Ing. Richard Hébr
VAKUS Praha

STROJ ROZMLOUVÁ S ČLOVĚKEM ANEB MALÁ ČESKÁ BANKA DAT

1. Úvod.

Před dvěma léty kdekdo psal a mluvil o databankách. Móda je proměnlivá a v té době byl už výraz "systémový přístup" natolik omšelý, že slovo "databanka" bylo vítanou inovací v písemných a ústních projevech. Vše se však vyvíjí. Dnes už se podle názvoslovných předpisů říká "banka dat" a navíc - jak to tak v životě chodí - bývají horlivým řečníkům stále častěji kladeny dotazy, kdy že už ony opěvované banky dat budou někde k mání. V této zdánlivě nepříjemné situaci bylo nalezeno řešení: soubor dat pro jakoukoliv agendu se hrdě označí jako "banka dat" (i když se mu původně říkalo docela prosaicky "kmen"). Odběrníci s podivem zjišťují, že se to u nás bankami dat jen hemží.

Na druhé straně však existují seriozní pokusy o řešení specifických problémů banky dat. V tomto příspěvku budou některé speciální otázky banky dat rozebrány a možné přístupy k jejich řešení ukázány na jednom z realizovaných modelů banky dat.

2. Souhrnný popis systému.

2.1 Pojmy:

Názory na názvosloví se dosud různí a proto si uvedeme základní pojmy (viz na př. lit./1/):

- "báze dat" je soubor nebo několik souborů dat, splňujících níže popsané požadavky;
- "banka dat" je báze dat, doplněná o systém řízení báze dat;
- "systém řízení báze dat" (dále jen SŘBD) je soustava programů, zajišťujících provoz banky dat a plnění základních funkcí.

Základními funkcemi banky dat se rozumí:

- definování dat v bázi dat
- vytváření obsahu báze dat
- výběr a výstup dat a informací.

Báze dat má splňovat alespoň tyto základní požadavky:

- minimální nadbytečnost (vyloučení duplicitních dat)
- vyjádření a uchování vazeb mezi daty
- umožnění nezávislosti programů na struktuře dat
- zajištění ochrany dat

atd.

2.2 Experimentální banka dat:

V rezortu spojů bylo rozhodnuto zpracovat experimentální model banky dat, jehož hlavní cíle byly definovány takto:

- vyřešit a ověřit některé principy banky dat
- získat praktické zkušenosti, které budou využívány při aplikacích banky dat ve velkých spojových úlohách
- použít experimentální model pro rutinní provoz vybraných úloh.

Řešitelská práce měla být zaměřena k rozboru těchto problémových oblastí:
- efektivní uspořádání dat
- dotazovací systém, určený pro uživatele-laiky.

2.3 Podmínky konkrétního řešení:

Základní podmínkou byla jednoduchost a to jak systému, tak i jeho obsluhy. Další podmínkou byla důsledná modularita, aby bylo možno postupem času do systému zabudovávat nové funkce.

Pro řešitele byl zajímavý požadavek terminálového provozu. Řešily se na př. i psychologické aspekty uživatele laika, který je sám před terminálem a jeho partnerem, služebníkem i rádcem je pouze počítač. Předpokládáme dokonce uživatele líného, netrpělivého a celkově nevlídného. Systém banky dat by ho neměl obtěžovat, ale spíše mu pozvednout náladu.

2. Uspořádání datové báze.

2.1 Klasické modely:

Většina systémů banky dat se opírá o stromové a síťové modely. Jejich uspořádání a výhody jsou obecně známé. Uveďme si nevýhody těchto modelů:
- složité programové řešení SŘBD
- velké nároky na rozsah vnitřní i vnější paměti.

Zajímavé je, že stromová i síťová struktura plně vyhovují programátorskému pojetí struktury dat, ale méně již zapadají do logiky myšlení uživatele.

V "klasické" bance dat je nutné pojmenovat nejen položky, ale i vztahy. Uživatel pak musí vědět
- že vztah v bance dat existuje
- jak se jmenuje

aby mohl s bankou dat úspěšně komunikovat. Složitá soustava jmen neúměrně zatěžuje uživatele.

