## TALLER DE ADMINISTRACIÓN DE DATOS

## TALLER 2 - PostgreSQL

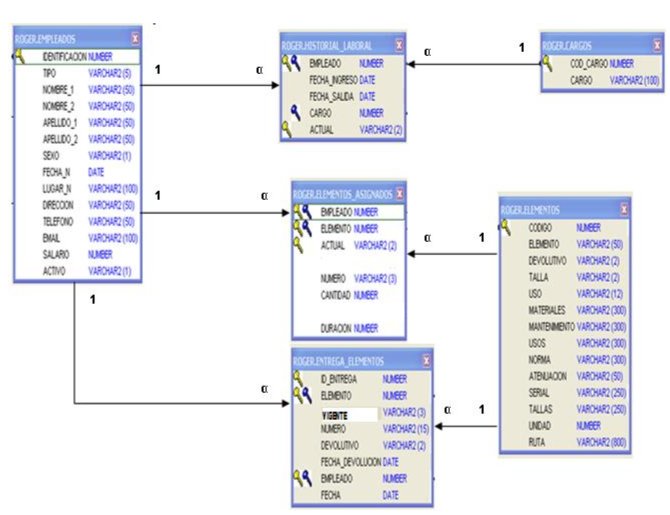
**INTEGRANTES:**

Johan Sneider Albarracin Gómez - 160003901

Edgar David Lozada González – 160004023

Carlos Danilo Núñez Gil – 160004032

* Dado el siguiente modelo entidad relación cree las tablas y las respectivas relaciones utilizando para ello la herramienta PgAdmin y el usuario que le fue asignado. Las instrucciones sql que utilice deben quedar guardadas en un archivo plano (tablas\_taller2.sql). Puede utilizar como ejemplos para crear las tablas el anexo de este documento.



1. /\*
2. CREACION DE LA ESTRUCTURA DE TABLAS
3. \*/
4. **CREATE** **TABLE** cargos(
5. cod\_cargo **INT** **NOT** **NULL** **PRIMARY** **KEY**,
6. cargo VARCHAR(100) **NOT** **NULL**
7. );
9. **CREATE** **TABLE** empleados(
10. identificacion **INT** **NOT** **NULL** **PRIMARY** **KEY**,
11. tipo VARCHAR(5),
12. nombre\_1 VARCHAR(50),
13. nombre\_2 VARCHAR(50),
14. apellido\_1 VARCHAR(50),
15. apellido\_2 VARCHAR(50),
16. sexo VARCHAR(1),
17. fecha\_n **DATE**,
18. lugar\_n VARCHAR(100),
19. direccion VARCHAR(50),
20. telefono VARCHAR(50),
21. email VARCHAR(100),
22. salario **NUMERIC**,
23. activo VARCHAR(1)
24. );
26. **CREATE** **TABLE** historial\_laboral(
27. empleado **INT** **NOT** **NULL**,
28. fecha\_ingreso **DATE**,
29. fecha\_salida **DATE**,
30. cargo **INT** **NOT** **NULL**,
31. actual VARCHAR(2) **NOT** **NULL**,
32. **PRIMARY** KEY(empleado,actual),
33. **CONSTRAINT** fk\_empleado **FOREIGN** KEY(empleado) **REFERENCES** empleados(identificacion),
34. **CONSTRAINT** fk\_cargo **FOREIGN** KEY(cargo) **REFERENCES** cargos(cod\_cargo)
35. );
37. **CREATE** **TABLE** elementos(
38. codigo **INT** **NOT** **NULL** **PRIMARY** **KEY**,
39. elemento VARCHAR(50),
40. devolutivo VARCHAR(2),
41. talla VARCHAR(2),
42. uso VARCHAR(12),
43. materiales VARCHAR(300),
44. mantenimiento VARCHAR(300),
45. usos VARCHAR(300),
46. norma VARCHAR(300),
47. atenuacion VARCHAR(50),
48. serial VARCHAR(250),
49. tallas VARCHAR(250),
50. unidad **INT**,
51. ruta VARCHAR(800)
52. );
54. **CREATE** **TABLE** entrega\_elementos(
55. id\_entrega **INT** **NOT** **NULL**,
56. elemento **INT** **NOT** **NULL**,
57. vigente VARCHAR(3),
58. numero VARCHAR(15),
59. devolutivo VARCHAR(2),
60. fecha\_devolucion **DATE**,
61. empleado **INT** **NOT** **NULL**,
62. fecha **DATE**,
63. **PRIMARY** KEY(id\_entrega,elemento,empleado),
64. **CONSTRAINT** fk\_elemento **FOREIGN** KEY(elemento) **REFERENCES** elementos(codigo),
65. **CONSTRAINT** fk\_entrega\_empleado **FOREIGN** KEY(empleado) **REFERENCES** empleados(identificacion)
66. );
68. **CREATE** **TABLE** elementos\_asignados(
69. empleado **INT** **NOT** **NULL**,
70. elemento **INT** **NOT** **NULL**,
71. actual VARCHAR(2),
72. numero VARCHAR(3),
73. cantidad **INT**,
74. duracion **INT**,
75. **PRIMARY** KEY(empleado,elemento,actual),
76. **CONSTRAINT** fk\_ele\_asig\_empleado **FOREIGN** KEY(empleado) **REFERENCES** empleados(identificacion),
77. **CONSTRAINT** fk\_ele\_asig\_elemento **FOREIGN** KEY(elemento) **REFERENCES** elementos(codigo)
78. );

