



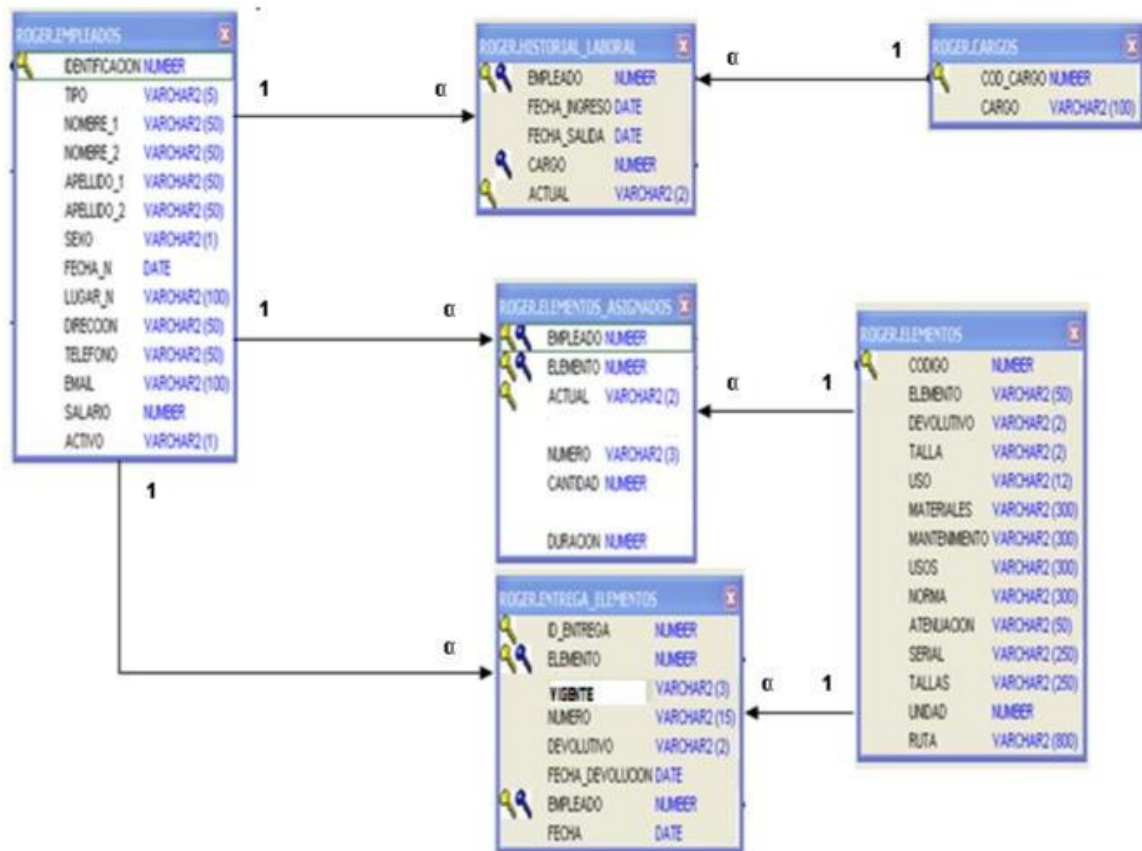
TALLER DE ADMINISTRACIÓN DE DATOS

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. );
8.
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. );
25.
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. );
36.
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.     );
53.
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) REFERE
NCES elementos(codigo),
65.         CONSTRAINT fk_entrega_empleado FOREIGN KEY(empleado
) REFERENCES empleados(identificacion)
66.     );
67.
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(emplead
o) REFERENCES empleados(identificacion),
77.         CONSTRAINT fk_ele_asig_elemento FOREIGN KEY(element
o) REFERENCES elementos(codigo)
78.     );
```



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

tabla cargos:

```
proyecto=# \d cargos;
               Tabla "public.cargos"
  Columna  | Tipo          | Ordenamiento | Nulable | Por omisión
-----+-----+-----+-----+-----
cod_cargo | integer       |              | not null |
cargo     | character varying(100) |              | not null |
Índices:
" cargos_pkey" PRIMARY KEY, btree (cod_cargo)
Referenciada por:
TABLE "historial_laboral" CONSTRAINT "fk_cargo" FOREIGN KEY (cargo) REFERENCES cargos(cod_cargo)
```

tabla empleados:

