

UNIVERZITA PALACKÉHO V OLOMOUCI

PEDAGOGICKÁ FAKULTA

Katedra technické a informační výchovy

## **Bakalářská práce**

Miroslav Paráček

**Návrh relační databáze pro evidenci zaměstnaneckých dat**

## **Prohlášení**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a uvedl jsem v ní veškerou literaturu a ostatní informační zdroje, které jsem použil.

V Olomouci dne 14.04.2023

A handwritten signature in blue ink, appearing to be 'P. P. P.', written over a dotted line.

vlastnoruční podpis

## **Poděkování**

Chtěl bych poděkovat Mgr. Tomáši Dragonovi za vedení mé bakalářské práce a za pomoc s jejím vyhotovením v náležitém stavu. Děkuji panu prof. PhDr. Milanovi Klementovi, Ph.D. za konzultace mé práce. Také velice děkuji panu Dipl.-Ing. Petrovi Jozefovičovi za možnost praktického vypracování, teoretického vedení a cenných rad, které mi velmi pomohly jak při psaní práce, tak i v osobním životě. Samozřejmě děkuji své rodině za podporu ve studiu, výchovu, kterou mi předali nejdůležitější hodnoty života a za trpělivost se mnou.

Miroslav Parák

# Obsah

ÚVOD.....	5
1 DATABÁZE .....	6
1.1 HISTORIE DATABÁZÍ .....	6
2 ARCHITEKTURA DATABÁZOVÝCH SYSTÉMŮ.....	9
3 DATABÁZOVÉ MODELY.....	12
4 SOFTWARE PRO SPRÁVU DATABÁZÍ .....	16
4.1 SOFTWARE PRO SPRÁVU NERELAČNÍCH DATABÁZÍ.....	16
4.2 SOFTWARE PRO SPRÁVU RELAČNÍCH DATABÁZÍ .....	17
4.3 MICROSOFT ACCESS .....	18
5 PRAKTICKÁ ČÁST.....	20
5.1 ÚVOD DO PRAKTICKÉ ČÁSTI .....	20
5.2 VYTVOŘENÍ DATABÁZE A TABULEK .....	22
5.3 VYTVOŘENÍ FORMULÁŘE A JEHO ÚPRAVA .....	32
ZÁVĚR .....	59
Seznam použitých zdrojů .....	60
Seznam obrázků .....	63
Seznam tabulek .....	65
Anotace .....	66
1.	

# ÚVOD

Téma této bakalářské práce bude zaměřeno na problematiku správy zaměstnaneckých dat v relačních databázích. K zpracování tohoto tématu jsem se inspiroval prací ve firemní sféře, kde jsem se nad touto problematikou zamýšlel. Viděl jsem velký potenciál k usnadnění, některých pracovních úkonů díky databázím. Práce bude logicky členěna na dvě části, a to teoretickou a praktickou. V teoretické části se zaměřím na základní informace, které je nevyhnutelné vědět před prací s relačními databázemi. Také popíšu historický vývoj databází jako takových. Chtěl bych i nastínit v teoretické části to, jak se databáze proměnili a dokážou nám díky tomu usnadnit mnohé práce. Myslím si že, je důležité povšimnout si co vše dnešní databázové systémy dokážou. V praktické části navrhnu databázi na zpracování zaměstnaneckých dat v provozu firmy. Zde budu demonstrovat pojmy z teoretické části. Praktická část bude vytvořena v programu Access, zde také popíšu odlišnost tohoto programu od jiných databázových programů. Četl jsem v knize Databázové systémy od Jaroslava Pokorného a Michala Valenty velice krásný citát, který si myslím že by se do úvodu mohl hodit. T. S. Elliot: „Kde je moudrost? Ztracena ve znalostech. Kde jsou znalosti? Ztraceny v informacích“ Joe Celko navazuje: „Kde jsou informace? Ztraceny v datech. Kde jsou data? Ztracena v databázích.“

# 1 DATABÁZE

Databáze je systém, který spravuje data. Díky databázím můžeme data organizovaně ukládat, aktualizovat a vybírat data podle předepsaných pravidel. Dále můžeme modelovat vzájemné vztahy mezi daty a celými datovými soubory. Obvykle data ukládáme ve formě tabulek, ve kterých jsou řádky názvy zadávaných dat a sloupce data samotná (neboli relační databáze). Zkráceně bychom mohli databázi popsat jako úložiště dat, které nám umožňuje data následně třídit, snadněji vyhledávat a analyzovat.<sup>1</sup>

## 1.1 HISTORIE DATABÁZÍ

Historii Databází považuji za téma, které je důležité zmínit pro porovnání s dnešními databázemi. Také mi přijde velice zajímavé, jak se dnešní typ databází oproti jejich předchůdcům proměnil.

### VVZNIK PRVNÍCH DATABÁZÍ

S rychlým vývojem lidstva a přibývajícimi informacemi se začaly vyvíjet různé systémy na třídění dat a informací. Což bylo nevyhnutelné pro zrychlení chodu práce.

Za první databáze můžeme považovat papírové kartotéky (Lékařské kartotéky se záznamy pacientů, kartotéky knih v knihovnách, seznamy a údaje o zaměstnancích ve firmách a tak dále). O správu a vyhledávání dat v těchto kartotékách se staral daný člověk (sestra u lékaře, sekretářka, knihovnice). Někdy bylo za potřebí i založit více kartoték, aby bylo možno vyhledávat třeba podle jména nebo podle roku narození.<sup>2</sup>

### STROJEM ŘÍZENÉ DATABÁZE

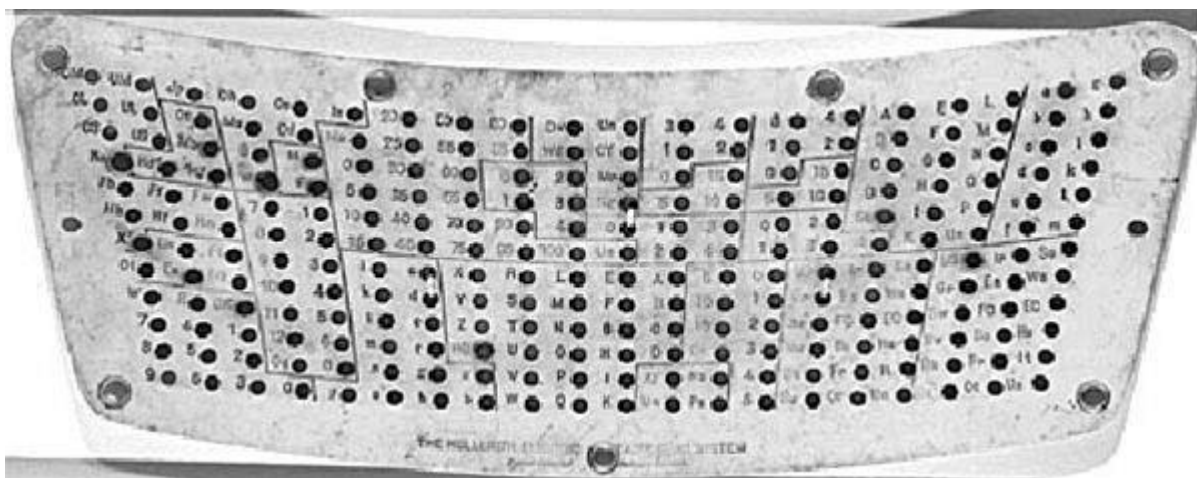
První stroj řízené zpracování dat proběhlo v USA v roce 1890 a sčítal se lid. Samozřejmě se nejednalo o počítač. Lidé vyplňovali speciální formuláře, které později byly prepisovány do děrných štítků. Děrné štítky následně byly po jednom dávány do čtecího stroje, který následně zaznamenával údaje. To přineslo velké zrychlení toku informací a díky tomuto

---

<sup>1</sup> BULISOVÁ Jiřina, 2010 (Str.257)

<sup>2</sup> ŠKODA Jan, 2017 (Kap. 1)

systemu byl lid sečten za pouhých šest týdnů. Ano pouhých, protože předchozí sčítání trvalo dva roky. Toto sčítání lidu odstartovalo velký vývoj v třídění a uchovávání dat.<sup>3</sup>



Obr. 1: Děrný štítek z amerického sčítání lidu v roce 1890

Zdroj: [https://www.idnes.cz/technet/technika/scitani-lidu-odpichlo-prichod-prvnich-pocitacu.A080104\\_172231\\_tec\\_technika\\_pka](https://www.idnes.cz/technet/technika/scitani-lidu-odpichlo-prichod-prvnich-pocitacu.A080104_172231_tec_technika_pka) [cit. 13.02.2022]

## PRVNÍ POČÍTAČOVÉ DATABÁZE

Nástupem počítačů nastal velký vývoj databází. Počítače s sebou přinesly velké zjednodušení práce s daty a urychlení práce zaměstnanců. Proto se začalo tomuto odvětví věnovat více pozornosti. To si můžeme nastínit na datové ose:

- 1959 – Vyvinutí univerzálního databázového jazyka COBOL;
- 1970 – Vznik prvních relačních databázových systémů E. F. Codd;
- 1974 – Vzniká dnes nejrozšířenější databázový jazyk SQL;
- 1978 – Vznikl rozsáhlý databázový systém ORACLE, primárně pro podniky;
- 1980 – Začínají vznikat databázové systémy i pro uživatele osobních počítačů, např.: dBase, FoxBase, Paradox;
- 1991 – Vzniká program MS Access;
- 1995 – Příklad MySQL, které využívá jazyk SQL a je vhodný na využití s php (internetové obchody).<sup>4</sup>

<sup>3</sup> Ing. KOTÁSEK Jaroslav, 2012 (Str. 5)

<sup>4</sup> Ing. KOTÁSEK Jaroslav, 2012 (Str.6)

## MODERNÍ DATABÁZE

Ke konci této kapitoly mi přijde podstatné i zmínit pro porovnání, nějaké požadavky a možné podoby dnešních databází.

V minulosti bylo zpracování dat a pochopení jejich významu velmi zdlouhavé, často trvalo hodiny až dny. Nicméně dnes organizace potřebují zpracovávat a analyzovat data v reálném čase, aby mohly vyhovět požadavkům zákazníků. Pro poskytování aktuálních dat se zvyšujícím se počtem analytiků, jsou potřeba systémy s optimalizovanou pamětí, které dokáží zpracovávat transakce a analýzy současně. Schopnost provádět dotazy v reálném čase se tak stala základním požadavkem pro moderní databáze. Moderní databáze by měla podporovat i vzdálený přístup což je v dnešní době velice důležité.<sup>5</sup>

Pro představu v odstavci níže uvedu nějaké možné podoby dnešních databází.

Využití a podoba databází dnes je velice rozmanitá. Téměř každý člověk se v současnosti denně setkává s databází. Můžeme si uvést příklad: Přihlášení do bankovního portálu. Pomocí našeho zvoleného přihlašovacího jména a k tomu hesla po přihlášení do systému se v databázi banky přiřadí všechny údaje k nám. To znamená zůstatek peněz na kartě, naše jméno, rodné číslo, stav spoření a tak dále... Toto vše je uloženo v databázi, a tak bychom mohli pokračovat všemi nejrůznějšími webovými aplikacemi. Dále bychom mohli zmínit PC hry, k našemu jménu a heslu máme různý počet postav, tanků a kreditů v dané hře a vše se ukládá do databáze. Také webové články jsou založeny na databázích, podle názvu článku, herce, o kterém článek je nebo něco co s článkem souvisí si článek dokážu vyhledat.

Možná podoba databáze zaměstnanců.

Zaměstnanecké číslo	Jméno	Příjmení	Pozice	Plat	Nástup
1458	Luděk	Noha	Účetní	23 000,00 Kč	15.03.2019
1582	Václav	Jeřábek	Uklízeč	19 500,00 Kč	13.11.2020
8520	Mikuláš	Lampička	Operátor výroby	26 000,00 Kč	23.09.2021
5570	Jindřich	Novák	Operátor výroby	26 000,00 Kč	24.12.2020
7809	Matyáš	Horák	Seřizovač	27 500,00 Kč	28.09.2021
1032	Zbyněk	Skočdopole	Skladník	24 500,00 Kč	24.05.2018
4589	Aleš	Stolář	Ředitel	42 000,00 Kč	14.03.2016

Obr. 2: Databáze zaměstnanců [13.02.2022]

Zdroj: vlastní archiv (data jsou fiktivní)

<sup>5</sup> WHITE Kevin, 2015



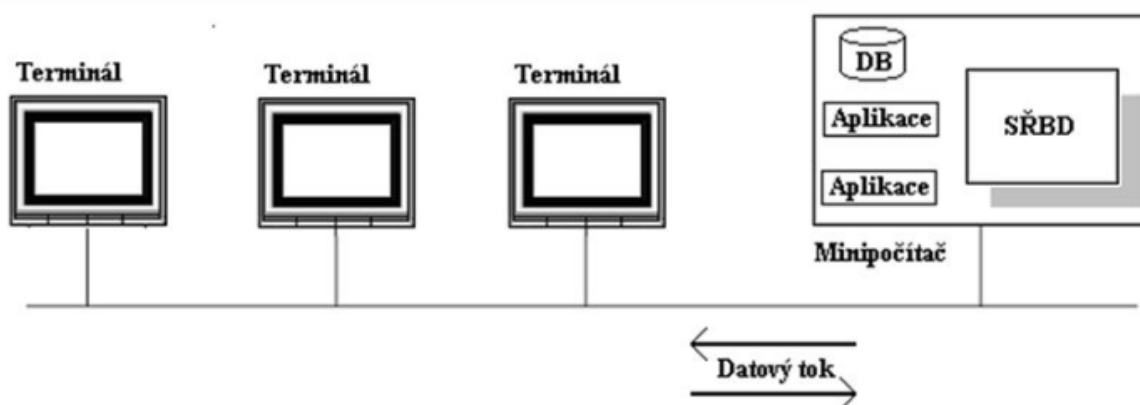
## 2 ARCHITEKTURA DATABÁZOVÝCH SYSTÉMŮ

Architektura databázových systémů DBMS (Database Management System) se zabývá návrhem a využitím softwarových systémů, které se používají k správě, ukládání a zpracování velkého množství dat. Pro optimalizaci a efektivní využití výkonu je správné zvolení DBMS podstatné. Architekturu databázových systémů bychom mohli shrnout jako soubor principů, postupů, pravidel a návrhů, jakým by měl databázový systém odpovídat.<sup>6</sup>

Z hlediska zákazníka bychom se mohli bavit o tom jaké mám potřeby, finanční možnosti, počet uživatelů využívajících databázi a jakým stylem budu informace vyhledávat a zpracovávat.

Další dělení DBMS je na: jednovrstvou architekturu, dvouvrstvou architekturu a vícevrstvou architekturu.<sup>7</sup>

- Jednovrstvá architektura – Všechny komponenty jsou umístěny na jednom počítači. Využívá se pro menší databáze, kde přistupuje menší počet uživatelů. Je náchylnější k výpadkům a poruchám. Všechny úkony se zpracovávají na jednom počítači tudíž dochází k větším časovým prodlevám. Také je kladena větší náročnost na hardware.



Obr. 3: Jednovrstvá architektura

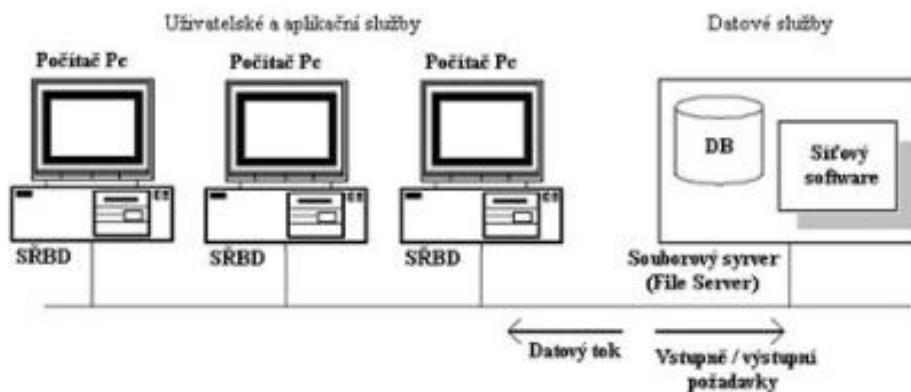
Zdroj: <https://slideplayer.cz/slide/2444856/>

- Dvouvrstvá architektura – Dělí se dále na „klient-server“ a „file-server“. Základní rozdíl mezi těmito typy je v tom, že u „file-server“ běží aplikace

<sup>6</sup> Škoda Jan, 2019 (Kap. 2)

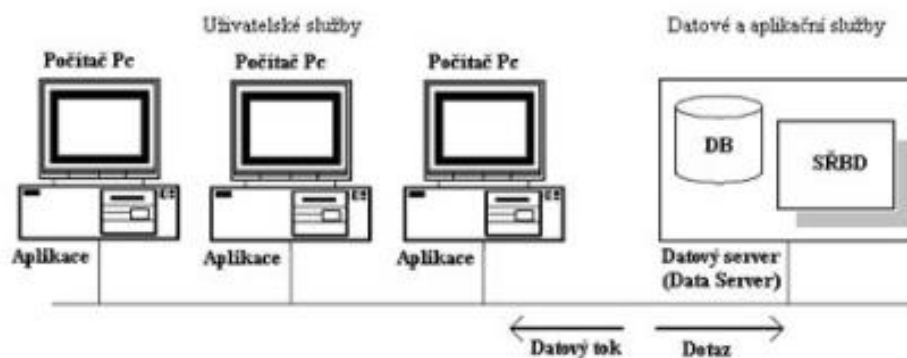
<sup>7</sup> POKORNÁ Dagmar (SlidePlayer)

a databázový server na jednom počítači. U „klient-server“ dochází ke komunikaci mezi databázovým server a aplikací odděleně pomocí síťového připojení.



