DAIS – Semestrální projekt – Časté chyby

Petr Lukáš

30. března 2017

1 Dotazník

1	**Obecné zásady**		
1.1	Je součástí analýzy hlavička (jméno, příjmení, login, předmět, ročník, cvičící, fakulta, katedra)?	☐	(2.1)
1.2	Je součástí analýzy obsah (seznam kapitol + odkazy na stránky)?	☐	
1.3	Je analýza dobře strukturovaná? Tj. používají se číslované nadpisy?	☐	
1.4	Je analýza přehledná? Tj. používá se konzistentně určitý typ fontu v určité velikosti pro text a nadpisy na různých úrovních?	☐	
1.5	Je analýza srozumitelná? Jsou srozumitelné a čitelné jednotlivé věty, neobsahují množství pravopisných a stylystických chyb?	☐	
1.6	Obsahuje analýza všechny části: specifikace zadání, datový model, stavová analýza, funkční analýza, analýza uživatelského rozhraní?	☐	(1.1 – 1.5)
1.7	Je možné dle analýzy implementovat systém tak, aby splňoval požadavky zadavatele?	☐	
2	**Specifikace zadání**		
2.1	Je součástí popisu motivace, proč systém vzniká?	☐	
2.2	Vyplývají z popisu uživatelské role?	☐	
2.3	Mám v popisu napsáno, co bude do systému vstupovat?	☐	
2.4	Mám v popisu uvedeno, jaké budou výstupy, tj. co od systému očekávám?	☐	
2.5	Mám v popisu uveden stručný popis netriviální funkcionality, který bude dále rozebrán v kapitole Funkční analýza (viz bod 5)?	☐	
3	**Datový model**		
3.1	Obsahuje datový model relační E-R diagram (s cizími klíči)?	☐	
3.2	Obsahuje datový model minimálně 7 tabulek, kde minimálně 4 nejsou číselníky?	☐	
3.3	Je datový model správně navržen?	☐	
3.4	Je součástí datového modelu datový slovník (atribut, datový typ, délka, PK, index, IO, popis)?	☐	
3.5	Ja součástí datového modelu seznam integritních omezení pro jednotlivé atributy (alespoň 3)?	☐	
4	**Stavová analýza**		
4.1	Je součástí analýzy stavová analýza, pokud se v databázi vyskytují entity, které to vyžadují?	☐	
4.2	Je z popisu jasné, jak jednotlivé stavy souvisí s obsahem tabulek?	☐	(3.1)

