✓ **Q. 6HU219** 1/1

- [x] Hybrid (i.e. both virtualization and materialization) approach ✓
- [ ] Straight Through Processing approach
- [ ] Data warehousing approach
- [ ] Virtualization approach

✗ **Q. 11987T** 0/1

- [ ] Use string matching techniques to find similarities
- [x] Normalize field names in the source data ✗
- [ ] Create separate fields for each synonym
- [ ] Apply manual mappings based on context

Correct answer

- [x] Use string matching techniques to find similarities
- [x] Apply manual mappings based on context

✓ **Q. 89BH26** 1/1

- [ ] 0.996
- [x] 0.896 ✓
- [ ] 0.921
- [ ] 0.912

✗ Q. 7BHU29

0/1

- [ ] It can lead to an increase in precision.
- [ ] It can decrease overall recall of the matching process.
- [ ] It can lead to false negatives if matches are missed.
- [x] It has no impact if using deterministic methods. ✗

Correct answer

- [x] It can lead to false negatives if matches are missed.
- [x] It can decrease overall recall of the matching process.

✓ Q. 893B67

1/1

- [x] (BranchID, BranchCity) ✓
- [ ] Branch_Managed_By
- [ ] (BranchCity, Branch_Managed_By)
- [ ] BranchCity

✗ Q. 789B62

0/1

- [ ] By defining transformations to standardize formats
- [ ] By creating new schema designs
- [x] By performing data normalization ✓
- [ ] By creating materialized views

Correct answer

- [x] By defining transformations to standardize formats
- [x] By performing data normalization

✗ Q. 289BG8 0/1

- [ ] Global AS View (GAV)
- [ ] Local AS View (LAV)
- [x] All of the above ✗
- [ ] Hybrid View (LAV)

Correct answer

- [x] Global AS View (GAV)

✓ Q. TUY342 1/1

- [ ] Rule-based systems are more accurate when handling large datasets
- [x] Learning-based algorithms can automatically adapt to new data patterns ✓
- [ ] Learning-based algorithms do not require data pre-processing
- [ ] Learning-based algorithms do not require labelled training data

✗ Q. 56B78G 0/1

- [x] Data normalization across sources ✗
- [x] Efficiency of materialized view creation ✗
- [x] Data integrity constraints ✓
- [x] Consistency of field names across sources ✓

Correct answer

- [x] Consistency of field names across sources
- [x] Data integrity constraints

✓ **Q. 2BHU81** 1/1

- [x] Jaccard Similarity ✓
- [x] Cosine Similarity ✓
- [x] Levenshtein Distance ✓
- [ ] K-means Clustering

✗ **Q. 6NH241** 0/1

- [ ] By identifying relevant data sources for integration
- [ ] By ensuring data quality across different schemas
- [x] By automating the process of schema integration ✗
- [ ] By providing context for field names and data types

Correct answer

- [x] By providing context for field names and data types

✓ **Q. U345T2** 1/1

- [x] All options given in this question ✓
- [ ] Extracting attributes
- [ ] Extracting entities
- [ ] Extracting relationships (between entities)

✓ **Q. 789T12** 1/1

- [ ] CustomerID as Blocking attribute and (FullName, BranchCity) as matching attributes
- [ ] (BranchCity, CustomerID) as Blocking attributes and FullName as matching attribute
- [x] BranchCity as Blocking attribute and FullName as matching attribute ✓
- [ ] BranchCity as Blocking attribute and (FullName, CustomerID) as matching attributes

✗ **Q. 89GH27** 0/1

- [x] Match 'FullName' first token (say FirstName) with the first token of 'Name' attribute and same for other tokens ✗
- [x] Match 'FullName' value and 'Name' value as one data string ✓
- [x] Normalize the 'FullName' and 'Name' attributes into 'First Name' and 'Last Name', and then match correspondingly ✗
- [ ] None of the Above

Correct answer

- [x] Match 'FullName' value and 'Name' value as one data string

✓ **Q. 189B32** 1/1

- [x] To provide a unified view of branch information across the CB and MFS ✓
- [ ] To provide a unified view of branch information across the CB, MFS and ZERO
- [ ] To summarize transaction data by branch
- [ ] To store customer information from all entities

✗ Q. Q789U1

0/1

- [ ] Naive Bayes classifier
- [x] Lookup table based pattern matching that normalizes common synonyms like “Marg” to “Road” ✓
- [x] Token-based similarity matching ✗
- [ ] Phonetic matching using Soundex

