

Queries on a single table without any aggregate

SID:12310401

Name: Ziheng Wang

1.

	Filter	Filter	Filter	Filter	Filter
>	15	Carlota Joaquina - Princesa do br		1995	(NULL)
>	26	Dona Flor e Seus Dois Maridos br		1976	110
>	70	O Quatrilho	br	1995	(NULL)
>	74	Orfeu Negro	br	1959	107
>	80	Pixote	br	1981	128
>	262	O Cangaceiro	br	1953	105
>	913	O Pagador de Promessas	br	1962	98
>	935	Central do Brasil	br	1998	113

2.

1 -- Active: 1727157595209@127.0.0.1@5432@lab1@public PostgreSQL lab1
2 ▷ Run | New Tab | JSON

3 select * from movies
4 where country = 'us'
5 and year_released = 2004; 29ms

movies ×
搜索结果集 导出 耗时: 33ms 1 > Total 89

	* movieid integer	* title varchar(100)	* country character(2)	* year_released integer	runtime integer
>	203	Ray	us	2004	112
>	278	Crash	us	2004	112
>	292	Million Dollar Baby	us	2004	75
>	906	Fahrenheit 9/11	us	2004	122
>	1255	The Incredibles	us	2004	115
>	1256	Shark Tale	us	2004	90
>	1257	The Forgotten	us	2004	91
>	1258	Collateral	us	2004	120

3.

```
▷ Run | New Tab | JSON  
6 select * from movies  
7 where country = 'sp' and lower(title) not like '%o%' and lower(title) not like '%a%'  
8
```

movies

搜索结果集

* movieid integer
* title varchar(100)
* country character(2)
* year_released integer
runtime integer

	Filter	Filter	Filter	Filter	Filter
>	1984	Buried	sp	2010	95
>	4228	REC	sp	2007	75
>	5127	Tesis	sp	1996	125
>	8029	Intruders	sp	2011	100
>	8613	Che	sp	2008	257

4.

▷ Run | New Tab | JSON

```
2 select * from movies where country in ('cn','hk','tw','mo') and year_released between 1940 and 1949;
```

movies X

	Filter	Filter	Filter	Filter	Filter
>	6343	Bāqiān Lǐ Lù Yún Hé Yuè	cn	1947	124
>	6344	Lìréng Xìng	cn	1949	102
>	7152	Tiě Shàn Gōngzhǔ	cn	1941	73
>	7256	Ai Le Zhongnian	cn	1949	100
>	9006	Xiǎochéng Zhī Chūn	cn	1948	98
>	9017	Wūyā Yǔ Máquè	cn	1949	111
>	9018	Yī Jiāng Chūn Shuǐ Xiàng Dōng	cn	1947	190
>	9029	Wànjiā Dēnghuǒ	cn	1948	121

5.

15 Run | New Tab | JSON

14 select * from people where born <= '1920' and died is null; 43ms

people

搜索结果集

* peopleid integer first_name varchar(30) * surname varchar(30) * born integer died integer * gender character(1)

	Filter	Filter	View On Dialog	Filter	Filter	Filter
>	340	Michael	Anderson	1920	(NULL)	M
>	824	Karuna	Bandyopadhyay	1919	(NULL)	F
>	2159	Mary	Carlisle	1914	(NULL)	F
>	2358	Marge	Champion	1919	(NULL)	F
>	3384	Eddie	Davis	1903	(NULL)	M
>	3481	Chandrabati	Debi	1909	(NULL)	F
>	3509	Suzy	Delair	1917	(NULL)	F
>	3821	Kirk	Douglas	1916	(NULL)	M
>	5395	Lee	Grant	1920	(NULL)	F

6.

15

▷ Run | New Tab | JSON

16 select * from alt_titles where title like '%山%'; 21ms

alt_titles

搜索结果集



导出

* titleid
integer* movieid
integer* title
varchar(250)

	Filter	Filter	Filter
>	362	9111	巴山夜雨
>	763	8224	道士下山
>	1008	9109	观音山
>	1197	6690	聯合艦隊司令長官 山
>	1845	9027	狼山喋血记
>	1862	7180	庐山恋
>	2165	9034	那山那人那狗
>	2449	7155	人山人海
>	2484	1466	山椒大夫

7.

