**Modul 7: SQL für Fortgeschrittene Tag 2**

**Themen:**

**JOINS:**

Wird verwendet, um die Zeilen und Spalten von 2 oder mehr Tabellen zu verbinden, basierend auf einer verwandten Spalte zwischen ihnen.

**INNER JOIN-Syntax:**

SELECT Spalte FROM Tab1

INNER JOIN Tab2

ON Tab1.Spalte = Tab2.Spalte;

Wählt Datensätze aus, die in beiden Tabellen übereinstimmende Werte aufweisen.

**LEFT JOIN-Syntax:**

SELECT Tab1.Spalte FROM Tab1

LEFT JOIN Tab2

ON Tab1.Spalte = Tab2.Spalte;

Gibt alle Datensätze aus der linken Tabelle (Tab1) und die übereinstimmenden Datensätze aus der rechten Tabelle (Tab2) zurück. Das Ergebnis ist 0 Datensätze von der rechten Seite, wenn es keine Übereinstimmung gibt.



**RIGHT JOIN-Syntax:**

SELECT Tab1.Spalte FROM Tab1

RIGHT JOIN Tab2

ON Tab1.Spalte = Tab2.Spalte;

Gibt alle Datensätze aus der rechten Tabelle (Tab2) und die übereinstimmenden Datensätze aus der linken Tabelle (Tab1) zurück. Das Ergebnis ist 0 Datensätze von der linken Seite, wenn es keine Übereinstimmung gibt.

**FULL OUTER JOIN-Syntax:**

SELECT Tab1.Spalte FROM Tab1

FULL OUTER JOIN Tab2

ON Tab1.Spalte = Tab2.Spalte

WHERE Bedingung;

Gibt alle Datensätze zurück, wenn eine Übereinstimmung in linken (Tab1) oder rechten (Tab2) Tabellendatensätzen vorliegt.

**SELF-JOIN-Syntax:**

SELECT Tab1.Spalte FROM Tab1 T1, Tab1 T2

ON Tab1.Spalte = Tab2.Spalte;

Ein SELF-JOIN ist ein regulärer JOIN, aber die Tabelle ist mit sich selbst verbunden. T1 und T2 sind unterschiedliche Tabellenaliase für dieselbe Tabelle

**UNION-Syntax:**

SELECT Tab1.Spalte FROM Tab1

UNION

SELECT Tab2.Spalte FROM Tab2;

Der Operator wird verwendet, um die Ergebnismenge von zwei oder mehr Anweisungen zu kombinieren. Jede Anweisung muss die gleiche Anzahl von Spalten haben SLECT UNION. Die Spalten müssen auch ähnliche Datentypen aufweisen. Die Spalten in jeder Anweisung müssen ebenfalls in der gleichen Rheinenfolge sein.

**UNION ALL-Syntax:**

SELECT Tab1.Spalte FROM Tab1

UNION ALL

SELECT Tab2.Spalte FROM Tab2;

Der Operator wählt standardmäßig nur unterschiedliche Werte aus. Um doppelte Werte zuzulassen, verwende: UNION UNION ALL.

**Material:**

Sehen Sie sich die Videos 29-32, 54 der folgenden Playlist an:

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

*Sehen Sie sich auch das folgende Video an:*

[*https://www.youtube.com/watch?v=vncBSUNb4NA*](https://www.youtube.com/watch?v=vncBSUNb4NA)

Gehen Sie das folgende Schulungsmaterial durch – „SQL Joins“ – „SQL Unions“:

[*https://www.w3schools.com/sql/sql\_join.asp*](https://www.w3schools.com/sql/sql_join.asp)

