



Proyecto Final, Primer Avance

🔗 Git Hub	https://github.com/JHludwolf/SantasWorkshop
👤 Integrantes	 José Luis Lobera del Castillo  Rafael Andrade Rafael Andrade Ruíz Capetillo
☰ Materia	Introducción a las bases de datos
☰ Profesor	José Manuel Velarde Medina

Santa's Workshop



Para ir acorde a las fechas, la temática escogida es la navidad, siendo el tema central los niños al rededor del mundo que, con ilusión envían año tras año su carta al polo norte con la esperanza de recibir un presente el 25 de Diciembre.

Aun que es bien sabido que Santa Claus lleva registro por escrito de todos los niños del planeta, este año ha decidido comenzar a utilizar bases de datos para, con ayuda de `Queries` de SQL, hacer sus cálculos y estadísticas con mayor velocidad. Para esto ha empezado por migrar todos los datos los niños (Nombre, Apellido, Fecha de Nacimiento, Dirección, País y Bondad), de sus hojas de texto `.csv` a tablas de SQL.

También ha decidido registrar a sus elfos, para llevar mejor control de que juguetes fabrican y en que región deben entregar regalos, por lo que también ha creado una tabla para los elfos (Nombre,

Región en la que entrega, Regalos fabricados, Años de servicio y (no podía faltar) su Salario (Siendo las galletas su unidad monetaria).

Los regalos fueron otra de las tablas que Santa detectó necesarias, en ellas lleva registro de a qué niño pertenece el regalo y que elfo lo ha fabricado (Nombre, Precio, Disponibilidad, Edad Mínima, Elfо creador).

Modelo Entidad Relación

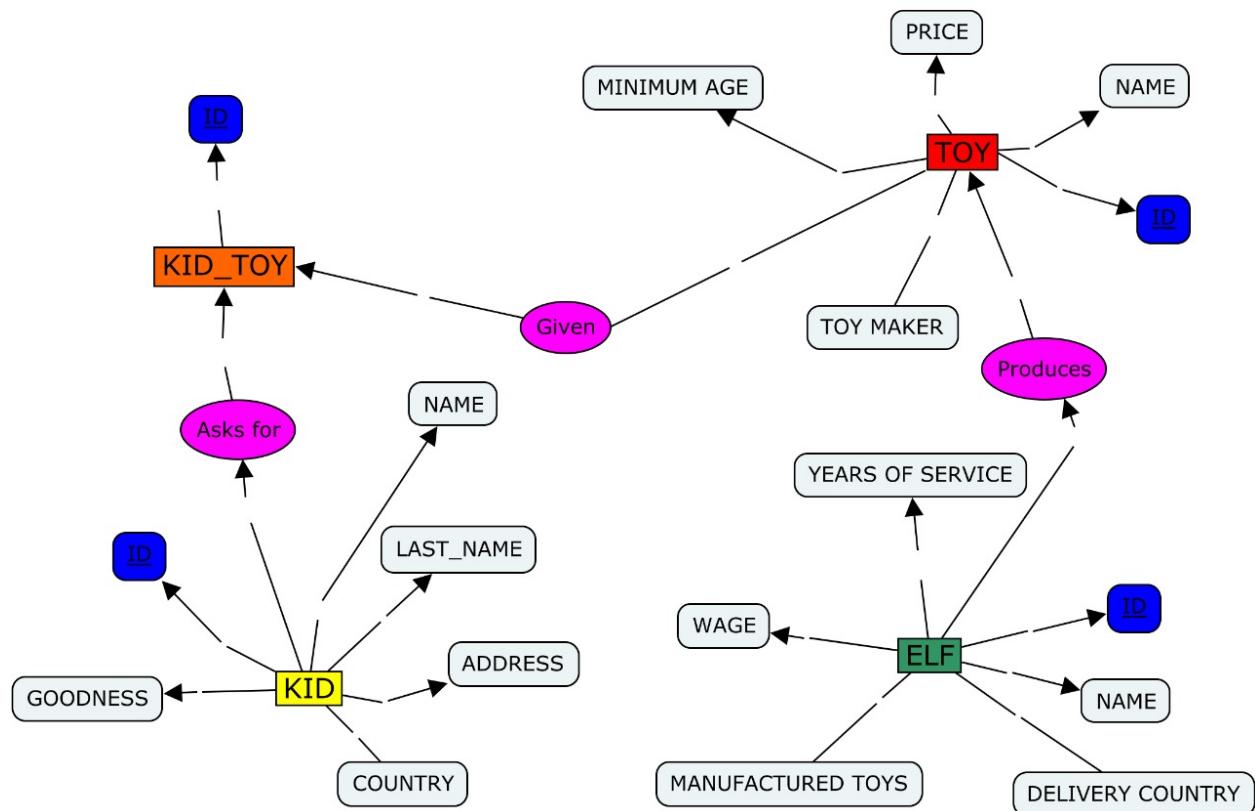
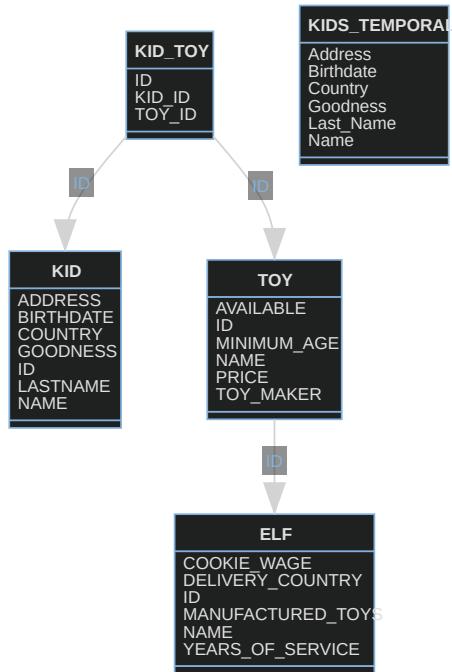


