

Príklady SQL 1

Pokračovanie [Aplikácia evidencie DVD](#)

Výber názvu seriálu pre zvolený 1 film

```
SELECT cs.serial,f.id_film, f.name, c.genre, f.length, cl.lang FROM film f,cis_genre c,langs l,cis_lang cl, serials s,cis_serial cs WHERE (f.id_genre=1) AND (f.id_genre=c.id_genre) AND (f.id_film=l.id_film) AND (l.id_lang=cl.id_lang) AND (f.id_film=s.id_film) AND (s.id_serial=cs.id_serial) AND (f.id_film=1)
```

serial	id_film	name	genre	length	lang
Dvanáct měsíčků	1	Dvanáct měsíčků 1.DVD	rozprávka	62	česky
Dvanáct měsíčků	1	Dvanáct měsíčků 1.DVD	rozprávka	62	rusky

Výber názvu seriálu pre všetky filmy daného seriálu

```
SELECT cs.serial,f.id_film, f.name, c.genre, f.length, cl.lang FROM film f,cis_genre c,langs l,cis_lang cl, serials s,cis_serial cs WHERE (s.id_serial=1) AND (f.id_genre=c.id_genre) AND (f.id_film=l.id_film) AND (l.id_lang=cl.id_lang) AND (f.id_film=s.id_film) AND (s.id_serial=cs.id_serial)
```

serial	id_film	name	genre	length	lang
Dvanáct měsíčků	1	Dvanáct měsíčků 1.DVD	rozprávka	62	česky
Dvanáct měsíčků	1	Dvanáct měsíčků 1.DVD	rozprávka	62	rusky
Dvanáct měsíčků	5	Dvanáct měsíčků 2.DVD	rozprávka	76	česky
Dvanáct měsíčků	5	Dvanáct měsíčků 2.DVD	rozprávka	76	rusky

Operátor LIKE

Využitie LIKE. Operátor LIKE slúži na porovnávanie reťazcov. Používa sa namiesto relačných operátorov v klauzule WHERE pre výbery zo stĺpcov typu text alebo reťazec (typy char, varchar, text atď.). Poskytuje aj istú možnosť akýchsi jednoduchých regulárnych výrazov. LIKE 'K%' znamená reťazec, ktorý začína písmenom 'K'. Znak '%' nahrádza ľubovoľný počet znakov, aj nulový počet. Môže byť umiestnený pred aj za hľadaným znakom alebo podreťazcom. Príklad LIKE 'L____' je zápis reťazca, ktorý začína písmenom 'L' a má práve 5 znakov - znak '_' nahrádza jeden ľubovoľný znak. Príklad LIKE '_m%n' je kombináciou dvoch predchádzajúcich prípadov. Udáva, že reťazec obsahuje na druhej pozícii písmeno 'm', nasledované ľubovoľným počtom znakov, ktoré sú ukončené písmenom 'n'.

Výber z jedného znakového stĺpca (name)

```
SELECT * FROM `film` WHERE (`name` LIKE '%va%')
```

id_film	name	name_orig	year	length	id_genre	content	bonusy
1	Dvanáct měsíčků 1.DVD	Dvenadcat mesjacev	1972	62	1	Pohádka	
2	O statočnom kováčovi	O statečném kováři	1983	80	1	Neohrožený Mikeš	Filmografia, fotogaléria
5	Dvanáct měsíčků 2.DVD	Dvenadcat mesjacev	1972	76	1	Pohádka	

Výber z viacerých znakových stĺpcov súčasne (operácia OR) dáva rovnaký výsledok ako predchádzajúci príklad:

```
SELECT * FROM `film` WHERE (`name` LIKE '%va%') OR (`name_orig` LIKE '%va%') OR (`content` LIKE '%va%')
```

Malá zmena výberu

```
SELECT * FROM `film` WHERE (`name` LIKE '%va%') OR (`name_orig` LIKE '%ve%') OR (`content` LIKE '%va%')
```

id_film	name	name_orig	year	length	id_genre	content
1	Dvanáct měsíčků 1.DVD	Dvenadcat mesjacev	1972	62	1	Pohádka
2	O statočnom kováčovi	O statečném kováři	1983	80	1	Neohrožený Mikeš
3	Da Vinciho kód	Da Vinci The Code He Lived by	2005	92	3	
5	Dvanáct měsíčků 2.DVD	Dvenadcat mesjacev	1972	76	1	Pohádka

