

HIT 2.Jahr	Datum:
SQL – Grundlagen für Abfragen mit Berechnungen, Gruppierung, Sortierung, Unterabfrage, Datumsfunktionen	

SELECT-Anweisung mit Erweiterungen

1.1. Sortierung

Die Ergebnisse einer Abfrage können mit der Funktion ORDER BY sortiert werden.

```
SELECT * FROM tbl_name ORDER BY spalte1, spalte2;
```

Um die Sortierreihenfolge fest zu legen gibt es die Schlüsselworte ASC (ascending = aufsteigend) und DESC (descending = absteigend).

```
SELECT * FROM tbl_name ORDER BY spalte1 ASC;
SELECT * FROM tbl_name ORDER BY spalte1 DESC;
```

Beispiel: `SELECT Name, Vorname FROM Mitglied ORDER BY Name ASC;`

Wenn kein Schlüsselwort verwendet wird, wird das Ergebnis aufsteigend sortiert.

1.2. Ergebnismenge eingrenzen: LIMIT

Mit LIMIT kann die Ergebnismenge der Datensätze einer Abfrage eingeschränkt werden. Beispielsweise wird nach einer Sortierung mit LIMIT 1 nur der erste (oberste) Datensatz angezeigt.

Beispiele für den Einsatz von LIMIT:

Die Select-Anweisung liefert das Datum und die Bezeichnung aller Veranstaltungen der DB. Sie werden nach dem Datum absteigend sortiert, d.h. die neueste Veranstaltung wird zuerst angezeigt. Durch die Einschränkung mit LIMIT 1 wird auch nur diese einzige, neueste Veranstaltung angezeigt.

```
SELECT Datum, Bezeichnung FROM Veranstaltung ORDER BY DESC LIMIT 1;
```

1.3. Berechnungen: SUM(), AVG(), MIN(), MAX()

SUM und AVG werden zur Berechnung von Summen und Durchschnittswerten herangezogen. Mithilfe von MIN und MAX wird ein kleinster bzw. größter Wert ermittelt.

Beispiele für einfache Berechnungen:

Die erste Select-Anweisung berechnet das Gesamtbudget aller Veranstaltungen des Sportvereins ab 2013. Die zweite Select-Anweisung bestimmt den höchsten Beitrag von allen Beitragsklassen.

```
SELECT SUM(Budget) FROM Veranstaltung WHERE YEAR(Datum) >= 2013;
SELECT MAX(Beitrag) FROM Beitragsklasse;
```

1.4. Gruppenfunktion (Gruppierung/Aggregation)

- GROUP BY gruppiert die Datensätze nach einem bestimmten Kriterium bzw. Spalte.

Beispiel: Liefert die Anzahl der Mitglieder gruppiert/unterschieden nach dem Wohnort, d.h. Die jeweilige Anzahl der Mitglieder für die unterschiedlichen in der DB angegebenen (Wohn)Orte.

```
SELECT COUNT(MitgliedNr), Ort FROM Mitglied GROUP BY Ort;
```

- HAVING wird verwendet, um gruppierte Ergebnisdatensätze einzuschränken.

Beispiel: Liefert die Anzahl der Mitglieder gruppiert nach dem Wohnort, wobei die Anzahl nur angezeigt wird, wenn mehr als 2 Mitglieder in einem Ort wohnen.

```
SELECT COUNT(MitgliedNr) AS 'Anzahl' FROM Mitglied (AS s. nachfolg. Hinweis)
GROUP BY Ort HAVING Anzahl > 2; (HAVING nur in Verbindung mit GROUP)
```

HIT 2.Jahr	Datum:
SQL – Grundlagen für Abfragen mit Berechnungen, Gruppierung, Sortierung, Unterabfrage, Datumsfunktionen	

- **Hinweis:** AS steht für ALIAS

AS wird eingesetzt, um einen anderen Namen als den Spaltennamen zu verwenden. Wird zum Beispiel für eine aussagekräftige Bildschirmausgabe verwendet oder um Namen abzukürzen.

Beispiel:

```
SELECT COUNT(*) AS Anzahl_der_Veranstaltungen FROM Veranstaltung;
SELECT GebDatum AS 'Geburtsdatum' FROM Mitglied;
```

1.5. Unterabfragen

Eine Unterabfrage ist eine SELECT-Abfrage in einer anderen Abfrage, die vom Typ SELECT, INSERT, UPDATE, DELETE oder SET sein kann.

