

VŠB – Technická univerzita Ostrava
Ekonomická fakulta
Katedra aplikované informatiky

Návrh a implementace databázové aplikace pro výrobní firmu
Design and Implementation of Database Application for Manufacturing Company

Student:

Martin Báča

Vedoucí bakalářské práce:

Ing. Vítězslav Novák, Ph.D.

Ostrava 2018

VŠB - Technická univerzita Ostrava
Ekonomická fakulta
Katedra aplikované informatiky

Zadání bakalářské práce

Student: **Martin Báča**

Studijní program: B6209 Systémové inženýrství a informatika

Studijní obor: 6209R017 Informatika v ekonomice

Téma: **Návrh a implementace databázové aplikace pro výrobní firmu**
Design and Implementation of Database Application for a
Manufacturing Company

Jazyk vypracování: čeština

Zásady pro vypracování:

1. Úvod
 2. Teoretická východiska pro tvorbu databázové aplikace
 3. Analýza současného stavu a požadavků
 4. Návrh a implementace databázové aplikace
 5. Závěr
- Seznam použité literatury
Seznam zkratk
Prohlášení o využití výsledků bakalářské práce
Seznam příloh
Přílohy

Seznam doporučené odborné literatury:

CONNOLLY, T., C. BEGG a R HOLOWCZAK. *Mistrovství – databáze: profesionální průvodce tvorbou efektivních databází*. Brno: Computer Press, 2009. ISBN 978-80251-2328-7.
OPPEL, Andrew J. *Databáze bez předchozích znalostí*. Brno: Computer Press, 2006. ISBN 80-251-1199-7.
VIESCAS, John a Jeff CONRAD. *Mistrovství v Microsoft Office Access 2007*. Brno: Computer Press, 2008. ISBN 978-80-251-2162-7.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Vítězslav Novák, Ph.D.**

Datum zadání: 24.11.2017

Datum odevzdání: 11.05.2018



Ing. Petr Rozehnal, Ph.D.
vedoucí katedry



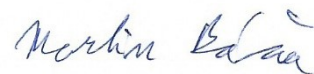
prof. Dr. Ing. Zdeněk Zmeškal
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem celou bakalářskou práci včetně příloh vypracoval samostatně pod vedením vedoucího bakalářské práce a uvedl jsem všechny použité podklady a literaturu.

Zároveň bych chtěl na tomto místě poděkovat svému vedoucímu bakalářské práce, Ing. Vítězslavu Novákovi, Ph.D. za vedení bakalářské práce, cenné rady a připomínky. Dále také své rodině za možnost studovat.

V Ostravě 2. 5. 2018



podpis studenta

Obsah

| | | |
|----------|--|----------|
| 1 | Úvod..... | 5 |
| 2 | Teoretická východiska pro tvorbu databázové aplikace..... | 6 |
| 2.1 | Základní pojmy databází..... | 6 |
| 2.2 | Databázový systém (SŘBD)..... | 6 |
| 2.3 | Vrstvy datové abstrakce..... | 7 |
| 2.3.1 | Fyzická vrstva..... | 8 |
| 2.3.2 | Logická vrstva..... | 9 |
| 2.3.3 | Externí vrstva..... | 9 |
| 2.4 | Datové modely..... | 9 |
| 2.4.1 | Konceptuální model..... | 9 |
| 2.4.1.1 | Entita..... | 10 |
| 2.4.1.2 | Atribut..... | 10 |
| 2.4.1.3 | Relace..... | 11 |
| 2.4.1.4 | Klíč..... | 12 |
| 2.4.2 | Logický model..... | 13 |
| 2.4.2.1 | Hierarchický model..... | 13 |
| 2.4.2.2 | Síťový model..... | 14 |
| 2.4.2.3 | Objektově orientovaný model..... | 14 |
| 2.4.2.4 | Objektově relační model..... | 15 |
| 2.4.2.5 | Not only SQL databáze..... | 15 |
| 2.4.2.6 | Relační model..... | 16 |
| 2.4.3 | Fyzický model..... | 19 |
| 2.5 | Structured Query Language (SQL)..... | 20 |
| 2.5.1 | Kategorie příkazů jazyka SQL..... | 20 |
| 2.6 | Microsoft Access..... | 21 |
| 2.6.1 | Tabulky..... | 21 |
| 2.6.1.1 | Přehled datových typů..... | 21 |
| 2.6.2 | Vlastnosti datových typů..... | 22 |
| 2.6.3 | Dotazy..... | 23 |
| 2.6.4 | Formuláře..... | 23 |
| 2.6.5 | Sestavy..... | 24 |

| | | |
|----------|---|-----------|
| 2.7 | Visual Basic for Applications | 24 |
| 2.8 | XML..... | 24 |
| 3 | Analýza současného stavu a požadavků | 26 |
| 3.1 | Charakteristika podniku..... | 26 |
| 3.2 | Analýza současného stavu | 26 |
| 3.3 | Požadavky pro tvorbu databázové aplikace..... | 26 |
| 4 | Návrh a implementace databázové aplikace | 28 |
| 4.1 | Konceptuální návrh databáze | 28 |
| 4.2 | Fyzický návrh databáze | 30 |
| 4.3 | Návrh jednotlivých tabulek | 30 |
| 4.3.1 | Tabulka Dodavatel..... | 31 |
| 4.3.2 | Tabulka Odberatel..... | 32 |
| 4.3.3 | Tabulka Material..... | 33 |
| 4.3.4 | Tabulka Objednavka..... | 34 |
| 4.3.5 | Tabulka PolozkaObjednavky | 35 |
| 4.3.6 | Tabulka Vyrobek..... | 36 |
| 4.3.7 | Tabulka PolozkaVyrobku..... | 36 |
| 4.3.8 | Tabulka Zamestnanec..... | 37 |
| 4.4 | Formuláře a sestavy..... | 38 |
| 4.4.1 | Úvodní formulář | 38 |
| 4.4.2 | Formulář Výrobky | 39 |
| 4.4.3 | Formulář Materiál | 40 |
| 4.4.4 | Formulář Odběratelé | 42 |
| 4.4.5 | Formulář Zaměstnanci..... | 43 |
| 4.4.6 | Formulář Dodavatelé | 43 |
| 4.4.7 | Formulář Objednávky | 45 |
| 4.5 | Zabezpečení databázové aplikace..... | 47 |
| 5 | Závěr..... | 48 |

1 Úvod

Cílem této bakalářské práce je vytvoření databázové aplikace, která poskytuje evidenci veškerých ekonomických subjektů, materiálu, nabízených výrobků a správu objednávek prostřednictvím softwaru Microsoft Access. Neboť v současné době se úspěšný podnik už zcela neobejde bez základní databázové aplikace a řízení vztahů se zákazníky. Dobrá databázová aplikace totiž dokáže firmě pomoci řídit veškerá obchodní data a také je centralizovat a zabezpečit. Je také dobrým pomocníkem při evidenci zaměstnanců či klientů, což dokáže zautomatizovat rutinní firemní procesy. Tato databázová aplikace by měla zlepšit efektivitu práce tím, že sníží administrativní zátěž, čímž také ušetří náklady.

Bakalářská práce bude realizována pro výrobní firmu se zaměřením na zámečnictví a drátěný program. Firma se specializuje převážně na výrobu železných drátěných konstrukcí, jako jsou například železné koše či regály. Vzhledem k nárůstu objemu produkce a větší spotřeby materiálu vznikla firmě potřeba výrobní zakázky a informace s nimi spjaté evidovat. Tato aplikace bude primárně určena pro vedoucího firmy, který bude aplikaci nejčastěji obsluhovat. Bližší charakteristika firmy a přehled požadavků na aplikaci bude popsán ve třetí kapitole.

Databázová aplikace bude vyvinuta v prostředí softwaru od společnosti Microsoft, která je součástí produktu Microsoft Office. Majitel firmy disponuje licencí k Microsoft Office 365 a nepřál si investovat velké množství finančních prostředků, proto jsem mu navrhl toto řešení. Funkční logika aplikace bude naprogramována v programovacím jazyku Visual Basic for Applications od stejnojmenné společnosti. Databáze vytvořené v Microsoft Access se řadí mezi databáze relační, a proto bude využit jazyk SQL neboli Structured Query Language. Uživatelské rozhraní bude intuitivní a přehledné. Pro jeho vytvoření se využije grafických prvků z Microsoft Access, jako jsou například sestavy či formuláře, které obsahují tlačítka, textové pole a jiné. Návrh a implementace databázové aplikace spolu s programováním funkcionalit bude popsána v kapitole čtvrté.

2 Teoretická východiska pro tvorbu databázové aplikace

Tato část práce shrnuje obecné definice a pojmy, které je nutné znát při realizaci databázové aplikace.

2.1 Základní pojmy databází

V kapitole je uvedeno, co vlastně databáze znamená a co je možné si pod tímto pojmem představit. Jsou zde popsány základní pojmy, o kterých je nutné mít povědomí při tvorbě databázové aplikace.

Prvním pojmem jsou data. Tímto se rozumí veškeré získané údaje, které jsou uloženy v databázových tabulkách. Nemají žádnou strukturu, dokud nejsou zpracována. Zpracováním dat pak vzniknou možné informace (Conolly, Begg a Holowczak, 2009).

Informace jsou již data, která prošla určitým zpracováním či strukturací, dávají tedy smysl a dá se jim porozumět (Conolly, Begg a Holowczak, 2009).

Pojem databáze je kolekce vzájemně souvisejících dat, s nimiž se dá pracovat jako s ucelenou jednotkou. Tato kolekce je pak typicky uložena na specifickém paměťovém médiu (Oppel, 2006).

Posledním pojmem je pojem Databázová aplikace. Databázová aplikace představuje počítačový program, který je integrovaný s určitou databází. Zároveň funguje jako uživatelské rozhraní, které zajišťuje komunikaci s databází. V současnosti se vytvářejí především webové aplikace, které jsou dostupné prostřednictvím webového prohlížeče. Tyto online aplikace lze programovat v některém z vyšších programovacích jazyků, například C#, Java, Visual Basic, aplikace pak posílají požadavky do databázového systému, nejčastěji pomocí příkazů jazyka SQL (Conolly, Begg a Holowczak, 2009).

2.2 Databázový systém (SŘBD)

Databázový systém, někdy také označovaný jako Systém řízení báze dat, je software, který zajišťuje práci s databází. Jinými slovy vytváří rozhraní mezi

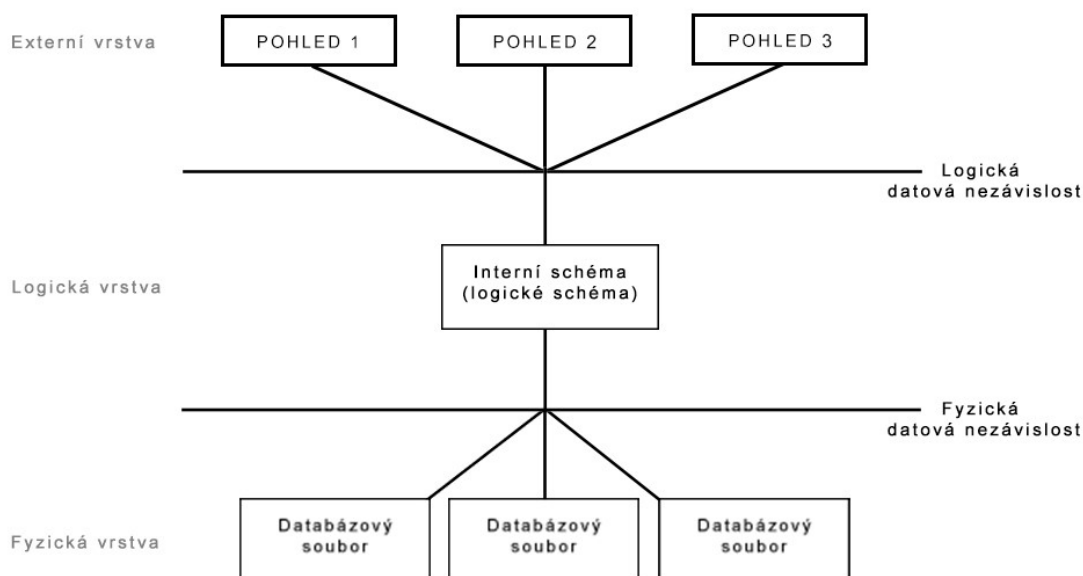
programy a uloženými daty. Dodává jej výrobce daného systému řízení báze dat. Mezi tyto softwarové produkty databázových systémů se řadí například: Microsoft Access, Microsoft SQL Server, Oracle, DB2, MySQL. Databázový systém zajišťuje následující nezbytné služby pro chod celé databáze:

- manipulaci s daty,
- správu přístupu při přístupu více uživatelů k jednotlivým datům, včetně zabezpečení proti konfliktům při aktualizaci dat,
- správu transakcí, díky kterým je možné současně vykonávat změny v databázi v rámci jedné nedělitelné jednotky,
- podporu dotazovacích jazyků (např. SQL),
- funkci na zálohování a obnovu databáze,
- zabezpečovací mechanismy (Oppel, 2006).