To isté usporiadané podľa názvu

```
SELECT * FROM `film` WHERE (`name` LIKE '%va%') OR (`name_orig` LIKE '%ve%') OR (`content` LIKE '%va%') ORDER BY name
```

id_film	name ▲	name_orig	year	length	id_genre	content
3	Da Vinciho kód	Da Vinci The Code He Lived by	2005	92	3	
1	Dvanáct měsíčků 1.DVD	Dvenadcat mesjacev	1972	62	1	Pohádka
5	Dvanáct měsíčků 2.DVD	Dvenadcat mesjacev	1972	76	1	Pohádka
2	O statočnom kováčovi	O statečném kováři	1983	80	1	Neohrožený Mikeš

To isté usporiadané podľa názvu v opačnom poradí

```
SELECT * FROM `film` WHERE (`name` LIKE '%va%') OR (`name_orig` LIKE '%ve%') OR (`content` LIKE '%va%') ORDER BY name DESC
```

id_film	name ▼	name_orig	year	length	id_genre	content
2	O statočnom kováčovi	O statečném kováři	1983	80	1	Neohrožený Mikeš
5	Dvanáct měsíčků 2.DVD	Dvenadcat mesjacev	1972	76	1	Pohádka
1	Dvanáct měsíčků 1.DVD	Dvenadcat mesjacev	1972	62	1	Pohádka
3	Da Vinciho kód	Da Vinci The Code He Lived by	2005	92	3	

Všetky uvedené ukážky a spôsoby je možné podľa potreby ľubovoľne kombinovať tak, aby sme dosiahli splnenie cieľov aplikácie.

Operátor ORDER BY je uvádzaný ako posledná klauzula príkazu SELECT a môže obsahovať aj viac stĺpcov, napr.:

```
SELECT * FROM `film` WHERE (`name` LIKE '%va%') OR (`name_orig` LIKE '%ve%') OR (`content` LIKE '%va%') ORDER BY `year`,`length`
```

id_film	name	name_orig	year	length	id_genre	content
1	Dvanáct měsíčků 1.DVD	Dvenadcat mesjacev	1972	62	1	Pohádka
5	Dvanáct měsíčků 2.DVD	Dvenadcat mesjacev	1972	76	1	Pohádka
2	O statočnom kováčovi	O statečném kováři	1983	80	1	Neohrožený Mikeš
3	Da Vinciho kód	Da Vinci The Code He Lived by	2005	92	3	

Výpis daných položiek celej tabuľky film, zrušenie riadku 5 a oprava roku a dĺžky v riadku 4:

```
SELECT id_film, name, year, length FROM film
```

```
DELETE FROM film WHERE film.id_film = 5
```

```
UPDATE film SET year=2002, length=145 WHERE id_film=4
```

id_film	name	year	length
1	Dvanáct měsíčků 1.DVD	1972	62
2	O statočnom kováčovi	1983	80
3	Da Vinciho kód	2005	92
4	Čistá duše	2001	130
5	Dvanáct měsíčků 2.DVD	1972	76

id_film	name	year	length
1	Dvanáct měsíčků 1.DVD	1972	62
2	O statočnom kováčovi	1983	80
3	Da Vinciho kód	2005	92
4	Čistá duše	2002	145

Operátor EXISTS

Operátor EXISTS vracia logickú hodnotu TRUE alebo FALSE podľa toho, či vnorený dotaz vracia alebo nevracia nejakú hodnotu. Používa sa preto pri vnorených dotazoch a má tvar EXISTS(vnorený SQL) alebo NOT EXISTS (vnorený SQL).

Ako príklad uveďme zoznam filmov, ktoré sú seriály (vľavo):

```
SELECT id_film, name, year, length FROM `film` f WHERE
  EXISTS(
    SELECT id_serial FROM serials
    WHERE f.id_film=id_film)
```

Podobne bude zoznam filmov, ktoré nie sú seriály (vpravo):

```
SELECT id_film, name, year, length FROM `film` f WHERE
NOT EXISTS(
SELECT id_serial FROM serials
WHERE f.id_film=id_serial)
```

id_film	name	year	length
1	Dvanáct měsíčků 1.DVD	1972	62
5	Dvanáct měsíčků 2.DVD	1972	76

id_film	name	year	length
2	O statočnom kováčovi	1983	80
3	Da Vinciho kód	2005	92
4	Čistá duše	2001	130