Obr. 4: Dvoustupňová architektura File-Server [01.03.2023]

Zdroj: <https://slideplayer.cz/slide/2444856/>

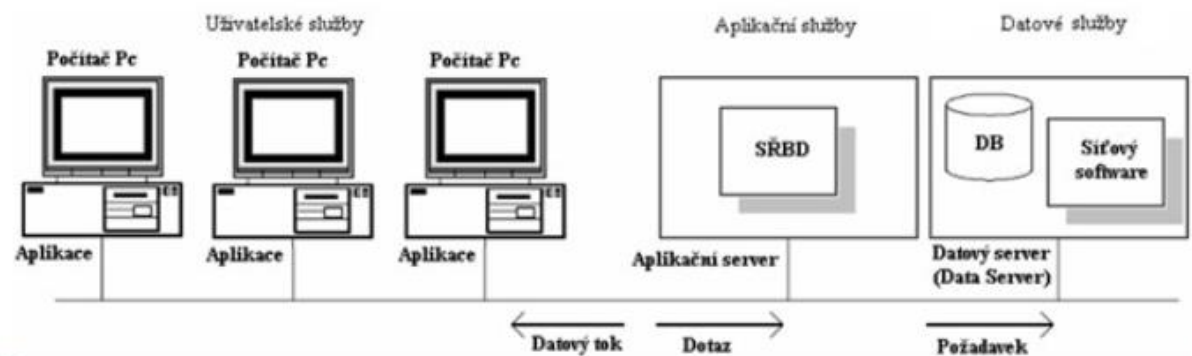


Obr. 5: Dvoustupňová architektura Klient-Server [01.03.2023]

Zdroj: <https://slideplayer.cz/slide/2444856/>

- Vícevrstvá architektura – Dělí se do více vrstev, prezentační vrstva, aplikační vrstva, datová vrstva, vrstva pro ukládání dat. Uživatel pracuje na svém zařízení s aplikací a veškerý výkon je soustředěn na serveru kam dále uživatel jen přistupuje. Vše je rozděleno do dvou samostatných celků. <sup>8</sup>

<sup>8</sup> POKORNÁ Dagmar (SlidePlayer)



Obr. 6: Vícevrstvá architektura [01.03.2023]

Zdroj: <https://slideplayer.cz/slide/2444856/>

## RELAČNÍ DATABÁZOVÝ SYSTÉM

System správy relační databáze (Relational Database Management System). Je důležité zmínit, že RDBMS a relační databáze jsou dva různé pojmy. RDBMS totiž umožňuje správu relační databáze a relační databáze pouze charakterizuje způsob organizace dat v tabulkách.

### 3 DATABÁZOVÉ MODELY

Pro práci s databázemi je důležité znát všechny databázové modely abychom si mohli udělat představu, jak fungují. Díky tomu si i můžeme na míru vybrat ten, který nám bude vyhovovat pro naši potřebu nejvíce.

Požadavky na funkce databází jsou rozličné, a to hned v několika směrech. Pro někoho je důležitá kapacita, formát dat, zabezpečení dat nebo rychlost zpracování. I proto existuje více typů a podtypů databázových systémů, které pro uchování a práci s daty můžeme využít. Dva nejvíce používané typy databázových systémů jsou relační (SQL) a nerelační (NoSQL).

**Hierarchická databáze** byla vyvinuta v šedesátých letech společností IBM. Hierarchický databázový model je návrh, který využívá vztah jeden k mnoha pro uspořádání datových prvků. Tyto databázové modely se opírají o stromovou strukturu, která propojuje různorodé prvky s jedním nadřazeným primárním záznamem či vlastníkem.<sup>9</sup> Právě tuto nevýhodu odstraňuje databáze síťová.

**Síťová databáze** umožňuje, aby určitý záznam mohl mít více nadřazených záznamů. Tyto databáze byly nejvíce využívány v osmdesátých letech. Po čase, ale hierarchickou a síťovou databázi vytlačila relační databáze. Ta převažuje společně s nerelační dodnes. Hierarchické i síťové databáze úplně nevymizely a stále se využívají v určitých případech. Jsou vyhledávány díky jejich jednoduchosti a funkčnosti.<sup>10</sup>

**Objektová databáze** je založena na objektově orientovaném programování. Databáze relační bývají většinou odděleny od aplikací, ty objektové souvisí s konkrétními programovacími jazyky například Perl, Ruby nebo Python. Místo tabulek, které jsou velice typické pro relační databáze, pracuje objektová databáze přímo s objekty, na které jsou navázány jejich vlastnosti a samostatné objekty.

**Relační databáze** jsou v současné době nejpoužívanějším typem databáze. Jejím základním kamenem jsou takzvané relace, což jsou vlastně tabulky. Ty jsou složeny ze sloupců (atributů) a z řádků (záznamů). Relací je tedy tabulka příklad: „Žáci“ (viz *Tab. 1*). Atributem nám budou sloupce „ID žáka“, „Jméno“, „Rodné číslo“, „Třídní učitel“.

---

<sup>9</sup> Theastrologypage, 2023 (Co je to hierarchická databáze?)

<sup>10</sup> Jan MŮČKA, 2021

Tabulka: Žáci

ID žáka	Jméno	Rodné číslo	Třídní učitel
125	Aleš Bota	140506/7899	Pospíšil
147	Adam Kudla	020406/1266	Pospíšil
236	Barbora Myslivcová	030909/4411	Horák
569	Bronislav Ustál	141212/2266	Kvapil

Tab. 1: Ukázka tabulky relační databáze

Zdroj: vlastní tvorba (data jsou fiktivní)

Záznamy v relaci jsou čtyři. Naše atributy poté mohou plnit funkci klíčů a to primárního, cizího, kandidátního či alternativního.<sup>11</sup>

- Primární klíč – nikdy nesmí obsahovat hodnotu null, vždy musí být jedinečný. Označuje přímo soubor dat v databázi, které se podle něj poté vyhledávají.
- Cizí klíč – pomocí cizího klíče vyjadřujeme vztah mezi dvěma tabulkami. Do sloupců přiřazujeme hodnoty z jiné tabulky. (Např. Účetní pozice má pracoviště na účtárně)
- Kandidátní klíč – je sloupec nebo kombinace více sloupců. Jeden z kandidátních klíčů je klíčem primárním.
- Alternativní klíč – Používá se k primárnímu klíči jako jedinečný identifikátor.

Kandidátním klíčem v relaci Žáci by bylo „ID žáka“ a „Rodné číslo“. Jeden z kandidátních klíčů se může stát klíčem primárním, ke kterému se poté mohou navázat ostatní záznamy (řádky). Kandidátní klíče, které se nestanou klíčem primárním se stávají klíčem alternativním.

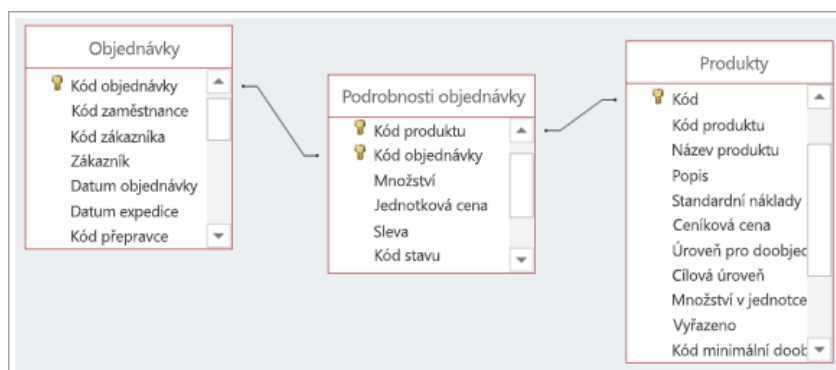
Primární klíč musí být unikátní, nesmí obsahovat nulovou hodnotu a měl by být neměnný. Pokud žádný atribut (sloupec) tyto kritéria nespĺňuje, můžeme vytvořit umělý identifikátor. Ten se poté automaticky přiřadí ke každému novému záznamu.

Relační databáze jsou tvořeny za pomoci jazyka SQL (Structured Query Language). Díky dobrému uspořádání je možné záznamy v tabulkách lehce filtrovat, třídít a provádět s nimi různé úkony. Samotné relace (tabulky) lze provázat i mezi sebou. K tomu právě slouží cizí klíč, který se shoduje s primárním klíčem v jiné relaci. Příkladem z praxe může být propojení

---

<sup>11</sup> Jan MŮČKA, 2021

záznamů o odběrateli, objednavce a dodaného produktu. Tyto záznamy jsou uvedeny v odlišných relacích (tabulkách).



Obr. 7: Relace [01.03.2023]

Zdroj: <https://support.content.office.net/cs-cz/media/948134e3-2c9d-4767-88ff-f8bbd27d5f58.png>

Do SQL databázových systémů patří MySQL, PostgreSQL, Oracle, Microsoft SQL, MariaDB, Access. Firmu MySQL odkoupila firma Oracle a bývalí majitelé MySQL vytvořili systém MariaDB tento systém je open source (otevřený systém) a to znamená že je zdarma. Systém Access je součástí balíčku Office 365 od společnosti Microsoft, tudíž náklady na pořízení nejsou vysoké. Zbylé programy jsou placené.

**Nerelační databáze** také NoSQL databáze, jsou oproti relačním databázím velice odlišné. Ukládání dat totiž neprobíhá v tabulkách, co jsou pevně definovány. NoSQL databáze představují kategorii datových úložišť, které nejsou relační a většinou nepoužívají SQL jako svůj dotazovací jazyk. NoSQL může být zavádějící a v databázové oblasti se spíše vysvětluje jako "nejen SQL". Mezi NoSQL databáze patří například XML databáze, objektové databáze a databáze dokumentů. Nerelační databáze jsou velice flexibilní a umožňují kombinovat odlišné složení dat, bez jakýchkoli zásahů do struktury databáze.<sup>12</sup>

Nejběžnější a nejjednodušší variantou NoSQL databází jsou:

- **„Key-value“** databáze. Tento podtyp přiděluje při uložení každé hodnotě její vlastní speciální klíč. Tato hodnota smí být číslem, řetězcem nebo také odkazem na další dvojici „klíč-hodnota“. Oproti SQL databázi zde není předem vymezena přesně daná

<sup>12</sup> POKORNÝ J. VALENTA M., 2020

struktura, což nám umožňuje jednoduché a efektivní propojení záznamů mezi sebou. Následujícím podtypem je

- „**Document-oriented**“ databáze, která je založena na ukládání dat v dokumentu, a to ve formátech JSON, BSON nebo XML. Tyto dokumenty obsahují hodnotu a klíč. Oproti relačním databázím je jejich výhodou zálohování různých informací v jednom dokumentu.
- „**Wide-column store**“ je dalším podtypem NoSQL databází, tento podtyp připomíná relační databáze nejvíce. Ve své struktuře využívá pouze sloupce, bez řádků. Posledním podtypem nerelačních databází, který zmíním je
- „**Graph database**“. Graph database je založena na dvou hlavních částích uzlech neboli nodes a vztazích mezi nimi edges. Nodes, uzly představují jednotlivé záznamy v databázi. Edges, vztahy určují, jak jsou záznamy mezi sebou propojeny. Systémy nerelačních databází jsou například MongoDB, Elasticsearch, Apache Cassandra, Amazone Neptun nebo Redis. <sup>13</sup>

---

<sup>13</sup> Jan MŮČKA, 2021

## 4 SOFTWARE PRO SPRÁVU DATABÁZÍ

V Softwaru pro správu databází budeme pracovat s databází a je důležité vybrat si ten software co nám bude vyhovovat nejvíce. V podstatě jde jen o nástroj s, kterým budeme dále pracovat. Proto v této kapitole představím ty, které považuji za nejznámější a nejvíce intuitivní, nebo nejlépe dostupné.

### 4.1 SOFTWARE PRO SPRÁVU NERELAČNÍCH DATABÁZÍ

**MongoDB** spadá do dokumentových nerelačních databází, byl vytvořen roku 2009 firmou MongoDB Inc. Tento software má dvě verze, jedna je zcela zdarma. Ta druhá, je placená a nabízí uživateli více funkcí pro správu databáze. MongoDB ukládá dokumenty ve formátu BSON (Binary JSON). Běžně dokumentové databáze nepodporují transakci dokumentů ACID, to právě řeší databáze MongoDB, která tuto transakci umožňuje.<sup>14</sup>

**Elasticsearch** je „fulltextový“ vyhledávač a analytický nástroj. Velikou doménou tohoto softwaru je jeho rychlost vyhledávání dat. Používá se především ve webových aplikacích, kde umožňuje v podstatě okamžitě používat nejrůznější filtry. První vydání tohoto programu bylo v roce 2010 a byl vyvinut v programovacím jazyce Java. Uživateli také poskytuje výkonné analýzy, které se dají snadno škálovat. Velikou výhodou je také bezesporu to že program je „open source“. K bezplatné verzi se samozřejmě dají zakoupit nejrůznější placené balíčky. Příklady použití Elasticsearch: Podnikové vyhledávání, Mapy, Monitorování infrastruktury, Sledování protokolu, Syntetické monitorování, Zabezpečení koncového bodu.<sup>15</sup>

**Amazon Neptune** spadá do balíčku aplikací Amazon Web Service, s kterými je propojen a data na sebe navazují. Amazon Neptune je grafová databázová služba. Engine jež využívá databáze od Amazon.com je optimalizovaný pro ukládání velkého množství dat s velice rychlým dotazováním na grafický náhled. Amazon Neptune se může využívat k vytváření propracovaných, interaktivních grafových aplikací. Amazon nabízí i bezplatnou verzi tohoto programu.<sup>16</sup>

**Apache Cassandra** je bezplatný a „open source“ NoSQL databázový program. Tento databázový software vznikl v roce 2008 a je napsán v jazyce Java. Apache Cassandra se prezentuje velmi malou náročností na hardware v lokálních sítích při připojení více uživatelů

---

<sup>14</sup> Mongo DB Inc, 2023

<sup>15</sup> Elastic NV, 2023

<sup>16</sup> Amazon.com, 2023



najednou. Jazyk, s kterým se pracuje při tvorbě databází v tomto programu se nazývá CQL (Cassandra Query Language). CQL je alternativou klasického SQL, je jednoduchý na pochopení a vhodný i pro nezkušené vývojáře. Apache Cassandra propojuje databázový model klíč-hodnota s klasickým sloupcovým modelem databází. To nabízí velikou škálovatelnost tohoto programu v praxi.<sup>17</sup>

## 4.2 SOFTWARE PRO SPRÁVU RELAČNÍCH DATABÁZÍ

**MySQL** je open source program, který byl založen firmou MySQL AB. Nyní je jeho majitelem firma ORACLE. Původní datum vydání byl již v roce 1995 a napsán byl pomocí jazyka C, C++. Podporuje všechny operační systémy jako: Windows, GNU/Linux, Solaris nebo třeba macOS. Zajímavostí je že, jej využívají velké společnosti jako třeba YouTube, PayPal, Facebook, Twitter, Toyota, Netflix a tak dále. Velká výhoda je propojení MySQL s ORACLE Cloud, díky tomu můžeme k databázím přistupovat vzdáleně. Také můžeme přenášet data mezi MySQL a ORACLE databázemi. MySQL je k dispozici ve dvou verzích, a to volně dostupné a licencované verze, která nabízí více nástrojů pro správu databáze a také lepší zabezpečení dat. K vytvoření databáze v MySQL používáme jazyk SQL (Structured Query Language).<sup>18</sup>

**ORACLE Database** je jeden z nejstarších databázových systémů vytvořený roku 1979 společností Oracle Corporation. Vyvíjen byl v jazycích Java, C, C++ a Python. Podpora operačních systémů je multiplatformní což znamená, že může být spuštěn ve všech operačních systémech. Oracle databáze se programuje pomocí jazyka SQL ve vývojovém prostředí Oracle SQL Developer.

Zde také spravujeme svoje vytvořené databáze, přidělujeme hesla, přístupy a zakládáme nové tabulky se sloupci. Oracle databáze má velikou výhodu ve svém cloudovém řešení databáze. Zde můžeme spravovat databáze vzdáleným přístupem, a to i z mobilního zařízení. Práce s databází Oracle je velice intuitivní a většinu úkonů zde můžeme provádět z grafického rozhraní a SQL jazyk nemusíme využít. Pro složitější úkony je ale SQL výhodou.<sup>19</sup>

**MariaDB** je databáze vyvíjená komunitou a je následnicou větví MySQL. Práce na této databázi byla zahájena vývojáři MySQL, kteří měli obavu o fungování databáze MySQL, kterou odkoupil Oracle Corporation.

---

<sup>17</sup> Apache Software Foundation, 2023

<sup>18</sup> My SQL, 2023 (Why my SQL)

<sup>19</sup> Oracle Corporation, 2023

Tento databázový software má své licenční podmínky a vyvíjet jej může takřka kdokoliv, ale zavazuje se, že jeho práce bude spadat pod licenci MariaDB. První vydání tohoto softwaru bylo v roce 2009 vývojářskou společností MariaDB Corporation Ab. MariaDB je spustitelná na všech operačních systémech stejně jako Oracle či MySQL. Tento „open-source“ software ve velkém peněženě sponzoruje například Intel, Microsoft a IBM.<sup>20</sup> MariaDB si stále zachovává vysokou konektivitu s databázemi vytvořenými v softwaru MySQL. Jediný rozdíl je, že MariaDB od verze 10.0 pozměnila některé vývojové fáze tak aby byla stabilnější a méně náročná na požadavky hardwaru. Dále je nutné zmínit, že MariaDB přepracovala systém vyhledávání častých požadavků v paměti cache, a tak dokáže pracovat s menším objemem dat a rychleji vracet výsledky vyhledávání v databázi.<sup>21</sup>

**Microsoft SQL Server** je systém vyvinutý společností Microsoft v roce 1989 pomocí jazyků C Sharp, C a C++. Tento databázový software nabízí uživateli jednoduchou správu dat, uložení dat a analytickou činnost. Také je zde možností pracovat i v grafickém rozhraní Microsoft SQL Server Management Studio nebo Visual Studio. Programování vlastní databáze probíhá v jazyce SQL. Microsoft nabízí více edic tohoto softwaru. Zdarma dostupnou pro databáze o velikosti do 10 GB s názvem Express. Poté je verze Standart vhodná pro použití na vlastních serverech nebo v cloudovém úložišti. Dále Microsoft nabízí verzi Enterprise, která nabízí pokročilejší funkce vhodné pro velké podniky s vysokou náročností na kapacitu výkon a dostupnost. Poslední verzí je Developer, ta je určená pro vývoj a testování, nabízí všechny funkce edice Enterprise a je zcela zdarma.

