

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Katedra informačních technologií

Studijní program : Aplikovaná informatika

Obor: Informační systémy a technologie

**Automatizace a vizualizace
projektových nástěnek v konkrétním
podniku**

DIPLOMOVÁ PRÁCE

Student : Bc. Jakub Vancí

Vedoucí : Ing. David Slánský, Ph.D.

Oponent : doc. Ing. Jan Pour, CSc.

2016

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a že jsem uvedl všechny použité prameny a literaturu, ze které jsem čerpal.

V Praze dne 27. dubna 2016

.....

Jakub Vancí

Poděkování:

Děkuji vedoucímu práce, Ing. Davidovi Slánskému Ph.D., za ochotu a čas věnovaný konzultacím.

Dále děkuji svým rodičům, kteří mi umožnili studovat.

Abstrakt:

Cílem práce je vytvořit a popsat řešení automatizace a vizualizace projektových nástěnek v konkrétním výrobním podniku, při čemž jsou využity metody a nástroje Business Intelligence. Projektovou nástěnku si v kontextu této práce lze představit jako uspořádaný souhrn informací, které je potřeba v rámci projektu sledovat. Vyskytují se zde tedy například informace o vyrobených kusech, nákladech na nekvalitu, project scorecard, analýza poruch, úzké místo projektu, a další. Teoretická část se zabývá popisem metod a komponent Business Intelligence a slouží především pro vysvětlení základních principů, na základě který je vytvořeno dočasné uložisko, datový sklad, OLAP kostky a dashboardy. Hlavní důraz je kladen na praktickou část, ve které je již detailně popsán proces zpracování dat od těch zdrojových, kdy je čerpáno především ze souborů typů MS Excel, až po vizualizaci dat v podobě dashboardu. Jako softwarové prostředí byly vybrány produkty od firmy Microsoft a to SQL Server Management Studio, Visual Studio a již zmíněný Excel.

Hlavním výstupem je řešení, které postupně plně nahradí papírové projektové nástěnky. Toto řešení dále přináší jak finanční, tak časovou úsporu při tvorbě a správě projektových nástěnek, jelikož je tento proces automatizován.

Klíčová slova:

Vizualizace, automatizace, Business Intelligence, dashboard, projektová nástěnka

Abstract:

The aim of the work is create and describe solution of automation and visualization of project boards in the specific manufacturing company. Methods and tools of Business Intelligence are used in this process. Project board is arranged summary of information, which is part of the monitor, in this context. There is for example information about produced pieces, the cost of poor quality, project Scorecard, downtime analyses, bottleneck of project, and so on. The theoretical part is dealing with description methods and components of Business Intelligence and serves to explanation basic principles on the basis of which are formed data staging area, data warehouse, OLAP cubes and dashboards. The main emphasis is on the practical part, in which is detail describe process of data elaboration from source data, which are taken primarily from MS Excel files, till data visualization in the form of dashboard. As software workspace were selected Microsoft products, specifically SQL Server Management Studio, Visual Studio and Excel which has been mentioned before.

The main output is solution that will gradually fully replace paper project boards. This solution also brings both financial and time savings in the creation and managing project boards, because this process is automated.

Keywords:

Visualization, automation, Business Intelligence, dashboard, project board

Obsah

| | |
|--|----|
| Úvod | 8 |
| Teoretická část | 10 |
| 1 Strategie rozvoje | 11 |
| 1.1 Analýza požadavků | 11 |
| 1.1.1 Počáteční workshop | 12 |
| 1.1.2 Interview | 12 |
| 1.2 Strategické záměry | 13 |
| 1.3 Analýza připravenosti | 14 |
| 1.4 Návrh architektury | 15 |
| 2 Analýza dat | 16 |
| 2.1 Top-Down logické modelování | 16 |
| 2.1.1 Standardizace businessových meta dat | 17 |
| 2.2 Bottom-Up analýza zdrojových dat | 18 |
| 2.2.1 Pravidla změny datového typu | 18 |
| 2.2.2 Pravidla datové domény | 18 |
| 2.2.3 Pravidla datové integrity | 19 |
| 2.3 Čištění dat | 19 |
| 2.3.1 Zodpovědnost za datovou kvalitu | 19 |
| 2.3.2 Určení zdrojových dat | 20 |
| 2.4 Proč je datová analýza důležitá | 21 |
| 3 Návrh řešení business intelligence | 22 |
| 3.1 Ukládání dat | 23 |
| 3.1.1 Datový sklad | 24 |
| 3.1.2 Datové tržiště | 25 |
| 3.1.3 Dočasné uložení dat | 26 |
| 3.1.4 Operativní datové uložení | 28 |
| 3.1.5 Porovnání vlastností datových uložení | 28 |
| 3.2 ETL – Extraction, Transformation and Loading | 29 |
| 4 Dashboard | 31 |
| 4.1 Charakteristika | 31 |
| 4.2 Základní chyby návrhu dashboardu | 32 |

| | |
|---|----|
| Praktická část | 38 |
| 5 Analýza požadavků | 39 |
| 5.1 Analýza připravenosti..... | 41 |
| 5.2 Návrh architektury | 42 |
| 6 Analýza dat..... | 43 |
| 6.1 Zdrojová data | 43 |
| 6.1.1 Pyramida BOZP, Pyramida kvality a Pyramida produkce | 43 |
| 6.1.2 Plán opatření | 45 |
| 6.1.3 Pracovní úraz; Informace o projektu; Foto týmu; Úzké místo projektu | 45 |
| 6.1.4 Z-Grafy..... | 45 |
| 6.1.5 Analýza poruch..... | 46 |
| 6.1.6 Project Scorecard | 46 |
| 6.1.7 OEE, HIP..... | 47 |
| 7 Návrh řešení | 49 |
| 7.1 Dočasné uložení dat | 49 |
| 7.2 Datový sklad | 49 |
| 7.3 ETL proces | 51 |
| 7.3.1 Project Scorecard | 51 |
| 7.3.2 Analýza poruch..... | 57 |
| 7.3.3 Z-Grafy..... | 60 |
| 7.3.4 OEE, HIP..... | 64 |
| 7.3.5 Pyramida BOZP, Pyramida kvality a Pyramida produkce | 67 |
| 7.3.6 Plán opatření..... | 68 |
| 7.4 OLAP kostky..... | 69 |
| 8 Projektová nástěnka – dashboard..... | 70 |
| 9 Vlastní přínos..... | 72 |
| Závěr | 73 |
| Seznam literatury:..... | 74 |
| Seznam obrázků:..... | 75 |
| Seznam tabulek:..... | 77 |

Úvod

Business Intelligence se pro mnohé firmy stalo nezbytnou součástí informačních technologií. Některé se bez business intelligence neobejdou z důvodu reportingu státním organizacím jako jsou například regulátoři. Ti předepisují, jaké reporty s jakou intenzitou musí firma vykazovat. Některé podniky zase mají business intelligence z vlastního přesvědčení. Vědí, že dnešní doba, kdy je téměř každý druh podnikání zatěžován velkým množstvím dat, si žádá řešení, které zajistí lepší čitelnost dat tedy jakýsi převod od dat k informacím a to dobře zajišťuje právě business intelligence.

Cílem této diplomové práce je popsat a vytvořit řešení pro automatizaci a vizualizaci projektových nástěnek pro konkrétní výrobní firmu působící v automobilovém průmyslu, jako dodavatel dílů pro automobilové i motocyklové výrobce. Řešení tohoto problému spočívá ve vytvoření datového skladu, do kterého budou nahrána zdrojová data, která jsou dodávána především skrze aplikaci MS Excel. Dalším krokem je vytvoření vhodné struktury OLAP kostek, které se budou opakovaně vytvářet podle četnosti požadovaných vizualizačních prvků. Na základě OLAP kostek, jsou dále vytvářeny jednotlivé vizualizační prvky pro jednotlivé projekty, které dohromady tvoří celkový dashboard pro daný projekt, čili projektovou nástěnku. Tyto nové digitální projektové nástěnky budou postupně nahrazovat ty papírové, které se musí pokaždé ručně vytvářet. Největší přínos je tedy v úspoře lidské práce. Vedlejší přínosy jsou zpřehlednění a zatraktivnění vzhledu projektových nástěnek, možnost vytváření různých porovnání projektů mezi sebou a v neposlední řadě historizace nasbíraných dat. Jako vývojové prostředí použiji převážně software od firmy Microsoft.

Řešení problému v této diplomové práci není typický problémem business intelligence. Jsou však využity metody a nástroje, které business intelligence používá a následně jsou aplikovány na vytvoření projektových nástěnek.

To co nejvíce uživatelů business intelligence systémů vidí, je uživatelské rozhraní jejich reportingového programu, což je vlastně správně. Pro uživatele je všechno to co předchází reportu španělská vesnice (black box). Uživatelé sice většinou vědí, že data, která potřebují pro svá rozhodnutí, jsou sbírána z produkčních systémů, ale už pro ně není podstatné, jakou cestou tato data šla, než se z nich stal report. Dokud mají přístup ke správným datům a odezva je dostatečně rychlá, tak jsou spokojeni. Od zdrojových dat k reportům, ale vede dlouhá cesta. [1, str. 30]

Prací, zabývajících se problematikou business intelligence je v současné době poměrně hodně. Jsou však často zaměřené na teoretické poznatky z této oblasti. Mnoho prací se také zabývá reportingem a dashboardingem, například [2], kde se autorka zaměřila především na srovnání tří nástrojů pro tvorbu dashboardů a na základě Fullerovy metody interpretuje dosažené výsledky.

David Sochora se ve své diplomové práci [3] zabývá také procesem automatizace, ovšem s jiným pojetím. Automatizuje proces zpracování finančních dat pro potřeby auditu. Zdrojovými daty jsou finanční výkazy ze systému SAP.

Další prací zabývající se implementací business intelligence v konkrétním podniku je [4]. Jedná se o podnik působící ve farmacii. Autor ve své práci klade důraz především na praktickou stránku problému. Nicméně popisuje i teoretické náležitosti. Některé pasáže jsou obecnějšího charakteru.

Tato diplomová práce je jedinečná především proto, že se zabývá implementací automatizace a vizualizace projektových nástěnek v konkrétním výrobním podniku, přičemž využívá metod business intelligence. Pro praktickou část práce jsou použity převážně zdroje pojednávající jak o business intelligence obecně, tak i konkrétněji o jednotlivých komponentách.

Kniha [1] je použita jako zdroj pro vysvětlení datových uložišť. Je zde dobře popsáno kdy a za jakých okolností je vhodné využívat volitelné komponenty, jako například dočasné či operativní uložště dat. Autorem je Rick F. van der Lans, který má bohaté zkušenosti jak z oblasti business intelligence, tak i z oblasti servisně-orientované architektury a databázových technologií.

Druhým zdrojem je [5]. Na této knize se podíleli pedagogové Vysoké školy ekonomické v Praze. Poskytuje obecný přehled o problematice business intelligence. Je zde obsažen jak proces příprav a řízení projektu nasazení business intelligence, tak i samotný proces implementace.

Zdroj [6] je psán jako jakýsi plán budování business intelligence. Vzhledem k počtu stran, je rozsah knihy opravdu široký. Pojednává detailně jak o procesu plánování a přípravy, tak i o samotném provedení implementace a následném zlepšování business intelligence. Velmi podrobně je zde popsán i proces analýzy dat.

William Immon ve své knize [7] pojednává o datawarehousingu. Autor je považován za průkopníka v této oblasti.

Kniha [8] je zaměřena na dashboarding. Popisuje principy, jak správně data vizualizovat, aby přinesla co největší přidanou hodnotu. Součástí knihy je i část, která poukazuje na nejčastější chyby při vizualizaci dat.

Teoretická část

1 Strategie rozvoje

Tato kapitola se zabývá detailnější specifikací celkové koncepce rozvoje řešení business intelligence. Dále pak konkretizuje dílčí cíle. Před samotnou realizací řešení je potřeba si uvědomit několik faktorů, mezi něž patří: jaké jsou požadavky, strategické záměry, zdali je firma připravená pro nasazení business intelligence a samozřejmě také návrh architektury. Tyto oblasti jsou blíže specifikovány v následujících podkapitolách. Než k nim však přistoupím, měl bych ujasnit, proč je důležité zabývat se touto kapitolou na prvním místě a nezačít rovnou s výstavbou datového skladu. Ač se to z počátku nemusí jevit možné, přípravou téměř každého projektu je ve finále ušetřen čas. Je totiž velmi pravděpodobné, že pokud dojde k zanedbání příprav, dochází poté k nesrovnalostem a častému předělování komponent, což nakonec často zabere více času než důkladná příprava. V horším případě se může stát, že výsledné řešení, které zabralo mnoho času, není v souladu s představami zadavatele a musí dojít k nápravě. Úprava řešení přitom může být tak rozsáhlá, že je potřeba se vrátit na úplný začátek. Čili to, že bylo ušetřeno několik hodin na začátku projektu a nebyla provedena strategie rozvoje, může vést k mnoha hodinám zbytečné práce. Ekvivalentní příklad z jiného oboru než informační technologie, na kterém lze ilustrovat důležitost příprav je náčrt obchodního domu. Pokud architekt nedostane jasné a přesné zadání o stavbě, má v základě dvě možnosti. První je navrhnout obchodní dům podle jeho představ o tom jak by měl taková správný obchodní dům vypadat a co všechno by měl mít. Druhou možností je domluvit si schůzku se zadavatelem a na všechny náležitosti se zeptat a postupně si tím vytvářet obrázek o tom jak bude obchodní dům vypadat. Jistě tušíte, že správně je druhá možnost. Tím však nechci vyloučit případy, kdy dostane architekt volnou ruku a celá stavba je prakticky pouze na jeho představách. Ale těchto případů je jistě minimum. Vráťme-li se k tématu, tedy business intelligence, jen těžko si lze představit situaci, kdy by bylo možné vytvořit kvalitní řešení pouze na základě strohé požadavky vytvořit business intelligence pro výrobní podnik. Domnívám se, že nyní už není pochyb o tom, že úloha přípravy a plánování by měla být nedílnou součástí téměř každého projektu.

1.1 Analýza požadavků

Obsahem řešení business intelligence jsou prostředky, které vedou k odpovědím na otázky, které uživatelé hledají v tomto určených aplikacích. V tomto kroku, tedy dochází k formulaci základních požadavků na řešení, které se v dalších fázích konkretizují.

K základním technikám, které pomohou zjistit požadavky, patří interview a průběžné oponentury neboli workshopy. Z praxe vyplývá, že k prvotní analýze požadavků je vhodné uskutečnit tak zvaný seznamovací workshop, který je následován jednotlivými interview, která jsou vedena v malých skupinách nebo s jednotlivci. Poznatky je poté vhodné ověřovat na větších setkání a oponenturách. V menších skupinách lze totiž lépe zaznamenávat dílčí názory účastníků a také odhalit různé nekonzistence v chápání podnikové reality. V souvislosti s praktickou částí této diplomové práce se za nekonzistentní chápání mohou považovat různá

pojmenování jednoho projektu. Například některá oddělení nazývají jeden z projektů MQB VAT, ale jiná oddělení tento stejný projekt nazývají MQB Hilfsrahmen. Tyto dílčí nekonzistence by měly být během sezení dokumentovány a dále řešeny na větších setkáních, která by se mimo jiné měla pokusit nalézt souladu mezi tímto nekonzistentním chápáním. [5, str. 97]

1.1.1 Počáteční workshop

Úlohou počátečního workshopu je seznámení se účastníků analýzy. Tím pádem by se ho měli zúčastnit všichni členové řešitelského týmu. Řešitel by měl prezentovat následující skutečnosti:

- Čeho se bude analýza týkat a co je jejím cílem.
- Z jakého důvodu je potřebná přítomnost vybraných pracovníků firmy.
- Harmonogram analýzy.
- Jaké nároky jsou kladeny na pracovníky podniku (časové, znalostní, technické).

Počáteční workshop může zásadně ovlivnit další postup projektu a to především negativně, pokud bude tento workshop špatně organizován či dokonce vůbec neproběhne. [5, str. 98]

1.1.2 Interview

Ke shromáždění nezbytných detailů v důsledku pochopení business procesů, musí osoba nebo tým, který je zodpovědný za interview strávit nějaký čas rozhovorem s pracovníky podniku, kteří mají na starosti aplikace související s analýzou a pochopit jejich prostředí. K dokumentování je vhodné používat grafické pomůcky jak jen je možné, například bublinkové grafy, digram příčin a následků, digramy zachycující vztahy mezi entitami a mnoho dalších diagramů. Diagramy jsou totiž výborným komunikačním nástrojem. Skrze vizualizaci může dotazovaný lépe pochopit, na co se dotazující ptá a tím pádem dochází k usnadnění komunikace.

Před tím než dojde k naplánování interview, je dobré se nejprve připravit. Především je vhodné zvážit tyto položky:

- Interview tým: Doporučuje se, aby člověk, který vede interview, zároveň nesepisoval poznámky. To by měla dělat osoba, na které a se kterou se předem dohodneme, aby už samotné interview nebylo narušováno dohady, kdo bude zapisovat. Zapisování odpovědí totiž není dobře slučitelné s přemýšlením nad další otázkou.
- Dotazování: Interview mohou být vedeny s jednotlivci nebo se skupinou. Skupinové interview jsou vhodné, především pokud členové této skupiny jsou sobě rovni ve smyslu stejné pracovní náplně či odpovědnosti. Na základě toho, co říká jeden člen, mohou reagovat ostatní a tím vzniká plodná diskuze. Oproti tomu nevýhodou skupinového interview je skutečnost, že někteří dotazovaní nemusejí být z nějakého důvodu upřímní, jelikož nejsou anonymní. Mohou se například obávat před svými kolegy zmínit nějaký nedostatek, jelikož by mohli být později napomínáni.

- Výzkum: Před naplánováním interview by si měl tazatel projít existující dokumenty, reporty, webové stránky včetně těch konkurenčních a zjistit co nejvíce informací o daném trhu, podnikových procesech, firemní terminologii a zkratkách.
- Tazatel: Tazatel by měl mít dopředu připravená hlavní témata. Je dobré je ještě před samotným interview rozeslat účastníkům (dotazovaným), aby se také mohli připravit a třeba i přinést s sebou nějaké podpůrné dokumenty.
- Plán interview: Nedoporučuje se plánovat více než čtyři jednohodinové interview denně, protože minimálně jednu hodinu zabere zpracování a posouzení poznámek. Je nezbytné dokončit nebo přepsat poznámky již tentýž den, který proběhlo interview, jelikož je ještě interview v živé paměti a je možné na základě poznámek správného vyhodnocení.

Tipy pro interview

- Počáteční interview by se mělo zaměřit na základní požadavky nezbytné k řešení problému. Naopak by nemělo obsahovat mnoho technických aspektů. Dále by nemělo docházet k bezhlavým slibům, například co všechno bude vytvořeno a jak to bude rychle. Ten čas kdy tyto věci budou zodpovězeny, přijde.
- Často dotazovaní rádi mluví o tom, co v současné době mají, ale dovedou poskytnout pouze minimum informací o tom, co chtějí, ale nemají to. Je potřeba být připraven navést je k odpovědím.
- Je možné se setkat se střetem různých pohledů na jednu věc. Nejčastěji k tomu dochází, když probíhá jednání s podnikovými analytiky a manažery z různých oddělení a z různých pozic v podnikové hierarchii.
- Pokud se interview tým skládá pouze z jednoho člena, je dobré si pořizovat audio záznam. Díky tomu nedochází k rozptylování psaním poznámek a je možné se soustředit na otázky a odpovědi. Dotazovaný by však měl být na začátku požádán o souhlas se záznamem. S tím souvisí však nevýhoda, která již byla zmíněna v souvislosti se skupinovým interview. Dotazovaný, který ví, že je nahráván, nemusí být zcela upřímný.
- Po zpracování poznámek z interview do přehledné a jasné podoby je možné tyto zápisky zaslat dotazovaným s tím, že mohou něco případně doplnit, pokud je něco napadlo. To však vyžaduje větší časovou zátěž pro obě strany, proto je tento krok na zvážení. [6, str. 162-163]

1.2 Strategické záměry

Výsledky předchozích zkoumání nyní poslouží k formulaci hlavních strategických záměrů řešení business intelligence. Důležité otázky pro další postup zahrnují:

- Určení potřeb vedení podniku, manažerů a specialistů na implementaci či inovaci BI. Tyto potřeby musí být v souladu s podnikovou strategií, strategií rozvoje podnikového řízení i informační strategií.

- Vyjmenování efektů, kterých by mělo být docíleno od aplikací business intelligence. Mezi tyto efekty patří například obchodní, ekonomické a další efekty, které mají pro podnik strategický význam.
- Definování základní funkcionality IS/ICT jako celku. Tedy identifikace funkcí, které budou zajišťovány na úrovni business intelligence a které funkce budou zajišťovány ostatními systémy jako například CRM. Tím lze také mimo jiné určit, kde se tyto funkce překrývají nebo doplňují.
- Přibližná specifikace funkcionality a určení osob zodpovědných za skupiny uživatelů a informatiku jako celek.
- Určení pozice business intelligence v architektuře IS/ICT včetně vazeb na aplikace, které budou poskytovat zdrojová data.
- Popis architektury řešení business intelligence a určení technologické platformy.
- Seznámení s dopady business intelligence na personální organizaci firmy. Rozhodnutí zda se vytvoří speciální oddělení zabývající se analytickými aplikacemi a také určení kdo bude zodpovědný za chod těchto aplikací a kdo bude poskytovat podporu uživatelům.
- Vytvoření harmonogramu projektu zavedení business intelligence řešení.

Strategii business intelligence lze řešit více způsoby. Je možné ji integrovat do informační strategie podniku nebo ji oddělit jako zvláštní úlohu, včetně speciální dokumentace. Podstatné však je dbát na to, aby strategie business intelligence byla s informační strategií v souladu.

1.3 Analýza připravenosti

Plánování business intelligence se příliš neliší od plánování ostatních součástí IS/ICT, avšak specifickou otázkou je vstupní vyhodnocení připravenosti firmy na business intelligence. Připravenost především z pohledu personálního, technologického a datového. Analýza stavu je na této úrovni prováděna pouze v základním rozměru a má poměrně nízkou úroveň detailu. Hledá odpovědi na tyto základní otázky:

- Jaké jsou současné klíčové aplikace IS/ICT a jaké další aplikace jsou do budoucna plánované. Zajímá nás především, jaká data tyto aplikace produkují a spravují a zdali jsou použitelná pro řešení business intelligence.
- Jaké jsou k dispozici interní i externí datové zdroje, dále pak jaká je jejich dostupnost, validita a kvalita. Důležitým údajem je tak informace o tom, kdo za tyto zdroje zodpovídá a jak jsou dislokovány.
- Jaké technické vybavení je schopna poskytnout současná infrastruktura pro řešení business intelligence. Konkrétně nás zajímají konfigurace serverů, koncových stanic, verze operačních a databázových systémů.

Nejspíše nejpodstatnější součástí této analýzy je posouzení stavu datových zdrojů a jejich kvality.

1.4 Návrh architektury

Podstatným aspektem návrhu architektury je nahlížet na business intelligence jak na jeho celkovou pozici v IS/ICT, tak i na jeho vnitřní architekturu, znázorňující rozložení hlavních komponent a relací mezi nimi.

Business intelligence řešení je charakteristické silným provázáním cílových analytických a plánovacích aplikací, ale i různých technologických nástrojů, kterými jsou například ETL pumpy a OLAP nástroje, s datovými zdroji.

Návrh architektury business intelligence řešení se skládá z těchto hlavních činností:

- Definování produkčních aplikací a databází, které slouží pro vstup do business intelligence řešení.
- Rozhodnutí o využití a případně umístění dočasného uložení – (DSA) Data Staging Area a operativního uložení dat – ODS (Operational Data Store).
- Definování nástrojů pro extrakci, transformaci a načtení – ETL (Extraction Transformation and Loading).
- Definování nástrojů pro tvorbu a správu datových skladů – DWH (Data Warehouses) a datových tržišť – DMA (Data Marts).
- Definování nástrojů pro vytvoření OLAP databází.
- Určení klientských nástrojů.
- Určení business intelligence aplikací a jejich využití.
- Definování provázanosti jednotlivých produktů nebo modulů.
- Vysvětlení, jak je řešení schopné v případě potřeby postupně připojovat další produkty a nástroje.
- Definování jakým způsobem je zajištěna správa a monitoring provozu business intelligence.
- Vysvětlení, jak jsou implementovány dokumentace a metodiky. [5, str. 101-103]

2 Analýza dat

V této kapitole je čerpáno z [6, str. 173-198] a zabývá se okolnostmi, které je třeba zvážit při analýze dat pro business intelligence, Top-Down (shora dolů) logickým datovým modelováním, Bottom-Up (zespodu nahoru) analýza zdrojových dat včetně toho jak aplikovat tři soubory pravidel transformace na zdrojová data – pravidlo změny datového typu, pravidlo datové domény a pravidlo datové integrity. Dále jsou zde pak rozebírány odpovědnosti za čištění dat a dodržování datové kvality. V neposlední řadě je zde také rozebráno jaké následky může přinést zanedbání analýzy dat.

Pro mnoho podniků je business intelligence podnět k prvnímu pokusu shromáždit data z různých zdrojů do jednotné podoby. Firmy, které používají tradiční systémy metodiky vývoje k jejich business intelligence projektům obvykle naráží na velké problémy s datovými zdroji, když se pokouší implementovat ETL procesy. To proto, že tradiční metodiky vývoje nemají postupy pro brzkou analýzu datových domén v procesu vývoje. Při nejlepším mají fázi analýzy pouze pro klíčová data.

2.1 Top-Down logické modelování

Jednou z nejefektivnějších technik pro objevení a zdokumentování pohledu na podniková data je entitně-relační modelování, známé též jako logické modelování. Oblíbeným přístupem k entitně-relačnímu modelování, se kterým se přišlo začátkem 80. let, bylo modelování všech dat pro celou firmu najednou. I když byl tento přístup návrhově zajímavý, bohužel to nestačilo na lepší systémy, protože proces nebyl integrován se systémem vývojového životního cyklu. Efektivnější přístup je zahrnout entitně-relační modelování do každého jednotlivého projektu a poté sloučit jednotlivé logické modely do jednoho konsolidovaného firemního datového modelu.

