

**Univerzita Hradec Králové**  
**Fakulta informatiky a managementu**  
**Katedra informatiky a kvantitativních metod**

**Self-service Business Intelligence**

Bakalářská práce

Autor: Tomáš Karásek  
Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Barbora Tesařová, Ph.D.

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně pod vedením vedoucího bakalářské práce a s použitím uvedené literatury a dalších odborných zdrojů.

V Hradci Králové dne 30.4.2020

Jméno a Příjmení

#### Poděkování:

Děkuji vedoucímu bakalářské práce Ing. Barboře Tesařové, Ph.D. za čas, trpělivost a spoustu rad věnovaných této práci. Děkuji také vedení zemědělského podniku za poskytnutá data pro zpracování praktické části.



## **Anotace**

Cílem práce je bude vymezení pojmu Self-service Business Intelligence a její komponent. V teoretické části dojde k představení základních pojmů týkajících se Self-service Business Intelligence, pojmů vizualizace a seznámení s technologií Power BI. V praktické části bude vypracováno BI řešení založené na datech z reálného prostředí. Data budou pocházet z prostředí pro pěstování a sklizně plodin. Bude zde znázorněn celý postup od transformací holých dat, až po konečné vizuální sestavy. Transformace budou vysvětleny na skriptech jazyka T-SQL a vizualizace budou provedeny v prostředí Power BI za pomoci jazyku DAX. Celkem budou vytvořeny 3 vizualizační sestavy, ve kterých budou demonstrovány průřezy a filtry pro každou hlavní dimenzi zvlášť.

## **Annotation**

### **Title: Self-service Business Intelligence**

The aim of the work is to define the concept of Self-service Business Intelligence and its components. In the theoretical part, there will be an introduction to the basic concepts related to Self-service Business Intelligence, the concepts of visualization and introduction to Power BI technology. In the practical part, there will be a BI solution to a problem based on real life data. The data focuses on the subject of growing and harvesting crops. The whole process, from the transformation of bare data to the final visual assembly, will be shown. Transformations will be explained in T-SQL scripts and visualizations will be performed in Power BI using DAX. In total, 3 visualization sets will be created, in which the cross-sections and filters for each main dimension will be demonstrated separately.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Cíl práce a metodika zpracování</b>	<b>2</b>
<b>3</b>	<b>Teoretická část</b>	<b>3</b>
<b>3.1</b>	<b>Business Intelligence</b>	<b>3</b>
3.1.1	Historie Business Intelligence	3
3.1.2	Definice BI	4
<b>3.2</b>	<b>Datové sklady</b>	<b>4</b>
3.2.1	Definice	5
3.2.2	Transakční vs. analytické databáze	5
3.2.3	Datová tržiště	6
<b>3.3</b>	<b>Datový model</b>	<b>7</b>
3.3.1	Star schéma	7
3.3.2	Snowflake schéma	8
<b>3.4</b>	<b>ETL</b>	<b>8</b>
3.4.1	Extrakce	9
3.4.2	Transformace	9
3.4.3	Načtení	10
<b>3.5</b>	<b>OLAP</b>	<b>10</b>
<b>3.6</b>	<b>Dolování dat</b>	<b>11</b>
3.6.1	Průběh procesu dolování dat	12
<b>3.7</b>	<b>Vizualizace dat</b>	<b>12</b>
3.7.1	Tabulky	14
3.7.2	Grafy	14
3.7.2.1	Spojnicový graf (Line Chart)	15
3.7.2.2	Sloupcový graf	15
3.7.2.2.1	Pruhový graf	16
3.7.2.2.2	Dvojitý sloupcový graf	17
3.7.2.3	Výšečový graf (Pie Chart)	17
3.7.2.3.1	Prstencový graf	18
3.7.2.4	Bodový graf (Scatter plot)	19
3.7.2.5	Bublinový graf	20
3.7.2.6	Krabicový graf (Box Plot)	21
3.7.3	Scorecard	22
3.7.4	Dashboard	23

<b>3.8</b>	<b>Self-service Business Intelligence .....</b>	<b>24</b>
3.8.1	Výhody a nevýhody SSBI.....	25
3.8.2	Power Pivot.....	26
3.8.2.1	Power Pivot pro Excel .....	26
3.8.2.2	Power BI .....	27
3.8.3	DAX .....	28
3.8.3.1	Calculated Column .....	29
3.8.3.2	Measure .....	29
3.8.3.3	Funkce CALCULATE.....	29
<b>4</b>	<b>Praktická část .....</b>	<b>31</b>
<b>4.1</b>	<b>Představení projektu .....</b>	<b>31</b>
<b>4.2</b>	<b>Návrh a analýza řešení.....</b>	<b>31</b>
<b>4.3</b>	<b>ETL.....</b>	<b>33</b>
4.3.1	Fakta Sklizeň .....	33
4.3.2	Dimenze Hony .....	41
4.3.3	Dimenze Plodiny .....	44
4.3.4	Fakta Sklizeň ČSÚ .....	45
4.3.5	Načtení dat do Power BI.....	48
4.3.6	Dimenze Čas .....	48
4.3.7	Relace v modelu .....	49
<b>4.4</b>	<b>Tvorba measures .....</b>	<b>51</b>
<b>4.5</b>	<b>Vizualizace dat.....</b>	<b>51</b>
4.5.1	Vizualizační komponenty.....	51
4.5.1.1	Skupinový sloupcový graf.....	52
4.5.1.2	Spojnicový graf.....	52
4.5.1.3	Spojnicový a skládaný graf .....	53
4.5.1.4	KPI .....	53
4.5.1.5	Matice .....	54
4.5.1.6	Mapa .....	54
4.5.2	Dashboardy.....	55
<b>4.6</b>	<b>Shrnutí .....</b>	<b>58</b>
<b>5</b>	<b>Závěr .....</b>	<b>59</b>
<b>6</b>	<b>Seznam zdrojů.....</b>	<b>60</b>
6.1	Tištěné zdroje.....	60
6.2	Online zdroje.....	61

## Seznam obrázků

<i>Obr. 1: Ukázka ukládání dat v relačních databázích do řádků a v analytických do sloupců .....</i>	<i>6</i>
<i>Obr. 2: Star vs snowflake schema.....</i>	<i>8</i>
<i>Obr. 3: Příklad návrhu OLAP kostky.....</i>	<i>11</i>
<i>Obr. 4: Ukázka pruhového a dvojitého pruhového grafu .....</i>	<i>17</i>
<i>Obr. 5: Srovnání vzhledu koláčového a sloupcového grafu pro stejná data .....</i>	<i>18</i>
<i>Obr. 6: Bublinový graf aplikovaný na mapě .....</i>	<i>20</i>
<i>Obr. 7: Zjednodušená verze krabicového grafu vhodná pro dashboard .....</i>	<i>22</i>
<i>Obr. 8: Ukázka scorecardu.....</i>	<i>23</i>
<i>Obr. 9: Ukázka dashboardu .....</i>	<i>24</i>
<i>Obr. 10: Ukázka vizuálního rozhraní- (nalevo) Power Pivot pro Excel a (napravo) Power BI Desktop .....</i>	<i>28</i>
<i>Obr. 11: Ukázka návrhu podle schématu snowflake .....</i>	<i>32</i>
<i>Obr. 12: Ukázka návrhu podle schématu star .....</i>	<i>32</i>
<i>Obr. 13: Struktura sešitů a souboru sklizní .....</i>	<i>33</i>
<i>Obr. 14: Struktura sešitů s hony a katastry .....</i>	<i>41</i>
<i>Obr. 15: Ukázka použití nástroje pro generování zeměpisných souřadnic .....</i>	<i>42</i>
<i>Obr. 16: Struktura sešitů a souborů sklizní ČSÚ.....</i>	<i>45</i>
<i>Obr. 17: Ukázka načtení tabulek z databáze .....</i>	<i>48</i>
<i>Obr. 18: Ukázka relací pro model v Power BI .....</i>	<i>50</i>
<i>Obr. 19: Konečná podoba modelu v Power BI .....</i>	<i>50</i>
<i>Obr. 20: Příklad skupinového sloupcového grafu v Power BI Desktop.....</i>	<i>52</i>
<i>Obr. 21: Příklad spojnicového grafu v Power BI Desktop.....</i>	<i>52</i>
<i>Obr. 22: Příklad spojnicového a skládaného grafu v Power BI Desktop .....</i>	<i>53</i>
<i>Obr. 23: Příklad KPI v Power BI Desktop .....</i>	<i>53</i>
<i>Obr. 24: Příklad matice v Power BI Desktop .....</i>	<i>54</i>
<i>Obr. 25: Příklad mapy v Power BI Desktop .....</i>	<i>54</i>
<i>Obr. 26: Dashboard pro celkový přehled podniku podle plodin.....</i>	<i>55</i>
<i>Obr. 27: Dashboard pro odrůdy plodin .....</i>	<i>56</i>
<i>Obr. 28: Dashboard pro katastry a hony .....</i>	<i>57</i>



## Seznam bloků kódu

<i>Blok 1: Dotaz pro vytvoření a naplnění tabulky pro pomocné názvy plodin.....</i>	<i>34</i>
<i>Blok 2: Dotaz pro vytvoření tabulky pro data z excelu sklizní.....</i>	<i>34</i>
<i>Blok 3: Část bloku 6 – proměnné .....</i>	<i>35</i>
<i>Blok 4: Část bloku 6 – cykly.....</i>	<i>35</i>
<i>Blok 5: Část bloku 6 – import pomocí OPENROWSET .....</i>	<i>36</i>
<i>Blok 6: Dotaz pro import všech dat z excelových sešitů sklizní .....</i>	<i>37</i>
<i>Blok 7: Dotaz pro odstranění prázdných řádků v tabulce sklizní .....</i>	<i>37</i>
<i>Blok 8: Dotaz pro vytvoření tabulky pro konečná data.....</i>	<i>38</i>
<i>Blok 9: Část bloku 13 – Podmínka pro prázdná datumová pole .....</i>	<i>38</i>
<i>Blok 10: Část bloku 13 – Podmínka pro datumový interval .....</i>	<i>39</i>
<i>Blok 11: Část bloku 13 – Tvorba datumu z levé části intervalu .....</i>	<i>39</i>
<i>Blok 12: Část bloku 13 – Tvorba datumu z pravé části intervalu.....</i>	<i>39</i>
<i>Blok 13: Dotaz pro transformaci dat sklizní do konečné podoby.....</i>	<i>40</i>
<i>Blok 14: Dotaz pro tvorbu tabulky a import dat honů .....</i>	<i>42</i>
<i>Blok 15: Dotaz pro transformaci dat do konečné tabulky honů .....</i>	<i>43</i>
<i>Blok 16: Dotaz pro přidání sloupce do tabulky sklizní pro id_honu .....</i>	<i>43</i>
<i>Blok 17: Dotaz pro tvorbu tabulky a jejím naplněním plodinami .....</i>	<i>44</i>
<i>Blok 18: Dotaz pro import excelových sešitů sklizní z ČSÚ.....</i>	<i>46</i>
<i>Blok 19: Dotaz pro přejmenování názvů plodin z ČSÚ do jednotné konvence .....</i>	<i>47</i>
<i>Blok 20: Dotaz pro transformaci a výběr potřebných dat z ČSÚ.....</i>	<i>47</i>

# 1 Úvod

Každá fyzická nebo právnická osoba má povinnost vést účetní záznamy. Z těchto záznamů lze vyčíst číselné údaje a zhodnotit, která období byla výnosná a která nikoliv. Proč k výdělkům a ztrátám dochází, je tedy nutné vést další druhy záznamů. S příchodem počítačů a jejich dalšími evolucemi a postupně zvyšujícím se výkonem a pamětí bylo vhodné záznamy právnických a fyzických osob elektronizovat. Elektronizace měla veliký potenciál s příchodem databází, kde se informace uspořádaly – vznikla data, a tím vznikl jednotný systém pro správu záznamů.

Rostoucí počet dat, náročnější a složitější operace převýšili možnosti dnešních ORD systémů, proto vznikl nový typ tzv. analytických databází. Tyto databáze jsou přizpůsobeny pro rychlý průzkum velkého množství dat. Pro průzkum a porozumění datům, vznikají nové obory zaměřující se na vizualizační techniky a jejich správné použití. Cílem vizualizace je ulehčení vedení a manažerům porozumění datům a celkovému dění v jejich firmách.

Mít data, zpracovat data, zhotovit výsledky. Bohužel i přes neustálý vývoj je sestavení Business Intelligence stále drahá záležitost. Ať už na nákup serverů, zaplacení IT teamu, samostatné sestavení řešení apod. stojí obrovské množství peněz, a ne každý si tuto investici může dovolit. Z tohoto důvodu se začalo přemýšlet nad řešením, které bude dostupné pro všechny. Ve výsledku vznikly aplikace a řešení podporující tzv. Self-service Business Intelligence. Jedná se o typ Business Intelligence zaměřený na jednoduché a efektivní získávání dat především pro malé a střední firmy. Aplikace typu Self-service by měli uživatelé nabídnout intuitivní funkce a prostředí a uživatel by se měl stát méně závislý na IT oddělení. Cílem bylo zpřístupnit Business Intelligence i mezi menší podniky minimalizováním nákladů na sestavení a provoz řešení. Pro velké podniky přináší Self-service Business Intelligence výhodu především pro manažery, protože snižuje jejich závislost na IT oddělení, a ve výsledku urychluje proces získávání dat.

## 2 Cíl práce a metodika zpracování

Cílem bakalářské práce je definovat termín Self-service Business Intelligence a představit další pojmy a metodiky se Self-service Business Intelligence související. Dále bude demonstrován příklad, ve kterém bude shrnuta celková tvorba procesu Self-service Business Intelligence.

Teoretická část bude vypracována za použití odborných literárních i elektronických publikací a článků. Pro vysvětlení Self-service Business Intelligence je nejdříve nutné charakterizovat Business Intelligenci, ETL proces, Datový sklad, Vizualizaci dat a pojmy k nim vázané. Dále budou představeny nástroje Power Pivot a Power BI a základy jazyku DAX.

V praktické části bude vypracován příklad založený na reálných datech z nejmenovaného zemědělského podniku. V příkladu budou postupně znázorněny všechny kroky, aby byl proces Self-service Business Intelligence představen od začátku (holých dat) až do konce (report). Zemědělský podnik poskytl informace o jejich sklizních za posledních 5 let a další informace potřebné sestavení řešení, jako je třeba seznam polí, plodin atd. Dále budou do řešení přidány průměrné výnosy zemědělských plodin v rámci ČR pro každý rok z webu Českého statistického úřadu. V projektu bude ukázán konkrétní postup importu a zpracování dat v rámci SQL Serveru. Poté budou data importována do Power BI, kde dojde k vytvoření datového modelu, tvorbě vizualizací a dalších potřebných kroků spojených k dokončení procesu Business Intelligence. K vypracování řešení bude využito Excelu, SQL Server Management Studia a Power BI Desktop od společnosti Microsoft.

## 3 Teoretická část

### 3.1 Business Intelligence

Business intelligence (dále jen BI) je termín, který usnadňuje podnikové rozhodování. Vztahuje se ke znalostem, procesům, technologiím, aplikacím a postupům v podniku. BI pracuje s historickými daty podniku a pomáhá přijímat rozhodnutí pro jeho budoucnost. Soustřeďuje se na interní informace, které souvisejí se strategickým plánováním. (Laberge, 2012)

Pro správně implementované BI systémy se vyznačuje, že bývají:

- přesné – datům se dá důvěřovat,
- aktuální – řídí se pravidelným rozvrhem,
- mají vysokou hodnotou – užitečnost pro podnikové pracovníky a možnosti rozhodování. (Laberge, 2012)

#### 3.1.1 Historie Business Intelligence

Pro podporu manažerských a analytických úloh se začala objevovat první řešení na konci 70. let. Podnikový pracovník, který měl dotaz vyžadující přístup k datům, nucen odeslat žádost adresovanou IT oddělení. Poté bylo nutné, aby se pracovník přes telefonáty nebo poštu domluvil s IT pracovníkem, kterému byla žádost svěřena, na konkrétních požadavcích projektu. Po vyjasnění všech požadavků napsalo IT oddělení program, který provedl definovaný výstup. Celý proces mohl trvat několik dnů, dokonce týdnů, podle složitosti požadavku. Často docházelo k potížím při komunikaci, protože jak IT zaměstnanci, tak podnikoví uživatelé nedokázali efektivně formulovat své dotazy a požadavky. (Novotný, 2005)

Koncem 80. let se začaly prosazovat multidimenzionální technologie, například datové sklady. S narůstajícím objemem dat v datových skladech byly v průběhu 90. let zavedeny technologie dolování dat založené na analýzách dat s pomocí matematických postupů. (Novotný, 2005)

Dnešní systémy BI umožňují, aby se podnikový uživatelé na data dotazovali přímo. Touto změnou se podařilo zkrátit rozhodovací procesy i na minuty. Aby se předešlo

problémům s kvalitou dat, IT oddělení vytvořila zásahové jednotky. Zásahové jednotky pohotověji reagují na variabilní potřeby podnikových uživatelů a data jsou tímto strukturovanější, přesnější, kvalitnější. Vše spěje k tomu, aby podnik dokázal rychleji reagovat na změny trhu. (Laberge, 2012)

### 3.1.2 Definice BI

BI patří mezi nové pojmy, a proto pro něj neexistuje jednotná definice. V roce 1989 jej zavedl Howard J. Dresner. Definoval ho jako „*sadu konceptů a metod určených pro zkvalitnění rozhodnutí firmy*“. Soustřeďuje se na podnikové uživatele, kteří provádějí pomocí dat dotazy, analýzy, reporting. (Novotný, 2005)

Další možné definice:

*„Termín Business Intelligence (BI) označuje technologie, aplikace a postupy pro sběr, integraci, analýzu a prezentaci obchodních informací. Účelem Business Intelligence je podporovat lepší rozhodování v oblasti podnikání.“* (OLAP.com)

*„Aplikace a technologie Business Intelligence (BI) jsou v současné době integrální součástí informačních systémů. Jsou orientovány na podporu analytických, rozhodovacích a plánovacích činností prakticky ve všech oblastech podnikového řízení. Předmětem bloku „Business Intelligence“ jsou otázky řízení, projektování a provozování BI aplikací, otázky jejich očekávaných i reálných efektů, otázky organizační a faktory úspěšnosti BI řešení v praxi.“* (Pour, 2005)

*„Business Intelligence umožňuje lidem všech hierarchických celků podniku přistupovat, pracovat, analyzovat data, pro zlepšení výkonu, objevování nových příležitostí a efektivní provoz podniku.“* (Howson, 2014)

## 3.2 Datové sklady

Datové sklady neboli Data Warehouses (dále jen datové sklady) ve většině případech vznikají proto, aby BI mohla poskytovat určitou podnikovou hodnotu. (Laberge, 2012, s. 25)

### 3.2.1 Definice

Bill Inmon (Inmon, 1989) definoval datový sklad jako „*podnikové strukturovaný depozitář subjektivě orientovaných, integrovaných, časově proměnných, historických dat použitých pro získávání informací a podporu rozhodování. V datovém skladu jsou uložena atomická a sumární data.*“ Jedná se asi o nejznámější definici. (Lacko, 2009)

