ZÁPADOČESKÁ
UNIVERZITA
V PLZNI

Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

DIPLOMOVÁ PRÁCE

Plzeň, 2006

Markéta Vlášková

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

Návrh hybridního úložiště dat

Plzeň, 2006 Markéta Vlášková

Diploma Thesis

The Hybrid Data Warehouse Design

The aim of the project is hybrid data warehouse design above hospital database of patients and theirs clinical incidents. Project results should serve as information system for decision making support.

Thesis starts with description of data warehouse methodology and techniques of its creation and filling. Explanation of Online Analytical Processing, its options and dimensional modelling follow. One part is dedicated to analysis of existing source database, next part to dimension and cube design. Data warehouse creation process is described in own practical chapter. Oracle Enterprise Manager was used in this practical part of thesis. The use of Oracle Discoverer for getting required information out of data warehouse is described in last section. Some examples of particular data extraction, including numbers of clinical events and devices are presented.

Obsah

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracovala samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne …… … ……..…….

Markéta Vlášková

Poděkování

Děkuji vedoucí mé diplomové práce Doc. Dr. Ing. Janě Klečkové za odbornou pomoc, poskytnutí literatury a dat, Ing. Martinu Zímovi, Ph.D. za pomoc s instalací databázového serveru Oracle a Ing Kamilu Buriánkovi a Ing. Pavlu Jůzovi za poskytnutí praktických rad s vytvářením datového skladu a použití FrontEnd nástrojů.

Dále děkuji svým rodičům a přátelům, kteří mi pomáhali při studiu.

1. Úvod

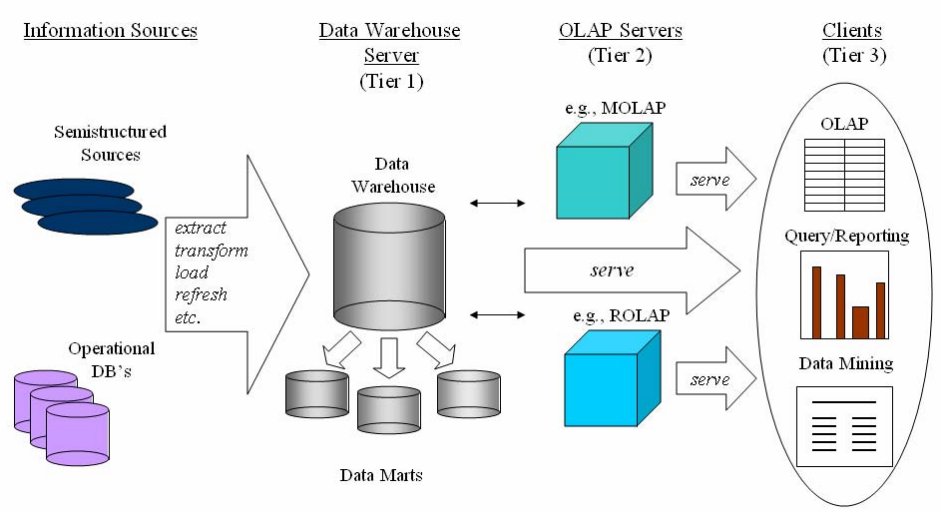
1.1. Popis problému

Technologie datových skladů představuje v současné době jeden z nejvýznamnějších trendů v rozvoji podnikových informačních systémů a to zejména vzhledem k velkému množství různorodých dat, které firmy skladují. Toto množství se neustále zvětšuje a data je potřeba nějakým způsobem uspořádat, aby se z nich daly získat informace.

Bill Inmon, jeden ze zakladatelů Data Warehousingu, definoval datový sklad jako integrovaný, subjektově orientovaný, stálý a časově rozlišený souhrn dat, uspořádaný pro podporu potřeb managementu.

Dá se tedy říci, že se jedná o ucelenou databázi optimalizovanou pro dotazování, analýzu a predikci dat. V datovém skladu jsou integrována a ukládána data jak z interních tak z externích zdrojů, jak data historická, která jsou nejprve upravena, tak data nová. Datový sklad je tedy pravidelně aktualizován, čištěn a zkvalitňován moderními metodami.

Toto určitým způsobem strukturované úložiště údajů je základním stavebním kamenem systémů pro podporu rozhodování. Cílem je umožnit vedoucím pracovníkům, manažerům a analytikům dělat kvalitnější a rychlejší rozhodnutí. Přínosem datového skladu je umožnění přístupu ke komplexním a předzpracovaným datům, získání relevantní informace včas a v požadované struktuře. Dále umožňuje rychle a jednoduše analyzovat data, vytvářet potřebné výkazy a přehledy, hledat odchylky od normálního průběhu, sledovat a porovnávat údaje za jednotlivé oblasti, sledovat časové řady vývoje a podobně.



Obr. 1.1.:[8] Schéma rozhodovacího stromu

1.2. Struktura diplomové práce

Tato první kapitola je věnována uvedení do problému a struktuře práce..

Druhá kapitola se věnuje detailnímu popisu teorie datových skladů, popisuje jeho strukturu, plnění i různé typy datových skladů.

Metodou OLAP se zabývá kapitola třetí. Podrobně rozebírá jednotlivé varianty, jejich přednosti i nedostatky. Nechybí zde ani vysvětlení pojmů jako je datová kostka, dimenze, tabulka faktů. V této kapitole je také několik názorných obrázků pro lepší porozumění problému.

Čtvrtá kapitola je zaměřena na produkty firmy Oracle, které byly při práci použity. Jedná se o databázi Oracle9i a Discoverer.

Analýza projektu, tedy popis zdrojových tabulek, návrh dimenzí, tabulek faktů a datových kostek je popsána v kapitole páté.

Praktickým návodem k implementaci datového skladu pomocí nástroje Oracle Enterpise Manager Console je kapitola šestá. Popisuje se zde krok po kroku vytváření datového skladu a práce s průvodci, které pro daný krok tento nástroj poskytuje.

Kapitola sedmá se věnuje nástroji OracleBI Discoverer. Obsahuje popis práce s tímto nástrojem a výsledné výstupy.

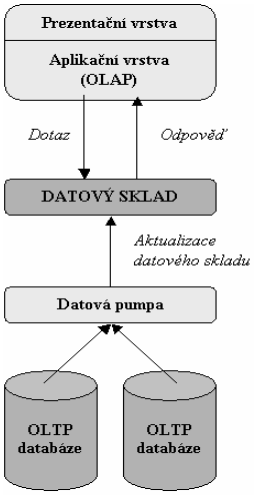
V poslední kapitole je shrnutí celé práce a uvedeny také osobní zkušenosti nasbírané během projektu.

Součástí práce je také CD, které obsahuje soubory exportované databáze a skripty pro vytvoření a zaplnění datového skladu.

2. Datové sklady

2.1. Definice

Pojem datový sklad je překladem anglického termínu Data Warehouse (DW). Ve třívrstvé architektuře data warehousu rozlišujeme tyto následující vrstvy [9]:



Obr. 2.1.: [9] Struktura datového skladu

a) spodní

Do této vrstvy patří server skladu, na kterém jsou uloženy relační databáze. Této vrstvě odpovídá položka *Datový sklad*.

b) prostřední

Tato vrstva zahrnuje OLAP server, který implementuje buď relační OLAP model, multidimenzionální OLAP nebo hybridní OLAP. Tato vrstva koresponduje s *Aplikační vrstvou*

c) vrchní

Vrchní vrstvu označujeme jako klienta. Obsahuje nástroje pro provádění dotazů a vytváření zpráv, analýzy a/nebo data miningové nástroje (analýzy trendu, predikce, apod.). Shoduje se s *Prezentační vrstvou*.

Nejprve je třeba rozlišit pojmy **Data Warehouse** a **Data Warehousing**:

Data Warehouse lze definovat jako jednoduchý, úplný a konzistentní sklad dat, ve kterém jsou uložena a organizována data, získaná z různých zdrojů (za různé zdroje mohou být považována data z interní systémů, relace mezi databázovými objekty, soubory, externí zdroje nebo objektové databáze) a vytvořená různými uživateli či používána v různých procesech.

Data Warehousing je návrh a implementace procesů, nástrojů a prostředků na data pro získání a doručení kompletní, včasné, přesné a srozumitelné informace pro rozhodování. To zahrnuje všechny aktivity, které umožňují tvorbu, správu a údržbu datového skladu. Je důležité si uvědomit, že Data Warehousing není produkt, ale proces.

Často si myslíme, že datový sklad je produkt nebo skupina produktů, které si můžeme koupit, aby nám pomohly v řešení našich otázek a ke zlepšení naší rozhodovací schopnosti. Datový sklad nám opravdu může pomoci získat odpovědi pro lepší rozhodování, ale to je jen jedna z jeho mnoha schopností. Využívá přístupu ke zdrojům heterogenních dat; čištění, zařazování a transformace dat; a ukládání dat do struktury, ke které se snadněji přistupuje, rozumí i se snadněji užívá. Data se potom používají pro dotazování, referování a analýzu. Přístup, užití, technologie a požadavky na provedení jsou absolutně jiné než pro transakčně-orientované operační prostředí (OLTP systémy). Množství dat v datových skladech může být velmi velké (až desítky terabyte), zvláště když uvažujeme nároky na analýzu historických dat.

2.2. Historie

Prvotní koncepce datového skladu je datována počátkem 80. let minulého století. Základem byl relační model a dotazovací schopnost byla poskytována SQL jazykem. Data byla extrahována z online databází společností a ukládána v nově vytvářených databázových systémech specializovaných na dotazování koncových uživatelů a reportování všech druhů. Funkce a účel datového skladu pro zpracování dat se od svých počátků značně vyvinul a stále se rychle vyvíjí.

Myšlenka datového skladu vznikla z potřeby jednoduchého přístupu ke strukturovanému úložišti kvalitních dat, která mohou být použita pro rozhodování. Informace je velmi mocným aktivem, které může poskytnout značné výhody každé organizaci a konkurenční výhody v obchodním světě. Organizace mají rozsáhlá množství dat, ale přístup k nim a jejich užití se s rostoucím množstvím stále stěžuje. Je to tím, že data jsou v různých formátech, existují na mnoha platformách a jsou uložena v mnoha různých souborech a struktury databází jsou vyvíjeny různými prodejci. Takovéto organizace musí zapisovat a uchovávat možná stovky programů používaných pro získávání, přípravu a sjednocování dat pro užití mnoha různých aplikací na analýzu a referování. Také je často potřeba hledat hlouběji v datech potom, co jsou počáteční vyhledávání provedena. To typicky vyžaduje modifikace extrakčních programů nebo vývoj nových. Tento postup je nákladný, nevýkonný a časově náročný. Data Warehousing nabízí lepší přístup.

2.3. Struktura datového skladu

Datový sklad je navrhován a konstruován na základě potřeb podniku jako celku. Může být fyzicky centralizován nebo fyzicky distribuován po organizaci (většinou geografická lokalizace). Fyzicky centralizovaný data warehouse je využíván celou organizací, je na jednom místě a spravuje jej oddělení Informačních Služeb (IS). Distribuovaný data warehouse také užívá celá organizace a také jej spravuje oddělení IS, ale data jsou rozdělena na několika místech. To, že IS spravuje datový sklad nemusí nutně znamenat, že datový sklad kontroluje. Pouze rozhoduje, která data se uloží, kdy bude sklad aktualizován, která další oddělení budou moci k datům přistupovat, kteří jedinci z těchto oddělení budou mít přístup atd.

Pro vytvoření vlastního datového skladu je třeba navrhnout odpovídající datovou strukturu. Návrh struktury je záležitostí zásadního významu - kvalita návrhu následně přímo ovlivňuje funkčnost vlastního datového skladu.

Struktura datového skladu, návrh tabulek dimenzí a faktů, definice počítaných položek a celá řada dalších činností, spojených s tvorbou vlastního datového skladu, je vždy opřena o kvalitní analýzu řešené problematiky u konkrétního uživatele. Ve zvýšené míře platí pro případ, kdy uvažujeme o širším využití datového skladu. Podstatnou částí se tak stává i analýza provozovaných aplikací, poskytujících například výstupní sestavy, a reportovacích nástrojů. Cíle analýzy níže popsané se vztahují zejména pro tuto práci. V praxi je jich mnohem více (struktura realizačního týmu, odpovědnost za financování, definování rolí apod.) a předchází IT audit.

1. Analýza požadovaných funkcí

Abychom získali takové údaje, které je třeba zanést do datových struktur datového skladu, je třeba:

- určit společně s uživatelem skutečně používané činnosti (výstupní sestavy apod.)
- v určených činnostech dohodnout provedení případných změn
- vytipovat a navrhnout funkce (příklady výstupních sestav apod.)

2. Návrh struktur datového skladu

Údaje získané při analýze požadovaných funkcí se musí odpovídajícím způsobem zanést do datových struktur. V případě již provozovaného datového skladu to znamená jak rozšíření na úrovni dimenzionálních (číselníkových) údajů, tak i doplnění či úpravu částí, které zajišťují plnění příslušných datových struktur (ETL nástroje). Při tvorbě nově vznikajícího datového skladu se zohledňují získaná data od samého počátku. Pokud dodržíme uvedený postup, bude zajištěno, že výše zmíněné funkce budou v rámci datového skladu plně podporovány.

2.4. Budování datového skladu

Nejdůležitějším krokem budování datové skladu je výběr metody jeho budování.

Metoda „velkého třesku“

Jak již název napovídá, tato metoda spočívá v realizaci implementace datového skladu během jediného projektu. Samotný proces vývoje nesmí trvat neúměrně dlouho, protože se mohou mezitím změnit jak požadavky uživatelů, tak technologie.

Výhodou této metody je kompletní propracovanost ještě před začátkem realizace.

Tato metoda se skládá ze tří etap:

- analýza požadavků podniku
- vytvoření podnikového datového skladu
- vytvoření datových trhů

Přírůstková metoda

Princip přírůstkové metody, jinak zvané evoluční, spočívá v budování datového skladu po jednotlivých etapách, tedy postupně. Nespornou výhodou je průběžné testování; když se částečné řešení skutečnými uživateli osvědčí, přidáme další část atd. Je tedy bezpečnější a není tak náročná na analýzu, jako metoda „velkého třesku“.

Budování datového skladu touto metodou je tedy iterativní proces v čase, který má neustálou spojitost mezi datovým skladem a potřebami uživatelů.

2.5. Plnění datového skladu

Plnění datového skladu má dvě části - prvopočáteční naplnění a pravidelnou (iterační) aktualizaci.

1. Prvopočáteční naplnění

Při prvopočátečním naplnění se do datového skladu ukládají provozní údaje z rutinně provozovaného informačního systému (ERP systémy), případně archivovaná data historická či převzatá z jiných dříve provozovaných informačních systémů, která jsou nejprve vyčištěna. Tato činnost bývá časově poměrně náročná a zejména v počáteční fázi je nutná maximální pozornost jak při plnění, tak poté při ověřování správnosti naplnění údajů.

2. Pravidelná aktualizace

Při aktualizaci dat v datovém skladu se provádí již "pouze" plnění přírůstků dat. Toto plnění by při správné realizaci nemělo ze strany uživatele vyžadovat jakékoli nároky na

činnost. Musí být zřejmé, jaké údaje, kdy a kým byly zpracovány, a zejména je nutno podchytit výskyt případných chybových či jinak problematických údajů pro jejich následnou kontrolu a zpracování.

Plnění datového skladu zajišťují datové pumpy. Jejich cílem je transformovat data ze vstupních do výstupních datových zdrojů, tzn. shromáždění z mnoha zpravidla nehomogenních a různorodých zdrojů z databází OLTP a naplnění datového skladu určitými informacemi. Je možné použít pro tyto procesy některý z mnoha nástrojů nebo napsat proceduru (např. v PL/SQL editoru).

V jiné terminologii se můžete setkat s nástroji a postupy ETL (Extraction, Transformation, Loading) nebo ETT (Extraction, Transformation, Transport).

Podíváme se blíže na jednotlivé etapy procesu:

Extrakce - výběr dat prostřednictvím různých metod

Transformace - ověření, čištění, integrování a časové označení dat

Loading - přemístění dat do datového skladu.

2.6. Typy datových skladů

Rozlišujeme tři modely datových skladů [9]:

Podnikový sklad (enterprise warehouse*)*

Podnikový sklad sbírá všechny informace o subjektech, které obklopují celou organizaci. Provádí integraci celopodnikových dat pocházejících obvykle z jednoho nebo více provozních systémů nebo od externího poskytovatele informací. Tato data zasahují do řady oborů. Obvykle obsahují jak hodnoty detailní, tak i sumarizované. Jeho velikost se může pohybovat od několika gigabyte až po stovky terabyte.

Datové tržiště (data mart)

Datové tržiště obsahuje pouze podmnožinu celopodnikových dat, která je určená pro specifickou skupinu uživatelů. Rozsah dat je omezen na určité vybrané subjekty. Např. v marketingovém tržišti jsou obsaženy informace týkající se zákazníků, zboží a prodejů. Tyto hodnoty bývají sumarizovány.

Tvorba datového tržiště se pohybuje v řádu týdnů. Podle zdroje získávání dat rozlišujeme data marty na nezávislé, jejichž data se získávají z provozních systémů nebo z externích informačních zdrojů a závislé, jimž jsou data dodávána z podnikového datového skladu.

Virtuální sklad (virtual warehouse)

Virtuální sklad je sadou náhledů na provozní databáze. Pro efektivnější provádění dotazů jsou některé náhledy na sumarizace provedeny před vznikem vlastního požadavku a uloženy. Virtuální sklad je snadné vytvořit, ale vyžaduje dodatečné kapacity na provozních serverech.

3. OLAP

3.1. Definice

OLAP (Online Analytical Processing) je technologie zpracování databáze na aplikačním serveru, která umožňuje uspořádat velké objemy dat tak, aby byla data přístupná a srozumitelná uživatelům zabývajícím se analýzou trendů (zejm. obchodních) a výsledků. Tyto databáze uspořádávají kategorie dat do skupin polí, nazývaných dimenze a úrovní podrobností a mohou být i několikadimenzionální. Je to tedy technologie používající multidimenzionální pohled na shromažďování dat pro poskytování rychlého přístupu ke strategickým informacím pro další analýzy.