Entitně-relační modelování je založeno na normálních formách, které jsou aplikovány během Top-Down modelování stejně tak jako během Bottom-Up analýze zdrojových dat. Použitím normálních forem společně s ostatními administrativními principy je zajištěno, že každý datový prvek hodnotný pro business intelligence řešení je jednoznačně identifikovatelný, správně pojmenovaný a řádně definovaný a že jeho datové domény jsou schválené pro všechny uživatele, kteří budou přistupovat k datům. To znamená, normalizovaný logický model reprezentuje data přesně v té podobě, jako jsou v reálném světě bez redundance a dvojznačnosti.

Významnou vlastností, kterou má logický model je fakt, že je nezávislý na budoucí databázi, programu, nástrojích, hardwaru. Je to z toho důvodu, že logický model představuje pohled businessu, nikoli databázový ani aplikační pohled. Proto tedy nějaká konkrétní data, která existují pouze jednou v reálném business světě, také tím pádem existují pouze jednou v logické modelu, ačkoli mohou být fyzicky uložena v různých zdrojových souborech.

Při tvorbě logického modelu může výrazně práci ulehčit datový administrátor, který má slušné znalosti o daném businessu. Pokud však tyto znalosti nemá, musí vyhledat pomoc experta v dané oblasti a s ním se při návrhu radit. Pokud jsou zdrojová data čerpána z různých zdrojů, pak je s tím spojeno, že by se vlastníci či zodpovědné osoby za tyto data měli účastnit projektu. Vlastníci dat jsou totiž ti lidé, kteří mají pravomoc zřídit business pravidla a nastavit business politiky pro ta data, který pochází z jejich oddělení. Pokud se objeví v datech nesrovnalosti, má právě vlastník těchto dat zodpovědnost za uspořádání rozdílných business pohledů a schválit legitimní použití těchto dat. Konsolidace dat je a měla by být business funkce, nikoli funkce informačních technologií, ačkoli je datový administrátor zaměstnanec IT. Dalšími osobami, kteří by se měli podílet na business intelligence projektu jsou systémoví analytici, vývojáři a databázoví administrátoři. Tito techničtí pracovníci, kteří mají na starosti správu aplikací firmy a datovou strukturu, často vědí o datech více než kdokoli jiný. Jde především o znalosti, kde a jak jsou data uložena, jak jsou zpracovávána a jak je používají lidé z businessu. Navíc tyto pracovníci často mají detailnější znalosti o přesnosti dat, jak jedny data souvisí s ostatními a jak se význam konkrétních dat měnil v čase. Je důležité dopředu zajistit součinnost s těmito lidskými zdroji, protože právě tyto lidé jsou často zaneprázdnění „hašením“ problémů a vylepšováním stávajících aplikací.

2.1.1 Standardizace businessových meta dat

Business meta data obsahují informace o business datech, datových prvcích a o vazbách mezi nimi. Meta data jsou přidávána během návrhu a zajišťují konzistenci a lepší pochopení dat a také interpretaci dat v business intelligence. Jednotlivé prvky meta dat jsou rozebrány níže:

- **Jméno:** Oficiální označení vytvořeno z formální taxonomie. Pojmenování by mělo být jednoznačné a v jednom logickém modelu by se neměly vyskytovat synonyma a homonyma.
- **Definice:** Stručný popis, složený z jedné nebo dvou vět. Podobný například tomu z jazykových slovníků. Pokud má jeden datový objekt více podtypů, každý z nich by měl mít vlastní definici. Definice tedy vysvětluje význam datové objektu či prvku. Neměla by obsahovat, kdo daný datový objekt vytvořil nebo například kdy proběhla poslední aktualizace. Pro tyto informace jsou určeny jiné prvky meta dat.
- **Vazba:** Je business asociace mezi daty. Každá taková vazba je založena na business pravidlech a business politikách.
- **Identifikátor:** Jednoznačně identifikuje výskyt datového objektu. Identifikátor by měl být znám business lidem. Měl by být minimální čímž je myšleno, že identifikátor by měl být co nejkratší. Navíc by měl být jednoduchý, bez žádné vložené logiky. Identifikátor v tomto kontextu není to samé jako primární klíč. Ačkoliv může být tak využit, často je nahrazován náhradním (vymyšleným) klíčem během návrhu databáze.
- **Datový typ:** Popisuje strukturu datových prvků, kategorizuje typ hodnot – čísla, text, datum, ..., které mohou být uloženy.
- **Délka:** Specifikuje velikost datového prvku pro konkrétní datový typ. Například desetinné číslo může mít za desetinnou čárkou jedno, dvě či více čísel, v závislosti na tom, jak se nadefinováno.

- **Obsah (doména):** Identifikuje aktuální přípustné hodnoty pro datový prvek. Tyto hodnoty musí být v souladu s datovým typem a délkou. Doména může být vyjádřena jako rozsah hodnot, seznam přípustných hodnot, obecné business pravidlo nebo závislost mezi dvěma či více datovými prvky.
- **Pravidlo:** Představuje jakési omezení datového objektu, prvku či vazby. Datová omezení mohou být ve formě business pravidla nebo závislosti mezi datovými objekty nebo prvky. Například vrchní úroková sazba musí být vyšší než dolní úroková sazba.
- **Politiky:** Spravují obsah a chování datových objektů nebo prvků. Obvykle jsou vyjádřeny jako organizační politiky nebo legislativní regulace. Například aby si člověk mohl zřídit bankovní účet, musí být starší patnácti let.
- **Vlastnictví dat:** Určuje osoby, které mají pravomoci zřídit a schválit business meta data pro datové objekty nebo prvky pod jejich kontrolou.

2.2 Bottom-Up analýza zdrojových dat

Datová analýza by neměla skončit po top-down logickém modelování, protože zdrojová data často nenásledují business pravidla a politiky zachycené během modelování. Jestliže by bottom-up analýza zdrojových dat nebyla provedena, problémy s daty a porušení business pravidel by nemuselo být odhaleno, dokud se nezačnou implementovat ETL procesy. Některé problémy s datovou kvalitou však nemusí být odhaleny ani při implementaci ETL a přijde se na ně, až když si někdo stěžuje. Zdrojová data se nemapují pouze na pravidla změny datového typu, ale také na pravidla datové domény a pravidla datové integrity. Tato pravidla jsou dále přiblížena.

2.2.1 Pravidla změny datového typu

Vždy když jsou data mapována z jednoho systému do druhého, musí být uplatněna následující pravidla.

- Datové typy zdrojových dat se musí shodovat s datovými typy cílových dat.
- Datová délka musí být přiměřená, aby byl možný přesun či rozšíření zdrojových dat do cílových dat.
- Logika programů, které manipulují se zdrojovými daty, musí být kompatibilní a použitelná na obsah zdrojových dat. V opačném případě je možné dostat nepředvídatelné výsledky.

2.2.2 Pravidla datové domény

Tato pravidla jsou mnohem důležitější pro business lidi než předchozích pravidlo. Na zdrojová data mohou být aplikována všechna tři pravidla o změně datového typu, a přesto obsahují nesprávná hodnoty. Pravidla datové domény jsou o sémantice, tedy o významu datového obsahu. Jsou používána k identifikaci a opravě chyb v datech. Příklady takových chyb jsou znázorněny v bodech níže.

- Chybějící hodnoty – velký problém business intelligence řešení

- Výchozí hodnoty – například „0“, „999“ nebo prázdná hodnota
- Inteligentní hodnoty, které jsou specifikovány výchozí nebo inteligentními hodnotami, které mají význam – například použitím hodnoty 000000/0000 pro rodné číslo, může být vyjádřena skutečnost, že tato osoba není občanem České republiky.
- Záhadný datový obsah – například použití hodnot „A, B, C, D“ pro datové prvky za účelem definování kategorie zákazníka, zatímco hodnoty „E, F, G, H“ stejného datového prvku definují typ propagace a hodnoty „I, J, K, L“ definují oblast umístění.
- Víceúčelové datové prvky vložené ve zřetěžených volných textových polích – například řádek adresy 1 – 5 obsahující jména a adresová data.

2.2.3 Pravidla datové integrity

Podobně jako tomu tak bylo u pravidel datové domény, tak pravidla datové integrity jsou mnohem více důležitá k dokázání kvality informací než pravidla o změně datových typů. Pravidla datové integrity spravují sémantický obsah napříč závislostí nebo souvisejícími datovými prvky, stejně tak jako omezení stanovená business pravidly a politikami. Příklady, kde jsou porušena pravidla datové integrity, jsou vyspána níže.

- Rozpor datového obsahu mezi dvěma či více datovými prvky – například „Praha, SVK“ místo „Praha, ČR“.
- Porušení Business pravidla – například pro stejnou osobu je uvedeno datum narození 13.2.1970 a datum úmrtí 19.9.1965.
- Opakované použití primárního klíče – například dva zaměstnanci mající stejné zaměstnanecké číslo. To může nastat, pokud například po odchodu jednoho zaměstnance dostane jeho číslo nový zaměstnanec.
- Primární klíč není unikátní – například jednomu zákazníkovi bylo přiřazeno více identifikačních čísel (primárních klíčů).
- Objekty, bez k nim přiřazených závislých objektů – například práce je přiřazena zaměstnanci s ID 2456, ovšem v databázi žádný zaměstnanec s tímto ID není.

2.3 Čištění dat

Jedním z cílů, který je uváděn stále častěji v souvislosti se zavedením business intelligence, je dodat data čistá (kvalitní) a integrovaná. Mnoho organizací zjistí, že jejich data ve zdrojových systémech jsou mnohem více nekvalitní, než očekávali. Tím před ně vyvstává náročné rozhodnutí a to do jaké hloubky budou data čištěna.

2.3.1 Zodpovědnost za datovou kvalitu

Datová archeologie – proces hledání špatných dat, čištění dat – proces opravy špatných dat a vynucení datové kvality – proces prevence znečištění dat jsou všechno procesy spadající pod zodpovědnost businessu, nikoli IT. To znamená, že business lidé, tedy ti co spotřebovávají informace a také datový vlastníci, musí být zapojeni do datové analýzy a musí být dobře obeznámeni s pravidly mapování zdrojových dat.

Od té chvíle, kdy se vlastníkům dat vytvoří data a dojde k ustanovení business pravidel a politik, jsou přímo zodpovědní uživatelům dat, kteří potřebují tato data využívat. Pokud tito uživatelé na základě špatných (nekvalitních) dat založí svá rozhodnutí, která povedou k finančním či jiným ztrátám, musí za to nést vlastníci dat zodpovědnost.

Výzvou pro IT a sponzora business intelligence projektu je vynucení nevyhnutelných úloh datové archeologie a čištění dat, aby byla dosažena požadovaná kvalita výsledného business intelligence řešení. Ačkoli jsou nástroje, které mohou být nápomocné při datové archeologii a čištění dat, jsou tyto procesy spíše manuální záležitosti. Automatické nástroje tedy neeliminují manuální práci, ale mohou ji redukovat.

2.3.2 Určení zdrojových dat

Nízká kvalita dat je pro většinu firem nepřekonatelným problémem protože nejsou schopni opravit všechny nesrovnalosti. Při výběru zdrojových dat pro aplikace business intelligence je dobré zvážit následujících pět kroků:

- Identifikace potřebných dat: Čištění dat je společné úsilí business analytiků, kteří jsou dobře seznámeni se sémantickým významem dat a analytiků zabývajících se datovou kvalitou.
- Analýza obsahu dat: Mnoho organizací nashromáždí obrovské množství dat v souborech a v databázích. Tato data představují případný zlatý důl cenných znalostí a je potenciálním zdrojem dat pro data mining. Nicméně kvalita datového obsahu musí být posouzena jako první, jelikož data mining na špatných datech nepřináší žádnou nebo jen malou hodnotu.
- Výběr dat pro business intelligence: Určení, která data zahrnout do business intelligence aplikací. Vybrána jsou pouze ta data, která jsou nejdůležitější pro core business. Některé otázky, které je dobré zvážit při výběru dat, jsou vypsány níže.
 - Jsou data dostatečně kvalitní pro rozhodovací účely?
 - Pokud ne, lze tato data alespoň částečně očistit? Vím jak?
 - Jsou nekvalitní data důvodem pro zavedení business intelligence? Je tudíž čištění dat nezbytné?
 - Kolik úsilí zabere vyčištění dat? Kolik to bude stát?
 - Jaký je přínos čištění dat na rozdíl od použití současných dat, která jsou částečně nekvalitní?
 - Jaké mají představu o datové kvalitě uživatelé?
- Příprava čištění dat: Zástupci businessu musí spolupracovat se zaměstnanci IT a vysvětlit jim, business pravidla, na základě kterých bude napsána specifikace čištění dat. V podstatě jde o proces reengineeringu zdrojových dat.
- Vybrání vhodných nástrojů: Stanovení zda je vhodné a nákladově efektivní pořízení ETL nástrojů a nástrojů pro čištění dat. Dále pak je dobré přezkoumat vhodnost a efektivnost těchto nástrojů. Některé nástroje, zejména pro čištění dat, jsou velmi komplikované. Je potřeba si být jist, že práci s nimi zvládnou.

2.4 Proč je datová analýza důležitá

Business manažeři, IT manažeři a IT technici se často nechtějí zabývat datovou analýzou, která zahrnuje logické modelování, datová archeologie a čištění dat. Tyto aktivity vidí spíše jako plýtvání časem. Posuzují úspěch business intelligence řešení spíše podle rychlosti zavedení, než podle kvality zpracování. Výsledkem je, že firmy často vytvoří rádobu datová tržiště, která naplní nekvalitními daty, stejnými, která mají ve zdrojových souborech a databázích, čímž kopírují všechna existující nekvalitní data do nového business intelligence řešení.

Datová analýza je jedním z nejvíce kritických kroků. Proto také v této práci, zabírá velkou část. Aktivita spojené s analýzou dat zaměřených na business, nutí uživatele a datové vlastníky k upravení pohledu na celou firmu a k vyčištění chaos v datech a to ne pouze v business intelligence, ale i ve zdrojových systémech. Zde popsané postupy, slouží ke zlepšení schopnosti udělat kvalifikované rozhodnutí zakládající se na správných datech. Bez provedení těchto kroků, dochází pouze k výstavbě dalšího tradičního nesprávného řešení pro podporu rozhodování a ne k business intelligence řešení.

3 Návrh řešení business intelligence

Tato kapitola se zabývá komponentami business intelligence řešení. Rozebírá jak jejich funkci a popis, tak principy jejich návrhu a konstrukcí.

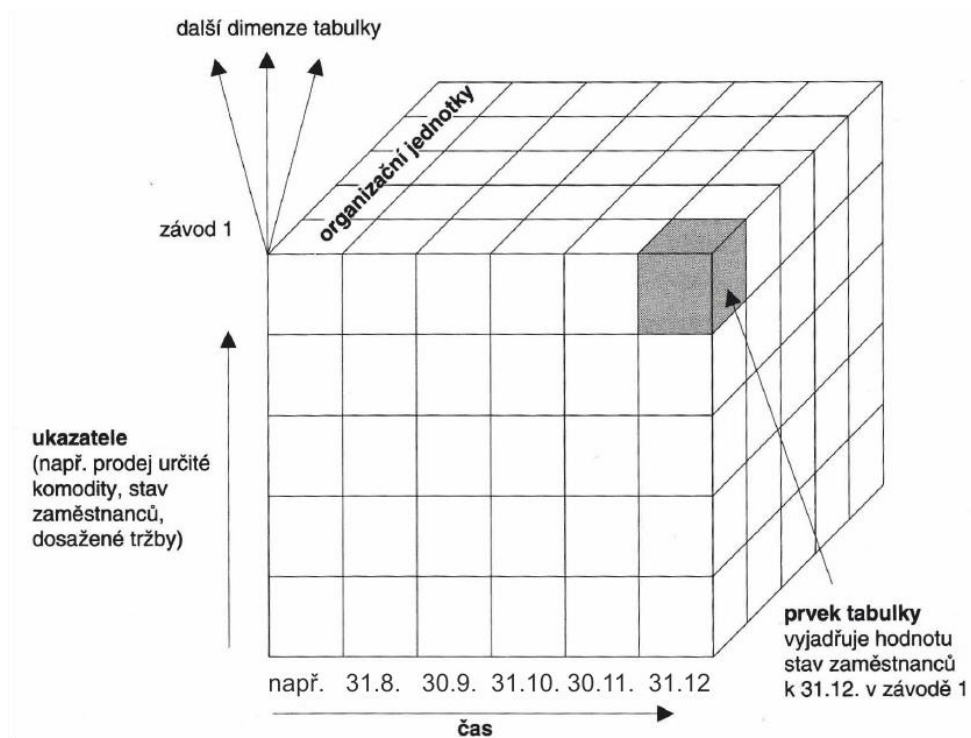
Jelikož je zaváděno business intelligence řešení, které je založeno na multidimenzionalitě, je nutné také rozebrat některé s tím spojené výrazy. Například rozdíl mezi OLTP – On Line Transaction Processing a OLAP – On Line Analytical Processing. První zmíněná technologie tedy OLTP, pracuje s operativními informacemi, které slouží pro realizaci obchodních a dalších transakcí v podniku. Uloženy jsou především v relačních databázích a k jejich aktualizaci může docházet i mnohokrát denně. Ve vztahu k business intelligence lze označit OLTP systémy jako systémy, které poskytují zdrojová data pro datové sklady či tržiště. Oproti tomu OLAP systémy pracují s analytickými informacemi. Tato technologie je založena především na multidimenzionální databázi. Klíčovým principem jsou dva typy tabulek a to faktové tabulky a dimenzionální tabulky. Faktové tabulky slouží pro agregaci dat, zatímco dimenzionální určují, z jakého pohledu (dimenze) bude na data nahlíženo. Nejstarší a nejrozšířenější funkcionalitou OLAP technologie je pokrytí požadavků na reporting. Základní rozdíly mezi OLTP a OLAP technologiemi jsou zaneseny do tabulky 1. [5, str. 20-21]

Tabulka 1: Porovnání OLTP a OLAP technologií

| | OLTP | OLAP |
|-------------------------------|--|---|
| Zdrojová data | Operační data; prvotní zdrojová data. | Konsolidovaná data; vychází z OLTP databází. |
| Účel dat | Řízení a provoz elementárních business úloh | Podpora při plánování, rozhodování a řešení problémů. |
| Jaká data | Odhaluje stav aktuálních procesů. | Multidimenzionální pohled na různé typy business aktivit |
| Vkládání a aktualizace | Krátké a rychlé, vyvoláno koncovým uživatele. | Periodické dlouho trvající dávkové úkoly. |
| Dotazy | Relativně jednoduché dotazy. | Často komplexní dotazy zahrnující agregace. |
| Rychlost | Typicky velmi rychlé. | Ovlivněno objemem dat; komplexní dotazy mohou trvat i mnoho hodin. |
| Požadavky na kapacitu | Relativně malé, pokud jsou historická data archivována | Velké v důsledku existence agregačních struktur a historických dat. |
| Návrh databáze | Velká míra normalizace s mnoha tabulkami. | Typicky nenormalizované s méně tabulkami; používání schémat vločky a hvězdy. |
| Záloha a obnovení | Operační data jsou pro business kritická. Ztráta dat vede k finanční ztrátě. | Místo standardních záloh, mohou některá odvětví zvážit možnost jednoduchého znovunačtení OLTP dat jako metodu obnovení. |

Zdroj: <http://datawarehouse4u.info/OLTP-vs-OLAP.html>

Princip multidimenzionální databáze lze dobře ilustrovat na obrázku 1, který představuje n-dimenzionální kostku obsahující podniková data. Hrany kostky představují dimenze a jednotlivé průniky souřadnic (zde malé kostky) představují konkrétní hodnotu. V tomto případě tedy kostku tvoří dimenze ukazatele, času a závodu. Prvek tabulky (zvýrazněná malá kostka), je konkrétní hodnotu, tedy počet zaměstnanců ke dni 31.12. v závodě číslo 1. Tato hodnota je před vypočítaná, dotazování je tedy pouze na její umístění a tím pádem je rychlost odezvy velká a výsledek je k dispozici prakticky ihned. Prvky dimenzí jsou většinou rozděleny do skupin prvků, podskupin až na jednotlivé prvky. Tvoří tedy hierarchickou strukturu. Business intelligence řešení zajišťuje automatické agregace hodnot podle těchto hierarchických struktur. [5, str. 21-22]



Obrázek 1: Princip multidimenzionální databáze.

Zdroj: [5, str. 22]

3.1 Ukládání dat

Stejně jako motor potřebuje k chodu palivo, reportovací nástroje potřebují ke své funkci data. Bez dat nemají tyto nástroje žádnou cenu. Je tedy potřeba zajistit těmto nástrojům přístup k datům, která jsou shromažďována v produkční databázi. Technicky je možné spojit tyto reportovací nástroje přímo s produkční databází, nicméně tento přístup navzdory své jednoduchosti, není pro většinu organizací vhodný a to hned z několika důvodů. Ty nejdůležitější jsou popsány níže.

- **Datová integrita:** Data potřebná pro reporting můžou být (a častou jsou) uložena v různých produkčních databázích. Například všechna data spojená se zákazníkem

mohou být rozšířena přes více systémů. To znamená, že by reportovací nástroj potřeboval přístup do všech těchto systémů a byl by zodpovědný za integrování všech dat. Celý tento integrační proces dělá tvorbu reportů více komplexní a integrační logika je opakována v mnoha reportech. V případě chyby to může vést k nekonzistentním výsledkům.

- Špatná (nekvalitní) data: Data uložená v produkčních systémech mohou být špatná nebo dokonce některé hodnoty mohou chybět. Pokud by byly reporty tvořeny přímo z těchto dat, museli by se vypořádat s těmito skutečnostmi. To by mohlo být složité, jelikož většina těchto nástrojů nepodporuje správné funkce, které by tento nedostatek odstranily. I kdyby to bylo možné, neexistuje žádná garance toho, že všechny různé reportovací nástroje transformují špatná data stejně.
- Konzistence dat: Pokud různé reportovací nástroje používají různé techniky integrace dat z rozličných produkčních databází, kdo může garantovat, že je použita stejná integrační logika. Jinými slovy, jak možné garantovat konzistenci reportů? Mohlo by se stát, že by ve dvou reportech mohly být použity trochu rozdílné transformace a tím zkreslené výsledky.
- Historická data: Ne všechny produkční databáze sledují historii. V mnoha systémech se stane například to, že pokud dojde ke změně adresy zákazníka, tak je stará adresa zkrátka přepsána tou novou. Některé reporty však potřebují historická data. Například pro různé predikce.
- Interference: Dotazy vyvolané reportovacími nástroji mohou způsobit v produkční databázi přílišnou interferenci. To může mít za následek zvýšení odezvy pro ostatní uživatele, například pro někoho, kdo vkládá nová data.
- Výkonnost dotazu: Produkční databáze je primárně učená pro podporu produkčních aplikací. Dotazy vyvolané reportovacími nástroji mohou být poměrně komplexní. Spuštěním těchto dotazů v produkční databázi může způsobit velmi malou výkonnost. Ve skutečnosti může být výkon databáze omezen na tolik, že uživatelé budou muset čekat i několik hodin než dostanou výsledky. Mezitím tento dotaz vyprodukuje mnoho inference.
- Externí data: V tomto typu řešení není možnost jak analyzovat externí data přímo. Pokud je například, potřeba vytvořit report, který analyzuje vliv počasí na prodejnost konkrétního zboží. Pro většinu organizací jsou data spojená s počasím čerpána z externího zdroje dat a ne z vlastní produkční databáze.

Aby bylo předejito většině těchto problémů, je business intelligence vyvíjeno na základě datového skladu či datových tržišť. [1, str. 30-32]

3.1.1 Datový sklad

Datový sklad je oddělené datové uložistiště, speciálně navržené pro reporting analytické operace. Data jsou do něj periodicky kopírována z produkčních databází (zdrojová data). Nejčastěji k tomu dochází skrze ETL procesy. Jedna z definice datového skladu podle [7, str. 31] je volně přeložena takto: Datový sklad je předmětově orientovaný, integrovaný, stálý a časově

rozlišený souhrn dat, který slouží pro podporu manažerských rozhodování. Tyto základní vlastnosti datového skladu jsou blíže ozřejmena níže.

- **Předmětově orientovaný:** Data jsou rozdělena podle jejich typu, ne podle zdrojových aplikací. Například tedy data o zaměstnancích jsou uložena pouze v jedné databázi, nikoli jak je tomu v produkčních systému, kde jsou tato data v různých souborech, podle toho, která aplikace je využívá.
- **Integrovaný:** Data jsou ukládána v rámci celého podniku, nikoli v rámci jednotlivých oddělení.
- **Stálý:** Datový sklad uplatňuje princip „Read Only“ – tedy pouze pro čtení. Znamená to, že data nejsou ručně měněna ani mazána.
- **Časově rozlišený:** Do datového skladu je uložena i historie dat. Díky tomu je možné provádět reporty za konkrétní časová období. [5, str. 32]

Pokud budou znovu zváženy nevýhody reportingu na základě přímého spojení s produkční databází, datový sklad přichází s následujícími:

- **Datová integrita:** Data přicházející z různých systémů jsou integrována a uložena jedním integrovaným způsobem. Reportovací nástroje se tudíž už nemusí datovou integritou zabývat.
- **Špatná (nekvalitní) data:** Za použití ETL procesu mohou být data očištěna, což vede k tomu, že v datovém skladě jsou již data čistější (kvalitnější). V důsledku toho neobsahují reporty špatná data a všechny reporty sdílejí výsledky stejných čistících operací.
- **Konzistence dat:** Díky tomu, že všechny reporty obsahují data z datového skladu, ve kterém jsou integrovaná data, je tedy mnohem menší pravděpodobnost výskytu nekonzistentních dat v reportu.
- **Historická data:** Dokonce pokud produkční databáze nezachycuje historii, datový sklad může být navržen tak, aby historii evidoval. Datový sklad se stává místem pro uložení všech starých (historických) dat.
- **Interference:** Dotazy generované reportovacími nástroji neběží na produkčních databázích přímo, čímž nevzniká interference. Avšak data jsou kopírována z produkčních databází do datového skladu periodicky, při čemž může docházet k interferenci. Nicméně tyto periodické akce jsou většinou plánované přes noc, kdy nejsou produkční databáze vytěžovány.
- **Výkonnost dotazu:** Datový sklad může být navržen a optimalizován speciálně pouze pro reportingové dotazy. To samozřejmě zlepšuje výkonnost dotazů.
- **Externí data:** Pokud jsou potřeba pro reporting data z externích zdrojů, není problém je uložit v datovém skladu, čímž budou stejně přístupná jako interní data. [1, str. 32-34]

3.1.2 Datové tržiště

Pokud představuje datový sklad opravdu rozsáhlé uložení dat a všechny reportovací nástroje přistupují k tomuto jedinému datovému uložení, mohou nastat komplikace v podobě

přetížení datového skladu a tím pádem bude pomalá odezva. Z toho důvodu mnoho firem implementuje datová tržiště, která odkloní vytížení dotazy.

