



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

NÁVRH DATABÁZE PRO VYBRANOU SPOLEČNOST

DATABASE DESIGN FOR A SELECTED COMPANY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Jan Bartoš

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jan Luhan, Ph.D., MSc

BRNO 2021

Zadání bakalářské práce

Ústav: Ústav informatiky
Student: **Jan Bartoš**
Studijní program: Systémové inženýrství a informatika
Studijní obor: Manažerská informatika
Vedoucí práce: **Ing. Jan Luhan, Ph.D., MSc**
Akademický rok: 2020/21

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává bakalářskou práci s názvem:

Návrh databáze pro vybranou společnost

Charakteristika problematiky úkolu:

Úvod
Cíle práce, metody a postupy zpracování
Teoretická východiska práce
Analýza současného stavu
Vlastní návrhy řešení
Závěr
Seznam použité literatury
Přílohy

Cíle, kterých má být dosaženo:

Cílem práce je identifikace klíčových vlastností v oblasti používání a funkcionality databázového systému a jeho následný návrh pro potřeby konkrétního subjektu se zaměřením na výrobní činnost pro zajištění efektivity procesů. Výstupem práce je realizovaný datový návrh ve zvolené databázové platformě.

Základní literární prameny:

CONOLLY, T., C. E. BEGG a R. HOLOWCZAK. Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází. 1. vyd. Brno: Computer Press, 2009. 584 s. ISBN 978-80-251-2328-7.

KROENKE, D., D. J. AUER a J. GONER. Databáze. 6. vyd. Brno: Computer Press, 2015. 496 s. ISBN 978-80-251-4352-0.

LACKO, L. 1001 tipů a triků pro SQL. 1. vyd. Brno: Computer Press, 2011. 416 s. ISBN 978-80-2-1-3010-0.

OPPEL, A. J. SQL bez předchozích znalostí: průvodce pro samouky. 1. vyd. Brno: Computer Press, 2008. 240 s. ISBN 978-80-251-1707-1.

STEPHENS, R. K. a R. R. PLEW. Naučte se SQL za 21 dní. 1. vyd. Brno: Computer Press, 2004. 491 s. ISBN 80-722-6870-8.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2020/21

V Brně dne 28.2.2021

L. S.

Mgr. Veronika Novotná, Ph.D.
ředitel

doc. Ing. Vojtěch Bartoš, Ph.D.
děkan

Abstrakt

Cílem této práce je návrh databáze pro vybranou společnost, která se týká tvorby výrobních štítků. V rámci této práce se budu zabývat především návrhem struktury této databáze a následně její implementací. Databáze bude realizována na platformě Microsoft SQL Server za pomoci nástrojů Microsoft Management Studia.

Klíčová slova

databáze, databázový systém, Microsoft SQL Server, SQL, ER–diagram, relace

Abstract

The aim of this thesis is to design a database for a selected company, which concerns the creation of production labels. Within this thesis I am going to concentrate on designing the structure of the database and its implementation. The database will be implemented on the Microsoft SQL Server platform using the tools of Microsoft Management Studio.

Keywords

database, database system, Microsoft SQL Server, SQL, ER-diagram, relation

Bibliografická citace

BARTOŠ, Jan. Návrh databáze pro vybranou společnost. Brno, 2021. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/135353>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Jan Luhan.

Čestné prohlášení

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

V Brně dne 16. května 2021

podpis autora

Poděkování

V první řadě bych rád poděkoval vedoucímu mé bakalářské práce panu Ing. Janu Luhanovi, Ph.D., MSc za obětavou pomoc a cenné rady při její tvorbě. Dále bych rád poděkoval panu Pavlovi Helešicovi, Jirkovi Lochmanovi a ostatním zaměstnancům ze společnosti IMI Precision Engineering za poskytnutí cenných informací k této práci. V neposlední řadě chci poděkovat své rodině a přátelům, kteří mě v průběhu tvorby práce neustále podporovali.

OBSAH

ÚVOD	11
CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ.....	12
1 TEORETICKÁ VÝCHODISKA PRÁCE	13
1.1 Data, informace a znalosti	13
1.1.1 Data	13
1.1.2 Informace	13
1.1.3 Znalosti.....	14
1.2 Databázový systém.....	14
1.2.1 Databáze	14
1.2.2 Databázové aplikace.....	15
1.2.3 Systém řízení databáze (DBMS).....	15
1.3 Datové modely.....	16
1.3.1 Lineární datový model	16
1.3.2 Objektový datový model	16
1.3.3 Relační datový model.....	17
1.3.4 Síťový datový model.....	17
1.4 Relační datový model	17
1.4.1 Relace	17
1.4.2 Integrita relačního modelu	18
1.4.3 Typy klíčů	21
1.5 Normalizace.....	21
1.5.1 První normální forma	22
1.5.2 Druhá normální forma.....	22
1.5.3 Třetí normální forma	22

1.6	Návrh databáze	22
1.6.1	Konceptuální návrh databáze	22
1.6.2	Logický návrh	23
1.6.3	Fyzický návrh.....	24
1.7	Jazyk SQL.....	25
1.7.1	Historie SQL	25
1.7.2	Kategorizace příkazů SQL	25
1.7.3	Využití SQL v praxi	26
1.7.4	Microsoft SQL Server	27
1.8	Datové typy v Microsoft SQL Serveru.....	27
1.8.1	Znakové.....	27
1.8.2	Celá čísla	27
1.8.3	Přibližná čísla	28
1.8.4	Čas.....	28
1.9	Program Codesoft.....	28
2	ANALÝZA SOUČASNÉHO STAVU	29
2.1	Představení společnosti.....	29
2.2	Formální údaje společnosti.....	30
2.3	Organizační struktura	30
2.4	Hardwarové vybavení.....	31
2.5	Softwarové vybavení	32
2.6	Analýza procesu tvorby výrobních štítků	33
2.7	Interní proces tvorby výrobního štítku	35
3	VLASTNÍ NÁVRHY ŘEŠENÍ.....	36
3.1	Požadavky na databázi.....	36

3.2	Konceptuální návrh.....	37
3.2.1	Identifikace entit.....	37
3.2.2	Identifikace vztahů mezi entitami	38
3.3	Logický návrh.....	40
3.3.1	Popisy tabulek	40
3.3.2	ER diagram.....	55
3.4	Fyzický návrh	55
3.4.1	Vytvoření tabulek.....	56
3.4.2	Propojení tabulek pomocí cizích klíčů	56
3.4.3	Naplnění tabulek testovacími daty	57
3.4.4	Propojení databáze s programem Codesoft	57
3.5	Budoucí uživatelské rozhraní	58
3.6	Předpokládané přínosy návrhu pro podnik	58
	ZÁVĚR.....	60
	SEZNAM POUŽITÝCH ZDROJŮ	61
	SEZNAM POUŽITÝCH OBRÁZKŮ	63
	SEZNAM POUŽITÝCH TABULEK	64
	SEZNAM POUŽITÝCH ZKRATEK	65
	SEZNAM PŘÍLOH.....	66

ÚVOD

Ve 21. století si téměř žádný člověk nedokáže svět představit bez informačních technologií, které nás v mnoha aspektech zcela obklopují. Tyto technologie jsou všude kolem nás. Potkáváme se s nimi v osobním životě při sledování filmů, surfování na internetu, hraní počítačových her, nebo při poslechu hudby. Tyto technologie také využíváme v práci, ale v mnoha jiných podobách a využitích. Je jasné, že s informačními technologiemi úzce souvisí i oblasti shromažďování informací a jejich evidování. Dnešní svět je postaven na databázových systémech. Jejich využití můžeme vidět v mnoha směrech od evidence obyvatel, přes školství, zdravotnictví, evidenci firemních dokumentů, až například po výzkum či tisk výrobních štítků.

Databázové systémy každodenně ulehčují fungování a provoz ve větších i menších firmách. Díky jim jsme schopni efektivněji zpracovávat vstupní i výstupní data. Společnosti tak mohou poměrně lehce zpracovávat informace o svých zákaznících, pracovat s informacemi o zboží, dodavatelích, nebo například shromažďovat data z výrobních linek, které pak mohou sloužit dalším analýzám.

Cílem této bakalářské práce bude identifikace klíčových vlastností v oblasti používání a funkcionality databázového systému se zaměřením na tvorbu výrobních štítků a následný návrh pro potřeby výrobní společnosti k zajištění zvýšení efektivity výrobních procesů.

Dílními cíli práce bude zpracování teoretických východisek, zpracování analýzy současného stavu a vlastní návrhy řešení. Výstupem práce bude realizovaný datový návrh v Microsoft SQL Serveru za pomoci nástroje Microsoft SQL Management Studio.

CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ

Prvním úkolem, pro mou práci, bylo nutné určit požadavky na databázi. K tomu jsem využil rozhovor s IT manažerem společnosti, během kterého mi bylo ukázáno a vysvětleno, jak probíhá současný proces tvorby výrobních štítků. Dále jsme hodnotili klady i zápory tohoto záměru, při čemž jsme došli k závěru, že současný proces není vyhovující zejména kvůli tomu, že se tiskne z několika Accessovských databází a bude nutné navrhnou databázovou strukturu, ze které se umožní tisknout tak, aby bylo vše zpracovatelné z jednoho místa za pomoci nástroje Microsoft SQL Management Studio. Následně jsme postupně určili, co má všechno nová databáze obsahovat, aby došlo ke zefektivnění výrobních procesů týkající se tvorby výrobních štítků.

Díky těmto požadavkům a mým záměrům jsem následně mohl sestavit analýzu současného stavu, kde je popsána samotná společnost a současný proces tvorby výrobních štítků, včetně jejich interních procesů.

Následně jsem přešel k vlastnímu řešení návrhu databáze. Bylo nutné nejdříve sestavit konceptuální návrh, kde jsou definovány objekty, se kterými budu v databázi pracovat. Na základě toho bylo potřeba určit jednotlivé entity databáze a poté zajistit spojení mezi těmito entitami. Následuje logický návrh řešení. Zde se definují jednotlivé tabulky, které databáze obsahuje. Bylo také třeba si určit jejich atributy, primární a cizí klíče, datové typy a jejich délky. Z těchto výchozích informací lze z hotových tabulek vytvořit ER diagram, podle kterého nakonec bude vytvořena databáze pomocí příkazů jazyka SQL.

Tato bakalářská práce se zabývá návrhem databáze pro tvorbu výrobních štítků. Cílem této práce je navrhnout takový databázový systém, který následně povede ke zlepšení funkcionality a efektivity procesů ve zvolené výrobní společnosti.

1 TEORETICKÁ VÝCHODISKA PRÁCE

Tato kapitola obsahuje základní teoretická východiska, která jsou stěžejní pro pochopení tématu týkajícího se databázových systémů. V této první kapitole si představíme mnoho klíčových pojmů, které jsou základem k tomu, abychom lépe porozuměli dané problematice týkající se tvorby databáze pro tisk výrobních štítků. Seznámíme se s pojmy jako jsou databáze, databázové systémy, datový model a v neposlední řadě i s jazykem SQL a programem Codesoft, který umožňuje tisk výrobních štítků.

1.1 Data, informace a znalosti

Pro porozumění problematice týkající se datového modelování je třeba chápat pojmy data, informace a znalosti (2, s. 4).

1.1.1 Data

„Surová (nezpracovaná) fakta, která mají určitou důležitost pro jednotlivce nebo organizaci.“ (1, s. 36)

Data představují fakta, text, obraz, měření, zvuk, video, které jsou spojovány nejčastěji v kontextu sledovaného procesu nebo situace. Vlastnosti dat:

- Nezávislá na uživateli – odráží současný stav reality
- Zjednodušují komplexnost reality
- Představují velký objem
- Poměrně často a rychle se mění
- Verifikovatelnost dat je možnost u většiny případů znovu objektivně ověřit správnost a přesnost dat (10, str. 21)

1.1.2 Informace

„Data, která prošla zpracováním nebo dostala strukturu, která jim dává pro jednotlivce nebo organizaci význam.“ (1, s. 36)

Aby informace mohla být považována za kvalitní, musí být srozumitelná, včasná, relevantní a přesná. Informace jsou odvozovány z dat a jsou uživateli předkládány „na míru“ (10, str. 21).

1.1.3 Znalosti

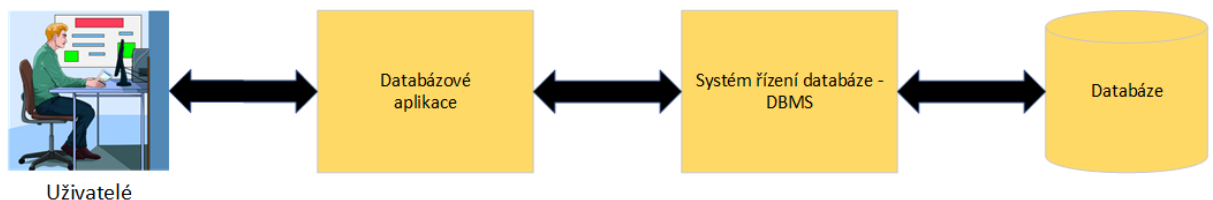
Znalosti jsou informace o tom, jak využít jiné informace a data v rozdílných situacích, které vychází z předchozí integrace s informacemi (2, s. 6).

Znalosti, je možné chápat jako to, co jednotlivec ví po osvojení si dat a informací a po jejich začlenění do souvislostí. Jsou spoluvytvářeny individuálním vzděláním a zkušenostmi experta. Je to výsledek poznávacího procesu a také předpoklad uvědomělé činnosti. Vlastnosti znalostí jsou:

- Staví na informacích
- Informace jsou parciální – představují malou část znalostí
- Oproti datům jsou méně přesné
- Zahrnují proces abstrakce, generalizace a kategorizace dat a informací (10, str. 24)

1.2 Databázový systém

Databázový systém je tvořen čtyřmi hlavními komponenty: uživatelem, databázovými aplikacemi, systémem řízení databáze (DBMS – database management systém) a vlastní databází (3, s.29).



Obrázek 1: Schéma databázového systému (Zdroj: Vlastní zpracování dle: 3, s. 30)

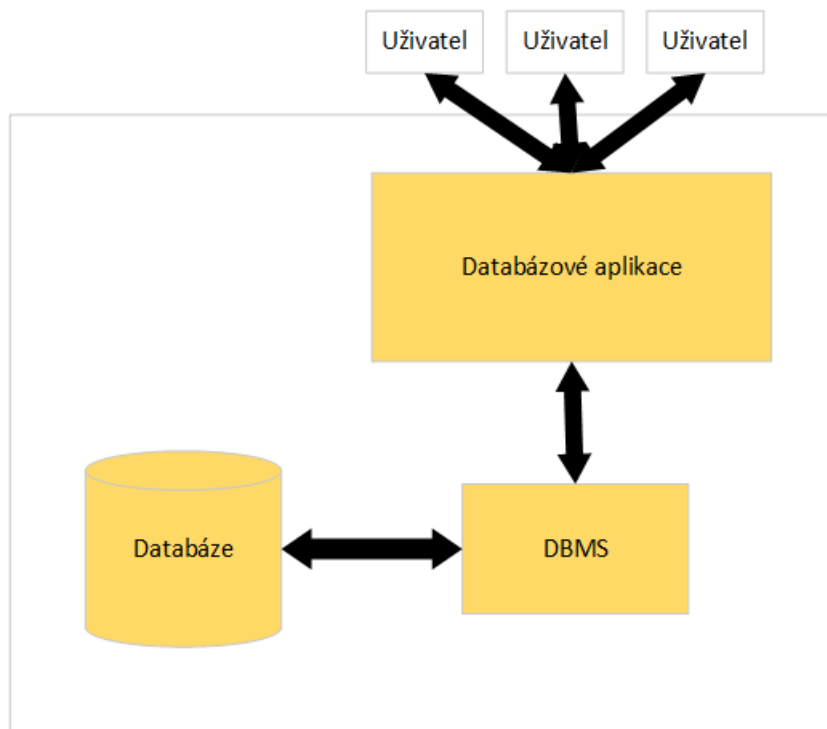
1.2.1 Databáze

Databázi si lze představit jako soubor dat, které slouží pro popis reálného světa. V databázích se můžeme setkat s pojmy jako atribut, entita, vazba mezi entitami. Entitou chápeme prvek reálného světa, který je popsán svými charakteristikami. Ty se poté označují jako atributy, tedy vlastnosti. Mezi další důležité pojmy řadíme například databázový model, primární klíč, cizí klíč a normalizace (6, s. 24).

