Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství

Bakalářská práce

Databázově koncipované informační zabezpečení provozování e-shopu

Monika Debreczényiová Krammerová

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Monika Debreczényiová Krammerová

Informatika

Název práce

Databázově koncipované informační zabezpečení provozování E-shopu

Název anglicky

Database conceived information assurance of E-shop operation

Cíle práce

Bakalářská práce je zaměřena na problematiku využití relačně databázové technologie v informačním zabezpečení provozu E-shopů. Hlavním cílem této práce je:
a) objasnit teoretické principy relačně databázové technologie se zřetelem na problematiku informačního zabezpečení provozování E-shopů na platformě DBMS ORACLE,
b) zmapovat momentální stav této problematiky a vymezit její relevantnost včetně požadavků na ni kladených,
c) navrhnout přijatelné řešení této problematiky v souladu s identifikovanými požadavky,
d) ověřit funkčnost navrženého řešení v rámci funkčního prototypu,
e) ověřené záležitosti zobecnit pro další možná uplatnění.

Metodika

Použitá metodika zadané bakalářské práce bude založena na studiu a analýze dostupných informačních zdrojů a existujících řešení v dané oblasti. Stěžejními metodami této práce budou metody a techniky relačně databázové technologie. Navrhované řešení bude zohledňovat identifikované požadavky a očekávání spojená s řešenou záležitostí. Na podkladě syntézy teoretických poznatků a dosažených výsledků budou formulovány závěry této bakalářské práce a následně zobecněny pro další možná použití.

Závazný harmonogram:

Teoretické principy řešené problematiky, literární rešerše – předmět 1. zápočtu z BP: do 5.9.2016

Zmapování momentální situace řešené problematiky, identifikace požadavků s tím spojených: do 15.11.2016

Navržení možného řešení a jeho následné ověření – předmět 2. zápočtu z BP: do 28.1.2017

Zobecnění navržených záležitostí pro další možná použití – předmět 3. zápočtu z BP: do 15.3.2017

Doporučený rozsah práce

45-55 stran

Klíčová slova

Relačně databázová technologie, informační zabezpečení, DBMS ORACLE, SQL, datová integrita, normalizace dat, datové modelování

Doporučené zdroje informací

GROFF, James R. a Paul N. WEINBERG. SQL: kompletní průvodce. 2005. Brno: CP Books, 2005. Programování. ISBN 8025103692.

Itnetwork . MS-SQL databáze krok za krokem. [online]. 8.6.2016 [cit. 2016-05-04]. Dostupné z: http://www.itnetwork.cz/ms-sql

LACKO, Ľuboslav. Databáze: datové sklady, OLAP a dolování dat s příklady v Microsoft SQL Serveru a Oracle. Brno: Computer Press, 2003. ISBN 8072269690.

MORKES, David. Microsoft SQL Server 2000: tvorba, úprava a správa databází. Praha: Grada, 2004. Podrobný průvodce začínajícího uživatele. ISBN 8024707322.

VALENTA, M., POKORNÝ, J. Databázové systémy. Praha: České vysoké učení technické v Praze, 2013. ISBN 978-80-01-05212-9.

Předběžný termín obhajoby

2016/17 LS – PEF

Vedoucí práce

doc. Dr. Ing. Václav Vostrovský

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 1. 11. 2016

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 1. 11. 2016

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 03. 03. 2017

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Databázově koncipované informační zabezpečení provozování e-shopu" jsem vypracovala samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autorka uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 12.3.2017 _____

Poděkování

Ráda bych touto cestou poděkovala Dr. Ing. Václavu Vostrovskému, Ph.D., za odborné vedení bakalářské práce a za cenné rady týkající se jejího zpracování.

Databázově koncipované informační zabezpečení provozování e-shopu.

Souhrn

Hlavním cílem této bakalářské práce je vytvořit databázový návrh pro aplikaci, která se zabývá provozováním e-shopu. Kromě objasnění teoretických principů relačně databázové technologie je významnou částí definice požadavků na datový model, ovlivněna především současnými trendy v oblasti datové evidence e-shopu. Stěžejní částí je ovšem samotný návrh konceptuálního datového modelu odpovídající požadavkům stanovených v předchozí části.

Klíčová slova: Relačně databázová technologie, informační zabezpečení, DBMS ORACLE, SQL, datová integrita, normalizace dat, datové modelování

Database conceived information assurance of e-shop operation

Summary

The main goal of this Bachelor thesis is to create a database design for an application that deals with the operation of a shop. In addition to the clarification of the theoretical principles of relational database technology is an important part of the definition of the requirements for the data model, influenced by current trends in the field of the data record shop. The core parts of the proposal itself, however, is a conceptual data model corresponding to the requirements set out in the previous section.

Keywords: Relational database technology, information security, ORACLE DBMS, SQL, data integrity, data normalization, data modeling

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

Tématem bakalářské práce je návrh řešení relační databáze a zabezpečení databázové evidence potřebné pro úspěšné fungování e-shopu. Téma jsem si zvolila z důvodu osobního zájmu o tuto oblast, ale také kvůli možnosti využití teoretických poznatků z České zemědělské univerzity v rámci předmětu Databáze.

Pojem databáze dnes asi není zcela jistě nikomu cizí. Lidé mají potřebu evidovat a shromažďovat informace už odpradávna. Celá dnešní společnost je postavena na databázových systémech, od evidence občanů, automobilů, školství, hospodářství, až po síť mobilních telefonů. I proto se v dnešní době téměř žádná organizace neobejde bez použití informačních technologií. Pokud organizace zpracovává a uchovává data v elektronické podobě, nejčastěji si jako vlastní úložiště dat volí databázový systém. Podle svých potřeb a možností si především vybírá z komerčních databázových produktů firem Oracle, a Microsoft, IBM a dalších. Za těmito databázovými produkty stojí silná společnost, a proto organizace využívají pro evidenci svého podnikání databázovou evidenci, jež slouží pro uživatele jako přehled všech údajů, se kterými můžou dále pracovat nebo zakládat nové údaje, upravovat stávající a vytvářet potřebné výstupy pro další zpracování.

V relačních databázích jsou údaje uloženy v tabulkách, které spolu nějak souvisí, a tak je možno dosáhnout vysoké míry komplexity a zároveň přehlednosti databáze.
Vytvořenou databázi firmy využívají k tvorbě e-shopů, protože se předhánějí v získávání zákazníků. V této době je základem mít dobrou dostupnost zboží pro zákazníky, protože ne každý zákazník má čas nakupovat zboží v obchodě. Z tohoto důvodu vznikají e-shopy. Pokud společnost nemá na tvorbu e-shopu vlastní zaměstnance, je nutné při samotné tvorbě e-shopu vybrat firmu, která zpracuje návrh. Ten musí obsahovat všechny požadavky podle cíle, kterého chce společnost pomocí e-shopu dosáhnout. Předpokladem je, že na prvním místě bude skvělá dostupnost zboží, pak přehlednost zboží a samozřejmostí v dnešní době je i důraz na design. Nejdůležitější informací jsou jasně popsané postupy prací ve společnosti a po vymezení pracovní části je potřebné zpracovat všechny požadavky do přehledného diagramu.

Důležitou částí této práce je zabezpečení databáze jak ze strany zákazníka, tak i ze strany zaměstnanců firmy. Podniky své databáze používají při programování webových aplikací, proto je nedílnou součástí před spuštěním e-shopu do provozu tuto databázi zabezpečit před možným napadnutím. Převážná většina kvalitních internetových serverů pracuje nějakým způsobem s databází.

Ochrana dat je pro každou organizaci klíčovou tématikou. Data jsou jedním z nejcennějších aktiv a zároveň primárním cílem útočníka. Bezpečnost informačního systému nekončí nastavením firewallu, je nutné minimalizovat i možnosti přístupu k datům. Zabezpečení databáze je nutné vykonat na několika úrovních a to způsobem, které jsou běžně dostupné, bohužel častokrát málo využívané.

Databáze obsahují citlivá data o zákaznících, jejich osobní údaje, dále také finanční pohyby, údaje o svých zaměstnancích, jejich osobní údaje, platové zařazení a v neposlední řadě údaje o dodavatelích, finanční údaje a kompletní přehled sortimentu včetně nákupních cen. Proto je nutno tuto databázi zabezpečit tak, aby každý uživatel měl nastavenou svoji roli, a tou rolí se omezí jeho přístup k datům, aby nemohlo dojít k nežádoucí změně, odcizení nebo úplné ztrátě dat. Zajištění bezpečnosti databáze je ve dvou úrovních, a to autentizace uživatele na úrovni jeho přihlašovacího jména, tedy loginu, a přidělení práv uživatelům, tedy uživatelským účtům.

Na druhé straně je důležité, aby přihlášení zákazníka či zaměstnance do e-shopu bylo jednoduché, intuitivní a aby bylo podle předem stanovených pravidel, co se týče skladby loginu a hesla. Proto je nutné, aby se firmy věnovaly zabezpečení vlastní databáze, je to konec konců v jejich vlastním zájmu. Na zabezpečení databáze je potřeba vynaložit velké finanční zdroje, ale tato investice je z dlouhodobého hlediska výhodnou investicí.

Téma je aktuální zejména z důvodu využívaní databázových evidencí, počtu nově vzniklých e-shopů a aktualizace stávajících prosperujících e-shopů, kde se společnosti předhání s cílem získat co nejvíc zákazníků. To je hlavním důvodem pro co nejlepší zabezpečení relační databáze.

2 Cíl práce a metodika

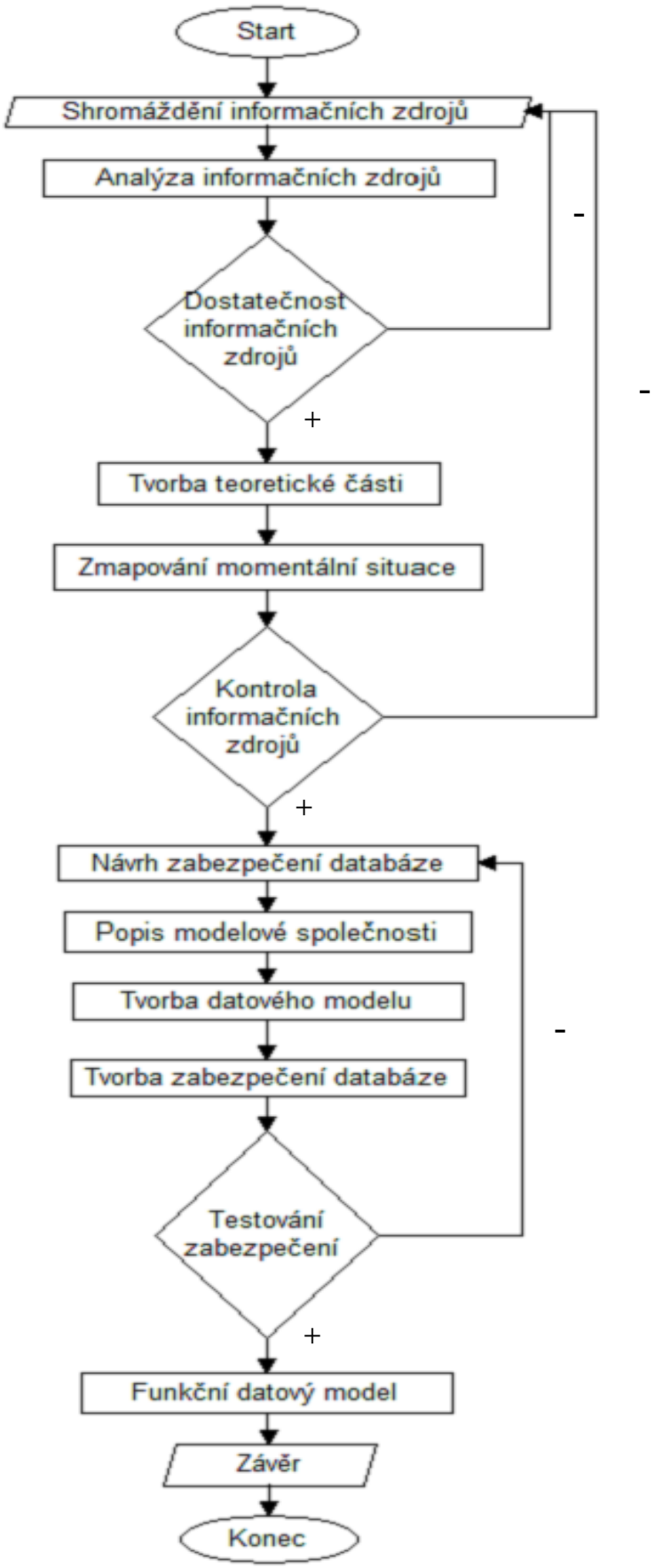
2.1 Cíl práce

Bakalářská práce je zaměřena na problematiku využití relačně databázové technologie v informačním zabezpečení provozu e-shopu. Hlavními cíli této práce je zmapovat momentální stav této problematiky a vymezit její relevantnost včetně požadavků na ni kladených. Objasnit teoretické principy relačně databázové technologie v souvislosti s problematikou zabezpečení databázově evidovaných dat. Na základě teoretických poznatků pak navrhnout přijatelné řešení této problematiky v souladu s identifikovanými požadavky. Získané poznatky následně zobecnit pro další možná uplatnění.