Výber filmov, ktoré sú seriály, doplnené názvom seriálu:

```
SELECT cs.serial,f.id_film, f.name, f.year, f.length FROM `film` f, serials s,cs_serial
cs WHERE
EXISTS(
SELECT id_serial FROM serials
WHERE f.id_film=id_serial
)
AND (f.id_film=s.id_film) AND (s.id_serial=cs.id_serial)
```

serial	id_film	name	year	length
Dvanáct měsíčků	1	Dvanáct měsíčků 1.DVD	1972	62
Dvanáct měsíčků	5	Dvanáct měsíčků 2.DVD	1972	76

Výber filmov so zadaným jazykom (rusky – 4)

```
SELECT f.id_film, name,`name_orig`, year, length FROM `film` f, langs l WHERE
(f.id_film=l.id_film) AND (l.id_lang=4)
```

id_film	name	name_orig	year	length
1	Dvanáct měsíčků 1.DVD	Dvenadcat mesjacev	1972	62
5	Dvanáct měsíčků 2.DVD	Dvenadcat mesjacev	1972	76

Výber filmov so zadaným jazykom (anglicky – 3)

```
SELECT f.id_film, name,`name_orig`, year, length,`id_genre` FROM `film` f,langs l
WHERE (f.id_film=l.id_film) AND (l.id_lang=3)
```

id_film	name	name_orig	year	length	id_genre
3	Da Vinciho kód	Da Vinci The Code He Lived by	2005	92	3
4	Čistá duše		2001	130	5

Výber filmov so zadaným jazykom (anglicky – 3) a žánrom (dráma – 3)

```
SELECT f.id_film, name,`name_orig`, year, length,`id_genre` FROM `film` f,langs l
WHERE (f.id_film=l.id_film) AND (l.id_lang=3) AND (l.id_genre =3)
```

id_film	name	name_orig	year	length	id_genre
3	Da Vinciho kód	Da Vinci The Code He Lived by	2005	92	3

Ukážky agregáčnych funkcií

Agregačné funkcie slúžia na získanie súhrnných údajov niekoľkých riadkov. Pre danú skupinu riadkov môžeme počítať matematické operácie (min, max, sum, avg a count. Nové názvy stĺpcov môžeme definovať operátorom AS. Inak ostáva pôvodný názov stĺpca alebo agregáčnej funkcie.

Pre filmy, ktoré nie sú seriály (NOT EXISTS) určte ich počet a priemernú dĺžku:

```
SELECT count(*) AS pocet,AVG(length) AS priem_dlzka FROM `film` f WHERE
NOT EXISTS(
SELECT id_serial FROM serials
WHERE f.id_film=id_film)
```

pocet	priem_dlzka
3	100.6667

Pre filmy, ktoré sú seriály (EXISTS) určte názov seriálu, počet jeho častí a ich celkovú dĺžku:

```
SELECT cs.serial,count(f.id_film) AS počet_častí, sum(f.length) AS dĺzka_celk
FROM `film` f, serials s,cis_serial cs WHERE
(f.id_film=s.id_film) AND (s.id_serial=cs.id_serial) AND EXISTS(
SELECT id_serial FROM serials
WHERE f.id_film=id_film)
```

serial	počet_častí	dĺzka_celk
Dvanáct měsíčků	2	138

Podmienené funkcie - operátory

Sú podporované v MySQL a umožňujú vykonanie rôznych testov nad stĺpcami tabuliek. Majú tvar IF, IFNULL a CASE a s výhodou ich využívame najmä nad stĺpcami číselného typu. Operátor IF() má tri argumenty – ak je prvý vyhodnotený ako pravda, vráti druhý argument, inak tretí. Operátor IFNULL() má dva argumenty – ak je výsledkom prvého NULL, vráti druhý argument. Operátor CASE slúži na porovnanie viacerých stĺpcov. Príklad na IF vyberie väčšiu z hodnôt year,length:

```
SELECT id_film, name, year, length, IF(year>length,year,length) AS väčší FROM `film`
```

id_film	name	year	length	väčší
1	Dvanáť mesáčikov 1.DVD	1972	62	1972
2	O statočnom kováčovi	1983	80	1983
3	Da Vinciho kód	2005	92	2005
4	Čistá duše	2001	130	2001
5	Dvanáť mesáčikov 2.DVD	1972	2076	2076