2.2 Relační model:

V roce 1970 popsal E.F.Codd tak zvaný relační model datové struktury (lit./2/). Využívá se zde matematické teorie množin a zavádí v podstatě zcela nový pohled na datové struktury. Vazby v datové bázi se řeší jiným způsobem než v klasických bankách dat.

Některá hlediska relačního pohledu na data můžeme ukázat na tomto příkladu:
Věta zaměstnance v personální bance dat obsahuje tyto položky:

● číslo zaměstnance / jméno / datum nar. / platový vývoj ●

Položka "platový vývoj" je t.zv. násobná položka; skládá se ze dvou částí:

> datum / plat <

Počet těchto dvojic je pro různé zaměstnance různý a je obtížné stanovit maximální počet výskytů. V "klasické" bance dat volíme buď dostatečnou rezervu ve větě nebo proměnnou délku věty nebo jiný komplikovaný způsob.
V relační bázi dat vytvoříme dva typy vět:

● číslo zaměstnance / jméno / datum nar. ●
● číslo zaměstnance / datum / plat ●

Z příkladu je zřejmý efekt řešení a současně možnosti, které se zde nabízejí.

Nadšence, kteří by po přečtení těchto řádek chtěli ihned realisovat relační báze dat nutno upozornit, že věc je v praxi poněkud složitější. Zájemce odkazuji na př. na lit. /2/,/3/ a upozorňuji, že se od analytika vyžadují znalosti poměrně odlehlých matematických a logických disciplín, které se na technických školách zatím opomíjejí (začínají však pronikat do náročné školy).

Relační model dat má závažnou výhodu z hlediska uživatele: vyhovuje logice laického myšlení a navíc i logice přirozeného jazykového projevu. Až dorostě "množinová generace", bude pro ni relační přístup k datům samozřejmostí a školení uživatelů nebude činit vůbec problémy.

2.3 Logická a fyzická struktura dat:

Logická struktura dat je uspořádání dat, jak je vidí uživatel, programátor nebo většina systémových programů banky dat. Fyzická struktura je konkrétní realizace dat v paměti počítače.

Banka dat musí tyto struktury učinit na sobě nezávislými. Programy nikdy s fyzickou strukturou přímo nepracují.

Poznámka:
Uvedené pojmy jsou definovány poněkud stroze a zjednodušeně. Přesné vymezení je na př. v lit./1/,/4/ a pod.

V praxi se toho dosahuje takto:
- vkládání a výběry dat provádějí zásadně pouze systémové programy;
- struktury dat jsou popsány v tabulkách a seznamech; programy si z těchto tabulek a seznamů berou informace, potřebné pro jejich práci.

SŘBD tedy existuje v obecné, neměnné podobě. Při aplikaci systému na konkrétní datovou bázi nutno nejprve provést "definici" struktur. Při této definici sdělíme systému naše požadavky na logickou a fyzickou strukturu datové báze. Systém vytvoří příslušné tabulky a seznamy a zapamatuje si je. SŘBD je od tohoto okamžiku schopen pracovat s námi definovanými daty.

2.4 Zajímavá zkušenost:

Při konkrétních aplikacích naší banky dat jsme s podivem konstatovali, že ani autoři systému nebyli zprvu schopni navrhovat účelné struktury dat. Chybí zde výchova i zkušenosti v práci s bankami dat. Rodí se patrně nová profese : "datový analytik" by měl ovládat různé systémy banky dat a navrhovat efektivní datové struktury pro konkrétní aplikace.

Autora jímá hrůza při představě, jak bude účelnost této nové funkce vysvětlována zodpovědným úředníkům, kteří dosud nenabyli reálné představy ani o náplni práce

analytiků, programátorů a (čtenář promine tu smělost) systémových inženýrů.