- Unterabfragen müssen immer in Klammern angegeben werden.
- Es können mehrere Unterabfragen ineinander verschachtelt werden.
- Jedoch darf eine Unterabfrage nicht mehr als einen Datensatz zurückliefern.

Beispiel: In einer SELECT-Abfrage, die Mitgliedsnummer, Name, Vorname und Telefonnummer aller Mitglieder des Sportvereins liefern soll, die einen Schülerbeitrag bezahlen, ist eine SELECT-Unterabfrage mit der Abfrage nach der Schüler-Beitragsklasse enthalten.

```
SELECT MitgliedNr, Name, Vorname FROM Mitglied
WHERE Mitglied.Klasse=(SELECT Beitragsklasse.Klasse FROM Beitragsklasse
                        WHERE Bezeichnung LIKE 'Schueler');
```

Hinweis: Bei mehrfachem Vorkommen von Spaltennamen, z.B. Klasse, müssen diese unterschieden werden können. Dies geschieht mit Hilfe des Tabellennamens. Der Tabellename zusammen mit dem Spaltennamen identifiziert (qualifiziert) jede Spalte eindeutig, wie z.B.

Mitglied.Klasse und Beitragsklasse.Klasse

1.6. Datumsfunktionen

- CURDATE(), CURTIME(), NOW() liefern das aktuelle Datum, die aktuelle Zeit sowie Datum und Zeit zusammen.

Beispiel: Das aktuelle Datum und die Uhrzeit werden geliefert.
SELECT NOW();

- DAY(), MONTH(), YEAR() splitten ein Datum auf.

Beispiel: Es werden alle Mitglieder gesucht, die im Dezember Geburtstag haben.
SELECT GebDat, Name FROM Mitglied WHERE MONTH(Datum)=12;

- HOUR(), MINUTE(), SECOND() splitten eine Zeitangabe auf.
- DATE_ADD(), DATE_SUB() addieren bzw. subtrahieren von einem Datum ein Zeitintervall, das in Tagen (DAY), Monaten (MONTH) oder Jahren (YEAR) angegeben wird.

Beispiel: Im Beispiel werden alle Aufgaben mit Datum und Aufgabennummer sowie der Bezeichnung aufgeführt, die in den letzten 10 Tagen erledigt wurden.

```
SELECT Datum, AufgNr, Bezeichnung FROM Aufgabe
WHERE Datum DATE_ADD(CURDATE(), - INTERVAL 10 DAY);
```

Es gibt noch eine Vielzahl weiterer Datums- und Zeitfunktionen.

HIT 2.Jahr	Datum:
SQL – Grundlagen für Abfragen mit Berechnungen, Gruppierung, Sortierung, Unterabfrage, Datumsfunktionen	

Aufgaben: Beantworten Sie aus der Sportverein-Datenbank „sv“ die nachfolgenden Fragestellungen.

1. Bestimmen Sie das Gesamt-Budget des Sportvereins. Es setzt sich aus den Budgets aller Abteilungen zusammen.
2. Welche Mitglieder haben im Mai Geburtstag?
3. Welches Mitglied (Vorname, Name, Geburtsdatum) feiert als erster im Jahr Geburtstag? (Hinweis: limit)
4. Ermitteln Sie die unterschiedlichen Wohnorte der Sportvereins-Mitglieder, die mit der Spaltenüberschrift 'Unterschiedliche_Wohnorte' ausgegeben werden sollen.
5. Bestimmen Sie die Anzahl der Mitglieder (Mitgliedsanzahl), die im Verein keine besondere Funktion einnehmen.
6. Welche Veranstaltungen finden im November statt?
7. Welche Veranstaltungen finden zwischen dem 15. November und dem 15. Dezember dieses Jahres statt?
8. Wie viel Stunden sind für die Vorbereitung der Weihnachtsfeier geplant? (Hinweis: Unterabfrage)
9. Jedes Vereinsmitglied erhält zu seinem Geburtstag eine Geburtstags-E-Mail. Listen sie für jeden Monat die Anzahl der E-Mails, die zu verschicken sind.
10. Suchen Sie das jüngste Mitglied des Sportvereins.
11. Suchen Sie alle Aufgaben (AUFG_ID), für die noch keine Helfer eingetragen sind.
12. Stellen Sie die Adressdaten des Kassenwartes zusammen? (Hinweis: Unterabfrage)
13. Ermitteln Sie alle Mitglieder (Vorname, Name), die einen Familienbeitrag zahlen. (Hinweis: Unterabfrage)