Napredne baze podataka

Vježbe

Domaća zadaća br. 2 –rješenja

U ovom dokumentu…

- Slike ekrana koje pokazuju što je trebalo napraviti u drugoj domaćoj zadaći
- Upit za izvoz podataka iz northwind baze podataka u formatu JSON dokumenata (jedan redak = jedan dokument)

- Datoteka orders.csv je preuzeta i nalazi se u direktoriju *working*

Priprema mongo kontejnera i uvoz podataka

1. *ls *csv* – potvrdimo da je datoteka ovdje

2. pokrenemo kontejner (ili docker start nbp-mongo)

3. potvrdimo da je kontejner pokrenut s *docker ps*

4. kopiramo datoteku u kontejner

5. uđemo u *shell* kontejnera

6. potvrdimo da je datoteka tu

7. importiramo podatke

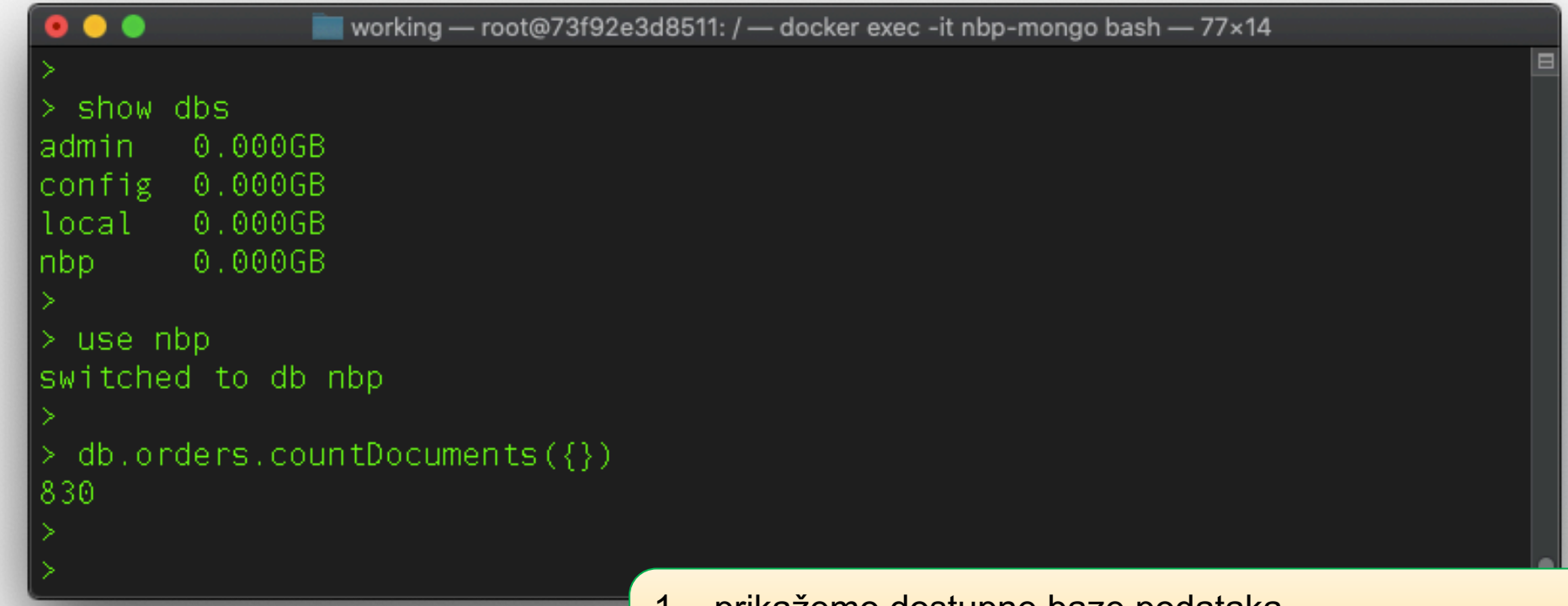
```
/ — docker exec -it nbp-mongo bash — 141×33
[working$ ls *csv
orders.csv
[working$ docker run --name nbp-mongo -d -p 27017:27017 -v /tmp:/tmp mongo
73f92e3d8511f3b0b58c253c207c4c47e28247bb9a8b1f8dc5cc1d3db4dfa9bc
[working$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                     NAMES
73f92e3d8511        mongo               "docker-entrypoint.s…"   5 seconds ago       Up 4 seconds        0.0.0.0:27017->27017/tcp  nbp-mongo
[working$
[working$ docker cp orders.csv nbp-mongo:/home
[working$
[working$ docker exec -it nbp-mongo bash
[root@73f92e3d8511:/#
[root@73f92e3d8511:/# l /home/*csv
/home/orders.csv
[root@73f92e3d8511:/# mongoimport -d nbp -c orders --file /home/orders.csv
2020-04-13T18:35:23.047+0000    connected to: mongodb://localhost/
2020-04-13T18:35:23.167+0000    830 document(s) imported successfully. 0 document(s) failed to import.
root@73f92e3d8511:/#
```

