**CASE STUDY: Data Warehouse for Cancer treatment facility**

**Introduction**

Health care organizations are now receiving important requests from their customers, government and regulatory bodies that advanced clinical information systems and other related automation must be implemented to achieve the necessary improvement in the efficiency and quality of patient care.

There has never been a more challenging time in healthcare to address the issues of improving the quality using information, knowledge and technology to address the high numbers of adverse events occurring in a cancer treatment center. Health care should be supported by systems that are carefully and consciously designed to produce care that is safe, effective, patient-centered, timely, efficient, and equitable. This environment involves reducing medical errors, focusing on the treatment of chronic illness, and reducing fragmentation in the health care delivery system through the automation of the patient record.

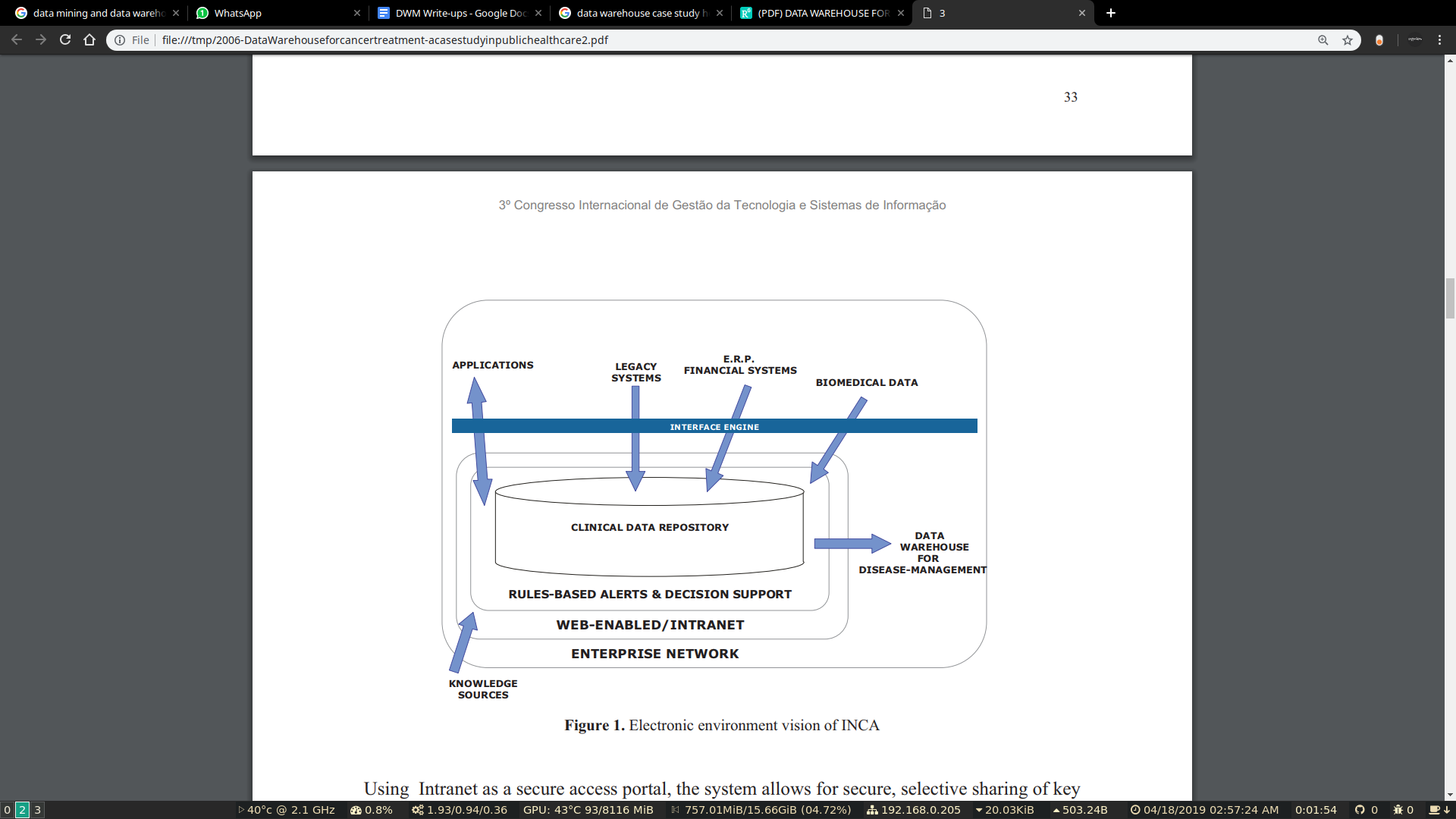
It is clear that advanced clinical information systems will be a driver for quality improvements. Effectively managing data and then translating the data into information will result in substantial benefits. The ability to integrate data to have valuable information will result in a competitive advantage, enabling health care organizations to operate more efficiently.