* Verifique la existencia de las tablas y describa la estructura de cada una.

tabla cargos:

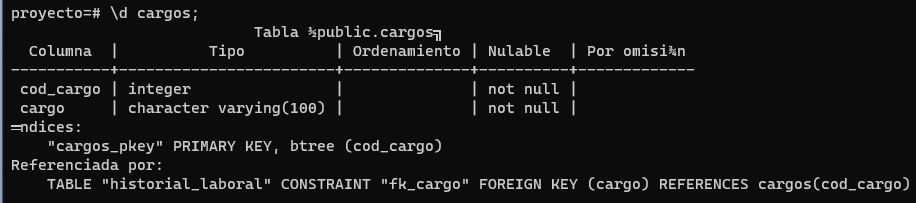


tabla empleados:

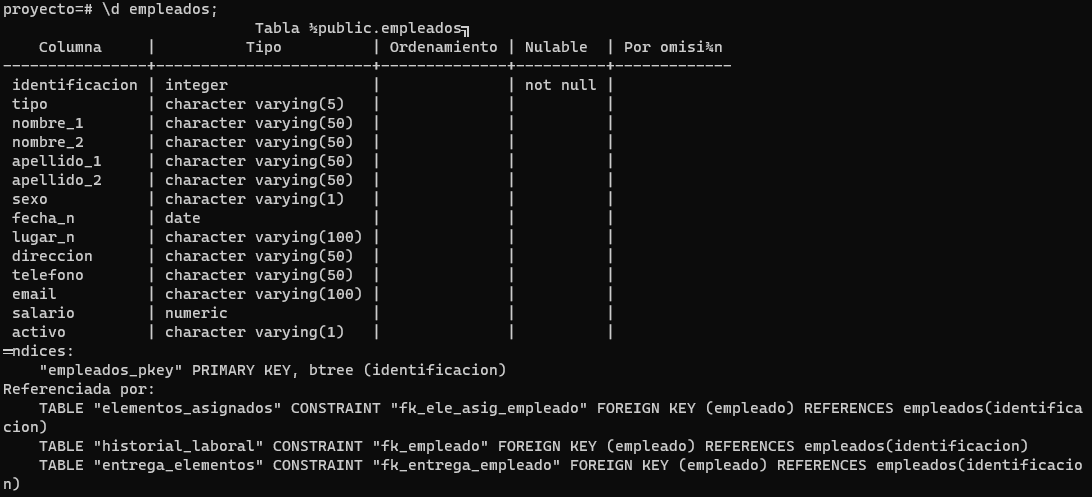


tabla historial\_laboral:

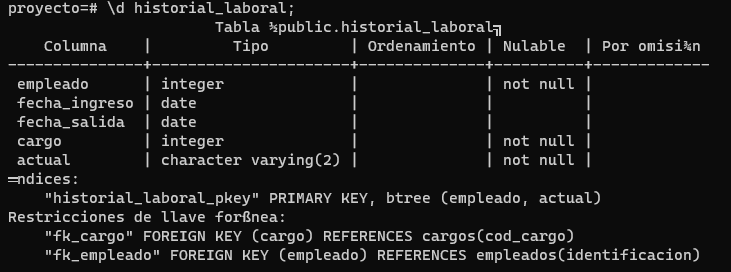


tabla elementos:

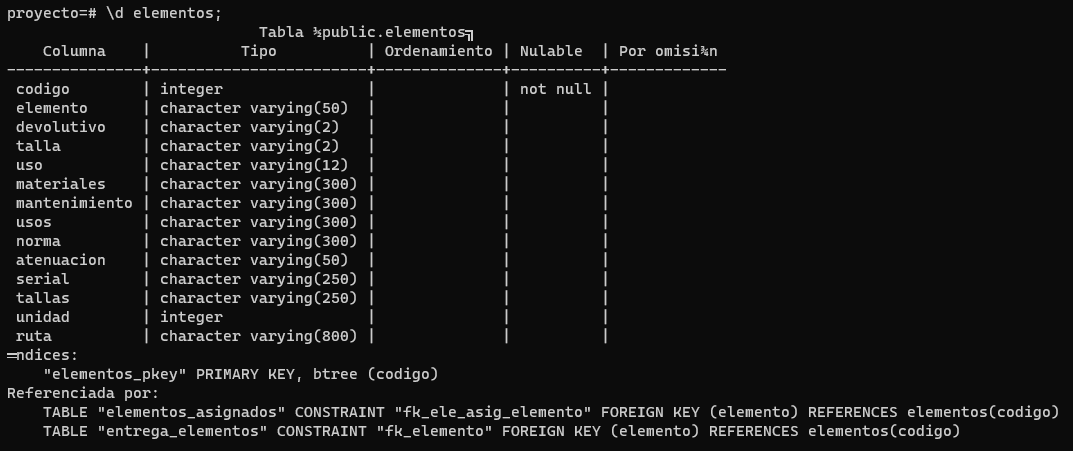


tabla entrega\_elementos:

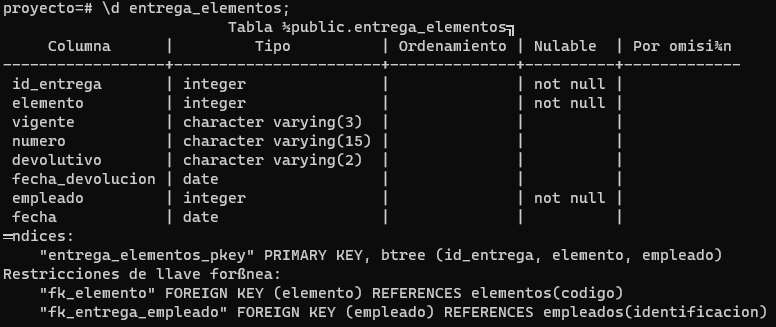
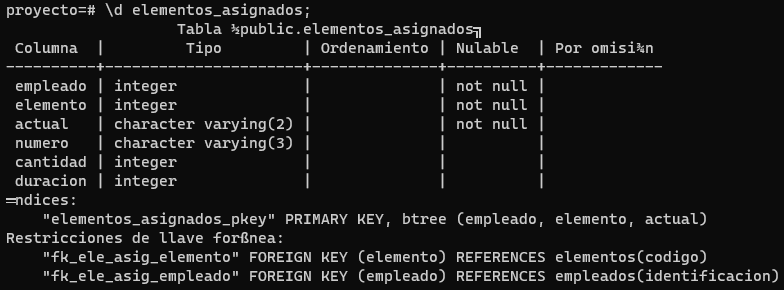
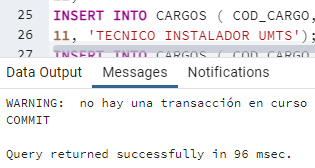


tabla elementos\_asignados:

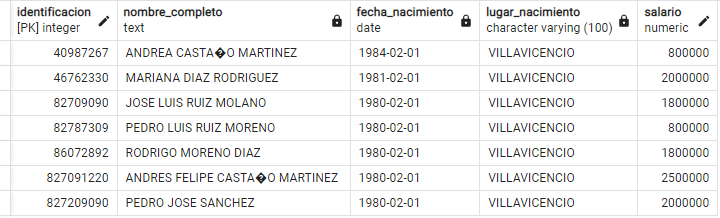


* Cargue la información del archivo misdatos.sql.



