

# *Analýza databázového modelu a návrh nového modelu pro aplikaci společnosti Investiční aukce, s.r.o.*

## Zpráva z řešení projektu

**Řešitelé za FIT VUT: Ing. Vladimír Bartík, Ph.D., Ing. Tomáš Volf**

Výsledky projektu zahrnují především analýzu stávajícího databázového modelu firmy Investiční aukce, odstranění nalezených nedostatků, návrh plánu migrace z původního do nového databázového modelu a úpravu stávajícího informačního systému tak, aby pracoval s upravenou databází. V rámci projektu byl použit databázový systém PostgreSQL.

Projekt by rozdělen do tří fází:

V první fázi proběhla analýza databázového modelu s cílem nalezení designových nedostatků současného modelu, odstranění nedostatků proti normálním formám, odstranění redundantních a nepoužívaných entit a atributů, byly optimalizovány datové typy atributů a zapracována podpora pro souběžně vyvíjené funkcionality systému.

Ve druhé fázi proběhla příprava plánu na inkrementální migraci dat ze starého na nový model, byl vytvořen datový konverzní nástroj a provedeny úpravy codebase systému pro celky, které byly vyhodnoceny jako klíčové.

Ve třetí fázi byly dokončeny úpravy designových a formálních nedostatků databázového modelu. Dále byly dokončeny úpravy pro zbývající celky systému.

## Zpráva k 1. fázi projektu

**Cíl:** V první fázi proběhne analýza databázového modelu, s cílem nalezení designových nedostatků současného modelu, odstraní nedostatky proti normálním formám, odstraní redundantní a nepoužívané entity a atributy, optimalizuje datové typy atributů a zapracuje podporu souběžně vyvíjených funkcionalit systému.

**Výstup:** Analýza stávajícího databázového modelu, popis jednotlivých tabulek a atributů; odstranění nepotřebných atributů. Stanovení celků pro inkrementální optimalizaci databáze.

### Definice používaných zkratk

- Investiční aukce, s.r.o. dále jen jako “iA”

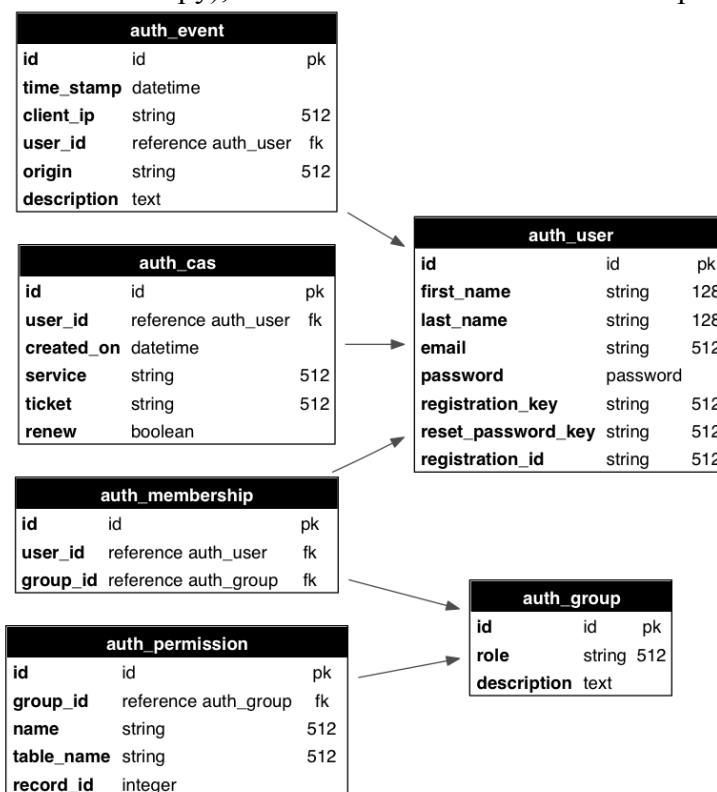
## Analýza stávajícího databázového modelu

Na základě poskytnutého diagramu logického schématu databáze a konzultací se zástupci iA jsme analyzovali nedostatky v návrhu databáze. Zde je uveden jejich stručný výčet a nástin jejich možného řešení v následujících fázích projektu:

- Nevyužívané atributy tabulek – analýzou zdrojového kódu aplikace firmy iA bylo nalezeno množství atributů, které jsou nadbytečné, protože nejsou aplikací vůbec využívány. Budou proto z odpovídajících tabulek odstraněny.
- Ojediněle se v databázi nachází příznakové atributy (datového typu boolean), na základě jejichž hodnoty se vybírá buď hodnota atributu A nebo hodnota atributu B. Pokud je jeden z atributů, např. A nevyužívaný, pak je zbytečný i tento příznakový atribut.
- Redundantní atributy v tabulkách – analýzou schématu databáze bylo nalezeno několik výskytů redundantních atributů, tj. atributů obsahujících informace, které jsou již obsaženy jinde v databázi. Takové atributy budou v následující etapě také odstraněny z příslušných tabulek.
- Nevhodné datové typy – v několika případech je možné nahradit řetězcový datový typ sloupce číselným, protože obsahuje pouze číselné hodnoty. Také je v několika případech možné pro zvýšení efektivity snížit maximální rozsah u některých atributů typu řetězec.
- Nejednotné pojmenování, nejasné nebo zavádějící pojmenování - tabulky a atributy trpí nejednotným pojmenováním (viz příklady níže), některé atributy jsou pojmenovány nejasně nebo zavádějící.
  - Tabulky - např. některé tabulky mají prefix `email_*`, jiné zase `mail_*`, přitom se vztahují ke stejné oblasti.
  - Atributy - např. v tabulce `auction_filter` jsou některé atributy značeny se sufixem `*_min` a `*_max`, jiné atributy pro podobné využití pro změnu užívají prefix `min_*` a `max_*`.
- Porušení zásad návrhu relačních tabulek (normalizace) - ojediněle se v databázi (např. v tabulce `client_fee_invoice`) vyskytuje porušení 3. normální formy (tranzitivní funkční závislost) způsobující redundanci. Atributy způsobující porušení 3NF budou odstraněny, resp. nahrazeny odkazem na tabulku obsahující tyto atributy.
- Chybějící integritní omezení typu cizí klíč - v několika případech chybí ve výchozí databázi odkazování mezi tabulkami s využitím cizích klíčů. Např. záznamy tabulky `mail_queue` jsou tvořeny s využitím šablon v tabulce `email_template`. Pro lepší zajištění konzistence databáze je vhodné vytvořit cizí klíč mezi těmito tabulkami, který zajistí, že bude vždy z databáze patrné, která šablona byla použita k vygenerování daného emailu.
- Nutnost oddělení tabulky Faktura (Invoice) od tabulky všech dokumentů (`doc_auction`) – doposud bylo vše ukládáno v rámci jedné tabulky. Protože však existuje několik atributů, které jsou specifické pro faktury, bude pro ně vytvořena nová tabulka a původní tabulka `doc_auction` bude obsahovat všechny ostatní typy dokumentů (typicky smlouvy atd.)
- Stávající databáze neumožňuje pro jednu fakturu vytvořit více plateb, tj. neumožňuje rozložení placení faktury do více plateb. K tomuto účelu bude potřeba oproti stávající databázi vytvořit novou tabulku Platba (Payment), kam bude pro jednu fakturu možné uložit libovolné množství plateb.
- Faktury vztahující se k více aukcím - v nové databázi bude také potřeba zajistit, aby jednu fakturu bylo možné přiřadit k většímu množství aukcí současně. To bude

zajištěno vazební tabulkou reprezentující vztah M:M mezi tabulkami reprezentujícími entitní množiny auction a invoice.