2.2 Metodika

Metodika této bakalářské práce se bude skládat z posloupností následujících kroků:

1. Shromáždění informačních zdrojů: metodika řešené problematiky je založena na analýze a studiu dostupných informačních zdrojů a existujících řešení v dané oblasti.
2. Podrobná analýza těchto shromážděných zdrojů: informační zdroje jsou především odborné publikace, které se věnují problematice relační databáze a jejímu zabezpečení.
3. Dostatečnost informačních zdrojů – v případě zjištění chybějících informací je nutné se znova vrátit do bodu 1 a doplnit tyto informace.
4. Tvorba teoretické části: tato teoretická část je tvořena s dostupných informačních zdrojů a jsou v ní uvedeny důležité části při tvorbě databáze a její zabezpečení.
5. Zmapování momentální situace řešené problematiky
6. Znova probíhá kontrola informačních zdrojů a v případě potřeby se informace doplní
7. Návrh zabezpečení databáze: návrh pro zabezpečení databáze je tvořen na základě dostupných informací, které jsou popsané v teoretické části této práce.
8. Popis modelové společnosti
9. Použití těchto vstupních dat na tvorbu datového modelu
10. Tvorba zabezpečení databáze
11. Testování zabezpečení – testováním se zajistí funkčnost zabezpečení a v případě chybného zabezpečení se opraví nebo doplní.
12. Po úspěšném testování vzniká funkční datový model
13. Závěr



Obrázek 1 Schématické znázornění zvoleného postupu řešení - Vlastní tvorba

3 Teoretická východiska

3.1 Databáze a její historie

Jedním z hlavních úkolů počítačového systému je ukládat a spravovat data. [1]
První SŘBD se objevují koncem 60. let. Vycházely ze dvou přístupů, a to z vzájemně propojených souborů dat (síťové a hierarchické databáze) a z fyzicky nezávislých souborů dat (relační databáze) [2]
Koncepce relační databáze byla původně vyvinuta E. F. Coddem, vývojářem z IBM.
V červnu 1970 publikoval DR. Codd článek nazvaný „Relační model dat pro velké sdílené datové banky“, který nastínil matematickou teorii toho, jak by bylo možno ukládat data a manipulovat s nimi za použití tabulkové struktury. [1]
První komerční sytémy založené na relačním modelu se začaly objevovat na přelomu 70. a 80. let. [2]
ORACLE se v roce 1979 stal prvním komerčně dostupným produktem SŘBD. [1]
V současné době se používají relační databáze téměř v každém podniku a vytvořily tak průmysl s obratem v miliardách dolarů ročně. [2]

3.2 Databázové technologie

Databáze je určitá uspořádaná množina informací (dat) uložená na paměťovém médiu. V širším slova smyslu jsou součástí databáze i softwarové prostředky, které umožňují manipulaci s uloženými daty a přístup k nim. Tento systém se v české odborné literatuře nazývá systém řízení báze dat SŘBD. Běžně se označením databáze – v závislosti na kontextu – myslí jak uložená data, tak i software SŘBD. Uspořádání databáze je hierarchické-nejvýše je databáze, ta obsahuje tabulky, každá z těchto tabulek je pak složena z několika sloupců. A konečně data, která jsou ukládána v řádcích, kterým se říká záznamy. Jednotlivé položky záznamu se nazývají pole tabulky. Databázové systémy v dnešní době umožňují a ulehčují práci s daty nejen programátorům ale hlavně uživatelům. [3]

V tuto chvíli je nutné, aby se vysvětlily jednotlivé zkratky, které popisují celý systém.

IS - Informační systém je tvořen technickými prostředky a také svými uživateli
DBS - Databázový systém je tvořen softwarem – systémem řízení báze dat a také samotnými daty, které jsou tímto systémem zpracovávána. Pod pojmem databázový systém rozumíme jak samotný program, tak i data, se kterými pracujeme. [2]
DB - Databázi chápeme jako úložiště údajů, které jsou uloženy a zpracovávány nezávisle na aplikačních programech.
Databázový systém můžeme chápat jako počítačový systém správy záznamů uložených v bázi dat. V takovém systémů jsou nejdůležitější následné funkce:

- přidání nového prázdného souboru do databáze
- vložení nových dat do existujícího souboru
- výběr dat z existujícího souboru
- oprava dat v existujícím souboru
- zrušení dat v existujícím souboru
- zrušení existujícího souboru z databáze

Definice databázového systému a jedna z nejčastějších používaných je následovná: Databázový systém DBS tvoří databáze DB a systém řízení báze dat SŘBD. Můžeme to napsat také jako vzorec DBS = DB+SŘBD

SŘBD – systém řízení báze dat je programové vybavení, které umožní zabezpečit všechny požadované vlastnosti databázového systému a manipulovat s těmito daty.

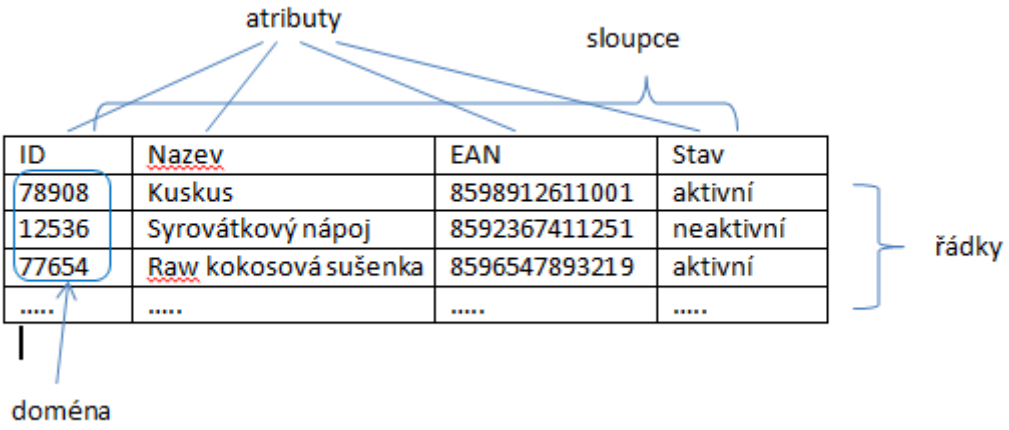
3.3 Relační datový model

Relační model navrhnutý Dr. Coddem představoval pokus o zjednodušení struktury databáze. Eliminoval z databáze explicitní struktury rodič-potomek a místo toho reprezentoval v databázi všechna data jako prosté tabulky datových hodnot sestávající z řádků a sloupců. Dr. Codd definoval v roce 1985 ve svém článku v Computerworldu 12 pravidel, kterými se musí databáze řídit, pokud mají být považovány za skutečně relační.

Coddových 12 pravidel se od té doby stalo polooficiální definicí relační databáze. Pravidla vychází z Coddovy teoretické práce na relačním modelu a v podstatě vyjadřují víc ideální cíl než vlastní definici relační databáze. Méně formální definice: Relační databáze je databáze, kde se veškerá pro uživatele viditelné data striktně uspořádají jako tabulky datavých hodnot a kde veškeré databázové operace pracují s těmito tabulkami. [1]

Relační databáze ukládá data ve vztazích, které uživatel vidí jako tabulky. Každý vztah je složen z uspořádaných n-tit, neboli záznamů, a atributů neboli polí. Skutečné uspořádání záznamů v databázi je zcela nepodstatné a každý záznam v tabulce je identifikován polem, které obsahuje unikátní hodnotu. Toto jsou dva základní rysy relační databáze, které umožňuje, že data mohou existovat nezávisle na svém fyzickém uložení v počítači. [5]

Základem relační databází jsou databázové tabulky. V rámci tabulek rozlišujeme sloupce a řádky. Pro sloupce databázové tabulky volíme název tak, aby byl v rámci tabulek jednoznačný. Dále pro sloupec definujeme jeho datový typ, který určuje, jaká data zde budeme ukládat. Řádek tabulky slouží k vlastnímu uložení dat a představuje samostatnou fyzickou entitu. Bývá také označován jako záznam. [3]



Obrázek 2 Znázornění obecného tvaru relační tabulky. Vlastní tvorba

3.4 Relační vztahy

Aby bylo možné aplikovat relační teorii, je zpravidla nutné vytvořit více databázových tabulek a definovat relační vztahy mezi nimi. [6]
Vztah mezi tabulkami existuje, pokud nějakým způsobem můžeme propojit záznamy z první tabulky se záznamy z druhé tabulky. Vztah můžeme zřídit prostřednictvím množiny primárních a cizích klíčů. [5]

3.4.1 Klíče

Klíče jsou speciálními poli, která hraje v tabulce velmi významnou roli. Typ klíče určuje jeho význam v tabulce. Existuje několik typů klíčů, ale nejvýznamnější jsou primární klíč a cizí klíč. [5]

Primární klíč

Primární klíč je jednoznačný identifikátor každého záznamu. Může to být sloupec, případně kombinace více sloupců, které slouží pro jednoznačnou identifikaci každého řádku tabulky. Hodnota pole primárního klíče musí být v rámci tabulky jedinečná. Pole primárního klíče musí obsahovat konkrétní hodnotu, tedy nesmí nikdy nabýt hodnoty NULL. Bez primárního klíče není možné definovat relace mezi tabulkami. Při výběru atributů, které budou tvořit primární klíč, je potřeba dbát na to, aby vybrané atributy skutečně plnily úlohu identifikačního klíče. [6]

Cizí klíč

Zjistíme-li, že dvě tabulky by k sobě měly mít vztah, zřídíme mezi nimi vztah většinou tak, že vezmeme kopii primárního klíče první tabulky a začleníme jí do druhé tabulky, kde se stane cizím klíčem. Jméno „cizí klíč“ je odvozeno od faktu, že druhá tabulka už má svůj primární klíč a že primární klíč první tabulky je z pohledu druhé tabulky „cizí“. [5]

Kromě pomoci se zřizováním vztahu mezi dvojicemi tabulek, pomáhá cizí klíč také implementovat a zajišťovat integritu na úrovni vztahů. Neboli, záznamy v obou tabulkách

jsou vždy ve správném vztahu, protože hodnota cizího klíče musí odpovídat hodnotě primárního klíče, na který se odkazuje. Integrita na úrovni vztahů také předchází obávaným „sirotčím“ záznamům, například záznamu o objednávce, ke kterému není přirazen jednoznačný zákazník. Pokud nevíme, kdo objednávku udělal, nemůžeme ji zpracovat a zcela jasně ji nemůžeme nikomu naúčtovat. [5]

3.4.2 Datová normalizace

U návrhu relačního schématu je důležité dodržovat určitá pravidla, která nazýváme normálny formy. Normalizace obvykle vede k odstranění redundancí a značně zefektivňuje práce s databázovými tabulkami. [6]

První normální forma 1NF

Tabulka splňuje podmínku 1NF tehdy, když všechny atributy (sloupce) jsou atomické, což znamená dále nedělitelné. [6]

Druhá normální forma 2NF

Tabulka splňuje podmínku pro zařazené do 2NF tehdy, když splňuje podmínku 1NF a každý atribut kromě primárního klíče musí být úplně závislý na celém primárním klíči. Druhá normální forma se proto týká jen tabulek, které mají více primárních klíčů. Když má tabulka jen jeden primární klíč, tak podmínka pro 2NF je splněna automaticky. [6]

Třetí normální forma 3NF

Tabulka je v třetí normální formě tehdy, když je v druhé normální formě a zároveň neexistují závislosti neklíčových sloupců tabulky. [6]

Boyce-Coddova normální forma BCNF

Tato normální forma, někdy nazývaná jako 3.5NF je vlastně původní definicí 3NF, jak ji v 70. letech publikovali její autoři. BCNF je vymezena stejným pravidlem jako 3NF, ale je

přísnější v tom, že toto pravidlo musí platit i mezi hodnotami uvnitř složeného primárního klíče. [6]

3.4.3 Datová integrita

Pojem integrita dat se týká správnosti a úplnosti dat v databázi. Když je obsah databáze změněn příkazy INSERT, DELETE nebo UPDATE, může dojít ke ztrátě integrity uložených dat, a to mnoha způsoby. Například:

- do databáze mohou být přidána neplatná data, jako např. objednávka, která udává neexistující produkt
- stávající data mohou být změněna na špatnou hodnotu, jako např. přeřazení prodejce do neexistujícího kalendáře.
- změny v databázi mohou být ztraceny z důvodu systémové chyby nebo výpadku paměti
- změny mohou být aplikovány částečně, jako např. zapsání objednávky na určitý produkt bez upravení množství, jenž je k dispozici na skladě

Jednou z nejdůležitějších rolí relačního DBŘS (DMBS) je nejvyšší možná míra ochrany integrity uložených dat. [1]

Entitní integrita

Primární klíč databáze musí mít jednoznačnou hodnotu pro každý řádek tabulky, jinak databáze ztratí svou integritu jako model vnějšího světa.

Když je pro tabulku definován primární klíč, databáze automaticky kontroluje jednoznačnost hodnoty primárního klíče pro každý příkaz INSERT nebo UPDATE provádění na tabulce.

