Data Mining- Assignment 3

Question:1

1- Find names of customers who have purchased tours that cost less than €500 - **simple query of data retrieval**

2- List the names of the customers, the number of tour packages that the customers have purchased, and the total cost for the tours- **simple query of data retrieval**

3- Calculate the difference in quarterly sales of tours between this year and the previous two years- **Online Analytical Processing**

4- Find a rule such as "IF customers purchase a tour package to France, THEN it is 80% likely
that the same customers also purchase a tour package to Spain- **Data Mining.**

5- From the customer purchase history, build a model for predicting the kinds of customer who are likely to purchase tours to a certain country- **Data Mining**.

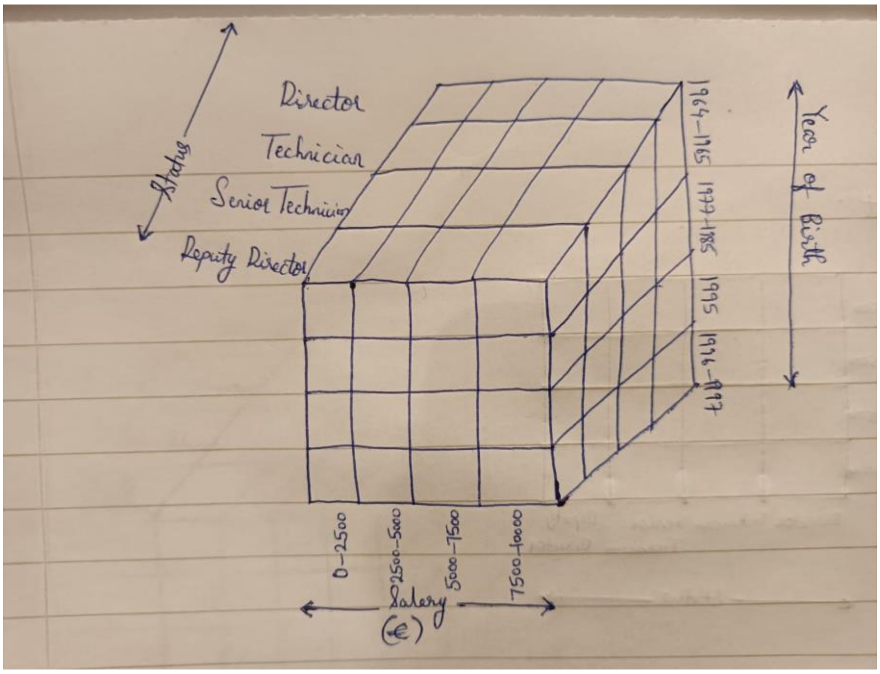
Question:2

1) We can discretize the 'salary' into three pay bands by dividing it into equal width. This approach divides the range of values into a user defined number of intervals in a way that each interval has equal width. The salary data is as follows:
Salary : 10000, 1800, 4014.28(Missing value), 1600, 1700, 1700, 3300, 8000
Band1(Bin1) : 1600, 1700, 1700, 1800
Band2(Bin2): 3300,4014.28
Band3(Bin3):8000,10000

2) In this dataset, Miss Davis's salary is unknown and we have to fill the missing value. This can be done by replacing the missing value with the mean of the salary column. In this process, we calculate the mean of the non-missing values in a column and then replace the missing value with the computed mean. In this case, the salary of Miss Davis will be €4014.28.