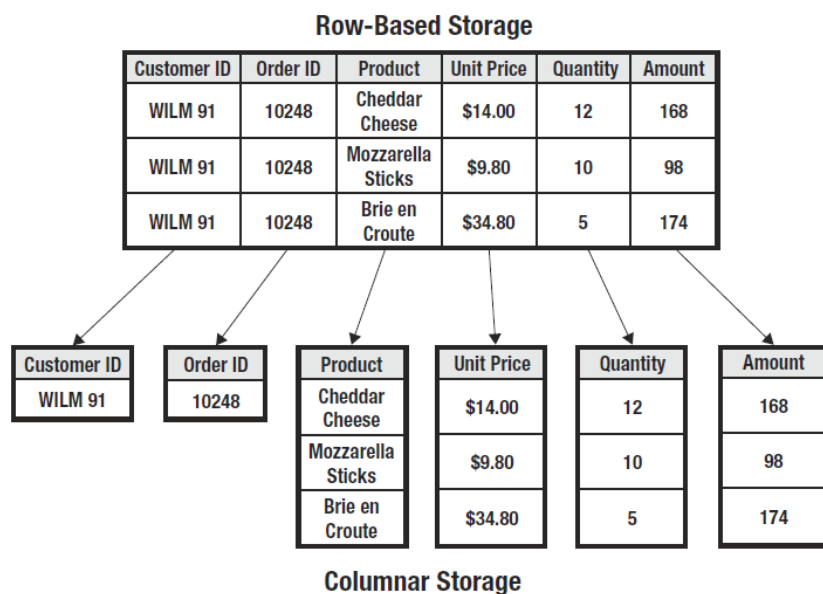
Definice Inmona vychází z databázové teorie. (Lacko, 2009) Datový sklad by se také dal definovat jako „*soubor dat vytvořených pro podporu rozhodování; je také repositářem současných a historických dat potenciálního zájmu pro manažery v celé organizaci. Data jsou obvykle strukturována tak, aby byla k dispozici ve formě připravené pro analytické zpracování.*“ (Turban, 2010)

### 3.2.2 Transakční vs. analytické databáze

Transakční systémy založené především na relačních databázových systémech jsou pro řízení dat výhodné řešení. Data jsou přehledně uspořádána a systémy (v případě, že jsou efektivně navrženy) poskytují odpovídající dobu odezvy na jednotlivé dotazy. Dále například zajišťují integritu a bezpečnost dat. Tento druh systému ale nabízí určitá omezení z hlediska analytických a plánovacích činností. Neumožňuje rychlé a pružné změny kritérií pro analýzy. Při obrovských objemech dat obtížně řeší okamžité přístupy více pracovníků k agregovaným datům. Tyto systémy jsou především určeny pro pořizování a aktualizace dat, přičemž tyto operace zabírají až 100 % systémového výkonu, a analytické úlohy tyto systémy nadměrně zatěžují, a dokonce v některých případech nejsou tyto úlohy proveditelné. (Novotný, 2005)

Obr. 1 demonstruje rozdíl v přístupu k datům v transakčních a analytických databázích. Při požadavku zjistit, co zákazník nakoupil, je u transakční databáze nutné načíst všechny informace, v našem případě 3 řádky, které jsou v obrázku a poté dojde k vyselektování zadaných požadavků. Oproti tomu u analytické databáze při stejném zadání načte rovnou požadované informace, a to Customer ID a Amount. (Howson, 2014)

**Obr. 1: Ukázka ukládání dat v relačních databázích do řádků a v analytických do sloupců**



*Zdroj: HOWSON, Cindi. Successful business intelligence: unlock the value of BI & big data. Second edition. New York: McGraw-Hill Education, [2014]. ISBN 978-0-07-180918-4.*

### 3.2.3 Datová tržiště

*„Datové trhy jsou určité přesně specifikované podmnožiny datového skladu, které jsou určené pro menší organizační složky firmy.“ (Lacko, 2009)*

V datovém skladu se kombinují data z celého podniku. Oproti němu jsou datová tržiště obvykle menší a obsahují data konkrétního oddělení. Datová tržiště jsou tím pádem podmnožiny datových skladů, které se skládají z jedné oblasti zkoumání. Datová tržiště bývají dvojího typu:

- **Dependent (závislá)** – Závislá datová tržiště jsou přímou součástí datového skladu. Hlavní výhoda spočívá v konzistentnosti a aktuálnosti dat pro všechny uživatele datového skladu. Nutnou podmínkou pro závislé datové tržiště je mít vybudovaný datový sklad. Bohužel vybudování datového skladu je finančně náročné a malé podniky si datový sklad nemohou dovolit.

- Independent (nezávislá) – Nezávislá datová tržiště jsou charakterizována jako lower-cost datové sklady. Jedná se o malý datový sklad vytvořený pro určité firemní oddělení, jehož zdrojem není celopodnikový datový sklad. (Turban, 2010)

### 3.3 Datový model

Modely analytických aplikací jsou tvořeny dvěma hlavními entitami. Tabulkami faktů a dimenzí.

- Pro tabulky faktů je specifický obrovský počet řádků. Obsahují například záznamy pro každý prodej.
- Oproti tomu tabulky dimenzí jsou něco jak podnikové číselníky. Příkladem jsou dimenze zboží nebo zákazníků. (Pour, 2018)

Při tvorbě modelu nastává mnoho různých otázek. Například zda je vhodné vytvořit více menších dimenzí, nebo méně větších a komplexnějších. Například dimenze zvlášť pro regiony a pro prodejny nebo společná dimenze prodejen, ve které budou i podrobnější geografické údaje. K jedné faktové tabulce by bylo vhodné mít maximálně 15 připojených. Při větším počtu je vhodné dimenze slučovat. U dimenzí je jedna z hlavních charakteristik dodržování hierarchické struktury dat pro získávání agregovaných ukazatelů. (Pour, 2018)

V datových skladech uchováváme data především z provozu. Z tohoto důvodu data obsahují jakýkoliv časový aspekt, a právě proto se téměř v každém modelu vyskytuje časová dimenze. (Laberge, 2012)

#### 3.3.1 Star schéma

U star schématu jsou v základní dimenzi i všechna další data z nadřazených hierarchických úrovní. Toto řešení způsobuje vysokou redundanci dat, jedná se tedy o denormalizované tabulky. (Pour, 2018)

Star schéma je založeno na centrální tabulce faktů a větších, komplexních dimenzí. Hlavní výhodou je lehká srozumitelnost a použitelnost. Nejoblíbenější varianta star



schématu je rozdělení na statické a dynamické dimenze pro zlepšení výkonu. (Laberge, 2012)

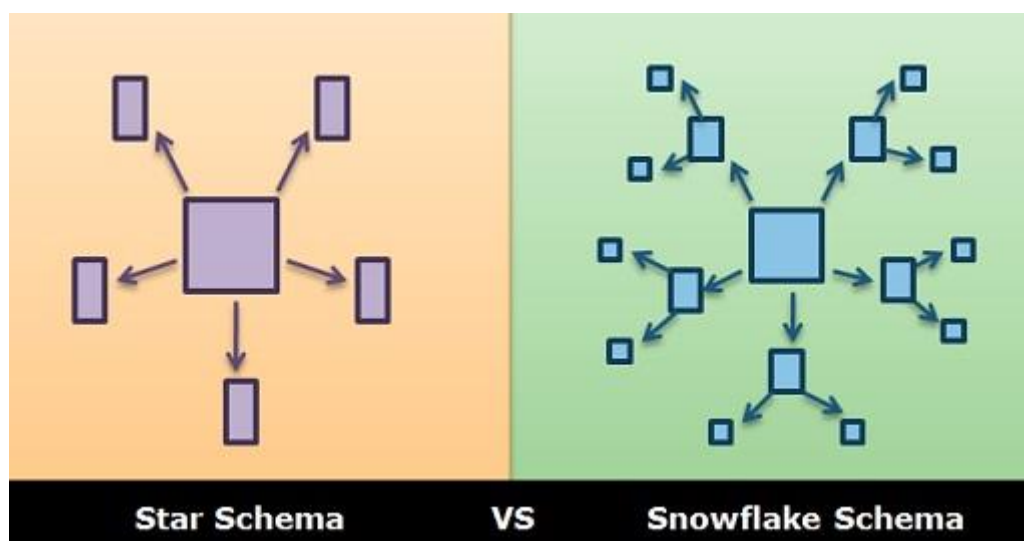
### 3.3.2 Snowflake schéma

U dimenzí ve snowflake schématu je hierarchie tvořena řetězcem provázaných tabulek kardinalitou 1:N. Redundance dat je tímto snížena a můžeme hovořit o normalizované struktuře. (Pour, 2018)

Snowflake schémata popisují stejné typy datových modelů jako star schémata, s tím rozdílem, že jsou normalizované. Normalizace může být částečná nebo úplná podle pravidel třetí normalizační podoby. (Laberge, 2012)

Na obrázku níže je znázorněna struktura obou schémat. Ve středu je tabulka faktů, kolem ní jsou dimenze.

**Obr. 2: Star vs snowflake schema**



Zdroj: TechDifferences [online]. 2017 [cit. 2020-04-22]. Dostupné z: <https://techdifferences.com/difference-between-star-and-snowflake-schema.html>

### 3.4 ETL

ETL představuje v teorii datového skladu vstupní aspekt systému. ETL zajišťuje dva procesy. První se označuje jako pořizování dat (data manipulation) a druhý jako distribuce dat. Prostředí pro pořizování dat vyžaduje transformaci dat z mnoha zdrojových systémů a provádí mapování zdroje na cíl. Distribuční forma zajišťuje

převod dat do formátů datových trhů a jejich následné načtení v datových trzích. (Laberge, 2012)

Etapy ETL jsou:

1. extrakce (extraction) – čtení dat z jednoho nebo více databázových zdrojů,
2. transformace (transformation) – konverze dat z jejich předchozí formy do potřebné formy pro umístění do datového skladu,
3. načtení (load) – zavedení dat do datového skladu. (Turban, 2010)

### **3.4.1 Extrakce**

Data potřebná k převedení do datového skladu mohou být uložena v nepřehledném množství různých operačních prostředcích. Například hardwarové platformy, operační systémy, databázové systémy, podnikové prostředky, dokumenty atd. Úlohou extrakce je získání dat z jakéhokoliv zdroje dat. Extrahovat lze podle různých postupů a technologií. Lze k tomu použít aplikace vytvořené ve vyšších programovacích jazycích nebo v procedurách jazyka SQL, například T-SQL (SQL Server) nebo PL/SQL (Oracle). (Lacko, 2009)

### **3.4.2 Transformace**

Nekvalitní data ve zdrojových systémech vedou k chybným a nepřesným obchodním rozhodnutím. Mezi nejčastějšími problémy při transformaci dat bývají:

- Nejednoznačnost dat – Například údaj o pohlaví zákazníka může mít různé druhy zapsání (F, M, man, woman, female).
- Chybějící hodnoty a duplicitní záznamy – Záznamy mohou chybět v některých relačních databázích, kde se v ve sloupci může objevit hodnota NULL, duplicita je menší problém. Lze ji vždy odstranit.
- Konvence názvů pojmů a objektů – Data slučující se z různých zdrojů popisující stejný jev mohou mít různé názvy, proto je nutné zavést jednotnou konvenci pro sloučení názvů.

- Formáty čísel a textových řetězců – Numerickou hodnotu lze zapsat v číselných hodnotách i v textových řetězcích. Proto je nutné odlišit do jakých datových typů čísla uložit.
- Chybějící datum – Časový údaj je pro datový sklad velice důležitý, protože má téměř každá analytická databáze časovou dimenzi. (Lacko, 2009)

Všechny tyto problémy je nutné vyřešit, protože „*když jsou data, která slouží jako podklad pro získávání informací pro podporu rozhodování, nebo data v datovém skladu nekvalitní, snižuje to důvěru v takové řešení a datový sklad se oprávněně nevyužívá.*“ (Lacko, 2009)

### 3.4.3 Načtení

Finální etapou ETL je fyzický přenos dat ze zdroje, či přechodné oblasti do cílové databáze/datového skladu, konkrétně se jedná o uložení do databázových tabulek. Celý proces načtení by měl být plánovaný, automatizovaný a optimalizovaný. (Lacko, 2009)

## 3.5 OLAP

„*OLAP (Online Analytical processing) je technika, která se soustřeďuje na analýzu a prohledávání dat, dotazy a reportovací nástroje umožňují přístup k datům pro monitorovací účely.*“ (Howson, 2014)

Jedná se pravděpodobně o nejpoužívanější techniku v datových skladech umožňující analýzu dat. Popularita rostla kvůli rychlým odezvám, i při navyšování objemů datových skladů. (Turban, 2010)

OLAP využívá koncept nazvaný krychle či kostka. Za kostku je považována vícerozměrná datová struktura, která má schopnost efektivně manipulovat s daty z různých perspektiv. (Turban a kol, 2010, s. 57) Sestavy lze vytvářet dynamicky díky úrovním podobností neboli granularitou. Granularita může nabývat různých hodnot, jako například pro:

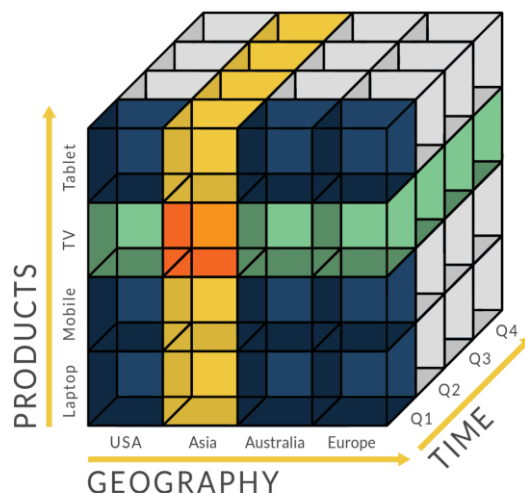
- Hierarchii dat – časové období, datum (v rocích, měsících, čtvrtletích), ...,
- Geografické hierarchii – země, kraj, větší město, ...

- Produkty – mohou mít vlastní několika úrovněnou hierarchii.

(Laberge, 2012)

Na obr. 2 jsou vidět příklady konkrétních hodnot granularity v OLAP kostce.

**Obr. 3: Příklad návrhu OLAP kostky**



Zdroj: WHAT IS THE DEFINITION OF OLAP? OLAP.com [online]. PARIS Technologies [cit. 2020-03-27].  
Dostupné z: <https://olap.com/olap-definition/>

### 3.6 Dolování dat

Dolování dat neboli „*data mining je proces analýzy dat z různých perspektiv a jejich přeměna na užitečné informace. Z matematického a statistického hlediska jde o hledání korelací, tedy vzájemných vztahů nebo vzorů v datech. Data mining je proces, jehož cílem je těžba informací v databázích. Využívá statistické metody a další metody hraničící s oblastí umělé inteligence.*“ (Lacko, 2009)

Pro využití dolování dat je nutné přijmout určité hypotézy a poté pomocí dotazů hledat v datech smysl a komplexnější souvislosti. Dotazy při dolování dat by neměly být zaměřeny na standardní pátrání v datech, jako je například dotaz pro hledání počtu produktů, které se prodaly v nějaký čas na nějakém místě. Při dolování dat by se například mělo dotazovat, jaké produkty se prodávají společně, nebo jaké vlastnosti mají zákazníci, co dané produkty nakupují. (Laberge, 2012)

Dolování dat se v podnikové informatice nejčastěji využívá v oblasti marketingu, například pro vytipování vhodných zákazníků do marketingové kampaně. Z chování zákazníků, uloženého v databázi ve formě záznamů, je za pomoci dolovacích modelů možné předpovědět chování klienta při nastání určité situace. (Lacko, 2009)

### 3.6.1 Průběh procesu dolování dat

Základní kroky procesu bývají tyto:

- Definice problému – Před zahájení dolování by měl být dobře definovaný projekt daného problému či příležitosti, na který se dolování zaměří.
- Výběr dat – Po definování problému je nutné určit zdroje dat. Ve většině případů jsou data ze zdrojů extrahována na předpřipravený server, kde je dolování realizováno.
- Příprava dat – Dolování vyžaduje, aby analyzovaná data byla připravena do jedné tabulky. Často bývá vytvořeno i tisíc proměnných, které vstupují do procesu modelování, a proto je tato část procesu dolování časově nejnáročnější.
- Data Mining – V této části procesu dolování dochází k použití statistických a nestatistických prostředků pro tvorbu matematických modelů. Data Mining se provádí na samostatném serveru odděleného od datového skladu.
- Zprovoznění modelu – Jedná se o proces, kdy dochází k implementaci vytvořených modelů z procesu Data Miningu do operačního systému.
- Obchodní akce – Jedná se o využití implementovaných modelů pro získání řešení pro zadaný problém či příležitost. (Novotný, 2005)

### 3.7 Vizualizace dat

Aby byla data z pohledu uživatele uspokojivá, je jim potřeba porozumět. Data je potřeba správně interpretovat a prezentovat, jinak se mohou vytrácet jejich nejdůležitější aspekty. Z tohoto důvodu je důležitá správná volba vizualizace. (Pour, 2018)

Složitost vizualizačního problému, vyplývá z mnoha aspektů, které mají vliv na vizuální reprezentaci naznačují, že musí existovat řada technik – skutečně jich

existuje mnoho. Existují techniky, které jsou velmi specifické pro konkrétní aplikační doménu, jiné obecnější s možnou použitelností v jiných dalších oblastech. Důvodem existence velkého množství technik je, že vizualizace časově orientovaných dat je horkou oblastí výzkumu, která neustále přináší nové techniky. Vizualizační řešení mohou být vysoce závislá na specifické případy, a proto je prakticky nemožné dohledat každou drobnou variantu existujících vizualizačních přístupů. (Aigner, 2011)

Techniky vizualizace se dělí podle třech klíčových kritérií: data, čas a typem vizualizace. Pro každé klíčové kritérium zavádíme další dílčí kritéria a odpovídající charakteristiky:

- *„Data*
  - *Referenční rámec – abstraktní vs. prostorový*
  - *Proměnné – jedna vs. více*
- *Čas*
  - *Uspořádání – lineární vs. cyklické*
  - *Časová primitiva – instantní vs. intervalové*
- *Vizualizace*
  - *Mapování – statický vs. dynamický*
  - *Dimenze – 2D vs. 3D“*

(Aigner, 2011)

Data se v reportech velmi často prezentují ve formách tabulek. V rozsáhlých tabulkách bývá velmi často velice špatná orientace, a uživatelé mohou přehlížet velmi důležité odchylky nebo limitní hodnoty. Grafické znázornění může tyto nedostatky eliminovat a uživatel je schopen co nejrychleji získat informace, které potřebuje. (Pour, 2018)

Nepodstatné prvky při prezentaci dat mohou odvádět pozornost od základních informací, proto je důležité se jim vyhnout. (Pour, 2018)

Prezentované hodnoty můžeme dělit na:

- *„Kvantitativní (číselné): hodnoty, které můžeme měřit a kumulovat, například u zaměstnance je to počet odpracovaných dnů, počet dnů nemocenské, výše mzdových příplatků atd.*
- *Kategorické (nominální): kvalitativní vlastnosti, které představují abstraktní vlastnosti. Nemůžeme je počítat, ale lze je využít jako kategorie pro kumulované počty podle hodnot ze seznamu. Například u zaměstnance jsou kategorickými hodnotami jeho jméno, příjmení, pohlaví, profese, typ řidičského průkazu apod.*
- *Ordinální: stejně jako kategorické hodnoty je nemůžeme měřit, ale lze u nich určit pořadí. Jedná se o kvalitativní hodnoty (převážně číselné), u nichž lze určit pořadí podle rozsahu hodnot v rozmezí od-do, například stupeň vzdělání, mzdová tarifní třída, datum narození atd.“*

