



- ✓ Konvertering av befintlig tabell till ny(a) tabeller
- ✓ Indata kontroller
Check-constraints, Lagrade Procedurer / Script



Bakgrund

Du har fått en eller flera tabeller som ska konverteras till en ny databas. Av den anledningen ska du analysera tabellen/rna och försöka hitta en ny förnuftig utformning/design på databasen.

- 1 Ta reda på vad varje fält används till så du kan bedöma innehållet inför konvertering.
- 2 Genomför normalisering 1NF-4NF.
- 3 Kontrollera datainnehåll och fundera över vilken förändring som kan vara förnuftig. Gör följande kontroller:
 - Hitta fält som har Nullvärden
 - Fält som har samma värden på många poster
 - Är fältet någon form av regelverk eller kan det betraktas som typtabell?
 - Är datatyperna bra eller bör de förändras? Längderna på fälten?
 - Är det något annat som du anser inte är bra?
- 4 Utveckla datamodell utifrån dina nya kunskaper.



Column Name	Data Type
ID	int
Enamn	nvarchar(30)
Fnamn	nvarchar(20)
Conamn	nvarchar(255)
Gata	nvarchar(40)
Nr	int
Uppgång	nvarchar(10)
Postnr	int
Ort	nvarchar(20)
Slutdatum	int
Betaltom	int
Indatum	int
Fastnr	nvarchar(255)
Telefon	nvarchar(16)
Epost	nvarchar(80)
Betalkod	nvarchar(4)
Tidning	tinyint
Andrahand	tinyint
Medlemskort	tinyint
Huvudkort	int
Erskort	int

1

Unik nyckel, atomära fält

2

Attributen ska vara beroende av **hela** nyckeln

3

Inga inbördes beroenden mellan icke nyckelfält

4

En tabell ska inte växa i bredd, samma data ska inte återkomma

Medlem



Skriv upp typiska frågeställningar som är viktiga för din analys.

Omsätt till SQL-satser och testkör mot databasen som ska konverteras.

Exempelvis:

1. Vilka fält är det som har Null värden och förekommer det ofta?
2. Finns det flera som har samma adress?
3. Hur många tecken motsvaras gatuadressen av som mest?
4. Finns det fält som vi kan "ersätta" med typtabell?

Medlem



Uttryck	Funktionalitet	Exempel
LEN(a)	Returnerar längden av a	LEN(namn)
LTRIM(a)	Tar bort blanktecken till vänster	LTRIM(namn)
RTRIM(a)	Tar bort blanktecken till höger	RTRIM(namn)
LEFT(a,pos)	Läser antal tecken från vänster	LEFT(postnr,3)
RIGHT(a,pos)	Läser antal tecken från höger	RIGHT(pnr,4)
CAST/CONVERT	Konverterar datatyp	CAST(Postnr as varchar(5))
ISNULL(a,b)	Om a är Null gäller b	ISNULL(uppgång, '-')
	LEN(LTRIM(RTRIM(gata)))	



Operator	Utför	Exempel
=	Lika med	A = B
<> !=	Inte lika med	A <> B, A!=B
<	Mindre än	10 < 20
<=	Mindre än eller lika med	10<=10
>	Större än	25>12
>=	Större än eller lika med	25>=24
BETWEEN	Mellan i området	BETWEEN 10 AND 20
IN	Finns i specifikation	7 IN (1,3,4,5,8,9)
IS NULL	Är Null	tel IS NULL
IS NOT NULL	Är inte Null	namn IS NOT NULL
LIKE med %	lika som 'många tecken'	postnr like '393%'
LIKE med _	lika som 'ett tecken'	postnr like '39_ 51'

Operator	Utför	Exempel
NOT eller !	INTE	Lon > 7000 AND NOT Ort='Kalmar'
AND eller &&	OCH	Ort='Kalmar' AND Namn Like '%son'
OR eller	ELLER	Lon <10000 OR Lon>14000
XOR	Exclusive OR	Select a XOR b
a XOR b är matematiskt lika med		a AND NOT b



Kontroll med hakparentes []

Hakparenteser [brackets] är användbara vid indatakontroller:

CHECK constraints

SQL satser i lagrade procedurer/triggers/script

Uttryck	Funktionalitet
Postnr like '[1-9][0-9][0-9][0-9][09]'	Sammanskrivet postnr och en siffra i varje position
Postnr like '[1-9][0-9][0-9] [0-9][09]'	Med en tomposition i 4:e positionen.
Postnr like '[3]%'	Börjar med en 3:a och därefter följer vad som helst som är tillåtet för fältet.
Postnr NOT like '[3]%'	Postnr börjar inte med en 3:a
Postnr like '[3][6-9]%'	Börjar med en 3:a. Andra siffran kan vara 6, 7, 8 eller 9.
Postnr like '%[3][9]%'	I postnr ska talen 39 finnas efter varandra. I början, mitten eller slutet.
Personnr like '[0-9][0-9][0-1][0-9][0-3][0-9]-[0-9HTR][0-9][0-9][0-9] '	



Kontroll med hakparentes []

I det fall man vill kontrollera ett innehåll som kan variera i längd är nedanstående modell tillämpar. Dubbla negationer.

Uttryck	Funktionalitet
Kategori like '[a-z][a-z][a-z][a-z][a-z]'	Fast textlängd kontrollerar tecknen i varje position.
Kategori like '%[a-z]%'	Det räcker med att tecknen a-z kan finnas i EN position
Kategori like '%[^a-z]%'	Tecknen a-z ska inte finnas i en position!! Omvänt.
Kategori NOT like '%[^a-z]%'	Tecknen a-z ska inte finnas i en position som inte ska finns i fältet! Omvänt - omvänt.