Pokus o vložení řádku s duplicitním primárním klíčem nebo o aktualizaci řádku tak, aby byl primární klíč duplicitní, skončí chybou. [1]

Referenční integrita

Referenční integrita zajišťuje, že relace (vztah) mezi primárními klíči a cizími klíči v odkazujících se tabulkách je vždy zachována. Například databáze zaměstnanců obsahující tabulku zaměstnanec a město. Každé město má svůj primární klíč. Tabulka zaměstnanec má pole cizího klíče, jehož hodnota je shodná s primárním klíčem města, z kterého zaměstnanec pochází. Referenční integritou lze zajistit, aby nebylo možné smazat město, z něhož pocházejí některý zaměstnanci, případně změnit jeho hodnotu primárního klíče. [7]

Doménová integrita

Doména je množinou platných datových hodnot. Výhodou použití domén je to, že definice platnosti dat je uložena na jednom místě v databázi. Pokud se definice později změní např., pokud se společnost rozroste a je potřeba použít i rozsah 200 až 299, je mnohem jednodušší změnit jednu definici domény, než měnit mnoho omezení sloupců rozptýlených v databázi. Ve velké podnikové databázi mohou být doslova stovky nadefinovaných domén a výhody domén pro správu změn mohou být velmi značné. [2]

3.5 Jazyk SQL

Důležitou rolí jazyka SQL je definování struktury a uspořádání dat v databázi. [1]
V této části bude popsáno, jak se vytváří databáze a její tabulky, dále práce s daty a v poslední části bude popsáno zabezpečení SQL, které chrání data před poškozením, zneužitím či samotným smazáním. Dále budou popsány prostředky jazyka SQL, jež umožňují vytvářet databáze a tabulky a definovat jejich strukturu. Příkazy SELECT, INSERT, DELETE, UPDATE, COMMIT a ROLLBACK, provádějí manipulaci s daty v databázi. Tyto příkazy se souhrnně označují jako DML (Data Manipulation Languge, jazyk pro manipulaci s daty). Příkazy jazyka DML mohou měnit data uložená v databázi, avšak nemohou měnit jejich strukturu. Žádný z těchto příkazů například nevytváří a neodstraňuje tabulky nebou sloupce. Změny struktury databáze se realizují jinou množinou příkazů jazyka SQL, která je obvykle nazývána DDL (Data Definition Language, jazyk pro definici dat). Pomocí příkazů jazyka DDL můžeme definovat a vytvořit novou tabulku, odstranit nepotřebnout abulku, změnit definici stávající tabulky a mnoho dalších. Jádro

jazyk DDL je založeno na třech anglických slovesech SQL a to CREATE, DROP a ALTER. [1]
Pro řízení přístupových práv jednotlivých uživatelů systému a také např. pro řízení transakcí je označována jako jazyk pro řízení dat DCL – data control language. Do této skupiny patří následující příkazy GRANT, ALTER USER, REVOKE GRANT USER, DROP USER, REVOKE.
Speciální oblastí jazyka SQL je Transaction Control Commands TCC, která obsahuje příkazy pro řízení transakcí. Do této skupiny patří příkazy COMMIT, ROLLBACK, SAVEPOINT [3]

COMMIT

Všechny změny, které v databázi provedete, je potřeba potvrdit. Toto potvrzení provádí řada aplikací automaticky za uživatele, ale běžně k potvrzení transakce slouží příkaz commit. Jakmile tedy provedeme změnu, je nutné tento příkaz zavolat.
COMMIT; [3]

ROLLBACK

Předtím, než transakci potvrdíme, lze ji odvolat. Odvolání všech nepotvrzených transakcí je řešeno příkazem rollback. [3]
ROLLBACK;

SAVEPOINT

Pokud si nejsme zcela jisti změnou, kterou hodláme provést anebo plánujeme provedění celé řady změn, které spolu souvisí, můžeme vytvořit tzv. návratový bod. Jakmile zavoláme rollback s názvem bodu návratu, tak se odvolají všechny změny, které jsme od tohoto návratového bodu provedli. [3]

4 Zabezpečení databáze

Zabezpečení databáze, je nutností, je hlavní prioritou každé společnosti. Každá společnost, větší či menší, musí svoji databázi zabezpečit proti ztrátě informací, jejich neúmyslné změně nebo proti konkurenci. Uživatelé musí mít přesně stanovena svoje práva a možnosti, proto je nutné po fázi datového modelu přejít na přesná pravidla, podle kterých se tito uživatelé budou zakládat do systému, s jakou rolí nebo popřípadě jaké pohledy se vytvoří, aby se pokryly všechny potřeby uživatelů.

Vytváření a vyžadování procedur zabezpečení umožňuje ochraňovat to, co se stále více stává nejdůležitějším majetkem společnosti: data. Díky ukládání do databáze jsou data pro společnosti užitečnější a dostupnějším jsou však také náchylnější k neoprávněnému přístupu. Takové pokusy o přístup je třeba zjišťovat a také je třeba jim zabraňovat. Databáze Oracle má několik vrstev zabezpečení a umožňuje každou z těchto vrstev kontrolovat. Mezi možnosti zabezpečení patří v databázi Oracle role, profily a přímé udělování oprávnění. [8]

Vytvořený uživatelský účet je pouze prázdnou schránkou, která definuje uživatele a jeho jméno a heslo, ale nikoli práva. Ta je potřeba nadefinovat zvlášť. K definici práv slouží dvojice příkazů GRANT a REVOKE. První zmínění příkaz práva přiděluje, druhý je odebírá. Pokud se podílíme na velkém databázovém projektu, vytváříme řadu uživatelů, kteří mají stejná oprávnění. Například zaměstnanci vytvářejí databázové procedury, ale nemohou modifikovat tabulky apod. Abychom pro každý uživatelský účet nemuseli definovat znovu stejná práva, nadefinujeme tzv. roli. Role získá práva a pak snadno přiřadíme práva uživateli nepřímo prostřednictvím této role. [3]

Kromě role je dobrá možnost zpřístupnit data také formou nadefinovaného pohledu.

Další důležitou součástí zabezpečení databáze jsou profily. Profilem se uživateli určí tvar hesla, jeho délka nebo možnost opakováni.

4.1 Uživatelé

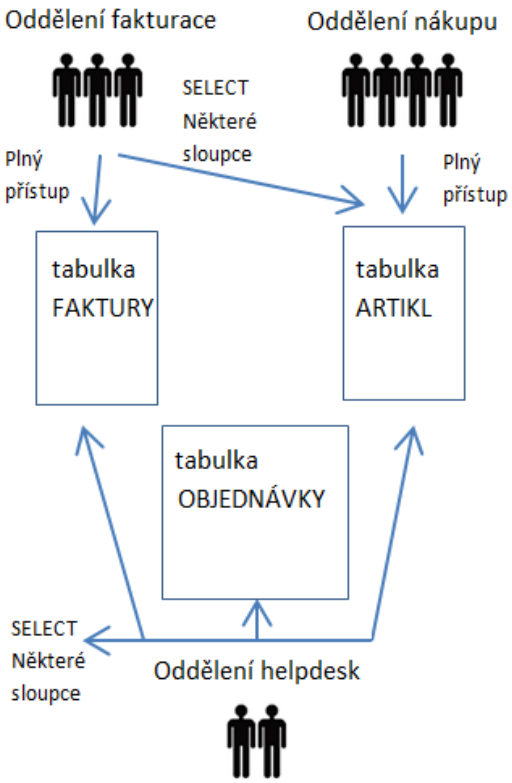
Když mluvíme o uživatelích v databázi, obvykle máme na mysli tři různé typy uživatelů: koncoví uživatelé, vlastníci aplikací a administrátoři.

První skupinu tvoří lidé, kterým se obvykle říká koncoví uživatelé a kteří se přihlašují k databázi, aby používali a udržovali vlastní data. Je potřeba, aby dokázali vytvořit relaci a tak se přihlásit k databázi. Potřebují oprávnění k objektům v oblastech, které mohou prohlížet a modifikovat, ale obvykle nepotřebují oprávnění, která by jim umožňovala neomezený přístup ke všem oblastem v databázi. Jinými slovy, jejich přístup je obvykle omezen na oblasti, které potřebují ke své práci. [9]

Další skupinou jsou uživatelé, kteří vlastní objekty v databázi. Nejsou to uživatelé ve smyslu koncových uživatelů. Tuto skupinu tvoří jeden nebo několik programů, označovaných jako aplikace, které napsali vývojáři nebo jiný dodavatelé, aby usnadnili uživatelům práci a umožnili jim pracovat efektivněji. [9]

Třetí skupinou uživatelů jsou správci databáze. Jak již jméno říká, správci se starají o údržbu vlastní databáze. DBA mají obvykle k databázi nejvyšší přístupová práva. Správce databáze, výtváří databáze role a přiděluje uživatelům jejich oprávnění. [9]

V naší vzorové společnosti a její databázi se budeme věnovat jenom koncovým uživatelům. Každému uživateli bývá zpravidla udělen identifikátor, který přesně identifikuje uživatele. Je to z důvodu toho, aby nedošlo k omylu při udělování práv nebo role. Uživatelé jsou aktéři v databázi. Pokaždé, když DBŘS načítá, vkládá, odstraňuje nebo aktualizuje data, činí tak jménem nějakého uživatele. Systém umožňuje nebo zamezuje akci v závislosti na tom, který uživatel požaduje vytváření, jak je vidět na obrázku číslo 3. [1]



Obrázek 3 Schéma zabezpečení pro vzorovou databázi. Vlastní tvorba

Při vytváření uživatelských účtů je naším úkolem zajistit zabezpečený účet s adekvátními oprávněními a správným výchozím nastavením. Nový databázový účet je možné vytvořit pomocí příkazu CREATE USER. Po vytvoření nebude mít účet žádné možnosti a uživatelé se nebudou schopni přihlásit, dokud jim nebudou přidělena příslušná oprávnění. Všechna nezbytná oprávnění pro uživatelský účet je možné specifikovat v jednom příkazu CREATE USER. [8]

Pro tvorbu nového uživatele použijeme CREATE USER
CREATE USER jméno **IDENTIFIED BY** heslo;

Chceme-li donutit uživatele, aby si změnil své heslo ihned po prvním přihlášení, použijeme nato příznak v této formě:
CREATE USER jméno **IDENTIFIED BY** heslo **PASSWORD EXPIRE** ;

Je-li zvolena možnost BY, uživatel se bude muset při každém přihlášení do systému identifikovat svým heslem. Pokud se zvolí možnost EXTERNALY, spolehne se databázový systém Oracle na přihlašovací údaje použity při přihlášení do systému. [10]

Kromě uživatelského jména je možné všechny parametry příkazu CREATE USER měnit prostřednictvím příkazu ALTER USER.
ALTER USER jméno **IDENTIFIED BY** nové_heslo;

Uživatelské účty je možné z databáze odstranit pomocí příkazu DROP USER. Příkaz má jediný parametr- CASCADE, který umožňuje odstranit všechny objekty ve schématu uživatele před odstraněním uživatelského účtu. [10]
DROP USER jméno;

4.2 Práva

Nyní budu popisovat akce, které uživatel může provádět na daném databázovém objektu. Uživatel například může mít právo načítat nebo vkládat řádky do konkrétní tabulky, ale nemusí mít právo odstraňovat nebo aktualizovat. Různí uživatelé mohou mít různé sady práv.

Standart SQL1 specifikuje čtyři základní práva pro tabulky a pohledy:

- Právo SELECT umožňuje načítání dat z tabulky
- Právo INSERT umožňuje vkládat nové řádky do tabulky
- Právo DELETE umožňuje odstraňovat řádky dat z tabulky
- Právo UPDATE umožňuje upravovat řádky dat v tabulce

Tato čtyři práva jsou podporována téměř všemi komerčními produkty SQL
Vyvoříme-li tabulku pomocí příkazu CREATE TABLE, stáveme se jejím vlastníkem a máme pro ni plná práva. Ostatní uživatelé zpočátku nemají k nově vytvořené tabulce žádná práva. Máli jim být povolen přístup do tabulky, musíme jim explicitne přidělit práva pomocí příkazu GRANT [1]

4.2.1 Přidělování práv

Základní příkaz GRANT se používá pro přidělování bezpečnostních práv k objektům databáze pro uživatele. Příkaz GRANT zahrnuje seznam práv, která mají být přidělena, název tabulky, již se práva týkají, a identifikátor uživatele, kterému jsou tato práva přidělována. Příkaz poskytuje dva zkrácené povely, které můžeme použít při přidělování mnoha práv nebo při jejich přidělování mnoha uživatelům. Namísto uvedení všech práv dostupných pro konkrétní objekt můžeme použít klíčová slova ALL PRIVILEGES. Tento příkaz dává uživateli plný přístup k tabulce. [1]

4.2.2 Odebírání práv

Ve většině SQL mohou být práva, která jsme přidělili pomocí příkazu GRANT, odebrána pomocí příkazu REVOKE. Příkaz REVOKE má strukturu, která je téměř stejná s příkazem GRANT a udává seznam práv, jež mají být odebrána pro konkrétní databázový objekt jednomu nebo více uživatelům. Příkazem lze odebrat některá nebo všechna práva, jež jsme dříve uživateli přidělili. [1]

Syntaxe příznaků GRANT a REVOKE

GRANT {ALL|<seznam-oprávnění>[(sloupec [sloupec...])]} ON <jméno-tabulky | jméno-náhledu> TO {PUBLIC |<seznam-uživatelů>} [WITH GRANT OPTION]

REVOKE [GRANT OPTION FOR] {ALL|<seznam-oprávnění>[(sloupec [sloupec...])} ON <jméno-tabulky | jméno-náhledu> FROM {PUBLIC |<seznam-uživatelů>}

Tabulka 1: Popis syntaxe příznaků GRANT a REVOKE

Popis:	
sloupec	je sloupec, ke kterému se přidělovaná oprávnění vztahují.
jméno_tabulky	je jméno existující tabulky, ke které se přidělovaná oprávnění vztahují.
jméno_náhledu	je jméno existujícího náhledu, ke kterému se přidělovaná oprávnění vztahují.

< seznam-uživatelů >	je seznam uživatelů, kterým se oprávnění přidělují.
WITH GRANT OPTION	předává GRANT pravomoci pro oprávnění uvedené v GRANT příkazu pro < seznam-uživatelů>.

