
5	1750456789		Ana Sof;a		Dom;nguez Escobar		ING-ELE		Ingenier;a El,ctrice
6	1750700008		Hugo		Hidalgo Ibarra		ING-ELE		Ingenier;a El,ctrice
7	1750700001		Andrea		Alvarez Brito		ING-SIS		Ingenier;a en Sistemas I
8	1750123456		Juan Pablo		Andrade Morales		ING-SIS		Ingenier;a en Sistemas I
9	1750234567		Mar;a Jos,		Ben;tez Castro		ING-SIS		Ingenier;a en Sistemas I
10	1750700002		Bryan		Bravo Cárdenas		ING-SIS		Ingenier;a en Sistemas I
11	1750567890		Pedro Luis		Espinoza Flores		ING-SIS		Ingenier;a en Sistemas I
12	1750700005		Elena		Estrella Flores		ING-SIS		Ingenier;a en Sistemas I
13	1750700006		Fernando		Fuentes García		ING-SIS		Ingenier;a en Sistemas I
14	1750444444		Test		Usuario		ING-SIS		Ingenier;a en Sistemas I
15	1750777777		Otro		Usuario		ING-SIS		Ingenier;a en Sistemas I
16	1750333333		Test		Usuario		ING-SIS		Ingenier;a en Sistemas I

❖	I...	❖	ESTUDIANTE	❖	..	❖	DOCENTE	❖	..	❖	NOTA_FINAL	❖	ESTADO
1	1		Juan Pablo Andrade		Carlos Alberto P,rez Gonz lez		(null)		CURSANDO
2	2		Mar;a Jos, Ben;tez		Carlos Alberto P,rez Gonz lez		7,75		APROBADO
3	3		Pedro Luis Espinoza...		...		Carlos Alberto P,rez Gonz lez		(null)		CURSANDO
4	4		Carlos Andr,s Casti...		...		Pedro Jos, Mart;nez Torres		..		(null)		CURSANDO
5	5		Ana Sof;a Dom;nguez...		...		Ana Luc;a Garc;a Ram;rez		..		(null)		CURSANDO
6	8		Andrea Alvarez Brito		...		Carlos Alberto P,rez Gonz lez		8,75		APROBADO
7	9		Bryan Bravo Cárdenas		...		Carlos Alberto P,rez Gonz lez		7,25		APROBADO
8	10		Elena Estrella Flores		...		Carlos Alberto P,rez Gonz lez		9,25		APROBADO
9	11		Andrea Alvarez Brito		...		Pedro Jos, Mart;nez Torres		..		6,25		REPROBADO
10	12		Bryan Bravo Cárdenas		...		Pedro Jos, Mart;nez Torres		..		8,25		APROBADO



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

..	ESTUDIANTE	ASIGNATURA	DOCENTE	PERIODO	NO...	ESTADO
1	Juan Pablo Andrade...	Fundamentos d...	Carlos A...	2024-2S	(null)	CURSANDO
2	Marja Jos, Benítez...	Fundamentos d...	Carlos A...	2024-2S	7,75	APROBADO
3	Pedro Luis Espinoz...	Fundamentos d...	Carlos A...	2024-2S	(null)	CURSANDO
4	Carlos Andr,s Cast...	Dibujo T,cnico	Pedro Jo...	2024-2S	(null)	CURSANDO
5	Ana Sofja Domíngue...	Circuitos El,...	Ana Lucj...	2024-2S	(null)	CURSANDO
6	Andrea Alvarez Brito	Fundamentos d...	Carlos A...	2024-2S	8,75	APROBADO
7	Bryan Bravo Cárdenas	Fundamentos d...	Carlos A...	2024-2S	7,25	APROBADO
8	Elena Estrella Flores	Fundamentos d...	Carlos A...	2024-2S	9,25	APROBADO
9	Andrea Alvarez Brito	Programacién ...	Pedro Jo...	2024-2S	6,25	REPROBADO
10	Bryan Bravo Cárdenas	Programacién ...	Pedro Jo...	2024-2S	8,25	APROBADO

ASIGN...	ASIGNATURA_NOMBRE	PRERREQUISITO_CODIGO	PRERREQUISITO_NOMBRE
1 BD-601	Bases de Datos A...	EST-301	Estructura de Datos
2 BD-601	Bases de Datos A...	BD-501	Fundamentos de Bases de Datos
3 EST-301	Estructura de Datos	PRG-201	Programacién Orientada a Objetos
4 MAT-201	C lculo Integral	MAT-101	C lculo Diferencial
5 PRG-201	Programacién Ori...	PRG-101	Fundamentos de Programacién

LEFT JOIN (LEFT OUTER JOIN)

```
-- LEFT JOIN: Todos los registros de la tabla izquierda
-- =====

-- Todos los estudiantes con o sin matrículas
SELECT
    e.cedula,
    e.nombres,
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

e.apellidos,

COUNT(m.id_matricula) AS total_matriculas

FROM Estudiante e

LEFT JOIN Matricula m ON e.cedula = m.cedula_estudiante

GROUP BY e.cedula, e.nombres, e.apellidos

ORDER BY total_matriculas DESC, e.apellidos;

	⚡ CEDULA	⚡ NOMBRES	⚡ APELLIDOS	⚡ TOTAL_MATRICULAS	
1	1750700001	Andrea	Alvarez Brito	2	
2	1750700002	Bryan	Bravo Cárdenas	2	
3	1750123456	Juan Pablo	Andrade Morales	1	
4	1750234567	María Jos,	Benítez Castro	1	
5	1750345678	Carlos Andr,s	Castillo Díaz	1	
6	1750456789	Ana Sofía	Domínguez Escobar	1	
7	1750567890	Pedro Luis	Espinoza Flores	1	
8	1750700005	Elena	Estrella Flores	1	
9	1750700003	Carla	Cruz Delgado	0	
10	1750700004	David	Díaz Espinoza	0	
11	1750700006	Fernando	Fuentes García	0	
12	1750700007	Gabriela	González Herrera	0	
13	1750700008	Hugo	Hidalgo Ibarra	0	
14	1750333333	Test	Usuario	0	
15	1750444444	Test	Usuario	0	
16	1750777777	Otro	Usuario	0	

SELECT

e.cedula, e.nombres, e.apellidos,

COUNT(m.id_matricula) AS total_matriculas



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

FROM Estudiante e, Matricula m

where e.cedula = m.cedula_estudiante (+)

GROUP BY e.cedula, e.nombres, e.apellidos

ORDER BY total_matriculas DESC, e.apellidos;

	⚡ CEDULA	⚡ NOMBRES	⚡ APELLIDOS	⚡ TOTAL_MATRICULAS
1	1750700001	Andrea	Alvarez Brito	2
2	1750700002	Bryan	Bravo Cárdenas	2
3	1750123456	Juan Pablo	Andrade Morales	1
4	1750234567	María Jos,	Benítez Castro	1
5	1750345678	Carlos Andr,s	Castillo Díaz	1
6	1750456789	Ana Sofía	Domínguez Escobar	1
7	1750567890	Pedro Luis	Espinoza Flores	1
8	1750700005	Elena	Estrella Flores	1
9	1750700003	Carla	Cruz Delgado	0
10	1750700004	David	Díaz Espinoza	0
11	1750700006	Fernando	Fuentes García	0
12	1750700007	Gabriela	González Herrera	0
13	1750700008	Hugo	Hidalgo Ibarra	0
14	1750333333	Test	Usuario	0
15	1750444444	Test	Usuario	0
16	1750777777	Otro	Usuario	0

-- Estudiantes sin matrículas (usando LEFT JOIN y WHERE NULL)

SELECT

e.cedula,
e.nombres,
e.apellidos,
e.estado

FROM Estudiante e

LEFT JOIN Matricula m ON e.cedula = m.cedula_estudiante



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

WHERE m.id_matricula IS NULL

ORDER BY e.apellidos;

-- Todas las asignaturas con conteo de matrículas

SELECT

a.codigo,

a.nombre,

a.nivel,

a.creditos,

COUNT(m.id_matricula) AS total_matriculas

FROM Asignatura a

LEFT JOIN Matricula m ON a.codigo = m.codigo_asignatura

GROUP BY a.codigo, a.nombre, a.nivel, a.creditos

ORDER BY total_matriculas DESC, a.codigo;

-- Asignaturas sin matrículas

SELECT

a.codigo,

a.nombre,

a.nivel

FROM Asignatura a

LEFT JOIN Matricula m ON a.codigo = m.codigo_asignatura

WHERE m.id_matricula IS NULL

ORDER BY a.nivel, a.codigo;

-- Docentes con su carga (incluyendo docentes sin asignaciones)

SELECT



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
d.cedula,  
d.nombres || ' ' || d.apellidos AS docente,  
d.tipo_contrato,  
COUNT(m.id_matricula) AS estudiantes_asignados  
FROM Docente d  
LEFT JOIN Matricula m ON d.cedula = m.cedula_docente  
GROUP BY d.cedula, d.nombres, d.apellidos, d.tipo_contrato  
ORDER BY estudiantes_asignados DESC;
```