Databáze podporující technologii OLAP jsou hojně využívány podniky pro objevování cenných trendů z data martů nebo datových skladů. Poskytují historický pohled na data. I když je technologie OLAP sama o sobě velmi silným nástrojem, opravdovou sílu získává teprve v kombinaci s data miningem a vhodnými FrontEnd nástroji.

OLAP je informační technologie založena především na koncepci multidimenzionálních databází. Jejím hlavním principem je několikadimenzionální tabulka umožňující rychle a pružně měnit jednotlivé dimenze, a měnit tak pohledy uživatele na modelovanou realitu (zejména ekonomickou).

3.2. Historie

Obliba OLAP vzrostla vzhledem ke stále se zvětšujícímu množství dat a zjištění, že analýzy mají pro podnikání velký přínos. Do poloviny devadesátých let minulého století byly OLAP analýzy vzhledem ke své nákladnosti prováděny téměř výhradně ve velkých organizacích. To se změnilo, jelikož větší prodejci databází začali zahrnovat OLAP do svých databází – Microsoft SQL Server se svým Analysis Services, Oracle s Express (DB) a Darwinem, IBM a jeho DB2.

3.3. Stručný přehled

Obvykle jsou data v podnicích distribuovaná do několika zdrojů a jsou navzájem nekompatibilní. Jsou uložena na různých místech a v různých formátech. Získání OLAP reportu jako třeba „Jaký produkt je zákazníky nejčastěji kupovaný v letech 2002 až 2005?“ by bylo velmi časově náročné. OLAP je tedy navržen tak, aby podával přehledné analýzy toho, co se stalo.

Popisuje data, která jsou uložena v poli a ne v ploché mřížce. Pole vypadá jako krychle, každá strana je dimenzí, která reprezentuje nějaký business faktor, jako například čas, produkt, množství nebo region. Krychle mohou být přeskupovány a otáčeny, aby byly vyzvednuty konkrétní srovnání a vztahy. Většina OLAP nástrojů umožňuje managerům dostat se na detailnější úroveň (drilling) nebo naopak získání obecnějších pohled.

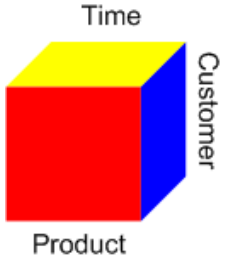
Základní vlastností OLAP je, že poskytuje uživatelům schopnost analyzovat data z jakéhokoli pohledu si oni přejí a nejsou nuceni spoléhat se na stanoviska předgenerovaná někým jiným (mohou vytvářet Ad hoc sestavy - uživatelské sestavy vytvořené koncovým uživatelem, říká se jim také náhodné dotazy; náhodné jsou z pohledu systému - ten neví s jakým dotazem ho budeme obtěžovat).

3.4. Datové krychle

OLAP nástroje jsou založeny na multidimenzionálním datovém modelu. Tento model zobrazuje data ve formě datové kostky. Datový sklad tedy nemá tabulky, ale kostky. Datová kostka je multidimenzionální reprezentace dat, která umožňuje rychlé vyhledávání a drillování.

Kostka má dimenze a míry.

Datová krychle tvořená z *m* atributů může být uložena jako m-dimenzionální pole. Každý element (prvek) pole obsahuje svou hodnotu míry. Pole je samo o sobě representováno jako 1-rozměrné pole. Nevýhodou přímého ukládání krychlí jako pole je, že většina krychlí je řídkých, takže pole obsahuje mnoho prázdných buněk (buněk s nulovou hodnotou).

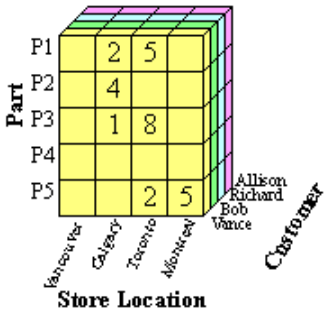


Obr. 3.1: OLAP Krychle s dimenzemi Customer, Product a Time

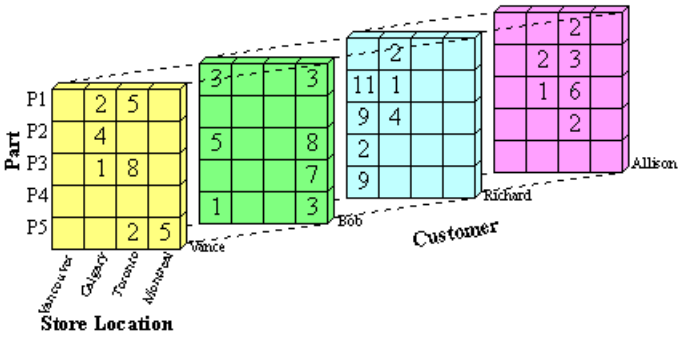
Krychle je používána pro reprezentaci dat dle zájmu. Ačkoli je nazývána „krychle“, může být 2-dimenzionální, 3- dimenzionální a více-dimenzionální. Každá dimenze reprezentuje nějaký atribut databázové tabulky a buňky v datové krychli mají hodnotu míry. Míry obsahují údaje za účelem agregace, například objemy prodeje, kdy se výpočet atributů objeví v databázi, nebo minimum, maximum, součet nebo průměr hodnot určitých atributů.

Příklad:

Máme databázi obsahující informace obchodní společnosti. Datová krychle vytvořená z této databáze je 3-dimenzionální - dimenze jsou part (část zboží), customer (zákazník) a store-location (obchod). Každá buňka datové krychle (p,c,s) představuje kombinaci každé z dimenzí. Ukázka datové krychle pro tuto kombinaci je na Obr.3.2. Obsah každé buňky je číslo, které znamená, kolikrát se ta určitá kombinace hodnot objevila společně v databázi. Prázdné buňky mají nulovou hodnotu. Datová krychle tak může být použita například pro získání informací, na základě kterých bude moci být rozhodnuto, kterému obchodu by měla být dána jaká část zboží, aby byl co největší prodej.



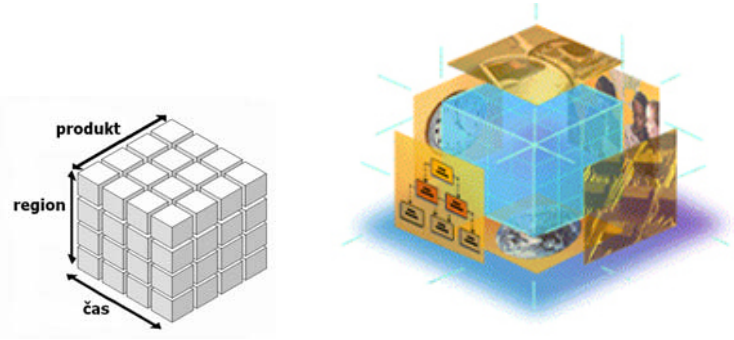
Obr. 3.2:[13] Čelní pohled na datovou krychli



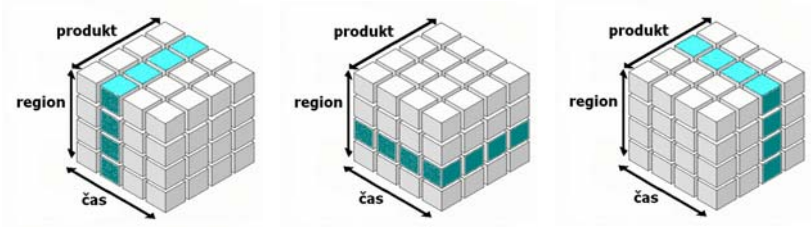
Obr. 3.3: [13] Celkový pohled na datovou krychli

3.4.1. Fakta a dimenze

Používání dimenzí může pomoci lépe pochopit obchodní data. Dimenze datové krychle reprezentují rozdílné kategorie pro analýzu dat. Kategorie jako například čas, geografické umístění nebo různé výrobkové řady jsou typickými dimenzemi v datových kostkách.



Obr. 3.4:[11] Dimenze krychlí



Obr. 3.5: [11] Řezy kostkou podle časové, regionální a produktové dimenze

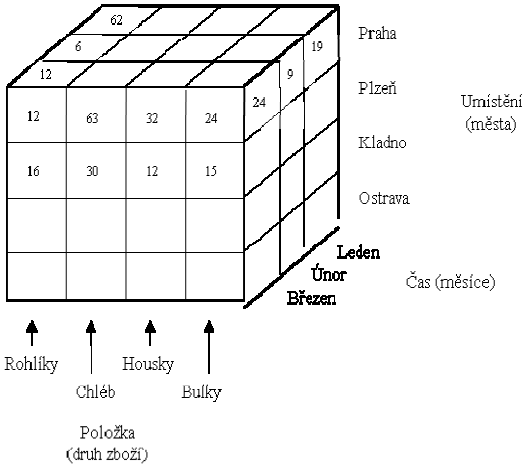
Dimenze jsou obvykle uspořádány do hierarchií tak, že mapují položky tabulek relačních databází. Hierarchie dimenzí jsou seskupovány do úrovní obsahujících hodnoty dané dimenze. Každá úroveň v dimenzi může být sumarizována, aby vytvořila hodnoty pro vyšší úroveň. Např. v dimenzi času sumarizací hodnot v úrovni den získáme hodnoty pro vyšší úroveň měsíc.

Multidimenzionální uložení dat je většinou realizováno na bázi metadatové nadstavby nad relačními tabulkami. Metadata přiřazují řádky a sloupce relačních databází jednotlivým dimenzím a buňkám v n-dimenzionální tabulce. V metadatech jsou také obsažena pravidla agregace dat na jednotlivých úrovních definovaných dimenzí. To je princip ukládání v OLAP technologiích.

3.4.2. Měrné jednotky (míry)

Míry jsou kvantitativní hodnoty v databázi, které mají být analyzovány. Typickými mírami bývají prodeje, náklady a rozpočty. Míry jsou analyzovány oproti různým kategoriím dimenzí datové kostky. Např. analýza prodejů (míra) určitého výrobku

(dimenze) v různých zemích (konkrétní úroveň dimenze geografická poloha) během dvou určitých roků (úroveň dimenze čas).



Obr. 3.6: [9] Míry krychle

Datová kostka reprezentuje data ve třech dimenzích. A to dimenze *Umístění*, *Času* a *Položky*. Aktuálně zobrazenou úrovní dimenze *Času* je úroveň *Měsíc*. U *Umístění* je to *Město* a u *Položky Druh zboží*. Mírou tohoto zobrazení jsou *Prodané kusy* (v tisících). Potom např. hodnota „12“ udává, že v Praze v měsíci Březnu bylo prodáno 12 000 kusů Rohlíků.

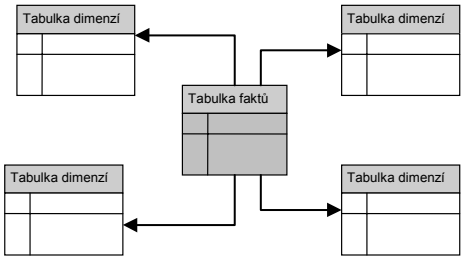
3.5. Schémata tabulek datového skladu

Aby datové sklady vyhovovaly potřebám uživatelů, mají jiné struktury i jiná pravidla spravování a trochu odlišný přístup k datům.

Star schéma (hvězda)

Star schéma je nejčastějším způsobem, jak převést relační model dat na multidimenzionální. Hvězdicové schéma se skládá z centrální tabulky s hodnotami, tzv. tabulka faktů a řadou doprovodných tabulek pro každou dimenzi. Grafické vyjádření schématu připomíná hvězdu, s tabulkami dimenzí zobrazenými v paprskovité struktuře okolo centrální tabulky faktů.

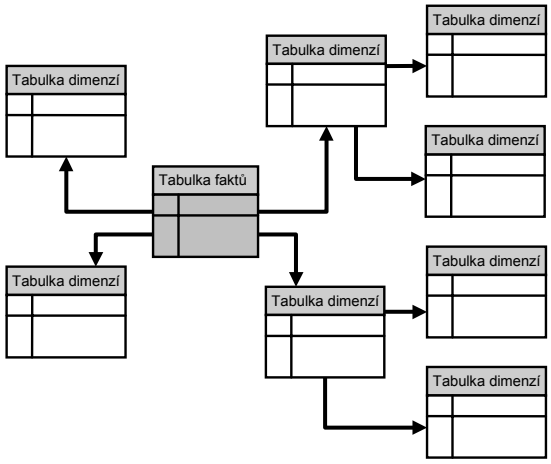
Ve hvězdicovém schématu je každá dimenze reprezentována právě jednou tabulkou. A každá tabulka obsahuje několik atributů. Např. dimenze „čas“ může mít tyto úrovně: den, měsíc, kvartál, rok.



Obr. 3.7: [11] Star Schéma

Snowflake (sněhová vločka)

Snowflake je určitým druhem hvězdicového schéma, ve kterém jsou tabulky dimenzí normalizovány, čímž se data rozdělují do dalších tabulek. Výsledné grafické schéma pak vytváří tvar podobný sněhové vločce. Hlavní rozdíl mezi těmito dvěma modely spočívá v tom, že tabulky dimenzí jsou normalizované, aby snížili redundance v uložených datech. Takováto tabulka je snadno udržovatelná a šetří diskový prostor. Ovšem tato úspora je zanedbatelná ve srovnání s typickou velikostí tabulky faktů. Navíc toto schéma může snižovat efektivnost analýz dat, neboť je zapotřebí provést více spojení tabulek, aby mohl být dotaz proveden. Proto může být výkon systému nepříznivě ovlivněn. Z tohoto důvodu není schéma sněhové vločky tak časté při návrhu datového skladu jako hvězdicové schéma.



Obr. 3.8: [11] Snowflake schéma

Fact Constellation

Aplikace mohou vyžadovat více tabulek faktů, aby mohly sdílet tabulky dimenzí. Toto schéma může být zobrazeno jako soubor hvězd a proto se nazývá „Constellation“ (galaxie nebo souhvězdí).

3.6. OLAP varianty

OLAP databáze představují jednu nebo několik souvisejících OLAP kostek. Ty většinou, na rozdíl od datových skladů, již zahrnují předzpracované agregace dat podle definovaných hierarchických struktur dimenzí a jejich kombinací.

Podle způsobu uložení dat rozlišujeme multidimenzionální OLAP, relační OLAP a hybridní OLAP.

Všechny tři možnosti poskytují určité přínosy, které záleží na velikosti databáze a na způsobu, jakým budou data využívána. Před zvolením si nějakého z následujících stylů OLAP je nutné zvážit požadavky uživatele. Vzhledem ke stále se snižujícím cenám hardwaru a zpracování, je nejčastěji používán MOLAP. HOLAP je lepším řešením pro nezávislé (samostatné) databáze. Zavedení ROLAPu je nejvhodnější, pokud jsou dotazovací nároky relativně nízké a také pro samostatné databáze.

3.6.1. Relační OLAP (ROLAP)

Při použití způsobu uložení dat ROLAP data zůstávají v původních relačních databázích - data ani agregáty nejsou ukládány do speciální databáze. Oddělená sada relačních tabulek je většinou použita k uložení agregací.

ROLAP je vhodný pro rozsáhlé databáze nebo na stará data, která nejsou často analyzována (ROLAP má velmi pomalé odezvy na požadovaná data). Pokud uživatel zadá požadavek na zobrazení dat, vygeneruje se SQL dotaz, který se přes rozhraní ODBC zašle do databáze primárního systému.

Přednosti:

- Přístup k datům v reálném čase.
- Vysoká datová kapacita (omezeno jen kapacitou databáze primárního systému).
- Přístup k detailním informacím.
- Automatická údržba dimenzí a mír.
- ROLAP nevyžaduje žádné speciální školení pro uživatele.

Nedostatky:

- Dlouhé odezvy - protože se nikde neukládají předpočítané součty (jako je tomu v MOLAP), musí se vše dopočítávat až při vykonávání SQL příkazu. Abychom toto eliminovali, musíme agregace udržovat např. ve speciálních agregovaných tabulkách.
- Nutná existence databáze - u ROLAP řešení se připojujeme k databázi primárního systému či datového skladu pomocí ODBC. V MOLAP postačí textový soubor a pod.

Použití:

Technologie ROLAP se většinou využívá ve spojení s datovými sklady. Může mít neomezený počet dimenzí v ukazateli, teoretická velikost ukazatele je také neomezená. Počet prvků v dimenzi se může pohybovat v jednotkách tisíc (cca do 10 000). Poněvadž se prvky v dimenzi vytváří až v době dotazu, je technologie ROLAP vhodná i pro aplikace s častými změnami, např. v číselnících primárních systémů (aplikace typu Saldo…).

3.6.2. Multidimenzionání OLAP (MOLAP)

MOLAP je multidimenzionální způsob uložení dat s vysokým výkonem. V tomto přístupu jsou data ukládána na OLAP server (ukládá data importovaná z primárních systémů do vlastní, multidimenzionální databáze (MDDB)). MOLAP poskytuje nejlepší výkon ve fázi dotazování (analýzy), neboť je právě pro multidimenzionální dotazy speciálně optimalizován.

MOLAP je nejrychlejší v získávání dat (zejména protože je možné indexovat přímo do struktury datové krychle pro výběr podmnožiny dat), ale zato vyžaduje nejvíce prostoru na disku. Diskový prostor již dnes není tak důležitý, vzhledem se snižujícím se skladovacím a zpracovávacím nákladům.

Bohužel není tento způsob vhodný pro velké množství dat a pro mnoho dimenzí. S rostoucím počtem dimenzí jsou krychle řidší - mnoho buněk reprezentujících určité kombinace atributů je prázdných, tzn. že neobsahují žádná agregační data. To vede k větším skladovacím požadavkům.

Přednosti:

- Velmi rychlá odezva na pokládané analytické dotazy (má předpočítané všechny agregace apod.).
- Možnost pracovat off-line (bez připojení k primárnímu zdroji dat).
- Bezpracná údržba agregátů .
- Data mohou vstupovat z různých datových zdrojů (TXT, DBF, ODBC…) (možné riziko chyb konverze), bez nutnosti speciální relační databáze.

