

**Part I. Virtual Data Integration and Data Cleaning**

1. Given the following relational database:

<i>album</i> (<u>album_id</u> , title, year, <u>artist_id</u>)	<i>artist_id</i> : FK(<i>artist</i>)
<i>artist</i> (<u>artist_id</u> , name, country)	
<i>song</i> (<u>song_id</u> , title, duration, <u>album_id</u>)	<i>album_id</i> : FK(<i>album</i>)

- a) Write a query expression in Datalog that returns the title and duration of songs by artists from Portugal.

```
Q(T, D) :- song(_, T, D, X), album(X, _, _, Y), artist(Y, _, 'Portugal')
```

- b) Write a Datalog expression for the same query, except that artists can be either from Portugal or Brazil.

```
Q(T, D) :- song(_, T, D, X), album(X, _, _, Y), artist(Y, _, 'Portugal')
Q(T, D) :- song(_, T, D, X), album(X, _, _, Y), artist(Y, _, 'Brazil')
```

- c) Consider the concept of query equivalence. What is the role of this concept in data integration? Describe a scenario where it is important to prove that two query expressions are equivalent.

Query equivalence plays an important role in query reformulation. Given a query over a mediated schema, this query must be reformulated into a query over the underlying data sources, or over existing views over those data sources. Query equivalence is a means to prove that the resulting query over the data sources is equivalent to the original query over the mediated schema.

2. Answer the following questions:

- a) Suppose you have two query expressions: $Q1(X) :- Q2(X, Y), Q3(Y, Z), Q4(Z, T), T > 0$ and $Q'1(X) :- Q2(X, Y), Q3(Y, Y), Q4(Y, T), T > 2$. If these two queries are proved to be equivalent, what can you conclude?

If the two queries are equivalent, their results are the same. Therefore, it is possible to conclude that there are no Z values different from Y, and there are no T values between 0 and 2.

- b) Suppose $Q1$ is a Local-as-View schema mapping, written as $Q1(X) \subseteq Q2(X, Y), Q3(Y, Z), Q4(Z, T), T > 0$. What is the meaning of the symbol \subseteq in this expression? In practice, what does it mean?

The symbol \subseteq means that the results for the expression on the left side are a subset of the results for the expression on the right side. In Local-as-View, this means that the data source $Q1$ is defined as a view over the mediated schema $Q2, Q3, Q4$. The expression for that view is given on the right side.



3. The table *artist(artist_id, name, country)* may contain duplicate records, where the *country* is the same but the *name* is slightly different. The idea is to use the Jaccard measure to compute the similarity between names.

- a) For the two names below, indicate the values of $|B_x \cap B_y|$ and $|B_x \cup B_y|$ that should be placed in the formula of the Jaccard measure using bi-grams. Show how you arrived at those values.

LimpBizkit
LimpBiscuit

$$jaccard(x, y) = \frac{|B_x \cap B_y|}{|B_x \cup B_y|}$$

LimpBizkit	{#L, Li, im, mp, pB, Bi, iz, zk, ki, it, t#}	11 bigrams
LimpBiscuit	{#L, Li, im, mp, pB, Bi, is, sc, cu, ui, it, t#}	12 bigrams
= = = = = =	= =	8 common bigrams

$$\begin{aligned}|B_x \cap B_y| &= 8 \\ |B_x \cup B_y| &= 11 + 12 - 8 = 15\end{aligned}$$

- b) If *country* can also be misspelled, present a formula to identify pairs of potential duplicates based on both fields (*name* and *country*). Explain the meaning of every term in that formula.

$$\text{sim_total} = \text{weight_name} * \text{sim_name} + \text{weight_country} * \text{sim_country}$$

- c) With a list of pairs of approximate duplicates, how do you proceed to remove the duplicates from the original table? Explain what you would do to consolidate the records into a table without duplicates.

First, it may be necessary to use transitive closure to find more pairs of duplicates based on the given ones. Then, once all clusters of duplicates have been identified, I would choose for each field the most common value within each cluster. This means that each cluster will yield a single record with the most common values in that cluster. The id for the record corresponds to the cluster id.

4. In the database of question 1, some songs may have no album (i.e. they are released outside an album).

- a) How can you use a data profiling tool (such as DataCleaner) to find those songs? Describe in words what you would do when using that tool for this purpose.

Songs without an album will have a NULL in *album_id*. In DataCleaner, I would define a data source as a database connection to access the database. Then I would apply a Completeness Analyzer to the song table, selecting the column *album_id* to be analyzed. Finally, in the results, I would have a list of all the songs which have NULL in *album_id*.



- b) Using a data profiling tool, it is possible to find the number of artists by country. But this type of query can be also done in an analytical (OLAP) tool. Therefore, what is the difference between data profiling and data analysis tools? Explain these concepts and the purpose of both tools.