* Realice las consultas necesarias para obtener la siguiente información de la base de datos:

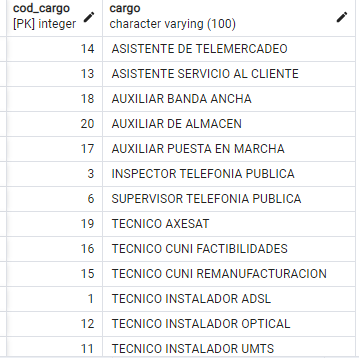
1. Listado de empleados: Identificación, nombres y apellidos en una sola columna, fecha de nacimiento, lugar de nacimiento y salario. Los datos deben estar ordenados por la identificación.
2. **SELECT** emp.identificacion,
3. CONCAT\_WS(' ',emp.nombre\_1,emp.nombre\_2,emp.apellido\_1,emp.apellido\_2) **AS** nombre\_completo,
4. emp.fecha\_n **AS** fecha\_nacimiento, emp.lugar\_n **AS** lugar\_nacimiento, emp.salario
5. **FROM** empleados **AS** emp
6. **ORDER** **BY** emp.identificacion;



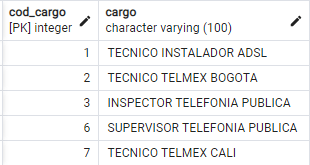
1. Listado de empleados: Identificación, nombres y apellidos, cargo y fecha ingreso. Los datos deben estar ordenados por la identificación y la fecha de ingreso de forma descendente.
2. **SELECT** emp.identificacion,
3. CONCAT\_WS(' ',emp.nombre\_1,emp.nombre\_2,emp.apellido\_1,emp.apellido\_2) **AS** nombre\_completo,
4. car.cargo, hist.fecha\_ingreso **FROM** empleados **AS** emp
5. **INNER** **JOIN** historial\_laboral **AS** hist **ON** hist.empleado = emp.identificacion
6. **INNER** **JOIN** cargos **AS** car **ON** car.cod\_cargo = hist.cargo
7. **ORDER** **BY** emp.identificacion **DESC**, hist.fecha\_ingreso **DESC**;



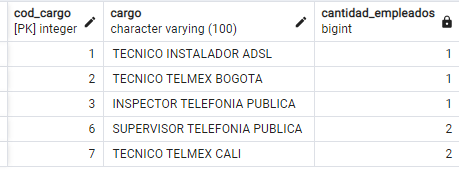
1. Listado de cargos.
2. **SELECT** \* **FROM** CARGOS;



1. Listado de cargos que actualmente están asignados
2. **SELECT** **DISTINCT** car.cod\_cargo, car.cargo **FROM** cargos **AS** car
3. **INNER** **JOIN** historial\_laboral **AS** hist **ON** hist.cargo = car.cod\_cargo
4. **AND** hist.actual ='S';



1. Listado de cargos que actualmente están asignados y la cantidad de empleados que lo tienen asignado.
2. **SELECT** car.cod\_cargo, car.cargo,
3. **COUNT**(hist.empleado) **AS** cantidad\_empleados
4. **FROM** cargos **AS** car
5. **INNER** **JOIN** historial\_laboral **AS** hist **ON** hist.cargo=car.cod\_cargo
6. **AND** hist.actual = 'S'
7. **GROUP** **BY** car.cod\_cargo;



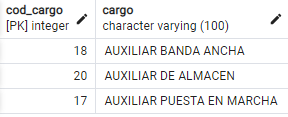
1. Historial laboral de cada empleado ordenado por empleado y la fecha de ingreso, debe incluir el nombre completo del empleado y la identificación.
2. **SELECT** emp.identificacion,
3. CONCAT\_WS(' ',emp.nombre\_1,emp.nombre\_2,emp.apellido\_1,emp.apellido\_2) **AS** nombre\_completo,
4. hist.fecha\_ingreso, hist.fecha\_salida, hist.cargo, hist.actual
5. **FROM** empleados **AS** emp
6. **INNER** **JOIN** historial\_laboral **AS** hist **ON** hist.empleado = emp.identificacion
7. **ORDER** **BY** hist.empleado, hist.fecha\_ingreso;



