

DQL \Rightarrow SELECT \Rightarrow Datenbankabfrage



Die wichtigste Anweisung beim Arbeiten mit einer Datenbank ist das gezielte Abfragen der gespeicherten Daten. In SQL wird dazu die SELECT Anweisung verwendet. Die SELECT Anweisung ähnelt einer Frage bzw. Aufforderung an das Datenbanksystem, die gewünschten Daten zu liefern.



SELECT [**DISTINCT** | **DISTINCTROW**] * | **Datenfelder** **FROM** **Tabellenname**
[WHERE **Suchbedingung]**
[GROUP BY **GruppierungsDatenfelder]** **[HAVING** **Bedingung]**
[ORDER BY **SortierDatenfelder** **[ASC | DESC], ...]**
[LIMIT **Start, Anzahl];**



- SELECT** \Rightarrow Abfrage $\langle * \rangle$ alle Dtenfelder, einzelne oder mehrere mit Komma getrennt
- DISTINCT** \Rightarrow Mehrfach auftretende identische Werte werden nur einmal angezeigt
- DISTINCTROW** \Rightarrow Mehrfach auftretende identische Datensätze werden nur einmal angezeigt
- FROM** \Rightarrow Angabe der Tabelle bzw Tabellen auf die zugegriffen wird
- WHERE** \Rightarrow Such- bzw. Filterbedingng, die erfüllt werden muss



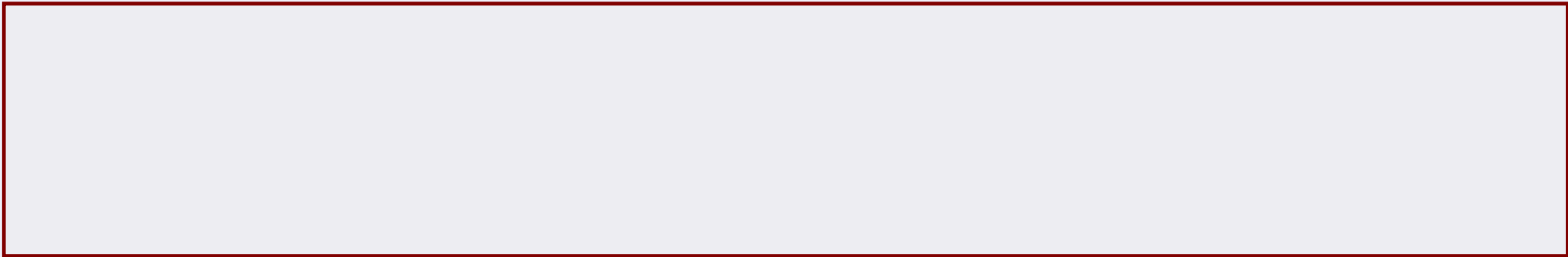
- GROUP BY** \Rightarrow Gruppierung nach Datenfeld bzw. Datenfelder
- HAVING** \Rightarrow Bedingung für die Gruppierung von Datenfeldern auf Basis der Aggregatfunktionen (AVG, COUNT, MAX, MIN, SUM)
- ORDER BY** \Rightarrow Sortierung nach Datenfeld bzw Datenfeldern, ab dem ersten Feld beginnend
- ASC** \Rightarrow Sortierung in aufsteigender Reihenfolge
- DESC** \Rightarrow Sortierung in absteigender Reihenfolge
- LIMIT** \Rightarrow Ausgabe einer bestimmten Menge von Datensätzen laut <Anzahl> und optional ab einem bestimmten Datensatz <Start>

