

**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

**FAKULTA STROJNÍ**

**Ústav řízení a ekonomiky podniku**



**DIPLOMOVÁ PRÁCE**

Vytvoření relační databáze pro sběr a uchování dat ze zkoušek  
vzorků sypkých materiálů pro firmu Sklenář s.r.o.

Creation of a relational database for the collection and storage of  
data from tests of samples of bulk materials for the company  
Sklenář s.r.o.

Praha 2021

Bc. Denis Lom

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: Lom Jméno: Denis Osobní číslo: 459578  
Fakulta/ústav: Fakulta strojní  
Zadávací katedra/ústav: Ústav řízení a ekonomiky podniku  
Studijní program: Strojní inženýrství  
Studijní obor: Řízení a ekonomika podniku

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Vytvoření relační databáze pro sběr a uchování dat ze zkoušek vzorků sypkých materiálů pro firmu Sklenář s.r.o

Název diplomové práce anglicky:

Creation of a relational database for the collection and storage of data from tests of samples of bulk materials for the company Sklenář s.r.o

Pokyny pro vypracování:

1. Úvod – zdůvodnění tématu diplomové práce
2. Teorie SQL databází s důrazem na relační databáze
3. Definování potřeb a funkcí DBS pro podnik
4. Návrh datového modelu a transakcí,
5. Zhodnocení přínosů návrhu

Seznam doporučené literatury:

Riordan, R.M.: Vytváříme relační databázové aplikace. Praha : Computer Press, 2000. 280 s. ISBN 80-7226-360-9  
Pokorný, J.: Databázové systémy a jejich použití v informačních systémech. Praha : Academia, 1992. 313 s. ISBN 80-200-0177-8  
Silberschatz, A., Korth H.F, Sudarshan, S.: Database System Concepts. Fourth Edition. Mc-Graw-Hill. 2002, 1064 p.  
Hoffer, J.A., Prescott, Ma B., Mc Fadden, F.R.: Modern Database Management. 6 ed. Upper Saddle River : Prentice Hall, 2002. 638 s. ISBN 0-13-033969-5  
Bíla, Šmíd, Král, Hlaváč: Informační technologie. Databázové a znalostní systémy. ČVUT, Praha, 2009.

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Vladimír Brdek, Ph.D., ústav řízení a ekonomiky podniku FS

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: 30.04.2021 Termín odevzdání diplomové práce: 23.07.2021

Platnost zadání diplomové práce: 28.02.2022

Ing. Vladimír Brdek, Ph.D.  
podpis vedoucí(ho) práce

Ing. Miroslav Žilka, Ph.D.  
podpis vedoucí(ho) ústavu/katedry

prof. Ing. Michael Valášek, DrSc.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta

# Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, atd.) uvedené v příloženém seznamu.

V Praze dne .....

.....

Denis Lom

## Poděkování

Chtěl bych poděkovat panu Ing. Vladimíru Brdkovi, Ph.D. za vedení mé diplomové práce a za poskytnutí cenných rad.

Dále bych chtěl poděkovat panu Ing. Petru Sklenářovi a panu Ing. Ludku Sklenářovi za možnost spolupráce a v neposlední řadě děkuji panu Ing. Miroslavu Keprtovi za všechny poskytnuté informace a ochotu.

Část mého poděkování bych také rád věnoval mé alma mater a všem lidem, kteří umožňují studentům vyjíždět na studijní zahraniční pobyty. Moc si vážím, že jsem měl možnost během svého studia studovat také v zahraničí.

Děkuji svým rodičům za možnost studovat a získat vědomosti a zkušenosti, za které jsem vděčný. Své sestře a přítelkyni děkuji za poskytnutou podporu a inspiraci. Nakonec děkuji všem českým i zahraničním přátelům, které jsem poznal během vysokoškolského studia za možnost sdílet zkušenosti a zážitky.

# Anotační list

**Jméno autora:** Denis Lom

**Fakulta:** Fakulta strojní

**Zadávací ústav:** Ústav řízení a ekonomiky podniku

**Studijní program:** Strojní inženýrství

**Studijní obor:** Řízení a ekonomika podniku

**Rok:** 2021

**Název diplomové práce:** Vytvoření relační databáze pro sběr a uchování dat ze zkoušek vzorků sypkých materiálů pro firmu Sklenář s.r.o

**Title:** Creation of a relational database for the collection and storage of data from tests of samples of bulk materials for the company Sklenář s.r.o

**Vedoucí práce:** Ing. Vladimír Brdek, Ph.D.

<b>Bibliografické údaje:</b>	počet stran	66
	počet obrázků	34

**Klíčová slova:** relační databáze, databázový systém, MS Access, entitně-relační diagram, normalizace, SQL

**Key words:** relational database, database system, MS Access, entity-relational diagram, normalization, SQL

**Anotace:** Diplomová práce se zabývá vytvořením relační databáze pro sběr a uchování dat ze zkoušek vzorků sypkých materiálů. V rámci této diplomové práce je popsána problematika teorie databázových systémů. V praktické části je vytvořena relační databáze, která slouží pro sběr a uchování dat ze zkoušek vzorků sypkých materiálů.

**Abstract:** The thesis deals with the creation of a relational database for collecting and storing data from tests of samples of bulk material. Within this thesis the issues of database systems theory are described. In the practical part, a relational database is created to collect and store data from bulk material sample tests.

# Obsah

1	Úvod .....	9
2	Teorie databází.....	9
2.1	Historie databází .....	9
2.2	Databáze .....	11
2.3	Databázový systém .....	11
2.4	Databázové modely z hlediska uchování dat .....	12
2.5	Relační databáze .....	14
2.6	Konceptuální datový model .....	15
2.7	Elementy konceptuálního modelu .....	15
2.7.1	Entita .....	15
2.7.2	Atribut .....	16
2.7.3	Doména .....	16
2.7.4	Vztahy (asociace).....	16
2.7.5	Identifikátor.....	17
2.7.6	Spojovací tabulka .....	18
2.7.7	E-R diagram .....	18
2.8	Normalizace .....	18
2.8.1	První normální forma .....	19
2.8.2	Druhá normální forma.....	19
2.8.3	Třetí normální forma .....	20
2.8.4	Konceptuální model v kanonickém tvaru.....	20
2.8.5	Požadavky na informace uživatelů.....	20
2.8.6	Relační model.....	21
2.8.7	Pravidla transformace E-R diagramu na relační model.....	21
2.9	SQL .....	24
2.9.1	Data Definition Language (DDL) .....	25

2.9.2	Data Manipulation Language (DDL) .....	25
2.9.3	Data Query Language (DQL).....	25
2.9.4	Typy dat v SQL.....	26
2.10	MS Access.....	26
2.10.1	Desktopová a webová aplikace Access .....	27
2.10.2	Tabulky .....	27
2.10.3	Formuláře.....	28
2.10.4	Sestavy .....	28
2.10.5	Dotazy .....	28
2.10.6	Makra .....	28
2.10.7	Moduly .....	29
2.10.8	Datové typy v MS Accessu (verze 2013 a novější) .....	29
3	Definování potřeb a funkcí databázového systému pro podnik.....	30
4	Návrh datového modelu a transakcí.....	32
4.1	Analýza procesu .....	32
4.1.1	Procesní mapa.....	32
4.1.2	Dosavadní dokumentace.....	33
4.2	Návrh řešení.....	34
4.2.1	E-R diagram .....	34
4.3	Implementace řešení .....	35
4.3.1	E-R Diagram v MS Access .....	36
4.3.2	Import dat z xls souborů do tabulek .....	38
4.3.3	Vytvoření dotazu pro výsledný protokol – Dotaz1.....	40
4.3.4	Vytvoření dotazu pro vytištění štítku – Dotaz2.....	41
4.3.5	Vytvoření dotazu pro zobrazení všech údajů o zkoušce – Dotaz3 .....	43
4.3.6	Vytvoření dotazu pro zobrazení všech údajů o vzorku – Dotaz4 .....	43
4.3.7	Vytvoření dalších výběrových dotazů .....	44
4.3.8	Vytvoření přidávacích dotazů.....	44

4.3.9	Vytvoření odstraňovacích dotazů .....	48
4.3.10	Vytvoření formuláře – Úvodní stránka.....	49
4.3.11	Vytvoření formuláře – Evidence vzorků .....	49
4.3.12	Vytvoření formuláře - Provádění zkoušky.....	50
4.3.13	Vytvoření formuláře – Všechny údaje o vzorku .....	52
4.3.14	Vytvoření formuláře – Všechny údaje o zkoušce .....	53
4.3.15	Sestava výsledného protokolu ze zkoušky .....	54
4.3.16	Vytvoření sestavy za účelem tisku štítků – sestava Štítky pro Dotaz2.....	55
5	Zhodnocení přínosů návrhu .....	56
6	Závěr.....	57
7	Zdroje .....	58
7.1	Knižní zdroje .....	58
7.2	Elektronické zdroje.....	58
7.3	Jiné zdroje .....	60
8	Seznam obrázků .....	61
A	SQL kód pro vytvoření tabulek z E-R diagramu.....	63
B	Obsah příloženého CD.....	66



# 1 Úvod

Cílem této diplomové práce je navrhnout relační databázi pro sběr a úschovu dat ze zkoušek sypaných materiálů pro firmu Sklenář s.r.o.

Tato diplomová práce je rozdělena na teoretickou a praktickou část. Teoretická část je rešerší na téma relační databáze a pojednává o teorii, kterou je třeba znát pro vytvoření relační databáze. Praktická část práce je věnována analýze procesu zkoušek sypaných materiálů v podniku a dosavadní dokumentaci protokolů z těchto zkoušek. Na závěr je vytvořena relační databáze v programu MS Access.

V dnešní době jsou alespoň základní znalosti databázových systémů velice užitečné. Databáze se používají prakticky všude. Jsou součástí webových aplikací nebo různých podnikových aplikací. Databáze jsou součástí jádra jakékoliv aplikace na řízení podniků, které mohou sloužit např. jako podpora podnikového rozhodování.

Databáze umožňují mít přehled o různých procesech nebo skutečnostech. Dobře navrhnutá databáze může ulehčit a zefektivnit práci v podnicích. K dobrému návrhu databáze nestačí jen dobré znalosti informačních technologií. Při návrhu databáze je důležité dobře poznat fungování podniku a obeznámit se s procesy uvnitř podniku.

Během magisterského studia jsem studoval dva kurzy, které se týkaly databází a informačních systémů. Tyto kurzy ve mně probudily zájem o informační technologie a jejich aplikaci v podnicích. Dle mého názoru dobrá manipulace s daty a mapování všech procesů v podniku může vést k lepšímu fungování podniku.

## 2 Teorie databází

### 2.1 Historie databází

První databázové systémy se objevují koncem šedesátých let dvacátého století. Tyto databáze vycházely ze dvou přístupů:

1. vzájemně propojené soubory dat (síťové a hierarchické databáze),
2. fyzicky nezávislé soubory dat (relační databáze).<sup>1</sup>

V roce 1965 se konala konference CODASYL (Conference on Data System Languages). Během

---

<sup>1</sup> POKORNÝ, Jaroslav a Michal VALENTA. Databázové systémy. 2. přepracované vydání. Praha: Česká technika - nakladatelství ČVUT, 2020. ISBN 978-80-01-06696-6.

této konferenci byl vytvořen výbor DBTG (Database Task Group). Úkolem tohoto výboru bylo standardizačním postupem vytvořit koncepci databázového systému. Tímto způsobem vznikl např. databázový systém IDMS (Integrated Database Management System). V této době začala firma General Electric prodávat produkt IDS (Integrated Data Store), který byl vyvinut týmem vedeným Charlesem Bachmanem.

V roce 1971 byla výborem DBTG vydána zpráva „The DBTG April 1971“. Tato zpráva obsahovala pojmy jako schéma databáze, jazyk pro definici schématu, subschéma nebo celková architektura síťového databázového systému. Původní síťový model CODASYLu byl následně aktualizován čtyřikrát v letech 1973, 1987, 1981 a 1984.<sup>2</sup>

Historie hierarchického modelu je spojena s databázovým systémem IMS (Information Management System). Hierarchický model (na rozdíl od síťového) nemá standard. První verze databázového systému IMS byla vyvinuta v šedesátých letech firmami IBM a North American Aviation pro program Apollo. Dnes se IMS stále používá na sálových počítačích. IBM stále nabízí komerční využití. Momentálně je dostupná k využití např. verze IBM IMS 15.<sup>3 4</sup>

V roce 1970 vydává Edgar Frank Codd článek v časopise Communications of ACM. Jeho teoretický přístup vrátil data do fyzicky izolovaných souborů. Data byla pro uživatele vizualizována jako tabulky. Šlo o počátek éry relačních databází.

Přijetí relačních databází v praxi z optiky IT světa probíhalo poměrně pomalu. Trvalo téměř 10 let než se relační databáze vyvinuly natolik, že výkon jejich databázových systémů byl srovnatelný se síťovými a hierarchickými prostředky v reálném prostředí.

V roce 1989 vychází Manifest skupiny Altair, ve kterém jsou identifikovány objektové databáze. Počet nasazení těchto objektových databází v praxi nerostl tak rychle, jak se očekávalo. Na konci devadesátých let se v praxi začaly využívat tzv. objektově relační databáze, které kombinují vlastnosti relačních databází s vlastnostmi objektových databází. V objektově relačních databázích jsou základním konstruktem relace.

Po příchodu značkovacího jazyka XML na konci devadesátých let se objevila potřeba ukládat XML data databázovým způsobem. Pro ukládání XML dat se využívá relační databáze nebo

---

<sup>2</sup> POKORNÝ, Jaroslav a Michal VALENTA. Databázové systémy. 2. přepracované vydání. Praha: Česká technika - nakladatelství ČVUT, 2020. ISBN 978-80-01-06696-6.

<sup>3</sup> IBM IMS documentation. IBM [online]. Armonk, New York: IBM, 2021 [cit. 2021-7-15]. Dostupné z: <https://www.ibm.com/docs/en/ims>

<sup>4</sup> POKORNÝ, Jaroslav a Michal VALENTA. Databázové systémy. 2. přepracované vydání. Praha: Česká technika - nakladatelství ČVUT, 2020. ISBN 978-80-01-06696-6.

samostatné nástroje se speciálními technikami ukládání a indexace.

Mezi nové směry v oblasti databází patří NoSQL databáze a grafové databáze.<sup>5</sup>

## 2.2 Databáze

Dle společnosti Oracle databáze je organizovaný soubor strukturovaných dat, která jsou obvykle ukládána v počítačovém systému.<sup>6</sup>

Databáze tedy může být jednoduchý seznam s položkami co nakoupit v supermarketu až po složitou množinu nástrojů, bez kterých by se neobešel žádný e-shop.

## 2.3 Databázový systém

Databázový systém je program, který umožňuje přístup k uloženým datům a manipulaci s nimi.<sup>7</sup>

Databázový systém se skládá ze čtyř komponent: uživatel, databázová aplikace, systém řízení databáze (DBMS – database management system) a vlastní databáze.

**Databáze** je souhrn spolu souvisejících tabulek a jiných struktur.

**Systém řízení databáze** je počítačový program, pomocí kterého můžeme vytvářet, zpracovávat a spravovat databáze. Tento počítačový program přijímá požadavky v jazyce SQL a umí je převést na aktivity databáze.

Mezi systémy řízení databáze patří např.: Microsoft Access, Microsoft SQL Server, MySQL od Oracle Corporation, DB2 od IBM.

**Databázová aplikace** je tvořena jedním nebo více počítačovými programy. Tyto programy slouží jako prostředník mezi uživatelem a systémem řízení databáze. Aplikační program umí číst nebo upravovat databázová data za pomoci odeslání příkazu SQL systému řízení databáze.

V praxi si firmy systémy řízení databází většinou netvoří samy. Databázové aplikace si často vytvoří interně, popř. je získají od dodavatelů softwaru.<sup>8</sup>

---

<sup>5</sup> POKORNÝ, Jaroslav a Michal VALENTA. Databázové systémy. 2. přepracované vydání. Praha: Česká technika - nakladatelství ČVUT, 2020. ISBN 978-80-01-06696-6.

<sup>6</sup> Co je to databáze. Oracle Česká republika [online]. Praha: Oracle Czech, 2021 [cit. 2021-7-15].

Dostupné z: <https://www.oracle.com/cz/database/what-is-database/>

<sup>7</sup> RIORDAN, Rebecca M. *Vytváříme relační databázové aplikace*. Praha: Computer Press, 2000. Databáze. ISBN 80-7226-360-9.

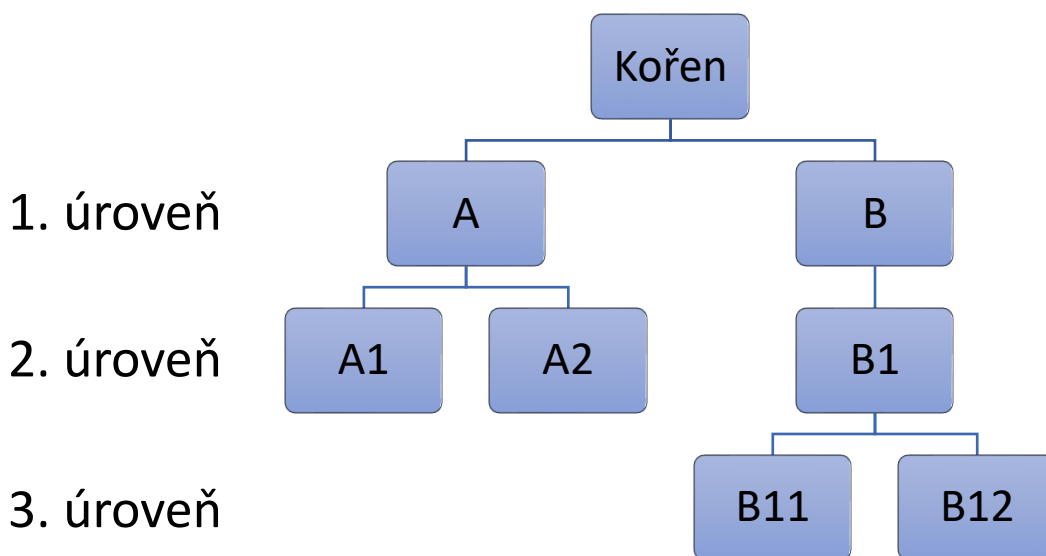
<sup>8</sup> KROENKE, David a David J. AUER. Databáze. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.

## 2.4 Databázové modely z hlediska uchování dat

Databáze můžeme rozdělit do základních typů, podle toho, jak data ukládáme a jak s nimi pracujeme:

- hierarchická databáze,
- síťová databáze,
- relační databáze,
- objektová databáze,
- objektivě relační databáze.<sup>9</sup>

**Hierarchické databázové systémy** jsou založeny na principu stromu. V těchto databázových systémech jsou data organizována na základě stromové struktury. Stromová struktura vychází z kořene a jednotlivé datové struktury jsou umístěny na různých úrovních. Tyto datové struktury se nazývají uzly. Pokud z uzlu již nevychází žádná další větev tento uzel se nazývá list.<sup>10</sup>



Obrázek 1: Model hierarchického databázového systému<sup>11</sup>

V případě, že uživatel chce najít nějakou informaci v položce B11, nemusí systém řízení báze

---

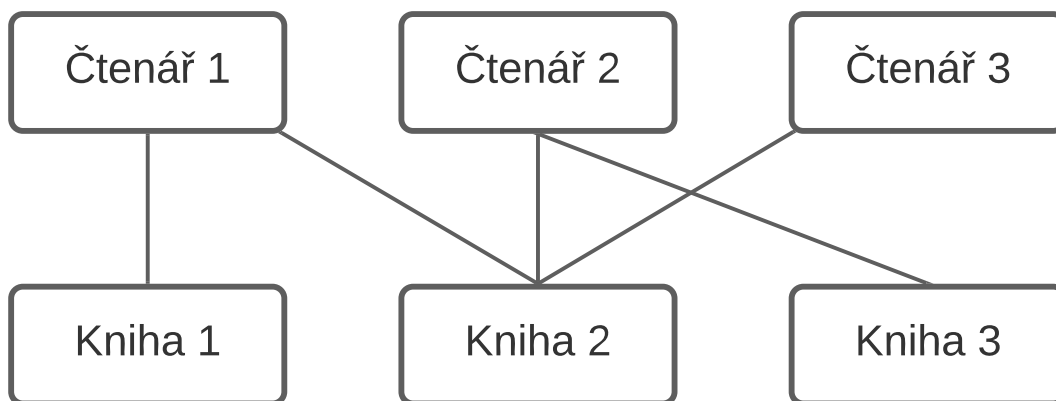
<sup>9</sup> POKORNÝ, Jaroslav a Michal VALENTA. Databázové systémy. 2. přepracované vydání. Praha: Česká technika - nakladatelství ČVUT, 2020. ISBN 978-80-01-06696-6.

