

## FULLTEXTOVÉ VYHLEDÁVÁNÍ A PROSTOROVÁ DATA PRO ZDOLÁVÁNÍ MIMOŘÁDNÝCH UDÁLOSTÍ, OCHRANU OBYVATELSTVA, HAVARIJNÍ PLÁNOVÁNÍ A KRIZOVÉ ŘÍZENÍ

Pavel ŠPULÁK<sup>1</sup>

<sup>1</sup> Ministerstvo vnitra - Generální ředitelství HZS ČR, Kloknerova 26, 148 01, Praha 414, Česká republika  
*pavel.spulak@grh.izscr.cz*

### Abstrakt

Rychlé a spolehlivé určení polohy je nezbytným předpokladem pro účinné poskytnutí pomoci při mimořádné události. Fulltextové vyhledávání umožňuje rychlé vyhledávání v textových datech a to i při omezené znalosti vstupních údajů.

Předkládaný příspěvek se proto zabývá přípravou vybraných prostorových dat, pro fulltextové vyhledávání nad jejich atributovou složkou, pro potřeby zdolávání mimořádných událostí, ochrany obyvatelstva, krizového řízení a havarijního plánování.

### Abstract

Fast and reliable determination of location is necessary prerequisite of effective provision of help during emergency event. Full-text search enable fast text data search, even if the knowledge of input parameters is limited.

Owing to this fact, this contribution is focused on the preparation of selected spatial data, for full-text search over its attribute parts, for management of emergencies, population protection, crisis management and disaster planning.

**Klíčová slova:** fulltextové vyhledávání, prostorová data, lokalizace, mimořádná událost, ochrana obyvatelstva, havarijní plánování, krizové řízení, prostorová databáze, ArcGIS Server, Oracle, Python.

**Keywords:** full-text search, spatial data, localization, emergency event, population protection, disaster planning, crisis management, spatial database, ArcGIS Server, Oracle, Python.

### ÚVOD

Rychlé a účinné poskytnutí pomoci při mimořádné události je sotva myslitelné bez přesného a rychlého určení jejího místa. Na určení místa mimořádné události dále může navazovat určení polohy objektů, které mohou ovlivnit průběh mimořádné události a její další řešení. Například to mohou být místa, kde dochází ke shromažďování osob, případně osob se sníženou schopností pohybu a orientace, nebo kde se nachází nebezpečné látky.

Pro určení místa mimořádné události je možno použít jak přirozených objektů v krajině, tak i objektů, které vytvořil člověk. Některé objekty vytvořené člověkem mohou být navíc speciálně uzpůsobeny pro určování polohy, například tak, že jsou opatřeny nejrůznějšími identifikačními štítky (přejezdy, lampy), nebo již jsou ze své podstaty opatřeny jednoznačnou identifikací (kilometrůž vodních toků, staničníky (kilometrovníky a hektometrovníky) na železničních tratích, kilometrůž pozemních komunikacích. Je možné též využít určení polohy pomocí adresy (adresní místa či jiné zájmové body jako například obchodní centra, bankomaty či čerpací stanice pohonných hmot).

Střet vlaku se silničním vozidlem na přejezdech patří bohužel mezi poměrně časté typy mimořádných událostí. Při zdolávání mimořádné události tohoto typu je potřeba řešit regulaci provozu jak na zasažené železniční trati, tak i na pozemní komunikaci. Z toho vyplývá nezbytnost přesné lokalizace místa mimořádné události.

Lokalizace železničního přejezdu se může jevit jako zdánlivě jednoduchý úkol, ale jak dokládá obrázek 1, v případě míst, kde se kříží více železničních tratí a pozemních komunikací, může být přesná lokalizace problematická. Z tohoto důvodu jsou železniční přejezdy vybaveny jednoznačnými identifikačními štítky, které umožní přesné určení místa mimořádné události.

Pochybení při přesné lokalizaci přejezdu může mít mimořádně závažné následky (1) a proto je nezbytné určit polohu přejezdu rychle a spolehlivě.



Obr. 1 Železniční přejezdy v oblasti Beroun-Závodí.

V souvislosti s určením polohy při mimořádných událostech je třeba vzít v úvahu jak přesnost a spolehlivost určení místa mimořádné události, tak i rychlost tohoto určení.

Někdy i jen velmi malé zpoždění při určování místa mimořádné události může mít mimořádně závažné následky. Může dojít k takovému rozvoji mimořádné události, kdy její zdolávání si díky zpoždění vyžádá řádově větší nasazení sil a prostředků, nebo dojde ke ztrátám na životech, zraněním, či vzniku řádově větší škody. Může též dojít ke vzniku dalších mimořádných událostí navazujících na prvotní mimořádnou událost.

Proto je třeba hledat a aplikovat nástroje, které co nejvíce urychlí a zefektivní využití nejrůznějších prostorových dat pro určení místa mimořádné události.

Prostorová data obsahují vedle údajů, které se týkají polohy, též atributovou složku s různými textovými a číselnými údaji, které je možné použít pro vyhledávání v těchto datech.

Nad těmito údaji je možno provádět vyhledávání pomocí standardních vyhledávacích mechanismů založených na shodě hodnoty atributu s vyhledávacím kritériem. Je však možné použít též fulltextové vyhledávání, které může značně urychlit a zefektivnit vyhledávání nad atributovou částí prostorových dat a umožní tak rychlejší a efektivnější nalezení polohy.

