

## Kap6. Virtuelle kolonner, funksjoner og uttrykk

- Konstantverdier
- Aritmetiske uttrykk
- Funksjoner

## Virtuelle kolonner

SELECT-setningen kan “lage” kolonner som ikke finnes i tabellen.

Disse kalles for “virtuelle” kolonner og kan bestå av:

- ☞ Konstantverdier
- ☞ Aritmetiske uttrykk
- ☞ Funksjoner
- ☞ Kombinasjoner av disse

Virtuelle kolonner kan være med i select-listen (de kolonnene som skal skrives ut), og de kan også forekomme andre steder i SELECT-setningen. (WHERE, ORDER BY osv.)

EKSEMPEL:

```
SELECT LEVNR, VAREN, PRIS, PRIS*1.22  
FROM VARE_LEV
```

## Eksempler på virtuelle kolonner:

```
SQL> select varenavn,1,'hallo'
2 from vare;
```

VARENAVN	1	'HALL
Superbensi	1	hallo
Normalbensin	1	hallo
Diesel	1	hallo
Campingstol	1	hallo
Campingbord	1	hallo
Fotball	1	hallo
Tennisball	1	hallo
Karoseriskruer	1	hallo

8 rows selected.

Alfanumeriske konstanter står mellom apostrofer.

```
SQL> select levnr,pris,pris*1.2 bruttopris
2 from vare_lev
3 Order by bruttopris
```

LEVNR	PRIS	BRUTTOPRIS
3010	4.1	4.92
3010	4.02	4.824
3010	2.04	2.448
3011	4.1	4.92
3011	4.02	4.824
3011	2.02	2.424

Vanligvis setter vi et midlertidig merke på en virtuell kolonne. Midlertidig merke kan også benyttes i ORDER BY -delen..

## Kolonner og konstanter

En tekst i enkeltapostrofer blir alltid oppfattet som en konstant. Det samme gjelder numeriske verdier.

En tekst som ikke står i enkeltspostrofer, vil bli oppfattet som et kolonnenavn. Det samme gjelder en tekst i dobbeltapostrofer. (OBS: dobbelt- apostrofer oppfattes som en del av kolonnenavnet.)

EKSEMPEL:

```
SELECT 'VARENAVN', VARENAVN  
FROM VARE
```

## Funksjoner

Det finnes et antall funksjoner som kan brukes i SELECT-setninger. Her skal vi se på noen av dem. Funksjonene kan deles i fire grupper:

- ☞ Aritmetiske funksjoner
- ☞ Tekst-funksjoner
- ☞ Dato-funksjoner
- ☞ Gruppe-funksjoner

En funksjon har ett eller flere argumenter og returnerer en verdi.

Argumentene kan være konstanter, kolonnenavn, aritmetiske uttrykk, funksjoner eller kombinasjoner av dette.

## Bruk av funksjoner

Funksjonene brukes alle steder der en kan ha virtuelle kolonner.

EKSEMPEL:

```
SELECT PRIS, ROUND(SQRT(PRIS),2)
FROM VARE_LEV
WHERE SQRT(PRIS/2) > 2
```

## SQRT / ROUND

SQRT er en aritmetisk funksjon som beregner kvadratroten. Den har ett argument: den verdien (konstant, kolonne eller uttrykk) som vi vil ha beregnet kvadratroten av.

ROUND er en aritmetisk funksjon som gjør avrunding. Den har to argumenter: Den verdien som skal avrundes, og antall ønskede desimaler etter avrunding. Sistnevnte kan utelates hvis det er 0, dvs. avrunding til heltall.

Vi ser her at en funksjon eller et aritmetrisk uttrykk kan være argument til en funksjon.

Vi ser også hvordan funksjoner kan benyttes i ulike deler av SELECT-setningen.

## Noen tekst-funksjoner

☞ SUBSTR(tekst, startpos, lengde)

EKSEMPEL:

```
SELECT SUBSTR(VARENAVN,3,5)
FROM VARE
```

Plukker ut bokstav 3,4,5,6,7 fra varenavnet.

Hvis siste argument utelates, tas resten av strengen.

☞ UPPER(tekst) og LOWER(tekst)

Oversetter til hhv. store og små bokstaver.

☞ tekst || tekst

Konkatenerer (skjøter sammen) to tekster.



## NULL-value funksjonen

Vi har tidligere lært at felter som er NULL (dvs. ingen verdi) oppfører seg helt spesielt. De kan ikke testes med vanlige testoperatører, og hvis man bruker dem i aritmetiske beregninger, blir resultatet også NULL-value. Funksjonen NVL hjelper oss å håndtere NULL-verdier:

☞ NVL(arg1,arg2)

Hvis arg1 ikke er NULL, returneres arg1, men hvis det er NULL, returneres arg2.