SELECT ⇒ Beispiel „T_Mitarbeiter“



Anrede	Name	Vorname	Strasse	HausNr	Abteilung
Herr	Müller	Max	Schillerstraße	1	Einkauf
Herr	Kaiser	Karl	Schillerstraße	3a	Verkauf
Frau	Moser	Mechthild	Am Dom	4	Entwicklung
Frau	Peterson	Paula	Am Petershof	5	Management
Herr	Betholdt	Bert	Bauhofstraße	6	Einkauf
Frau	Blumig	Berta	Rosenstraße	7	Einkauf
Herr	Hauser	Hans	Zur alten Villa	8	Einkauf
Frau	Eisenreich	Edeltraud	Am Schmelzkübel	9	Verkauf
Frau	Kaiser	Karoline	Am Waldesrain	10	Verkauf
Herr	Bunt	Fritz	Farbengasse	4c	Verkauf
Herr	Ehrlich	Eddi	Am Markt	3	Entwicklung
Herr	Trinkus	Hans	Am Hofbräuhaus	5b	Entwicklung
Frau	Fischer	Willie	Zur letzten Mole	5xy	Entwicklung
Herr	Wichtig	Willibald	McDonalds-Allee	1xx	Management

SELECT ⇒ Beispiel „T_Mitarbeiter“



SELECT ⇒ Beispiel „T_Mitarbeiter“



t)



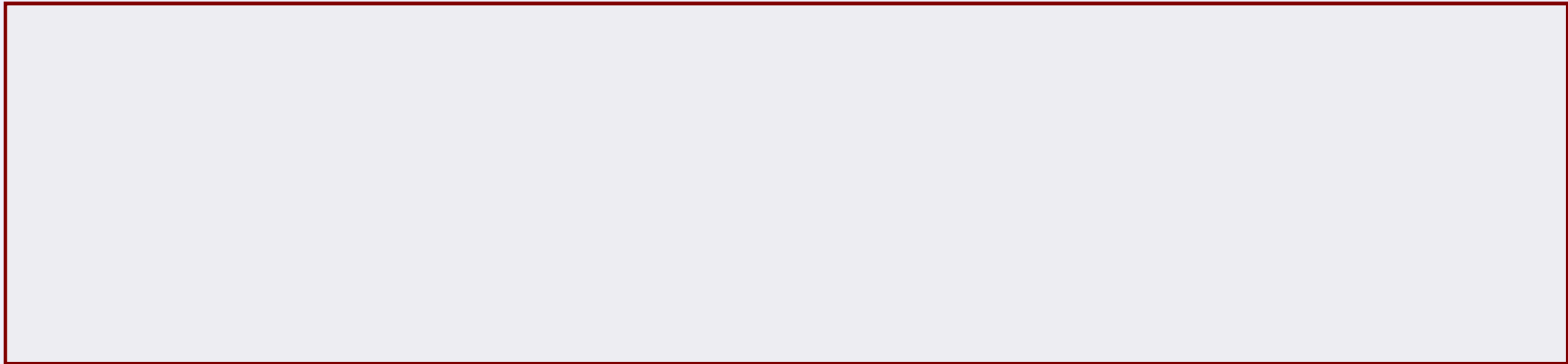
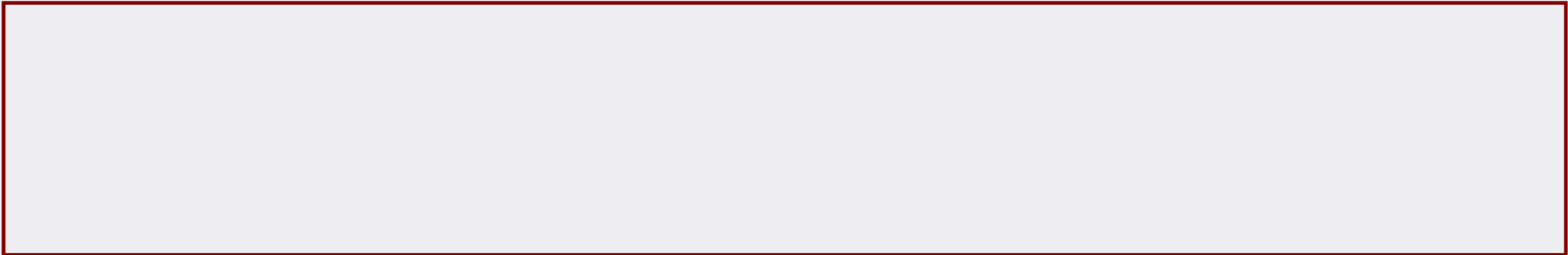
Der LIKE-Operator dient zum flexiblen Vergleich von Texten mit einen vorgegebenen Filter-Wert. Dies ist besonders nützlich wenn als Such bzw. Filter-Wert nur ein Teil der Informationen zur Verfügung steht.