Nedostatky:

- Nutnost dávkového zpracování.
- Nutná úprava zaváděcích skriptů při změně struktury dat v primárním systému či jakékoliv změně v dimenzích (číselnících).
- Práce off-line (přestože může být na jednu stranu výhodné prezentovat výsledky bez nutnosti připojení ke zdroji, na stranu druhou je nutné si uvědomit, že prezentovaná data jsou v daný čas vždy neaktuální.).

- Velikost ukládaných dat je omezena možnostmi a velikostí MDDB databáze. Varianta MOLAP není příliš vhodná pro zobrazování detailních dat (např. docházka zaměstnanců …).
- Drahé licence a nutnost kupovat HW.

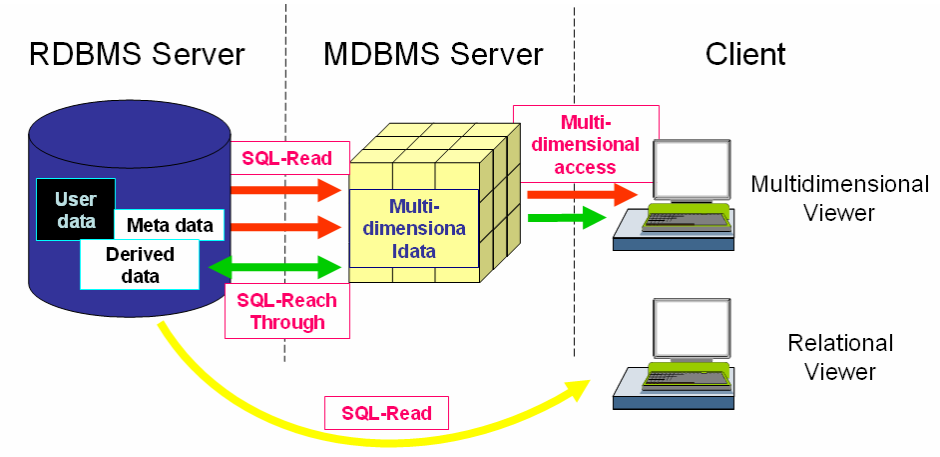
Použití:

Technologie MOLAP je vhodná pro malé až středně velké objemy dat, kdy není bezpodmínečně nutné vytvářet databázi datového skladu. Omezení velikosti počtu prvků pro daný ukazatel v MDDB nás také limituje v počtu dimenzí, podle kterých ukazatel sledujeme a v počtu prvků v jedné dimenzi, který by neměl přesahovat hodnotu 5000 prvků. Nehodí se také řešení, kde dochází k častým změnám v dimenzích (noví partneři, zboží …).

3.6.3. Hybridní OLAP (HOLAP)

HOLAP slučuje prvky z předešlých dvou přístupů a eliminuje nedostatky obou předchozích řešení. Základní technologií datového skladu je relační technologie a jisté často zpřístupňované výseky tohoto datového skladu (tzv. data marts) jsou duplicitně uloženy v datových krychlích implementovaných jako multidimenzionální databáze, které poskytují řádově rychlejší časové odezvy než základní relační struktura.

Tento způsob řešení nám umožní zvolit, jaká data se budou automaticky ukládat do MDDB a jaká ponecháme pro On-Line dotazování. Pro uživatele je pak přechod z MOLAP do ROLAP naprosto transparentní. V praxi to například znamená, že data za celý holding a sumy za jednotlivé společnosti se mohou ukládat do MOLAP a data detailní, jako střediska, jednotliví zaměstnanci apod., budou zobrazována On-Line pomocí technologie ROLAP.



Obr.3.9.: [10] Schéma HOLAP

Použití:

Technologie HOLAP najde uplatnění především u rozsáhlých projektů a datových skladů, kdy je žádoucí uskladnit data s vyšší mírou agregace do multidimenzionální databáze a data s vyšší mírou detailu je vhodné ponechat v relačních strukturách.

3.6.4. Kdy MOLAP, kdy ROLAP, kdy HOLAP?

Není to tak jednoduchá otázka, jak by se mohlo zdát. Velmi jednoduchá odpověď by mohla být: „Použít MOLAP dokud to bude pracovat a pak použít HOLAP.“. Něco pravdy na tom je. V zásadě MOLAP databáze (ve skutečnosti všechny multidimenzionální) jsou rychlejší a jednodušší na údržbu, jsou jen jedním souborem v souborovém systému. Řízení je vlastní, ne implicitní. Indexy jsou automaticky tvořeny a navrhovány tak, aby byly OLAP dotazy zpracovány co nejefektivněji. Omezení pro MOLAP je takové, že není tak dostupný jako ROLAP nebo HOLAP. Zatímco MDDB datová skladiště se zlepšují, je zde stále omezení co do množství dat, které je možné uchovávat.

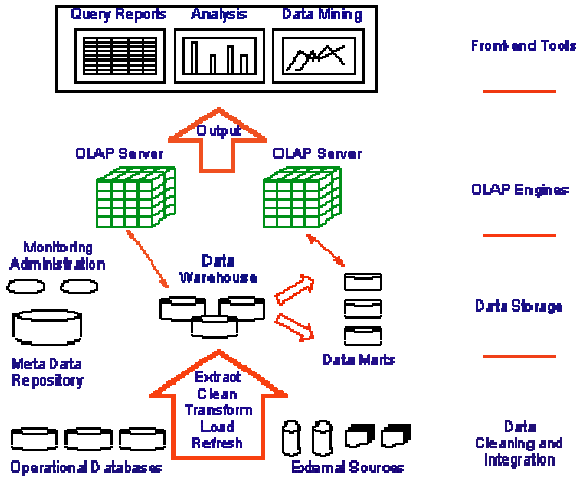
3.7. Základní operace v OLAP systémech

Důvodem budování datových skladů je snadná možnost provádět přijatelným způsobem analýzy nahromaděných dat. Nejzákladnějšími operacemi umožňujícími tyto analýzy jsou:

- roll-up (drill-up) - umožňuje uživateli v jedné či více zvolených instancích jisté agregační úrovně nastavit nižší (jemnější) agregační úroveň.
- roll-down (drill-down) - naopak od předešlé operace, ve zvolených instancích jisté agregační úrovně nastavuje vyšší (hrubší) agregační úroveň.
- slicing - dovoluje provádět řezy datovou kostkou, tj. nalézt pohled, v němž je jedna dimenze fixována v určité instanci (nebo více instancích) jisté agregační úrovně. Jinými slovy tato dimenze aplikuje filtr na instance příslušné agregační úrovně dané dimenze.
- dicing - je obdobou „slicingu“, jenž umožňuje nastavit takový filtr pro více dimenzí.
- pivoting - je jednoduchá, ale efektivní operace umožňující uživatelům OLAP vizualizovat hodnoty krychlí pochopitelnějšími a intuitivnějšími způsoby, tzn. dovoluje „otáčet“ datovou krychlí, tj. měnit úhel pohledu na data na úrovni presentace obsahu dat.

3.8. Fyzická architektura

Obecně můžeme datové sklady popsat tří-vrstvým modelem. Jak je zobrazeno na obrázku dole, informace je nejprve z provozního zdroje, poté je vyčištěna, transformována a uložena do datového skladu. Ačkoli není tento první krok částí datového skladu, je klíčovou aktivitou pro OLAP dodavatele. Často musí být data umístěná v různých heterogenních úložištích nejprve řádně zpracována, než mohou být uložena do skladu.



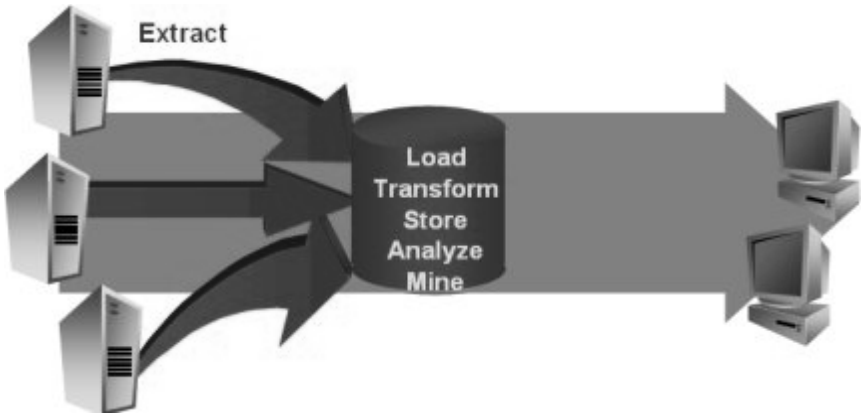
Obr. 3.9: Architektura OLAP

4. Oracle

4.1. Oracle9i Release 2

Pro svou práci jsem si zvolila verzi Oracle9i Release 2, která obsahuje kompletní podporu pro OLAP a je integrována i podpora pro dolování dat přímo do databázového serveru. Také sama o sobě umožňuje realizaci fáze ETL, což přispívá k zjednodušení budování datových skladů, a možnosti pohotovějších analýz pro podporu rozhodování a také ke zjednodušení data miningu.

Databáze Oracle9i a související vývojové nástroje pokrývají všechny základní potřeby návrhu, správy a provozu datového skladu. Součástí databáze je kompletní infrastruktura pro plnění datového skladu (ETL procesu) ze zdrojových systémů. Tak lze zajistit provoz datového skladu pouze prostředky databáze. Pokročilé návrhy databáze umožňují vyjádřit i složité manipulace s daty prováděné v rámci ETL procesu jednoduchým způsobem. Díky tomu je návrh ETL procesu snadnější a riziko chyb při provozu se snižuje. Vysoký výkon databáze v rámci ETL procesu vychází z maximální paralelizace a zřetězení (pipelineningu) všech prováděných operací. Zkracuje se tak výrazně doba potřebná k aktualizaci dat v datovém skladu a minimalizuje se její vliv na dostupnost všech souvisejících systémů.



Obr. 4.1.: [2] Oracle9i Release 2 - Schéma pro OLAP a dolování dat

Rozšířením databáze o prvky pro podporu rozhodování je umožněno naprostou většinu požadavků na analýzu dat provádět přímo v databázi bez nutnosti přesunu dat do specializovaných úložišť, což zjednodušuje návrh i provoz celého systému. Navíc lze pro základní analýzy využít standardní nástroje používané pro reporting a dotazování nad detailními daty a provozními systémy. Pro pokrytí pokročilých požadavků na analýzy je k dispozici OLAP úložiště úzce integrované s vlastní databází. Díky integraci

je pro uživatele při pokročilých analýzách zcela transparentní, zda jsou data uložena v relační databázi nebo v OLAP úložišti.

Typ instalace

Enterprise Edition - jde o kompletní instalaci. Spravuje data high-end aplikací včetně OLTP systémů, datových skladů a internetové aplikace, ve kterých se počítá se zpracováním velkého počtu transakcí. Poskytuje nejvyšší funkcionalitu a škálovatelnost pro nejnáročnější aplikace. Instalace zabere přibližně 2,86 GB místa na disku.

4.2. OracleBI Discoverer

Oracle Discoverer je univerzální generátor tiskových sestav a manažerských výstupů z datových struktur (DW, tabulky) v prostředí ORACLE. Pomocí Oracle Discoverer je možné realizovat i ty nejnáročnější požadavky vrcholového managementu v krátkém čase, vlastními silami a v požadované grafické úpravě.

5. Analýza

5.1. Popis tabulek a vazeb zdrojové databáze

Tabulky zdrojové databáze jsou pomocí prefixů logicky tříděny. Prefix `CI_` značí číselník, `KS_` tabulky s informacemi vztahujícími se ke klinickým situacím a `RP_` tabulky s údaji souvisejícími s pacienty.

- `CI_DIAGNOZY(DGKOD)` - kódy diagnóz (celosvětově platný číselník MKN10 - mezinárodní klasifikace nemocí ve verzi 10).
- `CI_DIAGTXT(PORADI)` - textové interpretace diagnóz.
- `CI_KLINODD(KLINODD_ID)` - číselník klinik nebo oddělení.
- `CI_PLATCI(PLATCE_ID)` - číselník plátců zdravotního pojištění.
- `CI_PRACOVISTE (PRACOV_ID)` - číselník pracovišť jednotlivých klinik a oddělení.
- `CI_PRISTROJE_SG(PRISTDG_ID)` - číselník používaných přístrojů.
- `CI_SPECUDAL (KOD_SPEC)` - číselník klinických událostí.
- `CI_TYPYUDAL (KOD_TYPU)` - číselník typů klinických událostí.
- `CI_UZIVATELE(UZIVATEL_ID)` - číselník uživatelů systému, tzn. lékařů.
- `KS_AMBPECE(AMBPECE_ID)` - informace k ambulantní péči.
- `KS_HOSPSTA(HOSPSTA_ID)` - data o hospitalizaci.
- `KS_KLINDG(KLINDG_ID)` - tabulka klinických diagnóz.
- `KS_KLINUDAL(KLINUDAL_ID)` - tabulka klinických událostí. Těmi mohou být buď hospitalizace nebo ambulantní péče.
- `KS_KLINUDAL_PRIST` - tabulka propojující tabulku klinických událostí a přístrojů, které byly použity.
- `KS_OSTDG(OSTRV_ID)` - tabulka ostatních (vedlejších diagnóz). Propojuje tabulku klinických událostí a diagnóz.
- `RP_ADRESY(ADRESA_ID)` - tabulka adres pacientů.
- `RP_PACIENTI(PACIENT_ID)` - základní informace o pacientovi.
- `RP_POJISTENI(POJIST_ID)` - údaje o pacientově pojistce.

Během návrhu byly vytvořeny následující tabulky:

- `BODY_CZK` - cena jednoho bodu v českých korunách.
- `CAS(CAS_ID)` - základní tabulka pro časovou dimenzi.
- `CI_UZIVATELE_ETL(UZIVATEL_ID)` - číselník uživatelů, který byl pomocí ETL procesu upraven (kontrola titulů před a za jménem lékaře pomocí tabulky `TITULY`, procedura je na přiloženém CD).
- `PRISTROJ_BODY(PRISTROJ_ID)` - tabulka obsahující bodové ohodnocení přístrojů.
- `PRISTROJ_FAKT` - tabulka faktů použitá pro návrh datové krychle `Pristroj_K` (viz. dále).
- `TITULY` - tabulka obsahující tituly, která byla použita při ETL procesu pro čištění údajů o lékařích.
- `UDALOST_BODY(KOD_TYPU)` - tabulka obsahující bodové ohodnocení klinických událostí.
- `UDALOST_FAKT(KOD_TYPU)` - faktová tabulka použitá pro návrh datové krychle `Udalost_K` (viz. dále).

5.2. Definice dimenzí a jejich mapování

Je třeba identifikovat, jaká data budou uživatelé prohlížet a v jakých úrovních. To určíme na základě požadavků na zpracování dotazů.

Na projekt bylo pohlíženo jako na návrh manažerského systému pro finančního vedoucího nemocnice. Toho nejvíce zajímá, kolik peněz a bodů bylo na co vynaloženo.

Po analýze zdrojových tabulek bylo rozhodnuto o zpracování údajů o klinických událostech a používaných přístrojích. Na tyto údaje se bude pohlížet z pohledu času, lékaře a pracoviště, kde se úkon odehrál. Pro návrh dimenzí byly použity číselníky ze zdrojové databáze.

Návrh dimenzí je třeba důkladně zvážit, jelikož se projeví na granularitě v tabulce faktů, tedy úrovni podrobnosti údajů faktů uložených ve faktové tabulce. Nízká granularita, tedy nízká úroveň detailu uložených dat, znemožňuje pracovat s detailními daty. Naopak vysoká granularita, tedy vysoká úroveň detailu dat, možnosti detailních analýz nabízí, ale na druhé straně znamená i vyšší nároky na diskový prostor datového skladu a výkon HW.

Všechny navržené dimenze mají pouze jednu hierarchii.

Je samozřejmě možné zpracovávat i jiné údaje, například by mohly být zajímavé počty pojištěnců u jednotlivých pojišťoven, konkrétní diagnózy a další.

5.2.1. Dimenze `CAS_DIM`

Časová dimenze bude pro nás tou nejdůležitější ze všech, proto se jí budeme zabývat jako první. Slouží k prohlížení dat podle času. Dimenze je vytvořena nad tabulkou `CAS`.

Často mívá časová dimenze dvě hierarchie, časovou a fiskální. Časová dimenze, která byla pro tento projekt navržena, používá pouze kalendářní řazení. Má čtyři úrovně:

- Rok
- Kvartál
- Měsíc
- Týden

V praxi (např. trhy akcií, kde se údaje ukládají dokonce po 30-ti sekundových intervalech) se mohou používat i úrovně den a hodina. Určitě by bylo možné pomocí této úrovně podrobnosti vytvořit zajímavé výstupy a informace, ale také by poté bylo nutné data do tabulky faktů ukládat po hodinách. Tímto způsobem by díky vysoké granularitě tabulky faktů vznikly i vysoké nároky na diskový prostor. Zejména z důvodu, že datové zdroje byly omezené, zvolila jsem výše uvedenou hierarchii.