The objective of this article is to show the contribution of a disease data warehouse in solving health care management problems. The management and integration of key data can provide even more valuable information. The ability to research various conditions, evaluating medical procedures and protocols to define statistically optimal outcomes is an incredibly powerful tool to improve the efficiency and quality in a cancer treatment center.

The subject of this case study is to show a information system which is capable of managing thousands of patients. The objective is to ensure a solution that meets physician’s needs but also allows them to leverage extremely valuable information resulting from combining medical and managerial information.

INCA is an agency under the direct administration of the ministry of health, associated with the health care secretariat. INCA has five specialized hospital units and is a large group practice with over 650 staff and physicians-in-training plus an allied health staff of nearly 3,400. The practice sees over 50.000 outpatients per year and has approximately 350 inpatient beds with over 13.000 hospital admissions per year.

The project to develop and deploy the computer-based patient record (CPR) components took approximately 3 years. Over a period of several years, the process of documentation has evolved from handwritten, typed notes and reports to the electronic capture of information. Nowadays the electronic environment vision of INCA is to significantly contribute to the success, practice, education, and research through excellence in information management, systems, processes, and technology as shown in Figure.



**Methodology**

From a quality-management point of view, the systems that support disease management represent a significant step forward in measuring and improving longitudinal care. An understanding of how a patient carries out cancer care from month to month or year to year, is not easily obtained today. A well designed disease management tool can impact in different areas such as problem identification, information sharing and logistical support .

Proactive problem identification involves designing alerts and reminders for patients who are at-risk or high-risk state. From a clinical point of view, it should identify the characteristics of members who demand special attention.

The power of changing behavior by sharing information about longitudinal patient history demonstrates tangible benefits. The ability of healthcare entities to integrate data will result in a competitive advantage, enabling them to operate more efficiently.

Logistical support is essential to ensure that the patient with chronic illness like cancer receives an efficient, clinically correct attention by physicians in charge of patient treatment.

**Results**

This software was developed and implemented as a decision support system that became one of the main tools for management of the attendance of patients at INCA. It is being used by directors of the hospitals and its respective command of clinics.

The system identifies the patients who already have been registered and are in roll-out. In this phase the patient is submitted to a series of examinations of pathological anatomy, clinical pathology, radiology, among others, to detect the localization and evolution of the illness.

After these examinations, the localization, stage and diagnosis of the tumor can be identified. Only then can treatment of cancer that corresponds, predominantly, to a surgery, applications of radiotherapy and chemotherapy initiate.

The information in the Disease Management System is available from the more aggregated level to the more specific level. The physician can start the analysis from the size of a patient’s waiting queue and can go as specific as an exam of a particular patient.

The first available information in the system is the number of patients who are waiting for the beginning of treatment. This information is available for each hospital unit at INCA. It is segmented by 3 hospitals HC1, HC2 and HC3.

All the data presented is fictitious to guarantee the privacy and security of the information on the system.

The manager can select, according to his security profile access, a unit to be analyzed by drilling down to the level of segmentation that shows the size of the queue in each specialty.

The next step is to select a clinic to be analyzed. The manager can evaluate a queue spread throughout the previous year.

From the presented historical series, the manager can select a month of interest and identify patients distributed according to each treatment.

The manager can drill down to a list of patients by selecting a specific treatment.

From a list of patients, the manager can pick a particular patient and analyse his clinical history. By selecting a Electronic Medical Record (EMR), the manager can choose a specific exam for this particular patient.

The patients’ data is aggregated in a clinical data repository to create a patient’s treatment flow. This electronic flow joins medical records and the sequence of events of a patient in only one screen . This approach is totally new allowing doctors to examine clinical evolution of a patient quickly by using past events, current situation and programmed examinations.

