**Rapport 2**

**Jonathan Meyer, Sergio Fenoll,**

**Dawid Miroyan, Cedric De Schepper**

**25 april 2018**

## 1. Inleiding

### Geïmplementeerde vereisten

Alle basisvereisten zijn geïmplementeerd.

User service:

* Gebruikers registreren (Iedereen)
* Gebruikers aanmelden (Iedereen)
* Administrator beheert gebruikers (Cedric & Sergio)

Data-import service:

* CSV-files importeren (Jonathan)
* ZIP-files importeren (Jonathan)
* Dump-files importeren (Sergio)
* Tabellen joinen op kolom(men) (Dawid)
* Metadata (Jonathan & Cedric)
* Data-sharing (Jonathan)

Data-transform service:

* Data selecteren voor opkuis (Iedereen)
* Type aanpassen (Sergio)
* Sorteren (Sergio)
* Data verwijderen op predicaat (Jonathan)
* Imputeren van ontbrekende data (Cedric)
* Historiek (Sergio)
* Transformaties:
  + Verwijderen van kolommen (Sergio)
  + Find-and-replace (alles) (Cedric)
  + One Hot Encoding (Dawid)
  + Numerieke transformaties (Sergio)
  + Datum/Tijd (Cedric)

View-service

* Basis view (Iedereen)
* Grafieken (Sergio)
* Statistieken (Cedric)
* Data downloaden (Jonathan)
* Berichten aan user (Cedric)

### 

### Software Stack

* *VCS*
  + *Git*
  + *Github (publieke repository)*
  + *Travis*
* *Hosting*
  + *‘Server’ bij Sergio thuis*
  + *NGINX*
  + *Gunicorn*
* *Frameworks/libraries*
  + *Flask*
  + *Flask-login*
  + *Jinja2*
  + *SQLAlchemy*
  + *Pandas*
  + *Numpy*

### 

### Specialisatie

Aangezien we als team weinig individuele ervaring hebben met de technologieën, hebben we besloten om geen zeer duidelijk onderscheid te maken in wie wat voor taken onderneemt.   
Momenteel hebben we die werkwijze dus aangenomen, waardoor elk teamlid zowel aan de front- als backend heeft gewerkt.

Gedurende de ‘eerste fase’ van het project (d.i. de periode tot de eerste tussentijdse evaluatie) heeft Sergio echter voornamelijk gezorgd voor de frontend (of toch, de uiteindelijke), terwijl Jonathan en Cedric zich meer bezig hebben gehouden met backend. Dawid heeft voornamelijk unit tests voorzien en aan debugging voor de bestaande code gedaan.

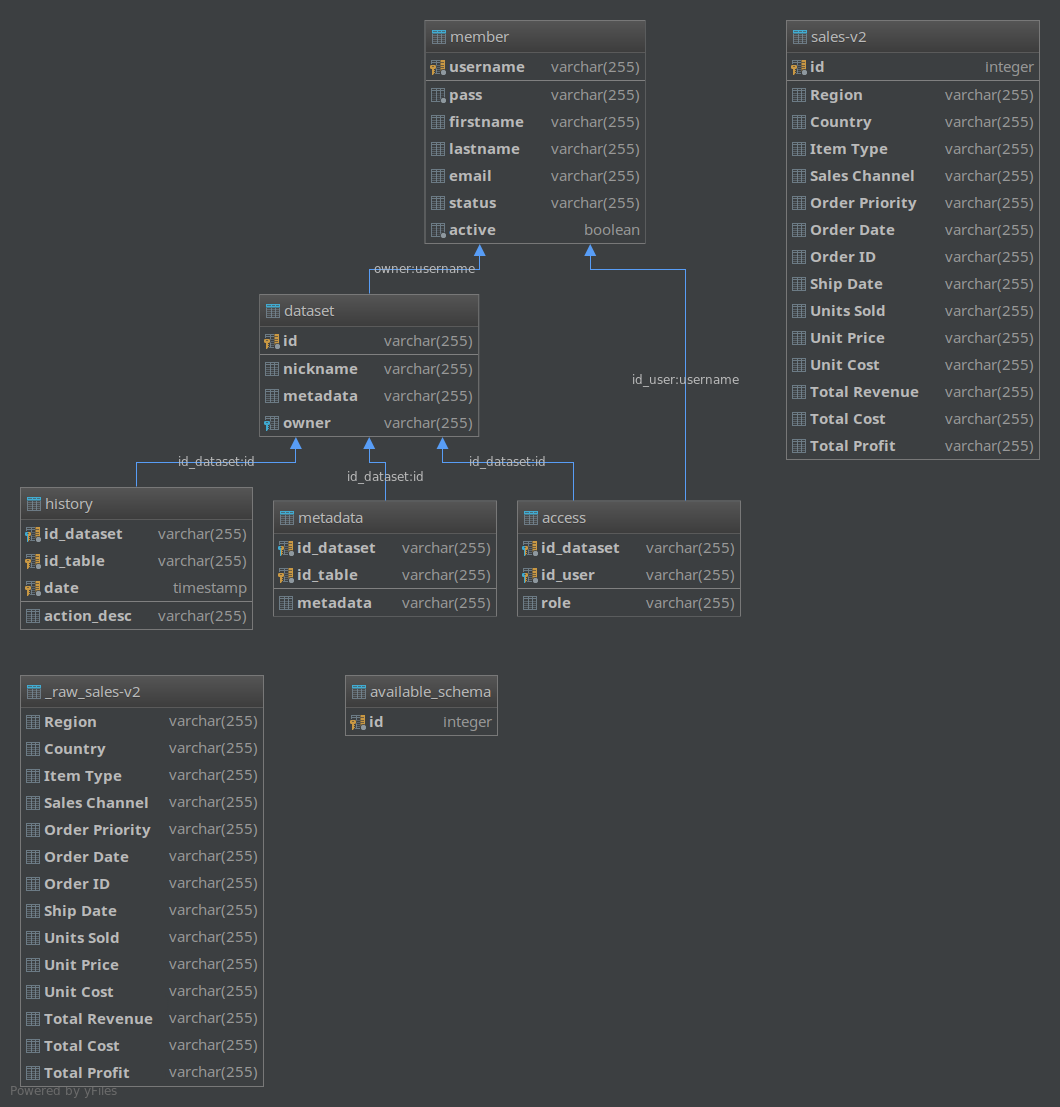
Hierna is de scheiding tussen front- en backend nog vervaagd, en hebben we per toe te voegen functionaliteit beslist wie daarvoor verantwoordelijk zou zijn. Die persoon nam dan het geheel voor zijn rekening, indien mogelijk, met hier en daar hulp van andere teamleden.  
Gedurende het hele project hebben we de taakverdeling geregeld via Github issues, zodat we niet in elkaars vaarwater komen.  
Exacte details zijn dan ook daar beschikbaar (zie link onderaan).

## 

## 2. Design

### Databank ontwerp

*Figuur: ER-diagram van de huidige database*

**

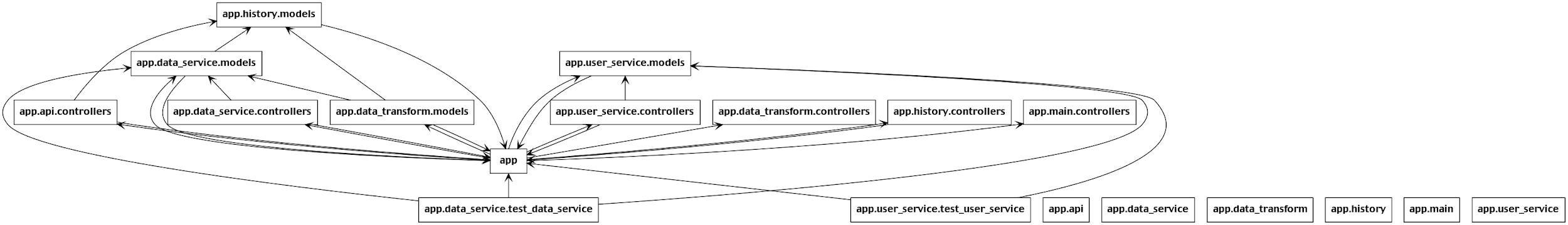
Ons huidig database model laat toe:

* gebruikers toe te voegen,
* datasets bij te houden,
* metadata over datasets op te slaan,
* te beheren welke gebruikers toegang hebben tot welke datasets,
* een ‘geschiedenis’ bij te houden met informatie over  
  de gebruikte transformaties op tabellen

Verder kunnen gebruikers data uploaden en opslaan in de database. De datasets komen overeen met schema’s in de database, en data van gebruikers wordt als nieuwe tabellen aan de bijbehorende schema’s toegevoegd. Per schema wordt ook een tabel bijgehouden waarin de metadata voor de data van de gebruiker wordt opgeslagen. In de ‘History’ tabel staat per tabel voor elke gebruikte transformatie of actie een beschrijving en timestamp. De tabel ‘available\_schema’ wordt gebruikt om beschikbare namen voor datasets (schemas) bij te houden. Intern worden deze gewoon oplopend genummerd, en om iets efficiënter om te gaan met deze nummers en te vermijden dat er conflicten optreden bij het verwijderen en daarna terug toevoegen van datasets, houden we ‘verwijderde’ namen bij voor hergebruik.

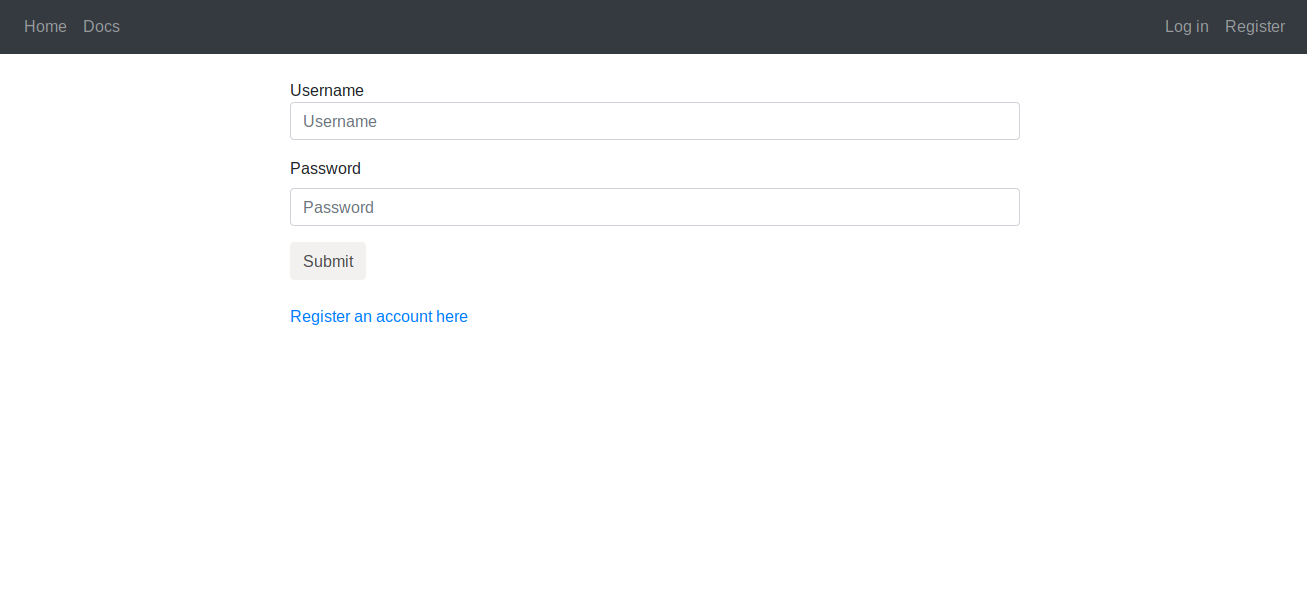
### Programma ontwerp

*Alle geïmplementeerde klassen*

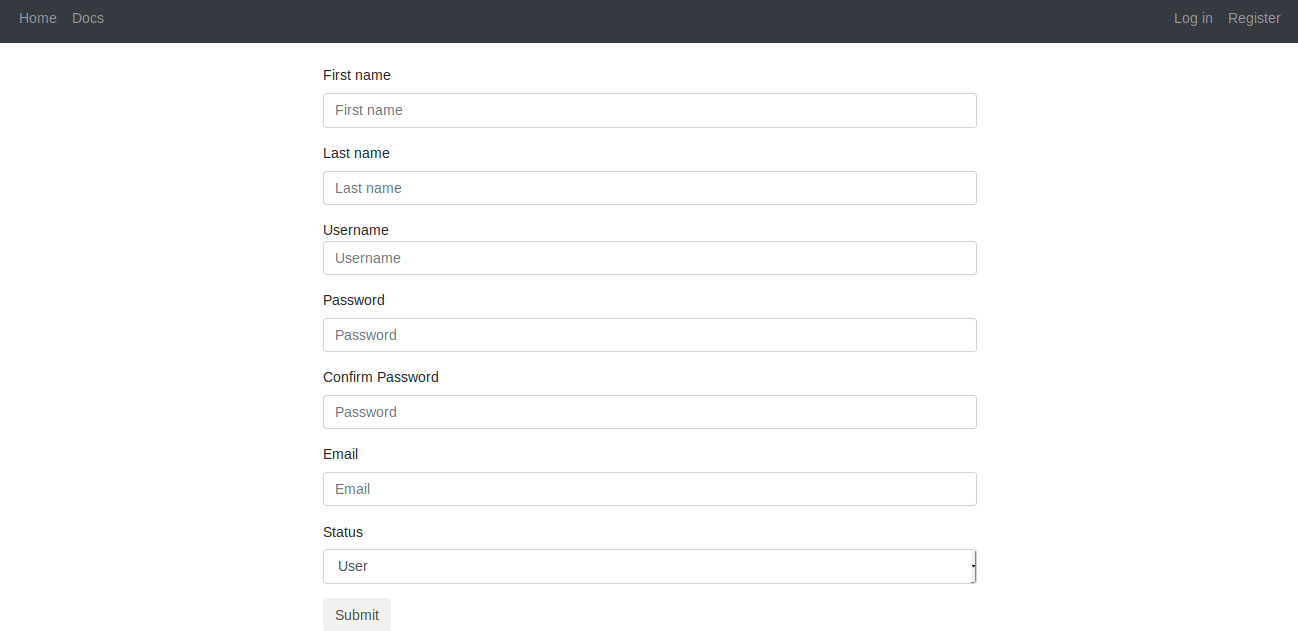
  
*De samenhang van de modules*

Het programma is opgedeeld in modules die ieders een ‘logisch’ geheel vormen van samenhorende functionaliteit.

