

IBM DB2 Cube Views



Scenarier for forretningsmodellering

Versjon 8

Merk:

Før du bruker opplysningene i denne boken og det produktet det blir henvist til, må du lese "Merknader" på side 31.

Dette dokumentet inneholder informasjon som eies av IBM. Det leveres i henhold til lisensbetingelser og er opphavsrettslig beskyttet. Informasjonen i denne håndboken omfatter ingen produktgarantier, og eventuelle merknader i denne håndboken må ikke tolkes som garantier.

Du kan bestille IBM-publikasjoner elektronisk eller via IBM-representanten.

- Hvis du vil bestille publikasjoner elektronisk, går du til IBM Publications Center på www.ibm.com/shop/publications/order
- IBM-representanten finner du ved å gå til IBM Directory of Worldwide Contacts på www.ibm.com/planetwide

Hvis du vil bestille DB2-publikasjoner fra DB2 Marketing and Sales i USA eller Canada, må du ringe 1800-IBM-4YOU (425-4968).

Når du sender informasjon til IBM, gir du IBM en ikke-eksklusiv rett til å bruke eller distribuere informasjonen på den måten IBM mener er best, uten forpliktelser i noen retning.

© Copyright International Business Machines Corporation 2003. All rights reserved.

Innhold

Om denne boken	v	Kapittel 4. Telle antall Internett-bestillinger	19
Hvem boken er beregnet på	v	Detaljer for scenariet	19
Kapittel 1. Beregne flyt og verdi av lagerbeholdning i et varelager over tid.	1	Opprette målene	20
Detaljer for scenariet	1	Kapittel 5. Rangere salgstall	23
Opprette målene	3	Detaljer for scenariet	23
Kapittel 2. Korrelere annonseringskostnader og salg.	7	Opprette målene	25
Detaljer for scenariet	7	Kapittel 6. Bruke tidsdata lagret i faktatabellen til å opprette en tidsdimensjon	27
Opprette målene	7	Detaljer for scenariet	27
Kapittel 3. Beregne fortjeneste og fortjenestemargin for en butikk	13	Opprette attributtene og dimensjonen	29
Detaljer for scenariet	13	Merknader	31
Opprette målene	14	Varemerker	33

Om denne boken

Denne boken inneholder tilleggsinformasjon til *DB2 Cube Views Installerings- og brukerhåndbok*, og beskriver hvordan du kan bruke metadata fra DB2 Cube Views med eksempler fra praktisk forretningsdrift.

Hvem boken er beregnet på

Les denne boken hvis du er en databaseadministrator som arbeider med OLAP-metadata og DB2 Universal Database (DB2). Du bør kjenne til dette:

- OLAP-begreper, for eksempel stjerneskjemaer
- DB2 Cube Views-metadataobjekter, for eksempel kubemodeller, faktaobjekter, dimensjoner, kombineringer, mål og attributter

Kapittel 1. Beregne flyt og verdi av lagerbeholdning i et varelager over tid

En detaljist, XYZ Detalj, lagrer varene i en lagerbygning før varene blir sendt til en av butikkene for å selges. XYZ Detalj har data for tilstanden til lagerbeholdningen i varelageret over tid og ønsker å analysere disse dataene. Firmaet ønsker spesielt å undersøke to aspekter av varelageret:

- Flyten av varer inn og ut av lageret
- Verdien av varene som finnes i varelageret på et gitt tidspunkt

Det første aspektet, flyten av varer, ser på data over tid. Det andre aspektet, verdien av varene, tar et snapshot av varelageret på et bestemt tidspunkt.

Detaljer for scenariet

XYZ Detalj har en faktatabell med følgende kolonner som gjelder varelageret: QUANTITY_IN, QUANTITY_OUT, CURRENT_QUANTITY, PRODUCT_VALUE, PRODUCT_ID og TIME_ID. Disse dataene registreres i tabellen ukentlig. Databasen har også tabellene Product og Time. Et eksempel på et sett med tabelldata er vist i tabell 1.

Tabell 1.

PRODUCT_ID	TIME_ID	QUANTITY_IN	QUANTITY_OUT	CURRENT_QUANTITY	PRODUCT_VALUE
1234	1	5	0	5	5
1234	2	20	10	15	5
1234	3	10	20	5	5

PRODUCT_ID-verdien for hver av de tre eksempeldatapostene er den samme fordi en produkttype kan gå inn og ut av lageret flere ganger.

Databaseadministratoren for XYZ Detalj må opprette tre forskjellige mål:

Flow In

Modell for flyten av varer inn til lageret.

Flow Out

Modell for flyten ut av lageret.

Current Value

Modell for verdien av varene på et gitt tidspunkt.

Når de to første målene skal opprettes, Flow In og Flow Out, oppretter databaseadministratoren mål som er tilordnet kolonnene QUANTITY_IN og QUANTITY_OUT, og summerer dataene på tvers av alle dimensjoner. Dette kalles et fullstendig additivt mål fordi dataene samles kun ved hjelp av funksjonen SUM på tvers av alle dimensjoner. I tabell 2 ser du et eksempel på data for kolonnene QUANTITY_IN og QUANTITY_OUT i tre måneder for produktet med PRODUCT_ID 1234. Målene Flow In og Flow Out summerer disse månedlige verdiene og beregner totalt antall som kom inn og gikk ut av lageret i første kvartal.

Tabell 2. Beregning av eksempeldata for de fullstendig additive målene Flow In og Flow Out for PRODUCT_ID 1234

	Januar	Februar	Mars	1. kvartal
QUANTITY_IN	5	20	10	35
QUANTITY_OUT	0	10	20	30

Fullstendig additive mål er de enkleste og vanligste målene du oppretter, og de brukes ofte som byggestener i mer komplekse mål. For mål som er basert på numeriske kildedata, oppretter OLAP-senteret som standard et fullstendig additivt mål.

Når det tredje målet skal opprettes, Current Value, oppretter databaseadministratoren et beregnet mål som beregner verdien ved å multiplisere PRODUCT_VALUE med CURRENT_QUANTITY. Hvis for eksempel verdien til produktet med PRODUCT_ID=1234 er 5, blir målet Current Value for eksempeldataene slik det er vist i tabell 3.

Tabell 3. Beregning av eksempeldata for målet Current Value for PRODUCT_ID 1234

	Januar	Februar	Mars
CURRENT_QUANTITY	5	10	20
Current Value	25	50	100

Disse dataene må deretter samles på tvers av dimensjonene. Siden dette målet beregner verdien på et bestemt tidspunkt, er det imidlertid uten mening å summere på tvers av tidsdimensjonen. Samlingen summerer derimot dataene på tvers av produktdimensjonen og finner gjennomsnittet for dataene over tid. Dette kalles et semi-additivt mål fordi bare deler av samlingen omfatter funksjonen SUM.

Mål som beregner snapshotdata, data som representerer et bestemt tidspunkt (som en måneds lagerdata), er ofte semi-additive mål fordi det er uten mening å summere månedene i kvartaler. Hvis et produkt ligger på lager i hele kvartalet, blir dette produktet tatt med i CURRENT_QUANTITY-

snapshotdataene for varebeholdningen i hver av de tre månedene i kvartalet. Hvis CURRENT_QUANTITY-dataene summeres over tid, blir produktet som lå på lager i tre måneder, telt tre ganger. Som vist i tabell 4 er verdien 25 for første kvartal uten verdi for aktiviteten på lageret. Tabellen viser at det aldri var 25 produkter på lager, og beregningen av verdien 25 er uten mening.

Tabell 4. Beregning av eksempeldata for kolonnen CURRENT_QUANTITY med funksjonen SUM for tidsdimensjonen for PRODUCT_ID 1234

	Januar	Februar	Mars	1. kvartal
SUM(CURRENT_QUANTITY)	5	15	5	25

I stedet for å bruke funksjonen SUM på tvers av alle dimensjoner, kan du utføre andre samlingsfunksjoner som AVG, MIN og MAX for tidsdimensjonen. Med samme sett med eksempeldata for januar, februar og mars kan du bruke andre samlingsfunksjoner for tidsdimensjonen slik det er vist i tabell 5, for å opprette meningsfylte verdier for kvartalet. Målet Current Value kan representere gjennomsnittlig total verdi for varene som er lagret på lageret i løpet av kvartalet, eller maksimums- eller minimumsverdien på et hvilket som helst tidspunkt i løpet av kvartalet.