2.3 Vrstvy datové abstrakce

Databáze mají schopnost nabízet různým uživatelům odlišné samostatné pohledy na data. Tyto pohledy se nazývají typicky „uživatelské pohledy“. Za uživatele se může považovat osoba či aplikace, která je oprávněna se přihlásit do databáze a smí načíst nebo uložit data.

V databázovém systému může být každému uživateli nabídnut jiný pohled na stejná data. Tento pohled se vyjadřuje jako již zmíněný, uživatelský pohled. Uživatelské pohledy mohou být individuálně přizpůsobeny každému uživateli. To je možné díky takzvaným vrstvám abstrakce, které znázorňuje obrázek 2.1 (Oppel, 2006).



Obrázek 2.1: Vrstvy abstrakce v databázi, zdroj Oppel, (2006, s. 19), upraveno.

Architekturu, která je zobrazena na obrázku 2.1, navrhl v sedmdesátých letech výbor SPARC (Standards Planning and Requirements Committee) amerického ANSI (American National Standards Institute), stala se základním stavebním kamenem v oblasti výzkumu a vývoje databází. V současnosti je to jedna z nejrozšířenějších architektur a je na ní založena většina moderních databázových systémů. Je složena ze tří vrstev: fyzické vrstvy, logické vrstvy a externí vrstvy (Oppel, 2006).

2.3.1 Fyzická vrstva

Fyzická vrstva zahrnuje datové soubory, do kterých se ukládají všechna data z dané databáze. V této datové vrstvě může být jedna databáze uložena do více datových souborů, které mohou být umístěny v několika různých uložiscích. Díky tomuto efektivnějšímu uspořádání mohou všechny jednotky pracovat současně a celkový výkon databáze se tak zvýší.

Jednou z výjimek je Microsoft Access, ve kterém je celá databáze uložena do jednoho fyzického datového souboru. Díky tomuto přístupu si databázový systém nedokáže poradit s větším počtem pracujících uživatelů, proto není Microsoft Access vhodný pro větší podnikové systémy. Na druhou stranu se tím

ale zjednodušuje práce s databází jednotlivému uživateli, protože se celá databáze nachází na jednom osobním počítači (Oppel, 2006).

2.3.2 Logická vrstva

Další z datových vrstev abstrakce je vrstva logická, tato vrstva je pouze součástí abstraktních logických struktur, které se podle potřeby mohou skládat z objektů fyzické vrstvy. Vrstva má schopnost změny logické struktury databáze, aniž by tím byli poškozeni samotní uživatelé. Logická vrstva se též někdy označuje pojmem „schéma“, jenž určuje kolekci všech datových položek, které jsou uloženy v dané databázi (Oppel, 2006).

2.3.3 Externí vrstva

Externí vrstva je vrstva, ve které se k databázi připojují jednotliví uživatelé a aplikační programy, dále s databází pracují a zadávají v ní dotazy. Tato vrstva je tvořena uživatelskými pohledy, umožňuje tak uživatelům přístup k datům způsobem, který je individuálně přizpůsoben podle nich, respektive stejná data jsou současně viditelná různými uživateli, různými způsoby. Jelikož se tyto uživatelské pohledy ukládají přímo do databáze, mohou se kdykoliv znovu využít. Vrstva také díky těmto pohledům poskytuje výkonný a flexibilní bezpečnostní mechanismus, který skrývá určité části databáze od uživatelů, kteří nemají oprávnění na zobrazení některých specifických dat (Oppel, 2006).

2.4 Datové modely

Datový model je množina všech prostředků a návrhů, které definují formát a strukturu jednotlivých dat a jejich vztahů (Novák, 2015).

Tato kapitola popisuje jednotlivé datové modely a pojmy spojené se základní tvorbou tabulek v relačních databázích.

2.4.1 Konceptuální model

Konceptuální model slouží k identifikaci klíčových entit a jejich vztahů, které zachycují situace reálného světa a je potřeba je přenést do samotné databáze. Z konceptuálního modelu se posléze vytváří model logický, který z konceptuálního modelu přímo vychází (Connolly, Begg a Holowczak, 2009).

V této kapitole budou popsány jednotlivé komponenty konceptuálního modelu.

2.4.1.1 Entita

Entitu si lze představit například jako věc, osobu, zvíře nebo myšlenku, o níž jsou shromažďovány nějaká data. Jinými slovy je to množina objektů se stejnými vlastnostmi. Tyto entity se pak zaznamenávají do databází. Každá entita je identifikována jedinečným jménem a výčtem vlastností, kterým se říká atributy. Na obrázku 2.2 je zobrazen příklad entity Zákazník (Oppel, 2006).



Obrázek 2.2: Entita, zdroj: Oppel (2006, s. 40), upraveno.

2.4.1.2 Atribut

Atribut je jednotka faktu či vlastnost, která charakterizuje určitým způsobem entitu. V konceptuálním návrhovém modelu jsou typicky zobrazeny jako jednotlivé názvy uvnitř entity. Atribut, který je zobrazen nad vodorovnou čarou na obrázku 2.3, tvoří takzvaný jedinečný identifikátor entity, který pro každou entitu vždy obsahuje jedinečnou hodnotu. Atributy lze rozdělit na jednoduché a složené. Složeným atributem může být například Adresa, která se skládá z jednotlivých atributů: Ulice, Město, PSČ. Hodnota jednoduchého atributu by měla být atomická, to znamená, že není možné, aby se hodnota atributu rozdělila na několik menších jednotek (Oppel, 2006; Conolly, Begg a Holowczak, 2009).

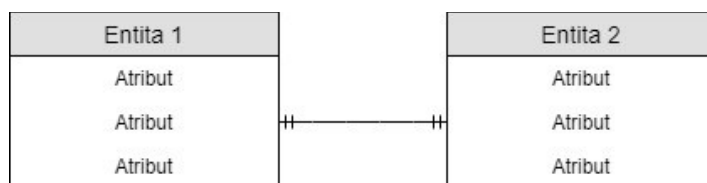


Obrázek 2.3: Atribut, zdroj: Oppel (2006, s. 40), upraveno.

2.4.1.3 Relace

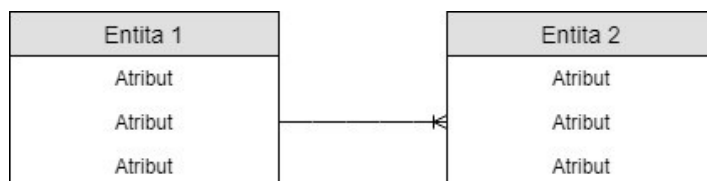
Jelikož se do databází ukládají data, které spolu nějakým způsobem souvisí, je proto zapotřebí tato data propojit dohromady, toto propojení zajišťují relace. Jinými slovy, relace definují vzájemné vztahy mezi jednotlivými entitami. Lze je rozdělit do tří základních typů (Oppel, 2006).

Prvním typem je relace jedna k jedné (1:1), která zobrazuje vztah, kdy jednomu atributu v první entitě odpovídá atribut v entitě druhé. Tento vztah lze také řešit sloučením těchto entit do jediné viz obrázek 2.4.



Obrázek 2.4: Relace 1:1, zdroj: vlastní.

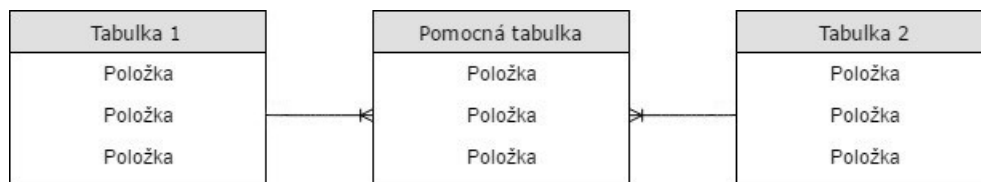
Dalším typem relace je relace jedna k více (1:N), kde jednomu atributu v první entitě odpovídá n-atributů v druhé entitě. Tento vztah je znázorněn na obrázku 2.5.



Obrázek 2.5: Relace 1:N, zdroj: vlastní.

Posledním typem je relace více k více (N:M), kde n-atributům v první entitě odpovídá m-atributů v entitě druhé.

V relační koncepci se tento vztah řeší vytvořením takzvané pomocné či spojovací tabulky (entity), která obsahuje primární klíče ze dvou původních tabulek. Díky pomocné tabulce se pak může relace převést na relaci 1:N, se kterou si již relační databáze poradí viz obrázek 2.6.



Obrázek 2.6: Relace N:M, zdroj: vlastní.

2.4.1.4 Klíč

Každý záznam v entitě (tabulce) musí být nějak jednoznačně identifikován, proto jsou pro tyto účely stanoveny speciální atributy, které tuto jedinečnost dokáží zajistit (Conolly, Begg a Holowczak, 2009).

- **Superklíč** – jedná se o jakékoliv uskupení atributů (sloupců), které jednoznačně identifikuje jednotlivé záznamy v entitě. Některé prvky mohou obsahovat atributy, které jsou pro tuto jednoznačnou identifikaci zbytečné, proto je lepší použít například klíč primární (Conolly, Begg a Holowczak, 2009).
- **Kandidátní klíč** – jedná se o zjednodušený superklíč, který se skládá z minimálního počtu atributů, které jsou nutné k jedinečné identifikaci jednotlivých záznamů (Conolly, Begg a Holowczak, 2009).
- **Primární klíč** – primární klíč je kandidátní klíč, který je zvolen pro jedinečnou identifikaci entity. Typicky se značí PK. Na obrázku 2.7 je vidět ukázka primárního klíče, který je součástí entity Zákazník (Conolly, Begg a Holowczak, 2009; Opperl 2006).



Obrázek 2.7: Ukázka primárního klíče, zdroj: vlastní.

- **Cizí klíč** – je jeden nebo množina atributů v jedné entitě, která je stejná jako kandidátní klíč v entitě jiné. Jinými slovy identifikuje záznamy z jiných entit, které spolu souvisí. Obvykle se značí FK (Conolly, Begg a Holowczak, 2009).

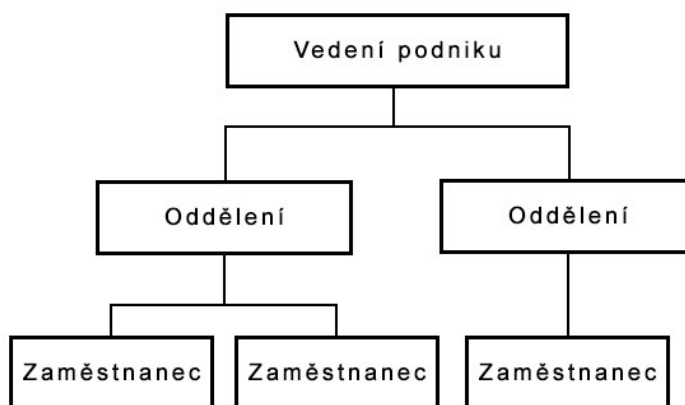
2.4.2 Logický model

Logický model vychází z konceptuálního modelu. V případě relační koncepce převádí logický model jednotlivé entity a jejich vzájemné vztahy do relační podoby, respektive do relačních tabulek. Z logického modelu pak vychází model fyzický (Conolly, Begg a Holowczak, 2009).

Podle způsobu ukládání dat a závislostí mezi nimi dělíme databáze do několika typů. Následující podkapitoly charakterizují možné databázové modely.

2.4.2.1 Hierarchický model

Hierarchický model umožnil vznik prvotních databázových systémů. Byl inspirován tradičními souborovými systémy. Všechny záznamy v modelu jsou obvykle uspořádány do takzvané stromové datové struktury. Záznamy jsou zde propojeny pomocí ukazatelů, které zahrnují adresu příslušného záznamu. Ukazatel tak počítačovému systému přímo říká, kde se svázaný záznam nachází. Ukazatel tedy definuje vztah rodič-potomek, který lze shrnout pojmem relace jedna k více. Z toho vyplývá, že právě jeden rodič může mít více potomků a každý potomek má pouze jednoho rodiče. Podobnou strukturu lze vidět na obrázku 2.8, který znázorňuje strukturu uspořádání podniku (Oppel, 2006).