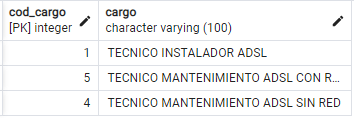
1. Listado de cargos que comiencen por la palabra “TECNICO”
2. **SELECT** \* **FROM** cargos **AS** car **WHERE** car.cargo **LIKE**'TECNICO%';



1. Listado de cargos que comiencen por la palabra “AUXILIAR”.
2. **SELECT** \* **FROM** cargos **AS** car **WHERE** car.cargo **LIKE**'AUXILIAR%';



1. Listado de cargos que contenga las letras “ADSL”.
2. **SELECT** \* **FROM** cargos **AS** car **WHERE** car.cargo **LIKE**'%ADSL%'



1. Listados de los empleados cuya fecha de salida sea nula, debe incluir el nombre completo del empleado y la identificación.
2. **SELECT** CONCAT\_WS(' ',emp.nombre\_1,emp.nombre\_2,emp.apellido\_1,emp.apellido\_2) **AS** nombre\_completo, emp.identificacion
3. **FROM** empleados **AS** emp
4. **INNER** **JOIN** historial\_laboral **AS** hist **ON** hist.empleado = emp.identificacion
5. **WHERE** hist.fecha\_salida **IS** **NULL**;



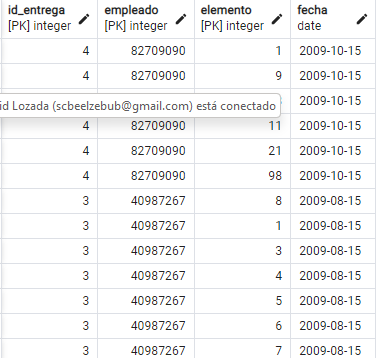
1. Elementos asignados a cada empleado que estén vigentes (actual=’S’), debe incluir el nombre completo del empleado, la identificación y el nombre del elemento asignado. Ordenar los resultados por identificación y elemento de forma ascendente.
2. **SELECT** CONCAT\_WS(' ',emp.nombre\_1,emp.nombre\_2,emp.apellido\_1,emp.apellido\_2) **AS** nombre\_completo, emp.identificacion,
3. el.elemento **FROM** empleados **AS** emp
4. **INNER** **JOIN** elementos\_asignados **AS** ela **ON** ela.empleado = emp.identificacion
5. **INNER** **JOIN** elementos **AS** el **ON** el.codigo = ela.elemento
6. **WHERE** ela.actual = 'S'
7. **ORDER** **BY** emp.identificacion **ASC,** el.elemento ASC;



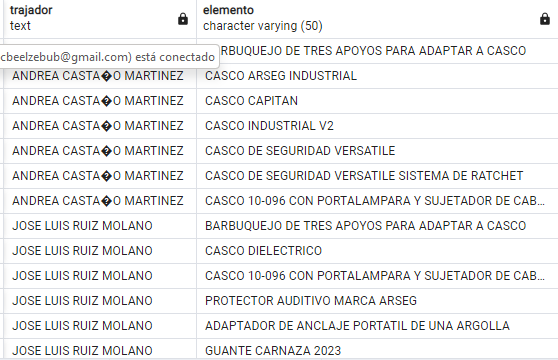
1. Historial de elementos asignados a cada empleado, debe incluir el nombre completo del empleado, la identificación, el nombre del elemento asignado, cantidad, duración y talla (es el campo número). Ordenar los resultados por identificación y elemento de forma ascendente.
2. **SELECT** CONCAT\_WS(' ',emp.nombre\_1,emp.nombre\_2,emp.apellido\_1,emp.apellido\_2) **AS** nombre\_completo, emp.identificacion,
3. el.elemento, ela.cantidad, ela.duracion, ela.numero **FROM** elementos\_asignados **AS** ela
4. **INNER** **JOIN** elementos **AS** el **ON** el.codigo = ela.elemento
5. **INNER** **JOIN** empleados **AS** emp **ON** emp.identificacion = ela.empleado
6. **ORDER** **BY** emp.identificacion **ASC,** el.elemento ASC;