Mimo jiné lze databázi chápat jako kolekci souvisejících záznamů obsahujících vlastní popis. U všech relačních databází se jedná o kolekci souvisejících tabulek, které obsahují vlastní popis. V této definici jsou velice podstatné dva aspekty: vlastní popis a související tabulky. Z vlastního popisu vyplývá, že popis struktury databáze je součástí samotné databáze. Související tabulky jsou propojovány společným sloupcem (3, s.30).

1.2.2 Databázové aplikace

Jedná se o počítačový program, díky němuž můžeme pracovat s databází. S databází je v kontaktu tak, že vyvolává SQL příkazy pro DBMS (1, s. 39).



Obrázek 2: Komunikace mezi uživateli a databází
(Zdroj: Vlastní zpracování dle: 1, s. 40)

1.2.3 Systém řízení databáze (DBMS)

Chápeme jako počítačový program, který je schopen vytvářet, zpracovávat a spravovat databáze. Jedná se o komplexně propracované programy, které nejsou schopny si sami vytvořit samotné firmy. Proto využívají licencovaný software, který jim poskytují specializovaní dodavatelé. DBMS přijímá požadavky v SQL jazyce a následně je převádí na aktivity databáze (3, s.29).

V dnešní době je již mnoho softwarů pro řízení databází. Mezi hlavní programy řadíme například Microsoft Access, Microsoft SQL Server, MySQL, Oracle Database (3, s. 31).

1.3 Datové modely

Všechny datové objekty, kterým může být například i člověk, reprezentují v datovém modelu určitou entitu. Pro tuto entitu musí být definovány atributy entity. Tím se nám vytváří struktura objektu, která se nazývá věta (2, str. 11).

Při tvorbě informačních systémů potřebujeme pracovat s více větnými strukturami. Typ datového objektu potřebuje navrhnout samostatnou datovou strukturu věty. V informačním systému je nutné vytvořit odpovídající obraz reality tak, aby data, která jsou v systému, odpovídala realitě (2, s. 20).

Existuje celkem pět možných typů datových modelů, které se pro práci dají využít. Ovšem dva už nemají technickou podporu ve stávajících databázových systémech. Jedná se o datové modely (2, s. 20).

- Lineární datový model
- Objektový datový model
- Relační datový model
- Síťový datový model
- Hierarchický datový model

1.3.1 Lineární datový model

Jde o datový model, který je možno implementovat na jakémkoliv médiu. Příčinou je to, že tabulky, které se nachází v tomto modelu mezi sebou nemají žádné vazby. (2, s. 20)

Klasickým příkladem lineárního datového modelu jsou karty pacientů v kartotéce u lékaře. Mezi jednotlivými kartami pacientů není žádný vztah, kromě vztahu „předchůdce“ a „následovníka“ (2. s. 21).

1.3.2 Objektový datový model

Jde o nejnovější datový model, který je vystavěný na základním prvku – objektu. Tenhle objekt má kromě svých atributů i definované metody, které určují jeho chování (2, s. 23).

Takovým objektem může být například „zkouška“ studenta. Možnými atributy tohoto objektu mohou být údaje týkající se datumu, známky, čísla studenta, termínů, předmětu a zkoušejícího. Objekt „zkouška“ má definované i své metody, tou je například „vytvoř záznam o zkoušce“, který kontroluje, zda daný posluchač již nemá tuto zkoušku hotovou (2, s. 23).

1.3.3 Relační datový model

K těm nejpoužívanějším datovým modelům patří relační datový model. Vzniká z několika lineárních modelů, které jsou spojené dohromady pomocí položky, kterou nazýváme relační klíč. Spojení není trvalé, ovšem vzniká v okamžiku, kdy potřebujeme mít společně k dispozici data ze všech tabulek. Zaniká ve chvíli, kdy práci s modulem ukončíme. Lineární modely lze využívat i samostatně (2, s. 22).

1.3.4 Síťový datový model

Síťový model je obdobou modelu hierarchického, kdy vedou pointery z rodičovského segmentu na jeho dětské segmenty, ale také obecně mezi segmenty databáze v různých směrech. V síťovém modelu již hovoříme pouze o segmentech, nikoliv o rodičích a dětech. Mezi výhody patří libovolné propojení požadovaných segmentů, a tedy i rychlý přístup k datům (2, s. 22).

1.4 Relační datový model

Jedná se o datové modely, které jsou založeny na teorii relací. Relační datové modely nám umožňují v modelu zachytit data o zkoumaných objektech, ale zachycují také vzájemné vztahy těchto objektů (2, s. 24).

1.4.1 Relace

Relace se dá definovat jako dvourozměrná tabulka, která se skládá z řádků a sloupců.

Relace má následující vlastnosti:

1. Řádky tabulky obsahují data o entitě.
2. Sloupce tabulky obsahují data, které reprezentují atributy entity.
3. Buňky v tabulce musí uchovávat jedinou hodnotu.
4. Položky ve sloupcích musí být stejného druhu.

5. Všechny sloupce v tabulce musí mít jedinečný název.
6. Na pořadí sloupců nezáleží.
7. V tabulce na pořadí řádku nezáleží.
8. Žádné dva řádky nesmí obsahovat shodné datové hodnoty (3, s. 78).

1.4.2 Integrita relačního modelu

Modelování dat nám přináší určitá omezení teoretického modelu. Integritu modelu chápeme tedy jako stav, kdy data uložená v modelu, odpovídají vlastnostem objektů reálného světa. Integritní omezení rozlišujeme na:

- Integritní omezení pro entity (relace)
- Integritní omezení pro vztahy entit (relační vazby) (2, s. 28)

Integritní omezení pro entity

a) Doménová integrita

- Snaží se zajistit, aby hodnoty daného atributu byly v množině povolených a přípustných hodnot (2, s. 28).

b) Entitní integrita

- Jedná se o integritu, která definuje, že každá relace má primární klíč, který určuje daný řádek relace (2, s. 29).
- Primární klíč – Jde o množinu atributů relace. Vlastnosti primárního klíče je jeho **jednoznačnost** (neexistuje n-tice, která by měla stejné hodnoty) a to že je **minimální** (nelze vypustit žádný atribut) (2, s. 29).

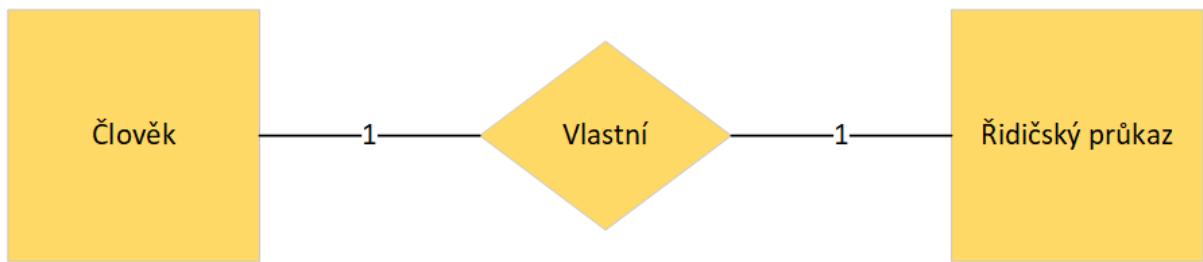
c) Referenční integrita

- Primární a cizí klíč musí mít definovanou stejnou doménu. Oba klíče si musí navzájem odpovídat, tím nám umožňují spojení mezi relacemi (2, s. 29).

Integritní omezení pro vztahy entit

a) 1:1

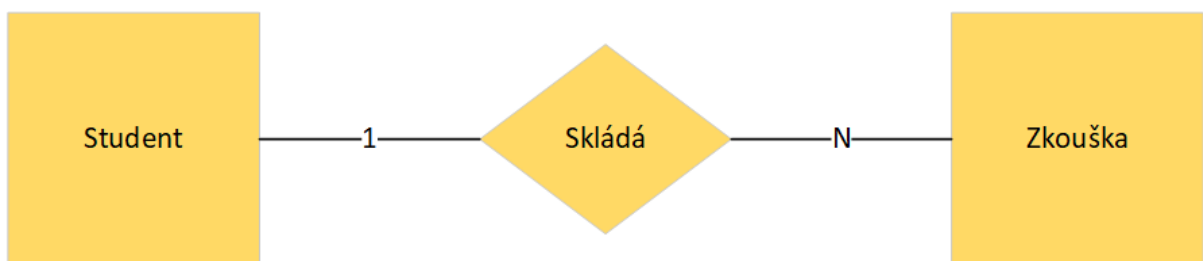
- Říká nám, že vždy jedné n-tici relace odpovídá jedna n-tice další relace (2, s. 31).
- Příkladem můžou být datové objekty „člověk“ a „řidičský průkaz“. Jeden člověk může mít pouze jeden nebo žádný řidičský průkaz (2, s. 32).



Obrázek 3: Vztah 1:1 (Zdroj: Vlastní zpracování dle: 2, s. 32)

b) 1:N

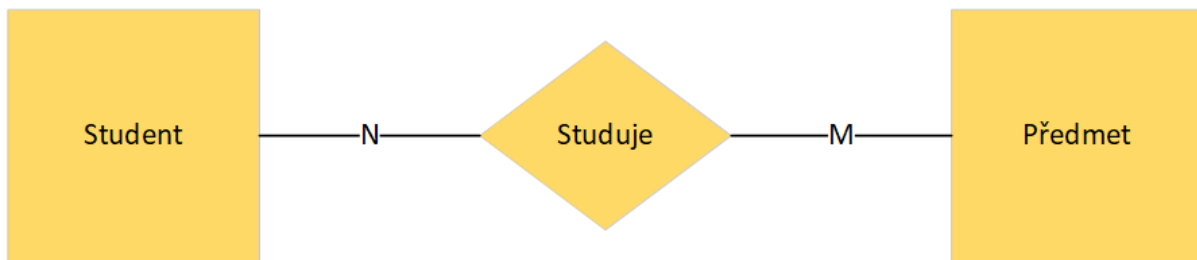
- Říká nám, že vždy jedné n-tici relace odpovídá jedna nebo více n-tic relace (2, s. 32).
- Příklad můžeme najít mezi vztahem entit „student“ a „zkouška“. Jeden student vykonává více zkoušek, přitom jedna konkrétní zkouška přísluší jednomu konkrétnímu studentovi (2, s. 33).



Obrázek 4: Vztah 1:N (Zdroj: Vlastní zpracování dle: 2, s. 32)

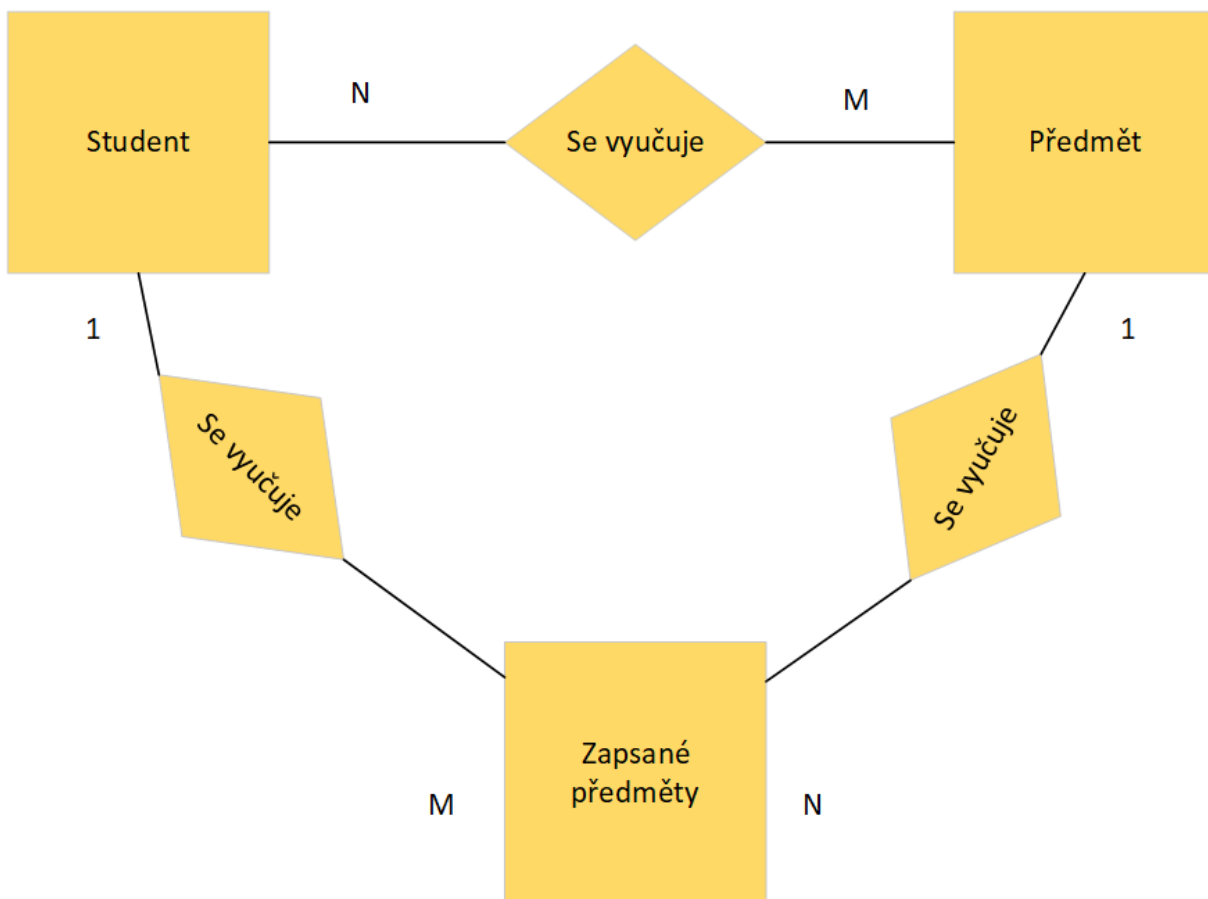
c) N:M

- Říká nám, že obecně několika n-ticím relace odpovídá jedna nebo více n-tic jiné relace (2, s. 33).
- Příklad můžeme najít mezi vztahem entit „student“ a „předmět“. Jeden student si vybere více předmětů, ale rovněž jeden předmět navštěvuje také více studentů (2, s.34).