- Efektivnější výpočet celkové proinvestované částky - pro účely tohoto výpočtu bude vytvořena nová tabulka obsahující celkovou částku k danému datu. Nebude tak nutné počítat tuto hodnotu ze všech záznamů databáze, ale bude možné využít hodnotu v této tabulce a přičíst pouze hodnoty nově vzniklých aukcí. Analogicky totéž bude možné provést i u výpočtu průměrného výnosu.
- V databázi se vyhledává dle specifických atributů. Zavedením indexů nad těmito atributy se zrychlí získání vyhledávaných dat tím spíše, čím rozsáhlejších tabulek se indexování bude týkat. Indexy budou v databázi zavedeny tak, aby především vedly k urychlení
- Tabulky auth\_\* jsou specifickými tabulkami pro framework web2py (viz následující schéma web2py), avšak iA do těchto tabulek přidává svoje vlastní atributy.



To není vhodné, protože názvy přidávaných atributů jsou potenciálně rizikové vzhledem k možnému budoucímu rozvoji web2py, konkrétně k přidání potenciálně konfliktních názvů atributů ze strany web2py.

Dalším ohledem je riziko bezpečnostní, kdy framework web2py může získávat celé řádky tabulek, čímž získá přístup ke specifickým atributům firmy iA, ve kterých mohou být uložena citlivá data (např. číslo občanského průkazu a rodné číslo uživatele). Druhou možností jsou potenciální zranitelnosti ve frameworku web2py, kdy útočník např. výběrem celých řádků získá přístup k těmto citlivým údajům.

## Popis tabulek a atributů

Výchozí databáze společnosti iA obsahovala v době analýzy 41 tabulek (některé další mohou vzniknout pokračujícím vývojem v režii iA), které byly pouze částečně dokumentovány. Díky

několika konzultacím jsme mohli vytvořit podrobný popis jednotlivých tabulek, kdy u každé tabulky je uveden kompletní seznam všech atributů, včetně jejich datových typů a případně dalších integritních omezení vztahujících se k dané tabulce.

Zde uvádíme příklad výsledného popisu jedné z tabulek, včetně návrhů na úpravu některých nedostatků. Příklad ukazuje tabulku `bank_account` určenou pro ukládání informací o bankovních účtech. Ve stejném duchu byl vytvořen i popis ostatních tabulek databáze společnosti iA.

## Tabulka „`bank_account`“

- bankovní účty subjektů
- Atributy
  - `name` (string/64) – název účtu
  - `prefix_num` (string/6, null) – číslo bankovního účtu – prefix
  - `account_num` (string/10, null) – číslo bankovního účtu
  - `bank_num` (string/4, null) – číslo bankovního účtu – kód banky
  - `iban` (string/34, null) – číslo bankovního účtu ve formátu IBAN
  - `swift` (string/11, null) – mezinárodní kód banky, tzv. swift
  - `currency` (string/3, NOT null) – měna účtu
  - `primary_account` (bool, false) – příznak hlavního účtu
  - `account_type` (string/12, NOT null, default = „common“) – typ účtu
    - **definovat výčet**
    - **COLLECT:** sběrný účet pro iA (na něj posílají odběratelé, když hradí faktury)
    - **COMMON:** běžný účet klienta / investora
    - **FROMINV:** příjmový účet pro iA (na něj chodí financování investorů a přeposílá se klientům)
    - **OPERATING:** provozní účet pro iA (firemní účty pro příjmy a výdaje; chodí na něj poplatky za zprostředkování faktury)
  - `hidden` (bool, false) – příznak vymazání
  - `currency_default` (bool, false) – příznak pro iA účty, že je pro tu měnu hlavní (defaultní)
  - `group_id` – navázání na subjekt
    - **Je to potřeba? To samé dělá `client_bank_account_include`**
- **Normalizace**
  - Swift je pro banky stejný – převést do tabulky se záznamy bank
    - současně i přejmenovat na `swift_bic`
- **iA app: neumožnit změnu čísla účtu! (jsou navázané cizí klíče - aplikace případně umře na porušení integritních omezení)**

## Odstranění nepotřebných atributů

V rámci analýzy současného databázového modelu bylo také kontrolováno využívání jednotlivých atributů ve stávajícím systému iA.

Byly identifikovány následující atributy, které nejsou používány nebo jsou již zbytečné, a mohou tedy být odstraněny:

- `auction.max_price`, `auction.min_price`
- `auction.when*` - celkem 9 atributů

- auction.createdbycustomer
- auction.sidenotes
- auction.status - bude nahrazen FK do tabulky auction\_status
- auction\_filter.due\_date\_m\* - pro dané použití nedává smysl
- auction\_filter.end\_date\_m\* - pro dané použití nedává smysl
- auction\_filter.status - pro dané použití nedává smysl
- auction\_filter.edi
- auth\_group.file\_num - iA již dříve rozdělilo na auth\_group.file\_number + auth\_group.file\_number\_institution + auth\_group.file\_number\_city
- auth\_group.bank\_account
- auth\_group.ia\*\_account - iA již dříve nahradilo za auth\_group.\*\_account\_id
- auth\_group.fee\_based\_on\_face\_value - atribut sloužil na rozhodování, na základě kterého atributu se bude počítat odměna. Vzhledem k tomu, že jeden z atributů (bid.amount) je již nepoužívaný, nemá smysl tento příznakový parametr nadále udržovat.
- auth\_user.lang
- auth\_user.notify\_auction\_new
- bid.amount
- client\_fee\_invoice.ia\* - celkem 10 atributů; bude nahrazeno cizím klíčem do tabulky ia\_info
- client\_fee\_invoice.client\* - celkem 13 atributů; bude nahrazeno cizím klíčem do tabulky auth\_group (nebo nové tabulky z ní vytvořené)
- doc\_auction.debtor\_ico - redundantní atribut
- ia\_info.file\_num
- investor\_questionnaire.changed
- offer.amount
- rating.provider
- rating.inserted
- reg\_request.note
- reg\_request.created - bude duplicitní vzhledem k zamýšlené historii stavů
- reg\_request.createdbycustomer
- settings.sandbox\_auction\_limit

V rámci optimalizací prováděných v dalších fázích projektu bude zachování (příp. odstranění) těchto nepoužívaných nebo zbytečných atributů konzultováno s firmou iA s ohledem na jejich potencionální možné budoucí využití.