	⚡ CEDULA	⚡ NOMBRES	⚡ APELLIDOS	⚡ ESTADO
1	1750700003	Carla	Cruz Delgado	INACTIVO
2	1750700004	David	Diaz Espinoza	INACTIVO
3	1750700006	Fernando	Fuentes García	INACTIVO
4	1750700007	Gabriela	González Herrera	INACTIVO
5	1750700008	Hugo	Hidalgo Ibarra	INACTIVO
6	1750777777	Otro	Usuario	INACTIVO
7	1750444444	Test	Usuario	INACTIVO
8	1750333333	Test	Usuario	INACTIVO



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

	❖ CODIGO	❖ NOMBRE	❖ NIVEL	❖ CREDITOS	❖ TOTAL_MATRICULAS
1	BD-501	Fundamentos de Bases de Datos	5	4	6
2	PRG-201	Programación Orientada a Objetos	2	5	2
3	CIV-101	Dibujo Técnico	1	4	1
4	ELE-101	Circuitos Eléctricos I	1	5	1
5	BD-601	Bases de Datos Avanzadas	6	5	0
6	EST-301	Estructura de Datos	3	5	0
7	FIS-101	Física I	1	4	0
8	MAT-101	Cálculo Diferencial	1	5	0
9	MAT-201	Cálculo Integral	2	5	0
10	PRG-101	Fundamentos de Programación	1	5	0

	❖ CEDULA	❖ DOCENTE	❖ TIPO_CONTRATO	❖ ESTUDIANTES_ASIGNADOS
1	1712345678	Carlos Alberto Pérez González	TIEMPO_COMPLETO	6
2	1734567890	Pedro José Martínez Torres	TIEMPO_COMPLETO	3
3	1745678901	Ana Lucía García Ramírez	TIEMPO_COMPLETO	1
4	1767890123	Luis Vega	HORA_CLASE	0
5	1756789012	Roberto Carlos Sánchez Vera	HORA_CLASE	0
6	1723456789	María Fernanda López Sánchez	MEDIO_TIEMPO	0

RIGHT JOIN (RIGHT OUTER JOIN)

-- RIGHT JOIN: Todos los registros de la tabla derecha

-- =====

-- Todas las carreras con conteo de estudiantes

SELECT



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
c.codigo,  
c.nombre AS carrera,  
COUNT(e.cedula) AS total_estudiantes  
FROM Estudiante e  
RIGHT JOIN Carrera c ON e.codigo_carrera = c.codigo  
GROUP BY c.codigo, c.nombre  
ORDER BY total_estudiantes DESC;
```

-- Carreras sin estudiantes

```
SELECT e.cedula,  
c.codigo,  
c.nombre,  
c.facultad  
FROM Estudiante e  
RIGHT JOIN Carrera c ON e.codigo_carrera = c.codigo  
WHERE e.cedula IS NULL;
```

-- NOTA: RIGHT JOIN es menos común que LEFT JOIN

-- Normalmente se usa LEFT JOIN reorganizando las tablas

-- Estos dos son equivalentes:

-- Con RIGHT JOIN

```
SELECT c.nombre, COUNT(e.cedula)  
FROM Estudiante e  
RIGHT JOIN Carrera c ON e.codigo_carrera = c.codigo
```

-- Con LEFT JOIN (más común)



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
SELECT c.nombre, COUNT(e.cedula)
FROM Carrera c
LEFT JOIN Estudiante e ON c.codigo = e.codigo_carrera
GROUP BY c.nombre;
```

FULL OUTER JOIN

-- *FULL OUTER JOIN: Todos los registros de ambas tablas*

-- =====

-- *Ejemplo conceptual (menos común en práctica)*

-- *Todos los estudiantes y todas las carreras*

```
SELECT
    e.cedula,
    e.nombres,
    e.apellidos,
    c.codigo AS codigo_carrera,
    c.nombre AS nombre_carrera
FROM Estudiante e
FULL OUTER JOIN Carrera c ON e.codigo_carrera = c.codigo
ORDER BY c.nombre, e.apellidos;
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

	⚡ CEDULA	⚡ NOMBRES	⚡ APELLIDOS	⚡ CODIGO_...	⚡ NOMBRE_CARRERA
1	1750345678	Carlos Andr,s	Castillo Díaz	ING-CIV	Ingenier;a Civil
2	1750700003	Carla	Cruz Delgado	ING-CIV	Ingenier;a Civil
3	1750700007	Gabriela	González Herrera	ING-CIV	Ingenier;a Civil
4	1750700004	David	Diaz Espinoza	ING-ELE	Ingenier;a El,ctrice
5	1750456789	Ana Sof;a	Domínguez Escobar	ING-ELE	Ingenier;a El,ctrice
6	1750700008	Hugo	Hidalgo Ibarra	ING-ELE	Ingenier;a El,ctrice
7	1750700001	Andrea	Alvarez Brito	ING-SIS	Ingenier;a en Sistemas Inform
8	1750123456	Juan Pablo	Andrade Morales	ING-SIS	Ingenier;a en Sistemas Inform
9	1750234567	Mar;a Jos,	Benítez Castro	ING-SIS	Ingenier;a en Sistemas Inform
10	1750700002	Bryan	Bravo Cárdenas	ING-SIS	Ingenier;a en Sistemas Inform
11	1750567890	Pedro Luis	Espinoza Flores	ING-SIS	Ingenier;a en Sistemas Inform
12	1750700005	Elena	Estrella Flores	ING-SIS	Ingenier;a en Sistemas Inform
13	1750700006	Fernando	Fuentes García	ING-SIS	Ingenier;a en Sistemas Inform
14	1750444444	Test	Usuario	ING-SIS	Ingenier;a en Sistemas Inform
15	1750777777	Otro	Usuario	ING-SIS	Ingenier;a en Sistemas Inform
16	1750333333	Test	Usuario	ING-SIS	Ingenier;a en Sistemas Inform
17	(null)	(null)	(null)	ING-GEO	Ingenier;a Geológica
18	(null)	(null)	(null)	ING-IND	Ingenier;a Industrial
19	(null)	(null)	(null)	ING-MEC	Ingenier;a Mec nica
20	(null)	(null)	(null)	ING-QUI	Ingenier;a Química
21	(null)	(null)	(null)	ING-AMB	Ingeniería Ambiental

-- Encontrar estudiantes sin carrera Y carreras sin estudiantes

