**MODELIRANJE I ADMINISTRACIJA BAZA PODATAKA**

- UPUTE ZA IZRADU I OBRANU PROJEKTA -

Projekt iz kolegija „Modeliranje i administracija baza podataka“ treba biti aplikacija sa bazom podataka. Aplikacija treba omogućiti unos, čitanje te analizu unesenih podataka, pri tome demonstrirajući pozive uskladištenih procedura i funkcija te izvršavanje okidača.

Primjer: FINANCIJSKI MANAGER

* Aplikacija treba omogućiti unos izdanih računa s popisom svih stavki na pojedinom računu, njihovim količinama i cijenom, te podacima o samom kupcu.
* Aplikacija treba prikazati određenu analizu unesenih podataka. Primjerice, koji kupac najčešće kupuje u kojem dućanu? Što najčešće kupuje u nekom vremenskom razdoblju? Koji tip artikla (proizvoda) najčešće kupuju svi postojeći kupci? U kojem dućanu su određeni tipovi artikala najjeftini/najskuplji.

Da bi se napravio ovaj projekt treba ga razdijeliti u više faza:

1. **Analiza potreba**

Potrebe u pravilu definira klijent, no u ovom slučaju vi ćete sami odlučiti o temi vašeg projekta. Primjerice, to može biti nešto što bi vama osobno bilo zanimljivo ili korisno (poput možda prethodnog primjera financijskog managera), ili nešto što možete iskoristiti i u tvrtki u kojoj radite, gdje bi i njima mogli ponuditi vaš rad kao nešto konkretno i korisno.

1. **Konceptualni ER model baze (maks. 4 boda)**

Kada se odlučite na temu potrebno je napraviti konceptualni ER (*Entity-Relationship*) model baze podataka koji treba biti pregledan i prihvaćen od strane predmetnog nastavnika prije daljnjeg nastavka razvijanja projekta. Za izradu konceptualnog ER modela možete upotrijebiti alate poput MS Visio ili Enterprise Architect. Također, na Internetu postoje već gotovi besplatni alati poput <https://www.draw.io/>

**Napomena:** konceptualni podatkovni model mora sadržavati minimalno 5 entiteta te minimalno jednu „više-prema-više“ vezu. Za maksimalni broj bodova potrebno je imati minimalno 8 entiteta.

1. **Fizički podatkovni model baze podataka (maks. 5 bodova)**

Veze među tablicama te atribute (stupce) sa primarnim i stranim ključevima možete prikazati i opisati fizičkim podatkovnim modelom baze podataka. Logički ER podatkovni model nije potrebno izrađivati.

1. **Ostale potrebe prema bazi podataka**

Već pri izradi fizičkog podatkovnog modela u prethodnoj točki trebate se odlučiti na tip baze podataka. U odabiru baze nema ograničenja, osim što odabrana baza podataka mora podržavati sve funkcionalnosti tražene u ovom projektu. Zato kao preporuku predlažem MS SQL Server. Besplatno možete preuzeti punu Standard/Enterprise inačicu SQL Servera preko MSDNAA-a. Express inačica neće biti dovoljna!

* *Sadržaj baze podataka*
  + **Indeksi i ograničenja (maks. 3 boda)**

U tablici po izboru kreirati minimalno dva ne-grupirajuća indeksa, od kojih jedan mora biti filtrirajući, a drugi ne smije dopustiti duplikate u odabranom stupcu. Također, kreirati minimalno jedno ograničenje (objekt tipa *constraint*) nad tablicom po izboru.

* + **Uskladištene procedure (stored procedure) i funkcije (maks. 5 bodova)**

Vaša aplikacija mora analizirati i prikazati podatke pozivajući uskladištene procedure koje se trebaju nalaziti u bazi podataka. U bazi podataka moraju postojati barem četiri uskladištene procedure od kojih barem jedna od njih poziva jednu vašu skalarnu (ili neku drugu) funkciju.

* + **Okidači (triggers), (maks. 4 boda)**

Kreirati minimalno 2 DML, DDL i/ili CLR okidača.

* + **Sheme (maks. 2 boda)**

Sve matične tablice (šifrarnici) trebaju biti unutar sheme **Sifrarnik**, dok prometne tablice u shemi **IO**.