Obrázek 5: Vztah M:N (Zdroj: Vlastní zpracování dle: 2, s. 33)

- U vztahu N:M je nutné provést tak zvanou dekompozici vztahu N:M na vztah 1:N a 1:M. To znamená, že řešení vazby N:M spočívá ve vytvoření dané entity, která má primární klíč složený z obou primárních klíčů původních entit (2, s. 35).



Obrázek 6: Dekompozice (Zdroj: Vlastní zpracování dle: 2, s. 35)

1.4.3 Typy klíčů

V tabulce musí být každý záznam jedinečný a také mu musíme tuto jedinečnost zajistit. K tomu slouží klíče relace. Máme tři typy klíčů: kandidátní klíč, primární klíč, cizí klíč (1, s. 65).

Kandidátní klíč

„Superklíč, který obsahuje jen minimální počet sloupců nutných k jedinečné identifikaci záznamů“ (1, s. 66)

Primární klíč

„Kandidátní klíč, který je vybrán tak, aby jedinečně určoval záznamy v tabulce.“ (1, s. 67)

Cizí klíč

„Sloupec nebo skupina sloupců v jedné tabulce, která odpovídá kandidátnímu klíči některé tabulky“ (1, s. 67)

1.5 Normalizace

Jde o činnost, při které upravujeme návrhy datových struktur tak, aby splňovaly zvolené normalizační úrovně. Tyto normalizační úrovně vycházejí z požadavku na efektivní ukládání dat a minimalizující se redundance při zachování integrity a konsistence dat (2, s. 55).

V průběhu normalizace je nutné určit, jestli jednotlivé tabulky databáze vyhovují jednotlivým normálním formám. V současné době je známo šest druhů normálních forem:

- 1) První normální forma
- 2) Druhá normální forma
- 3) Třetí normální forma
- 4) Boyce – Coddova normální forma
- 5) Čtvrtá normální forma
- 6) Pátá normální forma

Nejvíce jsou využívány první tři normální formy (1, s. 188).

1.5.1 První normální forma

Je kriticky důležitá pro vytvoření správných tabulek pro relační databáze. Je také tabulkou, v níž každý průsečík sloupce a záznamu obsahuje jen jedinou hodnotu (1, s. 191).

1.5.2 Druhá normální forma

Je tabulkou, která je součástí první normální formy a ve které jsou hodnoty každého sloupce, který není součástí primárního klíče. Je rovněž determinován všemi hodnotami sloupců, které tvoří primární klíč (1, s. 192).

1.5.3 Třetí normální forma

Je tabulkou, která je součástí první normální formy a druhé normální formy a ve které hodnoty ve sloupcích, které nepatří k primárnímu klíči, jsou determinovány pouze sloupci primárního klíče a nejsou determinovány jinými sloupci tabulky (1, s. 195).

1.6 Návrh databáze

Návrh databáze se rozděluje do tří hlavních fází:

- Konceptuální návrh databáze,
- Logický návrh databáze,
- Fyzický návrh databáze. (1, s. 206)

1.6.1 Konceptuální návrh databáze

„Proces vytvoření modelu dat používaných v organizaci bez jakýchkoli úvah o fyzické implementaci.“ (1, s. 206)

Vytváří se ER model, který představuje úplnou a přesnou reprezentaci datových požadavků společnosti, kterou má databáze podporovat (1, s. 207).

Pro vytvoření ER modelu je nutné splnit následující kroky, a to i jejich cíle:

- 1) Identifikace entit – Identifikovat požadované entity (1, s. 210).

- 2) Identifikace relací – Identifikovat důležité relace, které existují mezi entitami (1, s. 211).
- 3) Identifikace atributů a spojení atributů s entitami nebo relacemi – Spojit atributy s odpovídajícími entitami nebo relacemi (1, s. 214).
- 4) Určení domén atributů – Určit domény pro atributy z ER modelu (1, s. 218).
- 5) Určení atributů, které budou kandidátními, primárními a alternativními klíči – Určit kandidátní klíče a pokud existuje více než jeden kandidátní klíč, vybrat mezi nimi primární klíč a určit ostatní jako alternativní (1, s. 219).
- 6) Specializace entit – Určit entity nadtříd a podtříd, pokud je to vhodné (1, s. 220).
- 7) Kontrola redundance v modelu – Zkontrolovat přítomnost redundance v ER modelu (1, s. 220).
- 8) Kontrola, zda model podporuje uživatelské transakce. Zjistit, aby ER model podporoval požadované transakce (1, s. 221).
- 9) Posouzení konceptuálního návrhu databáze s uživateli – Posoudit ER model spolu s uživateli, aby se zjistilo, že model správnou reprezentací datových požadavků organizace, kterou má databáze podporovat (1, s. 221).

1.6.2 Logický návrh

„Proces vytvoření modelu dat používaných organizací, který je založen na specifickém modelu dat, ale nezávislý na konkrétním DBMS a jiných úvahách o fyzické implementaci.“ (1, s. 206)

ER model se převede do množiny relačních tabulek. Struktury tabulek se zkontrolují pomocí normalizace, aby se minimalizovala redundance. Rovněž se definují integritní omezení databáze (1, s. 207).

Logický návrh musí splnit následující kroky a do toho i jejich cíle:

- 1) Vytvoření tabulek – Vytvořit podle ER modelu množinu tabulek. (1, s. 232).
- 2) Kontrola struktury tabulek pomocí normalizace. – Zkontrolovat pomocí normalizace, že každá tabulka má vhodnou strukturu. (1, s. 246).
- 3) Kontrola, zda tabulky podporují uživatelské transakce. – Ujistit se, že tabulky podporují požadované transakce. (1, s. 246).

- 4) Kontrola integritních omezení. – Zkontrolovat, zda jsou v logickém návrhu databáze reprezentována integritní omezení. (1, s. 247).
- 5) Posouzení logického návrhu databáze s uživateli. – Zajistit, aby logický návrh databáze byl skutečně reprezentací datových požadavků organizace, kterou umí databáze podporovat. (1, s. 250).

1.6.3 Fyzický návrh

„Proces vytvoření popisku implementace databáze ve vnější paměti; popisuje podkladové tabulky, organizaci souborů, indexy používané pro dosažení efektivního přístupu k datům, všechna související integritní omezení a bezpečnostní omezení.“ (1, s. 207)

Logický návrh musí splnit následující kroky a do toho i jejich cíle:

- 1) Převod logického návrhu databáze do cílového DBMS – Vytvořit z logického návrhu databáze základ fungující databáze (1, s. 259).
- 2) Volba organizace souborů a indexů – Určit optimální organizaci souborů pro uložení podkladových tabulek a určit indexy potřebné pro dosažení přijatelné výkonnosti. (1, s. 268).
- 3) Návrh uživatelských pohledů. – Navrhnout uživatelské pohledy, které byly identifikovány během fáze sběru a analýzy požadavků životního cyklu návrhu databáze (1, s. 272).
- 4) Návrh bezpečnostních mechanismů – Navrhnout bezpečnostní opatření pro databázi podle požadavků uživatelů zjištěných ve fázi sběru a analýzy požadavků životního cyklu vývoje databázového systému (1, s. 273).
- 5) Zvážení kontrolovaného zavedení redundance – Určit, jestli zavedení kontrolované redundance uvolnění normalizačních pravidellepší výkonnost systému (1, s. 277).
- 6) Monitorování a vyladění systému v provozu – Monitorovat systém v provozu a zlepšit výkonnost systému, oprava nevhodného návrhu nebo zohlednění změněných požadavků (1, s. 279).

1.7 Jazyk SQL

SQL (Structured Query Language) se používá při komunikaci s relačními databázemi. Je nejrozšířenější jazyk, který umožňuje tvořit databázové dotazy. Dotaz lze chápat jako požadavek, který se odesílá databázi. Díky požadavku databáze žadateli zpětně poskytne odpověď (5, s. 33).

1.7.1 Historie SQL

IBM koncem 70. let vyvinula skupina výzkumníků experimentální relační databázi s názvem Systém/R, jehož součástí byl jazyk označovaný SEQUEL. Umožňoval manipulaci s daty a jejich načítání. Později byl název SEQUEL zkrácen na SQL (4, s. 37).

Největším konkurentem IBM byla společnost Oracle, která si držela prvenství mezi komerčními relačními databázemi (5, s. 37).

1.7.2 Kategorizace příkazů SQL

Příkazy SQL se dělí do kategorií. Tyto kategorie buď samostatnými jazyky, nebo jejich částmi. Všechny kategorie SQL mají stejnou základní syntaxi a pravidla. Proto je lze považovat za kategorie příkazů v rámci jediného jazyka. Kategorie jsou následující:

- Jazyk DDL (Data Definition Language),
- Jazyk DQL (Data Query Language),
- Jazyk DML (Data Manipulation Language),
- Jazyk DCL (Data Control Language),
- Příkazy řízení transakcí (5, s. 40).

Jazyk DDL

Jazyk zahrnuje příkazy SQL, jež umožňují uživatelům databáze vytvářet databázové objekty a upravovat jejich strukturu. Součástí DDL jazyka jsou příkazy SQL jako je například CREATE, ALTER a DROP (5, s. 40).

Jazyk DQL

Jedná se o důležitou součást jazyka SQL. Ovšem příkazy jazyka DQL jsou pouze založeny na jediném klíčovém slově: SELECT (5, s. 41).

Jazyk DML

Příkazy SQL, které umožňují uživatelům přidávat data do databáze, odebírat je z databáze a měnit stávající data. Jazyk DML se vyznačuje příkazy INSERT, UPDATE a DELETE (5, s. 41).

Jazyk DCL

Nabízí funkce jako je spuštění nebo vypnutí databáze. DCL sdružuje příkazy jazyka SQL s klíčovými slovy GRANT a ALTER (5, s. 41).

Příkazy řízení transakcí

Jde o sadu příkazů, kterou databázový uživatel požaduje zpracovat jako nedělitelnou jednotku. Transakce může být kompletně úspěšná nebo kompletně neúspěšná. Příkazy transakcí neodpovídají syntaxi příkazů jazyka SQL, ale mají značný vliv na jejich chování (5, s. 41).

1.7.3 Využití SQL v praxi

Za pomoci jazyka SQL může člověk pracující s databází provádět:

- Úpravy struktury databáze
- Změny nastavení zabezpečení systému
- Přidávat uživatelská oprávnění k databázím či tabulkám
- Dotazy nad tabulkami databáze
- Aktualizace obsahu tabulek (7)

Jedním z nejčastějších příkazů jazyka SQL je příkaz SELECT. Ten primárně slouží k získávání dat z databáze a vrací je uživateli. Dalšími typickými příkazy jsou:

- CREATE – slouží k vytváření objektů databáze, zejména tabulek
- INSERT – vkládá data do databáze
- UPDATE – aktualizuje data
- DELETE – smaže záznamy

Nejpopulárnějšími implementacemi jazyka SQL jsou MySQL, Oracle Database, Microsoft SQL Server, Sybase a IBM DB2 (7).

1.7.4 Microsoft SQL Server

SQL Server je produktem společnosti Microsoft a nabízí komplexní databázový systém včetně pokročilých funkcí (od analytických nástrojů po dolování dat), které jsou závislé na použité verzi (8).

1.8 Datové typy v Microsoft SQL Serveru

Datové typy v SQL Serveru můžeme rozdělit na znakové, číselné, datové a časové.

1.8.1 Znakové

Tabulka 1: Tabulka datových typů pro text (Zdroj: Vlastní zpracování dle: 12)

Název	Maximální velikost	Úložiště
CHAR	8 000 znaků	Definovaná délka
VARCHAR	8 000 znaků	2 Bytes + počet znaků
TEXT	2 GB textových dat	4 Bytes + počet znaků

1.8.2 Celá čísla

Tabulka 2: Tabulka datových typů pro celá čísla (Zdroj: Vlastní zpracování dle: 12)

Název	Rozsah	Úložiště
TINYINT	od 0 do 255	1 Bytes
SMALLINT	od -32 768 do 32 767	2 Bytes
INT	od -2 147 483 648 do 2 147 483 647	4 Bytes
BIGINT	Od -9 233 372 036 854 755 808 do -9 233 372 036 854 755 807	8 Bytes

1.8.3 Přibližná čísla

Tabulka 3: Tabulka pro přibližná čísla (Zdroj: Vlastní zpracování dle: 12)

Název	Popis	Úložiště
FLOAT	Od $-1.79E + 308$ Do $1.79E + 308$	4 nebo 8 Bytes
REAL	Od $-3.40E + 38$ Do $-3.40E + 38$	4 Bytes

1.8.4 Čas

Tabulka 4: Tabulka datových typů pro čas (Zdroj: Vlastní zpracování dle: 12)

Název	Popis	Úložiště
DATE	Od January 1, 0001 Do December 31, 9999	3 Bytes
DATETIME	Od January 1, 1753 Do December 31, 9999	8 Bytes
TIME	Ukládá čas s přesností 100 nanosekund	3-5 Bytes

1.9 Program Codesoft

Program CODESOFT umožňuje návrh a tisk etiket, obsahujících textové a číselné informace, čárové kódy a grafické objekty. Také podporuje velké množství tepelných a termotransferových tiskáren (11).

2 ANALÝZA SOUČASNÉHO STAVU

Následující kapitola představuje společnost, pro kterou bude navržen databázový systém týkající se tvorby výrobních štítků. Nejprve představím podnik samotný, tj. obecné informace jako jsou název, sídlo a majitel. Dále se bude věnovat postavení společnosti na trhu. Na konci kapitoly bude provedeno několik analýz, ze kterých vyplynou poznatky sloužící zejména pro návrhovou část této práce.

2.1 Představení společnosti

IMI Precision Engineering je výrobní podnik, který se nachází v Modřicích u Brna. Společnost je divizí nadnárodní společnosti IMI plc., která sídlí v Birminghamu. Korporace je tvořena ze tří divizí, které jsou:

- IMI Precision Engineering
- IMI Hydronic Engineering
- IMI Critical Engineering

Tato korporace zaměstnává přes 10 900 zaměstnanců ve více než 70 různých zemích po celém světě.

IMI Precision Engineering se specializuje především na výrobu a vývoj pneumatických pohonů, tlakových spínačů a dalších systémů pro řízení proudů a kapalin. Tyto výrobky můžeme najít v automobilovém, energetickém nebo chemickém průmyslu. Divize zaměstnává po celém světě 5 300 zaměstnanců a odborníků. Významné závody jsou například v Brazílii, Číně, České republice, Mexiku a Německu



Obrázek 7: Logo společnosti (Zdroj: <https://www.norgren.com/cz/cs>)

2.2 Formální údaje společnosti

Formální název:

IMI International s.r.o.

Právní forma:

Společnost s r.o.