Uvedu nyní příklad přístupu ke struktuře dat (bude poněkud extrémní, ale je vzat z praxe).
Evidence pracovníků určitého oboru může být (zcela nesprávně) řešena jako obdoba platných výkazů:

Pro každou organizaci, podnik a závod věta

● /název organizace / sídlo / ..... ●
/počet vedoucích pracovníků specializace A/
/z toho vysokoškoláků/
/počet výkonných pracovníků specializace A/
/z toho vysokoškoláků/
/počet vedoucích pracovníků specializace B/
/z toho vysokoškoláků/
/počet výkonných pracovníků specializace B/
/z toho vysokoškoláků/
atd.

Lepší řešení pro banku dat (konkrétně pro relační model):

Věta "organizace"

● /číslo org. / název / sídlo / ..... ●

Věta "pracovníci"

● /číslo org. / funkce / specializace / vzdělání / počet / ●

V prvém návrhu působí nepříjemnosti na př. zavedení nové specializace; přímo katastrofou je požadavek na evidenci další funkce (na př. pomocný pracovník). První návrh je navíc velmi ztrnulý co do evidence stupňů vzdělání. Druhý návrh je pružný a bez obtíží realizuje různé požadavky.

První návrh vyžaduje v každé větě rezervovat místo pro všechny možné specializace, funkce a vzdělání (nebo proměnnou délku věty); druhý návrh je z tohoto hlediska hospodárnější a programátorsky pohodlnější. Jestliže pak posuzujeme oba návrhy z hlediska možných požadavků výběru, zvítězí druhý návrh na celé čáře.

3. Komunikace se systémem.

3.1 Východiska řešení:

Problém komunikace laika se systémem je velice ožehavý (viz lit./5/). Autor i dnes trvé na názoru, že uživatel laik zcela právem nechce být programátorem a odmítá se učit různé komplikované jazyky pro styk se systémem.

Dále je nutno mít na paměti neblahou zkušenost tvůrců ALGOLu : tento jazyk je navržen nesmírně důmyslně a umožňuje tvorbu fantastických logicko-aritmetických výrazů a kdovičeho ještě. Existují fanatikové, kteří dokáží v ALGOLu napsat celý program jediným příkazem; experti se vyžívají v hledání paradoxů jazyka, demonstrují jemnosti v odstínění různých možností zápisu a tvoří složité struktury "vnořených" závorek a bloků. Prostý programátor žasne, co všechno ALGOL umí. Celý svět však programuje v COBOLu a FORTRANu a ALGOL zůstal hříčkou pro teoretiky. V jednoduchosti totiž bývá síla.

Složitý dotaz na banku dat přijde v praxi jednou za dlouhý čas. Uživatel zapomíná náročné finesy, kterých běžně nepoužívá a neumí tedy komplikovaný dotaz sestavit. Než by hledal v manuálech, raději položí dva nebo tři jednoduché dotazy pomocí prostředků, které ovládá.

Chápu hřejivý pocit autora chytrého jazyka, který dovede negovat negaci negace logického součinu. Pro praxi je to však dost zbytečné.

Proto jsme se rozhodli volit pro naši banku dat komunikaci co nejjednodušší.

3.2 Kombinace "jazyk - dialog":

Každá banka dat musí "umět" alespoň tyto funkce:
- oživení (přírůstek - úbytek - změna)
- výběr a tisk informací, získaných z datové báze.

Porovnejme nyní obě tyto funkce z hlediska uživatele. Oživování je funkce komplikovaná (liší se dost podstatně od klasického oživování kmenů), avšak postup je vždy stejný

(nebo podobný). Výběr a tisk jsou funkčně jednodušší, postup však je značně mnohotvárný, neboť uživatelova přání nelze předem odhadnout.

Proto jsme zvolili pro oživení dialog (systém klade dotazy a uživatel na ně odpovídá), pro výběr a tisk dotazovací jazyk (uživatel sám formuluje příkazy systému). O dialogu je dosti podrobně pojednáno v /5/, věnujeme tedy pozornost dotazovacímu jazyku.

3.3 Principy českého dotazovacího jazyka:

Základním požadavkem byla jednoduchost jazyka a snadná zvládnutelnost pro uživatele.