(Pour, 2018)

Pro prezentování v analytických úlohách se používají reporty obsahující dashboardy a scorecardy. V nich bývají hlavní vizualizační komponenty grafy. (Pour, 2018)

### **3.7.1 Tabulky**

Tabulky dat se využívají v případech, kdy jsou vyžadovány detailní informace o potřebných veličinách. „Tabulka představuje jednoduchou strukturu uspořádání dat ve sloupcích a řádcích, tedy ve dvojrozměrné vazbě.“ Díky jednoduché struktuře se dá v tabulce velice jednoduše orientovat, bohužel, pokud se zobrazuje více dat, je potřeba zvážit, zda nejsou v tabulce až moc podrobná a data, které uživatel nevyžaduje. (Pour, 2018)

Vizualizace v tabulkách je podmíněna formátováním buněk. Trendy a odchylky bývají vyznačeny barvami písma, barvami buňky nebo různými ikonami. (Pour, 2018)

### **3.7.2 Grafy**

*„Grafy jsou velmi častou vizualizační komponentou pro zobrazení ekonomických (statistických) dat.“* Bohužel se stává, že bývá pro prezentaci daných dat vybrán

nevhodný výběr grafu, proto je důležité si uvědomit, jaký je hlavní cíl neboli myšlenka prezentace dat. Data lze totiž prezentovat různými způsoby a tím zdůrazňovat odlišné aspekty výstupu. Důležitý je pro nás uživatel a jeho rychlá orientace v informacích, které potřebuje získat. (Pour, 2018)

Vzhledem k převaze kvantitativních údajů spadá většina zobrazovacích médií v dashboardech do kategorie grafů. (Few, 2006)

### **3.7.2.1 Spojnicový graf (Line Chart)**

Jedná se o nejběžnější formu reprezentace časových řad. Rozšiřují bodové grafy propojením datových bodů s linkami, které zdůrazňují jejich časový vztah. Spojnicové grafy se proto zaměřují na celkový tvar dat v průběhu času. (Aigner, 2011)

*„Spojnicový graf slouží pro zobrazení vztahu kvantitativních a kvalitativních hodnot jedné i více proměnných, kdy se velmi často jako kvalitativní ordinální hodnota využívá čas.“* Z tohoto důvodu je graf vhodný například pro zobrazování časových řad a jejich korelací nebo například vývoje odchylek v určitém časovém období. (Pour, 2018)

Spojnicové grafy odhalují tvar pohybu datových dat nahoru a dolů z jedné hodnoty na další, zejména když se mění v průběhu času. Kdykoli chcete zdůraznit trendy, fluktuace, cykly, míry změn a jak se dvě sady dat liší ve vztahu k sobě navzájem, nejlepší variantou jsou čárové grafy. V kontextu dashboardů jsou čárové grafy často nejlepším prostředkem k rychlému přehledu časové řady. (Few, 2006)

### **3.7.2.2 Sloupcový graf**

Sloupcový graf bývá často využíván a *„představuje nástroj pro porovnávání pro porovnání kvantitativních a kvalitativních hodnot, kdy kvalitativními hodnotami jsou často jednotky času. Slouží pro nominální porovnání kvantitativních hodnot pomocí sloupců a využívá výšky sloupce pro znázornění hodnot.“* Díky různým výškám sloupců získává uživatel rychlý přehled při porovnávání hodnot. Data bývají znázorněna i ve více datových řadách. (Pour, 2018)

Sloupcové grafy jsou nejlepším prostředkem pro zobrazení případů rozdělených podél jmenovité nebo pořadové stupnice. Vizuální váha sloupců klade důraz na



jednotlivé hodnoty v grafu a usnadňuje srovnání hodnot jednoduchým porovnáním výšky sloupců. (Few, 2006)

*„Sloupcové grafy jsou dobře známým a široce používaným typem zobrazení, kde se sloupce používají k zobrazení datových hodnot.“* Lze reprezentovat proměnné pouze s poměrovou stupnicí (mající přirozenou nulu). V důsledku toho musí stupnice hodnot také začínat nulou, aby se umožnilo spravedlivé vizuální zobrazení srovnání. (Aigner, 2011)

Hlavní prioritou sloupcového grafu je vysoká přehlednost a srozumitelnost, nicméně jej lze použít pouze pro omezené množství položek na ose x. Orientace se mezi mnoha sloupci na ose snižuje, a uživatel stěží vyvozuje rozdíly a sloupce do sebe splývají. (Pour, 2018)

Při grafické úpravě je třeba dbát na vhodnou šířku sloupců a mezer mezi nimi. To lze docílit jednoduchou metodou: šířka sloupce by měla být dvakrát větší než mezera mezi sloupci. (Pour, 2018)

#### **3.7.2.2.1 Pruhový graf**

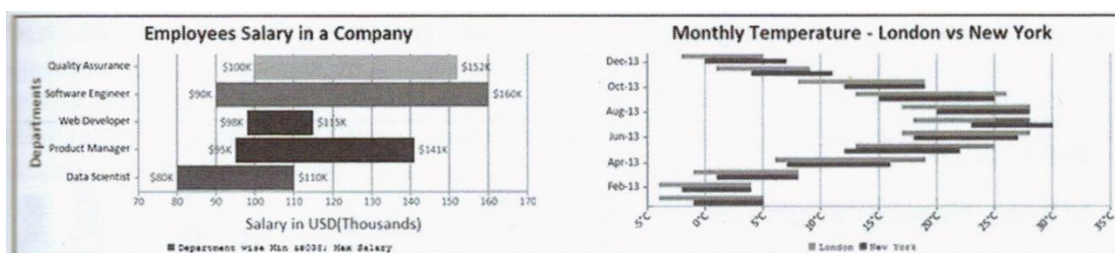
Graf pruhový je obdoba sloupcového grafu, od kterého se liší svým horizontálním provedením. *„Pruhový graf je vhodný, pokud zobrazujeme více položek, které by při omezené šířce na ose x byly nesnadno zobrazitelné.“* (Pour, 2018)

Například lze přehledně porovnat pláty, které mají různá rozpětí. V levé části obrázku vidíme porovnávání mezd pro vybrané profese. V pravé části je ukázka porovnání rozpětí teplotních hodnot pro různá města. (Pour, 2018)

### 3.7.2.2 Dvojitý sloupcový graf

Dvojitý sloupcový graf je varianta sloupcového grafu, která se často využívá pro porovnání hodnot dvou proměnných. Většinou se porovnávají hodnoty skutečné a plánované (například tržby, náklady, zisk). (Hassan, 2014)

Obr. 4: Ukázka pruhového a dvojitého pruhového grafu



Zdroj: POUR, Jan, Miloš MARYŠKA, Iva STANOVSKÁ a Zuzana ŠEDIVÁ. *Self service business intelligence: jak si vytvořit vlastní analytické, plánovací a reportingové aplikace*. Praha: Grada Publishing, 2018. *Management v informační společnosti*. ISBN 978-80-271-0616-5.

### 3.7.2.3 Výsečový graf (Pie Chart)

„Výsečový graf slouží pro zobrazení vztahu částí k celku pro jednu kvantitativní proměnnou.“ Graf má tvar 100 % kruhu, do kterého se hodnoty vynášejí jako výseče. Součástí grafu nejsou osy. „Úhel výseče odpovídá podílu části celku.“ (Pour, 2018)

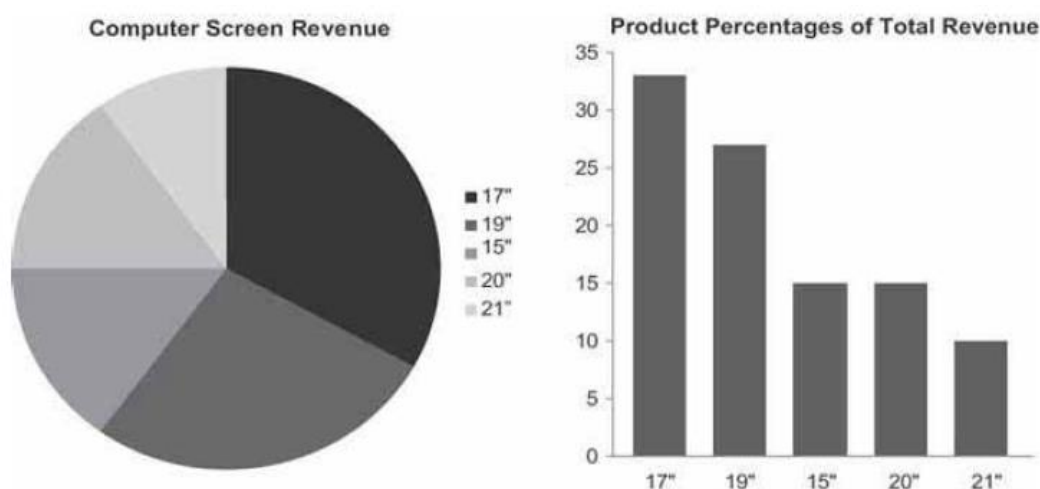
Výsečový graf je v praxi vysoce využívaným grafem, bohužel při jeho použití mohou vznikat chybné interpretace hodnot. Kruh jako tvar je mezi uživateli velice vítán, ale z pohledu odborníků v oblasti vizualizace je velice nepřesný z hlediska lidského vnímání velikostí výsečí. Úhly 90°, 180°, 270°, tedy čtvrtiny kruhu jsou pro lidské oko dobře rozpoznatelné. Právě proto dochází často ke špatnému pochopení hodnot u ostrých nebo tupých úhlů, protože lidské oko nedokáže přesně určit reálnou hodnotu takového výseče. (Pour, 2018)

V grafu je pro lepší přehlednost doporučeno řadit kategorie podle hodnot sestupně, od největších podílů po nejmenší. U příkladu s větším počtem kategorií dochází, že jsou drobné výseče umístěny velice blízko u sebe, a proto nejsou procentuální informace o hodnotách dobře čitelné. Tento problém lze vyřešit použitím dalšího

grafu, který bude tvořit jednu z výsečí z hlavního grafu. Např. na obrázku 2.18 jsou všechny podíly do hodnoty 5% z hlavního grafu spojeny do jedné výseče a pro vizuální reprezentaci této výseče je využit graf sloupcový. Použitím této výseče je zdůrazněn kontrast k ostatním výsečím. (Pour, 2018)

I když se jedná o hodnoty, které jsou částí celku, měl by být použit sloupcový graf, nikoli populární koláčový graf. Ve sloupcovém grafu budou data prezentována mnohem jasněji. Například někde v textu (například v nadpisu grafu) uvedete, že sloupce představují části celku. (Few, 2006)

**Obr. 5: Srovnání vzhledu koláčového a sloupcového grafu pro stejná data**



Zdroj: FEW, Stephen. *Information dashboard design: the effective visual communication of data*. Cambridge [MA]: O'Reilly, c2006. ISBN 0-596-10016-7.

### 3.7.2.3.1 Prstencový graf

Prstencový graf je hlavní varianta grafu výsečového. Stejně jako výsečového, i tento graf slouží k prezentování podílů částí z celku. Jednotlivé umístění oblouků je spojeno se stejnými pravidly jako u výsečového grafu. Hlavní rozdíl mezi výsečovým a prstencovým grafem je chybějící středová část grafu, proto se velikost výsečí vyjadřuje délkou výseče, nikoliv úhlem ze středu grafu. Vlastnost prázdného středu se dá využít například u vynášení více datových řad u oblouku. (Pour, 2018)

Prstencový graf vypadá jako blízký příbuzný tradičního výsečového grafu, nicméně s jedním velkým rozdílem. Prostředek koláče je pryč. Prostředek koláče slouží ke stanovení úhlu, což je to, na co by se lidé měli soustředit, aby se mohli jednoduše orientovat v datech a pochopili celý význam. Když odstraníme střed výseče, nejsme přesní při interpretaci úhlů, pochopení grafu se zhoršuje. (Evergreen, 2017)

*„Podobně jako u výsečového grafu je třeba dobře zvážit vhodnost použití tohoto typu, neboť při velmi podobném rozsahu hodnot jsou jednotlivé délky oblouků mezi sebou těžko rozlišitelné a grafické znázornění neplní dobře svou funkci.“* (Pour, 2018)

### **3.7.2.4 Bodový graf (Scatter plot)**

Bodový graf lze vytvořit použitím kartézského souřadného systému s časem na horizontální ose a odpovídající hodnotou na vertikální ose. Bod je vynesena do grafu pro každý měřený pár: např. čas-hodnota. (Aigner, 2011)

Bodový graf zobrazuje, zda, v jakém směru a do jaké míry jsou korelovány dvě propojené množiny kvantitativních hodnot. Například, pokud chcete ukázat, že existuje vztah mezi počtem vysílaných reklam a tržbami z prodeje, graf by sloužil velmi dobře. (Few, 2006)

Bodový graf se dá také popsat jako dvourozměrné zobrazení, kde jsou kvantitativní údaje vizualizovány podle vzdálenosti od hlavní osy. (Harris, 1999)

*„Bodový graf je velmi vhodný pro grafické znázornění velké množiny kvantitativních hodnot ve formě bodů. Jejich rozptyl (scatter) tak velmi dobře představuje vztahy mezi hodnotami.“* Barva se zde využívá jako třetí dimenze. Díky barvě lze rozlišit kategorie pro porovnání hodnot. V grafech lze místo bodů používat různé tvary jako např. čtverce, trojúhelníky apod. Nicméně časté používání těchto značek může snížit přehlednost a snížit orientaci v grafu. (Pour, 2018)

Někdy jsou bodové grafy vykreslovány trojrozměrně, aby se zobrazila korelace tří kvantitativních proměnných, nikoli pouze dvou. Někdy se také používají i jiné metody ke zvýšení počtu korelovaných proměnných, ale tyto metody se nedoporučují používat, protože i když jsou grafy navrženy dobře, vyžadují příliš

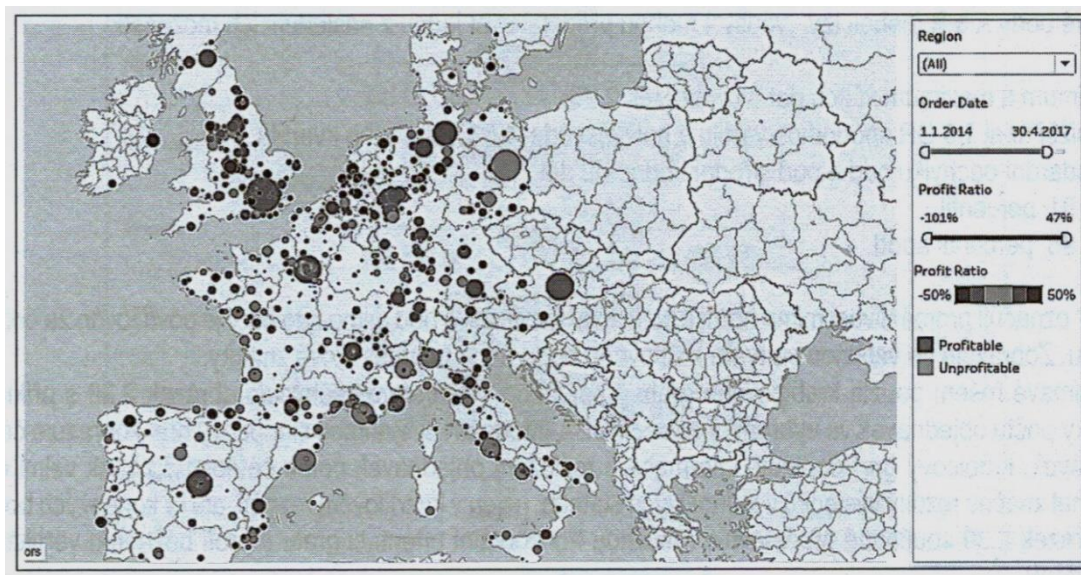
mnoho času na pochopení, který diváci při pohledu na dashboard nemají. (Few, 2006)

### 3.7.2.5 Bublinový graf

„Bublinový graf je zvláštní variantou bodového grafu. Umožňuje zobrazit velké množství dat, kvantitativních hodnot a vztahů mezi nimi.“ Bublinové grafy jsou vhodné pro opravdu velká množství dat. Jako u bodového grafu se umísťuje bod do prostoru souřadnic x a y, bodu lze přiřadit i barvu. Navíc je možné přidat další dimenzi bodu – velikost (bubliny). Možnost využití bublinových grafů je spojení s geografickými celky, tak že se data promítají na mapě. (Pour, 2018)

Na rozdíl od bodových grafů, kde je čas mapován na horizontální nebo vertikální ose, se k reprezentaci času lze využít animace. Proto jsou dvě datové proměnné mapovány na osy kartézského souřadnicového systému a animace se používá pro krok v čase. Velikost tečky představuje třetí proměnnou a barva se používá pro rozlišení skupin. Animaci lze ovládat pomocí posuvníku času, tlačítka přehrávání/pauzy a posuvníku pro nastavení rychlosti animace. (Few, 2006)

**Obr. 6: Bublinový graf aplikovaný na mapě**



Zdroj: POUR, Jan, Miloš MARYŠKA, Iva STANOVSKÁ a Zuzana ŠEDIVÁ. *Selfservice business intelligence: jak si vytvořit vlastní analytické, plánovací a reportingové aplikace*. Praha: Grada Publishing, 2018. *Management v informační společnosti*. ISBN 978-80-271-0616-5.