```
proyecto=# \d empleados;
               Tabla "public.empleados"
  Columna  | Tipo          | Ordenamiento | Nulable | Por omisión
-----+-----+-----+-----+-----
identificacion | integer       |              | not null |
tipo          | character varying(5) |              |          |
nombre_1      | character varying(50) |              |          |
nombre_2      | character varying(50) |              |          |
apellido_1     | character varying(50) |              |          |
apellido_2     | character varying(50) |              |          |
sexo          | character varying(1) |              |          |
fecha_n       | date          |              |          |
lugar_n       | character varying(100) |              |          |
direccion     | character varying(50) |              |          |
telefono      | character varying(50) |              |          |
email         | character varying(100) |              |          |
salario       | numeric       |              |          |
activo        | character varying(1) |              |          |
Índices:
"empleados_pkey" PRIMARY KEY, btree (identificacion)
Referenciada por:
TABLE "elementos_asignados" CONSTRAINT "fk_ele_asig_empleado" FOREIGN KEY (empleado) REFERENCES empleados(identificacion)
TABLE "historial_laboral" CONSTRAINT "fk_empleado" FOREIGN KEY (empleado) REFERENCES empleados(identificacion)
TABLE "entrega_elementos" CONSTRAINT "fk_entrega_empleado" FOREIGN KEY (empleado) REFERENCES empleados(identificacion)
```

tabla historial_laboral:



```
proyecto=# \d historial_laboral;
          Tabla "public.historial_laboral"
+-----+-----+-----+-----+-----+
| Columna | Tipo          | Ordenamiento | Nulable | Por omisión |
+-----+-----+-----+-----+-----+
| empleado | integer       |               | not null |              |
| fecha_ingreso | date         |               |          |              |
| fecha_salida | date         |               |          |              |
| cargo      | integer       |               | not null |              |
| actual     | character varying(2) |           | not null |              |
+-----+-----+-----+-----+-----+
Índices:
    "historial_laboral_pkey" PRIMARY KEY, btree (empleado, actual)
Restricciones de llave foránea:
    "fk_cargo" FOREIGN KEY (cargo) REFERENCES cargos(cod_cargo)
    "fk_empleado" FOREIGN KEY (empleado) REFERENCES empleados(identificacion)
```

tabla elementos:

```
proyecto=# \d elementos;
          Tabla "public.elementos"
+-----+-----+-----+-----+-----+
| Columna | Tipo          | Ordenamiento | Nulable | Por omisión |
+-----+-----+-----+-----+-----+
| codigo  | integer       |               | not null |              |
| elemento | character varying(50) |           |          |              |
| devolutivo | character varying(2) |           |          |              |
| talla   | character varying(2) |           |          |              |
| uso     | character varying(12) |           |          |              |
| materiales | character varying(300) |         |          |              |
| mantenimiento | character varying(300) |         |          |              |
| usos    | character varying(300) |         |          |              |
| norma   | character varying(300) |         |          |              |
| atenuacion | character varying(50) |           |          |              |
| serial  | character varying(250) |         |          |              |
| tallas  | character varying(250) |         |          |              |
| unidad  | integer       |               |          |              |
| ruta    | character varying(800) |         |          |              |
+-----+-----+-----+-----+-----+
Índices:
    "elementos_pkey" PRIMARY KEY, btree (codigo)
Referenciada por:
    TABLE "elementos_asignados" CONSTRAINT "fk_ele_asig_elemento" FOREIGN KEY (elemento) REFERENCES elementos(codigo)
    TABLE "entrega_elementos" CONSTRAINT "fk_elemento" FOREIGN KEY (elemento) REFERENCES elementos(codigo)
```

tabla entrega_elementos:



```
proyecto=# \d entrega_elementos;
          Tabla "public.entrega_elementos"
+-----+-----+-----+-----+-----+
| Columna | Tipo          | Ordenamiento | Nulable | Por omisión |
+-----+-----+-----+-----+-----+
| id_entrega | integer      |               | not null |              |
| elemento   | integer      |               | not null |              |
| vigente    | character varying(3) |               |          |              |
| numero     | character varying(15) |               |          |              |
| devolutivo | character varying(2) |               |          |              |
| fecha_devolucion | date        |               |          |              |
| empleado   | integer      |               | not null |              |
| fecha      | date         |               |          |              |
+-----+-----+-----+-----+-----+
Índices:
    "entrega_elementos_pkey" PRIMARY KEY, btree (id_entrega, elemento, empleado)
Restricciones de llave foránea:
    "fk_elemento" FOREIGN KEY (elemento) REFERENCES elementos(codigo)
    "fk_entrega_empleado" FOREIGN KEY (empleado) REFERENCES empleados(identificacion)
```

tabla elementos_asignados:

```
proyecto=# \d elementos_asignados;
          Tabla "public.elementos_asignados"
+-----+-----+-----+-----+-----+
| Columna | Tipo          | Ordenamiento | Nulable | Por omisión |
+-----+-----+-----+-----+-----+
| empleado | integer      |               | not null |              |
| elemento | integer      |               | not null |              |
| actual   | character varying(2) |               | not null |              |
| numero   | character varying(3) |               |          |              |
| cantidad | integer      |               |          |              |
| duracion | integer      |               |          |              |
+-----+-----+-----+-----+-----+
Índices:
    "elementos_asignados_pkey" PRIMARY KEY, btree (empleado, elemento, actual)
Restricciones de llave foránea:
    "fk_ele_asig_elemento" FOREIGN KEY (elemento) REFERENCES elementos(codigo)
    "fk_ele_asig_empleado" FOREIGN KEY (empleado) REFERENCES empleados(identificacion)
```

- Cargue la información del archivo misdatos.sql.