Zdroj: vlastní tvorba

4.3 Role

Pokud začneme vytvářet uživatele pro informační systém, třeba i jen pro malé a střední firmy, a přidělovat jim přístupová práva, brzy přijdeme na to, že pro mnoho z nich jsou tato privilegia stejná. Proto jsou důležitou součástí přístupových práv role, které umožňují sdružovat uživatele do skupin. Známe serverové a databázové role.
Databázové role jsou specifické pro danou databázi a umožňují přístup individuálnímu uživateli nebo skupině uživatelů k dané databázi v rozsahu, který je určený tou kterou rolí. Zajímavá je přístupová role PUBLIC, protože jejím členem je každý nově vytvořený uživatel databáze. [4]

Uživatelské role jsou role, pod kterými bude uživatel vystupovat. Základ, je role CONNECT, která umožňuje připojení k databázi, dotazování, změnu dat, ale nikoli změnu struktury databáze. Tato role je základem pro role další a je minimem, které by měl uživatel mít přiděleno. Další rolí je RESOURCE, která dovoluje i změnu struktury databáze, vytváří procedury, funkce, triggery, sekvence apod. Nejvyšší role je DBA, která označuje databázového administrátora. [3]

V databázi je možné vytvořit role s určitými právy pomocí příkazu CREATE ROLE. Zjednodušeně by se mohla syntaxe toho příkazu definovat následovně:

CREATE ROLE <Role_Name>[Not Identified] | [Identified By <heslo> | Externally | globally]

Chceme-li například vytvořit roli NEW_ROLE, ke které je přístup chráněn heslem, můžeme použít následující příkaz:
CREATE ROLE new_role IDENTIFIED BY my_new1;

Role, u které není vyžadováno heslo, se vytvoří následovně:

CREATE ROLE new_role; [9]

Tabulka 2: Parametry pro vytváření rolí.

Paremetr	**Popis**
Role_name	Určuje jméno role
Not Identified	Určuje, že pro roli nebude vyžadováno heslo.
Identified by <heslo>	Určuje, že pro aktivaci role je nutné použít příkaz set role a že je potřeba uvést heslo
Identified Externally	Určuje, že se při aktivaci role má použít ověření uživatele operačním systémem. Příkaz vytvoří uživatel, který se může do databáze přihlásit na základě externí identifikace.
Identified Globally	Určuje, že aby mohl uživatel použít tuto roli, musí být ověřen adresátovou službou. Příkaz vytvoří uživatel, který má právo se přihlásit k databázi bez uvedení hesla a provést administrativní úkon v databázi.

Zdroj: THERIAULT, Marlene a Aaron NEWMAN [9]

Jsou dva druhy rolí, jedny jsou automaticky aktivní, když se uživatel přihlásí, a druhé, které třeba aktivovat pomocí příkazu SET ROLE. Příkaz SET ROLE obvykle provádí za uživatele aplikace a často je k tomu potřeba heslo nebo ověření operačním systémem. Pokud je vyžadováno heslo, musíme dříve, než můžeme roli použít, provést příkaz SET ROLE.

SET ROLE new_role IDENTIFIED BY my_new1;

V tomto příkladu se role NEW_ROLE aktivuje, když je proveden příkaz set role a je zadáno správné heslo. [9]

4.4 Pohledy

Pohled je virtuální tabulka v databázi, jejíž obsah je definován dotazem. Pro uživatele databáze pohled vypadá jako skutečná tabulka s množinou pojmenovaných sloupců a řádků dat. Na rozdíl od tabulky však u pohledu nejsou ve skutečnosti v databázi uloženy hodnoty. [1]

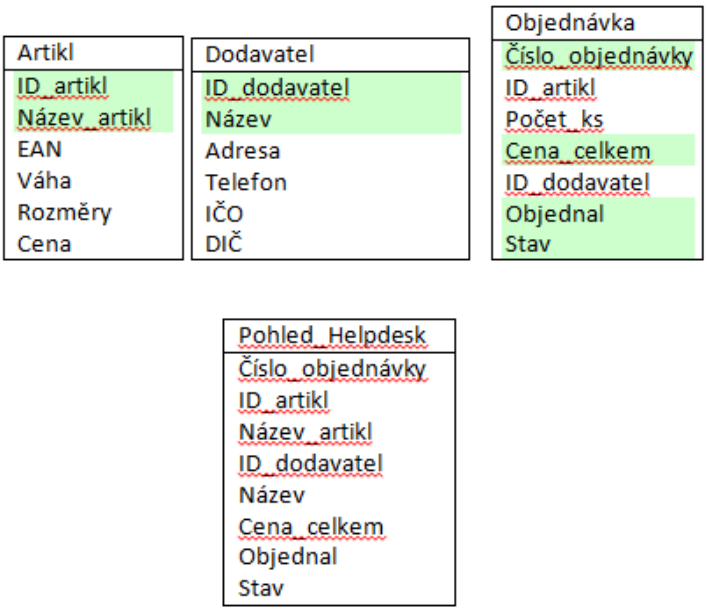
Výhody pohledů

- Zabezpečení – každému uživateli lze udělit oprávnění k přístupu do databáze pouze prostřednictvím malé sady pohledů, které obsahují konkrétní data, k nimž má uživatel povolen přístup, takže omezují přístup uživatele k uloženým datům
- Jednoduchost dotazu – pohled může zobrazovat data z několika odlišných tabulek a překládat je jako jedinou tabulku a měnit tak více tabulkové dotazy na pohled.
- Jednoduchost struktury – pohledy mohou dávat uživateli přizpůsobený pohled na strukturu databáze a prezentovat databázi jako sadu virtuálních tabulek, které mají pro tohoto uživatele význam
- Izolování od změn – pohled může předávat konzistentní a nezměněný obraz struktury databáze
- Integrita dat – pokud je přístup k datům prováděn pouze skrze pohledy nebo jsou data zadávána skrze pohledy, databáze může automaticky kontrolovat data, aby byla zajištěna omezení integrity. [1]

Nevýhody pohledů

- Výkon – pohledy vytváří vzhled tabulky, ale DBŘS musí stále překládat dotazy na pohledy na dotazy na zdrojové tabulky. Pokud je pohled definován složitým dotazem na více tabulek, pak se i jednoduchý dotaz na pohled stane komplikovaným a jeho provedení může trvat dlouho
- Omezení aktualizací – když si uživatel pokouší aktualizovat řádky pohledu, databáze musí přeložit požadavek na aktualizaci řádků zdrojových tabulek. To je možné u jednoduchých pohledů, ale složitější pohledy nemohou být aktualizovány a slouží jen pro čtení. [1]

K vytváření pohledů složí příkaz CREATE VIEW jak je znázorněno na obrázku 4. Pro úspěšné vytvoření pohledu musíme mít přístup ke všem tabulkám, na které se odkazujeme v dotazu. Pečlivým definováním pohledu a přidělením práv uživatelům pro přístup k pohledům, ale ne jeho zdrojovým tabulkám, můžeme efektivně omezit přístup uživatele pouze na vybrané sloupce a řádky. Pohledy tedy nabízí způsob uplatňování velmi přesné kontroly nad tím, která data jsou viditelná, pro jaké uživatele [1]



Obrázek 4 Použití pohledu pro omezený přístup k sloupcům. Vlastní tvorba

Syntaxe je následující:

CREATE OR REPLACE VIEW jméno_pohledu **AS** dotaz;

Je-li pohled vytvořen s dodatkem OR REPLACE a pokud už existuje pohled stejného jména, bude původní pohled nahrazen novou definicí. Původní oprávnění zůstanou beze změny. [9]

4.5 Profily

Pomocí uživatelských profilů je možné omezovat množství systémových a databázových prostředků, které budou k dispozici uživateli a pro správu omezení z hlediska hesel. Pokud nejsou v databázi vytvořené žádné profily, bude použit výchozí profil DEFAULT určující neomezené prostředky pro všechny uživatele. Profily se vytvářejí prostřednictvím příkazu CREATE PROFILE. K vytvoření profilu je potřeba mít systémové oprávnění. [8]

Správa hesel

Pomocí profilu je možné spravovat dobu platnosti, možnost opětovného používaní a složitost hesel. Je například možné omezit dobu platnosti hesla a zablokovat účet, jehož heslo je příliš staré. Můžeme požadovat, aby bylo heslo středně složité a blokovat účty s opakovanými selháními při pokusech o přihlášení. [8]

Tabulka 3: Parametry hesla v profilu

Parametr	**Popis**
Failed_login_attempts	Povolený počet neúspěšných pokusů o přihlášení do zamčení hesla.
Password_life_time	Počet dní, během kterých je mořné používat stejné heslo. Není-li nastaven parametr grace_time a není-li během určené doby heslo změněno, je po této době platnost hesla ukončena a další přihlášení již není možné.
Password_reuse_time	Počet dní, kdy není možné použít stejné heslo. Pokud je hodnota nastavena, musí být parametr password_reuse_max na hodnotu UNLIMITED.
Password_reuse_max	Počet změn hesla, během kterých není možné použít stejné heslo. Pokud je hodnota nastavena, musí být parametr password_reuse_time nastaven na hodnotu UNLIMITED .
Password_lock_time	Čas po dosažení maximálního počtu neúspěšných pokusů o přihlášení, po který zůstane heslo uzamčeno.
Password_grace_time	Počet dní, během kterých je uživatel varován, že heslo je potřeba změnit, a je mu umožněn přístup. Pokud během této

	periody nedojde ke změně hesla, ukončí se platnost hesla a účet se uzamkne.
Password_verify_function	Tento parametr umožňuje specifikovat skript, pomocí kterého se ověří síla hesla. Můžeme si vytvořit vlastní proceduru nebo použít software třetí strany.

Zdroj: THERIAULT, Marlene a Aaron NEWMAN [9]

Je-li nastaven parametr failed_login_attempts, uzamkne server účet automaticky po překročení stanoveného počtu neúspěšných pokusů o přihlášení v řadě. Došlo-li k uzamčení hesla kvůli překročení stanoveného počtu neúspěšných pokusů o přihlášení v řadě, může být účet odemčen automaticky po stanovené době, která je dána nastavením parametru password_lock_time. Je-li tento parametr nastaven na hodnotu unlimited, k automatickému odemčení účtu nedojde. Účty, které byly uzamčeny z jiného důvodu, musí být odemčeny uživatelem s oprávněním DBA. Pro každého uživatele se udržuje počítadlo neúspěšných pokusů o přihlášení. Když se uživatel úspěšně přihlásí, je toto počítadlo nastaveno na 0. Na hodnotu 0 je počítadlo také nastaveno v případě, že byl účet odemčen po uplynutí doby stanovené parametrem password_lock_time [8]

4.6 Zálohování

Při sebevětší péči o technický stav databázového serveru může dojít k havárii tohoto počítače a k destrukci dat na jeho pevných discích. I při bezchybném provozu databázového serveru může nastat havarijní situace např. díky omylu administrátora při manipulaci s databází, při hrubém omylu uživatelů např. mazání dat, která smazána být neměla a podobně. To vše a ještě řada dalších potenciálních problémů je důvodem pro velmi pečlivé zálohování databázového souboru, které je nezbytným prvek při provozu informačního systému. [3]

Pro zálohování by mělo platit:

- zálohování je prováděno dostatečně často – obvykle jedenkrát za den

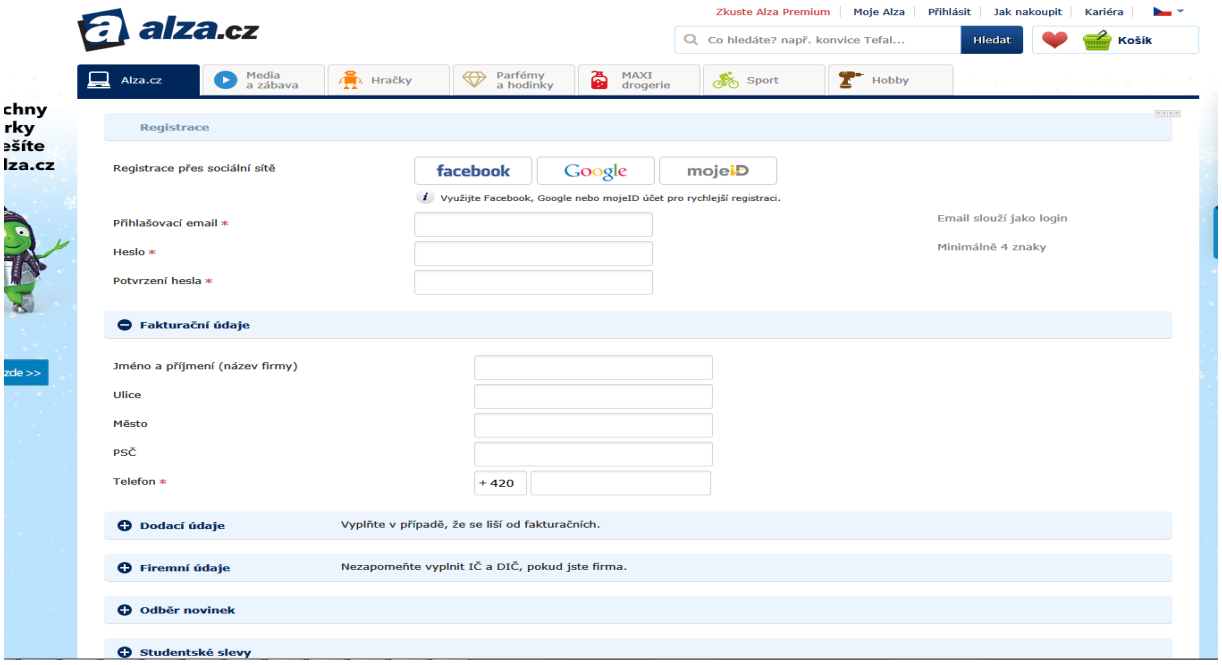
- zálohování probíhá v době nejmenšího provozu – obvykle v hlubokých nočních hodinách
- záloha se vytváří (nebo dodatečně přesouvá) na jiném počítači, než na kterém je provozován databázový server
- zálohy databáze jsou uchovávány po určitou dobu, doporučuje se následující scénář
    - v rámci posledního týdne jsou uchovávány zálohy každého dne
    - v rámci posledního měsíce je uchovávána jedna záloha z každého týdne
    - v rámci posledního roku je uchovávána jedna záloha z každého měsíce [3]