Name	Datatype	Size
CAS_ID	DATE	7
DEN_NAZEV	VARCHAR2	9
POCET_DNI_V_TYDNU	NUMBER	1
DEN_PORADI_V_MESICI	NUMBER	2
TYDEN_PORADI	NUMBER	2
TYDEN_POSL_DEN	DATE	7
MESIC_CISLO	NUMBER	2
MESIC_POPIS	VARCHAR2	8
MESIC_POCET_DNI	NUMBER	
MESIC_POSL_DEN	DATE	7
MESIC_NAZEV	VARCHAR2	9
KVARTAL_POPIS	CHAR	7
KVARTAL_POCET_DNI	NUMBER	
KVARTAL_POSL_DEN	DATE	7
KVARTAL_CISLO	NUMBER	1
ROK_CISLO	NUMBER	4
ROK_POCET_DNI	NUMBER	
ROK_POSL_DEN	DATE	7

Obr.5.1.: Sloupce tabulky CAS

SQL příkaz na vytvoření časové dimenze:

```
CREATE DIMENSION CAS_DIM
  LEVEL ROK IS (CAS.ROK_CISLO)
  LEVEL KVARTAL IS (CAS.KVARTAL_CISLO)
  LEVEL MESIC IS (CAS.MESIC_CISLO)
  LEVEL DEN IS (CAS.CAS_ID)
  HIERARCHY KALENDAR
    (DEN CHILD OF MESIC CHILD OF KVARTAL CHILD OF ROK)
  ATTRIBUTE ROK DETERMINES
    (CAS.ROK_CISLO,
     CAS.ROK_POCET_DNI,
     CAS.ROK_POSL_DEN)
  ATTRIBUTE KVARTAL DETERMINES
    (CAS.KVARTAL_POPIS,
     CAS.KVARTAL_POCET_DNI,
     CAS.KVARTAL_POSL_DEN,
     CAS.KVARTAL_CISLO)
  ATTRIBUTE MESIC DETERMINES
    (CAS.MESIC_NAZEV,
     CAS.MESIC_POCET_DNI,
     CAS.MESIC_POSL_DEN,
     CAS.MESIC_POPIS)
  ATTRIBUTE DEN DETERMINES
    (CAS.CAS_ID,
     CAS.DEN_NAZEV);
```

5.2.2. Dimenze NEMOCNICE_DIM

Pro prohlížení údajů podle různých klinik, oddělení nebo pracovišť slouží dimenze nemocnic. Dimenze NEMOCNICE_DIM je namapována na číselníky klinik CI_KLINODD a pracovišť CI_PRACOVISTE.

Tato dimenze má dvě úrovně:

- Klinodd
- • Pracoviště

SQL příkaz na vytvoření dimenze NEMOCNICE_DIM:

```
CREATE DIMENSION NEMOCNICE_DIM
  LEVEL PRACOVISTE IS (CI_PRACOVISTE.PRACOV_ID,CI_PRACOVISTE.KLINODD_ID)
  LEVEL KLINODD IS (CI_KLINODD.KLINODD_ID)
  HIERARCHY NEMOCNICE
    (PRACOVISTE CHILD OF KLINODD
      JOIN KEY (CI_PRACOVISTE.KLINODD_ID) REFERENCES KLINODD)
  ATTRIBUTE PRACOVISTE DETERMINES
    (CI_PRACOVISTE.PRACOV_ID,
     CI_PRACOVISTE.TELEFON,
     CI_PRACOVISTE.VEDOUCI,
     CI_PRACOVISTE.EXIST_DO,
     CI_PRACOVISTE.EXIST_OD,
     CI_PRACOVISTE.ICP,
     CI_PRACOVISTE.ICZ,
     CI_PRACOVISTE.DRUH,
     CI_PRACOVISTE.ZKRATKA,
     CI_PRACOVISTE.NAZEV,
     CI_PRACOVISTE.CISLO)
  ATTRIBUTE KLINODD DETERMINES
    (CI_KLINODD.KLINODD_ID,
     CI_KLINODD.PRIMAR,
     CI_KLINODD.PREDNOSTA,
     CI_KLINODD.ZKRATKA,
     CI_KLINODD.CISLO,
     CI_KLINODD.NAZEV);
```

5.2.3. Dimenze PRISTROJ_DIM

Aby bylo možno prohlížet údaje podle užitých přístrojů, byla zavedena dimenze PRISTROJ_DIM, která je vytvořena nad číselníkem CI_PRISTROJE_SG.

Dimenze je tvořena jen jednou úrovní:

- Přístroj

SQL příkaz na vytvoření přístrojové dimenze:

```
CREATE DIMENSION PRISTROJ_DIM
  LEVEL PRISTROJ IS
    (CI_PRISTROJE_SG.PRISTSG_ID)
  ATTRIBUTE PRISTROJ DETERMINES
    (CI_PRISTROJE_SG.INV_CIS,
```

```
    CI_PRISTROJE_SG.TRIDA,
    CI_PRISTROJE_SG.VYROB_CIS,
    CI_PRISTROJE_SG.PRISTSG_ID,
    CI_PRISTROJE_SG.NAZEV);
```

5.2.4. Dimenze LEKAR_DIM

Na klinické události a přístroje je možné dotazovat se i podle lékaře, který je za její provedení či užití zodpovědný. Proto byla vytvořena tato dimenze, která je mapována na tabulku CI_UZIVATELE.

Jedná se opět o jednoúrovňovou dimenzi:

- Lékař

SQL příkaz na vytvoření dimenze LEKAR_DIM:

```
CREATE DIMENSION LEKAR_DIM
  LEVEL LEKAR IS
    (CI_UZIVATELE.UZIVATEL_ID)
  ATTRIBUTE LEKAR DETERMINES
    (CI_UZIVATELE.EXIST_DO,
     CI_UZIVATELE.EXIST_OD,
     CI_UZIVATELE.TITUL_ZA,
     CI_UZIVATELE.PRIJMENI,
     CI_UZIVATELE.JMENO,
     CI_UZIVATELE.TITUL_PRE,
     CI_UZIVATELE.LOGIN_NAME,
     CI_UZIVATELE.TYP,
     CI_UZIVATELE.CISLO,
     CI_UZIVATELE.UZIVATEL_ID);
```

5.2.5. Dimenze UDALOST_DIM

Jako poslední dimenze byla navržena UDALOST_DIM pro výběr údajů dle různých událostí. Dimenze je mapována na číselník CI_TYPYUDAL.

I tato dimenze je jednoúrovňová:

- Událost

SQL příkaz na vytvoření dimenze událostí:

```
CREATE DIMENSION UDALOST_DIM
  LEVEL UDALOST IS
    (CI_TYPYUDAL.KOD_TYPU)
  ATTRIBUTE UDALOST DETERMINES
    (CI_TYPYUDAL.KOD_TYPU,
     CI_TYPYUDAL.SKUPINA,
     CI_TYPYUDAL.NAZEV);
```

5.2.6. Tabulky faktů

Jelikož dimenze, podle kterých budou vyhledávána fakta, jsou již známé, může být navržena tabulku faktů. V tomto projektu budou vytvořeny dvě datové kostky, proto byly navrženy dvě tabulky faktů.

Hlavními atributy tabulek faktů budou peněžní a bodová ohodnocení klinických událostí a přístrojů. Protože tyto údaje nemohly být nemocnicí poskytnuty, byly tabulky naplněny fiktivními hodnotami. V tabulce `PRISTROJ_BODY` je bodové ohodnocení použití daného přístroje, v tabulce `UDALOST_BODY` bodové ohodnocení události a v tabulce `BODY_CZK` je cena jednoho bodu v českých korunách.

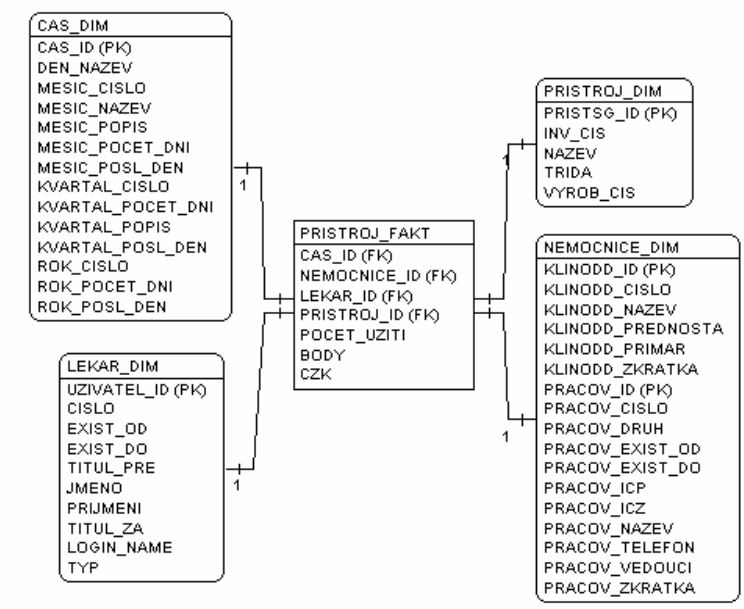
První tabulka `PRISTROJ_FAKT` bude mít uložena fakta ohledně používání přístrojů. Atributy této tabulky jsou `cetnost_uziti`, `body` a `CZK` a cizí klíče dimenzí `cas_id`, `nemocnice_id`, `lekar_id` a `pristroj_id`. Atribut `cetnost_uziti` nám říká, kolikrát byl přístroj použit, `body` vypovídají o bodovém ohodnocení užití daného přístroje a `CZK` o jeho finanční nákladnosti. Tato data jsou předem předpočítána z výše uvedených „fiktivních“ číselníků. Výpočet agregovaných údajů do tabulky faktů a jejich uložení zajišťuje proces ETL.

Všechny potřebné informace jsou obsaženy ve zdrojové tabulce klinických událostí `KS_KLINUDAL` a číselníku `CI_PRISTROJE_SG`.

Name	Datatype	Size
CAS_ID	DATE	7
NEMOCNICE_ID	NUMBER	15
LEKAR_ID	NUMBER	15
PRISTROJ_ID	NUMBER	15
CETNOST_UZITI	NUMBER	15
BODY	NUMBER	15
CZK	NUMBER	15

Obr.5.2.: Sloupce tabulky faktů `PRISTROJ_FAKT`
pro krychli `Pristroj_K` *(viz. dále)*

Granularitou této tabulky je četnost užití jednoho přístroje jedním lékařem na jednom pracovišti za jeden den. Takto jsou data uložena v tabulce.



Obr.5.3.: Tabulka faktů `PRISTROJ_FAKT` *, schéma dimenzionálního modelu (star)*

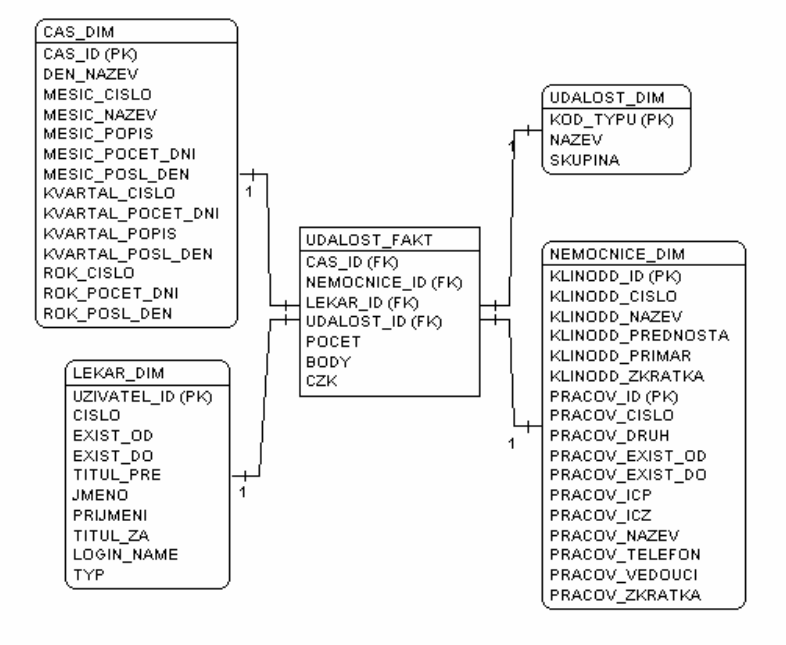
Druhá tabulku faktů byla nazvána `UDALOST_FAKT`. Její atributy jsou `pocet`, `body` a `CZK` a cizí klíče dimenzí `cas_id`, `nemocnice_id`, `lekar_id` a `udalost_id`. Atribut `pocet` nám říká, kolikrát se událost přihodila, `body` reprezentují bodové ohodnocení události a `CZK` je finanční ohodnocení události. Pro výpočet agregovaných dat do tabulky faktů platí stejná pravidla jako pro předchozí tabulku faktů.

Data potřebná pro zaplnění této tabulky faktů jsou uložena v `KS_KLINUDAL`:

Name	Datatype	Size
CAS_ID	DATE	7
NEMOCNICE_ID	NUMBER	15
LEKAR_ID	NUMBER	15
UDALOST_ID	VARCHAR2	3
POCET	NUMBER	15
BODY	NUMBER	15
CZK	NUMBER	15

Obr.5.4.: Sloupce tabulky faktů `UDALOST_FAKT`
pro krychli `Udalost_K` *(viz. dále)*

Granularita tabulky `UDALOST_FAKT` je „počet událostí u jednoho lékaře na jednom pracovišti za jeden den“.



Obr.5.5.: Tabulka faktů UDALOST_FAKT, schéma dimenzionálního modelu (star)

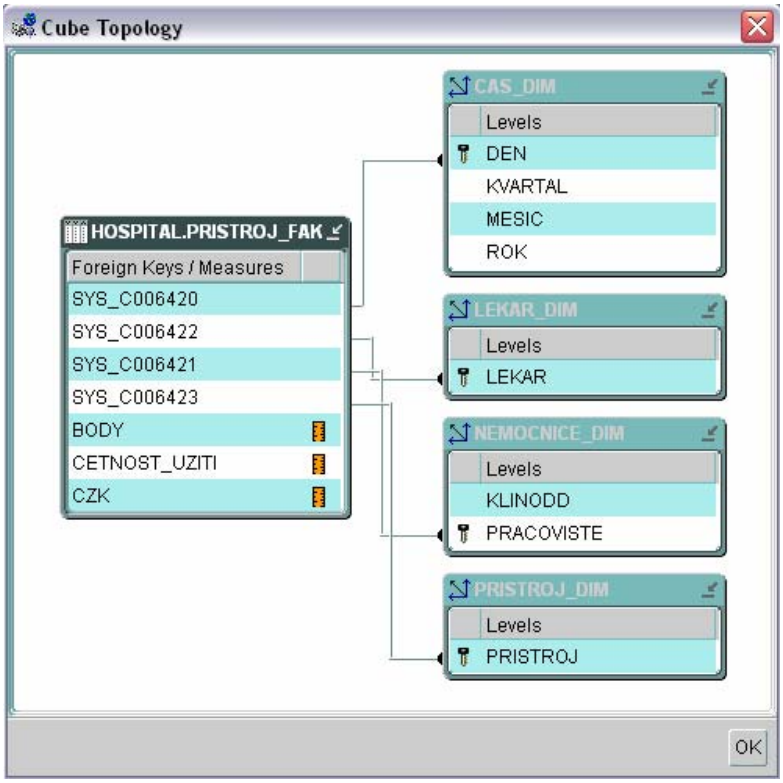
Poznámka: Skripty pro zaplnění tabulek faktů jsou uloženy na přiloženém CD.

5.3. Definice datových krychlí

Jak již bylo napsáno výše, krychle budou dvě a to `Pristroj_K` a `Udalost_K`. Obě krychle budou mít 4 dimenze.

5.3.1. Krychle přístrojová

Tato krychle byla pojmenována `Pristroj_K` a to proto, že bude uchovávat informace o užívání přístrojů. Tabulkou faktů je `PRISTROJ_FAKT` a používané dimenze jsou časová `CAS_DIM`, přístrojová `PRISTROJ_DIM`, dimenze lékařů `LEKAR_DIM` a nemocničních pracovišť `NEMOCNICE_DIM`.

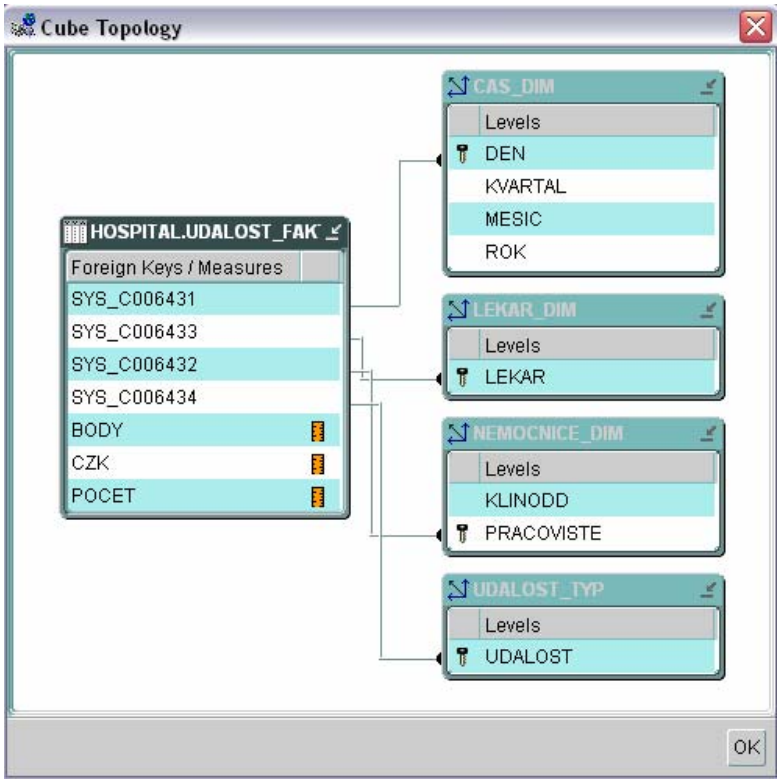


Obr. 5.6.: Topologie krychle `PRISTROJ_K`

Měrnými jednotkami této krychle jsou `CETNOST_UZITI`, `BODY` a `CZK`.

5.3.2. Krychle událostí

Krychle UDALOST_K uchovává souhrnné informace o různých typech událostí v nemocnici, jsou to různá vyšetření, operace a léčby. Pro tuto kostku je vytvořena faktová tabulka s názvem UDALOST_FAKT, obsahující kromě faktů klíče do dimenzí CAS_DIM, NEMOCNICE_DIM, LEKAR_DIM a samozřejmě v neposlední řadě UDALOST_DIM.



Obr. 5.7.: Topologie krychle UDALOST_K

Měrnými jednotkami krychle UDALOST_K jsou POCET, BODY a CZK.

5.4. Materializované pohledy

Projekt má být návrhem hybridního úložiště dat. To znamená, že detailní data zůstávají v relační databázi a spočítané agregace se ukládají do multidimenzionálních struktur. Datové krychle, dimenze a tabulky faktů jsou již navrženy, přístup do relační databáze bude realizován pomocí materializovaných pohledů.