Další atributy budou odstraněny nebo přesunuty v dalších fázích projektu při normalizaci a optimalizaci databáze.

Atributy, které zatím nejsou používané, ale odstraněny být nemají:

- settings.sandbox\_file\_limit

## Stanovení celků pro inkrementální optimalizaci DB

Pro inkrementální optimalizaci databáze jsme stanovili následující celky, které obsahují tabulky vztahující se k:

- Aukcím - tabulky auction, auction\_status, industry, bid, offer
- Fakturám za aukce - client\_fee\_invoice, invoice\_item, ia\_info
- Souborům k aukcím - doc\_auction, source\_file\_auction, file\_auction, doc\_include

- Sandboxu aukcí - auction\_sandbox, doc\_sandbox, file\_sandbox
- Autentizační logice pro subjekty - auth\_group, auth\_permission, auth\_membership, rating, client\_debtor
- Autentizační logice pro uživatele - auth\_user, reg\_request, auth\_event, auth\_cas
- Odesílání e-mailů - email, mail\_queue, email\_attachment, mail\_template
- Bankovním účtům - bank\_account, client\_bank\_account\_include, fio\_token
- Historii kurzů a daní - exchange\_rate, tax
- Ostatním výše nezařazeným tabulkám - auction\_filter, news

Poznámka: Tento seznam celků nemá za cíl určit celky striktně a neměnně. Pokud bude v průběhu fáze normalizace a optimalizace zjištěno, že by daný celek znamenal příliš rozsáhlé zásahy a změny, může být některý z celků rozdělen na více menších celků. Podobně, pokud bude dávat smysl některou z tabulek z hlediska normalizace a optimalizace přeradit do jiného celku, může tak být v průběhu dalších fází projektu učiněno.

# Zpráva k 2. fázi projektu

**Cíl:** Ve druhé fázi proběhne příprava plánu na jednorázovou nebo inkrementální migraci ze starého na nový model, datový konverzní nástroj a úprava codebase systému pro celky, které byly vyhodnoceny jako klíčové.

**Výstup:** Průběžná zpráva o postupné optimalizaci databáze, zpracování plánu migrace a úprav implementace stávajícího systému.

## Celky zahrnuté do 2. fáze projektu □

- Registrace uživatelů a subjektů
- Aukce
- Dokumenty k aukcím (část návrhu)
- Autentizační logika pro subjekty

## Registrace uživatelů a subjektů

### Změny databázového schématu

#### reg\_request

Tabulka “reg\_request” zcela zanikne a nahradí ji nové tabulky “registration\_subject”, kde budou uložena data k registrujícímu se subjektu, a “registration\_user”, kde budou uchovávána data k registrujícímu se uživateli nebo také data ke statutárnímu zástupci subjektu. Další pomocná tabulka vznikne také k uchovávání dokumentů požadované při registraci (kopie dokladů totožnosti, smlouva a případně další).

- **Zachované atributy**
  - ze současné tabulky se do nových tabulek zachovávají také následující atributy:
    - created
    - hidden
- **Nově vytvořené tabulky a jejich atributy:**
  - tabulka **registration\_subject:**
    - **atributy**
      - id, ico, company, dic, ic\_dph, file\_number, file\_number\_institution, file\_number\_city, address, zip\_code, city, country\_code, phone, created, hidden
    - **přidaná integritní omezení tabulky a další úpravy**
      - většina atributů je vyžadována (kromě “dic”, “ic\_dph”, “file\_number\_\*” a “phone”)
      - oproti původní tabulce “reg\_request” byly upraveny (zkráceny a sjednoceny) velikosti atributů datového typu string
  - tabulka **registration\_user:**

- **atributy:** id, email, passwd, first\_name, last\_name, sex, address, zip\_code, city, country\_code, phone, registration\_subject\_id, created, hidden
- **přidaná integritní omezení tabulky a další úpravy**
  - unikátní index nad atributem “email” - není dovoleno vkládat duplicitní záznamy se stejným emailem, neboť ten je jedním z přihlašovacích údajů
  - většina atributů je vyžadována (kromě phone)
  - oproti původní tabulce “reg\_request” byly opraveny (zkráceny a sjednoceny) velikosti atributů datového typu string
  - atribut “sex” je kontrolován na hodnoty “F” nebo “M” - jiné hodnoty není dovoleno vkládat
- tabulka **registration\_user\_document:**
  - **atributy:**
    - registration\_user\_id ⇒ FK na “registration\_user.id”
    - registration\_user\_document\_id ⇒ FK na “registration\_user\_document.id”
  - **přidaná integritní omezení tabulky:**
    - unikátní index nad atributem “registration\_user\_document\_id” (⇒ simulace vztahu 1:N přes vazební tabulku) - není dovoleno přiřazovat stejný dokument více uživatelům (toto specifické řešení si klade za cíl zabránit případné chybě v aplikaci, aby se nedala např. již přiřazená kopie OP přiřadit znovu jinému uživateli)
  - **další poznámky:**
    - tabulka je modelována bez speciálního primárního klíče, primární klíč tvoří oba atributy tabulky společně.
- tabulka **file\_storage:**
  - **univerzální tabulka pro různé dokumenty** (nikoliv pouze pro registraci) - data souborů tak nebudou součástí řádků původních tabulek, ale budou uložena zvlášť, což tabulky zkrátí, a tak bude při vyhledávání v tabulkách načteno méně stránek (datových bloků).
  - **atributy:** id, filename, filedata, created, hidden
- **Další doporučené změny:**
  - **číselník států** - nová tabulka “country” s definicí povolených zkratk států, z kterých je možno vybírat
  - **číselník měst** - nová tabulka “city”
    - z tabulky registrací by se vyčlenily následující atributy:
      - country\_code, zip\_code, city
    - v tabulce “registration\_subject” by byl nově zaveden cizí klíč “country\_id”, který by odkazoval na “city.id”
    - pozor, zde by bylo třeba uvážlivě rozmyslet, zda tato města (kombinace město + PSC + stát) ponechat pouze jako návrhy ke schválení nebo zda je již dále neschvalovat a rovnou je nabízet při další registraci (nebezpečí chyb, potenciální riziko úmyslného zneužívání pro reklamy nebo šíření vulgárností či nenávisti apod.)