Diagrama de Base de Datos



SQL Table Scripts:

KID:

```

CREATE TABLE KID (
    ID INT IDENTITY(1,1) PRIMARY KEY,
    NAME NVARCHAR(25) NOT NULL,
    LASTNAME NVARCHAR(25) NOT NULL,
    BIRTHDATE DATE NOT NULL,
    ADDRESS NVARCHAR(200) NOT NULL,
    COUNTRY NVARCHAR(50) NOT NULL,
    GOODNESS FLOAT NOT NULL
);
  
```

ELF:

```

CREATE TABLE ELF (
    ID INT IDENTITY(1,1) PRIMARY KEY,
    NAME NVARCHAR(50) NOT NULL,
    DELIVERY_COUNTRY NVARCHAR(15) NOT NULL,
    YEARS_OF_SERVICE INT NOT NULL,
    MANUFACTURED_TOYS INT NOT NULL,
    COOKIE_WAGE INT NOT NULL
);
  
```

TOY:

```
CREATE TABLE TOY (
    ID INT IDENTITY(1,1) PRIMARY KEY,
    NAME NVARCHAR(25) NOT NULL,
    PRICE MONEY NOT NULL,
    MINIMUM_AGE INT NOT NULL,
    AVAILABLE BIT NOT NULL,
    TOY_MAKER INT NOT NULL,
    FOREIGN KEY (TOY_MAKER) REFERENCES ELF(ID)
);
```

KID-TOY:

```
CREATE TABLE KID_TOY (
    ID INT IDENTITY(1,1) PRIMARY KEY,
    KID_ID INT NOT NULL,
    TOY_ID INT NOT NULL,
    FOREIGN KEY (KID_ID) REFERENCES KID(ID),
    FOREIGN KEY (TOY_ID) REFERENCES TOY(ID)
);
```

Justificación del Origen de los Datos

Para apegarnos lo más posible a la realidad decidimos crear nuestros propios datos con ayuda de librerías en Python, con las cuales pudimos generar aleatoriamente Nombres, Apellidos y Direcciones de una región en específico, además de generar fechas aleatorias, en un rango del 2006 al 2019, así como el cálculo de bondad de los niños con ayuda de distribuciones de probabilidad, siendo el porcentaje de niños malos (`goodness < 5`) del 1%, pero la media de bondad es de 8.33. Finalmente con la ayuda de la librería `csv` se creó un documento que recibiría 1 Millón de niños aleatoriamente creados, documento que más tarde sería utilizado para la creación de las tablas en SQL.

```
import datetime
import random
from faker import Faker
from dateutil.relativedelta import relativedelta

fakers = [] # Required country codes

def getRandomDate():
    start_date = datetime.date(2006, 1, 1)
    end_date = datetime.date(2019, 12, 31)

    time_between_dates = end_date - start_date
    days_between_dates = time_between_dates.days
```

```

random_number_of_days = random.randrange(days_between_dates)
birthdate = start_date + datetime.timedelta(days=random_number_of_days)

return birthdate

def getRandomNameLastAddressCountry():
    randomCountryCode = fakers[random.randint(0, len(fakers))-1]
    fake = Faker(randomCountryCode)

    address = fake.address()
    address = address.replace('\n', '. ')
    country = fake.current_country()

    countryLastName = fake.last_name()
    countryName = fake.first_name_male() if random.randint(0,1) is 0 else fake.first_name_female()

    return countryName, countryLastName, address, country

def getRandomGoodness():
    randomNum = random.uniform(0.0, 500.0)
    goodness = randomNum if randomNum <= 10 else random.uniform(6.0, 11)
    if goodness > 10: goodness = 10

    return goodness

def getRandomKid():
    name, lastName, address, country = getRandomNameLastAddressCountry()
    birthdate = getRandomDate()
    goodness = getRandomGoodness()

    return name, lastName, birthdate, address, country, goodness