Pokretanje mongo shella

```
2020-04-13T18:35:23.167+0000    830 document(s) imported successfully. 0 document(s) failed to import.
root@73f92e3d8511:/#
root@73f92e3d8511:/# mongo
MongoDB shell version v4.2.3
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("8f90d52b-b19c-42d2-bb26-7f81f0782a58") }
MongoDB server version: 4.2.3
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        http://docs.mongodb.org/
```

Koristimo samo *mongo* naredbu, nakon čega se ispiše početni tekst…

Provjera učitanih podataka



1. prikažemo dostupne baze podataka
2. spojimo se na nbp bazu
3. provjerimo koliko dokumenata ima u kolekciji *orders*

Što tu ima? (findOne)

Iz ovoga shvatimo implicitnu shemu podataka, imena atributa i njihovu organizaciju. Npr. znamo da do cijene dolazimo s *details.unit_price*

working — root@73f92e3d8511: / — docker exec -it nbp-mongo bash — 82×50

```
> db.orders.findOne()
{
        "_id" : ObjectId("5e94b0eb531c980301e6a7cf"),
        "order_id" : 10248,
        "order_date" : "1996-07-04",
        "required_date" : "1996-08-01",
        "shipped_date" : "1996-07-16",
        "customer" : {
                "name" : "Vins et alcools Chevalier",
                "city" : "Reims",
                "country" : "France"
        },
        "shipper" : {
                "id" : 3,
                "name" : "Federal Shipping"
        },
        "details" : [
                {
                        "product" : {
                                "id" : 11,
                                "name" : "Queso Cabrales",
                                "category" : "Dairy Products"
                        },
                        "unit_price" : 14,
                        "quantity" : 12,
                        "discount" : 0
                },
                {
                        "product" : {
                                "id" : 42,
                                "name" : "Singaporean Hokkien Fried Mee",
                                "category" : "Grains/Cereals"
                        },
                        "unit_price" : 9.80000019,
                        "quantity" : 10,
                        "discount" : 0
                },
                {
                        "product" : {
                                "id" : 72,
                                "name" : "Mozzarella di Giovanni",
                                "category" : "Dairy Products"
                        },
                        "unit_price" : 34.7999992,
                        "quantity" : 5,
                        "discount" : 0
                }
        ]
}
>
```

Zadatak 1.

- Dohvatiti sve podatke narudžbe 11054

working — root@73f92e3d8511: / — docker exec -it nbp-mongo bash — 82×41

```
> db.orders.findOne( {"order_id" : 11054 } )
{
        "_id" : ObjectId("5e94b0eb531c980301e6aaf5"),
        "order_id" : 11054,
        "order_date" : "1998-04-28",
        "required_date" : "1998-05-26",
        "shipped_date" : null,
        "customer" : {
                "name" : "Cactus Comidas para llevar",
                "city" : "Buenos Aires",
                "country" : "Argentina"
        },
        "shipper" : {
                "id" : 1,
                "name" : "Speedy Express"
        },
        "details" : [
                {
                        "product" : {
                                "id" : 33,
                                "name" : "Geitost",
                                "category" : "Dairy Products"
                        },
                        "unit_price" : 2.5,
                        "quantity" : 10,
                        "discount" : 0
                },
                {
                        "product" : {
                                "id" : 67,
                                "name" : "Laughing Lumberjack Lager",
                                "category" : "Beverages"
                        },
                        "unit_price" : 14,
                        "quantity" : 20,
                        "discount" : 0
                }
        ]
}
>
```

Mogli smo koristiti i find() umjesto findOne(), samo ispis nije formatiran.

Formatiranje svakog ispisa se dobiva dodavanjem funkcije .pretty() na kraj, npr: db.orders.find({"order_id": 11054}).pretty()

Zadatak 2.

