



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA PODNIKATELSKÁ**  
**USTAV INFORMATIKY**

FACULTY OF BUSINESS AND MANAGEMENT  
INSTITUTE OF INFORMATICS

## **ZAVEDENÍ SAP HANA A SLT DO FIRMY PRO AKCELERACI ANALÝZY ZISKOVOSTI**

INTEGRATION OF SAP HANA AND SLT INTO A COMPANY FOR PROFITABILITY ANALYSIS  
ACCELERATION

**BAKALÁŘSKÁ PRÁCE**  
BACHELOR'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**PETR SVITÁLEK**

**VEDOUCÍ PRÁCE**  
SUPERVISOR

**Ing. JAN LUHAN, Ph.D.**

BRNO 2015

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

**Svitálek Petr**

---

Manažerská informatika (6209R021)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává bakalářskou práci s názvem:

**Zavedení SAP HANA a SLT do firmy pro akceleraci analýzy ziskovosti**

v anglickém jazyce:

**Integration of SAP HANA and SLT into a Company for Profitability Analysis Acceleration**

Pokyny pro vypracování:

Úvod

Cíle práce, metody a postupy zpracování

Teoretická východiska práce

Analýza současného stavu

Vlastní návrhy řešení

Závěr

Seznam použité literatury

Přílohy

Seznam odborné literatury:

LACKO, L. Databáze: datové sklady, OLAP a dolování dat s příklady v Microsoft SQL Serveru a Oracle: podrobná uživatelská příručka. Brno: Computer Press, 2003. 490 s. ISBN 80-722-6969-0.

PATEL, M. SAP ERP Financials: podrobná uživatelská příručka. Brno: Computer press, 2010. 464 s. ISBN 978-80-251-2488-8.

SCHWALBE, K. Řízení projektů v IT: Kompletní průvodce. Praha: Computer Press, 2011. 632 s. ISBN 978-80-251-2882-4.

SILBERSCHATZ, A., H. F. KORTH and S. SUDARSHAN. Database system concepts. 6th ed. New York: McGraw-Hill, 2011. 1376 p. ISBN 978-00-735-2332-3.

SODOMKA, P. a H. KLČOVÁ. Informační systémy v podnikové praxi. 2. aktualizované a rozšířené vyd. Brno: Computer Press, 2010. 504 s. ISBN 978-80-251-2878-7.

Vedoucí bakalářské práce: Ing. Jan Luhan, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2014/2015.

L.S.

---

doc. RNDr. Bedřich Půža, CSc.  
Ředitel ústavu

---

doc. Ing. et Ing. Stanislav Škapa, Ph.D.  
Děkan fakulty

V Brně, dne 28.2.2015

## **Abstrakt**

Podstatou této bakalářské práce je dokončení zadaného projektu zavedení SAP HANA technologie v roli akcelérátoru pro podnikový ERP systém, konkrétně pro modul CO-PA. Součástí je i základní teorie z oblasti databázových systémů a popis SAP HANA a SLT technologií. Díky nim by měla firma být schopná rychlé analýzy vlastních financí a na základě jejich výsledků pružně reagovat změnami firemní struktury nebo strategie.

## **Abstract**

The main purpose of this bachelor's thesis is to complete the given project to implement SAP HANA technology in the role of accelerator for ERP business system, specifically for CO-PA module. There is also basic theory of database systems included as well as description of SAP HANA and SLT technologies. Thanks to them, company should be able to quickly analyze internal financials and based on the given results flexibly respond by changing corporate structure or strategy.

## **Klíčová slova**

SAP HANA, SLT, CO-PA, OLAP, OLTP, databázové systémy, in-memory databáze, ERP, analýza ziskovosti

## **Key words**

SAP HANA, SLT, CO-PA, OLAP, OLTP, database systems, in-memory database, ERP, profitability analysis

## **Bibliografická citace**

SVITÁLEK, P. *Zavedení SAP HANA a SLT do firmy pro akceleraci analýzy ziskovosti*.  
Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2015. 67 s. Vedoucí  
bakalářské práce Ing. Jan Luhan, Ph.D.

## **Čestné prohlášení**

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne .....

.....

Podpis studenta

## **Poděkování**

Děkuji především panu Ing. Janu Luhanovi, Ph.D. za vstřícnost a odborné vedení při tvorbě této bakalářské práce. Stejně tak za veškeré podněty, rady a připomínky, které mi jím byly uděleny.

# OBSAH

ÚVOD.....	11
CÍLE PRÁCE.....	12
1 TEORETICKÁ VÝCHODISKA.....	13
1.1 Od dat k moudrosti .....	13
1.2 Databázové systémy .....	14
1.2.1 Historie.....	15
1.2.2 Relační databáze .....	15
1.2.3 Schéma a instance .....	16
1.3 ER model .....	16
1.3.1 Kardinality vztahů mezi entitami.....	17
1.3.2 Normální formy .....	18
1.4 Databázové jazyky .....	19
1.4.1 DDL .....	19
1.4.2 DML.....	20
1.4.3 DCL .....	20
1.5 Rozdíly OLAP a OLTP databází .....	20
1.5.1 OLTP (online transaction processing) .....	20
1.5.2 OLAP (online analytical processing).....	21
1.6 Multidimenzionální databáze.....	21
1.6.1 Restrikce a agregace .....	22
1.7 Podnikové informační systémy.....	23
1.7.1 Součásti IS .....	24
1.8 Typy zavedení aplikací .....	25
1.8.1 On-premise.....	25
1.8.2 Cloud computing.....	26
1.8.3 Možnosti rozšíření .....	26
2 ANALÝZA SOUČASNÉHO STAVU.....	27
2.1 Profil společnosti .....	27
2.1.1 Hlavní činnosti.....	27
2.1.2 Klíčové údaje .....	28

2.1.3	Strategie .....	28
2.1.4	SWOT analýza .....	28
2.2	IT infrastruktura .....	30
2.2.1	Hardwarová vrstva .....	31
2.2.2	Vrstva operačního systému .....	31
2.2.3	Databázová vrstva .....	31
2.3	Rozbor informačního systému .....	32
2.3.1	SAP .....	32
2.3.2	SAP Business Suite .....	32
2.3.3	SAP ERP .....	32
2.3.4	ERP Financials .....	34
2.4	Technologie in-memory databáze .....	36
2.4.1	SAP HANA Studio .....	36
2.4.2	Perzistence .....	37
2.4.3	Column-based tabulky .....	38
2.5	Kompresce dat .....	40
2.6	Hlavní a rozdílové úložiště .....	41
3	VLASTNÍ NÁVRH ŘEŠENÍ .....	43
3.1	Zadání projektu .....	43
3.2	Databázová platforma pro CO-PA .....	43
3.2.1	Instalace .....	44
3.3	Hardwarová platforma pro SAP HANA .....	45
3.3.1	Datová úložiště .....	46
3.3.2	Síťová rozhraní .....	47
3.4	Schéma propojení .....	48
3.5	SAP LT replikace .....	49
3.5.1	Dimenzování (sizing) .....	50
3.5.2	Instalace SAP NetWeaver .....	51
3.5.3	Úpdaty softwarových součástí .....	53
3.5.4	Instalace DMIS .....	55
3.6	Propojení systémů .....	55
3.6.1	Aplikační uživatelé .....	56

3.6.2	RFC a databázové připojení.....	56
3.6.3	Replikace a akcelerace.....	57
3.7	Porovnání rychlosti zpracování dat.....	58
4	ZÁVĚR.....	60
	SEZNAM POUŽITÝCH ZDROJŮ.....	61
	SEZNAM OBRÁZKŮ.....	63
	SEZNAM TABULEK.....	64
	SEZNAM ZKRATEK.....	65
	SEZNAM PŘÍLOH.....	67

## ÚVOD

Databázové systémy existují již desítky let jako pomocníci v oblasti podnikové sféry. Vše začalo v podobě dřevěných štítků, které museli být obsluhou tehdejších počítačů neustále manuálně vkládány ke zpracování. Každý takový štítek svým způsobem představoval jeden prodaný produkt neboli položku.

Data ze štítků pak byla ukládána zprvu na magnetické pásky, které byly později nahrazeny pevnými (mechanickými) disky. Kapacita disků umožňovala ukládat větší množství dat a uchovávat tak například historická data a především i předem definované sumy neboli agregace, které sloužily pro zpětnou analýzu prodejů. Agregace pomáhaly zjišťovat nepřehledné množství informací, například kolik se prodalo určitých produktů za daný čas, jaké byly celkové tržby nebo která barva daného výrobku šla nejvíce na odbyt během určitého období.

Ale právě množství těchto sum se stalo později problémem, protože s každou zapsanou položkou se musely zároveň zpracovávat i agregace, které chtěl mít zákazník k dispozici. Díky tomu se prodlužovala doba potřebná pro zapsání a sečtení dat, což vedlo k nutné separaci transakční a analytické části aplikačních systémů.

Nyní ovšem díky rychlosti a možnostem, jež současný hardware skýtá, vznikají databáze běžící přímo v operační paměti, která je mnohonásobně rychlejší než dnes nejčastěji použitý typ uložení dat: mechanický disk. Výsledkem je, že složité analytické dotazy mohou být při správném rozložení dat a správném položení dotazu zodpovězeny v rámci sekund i pro tabulky o stovkách milionů záznamů.

## CÍLE PRÁCE

Stěžejním cílem práce je navrhnout a zrealizovat implementaci databázového systému SAP HANA, jež bude sloužit jako akcelerátor pro analýzy ziskovosti pro konkrétní firemní subjekt. Realizace se přitom bude skládat z více dílčích částí, které se zavedením této moderní technologie souvisí.

Budou zde popsány základní principy použité in-memory databáze, její jednotlivé součásti, výhody a slabá místa. Vysvětlen bude způsob, jakým jsou data ukládána a chráněna proti hardwarovému selhání. Co může přinést pro podnikové systémy, kde se nejvíce uplatní a proč. Bakalářská práce se bude zabývat také aplikací SLT, která je použita jako prostředník, sloužící pro okamžitou replikaci zdrojových podnikových dat do cílové databáze běžící v operační paměti.

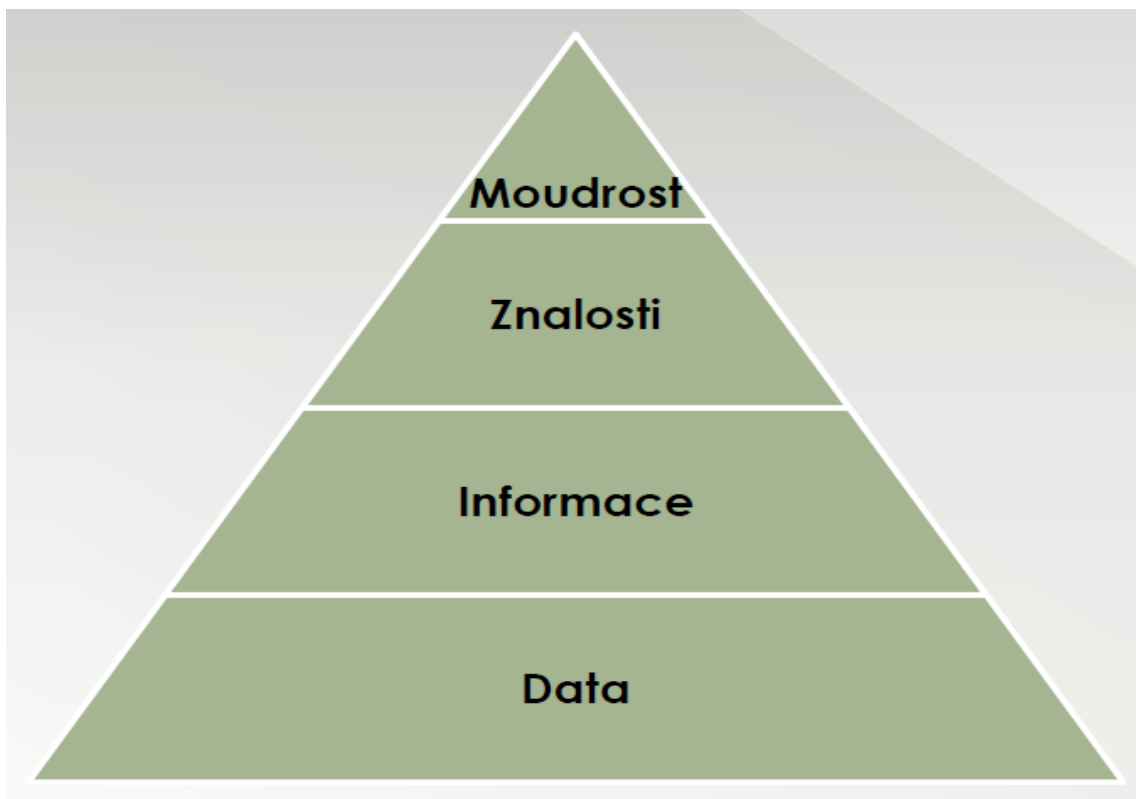
Dále bude zanalyzován současný stav společnosti, k čemu slouží část podnikového systému, kterou chce firma akcelarovat a jaké konkrétní výhody přinese zdárné dokončení projektu. Na základě prvotního zadání a informací získaných od zákazníka během přípravné fáze pak budou vybrány konkrétní postupy, nastavení, hardware a software potřebný pro implementaci.

# 1 TEORETICKÁ VÝCHODISKA

V teoretické části jsou vysvětleny základní pojmy spojené s databázemi, protože bez jejich vysvětlení by bylo složité se orientovat ve velkém množství slovních obrátů a zkratkou použitých v ostatních částech práce. Přes relační databáze pak plynule ke konci přechází k multidimenzionálnímu pohledu a krátce popisuje informační systémy a možnosti jak je do firmy zavést.

## 1.1 Od dat k moudrosti

Co se kvantitativně týká, jsou data na prvním místě mezi pojmy, které budou nyní blíže představeny. Jedná se ve své podstatě o stavy objektů, nebo procesů, které probíhají v reálném světě kolem nás. Jsou nejčastěji uložena na nějakém nosiči. Data představují potenciaální informace, což jsou poznatky z dat načerpáné, sloužící nějakému účelu. Na jejich základě přijímající subjekt mění svůj stav nebo chování. Dalším stupněm využití dat jsou znalosti. Nabývá jich pomocí zpracování informací, jejich porozumění a jakési zobecnění. Na samém vrcholu poznání je moudrost jakožto hlubší pochopení zákonitostí [1].



Obr. 1: Od dat k moudrosti (Zdroj: 1)

## 1.2 Databázové systémy

Pod pojmem databáze rozumíme prostor pro ukládání údajů, které jsou pak uchovány a zpracovávány odděleně od aplikací. Patří sem nejen jednotlivé hodnoty a údaje, ale i režie v podobě vztahů mezi prvky v databázi, popis struktury dat a různá integritní omezení. Jinými slovy by data neměla být na médiu uložena bez jakýchkoli logických vztahů, jinak bychom hovořili o obyčejném souborovém systému, i když i v něm musí některá pravidla existovat [2].

V této oblasti se nejčastěji setkáme se zkratkou DBMS z anglického database-management systém. Je to v podstatě kolekce dat, majících mezi sebou určité vazby a soubor programů, jejichž účelem je k těmto datům přistupovat. Hlavní cíl databázových systémů je bezpečné uložení velkého objemu dat a rychlý přístup k nim. Nejde zde samozřejmě jen o prosté čtení, ale systémy musí zvládat i ukládání, mazání a změnu. Zároveň poskytují ochranu dat jak před hardwarovou poruchou, tak před neautorizovaným přístupem nebo útokem z vnější. Možnosti použití těchto systémů jsou nepřehledné. Od prodeje, skladování, zásobování a výroby, až po školství, lidské zdroje nebo bankovníctví [3].

Právě pro oblast bankovníctví bylo důležité zavedení pojmu transakce. Transakce je soubor operací nad databází, které dohromady tvoří jednu logickou operaci v rámci aplikace. Pro představu vezměme jednoduchý příklad převodu peněz mezi dvěma účty. Pochopitelně je důležité, aby proběhlo odečtení z jednoho a přičtení té stejné částky na druhý účet. Transakce tedy musí proběhnout celá nebo je třeba všechny změny kompletně vrátit zpět. Tomuto jevu říkáme atomičnost. Druhou důležitou vlastností je pak konzistentnost. Součet hodnot obou účtů se má rovnat před i po provedení transakce, čili nesmí se narušit konzistence databáze. Pro úspěšné provedení transakce je bezpodmínečně nutné zajistit, aby změna dat zůstala v databázi uložena bez ohledu na to, zda dojde později k systémové chybě [3].

Denně je po celém světě zpracováváno neuvěřitelné množství informací z různých zařízení. Automobily, mobilní telefony ale i pračky nebo ledničky posílají data o svém stavu ať již kvůli předvídání poruchy nebo za účelem získat o zákazníkovi bližší

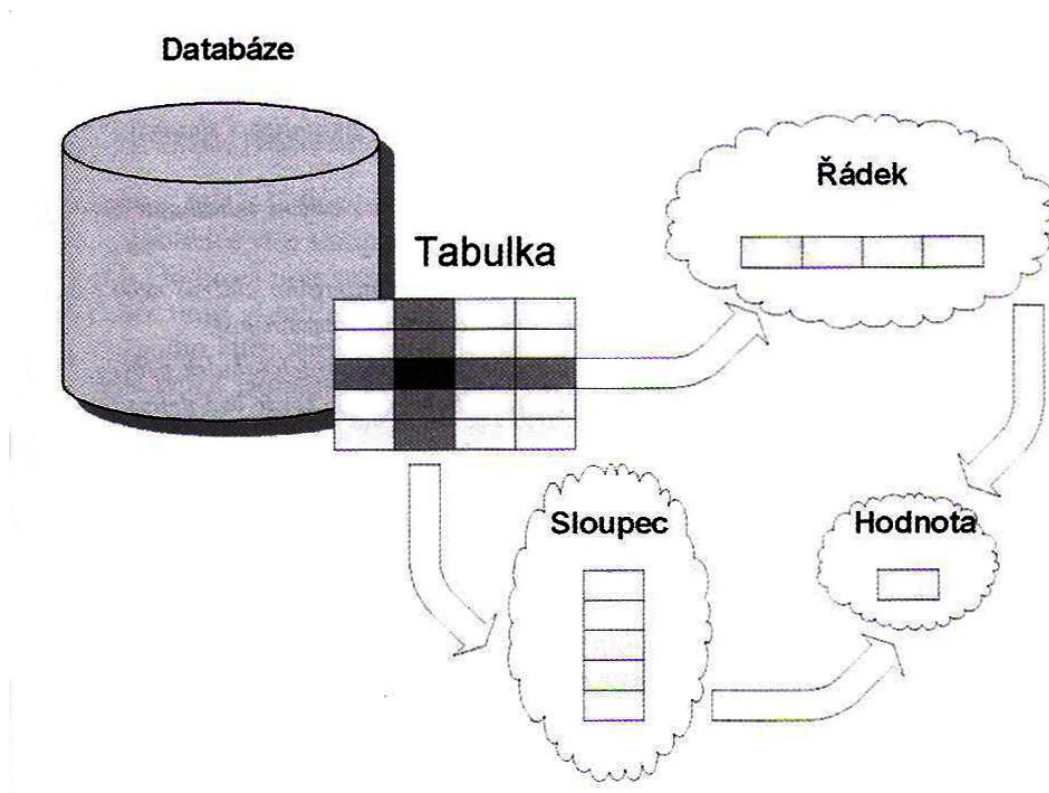
informace a použit je pak pro cílenou reklamu, případně pro zvýšení prodeje komplementů. Generální ředitelé chtějí mít po ruce detailní analýzy prodejů, aby se na jejich základě mohli rozhodovat, do které divize více investovat nebo jaké změny je nutné podniknout pro zlepšení prodeje. Všechny tyto informace musí být uloženy v databázích k tomu vhodně zvolených.