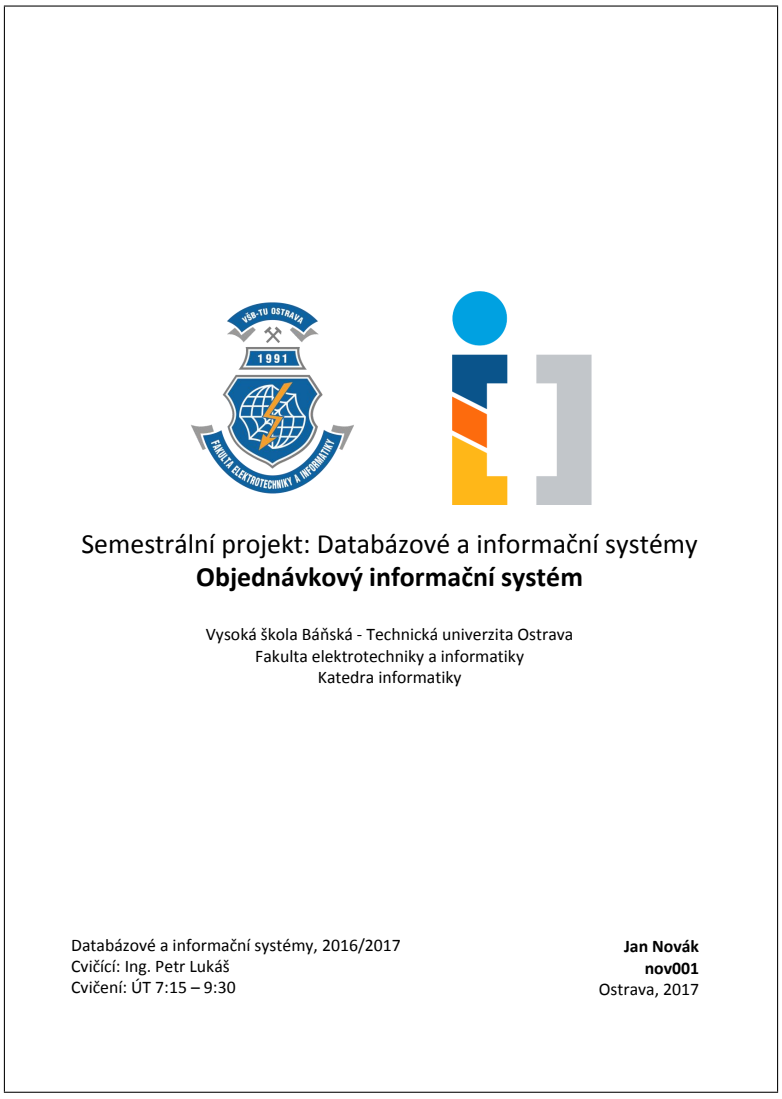
5	**Funkční analýza**		
5.1	Je funkční analýza rozdělena na dvě podkapitoly - seznam funkcí a detailní popis funkcí?	☐	
	Seznam funkcí		
5.2	Jsou v seznamu funkcí uvedeny všechny funkce (včetně CRUD operací)?	☐	
5.3	Je ze seznamu funkcí zjevné, které uživatelské role (viz bod 2.2) budou využívat které funkce?	☐	
5.4	Je ze seznamu funkcí u operací DELETE zjevné, co se má stát se souvisejícími záznamy?	☐	
5.5	Je v seznamu funkcí alespoň 5 netriviálních funkcí, kde alespoň 3 jsou implementovány jako procedura (popř. trigger)?	☐	
5.6	Neobsahuje seznam funkcí další slovní popis (např. zbytečný popis CRUD operací)?	☐	
	Detailní popis funkcí		
5.7	Obsahují netriviální funkce implementované procedurou alespoň 3 body, které nelze dále zredukovat?	☐	(5.2)
5.8	Nepředstavuje některá z funkcí kaskádový DELETE?	☐	(5.1)
5.9	Jsou funkce implementované procedurou popsány v bodech minispecifikací?	☐	(5.3)
5.10	Je každý bod, se kterým souvisí nějaká operace (SELECT, INSERT, UPDATE, DELETE) doplněn o příslušný SQL příkaz?	☐	(5.6)
5.11	Nevyskytuje se v popisu žádná funkce popsáná pouze kódem T-SQL nebo PL/SQL?	☐	(5.7)
5.12	Nevyskytují se v popisu funkcí rysy specifické pro T-SQL nebo PL/SQL?	☐	(5.7)
5.13	U funkcí implementovaných pomocí procedur, jsou součástí popisu vstupní (popř. výstupní) parametry?	☐	(5.8)
5.14	Jsou všechny body popisu funkcí nedělitelné, tj. nevyskytují se v popisu body, které obsahují třeba dvě SQL operace?	☐	(5.9)
5.15	Nevyskytuje se ve funkcích zbytečné použití kurzoru?	☐	(5.11 – 5.14)
5.16	Jsou SQL příkazy zapsány bez syntaktických a sémantických chyb?	☐	(5.15)
5.17	Je jasná vazba mezi seznamem funkcí a detailním popisem funkcí?	☐	(5.16)
5.18	Popisuje každá funkce pouze operace na straně SŘBD, tj. nejde o popis funkce v uživatelském rozhraní?	☐	(5.17)
5.19	Jsou SQL příkazy přehledně odděleny od zbytku textu?	☐	(5.18)
6	**Analýza uživatelského rozhraní**		
6.1	Obsahuje analýza uživatelského rozhraní dvě podkapitoly - struktura menu a návrh formulářů?	☐	
6.2	Je jasná vazba mezi strukturou menu a seznamem funkcí (viz bod 5)?	☐	
6.3	Je jasná vazba mezi ovládacími prvky formuláře a seznamem funkcí (viz bod 5)?	☐	(4.3)
6.4	Jsou součástí návrhu formulářů alespoň 2 netriviální formuláře?	☐	(4.1 – 4.2)

pozn. Čísla v závorce představují kód chyby v dokumentu *Funkční analýza ukázkového projektu* na `dbedu.cs.vsb.cz`.