<sup>10</sup> BÍLA, Jiří, František KRÁL a Vladimír HLAVÁČ. Informační technologie: databázové a znalostní systémy. Vyd. 2. přeprac. Praha: Vydavatelství ČVUT, 2003. ISBN 80-01-02790-2.

<sup>11</sup> Vlastní tvorba

dat prohledávat celý soubor. Stačí rozložit požadavek na jednotlivé složky a jít po větvi B.

**Síťové databázové** systémy jsou založeny na lineárních nebo cyklických ukazatelích. Tyto ukazatele vyjadřují vztahy mezi položkami. Hlavní výhodou síťových modelů je jejich flexibilita. Nevýhodou je obtížnost mapování vztahů mezi různými množinami.<sup>12</sup>



Obrázek 2: Model síťového databázového systému<sup>13</sup>

Z obrázku 2 je např. patrné, že čtenář 1 přečetl knihu 1 a knihu 2.

**Objektově orientované databázové systémy** jsou databázové produkty, které slouží k ukládání objektových dat. Na začátku devadesátých let několik dodavatelů vyvinulo tyto databázové produkty pro komerční použití. V té době již spousta podniků měla uložená data ve formátu relačních databází a byla zde malá ochota data převést do formátu objektově orientovaného databázového systému.<sup>14</sup>

Pomocí objektově orientovaného programování probíhá v současnosti tvorba většiny nových programů. Mezi objektově orientované programovací jazyky patří např. Java, C++, C#, Visual Basic.NET nebo Python. **Objekty** jsou datové struktury. Tyto datové struktury mají metody (počítačové programy provádějící nějaký úkol) a vlastnosti (datové položky popisující daný objekt).

Uložení hodnot vlastností objektu se nazývá object persistence (trvalost objektu). K zajištění trvalosti objektu se může použít databázová technologie. K zajištění trvalosti objektu mohou sloužit i relační databáze, ale to může často vyžadovat značné úsilí programátora. Někteří

---

<sup>12</sup> BÍLA, Jiří, František KRÁL a Vladimír HLAVÁČ. Informační technologie: databázové a znalostní systémy. Vyd. 2. přeprac. Praha: Vydavatelství ČVUT, 2003. ISBN 80-01-02790-2.

<sup>13</sup> Vlastní tvorba

<sup>14</sup> KROENKE, David a David J. AUER. Databáze. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.

dodavatelé (např. Oracle) doplnili do svých relačních databázových produktů vlastnosti a funkce a tím vznikli **objektově-relační databáze**. Tyto nové vlastnosti a funkce zajišťují trvalost objektů a usnadňují ukládat jednodušeji data než čistě relační databáze. Objektově-relační databáze umí zpracovávat i relační data.<sup>15</sup>

## 2.5 Relační databáze

Relační databáze je dána množinou relací. Relace je množinou prvků. Každý prvek relace se tváří jako řádek tabulky (záznam). Databázová relace může být reprezentována tabulkou.<sup>16</sup>

Základními charakteristikami relačních databází jsou:

- Všechna data jsou reprezentovaná v relacích.
- V každé konkrétní pozici řádku a sloupce dané relace se vyskytuje jedna hodnota. Jinými slovy hodnoty v databázi jsou skalární.
- Relační databáze využívají mechanismu „uzávěr“. To znamená, že operace v databázi se provádějí vždy nad celou relací a výsledkem operace je další relace.

Řádek je vektor hodnot (souřadnic). Počet řádků určuje kardinalitu.

Sloupec se nazývá atribut. Počet atributů v relaci se nazývá stupeň.

Relace se nenazývá tabulkou, protože relace je v podstatě množina. Prvky množiny jsou jednoznačně identifikovatelné, proto se v relaci nemohou vyskytovat opakované záznamy (řádky). Tabulka je tedy relací, pouze jestli se v ní žádný řádek neopakuje.<sup>17</sup>

Tabulku můžeme považovat za relaci, pokud splňuje následující podmínky.

1. Řádky obsahují data o entitě.
2. Sloupce obsahují data o attributech entity.
3. Buňky v tabulce uchovávají jedinou hodnotu.
4. Všechny položky ve sloupci jsou stejného druhu.
5. Na pořadí sloupců nezáleží.
6. Na pořadí řádků nezáleží.

---

<sup>15</sup> KROENKE, David a David J. AUER. Databáze. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.

<sup>16</sup> POKORNÝ, Jaroslav a Michal VALENTA. Databázové systémy. 2. přepracované vydání. Praha: Česká technika - nakladatelství ČVUT, 2020. ISBN 978-80-01-06696-6.

<sup>17</sup> RIORDAN, Rebecca M. *Vytváříme relační databázové aplikace*. Praha: Computer Press, 2000. Databáze. ISBN 80-7226-360-9.

7. Žádné dva řádky nesmí obsahovat identické sady datových hodnot.<sup>18</sup>

## 2.6 Konceptuální datový model

Datový model je myšlenkový popis problému. Skládá se z entit, atributů, domén (obory hodnot) a vztahů.

Nejvíce používaným konceptuálním datovým modelem je E-R diagram vyvinutý Peterem Chenem v roce 1976.

Jakmile je konceptuální datový model vytvořen, je jednoduché jej implementovat na jakýkoliv datový model, který je vhodný pro použitou technologii. Při vytvoření se musí jasně definovat elementy, vztahy a organizace daného systému, aby bylo možné pochopit fungování systémů na základě abstraktních formulací.<sup>19</sup>

## 2.7 Elementy konceptuálního modelu

### 2.7.1 Entita

**Entita** je osoba, místo, koncept nebo třída o které chceme uchovávat informace.

- Silná entita existuje sama o sobě. Má identifikátor tvořený vlastními atributy.
- Slabá entita existuje v závislosti na jiné entitě.
- Činitel je entita, jejíž existence nezávisí na čase.
- Událost je entita, jejíž existence závisí na čase.<sup>20</sup>

Entity se mohou také dělit na konkrétní a abstraktní. Konkrétní entity mohou být konkrétní objekty nebo události, které existují ve fyzickém světě. Abstraktní entitou může být např. myšlenka. Může to být např. vztah, který znamená, že student je zapsán do určitého kurzu.<sup>21</sup>

Předměty musí splnit následující podmínky, aby mohly být entitami:

1. Musí mít vlastní existenci.
2. Každá z instancí předmětu se musí odlišit od ostatních.

---

<sup>18</sup> KROENKE, David a David J. AUER. Databáze. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.

<sup>19</sup> RIORDAN, Rebecca M. *Vytváříme relační databázové aplikace*. Praha: Computer Press, 2000. Databáze. ISBN 80-7226-360-9.

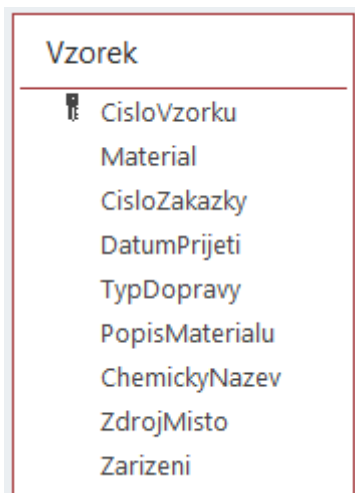
<sup>20</sup> FAVEREAU, Rafael. *Diseño de Bases de Datos [přednáška]*. Santiago de Chile: Universidad Técnica Federico Santa María, 24. května 2020.

<sup>21</sup> RIORDAN, Rebecca M. *Vytváříme relační databázové aplikace*. Praha: Computer Press, 2000. Databáze. ISBN 80-7226-360-9.

3. Všechny instance předmětu musí mít jiné charakteristiky.<sup>22</sup>

### 2.7.2 Atribut

Atribut je element informace, který popisuje nějakou charakteristiku entity.<sup>23</sup>



Obrázek 3: Entita s atributy<sup>24</sup>

Obrázek 3 ukazuje entitu pojmenovanou Vzorek, která má celkem devět atributů: CisloVzorku, Material, CisloZakazky, DatumPrijeti, TypDopravy, PopisMaterialu, ChemickyNazev, ZdrojMisto a Zarizeni.

### 2.7.3 Doména

Doména neboli obor hodnot popisuje typ dat, která jsou daným atributem reprezentována. Doména je logický pojem. Doménou může být např. věk. Doména se může definovat prostým výčtem členů nebo množinou různých pravidel.

### 2.7.4 Vztahy (asociace)

**Asociace (vztah)** je element, pomocí kterého se spojují instance jedné nebo více entit. Nejčastěji náleží nějaké akci (slovesu), která spojuje dva nebo více konceptů. Asociace může být např.: „Nemocnice pečuje o pacienty.“<sup>25</sup>

Vztahy se definují mezi různými entitami. Entity, které jsou zapojeny do určitého vztahu, se

---

<sup>22</sup> RIORDAN, Rebecca M. *Vytváříme relační databázové aplikace*. Praha: Computer Press, 2000. Databáze. ISBN 80-7226-360-9.

<sup>23</sup> FAVEREAU, Rafael. *Diseño de Bases de Datos [přednáška]*. Santiago de Chile: Universidad Técnica Federico Santa María, 24. května 2020.

<sup>24</sup> Vlastní tvorba v programu Access

<sup>25</sup> RIORDAN, Rebecca M. *Vytváříme relační databázové aplikace*. Praha: Computer Press, 2000. Databáze. ISBN 80-7226-360-9.



nazývají účastníci. Vztah je definován svým stupněm. Stupeň vztahu znamená počet účastníků.

Ve většině případů se setkáváme se vztahy binárními jako například: „E-shopy prodávají produkty,“ nebo „Klienti kupují produkty.“ Ternárním vztahem by byl případ, kdy řekneme: „Zákazníci kupují produkty od e-shopů.“

Pomocí těchto dvou binárních vztahů nejsme schopni zjistit, který e-shop prodal jaký produkt klientovi. Tuto informaci dokážeme zjistit pomocí ternárního vztahu.

Speciálním případem je entita, která se účastní vztahu sama se sebou. Tento případ binárního vztahu se nejčastěji používá k reprezentaci hierarchických struktur.

Mezi dvěma libovolnými entitami mohou existovat vztahy typu jedna k jedné, jedna k více nebo více k více. Je důležité zmínit, že vztah více k více se v relačním modelu nedá aplikovat přímo.

Účast entity na vztahu může být úplná nebo částečná. Pokud entita nemůže existovat bez účasti v relaci, jedná se o účast úplnou. V opačném případě se jedná o účast částečnou.<sup>26</sup>

Vedle pojmů jako jsou entita, atribut, doména nebo vztah je důležité vysvětlit další pojmy, jako jsou identifikátor, asociace, stupeň, mohutnost, generalizace nebo specializace.

### 2.7.5 Identifikátor

**Identifikátor** identifikuje jednoznačně instance v relaci, aby se daly rozlišit mezi sebou. Jedna entita může mít více identifikátorů. Pokud entita má více identifikátorů, musí se vybrat jeden hlavní.

Identifikátor se vybírá na základě následujících pravidel.

- Vybere se atribut, který nemění svoji hodnotu během existence instance.
- Vybere se takový atribut, který garantuje pro každou instanci rozdílnou a nenulovou hodnotu v rámci jedné relace.
- Je potřeba zabránit použití struktur, které mohou naznačovat několik věcí.
- Je potřeba zvážit nahrazení identifikátorů nějakou kombinací ostatních.<sup>27</sup>

Definice identifikátoru je velice podobná definici **primárního klíče** v relačním modelu.

---

<sup>26</sup> RIORDAN, Rebecca M. *Vytváříme relační databázové aplikace*. Praha: Computer Press, 2000. Databáze. ISBN 80-7226-360-9.

<sup>27</sup> FAVEREAU, Rafael. *Diseño de Bases de Datos [přednáška]*. Santiago de Chile: Universidad Técnica Federico Santa María, 24. května 2020.

Identifikátor je logická koncepce. Když mluvíme o identifikátoru, jedná se o jeden nebo více atributů. Tyto identifikátory mohou být reprezentovány pomocí klíčů. Primární klíč musí být jedinečný (identifikátor nutně nikoli).<sup>28</sup>

Někdy se může stát, že jako primární klíč musíme použít více polí v tabulce, abychom jednoznačně identifikovali řádek v tabulce. V tomto případě můžeme použít dvě nebo více polí tabulky jako primární klíč a nazýváme je **složený klíč**.<sup>29</sup>

### 2.7.6 Spojovací tabulka

Spojovací tabulka (někdy také propojovací nebo středně pokročilá tabulka) slouží k propojení jiných dvou tabulek. Využívá se, když máme vztah více k více. V relační databázi tento vztah implementovat přímo nemůžeme. K tomu nám slouží právě spojovací tabulka.<sup>30</sup>

### 2.7.7 E-R diagram

E-R diagram (anglicky Entity-relationship model) neboli diagram entit popisuje daný prostor problému za pomoci entit, atributů a domén. Entita v E-R diagramu musí mít nezávislou existenci a musí být jednoznačně odlišitelná od ostatních entit. Peter Chen v roce 1976 zformuloval základy E-R modelu. Původní model od té doby zaznamenal mnohostranný vývoj a dnes mluvíme o rodině E-R modelů.<sup>31</sup>

## 2.8 Normalizace

Při návrhu relačních databázových modelů je důležité zabezpečit, aby model byl schopen odpovědět na rozumně položené otázky a aby se minimalizovaly veškeré redundance a problémy s nimi spojené.<sup>32</sup>

Principy normalizace specifikují pro strukturu relací určitá pravidla, která zabraňují aktualizacím anomáliím. Normalizace není univerzálním nebo povinným předpisem pro vytvoření správného datového modelu. Pokud ovšem datový model dodržuje pravidla relační

---

<sup>28</sup> KROENKE, David a David J. AUER. Databáze. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.

<sup>29</sup> Přidání nebo změna primárního klíče tabulky v Accessu. Podpora Microsoftu [online]. Redmond: Microsoft, 2021 [cit. 2021-7-19]. Dostupné z: <https://support.microsoft.com/cs-cz/office/p%C5%99id%C3%A1n%C3%AD-nebo-zm%C4%9Bna-prim%C3%A1rn%C3%ADho-kl%C3%AD%C4%8De-tabulky-v-accessu-07b4a84b-0063-4d56-8b00-65f2975e4379>

<sup>30</sup> Vytváření relací M:N. Podpora Microsoftu [online]. Redmond: Microsoft, 2021 [cit. 2021-7-19]. Dostupné z: <https://support.microsoft.com/cs-cz/office/video-vytv%C3%A1%C5%99en%C3%AD-relac%C3%AD-m-n-e65bcc53-8e1c-444a-b4fb-1c0b8c1f5653>

<sup>31</sup> POKORNÝ, Jaroslav a Michal VALENTA. Databázové systémy. 2. přepracované vydání. Praha: Česká technika - nakladatelství ČVUT, 2020. ISBN 978-80-01-06696-6.

<sup>32</sup> RIORDAN, Rebecca M. *Vytváříme relační databázové aplikace*. Praha: Computer Press, 2000. Databáze. ISBN 80-7226-360-9.

struktury (datový model je normalizovaný), je pravděpodobné, že dostaneme funkční a efektivní databázi.

Princip bezztrátové dekompozice je proces, kdy se rozdělí původní relace tak, aby se daly nové relace zpětně spojit bez ztráty informace a aby došlo k odstranění redundance.<sup>33</sup>

Normalizace je proces, který analyzuje závislosti mezi atributy za účelem zjednodušení modelu. V tomto procesu se kombinují atributy v entitách a asociacích méně komplexních.<sup>34</sup>

### 2.8.1 První normální forma

Relace je v první normální formě, pokud každý atribut vektoru hodnot obsahuje pouze jednu skalární hodnotu.<sup>35</sup>

Prvně se identifikují skupiny opakujících se atributů, které existují nebo mohou existovat v každé entitě. Poté se tyto atributy oddělí od entity a vytvoří entitu novou.

Nové entity musí mít nějaký atribut společný s původní entitou. Jedna z relací, která je závislá nebo slabá musí obsahovat složený primární klíč nebo jeden atribut, který slouží jako identifikátor a cizí klíč.

Cizí klíč je atribut, který slouží jako identifikátor jedné entity, který je přenesen do jiné entity.<sup>36</sup>

Výsledkem první normální formy je, že každý atribut obsahuje pouze atomické hodnoty. To znamená, že obsahuje hodnoty, které se z pohledu databáze nedají dělit.<sup>37</sup>

### 2.8.2 Druhá normální forma

První normální forma obsahuje částečné závislosti. Částečná závislost znamená, že jeden nebo více atributů, které nejsou klíčem, jsou závislé na složeném primárním klíči.

Pokud entita obsahuje jedinečný primární klíč, druhá normální forma se na entitu neaplikuje.<sup>38</sup>

---

<sup>33</sup> RIORDAN, Rebecca M. *Vytváříme relační databázové aplikace*. Praha: Computer Press, 2000. Databáze. ISBN 80-7226-360-9.

<sup>34</sup> FAVEREAU, Rafael. *Diseño de Bases de Datos [přednáška]*. Santiago de Chile: Universidad Técnica Federico Santa María, 24. května 2020.

<sup>35</sup> RIORDAN, Rebecca M. *Vytváříme relační databázové aplikace*. Praha: Computer Press, 2000. Databáze. ISBN 80-7226-360-9.

<sup>36</sup> FAVEREAU, Rafael. *Diseño de Bases de Datos [přednáška]*. Santiago de Chile: Universidad Técnica Federico Santa María, 24. května 2020.

<sup>37</sup> BÍLA, Jiří, František KRÁL a Vladimír HLAVÁČ. *Informační technologie: databázové a znalostní systémy*. Vyd. 2. přeprac. Praha: Vydavatelství ČVUT, 2003. ISBN 80-01-02790-2.

<sup>38</sup> FAVEREAU, Rafael. *Diseño de Bases de Datos [přednáška]*. Santiago de Chile: Universidad Técnica Federico Santa María, 24. května 2020.

Částečná závislost se odstraňuje vytvořením nových entit souvisejících s entitou originální. Jedna z nových entit obsahuje atributy závislé na kompletním složeném primárním klíči a další entita nebo entity obsahují atributy závislé jen na části daného složeného primárního klíče.

Vytvořené entity musí obsahovat nějaký společný atribut reprezentující jejich asociaci.

### 2.8.3 Třetí normální forma

Druhá normální forma obsahuje problémy a anomálie, jelikož obsahuje transitivní závislosti.

Třetí normální forma odstraňuje závislosti neklíčových atributů, které závisí na jiných neklíčových attributech.

Tyto závislosti se opět odstraňují vytvořením nových entit spojených s původní entitou. Původní entita obsahuje atributy závislé na primárním klíči a musí obsahovat atribut, který je primárním klíčem nové entity. Tento atribut v původní entitě se nazývá cizí klíč.

### 2.8.4 Konceptuální model v kanonickém tvaru

Konceptuální model v kanonickém tvaru je konceptuální datový model s jednodušší strukturou, která dovoluje budoucí údržbu databáze, dokumentaci a lehčí porozumění.

Je zapotřebí alespoň třetí normální formy. Existují vyšší řády normalizace, které umožňují optimalizaci modelu. První tři normální formy ovšem zaručují efektivní a dostatečnou optimalizaci modelu.

### 2.8.5 Požadavky na informace uživatelů

Po vytvoření konceptuálního modelu v kanonickém tvaru je potřeba aplikovat požadavky na informace uživatelů. Především je nutné definovat:

- Vstupní body do datové struktury,
- Cesty vyhledávání při procházení, aby byly splněny všechny požadavky na informace.

