**Ejercicio de repaso de programación en SQL.**

Con el fin de entender todo el proceso que hemos hecho, continuación hare un ejercicio con las soluciones a la mitad, en la cual continuaremos el lunes en clase, para así dar por finalizado la explicación del uso de los diferentes procesos para hacer una base de datos.

**Ejercicio.**

**Universo del discurso.**

Una Tienda de muebles planea expandir su venta en diferentes sucursales del país (Colombia), esto debido al gran éxito que tuvo en la ciudad de Bogotá. Esta tienda busca además hacer que las ventas realizadas generan una eliminación del inventario, esto con el fin de poder ver cual es el inventario real a nivel nacional, con esto podrán generar ventas en cualquier parte del territorio sabiendo cual es el stock y disponibilidad total de la mercancía. Hay que tener en cuenta que todos los productos son únicos y no tienen una repetición en ninguna tienda.

Esta aplicación debe permitir que un vendedor de cualquier tienda del país pueda ver la mercancía existente en tiempo real, además reservar esta compra. Los productos de las tiendas tienen una información específica, nombre del producto, precio de venta al cliente, porcentaje de descuento, código de producto y ubicación de este.

Las tiendas tienen una cantidad indeterminada de vendedores, siempre varía de acuerdo con la demanda de estos. La tienda es conocida por un nombre, una ubicación especifica y un administrador. Los administradores son empleados a cargo de la tienda y los vendedores tienen jefe directo a los administradores. Los empleados en general tienen la información general de una persona y se identifican con la cedula de ciudadanía.

La tienda espera que la aplicación este en la nube y tenga una disponibilidad de 7 de la mañana hasta las 7 de la noche, con un uso mayor en entre las 10 de la mañana ay las 4 de la tarde. Esta debe estar en la nube.

**Notas del universo del discurso.**

Este universo del discurso le falta mucha información de la aplicación, pero entre líneas se encuentra toda la información necesaria para realizar la base de datos que nos corresponde de la aplicación.

Recuerden que con practica van entiendo que información les servirá para poder aplicar en la base de datos.

**Análisis preliminar.**

Se realiza la extracción de la información necesaria del universo del discurso.

Se necesitas tener las tiendas y los productos únicos relacionadas a las tiendas donde están ubicados. Estos productos son únicos y solo se encuentran en una sola tienda, pero los vendedores podrán ver los productos y su ubicación.

Los empleados están en una sola tienda y podrán vender los productos. Los productos tienen valor neto de precio al público y porcentaje máximo de descuento para el vendedor, además del valor del costo de producto en bruto o de realización y ganancia mínima sin impuestos del producto.

**Tablas Para Generar.**

Genéricas.

Departamentos, Ciudades.

Del negocio.

Tienda, Empleados, Productos.

**Diagramas Creados.**

Diagrama Lógico

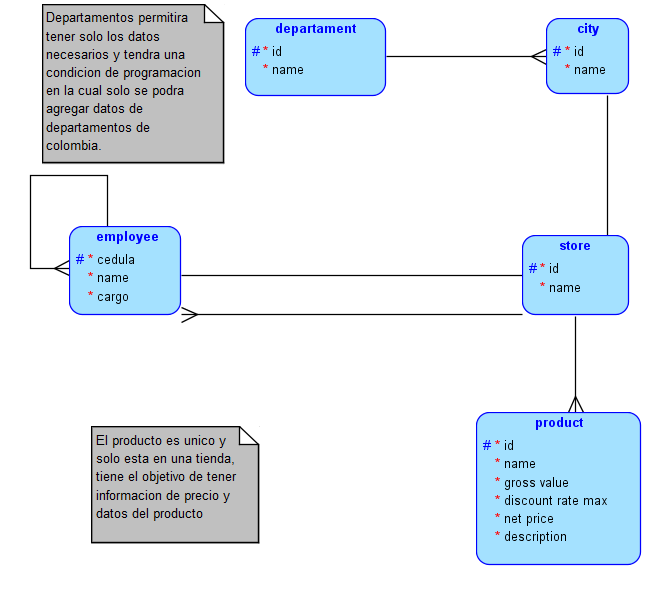


Diagrama físico o Entidad Relación

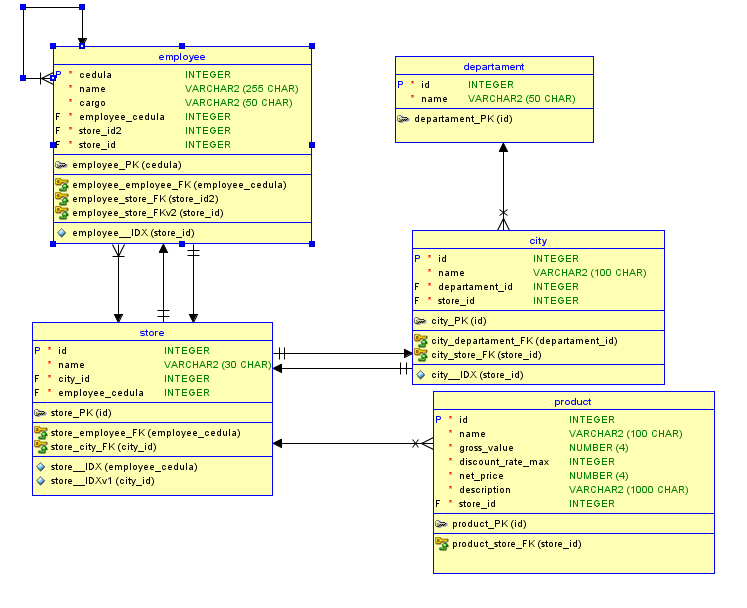


Diagrama Físico y Tablas Física.