Sídlo:

Humpolec 1573, 396 01 Humpolec

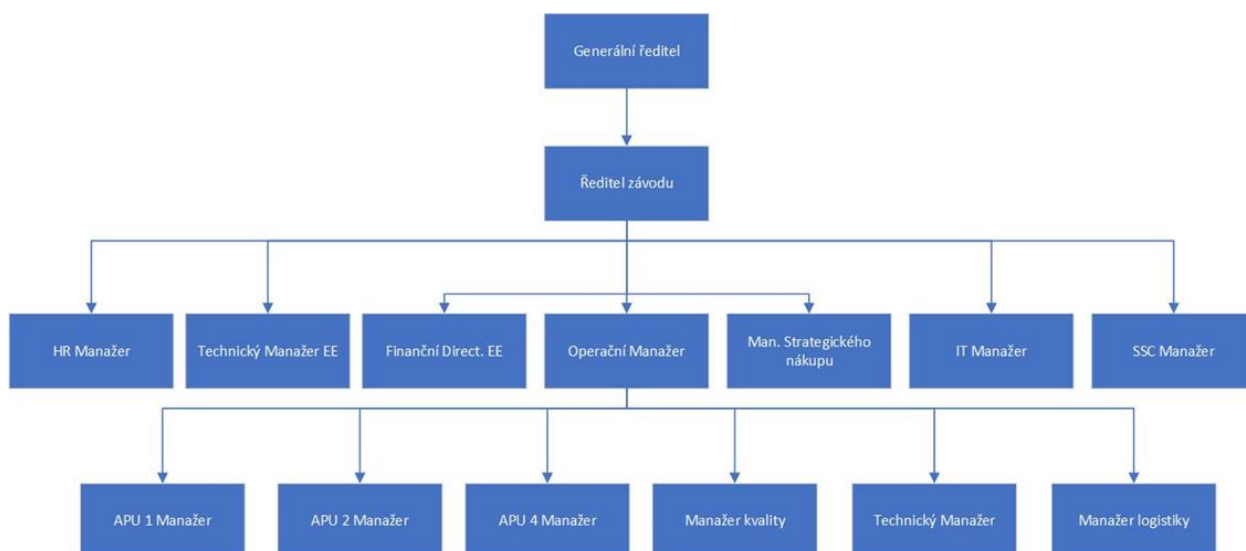
Datum vzniku:

4. září 1998

2.3 Organizační struktura

Výrobní proces se ve společnosti dělí na tři základní výrobní oblasti. Tyto oblasti se nazývají APU (Autonomous Production Unit) a jsou dále děleny podle druhu výroby. Ty se dělí na výrobní oblasti APT, pod které spadají konkrétní výrobní linky. Kromě těchto oblastí je v podniku několik podpůrných oddělení, která zabezpečují bezproblémový chod výroby ve firmě. Tyto organizační celky jsou: APU 1, APU 2, APU 4, IT, Logistika, Strategický nákup, Finance, Kvalita, Technické oddělení, Lidské zdroje, Design, Štíhlá výroba a BOZP.

V podniku je maticová organizační struktura, která se stará o plynulou výrobu a chod závodu. Díky efektivní organizaci výrobního procesu je firma schopna co nejúčelněji využívat veškeré dostupné zdroje, a to jak humanitní, tak nehumanitní. Problémem v takových komplexních organizačních strukturách je často nedostatečná komunikace mezi jednotlivými odděleními, popřípadě ztráta přímé vize cílů podniku.



Obrázek 8: Organizační struktura společnosti
(Zdroj: Vlastní zpracování)

2.4 Hardwarové vybavení

Ve firmě, zejména v administrativní činnosti, zaměstnanci pracují na svém pracovním laptopu. Zaměstnanci, kteří potřebují výkonnější laptop, jedná se především o designéry, technology, manažery atd., využívají ke své práci Dell Precision 15. Ostatní zaměstnanci potom využívají Dell Latitude 14. Všichni uživatelé laptopů mají u svého pracovního místa také k dispozici dokovací stanici Dell WD19 + monitory Dell P2419H.

Na výrobě jsou využívány především stolní počítače, a to konkrétně Dell Optiplex 3070. Ty slouží k mnoha účelům. Například jsou propojeny k mikroskopům, pro kontrolu kvality výrobku. Nebo po přidání COM portu měří určité hodnoty daných strojů na linkách. Také slouží jako počítače u tiskových center.

V tiskových centrech na výrobě a na pracovištích je několik druhů tiskáren. Pro výrobu se primárně tisknou výrobní štítky. Jsou jimi například termotransférové tiskárny Zebra ZM600 nebo Toshiba B-EX4. V kancelářích tisk zajišťuje plynulou práci. Nejčastěji využívanými tiskárnami jsou Konica Minolta Bizhub 227 a HP LaserJet Pro M28a.

Jako své firemní telefony dostávají zaměstnanci v administrativní činnosti, supervizoři a team leadeři mobilní telefon iPhone 7.

2.5 Softwarové vybavení

Firma využívá Enterprise resource planning systém ORACLE JD Edwards. Tento warehouse management systém propojuje nakládání s materiálem, výrobu, plánování, oceňování, strategický nákup, finance a účetnictví. Také je zde propojení s dalšími pobočkami a závody IMI po celém světě. Systém umožňuje nastavení rolí pro různé skupiny uživatelů s odlišnými možnostmi a pravomocemi. Není žádoucí, aby každý uživatel měl možnost přesouvat materiál nebo plánovat výrobu. Z tohoto důvodu existuje například role Planner pro plánovače výroby a další uživatele, kteří potřebují mít přístup k datům bez možnosti je měnit. Mezi další role patří Buyer pro nákupčí nebo Inter pro logistiku.

Druhým používaným systémem je německý systém LOSSY (Logistik Organisation Steuerung SYsteme). Jedná se o logistickou nástavbu pro JD Edwards. Slouží primárně pro logistiku a sklad k evidenci, řízení a vychystávání materiálu.

Zaměstnanci mimo jiné především v administrativní činnosti využívají programy dle svých pracovních rolí. Designéři, 3D modeláři a konstruktéři pracují v Autodeskových programech jako je AutoCAD, Inventor, Vault, DWG TrueView. Kvalitáři využívají například statistický software Minitab. Ve výrobě pracují zaměstnanci především s programy, které měří hodnoty u výrobních linek, sbírají data nebo zde jsou například aplikace k mikroskopům pro kontrolory kvality výrobku.

Počítače a notebooky fungují na operačním systému Windows 10 Enterprise a v rámci kancelářského softwaru mají všichni administrativní pracovníci nainstalovaný balík Microsoft 365, kde jsou programy jako Word, Excel, Outlook, Access a OneDrive. Firma má rovněž licenci na antivirový program McAfee, který je na všech zařízeních.

2.6 Analýza procesu tvorby výrobních štítků

Firma tiskne výrobní štítky, které jsou primárně určeny na výrobky a krabice. Štítky se tisknou v tzv. tiskových centrech. Existuje zde několik tiskových center. Každé z těchto center je v několika ohledech rozdílné. Po každé se na výrobu transferují z odlišných míst a jsou pro různé typy výrobků. Musí se brát také v úvahu, že tyto výrobky se liší různými hodnotami (tlak, síla, zdvih válce atd.).

Společnost pracuje s několika systémy jako je např. AutoCAD, Vault apod., ovšem data o štítcích v těchto programech nejsou obsaženy. Ty se nacházejí v jednotlivých výkresech. Proto je nutné data shromáždit a udělat z nich databázi. Databáze obsahují data o jednotlivých part numbers. K výrobku jsou tři různé part numbers a to k JDE (systémový), ten který se tiskne a zákaznický part number. Obsahuje také typy štítků. Ty mohou být kovové, patřící na výrobky, nebo papírové, které se lepí na sáčky. Někdy se tiskne na papír s logem. To v případě, že přijdou na firemní výrobky. Bez loga se tiskne, když společnost výrobek distribuuje do oblasti automobilu, nebo také může tisknout logo dané firmy, která distribuuje finální produkt. Také obsahuje hodnoty, kdy například tlak se udává v barech, ale když jde odyt například do Japonska, tak se musí uvádět v jiných hodnotách (PSI). Je nutné taky nastavovat správné množství štítků, které je možno maximálně vytisknout. Někdy je hodnota fixní, ale někdy proměnné. Také je nutné rozlišovat, jestli se balí do velkých nebo malých krabic. Na štítcích se taky využívají date kódy, které se taky ovšem liší v tom, jaký má firma a jaký zákazník.

Nyní tiskové centrum běží na 6 různých Accessových databázích. Je nutné navrhnout databázovou strukturu, ze které se umožní tisknout tak, aby bylo vše zpracovatelné z jednoho místa. Taky je nutné rozeznávat i např. vlastnosti jednotlivých výrobků (kde byly vyrobeny, hodnoty tlaku apod.). Pro fyzický návrh databáze budu volit program Microsoft SQL Server.



Obrázek 9: Ukázka výrobního štítku
(Zdroj: Převezato od IMI Precision Engineering)



Obrázek 10: Ukázka výrobního štítku
(Zdroj: Převezato od IMI Precision Engineering)

2.7 Interní proces tvorby výrobního štítku

Samotný interní proces tvorby výrobního štítku začíná ve chvíli, kdy si zákazník objedná výrobek. Toto může provést dvěma způsoby. Buď se výrobek objednává přímo z webu (centrálního skladu), nebo za pomoci sales organizace. Po objednání výrobku přijde společnosti systémová objednávka. Zboží je buď skladem a lze jej ihned poslat, nebo skladem není a je třeba jej vyrobit. V tomto případě plánovač výroby naplánuje výrobu, vytvořením zakázky, kde mimo procesu výroby samotného výrobku je třeba k němu vydat i samotný štítek. Plánovač předá informace o produktu zaměstnancům, kteří pracují v tiskovém centru. Ti ve webovém rozhraní zadají part number konkrétního výrobku a následně mají přístup ke Codesoft šablonám, které přijdou do tisku. Jakmile jsou štítky vytištěny mohou být umístěny na dané výrobky.

3 VLASTNÍ NÁVRHY ŘEŠENÍ

V následující kapitole představím hlavní část mé bakalářské práce. Na prvním místě je podstatné si ujasnit, které významné aspekty by měla databáze obsahovat a jak by měla celkově vypadat.

V rámci konceptuálního řešení databáze je nutné definovat objekty, se kterými budu v rámci své práce pracovat. Díky tomu lze určit entity, které se v databázi vyskytnou a dá se navázat na logický návrh, kde definuji jednotlivé její atributy, klíče, datové typy a jejich délky. Následně lze z hotových tabulek vytvořit ER diagram.

Na závěr můžu přistoupit k tvorbě fyzického návrhu databáze, který budu vytvářet v programu Microsoft SQL Server Management Studio 18.

3.1 Požadavky na databázi

Mým úkolem bude vytvoření centrální databáze, která umožní na každé tiskové lokaci tisknout právě z této centrální databáze. Nyní se na tiskových centrech data tisknou z šesti Accessovských databází. Na prvním místě je tedy nutné určit v čem se dané databáze podobají a v čem se ba naopak liší.

Když vyhodnotím dané tabulky a jejich jednotlivé atributy, tak docházím k závěru, že jejich struktura je odlišná především v tom, že jednotlivé výrobky, na které se potom štítky lepí, mají rozdílné vlastnosti. U některých výrobků jsou důležité údaje o tlaku, zdvihu, průměru apod. U jiných jsou třeba podstatné informace o teplotě, tlaku vzduchu, či velikosti napětí. Následně u některých výrobních štítků se uvádí různé atributy, například zákaznicko logo, zákaznický part number, stát kde byl výrobek vyroben nebo třeba ISO date.

Databáze jsou jinak podobné v tom, že každý výrobek má svůj part number a pomocí part numberu se dají zjistit jednotlivé informace o štítku. Detaily o štítku obsahují atributy s informacemi o samotném štítku, jeho šabloně, vzorkovému obrázku a pro koho se štítek tiskne.

3.2 Konceptuální návrh

V rámci konceptuálního návrhu je nutné si definovat objekty, se kterými budu v databázi pracovat. Na základě toho lze dále určit jednotlivé entity a dále je důležité zjistit, jaké jsou relace mezi těmito entitami.

3.2.1 Identifikace entit

V této kapitole se budu věnovat identifikaci entit. U každé entity se vždy musí určit název, její alias a dále bude jednoduše vysvětleno, k čemu entita slouží a jaký počet výskytů obsahuje.

Tabulka 5: Identifikace entit (Zdroj: Vlastní zpracování)

Název entity	Popis entity	Alias entity	Počet výskytů
Master_main	Hlavní tisková tabulka obsahující informace o tiskových lokacích		TISÍCE
Master_printed	Tabulka vytisknutých štítků		STOVKY
Master_labelsforpn	Tabulka s informacemi k tisku		STOVKY
Master_labelsdetails	Tabulka obsahující informace o štítku k tisku		STOVKY
Stanice_pristup	Přístupová tabulka		DESÍTKY
ISO_main	Hlavní tabulka pro ISO ventily	Ventily	STOVKY

FRL_main	Hlavní tabulka pro filtry, regulátory, lubrikátory	Filtry, regulátory, lubrikátory	STOVKY
HDM_main	Hlavní tabulka pro HDM válce	Válce	STOVKY
MEILLER_main	Hlavní tabulka pro MEILLER produkty		STOVKY
ACTUATORS_main	Hlavní tabulka pro pohony	Pohony	STOVKY
FITTINGS_main	Hlavní tabulka pro šroubení a trubky	Šroubení	STOVKY
ISO_vlastnosti	Tabulka vlastností ISO ventilů		STOVKY
FRL_vlastnosti	Tabulka vlastností pro filtry, regulátory, lubrikátory		STOVKY
HDM_vlastnosti	Tabulka vlastností HDM válců		STOVKY
ACTUATORS_vlastnosti	Tabulka vlastností pohonů		STOVKY
FITTINGS_vlastnosti	Tabulka vlastností pro šroubení a trubky		STOVKY

3.2.2 Identifikace vztahů mezi entitami

Tato kapitola se zabývá identifikací vztahů mezi jednotlivými entitami, které se nachází v návrhu databázového systému. Identifikaci relací zobrazuje tabulka č. 6.

Tabulka 6: Identifikace vztahů mezi entitami (Zdroj: Vlastní zpracování)

Název entity	Vztah	Název entity
Master_main	Obsahuje	Master_labelsforpn
Master_main	Obsahuje	ISO_main
Master_main	Obsahuje	FRL_main
Master_main	Obsahuje	HDM_main
Master_main	Obsahuje	MEILLER_main
Master_main	Obsahuje	ACTUATORS_main
Master_main	Obsahuje	FITTINGS_main
Master_main	Zobrazuje	Master_printed
Master_main	Uděluje	Stanice_pristup
Master_printed	Obsahuje	Stanice_pristup
Master_labelsforpn	Tvoří	Master_labelsdetails
ISO_main	Má	ISO_vlastnosti
ISO_main	Má přidělené	ISO_isocode
FRL_main	Má	FRL_vlastnosti
HDM_main	Má	HDM_vlastnosti
ACTUATORS_main	Má	ACTUATORS_vlastnosti
FITTINGS_main	Má	FITTINGS_vlastnosti

3.3 Logický návrh

Díky konceptuálnímu řešení můžu nyní přejít na logický návrh. Zde budu definovat jednotlivé tabulky, které databáze obsahuje. Určím její atributy, primární a cizí klíče, datové typy a jejich délky. Následně z hotových tabulek vytvořím ER diagram.