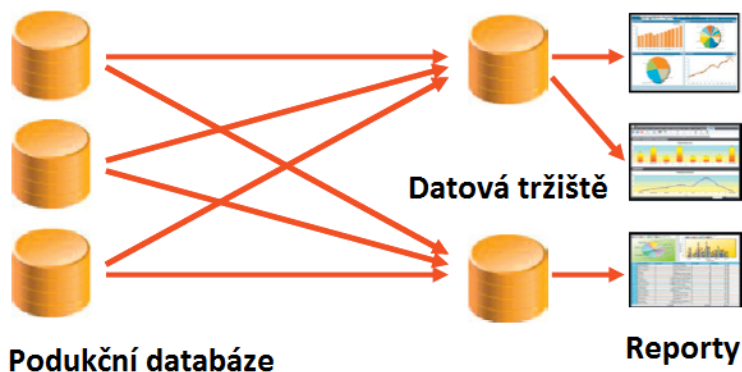
Každé datové tržiště je vytvořeno pro specifickou skupinu uživatelů. Obsahuje jakousi podmnožinu dat, která konkrétní skupina uživatelů potřebuje. Často také datové tržiště obsahuje o něco více agregovaná data než datový sklad.

Existují dva typy architektury s datovými tržišti a to s použitím datového skladu a bez použití datového skladu. Výhodou použití architektury bez datového skladu je především rychlost vývoje. Jistě totiž zabere méně času vytvoření datového tržiště, které bude sloužit pro skupinu lidí než komplexní datový sklad. Nevýhody tohoto řešení jsou podobné jako u spojení produkčních databází přímo s reportovacími nástroji, tedy integrita dat, nekvalitní data a konzistence dat. [1, str. 34-35]



Obrázek 2: Architektura s datovým skladem.

Zdroj: [1, str. 34]



Obrázek 3: Architektura bez datového skladu.

Zdroj: [1, str. 35]

3.1.3 Dočasné uložisko dat

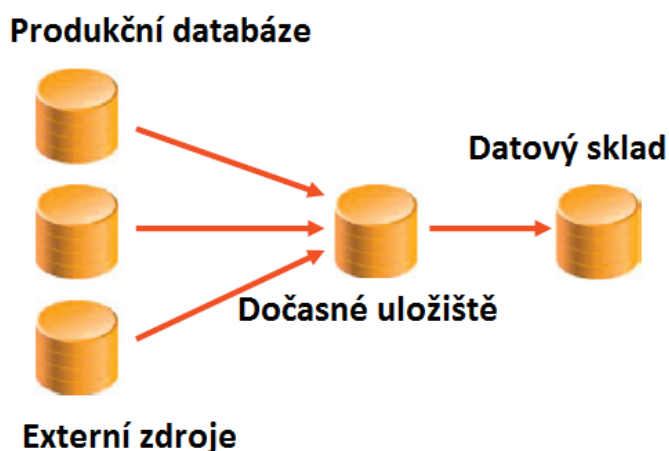
Z technických a konceptuálních důvodů mohou nastat situace, kdy je kopírování dat z produkčních databází do datového skladu příliš složité. Proto je v mnoha business intelligence řešeních implementováno takzvané dočasné uložisko dat. Na toto místo jsou některá data z produkčních databází kopírována ještě před tím, než budou uložena do datového skladu. Při kopírování dat do dočasného uložiska, mohou být provedeny některé změny týkající se obsahu

nebo struktury dat. V ideálním případě je struktura tabulek v dočasném uložišti stejná jako produkční databázi. To znamená, že během ETL procesu nedošlo k žádným změnám v datech a ta byla pouze zkopírována jedna ku jedné. Všechny potřebné transformace a čistící operace budou provedeny v druhém kroku kopírování dat a to z dočasného uložště dat do datového skladu. Po úspěšném nakopírování dat do datového skladu, mohou být všechna data z dočasného uložště smazána. Díky tomu je velikost dočasného uložště mnohem menší než velikost produkční databáze a datového skladu. Dočasné uložště dat je tedy dočasné a přechodné uložště dat obsahující data z produkčních systémů.

Existují různé důvody, proč je vhodné použít v řešení business intelligence toto dočasné uložště dat a mezi ně patří:

- Data z produkčních databází jsou podrobena mnoha procesům před tím, než jsou nahrána do datového skladu. Pokud jsou data kopírována přímo z produkční databáze do datového skladu, všechny tyto procesy vedou k vážným interferencím. Dobré řešení je tedy zatěžovat produkční systémy co nejméně a udělat pouze kopii jedna ku jedné a veškeré transformace provést až na po té.
- Dočasné uložště dat může být také použito ke sledování pohybu dat, která nemusí být potřebná v datovém skladě. To znamená, že sice je požadováno sledování pohybu dat, ale ještě přesně se neví, na co tato data budou využita. Na druhou stranu se předpokládá, že by tato data mohla být v budoucnu použita. Nakopírováním dat do dočasného uložště je garance toho, že se už neztratí.

Jakmile jsou data z dočasného uložště transformována a nakopírována do datového skladu, mohou být smazána. Ale mohou nastat i důvody proč je zachovat. V tom případě se jedná o tak zvané trvalé datové uložště. Uchování všech dat v tomto uložšti může být využito, když jsou data znovu načtena do datového skladu. Dočasné uložště dat může být použito i v případě, kdy je business intelligence architektura založena na modelu bez datového skladu, který zastupují datová tržiště. V tomto případě jsou datová tržiště plněna novými daty z dočasného uložště. [1, str. 35-36]



Obrázek 4: Architektura s dočasným uložštěm dat.
Zdroj: [1, str. 36]

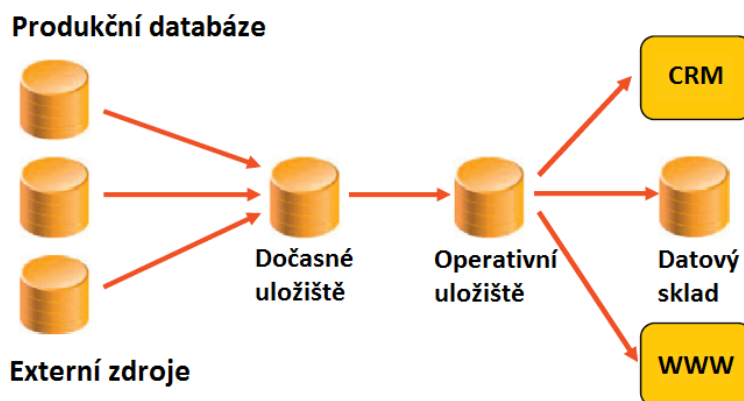
3.1.4 Operativní datové uložště

Ať už data přichází z produkčních systémů nebo z dočasného uložště, musí být transformována, než se nahrají do datového skladu. Mohou však existovat IT systémy, které využijí integrovaná a transformovaná data. Například firemní webové stránky nebo třeba CRM systém mohou vyžadovat přístup k těmto datům. Pokud jsou tedy požadavky na integrovaná data i z jiných systému ICT, může být užitečným řešením použití operativního datového uložště. To poskytuje integrovaný pohled na operativní data, která jsou dostupná v produkčních systémech.

Operativní uložště dat, je stejně jako datový sklad předmětně orientované, integrované, ale oproti datovému skladu nestálé a časově nerozlišené. Jakmile jsou data změněna (přidána, smazána, aktualizována) v produkčních databázích, tak tato skutečnost je reflektována do operativního uložště co nejrychleji je to možné. Tím pádem by operativní uložště mělo být schopno prezentovat relativně aktuální data.

Data mohou být do operativní uložště nahrávána buď přímo z produkčních databází nebo z dočasného uložště dat. V některých případech jsou operativní uložště a dočasné uložště sloučeny do jedné komponenty. Ve většině případů to znamená, že operativní uložště přebírá funkce dočasného uložště.

Operativní uložště je volitelná komponenta, která však přináší několik výhod. Největší z nich je fakt, že data v něm již podstoupila proces integrace a transformace, čímž je výrazně zjednodušen proces nahrávání dat do datového skladu. [1, str. 37]



Obrázek 5: Architektura s operativním uložštěm dat.
Zdroj: [1, str. 37]

3.1.5 Porovnání vlastností datových uložšť

Následující tabulka poskytuje rychlý přehled vlastností výše zmíněných datových uložšť.

Tabulka 2: Porovnání datových uložišť.

| | Data | | | | | |
|---------------------|---------|----------|-------------|----------------|------------|-------------|
| | Nestálá | Detailní | Integrovaná | Časově rozliš. | Agregovaná | Celopodnik. |
| Produkční databáze | ✓ | ✓ | | | | |
| Datový sklad | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Datové tržiště | | ✓ | ✓ | ✓ | ✓ | |
| Dočasné uložisko | ✓ | ✓ | | | | |
| Operativní uložisko | ✓ | ✓ | ✓ | | | ✓ |

Zdroj:[1, str. 39]

3.2 ETL – Extraction, Transformation and Loading

ETL je zkratka slov Extraction, Transformation and Loading. Český překlad zní vytažení, transformace (změna) a načítání. Časté označením pro ETL je také pojem datová pumpa. Funkcí datové pumpy je vytažení dat z produkčních systémů, změna dat do požadované podoby a očištění a dále načtení těchto dat do některého z business intelligence datového uložiska. ETL pracuje v dávkovém režimu, což znamená, že neběží neustále, ale pouze v určité periodě. Spuštění je vhodné naplánovat na dobu, kdy nejsou produkční systémy vytíženy, například pozdní večerní nebo brzké ranní hodiny. V tuto dobu se pumpy spustí a načtou data do datového uložiska, dle zvolené architektury. [5, str. 29]

Obsahem návrhu datových pump je především:

- Navržení algoritmů pro transformaci zdrojových dat do datových uložisk.
- Integrace dat.
- Kontrola načtených dat, zdali nedošlo k chybám a data nebyla zkreslena.
- Opravy nekonzistentních dat.
- Definování datových standardů.

Nejčastějšími problémy, se kterými se během procesu ETL lze setkat, jsou převážně záležitosti kvality zdrojových dat. Problémy, které jsou nejčastější, jsou popsány níže.

- Jelikož jsou data čerpána z rozličných zdrojů, nejsou sjednocené jejich číselníky.
- Data z produkčních systémů nebývají uložena v souladu s relačními zásadami a to zejména:
 - Absence primárních klíčů.
 - Některé atributy nemají vyplněné hodnoty a na místo nich, jsou pouze NULL hodnoty.
 - Jeden sloupec v tabulce obsahuje více atributů.
- Problémy při konverzi datových typů.

Pokud při návrhu business intelligence řešení byly použity dočasné uložení a/nebo operativní uložení dat a také datová tržiště, potom se datové pumpy většinou konstruují ve třech základních etapách, které jsou:

- 1) Kopírování dat z produkčních databází do dočasného nebo operativního uložení dat. Data jsou kopírována tak jak jsou, tedy jedna ku jedné.
- 2) Transformace a nahrání dat z dočasného nebo operativního uložení do datového skladu. Data jsou převedeny do provázaných tabulek s respektovanou referenční i doménovou integritou.
- 3) Transformace a distribuce dat z datového skladu do jednotlivých datových tržišť. Data jsou agregována pro další náročnější analýzy a reporting.

Dále je pak při implementaci vhodné definovat:

- U tabulek dimenzí:
 - Zohlednění změn v dimenzích.
 - Zaznamenání historie provedených změn.
 - Normalizace dat dle principů schématu sněhové vločky.
 - Denormalizace dle principů schématu hvězdy.
- U tabulek faktů:
 - Zajištění přírůstkového doplňování časových snímků dat
 - Zajištění referenční integrity s dimenzemi
 - Pokud je třeba, tak zajištění denormalizace.
- Definování primárních klíčů ve všech tabulkách. V případě, že budou již primární klíče definovány z produkčních systémů, je nutné předpokládat, že se zde mohou vyskytovat duplicity nebo například NULL hodnoty.
- V rámci řešení doménové integrity jsou stanoveny povinnosti a nepovinnosti atributů, datové typy a implicitní hodnoty pro povinné atributy.
- V rámci řešení referenční integrity jsou zjištěny porušené nebo neexistující vazby nadřazených a podřazených záznamů.
- Řešení historických dat a historických číselníků je podmíněné generováním vlastního identifikátoru pro každý číselníkový záznam a dále provedením záměny původních cizích klíčů za vytvořené vlastní identifikátory. [5, str. 142-144]

4 Dashboard

V této kapitole se dostávám již k hlavním výstupům business intelligence řešení a to jsou dashboards. V dnešní době firmy zaznamenávají enormní nárůst dat. S tím souvisí potřeba tato data využívat a vizualizovat. Jedním z nástrojů, který obstarává tuto funkci je informační dashboard. Jedná se o zobrazení nejdůležitějších informací na jedné obrazovce, které lidé potřebují ke své práci. Jde o způsob prezentace, který jim umožňuje monitorování co se právě děje. Informační dashboards tedy představují další způsob komunikace. Ale pouze za předpokladu, že jsou správně navrženy. Největším problémem nebývají technologická omezení, ale spíše špatný design. Aby byl využit plný potenciál dashboardu, musí tento dashboard zobrazit mnoho informací na relativně malém prostoru. To vyžaduje navrhnout dashboard, tak aby využíval sílu vizuálního vnímání a byl schopen zpracovat velké množství informací. [8, str. 6]

4.1 Charakteristika

Nyní je důkladněji vysvětleno co to vlastně dashboard je. Volně přeložená definice zní takto: Dashboard je vizuální zobrazení nejdůležitějších informací potřebných k dosažení jednoho či více cílů; konsolidovaný a uspořádaný na jedné obrazovce, aby mohly být informace monitorovány na první pohled.

Anglické slovo dashboard zde záměrně nepřekládám, jelikož jeho doslovný český překlad zní přístrojová deska, což v souvislosti s business intelligence není vhodné. Nicméně stejně jako přístrojová deska u automobilu zobrazuje kritické informace potřebné k ovládání vozidla, náš dashboard v business intelligence má prakticky stejnou úlohu.

Informace v dashboardu jsou prezentovány vizuálně, obvykle jako kombinace textu a grafiky, avšak s důrazem na grafiku. Není to z důvodu, že grafické prvky jsou zkrátka roztomilé, ale protože grafické zobrazení, často komunikuje s větší účinností a bohatším významem než text samotný. Klíčem k tomu, aby lidské oko mohlo rychle zachytit informace a aby z nich lidský mozek mohl správně a jednoduše zjistit jejich význam, jsou následující vlastnosti dashboardů:

- Dashboard zobrazuje informace potřebné k dosažení specifických cílů: K dosažení dokonce i jediného cíle, je často potřebný přístup k různým sestavám informací, které spolu jinak nesouvisí. Proto jsou také často získávány z rozdílných zdrojů.
- Dashboard vyplňuje jednu obrazovku (počítačový monitor): Informace musí být zobrazeny pouze na jedné obrazovce, aby byly okamžitě dostupné a viditelné na první pohled. Pokud je nutné vertikálně či horizontálně listovat nebo v případě, že je nezbytné přepínat mezi obrazovkami, aby byly viditelné všechny informace, je vhodné vytvořit více dashboardů a informace mezi ně rozdělit. Cílem je mít nejdůležitější informace snadno a bez námahy k dispozici, aby mohlo být rychle vstřebáno to, co je potřeba vědět.

- Dashboard zobrazuje informace na první pohled: Navzdory skutečnosti, že téměř jakékoli informace mohou být vhodně zobrazeny v dashboardu, je zde při nejmenším jedna věc, kterou vyjádřit nelze a tou jsou detailní data. Není to z důvodu, že by to nebylo technicky možné, ale z důvodu, že by nebyla zachována hlavní přednost dashboardu a to přehlednost. Dashboard musí být schopen rychle poukázat na to, že nějaký ukazatel vyžaduje pozornost. Na základě toho upozornění si pak uživatel najde detailnější informace například proklikáním mimo dashboard.
- Musí být informace zobrazeny přes webový prohlížeč? Vzhledem k univerzálnosti přes všechny platformy je webový prohlížeč vynikajícím nástrojem pro zobrazení dashboardu. Nicméně není to jediné možné médium. [8, str. 26-27]

4.2 Základní chyby návrhu dashboardu

Nyní následuje přehled základních chyb, kterých se často firmy při návrhu dashboardů dopouštějí. Tato podkapitola čerpá z [8, str. 39-62]

Pevně věřím, že zmíněním těchto chyb, je zamezeno jejich opakování v tomto řešení a tím pádem budou dashboardy vypadat tak, jak mají.

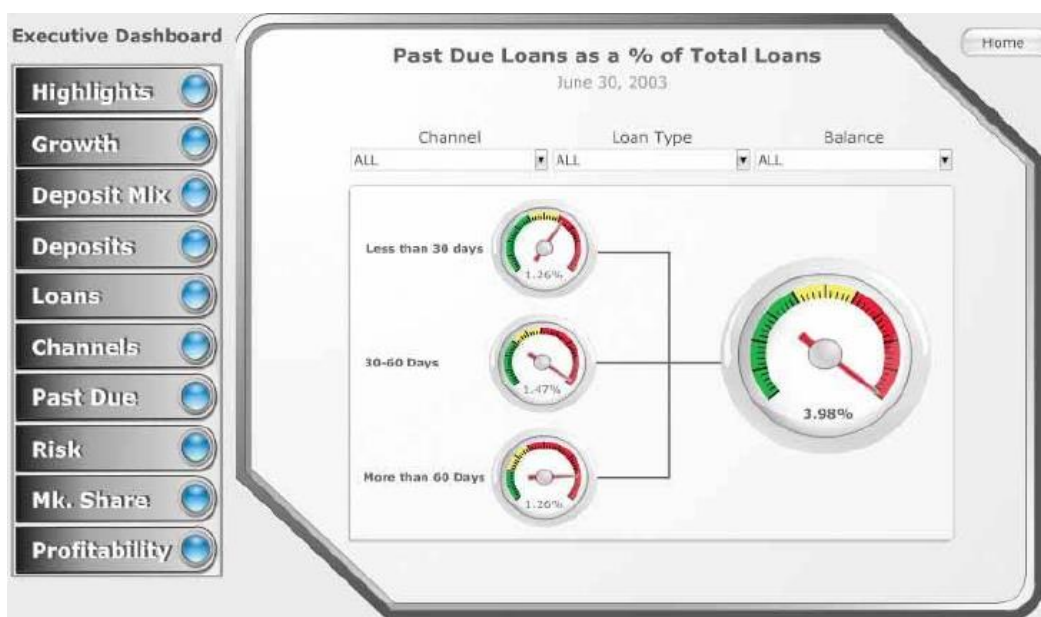
- Přesahování hranic jedné obrazovky

Ač je zásada o zobrazení dashboardu v tomto textu omílána již po několikáté, přesto je třeba ji zde zmínit v rámci připomenutí. Zároveň se jedná o častou chybu při návrhu dashboardu. Částečně tento problém způsobuje fakt, že člověk je schopen udržet v krátkodobé paměti pouze několik prvků. Jednou z největších výhod dashboardu jakožto komunikačního média je, že je souběžně viditelné vše, co je potřeba. To umožňuje srovnání, která vedou k postřehům, které jinak nemusí nastat. Přesáhnutí hranice jedné obrazovky tento benefit vyloučí.

- Členění dat do oddělených obrazovek

Informace, které jsou vizualizovány v dashboardu jsou často rozděleny dvěma způsoby. Jedním z nich je rozdělení do více oddělených obrazovek, mezi nimiž je přepínáno a druhým způsobem je rozdělení do různých instancí v rámci jedné obrazovky, které jsou přístupné například skrz menu.

Rozdělení dat, která mohou být zobrazena společně, je chyba. Jak je vidět na obrázku 6, jednotlivé prvky dashboardu jsou rozděleny, a pro jejich zobrazení musí uživatel překlíkat v navigačním menu po levé straně. Je tedy ochuzen o celkový pohled, který by umožnil porovnání.



Obrázek 6: Ukázka špatného dashboardu: vše není zobrazeno na první pohled.

Zdroj: [8, str. 40]

- Nutné vertikální či horizontální posouvání

Chyba, při které je celkový dashboard příliš velký než aby se vešel na jednu obrazovku a k pohybu po něm, je nutné používat posuvníky.

- Poskytování neadekvátního kontextu k datům

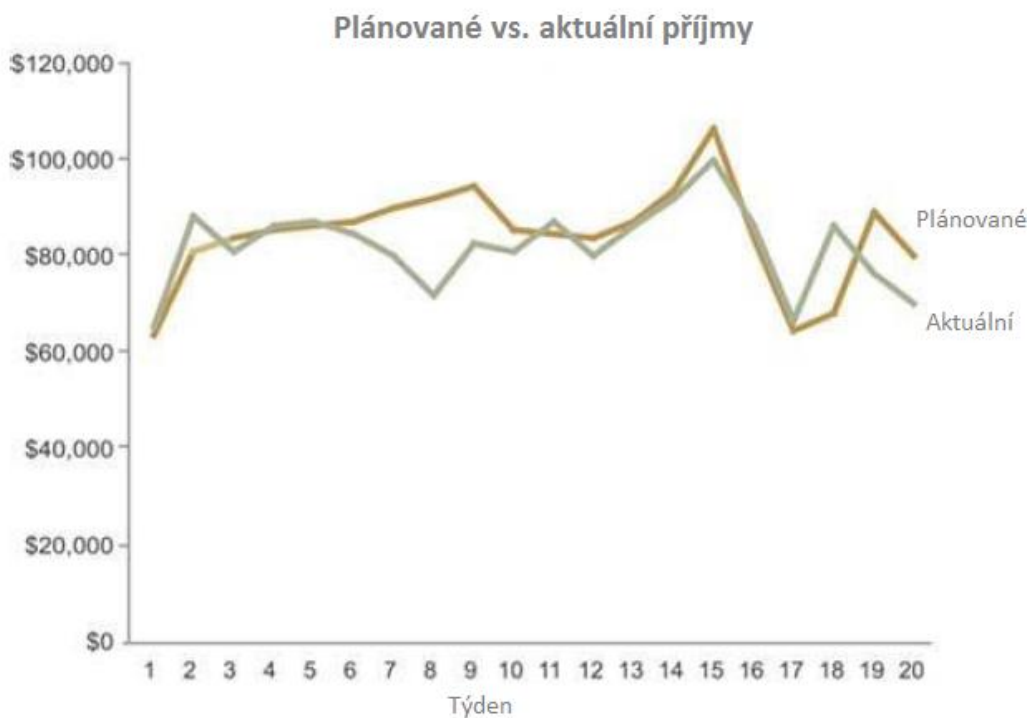
Například pouhá čísla, bez nějaké další dodatečné informace toho na první pohled mnoho neřeknou. Doporučuje se přidat údaj, se kterým se dá hodnota porovnat, například průměrná, minimální a maximální hodnota. Množství kontextu, kterým je podpořen daný ukazatel, záleží na jeho účelu a požadavcích uživatele. Více neznamená v tomto případě vždy lépe. Příliš mnoho kontextu by totiž mohlo vyvolat nepřehlednost.

- Zobrazení příliš mnoho detailů a přesnosti

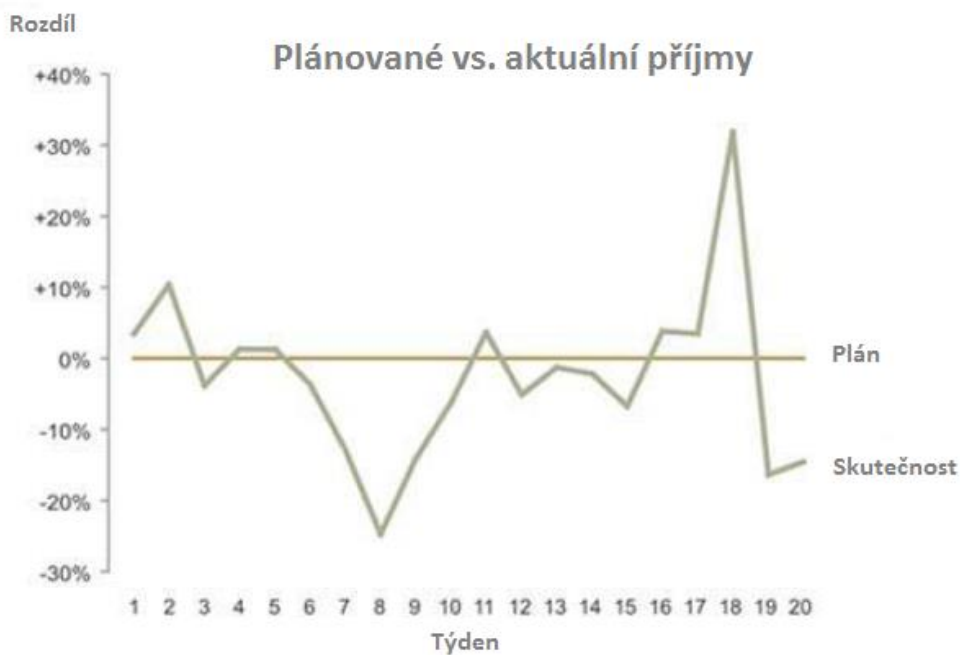
Příliš mnoho detailů nebo ukazatelů, které jsou vyjádřeny příliš přesně, například roční obrát firmy na jednotky a desítky korun, uživatele pouze zpomalují, aniž by jim přinesly nějakou výhodu. Na druhou stranu pokud je vzat v úvahu opačný extrém, kdy jsou data příliš obecná, než aby přinesla přidanou hodnotu, je to také špatně. Platí zde určitě spíše více detailů než málo. Je tedy důležité citlivě posoudit, jakou hloubku detailu zvolit.

