

Aplicar filtros a consultas SQL

Descrição do Projeto

Minha empresa está trabalhando para melhorar a segurança do sistema. Minha função é garantir que tudo esteja protegido, investigar possíveis vulnerabilidades e atualizar os computadores dos funcionários quando necessário. A seguir, mostro como utilizei SQL com filtros para executar tarefas relacionadas à segurança.

Recuperar tentativas de login com falha após o expediente

Houve um possível incidente de segurança após o horário comercial (depois das 18h). Todas as tentativas de login após o expediente que falharam nesse período precisam ser investigadas. O código a seguir mostra como criei uma consulta SQL para filtrar as tentativas de login malsucedidas que ocorreram após o expediente.

```
MariaDB [organization]> clear
MariaDB [organization]> SELECT *
  ->
  -> FROM log_in_attempts
  ->
  -> WHERE login_time > '18:00' AND success = FALSE;
```

event_id	username	login_date	login_time	country	ip_address	success
2	apatel	2022-05-10	20:27:27	CAN	192.168.205.12	0
18	pwashing	2022-05-11	19:28:50	US	192.168.66.142	0
20	tshah	2022-05-12	18:56:36	MEXICO	192.168.109.50	0

Foram 19 tentativas de login com falha:

```
199 | yappiah | 2022-05-11 | 19:34:48 | MEXICO | 192.168.44.232 | 0
```

event_id	username	login_date	login_time	country	ip_address	success
199	yappiah	2022-05-11	19:34:48	MEXICO	192.168.44.232	0

19 rows in set (0.102 sec)

A primeira parte da captura de tela mostra minha consulta, e a segunda parte exibe uma parte da saída. Essa consulta tem como objetivo filtrar as tentativas de login mal sucedidas que aconteceram após as 18h.

Para isso, comecei selecionando todos os dados da tabela **log_in_attempts**. Depois, usei a cláusula **WHERE** com o operador **AND** para refinar os resultados e exibir apenas as tentativas de login que ocorreram após as 18h e não foram bem-sucedidas. A primeira condição, **login_time > '18:00'**, filtra as tentativas feitas após esse horário. A segunda condição, **success = FALSE**, garante que só apareçam as tentativas mal sucedidas.

Recuperar tentativas de login em datas específicas

Houve um evento suspeito em 09/05/2022, e qualquer tentativa de login realizada nesse dia ou no dia anterior precisa ser investigada.

O código a seguir mostra como criei uma consulta SQL para filtrar as tentativas de login com base em datas específicas.

```
MariaDB [organization]> SELECT *
->
-> FROM log_in_attempts
->
-> WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';
```

event_id	username	login_date	login_time	country	ip_address	success
1	jrafael	2022-05-09	04:56:27	CAN	192.168.243.140	1
3	dkot	2022-05-09	06:47:41	USA	192.168.151.162	1
4	dkot	2022-05-08	02:00:39	USA	192.168.178.71	0
8	bisles	2022-05-08	01:30:17	US	192.168.119.173	0
12	dkot	2022-05-08	09:11:34	USA	192.168.100.158	1

A primeira parte da captura de tela mostra a consulta que criei, e a segunda parte exibe um trecho da saída. Essa consulta retorna todas as tentativas de login realizadas em 09/05/2022 ou 08/05/2022.

Foram 75 tentativas de login com falha neste período de dois dias:

```
+-----+-----+-----+-----+-----+-----+
-+
75 rows in set (0.001 sec)
```

Para isso, comecei selecionando todos os dados da tabela **log_in_attempts**. Em seguida, utilizei a cláusula **WHERE** com o operador **OR** para filtrar os resultados e exibir apenas as tentativas de login ocorridas nas datas desejadas. A primeira condição, **login_date = '2022-05-09'**, filtra os logins de 09/05/2022, e a segunda condição, **login_date = '2022-05-08'**, filtra os logins de 08/05/2022.

Recuperar tentativas de login fora do México

Após analisar os dados da organização sobre as tentativas de login, percebi que há um possível problema com aquelas realizadas fora do México e elas precisam ser investigadas.

O código a seguir ilustra como criei uma consulta SQL para filtrar as tentativas de login que ocorreram em locais fora do México.

```
MariaDB [organization]> SELECT *
->
-> FROM log_in_attempts
->
-> WHERE NOT country LIKE 'MEX%';
+-----+-----+-----+-----+-----+-----+
-+
| event_id | username | login_date | login_time | country | ip_address      | success |
+-----+-----+-----+-----+-----+-----+
-+
|         1 | jrafael  | 2022-05-09 | 04:56:27  | CAN     | 192.168.243.140 |      1  |
|         2 | apatel   | 2022-05-10 | 20:27:27  | CAN     | 192.168.205.12  |      0  |
|         3 | dkot     | 2022-05-09 | 06:47:41  | USA     | 192.168.151.162 |      1  |
```

A primeira parte da captura de tela mostra minha consulta, enquanto a segunda exibe uma parte do resultado. Essa consulta foi elaborada para retornar todas as tentativas de login feitas a partir de países diferentes do México. O processo começou com a seleção de todos os dados da tabela **log_in_attempts**. Em seguida, apliquei uma cláusula **WHERE** utilizando **NOT** para excluir os registros do México. Para garantir que tanto "MEX" quanto "MÉXICO" fossem considerados, utilizei o operador **LIKE** com o padrão "MEX%", onde o sinal de porcentagem (%) representa qualquer sequência de caracteres.

Foram 144 tentativas de login com falha feitas fora do México:

```
144 rows in set (0.061 sec)
```

Recuperar funcionários do departamento de Marketing

Minha equipe planeja atualizar os computadores de alguns funcionários do departamento de Marketing. Para isso, preciso identificar quais máquinas precisam ser atualizadas.