```
25 INSERT INTO CARGOS ( COD_CARGO,
26 11, 'TECNICO INSTALADOR UMTS');
27 INSERT INTO CARGOS ( COD_CARGO,
```

Data Output Messages Notifications

WARNING: no hay una transacción en curso
COMMIT

Query returned successfully in 96 msec.



- 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.

```
1. SELECT emp.identificacion,  
2. CONCAT_WS(' ', emp.nombre_1, emp.nombre_2, emp.apellido_1, emp.apellido_2) AS nombre_completo,  
3. emp.fecha_n AS fecha_nacimiento, emp.lugar_n AS lugar_nacimiento, emp.salario  
4. FROM empleados AS emp  
5. ORDER BY emp.identificacion;
```

identificacion [PK] integer	nombre_completo text	fecha_nacimiento date	lugar_nacimiento character varying (100)	salario numeric
40987267	ANDREA CASTAÑO MARTINEZ	1984-02-01	VILLAVICENCIO	800000
46762330	MARIANA DIAZ RODRIGUEZ	1981-02-01	VILLAVICENCIO	2000000
82709090	JOSE LUIS RUIZ MOLANO	1980-02-01	VILLAVICENCIO	1800000
82787309	PEDRO LUIS RUIZ MORENO	1980-02-01	VILLAVICENCIO	800000
86072892	RODRIGO MORENO DIAZ	1980-02-01	VILLAVICENCIO	1800000
827091220	ANDRES FELIPE CASTAÑO MARTINEZ	1980-02-01	VILLAVICENCIO	2500000
827209090	PEDRO JOSE SANCHEZ	1980-02-01	VILLAVICENCIO	2000000

2. 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.

```
1. SELECT emp.identificacion,  
2. CONCAT_WS(' ', emp.nombre_1, emp.nombre_2, emp.apellido_1, emp.apellido_2) AS nombre_completo,  
3. car.cargo, hist.fecha_ingreso FROM empleados AS emp  
4. INNER JOIN historial_laboral AS hist ON hist.empleado = emp.identificacion  
5. INNER JOIN cargos AS car ON car.cod_cargo = hist.cargo  
6. ORDER BY emp.identificacion DESC, hist.fecha_ingreso DESC;
```




identificacion integer	nombre_completo text	cargo character varying (100)	fecha_ingreso date
827209090	PEDRO JOSE SANCHEZ	TECNICO TELMEX CALI	2002-02-01
827209090	PEDRO JOSE SANCHEZ	TECNICO INSTALADOR OPTICAL	2001-02-01
827091220	ANDRES FELIPE CASTAÑO MARTINEZ	SUPERVISOR TELEFONIA PUBLICA	2002-02-02
827091220	ANDRES FELIPE CASTAÑO MARTINEZ	TECNICO INSTALADOR UMTS	2001-02-01
86072892	RODRIGO MORENO DIAZ	TECNICO INSTALADOR ADSL	2000-01-01
82787309	PEDRO LUIS RUIZ MORENO	INSPECTOR TELEFONIA PUBLICA	2002-02-02
82787309	PEDRO LUIS RUIZ MORENO	TECNICO INSTALADOR ADSL	2001-02-01
82709090	JOSE LUIS RUIZ MOLANO	TECNICO TELMEX BOGOTA	2001-02-01
46762330	MARIANA DIAZ RODRIGUEZ	TECNICO TELMEX CALI	2002-02-02
46762330	MARIANA DIAZ RODRIGUEZ	TECNICO TELMEX MEDELLIN	2001-02-01
40987267	ANDREA CASTAÑO MARTINEZ	SUPERVISOR TELEFONIA PUBLICA	2002-02-02
40987267	ANDREA CASTAÑO MARTINEZ	TECNICO INSTALADOR ADSL	2001-02-01

3. Listado de cargos.

```
1. SELECT * FROM CARGOS;
```

cod_cargo [PK] integer	cargo character varying (100)
14	ASISTENTE DE TELEMERCADERO
13	ASISTENTE SERVICIO AL CLIENTE
18	AUXILIAR BANDA ANCHA
20	AUXILIAR DE ALMACEN
17	AUXILIAR PUESTA EN MARCHA
3	INSPECTOR TELEFONIA PUBLICA
6	SUPERVISOR TELEFONIA PUBLICA
19	TECNICO AXESAT
16	TECNICO CUNI FACTIBILIDADES
15	TECNICO CUNI REMANUFACTURACION
1	TECNICO INSTALADOR ADSL
12	TECNICO INSTALADOR OPTICAL
11	TECNICO INSTALADOR UMTS



4. Listado de cargos que actualmente están asignados

```
1. SELECT DISTINCT car.cod_cargo, car.cargo FROM cargos AS car
2. INNER JOIN historial_laboral AS hist ON hist.cargo = car.cod_carg
   o
3. AND hist.actual = 'S';
```

cod_cargo [PK] integer	cargo character varying (100)
1	TECNICO INSTALADOR ADSL
2	TECNICO TELMEX BOGOTA
3	INSPECTOR TELEFONIA PUBLICA
6	SUPERVISOR TELEFONIA PUBLICA
7	TECNICO TELMEX CALI

5. Listado de cargos que actualmente están asignados y la cantidad de empleados que lo tienen asignado.