### 4.3 MICROSOFT ACCESS

Microsoft Access který je součástí kancelářského balíčku Microsoft Office byl vytvořen roku 1992. Kombinuje grafické prostředí s relační databází Microsoft Jet Database. Jeho nespornou výhodou je právě již zmíněné grafické prostředí. Díky tomuto grafickému prostředí v tomto softwaru dokáže relační databázi vytvořit i úplný začátečník. Access byl naprogramován v jazyce C++. Další možností, jak si usnadnit práci s daty je ve formulářích Visual Basic. Tato funkce, která nám dovoluje programovat formuláře ve kterých zapisujeme, spravujeme nebo analyzujeme data nám nabízí velkou škálu možností, jak dále pracovat s daty. Access také využívá jazyk SQL. Náklady na pořízení tohoto softwaru díky balíčku Office 365

---

<sup>20</sup> MariaDB Foundation, 2023

<sup>21</sup> Websupport s.r.o., 2021

je minimální oproti složitějším databázovým softwarům.<sup>22</sup> Další výbornou funkcí je propojení tabulek z programu Excel, které můžeme v reálném čase přesouvat do Microsoft Access. Při exportu jen stačí vybrat soubor Excelu a při každé aktualizaci v databázi Access dostaneme aktuální data. Stejně to funguje i když chceme propojit tabulky mezi dvěma databázemi Access. SQL Server a Microsoft Azure SQL nám poskytnou online úložný prostor, který můžeme zpracovat i z mobilního telefonu. Díky Microsoft Access můžeme celou databázi pomocí pár kliknutí přesunout do formátu, který podporuje Microsoft Excel a dále pracovat v tomto programu. Operační systém, ve kterém můžeme Access použít je primárně Windows. Pro zjednodušení vytvoření vlastní databáze v MS Access připravil Microsoft svým uživatelům spoustu různých šablon.<sup>23</sup>

---

<sup>22</sup> Luboš Marvan, 2021

<sup>23</sup> Microsoft, 2023 (Access)

## 5 PRAKTICKÁ ČÁST

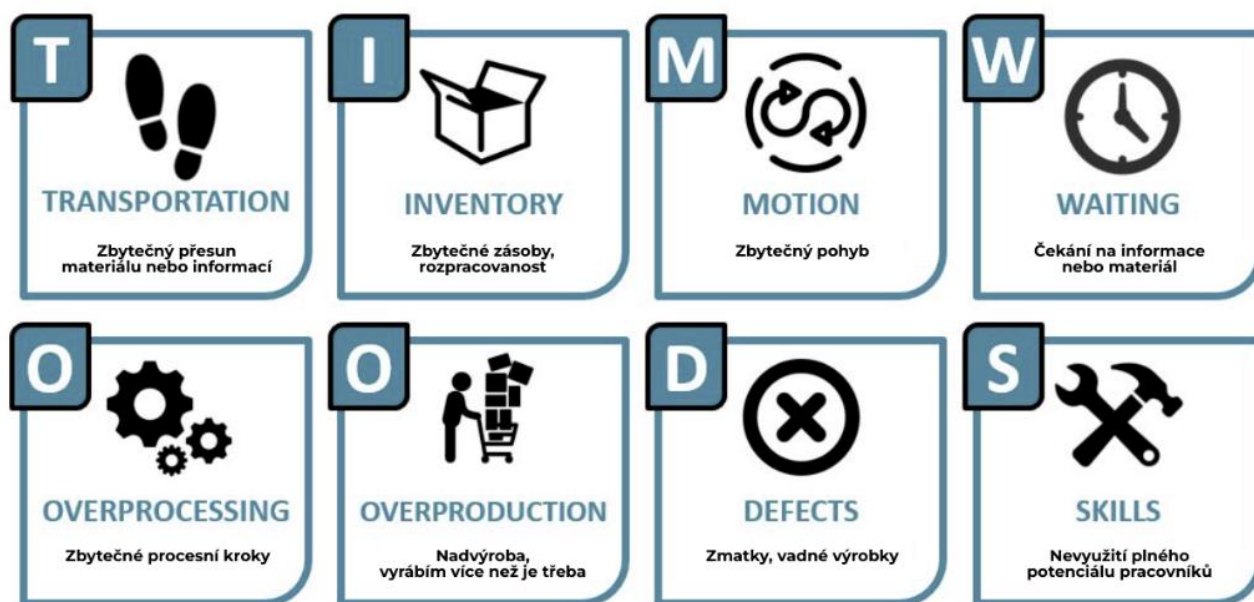
### 5.1 ÚVOD DO PRAKTICKÉ ČÁSTI

V praktické části navrhnu databázi v programu Access. Tato databáze bude sloužit k propojení tří pracovišť v procesu výroby. Budu demonstrovat všechny známé funkce programu Access. V pozdější části se zaměřím i na programování formulářů pomocí jazyka VisualBasic, který Access nabízí. Pokusím se zde v úvodu nastínit, jak bude vypadat práce zaměstnanců a jaké zlepšení jim databáze přinese. První zaměstnanec bude na konci výrobní linky, kde budou dojíždět již hotové výrobky. Toto pracoviště můžeme nazvat jako „balírna“. Zde bude probíhat ukládání výrobků do krabic a jejich následné balení. Druhé pracoviště bude sloužit k chystání krabic a balícího materiálu pro balírnu. Zde bude probíhat nachystání balení po vyžádání pracovníkem na balírně. Třetí pracoviště je logistika materiálu po výrobní hale. Toto pracoviště nám bude distribuovat nachystané krabice k výrobní lince.

Předchozí stav pracoviště je následující. Zaměstnanec chodí od své linky na chystání balení pěšky, aby si objednal bednu. To si pracovník, co bedny chystá zapíše na papír a po nachystání jde opět chůzí předat informaci na logistiku, že balení je nachystáno. Nevýhody jsou v tom že pracovníci si často nechají nachystat více kusů, aby se nezdržovali chozením s objednávkami. To má za následek, že vysokozdvizný vozík jezdí s velkými počty balení najednou. Poté nějakou dobu nejedí vůbec, než se balení zaplní výrobky. Proces expedice je tím velice zpomalen. Také zaměstnanci porušují nařízení o BOZP, protože se jim nechce jezdit vícekrát s menším počtem beden. Také dochází k zaskládání plochy kolem výrobních linek vyšším počtem kusů balení. To má také za následek porušení pravidel BOZP a bedny ve většině případů překáží ostatním pracovníkům, kteří procházejí kolem.

Mým cílem bude odstranit chyby, které se vážou na plýtvání. Toto plýtvání lze shrnout osmi písmeny TIMWOODS, je to jeden ze základních stavebních kamenů fungování úspěšné firmy. Zabývá se tím odvětví Lean Six Sigma. Toto plýtvání se nazývá osm druhů Lean Six

Sigma. TIMWOODS se zabývá zvýšením efektivity výrobního procesu a eliminuje plýtvání všech osmi písmen.

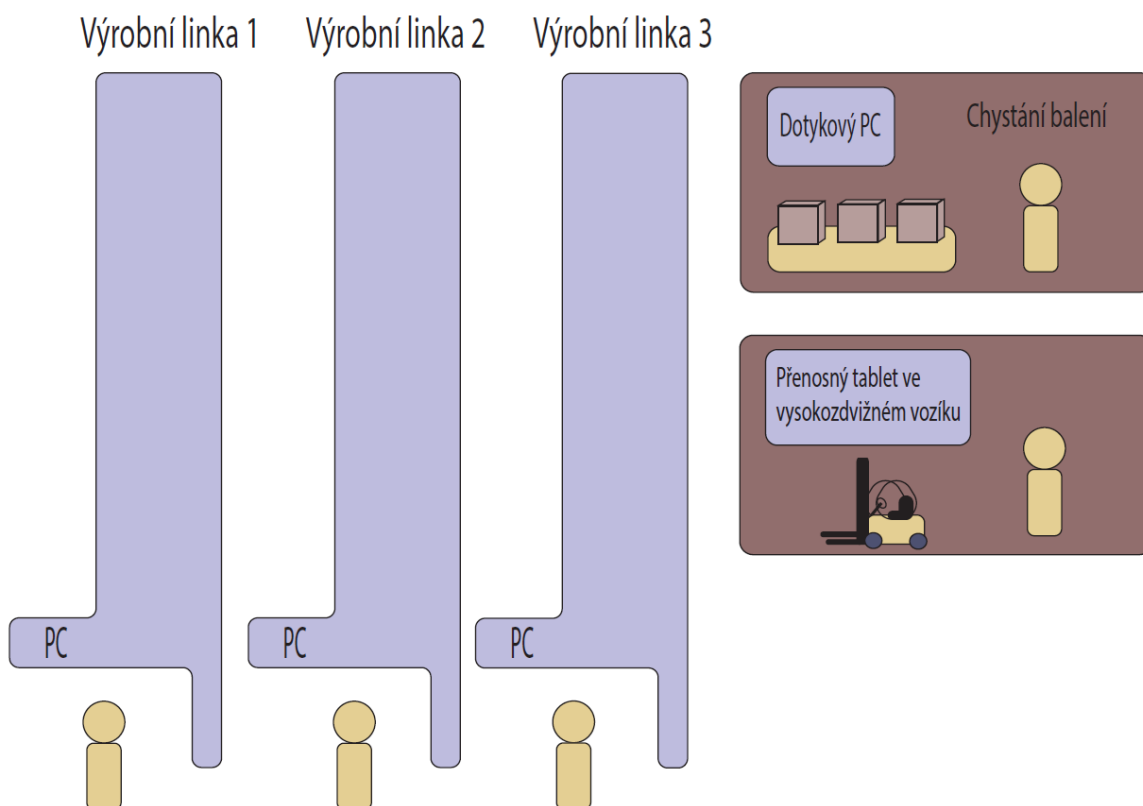


Obr. 8: TIMWOODS

Zdroj: <https://www.cems-cz.com/blog/476-timwoods-v-lean-six-sigma-8-druhu-plytvani>

Můj návrh řešení k ulehčení práce bude založen na třech formulářích propojených tabulkou v databázi. Pracovník na konci pásu (balírna) si objedná daný počet beden na balení. Bedny budou specifické podle odběratele. Zaměstnanec na přípravě balení potvrdí na dotykové obrazovce vyžádané bedny na balení. Po nachystání pomocí tlačítka potvrdí ukončení jeho práce. Toto potvrzení se zobrazí pracovníkovi logistiky na tabletu, který bude mít ve vysokozdvizném vozíku. Pracovník logistiky bednu naloží a doveze přímo k lince. Pro následnou kontrolu pracovníků, bude pro jejich nadřizového zaznamenáván čas proběhlých úkonů. Nejprve se zaznamená čas objednání, poté čas nachystání, a nakonec čas kdy bylo balení dovezeno. Tato databáze nám bude řešit složitost komunikace. Když zaměstnanec na balírně pracuje v nepřetržitém provozu nemá moc času na to, aby si šel objednat bednu sám. Možností je telefonicky si objednat bednu. Pokud ale na nachystání balení zrovna zavolá několik linek v časovém pořadí, se specifickými balíčky, je neřešitelné fungovat přes telefon. Mé řešení nabízí, vizuální představení objednávky bez zdlouhavého telefonování s promítnutím desítek posledních objednávek. Také nabízí časový přehled o průběhu práce dané směny s odhadem na

počet vyrobených kusů. Také nabízí archivaci vyexpedovaných balení s datem a časem. Návrh pracovních linek s chystáním balení a logistikou je na přiloženém obrázku č: 9.

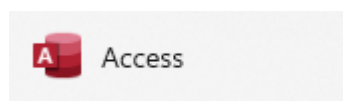


Obr. 9: Ukázka výrobního procesu

Zdroj: Vlastní tvorba

## 5.2 VYTVOŘENÍ DATABÁZE A TABULEK

V této podkapitole Vás provedu základními korky databáze Access. Vytvoříme prázdný návrh databáze. Je to důležité proto, že budeme celou databázi stavět na míru našim potřebám. Poté si založíme dvě tabulky a jednu z nich naplníme záznamy o typech balení v naší firmě. Druhá tabulka bude sloužit pro vyplňování. Zde budeme vkládat informace jako datum objednání, počet objednaných kusů balení, druh balení a tak dále. Důležitým krokem bude vytvoření relace mezi tabulkou s druhy balení a sloupcem druh balení v tabulce se záznamy.



Obr. 10: Ikona aplikace Access

*Zdroj: Aplikace Access*

Zapneme si Access pomocí této ikonky v našem počítači.

Na úvodní straně, kterou nás Access přivítá si vybereme ikonku s názvem „Prázdna databáze“.

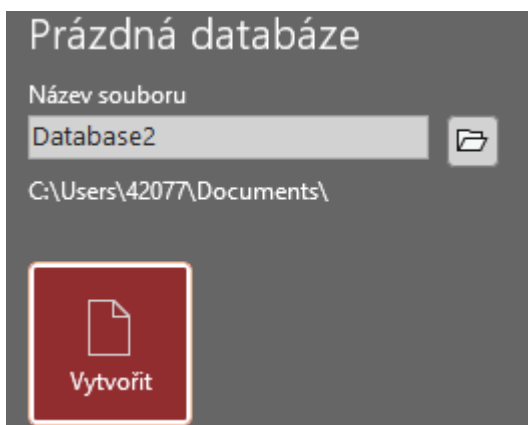
V ní není nahraná žádná šablona a celou databázi budeme vytvářet sami



*Obr. 11: Ikona prázdná databáze*

*Zdroj: Aplikace Access*

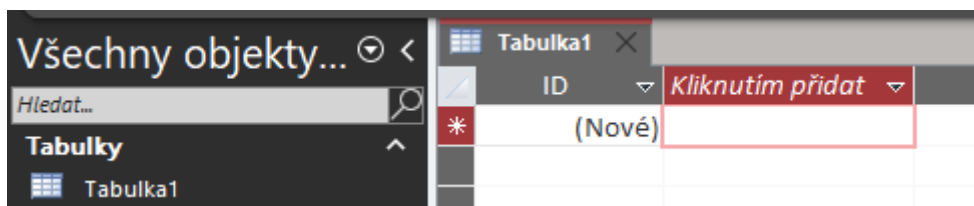
Po zvolení prázdné databáze se nás Access dotáže, kam chceme databázi uložit a jak ji chceme pojmenovat. Já si zvolím název „Návrh balení“ a databázi uložím do dokumentů.



*Obr. 12: Pojmenování databáze*

*Zdroj: Aplikace Access*

Po zvolení názvu a umístění souboru klikneme na „Vytvořit“. To nás převede do prázdné první tabulky v naší databázi. Tuto tabulku si společně začneme vyplňovat.

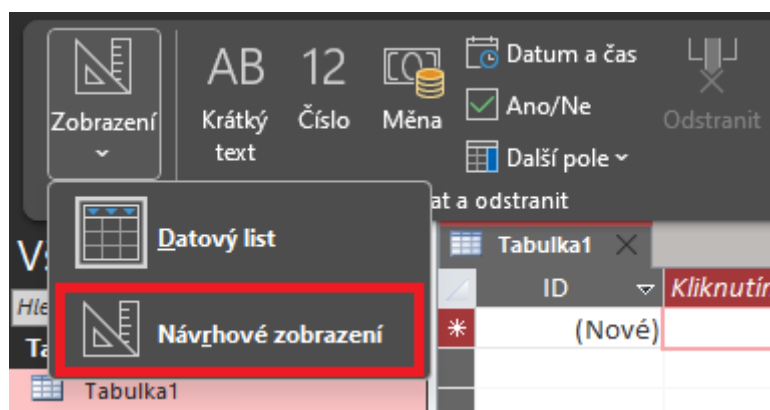


Obr. 13: Prázdná tabulka

Zdroj: Aplikace Access

Sloupec „ID“ si necháme pro speciální označení beden. Toto označení bude náš primární klíč, protože předpokládám, že každá bedna bude jiná. Minimálně v nějakých drobnostech a nebude se opakovat. Tudiž dvě rozdílné bedny nesmějí být označeny stejně, jinak bychom se v tom ani my nevyznali.

Pro lepší správu tabulky je dobré si ji přepnout do návrhového zobrazení. Návrhové zobrazení nabízí více možností, jak tabulku upravit oproti klasickému zobrazení. V klasickém zobrazení můžeme také přidávat sloupce a nastavovat je, ale chybí zde určité funkce a přehlednost sloupců.

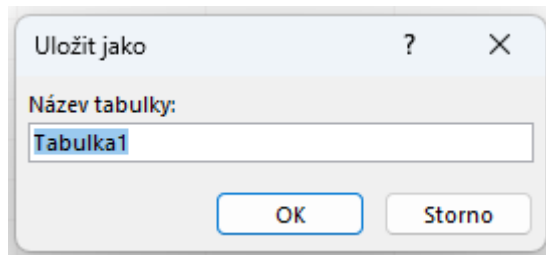


Obr. 14: Návrhové zobrazení

Zdroj: Aplikace Access

Když máme více tabulek je nutné si kliknout dvakrát levým tlačítkem myši na tu tabulku, kterou chceme upravovat a poté si ji přepnout do návrhového zobrazení. Po otevření dané tabulky v levém horním rohu narazíme na zobrazení. Vybereme si již zmiňované návrhové. To se ukrývá pod šipkou dolů, v obrázku jsem jej označil červeně.





Obr. 15: Uložení tabulky

Zdroj: Aplikace Access

Pro přepnutí do návrhového zobrazení musíme nejdřív tabulku uložit a pojmenovat. Já zvolím název „Druhy balení“.

Druhy balení	
Název pole	Datový typ
ID	Automatické číslo

Obr. 16: Datový typ primárního klíče

Zdroj: Aplikace Access

Pro náš účel bude lepší si datový typ primárního klíče přepnout na „Číslo“. Automatické číslo by nám pojmenovalo druhy balení jako: 1,2,3,4,5 .... To nechceme.

Název pole	Datový typ
ID	Automatické číslo
	Krátký text
	Dlouhý text
	Číslo
	Velké číslo

Obr. 17: Změna datového typu ID

Zdroj: Aplikace Access

Po přepnutí si můžeme ID balení zvolit sami nebo použít již to co máme vytvořené v rámci firmy.

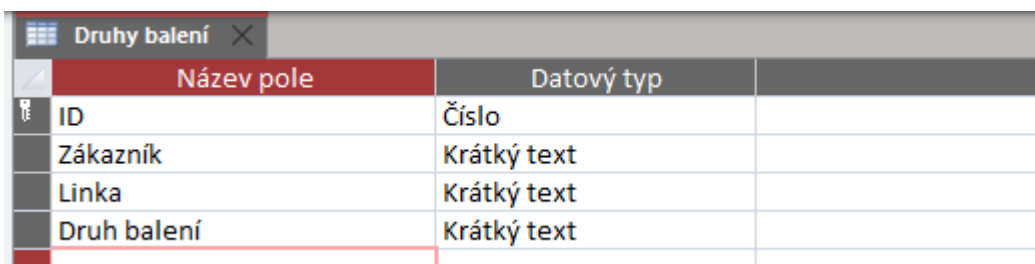
Když klikneme o řádek níže pod ID tak budeme vytvářet další sloupec naší tabulky. Ten můžeme pojmenovat třeba podle druhu odběratele „Zákazník“. Pokud naše firma má více odběratelů nejspíš každý bude vyžadovat jiné speciální balení, nebo se to bude lišit velikostí našich výrobků pro daného odběratele.

Dále uvedu sloupec s názvem „Linka“ zde bude uváděna Linka, na kterou balení patří. Většinou to funguje tak, že každá linka je stavěna na několik odběratelů. Další linka je stavěna na jiné odběratele, to zvýší celkovou produktivitu daných Linek.

Jako poslední sloupec bude „Druh balení“. Je podstatné vědět u jakého zákazníka máme jaké balení.

To se dále samozřejmě může ještě měnit podle použitých přebalových papírů v bedně a tak dále. Ale je podstatné uvést jaký druh bedny to bude. Například dřevěná bedna, ocelová, kartonová na paletě a tak dále.

Výsledná tabulka bude vypadat takto, viz obrázek 18 níže.