**Conclusions**

The creation and implementation of this integrated queue manager application has changed the entire workflow process at INCA. Not only is patient management becoming more organized, but the utilization of resources is improving as well. Despite resistance to change, physicians feel that INCA will prove to be a long-term success. Currently, IT analysts are working toward the integration with other departments. As time progresses, the project measurements continue to show positive results for both the facility and patient.

Future trends will likely include a greater emphasis on disease management and protocols as well as more sophisticated rules and logic as additional types of clinical data become available.

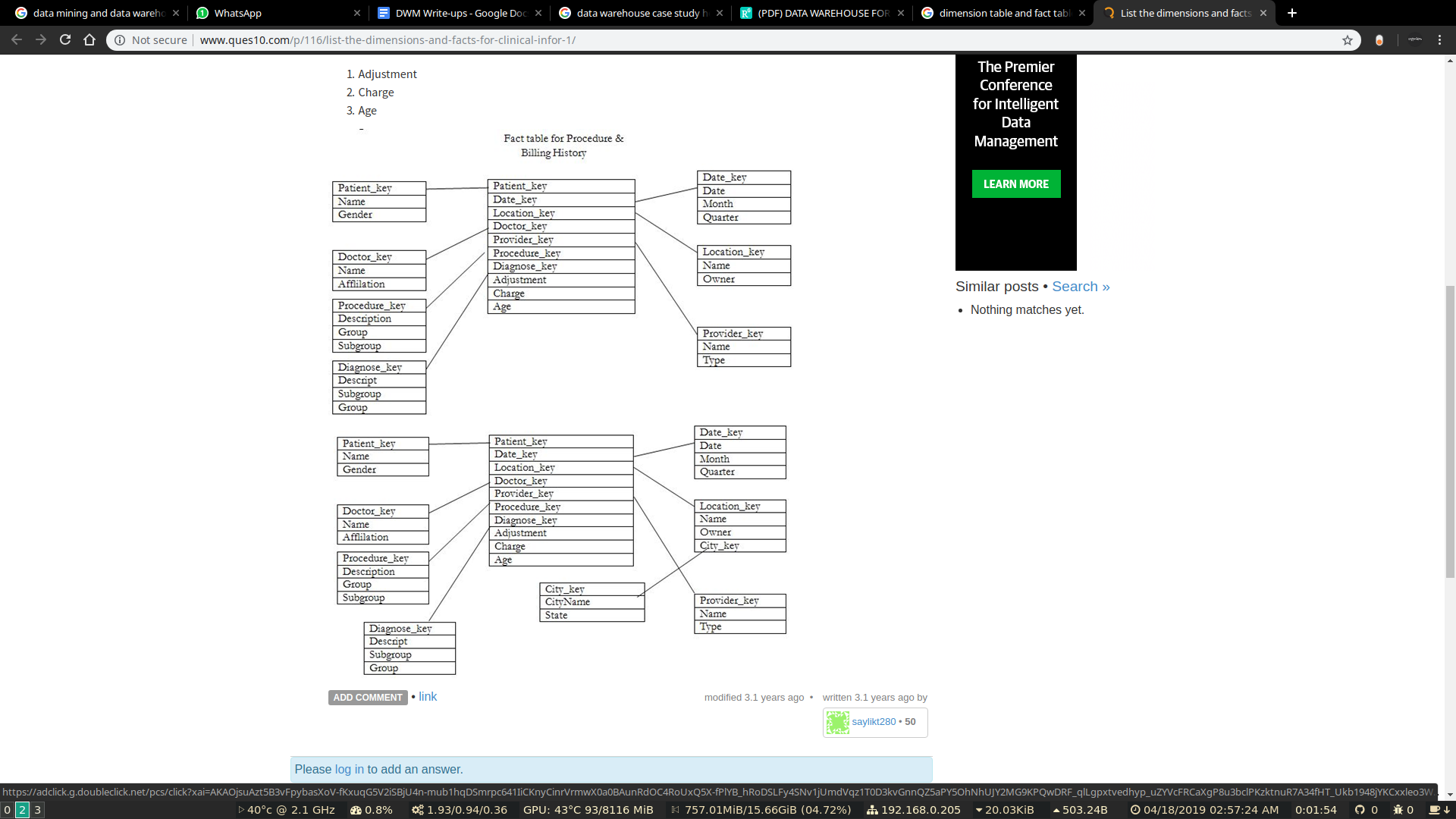
**Experiment 1B : Dimensional modeling of Case Study**