```
1. SELECT car.cod_cargo, car.cargo,
2. COUNT(hist.empleado) AS cantidad_empleados
3. FROM cargos AS car
4. INNER JOIN historial_laboral AS hist ON hist.cargo=car.cod_cargo
5. AND hist.actual = 'S'
6. GROUP BY car.cod_cargo;
```

cod_cargo [PK] integer	cargo character varying (100)	cantidad_empleados bigint
1	TECNICO INSTALADOR ADSL	1
2	TECNICO TELMEX BOGOTA	1
3	INSPECTOR TELEFONIA PUBLICA	1
6	SUPERVISOR TELEFONIA PUBLICA	2
7	TECNICO TELMEX CALI	2

6. Historial laboral de cada empleado ordenado por empleado y la fecha de ingreso, debe incluir el nombre completo del empleado y la identificación.



```
1. SELECT emp.identificacion,  
2. CONCAT_WS(' ',emp.nombre_1,emp.nombre_2,emp.apellido_1,emp.apellido_2) AS nom  
bre_completo,  
3. hist.fecha_ingreso, hist.fecha_salida, hist.cargo, hist.actual  
4. FROM empleados AS emp  
5. INNER JOIN historial_laboral AS hist ON hist.empleado = emp.ident  
ificacion  
6. ORDER BY hist.empleado, hist.fecha_ingreso;
```

identificacion integer	nombre_completo text	fecha_ingreso date	fecha_salida date	cargo integer	actual character varying (2)
40987267	ANDREA CASTAÑO MARTINEZ	2001-02-01	2001-06-01	1	N
40987267	A Edgar David Lozada (scbeelzebub@gmail.com) está conectado		[null]	6	S
46762330	MARIANA DIAZ RODRIGUEZ	2001-02-01	2001-12-01	10	N
46762330	MARIANA DIAZ RODRIGUEZ	2002-02-02	[null]	7	S
82709090	JOSE LUIS RUIZ MOLANO	2001-02-01	2001-06-01	2	S
82787309	PEDRO LUIS RUIZ MORENO	2001-02-01	2001-12-01	1	N
82787309	PEDRO LUIS RUIZ MORENO	2002-02-02	[null]	3	S
86072892	RODRIGO MORENO DIAZ	2000-01-01	2001-02-01	1	S
827091220	ANDRES FELIPE CASTAÑO MARTINEZ	2001-02-01	2001-12-01	11	N
827091220	ANDRES FELIPE CASTAÑO MARTINEZ	2002-02-02	[null]	6	S
827209090	PEDRO JOSE SANCHEZ	2001-02-01	2001-12-01	12	N
827209090	PEDRO JOSE SANCHEZ	2002-02-01	[null]	7	S

7. Listado de cargos que comiencen por la palabra “TECNICO”

```
1. SELECT * FROM cargos AS car WHERE car.cargo LIKE 'TECNICO%';
```



cod_cargo [PK] integer	cargo character varying (100)
19	TECNICO AXESAT
16	TECNICO CUNI FACTIBILIDADES
15	TECNICO CUNI REMANUFACTURACION
1	TECNICO INSTALADOR ADSL
12	TECNICO INSTALADOR OPTICAL
11	TECNICO INSTALADOR UMTS
5	TECNICO MANTENIMIENTO ADSL CON R...
4	TECNICO MANTENIMIENTO ADSL SIN RED
8	TECNICO TELMEX BARRANQUILLA
2	TECNICO TELMEX BOGOTA
7	TECNICO TELMEX CALI
9	TECNICO TELMEX CARTAGENA
10	TECNICO TELMEX MEDELLIN

8. Listado de cargos que comiencen por la palabra “AUXILIAR”.

```
1. SELECT * FROM cargos AS car WHERE car.cargo LIKE 'AUXILIAR%';
```

cod_cargo [PK] integer	cargo character varying (100)
18	AUXILIAR BANDA ANCHA
20	AUXILIAR DE ALMACEN
17	AUXILIAR PUESTA EN MARCHA

9. Listado de cargos que contenga las letras “ADSL”.

```
1. SELECT * FROM cargos AS car WHERE car.cargo LIKE '%ADSL%';
```

cod_cargo [PK] integer	cargo character varying (100)
1	TECNICO INSTALADOR ADSL
5	TECNICO MANTENIMIENTO ADSL CON R...
4	TECNICO MANTENIMIENTO ADSL SIN RED



10. Listados de los empleados cuya fecha de salida sea nula, debe incluir el nombre completo del empleado y la identificación.