Po definování těchto požadavků se ověří, jestli je možno každý z nich uspokojit za pomoci navrhnutého datového modelu. Pokud některý z požadavků není splněn je třeba vrátit se k procesu vytvoření konceptuálního modelu (E-R diagramu).<sup>39</sup>

---

<sup>39</sup> FAVEREAU, Rafael. Diseño de Bases de Datos [přednáška]. Santiago de Chile: Universidad Técnica Federico Santa María, 24. května 2020.

### 2.8.6 Relační model

Pokud návrh konceptuálního modelu odpovídá požadavkům na informace uživatelů, musí konečná implementace databáze odpovídat kritériím technologické efektivity pro její časté, efektivní a účinné použití (použití v reálném hardwaru).<sup>40</sup>

V roce 1970 Edgar Frank Codd vyvinul relační model, který je dnes nejrozšířenější k uložení a dynamické administraci dat v databázi při průmyslovém využití reálných problémů.<sup>41</sup>

Relační model se skládá z datových tabulek a vztahů. Uživatelé mohou přistupovat k řadě dat prostřednictvím dotazů nebo operací prováděných relačním systémem RDBMS (Relational Database Management System).

- Relační model vytváří propojení mezi různými soubory (tabulkami) prostřednictvím společných údajů v nich, takže umožňuje větší diverzifikaci dat a flexibilitu při jejich konstrukci.
- Každá tabulka v relačním modelu je sada záznamů a každý záznam je sada polí.
- Vztah mezi nadřazenou a podřazenou tabulkou se uskutečňuje pomocí primárních a cizích klíčů.
- V současné době se jedná o nejpoužívanější konstrukci při vytváření podnikových databází.

Relační tabulky jsou tabulky (entity), které splňují následující podmínky.

- Každý řádek nebo záznam musí být jedinečný, bez duplicit.
- Každý sloupec nebo pole musí být jedinečné, bez duplicit.
- Hodnoty polí musí patřit do domény každého atributu.
- Každý řádek musí mít předem definovaný jedinečný formát.
- Každé pole (sloupec) každého záznamu (řádku) musí obsahovat jednu hodnotu.<sup>42</sup>

### 2.8.7 Pravidla transformace E-R diagramu na relační model

Každá entita v kanonickém modelu je transformována do tabulky v relačním modelu a jakýkoli atribut entity je transformován na pole v tabulce, přičemž jsou zachovány primární, cizí nebo složené klíče.

---

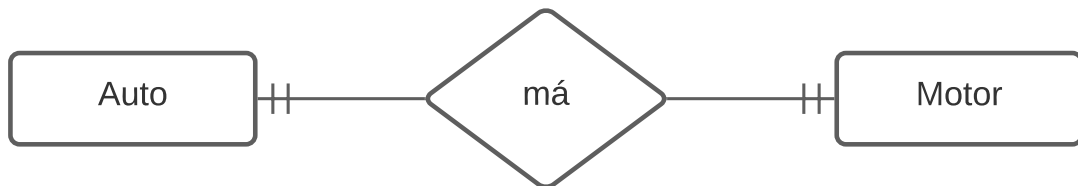
<sup>40</sup> FAVEREAU, Rafael. Diseño de Bases de Datos [přednáška]. Santiago de Chile: Universidad Técnica Federico Santa María, 24. května 2020.

<sup>41</sup> KROENKE, David a David J. AUER. Databáze. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.

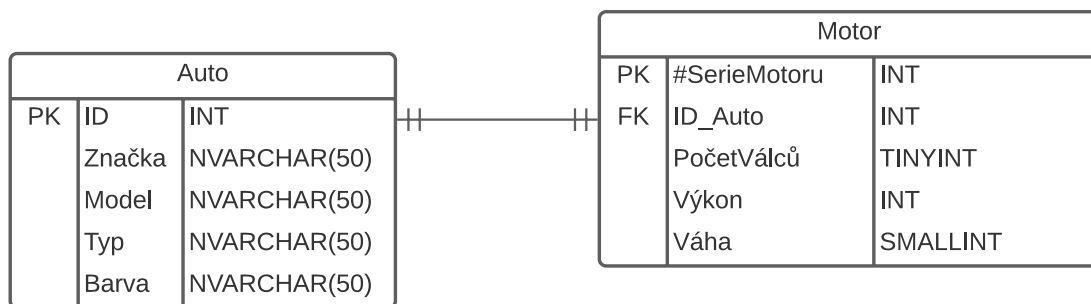
<sup>42</sup> FAVEREAU, Rafael. Diseño de Bases de Datos [přednáška]. Santiago de Chile: Universidad Técnica Federico Santa María, 24. května 2020.

**Při výskytu vztahů 1:1** se transformace provádí dle následujících pravidel.

- Primární klíč silné entity se propaguje jako cizí klíč v závislé nebo slabé entitě.
- Od této chvíle jsou obě tabulky propojeny dvojicí klíčů primární klíč - cizí klíč, který důsledně odpovídá stejnému poli (metadatům).
- Pokud má jedna z entit volitelnou kardinalitu, primární klíč entity s povinnou kardinalitou (bez nuly) je propagován do entity s nepovinnou kardinalitou (s nulou).<sup>43</sup>



Obrázek 4: Grafické zobrazení E-R diagramu se vztahem 1:1<sup>44</sup>



Obrázek 5: E-R diagram se vztahem 1:1<sup>45</sup>

**Při výskytu vztahů 1:N (povinných nebo nepovinných)** se transformace provádí dle následujících pravidel.

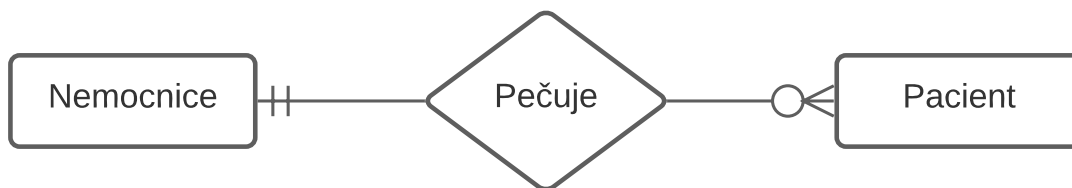
- Silná entita (s kardinalitou jedna) propaguje svůj primární klíč do slabé entity, který je přidán do seznamu atributů slabé entity jako cizí klíč.
- Vytvořené tabulky spolu souvisejí prostřednictvím primárního klíče - cizího klíče, které souvisle odpovídají stejnému poli. Tím zmizí název vztahu.
- Pokud je entita volitelná (minimální kardinalita vztahu je 0), můžeme nastavit, že atributy cizího klíče relace mohou nabývat prázdných hodnot. Další možností je vytvořit spojovací tabulku. Primární klíč nové spojovací tabulky bude primární klíč

<sup>43</sup> FAVEREAU, Rafael. Diseño de Bases de Datos [přednáška]. Santiago de Chile: Universidad Técnica Frederico Santa María, 24. května 2020.

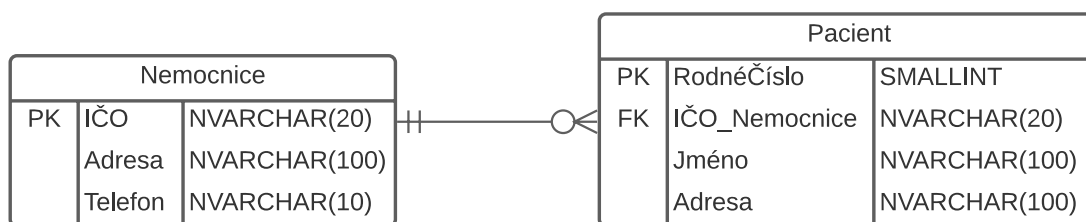
<sup>44</sup> Vlastní zpracování

<sup>45</sup> Vlastní tvorba

entity s kardinalitou N.<sup>46 47</sup>



Obrázek 6: Grafické zobrazení E-R diagramu se vztahem 1:N<sup>48</sup>



Obrázek 7: E-R diagram se vztahem 1:N<sup>49</sup>

**Při výskytu vztahů M:N** je potřeba vytvořit spojovací tabulku dle následujících pravidel.

- Vytvoří se nová spojovací tabulka. Primární klíče nové spojovací tabulky tvoří primární klíče obou entit. Spojovací tabulka dále obsahuje atributy vztahu (pokud existují).
- Každá z původních tabulek je spojena s novou tabulkou prostřednictvím příslušného propagovaného klíče.<sup>50</sup>



Obrázek 8: Grafické zobrazení E-R diagramu se vztahem M:N<sup>51</sup>

<sup>46</sup> FAVEREAU, Rafael. Diseño de Bases de Datos [přednáška]. Santiago de Chile: Universidad Técnica Federico Santa María, 24. května 2020.

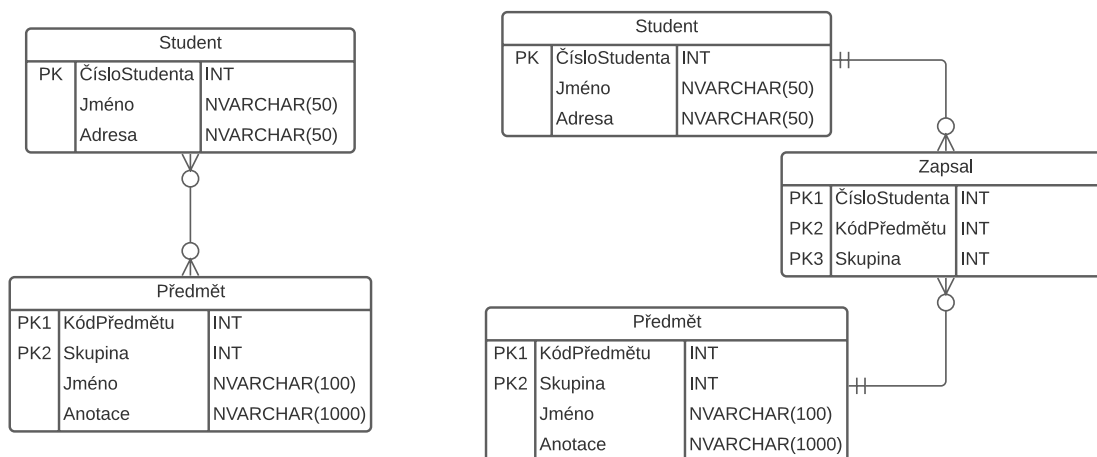
<sup>47</sup> POKORNÝ, Jaroslav a Michal VALENTA. Databázové systémy. 2. přepracované vydání. Praha: Česká technika - nakladatelství ČVUT, 2020. ISBN 978-80-01-06696-6.

<sup>48</sup> Vlastní tvorba

<sup>49</sup> Vlastní tvorba

<sup>50</sup> FAVEREAU, Rafael. Diseño de Bases de Datos [přednáška]. Santiago de Chile: Universidad Técnica Federico Santa María, 24. května 2020.

<sup>51</sup> Vlastní tvorba



Obrázek 9: E-R diagram se vztahem M:N<sup>52</sup>

## 2.9 SQL

SQL je strukturovaný dotazovací jazyk pro relační databáze, který umožňuje vytvářet a zadávat různé typy databázových operací.<sup>53</sup>

Jazyk SQL (Structured Query Language) není úplný programovací jazyk. Tento jazyk obsahuje konstrukce, pomocí kterých můžeme definovat a zpracovávat databáze.

Jazyk SQL vznikl na konci sedmdesátých let ve společnosti IBM Corporation. Konkrétní verze jazyka SQL schválila jako americký standard organizace ANSI (American National Standards Institute) nebo jako mezinárodní standard organizace ISO (International Organization for Standardization). Mezi poměrně novou a významnou vlastnost jazyka SQL patří například podpora jazyka XML (Extensible Markup Language).

SQL je textově orientovaný jazyk a k psaní jeho příkazů stačí pouze textový editor. Současné systémy jako Microsoft Access, Microsoft SQL Server, MySQL a další poskytují grafické nástroje plnící různé funkce jazyka SQL. Tyto grafické nástroje ovšem neposkytují všechny možnosti jazyka SQL.<sup>54</sup>

Příkazy SQL se rozdělují do čtyřech hlavních kategorií:

1. Data Definition Language ( DDL),
2. Data Manipulation Language (DML),

<sup>52</sup> Vlastní tvorba

<sup>53</sup> FAVEREAU, Rafael. Structured Query Language [přednáška]. Santiago de Chile: Universidad Técnica Federico Santa María, 9. června 2020.

<sup>54</sup> KROENKE, David a David J. AUER. Databáze. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.



3. Data Control Language (DCL).
4. Data Query Language (DQL).<sup>55</sup>

Příkazy jazyka DDL umožňují definovat databázové struktury.

Příkazy jazyka DML se používá k dotazování a k upravování databázových dat.<sup>56</sup>

Příkazy DCL slouží k ovládání a zabezpečení systému.<sup>57</sup>

### 2.9.1 Data Definition Language (DDL)

Před vytvořením tabulek je nezbytné vytvořit databázi. Vytvořit databázi můžeme pomocí SQL příkazu, dle Davida Kroenkeho a Davida Auera ovšem většina vývojářů databáze používá k tomuto úkonu grafické nástroje.<sup>58</sup>

Jazyk DDL obsahuje příkazy:

- CREATE slouží k vytváření databáze nebo databázových objektů,
- ALTER slouží k úpravě databáze nebo databázových objektů,
- DROP slouží k vymazání databáze nebo databázových objektů,
- TRUNCATE slouží k vymazání všech záznamů tabulky.

### 2.9.2 Data Manipulation Language (DDM)

- INSERT slouží k vkládání dat do tabulky,
- UPDATE slouží k aktualizaci dat v tabulce,
- DELETE slouží k mazání dat z tabulky,
- MERGE slouží k porovnání cílové a zdrojové tabulky kombinací příkazů INSERT, UPDATE a DELETE.

### 2.9.3 Data Query Language (DQL)

Jazyk DQL je využíván k provádění dotazu nad daty v tabulkách nebo nad daty databázových objektů. K tomu slouží příkaz SELECT, pomocí kterého získáváme data z databáze nebo

---

<sup>55</sup> Lekce 9: Co je DDL, DML a DQL? Learn2Code [online]. Žilina: Learn2Code o.z., 2021 [cit. 2021-6-22]. Dostupné z: <https://learn2code.cz/aplikace/skupiny#/sql-data-analyza-i/chapters/lekce-9-co-je-ddl-dml-a-dql>

<sup>56</sup> KROENKE, David a David J. AUER. Databáze. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.

<sup>57</sup> FAVEREAU, Rafael. Structured Query Language [přednáška]. Santiago de Chile: Universidad Técnica Frederico Santa María, 9. června 2020.

<sup>58</sup> KROENKE, David a David J. AUER. Databáze. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.

vytváříme databázové objekty.<sup>59</sup>

#### 2.9.4 Typy dat v SQL

V SQL můžeme definovat následující datové typy:

- numerické (přesné a aproximativní),
- znakové řetězce,
- bitové řetězce,
- temporální data,
- časové datové typy.

Mezi přesné numerické typy patří INT, BIGINT, SMALLINT, NUMERIC a DECIMAL. BIGINT je celé číslo a disponuje větším rozsahem než INT. INT je celé číslo v rozsahu od – 2 147 483 648 až 2 147 483 648, SMALLINT je celé číslo v rozsahu -32 768 až 32 767, NUMERIC a DECIMAL jsou desetinná čísla s fixním počtem desetinných míst. Mezi aproximativní numerické typy řadíme FLOAT, REAL a DOUBLE PRECISION.<sup>60</sup>

Mezi znakové řetězce patří CHAR a VARCHAR. CHAR je znakový řetězec obsahující přesný počet znaků. VARCHAR má libovolnou délku řetězce s nejvyšším povoleným počtem znaků. Je možno využít také datové typy NCHAR a NVARCHAR, které umožňují ukládat znaky ze sady Unicode (speciální znaky, háčky, čárky, atd.).

Mezi časové datové typy patří DATE, DATETIME, DATETIME2, SMALLDATETIME a TIME. Např. pro zobrazení pouze datumu ve formátu rok-měsíc-den (RRRR-MM-DD) můžeme použít DATE.<sup>61</sup>

### 2.10 MS Access

MS Access je relační databáze, která pracuje s dotazovacím jazykem SQL. Zároveň to je program, který je součástí kancelářského balíku MS Office.<sup>62</sup>

---

<sup>59</sup> Lekce 9: Co je DDL, DML a DQL? Learn2Code [online]. Žilina: Learn2Code o.z., 2021 [cit. 2021-6-22]. Dostupné z: <https://learn2code.cz/aplikace/skupiny#/sql-data-analyza-i/chapters/lekce-9-co-je-ddl-dml-a-dql>

<sup>60</sup> POKORNÝ, Jaroslav a Michal VALENTA. Databázové systémy. 2. přepracované vydání. Praha: Česká technika - nakladatelství ČVUT, 2020. ISBN 978-80-01-06696-6.

<sup>61</sup> SQL Data analýza I. Learn2Code [online]. Žilina: Learn2Code, 2021 [cit. 2021-7-19]. Dostupné z: <https://learn2code.cz/kurzy/sql-data-analyza-i>

<sup>62</sup> Lekce 1 - Úvod a prostředí programu MS Access. ITnetwork.cz [online]. Praha: Luboš Marvan, 2021 [cit. 2021-7-19]. Dostupné z: <https://www.itnetwork.cz/ms-office/microsoft-access-zaklady/uvod-a-prostredi-programu-ms-access>

Pomocí Accessu je možno vytvářet podnikové aplikace. K vytvoření takovéto aplikace není zapotřebí umět psát kód.<sup>63 64</sup>

Microsoft Access byl společností Microsoft navržen jako „osobní databázový systém“. Při práci s MS Access můžeme tedy využívat funkcí databázové aplikace i funkcí systému řízení databáze.<sup>65</sup>

V Accessu je možno:

- přidávat nová data do databáze,
- upravovat existující data v databázi,
- odstraňovat informace,
- uspořádat a prohlížet data,
- sdílet data pomocí sestav, e-mailových zpráv, intranetu nebo internetu.<sup>66</sup>

Accessová databáze se skládá z tabulek, formulářů, sestav, dotazů, maker a modulů.

### 2.10.1 Desktopová a webová aplikace Access

V Accessu je možno vytvořit desktopovou i webovou aplikaci. Desktopová aplikace se používá, když se každý uživatel aplikace může připojit k počítači na kterém je databáze Accessu uložena. Webová aplikace Accessu se používá, když uživatelé aplikace potřebují mít přístup k datům z různých míst a z mnoha různých zařízení.<sup>67</sup>

### 2.10.2 Tabulky

Tabulka v Accessu musí splňovat podmínky relační tabulky popsané v bodě 2.8.6. Je nutné zajistit uspořádání dat, aby se zamezilo redundancím. Tím se docílí normalizací popsané v bodě 2.8.

---

<sup>63</sup> Databázový software a aplikace | Microsoft Access. Podpora Microsoftu [online]. Redmond: Microsoft, 2021 [cit. 2021-7-19]. Dostupné z: <https://www.microsoft.com/cs-cz/microsoft-365/access>

<sup>64</sup> Co je Access? Podpora Microsoftu [online]. Redmond: Microsoft, 2021 [cit. 2021-7-19]. Dostupné z: <https://support.microsoft.com/cs-cz/office/video-co-je-access-f2338765-ff59-4cfc-b8ba-74059fcb1874>

<sup>65</sup> KROENKE, David a David J. AUER. Databáze. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.

<sup>66</sup> Základní informace o databázích. Podpora Microsoftu [online]. Redmond: Microsoft, 2021 [cit. 2021-7-19]. Dostupné z: <https://support.microsoft.com/cs-cz/office/z%C3%A1kladn%C3%AD-informace-o-datab%C3%A1z%C3%ADch-a849ac16-07c7-4a31-9948-3c8c94a7c204>

<sup>67</sup> Rozhodněte se, jestli chcete vytvořit desktopovou databázi nebo webovou aplikaci pro Access. Podpora Microsoftu. [online]. Redmond: Microsoft, 2021 [cit. 2021-7-21]. Dostupné z: <https://support.microsoft.com/cs-cz/office/rozhodn%C4%9Bte-se-jestli-chnete-vytvo%C5%99it-desktopovou-datab%C3%A1zi-nebo-webovou-aplikaci-pro-access-7bf7ccc9-0850-48f2-858f-273271d30fa0>

### 2.10.3 Formuláře

Pomocí formulářů lze vytvořit uživatelské rozhraní, ve kterém se zadávají a upravují data. Formuláře mohou např. obsahovat příkazová tlačítka nebo další ovládací prvky. Databázi je možno upravit přímou úpravou dat v datových listech tabulek i bez použití formulářů.