In dem Vergleichswert kann mit Platzhaltern gearbeitet werden.

Platzhalter	Erklärung	Beispiel	Ergebnis
'%'	Prozent steht für kein, ein oder mehrere beliebige Zeichen.	Name Like ' F%' Name Like ' %son' Name Like ' %ill%'	Funke, Franz Benson, Jenson, Morrison Miller, Filler, Ofillson
'_'	Unterstrich steht für genau ein beliebiges Zeichen.	Name Like ' M_ller' Name Like ' ____'	Müller, Möller, Miller Adas, Funk, Tier

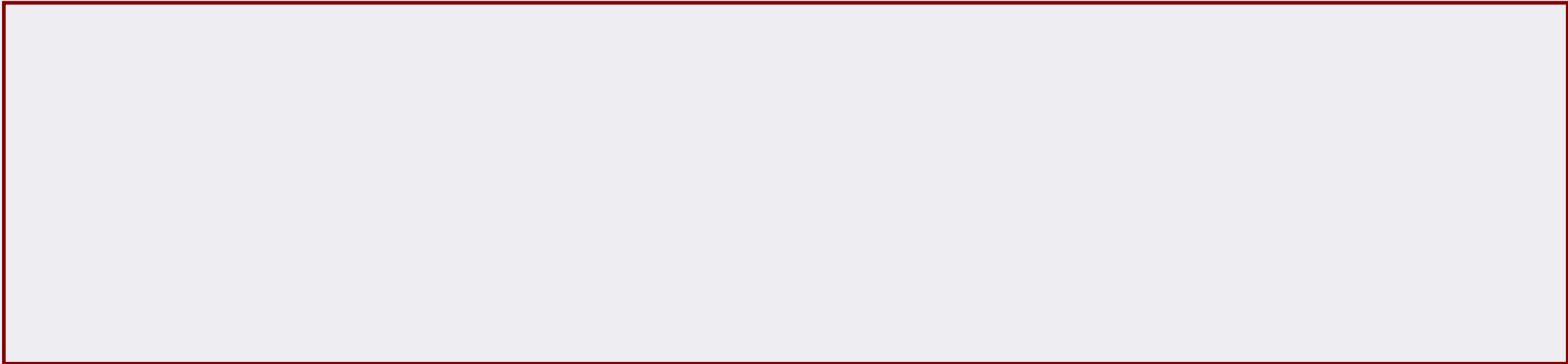
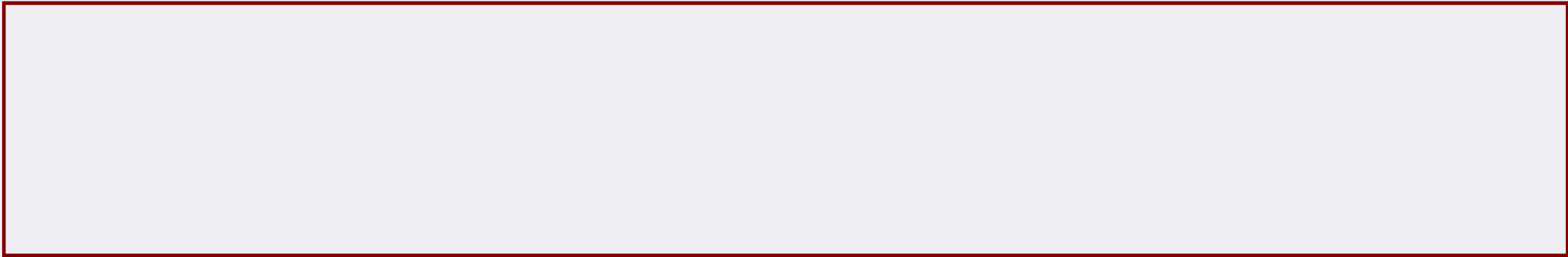


Der IN-Operator dient zum prüfen ob der Wert eines Datenfeldes in einer Liste von Werten enthalten ist. Dazu muss eine Werteliste mit den entsprechenden Suchwerten definiert werden.