### 3.7.2.6 Krabicový graf (Box Plot)

Krabicový graf je poměrně novým přírůstkem do lexikonu grafů. Byl vynalezen v 70. letech matematikem John Wilder Tukey. Tento konkrétní typ grafu zobrazuje rozložení hodnot v celém rozsahu, od nejmenšího po největší s mnoha užitečnými aspekty. (Few, 2006)

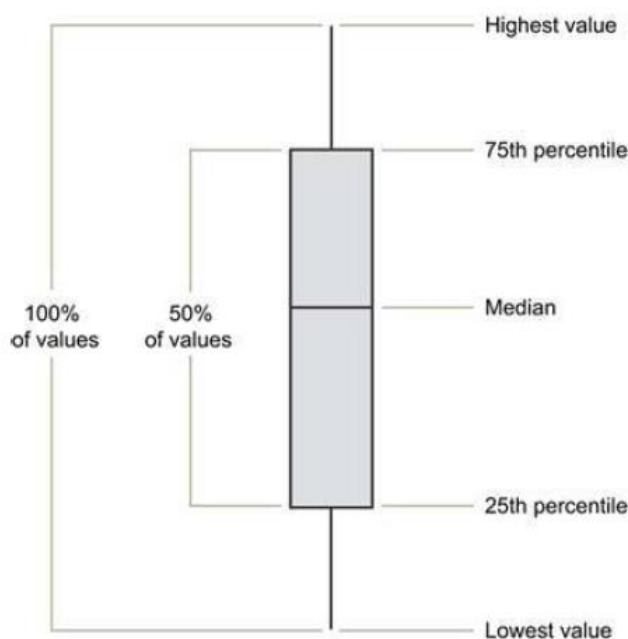
Často je neadekvátní popisovat soubor hodnot jako jediné souhrnné měřítko, jako je součet nebo průměr. Někdy je důležité popsat, jak jsou tyto hodnoty distribuovány v celém rozsahu. Například, abyste plně porozuměli povaze odměňování zaměstnanců ve vaší společnosti pro každou platovou třídu, byste měli vidět více než součet těchto platů. Hodnota, jako např. průměr nebo medián, by vám toho dostatečně neřekla. (Few, 2006)

Hlavní princip krabicového grafu je v rozdělení promítaných hodnot na stejně početně velké části – kvartily. „Kvartily dělí soubor na čtyři stejné části, z nichž každá představuje 25 % podíl.“ (Pour, 2018)

Hlavní část grafu má tvar boxu. Spodní část představuje dolní kvartil, což je 25 % nejnižších hodnot a horní část představuje horní kvartil, což je 75 % nejvyšších hodnot. Medián se značí čarou mezi horním a dolním kvartilem a představuje hodnotu uprostřed, tedy 50 %. Mezi další údaje, které jsou potřeba k sestavení grafu je kvartilový rozptyl a koncové limitní hodnoty. Kvartilový rozptyl se určuje rozdílem horního a dolního rozptylu. Za koncovými limitními hodnotami se mohou nacházet body tzv. odlehlé hodnoty, které se svou hodnotou výrazně liší od ostatních hodnot celku. (Pour, 2018)

Krabicový graf se všemi údaji, které lze vyznačit je možná až příliš složitý pro většinu dashboardů a většinu lidí, kteří je používají. Pro použití v dashboardu může být výhodnější jednodušší verze vykreslování krabicového grafu, jako je ta na obrázku 6–39 na následující stránce. (Few, 2006)

**Obr. 7: Zjednodušená verze krabicového grafu vhodná pro dashboard**



*Zdroj: FEW, Stephen. Information dashboard design: the effective visual communication of data. Cambridge [MA]: O'Reilly, c2006. ISBN 0-596-10016-7.*

### 3.7.3 Scorecard

*„Scorecard je specializované aplikace, která slouží pro porovnávání vývoje hodnot vybraných klíčových ukazatelů výkonnosti ve vztahu k jejich předem definovaných cílovým hodnotám.“ (Pour, 2018)*

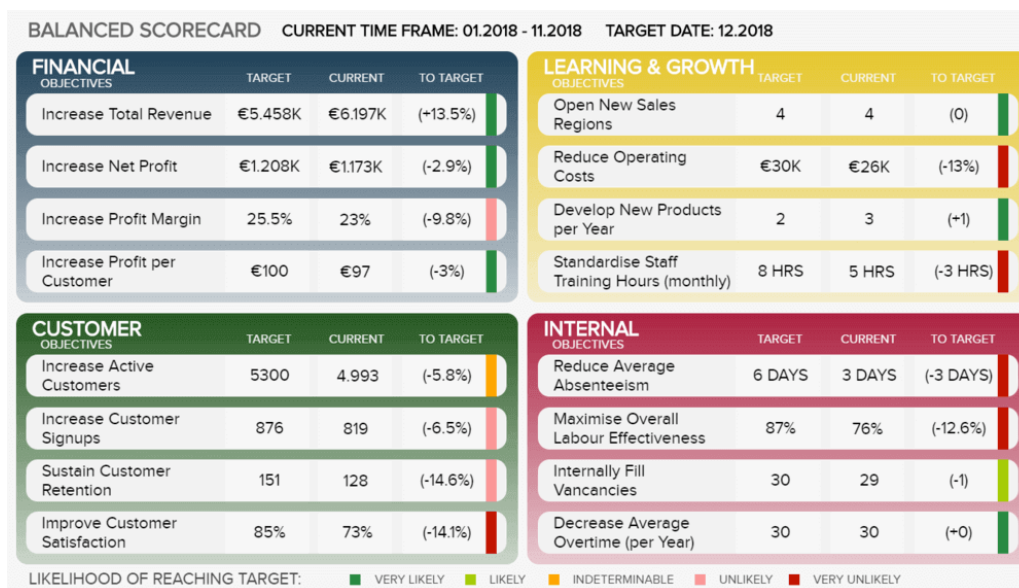
*„Scorecard je strategický manažerský nástroj, jenž napomáhá měřit, monitorovat a komunikovat strategický plán a cíle organizace způsobem, který je všem srozumitelný.“ Scorecard, stejně jako dashboard, zobrazuje data v grafech, ale obsahuje i textové komentáře vysvětlující výsledky a předvídá budoucí vývoj. (Few, 2006)*

*„Scorecardy se nejčastěji používají ke sledování KPI, protože se zaměřují jak na aktuální stav sledované metriky, tak na cílovou hodnotu.“ Data nejsou aktualizována v reálném čase, a proto se používají ke sledování strategických cílů a k přijímání podnikových rozhodnutí ve větším měřítku. (Liberty, 2018)*

Ve scorecardu jsou typy grafů, které prezentují scóre určitých vztahů z různých reportů. Ikony i jiná doprovodná infografika pomáhá dosažení rychlého přehledu o vztazích mezi hodnotami nebo vývojovým trendem. Díky scorecardu získává manažer rychlý přehled o kladných či záporných trendech vůči plánovaným hodnotám. (Pour, 2018)

Sofistikovanější scorecardy a zobrazovací media jsou užitečná pro analytiku, kteří musí zkoumat složitá data a vztahy a jsou ochotni investovat čas potřebný k tomu, aby se naučili, jak fungují. Scorecardy pro analytiku by měly podporovat interakce s údaji, jako je zkoumání podkladových detailů a aby umožnily možnost k pochopení tomu, aby se dalo zjistit, co se děje, ale aby se prozkoumaly příčiny. (Few, 2006)

**Obr. 8: Ukázka scorecardu**



Zdroj: DURCEVIC, Sandra. Datapine [online]. 2019 [cit. 2020-04-22]. Dostupné z: <https://www.datapine.com/blog/kpi-scorecard-examples-templates-to-track-performance/>

### 3.7.4 Dashboard

„Dashboard je aplikace, která umožňuje uspořádat prezentaci vybraných, předem definovaných klíčových indikátorů výkonnosti v přehledné a intuitivní grafické podobě.“ (Pour, 2018)

Dashboard se dá popsat jako „vizuální zobrazení nejdůležitějších informací potřebných k dosažení jednoho či více cílů. Informace jsou konsolidované a uspořádané na jedné obrazovce tak, aby byly přehledné na první pohled.“ (Few, 2006)



Dashboards jsou centrální rozhraní, které poskytují monitoring různých datových sad. Jedná se o nástroj, který monitoruje informace v reálném čase. Data jsou stále aktualizována, a proto je možné sledovat provozní výkon v reálném čase. Není běžné, aby dashboards porovnávaly sledované metriky s cílovými hodnotami. Pro tuto práci jsou určeny scoreboardy. (Liberty, 2018)

Rozdíl mezi dashboardem a scorecardem je závislý na úhel pohledu na monitoring dat o podniku. Dashboard monitoruje základní klíčové informace podniku, zatímco scorecard napomáhá „kontrolovat data o průběhu a řešení určité strategie.“ (Pour, 2018)

**Obr. 9: Ukázka dashboardu**



Zdroj: MICROSOFT. Úvod do řídicích panelů pro návrháře Power BI [online]. 19. září 2019 [cit. 2020-01-20]. Dostupné z: <https://docs.microsoft.com/cs-cz/power-bi/service-dashboards>

### 3.8 Self-service Business Intelligence

Self-service Business Intelligence neboli SSBI nebo samoobslužné BI (dále jen SSBI) je vývojový směr „v rámci prostředí BI, který umožňuje BI uživatelům stát se více samostatnými a méně závislými na IT organizaci. Tento směr se zaměřuje na čtyři hlavní cíle:

- *snadnější přístup ke zdrojovým datům pro reporting a analýzu,*
- *snadnější a vylepšenou podporu funkci pro analýzu dat,*
- *rychlejší možnosti nasazení, jako jsou například zařízení a cloud computing,*
- *jednodušší, přizpůsobitelné a intuitivní rozhraní pro koncového uživatele.“*

(Imhoff, 2011)

Smyslem SSBI je poskytnout uživatelům prostředí pro realizaci analytických úloh bez využití složitých a komplexních BI systémů. (Pour, 2014)

### **3.8.1 Výhody a nevýhody SSBI**

Na trhu se vyskytuje řada produktů, řešící SSBI. I přes odlišnosti každého produktu jsou výhody a nevýhody společné. (Pour, 2014)

Mezi výhody například patří:

- *„podstatně se zkracuje doba potřebná na implementaci potřebných analytických aplikací,*
- *SSBI aplikace nabízejí obdobnou flexibilitu a výkonnost, jako je tomu většinou u aplikací založených na OLAP databázích,*
- *základní příprava aplikací, včetně transformací zdrojových dat je výrazně zjednodušená, takže je dostupná i ne IT pracovníkům, samozřejmě po nezbytném zaškolení,*
- *finanční, zdrojová i provozní náročnost SSBI je oproti standardním projektům a provozovaným aplikacím výrazně nižší,*
- *self-service poskytuje i velmi dobrou podporu pro tvorbu analytické i vizuálně náročnějších aplikací, jako např. dashboardů, naklikávacích map apod.“*

Mezi nevýhody například patří:

- *„SSBI řešení se mohou realizovat pouze pro úlohy určité kategorie, tedy relativně jednodušší a izolovanější úlohy, obvykle nikoli úlohy celopodnikového charakteru,*

- *SSBI má omezené možnosti čištění a konsolidace dat, které u standardních BI aplikací představují jednu z pracovně nejnáročnějších, ale i finálně nejefektivnějších jejich částí,*
- *problémem je dosažení integrace dat v rámci podniku, které nabízejí celopodnikové datové sklady,*
- *s předchozí poznámkou souvisejí i podstatně omezené možnosti celopodnikového reportingu, např. centrálního reportingu v rámci nadnárodních společností apod.,*
- *i když je příprava analytických aplikací na bázi SSBI uživatelsky výrazně dostupnější, přesto je nutné, aby pro kvalitnější aplikace uživatelé ovládali alespoň základní principy analytických metod, tj. především dimenzionálního a datového modelování.“*

(Pour, 2014)

### **3.8.2 Power Pivot**

Jedná se o nástroj od společnosti Microsoft, který je integrován do aplikace Excel z řady Microsoft Office a aplikace Power BI Desktop. Primárně slouží k tvorbě analytických zpráv a dashboardů. (Clark, 2017)

#### **3.8.2.1 Power Pivot pro Excel**

Před vydáním Power Pivotu používali analytici pro analýzu dat samostatně v aplikaci Excel. V ní vytvořili kontingenční tabulky a prováděli analýzu dat nasbíraných z různých zdrojů. Toto řešení mělo nedostatky u aktuálnosti verzí jednotlivých sešitů Excelu, v kvalitě a bezpečnosti dat. (Clark, 2017)

Kvůli těmto nedostatkům přišel Microsoft s řešením. „*Pomocí nástroje Power Pivot můžete vytáhnout data z datového skladu, rozšířit je o další zdroje dat, například textové soubory nebo webové datové kanály, vytvořit vlastní opatření a analyzovat data pomocí kontingenčních tabulek a pivotních grafů. Můžete vytvořit rychlé důkazy o konceptech, které lze snadno propagovat, aby se staly součástí celopodnikového řešení.“* (Clark, 2017)

Mezi výhody Power Pivotu patří:

- jedná se o bezplatný doplněk k aplikaci Excel,
- známé prostředí Excelu,
- možnost integrace a uložení velkého množství dat,
- obsahuje výkonný jazyk DAX. (Clark, 2017)

Microsoft dále nabízí rozšíření funkcionality Excelu s Power Pivotem v nástrojích jako je SharePoint nebo OneDrive což přináší další možné výhody:

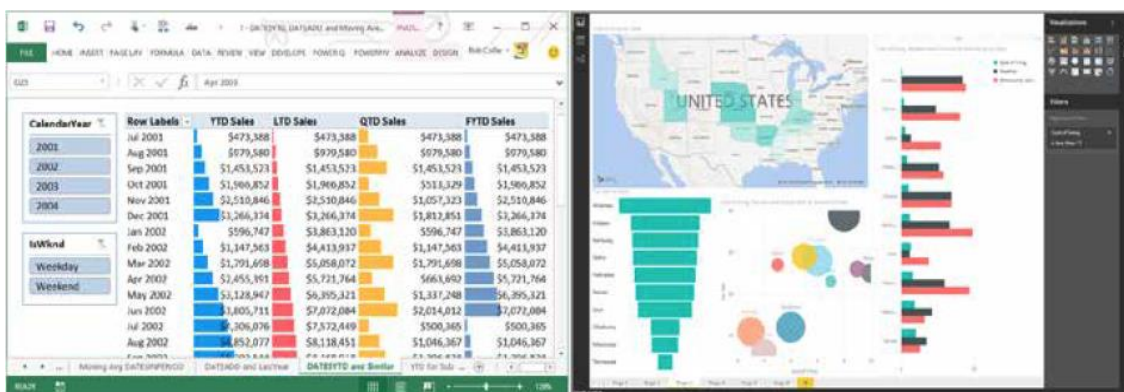
- možnost sdílení,
- možnost plánování automatizované obnovy dat,
- provedení auditu,
- umožnění uživatelům přístupu jen pro čtení či aktualizaci dat. (Clark, 2017)

### **3.8.2.2 Power BI**

Možnosti Excelu jsou primárně určeny pro individuální analýzu a sdílení výsledku mezi malou skupinku spotřebitelů. Chybí zde sdílení výsledků mezi širším publikem. Proto Microsoft vytvořil platformu Power BI, kde můžete hostit, sdílet a zabezpečit interaktivní dashboardy pro ostatní. Power BI Desktop je aplikace, kde se právě tyto modely, vizuály a dashboardy tvoří. Power BI Desktop navíc používá stejné nástroje jako Excel. (Clark, 2017)

Krom již zmíněných rozdílů mezi Power BI Desktop a Power Pivot pro Excel lze říct, že obě aplikace mají téměř stejné funkcionality. Obě aplikace obsahují stejný engine, pouze vizuální stránka aplikaci je jiná (viz obr. 3). Dá se tedy říct, že pokud umíte pracovat s jednou z aplikací, zvládnete pracovat i s druhou. (Collie, 2015)

**Obr. 10: Ukázka vizuálního rozhraní- (nalevo) Power Pivot pro Excel a (napravo) Power BI Desktop**



Zdroj: COLLIE, Robert Clinton. *Dax formulas for power pivot: the excel pro's guide to mastering dax. 2nd edition. Uniontown, OH: Holy Macro! Books, 2015. ISBN 978-1-61547-039-6.*

### 3.8.3 DAX

Data Analysis Expressions (dále jen DAX) „je výkonný programovací jazyk Microsoft SQL Server Analysis Services (SSAS), implementovaný i v prostředí Power Pivot a Power BI pro pokročilé BI a SSBI aplikací.“ (Pour, 2018)

DAX je dotazovací a funkcionální jazyk. První vydání bylo v roce 2009 jako doplněk pro Microsoft Excel 2010. DAX není plnohodnotný programovací jazyk a nepodporuje všechny mechanismy řízení toku. Byl navržen tak, aby vylepšil modelování, vykazování a analýzu dat. Aktualizace a přidávání nových funkcí do tohoto jazyku jsou přidávány pravidelně. (Seamark, 2018)

Výhodou DAXu je vysoká efektivita pro práci s daty v multidimenzionálním prostředí. Obsahuje prostředky pro tvorbu analytických aplikací a uživatelům umožňuje lépe pochopit multidimenzionální problematiku. (Pour, 2018)

DAX je především brán jako funkcionální jazyk, protože primárně ke všem výpočtům používají funkce. Je možné využít funkce pro aritmetiku, úpravy textových řetězců, zpracování data a času a mnoho dalších funkcí. Funkce lze vnořit, takže výstup jedné funkce lze použít jako vstup jiné funkce, ale nelze vytvořit funkce vlastní. (Seamark, 2018)

Rozdílem od T-SQL v DAXu chybí koncept funkcí INSERT, UPDATE nebo DELETE pro úpravy dat v modelu. Jakmile v modelu existuje fyzická tabulka, nelze v ní data přidávat, měnit ani odebrat. Dalším rozdílem je, že DAX ukládá data ve sloupcích, nikoliv v řádcích. (Seamark, 2018)

Prostředí Power Pivot a Power BI obsahuje dva typy výpočtů: Calculated Columns a Measures. (Pour, 2018)

### **3.8.3.1 Calculated Column**

Calculated column (kalkulovaný sloupec) v tabulce vytváří nový sloupec na základě kalkulací z hodnot v rámci jedné řádky. Výpočet hodnot není závislý na průběžných aktivitách uživatele a měnících se filtrech. (Pour, 2018)

Důležitý koncept calculated columns je, že jejich kalkulace probíhá při zpracování databáze, až poté se ukládají do modelu. Jsou umístěny do tabulky, a tím pádem zabírají místo v paměti. (Ferrari, 2015)

### **3.8.3.2 Measure**

Measure (míra) se počítá za tabulkou nebo její podmnožinou a respektuje kontext daných buněk včetně nastavených filtrů a průřezů. Measure se nevztahuje k jednotlivým řádkům, vztahovat se může ke kalkulovaným sloupcům. (Pour, 2018)

Výstupem measure může být jedna hodnota, kterou lze použít ve vizuálech nebo jako součást dalších výpočtů. (Seamark, 2018)

Rozdíl mezi measures a calculated columns je tedy v kontextu zpracování. Measure se vyhodnňuje v rámci buňky pro DAX dotaz, calculated column na úrovni řádku tabulky, do kterého patří. (Ferrari, 2015)

### **3.8.3.3 Funkce CALCULATE**

*„Funkce CALCULATE se považuje za nejdůležitější, nejužitečnější a nejkomplexnější funkcí jazyka DAX.“* (Pour, 2018)

Umožňuje změnit filtr kontext tím, že vytvoří nový filtr kontext a poté výraz vyhodnotí podle tohoto nového kontextu. (Ferrari, 2015)

Ve funkci CALCULATE je povinný první výraz: measure. Zbylé nepovinné podmínky, resp. filtry nejsou povinné a nejsou početně omezené. (Ferrari, 2015)

```
CALCULATE( Measure; Condition1; Condition2; ... )
```

## 4 Praktická část

### 4.1 Představení projektu

Zemědělská společnost poskytla následující data a informace: seznam honů (polí) + jejich začlenění do oblastí; seznam pěstovaných plodin + jejich odrůdy; výnosy jednotlivých plodin a odrůd pěstovaných na polích v rámci pěti let + jejich datумы sklizně. Všechny výnosy byly po dohodě s vedením podniku před začátkem práce vynásobeny náhodnými koeficienty, aby zveřejnění práce neohrozilo žádným způsobem chod podniku. Dále budou do projektu přidány průměrné hodnoty ze sklizní plodin z Českého statistického úřadu (dále jen ČSÚ).

Cílem projektu bude představení tvorby procesu SSBI. Bude zde zahrnuta tvorba včetně práce s holými daty a výsledkem bude tvorba grafického řešení vhodného k původu dat. Ve vizualizacích bude znázorněno dynamické filtrování mezi. Výstupem projektu bude grafická vizualizace obsahující základní přehled grafů obsahující základní přehledy dané problematiky podniku.

