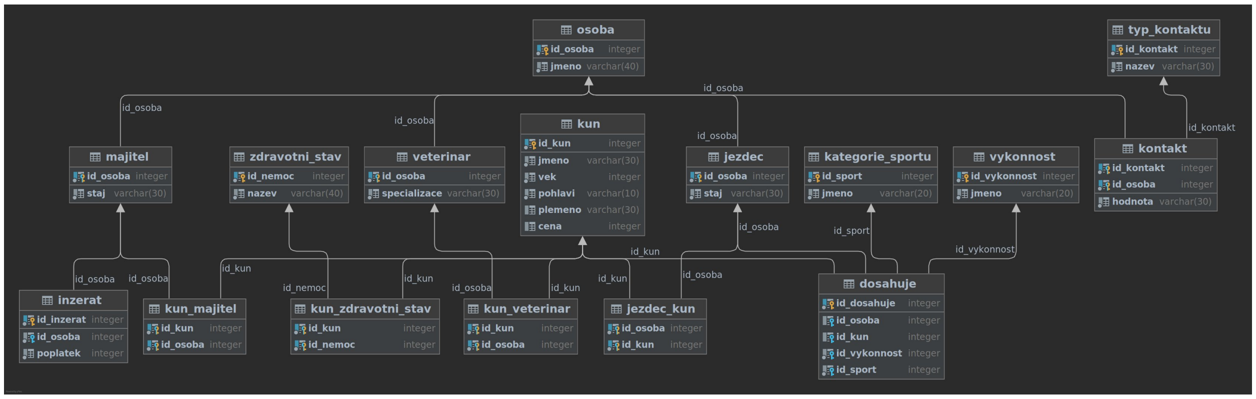
Sql optimalizace

Popis databáze



Dotaz kategorie A

Zadání: Jezdec, který jezdí koně jménem Amur nebo koně jménem Tessie, ale nemá tyto koně na ježdění zároveň.