```
SELECT
  e.cedula,
  e.nombres,
  c.codigo AS codigo_carrera,
  c.nombre AS nombre_carrera,
CASE
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
    WHEN e.cedula IS NULL THEN 'Carrera sin estudiantes'
    WHEN c.codigo IS NULL THEN 'Estudiante sin carrera'
    ELSE 'Ambos tienen match'
END AS estado_relacion
FROM Estudiante e
FULL OUTER JOIN Carrera c ON e.codigo_carrera = c.codigo
WHERE e.cedula IS NULL OR c.codigo IS NULL;
```

CROSS JOIN (Producto Cartesiano)

```
-- CROSS JOIN: Todas las combinaciones posibles
-- =====
```

```
-- ADVERTENCIA: Genera muchas filas ( $n \times m$ )
```

```
-- Ejemplo: Combinar todas las asignaturas con todos los docentes
```

```
-- (Útil para generar horarios posibles)
```

```
SELECT
    a.codigo AS codigo_asignatura,
    a.nombre AS asignatura,
    d.cedula,
    d.nombres || ' ' || d.apellidos AS docente
FROM Asignatura a
CROSS JOIN Docente d
WHERE a.codigo_carrera = 'ING-SIS'
AND ROWNUM <= 20; -- Limitar resultados
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

-- Combinaciones de periodos y paralelos

SELECT

periodo,

paralelo

FROM (SELECT '2024-1S' AS periodo FROM DUAL UNION ALL SELECT '2024-2S' FROM DUAL)

CROSS JOIN (SELECT 'A' AS paralelo FROM DUAL UNION ALL SELECT 'B' FROM DUAL);

	❖ CODIGO_ASIGNATURA	❖ ASIGNATURA	❖ CEDULA	❖ DOCENTE
1	MAT-101	C lculo Diferencial	1712345678	Carlos Alberto P,rez
2	FIS-101	Física I	1712345678	Carlos Alberto P,rez
3	PRG-101	Fundamentos de Programaciën	1712345678	Carlos Alberto P,rez
4	MAT-201	C lculo Integral	1712345678	Carlos Alberto P,rez
5	PRG-201	Programaciën Orientada a Objetos	1712345678	Carlos Alberto P,rez
6	EST-301	Estructura de Datos	1712345678	Carlos Alberto P,rez
7	BD-501	Fundamentos de Bases de Datos	1712345678	Carlos Alberto P,rez
8	BD-601	Bases de Datos Avanzadas	1712345678	Carlos Alberto P,rez
9	MAT-101	C lculo Diferencial	1723456789	María Fernanda Lopez
10	FIS-101	Física I	1723456789	María Fernanda Lopez
11	PRG-101	Fundamentos de Programaciën	1723456789	María Fernanda Lopez
12	MAT-201	C lculo Integral	1723456789	María Fernanda Lopez
13	PRG-201	Programaciën Orientada a Objetos	1723456789	María Fernanda Lopez
14	EST-301	Estructura de Datos	1723456789	María Fernanda Lopez
15	BD-501	Fundamentos de Bases de Datos	1723456789	María Fernanda Lopez
16	BD-601	Bases de Datos Avanzadas	1723456789	María Fernanda Lopez
17	MAT-101	C lculo Diferencial	1734567890	Pedro Jos, Martínez
18	FIS-101	Física I	1734567890	Pedro Jos, Martínez
19	PRG-101	Fundamentos de Programaciën	1734567890	Pedro Jos, Martínez
20	MAT-201	C lculo Integral	1734567890	Pedro Jos, Martínez



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

	PERIODO	PARALELO
1	2024-1S	A
2	2024-1S	B
3	2024-2S	A
4	2024-2S	B

SELF JOIN

```
-- =====  
-- SELF JOIN: Unir tabla consigo misma  
-- =====
```

-- Encontrar pares de estudiantes de la misma carrera

```
SELECT  
    e1.nombres || ' ' || e1.apellidos AS estudiante1,  
    e2.nombres || ' ' || e2.apellidos AS estudiante2,  
    e1.codigo_carrera  
FROM Estudiante e1  
JOIN Estudiante e2 ON e1.codigo_carrera = e2.codigo_carrera  
WHERE e1.cedula < e2.cedula -- Evitar duplicados y comparar consigo mismo  
AND ROWNUM <= 10;
```

-- Asignaturas que comparten el mismo prerrequisito

```
SELECT  
    p1.codigo_asignatura AS asignatura1,  
    p2.codigo_asignatura AS asignatura2,  
    p1.codigo_prerrequisito AS prerrequisito_comun
```




ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
FROM Prerrequisito p1
JOIN Prerrequisito p2 ON p1.codigo_prerrequisito = p2.codigo_prerrequisito
WHERE p1.codigo_asignatura < p2.codigo_asignatura;
```

Resultado de la Consulta x			
Todas las Filas Recuperadas: 10 en 0,005 segundos			
ESTUDIANTE1	ESTUDIANTE2	CODIGO_CARRERA	
1 Andrea Alvarez Brito	Bryan Bravo Cárdenas	ING-SIS	
2 Andrea Alvarez Brito	Elena Estrella Flores	ING-SIS	
3 Andrea Alvarez Brito	Fernando Fuentes García	ING-SIS	
4 Andrea Alvarez Brito	Otro Usuario	ING-SIS	
5 Bryan Bravo Cárdenas	Elena Estrella Flores	ING-SIS	
6 Bryan Bravo Cárdenas	Fernando Fuentes García	ING-SIS	
7 Bryan Bravo Cárdenas	Otro Usuario	ING-SIS	
8 Carla Cruz Delgado	Gabriela González Herrera	ING-CIV	
9 David Diaz Espinoza	Hugo Hidalgo Ibarra	ING-ELE	
10 Elena Estrella Flores	Fernando Fuentes García	ING-SIS	

Ejemplos Complejos con Múltiples JOINS

```
-- CONSULTAS COMPLEJAS CON MÚLTIPLES JOINS
-- =====

-- Reporte completo de matrículas
SELECT
  m.id_matricula,
  e.cedula,
  e.nombres || ' ' || e.apellidos AS estudiante,
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

c.nombre AS carrera,
a.codigo AS codigo_asignatura,
a.nombre AS asignatura,
a.creditos,
d.nombres || ' ' || d.apellidos AS docente,
m.periodo,
m.paralelo,
m.nota_parcial1,
m.nota_parcial2,
m.nota_final,
m.estado,

CASE

WHEN m.nota_final >= 9.0 THEN 'Excelente'
WHEN m.nota_final >= 8.0 THEN 'Muy Bueno'
WHEN m.nota_final >= 7.0 THEN 'Bueno'
ELSE 'Insuficiente'

END AS calificacion_cualitativa

FROM Matricula m

INNER JOIN Estudiante e ON m.cedula_estudiante = e.cedula

INNER JOIN Carrera c ON e.codigo_carrera = c.codigo

INNER JOIN Asignatura a ON m.codigo_asignatura = a.codigo

INNER JOIN Docente d ON m.cedula_docente = d.cedula

WHERE m.periodo = '2024-2S'

ORDER BY m.id_matricula;