Prvním krokem bylo rozhodnutí o vazbě funkcí "výběr" a "tisk". Některé známé banky dat slučováním těchto funkcí komplikují uživateli život; struktura dotazu

FIND / určení tisku /
IF / výběrová podmínka /

vypadá v praxi takto:

FIND JMENO , ROKNAR , PLAT
IF PLAT BT 2000 3000
END

(Pozn.: BT znamená ve většině jazyků "od-do").
Uživatel tedy musí mít již při dotazu kompletní představu o výsledku. V naší bance dat funkce oddělujeme. Po příkazu

VYBER / podmínka /

provede SŘBD výběr, nalezené věty zapamatuje a stručně oznámí výsledek výběru. Přeje-li si uživatel, může pak pomocí příkazu TISK pořídit z vybraných vět i několik výstupních sestav. Uživatel tedy zadá příkaz

VYBER PLAT 2000 AZ 3000

a SŘBD provede výběr a oznámí:

POCET PRACOVNIKU = 32

Nyní zadá uživatel

TISK JMENO , ROKNAR , PLAT

a získá sestavu. Může však dále požadovat na př.

TISK JMENO , ADRESA , UTVAR

a podobně; dokud nepřijde další příkaz VYBER, jsou vybrané věty neustále k dispozici.

Nové je použití češtiny. Podle principu jednoduchosti byly použity pouze operátory

= (rovné se) - lze vynechat

AŽ ("až" - určuje rozpětí).

Vůbec se nepoužívají operátory AND , OR , NOT a závorkování výrazů. Z logiky české věty lze totiž tyto operátory odvodit. Česká výběrová věta se převádí na tento interní obecný tvar:

[(A OR B OR ..)AND(C OR D..)AND..] OR [(E OR F..)AND..]OR ...

To postačí na zadání převážné většiny výběrových požadavků.

Česká věta je tedy dosti mocná, aby svou syntaxí vyjádřila poměrně složité logické uspořádání (aniž by to uživatele muselo zajímat). Upozorňujeme zde na dvojznačnost spojky "A" a čárky, které mají význam AND nebo OR podle souvislostí (a s tím si musí analyzátor poradit).

V příkazu VYBER používá uživatel názvy souborů, názvy položek, hodnoty položek a výše uvedené operátory. Je dovoleno libovolně používat spojky A a čárky. Celý příkaz má tvořit srozumitelnou českou rozkazovací větu.

Syntaktický analyzátor převede příkaz na popis množinových vztahů, podle kterého pak pracuje výběrový modul. Uživatel má možnost kdykoliv požádat o zobrazení vztahů v běžném množinovém vyjádření, aby mohl sledovat, jak systém příkazu porozuměl. Na př. příkaz

VYBER PRACOVNIKY VZDEL VS A USO , UTVAR MTZ A UKOLY
ROK 78 , 79 , MES=12

systém převede (a na požádání zobrazí) takto:

[(PRAC)&(VS∨USO)&(MTZ)] ∪ [(UKOLY)&(78∨79)&(PROSINEC)]

(Systém používá znaků ∪ pro sjednocení /OR/ a & pro průnik /AND/ množin).
V uvedeném příkladu vidíme dobře dvojznačnost á a čárky.

Systém má vestavěny prostředky pro práci s kořeny slov (lze tedy psát UKOL,UKOLY,UKOLU,UKOLUM atd) a některé další možnosti, na jejichž popis zde není místo.

Podle našich zkušeností lze tímto poněkud omezeným, avšak velmi jednoduchým způsobem sestavit většinu dotazů.

3.4 Tvorba výstupních sestav:

Jedním z největších problémů pro uživatele je tvorba výstupní sestavy. Při návrhu prostředků pro ovládání tisku narazíme na dva extrémy:

- formátuje systém; velmi pohodlné pro uživatele, grafická úroveň sestav je nevalná, někdy až příšerná;
- formátuje uživatel; možno předepsat libovolný tvar sestavy, uživatel však musí ovládat komplikovaný jazyk.

