Dimensionel modellering for Nedrivning

Indhold

[1. Teknisk design 3](#_Toc156550740)

[1.1 Udtrækning 3](#_Toc156550741)

[1.2 Dimensionel modellering 4](#_Toc156550742)

[1.3 Historisk logning 4](#_Toc156550743)

[1.3.1 DW\_ValidFrom og DW\_ValidTo 4](#_Toc156550744)

[1.3.2 DW\_IsCurrent 5](#_Toc156550745)

[1.3.3 DW\_MostRecentOnday 5](#_Toc156550746)

# 1. Teknisk design

Overordnet set består det tekniske design af tre dele per miljø, som illustreret i figuren ovenfor:

1. Først trækkes der rådata fra driftssystemet (Nedrivning) over til Rapport-databasen på CB-DW.
2. Derefter bliver data forarbejdet for at opfylde et rapporteringsbehov.
3. Resultatet er en datamodel, hvor entiteterne og kolonnerne har meningsfyldte navne og entiteterne er entydigt relateret til hinanden.

## 1.1 Udtrækning

Følgende tabeller hentes fra Nedrivning databasen, hvor der ikke er teknisk opsatte relationer mellem dem og hvor data foreligger i et normaliseret skema.

|  |
| --- |
| **Tabel navn** |
| Adresser |
| Brevproces |
| Faser |
| Intern\_screening |
| Maengdeopgoerelse |
| Miljoeproces |
| Miljoeraadgiverdelaftale |
| Minkavlere |
| Minkavlere\_Minkfarme |
| MinkavlersRaadgiver |
| Minkfarme |
| Minkfarme\_Faser |
| Minkfarme\_Processer |
| Myndighed |
| Nedriver\_Kontaktpersoner |
| Nedrivere |
| Nedrivergrupper |
| Nedrivergrupper\_Nedrivere |
| Nedrivningsproces |
| Nedrivningstilladelse |
| Projektledere |
| Sagsstatus |
| TekniskRaadgiver\_Kontaktpersoner |
| TekniskRaadgivere |
| TekniskRaadgivergrupper |

## 1.2 Dimensionel modellering

For alle tabellerne gælder det, at det skal være hurtigere og nemmere for brugeren at arbejde med tabellerne. Data fra enkelte tabeller samles og sammensættes på anden vis end de er i rådata, så forretningsprocesserne overføres korrekt i dimensionelle tabeller.

For hver eneste rapporteringsbehov/måling bliver der ligeledes bygget en fact-tabel, der sammen med dimensions-tabellerne udgør et stjerne-skema, som er velegnet til brug af analyse- og rapporteringsbehov.

De normaliserede tabeller der er specificeret i afsnit 1.1 grupperes til større tabeller, så det ikke er relevant for brugeren at forstå datamodellen og samtidig minimere det antallet af joins.

*Dokumentationen for dimensionelle, fact og stjerneskemaet findes i GitHub / Draw.io.*

## 1.3 Historisk logning

Der er blevet brugt Type 2 SCD (Slowly Changing Dimensions) for at bibeholde hele historikken fordelt på pågældende primær- og historiktabel.

Historikken skabes i Nedrivning databasen og overføres til rapportlaget, så der ikke opstår en forskel mellem disse to.

Princippet for historiklogning er ens på tværs af entiteterne.

### 1.3.1 DW\_ValidFrom og DW\_ValidTo

Den historiske logning finder sted ved at der i hver række i de tabellerne eksisterer et ”DW\_ValidFrom” og ”DW\_ValidTo” felt.

Hvis man eksempelvis har tre minkfarme vil data første gang ovenstående job kører komme til at se således ud:

|  |  |  |  |
| --- | --- | --- | --- |
| **CHR** | **Nedrivergruppe\_id** | **DW\_ValidFrom** | **DW\_ValidTo** |
| 93 | 1 | 01-01-2021 | 31-12-9999 |
| 253 | 2 | 01-01-2021 | 31-12-9999 |
| 88784 | 1 | 01-01-2021 | 31-12-9999 |

Lad os sige at en bruger den 10. januar ændrer. Næste gang vil indholdet af tabellen dermed blive:

|  |  |  |  |
| --- | --- | --- | --- |
| **CHR** | **Nedrivergruppe\_id** | **DW\_ValidFrom** | **DW\_ValidTo** |
| 93 | 1 | 01-01-2021 | 31-12-9999 |
| 253 | 2 | 01-01-2021 | 31-12-9999 |
| 88784 | 1 | **01-01-2021** | **10-01-2021** |
| 88784 | 2 | **10-01-2021** | **31-12-9999** |

Ved at denormalisere tabellerne samles historikken fra flere tabeller i én. I stedet for at inkludere alle dato kolonnerne fra tabellerne, samles værdierne af disse i en samlet kolonne.