- Dohvatiti podatke o kupcima, sortirano silazno po državi i uzlazno po gradu, koji su kupovali mliječne proizvode (Dairy Products). Ispisati prvih 15 zapisa

working — root@73f92e3d8511: / — docker exec -it nbp-mongo bash — 147×20

```
> db.orders.find( {"details.product.category" : "Dairy Products"}, {"customer" : 1}).
... sort( {"customer.country" : -1, "customer.city" : 1} ).
... limit( 15 )
{ "_id" : ObjectId("5e94b0eb531c980301e6a7f2"), "customer" : { "name" : "LILA-Supermercado", "city" : "Barquisimeto", "country" : "Venezuela" } }
{ "_id" : ObjectId("5e94b0eb531c980301e6a7ff"), "customer" : { "name" : "LILA-Supermercado", "city" : "Barquisimeto", "country" : "Venezuela" } }
{ "_id" : ObjectId("5e94b0eb531c980301e6a820"), "customer" : { "name" : "LILA-Supermercado", "city" : "Barquisimeto", "country" : "Venezuela" } }
{ "_id" : ObjectId("5e94b0eb531c980301e6a83b"), "customer" : { "name" : "LILA-Supermercado", "city" : "Barquisimeto", "country" : "Venezuela" } }
{ "_id" : ObjectId("5e94b0eb531c980301e6a8f6"), "customer" : { "name" : "LILA-Supermercado", "city" : "Barquisimeto", "country" : "Venezuela" } }
{ "_id" : ObjectId("5e94b0eb531c980301e6aa0f"), "customer" : { "name" : "LILA-Supermercado", "city" : "Barquisimeto", "country" : "Venezuela" } }
{ "_id" : ObjectId("5e94b0eb531c980301e6aabc"), "customer" : { "name" : "LILA-Supermercado", "city" : "Barquisimeto", "country" : "Venezuela" } }
{ "_id" : ObjectId("5e94b0eb531c980301e6a7e2"), "customer" : { "name" : "GROSELLA-Restaurante", "city" : "Caracas", "country" : "Venezuela" } }
{ "_id" : ObjectId("5e94b0eb531c980301e6a956"), "customer" : { "name" : "LINO-Delicateses", "city" : "I. de Margarita", "country" : "Venezuela" } }
{ "_id" : ObjectId("5e94b0eb531c980301e6aa92"), "customer" : { "name" : "LINO-Delicateses", "city" : "I. de Margarita", "country" : "Venezuela" } }
{ "_id" : ObjectId("5e94b0eb531c980301e6a861"), "customer" : { "name" : "HILARION-Abastos", "city" : "San Cristóbal", "country" : "Venezuela" } }
{ "_id" : ObjectId("5e94b0eb531c980301e6a8bd"), "customer" : { "name" : "HILARION-Abastos", "city" : "San Cristóbal", "country" : "Venezuela" } }
{ "_id" : ObjectId("5e94b0eb531c980301e6a8c0"), "customer" : { "name" : "HILARION-Abastos", "city" : "San Cristóbal", "country" : "Venezuela" } }
{ "_id" : ObjectId("5e94b0eb531c980301e6a8fe"), "customer" : { "name" : "HILARION-Abastos", "city" : "San Cristóbal", "country" : "Venezuela" } }
{ "_id" : ObjectId("5e94b0eb531c980301e6a930"), "customer" : { "name" : "HILARION-Abastos", "city" : "San Cristóbal", "country" : "Venezuela" } }
>
```

Sortiranje je trebalo obaviti ovim redoslijedom.

Zadatak 2.

Nije greška ako je ispisan _id, a preglednije je bez njega.

- Dohvatiti podatke o kupcima, sortirano silazno po državi i uzlazno po gradu, koji su kupovali mliječne proizvode (Dairy Products). Ispisati prvih 15 zapisa