Nejobvyklejším způsobem zálohování je zkopírování a komprese celého databázového souboru. [3]

5 Zmapování současné situace

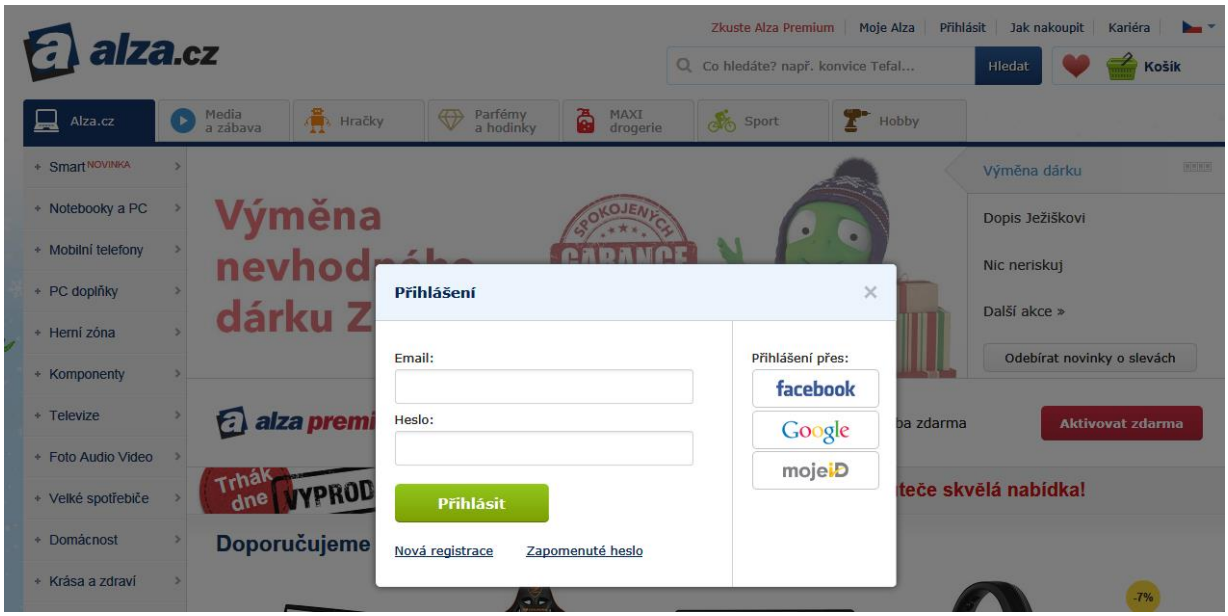
5.1 Registrace a přihlášení do e-shopu

Současná situace v zabezpečení e-shopu je na první pohled ze strany zákazníka stejná. Skoro na každém e-shopu je nutno se před potvrzením objednávky registrovat. Tato registrace v sobě ukrývá i přihlašovací údaje. Samozřejmě ne každý e-shop má tuto registraci stejnou. Jak je vidět na obrázku 5, e-shop společnosti Alza nabízí i přihlášení přes Facebook, Google a nebo mojeID. Tato možnost je rychlá a v případě, že tuto možnost zákazník využije, použijí se jeho údaje, které už vyplnil na webových stránkách Facebook, Google a nebo mojeID, a tak se celá jeho registrace značně urychlí. Samozřejmě toto není podmínkou registrace a zákazník, který nepoužívá Facebook, Google, nebo mojeID, si vyplní svou prázdnou registraci. Jak je z obrázku zřejmé, je nutné vyplnit minimálně ta políčka, která jsou označena hvězdičkou, ale pro odeslání objednávky je samozřejmě nutné vyplnit také políčka bez označení hvězdičky. Můžeme předpokládat, že tyto údaje musí být vyplněny podle nějakých klíčů. Klíčem jsou myšleny například délka hesla a jeho samotná skladba. Například heslo musí obsahovat číslo, text s minimální délkou znaků, tyto možnosti a omezení se nastavují také pro uživatele databáze, kteří zpracovávají údaje zákazníka, jeho objednávky, storna apod.



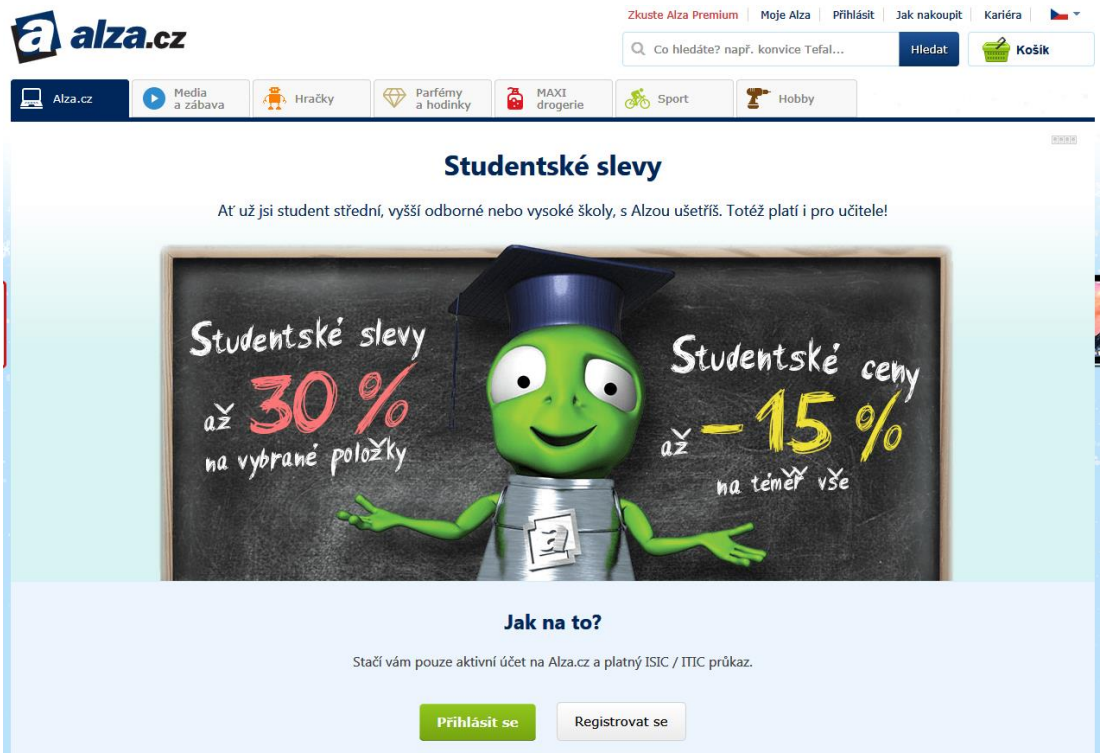
Obrázek 5 Maska-registrace zákazníka na e-shopu Alza

Po úspěšné registraci se zákazník bude mít možnost přihlásit se svými údaji a může začít nakupovat a využívat slevy. Bude moci nahlížet na své osobní údaje, které může také měnit podle potřeby, například při změně doručovací adresy. Dále bude mít přístup k náhledu svých objednávek a případných storen.



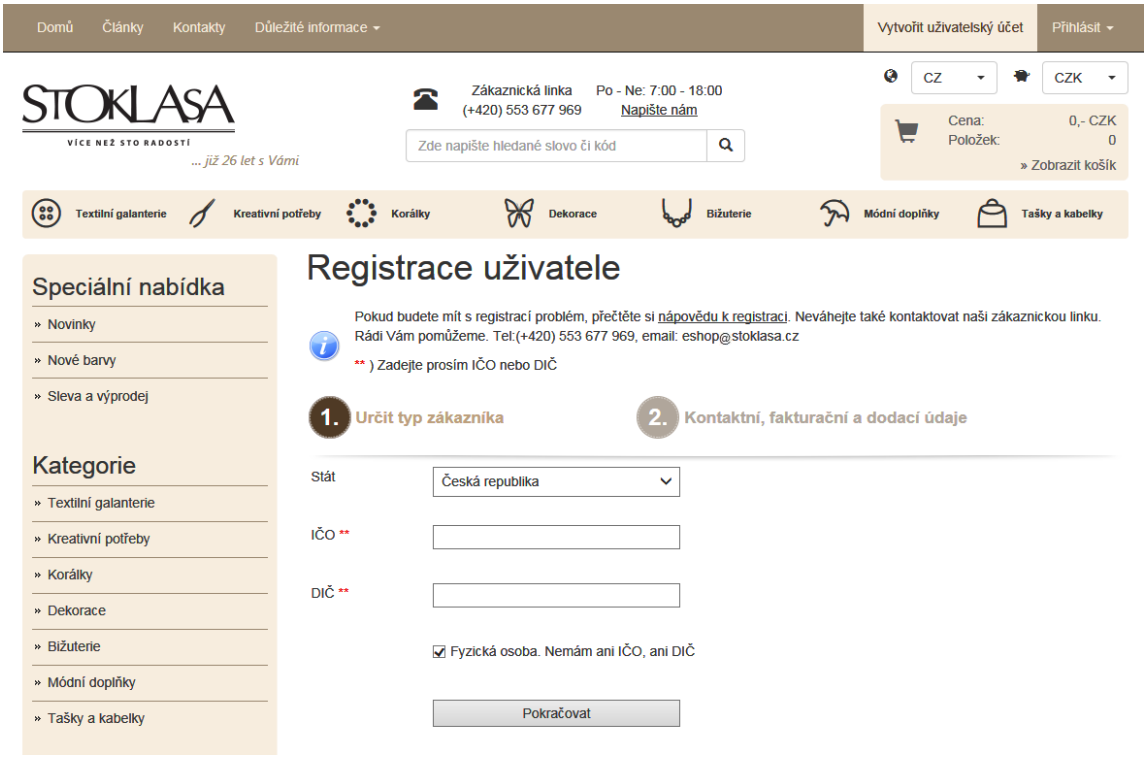
Obrázek 6 Maska-přihlášení na e-shopu Alza

Cílená registrace, určená v tomto případě jenom studentům, je skvělým lákadlem pro registraci a následný nákup.



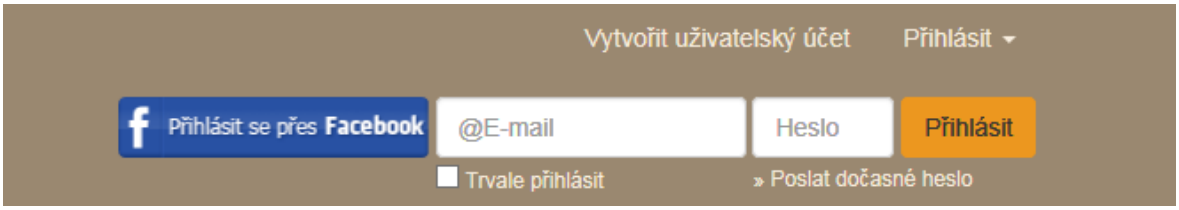
Obrázek 7 Maska-cílená registrace

Na další ukázce je možnost vidět, že e-shop společnosti Stoklasa umožňuje svým zákazníkům využít i toho, že jsou podnikatelé a to tak, že při registraci mají možnost uvést své IČO a DIČ. Pomocí těchto údajů má zákazník po vyplnění celé registrace nižší prodejní cenu oproti běžným zákazníkům. Tato možnost je velkým lákadlem pro vytvoření možné spolupráce. Již mnoho menších e-shopů nakupuje se zvýhodněnou cenou a prodává toto zboží na svých e-shopech, nebo tyto výhody využívají při tvorbě výrobků, které na e-shopech prodávají.



Obrázek 8 Maska-registrace zákazníka na e-shopu Stoklasa

E-shopy umožňují trvalé přihlášení, jak je vidět na obrázku 9, což znamená, že zákazník už nemusí vyplňovat své přihlašovací údaje a po otevření webové stránky už budou jeho přihlašovací údaje vyplněny, a tím pádem bude už zákazník přihlášen. Tato možnost není povinná.V případě, že zákazník zapomene své heslo, stačí, aby využil možnost poslat dočasné heslo, které se zašle na regitrovanou e-mailovou adresu. Práve z tohoto důvodu mnoho e-shopu vyžaduje pri registraci uvádět e-mail jako login. Předpokládejme, že zákazník v minulosti zapomněl svůj login, a proto již nebylo možné zákazníkův účet obnovit.



Obrázek 9 Maska-přihlášení na e-shopu Stoklasa

Po úspěšné registraci musí ještě zákazník souhlasit, že jeho údaje budou sloužit pro danou společnost pro zasílání novinek a akčních nabídek. Zároveň společnost tímto potvrzuje, že nebude poskytovat osobní údaje třetím stranám, což vyplývá ze zákona o ochraně osobních údajů.

6 Návrh zabezpečení databáze

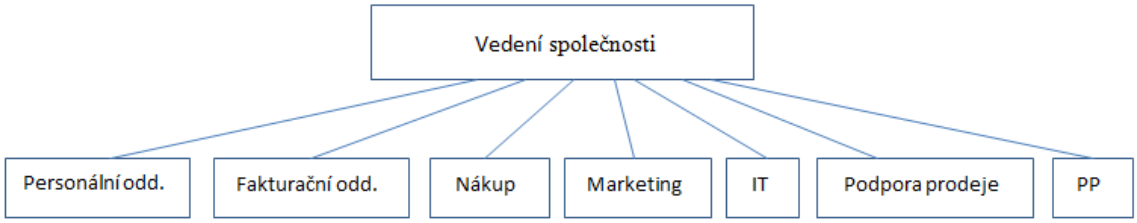
Návrh pro zabezpečení relační databáze je tvořen na základě dostupných informací, které jsou popsané v teoretické části této práce. Samotný návrh databáze „projekt“ je postavený na požadavcích smyšlené společnosti, který je popsán v kapitole 6.1. Modelová databáze i samotné zabezpečení jsou v rámci této práce omezeny pouze na ukázky, které vystihují popsané teoretické části a dané téma. Skutečný projekt v reálné společnosti by byl mnohem rozsáhlejší. Cílem prvního kroku je provést analýzu procesů v podnikatelském subjektu. Účelem je popis stavu a návrh zabezpečení.