Data profiling is a means to gather statistics and discover anomalies in the data. For example, if the distribution of artists by country should be uniform, but there is a country with many more artists than the rest, then this could be an anomaly in the data. On the other hand, data analysis is a means to gain insight into the business. In this case, if there is a country with many more artists than the rest, then the company should probably realize that this is its biggest market.

Part II. Data Warehousing and OLAP

For the questions below, consider a data warehouse for a music streaming platform (e.g. Spotify). The DW has three dimensions (dim_user, dim_song, dim_date) and a fact table (fact_plays) that stores how many times a given user has played a given song on a given date. The measure is n_plays. Each dimension has the following levels:

dim_user: name -> city -> country
dim_song: title -> album -> artist
dim_date: day -> month -> year

5. Answer the following questions:

- a) How many transformations and jobs would you need to implement an ETL process for this DW in PDI? Briefly describe the purpose of each job/transformation.

I would need one transformation to load each dimension table, plus one transformation to load the fact table. Once all these transformations have been developed, I could use a job to execute the whole ETL process at once, by running the transformations for the dimension tables first, and finally running the transformation for the fact table.

- b) Present the relational schema (star schema) for the data warehouse above using the following notation, where FK means foreign key:

table1(primary key, attribute1, attribute2) *attribute2: FK(table2)*

Use surrogate keys in all dimension tables.

```
dim_user(user_key, name, city, country)
dim_song(song_key, title, album, artist)
dim_date(date_key, day, month, year)
fact_plays(user_key, song_key, date_key, n_plays)
    user_key: FK(dim_user)
    song_key: FK(dim_song)
    date_key: FK(dim_date)
```



- c) Based on the relational schema that you just presented, write an SQL query to get a ranking of artists based on their total number of plays. The artist with most plays should appear at the top.

```
SELECT dim_song.artist, SUM(fact_plays.n_plays) AS total_plays
FROM fact_plays NATURAL JOIN dim_song
GROUP BY dim_song.artist
ORDER BY total_plays DESC
```

- d) Write a single query in SQL/OLAP that returns the union of all the following results:
- Number of plays by song title.
 - Number of plays by album.
 - Number of plays by artist.

```
SELECT SUM(fact_plays.n_plays)
FROM fact_plays NATURAL JOIN dim_song
GROUP BY GROUPING SETS ((dim_song.title), (dim_song.album), (dim_song.artist))
```

6. Consider the concept of *slowly changing dimensions* (SCDs).

- a) In the data warehouse above, give an example of a dimension that could become an SCD, and give an example of a change that could occur in that dimension in order to justify a SCD.

The dimension dim_user could be a slowly changing dimension. For example, if a user moves to another city, aggregating the number of plays by city might give different results depending on whether it is before or after the user moved. In this case, there would be two (or more) versions of the same user.

- b) What type of support does PDI provide for SCDs? Which steps does it provide and how do you configure those steps for SCDs?

PDI provides support for Type 2 SCDs by allowing multiple versions of the same record with validity intervals. This can be done with the Dimension Lookup/Update step by configuring the surrogate key (technical key) field, the version field, and the date range fields (from ... to ...).

7. Consider the dim_song dimension in the data warehouse above.

- a) What kind of hierarchy should be used if we want to include albums that are compilations of songs from multiple artists? Justify.



If an album belongs to multiple artists, then there is a many-to-many relationship between album and artist. Also, if a song can appear in multiple albums/compilations, then there is a many-to-many relationship between title and album. In both cases, it is a non-strict hierarchy.

- b) Suppose that your DW is ready for MDX queries in Saiku, but someone asks you to add a new top level (*record label*) to the dim_song dimension. The hierarchy should be: title -> album -> artist -> label. What is the impact of this change? Describe all the changes that you would have to do to your implementation.

- First, we would have to add the *label* field into the dim_song SQL table.
- We would need to change the transformation for dim_song. In PDI, we would need to include the *label* field, by modifying the Table Input step (to read it from the database) and the Insert/Update step (to load it into the DW).
- In Pentaho Schema Workbench, we would need to add a new top-level to the song hierarchy.
- We would have to upload the new XML cube definition into Saiku.

8. Answer the following questions:

- a) Suppose the data cube is called Plays. How would you write the following query in MDX?
Show the number of plays by country and year, for the artist Eminem.

```
SELECT User.Country.Members ON ROWS,  
       Time.Year.Members ON COLUMNS  
FROM Plays  
WHERE Song.Artist.Eminem
```

- b) Using a reporting tool such as Pentaho Report Designer (PRD), is it possible to build a single report with the results of both the MDX query in 8.a) and the SQL query in 5.c)? Justify your answer.

Yes, it is possible. In PRD it is possible to define one or more data sources to be used in a report. Each data source may have a different query. The report can then be configured to use the results of those queries for instance by drag-and-drop of its elements to the report sections.