3.3.1 Popisy tabulek

Cílem této kapitoly je provedení identifikace jednotlivých atributů. Krok za krokem se budu věnovat všem atributům v návrhu databázového systému a budu postupně určovat jejich název, typ a délku atributu, následně zda musí být atribut vyplněný či nikoli a jeho stručný popis. V neposlední řadě také určíme primární klíče a cizí klíče daných atributů.

3.3.1.1 Tabulka Master_main

Tato tabulka primárně slouží jako propojovací tabulka k hlavním tabulkám tiskových lokací. Důležitým atributem je *pn* (part number), který slouží jako jednoznačný identifikátor pro konkrétní výrobek, který se ve firmě vyrábí. Rovněž obsahuje atribut ohledně zákaznického part numberu, včetně part numberu, který je určen k tisku. Tabulka se váže na tabulky *Master_labelsforpn*, *Master_printed*, *Stanice_pristup ISO_main*, *FRL_main*, *HDM_main*, *MEILLER_main*, *ACTUATORS_main* a *FITTINGS_main*.

Tabulka 7: Atributy tabulky Master_main (Zdroj: Vlastní zpracování)

Název atributu	PK/ FK	Typ a délka	Not Null	Popis
ID_main	PK	INT	ANO	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
pn		VARCHAR (25)	ANO	Part number – jednoznačná identifikace položky, která se ve firmě vyrábí

pntoprint		VARCHAR (25)	NE	Part number k tisku
customerpn		VARCHAR (25)	NE	Zákaznický part number
labelsforpn_ID	FK	INT	ANO	Propojení s tabulkou Master_labelsforpn
master_pristup_ID	FK	INT	ANO	Propojení s tabulkou Stanice_pristup
printed_ID	FK	INT	ANO	Propojení s tabulkou Master_printed
ISO_main_ID	FK	INT	ANO	Propojení s tabulkou ISO_main
FRL_main_ID	FK	INT	ANO	Propojení s tabulkou FRL_main
HDM_main_ID	FK	INT	ANO	Propojení s tabulkou HDM_main
MEILLER_main_ID	FK	INT	ANO	Propojení s tabulkou MEILLER_main
ACTUATORS_main_ID	FK	INT	ANO	Propojení s tabulkou ACTUATORS_main
FITTINGS_main_ID	FK	INT	ANO	Propojení s tabulkou FITTINGS_main

3.3.1.2 Tabulka Master_printed

Tabulka *Master_printed* slouží především k tomu, aby zde bylo možné vyhledat, jaké štítky byly vytisknuty. Obsahuje atributy o part numberu výrobku, jeho číslo výrobní zakázky a zároveň datum, kdy byly tisknuty. Na tabulku se váže tabulka *Stanice_pristup*.

Tabulka 8: Atributy tabulky Master_printed (Zdroj: Vlastní zpracování)

Název atributu	PK/FK	Typ a délka	Not Null	Popis
ID_printed	PK	INT	ANO	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
pn_printed		VARCHAR (25)	ANO	Part number – jednoznačná identifikace výrobku, který se ve firmě vyrábí
wo		VARCHAR (10)	NE	Work order – číslo výrobní zakázky
date		DATE	NE	Datum
pristup_ID	FK	INT	ANO	Propojení s tabulkou Stanice_pristup

3.3.1.3 Tabulka Master_labelsforpn

V této tabulce jsou uvedeny informace ohledně tisku štítků. Obsahuje atributy o part numberu, počtu vtištěných štítků a číslu tiskárny. Tabulka je provázána s tabulkou *Master_labelsdetails*.

Tabulka 9: Atributy tabulky Master_labelsforpn (Zdroj: Vlastní zpracování)

Název atributu	PK/FK	Typ a délka	Not Null	Popis
ID_labelsforpn	PK	INT	ANP	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu

pn_labelsforpn		VARCHAR (25)	ANO	Part number – jednoznačná identifikace výrobku, který se ve firmě vyrábí
kusulbl		INT	NE	Počet vytištěných štítků
InLinePrinter		DATE	NE	Číslo tiskárny
labels_ID	FK	INT	ANO	Propojení s tabulkou Master_labelsdetails

3.3.1.4 Tabulka Master_labelsdetails

Tato tabulka obsahuje detailní informace o štítku. Konkrétně název šablony a název vzorkového obrázku a určení pro koho je štítek určený. U některých štítků se uvádí i poznámky.

Tabulka 10: Atributy tabulky Master_labelsdetails (Zdroj: Vlastní zpracování)

Název atributu	PK/FK	Typ a délka	Not Null	Popis
ID_label	PK	INT	ANO	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
template		VARCHAR (30)	NE	Název šablony
sampleimg		VARCHAR (10)	NE	Název pro vzorkový obrázek
usedfor		VARCHAR (50)	NE	Pro koho je štítek určený
mynote		VARCHAR (50)	NE	Komentář, poznámka

3.3.1.5 Tabulka Stanice_pristup

Tabulka ohledně přístupu k databázi. Obsahuje atributy o názvu počítače, jménu uživatele a jeho pracovní pozici.

Tabulka 11: Atributy tabulky Stanice_pristup (Zdroj: Vlastní zpracování)

Název atributu	PK/FK	Typ a délka	Not Null	Popis
ID_pristup	PK	INT	ANO	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
pcname		VARCHAR (10)	ANO	Název počítače
username		VARCHAR (10)	ANO	Jméno uživatele
rolename		VARCHAR (10)	ANO	Pracovní role

3.3.1.6 Tabulka ISO_main

Tabulka pro ISO ventily obsahuje atributy týkající se procesu pro tvorbu výrobních štítků. Entita je rovněž provázána s tabulkami *ISO_vlastnosti* a *ISO_isocode*.

Tabulka 12: Atributy tabulky ISO_main (Zdroj: Vlastní zpracování)

Název atributu	PK/FK	Typ a délka	Not Null	Popis
ID_ISO_main	PK	INT	ANO	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
pn_ISO		VARCHAR (25)	ANO	Part number – jednoznačná identifikace výrobku, který se ve firmě vyrábí

pn_movex		VARCHAR (25)	NE	Part number firmy Movex
skupina		VARCHAR (10)	NE	Druh skupiny výrobku
changecode		VARCHAR (1)	NE	Revize
symbol		VARCHAR (30)	NE	Název obrázkový symbol
short_code		VARCHAR (3)	NE	Číslo revize
boxlabeltxt		VARCHAR (30)	NE	Text na papírový štítek na krabičku
extratext		VARCHAR (30)	NE	Doplňkový text
extraimg		VARCHAR (30)	NE	Název doplňkového obrázku
vlastnosti_ISO	FK	INT	ANO	Propojení s tabulkou ISO_vlastnosti
isocode_ID	FK	INT	ANO	Propojení s tabulkou ISO_isocode

3.3.1.7 Tabulka ISO_vlastnosti

V této tabulce jsou uvedeny atributy, které se týkají vlastností ISO ventilů. Jedná se o hodnoty tlaku a pilotdrucku.

Tabulka 13: Atributy tabulky ISO_vlastnosti (Zdroj: Vlastní zpracování)

Název atributu	PK/FK	Typ a délka	Not Null	Popis
ID_ISO_vlastnosti	PK	INT	ANO	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
bar		VARCHAR (20)	NE	Hodnota tlaku
pilotdruck		VARCHAR (20)	NE	Hodnoty pilotdrucku

3.3.1.8 Tabulka FRL_main

Tabulka pro filtry, regulátory a lubrikátory, která obsahuje atributy týkající se procesu pro tvorbu výrobních štítků. Entita je rovněž provázána s tabulkou *FRL vlastnosti*.

Tabulka 14: Atributy tabulky FRL_main (Zdroj: Vlastní zpracování)

Název atributu	PK/FK	Typ a délka	Not Null	Popis
ID_FRL_main	PK	INT	ANO	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
pn_FRL		VARCHAR (25)	ANO	Part number – jednoznačná identifikace výrobku, který se ve firmě vyrábí
madein		VARCHAR (30)	NE	Kde byl výrobek vyroben
boxlabpn		VARCHAR (20)	NE	Part number na papírové štítky na krabičce
boxlabdesc		VARCHAR (20)	NE	Popis papírového štítku
behindslah		VARCHAR (20)	NE	Hodnota určující specialitu výrobku
logo_norgren		VARCHAR (10)	NE	Název pro Norgrenovské logo
logo_customer		VARCHAR (30)	NE	Název pro zákaznické logo
logo_atex		VARCHAR (30)	NE	Název pro Atexové logo
productimg		VARCHAR (25)	NE	Název pro produktový obrázek

customertxt		VARCHAR (35)	NE	Zákaznický text
boxlabeltxt		VARCHAR (10)	NE	Text na papírkové štítky na krabičce
eac_logo		VARCHAR (30)	NE	Název pro eac logo
vlastnosti_FRL	FK	INT	ANO	Propojení s tabulkou FRL_vlastnosti

3.3.1.9 Tabulka FRL_vlastnosti

Tato tabulka obsahuje atributy, které se týkají vlastností filtrů, regulátorů a lubrikátorů. Jedná se především o Intel, Range, Temp a Preset hodnoty.

Tabulka 15: Atributy tabulky FRL_vlastnosti (Zdroj: Vlastní zpracování)

Název atributu	PK/FK	Typ a délka	Not Null	Popis
ID_FRL_vlasnotsti	PK	INT	ANO	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
inlet_txt1		VARCHAR (10)		Hodnota MAX INLET/INLET
inlet_txt2		VARCHAR (10)		Doplňková Inlet hodnota
inlet_val1		VARCHAR (10)		Hodnota Inlet v jednotkách PSIG
inlet_val2		VARCHAR (10)		Hodnota Inlet v jednotkách bar
range_txt1		VARCHAR (10)	NE	Hodnota RANGE/PRE SET
range_txt2		VARCHAR (10)	NE	Doplňková Range hodnota

range_val1		VARCHAR (10)	NE	Hodnota Range v jednotkách PSIG
range_val2		VARCHAR (10)	NE	Hodnota Range v jednotkách bar
temp_txt1		VARCHAR (10)	NE	Hodnota MAX TEMP/TEMP
temp_txt2		VARCHAR (10)	NE	Doplňková TEMP hodnota
temp_val1		VARCHAR (20)	NE	Hodnota teploty v Celsiech
temp_val2		VARCHAR (20)	NE	Hodnota teploty ve Fahrenheitech
preset_txt		VARCHAR (10)	NE	Hodnoty PRE SET/SET
preset_val1		VARCHAR (20)	NE	Hodnota Set v PSIG
preset_val2		VARCHAR (20)	NE	Hodnota Set v barech

3.3.1.10 Tabulka HDM_main

Tabulka pro HDM válce, která obsahuje atributy týkající se procesu pro tvorbu výrobních štítků. Entita je rovněž provázána s tabulkou *HDM_vlastnosti*.

Tabulka 16: Atributy tabulky HDM_main (Zdroj: Vlastní zpracování)

Název atributu	PK/FK	Typ a délka	Not Null	Popis
ID_HDM_main	PK	INT	ANO	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu

pn_HDM		VARCHAR (25)	ANO	Part number – jednoznačná identifikace výrobku, který se ve firmě vyrábí
dcodetypeforklt		VARCHAR (20)	NE	Název konkrétního typu date kódu.
vlastnosti_HDM	FK	INT	ANO	Propojení s tabulkou HDM_vlastnosti

3.3.1.11 Tabulka HDM_vlastnosti

Tato tabulka obsahuje atributy, které se týkají vlastností HDM válců. Jde o atributy týkající se tlaku, hubu, maximálních a minimálních hodnot tlaku a také dalších vlastností produktu jako je U, W, X, Z.

Tabulka 17: Atributy tabulky HDM_vlastnosti (Zdroj: Vlastní zpracování)

Název atributu	PK/FK	Typ a délka	Not Null	Popis
ID_HDM_vlastnosti	PK	INT	ANO	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
tlak		INT	NE	Hodnota tlaku
hub		VARCHAR (10)	NE	Hodnota hubu
bild		VARCHAR (30)	NE	Název pro obrázek
Pmin		INT	NE	Minimální hodnoty tlaku
Pmax		INT	NE	Maximální hodnoty tlaku

Pvmin		INT	NE	Minimální hodnoty Pv
Pvmax		INT	NE	Maximální hodnoty Pv
U		VARCHAR (5)	NE	Hodnota U výrobku
W		VARCHAR (5)	NE	Hodnota W výrobku
X		VARCHAR (15)	NE	Hodnota X výrobku
Z		VARCHAR (15)	NE	Hodnota Z výrobku

3.3.1.12 Tabulka MEILLER_main

Tabulka MEILLER, která obsahuje atributy týkající se procesu pro tvorbu výrobních štítků.

Tabulka 18: Atributy tabulky MEILLER_main (Zdroj: Vlastní zpracování)

Název atributu	PK/ FK	Typ a délka	Not Null	Popis
ID_MEILLER_main	PK	INT	ANO	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
pn_MEILLER		VARCHAR (25)	ANO	Part number – jednoznačná identifikace výrobku, který se ve firmě vyrábí
line		VARCHAR (20)	NE	Poznámka k výrobku
packagingty		INT	NE	Balení
toplevel		INT	NE	Revize

3.3.1.13 Tabulka ACTUATORS_main

Tato tabulka je hlavní tabulkou pro pohony. Obsahuje atributy týkající se procesu pro tvorbu výrobních štítků. Na entitu je navázána tabulka *ACTUATORS_vlastnosti*.

Tabulka 19: Atributy tabulky ACTUATORS_main (Zdroj: Vlastní zpracování)

Název atributu	PK/ FK	Typ a délka	Not Null	Popis
ID_ACTUATORS_main	PK	INT	ANO	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
pn_ACTUATORS		VARCHAR (25)	ANO	Part number – jednoznačná identifikace výrobku, který se ve firmě vyrábí
cislodilu		VARCHAR (30)	NE	Part number
typ		VARCHAR (30)	NE	Typ výrobku
serie		VARCHAR (1)	NE	Typ serie výrobku
symbolpicture		VARCHAR (25)	NE	Název obrázku
symbol1		VARCHAR (25)	NE	Název pro první symbol
symbol2		VARCHAR (25)	NE	Název pro druhý symbol
customerlogo		VARCHAR (25)	NE	Název zákaznického loga
madein		VARCHAR (25)	NE	Kde byl výrobek vyroben

dcodetype		VARCHAR (10)	NE	Název konkrétního typu date kódu
poznamka1		VARCHAR (20)	NE	Poznámka na štítek
vlastnosti_ACTUATOR S	FK	INT	ANO	Propojení s tabulkou ACTUATORS_vlastno sti

3.3.1.14 Tabulka ACTUATORS_vlastnosti

Tato tabulka uvádí atributy, které se týkají vlastností pohonů. Jde o atributy týkající se průměru válce, hodnoty zdvihu, tlaku, teploty a napětí daného výrobku.

Tabulka 20: Atributy tabulky ACTUATORS_vlastnosti (Zdroj: Vlastní zpracování)

Název atributu	PK/ FK	Typ a délka	Not Null	Popis
ID_ACTUATORS_vlastnosti	PK	INT	ANO	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
prumer		INT	NE	Průměr válce
zdvih		INT	NE	Hodnota zdvihu
tlak		VARCHAR (15)	NE	Hodnota tlaku
teplota		VARCHAR (15)	NE	Hodnota teploty
napeti		VARCHAR (10)	NE	Hodnota napětí

3.3.1.15 Tabulka FITTINGS_main

Tabulka pro šroubení a trubky, která obsahuje atributy týkající se procesu pro tvorbu výrobních štítků. Entita je rovněž provázána s tabulkou *FITTINGS_vlastnosti*.