- Zvolení nesprávné formy zobrazení ukazatelů

Aby byl ukazatel vypovídající, je potřeba vědět, co je měřeno, v jakých jednotkách a co uživatele zajímá. Například pokud jsou porovnávány plánované příjmy s těmi aktuálními, není úplně vhodné porovnávat konkrétní hodnoty v čase obou veličin tak, jak znázorňuje graf na obrázku 7, ale je přehlednější zvolit například procentuální rozdíl, viz obrázek 8.



Obrázek 7: Ukázka špatného dashboardu: Nepřehledná forma.
Zdroj: [8, str. 46]



Obrázek 8: Ukázka správného dashboardu.
Zdroj: [8, str. 47]

- Volba nevhodného typu zobrazení

Tato chyba je velmi podobná té předcházející, avšak liší se tím, že zde je zvolen špatný typ média, například grafu, zatímco v předcházejícím případě byl špatný i správný typ média stejný, jednalo se však špatnou formou ukazatele.

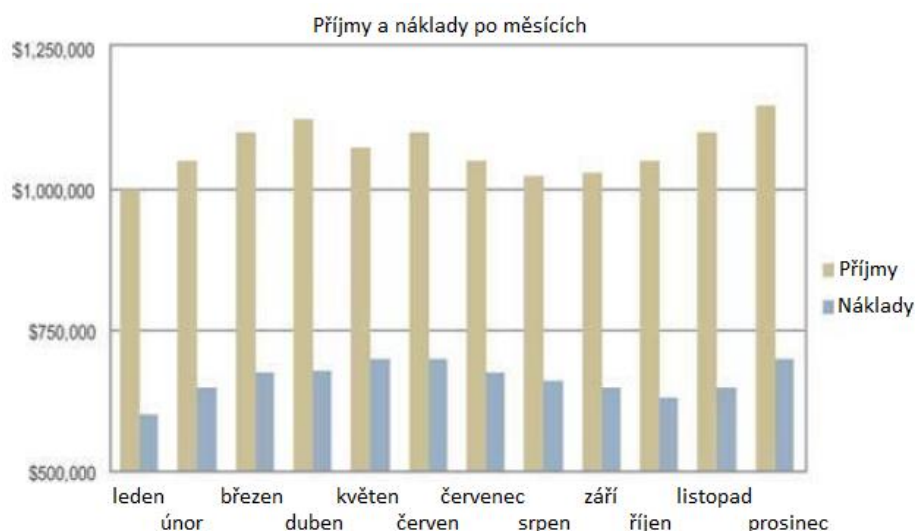
Jedná se o častou chybu nejen v dashboardech, ale ve všech kvantitativních vizuálních prezentacích dat. Například použití grafu v situaci, kdy je tabulka přehlednější a naopak. Například koláčové grafy bývají častým zdrojem problémů. Jsou použitelné prakticky pouze v případě, že je potřeba znázornit části v celku a za podmínky, že části tvoří 100%. Nicméně obecně se nedoporučuje používat koláčové grafy, protože člověk nedokáže porovnávat dvou dimenzionální prostory nebo úhly s velkou přesností. Sloupkové grafy jsou tedy mnohem lepším způsobem, jak zobrazit informace.

- Používání bezvýznamné různorodosti

Lidé často váhají s použitím stejného typu média v dashboardu vícekrát. Je to zřejmě způsobeno tím, že se obávají, aby se uživatel při koukání na dashboard nenudil. Různorodost je sice kořením života, pokud je každý prvek v dashboardu jiný, uživatel trpí. Vždy bychom měli zvolit médium, které je pro danou situaci nejvhodnější i když není v dashboardu žádné jiné médium. Pokud jsou uživateli poskytovány informace, které nutně potřebuje ke své práci, data ho nebudou nudit kvůli tomu, že je vše stejné. Konzistence ve smyslu zobrazení, umožňuje uživateli používat stejnou vjemovou strategii pro interpretaci data a to ušetří jeho čas i energii.

- Neadekvátní zobrazení kvantitativních dat

Někdy jsou grafické prezentace kvantitativních dat mylně navrhovány, ve smyslu, že zobrazují hodnoty nepřesně. Na obrázku 9 je uveden graf, na kterém byla kvantitativní stupnice připadající vertikální ose nesprávně nastavena. Délka sloupců reprezentuje jejich kvantitativní hodnotu. Sloupce v tomto grafu konkrétně vyjadřují příjmy a náklady. Nahlédnutím konkrétně například na sloupce připadající měsíci lednu, je možné si podle délky sloupečků všimnout, že příjmy jsou asi čtyřikrát vyšší než příjmy. Bližším prozkoumáním lze však zjistit, že zdání klame. Příjmy jsou totiž menší než dvojnásobek nákladů. Problém totiž je v tom, že hodnoty začínají na \$500,000 namísto 0, jak by vždy ve sloupcovém grafu měly.



Obrázek 9: Chyba: Hodnoty nezačínají od nuly.

Zdroj: [8, str. 56]

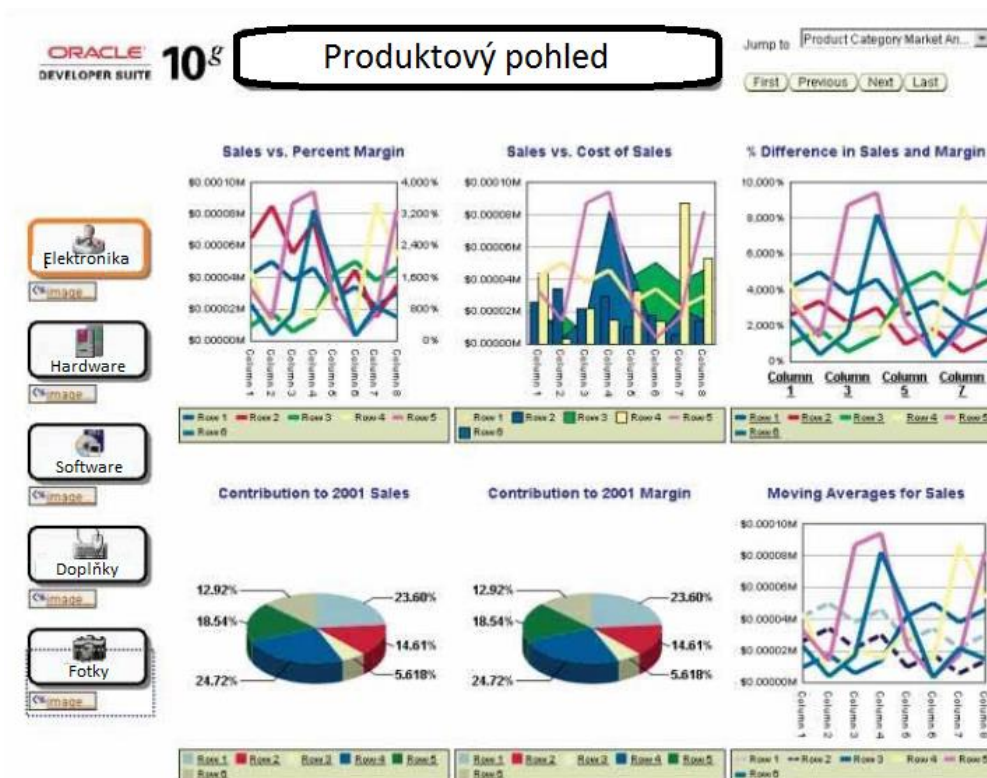
- Špatné aranžmá dat

Dashboards prezentují velké množství informací na omezeném místě. Pokud informace nejsou dobře organizované, jako například vhodné umístění jednotlivých médií. Tato média by měla být v dashboardu rozmístěna podle důležitosti a pokud to je možné, seskupovat související data. Cílem není pouze udělat dashboard tak, aby dobře vypadal, ale uspořádat v něm data takovým způsobem, který je vyhovující. Nejdůležitější data by měla být výrazná. Data, která vyžadují okamžitou pozornost, by měla vyniknout a data, která jsou porovnávána s ostatními, by měla být uspořádána tak, aby bylo porovnání co nejjednodušší.

- Neefektivní nebo žádné zvýraznění důležitých dat

Pohledem na dashboard, by oči měly okamžitě sjet k informacím, které jsou nejdůležitější, dokonce i když nejsou ve vizuálně nejvíce ceněném místě na obrazovce, což je vlevo nahoře.

Problém v dashboardu na obrázku 10 je, že vše je výrazné, a tudíž nic nevystupuje. Dokonce i logo a navigační panel (vlevo) jsou výrazné díky tomu, že mají prominentní umístění a tučné orámování. Vzhledem k tomu, že však nejde o důležitá data, neměly by být takto výrazné. Dále jsou zde grafy, které jsou stejně tučné, stejně barevné a neposkytují žádnou nápovědu, co je důležité a na co se zaměřit. Jistěže vše co má své místo v dashboardu je důležité, ale ne se stejnou vahou, a proto by měly být oči uživatele vždy nasměrovány k nejvíce stěžejním informacím jako první.



Obrázek 10: Chyba: Všechny objekty jsou stejně výrazné.

Zdroj: [8, str. 58]

- Nepřehledná obrazovka s nepotřebnou dekorací

Dalším základním problém v dashboardech bývá přebytek nepotřebné dekorace. Tvůrce si nejspíše myslí, že bude uživatel očarován krásou umění nebo se domnívá, že výzdoba je nezbytná k tomu aby uživatele pobavil. Nicméně i lidé, kterým se z počátku může nějaká dekorace líbit, budou po pár dnech touto dekorací otráveni. Dobrým příkladem je dashboard na obrázku 11, kde je v levé části dashboardu požitá dekorace imitující kroužkovou vazbu. To tam zkrátka nepatří a zbytečně to zabírá drahocenné místo.



Obrázek 11: Chyba: Dashboard obsahuje bezvýznamnou dekoraci.

Zdroj: [8, str. 59]

- Nesprávné používání barev

Již v předchozích případech bylo možné spatřit dashboards, kde bylo příliš mnoho barev nebo byly špatně použity. Důležitou radou tedy je, že by barvy neměly být používány jen tak ledabyle.

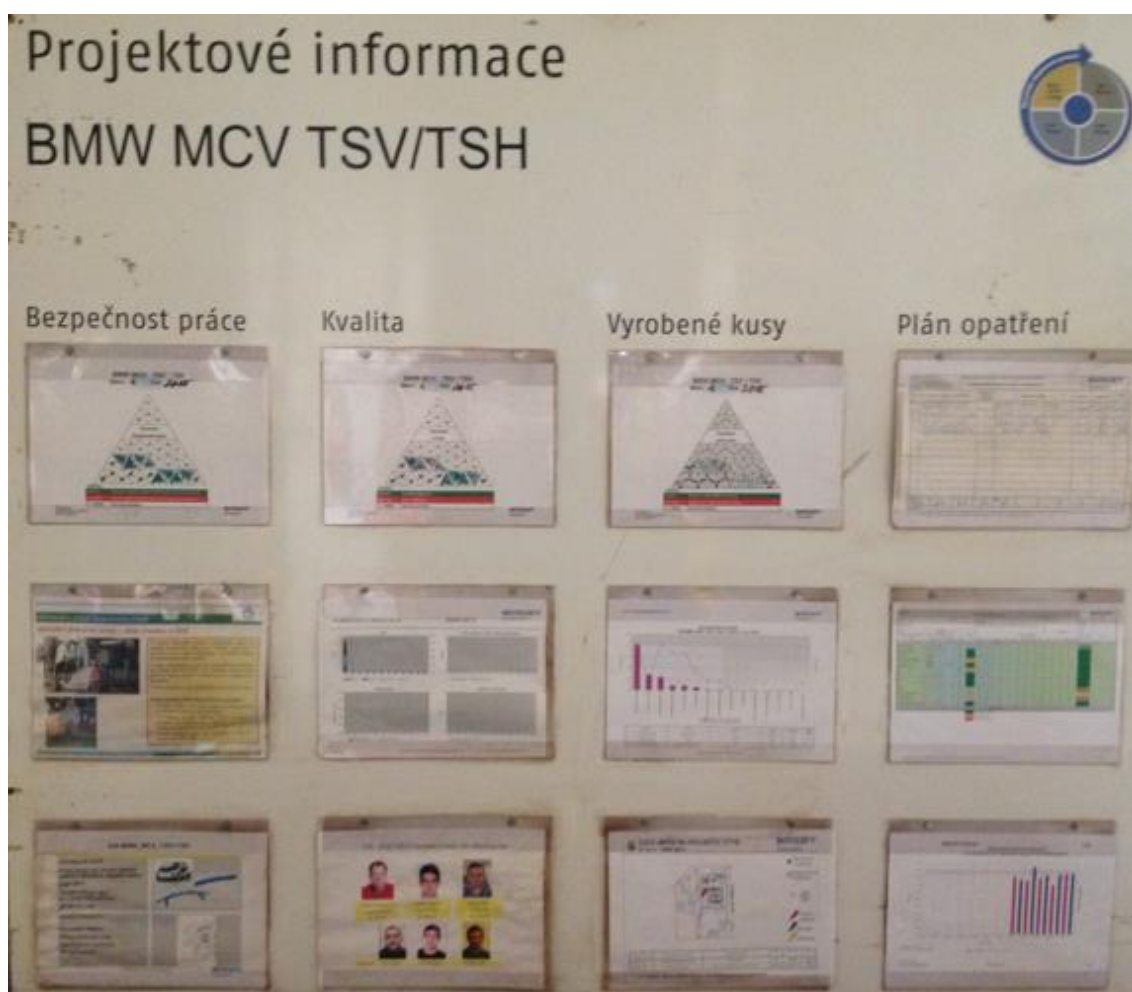
Volba barev by měla být promyšlená, se zvažováním toho, jak jsou barvy a významy barevných rozdílů vnímány. Některé barvy jsou teplé a vyžadují naši pozornost a jiné zase chladnější a méně viditelné. Když se nějaká barva objeví jako kontrast ve vztahu k standardu, naše oči zpozorní a náš mozek se snaží přiřadit význam kontrastu. Pokud jsou barvy ve dvou rozdílných sekcích dashboardu stejné, je uživatel sváděn, vztáhnout tyto stejně barevné objekty k sobě.

Praktická část

5 Analýza požadavků

Prvotní vznik požadavku na automatizaci a vizualizaci projektových nástěnek formuloval zaměstnanec z oddělení štihlé výroby. Docházelo totiž k neshodám, kdo má tyto projektové nástěnky na starost a kdo je má aktualizovat. S tím souvisely i problémy, kdy tyto projektové nástěnky nebyly aktuální a aktualizovali se často jen před auditem. Primárním cílem je tedy vytvořit digitální formu projektových nástěnek, která by byla z velké části aktualizovaná automaticky. Jedná se tedy o jakýsi dashboard. Vedlejším cílem je vytvořit takovou datovou strukturu, která umožňuje další reporting, jako například porovnání projekt mezi jednotlivými roky.

Obrázek 12 znázorňuje, jak taková papírová projektová nástěnka vypadá.



Obrázek 12: Ukázka papírové projektové nástěnky.

Zdroj: [vlastní]

Bližší popis toho, co a kde je na projektové nástěnce zobrazeno, poskytuje obrázek 13.

| Projektové informace BMW MCV TSV/TSH | | | |
|---|------------------|---------------------|-------------------|
| Bezpečnost práce | Kvalita | Vyrobené kusy | Plán opatření |
| Pyramida BOZP | Pyramida kvality | Pyramida kusy | Plán opatření |
| Pracovní úraz | Z-grafy | Analýza poruch | Projectscore card |
| Informace o projektu | Foto týmu | Úzké místo projektu | OEE, HIP |

Obrázek 13: Popis a rozložení prvků v projektové nástěnce.
Zdroj: [vlastní]

- **Pyramid BOZP**
Jako grafické médium je zde pyramida složená z pořadových dnů měsíce. Zaznamenávají se tři hodnoty. Pokud byl v ten den zaznamenán nějaký pracovní úraz, vybarví se políčko červeně. Pokud nebyl zaznamenán pracovní úraz, vybarví se políčko zeleně a pokud v ten den neprobíhala produkce, udělá se křížek.
- **Pyramida kvality**
Je založena na stejném principu jako pyramida BOZP, akorát místo pracovního úrazu eviduje, zdali se ten den vyskytla reklamace.
- **Pyramida kusy**
Opět se jedná o stejný způsob jako v předchozích dvou případech. Zde se zaznamenává, zdali byla v ten den splněna norma či nikoli. Pokud norma byla splněna, políčko je vybarveno zeleně, pokud nebyla, tak červeně. V případě, že se daný den nevyrábělo, je zde křížek.
- **Plán opatření**
Tento prvek se vztahuje k výše zmíněným pyramidám. Pokud je v některé pyramidě červeně vybarvený den, musí být k tomu uveden záznam v plánu opatření. To znamená, že pokud nebyla splněna norma je potřeba zapsat opatření. Uvádějí se zde informace jako datum, popis, příčina, okamžité opatření, dlouhodobé opatření atd.
- **Pracovní úraz**

V souvislosti s posledním pracovním úrazem v rámci daného projektu je vyhotoven dokument, který obsahuje popis, jak a za jakých okolností k úrazu došlo a dále pak doporučení či dokonce nařízení jak se chovat, aby se tento typ úrazu neopakoval.
- **Z-grafy**

Tento prvek je složen ze čtyř grafů, kterým se souhrnně říká interně Z-grafy. Zaznamenávají údaje o šrotu, počtu reklamací a reklamovaných kusech, audit výrobku a náklady na nekvalitu.

- Analýza poruch

Pojmenování může být poněkud matoucí, jelikož se jedná o záznamy o prostojích. Tedy pokud například dojde k závadě na stroji (odtud pojmenování Analýza poruch), nedostatek obsluhujícího personálu nebo třeba manipulace s materiálem. Mezi prostoje také patří povinné přestávky, mezi něž patří například polední a hluková přestávka. Tyto povinné přestávky se však v tomto případě neevidují.

U konkrétního projektu se eviduje, jaké prostoje v daný měsíc nastaly a jak dlouho trvaly.

- ProjectScorecard

Jedná se o tabulku, kde jsou na měsíční bázi zaznamenávány hodnoty ke klíčovým ukazatelům, které jsou: doba cyklu, dobré kusy, FTTQ, QEE, počet dělníků v procesu, potřebná plocha, PPM, produktivita, sériová obrátka zásob, stupeň rozpracovanosti KVP, šrot, vytížení a zákaznický tak.

- Informace o projektu

Dokument obsahující nejdůležitější informace o daném projektu. Je zde například, kontakt na projektového vedoucího, základní manuál výroby a další. Tento prvek je prakticky po celou dobu trvání projektu neměnný. Zřídka dochází k jeho aktualizaci například v souvislosti s reklamacemi nebo personálních změn.

- Foto týmu

Podobně jako u předchozího prvku, foto týmu se často nemění. Pouze pokud dojde k migraci zaměstnanců na jiný projekt či odchodu zaměstnance z podniku. Obsahem jsou jména a fotografie lidí, kteří na projektu pracují.

- Úzké místo projektu

Dokument zachycující, kde v čem spočívá úzké místo projektu. U zaběhlého projektu se také příliš často nemění.

- OEE, HIP

Jako médium je zde sloupcový graf a ukazateli jsou zde OEE a HIP. OEE je zkratka z anglického Overall Equipment Effectiveness, tedy celková efektivnost zařízení. Tento ukazatel umožňuje porovnat efektivnost výrobních zařízení. [9] Druhým ukazatelem je HIP – hodinový index produktivity, který určuje vztah mezi vyrobenými dobrými kusy a odpracovanou dobou pracovníka. [10]

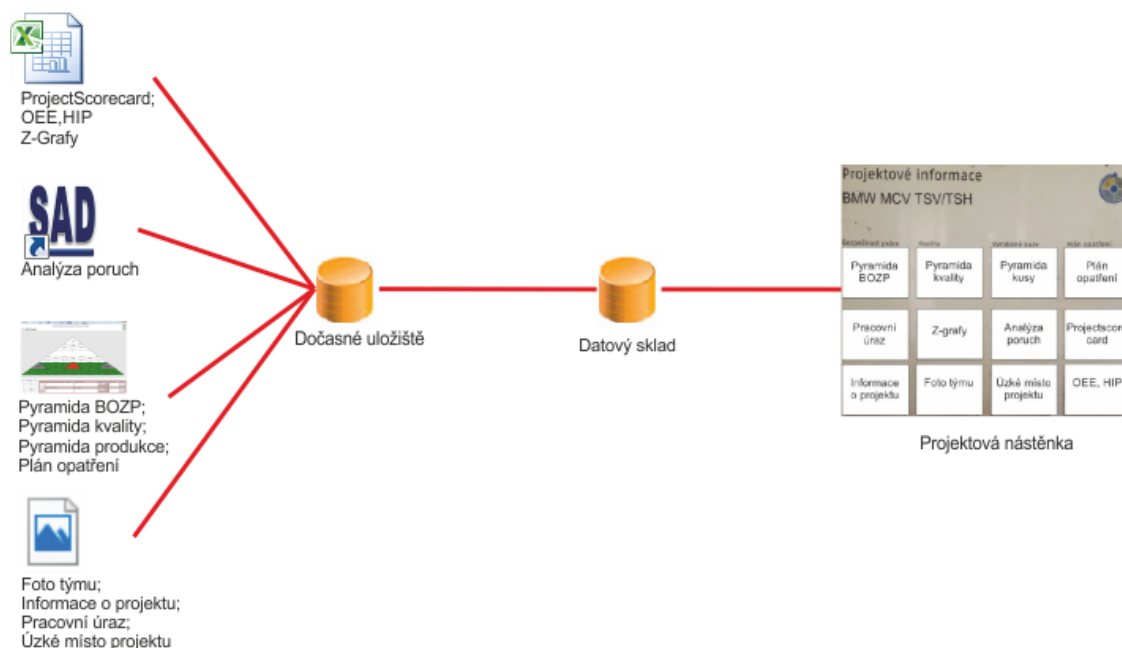
5.1 Analýza připravenosti

Zdrojová data budou čerpána z několika zdrojů. Z aplikace SAD, která je vyvíjena přímo ve firmě a ukládá data do Microsoft SQL databáze. Dalším zdrojem jsou soubory MS Excel. Pro sběr některých dat, která jsou zaznamenávána nyní pouze papírově, bude vytvořen elektronický formulář, skrze který bude ukládána data do SQL databáze a odtud nahráván na projektovou nástěnku. Poněkud netypickým zdrojem jsou dokumenty a obrázky, se kterými

nejdou prováděny žádné analýzy a jsou pouze zobrazeny, tak jak jsou vytvořené. Jako kvalitativně nejhorší shledávám zdrojová data uložená v souborech MS Excel. Tyto soubory sice mají pro danou kategorii projektové nástěnky poměrně standardizovanou strukturu, ovšem ne vše je tato struktura plně dodržována. Osoby, které tyto dokumenty vytváří, budou upozorněny, aby dodržovali standardizovanou strukturu a pojmenování. Obecným problémem souvisejícím s datovou kvalitou je nejednotné pojmenovávání jednotlivých projektu napříč organizací. Situace je složitější i tím, že ne pro každý projekt jsou sledovány všechny kategorie na projektové nástěnce a některé projekty jsou sloučeny. Technické zázemí ve firmě je pro zavedení business intelligence velmi příznivé. K dispozici je virtuální server s Microsoft SQL databází a programové vybavení je také zajištěno.

5.2 Návrh architektury

Jelikož jsou zdrojová data čerpána z různých typů uložení a je třeba je transformovat, je využito dočasné uložení dat, které je zařazeno do architektury řešení. Tato architektura je na obrázku 14. K procesu ETL je využit nástroj Visual Studio, modul pro Business Intelligence. Obě uložení, jak dočasné, tak datový sklad, budou vytvářena přes SQL příkazy v programu SQL Server Management Studio.



Obrázek 14: Návrh architektury.
Zdroj: [vlastní]

6 Analýza dat

Podnik, pro který je zpracováno toto řešení business intelligence, je zásadně ovlivňován standardy německé mateřské firmy. Změna způsobu ukládání dat do produkčních databází a souborů MS Excel, by znamenal procesně velmi náročnou změnu, která by trvala delší dobu. Dále pak v současné době probíhá příprava zavádění nového systému pro sběr dat z výroby. Tato diplomová práce se tedy nezabývá změnami ve způsobu získávání a ukládání zdrojových dat.

Jak již bylo zmíněno dříve, největším kvalitativním problémem ve zdrojových datech je pojmenování jednotlivých projektů. Aby byla jistota, že jsou použita ke konkrétnímu projektu správná data, byla vytvořena tabulka – Tabulka 3, která konsoliduje názvy projektů z nejpodstatnějších prvků. Pro účely této diplomové práce bylo vybráno deset reálných běžících projektů, pro které se bude vytvářet řešení business intelligence, potažmo automatizace a vizualizace. Projekt nesoucí ID 11 je fiktivní a je vytvořen jako modelový projekt s fiktivními daty, na kterém budou ilustrovány obsahy tabulek, grafy a reporty. Tabulka 3 zároveň představuje budoucí číselníkovou tabulku pro projekty.

Sloupec Název obsahuje přesný název projektu, převzatý ze současných projektových nástěnek. Sloupec Název_PSC zachycuje jména projektů, jak jsou uváděna v dokumentu MS Excel pro ProjectScorecard. Jak je možné si všimnout, u projektu ID 2 není vyplněn atribut Název_PSC. Je to z toho důvodu, že se pro tento projekt Scorecard nevytváří. Další sloupce jsou již z pojmenování jasné.