```
1. SELECT CONCAT_WS(' ', emp.nombre_1, emp.nombre_2, emp.apellido_1, emp.apellido_2) AS nombre_completo, emp.identificacion
2. FROM empleados AS emp
3. INNER JOIN historial_laboral AS hist ON hist.empleado = emp.identificacion
4. WHERE hist.fecha_salida IS NULL;
```

nombre_completo text	identificacion [PK] integer
PEDRO LUIS RUIZ MORENO	82787309
PEDRO JOSE SANCHEZ	827209090
ANDRES FELIPE CASTAÑO MARTINEZ	827091220
ANDREA CASTAÑO MARTINEZ	40987267
MARIANA DIAZ RODRIGUEZ	46762330

11. 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.

```
1. SELECT CONCAT_WS(' ', emp.nombre_1, emp.nombre_2, emp.apellido_1, emp.apellido_2) AS nombre_completo, emp.identificacion,
2. el.elemento FROM empleados AS emp
3. INNER JOIN elementos_asignados AS ela ON ela.empleado = emp.identificacion
4. INNER JOIN elementos AS el ON el.codigo = ela.elemento
5. WHERE ela.actual = 'S'
6. ORDER BY emp.identificacion ASC, el.elemento ASC;
```



nombre_completo text	identificacion integer	elemento character varying (50)
ANDREA CASTAÑO MARTINEZ	40987267	BARBUQUEJO DE TRES APOYOS PARA ADAPTAR A CASCO
ANDREA CASTAÑO MARTINEZ	40987267	CASCO 10-096 CON PORTALAMPARA Y SUJETADOR DE CAB...
ANDREA CASTAÑO MARTINEZ	40987267	CASCO ARSEG INDUSTRIAL
ANDREA CASTAÑO MARTINEZ	40987267	CASCO CAPITAN
ANDREA CASTAÑO MARTINEZ	40987267	CASCO DE SEGURIDAD VERSATILE
ANDREA CASTAÑO MARTINEZ	40987267	CASCO DE SEGURIDAD VERSATILE SISTEMA DE RATCHET
ANDREA CASTAÑO MARTINEZ	40987267	CASCO INDUSTRIAL V2
ANDREA CASTAÑO MARTINEZ	40987267	CASCO MAC
MARIANA DIAZ RODRIGUEZ	46762330	BARBUQUEJO DE TRES APOYOS PARA ADAPTAR A CASCO
MARIANA DIAZ RODRIGUEZ	46762330	CASCO 10-096 CON PORTALAMPARA Y SUJETADOR DE CAB...
MARIANA DIAZ RODRIGUEZ	46762330	CASCO ARSEG INDUSTRIAL
MARIANA DIAZ RODRIGUEZ	46762330	CASCO CAPITAN
MARIANA DIAZ RODRIGUEZ	46762330	CASCO DE SEGURIDAD VERSATILE

12. 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.

```
1. SELECT CONCAT_WS(' ', emp.nombre_1, emp.nombre_2, emp.apellido_1, emp.apellido_2) AS nombre_completo, emp.identificacion,
2. el.elemento, ela.cantidad, ela.duracion,
   ela.numero FROM elementos_asignados AS ela
3. INNER JOIN elementos AS el ON el.codigo = el.elemento
4. INNER JOIN empleados AS emp ON emp.identificacion = ela.empleado
5. ORDER BY emp.identificacion ASC, el.elemento ASC;
```



nombre_completo text	identificacion integer	elemento character varying (50)	cantidad integer	dur inte
ANDREA CASTAÑO MARTINEZ	40987267	BARBUQUEJO DE TRES APOYOS PARA ADAPTAR A CASCO	1	
ANDREA CASTAÑO MARTINEZ	40987267	CASCO 10-096 CON PORTALAMPARA Y SUJETADOR DE CAB...	5	
ANDREA CASTAÑO MARTINEZ	40987267	CASCO ARSEG INDUSTRIAL	1	
ANDREA CASTAÑO MARTINEZ	40987267	CASCO CAPITAN	5	
ANDREA CASTAÑO MARTINEZ	40987267	CASCO DE SEGURIDAD VERSATILE	5	
ANDREA CASTAÑO MARTINEZ	40987267	CASCO DE SEGURIDAD VERSATILE SISTEMA DE RATCHET	1	
ANDREA CASTAÑO MARTINEZ	40987267	CASCO INDUSTRIAL V2	1	
ANDREA CASTAÑO MARTINEZ	40987267	CASCO MAC	5	
MARIANA DIAZ RODRIGUEZ	46762330	BARBUQUEJO DE TRES APOYOS PARA ADAPTAR A CASCO	1	
MARIANA DIAZ RODRIGUEZ	46762330	CASCO 10-096 CON PORTALAMPARA Y SUJETADOR DE CAB...	5	
MARIANA DIAZ RODRIGUEZ	46762330	CASCO ARSEG INDUSTRIAL	1	
MARIANA DIAZ RODRIGUEZ	46762330	CASCO CAPITAN	5	

13. 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.