working — root@73f92e3d8511: / — docker exec -it nbp-mongo bash — 114×19

```
> db.orders.find( {"details.product.category" : "Dairy Products"}, {"customer" : 1, _id : 0}).
... sort( {"customer.country" : -1, "customer.city" : 1} ).
... limit( 15 )
{ "customer" : { "name" : "LILA-Supermercado", "city" : "Barquisimeto", "country" : "Venezuela" } }
{ "customer" : { "name" : "LILA-Supermercado", "city" : "Barquisimeto", "country" : "Venezuela" } }
{ "customer" : { "name" : "LILA-Supermercado", "city" : "Barquisimeto", "country" : "Venezuela" } }
{ "customer" : { "name" : "LILA-Supermercado", "city" : "Barquisimeto", "country" : "Venezuela" } }
{ "customer" : { "name" : "LILA-Supermercado", "city" : "Barquisimeto", "country" : "Venezuela" } }
{ "customer" : { "name" : "LILA-Supermercado", "city" : "Barquisimeto", "country" : "Venezuela" } }
{ "customer" : { "name" : "LILA-Supermercado", "city" : "Barquisimeto", "country" : "Venezuela" } }
{ "customer" : { "name" : "GROSELLA-Restaurante", "city" : "Caracas", "country" : "Venezuela" } }
{ "customer" : { "name" : "LINO-Delicateses", "city" : "I. de Margarita", "country" : "Venezuela" } }
{ "customer" : { "name" : "LINO-Delicateses", "city" : "I. de Margarita", "country" : "Venezuela" } }
{ "customer" : { "name" : "HILARION-Abastos", "city" : "San Cristóbal", "country" : "Venezuela" } }
{ "customer" : { "name" : "HILARION-Abastos", "city" : "San Cristóbal", "country" : "Venezuela" } }
{ "customer" : { "name" : "HILARION-Abastos", "city" : "San Cristóbal", "country" : "Venezuela" } }
{ "customer" : { "name" : "HILARION-Abastos", "city" : "San Cristóbal", "country" : "Venezuela" } }
{ "customer" : { "name" : "HILARION-Abastos", "city" : "San Cristóbal", "country" : "Venezuela" } }
>
```

Zadatak 3.

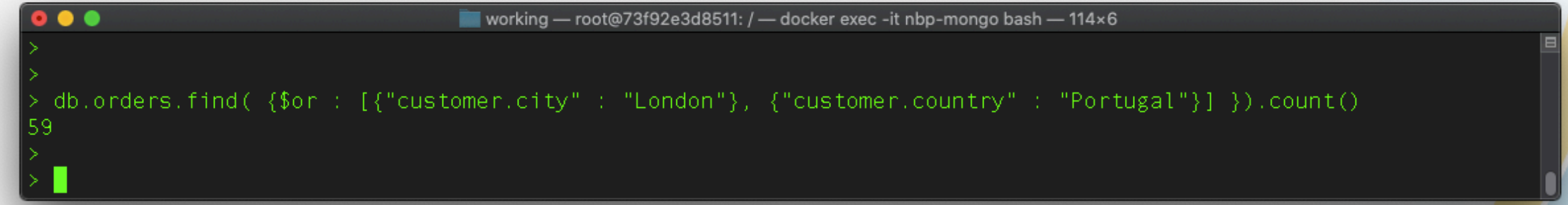
- Dohvatiti narudžbe u kojima je kupljeno više od 120 istih proizvoda. Ispisati identifikator narudžbe i ime kupca

Zadatak 4.

- Obrisati sve narudžbe iz Švedske

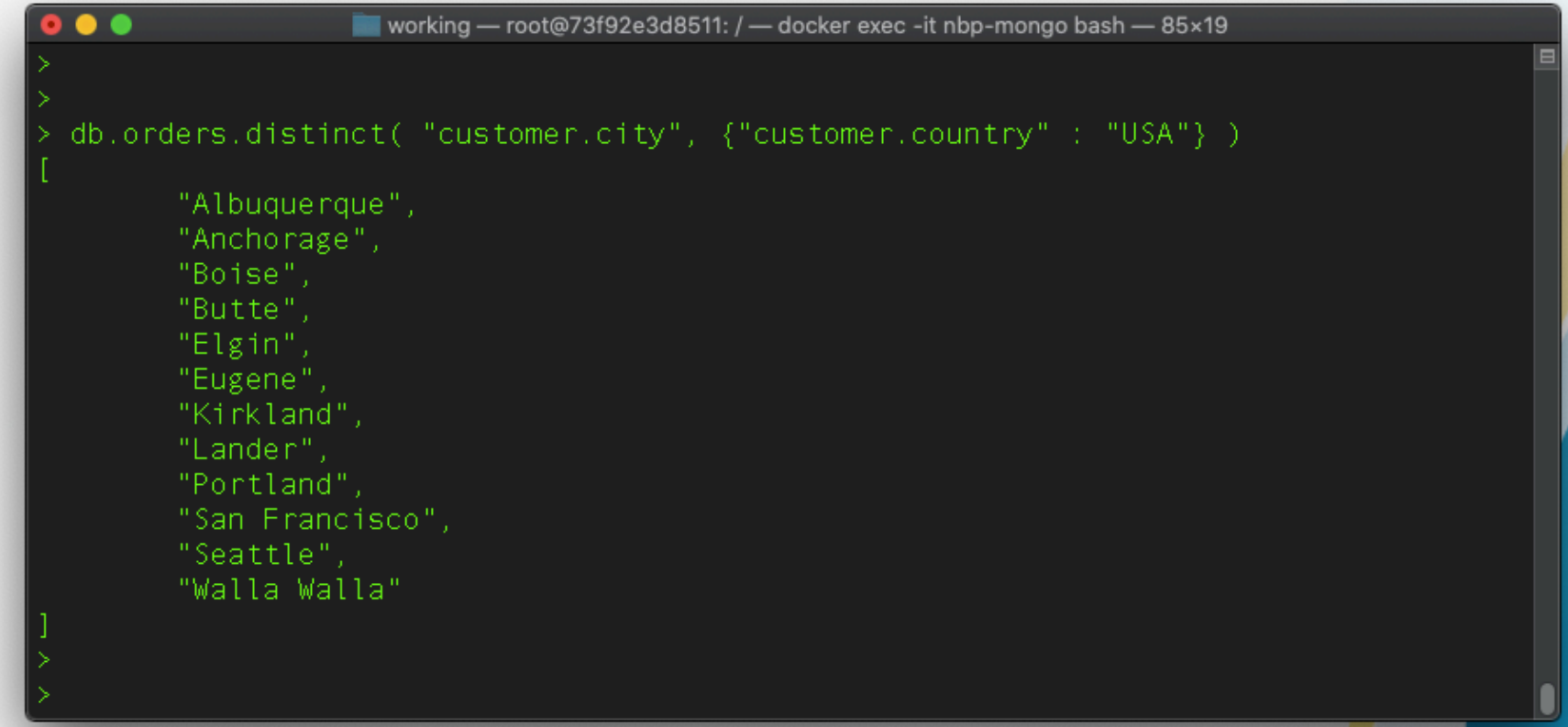
Zadatak 5.