### **1.2.1 Historie**

Prvním milníkem v oblasti databázových systémů bylo použití magnetických pásek v padesátých letech dvacátého století až do začátku let šedesátých. Vyznačovaly se kapacitou dalece přesahující tehdejší velikost hlavní operační paměti a přístupem k datům pouze sekvenčním způsobem. Obvykle se tedy postupně načítaly dvě pásky nebo i více a výsledky se ukládaly na další pásku. Ke zvýšení nebo snížení určitých hodnot sloužili děrné štítky, které se zpracovávaly ze zásobníku, mezitím co se synchronně četly původní hodnoty z hlavní pásky. Výsledek se opět ukládal na magnetickou pásku, přičemž ta původní mohla být uchována pro účely zálohy historických dat. Na konci šedesátých let došlo k masivnímu rozšíření pevných disků, které umožňovali přistupovat k datům přímo. Nebylo již déle nutné otrocky procházet data od začátku, protože mechanické disky měli přístupovou dobu v desítkách milisekund nehledě na fyzické umístění dat na kotoučích. Relační databáze pak vytlačily na začátku osmdesátých let doposud nejčastěji používané síťové a hierarchické [3].

### **1.2.2 Relační databáze**

Nejrozšířenějším modelem databáze se stal relační databázový model. Hlavním stavebním prvkem jsou sloupce (atributy) a řádky (záznam, n-tice) s jednotlivými hodnotami. Z nich se poté skládají tabulky (relace), mající mezi sebou relační vztahy, které jsou zajištěny pomocí primárních a cizích klíčů. *„Primární klíč je jednoznačný identifikátor každého záznamu. Může to být sloupec, případně kombinace více sloupců, které slouží pro jednoznačnou identifikaci každého řádku tabulky. Hodnota pole primárního klíče musí být v rámci tabulky jedinečná”* [2]. Cizí klíč je pak také sloupec nebo více sloupců, které se vztahují k primárnímu klíči v jiné tabulce, přičemž musí být zaručeno, že hodnoty cizího klíče se vyskytnou jako hodnoty primárního klíče odkazované tabulky.



Obr. 2: Relační databáze (Zdroj: 2, str. 25)

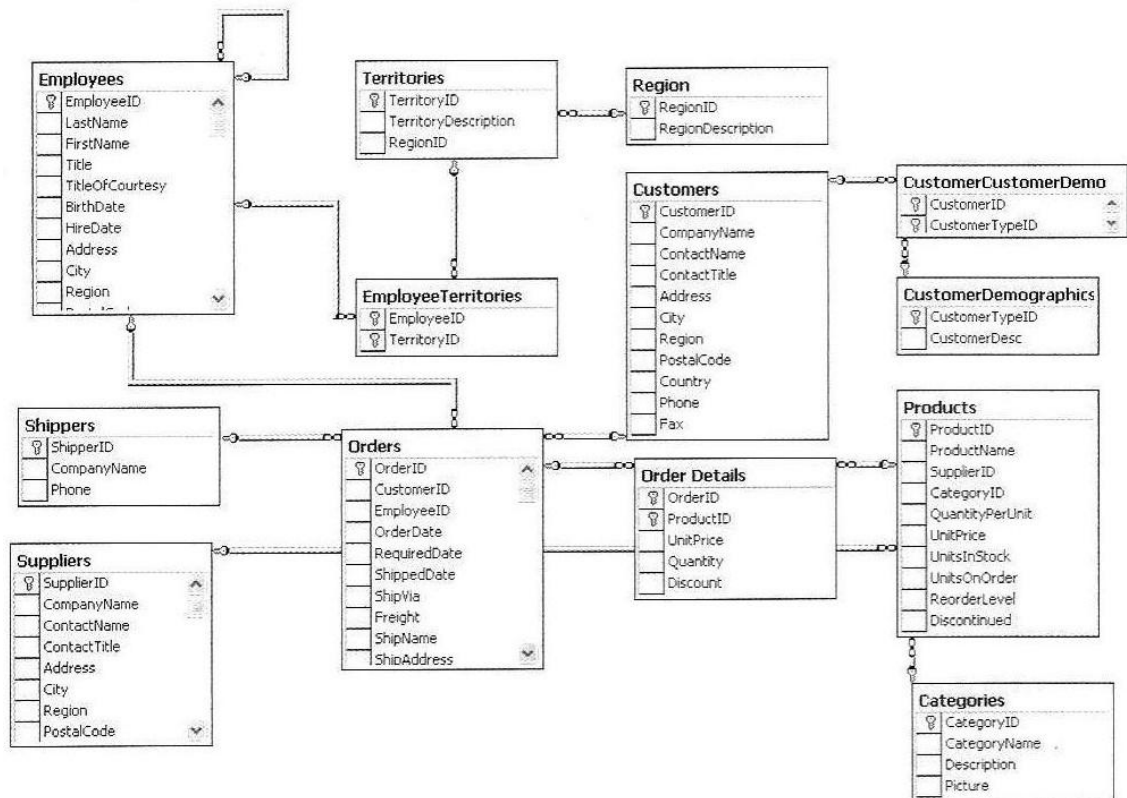
### 1.2.3 Schéma a instance

Schéma databáze je logický popis databáze, zatímco instance je její stav v určitém čase. Pro potřeby definice těchto dvou pojmů si vezměme jako příklad programovací jazyk. Schéma si můžeme představit jako seznam atributů a jejich možných hodnot. V oblasti programovacích jazyků bychom hovořili o definici proměnných a jejich typů. Naproti tomu u instance si představíme již konkrétní přiřazení hodnoty. Stejně jako u programu se hodnoty proměnných v čase mění, tak i obsah instance se liší v čase, tak jak jsou záznamy uvnitř databáze zrovna aktualizovány [3].

## 1.3 ER model

Co je databáze a databázový systém bylo popsáno v předchozích kapitolách. Nyní je ovšem potřeba podívat se blíže na proces vytváření designu a popisu vnitřních vztahů a prvků. Nejčastější formát pro relační databáze je takzvaný entity-relationship model, jehož diagramy, které jsou vytvářeny návrháři, slouží jako hlavní zdroj informací pro programátory při tvorbě prvotní podoby systému.

Abychom mohli popsat složitější pojmy z této oblasti, musíme si definovat pojem entita. Jedná se o objekt reálného světa, který je představován jednotlivými n-ticemi tabulky. Každá entita by měla být jedinečná a snadno odlišitelná od ostatních, k čemuž ve světě databází většinou slouží právě primární klíče, které mají nejčastěji formu takzvaných identifikačních čísel (ID). Databázové systémy umí tyto identifikátory samy vytvářet a hlídat zda má daný záznam opravdu unikátní ID [3].



Obr. 3: Příklad ER popisu databáze (Zdroj: 2, str. 27)

### 1.3.1 Kardinality vztahů mezi entitami

Mezi objekty neboli entitami v tabulkách existují tři základní druhy vztahů, které se zapisují do diagramů různými způsoby, přičemž by mělo být vždy jasné, o který vztah se jedná. Například na Obr. 3 jsou vyjádřeny pomocí symbolů klíče a nekonečna. Kardinality jsou tedy následující:

- Jedna ku jedné (1:1): každá n-tice z první tabulky odpovídá jedné nebo neodpovídá žádné n-tici z druhé tabulky a naopak.

- Jedna k mnoha (1:N): každý objekt první tabulky je spjatý s žádným, s jedním nebo i více objekty druhé tabulky. Zde záleží na tom, z jaké strany se na vztah díváme. Opačná verze je tedy N:1.
- Mnoho k mnoha (N:M): každá entita z první tabulky může být spojena s žádným, jedním nebo více entitami z druhé tabulky a naopak.

Poslední z trojice vztahů je pro databáze problém, který je v rámci ER diagramů nahrazován dvěma vztahy 1:N a třetí pomocnou tabulkou, přičemž je nutno dodržet, aby oba vztahy měli stranu „více“ u pomocné tabulky. Tuto náhradu lze vidět na Obr. 3, kde každá objednávka může mít jeden nebo mnoho produktů a zároveň každý produkt může být obsažen v žádné nebo více objednávkách. Proto byla vytvořena tabulka „detaily objednávky“ jako prostředník, u které je z každé strany symbol nekonečna, nahrazující v tomto případě písmeno N.

### 1.3.2 Normální formy

Kromě ER popisu databáze je potřeba podniknout další kroky pro zefektivnění struktury a odstranění některých problémů uvnitř tabulek. K tomuto účelu byla vytvořena pravidla souhrnně nazývána jako databázové normální formy. Nejdůležitější tři si nyní představíme detailněji.

**První normální forma** (1. NF) vyžaduje, aby všechny atributy tabulky byly dále nedělitelné (atomické), respektive, aby neobsahovaly žádné podstruktury. Příkladem atributu nesplňujícího 1. NF je jméno, pokud by se skládalo z křestního jména a příjmení. Pro splnění podmínky atomičnosti bychom museli vytvořit separátní sloupce pro obě části. Druhým příkladem jsou množiny hodnot. Jestliže jeden zaměstnanec by měl více telefonních čísel zapsaných v jednom atributu, nesplňovala by tabulka první normální formu. Korekce v tomto případě tkví v rozdělení čísel na mobilní, pracovní a telefon domů jako separátní položky [4].

**Druhá normální forma** (2. NF) potřebuje ke splnění, aby relace byla v 1. NF a zároveň platilo, že všechny atributy, které nejsou přímou součástí primárního klíče, jsou na něm závislé. Toto pravidlo se týká hlavně složených klíčů. Nesmí se tedy stát, aby například v tabulce, kde je primárním klíčem ID zaměstnance a divize, se vyskytoval atribut datum narození, který je zjevně závislý pouze na zaměstnanci [4].

Třetí normální forma (3. NF) si žádá, aby byly splněny podmínky 2. NF a zároveň tabulka nesmí neobsahovat takzvanou „tranzitivní závislost“. O té hovoříme, pokud je nějaký atribut nepřímo závislý na primárním klíči pomocí jiného neklíčového atributu. Představme si n-tici tabulky ve které je primárním číslem rodné číslo zaměstnance, dále obsahuje jméno, příjmení, zkratku divize a celý název divize. Název divize je v tomto případě závislý na primárním klíči zprostředkovaně (tranzitivně) přes zkratku divize [4].

## 1.4 Databázové jazyky

K tomu abychom mohli databázi smysluplně používat, je zapotřebí mít k dispozici prostředek schopný s daty manipulovat. Stejně jako existují jazyky programovací, tak i pro databáze vznikly jazyky, které stojí nad samotným úložištěm a operačním systémem. Tímto způsobem je práce s uloženými informacemi snazší, průhlednější a rychlejší. Ve světě se stal standardem jazyk SQL (structured query language), který se v různých verzích používá u většiny komerčních databázových systémů. Rozlišujeme tři hlavní typy jazyků: pro manipulaci, definici a kontrolu přístupu k datům. Ve skutečnosti ovšem všechny druhy bývají součástí jednoho a toto stejného jazyku.

### 1.4.1 DDL

Zkratka vychází a anglického „data definition language“. Stejně jako jiné programovací jazyky i tento funguje na principu zadávání příkazů, generujících patřičný výstup. V případě jazyku pro definici dat je výstup ukládán do slovníku dat (data dictionary), což je obvykle speciální typ tabulky nebo tabulek, ke kterým zpravidla mohou přistupovat jen systémoví uživatelé a která jsou jádrem celého schématu. Pomocí příkazů této části jazyka tedy vytváříme schéma databáze. DDL je také používáno pro upřesnění a nastavení některých detailů dat. Hodnoty, které jsou poté ukládány do vytvořené databáze, mohou podléhat pravidlům, která jsou taktéž popsána tímto jazykem [3].

Jedná se především o definice datových typu pro atributy (číselný, celočíselný, datum, textový řetězec a jiné). Dále o autorizaci, protože chceme zamezit přístupu k některým datům nebo znemožnit jejich změnu. Důležitá je taktéž referenční integrita, kde zjišťujeme, zda hodnota nějakého atributu v jedné relaci, se nachází v relaci jiné. Kupříkladu divizi nového zaměstnance bychom měli vybírat z množiny již existujících v tabulce divizí. Tři nejdůležitější příkazy jsou:

- Vytvoření objektu (CREATE)
- Modifikace (ALTER)
- Odstranění (DROP)

#### 1.4.2 DML

Jedná se o jazyk, umožňující uživatelům, administrátorům a analytikům přístup k datům jako takovým a manipulaci s nimi. Odtud zkratka DML (data manipulation language). Umožňuje zadávat nepřeberné množství příkazů od jednoduchých, které se týkají jen jedné tabulky až po složité dotazy obsahující mnoho podmínek a spojující několik tabulek. Základní čtyři akce, které s daty v této oblasti můžeme provést, jsou [3]:

- Čtení dat (v jazyce SQL se jedná o SELECT)
- Aktualizace (UPDATE)
- Vložení (INSERT)
- Smazání (DELETE)

#### 1.4.3 DCL

Příkazy *GRANT* a *SELECT*, které patří do této kategorie, slouží k definování práv pro uživatele. Přiřazujeme jim povolení k používání ostatních typů SQL příkazů pro určité objekty v rámci databáze. Mohou to být tabulky, pohledy, schémata, funkce a další.

### 1.5 Rozdíly OLAP a OLTP databází

Na dnešní databázové systémy se dá nahlížet ze dvou různých uživatelských pohledů. Na jednu stranu tu máme systémy, potýkající se denně s tisíci nebo i řádově více zápisy a změnami v tabulkách. Ať již jde o namarkované zboží zákazníka v supermarketu, převody peněz mezi bankovními účty nebo odbavenou položku ve firemním skladu. Na straně druhé jsou řídicí orgány společností a jejich komplikované selektivní dotazy, které jsou náročné na zpracování a čas potřebný k jejich zodpovězení v rámci DBMS.

#### 1.5.1 OLTP (online transaction processing)

Tyto databázové systémy slouží převážně pro zaznamenání každodenních činností. Aplikací běžících na databázích tohoto typu je nepřeberné množství. Ve stejném okamžiku k datům přistupují kvanta uživatelů, využívající systém pro čtení, změnu a zápis nových dat. Obsahují velké množství většinou menších tabulek, které jsou

normalizované podle prvních tří normálních forem. Strukturovanější analytické dotazy nad těmito systémy jsou velmi náročné. Stejně tak vytváření různých sum, průměrných hodnot, uchovávání minimálních a maximálních hodnot v rámci zpracování každé transakci by jen zpomalovalo práci s aplikací, což by vedlo k nespokojenosti koncových uživatelů OLTP, kterých je v podniku většina [2].

### 1.5.2 OLAP (online analytical processing)

Typicky jsou tyto systémy využívány menším množstvím uživatelů, jejichž dotazy data převážně čtou. Ke změnám by mělo docházet jen zřídka. Počet tabulek bývá v řádu jednotek, jsou zcela denormalizované a nejsou mezi nimi klasické relační vztahy, ale dělí se na tabulky faktů a dimenzí. Využívá se zde takzvaných multidimenzionálních struktur, kam ukládáme již předem upravená data z různých zdrojů [2].

Znak	OLTP databázové systémy	OLAP databázové systémy
Počet uživatelů	Tisíce	Stovky
Přístup k datům	Ke stovkám záznamů, v módu čtení i zápisu	K milionům záznamů, převážně čtení
Cíl	Podle aplikace	Podpora rozhodování
Data	Detailní, jak numerická tak i alfanumerická	Sumy, převážně numerická
Časové rozpětí	Pouze současná data	Současná a historická
Aktualizace	Kontinuální	Periodická
Model	Normalizovaný	Denormalizovaný, multidimenzionální
Pracovní náplň	Předvolené transakce	Specifické analytické dotazy

Tabulka 1: Rozdíly mezi OLAP a OLTP (Upraveno dle: 5, str. 6)

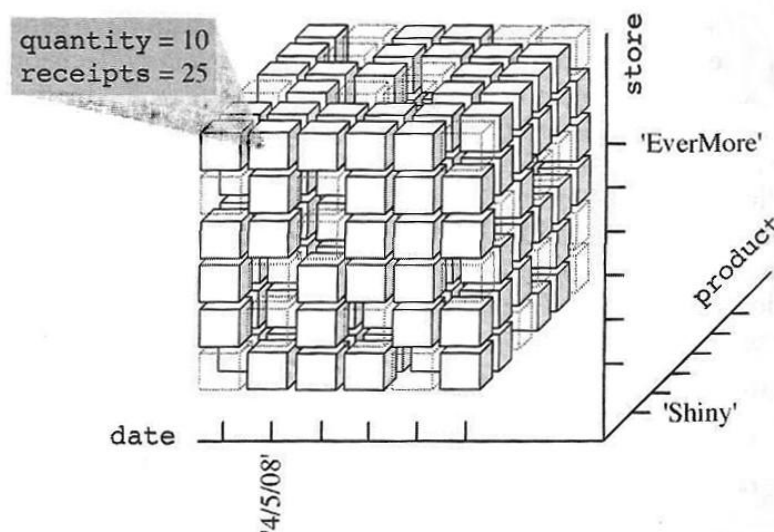
## 1.6 Multidimenzionální databáze

Pro systémy sloužící k analýze podnikových dat nejsou OLTP databáze příliš vhodné. Vzhledem k obrovskému počtu tabulek, normalizaci a jejich provázanosti by trvalo velice dlouhou dobu procházet všechna data k získání užitečných informací. Zvláště pak s přihlédnutím k tomu, že je většinou potřeba shromáždit data z více zdrojů, nejen z vnitropodnikových databází.

Multidimenzionální model vychází z pozorování, které ukazuje, že faktory ovlivňující rozhodovací proces jsou specifické skutečnosti (fakty) pro dané odvětví jako například prodeje, zásilky, reklamace atd. Každá instance takového faktu odpovídá výsledkům

nějakých událostí nastalých v podniku. Událost si můžeme představit jako prodej výrobku, naskladnění materiálu nebo dopravené zboží. Fakta jsou popsány pomocí měřených hodnot vyjadřujících kvantitu událostí nebo součet hodnot s událostmi spjatých [5].

Nejklasičtější pojetí vícerozměrných databází je pomocí kostek, kde každá hrana odpovídá nějaké dimenzi, a události jsou představovány buňkami uvnitř kostky. Ve skutečnosti může být použito i více než tři dimenze. V takovém případě hovoříme o hyperkostkách, ale pro lepší představu si představíme typický příklad třídímního modelu, kde jednotlivé dimenze jsou obchod, produkt a datum. Údaje se nalézají v průnicích dimenzí, jak je tomu patrné z obrázku Obr. 4, kde vidíme, že produktu s názvem Shiny se prodalo v obchodě EverMore 10 kusů za celkových 25 dolarů dne 4/5/2008 [5].



