



# **Snowflake – Aggregierte Abfragen 1**

Stephan Karrer

## Wichtige Gruppenfunktionen

Funktion	Beschreibung
AVG	Durchschnittswert, ignoriert NULL Werte
COUNT	Anzahl der ausgewählten Zeilen
MAX	Gibt den höchsten Wert einer Auswahl wieder
MIN	Gibt den niedrigsten Wert einer Auswahl wieder
STDDEV	Standardabweichung einer Gruppe von Werten, wobei NULL – Werte ignoriert werden
SUM	Summenwerte einer Gruppe von Werten, wobei NULL – Werte ignoriert werden
VARIANCE	Abweichung einer Gruppe von Werten, wobei NULL – Werte ignoriert werden

## Einfache Verwendung von Gruppenfunktionen

```
SELECT COUNT(*) "Anzahl Zeilen" FROM employees;  
  
SELECT AVG( COALESCE(salary, 0)) FROM employees;  
  
SELECT MAX(salary) FROM employees  
       WHERE job_id = 'IT_PROG' ;
```

- COUNT(\*) ist ein Spezialfall, bei dem einfach alle Zeilen gezählt werden.

## Verwendung von Gruppenfunktionen

```
SELECT [ALL|DISTINCT] Auswahlliste
      FROM Quelle
      [WHERE Where-Klausel]
      [GROUP BY (Group-by-Attribut)
              [HAVING Having-Klausel]]
      [ORDER BY (Sortierungsattribut) [ASC|DESC]]
```

- WHERE:  
schränkt die Ausgangsmenge ein.
- GROUP BY:  
zerlegt die Ausgangsmenge in Gruppen. Je Gruppe wird aggregiert.
- HAVING:  
nachträgliche Filterung der Ergebnisse.
- SELECT:  
in der Auswahlliste können nur die Gruppierungsattribute und Aggregate verwendet werden !!

## Verwendung von Gruppenfunktionen: GROUP BY und HAVING

```
SELECT department_id, COUNT(DISTINCT job_id)
      FROM employees
      GROUP BY department_id;
```

```
SELECT job_id, SUM(salary)
      FROM employees
      WHERE department_id <> 100
      GROUP BY job_id
      ORDER BY job_id DESC;
```

```
SELECT job_id, SUM(salary)
      FROM employees
      WHERE department_id <> 100
      GROUP BY job_id
      HAVING SUM(salary) > 20000
      ORDER BY job_id DESC;
```

- Einige Aggregatsfunktionen (AVG, COUNT, STDDEV, SUM, VARIANCE, ...) erlauben DISTINCT, wodurch keine Duplikate berücksichtigt werden.

## Nach mehreren Spalten gruppieren

```
SELECT department_id dept_id,  
       job_id,  
       SUM(salary)  
FROM employees  
GROUP BY department_id, job_id;
```

- Es wird anhand der tiefsten Gruppenbildung aggregiert.

## Gruppierung nach berechneten Werten

```
SELECT length(last_name) as "Length", count(*) as "Count"  
  FROM employees  
 GROUP BY length(last_name)    -- "Length"  
 ORDER BY "Length";
```

- Generell kann nach Ausdrücken und damit nach berechneten Werten aggregiert werden.
- In der ORDER BY– Klausel darf der Spaltenalias verwendet werden (ANSI), bei Snowflake auch in der GROUP BY- bzw. HAVING-Klausel !