Dají se naprogramovat příkazová tlačítka, která určují jaká data zobrazit, jaký formulář nebo sestavu otevřít nebo provádět jiné úkoly.

Pomocí formulářů se dá zajistit ochrana dat a zadávání dat správným způsobem. Dá se např. vytvořit formulář, který zobrazuje jenom určitá pole a dovoluje provést jen určité operace.

### 2.10.4 Sestavy

Pomocí sestav se formátují, sumarizují a prezentují data. Sestava vždy prezentuje aktuální data databáze a je možno ji spustit kdykoliv.

Sestava se může formátovat za účelem vytištění nebo exportu do jiné aplikace. Také je možno sestavu poslat jako přílohu e-mailové zprávy.

### 2.10.5 Dotazy

Pomocí dotazů je možné provádět různé funkce. Nejčastější funkcí bývá načtení specifických dat z tabulek. Rozlišujeme dva základní druhy dotazů: **výběrové** a **akční**.

Výběrový dotaz načte a zpřístupní data k použití (prohlédnutí na obrazovce, vytisknutí nebo zkopírování do schránky). Lze jej také použít jako zdroj záznamů pro formulář nebo sestavu.

Akční dotaz vytváří nové tabulky, přidává data do existujících tabulek, aktualizuje data nebo data odstraňuje.

Dotaz je „aktualizovatelný“, pokud jeho pomocí můžeme upravovat data v podkladových tabulkách prostřednictvím datového listu dotazu.<sup>68</sup>

### 2.10.6 Makra

Pomocí maker je možno přidat funkce do databáze. Makro je možno připojit k příkazovému tlačítku. V tom případě se v momentě stisku tlačítka spustí dané makro. Makra mohou otevírat sestavy, spouštět dotazy nebo zavírat databáze.

---

<sup>68</sup> Základní informace o databázích. Podpora Microsoftu [online]. Redmond: Microsoft, 2021 [cit. 2021-7-19]. Dostupné z: <https://support.microsoft.com/cs-cz/office/z%C3%A1kladn%C3%AD-informace-o-datab%C3%A1z%C3%ADch-a849ac16-07c7-4a31-9948-3c8c94a7c204>

## 2.10.7 Moduly

Pomocí modulů je možno přidávat funkce do databáze psaním kódu v programovacím jazyce Visual Basic for Applications (VBA).<sup>69</sup>

Program Access umožňuje převést makra na moduly nebo moduly třídy jazyka VBA.

Je doporučováno používat makra před jazykem VBA, pokud se databáze bude sdílet s jinými uživateli umístěním na souborový server s ohledem na bezpečnost. Pomocí jazyka VBA je totiž možno napsat kód, který ohrozí bezpečnost dat nebo poškodí soubory na počítači. Z tohoto důvodu je doporučováno používat jazyk VBA jenom pro operace, které nejdou provést pomocí maker.<sup>70</sup>

## 2.10.8 Datové typy v MS Accessu (verze 2013 a novější)

MS Access používá jiné datové typy než byly popsány v kapitole 2.9.4 Typy dat v SQL.

Jako textová pole jsou využívány datové typy **dlouhý text**, který může uložit až jeden gigabajt textu. Ve formulářích a sestavách je možno zobrazit jenom prvních 64 000 znaků. Dlouhý text ve webové aplikaci pro Access slouží jako ekvivalent datového typu SQL Serveru NVARCHAR(max). Dalším datovým typem je **krátký text**, který ukládá až 255 znaků v desktopové aplikaci a ve webové aplikaci až 4000 znaků. Krátký text je ekvivalentem NVARCHAR v SQL Serveru.<sup>71</sup>

Jako číselná pole jsou používané datové typ číslo a velké číslo. Datový typ **číslo** v Accessu může být celé číslo, číslo s plovoucí desetinnou čárkou nebo číslo s pevnou desetinnou čárkou. Odpovídajícím datovým typem SQL Serveru může být INT, DOUBLE nebo DECIMAL(28,6). Datový typ **dlouhé číslo** slouží k uchovávání velkých číselných hodnot.<sup>72</sup>

Jako časové pole se používají Datum a čas. **Datum a čas** slouží k uchování kalendářních a

---

<sup>69</sup> Základní informace o databázích. Podpora Microsoftu [online]. Redmond: Microsoft, 2021 [cit. 2021-7-19]. Dostupné z: <https://support.microsoft.com/cs-cz/office/z%C3%A1kladn%C3%AD-informace-o-datab%C3%A1z%C3%ADch-a849ac16-07c7-4a31-9948-3c8c94a7c204>

<sup>70</sup> Úvod k programování v aplikaci Access. Podpora Microsoftu [online]. Redmond: Microsoft, 2021 [cit. 2021-7-19]. Dostupné z: <https://support.microsoft.com/cs-cz/office/%C3%BAvod-k-programov%C3%A1n%C3%AD-v-aplikaci-access-92eb616b-3204-4121-9277-70649e33be4f#bmconvertmacrostvba>

<sup>71</sup> Datový typ Memo se teď nazývá Dlouhý text. Podpora Microsoftu [online]. Redmond: Microsoft, 2021 [cit. 2021-7-21]. Dostupné z: <https://support.microsoft.com/cs-cz/office/datov%C3%BD-ty-p-memo-se-te%C4%8F-naz%C3%BDv%C3%A1-dlouh%C3%BD-text-dffe5e34-953e-4451-a05e-fba5d9b564b5>

<sup>72</sup> Datové typy pro desktopové databáze Accessu. Podpora Microsoftu [online]. Redmond: Microsoft, 2021 [cit. 2021-7-21]. Dostupné z: [https://support.microsoft.com/cs-cz/office/datov%C3%A9-ty-py-pro-desktopov%C3%A9-datab%C3%A1ze-accessu-df2b83ba-cef6-436d-b679-3418f622e482#ID0EAABAAA=Newer\\_versions](https://support.microsoft.com/cs-cz/office/datov%C3%A9-ty-py-pro-desktopov%C3%A9-datab%C3%A1ze-accessu-df2b83ba-cef6-436d-b679-3418f622e482#ID0EAABAAA=Newer_versions)

časových údajů. Jejich ekvivalentem SQL Serveru může být DATE, TIME(3) nebo DATETIME2(3).<sup>73 74</sup>

MS Access má spoustu dalších datových typů. Zde jsem zmínil hlavně ty datové typy, které jsem následně použil v praktické části.

### 3 Definování potřeb a funkcí databázového systému pro podnik

Společnost se zabývá především zakázkovou výrobou a dělí se na následující divize:

- Technologie pro kolejovou dopravu,
- **Technologie pro manipulaci se sypkými materiály,**
- Navažovací technologie.



Obrázek 10: Divize firmy<sup>75</sup>

Ve společnosti vznikla potřeba vytvořit databázový systém pro divizi firmy Technologie pro manipulaci se sypkými materiály za účelem sběru a uchování dat ze zkoušek vzorků sypkých materiálů.

Společnost v rámci divize Technologie pro manipulaci se sypkými materiály disponuje od roku 2011 zkušební linkou a laboratoří pro testování sypkých materiálů. Od té doby bylo již provedeno 672 zkoušek vzorků sypkých materiálů. Z těchto zkoušek se uchovává 132 XLS

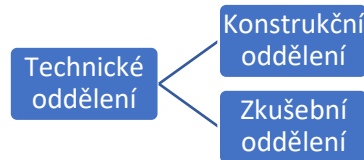
<sup>73</sup> Datové typy pro webové aplikace Accessu. Podpora Microsoftu [online]. Redmond: Microsoft, 2021 [cit. 2021-7-21]. Dostupné z: <https://support.microsoft.com/cs-cz/office/datov%C3%A9-typy-pro-webov%C3%A9-aplikace-accessu-33e10f90-5b06-4e56-ab0e-b0930f4c2367>

<sup>74</sup> Datové typy pro desktopové databáze Accessu. Podpora Microsoftu [online]. Redmond: Microsoft, 2021 [cit. 2021-7-21]. Dostupné z: [https://support.microsoft.com/cs-cz/office/datov%C3%A9-typy-pro-desktopov%C3%A9-datab%C3%A1ze-accessu-df2b83ba-cef6-436d-b679-3418f622e482#ID0EAABAAA=Newer\\_versions](https://support.microsoft.com/cs-cz/office/datov%C3%A9-typy-pro-desktopov%C3%A9-datab%C3%A1ze-accessu-df2b83ba-cef6-436d-b679-3418f622e482#ID0EAABAAA=Newer_versions)

<sup>75</sup> Divize firmy. Technologie pro kolejovou dopravu, manipulaci se sypkými materiály a vážící systémy - Sklenář [online]. Dlouhoňovice: Sklenář, c2021 [cit. 2021-6-24]. Dostupné z: <https://www.sklenar.cz/cs/m-2-divize-firmy>

souborů vytvořených v programu Microsoft Excel.

Technické oddělení divize Technologie pro manipulaci se sypkými materiály se dělí na Konstrukční oddělení a Zkušební oddělení.



Obrázek 11: Technické oddělení divize Technologie pro manipulaci se sypkými materiály <sup>76</sup>

Zkušební oddělení má k dispozici zkušební centrum a laboratoř, kde probíhají testy sypkých materiálů. Výsledky testů jsou dále sdíleny s konstruktéry, kteří mohou navrhnout efektivnější řešení pro zákazníka.

V rámci divize Technologie pro manipulaci se sypkými materiály společnost nabízí následující produkty:

- Pneumatická doprava,
- Rotační podavače a odbočky,
- Komponenty pro pneumatickou dopravu,
- Vyprazdňování zásobníků a sil,
- Mechanická doprava,
- Vzorkování,
- Balení.

Společnost využívá cloudových uložišť a disponuje informačním systémem K2, kde má zřízenou databázi především pro fakturaci a sledování skladu. Konstrukční kancelář je vybavena 3D CAD softwarem od Dassault Systemes SOLIDWORKS. Všechny modely a výkresy jsou ukládány bezpečně v cloudu.

Zkušební laboratoř je mimo měřícího vybavení a přístrojů vybavena stolním počítačem na kterém se vyplňují protokoly z měření do již vytvořené šablony v Excelu. Z Excelu je uložen XLS soubor, který se poté uloží do cloudu.

---

<sup>76</sup> Vlastní tvorba

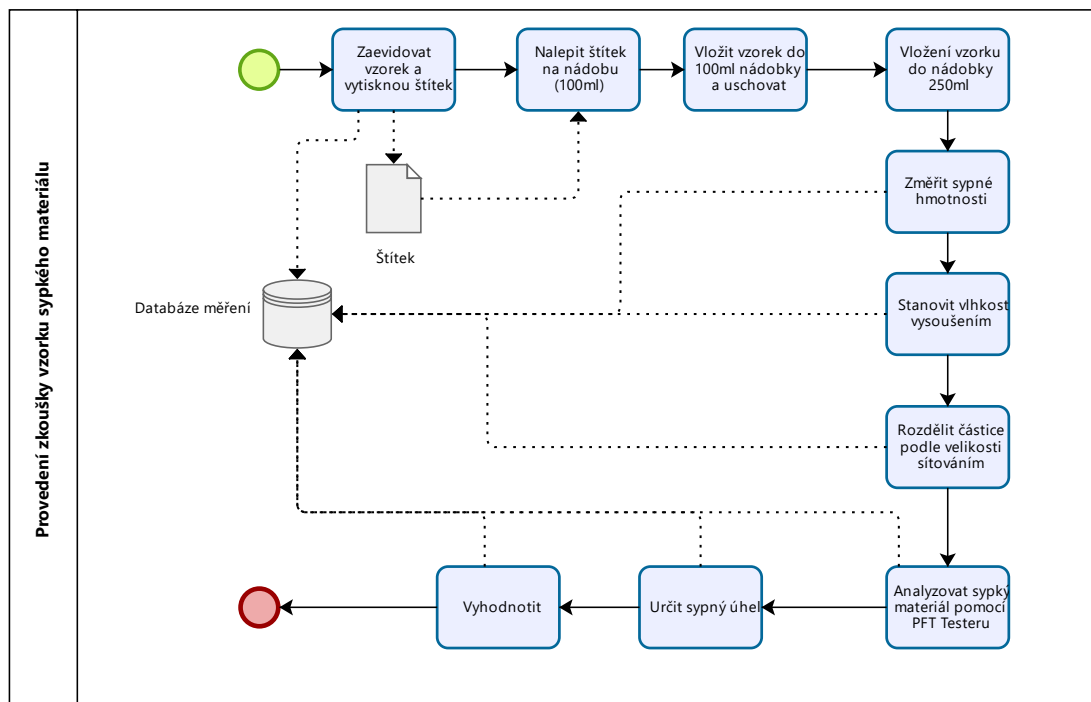
## 4 Návrh datového modelu a transakcí

V této části se věnuji analýze procesu a poté samotnému návrhu řešení.

### 4.1 Analýza procesu

V kapitole analýza procesu jsem zanalyzoval proces pomocí sledování procesu ve zkušební laboratoři podniku a pomocí analýzy poskytnuté dokumentace.

#### 4.1.1 Procesní mapa



Powered by  
**bizagi**  
Modeler

Obrázek 12: Procesní mapa - zkouška sypkých materiálů <sup>77</sup>

Na předešlém obrázku je vidět procesní mapa procesu zkoušky sypkých materiálů. V samotné laboratoři, kde se zkoušky sypkých materiálů provádí jsem se seznámil s procesem zkoušky sypkých materiálů. Procesní mapu jsem vytvořil za účelem dokumentace procesu pro jednodušší vytvoření konceptuálního datového modelu.

<sup>77</sup> Vlastní tvorba v programu Bizagi



#### 4.1.2 Dosavadní dokumentace

	A	B	C	D
8	<b>VYHODNOCENÍ ZKOUŠEK VZORKŮ MATERIÁLU</b>			
9				
10				
11	Zákazník / uživatel:	Johns Manville		
12	Číslo nabídky/zakázky:	20/2021/16	List: 1 z 1	
13				
14				
15	<b>CHARAKTERISTIKA MATERIÁLU</b>			
16	Materiál:	recyklát		
17	Číslo vzorku:	č.v.672		
18	Popis materiálu:	velmi jemný bílý materiál (vzorek po zkouškách linka)		
19	Sypná hmotnost volná:	1470	kg/m <sup>3</sup>	-
20	Sypná hmotnost sestřsná (1 dm <sup>3</sup> ):	-	kg/m <sup>3</sup>	-
21	Sypná hmotnost počáteční:	1405	kg/m <sup>3</sup>	PFT Tester
22	Sypná hmotnost konečná:	1659	kg/m <sup>3</sup>	PFT Tester
23	Sypný úhel: (sklon od vodorovné roviny)	-	°	-
24	Velikost částic:	0-1	mm	
25	Střední velikost částic:	0,121	mm	
26	Podíl částic pod 1 mm:	16,1	%	
27				
28	Vlhkost (stanovení vlhkosti vysoušením při 80°C :	0,10	%	MLB50-3N
29	stupeň tvrdosti dle Mohse:	-		
30				
31	Klasifikace vlastností:	snadno/volně tekoucí		PFT Tester
32	Efektivní úhel vnitřního tření:	36,5-34,7	°	PFT Tester
33	Efektivní úhel vnějšího tření:	23,8-19,4	°	PFT Tester
34	Předpoklad klenbování:	NE		PFT Tester
35	Předpoklad tvorby kominu:	ANO		PFT Tester
36	Navržený vnitřní úhel výstupního kužele při výpadu DN 300:	58	°	PFT Tester
37	Kritická měrná hmotnost pro tvoření klenby:	1412	kg/m <sup>3</sup>	PFT Tester
38	Kritická měrná hmotnost pro vznik středního tunelu:	1761	kg/m <sup>3</sup>	PFT Tester
39				
40				
41	<b>Závěr (vyhodnocení): Velikost částic změřena-rozpětí 0-1 mm.Střední velikost částic 0,121 mm. 16,1%</b>			
42	<b>částic je menších než 45um. Vlhkost materiálu 0,10 %.</b>			
43	<b>Úhel vnitřního tření je nízký (úhel 36,5-34,7°), napětí má malý vliv na hodnotu úhlu. Toková funkce-materiál</b>			
44	<b>je hodnocen jako snadno tekoucí, od napětí 2,5 kPa jako volně tekoucí. Hodnota koheze-nízká (0,100</b>			
45	<b>kPa).Úhel vnějšího-stěnového tření (kontaktní materiál ocel tř.17) je střední (23,8-19,45°). Během</b>			
46	<b>vyprazdňování se úhel zvyšuje (nejnižší je na začátku vyprazdňování (19,3-14,7°). Stlačitelnost-koefficient 1,07</b>			
47	<b>(nízký). Materiál je problematický při pseudopravě-je abrazivní a při malém rychlosti vzduchu hrozí nebezpečí ucpání.</b>			
48	<b>Není vhodné, aby tento materiál zůstal v potrubí. Při vyprazdňování zásobníku se problémy neočekávají.</b>			
49	Zkoušku provedl: ing.Keprta Miroslav		Datum: 11.5.2021	

Obrázek 13: Protokol ze zkoušky sypkého materiálu <sup>78</sup>

Na předchozím obrázku je k nahlédnutí protokol ze zkoušky, který vznikne po zkoušce a který je poté sdílen s konstruktéry.

<sup>78</sup> Interní dokumentace podniku Sklenář s.r.o.

**Sypná hmotnost volná** (angl. poured bulk density) má jednotky  $\text{kg/m}^3$ . Materiál se nasype do odměrné nádoby o určitém objemu po okraj. Materiál přesahující horní hranu nádoby se odstraní a vzorek se zváží. Z daného objemu a hmotnosti se zjistí sypná hmotnost volná.

**Sypná hmotnost setřesná** (angl. tapped density) má také jednotky  $\text{kg/m}^3$ . Materiál se nasype do odměrné nádoby, sklepe se (túkáním na dno), pak se materiál opět doplní a zase sklepe. Doplnění a sklepaní materiálu se několikrát zopakuje. Poté se vzorek zváží. Z objemu a z hmotnosti vzorku spočítáme sypnou hmotnost setřesnou. Existují různé metody i přístroje. Pro zjištění, jak hodně se materiál stlačuje, zda při stlačení dělá hrudky atd., často stačí provést měření ručně.

**Sypná hmotnost počáteční z přístroje PFT** (Powder Flow Tester) je plnicí sypná hmotnost a **sypná hmotnost konečná z přístroje PFT** je sypná hmotnost po zkoušce (stlačení).