Correct answer

- [x] Lookup table based pattern matching that normalizes common synonyms like “Marg” to “Road”

✗ Q. A78B41

0/1

- [x] Jaro-Winkler Distance ✗
- [ ] Cosine Similarity
- [x] Jaccard Similarity ✗
- [ ] Levenshtein Distance

Correct answer

- [x] Cosine Similarity

✗ Q. 31B789

0/1

- [ ] To create materialized views
- [x] To align fields from different schemas to a common schema ✓
- [x] To ensure data consistency across multiple sources ✓
- [x] To optimize database performance ✗

Correct answer

- [x] To align fields from different schemas to a common schema
- [x] To ensure data consistency across multiple sources

✗ Q. 56BG17

0/1

- [x] Real-time updates ✗
- [x] Faster query performance ✓
- [x] Improved data consistency ✗
- [x] Simplified schema design ✗

Correct answer

- [x] Faster query performance

✓ Q. 781C25

1/1

- [ ] One DT and One FT
- [x] Two DT and One FT ✓
- [ ] Two DT and Two FT
- [ ] One DT and Two FT

✗ Q. 91245G

0/1

- [ ] SELECT District, State, SUM(TotalAmount) FROM Branch_Transaction_Summary, Branch_Info WHERE Branch_Transaction_Summary.BranchName=Branch_Info.BranchName GROUP BY CUBE (District, State);
- [ ] SELECT BranchName, District, State, SUM(TotalAmount) FROM Branch_Transaction_Summary, Branch_Info WHERE Branch_Transaction_Summary.BranchName=Branch_Info.BranchName GROUP BY CUBE (BranchName, District);
- [ ] SELECT BranchName, District, State, SUM(TotalAmount) FROM Branch_Transaction_Summary, Branch_Info WHERE Branch_Transaction_Summary.BranchName=Branch_Info.BranchName GROUP BY (BranchName, District, State);
- [x] SELECT District, State, SUM(TotalAmount) FROM Branch_Transaction_Summary, Branch_Info WHERE Branch_Transaction_Summary.BranchName=Branch_Info.BranchName GROUP BY (District, State); ✗

Correct answer

- [x] SELECT District, State, SUM(TotalAmount) FROM Branch_Transaction_Summary, Branch_Info WHERE Branch_Transaction_Summary.BranchName=Branch_Info.BranchName GROUP BY CUBE (District, State);

✗ Q. 2347WE

0/1

- [ ] All of the above
- [x] The phonetic similarity of names and addresses ✗
- [ ] Exact matches on names
- [x] The geographic region of the customer’s address ✗

Correct answer

- [x] All of the above

✓ **Q. 23489B** 1/1

- [x] All the options given in this question, but depends on ‘data string’ ✓
- [ ] Phonetic based Soundex
- [ ] Sequence-based
- [ ] Set-based

✓ **Q. 356B81** 1/1

- [x] Phonetic matching using Soundex ✓
- [ ] Token-based matching
- [ ] Cosine similarity
- [ ] Levenshtein Distance

✗ **Q. 184BN3** 0/1

- [x] String matching ✓
- [ ] Structural Similarity
- [ ] Data normalization
- [ ] Domain-specific knowledge

Correct answer

- [x] String matching
- [x] Domain-specific knowledge
- [x] Structural Similarity

✗ Q. 278BH2

0/1

- [x] Different naming conventions for similar fields ✓
- [x] Variations in data types for common fields ✓
- [x] Redundant data in source systems ✗
- [x] Inconsistent data formats ✓

Correct answer

- [x] Different naming conventions for similar fields
- [x] Variations in data types for common fields
- [x] Inconsistent data formats

✗ Q. 256BN0

0/1

- [x] Ontology-based matching ✓
- [ ] Machine learning models based matching
- [ ] Manual alignment
- [ ] Data profiling

Correct answer

- [x] Ontology-based matching
- [x] Machine learning models based matching
- [x] Manual alignment

✗ Q. 89UT31

0/1

- [ ] Curating the schema of Customer_Accounts table
- [x] Making the entity matching algorithm scalable ✓
- [ ] Achieving the desired accuracy
- [ ] Normalizing the entities of Customer_Accounts table

Correct answer

- [x] Achieving the desired accuracy
- [x] Making the entity matching algorithm scalable