## Migrační skript

Do migračního skriptu schématu databáze byly definovány výše uvedené nové tabulky včetně jejich atributů a integritních omezení.

Migrační skript pro přenesení dat ze staré tabulky “reg\_request” po dohodě s iA nebyl vytvářen - staré registrace doběhnou ve staré tabulce, s nasazením nových registrací budou vedeny již jen v nových tabulkách. Hlavním důvodem, proč byla migrace dat zavržena, je skutečnost, že ve starém schématu nebylo vedeno spoustu atributů, které jsou nově vedeny. To by pro novou tabulku znamenal stav chybějících dat, s kterým by se bylo potřeba vypořádat buď změnou nových atributů na nepovinné, což není úplně žádoucí stav (ačkoliv si kontroly řeší také samotná aplikace, je rozumné mít kontroly i v databázi pro případ chyb v aplikaci nebo ručního doplňování přímo do databáze), nebo doplňováním chybějících dat (buď dle skutečnosti nebo falešnými daty - například prázdný řetězec nebo mezera).

## Úpravy v aplikaci

Vzhledem k tomu, že se jedná o zcela novou koncepci registrace v celé hloubce, kterou iA připravuje, úpravy v aplikaci provádí sama firma iA. Úkolem FIT tedy bylo připravit pro iA novou strukturu tabulek pro registraci, aby se už od počátku vyhnulo např. prohřeškům proti normalizaci a nemuselo tak být později do této části zasahováno (upravováním tabulek, migrováním dat a úpravami aplikace).

## Aukce

### Změna databázového schématu

Prvním krokem bylo odstranění atributů, které nejsou aplikací využívány. Jde o následující atributy:

- Tabulka auction: min\_price, max\_price, whencreated, whenreview, whencofirm, whenready, whenaborted, whenfinanced, whentransferred, whenpaidback, whencleared, sidenotes, createdbycustomer
- Tabulka bid: amount, createdbycustomer
- Tabulka offer: amount, createdbycustomer

Dále bylo potřeba zajistit, aby byly historické údaje z tabulek bid a offer archivovány. K tomuto účelu byly ve skriptu vytvořeny tabulky bid\_dump a offer\_dump, které budou archivovány, ale aplikací dále využívány nebudou.

Aby se zabránilo častému spojování tabulek auction a auction\_status, bude atribut status v tabulce auction obsahovat vždy aktuální hodnotu stavu (tj. atributu val z tabulky auction\_status). Aby toto bylo možné, z důvodu omezení frameworkem web2py bude tato funkcionální zajištěna pomocí databázového triggeru (viz dále) a nikoli klasickým cizím klíčem.

## Migrační skript

Výsledný migrační skript obsahuje následující příkazy:

- Příkazy pro vytvoření tabulek `bid_dump` a `offer_dump`, které budou obsahovat veškeré údaje z historie, včetně atributů, které budou z aktuálních tabulek `bid` a `offer` odstraněny.
- Příkazy pro vytvoření indexů – protože jde z hlediska rychlosti o klíčovou část, bylo pro zrychlení vyhledávání vytvořeno celkem 5 indexů:
  - `idx_status`: pro urychlení vyhledávání v tabulce `auction` podle hodnoty atributu `status`
  - `idx_auctionstat`: pro urychlení spojení tabulek `auction` a `auction_status`
  - `idx_bid` pro urychlení spojení tabulek `bid` a `auction`
  - `idx_offer`: pro urychlení spojení tabulek `offer` a `auction`
  - `idx_biduser`: pro urychlení spojení tabulek `bid` a `auth_user`
- Vytvoření databázového triggeru, který zajistí, aby se vždy při vložení nebo modifikaci záznamu v tabulce `auction_status` aktualizovala hodnota aktuálního stavu (atribut `status`) v tabulce `auction`. V těle triggeru je spouštěna funkce `status_update`, která provede samotnou aktualizaci hodnoty příslušného záznamu tabulky `auction`.
- Funkce `migrate_status` slouží k tomu, aby aktuální hodnota stavu aukce byla uložena také ve všech záznamech z minulosti. Pro každou aukci v databázi najde v tabulce `auction_status` nejnovější hodnotu stavu, a tu uloží do atributu `status` u příslušné aukce.

## Úpravy v aplikaci

Díky výše zmíněným úpravám databáze bylo možné zefektivnit některé dotazy spouštěné v kódu aplikace `iA`. Především bylo možné na několika místech odstranit nutnost spojení tabulek `auction` a `auction_status`.

Dále došlo ke zjednodušení funkce `get` ve třídě `AuctionLister`. Byly sloučeny dva dotazy do jednoho, čímž bylo dosaženo celkového zjednodušení procedury a mohlo by vést k vyšší efektivity získávání dat z databáze.

## Dokumenty k aukcím

### Změny databázového schématu

Nejpodstatnější úpravou databáze v rámci této části je nutnost vytvořit novou tabulku `invoice_auction` pro ukládání faktur, které jsou doposud ukládány do tabulky `doc_auction` spolu s ostatními dokumenty. Tím dojde ke zmenšení tabulky `doc_auction` a odstranění atributů `invoice_num`, `due_date`, `currency`, `debtor_ico`, `variable_symbol` a `paid`, které se týkají pouze faktur.

Toto musí být také zohledněno v tabulkách `doc_include` a `file_auction`, kde budou také přidány příslušné cizí klíče do tabulky `invoice`.

Pro podporu nové funkcionality, kdy bude možné zaplatit fakturu v několika splátkách, je potřeba také vytvořit tabulku `payment` – bude obsahovat cizí klíč do tabulky faktur, dále částku, datum a identifikátor transakce.

Z důvodu redundance je možné z tabulky `doc_auction` odstranit atribut `debtor_ico` a z důvodu nevyužívání v aplikaci atribut `description`.

## Migrační skript

V migračním skriptu bude především potřeba zajistit, aby byly záznamy z původní tabulky doc\_auction byly správně rozděleny do dvou nově vzniklých tabulek a musí být také správně nastaveny navazující cizí klíče z tabulek doc\_auction a file\_auction.

Bude také analyzován kód aplikace za účelem zjištění, zda by se nevyplatilo pro některou z tabulek vytvořit index pro zrychlení vyhledávání v tabulce.

## Úpravy v aplikaci

V aplikaci bude především potřeba upravit části pracující s fakturami, aby pracovaly s novou tabulkou invoice\_auction. V rámci analýzy zdrojového kódu aplikace budou případně navrženy další možnosti optimalizace dotazů za účelem celkového zrychlení aplikace.