-- Estudiantes con asignaturas pendientes por prerrequisitos

SELECT DISTINCT



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
e.cedula,  
e.nombres || ' ' || e.apellidos AS estudiante,  
a.codigo AS asignatura_bloqueada,  
a.nombre AS nombre_asignatura,  
apre.codigo AS prerequisito_codigo,  
apre.nombre AS prerequisito_nombre  
FROM Estudiante e  
CROSS JOIN Asignatura a  
INNER JOIN Prerrequisito p ON a.codigo = p.codigo_asignatura  
INNER JOIN Asignatura apre ON p.codigo_prerrequisito = apre.codigo  
LEFT JOIN Matricula m ON e.cedula = m.cedula_estudiante  
    AND apre.codigo = m.codigo_asignatura  
    AND m.estado = 'APROBADO'  
WHERE e.codigo_carrera = a.codigo_carrera  
AND m.id_matricula IS NULL  
AND ROWNUM <= 20  
ORDER BY estudiante;  
  
-- Top estudiantes por carrera  
SELECT  
    c.nombre AS carrera,  
    e.nombres || ' ' || e.apellidos AS estudiante,  
    e.creditos_aprobados,  
    ROUND(AVG(m.nota_final), 2) AS promedio_general,  
    COUNT(CASE WHEN m.estado = 'APROBADO' THEN 1 END) AS materias_aprobadas  
FROM Carrera c  
INNER JOIN Estudiante e ON c.codigo = e.codigo_carrera
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
LEFT JOIN Matricula m ON e.cedula = m.cedula_estudiante AND m.nota_final IS NOT NULL
WHERE e.estado = 'ACTIVO'
GROUP BY c.nombre, e.cedula, e.nombres, e.apellidos, e.creditos_aprobados
HAVING COUNT(CASE WHEN m.estado = 'APROBADO' THEN 1 END) > 0
ORDER BY c.nombre, promedio_general DESC, e.creditos_aprobados DESC;
```

	ID_MATRICULA	CEDULA	ESTUDIANTE	CARRERA
1	1	1750123456	Juan Pablo Andrade Morales	Ingeniería en Sistemas Inform t
2	2	1750234567	María Jos, Benítez Castro	Ingeniería en Sistemas Inform t
3	3	1750567890	Pedro Luis Espinoza Flores	Ingeniería en Sistemas Inform t
4	4	1750345678	Carlos Andr,s Castillo Díaz	Ingeniería Civil
5	5	1750456789	Ana Sofía Domínguez Escobar	Ingeniería El,ctrica
6	8	1750700001	Andrea Alvarez Brito	Ingeniería en Sistemas Inform t
7	9	1750700002	Bryan Bravo Cárdenas	Ingeniería en Sistemas Inform t
8	10	1750700005	Elena Estrella Flores	Ingeniería en Sistemas Inform t
9	11	1750700001	Andrea Alvarez Brito	Ingeniería en Sistemas Inform t
10	12	1750700002	Bryan Bravo Cárdenas	Ingeniería en Sistemas Inform t



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

	⚡ CEDULA	⚡ ESTUDIANTE	⚡ ASIGNATURA_BLOQUEADA	⚡ NOMBRE_ASIGNATURA
1	1750700001	Andrea Alvarez Brito	PRG-201	Programacion Orientada a
2	1750700001	Andrea Alvarez Brito	MAT-201	C lculo Integral
3	1750700002	Bryan Bravo Cárdenas	MAT-201	C lculo Integral
4	1750700002	Bryan Bravo Cárdenas	PRG-201	Programacion Orientada a
5	1750700005	Elena Estrella Flores	PRG-201	Programacion Orientada a
6	1750700005	Elena Estrella Flores	MAT-201	C lculo Integral
7	1750700006	Fernando Fuentes García	MAT-201	C lculo Integral
8	1750700006	Fernando Fuentes García	PRG-201	Programacion Orientada a
9	1750123456	Juan Pablo Andrade Morales	PRG-201	Programacion Orientada a
10	1750123456	Juan Pablo Andrade Morales	MAT-201	C lculo Integral
11	1750234567	María Jos, Benítez Castro	PRG-201	Programacion Orientada a
12	1750234567	María Jos, Benítez Castro	MAT-201	C lculo Integral
13	1750777777	Otro Usuario	MAT-201	C lculo Integral
14	1750777777	Otro Usuario	PRG-201	Programacion Orientada a
15	1750567890	Pedro Luis Espinoza Flores	MAT-201	C lculo Integral
16	1750567890	Pedro Luis Espinoza Flores	PRG-201	Programacion Orientada a
17	1750333333	Test Usuario	PRG-201	Programacion Orientada a
18	1750444444	Test Usuario	PRG-201	Programacion Orientada a
19	1750444444	Test Usuario	MAT-201	C lculo Integral
20	1750333333	Test Usuario	MAT-201	C lculo Integral

SUBCONSULTAS (SUBQUERIES)

Subconsultas en WHERE

-- SUBCONSULTAS EN WHERE



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

-- =====

-- *Estudiantes de la carrera con más estudiantes matriculados*

```
SELECT nombres, apellidos, codigo_carrera
FROM Estudiante
WHERE codigo_carrera = (
    SELECT codigo_carrera
    FROM Estudiante
    GROUP BY codigo_carrera
    ORDER BY COUNT(*) DESC
    FETCH FIRST 1 ROW ONLY
);
```

-- *Estudiantes con más créditos que el promedio*

```
SELECT nombres, apellidos, credits_aprobados
FROM Estudiante
WHERE credits_aprobados > (
    SELECT AVG(credits_aprobados)
    FROM Estudiante
)
ORDER BY credits_aprobados DESC;
```

-- *Asignaturas que son prerrequisito de otras*

```
SELECT codigo, nombre
FROM Asignatura
WHERE codigo IN (
    SELECT codigo_prerrequisito
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
FROM Prerrequisito
)
ORDER BY codigo;
```

-- Estudiantes que NO tienen matrículas

```
SELECT cedula, nombres, apellidos
FROM Estudiante
WHERE cedula NOT IN (
    SELECT DISTINCT cedula_estudiante
    FROM Matricula
);
```

-- Docentes que imparten asignaturas de nivel >= 5

```
SELECT DISTINCT d.cedula, d.nombres, d.apellidos
FROM Docente d
WHERE d.cedula IN (
    SELECT DISTINCT m.cedula_docente
    FROM Matricula m
    INNER JOIN Asignatura a ON m.codigo_asignatura = a.codigo
    WHERE a.nivel >= 5
);
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

	⚡ NOMBRES	⚡ APELLIDOS	⚡ CODIGO_CARRERA
1	Andrea	Alvarez Brito	ING-SIS
2	Bryan	Bravo Cárdenas	ING-SIS
3	Elena	Estrella Flores	ING-SIS
4	Fernando	Fuentes García	ING-SIS
5	Juan Pablo	Andrade Morales	ING-SIS
6	María Jos,	Benítez Castro	ING-SIS
7	Pedro Luis	Espinoza Flores	ING-SIS
8	Otro	Usuario	ING-SIS
9	Test	Usuario	ING-SIS
10	Test	Usuario	ING-SIS

	⚡ NOMBRES	⚡ APELLIDOS	⚡ CREDITOS_APROBADOS
1	Hugo	Hidalgo Ibarra	180
2	Gabriela	González Herrera	35
3	Juan Pablo	Andrade Morales	28
4	David	Diaz Espinoza	25

	⚡ CODIGO	⚡ NOMBRE
1	BD-501	Fundamentos de Bases de Datos
2	EST-301	Estructura de Datos
3	MAT-101	C lculo Diferencial
4	PRG-101	Fundamentos de Programaci�n
5	PRG-201	Programaci�n Orientada a Objetos



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

	↕ CEDULA	↕ NOMBRES	↕ APELLIDOS	
1	1750700003	Carla	Cruz Delgado	
2	1750700004	David	Diaz Espinoza	
3	1750700006	Fernando	Fuentes García	
4	1750700007	Gabriela	González Herrera	
5	1750700008	Hugo	Hidalgo Ibarra	
6	1750777777	Otro	Usuario	
7	1750444444	Test	Usuario	
8	1750333333	Test	Usuario	

