

Skupinový projekt do Bezpečnosti Databázových Systémů

Databázový dizajn s Data Definition Language (DDL) skriptom

Autori: Adam Kubiš (213431), Jakub Jarina (230086)

Odbor: Informačná bezpečnosť

Skupina: štvrtok 17:00 – 19:00

Rok: 2021/2022

Obsah

Úvod.....	3
Cieľ projektu	4
Dizajn a popis databázy	5
User_Layer	5
Product_Layer	5
Order_Layer	5

Úvod

V tejto štúdii si bližšie rozoberieme prvý skupinový projekt do predmetu Bezpečnosť Databázových Systémov. V prvých odsekoch Vám bližšie priblížime zadanie projektu a akým spôsobom sme sa ho v našej skupine ujali, respektíve akým smerom sme náš projekt ďalej rozvíjali. Hlavnou podstatou bude opísanie našich myšlienkových postupov, ukážeme a popíšeme Entity Relationship Diagram (ERD) s jednotlivými tabuľkami a nakoniec prejdeme či databáza splňuje všetky 3 normy.

Cieľ projektu

Keďže výber témy na projekt bol ľubovoľný, po dlhšom zvážení viacerých možností skupina dospela k záveru, že by rada venovala pozornosť na využitie databáz v e-shopoch. Rozmach e-shopov posledné desaťročie na internete exponenciálne vzrástol, taktiež obidvaja členovia skupiny majú dobré skúsenosti s e-shopmi so strany zákazníka, preto bolo jednoduchšie si predstaviť, navrhnuť a správne zostrojiť databázu.

Po dohode sme sa rozhodli, že zostrojíme komplexnú e-shop databázu, ktorá bude schopná uchovávať zákazníkov s danými adresami. Používateľ s oprávnením (zamestnanec) bude môcť nahrávať nové produkty a nakoniec samotná možnosť objednávky, kedy zákazník má možnosť si kúpiť naskladnený produkt, pomocou nejakej platobnej metódy. Z tohto dôvodu sme rozdelili databázu do troch hlavných častí a to nasledovne: *User_Layer*, *Product_Layer* a *Order_Layer*. Ako sa môže na prvý pohľad zdať najmä pri atribútoch produktov, e-shop by sa zameriaval skorej na elektroniku, z dôvodu potreby rozlišovať model, verziu poprípade typ a farbu produktu.

Dizajn a popis databázy

Ako je už spomínané vyššie celá databáza je rozdelená pre lepšiu organizáciu do 3 hlavných blokov: *User_Layer*, *Product_Layer* a *Order_Layer*. Následne si priblížime každý jeden blok, z akých tabuliek pozostáva a aké majú funkcie.

User_Layer

V tejto vrstve sa nachádza 5 tabuliek. Tabuľka *user*, medzi-tabuľka *users_has_address* prepájajúca tabuľku *address*, v ktorej sa uchováva potrebné údaje užívateľa a nakoniec tabuľka *account*, v ktorej sa ukladajú možné typy užívateľov a medzi-tabuľka *users_has_account*, ktorá ukladá údaje o kontách užívateľov. Tabuľka *users* uchováva potrebné atribúty ako sú:

- *id_user* - vo formáte integer, ktorý slúži ako hlavný kľúč pre identifikáciu
- *first_name/last_name* - vo formáte Variabilného reťazca, pre uloženie mena a priezviska.
- *email a phone_number* - vo formáte Variabilného reťazca, ako potrebné údaje (not null), z dôvodu overenia užívateľa

Pri zákazníkovi sú potrebné ďalšie údaje ako je adresa najmä kvôli možnosti doručenia objednávky alebo overenia. Keďže užívateľ má možnosť mať aj inú doručovaciu adresu, najlepšie riešenie bolo vytvoriť tabuľku *address* a prepojiť ju s *user* pomocou medzi-tabuľky *users_has_address*, ktorá ukladá a synchronizuje cudzie kľúče daných tabuliek. Tabuľka *address* uchováva potrebné atribúty ako sú:

- *id_address* - vo formáte integer, ktorý slúži ako hlavný kľúč pre identifikáciu
- *street, postcode, city a country* – vo formáte Variabilného reťazca
- *house_number* – vo formáte integer

Užívateľov treba rozlišovať na základe ich oprávnení, preto je vytvorená tabuľka *account*, v ktorej sa ukladá typ užívateľa (napr. admin, customer, employee)

Product_Layer

Product_Layer sa skladá z dvoch tabuliek. *Product* tabuľka, ktorá ukladá všetky potrebné informácie o produkte. Názvy elektroniky sú väčšinou dané nasledovne: *brand_name, model, version*, preto *Product* tabuľka obsahuje tieto atribúty a taktiež nevyhnutné atribúty ako sú *color a price*. Nakoniec tabuľka *product_type* slúži na uloženie typu produktu, ktorý sa priradzuje cez cudzí kľúč pomocou atribútu *id_product_type*.

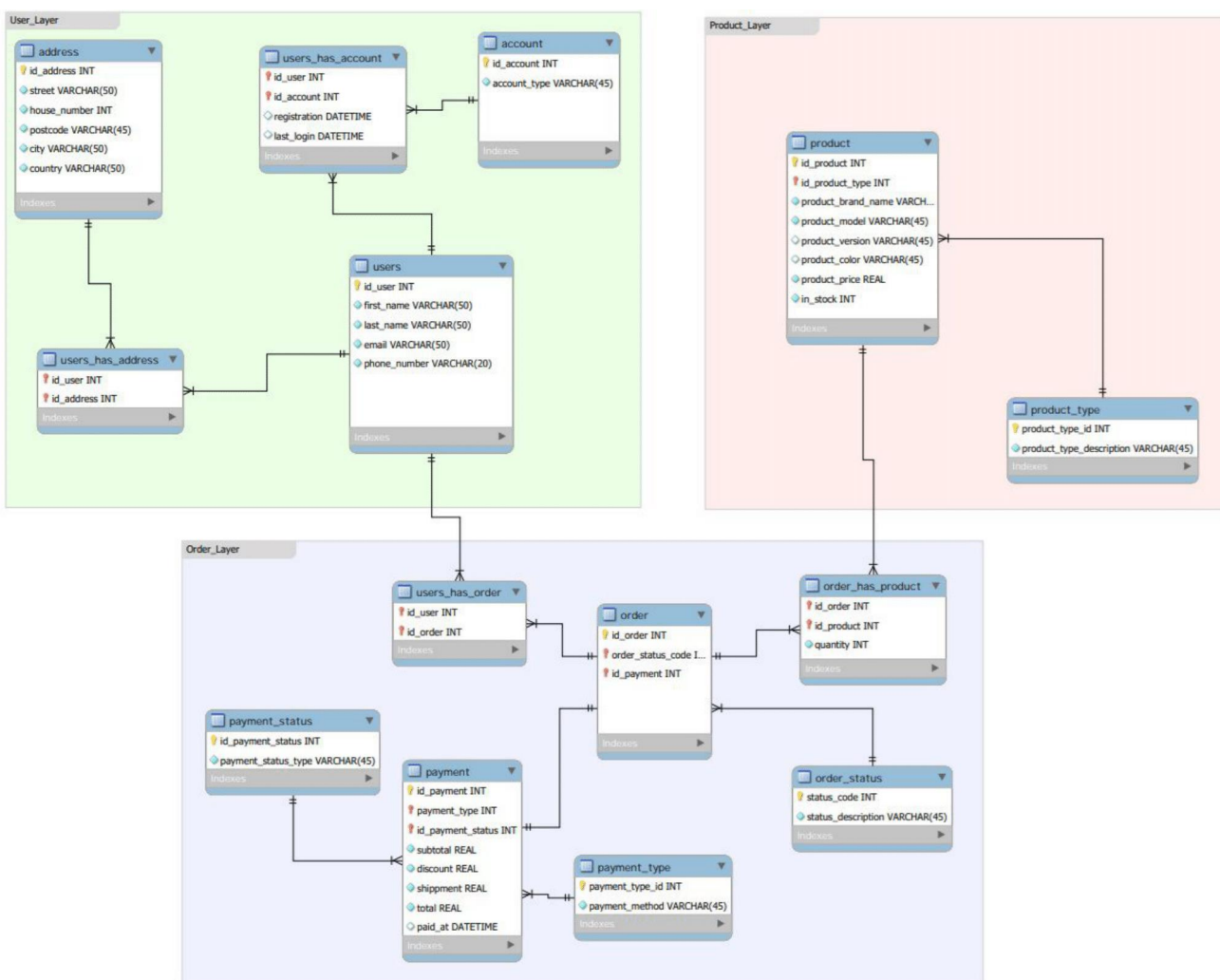
Order_Layer

V poslednej vrstve sú 2 hlavné tabuľky *order* a *payment*, ktoré majú vzťahy s vedľajšími tabuľkami *order/payment_status* a *payment_type*. Tabuľka *order* pozostáva z primárneho kľúča *id_order* a 2 cudzích kľúčov *order_status_code* a *id_payment*. Zákazník je priradený k danej objednávke pomocou medzi-tabuľky *users_has_order*, v ktorej sú uchovávané cudzie kľúče. Pomocou ďalšej medzi-tabuľky *order_has_product* priradzujeme k objednávke požadované produkty, ktoré si zákazník chce objednať. Potrebný je aj atribút *quantity*, ktorý udáva množstvo

