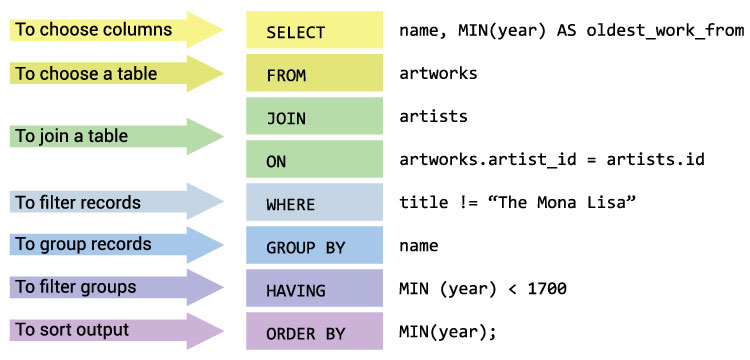
**Modul 7: SQL für Fortgeschrittene, Tag 1**

**Themen für Heute:**

* **GROUP BY**: Die GROUP BY-Klausel gruppiert Datensätze in einer Ergebnismenge nach identischen Werten in einer oder mehreren Spalten.
* **HAVING**: Die HAVING-Klausel wird verwendet, um die von der GROUP BY-Klausel bereitgestellten Ergebnismengengruppen weiter zu filtern.
* **Aggregate** **functions**: Aggregatfunktionen führen eine Berechnung mit einer Reihe von Werten durch und geben einen einzelnen Wert zurück.

**Material:**

Sehen Sie sich die Videos 26, 33-47, 51, 52 der folgenden Playlist an:

[*https://www.youtube.com/playlist?list=PL\_RGaFnxSHWr\_6xTfF2FrIw-NAOo3iWMy*](https://www.youtube.com/playlist?list=PL_RGaFnxSHWr_6xTfF2FrIw-NAOo3iWMy)

Gehen Sie das folgende Schulungsmaterial durch – „SQL Group By“ – „SQL Operators“:

[*https://www.w3schools.com/sql/sql\_groupby.asp*](https://www.w3schools.com/sql/sql_groupby.asp)

**Fragen:**

1. **Was ist der Unterschied zwischen WHERE und HAVING?**
   * WHERE wird zum Filtern von Datensätzen verwendet, bevor eine Gruppierung oder Aggregation erfolgt.
   * HAVING wird zum Filtern von Datensätzen nach einer Gruppierung oder einer Aggregation verwendet
   * WHERE is used to filter records before any grouping or aggregation occurs
   * HAVING is used to filter records by a grouping or an aggregation
2. **Was verstehst du unter einer Zeichenmanipulationsfunktion?**
   * Zeichenmanipulative Funktionen (CONCAT, LENGTH, SUBSTR, INSTR, LPAD, RPAD, TRIM und REPLACE) LOWER : Diese Funktion wandelt Alphazeichenwerte in Kleinbuchstaben um. LOWER gibt tatsächlich eine Zeichenfolge mit fester Länge zurück, wenn die eingehende Zeichenfolge eine feste Länge hat.
   * Character-Manipulative Functions (CONCAT, LENGTH, SUBSTR, INSTR, LPAD, RPAD, TRIM and REPLACE) LOWER : This function converts alpha character values to lowercase. LOWER will actually return a fixed-length string if the incoming string is fixed-length**.**
3. **Wie können Sie gemeinsame Datensätze aus zwei Tabellen abrufen?**
   * SQL-JOIN
4. **Was ist der Unterschied zwischen COUNT(\*), COUNT(Ausdruck) und COUNT(DISTINCT-Ausdruck)?**
   * COUNT(\*): zählt die gesamten Zeilen in der Tabelle, einschließlich der Werte von NULL.
   * COUNT(Ausdruck): zählt alle Zeilen in der angegebenen Spalte und schließt die Werte von NULL aus.
   * COUNT(DISTINCT-Ausdruck): zählt nur eindeutige (einzigartige) Zeilen in der definierten Spalte. Mit dem DISTINCT–Befehl werden überflüssige Informationen, die in einer Tabelle auftreten können, beseitigt und die Werte werden jeweils nur einmal angezeigt.
   * COUNT(\*): counts all rows in the table, including NULL values.
   * COUNT(expression): counts all rows in the given column and excludes the values ​​of NULL.
   * COUNT(DISTINCT expression): counts only distinct (unique) rows in the defined column. With the DISTINCT command, superfluous information that may appear in a table is eliminated and the values ​​are only displayed once at a time.
5. **Was wird dieser Code zurückgeben?**

SELECT Abteilung, Monat,

AVG(Gehalt) AS Durchschnitsgehalt

FROM Gehälter

GROUP BY Abteilung;

* + Das Durchschnittsgehalt der Gruppe Abteilungen aus der Tabelle Gehälter
  + The average salary of the group departments from the Salaries table

1. **Warum müssen wir Spalten haben, die nicht Teil der GROUP BY-Klausel als Teil der Aggregatfunktion sind?**

**Aufgaben:**

**Bestellungen** (**Orders**):

* + Bestellnr. (**ord\_no**),
  + Kaufbetrag (**purch\_amt**),
  + Bestell Datum (**ord\_date**),
  + KundenID (**customer\_id**) ,
  + VerkäuferID (**salesman\_id**)