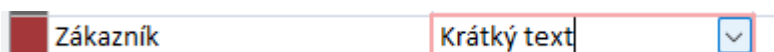


	Název pole	Datový typ
ID		Číslo
Zákazník		Krátký text
Linka		Krátký text
Druh balení		Krátký text

Obr. 18: Tabulka

Zdroj: Aplikace Access

Pro veliké usnadnění práce a snížení vypisování zde přidáme námi předvolené zákazníky, kteří od nás odebírají, druhy balení, které používáme a druhy linek, které máme. To nám urychlí zadávání dat do tabulky.



Obr. 19: Sloupec Zákazník

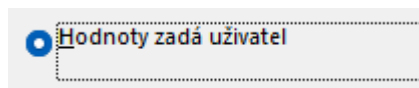
Zdroj: Aplikace Access

**Průvodce vyhledáváním.**

Obr. 20: Datový typ průvodce vyhledáváním

Zdroj: Aplikace Access

Pro přidání předvolených záznamů najedeme na sloupec ve, kterém tuto funkci chceme uplatnit. Klikneme na šipku vedle „Krátký text“ a vybereme „Průvodce vyhledáváním“ Obr. 20. Poté se nám objeví dialogové okno s výběrem, zdali chceme čerpat data z jiné tabulky nebo „Hodnoty zadá uživatel“. My vybereme druhou možnost.



Obr. 21: Výběr hodnot

Zdroj: Aplikace Access

Po výběru klikneme na tlačítko další.

Sloupec 1
Zákazník 1
Zákazník 2
Zákazník 3
Zákazník 4
Zákazník 5
Zákazník 6
Zákazník 7

Obr. 22: Návrh hodnot v poli

Zdroj: Aplikace Access

Zde budeme přidávat hodnoty, které dále nebudeme muset vyplňovat. I když budeme mít hotovou tuto část, můžeme sloupec vyplňovat jak pomocí zadaných hodnot, tak ručně.

Po vypsání klikneme na tlačítko dokončit, protože další nastavení pro nás není potřebné.

To samé provedeme i ve sloupcích „Linka“ a „Druh balení“.

Do Linky jsem si vypsál označení linek: Linka 1, Linka 2, Linka 3, Linka 4.

Do druhů balení už jsem zadal konkrétnější hodnoty z praxe a to „Kartonová krabice 5VVL“, „Krabice DFT-5“, „Zákazník 2 zelený gitterbox“, „Zákazník 3 růžový gitterbox“, „Dřevěná bedna“, „Paleta“.

ID	Zákazník	Linka	Druh balení	Kliknutím přidat
751234	Zákazník 1	Linka 1	Kartonová krabice 5VVL	
546328	Zákazník 1	Linka 1	Zákazník 2 zelený gitterbox	
465930	Zákazník 2	Linka 1	Zákazník 3 růžový gitterbox	
326589				
*	0	Zákazník 1		
		Zákazník 2		
		Zákazník 3		
		Zákazník 4		
		Zákazník 5		
		Zákazník 6		
		Zákazník 7		

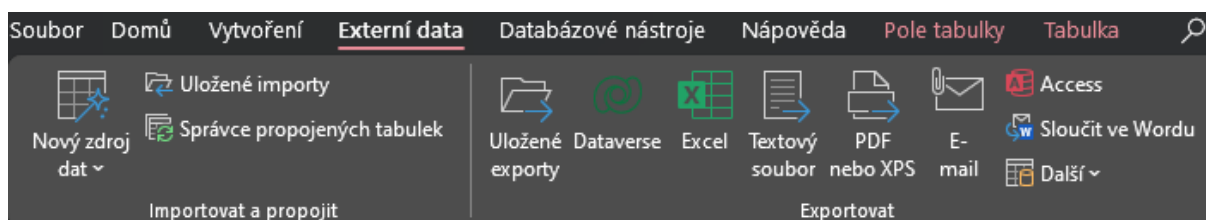
Obr. 23: Ukázka upravené tabulky

Zdroj: Aplikace Access

Na výše uvedeném obrázku je vidět, jak mohou záznamy vypadat a jak vypadá možnost zadávání předpřipravených hodnot. Do tabulky doplním několik fiktivních záznamů pro další práci s databází.

### Nahrání dat z jiného zdroje

Také je možnost data nahrát z „excelu“ nebo jiného programu a mít rovnou hotovou tabulku. Samozřejmě pokud máme již hotová data v jiném programu.

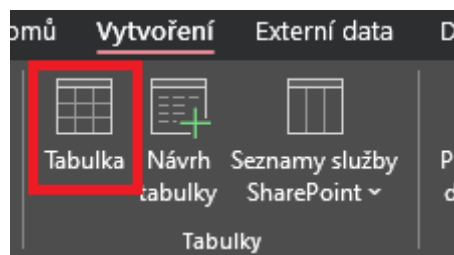


Obr. 24: Ukázka vložení dat

Zdroj: Aplikace Access

Tuto možnost máme v záložce „Externí data“. V kolonce Importovat a propojit se nachází tlačítko s názvem „Nový zdroj dat“. Po rozkliknutí šipky si vybereme odkud chceme zdroj dat nahrát a v z jakého programu. Také je možností vybrat jinou databázi Access a z ní tabulku. Tuto tabulku můžeme i sdílet v reálném čase a data se nám budou aktualizovat podle záznamů z databáze z které čerpáme. Naše data můžeme také Exportovat do různých programů a mailů. Což se nám hodí v momentě, kdy chceme data pouze sdílet s někým komu nechceme dát přístup do celé databáze, nebo daný člověk nemá možnost mít Access.

Další postup ve vytváření databáze bude založení druhé tabulky. Do této tabulky budeme ukládat nově vytvořená data. Můžeme ji pojmenovat jako „Záznamy“.



Obr. 25: Ukázka tvorby tabulky

Zdroj: Aplikace Access

Takže půjdeme do záložky „Vytvoření“ a klikneme na tlačítko Tabulka.

Tím se nám automaticky vytvoří nová prázdná tabulka. Zde je jen sloupec ID. Ten se vytváří vždy automaticky aby Access měl rovnou sloupec s primárním klíčem. Toho využijeme a ID si zanecháme. Dále jej upravovat nebudeme a ponecháme automatické číslo. To se nám bude hodit k tomu abychom věděli kolikátý záznam jsme již v naší tabulce vytvořili.

V tuto chvíli budeme tabulku opět nastavovat a vytvářet sloupce. Takže si ji přepneme do „Návrhového zobrazení“. Postup je stejný jako u minulé tabulky. Tato tabulka bude propojovat všechny tři Formuláře pro všechny tři jednotlivá pracoviště. Takže zde musíme vytvořit sloupce pro všechny tyto pracoviště. V této tabulce budeme chtít vytvořit sloupce Počet kusů balení, ID balení, Druh balení, Linka, Zákazník, Čas objednání, Čas nachystání, Čas dovozu. Vysvětlím proč zrovna tyto sloupce.

Počet kusů balení chceme z toho důvodu, aby bylo zřetelně jasné kolik kusů daného balení potřebujeme dovézt. ID balení nám bude sloužit k vyhledání balení z první vytvořené tabulky „Druhy balení“. Druh balení, aby bylo jasné, jaké balení potřebujeme. Sloupec Linka vytvoříme proto aby pracovník logistiky věděl, kam vyhotovené balení má odvézt. Zákazník se bude zobrazovat pro kontrolu, zda jsme zvolili správné balení. Také je důležité tuto informaci zobrazit z důvodu, že můžeme mít nové zaměstnance, kteří si nepamatují nebo neznají ID daného balení, ale orientují se podle zákazníka. Časy vytvoříme z kontrolních důvodů. Také mohou sloužit v nepřetržitém provozu k ujasnění, které balení se má udělat dříve podle času kdy bylo objednáno.

Druhy balení		Záznamy
	Název pole	Datový typ
☑	ID	Automatické číslo
	Počet kusů balení	Krátký text
	ID balení	Krátký text
	Druh balení	Krátký text
	Linka	Krátký text
	Zákazník	Krátký text
	Čas objednání	Krátký text
	Čas nachystání	Krátký text
	Čas dovozu	Krátký text

Obr. 26: Ukázka vytvořené tabulky "Záznamy"

Zdroj: Aplikace Access

Zde je vytvořená tabulka. Všechny sloupce až na „ID balení“ ponecháme jako datový typ krátký text. Je to z toho důvodu, že s nimi budeme dále pracovat v programovém prostředí a některé datové typy by nás mohly omezovat.

S ID balení budeme dále pracovat. Budeme zde vytvářet relaci díky které propojíme záznamy z tabulky „Druhy balení“ s tímto sloupcem v tabulce „Záznamy“.

Klikneme na pole „ID balení“ a Datový typ změním na „Průvodce vyhledáváním“.

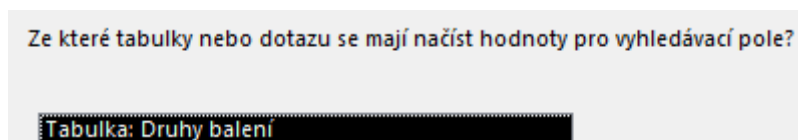
Po změně na „Průvodce vyhledáváním“ se nám otevře okno kde se nás Access dotáže, jestli chceme použít hodnoty z jiné tabulky. Toto pole je automaticky vybráno.



Obr. 27: Průvodce vyhledáváním

Zdroj: Aplikace Access

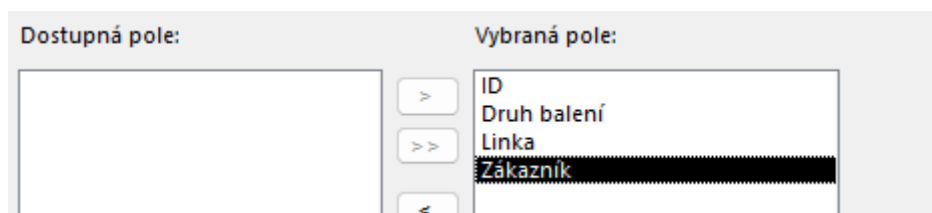
Pokračujeme stisknutím tlačítka Další.



Obr. 28: Výběr tabulky

Zdroj: Aplikace Access

Jako další máme na výběr danou tabulku. My si opět necháme první volbu a to tabulku „Druhy balení“. Pokračujeme stisknutím tlačítka Další.



Obr. 29: Výběr polí

Zdroj: Aplikace Access

V další části si volíme pole, které chceme dát do relace. Je dobré si je seřadit podle posloupnosti v tabulce „Záznamy“. Ulehčí nám to programování ve formuláři. Pořadí v rozevíracím seznamu necháme stejné jako navolení sloupců do relace. Opět zachováme posloupnost z tabulky „Záznamy“.

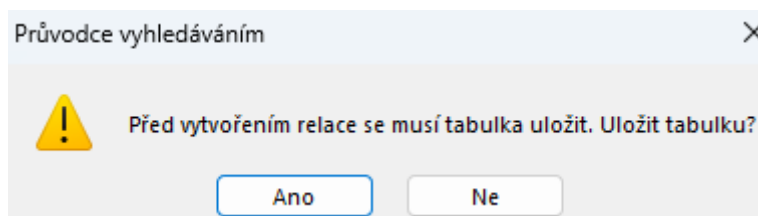
**Skrýt klíčový sloupec (doporučeno)**

ID	Druh balení	Linka	Zákazník
123154	Paleta	Linka 3	Zákazník 3
147963	Paleta	Linka 4	Zákazník 7

Obr. 30: Ponechání klíčového sloupce

Zdroj: Aplikace Access

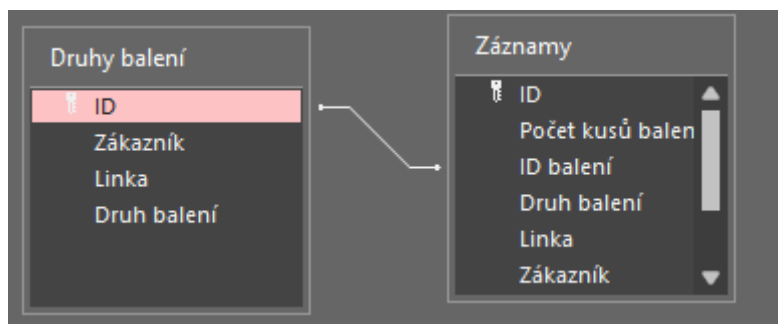
Po pokračování na další krok se nás Access zeptá, jestli chceme skrýt klíčový sloupec. Tento sloupec skrýt nechceme, podle ID budeme vyhledávat druhy balení. Má to jednu velikou výhodu a to tu, že můžeme použít skener čárových kódů a nemusíme nic vyplňovat. Po zanechání ID můžeme kliknout na dokončit.



Obr. 31: Uložení tabulky

Zdroj: Aplikace Access

Uložíme tabulku a máme hotovo.



Obr. 32: Ukázka relace

Zdroj: Aplikace Access

Takto vypadá vytvořená relace. Tuto relaci nazýváme „1:1“ a to proto že, jeden záznam (objednávka), který vytvoříme má v sobě jeden daný druh balení.

ID	Počet kusů	ID balení	Druh balení	Linka	Zákazník	Čas objedná	Čas nachyst	Čas dovozu
1		147963						
(Nové)		123154	Paleta	Linka 3	Zákazník 3			
		147963	Paleta	Linka 4	Zákazník 7			
		326589	Kartonová krab	Linka 2	Zákazník 2			

Obr. 33: Ukázka tabulky "Záznamy"

Zdroj: Aplikace Access

Takto vypadá naše tabulka. Máme zde všechny potřebné sloupce. Také zde můžete vidět vytvořenou relaci. Tato relace v programu Access funguje tak že když vyplňujeme „ID balení“ zvolíme si podle ID balení přesně to balení, které potřebujeme.

V dalším kroku se budeme věnovat vytvoření formuláře pro „Objednání balení“.

### 5.3 VYTVOŘENÍ FORMULÁŘE A JEHO ÚPRAVA

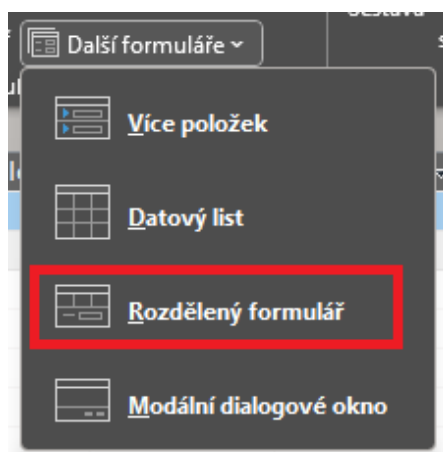
Nejprve popíšu, co budeme dále dělat a proč. Vytvoříme si nový formulář, jako první pro „Objednání balení“. Začneme tímto formulářem abychom si představili, jak postupně probíhá pracovní proces. I podle něj se budeme řídit. Jak jsem již nastínil v úvodu, tak nejdříve si objednáme balení poté jej nachystáme a na konec jej dopravíme na místo určení. Vytvoříme si obyčejný prázdný formulář a vše si vytvoříme a nastavíme sami. Budeme chtít mít tlačítka na vytvoření nové objednávky a na její uložení. To nám bude pro objednání stačit. Dále budeme od formuláře požadovat abychom vyplňovali ručně co nejméně informací a vlastně co nejvíce jich za nás vyplnila databáze. Tudiž se pustíme i do programování v jazyce VisualBasic, který nám Access nabízí. Zde si naprogramujeme automatické vyplňování určitých polí. Budeme



chtít vyplnit všechny pole, které používáme z předchozí tabulky „Druhy balení“. Dále budeme chtít automaticky vyplnit datum a čas. Pokud bychom jej vyplňovali ručně tak by nás to stálo spoustu času. Nejdříve si musíme určit, ale při jaké události proběhnou všechny ostatní operace, které naprogramujeme. Myslím si, že nejvhodnější bude vše vyplnit, když vyhledáme správný typ balení. Takže tuto větu převedu do programu a to: při změně pole „ID balení“ proved' náš program. Podstatná informace pro zaměstnance u Linky bude i kdy bylo jeho balení nachystáno, aby odhadl, kdy zhruba může přijet. Pokud třeba bude mít hotovou práci a bude čekat na balení, bude se mu tato informace hodit.

### Vytvoření formuláře

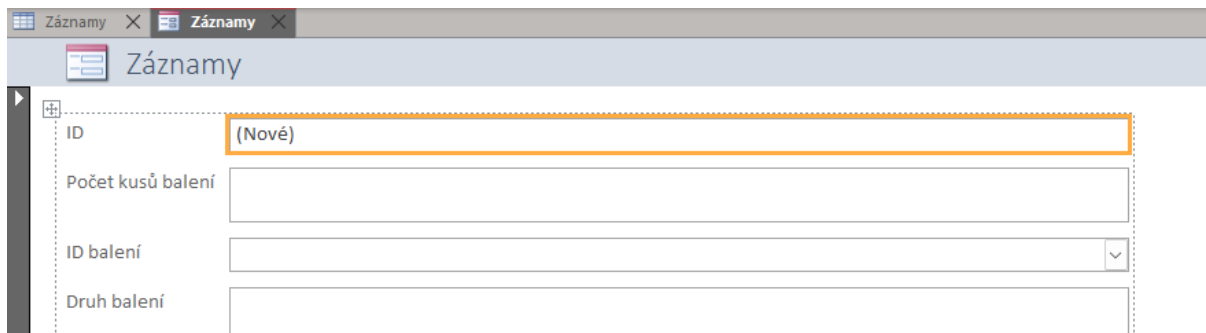
Nejsnadnější a nejrychlejší volbou pro náš účel bude automatické vytvoření formuláře, který následně upravíme. Access za nás pouze vytvoří návrh formuláře kam přidá všechny pole tabulky pro, kterou tento formulář vytváříme. Před vytvořením je důležité mít jedním klikem označenou tabulku pro, kterou tento formulář bude určen. Abychom zobrazili tabulku s informacemi ve Formuláři máme dvě možnosti, buďto vytvořit „Rozdělený formulář“, nebo do něj přidat „podformulář“. Vhodnější pro naše účely bude zvolit první variantu.



Obr. 34: Návod na založení Rozděleného formuláře

Zdroj: Aplikace Access

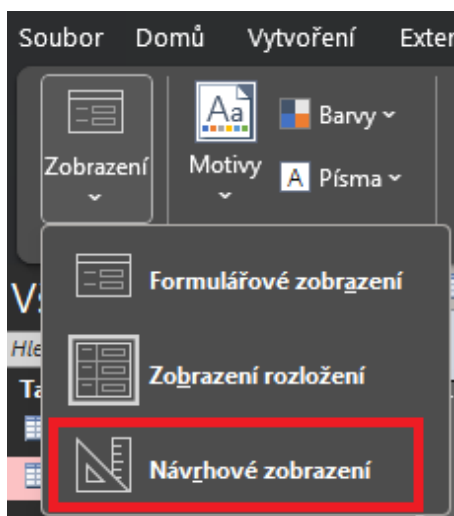
Formulář vytvoříme v kartě „Vytvoření“ v sekci „Formuláře“, kliknutím na rozbalovací tlačítko „Další formuláře“, zde zvolíme „Rozdělený formulář“ (označeno červeně viz. Obr. 34).