Tabell 5. Beregning av eksempeldata for kolonnen CURRENT_QUANTITY med funksjonene AVG, MAX og MIN for tidsdimensjonen for PRODUCT_ID 1234

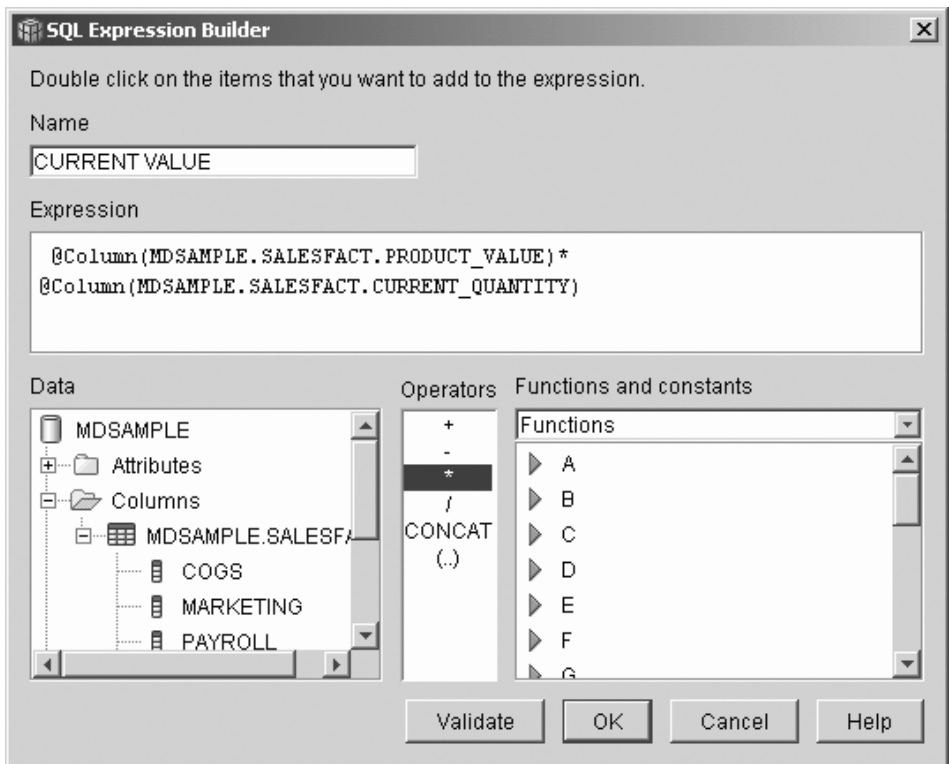
	Januar	Februar	Mars	1. kvartal
AVG(CURRENT_QUANTITY)	5	15	5	8,3
MAX(CURRENT_QUANTITY)	5	15	5	15
MIN(CURRENT_QUANTITY)	5	15	5	5

Opprette målene

Fremgangsmåten nedenfor beskriver hvordan du kan opprette målene Flow In, Flow Out og Current Value i et eksisterende faktaobjekt i vinduet Faktaegenskaper i OLAP-senteret:


1. Åpne vinduet Faktaegenskaper ved å høyreklikke på faktaobjektet i OLAP-senter-objektoversikten og klikke på **Rediger mål**. Vinduet Faktaegenskaper blir åpnet.
2. Opprett målet Flow In:
 - a. Klikk på **Opprett beregnet mål** på siden Mål. Vinduet Bygging av SQL-uttrykk blir åpnet.

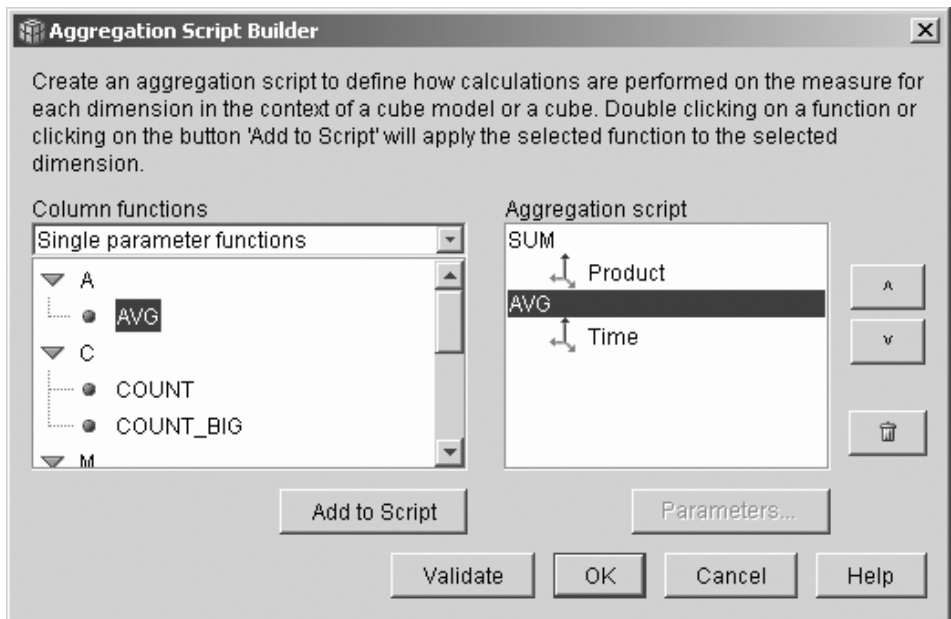
- b. Skriv FLOW IN i feltet **Navn** i vinduet Bygging av SQL-uttrykk.
 - c. Opprett Flow In-uttrykket:
 - Utvid mappen **Kolonner** og faktatabellen i listen **Data**.
 - Dobbeltklikk på kolonnen **QUANTITY_IN** for å tilføye den i uttrykket.
 - Klikk på **OK** for å lukke vinduet Bygging av SQL-uttrykk. Du trenger ikke å endre standard samlingsfunksjon, SUM, på siden Samlinger. Funksjonen SUM er standard for målet Flow In fordi datakilden er numerisk og målet refererer til en kolonne, ikke til bare eksisterende mål.
3. Opprett målet Flow Out:
- a. Klikk på **Opprett beregnet mål** på siden Mål. Vinduet Bygging av SQL-uttrykk blir åpnet.
 - b. Skriv FLOW OUT i feltet **Navn** i vinduet Bygging av SQL-uttrykk.
 - c. Opprett Flow Out-uttrykket:
 - Utvid mappen **Kolonner** og faktatabellen i listen **Data**.
 - Dobbeltklikk på kolonnen **QUANTITY_OUT**.
 - d. Klikk på **OK** for å lukke vinduet Bygging av SQL-uttrykk. Du trenger ikke å endre standard samlingsfunksjon, SUM, på siden Samlinger. Funksjonen SUM er standard for målet Flow Out fordi datakilden er numerisk og målet refererer til en kolonne, ikke til bare eksisterende mål.
4. Opprett målet Current Value:
- a. Klikk på **Opprett beregnet mål** på siden Mål. Vinduet Bygging av SQL-uttrykk blir åpnet.
 - b. Skriv CURRENT VALUE i feltet **Navn** i vinduet Bygging av SQL-uttrykk.
 - c. Opprett Current Value-uttrykket:
 - Utvid mappen **Kolonner** og faktatabellen i listen **Data**.
 - Dobbeltklikk på kolonnen **PRODUCT_VALUE** på listen **Data**.
 - Dobbeltklikk på operatoren * på listen **Operatorer**.
 - Dobbeltklikk på kolonnen **CURRENT_QUANTITY** på listen **Data**.
- I figur 1 på side 5 ser du Current Value-uttrykket du kan opprette i vinduet Bygging av SQL-uttrykk.



Figur 1. Fullført Current Value-uttrykk i vinduet Bygging av SQL-uttrykk

- d. Klikk på **OK** for å lukke vinduet Bygging av SQL-uttrykk.
- e. Klikk på samlingen for målet Current Value på siden Samlinger og velg **Samlingsskript** fra listen. Vinduet Bygging av samlingsskript blir åpnet. Standard samlingsskript har funksjonen SUM for alle dimensjoner.
- f. Hvis det er nødvendig, flytter du tidsdimensjonen nedover ved å velge

Tid og klikke på knappen  slik at det blir den siste dimensjonen som er oppgitt i skriptet. Mens tidsdimensjonen fortsatt er valgt, dobbeltklikker du på funksjonen **AVG** på listen **Kolonnefunksjoner**. Samlingsskriptet som er vist i figur 2 på side 6, summerer dataene på tvers av alle dimensjoner unntatt tid, mens det beregner gjennomsnitt for dataene over tid.



Figur 2. Samlingskript for målet Current Value

- g. Klikk på **Valider** i vinduet Bygging av samlingskript for å kontrollere at samlingskriptet er gyldig. Klikk på **OK** for å lagre samlingskriptet og lukke vinduet.
5. Klikk på **OK** for å lagre endringen i faktaobjektet og lukke vinduet Faktaegenskaper.

Du har nå tre beregnede mål for lagerbeholdningen i varelageret. Ved hjelp av disse tre målene kan du analysere mønsteret til produktflyten inn og ut av varelageret.

Kapittel 2. Korrelere annonseringskostnader og salg

En bilforhandler vurderer å bruke mer penger på annonsering. For å kunne treffe en velbegrunnet beslutning ønsker forhandleren først å analysere forholdet mellom tidligere annonseringskostnader og salg. Forhandleren er interessert i å se om forskjellige nivåer av annonsering har påvirket salget, og spesielt om økt annonsering er nært knyttet til økt salg.

Detaljer for scenariet

Forhandlerens database har en faktatabell med kolonnene Sales og Ad Costs. Databasen har dessuten flere andre dimensjonstabeller.