Logikken for historikken formateres således, at

DW\_ValidFrom beholder pr. række den seneste ændring for de tilknyttede tabeller.

DW\_ValidTo beholder pr. række den tidligste ændring for de tilknyttede tabeller.

Ved at samle to tabeller og deres historik vil tabellen se således ud:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabel 1 Gældende\_fra** | **Tabel 1**  **Gældende\_til** | **Tabel 2**  **Gældende\_fra** | **Tabel 2**  **Gældende\_til** | **DW\_ValidFrom DW\_ValidTo** |
| **01-01-2021** | **01-02-2021** | 01-01-2021 | 31-12-9999 | **01-01-2021** **01-02-2021** |
| **01-02-2021** | **01-03-2021** | 01-01-2021 | 31-12-9999 | **01-02-2021 01-03-2021** |
| **01-03-2021** | **31-12-9999** | 01-01-2021 | 31-12-9999 | **01-03-2021 31-12-9999** |

Konsekvensen af denne logik vil betyde, at man ikke får den seneste for ændring for (i dette tilfælde) tabel 2, men at historikken fra tabel 1 kommer til at repræsentere rækken.

Når der findes en procestabel, så anvendes denne historik og der ses bort fra tidligere logik. Det skyldes, at denne tabel afspejler forretningslogikken. *Det betyder så, at der ses bort fra historik på opslagstabellerne efter ønske fra brugerne.*

### 1.3.2 DW\_IsCurrent

For at kunne finde seneste registrering i data er der bygget en logik i kolonnen "DW\_IsCurrent".

Hvis der sættes et filter på DW\_IsCurrent = '1' findes seneste registrering.

DW\_IsCurrent = '0' finder historiske registreringer.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabel 1 Gældende\_fra** | **Tabel 1**  **Gældende\_til** | **Tabel 2**  **Gældende\_fra** | **Tabel 2**  **Gældende\_til** | **DW\_ValidFrom DW\_ValidTo DW\_IsCurrent** |
| 01-01-2021 | 01-02-2021 | 01-01-2021 | 31-12-9999 | 01-01-2021 01-02-2021 **0** |
| 01-02-2021 | 01-03-2021 | 01-01-2021 | 31-12-9999 | 01-02-2021 01-03-2021 **0** |
| 01-03-2021 | 31-12-9999 | 01-01-2021 | 31-12-9999 | 01-03-2021 31-12-9999  **1** |

For at kunne finde det nuværende billede af data skal der derfor altid DW\_IsCurrent = '1' på alle tilknyttede dimensionelle tabeller.

### 1.3.3 DW\_MostRecentOnday

Ændringer bliver registreret helt ned på klokkeslættet. Det vil sige, at der kan være flere ændringer i løbet af dagen. For, at nå til enighed om resultatet på dagen, er der implementeret en kolonne "DW\_MostRecentOnDay" der definere den seneste ændring på dagen.

Eksempelvis når samme minkfarm har 4 forskellige ændringer på samme dag, så viser DW\_MostRecentOnDay = 1 (Den seneste ændring på dagen) DW\_MostRecentOnDay = 0 (De tidligere ændringer på dagen)

|  |  |  |  |
| --- | --- | --- | --- |
| **CHR** | **Nedrivergruppe\_id** | **DW\_ValidFrom** | **DW\_ValidTo DW\_MostRecentOnDay** |
| 88784 | 1 | 01-01-2021 01:00:00 | 10-01-2021 01:00:01 **0** |
| 88784 | 2 | 01-01-2021 01:00:01 | 10-01-2021 01:00:02 **0** |
| 88784 | 3 | 01-01-2021 01:00:02 | 10-01-2021 01:00:03  **0** |
| 88784 | 4 | 01-01-2021 01:00:03 | 10-01-2021 01:00:04 **1** |