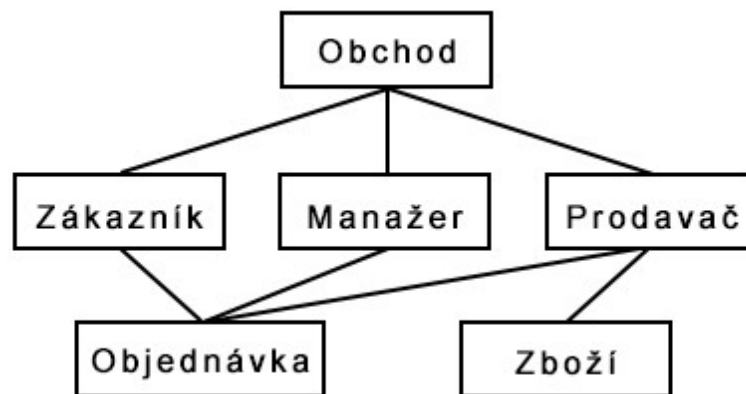


Obrázek 2.8: Struktura hierarchického modelu, zdroj: vlastní.

Hlavní nevýhodou tohoto modelu je především malá flexibilita. Další nevýhodou je pak složité řešení při vkládání nových záznamů, popřípadě jejich rušení. Proto byl v sedmdesátých letech hierarchický databázový model postupně nahrazen modelem síťovým (Oppel, 2006).

2.4.2.2 Síťový model

Síťový model vznikl ve stejné době jako model hierarchický. Jedná se v podstatě o rozšíření hierarchického modelu. Podobně jako tomu bylo u modelu hierarchického, se zde taktéž propojují příbuzné záznamy pomocí ukazatelů, které obsahují fyzické adresy. Hlavní výhodou tohoto modelu je podpora vazeb více k více, tedy jeden potomek může mít více rodičů, jak je možné vidět na obrázku 2.9.



Obrázek 2.9: Struktura síťového modelu, zdroj: vlastní.

Nevýhoda tohoto modelu nastává při potřebě změnit jeho strukturu. Při změně totiž může dojít k celkové obměně databáze, podobně jako u hierarchického modelu (Oppel, 2006).

2.4.2.3 Objektově orientovaný model

Objektově orientovaný model je databázová struktura, ve které jsou informace představovány objekty. Byl navrhnut pro spolehlivou práci s objektově orientovanými jazyky. Objekt si lze představit jako logické seskupení programové logiky a příbuzných dat, které společně představují nějakou reálnou věc či osobu z reálného světa, kupříkladu výrobek nebo zákazníka. Jednotlivým datovým položkám v objektově orientovaném modelu se říká atributy. Proměnné jsou uloženy u každého objektu. Objekt zákazníka schematicky demonstruje obrázek 2.10 (Oppel, 2006).



Obrázek 2.10: „Anatomie“ objektu, zdroj: Opperl (2006, s. 30).

Největší výhodou tohoto modelu je hlavně snadnější aktualizace dat. Nevýhodou této databázové struktury je pak její složitější návrh a následný vývoj.

2.4.2.4 Objektově relační model

Objektově relační model kombinuje objektový a relační model databází. Objektově relační model nabízí například větší flexibilitu v oblasti datových typů. Typickým příkladem využití tohoto druhu databází je online katalog zboží. Tento model patří k modernějším návrhům databáze a využívají jej největší výrobci databázových systémů, kterými jsou například IBM nebo Oracle. Výhodou objektově relačního modelu je podpora práce s čistě relačními daty (Opperl, 2006; IBM [online], 2018).

2.4.2.5 Not only SQL databáze

Not only SQL (NoSQL) je jedna z nejmodernějších databází, která je nejčastěji spojována s takzvanými velkými objemy dat (Big Data), ve kterých je zapotřebí uložit jejich obrovské množství. NoSQL databáze jsou totiž lépe optimalizované pro čtení a zápis než databáze relační, proto se k těmto účelům nejvíce hodí. Používají jiný koncept než klasická tabulková schémata relačních databází. Jinými slovy podporují nerelační datové modely, například grafové, dokumentové, klíč-hodnota a další. Většina NoSQL databází jsou open-source, což znamená, že jsou volně dostupné i bez nutnosti zakoupení licence. Mezi představitelé NoSQL databází se řadí například MongoDB, Apache Cassandra či Redis. Tento koncept využívají největší světové firmy, jako je například společnost Google nebo Amazon (Katsov [online], 2012; Bryant [online], 2014).

2.4.2.6 Relační model

U předchůdců relačního modelu, tedy u konceptuálního a síťového modelu, bylo zapotřebí znát předem definovanou cestu, aby bylo možné se dostat ke konečnému cíli. U relačního modelu tomu tak není, stačí svázat záznamy pouze podle potřeby, nikoli podle předem definované struktury záznamů, jako tomu bylo u předešlých databází. Relační databáze je místo, kam jsou data ukládána v podobě takzvaných databázových tabulek neboli relací. Jedná se o základní prvek pro tvorbu celé databáze (Oppel, 2006).

Každá relace by měla mít speciální sloupec, který identifikuje existující tabulku, tento prvek se nazývá primární (identifikační) klíč. Primární klíč má tu vlastnost, že jeho hodnota je zcela jedinečná, nemůže tedy nastat situace, kde by měly dva řádky stejnou hodnotu tohoto identifikačního klíče.

Jako příklad tabulky může být uvedena tabulka *Kniha*, která disponuje pěti sloupci, což je možné vidět na obrázku 2.11.

| Kniha | |
|-------|----------|
| PK | ID Knihy |
| | Název |
| | Datum |
| | Autor |
| | Cena |

Obrázek 2.11: Relační tabulka *Kniha*, zdroj: vlastní.

První sloupec reprezentuje identifikační klíč knihy (*ID Knihy*). Druhý sloupec je samotný název knihy. Třetí sloupec slouží k zápisu data publikování knihy. Ve čtvrtém sloupci se nachází autor knihy, poslední sloupec zobrazuje cenu dané knihy. Tabulka *Kniha* může být propojena vazbou (relací) například s tabulkou *Autor*. V tabulce 2.1, jsou pak vidět příklady jednotlivých položek (knih), které jsou uloženy v databázi.

Tabulka 2.1: Ukázka uložení záznamů v relační databázi, zdroj: vlastní.

| ID Knihy | Název | Datum | Autor | Cena |
|----------|---------------------|-------|----------------|--------|
| 1 | Staré pověsti české | 1894 | Alois Jirásek | 420 Kč |
| 2 | Babička | 1855 | Božena Němcová | 380 Kč |
| ... | ... | ... | ... | ... |

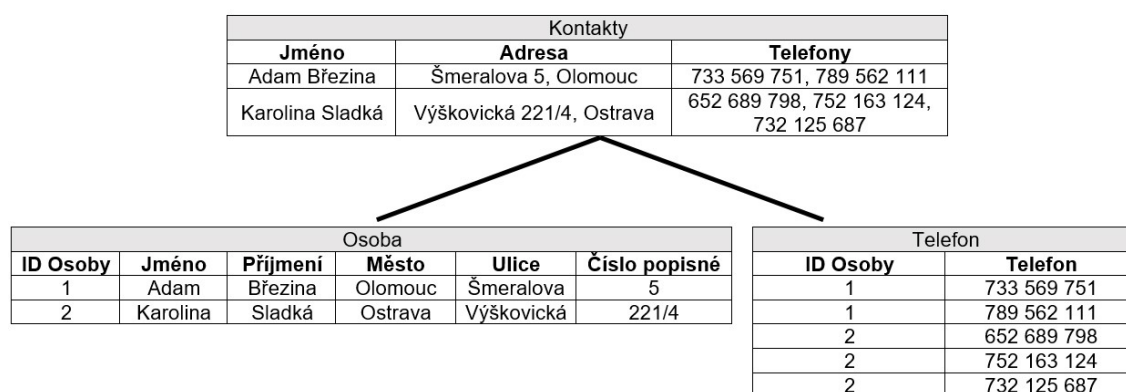
Tabulka

Jak již bylo zmíněno, tabulka je primární jednotkou pro ukládání dat v relační databázi. Je to dvourozměrná struktura, která se skládá ze záhlaví a těla. V záhlaví tabulky se nachází názvy jednotlivých sloupců, v těle jsou pak umístěny konkrétní řádky tabulky, které odpovídají prvkům dané relace. Důležité je zmínit, že relační tabulka je jen logickou záznamovou strukturou, ve fyzické vrstvě obvykle neexistuje ve skutečné tabulkové podobě. V tabulce se díky primárnímu klíči nikdy nemůže objevit duplicitní záznam. Pořadí uložených sloupců a záznamů nehraje žádnou roli, je proto možné je mezi sebou libovolně přehazovat (Oppel, 2006; Connoly, Begg a Holowczak, 2009).

Normalizace dat

Normalizace dat je proces, který řeší problematiku vztahů mezi atributy. Cílem normalizace je minimalizace redundantních, respektive nadbytečných dat. Jinými slovy to je postup, jak správně navrhnout a vytvořit strukturu databáze. Existuje několik takzvaných normálních forem. Z těchto forem je pro efektivní a správně fungující databázi zapotřebí z nich dodržet alespoň první tři (Oppel, 2006; Microsoft[online], 2018).

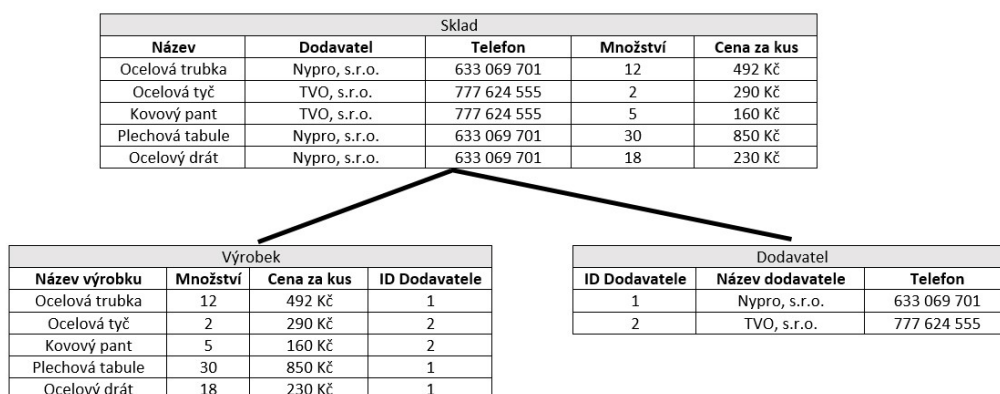
- **První normální forma** – tabulka je v této formě, pokud každý atribut obsahuje jen atomické, tedy dále nedělitelné údaje. Na obrázku 2.12 lze vidět příklad této normální formy (Connoly, Begg a Holowczak, 2009).



Obrázek 2.12: První normální forma, zdroj: vlastní.

Obrázek 2.12 zobrazuje tabulku *Kontakty*, ve které je vidět, že sloupec *Adresa* a *Telefony* nemají atomická data, proto je zapotřebí tabulku rozdělit na další dvě tabulky.

- **Druhá normální forma** – tabulka je ve druhé normální formě, pokud splní podmínky první normální formy a zároveň je v této formě zapotřebí, aby byly všechny neklíčové atributy závislé na primárním klíči. Řešit tuto normu je nutné v případě, když tabulka obsahuje složený primární klíč. Z toho vyplývá, že druhou normální formu splňují automaticky tabulky, které jsou v první normální formě a ve kterých se nachází pouze jeden jednoduchý primární klíč (Connoly, Begg a Holowczak, 2009).



Obrázek 2.13: Druhá normální forma, zdroj: vlastní.

Na obrázku 2.13 lze vidět tabulku *Sklad*, která využívá jako primární klíč kombinaci sloupců *Název* a *Dodavatel*. Položka *Telefon* (dodavatele) se tedy nevztahuje k celému primárnímu klíči, ale pouze k jeho části. Může zde dojít například k problému, že dojde ke smazání všech položek od jednoho dodavatele, tím pádem se ztratí jeho telefonní číslo. Je proto nutné tabulku *Sklad* opět rozdělit na dvě tabulky.

- **Třetí normální forma** – tabulka se nachází v této formě, pokud splňuje předchozí dvě formy a zároveň jsou všechny neklíčové atributy (sloupce) vzájemně nezávislé (Connoly, Begg a Holowczak, 2009).
- **Boyce-Coddova normální forma** – jedná se o vylepšenou třetí normální formu. K tomu, aby byla tabulka v této formě, musí být ve třetí normální formě a zároveň v relaci musí existovat více kandidátních klíčů, které jsou

složeny ze dvou či více atributů. Další skutečností je fakt, že mezi kandidátními klíči nesmí být žádná závislost (Oppel, 2006).

