

Conceptos de Bases de Datos

Trabajo Práctico N° 5: Lenguaje SQL- 2

1. Responda las siguientes consultas considerando la sentencia INSERT:

a. ¿Se pueden obviar los campos de la clave primaria? Justifique.

Ejemplifique.

No se puede omitir la clave primaria salvo que sea de autoincremento sino daría error al faltar la clave que diferencia a una fila de otras

```
CREATE TABLE persona (  
    dni INT PRIMARY KEY,  
    nombre VARCHAR(50)
```

```
);
```

```
INSERT INTO persona (nombre) VALUES ('Jorge'); -- ERROR: falta la clave
```

primaria

b. ¿Se puede obviar los campos de alguna clave candidata que no fue elegida clave primaria? Justifique. Ejemplifique.

Si se evade siempre cuando en la tabla no tenga una restricción de NOT NULL o UNIQUE que determina que el espacio debe de ser completado si o sí.

```
CREATE TABLE usuario (  
    id INT PRIMARY KEY,  
    email VARCHAR(100) UNIQUE, -- clave candidata  
    nombre VARCHAR(50)
```

```
);-- Se puede insertar sin email al no tener el NOT NULL NO ES NECESARIA
```

```
INSERT INTO usuario (id, nombre) VALUES (1, 'Jorge');
```

c. ¿Qué ocurre con los campos de la clave foránea? Vincule su respuesta según las ACCIONES que se utilizaron en las definiciones de las Tablas. Ejemplifique.

Depende de las acciones referenciales anteriormente creadas

ON Delete cascade: elimina toda la fila

ON Delete set null: pone en null toda la fila

ON Delete resting o no action: prohíbe la eliminación de la información

ON Update cascade: actualiza la información

2. Responda las siguientes consultas considerando la subsentencia WHERE:

a. Defina conceptualmente para qué existe.

La cláusula filtra filas en una sentencia (como select, update, delete) según una condición que permite aplicar la operación solo a los registros que cumplan esa condición

b. ¿Qué ocurre si no se utiliza en una sentencia DELETE?

Se borra todas las filas de la tabla

c. ¿Qué ocurre si no se utiliza en una sentencia UPDATE?

Se actualiza todas la filas de la tabla

- d. Ejemplifique para cada uno de los puntos anteriores.
- DELETE FROM alumno;
- Elimina TODOS los alumnos
- UPDATE alumno SET nombre = 'Juan';
- Todos los alumnos se llaman Juan
- e. ¿Para qué se utiliza la expresión “LIKE”?
- Para buscar textos similares o con el mismo patrón, usando el comodín “%”
- f. Defina y brinde ejemplos simples de cada una de las formas de usar la expresión del inciso anterior (al principio, en el medio, al final).
- Busca las personas con la terminación “ia” en su nombre
- SELECT * FROM persona WHERE nombre LIKE '%ia';
- Busca las personas con la inicial “AN” en su nombre
- SELECT * FROM persona WHERE nombre LIKE 'An%';
- Busca las personas que en la mitad de nombre tenga “Ar”
- SELECT * FROM persona WHERE nombre LIKE '%ar%';
3. ¿Se puede modificar un dato de una clave primaria? Justifique ejemplificando convenientemente.
- Si se puede pero puede ocasionar problemas si la misma esta relacionada con otras tablas excepto si las mismas tienen ON UPDATE CASCADE que permite el cambio de esta sin problema.
4. Enumere los operadores de comparación vistos que pueden utilizarse en la subsentencia WHERE. Ejemplifique.

Operador	Significado	Ejemplo
=	Igual a	WHERE nombre = 'Juan'
!= o <>	Distinto de	WHERE edad != 18
<	Menor que	WHERE edad < 30
<=	Menor o igual que	WHERE edad <= 25
>	Mayor que	WHERE edad > 40
>=	Mayor o igual que	WHERE edad >= 18
BETWEEN	Entre dos valores	WHERE edad BETWEEN 18 AND 25
IN	Dentro de una lista	WHERE nombre IN ('Ana','Luis')
LIKE	Coincidencia de texto	WHERE nombre LIKE 'M%'
IS NULL	Es nulo	WHERE email IS NULL