Run | New Tab | JSON | Copy

```
20 select * from movies
21 where upper(title) like 'MAN %'
22 or upper(title) like 'MAN''%
23 or upper(title) like '% MAN %'
24 or upper(title) like '% MAN''%
25 or upper(title) like '% MAN'; 46ms
```

movies

搜索结果集

* movieid integer
* title varchar(100)
* country character(2)
* year_released integer
runtime integer

		Filter	Filter	Filter	Filter
>	125	The Man Who Knew Too Much	gb	1934	75
>	126	The Man Who Knew Too Much	us	1956	120
>	168	The Third Man	gb	1949	(NULL)
>	220	The Man Who Shot Liberty Val	us	1962	123
>	265	A Man for All Seasons	gb	1966	120
>	301	Rain Man	us	1988	133
>	436	Dead Man Walking	us	1995	122
>	586	Little Big Man	us	1970	139
>	612	Man Hunt	us	1941	105

8.

Run | New Tab | JSON

27

select * from people where died >= born + 100; 25ms

people

x

搜索结果集



1



耗时: 30ms

1

>

Total 43

Q

* peopleid
integer

first_name
varchar(30)

* surname
varchar(30)

* born
integer

died
integer

* gender
character(1)

	Filter	Filter	Filter	Filter	Filter	Filter
>	16	George	Abbott	1887	1995	M
>	872	Richard L.	Bare	1913	2015	M
>	1149	Bruce	Bennett	1906	2007	M
>	1689	Carlo Ludovico	Bragaglia	1894	1998	M
>	1944	George	Burns	1896	1996	M
>	2523	Liting	Chen	1910	2013	M
>	2980	Irwin	Corey	1914	2017	M
>	3016	Diosa	Costello	1913	2013	F
>	3328	Danielle	Darrieux	1917	2017	F

9.

Run | New Tab | JSON

```
select * from people  
where died >= born + 100  
or (died is null and born <= 1924); | 23ms
```

people

	* peopleid integer	first_name varchar(30)	T * surname varchar(30)	* born integer	died integer	* gender character(1)
>	16	George	Abbott	1887	1995	M
>	340	Michael	Anderson	1920	(NULL)	M
>	665	Charles	Aznavour	1924	(NULL)	M
>	824	Karuna	Bandyopadhyay	1919	(NULL)	F
>	872	Richard L.	Bare	1913	2015	M
>	998	Cate	Bauer	1922	(NULL)	F
>	1074	Mumtaz	Begum	1923	(NULL)	F
>	1149	Bruce	Bennett	1906	2007	M
>	1689	Carlo Ludovico	Bragaglia	1894	1998	M

10.

Run | New Tab | JSON

33 select * from people where surname like '% '' %'; 17ms

people

搜索结果集

* peopleid integer first_name varchar(30) * surname varchar(30) * born integer died integer * gender character(1)

	Filter	Filter	Filter	Filter	Filter	Filter
>	3217	Luigi Filippo	D'Amico	1924	2007	M
>	3218	Beverly	D'Angelo	1951	(NULL)	F
>	3219	James	D'Arcy	1975	(NULL)	M
>	3220	Ileana	D'Cruz	1987	(NULL)	F
>	3221	Vincent	D'Onofrio	1959	(NULL)	M
>	3222	Genelia	D'Souza	1987	(NULL)	F
>	3449	Jewel	De'Nyle	1976	(NULL)	F
>	8070	Marcel	L'Herbier	1888	1979	M
>	8110	Jay	Laga'aia	1963	(NULL)	M

11.

▷ Run | New Tab | JSON

35 select * from countries where continent = 'EUROPE' and country_code like 'c%' 2ms

countries ×

搜索结果集



耗时: 4ms

1

Top



* country_code
character(2)

* country_name
varchar(50)

* continent
varchar(20)

Filter

Filter

Filter

>

cy

>

Cyprus

>

EUROPE

>

cz

>

Czech Republic

>

EUROPE

>

ch

>

Switzerland

>

EUROPE

12.

37 select * from people where surname like substr(first_name,1,1) || '%'; 6ms

people

搜索结果集

耗时: 8ms 1 2 3 4 ...

	* peopleid integer	first_name varchar(30)	* surname varchar(30)	* born integer	died integer	* gender character(1)
	Filter	Filter	Filter	Filter	Filter	Filter
>	11	Alex	Abbad	1978	(NULL)	M
>	21	Aleksandr	Abdulov	1953	2008	M
>	26	Alfred	Abel	1879	1937	M
>	32	Ary	Abittan	1974	(NULL)	M
>	34	Alon	Aboutboul	1965	(NULL)	M
>	42	Andrei	Abrikosov	1906	1973	M
>	52	Amy	Acker	1976	(NULL)	F
>	56	Alfred	Adam	1908	1982	M
>	58	Amy	Adams	1974	(NULL)	F

13.