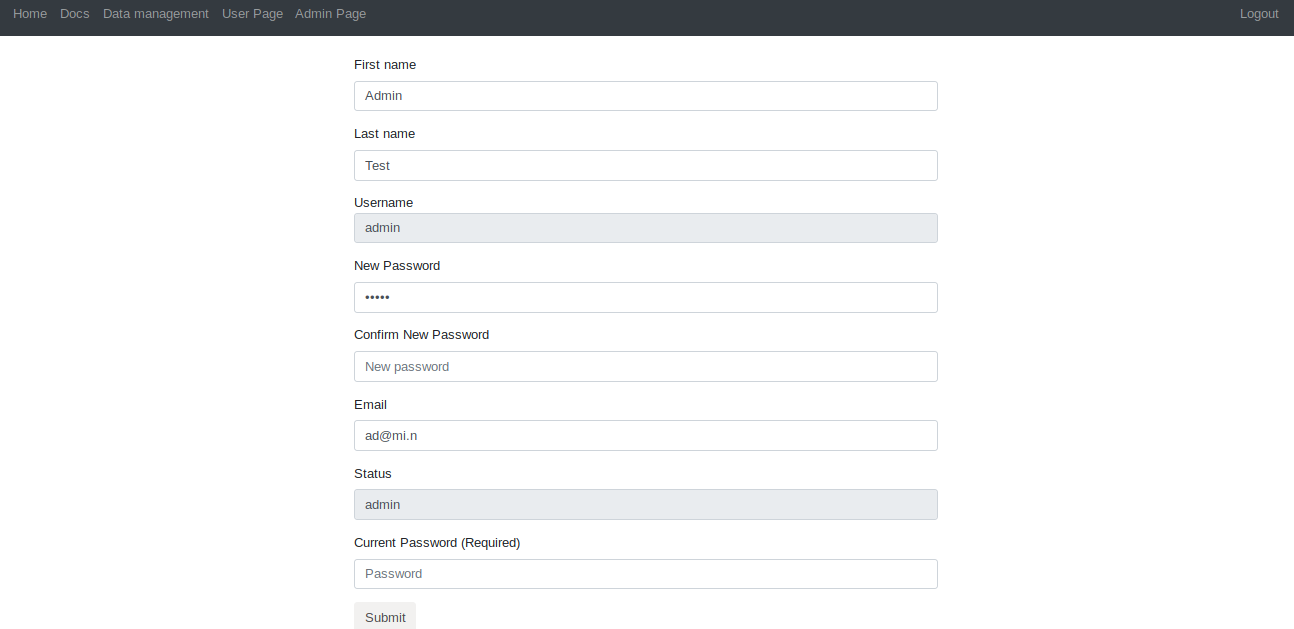
## 3. Manual

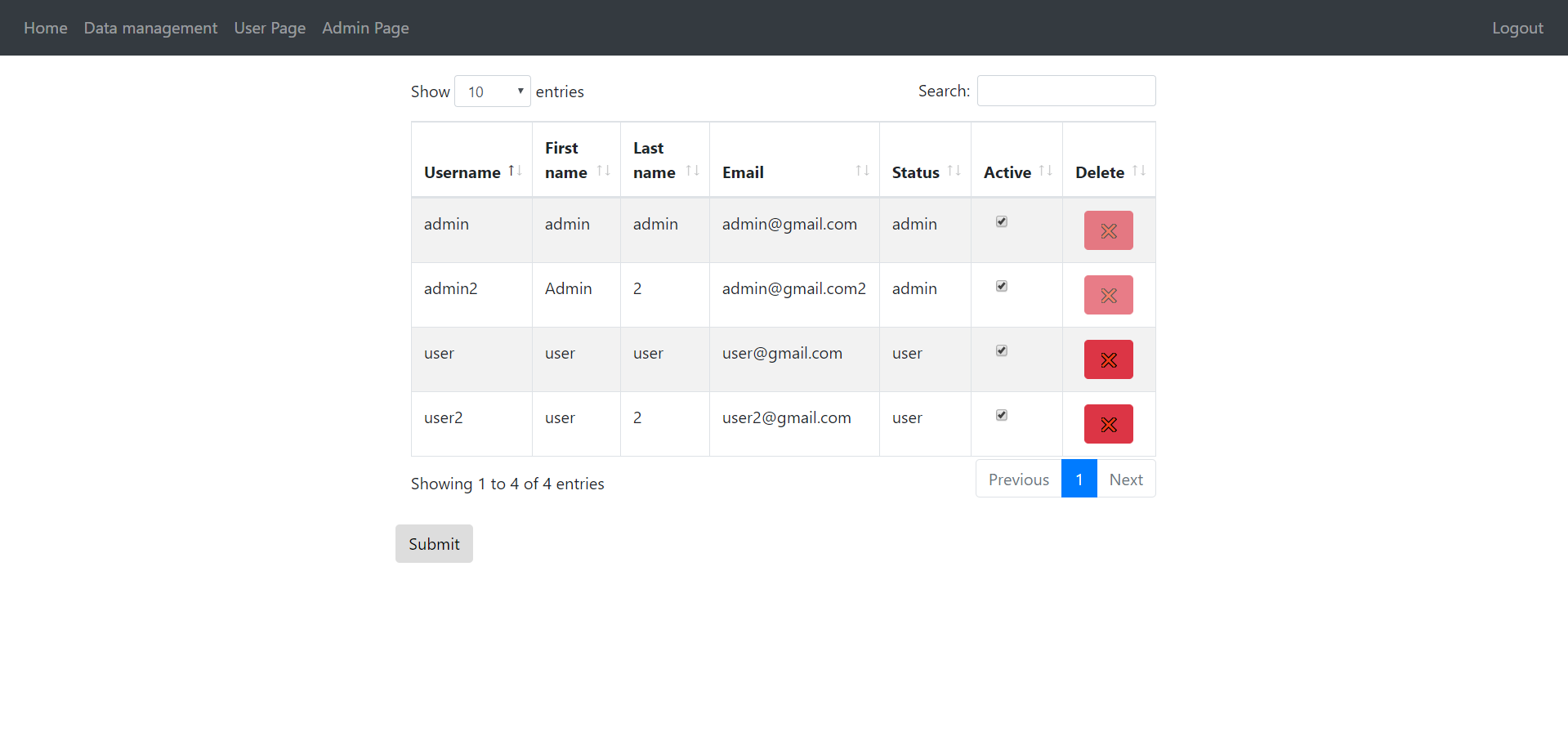
**

Op deze pagina kunnen gebruikers zich aanmelden. Indien het account gedeactiveerd is, of de logingegevens fout zijn, wordt de gebruiker op de hoogte gesteld.   
Er is ook een doorverwijzing naar de pagina waar nieuwe gebruikers zich kunnen registreren.

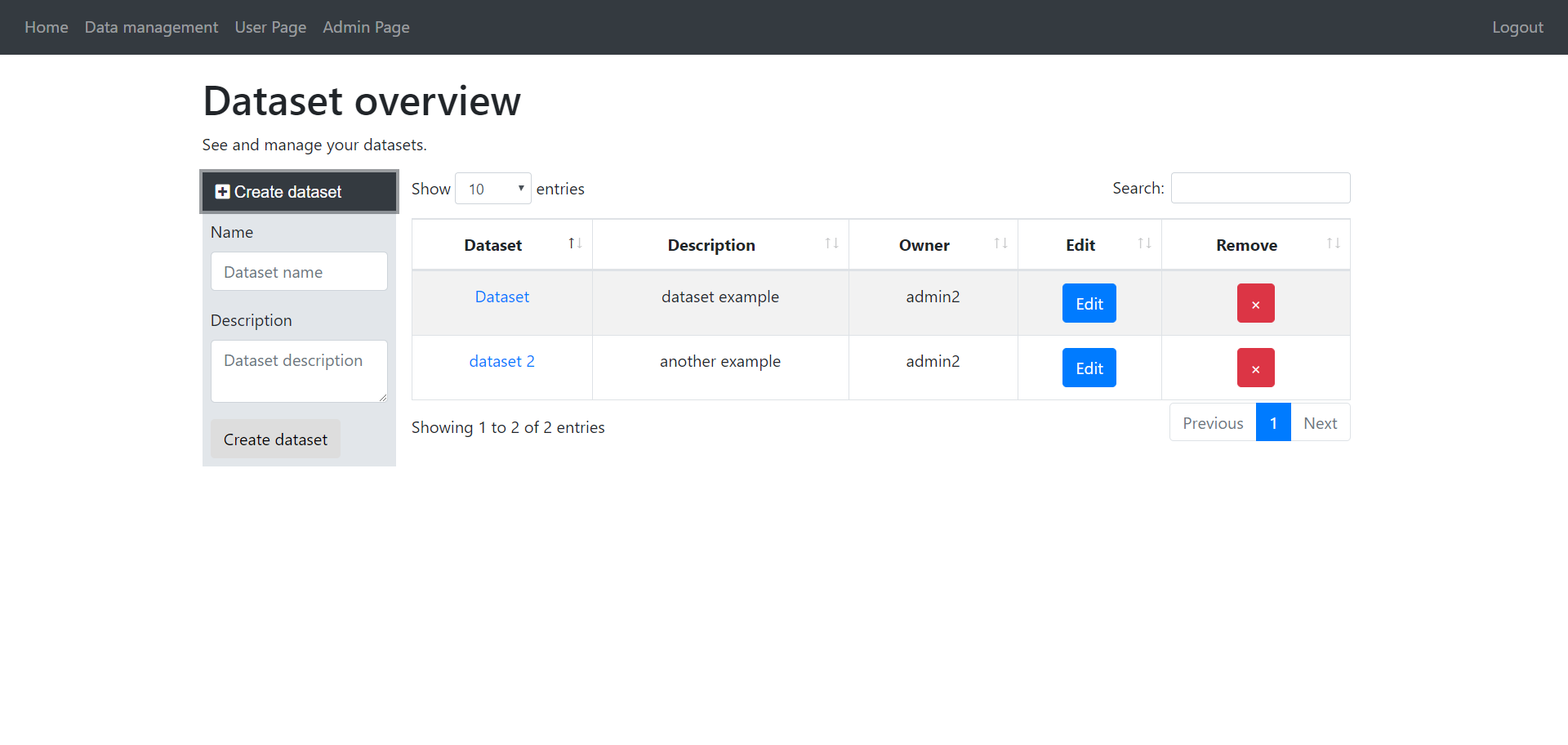


Op deze pagina kunnen gebruikers een nieuw account op de service aanmaken. Momenteel is het bij registratie mogelijk om de status van het account (admin of user) te kiezen.

