

LAURA ONTENIENTE

PRESENTACIÓN

PROYECTO DE TRANSFORMACIÓN DIGITAL

Lectura, limpieza e inserción de datos

A D A L A B



LA IDEA DE PROYECTO

Nuestro cliente, Adalab, está inmerso en un proceso de transformación digital y nos ha pedido crear unos métodos que les sean útiles para coger sus datos, limpiarlos e insertarlos en una base de datos.

BENEFICIOS PARA EL CLIENTE

1

ORDEN

Al crear una base de datos dónde exista una tabla por cada tipo de dato, podrán almacenar toda la información de su empresa.

2

AUTOMATIZACIÓN

Con las funciones que se han creado, podrán añadir nuevos registros a la base de datos sin necesidad de volver a crear todas las operaciones necesarias.

3

CONSULTAS

Con este producto, podrán realizar búsquedas de una forma fácil y cómoda, sin necesidad de procesos técnicos complicados.

OBJETIVOS

1 **CREACIÓN DE UNA
BASE DE DATOS**

2 **LIMPIEZA E INSERCIÓN
DE DATOS SQL**

3 **LIMPIEZA E INSERCIÓN
DE DATOS XML**

4 **LIMPIEZA E INSERCIÓN
DE DATOS TXT**

CREACIÓN DE LA BASE DE DATOS EN MYSQL

BBDD

```
1 CREATE DATABASE proyecto_1_equipo_1;
2
3 USE proyecto_1_equipo_1;
```

TABLA XML

```
26 CREATE TABLE data_xml (
27   index_xml INT NOT NULL AUTO_INCREMENT,
28   `time` VARCHAR(255),
29   age VARCHAR(255),
30   gender VARCHAR(255),
31   index_sql INT NOT NULL,
32   PRIMARY KEY (index_xml),
33   CONSTRAINT fk_tabla_sql_tabla_xml
34   FOREIGN KEY (index_sql)
35   REFERENCES tabla_sql (index_sql) ON DELETE
36
```

TABLA SQL

```
5 CREATE TABLE data_sql (
6   index_sql INT NOT NULL AUTO_INCREMENT,
7   q10_part_1 VARCHAR(255),
8   q10_part_2 VARCHAR(255),
9   q10_part_3 VARCHAR(255),
10  q10_part_4 VARCHAR(255),
11  q10_part_5 VARCHAR(255),
12  q10_part_6 VARCHAR(255),
13  q10_part_7 VARCHAR(255),
14  q10_part_8 VARCHAR(255),
15  q10_part_9 VARCHAR(255),
16  q10_part_10 VARCHAR(255),
17  q10_part_11 VARCHAR(255),
18  q10_part_12 VARCHAR(255),
19  q10_part_13 VARCHAR(255),
20  q10_part_14 VARCHAR(255),
21  q10_part_15 VARCHAR(255),
22  q10_part_16 VARCHAR(255),
23  d482xta VARCHAR(255),
24  PRIMARY KEY (index_sql));
```

TABLA TXT

```
38 CREATE TABLE data_txt (
39   index_txt INT NOT NULL AUTO_INCREMENT,
40   index_sql INT NOT NULL,
41   q3 VARCHAR(255),
42   q4 VARCHAR(255),
43   q5 VARCHAR(255),
44   q6 VARCHAR(255),
45   q7 VARCHAR(255),
46   q8 VARCHAR(255),
47   q9 VARCHAR(255),
48   q11 VARCHAR(255),
49   q12 VARCHAR(255),
50   q13 VARCHAR(255),
51   q14 VARCHAR(255),
52   q15 VARCHAR(255),
53   q16 VARCHAR(255),
54   q17 VARCHAR(255),
55   q20 VARCHAR(255),
56   q21 VARCHAR(255),
57   q22 VARCHAR(255),
58   q23 VARCHAR(255),
59   q24 VARCHAR(255),
60   q25 VARCHAR(255),
61   q26 VARCHAR(255),
62   q32 VARCHAR(255),
63   q33 VARCHAR(255),
64   q34 VARCHAR(255),
65   q35 VARCHAR(255),
66   q41 VARCHAR(255),
67   PRIMARY KEY (index_txt),
68   CONSTRAINT fk_data_sql_data_txt
69   FOREIGN KEY (index_sql)
70   REFERENCES data_sql (index_sql) ON DELETE CASCADE ON UPDATE CASCADE);
```

CREACIÓN DE LA BASE DE DATOS EN PYTHON

BBDD

```
import re
import os
import xml.etree.ElementTree as ET
import mysql.connector
import pandas as pd
```

```
# Esta función nos permite crear la base de datos para el cliente
def crear_bbdd(nombre_bbdd):
    cnx = mysql.connector.connect(user='root', password='AlumnaAdalab',
                                   host='127.0.0.1')

    mycursor = cnx.cursor()

    sql = (f"CREATE DATABASE {nombre_bbdd}")
    try:
        mycursor.execute(sql)
        print(f"Se ha creado la BBDD {nombre_bbdd} correctamente.")

    except mysql.connector.Error as err:
        print(err)
        print("Error Code:", err.errno)
        print("SQLSTATE", err.sqlstate)
        print("Message", err.msg)
    return mycursor.close()
```

```
crear_bbdd("proyecto_1_equipo_1")
```

