A. Summarize one real-world written business report that can be created from the DVD Dataset from the “Labs on Demand Assessment Environment and DVD Database” attachment.

A topic of interest to a DVD rental store may be the occurrences of each actor in the films that are in the stores inventory. This would allow the store to see which actors are popular for organization or advertising purposes.

1. Identify the specific fields that will be included in the detailed table and the summary table of the report.

- **Summary Table:**
  - **actor_name VARCHAR(100)**
  - **actor_film_count INTEGER**
- **Detailed Table:**
  - **actor_name VARCHAR(100)**
  - **actor_film_count INTEGER**
  - **actor_id INTEGER**
  - **current_month VARCHAR(9)**

2. Describe the types of data fields used for the report.

Integer and varchar will be used.

3. Identify *at least* **two** specific tables from the given dataset that will provide the data necessary for the detailed table section and the summary table section of the report.

Tables that will provide data for this report are the film, film_actor, and actor tables.

4. Identify *at least* **one** field in the detailed table section that will require a custom transformation with a user-defined function and explain why it should be transformed (e.g., you might translate a field with a value of N to No and Y to Yes).

The current month value in the detailed table will use a custom transformation for readability. This is done because updates are monthly so showing the current month makes this easy to understand

5. Explain the different business uses of the detailed table section and the summary table section of the report.

The detailed section can be used to obtain more specific information about the films and actors. This includes the ids of each listed actor and the current month

The summary section would be used if the only information that is needed is the number of movies each actor has been in as of that month.

6. Explain how frequently your report should be refreshed to remain relevant to stakeholders.

The report should be refreshed monthly, once there are enough new films to get actor data from.

B. Provide original code for function(s) in text format that perform the transformation(s) you identified in part A4.

```
CREATE OR REPLACE FUNCTION get_month_str(input_date TIMESTAMP)
RETURNS TEXT AS $$
DECLARE
 month_str TEXT;
BEGIN
 month_str := TO_CHAR(input_date, 'Month');
 RETURN month_str;
END;
$$ LANGUAGE plpgsql;
```

C. Provide original SQL code in a text format that creates the detailed and summary tables to hold your report table sections.

```
CREATE TABLE detailed_actor_frequency (
  actor_id INT,
   actor_name VARCHAR(100),
   film_count INT,
   current_month VARCHAR(9)
);

CREATE TABLE actor_frequency_summary (
  actor_name VARCHAR(100),
   film_count INT
);
```

D. Provide an original SQL query in a text format that will extract the raw data needed for the detailed section of your report from the source database.

```
INSERT INTO detailed_actor_frequency
SELECT
```

```
    a.actor_id AS actor_id,
    CONCAT(a.first_name || ' ' || a.last_name) AS actor_name,
    COUNT(f_a.film_id) AS film_count,
    get_month_str(a.last_update) AS current_month
FROM actor a
JOIN film_actor f_a ON a.actor_id = f_a.actor_id
GROUP BY a.actor_id
ORDER BY film_count DESC;
```

E. Provide original SQL code in a text format that creates a trigger on the detailed table of the report that will continually update the summary table as data is added to the detailed table.

```
--create trigger function to fill summary
  CREATE OR REPLACE FUNCTION summary_trigger()
  RETURNS TRIGGER AS $$
  BEGIN
        DELETE FROM actor_frequency_summary;
        INSERT INTO actor_frequency_summary
        SELECT actor_name, film_count
        FROM detailed_actor_frequency
        ORDER BY film_count DESC;
        RETURN NEW;
  END;
  $$ LANGUAGE plpgsql;

  --execute trigger function on change on detailed table
  CREATE TRIGGER refresh_actor_frequency_summary_trigger
  AFTER INSERT OR UPDATE OR DELETE ON detailed_actor_frequency
  FOR EACH STATEMENT
  EXECUTE FUNCTION summary_trigger();
```

F. Provide an original stored procedure in a text format that can be used to refresh the data in *both* the detailed table and summary table. The procedure should clear the contents of the detailed table and summary table and perform the raw data extraction from part D.

```
CREATE OR REPLACE PROCEDURE refresh_actor_data()
  LANGUAGE plpgsql
  AS $$
  BEGIN
        DELETE FROM detailed_actor_frequency;
        DELETE FROM actor_frequency_summary;
```

```
        INSERT INTO detailed_actor_frequency (actor_id, actor_name, film_count,
current_month)
        SELECT
        a.actor_id AS actor_id,
        CONCAT(a.first_name, ' ', a.last_name) AS actor_name,
        COUNT(f_a.film_id) AS film_count,
        get_month_str(a.last_update) AS current_month
        FROM actor a
        JOIN film_actor f_a ON a.actor_id = f_a.actor_id
        GROUP BY a.actor_id, a.first_name, a.last_name, a.last_update
        ORDER BY film_count DESC;
  END;
  $$;

  --execute procedure
  CALL refresh_actor_data();
```

1. Identify a relevant job scheduling tool that can be used to automate the stored procedure.

    If the system is unix-based a good option for automation would be Cron. This tool allows for SQL procedures to be stored and executed at a set time interval.

H. Acknowledge all utilized sources, including any sources of third-party code, using in-text citations and references. If no sources are used, clearly declare that no sources were used to support your submission.

No sources were used aside from the provided videos.