Vi kan få en kolonne til “oppføre seg” som 0 eller et blankt tegn hvis den er NULL:

☞ NVL(kolonne,0)

☞ NVL(kolonne,' ')

EKSEMPEL:

```
SELECT NVL(land,'Norge')  
FROM LEV;
```

## DECODE

DECODE er en meget kraftig funksjon. Den har minimum 3 argumenter. Hvert argument kan være en konstant, en kolonne, eller et uttrykk som kan inneholde en funksjon.

☞ `DECODE(arg1,arg2,arg3,arg4)`

Hvis DECODE har 4 argumenter, virker den slik: Hvis  $arg1 = arg2$ , returneres  $arg3$ , ellers returneres  $arg4$ .

Med 6 argumenter virker den slik:

Hvis  $arg1 = arg2$ , returneres  $arg3$ , ellers hvis  $arg1 = arg4$ , returneres  $arg5$ , ellers returneres  $arg6$ .

Med flere argumenter virker det tilsvarende.

Decode kan også ha et odde antall argumenter (3,5,7,...).

Vi får da ingen standard-verdi og den returnerer NULL dersom ingen av sammenlikningene slår til.

## EKSEMPEL på bruk av DECODE

```
SQL> SELECT DECODE(VAREN,1104, 'XXX', VARENAVN) varenavn  
FROM VARE;
```

```
VARENAVN
```

```
-----  
Superbensin  
Normalbensin  
xxx  
Campingstol  
Campingbord  
Fotball  
Tennisball  
Karosseriskruer
```

```
8 rows selected.
```

```
SQL>
```

☞ Sammenlign med IF..THEN..ELSE i andre språk.

## Omgjøring av datatyper

SQL kan til en viss grad gjøre om CHAR-felter til numeriske. F.eks. kan et CHAR-felt godt brukes i beregninger dersom det inneholder en numerisk verdi. Hvis vi gjør en beregning med et CHAR-felt, får vi en feilmelding hvis feltet har et ikke numerisk innhold.

Strengene '123' og ' 123' er ikke like. Vi kan bruke funksjonen TO\_NUMBER:

```
TO_NUMBER('123') = TO_NUMBER(' 123')
```

Numeriske kolonner presenteres med så mange desimaler som de har. Ønsker vi en pent formattert utskrift, med fast antall desimaler og evt. markering for hver 1000, kan vi bruke TO\_CHAR.

☞ TO\_NUMBER(streng)

☞ TO\_CHAR(verdi, 'format')

### EKSEMPLER:

G for gruppe-skilletegn (vanligvis ".")  
D for desimalkarakter (vanligvis ",")

*sett format i NLS\_LANGUAGE  
; ORACLE.111*

```
SELECT *
FROM VARE
ORDER BY TO_NUMBER(VARENØ)

SELECT TO_CHAR(PRIS, '99G999D99')
FROM VARE LEV
```

*NLS\_LANGUAGE 'NorwegianMk' 0*

## Dato-kolonner

Kolonner definert som DATE inneholder dato og klokkeslett.

Datoer kan beregnes med aritmetiske uttrykk:

☞ dato + antall

☞ dato - antall

☞ dato - dato

og med dato-funksjoner:

☞ ADD\_MONTHS(dato,antall)

☞ MONTHS\_BETWEEN(dato,dato)

☞ LAST\_DAY(dato)

☞ NEXT\_DAY(dato,ukedag)

ROUND og TRUNC kan benyttes til fjerne klokkeslett.  
Spesielt nyttig når du beregner dato-dato.

## ADD MONTHS

```
select add_months (sysdate,2) from dual;
```

## MONTHS BETWEEN

```
select months_between  
      (to_date('06.10.1994','dd.mm.yyyy'),  
       to_date('10.12.1994','dd.mm.yyyy'))  
      from dual;
```

## LAST DAY

```
select last_day(sysdate) from dual;
```

```
select last_day(to_date ('051195','ddmmyy'))  
      from dual;
```

## NEXT DAY

```
select next_day (sysdate, 'mandag') from dual;
```

## FORMATERING MED TEKST

```
select to_char(sysdate,'Day "er dagen i dag"') Dag  
      from dual;
```

## DD.MM.YYYY

Standard datoformat er avhengig av hvilket språk som brukes. For norsk språk er formatet DD.MM.YYYY. Dersom språk ikke er angitt vil standard format være DD-MON-YY.