První provedení:

```
Select jmeno
from
( ( (select id_osoba
from jezdec_kun
join kun using (id_kun)
where jmeno ='Tessie')
union
(select id_osoba
from jezdec_kun
join kun using (id_kun)
where jmeno='Amur') )
minus
( (select id_osoba
from jezdec_kun
join kun using (id_kun)
where jmeno ='Tessie')
intersect
(select id_osoba
from jezdec_kun
join kun using (id_kun)
where jmeno='Amur') ) ) result
join osoba using (id_osoba)
```

```
----------------------------------------------------------------------------------------------------
| Id  | Operation                          | Name          | Rows  | Bytes | Cost (%CPU)| Time     |
----------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                   |               |     1 |    48 |     8  (50)| 00:00:01 |
|   1 |  NESTED LOOPS                      |               |     1 |    48 |     8  (50)| 00:00:01 |
|   2 |   NESTED LOOPS                   |              |     2 |    48 |     8  (50)| 00:00:01 |
|   3 |    VIEW                          |              |     2 |    26 |     8  (50)| 00:00:01 |
|   4 |     MINUS                        |              |       |       |            |          |
|   5 |      SORT UNIQUE                 |              |     2 |   224 |     8  (50)| 00:00:01 |
|   6 |       UNION-ALL                  |              |       |       |            |          |
|   7 |        NESTED LOOPS SEMI         |              |     1 |    56 |     1   (0)| 00:00:01 |
|   8 |         INDEX FULL SCAN          | PK_JEZDEC_KUN |     1 |    26 |     1   (0)| 00:00:01 |
|*  9 |         TABLE ACCESS BY INDEX ROWID| KUN          |     1 |    30 |     0   (0)| 00:00:01 |
|* 10 |          INDEX UNIQUE SCAN       | PK_KUN       |     1 |       |     0   (0)| 00:00:01 |
|  11 |        NESTED LOOPS SEMI         |              |     1 |    56 |     1   (0)| 00:00:01 |
|  12 |         INDEX FULL SCAN          | PK_JEZDEC_KUN |     1 |    26 |     1   (0)| 00:00:01 |
|* 13 |         TABLE ACCESS BY INDEX ROWID| KUN          |     1 |    30 |     0   (0)| 00:00:01 |
|* 14 |          INDEX UNIQUE SCAN       | PK_KUN       |     1 |       |     0   (0)| 00:00:01 |
|  15 |      INTERSECTION                |              |       |       |            |          |
|  16 |       SORT UNIQUE                |              |     1 |    56 |            |          |
|  17 |        NESTED LOOPS SEMI         |              |     1 |    56 |     1   (0)| 00:00:01 |
|  18 |         INDEX FULL SCAN          | PK_JEZDEC_KUN |     1 |    26 |     1   (0)| 00:00:01 |
|* 19 |         TABLE ACCESS BY INDEX ROWID| KUN          |     1 |    30 |     0   (0)| 00:00:01 |
|* 20 |          INDEX UNIQUE SCAN       | PK_KUN       |     1 |       |     0   (0)| 00:00:01 |
|  21 |       SORT UNIQUE                |              |     1 |    56 |            |          |
|  22 |        NESTED LOOPS SEMI         |              |     1 |    56 |     1   (0)| 00:00:01 |
|  23 |         INDEX FULL SCAN          | PK_JEZDEC_KUN |     1 |    26 |     1   (0)| 00:00:01 |
|* 24 |         TABLE ACCESS BY INDEX ROWID| KUN          |     1 |    30 |     0   (0)| 00:00:01 |
|* 25 |          INDEX UNIQUE SCAN       | PK_KUN       |     1 |       |     0   (0)| 00:00:01 |
|* 26 |    INDEX UNIQUE SCAN             | PK_OSOBA     |     1 |       |     0   (0)| 00:00:01 |
|  27 |   TABLE ACCESS BY INDEX ROWID    | OSOBA        |     1 |    35 |     0   (0)| 00:00:01 |
----------------------------------------------------------------------------------------------------
```

```
"   9 - filter(""KUN"".""JMENO""='Tessie')"
"  10 - access(""JEZDEC_KUN"".""ID_KUN""=""KUN"".""ID_KUN"")"
"  13 - filter(""KUN"".""JMENO""='Amur')"
"  14 - access(""JEZDEC_KUN"".""ID_KUN""=""KUN"".""ID_KUN"")"
"  19 - filter(""KUN"".""JMENO""='Tessie')"
"  20 - access(""JEZDEC_KUN"".""ID_KUN""=""KUN"".""ID_KUN"")"
"  24 - filter(""KUN"".""JMENO""='Amur')"
"  25 - access(""JEZDEC_KUN"".""ID_KUN""=""KUN"".""ID_KUN"")"
"  26 - access(""RESULT"".""ID_OSOBA""=""OSOBA"".""ID_OSOBA"")"
```

Diskuze:

Z prováděcího plánu je zřejmé že operace minus, union mnoho vnořených selectů nám způsobují ne zcela efektivní dotaz. Pojďme to spravit..

Druhé provedení:

```
SELECT o.jmeno
FROM osoba o
JOIN jezdec_kun jk ON o.id_osoba = jk.id_osoba
JOIN kun k ON jk.id_kun = k.id_kun
WHERE k.jmeno IN ('Tessie', 'Amur')
GROUP BY o.jmeno
HAVING COUNT(DISTINCT CASE WHEN k.jmeno = 'Tessie' THEN k.id_kun END) = 1
   AND COUNT(DISTINCT CASE WHEN k.jmeno = 'Amur' THEN k.id_kun END) = 0 OR
    COUNT(DISTINCT CASE WHEN k.jmeno = 'Tessie' THEN k.id_kun END) = 0
   AND COUNT(DISTINCT CASE WHEN k.jmeno = 'Amur' THEN k.id_kun END) = 1;
```

```
-----------------------------------------------------------------------------------------------
| Id  | Operation                        | Name          | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                 |               |     1 |    91 |     2  (50)| 00:00:01 |
|*  1 |  FILTER                          |               |       |       |            |          |
|   2 |   SORT GROUP BY                  |               |     1 |    91 |     2  (50)| 00:00:01 |
|   3 |    NESTED LOOPS                  |               |     1 |    91 |     1   (0)| 00:00:01 |
|   4 |     NESTED LOOPS                 |               |     1 |    91 |     1   (0)| 00:00:01 |
|   5 |      NESTED LOOPS                |               |     1 |    61 |     1   (0)| 00:00:01 |
|   6 |       INDEX FULL SCAN            | PK_JEZDEC_KUN |     1 |    26 |     1   (0)| 00:00:01 |
|   7 |       TABLE ACCESS BY INDEX ROWID| OSOBA         |     1 |    35 |     0   (0)| 00:00:01 |
|*  8 |        INDEX UNIQUE SCAN         | PK_OSOBA      |     1 |       |     0   (0)| 00:00:01 |
|*  9 |      INDEX UNIQUE SCAN           | PK_KUN        |     1 |       |     0   (0)| 00:00:01 |
|* 10 |     TABLE ACCESS BY INDEX ROWID  | KUN           |     1 |    30 |     0   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------------
```

```
"   1 - filter(COUNT(DISTINCT CASE ""K"".""JMENO"" WHEN 'Tessie' THEN ""K"".""ID_KUN"" END )=1 AND "
"              COUNT(DISTINCT CASE ""K"".""JMENO"" WHEN 'Amur' THEN ""K"".""ID_KUN"" END )=0 OR COUNT(DISTINCT "
"              CASE ""K"".""JMENO"" WHEN 'Tessie' THEN ""K"".""ID_KUN"" END )=0 AND COUNT(DISTINCT CASE "
"              ""K"".""JMENO"" WHEN 'Amur' THEN ""K"".""ID_KUN"" END )=1)"
"   8 - access(""O"".""ID_OSOBA""=""JK"".""ID_OSOBA"")"
"   9 - access(""JK"".""ID_KUN""=""K"".""ID_KUN"")"
"  10 - filter(""K"".""JMENO""='Amur' OR ""K"".""JMENO""='Tessie')"
```