2 Příklady

1.1 Je součástí analýzy hlavička (jméno, příjmení, login, předmět, ročník, cvičící, fakulta, katedra)?

Častým problémem je, že součástí dokumentu není hlavička. Hlavička není pouze na okrasu, ale usnadňuje orientaci v projektech. Projekt, jehož součástí není hlavička vypadá na první pohled neúplně a obvykle naznačuje, že i obsahová kvalita bude na nízké úrovni. Ukázkovou hlavičku můžeme vidět např. na Obrázku 1. Je doporučeno hlavičku umístit na samostatnou stránku.



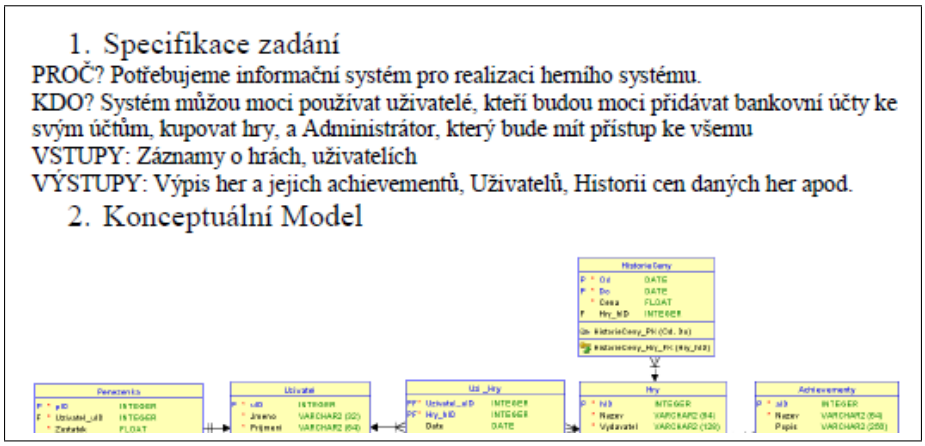
Obrázek 1: Ukázková hlavička projektu

1.2 Je součástí analýzy obsah (seznam kapitol + odkazy na stránky)?

Letmým pohledem na obsah je možé jednoduše zkontrolovat, zda je analýza kompletní. Obsah by měl být automaticky vygenerovaný tak, ať skutečně koresponduje s tím, co se v textu nachází dál. Obsah by se měl také nacházet na samostatné stránce.

1.3 Je analýza dobře strukturovaná? Tj. používají se číslované nadpisy?

Častým problémem analýz je nevhodné nebo nejednoznačné strukturování textu. Ukázku vidíme na Obrázku 2. Autor evidentně nepoužil číslované nadpisy, ale nadpisy udělal formou číslovaných odrážek. Chybí mezery, které by nadpisy oddělily od zbytku textu. Kromě toho je popis zadání až příliš stručný, ale to nesouvisí se strukturováním. Existují i horší případy, kdy se nadpisy neočíslují vůbec.


1. Specifikace zadání

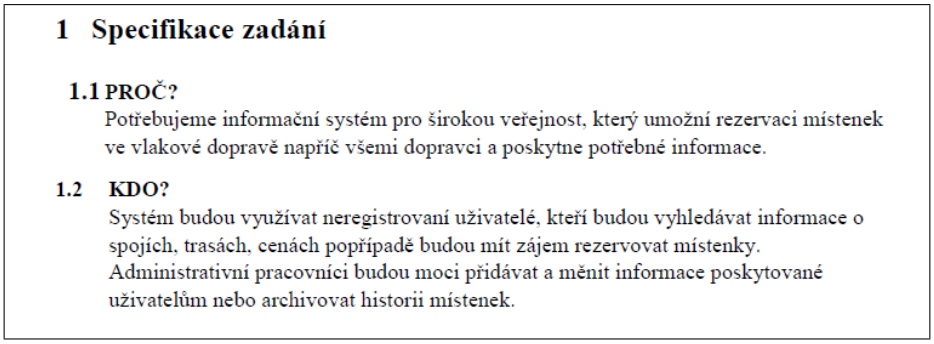
PROČ? Potřebujeme informační systém pro realizaci herního systému.
KDO? Systém můžou moci používat uživatelé, kteří budou moci přidávat bankovní účty ke svým účtům, kupovat hry, a Administrátor, který bude mít přístup ke všemu
VSTUPY: Záznamy o hrách, uživatelích
VÝSTUPY: Výpis her a jejich achievementů, Uživatelů, Historii cen daných her apod.

