

# Übung Northwind

## Einrichten des Projekts

### *Schritte zur Projektkonfiguration*

- In der Datei `application.properties` wurde folgende Zeile auskommentiert:  
`# quarkus.hibernate-orm.database.generation=drop-and-create`  
Dadurch sind wir selbst für die Erstellung der Datenbank verantwortlich!

### *Datenbank-Explorer einrichten*

- Öffnen Sie die bereitgestellte Datei `datasource.txt` und kopieren Sie deren Inhalt in die Zwischenablage.
- Wechseln Sie rechts in den Abschnitt `Database`. Klicken Sie dort auf das Symbol `+` und wählen Sie in dem erscheinenden Menu `Import from Clipboard`.

## Datenbank befüllen und JPA-Entities generieren

### *Datenbank mit Testdaten befüllen*

- Im Ordner `northwind_db_files` finden Sie SQL-Dateien zum Anlegen der Datenbankstruktur und zum Befüllen der Tabellen
- Rechtsklick auf die Datenbankverbindung und Menueintrag `Run SQL Script...` auswählen
- Wählen Sie nun die Datei ``Derby_Northwind_1_CreateObjects.sql`` aus. Dadurch werden die Tabellen in der Datenbank erstellt
- Wiederholen Sie nun den letzten Schritt für die Datei `Derby_Northwind_2_InsertData.sql`. Dadurch werden die Testdaten importiert

### *Möglichkeit, um JPA-Entities zu erstellen (wurde im Startprojekt bereits durchgeführt)*

- Rechtsklick auf die Datenbankverbindung und Menueintrag `Generate Persistence Mapping` auswählen
- Konfigurieren Sie das Mapping nach folgenden Regeln:
  - Einzahl verwenden (Orders → Order, Customers → Customer, ...)
  - Entfernen der bidirektionalen Beziehungen (die Beziehungsattribute, die jeweils eine Collection-Klasse referenzieren bitte abwählen)
  - Entfernen Sie die einfachen Beziehungsattribute (ID-Attribute) und wählen Sie stattdessen die Beziehungsattribute mit Objektreferenzen (z.B. Order: nicht CustomerID-Attribut einfügen sondern Customer-Referenz einfügen)
- Optional können Sie sich ein ERD generieren lassen, um eine Hilfestellung für die Abfragen zu haben.  
Wählen Sie dazu: `<Rechtsklick auf DB-Verbindung> → Diagrams → Show Visualization...`

# Reports erstellen

Erstellen Sie ein Repository mit dem Namen `ReportRepository`. Weiters erstellen Sie bitte eine REST-Resource mit dem Namen `ReportResource`. Mappen Sie die Url `/api/report/` auf diese Resource.

## Umsätze pro Jahr

Ermitteln Sie für jedes Jahr die Gesamtumsätze.

Addieren Sie dazu die entsprechenden Einträge aus `OrderDetails`. Berücksichtigen Sie auch die gewährten Rabatte (`Discount`).

*Schnittstelle*

### REST

GET `/api/report/revenue_per_year`

### Result

```
[
  {
    "year": 1996,
    "value": 208083.97
  },
  {
    "year": 1997,
    "value": 617085.2035
  },
  {
    "year": 1998,
    "value": 440623.866
  }
]
```

# ReOrder-Produkte

Ermitteln Sie alle Produkte, für die folgendes gilt:

- Sie werden noch weiter geführt (discontinued gleich 0)
- Produkte sollen nachbestellt werden (reorderlevel > 0)
- Lagerbestand + bestellte Produkte sind weniger als das ReOrderLevel

*Schnittstelle*

## REST

GET /api/report/reorder\_products

## Result

```
[
  {
    "productid": 30,
    "productname": "Nord-Ost Matjeshering",
    "quantityperunit": "10 - 200 g glasses",
    "unitprice": 25.89,
    "unitsinstock": 10,
    "unitsonorder": 0,
    "reorderlevel": 15,
    "discontinued": 0,
    "suppliersBySupplierid": { ... },
    "categoriesByCategoryid": { ... }
  },
  {
    "productid": 70,
    "productname": "Outback Lager",
    "quantityperunit": "24 - 355 ml bottles",
    "unitprice": 15.0,
    "unitsinstock": 15,
    "unitsonorder": 10,
    "reorderlevel": 30,
    "discontinued": 0,
    "suppliersBySupplierid": { ... },
    "categoriesByCategoryid": { ... }
  }
]
```

# Hersteller von Produkten einer Produktkategorie

Ermitteln Sie alle Hersteller von Produkten einer bestimmten Produktkategorie. Die Produktkategorie wird als Kategorienname übergeben.

*Schnittstelle*

**REST**

GET /api/report/suppliers\_for\_category/<categoryname> (Bsp: Seafood)

**Result**

```
[
  {
    "supplierid": 4,
    "companyname": "Tokyo Traders",
    "contactname": "Yoshi Nagase",
    "contacttitle": "Marketing Manager",
    "address": "9-8 Sekimai Musashino-shi",
    "city": "Tokyo",
    "region": "",
    "postalcode": "100",
    "country": "Japan",
    "phone": "(03) 3555-5011",
    "fax": "",
    "homepage": ""
  },
  {
    "supplierid": 6,
    "companyname": "Mayumi's",
    "contactname": "Mayumi Ohno",
    "contacttitle": "Marketing Representative",
    "address": "92 Setsuko Chuo-ku",
    "city": "Osaka",
    "region": "",
    "postalcode": "545",
    "country": "Japan",
    "phone": "(06) 431-7877",
    "fax": "",
    "homepage": "Mayumi's (on the World Wide
Web)#http://www.microsoft.com/accessdev/sampleapps/mayumi.htm#"
  }, ...
]
```

# Mitarbeiter mit der größten Anzahl an Bestellungen

Ermitteln Sie die Daten des Mitarbeiters, der die meisten Bestellungen erledigt hat.

*Schnittstelle*

## REST

GET /api/report/employee\_max\_orders

## Result

```
{
  "employeeid": 4,
  "lastname": "Peacock",
  "firstname": "Margaret",
  "title": "Sales Representative",
  "titleofcourtesy": "Mrs.",
  "birthdate": "1937-09-19",
  ...
}
```

# Gesamt-Umsatz pro Produktkategorie

Ermitteln Sie die gesamten gespeicherten Umsätze pro Produktkategorie (Tabelle OrderDetails)  
Discounts (Rabatte) können dabei ignoriert werden! Achten Sie darauf, dass evtl. vorkommende  
Produktkategorien ohne Umsätze bzw. ohne Produkte ebenfalls geliefert würden.  
Sortieren Sie das Ergebnis absteigend nach dem Umsatz.

*Schnittstelle*

**REST**

GET /api/report/revenue\_per\_category

## Result

```
[
  {
    "categoryname": "Beverages",
    "revenue": 286526.95
  },
  {
    "categoryname": "Dairy Products",
    "revenue": 251330.5
  },
  ....
]
```

# Ermitteln Sie für jede Region den Lieferdienst mit den meisten ausgelieferten Bestellungen

Ermitteln Sie für jede Region den Lieferdienst, der die meisten Bestellungen zugestellt hat. Sortieren Sie das Ergebnis aufsteigend nach der Bestell-Anzahl.

*Schnittstelle*

**REST**

GET /api/report/top\_shipper\_per\_region

**Result**

```
[
  {
    "regionDescription": "Southern",
    "companyName": "Federal Shipping",
    "count": 46
  },
  {
    "regionDescription": "Western",
    "companyName": "United Package",
    "count": 49
  },
  ...
]
```