Obr. 4: Třídímní krychle (Zdroj: 5, str. 20)

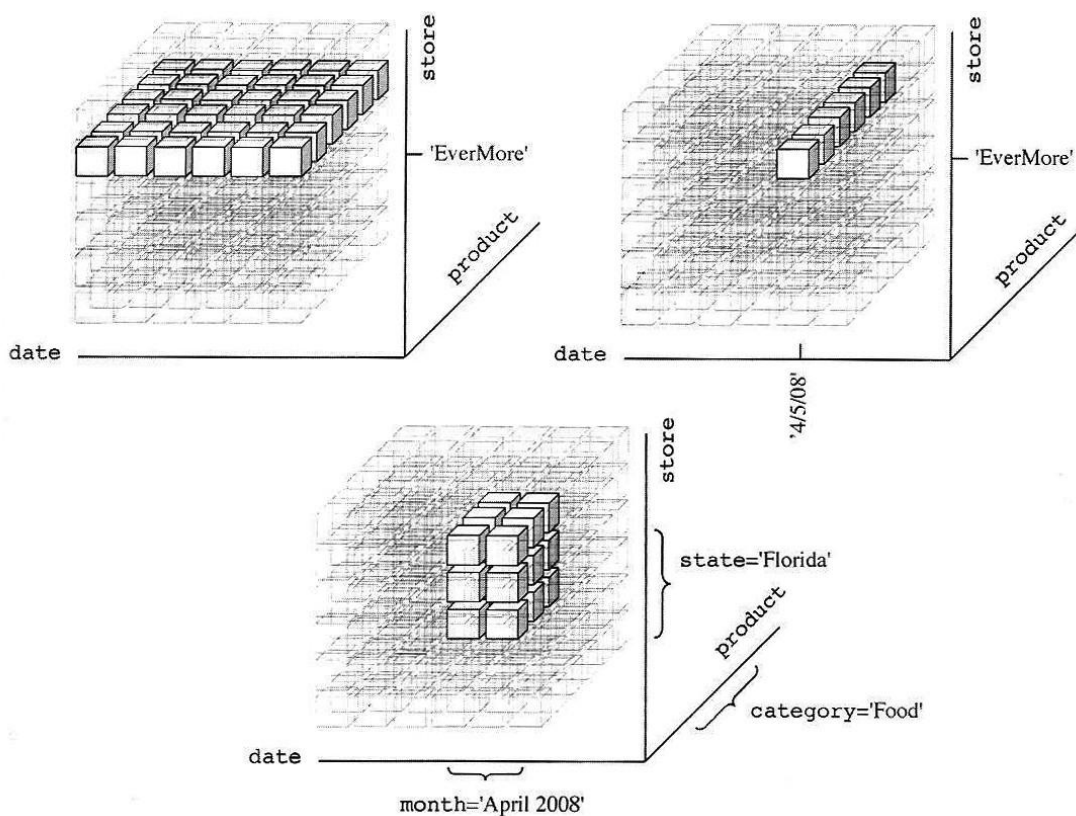
### 1.6.1 Restrikce a agregace

Protože množství dat ve větších společnostech je obrovské, je nutné se na multidímní kostky dívat z určitého pohledu. To nám umožňují restrikce a agregace.

Agregace sčítají data obsažená v agregovaných událostech. Pokud by uživatel nezájmal denní pohled na prodej určitého produktu, může použít pohled měsíční, nebo

na další úrovni pak roční atd. Agregovat se dá podle více různých dimenzí ve stejném čase, protože zajímavá informace může pro někoho být počet prodaných produktů z kategorie potravin během roku, nehledě na obchod, v němž byl prodej zaznamenán (agregujeme podle produktu a času) [5].

Restrikce se používá pro výběr určitých částí kostky. Nejjednodušším typem selekce je „slicing” neboli řez kostkou. Když vybereme konkrétní obchod, dostaneme jednu vrstvu, pokud vybereme konkrétní hodnoty pro všechny tři dimenze, dostaneme jednu buňku uvnitř kostky. Druhým typem výběru je takzvaný „dicing”. Geometrickým výsledkem je pak menší krychle nebo kvádr, který vznikne, pokud výběr neomezíme pouze na jeden konkrétní produkt, datum nebo obchod, ale vybereme více hodnot [5].



Obr. 5: Slicing a dicing (Zdroj: 5, str. 23)

## 1.7 Podnikové informační systémy

Hlavní náplní databázových systémů je předávat data uživatelům skrze různé aplikace. Běžný koncový uživatel nechce ba ani nemůže trávit čas studováním jazyka SQL, aby

mohl několika korektně vytvořenými dotazy získat informace, po kterých toužil. Proto ve firmách existuje nepřehledné množství aplikací, jakožto prostředníků mezi databází a člověkem (zaměstnancem). Patří sem vše od jednoduchých webových rozhraní až po robustní informační systémy (IS).

Podnikový IS je tvořen lidmi, ať již přímo zaměstnanci nebo externími pracovníky, kteří za pomoci technologií a postupů, jež jsou v danou chvíli dostupné, zpracovávají podniková data z různých zdrojů a z nich poté získávají informace pro budování informační a znalostní základny společnosti. Tyto znalosti jsou dále používány pro rozhodování, tvoření strategií a řízení procesů organizace [6].

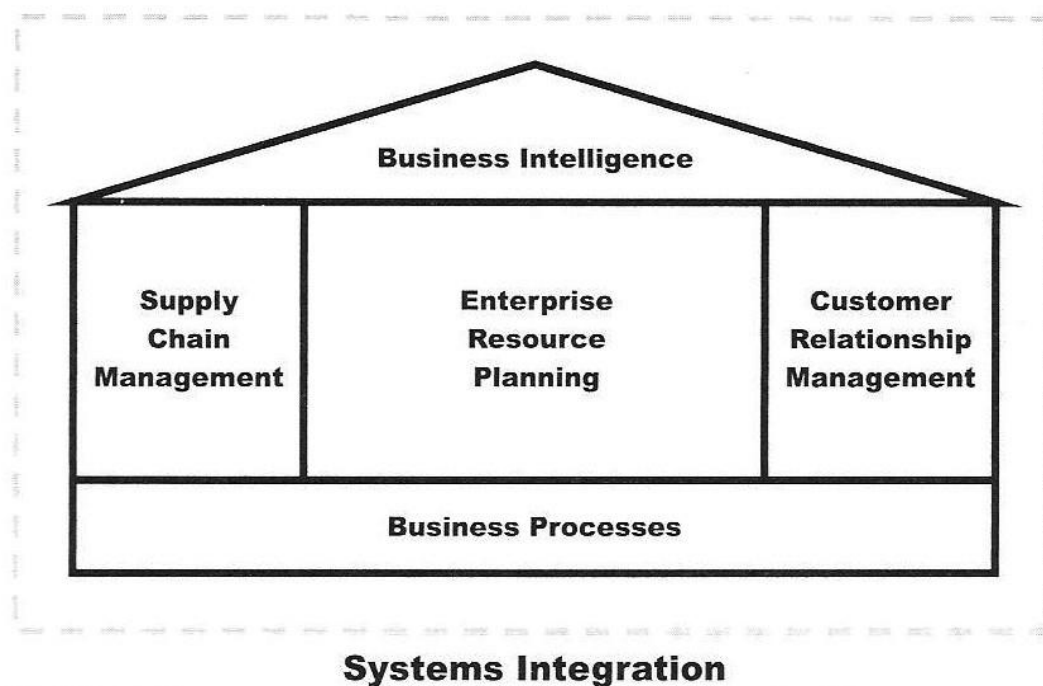
### **1.7.1 Součásti IS**

Uvnitř organizace se udává mnoho akcí, od nákupu materiálu po fakturaci prodaných výrobků nebo služeb. Všechny procesy, které v podniku probíhají, musí být informačním systémem pokryty. Aby nenastal v systému zmatek, rozděluje se do tematických okruhů, které by měly být navzájem efektivně provázané. Stěžejní části jsou:

1. ERP (enterprise resource planning) - jádro systému sloužící pro řízení hlavních interních procesů podniku, u nichž je vlastníkem samotná společnost. Spolupracuje s ostatními částmi IS
2. CRM (customer relationship management) - tato část se zaměřuje na řízení vnějších procesů, kde jsou spoluvlastníky zákazníci společnosti. Mezi vybrané funkce patří řízení kontaktů, marketingových a obchodních vztahů
3. SCM (supply chain management) - jak je z názvu patrné, primární činností sekce je vedení vnějších procesů, které jsou z části vlastněny odběrateli nebo dodavateli podniku. Poskytuje možnosti pro integraci společnosti do dodavatelského řetězce
4. BI (business intelligence) nebo BW (business warehouse) - zde se shromažďují data z ostatních oblastí informačního systému a někdy i z vnějších zdrojů, sloužící poté k analýzám pro vrcholové manažery [6]

Každý z nich se pak ještě dále dělí na menší podpůrné moduly, kterých mohou být desítky i stovky. Existuje i řada personalizací na míru umožňujících spojení již

používaných systémů s nově implementovanými. To se hodí v případě, že firma je spokojena s určitou částí svého IS/IT nebo ji chce mít zcela mimo pro snadnější kontrolu, a proto ji jen naváže, ale nepohlí uvnitř například ERP.



Obr. 6: Jeden z pohledů na podnikové informační systémy (Zdroj: 6, str. 78)

## 1.8 Typy zavedení aplikací

V dnešní době se čím dál více objevují nové směry umožňující firmám rychlou instalaci a téměř okamžitý provoz požadovaného softwaru nebo hardwaru. Existují dva hlavní pohledy, popsané v následujících podkapitolách.

### 1.8.1 On-premise

Zavedení pomocí on-premise znamená, že hardware nebo softwarové licence jsou zakoupeny a vlastněny společností, která je chce ke svému podnikání používat. Jednak zde hovoříme o tzv. „bare metal“, což je aplikace nainstalovaná přímo na fyzický server (respektive OS na něm běžícím) ve vlastnictví společnosti. Jako alternativní řešení se nabízí virtualizace pomocí speciálních operačních systémů, umožňující efektivnější využití technického vybavení serverů.

V souvislosti s on-premise se můžeme setkat s pojmem *appliance*. Jedná se o balík hardwarových a softwarových prostředků, které jsou určeny pro určitou cílovou

aplikaci, jež si zákazník přeje zavést. Většinou jsou všechny komponenty předem testovány a certifikovány, tak aby tvořily skvěle fungující sestavu, do níž se nesmí bez souhlasu dodavatele přidávat vlastní prvky, jinak by nemohl dodavatel garantovat stoprocentní funkčnost a záruku spokojenosti.

### **1.8.2 Cloud computing**

„Cloud computing představuje model poskytování aplikací prostřednictvím Internetu v podobě služby a rozvíjí předchozí modely o možnosti grid computingu a clusterových řešení. Cloud computing v podstatě zastřešuje několik specifických typů služeb” [6].

Patří sem:

- SaaS - pronájem konkrétní aplikace
- IaaS - pronájem infrastruktury
- PaaS - pronájem celé platformy pro provoz aplikací [6]

### **1.8.3 Možnosti rozšíření**

Ať se bavíme o *cloudu* nebo *on-premise* řešení, vždy se na konci nachází fyzické servery. Pokud jde pouze o jeden, zastávající funkci zdroje výpočetního výkonu pro konkrétní aplikaci hovoříme o *single-node*. Jestliže dojde v čase k okamžiku, kdy výkon nebo kapacita není již dostatečná, máme dvě možnosti rozšíření.

*Scale-up* znamená, že přidáváme komponenty přímo do serveru, čímž získáme dodatečný výkon. Zatímco *scale-out* je rozšiřování pomocí nového, většinou stejného serveru a zapojení k serverům již existujícím.

## **2 ANALÝZA SOUČASNÉHO STAVU**

Analytická část poskytuje seznámení se společností a zhodnocení současného stavu, z něhož je patrný důvod zadání projektu. Je zde prozkoumána nynější struktura informačního systému, použitého hardwaru a softwaru. Dále je zde představen koncept databází běžících v operační paměti na příkladu SAP HANA.

### **2.1 Profil společnosti**

Vzhledem k vnitřním nařízením a podepsané smlouvě se společností, ve které pracuji jako SAP HANA administrátor, nemohu prozrazovat, jaké firmy mají podporu smluvně sjednanou u mého zaměstnavatele. Proto zde neuvedu přesný název zadavatele ani jeho sídlo. Zdrojem zde uvedených údajů byly vnitřní materiály společnosti.

Projekt, o němž pojednává tato práce, byl zadán mezinárodní společností, která se převážně zaměřuje na oblast zdravotnictví. Hlavní pole působnosti z hlediska lokace je Evropa a Rusko. Její pobočky bychom našli téměř v každém státě střední a západní Evropy. Jedná se o významného hráče ve svém oboru podnikání, pomalu se rozrůstajícího do celého světa.

#### **2.1.1 Hlavní činnosti**

Zdravotnictví je velmi obecný pojem, proto musím upřesnit v jakých konkrétních oblastech se společnost nejvíce realizuje. Jedná se o vývoj a prodej farmaceutik a jednotlivých složek používaných pro jejich výrobu. Majoritní část financí přitéká do podniku z prodeje takzvaných generických léků, což jsou léky přesně kopírující složení původních patentovaných originálů a majících stejné účinky. Při jejich prodeji se sice potýkají s velkým počtem suplementů, ale jdou na odbyt ve velkém množství, jako jsou například prášky proti bolesti, na spaní, proti horečce a jiné. Druhou největší část, tvořící společně s generickými přípravky zhruba 90% tržeb jsou léky značkové, kde je vlastníkem patentu přímo společnost, která je prodává. Z pohledu lokace je stěžejní Evropa. Více než 95% odbytu se uskutečňuje v centrální a východní části, zbytek do sta procent doplňují pobočky v Asii a Pacifiku.

### 2.1.2 Klíčové údaje

Při srovnání mezi lety 2013 a 2012 se tržby zvýšily o 10% na 2 miliardy €. Počet zaměstnanců dosáhl hodnoty téměř deseti tisíc. Vlastních produktů uvedených na trh během roku bylo přes 700 v obou sledovaných obdobích. Akcie společnosti zaznamenaly taktéž razantní zvýšení o 47%.

Výsledek hospodaření před zdaněním, amortizací, odečtením úroků a odpisů stoupl o 18%. Tento ukazatel (EBITDA) se používá především v zahraničí pro srovnání firem na mezinárodní úrovni, a to z toho důvodu, že každý stát má různá pravidla a podmínky, takže by docházelo ke značnému zkreslení tohoto výsledku. U nás se nejčastěji setkáváme s ukazatelem EBIT (výsledek hospodaření před zdaněním a úroky), který odpovídá provoznímu výsledku hospodaření z výkazu zisků a ztrát. Přičemž platí rovnice  $EBITDA = EBIT + \text{odpisy}$  [7].

### 2.1.3 Strategie

Hlavní směry, kterými se firma dále chce ubírat a faktory, na které se hodlá převážně v budoucnu zaměřit, jsou:

- Dlouhodobě rostoucí trhy a zrychlený růst na nově se vyvíjejících trzích
- Rozšíření portfolia generických léků a propagace vlastních značkových léků
- Stop neúspěšným projektům, důkladnější sledování nákladů na vývoj nových produktů
- Efektivní, přesné a rychlé generování reportů a zkrácení procesu stanovení rozhodnutí

Právě s posledním z těchto cílů má pomoci projekt, o kterém pojednává tato práce. Pomocí moderních IT technologií chce společnost zkrátit dobu pro rozhodování, jehož proces je velmi důležitou částí, která rozhoduje o valné většině financí, které mohou do podniku v budoucnu přitéci.

### 2.1.4 SWOT analýza

Používá se velmi často pro vytvoření základního pohledu na firmu. Sleduje jak interní záležitosti a vztahy, tak i externí vlivy působící ať již přímo či nepřímo na běžný chod společnosti. Základní rozdělení SWOT analýzy je tedy:

- Vnitřní faktory (silné a slabé stránky)
- Vnější faktory (příležitosti a hrozby)

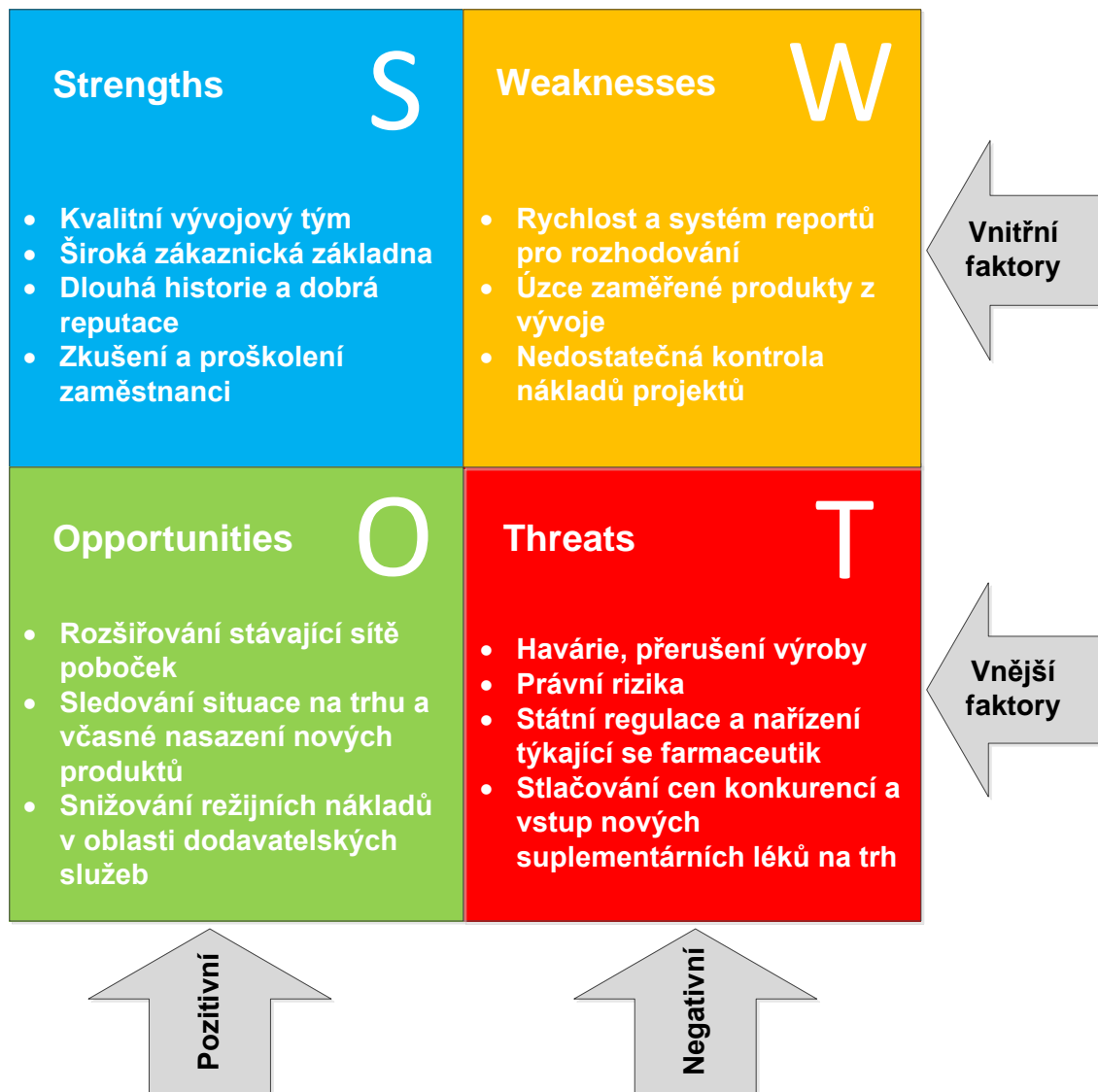
Mezi silné stránky společnosti patří vynikající tým vývoje nových druhů léčiv a farmaceutických látek obecně. Díky tomu bylo podáno mnoho patentů, které umožnili v určitých oblastech medicíny prodávat léky na konkrétní obtíže téměř bez konkurence. Kvalifikovaní a zkušení zaměstnanci jsou neméně podstatným faktorem pro správné fungování organizace. Důležitá je i skutečnost, že společnost funguje více než sto let, během kterých si vybudovala dobré jméno.

Slabé stránky reprezentuje pomalejší zpracování reportů, už vzhledem k množství dat. Organizace se snaží zaměřit na nákladový management a proto je rychlost důležitá. Mnoho produktů je určeno pouze pro velice úzkou skupinu pacientů s určitou chorobou. Bohužel tento problém se nedá snadno odstranit. Vývoj produktů je velmi nákladný, na tuto oblast si chce vedení více „posvítit“.

Příležitost, která se naskýtá, je rozšiřování mezinárodní sítě obchodů. Díky tomu by společnost nebyla závislá pouze na jednom či dvou hlavních trzích, což je velice nebezpečné především v oblasti farmaceutik, kde často zasahuje stát. Druhou příležitostí, kterou zmíním, je pečlivé sledování trhu a trendů pro včasné nasazení nových léků a to především těch negenerických, které nepodléhají tolik regulacím a je na nich vyšší marže. Další možností, kde vylepšit výsledky hospodaření je snižování nákladů na prodej tím, že přesune tržní riziko více na dodavatele a bude využívat ve větším množství služeb zemí s nízkými náklady na dopravu.

