MỘT SỐ LƯU Ý TRONG 5 PROJECT UDACITY

Mục lục

Project 1: Data Modeling with Postgres

Note 1: Datatype

Thường thì mọi người khi chạy file test.ipynb dễ bị mắc warning ở cột start_time trong bảng songplays.

```
[WARNING] Type 'bigint' may not be an appropriate data type for column 'start_time' in the 'songplays' table.
[WARNING] You may want to add appropriate NOT NULL constraints to the 'songplays' table.
```

Ở lỗi này mọi người chú ý là nên sử dụng kiểu dữ liệu timestamp thay vì bigint nhé.

Một điểm nữa mà bạn cần chú ý là cột songplay_id trong bảng songplays, cột này nên set cho nó tự động tăng vì vậy nên để là SERIAL

Note 2: Primary Keys

Mỗi bảng đều nên có một cột làm primary key, thường thì đó là cột mang id hoặc thời gian.

Note 3: NOT NULLs

Khi tạo bảng trong file sql_queries.py, thường thì bạn sẽ khó mà đặt đúng cột nào nên chọn là NOT NULL để khỏi bị ăn warning từ test.ipynb.
Do đó, tips ở đây là bạn nên xem qua file test.ipynb trước để xem họ check những cột nào NOT NULL thì quay ngược lại thêm vào cho file sql_queries.py như vậy giúp bạn tiết kiệm thời gian chạy đi chạy lại để test.

Note 4: Kết quả test sau khi chạy etl.py

Vì tập dataset là khá nhỏ và các cột song_id và artist_id của bảng songplays chứa rất nhiều giá trị null.

Do đó khi chạy dòng lệnh check cả 2 cột này đều không null thì chỉ có 1 dòng.

```
%sql SELECT * FROM songplays WHERE (song_id is not null and artist_id is not null)
```

```
 * postgresql://student:***@127.0.0.1/sparkifydb
1 rows affected.
```

songplay_id	start_time	user_id	level	song_id	artist_id	session_id	location	user_agent
4108	2018-11-21 21:56:47.796000	15	paid	SOZCTXZ12AB0182364	AR5KOSW1187FB35FF4	818	Chicago-Naperville-Elgin, IL-IN-WI	"Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/36.0.1985.125 Chrome/36.0.1985.125 Safari/537.36"

Nếu kết quả hiện ra như trên thì bạn đã làm đúng.

Note 5: INSERT…ON CONFLICT

Đối với hầu hết các bảng thì khi sẽ là “DO NOTHING” tuy nhiên với bảng user thì cần phải update cột level nhé.

Note 6: Docstrings

Hãy nhớ thêm docstrings vào mỗi function trong file etl.py

Project 2: Data Modeling with Apache Cassandra

Note 1: Not use ALLOW FILTERING
Trong các câu lệnh SELECT, bạn không nên sử dụng SELECT * vì như vậy dễ dẫn đến việc lãng phí bộ nhớ và đưa ra các cột dữ liệu mà có thể không dùng đến.
Thay vì đó hãy chọn đúng các cột cần query và viết cụ thể chúng cho câu lệnh SELECT.

Note 2: Thứ tự của các cột khi khai báo PRIMARY KEY
Trong các câu lệnh CREATE và INSERT thì thứ tự các cột nên được sắp xếp theo thứ tự của các COMPOSITE PRIMARY KEY và cột CLUSTING.
Điều này cần lưu ý vì Cassandra sẽ lưu trữ dữ liệu phân tán đến các node của cluster.
Cho dễ hiểu thì nó sẽ giống như ví dụ sau:

```
CREATE TABLE IF NOT EXISTS database_name
(a int, e text, i float, o int, u text)
PRIMARY KEY (a, e, i)
```

Thứ tự khai báo của a, e, i trùng với trong PRIMARY KEY.

Note 3: Mô tả sơ lược về câu lệnh truy vấn
Trong notebook, trên mỗi câu lệnh truy vấn nên sử dụng markdown để mô tả mục đích của câu lệnh SELECT. Ngoài ra, nên thể hiện rõ là đang sử dụng partition key, composite key như thế nào.