Tabulka 3: Tabulka projektů - konsolidace názvů.

| ID | Název | Název_PSC | Název_ZGraf | Název_Analýza poruch |
|----|----------------------------------|---------------------|--------------------------------------|---|
| 1 | BMW PL6 ZB Centrální trubka | BMW PL6 CT | 420_BMW_Centralni_trubka_PL6 | 420 BMW PL6 Centrální trubka |
| 2 | BMW MCV HVS víko | | 442_BMW_MCV_viko_baterie | 442 BMW MCV DECKEL |
| 3 | BMW MCV TSV/TSH | BMW MCV TSH | 438_BMW_MCV | 438 BMW_MCV_TSV_TSH |
| 4 | VW PQ35 Zadní náprava - svařovna | VWPQ35 | 526_VW_PQ35_Zadni_naprava | 526 VW HAT PQ35 Svařování |
| 5 | VW MQB Hi-Ra | MQB VAT | 578_VW_MQB_Hilfsrahmen | 578 MQB Hi-Ra |
| 6 | SUZUKI YP6 Přední nárazník | Suzuki YP6 TSV | 717_Suzuki_YP6_Předni_naraznik | 717 SUZUKI YP6 Front Bumper |
| 7 | SUZUKI YP6 Přední nápravnice | Suzuki YP6 VA | 719_Suzuki_YP6_předni_nápravnice | 719 SUZUKI YP6 Přední Náprava |
| 8 | SUZUKI YAX | Suzuki SX4 (OP40) | 728_Suzuki_YAX | 728 SUZUKI YAX |
| 9 | Škoda A7 TSV/TSH | Škoda A7 TSV | 565_Skoda_Octavia_A7_predni_naraznik | 565 TSV SKA7 |
| 10 | Fiat 312 Cinquecento B-sloupek | Fiat Cinquecento BS | 201_Fiat_312_Cinquecento_B_sloupek | 106,201,568,600 Ford Fiat Porsche (Z St.+Längsträger) |
| 11 | Škoda Superb CT - FICTIVE | Škoda Superb CT | 111_Skoda_Superb_Centralni_trubka | 111 Škoda Superb Centralni trubka |

Zdroj: [vlastní]

6.1 Zdrojová data

V této podkapitole jsou definována zdrojová data pro jednotlivé prvky včetně jejich umístění a typu.

6.1.1 Pyramida BOZP, Pyramida kvality a Pyramida produkce

Pro všechny tyto pyramidy neexistují data, dostupná v elektronické podobě. V případě pyramidy BOZP to v praxi to funguje tak, že pokud se stane pracovní úraz, směnový mistr nebo projektový vedoucí tuto skutečnost k příslušnému dni zakreslí červeně do pyramidy. Pyramida kvality funguje podobně. Na základě informace o reklamaci dojde k zakreslení do pyramidy.

Informaci o tom zdali je reklamáce, dostane projektový vedoucí ze systému SAP, z něhož by teoreticky šla data čerpat. Nicméně se mi pro tyto účely nepodařil zajistit přístup. Pyramidu produkce vyplňuje mistr na základě toho, zda-li byla splněna norma vyrobených kusů. Záznam dat je tedy pro tyto pyramidy řešen implementací elektronického formuláře do interního systému SAD. Pouze oprávněné osoby, tedy projektový vedoucí a mistři, budou mít přístup k tomuto formuláři.

Formulář je řešen vizuálně a jeho podoba je zachycena na obrázku 15. Jedná se tedy o pyramidu tvořenou menšími pyramidami, které symbolizují dny v měsíci. Správnou pyramidu uživatel vybere na základě záložek a kombo boxů. První kombo box dává na výběr z jednotlivých projektů, tedy pro jaký projekt budou data zadávána a další kombo box je formou kalendáře, kterým slouží k volbě měsíce, pro který je pyramida zobrazena. Volba měsíce je zde umožněna především z důvodu, kdy jsou údaje zadávány zpětně. Například když uživatel zapomene vyplnit poslední den měsíce, ale uvědomí si to již začátkem nového měsíce.

Zadávání dat probíhá kliknutím myši na příslušný den, čím dochází ke změně barvy. Jakmile je zadávání dokončeno, uživatel klikne na tlačítko se symbolem diskety a tím pyramidu uloží, tedy odešle data do SQL databáze.

Obrázek 15: Vizuální formulář pyramidy.

Zdroj: [vlastní]

Jelikož případů, kdy nastane pracovní úraz či reklamáce je málo, je tento formulář částečně automatizován tím, že jsou dny v měsíci, které již nastaly, zbarveny do zelena – tedy že k nežádoucí události nedošlo. Pokud však nastane pracovní úraz, reklamáce či den bez produkce, musí uživatel tuto skutečnost zanést do pyramidy. Každé pole (den) v pyramidě má tedy celkem čtyři stavy, které jsou:

- Bílá barva – den v měsíci ještě nenastal, tudíž se jedná o hodnotu null.
- Zelená barva – den, který již nastal a probíhala v něm produkce. Zároveň v něm nebyla zaznamenána žádná z negativních událostí.
- Červená barva – den, ve kterém probíhala produkce a došlo k negativní události.
- Šedá barva – den, ve kterém neprobíhala produkce.

6.1.2 Plán opatření

Ani pro tento prvek neexistovala elektronická forma uložení dat. Proto musel být vytvořen formulář, skrze který bude vyplňován. Stejně jako prvky pyramid, je integrován do modulu systému SAD a odesílat data do SQL databáze. Tento formulář je v systému umístěn pod pyramidami a již bylo možné ho spatřit na obrázku 15. Vyplňuje se tedy datum, ke kterému dni patří záznam, dále směna, pro niž je připraven kombo box, který nabízí hodnoty R jako ranní, O jako odpolední a N jako noční směna. Dále jsou zde volná pole, kde se popisuje, co se stalo, proč se to stalo a jaká jsou opatření. K opatřením se vážou pole termín, odpovědnost a status. Status je řešen opět kombo boxem, kde má uživatel na výběr z hodnoty 1 -4, které zastupují, v jaké fázi se opatření nachází.

6.1.3 Pracovní úraz; Informace o projektu; Foto týmu; Úzké místo projektu

Tyto prvky patří mezi ty, se kterými nebudou prováděny žádné další akce a budou pouze vizualizovány, tak jak jsou. Dokumenty jsou čerpány ze složky na síťovém disku.

6.1.4 Z-Grafy

Pro tento prvek budou data čerpána ze souborů typu MS Excel. Pro každý projekt je zvláštní soubor s daty. Tyto soubory jsou ukládány do složky na síťovém disku. Formu si je možné prohlédnout na obrázku 16, který znázorňuje fiktivní projekt Škoda Superb CT.

Sloupce obarvené oranžově vyplňuje vedoucího výroby. Modré sloupce vyplňuje projektový vedoucí daného projektu. Sloupce *Šrot /Kč/*, *Šrot /%/*, *Šrot /%/ SV+ ST* a *PPM* obsahují vzorce pro výpočty.

Plnění cílů kvality

Výrobek - zákazník: Škoda Superb CT - Fictive

| Měsíc/rok | Seriové tržby /Kč/ | SV /Kč/ | ST /Kč/ | Šrot /Kč/ | Šrot /%/ | Šrot /%/ SV+ST | Cíl /</ % | Náklady na nekvalitu v TCZK |
|-----------|--------------------|---------|---------|-----------|----------|----------------|-----------|-----------------------------|
| 1.15 | 8 008 712 | 52 721 | 8 600 | 61 322 | 0,66% | 0,77% | 0,64% | 0,000 |
| 2.15 | 4 052 912 | 18 164 | 8 564 | 26 728 | 0,45% | 0,66% | 0,64% | 0,000 |
| 3.15 | 5 042 496 | 22 146 | 4 884 | 27 030 | 0,44% | 0,54% | 0,64% | 0,000 |
| 4.15 | 5 997 476 | 377 294 | 8 236 | 385 530 | 6,29% | 6,43% | 0,64% | 0,000 |
| 5.15 | 4 041 933 | 28 730 | 7 690 | 36 421 | 0,71% | 0,90% | 0,64% | 0,000 |
| 6.15 | 3 089 835 | 181 614 | 4 700 | 186 314 | 5,88% | 6,03% | 0,64% | 0,000 |
| 7.15 | 6 784 270 | 108 014 | 7 213 | 115 226 | 1,59% | 1,70% | 0,64% | 0,000 |
| 8.15 | 4 542 163 | 50 689 | 5 246 | 55 935 | 1,12% | 1,23% | 0,64% | 0,000 |
| 9.15 | 6 044 828 | 281 469 | 5 246 | 286 715 | 4,66% | 4,74% | 0,64% | 0,000 |
| 10.15 | 4 019 189 | 62 206 | 8 852 | 71 058 | 1,55% | 1,77% | 0,64% | 0,000 |
| 11.15 | 2 762 303 | 479 088 | 5 573 | 484 662 | 17,34% | 17,55% | 0,64% | 0,000 |
| 12.15 | 1 931 005 | 55 977 | 3 279 | 59 256 | 2,90% | 3,07% | 0,64% | 0,000 |

| Měsíc/rok | Vydodané kusy | Počet reklamací (technické) | Počet reklam. kusů | Audit výrobku | Audit výrobku cíl (%) | PPM |
|-----------|---------------|-----------------------------|--------------------|---------------|-----------------------|-----|
| 1.15 | 13 780 | 0 | 0 | | 95% | 0 |
| 2.15 | 6 656 | 0 | 0 | | 95% | 0 |
| 3.15 | 8 788 | 0 | 0 | | 95% | 0 |
| 4.15 | 10 452 | 0 | 0 | | 95% | 0 |
| 5.15 | 7 124 | 0 | 0 | | 95% | 0 |
| 6.15 | 5 304 | 0 | 0 | | 95% | 0 |
| 7.15 | 11 908 | 1 | 1 | | 95% | 84 |
| 8.15 | 8 008 | 0 | 0 | | 95% | 0 |
| 9.15 | 10 658 | 0 | 0 | | 95% | 0 |
| 10.15 | 7 034 | 0 | 0 | | 95% | 0 |
| 11.15 | 4 654 | 0 | 0 | | 95% | 0 |
| 12.15 | 3 172 | 0 | 0 | | 95% | 0 |

Obrázek 16: Ukázka zdrojových dat pro Z-Grafy.
Zdroj: [vlastní]

6.1.5 Analýza poruch

Data pro prvek analýza poruch jsou čerpána z SQL serveru systému SAD. Na základě komplexního SQL dotazu jsou získána potřebná data z několika tabulek.

6.1.6 Project Scorecard

Project Scorecard je další prvek, jehož zdrojová data jsou čerpána z dokumentu typu MS Excel. Pro všechny projekty v rámci jednoho závodu však existuje pouze jeden dokument a data pro jednotlivé projekty jsou umístěny na listech. Tento soubor je umístěn na síťovém disku. Ukázka jak tento dokument vypadá, je na obrázku 17.

| K3 Project-Scorecard | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--|----------------------------|--|------------------|--|--|--|-------|--|-----------------|--|-----------------|--|-------|--|--------|--|-------|--|--------|--|-------------------------------|--|----------|--|-------|--|-------|--|-------|--|----------|--|----------|--|-------|--|
| Projekt: | | GPIN: | | 0 | | Systémové hranice | | | | Projektový tým: | | Verze: | | 5.0 | | | | | | | | | | | | | | | | | | | | | | | |
| Škoda Superb CT | | | | | | Svaňování (kromě průběžné doby a šifru v CZK) dohromady se Z. Stütze | | | | | | Výběr: | | | | | | | | | | | | | | | | | | | | | | | | | |
| Kategorie: ST A/B-/C-Pillar | | | | | | P. Kounek | | | | | | Datum: | | | | | | | | | | | | | | | | | | | | | | | | | |
| Pracovní doba za směnu: | | 7,50 h | | M/S <E> Forecast | | Realita | | Cíl | | YTD | | Průměr za měsíc | | | | | | | | | | Rozdíl měsíce oproti cíli v % | | Poznámka | | | | | | | | | | | | | |
| KPIs | | Datum: | | 02.02.12 | | Q 2014 | | 2015 | | | | leden | | únor | | březen | | duben | | květen | | červen | | červenec | | srpen | | září | | říjen | | listopad | | prosinec | | | |
| Povinný | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Zákaznický takt | | sml / kus | | 61,3 | | 178,1 | | 178,1 | | 197,9 | | 117,6 | | 232,5 | | 184,3 | | 155,0 | | 227,4 | | 305,4 | | 136,0 | | 202,3 | | 152,0 | | 224,1 | | 333,2 | | 510,7 | | 187% | |
| Doba cyklu (BDT) | | sml / kus | | 49,1 | | 107,9 | | 107,9 | | 113,1 | | 103,3 | | 117,1 | | 126,0 | | 106,0 | | 120,7 | | 116,8 | | 114,4 | | 94,8 | | 108,7 | | 132,2 | | 134,5 | | 83,1 | | -23% | |
| Dobré kusy | | sml / kus / směna | | 440,3 | | 206,1 | | 212,8 | | 200,9 | | 212,0 | | 190,1 | | 188,3 | | 204,5 | | 188,6 | | 171,3 | | 219,8 | | 233,2 | | 197,7 | | 165,1 | | 168,5 | | 258,9 | | 22% | |
| Počet dělníků v procesu | | sml / kus / směna | | 1,0 | | 1,2 | | 1,2 | | 1,0 | | 1,0 | | 1,0 | | 1,0 | | 1,0 | | 1,1 | | 1,1 | | 1,1 | | 1,1 | | 1,0 | | 1,1 | | 1,0 | | 0,9 | | -29% | |
| Produktivita | | Dobré kusy / pracovní doba | | 58,7 | | 23,5 | | 23,5 | | 26,0 | | 29,1 | | 25,7 | | 24,0 | | 26,1 | | 23,1 | | 21,0 | | 26,2 | | 29,1 | | 26,7 | | 20,2 | | 22,1 | | 40,4 | | 72% | |
| OEE | | v % | | 80,0% | | 80,4% | | 82,0% | | 82,7% | | 81,1% | | 82,5% | | 87,9% | | 80,3% | | 84,3% | | 74,1% | | 93,1% | | 81,9% | | 79,6% | | 80,8% | | 84,0% | | 79,7% | | -3% | |
| Šrot | | s % šrotu | | 2,00% | | 1,39% | | 0,64% | | 3,19% | | 0,77% | | 0,06% | | 0,54% | | 6,43% | | 0,90% | | 6,03% | | 1,70% | | 1,23% | | 4,74% | | 1,77% | | ##### | | 3,07% | | 379% | |
| PPM | | v % | | 14 | | 0 | | 10 | | 7 | | 0 | | 0 | | 0 | | 0 | | 0 | | 0 | | 84 | | 0 | | 0 | | 0 | | 0 | | 0 | | -100% | |
| FTTQ | | v % | | 97% | | 100% | | 90% | | 92% | | 100% | | 100% | | 100% | | 100% | | 100% | | 100% | | 100% | | 100% | | 100% | | 100% | | 100% | | 100% | | 11% | |
| Vytížení | | v % | | 100% | | 85% | | 100% | | 85% | | 120% | | 73% | | 91% | | 100% | | 80% | | 62% | | 106% | | 56% | | 101% | | 81% | | 64% | | 79% | | -21% | |
| Seriová obrátka zásob | | sml | | 8 | | 11 | | 10 | | 8,5 | | 4,8 | | 8,6 | | 8,8 | | 7,1 | | 7,1 | | 10,6 | | 7,9 | | 6,1 | | 7,7 | | 6,6 | | 12,8 | | 13,6 | | 36% | |
| Potřebná plocha | | m² | | 204 | | 100 | | 100 | | 100 | | 100 | | 100 | | 100 | | 100 | | 100 | | 100 | | 100 | | 100 | | 100 | | 100 | | 100 | | 100 | | 0% | |
| Stupeň rozpracovanosti KVP | | v % | | 100% | | 100% | | 100% | | 100% | | 100% | | 100% | | 100% | | 100% | | 100% | | 100% | | 100% | | 100% | | 100% | | 100% | | 100% | | 100% | | 0% | |
| Doplňková data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <div><div></div><div></div><div></div></div> <div><div>Cíl osazen / překročen</div><div>do 20% pod cílem</div><div>víc jak 20% pod cílem</div></div> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Obrázek 17: Ukázka zdrojových dat pro ProjectScorecard.

Zdroj: [vlastní]

Jak je vidět jedná se o ne zcela ideální zdrojová data. Pro snadnější převod těchto dat na SQL server byl použit skript, který po uložení dokumentu Excel vygeneruje soubor typu CSV a upraví data do podoby vhodnější pro SQL. Strukturu vygenerovaného CSV souboru znázorňuje obrázek 18. Na obrázku je vidět pouze jeden projekt, ovšem výpis je pouze zkrácen. CSV soubor obsahuje pod sebou všechny projekty.

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | |
|----|-----------|------------|------|-----------|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|------------|-----------|------------|---|
| 1 | Part | Pracovní c | Year | Zákaznick | Doba cyklů | Dobré kus | Počet děl | Produktiv | OEE | Šrot | PPM | FTTQ | Vytížení | Seriová ot | Potřebná | Stupeň roz | |
| 2 | Škoda Sup | MS <= Fo | 2015 | 61.32 | 49.1 | 440.3 | 1 | 58.7 | 0.8 | 0.02 | | 14 | 0.97 | 1 | 6 | 204 | |
| 3 | Škoda Sup | Realita | 2015 | 178.10679 | 107.91876 | 206.08973 | 1.1948291 | 23.528775 | 0.8040513 | 1.3927898 | 0 | 1 | 0.8499938 | 10.815573 | 100 | 1 | |
| 4 | Škoda Sup | Cil | 2015 | 178.10679 | 107.91876 | 212.83899 | 1.2 | 23.5 | 0.82 | 0.0064 | 10 | 0.9 | 1 | 1 | 10 | 100 | |
| 5 | Škoda Sup | YTD | 2015 | 197.85454 | 113.13379 | 200.88575 | 1.0330998 | 26.044946 | 0.8274932 | 3.1895420 | 6.9980965 | 0.9166666 | 0.8453395 | 8.4706820 | 100 | 1 | |
| 6 | Škoda Sup | leden | 2015 | 117.56168 | 103.34393 | 211.97532 | 0.9696222 | 29.148854 | 0.8113468 | 7.6587867 | 0 | 1 | 1.2009259 | 4.7501451 | 100 | 1 | |
| 7 | Škoda Sup | únor | 2015 | 232.49138 | 117.13807 | 190.12102 | 0.9863842 | 25.699386 | 0.8248300 | 6.5947249 | 0 | 1 | 0.7344444 | 8.5878003 | 100 | 1 | |
| 8 | Škoda Sup | březen | 2015 | 184.34228 | 126.00711 | 188.26829 | 1.0451219 | 24.018669 | 0.8786349 | 5.3603988 | 0 | 1 | 0.9111111 | 8.7402513 | 100 | 1 | |
| 9 | Škoda Sup | duben | 2015 | 154.99425 | 106.00619 | 204.52125 | 1.0447319 | 26.101910 | 0.8029822 | 6.4281962 | 0 | 1 | 1.0018518 | 7.0723306 | 100 | 1 | |
| 10 | Škoda Sup | květen | 2015 | 227.40033 | 120.69958 | 188.64212 | 1.0878680 | 23.120711 | 0.8432972 | 9.0106689 | 0 | 1 | 0.8038148 | 7.0809938 | 100 | 1 | |
| 11 | Škoda Sup | červen | 2015 | 305.42986 | 116.75804 | 171.31108 | 1.0870083 | 21.013157 | 0.7408128 | 0.0602990 | 0 | 1 | 0.6214814 | 10.026696 | 100 | 1 | |
| 12 | Škoda Sup | červenec | 2015 | 136.04299 | 114.39567 | 219.82105 | 1.12 | 26.169172 | 0.9313547 | 1.6984328 | 83.977158 | 0 | 1 | 1.0555555 | 7.9017467 | 100 | 1 |
| 13 | Škoda Sup | srpen | 2015 | 202.29770 | 94.787049 | 233.24777 | 1.0681145 | 29.116451 | 0.8188469 | 1.2314571 | 0 | 1 | 0.5627777 | 6.1016483 | 100 | 1 | |
| 14 | Škoda Sup | září | 2015 | 151.96998 | 108.69415 | 197.68062 | 0.9877446 | 26.684444 | 0.7958047 | 4.7431394 | 0 | 1 | 1.0214074 | 7.6651969 | 100 | 1 | |
| 15 | Škoda Sup | říjen | 2015 | 224.12838 | 132.17153 | 165.05133 | 1.0896646 | 20.195979 | 0.8079662 | 1.7679638 | 0 | 1 | 0.8116666 | 6.5500830 | 100 | 1 | |
| 16 | Škoda Sup | listopad | 2015 | 333.19621 | 134.45981 | 168.52941 | 1.0190311 | 22.050933 | 0.8395 | 0.1754557 | 0 | 1 | 0.6422222 | 12.830111 | 100 | 1 | |
| 17 | Škoda Sup | prosinec | 2015 | 510.71878 | 83.108307 | 258.88202 | 0.8554461 | 40.350413 | 0.7968610 | 3.0686499 | 0 | 1 | 0.7858188 | 13.693444 | 100 | 1 | |

Obrázek 18: Ukázka dat upravených skriptem.

Zdroj: [vlastní]

Skript tedy převrátí osy dat, na základě konkrétní buňky v Excelu vytáhne název projektu a rok a postupně takto zpracuje všechny záložky s projekty a uloží je do jednoho souboru pod sebe. V této podobě už je export dat na SQL server jednodušší.

6.1.7 OEE, HIP

Zdrojová data k tomuto prvku jsou uložena opět ve formátu MS Excel na síťovém disku. Data pro všechny hodnoty jsou uloženy v jednom souboru, kde jsou jednotlivé projekty rozděleny do záložek (listů).

| Měsíc | I.O. Stückzahl [Stück] | Bruttostückzahl [Stück] | Produktive Zeit [min] | Anlagenlaufzeit nach BOS | Produktive Anlagenlaufzeit | Qualitätsfaktor | Verfügbarkeitsfaktor | Leistungsfaktor | IST OEE | Cíl OEE | Hodiny HR | IST HIP | Cíl HIP |
|-------------|------------------------------|----------------------------|--------------------------|-----------------------------|-------------------------------|-----------------|----------------------|-----------------|---------|---------|-----------|---------|---------|
| leden 15 | 15274 | 15275 | 30760 | 32425 | 26308 | 100% | 95% | 86% | 81,13% | 81,00% | 524 | 29,15 | 23,50 |
| únor 15 | 8378 | 8383 | 16555 | 19830 | 16356 | 100% | 83% | 99% | 82,43% | 81,00% | 326 | 25,70 | 23,50 |
| březen 15 | 10292 | 10364 | 21622 | 24600 | 21614 | 99% | 88% | 100% | 87,25% | 81,00% | 429 | 24,02 | 23,50 |
| duben 15 | 12294 | 12321 | 21757 | 27050 | 21721 | 100% | 80% | 100% | 80,12% | 81,00% | 471 | 26,10 | 23,50 |
| květen 15 | 9098 | 9099 | 18311 | 21703 | 18302 | 100% | 84% | 100% | 84,32% | 81,00% | 394 | 23,12 | 23,50 |
| červen 15 | 6388 | 6388 | 14500 | 16780 | 12431 | 100% | 86% | 86% | 74,08% | 81,00% | 304 | 21,01 | 23,50 |
| červenec 15 | 13922 | 13932 | 26635 | 28500 | 26544 | 100% | 93% | 100% | 93,07% | 81,00% | 532 | 26,17 | 23,50 |
| srpen 15 | 7876 | 7876 | 13357 | 15195 | 12442 | 100% | 88% | 93% | 81,88% | 81,00% | 271 | 29,12 | 23,50 |
| září 15 | 12008 | 12008 | 23900 | 27335 | 21753 | 100% | 87% | 91% | 79,58% | 81,00% | 450 | 26,68 | 23,50 |
| říjen 15 | 8038 | 8052 | 18343 | 21915 | 17707 | 100% | 84% | 97% | 80,66% | 81,00% | 398 | 20,20 | 23,50 |
| listopad 15 | 6494 | 6506 | 15415 | 17340 | 14557 | 100% | 89% | 94% | 83,80% | 81,00% | 295 | 22,05 | 23,50 |
| prosinec 15 | 12206 | 12212 | 19062 | 21217 | 16907 | 100% | 90% | 89% | 79,65% | 81,00% | 303 | 40,35 | 23,50 |

Obrázek 19: Ukázka zdrojových dat pro OEE, HIP.

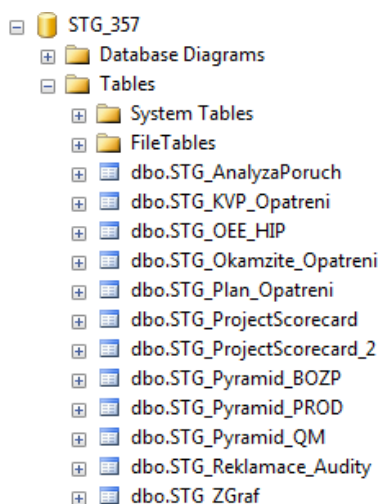
Zdroj:[vlastní]

7 Návrh řešení

V této kapitole se zabývá návrhem datového uložště a následně návrhem procesu naplnění toho datového uložště, tedy ETL a na závěr také ukázkou vytvoření OLAP kostky.

