**Problem statement Introduction:**

Freshdesk, a helpdesk system, allows the export of activity information of all tickets. The export takes the following form:

{

"metadata": {

"start\_at": "20-04-2017 10:00:00 +0000",

"end\_at": "21-04-2017 09:59:59 +0000",

"activities\_count": 2

},

"activities\_data": [

{

"performed\_at": "21-04-2017 09:33:38 +0000",

"ticket\_id": 600,

"performer\_type": "user",

"performer\_id": 149018,

"activity": {

"note": {

"id": 4025864,

"type": 4

}

}

},

{

"performed\_at": "21-04-2017 09:38:24 +0000",

"ticket\_id": 704,

"performer\_type": "user",

"performer\_id": 149018,

"activity": {

"shipping\_address": "N/A",

"shipment\_date": "21 Apr, 2017",

"category": "Phone",

"contacted\_customer": true,

"issue\_type": "Incident",

"source": 3,

"status": "Open",

"priority": 4,

"group": "refund",

"agent\_id": 149018,

"requester": 145423,

"product": "mobile"

}

}

]

}

The status column can be any of the following values:

"Open"

"Closed"

"Resolved"

"Waiting for Customer"

"Waiting for Third Party"

"Pending"

Steps

* Write a Python program which will randomly generate realistic ticket data based on the above JSON format and store the data in a JSON file on disk. It should generate a random activity distribution for a configurable number of tickets. The program will be checked for realism of data, and for the ability to handle large amounts of records.Example: ticket\_gen -n 1000 -o activities.json to generate 1000 tickets with random activities into the activities.json file.
* Write a program (in a language if your choice) to read the above generated JSON file and store the data into a SQLite database in a relational format. The program will be checked for relational modelling.
* Write a SQL script that can be run on the database to generate the following attributes for each ticket:

Time spent Open (open to wfc)

Time spent Waiting on Customer (wfc to wtp)

Time spent waiting for response (Pending Status) (wfc to resolved)

Time till resolution (open to resolved)

Time to first response (open to wfc)

Example:| ticket\_id | time\_spent\_open | time\_spent\_waiting\_on\_customer | time\_spent\_waiting\_for\_response | time\_till\_resolution | time\_to\_first\_response | | 704 | 12 | 90 | 1200 | 1300 | 10 |

* Ensure all the above programs can be run in sequence using a bash script, Makefile, or equivalent.