O código abaixo mostra como desenvolvi uma consulta SQL para filtrar as máquinas dos funcionários que estão no departamento de Marketing e no prédio Leste, num total de 7.

```
MariaDB [organization]> SELECT *
->
-> FROM employees
->
-> WHERE department='Marketing' AND office LIKE 'East%';
+-----+-----+-----+-----+-----+
| employee_id | device_id | username | department | office |
+-----+-----+-----+-----+-----+
|          1000 | a320b137c219 | elarson | Marketing | East-170 |
|          1052 | a192b174c940 | jdarosa | Marketing | East-195 |
|          1075 | x573y883z772 | fbautist | Marketing | East-267 |
|          1088 | k865l965m233 | rgosh | Marketing | East-157 |
|          1103 | NULL | randerss | Marketing | East-460 |
|          1156 | a184b775c707 | dellery | Marketing | East-417 |
|          1163 | h679i515j339 | cwilliam | Marketing | East-216 |
+-----+-----+-----+-----+-----+
7 rows in set (0.001 sec)
```

A primeira parte da captura de tela exibe minha consulta, enquanto a segunda mostra uma amostra do resultado. Essa consulta retorna todos os funcionários do departamento de Marketing que trabalham no prédio Leste. O processo começou com a seleção de todos os dados da tabela de funcionários. Em seguida, apliquei uma cláusula **WHERE** com **AND** para filtrar os registros dos funcionários que pertencem ao departamento de Marketing e estão localizados no prédio Leste. Usei o operador **LIKE** com o padrão "East%" para corresponder aos dados da coluna de escritório, já que o edifício Leste é representado de forma variável, incluindo o número do escritório. A primeira condição, **department = 'Marketing'**, filtra os funcionários desse departamento, enquanto a segunda, **office LIKE 'East%'**, filtra os funcionários localizados no prédio Leste.

Recuperar funcionários dos departamentos de Finanças ou Compras

As máquinas dos funcionários dos departamentos Financeiro e Vendas também precisam ser atualizadas. Como as atualizações de segurança para esses departamentos são diferentes, preciso identificar os funcionários exclusivamente desses dois setores.

O código abaixo mostra como desenvolvi uma consulta SQL para filtrar as máquinas dos funcionários dos departamentos Financeiro e Vendas.

```
MariaDB [organization]> SELECT *
->
-> FROM employees
->
-> WHERE department = 'Finance' OR department = 'Sales';
```

employee_id	device_id	username	department	office
1003	d394e816f943	sgilmore	Finance	South-153
1007	h174i497j413	wjaffrey	Finance	North-406
1008	i858j583k571	abernard	Finance	South-170
1009	NULL	lrodriqu	Sales	South-134
1010	k242l212m542	jlansky	Finance	South-109
1011	l748m120n401	drosas	Sales	South-292
1015	p61lq262r945	jsoto	Finance	North-271
1017	r550s824t230	iclark	Finance	North-188

A primeira parte da captura de tela exibe minha consulta, enquanto a segunda mostra uma parte do resultado. Esta consulta retorna todos os funcionários dos departamentos Financeiro e Vendas. O processo começou com a seleção de todos os dados da tabela de funcionários. Em seguida, apliquei uma cláusula **WHERE** com **OR** para filtrar os funcionários que pertencem aos departamentos Financeiro ou Vendas. Utilizei o operador **OR** em vez de **AND**, pois quero incluir funcionários de qualquer um dos dois departamentos. A primeira condição, **department = 'Finance'**, filtra os funcionários do departamento Financeiro, enquanto a segunda condição, **department = 'Sales'**, filtra os funcionários do departamento de Vendas.

Recuperar todos os funcionários que não são do departamento de TI

Minha equipe precisa realizar uma nova atualização de segurança para os funcionários que não pertencem ao departamento de Tecnologia da Informação. Para isso, é necessário primeiro obter informações sobre esses funcionários.

A seguir, mostro como criei uma consulta SQL para filtrar as máquinas dos funcionários que não estão no departamento de Tecnologia da Informação.

```
MariaDB [organization]> SELECT *
->
-> FROM employees
->
-> WHERE NOT department = 'Information Technology';
```

employee_id	device_id	username	department	office
1000	a320b137c219	elarson	Marketing	East-170
1001	b239c825d303	bmoreno	Marketing	Central-276
1002	c116d593e558	tshah	Human Resources	North-434
1003	d394e816f943	sgilmore	Finance	South-153
1004	e218f877g788	eraab	Human Resources	South-127

A primeira parte da captura de tela mostra a minha consulta, enquanto a segunda exibe uma amostra dos resultados. A consulta retorna todos os funcionários que não estão no departamento de Tecnologia da Informação. O processo começou com a seleção de todos os dados da tabela de funcionários. Em seguida, apliquei uma cláusula **WHERE** com **NOT** para filtrar os funcionários que não pertencem a esse departamento.

O resultado é um total de 161 funcionários:

1198	q308r573s459	jmartine	Marketing	South-117
1199	r520s571t459	areyes	Human Resources	East-100

161 rows in set (0.001 sec)

Resumo

Utilizei filtros em consultas SQL para extrair informações específicas sobre tentativas de login e computadores de funcionários. Trabalhei com duas tabelas distintas: **log_in_attempts** e **employees**. Para filtrar os dados conforme as necessidades de cada tarefa, empreguei os operadores **AND**, **OR** e **NOT**. Também utilizei o operador **LIKE** juntamente com o curinga de porcentagem (%) para filtrar padrões específicos.