2. Konceptuální Model

Obrázek 2: Nevhodné strukturování

1.4 Je analýza přehledná? Tj. používá se konzistentně určitý typ fontu v určité velikosti pro text a nadpisy na různých úrovních?

Opět viz Obrázek 2. Takováto analýza se nedá považovat za přehlednou. Jinou ukázku vidíme na Obrázku 3, kde na první pohled uskakuje nadpis 1.1 a používá se pro něj jiný typ písma.


1 Specifikace zadání

1.1 PROČ?

Potřebujeme informační systém pro širokou veřejnost, který umožní rezervaci místenek ve vlakové dopravě napříč všemi dopravci a poskytne potřebné informace.

1.2 KDO?

Systém budou využívat neregistrovaní uživatelé, kteří budou vyhledávat informace o spojích, trasách, cenách popřípadě budou mít zájem rezervovat místenky. Administrativní pracovníci budou moci přidávat a měnit informace poskytované uživatelům nebo archivovat historii místenek.

Obrázek 3: Nepřehledná analýza

1.5 Je analýza srozumitelná? Jsou srozumitelné a čitelné jednotlivé věty, neobsahují množství pravopisných a stylistických chyb?

Tato aplikace bude fungovat tak, že na serveru bude zpracovávat logika hry, která se následně bude zobrazovat na klientovi. Na klientovi tedy bude probíhat veškerá interakce s tímto systémem, tato interakce bude fungovat tak, že uživatel se nejdříve přihlásí na server a následně přihlásí.

Obrázek 4: Nesrozumitelná analýza

Jak vypadají pravopisné chyby snad není nutné komentovat. Problémem ale bývají nesrozumitelné formulace. Na Obrázku 5 je ukázka analýzy, která nedává smysl. V první větě chybí zvratné zájmeno ,,se“ a kromě toho je dost těžké předstvit si, jak se na klientovi zobrazuje nějaká logika. Konec druhé věty také stojí za povšimnutí.

1.6 Obsahuje analýza všechny části: specifikace zadání, datový model, stavová analýza, funkční analýza, analýza uživatelského rozhraní?

Kompletní analýza musí obsahovat všechny uvedené části. Diskutovat lze pouze o části Stavová analýza, která je povinná ,,pouze“ pokud lze u entit stavy definovat. Stavy lze ale definovat téměř vždy.

1.7 Je možné dle analýzy implementovat systém tak, aby splňoval požadavky zadavatele?

Toto je jedna z nejdůležitějších otázek, které si musíme položit. Účelem analýzy je, aby podle ní bylo možné systém naimplementovat. Nejjednodušší kontrola, zda máte analýzu správně je ta, že si jí prohodíte např. s kolegou a pokusíte se ji částečně implementovat.

2.1 Je součástí popisu motivace, proč systém vzniká?

Součástí popisu by měl být alespoň nějaký fiktivní důvod, který vysvětluje, proč je vůbec potřeba informační systém vytvořit. Důvodem obvykle je, že určité firmě XYZ přestanou stačit jednoduché metody evidence jako např. používání Excelovských tabulek.

2.2 Vyplývají z popisu uživatelské role?

Se systémem nejčastěji pracuje více typů uživatelů jako např. účetní, dělník, ředitel, administrátor apod. Specifikace zadání musí alespoň stručně naznačit, kteří uživatelé budou se systémem pracovat a jakou funkcionalitu budou využívat. Z toho potom vyplývá, jaké budeme v systému definovat role. Role a to, jaké části systému využívají, je možné zachytit pomocí Use-Case diagramu, který ale v tomto projektu není povinný. Na role se ale budeme odkazovat v části Funkční analýza. Je potřeba dát si pozor, aby to, co nadefinujeme v úvodu, korespondovalo s tím, co budeme používat ve funkční analýze. Tzn. pokud z úvodu vyplývá, že ze systémem pracuje např. skladník a účetní a potom ve funkční analýze řekneme, že určitou funkci bude používat ředitel, je někde něco špatně.

2.3 Mám v popisu napsáno, co bude do systému vstupovat?

Měli bychom stručně naznačit, že do systému budeme vkládat informace např. o studentech, zaměstnancích a katedrách.