Se ha creado la BBDD proyecto_1_equipo_1 correctamente.
True

TABLAS

```
# Esta función nos permite crear tablas en la base de datos
def creacion_tablas(query):

    cnx = mysql.connector.connect(user='root', password='AlumnaAdalab',
                                   host='127.0.0.1', database='proyecto_1_equipo_1')

    mycursor = cnx.cursor()

    try:
        mycursor.execute(query)
        cnx.commit()
        print(mycursor)

    except mysql.connector.Error as err:
        print(err)
        print("Error Code:", err.errno)
        print("SQLSTATE", err.sqlstate)
        print("Message", err.msg)

    else:
        mycursor.close()
        cnx.close()
```

```
data_xml = """CREATE TABLE IF NOT EXISTS data_xml(
                index_xml INT,
                `time` VARCHAR(255),
                age VARCHAR(255),
                gender VARCHAR(255),
                index_sql INT,
                PRIMARY KEY(index_xml),
                CONSTRAINT fk_sql_xml
                FOREIGN KEY (index_sql)
                REFERENCES data_sql(index_sql))"""
```

```
creacion_tablas(data_xml)
```

CMySQLCursor: CREATE TABLE IF NOT EXISTS data_xml(
..

INSERCIÓN Y LIMPIEZA DE DATOS SQL EN MYSQL

INSERCIÓN

```
USE proyecto1_team1;

-- MySQL dump 10.13  Distrib 8.0.25, for Linux (x86_64)
--
-- Host: localhost    Database: proyecto1
--
-----
-- Server version      8.0.29-0ubuntu0.21.10.2

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Dumping data for table `tabla_sql`
--

LOCK TABLES `tabla_sql` WRITE;
/*!40000 ALTER TABLE `tabla_sql` DISABLE KEYS */;
INSERT INTO `tabla_sql` VALUES (1,' Kaggle Notebooks','Colab
Notebooks','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERRC
Notebooks','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERRC
Notebooks','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERRC
(4,'ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERRC
Notebooks','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERRC
Notebooks','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERRC
','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERRC
Notebooks','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERRC
Notebooks','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERRC
Notebooks','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERRC
(10,'ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERRC
Notebooks','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERROR','ERRC
```

LIMPIEZA

```
1  USE proyecto_1_equipo_1;
2
3  UPDATE data_sql SET q10_part_1= "NULL" WHERE q10_part_1= "ERROR";
4  UPDATE data_sql SET q10_part_2= "NULL" WHERE q10_part_2= "ERROR";
5  UPDATE data_sql SET q10_part_3= "NULL" WHERE q10_part_3= "ERROR";
6  UPDATE data_sql SET q10_part_4= "NULL" WHERE q10_part_4= "ERROR";
7  UPDATE data_sql SET q10_part_5= "NULL" WHERE q10_part_5= "ERROR";
8  UPDATE data_sql SET q10_part_6= "NULL" WHERE q10_part_6= "ERROR";
9  UPDATE data_sql SET q10_part_7= "NULL" WHERE q10_part_7= "ERROR";
10 UPDATE data_sql SET q10_part_8= "NULL" WHERE q10_part_8= "ERROR";
11 UPDATE data_sql SET q10_part_9= "NULL" WHERE q10_part_9= "ERROR";
12 UPDATE data_sql SET q10_part_10= "NULL" WHERE q10_part_10= "ERROR";
13 UPDATE data_sql SET q10_part_11= "NULL" WHERE q10_part_11= "ERROR";
14 UPDATE data_sql SET q10_part_12= "NULL" WHERE q10_part_12= "ERROR";
15 UPDATE data_sql SET q10_part_13= "NULL" WHERE q10_part_13= "ERROR";
16 UPDATE data_sql SET q10_part_14= "NULL" WHERE q10_part_14= "ERROR";
17 UPDATE data_sql SET q10_part_15= "NULL" WHERE q10_part_15= "ERROR";
18 UPDATE data_sql SET q10_part_16= "NULL" WHERE q10_part_16= "ERROR";
19 UPDATE data_sql SET q10_other= "NULL" WHERE q10_other= "ERROR";
20
```

[illegible]

CREACIÓN DE CLASE Y LECTURA DE DATOS XML EN PYTHON

```
import re
import os
import xml.etree.ElementTree as ET
import mysql.connector
import pandas as pd
```

```
class Datos_xml:
```

```
# Esta clase almacenará la función para leer y limpiar los datos xml.
# Como parámetro recibirá el nombre del archivo.
```

```
    def __init__(self, nombre_archivo):
        self.nombre_archivo = nombre_archivo
```

```
    def lectura_xml(self):
```

```
        tree = ET.parse(self.nombre_archivo)
        root = tree.getroot()
```

```
        self.xml_lista = []
```

```
        for child in root:
```

```
            xml_dict = {}
```

```
            for subchild in child:
```

```
                xml_dict[subchild.tag] = subchild.text
```

```
            self.xml_lista.append(xml_dict) # Es una lista que contiene diccionarios
```

```
        return self.xml_lista
```

