

Slovenská technická univerzita

Fakulta informatiky a informačných technológií

Ilkovičova 3, 812 19 Bratislava

Databázové systémy

Prvá a druhá iterácia

Jakub Ďurech

Študijný odbor: Internetové technológie -4

Ročník: 3.

Akademický rok: 2015/2016

Zadanie:

Vo vami zvolenom prostredí vytvorte databázovú aplikáciu, ktorá komplexne rieši minimálne 6 scenárov vo vami zvolenej doméne. Presný rozsah a konkretizáciu scenárov si dohodnete s Vaším cvičiacim na cvičení. Aplikáciu vytvoríte v dvoch iteráciach. V prvej iterácii, postavenej nad relačnou databázou, musí aplikácia realizovať tieto všeobecné scenáre:

Vytvorenie nového záznamu,
Aktualizácia existujúceho záznamu,
Vymazanie záznamu,
Zobrazenie prehľadu viacerých záznamov (spolu vybranou základnou štatistikou),
Zobrazenie konkrétneho záznamu,
Filtrovanie záznamov spĺňajúcich určité kritériá zadané používateľom.
Aplikácia môže mať konzolové alebo grafické rozhranie. Je dôležité aby scenáre boli realizované realisticky - teda aby aplikácia (a teda aj jej používateľské rozhranie) naozaj poskytovala časť funkcionality tak, ako by ju očakával zákazník v danej doméne.

Scenáre, ktoré menia dáta musia byť realizované s použitím transakcií a aspoň jeden z nich musí zahŕňať prácu s viacerými tabuľkami (typicky vytvorenie záznamu a naviazanie cudzieho kľúča).

Špecifikácia scenárov:

Vytvorenie nového záznamu:

1. Vytvorenie záznamu Pacient s jeho potrebnými údajmi. Potrebné je zadať všetky údaje ako meno, bydlisko, pohlavie, telefónne číslo, ktoré sú pre jednotlivého pacienta dôležité
2. Vytvorenie záznamu Návšteva pre pacienta s potrebnými údajmi, ako sú čas a dátum návštevy, popis a u akého lekára táto návšteva bola vykonaná. Dôležitý je aj údaj o dôvode návštevy diagnózu, ktorú lekár vyšetрил

Aktualizácia existujúceho záznamu:

1. Aktualizácia záznamu Pacient. Pacient si môže zmeniť meno, telefónne číslo, alebo sa presťahovať. Prípadne mohol lekár pri zápise pacienta urobiť chybu, ktorú je nutné opraviť

Vymazanie záznamu:

1. Vymazanie záznamu Pacient. Ak pacient už nechce chodiť do danej nemocnice a želá si, aby boli jeho údaje vymazané spolu so všetkými vyšetreniami.
2. Vymazanie záznamu Návšteva pre pacienta. Ak lekár urobí chybu, prípadne usúdi, že návštevu nebolo nutné vymazať.

Zobrazenie prehľadu viacerých záznamov (spolu vybranou základnou štatistikou):

1. Zobrazenie názvu diagnózy s počtom pacientov danou diagnózou trpiacich. Ak by vedenie nemocnice chcelo vedieť napríklad pre výskumné účely, aká je najrozšírenejšia diagnóza v ich nemocnici.

Zobrazenie konkrétneho záznamu

1. Zobrazenie záznamu Pacienta a jeho informácií. Ide o zobrazenie základných informácií o pacientovi spolu so všetkými záznamami napríklad pre občerstvenie pamäte

lekára pri príchode pacienta na vyšetrenie, ktoré pacient má. Keďže pacient môže mať veľa záznamov, z každého záznamu sa vypíše iba časť správy.

2. Zobrazenie návštevy. Vzhľadom na to, že v náhlade je iba časť správy, lekár si musí vedieť zobrazit' aj celú správu.

3. Zobrazenie všetkých pacientov. V prípade, že vedenie nemocnice chce vidieť všetkých pacientov, alebo lekár potrebuje nájsť pacienta a nevie jeho ID.

Filtrovanie záznamov spĺňajúcich určité kritériá zadané používateľom:

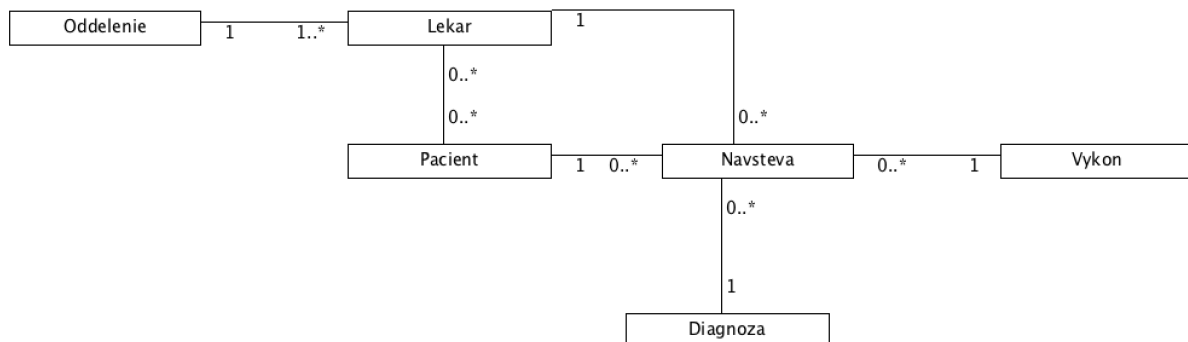
1. Zobrazenie registrovaných pacientov podľa mena lekára, ktorý danú registráciu urobil.