Závěr:

Po zbaveni se drahe operace minus a unionu a zkonstrovani dotazu pomoci jednoho selectu a joinu s having by klauzolí se nam podarilo snizit CPU time.

Dotaz kategorie C

Zadání: Vyber všechny koně které ošetřuje jenom veterinář, jehož id je 13 a zaroveň je neošetřuje nikdo jiný

První provedení:

```
(
select k.jmeno, k.vek, k.pohlavi
from kun k join kun_veterinar vet on k.id_kun = vet.id_kun
where
vet.id_osoba = 13
)
minus
(
select k.jmeno, k.vek, k.pohlavi
from kun k join kun_veterinar vet on k.id_kun = vet.id_kun
where
vet.id_osoba != 13
)
```

```
------------------------------------------------------------------------------------------------
| Id  | Operation                      | Name             | Rows  | Bytes | Cost (%CPU)| Time     |
------------------------------------------------------------------------------------------------
```

```
|   0 | SELECT STATEMENT               |                  |     9 |  2508 |     4  (50)| 00:00:01 |
|   1 |  MINUS                         |                  |       |       |            |          |
|   2 |   SORT UNIQUE                  |                  |     9 |   684 |     2  (50)| 00:00:01 |
|   3 |    NESTED LOOPS                |                  |     9 |   684 |     1   (0)| 00:00:01 |
|   4 |     NESTED LOOPS               |                  |     9 |   684 |     1   (0)| 00:00:01 |
|*  5 |      INDEX FULL SCAN           | PK_KUN_VETERINAR |     9 |   234 |     1   (0)| 00:00:01 |
|*  6 |      INDEX UNIQUE SCAN         | PK_KUN           |     1 |       |     0   (0)| 00:00:01 |
|   7 |     TABLE ACCESS BY INDEX ROWID| KUN              |     1 |    50 |     0   (0)| 00:00:01 |
|   8 |   SORT UNIQUE                  |                  |    24 |  1824 |     2  (50)| 00:00:01 |
|   9 |    NESTED LOOPS                |                  |    24 |  1824 |     1   (0)| 00:00:01 |
|  10 |     NESTED LOOPS               |                  |    24 |  1824 |     1   (0)| 00:00:01 |
|* 11 |      INDEX FULL SCAN           | PK_KUN_VETERINAR |    24 |   624 |     1   (0)| 00:00:01 |
|* 12 |      INDEX UNIQUE SCAN         | PK_KUN           |     1 |       |     0   (0)| 00:00:01 |
|  13 |     TABLE ACCESS BY INDEX ROWID| KUN              |     1 |    50 |     0   (0)| 00:00:01 |
---------------------------------------------------------------------------------------------------
```

```
"   5 - access(""VET"".""ID_OSOBA""=13)"
"       filter(""VET"".""ID_OSOBA""=13)"
"   6 - access(""K"".""ID_KUN""=""VET"".""ID_KUN"")"
"  11 - filter(""VET"".""ID_OSOBA""<>13)"
"  12 - access(""K"".""ID_KUN""=""VET"".""ID_KUN"")"
```