Note 4: Xoá các comment
Ở bài này reviewer chú trọng vào clean code nên là các comment nào không cần thiết thì các bạn nên xoá sạch để tránh bị ăn chửi.

Project 3: Data Warehouse

Note 1: Data type
Chú ý đồng bộ kiểu dữ liệu ở các cột giống nhau hoặc trích xuất giữa các bảng staging, dimension và fact.

Note 2: IDENTITY
Trong bảng songplays, cột songplay_id cần được định nghĩa là tăng tự động thì khác với postgres sử dụng SERIAL như ở project 1 thì trong bài này ta sử dụng IDENTITY.

Note 3: NOT NULL
Đặt ràng buộc NOT NULL đúng trường cần thiết, không nên đặt ràng buộc này cho tất cả các trường.

Note 4: Handle duplicates
Trong các câu lệnh INSERT hãy sử dụng keyword DISTINCT cho cột chỉ id của bảng nhằm tránh việc duplicate dữ liệu trong các bảng dimension.

Note 5: docstring
Nhớ thêm docstring vào từng hàm ở hai file etl.py và create_tables.py

Note 6: PEP8 style
Hãy chắc rằng code trong các file .py được định dạng theo chuẩn pep8

Note 7: distkey/sortkey
Trong bài này không bắt buộc việc định nghĩa distkey/sortkey, tuy nhiên bạn nên sử dụng chúng nhằm mục đích tăng performance cho pipeline và nó cũng sẽ dùng khá thường xuyên ở các dự án thật.

Project 4: Data Lake

Note 1: drop Duplicate
Ở mỗi câu lệnh tạo bảng nên sử dụng lệnh drop_duplicates() và subset để chỉ rõ drop duplicate trên cụ thể cột nào của bảng.

Note 2: Partition
Sử dụng write.partitionBy() để định nghĩa việc ghi file parquet và phân tán chia dữ liệu theo cột nào

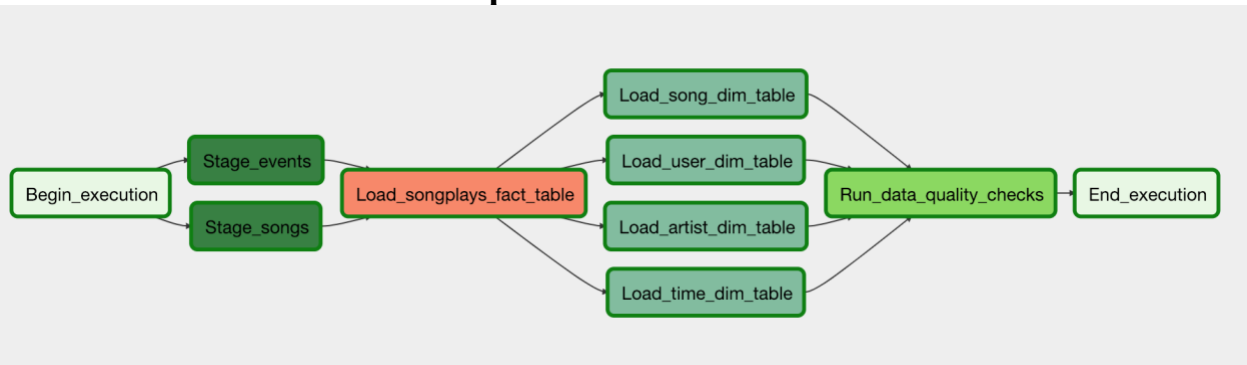
Note 3: songplay_id trong bảng songplays
songplay_id để có thể cho phép nó tăng một cách tự động thì ta sử dụng monotonically_increasing_id()

Note 4: fixed Schema
Để đảm bảo sử dụng đúng data type, bạn nên sử dụng StructType và StructField để cố định schema.

Note 5: PEP8 style
Hãy chắc rằng code trong các file .py được định dạng theo chuẩn pep8

Project 5: Data Pipelines

Note 1: Create Table Step



Project yêu cầu tạo pipeline theo các bước như hình trên, tuy nhiên sẽ thiếu đi bước tạo bảng ban đầu, để xử lý nó bạn có thể làm theo hai cách sau:

- Cách 1: Code SQL trên Redshift để tạo table
- Cách 2: Tạo task create table ở chính bước Begin_execution