↕ CEDULA	↕ NOMBRES	↕ APELLIDOS	
1 1712345678	Carlos Alberto P,rez Gonz lez		

Subconsultas en SELECT

-- SUBCONSULTAS EN SELECT (columnas calculadas)

-- =====

-- Estudiantes con total de matrículas

SELECT

e.cedula,

e.nombres,

e.apellidos,

(SELECT COUNT(*)

FROM Matricula m

WHERE m.cedula_estudiante = e.cedula) AS total_matriculas,



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
(SELECT COUNT(*)  
FROM Matricula m  
WHERE m.cedula_estudiante = e.cedula  
AND m.estado = 'APROBADO') AS materias_aprobadas  
FROM Estudiante e  
ORDER BY total_matriculas DESC;
```

-- Carreras con estadísticas

```
SELECT  
    c.codigo,  
    c.nombre,  
    (SELECT COUNT(*)  
     FROM Estudiante e  
     WHERE e.codigo_carrera = c.codigo) AS total_estudiantes,  
    (SELECT COUNT(*)  
     FROM Asignatura a  
     WHERE a.codigo_carrera = c.codigo) AS total_asignaturas,  
    (SELECT AVG(creditos_aprobados)  
     FROM Estudiante e  
     WHERE e.codigo_carrera = c.codigo) AS promedio_creditos  
FROM Carrera c  
ORDER BY total_estudiantes DESC;
```

-- Asignaturas con información de matrículas

```
SELECT  
    a.codigo,  
    a.nombre,
```



**ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS**

```
a.creditos,  
(SELECT COUNT(*)  
FROM Matricula m  
WHERE m.codigo_asignatura = a.codigo) AS veces_matriculada,  
(SELECT AVG(nota_final)  
FROM Matricula m  
WHERE m.codigo_asignatura = a.codigo  
AND m.nota_final IS NOT NULL) AS promedio_notas  
FROM Asignatura a  
WHERE a.codigo_carrera = 'ING-SIS'  
ORDER BY veces_matriculada DESC;
```

[illegible]



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

	❖ CODIGO	❖ NOMBRE	❖ CREDITOS	❖ VECES_MATRICULADA	❖ PROMEDIO_NO
1	BD-501	Fundamentos de Bases de Datos	4	6	8
2	PRG-201	Programación Orientada a Objetos	5	2	7
3	FIS-101	Física I	4	0	(nu
4	PRG-101	Fundamentos de Programación	5	0	(nu
5	MAT-201	Cálculo Integral	5	0	(nu
6	MAT-101	Cálculo Diferencial	5	0	(nu
7	BD-601	Bases de Datos Avanzadas	5	0	(nu
8	EST-301	Estructura de Datos	5	0	(nu

Subconsultas en FROM (Derived Tables)

```
-- SUBCONSULTAS EN FROM (tablas derivadas)
-- =====
```

-- Estadísticas por carrera usando tabla derivada

```
SELECT
  stats.codigo_carrera,
  c.nombre AS nombre_carrera,
  stats.total_estudiantes,
  stats.promedio_creditos
FROM (
  SELECT
    codigo_carrera,
    COUNT(*) AS total_estudiantes,
    ROUND(AVG(creditos_aprobados), 2) AS promedio_creditos
  FROM Estudiante
  WHERE estado = 'ACTIVO'
  GROUP BY codigo_carrera
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
) stats  
INNER JOIN Carrera c ON stats.codigo_carrera = c.codigo  
ORDER BY stats.total_estudiantes DESC;
```

-- Top 5 estudiantes por promedio

```
SELECT  
    notas.cedula,  
    e.nombres || ' ' || e.apellidos AS estudiante,  
    notas.promedio,  
    notas.total_materias  
FROM (  
    SELECT  
        cedula_estudiante AS cedula,  
        ROUND(AVG(nota_final), 2) AS promedio,  
        COUNT(*) AS total_materias  
    FROM Matricula  
    WHERE nota_final IS NOT NULL  
    GROUP BY cedula_estudiante  
    HAVING COUNT(*) >= 3  
) notas  
INNER JOIN Estudiante e ON notas.cedula = e.cedula  
ORDER BY notas.promedio DESC  
FETCH FIRST 5 ROWS ONLY;
```

-- Asignaturas populares (con más de N matrículas)

```
SELECT  
    popular.codigo,
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
a.nombre,  
popular.total_matriculas,  
popular.promedio_nota  
FROM (  
  SELECT  
    codigo_asignatura AS codigo,  
    COUNT(*) AS total_matriculas,  
    ROUND(AVG(nota_final), 2) AS promedio_nota  
  FROM Matricula  
  WHERE nota_final IS NOT NULL  
  GROUP BY codigo_asignatura  
  HAVING COUNT(*) >= 3  
) popular  
INNER JOIN Asignatura a ON popular.codigo = a.codigo  
ORDER BY popular.total_matriculas DESC;
```

	CO...	NOMBRE	TOTAL_MATRICULAS	PROMEDIO_NOTA
1	BD-501	Fundamentos de Bases de Datos	4	8,25

Subconsultas Correlacionadas

```
-- SUBCONSULTAS CORRELACIONADAS  
-- =====  
  
-- Estudiantes con nota superior al promedio de su carrera  
SELECT  
  e.cedula,
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
e.nombres,  
e.apellidos,  
e.codigo_carrera,  
(SELECT ROUND(AVG(m2.nota_final), 2)  
FROM Matricula m2  
INNER JOIN Estudiante e2 ON m2.cedula_estudiante = e2.cedula  
WHERE e2.codigo_carrera = e.codigo_carrera  
AND m2.nota_final IS NOT NULL) AS promedio_carrera,  
(SELECT ROUND(AVG(m1.nota_final), 2)  
FROM Matricula m1  
WHERE m1.cedula_estudiante = e.cedula  
AND m1.nota_final IS NOT NULL) AS promedio_estudiante  
FROM Estudiante e  
WHERE EXISTS (  
    SELECT 1 FROM Matricula m WHERE m.cedula_estudiante = e.cedula  
)  
AND (SELECT AVG(m1.nota_final)  
    FROM Matricula m1  
    WHERE m1.cedula_estudiante = e.cedula  
    AND m1.nota_final IS NOT NULL) >  
(SELECT AVG(m2.nota_final)  
    FROM Matricula m2  
    INNER JOIN Estudiante e2 ON m2.cedula_estudiante = e2.cedula  
    WHERE e2.codigo_carrera = e.codigo_carrera  
  
    AND m2.nota_final IS NOT NULL)  
ORDER BY promedio_estudiante DESC;
```




ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

-- Asignaturas con nota promedio superior a la general

```
SELECT
  a.codigo,
  a.nombre,
  (SELECT ROUND(AVG(nota_final), 2)
   FROM Matricula
   WHERE codigo_asignatura = a.codigo
   AND nota_final IS NOT NULL) AS promedio_asignatura
FROM Asignatura a
WHERE (SELECT AVG(nota_final)
       FROM Matricula
       WHERE codigo_asignatura = a.codigo
       AND nota_final IS NOT NULL) >
      (SELECT AVG(nota_final)
       FROM Matricula
       WHERE nota_final IS NOT NULL)
ORDER BY promedio_asignatura DESC;
```

Subconsultas con ANY, ALL

-- ANY, SOME, ALL con subconsultas

-- =====

-- Estudiantes con más créditos que CUALQUIER estudiante de ING-CIV