Materializovaný pohled je databázový objekt, který obsahuje výsledky dotazu. Data, která byla vypočítána z detailních tabulek, uchovává ve fyzické tabulce. Pokud se data

v tabulkách změní, je možné pohledy obnovit s novými daty. Materializovaný pohled může dotazovat tabulky, pohledy nebo jiné materializované pohledy.

Rozlišujeme dva typy materializovaných pohledů a to materializovaný pohled s agregacemi, jehož užití je jasné již z názvu a materializovaný pohled obsahující pouze spojení. Výhodou vytváření tohoto typu materializovaných pohledů je předpočítání náročných propojení.

Vzhledem k tomu, že agregace budu mít předpočítané v OLAP kostkách, využiji druhého typu materializovaných pohledů.

6. Vytvoření datového skladu v OEMC

Nejprve byla pro vytváření a implementaci datového skladu zvolena aplikace Oracle Warehouse Builder 10.1. V průběhu práce s tímto nástrojem se vyskytly problémy s exportem do databáze a jelikož instalace Oracle Database9i Release 2 Enterprise Edition umožňuje návrh i správu datového skladu, byl celý návrh nakonec uskutečněn pouze za pomoci této databázové aplikace.

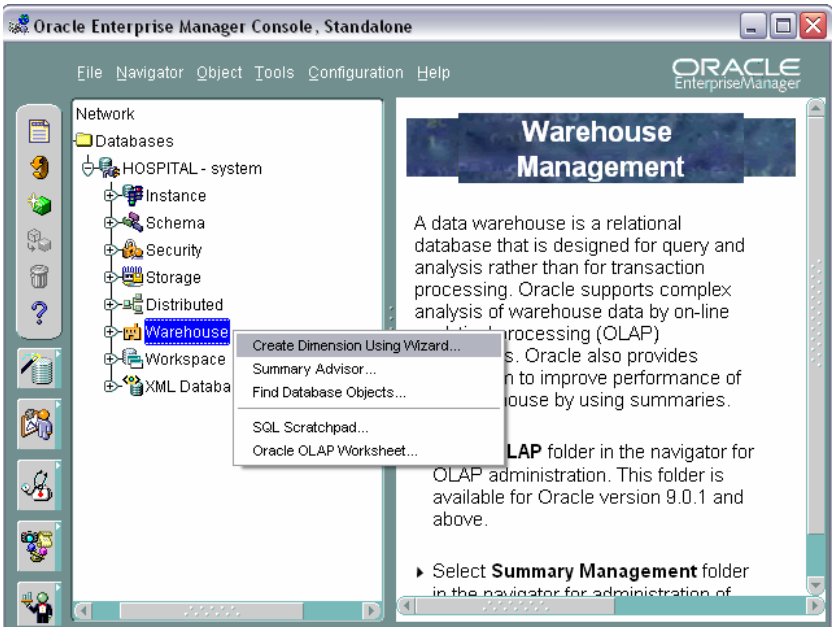
V této části bude popsán podrobně krok po kroku postup vytváření datového skladu, a to zejména pro neúspěch při hledání literatury popisující konkrétní postup. A tak tento text snad pomůže budoucím budovatelům datových skladů.

Jak během práce v Oracle Enterprise Manager Consoli poznáte, při vytváření jakéhokoli objektu Vás bude provázet průvodce (wizard), pokud si tak samozřejmě budete přát.

6.1. Vytvoření dimenzí

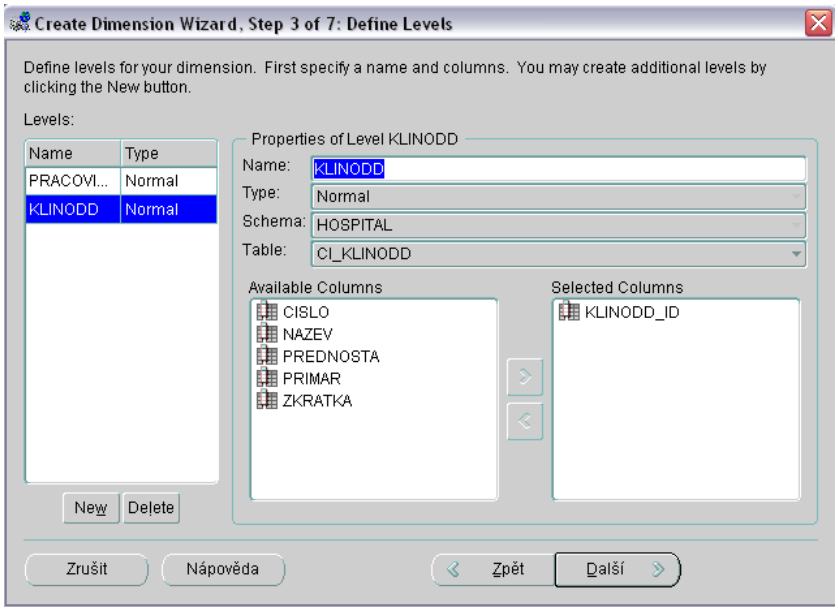
Velkou výhodou používání průvodce pro vytvoření dimenze, kromě značného usnadnění jejího návrhu, je její současné mapování.

Stisknutím pravého tlačítka na záložce *Warehouse* a výběrem položky *Create Dimension Using Wizard* (Obr. 6.1) spustíme *Create Dimensions Wizard*.



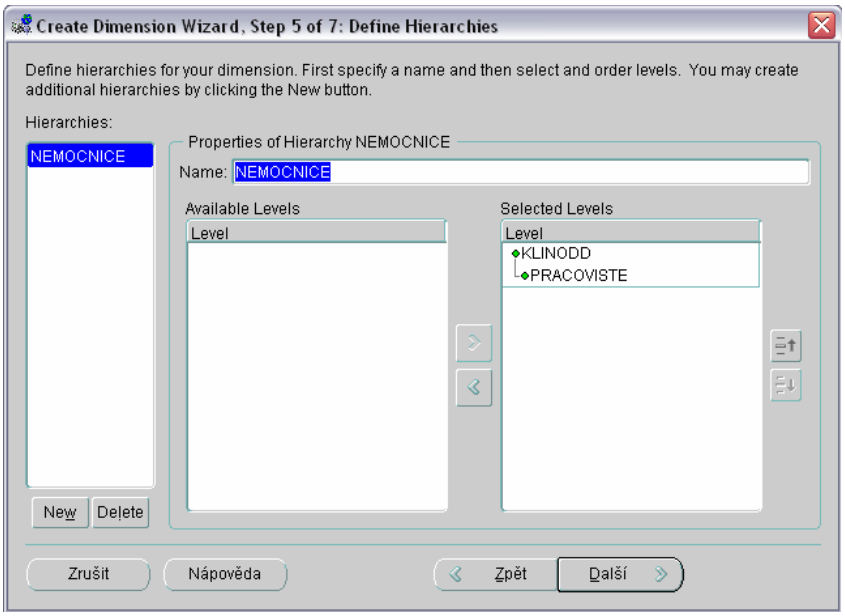
Obr. 6.1.: Spuštění průvodce vytvořením dimenze

V prvním okně průvodce si zvolíme, zda se bude jednat o dimenzi časovou či nikoli, podle toho průvodce přizpůsobí své další počínání. V okně druhém zadáme jméno dimenze a schéma pro její uložení. Dále se nadefinují úrovně a také si zvolíte tabulku, na kterou bude dimenze mapována (Obr. 6.2).



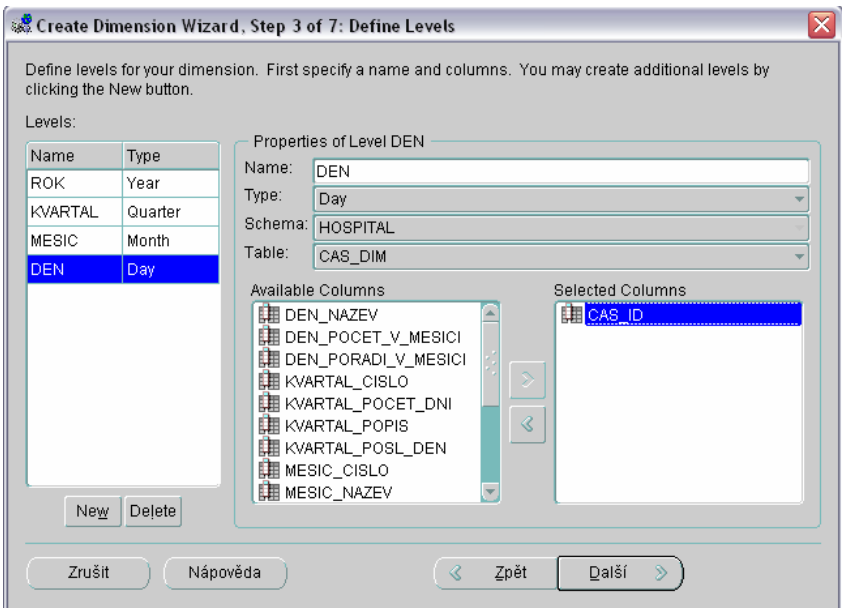
Obr. 6.2.: Definování úrovní a mapování

Následuje definování atributů úrovní, hierarchie (Obr. 6.3) a specifikace spojení úrovní.

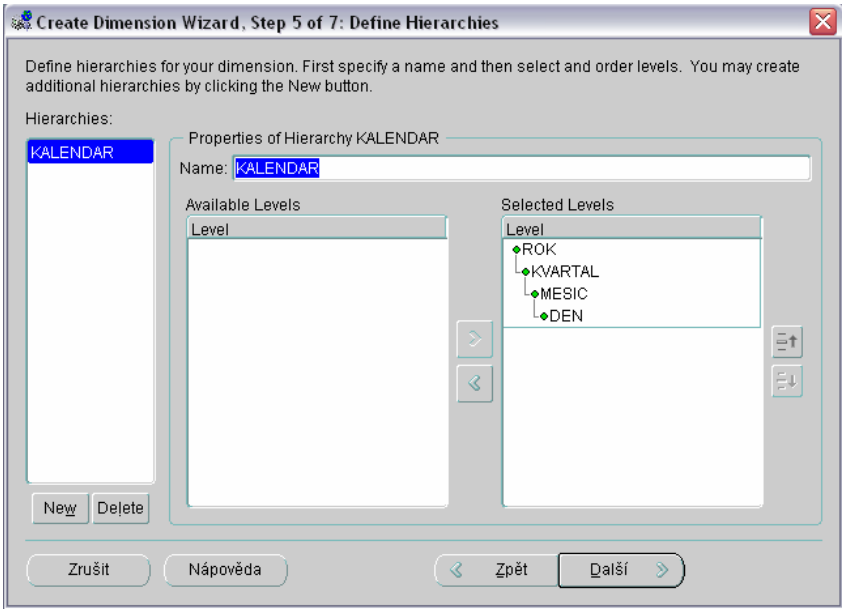


Obr. 6.3.: Definování hierarchie

Průvodce pro vytvoření časové dimenze se liší tím, že úrovně a některé atributy jsou předem dány. Máme samozřejmě možnost přidat další úrovně i atributy a změnit názvy stávajících.



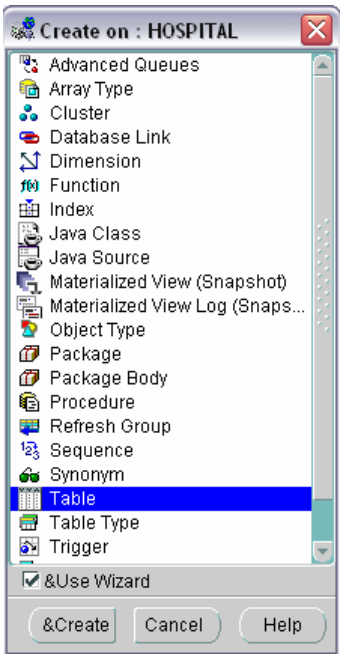
Obr. 6.4. : Definování úrovní a mapování časové dimenze



Obr. 6.5.: Hierarchie časové dimenze

6.2. Vytvoření tabulky faktů

Vytvoření tabulky faktů se nikterak neliší od vytvoření normální tabulky. Průvodce vytvořením tabulky se nazývá *Table Wizard* a provede nás 13-ti kroky. Spustíme ho výběrem *Object -> Create*, kde se otevře okno s nabídkou možností výběru objektů k vytvoření. V tomto případě nás zajímá možnost *Table*. V dolní části dialogového okna zaškrtneme možnost *&Use Wizard* (Obr. 6.6.).

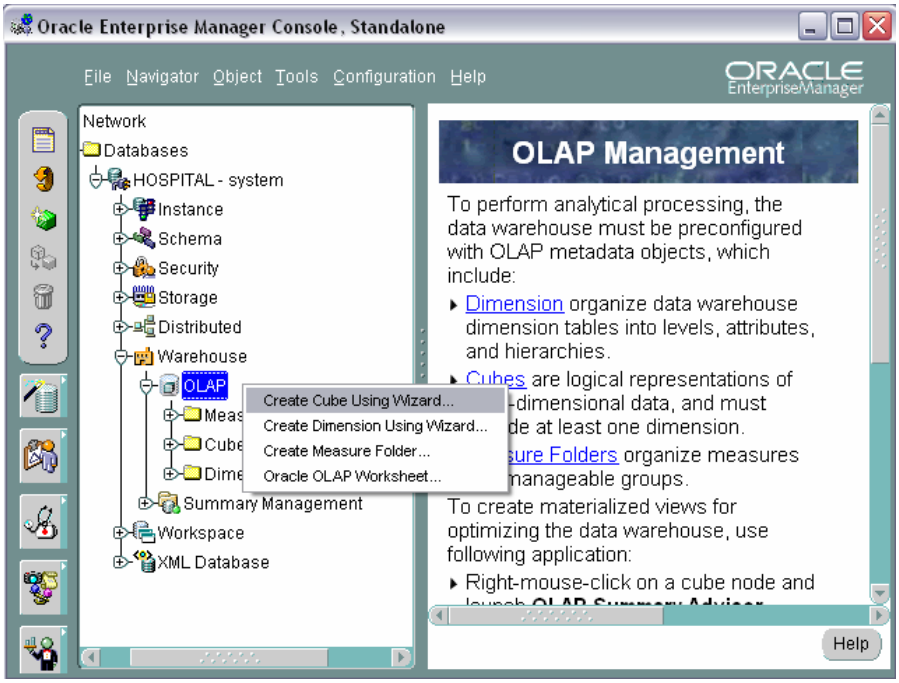


Obr. 6.6.: Dialogové okno s objekty pro vytvoření

Nejprve si zvolíme název tabulky a schéma pro uložení, nadefinujete jednotlivé atributy tabulky, zvolíte primární klíč, zda mohou být jednotlivé atributy null a zda musí být unikátní, cizí klíče, check constraints, podrobnosti uložení tabulky. Další možností skladování je fyzické uložení dat podle času a zvoleného rozsahu. V posledních krocích se tyto oddíly ještě více upřesní.

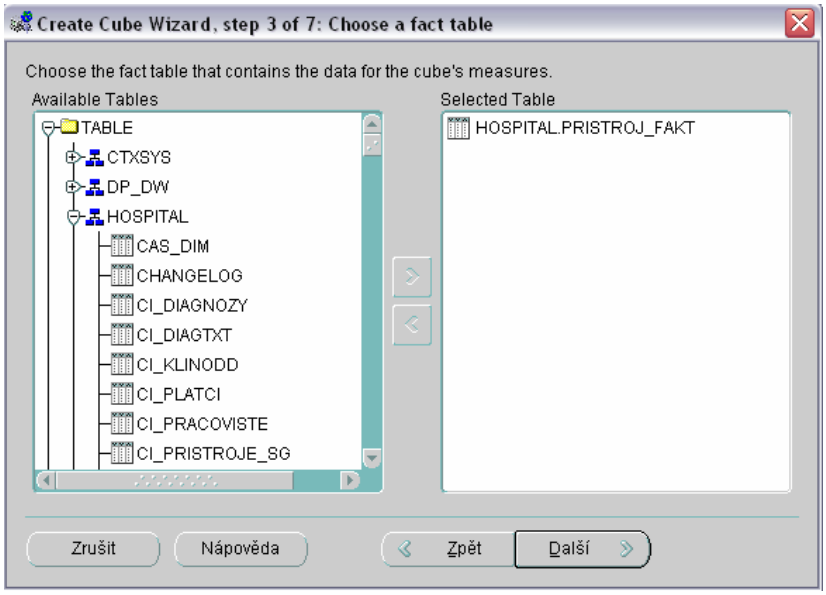
6.3. Vytvoření krychlí

I vytvořením datové krychle nás bude provázet průvodce. *Create Cube Wizard* spustíme kliknutím pravým tlačítkem myší na záložku *Warehouse -> OLAP* (Obr. 6.7.).



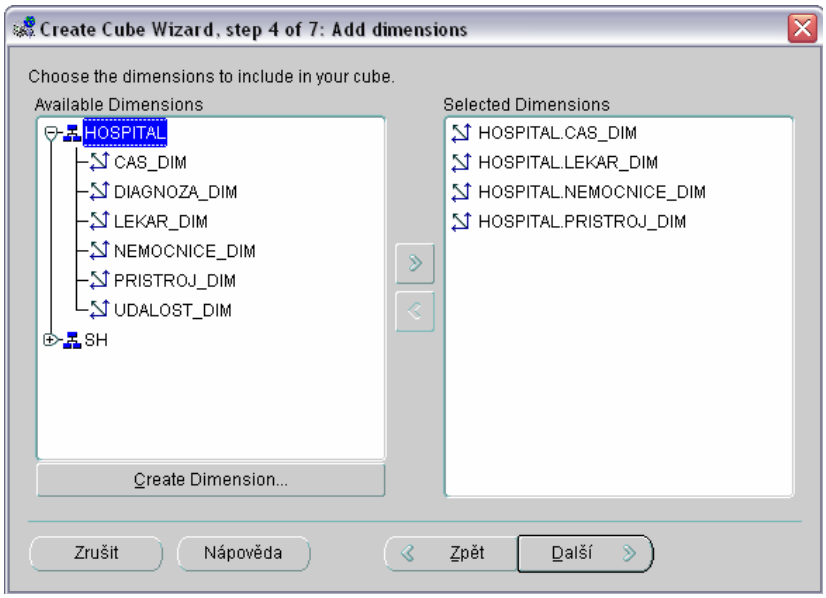
Obr. 6.7.: Spuštění průvodce pro tvorbu krychle

Tento wizard nás provede 7-mi kroky, kde nejprve zadáme název datové krychle, poté zvolíme tabulku faktů (Obr. 6.8.)