* + **Korisnički računi i uloge (2 boda)**

Baza podataka treba sadržavati korisnički račun **DB\_Admin** za administriranje baze podataka, te treba biti član nove uloge baze podataka (*database role*) imena **DB\_Administratori**. Svi članovi te grupe moraju imati sve ovlasti u bazi podataka.

* + **Aplikacijska uloga (5 bodova)**

Baza podataka treba sadržavati aplikacijsku ulogu **APP\_Uloga** za autorizaciju vaše aplikacije. Navedena aplikacijska uloga ima sve ovlasti nad shemama *Sifrarnik* i *IO*, osim brisanja podataka iz jedne tablice vama po izboru. Demonstrirajte autorizaciju vaše aplikacije korištenjem aplikacijske uloge.

* + **Impersonalizacija korisničkog računa (4 boda)**

Demonstrirajte autorizaciju vaše aplikacije korištenjem impersonalizacije korisničkog računa.

* + **Kriptografija (5 bodova)**

Demonstrirajte šifriranje stupca po izboru korištenjem kombinacije simetričnih i asimetričnih algoritama (ključ simetričnog ključa zaštitite asimetričnim algoritmom). Za postupak šifriranja i dešifriranja kreirajte dvije zasebne uskladištene procedure 'sifriraj' i 'desifriraj'.

* + **Dinamičko maskiranje podataka (2 boda)**

Demonstrirajte dinamičko maskiranje podataka u tablici po izboru na barem 2 stupca različitog tipa (npr. email, broj kartice itd.).

* + **Nadgledanje rada SQL Server instance i baze podataka – SQL Server Audit (4 boda)**

Demonstrirajte nadgledanje rada baze podataka i instance SQL Servera pomoću dva različita SQL Server audit objekta.

* + **Nadgledanje rada SQL Server instance i baze podataka – SQL Server Profiler (2 boda)**

Na proizvoljnom primjeru uz pomoć SQL Server Profiler alata demonstrirajte nadgledanje rada proizvoljno odabrane SQL Server baze podataka.

* + **Povrat (restore) baze podataka (5 boda)**

Demonstrirajte (uživo, prilikom obrane projekta) povrat baze podataka na način da lanac rezervnih kopija sadrži minimalno dvije rezervne kopije dnevnika transakcija, jednu diferencijalnu i jednu punu rezervnu kopiju.

* + **Plan održavanja baze podataka (3 boda)**

Kreirajte plan održavanja baze podataka (eng. Maintenance plan) koji se sastoji od barem 3 različita SQL Server posla, te se periodički izvršava barem jednom tjedno.

* + **Replikacija (5 boda)**

Demonstrirajte transakcijsku replikaciju između dvije različite instance SQL Servera.

**NAPOMENA:** U projektnom radu studenti ne smiju koristiti iste ili tek minimalno modificirane primjere iz priručnika, već je sukladno namjeni baze podataka potrebno kreirati vlastite (originalne) primjere uskladištenih procedura, okidača i ostalih traženih dijelova projekta!

1. **Klijent aplikacija**

Klijent aplikacija može biti pisana u bilo kojem programskom jeziku, te može biti za bilo koju platformu ili uređaj. U komunikaciji s bazom podataka aplikacija treba demonstrirati pozive uskladištenih procedura i funkcija.

1. **Obrana projekta (postupak)**

1. Student/tim mailom šalje konceptualni podatkovni model na odobrenje.

2. Nakon odobrenja konceptualnog podatkovnog modela student/tim pristupa izradi ostatka projekta sukladno objavljenim uputama.

3. Nakon što je student/tim završio projekt e-mailom dogovaramo termin obrane projekta.

4. Nakon obrane projekta student bira i prijavljuje jedan od raspoloživih ispitnih rokova za polaganje teorijskog dijela ispita u LMS sustavu.

5. Konačna ocjena ovisi o broju skupljenih bodova na projektu i teorijskom dijelu ispita, sukladno već objavljenim pravilima.

Za točke 1, 2 i 3 ne postoji vremenski rok - studenti se mogu javiti tokom cijele akademske godine. Međutim, teorijski dio ispita isključivo je moguć u službenim terminima ispitnih rokova koji će naknadno biti poznati (prvi ispitni rok očekuje se u veljači).

U slučaju pitanja i nejasnoća slobodno se javite mailom (zeljko.kovacevic@tvz.hr).