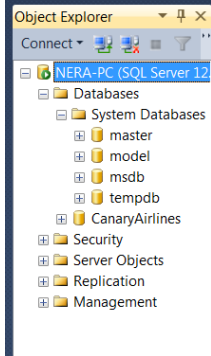


Врсте системских и корисничких db

Постоји пет врста системских db:



Master, чува све информације потребне за кориснички SQLS за добро функционисање, ту се смештају login и server конфигурације, ако ова db није доступна, SQLS неће моћи да се стартује;

MSDB, који чува детаље око послова извршених од стране SQLS Агента, бекапује детаље и историју рада, није неопходна за нормално функционисање SQLS;

Model, је template db и свака db креирана од стране корисника се стартује као копија ове db;

Resource, ова се једина може видети преко ОЕ, чува системске објекте као што су sys табеле; TempDB, њој приступају сви корисници SQLS, у њој се креирају сви привремени објекти, она се ствара и брише сваки пут када се сервер стартује.

Са административним дозволама корисник може мењати било коју db осим Resource. Препорука је да се не мењају ниједна db сем можда Модел. Једино Resource се не види на листи системских база података.

Програмери db највише времена проводе креирајући објекте унутар корисничких db.

Креирање база података се може обавити на два начина: кроз SSMS или коришћењем скрипти.

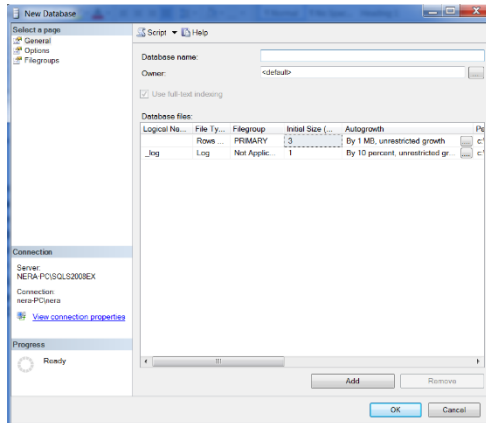
Креирање нове базе података

Прво се креира следећи фолдер у руту рачунара: C:\temp\ime i prezime usenika\sqlsnove.

Сваки ученик треба да креира свој фолдер са сопственим именом и презименом (нпр Ivan_Ivanovic) унутар фолдера temp а као подфолдер фолдер sqlsnove.

Фајлови који се креирају у Студију се називају скрипте и оне омогућавају креирање и уништавање db које ће се направити.

Пример03: Креирање нове базе података помоћу Студија.



Кликом на десно дугме миша на Databases и на падајућем менију се бира New Database. Тако се појављује нови прозор New Database.

Тражи се име нове базе података. Име увек треба да опише намену базе података. Администратори базе података (DBA) одржавају рад db, тестирају базу податка и издају скрипте. Developers пишу SQL код који се стартује у SQLS.

Циљ практичног рада на часовима ће бити креирање и модификовање базе података која треба да садржи једноставну базу података адреса. Зато ће се нова база података назвати Adresar.

Имена db могу да садрже специјалне знаке, бројеве и празна места. Ипак то није добра идеја пошто постоји опасност од некомпатабилности са старијим верзијама база података.

Примери04 давања имена базама података помоћу упита:

```
SELECT * FROM Adresar.dbo.Kontakti
```

```
SELECT * FROM [Knjiga Adresara].dbo.Kontakti
```

У другом примеру се види да ако хоћемо да имамо празно место у називу, мора се цело име базе података ставити у угласту заграду. Најбољи начин је да ако хоћемо да имамо у имену две или више речи да их пишемо у CamelCase стилу: KnjigaAdresara.

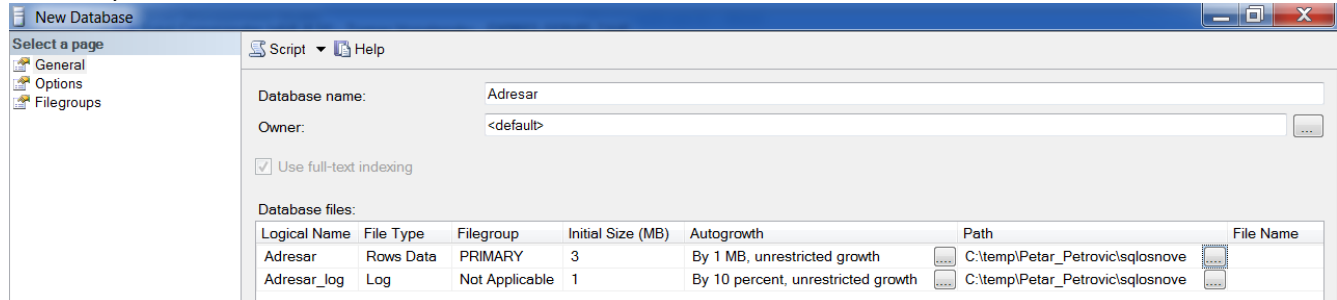
*Откуцати Adresar као име базе података.

У следећем дијалог боксу је могуће одредити ко је власник ове db.

По аутоматизму је постављено на дифолт, што значи да је особа која креира db истовремено и власник db.

Овај податак се преводи у објекат базе података и за сада то не треба мењати.

Види се да је после куцања имена db дошло до аутоматског попуњавања табеле испод у којој се налазе фајлови који чине нову db.



Колоне у овој табели су:

- **Logical Name:** име додељено фајлу за лакшу корисничку идентификацију
- **File Type:** постоји неколико али су најчешћи фајлови са подацима и log фајлови
- **Filegroup:** може да садржи више фајлова и користи се за дељење db на диску
- **Initial Size (MB):** колика треба да буде величина фајла када се креира и зависи од тога колико очекујемо да буде информација у db, реална величина може да се разликује од дефинисане
- **Autogrowth/Maxsize:** ако фајл достигне додељену иницијалну величину SQLS ће користити ове величине за повећање фајла и за његово ограничавање
- **Path:** фолдер са db
- **File Name:** право име физичког фајла што ће се аутоматски направити из имена db

*Променити path у онај са слике примера.

Проверити на нод Databases да ли постоји Adresar db док су у фолдеру sqlslove два фајла Adresar.mdf и Adresar_log.ldf.

MDF фајл је Master Data File и ту ће SQLS смештати корисничке податке.

LDF фајл је Log фајл који SQLS користи за одржавање db.

Постоји и трећи тип фајла Secondary Data File (NDF) који се креирају са filegroup. Користе се за побољшање перформанси.