```
1. SELECT entre_ele.id_entrega, entre_ele.empleado,  
entre_ele.elemento, entre_ele.fecha  
2. FROM entrega_elementos AS entre_ele  
3. ORDER BY entre_ele.fecha DESC;
```

id_entrega [PK] integer	empleado [PK] integer	elemento [PK] integer	fecha date
4	82709090	1	2009-10-15
4	82709090	9	2009-10-15
id Lozada (scbeelzebub@gmail.com) está conectado			2009-10-15
4	82709090	11	2009-10-15
4	82709090	21	2009-10-15
4	82709090	98	2009-10-15
3	40987267	8	2009-08-15
3	40987267	1	2009-08-15
3	40987267	3	2009-08-15
3	40987267	4	2009-08-15
3	40987267	5	2009-08-15
3	40987267	6	2009-08-15
3	40987267	7	2009-08-15



14. Elementos entregados a cada trabajador.

```
1. SELECT CONCAT_WS(' ', emp.nombre_1, emp.nombre_2, emp.apellido_1, emp.apellido_2) AS trabajador, ele.elemento
2. FROM entrega_elementos AS entre_ele
3. INNER JOIN empleados AS emp ON emp.identificacion = entre_ele.empleado
4. INNER JOIN elementos AS ele ON ele.codigo = entre_ele.elemento;
```

trabajador text	elemento character varying (50)
cbeelzebub@gmail.com) está conectado	
ANDREA CASTAÑO MARTINEZ	CASCO ARSEG INDUSTRIAL
ANDREA CASTAÑO MARTINEZ	CASCO CAPITAN
ANDREA CASTAÑO MARTINEZ	CASCO INDUSTRIAL V2
ANDREA CASTAÑO MARTINEZ	CASCO DE SEGURIDAD VERSATILE
ANDREA CASTAÑO MARTINEZ	CASCO DE SEGURIDAD VERSATILE SISTEMA DE RATCHET
ANDREA CASTAÑO MARTINEZ	CASCO 10-096 CON PORTALAMPARA Y SUJETADOR DE CAB...
JOSE LUIS RUIZ MOLANO	BARBUQUEJO DE TRES APOYOS PARA ADAPTAR A CASCO
JOSE LUIS RUIZ MOLANO	CASCO DIELECTRICO
JOSE LUIS RUIZ MOLANO	CASCO 10-096 CON PORTALAMPARA Y SUJETADOR DE CAB...
JOSE LUIS RUIZ MOLANO	PROTECTOR AUDITIVO MARCA ARSEG
JOSE LUIS RUIZ MOLANO	ADAPTADOR DE ANCLAJE PORTATIL DE UNA ARGOLLA
JOSE LUIS RUIZ MOLANO	GUANTE CARNAZA 2023

15. Total, de elementos entregados a cada trabajador, agrupados por elemento.

```
1. SELECT emp.identificacion, CONCAT_WS(' ', emp.nombre_1, emp.nombre_2, emp.apellido_1, emp.apellido_2) AS nombre_completo,
2. ele.elemento, COUNT(entre_ele.elemento) AS total_entregado
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
```




```
6. GROUP BY emp.identificacion,ele.elemento;
```

identificacion integer	nombre_completo text	elemento character varying (50)	total_entregado bigint
86072892	RODRIGO MORENO DIAZ	BARBUQUEJO DE TRES APOYOS PARA ADAPTAR A CASCO	2
82709090	JOSE LUIS RUIZ MOLANO	CASCO 10-096 CON PORTALAMPARA Y SUJETADOR DE CAB...	1
40987267	ANDREA CASTAÑO MARTINEZ	CASCO INDUSTRIAL V2	1
40987267	ANDREA CASTAÑO MARTINEZ	CASCO DE SEGURIDAD VERSATILE	1
40987267	ANDREA CASTAÑO MARTINEZ	CASCO ARSEG INDUSTRIAL	1
82709090	JOSE LUIS RUIZ MOLANO	ADAPTADOR DE ANCLAJE PORTATIL DE UNA ARGOLLA	1
82709090	JOSE LUIS RUIZ MOLANO	GUANTE CARNAZA 2023	1
40987267	ANDREA CASTAÑO MARTINEZ	CASCO 10-096 CON PORTALAMPARA Y SUJETADOR DE CAB...	1
82709090	JOSE LUIS RUIZ MOLANO	CASCO DIELECTRICO	1
86072892	RODRIGO MORENO DIAZ	CASCO DIELECTRICO	2
86072892	RODRIGO MORENO DIAZ	PROTECTOR AUDITIVO MARCA ARSEG	1
40987267	ANDREA CASTAÑO MARTINEZ	BARBUQUEJO DE TRES APOYOS PARA ADAPTAR A CASCO	1

16. Total elementos entregados, agrupados por elementos.