LIMPIEZA DE DATOS XML

EN PYTHON

```
def actualización_género(self,xml_lista):  
# Esta función actualiza los datos de la columna género.  
# Como argumento recibe la variable creada con el método lectura_xml.  
  
self.xml_lista = xml_lista  
  
for i in xml_lista:  
  
    if i["gender"] == '0':  
        i["gender"] = "Man"  
    elif i["gender"] == '1':  
        i["gender"] = "Woman"  
    elif i["gender"] == '2':  
        i["gender"] = "Nonbinary"  
    elif i["gender"] == '3':  
        i["gender"] = "Prefer not to say"  
    elif i["gender"] == '4':  
        i["gender"] = "Prefer to self-describe"  
  
return xml_lista
```

```
archivo_xml = Datos_xml ("data_xml.xml")
```

```
lectura_xml = archivo_xml.lectura_xml()
```

```
genero_xml = archivo_xml.actualización_género(lectura_xml)
```

```
archivo_xml_limpio = archivo_xml.limpieza_datos(genero_xml)
```

```
def limpieza_datos(self,xml_lista):
```

```
# Con esta función se preparan los datos para su inserción:
```

```
# Como argumento recibe la variable creada con el método actualización género.
```

```
self.xml_lista = xml_lista
```

```
# Eliminar la primera columna
```

```
try:
```

```
    for i in xml_lista:
```

```
        i.pop("level_0")
```

```
        i["index"] = int(i["index"])
```

```
except: pass
```

```
# Convertir los datos de la columna index en tipo integer
```

```
try:
```

```
    for i in xml_lista:
```

```
        i["index"] = int(i["index"])
```

```
except: pass
```

```
# Aplicar la condición if para evitar problemas con index_sql, se aplica e.
```

```
try:
```

```
    for i in xml_lista:
```

```
        if (xml_lista[0] != "index_sql"):
```

```
            i.setdefault("index_sql", int(i["index"]))
```

```
except: pass
```

```
# Convertir los datos en una lista de tuplas.
```

```
tupla_xml = []
```

```
for i in xml_lista:
```

```
    tupla_xml.append(tuple(i.values()))
```

```
return tupla_xml
```

INSERCIÓN DE DATOS XML EN PYTHON

```
class Insertar_datos_xml:
    # Esta clase creará una instancia para la inserción de los datos limpios de los archivos .xml.
    # Como argumento recibirá el nombre del archivo de xml limpio (archivo_xml_limpio).

    def __init__(self, fichero_xml):

        self.fichero_xml = fichero_xml

    def insercion_datos(self):
        # Con esta función se insertan los datos limpios del fichero xml wn la tabla_xml de la Base de datos.
        # No requiere parámetro.

        cnx = mysql.connector.connect(user='root', password='AlumnaAdalab',
                                       host='127.0.0.1', database= "proyecto_1_equipo_1")
        cursor = cnx.cursor()
        query_xml = "INSERT INTO data_xml(index_xml, `time`, age, gender, index_sql) VALUES (%s,%s,%s,%s,%s);"

        try:
            cursor.execute("SET GLOBAL FOREIGN_KEY_CHECKS=0;") # Se desactiva temporalmente la foreign key
            cursor.executemany (query_xml, self.fichero_xml )
            cnx.commit()
            print(cursor.rowcount, "registros insertados correctamente.")
            cursor.execute("SET GLOBAL FOREIGN_KEY_CHECKS=1;") # Se reactiva la foreign key

        except mysql.connector.Error as err:
            print(err)
            print("Error Code:", err.errno)
            print("SQLSTATE", err.sqlstate)
            print("Message", err.msg)
        cursor.close()
        cnx.close()
```

```
insercion_xml = Insertar_datos_xml(archivo_xml_limpio)
```

```
insercion_xml.insercion_datos()
```

```
25972 registros insertados correctamente.
```

| | index_xml | time | age | gender | index_sql | |
|---|-----------|------|-------|--------|-----------|--|
| ► | 2 | 784 | 50-54 | Man | 1 | |
| | 3 | 924 | 22-24 | Man | 2 | |
| | 4 | 575 | 45-49 | Man | 3 | |
| | 5 | 781 | 45-49 | Man | 4 | |
| | 6 | 1020 | 25-29 | Woman | 5 | |
| | 7 | 141 | 18-21 | Woman | 6 | |
| | 8 | 484 | 30-34 | Man | 7 | |
| | 9 | 1744 | 22-24 | Man | 8 | |
| | 10 | 655 | 30-34 | Man | 9 | |
| | 11 | 1777 | 40-44 | Man | 10 | |
| | 12 | 3081 | 18-21 | Woman | 11 | |
| | 13 | 1922 | 18-21 | Woman | 12 | |
| | 14 | 852 | 45-49 | Man | 13 | |
| | 15 | 838 | 22-24 | Man | 14 | |
| | 16 | 563 | 35-39 | Man | 15 | |
| | 17 | 1315 | 50-54 | Man | 16 | |
| | 18 | 479 | 18-21 | Man | 17 | |
| | 19 | 249 | 22-24 | Man | 18 | |
| | 20 | 650 | 30-34 | Man | 19 | |
| | 21 | 1461 | 70+ | Man | 20 | |
| | 22 | 551 | 25-29 | Woman | 21 | |
| | 23 | 258 | 30-34 | Man | 22 | |
| | 24 | 773 | 35-39 | Man | 23 | |
| | 25 | 630 | 18-21 | Man | 24 | |
| | 26 | 2467 | 25-29 | Woman | 25 | |
| | 27 | 7750 | 45-49 | Man | 26 | |
| | 28 | 607 | 22-24 | Man | 27 | |
| | 29 | 525 | 22-24 | Woman | 28 | |