```
SELECT nombres, apellidos, codigo_carrera, creditos_aprobados
FROM Estudiante
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
WHERE credits_aprobados > ANY (  
    SELECT credits_aprobados  
    FROM Estudiante  
    WHERE codigo_carrera = 'ING-CIV'  
)  
AND codigo_carrera != 'ING-CIV'  
ORDER BY credits_aprobados DESC;
```

-- Estudiantes con más créditos que TODOS los estudiantes de ING-CIV

```
SELECT nombres, apellidos, codigo_carrera, credits_aprobados  
FROM Estudiante  
WHERE credits_aprobados > ALL (  
    SELECT credits_aprobados  
    FROM Estudiante  
    WHERE codigo_carrera = 'ING-CIV'  
)  
ORDER BY credits_aprobados DESC;
```

-- Asignaturas con créditos >= a cualquier asignatura de nivel 1

```
SELECT codigo, nombre, nivel, credits  
FROM Asignatura  
WHERE credits >= ANY (  
    SELECT credits  
    FROM Asignatura  
    WHERE nivel = 1  
)  
AND nivel > 1
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

ORDER BY nivel, credits DESC;

OPERADORES DE CONJUNTOS

UNION y UNION ALL

-- UNION: Combinar resultados (sin duplicados)
-- =====

-- Todos los emails de estudiantes y docentes

```
SELECT email, 'Estudiante' AS tipo
FROM Estudiante
UNION
SELECT email, 'Docente' AS tipo
FROM Docente
ORDER BY email;
```

-- UNION ALL: Incluir duplicados (más rápido)

```
SELECT email
FROM Estudiante
UNION ALL
SELECT email
FROM Docente;
```

-- Combinar diferentes consultas con la misma estructura

```
SELECT cedula AS identificacion, nombres, apellidos, 'Estudiante' AS rol
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
FROM Estudiante
WHERE estado = 'ACTIVO'
UNION
SELECT cedula, nombres, apellidos, 'Docente' AS rol
FROM Docente
WHERE tipo_contrato = 'TIEMPO_COMPLETO'
ORDER BY apellidos;
```

-- Lista de todos los códigos (carreras y asignaturas)

```
SELECT codigo AS codigo, nombre AS descripcion, 'Carrera' AS tipo
FROM Carrera
UNION
SELECT codigo, nombre, 'Asignatura' AS tipo
FROM Asignatura
ORDER BY tipo, codigo;
```

INTERSECT

-- INTERSECT: Elementos comunes entre conjuntos

-- =====

-- Estudiantes que también son docentes (poco común, pero para ejemplo)

```
SELECT cedula, nombres, apellidos
FROM Estudiante
INTERSECT
SELECT cedula, nombres, apellidos
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

FROM Docente;

-- Códigos que aparecen tanto en Asignatura como en Prerrequisito

SELECT codigo

FROM Asignatura

INTERSECT

SELECT codigo_asignatura

FROM Prerrequisito;

-- Estudiantes matriculados en ambos periodos

SELECT cedula_estudiante AS cedula

FROM Matricula

WHERE periodo = '2024-1S'

INTERSECT

SELECT cedula_estudiante

FROM Matricula

WHERE periodo = '2024-2S';

MINUS (EXCEPT en otros SGBD)

-- MINUS: Elementos en el primer conjunto pero no en el segundo

-- =====

-- Estudiantes que NO tienen matrículas

SELECT cedula, nombres, apellidos

FROM Estudiante

MINUS



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
SELECT DISTINCT e.cedula, e.nombres, e.apellidos  
FROM Estudiante e  
INNER JOIN Matricula m ON e.cedula = m.cedula_estudiante;
```

-- Asignaturas que NO son prerrequisito de ninguna otra

```
SELECT codigo, nombre  
FROM Asignatura  
MINUS  
SELECT a.codigo, a.nombre  
FROM Asignatura a  
INNER JOIN Prerrequisito p ON a.codigo = p.codigo_prerrequisito;
```

-- Docentes que NO han impartido clases

```
SELECT cedula, nombres, apellidos  
FROM Docente  
MINUS  
SELECT DISTINCT d.cedula, d.nombres, d.apellidos  
FROM Docente d  
INNER JOIN Matricula m ON d.cedula = m.cedula_docente;
```

-- Estudiantes de ING-SIS que NO están matriculados en BD-501

```
SELECT cedula, nombres, apellidos  
FROM Estudiante  
WHERE codigo_carrera = 'ING-SIS'  
MINUS  
SELECT e.cedula, e.nombres, e.apellidos  
FROM Estudiante e
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
INNER JOIN Matricula m ON e.cedula = m.cedula_estudiante
WHERE m.codigo_asignatura = 'BD-501';
```

Combinación de Operadores

```
-- COMBINACIÓN DE OPERADORES DE CONJUNTOS
```

```
-- =====
```

```
-- Estudiantes activos o graduados, pero no retirados
```

```
(
  SELECT cedula, nombres, apellidos, estado
  FROM Estudiante
  WHERE estado = 'ACTIVO'
  UNION
  SELECT cedula, nombres, apellidos, estado
  FROM Estudiante
  WHERE estado = 'GRADUADO'
)
MINUS
SELECT cedula, nombres, apellidos, estado
FROM Estudiante
WHERE estado = 'RETIRADO'
ORDER BY apellidos;
```

```
-- Todos los involucrados en el sistema (estudiantes, docentes)
```

```
-- que tienen email @epn.edu.ec
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
SELECT 'E-' || cedula AS id, nombres, apellidos, email, 'Estudiante' AS tipo
FROM Estudiante
WHERE email LIKE '%@epn.edu.ec'
UNION
SELECT 'D-' || cedula AS id, nombres, apellidos, email, 'Docente' AS tipo
FROM Docente
WHERE email LIKE '%@epn.edu.ec'
ORDER BY apellidos;
```

CONSULTAS AVANZADAS

WITH (Common Table Expressions - CTE)

```
-- WITH: Consultas temporales nombradas (CTE)
-- =====

-- Ejemplo básico de CTE
WITH PromediosCarrera AS (
    SELECT
        codigo_carrera,
        ROUND(AVG(creditos_aprobados), 2) AS promedio_creditos,
        COUNT(*) AS total_estudiantes
    FROM Estudiante
    GROUP BY codigo_carrera
)
SELECT
```




ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
c.nombre AS carrera,  
pc.total_estudiantes,  
pc.promedio_creditos  
FROM PromediosCarrera pc  
INNER JOIN Carrera c ON pc.codigo_carrera = c.codigo  
WHERE pc.total_estudiantes >= 3  
ORDER BY pc.promedio_creditos DESC;
```

-- Múltiples CTEs

WITH

```
EstudiantesActivos AS (  
    SELECT cedula, nombres, apellidos, codigo_carrera, creditos_aprobados  
    FROM Estudiante  
    WHERE estado = 'ACTIVO'
```

),

```
MatriculasPorEstudiante AS (  
    SELECT  
        cedula_estudiante,  
        COUNT(*) AS total_matriculas,  
        ROUND(AVG(nota_final), 2) AS promedio_notas  
    FROM Matricula  
    WHERE nota_final IS NOT NULL  
    GROUP BY cedula_estudiante
```

```
)  
SELECT  
    ea.cedula,  
    ea.nombres || ' ' || ea.apellidos AS estudiante,
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
ea.codigo_carrera,  
ea.creditos_aprobados,  
NVL(mpe.total_matriculas, 0) AS total_matriculas,  
NVL(mpe.promedio_notas, 0) AS promedio_notas
```

```
FROM EstudiantesActivos ea
```

```
LEFT JOIN MatriculasPorEstudiante mpe ON ea.cedula = mpe.cedula_estudiante
```