```
1. SELECT ele.codigo,  
   ele.elemento, COUNT(ele.codigo) AS total_entregados  
2. FROM elementos AS ele  
3. INNER JOIN entrega_elementos AS entre_ele ON entre_ele.elemento =  
   ele.codigo  
4. GROUP BY ele.codigo, ele.elemento;
```

codigo [PK] integer	elemento character varying (50)	total_entregados bigint
98	GUANTE CARNAZA 2023	2
11	PROTECTOR AUDITIVO MARCA ARSEG	2
9	CASCO DIELECTRICO	3
21	ADAPTADOR DE ANCLAJE PORTATIL DE UNA ARGOLLA	1
3	CASCO ARSEG INDUSTRIAL	1
5	CASCO INDUSTRIAL V2	1
4	CASCO CAPITAN	1
6	CASCO DE SEGURIDAD VERSATILE	1
7	CASCO DE SEGURIDAD VERSATILE SISTEMA DE RATCHET	1
1	BARBUQUEJO DE TRES APOYOS PARA ADAPTAR A CASCO	4
8	CASCO 10-096 CON PORTALAMPARA Y SUJETADOR DE CAB...	3



17. Ficha técnica de cada elemento, ordenado por el nombre del elemento en orden ascendente.

```
1. SELECT * FROM elementos AS el ORDER BY el.elemento ASC;
```

codigo [PK] integer	elemento character varying (50)	devolutivo character varying (2)	talla character varying (2)	us ch
59	ABRIGO EN PVC CON CAPUCHA	No	No	In
19	ADAPTADOR DE ANCLAJE	No	No	In
21	ADAPTADOR DE ANCLAJE PORTATIL DE UNA ARGOLLA	No	No	In
20	ADAPTADOR DE ANCLAJE PORTATIL REFORZADO DE DOS ARG	No	No	In
96	ANTEOJO	No	No	In
107	ANTEOJO DE SEGURIDAD AR044C	No	No	In
104	ANTEOJO PARA SOLDADURA AUTOGENA	No	No	In
29	ARNES DETENC DE CAIDAS CON ANILLO DORSAL EN REATA	No	No	In
24	ARNES CUERPO ENTERO CON ARGOLLA DORSAL EN D	No	No	Gi
23	ARNES CUERPO ENTERO CON CINTURON SOPORTE LUMBAR	No	No	In
41	ARNES DE CUERPO ENTERO CON CINT. LUMBAL Y PORTAHE	No	No	In
33	ARNES DE POSIC, DETENCION Y RESTRIC. DE CAIDAS	No	No	In

18. Listado de empleados y los elementos que les están pendientes por entregar.

```
1. SELECT emp.identificacion,  
2. CONCAT_WS(' ', emp.nombre_1, emp.nombre_2, emp.apellido_1, emp.apellido_2) AS nom  
bre_completo, e.elemento  
3. FROM empleados AS emp  
4. INNER JOIN elementos_asignados AS ela ON ela.empleado = emp.ident  
ificacion  
5. INNER JOIN elementos AS e ON e.codigo = ela.elemento  
6. LEFT JOIN entrega_elementos AS el ON el.elemento = e.codigo  
7. WHERE el.elemento IS NULL  
8. ORDER BY emp.identificacion;
```



identificacion integer	nombre_completo text	elemento character varying (50)
40987267	ANDREA CASTAÑO MARTINEZ	CASCO MAC
46762330	MARIANA DIAZ RODRIGUEZ	CASCO MAC
82709090	JOSE LUIS RUIZ MOLANO	PROTECTOR AUDITIVO PELTOR TIPO COPA
82709090	JOSE LUIS RUIZ MOLANO	CASCO CAPITAN CON SISTEMA DE RATCHET
86072892	RODRIGO MORENO DIAZ	PROTECTOR AUDITIVO PELTOR TIPO COPA
86072892	RODRIGO MORENO DIAZ	CASCO CAPITAN CON SISTEMA DE RATCHET

19. El elemento más solicitado.

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

codigo [PK] integer	elemento character varying (50)	solicitudes bigint
1	BARBUQUEJO DE TRES APOYOS PARA ADAPTAR A CASCO	4
8	CASCO 10-096 CON PORTALAMPARA Y SUJETADOR DE CAB...	4

20. Listado de elementos entregados y la cantidad, durante el segundo periodo del año 2009.

