Logical Database

Customer (customerID, name, startDate, endDate, shippingAddress)
Primary Key (customerID)

Employee (employeeNumber, name, department)
Primary Key (employeeNumber)

Franchise (customerID, negotiatedPrice, information)
Primary Key (customerID)
Foreign Key (customerID) references Customer(customerID)

HandelsOrder (employeeNumber, orderNumber)
Primary Key (employeeNumber, orderNumber)
Foreign Key (employeeNumber) references Employee(employeeNumber)
Foreign Key (orderNumber) references Order(orderNumber)

HandlesSki (employeeNumber, productID)
Primary Key (employeeNumber, productID)
Foreign Key (employeeNumber) references Employee(employeeNumber)
Foreign Key (productID) references Ski(productID)

OrderContent (productID, orderNumber)
Primary Key (productID, orderNumber)
Foreign Key (productID) references Ski(productID)
Foreign Key (orderNumber) references Order(orderNumber)

Orders (orderNumber, quantity, totalPrice, state, date)
Primary Key (orderNumber)

ProductionPlan (period, employeeNumber, skiID, numberOfSki)
Primary Key (period)
Foreign Key (employeeNumber) references Employee(employeeNumber)
Foreign Key (skiID) references SkiType (skiID)

Shipment (shipmentNumber, orderNumber, transporterID, customerID, shippingAddress, scheduledPickUpDate, state)
Primary Key (shipmentNumber)
Foreign Key (transporterID) references Transporter(transporterID)
Foreign Key (customerID) references Customer(customerID)
Foreign Key (orderNumber) references Orders(orderNumber)

Ski (productID, skiID, length, weight, state)
Primary Key (productID)
Foreign Key (skiID) references SkiType(skiID)

SkiType (skiID, type, model, temperature, gripSystem, description, historical, url, retailPrice)
Primary Key (skiID)

Store (customerID, negotiatedPrice)
Primary Key (customerID)
Foreign Key (customerID) references Customer(customerID)

TeamSkier (customerID, dateOfBirth, club, numbersOfSkisYearly)
Primary Key (customerID)
Foreign Key (customerID) references Customer(customerID)

Transporter (transporterID, name)
Primary Key (transporterID)

Endpoints

Table 1: Endpoints

END POINT	METHOD	URI	DESCRIPTION
Customer	GET	/customer/order?{date}	Gets a list with the customer's orders, with since filter.
Customer	GET	/customer/order/{orderNumber}	Gets a specific order.
Customer	GET	/customer/plan	Gets a four-week production plan.
Customer	POST	/customer/order	Place an order.
Customer	PUT	/customer/order/{orderNumber}	Request to split an order if not all skis in the order are ready.
Customer	DELETE	/customer/order{orderNumber}	Cancel an order.
Transporter	GET	/transporter/order	Get orders that are "ready".
Transporter	PUT	/transporter/order/{shipmentNumber}?setstate={state}	Change state when an order is picked up.
Employee	GET	/employee/order?state={state}	Get order based on state.
Employee	GET	/employee/order	Get available order.
Employee	POST	/employee/shipment	Make a shipment request.
Employee	POST	/employee/order	Add information to a newly produced ski.
Employee	POST	/employee/plan	Make a four-week production plan.
Employee	PUT	/employee/order/{orderNumber}?setstate={state}	Change order state.
Public	GET	/ski?model={model}	Get skis based on model.
Public	GET	/ski?gripsystem={gripsystem}	Get skis based on grip system.

RESOURCE	REPRESENTATION
Customers	[customer*]
Customer	{"customerID": "integer", "customerName": "string", "startDate": "date", "endDate": "date", "shippingAddress": "string"}
Transporters	[transporter*]
Transporter	{"transporterID": "integer", "name": "string"}
Employees	[employee*]
Employee	{"employeeNumber": "integer", "name": "string", "department": "string"}