7.1 Dočasné uložště dat

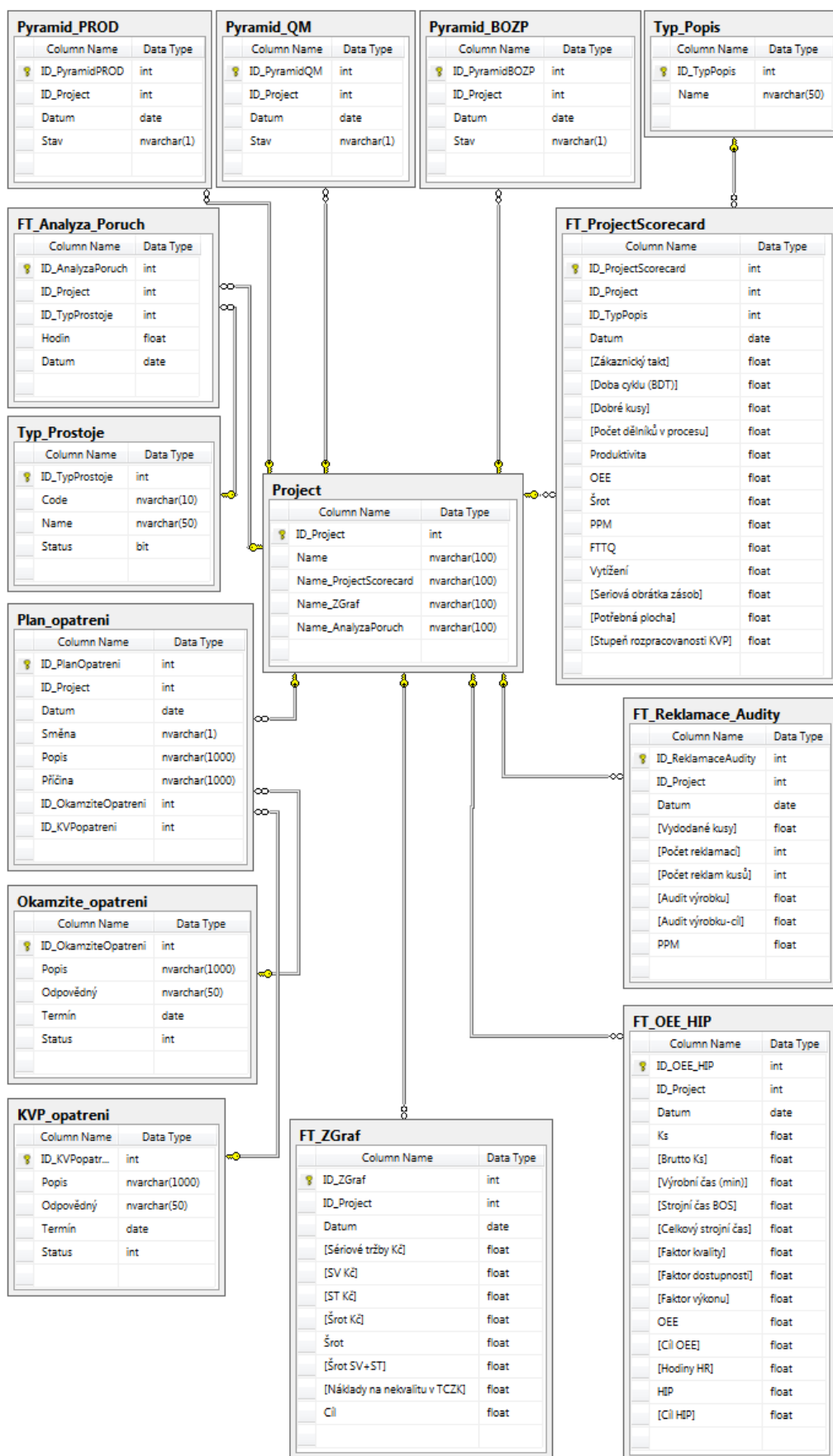
Aby bylo jednoznačně odděleno dočasné uložště od datového skladu, je pro něj vyčleněna zvláštní databáze na SQL serveru. Jména tabulek jsou psána s prefixem STG. Náhled do databáze dočasného uložště, které je pojmenováno STG_357 je na obrázku 20.



Obrázek 20: Dočasné uložště dat.
Zdroj:[vlastní]

7.2 Datový sklad

Obrázek 21, poskytuje náhled na schéma datového skladu. Tabulky s prefixem FT jsou tabulky faktů. Společnou dimenzí pro všechny tabulky faktů je tabulka Project. Další společnou dimenzí pro všechny tabulky faktů by mohla být číselníková tabulka pro čas. Ta je ovšem generována společně s OLAP kostkou.



Obrázek 21: Schéma databáze datového skladu.
Zdroj: [vlastní]

7.3 ETL proces

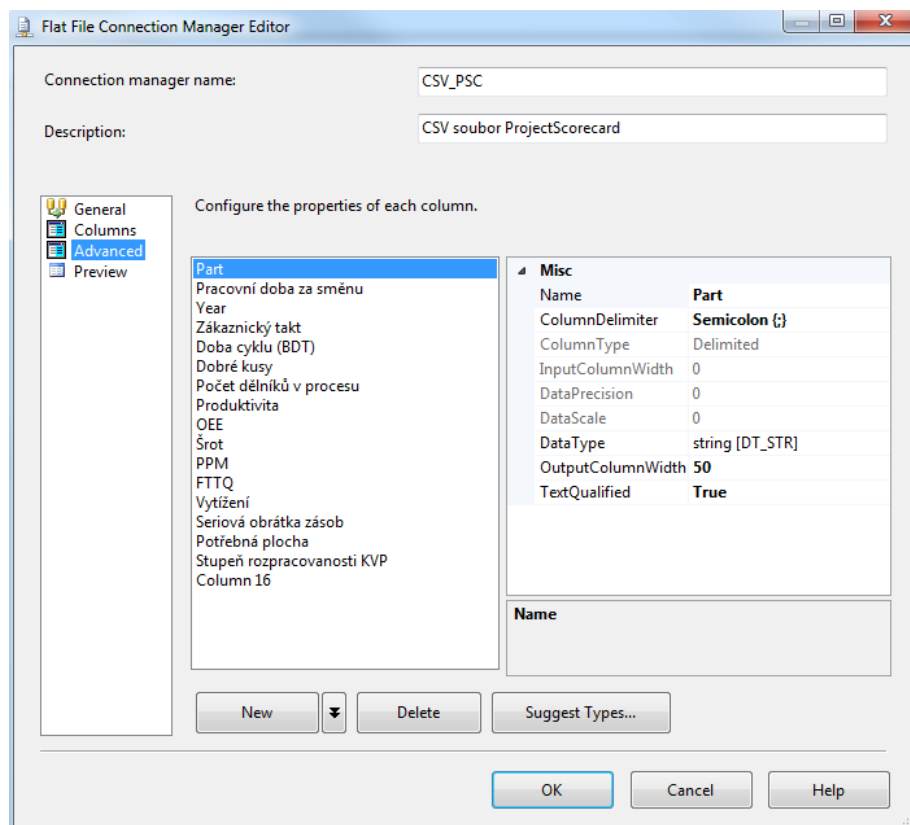
Nyní se dostávám k návrhu procesu nahrání dat do dočasného uložště a datového skladu. Jelikož pracuji v programu Microsoft Visual Studio s anglickým jazykovým prostředím, některé anglické termíny budou při popisu a na obrázcích zachovány.

Na úvod pouze velmi stručně připomenu práci s programem. Pro konstrukci ETL pump je založen nový projekt typu integrační služby (Integration Services Project). Jakmile je projekt vytvořen, tak se v pravé části obrazovky zobrazí okno s názvem Solution Explorer. V tomto okně se po kliknutí pravým tlačítkem myši na SSIS Packages zobrazí menu, kde je možnost založit nový „balíček“. Tím se otevře pracovní plocha a je možné začít s konstrukcí datových pump. V levé části obrazovky se nachází okno s názvem SSIS Toolbox, ve kterém jsou různé nástroje. Základním nástrojem je Data Flow Task (Datový tok). Přetáhnutím datového toku na pracovní plochu a následným otevřením toho nástroje se otevírají další možnosti, které již přímo souvisí s tvorbou datových pump.

7.3.1 *Project Scorecard*

Jako první je sestrojena datová pumpa pro získání dat pro prvek ProjectScorecard. Domnívám se, že tento prvek je tím nejproblematictější. Jedná se totiž o poměrně kompilovanou tabulku a to přesto, že už byla práce trochu ulehčena skriptem, který provedl několik úkonů. Je tedy zapotřebí data trochu upravit už v dočasném uložšti.

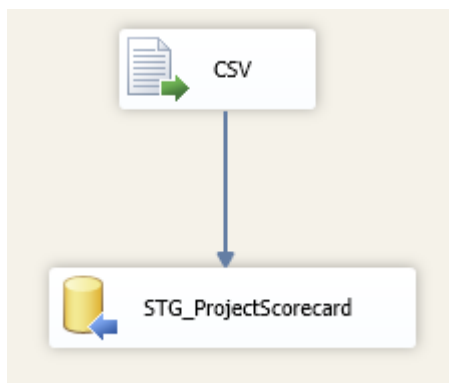
V prvním kroku je zkonstruována jednoduchá datová pumpa, která nahraje data z již zmiňovaného souboru CSV, který vygeneroval skript. Nejprve je zvolen nový zdroj, přetáhnutím nástroje Source Assistant. Jelikož je potřeba nahrát data ze souboru CSV, je zvolena možnost Flat File a poté je nastaveno nové připojení. Prozkoumáním manažera připojení detailněji, je zde možnost nahlédnout na data, atributy a jejich datové typy. V tomto prvním kroku však není třeba nic měnit a data jsou nakopírována ze souboru CSV do dočasného uložště jedna ku jedné. Tento krok ilustruje obrázek 22.



Obrázek 22: Manažer připojení.

Zdroj: [vlastní]

Jelikož není potřeba provádět žádné další úpravy dat, přejdu rovnou k volbě místa, kam mají být data nahrána. K tomu slouží Destination Assistant. Jako typ je zvolen SQL Server a je nastaveno nové připojení. Jelikož jde o kopírování dat jedna ku jedné, je možné novou SQL tabulku vygenerovat přímo z Visual Studia. Výslednou pumpu po pojmenování nástrojů zobrazuje obrázek 23.



Obrázek 23: Datová pumpa - nahrávání dat ze souboru CSV.

Zdroj: [vlastní]

Ve druhém kroku je provedena úprava datových typů. Je použit SQL příkaz namísto předpřipravených nástrojů, jelikož je potřeba použít komplexnější způsob převodu datového typu. Mezi hodnotami, které je potřeba převést, jsou totiž i prázdné hodnoty, se kterými si klasický příkaz CONVERT neporadí.

Je tedy vytvořen nový datový zdroj typu SQL server a na místo určení tabulky je vložen SQL příkaz:

```
SELECT CONVERT(nvarchar(20),[Part]) as Name_ProjectScorecard
, CONVERT(nvarchar(20),[Pracovní doba za směnu]) as Měsíc
, CONVERT(nvarchar(20),[Year]) as Rok
, TRY_PARSE([Zákaznický takt] as float) as [Zákaznický takt]
, TRY_PARSE([Doba cyklu (BDT)] as float) as [Doba cyklu (BDT)]
, TRY_PARSE([Dobré kusy] as float) as [Dobré kusy]
, TRY_PARSE([Počet dělníků v procesu] as float) as [Počet dělníků v procesu]
, TRY_PARSE([Produktivita] as float) as [Produktivita]
, TRY_PARSE([OEE] as float) as [OEE]
, TRY_PARSE([Šrot] as float) as [Šrot]
, TRY_PARSE([PPM] as float) as [PPM]
, TRY_PARSE([FTTQ] as float) as [FTTQ]
, TRY_PARSE([Vytížení] as float) as [Vytížení]
, TRY_PARSE([Seriová obrátka zásob] as float) as [Seriová obrátka zásob]
, TRY_PARSE([Potřebná plocha] as float) as [Potřebná plocha]
, TRY_PARSE([Stupeň rozpracovanosti KVP] as float) as [Stupeň rozpracovanosti KVP]
FROM [STG_357].[dbo].[STG_ProjectScorecard]
```

Díky funkci TRY_PARSE jsou zkonvertovány hodnoty požadovaných sloupců do datového formátu float. Nyní stačí vytvořit nové umístění, kam budou data nahrána. V tomto případě je použita nová tabulka v dočasném uložišti s názvem STG_ProjectScorecard_2.

Nyní jsou tedy data ze zdrojového souboru nahrána v SQL databázi, konkrétně v dočasném uložišti. Následuje konstrukce další datové pumpy, která nahraje data z dočasného uložíště do datového skladu.

Ještě před tím je však zapotřebí vytvořit číselníkovou tabulku, která bude obsahovat názvy jednotlivých projektů. Struktura a data v tabulce jsou zobrazena v tabulce 3, která je v kapitole 6 Analýza dat. SQL příkaz pro vytvoření a naplnění tabulky je:

```
CREATE TABLE [DWH_357].[dbo].[Project] (
ID_Project int PRIMARY KEY,
Name nvarchar(100),
Name_ProjectScorecard nvarchar(100),
Name_ZGraf nvarchar(100),
Name_AnalyzaPoruch nvarchar(100)
);
--Naplnění číselníku Project
insert into [DWH_357].[dbo].[Project] (ID_Project, Name,
Name_ProjectScorecard, Name_ZGraf, Name_AnalyzaPoruch)
values (1, 'BMW PL6 ZB Centrální trubka', 'BMW PL6 CT',
'420_BMW_Centralni_trubka_PL6', '420 BMW PL6 Centrální trubka');
insert into [DWH_357].[dbo].[Project] (ID_Project, Name,
Name_ProjectScorecard, Name_ZGraf, Name_AnalyzaPoruch)
values (2, 'BMW MCV HVS víko', null, '442_BMW_MCV_víko_baterie', '442 BMW
MCV DECKEL');
insert into [DWH_357].[dbo].[Project] (ID_Project, Name,
Name_ProjectScorecard, Name_ZGraf, Name_AnalyzaPoruch)
values (3, 'BMW MCV TSV/TSH', 'BMW MCV TSH', '438_BMW_MCV', '438
BMW_MCV_TSV_TSH');
insert into [DWH_357].[dbo].[Project] (ID_Project, Name,
Name_ProjectScorecard, Name_ZGraf, Name_AnalyzaPoruch)
values (4, 'VW PQ35 Zadní náprava - svařovna', 'VWPQ35',
'526_VW_PQ35_Zadni_naprava', '526 VW HAT PQ35 Svařování');
```

```

insert into [DWH_357].[dbo].[Project] (ID_Project, Name,
Name_ProjectScorecard, Name_ZGraf, Name_AnalyzaPoruch)
values (5, 'VW MQB Hi-Ra', 'MQB VAT', '578_VW_MQB_Hilfsrahmen', '578 MQB Hi-
Ra');
insert into [DWH_357].[dbo].[Project] (ID_Project, Name,
Name_ProjectScorecard, Name_ZGraf, Name_AnalyzaPoruch)
values (6, 'SUZUKI YP6 Přední nárazník', 'Suzuki YP6 TSV',
'717_Suzuki_YP6_Přední_nárazník', '717 SUZUKI YP6 Front Bumper');
insert into [DWH_357].[dbo].[Project] (ID_Project, Name,
Name_ProjectScorecard, Name_ZGraf, Name_AnalyzaPoruch)
values (7, 'SUZUKI YP6 Přední nápravnice', 'Suzuki YP6 VA',
'719_Suzuki_YP6_přední_nápravnice', '719 SUZUKI YP6 Přední Náprava');
insert into [DWH_357].[dbo].[Project] (ID_Project, Name,
Name_ProjectScorecard, Name_ZGraf, Name_AnalyzaPoruch)
values (8, 'SUZUKI YAX', 'Suzuki SX4 (OP40)', '728_Suzuki_YAX', '728 SUZUKI
YAX');
insert into [DWH_357].[dbo].[Project] (ID_Project, Name,
Name_ProjectScorecard, Name_ZGraf, Name_AnalyzaPoruch)
values (9, 'Škoda A7 TSV/TSH', 'Škoda A7 TSV',
'565_Skoda_Octavia_A7_predni_naraznik', '565 TSV SKA7');
insert into [DWH_357].[dbo].[Project] (ID_Project, Name,
Name_ProjectScorecard, Name_ZGraf, Name_AnalyzaPoruch)
values (10, 'Fiat 312 Cinquecento B-sloupek', 'Fiat Cinquecento BS',
'201_Fiat_312_Cinquecento_B_sloupek', '106,201,568,600 Ford Fiat Porsche (Z
St.+Längsträger)');
insert into [DWH_357].[dbo].[Project] (ID_Project, Name,
Name_ProjectScorecard, Name_ZGraf, Name_AnalyzaPoruch)
values (11, 'Škoda Superb CT - FICTIVE', 'Škoda Superb CT',
'111_Škoda_Superb_Centralni_trubka', '111 Škoda Superb Centralni trubka');

```

Druhou číselníkovou tabulkou pro popis hodnot je tabulka s názvem Typ_Popis. SQL příkaz pro vytvoření a naplnění je:

```

create table [DWH_357].[dbo].[Typ_Popis](
ID_TypPopis int PRIMARY KEY,
Name nvarchar(50)
);
--Naplnění číselníku Typ_Popis
INSERT INTO [DWH_357].[dbo].[Typ_Popis]
VALUES (1, 'MS <E> Forecast');
INSERT INTO [DWH_357].[dbo].[Typ_Popis]
VALUES (2, 'Realita');
INSERT INTO [DWH_357].[dbo].[Typ_Popis]
VALUES (3, 'Cíl');
INSERT INTO [DWH_357].[dbo].[Typ_Popis]
VALUES (4, 'YTD');
INSERT INTO [DWH_357].[dbo].[Typ_Popis]
VALUES (5, 'Hodnoty');

```

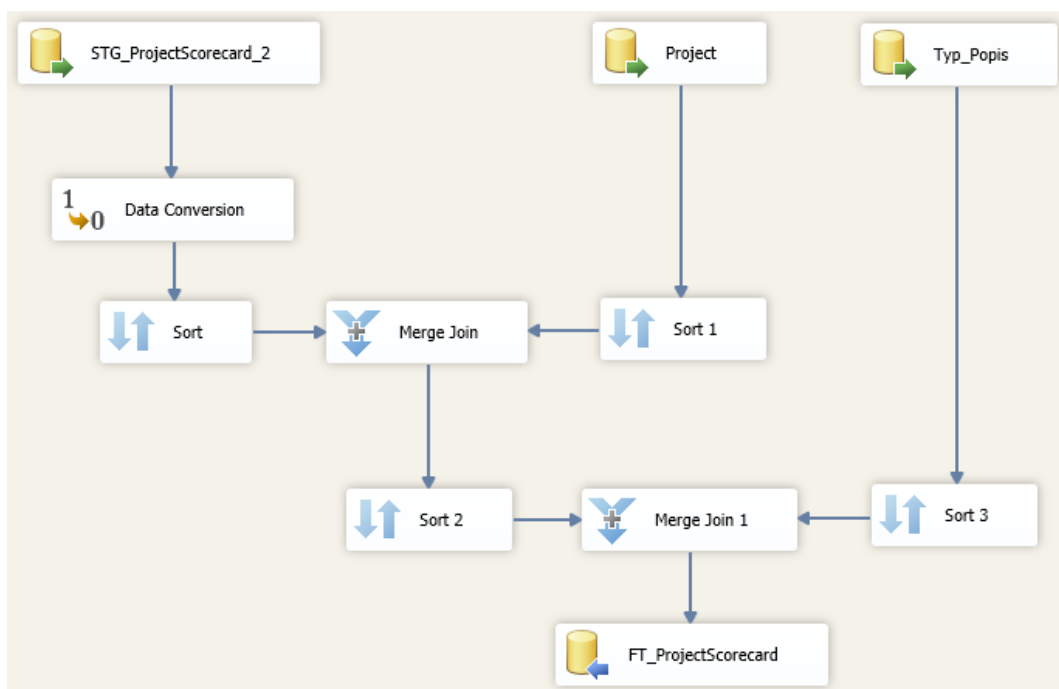
Nyní může začít z konstrukcí datové pumpy. Zdroje dat budou v tomto případě tři. Jedním je SQL dotaz, který vychází z tabulky STG_ProjectScorecard_2 z dočasného uložení. Tento dotaz nahradí názvy měsíců na jejich číselný ekvivalent a vytvoří nový sloupec s názvem Datum. Na základě čísla měsíce, roku a přidání prefixu „1.“ bude vytvořen datum. Dále řeší problém s tím, že ve sloupci Měsíc jsou i hodnoty MS <E> Forecast, Cíl, YTD a Realita. Těmto hodnotám je přidáno ID, které je umístěno do nové sloupce ID_TypPopis.

```

SELECT [Name_ProjectScorecard],[Zákaznický takt],[Doba cyklu (BDT)],[Dobré kusy],
[Počet dělníků
v procesu],[Produktivita],[OEE],[Šrot],[PPM],[FTTQ],[Vytížení] ,[Seriová
obrátká zásob],[Potřebná plocha],[Stupeň rozpracovanosti KVP],
'1.'+(CASE WHEN Měsíc like 'leden' THEN '1'
          WHEN Měsíc like 'únor' THEN '2'
          WHEN Měsíc like 'březen' THEN '3'
          WHEN Měsíc like 'duben' THEN '4'
          WHEN Měsíc like 'květen' THEN '5'
          WHEN Měsíc like 'červen' THEN '6'
          WHEN Měsíc like 'červenec' THEN '7'
          WHEN Měsíc like 'srpen' THEN '8'
          WHEN Měsíc like 'září' THEN '9'
          WHEN Měsíc like 'říjen' THEN '10'
          WHEN Měsíc like 'listopad' THEN '11'
          WHEN Měsíc like 'prosinec' THEN '12'
          ELSE '1'
        END)+'.'+[Rok] as Datum
,CASE WHEN Měsíc like 'MS <E> Forecast' THEN '1'
        WHEN Měsíc like 'Realita' THEN '2'
        WHEN Měsíc like 'Cíl' THEN '3'
        WHEN Měsíc like 'YTD' THEN '4'
        ELSE '5'
      END as ID_TypPopis
FROM [STG_357].[dbo].[STG_ProjectScorecard_2];

```

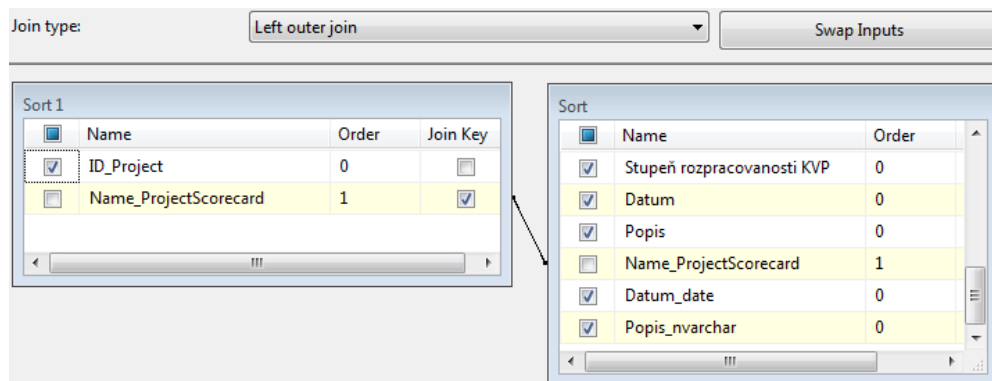
Druhým zdrojem je číselníková tabulka Project nacházející se v datovém skladu. Z ní stačí použít pouze sloupce ID_Project a Name_ProjectScorecard. Posledním zdrojem je číselníková tabulka Typ_Popis. Celá pumpa, která je dále ještě blíže popsána, je zobrazena na obrázku 24.



Obrázek 24: Datová pumpa - Načtení dat pro prvek ProjectScorecard do datového skladu.
Zdroj: [vlastní]

- Data Conversion: V tomto nástroji je převeden datový typ sloupce Datum z varchar na date, aby bylo možné tento sloupec korektně využívat.

- Sort, Sort 1: Aby bylo možné učinit spojení obou zdrojů je zapotřebí seřadit data, na základě kterých bude spojení provedeno. V tomto případě je spojení realizováno na základě názvu projektu pro ProjectScorecard, je tedy seřazen sloupec Name_ProjectScorecard.
- Sort 2, Sort 3: Zde je v obou případech seřazen sloupec ID_TypPopis.
- Merge Join: Jelikož bylo pro řešení projektové nástěnky vybráno jedenáct projektů, jsou přiřazena data ze zdroje STG_ProjectScorecard_2 pouze k těmto jedenácti projektům. Je tedy provedeno levé spojení (left join), tak jak znázorňuje obrázek 25.



Obrázek 25: Merge join - spojení zdrojů STG_ProjectScorecard_2 a Project.
Zdroj: [vlastní]

- Merge Join 1: Jsou spojeny tabulky na základě ID_TypPopis.
- FT_ProjectScorecard: Cílovou tabulkou, kam se výsledky této datové pumpky nahrávají, je tabulka v datovém skladu s názvem FT_ProjectScorecard. SQL příkaz na vytvoření této tabulky je zde:

```
CREATE TABLE [DWH_357].[dbo].[FT_ProjectScorecard] (
    ID_ProjectScorecard int IDENTITY(1,1) PRIMARY KEY,
    ID_Project int FOREIGN KEY REFERENCES Project(ID_Project),
    ID_TypPopis int FOREIGN KEY REFERENCES Typ_Popis(ID_TypPopis),
    Datum date,
    [Zákaznický takt] float,
    [Doba cyklu (BDT)] float,
    [Dobré kusy] float,
    [Počet dělníků v procesu] float,
    [Produktivita] float,
    [OEE] float,
    [Šrot] float,
    [PPM] float,
    [FTTQ] float,
    [Vytížení] float,
    [Seriová obrátka zásob] float,
    [Potřebná plocha] float,
    [Stupeň rozpracovanosti KVP] float
);
```

Tím je naplněn datový sklad pro prvek Project Scorecard a je možné začít s konstrukcí další datové pumpky.

7.3.2 Analýza poruch

Konstrukce samotné datové pumpy pro získání dat pro analýzu poruch z SQL databáze systému SAD do dočasného uložště není nikterak složité, jelikož data budou získána prostřednictvím komplexního SQL dotazu. Jak je možné si všimnout, SQL dotaz spojuje několik tabulek a definuje některá omezení, jako například, že do výběru nezahrnuje poruchu (prostoj), s označením G5 což jsou povinné přestávky, jako třeba polední či hluková.