Mezi rizika, kterým musí firma čelit, patří možnost přerušení výroby vlivem špatného plánování, poruchy nebo havárie ve výrobních halách. Tato skutečnost by se nejen špatně podepsala na ziscích a jménu společnosti, ale hrozí zde únik nebezpečných látek, ohrožení zdraví zaměstnanců a při distribuci léku se špatným složením, byť jen o setiny gramu by mohla mít nedozírné následky. Hrozbou se můžou také stát soudní spory, v případě, že nedojde k dostatečné kontrole všech právních norem a patentů zvláště pak u generických léků. Tento problém může nakonec vyústit zakázáním prodeje určitého produktu a obrovskými mimosoudními vyrovnáními. Regulace cen ať již nařízené evropskou unií nebo konkrétním státem představují pro farmaceutické společnosti velký

problém. A nejedná se pouze o regulaci cen, ale také schvalování nebo zamítnutí použití určitých léků na národním trhu. Oblast farmaceutik je velmi citlivá na ceny, marže jsou stlačovány a konkurence je velká. Taktéž nesmíme opomenout časté vstupování nových firem na trh s generiky.



Obr. 7: SWOT analýza (Zdroj: vlastní zpracování)

## 2.2 IT infrastruktura

Společnost se rozhodla, jak již bylo řečeno, používat produkty společnosti SAP AG pro veškeré své části informačního systému. Ovšem každý software musí zákonitě mít i hardwarovou vrstvu a většinou i vrstvu operačního systému.

### 2.2.1 Hardwarová vrstva

Virtualizované servery, na které byly aplikace nainstalovány, jsou logické oddíly rozdělené pomocí technologie PowerVM na architektuře Power7, obojí od společnosti IBM. Každé takové virtualizované jednotce může být přidělen nebo odebrán výpočetní výkon z hardwaru jednotlivých serverů, které jsou v tomto případě modelového typu 9117-MMB.

### 2.2.2 Vrstva operačního systému

Operační systém nainstalovaný na většině těchto serverů je IBM AIX, konkrétně ve verzi 7.1. Ten je aktualizován pravidelně pomocí „service packů“ a „technology levelů“. Jedná se samozřejmě o 64 bitový systém běžící na *Power* procesorech. Tento systém je velmi stabilní a nabízí jak komfort pokročilých nástrojů pro správu, tak široké možnosti velmi detailního nastavení. Zejména pak způsob práce s disky a vytváření diskových skupin je velmi přehledné a intuitivní. Samozřejmostí je pak podpora tohoto operačního systému společností SAP, jako platformy pro své aplikace.



Obr. 8: IBM Power 770 (9117-MMB) (Zdroj: 8)

### 2.2.3 Databázová vrstva

Jediná databázová platforma, kterou doposud používali uvnitř firmy, je IBM DB2. Jedná se o další z řady produktů korporace IBM s dlouholetou historií. Konceptně spadá do klasického relačního databázového softwaru, využívajícího jazyka SQL, logování transakcí a kontejnerů jako místa pro zápis dat. Důležitou vlastností je pak výborná integrace do podnikových aplikací SAP.

## **2.3 Rozbor informačního systému**

Podnik nyní používá pro své stěžejní informační systémy řešení od společnosti SAP AG, které v současné podobě, pouze s drobnými obměnami a samozřejmými aktualizacemi, funguje ve společnosti již řadu let. Za rozhodnutím výběru jejich softwaru a know-how stály možnosti budoucího rozšíření o různé prvky, široké spektrum zákazníků a existující technologické postupy pro migrace, systém transportů změn a výborná integrace jednotlivých produktových částí do jednoho funkčního celku.

### **2.3.1 SAP**

SAP má více než 42 let historie v oblasti podnikových informačních systémů. Byl založen bývalými zaměstnanci IBM, kterým se nepodařilo prosadit svůj nápad, a proto vytvořili společnost Systems applications and products in data processing (SAP). Jejich hlavní hnací myšlenkou bylo sestavit real-time transakční systém, což se jim záhy podařilo. Netrvalo dlouho a získali si přízeň mnoha zákazníků.

Dnes jejich aplikace a služby využívá přes 282 000 zákazníků napříč 190 zeměmi světa. Od svého založení v roce 1972 se tato firma se sídlem ve Walldorfu v Německu rozrostla do 120 států a zaměstnává 74 406 pracovníků. Její příjem za rok 2014 činil 17,6 miliard € (o 5% více oproti předchozímu roku). Existující produkty pokrývají 26 průmyslových odvětví. V současné době se zaměřují převážně na in-memory computing, mobilní technologie a cloud [9].

### **2.3.2 SAP Business Suite**

Produkt, který byl zakoupen (licence), se jmenuje SAP Business Suite. Ten v sobě zahrnuje soustavu aplikací pokrývajících všechny hlavní procesy probíhající ve firmě. Obsahuje aplikace nejen pro zajištění provázanosti s dodavateli, tak i vztahy se zákazníky, práci s lidskými zdroji, výplatami a jiné. Tyto aplikace jsou postaveny na platformě SAP NetWeaver (dále jen NW), psané ve vyšším programovacím jazyce ABAP, taktéž vytvořeným firmou SAP, nebo v jazyce Java.

### **2.3.3 SAP ECC**

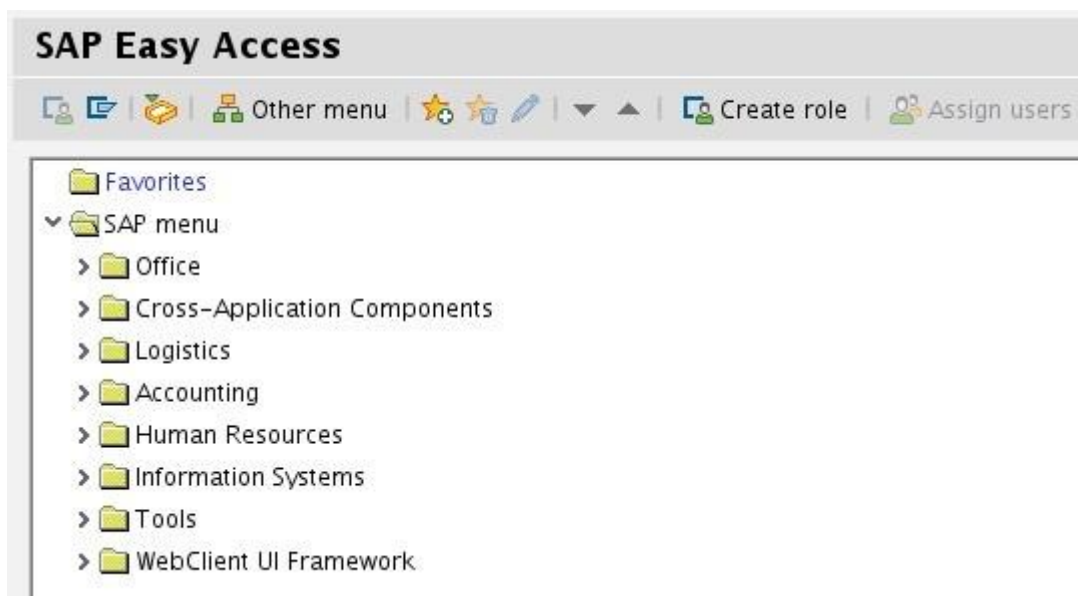
Stěžejní součástí SAP Business suite je SAP ECC 6.0 (SAP ERP Central Component), dříve známá jako SAP ERP. Ten se skládá z několika komponent (SRM, SCM, CRM a

další), přičemž ve středu informačního systému se nachází ERP. SAP ECC je proto typickým příkladem centralizovaného systému. Výhody oproti decentralizovaným systémům jsou:

- Jedna verze dat přístupná všem oddělením
- Odstranění redundantních dat
- Jednodušší monitorování a analýza
- Větší průhlednost a snazší kontrola konzistence dat

Software SAP ECC je dále rozdělen do modulů, které slouží ve firmě jednotlivým oddělením pro přístup k datům. Základní řešení jsou:

- Human Capital Management (lidské zdroje)
- Operations (výroba, nákup, logistika)
- Corporate Services (správa majetku, nemovitostí, řízení kvality)
- Financials (kontrola, analýza, plánování) [10]

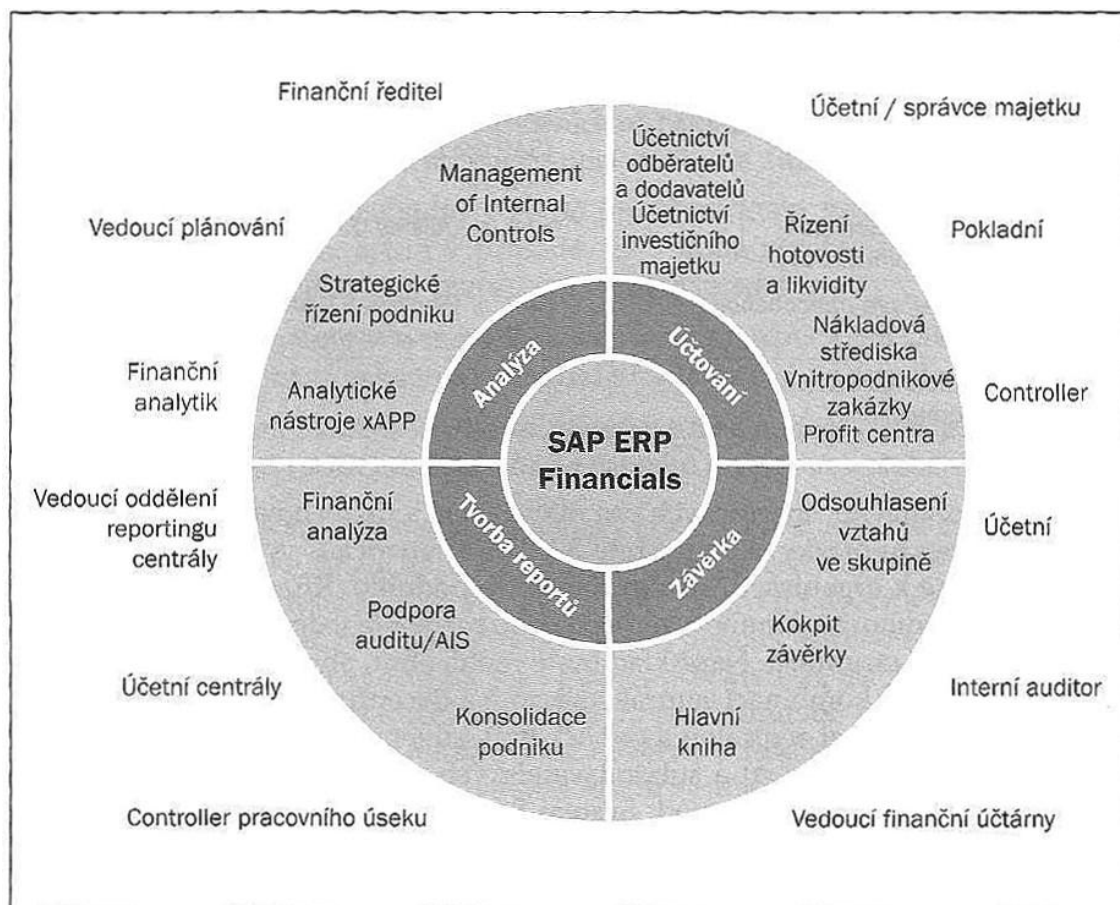


Obr. 9: Nainstalované komponenty SAP ERP (Zdroj: vlastní zpracování)

Klíčové aplikace společnosti, která projekt zadala, mají svá data uložená v databázi DB2 od firmy IBM a servery hostující jak databáze, tak aplikace jsou od stejného dodavatele, přičemž operační systém používají AIX. Co si organizace slibuje pod zavedením databáze SAP HANA je jednak její samotný účel (viz název této práce) a za druhé vyzkoušet tuto poměrně novou technologii jakožto možného náhradníka stávajícího DBMS v budoucnu i pro ostatní části podnikového informačního systému.

### 2.3.4 ERP Financials

Abychom pochopili podstatu projektu a proč se firma rozhodla změnit stávající systém, je nutné si blíže popsat, z čeho se skládá část Financials, jejich používaného ERP softwaru. Protože i zde existuje další členění na moduly a ty se v zásadě dají rozdělit do čtyř základních skupin (viz Obr. 10). Společnost se rozhodla zaměřit se na část v oblasti tvorby reportů a to konkrétně na analýzu ziskovosti.



Obr. 10: SAP ERP Financials (Zdroj: 10, str. 26)

Díky dnešním podmínkám je snazší vstupovat na nové zahraniční trhy, proto i tato firma se snaží zakládat nové pobočky v různých zemích. Avšak to sebou nese i značné výzvy v podobě různých očekávání zákazníků, finanční situace uvnitř státu, jiné druhy regulací, měnové kurzy, daňové zatížení apod. Proto vznikla potřeba podrobného sledování a získávání informací použitelných pro rozhodování, rozříděných podle zákazníků, segmentů a/nebo výrobků [10].

System nabízí dva základní přístupy k analýzám ziskovosti. Podle toho jsou pak používány různé hodnoty. V první řadě je to účetní analýza ziskovosti (account based), která je založena na práci s jednotlivými účty, tím se zde myslí druhy nákladů a výnosů tak jak jsou vytvořeny v systému SAP. To přináší výhodu snadnějšího srovnání s finančními výkazy, ale značně limituje při tvorbě složitějších reportů.

Proto ERP Financials nabízí druhou možnost použití odlišných hodnot pro náklady a výnosy a tou je kalkulační analýza ziskovosti (costing based). Nabízí totiž nepřeberné množství nastavení charakteristik a hodnotových polí, kde hodnotová pole jsou náklady, výnosy, dedukce výnosů a jiné číselné hodnoty, které zákazníka zajímají, zatímco charakteristiky nám mění úhel, z něž lze na tyto hodnoty pohlížet (zákazník, zákaznická skupina, produkty).

Na základě kalkulací		Na základě účtů	
<i>Hodnotová pole</i>		<i>Nákladové a výnosové elementy</i>	
Výnos	1 000 000	800000 Výnos	1 000 000
Snížení výnosu	100 000	808000 Snížení výnosu	100 000
<b>Čistý výnos</b>	<b>900 000</b>	<b>Čistý výnos</b>	<b>900 000</b>
Var. materiálové náklady	400 000	893000 Prodejní náklady	690 000
Var. výrobní náklady	190 000	231000 Účtování kurzových rozdílů	10 000
Odchylky ve výrobě	10 000	651000 Výzkum a vývoj	10 000
<b>Krycí příspěvek 1</b>	<b>300 000</b>	671000 Marketing	50 000
Materiálová režie	50 000	655000 Administrativní náklady	40 000
Výrobní režie	50 000		
<b>Krycí příspěvek 2</b>	<b>200 000</b>		
Výzkum a vývoj	10 000		
Marketing	50 000		
Administrativní náklady	40 000		
<b>Krycí příspěvek 3</b>	<b>100 000</b>	<b>Hospodářský výsledek</b>	<b>100 000</b>

Tabulka 2: Dva různé pohledy na analýzu ziskovosti (Upraveno dle: 11)

Právě posledně jmenovaný přístup používá i společnost zadávající projekt. Přičemž používá jednu hlavní tabulku obsahující všechny položky a jejich důležité vlastnosti, se kterými bylo manipulováno, což představuje více než 380 milionů řádků. K tomu dvě další menší s doplňujícími informacemi. Nad nimi je pak vytvořena finální soustava reportů, vytvořená interními analytiky. Cílem projektu je pak tyto reporty urychlit.

## 2.4 Technologie in-memory databáze

HANA je nejnovější databázový systém od společnosti SAP, který používá pro uložení dat operační paměť serveru namísto pevných disků. Byl jedním z prvních hráčů na trhu databází, který představil koncept in-memory. Dlouhou dobu byla limitována výběrem platformem certifikovaných pro běh SAP HANA. Hlavním důvodem byla garance rychlosti pro koncové zákazníky, proto veškerá *appliance*, případně zákazníkem naplánovaná sestava musí být prověřena aplikací Hardware Configuration Check Tool (HWCCT), zda splňuje KPI (key performance indicator) stanovené společností SAP.

Byla vytvořena jako reakce na vysoké nároky zákazníků na rychlost zpracování dat z tabulek o stovkách milionů záznamů, které dnes běžně můžeme vidět. Nejedná se pouze o DBMS, ale i soustavu aplikací, knihoven a funkcí, které umožňují efektivní správu nebo vývoj webových aplikací, pohledů, profilů a dalších částí.

V současnosti kromě certifikovaných *appliance* na platformě Intel x86 se v budoucnu počítá i s Power architekturou. Mimo „on-premise” nabízí i několik cloudových licencí podle velikosti aplikačních dat. Co se operačního systému týče z prvotního Suse Linux Enterprise Server for Sap Applications (zkráceně SLES for SAP) se možnosti rozrostly o Redhat Linux, přičemž je povolena virtualizace od společnosti VMware v podobě produktu ESXi. Pravděpodobně se dočkáme i povolení jiných unixových operačních systémů během tohoto roku.

### 2.4.1 SAP HANA Studio

Jednou z aplikací, které patří do skupiny doplňků k databázi HANA je SAP HANA Studio. Jedná se o hlavní administrativní prostředí, které v sobě ukrývá řadu dalších modulů. Díky tomuto nástroji je veškeré nastavení možno provádět z jednoho místa a to i pro takřka neomezený počet databází, které lze jednoduše přidávat, třídit do složek a skupin nebo vytvářet připojení do jedné databáze pomocí různých uživatelů. Studio je založeno na platformě Eclipse a lze jej stáhnout jak pro Windows, tak i pro Linux a Mac OS. Protože seznam aktivit, jež je možno ze studia provádět je velmi rozsáhlý, existují čtyři základní předdefinované pohledy:

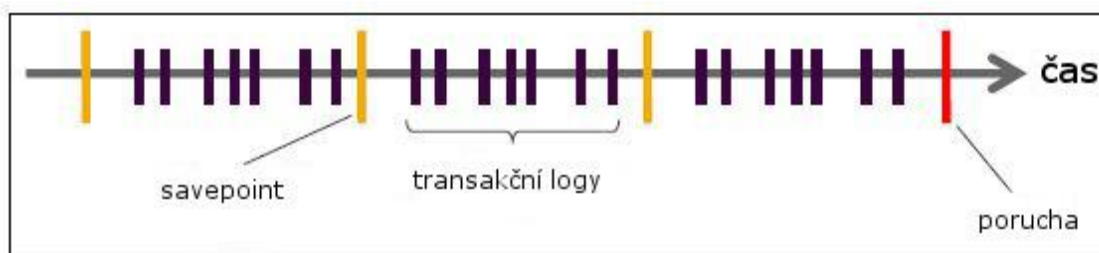
- Modelovací perspektiva - nabízí prostředí pro definování analytických modelů (například atributové, analytické a kalkulační pohledy)
- Vývojová perspektiva - slouží pro vývoj aplikací pro webový engine SAP HANA XS, vytváření projektů, balíčků nebo třeba rolí
- Debug perspektiva - obsahuje menu a pohledy pro odlaďování a testování aplikací
- Administrativní perspektiva - nabízí rozsáhlé možnosti pro snadné plnění úkolů administrátora [12]

#### 2.4.2 Perzistence

Protože operační paměť je nestálá a v případě poruchy nebo výpadku proudu přicházíme o veškerá data do ní zapsaná, musí SAP HANA řešit problém ztráty dat a zotavení. Proto i zde musí server disponovat klasickými nebo SSD disky, jakožto relativně bezpečné místo pro uložení obsahu databáze.

V základním nastavení probíhá každých pět minut proces tvorby takzvaného „savepoint“ na pevné disky. Vedle toho neustále zapisuje transakční změny do logů jako je tomu u ostatních databází, taktéž na perzistentní úložiště. Aby byl zápis logů co nejrychlejší, využívají výrobci certifikovaného hardwaru pro databáze SAP HANA v praxi speciálních flash karet nebo SSD disků. Pro zvýšení bezpečnosti jsou servery vybaveny hardwarovým řadičem diskových polí a používá se RAID5 s jedním hot-spare diskem.