## Autentizační logika pro subjekty

### Změny databázového schématu

Vlastnosti jednotlivých atributů v tabulkách subjektů byly co nejvíce sjednoceny s vlastnostmi jednotlivých atributů registrace subjektů. Jediný rozdíl, na který je třeba dát pozor, je, že v registraci subjektů se ukládají pouze právnické osoby, kdežto v tabulce subjektů jsou i fyzické osoby, proto se subjekty od registrací liší například v tom, že například IČO není povinné.

### Tabulka auth\_group

- **ia**
  - **sjednotit délku atributů => registrace vs. auth\_group**
  - **sjednotit povinnosti => pozor, do subjektů se ukládají jak FO, tak PO**
- **Odstraněné atributy:**
  - country - řeší se překladem na základě "country\_code"
  - file\_num - nepoužívané (rozděleno do "file\_number" + "file\_number\_institution" + "file\_number\_city")
  - bank\_account - nepoužívané
  - ia\_collect\_account - nepoužívané
  - ia\_frominv\_account - nepoužívané
  - type\_auth - vznikem tabulek "buyer", "seller", "debtor" již nebude zapotřebí rozlišovat tento stav neboť je každá entita vyčleněna do vlastní tabulky
- **Nově vytvořené tabulky:**
  - tabulka **subject**:
    - **atributy vyčleněné z tabulky auth\_group:**
      - name (*přejmenováním z auth\_group.role*), address, city, zip\_code, country\_code, ico, dic, ic\_dph, file\_number, file\_number\_city, file\_number\_institution, added, hidden
    - **nové atributy:**
      - id
  - tabulka **subject\_spv**:
    - **nové atributy:**
      - id

- subject\_id ⇒ FK na “subject.id”
- tabulka **subject\_buyer**:
  - **atributy vyčleněné z tabulky auth\_group**:
    - investor\_auctions\_win\_limit
    - investor\_days\_limit
  - **nové atributy**:
    - id
    - subject\_id ⇒ FK na “subject.id”
- tabulka **subject\_seller**:
  - **atributy vyčleněné z tabulky auth\_group**:
    - collect\_account\_id
    - frominv\_account\_id
    - fee\_with\_vat
    - subject\_spv\_id ⇒ FK na “subject\_spv.id”
  - **nové atributy**:
    - id
    - subject\_id ⇒ FK na “subject.id”
- tabulka **subject\_debtor**:
  - **atributy vyčleněné z tabulky auth\_group**:
    - debtor\_seller (přejmenovaný atribut “debtor\_owner”)
    - fee\_percent
    - fee\_based\_on\_face\_value
  - **nové atributy**:
    - id
    - subject\_id ⇒ FK na “subject.id”
- **Změny atributů v ostatních tabulkách**:
  - group\_id ⇒ subject\_id
    - **migrace atributu mezi tabulkami**:
      - auth\_membership.group\_id ⇒ auth\_user.subject\_id
    - **migrace atributu v rámci tabulek**:
      - bank\_account
      - client\_bank\_account\_include
  - group\_id ⇒ subject\_debtor\_id
    - **migrace atributu v rámci tabulek**:
      - rating
  - group\_id ⇒ subject\_seller\_id
    - **migrace atributu v rámci tabulek**:
      - auction
      - doc\_auction
      - auction\_sandbox
  - client\_id ⇒ subject\_seller\_id
    - **migrace atributu v rámci tabulek**:
      - client\_debtor
  - group\_id ⇒ subject\_buyer\_id
    - **migrace atributu v rámci tabulek**:
      - investor\_private\_map
      - price\_limit (*pozn.: nově vzniklá tabulka od iA*)
      - rating\_limit (*pozn.: nově vzniklá tabulka od iA*)
      - due\_limit (*pozn.: nově vzniklá tabulka od iA*)
      - overdue\_limit (*pozn.: nově vzniklá tabulka od iA*)

- debtor\_id ⇒ subject\_debtor\_id
  - **migrace atributu v rámci tabulek:**
    - auction
    - auction\_sandbox
    - client\_debtor
- **Další doporučené změny:**
  - **číselník států** – nová tabulka “country” s definicí povolených zkratk států, z kterých je možno vybírat
  - **číselník měst** – nová tabulka “city”
    - ze “subject” by se vyčlenily následující atributy:
      - country\_code
      - zip\_code
      - city
    - v tabulce “subject” by byl nově zaveden cizí klíč “country\_id”, který by odkazoval na “city.id”
- **Změna vlastností atributů:**
  - name:
    - string/128 ⇒ string/64
  - address:
    - string/512 ⇒ string/128
    - NULL ⇒ NOT NULL
  - city:
    - string/128 ⇒ string/64
    - NULL ⇒ NOT NULL
  - zip\_code: NULL ⇒ NOT NULL
  - country\_code: NULL ⇒ NOT NULL
  - file\_number: string/512 ⇒ string/16
  - file\_number\_institution: string/512 ⇒ string/32
  - file\_number\_city: string/512 ⇒ string/32
  - collect\_account\_id: integer ⇒ FK na “bank\_account.id”
  - frominv\_account\_id: integer ⇒ FK na “bank\_account.id”
  - debtor\_owner:
    - integer ⇒ FK na “seller.id”
    - přejmenování na “debtor\_seller”
  - added: výchozí hodnota ⇒ aktuální časová značka

## Migrační skript

Migrace této části je náročná, neboť zasahuje do větší části databáze. Jelikož se data z tabulky auth\_group rozpadnou do vícero tabulek, bude třeba upravit i atributy v ostatních tabulkách, které do té doby na auth\_group ukazovaly cizím klíčem.

Další problematickou částí migrace je to, že původní tabulka auth\_group obsahovala řádky, v nichž byly několikrát uvedeny téže firmy. Někdy však v některém z atributů (název firmy, různé části adresy, IČO, DIČ, IČ DPH a různé části spisové značky) chyběly údaje, jindy tam byly uvedeny stejné údaje v jiném formátu (např. různý zápis ulice: Třída Tomáše Bati, tř. Tomáše Bati, tř. T. Bati, T. Bati apod.), případně s překlady. Takovéto údaje budou muset být ručně opraveny před samotnou migrací - pro tento účel jsem pro iA napsali speciální předmigrační skript na kontrolu takovýchto problémových řádků a atributů, který vypíše

kterých řádků a kterých atributů se to týká a co je zde za problém - viz následující příklad na testovacích datech:

```
WARNING: Check auth_group.ico = "11223344". Following column(s) has different values for same ICO in auth_group table:
WARNING: * role contains values:
WARNING:   - client set_status_test
WARNING:   - set_status_test client
WARNING: * address contains values:
WARNING:   -
WARNING:   - Prašná 14
WARNING: Check auth_group.ico = "44332211". Following column(s) has different values for same ICO in auth_group table:
WARNING: * role contains values:
WARNING:   - investor set_status_test
WARNING:   - set_status_test investor
WARNING: Check auth_group.id = 183739. Following column(s) can not be NULL: address, city, zip_code.

ERROR: Can not complete migration due to missing data - see details above.
***** Chyba *****

ERROR: Can not complete migration due to missing data - see details above.
Stav SQL: P0001
```

Do migračního skriptu schématu databáze byly definovány výše uvedené nové tabulky včetně jejich atributů a integritních omezení.