A screenshot of a database interface, likely DBeaver, showing a query result. The top bar has a 'Run | New Tab' button and a status message '20ms'. Below it, a query window shows the SQL command: 'select min(2024-born) from people;'. The result pane displays a single row with the value '20ms'. The left sidebar shows a tree view with a 'people' node selected. The bottom pane shows a 'Filter' table with one row containing '> 15'.

	Filter
>	15

14.

▷ Run | New Tab | JSON

```
83 select country, count(movieid) as movie
84 from movies
85 where country like 'm%'
86 group by country; 9ms
```

movies

搜索结果集

* country character(2) movie TRUE

	Filter	Filter
>	mx	59
>	ma	2
>	mn	1
>	ml	2
>	my	5

15.

88

```
select count(DISTINCT country) from movies;| 12ms
```

The screenshot shows a database interface with a dark background featuring a green and purple abstract design. At the top, there's a header bar with a magnifying glass icon, the text '搜索结果集' (Search Results), and several other icons including a gear, envelope, and a red circle with the number '1'. Below the header is a table with the following data:

	count
Filter	
>	85

16.

```
89 ▶ Run | New Tab | JSON  
90 select min(year_released) from movies  
91 where country in ('cn', 'tw', 'hk'); 9ms
```

movies

搜索结果集

	Filter
>	1913

17.

Run | New Tab | JSON

```
93 select count(movieid) from movies  
94 where year_released =2010; 19ms
```

The screenshot shows a database interface with a dark theme. At the top, there's a toolbar with icons for Run, New Tab, and JSON. Below the toolbar, a code editor window displays a SQL query and its execution results. The results show a single row with the count of movies released in 2010.

count
198

18.

```
Run | New Tab | JSON  
96 select year_released, count(*) cnt from movies  
97      where year_released >= 1960  
98      group by year_released;| 19ms
```

movies ×

搜索结果集 S

* year_released
integer

cnt TRUE

	Filter	Filter
>	1964	104
>	1969	87
>	2008	235
>	1991	106
>	1989	80
>	1974	76
>	1977	63

19.

Run | New Tab | JSON

100 select count(movieid) from movies
101 where country='gb' and year_released=1949; 9ms

movies

搜索结果集 S

count

	Filter
>	11

20.

```
    ▷ Run | New Tab | JSON | Copy
75  select dire, count(movieid) films from
76    (select movieid, count(*) dire from credits
77      where credited_as='D'
78        group by movieid) dir_cnt
79  group by dire; 24ms
80
```

Result(RO) ×



搜索结果集



Q

dire



films



TRUE

	Filter	Filter
>	3	38
>	5	7
>	4	13
>	10	1
>	6	8
>	2	361
>	1	7699

21.

▷ Run | New Tab | JSON | Copy

```
103 select
104     count(*) as total_people,
105     count(case when died is null then 1 end) as alive,
106     count(case when died is not null then 1 end) as died
107 from people; 9ms
```

people

搜索结果集

	total_people	alive	died
>	16489	10836	5653

22.

Run | New Tab | JSON

```
✓ 109 select max(cnt) from
110   (select count(*)cnt from people
111    GROUP BY surname) x; 12ms
```

Result(RO) ×

搜索结果集 S

max

Filter

> 75

23.

Run | New Tab | JSON

```
✓ 113 select count(distinct peopleid) from people
114 where peopleid in (select peopleid from credits where credited_as='A')
115 and peopleid in (select peopleid from credits where credited_as='D')| 4
```

Result(RO) X



搜索结果集



导出



耗时: 45



count



Filter

> 467

24.

```
Run | New Tab | JSON | Copy  
✓ 133 select ROUND((cast(cnt_woman as NUMERIC) / cast(cnt as NUMERIC) ),1) as percentage_women  
134 from  
135     (select count(*) as cnt from people) x  
136     cross join  
137     (select count(*) as cnt_woman from people where gender = 'F') y; 35ms
```

Result(RO) ×

搜索结果集 S

1

导出

耗时: 35ms

percentage_women

Filter > 0.3

25.

```
▷ Run | New Tab | JSON  
139 select country,count(*) as cnt from movies  
140 where runtime >=180  
141 group by country; 8ms
```

movies

搜索结果集

* country character(2) cnt TRUE

	Filter	Filter
>	nz	3
>	ru	9
>	cn	2
>	sp	1
>	cz	1
>	ro	1
>	gb	6