## PROSTOROVÁ DATABÁZE

Prostorová databáze (2) je relační databázový systém schopný spravovat (ukládat, zpracovávat, manipulovat, dotazovat) prostorová data. Tato data jsou uložena v tabulkách relační databáze. Díky tomu je možné, při práci s těmito daty, používat veškeré nástroje daného relačního databázového systému, včetně pokročilých metod vyhledávání a fulltextového vyhledávání. Uložení prostorových dat pomocí systému databázových tabulek umožňuje pracovat odděleně jak s prostorovou složkou, tak i atributovou složkou záznamu. Přitom nesmí dojít k porušení vazby mezi těmito složkami.

Pokud je prováděno vyhledávání nad prostorovými daty uloženými v tabulce, pak záleží pouze na definici dotazu, zda jeho výsledek bude obsahovat prostorovou složku, neprostorové atributy záznamu, či obojí. Prostřednictvím rozhraní může zvolený databázový systém zpřístupnit vybraná prostorová data a případně i vyhledávací nástroje pro práci s těmito daty dalším aplikacím. Na aplikaci, která dotaz odeslala, pak záleží, jak výsledek dotazu zpracuje. Například zda bude výsledek dotazu zobrazen nad mapou, nebo ve formě tabulky.

Prostorová databáze může být realizována jako databázová tabulka pomocí různých relačních databázových systémů. Například komplexní nástroj, umožňující pořizování, správu, analýzu a vizualizaci prostorových dat v on-line prostředí, který využívá prostorové databáze, ArcGIS Server může využívat databázové systémy IBM DB2, IBM Informix Dynamic Server, Microsoft SQL Server, Microsoft SQL Azure, Netezza, Oracle a PostgreSQL pro uložení a práci s prostorovými daty (3).

## VYHLEDÁVÁNÍ A DATABÁZE

Pro práci s daty v relačních databázových systémech je dnes většinou používán jazyk SQL - Structured Query Language (4). Jazyk SQL je standardizován a standard podporuje prakticky každý relační databázový systém, ale obvykle nejsou implementovány vždy všechny požadavky normy. A naopak, každý z databázových systémů obvykle obsahuje prvky a konstrukce, které nejsou ve standardech obsaženy. Přenositelnost SQL dotazů mezi jednotlivými databázemi je proto omezená.

Data manipulation language (DML) tvoří podmnožinu jazyka SQL (5). Příkazy jazyka DML umožňují vkládání, aktualizaci, mazání dat a také výběr dat pomocí příkazu.

Při vyhledávání pomocí příkazu `SELECT` lze při sestavování jeho klausule `WHERE`, která specifikuje vyhledávací kritéria, využít následující nástroje:

1. Booleovu algebru (5).
2. Zástupné znaky (`%` a `_`) a operátor `LIKE` (5).
3. Fulltextové vyhledávání (6).
4. Kombinaci předešlých nástrojů.

Následující dotaz provede výběr všech atributů konkrétního staničníku z databázové tabulky v relačním databázovém systému Oracle. Pro jeho vyhledání využije v klausuli `WHERE` název obce, na jejímž území se staničník nachází, název části obce, název trati dle občanského jízdního řádu a vzdálenost staničníku od nultého kilometru trati.

```
SELECT *
FROM fts_km_zeleznic
WHERE naz_obec_p = 'Beroun'
AND naz_cast_d = 'Beroun-Závodí'
AND trat_ojr_n = 'Praha - Rudná u Prahy - Beroun'
AND km_h = '2,1';
```

Takto formulovaný dotaz pomocí příkazu `SELECT` jazyka SQL (5) vyžaduje přesnou znalost hodnot použitých v klauzuli `WHERE`. Použití správných hodnot v klauzuli `WHERE` nemusí být vždy zcela jednoduché, pokud je k dispozici velké množství různých hodnot ze kterých je možno vybírat nebo pokud jsou mezi jednotlivými možnostmi velmi subtilní rozdíly, například přítomnost či nepřítomnost mezery v názvu.

Při formulaci dotazu je možné využít též zástupných znaků (`%` a `_`) a operátoru `LIKE`. Předešlý dotaz je možné s využitím těchto nástrojů přeformulovat například tímto způsobem:

```
SELECT *
FROM fts_km_zeleznic
WHERE naz_obec_p LIKE 'Bero%'
AND naz_cast_d LIKE '%Zá%'
AND trat_ojr_n LIKE '%Ru%'
AND km_h = '2,1';
```

Ze srovnání tohoto dotazu s předešlým dotazem je patrné, že použití operátoru `LIKE` a zástupných znaků (`%`) nevyžaduje natolik přesnou znalost hodnot použitých v podmínkách klauzule `WHERE`, jako při vyhledávání bez operátoru `LIKE`. Umožňuje též vyhledávat pomocí menšího počtu znaků než při vyhledávání bez operátoru `LIKE`.

Dotaz pomocí operátoru `LIKE` může být však mnohonásobně (až 10x) pomalejší než vyhledávání s uvedením přesných hodnot v klauzuli `WHERE` (7). Navíc použití operátoru `LIKE` činí dotaz poměrně složitým.

