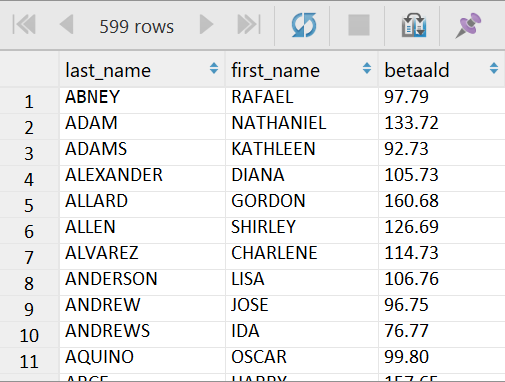
Databanken – SQL – oefeningen 2

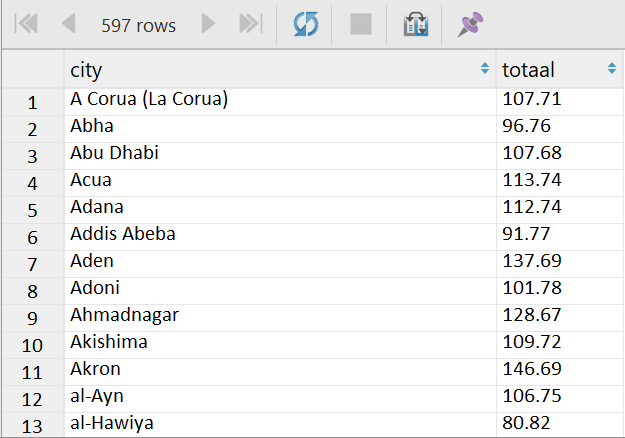
Maak gebruik van de **sakila** database om volgende oefeningen op te lossen.

1. Geef een alfabetisch overzicht van klanten (customer) en het totaal van hun betalingen (payment):



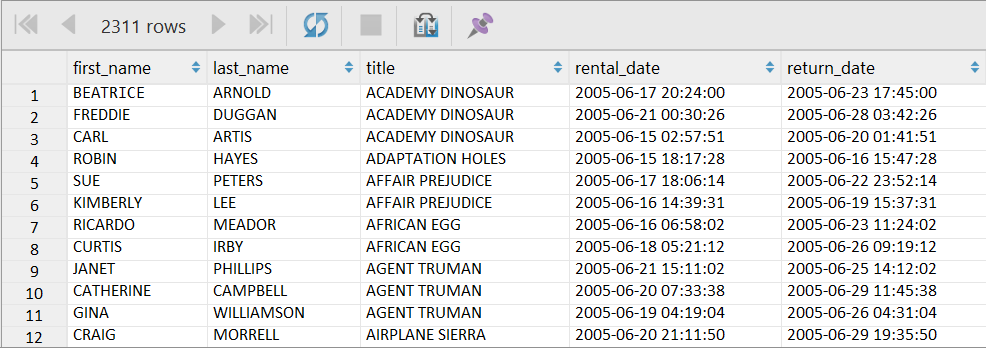
select last\_name, first\_name, sum(amount) as betaald  
from payment  
join customer c on payment.customer\_id = c.customer\_id  
group by c.customer\_id  
order by c.last\_name

1. Idem, maar dan per stad:



select city, sum(amount) as betaald  
from payment  
inner join customer c on payment.customer\_id = c.customer\_id  
inner join address a on c.address\_id = a.address\_id  
inner join city on a.city\_id = city.city\_id  
group by city.city\_id  
order by city

1. Geef een lijst van klanten met de films die ze gehuurd hebben in de maand juni 2005 zoals hieronder weergegeven:



select first\_name, last\_name, title, rental\_date, return\_date  
from rental  
join customer on rental.customer\_id = customer.customer\_id  
join inventory on inventory.inventory\_id = rental.inventory\_id  
join film on film.film\_id = inventory.film\_id = film.film\_id  
where date(rental\_date) between '2005-06-01' and '2005-06-30'

1. Zet de datum-tijd laatste wijziging van de landen (country) op vandaag, 14u05

update country  
 set last\_update = now();

1. Haal in DataGrip de **DDL** van de tabel **actor** op (“Generate DDL to Clipbboard”), en plak die in Kladblok (op Windows) of TextEdit (op Mac).

Breng dan volgende wijzigingen aan in het DDL statement:

* Verander de naam van de tabel in **actor\_backup**
* Verwijder de index
* Verwijder het automatisch invullen van het veld **last\_update**

Voer het statement dan uit in de sakila database. Je maakt zo een lege kopie van de **actor** tabel. Controleer of de tabel **actor\_backup** wel bestaat in de databank. Bewaar het DDL statement.  
  
Kopieer dan de inhoud van de tabel **actor** naar de tabel **actor\_backup** met de MySQL variant van het SELECT INTO statement. Zoek op internet hoe die variant er uitziet in MySQL.

1. Voeg de acteur MATIAS SCHOENAERTS toe aan de tabel actor met een INSERT statement

insert into actor(first\_name, last\_name) values('MATIAS', 'SCHOENAERTS');

1. Controleer of MATIAS SCHOENAERTS goed toegevoegd werd met een SELECT statement. Hoe doe je dat? Welk ID heeft deze acteur gekregen? Id 201

select \* from actor  
order by last\_update desc

1. Welk alternatief INSERT statement kon je in vraag 5 ook gebruiken? Test het uit. Onze acteur zit dan tweemaal in de tabel actor.
2. Wijzig de voornaam van MATIAS SCHOENAERTS in MATTHIAS, met een UPDATE statement. Let op! Gebruik het ID om het juiste record te selecteren (en te wijzigen). Controleer het resultaat.

update actor  
 set first\_name = 'MATTHIAS'  
where actor\_id = 201;

1. Verwijder de twee records betreffende MATTHIAS SCHOENAERTS uit de tabel actor met een DELETE statement. Gebruik opnieuw het ID (of de ID’s) om de juiste records te selecteren (en te verwijderen). Controleer het resultaat. Kan je dat met 1 statement?

delete from actor  
where actor\_id = 201;