Databaseadministratoren kan opprette et mål som bruker DB2-funksjonen CORRELATION, for å utføre korrelasjonsberegninger mellom kostnadene og salget. Funksjonen CORRELATION er en funksjon med flere parametere, som krever to inndataparametere. I dette tilfellet bruker databaseadministratoren kolonnene Sales og Ad Costs som de to inndataparameterne.

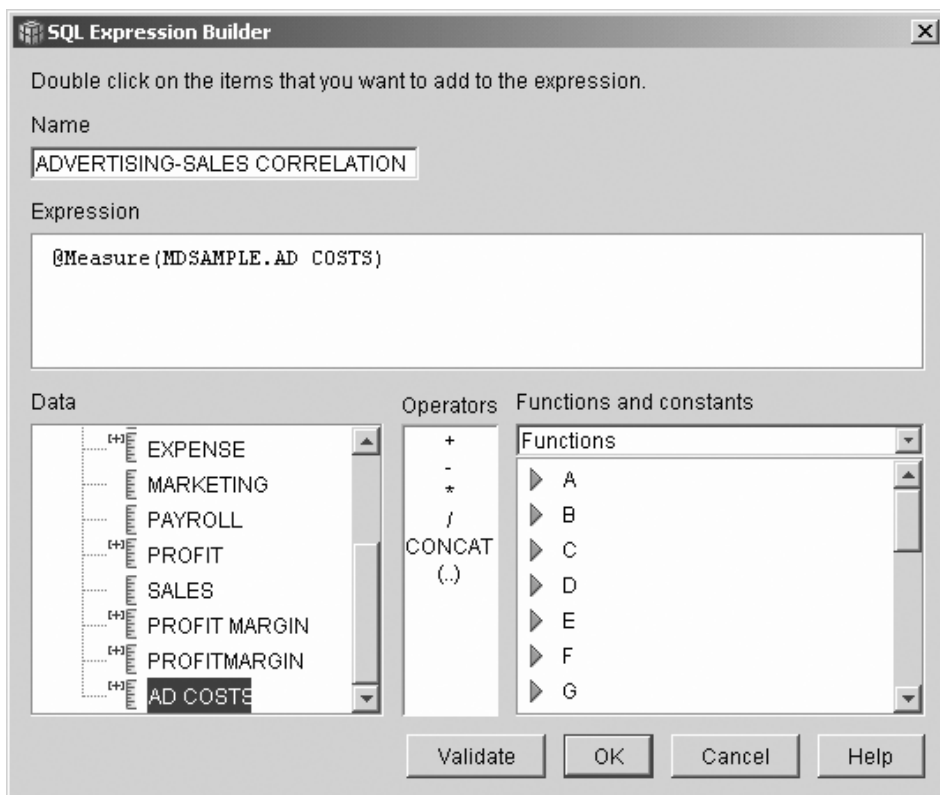
Databaseadministratoren må bruke samlingsfunksjonen med flere parametere først i samlingskriptet. Funksjonen med flere parametere kan brukes på tvers av alle dimensjonene, eller den kan brukes først for alle dimensjonene unntatt tidsdimensjonen, og en annen funksjon, for eksempel funksjonen MAX, kan brukes for tidsdimensjonen. Databaseadministratoren definerer SQL-uttrykket for målet slik at det er direkte tilordnet kolonnen Ad Costs. SQL-uttrykket er den første av de to parameterne som brukes i funksjonen med flere parametere. Databaseadministratoren definerer den andre parameteren som et SQL-uttrykk som er direkte tilordnet kolonnen Sales. Funksjonen CORRELATION er definert som den eneste samlingsfunksjonen slik at målet kan beregne den statistiske korrelasjonen mellom annonseringskostnader og salgsresultater på tvers av alle dimensjoner.

Opprette målene

Fremgangsmåten nedenfor beskriver hvordan du kan opprette målet Advertising-Sales Correlation i et eksisterende faktaobjekt i vinduet Faktaegenskaper i OLAP-senteret:

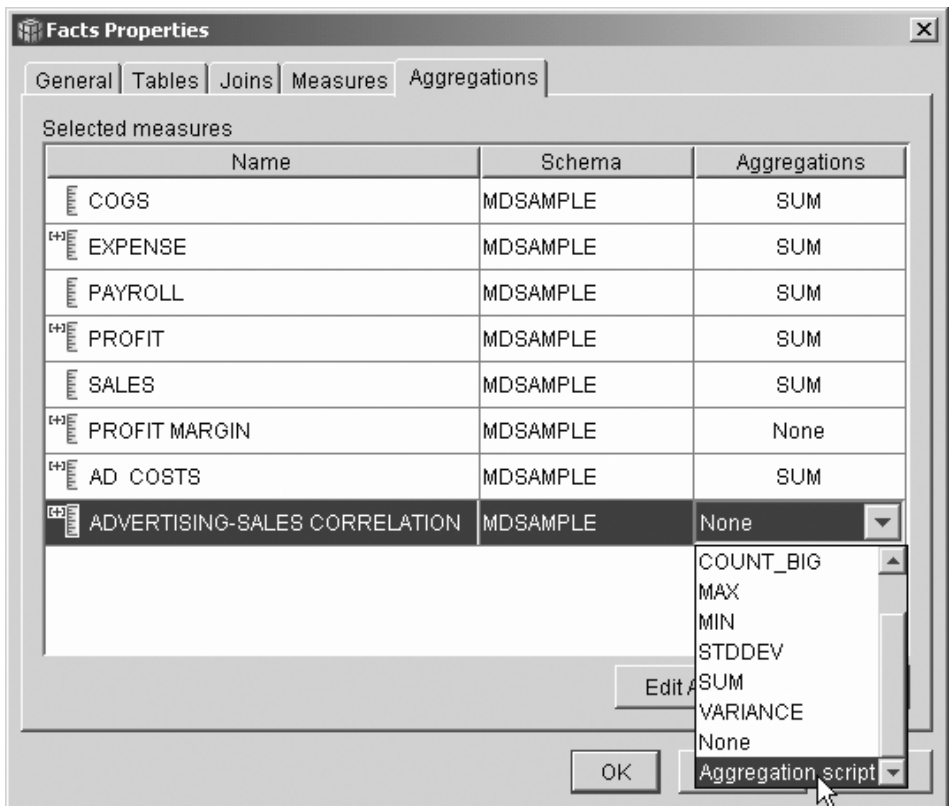
1. Åpne vinduet Faktaegenskaper ved å høyreklikke på faktaobjektet i OLAP-senter-objektoversikten og klikke på **Rediger mål**.
2. Klikk på knappen **Opprett beregnet mål**. Vinduet Bygging av SQL-uttrykk blir åpnet.
3. Skriv ADVERTISING-SALES CORRELATION i feltet **Navn** i vinduet Bygging av SQL-uttrykk.

4. Definer målets uttrykk, som også skal brukes som første parameter for funksjonen CORRELATION i samlingskriptet. Du definerer uttrykket ved å utvide mappen **Mål** i listen **Data** og dobbeltklikke på målet **AD COSTS** for å tilføye det til listen **Uttrykk**. I figur 3 ser du uttrykket du oppretter i vinduet Bygging av SQL-uttrykk.



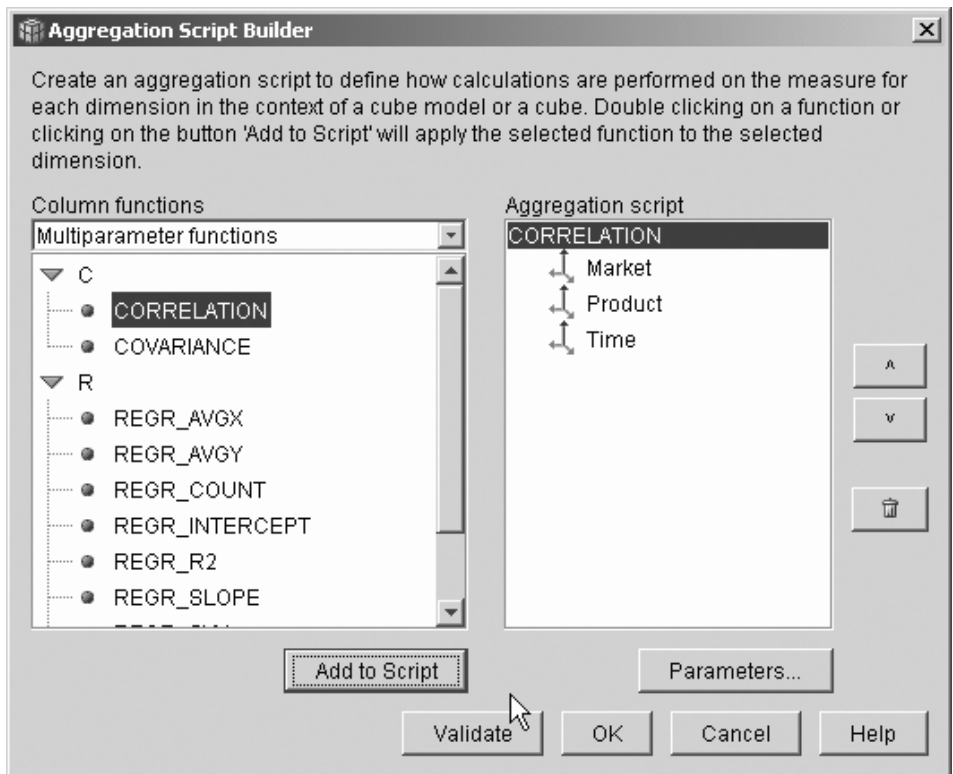
Figur 3. Fullført Advertising–Sales Correlation-uttrykk i vinduet Bygging av SQL-uttrykk

5. Klikk på samlingsfunksjonen for målet **ADVERTISING-SALES CORRELATION** på siden Samlinger og velg **Samlingskript**, slik det er vist i figur 4 på side 9. Vinduet Bygging av samlingskript blir åpnet.