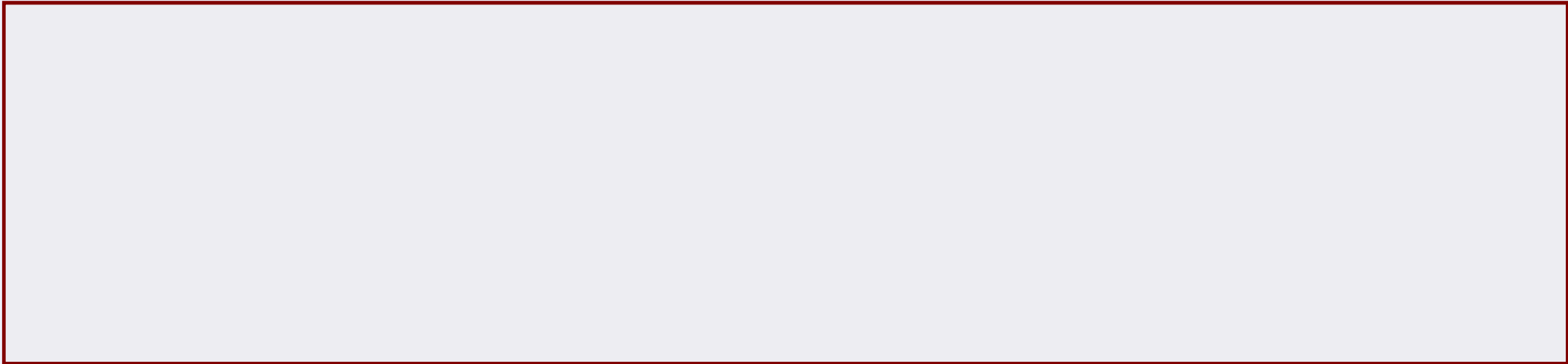
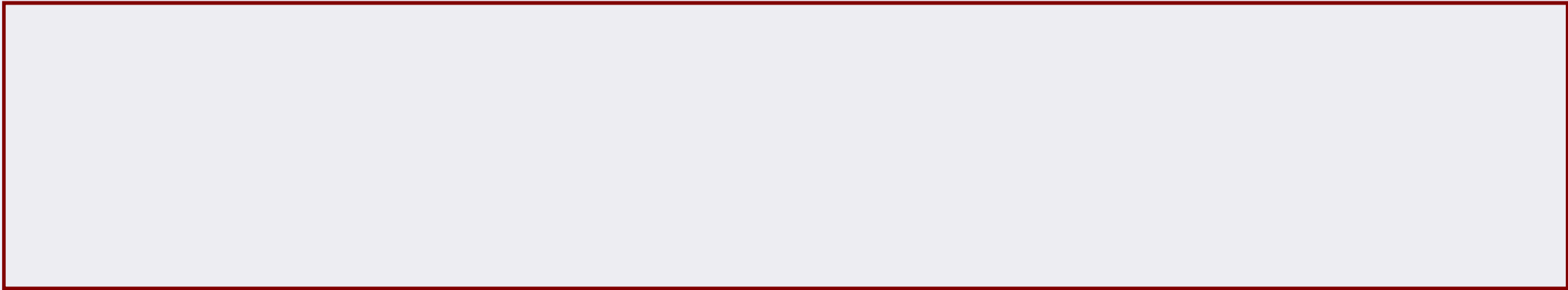


Der BETWEEN-Operator dient zur Prüfung bzw. Filterung ob der Wert eines Datenfeldes in einem bestimmten Wertebereich liegt.





Der Nullwert-Operator dient zur Prüfung bzw. Filterung ob der Wert eines Datenfeldes Daten bzw. keine Daten enthält.





Beim Vergleich werden zwei Ausdrücke mittels eines Operators miteinander verglichen. Die Ausdrücke können Datenfelder bzw. konstante Werte sein.