Operador	Significado	Ejemplo
IS NOT NULL	No es nulo	WHERE email IS NOT NULL
5. ¿Cómo calcularla la antigüedad en años que tienen los empleados de una empresa? Considere una tabla expresada en el Modelo Relacional como: Empleado (DNI, nombre, fecha Ingreso)		
SELECT TIMESTAMPDIFF(YEAR, fechaIngreso, CURDATE()) AS antigüedad_anios FROM EMPLEADO;		
6. Describa cada una de las funciones agregadas presentadas en la teoría, con sendos ejemplos en los que:		
Función	¿Qué hace?	
COUNT()	Cuenta la cantidad de filas	
SUM()	Suma los valores numéricos	
AVG()	Calcula el promedio	
MIN()	Devuelve el valor mínimo	
MAX()	Devuelve el valor máximo	
a) No se utilice ningún atributo en el SELECT. Cantidad de empleados en la tabla SELECT COUNT(*) FROM Empleado;		
b) Se utilice un único atributo en el SELECT. La fecha de ingreso mas antigua SELECT MIN(fechaIngreso) FROM Empleado;		
c) Se utilicen más de un atributo en el SELECT. Promedio de antigüedad por cada año de ingreso SELECT YEAR(fechaIngreso) AS aniIngreso, AVG(TIMESTAMPDIFF(YEAR, fechaIngreso, CURDATE())) AS promedio_antigüedad FROM Empleado GROUP BY aniIngreso;		
7. A partir del ejemplo definido en el Ejercicio 8.a “Banco”, del Trabajo Practico N°4- SQL, codifique en el lenguaje SQL los siguientes requerimientos:		
(a) Liste los Números y Tipos de las Cajas de Ahorro que se abrieron a partir del '01/02/2024'.		
SELECT nroCaja, tipoA FROM CAJADEAHORRO		

WHERE fecha_Apertura > '01/02/2024';

- (b) Liste los Números, Tipos de las Cajas de Ahorro y el saldo resultante para las Extracciones realizadas entre los días '02/02/2025' y '10/02/2025'.

```
SELECT nroCaja, tipoA, totalEnCaja  
FROM CAJASDEAHORRO
```

WHERE fecha_Apertura > '02/02/2025' AND fecha_Apertura < '10/02/2025';

- (c) Liste el Apellido, el Nombre y el DNI de los Clientes que realizaron extracciones durante el mes de febrero del año pasado.

```
SELECT c.apellido, c.nombre, c.DNI  
FROM CLIENTES
```

INNER JOIN CAJASDEAHORRO ca ON c.idCliente = ca.idCliente

WHERE MONTH(ca.fecha_Apertura) = 2 AND YEAR(ca.fecha_Apertura) = 2024;

- (d) Liste los saldos totales en la caja de todas las cajas que se hayan abierto durante el año 2025 agrupados por mes.

```
SELECT totalEnCaja
```

FROM CAJASDEAHORRO

WHERE YEAR(fecha_Apertura) = 2025;

- (e) Liste el Apellido, el Nombre y el DNI de los Clientes que tienen más de una Caja de Ahorro.

```
SELECT c.apellido, c.nombre, c.DNI  
FROM CLIENTES c
```

INNER JOIN CAJASDEAHORRO ca ON c.idCliente = ca.idCliente

GROUP BY ca.idCliente, c.nombre, c.apellido

HAVING COUNT(nroCaja) > 1;

8. A partir del ejemplo definido en el Ejercicio 8.b “Universidad” del Trabajo Práctico N° 4- SQL, codifique en el lenguaje SQL los siguientes requerimientos:

- (a) Agregue un Alumno con legajo 4545. Realice Todas las operaciones necesarias. Si le hacen falta más datos debe inventarlos.

```
INSERT INTO PERSONA (tipodocumento, numerodocumento, apellido,  
nombre, fechanacimiento, localidad, email)
```

```
VALUES ('DNI', 2356633, 'Herrera', 'Luis', '2004-05-02', 'Bariloche',  
'luis283@gmail.com');
```

```
INSERT INTO ALUMNO (tipodocumento, numerodocumento, legajo)
```

```
VALUES ('DNI', 2356633, 4545);
```

```
INSERT INTO CURSA (tipodocumentoAlumno, numerodocumentoAlumno,  
fechaAsistencia, numeroAula, nombreEdificio, NombreMateria, temaVisto)
```

```
VALUES ('DNI', 2356633, '2024-04-22', 202, 'Pabellón B', 'Álgebra',  
'Matrices');
```

- (b) Realice el registro de cursado del legajo 4545 en la materia “Conceptos de Bases de Datos” para la fecha “10/05/2024”. Realice todas las operaciones previas SQL necesarias. Si le hacen falta más datos debe inventarlos.

```
INSERT INTO CURSA (tipodocumentoAlumno, numerodocumentoAlumno,
fechaAsistencia, numeroAula, nombreEdificio, NombreMateria, temaVisto)
VALUES ('DNI', 2356633, '2024-05-10', 202, 'Pabellón B', 'Bases de Datos I',
'SQL básico');
```