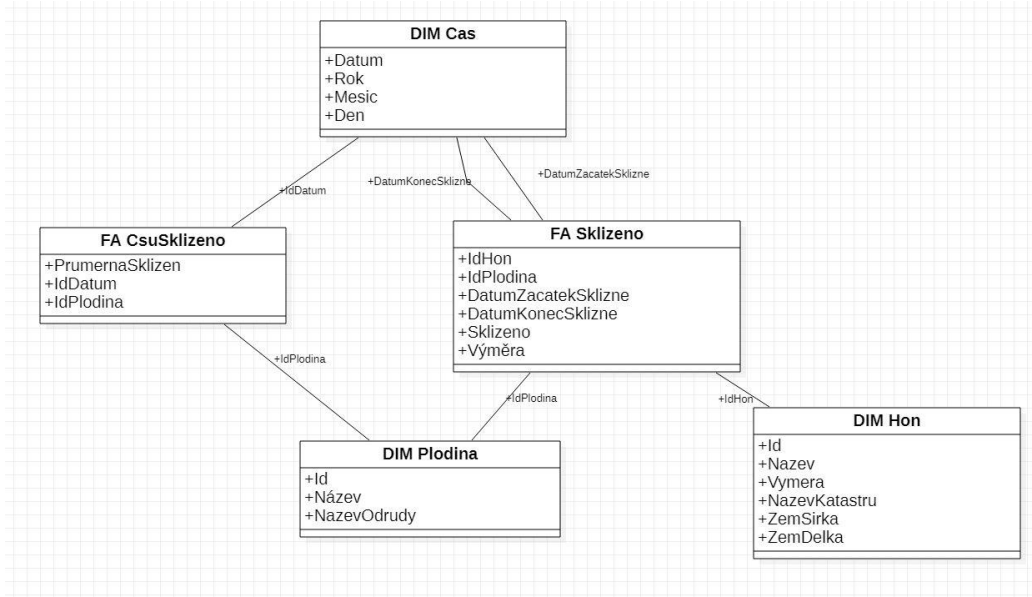
### 4.2 Návrh a analýza řešení

Data a informace je nutné sjednotit a vytvořit tabulkový model v Power BI. Před začátkem každého řešení je nutné stanovit návrh řešení, podle kterého bude proveden ETL proces. Podle povahy dat jsou tabulky jsou v rámci BI děleny na tzv. Dimenze a Fakta (viz. kapitola 3.3).

Tabulky je vhodné uspořádat do schématu. V teoretické části (viz. kapitola 3.3.1; 3.3.2) byly představeny schémata snowflake a star. Pro vytvoření návrhů byl využit program StarUML volně dostupný na adrese [HTTP://STARUML.IO](http://staruml.io).

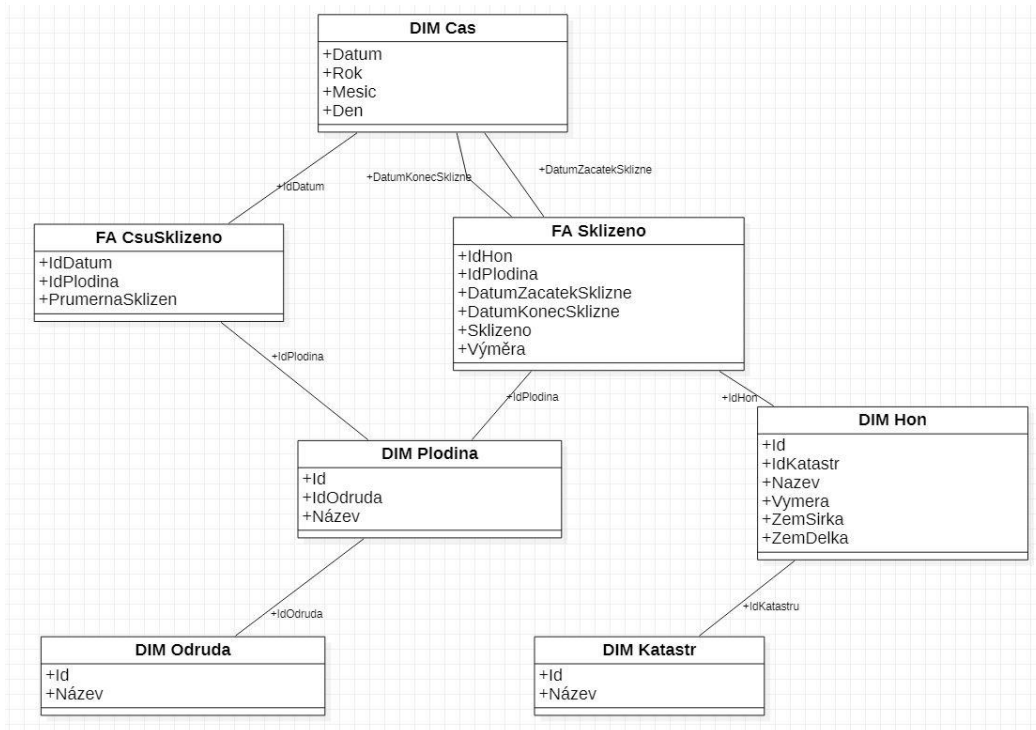


**Obr. 12: Ukázka návrhu podle schématu star**



*Zdroj: vlastní zpracování*

**Obr. 11: Ukázka návrhu podle schématu snowflake**



*Zdroj: vlastní zpracování*

## 4.3 ETL

Při tvorbě ETL je nutné dodržovat předem stanovený návrh řešení, veškeré informace a data uspořádat do stanovené struktury. Pro efektivnější práci byl použit SQL Server v rámci prostředí Microsoft SQL Server Management Studio (dále jen SSMS) a pro tvorbu dimenzí byly využity možnosti databázového jazyka T-SQL. Pro import dat z excelových souborů do SQL Serveru slouží T-SQL funkce OPENROWSET nebo OPENDATASOURCE. V projektu bude využita funkce OPENROWSET. Pro import excelového souboru je nutné mít nainstalovanou stejnou verzi (64-bit nebo 32-bit) sady Office a SQL Serveru, jinak nebude import umožněn. V projektu je použita 64-bitová verze. Dále je potřeba tzv. providera, který import provede. V projektu je jako provider použit Access Database Engine x64 verze 16.0. Všechny excelové soubory musejí mít zakázaný atribut „Jen pro čtení“.

Pro import z excelových sešitů bude sloužit databáze s názvem 'BI\_excel' a pro připravená a upravená data pro import do Power BI bude sloužit databáze 'BI'.

### 4.3.1 Fakta Sklizeň

Data jsou přehledně uspořádána v excelových souborech. Každý rok sklizní má vlastní sešit a jednotlivé listy obsahují data zvlášť pro každou plodinu (viz. obr. 13).

Obr. 13: Struktura sešitů a souboru sklizní

A	B	C	D	E
Název honu	Výměra (ha)	Odrůda	Sklizeno (t)	Sklizeno
U Studánky	26,12	Silver	77,88	1.7.
Za Hřbitovem Babice	16,6	In Vigor	64,05	1.7.
Proti Hřbitovu Babice	1,19	Arsenal	4,3	1.7.
Zelnice	7,8	DK Exception	31,7	4.7.
Březiny	57,4	Sesypy	210,4	2.7.
Nad Barchovskou cestou	34,52	Arsenal	126,28	1.7.-2.7.
Představenco	30	Hybrirock	83,6	30.6.
Kopa	20	Arabela	55,58	1.7.
Zádušník	21,66	Jumper	71,64	3.7.
U Letiště	9,32	Sherk	33,5	30.6.
Vrše	17,74	Sherk	57,04	3.7.
U Háje Kosice	17,03	Arabela	58,18	4.7.
Kuchyňka	51,63	DK Exception	208,12	2.7.-3.7.
U Háje Barchov	21,14	Kuga	76,38	2.7.
Ulehlá	18	Sesypy	59,32	3.7.

Zdroj: vlastní zpracování

Je potřeba provést import souborů (nejlépe najednou) do SQL serveru. Bohužel je funkce OPENROWSET statická a její struktura umožňuje import pouze jednoho listu z jednoho sešitu. Z tohoto důvodu je potřeba vytvořit dynamický dotaz, kterým se data ze sešitů namapují do jedné tabulky. Pro vytvoření dotazu je vhodné v SQL Serveru vytvořit pomocnou tabulku s plodinami, které se v excelových souborech vyskytují. Pomocná tabulka bude sloužit pro vytvoření algoritmického cyklu.

### **Blok 1: Dotaz pro vytvoření a naplnění tabulky pro pomocné názvy plodin**

```
--vytvoření pomocné tabulky
CREATE TABLE [Bi_excel].[dbo].[plodiny_pom] ([id] INT IDENTITY (1,1),
[nazev] VARCHAR(50))
GO

--naplnění tabulky
INSERT INTO [Bi_excel].[dbo].[plodiny_pom] (nazev)
VALUES ('pšenice ozimá'), ('ječmen jarní'), ('hrách'), ('vojtěška'),
('ječmen ozimý'), ('řepka ozimá')
GO
```

*Zdroj: vlastní zpracování*

Nejdříve proběhne deklarace tabulky, do které budou data z excelových sešitů naimportována.

### **Blok 2: Dotaz pro vytvoření tabulky pro data z excelu sklizní**

```
--vytvoření tabulky, do které budou importována data z excelových sešitů
CREATE TABLE [Bi_excel].[dbo].[sklizen]
([Název honu] varchar(50), [Výměra (ha)] varchar(50), [Odrůda]
varchar(50), [Sklizeno (t)] varchar(50), [Sklizeno] varchar (50), [plodina]
varchar(50), [rok] int);
```

*Zdroj: vlastní zpracování*

V dalších krocích bude ukázáno několik částí dotazu.

Bude potřeba deklarovat dvě proměnné (@minRok a @maxRok) obsahující min a max rok souborů, pro které bude import proveden. Další 2 proměnné (@minID a @maxID) budou sloužit pro určení min a max id z pomocné tabulky vytvořené v minulém kroku. Díky znalosti id bude možné deklarovat dynamickou proměnnou obsahující název plodiny

### Blok 3: Část bloku 6 – proměnné

```
--proměnné podle roků u souborů sklizní
DECLARE @minRok int = 2013
DECLARE @maxRok int = 2018

--proměnné podle id z pomocné tabulky
DECLARE @minID int = 0
DECLARE @maxID int = (SELECT id FROM [Bi_excel].[dbo].[plodiny_pom] WHERE
id=(SELECT MAX(id) FROM [Bi_excel].[dbo].[plodiny_pom]))
```

*Zdroj: vlastní zpracování*

Dále je proveden dvojitý cyklus, ve kterém dochází k několikanásobnému volání funkce OPENROWSET, která se dynamicky pro každé volání mění a postupně importuje všechna data z excelových sešitů do databáze.

První volání cyklu bude porovnávat proměnné @minRok a @maxRok a po každém provedení cyklu se proměnná @minRok zvýší o jedničku a @minID vynuluje, aby druhý cyklus mohl proběhnout pokaždé, co bude probíhat první. Druhý cyklus porovnává proměnné @minID a @maxID z pomocné tabulky. Uvnitř druhého cyklu se proměnná @minID zvýší o jedničku a deklaruje se proměnná @plodinaNazev obsahující název aktuální plodiny.

### Blok 4: Část bloku 6 – cykly

```
--cyklus importující všechny soubory sklizní, podle nastavených proměnných
WHILE @minRok <= @maxRok
BEGIN
    WHILE @minID < @maxID
    BEGIN
        SET @minID = @minID + 1
        --název aktuální plodiny
        DECLARE @plodinaNazev varchar(50) = (SELECT nazev FROM
[test_import].[dbo].[plodiny_pom] WHERE id = @minID)

        .
        .
        .

    END
    SET @minID = 0
    SET @minRok = @minRok + 1
END
```

*Zdroj: vlastní zpracování*

Druhý cyklus obsahuje volání funkce OPENROWSET. Struktura statické funkce pro import dat z excelového souboru je znázorněna níže:

```
SELECT a.* FROM OPENROWSET( Provider, Data source, Sheet selection ) AS a
```

Pro vytvoření dynamického SQL dotazu se využívá proměnné @sql, do které se umístí text, a do tohoto textu se dosazují předem deklarované proměnné, které se v rámci cyklů průběžně mění. Důležitá je zde především proměnná s názvem @plodinaNazev, pomocí které se určuje, z jakého sešitu import probíhá. Dotaz umístěný v proměnné @sql spustí funkce EXEC.

### Blok 5: Část bloku 6 – import pomocí OPENROWSET

```
DECLARE @sql varchar(max) = '  
    INSERT INTO [Bi_excel].[dbo].[sklizeni]  
    SELECT b.[Název honu], b.[Výměra (ha)], b.[Odrůda], b.[Sklizeno  
(t)], b.[Sklizeno], '''+@plodinaNazev+''', '+CAST(@minRok as varchar(50))+'  
    FROM OPENROWSET(  
        'Microsoft.ACE.OLEDB.16.0'  
        , 'Excel 12.0;Database=C:\BI\Sklizeň '+CAST(@minRok as  
varchar(50))+'.xlsx;HDR=YES'  
        , 'SELECT * FROM[''+@plodinaNazev+'$]') as b'  
EXEC(@sql)
```

*Zdroj: vlastní zpracování*

Celý dotaz vypadá následovně:

### Blok 6: Dotaz pro import všech dat z excelových sešitů sklizní

```
DECLARE @minRok int = 2013
DECLARE @maxRok int = 2018

DECLARE @minID int = 0
DECLARE @maxID int = (SELECT id FROM [Bi_excel].[dbo].[plodiny_pom] WHERE
id=(SELECT MAX(id) FROM [Bi_excel].[dbo].[plodiny_pom]))

WHILE @minRok <= @maxRok
BEGIN
    WHILE @minID < @maxID
    BEGIN
        SET @minID = @minID + 1
        DECLARE @plodinaNazev varchar(50) = (SELECT nazev FROM
[test_import].[dbo].[plodiny_pom] WHERE id = @minID)

        DECLARE @sql varchar(max) = '
INSERT INTO [Bi_excel].[dbo].[sklizeni]
SELECT b.[Název honu], b.[Výměra (ha)], b.[Odrůda],
b.[Sklizeno (t)], b.[Sklizeno], '''+@plodinaNazev+''', '+CAST(@minRok as
varchar(50))+
FROM OPENROWSET(
'Microsoft.ACE.OLEDB.16.0'
,'Excel 12.0;Database=C:\BI\Sklizeň '+CAST(@minRok as
varchar(50))+'.xlsx;HDR=YES'
,'SELECT * FROM[''+@plodinaNazev+'$]') as b'
EXEC(@sql)
    END
    SET @minID = 0
    SET @minRok = @minRok + 1
END
```

*Zdroj: vlastní zpracování*

Data jsou nyní naimportována v pomocné databázi. Dalším krokem je vyčištění těchto dat a jejich přesunutí do konečné databáze. Data obsahují NULL hodnoty v řádcích názvu honu, výměře a odrůdě. To je způsobeno, různými ručními úpravami zdrojových excelových souborů. Všechny tyto řádky je nutné odstranit.

### Blok 7: Dotaz pro odstranění prázdných řádků v tabulce sklizní

```
--odstranění prázdných řádků
DELETE FROM [Bi_excel].[dbo].[sklizeni]
WHERE [Název honu] IS NULL AND
[Výměra (ha)] IS NULL AND
[Odrůda] IS NULL AND
[Sklizeno (t)] IS NULL;
GO
```

*Zdroj: vlastní zpracování*

Datумы jsou zapsány jednotlivě, v intervalech oddělenými pomlčkou nebo datum chybí. Je nutné všechny možnosti rozlišit a sepsat do vhodné podmínky. Vzhledem k existenci rozsahů, bude sloupec s datумы rozdělen do dvou: 'Začátek sklizně' a 'Konec sklizně'. Pro datумы bez intervalu budou oba řádky stejné. Datумы s rozsahem budou rozděleny pravou a levou část. Pokud datum nemá hodnotu bude vzhledem k znalosti roku použit formát 31.12.ROK.

Transformace dat sklizní do konečné tabulky bude opět rozdělena na několik částí.

Nejprve je potřeba vytvořit tabulku, do které budou data transformována.

### **Blok 8: Dotaz pro vytvoření tabulky pro konečná data**

```
--vytvoření konečné tabulky pro data sklizeň
CREATE TABLE [BI].[dbo].[sklizzen] (
[Název honu] VARCHAR(50),
[Výměra (ha)] DECIMAL(15,2),
[Odrůda] VARCHAR(50),
[Sklizeno (t)] DECIMAL(15,2),
[Začátek sklizně] DATE,
[Konec sklizně] DATE,
[Plodina] VARCHAR(50)
)
GO
```

*Zdroj: vlastní zpracování*

Není-li potřeba data žádným způsobem editovat, lze použít funkci pouze funkce pro ořezání výrazu od mezer TRIM. Složitější transformace byla u datumů. Bylo zmíněno několik zmíněných možností formátu výchozích dat. Aby došlo k pokrytí všech možností musejí být vytvořeny podmínky.

'Pokud není datum prázdné, proved' dotaz, jinak do datumu přiřad' 31.12.ROK.'

### **Blok 9: Část bloku 13 – Podmínka pro prázdná datumová pole**

```
IIF (a.[Sklizeno] IS NOT NULL,
/dotaz/,
DATEFROMPARTS(a.[rok], 12, 31)
),
```

*Zdroj: vlastní zpracování*

‘Pokud má být datum interval (je v něm pomlčka), proved’ dotaz, jinak vytvoř datum.’ Spojením části data a roku pomocí funkce CONCAT vznikne varchar, který stačí konvertovat do date. Zkratka pro konverzi varcharu do date je v našem případě 104 (DD.MM.RRRR)

### Blok 10: Část bloku 13 – Podmínka pro datumový interval

```
CASE WHEN a.[Sklizeno] LIKE '%-%'  
THEN /dotaz/  
ELSE CONVERT(DATE, CONCAT(TRIM(a.[Sklizeno]),a.[rok]), 104)  
END
```

*Zdroj: vlastní zpracování*

Nyní už zbývá vytvořit datum z levé nebo pravé části intervalu. Pro výběr potřebné části intervalu byla zvolena funkce SUBSTRING, do které je vložen nultý a index pomlčky (pro levou část) nebo index pomlčky +1 a délka výrazu. Levá část bude použita pro začátek sklizně, pravá pro konec sklizně.

Dotaz pro levou část vypadá následovně:

### Blok 11: Část bloku 13 – Tvorba datumu z levé části intervalu

```
CONVERT(DATE,  
CONCAT(  
SUBSTRING(TRIM(a.[Sklizeno]), 0, CHARINDEX('-',  
TRIM(a.[Sklizeno]))),  
a.[rok]),  
104)
```

*Zdroj: vlastní zpracování*

Dotaz pro pravou část vypadá následovně:

### Blok 12: Část bloku 13 – Tvorba datumu z pravé části intervalu

```
CONVERT(DATE,  
CONCAT(  
SUBSTRING(TRIM(a.[Sklizeno]), CHARINDEX('-',  
' ,TRIM(a.[Sklizeno]))+1, LEN(TRIM(a.[Sklizeno]))),  
a.[rok]),  
104)
```

*Zdroj: vlastní zpracování*



Celý dotaz vypadá následovně:

### Blok 13: Dotaz pro transformaci dat sklizní do konečné podoby

```
INSERT INTO [BI].[dbo].[sklizen]
SELECT
    TRIM(a.[Název honu]),
    TRIM(a.[Výměra (ha)]),
    TRIM(a.[Odrůda]),
    TRIM(a.[Sklizeno (t)]),

    IIF (a.[Sklizeno] IS NOT NULL,
        CASE WHEN a.[Sklizeno] LIKE '%-%'
        THEN
            CONVERT(
                DATE,
                CONCAT(
                    SUBSTRING(TRIM(a.[Sklizeno]), 0,
CHARINDEX('-', TRIM(a.[Sklizeno]))),
                    a.[rok]),
                104)
            ELSE CONVERT(DATE, CONCAT(TRIM(a.[Sklizeno]),a.[rok]), 104)
        END,
        DATEFROMPARTS(a.[rok], 12, 31)
    ),

    IIF (a.[Sklizeno] IS NOT NULL,
        CASE WHEN a.[Sklizeno] LIKE '%-%'
        THEN
            CONVERT(
                DATE,
                CONCAT(
                    SUBSTRING(TRIM(a.[Sklizeno]), CHARINDEX('-',
', TRIM(a.[Sklizeno]))+1, LEN(TRIM(a.[Sklizeno]))),
                    a.[rok]),
                104)
            ELSE CONVERT(DATE, CONCAT(TRIM(a.[Sklizeno]),a.[rok]), 104)
        END,
        DATEFROMPARTS(a.[rok], 12, 31)
    ),

    TRIM(a.[plodina])
FROM [Bi_excel].[dbo].[sklizen] a
```

*Zdroj: vlastní zpracování*

### 4.3.2 Dimenze Hony

Struktura poskytnutého sešitu s hony a katastry je znázorněna na obrázku níže.

