



Namn på kolumn	DataTyp på kolumn	Kommentar
----------------	-------------------	-----------

Product

ID	smallint	
Name	nvarchar(50)	
Description	nvarchar(100)	
Price	int	
Producer	nvarchar(50)	
Weight	decimal(6,2)	
PackageInfoID	smallint	
WarehouseID	tinyint	
ProductDiscountID	int	tillåter null om ingen rabatt ges

Constraints:

- **Unique:** Name & Producer - varje producent har ett unikt produktnamn, men produktnamnet behöver inte vara unikt. Tex "cola".
- **Check:** LEN (Name) > 1
- **Check:** Price > 2

Exempeldata

ID	Name	Description	Price	Producer	Weight	PackageInfoID	WarehouseID	DiscountID
1	Cola	Cola zero	10	Coca Cola	0,5	1	1	1
2	Red Bull	Testy 250 cl	14	Red Bull	0,5	2	1	null

PackageInfo

ID	int	
Width	decimal(6,2)	
Height	decimal(6,2)	
Depth	decimal(6,2)	

Constraints:

- **Unique:** Width, Height och Depth - skall inte finnas olika storlekar som är samma

Exempeldata

ID	Width	Height	Depth
1	20	30	15

ProductDiscount

ID	int	
Type	nvarchar(15)	
Discount	int	
StartDate	date	
EndDate	date	

Constraints:

- **Unique:** Type, DiscountAmount, StartDate och EndDate - Ej gå att lägga in samma rabatt 2 gånger.
- **Check:** StartDate < EndDate
- **Check:** Type IN ('Fixed', 'Percentage')

Exempeldata

ID	Type	Discount	StartDate	EndDate
1	Fixed	1	2019-12-01	2020-01-20
2	Percentage	10	2019-10-01	2019-12-30

Warehouse

ID	tinyint	
WebshopID	tinyint	Unique Key 1-1 förhållande. Finns endast 1st Webshop

Exempeldata

ID	WebShopID
1	1

Webshop

ID	tinyint	
Name	Varchar(20)	Unique Key

Constraints:

- **Unique:** Name - Finns bara 1 webshop
- **Check:** Name = 'SaMic Webshop'

Exempeldata

ID	Name
1	SaMic Webshop

Customer

ID	int	
FirstName	nvarchar(20)	
LastName	nvarchar(20)	
PersonalIDNr	bigint	personnr
UserName	nvarchar(30)	
EMail	nvarchar(300)	kontroll av korrekt emailformat sker i program, mer flexibelt
MembershipDate	date	
FavouriteStoreID	smallint	
WebshopID	tinyint	
MembershipType	varchar(20)	
PhoneNr	int	Tillåt null, valfritt fält för kunden att fylla i. Format justeras i program

Constraints:

- **Unique:** UserName - vid registrering/inloggning
- **Unique:** Email - profilinfo för att skicka beställningsbekräftelse/kvitto/registrering, mfl.
- **Check:** Len(FirstName) > 1
- **Check:** Len(LastName) > 1
- **Check:** Type IN ('Gold', 'Silver', 'Bronze')
- **Check:** LEN (PersonalIDNr) = 12

Exempeldata

ID	FirstName	LastName	PersonalIDNr	UserName	EMail	MembershipDate	FavouriteStoreID	WebshopID	MembershipType	PhoneNr
1	Samir	Ehsani	876789	Tjabo	g@example.com	2012-12-01	2	1	Gold	0705445566

WebOrder

ID	int	
Date	date	
CustomerID	int	
DiscountCodeID	int	Null tillåt, om rabattkoden ogiltig
TotalPrice	int	

Constraints:

- Unique: CustomerID och DiscountCodeID - förhindrar att en kund kan använda samma kod flera köp
- Check: TotalPrice > 1

Exempeldata

ID	Date	CustomerID	DiscountCodeID	TotalPrice
1	2019-12-20	1	1	2000
2	2020-01-02	2	null	200

DiscountCode

ID	int	
Code	varchar(30)	
StartDate	Date	
EndDate	Date	
Discount	tinyint	räcker då det är i %

Constraints:

- Unique: Code, StartDate och EndDate - Bör endast finnas en kod med samma datum
- Check: LEN (Code) > 2
- Check: StartDate < EndDate

Exempeldata

ID	Code	StartDate	EndDate	Discount
1	ABC123	2019-08-01	2020-01-01	20

OrderProduct

WebOrderID	int	
ProductID	int	
ProductAmount	smallint	

Constraints:

- Rimligt att program ser till att endast en unik produktrad (ProductID) med summering av ProductAmount läggs på samma rad, om anv. köper samma produkt i flera omgångar, för samma order. Ergo ej unik i databasen
- Check: Amount > 0

Exempeldata

WebOrderID	ProductID	ProductAmount
1	1	1

StoreStock

StoreID	tinyint	
ProductID	int	
Amount	smallint	

Constraints:

- Check: Amount > 0

Exempeldata

StoreID	ProductID	Amount
1	1	100

Store

ID	smallint	
Name	nvarchar(50)	
StreetName	nvarchar(100)	
StreetNumber	smallint	
City	nvarchar(50)	
PostNumber	int	
StoreSize	smallint	
ManagerID	smallint	Storlek beroende av Employee.ID

Constraints

- **Unique:** StreetName, StreetNumber, City - flera affärer kan inte finnas på samma adress
- **Unique:** ManagerID - 1-to-1 med Employee, endast 1 manager per affär
- **Check:** Name = 'SaMic' - alla butiker måste heta SaMic (en butikskedja, som tex HM, Zara, Inet mm.)