objednaného produktu. Každá objednávka musí mať vlastnú platbu, preto je vytvorená tabuľka *payment*, ktorá uchováva sumy pred zdanením (subtotal), prípadná cena dopravy. Nakoniec je možnosť uložiť *discount* na platbu a výsledná cena sa ukladá do atribútu *total*. Všetky vyššie zmienené atribúty sa ukladajú vo formáte real. Zákazník má možnosť vybrať spôsob platby, preto sme rozšírili tabuľku *payment* o *payment_type*. Pri oboch tabuľkách *order* a *payment* je nutné vedieť o momentálnom stave, a preto bolo potrebné vytvoriť ďalšie 2 tabuľky *payment/order_status*, v ktorých je uložený možný opis buď zásielky alebo už danej platby.

V obrázku nižšie je možné vidieť grafické prevedenie databázy cez ERD diagram za pomoci Crow's Foot Notation.

Obrázok č.1



Primárnym cieľom pri dizajne samotnej databázy bolo dodržanie prvých 3 noriem. Prvá norma požaduje aby každý atribút bol atomický, čo znamená, ďalej nedeliteľný. Tohto sme sa snažili držať najmä pri tabuľkách *user*, *address* a *product*, kde sme názvy mien, produktu alebo adresy rozdelili do čo najmenších možných častí. Druhá norma je rozšírená o fakt, že každý neklúčový atribút musí byť funkčne závislý od primárneho kľúča. Ako je možné z diagramu vidieť, tabuľka *order* a *payment* je rozdelená, pretože atribúty tabuľky *payment* nie sú funkčne závislé od primárneho kľúča tabuľky *order*, aj keď majú medzi sebou vzťah a to, že každá objednávka má vlastnú platbu. V našom projekte sme sa snažili dosiahnuť aj tretej normy, ktorá je rozšírená o pravidlo, že žiaden neklúčový atribút nie je tranzitívne funkčne závislý od primárneho kľúča. To znamená že všetky neklúčové atribúty musia byť navzájom nezávislé. Toho sme využili pri tabuľkách *order/payment_status* a *payment/product_type*, kedy sme každému statusu a typu priradili ID, ktoré sa ukladajú do týchto tabuliek namiesto tabuliek *order* alebo *product*.