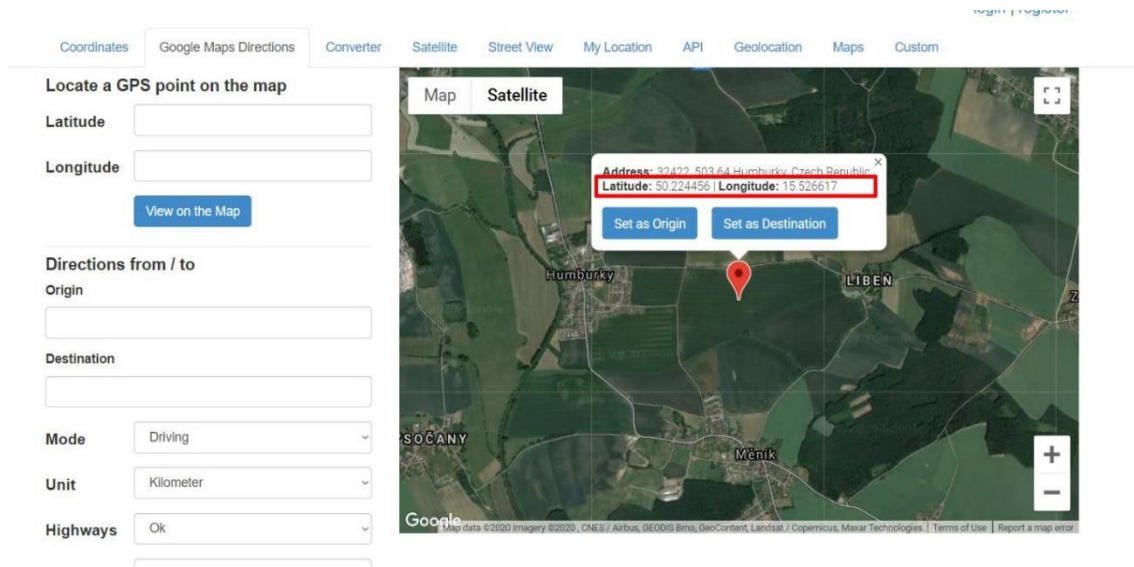
**Obr. 14: Struktura sešitu s hony a katastry**

	A	B	C
1	<b>Název honu</b>	<b>Výměra</b>	<b>Katastr</b>
2	Bělizna	14,60	Humburky
3	Skleník	9,83	Humburky
4	Za Kravínem Humburky	15,36	Humburky
5	Chlum	23,83	Humburky
6	Pod křížem	19,74	Humburky
7	Trníčko	49,45	Humburky
8	Stezky	22,29	Humburky
9	Vokálova	18,68	Humburky
10	Za Skleníkem	1,23	Humburky
11	U Koláče	2,74	Humburky
12	Pod továrnou	2,38	Humburky
13	Malá strana	0,46	Humburky
14	Závodiště	10,75	Humburky
15	Pořadovo	5,36	Měník
16	Letiště	9,36	Měník
17	Za Válkem	13,68	Měník
18	Komisárka	9,91	Měník
19	Konšel	11,19	Měník
20	Hladíkovo	3,86	Měník
21	Stejskalovo	11,97	Měník
22	Pod skladovákem	18,12	Měník
23	Pod vepřínem	4,74	Měník
24	Lán	22,17	Měník
25	Pod Kavkou	1,72	Měník
26	Široká	29,15	Měník

*Zdroj: vlastní zpracování*

K excelovému sešitu byla dodána i mapa všech honů. Díky znalosti, kde se jednotlivé hony nacházejí bylo vhodné určit jejich zeměpisnou šířku a délku, a využít je v projektu. K určení šířky a délky lze využít volně dostupné webové aplikace na adrese <HTTPS://WWW.GPS-COORDINATES.NET/DRIVING-DIRECTIONS> založenou na aplikaci Mapy Google. Pro získání souřadnic honů se použije jako výchozí bod střed pozemku (odhadnutý podle oka), aplikace poté zobrazí zeměpisné souřadnice (viz obr. 15), které se doplní do excelového souboru k danému honu, konkrétně do sloupců 'Šířka' a 'Délka'.

Obr. 15: Ukázka použití nástroje pro generování zeměpisných souřadnic



Zdroj: vlastní zpracování z webu: [www.gps-coordinates.net](http://www.gps-coordinates.net)

Dotaz, ve kterém se vytvoří tabulka pro import honů je znázorněn níže.

#### Blok 14: Dotaz pro tvorbu tabulky a import dat honů

```
--vytvoření pomocné tabulky
CREATE TABLE [BI_excel].[dbo].[hony]
    ([Název honu] VARCHAR(50),[Výměra] VARCHAR(50),[Katastr]
    VARCHAR(50),[Šířka] VARCHAR(50),[Délka] VARCHAR(50))

--vlození dat ze sešitu pomocí funkce OPENROWSET
INSERT INTO [BI_excel].[dbo].[hony]
SELECT b.[Název honu], b.[Výměra], b.[Katastr], b.[Šířka], b.[Délka]
FROM OPENROWSET(
    'Microsoft.ACE.OLEDB.16.0'
    , 'Excel 12.0;Database=C:\BI\Hony.xlsx;HDR=YES'
    , 'SELECT * FROM [Sheet1$]') as b
```

Zdroj: vlastní zpracování

Při kopírování zeměpisných souřadnic se do excelového sešitu vložil i speciální znak vypadající jako mezera. Ten je potřeba odstranit pomocí funkce REPLACE, protože funkce TRIM tento znak nepodporuje a tím pádem ho neodstraní. Dotaz pro transformaci dat je znázorněn v bloku 15.

### Blok 15: Dotaz pro transformaci dat do konečné tabulky honů

```
--vytvoření konečné tabulky
CREATE TABLE [BI].[dbo].[hony]
  ([ID] INT IDENTITY(1,1), [Název honu] VARCHAR(50), [Výměra]
DECIMAL(15,2), [Katastr] VARCHAR(50), [Šířka] DECIMAL(15,6), [Délka]
DECIMAL(15,6))

--přesunutí a mapování dat z pomocné tabulky
INSERT INTO [BI].[dbo].[hony] ([Název honu], [Výměra], [Katastr], [Šířka],
[Délka])
SELECT
  TRIM(a.[Název honu]),
  TRIM(a.[Výměra]),
  TRIM(a.[Katastr]),
  TRY_CONVERT(DECIMAL(15,6), REPLACE(a.[Šířka], ' ', '')),
  TRY_CONVERT(DECIMAL(15,6), REPLACE(a.[Délka], ' ', ''))
FROM [Bi_excel].[dbo].[hony] AS a
```

*Zdroj: vlastní zpracování*

Dalším krokem je potřeba do konečné tabulky se sklizní nahradit 'Název honu' za sloupec obsahující id\_honu.

### Blok 16: Dotaz pro přidání sloupce do tabulky sklizní pro id\_honu

```
--přidání sloupce
ALTER TABLE [BI].[dbo].[Sklizen]
ADD id_honu int;
GO

--přidání hodnot id podle shodných názvů
UPDATE [BI].[dbo].[Sklizen]
SET [id_honu] = a.Id
FROM [BI].[dbo].[Hony] a
WHERE a.[NÁZEV HONU] = [BI].[dbo].[Sklizen].[Název honu]
--odstranění nepotřebného sloupce s názvem
ALTER TABLE [BI].[dbo].[sklizen]
DROP COLUMN [Název honu]
GO
```

*Zdroj: vlastní zpracování*

### 4.3.3 Dimenze Plodiny

Samostatný soubor s plodinami nebyl dodán. Je potřeba vystačit s daty v souborech sklizní, kde jsou všechny informace o plodinách i odrůdách. Využije se již hotové tabulky 'Sklizeň', kde se pomocí dotazu s výrazem DISTINCT získá výpis jedinečných názvů plodin a odrůd.

Data o sklizních z ČSÚ nemají odrůdy, proto je vhodné již nyní připravit prázdná pole s odrůdou, kam vložíme hodnotu 'empty', aby mohlo dojít k napojení ID i mezi těmito tabulkami. Tak jako s hony, se nahradí název plodiny za id\_plodiny.

#### Blok 17: Dotaz pro tvorbu tabulky a jejím naplněním plodinami

```
--vytvoření tabulky pro dimenzi plodin
CREATE TABLE [BI].[dbo].[plodiny] ([ID] INT IDENTITY(1,1), [Odrůda]
VARCHAR(50), [Plodina] VARCHAR(50))
GO

--vlození jedinečných záznamů plodin a odrůd z tabulky sklizeň
INSERT INTO [BI].[dbo].[plodiny] ([Odrůda], [Plodina])
SELECT DISTINCT a.Odrůda, a.Plodina
FROM [BI].[dbo].[Sklizen] AS a
WHERE a.Odrůda IS NOT NULL
GO

--vlození jedinečných plodin s prázdnou odrůdou
INSERT INTO [BI].[dbo].[plodiny] ([Odrůda], [Plodina])
SELECT DISTINCT '-empty-', a.[plodina]
FROM [BI].dbo.[Sklizen] a;
GO

--přidání sloupce pro id plodiny do sklizně
ALTER TABLE [BI].dbo.[Sklizen]
ADD id_plodiny int;
GO

--naplnění sloupce pro id plodiny
UPDATE [BI].dbo.[Sklizen]
SET [id_plodiny] = a.Id
FROM [BI].dbo.plodiny a
WHERE a.Odrůda = [BI].dbo.[Sklizen].Odrůda AND a.Plodina =
[BI].dbo.[Sklizen].plodina;
GO

--odstranění nepotřebného sloupce s názvem
ALTER TABLE [BI].[dbo].[sklizen]
DROP COLUMN [Plodina],[Odrůda]
GO
```

*Zdroj: vlastní zpracování*

### 4.3.4 Fakta Sklizeň ČSÚ

Stažené soubory byly v rámci projektu přejmenovány podle let, aby mohlo dojít k jejich jednoznačné identifikaci. Data pro jeden rok jsou uložena do dvou sešitů (viz. obr. 16). Všechny sešity nemají v prvním řádku název sloupce, proto dojde k označení rozpětí všech sloupců obsahující potřebná data (A:E).

**Obr. 16: Struktura sešitů a souborů sklizní ČSÚ**

Plodina	Plocha (ha) Area (ha)	Výnos (t/ha) Yield (t/ha)	Sklizeň (t) Harvest (t)	Crop
a	1	2	3	b
Zrniny úhrnem	1 373 933	5,13	7 050 434	Grain crops, total
Obiloviny celkem	1 338 780	5,21	6 970 919	Cereals, total
základní obiloviny celkem	1 250 441	5,17	6 468 845	Basic cereals, total
pšenice celkem	819 690	5,39	4 417 841	Wheat, total
pšenice ozimá	773 678	5,46	4 227 344	Winter wheat
pšenice jarní	46 012	4,14	190 497	Spring wheat
žito	25 355	4,74	120 180	Rye
ječmen celkem	324 724	4,95	1 606 034	Barley, total
Pícniny na orné půdě celkem - ser	468 604	8,47	3 967 377	Arable fodder crops, as hay
Jednoleté pícniny celkem	275 405	25,62	7 055 027	Annual fodder crops, total
kukuřice na zeleno a siláž	224 105	29,84	6 686 996	Green and silage maize
obiloviny na zeleno (mimo kukuřici)	16 349	19,48	318 423	Cereals harvested green (excl. maize)
jednoleté luskoviny na zeleno	21 280	14,26	303 437	Annual leguminous plants harvested green
ostatní jednoleté pícniny	13 671	9,94	135 851	Other annual fodder crops
Víceleté pícniny na orné půdě	193 199	5,50	1 062 366	Perennial fodder crops, total
jetel luční (červený)	60 020	5,77	346 103	Red clover
vojtěška	65 412	6,40	418 333	Lucerne
ostatní víceleté pícniny	34 588	5,53	191 342	Other perennial fodder crops
dočasné travní porosty a pastviny	33 180	3,21	106 587	Temporary grasses and grazings
Zelenina konzumní celkem <sup>9)</sup>	10 404	x	198 863	Fresh vegetables, total
celer bulvový	322	26,25	8 445	Celeriac
mrkev	699	38,15	26 653	Carrots

Zdroj: vlastní zpracování

Následující skript v bloku provede import celého sešitu v rozpětí potřebných sloupců.

Názvy souborů jsou opět rozlišeny jednotlivými roky, proto deklaruje proměnné @minRok a @maxRok. Import bude rozdělen také do dvou proměnných @sql1 a @sql2 kvůli dvěma souborům pro jeden rok. K názvu listu ve funkci OPENROWSET bude ještě připsáno rozpětí A:E, v jakém mají být data naimportována.

### Blok 18: Dotaz pro import excelových sešitů sklizní z ČSÚ

```
--vytvoření pomocné tabulky, pro import dat z excelových sešitů
CREATE TABLE [BI_excel].[dbo].[sklizen_csu]
  ([Plodina] VARCHAR(MAX), [Plocha] VARCHAR(MAX), [Výnos]
  VARCHAR(MAX), [Sklizeň] VARCHAR(MAX), [Crop] VARCHAR(MAX), [Rok] int)

DECLARE @minRok int = 2013
DECLARE @maxRok int = 2018

WHILE @minRok <= @maxRok
BEGIN
    DECLARE @sql1 varchar(max) = '
    INSERT INTO [BI_excel].[dbo].[sklizen_csu]
    SELECT *, '+CAST(@minRok as varchar(50))+
    FROM OPENROWSET(
    'Microsoft.ACE.OLEDB.16.0'
    , 'Excel 12.0;Database=C:\BI\czso\' +CAST(@minRok as
    varchar(50))+'.xlsx;HDR=YES'
    , 'SELECT * FROM[List1$A:E]')' -- A:E značí rozpětí sloupců,
    které se importují

    DECLARE @sql2 varchar(max) = '
    INSERT INTO [BI_excel].[dbo].[sklizen_csu]
    SELECT *, '+CAST(@minRok as varchar(50))+
    FROM OPENROWSET(
    'Microsoft.ACE.OLEDB.16.0'
    , 'Excel 12.0;Database= C:\BI\czso\' +CAST(@minRok as
    varchar(50))+ ' (2).xlsx;HDR=YES'
    , 'SELECT * FROM[List1$A:E]')'
    EXEC(@sql1)
    EXEC(@sql2)
    SET @minRok = @minRok + 1
END
```

*Zdroj: vlastní zpracování*

Při bližším prozkoumání dat v ČSÚ souborech si lze všimnout, že místo řepky jarní a ozimé se využívá průměr z těchto dvou plodin označený jako 'řepka'. Pro účely v projektu tento rozdíl nebude brán zřetel a 'řepka' z ČSÚ se přejmenuje na 'řepka ozimá'. Dále bude 'hrách setý' a 'hrách na zrno' přejmenován na 'hrách', aby odpovídal konvenci v projektu.

### Blok 19: Dotaz pro přejmenování názvů plodin z ČSÚ do jednotné konvence

```
--přejmenování dat podle konvence v projektu
UPDATE [Bi_excel].[dbo].[sklizen_csu]
SET [Plodina] = 'řepka ozimá'
WHERE TRIM([Plodina]) LIKE 'řepka'

UPDATE [Bi_excel].[dbo].[sklizen_csu]
SET [Plodina] = 'hrách'
WHERE TRIM([Plodina]) LIKE 'hrách setý' OR TRIM([Plodina]) LIKE 'hrách na
zrno'
```

*Zdroj: vlastní zpracování*

Následuje výběr a mapování potřebných dat do konečné tabulky pro sklizně z ČSÚ a nahrazení názvu plodiny za id\_plodiny.

### Blok 20: Dotaz pro transformaci a výběr potřebných dat z ČSÚ

```
--vytvoření konečné tabulky
CREATE TABLE [BI].[dbo].[sklizen_csu]
    ([Plodina] varchar(MAX), [Výnos] DECIMAL(15,5), [Datum] DATE)

--vlození dat do konečné tabulky
INSERT INTO [BI].[dbo].[sklizen_csu] ([Plodina], [Výnos], [Datum])
SELECT
    TRIM([Plodina]),
    TRIM([Výnos]),
    DATEFROMPARTS([Rok], 12, 31)
FROM [Bi_excel].[dbo].[sklizen_csu]
WHERE TRIM([Plodina]) in (SELECT Plodina FROM [Bi].[dbo].[plodiny])

ALTER TABLE [BI].[dbo].[sklizen_csu]
ADD id_plodiny int;
GO

--přidání dat do sloupce pro id plodin
UPDATE [BI].[dbo].[sklizen_csu]
SET [id_plodiny] = a.Id
FROM [BI].[dbo].[plodiny] AS a
WHERE a.Plodina = [BI].[dbo].[sklizen_csu].Plodina AND a.odrůda = '-empty-'
```

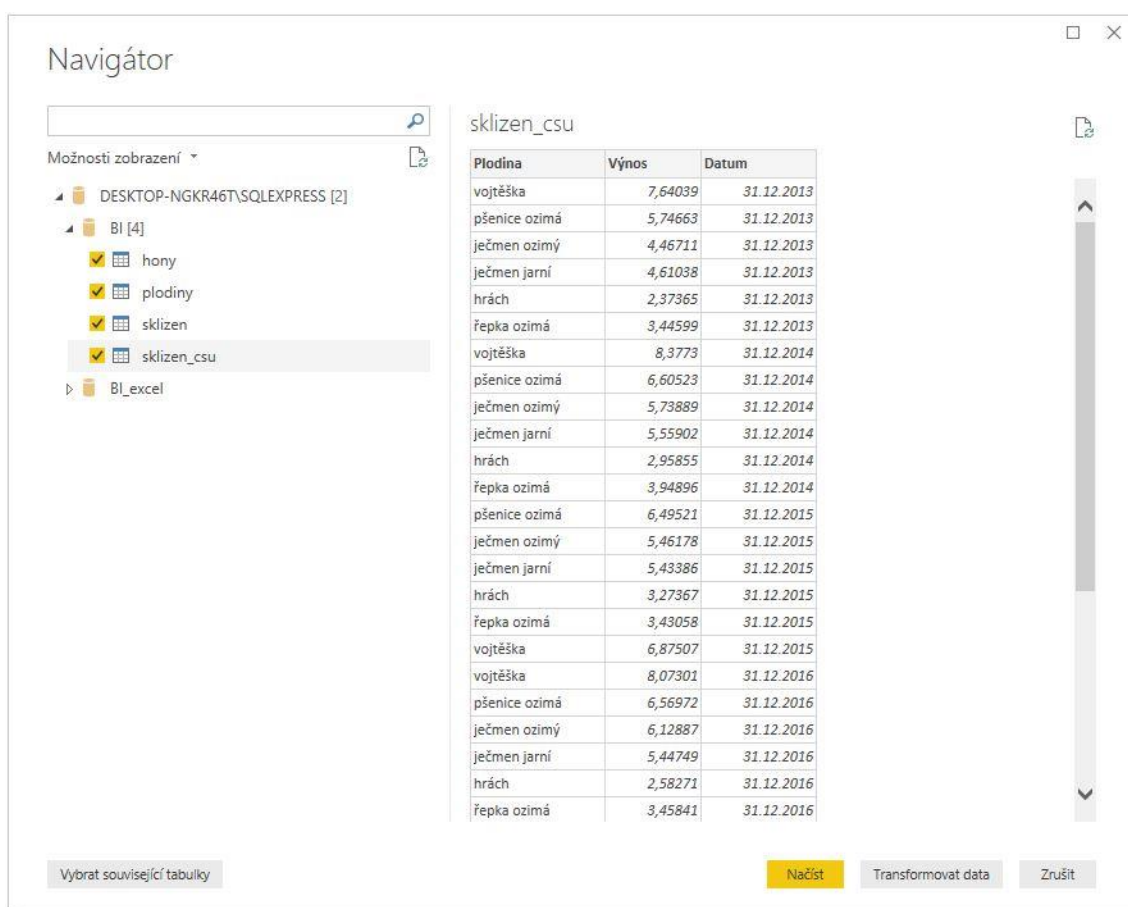
*Zdroj: vlastní zpracování*



### 4.3.5 Načtení dat do Power BI

Nyní jsou všechna data připravena pro import do Power BI. Nejdříve je v Power BI potřeba vytvořit nový projekt. Pro import slouží přednastavené funkce Power BI, které vytvoří spojení databáze s Power BI a poté stačí vybrat tabulky potřebné k importu. (viz. obr. 17).