CREACIÓN DE CLASE Y LECTURA DE DATOS TXT EN PYTHON

CLASE Y LECTURA

```
import re
import os
import xml.etree.ElementTree as ET
import mysql.connector
import pandas as pd
```

```
class Datos_txt:

    def __init__ (self,txt):

        self.txt=txt

#Función para leer el archivo txt

    def abrir_txt(self):

        with open(f'{self.txt}','r') as file:
            file_txt = file.readlines() # Es una

        return file_txt
```

```
def limpieza_txt(self,lista):

    for i in lista:
        lista[lista.index(i)] = lista[lista.index(i)].replace("null","NULL").replace("<","under").replace("\n","").split(";")

    for i in lista:
        for n in i:
            i[i.index(n)] = n.strip()
    contador = 0
    for i in lista:
        i.insert(1,contador)
        contador +=1
    try:
        i[0] = int(i[0])
    except:
        continue

    tupla_txt = [tuple(i) for i in lista]

    tupla_txt.pop(0)

    return tupla_txt
```

```
datos_txt = Datos_txt('data_txt.txt')
```

```
lectura_txt = datos_txt.abrir_txt()
```

```
txt_limpios = datos_txt.limpieza_txt(lectura_txt)
```

LIMPIEZA

INSERCIÓN DE DATOS TXT EN PYTHON

```
class Insertar_datos_txt:
    # Esta clase creará una instancia para la inserción de los datos limpios de los archivos .txt.
    # Como argumento recibirá el nombre del archivo de xml limpio (txt_limpios).

    def __init__(self,txt):

        self.txt = txt

    def insertar_datos_txt(self):
        # Con esta función se insertan los datos limpios del fichero txt en la tabla data_txt de la Base de datos.
        # No requiere parámetro.
        cnx = mysql.connector.connect(user='root', password='AlumnaAdalab',
                                       host='127.0.0.1', database='proyecto_1_equipo_1')

        mycursor = cnx.cursor()

        datos = """ INSERT INTO data_txt(index_txt, index_sql, q3, q4, q5, q6, q7, q8, q9, q11, q12, q13, q14, q15, q16, q17, q20, q21, q22, q23, q24, q25, q26, q27, q28, q29, q30, q31, q32, q33, q34, q35, q36, q37, q38, q39, q40, q41, q42, q43, q44, q45, q46, q47, q48, q49, q50, q51, q52, q53, q54, q55, q56, q57, q58, q59, q60, q61, q62, q63, q64, q65, q66, q67, q68, q69, q70, q71, q72, q73, q74, q75, q76, q77, q78, q79, q80, q81, q82, q83, q84, q85, q86, q87, q88, q89, q90, q91, q92, q93, q94, q95, q96, q97, q98, q99, q100, q101, q102, q103, q104, q105, q106, q107, q108, q109, q110, q111, q112, q113, q114, q115, q116, q117, q118, q119, q120, q121, q122, q123, q124, q125, q126, q127, q128, q129, q130, q131, q132, q133, q134, q135, q136, q137, q138, q139, q140, q141, q142, q143, q144, q145, q146, q147, q148, q149, q150, q151, q152, q153, q154, q155, q156, q157, q158, q159, q160, q161, q162, q163, q164, q165, q166, q167, q168, q169, q170, q171, q172, q173, q174, q175, q176, q177, q178, q179, q180, q181, q182, q183, q184, q185, q186, q187, q188, q189, q190, q191, q192, q193, q194, q195, q196, q197, q198, q199, q200, q201, q202, q203, q204, q205, q206, q207, q208, q209, q210, q211, q212, q213, q214, q215, q216, q217, q218, q219, q220, q221, q222, q223, q224, q225, q226, q227, q228, q229, q230, q231, q232, q233, q234, q235, q236, q237, q238, q239, q240, q241, q242, q243, q244, q245, q246, q247, q248, q249, q250, q251, q252, q253, q254, q255, q256, q257, q258, q259, q260, q261, q262, q263, q264, q265, q266, q267, q268, q269, q270, q271, q272, q273, q274, q275, q276, q277, q278, q279, q280, q281, q282, q283, q284, q285, q286, q287, q288, q289, q290, q291, q292, q293, q294, q295, q296, q297, q298, q299, q300, q301, q302, q303, q304, q305, q306, q307, q308, q309, q310, q311, q312, q313, q314, q315, q316, q317, q318, q319, q320, q321, q322, q323, q324, q325, q326, q327, q328, q329, q330, q331, q332, q333, q334, q335, q336, q337, q338, q339, q340, q341, q342, q343, q344, q345, q346, q347, q348, q349, q350, q351, q352, q353, q354, q355, q356, q357, q358, q359, q360, q361, q362, q363, q364, q365, q366, q367, q368, q369, q370, q371, q372, q373, q374, q375, q376, q377, q378, q379, q380, q381, q382, q383, q384, q385, q386, q387, q388, q389, q390