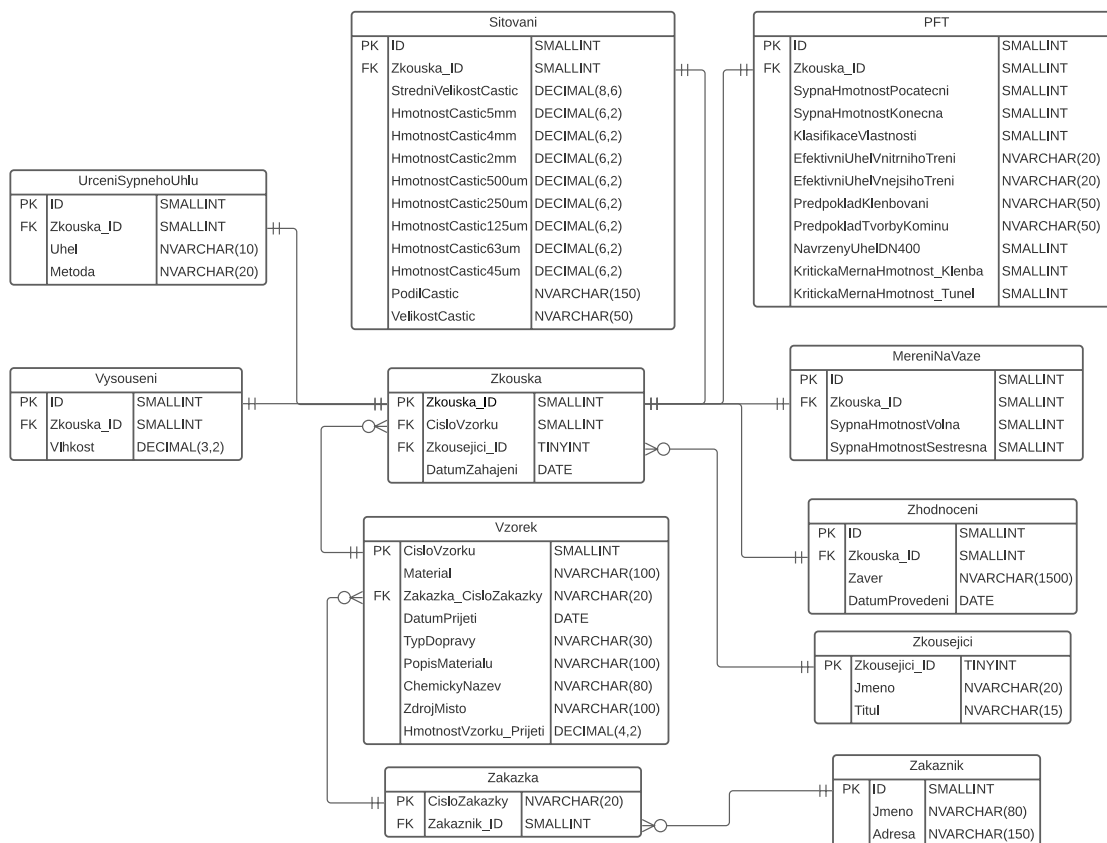
**Sypný úhel** (angl. angle of repose) je sklon hromádky od vodorovné roviny, kterou vytvoří materiál, který např. pomalu vypadává z pomalu se zvedajícího trychtýře nebo nádoby. K měření tohoto parametru existuje mnoho metod. Samotný úhel je měřen úhломěrem.

**Velikost částic** se zjišťuje pomocí síťování. Podnik má k dispozici přístroj se síty s otvory 45  $\mu\text{m}$ , 63  $\mu\text{m}$ , 125  $\mu\text{m}$ , 250  $\mu\text{m}$ , 500  $\mu\text{m}$ , 1 mm, 2 mm, 4 mm a 5 mm. Po síťování se každé síto zváží a údaje se zadají do speciálního programu. Tím se zjistí střední velikost částic, rozsah velikosti částic, křivka zrnitosti atd.

## 4.2 Návrh řešení

### 4.2.1 E-R diagram

Po zanalyzování procesu a dokumentace více jak sto protokolů jsem navrhl entitně-relační diagram znázorněný na nadcházejícím obrázku. Tento E-R diagram se skládá z jedenácti entit. Typ dat je navržený s ohledem na úsporu množství uložených dat (např. místo INT používám SMALLINT) a s ohledem na formát dat, která je potřeba uložit. Tyto datové typy by bylo možné použít, pokud by se použilo např. technologie SQL Server 2019 od Microsoftu. Navržené datové typy se nebudou shodovat s datovými typy použitými v relačním modelu v programu MS Access, jelikož MS Access používá jiné datové typy.



Obrázek 14: Navržený E-R diagram <sup>79</sup>

SQL kód pro vytvoření tabulky Zkouska na SQL Serveru by mohl vypadat takto:

```
CREATE TABLE [Zkouska] (
[Zkouska_ID] SMALLINT,
[CisloVzorku] SMALLINT,
[Zkousejci_ID] TINYINT,
[DatumZahajeni] DATE,
PRIMARY KEY ([Zkouska_ID]),
);
```

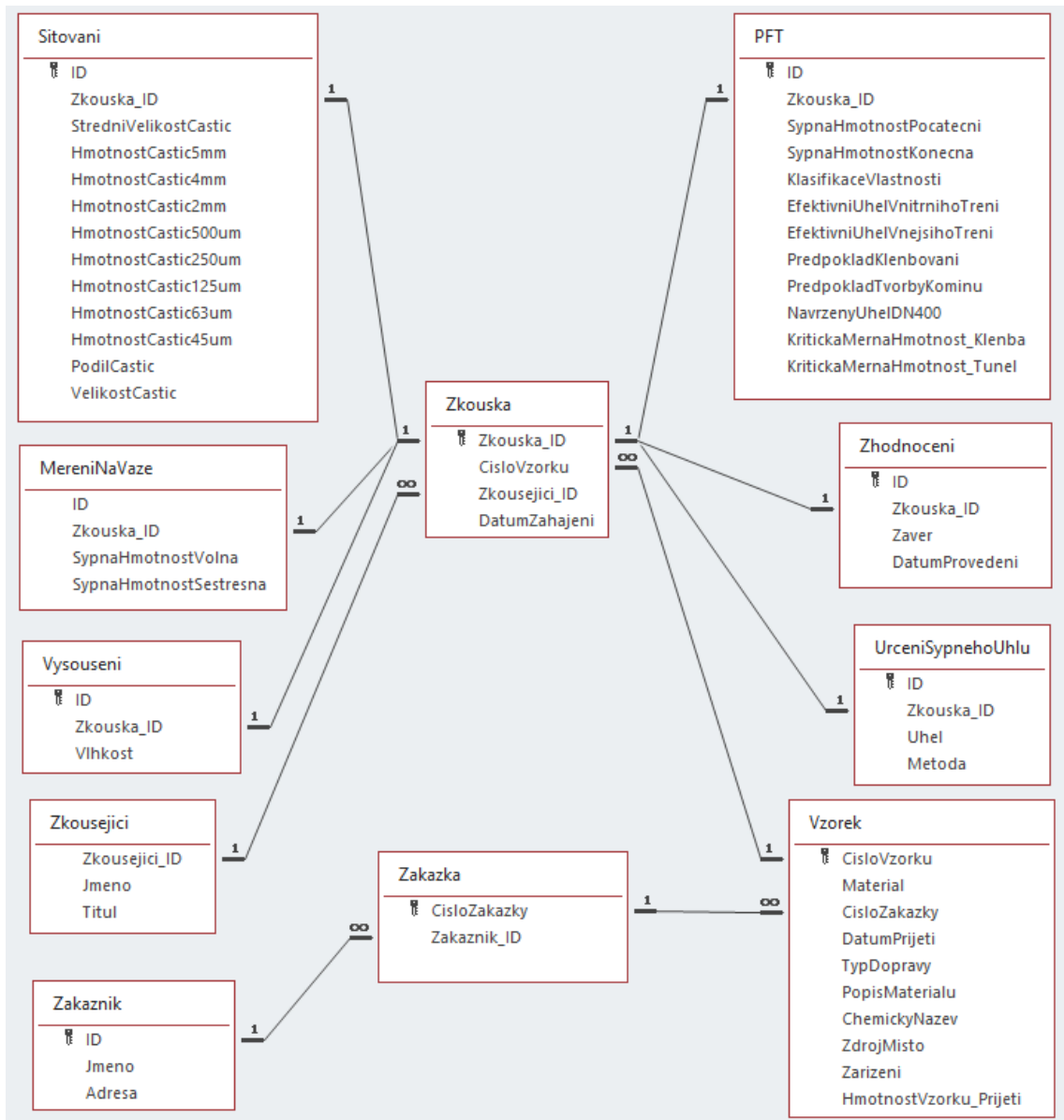
SQL kód pro vytvoření všech tabulek je uveden v Příloze 1.

### 4.3 Implementace řešení

V této části vytvářím relační model v programu MS Access.

<sup>79</sup> Vlastní tvorba

### 4.3.1 E-R Diagram v MS Access



Obrázek 15: Relace v MS Access <sup>80</sup>

V MS Access jsem vytvořil celkem jedenáct tabulek, definoval atributy, určil datové typy a vytvořil relace.

V E-R diagramu je vidět, že mezi entitami **Zakaznik** a **Zakazka** je vztah jedna k více. To znamená, že v databázi můžeme evidovat více zakázek pro jednoho zákazníka, ale jedna zakázka může náležet právě jednomu zákazníkovi.

**Tabulka Zakaznik** obsahuje tři atributy: primární klíč ID (automatické číslo), Jmeno (krátký text

<sup>80</sup> Vlastní tvorba v programu MS Access

s velikostí pole 80) a Adresa (krátký text s velikostí pole 150). Jako primární klíč jsem nastavil atribut ID.

**Tabulka Zakazka** obsahuje dva atributy: primární klíč CisloZakazky (krátký text s velikostí pole 50) a Zakaznik\_ID (číslo).

Mezi entitami **Zakazka a Vzorek je vztah jedna k více**. To znamená, že můžeme pro jednu zakázku evidovat více vzorků a jeden vzorek může náležet právě jedné zakázce.

**Tabulka Vzorek** obsahuje deset atributů: primární klíč CisloVzorku (číslo s velikostí pole celé číslo), Material (krátký text s velikostí pole 200), CisloZakazky (krátký text se stejnou specifikací jako v tabulce Zakazka), DatumPrijeti (datum a čas ve formátu krátké datum), TypDopravy (krátký text s velikostí pole 80), PopisMaterialu (krátký text s velikostí pole 200), ChemickyNazev (krátký text s velikostí pole 255), ZdrojMisto (krátký text s velikostí pole 255), Zarizeni (krátký s velikostí pole 255) a HmotnostVzorku\_Prijeti (desetinné číslo).

Mezi entitami **Vzorek a Zkouska je vztah jedna k více**. Podnik tedy může provést více zkoušek pro jeden vzorek a jedna zkouška se vždy musí vztahovat pouze k jednomu vzorku.

**Tabulka Zkouska** obsahuje čtyři atributy: primární klíč Zkouska\_ID (číslo s velikostí pole celé číslo), CisloVzorku (stejný datový typ jako primární klíč tabulky Vzorek), Zkousejici\_ID (stejný datový typ jako primární klíč tabulky Zkousejici) a DatumZahajeni (datum a čas ve formátu krátkého datumu).

Mezi entitami **Zkousejici a Zkouska je vztah jedna k více**. To znamená, že jeden zkoušející může provést spoustu provést více zkoušek a jedna zkouška může být provedena pouze jedním zkoušejícím.

**Tabulka Zkousejici** obsahuje tři atributy: primární klíč Zkousejici\_ID (číslo s velikostí pole celé číslo), Jmeno (krátký text s velikostí pole 20) a Titul (krátký text s velikostí pole 15).

Poté zde máme celkem **šest entit UrceniSypnehoUhlu, Zhodnoceni, PFT, Sitovani, MereniVaze a Vysouseni**, které mají vztah s **entitou Zkouska jedna k jedné**. Všech těchto šest entit obsahuje atribut Zkouska\_ID, který slouží jako cizí klíč. To znamená, že v rámci jedné zkoušky můžeme uložit pouze jednu údaje ze síťování. Pokud bychom z nějakého důvodu chtěli provést Síťování dvakrát, museli bychom pro druhé síťování evidovat zkoušku s rozdílnou hodnotou atributu Zkouska\_ID.

**Tabulka UrceniSypnehoUhlu** obsahuje čtyři atributy: primární klíč ID (automatické číslo s velikostí pole dlouhé celé číslo), Zkouska\_ID (stejný datový typ jako primární klíč v tabulce

Zkouska), Uhel (krátký text s velikostí pole 40) a Metoda (krátký text s velikostí pole 40).

**Tabulka Zhodnoceni** obsahuje čtyři atributy: primární klíč ID (automatické číslo s velikostí pole dlouhé celé číslo), Zkouska\_ID (stejný datový typ jako primární klíč v tabulce Zkouska), Zaver (dlouhý text) a DatumProvedeni (datum a čas ve formátu krátkého datumu).

**Tabulka PFT** obsahuje dvanáct atributů: primární klíč ID (automatické číslo s velikostí pole dlouhé celé číslo), Zkouska\_ID (stejný datový typ jako primární klíč v tabulce Zkouska), SypnaHmotnostPocatecni (číslo s velikostí pole celé číslo), SypnaHmotnostKonecna (číslo s velikostí pole celé číslo), KlasifikaceVlastnosti (krátký text s velikostí pole 50), EfektivniUhelVnitrihoTreni (krátký text s velikostí pole 20), EfektivniUhelVnejsihoTreni (krátký text s velikostí pole 20), PredpokladKlenbovani (krátký text s velikostí pole 50), PredpokladTvorbyKominu (krátký text s velikostí pole 50), NavrzenyUhelDN400 (krátký text s velikostí pole 100), KritickaMernaHmotnost\_Klenba (číslo s velikostí pole celé číslo) a KrickaMernaHmotnost\_Tunel (číslo s velikostí pole celé číslo).

**Tabulka Sitovani** obsahuje třináct atributů: primární klíč ID (automatické číslo s velikostí pole dlouhé celé číslo), Zkouska\_ID (stejný datový typ jako primární klíč v tabulce Zkouska), StredniVelikostCastic (krátký text s velikostí pole 50), HmotnostCastic5mm (desetinné číslo s přesností 6 a měřítkem 2), HmotnostCastic4mm (desetinné číslo s přesností 6 a měřítkem 2), HmotnostCastic2mm (desetinné číslo s přesností 6 a měřítkem 2), HmotnostCastic500um (desetinné číslo s přesností 6 a měřítkem 2), HmotnostCastic250um (desetinné číslo s přesností 6 a měřítkem 2), HmotnostCastic125um (desetinné číslo s přesností 6 a měřítkem 2), HmotnostCastic63um (desetinné číslo s přesností 6 a měřítkem 2), HmotnostCastic45um (desetinné číslo s přesností 6 a měřítkem 2), PodilCastic (krátký text s velikostí pole 150) a VelikostCastic (krátký text s velikostí pole 50).

**Tabulka MereniNaVaze** obsahuje čtyři atributy: primární klíč ID (automatické číslo s velikostí pole dlouhé celé číslo), Zkouska\_ID (stejný datový typ jako primární klíč v tabulce Zkouska), SypnaHmotnostVolna (krátký text s velikostí pole 50) a SypnaHmotnostSestresna (krátký text s velikostí pole 50).

**Tabulka Vysouseni** obsahuje tři atributy: primární klíč ID (automatické číslo s velikostí pole dlouhé celé číslo), Zkouska\_ID (stejný datový typ jako primární klíč v tabulce Zkouska) a Vlhkost (krátký text s velikostí pole 20).

#### 4.3.2 Import dat z xls souborů do tabulek

Manuální import dat z 132 xls souborů do tabulek by byl poměrně zdlouhavý. Vzhledem

k tomu, že protokoly uložené XLS souborech mají velice podobnou strukturu bylo možno použít následující postup:

1. otevřít Excel,
2. přejít do záložky Data → Načíst data → Ze souboru → Ze složky → Označit složku a otevřít → Kombinovat a načíst → Vybrat ukázkový soubor a kliknout na „OK“.

Tím se vytvořila tabulka v Excelu se záhlavím a s možností filtrování. V pravé části napravo se objevila záložka „Dotazy a připojení“ s jedním dotazem s počtem 6 319 načtených řádků.

1	Zdroj	Název	Column1	Column2	Column3
2	Formulář zkoušek keramický granulát-NA-2013-63.xls		VYHODNOCENÍ ZKOUŠEK VZORKŮ MATERIÁLU		
3	Formulář zkoušek keramický granulát-NA-2013-63.xls				
4	Formulář zkoušek keramický granulát-NA-2013-63.xls				
5	Formulář zkoušek keramický granulát-NA-2013-63.xls	Zákazník / uživatel:		PPC ČAB Nové Sady	
6	Formulář zkoušek keramický granulát-NA-2013-63.xls	Číslo nabídky/zakázky:		NA/2013/63	
7	Formulář zkoušek keramický granulát-NA-2013-63.xls				
8	Formulář zkoušek keramický granulát-NA-2013-63.xls				
9	Formulář zkoušek keramický granulát-NA-2013-63.xls		CHARAKTERISTIKA MATERIÁLU		
10	Formulář zkoušek keramický granulát-NA-2013-63.xls	Materiál:		keramický granulát	
11	Formulář zkoušek keramický granulát-NA-2013-63.xls	Číslo vzorku:		č.v.367	
12	Formulář zkoušek keramický granulát-NA-2013-63.xls	Popis materiálu:		práškový materiál,snadno drobitvý	
13	Formulář zkoušek keramický granulát-NA-2013-63.xls	Sypná hmotnost volná:		997	kg/m3
14	Formulář zkoušek keramický granulát-NA-2013-63.xls	Sypná hmotnost sestřesná (1 dm3)		-	kg/m3
15	Formulář zkoušek keramický granulát-NA-2013-63.xls	Sypná hmotnost počáteční:		980	kg/m3
16	Formulář zkoušek keramický granulát-NA-2013-63.xls	Sypná hmotnost konečná:		1167	kg/m3
17	Formulář zkoušek keramický granulát-NA-2013-63.xls	Sypný úhel: (sklon od vodorovné roviny)		cca 33°	°
18	Formulář zkoušek keramický granulát-NA-2013-63.xls	Velikost částic:		0-0,5	mm
19	Formulář zkoušek keramický granulát-NA-2013-63.xls	Střední velikost částic:		cca 0,18 mm	mm
20	Formulář zkoušek keramický granulát-NA-2013-63.xls	Podíl částic pod 0,045 mm:		15	%
21	Formulář zkoušek keramický granulát-NA-2013-63.xls				

Obrázek 16: Ukázka načtení dat z více xls souborů <sup>81</sup>

Např. entita PFT obsahuje atribut EfektivniUhelVnejsihoTreni, v Column1 jsem si vyfiltroval daný atribut a zkontroloval jsem, že bylo nalezeno 132 záznamů.

Column1	Column2
Efektivní úhel vnitřního tření:	31,5
Efektivní úhel vnějšího tření:	20-25
Efektivní úhel vnitřního tření:	45,2-54,3
Efektivní úhel vnějšího tření:	17-25
Efektivní úhel vnitřního tření:	36,3-33,6
Efektivní úhel vnějšího tření:	24,7-16,4
Efektivní úhel vnitřního tření:	44,4-40,6
Efektivní úhel vnějšího tření:	20,1-16,5
Efektivní úhel vnitřního tření:	47,9-43,6
Efektivní úhel vnějšího tření:	35,7-26,2
Efektivní úhel vnitřního tření:	38,1-36,7
Efektivní úhel vnějšího tření:	27,2-22,5
Efektivní úhel vnitřního tření:	51,9-48,7

Obrázek 17: Získání potřebných dat <sup>82</sup>

<sup>81</sup> Vlastní tvorba v programu Microsoft Excel s reálnými daty podniku

<sup>82</sup> Vlastní tvorba v programu Microsoft Excel s reálnými daty podniku

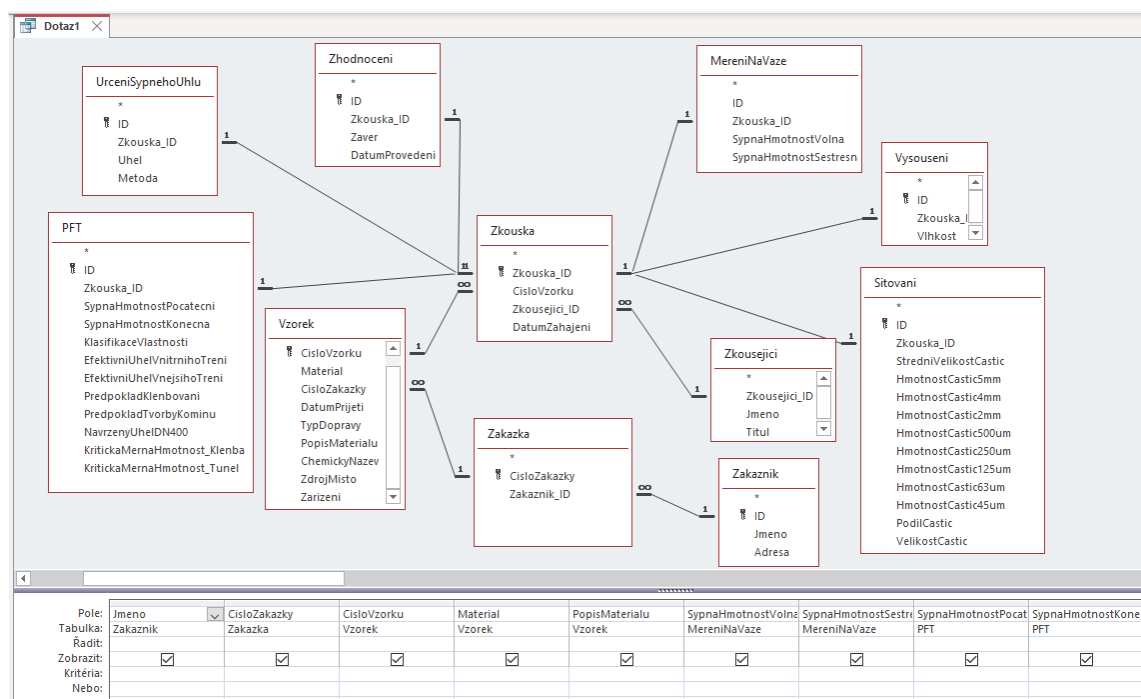
Následně jsem připravil tabulku v Excelu pro import dat do Accessu.