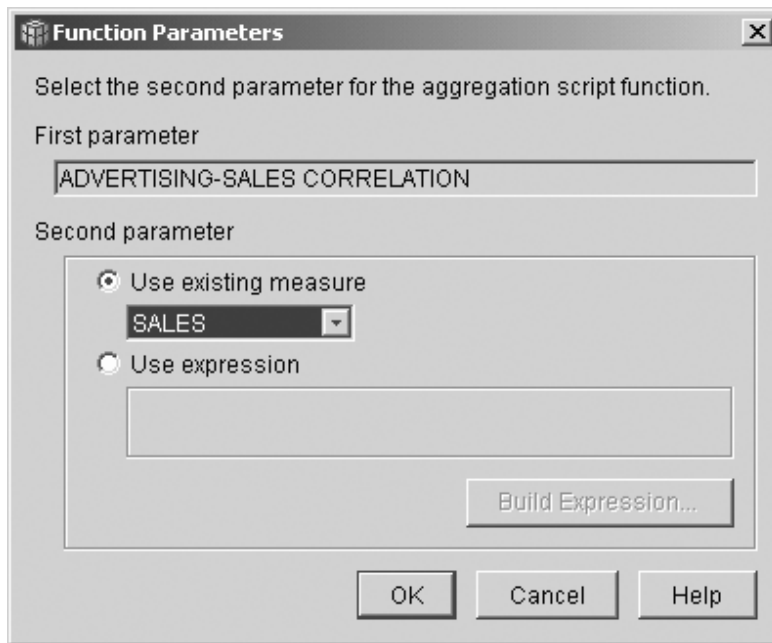
Figur 4. Siden Samlinger i vinduet Faktaegenskaper

6. Velg **Funksjoner med flere parametere** i feltet **Kolonnefunksjoner**. Velg funksjonen **CORRELATION** fra listen over funksjoner med flere parametere og klikk på **Tilføy i skript**. I figur 5 på side 10 ser du at funksjonen CORRELATION ligger øverst på listen over dimensjoner i skriptet.



Figur 5. Samlingskript for målet Advertising-Sales Correlation

7. Klikk på knappen **Parametere** for å oppgi den andre parameteren for funksjonen CORRELATION. Vinduet Funksjonsparametere blir åpnet. Merk **Bruk eksisterende mål** og velg **SALES**. I figur 6 på side 11 ser du vinduet Funksjonsparametere med målet Sales oppgitt som andre parameter. Klikk på **OK** for å lagre det du har valgt og lukke vinduet Funksjonsparametere.



Figur 6. Målet Sales oppgitt som andre parameter i vinduet Funksjonsparametere

8. Klikk på **Valider** i vinduet Bygging av samlingskript for å kontrollere at samlingskriptet er gyldig. Klikk på **OK** for å lagre samlingskriptet og lukke vinduet.
9. Klikk på **OK** for å lagre endringen i faktaobjektet og lukke vinduet Faktaegenskaper.

Du har nå et mål som korrelerer to typer av data i databasen. Ved hjelp av dette målet kan du treffe beslutninger om fremtidig bruk av annonsering på grunnlag av historiske resultater.

Kapittel 3. Beregne fortjeneste og fortjenestemargin for en butikk

Butikksjefen i en leketøysbutikk vil gjerne kunne analysere hvordan forskjellige faktorer som tid på året og type produkt påvirker fortjenesten og fortjenestemarginen. Før en mer avansert analyse kan utføres, må databaseadministratoren for leketøysbutikken opprette målet Profit for fortjeneste og Profit Margin for fortjenestemargin. Deretter kan databaseadministratoren opprette flere mål som korrelerer og sammenlikner forskjellige faktorer med målene Profit og Profit Margin.

Detaljer for scenariet

Leketøysbutikkens database har en faktatabell med kolonnene Sales, Costs of Goods Sold (COGS) og Expense i tillegg til tilsvarende fremmednøkkelkolonner for hver av de mange dimensjonstabellene. Databaseadministratoren har allerede opprettet målene Sales, COGS og Expense, som er tilordnet kolonnene Sales, COGS og Expense. Målene Profit og Profit Margin kan opprettes i sin helhet fra disse eksisterende målene.

Når databaseadministratoren skal opprette målet Profit, oppretter han et mål som beregner $\text{SALES} - (\text{COGS} + \text{EXPENSE})$ i SQL-uttrykket og summerer de beregnede dataene på tvers av alle dimensjoner. Målet Profit kan opprettes ved at det blir referert til eksisterende mål eller kolonner, eller til begge deler.

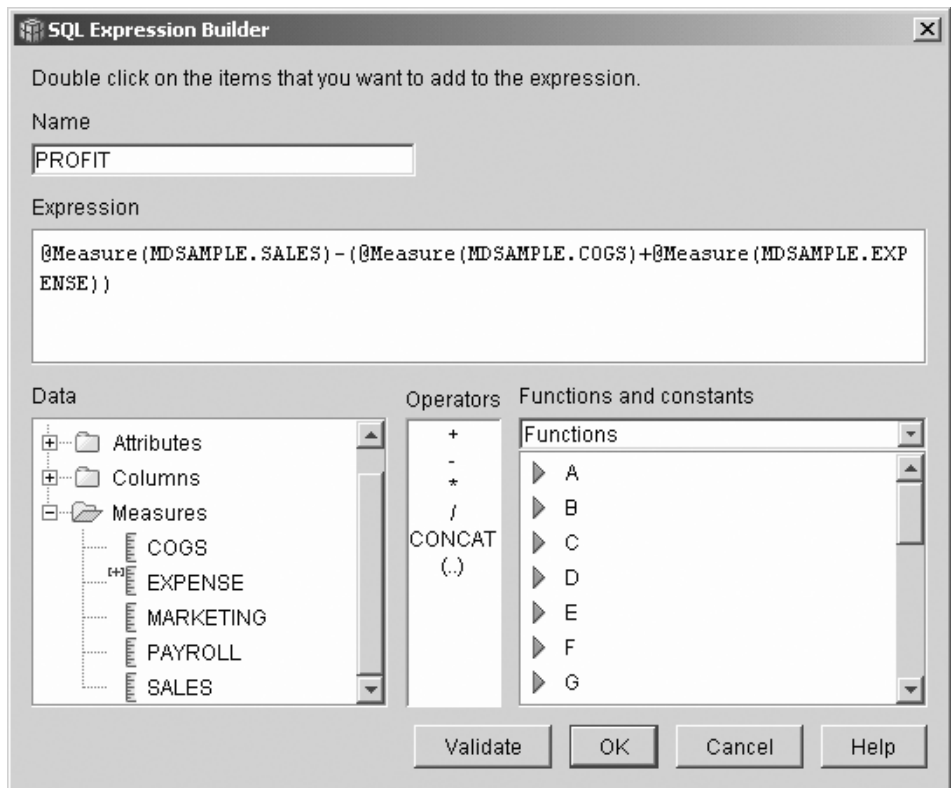
Når målet Profit er opprettet, kan databaseadministratoren opprette målet Profit Margin. Målet Profit Margin er et forhold mellom to eksisterende mål uttrykt som en prosentdel $(\text{Profit} / \text{Sales}) * 100$, og det krever ikke en egen samlingsfunksjon. Det er ikke nødvendig med en samlingsfunksjon fordi målet bare refererer til to andre mål der dataene allerede er samlet. Hvis databaseadministratoren bruker et sammensatt mål, et mål som bare refererer til andre mål, til beregningen av et forhold, trenger ikke databaseadministratoren å definere en ekstra samling. De fleste samlingsfunksjonene, for eksempel SUM, kan ikke brukes sammen med forhold. Hvis for eksempel leketøysbutikken har fortjenestemarginer på 40 %, 32 %, 28 % og 37 % i fire påfølgende kvartaler, ville en summering av forholdene gi en fortjenestemargin på 137 % for året, noe som er uten mening.

