|  |  |
| --- | --- |
|  | **Data Fellowship Program Batch 6**  by IYKRA | Analytics for Everyone |

**Practice Case - Coding (Python)**

|  |  |
| --- | --- |
| **Module name** | Coding - Python |
| **Key objectives** | To assess student’s understanding, competencies and skills in python programming language |
| **Tools needed** | Jupyter Notebook / VScode / Pycharm |
| ***Dataset A*** | {  "table\_name": "transactions",  "columns": [  {  "name": "id",  "type": "UUID"  },  {  "name": "insert\_time",  "type": "TIMESTAMP"  },  {  "name": "tx\_amount",  "type": "FLOAT"  },  {  "name": "tx\_type",  "type": "STRING"  },  {  "name": "status",  "type": "STRING"  }  ]  } |
| ***Dataset B*** |  |
| ***Dataset C*** | [  {  "user": {  "id": 1,  "name": "agus",  "email": "agus@example.com"  },  "txs": [  {  "id": 1,  "tx\_date": "2020-01-01",  "tx\_amount": 10,  "tx\_type": "buy"  },  {  "id": 3,  "tx\_date": "2020-02-01",  "tx\_amount": 10,  "tx\_type": "buy"  },  {  "id": 5,  "tx\_date": "2020-03-01",  "tx\_amount": 20,  "tx\_type": "sell"  }  ]  },  {  "user": {  "id": 2,  "name": "asep",  "email": "asep@example.com"  },  "txs": [  {  "id": 2,  "tx\_date": "2020-01-01",  "tx\_amount": 10,  "tx\_type": "sell"  },  {  "id": 4,  "tx\_date": "2020-02-01",  "tx\_amount": 10,  "tx\_type": "sell"  },  {  "id": 6,  "tx\_date": "2020-03-01",  "tx\_amount": 20,  "tx\_type": "buy"  }  ]  }  ] |
| **Effort (time/duration)** | 1 x 10 hours |
| **Students workflow** | * Read the questions * Find the dataset related to question * Prepare your jupyter notebook / IDE * Code for find the solution * Paste your code into gdocs and generate into pdf file |
| **Output** | PDF, uploaded on Olaas |

**Scenario**

1. Using Dataset A above, The table given to you cannot be updated, so if there is a change in the status of a transaction the new data is inserted instead of updating the status of the transaction, it will insert a new row with the same id and insert time based on the time when the row is inserted. You are assigned to create a view that only shows the most updated transactions for each id, so there is no duplicate id inside the view.
2. Using Dataset B above, Write a query to get the rolling sum total of sales using the window function!
3. Using Dataset C above, write a code with following requirements:
4. Your task is to create an automation (python script preferable) that convert JSON file above into three CSV files below:

CSV\_1 :

user\_id,tx\_id

1,1

1,3

1,5

2,2

2,4

2,6

CSV\_2 :

Id,name,email

1,agus,agus@example.com

2,asep,[asep@example.com](mailto:asep@example.com)

CSV\_3 :

Id,tx\_date,tx\_amount,tx\_type

1,2020-01-01,10,buy

2,2020-01-01,10,sell

3,2020-02-01,10,buy

4,2020-02-01,10,sell

5,2020-03-01,20,sell

6,2020-03-01,20,buy

**Aspects Rated Description (Total Score: 100)**

|  |  |
| --- | --- |
| ***Completeness*** | Right code with no error or issue (code running well)  Max Score: 80. |
| ***Creativity*** | Code wrapped with class or function (Not a hard code)  Max Score: 20 |