Datové migrace jsou poněkud komplikované, neboť v nové tabulce vznikají nové identifikátory řádků, které je třeba namapovat tak, aby vazby s dalšími tabulkami zůstaly zachovány. Pro tyto účely je nutné mapovat starý identifikátor ⇒ nový identifikátor; navíc z tabulky auth\_group vznikly další 4 nové tabulky, do kterých se budou jiné tabulky (dle významu atributů v těchto tabulkách) odkazovat cizím klíčem. Vzniká tak potřeba uchování dalších 4 map.

Jelikož se jedná o podstatný zásah do celé databáze, bude nutné ve všech měněných tabulkách ověřit, zda celá migrace proběhla v pořádku.

Další komplikací je to, že migrace schématu bude muset proběhnout dvoustupňově. Prvně bude schéma databáze rozšířeno o nové tabulky a nové atributy cizích klíčů, které však nemohou být, vzhledem k již existujícím datům v produkční databázi, vyžadovány jako povinné. Následně v druhém kroku, po migraci dat, musí proběhnout ještě jedna migrace schématu databáze, kde budou zavedeny některé z nových cizích klíčů jako povinné (po migraci dat již tyto atributy budou obsahovat data a nebudou NULL jako v okamžiku po vytvoření / před provedenou datovou migrací) a budou odstraněny staré cizí klíče a přidány atributy do tabulky auth\_group.

## Úpravy v aplikaci

Jak již bylo uvedeno u kapitoly migračního skriptu, tento celek zasahuje do podstatné části aplikace – úpravy aplikace jsou tedy rovněž poněkud rozsáhlejší. Bude tedy třeba tento celek velmi důkladně otestovat a doporučujeme jej nasadit separátně (kvůli velkému množství změn a velkému dopadu zásahů).

# Zpráva k 3. fázi projektu

**Cíl:** Ve třetí fázi budou zpracovány designové a formální nedostatky databázového modelu. Budou optimalizovány nevhodné datové typy, hledání přítomnosti redundantních dat. Při řešení bude využit databázový systém PostgreSQL, další technologie dle provedené analýzy.

**Výstup:** Závěrečná zpráva o úpravách a optimalizaci databáze, plánu migrace a celkových úpravách stávajícího systému.

## Celky zahrnuté do 3. fáze projektu

- Autentizační logika pro subjekty (dokončení)
- Faktury za aukce
- Sandbox k aukcím
- Autentizační logika pro uživatele
- Odesílání e-mailů
- Bankovní účty
- Historie kurzů a daní
- Ostatní nezařazené tabulky

## Autentizační logika pro subjekty

Tento celek byl popsán již v 2. zprávě (průběžná optimalizace a migrace). Při testování a nasazování vzniklo ještě několik požadavků na úpravy vzhledem k specifičnosti produkčních dat.

## Změny databázového schématu

Některé z atributů v tabulce subject se ukázaly být zmenšené až příliš pro produkční data, kdy se našel pro každý atribut jeden exemplář, který se do menších sloupců nevešel. Délka atributu name proto byla navýšena z 64 na 192, délka atributu file\_number ze 16 na 64 (v souvislosti s touto změnou bylo provedeno stejné navýšení u odpovídajících atributů také u tabulky registration\_subject).

Všechny zbývající cizí klíče subject\_id (vedoucí na tabulku subject) kromě těch, které se nacházejí v tabulkách subject\_\*, byly změněny za konkrétní typy subjektů subject\_<typ>\_id (vedoucí na příslušnou tabulku subject\_<typ>):

- V tabulce client\_bank\_account\_include byl cizí klíč subject\_id nahrazen atributem subject\_seller\_id.
- V tabulce auth\_user byl cizí klíč subject\_id nahrazen dvěma konkrétními cizími klíči subject\_buyer\_id a subject\_seller\_id
- V tabulce bank\_account byl cizí klíč subject\_id nahrazen konkrétními cizími klíči všech typů subjektů, pro větší přehlednost byl, narozdíl od předešlých, zvolen prefix owner\_\*, který by měl naznačit, že se jedná o majitele účtu daného typu (investor / klient / odběratel): owner\_buyer (vedoucí na tabulku subject\_buyer), owner\_seller (vedoucí na tabulku subject\_seller) a owner\_debtor (vedoucí na tabulku subject\_debtor)

## Migrační skript a úpravy v aplikaci

Protože se jednalo o změnu před nasazením do produkce (v době ladění aplikace s produkčními daty na testovacím serveru), byly výše uvedené změny promítnuty do migračního skriptu popsaném v předchozí zprávě a odpovídajícím způsobem upraveny také v aplikaci.

## Autentizační logika pro uživatele

Z části “Autentizační logika pro uživatele” byla v druhé fázi projektu vyčleněna část týkající se registrací do samostatné části. Pro tuto část tak zbývají tabulky `auth_user`, `auth_event` a `auth_cas`. Do tabulky `auth_cas` nebylo iA nijak zasahováno, proto v ní žádné změny neprovádíme.

## Změny databázového schématu

Do tabulky `auth_event` byl iA přidán pouze jediný atribut `auction_id`. V této tabulce, jako jediné, jsme přidané atributy ponechali, pouze jsme je trochu upravili. Všechny přidané atributy dostaly prefix `ia_`. Atribut `auction_id` jsme změnili na `ia_entity_id`, kde jsou uchovávány `auction_id`, zároveň zde mohou být pro účely logování uchovávány id dalších entit. Typy těchto entit jsou rozlišovány dle atributu `ia_entity_target`.

V tabulce `auth_user` bylo přidáno firmou iA několik atributů. Všechny tyto atributy, které nepocházejí z `web2py`, jsme z bezpečnostních důvodů vyčlenili do nové tabulky `person`. Primární klíč této tabulky je zároveň cizím klíčem do tabulky `auth_user`. Dále jsme odstranili nepoužívané atributy `lang`, `notify_auction_new` a `reg_request_id`.

## Migrační skript

V rámci migračního skriptu bylo potřeba zajistit přesunutí dat z přidaných atributů tabulky `auth_user` do nové tabulky `person`.

Podobně, pro migraci v rámci tabulky `auth_event` bylo třeba migrovat data a také přidat typ entity do tabulky `ia_entity_target`.