Využití nástrojů fulltextového vyhledávání umožňuje podstatné zjednodušení a zkrácení dotazu oproti předchozím variantám. Pro vyhledání stejného objektu lze použít s pomocí fulltextového vyhledávání následující dotaz:

```
SELECT *  
FROM fts_km_zeleznic  
WHERE CONTAINS(popisek_fts, 'bero% AND zá% AND ru% AND 2\,1') > 0;
```

Použití fulltextového vyhledávání nevyžaduje přesnou znalost hodnot pro vyhledávání. Nevyžaduje ani rozlišování mezi malými a velkými písmeny. Umožňuje výrazným způsobem zkrátit parametry dotazu.

Vyžaduje tvorbu fulltextového indexu (6). Pokud je možné fulltextové vyhledávání bez fulltextového indexu, pak je velmi pomalé.

Pokud není v databázové tabulce vhodný atribut, pak vyžaduje tvorbu zvláštního atributu pro generaci fulltextového indexu.

Syntaxe fulltextového vyhledávání se poměrně značně liší napříč různými relačními databázovými systémy. Fulltextové dotazy proto vyžadují další úpravy, pokud mají být přeneseny napříč různými databázovými systémy.

## FULLTEXTOVÉ VYHLEDÁVÁNÍ

Fulltextové vyhledávání, (z ang. full – celý, plný a text) je speciální způsob vyhledávání informací v textu, které jsou obvykle předem připraveny, tj. indexovány, aby bylo možno nalézt libovolné slovo (řetězec znaků) v co nejkratším možném čase (8). Při fulltextovém vyhledávání vyhledávací algoritmus prozkoumává všechna slova v prohledávaném textu a pokouší se je porovnat se slovy zadanými uživatelem.

Pokročilé relační databázové systémy (např. Oracle) umožňují vyhledávání v textu uloženém jak přímo v databázové tabulce, tak pod souborovým systémem či na internetu. Pokud text, ve kterém má být prováděno vyhledávání, není uložen přímo v databázové tabulce, pak musí tabulka obsahovat platné adresy souborů s textem. Prohledávaný text nemusí mít jen formu prostého textového řetězce v databázové tabulce, ale může být uložen v různých formátech (HTML, PDF, ...).

Při velkém rozsahu prohledávaného textu, je pro udržení rychlé odezvy třeba nejprve připravit index (8). Při indexování je programem provádějícím indexování - indexerem procházen text určený k prohledávání a vytvářen seznam klíčových termínů - index. Ve vyhledávací fázi, kdy se provádí specifický dotaz, je prohledáván pouze k tomu připravený index, namísto originálních dokumentů.

Indexer vytváří záznam v indexu pro každý výraz, nebo slovo, které najde v dokumentu a jeho relativní pozici v dokumentu. Obvykle indexer ignoruje tzv. stop-slova, jako jsou spojky a předložky, které jsou příliš obecné a mají malý, či zanedbatelný význam při vyhledávání. Některé indexery provádějí jazykově specifické úpravy, jako lemmatizaci (nalezení kmene slova) a derivaci (nalezení odvozených tvarů, tj. skloňování nebo časování).

Fulltextové vyhledávání lze zefektivnit pomocí následujících nástrojů (8):

1. Klíčová slova – vytváření seznamu slov, které popisují předmět textu, včetně synonym.
2. Hledání v polích – slova jsou vytříděna do specifických polí, je možné vyhledávat text pouze ve vyhrazených polích.
3. Booleovské operátory – používání operátorů `AND`, `NOT` a `OR` může podstatně zpřesnit vyhledávání.
4. Vyhledávání frází – vyhledávají se dokumenty, které obsahují přesnou frázi.

5. Konkordance – vyhledávání produkuje abecední seznam všech nejdůležitějších slov nacházejících se v aktuálním kontextu.
6. Vyhledávání podle blízkosti – hledá se fráze z více slov, které jsou od sebe vzdálené o specifikovaný počet slov.
7. Zástupné (žolíkové) znaky – hledání nahrazuje jeden nebo více znaků ve vyhledávacím dotazu.
8. Regulární výraz – poměrně složité, ale velice silné definování dotazu.

## DATA PRO LOKALIZACI

Z řady prostorových dat, která jsou u HZS ČR k dispozici, byla následující data vybrána jako vhodná pro potřeby lokalizace místa mimořádné události nebo jako místa, kde se nacházejí objekty, který by mohly ovlivnit průběh mimořádné události nebo její zdolávání:

1. Adresní místa (2 726 292 záznamů).
2. Bankomaty (2 387 záznamů).
3. Instituce – školy a pošty (22 392 záznamů).
4. Kilometráž silnic (17 840 záznamů).
5. Kilometráž vodních toků (52 561 záznamů).
6. Kilometráž železničních tratí (95 576 záznamů).
7. Sloupy veřejného osvětlení (233 014 záznamů).
8. Prodejny obchodních řetězců (8 580 záznamů).
9. Ulice (404 047 záznamů).
10. Vodní nádrže (16 508 záznamů).
11. Zeměpisné názvy (213 002 záznamů).
12. Železniční přejezdy (9 476 záznamů).