Diskuze:

Z prováděcího plánu je zřejmé že operace minus není zcela efektivní a nejspíže by šla nahradit nejakym joinem v kombinaci s ORDER BY klauzoli, pojďme na to s chutí…

Druhé provedení:

```
SELECT k.jmeno, k.vek, k.pohlavi
FROM kun k
JOIN (
    SELECT id_kun, COUNT(ID_OSOBA) AS vet_count
    FROM kun_veterinar
    GROUP BY id_kun
) kvc ON k.id_kun = kvc.id_kun
JOIN kun_veterinar kv ON kvc.id_kun = kv.id_kun AND kv.id_osoba = 13 and vet_count = 1
```

```
-------------------------------------------------------------------------------------------------
| Id  | Operation                    | Name             | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |                  |    10 |   890 |     1   (0)| 00:00:01 |
|   1 |  NESTED LOOPS                |                  |    10 |   890 |     1   (0)| 00:00:01 |
|   2 |   NESTED LOOPS               |                  |    10 |   890 |     1   (0)| 00:00:01 |
|   3 |    NESTED LOOPS              |                  |    10 |   390 |     1   (0)| 00:00:01 |
|   4 |     VIEW                     |                  |    33 |   429 |     1   (0)| 00:00:01 |
|*  5 |      FILTER                  |                  |       |       |            |          |
|   6 |       HASH GROUP BY          |                  |    33 |   429 |     1   (0)| 00:00:01 |
|   7 |        INDEX FULL SCAN       | PK_KUN_VETERINAR |    33 |   429 |     1   (0)| 00:00:01 |
|*  8 |     INDEX UNIQUE SCAN        | PK_KUN_VETERINAR |     1 |    26 |     0   (0)| 00:00:01 |
|*  9 |    INDEX UNIQUE SCAN         | PK_KUN           |     1 |       |     0   (0)| 00:00:01 |
|  10 |   TABLE ACCESS BY INDEX ROWID| KUN              |     1 |    50 |     0   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------------
```

```
5 - filter(COUNT(*)=1)
"   8 - access(""KVC"".""ID_KUN""=""KV"".""ID_KUN"" AND ""KV"".""ID_OSOBA""=13)"
"   9 - access(""K"".""ID_KUN""=""KVC"".""ID_KUN"")"
```

Závěr:

Po zbaveni se drahe operace minus a zkonstrovani dotazu pomoci jednoho selectu a joinu s klauzoli having by se nam podarilo snizit CPU time.

Dotaz kategorie D

Zadání: Jezdec, co jezdí všechny koně plemena Starokladrubský kůň

První provedení:

```
SELECT r.JMENO
FROM OSOBA r
WHERE NOT EXISTS (
  SELECT *
  FROM kun k
  WHERE k.plemeno = 'Starokladrubský kůň'
  AND NOT EXISTS (
    SELECT *
    FROM jezdec_kun jk
    WHERE jk.id_kun = k.id_kun
    AND jk.id_osoba = r.id_osoba
  )
);
```

```
--------------------------------------------------------------------------------------
| Id  | Operation           | Name          | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT    |               |    20 |   700 |    11   (0)| 00:00:01 |
|*  1 |  FILTER             |               |       |       |            |          |
|   2 |   TABLE ACCESS FULL | OSOBA         |    32 |  1120 |     3   (0)| 00:00:01 |
|*  3 |   FILTER            |               |       |       |            |          |
|*  4 |    TABLE ACCESS FULL| KUN           |     2 |    60 |     3   (0)| 00:00:01 |
|*  5 |    INDEX UNIQUE SCAN| PK_JEZDEC_KUN |     1 |    26 |     0   (0)| 00:00:01 |
--------------------------------------------------------------------------------------
```

```
"   1 - filter( NOT EXISTS (SELECT 0 FROM ""KUN"" ""K"" WHERE  NOT EXISTS (SELECT "
"              0 FROM ""JEZDEC_KUN"" ""JK"" WHERE ""JK"".""ID_KUN""=:B1 AND ""JK"".""ID_OSOBA""=:B2) "
"              AND ""K"".""PLEMENO""='Starokladrubský kůň'))"
"   3 - filter( NOT EXISTS (SELECT 0 FROM ""JEZDEC_KUN"" ""JK"" WHERE "
"              ""JK"".""ID_KUN""=:B1 AND ""JK"".""ID_OSOBA""=:B2))"
"   4 - filter(""K"".""PLEMENO""='Starokladrubský kůň')"
"   5 - access(""JK"".""ID_OSOBA""=:B1 AND ""JK"".""ID_KUN""=:B2)"
```