**Kunden** (**customer**):

* + KundenNr. (**customer\_id**),
  + Kundenname (**cust\_name**),
  + Stadt (**City**),
  + Note (**grade**),
  + VerkäuferID (**salesman\_id**)

**Verkäufer** (**salesman**):

* + VerkäuferID (**salesman\_id**),
  + Name (**name**),
  + Stadt (**City**),
  + Kommission (**commission**)

1. **Schreiben Sie in der Tabelle „Bestellungen“ eine SQL-Abfrage, um den Gesamtkaufbetrag aller Bestellungen zu berechnen. Geben Sie den gesamten Kaufbetrag zurück.**
   * SELECT sum(purch\_amt) FROM orders;
2. **Schreiben Sie aus der Tabelle „Bestellungen“ eine SQL-Abfrage, um den durchschnittlichen Kaufbetrag aller Bestellungen zu berechnen. Geben Sie den durchschnittlichen Kaufbetrag zurück.**
   * SELECT AVG(purch\_amt) AS average\_purch\_amt FROM orders;
3. **Schreiben Sie aus der Tabelle „Bestellungen“ eine SQL-Abfrage, die die Anzahl der eindeutigen Verkäufer zählt. Geben Sie die Anzahl der Verkäufer zurück**
   * SELECT COUNT(salesman\_id) as seller\_count FROM orders;
4. **Schreiben Sie aus der Tabelle „Kunde“ eine SQL-Abfrage, um die Anzahl der Kunden zu zählen. Anzahl der Kunden zurückgeben.**
   * SELECT COUNT(customer\_id) as customer\_count FROM customer;
5. **Schreiben Sie aus der Tabelle „Kunde“ eine SQL-Abfrage, um die Anzahl der Kunden zu ermitteln, die mindestens eine Note für ihre Aktivität erhalten haben**
   * SELECT COUNT(customer\_id) as customer\_with\_grade\_count FROM customer

WHERE grade is not null;

1. **Schreiben Sie in der Tabelle „Bestellungen“ eine SQL-Abfrage, um den maximalen und minimalen Kaufbetrag zu ermitteln.**
   * SELECT min(purch\_amt), max(purch\_amt) FROM orders;
2. **Schreiben Sie aus der Tabelle „Kunde“ eine SQL-Abfrage, um die höchste Bewertung der Kunden in jeder Stadt zu finden. Zurück Stadt, maximale Note**
   * SELECT city, max(grade) FROM customer

group by city;

1. **Schreiben Sie aus der Tabelle „Bestellungen“ eine SQL-Abfrage, um den höchsten Kaufbetrag zu ermitteln, der von jedem Kunden bestellt wurde. Kunden-ID zurückgeben, maximaler Einkaufsbetrag.**
   * SELECT customer\_id, max(purch\_amt) FROM orders

group by customer\_id;

1. **Schreiben Sie aus der Tabelle „Bestellungen“ eine SQL-Abfrage, um den höchsten Kaufbetrag zu finden, der von jedem Kunden an einem bestimmten Datum bestellt wurde. Retoure, Bestelldatum und höchster Einkaufsbetrag.**
   * SELECT customer\_id, ord\_date, MAX(purch\_amt) AS highest\_purch\_amt FROM orders

GROUP BY (customer\_id) ORDER BY customer\_ID;

1. **Schreiben Sie aus der Tabelle „Bestellungen“ eine SQL-Abfrage, um den höchsten Kaufbetrag zu ermitteln, der von jedem Verkäufer am „2012-08-17“ getätigt wurde. Retourenverkäufer-ID, Kaufbetrag**
   * SELECT customer\_id, max(purch\_amt) FROM orders

WHERE ord\_date LIKE '%2012-08-17%';

1. **Schreiben Sie aus der Tabelle „Bestellungen“ eine SQL-Abfrage, um den höchsten Bestellbetrag (Kauf) jedes Kunden an einem bestimmten Bestelldatum zu ermitteln. Filtern Sie das Ergebnis nach dem höchsten Bestellbetrag (Kauf) über 2000,00. Kunden-ID, Bestelldatum und maximaler Kaufbetrag zurücksenden.**
   * SELECT customer\_id, ord\_date, max(purch\_amt) FROM orders

WHERE purch\_amt > 2000.00 AND ord\_date=2012-08-17;

1. **Schreiben Sie aus der Tabelle „Bestellungen“ eine SQL-Abfrage, um den maximalen Bestell- (Kauf-) Betrag im Bereich von 2000–6000 (Anfangs- und Endwerte sind enthalten) durch Kombination von jedem Kunden und Bestelldatum zu finden. Kunden-ID, Bestelldatum und maximaler Kaufbetrag zurücksenden.**
   * SELECT customer\_id, ord\_date, max(purch\_amt) FROM orders

WHERE purch\_amt BETWEEN 2000 and 6000;

1. **Schreiben Sie aus der Tabelle „Verkäufer“ eine SQL-Abfrage, um die Anzahl der Verkäufer in einer Stadt zu zählen. Geben Sie die Anzahl der Verkäufer zurück.**
   * SELECT COUNT(salesman\_id) FROM Salesman

group by City;