, q391, q392, q393, q394, q395, q396, q397, q398, q399, q400, q401, q402, q403, q404, q405, q406, q407, q408, q409, q410, q411, q412, q413, q414, q415, q416, q417, q418, q419, q420, q421, q422, q423, q424, q425, q426, q427, q428, q429, q430, q431, q432, q433, q434, q435, q436, q437, q438, q439, q440, q441, q442, q443, q444, q445, q446, q447, q448, q449, q450, q451, q452, q453, q454, q455, q456, q457, q458, q459, q460, q461, q462, q463, q464, q465, q466, q467, q468, q469, q470, q471, q472, q473, q474, q475, q476, q477, q478, q479, q480, q481, q482, q483, q484, q485, q486, q487, q488, q489, q490, q491, q492, q493, q494, q495, q496, q497, q498, q499, q500, q501, q502, q503, q504, q505, q506, q507, q508, q509, q510, q511, q512, q513, q514, q515, q516, q517, q518, q519, q520, q521, q522, q523, q524, q525, q526, q527, q528, q529, q530, q531, q532, q533, q534, q535, q536, q537, q538, q539, q540, q541, q542, q543, q544, q545, q546, q547, q548, q549, q550, q551, q552, q553, q554, q555, q556, q557, q558, q559, q560, q561, q562, q563, q564, q565, q566, q567, q568, q569, q570, q571, q572, q573, q574, q575, q576, q577, q578, q579, q580, q581, q582, q583, q584, q585, q586, q587, q588, q589, q590, q591, q592, q593, q594, q595, q596, q597, q598, q599, q600, q601, q602, q603, q604, q605, q606, q607, q608, q609, q610, q611, q612, q613, q614, q615, q616, q617, q618, q619, q620, q621, q622, q623, q624, q625, q626, q627, q628, q629, q630, q631, q632, q633, q634, q635, q636, q637, q638, q639, q640, q641, q642, q643, q644, q645, q646, q647, q648, q649, q650, q651, q652, q653, q654, q655, q656, q657, q658, q659, q660, q661, q662, q663, q664, q665, q666, q667, q668, q669, q670, q671, q672, q673, q674, q675, q676, q677, q678, q679, q680, q681, q682, q683, q684, q685, q686, q687, q688, q689, q690, q691, q692, q693, q694, q695, q696, q697, q698, q699, q700, q701, q702, q703, q704, q705, q706, q707, q708, q709, q710, q711, q712, q713, q714, q715, q716, q717, q718, q719, q720, q721, q722, q723, q724, q725, q726, q727, q728, q729, q730, q731, q732, q733, q734, q735, q736, q737, q738, q739, q740, q741, q742, q743, q744, q745, q746, q747, q748, q749, q750, q751, q752, q753, q754, q755, q756, q757, q758, q759, q760, q761, q762, q763, q764, q765, q766, q767, q768, q769, q770, q771, q772, q773, q774, q775, q776, q777, q778, q779, q780, q781, q782, q783, q784, q785, q786, q787, q788, q789, q790, q791, q792, q793, q794, q795, q796, q797, q798, q799, q800, q801, q802, q803, q804, q805, q806, q807, q808, q809, q810, q811, q812, q813, q814, q815, q816, q817, q818, q819, q820, q821, q822, q823, q824, q825, q826, q827, q828, q829, q830, q831, q832, q833, q834, q835, q836, q837, q838, q839, q840, q841, q842, q843, q844, q845, q846, q847, q848, q849, q850, q851, q852, q853, q854, q855, q856, q857, q858, q859, q860, q861, q862, q863, q864, q865, q866, q867, q868, q869, q870, q871, q872, q873, q874, q875, q876, q877, q878, q879, q880, q881, q882, q883, q884, q885, q886, q887, q888, q889, q890, q891, q892, q893, q894, q895, q896, q897, q898, q899, q900, q901, q902, q903, q904, q905, q906, q907, q908, q909, q910, q911, q912, q913, q914, q915, q916, q917, q918, q919, q920, q921, q922, q923, q924, q925, q926, q927, q928, q929, q930, q931, q932, q933, q934, q935, q936, q937, q938, q939, q940, q941, q942, q943, q944, q945, q946, q947, q948, q949, q950, q951, q952, q953, q954, q955, q956, q957, q958, q959, q960, q961, q962, q963, q964, q965, q966, q967, q968, q969, q970, q971, q972, q973, q974, q975, q976, q977, q978, q979, q980, q981, q982, q983, q984, q985, q986, q987, q988, q989, q990, q991, q992, q993, q994, q995, q996, q997, q998, q999, q1000, q1001, q1002, q1003, q1004, q1005, q1006, q1007, q1008, q1009, q1010, q1011, q1012, q1013, q1014, q1015, q1016, q1017, q1018, q1019, q1020, q1021, q1022, q1023, q1024, q1025, q1026, q1027, q1028, q1029, q1030, q1031, q1032, q1033, q1034, q1035, q1036, q1037, q1038, q1039, q1040, q1041, q1042, q1043, q1044, q1045, q1046, q1047, q1048, q1049, q1050, q1051, q1052, q1053, q1054, q1055