Obr. 35: Vytvořený formulář

Zdroj: Aplikace Access

Po kliknutí se nám objeví náš nový formulář se všemi poli z tabulky. Půjdeme do návrhového zobrazení a dále budeme tento formulář upravovat.



Obr. 36: Ukázka přepnutí do návrhového zobrazení

Zdroj: Aplikace Access

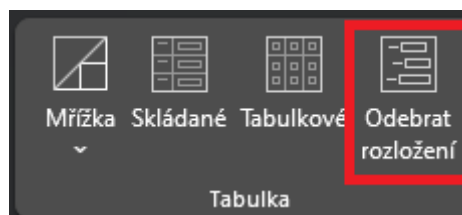
Návrhové zobrazení se nachází v levé horní části v rozbalovacím okně tlačítka „Zobrazení“.



Obr. 37: Označení mřížky

Zdroj: Aplikace Access

Zde si označíme mřížku a odebereme její rozložení. Budeme moci dále manipulovat s poli dle našich potřeb. Toto rozložení odebereme v záložce „Uspořádání“ v sekci „Tabulka“.



Obr. 38: Odebrání rozložení

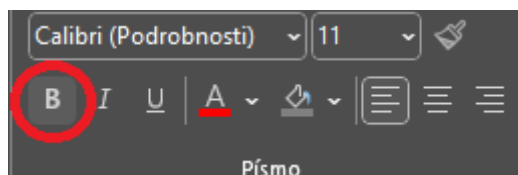
Zdroj: Aplikace Access

Teď si můžeme pole uspořádat dle libosti. Podstatné bude změnit velikosti daných polí, některé jsou podstatnější než jiné a my chceme na první pohled zaujmout zaměstnance a naši databázi udělat intuitivní. Popíšu možné rozložení formuláře.

Pole ID si myslím, že není tak podstatné, aby bylo veliké. Je to z toho důvodu, že k práci našeho zaměstnance nepřináší žádné podstatné informace. Jedná se pouze o počet již vytvořených objednávek v naší databázi. Já volím cestu zmenšení a přesunutí do levého horního rohu.

ID Balení je podstatná informace a první věc kterou budeme chtít vyplňovat proto jej umístím pod ID a velikost zanechám stejnou. To samé si myslím o poli „Druh balení“, je to podstatná informace, tudíž také zanechám stejnou velikost. Toto pole umístím hned z pravé strany vedle pole ID balení. Pole Linka umístím hned pod pole ID balení, opět zanechávám stejnou velikost. A to samé provedu i s polem „Zákazník“ je to pro zaměstnance také velice podstatná informace. Toto pole umístím z pravé strany k poli Linka.

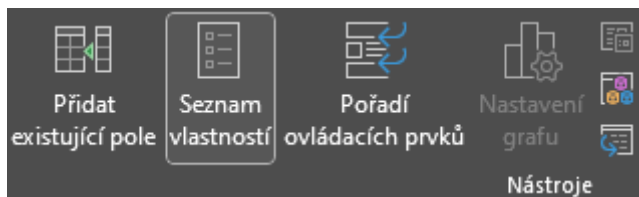
Všechny časy si seřadím vedle sebe do spodní části formuláře a změním pouze jejich výšku na nižší. V tento moment si označím všechna pole až na ID a změním jejich písmo na tučné, ať je v provozu lépe viditelné a při práci se dá přečíst i z větší dálky, nebo jiných úhlů. V horní liště zvolím kartu „Formát“ a v sekci písmo budu dále pracovat. Já nastavím vše na tučné a černé, protože popisky jsou automaticky nastaveny na šedou barvu.



Obr. 39: Tučné písmo

Zdroj: Aplikace Access

Poté si označíme pole „ID balení“ a v záložce „Návrh formuláře“ klikneme na „Seznam vlastností“.



Obr. 40: Výběr seznamu vlastností

Zdroj: Aplikace Access

V pravé části obrazovky se nám zobrazí okno seznamu vlastností. Zde budeme mít automaticky vybranou možnost „Událostní“.

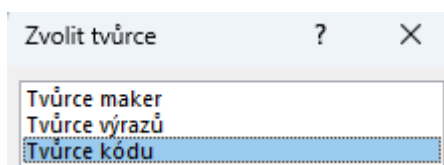


Obr. 41: Karta "Událostní"

Zdroj: Aplikace Access

Klikneme na tři tečky v kolonce „Při změně“ označené v obr.:41 modře.

Access se nás dotáže na volbu tvůrce, zvolíme „Tvůrce kódu“.

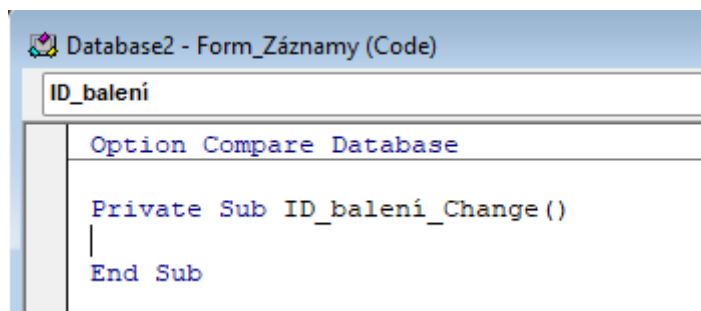


Obr. 42: Výběr tvůrce kódu

Zdroj: Aplikace Access

Zde budeme programovat automatické vyplňování polí, jak jsem již nastínil.

Po vybrání této možnosti se nám zobrazí programovací prostředí, které využívá jazyk VisualBasic. Zde si vytvoříme vlastní program, složený z pár jednoduchých vět.

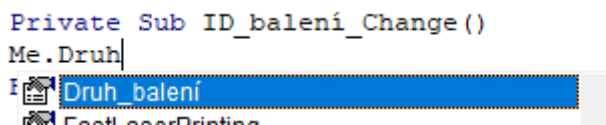


Obr. 43: Programové prostředí jazyka VBA

Zdroj: Aplikace Access

Mezi částí Private Sub a End Sub se píše program, je to označení začátku a konce programu. Píšeme jej pro pole ID\_balení, když proběhne změna (change). Do programu napíšeme větu: „Me.Druh\_balení.Value = Me.ID\_balení.Column(1)“

Tato věta znamená, že hodnota pole „Druh balení“ bude rovna hodnotě sloupce (1) v poli „ID balení“. Takže po vybrání daného balení se nám automaticky změní jakákoli hodnota pole „Druh balení“ na danou hodnotu náležící námi vybranému ID balení.



Obr. 44: Výběr pole ve formuláři

Zdroj: Aplikace Access

V jazyce VisualBasic se nám po zadání tečky objevuje výběr možných polí či programů s kterými chceme dále pracovat. Po napsání věty do programu budeme dále pokračovat. Napíšeme stejné vety pro všechny pole, které budeme měnit. Výsledek bude vypadat takto:

```
Private Sub ID_balení_Change()
Me.Druh_balení.Value = Me.ID_balení.Column(1)
Me.Linka.Value = Me.ID_balení.Column(2)
Me.Zákazník.Value = Me.ID_balení.Column(3)
End Sub
```

Po zvolení balení se automaticky vypíší všechny čtyři pole, které jsme zadali do programu.

Obr. 46: Ukázka vypsaných záznamů

Zdroj: Aplikace Access

Vždy měníme hodnotu daného pole, jen určujeme, který sloupec této hodnotě z pole „ID balení“ mu bude odpovídat.

V tuto chvíli bychom stejným způsobem mohli přidat i automatické vyplnění času, navážeme ho také na změnu pole „ID balení“. Je to jednodušší provedení, je možné tento čas ještě navázat na tlačítko uložit. V našem případě to dělat nebudeme, protože tlačítko uložit budeme mít automaticky přidáno programem Access a bude využívat místo jazyka „VisualBasic“ Makra. Je to usnadnění a pro kontrolu správnosti se nám bude hodit, že čas uvidíme o chvíli dříve, než uložíme záznam.

Opět otevřeme stejným způsobem programovací okno pro pole „ID balení“ při změně. Do programu pod poslední námi napsanou větu „Me.Zákazník.Value = Me.ID\_balení.Column(3)“ napíšeme nový kód programu.

```
Me.Čas_objednání.Value = Now()
```

Tahle věta nám říká, že hodnota pole „Čas objednání“ bude rovna času, který je přesně na vteřinu teď ( Now). Po zvolení ID balení (až proběhne změna pole) se nám společně s Druhem balení, Zákazníkem a Linkou také vyplní Čas objednání. Bude to vypadat takto:

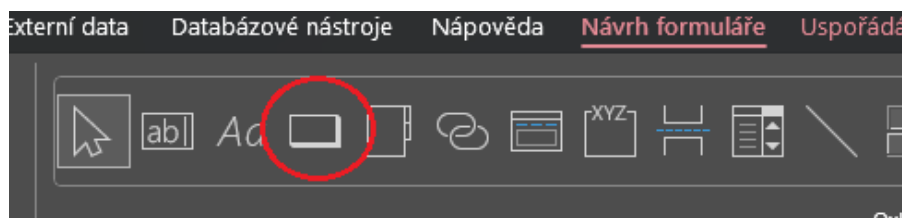
**Čas objednání**

Obr. 47: Ukázka vypsaného formátu času

Zdroj: Aplikace Access

V tuto chvíli máme základní programovací část hotovou a budeme přidávat tlačítka na nový záznam a uložení záznamu. Tyto tlačítka jsou již připravena a my je pouze využijeme. Budeme

stále pokračovat v Návrhovém zobrazení. Přejdeme na kartu „Návrh formuláře“ zde si vybereme ikonu tlačítka viz. Obr.:48.



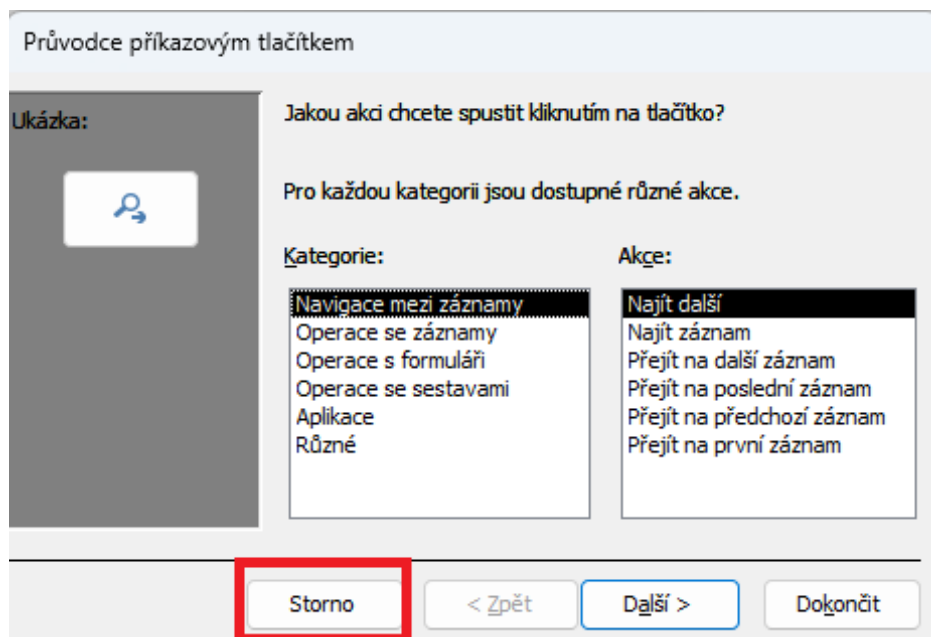
Obr. 48: Tvorba tlačítka

Zdroj: Aplikace Access

Po kliknutí na tuto ikonu se nám zobrazí místo kurzoru křížek, kterým vybereme jak velké tlačítko a v jaké části formuláře chceme. Po tomto výběru se otevře okno s nabídkou možných druhů tlačítek, které můžeme použít. My si zvolíme kategorie: „Operace se záznamy“ akce: „Uložit záznam“ pokračujeme stiskem tlačítka další. Na další stránce máme výběr, zda chceme mít v ikoně tlačítka nápis nebo obrázek. Já zvolím nápis a změním jej na „Objednat“. Ne každý zaměstnanec musí obrázek pochopit, tudíž volím tuto variantu. Po pokračování dále si volíme přímý název tlačítka, to je dobré při dalším programování, pokud s tímto tlačítkem budeme pracovat.

My jej dále využívat nebudeme tudíž můžeme nechat název na „Příkaz“ a klikneme na tlačítko „Dokončit“. Máme vytvořené tlačítko, je ale malé a nevýrazné což pro usnadnění práce není vhodné. Proto jej zvětšíme a změním velikost písma na 16 a dáme ho tučně. Stejně budeme pokračovat i s dalším tlačítkem na nový záznam ve formuláři. Postup bude stejný. Zvolíme ikonu tlačítka, zadáme velikost a pozici. Poté vybereme kategorii: „Operace se záznamy“ Akce: „Přidat nový záznam“. V dalším kroku si vybereme opět jestli chceme text či obrázek. Já zvolil text a zadal jsem „Nová objednávka“. Vizuální nastavení tlačítka bude stejné, a to velikost písma na 16 a tučné zvýraznění.

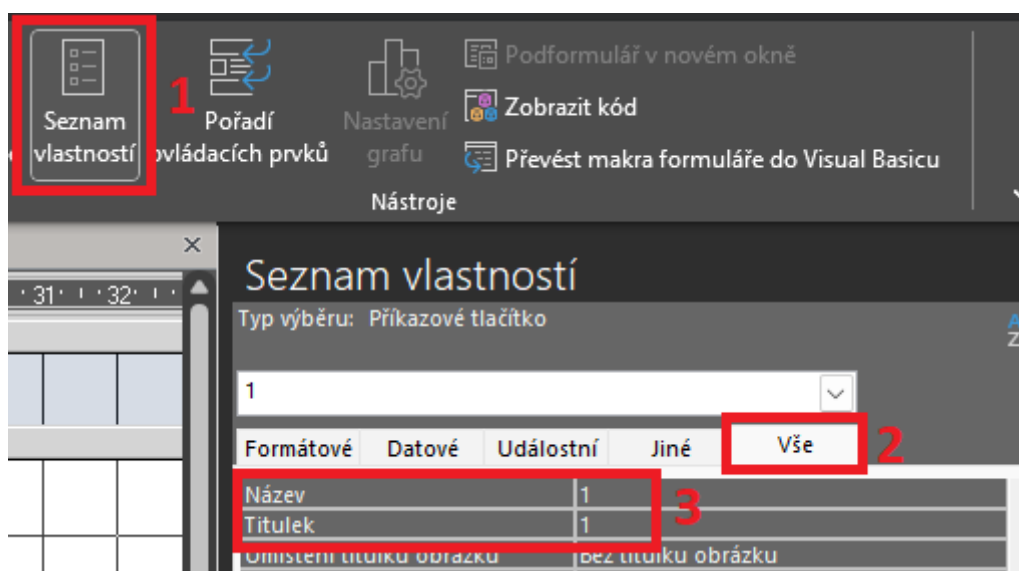
Dále nám ještě zbývají tlačítka na počet kusů beden. Zvolil jsem tento způsob, protože se domnívám, že je to intuitivnější a rychlejší pro zaměstnance než zadávat číslo na klávesnici. Postup bude obdobný jako u předchozích tlačítek. S tím rozdílem že při vytváření si nevybereme žádné přednastavené tlačítko.



Obr. 49: Výběr tlačítka bez programu

Zdroj: Aplikace Access

V průvodci dáme pouze storno. To nám vytvoří tlačítko bez programu, který si do něj vložíme sami. Po vytvoření prázdného tlačítka přejdeme do seznamu vlastností do sekce vše a změníme jeho název. Access automaticky zadal název na „Příkaz číslo...“



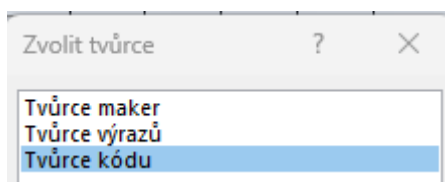
Obr. 50: Pojmenování tlačítka

Zdroj: Aplikace Access



Jelikož toto tlačítko bude sloužit k zadávání počtu objednaných kusů balení pojmenujeme jej jako „1“ (Jeden kus balení).

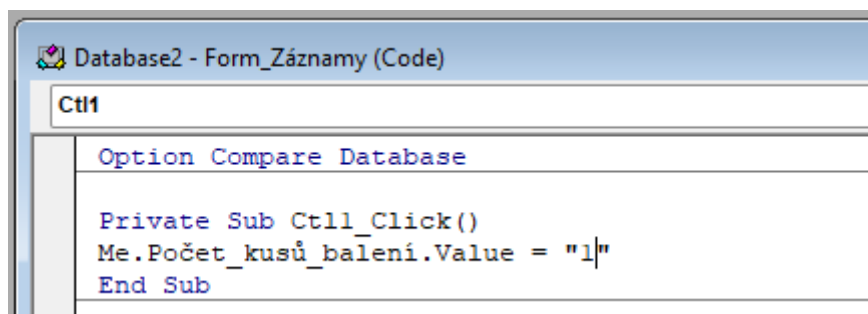
Další postup bude přejít do karty „Událostní“ a „Při kliknutí“. Zde vytvoříme program na změnu hodnoty pole na námi vybranou číslovku. Vše budeme opět vytvářet v programovém prostředí jazyka VisualBasic.



Obr. 51: Výběr tvůrce kódu

Zdroj: Aplikace Access

Program bude podobný jako předchozí, protože budeme opět měnit hodnotu. Jen s tím rozdílem že hodnota bude rovna jedné.



Obr. 52: Ukázka hotového kódu

Zdroj: Aplikace Access

```
Me.Poččet_kusů_balení.Value = "1"
```

Tento postup zopakujeme třikrát a vždy jen změníme konečnou hodnotu na dané číslo, které chceme.

```
Takže: Me.Poččet_kusů_balení.Value = "2"
```

```
Me.Poččet_kusů_balení.Value = "3"
```

Výsledek bude vypadat takto:

Počet kusů balení

1 2 3

Obr. 53: Výsledná podoba tlačítek

Zdroj: Aplikace Access

Pro formulář Objednání balení, kde máme prvního zaměstnance, který si objednává balení bude vhodné zvýraznit v tabulkové části „Čas nachystání“.