Dimensions:

1. Patient
2. Doctor
3. Procedure
4. Diagnose
5. Date of Service
6. Location
7. Provider

Fact:

1. Adjustment
2. Charge
3. Age

****

Schema used: snowflake schema

**Experiment 2**

**Aim**: To implement Extract, Transform and Load (ETL) process

**Theory:**

# ETL

ETL is an abbreviation of Extract, Transform and Load. In this process, an ETL tool extracts the data from different RDBMS source systems then transforms the data like applying calculations, concatenations, etc. and then load the data into the Data Warehouse system.

It's tempting to think a creating a Data warehouse is simply extracting data from multiple sources and loading into database of a Data warehouse. This is far from the truth and requires a complex ETL process. The ETL process requires active inputs from various stakeholders including developers, analysts, testers, top executives and is technically challenging.

In order to maintain its value as a tool for decision-makers, Data warehouse system needs to change with business changes. ETL is a recurring activity (daily, weekly, monthly) of a Data warehouse system and needs to be agile, automated, and well documented.

Need for ETL

There are many reasons for adopting ETL in the organization:

* It helps companies to analyze their business data for taking critical business decisions.
* Transactional databases cannot answer complex business questions that can be answered by ETL.
* A Data Warehouse provides a common data repository
* ETL provides a method of moving the data from various sources into a data warehouse.
* As data sources change, the Data Warehouse will automatically update.
* Well-designed and documented ETL system is almost essential to the success of a Data Warehouse project.
* Allow verification of data transformation, aggregation and calculations rules.
* ETL process allows sample data comparison between the source and the target system.
* ETL process can perform complex transformations and requires the extra area to store the data.
* ETL helps to Migrate data into a Data Warehouse. Convert to the various formats and types to adhere to one consistent system.
* ETL is a predefined process for accessing and manipulating source data into the target database.
* ETL offers deep historical context for the business.
* It helps to improve productivity because it codifies and reuses without a need for technical skills.

**Conclusion:**

Thus, we have studied and implemented Extract, Transform and Load (ETL) process.

**CODE:**

from prettytable import PrettyTable

from mapping import \*

import datetime

from datetime import \*

def read\_file(filename = "data.csv"):

data = None

with open(filename,'r') as f:

data = f.readlines()

if data is None:

print "[Error::FileHandling], File is empty"

return

data = [each.rstrip("\r\n") for each in data]

data\_lists = [each.split(",") for each in data]

return data, data\_lists

def write\_file(data\_lists, name):

f = open(name, "w")

for each in data\_lists:

f.write(''.join([str(str(a) + ',') for a in each]))

f.write("\n")

f.close()

def print\_table(data\_lists):

print "Original Data"

t = PrettyTable(name\_columns)

for each in data\_lists:

t.add\_row(each)

print t

print "\n"\*2

def calculate\_age(d):

born = datetime.strptime(d,"%Y-%m-%d")

today = date.today()

age = today.year - born.year - ((today.month, today.day) < (born.month, born.day))

return age

def transform\_data\_age(data\_lists):

print "TRANSFORMING ACCORDING TO AGE"

name\_cols2 = []

name\_cols2[:] = name\_columns

name\_cols2[1] = "Age"

t = PrettyTable(name\_cols2)

for i in range(len(data\_lists)):

data\_lists[i][1] = calculate\_age(data\_lists[i][1])

t.add\_row(data\_lists[i])

print t

write\_file(data\_lists, "age.csv")

def remove\_duplicates(data\_lists):

print "REMOVING DUPLICATES FROM THE DATA"

new\_data = []

new\_data\_keys = []

dont\_print = []

t = PrettyTable(name\_columns)

for i, each in enumerate(data\_lists):

if each[0] not in new\_data\_keys:

new\_data\_keys.append(each[0])