2. Zobrazenie vyšetrených pacientov podľa mena lekára, ktorý dané vyšetrenie urobil

NoSQL:

Použitá databáza Elasticsearch na vyhľadanie pacienta podľa jeho mena. Lekárovi sa môže stať, že si pacient zabudol doma svoju dokumentáciu so svojimi údajmi, preto lekár musí vedieť vyhľadať aj pacientov podľa mena, kde mu pri mene ukáže pacientove ID pre ďalšiu prácu.

Diagram logického návrhu:

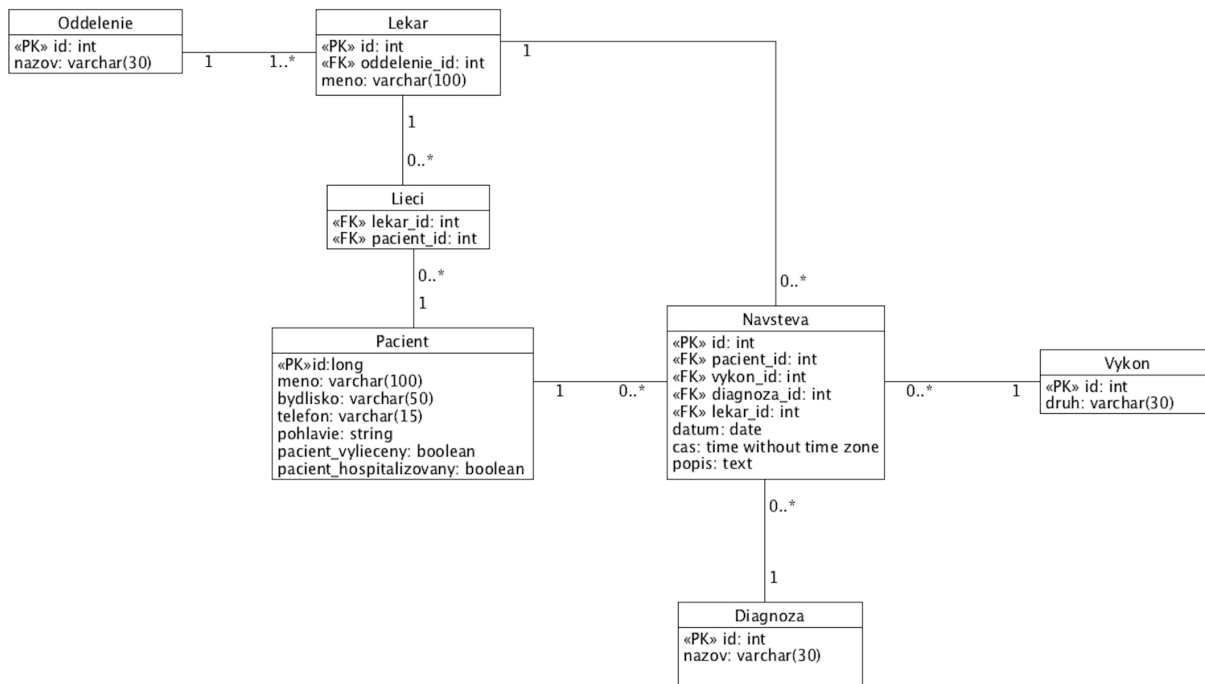


Model pozostáva zo vzťahu šiestich entít Oddelenie, Lekár, Pacient, Návšteva, Diagnóza, Výkon.

Vzťahy:

- Oddelenie - Lekár: V nemocnici na jednom oddelení pracuje viacero lekárov, pričom ale jeden lekár na základe svojej špecializácie pracuje len na jednom oddelení
- Lekár – Pacient: Keďže v jednej nemocnici je veľa oddelení, môže sa stať, že jeden pacient, ležal na viacerých oddeleniach a teda sa mu venovalo viacero lekárov. Prípadne sa mu venovalo viacero lekárov práve na jednom oddelení.
- Pacient – Návšteva – Lekár – Diagnóza – Výkon: Návšteva charakterizuje jednu návštevu u lekára. Pri každej návšteve je jeden lekár a jeden pacient, ktorý je lekárom vyšetrovaný pre určenie, alebo práve kvôli jednej diagnóze. Každá návšteva má jeden výkon, ktorý značí typ návštevy u lekára.