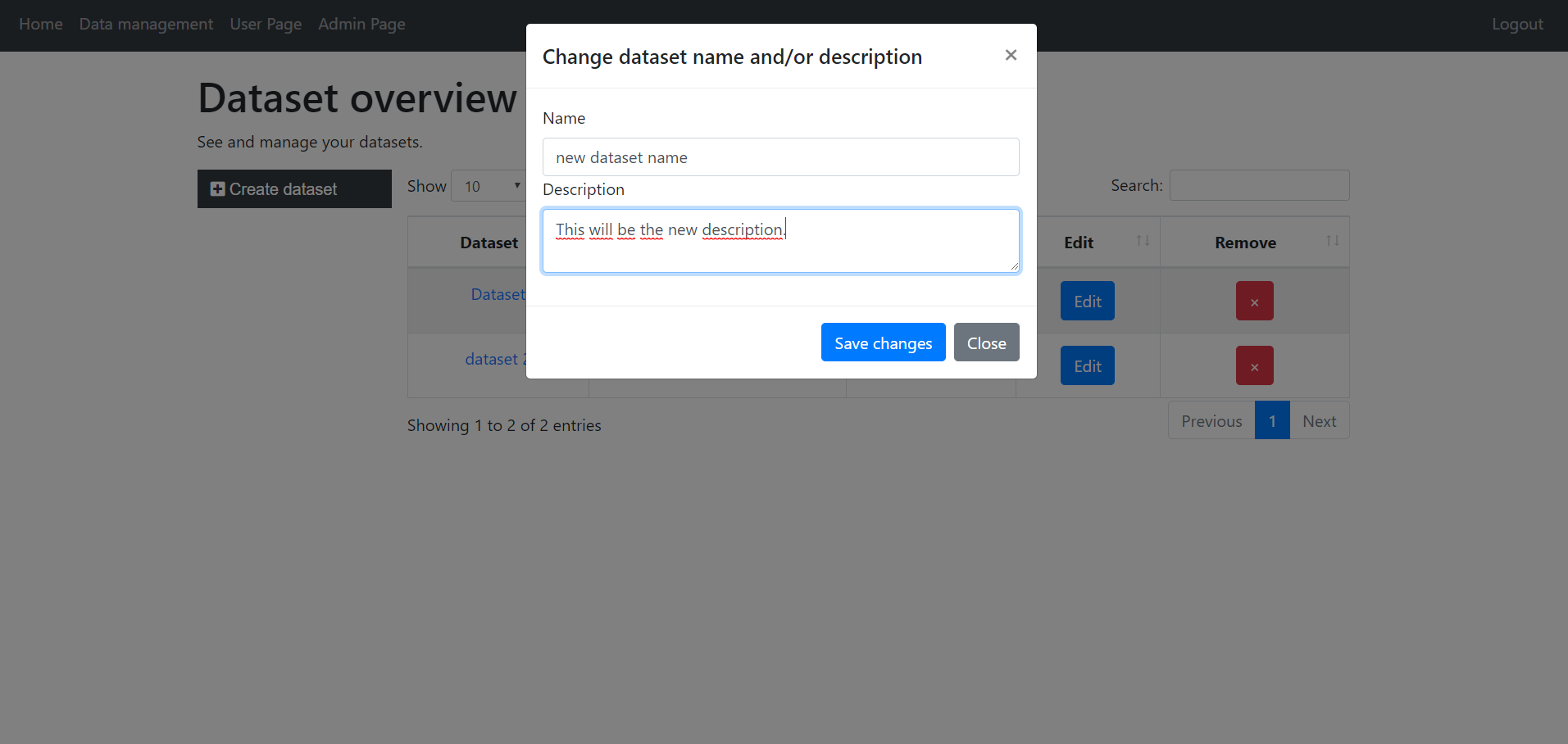
Hier kunnen gebruikers hun gegevens aanpassen.



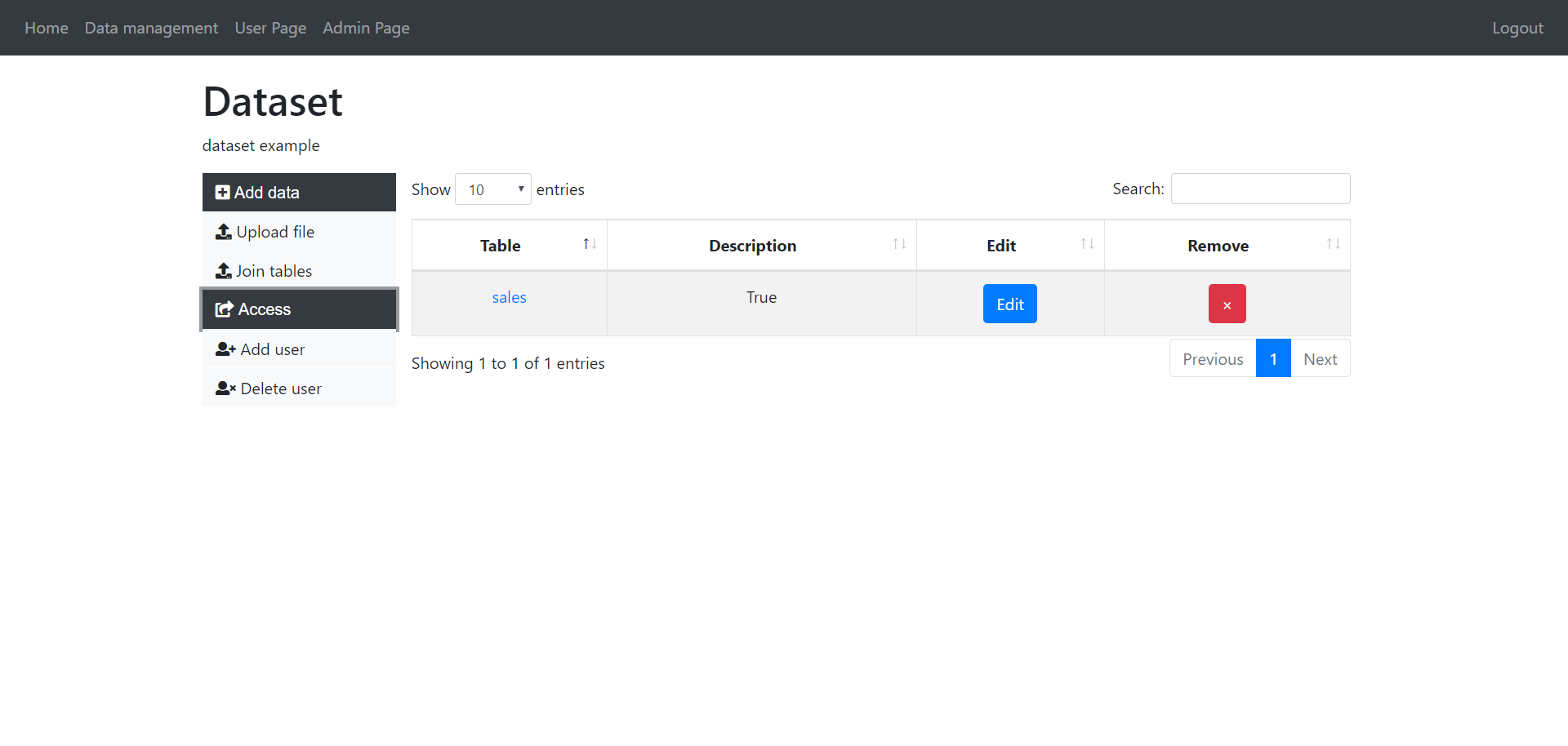
Hier kunnen admin gebruikers de active state van alle users aanpassen of deze users verwijderen.



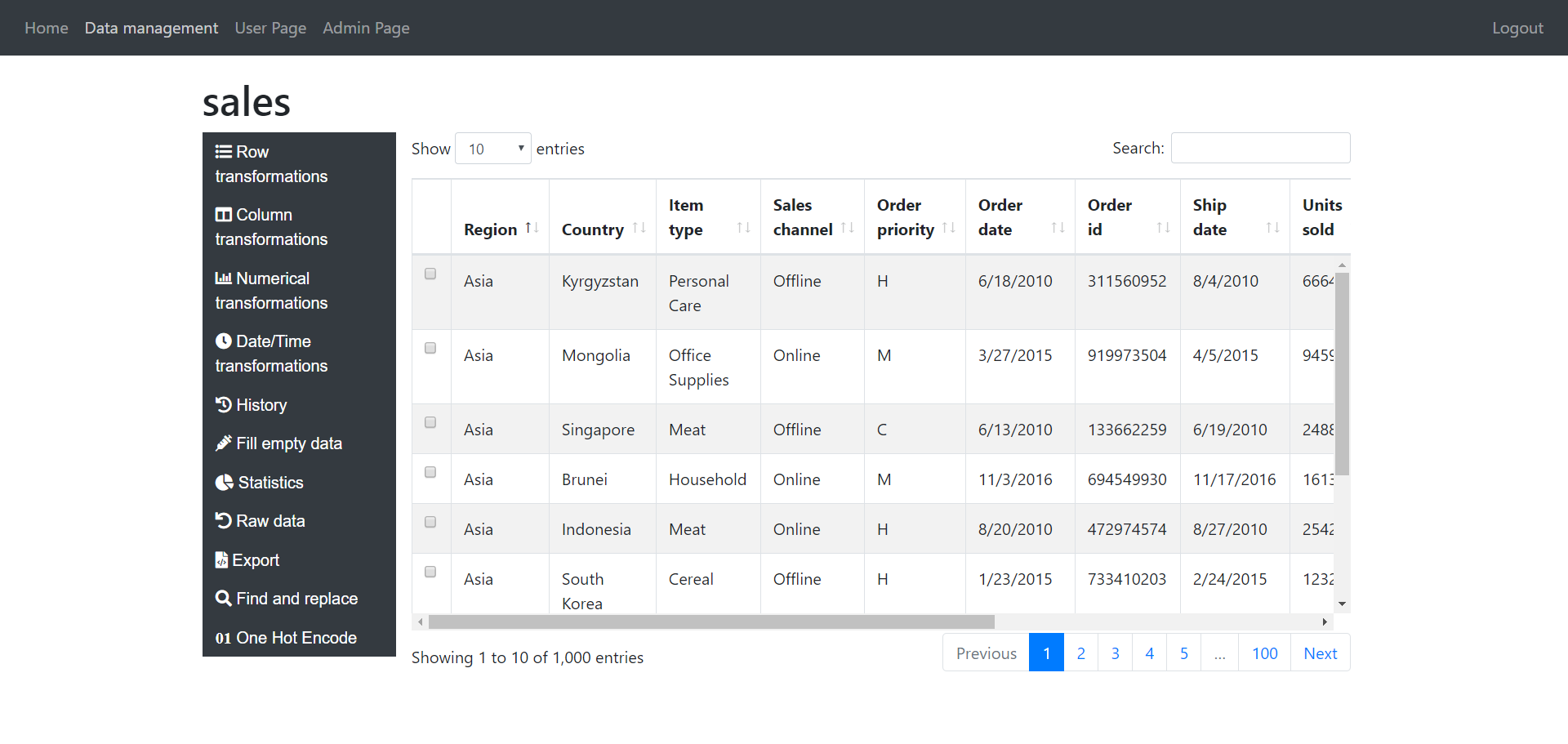
Hier kunnen gebruikers datasets bekijken/aanpassen/verwijderen/toevoegen.



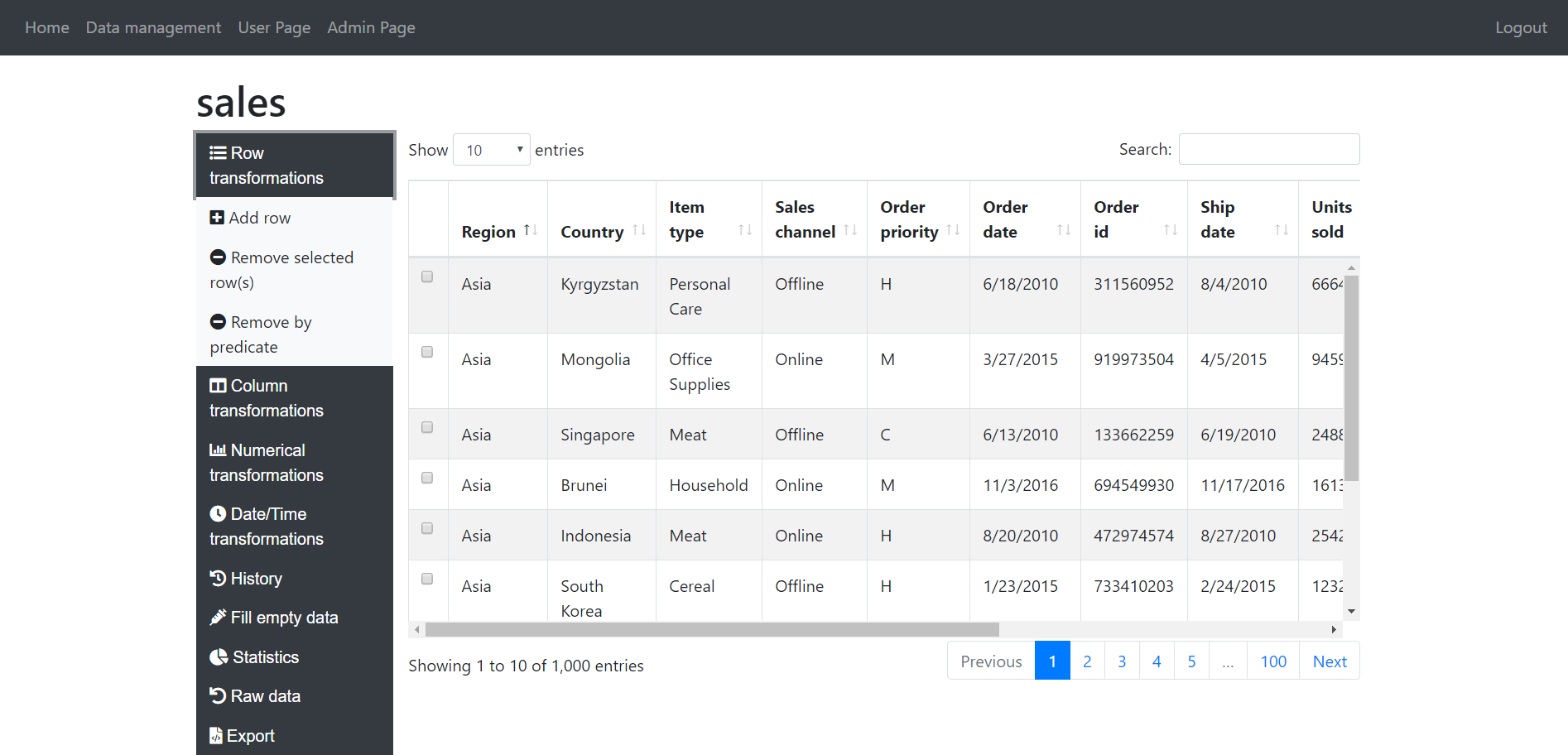
Gebruikers kunnen de metadata van een dataset aanpassen.



