

Met dbMaestro foutloze en zorgeloze deployment

Nieuwe versie database-schema: geen paniek

Ben Suurmeijer

Bent u ontspannen als u een nieuwe versie van het database-schema moet uitrollen naar productie? Hoe vaak gaat dit zonder problemen? Hoe vaak moest u naar het hoofdstuk 'rollback' van de change request? Ik wil de vinger niet te veel op de zere plek leggen, en in de gesprekken die ik heb rond dit thema krijg ik niet vaak een duidelijk antwoord op deze vragen, maar ik krijg wel de bevestiging dat het regelmatig niet zonder problemen verloopt.

Dit probleem is toch relatief eenvoudig te voorkomen. Het begint met goed versiebeheer van het schema. En de mogelijkheid om schema's te vergelijken om het verschil te bepalen. Daarbij is belangrijk dat niet alleen de schema's met elkaar worden vergeleken, maar ook wat de verschillen zijn ten opzichte van de baseline. Waarom? Dat leest u verderop.

Versiebeheer

Natuurlijk, om versies te vergelijken moet u ook versies bewaren. Maar hoe bewaart u de versies, als files? Een export van de DDL? En hoe bepaalt u bij deployment of een update van een object mogelijk is, of dat het object (opnieuw) gecreëerd moet worden?

In de software-ontwikkeling zijn er voldoende oplossingen aanwezig die dit regelen: SVN, Subversion, Rational TeamConcert of ClearQuest om er een paar te noemen. Dit zijn hele goede

oplossingen voor software-ontwikkeling. Maar een database of databaseschema is niet te vergelijken met software. Software bestaat uit verschillende files, en Oracle-objecten zijn geen files.

Tussentijdse veranderingen

Hoe vaak gebeurt het dat er in productie 'snel' even iets wordt gewijzigd. Allemaal met goede redenen natuurlijk. Als er 10, 100 of nog meer gebruikers op wachten, dan wordt de druk natuurlijk hoog. Maar hoe vaak gebeurt het dan dat zo'n verandering bij de uitrol van een nieuwe versie weer 'verdwijnt'? Dit kan eenvoudig worden voorkomen. Als de ontwikkeling van een nieuwe versie start, legt u het startpunt vast, de *baseline*. Als u na een tijdje zo ver bent om van ontwikkeling naar test, naar QA en uiteindelijk naar productie te gaan, vergelijkt u niet alleen de schema's onderling, maar ook beide met de baseline, zie afbeelding 1. Vergelijk niet alleen bron met doel, maar ook met baseline. Hiermee wordt duidelijk wat er in bron en doel is gewijzigd ten opzichte van de baseline. Als één object in zowel bron als doel is gewijzigd, levert dit een conflict op, zie afbeelding 2.

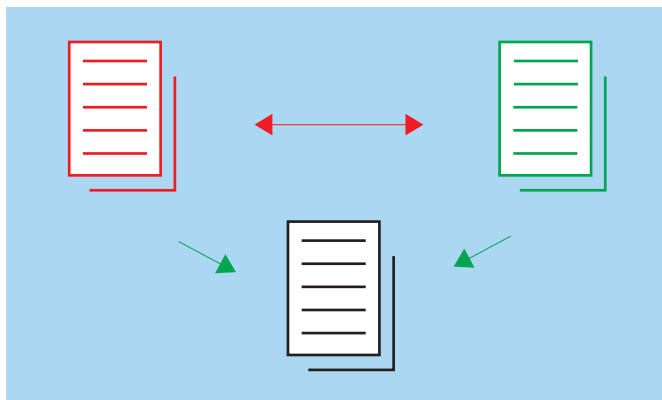
Uitrol: paniek?

Het nieuwe schema is klaar. De bijbehorende software is ook klaar, het is getest in Test & Acceptatie, de change is goedgekeurd, de risico's zijn bekend. Slaat de paniek nu toe? Of bent u gerust op de uitkomsten?

Het is belangrijk dat het script voor de upgrade gaat werken. Dat objecten in de juiste volgorde worden aangemaakt, constraints worden verwijderd en weer aangemaakt, indien een table niet 'eenvoudig' kan worden aangepast, er een nieuwe versie wordt gemaakt en de data gekopieerd worden, en dan kan de oude versie verwijderd worden. Is dat allemaal geregeld? Maar het werkte op de Acceptatie-omgeving zult u zeggen. Maar was de Acceptatie-omgeving gelijk aan de productie-omgeving, waren er geen wijzigingen in productie? Kortom, gaat het script dat werkte voor Acceptatie ook werken op Productie?

dbMaestro TeamWork

Teamwork van dbMaestro maakt het mogelijk om database-ontwikkeling een integraal onderdeel van een gedisciplineerd SCM-proces te maken in plaats van een handmatig proces of, op zijn best, semigeautomatiseerd proces. TeamWork voldoet in



Afbeelding 1: Vergelijk met de baseline.

Object Name	Object Type	Source vs BaseLine	Destination vs BaseLine	Action
DEPARTMENTS	TABLE	Sub Objects were chang...	Same As Base Line	Override with s...
EMPLOYEES	TABLE	Sub Objects were chang...	Same As Base Line	Override with s...
EMPLOYEES_PRIVILEG...	TABLE	Sub Objects were chang...	Object was dropped	Override with s...
JOB	TABLE	Same As Base Line	Sub Objects were chang...	Ignore source ...
LOCATIONS	TABLE	Object was changed	Same As Base Line	Override with s...
REGIONS	TABLE	Object was changed	Same As Base Line	Override with s...
TEST_TABLE	TABLE	Same As Base Line	Object was dropped	Override with s...
SECURE_DML	PROCEDURE	Object was changed	Object was changed	Merge conflicts
MV_JOBS	MATERIALIZED ...	Same As Base Line	Object was dropped	Override with s...

Afbeelding 2: Vergelijking tussen bron, doel en baseline detecteert een conflict.

alle opzichten aan de eisen voor een nieuwe generatie database change management oplossingen, en vergroot in belangrijke mate de voordelen voor organisaties die een SCM-proces volgen voor zowel software-ontwikkeling als ook database-ontwikkeling.

Veranderingen in databases kunnen nu op eenzelfde manier gevolgd worden als veranderingen in de softwarecode, waardoor het complete ontwikkelproject, inclusief database-veranderingen, op een gecontroleerde, geregistreerde en gestandaardiseerde wijze uitgerold kan worden. TeamWork dwingt dit proces af, en maakt teamwerk mogelijk voor beheerste omgevingen, door middel van check-out en check-in. Indien een object niet uitgecheckt is, zal een poging tot verandering tot een foutmelding leiden. En de 'deployment manager' vergelijkt de versies, niet alleen met elkaar maar ook met de baseline. Het belang daarvan is eerder uitgelegd.

Conclusie

Software-ontwikkelaars hebben meer dan twintig jaar de voordelen ervaren van het gebruik van SCM-oplossingen. Het is nu tijd om een database change management proces naar hetzelfde niveau te tillen als een SCM-proces, en database-ontwikkelaars en DBA's dezelfde mate van eenvoud en procesmatige beveiliging te bieden bij database-ontwikkeling.

Ben Suurmeijer is zelfstanding pre sales consultant.