**Código Generado.**

**DDL**

-- create tables

create table departments (

    id                             number generated by default on null as identity

                                   constraint departments\_id\_pk primary key,

    name                           varchar2(50 char)

                                   constraint departments\_name\_unq unique not null

)

;

create table citys (

    id                             number generated by default on null as identity

                                   constraint citys\_id\_pk primary key,

    department\_id                  number

                                   constraint citys\_department\_id\_fk

                                   references departments on delete cascade,

    name                           varchar2(100 char)

                                   constraint citys\_name\_unq unique not null

)

;

-- table index

create index citys\_i1 on citys (department\_id);

create table stores (

    id                             number generated by default on null as identity

                                   constraint stores\_id\_pk primary key,

    city\_id                        number

                                   constraint stores\_city\_id\_fk

                                   references citys on delete cascade,

    name                           varchar2(30 char)

                                   constraint stores\_name\_unq unique not null,

    administrator                  number

                                   constraint stores\_administrator\_fk

                                   references employees on delete cascade not null

)

;

-- table index

create index stores\_i1 on stores (administrator);

create index stores\_i2 on stores (city\_id);

create table employees (

    id                             number generated by default on null as identity

                                   constraint employees\_id\_pk primary key,

    store\_id                       number

                                   constraint employees\_store\_id\_fk

                                   references stores on delete cascade,

    name                           varchar2(255 char) not null,

    position                       varchar2(40 char) constraint employees\_position\_cc

                                   check (position in ('SALLER','ADMINISTRATOR')) not null,

    email                          varchar2(200 char) not null,

    hiring\_date                    date not null,

    boss                           number

                                   constraint employees\_boss\_fk

                                   references employees on delete cascade not null

)

;

-- table index

create index employees\_i1 on employees (boss);

create index employees\_i2 on employees (store\_id);

create table products (

    id                             number generated by default on null as identity

                                   constraint products\_id\_pk primary key,

    store\_id                       number

                                   constraint products\_store\_id\_fk

                                   references stores on delete cascade,

    name                           varchar2(50 char)

                                   constraint products\_name\_unq unique not null,

    gross\_value                    number not null,

    discount\_rate\_max              number not null,

    net\_price                      number not null,

    description                    varchar2(1000 char)

)

;

-- table index

create index products\_i1 on products (store\_id);

-- triggers

create or replace trigger employees\_biu

    before insert or update

    on employees

    for each row

begin

    :new.email := lower(:new.email);

end employees\_biu;

/

**DML**

insert into departments (

    id,

    name

) values (

    1,

    'Customer Support'

);

insert into citys (

    id,

    department\_id,

    name

) values (

    1,

    5,

    'Load Balancer Certificate Upgrade'

);

insert into stores (

    id,

    city\_id,

    name,

    administrator

) values (

    1,

    6,

    'Load Balancer Certificate Upgr',

    1

);

insert into employees (

    id,

    store\_id,

    name,

    position,

    email,

    hiring\_date,

    boss

) values (

    1,

    2,

    'Gricelda Luebbers',

    'SALLER',

    'gricelda.luebbers@aaab.com',

    sysdate - 48,

    1

);

insert into products (

    id,

    store\_id,

    name,

    gross\_value,

    discount\_rate\_max,

    net\_price,

    description

) values (

    1,

    6,

    'M-AADG-84',

    11,

    16,

    90,

    'Nec. Donec convallis sollicitudin elementum. Nulla facilisi. In posuere blandit leoeget malesuada. Vivamus efficitur ipsum tellus, quis posuere mi maximus vitae. Quisque tortor odio, feugiat eget sagittisvel, pretium ut metus. Duis et commodo arcu, in vestibulum tellus. In sollicitudin.'

);

**Quick SQL**

departments /select /insert 5

    name vc50 /nn /unique

    citys /select /insert 7

        name vc100 /nn /unique

        stores /select /insert 7

            name vc30 /nn /unique

            administrator /fk employees /nn

            employees /select /insert 10

                name /nn vc255

                position /nn vc40 /check Saller, administrator

                email /nn vc200 /lower

                hiring date date /nn

                boss /nn /fk employees

            products /select /insert 20

                name vc50 /nn /unique

                gross value num /nn

                discount rate max num /nn

                net price num /nn

                description vc1000