✓ Q. 782B79

1/1

- [ ] Branches.BranchCity → Unified_Branches.OfficeCity
- [ ] Offices.OfficeCity → Unified_Branches.OfficeCity
- [x] Branches.BranchID → Unified_Branches.BranchID ✓
- [x] Offices.OfficeCode → Unified_Branches.BranchID ✓

✗ Q. 89G674

0/1

- [ ] To merge records that refer to the same customer across the CB and MFS
- [ ] To standardize the field names in the schemas
- [ ] To run the de-duplication algorithm by matching 'FullName' attribute string on Unified_Customers relation
- [x] To run the entity matching algorithm by matching 'FullName' attribute string on Unified_Customers relation ✗

Correct answer

- [x] To merge records that refer to the same customer across the CB and MFS

✗ Q. 4BH23I

0/1

- [ ] Data augmentation
- [ ] Exact string comparison
- [ ] Fuzzy matching
- [ ] Phonetic algorithms (e.g., Soundex)

✗ Q.978V45

0/1

- [ ] Exact matching
- [ ] Levenshtein distance
- [ ] Jaro similarity score
- [ ] Entity resolution

✗ Q. 4BA627

0/1

- [ ] CREATE MATERIALIZED VIEW Unified_Customers AS SELECT * FROM Customers_Accounts JOIN Customers_Profiles ON Customers_Accounts.BranchCity = Customers_Profiles.OfficeCity;
- [ ] CREATE TABLE Unified_Customers (CustomerID INT PRIMARY KEY, FullName VARCHAR(255), AccountNumber VARCHAR(20), BranchCity VARCHAR(50), Balance DECIMAL(15, 2));
- [ ] CREATE TABLE Unified_Customers AS SELECT * FROM Customers_Accounts UNION SELECT * FROM Customers_Profile;
- [x] CREATE VIEW Unified_Customers AS SELECT CustomerID, FullName, AccountNumber, BranchCity, Balance FROM Customers_Accounts UNION SELECT ClientNumber AS CustomerID, Name AS FullName, AccountCode AS AccountNumber, OfficeCity AS BranchCity, Invest_Amount AS Balance; ✓

Correct answer

- [x] CREATE TABLE Unified_Customers (CustomerID INT PRIMARY KEY, FullName VARCHAR(255), AccountNumber VARCHAR(20), BranchCity VARCHAR(50), Balance DECIMAL(15, 2));
- [x] CREATE VIEW Unified_Customers AS SELECT CustomerID, FullName, AccountNumber, BranchCity, Balance FROM Customers_Accounts UNION SELECT ClientNumber AS CustomerID, Name AS FullName, AccountCode AS AccountNumber, OfficeCity AS BranchCity, Invest_Amount AS Balance;

✗ Q. 613B78

0/1

- [ ] It is used to identify transactions by their unique identifier
- [ ] It maps directly to branch information
- [ ] It connects customer accounts with their transaction records
- [ ] It links transactions to customers in the unified schema

✓ **Q. 789BU1** 1/1

- [ ] Learning based method
- [x] Given a list of names and generate the variant of these names ✓
- [ ] Rule-based method
- [ ] Defining regular expressions for person-name entities

✗ **Q. 562UV2** 0/1

- [x] It cannot handle data inconsistencies ✗
- [ ] It results in high false positives
- [x] It can only handle exact matches and fails to detect variations in the data ✓
- [ ] It requires significant computation time

Correct answer

- [x] It can only handle exact matches and fails to detect variations in the data

✗ **Q. 220112** 0/1

- [x] Jaccard Similarity Measure ✗
- [x] Levenshtein Distance Measure ✗
- [x] Jaro Measure ✓
- [ ] Soundex Measure

Correct answer

- [x] Jaro Measure

✓ Q. WER671 1/1

- [ ] Handling the Differences in country codes for phone numbers
- [x] Handling the spelling variations due to typographical errors ✓
- [ ] Handling the presence of numerical digits in the names
- [ ] Handling the lack of address data

✗ Q. 41G678 0/1

- [x] They ensure data consistency across sources ✗
- [x] They improve query performance by storing precomputed results ✓
- [x] They provide real-time data updates ✗
- [x] They help in optimizing schema mapping ✗

Correct answer

- [x] They improve query performance by storing precomputed results

✓ Q. 6NUTB1 1/1

- [ ] High precision always indicates high recall.
- [x] Recall measures the ratio of true positives to the total actual positives. ✓
- [ ] None of the above
- [x] Precision measures the ratio of true positives to the total predicted positives. ✓