Exempeldata

ID	Name	StreetName	StreetNumber	City	PostNumber	StoreSize	ManagerID
1	SaMic	Kungsgatan	1	Gothenburg	45323	1000	11

Employee

ID	smallint	
FirstName	nvarchar(50)	
LastName	nvarchar(50)	
PersonalIDNr	bigint	Personnr
Role	nvarchar (20)	Befattning av anställning

Constraints

- **Unique:** FirstName, LastName och PersonalIDNr - kan finnas flera personer anställda med samma namn, men inte med samma personnr
- **Check:** Role IN ('Employee', 'Manager') - kan utökas om fler befattningar läggs till
- **Check:** LEN (FirstName) > 1 AND LEN (LastName) > 1

- **Check:** LEN (PersonalIDNr) = 12 - personnr måste vara 12 siffror

Exempeldata

ID	FirstName	LastName	PersonalIDNr	Role
1	Kalle	Kula	198408234671	Employee

EmployeeSchedule

EmployeeID	smallint	
StoreID	smallint	
Day	nvarchar(15)	veckodagar
WorkingTimeID	tinyint	

Constraints

- **Unique:** EmployeeID, StoreID, Day, WorkingTimeID - en anställd skall kunna jobba dubbelpass samma dag på samma butik, men inte samma pass flera gånger/ på olika butiker.
- **Check:** Day IN ('Monday', 'Tuesday' 'Sunday') - kan bara skriva in veckodagar

Exempeldata

EmployeeID	StoreID	Day	WorkingTimeID
1	1	Monday	1
1	2	Monday	2

WorkingTime

ID	tinyint	
StartTime	time(0)	(0) = format(20:00:00)
EndTime	time(0)	(0) = format(20:00:00)

Constraints

- **Unique:** StartTime, EndTime - kan ej lägga in schemarad som är exakt samma
- **Check:** StartTime < EndTime

Exempeldata

ID	StartTime	EndTime
1	06:00:00	14:00:00
2	14:00:00	22:00:00

OpeningHour

StoreID	smallint	
Day	nvarchar(15)	veckodagar
Open	time(0)	(0) = format(20:00:00)
Close	time(0)	(0) = format(20:00:00)

Constraints

- **Unique:** StoreID, Day, Open och Close - Samma butik skall ej kunna ha flera öppettider samma dag
- **Check:** Day IN ('Monday', 'Tuesday' 'Sunday') - kan bara skriva in veckodagar
- **Check:** StartTime < EndTime - öppningstid måste vara mindre än stängningstid

Exempeldata

StoreID	Day	Open	Close
4	Monday	09:00:00	20:00:00

Triggers

- **WebOrder** - Vi skapade en trigger för att kontrollera att datumet för WebOrder är Större än datumet för DiscountCode StartDate och mindre än EndDate. På så sätt kontrollerar om DiscountCode är giltig. Denna trigger behövs eftersom vi kontrollerar två tabeller som är kopplat direkt till varandra och det går inte med en check Constraints eftersom vi saknar join tabell.
- **OrderProduct** - En trigger som joinar WebOrder, Product samt ProductDiscount, och omvänt letar upp om WebOrder.Date är större eller mindre än slut-/startdatm för ProductDiscount. Behövdes för att kontrollera att produktrabatterna är giltiga när en kund beställer något från webshopen.
IF EXISTS (
SELECT *
FROM inserted
JOIN WebOrder ON inserted.WebOrderID = WebOrder.ID
JOIN Product ON inserted.ProductID = Product.ID
JOIN ProductDiscount ON Product.ProductDiscountID = ProductDiscount.ID
WHERE WebOrder.Date > ProductDiscount.EndDate OR WebOrder.Date < ProductDiscount.StartDate
)
- **Store** - Kontrollerar via omvänd WHERE att det endast går att ange ManagerID från Employee.ID till fältet Store.ManagerID. Annars kan man skriva in vilken anställd som helst.:
JOIN Employee ON inserted.ManagerID = Employee.ID
WHERE Role != 'Manager'
- **Product** - En trigger som JOINAR med ProductDiscount för att kontrollera där det är "Fixed" rabatt, att produktens värde - fixed rabattvärde inte understiger 1.