Tabulka 21: Atributy tabulky FITTINGS_main (Zdroj: Vlastní zpracování)

Název atributu	PK/ FK	Typ a délka	Not Null	Popis
ID_FITTINGS_main	PK	INT	ANO	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
pn_FITTINGS		VARCHAR (25)	ANO	Part number – jednoznačná identifikace výrobku, který se ve firmě vyrábí
symbol		VARCHAR (30)	NE	Název pro obrázkový symbol
seriál		INT	NE	Číslo serie
quantity		INT	NE	Množství v krabici
boxquantity		INT	NE	Množství krabic
description		VARCHAR (30)	NE	Doplňkový popis
vlastnosti_FITTINGS	FK	INT	ANO	Propojení s tabulkou FITTINGS_vlastnosti

3.3.1.16 Tabulka FITTINGS_vlastnosti

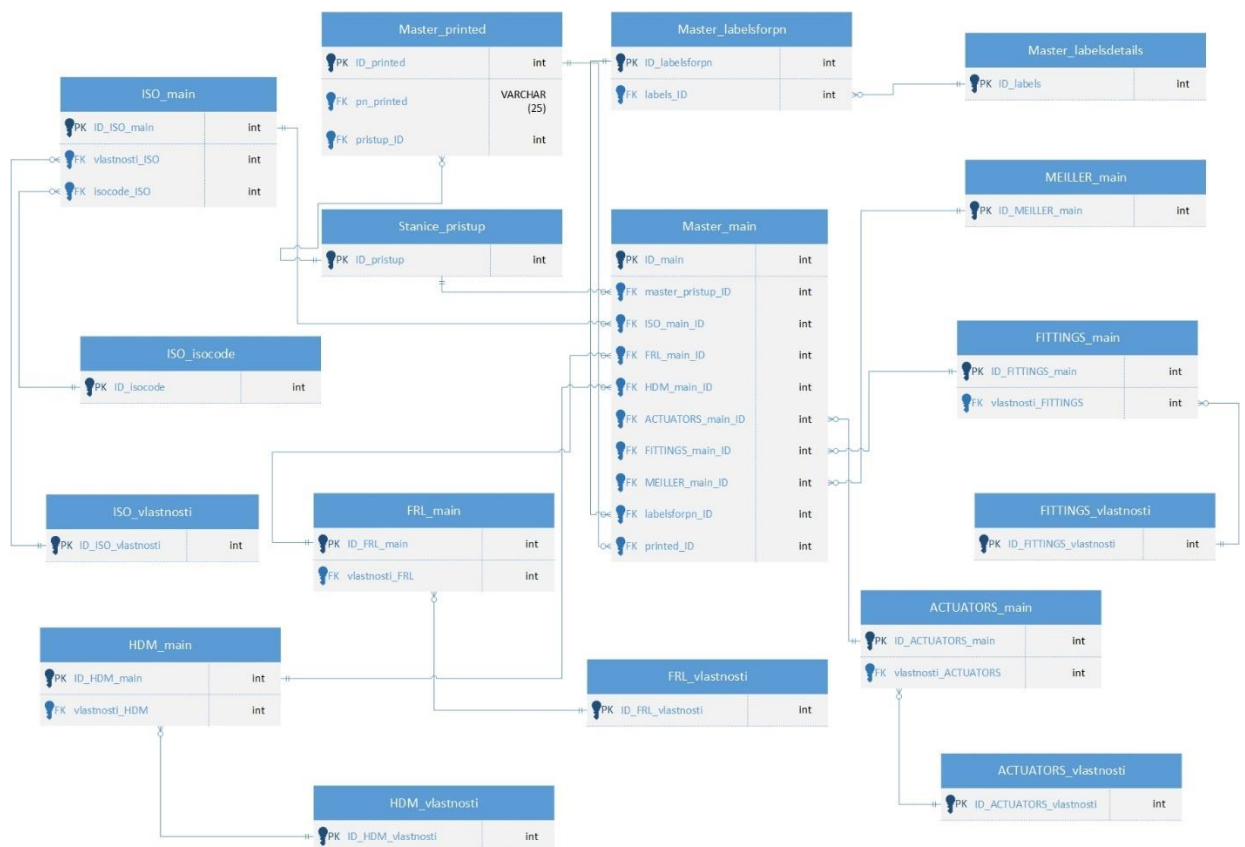
V následující tabulce jsou atributy, které se týkají vlastnosti šroubení a trubek. Jde o atributy týkající se rozměrů portu A, B, C, D, rovněž jsou zde atributy, kde jsou uvedeny hodnoty tlaku vzduchu a teploty.

Tabulka 22: Atributy tabulky FITTINGS_vlastnosti (Zdroj: Vlastní zpracování)

Název atributu	PK/ FK	Typ a délka	Not Null	Popis
ID_FITTINGS_vlastnosti	PK	INT	AN O	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
port_a		VARCHAR (10)	NE	Rozměry A portu
port_b		VARCHAR (10)	NE	Rozměry B portu
port_c		VARCHAR (10)	NE	Rozměry C portu
port_d		VARCHAR (10)	NE	Rozměry D portu
air_pressure		VARCHAR (10)	NE	Hodnota tlaku vzduchu
temp		VARCHAR (5)	NE	Hodnota teploty

3.3.2 ER diagram

V ER diagramu jsou popsány všechny tabulky navrhovaného databázového systému s označenými primárními i cizími klíči a jednotlivé vazby mezi těmito tabulkami. Všechny tabulky jsou normalizované, a to až do úrovně třetí normální formy neboli tranzitivní závislosti. Na následujícím obrázku číslo 11 je zkrácená verze ER diagramu. Diagram, kde jsou obsaženy všechny atributy v jednotlivých tabulkách jsou k nalezení v příloze.



Obrázek 11: Zkrácená verze ER diagramu (Zdroj: Vlastní zpracování)

3.4 Fyzický návrh

Základem fyzického návrhu je převedení ER – diagramu logického návrhu do formy fyzických tabulek na SQL serveru, a to včetně všech jejich vazeb a integritních omezení.

Převedení proběhne prostřednictvím dvou velice důležitých kroků. V prvním kroku je podstatné udělat script na vytvoření jednotlivých tabulek a v rámci druhého kroku tyto tabulky za pomoci cizích klíčů (FK) vzájemně propojíme. V závěr je nutné databázi naplnit testovacími daty.

3.4.1 Vytvoření tabulek

Smyslem této části je vytvořit jednotlivé tabulky, které byly ukázány a představeny v předchozích kapitolách. Ve všech tabulkách je nutné nastavit primární klíče a u jednotlivých atributů jejich typ a délku, případně omezení not null.

```
create table FRL_vlastnosti(  
ID_FRL_vlastnosti int not null identity (1,1) primary key,  
intel_txt1 varchar(10),  
intel_txt2 varchar(10),  
intel_val1 varchar(10),  
intel_val2 varchar(10),  
range_txt1 varchar(10),  
range_txt2 varchar(10),  
range_val1 varchar(10),  
range_val2 varchar(10),  
temp_txt1 varchar(10),  
temp_txt2 varchar(10),  
temp_val1 varchar(20),  
temp_val2 varchar(20),  
preset_txt varchar(10),  
preset_val1 varchar(20),  
preset_val2 varchar(20),  
);
```

Obrázek 12: Příklad kódu pro vytvoření tabulky (Zdroj: Vlastní zpracování)

3.4.2 Propojení tabulek pomocí cizích klíčů

V této kapitole se zobrazuje propojení tabulek, které byly vytvořeny v předešlé kapitole, prostřednictvím cizích klíčů. Cizí klíče není nutné přidávat do všech tabulek, ale pouze do těch, které jsou jejich prostřednictvím napojeny na další tabulky.


```

master_pristup_ID int not null constraint master_pristup_ID references Stanice_pristup (ID_pristup),
ISO_main_ID int not null constraint ISO_main_ID references ISO_main (ID_ISO_main),
FRL_main_ID int not null constraint FRL_main_ID references FRL_main (ID_FRL_main),
HDM_main int not null constraint HDM_main_ID references HDM_main (ID_HDM_main),
ACTUATORS_main_ID int not null constraint ACTUATORS_main_ID references ACTUATORS_main (ID_ACTUATORS_main),
FITTINGS_main_ID int not null constraint FITTINGS_main_ID references FITTINGS_main (ID_FITTINGS_main),
MEILLER_main_ID int not null constraint MEILLER_main_ID references MEILLER_main (ID_MEILLER_main),
labelsforpn_ID int not null constraint labelsforpn_ID references Master_labelsforpn (ID_labelsforpn),
printed_ID int not null constraint printed_ID references Master_printed (ID_printed),

```

Obrázek 13: Příklad kódu pro propojení tabulek pomocí cizích klíčů
(Zdroj: Vlastní zpracování)

3.4.3 Naplnění tabulek testovacími daty

Pro to, aby byla zajištěna funkcionálnita databáze, je nutné ji naplnit testovacími daty. K tomu slouží SQL příkaz INSERT INTO. Následující obrázek ukazuje skript pro naplnění tabulky *ISO_isocode*. Celý kód pro naplnění všech tabulek je k nalezení v příloze.

```

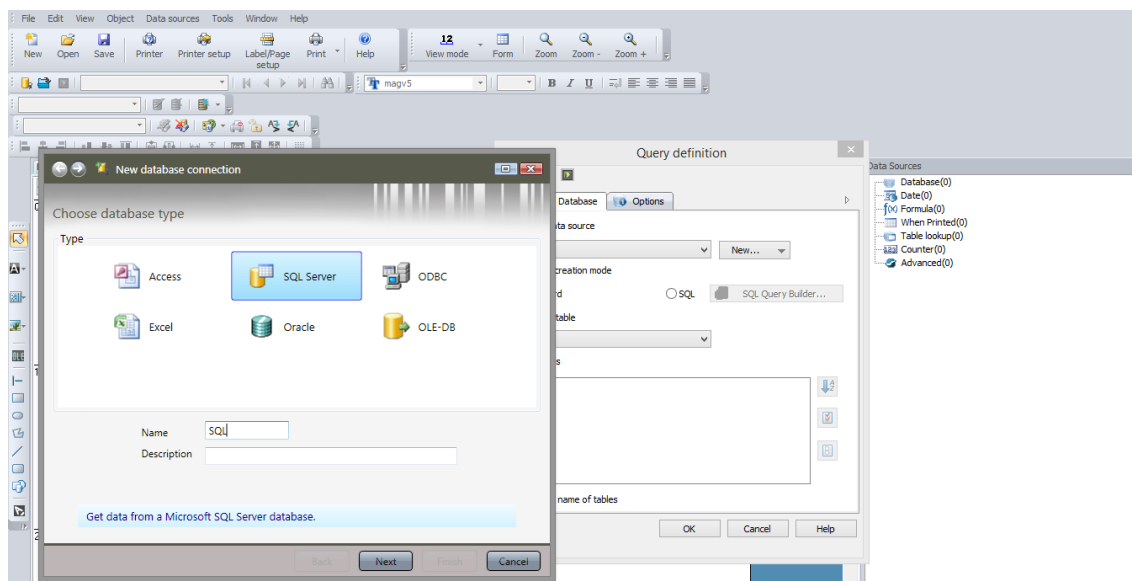
INSERT INTO ISO_isocode VALUES(12,2029,'980');
INSERT INTO ISO_isocode VALUES(11,2029,'979');
INSERT INTO ISO_isocode VALUES(10,2029,'978');
INSERT INTO ISO_isocode VALUES(9,2029,'977');
INSERT INTO ISO_isocode VALUES(8,2029,'976');
INSERT INTO ISO_isocode VALUES(7,2029,'975');
INSERT INTO ISO_isocode VALUES(6,2029,'974');
INSERT INTO ISO_isocode VALUES(5,2029,'973');
INSERT INTO ISO_isocode VALUES(4,2029,'940');
INSERT INTO ISO_isocode VALUES(3,2029,'939');
INSERT INTO ISO_isocode VALUES(2,2029,'938');
INSERT INTO ISO_isocode VALUES(1,2029,'937');

```

Obrázek 14: Příklad kódu pro naplnění tabulek testovacími daty
(Zdroj: Vlastní zpracování)

3.4.4 Propojení databáze s programem Codesoft

Codesoft umožňuje propojení mnoha různých databázových nástrojů, včetně Microsoft SQL Serveru. Program zvládne vytvářet date kódy na základě celosvětových norem, umí určovat vlastní primární klíče, vytvářet vlastní barcodes (čárové kódy, QR kódy, 2D kód, apod.), umí také slučovat variabilní pole s fixními. V programu se dají vkládat i různé grafické symboly a obrázky.



Obrázek 15: Ukázka přidání databáze do programu Codesoft
(Zdroj: Vlastní zpracování)

3.5 Budoucí uživatelské rozhraní

Společnost se do budoucna rozhodla investovat do vývoje aplikace, která bude propojena s navrhovanou databází. Aplikace by pak měla nahradit samotný program Codesoft a ulehčit tím práci u jistých administrativních procesů týkající se tisku výrobních štítků. Než bude aplikace zhotovena, navrhoval bych stále používání programu Codesoft, který bude propojen s SQL databází.

3.6 Předpokládané přínosy návrhu pro podnik

Hlavním cílem této bakalářské práce bylo vytvoření návrhu databáze, který má sloužit pro tvorbu výrobních štítků a následné propojení s programem Codesoft. Do budoucna se očekává vytvoření aplikačního rozhraní, které by dokázalo zaměstnancům usnadnit obsluhu této databáze.

Z ekonomické stránky společnost nepotřebuje nakoupit licence na Microsoft SQL Server, jelikož několik licencí již vlastní. To je určitě velká výhoda. Na druhou stranu bude potřeba investovat do vývoje aplikačního rozhraní. Nedokážu odhadnout přesnou cenu za vývoj aplikace, ale díky interní komunikaci se zaměstnancem, který na základě kvalifikovaného odhadu stanovil cenu za vývoj aplikace pro tisk výrobních štítků, by předpokládaná cena byla okolo 40 000 až 50 000 Kč. Zároveň je třeba připočítat náklady na proškolení zaměstnanců pro obsluhu zařízení v novém uživatelském prostředí. Předpokládám však, že by to bylo pro společnost velkým benefitem, jelikož roční náklady za licenci pro program Codesoft vychází firmu v přepočtu na 50 000 Kč.

Na závěr lze říct, že na základě uvedeného návrhu budou přínosy vidět zejména v oblasti administrativy. Štítky se budou moci tisknout z jedné databáze, dojde tak k urychlení a zefektivnění všech procesů týkajících se výroby štítků. Díky vytvoření nového návrhu databáze bude možné mnohem účinněji získávat aktuální údaje z databáze. Bude zde komplexní přehled jednotlivých výrobků a jejich vlastností, které se u některých výrobců tisknou na štítky.

ZÁVĚR

Hlavním cílem mé bakalářské práce bylo vytvoření návrhu databázového systému pro tvorbu výrobních štítků pro společnost IMI Precision Engineering.