Praktické využitie je napr. dosadenie textu v prípade nezadanej hodnoty – stĺpec bonusy:

```
SELECT id_film, name, year, length, IF(bonusy='', 'Nezadané', `bonusy`) AS bonusy FROM `film`
```

id_film	name	year	length	bonusy
1	Dvanáť mesáčikov 1.DVD	1972	62	Nezadané
2	O statočnom kováčovi	1983	80	Filmografia, fotogaléria
3	Da Vinciho kód	2005	92	Nezadané
4	Čistá duše	2001	130	Vynechané scény
5	Dvanáť mesáčikov 2.DVD	1972	2076	Nezadané

Operátor CASE má tvar

```
CASE value, WHEN [compare_value, ] THEN result, [WHEN [compare_value, ] THEN result, ...] [ELSE result] END  
CASE WHEN [condition, ] THEN result, [WHEN [condition, ] THEN result, ...] [ELSE result] END
```

Prvá verzia vracia result pre value=compare_value. Druhá verzia vracia result pre prvú splnenú podmienku. Ak sa nenašiel vyhovujúci výsledok (podmienka), vracia result po ELSE, alebo NULL, ak časť ELSE nie je uvedená. Príklad použitia CASE v modifikácii ukážky IF (zníži rok o 100, resp. vynásobí dĺžku dvoma):

```
SELECT id_film, name, year, length, CASE WHEN year>length THEN year-100  
ELSE length*2 END AS väčšíCASE FROM `film`
```

id_film	name	year	length	většíCASE
1	Dvanáct měsíčků 1.DVD	1972	62	1872
2	O statočnom kováčovi	1983	80	1883
3	Da Vinciho kód	2005	92	1905
4	Čistá duše	2001	130	1901
5	Dvanáct měsíčků 2.DVD	1972	2076	4152

Iné zdroje:

Skřivan, Jaromír. 2000. *Databáze a jazyk SQL*. [online] Interval.cz, 2000. dostupné z <http://interval.cz/clanky/databaze-a-jazyk-sql/> >.

Skřivan, Jaromír. 2000. *SQL - dotazy s agregací*. [online] Interval.cz, 2000. dostupné z <http://interval.cz/clanky/sql-dotazy-s-agregaci/> >.

Skřivan, Jaromír. 2000. *SQL - jak na dotazy 1*. [online] Interval.cz, 2000. dostupné z <http://interval.cz/clanky/sql-jak-na-dotazy-1/> >.

Skřivan, Jaromír. 2000. *SQL - jak na dotazy 2*. [online] Interval.cz, 2000. dostupné z <http://interval.cz/clanky/sql-jak-na-dotazy-2/> >.

Skřivan, Jaromír. 2001. *SQL - jak na funkce*. [online] Interval.cz, 2000. dostupné z <http://interval.cz/clanky/sql-jak-na-funkce/> >.

Skřivan, Jaromír. 2000. *SQL - jak na trigger*. [online] Interval.cz, 2000. dostupné z <http://interval.cz/clanky/sql-jak-na-trigger/> >.

Skřivan, Jaromír. 2000. *SQL - manipulace s tabulkami*. [online] Interval.cz, 2000. dostupné z <http://interval.cz/clanky/sql-manipulace-s-tabulkami/> >.

Skřivan, Jaromír. 2000. *SQL - pokročilá práce s tabulkami a daty*. [online] Interval.cz, 2000. dostupné z <http://interval.cz/clanky/sql-pokrocila-prace-s-tabulkami-a-daty/> >.

Skřivan, Jaromír. 2000. *SQL - skládání dotazů*. [online] Interval.cz, 2000. dostupné z <http://interval.cz/clanky/sql-skladani-dotazu/> >.

Skřivan, Jaromír. 2000. *SQL - spojování tabulek a tvorba pohledů*. [online] Interval.cz, 2000. dostupné z <http://interval.cz/clanky/sql-spojovani-tabulek-a-tvorba-pohledu/> >.

Skřivan, Jaromír. 2000. *SQL - tvorba tabulek*. [online] Interval.cz, 2000. dostupné z <http://interval.cz/clanky/sql-tvorba-tabulek/> >.

Skřivan, Jaromír. 2000. *SQL - vkládání a aktualizace dat v tabulce*. [online] Interval.cz, 2000. dostupné z <http://interval.cz/clanky/sql-vkladani-a-aktualizace-dat-v-tabulce/> >.

Skřivan, Jaromír. 2000. *SQL - vnořené dotazy.* [online] Interval.cz, 2000. dostupné z <http://interval.cz/clanky/sql-vnorene-dotazy/> >.