Druhé provedení:

```
select jmeno from osoba j
where
(select count (distinct id_kun)
from jezdec_kun k
join kun using (id_kun)
where (j.id_osoba = k.id_osoba)
and plemeno = 'Starokladrubský kůň' )
=
(
(select count (distinct id_kun)
from jezdec_kun k
join kun using (id_kun)
where plemeno = 'Starokladrubský kůň' )
)
```

```
-------------------------------------------------------------------------------------------------
| Id  | Operation                        | Name          | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                 |               |    32 |  1120 |     4   (0)| 00:00:01 |
|*  1 |  FILTER                          |               |       |       |            |          |
|   2 |   TABLE ACCESS FULL              | OSOBA         |    32 |  1120 |     3   (0)| 00:00:01 |
|   3 |   SORT GROUP BY                  |               |     1 |    56 |            |          |
|   4 |    NESTED LOOPS SEMI             |               |     1 |    56 |     1   (0)| 00:00:01 |
|*  5 |     INDEX RANGE SCAN             | PK_JEZDEC_KUN |     1 |    26 |     1   (0)| 00:00:01 |
|*  6 |     TABLE ACCESS BY INDEX ROWID  | KUN           |     2 |    60 |     0   (0)| 00:00:01 |
|*  7 |      INDEX UNIQUE SCAN           | PK_KUN        |     1 |       |     0   (0)| 00:00:01 |
|   8 |   SORT AGGREGATE                 |               |     1 |    13 |            |          |
|   9 |    VIEW                          | VM_NWVW_1     |     3 |    39 |     2  (50)| 00:00:01 |
|  10 |     SORT GROUP BY                |               |     3 |   129 |     2  (50)| 00:00:01 |
|  11 |      NESTED LOOPS SEMI           |               |     3 |   129 |     1   (0)| 00:00:01 |
|  12 |       INDEX FULL SCAN            | PK_JEZDEC_KUN |    27 |   351 |     1   (0)| 00:00:01 |
|* 13 |       TABLE ACCESS BY INDEX ROWID| KUN           |     1 |    30 |     0   (0)| 00:00:01 |
|* 14 |        INDEX UNIQUE SCAN         | PK_KUN        |     1 |       |     0   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------------
```

```
"   1 - filter( (SELECT COUNT(DISTINCT ""K"".""ID_KUN"") FROM ""KUN"" ""KUN"",""JEZDEC_KUN"" ""K"" "
"              WHERE ""K"".""ID_OSOBA""=:B1 AND ""K"".""ID_KUN""=""KUN"".""ID_KUN"" AND "
"              ""KUN"".""PLEMENO""='Starokladrubský kůň')= (SELECT COUNT(""$vm_col_1"") FROM  (SELECT "
"              ""K"".""ID_KUN"" ""$vm_col_1"" FROM ""JEZDEC_KUN"" ""K"",""KUN"" ""KUN"" WHERE "
"              ""K"".""ID_KUN""=""KUN"".""ID_KUN"" AND ""KUN"".""PLEMENO""='Starokladrubský kůň' GROUP BY "
"              ""K"".""ID_KUN"") ""VM_NWVW_1""))"
"   5 - access(""K"".""ID_OSOBA""=:B1)"
"   6 - filter(""KUN"".""PLEMENO""='Starokladrubský kůň')"
"   7 - access(""K"".""ID_KUN""=""KUN"".""ID_KUN"")"
"  13 - filter(""KUN"".""PLEMENO""='Starokladrubský kůň')"
"  14 - access(""K"".""ID_KUN""=""KUN"".""ID_KUN"")"
```

Závěr:

Obě řešení jsou si blízké z hlediska efektivity (CPU time), pokud by databáze mela více řádků tak by případ 2 byl o něco pomalejší ⇒ prohledává více řádků.