Při vytváření jsem se opíral především o obdržený počáteční návrh databáze a dále také o rozhovory s pracovníky společnosti. Největším problémem při tvorbě práce byla má, ne příliš velká znalost jednotlivých atributů u jednotlivých výrobků, což vyžadovalo mnoho práce se zjišťováním ohledně jejich významu.

Výsledkem této bakalářské práce je to, že na základě požadavků na databázi jsem byl schopen vytvořit návrh databáze, který splňuje funkcionalitu a zvyšuje efektivitu výrobních procesů ve společnosti. Samotný fyzický návrh jsem vytvářel v programu Microsoft SQL Server. Cíl práce byl tedy splněn. Přizpůsobení databáze požadavkům organizace vyžadovalo porozumět procesům probíhajícím uvnitř společnosti, tvorbě výrobních štítků pro jednotlivé výrobky, které firma vyrábí. Využití databáze přináší výhodu v tom, že data jsou uložena na jednom místě a tím do jisté míry dojde ke zlepšení jejich dostupnosti.

Databázový systém je navržený pro současnou situaci ve společnosti tak, aby vyhovoval jejím momentálním požadavkům. Do budoucna by se ovšem ještě hodilo pro efektivnější práci s databází vytvořit uživatelsky přívětivou aplikaci, která by nahradila samotný program Codesoft.

Téma této práce bylo pro mě skutečně obrovskou výzvou. Již od začátku jsem věděl, že se bude jednat o velmi rozsáhlou databázi, která mě přiměla rozšířit si znalosti v oboru návrhu a implementace databází.

SEZNAM POUŽITÝCH ZDROJŮ

- 1) CONOLLY, Thomas, Carolyn E. BEGG a Richard HOLOWCZAK. *Mistrovství-databáze: profesionální průvodce tvorbou efektivních databází*. Vyd. 1. Brno: Computer Press, 2009. ISBN 978-80-251-2328-7.
- 2) KOCH, Miloš a Bernard NEUWIRTH. *Datové a funkční modelování*. Vyd. 4., rozšířené. Brno: Akademické nakladatelství CERM, 2010, 142 s.: il., grafy, tab. ISBN 978-80-214-4125-5.
- 3) KROENKE, David, David J AUER a Jakub GONER. *Databáze*. 1. vyd. Brno: Computer Press, 2015, 496 s.: il., grafy, tab. ISBN 978-80-251-4352-0.
- 4) KOCH, Miloš a Viktor ONDRÁK. *Informační systémy a technologie*. Vyd. 3. Brno: Akademické nakladatelství CERM, 2008, 166 s.: il., grafy, tab. ISBN 978-80-214-3732-6.
- 5) OPPEL, Andrew J. *SQL bez předchozích znalostí: [průvodce pro samouky]*. Vyd. 1. Brno: Computer Press, 2008, 240 s. : il. ; 23 cm. ISBN 978-80-251-1707-1.
- 6) KRŽIŽ, J. a DOSTÁL, P. *Databázové systémy*. 1. vyd. Brno: Akademické nakladatelství CERM, 2005. 111 s. ISBN 80-214-3064-8.
- 7) STEPHENS, R., R. PLEW a A. D. JONES. *Naučte se SQL za 28 dní*. Brno: Computer Press, 2010. ISBN 978-80-251-2700-1.
- 8) SQL Server 2019 | Microsoft. [online]. Dostupné z: <https://www.microsoft.com/cs-cz/sql-server/sql-server-2019G>
- 9) RASSEOVÁ, Monika, Radek DUBEC a David ŘEHÁK. *Analýza v rukou manažera: 33 nejpoužívanějších metod strategického řízení*. Brno: Computer Press, 2010. ISBN 978-80-251-2621-9.
- 10) HRONEK, Jiří. *Informační systémy* [online]. Olomouc : Katedra informatiky. Přírodovědecká fakulta Univerzita Palackého, 2007 [cit. 2016-01-02]. Dostupné z: <https://phoenix.inf.upol.cz/esf/ucebni/infoSys.pdf>
- 11) *Informace o společnosti | TEKLYNX. Barcode label software | TEKLYNX* [online]. Copyright © Copyright 2021 Teklynx Newco SAS. Všechna práva vyhrazena. [cit. 13.05.2021]. Dostupné z: <https://www.teklynx.com/cz-EMEA/about/company-information>

12) SQL Data Types for MySQL, SQL Server, and MS Access. W3Schools Online
Web Tutorials [online]. Dostupné z:
http://www.w3schools.com/sql/sql_datatypes.asp

SEZNAM POUŽITÝCH OBRÁZKŮ

Obrázek 1: Schéma databázového systému	14
Obrázek 2: Komunikace mezi uživateli a databází.....	15
Obrázek 3: Vztah 1:1	19
Obrázek 4: Vztah 1:N	19
Obrázek 5: Vztah M:N.....	20
Obrázek 6: Dekompozice.....	20
Obrázek 7: Logo společnosti	29
Obrázek 8: Organizační struktura společnosti	31
Obrázek 9: Ukázka výrobního štítku	34
Obrázek 10: Ukázka výrobního štítku	34
Obrázek 11: Zkrácená verze ER diagramu	55
Obrázek 12: Příklad kódu pro vytvoření tabulky.....	56
Obrázek 13: Příklad kódu pro propojení tabulek pomocí cizích klíčů	57
Obrázek 14: Příklad kódu pro naplnění tabulek testovacími daty	57
Obrázek 15: Ukázka přidání databáze do programu Codesoft	58

SEZNAM POUŽITÝCH TABULEK

Tabulka 1: Tabulka datových typů pro text	27
Tabulka 2: Tabulka datových typů pro celá čísla	27
Tabulka 3: Tabulka datových typu pro přibližná čísla	28
Tabulka 4: Tabulka datových typů pro čas	28
Tabulka 5: Identifikace entit	37
Tabulka 6: Identifikace vztahů mezi entitami	39
Tabulka 7: Atributy tabulky Master_main	40
Tabulka 8: Atributy tabulky Master_printed	42
Tabulka 9: Atributy tabulky Master_labelsforpn	42
Tabulka 10: Atributy tabulky Master_labelsdetails	43
Tabulka 11: Atributy tabulky Stanice_pristup	44
Tabulka 12: Atributy tabulky ISO_main	44
Tabulka 13: Atributy tabulky ISO_vlastnosti	45
Tabulka 14: Atributy tabulky FRL_main	46
Tabulka 15: Atributy tabulky FRL_vlastnosti	47
Tabulka 16: Atributy tabulky HDM_main	48
Tabulka 17: Atributy tabulky HDM_vlastnosti	49
Tabulka 18: Atributy tabulky MEILLER_main	50
Tabulka 19: Atributy tabulky ACTUATORS_main	51
Tabulka 20: Atributy tabulky ACTUATORS_vlastnosti	52
Tabulka 21: Atributy tabulky FITTINGS_main	53
Tabulka 22: Atributy tabulky FITTINGS_vlastnosti	54

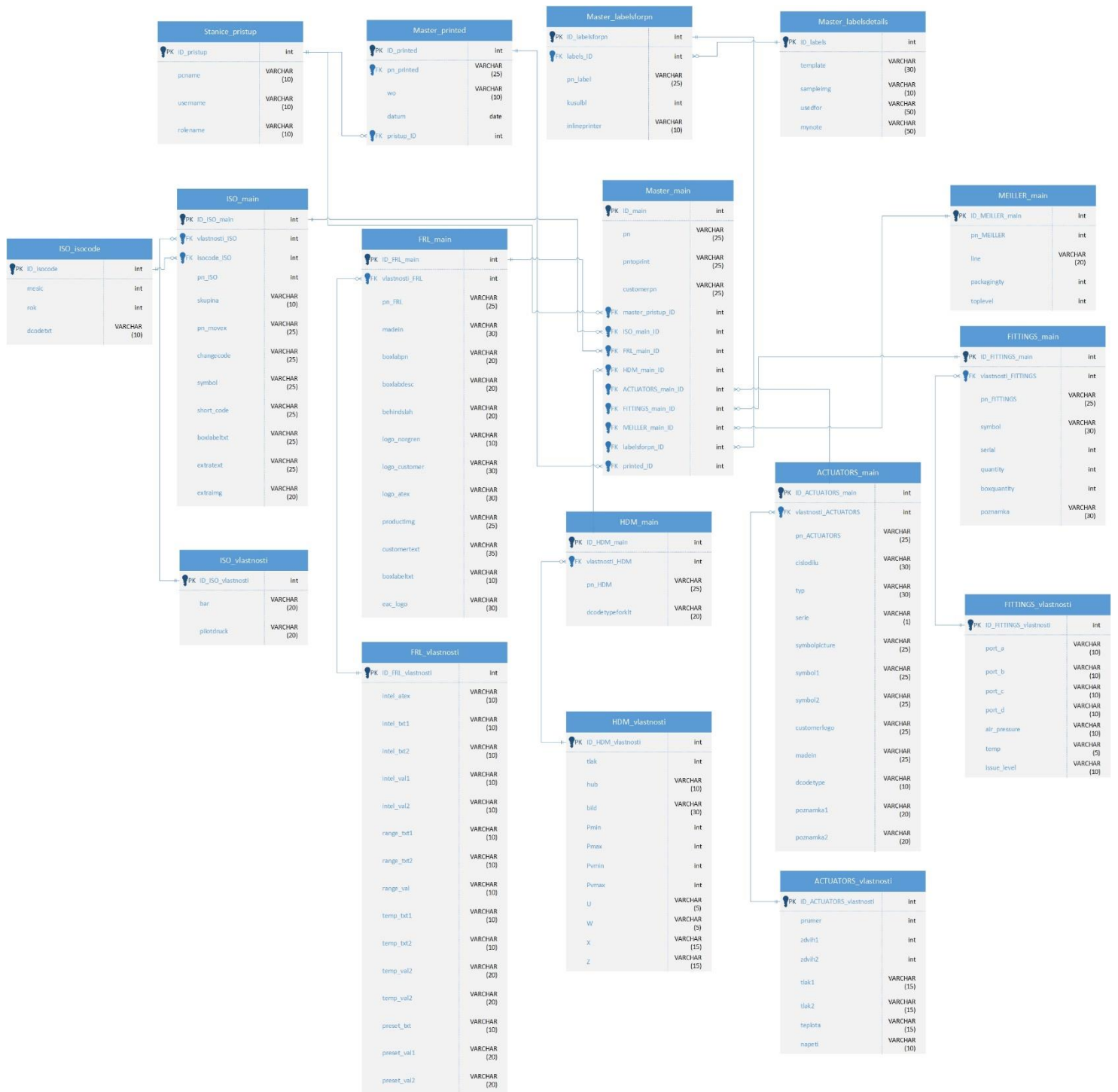
SEZNAM POUŽITÝCH ZKRATEK

DB	databáze
DBMS	database management system
DCL	data control language
DDL	data definition language
DML	data manipulation language
DQL	data query language
FK	foreign key
PK	primary key
SQL	structured query language

SEZNAM PŘÍLOH

Příloha 1: ER Diagram	I
Příloha 2: Kód pro vytvoření databáze	II

Příloha 1: ER Diagram



Příloha 2: Kód pro vytvoření databáze

```
-- 1. ISO_isocode --

create table ISO_isocode(
ID_isocode int not null identity (1,1) primary key,
mesic int,
rok int,
dcodetx varchar(10),
);

INSERT INTO ISO_isocode VALUES(12,2029,'980');
INSERT INTO ISO_isocode VALUES(11,2029,'979');
INSERT INTO ISO_isocode VALUES(10,2029,'978');
INSERT INTO ISO_isocode VALUES(9,2029,'977');
INSERT INTO ISO_isocode VALUES(8,2029,'976');
INSERT INTO ISO_isocode VALUES(7,2029,'975');
INSERT INTO ISO_isocode VALUES(6,2029,'974');
INSERT INTO ISO_isocode VALUES(5,2029,'973');
INSERT INTO ISO_isocode VALUES(4,2029,'940');
INSERT INTO ISO_isocode VALUES(3,2029,'939');
INSERT INTO ISO_isocode VALUES(2,2029,'938');
INSERT INTO ISO_isocode VALUES(1,2029,'937');

go

-- 2. ISO_vlastnosti --

create table ISO_vlastnosti(
ID_ISO_vlastnosti int not null identity (1,1) primary key,
bar varchar(20),
pilotdruck varchar(20),
);

INSERT INTO ISO_vlastnosti VALUES('-0.9 - 16','TBD');
INSERT INTO ISO_vlastnosti VALUES('3 - 10', '2 - 10');
INSERT INTO ISO_vlastnosti VALUES('-0.9 - 16','1.8 - 16');
INSERT INTO ISO_vlastnosti VALUES('2.5 - 10', '1.8 - 16');
INSERT INTO ISO_vlastnosti VALUES('-0.9 - 16', '2 - 10');

go

-- 3. FRL_vlastnosti --

create table FRL_vlastnosti(
ID_FRL_vlastnosti int not null identity (1,1) primary key,
intel_txt1 varchar(10),
intel_txt2 varchar(10),
intel_val1 varchar(10),
intel_val2 varchar(10),
range_txt1 varchar(10),
range_txt2 varchar(10),
range_val1 varchar(10),
range_val2 varchar(10),
temp_txt1 varchar(10),
temp_txt2 varchar(10),
temp_val1 varchar(20),
temp_val2 varchar(20),
preset_txt varchar(10),
preset_val1 varchar(20),
preset_val2 varchar(20),
);
```

```
);

INSERT INTO FRL_vlastnosti VALUES('MAX INLET','','145 psig','10
bar','RANGE','','1-60 psig','0,07-4 bar','MAX TEMP','','50°C','122°F','PRE-
SET','7.4 bar','9 bar');
INSERT INTO FRL_vlastnosti VALUES('MAX INLET','MAX','250 psig','17
bar','RANGE','','4-150 psig','0,7-17 bar','MAX TEMP','MAX','80°C','175°F','PRE-
SET','10 bar','13 bar');
INSERT INTO FRL_vlastnosti VALUES('INLET','','150 psig','11 bar','MAX SET','','5-
250 psig','0,3-10 bar','MAX TEMP','MAX','80°C','175°F','SET','2 bar','6 bar');
INSERT INTO FRL_vlastnosti VALUES('INLET','','300 psig','16 bar','SET','','7-100
psig','0,5-15 bar','TEMP','MAX','65°C','150°F','SET','4 bar','8 bar');
INSERT INTO FRL_vlastnosti VALUES('MAX INLET','MAX','250 psig','17 bar','MAX
SET','','1-60 psig','0,07-4 bar','MAX TEMP','','50°C','122°F','PRE-SET','7.4
bar','9 bar');
```

go

-- 4. HDM_vlastnosti --

```
create table HDM_vlastnosti(
ID_HDM_vlastnosti int not null identity (1,1) primary key,
tlak int,
hub varchar(10),
bild varchar(30),
Pmin int,
Pmax int,
Pvmin int,
Pvmax int,
U varchar(5),
W varchar(5),
X varchar(15),
Z varchar(15),
);
```

```
INSERT INTO HDM_vlastnosti
VALUES(10,'10','s5251054.bmp',0.4,3,3,6,'0,5','9,5','Bypass Z1','Bypass Z2');
INSERT INTO HDM_vlastnosti
VALUES(10,'25','s5261006.bmp',3,8','','','0,5','9,5','','4');
INSERT INTO HDM_vlastnosti
VALUES(10,'45','s5261006.bmp',1.5,10,3,6,'5','7','','abs. ');
INSERT INTO HDM_vlastnosti VALUES(0','','s5251021.bmp',1,10,5,7,'0,4','11','Bypass
Z1','Bypass Z2');
INSERT INTO HDM_vlastnosti VALUES(10,'10','3-
2_045.wmf',0.3,3,5,7,'0,2','6','4','30');
```