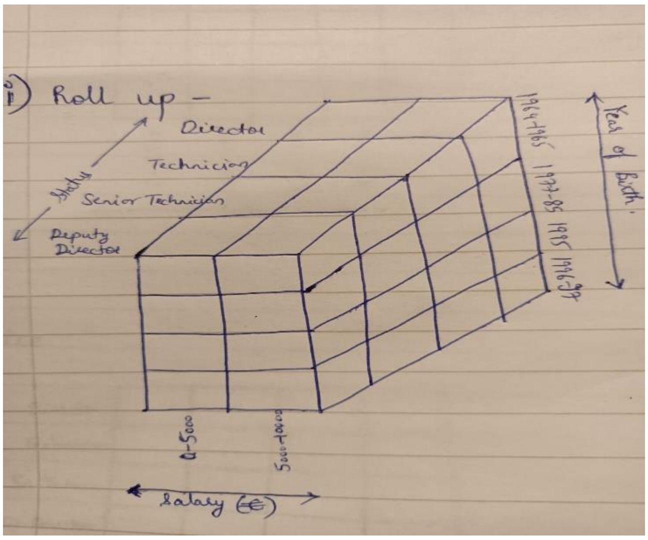
3) Out of the employee's record, the record of person named 'Smith' with salary €10000 can be considered as an outlier. Basically, an outlier affects the mean and standard deviation of a data distribution thus makes it difficult to analyse the data correctly. The data distribution can be skewed in one direction making it difficult to draw accurate and useful insights from the data.

4) 3D Diagram

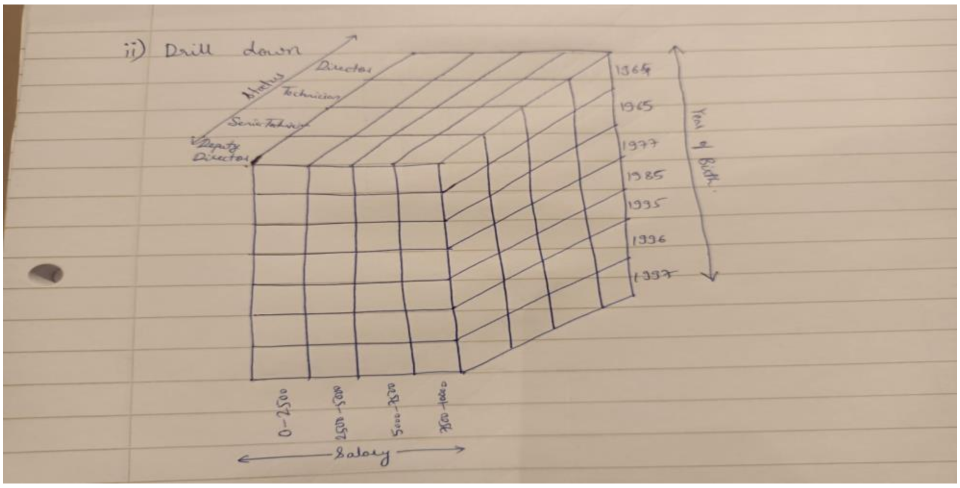


5) The data points inside the cube represent the salary of the respective status.

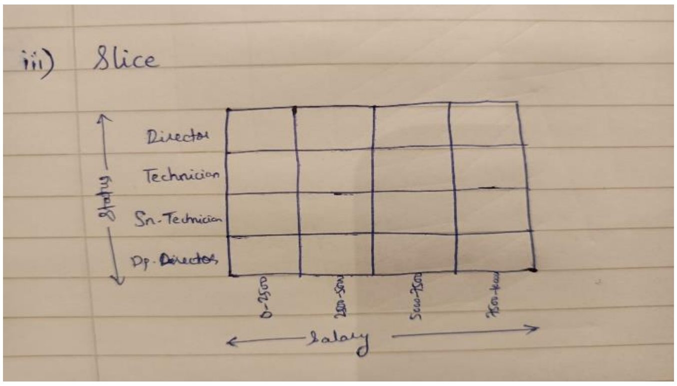
i) Roll up – The rollup operation is used to reduce the dimensions; it is also called as aggregation. In this process at least one or more dimensions need to be removed. In this example, the salary is rolled up to reduce the dimensionality.



