Aggregering

EVERYLOOP

www.everyloop.com

Aggregerad data

Data som är en sammanslagning av flera datapunkter, till exempel ett medeltal eller en totalsumma, kallas aggregerad data. Den vanligaste anledningen att man vill skapa ett sådant aggregat är för att öka överskådligheten och/eller visa statistik.

Antag att vi har en tabell med rådata som innehåller ett klockslag för varje gång en bil passerat över en bro. Om vi vill få en överblick över datat kan vi välja att aggregera antal bilar per timma, eller kanske per dag. Om det även finns uppgifter om vilket land varje bil är från i rådatat så kan vi välja att aggregera per land.

Aggregeringsfunktioner i SQL

En aggregeringsfunktion tar en lista med värden, gör en beräkning på dessa, och returnerar ett värde (skalär).

I SQL finns det ett antal funktioner som används för aggregering. Den kanske mest grundläggande är **Count** som helt enkelt räknar antalet datapunkter.

Andra vanliga aggregeringsfunktioner:

Sum()	Avg()	Stdev()
Min()	Max()	String_agg()

Count()

Count tar ett kolumnnamn som parameter och räknar alla värden i kolumnen som inte är NULL. Om man vill räkna samtliga rader, även de som är NULL, så skriver man Count(*)

```
Select count(stad) from städer where land = ’Sverige’;
```

Man kan även ange distinct i Count() för att bara räkna unika värden:

```
Select count(distinct land) from städer;
```

Gruppering av data

I aggregeringsexemplen ovan så får vi bara ut ett enda värde; antalet städer i sverige, samt i andra exemplet, antal unika länder i tabellen med städer.

Oftast vill vi dock **gruppera** data och få ut aggregatet för varje grupp i tabellform. Kanske vill vi inte bara veta antalet svenska städer som tabellen innehåller, utan vi vill veta antalet städer för varje land.

Då behöver vi gruppera vårat data per land och sedan räkna antalet städer i varje grupp. Låt oss se hur vi gör detta i SQL!

Group by

I slutet av vår select-sats kan vi lägga till **group by** följt av en eller flera kolumner som vi vill gruppera på.

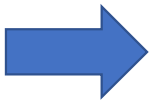
Det är endast de kolumner som vi grupperat på som vi kan ta ut direkt. Övriga måste vara i form av aggregat. Alltså, om vi grupperar på land så kan vi direkt ta ut landet som en kolumn eftersom det blir just en grupp per land, men vill vi ta ut antal städer (count) eller en summering av invånare i städerna (sum) så måste vi ange en aggregeringsfunktion.

```
Select land, count(stad) from städer group by land;
```

Rådata ➡ Aggregerad data

Exempel på hur rådata över städer ser ut i aggregerad form om man grupperar på land, räknar städer, och summerar invånare:

Land	Stad	Invånare
Sverige	Stockholm	932917
Sverige	Göteborg	549789
Norge	Oslo	658390
Norge	Bergen	278556
Sverige	Malmö	341457
Danmark	Köpenhamn	601448



Land	Städer	Invånare
Sverige	3	1 824 163
Norge	2	936 946
Danmark	1	601448

Having

Precis som att man ibland vill ge vilkor för vilka rader man vill få ut, så vill man ibland ge vilkor på vilka grupper man vill visa i resultatet. Detta gör man med **having.** Kanske vill vi gruppera på land, men bara ta ut de grupper som har fler än 10 städer:

```
Select land from städer group by land having count(stad) > 10;
```

Vi kan inte använda ”where” för grupper eftersom det redan används med en annan betydelse. Tänk om vi t.ex vill gruppera på land men bara ta med städer med fler än 100000 invånare (where) och sedan visa de grupper som har mer än 10 sådana städer (having).