Hier zien gebruikers de tabellen die behoren tot een gegeven dataset. Gebruikers kunnen ook nieuwe files uploaden, tabellen joinen of de toegang van andere gebruikers wijzigen.



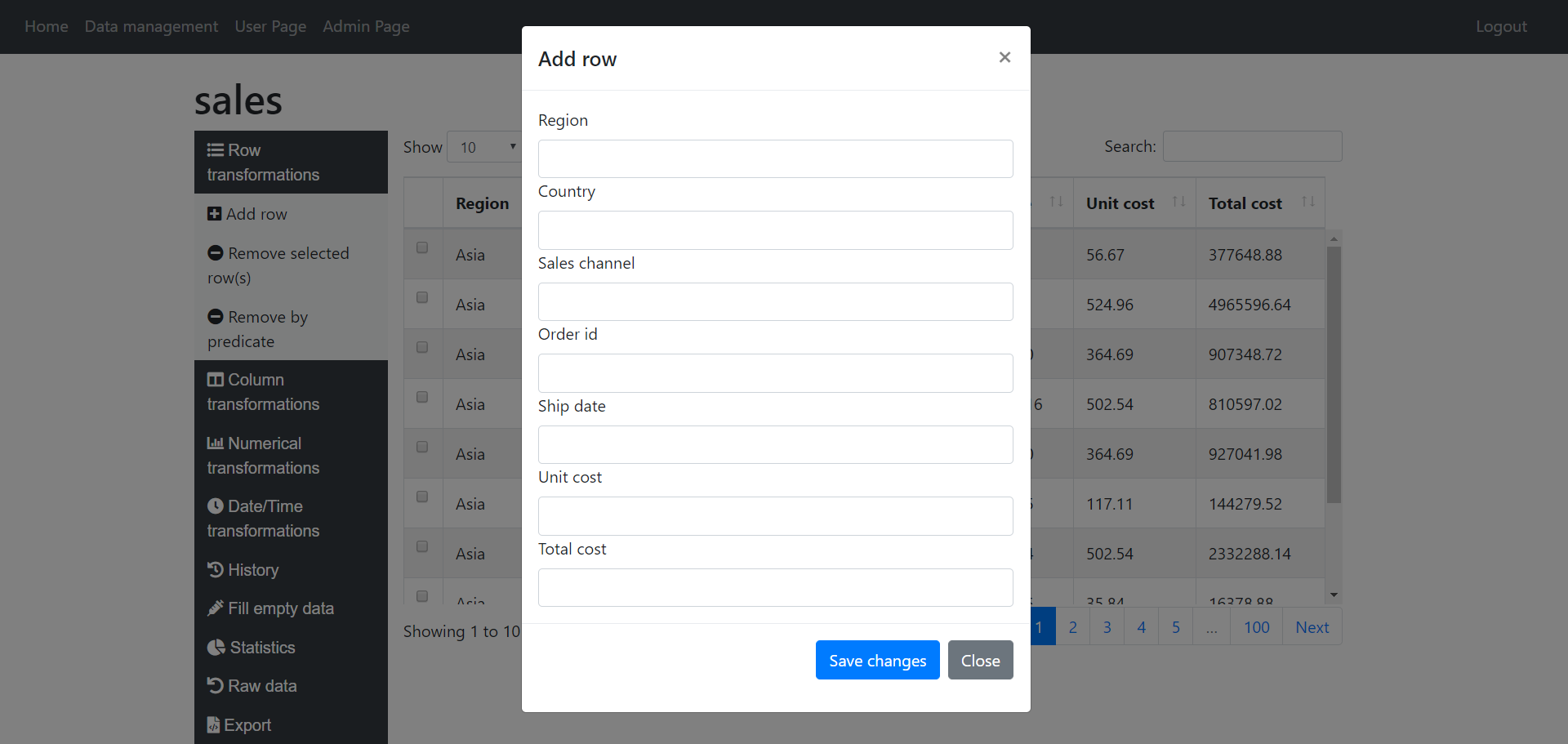
Hier zien gebruikers de werkelijke data. Links kan men ook schematisch verschillende aangeboden diensten zien. Bij het aanklikken van een soort dienst kan men meerdere opties verkrijgen.



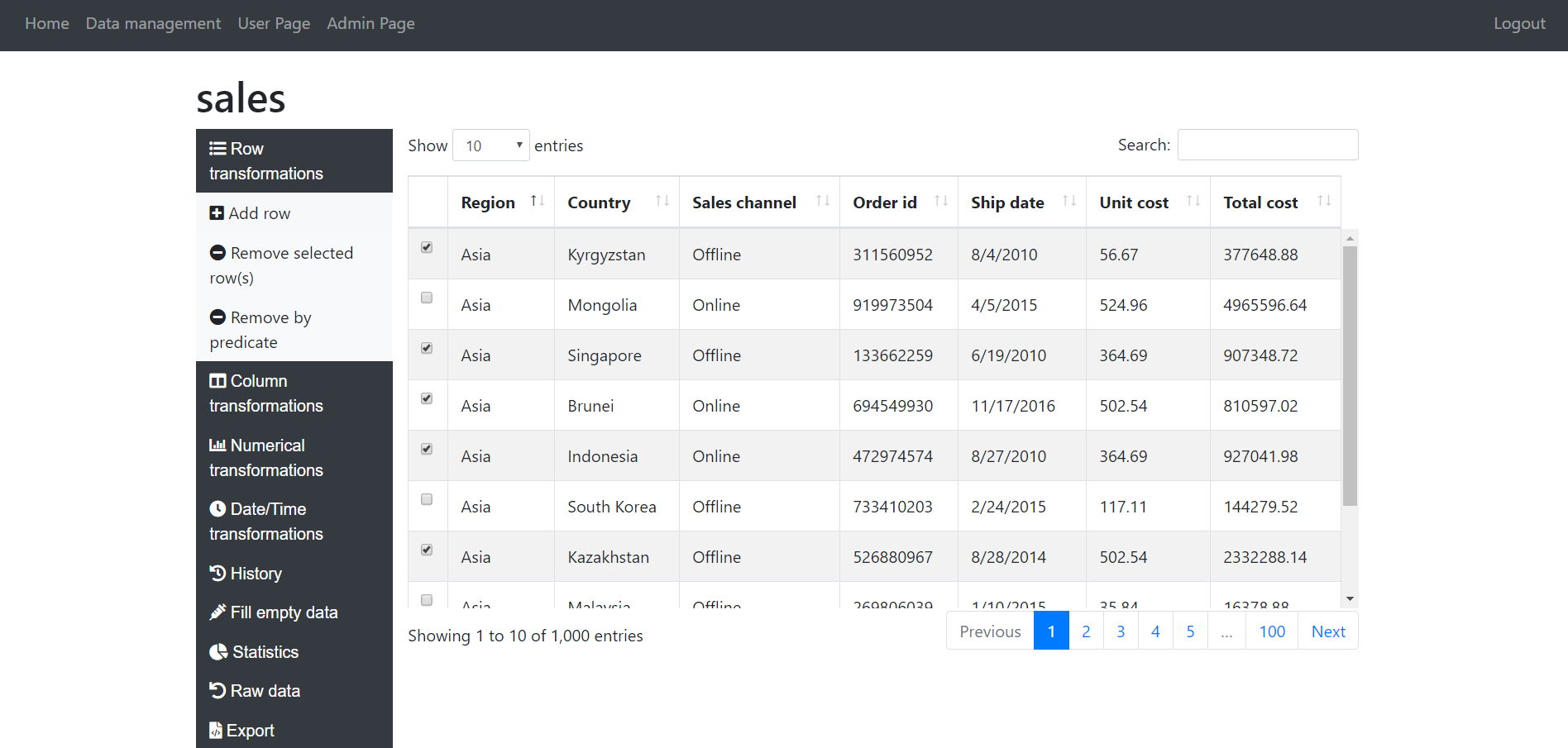
## 3. Data operaties

3.1 Rij transformaties

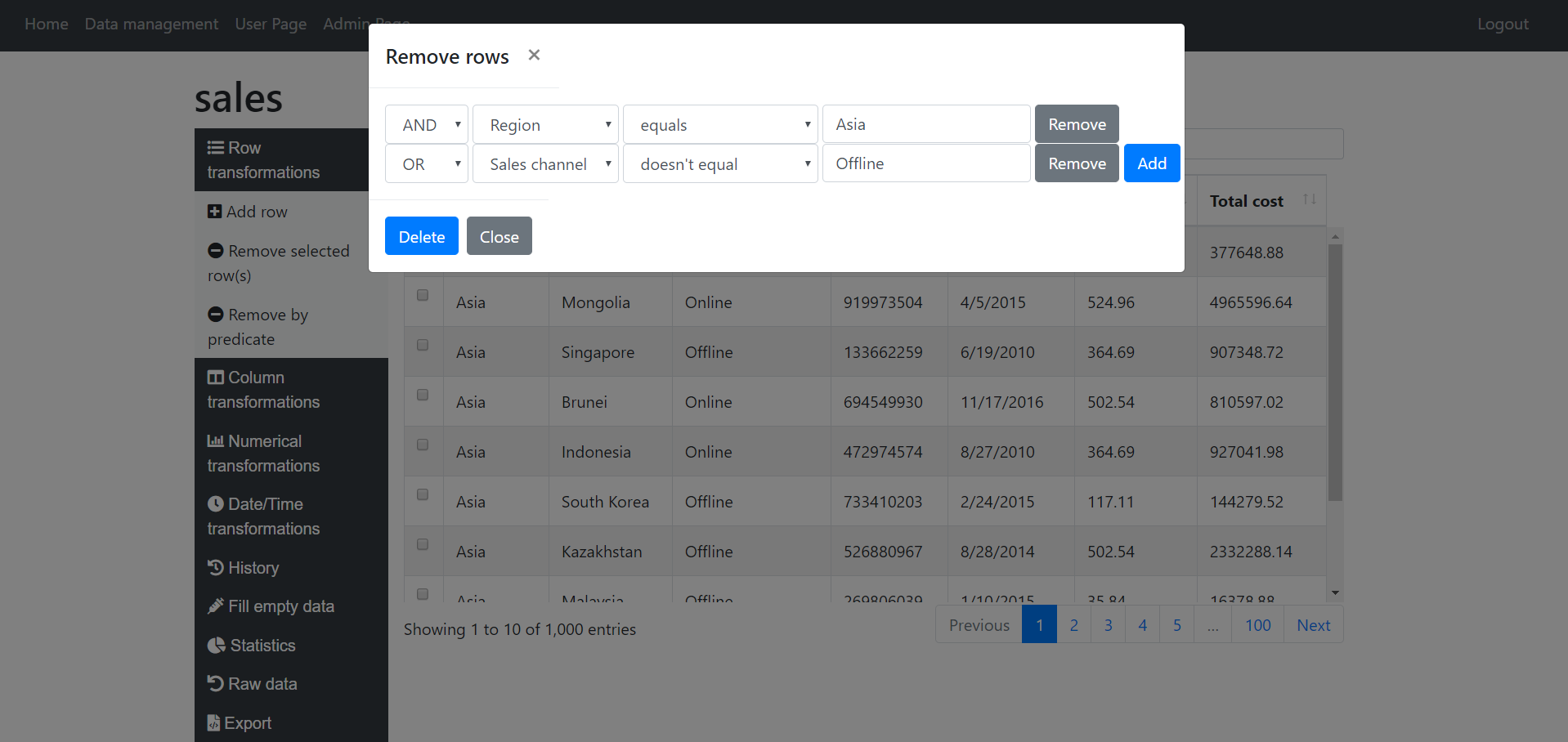
1. Een rij toevoegen.