Vytváření savepointů není jen prosté zapsání kompletního obsahu paměti na disky, v takovém případě bychom přišli o výhody a rychlost operační paměti, pokud bychom museli pokaždé čekat na zapsání. Proto se do těchto obrazů ukládají pouze změny od posledního obrazu, a to asynchronně. V případě poruchy a následného restartu (za předpokladu, že jsme nepřišli o disky) se nejprve načte poslední savepoint a pak se zrekonstruuje transakce z logů zapsaných od savepointu do času přerušení chodu databáze.



Obr. 11: Perzistence (Upraveno dle: 13, str. 4)

Pro zajištění maximální bezpečnosti jsou nutné zálohy logů a celé databáze. První možnost, kterou HANA nabízí, je zapisovat zálohy lokálně na server a poté použít komerční software třetí strany pro přesunutí souborů nejlépe mimo datacentrum, kde databáze běží. Druhou možností je pak zálohu posílat přímo otevřeným *socketem*. Zjednodušeně se dá říci, že data jdou přes síťové spojení z databáze přímo na pásy cílového serveru. Tento systém se nazývá „backint” a musí být podporovaný zálohovacím programem.

### 2.4.3 Column-based tabulky

Protože většina analytických dotazů sleduje data pouze několika atributů tabulky z mnoha, vznikl koncept sloupcového uložení. K operační paměti se totiž přistupuje sekvenčně, čili data jsou zapsána v řadě za sebou a čteme je nejčastěji po částech o velikosti dané velikostí zásobníku (*cache*). Například tabulka CE1ST00 obsahující jednotlivé položky prodeje, která je součástí SAP ERP Financials zákazníka, obsahuje v tuto chvíli (31. 3. 2015) přesně 381 839 812 záznamů. Pokud by aplikace položila dotaz, který by se dotýkal pouze sloupce USNAM s uživatelským jménem, museli bychom v row-store pojetí projít celou tabulku a načíst postupně všechny záznamy a jejich atributy, i když nás jejich obsah vůbec nezajímá. V případě column-based tabulky, kde jsou v paměti zapsány sloupce jeden za druhým, stačí přečíst pouze sloupec USNAM a nad ním provést žádanou operaci. Proto konečné množství dat, nutné načíst pro zodpovězení dotazu se může zmenšit až o několik řádů, jak je vidět na Obr. 12.

SELECT COUNT (USNAM) FROM SAPERQ.CE1ST00 WHERE USNAM='NOVAKJ'

Row-store	MANDT	VRGAR	USNAM	HZDAT	WADAT	...
1.řádek						
2.řádek						
3.řádek						
....						
381,839,812.						

Pokud jeden řádek zabírá  
přibližně 750 B, databáze musí  
projít téměř 267 GB dat.

načtená a použitá data

načtená ale nepoužitá data

Column-store	MANDT	VRGAR	USNAM	HZDAT	WADAT	...
1.řádek						
2.řádek						
3.řádek						
....						
381,839,812.						

Pokud atribut USNAM zabírá 8  
byte (NVARCHAR), pak musíme  
projít přibližně 2.85 GB dat.

načtená a použitá data

data která nebyla načtena

**Obr. 12: Rozdíl mezi row-store a column-store tabulkou (Zdroj: vlastní zpracování)**

Potřebný čas pro zpracování takového množství dat, kde pro jednoduchost uvažujeme CPU s rychlostí čtení dat z hlavní paměti 4MB/ms na jádro. Výpočet by tedy vypadal takto:

$$t = \frac{273\,113\,MB}{4MB/ms} \approx 68\,sekund$$

Nyní při stejných podmínkách, ale pro column-based uspořádání:

$$t = \frac{2\,913MB}{4MB/ms} \approx 0.7\,sekundy$$

Pro zrychlení můžeme rozdělit tabulky na menší části a použít paralelní procházení za využití vícejádrových procesorů. Dnes se běžně v serverech používá 4 až 8 procesorů, majících 8 jader jeden každý, což při správném rozložení, podporou „vícevláknových“ procesů DBMS a operačního systému zkrátí tyto výsledky o další řád. Mezi parametry ovlivňující rychlost zpracování dat uvnitř databázových systémů patří:

- Využití cache paměti procesoru
- Propustnost paměti
- Počet procesorů a jader
- Množství záznamů tabulky
- Použitá komprese
- Rozvržení tabulky [14]

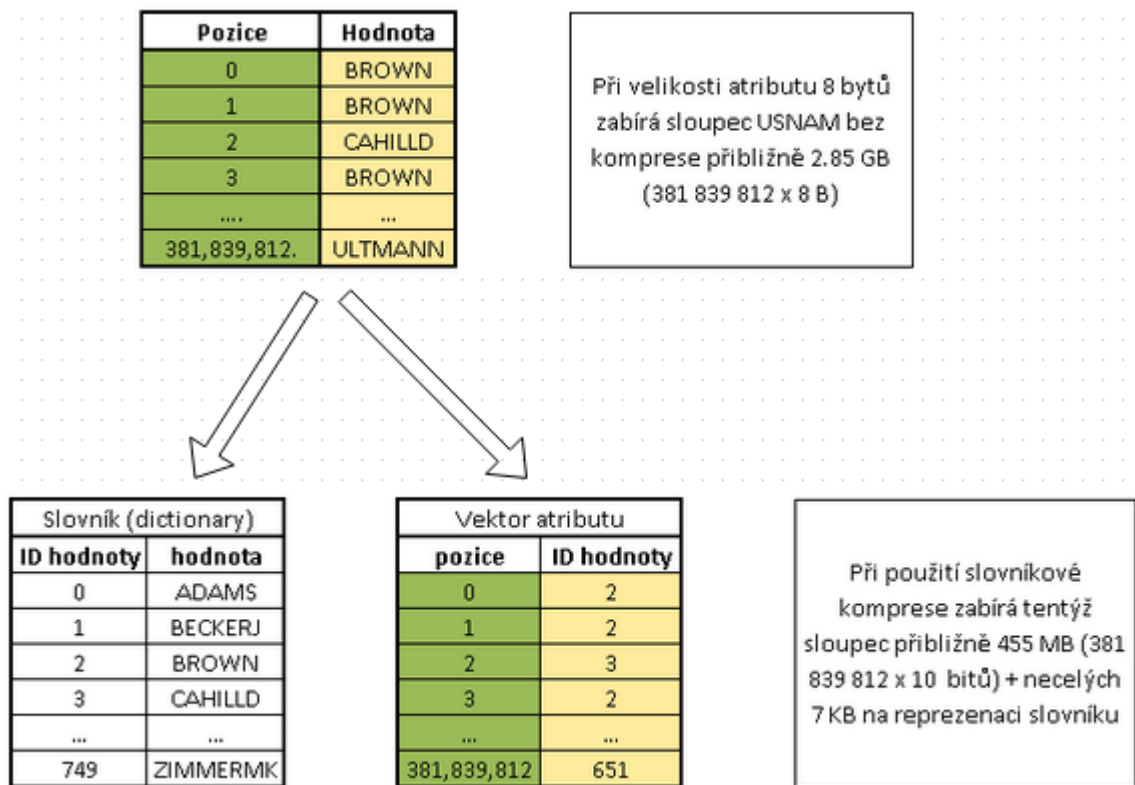
## 2.5 Komprese dat

Byla zde zmíněna komprese dat jako jeden z ovlivňujících faktorů pro rychlost odpovídání na SQL dotazy. Kromě tohoto zrychlení je také druhou spjatou vlastností, jak již z názvu vyplývá, redukce potřebného místa pro uložení záznamů, čímž šetříme prostor, jenž je nutný vymezit v paměti. SAP HANA používá dvě úrovně komprese tabulek.

První z nich je takzvaná slovníková komprese (dictionary-based compression). Ta je nastavena jako základní pro všechny sloupcově uložené tabulky. Používá se pro každý sloupec tabulky zvlášť. Vychází z prostého faktu, že většina atributů v podnikových tabulkách není příliš různorodých. Slovník je tedy tvořen všemi hodnotami vyskytujícími se pro daný atribut uvnitř tabulky a každé hodnotě je pak přiřazena číselná reprezentace uložená v binárním kódu. Databáze tedy pracuje s čísly uvnitř vektorů atributů namísto textových řetězců, šetří tedy místo a zvyšuje rychlost zpracování dat.

Tabulka CE1ST00 a její atribut USNAM má 750 unikátních hodnot. Pro číselnou reprezentaci všech těchto hodnot je zapotřebí deseti bitů. Na každém řádku tedy lze s použitím slovníkové komprese ušetřit téměř sedm bytů. Proto při výpočtu potřebné velikosti paměti se doporučuje počítat s faktorem komprese 4 až 8 v závislosti na zdrojovém systému.

Druhý typ uvnitř SAP HANA systému se nazývá pokročilá komprese. Zde existuje mnoho metod v závislosti na struktuře dat, typu, kardinalitě a entropii. Umožňuje další zredukování dat na úrovni bytů, či dokonce bitů. Mezi nejčastěji používané patří prefix encoding, run length encoding, cluster encoding, sparse encoding a indirect encoding [15]. Tyto techniky jsou aplikované na vektory atributů, kde již byla zavedena slovníková komprese. Dále existuje i možnost zmenšit potřebné místo pro uložení slovníku nejdelší společné části dvou po sobě jdoucích hodnot v řetězci.



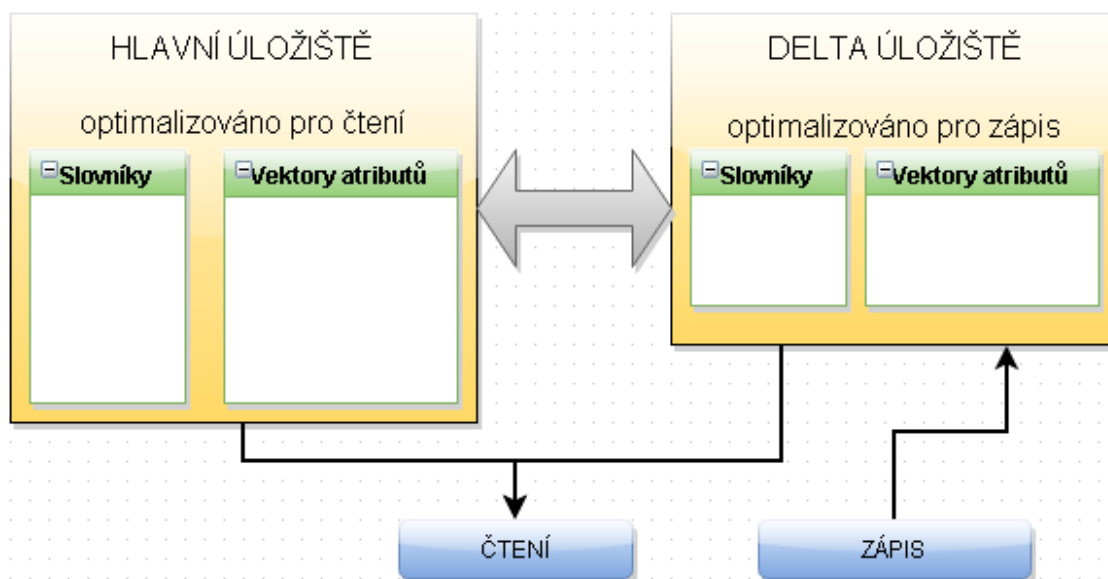
**Obr. 13: Slovníková komprese sloupce (Zdroj: vlastní zpracování)**

Největším problémem je, pokud se zapíše do některého záznamu nová hodnota, která se předtím nevyskytovala ve slovníku nebo je před poslední hodnotou. V takovém případě se celý slovník, který je seřazený podle abecedy, musí přeskládat. Tím se ovšem změní i ID reprezentující jednotlivé hodnoty a nastává situace, kdy se musí projít celý vektor atributů a změnit ID i zde.

## 2.6 Hlavní a rozdílové úložiště

SAP HANA v tomto případě chrání uživatele před degradací rychlosti vlivem neustálého přeskládávání technikou rozdělení každého sloupce a s ním spjatých slovníků a atributů na „main” a „delta storage”.

Veškeré změny (insert, delete, update) jsou zaznamenávány do delty a v hlavním je pouze označena skutečnost, že hodnota již není aktuální (je zneplatněna). Výsledkem je, že i pro čistě selektivní SQL dotaz, je nutné procházet obě části každého atributu. Vzhledem k tomu, že v rozdílovém úložišti nejsou záznamy seřazené a tudíž doba potřebná pro procházení se značně zvyšuje s rostoucím počtem záznamů, musí docházet k procesu zvanému delta merge.



**Obr. 14: Systém ukládání dat SAP HANA column-based tabulek (Zdroj: vlastní zpracování)**

Inicializace tohoto procesu může být způsobena mnoha podmínkami nebo událostmi nastalými uvnitř databáze. Například pokud počet hodnot uvnitř delty dosáhne určitého počtu, procent z celkového počtu hodnot (main + delta), konkrétní velikosti v paměti popřípadě jednou za stanovený časový úsek. Všechny tyto parametry se nacházejí v konfiguračních souborech SAP HANA databáze, stejně jako nastavení komprese a další.

## 3 VLASTNÍ NÁVRH ŘEŠENÍ

Tato kapitola se zabývá vlastním řešením projektu. Jsou zde analyzovány jednotlivé participující prvky a popsána jejich instalace se zaměřením na problematické stránky. Součástí je i krátký popis té části IT prostředí, která přímo souvisí s projektem. Pomocí schémat a tabulek je podrobněji charakterizován cílový stav.

### 3.1 Zadání projektu

Firma požadovala zvýšení rychlosti práce s daty týkající se financí. Jelikož používá systém SAP ECC a její nainstalované součásti pro analýzu ziskovosti, bavíme se zde právě o ní. Konkrétně se jedná o costing based profitability analysis neboli CO-PA. Jako jedno z možných řešení bylo přenést data speciálně pro modul analýzy ziskovosti na rychlejší platformu, než je DBMS DB2 od společnosti IBM.

Protože se zákazník snaží držet krok s dobou a používat nejmodernější technologie, což je vlastně i součástí strategie, rozhodla se poohlédnout po takzvané in-memory databázi, zkráceně IMDB. Na trhu je samozřejmě větší množství hráčů, ačkoli ze začátku to byly pouze systémy několika největších firem zabývajících se prodejem databází. Mezi ty nejznámější, které obsahují nějakou podobu in-memory technologie patří:

- IBM DB2 BLU Acceleration
- Oracle Database 12c
- SAP HANA
- Microsoft SQL server Hekaton

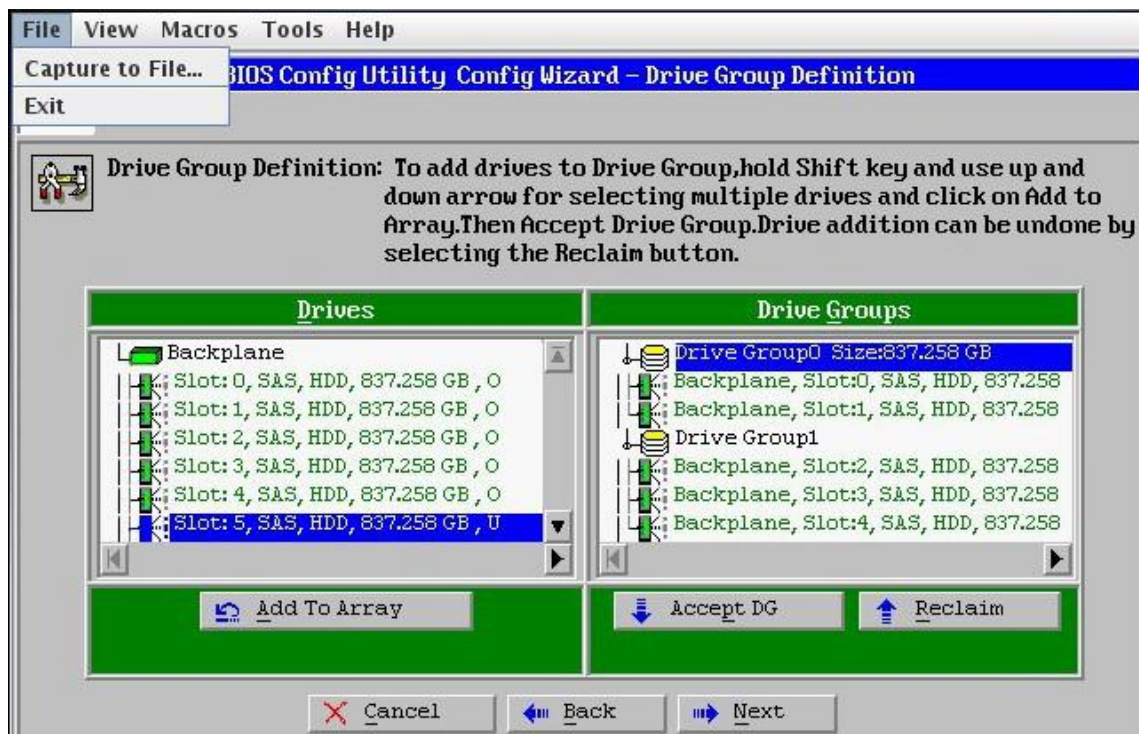
### 3.2 Databázová platforma pro CO-PA

Vybrána byla nakonec SAP HANA a to z několika důvodů. Tento produkt je z dílny dodavatele informačních systémů, používaných uvnitř firmy. To přináší úzkou vázanost funkcí a základního vztahu aplikace a databáze. Odzkoušené procedury pro akceleraci finanční analýzy, existující manuály a další informace. Dále je tu silná podpora této IMDB ve smyslu poruch, vývoje a řešení softwarových chyb. V neposlední řadě pak možnost vyjednání lepší ceny za licence, s přihlédnutím k tomu, že již několik produktů běžících ve firmě je od stejné společnosti. Nehledě na to, že pokud nechtějí přecházet kompletně s celým informačním systémem ke konkurenci, musí vzít v úvahu strategii

SAP, která vidí tuto databázi jako náhradu za ostatní dosud mezi zákazníky používané. Proto je možné očekávat ukončení podpory všech jiných DBMS ze strany společnosti SAP.

### 3.2.1 Instalace

Pro instalaci dodává IBM vlastní DVD. Obsahuje SAP HANA software, operační systém s předinstalovanými knihovnami a balíky pro databázi a GPFS (jedná se o druh file systému pro unixové OS). Využil jsem možnost vzdálené instalace za použití virtuální mechaniky, kam lze po síti „uploadovat“ celé DVD s instalací. Nutná je návštěva UEFI (náhrada za BIOS) a korektní nastavení boot zařízení a jejich pořadí. Předem jsem vytvořil RAID pole typu 1 (mirroring) pro operační systém a typu 5 pro ostatní data na GPFS.



Obr. 15: WebBIOS utility proces vytváření diskových polí (Zdroj: vlastní zpracování)

V tomto případě, kdy se nejednalo o scale-out řešení, byla instalace poměrně jednoduchá, protože na DVD se nachází soubor, který přednastaví všechny důležité parametry automaticky. Mezi ty, které je nutné zadávat přímo, patří hostname, IP adresy a výchozí brány pro GPFS, HANA a *front-end* připojení. Jako SID (system ID) jsem

zvolil HND, HNQ a HNP. Vytvoření a přiřazení oddílů probíhá automaticky, takže je stačí v jednom kroku zkontrolovat a potvrdit.

### 3.3 Hardwarová platforma pro SAP HANA

Vzhledem k tomu, že firma nepoužívá žádné *cloudové* aplikace ani hardware, i pro akceleraci CO-PA a jako podklad pro databázi i ostatní spjaté systémy bylo vybráno *on-premise* řešení. SAP HANA má několik partnerů, kteří dodávají na trh servery nebo rovnou celé *appliance* certifikované pro správný a rychlý chod této DBMS.