- **Čtvrtá normální forma** – v této normální formě se relace nachází, pokud splňuje podmínky Boyce-Coddovy normální formy, navíc však relace musí zobrazovat pouze příčinnou souvislost mezi atributem a identifikačním klíčem (Novák, 2015).
- **Pátá normální forma** – v páté normální formě se relace nachází, když je ve čtvrté normální formě a již není možné do ní přidat atribut či skupinu atributů, není jí tedy možné bezztrátově rozdělit (Novák, 2015).

Databázová transakce

Databázová transakce je skupina operací, které se buď všechny provedou správně nebo se neprovedou vůbec. Transakce musí splňovat následující vlastnosti:

- **Atomicita** – znamená, že transakce musí zůstat nerozdělená, z toho vyplývá, že se musí celá úspěšně dokončit nebo celá selhat. Pokud u transakce dojde k chybě, tedy selže, musí se všechny její změny vrátit do původního stavu. Při zápisu změn do databáze dochází k takzvanému potvrzení transakce. Naopak při selhání transakce se hovoří o vrácení transakce (Oppel, 2006).
- **Konzistence** – transakce převádí databázi z jednoho konzistentního stavu do druhého konzistentního stavu. Konzistentním stavem se rozumí stav, který poskytuje soudržná data, respektive data nejsou uloženy v „nesmyslném“ stavu, kupříkladu záporný věk (Oppel, 2006).
- **Izolovanost** – transakce musí vždy dokončit svou činnost nezávisle na ostatních transakcích, které se mohou zpracovávat ve stejný čas (Oppel, 2006).
- **Trvalost** – všechny změny, které byly provedeny úspěšně provedenými transakcemi, musí vždy trvale zůstat uloženy v databázi, a to i po kritickém selhání systému (Oppel, 2006).

2.4.3 Fyzický model

Fyzický model vychází z modelu logického a představuje nejnižší míru abstrakce. Přetváří logický model do konkrétního systému řízení báze dat,

například do Microsoft Access nebo SQL Server. Jinak řečeno zajišťuje fyzické uložení dat (Connoly, Begg a Holowczak, 2009).

2.5 Structured Query Language (SQL)

SQL je strukturovaný dotazovací jazyk a zároveň také standard, který se používá ke komunikaci s relačními databázemi. Řadí se mezi neprocedurální jazyky, to znamená, že počítači jsou sdělovány jen požadované výsledky, není třeba definovat, jak tyto výsledky získat. Prostřednictvím SQL jsou psány takzvané dotazy, což jsou požadavky, které jsou posílány konkrétní databázi, a uživateli je pak vrácena odpověď. Při psaní dotazů je vždy dodržováno přesné pořadí jednotlivých klauzulí. SQL se často využívá v kombinaci i s jinými programovacími jazyky (Oppel, 2008).

2.5.1 Kategorie příkazů jazyka SQL

Příkazy SQL se dělí podle své funkce do několika kategorií, respektive jazyků. Všechny tyto kategorie však dodržují stejná pravidla a syntaxi (zápis kódu). Jedná se o kategorie: DDL, DQL, DML a DCL (Oppel, 2008).

Data Definition Language (DDL)

Data Definition Language se využívá pro vytváření databázových objektů, například k vytváření tabulek, a jejich následnou úpravu. Mezi základní příkazy jazyka DDL se řadí:

- **CREATE** – vytváření databázových objektů,
- **DROP** – odstraňování databázových objektů,
- **ALTER** – úprava databázových objektů (Oppel, 2008).

Data Query Language (DQL)

Data Query Language slouží k načítání dat z databáze. Někteří autoři uvádí DQL jako součást jazyka DML. Příkazy jazyka DQL jsou založeny pouze na jediném SQL příkazu:

- **SELECT** – výběr dat z databáze (Oppel, 2008).

Data Manipulation Language (DML)

Data Manipulation Language se využívá pro přidávání dat do databáze nebo pro odebírání či aktualizaci dat. Do jazyka DML patří příkazy:

- **INSERT** – vkládání nových záznamů do databáze,
- **UPDATE** – úprava (aktualizace) stávajících záznamů,
- **DELETE** – odstranění záznamů (Oppel, 2008).

Data Control Language (DCL)

Data Control Language slouží k řízení přístupu k datům databáze. DCL využívá různá systémová oprávnění konkrétního systému řízení báze dat, příkladem může být spuštění nebo vypnutí databáze. Základními příkazy jazyka jsou:

- **GRANT** – udělování uživatelského oprávnění dalším subjektům,
- **REVOKE** – odebrání určitých uživatelských práv,
- **CREATE USER** – definování a vytvoření uživatele databáze (Oppel, 2008).

2.6 Microsoft Access

Program Microsoft Access je součástí balíku programů Microsoft Office a primárně je určen pro práci s relačními databázemi. Pomocí aplikace Microsoft Access je možné uchovávat data, například kontakty, doklady, jízdní řády, seznam produktů a jiné. Dále také pracovat s danými daty, načítat, sdílet data mezi uživateli, automatizovat procesy a další. Aplikace, vytvořená v programu Microsoft Access je ideálním řešením pro malé a středně velké podniky. Tato aplikace lze dokonce propojit s Microsoft SQL Serverem a docílit tak ještě větší efektivity a lepšího zabezpečení celé aplikace. Také je možné propojit aplikaci se službou SharePoint, díky které může nad jednou databázovou tabulkou pracovat více uživatelů (Kruczek, 2010).

2.6.1 Tabulky

Tabulky jsou základní objekty databáze, které obsahují veškerá data, a proto se také při tvorbě aplikace v programu Microsoft Access vychází právě z nich. Nejjednodušším způsobem, jak v Microsoft Access stvořit tabulku je použitím funkce Návrh tabulky, kde je možné nastavit hlavní atributy tabulky.

2.6.1.1 Přehled datových typů

Datové typy určují, jaké typy hodnot je možné zadat do jednotlivých polí. Mezi základní datové typy Microsoft Access 2016 patří:

- **Krátký text** – alfanumerická data, do maximální velikosti 255 znaků.
- **Dlouhý text** – velké množství alfanumerických dat o velikosti až 64 000 znaků, ve starších verzích MS Access se tento datový typ nazýval „Memo“.
- **Číslo, Velké číslo** – ukládání pouze číselných dat.
- **Datum a čas** – ukládání dat, které jsou založeny na čase.
- **Měna** – ukládání peněžních a finančních částek, které pracují s přesností na 15 míst vlevo a na 4 místa vpravo od desetinné čárky.
- **Automatické číslo** – automaticky generovaná hodnota, zajišťující jedinečnost každého záznamu, tento datový typ se typicky používá pro primární klíč.
- **Ano/Ne** – Reprezentuje jednu z těchto dvou hodnot, jinak řečeno slouží k uložení logických hodnot.
- **Objekt OLE** – slouží k ukládání dat, které databáze nedokáže jednoznačně určit, například obrázky, tabulky Microsoft Excel, PDF soubory a jiné.
- **Hypertextový odkaz** – odkaz na jiné internetové adresy, například pomocí URL odkazu nebo cesty k fyzicky uloženému souboru.
- **Počítané pole** – ukládání výrazů, které využívají data z jiných polí.
- **Příloha** – připojení souborů, obrázků či grafů. (Kruczek, 2010; Microsoft [online], 2018)

2.6.2 Vlastnosti datových typů

Každý datový typ v programu MS Access disponuje určitými charakteristikami, které specifikují jednotlivé zadávání dat do databázových tabulek. Mezi základní vlastnosti patří:

- **Velikost pole** – u textu velikost pole určuje maximální počet znaků, které je možné do pole vložit. U číselných polí je to pak množství místa, které jednotlivé záznamy zabírají v tabulce.
- **Formát** – používá se k nastavení zobrazení jednotlivých polí.
- **Vstupní maska** – určuje způsob, jakým se do pole budou zadávat jednotlivé znaky, například rodné číslo.

- **Titulek** – popisek, který se využívá, pokud je zapotřebí pojmenovat pole jinak než v databázi.
- **Výchozí hodnota** – při vložení nového záznamu je automaticky zobrazena tato nastavená výchozí hodnota, kterou je následně možné přepsat.
- **Ověřovací pravidlo** – jedná se o výraz či vzorec, který vymezuje hodnoty, jež do pole může uživatel vložit.
- **Je nutno zadat** – při vložení nového záznamu bude MS Access striktně vyžadovat, aby bylo toto pole vyplněno.
- **Povolit nulovou délku** – určuje, jestli může pole obsahovat prázdnou hodnotu.
- **Indexovat** – díky indexům se urychluje prohledávání a řazení podle aktuálního pole, což je velmi užitečná vlastnost u rozsáhlejších tabulek.
- **Zarovnání textu** – způsob, jakým se vložená hodnota v daném poli bude zarovnávat, standardně se zarovnávat text doleva a čísla doprava. (Pecinovský, 2013)

2.6.3 Dotazy

Dotazy v programu MS Access slouží k zobrazování konkrétních údajů z jedné nebo většího počtu tabulek, popřípadě z jiných dotazů. S použitím dotazů je možné vybrat a zobrazit pouze některá pole, uspořádat je v jiném pořadí, než jsou uloženy v tabulce. Následně je také možné s vybranými poli konfigurovat různá kritéria či s nimi realizovat výpočty. Dotazy, které se vytvoří v programu MS Access, se automaticky převádí do dotazovacího jazyka SQL. SQL dotazy lze tvořit i ručně (Belko, 2014).

2.6.4 Formuláře

Formuláře tvoří uživatelské rozhraní databázové aplikace a jsou určeny pro práci s databázovými daty. Cílem formulářů je tedy usnadnění a zpřehlednění koncovým uživatelům manipulaci se záznamy. Jsou založené na datech z jedné či více tabulek nebo dotazů. Stejně jako dotazy, formuláře mohou také obsahovat výpočty. Formuláře nabízí mnoho konfiguračních možností a ovládacích prvků, díky kterým si je lze uzpůsobit dle sebe (Pecinovský, 2013).

V aplikaci MS Access existuje mnoho způsobů vytvoření hned několika typů formulářů, například formulář jednoduchý, rozdělený nebo datový list. Nejjednodušší způsob vytvoření formuláře je pomocí tlačítka Formulář, které automaticky vygeneruje formulář podle nastavení jednotlivých sloupců v tabulce. Dalším způsobem je vytvoření prázdného formuláře nebo použití takzvaného návrhu formuláře, který k vytvoření využívá průvodcovskou nabídku (Belko, 2014; Pecinovský, 2013).

2.6.5 Sestavy

Sestavy se nejčastěji využívají k tisknutelným výstupům z tabulek či dotazů. Účelem sestav je poskytnout uživateli přehlednou strukturu (návrh) tisku. V sestavách nelze data žádným způsobem upravovat. Sestavy mohou taktéž obsahovat údaje z propojených tabulek nebo vložených sestav. Způsob vytvoření sestav je podobný vytváření formulářů. Příkladem sestav mohou být faktury, objednávky, dodací listy a jiné (Belko, 2014).

2.7 Visual Basic for Applications

Visual Basic for Applications (VBA) je programovací jazyk od společnosti Microsoft, který je součástí balíku Microsoft Office. Jedná se o procedurální programovací jazyk, který je strukturovaný. Vychází z jazyka Visual Basic. Základními prvky jazyka jsou moduly, do kterých se zaznamenávají jednotlivé procedury. Procedura je kód, kterým je definovaná požadovaná akce. Daná procedura se skládá z příkazů a metod. V aplikaci Microsoft Access je Visual Basic for Applications velmi užitečný, hlavně při tvorbě rozsáhlejších a složitějších aplikací, využívá se zde například během tvorby formulářů a vlastních maker. Nevýhodou tohoto jazyka je fakt, že nejde vytvořit samospustitelná aplikace (Microsoft [online], 2018; Shepher, 2012).

2.8 XML

XML (Extensible Markup Language) je značkovací jazyk, který dovoluje vytvářet vlastní značky. Jazyk je určen převážně pro výměnu informací a komunikaci, nezávislou na konkrétní aplikaci či platformě. Výhodou tohoto jazyka je jeho nezávislost, podpora národních kódování a jednoduchý převod do různých formátů (Kosek, 2000).

V Microsoft Access může být jazyk XML využit například pro tvorbu a následnou konfiguraci vlastních ribbonových tlačítek. Ribbon je v programu MS Access známý jako horní navigační pás či panel, ve kterém se nachází navigační tlačítka. Dále lze XML například využít k importu jednotlivých dat do MS Accessu.

3 Analýza současného stavu a požadavků

Hlavním předpokladem pro návrh a implementaci databázové aplikace je analýza současného stavu a požadavků.