- (c) Actualice la fecha de ingreso a “21/05/2023” a todas las personas que comiencen con cargo “ADMIN”

```
UPDATE PROFESOR
SET fechaingreso = '2023/05/21'
WHERE cargo LIKE 'Titular%';
```

- (d) Borre los cursados que se hayan realizado el día 24 de Marzo, ya que se trata de un día feriado.

```
DELETE FROM CURSA
WHERE MONTH(fechaAsistencia) = 3 AND DAY(fechaAsistencia) = 24;
```

- (e) Realice un listado de todas las personas de la base de datos Banco.

```
SELECT * FROM PERSONA;
```

- (f) Seleccionar el nombre y el programa de cada materia de más de 60 horas cátedra y cuyo nombre de la materia contenga la palabra “datos”.

```
SELECT nombre, programa
FROM MATERIA
WHERE horasCatedra >= 60 AND nombre LIKE '%datos%';
```

- (g) Realice un listado de cursados (todos los campos) de este año, de la materia “Estadística”, ordenados por fecha de asistencia y DNI de alumno.

```
SELECT p.numerodocumento, cu.fechaAsistencia, cu.numeroAula,
cu.nombreEdificio
FROM PERSONA p
INNER JOIN CURSA cu ON p.numerodocumento =
cu.numerodocumentoAlumno
WHERE NombreMateria = 'Bases de Datos I'
ORDER BY cu.fechaAsistencia ASC, p.numerodocumento ASC;
```

- (h) Realice un listado de cursados (todos los campos) del año pasado, de la materia “Conceptos de base de datos” ordenado por DNI de alumno, agregando además una nueva columna que muestre cuantos días han transcurrido desde la fecha de asistencia.

```
SELECT p.numerodocumento, cu.fechaAsistencia, cu.numeroAula,
cu.nombreEdificio, DATEDIFF(curdate(),cu.fechaAsistencia)
FROM PERSONA p
INNER JOIN CURSA cu ON p.numerodocumento =
cu.numerodocumentoAlumno
```

```
WHERE NombreMateria = 'Álgebra'
ORDER BY p.numerodocumento ASC;
```

- (i) Modifique la consulta realizada en el ítem anterior, con una nueva columna que muestre cuántos meses han transcurrido. Usar Datediff.

```
SELECT p.numerodocumento, cu.fechaAsistencia, cu.numeroAula,
cu.nombreEdificio, ROUND(DATEDIFF(curdate(),cu.fechaAsistencia)/30)
FROM PERSONA p
INNER JOIN CURSA cu ON p.numerodocumento =
cu.numerodocumentoAlumno
WHERE NombreMateria = 'Álgebra'
ORDER BY p.numerodocumento ASC;
```

- (j) Realizar un listado de los profesores que tengan el cargo “Titular” o bien hayan ingresado hace más de 5 años. El formato del listado de la consulta debe tener los siguientes datos: “Tipo de Documento”, “Número de Documento”, “Cargo que Ocupa” y “Fecha de Ingreso”.

```
SELECT tipodocumento, numerodocumento, cargo, fechaingreso
FROM PROFESOR
WHERE DATEDIFF(curdate(),fechaingreso) >1825;
```

- (k) Listar las aulas que pueden albergar a 20 alumnos o menos.

```
SELECT numero
FROM AULA
WHERE cantidadBancos <= 20;
```

- (l) Mostrar en un listado las aulas con una columna que informe el doble de los bancos asignados.

```
SELECT numero, cantidadBancos * 2 AS doble_bancos
FROM AULA ;
```

- (m) Los profesores con más de 10 años de antigüedad, siempre que hayan estado a cargo de alguna materia de más de 80 horas cátedra entre el '01/02/2024' y el día de hoy.

```
SELECT *
FROM PROFESOR pr
INNER JOIN MATERIA ma ON pr.numerodocumento = ma.NumeroDocProfesorACargo
INNER JOIN CURSA cu ON ma.nombre = cu.NombreMateria
WHERE DATEDIFF( curdate() , pr.fechaingreso ) >3650
AND ma.horasCatedra >80
BETWEEN '2024-02-01'
AND CURDATE();
```