Obr. 17: Ukázka načtení tabulek z databáze



Zdroj: vlastní zpracování

### 4.3.6 Dimenze Čas

Pro tvorbu časové dimenze je vhodné využít funkci jazyku DAX s názvem CALENDAR.

Tabulka se vytvoří po přepnutí do záložky 'Modelování' a stiskem tlačítka 'Nová tabulka'. Do nabídnutého textového pole stačí napsat 'DIM ČAS = CALENDAR(DATE(2013;1;1); DATE(2030;1;1))' a po stisknutí tlačítka na klávesnici ENTER se vytvoří tabulka s jedním sloupcem. Sloupec je vhodné přejmenovat na 'Datum' a změnit jeho datový typ na 'Datum'.

Den, měsíc, rok a další časové údaje v našem případě není potřeba generovat, protože Datovým typem 'Datum' se vytvoří hierarchie časových vrstev, podle které lze data efektivně filtrovat.

### **4.3.7 Relace v modelu**

Poslední částí ETL procesu je dokončení modelu vytvořením vazeb mezi tabulkami. V levém panelu stačí kliknout na panel 'Model', přepnout hlavní panel do 'Modelování' a možností 'Spravovat relace'. Nyní je potřeba nastavit relace podle návrhu. Do výběru 'Z: Tabulka (Sloupec)' se vybírají fakta a do výběru 'Do: Tabulka (Sloupec)' se vybírají dimenze. Relace pro aktuální projekt jsou znázorněny na následujícím obrázku.

**Obr. 19: Ukázka relací pro model v Power BI**

Spravovat relace

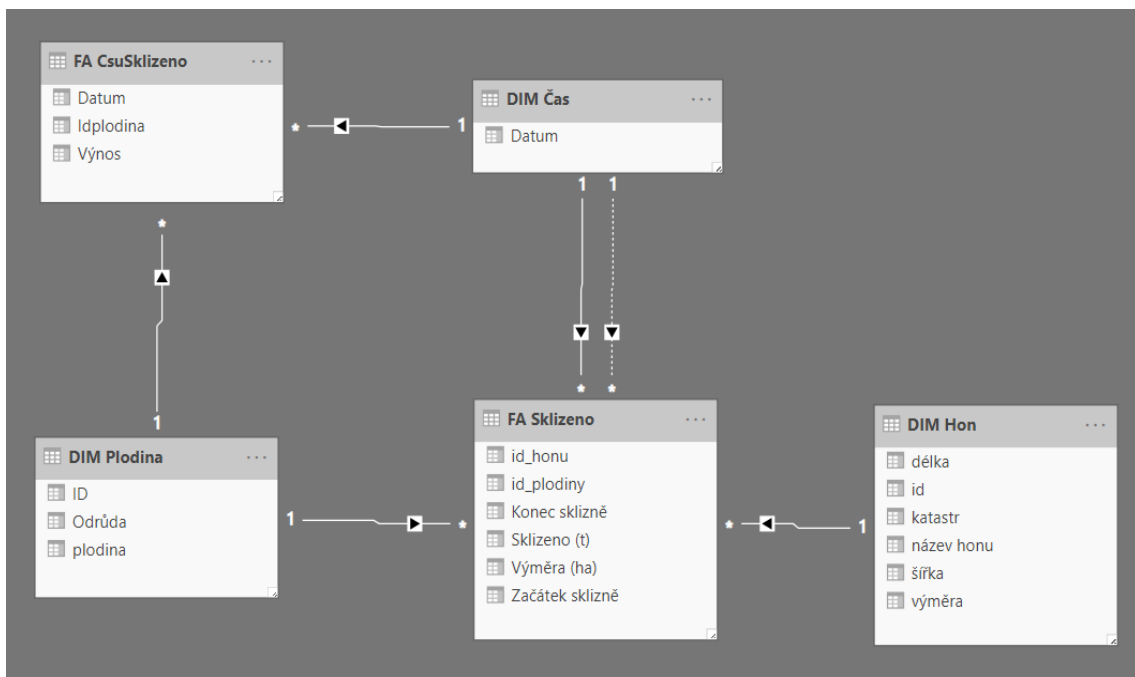
Aktivní	Z: Tabulka (Sloupec)	Do: Tabulka (Sloupec)
<input checked="" type="checkbox"/>	FA CsuSklizeno (Datum)	DIM Čas (Datum)
<input checked="" type="checkbox"/>	FA CsuSklizeno (Idplodina)	DIM Plodina (ID)
<input checked="" type="checkbox"/>	FA Sklizeno (id_honu)	DIM Hon (id)
<input checked="" type="checkbox"/>	FA Sklizeno (id_plodiny)	DIM Plodina (ID)
<input type="checkbox"/>	FA Sklizeno (Konec sklizně)	DIM Čas (Datum)
<input checked="" type="checkbox"/>	FA Sklizeno (Začátek sklizně)	DIM Čas (Datum)

Nová... Automatické rozpoznávání... Upravit... Odstranit

Zavřít

*Zdroj: vlastní zpracování*

**Obr. 18: Konečná podoba modelu v Power BI**



*Zdroj: vlastní zpracování*

## 4.4 Tvorba measures

Measures (viz. kapitola 3.8.3.2) použité v projektu:

- Prvním krokem by měla být kalkulace celkového výnosu podniku, dále díky znalosti výnosu kalkulovat výnos minulého roku

```
Výnos (t/ha) = SUM('FA SKLIZEŇ'[Sklizeno (t)]) / SUM('FA SKLIZEŇ'[Výměra (ha)])
```

```
Výnos minulý rok = CALCULATE([Výnos (t/ha)]; PREVIOUSYEAR('DIM ČAS'[Datum]))
```

- Pro sklizeň a osev (výměra) je vhodné také provést kalkulaci pro minulý rok

```
Výměra minulý rok = CALCULATE(SUM('FA SKLIZEŇ'[Výměra (ha)]); PREVIOUSYEAR('DIM ČAS'[Datum]))
```

```
Sklizeň minulý rok = CALCULATE(SUM('FA SKLIZEŇ'[Sklizeno (t)]); PREVIOUSYEAR('DIM ČAS'[Datum]))
```

- Časová osa sklizní lze zobrazit pomocí grafu, je také vhodné uživateli zobrazit datum začátku a konce sklizně pro rychlejší orientaci

```
První den sklizně = FIRSDATE('FA SKLIZEŇ'[Začátek sklizně])
```

```
Poslední den sklizně = LASTDATE('FA SKLIZEŇ'[Konec sklizně])
```

## 4.5 Vizualizace dat

V Power BI se vizualizace provádějí v záložce 'Sestava' umístěné na levém panelu.

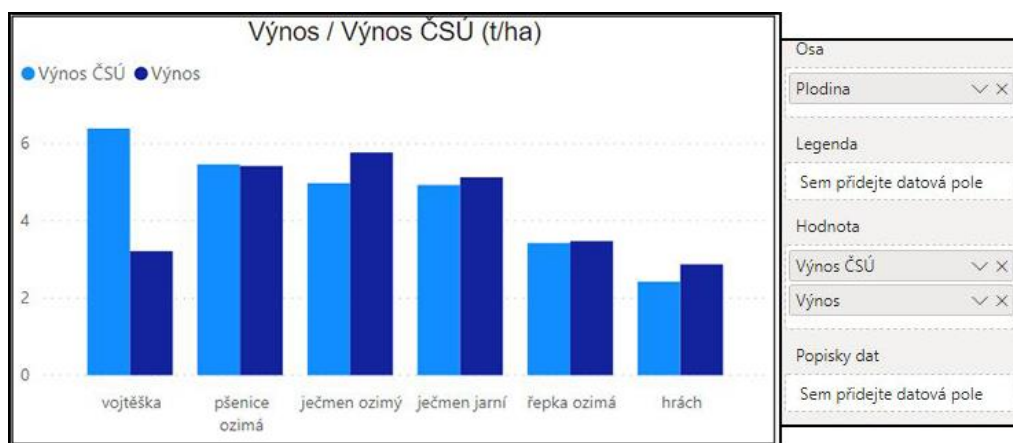
### 4.5.1 Vizualizační komponenty

Každá z vizualizačních komponent má v Power Bi Desktop své individuální nastavení. V následující části budou představeny vybrané komponenty se základním nastavením metrik. Filtr pro každou komponentu je rok: 2018.

#### 4.5.1.1 Skupinový sloupcový graf

Skupinový sloupcový graf se dá využít pro porovnání různých hodnot pro jednu nebo více kategorií. V tomto příkladu se srovnávají průměrné hodnoty sklizní pro jednotlivé plodiny.

**Obr. 20: Příklad skupinového sloupcového grafu v Power Bi Desktop**

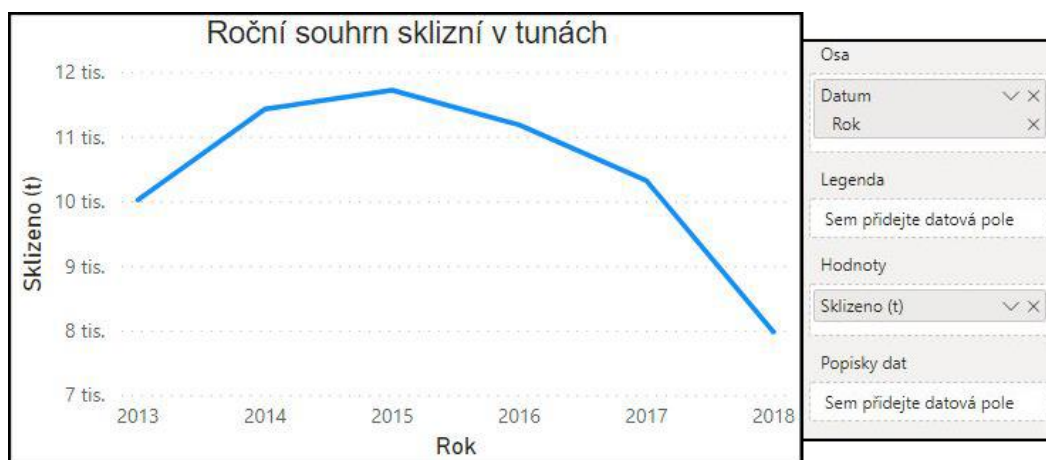


*Zdroj: vlastní zpracování*

#### 4.5.1.2 Spojnicový graf

Spojnicový graf lze využít pro sledování nějaké odchylky pro nějaké časové období. V příkladu je znázorněn vývoj celkových sklizní pro časovou osu. Filtr pro čas (rok) je pro tuto komponentu vypnut.

**Obr. 21: Příklad spojnícového grafu v Power Bi Desktop**

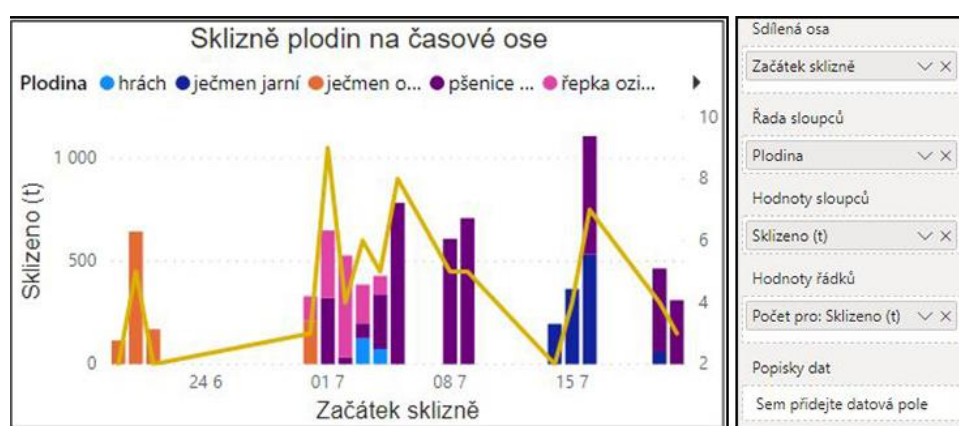


*Zdroj: vlastní zpracování*

### 4.5.1.3 Spojnicový a skládaný graf

Kombinace a spojnicového a skládaného grafu lze využít pro zobrazení více číselných hodnot faktů pro dimenze. Na ose x je znázorněna časová posloupnost po dnech, konkrétně datумы sklizní. Na ose Y je pro hodnoty sloupců nastavena celková sklizeň a pro spojnicový graf součet počtů sklizní, tedy kolik bylo daný den výjezdů. Sloupce skládaných grafů jsou rozděleny podle plodin. Z filtru grafu je vyloučeno datum 31.12.

**Obr. 22: Příklad spojnicového a skládaného grafu v Power BI Desktop**



*Zdroj: vlastní zpracování*

### 4.5.1.4 KPI

KPI popisují rozdíly pro vybranou hodnotu s hodnotou očekávaní. Vizualizace zobrazuje porovnání sklizní pro aktuální a minulý rok. Barevně indikuje splnění a nesplnění cíle a udává procentuální rozdíl mezi hodnotami.

**Obr. 23: Příklad KPI v Power BI Desktop**



*Zdroj: vlastní zpracování*

#### 4.5.1.5 Matice

Matice lze využít pro zobrazení dynamických hodnot v tabulkovém prostředí. V příkladu je pro řádky vytvořena hierarchie z katastrů a honů, na sloupcích je pozorovaná hodnota plodina a sledované hodnoty jsou počty sklizní.

Obr. 24: Příklad matice v Power BI Desktop

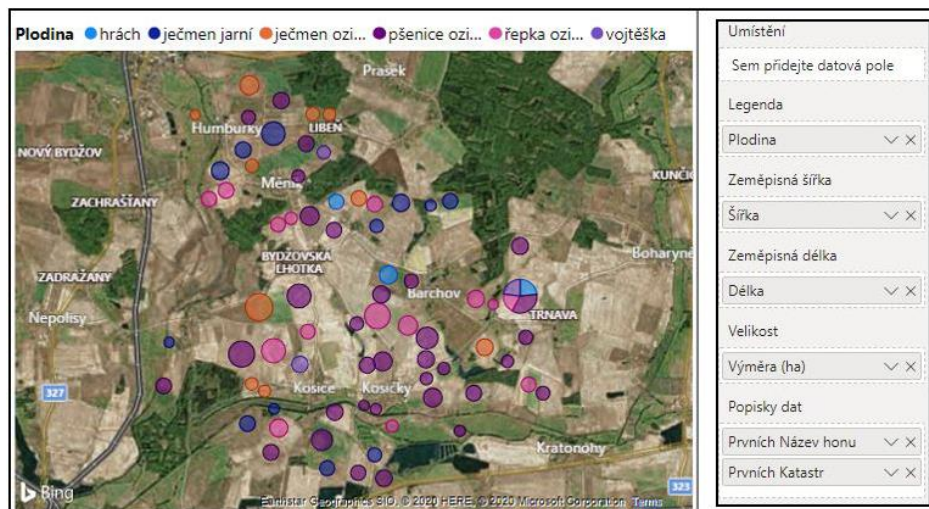
Katastr	hrách	ječmen jarní	ječmen ozimý	pšenice ozimá	řepka ozimá	vojtěška	Celkem
Trnava				166,80	59,32		226,12
Nový Bydžov			212,62				212,62
Mlékosrby		8,96	381,40	492,56			882,92
Měnik		118,80	211,16	224,26	160,72	44,48	759,42
Kosičky		89,10	113,67	1 364,48	368,38		1 935,63
Kosice		245,37	110,90	943,86	349,90	79,42	1 729,45
Humburky		391,90	1,70	241,26			634,86
Bydžovská				242,84	57,04		299,88
Za Váncem				100,44			100,44
Vrše					57,04		57,04
U Sušárny				142,40			142,40
Barchůvek	69,18	133,60	111,26		76,38		390,42
Barchov	73,60	176,30		79,62			329,52
Za Humny				79,62			79,62
Skok		93,60					93,60
Sklepy		59,70					59,70
Pluhovského		23,00					23,00
Kotchan	73,60						73,60
Babice	56,41			329,00	146,23	56,00	587,64
Celkem	199,19	1 164,03	1 142,71	4 084,68	1 217,97	179,90	7 988,48

Zdroj: vlastní zpracování

#### 4.5.1.6 Mapa

Mapa v Power BI Desktop využívá funkcí map Bing. Při znalosti zeměpisných souřadnic nebo názvu místa lze identifikovat pozici na mapách. V příkladu se využívá zeměpisných souřadnic honů, pro velikost bublin je využívána výměra honu a pro rozeznání, kde byla daná plodina pěstována je do legendy přidána plodina.