3.1 Charakteristika podniku

Podnik, pro který je bakalářská práce zpracovávána, byl založen roku 2010 v Ostravě. Jak již bylo zmíněno v úvodu, jedná se o výrobní firmu se zaměřením na zámečnictví a drátěný program. Specializuje se převážně na výrobu železných drátěných konstrukcí jako jsou: koše, stojany, regály, držáky a jiné. Její sídlo se nachází v Ostravě-Vítkovicích. Firma má celkem patnáct zaměstnanců a několik nestálých brigádníků, pracujících na dohodu o provedení práce.

3.2 Analýza současného stavu

Jedním z nezbytných úkolů, aby mohla být databázová aplikace vůbec realizována, musí být provedena analýza výchozího stavu, respektive jaká je aktuální situace ve firmě.

Nyní ve firmě neexistuje žádná elektronická databáze, vše se vede v papírové podobě. Firma využívá pouze aplikaci na správu pracovních směn.

Kvůli rostoucímu počtu zakázek a nepřehlednému vedení veškeré administrativy v papírové podobě tedy vznikla potřeba vést záznamy v elektronické podobě. Proto majitel, který je zároveň i provozní firmy, rozhodl o vytvoření databázové aplikace, která by tuto skutečnost naplnila.

Firma má zakoupen balík Microsoft Office 365, proto se provozní firmy rozhodl pro realizaci databázové aplikace v programu Microsoft Access, který je součástí tohoto balíku.

3.3 Požadavky pro tvorbu databázové aplikace

Analýzou současného stavu bylo zjištěno, že shromažďování informací o klientele a surovinách je neefektivní, až nedostačující. Z tohoto důvodu bude zhotovena databázová aplikace, která zajistí celkově efektivnější řešení.

Všechny záznamy budou vedeny přehledněji, a především dojde k urychlení některých firemních procesů. Zároveň bude také díky databázové aplikaci celkově zlehčena administrativní zátěž. Požadavky byly konzultovány přímo na pracovišti s provozním firmou. Aplikace bude sloužit především provoznímu této firmě.

Databázová aplikace by měla splňovat následující specifické požadavky:

- Přehled o všech odběratelích, dodavatelích.
- Evidence materiálu a výrobků.
- Zobrazitelné záznamy o objednávkách.
- Uložení do PDF nebo tisk seznamu všech dodavatelů, odběratelů, výrobků.
- Uložení do PDF nebo tisk vydaných faktur z konkrétních objednávek.
- Grafické intuitivní a přehledné rozhraní.

Do budoucna se počítá se zdokonalováním a poskytováním podpory pro tuto realizovanou aplikaci.

4 Návrh a implementace databázové aplikace

V této kapitole bude rozebrán návrh a následná implementace samotné databázové aplikace. V první části je zobrazen konceptuální návrh databáze, který zachycuje obecnou databázovou strukturu.

Druhá část kapitoly bude zaměřena na fyzický návrh databáze, který znázorňuje konečné uzpůsobení jednotlivých tabulek, včetně primárních a cizích klíčů, společně s propojovacími vazbami.

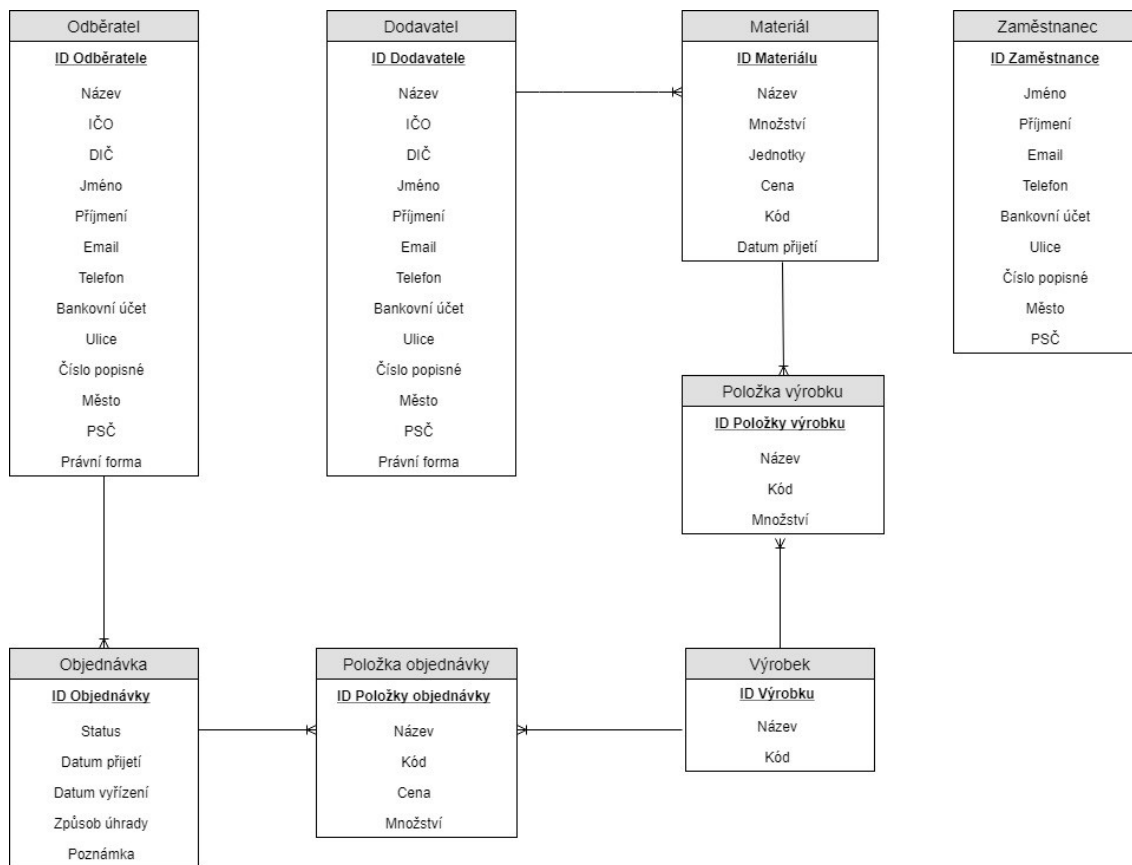
Ve třetí části této kapitoly budou detailně rozebrány jednotlivé tabulky, které byly vytvořeny v programu Microsoft Access.

Další, čtvrtá část kapitoly bude věnována konkrétní implementaci databázové aplikace, budou zde charakterizovány konkrétní formuláře a sestavy, ve kterých je daná funkční logika doprogramována pomocí jazyka Visual Basic for Applications.

V poslední části kapitoly bude popsáno zabezpečení této realizované databázové aplikace. Samotná aplikace pak bude transformována do uživatelského prostředí, ve kterém uživatel nebude mít přístup k jednotlivých databázovým položkám a programovým objektům. Takto vytvořená aplikace bude nakonec uložena na přenosné médium.

4.1 Konceptuální návrh databáze

Konceptuální návrh databáze byl navrhnout podle požadavků provozního firmy. Díky konceptuálnímu návrhu byly stanoveny základní entity, společně s jejich atributy a také vzájemné vztahy (relace) jednotlivých entit.

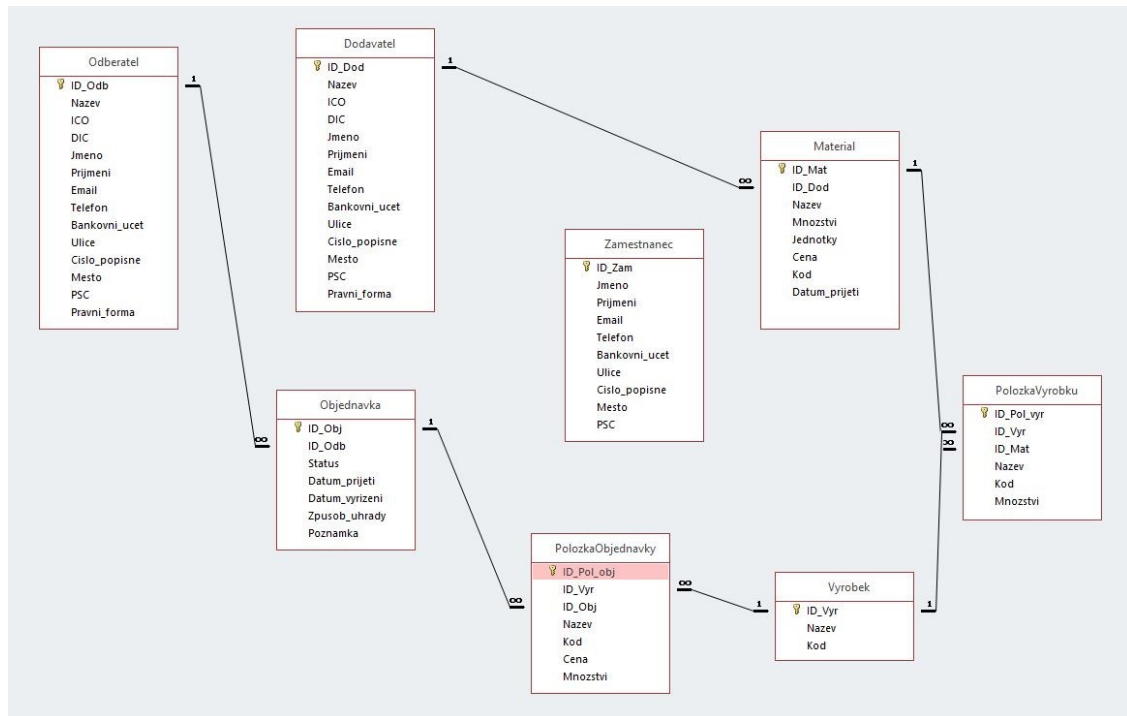


Obrázek 4.1: Konceptuální návrh databáze, zdroj: vlastní.

Obrázek 4.1 zobrazuje konceptuální návrh databáze, na kterém je možné vidět všechny entity s jejich atributy a vzájemnými vztahy. Dále je možné v každé entitě vidět názvy konkrétních atributů, společně s jejich jednoznačným identifikátorem, tedy primárním klíčem. Všechny entity, kromě entity *Zaměstnanec*, která nemá žádnou vazbu, jsou propojeny vztahem (relací) 1:N.

4.2 Fyzický návrh databáze

Fyzický návrh databáze vychází z konceptuálního návrhu databáze. Jedná se již o konkrétní implementaci systému řízení báze dat. Fyzický model databáze je tvořen celkem osmi tabulkami.



Obrázek 4.2: Fyzický návrh databáze, zdroj: vlastní.

Na obrázku 4.2 je znázorněn fyzický model databáze, který již odpovídá finální podobě jednotlivých tabulek v programu Microsoft Access. K Propojení těchto tabulek bylo využito primárních a cizích klíčů, například tabulka *Dodavatel*, která má vazbu s tabulkou *Material* prostřednictvím cizího klíče *ID_Dod*, který je umístěn v tabulce *Material*.

4.3 Návrh jednotlivých tabulek

Další a zároveň jednou z nejdůležitějších částí návrhu databázové aplikace v programu Microsoft Access je správný návrh jednotlivých tabulek. V této kapitole je popsán návrh všech databázových tabulek, ke každé z nich je vždy přiřazena textová tabulka, která jasně znázorňuje kritéria a vlastnosti dané tabulky.

4.3.1 Tabulka Dodavatel

Tabulka *Dodavatel* slouží k uložení jednotlivých dodavatelů, kteří pro firmu dodávají materiál, například železné dráty, plechové pláty, železné panty, jekly a jiné.

Tabulka se skládá celkem ze čtrnácti polí (sloupců). Primární klíč je zde nastaven na prvním poli *ID_Dod*. Datovým typem tohoto pole je automatické číslo, které se vždy generuje při přidání dalšího dodavatele a je tedy vždy jedinečné. Některé z polí mají nastavenou vstupní masku, to znamená, že lze zadat jen určité množství číslic či písmen. Konkrétní vstupní masku, společně se specifickými vlastnostmi je možné vidět v tabulce 4.1.

Dalším polem je *Nazev*, tedy název obchodní společnosti, vyplňuje se jen v případě, pokud je dodavatel právnická osoba. Osmimístné pole *ICO*, složené pouze z číslic, slouží k uložení identifikačního čísla osob. Pole *DIC* se skládá ze dvou písmen a deseti volitelných číslic, slouží k ukládání daňového identifikačního čísla. Sloupce (pole) *Jmeno* a *Prijmeni* se vyplňují jen v případě, že je dodavatel fyzická osoba.

Dalšími poli jsou *Telefon* a *Bankovni_ucet*, tyto pole disponují nastavenou vstupní maskou viz tabulka 4.1. Do pole *Telefon* se ukládá telefonní číslo s příslušnou předvolbou. Pole *Bankovni_ucet*, slouží k uložení čísla bankovního účtu konkrétního dodavatele.

Pole *Pravni_forma* uživatel explicitně nevyplňuje, vyplňuje se totiž automaticky, při přidávání nového dodavatele a následném zvolení právní formy podnikání. Slouží tedy pouze k uložení informace o dodavateli, zda se jedná o fyzickou či právnickou osobu.

