



<i>SELECT</i>	<i>Spaltenliste</i>	<ul style="list-style-type: none"> <li>- <b>DISTINCT</b> vermeidet Duplikate.</li> <li>- Aggregatfunktionen (<b>COUNT</b>, <b>SUM</b>, <b>MAX</b>, <b>MIN</b>, <b>AVG</b>) für Berechnungen</li> <li>- <b>SELECT *</b> für "alle Spalten"</li> <li>- <b>AS</b> Aliasname</li> </ul>
<i>FROM</i>	<i>Tabelle</i>	
<i>WHERE</i>	<i>Bedingung</i>	<ul style="list-style-type: none"> <li>- Wird meist mit <b>Vergleichen</b> (&lt;, &lt;=, =, &gt;, ...) formuliert.</li> <li>- Verknüpfung von mehreren Vergleichen mit <b>logischen Funktionen</b> (<b>AND</b>, <b>OR</b>, <b>NOT</b>)</li> </ul>
<i>GROUP BY</i>	<i>Spaltenliste</i>	
<i>HAVING</i>	<i>Bedingung</i>	
<i>ORDER BY</i>	<i>Spaltenliste</i>	<ul style="list-style-type: none"> <li>- <b>ASC</b> für aufsteigend (Standard)</li> <li>- <b>DESC</b> für absteigend</li> </ul>

Im Detail gilt:

Grundlegende SQL-Abfrage	
SELECT	Es muss mindestens ein Spaltenname angegeben werden. Die entsprechende(n) Spalte(n) sind dann Teil der Ergebnistabelle. SELECT * bewirkt, dass alle Spalten angezeigt werden.
DISTINCT	Duplikate von Datensätzen werden nicht angezeigt.
AS	Eine Spalte in der Ergebnistabelle kann anders benannt werden als in der Ausgangstabelle. Dies ist vor allem bei der Verwendung von Aggregatfunktionen hilfreich.
FROM	Hier muss angegeben werden, aus welcher Tabelle die Informationen für die Abfrage genommen werden sollen.
ORDER BY	Die Ergebnistabelle wird nach der oder den angegebenen Spalten sortiert. Standardmäßig wird aufsteigend sortiert. Mit dem Zusatz DESC bzw. ASC wird absteigend bzw. aufsteigend sortiert.
Beispiele	<pre>SELECT DISTINCT kontinent AS "enthaltene Kontinente" FROM Land  SELECT name, flaeche, hauptstadt FROM LAND ORDER BY flaeche DESC</pre>

Auswahl von Datensätzen über Bedingungen	
WHERE	In der Ergebnistabelle werden nur die Datensätze (Zeilen) angezeigt, welche die angegebene Bedingung erfüllen. Eine Bedingung wird mit einem Vergleich formuliert. Neben den typischen Vergleichsoperatoren wie <, <=, =, >, usw. sind insbesondere auch IS NULL und LIKE wichtig. Mehrere Vergleiche können durch die logischen

	<p>Funktionen AND, OR und NOT verknüpft werden. Ggf. müssen die einzelnen Ausdrücke dabei sinnvoll geklammert werden</p> <p>Beispiel</p> <pre>WHERE   jahr &gt; 2015 AND     laufzeit &lt;= 90 AND     NOT fsk = 18</pre>
LIKE	<p>Kann in einer Bedingung zur Mustererkennung von Einträgen verwendet werden. Folgende zwei Platzhalter (wildcards) werden häufig eingesetzt:</p> <ul style="list-style-type: none"> <li>• - % steht für beliebig viele Zeichen, auch keines (* bei MS Access)</li> <li>• , _ für genau ein beliebiges Zeichen (? bei MS Access)</li> </ul> <p>Beispiele:</p> <ul style="list-style-type: none"> <li>• WHERE titel LIKE "You%" – findet alle Titel die mit "You" beginnen Groß-/Kleinschreibung wird nicht berücksichtigt</li> <li>• WHERE titel LIKE "%love%" – findet alle Titel die "love" enthalten</li> <li>• WHERE titel LIKE "L____" – findet alle Titel die mit L beginnen und genau 4 Zeichen lang sind.</li> </ul>
NULL	Bedeutet, dass kein Wert in einer Zelle eingetragen ist.
IS NULL	Überprüft (in einer Bedingung), ob kein Wert in einer Zelle eingetragen ist.

Aggregatsfunktionen	
AVG	Berechnet den Durchschnitt aller Werte einer Spalte.
COUNT	Gibt die Anzahl der Einträge einer Spalte aus.
MAX bzw. MIN	Gibt das Maximum bzw. Minimum aller Werte einer Spalte aus.
SUM	Berechnet die Summe aller Werte einer Spalte.
Beispiel	<pre>SELECT   COUNT(*) AS "Anzahl afrikanischer Länder " FROM     Land WHERE    kontinent = "Afrika"</pre>

Gruppierung	
GROUP BY	Datensätze mit demselben Wert in der angegebenen Spalte werden gruppiert. Gruppierungen sind nur in Kombination mit Aggregatfunktionen sinnvoll.
HAVING	An gruppierte Datensätze werden Bedingungen mit HAVING formuliert.
Beispiel	<pre>SELECT fsk, MIN(laufzeit) FROM Film WHERE genre1="Filmkomödie" OR genre2="Filmkomödie" GROUP BY fsk HAVING fsk &lt;16</pre>