Aby bylo co nejvíce urychleno a zjednodušeno vyhledávání v těchto datech, bylo rozhodnuto, že tato data budou upravena pro fulltextové vyhledávání.

Jedná se o data s bezprostředním vztahem ke zdolávání mimořádných událostí, ochraně obyvatelstva, havarijnímu plánování a krizovému řízení. Například školy a prodejny představují místa s časově proměnlivou zvýšenou koncentrací osob. Takovýmto místům je třeba věnovat zvýšenou pozornost při mimořádných událostech, které by je mohly ohrozit.

Vybraná prostorová data jsou uložena jako prostorové databáze v tabulkách pokročilého relačního databázového systému Oracle 11.2 (9). To poskytuje možnost využít při práci s těmito daty veškeré nástroje, které poskytuje tento databázový systém.

Vedle toho, že k datům je možné přistupovat prostřednictvím rozhraní databázového systému Oracle, je možné k datům též přistupovat i prostřednictvím ArcGIS Serveru 10.1 firmy ESRI (3). Přístup k prostorovým datům uloženým v relačním databázovém systému Oracle prostřednictvím ArcGIS Serveru poskytuje řadu výhod:

1. Snadný přístup k datům uloženým v databázi přes rozhraní typu REST pomocí protokolu HTTP a zpráv ve formátu JSON.
2. Snadný přístup k datům uloženým v databázi přes rozhraní typu RPC (vzdálené volání procedur) pomocí protokolu HTTP a zpráv ve formátu SOAP.
3. Snadná publikace dat zpracovávaných v desktopových aplikacích firmy ESRI.

Přístup k datům uloženým v databázi prostřednictvím webových služeb, a to ať již pomocí rozhraní typu REST nebo typu RPC, je plně v souladu se současným trendem budování rozsáhlých informačních systémů podle zásad servisně orientované architektury.

Přístup k datům pomocí rozhraní typu REST se jeví jako mimořádně výhodný, pokud mají k datům přistupovat webové aplikace.

Firma ESRI poskytuje pokročilé nástroje pro tvorbu webových aplikací, které jsou zaměřené na práci s prostorovými daty, pro různé programovací jazyky, například ArcGIS API pro jazyk Adobe Flex a JavaScript.

Vzniká tak unikátní prostředí, platforma, které umožňuje snadno, rychle a kvalitně pomocí produktů ESRI zpracovávat, publikovat a prezentovat prostorová data jejich uživatelům.

## PŘÍPRAVA TABULKY PRO FULLTEXTOVÉ VYHLEDÁVÁNÍ

Fulltextové vyhledávání je velmi silným nástrojem při vyhledávání v databázových tabulkách relačního databázového systému. Nelze však pominout fakt, že mají-li být naplno využity možnosti fulltextového vyhledávání, je třeba databázovou tabulku pro fulltextové vyhledávání upravit.

Příprava tabulky pro fulltextové vyhledávání lze rozložit do následujících kroků:

1. Analýza dat v tabulce, nad kterými má probíhat fulltextové vyhledávání.
2. Volba atributu, nad kterým bude probíhat fulltextové vyhledávání, či generace nového atributu, pokud v tabulce již neexistuje vhodný atribut.
3. Pokud bude generován nový atribut pro fulltextové vyhledávání, pak musí být vytvořen program pro jeho generaci a musí být vybrány stávající atributy, které budou do nového atributu zahrnuty.
4. Generace fulltextového indexu.
5. Testování fulltextového vyhledávání nad tabulkou.

Analýza vstupních dat je často opomíjeným, ale přesto nezbytným, prvním krokem v přípravě databázové tabulky pro fulltextové vyhledávání. V rámci analýzy vstupních dat je třeba především zvážit jak celkový význam, charakter a vlastnosti prostorových dat uložených v tabulce, tak i jednotlivých atributů:

1. Formátu atributu (celé číslo, desetinné číslo, textový řetězec, přesnost,...).
2. Značení nedefinovaných hodnot ('', ' ', 0, '0', 0.0, '0.0', 'none', none,...).
3. Platné a neplatné hodnoty atributů.
4. Chybovost atributů.
5. Vazby na další tabulky databáze (například RÚIAN) (10).

Hlavním cílem analýzy vstupních dat je zjistit, zda již existuje v databázové tabulce atribut, nad kterým je možné provádět fulltextové vyhledávání, nebo zda musí být takovýto atribut vytvořen.

Pokud musí být vytvořen nový atribut, pak je třeba rozhodnout, které stávající atributy do něj mají být zahrnuty. Atributy, ze kterých bude nový atribut pro fulltextové vyhledávání vytvořen, se nemusí nutně nacházet ve stejné tabulce jako vytvářený atribut, ale mohou být umístěny i v jiných tabulkách databázového systému a přístupné pomocí systému klíčů.

U zvolených atributů je třeba zvážit, zda jsou definovány pro všechny záznamy v tabulce, nebo pouze pro některé:

1. Pro všechny záznamy v tabulce je (by měl být) daný atribut definován (např. obec, katastrální území, okres, ...).
2. Alespoň pro jeden záznam v tabulce je (by měl být) daný atribut definován (např. číslo popisné, ulice, část obce, ...).