Obr. 25: Příklad mapy v Power BI Desktop



Zdroj: vlastní zpracování

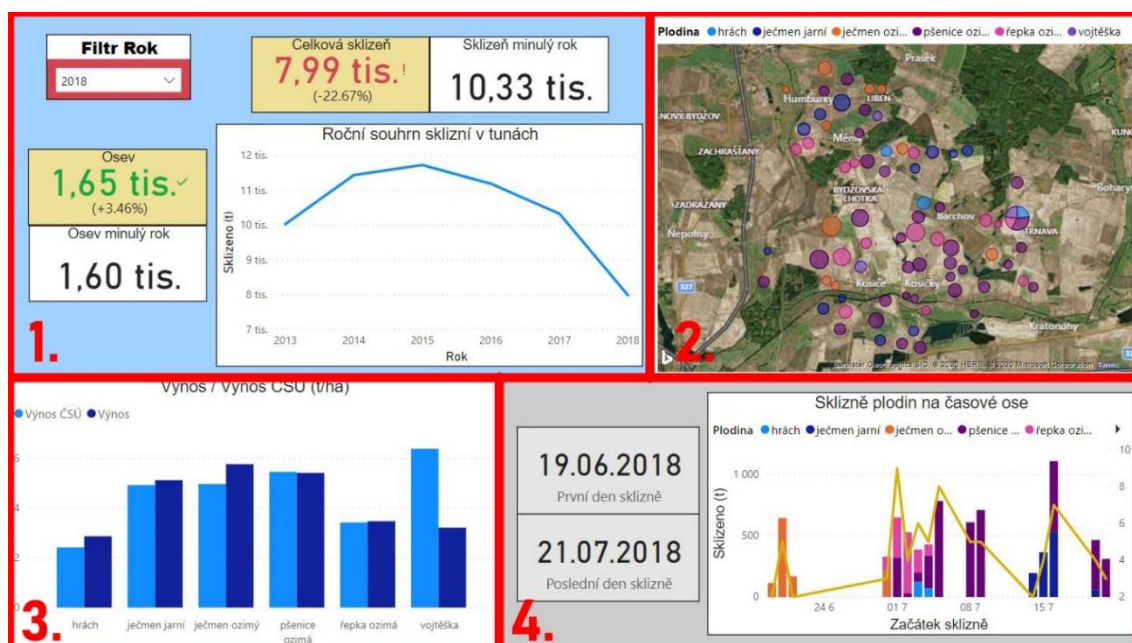
## 4.5.2 Dashboardy

V projektu byly vytvořeny 3 dashboardy. První dashboard zobrazuje celoroční přehled sklizní podniku, druhý je zaměřený na detail pro odrůdy plodin a třetí obsahuje detaily sklizní pro katastry a hony.

Celoroční přehled podniku (viz. obr. 26) je rozdělen na 4 hlavní části:

1. V této části je vidět porovnání celkových sklizní a osevů za aktuální a předešlý rok. Dále je zde spojnicový graf s výnosy za jednotlivé roky. Filtrovací tlačítka sloučí přepínání přehledů pro filtrovaný rok.
2. Na mapě jsou zobrazena všechna využitá pole z pohledu plodin pro aktuální rok.
3. Skupinový sloupcový graf zobrazuje porovnání výnosů podniku a celostátního průměru pro filtrovaný rok. V grafu je možné vybrat kliknutím plodinu a všechny číselné údaje v části 1 se aktuální plodnou vyfiltrují a dynamicky aktualizují.
4. Poslední část vizuálu je věnována časové ose a intenzitě výjezdů pro každý den. Do filtru grafu nebyl zahrnut datum 31.12.

**Obr. 26: Dashboard pro celkový přehled podniku podle plodin**



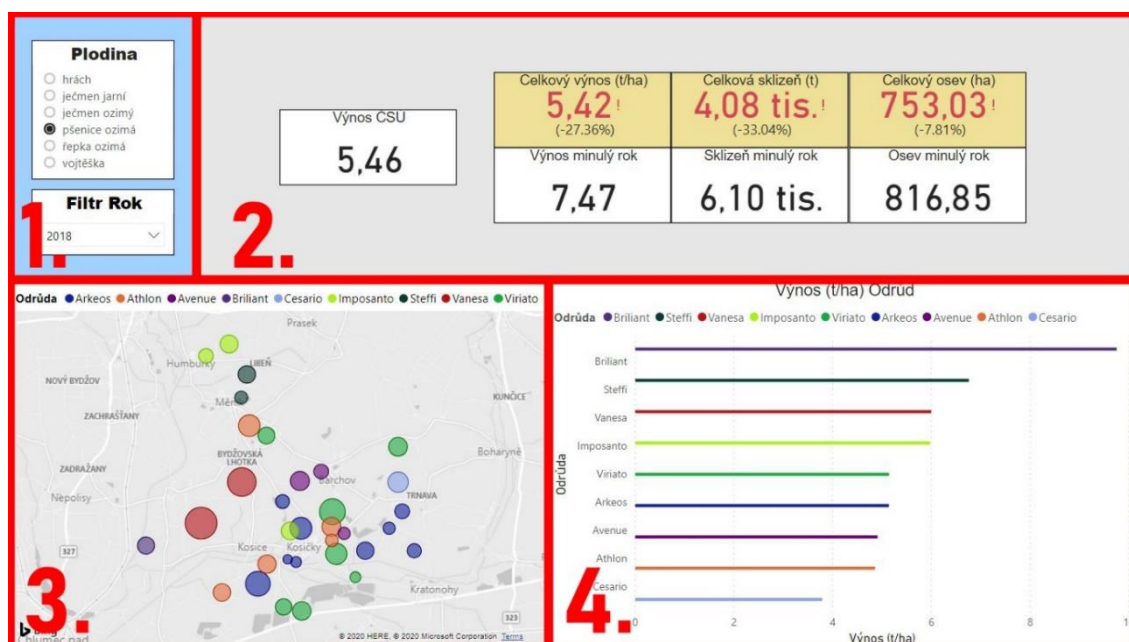
Zdroj: vlastní zpracování



Detail pro odrůdy plodin (viz. obr. 27) obsahuje následující části:

1. Filtr pro plodinu a pro rok.
2. Celkový výnos z ČSÚ pro danou plodinu; výnos podniku, celkovou sklizeň a osev podniku za filtrovaný a předešlý rok
3. Mapa, na které se odrůdy filtrované plodiny za daný rok nacházejí
4. Pruhový graf srovnávající výnosnost všech odrůd filtrované plodiny. Kliknutím na název odrůdy se vyfiltrují části 2 kromě dat z ČSÚ

**Obr. 27: Dashboard pro odrůdy plodin**

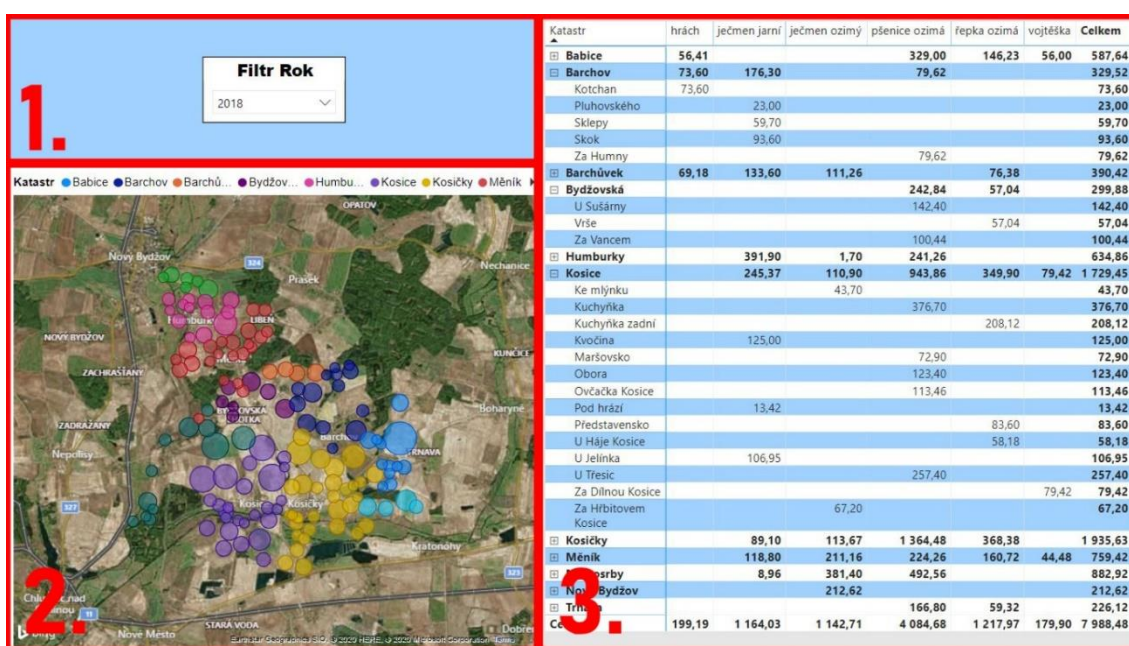


Zdroj: vlastní zpracování

Poslední vizualizací (viz. obr. 28) je zobrazení pro katastry a hony:

1. Filtr pro rok.
2. Mapa zobrazující všechny hony a katastry.
3. Matice zobrazující hodnoty sklizní pro katastry a hony v závislosti na plodinách. Pro interakci s mapou stačí kliknutím označit hon nebo katastr v matici.

Obr. 28: Dashboard pro katastry a hony



Zdroj: vlastní zpracování

## 4.6 Shrnutí

V teoretické části došlo představení pojmu BI. Dále byla rozvedena problematika datových skladů a základní principů tvorby datových modelů. Pro práci s daty byla uvedena metodika ETL, OLAP a dolování dat. Poté, co jsou data připravena v modelu nastává jejich vizualizace. Vizualizaci se věnovala samostatná kapitola, ve které byly zmíněny známé a často používané vizualizační komponenty a struktury, do kterých se uspořádávají: dashboardy a scorecardy. SSBI je jedním ze směrů BI a došlo k jemu specifikování a uvedení rozdílů s BI. Dále byla představena vývojová prostředí Power Pivot a Power BI od společnosti Microsoft, díky kterým lze projekty SSBI realizovat. Jejich motorem je jazyk DAX, který obsahuje mnoho užitečných funkcí a představena byla jedna z nejpoužívanějších.

Praktická část navazovala na teoretickou část, kde byl demonstrován kompletní postup tvorby BI řešení. Nejdříve proběhla analýza a tvorba návrhu pro strukturu dat. Zvoleno bylo schéma star. Dalším krokem byla ETL proces. Pro efektivnější práci s editací dat byla původní data z excelových sešitů importována na SQL Server a dále byla provedena jejich úprava do tabulek podle zpracované analýzy. Cílová destinace pro data bylo Power BI Desktop, kam byla naimportována. Po nastavení relací mezi dimenzemi a fakty byl datový model kompletní. Pro efektivní vizualizaci byly vytvořeny kalkulace pomocí measures. V posledním kroku byly ukázány a vysvětleny vizualizační komponenty využité v projektu a jejich celkové zasazení do dashboardů.

## 5 Závěr

V bakalářské práci byla znázorněna tvorba BI řešení pro zemědělský podnik. Veškeré postupy byly založené na teoretické části, sepsané z odborných zdrojů. Při tvorbě projektu byly použity kromě Microsoft Excelu volně a zdarma dostupné technologie. Do řešení byla použita pouze část dat, se kterými podnik pracuje, a která by šla do celkového procesu BI zahrnout. Dále by do procesu šly zahrnout například peněžní zisky, výdaje a záznamy dat s nimi spojenými. Datový trh se takto dá rozšiřovat o další části a tím vytvořit komplexní přehled podniku. Pokud se jedná pouze o řešení datového trhu ohledně osevní a sklizní, jako je v projektu (může být rozšířené třeba i o záznamy hnojení), jedná se o SSBI, pro které je řešení v Power BI Desktop naprosto dostačující. Otázka nastává, do jaké míry by byla režie tohoto řešení nákladná, pokud by se datový sklad stále rozšiřoval o nové druhy záznamů.

V projektu byl demonstrován proces ETL, ve kterém byla ukázána spolupráce software (Excel, SQL Server, Power BI) z rodiny společnosti Microsoft. Microsoft nabízí zajímavé možnosti pro tvorbu BI řešení a určitě by se na světovém trhu neměly přehlížet. Vizualizace byla provedena v Power BI Desktop, které nabízí intuitivní prostředí pro tvorbu vizualizačních komponent a sestav. Díky možnostem jazyku DAX, lze s daty efektivně pracovat a provádět výpočty podle aktuálních potřeb.

## 6 Seznam zdrojů

### 6.1 Tištěné zdroje

NOVOTNÝ, Ota, Jan POUR a David SLÁNSKÝ. Business intelligence: jak využít bohatství ve vašich datech. Praha: Grada, 2005. Management v informační společnosti. ISBN 80-247-1094-3.

HOWSON, Cindi. Successful business intelligence: unlock the value of BI & big data. Second edition. New York: McGraw-Hill Education, [2014]. ISBN 978-0-07-180918-4.

LABERGE, Robert. Datové sklady: agilní metody a business intelligence. Brno: Computer Press, 2012. ISBN 978-80-251-3729-1.

LACKO, Luboslav. Business Intelligence v SQL Serveru 2008: reportovací, analytické a další datové služby. Brno: Computer Press, 2009. ISBN 978-80-251-2887-9.

TURBAN, Efraim. Business intelligence: a managerial approach. 2nd ed. Boston: Prentice Hall, 2010. ISBN 013610066x.

COLLIE, Robert Clinton. Dax formulas for power pivot: the excel pro's guide to mastering dax. 2nd edition. Uniontown, OH: Holy Macro! Books, 2015. ISBN 978-1-61547-039-6.

CLARK, Dan. Beginning Power BI: a practical guide to self-service data analytics with Excel 2016 and Power BI Desktop. Second edition. Camp Hill, Pennsylvania: Apress, [2017]. ISBN 978-1-4842-2576-9.

POUR, Jan, Miloš MARYŠKA, Iva STANOVSKÁ a Zuzana ŠEDIVÁ. Self service business intelligence: jak si vytvořit vlastní analytické, plánovací a reportingové aplikace. Praha: Grada Publishing, 2018. Management v informační společnosti. ISBN 978-80-271-0616-5.

FEW, Stephen. Information dashboard design: the effective visual communication of data. Cambridge [MA]: O'Reilly, c2006. ISBN 0-596-10016-7.

AIGNER, Wolfgang. Visualization of time-oriented data. London: Springer, c2011. Human-computer interaction series. ISBN 978-0-85729-079-3.

HARRIS, Robert L. Information graphics: a comprehensive illustrated reference. New York: Oxford University Press, 1999. ISBN 978-0195135329.

EVERGREEN, Stephanie D. H. Effective data visualization: the right chart for the right data. Los Angeles: SAGE, [2017]. ISBN 9781506303055.

SEAMARK, Philip. Beginning DAX with Power BI. UPPER HUTT, New Zealand: Apress, 2018. ISBN 978-1-4842-3476-1.

FERRARI, Alberto a Marco RUSSO. The Definitive Guide to DAX: Business intelligence with Microsoft Excel, SQL Server Analysis Services, and Power BI. Redmond, Washington 98052-6399: Microsoft Press, 2015. ISBN 978-0-7356-9835-2.

## 6.2 Online zdroje

What is Business Intelligence? BI Definition. OLAP.com [online]. PARIS Technologies [cit. 2018-11-13]. Dostupné z: <http://olap.com/learn-bi-olap/olap-bi-definitions/business-intelligence/>

WHAT IS THE DEFINITION OF OLAP? OLAP.com [online]. PARIS Technologies [cit. 2020-03-27]. Dostupné z: <https://olap.com/olap-definition/>

POUR, J. a A. VINZE. Business Intelligence. ČSSI: Česká společnost pro systémovou integraci [online]. Praha: Vysoká škola ekonomická, 2005 [cit. 2018-11-13]. Dostupné z: <http://www.cssi.cz/cssi/business-intelligence>

IMHOFF, Claudia a Colin WHITE. Self-Service Business Intelligence. TDWI research: TDWI best practices Report [online]. TDWI, 2011, 2011(THIRD Quarter 2011), 5 [cit. 2018-11-13]. Dostupné z: [http://docs.media.bitpipe.com/io\\_10x/io\\_106625/item\\_583281/TDWI\\_Best\\_Practices\\_Report\\_Self-Service\\_BI\\_Q311%5B1%5D.pdf](http://docs.media.bitpipe.com/io_10x/io_106625/item_583281/TDWI_Best_Practices_Report_Self-Service_BI_Q311%5B1%5D.pdf)

POUR, Jan. Self-service business intelligence. Systémová integrace 1-2/2014 [online]. 2014, 2014, 21(1-2), 136-139 [cit. 2018-11-13]. ISSN 1804-2716. Dostupné z: <http://www.cssi.cz/cssi/self-service-business-intelligence>

FAZAL, Hasaan. Variance Analysis in Excel – Making better Budget Vs Actual charts [online]. 2014 [cit. 2020-01-20]. Dostupné z:

<https://pakaccountants.com/variance-analysis-excel-budget-vs-actual-charts/>

LIBERTY, Dana. Scorecard vs Dashboard – What Each Adds to Business Intelligence [online]. 4. červen 2018 [cit. 2020-01-20]. Dostupné z:

<https://www.sisense.com/blog/scorecard-vs-dashboard-adds-business-intelligence/>

## Podklad pro zadání BAKALÁŘSKÉ práce studenta

Jméno a příjmení: **Tomáš Karásek**  
Osobní číslo: **I1600553**  
Adresa: **Barchůvek 59, Měník – Barchůvek, 50401 Nový Bydžov, Česká republika**  
Téma práce: **Self-service Business Intelligence**  
Téma práce anglicky: **Self-service Business Intelligence**  
Vedoucí práce: **Ing. Barbora Tesařová, Ph.D.**  
**Katedra informatiky a kvantitativních metod**

### Zásady pro vypracování:

Cílem této práce bude vymezení pojmu Self-service Business Intelligence a její komponent. V praktických příkladech zabývajících se problematikou pěstování a sklizně plodin bude demonstrován postup pro vytvoření multidimenzionálního prostředí v samoobslužném BI nástroji PowerBI. Dále bude v nástroji PowerBI představena tvorba a vizualizace sestav za použití jazyku DAX sloužícího pro tvorbu vzorců a výpočtů datových hodnot.

- 1.Úvod
- 2.Teoretická část
  - 2.1 Business Intelligence
  - 2.2 Datové sklady
  - 2.3 ETL
  - 2.4 OLTP x OLAP
  - 2.5 Self-service BI
  - 2.5 PowerBI
- 3.Praktická část
  - 3.1 Analýza a implementace BI na problematiku pěstování a sklizně plodin v zemědělském podniku.
4. Shrnutí výsledků
5. Závěr

### Seznam doporučené literatury:

- NOVOTNÝ, Ota, Jan POUR a David SLÁNSKÝ. Business intelligence: jak využít bohatství ve vašich datech. Praha: Grada, 2005. Management v informační společnosti. ISBN 80-247-1094-3.
- HOWSON, Cindi. Successful business intelligence: unlock the value of BI & big data. Second edition. New York: McGraw-Hill Education, [2014]. ISBN 978-0-07-180918-4.
- LABERGE, Robert. Datové sklady: agilní metody a business intelligence. Brno: Computer Press, 2012. ISBN 978-80-251-3729-1.
- LACKO, Luboslav. Business Intelligence v SQL Serveru 2008: reportovací, analytické a další datové služby. Brno: Computer Press, 2009. ISBN 978-80-251-2887-9.
- TURBAN, Efraim. Business intelligence: a managerial approach. 2nd ed. Boston: Prentice Hall, 2010. ISBN 013610066x.
- COLLIE, Robert Clinton. Dax formulas for power pivot: the excel pro's guide to mastering dax. 2nd edition. Uniontown, OH: Holy Macro! Books, 2015. ISBN 978-1-61547-039-6.



CLARK, Dan. Beginning Power BI: a practical guide to self-service data analytics with Excel 2016 and Power BI Desktop. Second edition. Camp Hill, Pennsylvania: Apress, [2017]. ISBN 978-1-4842-2576-9.

Podpis studenta:

Datum:

Podpis vedoucího práce:

Datum:

© IS/STAG, Portál – Podklad kvalifikační práce, karasto1, 30. dubna 2020 17:34