Opprette målene

Fremgangsmåten nedenfor beskriver hvordan du kan opprette målene Profit og Profit Margin i et eksisterende faktaobjekt i vinduet Faktaegenskaper i OLAP-senteret:

1. Åpne vinduet Faktaegenskaper ved å høyreklikke på faktaobjektet i OLAP-senter-objektoversikten og klikke på **Rediger mål**. Vinduet Faktaegenskaper blir åpnet.
2. Opprett målet Profit:
 - a. Klikk på knappen **Opprett beregnet mål** på siden Mål. Vinduet Bygging av SQL-uttrykk blir åpnet.
 - b. Skriv PROFIT i feltet **Navn** i vinduet Bygging av SQL-uttrykk.
 - c. Utvid mappen **Mål** i listen **Data** og utfør disse trinnene for å opprette Profit-uttrykket:
 - Dobbeltklikk på målet **SALES** på listen **Data** for å tilføye det til uttrykket.
 - Dobbeltklikk på operatoren – på listen **Operatorer**.
 - Dobbeltklikk på målet **COGS** på listen **Data**.
 - Dobbeltklikk på operatoren + på listen **Operatorer**.
 - Dobbeltklikk på målet **EXPENSE** på listen **Data**.
 - I feltet **Uttrykk** merker du @Measure(MDSAMPLE.COGS)+@Measure(MDSAMPLE.EXPENSE) i uttrykket og dobbeltklikker på operatoren (..) på listen **Operatorer** for å sette den merkede delen av uttrykket i parentes.

I figur 7 på side 15 ser du Profit-uttrykket du kan opprette i vinduet Bygging av SQL-uttrykk.

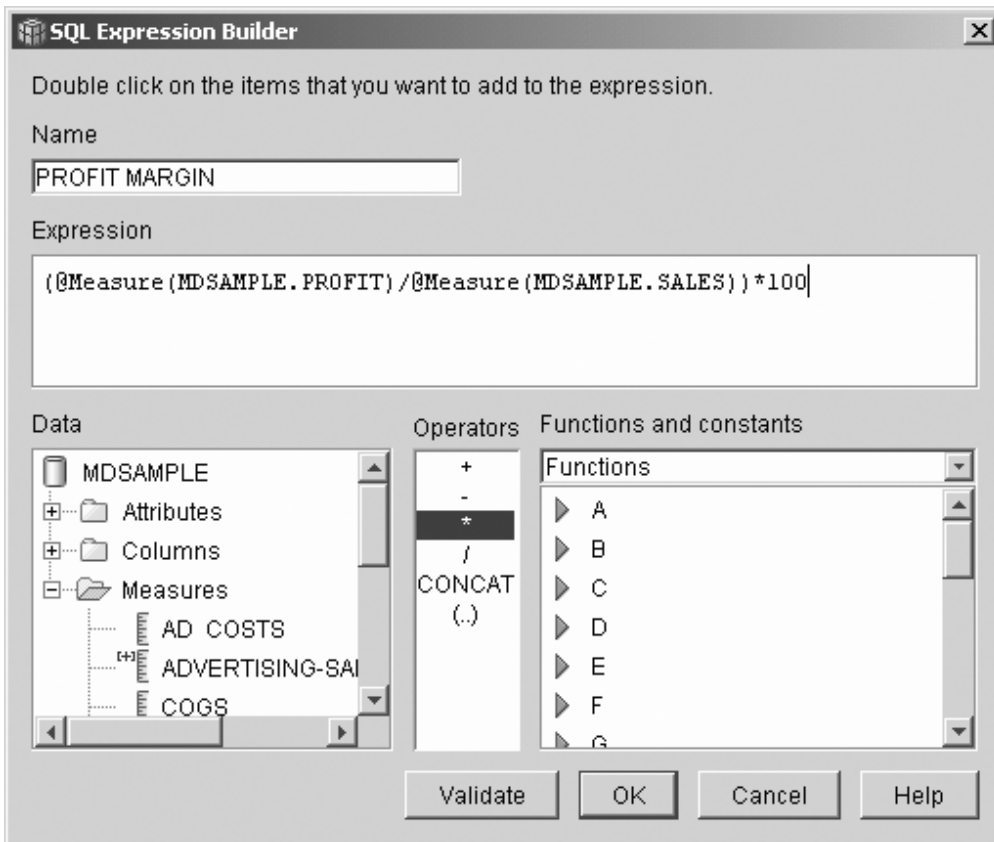


Figur 7. Fullført Profit-uttrykk i vinduet Bygging av SQL-uttrykk

- d. Klikk på **OK** for å opprette målet Profit og lukke vinduet Bygging av SQL-uttrykk.
 - e. Klikk på samlingen for målet Profit på siden Samlinger og velg funksjonen SUM. Målet Profit er opprettet.
3. Opprett målet Profit Margin:
 - a. Klikk på **Opprett beregnet mål** på siden Mål. Vinduet Bygging av SQL-uttrykk blir åpnet.
 - b. Skriv PROFIT MARGIN i feltet **Navn** i vinduet Bygging av SQL-uttrykk.
 - c. Utvid mappen **Mål** i listen **Data** og utfør disse trinnene for å opprette Profit Margin-uttrykket:
 - Dobbelklikk på målet **PROFIT** på listen **Data** for å tilføye det til uttrykket.
 - Dobbelklikk på operatoren **/** på listen **Operatorer**.
 - Dobbelklikk på målet **SALES** på listen **Data**.
 - Sett hele uttrykket i parentes ved å skrive inn parenteser i feltet **Uttrykk**.

- Plasser markøren på slutten av uttrykket og dobbeltklikk på operatoren * på listen **Operatører**.
- Skriv 100 på slutten av uttrykket i feltet **Uttrykk**.

I figur 8 ser du Profit Margin-uttrykket du kan opprette i vinduet Bygging av SQL-uttrykk.



Figur 8. Fullført Profit Margin-uttrykk i vinduet Bygging av SQL-uttrykk

- d. Klikk på **OK** for å opprette målet Profit Margin og lukke vinduet Bygging av SQL-uttrykk.
På siden Samlinger setter OLAP-senteret som standard samlingsfunksjonen til NONE for sammensatte mål, så du trenger ikke å endre samlingsfunksjonen.
4. Klikk på **OK** for å lukke vinduet Faktaegenskaper og lagre de to nye målene du har tilføyd til faktaobjektet.

Når databaseadministratoren har opprettet disse to målene, kan det utføres nyttige analyser ved hjelp av målene.

Kapittel 4. Telle antall Internett-bestillinger

En detaljist utvidet for noen få år siden forretningsdriften ved å starte salg på Internett. Nå ønsker firmaet å analysere virkningen av sitt Internett-salg. En av de første beregningene firmaet trenger, er hvor mange bestillinger som er levert over Internett.

Detaljer for scenariet

Firmaets database har en faktatabell for Internett-bestillinger med kolonnene ORDER_ID, PRODUCT_ID, QUANTITY og TIME_ID. Kolonnen PRODUCT_ID inneholder hvert enkelt produkt som er solgt i en bestilling, og kolonnen QUANTITY inneholder produktantallet som er solgt i bestillingen. Bestillinger med mer enn ett produkt har like mange rader som antallet av produkter som er solgt i bestillingen. I tabell 6 ser du et eksempel der bestilling 1 omfatter antallet 3 av produkt A, 1 av produkt O og 1 av produkt G.

Tabell 6. Innhold i faktatabell

ORDER_ID	PRODUCT_ID	QUANTITY
1	A	3
1	O	1
1	G	1
2	L	1
2	Q	2
3	P	5

Databaseadministratoren kan opprette målet Order Count som teller hver unike post i kolonnen ORDER_ID. Målet Order Count blir definert med nøkkelordet DISTINCT i SQL-uttrykket og funksjonen COUNT for samling på tvers av alle dimensjonene. Målets SQL-uttrykk oppretter en liste over distinkte bestillinger som telles under samlingen. Fordi målet ikke omfatter noen summering, blir det kalt et ikke-additivt mål.

Ikke-additive mål er også nyttige når du har tegndata eller andre data som du vil telle. Du kan for eksempel bruke ikke-additive mål for å telle antall postnumre du har sendt produkter til.