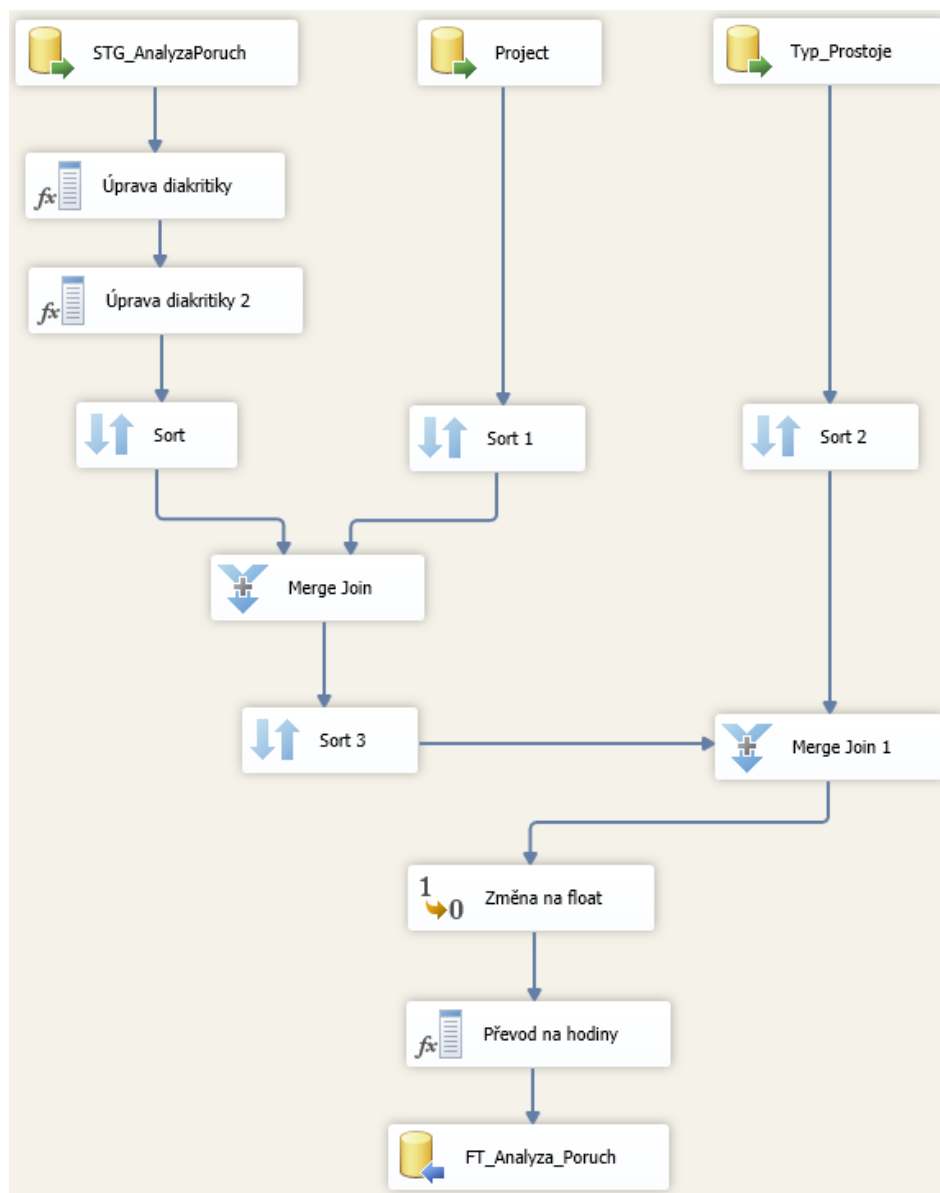
```
SELECT [Ttemp.cStation], [Ttemp.cCode], [Ttemp.cMinutes],
      (SELECT [cCzech] FROM SAD_Languages WHERE
[cModule]='LSV_Downtimes_Standards' AND
      [cLinkedID]=(SELECT [cTitle_LngLinkedID] FROM LSSV_SL_Downtimes_Head
                     WHERE [cHead_LngLinkedID]=[Ttemp.cDowntimeGroup] AND
                        [cCode]=[Ttemp.cCode])) as cDescription, [cDate]
FROM (SELECT T4.[cItem1] as [cStation], T1.[cCode], [cMinutes], T2.[cDate],
      (SELECT TOP 1 [cValue] FROM SAD_Stations_Settings
      WHERE [cTag]='DowntimeGroup' AND [_Station_]=T4.[_Station_]) as
cDowntimeGroup
FROM Shiftlog_Downtimes as T1
INNER JOIN Shiftlog T2 ON T2.[ID]=T1.[Shiftlog_ID]
INNER JOIN SAD_OverView_Tree T4 ON T4.[ID]=T2.[SAD_OverView_Tree_ID]
INNER JOIN Shiftlog_UnderGroups T5 ON T5.[Shiftlog_ID] =
T1.[Shiftlog_ID] AND
      T5.[cRemoved]=0 AND T5.[cState]=1 AND
T1.[cSequenceIndex]=T5.[cSequenceIndex] AND
      (T5.[LSSV_SL_UnderGroups_ID]=0 OR
      T5.[LSSV_SL_UnderGroups_ID]=T1.[LSSV_SL_UnderGroups_ID])
WHERE T1.[cRemoved]=0 AND T2.[cDate] BETWEEN '2015-01-01' AND '2015-12-31' AND
      T2.[cRemoved]=0 AND T2.[cStatus] > 0 AND T1.[cCode] <> 'G5') as Ttemp
ORDER BY [cStation] DESC
```

Výsledná data jsou nahrána do dočasného uložště, do tabulky s názvem STG_AnalyzaPoruch.

Nyní je přistoupeno k druhému kroku a to transformaci a nahrání dat do datového skladu. Ještě před tím je však potřeba vytvořit v datovém skladu číselníkovou tabulku pro typ prostoje. SQL příkaz pro vytvoření a naplnění tabulky je zde:

```
CREATE TABLE [DWH_357].[DBO].[Typ_Prostoje](
  ID_TypProstoje int PRIMARY KEY,
  Code nvarchar(10),
  Name nvarchar(50),
  [Status] bit
);
insert into [DWH_357].[DBO].[Typ_Prostoje]
values (1,'G1', 'Výměna nástroje / palety', 1);
insert into [DWH_357].[DBO].[Typ_Prostoje]
values (2,'G2', 'Úklid pracoviště', 1);
insert into [DWH_357].[DBO].[Typ_Prostoje]
values (3,'G2', 'Úklid pracoviště a předání směny', 1);
insert into [DWH_357].[DBO].[Typ_Prostoje]
values (4,'G3', 'Údržba, boxenstopp', 1);
insert into [DWH_357].[DBO].[Typ_Prostoje]
values (5,'G3', 'Týdenní pravidelná údržba', 1);
insert into [DWH_357].[DBO].[Typ_Prostoje]
values (6,'G4', 'Zpracování', 1);
insert into [DWH_357].[DBO].[Typ_Prostoje]
values (7,'G5', 'Přestávky', 1);
insert into [DWH_357].[DBO].[Typ_Prostoje]
values (8,'G6', 'Porada, příprava pracoviště', 1);
insert into [DWH_357].[DBO].[Typ_Prostoje]
values (9,'G7', 'Manipulace s materiálem', 1);
insert into [DWH_357].[DBO].[Typ_Prostoje]
values (10,'U1', 'Poruchy strojní', 1);
insert into [DWH_357].[DBO].[Typ_Prostoje]
values (11,'U2', 'Poruchy nástroje / palety', 1);
insert into [DWH_357].[DBO].[Typ_Prostoje]
values (12,'U3', 'Nestandardy', 1);
insert into [DWH_357].[DBO].[Typ_Prostoje]
values (13,'U3', 'Korekce rozměru a svaru', 1);
insert into [DWH_357].[DBO].[Typ_Prostoje]
values (14,'U3', 'Korekce', 1);
insert into [DWH_357].[DBO].[Typ_Prostoje]
values (15,'U4', 'Chybí materiál / obal', 1);
insert into [DWH_357].[DBO].[Typ_Prostoje]
values (16,'U5', 'Chybí personál', 1);
insert into [DWH_357].[DBO].[Typ_Prostoje]
values (17,'U6', 'Neplánovaná vícepráce', 1);
insert into [DWH_357].[DBO].[Typ_Prostoje]
values (18,'U7', 'Organizační chyba', 1);
insert into [DWH_357].[DBO].[Typ_Prostoje]
values (19,'U8', 'Ostatní technické poruchy', 1);
```

Výsledná datová pumpa je zobrazena na obrázku 26. Zdroje dat jsou celkem tři a to tabulka STG_AnalyzaPoruch z dočasného uložště a tabulky Project a Typ_Prostoje z datového skladu.

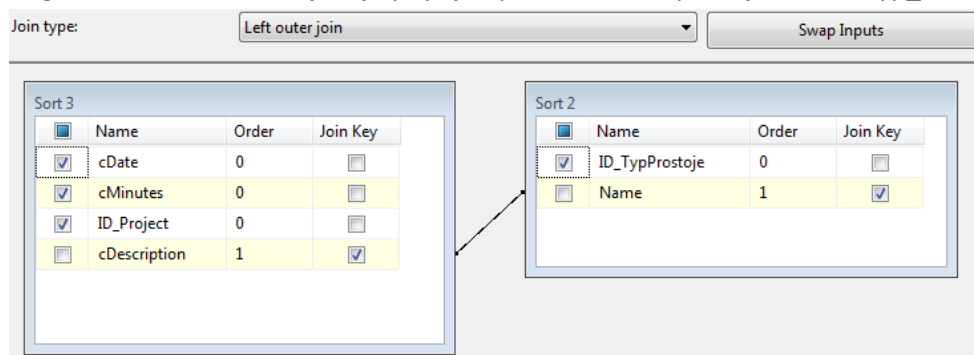


Obrázek 26: Datová pumpa - Načítání dat pro prvek Analýza poruch do datového skladu.
Zdroj: [vlastní]

Bližší popis použitých nástrojů:

- Úprava diakritiky, Úprava diakritiky 2: Z důvodu datové nekvality je potřeba u dvou typů prostoje opravit jejich chybný název. V obou případech se jedná o vynechání diakritiky. Konkrétně chybný název prostoje „Manipulace s materialem“ je nahrazen za „Manipulace s materiálem“ a „Organizacni chyba“ za „Organizační chyba“.
- Sort, Sort 1: Aby bylo možné spojit tabulky STG_AnalyzaPoruch a Project, je potřeba seřadit sloupec, přes který bude spojení probíhat a tím je Name_AnalyzaPoruch.
- Merge Join: Spojení je typu levého vnějšího, stejně jako u datové pumpy prvku ProjectScorecard.
- Sort 2, Sort 3: Opět je potřeba nejprve seřadit data ve sloupci, před tím než je možné dojít ke spojení tabulek. V tomto případě je seřazen sloupec cDescription z datového zdroje STG_AnalyzaPoruch a Name z Typ_Prostoje.

- Merge Join 1: Tímto nástrojem je připojen poslední datový zdroj, tabulka Typ_Prostoje.



Obrázek 27: Merge Join - detail spojení tabulek STG_AnalyzaPoruch, Project s tabulkou Typ_Prostoje.
Zdroj: [vlastní]

- Změna na float: Zde je změněn datový typ sloupce cMinutes na float, čímž je tento sloupec připraven pro následující operaci.
- Převod na hodiny: Je přidán nový sloupec s názvem Hodin, do kterého jsou převedeny hodnoty ze sloupce cMinutes, které jsou vyděleny 60, čímž je získána doba trvání prostoje v hodinách.
- FT_AnalyzaPoruch: SQL příkazem je vytvořena cílová tabulka, do které jsou data nahrána.

```
CREATE TABLE [DWH_357].[dbo].[FT_Analyza_Poruch](
    ID_AnalyzaPoruch int IDENTITY(1,1) PRIMARY KEY,
    ID_Project int FOREIGN KEY REFERENCES Project(ID_Project),
    ID_Typ_Prostoje int FOREIGN KEY REFERENCES TYP_Prostoje
(ID_Typ_Prostoje),
    Hodin float,
    Datum date
);
```

7.3.3 Z-Grafy

Data pro tento prvek jsou nahrávána do dočasného uložení ze souborů typu MS Excel. Pro každý projekt existuje jeden soubor. Je potřeba tedy postupně vytvořit jedenáct spojení typu Excel. Je zadáno umístění souboru, verze Excelu a zda-li první řádek obsahuje popisky. Poté co je vytvořeno spojení, je definováno, jaká data jsou z tohoto zdroje čerpána. V tomto případě je využit SQL dotaz pro Excel, který určuje oblast dat. Pro další úpravy je zahrnut do výběru oblasti dat i jeden prázdný sloupec, který je v záložce Columns přejmenován na ID_Project. Na záložce Columns naopak nejsou zahrnuty do výběru sloupce Šrot /Kč/, Šrot %/ a Šrot %/ SV+ST jelikož se jedná o hodnoty dopočítávané z dat obsažených v téže tabulce. Například pro projekt Škoda Superb CT vypadá detail zdroje, viz obrázek 28.

Excel connection manager:

111_Škoda_Superb_Centralni_trubka

New...

Data access mode:

SQL command

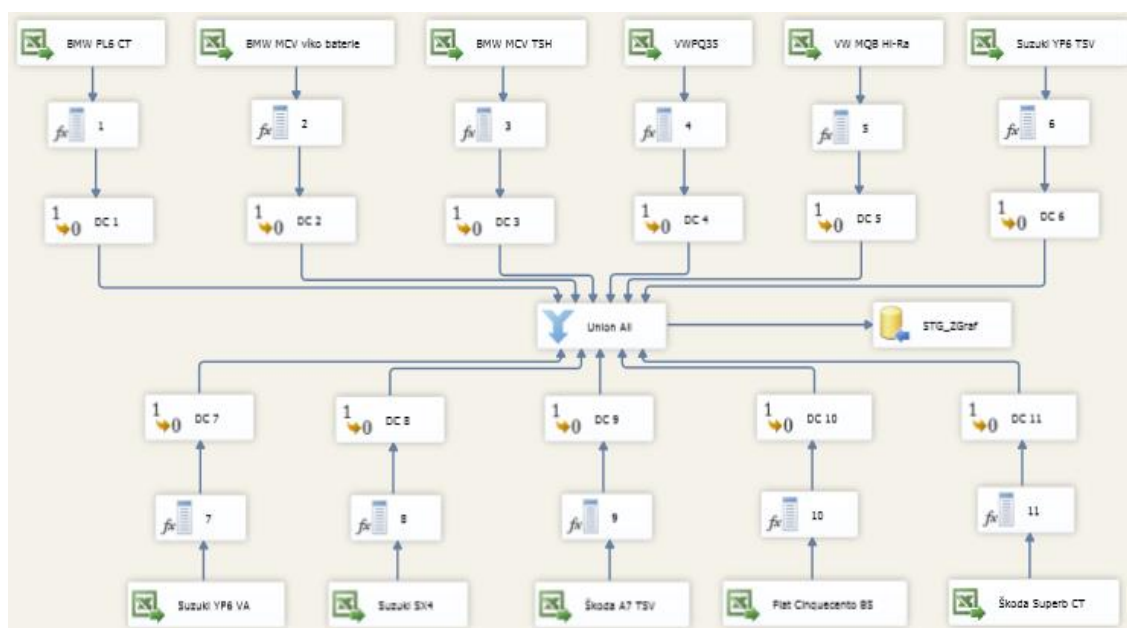
SQL command text:

select * from [zadání dat\$A7:J19];

Parameters...

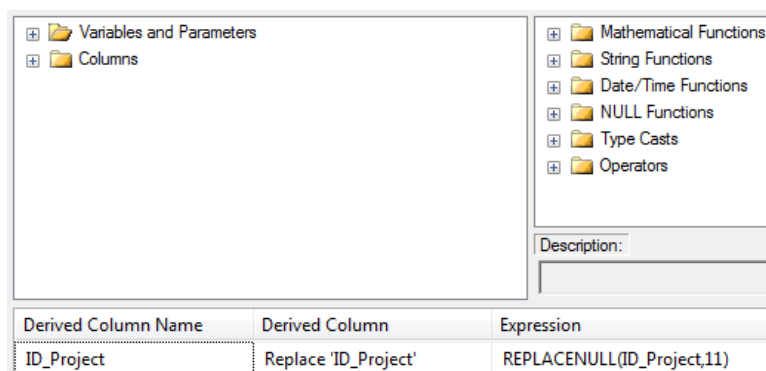
Obrázek 28: Detail definování zdroje dat pro ZGrafy pro projekt Škoda Superb CT.
Zdroj: [vlastní]

Obrázek 29 ilustruje hotovou datovou pumpu. Jednotlivé nástroje jsou blíže rozebrány pod obrázkem.



Obrázek 29: Datová pumpa - Načítání dat pro prvek Z-Grafy do dočasného uložště.
Zdroj: [vlastní]

- 1 (1 - 11): Nejjednodušším způsobem jak přiřadit k jednotlivým tabulkám ID projektu je použít již připravený sloupec ID_Project a nahradit hodnoty null číslem, příslušející ID projektu. Například tedy pro projekt Škoda Superb CT, jsou nahrazeny hodnoty null ve sloupci ID_Project hodnotou 11. Tento krok je zobrazen na obrázku 30. Tento způsob je aplikován na všechny projekty.



Obrázek 30: Nahrazení hodnoty null číslem odpovídajícím ID projektu.

Zdroj: [vlastní]

U některých projektů (konkrétně BMW PL6 ZB Centrální trubka a SUZUKI YP6 Přední nápravnice) je potřeba nahradit hodnoty null ve sloupci Náklady na nekvalitu v TCZK za hodnotu nula, aby bylo možné provádět další výpočty.

- DC 1 (1 - 11): V tomto kroku jsou provedeny změny datových typů u vytvořeného sloupce ID_Project na integer.
- Union All: Všechna vstupní data jsou sjednocena do sloupců jedné tabulky.
- STG_ZGraf: Jako cílová destinace je vytvořena tabulka v dočasném uložišti. Tabulku je možné jednoduše vytvořit přes tlačítko New, jelikož nejsou třeba žádné další úpravy.

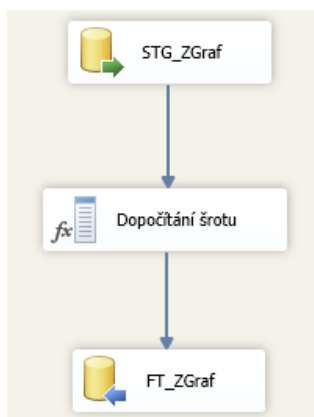
Jakmile jsou data nahrána v dočasném uložišti, je možné se pustit do konstrukce datové pumpy, která tyto data transformuje a nahraje do datového skladu. Zdrojem dat je tabulka STG_ZGraf z dočasného uložistě. Pro dopočítání sloupců se šroty, je použit nástroj Derived Column (ve výsledné datové pumpě přejmenován na Dopčítání šrotu) a jsou definovány vzorce pro výpočty. Tento krok znázorňuje obrázek 31. Po té již zbývá vytvořit cílovou tabulku FT_ZGraf v datovém skladu. To je provedeno SQL příkazem:

```
CREATE TABLE [DWH_357].[dbo].[FT_ZGraf](
  ID_ZGraf int IDENTITY(1,1) PRIMARY KEY,
  ID_Project int FOREIGN KEY REFERENCES Project(ID_Project),
  Datum date,
  [Sériové tržby Kč] float,
  [SV Kč] float,
  [ST Kč] float,
  [Šrot Kč] float,
  [Šrot] float,
  [Šrot SV+ST] float,
  [Náklady na nekvalitu v TCZK] float,
  [Cíl] float
);
```

| <div> <div> + Variables and Parameters + Columns </div> <div> + Mathematical Functions + String Functions + Date/Time Functions + NULL Functions + Type Casts + Operators </div> </div> <div>Description:</div> | | |
|---|---------------------|--|
| Derived Column Name | Derived Column | Expression |
| Šrot Kč | <add as new column> | [SV /Kč/] + [ST /Kč/] |
| Šrot % | <add as new column> | [SV /Kč/] / [Seriové tržby /Kč/] |
| Šrot SV+ST v % | <add as new column> | ([SV /Kč/] + [ST /Kč/]) / [Seriové tržby /Kč/] |

Obrázek 31: Dopotání šrotu - detail nástroje Derived column.
Zdroj: [vlastní]

Výsledná datová pumpa je zachycena na obrázku 32.



Obrázek 32: Datová pumpa - načtení dat pro prvek ZGrafy do datového skladu.
Zdroj: [vlastní]

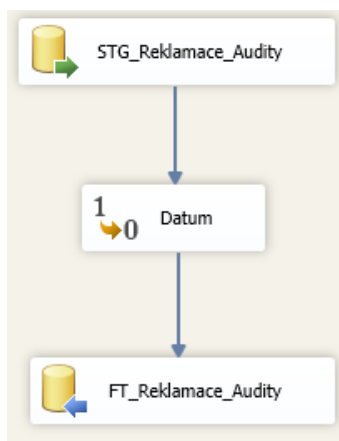
Jak již bylo dříve zmíněno, prvek Z-Graf je složen z více částí. Jednou z nich je část reklamace a audity. Data pro tuto část jsou uložena ve stejných souborech jako pro Z-Grafy. Pro zachování přehlednosti jsou vytvořeny zvláštní datové pumpy.

Pro načtení dat do dočasného uložení jsou využity stejné nástroje jako u Z-Grafů (je možné je zkopírovat), pouze je upraveno spojení tak, aby oblast zdrojových dat byla na data týkající se reklamací a audit. Tento krok zobrazuje obrázek 33. U nástrojů 1 a 7 je odstraněna funkce pro nahrazení hodnoty null ve sloupci Náklady na nekvalitu v TCZK, jelikož zde tento sloupec není. Dále ještě zbývá upravit cílovou destinaci, což je tabulka STG_Reklamace_Audity v dočasném uložení, která je vytvořena opět tlačítkem New. Po těchto jednoduchých úpravách, je datová pumpa pro načtení dat o reklamacích a auditech do dočasného uložení, hotová.

Obrázek 33: Detail definování zdroje dat pro reklamace a audity pro projekt Škoda Superb CT.
Zdroj: [vlastní]

Nyní je vytvořena datová pumpa, která nehraje data z dočasného uložště do datového skladu. Zdrojem je tedy tabulka STG_Reklamace_Audity. Jedinou úpravou, která je potřeba je změna datového typu u sloupce Měsíc/Rok na date. Cílovou destinací upravených dat je tabulka s názvem FT_Reklamace_Audity, která je vytvořena SQL příkazem:

```
CREATE TABLE [DWH_357].[DBO].[FT_Reklamace_Audity](
  ID_ReklamaceAudity int IDENTITY(1,1) PRIMARY KEY,
  ID_Project int FOREIGN KEY REFERENCES Project(ID_Project),
  Datum date,
  [Vydované kusy] float,
  [Počet reklamací] int,
  [Počet reklam kusů] int,
  [Audit výrobku] float,
  [Audit výrobku-cíl] float,
  [PPM] float
);
```

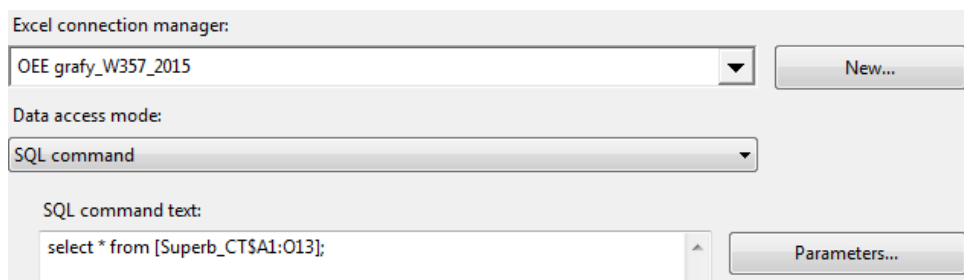


Obrázek 34: Datová pumpa – načtení dat pro reklamace a audity do datového skladu.
Zdroj: [vlastní]

7.3.4 OEE, HIP

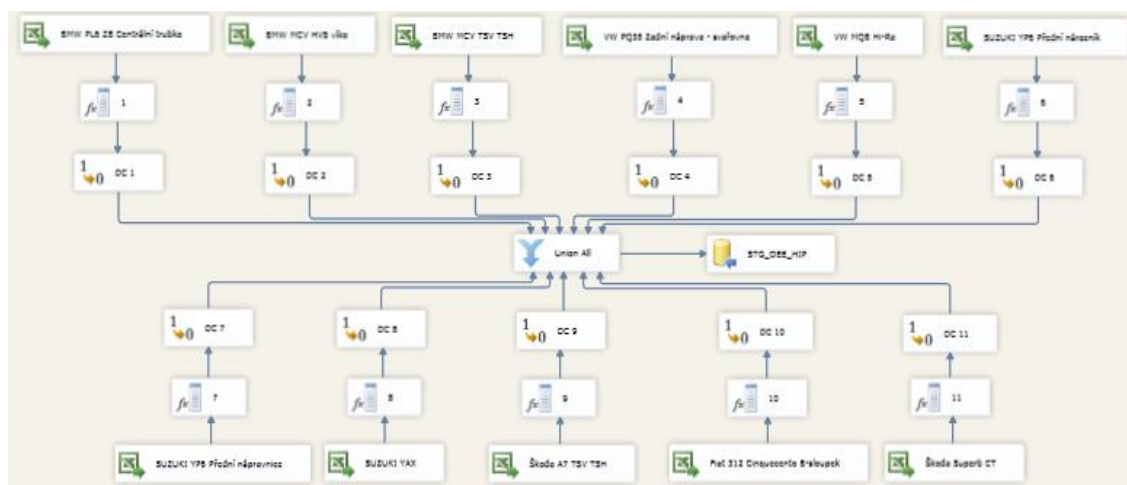
K nahrání dat do dočasného uložště je zkonstruována datová pumpa, kde zdrojem dat je soubor MS Excel, je tudíž nutné vytvořit nové spojení. Poté je použit SQL příkaz k definování oblasti dat, viz obrázek 35. Stejně jako například u prvku Z-Grafy je zahrnut do výběru oblasti dat jeden prázdný sloupec, který je v záložce Columns přejmenován na ID_Project a nejsou zahrnuty sloupce Qualitätsfaktor (faktor kvality), Verfügbarkeitsfaktor (faktor dostupnosti), Leistungsfaktor (faktor výkonu), IST OEE (skutečné OEE) a IST HIP (skutečné HIP). Tyto sloupce

totíž obsahují hodnoty vypočítané z ostatních sloupců. Takto je vytvořen zdroj pro každý projekt.



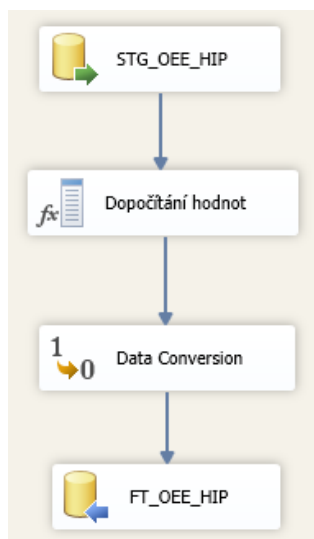
Obrázek 35: Detail definování zdroje dat pro OEE, HIP pro projekt Škoda Superb CT.
Zdroj: [vlastní]

Dále je postupováno obdobně, jako když byla vytvářena pumpa pro prvek Z-Grafy. Přes nástroj Derived Column (1-11) jsou nahrazeny pro sloupec ID_Project hodnoty null za konkrétní číslo projektu podle ID. Následně díky nástroji Data Conversion (DC 1 – DC 11) je změněn datový typ pro tento sloupec na integer. Všechny zdroje jsou spojeny do nástroje Union All a poté napojeny na cílové umístění, což je nová tabulka STG_OEE_HIP, která je vytvořena opět přes tlačítko New. Výsledná pumpa je zobrazena na obrázku 36.