```
ORDER BY promedio_notas DESC, creditos_aprobados DESC;
```

```
-- CTE recursiva (ejemplo: jerarquía de prerrequisitos)
```

Consultas Analíticas (Window Functions)

```
-- =====  
-- FUNCIONES DE VENTANA (WINDOW FUNCTIONS)  
-- =====
```

```
-- ROW_NUMBER: Numerar filas
```

```
SELECT
```

```
ROW_NUMBER() OVER (ORDER BY creditos_aprobados DESC) AS ranking,  
nombres,  
apellidos,  
codigo_carrera,  
creditos_aprobados
```

```
FROM Estudiante
```

```
WHERE estado = 'ACTIVO'
```

```
FETCH FIRST 10 ROWS ONLY;
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

-- RANK y DENSE_RANK

```
SELECT
    nombres,
    apellidos,
    creditos_aprobados,
    RANK() OVER (ORDER BY creditos_aprobados DESC) AS rank_con_gaps,
    DENSE_RANK() OVER (ORDER BY creditos_aprobados DESC) AS rank_sin_gaps
FROM Estudiante
WHERE estado = 'ACTIVO'
FETCH FIRST 15 ROWS ONLY;
```

-- PARTITION BY: Ranking por carrera

```
SELECT
    codigo_carrera,
    nombres,
    apellidos,
    creditos_aprobados,
    RANK() OVER (PARTITION BY codigo_carrera ORDER BY creditos_aprobados DESC) AS rank_en_carrera
FROM Estudiante
WHERE estado = 'ACTIVO'
ORDER BY codigo_carrera, rank_en_carrera;
```

-- Mejor estudiante por carrera (usando ROW_NUMBER)

```
SELECT *
FROM (
    SELECT
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
codigo_carrera,  
nombres || ' ' || apellidos AS estudiante,  
creditos_aprobados,  
ROW_NUMBER() OVER (PARTITION BY codigo_carrera ORDER BY creditos_aprobados DESC) AS rn  
FROM Estudiante  
WHERE estado = 'ACTIVO'  
)  
WHERE rn = 1  
ORDER BY codigo_carrera;
```

-- LAG y LEAD: Acceder a filas anteriores/siguientes

```
SELECT  
nombres,  
apellidos,  
creditos_aprobados,  
LAG(creditos_aprobados, 1) OVER (ORDER BY creditos_aprobados) AS creditos_anterior,  
LEAD(creditos_aprobados, 1) OVER (ORDER BY creditos_aprobados) AS creditos_siguiente,  
creditos_aprobados - LAG(creditos_aprobados, 1) OVER (ORDER BY creditos_aprobados) AS diferencia  
FROM Estudiante  
WHERE estado = 'ACTIVO'  
AND creditos_aprobados > 0  
ORDER BY creditos_aprobados;
```

-- SUM con OVER: Totales acumulados

```
SELECT  
codigo_carrera,  
nombres,
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
apellidos,
creditos_aprobados,
SUM(creditos_aprobados) OVER (PARTITION BY codigo_carrera ORDER BY apellidos) AS creditos_acumulados,
SUM(creditos_aprobados) OVER (PARTITION BY codigo_carrera) AS total_carrera
FROM Estudiante
WHERE estado = 'ACTIVO'
ORDER BY codigo_carrera, apellidos;

-- AVG con ventanas móviles
SELECT
    periodo,
    AVG(nota_final) AS promedio_periodo,
    AVG(AVG(nota_final)) OVER (ORDER BY periodo ROWS BETWEEN 1 PRECEDING AND CURRENT ROW) AS promedio_movil_2periodos
FROM Matricula
WHERE nota_final IS NOT NULL
GROUP BY periodo
ORDER BY periodo;
```

OPTIMIZACIÓN DE CONSULTAS

Buenas Prácticas

```
-- =====
-- BUENAS PRÁCTICAS PARA OPTIMIZACIÓN
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

-- =====

-- ✓ 1. Usar WHERE en lugar de HAVING cuando sea posible

-- MALO (filtra después de agrupar)

```
SELECT codigo_carrera, COUNT(*) AS total
FROM Estudiante
GROUP BY codigo_carrera
HAVING codigo_carrera = 'ING-SIS';
```

-- BUENO (filtra antes de agrupar)

```
SELECT codigo_carrera, COUNT(*) AS total
FROM Estudiante
WHERE codigo_carrera = 'ING-SIS'
GROUP BY codigo_carrera;
```

-- ✓ 2. Limitar columnas con SELECT específico

-- MALO

```
SELECT * FROM Estudiante;
```

-- BUENO

```
SELECT cedula, nombres, apellidos, email
FROM Estudiante;
```

-- ✓ 3. Usar EXISTS en lugar de IN con subconsultas grandes

-- MENOS EFICIENTE

```
SELECT * FROM Estudiante
WHERE cedula IN (SELECT cedula_estudiante FROM Matricula);
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

-- *MÁS EFICIENTE*

```
SELECT * FROM Estudiante e
WHERE EXISTS (
    SELECT 1 FROM Matricula m
    WHERE m.cedula_estudiante = e.cedula
);
```

-- ✓ 4. Evitar funciones en WHERE sobre columnas indexadas

-- *MALO (no usa índice)*

```
SELECT * FROM Estudiante
WHERE UPPER(email) = 'JUAN@EPN.EDU.EC';
```

-- *BUENO (usa índice)*

```
SELECT * FROM Estudiante
WHERE email = 'juan@epn.edu.ec';
```

-- ✓ 5. Usar INNER JOIN en lugar de WHERE para relacionar tablas

-- *ANTIGUO ESTILO (menos eficiente)*

```
SELECT e.nombres, c.nombre
FROM Estudiante e, Carrera c
WHERE e.codigo_carrera = c.codigo;
```

-- *ESTILO MODERNO (más eficiente)*

```
SELECT e.nombres, c.nombre
FROM Estudiante e
INNER JOIN Carrera c ON e.codigo_carrera = c.codigo;
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

-- ✓ 6. Limitar resultados con *FETCH FIRST*

```
SELECT nombres, apellidos, creditos_aprobados
FROM Estudiante
```

```
ORDER BY creditos_aprobados DESC
```

```
FETCH FIRST 10 ROWS ONLY; -- Más eficiente que ROWNUM en algunos casos
```

Índices y Rendimiento

-- *CREACIÓN DE ÍNDICES PARA OPTIMIZACIÓN*

-- =====

-- *Ver índices existentes*

```
SELECT
    index_name,
    table_name,
    column_name,
    column_position
FROM user_ind_columns
WHERE table_name IN ('ESTUDIANTE', 'MATRICULA', 'ASIGNATURA')
ORDER BY table_name, index_name, column_position;
```

-- *Crear índices en columnas frecuentemente consultadas*

```
CREATE INDEX idx_estudiante_carrera ON Estudiante(codigo_carrera);
CREATE INDEX idx_estudiante_estado ON Estudiante(estado);
CREATE INDEX idx_matricula_estudiante ON Matricula(cedula_estudiante);
```




ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
CREATE INDEX idx_matricula_asignatura ON Matricula(codigo_asignatura);
```

```
CREATE INDEX idx_matricula_periodo ON Matricula(periodo);
```

```
-- Índice compuesto
```

```
CREATE INDEX idx_matricula_periodo_estado ON Matricula(periodo, estado);
```

```
-- Índice en expresión
```

```
CREATE INDEX idx_estudiante_apellidos_upper ON Estudiante(UPPER(apellidos));
```

```
-- Eliminar índice
```

```
DROP INDEX idx_estudiante_apellidos_upper;
```

```
-- Reconstruir índice
```

```
ALTER INDEX idx_estudiante_carrera REBUILD;
```

```
-- Ver estadísticas de uso de índices
```

```
SELECT  
    i.index_name,  
    i.table_name,  
    s.num_rows AS table_rows,  
    i.distinct_keys,  
    i.clustering_factor  
FROM user_indexes i  
LEFT JOIN user_tables s ON i.table_name = s.table_name  
WHERE i.table_name IN ('ESTUDIANTE', 'MATRICULA')  
ORDER BY i.table_name, i.index_name;
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

Hints de Oracle

-- *HINTS: Sugerencias al optimizador*

-- =====

-- *Forzar uso de índice*