Do ostatních polí: *Email*, *Ulice*, *Cislo_popisne*, *Mesto*, *PSC* se ukládají kontaktní údaje konkrétního dodavatele.

Tabulka 4.1: Dodavatel, zdroj: vlastní.

| Klíč | Název pole | Datový typ | Velikost pole | Vstupní maska | Indexovat |
|------|------------|-------------------|----------------|---------------|---------------|
| PK | ID_Dod | Automatické číslo | Dlouhé celé č. | - | Bez duplicity |
| | Nazev | Krátký text | 50 | - | Ne |

| | | | | | |
|--|---------------|-------------|----|--------------------------------|----|
| | ICO | Krátký text | 8 | 00000000 | Ne |
| | DIC | Krátký text | 12 | LL9999999999 | Ne |
| | Jmeno | Krátký text | 30 | - | Ne |
| | Prijmeni | Krátký text | 30 | - | Ne |
| | Email | Krátký text | 50 | - | Ne |
| | Telefon | Krátký text | 12 | "+ "000\ 000\ 000\ 000 | Ne |
| | Bankovni_ucet | Krátký text | 20 | 000000\ - 00000000000\ 0000 | Ne |
| | Ulice | Krátký text | 50 | - | Ne |
| | Cislo_popisne | Krátký text | 10 | - | Ne |
| | Mesto | Krátký text | 30 | - | Ne |
| | PSC | Krátký text | 5 | 000\ 00 | Ne |
| | Pravni_forma | Krátký text | 30 | - | Ne |

4.3.2 Tabulka Odberatel

Tabulka *Odberatel* slouží k uložení jednotlivých zákazníků (odběratelů), kterým firma, pro kterou je databázová aplikace vyhotovována, dodává hotové výrobky.

Tabulka *Odberatel* se od tabulky *Dodavatel*, popsané výše, liší pouze v názvu primárního klíče, jinak je návrh tabulky stejný. Zde se primární klíč nazývá *ID_Odb* a jedná se taktéž o stejný datový typ – automatické číslo.

Tabulka 4.2: *Odberatel*, zdroj: vlastní.

| Klíč | Název pole | Datový typ | Velikost pole | Vstupní maska | Indexovat |
|------|------------|-------------------|----------------|---------------|---------------|
| PK | ID_Odb | Automatické číslo | Dlouhé celé č. | - | Bez duplicity |
| | Nazev | Krátký text | 50 | - | Ne |
| | ICO | Krátký text | 8 | 00000000 | Ne |
| | DIC | Krátký text | 12 | LL9999999999 | Ne |
| | Jmeno | Krátký text | 30 | - | Ne |

| | | | | | |
|--|---------------|-------------|----|--------------------------------|----|
| | | | | | |
| | Prijmeni | Krátký text | 30 | - | Ne |
| | Email | Krátký text | 50 | - | Ne |
| | Telefon | Krátký text | 12 | "+"000\ 000\ 000\ 000 | Ne |
| | Bankovni_ucet | Krátký text | 20 | 000000\ - 00000000000\ 0000 | Ne |
| | Ulice | Krátký text | 50 | - | Ne |
| | Cislo_popisne | Krátký text | 10 | - | Ne |
| | Mesto | Krátký text | 30 | - | Ne |
| | PSC | Krátký text | 5 | 000\ 00 | Ne |
| | Pravni_forma | Krátký text | 30 | - | Ne |

4.3.3 Tabulka Material

Tabulka *Material* slouží k ukládání materiálu, konkrétně železných trubek, drátů, plátů a jiného materiálu. Zároveň tato tabulka zachycuje množství dodaného materiálu od konkrétního dodavatele a stav aktuálního materiálu, který je aktuálně dostupný na skladě.

Tabulka *Material* je složena z osmi polí, kde první pole *ID_Mat* je primárním klíčem tabulky, a jak je možné vidět v tabulce 4.3, je opět datového typu automatické číslo.

Druhým polem v pořadí je *ID_Dod*, jedná se o totožné pole jako v tabulce *Dodavatel*, s tím rozdílem, že zde představuje cizí klíč, je to tedy identifikátor dodavatele, který dodal daný materiál. Sloupec (pole) je indexován s duplicitou a jeho hodnota nesmí být prázdná.

Pole *Nazev* slouží k uložení názvu materiálu, který může dosáhnout maximálně padesáti znaků. Do pole *Mnozstvi* se ukládá množství materiálu. Pole *Jednotky* slouží k uložení specifických jednotek materiálu, například ks, m, l a další. Do pole *Cena* se ukládá celková cena materiálu, včetně DPH. Pole *Kod* slouží k uložení kódu konkrétního materiálu. Poslední pole *Datum_prijeti* slouží k uložení datumu přijetí dodávky materiálu.

Tabulka 4.3: Material, zdroj: vlastní.

| Klíč | Název pole | Datový typ | Velikost pole | Vstupní maska | Indexovat |
|------|---------------|-------------------|----------------|---------------|--------------------|
| PK | ID_Mat | Automatické číslo | Dlouhé celé č. | - | Bez duplicity |
| FK | ID_Dod | Číslo | Dlouhé celé č. | - | Duplicita povolena |
| | Nazev | Krátký text | 50 | - | Ne |
| | Mnozstvi | Číslo | Dlouhé celé č. | - | Ne |
| | Jednotky | Krátký text | 10 | - | Ne |
| | Cena | Měna | Měna | - | Ne |
| | Kod | Krátký text | 20 | - | Ne |
| | Datum_prijeti | Datum a čas | Datum | - | Ne |

4.3.4 Tabulka Objednavka

Tabulka *Objednavka* slouží k uložení všech uskutečněných objednávek, ať už se jedná o objednávky nové, vyřízené či zrušené.

Tabulka je složena ze sedmi polí. Prvním polem, které se nachází v tabulce *Objednavka* je pole *ID_Obj*, toto pole je také datového typu automatické číslo, jako všechny ostatní primární klíče v databázi. Druhé pole *ID_Odb* je cizí klíč a zároveň také identifikátor, díky kterému je možné přesně určit o jakého odběratele dané objednávky jde.

Pole *Status* slouží k uložení aktuálního stavu objednávky, v rámci databázové aplikace jsou k dispozici tři stavy objednávky: nová, vyřízená a zrušená, díky těmto stavům je možné objednávky přehledně třídit.

Pole *Datum_prijeti* slouží k ukládání data přijetí nové objednávky. Další datumové pole *Datum_vyrizeni* se vyplní pouze v případě, že je objednávka označena jako vyřízená.

Do sloupce *Zpusob_uhrady* se ukládá způsob úhrady dané objednávky, který se vybírá při přidávání nové objednávky. Poslední pole *Poznamka* slouží k uložení poznámky k dílčí objednávce.

Tabulka 4.4: Objednavka, zdroj: vlastní.

| Klíč | Název pole | Datový typ | Velikost pole | Vstupní maska | Indexovat |
|------|----------------|-------------------|----------------|---------------|--------------------|
| PK | ID_Obj | Automatické číslo | Dlouhé celé č. | - | Bez duplicity |
| FK | ID_Odb | Číslo | Dlouhé celé č. | - | Duplicita povolena |
| | Status | Krátký text | 30 | - | Ne |
| | Datum_prijeti | Datum a čas | Datum | - | Ne |
| | Datum_vyrizeni | Datum a čas | Datum | - | Ne |
| | Zpusob_uhrady | Krátký text | 30 | - | Ne |
| | Poznamka | Dlouhý text | - | - | Ne |

4.3.5 Tabulka PolozkaObjednavky

Tabulka *PolozkaObjednavky* zachycuje všechny položky, každé konkrétní objednávky. Jinak řečeno, každá objednávka se vždy skládá z objednaných položek, tedy výrobků.

V tabulce se nachází sedm sloupců. Sloupec *ID_Pol_obj* je primární klíč s datovým typem automatické číslo. Pole *ID_Vyr* a *ID_Obj* jsou cizí klíče, které určují spojitost s konkrétním výrobkem či objednávkou a položkou objednávky.

Pole *Nazev*, *Kod* slouží k uložení názvu položky objednávky (výrobku) a kódu položky (výrobku). Tyto dva sloupce (pole) slouží především k ulehčení následné práce s podformuláři. Hodnoty těchto sloupců jsou předávány do tabulky prostřednictvím SQL dotazu, explicitně je tedy uživatel nemusí zadávat.

Sloupec *Cena* slouží k uložení ceny za jednotku výrobku bez DPH, součástí této ceny je i práce. Posledním polem je pole *Mnozstvi*, toto pole zachycuje množství dané položky (výrobku).

Tabulka 4.5: PolozkaObjednavky, zdroj: vlastní.

| Klíč | Název pole | Datový typ | Velikost pole | Vstupní maska | Indexovat |
|------|------------|-------------------|----------------|---------------|--------------------|
| PK | ID_Pol_obj | Automatické číslo | Dlouhé celé č. | - | Bez duplicity |
| FK | ID_Vyr | Číslo | Dlouhé celé č. | - | Duplicita povolena |

| | | | | | |
|----|----------|-------------|----------------|---|--------------------|
| FK | ID_Obj | Číslo | Dlouhé celé č. | - | Duplicita povolena |
| | Nazev | Krátký text | 50 | - | Ne |
| | Kod | Krátký text | 20 | - | Ne |
| | Cena | Měna | Měna | - | Ne |
| | Mnozstvi | Číslo | Dlouhé celé č. | - | Ne |

4.3.6 Tabulka Vyrobek

Tabulka *Vyrobek* slouží k uchovávání údajů o jednotlivých výrobcích.

Tabulka *Vyrobek* se skládá pouze ze tří sloupců. Prvním sloupem je sloupec *ID_Vyr*, který je primárním klíčem. Do sloupce *Nazev* se ukládá název konkrétního výrobku. Do posledního sloupce *Kod* se ukládá kód určitého výrobku.

Tabulka 4.6: *Vyrobek*, zdroj: vlastní.

| Klíč | Název pole | Datový typ | Velikost pole | Vstupní maska | Indexovat |
|------|------------|-------------------|----------------|---------------|---------------|
| PK | ID_Vyr | Automatické číslo | Dlouhé celé č. | - | Bez duplicity |
| | Nazev | Krátký text | 50 | - | Ne |
| | Kod | Krátký text | 20 | - | Ne |

4.3.7 Tabulka PolozkaVyrobu

Tabulka *PolozkaVyrobu* slouží k uložení všech položek (materiálu) jednoho konkrétního výrobku, výrobek se totiž skládá z určitého množství a druhu materiálu.

Struktura této tabulky je realizována podobně jako u tabulky *PolozkaObjednavky*. V tabulce *PolozkaVyrobu* je celkem šest polí. První pole *ID_Pol_vyr* je primární klíč, je tedy generováno automaticky při přidání nové položky, jako ve všech případech této databáze.

Sloupce *ID_Vyr* a *ID_Mat* jsou cizí klíče, to znamená, že *ID_Vyr* určuje, o jaký výrobek se přesně jedná, a díky tomu lze propojit položku s daným

výrobkem. *ID_Mat* určuje materiál a je tedy možno propojit daný materiál s konkrétní položkou.

Pole *Nazev* a *Kod* slouží k uložení názvu a kódu daného materiálu. Tyto pole slouží stejně jako v tabulce *PolozkaObjednavky*, také pro ulehčení práce s podformuláři a uživatel je tedy nemusí explicitně zadávat.

Do zbývajících pole *Mnozstvi* se ukládá zadané množství položky, respektive materiálu.

Tabulka 4.7: *PolozkaVyrobku*, zdroj: vlastní.

| Klíč | Název pole | Datový typ | Velikost pole | Vstupní maska | Indexovat |
|------|------------|-------------------|----------------|---------------|--------------------|
| PK | ID_Pol_vyr | Automatické číslo | Dlouhé celé č. | - | Bez duplicity |
| FK | ID_Vyr | Číslo | Dlouhé celé č. | - | Duplicita povolena |
| FK | ID_Mat | Číslo | Dlouhé celé č. | - | Duplicita povolena |
| | Nazev | Krátký text | 50 | - | Ne |
| | Kod | Krátký text | 20 | - | Ne |
| | Mnozstvi | Číslo | Dlouhé celé č. | - | Ne |

4.3.8 Tabulka Zamestnanec

Tabulka *Zamestnanec* slouží k uložení údajů o zaměstnancích firmy, pro kterou je databázová aplikace tvořena. Tato tabulka nemá žádnou vazbu na jinou tabulku v databázi. Slouží pouze k uložení již zmíněných údajů o zaměstnancích.