<	kleiner als
>	größer als
< >	Ungleich
=	Gleich
<=	Kleiner gleich
>=	größer gleich

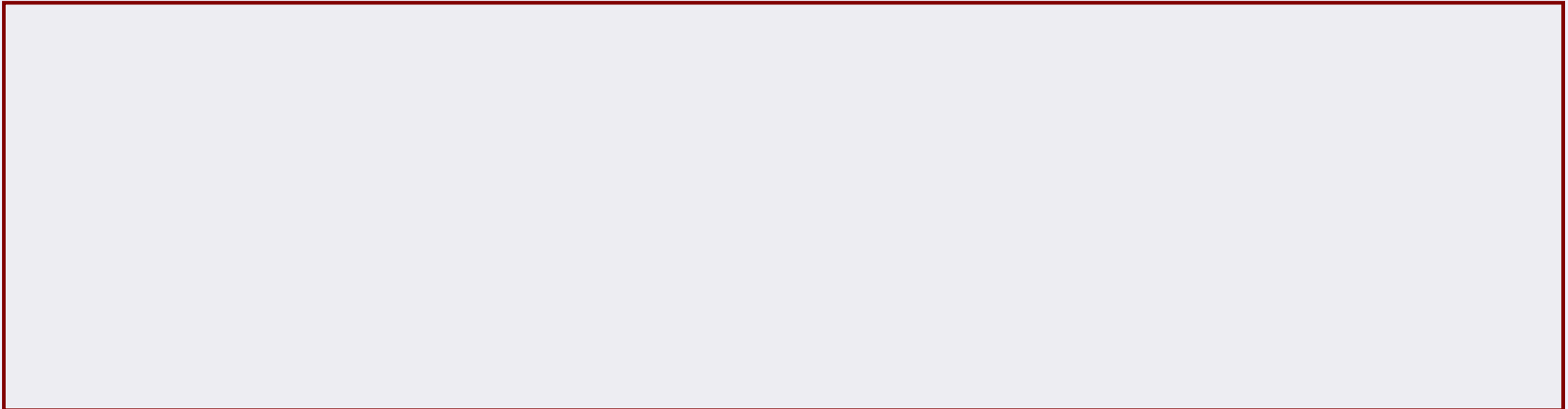
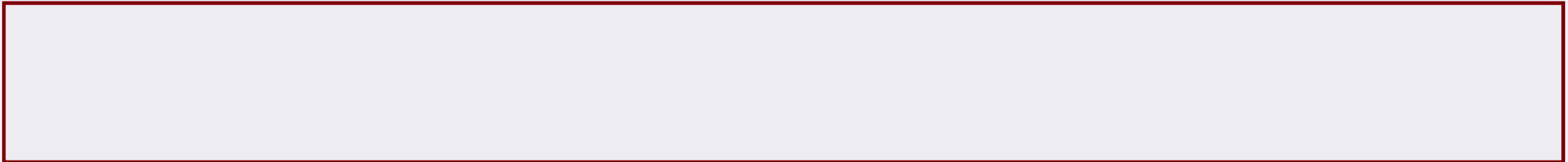


Mit Hilfe von logischen Operatoren werden mehrere Bedingungen miteinander verbunden.

AND	UND - Beide Bedingungen müssen erfüllt sein.
OR	ODER- Eine Bedingung muss erfüllt sein.
NOT	NICHT- Negiert die folgende Bedingung.



Der BETWEEN-Operator dient zur Prüfung bzw. Filterung ob der Wert eines Datenfeldes in einem bestimmten Wertebereich liegt.





In einer SELECT-Anweisung können verschiedene Berechnung durchgeführt werden. Die Ergebnisse werden als Inhalt eines Datenfeldes behandelt. Die Berechnung erfolgt mit numerischen Datenfelder oder konstanten Werten..

+ -

Addition und Subtraktion

*** /**

Multiplikation und Division



In vielen Fällen werden nicht die einzelnen Datensätze einer Abfrage benötigt, sondern Informationen über den Inhalt dieser Datensätze. Für diesen Zweck existieren Funktionen. Der Feldname ist in Runden Klammer anzugeben.

