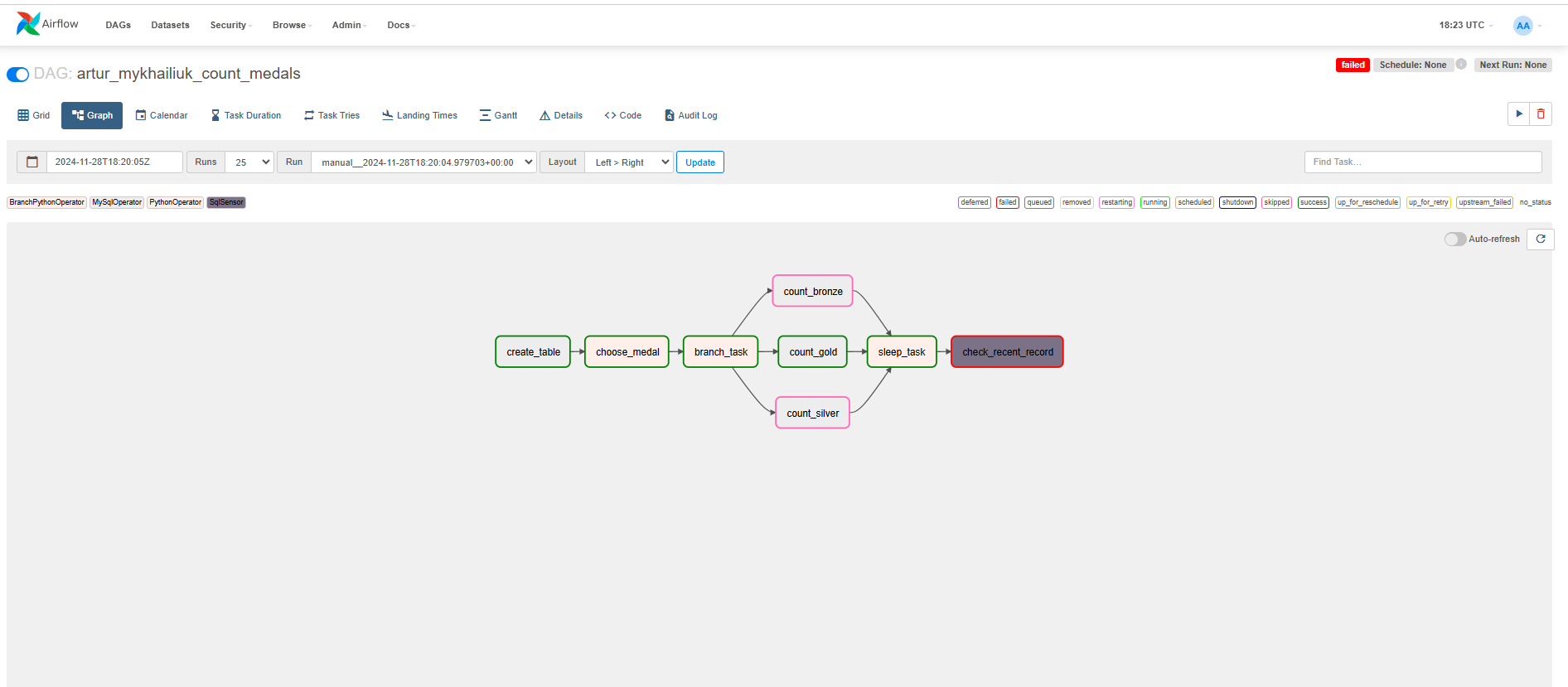
## **1.Загальна структура DAG**

**** У цьому скріншоті відображено граф мого DAG. Граф складається з наступних 5 завдань:

* create\_table - створеня таблиці в БД (з перевіркою чи така таблиця вже існує у базі);
* choose\_medal - вибір випадкової медалі з 3-ох можливих;
* branch\_task - тут відбувається підрахунок кількості тієї чи іншої медалі в залежності від того, яка медаль була обрана у попередньому завданні;
* sleep\_task - реалізація затримки виконання наступного завдання;
* check\_recent\_record - перевірка, чи найновіший запис у таблиці не старший за 30 секунд. У випадку, якщо запис був успішно записаний у таблицю, завдання буде у статусі success, якщо ні - failed