Tabulka *Zamestnanec* je složena z deseti polí. Primární klíč je zde nastaven u pole *ID_Zam*.

Pole *Jmeno*, *Prijmeni*, *Email* slouží k uložení jména, příjmení a emailu konkrétního zaměstnance. Pole *Telefon* slouží k uložení osobního telefonního čísla zaměstnance. Do sloupce *Bankovni_ucet* se ukládá číslo bankovního účtu zaměstnance, který může například sloužit k výplatě mezd. Zbylá pole: *Ulice*, *Cislo_popisne*, *Mesto*, *PSC* slouží k uložení poštovní adresy daného zaměstnance.

Tabulka 4.8: Zamestnanec, zdroj: vlastní.

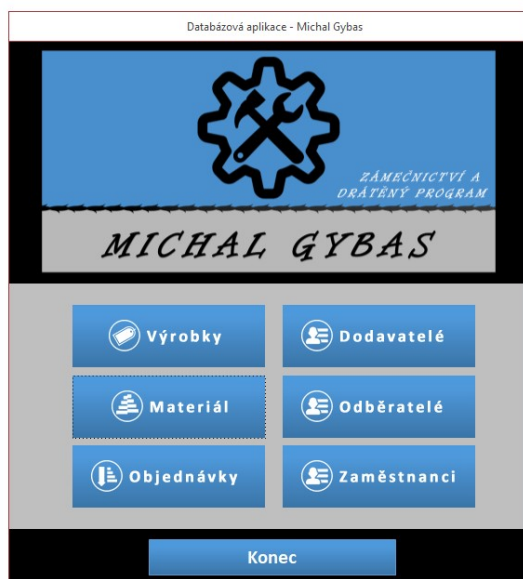
| Klíč | Název pole | Datový typ | Velikost pole | Vstupní maska | Indexovat |
|------|---------------|-------------------|----------------|--------------------------------|---------------|
| PK | ID_Zam | Automatické číslo | Dlouhé celé č. | - | Bez duplicity |
| | Jmeno | Krátký text | 30 | - | Ne |
| | Prijmeni | Krátký text | 30 | - | Ne |
| | Email | Krátký text | 50 | - | Ne |
| | Telefon | Krátký text | 12 | "+ "000\ 000\ 000\ 000 | Ne |
| | Bankovni_ucet | Krátký text | 20 | 000000\ - 00000000000\ 0000 | Ne |
| | Ulice | Krátký text | 50 | - | Ne |
| | Cislo_popisne | Krátký text | 10 | - | Ne |
| | Mesto | Krátký text | 30 | - | Ne |
| | PSC | Krátký text | 5 | 000\ 00 | Ne |

4.4 Formuláře a sestavy

Tato kapitola popisuje návrh základních formulářů a sestav, které byly vytvořeny podle požadavků vedoucího firmy. Samotný design a uživatelské rozhraní aplikace bylo koncipováno také podle zadavatele.

4.4.1 Úvodní formulář

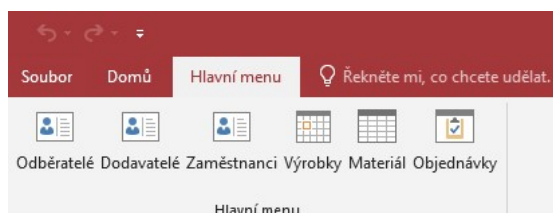
Úvodní formulář *frmUvod*, se zobrazí ihned po spuštění databázové aplikace, je ho možné vidět na obrázku 4.3. Tento formulář nabízí hlavní navigační nabídku, která slouží jako přepínač mezi jednotlivými formuláři a zároveň slouží i k ukončení samotné aplikace.



Obrázek 4.3: Hlavní nabídka databázové aplikace, zdroj: vlastní.

Další možností, jak otevřít jednotlivé formuláře je pás karet tedy *Ribbon* viz obrázek 4.4. Na ribbonu se nachází karta *Hlavní menu*, která obsahuje navigační tlačítka, jenž odkazují na konkrétní formuláře. Díky této nabídce je pak přepínání mezi formuláři jednodušší.

Tato navigační nabídka byla vytvořena pomocí XML kódu, který byl vložen přes formulář do skryté tabulky *USysRibbons*.



Obrázek 4.4: Navigační tlačítka jednotlivých formulářů na ribbonu, zdroj: vlastní.

4.4.2 Formulář Výrobky

Formulář *Výrobky*, který má v MS Access interní název *frmVyrobek*, nabízí přehled všech výrobků, které firma vyrábí. Na tomto formuláři, jak je vidět na obrázku 4.5, jsou zobrazeny všechny potřebné pole, které jsou potřeba pro identifikaci určitého výrobku. Tlačítko *Přidat položku* zde slouží k přidání materiálu, ze kterého je konkrétní výrobek vyroben. V případě nutnosti, je možné položku výrobku (materiál), odstranit. Pomocí tlačítek lze přidávat nové výrobky, odstraňovat stávající a tisknout seznam všech výrobků společně s materiálem.

K rychlému výběru výrobků slouží rozbalovací nabídka *Rychlý výběr výrobku*, díky které je možné ihned vybrat konkrétní výrobek.

Mezi jednotlivými výrobky se lze pohybovat také pomocí tlačítek, které jsou umístěny na formuláři vlevo dole. Funkcionalita těchto tlačítek je uskutečněna pomocí programovacího jazyka Visual Basic for Applications. Nad těmito tlačítky lze vidět textové pole s celkovým počtem výrobků, což je realizováno pomocí funkce *Count()*. Posledním funkčním tlačítkem na formuláři je tlačítko *Zavřít*, kterým lze formulář zavřít.

The screenshot shows a web form titled 'Výrobky'. At the top, there is a 'Výrobek' section with a 'Rychlý výběr výrobku' dropdown menu and a 'Přidat výrobek' button. Below this, there are input fields for 'ID výrobku' (94), 'Název' (Železné síto), and 'Kod' (V1056). To the right of these fields are buttons for 'Odstranit výrobek', 'Vytisknout seznam výrobků', and 'Přidat položku'. Below the input fields is a table with columns 'ID', 'Název', 'Kod', and 'Množství'. The table contains two rows: one for 'Železný drát 1m' (ID 99, Kod L250, Množství 25) and one for 'Železný drát 3m' (ID 98, Kod L600, Množství 3). Below the table, there is a label 'Celkový počet výrobků:' followed by the number '5'. At the bottom, there are navigation buttons (Home, Previous, Next, End) and a 'Zavřít' button.

Obrázek 4.5: formulář *Výrobky*, zdroj: vlastní.

4.4.3 Formulář *Materiál*

Formulář *frmMaterial* zobrazuje veškerý uložený materiál. Na obrázku 4.6 je možné vidět rozložení jednotlivých tlačítek a polí, které se nachází na formuláři.

Podobně, jako tomu bylo u formuláře *Výrobky*, zde je také možné přidat nový materiál a odstranit jej. Navíc se zde ale nachází podformulář, jenž zobrazuje dodavatele, který dodal konkrétní materiál. Pomocí jazyka Visual Basic for Applications, je zde nastaveno skrývání polí *Jméno* a *Příjmení*, záleží totiž na právní formě podnikání. Pokud je dodavatel právnická osoba, zobrazí se pole *Název firmy*, v případě fyzické osoby se zobrazí pole *Jméno* a *Příjmení*.

Materiál

Materiál
Materiál na skladě
Přidat materiál

ID materiálu:

Název materiálu:

Množství:

Jednotky:

Cena celkem (vč. DPH):

Kód materiálu:

Datum přijetí:

Dodavatel tohoto materiálu:

ID dodavatele:

Název firmy:

IČO:

DIČ:

E-mail:

Telefon:

Odstranit materiál

Celkový počet druhů materiálu: 5

Obrázek 4.6: formulář Materiál, zdroj: vlastní.

Tlačítko *Materiál na skladě* slouží k otevření formuláře *frmMaterialNaSklade*, který zobrazuje přehled aktuálního množství materiálu na skladě viz obrázek 4.7. Jednotlivý materiál lze v seznamu snadněji vybrat pomocí nastavení filtru, dle dodavatele materiálu.

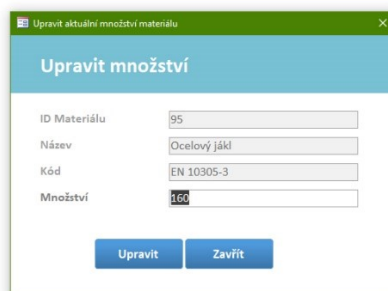
Materiál na skladě

Materiál na skladě
Filtrovat podle dodavatele
Zrušit filter

| ID Mat | Název | Cena celkem (vč. DPH) | Kód | Datum přijetí | Aktuální množství | Jednotky | |
|--------|-----------------------|-----------------------|------------|---------------|-------------------|----------|----------------------------------|
| 95 | Ocelový jákl | 17 600,00 Kč | EN 10305-3 | 12.04.2018 | 160 | ks | <input type="button" value="⌵"/> |
| 96 | Speciální lak - černý | 7 084,00 Kč | C4635215 | 11.04.2018 | 16 | l | <input type="button" value="⌵"/> |
| 97 | Železná tabule - malá | 11 160,00 Kč | P01687 | 10.04.2018 | 12 | ks | <input type="button" value="⌵"/> |
| 99 | Železný drát 1m | 28 000,00 Kč | L250 | 13.04.2018 | 200 | ks | <input type="button" value="⌵"/> |
| 98 | Železný drát 3m | 153 000,00 Kč | L600 | 13.04.2018 | 850 | ks | <input type="button" value="⌵"/> |

Obrázek 4.7: formulář Materiál na skladě, zdroj: vlastní.

K modifikaci množství konkrétního materiálu je k dispozici tlačítko, nacházející se vpravo u každé položky. Po stisknutí tohoto tlačítka se otevře další formulář, který slouží k úpravě množství materiálu, jak je patrné z obrázku 4.8.



Obrázek 4.8: formulář Upravit aktuální množství materiálu, zdroj: vlastní.

4.4.4 Formulář Odběratelé

Formulář *frmOdberatel* zachycuje přehled všech odběratelů. Pomocí tlačítek lze provádět s určitými odběrateli specifické akce viz obrázek 4.9.

Tlačítko *Upravit* slouží k upravení všech polí odběratele, kromě pole *ID Odběratele* a *Právní forma podnikání*. Zde opět záleží na právní formě podnikání odběratele. Když je odběratel právnickou osobou, zobrazí se pole *Název*, pokud je však fyzickou osobou, zobrazí se *Jméno* a *Příjmení*. Rychlý výběr odběratele slouží k vybrání konkrétního odběratele z rozbalovací nabídky. V rámci odběratelů je také možné vyhledávat podle zadaného výrazu do vyhledávacího pole. Toto vyhledávací pole funguje na principu filtrů. Vyhledávat lze podle: názvu, jména, příjmení a emailu. Je zde také možnost tisku všech odběratelů, který je realizován prostřednictvím sestavy.

Obrázek 4.9: formulář Odběratelé, zdroj: vlastní.

4.4.5 Formulář Zaměstnanci

Formulář *frmZamestnanec* se využívá k zaznamenání všech zaměstnanců. Vlastnosti a nastavení tohoto formuláře jsou stejné jako u formuláře *Odběratelé* či *Dodavatelé*, kromě specifických sloupců (polí).

4.4.6 Formulář Dodavatelé

Formulář *frmDodavatel* zobrazuje přehled všech dodavatelů. Vlastnosti tohoto formuláře jsou opět stejné jako u formuláře *Odběratelé*, zde je však navíc podformulář materiálu. Tento podformulář zachycuje, jaký materiál daný dodavatel dodal. Formulář *frmDodavatel* lze vidět na obrázku 4.10.

Obrázek 4.10: formulář Dodavatelé, zdroj: vlastní.

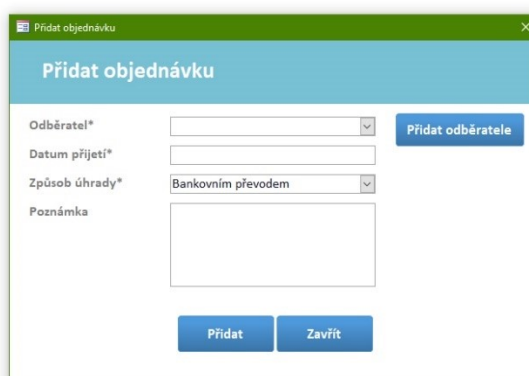
Stejně jako u formuláře *Odběratelé*, je zde také možnost přidání nového ekonomického subjektu, tedy dodavatele. Při kliknutí na tlačítko *Přidat dodavatele* se otevře formulář frmPridatDodavatel, ve kterém se zadají jednotlivé pole pro vytvoření nového dodavatele. Z přepínací nabídky je pak možné zvolit právní formu dodavatele. Při kliknutí na volbu *Právnícká osoba* dojde ke změně nabízených polí. Povinná pole jsou vždy označena hvězdičkami. Tento formulář je identický i pro odběratele. Formulář *Přidat dodavatele* je možné vidět na obrázku 4.11.