Vzhledem k tomu, že databázová tabulka obsahuje reálná data, je potřeba navíc vzít v úvahu i fakt, že tato data mohou být zatížena chybami. To je důvod podmínky v závorce „by měl být“ v předešlém výčtu.

Pro analýzu vstupních dat a i ostatní práci s databázovým systémem Oracle 11.2, jako například testování SQL příkazů a kontrolu výsledků fulltextového vyhledávání, byl používán volně dostupný multiplatformní databázový klient Oracle SQL Developer (11).

## PROGRAM PRO PŘÍPRAVU DATABÁZOVÉ TABULKY NA FULLTEXTOVÉ VYHLEDÁVÁNÍ

Při analýze vybraných dat se nepodařilo v žádné z vybraných prostorových databází nalézt vhodný atribut, který by bylo možno použít přímo bez dalších úprav pro generaci fulltextového indexu a následně i fulltextové vyhledávání.

V databázových tabulkách ale existují atributy, které obsahují vhodné údaje pro fulltextové vyhledávání. Tyto atributy tedy představují vhodné kandidáty pro sloučení do nového atributu - „fulltextového popisku“. Nad tímto popiskem může být následně vygenerován fulltextový index a prováděno fulltextové vyhledávání.

Některé z vybraných atributů, které je vhodné použít pro tvorbu fulltextového popisku, neumožňují přímé sloučení s ostatními atributy bez dalších úprav, protože nejsou textového charakteru. Jedná se například o numerické údaje, či je není vhodné do nového atributu začlenit bez dalších úprav, protože obsahují drobné chyby, například znak „?“ místo „-“ v názvu trati dle občanského jízdního řádu.

Navíc při pouhém prostém sloučení textových atributů do jediného atributu nemusí být vždy zřejmé, odkud pocházela původní část nového atributu.

Z výše uvedených skutečností proto vyplynula nutnost vytvořit program, který by tyto skutečnosti řešil. Protože již bylo potřeba přistoupit k tvorbě programu pro přípravu nového atributu pro fulltextové vyhledávání - fulltextového popisku, byla do tohoto programu zahrnuta i automatizace ostatních činností spojených s přípravou databázové tabulky pro fulltextové vyhledávání.

Vytvořený program pro generaci fulltextových popisků proto provádí následující činnosti:

1. Naváže spojení s databází a provede kontrolu spojení.
2. Zjistí počet záznamů v tabulce databáze, která má být upravena pro fulltextové vyhledávání.
3. Načte atributy uložené v tabulce, které mají být použity pro tvorbu nového atributu – fulltextového popisku.
4. Provede opravu drobných chyb v načtených vstupních datech. Opravy se neukládají do zdrojové tabulky databáze.
5. Vygeneruje fulltextový popis pro každý záznam v tabulce. Fulltextový popis vzniká sloučením jednotlivých vybraných atributů. Každý z atributů zahrnutých do fulltextového popisku je opatřených poznámkou o svém významu (obrázek 2).
6. Zjistí, zda databázová tabulka obsahuje sloupec s názvem, který je totožný s názvem sloupce určeného pro uložení fulltextových popisků. Pokud ano, pak nejprve odstraní veškeré indexy definované nad tímto sloupcem a následně odstraní i sloupec jako takový. Odstranění starého sloupce, pokud již existoval, zamezí vzniku problémů s částečnou aktualizací popisků, pokud by generování fulltextového popisku nebylo prováděno pro všechny záznamy v tabulce.
7. Vytvoří prázdný sloupec pro uložení fulltextových popisků a opatří tento sloupec komentářem. Šířka sloupce (maximální počet znaků, který lze do sloupce uložit) se řídí podle počtu znaků nejdelšího fulltextového popisku.
8. Uloží vygenerované fulltextové popisky do nově vytvořeného sloupce.
9. Zjistí, zda již existuje index se jménem, které je navrhováno pro nový fulltextový index.

10. Pokud není nalezen index se jménem, které by bylo totožné se jménem pro nový fulltextový index, pak je vytvořen fulltextový index nad sloupcem fulltextových popisků.
11. Program provede odhad času nezbytného pro zpracování všech záznamů v tabulce. To má význam tehdy, pokud je zpracovávána pouze část z celkového počtu záznamů uložených v tabulce.

Činnost programu je znázorněna na obrázku 2. Na tomto obrázku je zachycen stav jednoho záznamu databázové tabulky adresních bodů před jeho zpracováním programem pro tvorbu fulltextových popisků a po jeho zpracování.

OBJECTID	1679668		OBJECTID	
NAZ_OBEC	Beroun		NAZ_OBEC	
NAZ_CAST_D	Beroun-Hostim		NAZ_CAST_D	
NAZEV_UL_A	Kozelská		NAZEV_UL_A	
CIS_D	42		CIS_D	
CIS_O	0		CIS_O	
CISLO_ORIE	0		CISLO_ORIE	
CISLO_OR_1			CISLO_OR_1	
TYP_CIS	1		TYP_CIS	
ZAD_VCHOD	0		ZAD_VCHOD	
NAZ_NNUTS4	Beroun		NAZ_NNUTS4	
NAZ_KU_P	Hostim u Berouna		NAZ_KU_P	
NAZ_ZSJ_P	Hostim		NAZ_ZSJ_P	
CZNUTS3	3026		CZNUTS3	
IDADR	5927099		IDADR	
KOD_OBEC	531057		KOD_OBEC	
KOD_CAST_D	401897		KOD_CAST_D	
LAU1	CZ0202		LAU1	
ULICE_ID	0794406		ULICE_ID	
KOD_ZSJ_P	0457300		KOD_ZSJ_P	
IDSO	189588		IDSO	
IDOB	5927099		IDOB	
KN_BUDOVA	http://nahlizenidokn.cuzk.cz/...		KN_BUDOVA	
SHAPE	[SDE.ST_GEOMETRY]		SHAPE	
			POPISEK_FTS	"obec: Beroun, část obce: Beroun-Hostim, ulice: Kozelská, č.d.: 42, nuts4 (okres): Beroun, kú: Hostim u Berouna, zsj: Hostim"