I dette eksempelet valgte databaseadministratoren å definere målet Order ID som er direkte tilordnet kolonnen ORDER_ID. Du kan imidlertid velge å

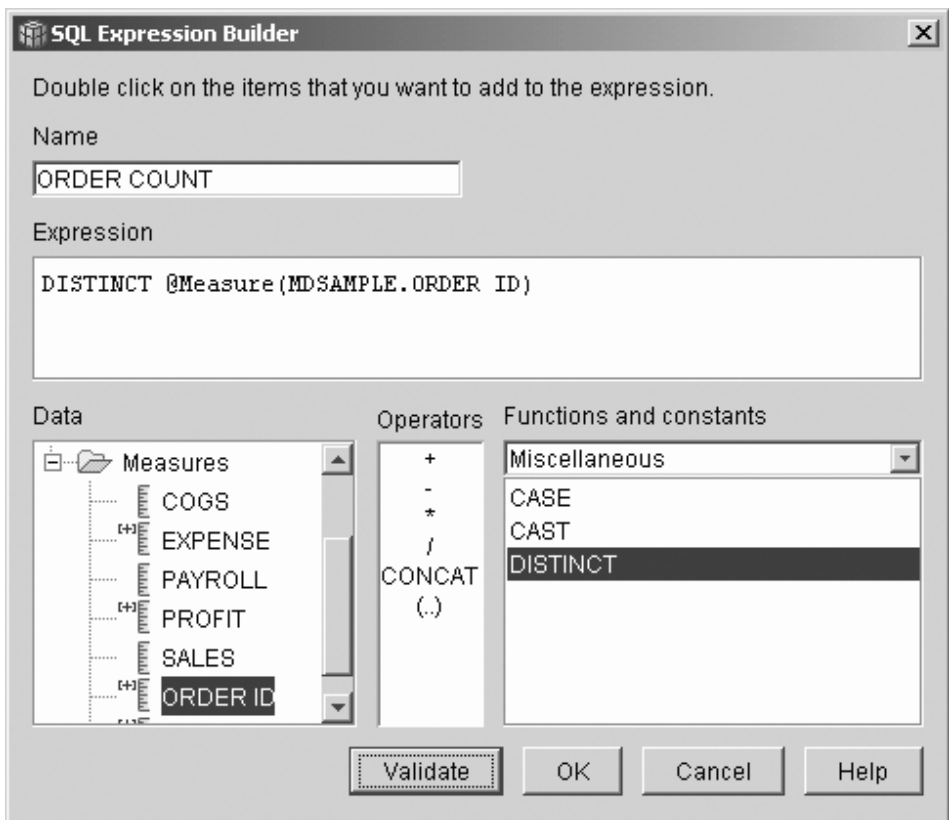
bruke kolonnen ORDER_ID på samme måte. Standardsamlingen er forskjellig basert på om du bruker en kolonne eller et mål i SQL-uttrykket, men i begge tilfeller må du endre standardsamlingen til funksjonen COUNT slik det er beskrevet i “Opprette målene”.

Opprette målene

Fremgangsmåten nedenfor beskriver hvordan du kan opprette målet Order Count i et eksisterende faktaobjekt i vinduet Faktaegenskaper i OLAP-senteret:

1. Åpne vinduet Faktaegenskaper ved å høyreklikke på faktaobjektet i OLAP-senter-objektoversikten og klikke på **Rediger mål**. Vinduet Faktaegenskaper blir åpnet.
2. Klikk på knappen **Opprett beregnet mål** på siden Mål. Vinduet Bygging av SQL-uttrykk blir åpnet.
3. Skriv ORDER COUNT i feltet **Navn** i vinduet Bygging av SQL-uttrykk.
4. Utvid mappen **Mål** i listen **Data** og utfør disse trinnene for å opprette Order Count-uttrykket:
 - Velg **Diverse** i feltet **Funksjoner og konstanter**. Dobbelklikk på nøkkelordet **DISTINCT** i listen over diverse funksjoner og konstanter.
 - Dobbelklikk på målet **ORDER ID** på listen **Data**.

I figur 9 på side 21 ser du Order Count-uttrykket du kan opprette i vinduet Bygging av SQL-uttrykk.



Figur 9. Fullført Order Count-uttrykk i vinduet Bygging av SQL-uttrykk

5. Klikk på **OK** for å lukke vinduet Bygging av SQL-uttrykk.
6. Klikk på samlingen for målet Order Count på siden Samlinger og velg funksjonen **COUNT**.
7. Klikk på **OK** for å lagre endringen i faktaobjektet og lukke vinduet Faktaegenskaper.

Du har nå et mål som teller antall distinkte Order ID-radverdier. Du kan bruke dette målet sammen med andre mål for å analysere dataene dine.

Kapittel 5. Rangere salgstall

En butikkjede som selger kontorrekvisita, har ekspandert raskt de siste årene. Ledelsen vurderer å legge ned enkelte av butikkene med dårligst resultat for å redusere kostnadene og øke fortjenesten. Salgshistorikken til en butikk er en viktig faktor ved vurderingen av om en butikk skal legges ned. Under analysen må det være mulig å rangere salgstallene og drille ned på tvers av dimensjonene.

Detaljer for scenariet

Butikkjedens database har en faktatabell med kolonnen Sales sammen med andre kolonner. Databasen har dessuten flere dimensjonstabeller. Databaseadministratoren kan opprette målet Sales Rank som bruker funksjonen RANK, en OLAP-funksjon som finnes i DB2 Universal Database (DB2 UDB).

DB2 Cube Views støtter følgende OLAP-funksjoner i DB2 UDB:

RANK

Ordner radene og tildeler en rangfølge til hver rad. Rangfølgen er definert som 1 pluss antall forutgående rader som er distinkte med hensyn til rekkefølge. Hvis den relative rekkefølgen til to eller flere rader ikke kan fastslås fordi de har like radverdier, får radene tildelt samme rangfølge. Rangeringsresultatene kan ha hull i rangfølgen hvis det finnes like radverdier. I tabell 7 på side 25 ser du et eksempel på rangeringsresultatene fra funksjonen RANK for et sett med radverdier.

Syntaksen for funksjonen RANK er

```
RANK ( ) OVER (ORDER BY sorteringsnøkkeluttrykk uttrykksrekkefølge)
```

der *sorteringsnøkkeluttrykk* er settet med data som skal rangeres, og *uttrykksrekkefølge* er et nøkkelord, **ASC** eller **DESC**, som ordner verdiene i *sorteringsnøkkeluttrykket* i stigende eller synkende rekkefølge. DB2 Cube Views krever at *sorteringsnøkkeluttrykket* er et eksisterende mål, ikke en kolonne eller et attributt. DB2 Cube Views støtter ikke PARTITION BY-leddet som leveres sammen med denne funksjonen i DB2 UDB. Du finner mer informasjon om funksjonen RANK i DB2 UDB Information Center.

DENSERANK

Ordner radene og tildeler en rangfølge til hver rad. Rangfølgen til en rad er definert som 1 pluss antall strengt forutgående rader. Derfor er

rangeringsresultatene sekvensielle og uten hull i rangordningen. I tabell 7 på side 25 ser du et eksempel på rangeringsresultatene fra funksjonen DENSERANK for et sett med radverdier.

Syntaksen for funksjonen DENSERANK er

```
DENSERANK ( ) OVER (ORDER BY sorteringsnøkkeluttrykk  
uttrykksrekkefølge)
```

der *sorteringsnøkkeluttrykk* er settet med data som skal rangeres, og *uttrykksrekkefølge* er et nøkkelord, **ASC** eller **DESC**, som ordner verdiene i sorteringsnøkkeluttrykket i stigende eller synkende rekkefølge. DB2 Cube Views krever at *sorteringsnøkkeluttrykket* er et eksisterende mål, ikke en kolonne eller et attributt. DB2 Cube Views støtter ikke PARTITION BY-leddet som leveres sammen med denne funksjonen i DB2 UDB. Du finner mer informasjon om funksjonen DENSERANK i DB2 UDB Information Center.

ROWNUMBER

Beregner sekvensielt radnummer for raden på grunnlag av rekkefølgen, og starter med 1 for den første raden. Hvis ORDER BY-leddet ikke er oppgitt, blir radnumrene tildelt til radene i tilfeldig rekkefølge.

Syntaksen for funksjonen ROWNUMBER er

```
ROWNUMBER ( ) OVER ([ORDER BY sorteringsnøkkeluttrykk  
uttrykksrekkefølge])
```

der *sorteringsnøkkeluttrykk* er settet med data som skal rangeres, og *uttrykksrekkefølge* er et nøkkelord, **ASC** eller **DESC**, som ordner verdiene i sorteringsnøkkeluttrykket i stigende eller synkende rekkefølge. DB2 Cube Views krever at et eksisterende mål, ikke en kolonne eller et attributt, blir brukt som datakilde for denne funksjonen. DB2 Cube Views støtter ikke PARTITION BY-leddet som leveres sammen med denne funksjonen i DB2 UDB. Du finner mer informasjon om funksjonen ROWNUMBER i DB2 UDB Information Center.

Disse OLAP-funksjonene er ikke med på listen Funksjoner og konstanter i vinduet Bygging av SQL-uttrykk.

