□ Low-Level Design (LLD) – Hospital Visit Analysis (PySpark)

Difficulty Level: Easy | **Total Marks:** 15
Standards Followed: 5 Functions | 5 Visible Test Cases

□ **Summary of Design (PySpark Version)**

- □ DataFrames loaded via `SparkSession.read.csv()` in `driver.py`
- □ Transformation logic split into 5 functions in `solution.py`
- □ Joins, groupBy, sort, and filter used for analytics
- □ Outputs strictly conform to expected formats
- □ Suitable for scalable PySpark operations

□ **Concepts Tested**

□ Reading CSVs with `spark.read.csv()`
□ Performing joins with `.join()`
□ Aggregation with `groupBy().count()` and `agg()`
□ Sorting using `.orderBy()`
□ Filtering using `.isNull()`

□ **Problem Statement**

You are provided with two CSV files containing hospital records:

- `patients.csv` – Patient demographic information
- `visits.csv` – Visit IDs, patient IDs, and durations

Using PySpark, perform key analyses:

- Join the datasets
- Identify the patient with the most visits
- Identify the longest visit
- List patients who never visited
- Compute the average visit duration

□ **Operations**

□ 1. Join DataFrames
□ Perform an inner join on `patient_id` between patients and visits.
□ Function Prototype:

```
python
CopyEdit
def join_data(patients_df: DataFrame, visits_df: DataFrame) -> DataFrame:
```

□ Output: Merged DataFrame
□ Implementation Hint:

- Use `DataFrame.join()` with `on="patient_id"` and `how="inner"`
- Return the merged result

□ 2. Most Frequent Visitor
□ Find the patient_id who visited the hospital most frequently.
□ Function Prototype:

```
python
CopyEdit
def most_frequent_visitor(df: DataFrame) -> int:
```

□ Output: patient_id (int)
□ Implementation Hint:

- Use `groupBy("patient_id").count()`
- Sort by count in descending order using `.orderBy()`
- Use `.first()` to get the top patient

□ 3. Longest Visit
□ Get the visit_id of the longest visit duration.
□ Function Prototype:

```
python
CopyEdit
def longest_visit_id(df: DataFrame) -> int:
```

□ Output: visit_id (int)
□ Implementation Hint:

- Use `.orderBy(desc("duration"))`
- Use `.first()` to get the row with the longest visit
- Extract and return the `visit_id`

□ 4. Patients with No Visits
□ Return a list of patient_ids who never visited.
□ Function Prototype:

```
python
CopyEdit
def patients_with_no_visits(patients_df: DataFrame, visits_df: DataFrame) ->
list:
```

□ Output: List of integers
□ Implementation Hint:

- Perform a **left join** on `patient_id`
- Filter rows where `visit_id.isNull()`
- Collect and return only the `patient_id` values

□ 5. Average Visit Duration
□ Calculate and return the average visit duration.
□ Function Prototype:

```
python
CopyEdit
def average_visit_duration(visits_df: DataFrame) -> float:
```

□ Output: Float
□ Implementation Hint:

- Use `agg()` with `{"duration": "avg"}`
- Extract the result from the first row using `.collect()`

□ **Implementation Hints for solution.py**

```
python
CopyEdit
# □ solution.py
# □ Do not read CSVs here - use the DataFrames passed as arguments (driver
handles loading)

from pyspark.sql import DataFrame
from pyspark.sql.functions import desc

class HospitalAnalyzer:

    def join_data(self, patients_df: DataFrame, visits_df: DataFrame) ->
DataFrame:
        # Hint: Use .join() with how="inner" on patient_id
        pass
    def most_frequent_visitor(self, df: DataFrame) -> int:
        # Hint: groupBy patient_id, count(), then orderBy count desc and take
the top row
        pass

    def longest_visit_id(self, df: DataFrame) -> int:
        # Hint: order by duration descending and extract visit_id from top
row
        pass

    def patients_with_no_visits(self, patients_df: DataFrame, visits_df:
DataFrame) -> list:
        # Hint: left join patients to visits, filter where visit_id is null,
collect patient_id list
        pass

    def average_visit_duration(self, visits_df: DataFrame) -> float:
        # Hint: use agg({"duration": "avg"}) and extract float result from
row
        pass
```

□ Test Cases & Marks Allocation

Test Case ID	Description	Associated Function	Marks
TC1	Join on patient_id	join_data()	□ 3
TC2	Patient with most visits	most_frequent_visitor()	□ 3
TC3	Longest visit ID	longest_visit_id()	□ 3
TC4	Patients without visits	patients_with_no_visits()	□ 3
TC5	Average duration calculation	average_visit_duration()	□ 3

□ **Total Marks: 15**

□ Visible Test Cases (5)

□ **TC1**: Join DataFrames
□ Input: patients_df, visits_df
□ Output: merged DataFrame on patient_id

□ **TC2**: Most Frequent Visitor
□ Input: merged DataFrame
□ Output: patient_id with most entries

□ **TC3**: Longest Visit ID
□ Input: merged DataFrame
□ Output: visit_id with max duration

□ **TC4**: No Visit Patients
□ Input: patients_df and visits_df
□ Output: list of patient_ids with no visit record

□ **TC5**: Average Visit Duration
□ Input: visits_df
□ Output: average duration as float