**Obr. 2** Činnosti programu pro tvorbu fulltextových popisků. Činnost programu je demonstrována na zpracování jednoho záznamu z databázové tabulky adresních míst. Do tabulky byl přidán nový atribut, fulltextový popisec, `popisek_fts`. Jeho hodnota byla vytvořena z vybraných, již existujících, atributů tabulky. K atributům použitým pro tvorbu fulltextového popisku byl přidán text, který stručně a srozumitelně informuje o významu atributů použitých ke tvorbě fulltextového popisku.

Pro zpracování každé databázové tabulky obsahující vybraná prostorová data byl vytvořen samostatný programový kód. Přestože je řada částí kódu pro různé databázové tabulky takřka identická, nejevilo se v současné době jako optimální vytvářet jeden univerzální programový kód pro zpracování všech databázových tabulek. Důvodem byla poměrná různorodost zpracovávaných dat (staničníky, vodní nádrže, bankomaty, školy, ...). Do budoucna lze uvažovat o tvorbě univerzálního programového kódu, který by zpracovával veškerá vstupní data. Činnost částí programu, které by byly specifické pro jednotlivé databázové tabulky, by byla řízena pomocí konfiguračních souborů.

Při tvorbě programů byly co nejdůsledněji uplatňovány postupy programování řízeného testy (Test-driven development (TDD)) (12, 13). Tyto postupy vedou ke zlepšení logiky programového kódu a snížení chybovosti vzniklého kódu.

Jednotkové testy, vzniklé při uplatňování zásad programování řízeného testy, mohou být navíc opakovaně použity při regresním testování, to jest při zjišťování, zda později provedené úpravy programu negativně



neovlivnily jeho již jednou úspěšně otestované vlastnosti. Při vývoji programů pro přípravu databázových tabulek pro fulltextové vyhledávání byly připraveny následující skupiny jednotkových testů:

1. Testy vlastního sestavení fulltextového popisku.
2. Testy fulltextového vyhledávání nad připravenou tabulkou v SŘBD Oracle.
3. Testy fulltextového vyhledávání nad připravenou tabulkou v SŘBD Oracle prostřednictvím ArcGIS Server REST API.

Dále byla při tvorbě programů věnována důsledná pozornost ošetření chybových stavů programů pomocí výjimek. Při formulaci kódu zpracovávajícího výjimky byla snaha o řádné ukončení všech vykonávaných kroků, jako například uzavření otevřeného souboru a ukončení navázaného spojení s databází.

Pro tvorbu programů pro přípravu jednotlivých databázových tabulek pro fulltextové vyhledávání byl použit programovací jazyk Python ve verzi 3.3. Důvody pro tuto volbu byly především následující:

1. Vysoká efektivita programování.
2. Krátká doba od vývoje aplikace do jejího nasazení.
3. Existence knihovny `cx_Oracle` pro připojení k SŘBD Oracle (14).
4. Existence řady dalších vyspělých knihoven, například pro jednotkové testování, logování, práci se soubory ve formátu JSON, ... .
5. Možnost jak objektového, tak i procedurálního programování.
6. Vylepšená podpora práce se specifickými znaky národních abeced. Ve verzi jazyka Python 3.X jsou veškeré řetězce sekvencí unikódových znaků.
7. Rozsáhlá a přitom srozumitelná a přehledná dokumentace (15, 16, 17, 18).

Veškerá činnost programu byla logována. To umožnilo kontrolu činnosti programu a poskytlo informace o časové náročnosti jednotlivých kroků programu.

Logování programů používá jednotný mechanismus záznamů činnosti programů jak pro výpis na obrazovce, tak i pro zápis do souboru. Veškeré zprávy jsou zaznamenávány do souboru a pouze zprávy od určité úrovně závažnosti jsou zobrazovány i na obrazovce. Soubor se záznamem o činnosti programu je možno použít k pozdější analýze činnosti programu. Průběh logování je možno nastavit pomocí externího konfiguračního souboru.

Pro logování je používána standardní knihovna jazyka Python - logging (19). Vzhledem k počtu zpracovávaných položek a objemu zpracovávaných dat je pro prohlížení textových souborů, které obsahují záznamy o činnosti programu, vhodnější použít textové editory uzpůsobené pro práci s velkými textovými soubory.

Pro tvorbu programu bylo využito integrované vývojové prostředí (IDE) Eclipse ve verzi Kepler (20). Do vývojového prostředí Eclipse byl doinstalován zásuvný modul PyDev, který umožňuje vývoj v jazyce Python v tomto prostředí (21).