```
SELECT /*+ INDEX(e idx_estudiante_carrera) */  
    e.nombres, e.apellidos, e.codigo_carrera  
FROM Estudiante e  
WHERE codigo_carrera = 'ING-SIS';
```

-- *Forzar FULL TABLE SCAN*

```
SELECT /*+ FULL(e) */  
    * FROM Estudiante e;
```

-- *Sugerir orden de JOIN*

```
SELECT /*+ LEADING(c e m) */  
    c.nombre,  
    e.nombres,  
    COUNT(m.id_matricula)  
FROM Carrera c  
JOIN Estudiante e ON c.codigo = e.codigo_carrera  
LEFT JOIN Matricula m ON e.cedula = m.cedula_estudiante  
GROUP BY c.nombre, e.nombres;
```

-- *Paralelizar consulta*



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

```
SELECT /*+ PARALLEL(e, 4) */  
    codigo_carrera,  
    COUNT(*) AS total  
FROM Estudiante e  
GROUP BY codigo_carrera;  
  
-- Usar hash join  
SELECT /*+ USE_HASH(e c) */  
    e.nombres,  
    c.nombre  
FROM Estudiante e  
JOIN Carrera c ON e.codigo_carrera = c.codigo;
```

EJERCICIOS PRÁCTICOS

Ejercicio 1: Consultas Básicas

Instrucciones: Resolver las siguientes consultas

1. Listar todos los estudiantes ordenados por apellido
2. Mostrar asignaturas con más de 4 créditos
3. Encontrar estudiantes con créditos entre 20 y 50
4. Buscar docentes cuyo apellido empiece con 'G' o 'M'
5. Listar estudiantes sin teléfono registrado
6. Mostrar las 10 asignaturas con más horas totales (teoría + práctica)
7. Calcular la edad de todos los estudiantes activos
8. Encontrar emails que no terminen en @epn.edu.ec
9. Listar carreras con duración mayor o igual a 10 semestres
10. Mostrar estudiantes nacidos en 2003

Entregable: Script SQL con las 10 consultas



Ejercicio 2: Funciones de Agregación y GROUP BY

Tareas:

1. Contar estudiantes por carrera y estado
2. Calcular promedio, máximo y mínimo de créditos por carrera
3. Encontrar carreras con más de 5 estudiantes activos
4. Calcular promedio de notas por asignatura
5. Mostrar docentes con más de 3 matrículas asignadas
6. Contar asignaturas por nivel y carrera
7. Calcular tasa de aprobación por periodo
8. Encontrar estudiantes con promedio ≥ 8.5
9. Mostrar distribución de estudiantes por género y carrera
10. Calcular estadísticas completas del sistema

Entregable: Consultas con interpretación de resultados

Ejercicio 3: JOINS y Relaciones

Desarrollar:

1. Reporte de estudiantes con sus carreras y número de matrículas
2. Asignaturas con sus prerrequisitos en formato legible
3. Matrículas completas (estudiante, asignatura, docente, notas)
4. Estudiantes sin ninguna matrícula (usar LEFT JOIN)
5. Docentes sin asignaciones actuales
6. Carreras sin estudiantes matriculados
7. Top 5 asignaturas más populares (más matrículas)
8. Estudiantes con todas sus calificaciones
9. Asignaturas que nadie ha aprobado aún
10. Reporte completo de un estudiante específico

Entregable: 10 consultas con JOINS variados



Ejercicio 4: Subconsultas

Resolver:

1. Estudiantes con más créditos que el promedio
2. Asignaturas más difíciles (menor promedio de notas)
3. Docente con más estudiantes asignados
4. Estudiantes que aprobaron todas las materias que cursaron
5. Carrera con mejor rendimiento académico promedio
6. Asignaturas sin prerequisites
7. Estudiantes que NO han cursado BD-501
8. Asignaturas que son prerequisite de más de 2 materias
9. Docentes que solo imparten asignaturas de nivel avanzado (≥ 5)
10. Estudiantes con todas las notas sobre 7.0

Entregable: Subconsultas en WHERE, SELECT y FROM

Ejercicio 5: Consultas Avanzadas

Implementar:

1. **Ranking de estudiantes** por carrera usando ROW_NUMBER
2. **CTE complejo** con estadísticas por carrera y comparación con promedio general
3. **Consulta PIVOT** mostrando matrículas por periodo y estado
4. **Window function** para mostrar tendencia de notas por estudiante
5. **Consulta con UNION** combinando información de diferentes periodos
6. **Subconsulta correlacionada** para comparar estudiante vs promedio de su carrera
7. **CASE complejo** clasificando estudiantes en categorías de rendimiento
8. **Consulta recursiva** para cadena de prerequisites
9. **Análisis temporal** con LAG/LEAD mostrando evolución de matrículas
10. **Reporte ejecutivo** combinando múltiples técnicas



**ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS**

Análisis de resultados:

El desarrollo de las consultas SQL permitió evidenciar el funcionamiento integral del lenguaje de consulta en una base de datos relacional académica, demostrando la capacidad de extraer, filtrar, ordenar, agrupar y relacionar información proveniente de múltiples tablas. Los resultados permitieron validar el comportamiento de los operadores lógicos y de comparación, así como la correcta segmentación de información mediante cláusulas WHERE, GROUP BY y HAVING. De igual forma, los JOINS confirmaron la integridad referencial del modelo relacional al combinar registros de estudiantes, carreras, docentes, asignaturas y matrículas, permitiendo análisis estadísticos avanzados como rendimiento académico, participación en periodos, carga docente y aprobación por asignatura. Las subconsultas y operadores de conjuntos evidenciaron que SQL facilita construir consultas cada vez más complejas sin comprometer legibilidad ni performance cuando se estructuran correctamente.

Conclusiones y recomendaciones:

El laboratorio permitió reforzar el dominio del lenguaje SQL como herramienta indispensable para la gestión y análisis de datos en bases de datos relacionales, demostrando que el uso adecuado de las distintas cláusulas y operadores incrementa significativamente la capacidad analítica del usuario. Se concluye además que la correcta comprensión de relaciones entre tablas y la aplicación de técnicas avanzadas como agrupamientos, subconsultas y uniones es clave para resolver problemas reales en sistemas de información, optimizando la obtención de datos y facilitando la toma de decisiones. Finalmente, se destaca que el orden lógico y estructurado en la construcción de consultas no solo mejora la precisión de los resultados, sino que también contribuye a un mejor desempeño y mantenibilidad del código SQL.

Bibliografía:

[1] R. Elmasri and S. Navathe, Fundamentals of Database Systems, 7th ed. Pearson, 2016. [Online]. Available: <https://www.pearson.com>

[2] Oracle Corporation, Oracle® Database SQL Language Reference. 2024. [Online]. Available: <https://docs.oracle.com/en/database/>

[3] A. Silberschatz, H. Korth, and S. Sudarshan, Database System Concepts, 7th ed. McGraw-Hill, 2020. [Online]. Available: <https://www.db-book.com>