6.1 Popis společnosti

Modelová společnost působí na českém trhu od roku 2009. Jde o středně velkou společnost, která má hlavní sídlo v Praze. Díky svým zákazníkům se společnosti daří neustále růst a rozvíjet své služby a portfolio. Nabízí bohatou škálu produktů v oblasti zdravé výživy. Pečlivě vybírají produkty, které svým zákazníkům pomáhají v dnešním světě se zdravým stravováním. V současnosti má společnost otevřené 4 prodejny v Praze. Díky tomu se můžou zařadit mezi střední společnosti na území České republice.

Filozofií společnosti je poskytovat komplexní služby v oblasti zdravé výživy, aby si zákazník mohl bez zbytečných pochybností vybrat ze sortimentu. Proto se společnost neustále snaží pracovat na tom, aby svým zákazníkům přinášela co nejlepší služby a potraviny zdravé výživy.

Od svého vzniku společnost stabilně roste. Obrat celé skupiny otevřených obchodů společnosti přesahuje již dostatečnou část financí a společnost je v neustálém zisku. Konkurenční společnosti se předhánějí v poskytování zboží, a proto je nutností se dostat do většího podvědomí zákazníků. V této době je skvělá příležitost, jak si může společnost sama vytvářet větší obrat, a to je e-shop. Je nutností danou databázi a pak i přístup do samotného e-shopu zabezpečit.



Obrázek 10 Organizační struktura společnosti-Vlastní tvorba

Vysvětlení zkratky PP – Procesní podpora

6.1.1 Databáze modelové společnosti

Databáze modelové společnosti je rozdělena do několika větví, jak je zobrazeno na obrázku číslo 10.
Personální oddělní má přístup k datům týkajících se zaměstnanců. Mají oprávnění zakládat nové zaměstnance, měnit jejich osobní údaje i údaje týkající se jejich mzdy nebo pracovního zařazení. V případě ukončení pracovního poměru mají oprávnění daného zaměstnance z evidence vymazat. Jde o údaje, které identifikují jednotlivé zaměstnance v rámci společnosti.

Fakturační oddělení má k dispozici data týkající se faktur, jakožto data o dodavatelích, odběratelích, jednotlivého zboží a v neposlední řade data o nákupních cenách. Všechny data kromě dodavatelů, objednávek a faktur má fakturační oddělení jenom ke čtení. Data o dodavatelích má fakturační oddělení právo zakládat, měnit, a v případě ukončení spolupráce a vyplacení všech plateb dodavatele i smazat. Na základě podkladů může fakturační oddělení měnit údaje na fakturách, a to v případě, že se měnil ceník artiklů a nebyl zadán do systému v dostatečném předstihu oddělením Nákupu.

Dalším oddělením je oddělení Nákupu, toto oddělení spravuje veškeré údaje o artiklech, a proto zaměstnanci tohoto oddělení musí být ve své podstatě výživoví poradci. Je to důležité zejména z toho důvodu, aby skladba zboží byla přehledná a rozmanitá a

obsahovala všechny druhy zboží zdravé výživy. Mají na starosti domlouvaní smluv mezi společností a dodavatelem, kde musí být domluveny obchodní podmínky pro dodání artiklů, nákupní ceny a smluvní pokuty, které můžou vzniknout při nedodržení domluvených obchodních podmínek. Po uzavření smlouvy oddělení Nákupu posílá smlouvu na Fakturační oddělení, kde budou data o dodavateli zavedena do systému. Po vzniku ID dodavatele může oddělení Nákupu založit údaje o smlouvě, kde se uvádí, že určitý dodavatel bude zavážet artikly na všechny nebo jen na konkrétní prodejny. Dále se v tabulce Smlouvy uvádí, jaké artikly bude určitý dodavatel zavážet na prodejny a odkdy je daná smlouva platná. Datum „do“, je při založení nové smlouvy do nekonečna. Toto datum se mění v případě změn původních podmínek nebo při vypovězení smlouvy a ukončení spolupráce. Po vyplnění těchto údajů může oddělení Nákupu zadat údaje o artiklech do systému. Při zadávání údajů o artiklech do systému se musí vyplnit všechna data, jinak řečeno, žádná data nesmí být nulová. Po zadání dat do systému se vygeneruje ID artiklu. Oddělení Nákupu vytváří také objednávky ve vztahu dodavatel a jednotlivé obchody, tedy odběratelé. Sledují jednotlivé prodeje a podle nich vytvářejí objednávky. Následně oddělení Nákupu sleduje nedodávky jednotlivých dodavatelů, čímž vzniká právo na pokuty, které jsou domluveny a odsouhlaseny v obchodních podmínkách.

Oddělení Marketingu zpracovává reklamní kampaně, aby byla společnost stále v povědomí všech potencionálních zákazníků. Data o artiklech, dodavatelích a prodejnách mají k dispozici jenom ke čtení.

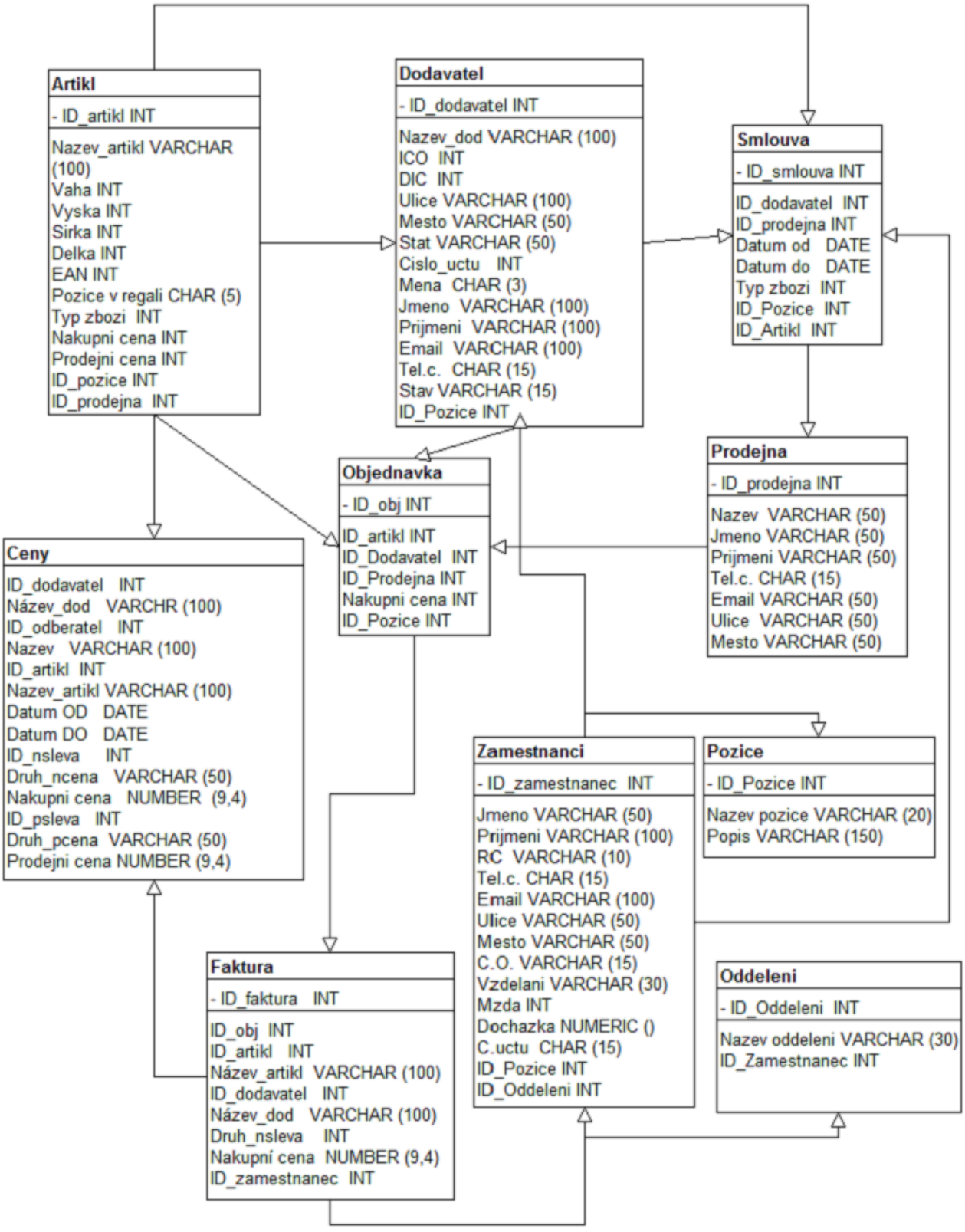
Oddělení IT zpracovává technické požadavky celé společnosti. Má na starosti jak nákup výpočetní techniky, tak správu této techniky a její bezproblémový chod. V případě nefunkčnosti samotného systému se jednotlivá oddělení obrací na IT. Mají na starosti také zálohování systému. Nespravují ale samotná data, je tím myšleno to, že nezpracovávají požadavky na úpravu, změnu nebo samotné smazání dat, ale v případě omylu nebo úmyslného poškození dat toto oddělení dokáže tato data nahradit zálohou.

Oddělení Podpora prodeje je v společnosti podpůrné oddělení, co se týče samotného prodeje v jednotlivých prodejnách. Toto oddělení zabezpečuje podporu a trénink pracovníků na jednotlivých prodejnách v oblasti uspořádaní a prezentace zboží. Také zajišťují školení pro pracovníky na prodejnách. Pracovníci na prodejně musí být vzdělaní

v oboru zdravé výživy, aby byli schopní zákazníkům poradit s výběrem zboží. Spolupracují s prodejnami při tvorbě prodejní plochy. Také připravují a spolupracují s prodejnami při realizaci přestaveb stávajících prodejen. Zajištují přenos informací a myšlenek mezi hlavním sídlem firmy a jednotlivými prodejnami. Toto oddělení má přístup ke všem údajům o artiklech, tyto údaje využívají pro tisk etiket, pro skladbu zboží v regálech. Tyto údaje mají k dispozici jenom ke čtení.

Oddělení Procesní podpory je velice důležitým oddělením. Má k dispozici všechny údaje o artiklech, objednávkách a akcích. Toto oddělení může informace o datech číst, zapisovat, měnit a také mazat ze systému. Toto oddělení pracuje jako podpora pro procesní problémy, které můžou vzniknout při zpracovávaní dat do systému. Zajišťují procesní a informační management. Hlavní úkoly tohoto oddělení můžeme rozdělit na tyto části: na podporu při zakládaní nových artiklů do systému, dále podporu při akčním plánování a podporu pro zakládání slev do systému. S těmito změnami v systému vzniká chybovost a toto oddělení spolupracuje se všemi odděleními napříč celou společností a pomáhá jim těmto chybám předejít nebo je opravit, upravit nebo dokonce smazat.

V modelové společnosti je zabezpečená integrita dat a to tak, že každá tabulka má přiřazen primární klíč, který plní funkci identifikačního klíče. Tabulky jsou mezi sebou propojeny pomocí cizích klíčů. Cizí klíč v jedné tabulce je v podstatě primární klíč v původní tabulce. Například identifikační číslo artiklů je jednoznačný identifikátor artiklu v tabulce Artikl, a v tabulce Objednavka je identifikační číslo artiklu již cizím klíčem. Samotná tabulka Objednavka má svůj primární klíč, a to ID_obj. Datová normalizace u návrhu modelové společnosti je dodržená. Všechny atributy jsou atomické a závislé na primárním klíči. Schématické znázornění modelové společnosti je zobrazeno na obrázku číslo 11.