1. Entregas realizadas, la cual debe incluir: el Id de entrega, el empleado, el elemento y la fecha de entrega, el listado debe estar ordenado por la fecha de entrega de forma descendente.
2. **SELECT** entre\_ele.id\_entrega, entre\_ele.empleado, entre\_ele.elemento, entre\_ele.fecha
3. **FROM** entrega\_elementos **AS** entre\_ele
4. **ORDER** **BY** entre\_ele.fecha **DESC**;



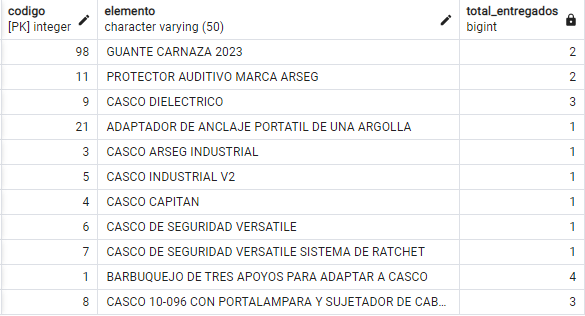
1. Elementos entregados a cada trabajador.
2. **SELECT** CONCAT\_WS(' ',emp.nombre\_1,emp.nombre\_2,emp.apellido\_1,emp.apellido\_2) **AS** trajador, ele.elemento
3. **FROM** entrega\_elementos **AS** entre\_ele
4. **INNER** **JOIN** empleados **AS** emp **ON** emp.identificacion = entre\_ele.empleado
5. **INNER** **JOIN** elementos **AS** ele **ON** ele.codigo = entre\_ele.elemento;



1. Total, de elementos entregados a cada trabajador, agrupados por elemento.
2. **SELECT** emp.identificacion, CONCAT\_WS(' ',emp.nombre\_1,emp.nombre\_2,emp.apellido\_1,emp.apellido\_2) **AS** nombre\_completo,
3. ele.elemento ,**COUNT**(entre\_ele.elemento) **AS** total\_entregado
4. **FROM** entrega\_elementos **AS** entre\_ele
5. **INNER** **JOIN** empleados **AS** emp **ON** emp.identificacion = entre\_ele.empleado
6. **INNER** **JOIN** elementos **AS** ele **ON** ele.codigo = entre\_ele.elemento
7. **GROUP** **BY** emp.identificacion,ele.elemento;



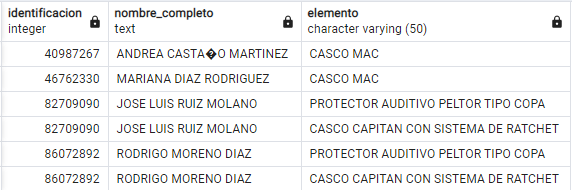
1. Total elementos entregados, agrupados por elementos.
2. **SELECT** ele.codigo, ele.elemento, **COUNT**(ele.codigo) **AS** total\_entregados
3. **FROM** elementos **AS** ele
4. **INNER** **JOIN** entrega\_elementos **AS** entre\_ele **ON** entre\_ele.elemento = ele.codigo
5. **GROUP** **BY** ele.codigo, ele.elemento;



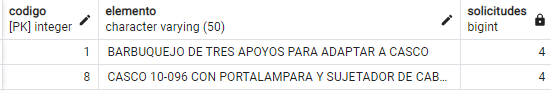
1. Ficha técnica de cada elemento, ordenado por el nombre del elemento en orden ascendente.
2. **SELECT** \* **FROM** elementos **AS** el **ORDER** **BY** el.elemento ASC;



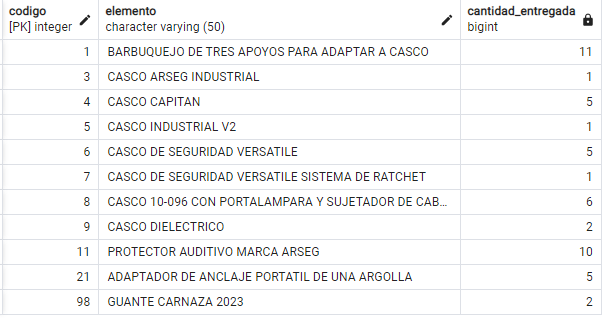
1. Listado de empleados y los elementos que les están pendientes por entregar.
2. **SELECT** emp.identificacion,
3. CONCAT\_WS(' ',emp.nombre\_1,emp.nombre\_2,emp.apellido\_1,emp.apellido\_2) **AS** nombre\_completo, e.elemento
4. **FROM** empleados **AS** emp
5. **INNER** **JOIN** elementos\_asignados **AS** ela **ON** ela.empleado = emp.identificacion
6. **INNER** **JOIN** elementos **AS** e **ON** e.codigo = ela.elemento
7. **LEFT** **JOIN** entrega\_elementos **AS** el **ON** el.elemento = e.codigo
8. **WHERE** el.elemento **IS** **NULL**
9. **ORDER** **BY** emp.identificacion;