1. Geselecteerde rijen verwijderen

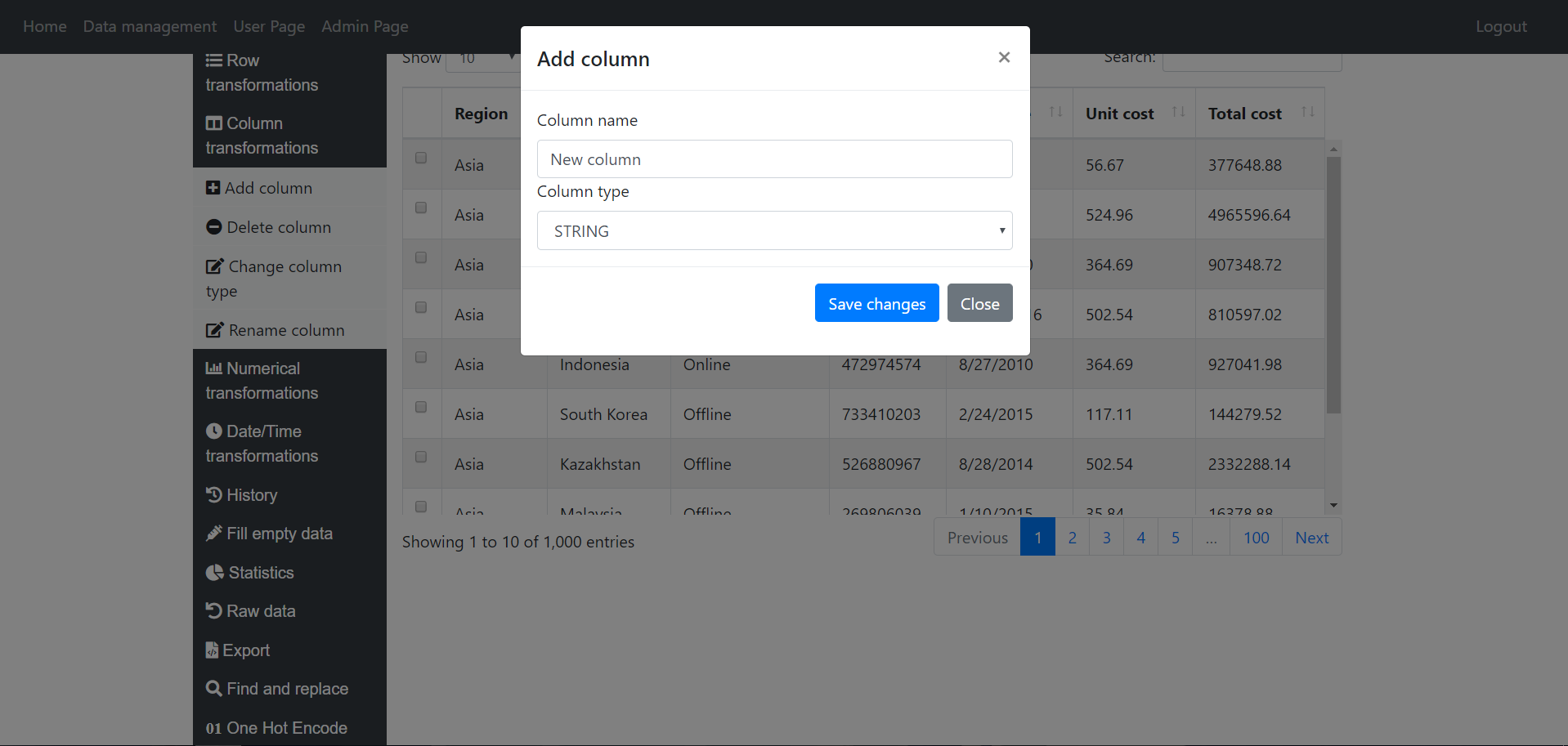


1. Rijen aan de hand van predicaten verwijderen

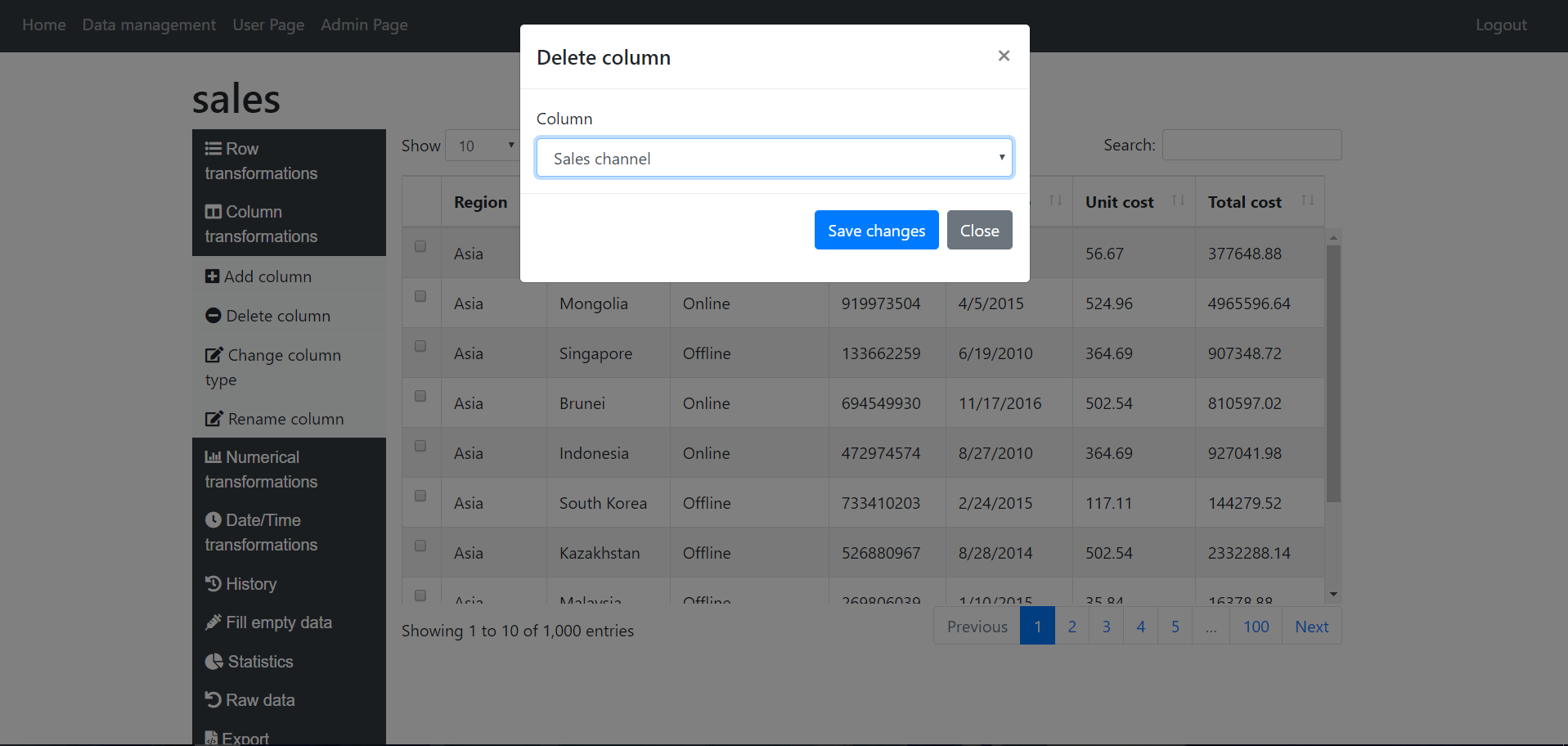


3.1 Kolom transformaties

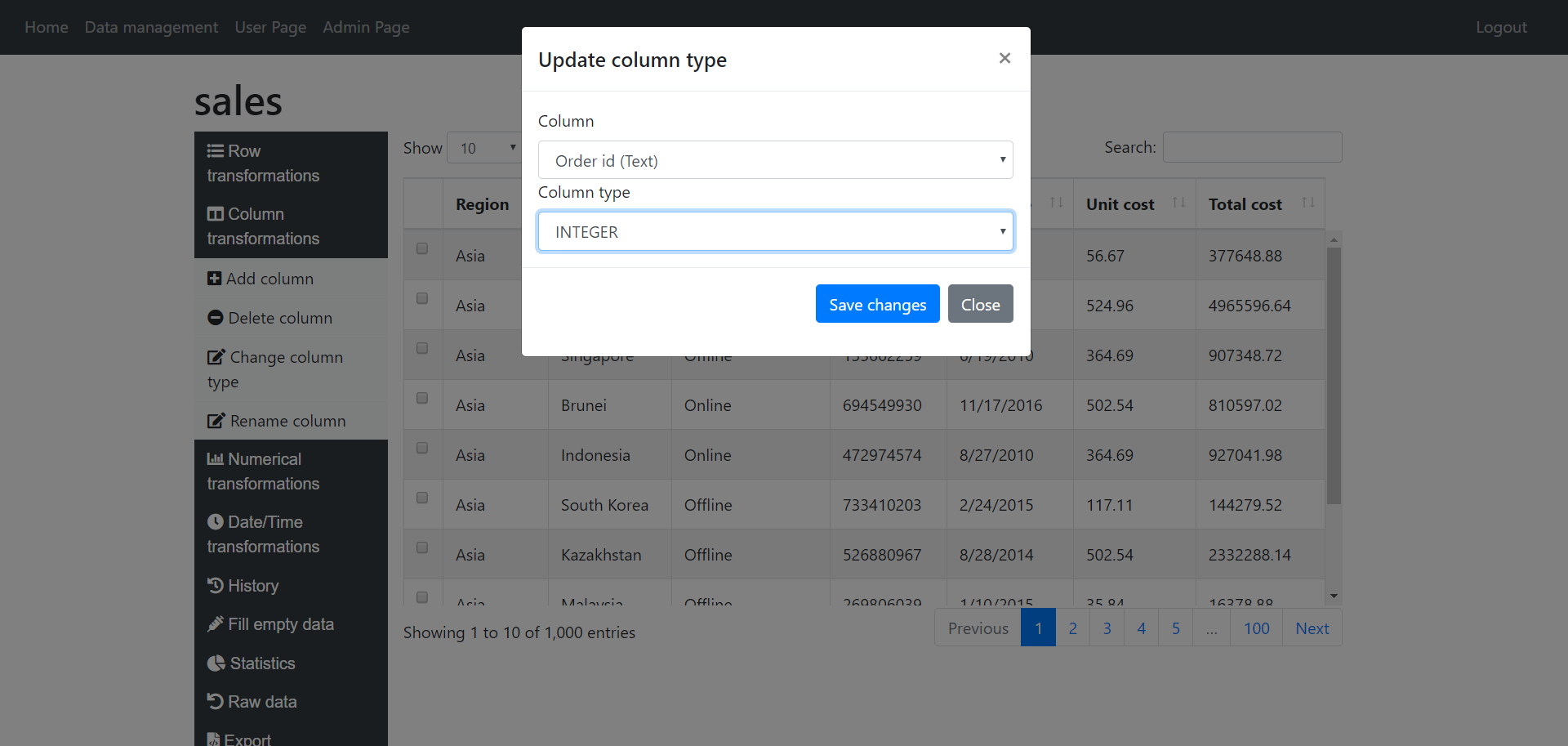
1. Kolom toevoegen



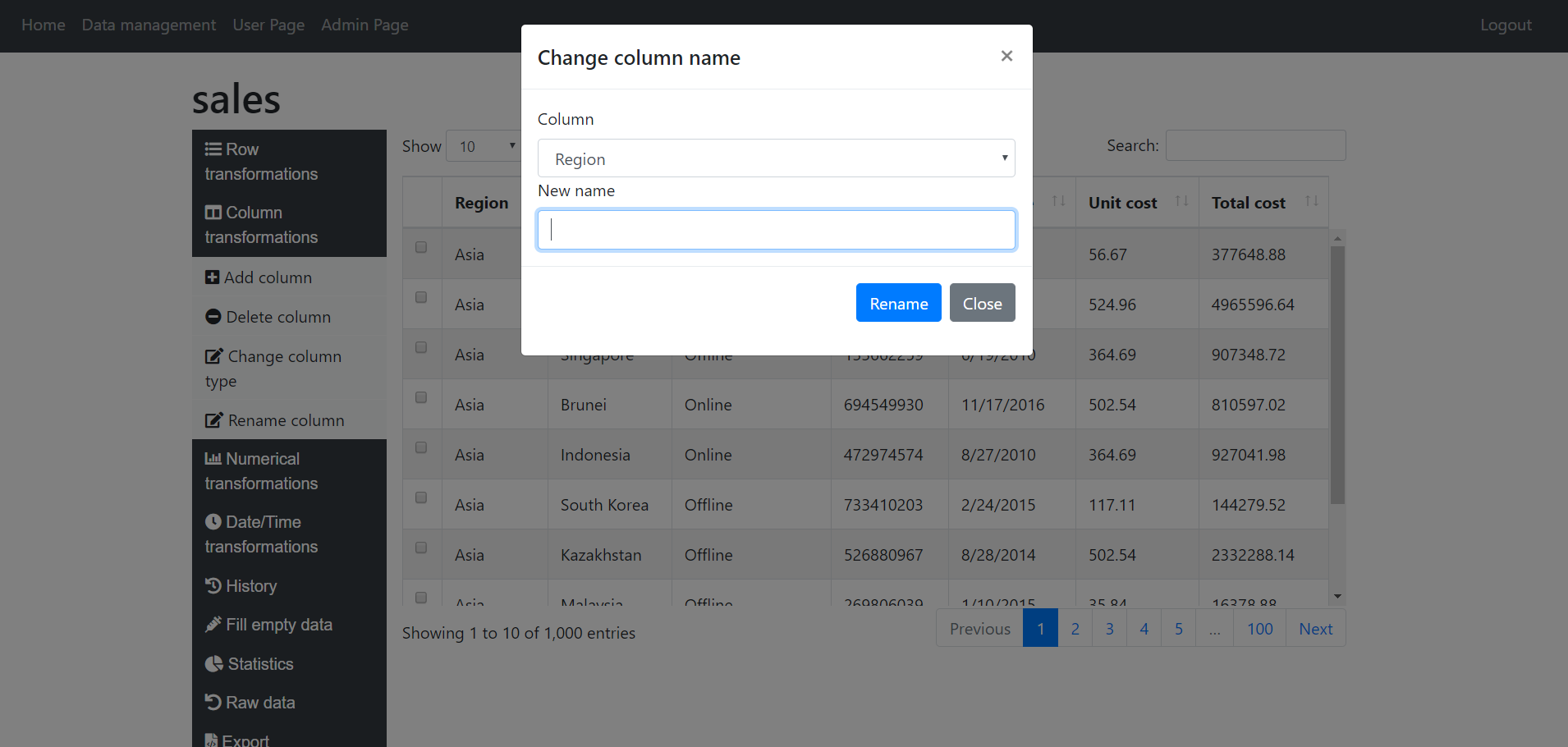
