

Regexp Vagy Like, melyiket szeressem

fajlok: m_like.php; m_regexp.php; mi_like.php; mi_regexp.php

A SQL Regexp és Like elemeit fogom, némileg tesztelni, de nem találati szempont alapján, most inkább az érdekel melyik gyorsabb. A tesztelés során ugyanazokat a kéréseket fogom futtatni mysql_ ; mysql_i ; Adodb Mysqli; Mysql; Pdo Sqlite és Mysqli adapterek és függvények segítségével. A tesztek 10-es körben 10.000-szer fognak lefutni, vagyis egy teljes kör 100,000-szer fogja végrehajtani a kéréseket. Ebből csak a minimum és a maximum értékeket veszem alapul.

A tesztek Hardver erőforrás és beállítás függőek. Elképzelhető hogy egy jól konfigurált szervergépen, más eredményt produkálna. Ez egy Ubuntu operációs rendszerre telepített LAMP szerveren lett tesztelve (2 x 1.6 proci 2GB Ram).

A tesztben szereplő kérések:

```
$query=(string)"SELECT * FROM test WHERE nev LIKE '%al%'";  
$query=(string)"SELECT * FROM `test` WHERE `nev` REGEXP '(al|ál)'";
```

A konkrét példában (amiben a Like '%al%' kifejezés volt) a Like 22 találatot hoz fel, míg a Regexp 14-et. A különbség oka, hogy a LIKE az ál szócskát is megeszi, míg a Regexp ragaszkodik az al szócskához. Ez rögtön egy fontos eldöntendő kérdés, melyiket is válasszuk. Itt az ember hajlamos a többre szavazni, mert ha legyen már talát a kérésre, de mondjuk egy keresőnél ez már akár hátrány is lehet. pl.: keres **alkat** szóra és a találatnál látjuk az **álmogyár** szócskát.

Ahhoz, hogy azonos számú elemmel dolgozzunk a REGEXP kérést módosítjuk (al|ál)-ra.

Különbségek felderítés:

```
SELECT * FROM `test` WHERE `nev` LIKE '%al%' AND `nev` NOT IN(SELECT `nev`  
FROM `test` WHERE `nev` REGEXP 'al')
```

MySQL Like:

```
$query=(string)"SELECT * FROM test WHERE nev LIKE '%al%'";  
for($j=0;$j<$c1;$j++){  
    $s=microtime(true);  
    for($i=0;$i<$c2;$i++){  
        $result=mysql_query($query);  
    }  
    $e=microtime(true);  
    $eredmeny[$j]=$e-$s;  
}
```

MySQL Regexp:

```
$query=(string)"SELECT * FROM `test` WHERE `nev` REGEXP '(al|ál)'";  
for($j=0;$j<$c1;$j++){  
    $s=microtime(true);  
    for($i=0;$i<$c2;$i++){  
        $result=mysql_query($query);  
    }  
}
```

```

    $e=microtime(true);
    $eredmeny[$j]=$e-$s;
}

```

Szárazanyag tartalom:

Min= a teszt során a legkisebb érték; **Max**=A legnagyobb érték amit kidobott a gép.

1, Like Használatával:

MySQL Like '%al%'	Min: 0.531497955322	Max: 0.63414812088
MySQLi Like '%al%'	Min: 0.567688941956	Max: 0.66382598877
PDO MySQLi Like '%al%'	Min: 0.610804796219	Max: 0.70207285881
AdoDB MySQLi Like	Min: 1.7728228569	Max: 2.06709098816
AdoDB MySQL Like	Min: 1.70053792	Max: 1.71485304832
PDO SQLite Like	Min: 0.807989120483	Max: 0.849720954895
AdoDB SQLite Like	Min: 0.750332832336	Max: 0.898746013641

2, Regexp használatával:

MySQL Regexp '(al ál)'	Min: 0.567535877228	Max: 0.618416070938
MySQLi Regexp '(al ál)'	Min: 0.570158004761	Max: 0.631108045578
PDO MySQLi Regexp	Min: 0.636610031128	Max: 0.699748039246
AdoDB MySQLi Regexp	Min: 1.7728228569	Max: 2.06709098816
AdoDB MySQL Regexp	Min: 1.73313379288	Max: 2.01311588287
PDO SQLite Regexp	Min: 0.309896945953	Max: 0.353369951248
AdoDB SQLite Regexp	Min: 0.728609085083	Max: 0.736239910126

Első körben döbbenet, mert az AdoDB-től sokkal többet vártam (ezek az értékek elfogadhatatlanul magasak). A MySQL kiterjesztésű függvények gyorsabbnak bizonyulnak mint a MySQLi függvények. Nem ítélnék elhamarkodottan, mert ez egy egyszerű teszt viszont ugyanaz a kérés (csoport) futott le mindegyiken, és csak a válaszidő volt a lényeg. A PDO SQLite függvények rendkívül gyorsak, a test adatbázis SQLite3 volt.

- A közepes méretű, komolyabb projekteknél a PDO - SQLite használata erősen javasolt. Közepes vagy kisebb méretű projektekhez nem is szabadna SQLite-on kívül mást használni.
- AdoDB nagy kedvencem, elég gyalázatos értékeket produkált, nagyobb terhelésű oldalaknál megfontolandó a használata. Ez volna a kényelem bére?
- MySQL_ vagy MySQLi kiterjesztésű függvények? A teszt alapján gyorsabb a mysql_ viszont a mysqli_ fejlettebb, és a különbségek igazán nem nagyok. Így nincs jelentősége annak, melyiket használja az ember.
- Túlzott Query optimalizálás: Csak a Gyártó által javasoltak, hozhatnak eredményt. A szintaktikai kérdés nincs túlzottan nagy hatással a válasz eredményekre.
- Regexp vagy Like?
Véleményem szerint Regexp mert sokkal több eszköz van a kezünkben és nem lassabb. A megértése viszont nem kis csapás :)

Egy rossz SQL adapter és adatbázis háttér választásnak köszönhetően, ha szerencsések vagyunk az erőforrások nagy részét el is kalapáljuk.