## ORACLE TEXT

Pod relačním databázovým systémem Oracle (22) je možno použít pro práci s textem technologii Oracle Text (23, 24, 25). Tato technologie umožňuje použití fulltextového vyhledávání v databázovém systému Oracle. Je součástí Oracle 11g Standard a Enterprise Edition.

Oracle Text používá standardní příkazy SQL pro fulltextové indexování, vyhledávání a analýzu textu a dokumentů uložených v databázi Oracle, v souborech na lokálních počítačích nebo na Webu.

Vedle základního fulltextového indexu typu `CONTEXT` umožňuje Oracle Text definovat i další speciální typy indexů pro specifické způsoby použití. Jednotlivé typy indexů mohou být dále přizpůsobeny charakteru indexovaných dokumentů.

Pro generování indexu typu CONTEXT lze použít následující příkaz:

```
CREATE
INDEX idx_fts_stanicniky_fts
ON fts_km_zeleznic (popisek_fts)
INDEXTYPE IS CTXSYS.CONTEXT;
```

Tento příkaz vytvoří fulltextový index typu CONTEXT se jménem idx\_fts\_stanicniky\_fts. Index bude vygenerována nad atributem popisek\_fts tabulky fts\_km\_zeleznic.

Jméno indexu musí být jedinečné v rámci schématu databáze. Schéma databáze je pojmenovaný soubor objektů databáze, včetně logických struktur jako jsou tabulky a indexy. Schéma má stejné jméno jako uživatel databáze, který ho vlastní (26).

Indexy mají vlastní jmenný prostor v rámci schématu databáze. V rámci jmenného prostoru nemohou mít žádné dva objekty stejné jméno (27).

## ZÁVĚR

Vybraná prostorová data byla úspěšně připravena pro fulltextové vyhledávání. Díky tomu mohou tato data sloužit k rychlé lokalizaci místa mimořádné události a lokalizaci dalších objektů, které by mohly, díky svému charakteru, významně ovlivnit řešení mimořádné události.

Fulltextové vyhledávání nad těmito daty je přístupné v principu jakékoliv aplikaci. Od desktopové aplikace pro krizové řízení, havarijní plánování či ochranu obyvatelstva, přes webového mapového klienta realizovaného pomocí JavaScriptu či Adobe Flexu, až po aplikace mobilních telefonů. Je tomu tak proto, že vyhledávání v datech není přístupné pouze prostřednictvím standardního rozhraní databáze, ale i prostřednictvím standardizovaných rozhraní ArcGIS Serveru formou webových služeb typu REST a RPC. Právě to jej činí přístupné v principu jakékoliv aplikaci schopné komunikovat prostřednictvím HTTP protokolu.

Fulltextové vyhledávání nad vybranými prostorovými daty bylo úspěšně otestováno a to jak v prostředí databázového systému Oracle, tak i pomocí rozhraní ArcGIS Server REST API. Již jednou vytvořené testy je možné opakovaně použít při ověřování funkčnosti fulltextového vyhledávání a to jak při aktualizaci zpracovávaných dat, tak i při dalším vývoji programů pro přípravu databázové tabulky pro fulltextové vyhledávání.

Programy pro přípravu databázových tabulek pro fulltextové vyhledávání se úspěšně vyrovnaly s faktem, že některé upravované tabulky obsahovaly poměrně značné množství dat, stovky tisíc až miliony záznamů.

Do budoucna lze uvažovat o začlenění dalších atributů do fulltextového vyhledávání, či přípravě dalších prostorových data pro fulltextové vyhledávání. Jako mimořádně zajímavá se jeví příprava RÚIANu pro fulltextové vyhledávání.

## LITERATURA

- (1) Soud uložil výpravčímu za nehodu vlaku u Vraňan podmínku. ZIKMUNDOVÁ, Lucie. *IDnes.cz / Zprávy* [online]. 2008 [cit. 2013-12-09]. Dostupné z: [http://zpravy.idnes.cz/soud-ulozil-vypravcimuz-nehodu-vlaku-u-vranan-podminku-pml-/krimi.aspx?c=A081222\\_111020\\_krimi\\_lf](http://zpravy.idnes.cz/soud-ulozil-vypravcimuz-nehodu-vlaku-u-vranan-podminku-pml-/krimi.aspx?c=A081222_111020_krimi_lf)
- (2) Spatial database. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2013 [cit. 2013-10-03]. Dostupné z: [http://en.wikipedia.org/wiki/Spatial\\_database](http://en.wikipedia.org/wiki/Spatial_database)
- (3) ArcGIS for Server: Deliver GIS as a Service. ESRI. *Esri - GIS Mapping Software, Solutions, Services, Map Apps, and Data* [online]. 2013 [cit. 2013-12-16]. Dostupné z: <http://www.esri.com/software/arcgis/arcgisserver>
- (4) SQL. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2013 [cit. 2013-10-04]. Dostupné z: <http://cs.wikipedia.org/wiki/SQL>