Obrázek 11 Schématické znázornění modelové společnosti. Vlastní tvorba

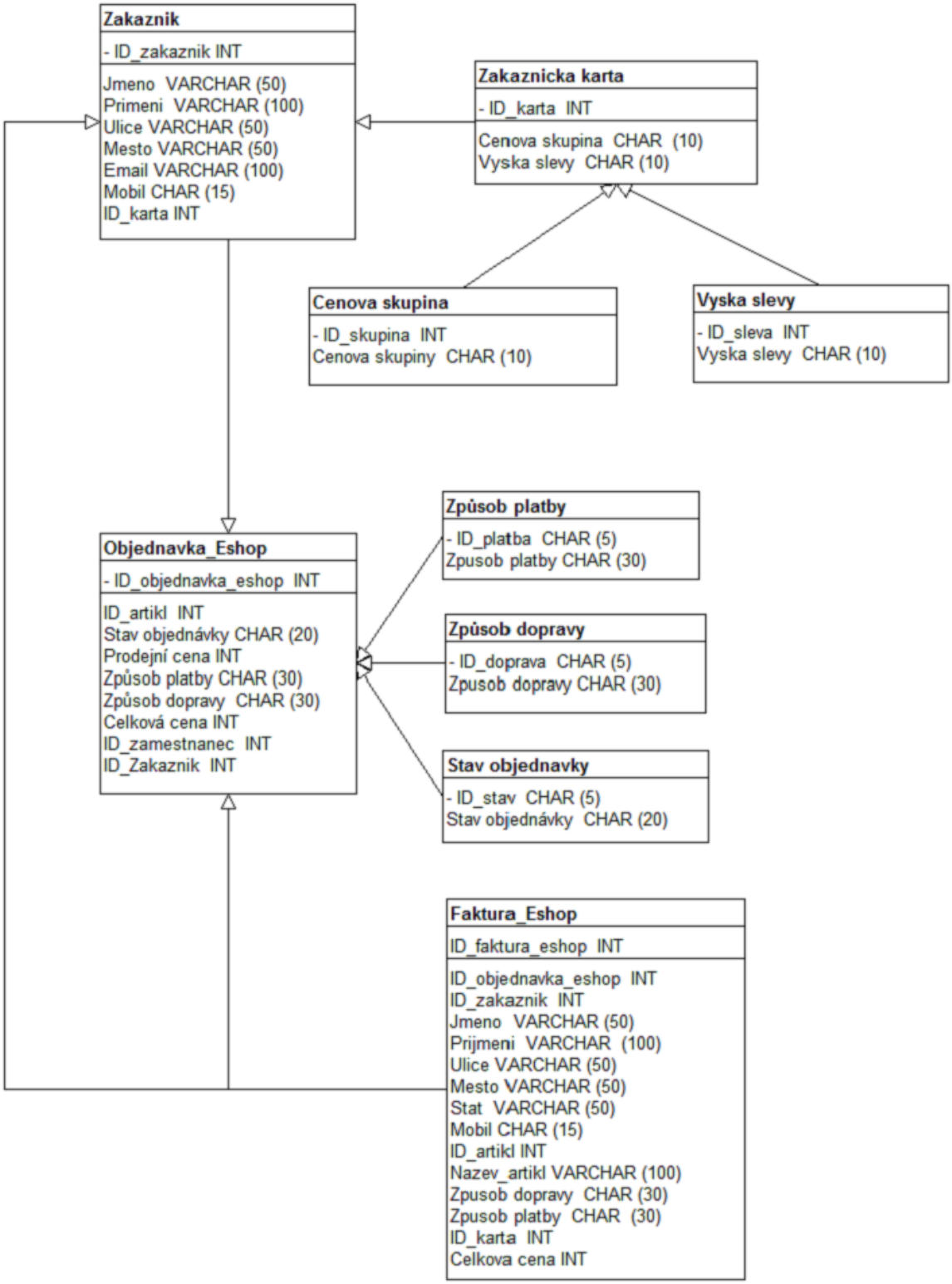
6.1.2 E-shop

Při tvorbě E-shopu je nutno se zamyslet, jak bude vystupovat e-shop v databázi. Proto je v této chvíli potřebné diagram rozšířit. E-shop bude v databázi veden jako další prodejna. E-shop bude zobrazovat data o artiklech, o prodejnách, o objednávkách a samozřejmě osobní údaje zákazníka. E-shop bude v databázi schraňovat důležité informace o zákaznících.

V dnešní době je velice moderní mít zákaznickou kartu a pomocí této karty získávat slevy, výhody nebo dostávat důležité informace do mailů nebo do pošty. Proto vzniká tabulka Zákaznická karta, která bude obsahovat cenovou skupinu a samotnou výšku slevy. Tyto údaje budou zobrazený formou ID, je to z důvodu přehlednosti. V budoucnu je možné, že vzniknou nové druhy slev nebo cenové skupiny a bude jen stačit přidat řádek s novými údaji v samostatných tabulkách pro cenovou skupinu a výšku slevy.

Tabulka objednávka_Eshop vznikne pro přehlednost objednávek. Je nutné oddělit objednávky na prodejny od objednávek pro zákazníky, i když by si chtěl zákazník své objednané zboží vyzvednou na prodejně. Při zpracovávání objednávky oddělení nákupu zjistí, jestli jsou všechny objednané artikly skladem a podle tohoto zjištění změní stav objednávky. Objednávky pro e-shop musí obsahovat také údaje o způsobu platby, dopravy a stavu objednávky.

Tabulka Faktura_Eshop obsahuje údaje, které vznikají postupně při zpracovávání objednávky po vybrání s možností platby a dopravy. Po zpracování samotné objednávky se vystaví faktura na konkrétního zákazníka s jeho osobními údaji, dále s přehledem fakturovaného zboží a samozřejmě celková suma.



Obrázek 12 Schématické znázornění E-shopu - Vlastní tvorba

6.2 Zabezpečení databáze

V databázi musí mít přístup k jednotlivým údajům vícero uživatelů, není proto vhodné, aby každý pracovník měl přehled o osobních údajích všech zaměstnanců, aby znal výšku jejich platu, nebo dokonce mohl něco změnit v údajích o zaměstnancích. Proto je potřebné zabezpečit databázi před neoprávněným přístupem k datům např. tak, že jednotlivým uživatelům se přidělují přístupová práva na jednotlivé objekty nebo skupiny objektů. Tato práva určují povolené operace pro manipulaci s těmito daty.

Zabezpečení se může rozdělit do několika skupin a tyto jednotlivé možnosti také určují, jak je databáze komplikovaná a jak moc je nutné ji zabezpečit. Prvním důležitým krokem je, aby byl uživatel nucen se do systému přihlásit se svým jedinečným uživatelským jménem a heslem. Toto heslo je nutné po určité době měnit, a to podle daných pravidel.

Pohledy se tvoří především pro zjednodušení práce s databází. Pro mnoho uživatelů je efektivnější zobrazit si předem navolený pohled, než pracně propojovat několik tabulek. Tyto pohledy se zobrazí jednoduchým příkazem. Pohled je forma zabezpečení, uživatel si může tento pohled, který je předem navolený, jenom zobrazit, ale již nemá žádné oprávnění nic měnit, upravovat nebo mazat.

Role se vytváří pro složitější práci s databází. Můžeme předpokládat, že jednotlivé oddělení pracují s databází různě. Jedno oddělení musí mít přístup jak pro vkládání dat, tak i k jejich úpravě. Naopak pro jiné oddělení je dostačující tyto údaje jenom vidět a použijí je k dalším výstupům, jako jsou například reporty. Oddělení IT má největší práva, co se týče správy dat v databázi, IT zaměstnanci můžou v případě největší potřeby data vkládat, upravovat, mazat a v nejhorším případě obnovit data pomocí zálohy. Dalším důležitým oddělením je v našem případě oddělení Podpory procesů, toto oddělení má také práva na všechny úkony v databázi, samozřejmě bez práva na zálohování.

Profily jsou soubory limitů na zdroje, jde o obdobu rolí. Každý uživatel má přiřazen právě jeden profil.

6.2.1 Uživatelský účet

V první řadě se před samotnou nutností vzniku rolí pro jednotlivá oddělení, skupiny uživatelů nebo jednotlivce musí vytvořit uživatel a jeho heslo.

Pro tvorbu nového uživatele použijeme CREATE USER

CREATE USER jméno **IDENTIFIED BY** heslo;

Předpokládejme, že modelová společnost přijala do pracovního poměru nového zaměstnance do oddělení Nákupu, jde o pana Dvořáka. Nového uživatele zadáme do systému následovně:

CREATE USER Dvorak **IDENTIFIED BY** PzHB588+ ;

Chceme-li donutit uživatele, aby si změnil své heslo ihned po prvním přihlášení, použijeme nato příznak v této formě:

CREATE USER Dvorak **IDENTIFIED BY** PzHB588+**PASSWORD EXPIRE** ;

Pokud své heslo pan Dvořák zapomene, jistě ocení možnost změny hesla. Na to slouží příkaz:

ALTER USER Dvorak **IDENTIFIED BY** nové_heslo;

Popsaný postup pro tvorbu uživatelského účtu v rámci zaměstnanců ve společnosti platí také pro zavedení údajů o zákazníkovi z e-shopu do databáze. Při vyplnění registrace si zákazník vyplní své údaje včetně povinného emailu a po odeslání svých údajů se zákazníkovi pošle vygenerované heslo, které si zákazník může nebo nemusí změnit. Tato povinnost je určena a pro vyšší bezpečnost se změna hesla vyžaduje. Se získanými údaji se zákazník může přihlásit do e-shopu. Zákazník bude mít přehled o svých osobních údajích, o objednávkách a případných stornech.

Po zaškolení pana Dvořáka může jeho uživatelský účet získat práva, která mu vymezí jeho oprávnění v databázi Oracle.

GRANT SELECT **ON** Artikl, Objednavka **TO** Dvorak;

Předpokládejme, že pan Dvořák postoupil z asistenta nákupčího na nákupčího, a proto se jeho role změní a přidělí se mu vyšší oprávnění.

```
GRANT  SELECT, UPDATE ON Artikl, Objednavka TO Dvorak;
```

Můžeme předpokládat pro prezentaci, že pan Dvořák byl ze své funkce sesazen a jeho účet se musí znova upravit.

```
REVOKE UPDATE ON Artikl, Objednavka TO Dvorak;
```

Poslední možnost účtu pana Dvořáka je smazání, předpokládejme, že s panem Dvořákem byl z organizačních důvodů ukončen pracovní poměr a již nebude nadále svůj účet využívat. Proto můžeme použít příkaz DROP. Tento příkaz se používá také pro smazání role nebo profilu.

```
DROP USER   Dvorak;
```

6.2.2 Pohledy

Použitím pohledu je omezen přístup uživatelů do celé databáze. V každé společnosti jsou zaměstnanci, kteří ke své práci potřebují ucelený výstup z databáze, a to formou již zmíněného pohledu. Je to z důvodu toho, že ne každý zaměstnanec ve společnosti může zasahovat do databáze a stačí mu náhled pro vykonávání jeho práce. Můžeme předpokládat, že v oddělení Nákupu asistent potřebuje výpis všech artiklů, které mají akční cenu.

```
CREATE OR REPLACE VIEW   Akce AS
SELECT ID_artikl, Nazev_artikl, Prodejní cena, ID_psleva, Druh_pceny
FROM Artikl, Ceny;
WHERE Druh_pceny= akční sleva
AND Artikl.ID_ artikl = Ceny.ID_artikl;
```

Přehled zaměstnanců, jejich zařazení do jednotlivých oddělení a samozřejmě jejich zařazení do pozice je velice potřebný pohled v Personálním oddělení. Toto oddělení s tímto pohledem pracuje, například při tvoření akcí pro jednotlivá oddělení nebo pro celou společnost.

```
CREATE OR REPLACE VIEW  Zamestnanci AS
SELECT ID_zamestnanec, Jmeno, Prijmeni, ID_oddeleni, Nazev oddeleni,
ID_pozice, Nazev pozice
FROM Zamestnanci, Pozice, Oddeleni
WHERE Zamestnanci.ID_pozice = Pozice.ID_pozice
AND Zamestnanci.ID_oddeleni = Oddeleni.ID_oddeleni;
```

V oddělení Fakturace ke své práci s fakturami potřebují ucelený pohled na artikly, které dodávají dodavatelé a jsou aktivní. Tím je myšleno, že dodávají zboží a není s nimi pozastavena nebo dokonce zrušená spolupráce. Dále potřebují vidět v pohledu nákupní ceny, dohodnuté slevy, jejich platnost a na jaké prodejně se zboží prodává.

```
CREATE OR REPLACE VIEW  Fakturace AS
SELECT ID_artikl, Nazev_artikl, ID_dodavatel, Název_dod, Stav, ID_nsleva,
Druh_nceny, Nákupní cena, Datum od, Datum do, ID_prodejna, Název_prodejna
FROM Artikl, Dodavatel, Ceny, Smlouva, Prodejna
WHERE Dodavatel.Stav = aktivní
AND Artikl.ID_ artikl = Smlouva.ID_artikl
AND Artikl.ID_ artikl = Ceny.ID_artikl
AND Smlova.ID_prodejna = Prodejna.ID_prodejna;
```

Pohled nezabírá na disku téměř žádné místo, protože neobsahuje žádná data, pouze předpis pro získání dat. Jde o zabezpečení databáze, které omezí zásahy do databáze dat a tímto řešením se může předejít chybovosti a zásahům do struktury dat. V reálné společnosti jsou vytvářeny složitější pohledy pro zobrazení většího množství dat z více tabulek.

6.2.3 Role

Role jsou dalším nezbytným zabezpečením databáze, jde o definování přesných úkonů, které budou moci uživatelé vykonávat. Je nutno uživatele řádně začlenit do předem vytvořených rolí, protože ne každý ve společnosti má stejné pracovní zařazení a pravomoci.

Jako první si můžeme vytvořit roli pro oddělení Nákupu pro sortiment nápoje. Toto oddělení se může skládat ze tří pozic, a to pozice vedoucí, asistent a operátor dat. Vedoucí si může v databázi data zobrazit a v nutných případech může tato data upravit. Asistent vedoucího má oprávnění jenom k zobrazení dat. Veškeré požadavky na změnu, úpravu nebo případné smazání dat deleguje na oddělení Podpory procesů.

Oddělení Nákupu

```
CREATE ROLE Nakup_vedouci;
GRANT SELECT, UPDATE  ON Artikl, Objednavka, Ceny TO Nakup_vedouci;
GRANT SELECT ON Dodavatel, Faktura TO Nakup_vedouci;

CREATE ROLE Nakup_asistent;
GRANT SELECT  ON Artikl, Objednavka, Dodavatel TO Nakup_asistent;
```

Oddělení Fakturace

Dalším oddělením pro tvorbu role bude oddělení Fakturace. Toto oddělení aktivně pracuje s daty, které jsou v tabulkách dodavatelé, objednávky a faktury. Ostatní tabulky jsou pro toto oddělení jenom ke čtení.

```
CREATE ROLE Fakturace;
GRANT SELECT, UPDATE  ON Dodavatel, Objednavka, Objednavka_Eshop,
Faktura, Faktura_Eshop TO Fakturace;
GRANT SELECT ON Artikl, Smlouva, Zakaznik, Zakaznicka_karta,  TO
Nakup_vedouci;
```

Oddělení Podpora procesů

Toto oddělení má v rámci společnosti nejvyšší práva, samozřejmě hned po oddělení IT. Na oddělení Podpory procesů se obracejí všechna oddělení v případě potřeby něco upravit, doplnit nebo také smazat, toto platí mimo Personální oddělení, které své požadavky řeší přímo s IT. Každý incident musí být zapsán na webovém portálu společnosti, aby byly důvody změny dat v databázi vždy dohledatelné.