, q1056, q1057, q1058, q1059, q1060, q1061, q1062, q1063, q1064, q1065, q1066, q1067, q1068, q1069, q1070, q1071, q1072, q1073, q1074, q1075, q1076, q1077, q1078, q1079, q1080, q1081, q1082, q1083, q1084, q1085, q1086, q1087, q1088, q1089, q1090, q1091, q1092, q1093, q1094, q1095, q1096, q1097, q1098, q1099, q1100, q1101, q1102, q1103, q1104, q1105, q1106, q1107, q1108, q1109, q1110, q1111, q1112, q1113, q1114, q1115, q1116, q1117, q1118, q1119, q1120, q1121, q1122, q1123, q1124, q1125, q1126, q1127, q1128, q1129, q1130, q1131, q1132, q1133, q1134, q1135, q1136, q1137, q1138, q1139, q1140, q1141, q1142, q1143, q1144, q1145, q1146, q1147, q1148, q1149, q1150, q1151, q1152, q1153, q1154, q1155, q1156, q1157, q1158, q1159, q1160, q1161, q1162, q1163, q1164, q1165, q1166, q1167, q1168, q1169, q1170, q1171, q1172, q1173, q1174, q1175, q1176, q1177, q1178, q1179, q1180, q1181, q1182, q1183, q1184, q1185, q1186, q1187, q1188, q1189, q1190, q1191, q1192, q1193, q1194, q1195, q1196, q1197, q1198, q1199, q1200, q1201, q1202, q1203, q1204, q1205, q1206, q1207, q1208, q1209, q1210, q1211, q1212, q1213, q1214, q1215, q1216, q1217, q1218, q1219, q1220, q1221, q1222, q1223, q1224, q1225, q1226, q1227, q1228, q1229, q1230, q1231, q1232, q1233, q1234, q1235, q1236, q1237, q1238, q1239, q1240, q1241, q1242, q1243, q1244, q1245, q1246, q1247, q1248, q1249, q1250, q1251, q1252, q1253, q1254, q1255, q1256, q1257, q1258, q1259, q1260, q1261, q1262, q1263, q1264, q1265, q1266, q1267, q1268, q1269, q1270, q1271, q1272, q1273, q1274, q1275, q1276, q1277, q1278, q1279, q1280, q1281, q1282, q1283, q1284, q1285, q1286, q1287, q1288, q1289, q1290, q1291, q1292, q1293, q1294, q1295, q1296, q1297, q1298, q1299, q1300, q1301, q1302, q1303, q1304, q1305, q1306, q1307, q1308, q1309, q1310, q1311, q1312, q1313, q1314, q1315, q1316, q1317, q1318, q1319, q1320, q1321, q1322, q1323, q1324, q1325, q1326, q1327, q1328, q1329, q1330, q1331, q1332, q1333, q1334, q1335, q1336, q1337, q1338, q1339, q1340, q1341, q1342, q1343, q1344, q1345, q1346, q1347, q1348, q1349, q1350, q1351, q1352, q1353, q1354, q1355, q1356, q1357, q1358, q1359, q1360, q1361, q1362, q1363, q1364, q1365, q1366, q1367, q1368, q1369, q1370, q1371, q1372, q1373, q1374, q1375, q1376, q1377, q1378, q1379, q1380, q1381, q1382, q1383, q1384, q1385, q1386, q1387, q1388, q1389, q1390, q1391, q1392, q1393, q1394, q1395, q1396, q1397, q1398, q1399, q1400, q1401, q1402, q1403, q1404, q1405, q1406, q1407, q1408, q1409, q1410, q1411, q1412, q1413, q1414, q1415, q1416, q1417, q1418, q1419, q1420, q1421, q1422, q1423, q1424, q1425, q1426, q1427, q1428, q1429, q1430, q1431, q1432, q1433, q1434, q1435, q1436, q1437, q1438, q1439, q1440, q1441, q1442, q1443, q1444, q1445, q1446, q1447, q1448, q1449, q1450, q1451, q1452, q1453, q1454, q1455, q1456, q1457, q1458, q1459, q1460, q1461, q1462, q1463, q1464, q1465, q1466, q1467, q1468, q1469, q1470, q1471, q1472, q1473, q1474, q1475, q1476, q1477, q1478, q1479, q1480, q1481, q1482, q1483, q1484, q1485, q1486, q1487, q1488, q1489, q1490, q1491, q1492, q1493, q1494, q1495, q1496, q1497, q1498, q1499, q1500, q1501, q1502, q1503, q1504, q1505, q1506, q1507, q1508, q1509, q1510, q1511, q1512, q1513, q1514, q1515, q1516, q1517, q1518, q1519, q1520, q1521, q1522, q1523, q1524, q1525, q1526, q1527, q1528, q1529, q1530, q1531, q1532, q1533, q1534, q1535, q1536, q1537, q1538, q1539, q1540, q1541, q1542, q1543, q1544, q1545, q1546, q1547, q1548, q1549, q1550, q1551, q1552, q1553, q1554, q1555, q1556, q1557, q1558, q1559, q1560, q1561, q1562, q1563, q1564, q1565, q1566, q1567, q1568, q1569, q1570, q1571, q1572, q1573, q1574, q1575, q1576, q1577, q1578, q1579, q1580, q1581, q1582, q1583, q1584, q1585, q1586, q1587, q1588, q1589, q1590, q1591, q1592, q1593, q1594, q1595, q1596, q1597, q1598, q1599, q1600, q1601, q1602, q1603, q1604, q1605, q1606, q1607, q1608, q1609, q1610, q1611, q1612, q1613, q1614, q1615, q1616, q1617, q1618, q1619, q1620, q1621, q1622, q1623, q1624, q1625, q1626, q1627, q1628, q1629, q1630, q1631, q1632, q1633