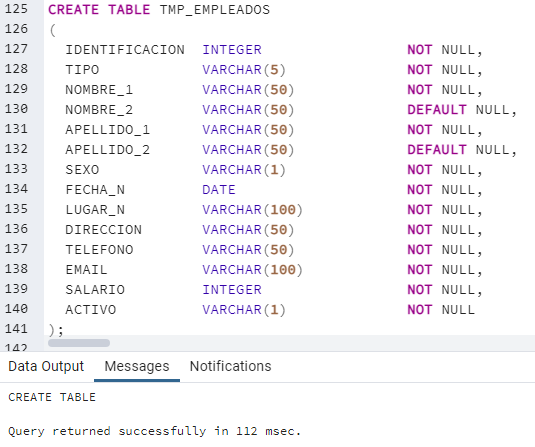
1. El elemento más solicitado.
2. **SELECT** el.codigo, el.elemento, **COUNT**(el.codigo) **AS** solicitudes
3. **FROM** elementos **AS** el
4. **INNER** **JOIN** elementos\_asignados **AS** ela **ON** ela.elemento = el.codigo
5. **GROUP** **BY** el.codigo, el.elemento
6. **HAVING** **COUNT**(el.codigo) = (**SELECT** **MAX**(cantidad.total) **FROM** (**SELECT** **COUNT**(elemento) **AS** TOTAL **FROM** elementos\_asignados **GROUP** **BY** elemento) **AS** cantidad);

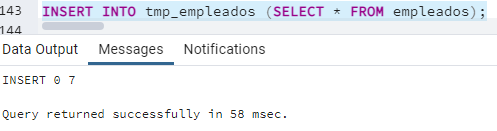


1. Listado de elementos entregados y la cantidad, durante el segundo periodo del año 2009.
2. **SELECT** el.codigo, el.elemento, **SUM**(ela.cantidad) **AS** cantidad\_entregada
3. **FROM** elementos **AS** el
4. **INNER** **JOIN** elementos\_asignados **AS** ela **ON** ela.elemento = el.codigo
5. **INNER** **JOIN** entrega\_elementos **AS** entre\_ele **ON** entre\_ele.elemento = ela.elemento **AND** entre\_ele.empleado = ela.empleado
6. **WHERE** entre\_ele.fecha **BETWEEN** '01/06/2009' **AND** '31/12/2009'
7. **GROUP** **BY** el.codigo, el.elemento;

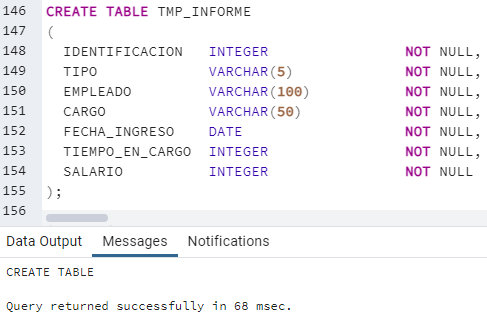


* Cree la siguiente tabla e inserte en ella los datos de la tabla empleado utilizando para ello la instrucción insert into mitabla\_destino (select \* from mitabla\_origen); .





* Cree la siguiente tabla e inserte en ella los datos requeridos, utilizando para ello la instrucción insert into mitabla\_destino (select \* from mitabla\_origen); .

