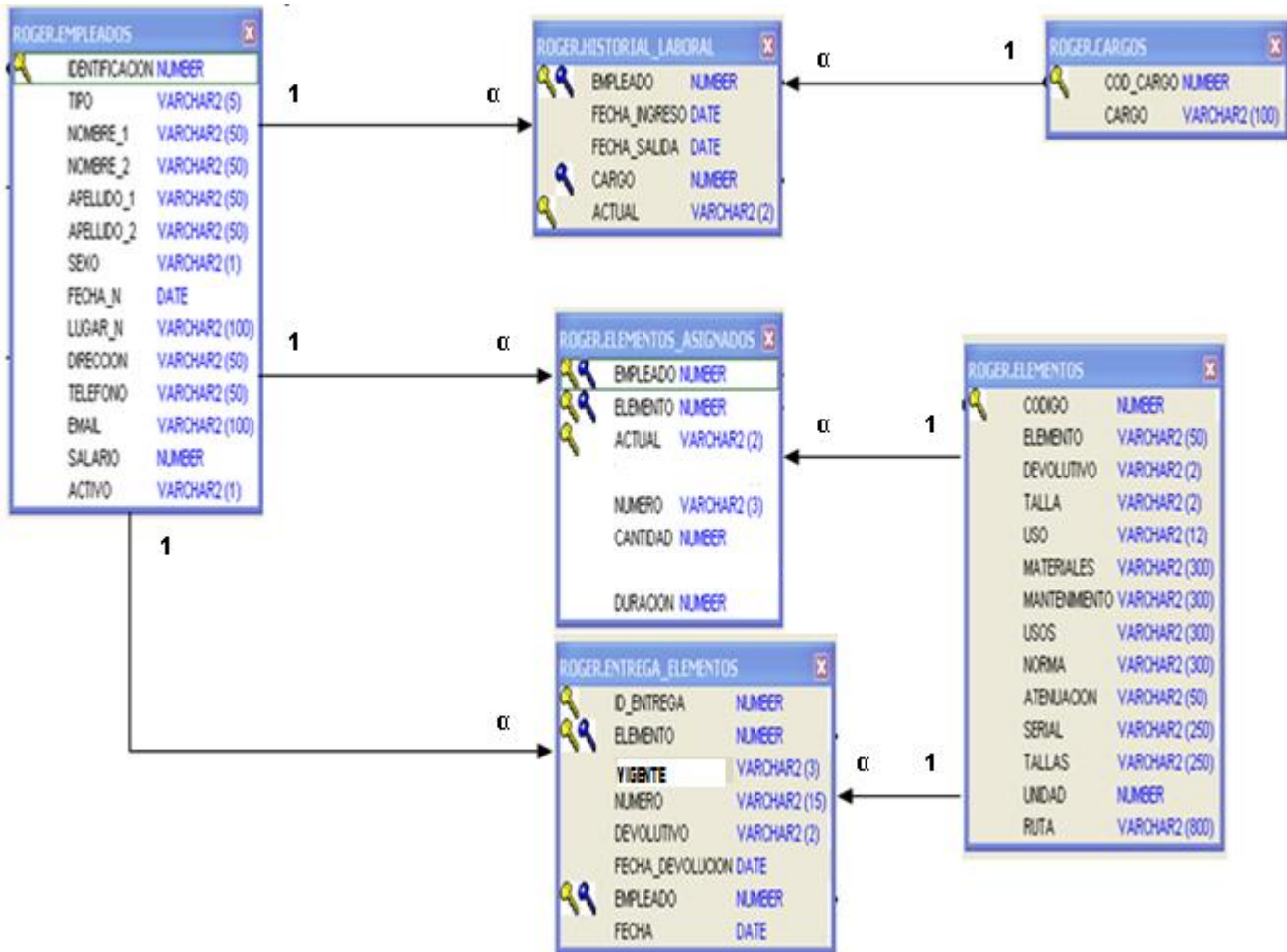


Taller 2.

- Dado el siguiente modelo entidad relación cree las tablas y las respectivas relaciones utilizando para ello la herramienta SqlDeveloper 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.



- Verifique la existencia de las tablas y describa la estructura de cada una.
- Cargue la información del archivo misdatos.sql.

ADMINISTRACIÓN DE DATOS

- 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. 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.
 3. Listado de cargos.
 4. Listado de cargos que actualmente están asignados.
 5. Listado de cargos que actualmente están asignados y la cantidad de empleados que lo tienen asignado.
 6. Historial laboral de cada empleado ordenado por empleado y la fecha de ingreso, debe incluir el nombre completo del empleado y la identificación.
 7. Listado de cargos que comiencen por la palabra "TECNICO".
 8. Listado de cargos que comiencen por la palabra "AUXILIAR".
 9. Listado de cargos que contenga las letras "ADSL".
 10. Listados de los empleados cuya fecha de salida sea nula, debe incluir el nombre completo del empleado y la identificación.
 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.
 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 numero). Ordenar los resultados por identificación y elemento de forma ascendente.
 13. Entregas realiza, 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.
 14. Elementos entregados a cada trabajador.
 15. Total de elementos entregados a cada trabajador, agrupados por elemento.
 16. Total elementos entregados, agrupados por elementos.
 17. Ficha técnica de cada elemento, ordenado por el nombre del elemento en orden ascendente.
 18. Listado de empleados y los elementos que les están pendientes por entregar.
 19. El elemento más solicitado.
 20. Listado de elementos entregados y la cantidad, durante el segundo periodo del año 2009.