2.4 Mám v popisu uvedeno, jaké budou výstupy, tj. co od systému očekávám?

Měli bychom stručně naznačit, co od systému očekáváme. Detailněji to potom rozebereme v části Funkční analzýa. Tzn. např. můžeme napsat, že systém bude poskytovat přehledy o měsíčních uzávěrkách objednávek. Funkční analýza pak bude obsahovat např. funkci ,,Uzávěrky“, která bude detailně popsána minispecifikací.

Tato aplikace bude fungovat tak, že na serveru bude zpracovávat logika hry, která se následně bude zobrazovat na klientovi. Na klientovi tedy bude probíhat veškerá interakce s tímto systémem, tato interakce bude fungovat tak, že uživatel se nejdříve přihlásí na server a následně přihlásí.

Obrázek 5: Nesrozumitelná analýza

2.5 Mám v popisu uveden stručný popis netriviální funkcionality, který bude dále rozebrán v kapitole Funkční analýza (viz bod 5)?

Netriviální funkce představují obvykle hromadné transakce. Tzn. najednou manipulujeme s větším množstvím záznamů. Např. hromadně posíláme notifikační e-mail uživatelům, kteří už se dlouho nepřihlásili do systému. Detailněji funkce rozebíráme až v části Funkční analýza, nicméně už v úvodu by se stručný nástin toho, jaké funkce bude systém poskytovat, měl objevit.

3.1 Obsahuje datový model relační E-R diagram (s cizími klíči)?

Ve cvičeních DAIS pracujeme pouze s relačním datovým modelem. Na projektu tedy vyžadujeme pouze E-R diagram relačního datového modelu, tj. ten, který obsahuje cizí klíče. Pro vytvoření takového E-R diagramu doporučujeme nástroj Oracle SQL Developer Data Modeler nebo přímo Microsoft SQL Management Studio. Konceptuální model v projekdu DAIS je zbytečný.

3.2 Obsahuje datový model minimálně 7 tabulek, kde minimálně 4 nejsou číselníky?

Navrhnout model s minimálním počtem tabulek obvykle nebývá problém. Na minimálním počtu tabulek je ale často obžížné vymyslet netriviální funkce. Doporučeno je tedy spíše větší množství tabulek, např. 10. Číselníkem se rozumí tabulka, ve které předpokládáme spíše menší počet záznamů, přičemž obsah tabulky se v průběhu času příliš nemění. Číselník typicky obsahuje pouze atributy jako *id* a *název*. Jde např. o seznam škol, seznam skladů, seznam typů výrobků apod. Naopak např. tabulka objednávek číselník není.

3.3 Je datový model správně navržen?

Správnému návrhu datového modelu se věnuje předmět UDBS. Musíme si zkontrolovat, zda máme správně navržené tabulky (relace) a vztahy mezi nimi. U vztahů si musíme zkontrolovat kardinalitu (1:1, 1:N, M:N) a povinnost členství.

5.1 Je funkční analýza rozdělena na dvě podkapitoly – seznam funkcí a detailní popis funkcí?

Kapitola Funkční analýza bude jednoznačně rozělena na dvě podkapitoly: Seznam funkcí a Detailní popis funkcí. Nebude se tedy jednat o jednu kapitolu, kde bude seznam kombinovaný s popisem.

5.2 Jsou v seznamu funkcí uvedeny všechny funkce (včetně CRUD operací)?

Seznam funkcí musí obsahovat kompletní seznam všech funkcí v systému a to včetně CRUD (create, read, update, delete) operací. Funkce budou seskupené podle toho, s jakými tabulkami pracují. Funkce samozřejmě může využívat i více tabulek, pak ji začleníme do samostatné skupiny, nebo ji dle uvážení přidáme k některé z tabulek. Zjednodušeně řečeno, pokud budu mít v systému 10 tabulek, pak tento seznam bude obsahovat $10 \times 4 + 5$ funkcí. Tzn. pro každou tabulku 4 CRUD operace + 5 netriviálních funkcí. Je doporučeno netriviální funkce v seznamu zvýraznit (např. tučně), usnadňuje to pak kontrolu.