Obrázek 4.11: formulář Přidat dodavatele, zdroj: vlastní.

4.4.7 Formulář Objednávky

Formulář frmObjednavka je klíčovým formulářem, který zachycuje všechny objednávky. Jak je možné vidět na obrázku 4.13, formulář disponuje mnoha funkcemi.

Kliknutím na tlačítko *Přidat objednávku* se otevře formulář, ve kterém je možné si ze seznamu vybrat odběratele, zadat datum přijetí, způsob úhrady a poznámku k objednávce. Povinná pole, která musí být bezpodmínečně zadána, jsou označeny hvězdičkami. Z tohoto otevřeného formuláře lze také tlačítkem přímo přidat nového odběratele viz obrázek 4.12.



Obrázek 4.12: formulář Přidat novou objednávku, zdroj: vlastní.

Objednávky je možné filtrovat pomocí tří filtrů, které lze vidět na obrázku 4.13. Rozbalovací nabídkami je možno vybrat pouze aktuálně existující záznamy v databázi. Zdroj dat těchto nabídek je uskutečněn pomocí SQL příkazu SELECT.

Na formuláři se také nachází podformulář, který zobrazuje odběratele, jenž si objednávku objednal. Tento podformulář disponuje stejnou funkcionalitou skrývání polí, jako je u formuláře *Materiál*.

Formulář zobrazuje celkovou cenu objednávky, která je uskutečněna SQL dotazem:

```
"SELECT Sum (PolozkaObjednavky.Cena * PolozkaObjednavky.Mnozstvi) AS SumOfCelkova_cena, PolozkaObjednavky.ID_Obj FROM PolozkaObjednavky WHERE PolozkaObjednavky.ID_Obj = " & ID_Obj & " GROUP BY PolozkaObjednavky.ID_Obj"
```

Tlačítkem *Přidat položku* je možné přidat konkrétní výrobek k dané objednávce. V ceně za kus jednotlivého výrobku je zahrnuta i práce. Pokud není objednávka označena jako vyřízená, je možné položku objednávky (výrobek) smazat.

Tlačítko *Zrušit objednávku* je možné použít, pouze pokud je objednávka ve stavu *Nová*. Tímto tlačítkem lze objednávku zrušit, tedy stav objednávky se nastaví na: *Zrušená*.

Tlačítkem *Vyřídít objednávku* se nastaví stav objednávky na: *Vyřízená*. Společně s tímto stavem se vyplní i *Datum vyřízení*.

Obrázek 4.13: formulář *Objednávky*, zdroj: vlastní.

Tlačítko *Vytisknout fakturu* lze použít v případě, že je objednávka vyřízená. Tímto tlačítkem je možné vytisknout či uložit do PDF vydanou fakturu pomocí sestavy. Na vygenerovanou fakturu se *číslo faktury* nastavuje spojením určitého řetězce, konkrétně: $= "VF" \& Year(Now()) \& [ID_Obj]$, což je aktuální rok a ID konkrétní objednávky. Při tisku faktury záleží také na právní formě podnikání konkrétního odběratele. Pokud je odběratel fyzická osoba, otevře se sestava *sesFakturaFO*, v případě právnické osoby se zobrazí sestava *sesFakturaPO*. Datum splatnosti vydané faktury je realizováno pomocí přičtení třiceti dnů k datu vyřízení: $= [Datum_vyřízení] + Day(30)$.

4.5 Zabezpečení databázové aplikace

Databázová aplikace bude v provozu zabezpečena heslem, které bude mít k dispozici pouze vedoucí (provozní) firmy.

Záloha databáze bude uložena na Microsoft OneDrive a zajišťovat ji bude sám provozní firmy ručně. V budoucnu bude záloha databáze zautomatizována.

Databázová aplikace je pro uživatele uzpůsobena tak, aby neměli přístup k vnitřní struktuře samotné databáze. Uživatel může aplikaci ovládat pouze prostřednictvím vytvořených formulářů. Tento krok zabezpečení je zapotřebí, aby uživatel neohrozil správné fungování aplikace.

Aplikace bude v období odevzdání bakalářské práce současně testována přímo v podniku, než bude nasazena do běžného provozu.

5 Závěr

Cílem této bakalářské práce bylo navrhnout a vyvinout účelnou databázovou aplikaci, která měla splňovat následující funkce jako je evidence ekonomických subjektů, objednávek a informací spojenými s nimi, prostřednictvím softwaru Microsoft Access.

Uvedený cíl bakalářské práce byl splněn, včetně veškerých funkcí, které měla databázová aplikace plnit. Databázová aplikace naplnila veškerá očekávání zadavatele.

V první fázi probíhala konzultace s vedoucím, respektive majitelem firmy. Následně byla navrhnutá struktura databáze. Dalším krokem pak byla implementace databázové aplikace, která probíhala v programu Microsoft Access, v němž byly vytvořeny tabulky pro uložení jednotlivých dat. Následně byly vytvořeny formuláře a sestavy, ve kterých byla doprogramována funkční logika v programovacím jazyku Visual Basic for Applications. Nakonec byl dotvořen design aplikace.

Práce na databázové aplikaci tímto nekončí, po odevzdání aplikace zadavateli bude totiž probíhat ve firmě interní testování. Během tohoto testování bude aplikace doladěna do finální verze.

V budoucnu se pak počítá s vylepšováním samotné aplikace a přidáním nových funkcionalit v souladu s činností podniku. Po několika měsících provozu aplikace se bude analyzovat objem uložených dat, aby zadavatel zjistil, jestli je Microsoft Access dostatečně robustním řešením pro jeho podnikatelskou činnost.

Seznam použité literatury

Monografické publikace

BELKO, Peter. *Microsoft Access 2013: podrobná uživatelská příručka*. Brno: Computer Press, 2014. ISBN 978-80-251-4125-0.

CONOLLY, T., C. BEGG a R. HOLOWCZAK. *Mistrovství – databáze: profesionální průvodce tvorbou efektivních databází*. Brno: Computer Press, 2009. ISBN 978-802-5123-287.

KOSEK, Jiří. *XML pro každého: podrobný průvodce*. Praha: Grada, 2000. ISBN 80-7169-860-1.

KRUCZEK, Aleš. *Microsoft Access 2010: podrobná uživatelská příručka*. Brno: Computer Press, 2010. Podrobná uživatelská příručka. ISBN 978-80-251-3289-0.

OPPEL, Andrew J. *Databáze bez předchozích znalostí*. Brno: Computer Press, 2006. ISBN 80-251-1199-7.

OPPEL, Andrew J. *SQL bez předchozích znalostí: [průvodce pro samouky]*. Brno: Computer Press, 2008. ISBN 978-80-251-1707-1.

PECINOVSKÝ, Josef. *Microsoft Office 2013: podrobná uživatelská příručka*. Brno: Computer Press, 2013. ISBN 978-80-251-4102-1.

SHEPHER, Richard. *Access VBA: výukový průvodce*. Brno: Computer Press, 2012. ISBN 978-80-251-3686-7.

STEPHENS, Rod. *Beginning database design solutions*. Indianapolis, IN: Wiley Pub., 2009. Wrox beginning guides. ISBN 978-0-470-38549-4.

Internetové zdroje

BRYANT, Christian. *A Guide to Open Source Cloud Computing Software*. tomsitpro.com [online]. 12.6.2014 [cit. 11.4.2018]. Dostupné z: <http://www.tomsitpro.com/articles/open-source-cloud-computing-software,2-754-8.html>.

KATSOV, Ilya. *NoSQL Data Modeling Techniques*. Highly Scalable Blog [online]. 1.3.2012 [cit. 10.4.2018]. Dostupné z: <http://highlyscalable.wordpress.com/2012/03/01/nosql-data-modeling-techniques>.

NOVÁK, Vítězslav. *Přednášky k předmětu: Návrh databáze* [online]. 2015. [cit. 15.4.2018]. Dostupné z: https://lms.vsb.cz/pluginfile.php/298791/mod_resource/content/2/ZDB_1_Navrh.pdf.

Description of the database normalization basics [online]. Microsoft, 2018 [cit. 11.4.2018]. Dostupné z: <https://support.microsoft.com/en-us/help/283878/description-of-the-database-normalization-basics>.

Object-Relational Databases [online]. IBM, 2018 [cit. 10.4.2018]. Dostupné z: https://www.ibm.com/support/knowledgecenter/en/SSGU8G_11.50.0/com.ibm.gsg.doc/ids_gsg_416.htm.

Seznam zkratek

ANSI – American National Standards Institute

DCL – Data Control Language

DDL – Data Definition Language

DML – Data Manipulation Language

DQL – Data Query Language

FK – Foreign key

IBM – International Business Machines Corporation

ID – Identification

MS – Microsoft

OLE – Object Linking and Embedding

PDF – Portable Document Format

PK – Primary key

PSČ – Poštovní směrovací číslo

SPARC – Standards Planning and Requirements Committee

SQL – Structured Query Language

VBA – Visual Basic for Applications

XML – Extensible Markup Language

Prohlášení o využití výsledků bakalářské práce

Prohlašuji, že

- byl jsem seznámen s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo;
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB – TUO) má právo nevýdělečně ke své vnitřní potřebě bakalářskou práci užít (§ 35 odst. 3);
- souhlasím s tím, že bakalářská práce bude v elektronické podobě archivována v Ústřední knihovně VŠB-TUO a jeden výtisk bude uložen u vedoucího bakalářské práce. Souhlasím s tím, že bibliografické údaje o bakalářské práci budou zveřejněny v informačním systému VŠB-TUO;
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona;
- bylo sjednáno, že užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše);

V Ostravě 2. 5. 2018



jméno a příjmení studenta

Seznam příloh

Příloha č. 1: Ukázka seznamu všech výrobků

Příloha č. 3: Ukázka konkrétní vygenerované vydané faktury z objednávky

Příloha č. 3: CD s databázovou aplikací

Přílohy

Příloha č. 1

Faktura daňový doklad č. VF2018148

Dodavatel:

Michal Gybas
Pohraniční 832/21
70300 Ostrava

IČO: 05093121
DIČ: CZ8806265974
Telefon: +420 601 119 940
E-mail: m.gybas@gmail.com
Bankovní účet: 1634589781/3030

Odběratel:

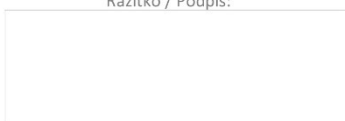
IKEA Česká republika, s.r.o.
Rudná 110
70030 Ostrava
IČO: 27081052
DIČ: CZ27081052
Telefon: +420 264 323 255
E-mail: koutny@ikea.cz

Variabilní symbol: 100148
Způsob úhrady: Bankovním převodem

Číslo objednávky 100148
Datum vystavení 19.04.2018
Datum splatnost 18.05.2018
DUZP: 19.04.2018

| Název položky | Kód výrobku | Počet | Cena bez DPH |
|---------------------|-------------|-------|--------------|
| Stojan na zboží - S | S560 | 125 | 3 200,00 Kč |
| Háček na zboží | Y109 | 30 | 430,00 Kč |

Razítko / Podpis:



Cena objednávky bez DPH: 412 900,00 Kč
DPH (21 %): 86 709,00 Kč
Celkem k úhradě včetně DPH: 499 609,00 Kč

Příloha č. 2

| Seznam všech výrobků | | | | | |
|----------------------|----------------------|-------------|--------------|-----------------------|---------------|
| ID Výrobku | Název výrobku | Kód výrobku | ID Materiálu | Název materiálu | Kód materiálu |
| 94 | Železné síto | V1056 | 98 | Železný drát 3m | L600 |
| | | | 99 | Železný drát 1m | L250 |
| | | | | | |
| 95 | Železné síto - černé | V1058 | 96 | Speciální lak - černý | C463S215 |
| | | | 98 | Železný drát 3m | L600 |
| | | | 99 | Železný drát 1m | L250 |
| 96 | Železný koš na zboží | J106 | 97 | Železná tabule - malá | P01687 |
| | | | 95 | Ocelový jákl | EN 10305-3 |
| | | | 99 | Železný drát 1m | L250 |
| 97 | Háček na zboží | Y109 | 99 | Železný drát 1m | L250 |
| | | | | | |
| 98 | Stojan na zboží - S | S560 | 98 | Železný drát 3m | L600 |
| | | | 95 | Ocelový jákl | EN 10305-3 |
| | | | 99 | Železný drát 1m | L250 |
| | | | 96 | Speciální lak - černý | C463S215 |