Det kan omformatteres ved hjelp av funksjonene

☞ TO\_CHAR (dato, 'format')

☞ TO\_DATE (streng, 'format')

På neste side er det en oversikt over hva som kan skrives inne i format-delen (dato-formatering).

Se ellers håndboken for nærmere beskrivelse.

Maskindatoen kan hentes ved hjelp av SYSDATE.

```
Select To_char(sysdate,'dd/mm/yy')  
from vare
```

```
Select sysdate from dual
```

## Dato-formattering

PICTURE-strengen i TO\_CHAR funksjonen bestemmer dato-formatet. Noen eksempler:

Kode (numerisk)	Forklaring	Kode (alfabetisk )	Forklaring
Y	Siste siffer i årstall	YEAR	Årstall stavet ut
YY	2 siste	YYSP	Staver tallet
YYY	3 siste	YYYSP	"
YYYY	4 siste	YYYYSP	"
SYYYY	Årstall med fortegn		
Q	Kvartal	MONTH	Måned stavet ut
MM	Månedsnr	MON	3-bokstavers forkortelse
IW	Ukenr	IWSP	Staver tallet
W	Ukenr i måneden	WSP	Staver tallet
DDD	Dagnr i året	DDDSP	"
DD	" i måneden	DDSP	"
D	Dagnr i uken	DAY	Ukedag
		DY	3-bokstavers fork.
HH	Time (1-12)	HHSP	Staver tallet
HH12	"		
HH24	Time 0-23		
MI	Minutt		
SS	Sekund		
SSSSS	Sekund etter midnatt	SSSSSP	Stavet ut (!)



- ☞ Skilletegn (/:-) og fritekst kan legges inn i masken.
- ☞ Fritekst må stå mellom dobbeltapostrofer ( " ).
- ☞ Alfabetiske koder: Store bokstaver i masken gir store bokstaver, små gir små. Kombinasjon er mulig.
- ☞ Format-masken som staver ut tall, fungerer bare på engelsk.

## Aritmetiske funksjoner

### Aritmetiske funksjoner

Funksjon	Forklaring
POWER(x,y)	Opphøyer x i y-te potens.
ROUND(nummer[,d])	Avrunder til d desimaler.
TRUNC(nummer[,d])	Trunkerer til d desimaler.
ABS(nummer)	Absolutt verdien.
SIGN(nummer)	+1 hvis > 0, -1 hvis < 0, 0 hvis 0.
MOD(num1,num2)	num1 modulo num2
SQRT(nummer)	Kvadratrot. NULL hvis arg. < 0
COS(x)	Cosinus av en vinkel x, x er vinkelen gitt i radianer
SIN(x)	Sinus av en vinkel x, x er vinkelen gitt i radianer
LN(x)	logaritemen til x, x > 0
TAN(x)	Tangens til en vinkel x, x er vinkelen gitt i radianer
EXP(x)	Opphøyer e i x-te potens

## Tekst-streng funksjoner

### Tekst-streng funksjoner.

Funksjon	Forklaring
streng1    streng2	Konkatenerer streng1 med streng2.
DECODE(col,c1,v1,def)	IF ... THEN ... ELSE ...
LENGTH(streng)	Lengden av strengen
SUBSTR(str,spos[,len])	Del-streng av str-strengen
INSTR(sstr,str[,n[,m]])	Startpos. til m-te forekomst av str i sstr, start søk i pos n
UPPER(streng)	Konverterer til store bokstaver
LOWER(streng)	Konvertere til små bokstaver.
TO_NUMBER(streng)	Fra CHAR til NUMBER.
TO_CHAR(nummer[,format])	Fra NUMBER/DATE til CHAR.
TO_DATE(streng[,format])	Fra CHAR/NUMBER til DATE.
SOUNDEX(streng)	Teste på "lik" uttale
VSIZE(streng)	Antall bytes som trengs for å lagre strengen.
LPAD(streng,len[,chr])	Høyre-justering (LEFT PAD).
RPAD(streng,len[,chr])	Venstre-justering (RIGHT PAD)
INITCAP(streng)	INITCAP('nILs dal a/s') = 'Nils Dal A/S'
TRANSLATE(str,m2,m1)	Oversetter eventuelle m2-tegn i str til m1-tegn.
REPLACE(streng,m2,m1)	Bytter ut m2 med m1 i str
LTRIM(streng[,m])	Fjerner alle tegn i mengde m fra venstre
RTRIM(streng[,m])	Fjerner alle tegn i mengde m fra høyre
GREATEST(liste)	Største verdi i liste.
LEAST(liste)	Minste verdi i liste.