

# Base de Datos I - Parcial 2 (2021C2)


Dado el siguiente MR de una empresa de reparacion de lavarropas, donde las PK están en **negrita** y las FK subrayadas, realice las operaciones que se detallan a continuación utilizando SQL.

Reparacion(**nro**, fecha, importe, id\_cliente, cuil\_empleado)  
Cliente(**id**, nombre, cod\_loc, fecha\_nac)  
Localidad(**cod**, descripcion)  
Empleado (**cuil**, nombre, cod\_loc, sueldo)  
Insumo(**nro**, descripcion, costo, cuit\_proveedor)  
Proveedor(**cuit**, razon\_social, telefono)  
Reparacion\_Insumo(**nro\_reparacion**, **nro\_insumo**)

*Nota: el campo fecha es de tipo DATE (no es DATETIME)*

...

Puntos: 10/10

1. Modificar los empleados que comienzan con la letra "A" de manera que tengan el mismo sueldo del empleado de CUIL 123456789   
(1/1 puntos)

```
UPDATE empleado emp
SET emp.sueldo = (SELECT emp1.sueldo
                  FROM empleado emp1
                  WHERE emp1.cuil = 123456789)
WHERE emp.nombre LIKE "A%";
```

2. Agregar un nuevo cliente de nombre "Juan Perez", id 15, fecha de nacimiento 21 de mayo de 1956 y código de localidad 34.  
(1/1 puntos)

```
Insert into cliente (id, nombre, cod_loc, fecha_nac)
values (15, "Juan Pérez", 34, "1956-05-21");
```

3. Listar las reparaciones realizadas en el año 2012. Por cada una de ellas mostrar: fecha, importe, nombre de cliente y descripción de su localidad y finalmente el nombre del empleado que realizó la reparación. Ordenar los resultados por fecha (las más recientes primero).  
(1/1 puntos)

```
SELECT rep.fecha as "Fecha Reparación", rep.importe as "Importe Reparación", cli.nombre as "Nombre Cliente",
loc.descripcion as "Localidad", emp.nombre as "Nombre del Empleado"
FROM reparacion rep JOIN cliente cli ON rep.id_cliente = cli.id
                  JOIN empleado emp ON rep.cuil_empleado = emp.cuil
                  JOIN localidad loc ON cli.cod_loc = loc.cod
WHERE rep.fecha BETWEEN "2012-01-01" AND "2012-12-31"
ORDER BY rep.fecha DESC;
```

4. Eliminar a todas las reparaciones que se hayan efectuado con un empleado que viva en "San Justo"  
(1/1 puntos)

```
DELETE FROM reparacion rep
WHERE rep.cuil_empleado IN (SELECT emp cuil
                           FROM empleado emp JOIN Localidad loc ON emp.cod_loc = loc.cod
                           WHERE loc.descripcion = "San Justo");
```

5. Indicar el nombre y sueldo de aquellos empleados que NO hayan realizado reparación alguna en todo el año 2014.  
(1/1 puntos)

```
SELECT emp.nombre as "Nombre Empleado", emp.sueldo as "Sueldo del Empleado"
FROM empleado emp
WHERE NOT EXISTS (SELECT 1
                  FROM reparacion rep
                  WHERE rep.cuil_empleado = emp.cuil
                  AND rep.fecha BETWEEN "2014-01-01" AND "2014-12-31");
```

6. Listar la descripción de aquellos insumos que se han utilizado en todas las reparaciones.  
(1/1 puntos)

```
SELECT ins.descripcion as "Descripción Insumo"
FROM insumo ins JOIN reparacion_insumo ri ON ins.nro = ri.nro_insumo
GROUP BY ins.nro, ins.descripcion
HAVING COUNT(DISTINCT ri.nro_reparacion) = (SELECT COUNT(rep.nro)
                                           FROM reparacion rep);
```

7. Indicar las tres posibles operaciones que intervienen en una transacción de base de datos y explique brevemente para qué sirve cada una de ellas.  
(1/1 puntos)

Las tres operaciones posibles que intervienen en una transacción de base de datos son Begin Transaction, Rollback Transaction y Commit Transaction. El primero, como su traducción lo indica, es para "avisar" que se va a iniciar la transacción; es donde se va a realizar todos los cambios. En la segunda mencionada, es para evitar errores, y tiene que ir siempre antes del commit; cuando se detectó un error en la etapa anterior, se puede tirar para atrás realizando el "Rollback". Y por último, el Commit es para enviar y confirmar la transacción. En este último paso, no se puede tirar atrás ningún cambio y no se puede realizar ninguna modificación.

8. Indicar el sueldo máximo de los empleados por cada localidad (descripción).  
(1/1 puntos)

```
SELECT loc.descripcion as "Localidad", MAX(emp.sueldo) as "Sueldo Máximo"
FROM empleado emp JOIN localidad loc ON emp.cod_loc = loc.cod
GROUP BY loc.cod, loc.descripcion;
```

9. Listar el nombre de aquellos clientes que hayan efectuado más de 5 reparaciones superiores a \$1000.  
(1/1 puntos)

```
SELECT cli.nombre as "Nombre del Cliente"
FROM cliente cli
WHERE cli.id IN (SELECT rep.id_cliente
                FROM reparacion rep
                WHERE rep.importe > 1000
                GROUP BY rep.id_cliente
                HAVING COUNT(rep.id_cliente) > 5);
```

10. Agregar un campo nuevo a la tabla Proveedor, que indique la localidad donde el mismo vive, utilizando el tipo de dato y las restricciones de integridad que correspondan.  
(1/1 puntos)

```
ALTER TABLE proveedor ADD COLUMN cod_loc;
ALTER TABLE proveedor ADD FOREIGN KEY (cod_loc) REFERENCES localidad (cod);
```

Este contenido lo creó el propietario del formulario. Los datos que envíes se enviarán al propietario del formulario. Microsoft no es responsable de las prácticas de privacidad o seguridad de sus clientes, incluidas las que adopte el propietario de este formulario. Nunca des tu contraseña.

Con tecnología de Microsoft Forms | [Privacidad y cookies \(https://go.microsoft.com/fwlink/p/?linkid=857875\)](https://go.microsoft.com/fwlink/p/?linkid=857875) | [Términos de uso \(https://go.microsoft.com/fwlink/p/?LinkId=2083423\)](https://go.microsoft.com/fwlink/p/?LinkId=2083423)