5.3 Je ze seznamu funkcí zjevné, které uživatelské role (viz bod 2.2) budou využívat které funkce?

Pro každou funkci v seznamu musí být jednoznačně vidět, jaká uživatelská role může tuto funkci využívat. Role můžeme uvést na úrovni tabulky (tzn. neuvádíme pak zvlášť pro každou CRUD operaci nebo netriviální funkci), výjimky (např. určitou funkci může navíc používat určitá role) dopíšeme k jednotlivým funkcím. Příklad vidíme na Obrázku 6.

2. Zaměstnanci
Role: *ředitel, administrátor, zaměstnanec*

2.1 Přidání zaměstnance
2.2 Zobrazení zaměstnance
2.3 Úprava zaměstnance (pouze role: *ředitel*, *zaměstnanec* pouze svůj záznam)
2.4 Odstranění zaměstnance (pouze role *administrátor*)

Obrázek 6: Příklad zápisu rolí v seznamu funkcí

5.4 Je ze seznamu funkcí u operací DELETE zjevné, co se má stát se souvisejícími záznamy?

Mějme v systému např. tabulky *Zaměstnanec* a *Nákup* ve vztahu 1:N. U operací DELETE (tzn. např. funkce „Odstranění zaměstnance“) musíme zvážit, jak ošetříme situaci, kdy zaměstnanec již provedl nějaké nákupy. Většinou nastane jedna z možností:

1. **Kaskádové mazání** – Vazbě mezi tabulkami je možné nastatit tzv. kaskádové mazání (`ALTER TABLE Zamestnanec ADD FOREIGN KEY (...) REFERENCES (...) ON DELETE CASCADE`). Je to standardní funkcionalita každého pokročilého relačního SŘBD. Kaskádové mazání znamená, že odstraníme-li zaměstnance, SŘBD automaticky odstraní i všechny související nákupy. Pokud nám tato možnost vyhovuje, napíšeme do analýzy např.: „Při odstranění záznamu o zaměstnanci se pomocí kaskádového mazání odstraní také související záznamy o nákupu.“
2. **Zamezení smazání** – Pokud to považujeme za vhodné, můžeme v analýze napst, že: „Jestliže zaměstnanec již provedl nákup, nebude tato funkce dostupná.“
3. **Nastavení příznaku o neaktivním záznamu** – V reálných systémech běžní uživatelé operaci DELETE obvykle neprovádějí. Systém musí být „blbuvzdorný“. Nechtěným smazáním můžeme způsobit nenávratnou škodu. Proto se obvykle mazání nahrazuje nastavením nějakého atributu typu BIT (např. `Zamestnanec.aktivni`) na určitou hodnotu, např. 0. Tím určíme, že záznam už se nikde nebude zobrazovat (musíme podle toho samozřejmě patřičně upravit SELECT dotazy nad příslušnou tabulkou).

5.5 Je v seznamu funkcí alespoň 5 netriviálních funkcí, kde alespoň 3 jsou implementovány jako procedura (popř. trigger)?

Ideální stav je, když projekt obsahuje 3 netriviální funkce řešené formou uložené procedury a 2 netriviální funkce řešení formou komplexního dotazu.

5.6 Neobsahuje seznam funkcí další slovní popis (např. zbytečný popis CRUD operací)?

5.7 Obsahují netriviální funkce implementované procedurou alespoň 3 body, které nelze dále zredukovat?

Funkce vytvoří zaměstnance s daným jménem.
vstup: *$jmeno*, *$prijmeni*

1. Do tabulky vložíme záznam o zaměstnanci pomocí následujícího SQL příkazu:
   `INSERT INTO Zamestnanec (jmeno, prijmeni) VALUES ($jmeno, $prijmeni)`

Obrázek 7: Příklad triviální funkce

Za netriviální funkci se bude považovat procedura (případně trigger), která **obsahuje alespoň 3 body (příkazy), které už není dále možno redukovat**. Příklad triviální funkce vidíme na Obrázku 7.

5.8 Nepředstavuje některá z funkcí kaskádový DELETE?

Jak již bylo uvedeno v bodě 5.4, relační SŘBD jako MS SQL Server nebo Oracle Database nabízí možnost kaskádového mazání. Mějme např. opět tabulky *Zaměstnanec* a *Nákup*, mezi kterými je vazba 1:N. Někoho by mohlo napadnout vytvořit proceduru s parametrem *idZaměstnance*, která nejprve odstraní všechny nákupy zaměstnance a poté zaměstnance samého. Takováto procedura ale nebude na projektu akceptována, protože jde jen o nahrazení kaskádového delete, který můžeme jednoduše aktivovat jako vlastnost vazby mezi zmíněnými tabulkami.