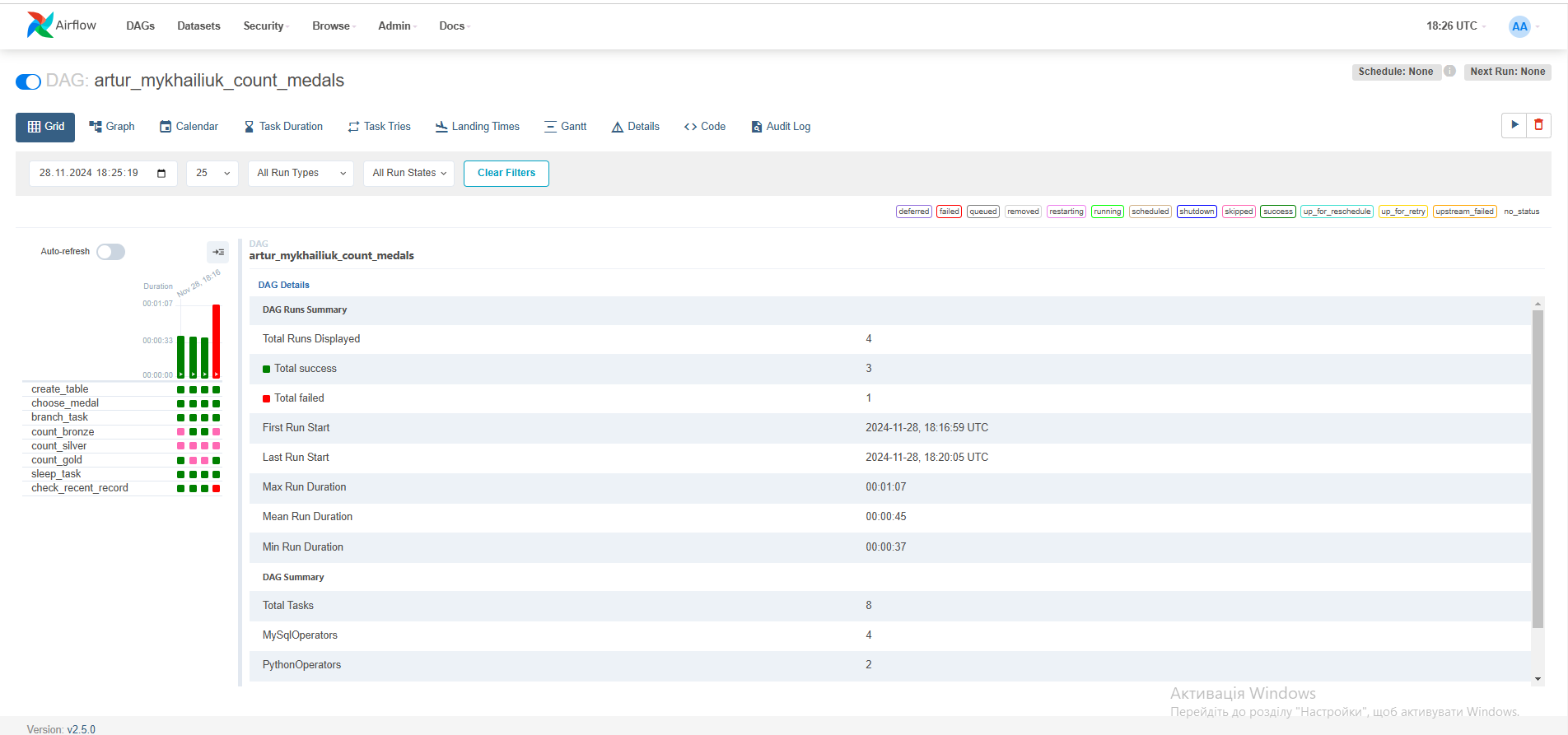
Якщо детально проаналізувати даний граф, то ми можемо прийти до висновку, що:

* відбулося успішне створення таблиці чи виконалась перевірка на існування таблиці в БД - success;
* рандомний вибір медалі - success
* відбувся підрахунок кількості медалей Gold і запис в таблицю - success;
* затримка виконання наступного завдання на 35 сек. - success;
* перевірка, чи дійсно записано результат в таблицю - false (імітація, результат було записано в таблицю). Основний висновок, що якщо умова в SqlSensor не виконується певний час, то завдання набуває статусу - Failed