- (5) LACKO, Luboslav. *Oracle: správa, programování a použití databázového systému*. 2. dopl. vyd. Překlad Marek Kocan. Brno: Computer Press, 2007, 576 s. ISBN 978-80-251-1490-2.
- (6) LONEY, Kevin. *Oracle Database: kompletní průvodce*. Vyd. 1. Brno: Computer Press, 2010, 1368 s. ISBN 978-80-251-2489-5.
- (7) SQL 'like' vs '=' performance: 8) It's a measureable difference. In: *Stackoverflow* [online]. 2011 [cit. 2013-10-03]. Dostupné z: <http://stackoverflow.com/questions/6142235/sql-like-vs-performance>
- (8) Full text search. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2013 [cit. 2013-10-03]. Dostupné z: [http://en.wikipedia.org/wiki/Full\\_text\\_search](http://en.wikipedia.org/wiki/Full_text_search)
- (9) ORACLE CZECH S. R. O. *Oracle Česká republika: Hardware and Software, Engineered to Work Together* [online]. 2013 [cit. 2013-12-10]. Dostupné z: <http://www.oracle.com/cz/index.html>
- (10) Výměnný formát RÚIAN (VFR). *Český úřad zeměměřický a katastrální* [online]. 2013 [cit. 2013-12-10]. Dostupné z: <http://www.cuzk.cz/Dokument.aspx?MENUID=10769&AKCE=DOC:10-vybudovani-RUIAN>
- (11) Oracle SQL Developer. ORACLE CORPORATION. *Oracle* [online]. 2013 [cit. 2013-10-07]. Dostupné z: <http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index.html>
- (12) SOMMERVILLE, Ian. *Softwarové inženýrství* 1. vyd. Brno: Computer Press, 2013, 680 s. ISBN 978-802-5138-267.
- (13) Programování řízené testy. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2013 [cit. 2013-11-28]. Dostupné z: [http://cs.wikipedia.org/wiki/Programování\\_řízené\\_testy](http://cs.wikipedia.org/wiki/Programování_řízené_testy)
- (14) ANTHONY TUININGA. *cx\_Oracle* [online]. 2013 [cit. 2013-11-28]. Dostupné z: <http://cx-oracle.sourceforge.net/>
- (15) PYTHON SOFTWARE FOUNDATION. *Overview - Python v3.3.3 documentation* [online]. 2013 [cit. 2013-11-28]. Dostupné z: <http://docs.python.org/3/>
- (16) SUMMERFIELD, Mark. *Python 3: výukový kurz*. Vyd. 1. Překlad Lukáš Krejčí. Brno: Computer Press, 2010, 584 s. ISBN 978-80-251-2737-7.
- (17) LUTZ, Mark. *Learning Python*. 5th ed. Beijing: O'Reilly Media, 2013, 1540 s. ISBN 978-1449355739.
- (18) BEAZLEY, David M a Brian K JONES. *Python cookbook*. 3rd ed. Beijing: O'Reilly, 2013, xvi, 687 s. ISBN 978-1-449-34037-7.
- (19) 16.7. logging: Logging facility for Python. *Python v3.3.3 documentation* [online]. 2013 [cit. 2013-12-02]. Dostupné z: <http://docs.python.org/3.3/library/logging.html>
- (20) THE ECLIPSE FOUNDATION OPEN SOURCE COMMUNITY. *Eclipse: The Eclipse Foundation open source community website* [online]. 2013 [cit. 2013-11-27]. Dostupné z: <http://www.eclipse.org/>
- (21) *PyDev* [online]. 2013 [cit. 2013-11-27]. Dostupné z: <http://pydev.org>
- (22) ORACLE. *Oracle Database Online Documentation 11g Release 2 (11.2)* [online]. 2013 [cit. 2013-10-07]. Dostupné z: <http://www.oracle.com/pls/db112/homepage>
- (23) ORACLE. *Oracle Text: An Oracle Technical White Paper June 2007* [online]. 2007 [cit. 2013-10-07] 133192. Dostupné z: <http://www.oracle.com/technetwork/database/enterprise-edition/11goracletextwp-133192.pdf>
- (24) ORACLE. *Oracle Text Application Developer's Guide: 11 g Release 2 (11.2)* [online]. 2013 [cit. 2013-10-07]. E24435-02. Dostupné z: [http://docs.oracle.com/cd/E11882\\_01/text.112/e24435.pdf](http://docs.oracle.com/cd/E11882_01/text.112/e24435.pdf)
- (25) ORACLE. *Oracle Text Reference: 11 g Release 2 (11.2)* [online]. 2013 [cit. 2013-10-07]. E24436-03. Dostupné z: [http://docs.oracle.com/cd/E11882\\_01/text.112/e24436.pdf](http://docs.oracle.com/cd/E11882_01/text.112/e24436.pdf)

- 
- (26) Glossary. ORACLE. *Oracle Database Concepts 11g Release 2 (11.2)* [online]. 2013 [cit. 2013-10-08]. Dostupné z: [http://docs.oracle.com/cd/E11882\\_01/server.112/e40540/glossary.htm](http://docs.oracle.com/cd/E11882_01/server.112/e40540/glossary.htm)
- (27) Database Object Names and Qualifiers . ORACLE. *Oracle Database SQL Language Reference 11g Release 2 (11.2)* [online]. 2013 [cit. 2013-10-08]. Dostupné z: [http://docs.oracle.com/cd/E11882\\_01/server.112/e41084/sql\\_elements008.htm](http://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements008.htm)