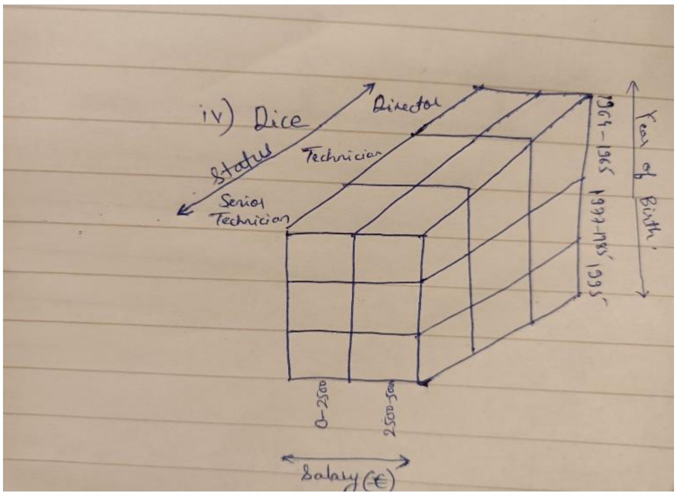
ii) Drill down – In drill down operation the data is fragmented into smaller parts. It is the exact opposite of the roll up operation. Here we are increasing the dimension. In this example, the year of birth dimension is fragmented into small intervals.



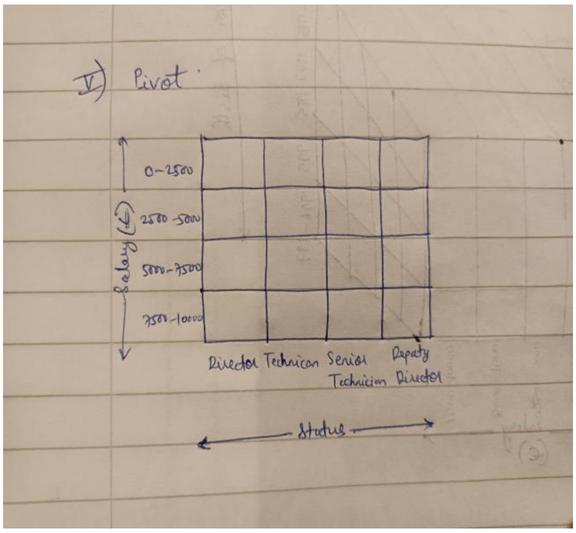
iii) Slice – In slice, we create a new sub cube based on one particular dimension. Here we have sliced the first dimension of ‘year of birth’.



iv) Dice – It is similar to slice just here we select two or more dimensions to form a sub cube.