```
1. SELECT el.codigo,
   el.elemento, SUM(ela.cantidad) AS cantidad_entregada
2. FROM elementos AS el
3. INNER JOIN elementos_asignados AS ela ON ela.elemento = el.codigo
4. INNER JOIN entrega_elementos AS entre_ele ON entre_ele.elemento =
   ela.elemento AND entre_ele.empleado = ela.empleado
5. WHERE entre_ele.fecha BETWEEN '01/06/2009' AND '31/12/2009'
6. GROUP BY el.codigo, el.elemento;
```



codigo [PK] integer	elemento character varying (50)	cantidad_entregada bigint
1	BARBUQUEJO DE TRES APOYOS PARA ADAPTAR A CASCO	11
3	CASCO ARSEG INDUSTRIAL	1
4	CASCO CAPITAN	5
5	CASCO INDUSTRIAL V2	1
6	CASCO DE SEGURIDAD VERSATILE	5
7	CASCO DE SEGURIDAD VERSATILE SISTEMA DE RATCHET	1
8	CASCO 10-096 CON PORTALAMPARA Y SUJETADOR DE CAB...	6
9	CASCO DIELECTRICO	2
11	PROTECTOR AUDITIVO MARCA ARSEG	10
21	ADAPTADOR DE ANCLAJE PORTATIL DE UNA ARGOLLA	5
98	GUANTE CARNAZA 2023	2

- 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);

```
125 CREATE TABLE TMP_EMPLEADOS
126 (
127     IDENTIFICACION INTEGER NOT NULL,
128     TIPO VARCHAR(5) NOT NULL,
129     NOMBRE_1 VARCHAR(50) NOT NULL,
130     NOMBRE_2 VARCHAR(50) DEFAULT NULL,
131     APELLIDO_1 VARCHAR(50) NOT NULL,
132     APELLIDO_2 VARCHAR(50) DEFAULT NULL,
133     SEXO VARCHAR(1) NOT NULL,
134     FECHA_N DATE NOT NULL,
135     LUGAR_N VARCHAR(100) NOT NULL,
136     DIRECCION VARCHAR(50) NOT NULL,
137     TELEFONO VARCHAR(50) NOT NULL,
138     EMAIL VARCHAR(100) NOT NULL,
139     SALARIO INTEGER NOT NULL,
140     ACTIVO VARCHAR(1) NOT NULL
141 );
142
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 112 msec.



```
143 INSERT INTO tmp_empleados (SELECT * FROM empleados);  
144
```

Data Output Messages Notifications

INSERT 0 7

Query returned successfully in 58 msec.

- Cree la siguiente tabla e inserte en ella los datos requeridos, utilizando para ello la instrucción `insert into mitabla_destino (select * from mitabla_origen);` .

```
146 CREATE TABLE TMP_INFORME  
147 (  
148     IDENTIFICACION    INTEGER                NOT NULL,  
149     TIPO               VARCHAR(5)            NOT NULL,  
150     EMPLEADO          VARCHAR(100)          NOT NULL,  
151     CARGO             VARCHAR(50)           NOT NULL,  
152     FECHA_INGRESO     DATE                  NOT NULL,  
153     TIEMPO_EN_CARGO   INTEGER               NOT NULL,  
154     SALARIO           INTEGER               NOT NULL  
155 );  
156
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 68 msec.

```
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 emp  
   leado,  
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.    );
```

 empleado character varying (100)	 cargo character varying (50)	 fecha_ingreso date	 tiempo_en_cargo integer	 salario integer
PEDRO LUIS RUIZ MORENO	TECNICO INSTALADOR ADSL	2001-02-01	303	800000
PEDRO JOSE SANCHEZ	TECNICO INSTALADOR OPTICAL	2001-02-01	303	2000000
ANDRES FELIPE CASTAÑO MARTINEZ	TECNICO INSTALADOR UMTS	2001-02-01	303	2500000
ANDREA CASTAÑO MARTINEZ	TECNICO INSTALADOR ADSL	2001-02-01	120	800000
MARIANA DIAZ RODRIGUEZ	TECNICO TELMEX MEDELLIN	2001-02-01	303	2000000
RODRIGO MORENO DIAZ	TECNICO INSTALADOR ADSL	2000-01-01	397	1800000
JOSE LUIS RUIZ MOLANO	TECNICO TELMEX BOGOTA	2001-02-01	120	1800000
PEDRO LUIS RUIZ MORENO	INSPECTOR TELEFONIA PUBLICA	2002-02-02	7653	800000
PEDRO JOSE SANCHEZ	TECNICO TELMEX CALI	2002-02-01	7654	2000000
ANDRES FELIPE CASTAÑO MARTINEZ	SUPERVISOR TELEFONIA PUBLICA	2002-02-02	7653	2500000
ANDREA CASTAÑO MARTINEZ	SUPERVISOR TELEFONIA PUBLICA	2002-02-02	7653	800000
MARIANA DIAZ RODRIGUEZ	TECNICO TELMEX CALI	2002-02-02	7653	2000000

- 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';
```

```
DELETE 2
Query returned successfully in 120 msec.
```

- Escriba la instrucción sql necesaria para actualizar la columna serial de la tabla elementos con el valor "S".

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

```
UPDATE 124
Query returned successfully in 71 msec.
```



- 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;
```

UPDATE 3

Query returned successfully in 79 msec.

email	
character varying (100)	
MORENODIAZ@unillanos.edu.co	
RUIZMORENO@unillanos.edu.co	
DIAZRODRIGUEZ@unillanos.edu.co	

- 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;
```