## 

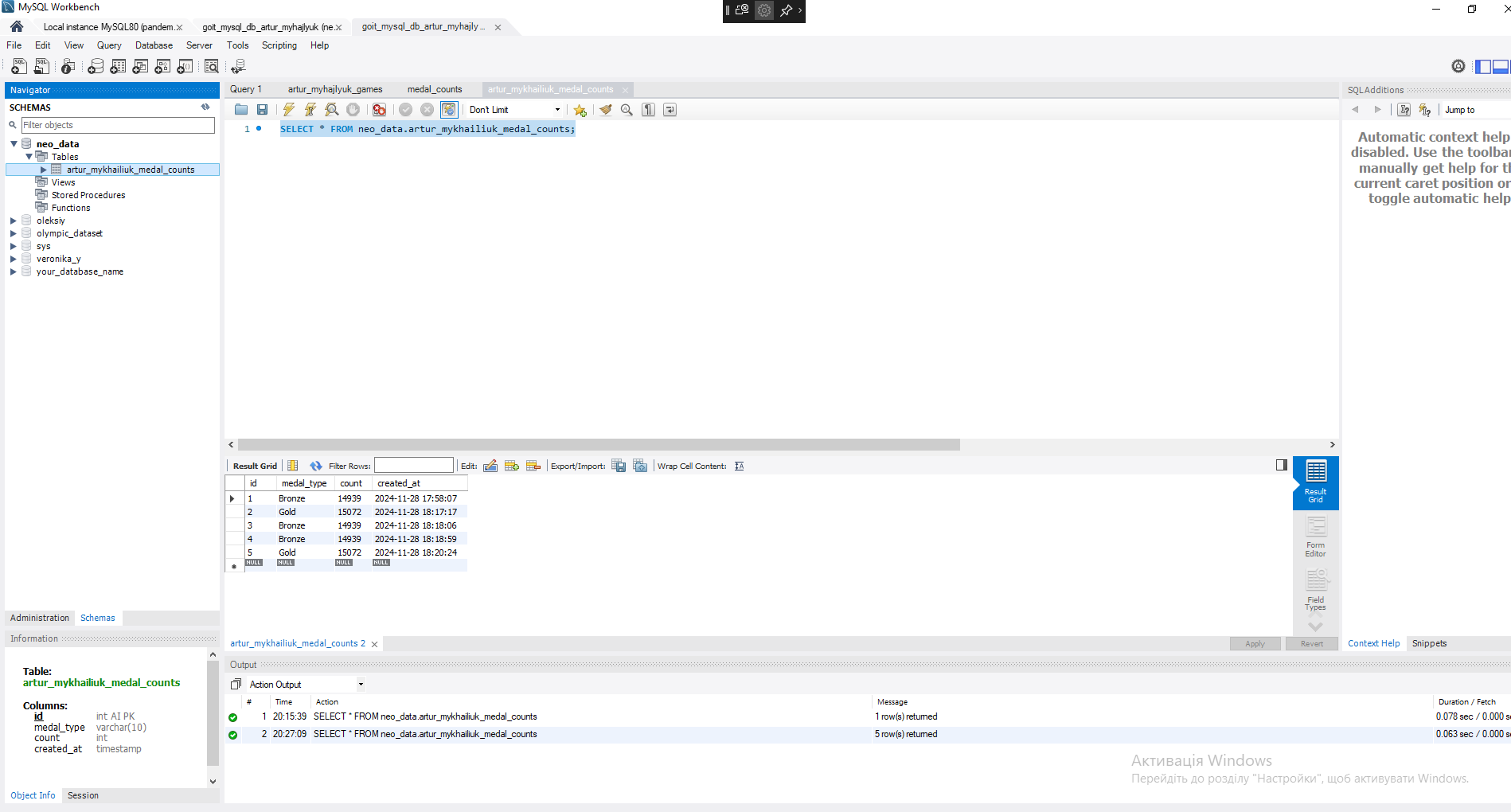
## 

## **2. Грід з результатами виконання DAG**

**** На цьому гріді відображено результати виконнаня DAG:

* Всього запусків - 4;
* Успішних запусків - 3 (позначено зеленим кольором);
* Неуспішних запусків - 1 (позначено червоним кольором);
* Дата та час першого та останнього запуску;
* Мінімальна, максимальна та середня тривалість запусків;
* Оператори, що застосовуються в DAG

## **3. Фактичний результат виконання DAG**

**** Даний скріншот є підтвердженням того, що було створено таблицю та під час запуску DAG результати обробки потрапляли у цю таблицю