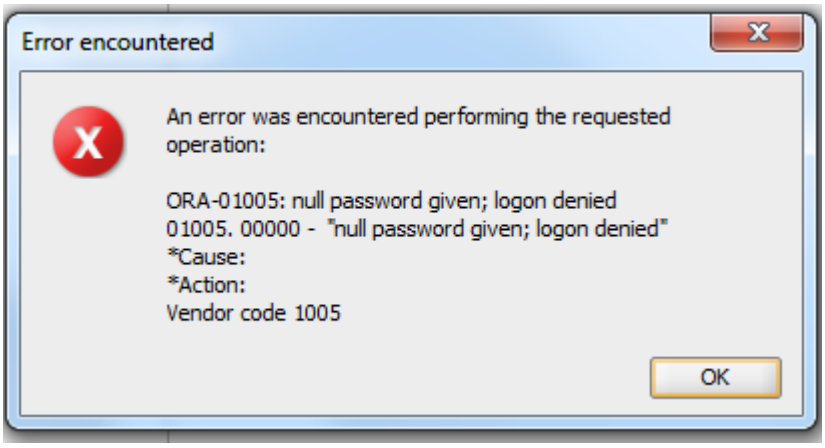
```
CREATE ROLE Proces;
GRANT ALL ON Artikl, Dodavatel, Smlouva, Objednavka, Prodejna, Zakaznik, Zakaznicka_karta, Cenova skupina, Vyska slevy, Objednavka_Eshop, Zpusob platby, Zpusob dopravy, Stav objednávky, Faktura_Eshop TO Proces;
```

Role se můžou považovat za lepší zabezpečení databáze dat, protože jde o jasně nastavená pravidla, která jsou určena pro všechny uživatele, kteří mají přístup do databáze dat. Uživatelé můžou být přiděleni do rolí založených na funkcích vykonávaných v modelové společnosti, a tak vlastnit jen ta práva k tabulkám v databázi, která jsou nevyhnutelná pro vykonávání činnosti jejich funkce. Role můžou být dynamicky přidělované nebo odebrané, aby byla práva omezená kontrolovaným způsobem.

Role můžou být zabezpečeny heslem. Uživatel musí znát heslo, aby byl oprávněn používat danou roli. V dané modelové společnosti by se mohla možnost s heslem využít v případě asistenta v oddělení Nákupu. Vedoucí nákupu má vyšší práva a v jeho nepřítomnosti by mohl asistent po přihlášení do role provést změny v databázi dat. Heslo do role by se nastavilo tímto způsobem.

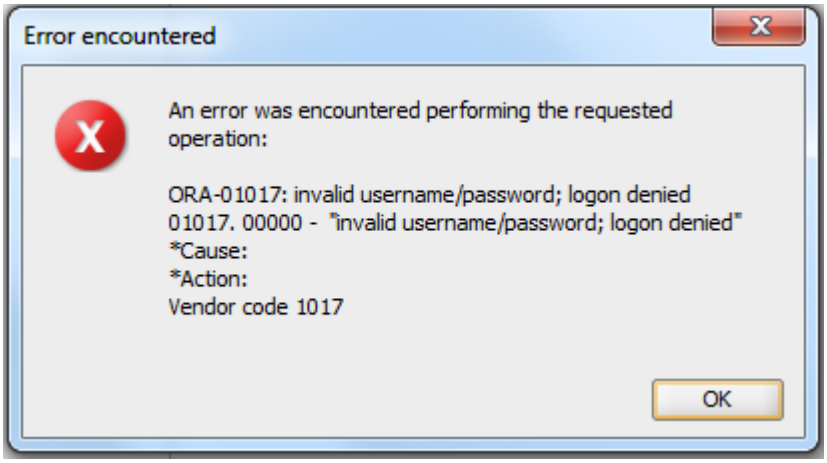
```
CREATE ROLE Nakup_vedouci IDENTIFIED BY PwDgt2569;
```

Pokud uživatel nevyplní heslo, Oracle zobrazí chybovou hlášku. Viz obrázek číslo 13.



Obrázek 13 Oracle chybová hláška - nevyplnené heslo. Vlastní tvorba

Při pokusu o přihlášení a zadání chybného hesla Oracle zobrazí tuto chybovou hlášku.



Obrázek 14 Oracle chybová hláška - chybní heslo. Vlastní tvorba

6.2.4 Profily

Každému uživateli lze přiřadit maximálně jeden profil. Pokud je uživateli přiřazen profil a uživatel již nějaký má, starý profil se nahradí novým. Profil se uplatní až při dalším přihlášení uživatele. Profily nelze přiřazovat rolím.

Pro modelovou společnost můžeme založit profil pro zaměstnance. Tento profil bude určovat, kolikrát se může uživatel přihlásit, jak dlouho může svoje heslo používat a kolikrát se může chybně přihlásit.

CREAT PROFILE zamestnanec **LIMT**
FAILED_LOGIN_AT-TEMPTS 3
PASSWORD_LOCK_TIME .1
PASSWORD_REUSE_MAX UNLIMITED
PASSWORD_LIFE_TIME 90
PASSWORD_REUSE_TIME 60
PASSWORD_VERIFY_FUNCTION VERIFY_FUNCTION
PASSWORD_GRACE_TIME 15;

Tímto příkazem se vytvořil profil zaměstnanec, který definuje, že po třech neúspěšných pokusech o přihlášení k účtu server, Oracle automaticky účet zablokuje. Po jedné hodině se účet, který byl uzamčen kvůli překročení stanoveného počtu neúspěšných pokusů o přihlášení, opět odemkne. Doba platnosti hesla je nastavena na 90 dní a poté ho bude možné ještě změnit následujících 15 dní po ukončení platnosti. Stejné heslo není možné použít 60 dní po změně. Složení hesla je kontrolováno funkcí VERIFY_FUNCTION. Parametr password_reuse_max je nastaven na hodnotu unlimited. Kdybychom tomuto parametru nastavili jinou hodnotu než unlimited, museli bychom nastavit na unlimited parametr password_reuse_time. [9]

Nastavení profilu při vytvoření nového uživatele.

CREATE USER Dvorak **IDENTIFIED BY** PzHB588 **PROFILE** zamestnanec**;**

Podobná logika může být použita i při nastavení uživatele z e-shopu v našem případě zákazníka. V dnešní době se na e-shopu řeší neúspěšné přihlášení a složení hesla.
V případě, že zákazník zapomene své heslo nebo se pokusí přihlásit a vyčerpá si možné pokusy, může požádat o dočasné heslo. Toto heslo bude zákazníkovi zasláno na jeho email, který musí na vyžádání vyplnit a musí se shodovat s emailem z registrace. Jestliže zákazník nevyplní email správně, nebude mu nové heslo zasláno.

6.2.5 Testování zabezpečení

Testování zabezpečení databáze je soubor procesů sloužící pro kontrolu nastavení a kvality zabezpečení databáze. Testování se provádí podle předem stanovených scénářů a za spolupráce se zástupci businessu.
Cílem je dosáhnout požadované kvality z hlediska funkčnosti, spolehlivosti, výkonnosti a použitelnosti. Kontrolu přístupů je nutné otestovat napříč celou společností. Výsledek testů musí být jednoznačný a to takový, že oprávněný uživatel se dostane po přihlášení jenom do té části databáze, která mu má být pod jeho rolí a pohledem zpřístupněna. Tím je myšleno, že například uživatel nákupčí se nemůže svými přihlašovacími údaji podívat na data o zaměstnancích. Po úspěšném otestování a vzájemné spokojenosti můžeme říct, že jsme dospěli k funkčnímu datovému modelu.

7 Závěr

V úvodu předložené práce byl stanoven cíl a tím cílem byl návrh řešení relační databáze a zabezpečení databázové evidence potřebné pro úspěšné fungování e-shopu. Vytyčena byla i omezení, která vzhledem k rozsahu bakalářské práce neumožňují v plném rozsahu poskytnout detailní rozbor modelové společnosti a přehled její databáze. Proto je model v rozsahu, který může poskytnout přehled o databázi a je možné na něm ukázat vztahy a modelovat přístupová práva a zabezpečit tím databázi. Modelová společnost a databáze jsou tvořeny tak, aby vyjadřovaly objekty, které existují v reálném světě.
Teoretická část byla popsána do takové míry, aby obsahovala potřebné informace, které byly následně použité pro vypracování vlastní části této bakalářské práce.

Jak již bylo zmíněno v úvodu, v dnešní době je potřebné kvůli zdravé konkurence-schopnosti a udržení společnosti na trhu neustále šokovat zákazníky něčím novým, a to buď vytvářet nové produkty a slevy, nebo dosáhnout lepší dostupnosti zboží pro zákazníky. Proto se vytvořená databáze pro modelovou společnost rozšířila pro potřebu evidence údajů v e-shopu. Zabezpečení společnosti se dá udělat různými způsoby. Je to dáno hlavně velikostí společnosti, dále tím, jak citlivá data společnost vlastní, jak velké má portfolio a v neposlední řadě také tím, kolik zaměstnanců vlastně bude pracovat s databází. Společnost pracuje s citlivými daty, která se mohou týkat například nákupních cen, obratů nebo také osobních údajů o zaměstnancích. Tato data jsou pro společnost velice cenná a musí se proto řádně zabezpečit. Při úniků takových dat by mohla společnost přijít o spolupráci s dodavateli, o zákazníky a také by se mohla dostat do právních sporů. Proto musí být pro společnost na prvním místě celkové zabezpečení databází, aby nemohlo dojít k úniku dat. Tato data podléhají utajení jak podle občanského zákoníku, tak i podle vnitřní politiky společnosti. Je namístě položit si otázky typu, jak důležitá data se musí zabezpečit, kdo bude mít pravomoc pro práci s celou databází, kdo všechno a do jaké úrovně bude zasahovat do databáze. Po zodpovězení těchto otázek se můžeme pustit do samotné práce se zabezpečením dat v databázi.

Všechna data pro evidenci artiklů, dodavatelů, prodejen, objednávek, zaměstnanců i zákazníků se musela řádně zabezpečit. Modelová společnost je zabezpečena napříč všemi

stupni přístupu do databáze. Je to řazeno podle zařazení zaměstnanců v jednotlivých odděleních. Od nejnižší úrovně až po úroveň, která umožňuje uživatelům pracovat s daty v databázi s různými oprávněními. Nejrozšířenějším nástrojem pro zabezpečení databáze je přístupové heslo, bez kterého není možné se přihlásit a pracovat v databázi. Dále pak pohledy, které zpřístupňují data jenom k náhledu, samotná změna dat není možná. Rolemi byla databáze zabezpečená do samotných úrovní zásahu do databáze. Tyto role se tvořily podle zařazení zaměstnanců v společnosti. Při tvorbě externího zabezpečení se pak velmi často využívají profily, které rozšiřují možnost zabezpečení uživatelských účtů.
V předložené práci byly využity úrovně zabezpečení, které jsou běžně dostupné, bohužel častokrát málo využívané.

Téma zabezpečení databáze je aktuální již více než čtyřicet let, a ještě stále se rozvíjí a přináší nové poznatky v problematice zpracování hromadných dat, a především pak nové metody usnadňující tvorbu databázového zabezpečení.

Seznam použitých zdrojů

1. GROFF, James R. a Paul N. WEINBERG. *SQL: kompletní průvodce*. 2005. Brno: CP Books, 2005. Programování. ISBN 8025103692.
2. VALENTA, M., POKORNÝ, J. *Databázové systémy*. Praha: České vysoké učení technické v Praze, 2013. ISBN 978-80-01-05212-9.
3. PROCHÁZKA, David. *Oracle: průvodce správou, využitím a programováním nad databázovým systémem*. Praha: Grada, 2009. Průvodce (Grada). ISBN 978-80-247-2762-2.
4. LACKO, Ľuboslav. *Mistrovství v SQL Server 2012: [kompletní průvodce databázového experta]*. Brno: Computer Press, 2013. ISBN 978-80-251-3773-4.
5. HERNANDEZ, Michael J. *Návrh databází*. GRADA, 2006. ISBN: 80-247-0900-7
6. LACKO, Ľuboslav. *Databáze: datové sklady, OLAP a dolování dat s příklady v Microsoft SQL Serveru a Oracle*. Brno: Computer Press, 2003. ISBN 8072269690.
7. MORKES, David. *Microsoft SQL Server 2000: tvorba, úprava a správa databází.* Praha: Grada, 2004. Podrobný průvodce začínajícího uživatele. ISBN 8024707322.
8. LONEY, Kevin a Marlene THERIAULT. *Mistrovství v Oracle: kompletní průvodce tvorbou, správou a údržbou databází : [platné pro Oracle 9i, 8i a 8]*. Praha: Computer Press, 2002. ISBN 80-7226-635-7.
9. THERIAULT, Marlene a Aaron NEWMAN. *Bezpečnost v Oracle: metody, nástroje a řešení problémů : [platné pro Oracle 9i, 8i, 8 a 7.x]*. Brno: Computer Press, 2004. Security (CP Books). ISBN 80-7226-979-8.
10. STEPHENS, Ryan K., Ronald R. PLEW a Arie JONES. *Naučte se SQL za 28 dní: [stačí hodina denně]*. Brno: Computer Press, 2010. ISBN 978-80-251-2700-1.