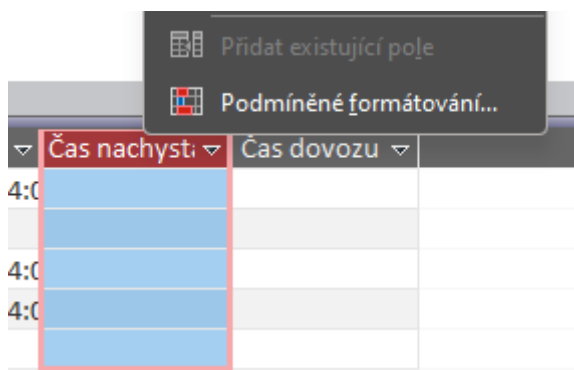
Toto zvýraznění upoutá jeho pozornost a bude v tabulce odlišné. Proto si do sloupce Čas nachystání přidáme podmíněné formátování. Nachystanou objednávkou označíme výraznou barvou, například zelenou.

ID	Počet kusů	ID balení	Druh balení	Linka	Zákazník	Čas objednání	Čas nachystání	Čas dovozu
1	3	465930	Zákazník 3 růžový glitterbox	Linka 1	Zákazník 2	06.03.2023 14:07:03		
2								
3		546328	Zákazník 2 zele	Linka 1	Zákazník 1	06.03.2023 14:07:03		
4		751234	Kartonová krabice	Linka 1	Zákazník 1	06.03.2023 14:07:03		
(Nové)								

Obr. 54: Označení sloupce v rozděleném formuláři

Zdroj: Aplikace Access

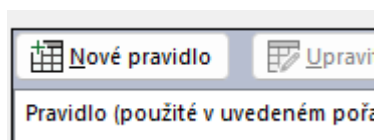
Do podmíněného formátování se dostaneme pomocí pravého kliknutí myši na sloupec v tabulce. Poté vybereme ve spodní části vyskakovacího pole „Podmíněné formátování“.



Obr. 55: Výběr Podmíněného formátování

Zdroj: Aplikace Access

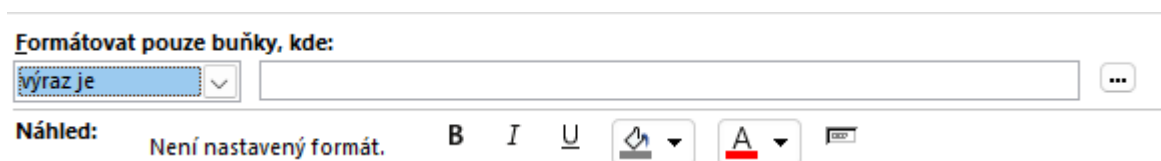
Zde vytvoříme nové pravidlo.



Obr. 56: Tvorba nového pravidla

Zdroj: Aplikace Access

Formát pravidla bude „výraz je“.



Obr. 57: Formát pravidla buňky

Zdroj: Aplikace Access

Zde si napíšeme naše pravidlo, které se bude vztahovat na celý sloupec.

`IsNull([Čas nachystání]) = False`

Vysvětlím tento výraz. Když je hodnota nulová v buňce Čas nachystání nepravda tak se změni barva pole na námi vybranou. Automaticky je nastavena barva na šedou. Já nastavil barvu na zelenou, jak jsem již zmínil.

Výsledek vypadá takto.

ID	Počet kusů	ID balení	Druh balení	Linka	Zákazník	Čas objedná	Čas nachyst.	Čas dovozu
1	3	465930	Zákazník 3 růžc	Linka 1	Zákazník 2	06.03.2023 14:0	123456	
2								
3		546328	Zákazník 2 zele	Linka 1	Zákazník 1	06.03.2023 14:0		
4		751234	Kartonová krať	Linka 1	Zákazník 1	06.03.2023 14:0		
(Nové)								

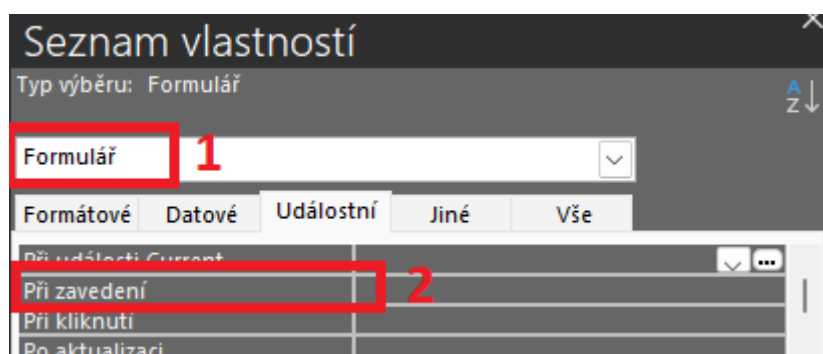
Obr. 58: Výsledek Podmíněného formátování

Zdroj: Aplikace Access

Nastavení je povoleno na jakoukoli hodnotu. To nám ale nevadí, protože my zablokujeme psaní do tohoto sloupce tudíž jej bude moci vyplňovat pouze námi vytvořený program a nedojde k záměně informací od zaměstnanců.

Teď si Formulář převedeme do maximálního zobrazení. Toto zobrazení bude mít výhodu v tom, že nebude vidět aplikace Access, ale pouze náš Formulář.

K tomuto zobrazení budeme potřebovat vytvořit jednoduchý program. Tuto práci nám Access velice zjednoduší. Převedeme Formulář do návrhového zobrazení a otevřeme si záložku Seznam vlastností. Zde budeme chtít upravovat celý Formulář, takže si jej vybereme v seznamu vlastností. Kdy budeme chtít, aby se formulář maximalizoval? Ve chvíli, kdy jej spustíme, takže budeme vytvářet pravidlo Při zavedení.



Obr. 59: Výběr pole "Při zavedení"

Zdroj: Aplikace Access

Při zavedení dáme tři tečky a přejdeme do Tvůrce kódu.

```
Private Sub Form_Load()
DoCmd.Maximize|
End Sub
```

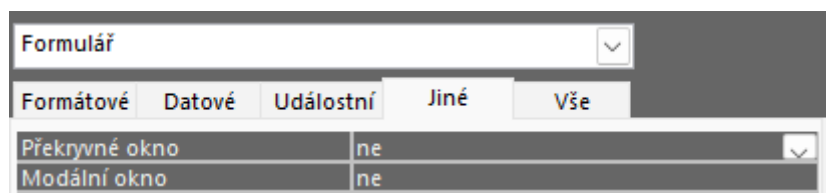
---

Obr. 60: Ukázka maximalizačního kódu

Zdroj: Aplikace Access

DoCmd.Maximize

DoCmd znamená: Data Object Command v češtině Příkaz datového objektu a maximize je maximalizace. Aby nám tento program, ale fungoval tak musíme v Seznamu vlastností v kartě jiné povolit Překryvané a Modální okno. Automaticky je to nastaveno na „ne“.



Obr. 61: Změna formuláře na okno

Zdroj: Aplikace Access

Tuto hodnotu změním na „ano“.

V tuto chvíli po uložení formuláře se nám při každém Formulářovém zobrazení náš formulář zobrazí přes celou obrazovku. To je velká výhoda a můžeme si díky tomu udělat masku naší databáze. Masku budeme vytvářet po dokončení ostatních Formulářů.

Zjednodušením by pro naše zaměstnance mohlo být, že hned po novém zapnutí se Formulář zobrazí na přidání nového záznamu. Access má automaticky nastaveno při otevření formuláře, zobrazení prvního záznamu v tabulce. To znamená, že pokaždé když se formulář zapne musíme kliknout na tlačítko s novým záznamem, v našem případě „Nová objednávka“. V některých případech je to dobré, v našem případě to akorát musíme vždy změnit.

Pro provedení této změny musíme otevřít návrhové zobrazení, a jít do seznamu vlastností. Zde si vybereme záložku Formulář a kartu Událostní. Náš program budeme chtít spouštět hned při otevření Formuláře, tudíž budeme programovat řádek“ Při zavedení“.



Obr. 62: Programování při zavedení

Zdroj: Aplikace Access

V této části jsme již programovali zvětšení formuláře na fullscreen teď pod tento program přidáme program nový.

```
Private Sub Form_Load()
DoCmd.Maximize
DoCmd.GoToRecord , , acNewRec
End Sub
```

Obr. 63: Ukázka kódu "Přejít na nový záznam"

Zdroj: Aplikace Access

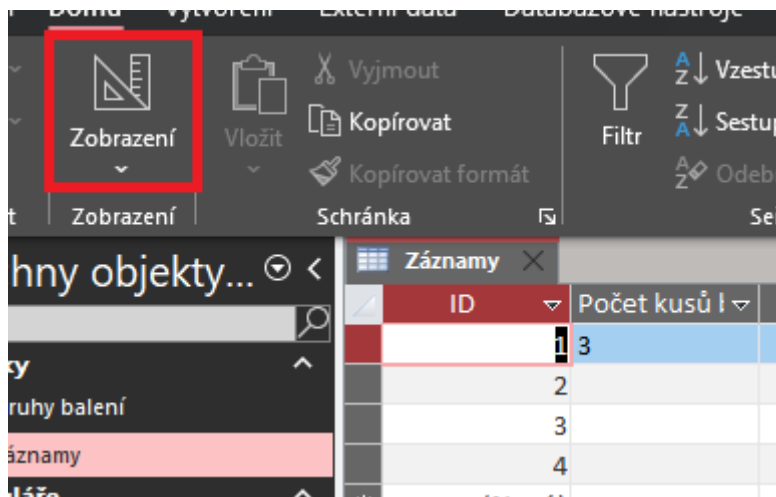
DoCmd.GoToRecord , , acNewRec

Příkaz datového objektu. Běž na záznam , , Nový záznam

Formulář máme ze základu hotový a v závěru jen upravíme drobnosti co budou třeba a přidáme zlepšující nápady pro snazší zacházení nebo pro usnadnění výkonu práce. To vyhodnotím až po vytvoření všech formulářů do základního nastavení.

Budeme pokračovat vytvořením formuláře pro Přípravu balení. Než jej vytvoříme bylo by dobré přidat sloupec pro potvrzení nachystání balení.

Takže přejdeme do tabulky „Záznamy „a dáme si ji do návrhového zobrazení.



Obr. 64: Změna na "Návrhové zobrazení"

Zdroj: Aplikace Access

Vytvoříme nový sloupec „Potvrzení nachystání“ a jeho Datový typ změním na Ano/Ne. Tím vytvoříme v tabulce zaškrtačací políčko.

Potvrzení nachystání	Ano/ne	
----------------------	--------	--

Obr. 65: Změna na zaškrtačací tlačítko

Zdroj: Aplikace Access

Čas dovozu	Potvrzení nachystání	Kliknutím př
	<input checked="" type="checkbox"/>	
	<input type="checkbox"/>	
	<input type="checkbox"/>	
	<input type="checkbox"/>	

Obr. 66: Ukázka sloupce v tabulce se zaškrtačacím polem

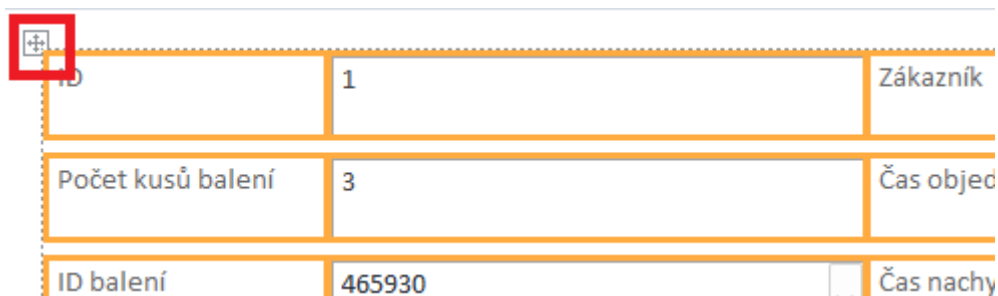
Zdroj: Aplikace Access

To bude sloužit na dotykové obrazovce pro urychlení práce s databází. Jedním kliknutím na toto zaškrtačací pole se nám vypíše čas nachystání a potvrdí se vyhotovení naší práce.

V tento moment můžeme pokračovat ve tvorbě Formuláře. Levým tlačítkem jednou klikneme na tabulku záznamy. Tím si jí označíme, poznáme to tak že její rámeček bude růžový. Poté přejdeme do karty „Vytvoření“ kde vytvoříme rozdělený formulář, stejně jako u minulého

formuláře pro „Objednání“. Rozdělený formulář se nachází v záložce „Další formuláře“. Jedním kliknutím levým tlačítkem jej vytvoříme.

Hned na úvod úprav formuláře si opět odebereme rozložení, které se automaticky vytvořilo na všechny pole ve formuláři. Rozložení označíme kliknutím v levém horním okraji na ikonu pohybu.

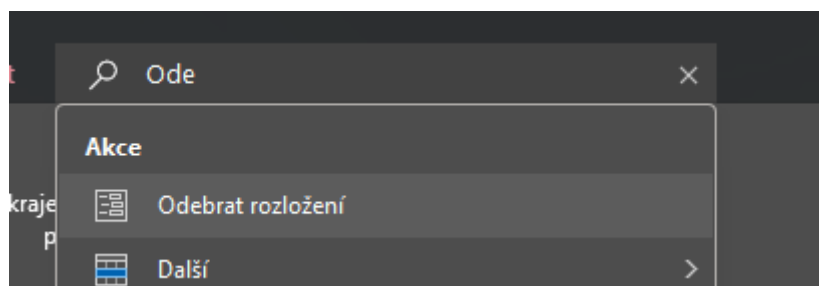


	1	Zákazník
Počet kusů balení	3	Čas objed
ID balení	465930	Čas nachy

Obr. 67: Výběr mřížky ve formuláři

Zdroj: Aplikace Access

Do lupy napíšeme Odebrat rozložení a potvrdíme jej jedním kliknutím levým tlačítkem.



Obr. 68: Odebrání rozložení

Zdroj: Aplikace Access

Po odebrání rozložení si uspořádáme pole ve formuláři. Já budu volit stejnou cestu jako ve formuláři předchozím. Nepodstatné pole pro toto pracoviště si zmenším a přesunu do okrajů. Důležité informace zvětším, zvětším si i písmo a dám jej tučně. Tyto pole přesunu do středu obrazovky, aby byly viditelné jako první. Pole ID zmenším a přesunu do levého horního rohu, písmo zanechám stejné. Čas dovozu a čas nachystání zmenším a přesunu do spodní části formuláře. U časů nechám písmo stejně velké, ale přidám tučné zobrazení. Pole Linka taktéž mírně zmenším a přesunu nad pole s časy. Protože našeho zaměstnance úplně nezajímá, z které linky byla objednávka vyhotovena on ji tam nepoveze, pouze bude chystat balení. ID balení, Zákazník nechám stejně velké přesunu do středu obrazovky a dám vedle sebe. ID balení bude



vlevo a Zákazník hned vedle něj vpravo. Pod ID balení přidám Počet kusů balení se stejnou velikostí. Pod Zákazníka přidám Druh balení taktéž se stejnou velikostí. Potvrzení nachystání a Čas objednání jsou velice důležitá pole proto je dám do úplného středu obrazovky, velikost polí nechám stejnou.

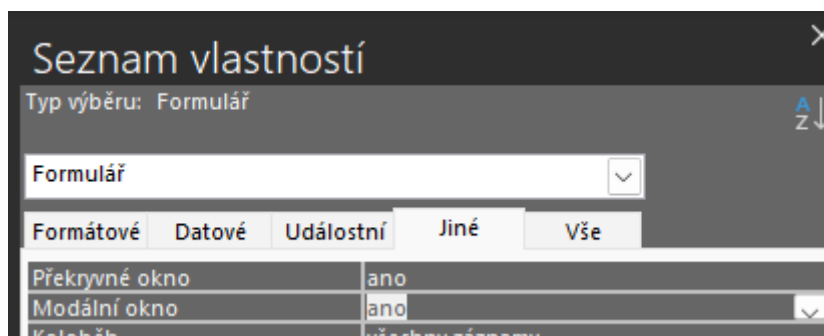
Teď si označím všechny důležité pole a zvětším jejich velikost písma a označím je tučně. Velikost písma nastavím na velikost 14.

Obr. 69: Ukázka formuláře

Zdroj: Aplikace Access

Takto vypadá můj formulář. Teď jej převedeme do full screen zobrazení jako formulář předchozí. Budeme to potřebovat s návazností na masku formuláře. Začneme tím, že formulář budeme mít v návrhovém zobrazení. Přejdeme do seznamu vlastností do karty událostní. V části „Při zavedení“ klikneme na tři tečky a vybereme tvůrce kódu. Postup i program je stejný jako u předchozího formuláře. Do programu tedy opět zadáme DoCmd.Maximize.

Teď přejdeme do karty jiné a povolíme „Překrývané okno“ a „Modální okno“ změnou z „ne“ na „ano“.



Obr. 70: Změna formuláře na překrývané okno

Zdroj: Aplikace Access

Při změně zobrazení formuláře z Návrhového zobrazení na Formulářové si můžeme všimnout námi provedených změn.

Když zaměstnanec bude otevírat tento formulář tak nebude chtít listovat na poslední záznam, bylo by to zdlouhavé. Nový záznam vytvářet také nebude, bude jen potvrzovat splnění své práce proto formulář převedeme při zapnutí na poslední záznam.

Provedeme to stejně jako u formuláře „Nachystání balení“. Přejdeme do návrhového zobrazení, seznamu vlastností karta událostní a zde vybereme tvůrce kódu pro pole „Při zavedení“. Napíšeme stejný program s tím rozdílem, že místo NewRec napíšeme Last, jako poslední.

```
Option Compare Database

Private Sub Form_Load()
DoCmd.Maximize
DoCmd.GoToRecord , , acLast
End Sub
```

Obr. 71: Ukázka kódu "Přechod na poslední záznam v tabulce"

Zdroj: Aplikace Access

DoCmd.GoToRecord , , acLast

V programové části si také vytvoříme program na přidání času po kliknutí na zaškrtačací pole. Budeme chtít, aby se nám ve chvíli potvrzení nachystání objednávky zapsal čas do pole „Čas nachystání“. Takže si klikneme na zaškrtačací pole a přejdeme do seznamu vlastností karta Událostní pole „Při kliknutí“. Pomocí tří teček na konci si otevřeme tvůrce kódu. Ve tvůrci kódu si napíšeme stejný program jako při změně času objednání s tím rozdílem, že budeme měnit hodnotu pole čas nachystání.

Program bude vypadat takto: „Me.Čas\_nachystání.Value = Now()“

```
Private Sub Potvrzení_nachystání_Click()
Me.Čas_nachystání.Value = Now()
End Sub
```

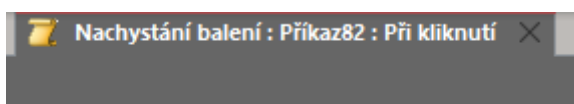
Obr. 72: Tvorba kódu na vyplnění času Now()

Zdroj: Aplikace Access

V tuto chvíli můžeme vytvořit tlačítko na uložení změn ve formuláři. Přejdeme do karty Návrh formuláře, vybereme tlačítko a přidáme si jej do formuláře.