5.9 Jsou funkce implementované procedurou popsány v bodech minispecifikací?

Na minispecifikaci se v této analýze můžeme dívat jako na další procedurální jazyk po T-SQL a PL/SQL. Každý bod popisu téměž vždy povede k jednomu příkazu v T-SQL nebo PL/SQL. Z popisu bodu by mělo být naprosto jasné, o jaký příkaz půjde. Ukázku špatného popisu vidíme na Obrázku 8. Z popisu není zřejmé, co se myslí ,,načtením“. Pokud chceme např. uložit jméno a příjmení zaměstnance do určitých proměnných, pak správny popis bude vypadat např.: ,,Do proměnných *$jmeno* a *$prijmeni* uložíme výsledek dotazu `SELECT jmeno, prijmeni FROM ....`“

1. Procedura načte zaměstnance pomocí příkazu:
   `SELECT * FROM Zamestnanec WHERE idZamestnance = $idZam`

Obrázek 8: Příklad nejasného bodu popisu

5.10 Je každý bod, se kterým souvisí nějaká operace (SELECT, INSERT, UPDATE, DELETE) doplněn o příslušný SQL příkaz?

V naší analýze je požadováno, abychom všude, kde je to možné, psali standardní SQL příkazy – SELECT, INSERT, UPDATE a DELETE (tedy to, co jsme se naučili v UDBS). Příklad, kde toto není splněno, vidíme na Obrázku 9. S prvním i druhým bodem evidentně souvisí SQL příkazy SELECT, které v popisu chybí.

5.11 Nevyskytuje se v popisu žádná funkce popsáná pouze kódem T-SQL nebo PL/SQL?

Toto snad není potřeba popisovat. Pokud popíšeme funkci pouze kódem T–SQL nebo PL/SQL, tzn. zkopírujeme hotovou proceduru jako popis, je to špatně.

1. Do proměnné *$jmeno* uložíme jméno zaměstnance s dle proměnné *$idZam*.
2. Do proměnné *$idNakupu* uložíme poslední ID nákupu.

Obrázek 9: Chybějící SQL příkazy

5.12 Nevyskytují se v popisu funkcí rysy specifické pro T-SQL nebo PL/SQL?

Anaýza musí být napsána tak, aby neobsahovala rysy specifické pro jazyk T-SQL nebo PL/SQL. Idea je taková, že teprve poté, co je analýza hotová, se rozhodneme, který systém budeme používat. Příklad této chyby vidíme na Obrázku 10. Chyby jsou tam hned dvě. `TOP 1` nám nebude fungovat na Oracle Database, jde o specifickou funkcionalitu MS SQL Serveru. Abychom se této chybě vyhli, můžeme jednoduše napsat: „Do proměnné *$idNakupu* uložíme první výsledek následujícího dotazu ...“ Další chybou je použití funkce `GetDate()`, která je opět specifická pro MS SQL Server. Tento problém můžeme vyřešit tak, že místo `GetDate()` napíšeme např. *aktuální_datum*. Tak bude každému jasné, co chceme v analýze říci a vyhneme se specifickým rysům konkrétních jazyků.

1. Do proměnné *$idNakupu* vložíme ID posledního nákupu, který byl vytvořen nejpozději před 30-ti dny, pomocí následujícího příkazu:

   ```
   SELECT TOP 1 idNakup FROM Nakup WHERE datum <= GetDate() - 30
   ```

Obrázek 10: Chybějící SQL příkazy

5.13 U funkcí implementovaných pomocí procedur, jsou součástí popisu vstupní (popř. výstupní) parametry?

5.14 Jsou všechny body popisu funkcí nedělitelné, tj. nevyskytují se v popisu body, které obsahují třeba dvě SQL operace?

5.15 Nevyskytuje se ve funkcích zbytečné použití kurzoru?

5.16 Jsou SQL příkazy zapsány bez syntaktických a sémantických chyb?

5.17 Je jasná vazba mezi seznamem funkcí a detailním popisem funkcí?

5.18 Popisuje každá funkce pouze operace na straně SŘBD, tj. nejde o popis funkce v uživatelském rozhraní?

5.19 Jsou SQL příkazy přehledně odděleny od zbytku textu?