go

-- 5. ACTUATORS_vlastnosti

```
create table ACTUATORS_vlastnosti(
ID_ACTUATORS_vlastnosti int not null identity (1,1) primary key,
prumer int,
zdvih1 int,
zdvih2 int,
tlak1 varchar(15),
tlak2 varchar(15),
);
```

```
teplota varchar(15),
napeti varchar(10),
);
```

```
INSERT INTO ACTUATORS_vlastnosti VALUES(12,20,'','p 2 - 10bar','','Tmax
80°C','240V AC');
INSERT INTO ACTUATORS_vlastnosti VALUES(25,50,'','p 2 - 10bar','','Tmax
80°C','240V AC');
INSERT INTO ACTUATORS_vlastnosti VALUES(20,160,'','p 1 - 10bar','','Tmax
80°C','240V AC');
INSERT INTO ACTUATORS_vlastnosti VALUES(100,125,150,'p 1 - 16bar','p 2 -
16bar','Tmax 80°C','240V AC');
INSERT INTO ACTUATORS_vlastnosti VALUES(200,150,'','p 2 - 16bar','','Tmax
80°C','240V AC');
```

```
go
-- 6. FITTINGS_vlastnosti
```

```
create table FITTINGS_vlastnosti(
ID_FITTINGS_vlastnosti int not null identity (1,1) primary key,
port_a varchar(10),
port_b varchar(10),
port_c varchar(10),
port_d varchar(10),
air_pressure varchar(10),
temp varchar(5),
issue_level varchar(10),
);
```

```
INSERT INTO FITTINGS_vlastnosti VALUES('6mm','','','1/8NPT','','1','0');
INSERT INTO FITTINGS_vlastnosti
VALUES('10mm','145979','M22X1.5','10mm','','1','0');
INSERT INTO FITTINGS_vlastnosti VALUES('6mm','','6mm','6mm','R1','1','0');
INSERT INTO FITTINGS_vlastnosti VALUES('R1/4','','6mm','6mm','','1','0');
INSERT INTO FITTINGS_vlastnosti VALUES('6mm','3/8','','1/8','','1','0');
```

```
go
-- 7. Stanice_pristup
```

```
create table Stanice_pristup(
ID_pristup int not null identity (1,1) primary key,
pcname varchar(30),
username varchar(10),
rolename varchar(10),
);
```

```
INSERT INTO Stanice_pristup VALUES('CZMODDT657T41','jbartos','IT');
```

```
go
-- 8. Master_labelsdetails --
```

```
create table Master_labelsdetails(
ID_labels int not null identity (1,1) primary key,
```

```

template varchar(30),
sampleimg varchar(10),
usedfor varchar(50),
mynote varchar(50),
);

INSERT INTO Master_labelsdetails VALUES('box1.lab','','Norgren PN','');
INSERT INTO Master_labelsdetails VALUES('box1_wwyycz.lab','','Norgren PN -
odlisny date code','');
INSERT INTO Master_labelsdetails VALUES('box2.lab','','Scania PN','');
INSERT INTO Master_labelsdetails VALUES('box3.lab','','SCANIA + NORGREN PN + W +
qty na krabicku','');

```

go

--9. Masater_labelsforpn ---

```

create table Master_labelsforpn(
ID_labelsforpn int not null identity (1,1) primary key,
pn_labels varchar(25) not null,
labels_ID int constraint labels_ID references Master_labelsdetails (ID_labels),
kusulbl varchar(50),
inlineprinter varchar(50),
);

```

```

INSERT INTO Master_labelsforpn VALUES('24661/A',1,'1','0');
INSERT INTO Master_labelsforpn VALUES('10-003-997',2,'1','0');
INSERT INTO Master_labelsforpn VALUES('10-003-997-24',3,'1','0');
INSERT INTO Master_labelsforpn VALUES('10-009-984',3,'1','0');
INSERT INTO Master_labelsforpn VALUES('FP 2001',2,'1','0');
INSERT INTO Master_labelsforpn VALUES('FP2885-22',1,'1','0');
INSERT INTO Master_labelsforpn VALUES('NME114-0007',2,'1','0');

```

go

-- 10. Master_printed --

```

create table Master_printed(
ID_Printed int not null identity (1,1) primary key,
pn_printed varchar(25) not null,
wo varchar(25),
datum date,
pristup_ID int not null constraint pristup_ID references Stanice_pristup
(ID_pristup),
);

```

```

INSERT INTO Master_printed VALUES('T64T-4GB-P1N','6408678','2021-2-5',1);
INSERT INTO Master_printed VALUES('KM/8012/M/25','6398817','2021-2-10',1);
INSERT INTO Master_printed VALUES('QOM/8025A/03','6413239','2021-2-3',1);
INSERT INTO Master_printed VALUES('M/P35344/15','6396853','2021-2-5',1);
INSERT INTO Master_printed VALUES('RA/192040/MX/10','6402438','2021-2-6',1);

```

go

-- 11. ISO_main --

```

create table ISO_main(
ID_ISO_main int not null identity (1,1) primary key,
ISO_vlastnosti_ID int not null constraint ISO_vlastnosti_ID references
ISO_vlastnosti (ID_ISO_vlastnosti),
ISO_isocode_ID int not null constraint ISO_isocode_ID references ISO_isocode
(ID_isocode),
pn_ISO varchar(25) not null,
skupina varchar(25),
PN_MOVEX varchar(25),
PN_JDE varchar(25),
changecode varchar(1),
symbol varchar(30),
short_code varchar(3),
boxlabeltxt varchar(30),
extraimg varchar(20),
);

INSERT INTO ISO_main VALUES(1,2,'VSP550070/PROT','ISO
3','VSP550070/PROT','VSP550070/PROT','C','031.BMP','','','cruus21.jpg');
INSERT INTO ISO_main VALUES(2,2,'SXE9773-155-00K','ISO 1','SXE9773-155-
00K','SXE9773-155-00','C','099.BMP','','','');
INSERT INTO ISO_main VALUES(3,3,'FP2885','BCC','FP 2885FP
2885','FP2885','C','015.BMP','LGU','','VS26_1.png');
INSERT INTO ISO_main
VALUES(4,3,'VS26G511DF513A','VS','VS26G511DF513A','VS26G511DF513A','C','033.BMP',
'LJU','','cruus21.jpg');
INSERT INTO ISO_main
VALUES(5,4,'VS18SA11DF313A','VS18','VS18SA11DF313AB','VS18SA11DF313A','C','031.BM
P','','','VS18_2.png');

```

```

go
-- 12. FRL_main --

```

```

create table FRL_main(
ID_FRL_main int not null identity (1,1) primary key,
FRL_vlastnosti_ID int not null constraint FRL_vlastnosti_ID references
FRL_vlastnosti (ID_FRL_vlastnosti),
pn_FRL varchar(25) not null,
madein varchar(30),
boxlabpn varchar(20),
boxlabdesc varchar(40),
behindslah varchar(20),
logo_norgren varchar(40),
logo_customer varchar(30),
logo_atex varchar(30),
productimg varchar(25),
customertext varchar(35),
boxlabeltxt varchar(10),
eac_logo varchar(30),
);

```

```

INSERT INTO FRL_main VALUES(1,'10-009-984','MADE IN CZECH REPUBLIC','10-009-
984','LUBRICATOR HEAD
ASSEMBLY','','iminorgrensmall.png','','','','','eaclabel.bmp');

```



```

INSERT INTO FRL_main VALUES(2,'10-993-004SA','MADE IN CZECH REPUBLIC','10-993-004','Head Assy for 10-993-004','','iminorgrensmall.png','','','','','eaclabel.bmp');
INSERT INTO FRL_main VALUES(3,'11-018-100-51','MADE IN CZECH REPUBLIC','11-018-100/51','REG SET MAX 70 PSI','','iminorgrensmall.png','','','','','eaclabel.bmp');
INSERT INTO FRL_main VALUES(4,'11-018-999','MADE IN CZECH REPUBLIC','11-018-999','REGULATOR','','iminorgrensmall.png','','','','','USE 5 MICRON FILTER','','eaclabel.bmp');
INSERT INTO FRL_main VALUES(5,'11-818-100-EX','MADE IN CZECH REPUBLIC','11-818-100','REGULATOR','','iminorgrensmall.png','','','','','USE 5 MICRON FILTER','','eaclabel.bmp');

```

```

go
-- 13. HDM_main --

```

```

create table HDM_main(
ID_HDM_main int not null identity (1,1) primary key,
HDM_vlastnosti_ID int not null constraint HDM_vlastnosti_ID references
HDM_vlastnosti (ID_HDM_vlastnosti),
pn_HDM varchar(25) not null,
dcodetypeforklt varchar(20),
);

```

```

INSERT INTO HDM_main VALUES(1,'4545325000000000','fellbach');
INSERT INTO HDM_main VALUES(2,'4545225000000000','fellbach');
INSERT INTO HDM_main VALUES(3,'4546353000000000','fellbach');
INSERT INTO HDM_main VALUES(4,'4545788000000000','fellbach');
INSERT INTO HDM_main VALUES(5,'6257925000000000','fellbach');

```

```

go
--14. ACTUATORS_main --

```

```

create table ACTUATORS_main(
ID_ACTUATORS_main int not null identity (1,1) primary key,
ACTUATORS_vlastnosti_ID int not null constraint ACTUATORS_vlastnosti_ID
references ACTUATORS_vlastnosti (ID_ACTUATORS_vlastnosti),
pn_ACTUATORS varchar(25) not null,
cislodilu varchar(30),
typ varchar(30),
serie varchar(1),
symbolpicture varchar(25),
symbol1 varchar(25),
symbol2 varchar(25),
customerlogo varchar(25),
madein varchar(25),
dcodetype varchar(10),
poznamka1 varchar(20),
poznamka2 varchar(20),
);

```

```

INSERT INTO ACTUATORS_main
VALUES(1,'RT/57312/M/20','RT/57312/M/20','RT/57312/M/20','A','2CSSOM.bmp','HUB.BMP','LEER.BMP','','MADE IN CZECH REPUBLIC','','');

```

```

INSERT INTO ACTUATORS_main
VALUES(2,'SPC/020162/25','SPC/020162/25','SPC/020162/25','D','2CSSOM.bmp','HUB.BM
P','LEER.BMP','','MADE IN CZECH REPUBLIC','','');
INSERT INTO ACTUATORS_main
VALUES(3,'RM/28010/M/50','RM/28010/M/50','RM/28010/M/50','C','2CSSOM.bmp','HUB.BM
P','LEER.BMP','','MADE IN CZECH REPUBLIC','','');
INSERT INTO ACTUATORS_main
VALUES(4,'RA/8063/200','RA/8063/200','RA/8063/200','A','2CSSOM.bmp','HUB.BMP','LE
ER.BMP','','MADE IN CZECH REPUBLIC','','');
INSERT INTO ACTUATORS_main
VALUES(5,'PRA/182100/M/500','PRA/182100/M/500','PRA/182100/M/500','B','2CSSOM.bmp
','HUB.BMP','LEER.BMP','','MADE IN CZECH REPUBLIC','','');

```

go

-- 15. FITTINGS_main --

```

create table FITTINGS_main(
ID_FITTINGS_main int not null identity (1,1) primary key,
FITTINGS_vlastnosti_ID int not null constraint FITTINGS_vlastnosti_ID references
FITTINGS_vlastnosti (ID_FITTINGS_vlastnosti),
pn_FITTINGS varchar(25) not null,
symbol varchar(30),
serial int,
quantity int,
boxquantity int,
poznamka varchar(30),
);

```

```

INSERT INTO FITTINGS_main VALUES(1,'98411404M','984114.bmp',0,400,1,'FLEETFIT');
INSERT INTO FITTINGS_main
VALUES(2,'97452141PSBP','974521.bmp',0,100,1,'FLEETFIT');
INSERT INTO FITTINGS_main VALUES(3,'48033723','480337.bmp',0,200,1,'FLEETFIT');
INSERT INTO FITTINGS_main
VALUES(4,'97413845PSBP','974138.bmp',0,100,1,'FLEETFIT');
INSERT INTO FITTINGS_main
VALUES(5,'97413844PSBP','974138.bmp',0,100,1,'FLEETFIT');

```

go

-- 16. MEILLER_main --

```

create table MEILLER_main(
ID_MEILLER_main int not null identity (1,1) primary key,
pn_MEILLER varchar(25) not null,
line varchar(20),
packagingty int,
toplevel int,
);

```

```

INSERT INTO MEILLER_main VALUES('0110594000000000','Verteiler',20,0);
INSERT INTO MEILLER_main VALUES('0111504200000000','Verteiler',20,0);
INSERT INTO MEILLER_main VALUES('0333400000000000','Verteiler',10,0);
INSERT INTO MEILLER_main VALUES('0548273000000000','Verteiler',0,0);
INSERT INTO MEILLER_main VALUES('0613172000000000','Verteiler',40,99);

```

```

-- 17. Master_main --
create table Master_main(
ID_main int not null identity (1,1) primary key,
pn varchar(25) not null ,
pntoprint varchar(25),
customerpn varchar(25),
master_pristup_ID int not null constraint master_pristup_ID references
Stanice_pristup (ID_pristup),
ISO_main_ID int not null constraint ISO_main_ID references ISO_main
(ID_ISO_main),
FRL_main_ID int not null constraint FRL_main_ID references FRL_main
(ID_FRL_main),
HDM_main int not null constraint HDM_main_ID references HDM_main (ID_HDM_main),
ACTUATORS_main_ID int not null constraint ACTUATORS_main_ID references
ACTUATORS_main (ID_ACTUATORS_main),
FITTINGS_main_ID int not null constraint FITTINGS_main_ID references
FITTINGS_main (ID_FITTINGS_main),
MEILLER_main_ID int not null constraint MEILLER_main_ID references MEILLER_main
(ID_MEILLER_main),
labelsforpn_ID int not null constraint labelsforpn_ID references
Master_labelsforpn (ID_labelsforpn),
printed_ID int not null constraint printed_ID references Master_printed
(ID_printed),
);

INSERT INTO Master_main
VALUES('VSP550070/PROT', 'VSP550070/PROT', '', 1,1,3,1,1,1,1,1,1);
INSERT INTO Master_main VALUES('SXE9773-155-00K', 'SXE9773-155-
00K', '', 1,2,3,1,1,1,1,1,2);
INSERT INTO Master_main
VALUES('0115042000000000', '0115042000000000', '', 1,1,3,1,1,1,2,2,1);
INSERT INTO Master_main
VALUES('RT/57312/M/20', 'RT/57312/M/20', '', 1,1,3,1,1,1,1,2,2);
INSERT INTO Master_main
VALUES('SPC/020162/25', 'SPC/020162/25', '', 1,1,3,1,1,2,1,2,3);
INSERT INTO Master_main
VALUES('4545325000000000', '4545325000000000', '', 1,1,3,1,1,1,1,2,3);

```