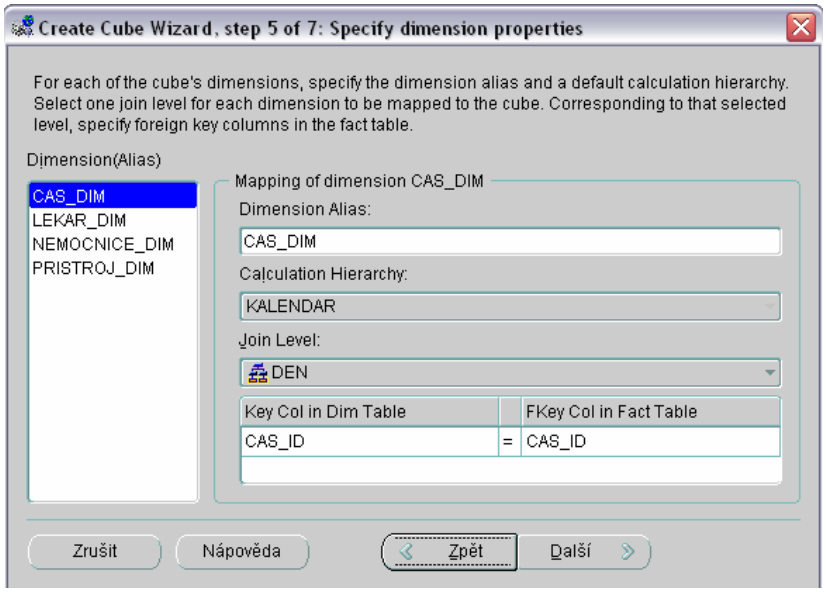
Obr. 6.8.: Výběr tabulky faktů

Na Obr. 6.9. je vidět výběr dimenzí pro danou krychli.

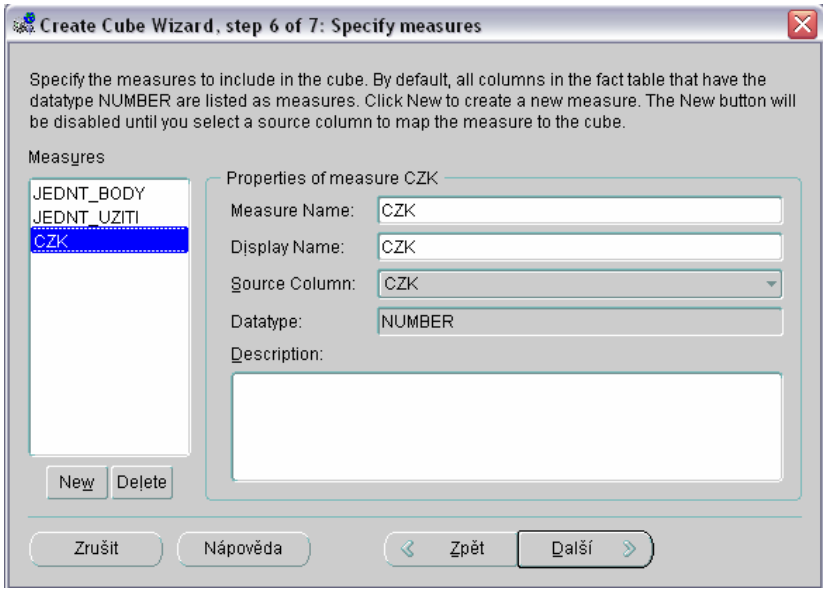


Obr. 6.9.: Výběr dimenzí

V dalších krocích se dimenze namapují (Obr. 6.10.) a nadefinují se míry (Obr. 6.11.)



Obr.6.10.: Mapování



Obr. 6.11.: Definice měrných jednotek

V posledním okně standardně zvaném *Summary* je možné zaškrtnout checkbox pro spuštění *Summary Advisor Wizard* pro optimalizaci krychle. Bohužel má příliš velké nároky na paměť (někdy i 3GB RAM jsou málo) a tak se nepodařilo ho spustit.

Pro multidimenzionální přehled všech možných kombinací dle vybraných dimenzí existuje v SQL klauzule `CUBE`, která rozšiřuje možnosti příkazu `SELECT`. Můžeme pomocí ní vykonat analýzu a sumarizaci údajů v tabulce současně podle více kritérií.

Syntaktický předpis:

```
SELECT ...  GROUP BY CUBE(seznam_seskupených_sloupců)
```

Příklad:

```
select k.PROV_PRAC_ID as Pracoviste, k.KOD_TYPU as
TypUdalosti, k.PROV_UZV_ID as Lekar, count(k.KLINUDAL_ID)
as pocet
from KS_KLINUDAL k
where k.PROV_PRAC_ID = 8
       and k.DATUM_PROV between '1.1.2005' and '31.3.2005'
group by cube (k.PROV_PRAC_ID, k.KOD_TYPU, k.PROV_UZV_ID);
```

PRACOVISTE	TYP	LEKAR	POCET
			15
		6	13
		8	2
	010		4
	010	6	3
	010	8	1
	041		6
	041	6	6
	095		5
	095	6	4
	095	8	1
8			15
8		6	13
8		8	2
8	010		4
8	010	6	3
8	010	8	1
8	041		6
8	041	6	6
8	095		5
8	095	6	4
8	095	8	1

22 řádek vybráno.

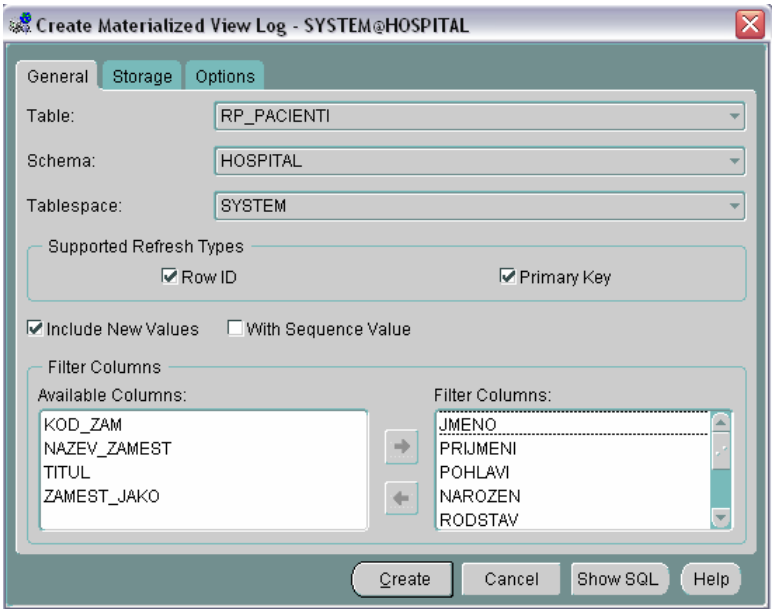
6.4. Vytvoření materializovaných pohledů

Před vytvořením materializovaného pohledu je nutné nejprve vytvořit protokol materializovaného pohledu pro všechny tabulky, nad kterými bude materializovaný pohled vytvořen.

Protokol vytvoříme pomocí SQL příkazu:

```
CREATE MATERIALIZED VIEW LOG ON název_tabulky
```

I pro tento úkon je v Oracle9i připraven průvodce, který spustíme v OEMC výběrem položky *Object → Create* a zvolením *Materialized Views Logs (Snapshot Logs)*. Okno průvodce má tři záložky. V *General* se vybírá tabulka, nad kterou bude log vytvořen a mohou zde být vybrány i konkrétní položky tabulky. Ve *Storage* je možnost zadat požadavky pro uložení materializovaného pohledu a v *Options* máme možnost ovlivnit logování a vyrovnávací paměť.

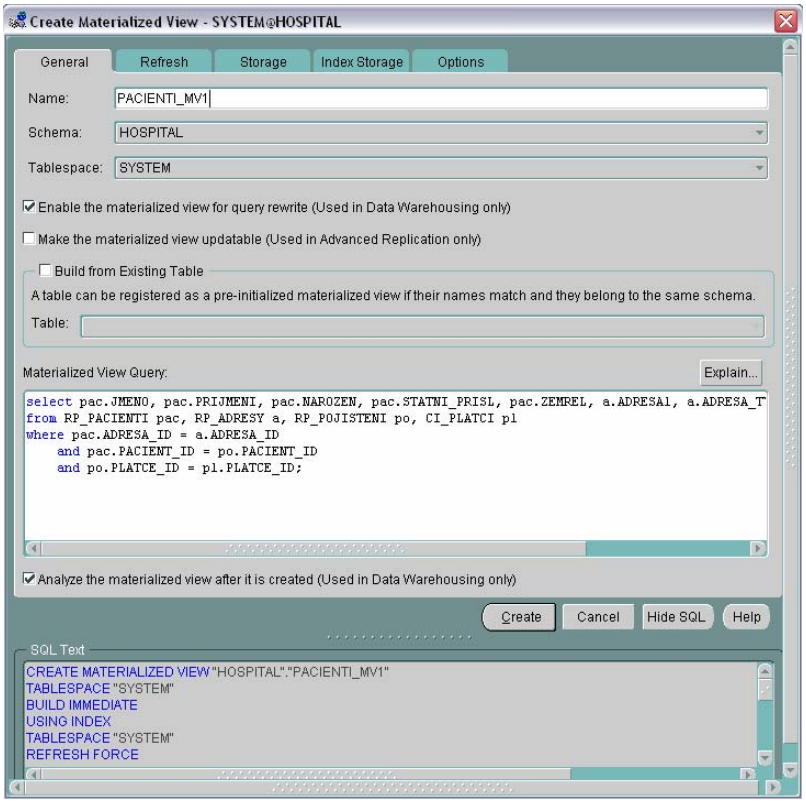


Obr. 6.12.: Vytvoření protokolu materializovaného pohledu

Průvodce pro vytvoření materializovaného pohledu se spouští položkou *Object → Create → Materialized Views (Snapshot)*.

Okno se otevře v záložce *General*, kde zadáme název materializovaného pohledu a do okna *Materialized View Query* vepíšeme SQL dotaz, jehož výsledky budeme chtít v materializovaném pohledu udržovat (Obr. 6.13.). V *Refresh* vybíráme, jakým způsobem budeme chtít materializovaný pohled obnovovat (*Complete* - kompletní obnovení pohledu, *Fast* - postupná aplikace změn dat, *Force* - implicitní možnost, Fast refresh pokud je to možné, jinak kompletní obnovení) a kdy obnovení budeme provádět (*On each commit* - obnovení proběhne vždy, když se změní jakákoli tabulka užívaná materializovaným pohledem, *On demand* - obnovení je spouštěno manuálně, poslední možností je zadání přesného data pro zahájení obnovování).

Zde je jako názorná ukázka zvolen jednoduchý příkaz na vypsání pacientů s iniciály, adresou a pojišťovnou, u které jsou pojištěni.



Obr. 6.13.: Vytvoření materializovaného pohledu

SQL skript takto vytvořeného materializovaného pohledu pak vypadá takto:

```
DROP MATERIALIZED VIEW HOSPITAL.PACIENTI_MV1;

CREATE MATERIALIZED VIEW HOSPITAL.PACIENTI_MV1
TABLESPACE SYSTEM
NOCACHE
LOGGING
NOPARALLEL
BUILD IMMEDIATE
REFRESH FORCE ON COMMIT
WITH ROWID
ENABLE QUERY REWRITE
AS
select pac.JMENO, pac.PRIJMENI, pac.NAROZEN, pac.STATNI_PRISL,
pac.ZEMREL, a.ADRESA1, a.ADRESA_TYP, a.TELEFON, pl.KOD_PLATCE
from RP_PACIENTI pac, RP_ADRESY a, RP_POJISTENI po, CI_PLATCI pl
where pac.ADRESA_ID = a.ADRESA_ID
        and pac.PACIENT_ID = po.PACIENT_ID
        and po.PLATCE_ID = pl.PLATCE_ID;
```

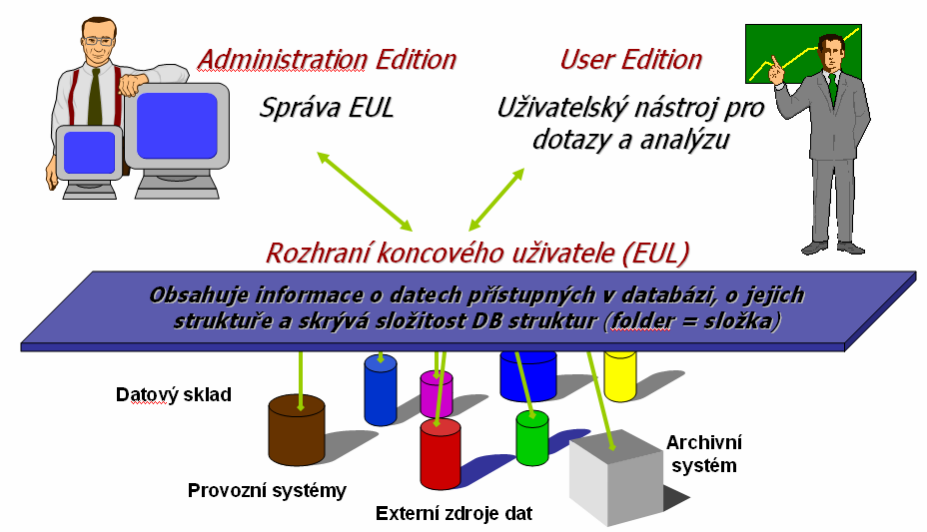
7. OracleBI Discoverer

Cílem projektu nebyla práce v OracleBI Discovereru, proto zde máme pouze běžné seznámení s tímto nástrojem. Byl zde použit hlavně pro prezentaci výsledků projektu.

OracleBI Discoverer je součástí balíku Oracle Business Intelligence a OracleBI Discoverer Administrator je v Oracle Business Intelligence Tools v řešení Oracle Application Server 10g, ale je možné zakoupit Oracle Business Intelligence jako samostatný produkt.

OracleBI Discoverer je integrovaný, intuitivní a interaktivní nástroj pro koncové uživatele, kteří potřebují snadno přistupovat k informacím (nikoliv jen k datům). Poskytuje kompletní podporu od tvorby reportů a jejich doručení až po jejich konečnou prezentaci. OracleBI Discoverer umožňuje vytvářet a editovat, analyzovat a přizpůsobovat, rozesílat a sdílet reporty vytvořené jak nad multidimenzionálními (OLAP kostkami) tak nad relačními datovými zdroji (OLTP systémy).

Komponenta OracleBI Discovereru umožňující jednoduchý přístup k relační i multidimenzionální databázi se nazývá OracleBI Discoverer Plus OLAP.



Obr. 7.1.: Základní pojmy

7.1. End User Layer (EUL)

EUL je rozhraní pro koncového uživatele (je tedy integrována jako čtyřvrstvé řešení systému), kde se spravují a tvoří metadata.

End User Layer izoluje uživatele od fyzického schématu databáze a poskytuje intuitivní pohledy do databáze, které mohou být uzpůsobeny každému koncovému uživateli nebo

skupině uživatelů. Cokoli uživatel udělá pomocí Discovereru, ovlivní to pouze metadata v EUL a ne v databázi (!!!).

EUL je kolekce přibližně 50 tabulek v databázi. Tyto tabulky jsou jediné tabulky, které může Discoverer Administration Edition v databázi změnit. Tabulky slouží k definování business areas (dále BA).

Na vytvoření EUL budete dotazováni hned po připojení se do databáze v Discoverer Administration Edition, pokud dosud žádná neexistuje. Spouštět Discoverer User Edition nemá tedy žádný smysl, pokud ještě neexistuje EUL vrstva.

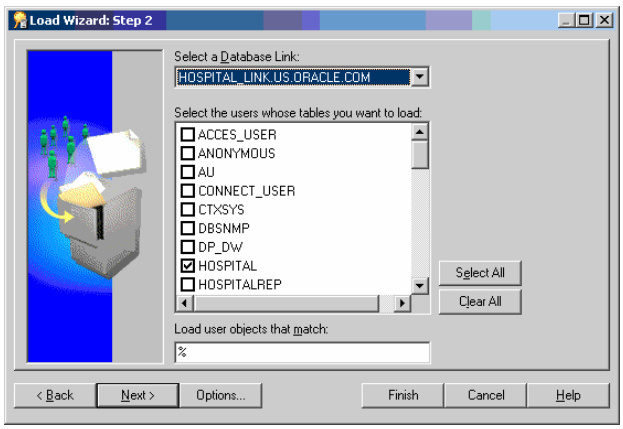
7.2. Business Area (BA)

Většinu uživatelů nezajímají data v databázi jako celek, ale zajímají se o část týkající se jejich práce. Pomocí Discoverer Administration Edition může být vytvořena jedna nebo více business area obsahující související informace.