Programátoři dobře vědí, že ve všech programovacích jazycích jsou prostředky pro úpravu výstupních sestav velmi složité. Nutno předpokládat, že se uživatel odmítne něco takového učit. Po dlouhých rozborech jsme zvolili tuto cestu:

- položky zásadně formátuje systém
- uživatel může pouze vkládat mezery a odřádkování.

Uživatel se tedy naučí jen dva speciální symboly. Systém je vybaven možností postupné tvorby sestavy:

- uživatel předepíše část tiskového příkazu
- systém zobrazí (na obrazovce terminálu) takto zadanou část ve formě: /hlavička/příklad konkétní věty/měřítko/
- uživatel má nyní možnost
  - zrušit jeden nebo více prvků tiskového příkazu
  - přidat nové prvky

a tak se pokračuje, až je uživatel s vytvořenou sestavou spokojen. Povel k tisku pak způsobí výpis celé sestavy. Tímto způsobem vytváříme jednoduše i komplikované, graficky úpravné sestavy.

4. Základní funkce systému.

4.1 Doba odezvy - vyhledávání:

V terminálovém režimu je důležitým hlediskem doba odezvy systému. V literatuře se uvádí maximální odmlka systému 3 vteřiny, ale podle našeho názoru jde o údaj příliš extremistický. Přikláníme se spíše k názoru, že uživatel nesmí zůstat více jak 5 vteřin bez zprávy o činnosti systému. Jestliže tedy v naší bance dat zjistí SŘBD, že uživatelem vyvolaná činnost bude trvat déle než 5 vteřin, upozorní na to uživatele.

Aby však k takovému případu nedocházelo často, musí být v bance dat zabudována nějaká metoda rychlého výběru. Pro náš systém jsme použili jednoduché řetězení hodnot pro předem určené (klíčové) položky. Stejné hodnoty klíčové položky jsou propojeny pomocí směrníků (pointerů) a vytváří řetězec. Při vyhledávání se procházejí pouze věty, ležící v řetězci žádané hodnoty. Předpokládá to ovšem disky s přímým přístupem; dále je nutno efektivně ošetřit vyhledávání podle několika hodnot různých položek. Na př.: při analýze příkazu

VYBER PRACOVNIKY PROFESE PROGR VZDELANI VS

systém v seznamech zjistí, že programátorů (PROGR) je v bázi dat více než pracovníků s vysokoškolským (VS) vzděláním. Berou se tedy věty podle řetězce VZDELANI VS a z nich se vybírají ty, u nichž PROFESE = PROGR.

4.2 Doplňkové funkce:

Kromě funkcí "oživení","výběr","tisk" má náš SŘBD vestavěny ještě tyto doplňkové funkce:

- třídění a součtování na výstupních sestavách
- možnost skupinového výběru; tato možnost váže na relační princip báze dat a umožňuje vybírat podle návaznosti.

Na př. příkaz

VYBER UKOLY UTVARU MTZ (PRAC)

vybere všechny úkoly MTZ a ke každému z nich všechny pracovníky, kteří se na něm podílejí. Skupinový výběr lze dále prohlubovat, na př.

VYBER UKOLY UTVARU MTZ (PRAC VZDELANI VS)

(ke každému úkolu se vyberou pracovníci,kteří se jím zabývají a mají vzdělání VS). Věc je v podstatě jednoduchá, výklad by však přesáhl rámec tohoto příspěvku.
Další doplňkové funkce jsou:

- výpočty hodnot; při výběru i tisku lze zavést "novou položku", která se získá z existujících položek výpočtem.

  Př.: VYBER PRAC *PRIJEM=PLAT+PREM* 3000 AZ 5000

- tabulka výskytů: dialogovým způsobem se zadává dvourozměrná tabulka, na př.

	VZDEL		
UTVAR	SO	USO	VS
MTZ	5	8	2
PROGR	1	6	3

atd.

která má význam pro tvorbu přehledů. V tabulce lze součtovat a počítat procenta.