Prodejce	Scale-out	Single node	Max. paměť (single node)	CPU architektura (single node)	Filesystem (single node)	Operační systém (single node)
Bull SAS	ne	ano	2 TB	Ivy Bridge E7	XFS	RHEL 6.5 for SAP HANA
Cisco Systems, Inc.	ano	ano	2 TB	Haswell Ivy Bridge E7	XFS	RHEL 6.5 for SAP HANA SLES for SAP 11 SP3
Dell	ano	ano	1,5 TB	Haswell Ivy Bridge E7	logy: ext3 data: XFS	RHEL 6.5 for SAP HANA RHEL 6.6 for SAP HANA SLES for SAP 11 SP3
Fujitsu Technology Solutions	ano	ano	3 TB	Haswell Ivy Bridge E7	XFS	RHEL 6.5 for SAP HANA SLES for SAP 11 SP3
Hewlett-Packard Company	ano	ano	2 TB	Haswell Ivy Bridge E7	XFS	RHEL 6.5 for SAP HANA RHEL 6.6 for SAP HANA SLES for SAP 11 SP3
Hitachi Data Systems	ano	ano	3 TB	Haswell Ivy Bridge E7	XFS	RHEL 6.5 for SAP HANA SLES for SAP 11 SP3
Huawei Technologies Co., Ltd	ano	ano	2 TB	Haswell Ivy Bridge E7	XFS	SLES for SAP 11 SP3
Lenovo	ano	ano	2 TB	Haswell Ivy Bridge E7	GPFS	RHEL 6.5 for SAP HANA RHEL 6.6 for SAP HANA SLES for SAP 11 SP3
NEC Corporation	ne	ano	1 TB	Ivy Bridge E7	XFS	RHEL 6.5 for SAP HANA SLES for SAP 11 SP3
Silicon Graphics International Corp.	ne	ano	2 TB	Ivy Bridge E7	XFS	SLES for SAP 11 SP3
Unisys Corporation	ne	ano	1 TB	Ivy Bridge E7	XFS	SLES for SAP 11 SP3
VCE	ano	ne	-----	-----	-----	-----

**Tabulka 3: Prodejci SAP HANA certifikovaného hardwaru (Zdroj: 16)**

Výběr hardwaru nakonec padl na prodejce IBM (dnes Lenovo). Hlavním důvodem přitom byl fakt, že podnik využívá *outsourcingových* služeb IBM pro podporu ostatních aplikací, bylo proto jednodušší získat i podporu pro SAP HANA *appliance* od IBM.

Zakoupeny byly dva servery z rodiny IBM System X. Jedná se o x3950 se čtyřmi sokety pro procesory Intel Xeon. Každý procesor má k sobě dva řadiče pro paměťové moduly, které obsahují celkově čtyři kanály (2+2), kam lze zasadit dohromady maximálně osm modulů (dva moduly na kanál). Platí zde, že pokud nemá některý procesor osazen kompletně oba řadiče nebo pokud je některý kanál prázdný či obsazený jen z poloviny, dochází k degradaci rychlosti, která může v konečném důsledku klesnout až na 29% maximální propustnosti. Proto jsou oba servery a všechny sloty pro DIMM obsazené [17].

Celková konfigurace vypadá takto:

- Procesor: 4 krát 10 jádrový Intel Xeon E7 8870 taktováno na 2,4 GHz
- Paměť: 64 krát 8 GB DIMM 1067 MHz
- Raid adaptér: ServeRAID 5015
- Úložiště: 8 krát 900 GB SAS + IBM 1,2 TB High IOPS MLC Mono adapter
- Network: 2 krát 10 Gbit 2-portová Emulex + 4-portová 1 Gbit Intel + 2-portová integrovaná 1 Gbit karta NetXtreme

Celková paměť u obou serverů je 512 GB. ERP modul analýzy ziskovosti se skládá ze tří tabulek, které zabírají dohromady ve zdrojovém systému necelých 120 GB. Po použití slovníkové komprese se nakonec velikost snížila na 36 GB. Kapacita serveru tedy poskytuje značnou možnost růstu, případně nainstalování více databází pro další systémy.

### 3.3.1 Datová úložiště

Pro operační systém je vytvořen RAID1 ze dvou disků aby byla zachována bezpečnost dat. V případě nutnosti vrácení důležitých změn je možné zapisovat změny jen na jeden disk při rozpojeném RAID poli a naboťovat pak zpětně z původního systému. Zároveň chrání před poruchou jednoho z nich. Ze zbylých šesti disků je poskládán RAID5, který přináší rozumný kompromis mezi rychlostí a bezpečností dat. Někdy bývá pro vyšší bezpečnost jeden z disků přiřazen do RAIDu jako takzvaný *hot-spare*, který v případě poruchy jednoho z disků zaskočí na jeho místo a celý RAID proto nespadne do degradovaného stavu. Zákazník se ale rozhodl kvůli zachování kapacity 4 TB nepoužívat *hot-spare* disk.

Kromě klasických pevných disků systémy X5 obsahují Fusion IO kartu IBM High IOPS MLC. Funguje na principu NAND flash paměti, která je několikanásobně rychlejší než magnetické disky a stejně jako například SSD neobsahuje žádné pohyblivé části. K serverové desce je připojena pomocí PCI express a její kapacita je 1200 GB. Vzhledem k vysoké ceně za GB slouží pouze jako úložiště pro databázové redo logy, kde je rychlost zápisu důležitá.



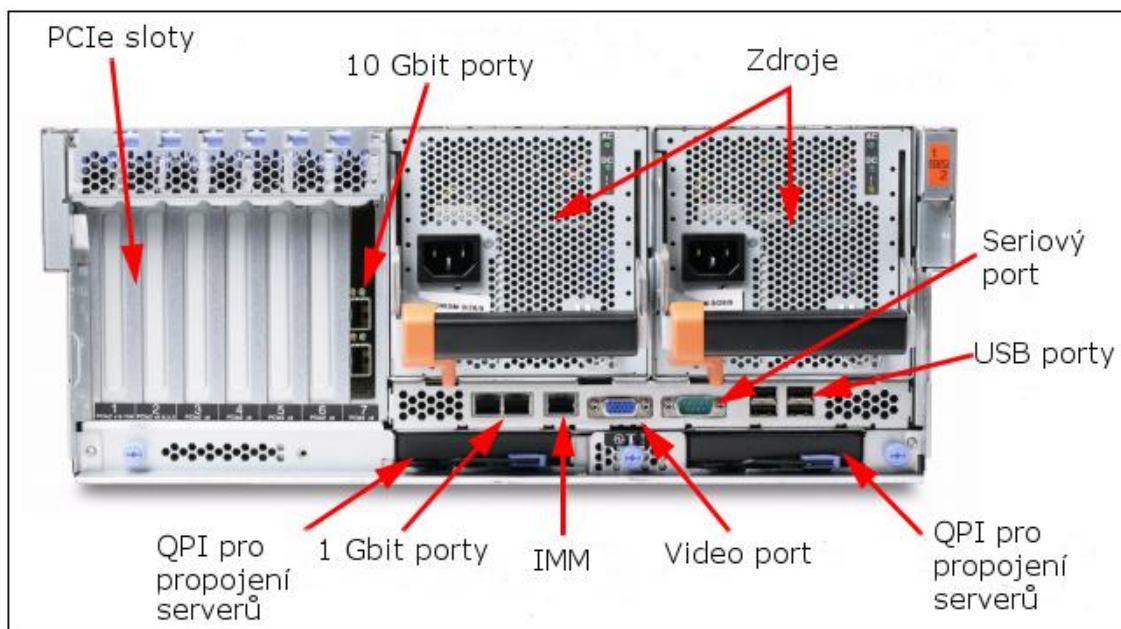
Obr. 16: IBM High IOPS MLC Mono adapter (Zdroj: 17)

### 3.3.2 Síťová rozhraní

Dvě 10 Gbitové karty jsou k dispozici pro budoucí rozšíření na *scale-out* systém. Jedna síť je pak použita pro GPFS a druhá pro samotnou SAP HANA databázi. Pro zachování redundance se využívá „bondovaných“ síťových rozhraní, které jsou tvořeny dvěma porty, tak aby každý byl z jiné fyzické karty. Na straně druhé se připojují ke dvěma switchům, znovu z důvodu vyhnutí se problému SPOF (single point of failure). Ostatní spojení jako například *front-end*, jsou na 1 Gbit kartách. Slotů pro rozšíření je sedm, není tedy problém dokoupit další síťové karty pro potřeby konkrétních aplikací. Na zadní straně najdeme také speciální port pro připojení IMM konzole.

Konzole IMM je nedílnou součástí všech systém X serverů. Jedná se v podstatě o malé zařízení uvnitř serveru, na kterém běží jednoduchý systém přístupný přes webový prohlížeč, který monitoruje zdraví serveru a umožňuje základní vzdálený management. V případě, že je server nedostupný přes běžné připojení pomocí ssh nebo telnet protokolů, nabízí možnost spustit javovou konzoli přímo z prohlížeče a zjistit, co je příčinou problému, případně restartovat celý server.

Kromě těchto portů jsou na zadní straně ještě 4 USB, sériový a video port. Dvě QPI rozhraní jsou přítomny z důvodu možnosti spojit k sobě dva servery a vytvořit tak jednu logickou jednotku s dvojnásobným hardwarovým výkonem.



Obr. 17: Zadní strana x3950, síťové karty se vkládají do PCIe slotů (Upraveno dle: 18, str. 47)

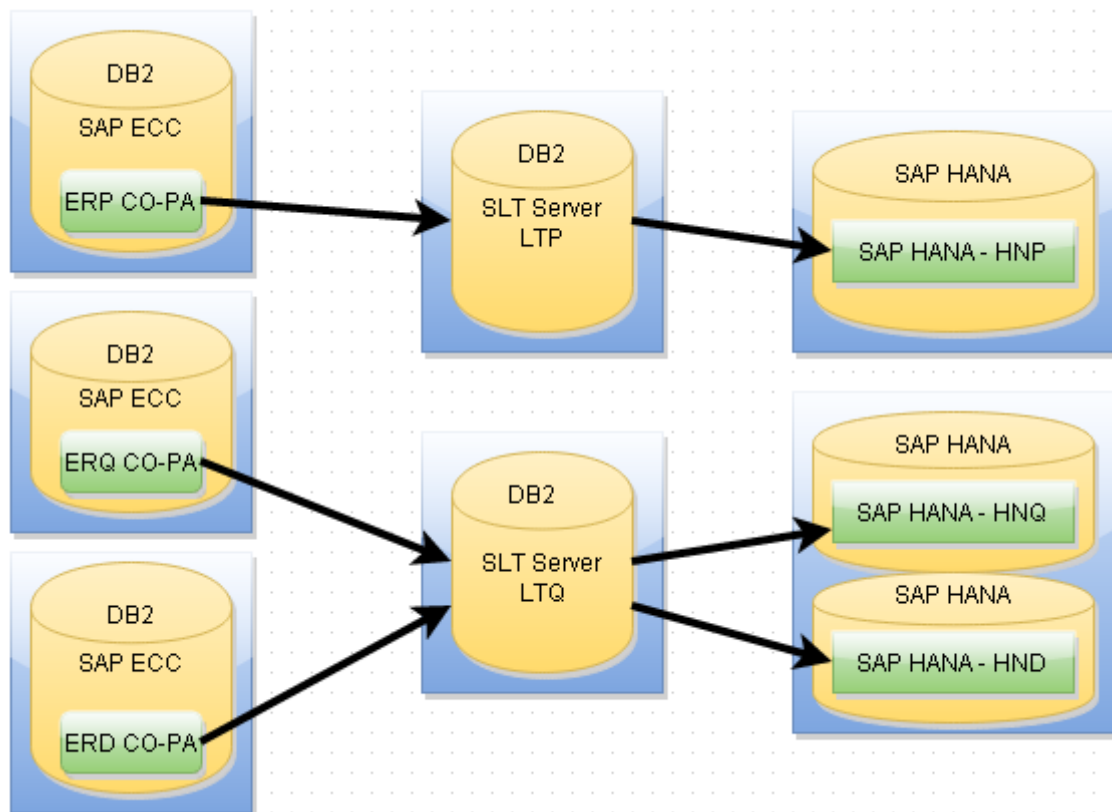
### 3.4 Schéma propojení

Většina společností používajících SAP pro své podnikové informační systémy má tři nebo i více verzí též aplikace. Tomuto přístupu se říká systém *landscape*. Nejčastější je trojkombinace *development*, *quality* a *production*. Stejně jako to je v tomto případě. Někdy se můžeme setkat ještě s *test* nebo *preproduction* systémem.

V praxi to poté funguje tak, že se vezme produkční systém, vytvoří se záloha a ta se importuje do ostatních verzí. Celá procedura, která se nazývá *system refresh*, se opakuje několikrát do roka. Na *development* aplikaci pracují vývojáři a veškeré změny provádí prve zde. Později, když jsou s výsledkem spokojeni, se vytvoří transport všech změn, který může mít určitá předem stanovená pravidla jako pořadí importu nebo různé kontrolní mechanismy. Vše se importuje do *quality* systému a zde se po určitou dobu sleduje chování aplikace před finálním transportem do produkce.

Pro akceleraci analýzy ziskovosti se na jednom serveru použilo MCOS (multiple components on one system) řešení, neboli dvě nainstalované instance pro *quality* a

*development* na tomtéž operačním systému. Na druhém serveru byla vytvořena samostatná instance pro produkci. Také SAP LT replikační servery byly vytvořeny dva ze stejného důvodu.



Obr. 18: Schéma propojení jednotlivých součástí projektu (Zdroj: vlastní zpracování)

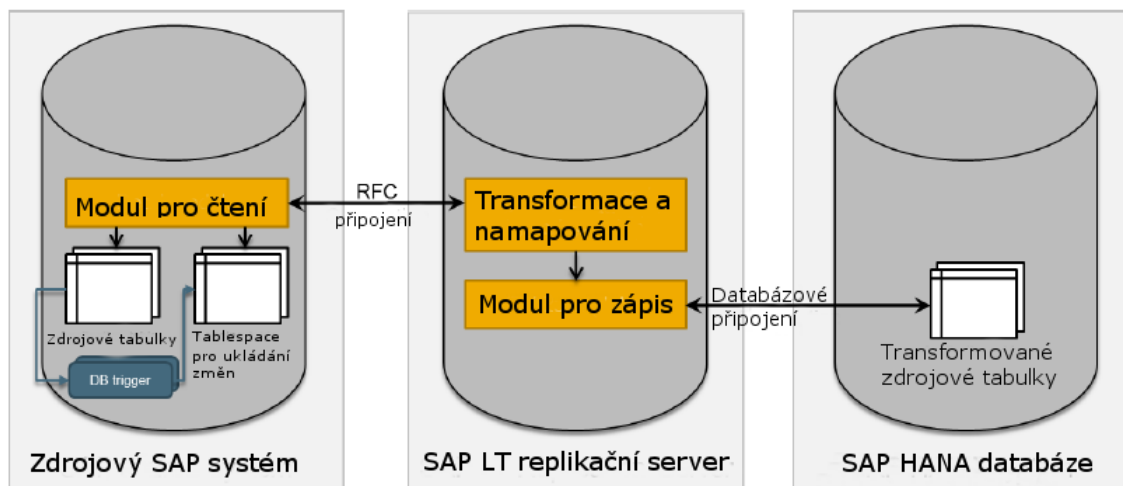
### 3.5 SAP LT replikace

SAP Landscape Transformation Replication server není nic jiného než doplněk (add-on) pro standardní SAP NW. Jeho hlavní účel je replikovat data z jedné či více tabulek zdrojového systému (nemusí to být SAP) do cílové databáze a to buďto *real-time* a nebo v určitých časových intervalech.

V případě tohoto zákazníka je zdrojový systém SAP ECC. Replikační server, jak bylo rozhodnuto, byl nainstalován na LPAR s operačním systémem AIX. Druhou možností je mít SLT přímo ve zdrojovém systému, zákazník ovšem nechtěl přetěžovat své servery, které hostí ERP aplikace.

Pokud je zdrojovým systémem produkt společnosti SAP, jako je tomu v tomto případě, spojení mezi SLT a zdrojem je pomocí RFC připojení. Mezi HANA databází a SLT se

používá běžné databázové spojení pomocí *jdbc/odbc* klientů. Do aplikace SAP ECC se doinstaluje doplněk DMIS, který přidá *table space* pro ukládání změněných položek tabulek, jež se budou replikovat. Změny jsou kontrolovány databázovým „triggerem“ a následně zapisovány do *table space* určeného pro ukládání změn. Odtud pak pomocí RFC přechází přes SLT systém, kde dochází k optimalizaci pro SAP HANA a následnému přenesení delty do této in-memory databáze.



Obr. 19: Schéma SAP LT replikace se samostatným SLT serverem (Zdroj: 19)

### 3.5.1 Dimenzování (sizing)

Aplikace SLT nemá vysoké nároky na výpočetní výkon. Faktory ovlivňující konfiguraci jsou: počet tabulek, velikost jednoho záznamu v bytech, počet sloupců a průměrné množství změněných záznamů za hodinu. Dále bylo potřeba vzít v úvahu potřeby ABAP NW aplikace, s jednou centrální instancí a několika dialogovými procesy běžící na DB2 databázi. Verze NetWeaver 7.0 EHP2 (enhancement package), AIX operačního systému 7.1 i databáze 10.5 byly vybrány tak, aby korespondovaly s již existujícím prostředím firmy. Parametry výsledného LPAR vytvořeného pro SLT server jsou vidět v Tabulka 4.

Firma má již existující profily pro vytváření virtuálních serverů. Navrhované specifikace se tedy porovnaly s jednotlivými profily a výsledkem je pak mírně navýšené množství RAM a virtuálních procesorů, oproti mnou navrhovaným 8 CPU a 32 GB RAM, jelikož si zákazník přál mít raději více výpočetního výkonu, přesto že je možné v budoucnu kdykoli jednoduše přidat další virtuální RAM nebo procesory pomocí HMC

konzole. Takovéto navyšování se ovšem neobejde bez následného rebootu. Filesystemy jsem vytvořil podle svého návrhu, ale není problém později některé z nich rozšířit, nebo přidat další disky, pokud by byl někde nedostatek místa.

	Finální konfigurace	Poznámka
CPU	12 desired virtual CPU	entitled capacity 0,3
Paměť	72 GB	ABAP NetWeaver + db2
rootvg	2x69 GB RAID1	klasické rozvržení pro AIX systém
sapstlvg	6x69 GB	přidělovány jsou z VIO serveru
SWAP	20 GB	unix SWAP space
/usr/sap	2 GB	základní mount point pro NetWeaver
/usr/sap/DAA	6 GB	SAP SMD agent
/usr/sap/<SID>	15 GB	soubory pro konkrétní instanci
/sapmnt/<SID>	12 GB	binární soubory NetWeaver
/usr/sap/trans	2 GB	sdílený filesystem pro transporty
/db2/<SID>	1 GB	základní mount point pro databázi
/db2/<SID>/db2dump	5 GB	trace soubory, logy, dumpy
/db2/<SID>/db2<sid>	4 GB	home databázového uživatele
/db2/<SID>/db2sw	6 GB	binární soubory DB2 databáze
/db2/<SID>/log_dir	20 GB	redo logy databáze
/db2/<SID>/sapdata1-4	80 GB	umístění pro tablespaces

**Tabulka 4: Konfigurace LPAR pro SLT systém (Zdroj: vlastní zpracování)**

### 3.5.2 Instalace SAP NetWeaver

Před samotnou instalací bylo potřeba stáhnout všechny potřebné balíky. Jednak jádro pro instalační program SWPM (Software Provisioning Manager) a SL (software logistics) Controller, dále NetWeaver, základní balíček jazyků pro NW, DB2 databázi, databázového klienta a konečně samotný DMIS add-on, což je právě SLT systém. V neposlední řadě jsem musel zvlášť zajistit fix pack 4 pro DB2, abych mohl aktualizovat na cílovou verzi.