Vybereme „Operace se záznamy“ a „Uložit záznam“. Při uložení by bylo žádoucí, aby se naše databáze přesunula na poslední uložený záznam tudíž poslední objednávku. Z tohoto důvodu si upravíme použité makro v tlačítku „Uložit“.

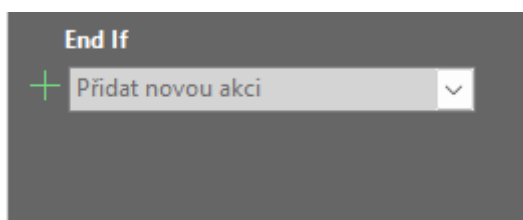
Otevřeme si Seznam vlastností, kartu událostní a klikneme na tři tečky vedle „Vložené makro“. Po kliknutí se nám zobrazí vytvořené makro.



Obr. 73: Ukázka ikony Makra tlačítka

Zdroj: Aplikace Access

Do přidat novou akci zadáme: „Přejít na záznam“.

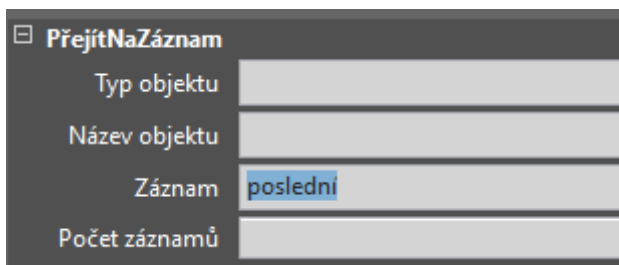


Obr. 74: Přidat novou akci

Zdroj: Aplikace Access

Vytvoří se nám skupina polí, kde můžeme intuitivně pracovat.

My si zvolíme záznam „Poslední“.



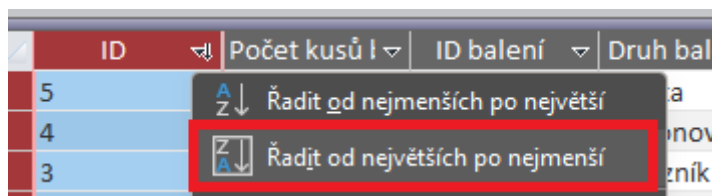
Obr. 75: Změna pole "Záznam"

Zdroj: Aplikace Access

Poté uložíme pomocí ctrl+s nebo tlačítka uložit v horním okraji obrazovky.

V tuto chvíli nám již tlačítko funguje a po kliknutí záznam uloží a přejde na poslední uložený záznam v tabulce.

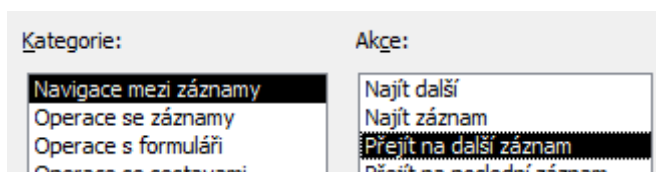
Záznamy v rozděleném formuláři by bylo dobré řadit od posledního. Nebudeme muset stále přejíždět na spodní část.



Obr. 76: Řazení záznamů v rozděleném formuláři

Zdroj: Aplikace Access

Pro snadnější navigaci mezi záznamy přidáme tlačítka „Předchozí záznam“ a „Další záznam“. Tyto tlačítka se vytvářejí stejně jako tlačítka jiná, přidáme tlačítko a vybereme si je z nabídky. Tentokrát z Kategorie: Navigace mezi záznamy Akce: Přejít na další záznam.



Obr. 77: Tvorba tlačítka Přejít na záznam

Zdroj: Aplikace Access

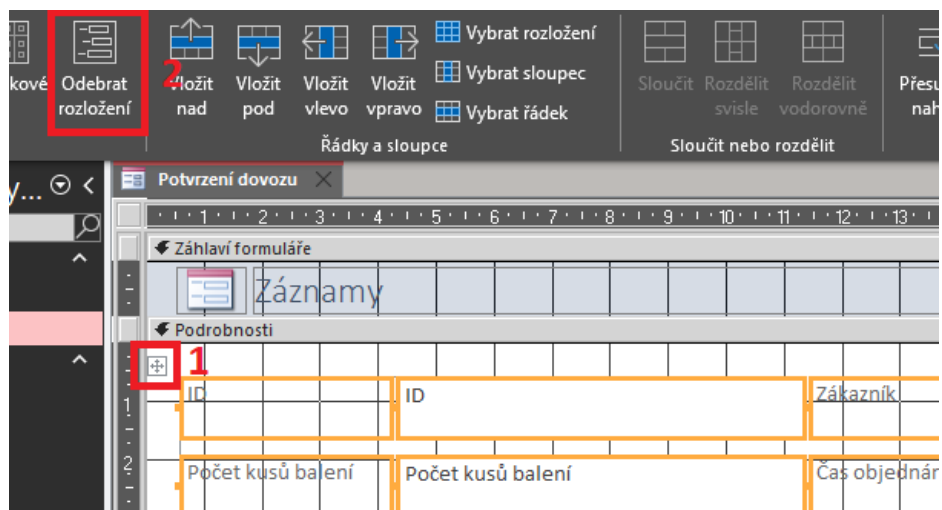
U těchto tlačítek si ponechám ikony šipek, přijde mi to pohledné a intuitivní.

Tlačítko s přechodem na předchozí záznam vytvoříme stejně jen změníme Akci na: Přejít na předchozí záznam. Po dokončení této operace se můžeme vrhnout na vytvoření posledního formuláře.

Vytvoříme formulář „Potvrzení dovozu“ a to stejným způsobem jako předešlé dva formuláře. Zde ponecháme všechny údaje o objednávce.

Zvětšíme pouze údaj o potvrzení nachystání a Čas nachystání. Poté přidáme do tabulky sloupec se zaškrťovacím polem „potvrzení dovozu“. Abychom mohli po zaškrtnutí zapsat do databáze potvrzení a na toto potvrzení navázat i „Čas dovozu“.

Přepneme si formulář do návrhového zobrazení a odebereme rozložení abychom mohli uspořádat libovolně pole ve formuláři. Označíme si opět pomocí čtverce s šipkami všechny pole a přesuneme se do karty „Uspořádání“.

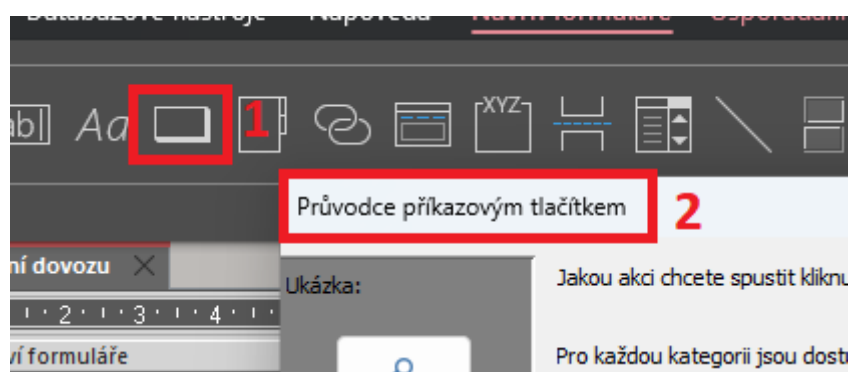


Obr. 78: Odebrání rozložení polí ve formuláři

Zdroj: Aplikace Access

Nepodstatné informace zmenšíme a přesuneme do horní části formuláře. Podstatné pro nás budou pole Linka, Čas nachystání, Potvrzení nachystání, Počet kusů balení, ID balení, Druh balení a Potvrzení dovozu.

Po rozvržení formuláře si přidáme tlačítko uložení, stejným způsobem jako u předchozích formulářů.



Obr. 79: Tvorba tlačítka s předinstalovaným kódem

Zdroj: Aplikace Access

V tuto chvíli můžeme v Seznamu vlastností v hlavičce Formulář a kartě událostí při zavedení přidat kód, aby se formulář otevíral na posledním dostupném záznamu. Průběh bude stejný jako u předchozího formuláře.

```
Option Compare Database

Private Sub Form_Load()
DoCmd.Maximize
DoCmd.GoToRecord , , acLast
End Sub
```

Obr. 80: Přejít na poslední záznam v tabulce při startu formuláře

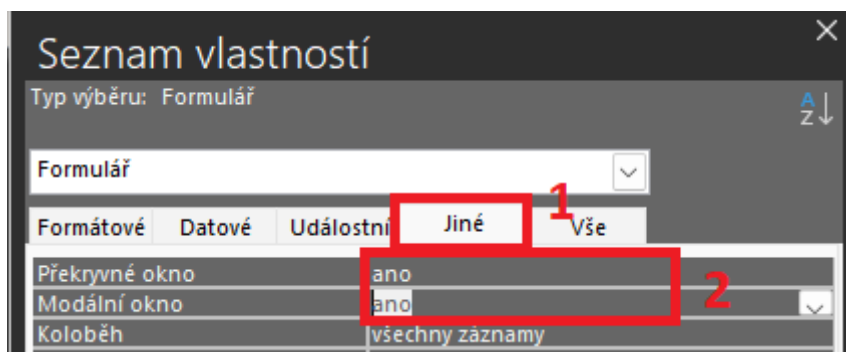
Zdroj: Aplikace Access

DoCmd.GoToRecord , , acLast

Když už máme programovací prostředí otevřené můžeme rovnou vložit větu pro maximalizaci formuláře.

DoCmd.Maximize

V kartě „Jiné“ změním „Překrývané okno“ a „Modální okno“ na ano.



Obr. 81: Překrývané okno

Zdroj: Aplikace Access

Přidáme do formuláře podmíněné formátování, aby zaměstnanec z logistiky byl vizuálně upozorněn na nachystanou objednávku. To samé přidáme i na objednávku, která již byla doručena. Zaměstnanec bude mít větší přehled v tom, které zakázky již dokončil. Opět budeme pracovat s příkazem IsNull.

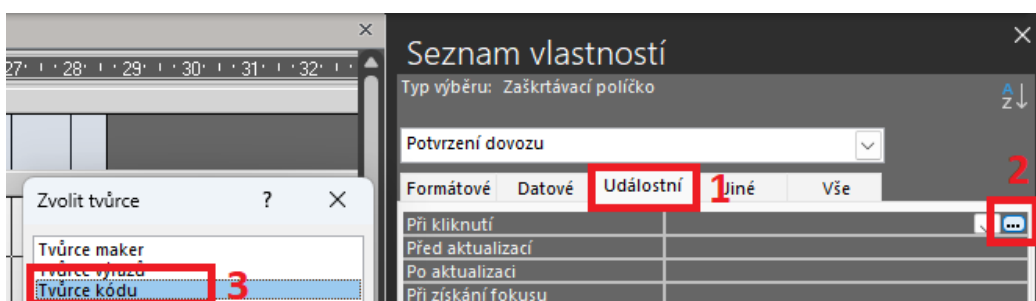
Pravým tlačítkem klikneme na sloupec „Čas nachystání“ a přejdeme do „Podmíněné formátování“. Zde vytvoříme nové pravidlo „výraz je“ a přidáme kód: `IsNull([Čas nachystání]) = False`

U Času dovozu provedeme totéž jen změníme v kódu pole, které budeme upravovat: `IsNull([Čas dovozu]) = False`

Barvu opět zvolíme zelenou, aby přitáhla pozornost.

Ještě je zapotřebí naprogramovat vypsání času po zakliknutí pole „Potvrzení dovozu“.

V návrhovém zobrazení si jedním kliknutím levým tlačítkem myši označíme pole „Potvrzení dovozu“ a přejdeme do Seznamu vlastností. V seznamu vlastností si otevřeme kartu událostní a klikneme na tři tečky v sekci „Při kliknutí“. Otevřeme „Tvůrce kódu“.



Obr. 82: Zadání kódu do prázdného tlačítka

Zdroj: Aplikace Access

```
Private Sub Potvrzení_dovozu_Click()  
Me.Čas_dovozu.Value = Now()  
End Sub
```

Obr. 83: Ukázka kódu pro čas Now()

Zdroj: Aplikace Access

Kód bude následující: `Me.Čas_dovozu.Value = Now()`

V tuto chvíli se nám čas automaticky vypíše po zakliknutí zaškrtnutého pole.

Jako poslední věc upravíme velikost a tučnost písma v polích.

Tučně nastavíme všechny pole a jen pole, které jsou důležité budou mít nastavenou větší velikost písma.

Pro lepší přehled v databázi přidáme do všech formulářů až na objednání automatickou aktualizaci dat. A to z toho důvodu, že máme formuláře převedeny do maximálního zobrazení a není možnost se k ruční aktualizaci dostat, tak i zamezíme tomu, že by si některý ze zaměstnanců formulář neaktualizoval. V každém z formulářů bude postup stejný, proto ho popíšu jen jednou a v databázi ho provedu dvakrát.

Přepneme si formulář do návrhového zobrazení. Přejdeme do seznamu vlastností, karta „Událostní“ pole „Při události časovače“. Zde přejdeme do tvůrce kódu, kam si zapíšeme: „DoCmd.Requery“ vše uložíme. Časovač nastavíme na 10 000 což je 10 sekund.

Při události časovače	[Obslužná procedura událostí] 1
Interval časovače	10000 2

Obr. 84: Tvorba časovače kódu

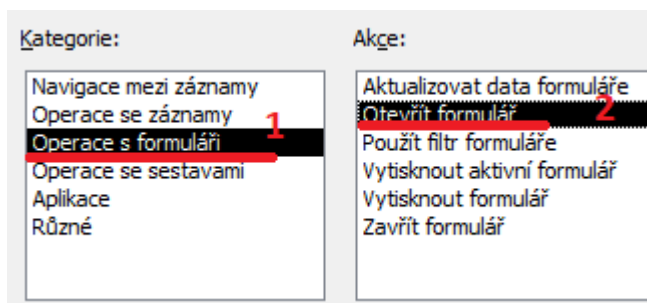
Zdroj: Aplikace Access

Tuto operaci provedeme ve všech vytvořených formulářích až na objednání.

Dále budeme pokračovat „Maskou“ databáze. Ta nám dodá větší přehled po zapnutí a také nějaký ten design. Takže vytvoříme nový formulář. Tentokrát „Prázdný formulář“, najdeme jej v kartě vytvoření. Po kliknutí na „Prázdný formulář“ se nám automaticky vytvoří, rovnou můžeme uložit a změnit název na „Maska“.

Přejdeme do návrhového zobrazení a budeme vytvářet čtyři tlačítka. První tři budou otevírat formuláře a poslední bude sloužit k vypnutí celé Access databáze. Vše se budeme snažit orientovat do středu obrazovky.

Přidáme tlačítko a budeme pokračovat výběrem „Operace s formuláři“ a „Otevřít formulář“.

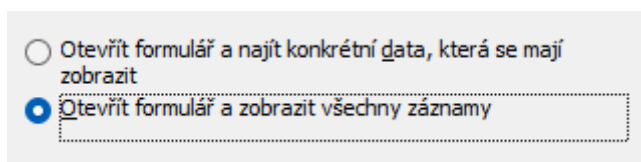


Obr. 85: Tvorba navigačního tlačítka na otevření formuláře

Zdroj: Aplikace Access



Výběrem tlačítka další se nám zobrazí seznam našich formulářů, my si vybereme jako první „Objednání“. Na další stránce ponecháme „Otevřít formulář a zobrazit všechny záznamy“.

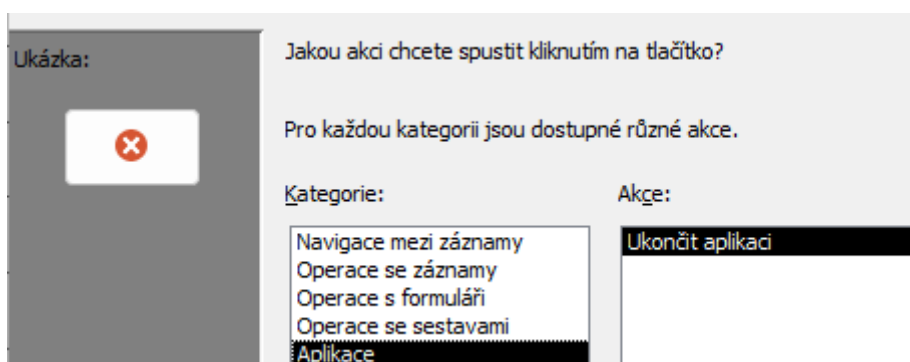


Obr. 86: Otevřít formulář a zobrazit všechny záznamy

Zdroj: Aplikace Access

V dalším kroku máme na výběr, jestli chceme obrázek či text, určitě zvolíme text, protože kdybychom měli na výběr ze třech stejných obrázků tak bychom nevěděli, který formulář se kde nachází. Do textu napíšeme název našeho formuláře.

Stejným způsobem vytvoříme i tlačítka pro otevření ostatních formulářů. V pořadí „Nachystání balení“ a poté „Potvrzení dovozu“. Jako poslední tlačítko si vytvoříme tlačítko na vypnutí celé aplikace. Nachází se v kategorii „Aplikace“ pod názvem „Ukončit aplikaci“.



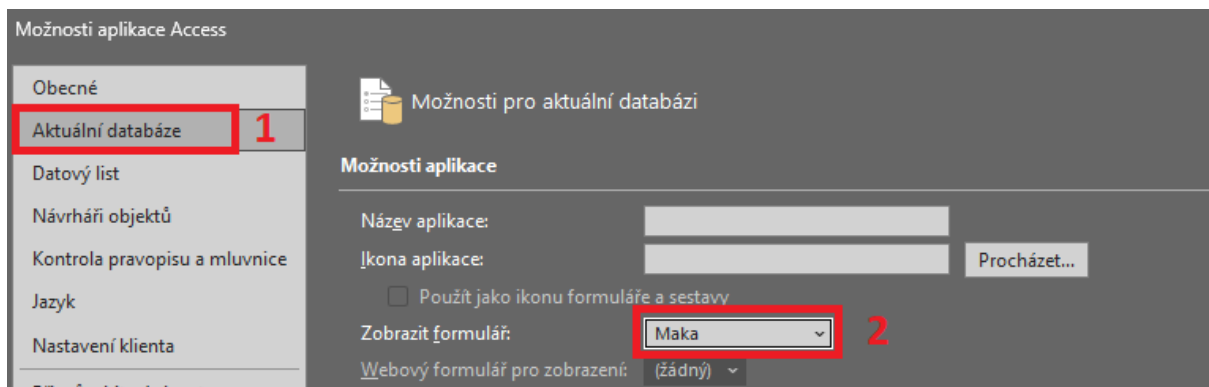
Obr. 87: Tlačítko na ukončení aplikace

Zdroj: Aplikace Access

Opět vybereme text a to: „Ukončit aplikaci“. Je to přehlednější. Jako poslední věc nastavíme velikosti tlačítek na stejné a pozadí tlačítek nastavím na světle šedou barvu. Pro jednodušší nastavení velikosti se hodí označit si všechny tlačítka a na jednom měnit velikost. Poté přejít do „Uspořádání“, „Velikost a mezery“ a nastavit si velikost „Podle nejvyššího“ a „Nejširšího“. To nám sjednotí velikost tlačítek. V tuto chvíli můžeme přidat rozložení, které nám vyrovná tlačítka pod sebe a uspořádá je do mřížky. V kartě „Uspořádání“ v levé části vybereme rozložení „Skládané“. Tlačítka se nám sice uspořádají, ale jsou u sebe velice blízko,

proto přejdeme do rozbalovacího tlačítka „Výplň ovládacího prvku“. Zde si výplň nastavíme na střední, ta mi přijde nejvíce ideální. Díky mezerám také zabráníme nechtěným překliknutím. Již nám zbývá pouze formulář převést do maximálního zobrazení v nastavení formuláře při události „Při zavedení“ a použitím kódu DoCmd.Maximize. Nesmíme zapomenout povolit modální a překrývané okno.