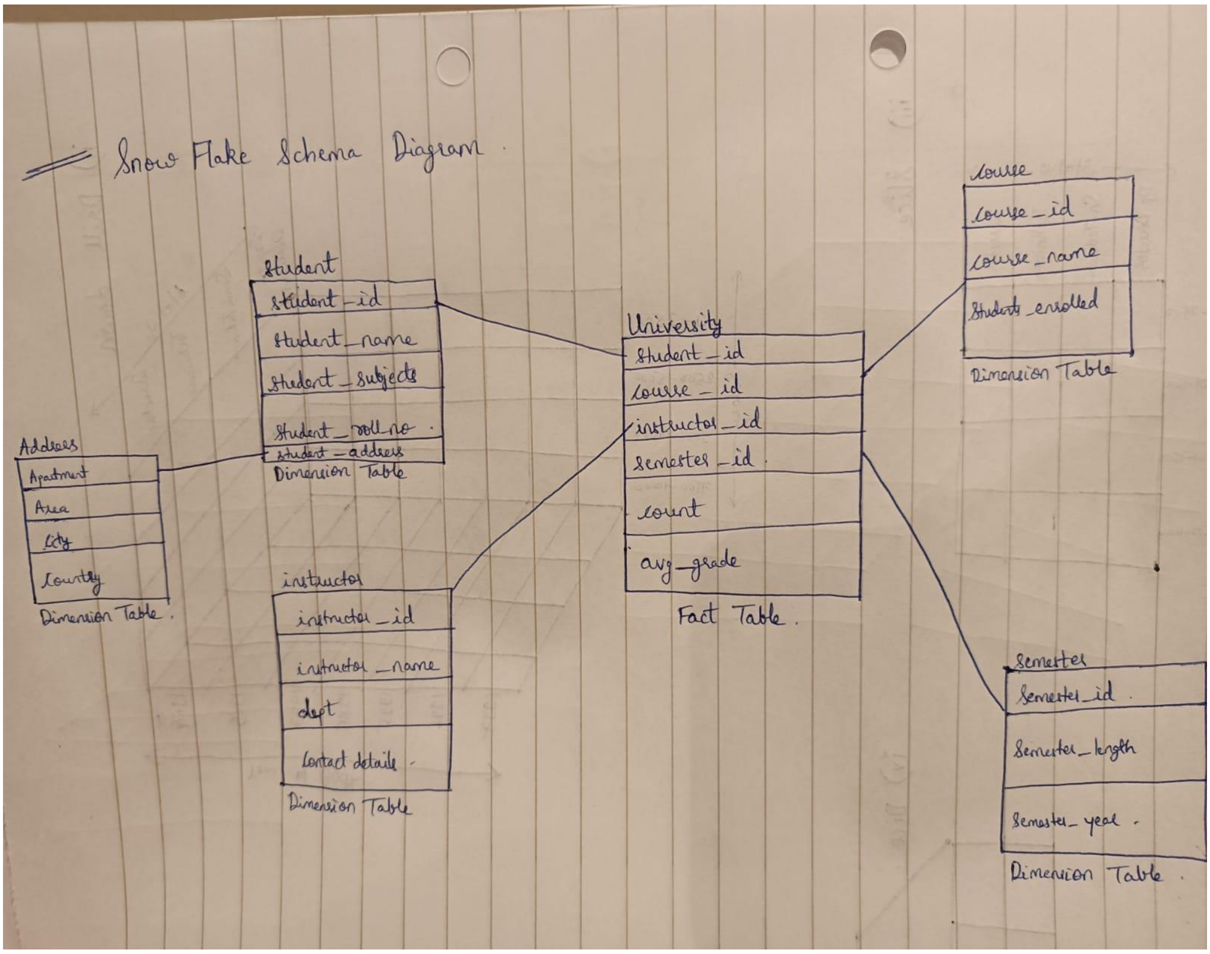


v) Pivot – In pivot, you rotate the data axes to form another representation of the data.



Question 3:

1)



2) Following are the OLAP operations that you can perform to list the average grade of CS courses for each Big University student.
i) Roll-up on student from (student key) to student_university.

ii) Dice on course - student with department =”CS” and university=” Big University”

iii) Drill-down on student from student_university to student_name

3) The cube will contain 5*5*5*5 = 625 cuboids.

4) Firstly, we need to make a connection with the database to create a table where we can store records and arrays of data. Here we use SQLAlchemy to establish a connection with the postgres database.

```
from sqlalchemy import create_engine
import psycopg2
engine = create_engine("postgresql://postgres:aniket123@localhost:5432/univdb")
```

5) Secondly, we need to create another connection to the Data Warehouse where you will be storing our datasets.

```
engine_dataset = create_engine("postgresql://postgres:aniket123@localhost:5432/testdb")
```

6) Created the dataset “input_DW_data.csv” and stored it in the data warehouse.

7)

```
engine.execute("CREATE TABLE IF NOT EXISTS student (name text PRIMARY KEY, id
integer)")

def write_record(name,id,engine):
   engine.execute("INSERT INTO student (name,id) VALUES ('%s','%s')" % (name,id))

def read_record(field,name,engine):
   result = engine.execute("SELECT %s FROM student WHERE name = '%s'" %
(field,name))
   return result.first()[0]

def update_record(field,name,new_value,engine):
   engine.execute("UPDATE student SET %s = '%s' WHERE name = '%s'" %
(field,new_value,name))

def write_dataset(name,dataset,engine):
   dataset.to_sql('%s' % (name),engine,index=False,if_exists='replace',chunksize=1000)
```

```
def read_dataset(name,engine):
    try:
        dataset = pd.read_sql_table(name,engine)
    except:
        dataset = pd.DataFrame([])
    return dataset

def list_datasets(engine):
    datasets = engine.execute("SELECT table_name FROM information_schema.tables
WHERE table_schema = 'public' ORDER BY table_name;")
    return datasets.fetchall()
```