4.3 Uživatelský modul:

Základní SŘBD je stavěn obecně, aby pokryl běžné požadavky terminálového provozu. Praktické aplikace někdy vyžadují takové funkce, které by základním SŘBD působily potíže. V takových případech je nutno napsat uživatelské programy, které se přilinkují k SŘBD. Uživatelské programy pracují pouze s logickou strukturou dat a využívají podprogramů,které dává systém k dispozici.

5. Aplikace.

Na závěr uvedeme velestručně příklady aplikací, které byly pomocí našeho systému LIDA (little database) realizovány ve VAKUS Praha. Podrobný popis těchto aplikací by byl sice zajímavý, přesahuje však rámec tohoto příspěvku.

A) Informační systém o výstavbě ASŘ ve spojích:
Báze dat obsahuje údaje o kádrech, technice, stavbách a řešených úlohách ASŘ v rezortu spojů. Údaje se získávají z pravidelných hlášení spojových organizací a vkládají do báze dat pomocí uživatelského modulu.

Systém je určen pro pracovníky federálního ministerstva spojů. Terminál je umístěn v příslušném odboru ministerstva a napojen pomocí modemů přes komutovanou telefonní síť na počítač ve VAKUS Praha. S terminálem pracují přímo pracovníci ministerstva.

B) Evidence úkolů odboru:
Pro další využití terminálu na ministerstvu byla pomocí systému LIDA realizována banka dat o úkolech odboru. Evidují se přijaté i vydané úkoly včetně návazností; pomocí banky dat se sledují termíny, vyhodnocuje plnění a provádějí i potřebné rozbory.

C) Evidence analyticko - programátorských prací:
Poměrně rozsáhlý úsek ASŘ ve VAKUS Praha řeší analyticky i programátorsky množství úkolů ASŘ spojů. Pro evidenci, plánování a sledování řešitelských prací se využívá systému LIDA; v bázi dat jsou údaje
- o pracovnících
- o zakázkách (úkolech)
- o náběhu pracovních hodin na řešitelské práce
a strojových hodin na ladění programů.

Banka dat poskytuje informace vedoucím pracovníkům a plánovacímu útvaru, dodává požadované přehledy jednotlivým pracovním týmům a pomocí uživatelského modulu zajišťuje rutinní úlohy: stavovku úseku, fakturaci prací a pod.

K tomu nutno dodat, že ve všech aplikacích se užívá nezměněného systému LIDA, pro který byly v jednotlivých případech definovány potřebné struktury dat. Definiční systém LIDEF vytvořil tabulky a seznamy, se kterými obecný SŘBD LIDA pracuje.

6. Závěr.

Autor tohoto příspěvku, který si obvykle libuje v efektních a poučných závěrech je na rozpacích. Co říci o systému, který teprve nesměle vykročil do praxe? Co říci nového o bankách dat, o kterých již bylo popsáno mnoho papíru? Co říci efektního a poučného k problémům, které nejsou dosud teoreticky ani prakticky dořešeny a o kterých se ještě bude mnohokrát diskutovat?

Východiskem z ponurých úvah je metoda, používaná moderní literaturou: ponecháme otevřený konec a je na čtenáři, aby sám domyslel, k čemu to všechno je.

Našel-li čtenář v příspěvku něco zajímavého nebo dokonce podnětného, je spokojenost na straně autora. Jestliže tomu tak není, omlouvá se autor, že čtenáře zdržel a vyrušil z poklidné programátorské práce.

7. Literatura:

/1/ Chvalovský : Banky dat. SNTL/ALFA, 1976.

/2/ Codd : A Relational Model of Data for Large Shared Data Banks. Comm.ACM vol.13,No. 6,June 1970.

/3/ Calenko : Reljacionnyje modeli bazy dannych. Algoritmy i organizacija rešenija ekonomičeskich zadač,vyp. 9,10,11 ; Moskva,"Statistika",1977.

/4/ Martin : Computer Data-base Organizetion. Rusky: Organizacija baz dannych v vyčislitělnych sistémach. Moskva,"Mir",1978.

/5/ Bébr : Člověk rozmlouvá se strojem. Metody programování počítačů III.generace, sborník. DT ČVTS Ostrava, 1977.