	A	B	C	D	E	F
1	ID	Zkouska_ID	SypnaHmotnostPocatecni	SypnaHmotnostKonecna	KlasifikaceVlastnosti	EfektivniUhelVnitrihoTreni
2	2	1	980	1167	volně tekoucí	31,5
3	3	2	491	857	velmi kohezní	45,2-54,3
4	4	3	1137	1494	snadno tekoucí-volně	36,3-33,6
5	5	4	677	1041	kohezní	44,4-40,6
6	6	5	450	793	velmi kohezní/kohezní	47,9-43,6
7	7	6	1223	1461	snadno tekoucí	38,1-36,7
8	8	7	558	990	velmi kohezní	51,9-48,7
9	9	8	897	1388	velmi kohezní/kohezní	53,8-42,6
10	10	9	595	706	snadno tekoucí/kohezní	36,9-38,9
11	11	10	372	610	velmi kohezní/kohezní	49,4-40,9
12	12	11	241	356	kohezní/snadno teko	39,2-37,6
13	13	12	473	696	velmi kohezní/kohezní	45,7-39,2
14	14	13	222	476	velmi kohezní	51,2-50,5
15	15	14	523	605	volně tekoucí	29,4-28,2
16	16	15	629	712	volně tekoucí	35,5-33,7
17	17	16	537	596	volně tekoucí	29,0-27,6

Obrázek 18: Část tabulky připravené v Excelu pro import do Accessu do tabulky PFT <sup>83</sup>

#### 4.3.3 Vytvoření dotazu pro výsledný protokol – Dotaz1

Pro načtení dat potřebných k vytvoření výsledného protokolu v podobné formě jako je uvedeno na obrázku Obrázek 13 je potřeba vytvořit následující výběrový dotaz.



Obrázek 19: Návrhové zobrazení dotazu pro získání potřebných dat k tvorbě protokolu ze zkoušky <sup>84</sup>

<sup>83</sup> Vlastní tvorba v programu Microsoft Excel s reálnými daty podniku

<sup>84</sup> Vlastní tvorba v programu Microsoft Excel s reálnými daty podniku



V tomto výběrovém dotazu jsem použil parametr. Aplikace se uživatele při použití dotazu zeptá na číslo zkoušky (Zkouska\_ID).

Níže je uveden SQL kód dotazu.

```
SELECT Zakaznik.Jmeno, Zakazka.CisloZakazky, Vzorek.CisloVzorku, Vzorek.Material,  
Vzorek.PopisMaterialu, MereniNaVaze.SypnaHmotnostVolna,  
MereniNaVaze.SypnaHmotnostSestresna, PFT.SypnaHmotnostPocatecni,  
PFT.SypnaHmotnostKonecna, UrceniSypnehoUhlu.Uhel, Sitovani.StredniVelikostCastic,  
Sitovani.PodilCastic, Sitovani.VelikostCastic, Vysouseni.Vlhkost, PFT.KlasifikaceVlastnosti,  
PFT.EfektivniUhelVnitrihoTreni, PFT.EfektivniUhelVnejsihoTreni, PFT.PredpokladKlenbovani,  
PFT.PredpokladTvorbyKominu, PFT.NavrzenyUhelDN400, PFT.KritickaMernaHmotnost_Klenba,  
PFT.KritickaMernaHmotnost_Tunel, Zhodnoceni.Zaver, Zkousejici.Jmeno, Zkousejici.Titul,  
Zhodnoceni.DatumProvedeni, Zkouska.Zkouska_ID
```

```
FROM ((((((Zkousejici INNER JOIN (Zakaznik INNER JOIN (Zakazka INNER JOIN (Vzorek INNER  
JOIN Zkouska ON Vzorek.CisloVzorku = Zkouska.CisloVzorku) ON Zakazka.CisloZakazky =  
Vzorek.CisloZakazky) ON Zakaznik.ID = Zakazka.Zakaznik_ID) ON Zkousejici.Zkousejici_ID =  
Zkouska.Zkousejici_ID) INNER JOIN MereniNaVaze ON Zkouska.Zkouska_ID =  
MereniNaVaze.Zkouska_ID) INNER JOIN PFT ON Zkouska.Zkouska_ID = PFT.Zkouska_ID) INNER  
JOIN Sitovani ON Zkouska.Zkouska_ID = Sitovani.Zkouska_ID) INNER JOIN UrceniSypnehoUhlu  
ON Zkouska.Zkouska_ID = UrceniSypnehoUhlu.Zkouska_ID) INNER JOIN Vysouseni ON  
Zkouska.Zkouska_ID = Vysouseni.Zkouska_ID) INNER JOIN Zhodnoceni ON Zkouska.Zkouska_ID  
= Zhodnoceni.Zkouska_ID
```

```
WHERE (((Zkouska.Zkouska_ID)=[Zadej číslo zkoušky]));
```

Jak je vidět z posledního řádku kódu jedná se o dotaz s parametrem a dotaz zobrazí data ze všech jedenácti entit, které se vyskytují v relačním modelu.

#### 4.3.4 Vytvoření dotazu pro vytištění štítku – Dotaz2

Účelem tohoto dotazu je následná možnost vytisknout štítek, který se lepí na nádobu se vzorkem materiálu, který nebude použit během zkoušky a který slouží k uchování materiálu pro budoucí potřeby podniku. Podoba požadovaného štítku je uvedena na následujícím obrázku.



Obrázek 20: Uchování vzorků pro budoucí potřeby<sup>85</sup>

K vytištění tohoto štítku jsou zapotřebí data ze tří entit. Konkrétně je třeba využít pět atributů: Jmeno z entity Zakaznik, Material z entity Vzorek, CisloVzorku z entity Vzorek, TypDopravy z entity Vzorek a CisloZakazky z entity Zakazka. Jako parametr jsem zvolil atribut CisloVzorku.

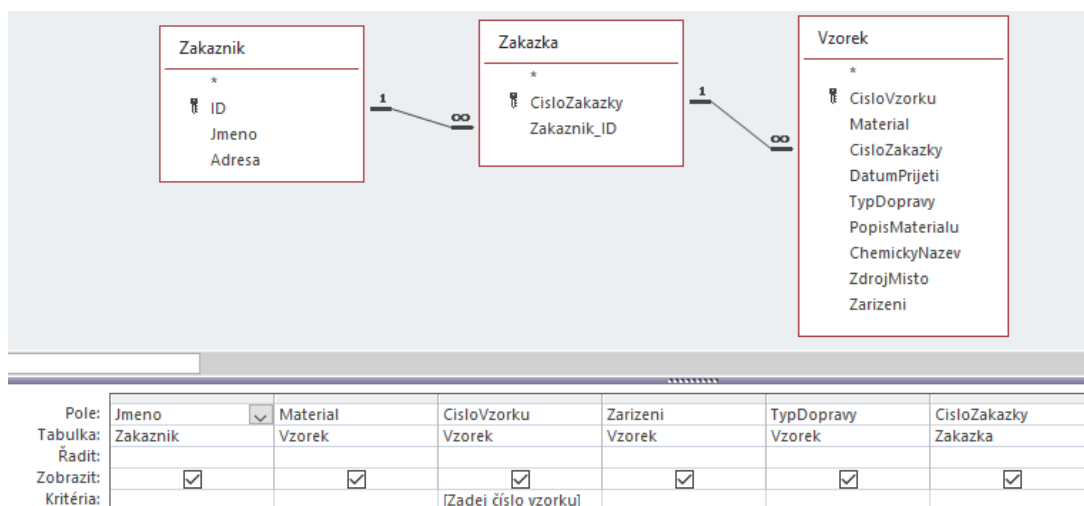
Dotaz se dá vytvořit napsáním následujícího SQL kódu.

**SELECT** Zakaznik.Jmeno, Vzorek.Material, Vzorek.CisloVzorku, Vzorek.Zarizeni,  
Vzorek.TypDopravy, Zakazka.CisloZakazky

**FROM** Zakaznik INNER JOIN (Zakazka INNER JOIN Vzorek ON Zakazka.CisloZakazky =  
Vzorek.CisloZakazky) ON Zakaznik.ID = Zakazka.Zakaznik\_ID

**WHERE** (((Vzorek.CisloVzorku)=[Zadej číslo vzorku]));

Tvorba dotazu přímo v Accessu bez použití SQL kódu je znázorněna na následujícím obrázku.



Obrázek 21: Grafické znázornění Dotazu2 sloužící k zobrazení sestavy pro tisk štítku<sup>86</sup>

<sup>85</sup> Vlastní foto z podniku

<sup>86</sup> Vlastní tvorba v programu MS Access

#### 4.3.5 Vytvoření dotazu pro zobrazení všech údajů o zkoušce – Dotaz3

V aplikaci jsem vytvořit také výběrový dotaz s názvem Dotaz3: Všechny údaje o zkoušce, který načítá všechny záznamy vztažené ke zkoušce. Pomocí tohoto dotazu jsem vytvořil formulář s názvem Formulář\_Dotaz3: Všechny údaje o zkoušce.

Aplikace se při vyvolání tohoto dotazu uživatele zeptá na číslo zkoušky, jejíž údaje chce načíst.

SQL kód tohoto dotazu je následující:

```
SELECT Zkouska.Zkouska_ID, Zkouska.CisloVzorku, Zkouska.Zkousejici_ID,
Zkouska.DatumZahajeni, MereniNaVaze.SypnaHmotnostVolna,
MereniNaVaze.SypnaHmotnostSestresna, Vysouseni.Vlhkost, Sitovani.StredniVelikostCastic,
Sitovani.HmotnostCastic5mm, Sitovani.HmotnostCastic4mm, Sitovani.HmotnostCastic500um,
Sitovani.HmotnostCastic250um, Sitovani.HmotnostCastic125um,
Sitovani.HmotnostCastic63um, Sitovani.HmotnostCastic45um, Sitovani.PodilCastic,
Sitovani.VelikostCastic, PFT.SypnaHmotnostPocatecni, PFT.SypnaHmotnostKonecna,
PFT.KlasifikaceVlastnosti, PFT.EfektivniUhelVnitrihoTreni, PFT.EfektivniUhelVnejsihoTreni,
PFT.PredpokladKlenbovani, PFT.PredpokladTvorbyKominu, PFT.NavrzenyUhelDN400,
PFT.KritickaMernaHmotnost_Klenba, PFT.KritickaMernaHmotnost_Tunel,
UrceniSypnehoUhlu.Uhel, UrceniSypnehoUhlu.Metoda, Zhodnoceni.Zaver,
Zhodnoceni.DatumProvedeni

FROM (((((Zkouska INNER JOIN MereniNaVaze ON Zkouska.Zkouska_ID =
MereniNaVaze.Zkouska_ID) INNER JOIN PFT ON Zkouska.Zkouska_ID = PFT.Zkouska_ID) INNER
JOIN Sitovani ON Zkouska.Zkouska_ID = Sitovani.Zkouska_ID) INNER JOIN UrceniSypnehoUhlu
ON Zkouska.Zkouska_ID = UrceniSypnehoUhlu.Zkouska_ID) INNER JOIN Vysouseni ON
Zkouska.Zkouska_ID = Vysouseni.Zkouska_ID) INNER JOIN Zhodnoceni ON Zkouska.Zkouska_ID
= Zhodnoceni.Zkouska_ID

WHERE (((Zkouska.Zkouska_ID)=[zadej číslo zkoušky]));
```

#### 4.3.6 Vytvoření dotazu pro zobrazení všech údajů o vzorku – Dotaz4

Analogicky jsem vytvořil další výběrový dotaz s názvem Dotaz4: Všechny údaje o vzorku. Pomocí tohoto dotazu jsem vytvořil formulář s názvem Formulář\_Dotaz4: Všechny údaje o vzorku.

Tentokrát aplikace uživatele vyzve, aby zadal číslo vzorku, o kterém chce být informován.

SQL kód tohoto dotazu je následující:

```
SELECT Vzorek.CisloVzorku, Vzorek.Material, Vzorek.CisloZakazky, Vzorek.DatumPrijeti,
Vzorek.TypDopravy, Vzorek.PopisMaterialu, Vzorek.ChemickyNazev, Vzorek.ZdrojMisto,
Vzorek.Zarizeni, Vzorek.HmotnostVzorku_Prijeti, Zakazka.CisloZakazky, Zakazka.Zakaznik_ID,
Zakaznik.ID, Zakaznik.Jmeno, Zakaznik.Adresa

FROM Zakaznik INNER JOIN (Zakazka INNER JOIN Vzorek ON Zakazka.CisloZakazky =
Vzorek.CisloZakazky) ON Zakaznik.ID = Zakazka.Zakaznik_ID

WHERE (((Vzorek.CisloVzorku)=[zadej číslo vzorku]));
```

#### 4.3.7 Vytvoření dalších výběrových dotazů

Pro zobrazení seznamu zkoušek a vzorků jsem vytvořil výběrový dotaz s názvem **Dotaz5: Seznam zkoušek a vzorků**. Tímto dotazem může zkoušející vyvolat sestavu, díky které zjistí jaké je číslo poslední zkoušky před zahájením nové zkoušky.

SQL kód dotazu je následující:

```
SELECT Zkouska.Zkouska_ID AS [Číslo zkoušky], Vzorek.CisloVzorku AS [Číslo vzorku],
Zakazka.CisloZakazky AS [Číslo zakázky], Zakaznik.Jmeno AS [Jméno zákazníka]

FROM Zakaznik INNER JOIN (Zakazka INNER JOIN (Vzorek INNER JOIN Zkouska ON
Vzorek.CisloVzorku = Zkouska.CisloVzorku) ON Zakazka.CisloZakazky = Vzorek.CisloZakazky) ON
Zakaznik.ID = Zakazka.Zakaznik_ID

ORDER BY Zkouska.Zkouska_ID;
```

Posledním výběrový dotaz, který jsem vytvořil, se jmenuje **Dotaz6: Zákazníci**. Pomocí tohoto dotazu jsem poté vytvořil sestavu, díky které si zkoušející může zobrazit seznam zákazníků.

SQL kód dotazu Dotaz6: Zákazníci je následující:

```
SELECT Zakaznik.ID AS [ID zákazníka], Zakaznik.Jmeno AS Jméno, Zakaznik.Adresa, *

FROM Zakaznik;
```

#### 4.3.8 Vytvoření přidávacích dotazů

Za účelem přidání údajů do databáze jsem vytvořil celkem deset přidávacích dotazů. Tyto údaje poté využiji při návrhu formulářů, kde využiju tlačítka a pomocí tvůrce maker vyvolám tyto přidávací dotazy při kliknutí na tlačítko.

**Přídávací dotaz „1-Přidání zákazníka“** má následující SQL kód:

```
INSERT INTO Zakaznik ( Jmeno, Adresa )
SELECT [Zadej jméno zákazníka] AS Výraz1, [Zadej adresu zákazníka] AS Výraz2;
```

Při vyvolání dotazu se tedy aplikace uživatele zeptá prvně na jméno zákazníka a poté na adresu zákazníka. Tímto se do tabulky Zakaznik přidá záznam , kde hodnota ID se vytvoří automaticky. Do atributů Jmeno a Adresa se přidají hodnoty, které uživatel zadal.

**Přídávací dotaz „2-Vytvoření zakázky“** má následující SQL kód:

```
INSERT INTO Zakazka ( CisloZakazky, Zakaznik_ID )
SELECT [Zadej číslo zakázky] AS Výraz1, [Zadej ID zákazníka] AS Výraz2;
```

Při vyvolání dotazu se aplikace uživatele zeptá na číslo zakázky a ID zákazníka pro kterého bude zakázka vedena a dojde k uložení záznamu do atributů CisloZakazky a Zakaznik\_ID.

**Přídávací dotaz „3-Přidání vzorku“** má následující SQL kód:

```
INSERT INTO Vzorek ( CisloVzorku, CisloZakazky, Material, DatumPrijeti, TypDopravy,
PopisMaterialu, ChemickyNazev, ZdrojMisto, Zarizeni )
```

```
SELECT [Zadej číslo vzorku] AS Výraz1, [Zadej číslo zakázky] AS Výraz2, [Zadej materiál] AS
Výraz3, [Zadej datum přijetí vzorku] AS Výraz4, [Zadej typ dopravy] AS Výraz5, [Zadej popis
materiálu] AS Výraz6, [Zadej chemický název materiálu] AS Výraz7, [Zadej zdroj vzorku] AS
Výraz8, [Zadej zařízení, které manipuluje se vzorkem] AS Výraz9;
```

Aplikace se uživatele zeptá na číslo vzorku, číslo zakázky, materiál, datum přijetí vzorku, typ dopravy, popis materiálu, chemický název materiálu, zdroj vzorku a zařízení. Zadané hodnoty se uloží do záznamu v tabulce Vzorek.

**Přídávací dotaz „4-Zahájení zkoušky“** má následující SQL kód:

```
INSERT INTO Zkouska ( Zkouska_ID, CisloVzorku, Zkousejici_ID, DatumZahajeni )
SELECT [Zadej číslo zkoušky] AS Výraz1, [Zadej číslo vzorku] AS Výraz2, [Zadej ID zkoušejícího]
AS Výraz3, Date() AS Výraz4;
```

Aplikace se uživatele zeptá na číslo zkoušky, číslo vzorku a ID zkoušejícího. Zadané hodnoty se uloží do záznamu v tabulce Zkouska. Do atributu DatumZahajeni se automaticky uloží aktuální datum.

**Přídávací dotaz „5-měření na váze“** má následující SQL kód:

```
INSERT INTO MereniNaVaze ( Zkouska_ID, SypnaHmotnostVolna, SypnaHmotnostSestresna )
```

```
SELECT [Zadej číslo zkoušky] AS Výraz1, [Zadej výsledek - Sypná hmotnost volná (kg/m3)] AS  
Výraz2, [Zadej Výsledek - Sypná hmotnost setřesná (kg/m3)] AS Výraz3;
```

Aplikace se uživatele zeptá na číslo zkoušky, sypnou hmotnost volnou a sypnou hmotnost setřesnou. Zadané hodnoty se uloží do záznamu v tabulce MereniNaVaze.

**Přídávací dotaz „6-Určení vlhkosti vysoušením“** má následující SQL kód:

```
INSERT INTO Vysouseni ( Zkouska_ID, Vlhkost )
```

```
SELECT [Číslo zkoušky] AS Výraz1, [Zadej výslednou vlhkost (%)] AS Výraz2;
```

Aplikace se uživatele zeptá na číslo zkoušky a výslednou vlhkost. Zadané hodnoty se uloží do záznamu v tabulce MereniNaVaze.

**Přídávací dotaz „7-Sítování“** má následující SQL kód:

```
INSERT INTO Sitovani ( Zkouska_ID, StredniVelikostCastic, HmotnostCastic5mm,  
HmotnostCastic4mm, HmotnostCastic2mm, HmotnostCastic500um, HmotnostCastic250um,  
HmotnostCastic125um, HmotnostCastic63um, HmotnostCastic45um, PodilCastic,  
VelikostCastic )
```

```
SELECT [Zadej číslo zkoušky] AS Výraz1, [Zadej střední velikost částic (mm) ] AS Výraz2, [Zadej  
hmotnost částic nad 5mm (kg) ] AS Výraz3, [Zadej hmotnost částic nad 4mm (kg) ] AS Výraz4,  
[Zadej hmotnost částic nad 2mm (kg) ] AS Výraz5, [Zadej hmotnost částic nad 500um (kg) ] AS  
Výraz6, [Zadej hmotnost částic nad 250um (kg) ] AS Výraz7, [Zadej hmotnost částic nad 125um  
(kg) ] AS Výraz8, [Zadej hmotnost částic nad 63 um (kg) ] AS Výraz9, [Zadej hmotnost částic nad  
45 um (kg) ] AS Výraz10, [Zadej vybraný podíl částic nad/pod vybraný rozměr (dle potřeby) ] AS  
Výraz11, [Zadej rozmezí velikosti částic (mm) ] AS Výraz12;
```

Při spuštění toho dotazu se aplikace zeptá celkem na dvanáct hodnot. Tyto hodnoty se uloží do záznamu v tabulce Sitovani.