BA je seskupení tabulek, pohledů nebo jen určitých položek tabulky navržené podle specifických požadavků jednotlivých uživatelů. Ty jsou koncovým uživatelům prezentovány jako složky (folder) a atributy tabulek jako položky (item). Uživatel může vytvářet vlastní složky, obsahující sloupce z více tabulek a to 4-mi způsoby:

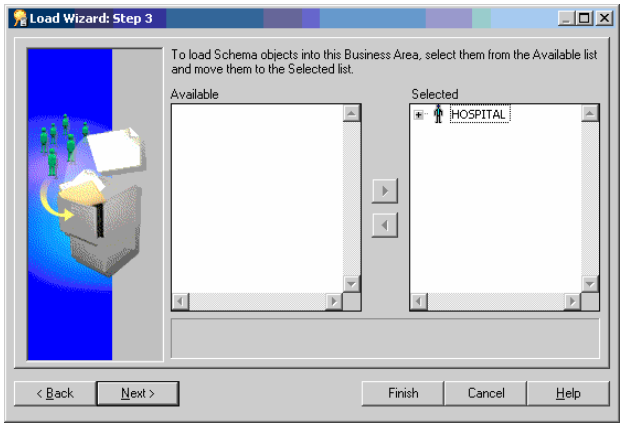
1. prostým mapováním z fyzické databáze (folder)
2. sloučením více folderů do jednoho (complex folder)
3. přesným SQL dotazem (SQL folder)
4. kombinací předcházejícího (complex folder)

BA je možné si vytvořit jak hned po vytvoření EUL, tak kdykoli jindy. Nejprve je potřeba si vybrat uživatele, jehož tabulky budete chtít do BA nahrát (Obr. 7.2.) .



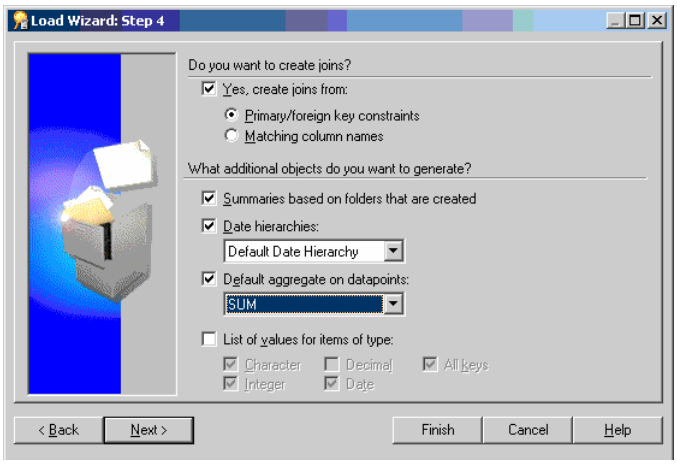
Obr. 7.2.: Výběr uživatele BA

V dalším okně vybereme schéma, tabulky a objekty (Obr. 7.3.).



Obr. 7.3.: Výběr schématu

Na Obr. 7.4. vidíme upřesnění pro vytvoření business area. V posledním okně pouze zadáme název nové BA.



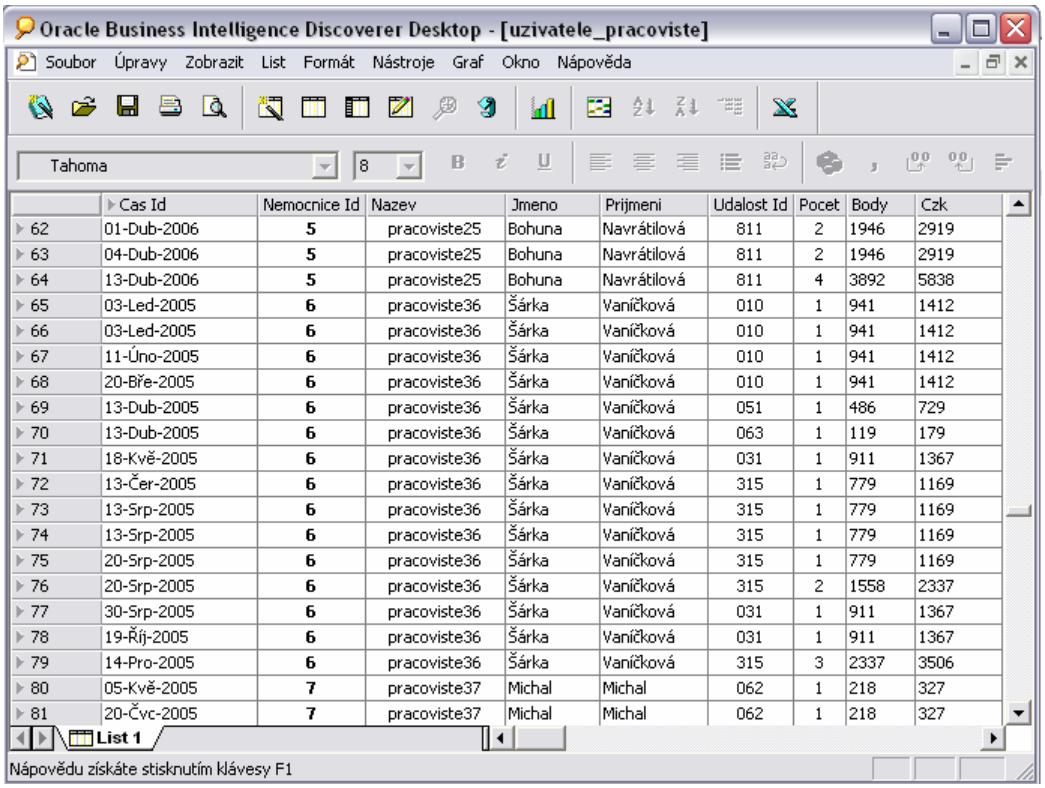
Obr. 7.4.: Specifikace pro generování BA

V Oracle Discoverer User Edition se vytváří sešity, které jsou složeny z listů. Listy zobrazují jednotlivé sestavy. Významově blízké sestavy zobrazené v listech se sdružují do sešitů.

Dále použité ukázky jsou z Discoverer User Edition.

7.3. Ukázky

Vytvořili jsme dva ukázkové sešity. V prvním je výpis všech klinických událostí na pracovišti pod dohledem určitého lékaře (Obr. 7.5.). V každém řádku je uveden i počet jednotlivých událostí v dané datum a jejich bodové i finanční ohodnocení.

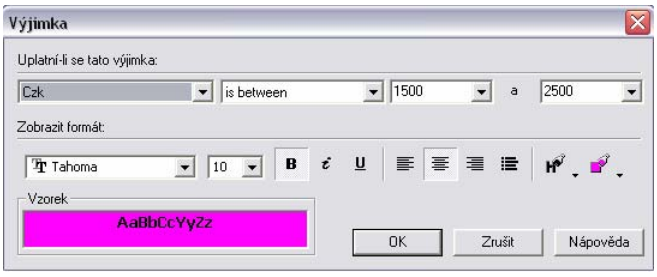


	Cas Id	Nemocnice Id	Nazev	Jmeno	Prijmeni	Udalost Id	Pocet	Body	Czk
62	01-Dub-2006	5	pracoviste25	Bohuna	Navrátilová	811	2	1946	2919
63	04-Dub-2006	5	pracoviste25	Bohuna	Navrátilová	811	2	1946	2919
64	13-Dub-2006	5	pracoviste25	Bohuna	Navrátilová	811	4	3892	5838
65	03-Led-2005	6	pracoviste36	Šárka	Vaníčková	010	1	941	1412
66	03-Led-2005	6	pracoviste36	Šárka	Vaníčková	010	1	941	1412
67	11-Úno-2005	6	pracoviste36	Šárka	Vaníčková	010	1	941	1412
68	20-Bře-2005	6	pracoviste36	Šárka	Vaníčková	010	1	941	1412
69	13-Dub-2005	6	pracoviste36	Šárka	Vaníčková	051	1	486	729
70	13-Dub-2005	6	pracoviste36	Šárka	Vaníčková	063	1	119	179
71	18-Kvě-2005	6	pracoviste36	Šárka	Vaníčková	031	1	911	1367
72	13-Čer-2005	6	pracoviste36	Šárka	Vaníčková	315	1	779	1169
73	13-Srp-2005	6	pracoviste36	Šárka	Vaníčková	315	1	779	1169
74	13-Srp-2005	6	pracoviste36	Šárka	Vaníčková	315	1	779	1169
75	20-Srp-2005	6	pracoviste36	Šárka	Vaníčková	315	1	779	1169
76	20-Srp-2005	6	pracoviste36	Šárka	Vaníčková	315	2	1558	2337
77	30-Srp-2005	6	pracoviste36	Šárka	Vaníčková	031	1	911	1367
78	19-Říj-2005	6	pracoviste36	Šárka	Vaníčková	031	1	911	1367
79	14-Pro-2005	6	pracoviste36	Šárka	Vaníčková	315	3	2337	3506
80	05-Kvě-2005	7	pracoviste37	Michal	Michal	062	1	218	327
81	20-Čvc-2005	7	pracoviste37	Michal	Michal	062	1	218	327

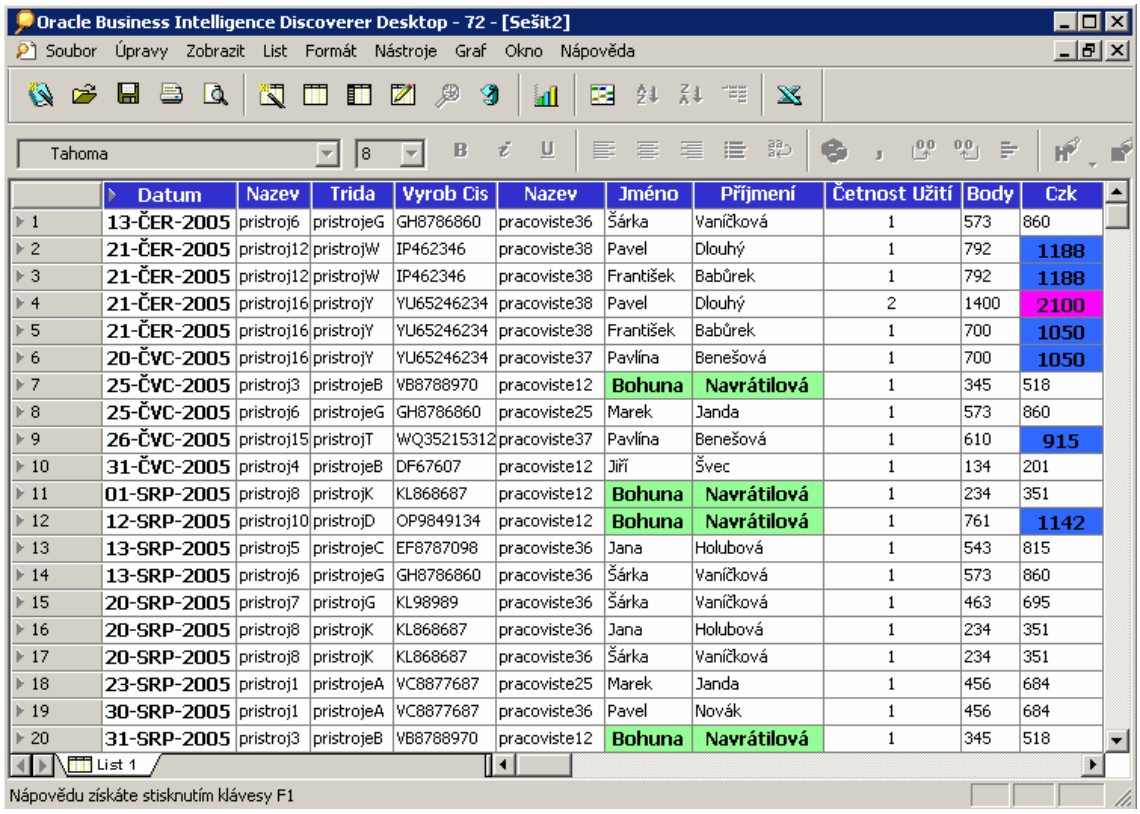
Obr. 7.5.: Sešit č.1

V druhém sešitu přístrojů za období prázdnin, tedy od 30.6.2005 až 31.8.2004. Na každém řádku je vyobrazen počet užití jednoho přístroje jedním lékařem v daný den a samozřejmě bodové a finanční ohodnocení.

Zde bylo pro porovnání užito množství grafických úprav a zvýraznění pomocí výjimek.



Obr. 7.6.:Výjimka



	Datum	Nazev	Trida	Vyrob Cis	Nazev	Jméno	Příjmení	Četnost Užití	Body	Czk
1	13-ČER-2005	pristroj6	pristrojeG	GH8786860	pracoviste36	Šárka	Vaníčková	1	573	860
2	21-ČER-2005	pristroj12	pristrojW	IP462346	pracoviste38	Pavel	Dlouhý	1	792	1188
3	21-ČER-2005	pristroj12	pristrojW	IP462346	pracoviste38	František	Babůrek	1	792	1188
4	21-ČER-2005	pristroj16	pristrojY	YU65246234	pracoviste38	Pavel	Dlouhý	2	1400	2100
5	21-ČER-2005	pristroj16	pristrojY	YU65246234	pracoviste38	František	Babůrek	1	700	1050
6	20-ČVC-2005	pristroj16	pristrojY	YU65246234	pracoviste37	Pavlína	Benešová	1	700	1050
7	25-ČVC-2005	pristroj3	pristrojeB	VB8788970	pracoviste12	Bohuna	Navrátilová	1	345	518
8	25-ČVC-2005	pristroj6	pristrojeG	GH8786860	pracoviste25	Marek	Janda	1	573	860
9	26-ČVC-2005	pristroj15	pristrojT	WQ35215312	pracoviste37	Pavlína	Benešová	1	610	915
10	31-ČVC-2005	pristroj4	pristrojeB	DF67607	pracoviste12	Jiří	Švec	1	134	201
11	01-SRP-2005	pristroj8	pristrojK	KL868687	pracoviste12	Bohuna	Navrátilová	1	234	351
12	12-SRP-2005	pristroj10	pristrojD	OP9849134	pracoviste12	Bohuna	Navrátilová	1	761	1142
13	13-SRP-2005	pristroj5	pristrojeC	EF8787098	pracoviste36	Jana	Holubová	1	543	815
14	13-SRP-2005	pristroj6	pristrojeG	GH8786860	pracoviste36	Šárka	Vaníčková	1	573	860
15	20-SRP-2005	pristroj7	pristrojG	KL98989	pracoviste36	Šárka	Vaníčková	1	463	695
16	20-SRP-2005	pristroj8	pristrojK	KL868687	pracoviste36	Jana	Holubová	1	234	351
17	20-SRP-2005	pristroj8	pristrojK	KL868687	pracoviste36	Šárka	Vaníčková	1	234	351
18	23-SRP-2005	pristroj1	pristrojeA	VC8877687	pracoviste25	Marek	Janda	1	456	684
19	30-SRP-2005	pristroj1	pristrojeA	VC8877687	pracoviste36	Pavel	Novák	1	456	684
20	31-SRP-2005	pristroj3	pristrojeB	VB8788970	pracoviste12	Bohuna	Navrátilová	1	345	518

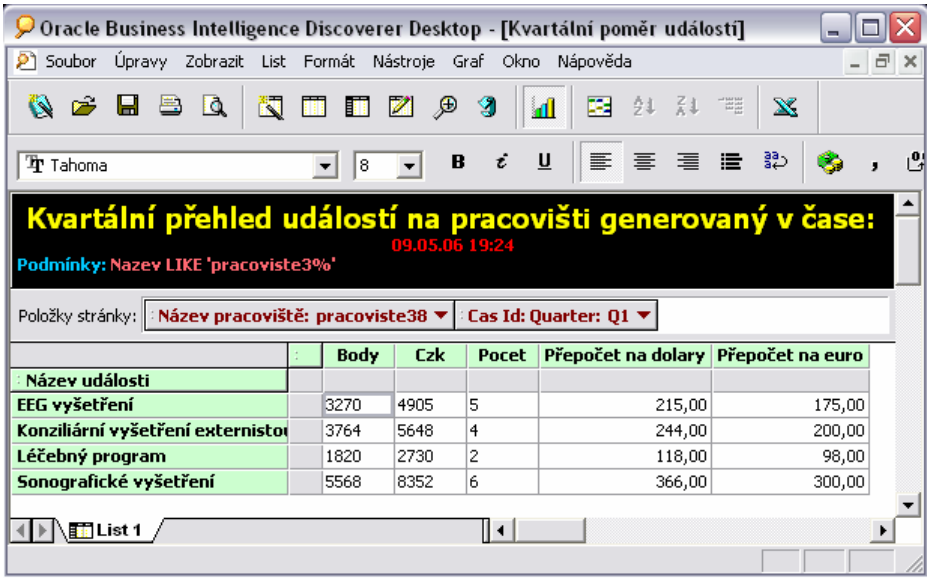
Obr. 7.7.: Sešit č.2

Možností aplikace v pohledech na data jsou mnohem větší, než bylo prezentováno. Umožňuje například vytvářet grafy, exportovat do Microsoft Excel a HTML (pouze ale staticky, dynamicky OracleBI Discoverer Plus) a mnoho dalšího.

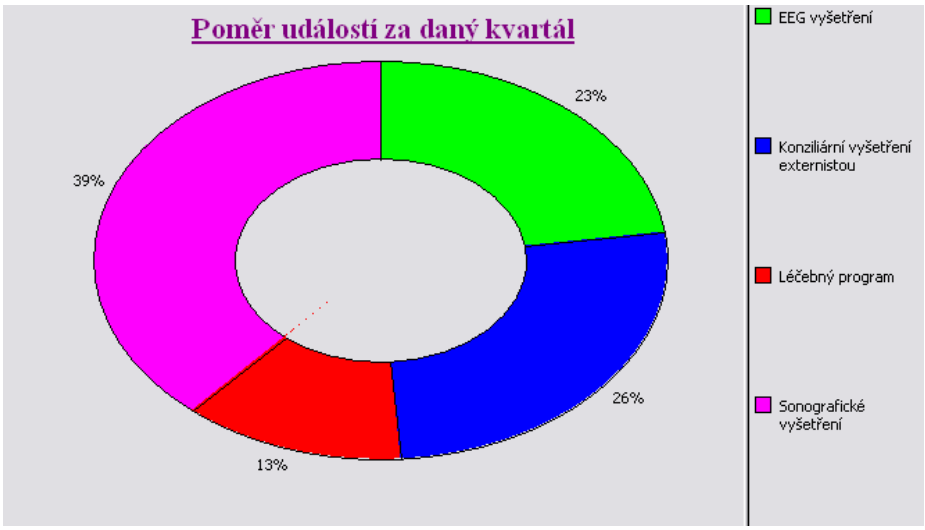
Připravili jsme ještě ukázku s grafem. Pro jeho zobrazení je potřeba použít křížovou tabulku (cross table). Dotaz vypíše v tabulce a následně zobrazí v grafu počet událostí po kvartálech nebo za celý rok (v našem případě 2005). Zároveň se zobrazí bodové a finanční ohodnocení. Pomocí *výpočtů* se finanční ohodnocení klinických událostí přepočte na dolary (s kurzem 23 Kč/$) a na euro (kurz 28 Kč/€) (lze zvolit uživatelem jako parametr). Zároveň byly použity podmínky, které dovolují zobrazovat pouze údaje z pracovišť, v jejichž názvu je ‚`pracoviste3`‘. Tato podmínka se zobrazuje v nadpisu i s datumem generování a s titulkem popisujícím list sešitu.