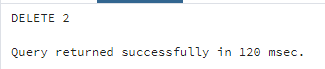


1. **INSERT** **INTO** tmp\_informe  (
2. **SELECT** emp.identificacion, emp.tipo, CONCAT\_WS(' ',emp.nombre\_1,emp.nombre\_2,emp.apellido\_1,emp.apellido\_2) **AS** empleado,
3. car.cargo, hs.fecha\_ingreso,
4. **CASE** **WHEN** hs.fecha\_salida
5. **IS** **NOT** **NULL** **THEN**
6. **ROUND**(hs.fecha\_salida-hs.fecha\_ingreso)
7. **ELSE**
8. **ROUND**(**CURRENT\_DATE**-hs.fecha\_ingreso)
9. **END** **AS** duracion
10. , emp.salario
11. **FROM** empleados **AS** emp
12. **INNER** **JOIN** historial\_laboral **AS** hs **ON** hs.empleado = emp.identificacion
13. **INNER** **JOIN**  cargos **AS** car **ON** car.cod\_cargo = hs.cargo
14. );



* Escriba la instrucción sql necesaria para eliminar los datos de los empleados de la tabla TMP\_INFORME Donde el cargo sea igual a “SUPERVISOR TELEFONIA PUBLICA”.

1. **DELETE** **FROM** tmp\_informe **WHERE** cargo = 'SUPERVISOR TELEFONIA PUBLICA';



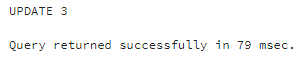
* Escriba la instrucción sql necesaria para actualizar la columna serial de la tabla elementos con el valor “S”.

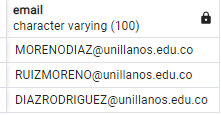
1. **UPDATE** elementos **SET** serial = 'S';



* Escriba las instrucciones sql necesarias para actualizar la columna email de la tabla empleados con la nueva dirección de correo asignada por la empresa. La nueva dirección asignada debe estar conformada por: La inicial del primer nombre + el primer apellido + el segundo apellido + unillanos.edu.co, esta actualización debe hacerla a los empleados con identificación 86072892, 82787309 y 46762330. Ejemplo: Ricardo Andrés Sandoval Morales -> rsandovalmorales@unillanos.edu.co

1. **UPDATE** empleados **SET** email = CONCAT(**SUBSTR**(nombre\_1,0,1),apellido\_1,apellido\_2,'@unillanos.edu.co')
2. **WHERE** identificacion = 86072892 **OR** identificacion = 82787309 **OR** identificacion = 46762330;





* Al ejecutar instrucciones sql de tipo DDL o DML, para qué caso debo confirmar una transacción (commit), de tal forma que los cambios sean permanentes.
  + Para instrucciones de tipo DDL
  + Para instrucciones de tipo DML

Explique el porqué de su respuesta e indique algunas instrucciones sobre las cuales debo confirmar las transacciones.

La instrucción commit para confirmar una transacción se debe usar para el caso de sentencias de tipo DML(Data Manipulation Language) y entonces así los cambios realizados serán permanentes.

Las sentencias De tipo DML son:

**SELECT:** se utiliza para realizar consultas y extraer información de la base de datos.

**INSERT:** se utiliza para insertar registros en las tablas de la base de datos.

**UPDATE**: se utiliza para actualizar los registros de una tabla.

**DELETE:** se utiliza para eliminar registros de una tabla.

Las instrucciones sobre las cuales se debe confirmar las transacciones son insert, update y delete

El siguiente ejemplo muestra una supuesta transacción bancaria:

**DECLARE**

importe NUMBER;

ctaOrigen VARCHAR2(23);

ctaDestino VARCHAR2(23);

**BEGIN**

importe :**=** 100;

ctaOrigen :**=** '2530 10 2000 1234567890';

ctaDestino :**=** '2532 10 2010 0987654321';

**UPDATE** CUENTAS **SET** SALDO **=** SALDO – importe **WHERE** CUENTA **=** ctaOrigen;

**UPDATE** CUENTAS **SET** SALDO **=** SALDO **+** importe **WHERE** CUENTA **=** ctaDestino;

**INSERT** **INTO** MOVIMIENTOS **VALUES** (ctaOrigen, ctaDestino, importe**\***(**-**1), SYSDATE);

**INSERT** **INTO** MOVIMIENTOS **VALUES** (ctaDestino,ctaOrigen, importe, SYSDATE);

**COMMIT**;

**EXCEPTION**

**WHEN** OTHERS **THEN**

dbms\_output.put\_line('Error en la transaccion:'**||**SQLERRM);

dbms\_output.put\_line('Se deshacen las modificaciones);

ROLLBACK;

END;