V tuto chvíli máme masku hotovou a můžeme ji nastavit jako výchozí zobrazení databáze. To znamená, že v momentě, kdy otevřeme aplikaci Access tak se nám zobrazí jako první tento formulář. Je to výhodné v tom, že zaměstnanec nemusí pracovat s databází, ale pouze intuitivně klikne na tlačítko. Také to urychlí a usnadní proces učení s databází. Nastavení primárního zobrazení formuláře na jdeme kartě „Soubor“, v levé spodní části klikneme na tlačítko „Možnosti“. V možnostech aplikace Access přejdeme do záložky „Aktuální databáze“ a v poli „Zobrazit formulář“ vybereme formulář „Maska“.



Obr. 88: Při zapnutí databáze zobrazit formulář

Zdroj: Aplikace Access

Potvrdíme stisknutím tlačítka OK v pravé dolní části okna.

Změna se projeví po vypnutí a novém zapnutí databáze. V tuto chvíli máme vše hotovo a můžeme naši databázi začít testovat v provozu naší linky. Doporučuji zřídit bezdrátové připojení pro tablety, které bude mít řidič vysokozdvížného vozíku. Pro efektivní práci s výslednými daty se dá aplikace Access propojit a aplikací PowerBi a v reálném čase si sdílet navzájem data. To nám umožní jednou vytvořit libovolné grafy, které se budou nadále automaticky aktualizovat.

Odkaz na stažení hotové databáze:

[https://drive.google.com/file/d/1IHQKOIN\\_fNgFnmAEVV0mdT8fPY17E\\_hZ/view?usp=sharing](https://drive.google.com/file/d/1IHQKOIN_fNgFnmAEVV0mdT8fPY17E_hZ/view?usp=sharing)

## ZÁVĚR

V teoretické části jsme se dozvěděli potřebné informace, s kterými jsme následně pracovali v praktické části. Jde o základní pojem o databázích jako celku a následně se dozvíme o různých modelech databází a softwarů pro jejich správu. Poté jsme v praktické části vytvořili databázi jako demonstraci aplikace Access. Praktická část se zabývá tvorbou databáze pro objednání balícího materiálu. Zaměstnanec místo různých složitých úkonů pouze párkrát klikne v databázi. Tato databáze se dá také využít pro shromažďování dat kolem výrobního procesu a jejich následnou analýzu. Databázi jsme transformovali do jednoduché intuitivní aplikace s přehlednými tlačítky a vlastním menu. Takže není zapotřebí vypisovat různé množství tabulek ručním způsobem. Závěrem bych chtěl shrnout program Access. Myslím si, že tento program je velice opomíjený. Je intuitivní a nabízí velké množství funkcí, které nenabízí ani jiné a dražší databázové modely. Pro správu středního objemu dat je to za mě nejlepší volba. Access se dokáže naučit spravovat i méně informaticky založený člověk. Access můžou využívat třeba i profese kde by byla klasická databáze například ORACLE nedostupná z finančního hlediska. V podstatě k obyčejnému kancelářskému balíčku nám byl poskytnut velice účinný a intuitivní nástroj. Dokážu si představit, že více takových vylepšení, které jsem navrhl by si různé firmy dokázaly vypracovat samy bez velkého utrácení peněz za externí firmy, které by jim vytvořily podobnou databázi. Další výhodou vnímám formuláře, kde si můžeme doprogramovat velké množství věcí, které nám usnadní práci s databází. Také ve světě je tento program vnímán úplně jinak jak u nás v České republice, více se s ním pracuje a vytváří se v něm spousta zlepšujících návrhů jako doplněk k Excelu. Díky tomu je na internetu spousta návodů, rad a ukázek, jak si vytvořit databázi. Vše se dá najít krok po kroku a naučit se. K tomu i slouží podpora od firmy Microsoft, která nám nabízí spousta návodů na MS Access online. Já se s tímto programem naučil pracovat sám s využitím internetových návodů, z počátku mi trvalo, než jsem přišel na některé věci, ale poté jak jsem pochopil základy, tak jsem se vždy doučil tu danou část, kterou jsem potřeboval. Pracoval jsem i s jinými databázovými programy a Access musím opravdu vyzdvihnout jako velice intuitivní nástroj, v případě že si chceme pouze vytvořit základní databázi, postačí nám znalosti kancelářských balíčků. Toto téma jsem si vybral sám a jsem rád za tuto zkušenost s bakalářskou prací a vnímám to velice poučně. Také v praktické části jsem demonstroval Access přesně tak jak jsem chtěl. Chtěl jsem právě ukázat tu jeho jednoduchost a sílu, nechtěl jsem dělat nějaké složité úkony a vykreslit ho v jiném světle, než ho vidím já.

## Seznam použitých zdrojů

### Literatura

BULISOVÁ, Jiřina. *Ottova všeobecná encyklopedie*. Praha: OTTOVO NAKLADATELSTVÍ, 2010. ISBN:978-80-7360-902-3.

prof.RNDr. Jaroslav POKORNÝ, CSc a Ing.Michal, Ph.D. VALENTA. *Databázové Systémy*. Praha: Česká technika – nakladatelství ČVUT, 2020. ISBN: 978-80-01-06696-6.

### Elektronické zdroje

*Co je nového v Accessu pro Microsoft 365* [online]. Microsoft, 2023 [cit. 2023-03-22].

Dostupné z: <https://support.microsoft.com/cs-cz/office/co-je-nov%C3%A9ho-v-accessu-pro-microsoft-365-76454345-f85d-47af-ace1-98a456cb3496?ui=cs-cz&rs=cs-cz&ad=cz>

Co je to hierarchická databáze?. Theastrologypage [online]. 2023 [cit. 2023-03-27]. Dostupné z: <https://cs.theastrologypage.com/hierarchical-database>

*Database* [online]. Oracle, 2023 [cit. 2023-03-22]. Dostupné z:<https://www.oracle.com/cz/database/>

*Elasticsearch* [online]. Elastic NV, 2023 [cit. 2023-03-22]. Dostupné z:<https://www.elastic.co/elasticsearch/>

HRANICKÝ, Jan. *Oracle krok za krokem: Vytvoření schématu a tabulky* [online]. In: . itnetwork, 2021 [cit. 2023-03-22]. Dostupné z: <https://www.itnetwork.cz/oracle/oracle-krok-za-krokem-vytvoreni-schematu-a-tabulky>

KOTÁSEK, Ing. Jaroslav. *Databáze: Pojem a historie* [online]. 2012 [cit. 2023-03-22]. Dostupné z: [https://www.sokolska.cz/DUMy/VYT/VY\\_32\\_INOVACE\\_31-01.pdf](https://www.sokolska.cz/DUMy/VYT/VY_32_INOVACE_31-01.pdf)

*Maria DB 10.1* [online]. Websupport, 2021 [cit. 2023-03-22]. Dostupné z:<https://www.websupport.cz/podpora/kb/ maria-db-10-1/>

*MariaDB Server* [online]. MariaDB Foundation, 2023 [cit. 2023-03-22]. Dostupné z: <https://mariadb.org/>

MARVAN, Luboš. *Úvod a prostředí programu MS Access* [online]. In: . itnetwork, 2021 [cit. 2023-03-22]. Dostupné z: [https://www.itnetwork.cz/ms-office/microsoft-access-zaklady/uvod-a-prostredi-programu-ms-access?gclid=CjwKCAiA9NGfBhBvEiwAq5vSy4preBbFnwWrJaIVugoDH-9S5S7rw7iHeTeJZn2T3wISS5oDI72sRoCOwcQAvD\\_BwE](https://www.itnetwork.cz/ms-office/microsoft-access-zaklady/uvod-a-prostredi-programu-ms-access?gclid=CjwKCAiA9NGfBhBvEiwAq5vSy4preBbFnwWrJaIVugoDH-9S5S7rw7iHeTeJZn2T3wISS5oDI72sRoCOwcQAvD_BwE)

*Microsoft Access* [online]. Microsoft, 2023 [cit. 2023-03-22]. Dostupné z: <https://www.microsoft.com/cs-cz/microsoft-365/access>

*MongoDB* [online]. MongoDB, 2023 [cit. 2023-03-22]. Dostupné z: <https://www.mongodb.com/document-databases>

MŮČKA, Jan. *Relační databáze vs. nerelační databáze: jaké jsou mezi nimi rozdíly?* [online]. 2021 [cit. 2023-03-22]. Dostupné z: <https://www.master.cz/blog/relacni-databaze-nerelacni-databaze-jake-jsou-rozdily/>

*MySQL* [online]. MySQL AB, 2022 [cit. 2023-03-22]. Dostupné z: <https://www.mysql.com/>

*MySQL vs MariaDB* [online]. MariaDB, 2023 [cit. 2023-03-22]. Dostupné z: <https://mariadb.com/kb/en/mariadb-vs-mysql-features/>

PLACHÝ, Patrik. *Trendy moderních databází* [online]. Systém OnLine, 2014 [cit. 2023-03-22]. Dostupné z: <https://www.systemonline.cz/business-intelligence/trendy-modernich-databazi.htm>

POKORNÁ, Dagmar. *Architektura databázových systémů*. In: Slideplayer.cz [online]. [cit. 2023-03-22]. Dostupné z: <https://slideplayer.cz/slide/2444856/>

*Průvodce výběrem Microsoft SQL Server* [online]. Českomoravské informační systémy, 2023 [cit. 2023-03-22]. Dostupné z: <https://www.mssql.cz/>

ŠKODA, Jan. *Databázové systémy*. Inovace VOV [online]. ČVUT, 2019 [cit. 2023-03-22]. Dostupné z: <https://www.vovcr.cz/odz/tech/393/page00.html>

ŠKODA, Jan. *Potřeba a význam databázových systémů*. Inovace VOV [online]. ČVUT, 2019 [cit. 2023-03-22]. Dostupné z: <https://www.vovcr.cz/odz/tech/393/page01.html>

TÁBOR, Pavel. *Kapitola 13: Transakce. Koncept transakce. ACID vlastnosti*. DocPlayer [online]. 2017 [cit. 2023-03-22]. Dostupné z: <https://docplayer.cz/45037336-Kapitola-13-transakce-koncept-transakce-acid-vlastnosti.html>

*What is a Document Database?* [online]. MongoDB, 2023 [cit. 2023-03-22]. Dostupné z: <https://www.mongodb.com/document-databases>

*What Is Amazon Neptune?* [online]. Amazon.com, 2023 [cit. 2023-03-22]. Dostupné z: <https://docs.aws.amazon.com/neptune/latest/userguide/intro.html>

*What is Apache Cassandra?* [online]. Apache Software Foundation, 2023 [cit. 2023-03-22]. Dostupné z: [https://cassandra.apache.org/\\_/cassandra-basics.html](https://cassandra.apache.org/_/cassandra-basics.html)

WHITE, Kevin. *Characteristics of a Modern Database* [online]. In: . 2015 [cit. 2023-03-24]. Dostupné z: <https://www.singlestore.com/blog/modern-database-characteristics/>

*Why MySQL?* [online]. MySQL AB, 2022 [cit. 2023-03-22]. Dostupné z: <https://www.mysql.com/why-mysql/>

ŽŮREK, Michal. *MS-SQL krok za krokem: Úvod do MS-SQL a příprava prostředí* [online]. In: . itnetwork, 2021 [cit. 2023-03-22]. Dostupné z: <https://www.itnetwork.cz/ms-sql/mssql-tutorial-uvod-a-priprava-prostredi>

## Seznam obrázků

Obr. 1: Děrný štítek z amerického sčítání lidu v roce 1890	7
Obr. 2: Databáze zaměstnanců [13.02.2022]	8
Obr. 3: Jednovrstvá architektura	9
Obr. 4:Dvouvrstvá architektura File-Server [01.03.2023]	10
Obr. 5:Dvouvrstvá architektura Klient-Server [01.03.2023]	10
Obr. 6: Vícevrstvá architektura [01.03.2023]	11
Obr. 7: Relace [01.03.2023]	14
Obr. 8: Ukázka výrobního procesu	22
Obr. 9: Ikona aplikace Access	22
Obr. 10: Ikona prázdná databáze	23
Obr. 11: Pojmenování databáze	23
Obr. 12: Prázdná tabulka	24
Obr. 13: Návrhové zobrazení	24
Obr. 14: Uložení tabulky	25
Obr. 15: Datový typ primárního klíče	25
Obr. 16:Změna datového typu ID	25
Obr. 17: Tabulka	26
Obr. 18: Sloupec Zákazník	26
Obr. 19: Datový typ průvodce vyhledáváním Zdroj: Aplikace Access	26
Obr. 20: Výběr hodnot	27
Obr. 21: Návrh hodnot v poli	27
Obr. 22:Ukázka upravené tabulky	28
Obr. 23:Ukázka vložení dat Zdroj: Aplikace Access	28
Obr. 24: Ukázka tvorby tabulky	29
Obr. 25:Ukázka vytvořené tabulky "Záznamy"	30
Obr. 26:Průvodce vyhledáváním	30
Obr. 27:Výběr tabulky	30
Obr. 28: Výběr polí	31
Obr. 29: Ponechání klíčového sloupce	31
Obr. 30: Uložení tabulky	31
Obr. 31: Ukázka relace	32
Obr. 32: Ukázka tabulky "Záznamy"	32
Obr. 33: Návod na založení Rozděleného formuláře	33
Obr. 34: Vytvořený formulář	34
Obr. 35: Ukázka přepnutí do návrhového zobrazení	34
Obr. 36: Označení mřížky	34
Obr. 37: Odebrání rozložení	35
Obr. 38: Tučné písmo	35
Obr. 39: Výběr seznamu vlastností	36
Obr. 40:Karta "Událostní"	36
Obr. 41: Výběr tvůrce kódu	36
Obr. 42: Programové prostředí jazyka VBA	37
Obr. 43: Výběr pole ve formuláři	37
Obr. 44: První ukázka programu ve formuláři	37
Obr. 45: Ukázka vypsaného formátu času	38

Obr. 46: Tvorba tlačítka	39
Obr. 47: Výběr tlačítka bez programu	40
Obr. 48: Pojmenování tlačítka	40
Obr. 49: Výběr tvurce kódu	41
Obr. 50: Ukázka hotového kódu	41
Obr. 51: Výsledná podoba tlačítek	42
Obr. 52: Označení sloupce v rozděleném formuláři	42
Obr. 53: Výběr Podmíněného formátování	43
Obr. 54: Tvorba nového pravidla	43
Obr. 55: Formát pravidla buňky	43
Obr. 56: Výsledek Podmíněného formátování	44
Obr. 57: Výběr pole "Při zavedení"	44
Obr. 58: Ukázka maximalizačního kódu	45
Obr. 59: Změna formuláře na okno	45
Obr. 60: Programování při zavedení	46
Obr. 61: Ukázka kódu "Přejít na nový záznam"	46
Obr. 62: Změna na "Návrhové zobrazení"	47
Obr. 63: Změna na zaškrťovací tlačítko	47
Obr. 64: Ukázka sloupce v tabulce se zaškrťovacím polem	47
Obr. 65: Výběr mřížky ve formuláři	48
Obr. 66: Odebrání rozložení	48
Obr. 67: Ukázka formuláře	49
Obr. 68: Změna formuláře na překrývané okno	49
Obr. 69: Ukázka kódu "Přechod na poslední záznam v tabulce"	50
Obr. 70: Tvorba kódu na vyplnění času Now()	50
Obr. 71: Ukázka ikony Makra tlačítka	51
Obr. 72: Přidat novou akci	51
Obr. 73: Změna pole "Záznam"	51
Obr. 74: Řazení záznamů v rozděleném formuláři	52
Obr. 75: Tvorba tlačítka Přejít na záznam	52
Obr. 76: Odebrání rozložení polí ve formuláři	53
Obr. 77: Tvorba tlačítka s předinstalovaným kódem	53
Obr. 78: Přechod na poslední záznam v tabulce při startu formuláře	54
Obr. 79: Překrývané okno	54
Obr. 80: Zadání kódu do prázdného tlačítka	55
Obr. 81: Ukázka kódu pro čas Now()	55
Obr. 82: Tvorba časovače kódu	56
Obr. 83: Tvorba navigačního tlačítka na otevření formuláře	56
Obr. 84: Otevřít formulář a zobrazit všechny záznamy	57
Obr. 85: Tlačítko na ukončení aplikace	57
Obr. 86: Při zapnutí databáze zobrazit formulář	58



## Seznam tabulek

Tab. 1: Ukázka tabulky relační databáze .....	13
---	----

## Anotace

<b>Jméno a příjmení:</b>	Parák Miroslav
<b>Katedra:</b>	Katedra technické a informační výchovy
<b>Vedoucí práce:</b>	Mgr. Tomáš Dragon
<b>Rok obhajoby:</b>	2023

<b>Název práce:</b>	Návrh relační databáze pro evidenci zaměstnaneckých dat
<b>Název v angličtině:</b>	Design of a relational database for recording employee data
<b>Anotace práce:</b>	<p>Práce je zaměřena na zjednodušení pracovních úkonů zaměstnanců ve firemní sféře. Hlavním cílem práce je vytvořit databázi v programu Access. Demonstruji většinu základních i programovatelných funkcí aplikace.</p>
<b>Klíčová slova:</b>	Access, Databáze, Relační, SQL, Formulář, VisualBasic, Relace
<b>Anotace v angličtině:</b>	<p>The thesis focuses on simplyfying work tasks of employees in the corporate sector. The main objective of the thesis is to create a database in the Access program. I demonstrate most of the basic and programmable functions of the application.</p>
<b>Klíčová slova v angličtině:</b>	Access, Database, Relational, SQL, Form, Visual Basic, Relation
<b>Přílohy vázané v práci:</b>	
<b>Rozsah práce:</b>	Stran 66
<b>Jazyk práce:</b>	čeština