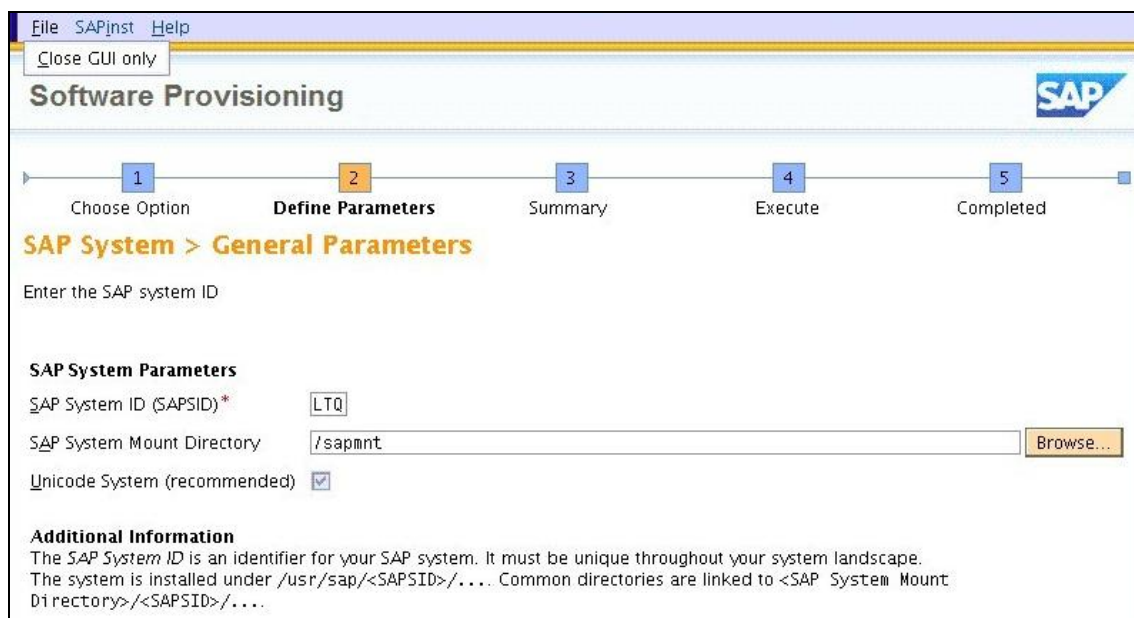
Mimo zkompletování softwaru bylo třeba projít veškeré doporučení spojené s instalací dané verze NW pro konkrétní databázi. Bylo třeba zkontrolovat kompatibilitu DMIS doplňku s NW 7.0 a použitým jádrem. Jednou z požadovaných úkolů byl update několika AIX knihoven a nastavení systémových parametrů.

Po rozbalení instalačních souborů pomocí programu SAPCAR, který je obdobou zipu či rarů jsem se připojil pomocí ssh na server s nastavenou možností *X forwarding*, umožňující spouštět grafické rozhraní programů. Pro kontrolu splnění všech potřebných

nastavení existuje uvnitř instalačního programu možnost „Prerequisites Check“. Výstupem je tabulka udávající status a slovní popis jednotlivých parametrů, jejich současnou a očekávanou hodnotu.

Po odstranění všech reportovaných odchylek od správného nastavení, jsem spustil samotnou instalaci NW centrální instance. Ta se skládá celkově z více než dvaceti kroků. Nejprve se zadávají cesty k instalačním balíkům, které jsem popsal. Poté lze konfigurovat četné možnosti personalizace.

V oblasti databáze jsem ponechal vypnutou kompresi dat, vzhledem k tomu, že jde o vnitřní tabulky a naopak jsem využil deferred tabulek. Prázdné tabulky tak nebudou na začátku vytvořeny, až teprve v případě potřeby do nich zapisovat. Jelikož bude systém využíván pouze pro účely replikace, ušetří se jinak zbytečně zabrané místo. V dalších krocích jsem zadal jméno pro instanci a schéma (LTQ) a heslo pro databázového administrátora. Během instalace se vytvoří několik skupin pro monitoring, maintenance, kontrolu a administrátora na úrovni operačního systému. Paměť určenou pro databázi se doporučuje mít na jedné třetině fyzické, proto jsem vybral hodnotu 24 GB. Čísla portů a přiřazení adresářů jsem pak již nechal na přednastavených hodnotách.



Obr. 20: Prostředí Software Provisioning Manager (Zdroj: vlastní zpracování)

Pro dokončení byly vloženy údaje domény zákazníka, čísla ABAP instancí a hostnami serverů. Nebudu na tomto místě popisovat SMD (solution manager diagnostics) agenta, jeho nastavení a zprovoznění, ačkoli byl součástí projektu, není z hlediska funkčnosti replikace potřebný. Jeho hlavní funkce je monitoring a reporting stavu aplikace. Teprve v posledním kroku začne SWPM kopírovat soubory a vytvářet potřebné hierarchie a tabulky. Během této části jsem narazil na několik problémů, ačkoli předinstalační kontrola proběhla v pořádku.

Jednalo se například o problém nekorektních práv nastavených pro balíky s databázovým exportem, který se dal lehce vyřešit příkazem *chmod* v OS AIX a povolením čtení tohoto souboru databázovým uživatelem. Dalším zádrhelem byla stížnost instalátoru na neprázdný cílový adresář pro binární soubory DB2. V tomto případě byla příčina operačním systémem vytvořený podadresář *lost+found*. Toto místo slouží jako skladiště pro ztracené fragmenty souborů a OS AIX jej vytváří automaticky během procesu kontroly filesystemu.

```
DB2 Setup log file finished at: Wed Dec 10 11:43:49 MEZ 2014
=====
The value provided for the installation path is not valid. The location
specified "/db2/LTQ/db2sw" exists and is not empty. Specify a new or empty
location.

The response file specified
"/tmp/sapinst_instdir/NW702/AS-ABAP/DB6/CENTRAL/unix_ese.rsp" is not valid.

For more information see the DB2 installation log at
"/tmp/sapinst_instdir/NW702/AS-ABAP/DB6/CENTRAL/db2setup.log".
~
```

Obr. 21: log SWPM: problém s cílovým filesystemem (Zdroj: vlastní zpracování)

### 3.5.3 Updaty softwarových součástí

Po úspěšném provedení všech kroků je na operačním systému připraven čistý ABAP NetWeaver běžící na DB2 databázi. Tím ovšem práce s ABAP instancí zdaleka nekončí. Během přípravné fáze jsem zkontroloval závislosti všech sledovaných částí softwaru. Všechno se odvíjelo od faktu, že si zákazník přál mít shodné verze SAP u svých ERP systémy a SLT serverů. Výsledek lze vidět v Tabulka 5, kde je tučně označeno, které komponenty je třeba aktualizovat.

Komponenta	Zdrojový systém		SAP LT server	
	původní verze	potřebná verze	verze po instalaci	potřebná verze
NetWeaver	7.02	4.6C a vyšší	7.02	7.02 a vyšší
ABAP kernel	721 patch 326	720 a vyšší	721 patch 100	<b>721 patch 326</b>
SAP_BASIS	702 SP11	<b>702 SP12</b>	702 SP2	<b>702 SP12</b>
DB2 level	10.5.0.4	10.5.0.0	10.5.0.0	<b>10.5.0.4</b>

**Tabulka 5: Rozbor verzí komponent (Zdroj: vlastní zpracování)**

*Kernel* update, za použití stažených souborů, se sestává z několika dílčích kroků. Zde je na ukázkou vložena sekvence příkazů, ve které je nejprve zastaven NW, poté na řádku 2 je vyčištěna paměť centrální a ASCS instance s následným odstraněním momentálně nepotřebných modulů z jádra OS a paměti knihoven. Na řádcích 3-5 je vytvořena záloha původního jádra NW, pro případ kdyby bylo nutné se vrátit zpět a opakovat postup, a rozbalení souborů pro update, kterými se přepíše stávající jádro. Script „saprootsh“, jenž je součástí instalace, slouží k vytvoření linků k novým binárním souborům a vykonání dalších nutných akcí spojených s aktualizací.

- 1: su – ltqadm; stopsap r3 <hostname>;
- 2: cleanipc 00 remove;cleanipc 01 remove;slibclean
- 3: cd /sapmnt/LTQ; mkdir exe\_old; cp -R exe exe\_old;
- 4: ../exe\_old/SAPCAR -xvf /tmp/stage/kernel\_upgrade/SAPEXE\_326-10011318.SAR
- 5: ../exe\_old/SAPCAR -xvf /tmp/stage/kernel\_upgrade/SAPEXEDB\_326-10011317.SAR
- 6: su – root;cd /sapmnt/LTQ/exe;./saprootsh LTQ
- 7: su - ltqadm; startsap r3 <hostname>

Podobným způsobem jsem pak povýšil *fix pack* DB2. Zde se pro zachování předchozí verze používá speciální způsob instalace do nového, předem vytvořeného prázdného adresáře pomocí parametru instalačního skriptu (řádek 8) a poté se dalším skriptem (řádek 9) donutí DB2 použít při příštím startu binární soubory již s novým *fix pack*.

- 8: /tmp/stage/AIX\_64/ESE/disk1/installFixPack -b /db2/LTQ/db2sw -p  
/db2/LTQ/db2sw/v10.5\_fp4
- 9: /db2/LTQ/db2sw/v10.5\_fp4/instance/db2iupdt db2ltq

Poslední komponenta potřebná pro korektní fungování celého projektu, u které bylo nutné povýšit verzi je SAP\_BASIS, a to jak pro zdrojový systém, tak pro SLT server.

Jedná se základní balík obsahující stěžejní funkce a transakce. Pro aktualizaci *support package* se užívá transakce SPAM uvnitř NW. Nejprve se zdrojový soubor nahraje do adresáře pro import, v tomto případě */usr/sap/trans*. Po rozbalení se vytvoří dva soubory v podadresáři *eps/in*. Ty jsou rozpoznány a načteny do fronty. V transakci SPAM se aplikují změny a výsledek lze sledovat v logu, kde se nacházejí případná varování a chyby.

Log Overview for SAPKB70212			
Display restricted to target system ERD			
Log Overview for SAPKB70212			
SAPKB70212 Basis Support Package 12 for 7.02			
ERD System ERD			
Generate Transport Information File	18.12.2014	10:24:25	(0) Successfully Completed
Import Request Piece List	18.12.2014	10:24:51	(0) Successfully Completed
Test Import	18.12.2014	10:30:24	(4) Ended with Warning
Import ABAP Dictionary Objects	18.12.2014	10:37:59	(4) Ended with Warning
ABAP Dictionary Activation	18.12.2014	10:39:18	(4) Ended with Warning
Import	18.12.2014	11:08:52	(4) Ended with Warning
Method Execution	18.12.2014	11:17:37	(0) Successfully Completed
Import steps not specific to transport request			

Obr. 22: Logy transportu obsahující SAP\_BASIS SP update (Zdroj: vlastní zpracování)

### 3.5.4 Instalace DMIS

Tento doplněk v sobě zahrnuje SAP LT Replication server. V současnosti se stále používá verze 2011, která se neustále vyvíjí, ale její základní funkčnost zůstala nezměněna. Vybral jsem v nejnovější *service pack 7*, který byl v době realizace projektu k dispozici. Add-on bylo zapotřebí nainstalovat do zdrojového NW i do NW určeného pro samostatný SLT server.

Použil jsem standardní způsob přidávání nových prvků přes transakci SAINT. Proces je téměř shodný jako v případě *service pack* aktualizace. I zde jsem rozbalil vše do transportního adresáře a načtl frontu ze systému. Kliknutí na tlačítko pokračovat se spustil proces přípravy a instalace DMIS doplnku.

### 3.6 Propojení systémů

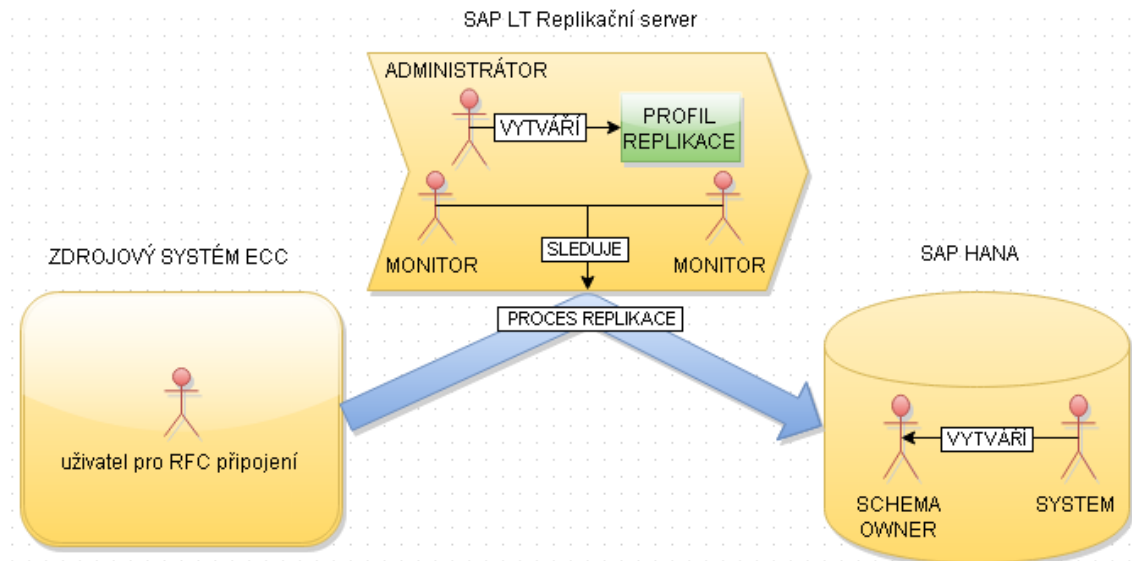
Ve finální fázi proběhly akce pro dokončení celého projektu, které zahrnují přípravu prostředí, veškerého připojení a tvorbu profilu. První naplnění daty ze zdroje a odladění detailů komunikace.

### 3.6.1 Aplikační uživatelé

Pro rozhýbání celého systému replikace bylo nutné vytvořit uživatele se správnými právy. Ve zdrojovém systému se nejprve vygeneruje role *SAP\_IUUC\_REPL\_REMOTE* v transakci *PFCG* a následně se vytvoří uživatel, kterému se role přiřadí. Jména jsem vybral *SAPLTQ* a *SAPLTP*, aby šlo snadno poznat podle názvu, za jakým účelem byly účty vytvořeny.

Na SAP LT serveru jsou v podstatě dva druhy autorizací. Ta první se přiřazuje administrátorům, kteří mohou vytvářet a editovat profily replikací, druhá se používá pro monitorování. Aktivoval jsem obě verze a vytvořil dva uživatele LTQADM (potažmo LTPADM) pro administrační účely a LTQMON (potažmo LTPMON) pro sledování procesu replikace.

Uvnitř SAP HANA není potřeba nic aktivně tvořit. Pro prvotní spojení lze užít nativního uživatele SYSTEM, který pak již při sestavování propojení vygeneruje a nastaví vše potřebné.



Obr. 23: Koncept uživatelských rolí (Zdroj: vlastní zpracování)

### 3.6.2 RFC a databázové připojení

Prvním druhem spojení, které jsem definoval v SLT systému, bylo RFC připojení. Jde o prosté zadání několika údajů v transakci SM59 jako název, IP adresa nebo host, druh

připojení a uživatel, vytvořený v předchozím kroku společně s číslem zákaznického klienta.

Databázové spojení do HANA serveru probíhá přes *hdbclient*, jež musí být doinstalován do SLT (kvůli replikaci) i do ERP (kvůli přístupu koncových uživatelů ke CO-PA tabulkám). Zde přichází na řadu transakce DBCO, kde vytvoření připojení funguje na stejném principu jako u RFC.

### 3.6.3 Replikace a akcelerace

Jakmile byly dokončeny všechny předchozí aktivity, mohl jsem přejít k samotnému nastavení replikace transakcí LTR, kde jsem zadal jména uživatelů a připojení, které jsem si připravil. Mezi další parametry nutné pro aktivaci replikace patří počet „work procesů“ starajících se o data transfer, kde jsem vložil hodnotu 15, což je více než dostačující pro tři tabulky analýzy ziskovosti a hardwarová konfigurace AIX LPARu je dimenzovaná dostatečně i pro vyšší počet procesů. Aby nedocházelo k přetěžování zdrojového ECC systému, nebyla zvolena real-time replikace, nýbrž v 30 minutovém intervalu. Pro zákazníka nebylo tolik důležité mít data aktuální, půl hodinové zpoždění není pro analýzy a uzávěrky kritická hodnota, protože se většinou pracuje s daty za minulé období.



Obr. 24: Status akcelerace analýzy ziskovosti (Zdroj: vlastní zpracování)

Aby koncový uživatel analýz a reportů mohl těžit z rychlosti in-memory databáze, musí se samozřejmě někde definovat správná cesta k datům. K tomu slouží transakce KEHC v ERP zdrojovém systému, kde se vybere jako hlavní spojení k tabulkám databázové spojení do SAP HANA. Kdykoli tedy přijde SQL dotaz na jednu z tabulek CE1ST00, CE2ST00 nebo CE4ST00, odehrává se zpracování na úrovni SAP HANA a výsledek je předán nejprve databázovým připojením zpět do ERP systému a poté zobrazen uživateli v aplikaci.

### 3.7 Porovnání rychlosti zpracování dat

Vzhledem k tomu, že stejná data se nacházejí v tabulkách DB2 i SAP HANA databáze, mohl jsem vytvořit několik jednoduchých SQL dotazů a porovnat rychlost odpovědi. Každý z nich jsem pak opakoval celkově třikrát, abych eliminoval možnost, že byla v té době databáze ovlivněna nějakou okamžitou událostí a zároveň aby v případě IBM DB2 byla data pokud možno nahrána do cache paměti.

V případě SAP HANA jsem použil přibalený software SAP HANA Studio, který obsahuje integrovanou SQL konzoli, umožňující zobrazení časových údajů o příkazu. Musel jsem sečíst čas pro provedení příkazu s časem nutným pro zobrazení hodnot. Bylo to z toho důvodu, že program *db2batch*, jež jsem použil v případě DB2 databáze, zobrazuje pouze celkový čas. Připomeňme si, že v době kdy jsem tento test prováděl, se již počet řádků tabulky CE1ST00 přehoupl přes 400 milionů.

SQL dotaz	IBM DB2 Průměrný čas odpovědi	SAP HANA Průměrný čas odpovědi
select distinct A.USNAME from CE1ST00 as A left join CE2ST00 as B on A.MANDT= B.MANDT where A.GYEAR='2012'	2 017 s ≈ 34 min	0,417 s
select * from CE1ST00 where USNAME='BECKERF' and POSTNR='000016' and PAOBJNR='0087686479' order by POSTNR desc	0,142 s	0,316 s
select PALEDGE, AVG(KURSFPP), SUM(KURSBK) from CE1ST00 group by PALEDGE	3 693 s ≈ 62 min	0,483 s
select count(distinct BUDATE) from CE1ST00 where USNAME like 'S%'	2 180 s ≈ 36 min	0,325 s

**Tabulka 6: SAP HANA a IBM DB2: Porovnání rychlosti vybraných SQL příkazů (Zdroj: vlastní zpracování)**

Většina dotazů byla směřována na celé sloupce. Použity byly funkce na sumu, počet různých hodnot jednoho atributu a průměr. IMDB ukázala svou rychlost tam, kde se předpokládalo. Získané údaje předčili mé očekávání, místo desítek minut jsme se u in-memory databáze pohybovali v rámci stovek milisekund. Pouze v případě dotazů na konkrétní hodnoty byla DB2 rychlejší. Důvodem je absence indexů pro sloupcové tabulky uvnitř SAP HANA. Indexy samozřejmě existují i zde, leč jejich použití nemá ve většině případů valné opodstatnění. Musím ale na tomto místě zmínit, že ačkoli jsem tento test prováděl během víkendu, mohly být výsledky ovlivněny využitím systému pro různé transformace, reporty a *loady* nastavené na automatické spuštění.