## Úpravy v aplikaci

Vyčleněním atributů z tabulky `auth_user` do nové tabulky bylo třeba změnit na poměrně dosti místech v aplikaci. Ačkoliv atributy jako `RČ` či `OP` používá pouze modul `subject`, některé další atributy jako například `notify_` jsou používány také například v profilu uživatele či v maileru.

Úpravy pro `auth_event` byly poměrně jednoduché v jediném modulu a jeho testovacím skriptu.

## Faktury za aukce

### Změny databázového schématu

Prvním krokem bylo odstranění atributu, který není v praxi využíván. V případě tohoto celku se jedná o atribut `file_num` v tabulce `ia_info`.



Dále je v tabulce `client_fee_invoice` mnoho redundantních atributů. Jde například o atributy začínající prefixem `ia_`, které uchovávají informace o firmě Investiční Aukce. Ty jsou však již uloženy v tabulce `ia_info`, a protože se tyto údaje příliš často nemění, mnohokrát se v rámci této tabulky opakují. Proto tyto atributy byly z tabulky odstraněny a nahrazeny cizím klíčem do tabulky `ia_info`.

Podobná je situace s atributy začínajícími prefixem `client_`, které jsou uvedeny v nově vzniklé tabulce `subject_client`. I tyto budou z tabulky odstraněny a nahrazeny cizím klíčem.

V tabulce se také nachází atribut `ia_bank_account`. Pro zajištění celkové konzistence databáze bude nahrazen cizím klíčem do tabulky `bank_account`, kde se nacházejí veškeré informace o bankovních účtech.

Aby byly informace o firmě iA na jednotlivých fakturách vždy pravdivé, bude potřeba v tabulce `ia_info` zavést historii, tj. umožnit změnu hodnot tak, aby historické hodnoty zůstaly zachovány. Informace se tedy z této tabulky v budoucnu nebudou mazat, do tabulky je přidán atribut `valid_to` a v případě změny některého atributu bude vložen nový záznam a u starého záznamu se nastaví hodnota tohoto atributu na aktuální datum.

## Migrační skript

V rámci migrace dat je potřeba zajistit, aby k jednotlivým fakturám byly dostupné aktuální informace o firmě Investiční aukce z doby vydání této faktury. To je ve skriptu realizováno seřazením faktur podle času a průchodem faktur v tomto pořadí. V případě změny údajů dochází k přidání nového záznamu do tabulky `ia_info`.

Analogicky je potřeba zajistit, aby po odstranění informací o klientovi z tabulky `client_fee_invoice` existoval vždy záznam se správnými hodnotami v tabulce `subject_client`, na který se tabulka bude odkazovat cizím klíčem. Zde historie není nutná, jen je potřeba, aby odkaz směřoval na správného klienta.

Dále je nutné nastavit cizí klíč do tabulky bankovních účtů. Proto pro každou hodnotu v rámci atributu `ia_bank_account` je nutné vyhledat v tabulce `bank_account` příslušný účet, jehož hodnota primárního klíče (atribut `id`) bude uložen jako hodnota zmiňovaného cizího klíče.

Dále byl v rámci skriptu u tohoto celku vytvořen nový index pro rychlejší vyhledávání v tabulce `client_fee_invoice`.

## Úpravy v aplikaci

Bylo potřeba upravit kód pracující s tabulkou `client_fee_invoice` tak, aby informace o firmě iA, o klientovi a o bankovních účtech byly čteny, resp. zapisovány do příslušných odkazovaných tabulek. Dále bylo potřeba zajistit, aby k nově vznikajícím fakturám byl vždy z tabulky `ia_info` přiřazen nejnovější záznam s informacemi o firmě iA.

# Sandbox k aukcím

## Změny databázového schématu

U tabulek patřících do tohoto celku (`auction_sandbox`, `doc_sandbox`, `file_sandbox`) nebyly nalezeny větší nedostatky v návrhu, proto se jejich struktura nebude měnit. Pro celkové zpřehlednění databáze byly tyto tabulky přejmenovány tak, aby začínaly prefixem `sandbox_`, tj. např. `sandbox_auction` což je jediná změna, která byla v rámci tohoto celku v databázovém schématu provedena.

## Migrační skript

Jedinou akcí v rámci tohoto celku je přejmenování zmíněných tří tabulek. V migračním skriptu tedy došlo k tomuto přejmenování a následně je potřeba toto přejmenování promítnout do souboru `10_db.py`, kde je uložena struktura databáze, se kterou pracuje framework `web2py`.

## Úpravy v aplikaci

Také úpravy aplikace jsou velmi jednoduché, zahrnují pouze přejmenování všech výskytů názvů tabulek z tohoto celku.

# Odesílání e-mailů

## Změny databázového schématu

V první řadě pro sjednocení názvosloví tabulek jsou tabulky `mail_template` a `mail_queue` přejmenovány na `email_template` a `email_queue`.

Bylo potřeba také vyřešit izolovanou tabulku `email_template`, která by měla být navázána pomocí cizího klíče na tabulku `email`. Vzhledem k tomu, že by se tato tabulka měla odkazovat na atribut `mail_type` (je v tabulce unikátní - toto integritní omezení bylo taktéž k tomuto atributu přidáno), bude tento cizí klíč realizován pomocí triggeru, s ohledem na použitý framework `web2py`.

Pro ukládání objektů typu BLOB byla vytvořena již dříve tabulka `file_storage`, kde budou také ukládány přílohy k e-mailům, proto je potřeba atribut `file_blob` z tabulky `email_attachment` odstranit a nahradit jej cizím klíčem do tabulky `file_storage`.

Na závěr této fáze bylo potřeba vyřešit redundantní atributy v tabulkách `email` a `email_queue`, které obsahují identické atributy `sender` a `recipient`. Tyto atributy jsou ponechány v tabulce `email` a z tabulky `email_queue` tedy byly odstraněny.

Vzhledem k množství emailů je vhodné pro rychlejší vyhledávání informací v této tabulce vytvořit databázový index.

## Migrační skript

Součástí bylo opět i přejmenování tabulek. Jako v předchozím celku, v migračním skriptu došlo k tomuto přejmenování a následně bylo potřeba toto přejmenování promítnout do souboru `10_db.py`, kde je uložena struktura databáze, se kterou pracuje framework `web2py`.

Migrační skript musí v první řadě zajistit doplnění hodnoty cizího klíče do tabulky `email_template`, kde bylo potřeba na základě hodnoty `mail_type` v tabulce `email_queue` zjistit typ daného e-mailu a přidat do tabulky `email` hodnotu cizího klíče na odpovídající šablonu v tabulce `email_template`. To zajišťuje uložená funkce `migrate_template`.