# t.add\_row(each)

else:

k = new\_data\_keys.index(each[0])

data\_lists[k][-1] += ", " + each[-1]

dont\_print.append(i)

for i in range(len(data\_lists)):

if i not in dont\_print:

t.add\_row(data\_lists[i])

print t

write\_file(data\_lists, "duplicatesRemoved.csv")

def expand\_abbreviations(data\_lists):

t = PrettyTable(name\_columns)

for i in range(len(data\_lists)):

for j in range(len(data\_lists[i])):

e = data\_lists[i][j]

if mapping.has\_key(e):

data\_lists[i][j] = mapping[e]

t.add\_row(data\_lists[i])

print "EXPANDING DATASET"

print t

write\_file(data\_lists, "abbreviations.csv")

def main():

raw\_data, data\_lists = read\_file()

print\_table(data\_lists)

transform\_data\_age(data\_lists)

remove\_duplicates(data\_lists)

expand\_abbreviations(data\_lists)

if \_\_name\_\_ == '\_\_main\_\_':

main()

mapping = {

"BT" : "Brain Tumor",

"GC" : "Gastric Cancer",

"RC": "Renal Cancer",

"TMMC": "Twin Mountains Medical Center",

"CH": "Crossroads Hospital",

"SLCH" : "Silver Lake Community Hospital",

}

num\_columns = 5;

name\_columns = [

"Name",

"Date of Birth",

"Cancer Type"

]

**DATA:**

------------------- data.csv file -------------------------

kushal,1965-06-01,BT

Raj,1965-06-11,GC

Abhishek,1960-10-03,GC

Sanket,1955-09-08,RC

kushal,1965-06-01,BT

Jeremy,1950-08-21,BT

Sanket,1965-09-08,RC

------------------ age.csv file --------------------------

kushal,51,1995-06-01,BT

Raj,51,1995-06-11,GC

Abhishek,56,1990-10-03,GC

Sanket,61,1985-09-08,RC

kushal,51,1995-06-01,BT

Jeremy,66,1980-08-21,BT

Sanket,61,1985-09-08,RC

---------------------- abbreviations.csv file --------------------------------

kushal,51,Brain Tumor

Raj,51,Gastric Cancer

Abhishek,56,Gastric Cancer

Sanket,61,Renal Cancer

kushal,51,Brain Tumor

Jeremy,66,Brain Tumor

Sanket,61,Renal Cancer

--------------------------- remove and concat duplicate -------------------------------------

kushal,51,Brain Tumor

Raj,51,Gastric Cancer

Abhishek,56,Gastric Cancer

Sanket,61,Renal Cancer

kushal,51,Brain Tumor

Jeremy,66,Brain Tumor

Sanket,61,Renal Cancer

**OUTPUT**:

$ python etl1.py

Original Data

+----------+---------------+--------------+

| Name | Date of Birth | Cancer Type |

+----------+---------------+--------------+

| kushal | 1995-06-01 | BT |

| Raj | 1995-06-11 | GC |

| Abhishek | 1990-10-03 | GC |

| Sanket | 1985-09-08 | RC |

| kushal | 1995-06-01 | BT |

| Jeremy | 1980-08-21 | BT |

| Sanket | 1985-09-08 | RC |

+----------+---------------+--------------+

TRANSFORMING ACCORDING TO AGE

+----------+---------------+----------------+

| Name | Age | Cancer Type |

+----------+---------------+----------------+

| kushal | 51 | BT |

| Raj | 51 | GC |

| Abhishek | 56 | GC |

| Sanket | 61 | RC |

| kushal | 51 | BT |

| Jeremy | 66 | BT |

| Sanket | 61 | RC |

+----------+---------------+----------------+

REMOVING DUPLICATES FROM THE DATA

+----------+---------------+----------------+

| Name | Age | Cancer Type |

+----------+---------------+----------------+

| kushal | 51 | BT |

| Raj | 51 | GC |

| Abhishek | 56 | GC |

| Sanket | 61 | RC |

| Jeremy | 66 | BT |

+----------+---------------+----------------+

EXPANDING DATASET

+----------+-----------+--------------------+

| Name | Age | Cancer Type |

+----------+-----------+--------------------+

| kushal | 51 | Brain Tumor |

| Raj | 51 | Gastric Cancer |

| Abhishek | 56 | Gastric Cancer |

| Sanket | 61 | Renal Cancer |

| Jeremy | 66 | Brain Tumor |

+----------+-----------+--------------------+