ADMINISTRACIÓN DE DATOS

- 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);** .

```
CREATE TABLE TMP_EMPLEADOS
(
  IDENTIFICACION NUMBER NOT NULL,
  TIPO VARCHAR2(5 BYTE) NOT NULL,
  NOMBRE_1 VARCHAR2(50 BYTE) NOT NULL,
  NOMBRE_2 VARCHAR2(50 BYTE) DEFAULT NULL,
  APELLIDO_1 VARCHAR2(50 BYTE) NOT NULL,
  APELLIDO_2 VARCHAR2(50 BYTE) DEFAULT NULL,
  SEXO VARCHAR2(1 BYTE) NOT NULL,
  FECHA_N DATE NOT NULL,
  LUGAR_N VARCHAR2(100 BYTE) NOT NULL,
  DIRECCION VARCHAR2(50 BYTE) NOT NULL,
  TELEFONO VARCHAR2(50 BYTE) NOT NULL,
  EMAIL VARCHAR2(100 BYTE) NOT NULL,
  SALARIO INTEGER NOT NULL,
  ACTIVO VARCHAR2(1 BYTE) NOT NULL
);
```

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

```
CREATE TABLE TMP_INFORME
(
  IDENTIFICACION NUMBER NOT NULL,
  TIPO VARCHAR2(5 BYTE) NOT NULL,
  EMPLEADO VARCHAR2(100 BYTE) NOT NULL,
  CARGO VARCHAR2(50 BYTE) NOT NULL,
  FECHA_INGRESO DATE NOT NULL,
  TIEMPO_EN_CARGO INTEGER NOT NULL,
  SALARIO INTEGER NOT NULL
);
```

- 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".
- Escriba la instrucción sql necesaria para actualizar la columna serial de la tabla elementos con el valor "S".
- Escriba las instrucciones sql necesaria 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**
- Al ejecutar instrucciones sql de tipo DDL o DML, para que 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.

Entregar un documento en formato digital que contenga los pasos, operaciones y resultados que se deben realizar para cada actividad del taller.

ANEXO

```
create [global temporary] table usuario.nombre_tabla(  
    {nom_columna tipo_dato [default expresion] [restriccion_columna | restriccion de tabla]},  
    {nom_columna tipo_dato [default expresion] [restriccion_columna | restriccion de tabla]},  
    {nom_columna tipo_dato [default expresion] [restriccion_columna | restriccion de tabla]}  
);
```

-----EJEMPLOS-----

```
CREATE TABLE libro(  
    libro NUMBER(5) PRIMARY KEY,  
    titulo VARCHAR2(30) NOT NULL,  
    autor VARCHAR2(30),  
    autor2 VARCHAR2(30) DEFAULT 'NO ASIGNADO',  
    numpag NUMBER(4),  
    constraint num_paginas check (numpag>0)  
);
```

```
CREATE TABLE socio(  
    socio NUMBER(5) PRIMARY KEY,  
    tipo_documento varchar2(2) NOT NULL,  
    nombre VARCHAR2(30) NOT NULL,  
    telefono VARCHAR2(10),  
    ingreso DATE,  
    constraint mitipo_docu check ('CC','TI','CE','NIT')  
);
```

```
CREATE TABLE prestamo(  
    libro NUMBER(5) NOT NULL,  
    socio NUMBER(5) NOT NULL,  
    fprestamo DATE DEFAULT SYSDATE,  
    fdevolucion DATE,  
    PRIMARY KEY (libro,socio,fprestamo),  
    FOREIGN KEY (libro) REFERENCES libro,  
    FOREIGN KEY (socio) REFERENCES socio  
);
```

ADMINISTRACIÓN DE DATOS

```
create table pedidos
(
    ID_PEDIDO      NUMBER(4) NOT NULL,
    EMPLEADO       INTEGER NOT NULL,
    ELEMENTO       INTEGER NOT NULL,
    CANTIDAD       NUMBER(10) DEFAULT 0 NOT NULL CONSTRAINT CANTIDAD_CHECK CHECK ((CANTIDAD>=0)),
    APROBADO       VARCHAR2(2) DEFAULT 'N' NOT NULL CONSTRAINT APROBADO_CHECK CHECK (APROBADO in ('S','N')),
    FECHA          DATE          DEFAULT SYSDATE NOT NULL,
    LLEGO          VARCHAR2(2) DEFAULT 'N' NOT NULL CONSTRAINT LLEGO_CHECK CHECK (LLEGO IN ('S','N')),
    CONSTRAINT pedidos_pk PRIMARY KEY (ID_PEDIDO,EMPLEADO,ELEMENTO),
    CONSTRAINT empleado_fk FOREIGN KEY (EMPLEADO) REFERENCES EMPLEADOS(IDENTIFICACION),
    CONSTRAINT elemento_fk FOREIGN KEY (ELEMENTO) REFERENCES ELEMENTOS(CODIGO)
);
```

```
create table elementos_asignados
(
    empleado number,
    elemento number,
    actual varchar2(2),
    numero varchar2(2),
    cantidad number,
    duracion number,
    constraint elementos_asignados_pk primary key (empleado,elemento,actual),
    constraint ele_asignados_empleado_fk foreign key (empleado) references empleados(identificacion),
    constraint ele_asignados_elemento_fk foreign key (elemento) references elementos(codigo)
);
```

```
--elimina la tabla y los constraint asociados a ella, sino se coloca la tabla se elimina pero quedan los constraints
drop table elementos_asignados cascade constraints;
```

```
-- para visualizar los constraints de un usuario y la tabla a la que pertenece puede utilizar la siguiente consulta
select owner,constraint_name,table_name
from user_constraints
```

ADMINISTRACIÓN DE DATOS

NOTA:

Eliminar las tablas y los constraints asociados a ella.

drop table elementos_asignados cascade constraints;

Para visualizar los constraints de un usuario y la tabla a la que pertenece puede utilizar la siguiente consulta

```
select owner,constraint_name,table_name  
from user_constraints;
```