Obrázek 36: Datová pumpa - načtení dat pro prvek OEE, HIP do dočasného uložště.
Zdroj: [vlastní]

Nyní je přistoupeno k transformaci a načtení dat z dočasného uložště do datového skladu. Jako zdroj zde slouží tabulka STG_OEE_HIP.



Obrázek 37: Datová pumpa - načtení dat pro prvek OEE, HIP do datového skladu.
Zdroj: [vlastní]

- **Dopočítání hodnot:** Tento nástroj umožňuje dopočítat hodnoty ke sloupcům. Použité vzorce jsou vidět na obrázku 38. Nově vytvořené sloupce jsou pojmenovány česky. Sloupec OEE, jehož sloupec není na obrázku kompletní, je součin Faktoru kvality * Faktor dostupnosti * Faktoru výkonu.

| <div> <div> Variables and Parameters Columns </div> <div> Mathematical Functions String Functions Date/Time Functions NULL Functions Type Casts Operators </div> </div> <div>Description:</div> | | |
|---|---------------------|---|
| Derived Column Name | Derived Column | Expression |
| Faktor kvality | <add as new column> | [[#O# Stückzahl (Stck)] / [Bruttostückzahl (Stck)] |
| Faktor dostupnosti | <add as new column> | [Produktive Zeit (min)] / [Anlagenlaufzeit nach BOS (min)] |
| Faktor výkonu | <add as new column> | [Produktive Anlagenlaufzeit Brutto (min)] / [Produktive Zeit (min)] |
| OEE | <add as new column> | ((([#O# Stückzahl (Stck)] / [Bruttostückzahl (Stck)]) * ([Produktive Z |
| HIP | <add as new column> | [[#O# Stückzahl (Stck)] / [Hodiny HR] |

Obrázek 38: Detail - vzorce pro dopočítání sloupců pro prvek HIP, OEE. Detail nástroje Derived Column.
Zdroj:[vlastní]

- **Data Conversion:** Změna datového typu sloupce Měsíc na date.
- **FT_OEE_HIP:** Cílovou destinací, kam jsou tato upravená data nahrána je tabulka v datovém skladu s názvem FT_OEE_HIP, která je vytvořena SQL příkazem:

```

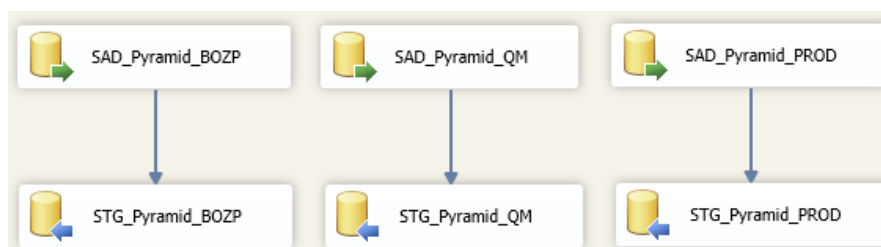
CREATE TABLE [DWH_357].[dbo].[FT_OEE_HIP](
  [ID_OEE_HIP] int IDENTITY(1,1) PRIMARY KEY,
  [ID_Project] int FOREIGN KEY REFERENCES Project(ID_Project),
  [Datum] date,
  [Ks] float,
  [Brutto Ks] float,
  [Výrobní čas (min)] float,
  [Strojní čas BOS] float,
  [Celkový strojní čas] float,
  [Faktor kvality] float,
  [Faktor dostupnosti] float,

```

```
[Faktor výkonu] float,
[OEE] float,
[Cíl OEE] float,
[Hodiny HR] float,
[HIP] float,
[Cíl HIP] float
);
```

7.3.5 Pyramida BOZP, Pyramida kvality a Pyramida produkce

Vzhledem k tomu, že byly tabulky pyramid navrženy speciálně pro tyto účely, zkonstruovat datové pumpy jsou velmi jednoduché. Jako zdroje jsou použity tabulky z SQL databáze systému SAD. Každá pyramida má svou tabulku. Po té již stačí určit cílové tabulky v dočasném uložšti, které jsou vytvořeny přes tlačítko New.



Obrázek 39: Datové pumpy - Načtení dat pro prvky pyramid do dočasného uložště.
Zdroj: [vlastní]

Nahrání dat z dočasného uložště do datového skladu je také velmi jednoduchý proces, jelikož není potřeba žádná transformace dat, je pouze změněno jméno řádku a upraven mapping. Zdroje jsou tabulky z dočasného uložště: STG_Pyramid_BOZP, STG_Pyramid_QM a STG_Pyramid_PROD. Tyto zdroje, jak již bylo řečeno, jsou napojeny bez jakékoli transformace na cílové destinace. Pro ty, vytvořím tabulky SQL příkazem:

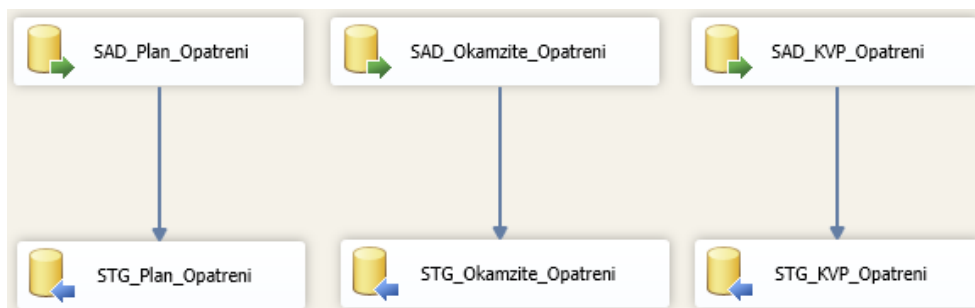
```
CREATE TABLE [DWH_357].[DBO].[Pyramid_BOZP] (
    ID_PyramidBOZP int PRIMARY KEY,
    ID_Project int FOREIGN KEY REFERENCES Project(ID_Project),
    Datum date,
    Stav nvarchar(1)
);

CREATE TABLE [DWH_357].[DBO].[Pyramid_PROD] (
    ID_PyramidPROD int PRIMARY KEY,
    ID_Project int FOREIGN KEY REFERENCES Project(ID_Project),
    Datum date,
    Stav nvarchar(1)
);

CREATE TABLE [DWH_357].[DBO].[Pyramid_QM] (
    ID_PyramidQM int PRIMARY KEY,
    ID_Project int FOREIGN KEY REFERENCES Project(ID_Project),
    Datum date,
    Stav nvarchar(1)
);
```

7.3.6 Plán opatření

Stejně jako u prvků pyramid, nebude pro Plán opatření složité sestavit datové pumpy. Postup je prakticky stejný jako právě u pyramid. Pumpa se skládá ze tří zdrojů, protože formulář je rozložen do tří tabulek, které jsou svázány cizími klíči. Cílové tabulky jsou opět vytvořeny přes tlačítko New.



Obrázek 40: Datová pumpa - načtení dat pro prvek Plán opatření do dočasného uložiště.
Zdroj: [vlastní]

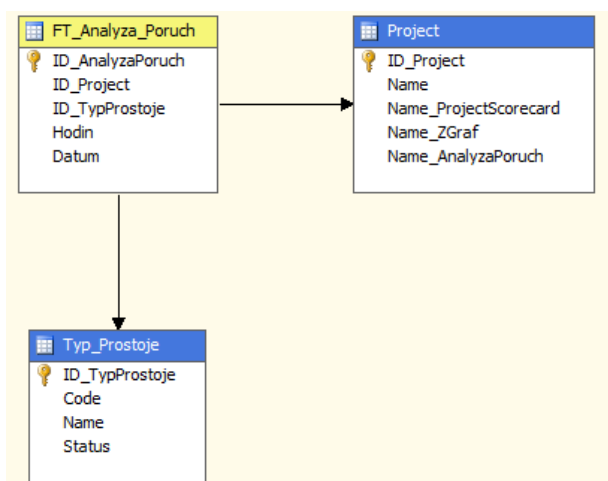
Podobně jsou nahrána i data z dočasného uložiště do datového skladu. Zdrojové tabulky jsou: STG_Plan_Opatreni, STG_Okamzite_Opatreni a STG_KVP_Opatreni. Není potřeba žádná transformace dat, a proto jsou napojeny zdroje přímo na cílové tabulky, které jsou vytvořeny SQL příkazem:

```
CREATE TABLE [DWH_357].[dbo].[Plan_opatreni](
[ID_PlanOpatreni] int PRIMARY KEY,
[ID_Project] int FOREIGN KEY REFERENCES Project(ID_Project),
[Datum] date,
[Směna] nvarchar(1),
[Popis] nvarchar(1000),
[Příčina] nvarchar(1000),
[ID_OkamziteOpatreni] int FOREIGN KEY REFERENCES
Okamzite_Opatreni(ID_PlanOpatreni),
[ID_KVPopatreni] int FOREIGN KEY REFERENCES KVP_Opatreni(ID_KVPopatreni),
);
CREATE TABLE [DWH_357].[dbo].[Okamzite_opatreni](
[ID_OkamziteOpatreni] int) PRIMARY KEY,
[Popis] nvarchar(1000),
[Odpovědný] nvarchar(50),
[Termín] date,
[Status] int
);
CREATE TABLE [DWH_357].[dbo].[KVP_opatreni](
[ID_KVPopatreni] int PRIMARY KEY,
[Popis] nvarchar(1000),
[Odpovědný] nvarchar(50),
[Termín] date,
[Status] int
);
```

7.4 OLAP kostky

Jakmile jsou data nahrána v datovém skladu, je možné začít vytvářet datové kostky. K tomu bude využit stejný program jako při ETL procesu, tedy MS Visual Studio.

Je tedy založen nový multidimenzionální projekt. Nejprve je definován nový datový zdroj (Data Source). V tomto kroku je vytvořeno spojení s datovým skladem. Dále jsou definovány nové datové pohledy pro jednotlivé prvky. V pohledech jsou vybrány tabulky, které souvisí s daným prvkem. Například pro prvek Analýza poruch jsou vybrány tabulky FT_Analyza_Poruch, Typ_Prostoje a Project. V dalším textu je postup ilustrován na prvku Analýza poruch. Na základě tabulek z datového pohledu, jsou vytvořeny potřebné dimenze. Těmi jsou pro Analýzu poruch dimenze Project, Typ prostoje a Čas. Dimenze Čas, je vygenerována programem, na server. Po té je již možné vytvořit OLAP kostku, pro kterou je definován datový pohled, tabulky faktů a dimenze. Dále je možnost definovat různé výpočty. Vzhledem k tomu, že však byly potřebné výpočty provedeny již v procesu ETL, není potřeba definovat další. Struktura OLAP kostky pro Analýzu poruch ukazuje obrázek 41. Obdobným způsobem jsou vytvořeny OLAP kostky i pro ostatní prvky.

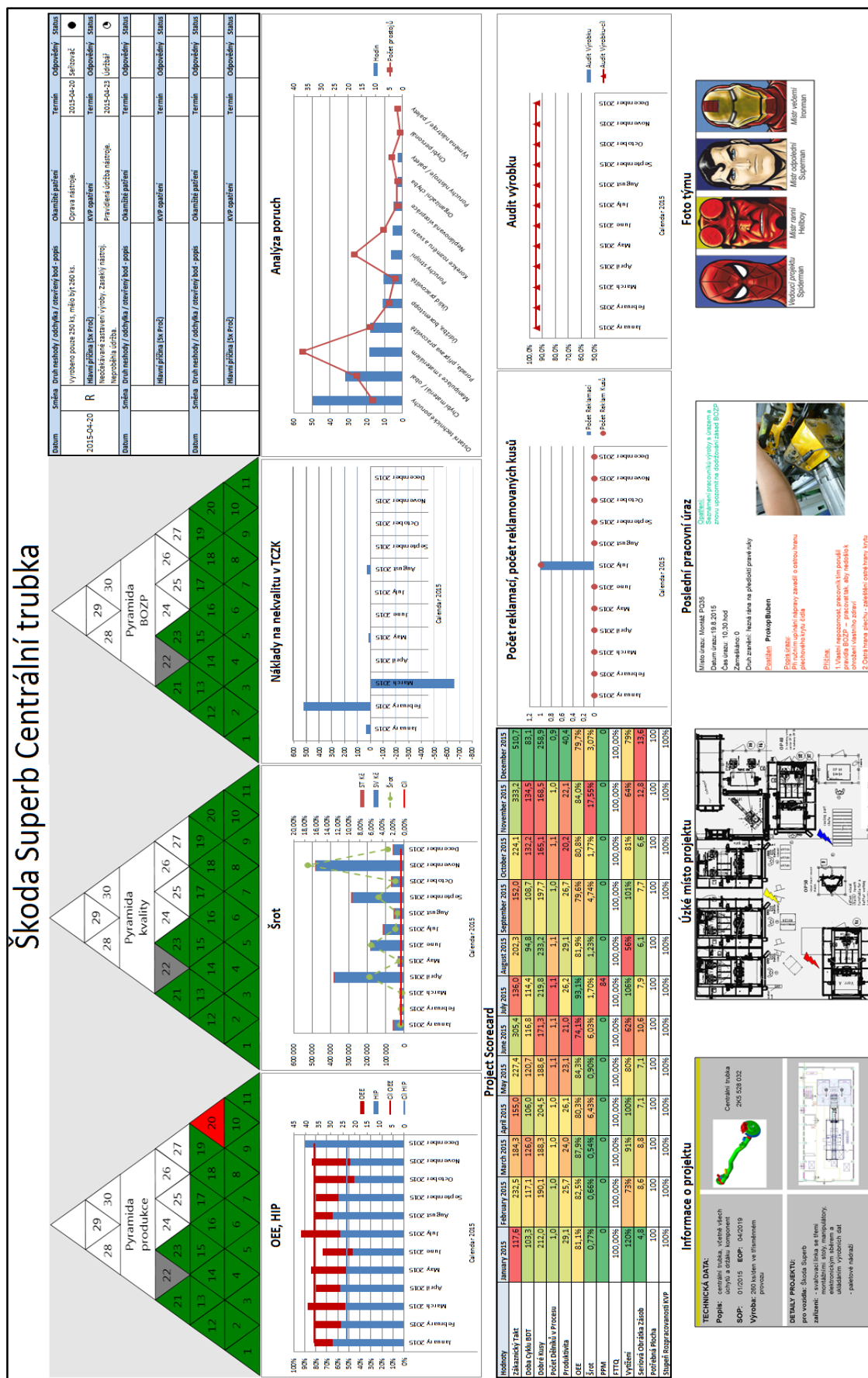


Obrázek 41: OLAP kostka - struktura kostky pro prvek Analýza poruch.
Zdroj: [vlastní]

8 Projektová nástěnka – dashboard

V této kapitole se konečně dostává k výstupům, které jsou viditelné pro uživatele. Program, který slouží pro zobrazení výsledného dashboardu, byl vybrán MS Excel, jelikož tím nevzniknou větší náklady. Firma totiž MS Office licencuje na základě multilicence, čímž je cena velmi příznivá. Zároveň nebrání nasazení nového řešení žádné žádné byrokratický proces, při kterém musí dojít ke schválení, výběru dodavatele, objednání, atd. Zároveň na řešení není kladena žádná funkcionalita, kterou by MS Excel nezvládl.

Soubor Excel, který je určen pro dashboard, obsahuje několik listů. Prakticky pro každý prvek skládající se z dat, načtená z SQL serveru, je vyčleněn jeden list, kde jsou data pro vizualizované prvky. Tyto prvky jsou následně uspořádány na list Dashboard, kde jsou všechny prvky uspořádány do finálního dashboardu. Data jsou načtena do Excelu pomocí nového připojení ze služeb pro analýzu, kde jsou uloženy OLAP kostky. Jednotlivé prvky jsou uspořádány a navrhnuty tak, aby neobsahovaly chyby zmíněné v kapitole 4.1 Základní chyby návrhu dashboardu. Výsledný dashboard je zobrazen na obrázku 42. Jednotlivé prvky dashboardu se automaticky na začátku dne aktualizují. Stáří dat je tedy maximálně jeden den, s čímž je zadavatel obeznámen.



Obrázek 42: Dashboard pro projekt Škoda Superb Centrální trubka.
Zdroj: [vlastní]

9 Vlastní přínos

- Vytvořil jsem automatizované řešení vizualizace projektových nástěnek pro konkrétní podnik.
- Toto řešení přináší finanční i časovou úsporu.
- Digitalizoval jsem data, která byla doposud zaznamenávána pouze papírovou formou.
- Zasadil jsem o zlepšení datové kvality.
- Vybudoval jsem datový sklad, ve kterém jsou data připravena pro další analýzy a porovnání projektů mezi sebou a v čase.
- Posunul jsem vizualizaci dat dalším směrem a vylepšil jsem metody vizualizace.
- Nové digitální nástěnky působí dobře na auditory a návštěvy zákazníků, čímž přispívají ke zlepšování dobrého jména firmy.
- Projektové nástěnky pro všechny projekty lze spravovat z jednoho místa.
- Pro nahlédnutí na projektovou nástěnku nemusí uživatel chodit do výrobních prostorů. Může tedy stav sledovat i z domova či služební cesty.

Závěr

Cílem této diplomové práce bylo vytvořit a popsat řešení, které automatizuje a vizualizuje projektové nástěnky v konkrétním podniku. Ke splnění cíle jsem postupoval v několika krocích. V kapitole 5 jsem provedl analýzu požadavků a dále jsem navrhl základní architekturu řešení. V kapitole 6 jsem provedl analýzu dat a definoval jsem zdrojová data, která budou pro řešení využita. Sedmá kapitola patří k těm nejrozsáhlejším, jelikož se zabývá návrhem datovým uložištěm, tvorbou OLAP kostek a detailně popisuje proces ETL. Hlavním výstupem pro běžného uživatele je až kapitola 8, ve které se nachází dashboard pro konkrétní projekt.

Cíl práce byl splněn, což ostatně dokazuje kapitola 8, konkrétně obrázek 42. Mnohdy byla největší překážkou při řešení špatná datová kvalita a také určitá svázanost firemními standardy. Řešení bylo navrženo tak, aby co nejvíce splňovalo požadavky zadavatele.

Jelikož se tato diplomová práce zabývá reálným problémem, bude toto řešení v blízké době implementováno v podniku. Tomu však musí předcházet zkušební provoz, při kterém dojde k ladění případných problémů a dalších požadavků. Dále je zapotřebí proškolení uživatelů, kteří budou zadávat data přes vytvořenou aplikaci pro pyramidy BOZP, kvality a produkce a také pro formulář pro plán opatření. Dále budou také definováni lidé, kteří nesou odpovědnost za zdrojová data a budou proškoleni, jak tato data spravovat. Jedná se především o zdrojová data ze souboru MS Excel. Není pochyb o tom, že dokumenty Excel nejsou vhodnými zdroji. V mém dalším působení ve firmě, se pokusím připravit půdu pro zavedení systému, který by nahradil soubory Excel v plném rozsahu. Nezačínal bych ani na „zelené louce“, jelikož už ve firmě jakýsi program pro sběr dat je a je vyvíjen zaměstnancem podniku. Pevně věřím, že ve spolupráci s tímto člověkem bychom mohli všechny data sbírat do SQL databáze, kde by byla připravena pro další využití. Je to však běh na dlouhou trať, jelikož se jedná o velkou nadnárodní společnost, kde se takovéto změny, ač k lepšímu, provádějí velmi složitě.

Seznam literatury:

- [1] LANS, Rick F. Data virtualization for business intelligence systems: revolutionizing data integration for data warehouses. Amsterdam: Elsevier, Morgan Kaufmann, 2012. ISBN 9780123944252.
- [2] KREČMEROVÁ, Petra. *Srovnání vybraných komerčních reportingových BI platforem*. Praha, 2015. Diplomová práce. Vysoká škola ekonomická v Praze. Vedoucí práce Novotný Ota.
- [3] SOCHORA, David. *Automatizace zpracování finančních výkazů*. Praha, 2015. Diplomová práce. Mendelova univerzita v Brně. Vedoucí práce Turčínek Pavel.
- [4] ČERNÝ, Lukáš. *Implementace Business intelligence řešení v podniku*. Praha, 2011. Diplomová práce. Vysoká škola ekonomická v Praze. Vedoucí práce Slánský David.
- [5] NOVOTNÝ, Ota, Jan POUR a David SLÁNSKÝ. Business intelligence: jak využít bohatství ve vašich datech. 1. vyd. Praha: Grada, 2005. Management v informační společnosti. ISBN 80-247-1094-3
- [6] MOSS, Larissa Terpeluk a S ATRE. Business intelligence roadmap: the complete project lifecycle for decision-support applications. 1. vyd. Boston, MA: Addison-Wesley, c2003. Management v informační společnosti. ISBN 02-017-8420-3.
- [7] INMON, William H. Building the data warehouse. 3rd ed. New York: J. Wiley, c2002. ISBN 0471081302.
- [8] FEW, Stephen. *Information dashboard design: the effective visual communication of data*. 1st ed. Cambridge [MA]: O'Reilly, c2006. ISBN 9780596100162.
- [9] PATOČKA, Miroslav. *OEE a odvozené ukazatele TEEP, PEE, OAE, OPE, OFE, OTE a CTE* [online]. 2013 [cit. 2016-04-07]. Dostupné z: <http://www.mescentrum.cz/clanky/mes-mom/133-oeo>.
- [10] WIMMER, Jan. Chrastava, 2016.

Seznam obrázků:

| | |
|---|----|
| Obrázek 1: Princip multidimenzionální databáze | 23 |
| Obrázek 2: Architektura s datovým skladem | 26 |
| Obrázek 3: Architektura bez datového skladu | 26 |
| Obrázek 4: Architektura s dočasným uložištěm dat. | 27 |
| Obrázek 5: Architektura s operativním uložištěm dat. | 28 |
| Obrázek 6: Ukázka špatného dashboardu: vše není zobrazeno na první pohled. | 33 |
| Obrázek 7: Ukázka špatného dashboardu: Nepřehledná forma. | 34 |
| Obrázek 8: Ukázka správného dashboardu. | 34 |
| Obrázek 9: Chyba: Hodnoty nezačínají od nuly. | 35 |
| Obrázek 10: Chyba: Všechny objekty jsou stejně výrazné. | 36 |
| Obrázek 11: Chyba: Dashboard obsahuje bezvýznamnou dekoraci. | 37 |
| Obrázek 12: Ukázka papírové projektové nástěnky. | 39 |
| Obrázek 13: Popis a rozložení prvků v projektové nástěnce. | 40 |
| Obrázek 14: Návrh architektury. | 42 |
| Obrázek 15: Vizualní formulář pyramid. | 44 |
| Obrázek 16: Ukázka zdrojových dat pro Z-Grafy. | 46 |
| Obrázek 17: Ukázka zdrojových dat pro ProjectScorecard. | 47 |
| Obrázek 18: Ukázka dat upravených skriptem. | 47 |
| Obrázek 19: Ukázka zdrojových dat pro OEE, HIP. | 48 |
| Obrázek 20: Dočasné uložiště dat. | 49 |
| Obrázek 21: Schéma databáze datového skladu. | 50 |
| Obrázek 22: Manažer připojení. | 52 |
| Obrázek 23: Datová pumpa - nahrávání dat ze souboru CSV. | 52 |
| Obrázek 24: Datová pumpa - Načtení dat pro prvek ProjectScorecard do datového skladu. | 55 |
| Obrázek 25: Merge join - spojení zdrojů STG_ProjectScorecard_2 a Project. | 56 |
| Obrázek 26: Datová pumpa - Načítání dat pro prvek Analýza poruch do datového skladu. | 59 |
| Obrázek 27: Merge Join - detail spojení tabulek STG_AnalyzaPoruch, Project s tabulkou Typ_Prostoj. | 60 |
| Obrázek 28: Detail definování zdroje dat pro ZGrafy pro projekt Škoda Superb CT. | 61 |
| Obrázek 29: Datová pumpa - Načítání dat pro prvek Z-Grafy do dočasného uložiště. | 61 |
| Obrázek 30: Nahrazení hodnoty null číslem odpovídajícím ID projektu. | 62 |
| Obrázek 31: Dopočítání šrotu - detail nástroje Derived column. | 63 |
| Obrázek 32: Datová pumpa - načtení dat pro prvek ZGrafy do datového skladu. | 63 |
| Obrázek 33: Detail definování zdroje dat pro reklamace a audity pro projekt Škoda Superb CT. | 64 |
| Obrázek 34: Datová pumpa – načtení dat pro reklamace a audity do datového skladu. | 64 |
| Obrázek 35: Detail definování zdroje dat pro OEE, HIP pro projekt Škoda Superb CT. | 65 |
| Obrázek 36: Datová pumpa - načtení dat pro prvek OEE, HIP do dočasného uložiště. | 65 |
| Obrázek 37: Datová pumpa - načtení dat pro prvek OEE, HIP do datového skladu. | 66 |
| Obrázek 38: Detail - vzorce pro dopočítání sloupců pro prvek HIP, OEE. Detail nástroje Derived Column. | 66 |
| Obrázek 39: Datové pumpy - Načtení dat pro prvky pyramid do dočasného uložiště. | 67 |

| | |
|--|----|
| Obrázek 40: Datová pumpa - načtení dat pro prvek Plán opatření do dočasného uložistě. | 68 |
| Obrázek 41: OLAP kostka - struktura kostky pro prvek Analýza poruch..... | 69 |
| Obrázek 42: Dashboard pro projekt Škoda Superb Centrální trubka. | 71 |

Seznam tabulek:

| | |
|--|----|
| Tabulka 1: Porovnání OLTP a OLAP technologií..... | 22 |
| Tabulka 2: Porovnání datových uložišť. | 29 |
| Tabulka 3: Tabulka projektů - konsolidace názvů..... | 43 |