, q1634, q1635, q1636, q1637, q1638, q1639, q1640, q1641, q1642, q1643, q1644, q1645, q1646, q1647, q1648, q1649, q1650, q1651, q1652, q1653, q1654, q1655, q1656, q1657, q1658, q1659, q1660, q1661, q1662, q1663, q1664, q1665, q1666, q1667, q1668, q1669, q1670, q1671, q1672, q1673, q1674, q1675, q1676, q1677, q1678, q1679, q1680, q1681, q1682, q1683, q1684, q1685, q1686, q1687, q1688, q1689, q1690, q1691, q1692, q1693, q1694, q1695, q1696, q1697, q1698, q1699, q1700, q1701, q1702, q1703, q1704, q1705, q1706, q1707, q1708, q1709, q1710, q1711, q1712, q1713, q1714, q1715, q1716, q1717, q1718, q1719, q1720, q1721, q1722, q1723, q1724, q1725, q1726, q1727, q1728, q1729, q1730, q1731, q1732, q1733, q1734, q1735, q1736, q1737, q1738, q1739, q1740, q1741, q1742, q1743, q1744, q1745, q1746, q1747, q1748, q1749, q1750, q1751, q1752, q1753, q1754, q1755, q1756, q1757, q1758, q1759, q1760, q1761, q1762, q1763, q1764, q1765, q1766, q1767, q1768, q1769, q1770, q1771, q1772, q1773, q1774, q1775, q1776, q1777, q1778, q1779, q1780, q1781, q1782, q1783, q1784, q1785, q1786, q1787, q1788, q1789, q1790, q1791, q1792, q1793, q1794, q1795, q1796, q1797, q1798, q1799, q1800, q1801, q1802, q1803, q1804, q1805, q1806, q1807, q1808, q1809, q1810, q1811, q1812, q1813, q1814, q1815, q1816, q1817, q1818, q1819, q1820, q1821, q1822, q1823, q1824, q1825, q1826, q1827, q1828, q1829, q1830, q1831, q1832, q1833, q1834, q1835, q1836, q1837, q1838, q1839, q1840, q1841, q1842, q1843, q1844, q1845, q1846, q1847, q1848, q1849, q1850, q1851, q1852, q1853, q1854, q1855, q1856, q1857, q1858, q1859, q1860, q1861, q1862, q1863, q1864, q1865, q1866, q1867, q1868, q1869, q1870, q1871, q1872, q1873, q1874, q1875, q1876, q1877, q1878, q1879, q1880, q1881, q1882, q1883, q1884, q1885, q1886, q1887, q1888, q1889, q1890, q1891, q1892, q1893, q1894, q1895, q1896, q1897, q1898, q1899, q1900, q1901, q1902, q1903, q1904, q1905, q1906, q1907, q1908, q1909, q1910, q1911, q1912, q1913, q1914, q1915, q1916, q1917, q1918, q1919, q1920, q1921, q1922, q1923, q1924, q1925, q1926, q1927, q1928, q1929, q1930, q1931, q1932, q1933, q1934, q1935, q1936, q1937, q1938, q1939, q1940, q1941, q1942, q1943, q1944, q1945, q1946, q1947, q1948, q1949, q1950, q1951, q1952, q1953, q1954, q1955, q1956, q1957, q1958, q1959, q1960, q1961, q1962, q1963, q1964, q1965, q1966, q1967, q1968, q1969, q1970, q1971, q1972, q1973, q1974, q1975, q1976, q1977, q1978, q1979, q1980, q1981, q1982, q1983, q1984, q1985, q1986, q1987, q1988, q1989, q1990, q1991, q1992, q1993, q1994, q1995, q1996, q1997, q1998, q1999, q2000, q2001, q2002, q2003, q2004, q2005, q2006, q2007, q2008, q2009, q2010, q2011, q2012, q2013, q2014, q2015, q2016, q2017, q2018, q2019, q2020, q2021, q2022, q2023, q2024, q2025, q2026, q2027, q2028, q2029, q2030, q2031, q2032, q2033, q2034, q2035, q2036, q2037, q2038, q2039, q2040, q2041, q2042, q2043, q2044, q2045, q2046, q2047, q2048, q2049, q2050, q2051, q2052, q2053, q2054, q2055, q2056, q2057, q2058, q2059, q2060, q2061, q2062, q2063, q2064, q2065, q2066, q2067, q2068, q2069, q2070, q2071, q2072, q2073, q2074, q2075, q2076, q2077, q2078, q2079, q2080, q2081, q2082, q2083, q2084, q2085, q2086, q2087, q2088, q2089, q2090, q2091, q2092, q2093, q2094, q2095, q2096, q2097, q2098, q2099, q2100, q2101, q2102, q2103, q2104, q2105, q2106, q2107, q2108, q2109, q2110, q2111, q2112, q2113, q2114, q2115, q2116, q2117, q2118, q2119, q2120, q2121, q2122, q2123, q2124, q2125, q2126, q2127, q2128, q2129, q2130, q2131, q2132, q2133, q2134, q2135, q2136, q2137, q2138, q2139, q2140, q2141, q2142, q2143, q2144, q2145, q2146, q2147, q2148, q2149, q2150, q2151, q2152, q2153, q2154, q2155, q2156, q2157, q2158, q2159, q2160, q2161, q2162, q2163, q2164, q2165, q2166, q2167, q2168, q2169, q2170, q2171, q2172, q2173, q2174, q2175, q2176, q2177, q2178, q2179, q2180, q2181, q2182, q2183, q2184, q2185, q2186, q2187, q2188, q2189, q2190, q2191, q2192, q2193, q2194, q2195, q2196, q2197, q2198, q2199, q2200, q2201, q2202, q2203, q2204, q2205, q2206, q2207, q2208, q2209, q22
```