Podobně je potřeba uskutečnit migraci příloh emailů do tabulky `file_storage`. Odpovídající objekt typu `BLOB` je přesunut do tabulky `file_storage` a do tabulky `email_attachment` vložen odkaz na tento nově vložený záznam pomocí cizího klíče.

## Úpravy v aplikaci

V aplikaci bylo potřeba upravit veškeré dotazy týkající se redundantních atributů v tabulkách `email` a `email_queue` tak, aby se odpovídající hodnoty načítaly ze správných tabulek. Také byly upraveny příkazy (pouze jeden výskyt v programu) pracující s objekty typu `BLOB` v rámci příloh e-mailů. V tomto případě bylo nutné upravit dotaz tak, aby přistupoval navíc do tabulky `file_storage`.

Kromě toho muselo také dojít k přejmenování všech výskytů tabulek začínajících prefixem `mail_` na `email_` u všech výskytů názvů těchto tabulek ve zdrojovém kódu programu.

## Bankovní účty

### Změny databázového schématu

V tabulce bankovních účtů (`bank_account`) byl pro snížení rizika porušení konzistence zaveden u atributu `account_type` výčet čtyř hodnot, kterých může tento atribut nabývat.

Vzhledem k rozdělení subjektů do tabulek dle typů subjektů byl atribut `group_id` přejmenován na `owner_seller`, `owner_buyer` či `owner_debtor`, což je cizí klíč na příslušnou tabulku typu subjektu. Lze tak velmi jednoduše rozlišit, zda daný účet patří některému z investorů, klientů či odběratelů. Zároveň je atribut čitelnější, neboť je jasné, že jde o majitele účtu. Tato změna (rozdělení `group_id` na `owner_*` atributy) byla provedena již v rámci části pojmenované Autentizační logika pro subjekty.

Další nutnou úpravou je odstranění redundance u atributu `swift` (ten má daná banka vždy stejný pro všechny své účty), hodnota se tak v tabulce mnohokrát opakuje. Řešením je vytvoření nové tabulky `bank`, kde pro každou banku bude existovat vždy jeden `swift` kód a bude možné k bance uvést i další informace.

U tabulky `client_bank_account_include` jde o poměrně zavádějící název tabulky, nejde totiž o účty, které by vlastnil sám klient - proto došlo k přejmenování na `client_ia_bank_account`.

U tabulky `fio_token` nebyly nalezeny nedostatky, které by vyžadovaly modifikaci této tabulky.

### Migrační skript

Migrační skript pro tento celek obsahuje pouze funkci pro migraci starých dat z tabulky `bank_account` tak, aby odpovídala novému schématu obsahujícímu i tabulku `banks`. Provádí tedy přesun hodnoty `swift` kódu do této nové tabulky a správné nastavení cizího klíče v tabulce `bank_account`. Zároveň ještě přidává již zmíněné omezení pro atribut `account_type`.

Vzhledem k tomu, že tyto tabulky nejsou nijak výrazně rozsáhlé, nebyly zde vytvářeny žádné nové indexy ani další databázové objekty.

## Úpravy v aplikaci

Práce s bankovními účty se nachází opět ve více modulech, proto bylo třeba upravit aplikaci hned na několika místech (bankovni\_ucty, mailer, permission, subject a jejich testy).

Úpravy aplikace pro tabulku client\_bank\_account\_include byly naopak relativně jednoduché, kde bylo zapotřebí upravit 5 řádků ve 2 souborech.

## Historie kurzů a daní

### Změny databázového schématu

Do tohoto celku spadají tabulky exchange\_rate a tax, jejichž použití v aplikaci je spíše okrajové. V tabulce exchange\_rate nebyly provedeny žádné modifikace, k danému účelu (výpočtu jedné odvozené hodnoty, která se zobrazuje v aplikaci) slouží správně a neobsahuje žádné nadbytečné atributy. Tabulka tax obsahující aktuální výši daně z přidané hodnoty také není potřeba příliš měnit, pro zvýšení obecnosti do budoucna by v ní mohla být historie, které je dosaženo přidáním nového atributu valid\_to (aktuální hodnota DPH bude mít tuto hodnotu prázdnou).

V rámci tohoto celku byla vytvořena také tabulky auction\_sum, která bude obsahovat jednu hodnotu udávající součet hodnot všech ukončených aukcí. Tato tabulka je určena ke zrychlení výpisu této hodnoty v aplikaci iA, která je dosud vždy počítána dotazem do databáze při každém spuštění aplikace.

### Migrační skript

Jelikož historii DPH není nutné zpětně ukládat do databáze, zůstane v tabulce jeden řádek DPH, u níž bude atribut valid\_to nastaven jako prázdný. Žádnou migraci tedy v rámci této části není nutné provádět.

Součástí skriptu je také vytvoření databázového triggeru, který po ukončení aukce automaticky aktualizuje hodnotu součtu všech aukcí v tabulce auction\_sum.

## Úpravy v aplikaci

Vzhledem k tomu, že byla provedena jen jedna drobná změna, nebude nutné v rámci této části zasahovat do zdrojového kódu aplikace s výjimkou místa, kde je zjišťován celkový součet všech ukončených aukcí, kde stávající komplikovaný dotaz bude nahrazen jednoduchým dotazem na hodnotu v tabulce auction\_sum.

# Ostatní nezařazené tabulky

## Změny databázového schématu

Do této části byly zahrnuty tabulky `auction_filter` a `news`. V tabulce `auction_filter` bylo nalezeno několik nadbytečných údajů (jde konkrétně o atributy `due_date_min`, `due_date_max`, `end_date_min`, `end_date_max`, `status` a `edi`). Dále bylo potřeba sjednotit názvosloví atributů tak, aby `min` a `max` bylo vždy na konci názvu atributu (to se týká atributů `min_due_days`, `max_due_days`, `min_days_to_end`, `max_days_to_end` - ty jsou přejmenovány na `due_days_min` atd. analogicky u zbývajících). Kromě toho stávající model umožňoval do jednoho filtru pouze jednu měnu (`currency`) a obor (`industry`), což je podstatné omezení. Toto je možné řešit novou tabulkou, kde by byly tyto atributy, a k tabulce `user` by byla ve vztahu M:N. Druhou možností je nastavit pevnou hodnotu maximálního počtu těchto hodnot. V našem případě jsme realizovali druhou možnost, aby nedošlo ke zbytečně složitému schématu databáze.

U tabulky `news` nejsou nutné žádné úpravy.

## Migrační skript

Tabulka doposud nebyla využívána (předpokládá se v budoucnu), není tedy nutné migrovat žádná data ani vytvářet další databázové objekty.

## Úpravy v aplikaci

Z výše uvedeného důvodu nebylo u tohoto celku nutné provádět žádné úpravy ve zdrojovém kódu aplikace.