**Přídávací dotaz „8-Měření na PFT“** má následující SQL kód:

```
INSERT INTO PFT ( Zkouska_ID, SypnaHmotnostPocatecni, SypnaHmotnostKonecna,  
KlasifikaceVlastnosti, EfektivniUhelVnitrihoTreni, EfektivniUhelVnejsihoTreni,  
PredpokladKlenbovani, PredpokladTvorbyKominu, NavrzenyUhelDN400,  
KritickaMernaHmotnost_Klenba, KritickaMernaHmotnost_Tunel )
```

SELECT [Zadej číslo zkoušky] AS Výraz1, [Zadej sypanou hmotnost počáteční (kg/m<sup>3</sup>) ] AS Výraz2, [Zadej sypanou hmotnost konečnou (kg/m<sup>3</sup>) ] AS Výraz3, [Jaká je klasifikace vlastností materiálů] AS Výraz4, [Jaký je efektivní úhel vnitřního tření? (°) ] AS Výraz5, [Jaký je efektivní úhel vnějšího tření? (°) ] AS Výraz6, [Existuje předpoklad klenbování?] AS Výraz7, [Existuje předpoklad tvorby komínu?] AS Výraz8, [Jaký je navržený úhel kuželu DN400 (°) ] AS Výraz9, [Jaká je kritická měrná hmotnost pro tvorbu klenby? (kg/m<sup>3</sup>) ] AS Výraz10, [Jaká je kritická měrná hmotnost pro tvorbu tunelu? (kg/m<sup>3</sup>) ] AS Výraz11;

Při spuštění dotazu je uživatel dotázán na jedenáct hodnot. Tyto hodnoty jsou následně uloženy do záznamu v tabulce PFT.

**Přidávací dotaz „9-Určení sypaného úhlu“** má následující SQL kód:

```
INSERT INTO UrceniSypnehoUhlu ( Zkouska_ID, Uhel, Metoda )
```

```
SELECT [Zadej číslo zkoušky] AS Výraz1, [Jaký je výsledný sypaný úhel? (°) ] AS Výraz2, [Jaká byla použita metoda?] AS Výraz3;
```











Aplikace se uživatele zeptá na číslo zkoušky, výsledný sypaný úhel a použitou metodu měření. Zadané hodnoty uživatelem jsou uloženy do záznamu v tabulce UrceniSypnehoUhlu.

**Přidávací dotaz „10-Vyhodnocení zkoušky“** má následující SQL kód:

```
INSERT INTO Zhodnoceni ( Zkouska_ID, Zaver, DatumProvedeni )
```

```
SELECT [Zadej číslo zkoušky] AS Výraz3, [Napiš závěr zkoušky] AS Výraz2, Date() AS Výraz1;
```

Aplikace se uživatele zeptá na číslo zkoušku a závěr. Zadané hodnoty jsou uloženy do záznamu v tabulce Zhodnoceni. Do atributu DatumProvedeni je uloženo aktuální datum.

-  10-Vyhodnocení zkoušky
-  1-Přidání zákazníka
-  2-Vytvoření zakázky
-  3-Přidání vzorku
-  4-Zahájení zkoušky
-  5-Měření na váze
-  6-Určení vlhkosti vysoušením
-  7-Sítování
-  8-Měření na PFT
-  9-Určení sypaného úhlu

Obrázek 22: Přidávací dotazy v MS Access <sup>87</sup>

---

<sup>87</sup> Vlastní tvorba v programu MS Access

#### 4.3.9 Vytvoření odstraňovacích dotazů

**Odstraňovací dotaz „OdstranitZakaznika“** má následující SQL kód:

```
DELETE Zakaznik.ID  
  
FROM Zakaznik  
  
WHERE (((Zakaznik.ID)=[ID zákazníka]));
```

Tento dotaz slouží k odstranění záznamu z tabulky Zakaznik. Aplikace se uživatele zeptá na ID zákazníka a dojde k jeho odstranění.

**Odstraňovací dotaz „OdstranitZakazku“** má následující SQL kód:

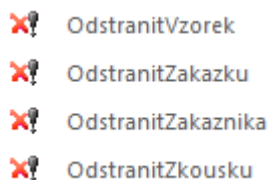
```
DELETE Zakazka.CisloZakazky  
  
FROM Zakazka  
  
WHERE (((Zakazka.CisloZakazky)=[Zadej číslo zakázky k odstranění]));
```

Tento dotaz slouží k odstranění záznamu z tabulky Zakazka. Uživatel je dotázán na číslo zakázky a poté dojde k odstranění souvisejícího záznamu.

**Odstraňovací dotaz „OdstranitVzorek“** má následující SQL kód:

```
DELETE Vzorek.CisloVzorku  
  
FROM Vzorek  
  
WHERE (((Vzorek.CisloVzorku)=[Zadejte číslo vzorku k odstranění]));
```

Tento dotaz slouží k odstranění záznamu v tabulce Vzorek. Aplikace se uživatele zeptá na číslo vzorku a dojde k jeho odstranění.



Obrázek 23: Odstraňovací dotazy v MS Access <sup>88</sup>

---

<sup>88</sup> Vlastní tvorba v programu MS Access



#### 4.3.10 Vytvoření formuláře – Úvodní stránka

Na následujícím obrázku je znázorněn formulář s názvem Úvodní strana s pěti tlačítky.

Formulář obsahuje tlačítka:

- Přejít na evidenci vzorků: po kliknutí se otevře formulář Evidence vzorků a zavře se formulář Úvodní strana,
- Přejít na zkoušení: po kliknutí se otevře formulář Provádění zkoušky a zavře se formulář Úvodní strana,
- Otevřít protokol ze zkoušky: po kliknutí se otevře sestava ProtokolZeZkoušky,
- Vytisknou štítek na nádobku vzorku: po kliknutí se otevře sestava Štítky pro Dotaz2,
- Ukončit program: po kliknutí se zavře program MS Access.



Obrázek 24: Formulář - Úvodní strana <sup>89</sup>

#### 4.3.11 Vytvoření formuláře – Evidence vzorků

Na následujícím obrázku je znázorněn formulář s názvem Evidence vzorků s devíti tlačítky.

Tento formulář obsahuje tlačítka:

- Přidat nového zákazníka: spustí se přidávací dotaz 1-Přidání zákazníka,
- Vytvořit novou zakázku: spustí se přidávací dotaz 2-Vytvoření zakázky,
- Zaevidovat nový vzorek: spustí se přidávací dotaz 3-Přidání vzorku,

---

<sup>89</sup> Vlastní tvorba, logo z interní dokumentace podniku

- Upravit záznamy vzorku: otevře se formulář Formulář\_Dotaz4\_Všechny údaje o vzorku,
- Vrátit se na úvodní stránku: otevře se formulář Úvodní stránka a zavře se formulář Evidence vzorků,
- Vytisknout štítek na nádobku vzorku: otevře se sestava Štítky pro Dotaz2,
- Zobrazit zákazníky: otevře se sestava Dotaz6: Zákazníci a zavře se formulář Evidence vzorků,
- Zobrazit seznam zkoušek a vzorků: otevře se sestava Seznam zkoušek a vzorků a zavře se formulář Evidence vzorků,
- Odstranit zákazníka: spustí se odstraňovací dotaz OdstranitZakaznika,
- Odstranit zakázku: spustí se odstraňovací dotaz OdstranitZakazku,
- Odstranit vzorek: spustí se odstraňovací dotaz OdstranitVzorek.



Obrázek 25: Formulář - Evidence vzorků<sup>90</sup>

#### 4.3.12 Vytvoření formuláře - Provádění zkoušky

Na následujícím obrázku je zobrazen formulář Provádění zkoušky.

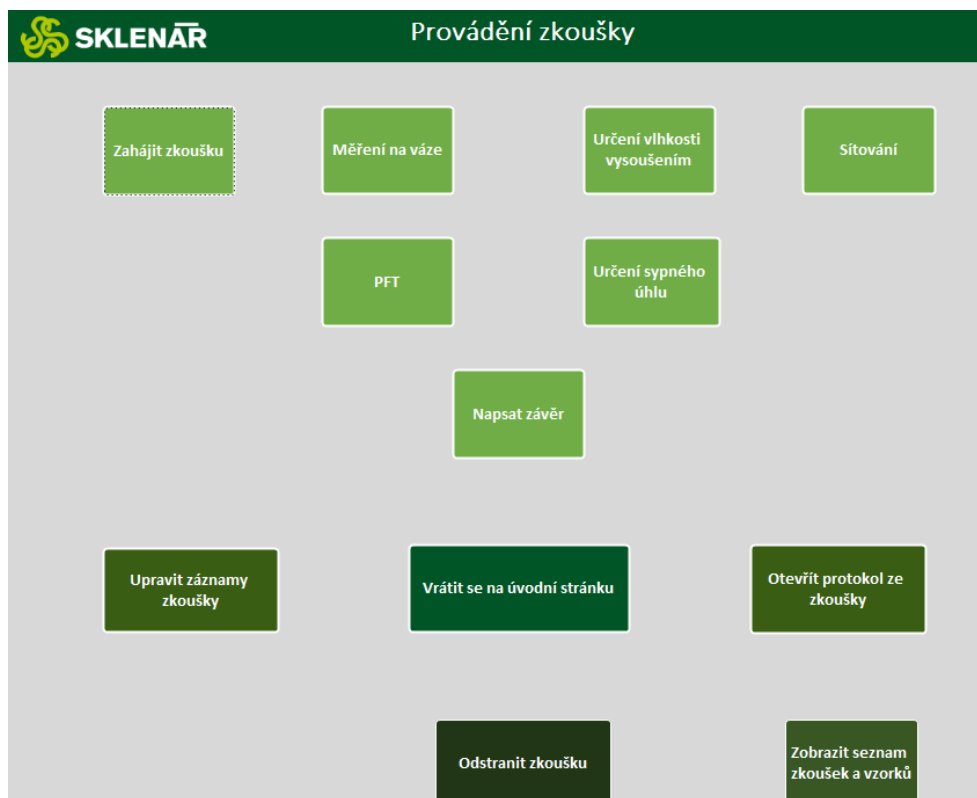
Tento formulář se obsahuje jedenáct tlačítek:

- Zahájit zkoušku: spustí se přidávací dotaz 4-Zahájení zkoušky,

---

<sup>90</sup> Vlastní tvorba, logo z interní dokumentace podniku

- Měření na váze: spustí se přidávací dotaz 5-Měření na váze
- Určení vlhkosti vysoušením: spustí se přidávací dotaz 6 – Určení vlhkosti vysoušením,
- Sítování: spustí se přidávací dotaz 7–Sítování,
- PFT: spustí přidávací dotaz 8-Měření na PFT,
- Určení sypného úhlu: spustí přidávací dotaz 9-Určení sypného úhlu,
- Napsat závěr: spustí přidávací dotaz 10-Vyhodnocení zkoušky,
- Upravit záznamy zkoušky: otevře Formulář\_Dotaz3: Všechny údaje o zkoušce a zavře formulář Provádění zkoušky,
- Vrátit se na úvodní stránku: otevře formulář Úvodní stránka a zavře formulář Provádění zkoušky,
- Otevřít protokol ze zkoušky: otevře sestavu ProtokolZeZkoušky a zavře formulář Provádění zkoušky,
- Zobrazit seznam zkoušek a vzorků: otevře se sestava Seznam zkoušek a vzorků a zavře se formulář Provádění zkoušky,
- Odstranit zkoušku: spustí odstraňovací dotaz OdstranitZkoušku.



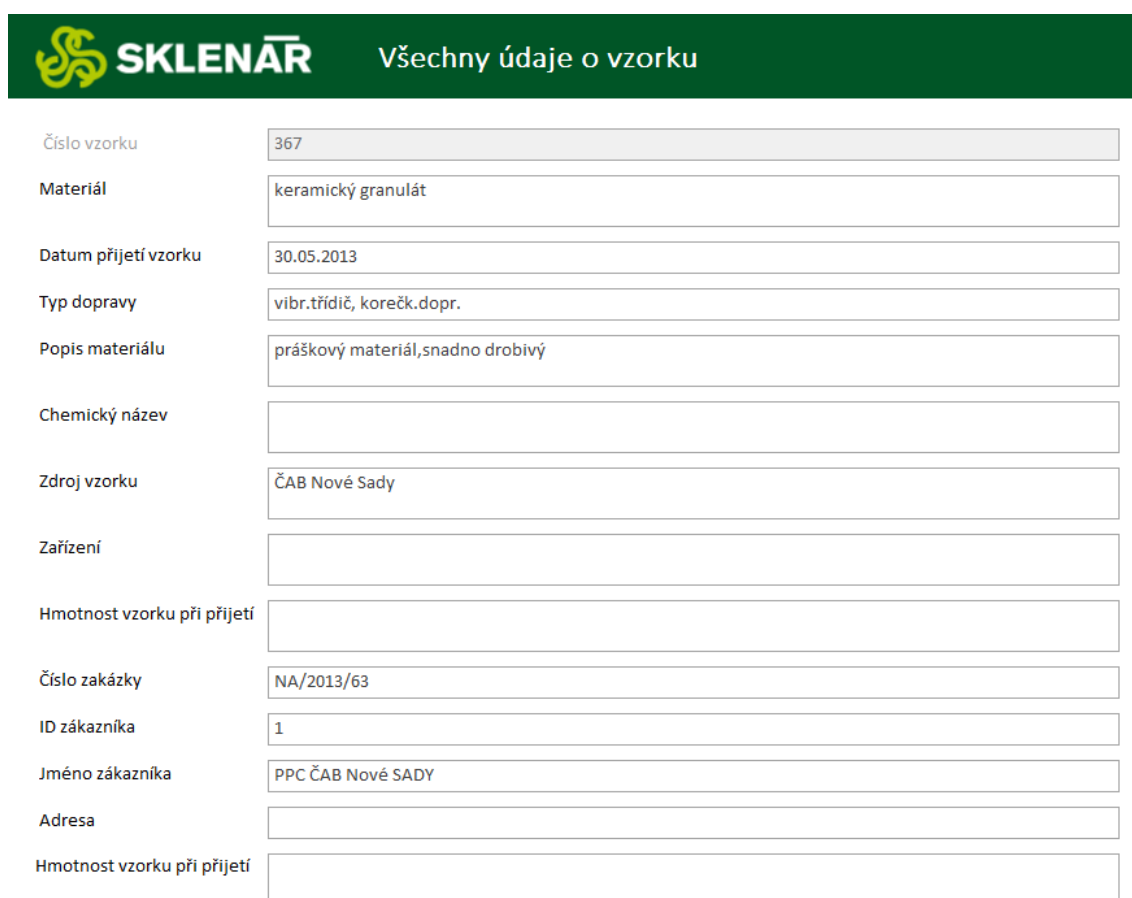
Obrázek 26: Formulář - Provádění zkoušky<sup>91</sup>

<sup>91</sup> Vlastní tvorba, logo z interní dokumentace podniku

#### 4.3.13 Vytvoření formuláře – Všechny údaje o vzorku

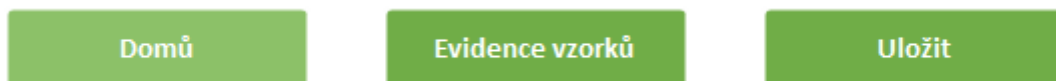
Dále jsem vytvořil formulář s názvem Formulář\_Dotaz4: Všechny údaje o vzorku, pomocí kterého je možné zobrazit všechny údaje již zaevidovaného vzorku.

V zápatí formuláře se vyskytují tři tlačítka. Tlačítko Domů slouží k otevření formuláře Úvodní stránka a k zavření aktuálního formuláře. Tlačítko Evidence vzorků slouží k návratu do formuláře Evidence vzorků a k zavření aktuálního formuláře a tlačítko Uložit slouží k aktualizaci záznamu, čím se také propíše změny do tabulek, ve kterých jsou hodnoty uloženy.



Číslo vzorku	367
Materiál	keramický granulát
Datum přijetí vzorku	30.05.2013
Typ dopravy	vibr.třídíč, korečk.dopr.
Popis materiálu	práškový materiál, snadno drobivý
Chemický název	
Zdroj vzorku	ČAB Nové Sady
Zařízení	
Hmotnost vzorku při přijetí	
Číslo zakázky	NA/2013/63
ID zákazníka	1
Jméno zákazníka	PPC ČAB Nové SADY
Adresa	
Hmotnost vzorku při přijetí	

Obrázek 27: Formulář - Všechny údaje o vzorku <sup>92</sup>



Obrázek 28: Záhlaví formuláře Všechny údaje o vzorku <sup>93</sup>

<sup>92</sup> Vlastní tvorba, logo z interní dokumentace podniku

<sup>93</sup> Vlastní tvorba

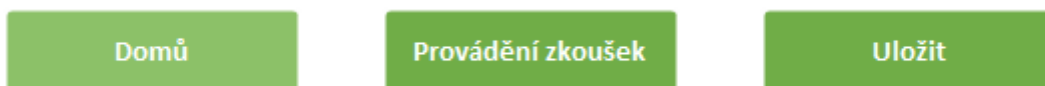
#### 4.3.14 Vytvoření formuláře – Všechny údaje o zkoušce

Na následujícím obrázku lze vidět formulář s názvem Formulář\_Dotaz4'3: Všechny údaje zkoušce. Tento formulář může sloužit k zobrazení všech dat již uskutečněné zkoušky nebo ke změně zobrazených hodnot.

V zápatí formuláře se vyskytují tři tlačítka. Tlačítko Domů slouží k otevření formuláře Úvodní stránka a k zavření aktuálního formuláře. Tlačítko Provádění zkoušek slouží k návratu do formuláře Provádění zkoušky a k zavření aktuálního formuláře a tlačítko Uložit slouží k aktualizaci záznamu, čím se také propíší změny do tabulek, ve kterých jsou hodnoty uloženy.

SKLENĀR		Všechny údaje o zkoušce	
Číslo zkoušky	132	Efektivní úhel vnitřního tření [°]	36,5-34,7
Číslo vzorku	672	Efektivní úhel vnějšího tření [°]	23,8-19,4
ID zkoušejícího	1	Předpoklad klenbování	NE
Datum zahájení zkoušky		Předpoklad tvorby komínu	ANO
Sypná hmotnost volná [kg/m <sup>3</sup> ]	1470	Navržený úhel DN400 [°]	58
Sypná hmotnost setřesná [kg/m <sup>3</sup> ]	-	Kritická měrná hmotnost pro tvorbu klenby [kg/m <sup>3</sup> ]	1412
Vlhkost [%]	0,10	Kritická měrná hmotnost pro tvorbu tunelu [kg/m <sup>3</sup> ]	1761
Střední velikost částic [mm]	0,121	Sypný úhel [°]	-
Hmotnost částic nad 5 mm [g]		Metoda	
Hmotnost částic nad 4 mm [g]			
Hmotnost částic nad 500 um [g]		Datum provedení zkoušky	11.05.2021
Hmotnost částic nad 250 um [g]		Závěr	Velikost částic změněna-rozpětí 0-1 mm. Střední velikost částic 0,121 mm. 16,1% částic je menších než 45um. Vlhkost materiálu 0,10 %.
Hmotnost částic nad 125 um [g]			
Hmotnost částic nad 63 um [g]			
Hmotnost částic nad 45 um [g]			
Podíl částic	16,1% pod 1mm		
Velikost částic [mm]	0-1 mm		
Sypná hmotnost počáteční [kg/m <sup>3</sup> ]	1405		
Sypná hmotnost konečná [kg/m <sup>3</sup> ]	1659		
Klasifikace vlastností	snadno/volně tekoucí		

Obrázek 29: Formulář - Všechny údaje o zkoušce <sup>94</sup>



Obrázek 30: Záhlaví formuláře Všechny údaje o zkoušce <sup>95</sup>

<sup>94</sup> Vlastní tvorba, logo z interní dokumentace podniku

<sup>95</sup> Vlastní tvorba

#### 4.3.15 Sestava výsledného protokolu ze zkoušky

Pomocí předešlého dotazu „Dotaz1“ jsem vytvořil sestavu s názvem „ProtokolZeZkousky“. Při spuštění této sestavy se aplikace uživatele zeptá na číslo zkoušky a po zadání čísla zkoušky vytvoří pohled s již daným formátováním s požadovanými daty.