Obr. 7.8. zobrazuje výpis klinických událostí za první kvartál `Q1` roku 2005 pro zvolené pracoviště `pracoviste38`. V řádcích s jednotlivými událostmi je jejich počet během kvartálu, ohodnocení bodová a finanční v korunách, eurech i dolarech.

Na Obr. 7.9. je graf k předchozímu dotazu.

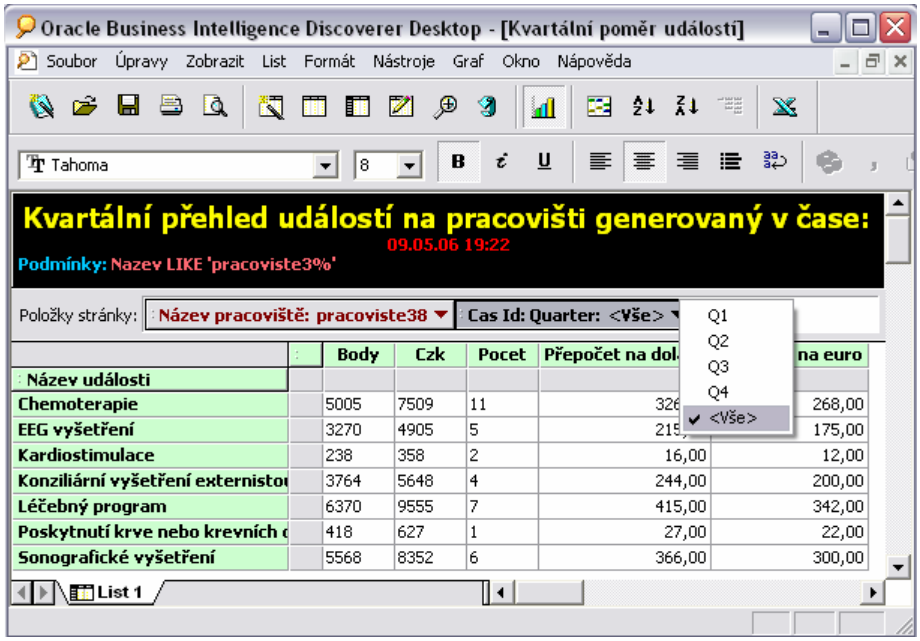


Obr.7.8.:Kvartální přehled událostí na pracovišti č. 38 v prvním kvartálu r.2005

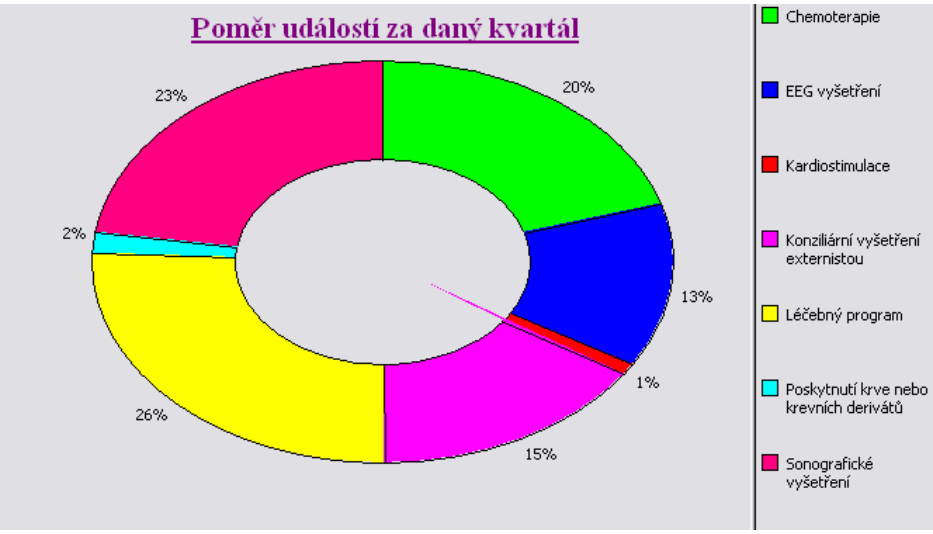


Obr.7.9.:Prstencový 2D graf zobrazující přehled událostí na pracovišti č. 38 v prvním kvartálu r.2005

Na Obr. 7.10. je vidět výpis klinických událostí za všechny kvartály roku 2005 na pracovišti č. 38 za stejných podmínek jako pro předchozí výpis. Obr. 7.11. je potom grafem k tomuto dotazu.



Obr.7.10.: Kvartální přehled událostí na pracovišti č. 38 za r.2005



Obr.7.11.: Prstencový 2D graf zobrazující přehled událostí na pracovišti č. 38 za r.2005

8. Závěr

Obsahem diplomové práce bylo navrhnout hybridní datové úložiště v prostředí Oracle. Jsem velmi ráda, že jsem si toto téma vybrala, i když problémy, které se během projektu vyskytly, nebyly vždy snadno odstranitelné. Byla to moje první zkušenost s datovými sklady, i přesto si myslím, že návrh je zdařilý. Celá práce pro mne byla zdrojem nových zkušeností a poznatků z oblasti databází, které bych nyní dokázala v praxi uplatnit i jiným způsobem.

8.1. Hardware a sotfware

Celá práce byla vytvářena na notebooku s operačním systémem Microsoft Windows XP Professional, procesorem Pentium M, 1.73 GHz a operační pamětí 512 MB. Nainstalovaný byl databázový systém Oracle9i Release 2 Enterpise Edition, Oracle9i Management and Integration 9.2. pro export a zálohování, balíky Oracle Business Inteligence 10g a Oracle Business Inteligence Tools 10g obsahující Discoverer pro zobrazování dat. Původně byl pro návrh datového skladu nainstalován Oracle Warehouse Builder 10g, ale nepodařilo se návrh exportovat do databáze, proto byl nakonec celý návrh realizován v databázovém server, který od této verze návrh datového skladu a OLAP podporuje.

8.2. Na co nezbyl prostor

Neúspěch s Oracle Warehouse Builderem byl velkým zklamáním a kdyby bylo více času, jistě by se podařilo problém vyřešit. Také Discoverer je velmi zajímavou aplikací s mnoha možnostmi, bylo by velmi zajímavé věnovat se mu déle.

Užití databázového serveru Oracle 10g by také mohlo být pro projekt velkým přínosem. Původně jsme zamýšleli tuto verzi použít, ale nejevila se příliš přehledná, a proto jsme se raději vrátili k předchozí verzi 9i, ke které existuje i větší množství literatury.

8.3. Funkčnost

Datový sklad je funkční. Tabulky faktů a dimenze umožňují rozbalování podrobností, dále je možné data třídit podle různých kategorií.

V rámci práce byla využita databáze menších rozměrů než je reálně používáno v praxi, nicméně byla postačující pro prověření, demonstrování a porovnání funkcionality hybridního úložiště dat.

8.4. Nepříjemné zkušenosti při zpracování

Bohužel, instalace žádného produktu firmy Oracle se neobešla zcela bez potíží. Nejčastěji se jednalo o nefunkční službu TNSListener, která se opravila editací souborů `listener.ora` a `tnsnames.ora`. Jako největší problém se ukázala instalace Oracle Warehouse Builder, kdy po uzavření aplikace došlo ke zhroucení operačního systému Windows. Po dlouhé době strávené pročítáním internetových diskuzí se příčinou ukázala býti verze ovladačů pro grafickou kartu – pomohla instalace nových driverů a nové verze JVM.

Přehled zkratek

BA -	Business Area
DB -	Database
DW -	Data Warehouse
ERP -	Enterprise Resource Planning
ETL -	Extraction, Transformation, Loading
ETT -	Extraction, Transformation, Transport
EUL -	End User Layer
HOLAP -	Hybrid Online Analytical Processing
HTML -	HyperText Markup Language
HW -	Hardware
IS -	Information Systems / Services
MDDB -	multi-dimensional database
MOLAP -	Multidimensional Online Analytical Processing
ODBC -	Open Database Connectivity
OEMC -	Oracle Enterprice Management Console
OLAP -	Online Analytical Processing
OLTP -	Online Transaction Processing
OWB -	Oracle Warehouse Builder
RDBMS -	Relational Database Management System
ROLAP -	Relational Online Analytical Processing
SQL -	Structured Query Language
SW -	Software

Slovník pojmů

BA - je seskupení tabulek, pohledů nebo jen určitých položek tabulky navržené pro potřeby koncového uživatele; pracovní oblast.

DB - data, která slouží více aplikacím, jsou v nich minimalizovány redundance dat a existuje vhodně centralizovaná správa těchto dat. Cílem databázového systému je uspořádat datové zdroje (datovou základnu) tak, aby tyto zdroje mohly být využívány.

DW - komplexní data uložená ve struktuře, která umožňuje efektivní analýzu a dotazování. Data do datového skladu jsou čerpána z primárních informačních systémů a dalších.

ERP - zejména finančně orientovaný informační systém pro určení a plánování podnikových zdrojů potřebných k přijetí, zhotovení, dodání a zaúčtování zákaznického obchodního případu. Tyto systémy bývají považovány za jádro celého informačního systému, nabízejí komplexní pohled na oblast zdrojů podniku.

ETL - proces, který umožňuje přesouvat data z více zdrojů, reformátovat, pročišťovat je a převádět do dalších databází nebo datových skladů k provedení analýzy nebo do jiného provozního systému pro podporu obchodních procesů. Je to tedy technologie umožňující přenos a transformaci dat ze zdrojových systémů do databáze datového skladu.

EUL - vrstva koncového uživatele; intuitivní provozně zaměřený pohled na databázi, který izoluje uživatele od obvyklé složitosti databáze.

Hardware - označuje veškeré fyzicky existující technické vybavení počítače.

HOLAP - kombinace technologií ROLAP a MOLAP, jsou data uložena z části v relační a z části v multidimenzionální databázi.

HTML - jazyk, který umožňuje publikovat on-line elektronické dokumenty, obsahující text, obrázky, video, apod. s možností jejich formátování, které se přibližuje formátování u klasických textových editorů.

IS - Informační systém.

IT - informační technologie.

MDDB - databáze, kde jsou data uložena na principu vícerozměrné matice. Hodnoty jsou přístupné přímo pro danou kombinaci prvků dimenzí.

MOLAP - realizace databází pouze s agregovanými, tzn. že data jsou uložena v multidimenzionální databázi.

ODBC - specifikace rozhraní pro přístup k relačním databázím z operačního systému Windows. ODBC je de-facto standardem, který respektuje většina SW firem.

OEM - nástroj databázového serveru Oracle pro práci s daty a správu databáze.

OLAP - způsob analýzy při kterém procházíme daty a podle zjištěných informací volíme nové pohledy, vysvětlující a doplňující zjištěné hodnoty a trendy.

OLTP - zpracování transakcí v reálném čase, režim práce s databází, kdy se zpracování požadavků neodkládá, ale provádí se okamžitě, jakmile požadavek od uživatele přijde. Je to způsob práce většiny primárních informačních systémů.

RDBMS - soubor algoritmů, které řídí relační databázi. Většina RDBMS dnes umožňuje přistupovat k datům pomocí SQL.

Databáze, kde jsou data uložena v jednotlivých tabulkách. Tabulka (relace) je tvořena záznamy s jednotnou strukturou polí. Pole (atributy) jsou atomické. Tabulky mohou být propojeny (joins).

ROLAP - analýza OLAP s přímým přístupem do relační databáze.

SQL - dotazovací jazyk, podobný angličtině, sloužící k získání dat z relačních databází. Standard jazyka pro práci s relačními databázemi.

Software - veškeré programové vybavení počítače.

Literatura

[1] Luboslav Lacko: Databáze: datové sklady, OLAP a dolování dat s příklady v Microsoft SQL Serveru a Oracle, *Computer Press 2003*

[2] Luboslav Lacko: Oracle: Správa, programování a použití databázového systému, *Computer Press 2002*

[3] Mark Humphries a kol.: Data warehousing - návrh a implementace, *Computer Press 2002*

[4] Kevin Loney, Marlene Theriault: Mistrovství v Oracle - Kompletní průvodce tvorbou, správou a údržbou databází, *Computer Press 2002*

[5] Michael Abbey, Mike Corey, Ian Abramson: Oracle 9i, *SotfPress 2002*

[6] Ota Novotný, Jan Pour, David Slánský: Business Intelligence - Jak využít bohatství ve vašich datech, *Grada 2005*

[7] R. Kimball, L. Reeves, M. Ross, W. Thornthwaite: The Data Warehouse Lifecycle Toolkit, *Wiley Computer Publishing 1998*

[8] www.oracle.com

[9] http://datamining.xf.cz

[10] Kishore Jaladi: *Data Warehousing: Data Models and OLAP operations* (prezentace)

[11] Luboslav Lacko: *Analysis Services* (2002, prezentace)

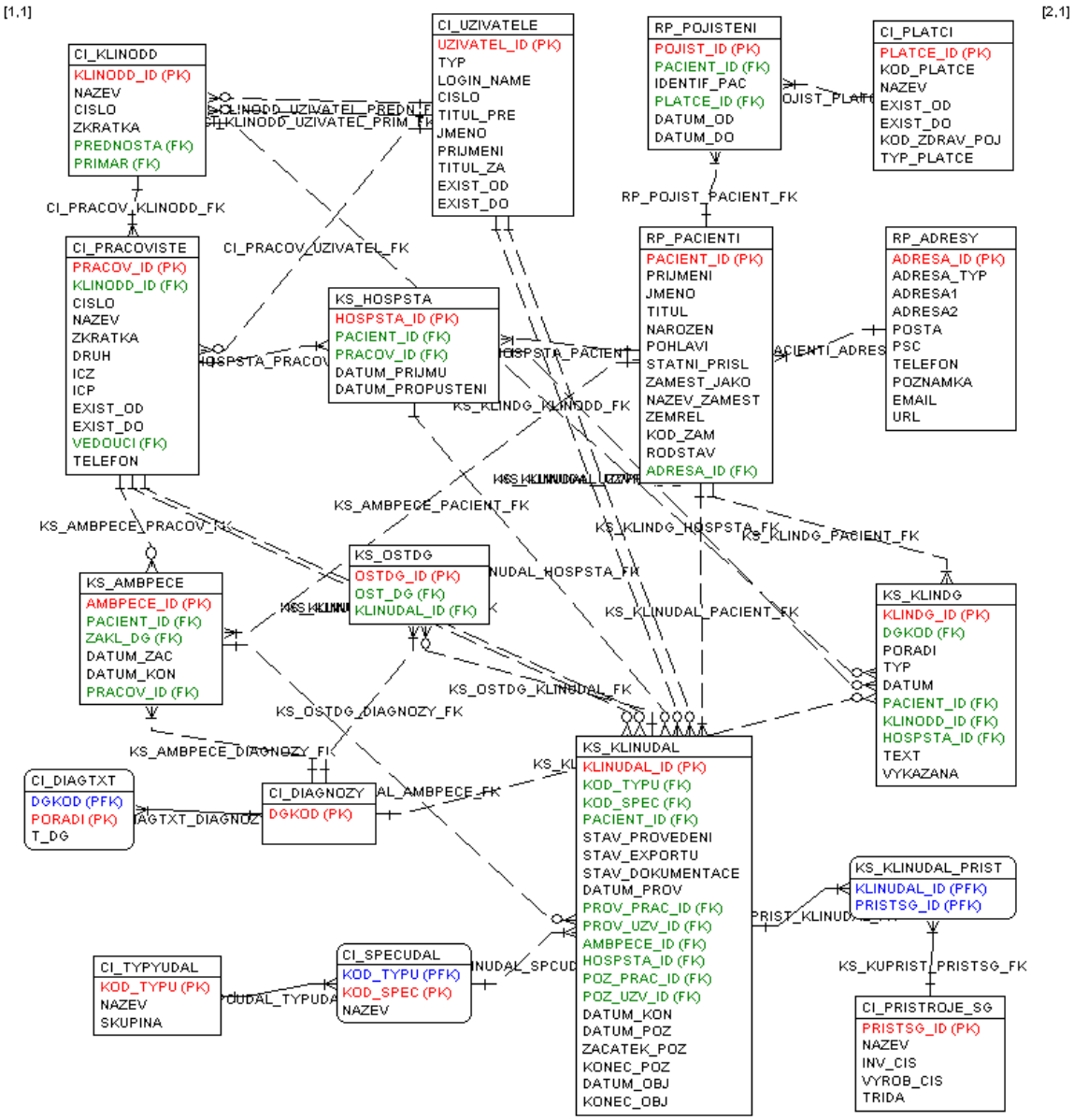
[12] Chris Claterbos, Dan Vlamis: *Using Oracle9i Warehouse Builder* (2003, prezentace)

[13] http://www2.cs.uregina.ca/~hamilton/courses/831/notes/dcubes/dcubes.html

[14] http://system.ccb.cz/site/data-warehousing

Přílohy

ERA model zdrojové databáze



Evidenční list

Souhlasím s tím, aby moje diplomová práce byla půjčována k prezenčnímu studiu v Univerzitní knihovně ZČU v Plzni.

Datum: ……. ..…….

podpis diplomanta

Uživatel stvrzuje svým čitelným podpisem, že tuto diplomovou práci použil ke studijním účelům a prohlašuje, že ji uvede mezi použitými prameny.

Jméno	Fakulta / Katedra	Datum	Podpis