Tabell 7. Rangeringsresultater for et sett med verdier når du bruker funksjonene RANK og DENSERANK

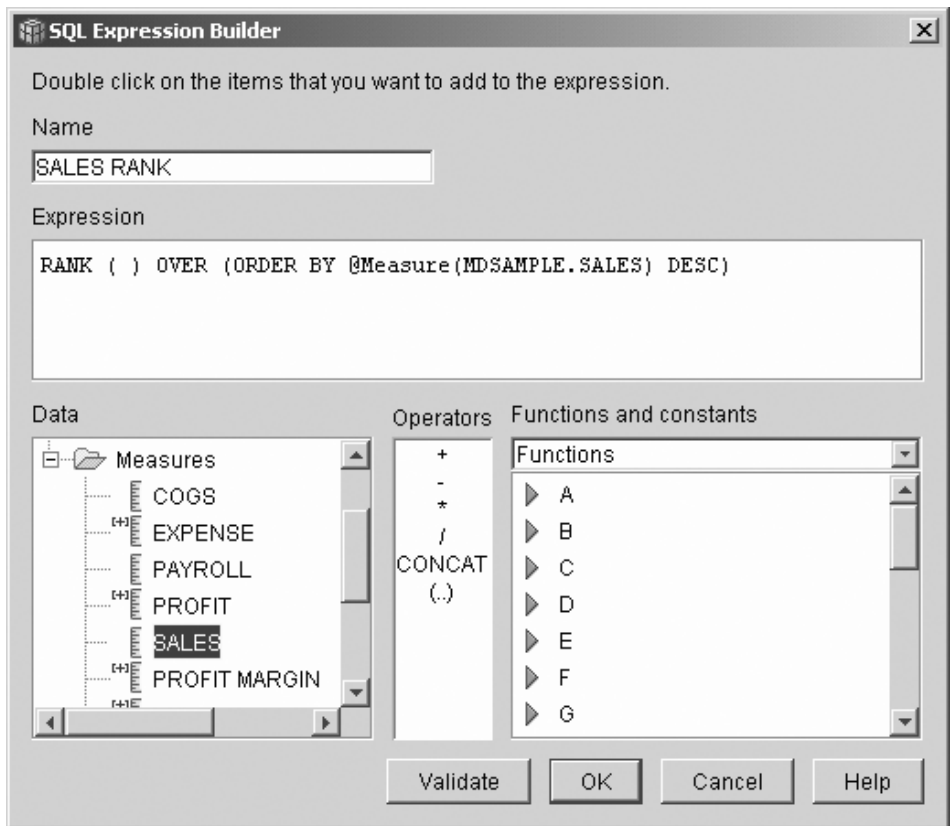
Radverdier	Rekkefølge	Rangeringsresultater fra funksjonen RANK	Rangeringsresultater fra funksjonen DENSERANK
100	1	1	1
35	2	2	2
23	3	3	3
8	4	4	4
8	4	4	5
6	5	6	6

Opprette målene

Fremgangsmåten nedenfor beskriver hvordan du kan opprette målet Sales Rank i et eksisterende faktaobjekt i vinduet Faktaegenskaper i OLAP-senteret:

1. Åpne vinduet Faktaegenskaper ved å høyreklikke på faktaobjektet i OLAP-senter-objektoversikten og klikke på **Rediger mål**. Vinduet Faktaegenskaper blir åpnet.
2. Klikk på **Opprett beregnet mål** på siden Mål for å opprette målet Sales Rank. Vinduet Bygging av SQL-uttrykk blir åpnet.
3. Skriv SALES RANK i feltet **Navn** i vinduet Bygging av SQL-uttrykk.
4. Opprett Sales Rank-uttrykket:
 - Skriv denne funksjonssyntaksen i feltet Uttrykk: RANK () OVER (ORDER BY measure DESC).
 - Utvid mappen **Mål** på listen **Data**.
 - Merk ordet measure i funksjonssyntaksen i feltet **Uttrykk** og dobbeltklikk på målet **SALES** for å tilføye målet SALES til uttrykket.

Det endelige uttrykket er vist i figur 10 på side 26.



Figur 10. Fullført Sales Rank-uttrykk i vinduet Bygging av SQL-uttrykk

5. Klikk på **Valider** for å kontrollere at uttrykket er gyldig. Klikk på **OK** for å lukke vinduet Bygging av SQL-uttrykk.

Du trenger ikke å endre standard samlingsfunksjon, Ingen, på siden Samlinger. Alternativet Ingen er standard for målet Sales Rank fordi datakilden er numerisk og bare refererer til eksisterende mål.

Ved å bruke funksjonen RANK i målets uttrykk for å ordne kolonnen Sales i synkende rekkefølge, kan den som utfører analysen for eksempel drille ned på tvers av andre dimensjoner for å finne ut hvilken butikk som har de dårligste salgsresultatene det siste året for en bestemt produktlinje eller med hensyn til andre dimensjonsdata som er lagret i databasen.

Kapittel 6. Bruke tidsdata lagret i faktatabellen til å opprette en tidsdimensjon

En detaljist, XYZ Detalj, modellerer sine salgstransaksjonsdata i DB2 Cube Views slik at dataene kan analyseres mer effektivt. Siden dette er transaksjonsdata, er den eneste tilgjengelige tidsinformasjonen en dato som er knyttet til hver enkelt transaksjon. Det er imidlertid nødvendig med tidsinformasjon som er modellert i en tidsdimensjon, hvis det skal være mulig å analysere salgstrender per kvartal eller beregne gjennomsnittlig lagerbeholdningsverdi for hver enkelt uke.

Mange databaseadministratorer prøver å unngå å lagre tidsdata som en dato eller som systemtid, for hvis det ikke finnes noen transaksjoner en dag, blir det hull i dataene, noe som kan gjøre det vanskelig å samle og vise dataene på riktig måte. Vanligvis er det bedre å modellere tidsdataene i en tidstabell. Databaseadministratoren hos XYZ Detalj er imidlertid sikker på at det blir minst en transaksjon hver dag, og velger å beholde den gjeldende strukturen til dataene.

Detaljer for scenariet

XYZ Detalj har en faktatabell med målbare data for hver transaksjon, som omfatter salg, kostnader, mengde solgt og dato. Databasen inneholder dessuten en dimensjonstabell for region og en for produkt. Problemet er at tidsdataene er tatt med i faktatabellen og ikke lagret i en egen dimensjonstabell. Databaseadministratoren må opprette et dimensjonsobjekt basert på datodataene i faktaobjektet.

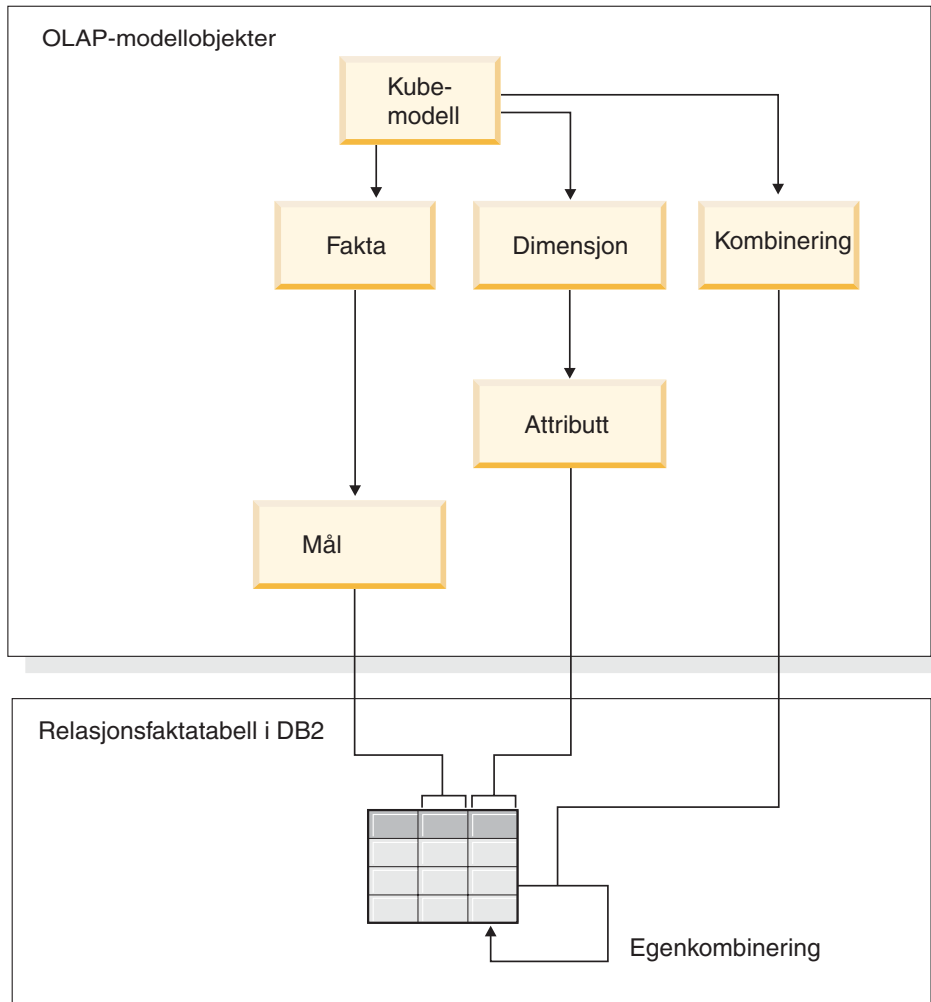
Når administratoren oppretter en tidsdimensjon på grunnlag av en enkelt kolonne med datodata i faktatabellen, stilles det to krav:

- Fordi alle dimensjonsobjektene i en gyldig kubemodell må være kombinert med faktaobjektet og tidsdimensjonsobjektet og faktaobjektet er basert på den samme faktatabellen, må tidsdimensjonsobjektet kombineres med faktaobjektet ved hjelp av en egenkombinering som kombinerer faktatabellen med seg selv.
- Databaseadministratoren må bygge beregnede attributter som samler datodataene i meningsfulle nivåer som uke, måned, kvartal og år.