- (n) Intente realizar la misma consulta anterior utilizando otros operadores. Justifique.

```
SELECT *
```

```

FROM PROFESOR pr
INNER JOIN MATERIA ma ON pr.numerodocumento =
ma.NumeroDocProfesorACargo
INNER JOIN CURSA cu ON ma.nombre = cu.NombreMateria
WHERE DATEDIFF(CURDATE(), pr.fechaingreso) > 3650
AND ma.horasCatedra > 80
AND cu.fechaAsistencia >= '2024-02-01'
AND cu.fechaAsistencia <= CURDATE();

```

- (o) Mostrar las materias de 50 horas catedras que se hayan cursado durante el año 2024. Utilizar Exists e IN para resolver.

```

SELECT *
FROM MATERIA ma
WHERE EXISTS (
SELECT 1
FROM CURSA cu
WHERE ma.nombre = cu.NombreMateria
AND YEAR(cu.fechaAsistencia) = 2024
);

```

- (p) Mostrar los Legajos de alumnos que nunca cursaron materias en el año 2023 y tampoco cursaron en el año 2024.

```

SELECT legajo
FROM ALUMNO a
WHERE NOT EXISTS (
SELECT 1
FROM CURSA cu
WHERE a.tipodocumento = cu.tipodocumentoAlumno AND
a.numerodocumento = cu.numerodocumentoAlumno AND
YEAR(cu.fechaAsistencia) = '2024'
);

```

- (q) Contar la cantidad de alumnos de la base de datos.

```

SELECT COUNT(*) cantidad_alumnos
FROM ALUMNO
HAVING COUNT(*) > 0;

```

- (r) ¿Cuántos bancos hay en el edificio "FAI"?

```

SELECT SUM(cantidadBancos) cantidad_asiento
FROM AULA
WHERE nombreEdificio = 'Pabellón B';

```

- (s) ¿Desde qué fecha la base de datos Universidad registra Profesores? ¿Cuándo fue el último ingreso de un profesor?

```

SELECT MIN(fechaingreso) AS primer_ingreso, MAX(fechaingreso) AS
ultimo_ingreso

```

FROM PROFESOR ;

- (t) ¿Cuántos profesores hay por cargo? Debe listar Cargo y Cantidad de profesores por cada cargo.

```
SELECT cargo, count(*) AS cantidad_profesores
FROM PROFESOR
GROUP BY cargo;
```

- (u) ¿Cuántos bancos hay por edificio?

```
SELECT ed.nombre , SUM(au.cantidadBancos) AS cantidadDEBancos
FROM EDIFICIO ed
INNER JOIN AULA au ON au.nombreEdificio = ed.nombre
GROUP BY ed.nombre;
```

- (v) Realizar un listado con tipo de documento, número de documento, apellido y nombre de los profesores que ingresaron a partir del año 2020 y que sean de las localidades de Neuquén o de General Roca.

```
SELECT pr.tipodocumento, pr.numerodocumento, pe.apellido, pe.nombre
FROM PROFESOR pr, PERSONA pe, MATERIA ma, CURSA cu, EDIFICIO ed
WHERE pr.tipodocumento = pe.tipodocumento
AND pr.numerodocumento = pe.numerodocumento
AND YEAR( pr.fechaingreso ) >= '2020'
AND pr.tipodocumento = ma.tipodocProfesorACargo
AND pr.numerodocumento = ma.NumeroDocProfesorACargo
AND ma.nombre = cu.nombreMateria
AND cu.nombreEdificio = ed.nombre
AND ed.ubicacion = 'Av. Siempreviva 742';
```

- (w) Realizar un listado de número de aula, Edificio, cantidad de bancos y ubicación del edificio, de aquellas aulas con más de 10 bancos.

```
SELECT au.numero, ed.nombre, au.cantidadBancos, ed.ubicacion
FROM AULA au
INNER JOIN EDIFICIO ed ON au.nombreEdificio = ed.nombre
WHERE au.cantidadBancos > 10 ;
```

- (x) Se necesita saber la cantidad de asistencias por alumno. El resultado debe contener todos los datos del alumno (recordar que son personas) y la cantidad de asistencias durante todo el año pasado hasta el día de hoy.

```
SELECT pe.apellido, pe.nombre, al.tipodocumento, al.numerodocumento,
count( * ) AS TOTALASISTENCIAS
FROM ALUMNO al
INNER JOIN PERSONA pe ON al.tipodocumento = pe.tipodocumento
AND al.numerodocumento = pe.numerodocumento
INNER JOIN CURSA cu ON al.tipodocumento = cu.tipodocumentoAlumno
AND al.numerodocumento = cu.numerodocumentoAlumno
WHERE YEAR( cu.fechaAsistencia ) >= '2024'
```



```
GROUP BY pe.tipodocumento, pe.numerodocumento;
```