Diagram fyzického návrhu:



- 1- Oddelenie: Uchováva informácie: názov oddelenia a ID oddelenia:
- 2- Lekar: Uchováva informácie: meno lekára, ID oddelenia, na ktorom lekár pracuje a ID lekára
- 3- Pacient: Uchováva informácie: meno, bydlisko, telefón a pohlavie pacienta, informáciu, či je pacient chorý hospitalizovaný a ID pacienta
- 4- Lieci: obsahuje cudzí kľúč ako ID lekára a ID pacienta
- 5- Navsteva: obsahuje cudzí kľúč ako ID lekára, ID pacienta, ID výkonu, ID diagnózy. Uchováva informácie: o čase a dátume návštevy a popise.
- 6- Diagnóza: Uchováva informácie: názov diagnózy a ID diagnózy
- 7- Vykon: Uchováva informácie o: druh výkonu a ID výkonu

Stručný opis návrhu a implementácie:

Programovacie prostredie:

Ako programovacie prostredie som zvolil jazyk Java, implementovaný pomocou vývojového prostredia Eclipse Luna. Grafické rozhranie bolo realizované za pomoci rozšírenia Window Builder for Eclipse. Z kódu v jazyku Java som sa pripájal na PostgreSQL server, ktorý je vytvorený na adrese localhost pomocou aplikácie PostgreSQL.

Návrhové rozhodnutia:

Program sa k databáze pripojí okamžite po spustení aplikácie s uloženým menom a heslom v kóde aplikácie. Nie je to bezpečné rozhodnutie, ale aplikácia nie je pripojená na internet a slúži iba pre edukačné účely a vytvorenie školského projektu. Program ostáva pripojený k databáze počas celého behu programu. K potrebným údajom z databázy sa prístupuje cez metódy, ktoré obsahujú napísané skripty jazyka SQL, ktoré sú dynamické a využívajú potrebné údaje zo vstupov používateľa do aplikácie.

Opis implementácie jednotlivých scenárov:

Vytvorenie nového záznamu:

Používateľ v aplikácii vyplní potrebné polia s informáciami pre pacienta, alebo záznam, ktoré chce uložiť a stlačí príslušné tlačidlo pre vytvorenie nového pacienta, alebo záznamu. Aplikácia dané údaje použije a pomocou dynamicky napísaného skriptu INSERT INTO hodnoty uloží. Pri vytvorení nového záznamu pacient si užívateľ vyberá údaje aj z ComboBoxov a následne sa spustí zložený skript, kde sa najprv uložia potrebné údaje pacienta a následne do lieči, ktorá je väzobnou tabuľkou. Použitie ComboBoxov je výhodné z hľadiska ošetrovania. Sú v nich uložené hodnoty z databázy a užívateľ si vyberá práve tie a znižuje sa riziko pomýlenia sa pri vyplňaní.

Aktualizácia existujúceho záznamu:

Používateľ v aplikácii zvolí možnosť aktualizácie záznamu pacienta. Následne mu do textového poľa program vypíše údaje na aktualizáciu pomocou dynamického skriptu SELECT. Užívateľ údaje zmení a stlačí tlačidlo aktualizovať kde aplikácie následne údaje pomocou dynamického skriptu UPDATE zapíše do databázy

Vymazanie záznamu:

Používateľ v aplikácii pomocou textového poľa zadá ID údajá pacienta, alebo záznamu, ktorý chce vymazať. Aplikácia následne pomocou dynamického skriptu DELETE FROM pre vymazanie pacienta, alebo záznamu, údaje vymaže. Pri pacientovi je na FK nastavené ON DELETE CASCADE, teda sa vymažú aj všetky záznamy daného pacienta.

Zobrazenie prehľadu viacerých záznamov (spolu vybranou základnou štatistikou):

Po stlačení tlačidla v aplikácii sa vykoná skript zložený z vytvorenia dočasnej tabuľky, kde sa uložia hodnoty zo skriptu SELECT pre nájdenie potrebných hodnôt. Z dočasnej tabuľky sa vyberú informácie pre výpis pomocou funkcie SELECT.

Zobrazenie konkrétneho záznamu

1. Zobrazenie záznamu Pacienta a jeho informácií.

Používateľ zadá požadované ID pacienta a stlačí tlačidlo pre zobrazenie záznamu. Aplikácia pomocou dynamického skriptu SELECT vypíše potrebné údaje.

2. Zobrazenie návštevy.

Používateľ zadá požadované ID návštevy a stlačí tlačidlo pre jej zobrazenie. Aplikácia pomocou dynamického skriptu SELECT vypíše text lekárskej správy.

Filtrovanie záznamov spĺňajúcich určité kritériá zadané používateľom:

Používateľ pomocou ComboBoxu vyberie meno lekára a stlačí príslušné tlačidlo na výpis. Aplikácia pomocou SELECTU vypíše požadované údaje.

NoSQL:

Pomocou ElasticSearch databázy sa na základe mena dá vyhľadať pacient. Stačí aj keď sa zadá len časť mena, program vráti každú zhodu. Samozrejmosťou je synchronizácia SQL a ElasticSearch pri pridávaní pacienta, mazaní a aktualizácií. V prípade výpadku služby ElasticSearch sa dáta neuložia do SQL databázy a používateľ bude o tejto skutočnosti informovaný.