```

```

from kidGeneratorFunctions import getRandomKid
from kidClass import Kid
import csv

kids = 1000000
with open('kids.csv', 'w', newline='') as csvfile:
    writer = csv.writer(csvfile, delimiter=',', quotechar='', quoting=csv.QUOTE_MINIMAL)

    for _ in range(kids):
        name, lastName, birthdate, address, country, goodness = getRandomKid()
        randomKid = Kid(name, lastName, birthdate, address, country, goodness)
        writer.writerow([name, lastName, birthdate, address, country, goodness])

```

Visualización de Datos:

```

SELECT TOP 30 *
FROM KID;

```

	Name	Last_Name	Birthdate	Address	Country	Goodness
1	Theodora	Stückler	2010-09-12	Luttenbergerweg 0/8. 1991...	Austria	6.216544312407769
2	Inga	Poulsen	2015-03-11	3441 Nørgaard Branch. Pal...	Denmark	8.668031157763059
3	Luka	Kunz	2017-07-14	Bodnerstraße 4/9. 6178 Ma...	Austria	10
4	Ιερεμίας	Παπαστηλιωτοπούλου	2019-06-13	Δημητρητσίου 920,. TK 205...	Greece	10
5	Carla	Pudjiastuti	2018-07-20	Jalan Stasiun Wonokromo N...	Indonesia	7.3391109066491085
6	Tammy	Ford	2007-01-22	PSC 0320, Box 4158. APO A...	United States	10
7	Sandra	Bērziņš	2006-06-18	903 Celminš Lock Suite 09...	Latvia	7.871344016510989
8	Tenis	Roze	2017-10-19	18941 Andrešs Squares Sui...	Latvia	6.719367842519214
9	Andrejka	Petrovič	2009-11-25	Podmilščakova ulica 742. ...	Slovenia	7.854879937261468
10	Mateja	Mavrič	2015-09-20	Ulica Staneta Severja 83...	Slovenia	6.690159344443345
11	Joseph	Perrin	2009-06-04	73, avenue Merle. 17729 S...	France	6.000937378476279
12	Denise	Sharp	2008-02-28	Gašparíkova 50. 839 26 Pa...	Slovakia	10
13	Daniel	Pinto	2010-02-24	boulevard Adam. 09716 Ler...	France	8.210753903903132
14	Hortens...	Dinu	2015-11-21	Bulevardul Oprea. Fagetel...	Romania	8.722514326620267
15	Αρης	Σαραντινός	2016-11-06	Νεοχωρούδας 117-128,. 950...	Greece	6.636238331907827
16	Марфа	Яковлев	2015-10-11	ст. Сергиев Посад, алл. С...	Russia	7.742795970247849
17	Francis...	Sarota	2009-05-21	plac Modrzewiowa 100. 90-...	Poland	6.608777311746613
18	直子	小林	2018-05-29	沖縄県東村山市芝浦12丁目9番18...	Japan	9.350660865554005
19	Tatjana	Volk	2015-01-10	Ulica borcev za severno m...	Slovenia	7.108001567797386
20	ନୀଳ୍ୟ	ତେବେ	2007-07-06	79 ମ୍ଲେ 67 ରୋଡ଼ମରାଧିକା ମୁନିସିପାଲିଟି ରେ...	Thailand	9.773570423847456
21	राम	खड्गी	2007-07-07	माननधर चौर वडा 6 . म्याफ्लुଡ. डो...	Nepal	6.461419111324445
22	Zenon	Schleich	2008-12-08	Löchelweg 66. 01530 Lucke...	Germany	10
23	Joško	Zorko	2013-01-10	Vodna steza 3c. 0857 Bled	Slovenia	7.779021583562146
24	Karen	Adams	2016-12-29	230 Donald Spurs. Fernand...	Egypt	8.830240469901677
25	Eugene	Guerin	2018-05-04	08 Setrick Street K13 Y40...	Republic of ...	7.206470195282137
26	Caius	Voinea	2015-08-21	Strada Lucreția Pușcașu N...	Romania	10
27	Charles	Ross	2019-09-21	Hájová 551. 983 51 Zubroh...	Slovakia	9.078618598682764
28	Angel	Manole	2017-11-16	Intrarea Popa. Alexandria...	Romania	10
29	Emanuel	Nemeš	2012-02-06	Soseaua Anabela Dochioiu ...	Romania	9.83694232670858
30	Jacek	Lemanowicz	2012-09-12	plac Wrocławska 709. 60-7...	Poland	9.026782130739324