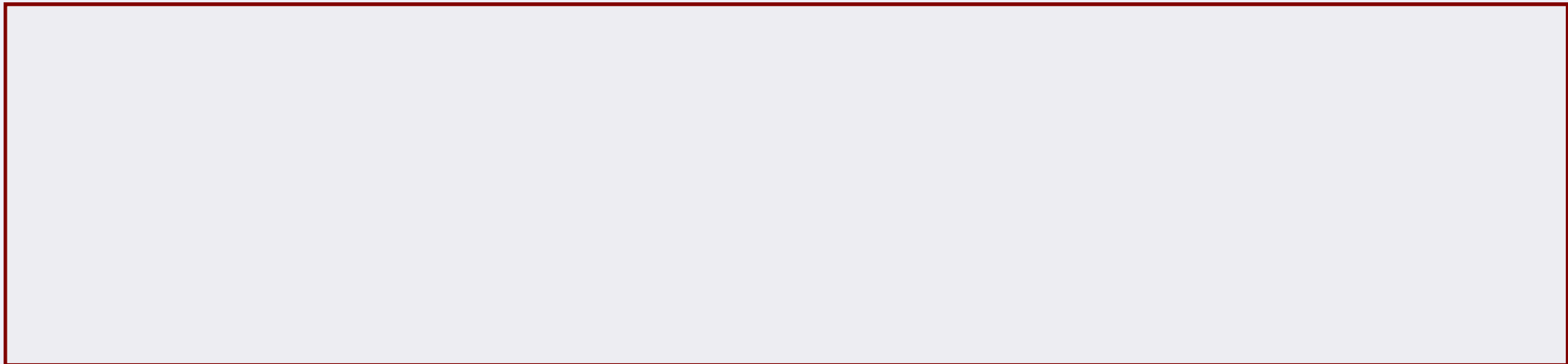
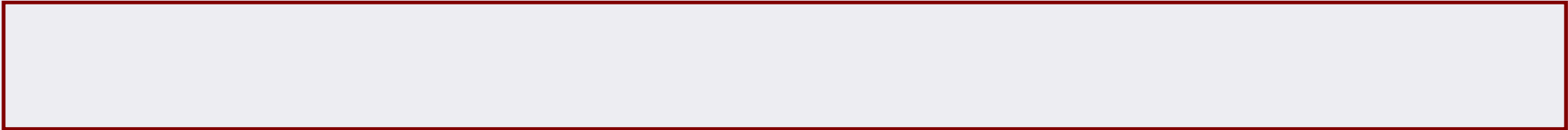
Count()	Anzahl der Werte, außer Nullwerte
AVG()	Durchschnittswert
MIN()	Kleinster Wert
MAX()	Größter Wert
SUM()	Summe aller Werte



abs()	absoluter Wert einer Zahl
round()	Gerundeter Wert einer Zahl
mod()	Rest einer Ganzzahldivision
pi()	Wert von Pi
log()	natürlicher Logarithmus
sin(), cos()	Winkelfunktionen
sqrt()	Quadratwurzel einer Zahl
length()	Länger einer Zeichenkette
lower()	Zeichenkette in Kleinbuchstaben
upper()	Zeichenkette in Großbuchstaben
substring()	Teilstück einer Zeichenkette



Das Gruppieren von Datensätzen ermöglicht es Datensätze mit gleichem Inhalt zu einer Gruppe zusammenzufassen. Durch die GROUP BY- Klausel lassen sich Abfragen nach einem oder mehreren Datenfeldern gruppieren. Dabei wird erst nach dem ersten Datenfeld gruppiert, innerhalb der ersten Gruppierung nach dem zweiten Datenfeld usw. Mittels der Aggregatfunktionen können Werte für jede Gruppe berechnet werden. Mit der HAVING – Klausel wird die Ergebnismenge einer gruppierten Abfrage eingeschränkt.





Die Select – Anweisung liefert die Datensätze in keiner definierten Reihenfolge. Um das Abfrageergebnis nach einem oder mehreren Datenfeldern zu sortieren wird die ORDER BY – Klausel verwendet. Standardmäßig wird in aufsteigender Reihenfolge sortiert. (ASC ascending) Um eine absteigende Reihenfolge zu erhalten muss der Parameter DESC (descending) verwendet werden..

