

Apply filters to SQL queries

Project description

Nasz zespół analityków wykrył podejrzany incydent. W związku z tym chcemy przeprowadzić dochodzenie, w którym ustalimy szereg informacji, tj. czas, data, poprawność logowania oraz miejsce prób logowania.

W drugiej części nasz zespół ma za zadanie przeprowadzić, niezbędne dla prawidłowego działania firmy. aktualizacje systemów dla wybranych działów. Do tego niezbędne będzie ustalenie urządzeń, które wymagają aktualizacji.

Retrieve after hours failed login attempts

Aby przefiltrować wyniki użyliśmy filtrów `-> WHERE login_time > '18:00:00';`
Tutaj został użyty operator porównania '>', który w tym przypadku oznacza (większe niż).
Otrzymaliśmy dzięki temu wszystkie próby logowania po godzinie 18:00.

Następnie zespół analityków zawęził zakres wyników do NIEUDANYCH prób logowania po godzinie 18:00. Do tego zostały użyte 2 warunki połączone **operatorem logicznym 'AND'**.

```
-> WHERE login_time > '18:00:00' AND success=0;
```

Pierwszy warunek został opisany wyżej, natomiast drugi 'success = 0' jest wyrażeniem logicznym (boolowskim), gdzie 0 oznacza FAŁSZ, a 1 oznacza PRAWDĘ. W SQL, wartości logiczne są często reprezentowane jako 0 i 1. Jako, że chcemy uzyskać tylko NIEUDANE próby logowania, ustawiliśmy tę wartość na 0.

Retrieve login attempts on specific dates

Nasz zespół zauważył, że podejrzane próby logowania z dnia '2022-05-09'. Postanowiono sprawdzić wszystkie próby logowania z tego dnia i poprzedniego. Jako, że chcieliśmy uzyskać dane z obu dni, do tego został zastosowany filtr składający się z dwóch argumentów i zastosowano operator logiczny '**OR**'. Ten operator sprawia, że otrzymamy wszystkie wyniki spełniające któryś z tych dwóch warunków. W tym przypadku dwa, interesujące nas dni.

```
WHERE login_date ='2022-05-09' OR login_date='2022-05-08';
```

Retrieve login attempts outside of Mexico

Zespół analityków postanowił przyjrzeć się próbom logowania spoza Meksyku.

W tym przypadku aby wyszukać próby logowania spoza Meksyku zastosowaliśmy filter

```
-> WHERE country NOT LIKE 'Mex%';
```

Meksyk został zapisany w tabeli w różne sposoby (MEX i MEXICO). Dlatego użyliśmy operatora **LIKE**, dzięki któremu filtrujemy wszystkie wyrazy rozpoczynające się na litery **'Mex'**, a pozwala na to znak **'%'** użyty po **'Mex'**. Został tu również użyty operator **NOT**, który neguje funkcję. Normalnie filter **'WHERE country LIKE 'Mex%'**; wyszukał by rekordy zaczynające się na **'Mex'**. Dzięki operatora **NOT** funkcja jest odwrotna. Otrzymujemy wyniki, które NIE zaczynają się na **'Mex'**. W ten sposób otrzymaliśmy wszystkie próby logowania spoza Meksyku.

Retrieve employees in Marketing

W tym przypadku nasz zespół miał za zadanie zaktualizować komputery wybranego działu (**Marketing**) i z konkretnego skrzydła budynku (**East**). Aby wyszukać interesujące nas wyniki użyliśmy filtru:

```
WHERE department='Marketing' AND office LIKE 'East%';
```

Mamy tutaj dwa warunki połączone operatorem **AND**, który sprawia że zostaną wyświetlone wyniki które spełniają oba warunki jednocześnie. Zostały tutaj użyte operatory:

'=' zwraca rekordy z dokładnie ustaloną nazwą

'AND' który sprawia że zostaną wyświetlone wyniki które spełniają oba warunki jednocześnie

LIKE który pozwala na wyszukiwanie wzorców w tekście, w połączeniu z symbolem

% który oznacza dowolne znaki i dowolną ich liczbę.

Retrieve employees in Finance or Sales

Tym razem zespół miał za zadanie przeprowadzić aktualizację systemu wszystkich pracowników dwóch innych działów (Finance i Sales). W tym celu został zastosowany filtr:

```
WHERE department='Sales' OR department='Finance';
```

Jako, że zespół chciał uzyskać dane z obu oddziałów - został użyty operator **'OR'**. Ten operator sprawia, że otrzymamy wszystkie wyniki spełniające któryś z tych dwóch warunków.

Retrieve all employees not in IT

Czas na ostatnią aktualizację dla pracowników. Zespół ustalił, że komputery działu 'Information Technology' zostały już wcześniej zaktualizowane, więc nie wymagają tego. Do tego potrzebowaliśmy informacji o wszystkich działach, z wyjątkiem 'Information Technology'.

Zastosowaliśmy w tym celu filter:

```
WHERE NOT department='Information Technology';
```

Występuje tutaj operator NOT (zaprzeczenie), który w połączeniu z dalszym argumentem oznacza, że otrzymamy informacje o wszystkich działach, z wyjątkiem 'Information Technology'. Dzięki czemu mogliśmy wykonać ostatnią, niezbędną aktualizację.

Summary

Nasz zespół wykonał wszystkie powierzone im zadania.

Został ustalony dzień, w którym wystąpiło podejrzanе logowanie. Sprawdzono również inne dni, logowania poza normalnych czasie pracy oraz różne regiony. Te informacje zostały przekazane do działu dochodzeniowego.

Wszystkie niezbędne aktualizacje zostały przeprowadzone zgodnie z instrukcją. Nie napotkano żadnych problemów.