- Dohvat broja svih narudžbi koje su napravili kupci iz Londona ili Portugala



Zadatak 6.

- Dohvat svih gradova iz USA iz kojih su pristizale narudžbe. Svaki grad se treba pojaviti samo jednom u listi.



Zadatak 7.

- Promijeniti naziv države kupca iz "USA" u "SAD" (hint: updateMany)

Nakon updateMany, za kontrolu, pokušamo dohvatiti sve iz USA i prebrojimo sve iz SAD

(!) Općenito je dobra praksa uvijek provjeriti sve modifikacije (i unose i brisanja) podataka.

SQL upit u postgresu za izvoz podataka

- Koristimo postgres funkciju *row_to_json* koja pojedini redak formatira u JSON
- Zbog toga imenujemo svaki atribut u projekciji, kako bi dobio to ime u JSON-u
- S obzirom na to da su neki atributi objekti za sebe ili su polja, tako i *row_to_json* koristimo više puta unutar vanjskog poziva
- Koristimo i funkciju *array_agg* koja agregira dobivene vrijednosti u polje te funkciju *array_to_json*, koja formatira polje u JSON
- Koristimo sintaksu `COPY (…) TO 'filename' …` za spremanje dobivenog u datoteku

SQL upit u postgresu za izvoz podataka

```
COPY (
    SELECT row_to_json(whole_row)
    FROM (SELECT o.order_id            AS "order_id"
               , order_date            AS "order_date"
               , required_date         AS "required_date"
               , shipped_date          AS "shipped_date"

               , (SELECT row_to_json(customer_row)
                  FROM (SELECT company_name AS "name",
                               city         AS "city",
                               country      AS "country"
                        FROM customers
                        WHERE customer_id = o.customer_id
                       ) customer_row) AS "customer"

               , (SELECT row_to_json(shipper_row)
                  FROM (SELECT shipper_id   AS "id"
                             , company_name AS "name"
                        FROM shippers
                        WHERE shipper_id = o.ship_via
                       ) shipper_row)  AS "shipper"

               , (SELECT array_to_json(array_agg(row_to_json(details_row)))
                  FROM (SELECT (SELECT row_to_json(product_row)
                                FROM (SELECT product_id    AS "id"
                                           , product_name  AS "name"
                                           , category_name AS "category"
                                      FROM products
                                           JOIN categories c ON products.category_id = c.category_id
                                      WHERE product_id = od.product_id) product_row) AS "product"
                             , od.unit_price                                         AS "unit_price"
                             , quantity                                              AS "quantity"
                             , discount                                              AS "discount"
                        FROM order_details od
                        WHERE o.order_id = od.order_id
                       ) details_row)  AS "details"
          FROM orders o) whole_row
    ) TO '/tmp/orders.csv' WITH (FORMAT TEXT , HEADER FALSE, DELIMITER E'\t');
```

Stay tuned…