1. Kolom verwijderen



1. Kolom type veranderen

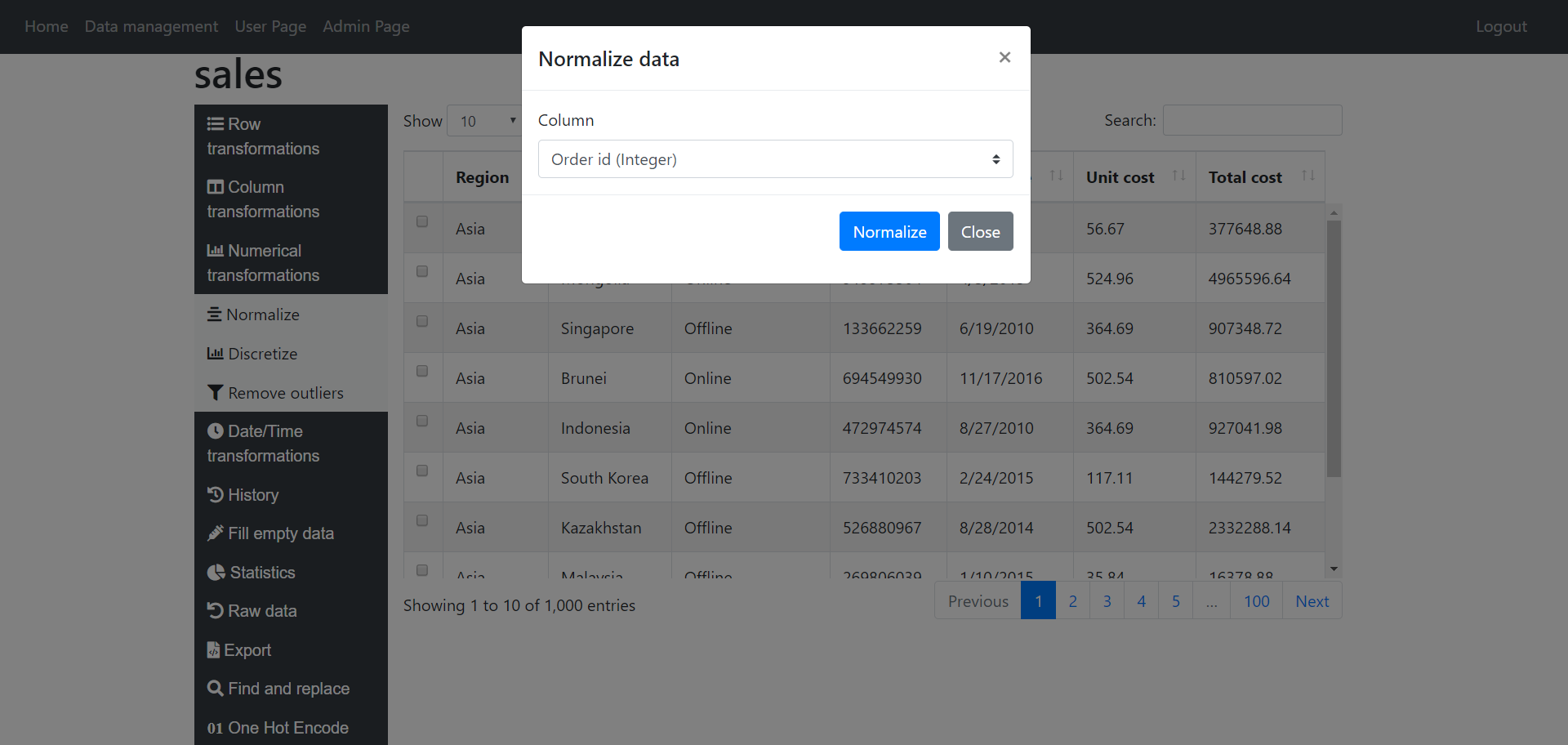


1. Kolom hernoemen

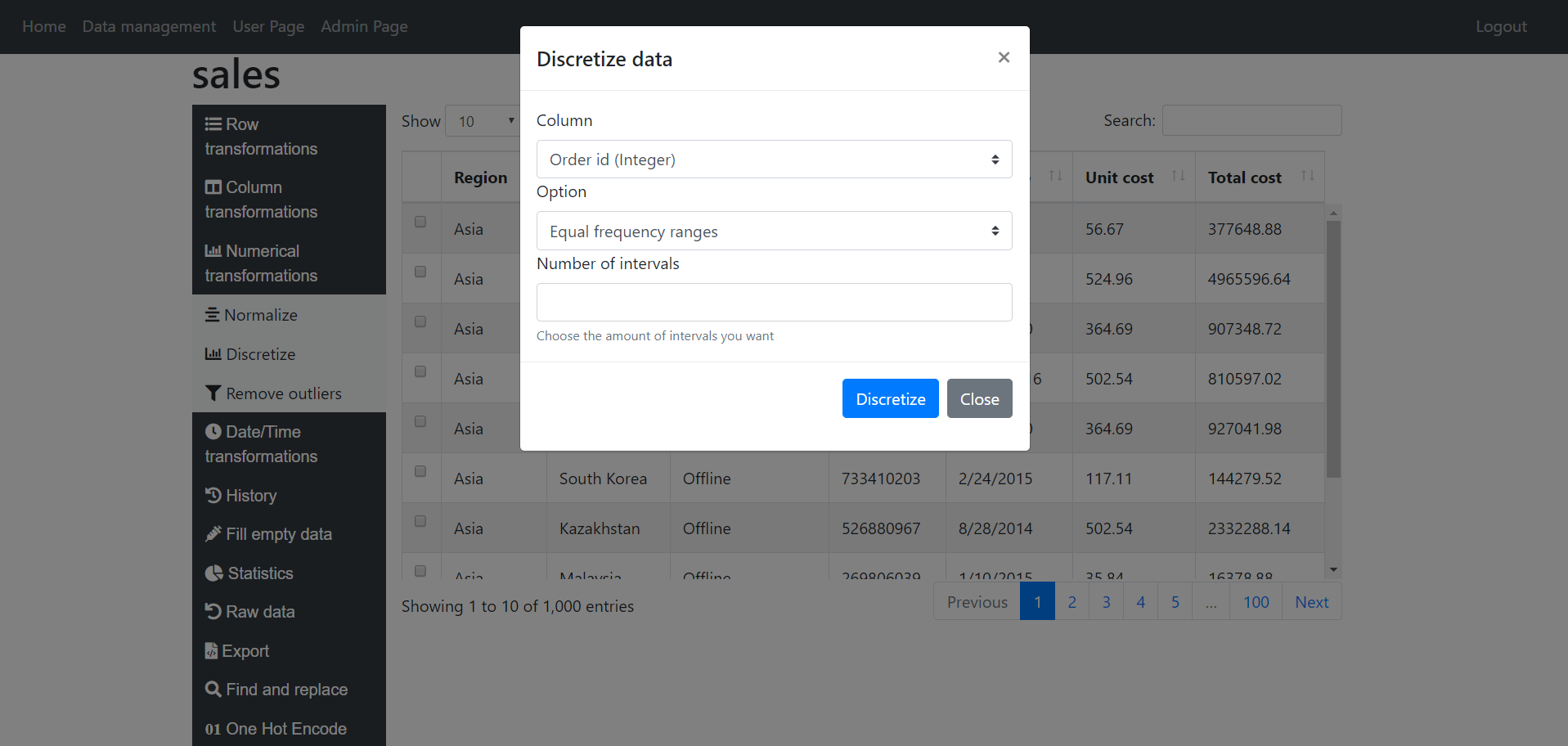


3.2 Rij transformaties

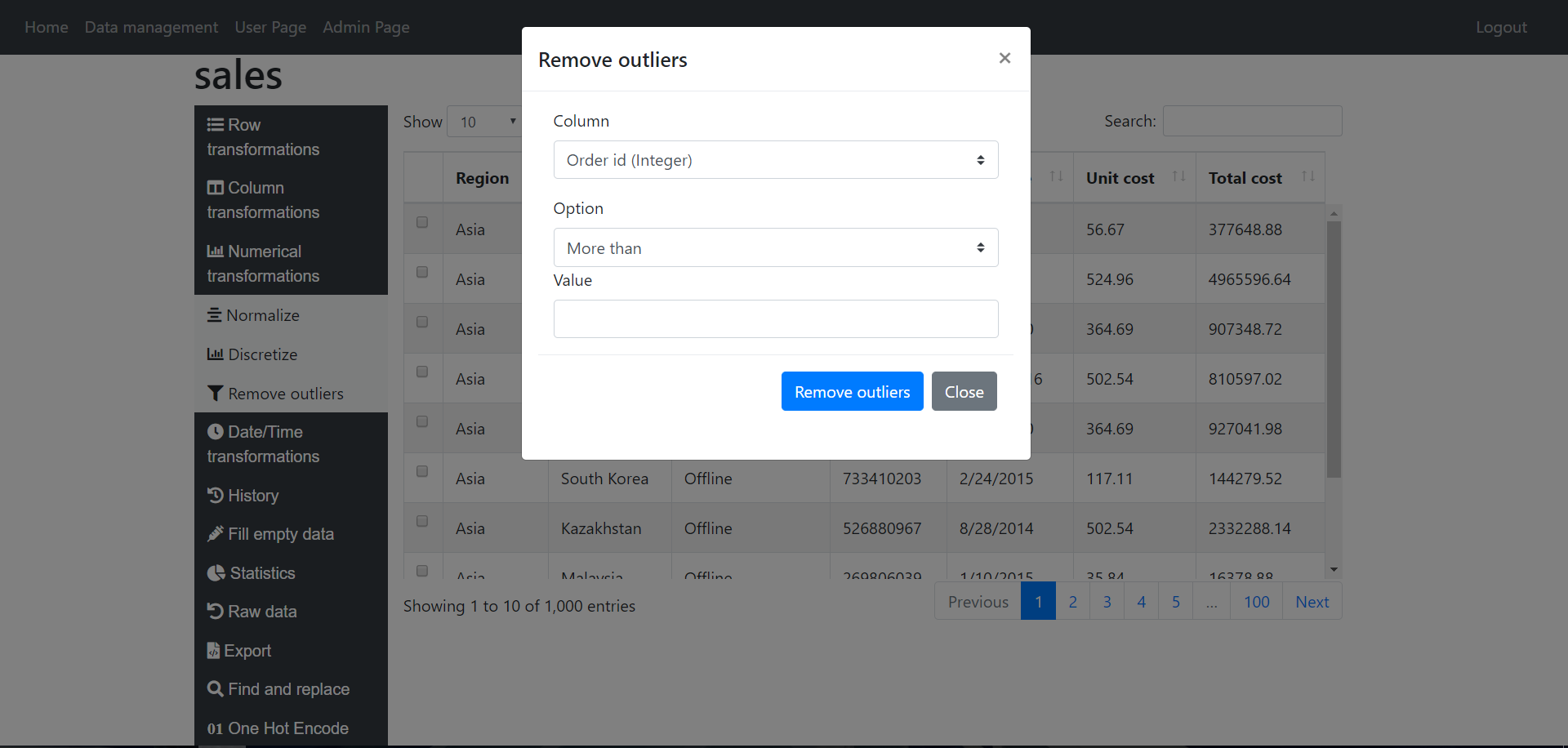
1. Normalizeren



1. Discretizeren