## 4 ZÁVĚR

Ve své bakalářské práci jsem se podíval nejprve na teoretické úrovni na základní koncept databázových systémů, jejich historii až po rozdělení na OLAP a OLTP. Díky tomu jsem mohl plynule navázat a zmínit potřebu dnešních firem pro co nejrychlejší zpracování OLAP dotazů nad stovkami miliony záznamů, což vedlo ke vzniku technologie in-memory databáze.

Popsal jsem základní principy, na kterých tento nový směr uložení dat stojí na příkladu SAP HANA databáze, která byla vybrána pro tento projekt. Nahlédl jsem a částečně zhodnotil ekonomickou stránku společnosti, její slabé stránky a z nich vyplývající potřebu pro zrychlení finančních analýz. Zadaný cíl projektu byla technická realizace akcelerace modulu finanční ziskovosti uvnitř jejich hlavního ERP systému od společnosti SAP. Navrhnuo a zrealizováno pak bylo řešení za pomoci SAP HANA jako databázové platformy, SAP LT serveru pro replikaci dat a hardwaru dodaného IBM.

Cíle bakalářské práce, potažmo projektu byly splněny. Všechny participující systémy byly nakonfigurovány a propojeny dle navrženého schématu. Výsledkem jsou pak uživatelem přístupná data z tabulek analýzy ziskovosti, uložená na platformě in-memory databáze, kde v případě jejího výpadku či nedostupnosti bude uživatel přesměrován k původnímu zdroji, takže data budou i tak přístupná. Porovnal jsem rychlost zpracování dat mezi zdrojovou a cílovou databází, kde jsem dokázal, že časová úspora je markantní, tudíž zákazník se může rychleji rozhodovat na základě výsledků reportů. Dalším logickým krokem pak budou optimalizace těchto reportů a objektů pro SAP HANA technologii, případně vybudování aplikace přímo nad SAP HANA databází pomocí volitelných doplňků.

## SEZNAM POUŽITÝCH ZDROJŮ

- [1] LUHAN, J. *Elektronický obchod. Přednášky*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2014/2015.
- [2] LACKO, L. *Databáze: datové sklady, OLAP a dolování dat s příklady v Microsoft SQL Serveru a Oracle: podrobná uživatelská příručka*. Brno: Computer Press, 2003. ISBN 80-722-6969-0.
- [3] SILBERSCHATZ, A., H. F. KORTH a S. SUDARSHAN. *Database system concepts*. 6th ed. New York: McGraw-Hill, 2011. ISBN 00-712-8959-3.
- [4] KOCH, M. a B. NEUWIRTH. *Datové a funkční modelování*. Vyd. 4., rozšířené. Brno: Akademické nakladatelství CERM, 2010, 142 s. Učební texty vysokých škol. ISBN 978-80-214-4125-5.
- [5] GOLFARELLI, M. a S. RIZZI. *Data warehouse design: modern principles and methodologies*. New York: McGraw-Hill, 2009, xxi, 458 s. ISBN 978-0-07-161039-1.
- [6] SODOMKA, P. a H. KLČOVÁ. *Informační systémy v podnikové praxi*. 2. aktualiz. a rozš. vyd. Brno: Computer Press, 2010. ISBN 978-80-251-2878-7.
- [7] MANAGEMENT MANIA. Zisk před zdaněním, úroky a odpisy (EBITDA). *Managementmania.cz* [online]. ©2011-2013 [cit. 2015-03-03]. Dostupné z: <https://managementmania.com/cs/zisk-pred-zdanenim-uroky-a-odpisy>
- [8] IBM. 9117-MMB (IBM Power 770). *www-01.ibm.com: IBM Knowledge center* [online]. ©2015 [cit. 2015-03-25]. Dostupné z: [http://www-01.ibm.com/support/knowledgecenter/POWER7/p7hdx/9117\\_mmb\\_landing.htm?lang=en](http://www-01.ibm.com/support/knowledgecenter/POWER7/p7hdx/9117_mmb_landing.htm?lang=en)
- [9] SAP AG. SAP Global Corporate Affairs. *Sap.com* [online]. 2015 [cit. 2015-03-08]. Dostupné z: <http://www.sap.com/corporate-en/factsheet>
- [10] PATEL, M. *SAP ERP Financials: podrobná uživatelská příručka*. Vyd. 1. Brno: Computer press, 2010, 464 s. ISBN 978-80-251-2488-8.
- [11] SAP AG. Profitability analysis (CO-PA). *Help.sap.com* [online]. 2015 [cit. 2015-03-11]. Dostupné z: [http://help.sap.com/saphelp\\_46c/helpdata/en/7a/4c37ef4a0111d1894c0000e829fbbd/content.htm](http://help.sap.com/saphelp_46c/helpdata/en/7a/4c37ef4a0111d1894c0000e829fbbd/content.htm)
- [12] SAP AG. SAP HANA Studio - Overview. *Snc.sap.com* [online]. 2013 [cit. 2015-03-29]. Dostupné z: <http://scn.sap.com/docs/DOC-60363>

- [13] BACHMAIER, M. a I. Krutov. *In-memory computing with SAP HANA on IBM eX5 and X6 systems*. 3rd ed. New York: IBM, International Technical Support Organization, 2014. ISBN 0738439908.
- [14] HASSO PLATTNER INSTITUT. *In-memory data management*. Přednáška. Postdam: Universität Postdam, 2014. Dostupné z: <https://open.hpi.de/courses/imdb2014>
- [15] SAP AG. Data compression in the column store. *Help.sap.com* [online]. 2015 [cit. 2015-05-04]. Dostupné z: [http://help.sap.com/saphelp\\_hanaplatform/helpdata/en/bd/9017c8bb571014ae79efae746940f3/frameset.htm](http://help.sap.com/saphelp_hanaplatform/helpdata/en/bd/9017c8bb571014ae79efae746940f3/frameset.htm)
- [16] SAP AG. Certified SAP HANA Hardware Directory. *Global.sap.com* [online]. ©2015 [cit. 06-05-2015]. Dostupné z: <http://global.sap.com/community/ebook/2014-09-02-hana-hardware/enEN/appliances.html>
- [17] IBM Corporation. IBM High IOPS MLC Adapters. *Redbooks.ibm.com* [online]. ©2012 [cit. 09-05-2015]. Dostupné z: <http://www.redbooks.ibm.com/redbooks.nsf/RedbookAbstracts/tips0907.html>
- [18] WATTS, D. a D. FURNISS. *IBM eX5 Portfolio Overview: IBM System x3850 X5, x3950 X5, x3690 X5, and BladeCenter HX5* [online]. 6th ed. New York: 2013 [cit. 07-05-2015]. REDP-4650-05. Dostupné z: <http://www.redbooks.ibm.com/redpapers/pdfs/redp4650.pdf>
- [19] SAP AG. Trigger-Based Data Replication Using SAP Landscape Transformation Replication Server: Installation Guide. *Help.sap.com* [online]. ©2014 [cit. 10-05-2015]. Dostupné z: [http://help.sap.com/hana/sap\\_hana\\_installation\\_guide\\_trigger\\_based\\_replication\\_slt\\_en.pdf](http://help.sap.com/hana/sap_hana_installation_guide_trigger_based_replication_slt_en.pdf)

## SEZNAM OBRÁZKŮ

Obr. 1: Od dat k moudrosti .....	13
Obr. 2: Relační databáze .....	16
Obr. 3: Příklad ER popisu databáze .....	17
Obr. 4: Třídídimenzionální krychle .....	22
Obr. 5: Slicing a dicing .....	23
Obr. 6: Jeden z pohledů na podnikové informační systémy .....	25
Obr. 7: SWOT analýza .....	30
Obr. 8: IBM Power 770 (9117-MMB).....	31
Obr. 9: Nainstalované komponenty SAP ERP.....	33
Obr. 10: SAP ERP Financials .....	34
Obr. 11: Perzistence .....	38
Obr. 12: Rozdíl mezi row-store a column-store tabulkou .....	39
Obr. 13: Slovníková komprese sloupce .....	41
Obr. 14: Systém ukládání dat SAP HANA column-based tabulek.....	42
Obr. 15: WebBIOS utility proces vytváření diskových polí.....	44
Obr. 16: IBM High IOPS MLC Mono adapter .....	47
Obr. 17: Zadní strana x3950, síťové karty se vkládají do PCIe slotů.....	48
Obr. 18: Schéma propojení jednotlivých součástí projektu .....	49
Obr. 19: Schéma SAP LT replikace se samostatným SLT serverem .....	50
Obr. 20: Prostředí Software Provisioning Manager.....	52
Obr. 21: log SWPM: problém s cílovým filesystémem.....	53
Obr. 22: Logy transportu obsahující SAP_BASIS SP update .....	55
Obr. 23: Koncept uživatelských rolí .....	56
Obr. 24: Status akcelerace analýzy ziskovosti.....	57

## **SEZNAM TABULEK**

Tabulka 1: Rozdíly mezi OLAP a OLTP.....	21
Tabulka 2: Dva různé pohledy na analýzu ziskovosti .....	35
Tabulka 3: Prodejci SAP HANA certifikovaného hardwaru.....	45
Tabulka 4: Konfigurace LPAR pro SLT systém .....	51
Tabulka 5: Rozbor verzí komponent .....	54
Tabulka 6: SAP HANA a IBM DB2: Porovnání rychlosti vybraných SQL příkazů .....	58

## **SEZNAM ZKRATEK**

<b>BI</b>	Business Intelligence
<b>CO-PA</b>	Controlling-Profitability Analysis
<b>CPU</b>	Central Processing Unit
<b>CRM</b>	Customer Relationship Management
<b>DBMS</b>	Database Management System
<b>DCL</b>	Data Control Language
<b>DDL</b>	Data Definition Language
<b>DIMM</b>	Dual in-line Memory Modul
<b>DML</b>	Data Manipulation Language
<b>EBIT</b>	Earnings Before Interest and Taxes
<b>EBITDA</b>	Earnings Before Interest, Taxes, Depreciation and Amortization
<b>ECC</b>	ERP Central Component
<b>ERP</b>	Enterprise Resource Planning
<b>GPFS</b>	General Parallel File System
<b>HMC</b>	Hardware Management Console
<b>HWCCT</b>	Hardware Configuration Check Tool
<b>IaaS</b>	Infrastructure as a Service
<b>IMDB</b>	In-memory Database
<b>IMM</b>	Integrated Management Module
<b>IS/IT</b>	Information System/Information Technology
<b>KPI</b>	Key Performance Indicator
<b>MCOS</b>	Multiple Components One System
<b>OLAP</b>	Online Analytical Processing
<b>OLTP</b>	Online Transaction Processing
<b>OS</b>	Operating System
<b>PaaS</b>	Platform as a Service
<b>QPI</b>	Intel QuickPath Interconnect
<b>RAID</b>	Redundant Array of Independent Disks
<b>RAM</b>	Random-Access Memory
<b>RFC</b>	Remote Function Call

<b>SaaS</b>	Software as a Service
<b>SCM</b>	Supply Chain Management
<b>SID</b>	System ID
<b>SLES</b>	Suse Linux Enterprise Server
<b>SLT</b>	SAP Landscape Transformation
<b>SMD</b>	Solution Manager Diagnostics
<b>SQL</b>	Structured Query Language
<b>SSD</b>	Solid State Disk
<b>SWOT</b>	Strengths, Weaknesses, Opportunities, Threats
<b>UEFI</b>	Unified Extensible Firmware Interface
<b>USB</b>	Universal Serial Bus

## **SEZNAM PŘÍLOH**

Příloha 1: Výsledky měření času pro zodpovězení SQL dotazů

## Příloha 1: Výsledky měření času pro zodpovězení SQL dotazů

##### SAP HANA #####

Statement 'select distinct A.USNAME from CE1ST00 as A left join CE2ST00 as B on A.MANDT=B.MANDT where A.GYEAR='2012''  
successfully executed in 332 ms 179 µs (server processing time: 295 ms 61 µs)  
Fetched 431 row(s) in 101 ms 484 µs (server processing time: 0 ms 247 µs)

Statement 'select distinct A.USNAME from CE1ST00 as A left join CE2ST00 as B on A.MANDT=B.MANDT where A.GYEAR='2012''  
successfully executed in 347 ms 473 µs (server processing time: 308 ms 69 µs)  
Fetched 431 row(s) in 71 ms 929 µs (server processing time: 0 ms 246 µs)

Statement 'select distinct A.USNAME from CE1ST00 as A left join CE2ST00 as B on A.MANDT=B.MANDT where A.GYEAR='2012''  
successfully executed in 334 ms 482 µs (server processing time: 278 ms 835 µs)  
Fetched 431 row(s) in 63 ms 208 µs (server processing time: 0 ms 397 µs)

---

Statement 'select \* from CE1ST00 where USNAME='BECKERF' and POSTNR='000016' and PAOBJNR='0087686479' order by POSTNR desc'  
successfully executed in 565 ms 432 µs (server processing time: 2 ms 891 µs)  
Fetched 8 row(s) in 0 ms 57 µs (server processing time: 0 ms 0 µs)

Statement 'select \* from CE1ST00 where USNAME='BECKERF' and POSTNR='000016' and PAOBJNR='0087686479' order by POSTNR desc'  
successfully executed in 218 ms 405 µs (server processing time: 2 ms 500 µs)  
Fetched 8 row(s) in 0 ms 57 µs (server processing time: 0 ms 0 µs)

Statement 'select \* from CE1ST00 where USNAME='BECKERF' and POSTNR='000016' and PAOBJNR='0087686479' order by POSTNR desc'  
successfully executed in 163 ms 924 µs (server processing time: 2 ms 901 µs)  
Fetched 8 row(s) in 0 ms 56 µs (server processing time: 0 ms 0 µs)

---

Statement 'select PALEDGE,AVG(KURSFP), SUM(KURSBK) from CE1ST00 group by PALEDGE'  
successfully executed in 458 ms 382 µs (server processing time: 419 ms 850 µs)  
Fetched 2 row(s) in 0 ms 19 µs (server processing time: 0 ms 0 µs)

Statement 'select PALEDGE,AVG(KURSFP), SUM(KURSBK) from CE1ST00 group by PALEDGE'  
successfully executed in 507 ms 309 µs (server processing time: 463 ms 966 µs)  
Fetched 2 row(s) in 0 ms 12 µs (server processing time: 0 ms 0 µs)

Statement 'select PALEDGE,AVG(KURSFP), SUM(KURSBK) from CE1ST00 group by PALEDGE'  
successfully executed in 484 ms 188 µs (server processing time: 446 ms 766 µs)  
Fetched 2 row(s) in 0 ms 15 µs (server processing time: 0 ms 0 µs)

---

Statement 'select count(distinct BUDATE) from CE1ST00 where USNAME like 'S%''  
successfully executed in 322 ms 665 µs (server processing time: 264 ms 303 µs)  
Fetched 1 row(s) in 0 ms 8 µs (server processing time: 0 ms 0 µs)

Statement 'select count(distinct BUDATE) from CE1ST00 where USNAME like 'S%''  
successfully executed in 341 ms 719 µs (server processing time: 294 ms 426 µs)  
Fetched 1 row(s) in 0 ms 8 µs (server processing time: 0 ms 0 µs)

Statement 'select count(distinct BUDATE) from CE1ST00 where USNAME like 'S%''  
successfully executed in 309 ms 737 µs (server processing time: 263 ms 752 µs)  
Fetched 1 row(s) in 0 ms 9 µs (server processing time: 0 ms 0 µs)

##### IBM DB2 #####

\* Timestamp: Fri May 22 2015 20:30:21 MESZ

-----  
\* SQL Statement Number 1:  
select distinct A.USNAME from SAPERP.CE1ST00 as A left join SAPERP.CE2ST00 as B on A.MANDT=B.MANDT where A.GYEAR='2012';  
\* 431 row(s) fetched, 431 row(s) output.  
\* Elapsed Time is: 1547.344050 seconds

\* Timestamp: Fri May 22 2015 23:47:07 MESZ

-----  
\* SQL Statement Number 1:  
select distinct A.USNAME from SAPERP.CE1ST00 as A left join SAPERP.CE2ST00 as B on A.MANDT=B.MANDT where A.GYEAR='2012';  
\* 431 row(s) fetched, 431 row(s) output.

\* Elapsed Time is: 2181.579406 seconds

\* Timestamp: Sat May 23 2015 00:42:11 MESZ

---

\* SQL Statement Number 1:

```
select distinct A.USNAME from SAPERP.CE1ST00 as A left join
SAPERP.CE2ST00 as B on A.MANDT=B.MANDT where A.GYEAR='2012';
```

\* 431 row(s) fetched, 431 row(s) output.

\* Elapsed Time is: 2322.721470 seconds

---

\* Timestamp: Sat May 23 2015 09:34:02 MESZ

---

\* SQL Statement Number 1:

```
select * from SAPERP.CE1ST00 where USNAME='BECKERF' and
POSTNR='000016' and PAOBJNR='0087686479' order by POSTNR desc;
```

\* 8 row(s) fetched, 8 row(s) output.

\* Elapsed Time is: 0.395838 seconds

\* Timestamp: Sat May 23 2015 09:38:11 MESZ

---

\* SQL Statement Number 1:

```
select * from SAPERP.CE1ST00 where USNAME='BECKERF' and
POSTNR='000016' and PAOBJNR='0087686479' order by POSTNR desc;
```

\* 8 row(s) fetched, 8 row(s) output.

\* Elapsed Time is: 0.015407 seconds

\* Timestamp: Sat May 23 2015 09:39:58 MESZ

---

\* SQL Statement Number 1:

```
select * from SAPERP.CE1ST00 where USNAME='BECKERF' and
POSTNR='000016' and PAOBJNR='0087686479' order by POSTNR desc;
```

\* 8 row(s) fetched, 8 row(s) output.

\* Elapsed Time is: 0.014441 seconds

---

\* Timestamp: Sat May 23 2015 09:41:43 MESZ

---

\* SQL Statement Number 1:

```
select PALEDGE,AVG(KURSFP), SUM(KURSBK) from SAPERP.CE1ST00
group by PALEDGE;
```

\* 2 row(s) fetched, 2 row(s) output.

\* Elapsed Time is: 4695.474591 seconds

\* Timestamp: Sat May 23 2015 11:04:52 MESZ

---

\* SQL Statement Number 1:

```
select PALEDGE,AVG(KURSFP), SUM(KURSBK) from SAPERP.CE1ST00
group by PALEDGE;
* 2 row(s) fetched, 2 row(s) output.
* Elapsed Time is:      3370.118383 seconds
```

```
* Timestamp: Sat May 23 2015 12:28:40 MESZ
-----
```

```
* SQL Statement Number 1:
select PALEDGE,AVG(KURSFP), SUM(KURSBK) from SAPERP.CE1ST00
group by PALEDGE;
* 2 row(s) fetched, 2 row(s) output.
* Elapsed Time is:      3011.545325 seconds
```

```
* Timestamp: Sat May 23 2015 15:53:51 MESZ
-----
```

```
* SQL Statement Number 1:
select count(distinct BUDATE) from SAPERP.CE1ST00 where USNAME
like 'S%';
* 1 row(s) fetched, 1 row(s) output.
* Elapsed Time is:      2468.670143 seconds
```

```
* Timestamp: Sat May 23 2015 16:45:40 MESZ
-----
```

```
* SQL Statement Number 1:
select count(distinct BUDATE) from SAPERP.CE1ST00 where USNAME
like 'S%';
* 1 row(s) fetched, 1 row(s) output.
* Elapsed Time is:      2287.380902 seconds
```

```
* Timestamp: Sat May 23 2015 18:06:11 MESZ
-----
```

```
* SQL Statement Number 1:
select count(distinct BUDATE) from SAPERP.CE1ST00 where USNAME
like 'S%';
* 1 row(s) fetched, 1 row(s) output.
* Elapsed Time is:      1782.739417 seconds
```