**Домашнє завдання: Реалізація DAG у Apache Airflow**

**Опис виконаного завдання**

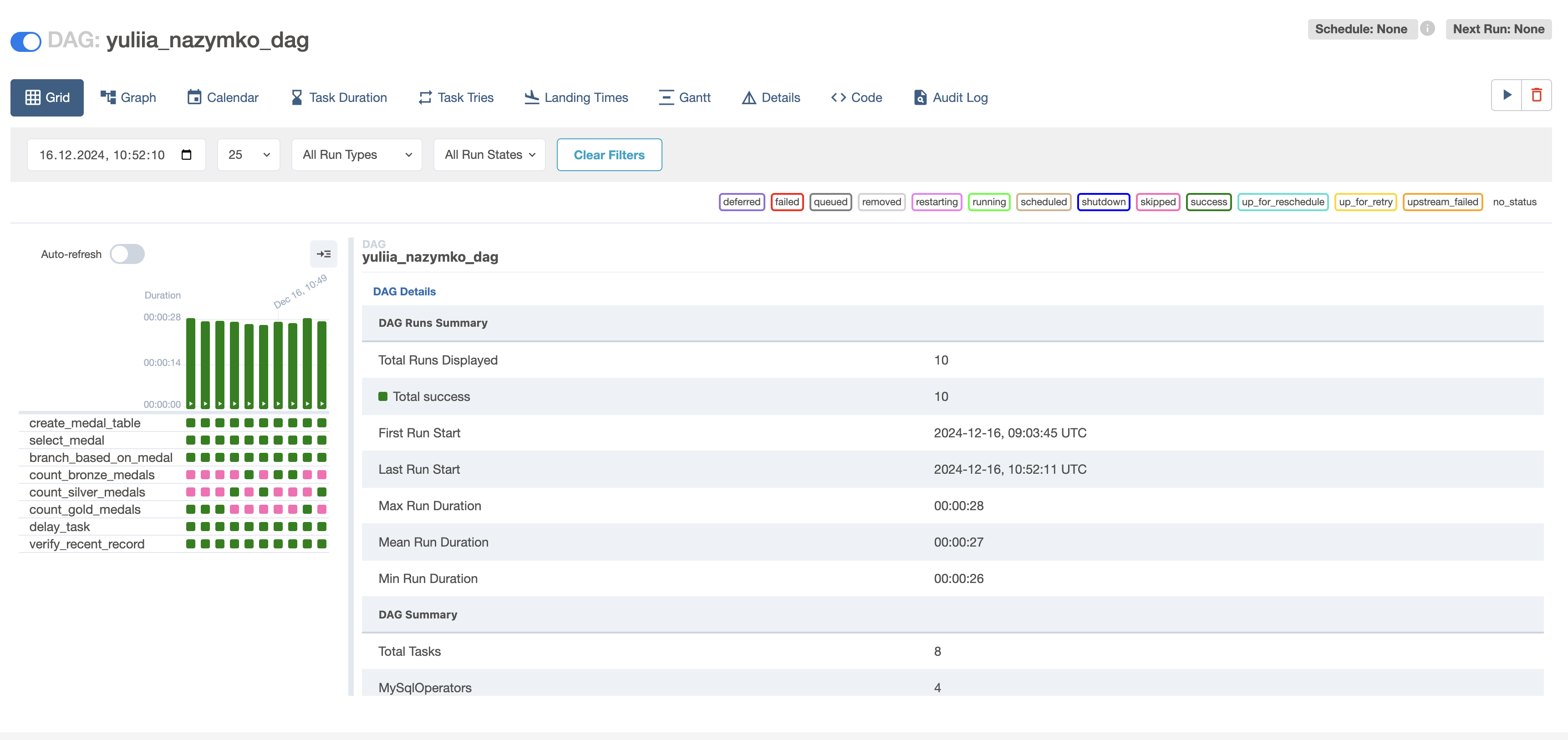
В рамках даного завдання був створений **Airflow DAG** під назвою yuliia\_nazymko\_dag, який виконує наступні етапи:

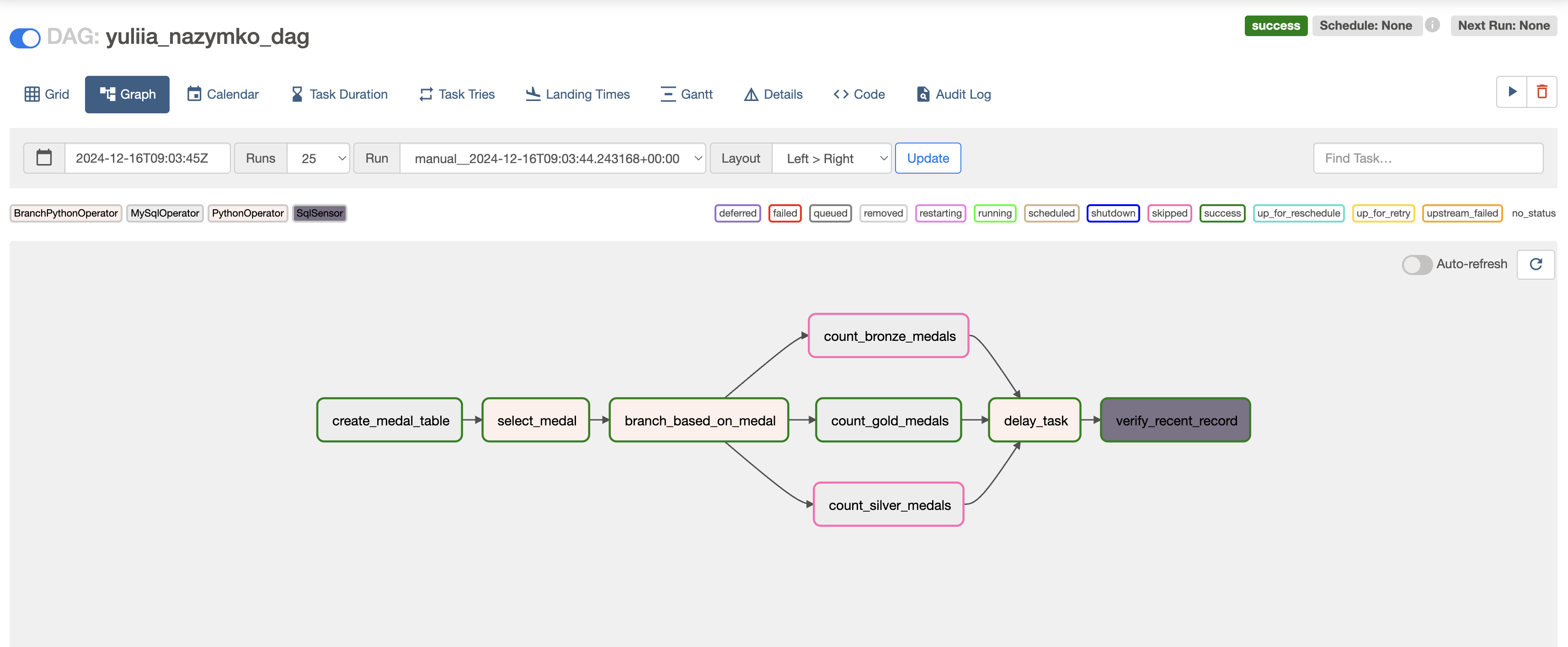
1. **Створення таблиці**  
   Завдання create\_medal\_table створює таблицю neo\_data.yuliia\_nazymko\_medal\_counts для збереження підсумкових даних про медалі.
2. **Випадковий вибір медалі**  
   Використовується функція random\_medal\_choice, щоб випадково обрати одну з медалей: Gold, Silver або Bronze.
3. **Розгалуження логіки**  
   На основі випадково обраної медалі відбувається розгалуження за допомогою BranchPythonOperator. Подальше виконання переходить до одного з трьох завдань:
   * count\_bronze\_medals
   * count\_silver\_medals
   * count\_gold\_medals
4. **Підрахунок медалей**  
   Завдання для підрахунку медалей додають записи у таблицю yuliia\_nazymko\_medal\_counts з підрахунком кількості медалей за відповідним типом на основі даних з olympic\_dataset.athlete\_event\_results.
5. **Затримка виконання**  
   Для імітації тривалої обробки використовується завдання delay\_task, що викликає затримку у 20 секунд.
6. **Перевірка наявності нових даних**  
   За допомогою SqlSensor виконується перевірка, чи з'явилися нові записи у таблиці протягом останніх 30 секунд.

**Висновок**  
DAG успішно автоматизує процес:

* Вибору випадкової медалі.
* Підрахунку медалей з використанням MySQL-запитів.
* Запису результатів у базу даних.
* Контролює наявність нових даних у результатах завдяки сенсору.

**Скриншоти результатів**

**Результати запусків**: Лог та статуси виконання DAG.

**Структура DAG**: Граф виконання завдань.

**Результати з таблиці MySQL**.