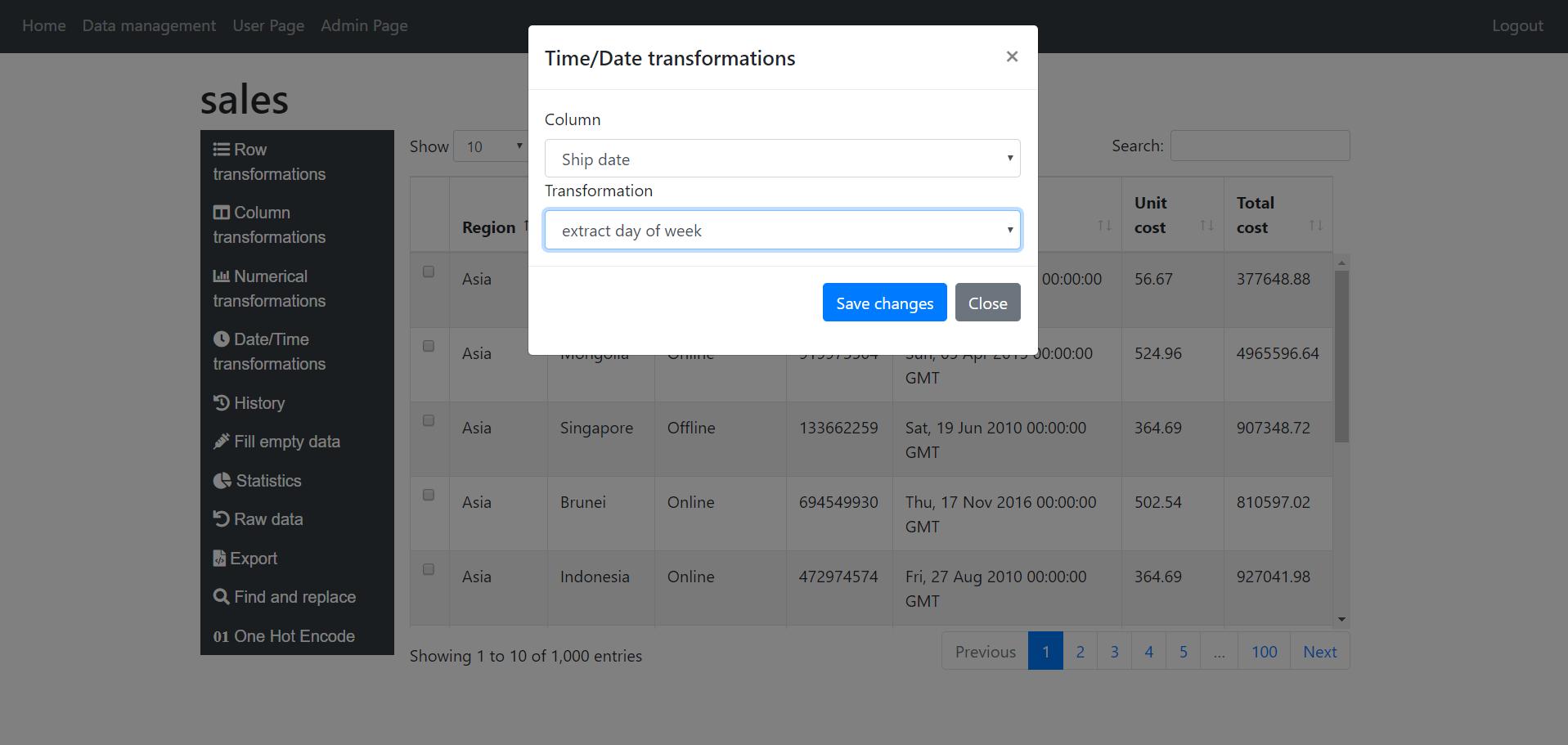
1. Outliers verwijderen

**

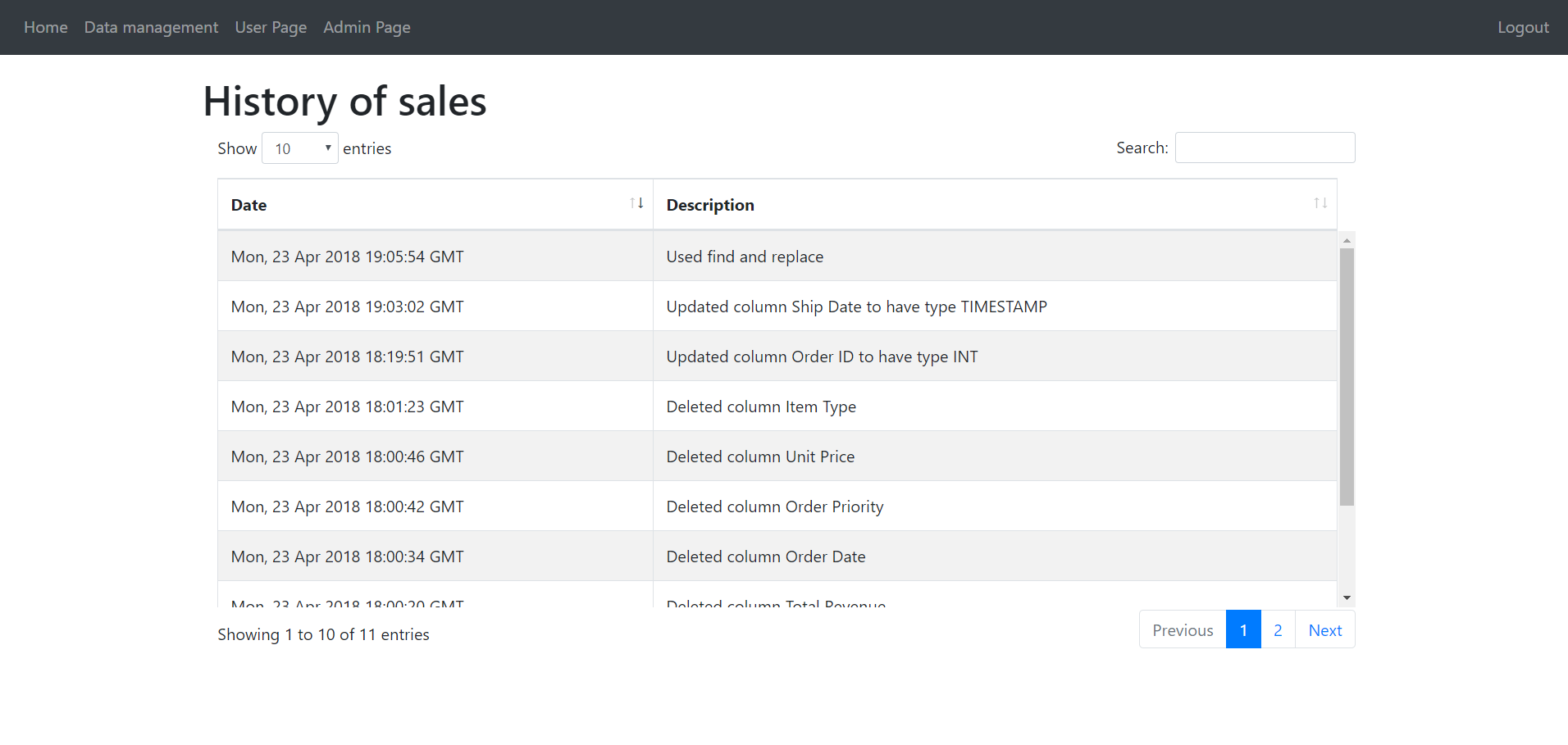
3.3 Datum/Tijd transformaties

Mogelijkheden:

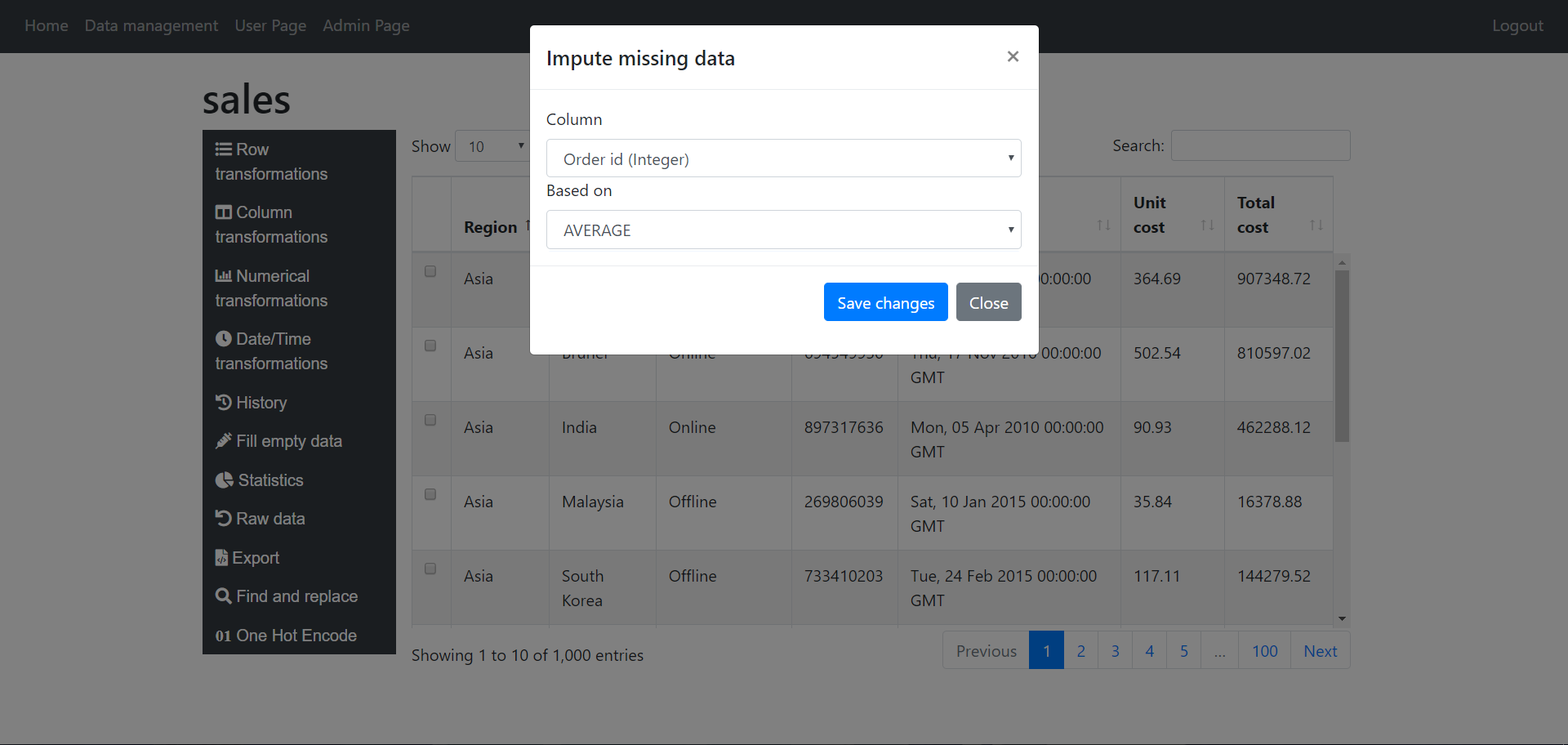
* Dag van de week extraheren
* Maand extraheren
* Jaar extraheren
* Datum extraheren
* Tijd extraheren