| | index_txt | index_sql | q3 | q4 | q5 | q6 | q7 | q8 | q9 |
|---|-----------|-----------|--------------|--|---------------------------|---------------|------------------------------|--------|-------------------|
| ▶ | 1 | 1 | Indonesia | Master's degree | Program/Project Manager | 20+ years | NULL, SQL, C, C++, Java | Python | NULL, Notepad- |
| | 2 | 2 | Pakistan | Master's degree | Software Engineer | 1-3 years | Python, C++, Java | Python | NULL, PyCharm |
| | 3 | 3 | Mexico | Doctoral degree | Research Scientist | 20+ years | Python | Python | NULL, Spyder , |
| | 4 | 4 | India | Doctoral degree | Other | under 1 years | Python, C, MATLAB | Python | NULL, Spyder |
| | 5 | 5 | India | I prefer not to answer | Currently not employed | under 1 years | Python | Python | Jupyter (JupyterL |
| | 6 | 6 | India | Some college/university study without earning a... | Student | 1-3 years | NULL, C++, Java, Javascript | Python | NULL, Visual Stu |
| | 7 | 7 | India | Bachelor's degree | Data Scientist | 5-10 years | Python | Python | NULL, Jupyter N |
| | 8 | 8 | Russia | Bachelor's degree | Currently not employed | 3-5 years | Python, SQL | Python | NULL, Other |
| | 9 | 9 | Turkey | I prefer not to answer | Other | 1-3 years | Python, SQL | SQL | NULL, Spyder , |
| | 10 | 10 | Australia | Doctoral degree | Other | 1-3 years | Python, R, SQL | R | NULL, RStudio , |
| | 11 | 11 | India | Master's degree | Student | under 1 years | Python, R, C++ | R | NULL, RStudio , |
| | 12 | 12 | India | Master's degree | Student | under 1 years | Python, MATLAB | Python | NULL, RStudio , |
| | 13 | 13 | Nigeria | Master's degree | Program/Project Manager | 5-10 years | Python, SQL | Python | NULL, Spyder , |
| | 14 | 14 | Nigeria | Bachelor's degree | Other | under 1 years | Python | Python | NULL, Visual Stu |
| | 15 | 15 | Greece | Doctoral degree | Research Scientist | 10-20 years | Python, C, C++, MATLAB | Python | NULL, Spyder |
| | 16 | 16 | Belgium | Bachelor's degree | Data Analyst | 20+ years | Python, SQL | Python | Jupyter (JupyterL |
| | 17 | 17 | Pakistan | Bachelor's degree | Data Scientist | 1-3 years | Python, SQL | Python | Jupyter (JupyterL |
| | 18 | 18 | Japan | Master's degree | Software Engineer | 3-5 years | Python, SQL, C, Java, Jav... | Python | Jupyter (JupyterL |
| | 19 | 19 | Egypt | Bachelor's degree | Other | under 1 years | NULL, None | R | NULL, PyCharm |
| | 20 | 20 | Singapore | Bachelor's degree | Other | under 1 years | Python | Python | Jupyter (JupyterL |
| | 21 | 21 | Turkey | Bachelor's degree | Data Scientist | 3-5 years | Python, R, SQL, C++ | R | Jupyter (JupyterL |
| | 22 | 22 | Indonesia | Master's degree | Student | 1-3 years | NULL, R | R | NULL, RStudio |
| | 23 | 23 | Brazil | Master's degree | Machine Learning Engin... | 20+ years | Python, SQL, C++ | SQL | NULL, PyCharm |
| | 24 | 24 | India | Bachelor's degree | Student | 1-3 years | Python, R, SQL, C, C++, M... | Python | NULL, RStudio , |
| | 25 | 25 | Poland | Master's degree | Machine Learning Engin... | 3-5 years | Python, C++ | Python | NULL, Visual Stu |
| | 26 | 26 | Brazil | Doctoral degree | Research Scientist | under 1 years | Python, R | Python | NULL, RStudio , |
| | 27 | 27 | China | Master's degree | Student | 3-5 years | Python, C, C++ | Python | NULL, PyCharm |
| | 28 | 28 | Iran, Isl... | Bachelor's degree | Data Scientist | 3-5 years | Python | Python | NULL, Visual Stu |
| | 29 | 29 | India | Bachelor's degree | Student | under 1 years | Python, Java | Python | NULL, PyCharm |
| | 30 | 30 | India | Bachelor's degree | Other | 1-3 years | Python | Python | NULL, Visual Stu |
| | 31 | 31 | India | Master's degree | Data Scientist | 5-10 years | Python, R, SQL | Python | NULL, Visual Stu |
| | 32 | 32 | United... | Some college/university study without earning a... | Data Analyst | under 1 years | NULL, R | R | NULL, RStudio |
| | 33 | 33 | Egypt | Bachelor's degree | Data Analyst | 3-5 years | Python | R | NULL, Notepad- |



**¡PASAMOS A LAS CONSULTAS DE EJEMPLO
PARA COMPROBAR NUESTRO CÓDIGO!**

¿DUDAS?