Do zápatí sestavy jsem umístil čtyři tlačítka: tlačítko „Poslat PDF e-mailem“ s funkcí odeslání e-mailu s přílohou, tlačítko „Uložit PDF“ s funkcí uložení protokolu ve formátu .pdf na místní disk, tlačítko „Vytisknout“ s funkcí vytisknout protokol a tlačítko „Domů“, které otevře formulář Úvodní stránka a zavře sestavu ProtokolZeZkousky.

Na následujícím obrázku je znázorněn protokol s reálnými daty z databáze, který byl exportován do formátu .pdf. Na tomto obrázku nejsou schválně vidět tlačítka, čehož jsem docílil tím, že jsem je umístil do zápatí sestavy.



Obrázek 31: Záhloví sestavy ProtokolZeZkousky<sup>96</sup>

Záhloví této sestavy obsahuje čtyři tlačítka:

- Poslat PDF e-mailem: po kliknutí se otevře e-mailový klient s již přiloženou přílohou,
- Uložit PDF: po kliknutí se uloží PDF na místní disk,
- Vytisknout: po kliknutí se vytiskne sestava,
- Domů: po kliknutí se otevře formulář Úvodní stránka a zavře se sestava ProtokolZeZkousky.

---

<sup>96</sup> Vlastní tvorba

Číslo zkoušky	132	Datum provedení zkoušky	11.05.2021
Jméno zákazníka	Johns Manville	Číslo zakázky	20/2021/16
Číslo vzorku	672	Materiál	recyklát
Popis materiálu	velmi jemný bílý materiál (vzorek po zkouškách linka)		
Sypná hmotnost volná [kg/m <sup>3</sup> ]	1470	Sypný úhel [°]	-
Sypná hmotnost setřesná [kg/m <sup>3</sup> ]	-	Vlhkost [%]	0,10
Střední velikost částic	0,121	Podíl Částic	16,1% pod 1mm
Velikost Částic	0-1 mm		

Sypná hmotnost počáteční [kg/m <sup>3</sup> ]	1405
Sypná hmotnost konečná [kg/m <sup>3</sup> ]	1659
Klasifikace vlastností materiálu	snadno/volně tekoucí
Efektivní úhel vnitřního tření [°]	36,5-34,7
Efektivní úhel vnějšího tření [°]	23,8-19,4
Navržený úhel DN400 [°]	58
Předpoklad klenbování	NE
Předpoklad tvorby kominu	ANO
Kritická měrná hmotnost pro tvoření klenby [kg/m <sup>3</sup> ]	1412
Kritická měrná hmotnost pro vznik středního tunelu [kg/m <sup>3</sup> ]	1761

Závěr	Velikost částic změřena-rozpětí 0-1 mm. Střední velikost částic 0,121 mm. 16,1% částic je menších než 45um. Vlhkost materiálu 0,10 %.
Zkoušející	Ing. Miroslav Keprta

 Obrázek 32: Výsledná sestava protokolu ze zkoušky exportovaná do formátu PDF <sup>97</sup>

#### 4.3.16 Vytvoření sestavy za účelem tisku štítků – sestava Štítky pro Dotaz2

Při spuštění sestavy se aplikace uživatele zeptá na číslo vzorku následně vygeneruje pohled.

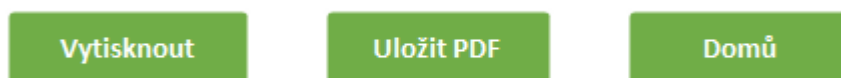
<sup>97</sup> Vlastní tvorba z reálných podnikových dat

PPC ČAB Nové SADY  
keramický granulát  
Číslo vzorku: 367  
vibr.třídíč, korečk.dopr. NA/2013/63

Obrázek 33: Vygenerovaný štítek pro tisk sestavou <sup>98</sup>

Záhlaví této sestavy jsou tři tlačítka:

- Vytisknout: po kliknutí se vytiskne sestava,
- Uložit PDF: po kliknutí se uloží PDF na místní disk,
- Domů: po kliknutí se otevře formulář Úvodní stránka a zavře se sestava ProtokolZeZkousky.



Obrázek 34: Záhlaví sestavy Štítky pro Dotaz<sup>99</sup>

## 5 Zhodnocení přínosů návrhu

Navrhnutá relační databáze umožňuje podniku uchovávat a sbírat data ze zkoušek sypkých materiálů. V době psaní tohoto textu má soubor .accdb asi 4 MB, což je značná úspora oproti původním asi 47 MB, které zabírá 132 .xls souborů.

Hlavní přínos vytvořené aplikace vidím v intuitivním zadávání dat do databáze. Dále užíváním navrhnuté aplikace očekávám značné snížení rizika zadání špatných údajů, které mohlo vznikat např. při kopírování excelovských souborů a přepisování dat starších zkoušek.

Aplikace nedovolí, že by zkoušející zapomněl zadat nějaký údaj, jelikož uživatel aplikace je vždy vyzván k vyplnění všech údajů. Zkoušejícímu je umožněno nevyplnit údaje na základě jeho úsudku, když např. uzná, že není nutné dané měření provést.

Dalším přínosem je jednoduchý a rychlý přístup k datům, což může znamenat časové úspory např. při hledání protokolu ze zkoušek, které může uživatel aplikace snadno zobrazit,

---

<sup>98</sup> Vlastní tvorba s reálnými podnikovými daty

<sup>99</sup> Vlastní tvorba



vytisknout, poslat e-mailem nebo uložit ve formátu .pdf.

Tato aplikace může sloužit jako prototyp pro budoucí vytvoření informačního systému, což může do budoucna značně snížit finanční náklady a vést k časové úspoře doby implementace.

## 6 Závěr

V teoretické části práce jsem se věnoval teorii, jejíž znalost byla nutná pro vytvoření dané relační databáze. V praktické části jsem pro podnik Sklenář s.r.o. vytvořil relační databázi v programu MS Access, která slouží k uchování a sběru dat ze zkoušek sypkých materiálů.

Vytvořená aplikace v MS Accessu splňuje všechny potřebné požadavky pro sběr a úschovu dat ze zkoušek, což je jejím hlavním účelem.

Aplikace byla vytvořena s ohledem na průběh procesu zkoušení sypkých materiálů. Zároveň umožňuje zkoušejícím změnit pořadí po sobě jdoucích procesů.

Podniku bych navrhnul danou aplikací nahradit dosavadní používání excelovských tabulek a po provedení pár zkoušek zhodnotit, jestli je třeba aplikaci přidat nějaké funkce. Po otestování bych zvážil, jestli by nebylo vhodné tuto databázi napojit na již existující informační systém, který eviduje např. zákazníky a zakázky za účelem vytváření faktur.

V praktické části jsem také popsal metodiku převodu excelovských tabulek do relační databáze. Tato metodika může posloužit spoustu podnikům při převodu dat, která se nacházejí ve více excelovských souborech do jedné databáze.

## 7 Zdroje

### 7.1 Knižní zdroje

BÍLA, Jiří, František KRÁL a Vladimír HLAVÁČ. Informační technologie: databázové a znalostní systémy. Vyd. 2. přeprac. Praha: Vydavatelství ČVUT, 2003. ISBN 80-01-02790-2.

KROENKE, David a David J. AUER. Databáze. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.

POKORNÝ, Jaroslav a Michal VALENTA. Databázové systémy. 2. přepracované vydání. Praha: Česká technika - nakladatelství ČVUT, 2020. ISBN 978-80-01-06696-6.

RIORDAN, Rebecca M. *Vytváříme relační databázové aplikace*. Praha: Computer Press, 2000. Databáze. ISBN 80-7226-360-9.

### 7.2 Elektronické zdroje

Co je Access? Podpora Microsoftu [online]. Redmond: Microsoft, 2021 [cit. 2021-7-19]. Dostupné z: <https://support.microsoft.com/cs-cz/office/video-co-je-access-f2338765-ff59-4cfc-b8ba-74059fcb1874>

Co je to databáze. Oracle Česká republika [online]. Praha: Oracle Czech, 2021 [cit. 2021-7-15]. Dostupné z: <https://www.oracle.com/cz/database/what-is-database/>

Databázový software a aplikace | Microsoft Access. Podpora Microsoftu [online]. Redmond: Microsoft, 2021 [cit. 2021-7-19]. Dostupné z: <https://www.microsoft.com/cs-cz/microsoft-365/access>

Datové typy pro desktopové databáze Accessu. Podpora Microsoftu [online]. Redmond: Microsoft, 2021 [cit. 2021-7-21]. Dostupné z: [https://support.microsoft.com/cs-cz/office/datov%C3%A9-typy-pro-desktopov%C3%A9-datab%C3%A1ze-accessu-df2b83ba-cef6-436d-b679-3418f622e482#ID0EAABAAA=Newer\\_versions](https://support.microsoft.com/cs-cz/office/datov%C3%A9-typy-pro-desktopov%C3%A9-datab%C3%A1ze-accessu-df2b83ba-cef6-436d-b679-3418f622e482#ID0EAABAAA=Newer_versions)

Datové typy pro webové aplikace Accessu. Podpora Microsoftu [online]. Redmond: Microsoft, 2021 [cit. 2021-7-21]. Dostupné z: <https://support.microsoft.com/cs-cz/office/datov%C3%A9-typy-pro-webov%C3%A9-aplikace-accessu-33e10f90-5b06-4e56-ab0e-b0930f4c2367>

Datový typ Memo se teď nazývá Dlouhý text. Podpora Microsoftu [online]. Redmond: Microsoft, 2021 [cit. 2021-7-21]. Dostupné z: <https://support.microsoft.com/cs-cz/office/datov%C3%BD-typ-memo-se-te%C4%8F-naz%C3%BDv%C3%A1-dlouh%C3%BD-text-dffe5e34-953e-4451-a05e-fba5d9b564b5>

Divize firmy. Technologie pro kolejovou dopravu, manipulaci se sypkými materiály a vážící systémy - Sklenář [online]. Dlouhoňovice: Sklenář, c2021 [cit. 2021-6-24]. Dostupné z: <https://www.sklenar.cz/cs/m-2-divize-firmy>

IBM IMS documentation. IBM [online]. Armonk, New York: IBM, 2021 [cit. 2021-7-15]. Dostupné z: <https://www.ibm.com/docs/en/ims>

Lekce 1 - Úvod a prostředí programu MS Access. ITnetwork.cz [online]. Praha: Luboš Marvan, 2021 [cit. 2021-7-19]. Dostupné z: <https://www.itnetwork.cz/ms-office/microsoft-access-zaklady/uvod-a-prostredi-programu-ms-access>

Lekce 9: Co je DDL, DML a DQL? Learn2Code [online]. Žilina: Learn2Code o.z., 2021 [cit. 2021-6-22]. Dostupné z: <https://learn2code.cz/aplikace/skupiny#/sql-data-analyza-i/chapters/lekce-9-co-je-ddl-dml-a-dql>

Přidání nebo změna primárního klíče tabulky v Accessu. Podpora Microsoftu [online]. Redmond: Microsoft, 2021 [cit. 2021-7-19]. Dostupné z: <https://support.microsoft.com/cs-cz/office/p%C5%99id%C3%A1n%C3%AD-nebo-zm%C4%9Bna-prim%C3%A1rn%C3%ADho-kl%C3%AD%C4%8De-tabulky-v-accessu-07b4a84b-0063-4d56-8b00-65f2975e4379>

Rozhodněte se, jestli chcete vytvořit desktopovou databázi nebo webovou aplikaci pro Access. Podpora Microsoftu. [online]. Redmond: Microsoft, 2021 [cit. 2021-7-21]. Dostupné z: <https://support.microsoft.com/cs-cz/office/rozhodn%C4%9Bte-se-jestli-chcete-vytvo%C5%99it-desktopovou-datab%C3%A1zi-nebo-webovou-aplikaci-pro-access-7bf7ccc9-0850-48f2-858f-273271d30fa0>

SQL Data analýza I. Learn2Code [online]. Žilina: Learn2Code, 2021 [cit. 2021-7-19]. Dostupné z: <https://learn2code.cz/kurzy/sql-data-analyza-i>

Úvod k programování v aplikaci Access. Podpora Microsoftu [online]. Redmond: Microsoft, 2021 [cit. 2021-7-19]. Dostupné z: <https://support.microsoft.com/cs-cz/office/%C3%BAvod-k-programov%C3%A1n%C3%AD-v-aplikaci-access-92eb616b-3204-4121-9277-70649e33be4f#bmconvertmacrostovba>

Vytváření relací M:N. Podpora Microsoftu [online]. Redmond: Microsoft, 2021 [cit. 2021-7-19]. Dostupné z: <https://support.microsoft.com/cs-cz/office/video-vytv%C3%A1%C5%99en%C3%AD-relac%C3%AD-m-n-e65bcc53-8e1c-444a-b4fb-1c0b8c1f5653>

Základní informace o databázích. Podpora Microsoftu [online]. Redmond: Microsoft, 2021 [cit. 2021-7-19]. Dostupné z: <https://support.microsoft.com/cs-cz/office/z%C3%A1kladn%C3%AD-informace-o-datab%C3%A1z%C3%ADch-a849ac16-07c7-4a31-9948-3c8c94a7c204>

### 7.3 Jiné zdroje

FAVEREAU, Rafael. Diseño de Bases de Datos [přednáška]. Santiago de Chile: Universidad Técnica Federico Santa María, 24. května 2020.

FAVEREAU, Rafael. Structured Query Language [přednáška]. Santiago de Chile: Universidad Técnica Federico Santa María, 9. června 2020.

## 8 Seznam obrázků

Obrázek 1: Model hierarchického databázového systému .....	12
Obrázek 2: Model síťového databázového systému .....	13
Obrázek 3: Entita s atributy.....	16
Obrázek 4: Grafické zobrazení E-R diagramu se vztahem 1:1 .....	22
Obrázek 5: E-R diagram se vztahem 1:1 .....	22
Obrázek 6: Grafické zobrazení E-R diagramu se vztahem 1:N .....	23
Obrázek 7: E-R diagram se vztahem 1:N .....	23
Obrázek 8: Grafické zobrazení E-R diagramu se vztahem M:N.....	23
Obrázek 9: E-R diagram se vztahem M:N .....	24
Obrázek 10: Divize firmy .....	30
Obrázek 11: Technické oddělení divize Technologie pro manipulaci se sypkými materiály .....	31
Obrázek 12: Procesní mapa - zkouška sypkých materiálů .....	32
Obrázek 13: Protokol ze zkoušky sypkého materiálu .....	33
Obrázek 14: Navržený E-R diagram .....	35
Obrázek 15: Relace v MS Access .....	36
Obrázek 16: Ukázka načtení dat z více xls souborů .....	39
Obrázek 17: Získání potřebných dat .....	39
Obrázek 18: Část tabulky připravené v Excelu pro import do Accessu do tabulky PFT .....	40
Obrázek 19: Návrhové zobrazení dotazu pro získání potřebných dat k tvorbě protokolu ze zkoušky .....	40
Obrázek 20: Uchování vzorků pro budoucí potřeby .....	42
Obrázek 21: Grafické znázornění Dotazu2 sloužící k zobrazení sestavy pro tisk štítku .....	42
Obrázek 22: Přídávací dotazy v MS Access .....	47
Obrázek 23: Odstraňovací dotazy v MS Access .....	48
Obrázek 24: Formulář - Úvodní strana .....	49
Obrázek 25: Formulář - Evidence vzorků .....	50
Obrázek 26: Formulář - Provádění zkoušky .....	51
Obrázek 27: Formulář - Všechny údaje o vzorku .....	52
Obrázek 28: Záhlaví formuláře Všechny údaje o vzorku .....	52
Obrázek 29: Formulář - Všechny údaje o zkoušce .....	53
Obrázek 30: Záhlaví formuláře Všechny údaje o zkoušce .....	53

Obrázek 31: Záhloví sestavy ProtokolZeZkousky .....	54
Obrázek 32: Výsledná sestava protokolu ze zkoušky exportovaná do formátu PDF .....	55
Obrázek 33: Vygenerovaný štítek pro tisk sestavou .....	56
Obrázek 34: Záhloví sestavy Štítky pro Dotaz2 .....	56

## Příloha A

### SQL kód pro vytvoření tabulek z E-R diagramu

```
CREATE TABLE [Zhodnoceni] (  
  [ID] SMALLINT,  
  [Zkouska_ID] SMALLINT,  
  [Zaver] NVARCHAR(1500),  
  [DatumProvedeni] DATE,  
  PRIMARY KEY ([ID])  
);
```

```
CREATE TABLE [Zkouska] (  
  [Zkouska_ID] SMALLINT,  
  [CisloVzorku] SMALLINT,  
  [Zkousejici_ID] TINYINT,  
  [DatumZahajeni] DATE,  
  PRIMARY KEY ([Zkouska_ID]),  
);
```

```
CREATE TABLE [Sitovani] (  
  [ID] SMALLINT,  
  [Zkouska_ID] SMALLINT,  
  [StredniVelikostCastic] DECIMAL(8,6),  
  [HmotnostCastic5mm] DECIMAL(6,2),  
  [HmotnostCastic4mm] DECIMAL(6,2),  
  [HmotnostCastic2mm] DECIMAL(6,2),  
  [HmotnostCastic500um] DECIMAL(6,2),  
  [HmotnostCastic250um] DECIMAL(6,2),  
  [HmotnostCastic125um] DECIMAL(6,2),  
  [HmotnostCastic63um] DECIMAL(6,2),  
  [HmotnostCastic45um] DECIMAL(6,2),  
  [PodilCastic] NVARCHAR(150),  
  [VelikostCastic] NVARCHAR(50),  
  PRIMARY KEY ([ID])  
);
```

```

CREATE TABLE [Vzorek] (
[CisloVzorku] SMALLINT,
[Material] NVARCHAR(100),
[Zkouska_ID] SMALLINT,
[Zakaznik] NVARCHAR(50),
[CisloZakazky] NVARCHAR(20),
[DatumPrijeti] DATE,
[TypDopravy] NVARCHAR(30),
[PopisMaterialu] NVARCHAR(100),
[HmotnostVzorku_Prijeti] DECIMAL(4,2),
PRIMARY KEY ([CisloVzorku])
);

```

```

CREATE TABLE [PFT] (
[ID] SMALLINT,
[Zkouska_ID] SMALLINT,
[SypnaHmotnostPocatecni] SMALLINT,
[SypnaHmotnostKonecna] SMALLINT,
[KlasifikaceVlastnosti] SMALLINT,
[EfektivniUhelVnitrihoTreni] NVARCHAR(20),
[EfektivniUhelVnejsihoTreni] NVARCHAR(20),
[PredpokladKlenbovani] NVARCHAR(3),
[PredpokladTvorbyKominu] NVARCHAR(3),
[NavrzenyUhelDN400] SMALLINT,
[KritickaMernaHmotnost_Klenba] SMALLINT,
[KritickaMernaHmotnost_Tunel] SMALLINT,
PRIMARY KEY ([ID])
);

```

```

CREATE TABLE [UrceniSypnehoUhlu] (
[ID] SMALLINT,
[Zkouska_ID] SMALLINT,
[Uhel] NVARCHAR(10),
[Metoda] NVARCHAR(20),
PRIMARY KEY ([ID])
);

```



```
CREATE TABLE [Zkousejici] (  
[Zkousejici_ID] TINYINT,  
[Jmeno] NVARCHAR(20),  
[Titul] NVARCHAR(15),  
PRIMARY KEY ([Zkousejici_ID])  
);
```

```
CREATE TABLE [MereniNaVaze] (  
[ID] SMALLINT,  
[Zkouska_ID] SMALLINT,  
[SypnaHmotnostVolna] SMALLINT,  
[SypnaHmotnostSestresna] SMALLINT,  
PRIMARY KEY ([ID])  
);
```

## Příloha B

### Obsah přiloženého CD

LomDenis-DiplomovaPrace.pdf.....text práce ve formátu PDF

ZkouskaVzorkuSypkychMaterialu.accdb.....databáze Microsoft Access