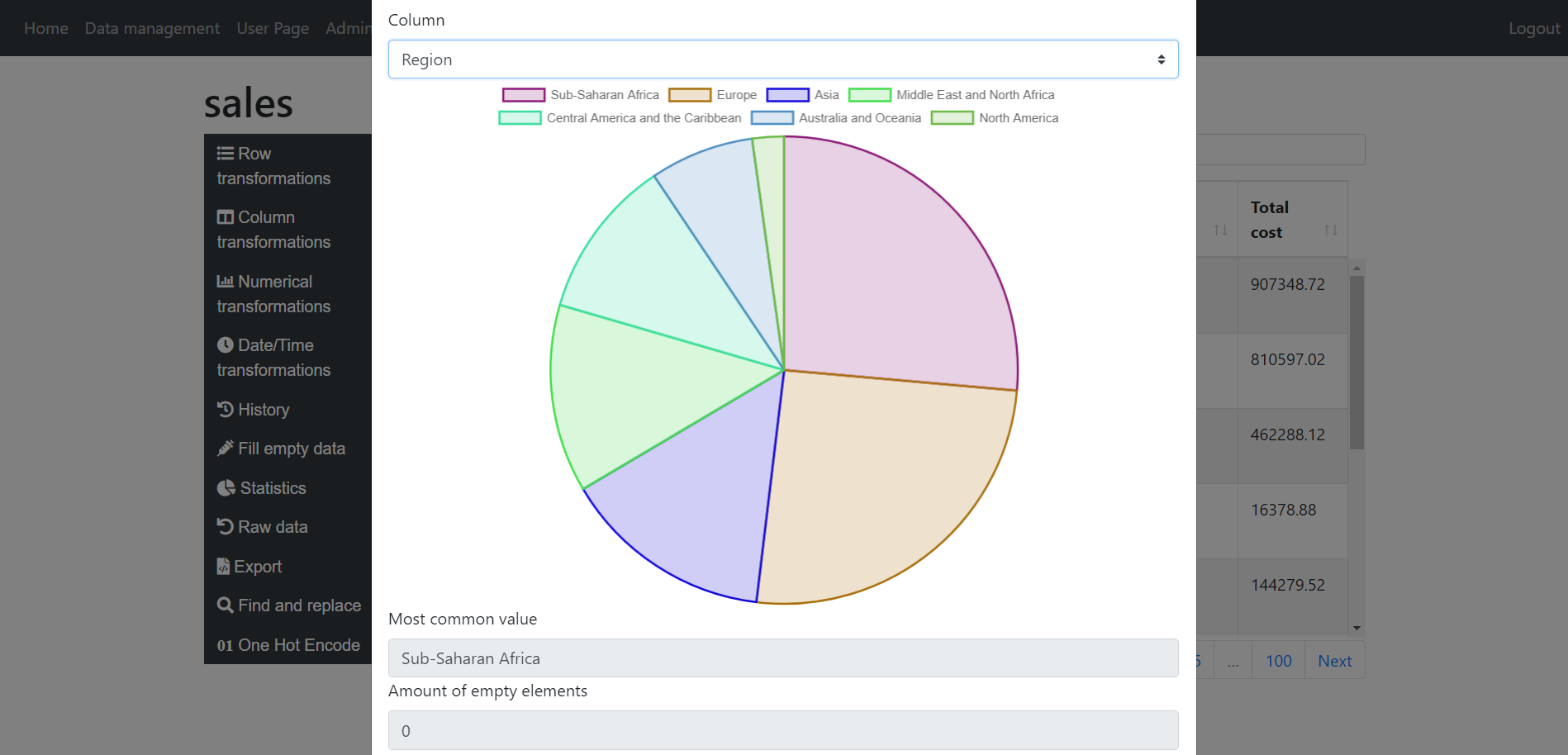
3.4 Actie geschiedenis



3.5 Imputeren van ontbrekende data

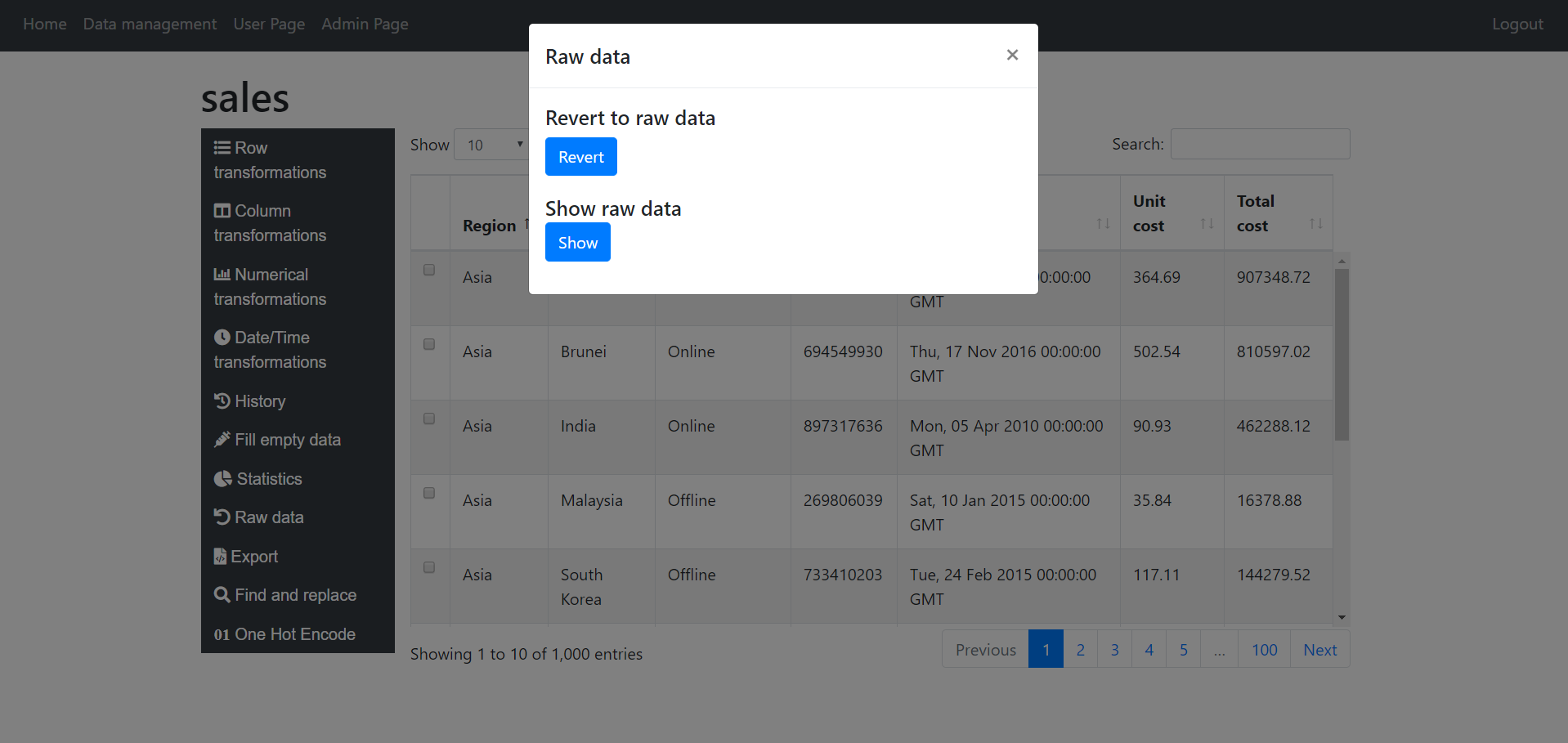


## 3.6 Statistieken en grafieken

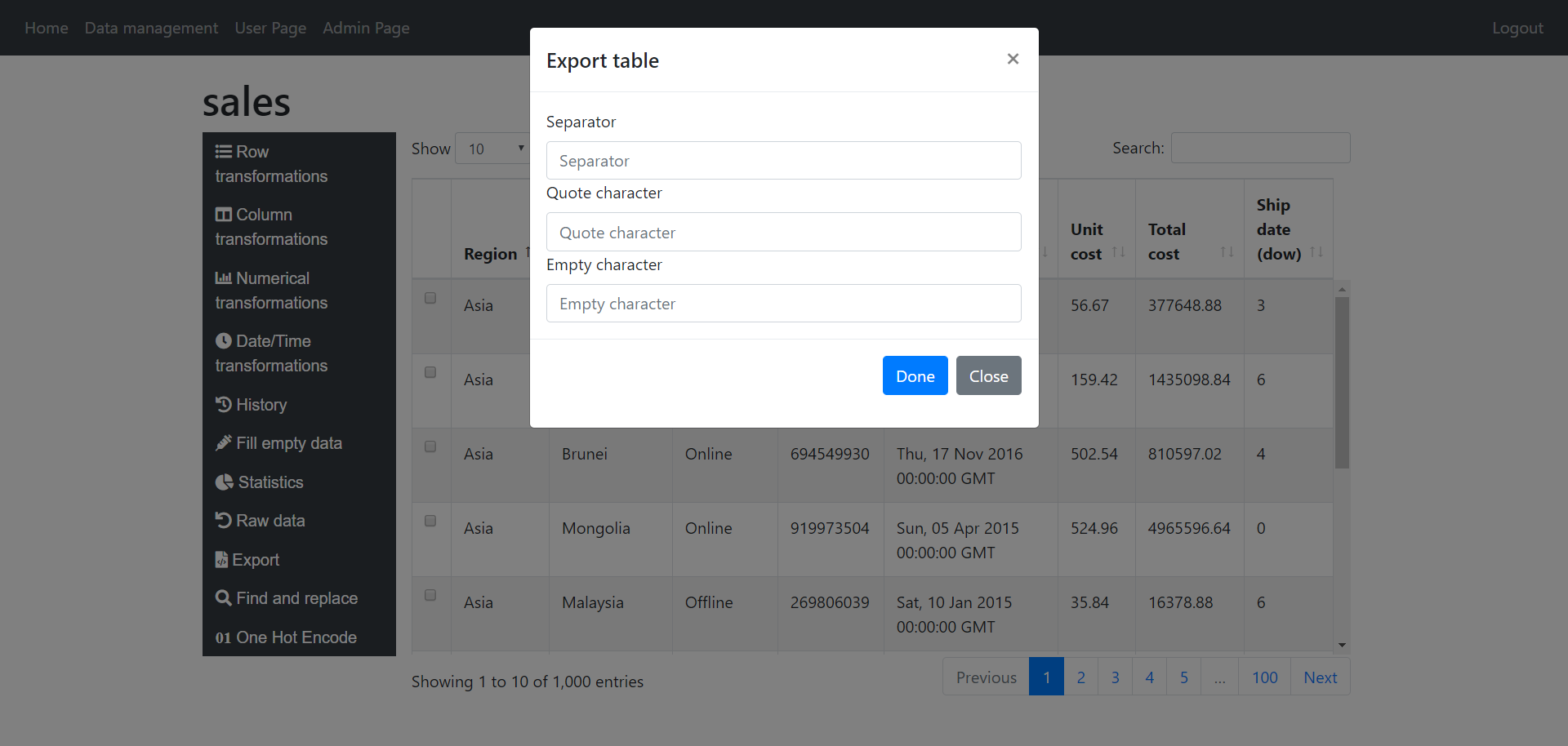


## 3.7 Ruwe data

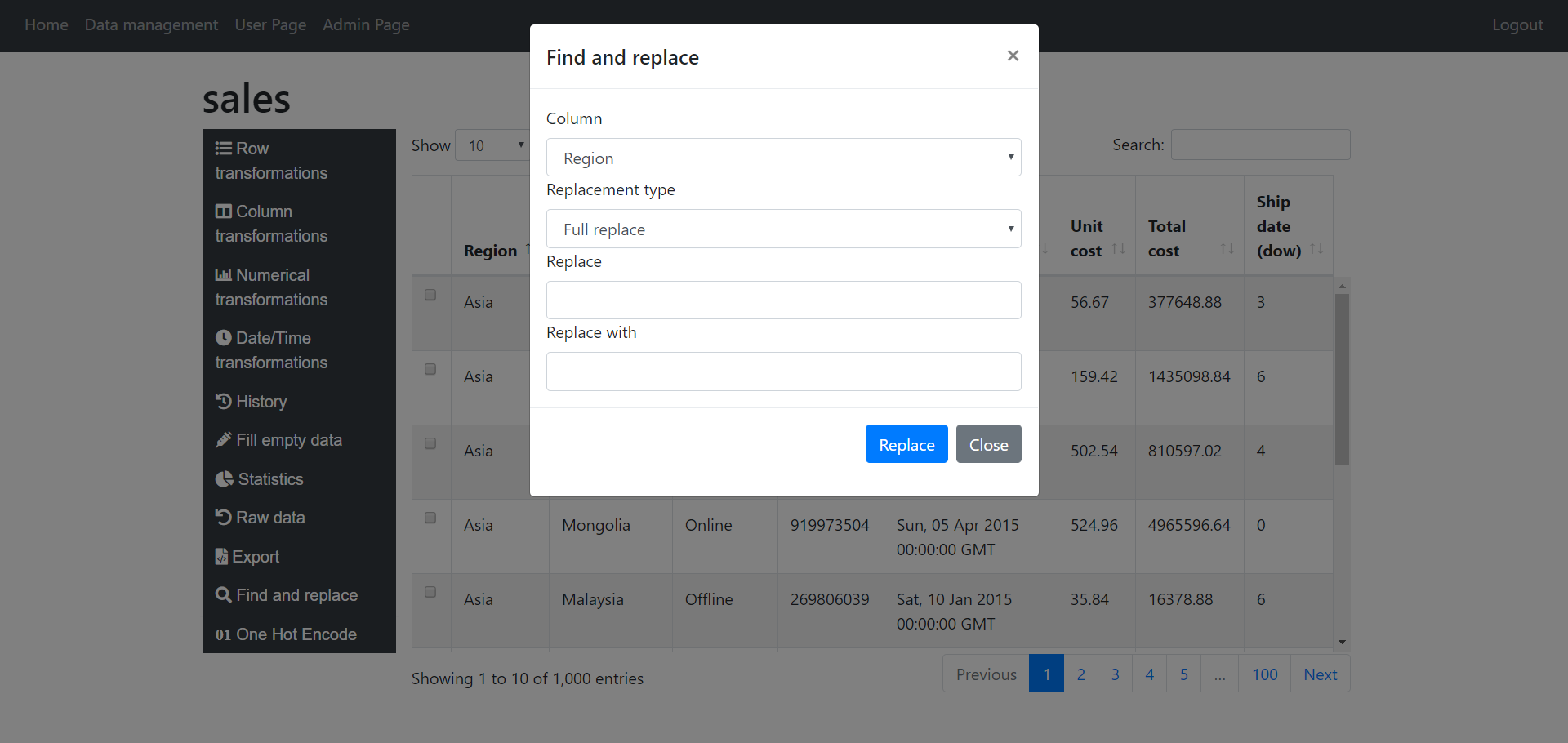
Er wordt ruwe data bijgehouden van tabellen die meegegeven zijn aan de hand van bestanden. Deze onbehandelde data kan de gebruiker nadien opvragen en terug resetten om opnieuw te beginnen.



## 3.8 Exporteren van data



## 3.9 Find and replace



## 3.10 One hot encoding

## 

## 3.10 Tabellen joinen

## 

4. Extra functionaliteit

Afgewerkt:

* Zorg ervoor dat de weergave van grote datasets gepagineerd gebeurt, dus laadt enkel data die effectief weergegeven wordt.
* Simpele zoekfunctie voor tabellen

Gepland:

* Data deduplicatie
* Ontbrekende waardes voorspellen
* History undo
* Voor tabellen met coördinaten (longitude, latitude) de mogelijkheid geven om een heatmap weer te geven

Optioneel:

* SQL-console voor gevorderde gebruikers
* Bag-of-words implementatie