En egenkombinering er en type kombinering som kombinerer en tabell med seg selv, og i dette tilfellet gjelder det faktatabellen. Egenkombineringen må kombinere en eller flere kolonner som sammen kan identifisere entydig en hvilken som helst rad i faktatabellen. Primærnøkkelen er det beste valget.

Hvis det ikke er definert noen primærnøkkel, er settet med kolonner som blir brukt til å kombinere faktatabellen med dimensjonstabellene, en god kandidat for primærnøkkelen. Når du skal optimalisere kubemodellen, må det være definert en primærnøkkel. Kombinerings kardinalitet må være 1:1, og kombineringsstypen må være intern.

I figur 11 ser du hvordan et faktaobjekt, en dimensjon basert på faktatabellen og en kombinerings av fakta og dimensjon kan være tilordnet samme faktatabell.



Figur 11. Hvordan en egenkombinering kombinerer tabellen med seg selv

Opprette attributtene og dimensjonen

Fremgangsmåten nedenfor beskriver hvordan du kan opprette dimensjonen Time og beregnede attributter på grunnlag av faktatabellen i dimensjonsveiviseren i OLAP-senteret:

1. Åpne dimensjonsveiviseren ved å høyreklikke på kubemodellen i OLAP-senter-objektoversikten og velge **Opprett dimensjon**. Dimensjonsveiviseren blir åpnet.
2. Skriv Time i feltet **Navn** på siden Navn. Du kan eventuelt endre forretningsnavnet og skrive inn en kommentar. Klikk på **Neste**.
3. Velg kubemodellens faktatabell. Klikk på **Neste**. Du trenger ikke å oppgi dimensjonskombineringer fordi du bare har en tabell i dimensjonen. Klikk på **Neste** på siden Dimensjonskombineringer.
- 4.
5. Velg kolonnen **Systemtid** på siden Dimensjonsattributter.
6. Valgfritt: Opprett de beregnede attributtene som samler systemtidsdataene i større enheter som måned, kvartal og år. Når du skal opprette de beregnede attributtene, klikker du på knappen **Opprett beregnet attributt** for å åpne Bygging av SQL-uttrykk og definere uttrykket for hvert attributt som beregner måneder, kvartal og år på grunnlag av systemtidskolonnen. Når du har definert hvert av de beregnede attributtene, klikker du på **Valider** for å kontrollere at uttrykket er gyldig, og deretter på **OK** for å lukke Bygging av SQL-uttrykk og gå tilbake til dimensjonsveiviseren. Klikk på **Neste** når du har opprettet alle attributtene du ønsker.
7. Velg **Tid** på siden Dimensjonstype. Klikk på **Neste**.
8. Klikk på **Opprett kombineringsveiviseren** på siden Kombineringsveiviseren blir åpnet, der du oppretter egenkombineringen. Oppgi et navn og klikk på **Neste**. Velg kolonnen eller settet med kolonner som entydig definerer en rad i faktatabellen, for eksempel primærnøkkelen, for både venstre- og høyreattributtet. Velg ett par om gangen og klikk på **Tilføy** for å tilføye attributtparet til kombineringsveiviseren. Velg kombineringsstypen indre kombineringsveiviser og kardinaliteten 1:1. Klikk på **Fullfør** når du har tilføyd alle de nødvendige attributtparene. Kombineringsveiviseren blir lukket.
9. Klikk på **Fullfør** på siden Kombineringsveiviseren og dimensjon.

Når tidsdimensjonen er definert i kubemodellen, har XYZ Detalj fått nye muligheter i sin dataanalyse. De kan nå utføre tidsrelaterte analyser som omfatter varelageret.

Merknader

Denne boken er utarbeidet for produkter og tjenester som er tilgjengelige i USA.

Henvisninger i boken til IBMs produkter, programmer eller tjenester betyr ikke at IBM har til hensikt å gjøre dem tilgjengelige i alle land der IBM driver virksomhet. Kontakt din lokale IBM-representant for å få informasjon om hvilke produkter og tjenester som er tilgjengelige i Norge. Henvisninger til IBMs produkter, programmer eller tjenester betyr heller ikke at det bare er de som kan benyttes. Andre produkter, programmer eller tjenester som har tilsvarende funksjoner, kan brukes i stedet, forutsatt at de ikke gjør inngrep i noen av IBMs patent- eller opphavsrettigheter eller andre lovbeskyttede rettigheter. Vurdering og verifisering ved bruk sammen med andre produkter, programmer eller tjenester enn de som uttrykkelig er angitt av IBM, er brukerens ansvar.

IBM kan ha patent på eller patentsøknader til behandling for de produktene som er omtalt i denne publikasjonen. At du har mottatt denne publikasjonen, innebærer ikke at du får lisensrettighet til disse produktene. Du kan sende spørsmål angående lisenser til

Director of Commercial Relations - Europe
IBM Deutschland GmbH
Schönaicher Str. 220
D - 7030 Böblingen
Tyskland

Lisensforespørsler om dobbeltbyteinformasjon (DBCS) kan rettes til IBMs advokat eller til:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION LEVERER
DENNE BOKEN SOM DEN ER ("AS IS") UTEN FORPLIKTELSER AV NOE
SLAG.

Denne boken kan inneholde tekniske unøyaktigheter eller typografiske feil. Opplysninger i denne boken kan bli endret. Slike endringer blir tatt med i nye utgaver av boken. IBM kan uten varsel endre produktene og/eller programmene som er beskrevet i denne boken.

IBM kan bruke eller distribuere informasjonen du gir til IBM på den måten IBM mener er best, uten forpliktelser i noen retning.

Hvis du som lisensinnehaver av dette programmet ønsker informasjon om programmet for å kunne (i) utveksle informasjon mellom selvstendig utviklede programmer og andre programmer (inkludert dette) og (ii) dra gjensidig nytte av informasjonen som er utvekslet, kan du kontakte:

IBM Norge
Software Marketing
Postboks 500
1411 Kolbotn

Slik informasjon kan være tilgjengelig under gjeldende betingelser, eventuelt mot betaling.

Det lisensierte programmet som er beskrevet i denne boken, og alt lisensiert materiale som er tilgjengelig for programmet, leveres av IBM i henhold til IBMs generelle betingelser, IBMs internasjonale bruksbetingelser eller en tilsvarende avtale mellom partene.

Informasjon om andre produkter enn IBMs egne er hentet fra leverandørene av produktene, fra deres annonseringer eller fra andre tilgjengelige kilder. IBM har ikke testet disse produktene, og kan ikke bekrefte påstander om ytelse, kompatibilitet eller andre egenskaper ved dem. Spørsmål om funksjonene i ikke-IBM-produkter må rettes til leverandøren av produktet.

Denne boken kan inneholde eksempler på data og rapporter som brukes i den daglige driften av et firma. For å illustrere eksemplene så godt som mulig, blir det brukt navn på personer, firmaer og produkter. Alle disse navnene er fiktive, og enhver likhet med virkelige navn er tilfeldig.

Alle ytelsesdataene du finner i dette dokumentet, ble hentet i et kontrollert miljø. Resultatene du kan oppnå i andre operativmiljøer, kan variere betraktelig. Noen av målingene er foretatt på systemer som er under utvikling, og det er ikke sikkert at du oppnår samme resultat på alminnelige tilgjengelige systemer. Noen av målingene kan dessuten ha blitt beregnet ved hjelp av ekstrapolasjon. De faktiske resultatene kan variere. Brukerne av dette dokumentet bør bekrefte dataene som brukes i sitt bestemte miljø.

Informasjon om andre produkter enn IBMs egne er hentet fra leverandørene av produktene, fra deres annonseringer eller fra andre tilgjengelige kilder. IBM har ikke testet disse produktene, og kan ikke bekrefte påstander om ytelse, kompatibilitet eller andre egenskaper ved dem. Spørsmål om funksjonene i ikke-IBM-produkter må rettes til leverandøren av produktet.

Hvis du ser på den maskinleselige versjonen av denne boken, kan det hende at fotografier og fargeillustrasjoner ikke blir vist.

Varemerker

Navnene nedenfor er varemerker for International Business Machines i USA og/eller andre land.

DB2
DB2 Universal Database
IBM
Office Connect
Redbooks

Navnene nedenfor er varemerker eller registrerte varemerker for andre selskaper.

Microsoft, Windows, Windows NT, Windows 2000, Windows XP og Microsoft Excel er varemerker eller registrerte varemerker for Microsoft Corporation.

Java og alle Java-baserte varemerker og logoer, samt Solaris er varemerker for Sun Microsystems, Inc. i USA og/eller andre land.

UNIX er et registrert varemerke i USA og/eller andre land, og lisensieres kun gjennom X/Open Company Limited.

Linux er et registrert varemerke for Linus Torvalds. Red Hat og alle Red Hat-baserte varemerker og logoer er varemerker eller registrerte varemerker for Red Hat, Inc. i USA og/eller andre land.

Andre navn på selskaper, produkter eller tjenester kan være varemerker for andre selskaper.