**Fragen:**

1. **Auflisten der Möglichkeiten zum Abrufen der Anzahl der Datensätze in einer Tabelle?**
   * C
2. **Wozu sind Gruppenfunktionen/Aggregatfunktionen in SQL erforderlich?**
   * Zum Auswerten der Datensätze
3. **Wie können Sie die ersten 5 Zeichen der Zeichenfolge abrufen?**
   * Mit LEFT(5)
4. **Was ist Self Join und Cross Join?**
   * Ein **Self Join** ist ein **Join** einer Tabelle mit sich selbst. Das bedeutet, dass Datensätze der Tabelle mit anderen Datensätzen derselben Tabelle verglichen werden müssen.
   * Der **Cross Join** (auch als Kartesisches Produkt oder Kreuzprodukt bezeichnet)verbindet jede Zeile der ersten Tabelle mit jeder Zeile der Zweiten Tabelle. Die Ergebnistabelle eines Cross Joins kann sehr groß werden und ist häufig nutzlos.
   * A self join is a join of a table with itself. This means that records of the table have to be compared with other records of the same table.
   * The cross join (also known as Cartesian product or cross product) connects each row of the first table with each row of the second table. The result table of a cross join can get very large and is often useless
5. **Was ist der Unterschied zwischen den Operatoren Union und Union All?**
   * UNION: Der Operator wird verwendet, um die Ergebnismenge von zwei oder mehr Anweisungen zu kombinieren. Jede Anweisung muss die gleiche Anzahl von Spalten haben SLECT UNION. Die Spalten müssen auch ähnliche Datentypen aufweisen. Die Spalten in jeder Anweisung müssen ebenfalls in der gleichen Rheinenfolge sein.
   * UNION ALL: Der Operator wählt standardmäßig nur unterschiedliche Werte aus. Um doppelte Werte zuzulassen, verwende: UNION UNION ALL.
   * UNION: The operator is used to combine the result set of two or more statements. Each statement must have the same number of columns SLECT UNION. The columns must also be of similar data types. The columns in each statement must also be in the same row order.
   * UNION ALL: The operator selects only different values ​​by default. To allow duplicate values ​​use: UNION UNION ALL
6. **Was ist die SQL-Abfrage, um das aktuelle Datum anzuzeigen?**
   * SELECT \* FROM tabelle

WHERE DATE\_SUB(NOW);

1. **Welche Arten von Unterabfragen gibt es?**
   * Entweder liefert die Abfrage einen Wert oder eine Zeile oder mehrere Zeilen.
   * The query returns either a value or a row or multiple rows.
2. **Was ist der Unterschied zwischen einer verschachtelten Unterabfrage und einer korrelierten Unterabfrage?**
   * Eine **korrelierte Unterabfrage** kann als Filter für die Tabelle betrachtet werden, auf die sie verweist, als ob die **Unterabfrage** für jede Zeile der Tabelle in der äußeren Abfrage ausgewertet würde.
   * Eine **unkorrelierte Unterabfrage** enthält keine derartigen externen Spaltenverweise
   * A correlated subquery can be viewed as a filter on the table it references, as if the subquery were evaluated for each row of the table in the outer query.
   * An uncorrelated subquery contains no such external column references

**Aufgaben:**

1. **Schreiben Sie eine Abfrage, um den Namen (Vorname, Nachname), die Abteilungs-ID und den Namen aller Mitarbeiter zu finden.**
   * SELECT employee.first\_name, employee.last\_name, employee.department\_id FROM employee;
2. **Schreiben Sie eine Abfrage, um den Namen (Vorname, Nachname), die Stelle, die Abteilungs-ID und den Namen der Mitarbeiter zu finden, die in London arbeiten.**
   * SELECT employees.first\_name, employees.last\_name, employees.job\_id, employees.department\_id, departments.location\_id, locations.city

FROM employees

LEFT JOIN departments

ON employees.department\_id = departments.department\_id

Left JOIN locations

ON departments.location\_id = locations.location\_id

WHERE Locations.city ='London';

1. **Schreiben Sie eine Abfrage, um die Mitarbeiter-ID, den Namen (Nachname) zusammen mit der Manager-ID und dem Namen (Nachname) zu finden.**
   * SELECT employee\_id, last\_name, manager\_id, FROM employees;
2. **Schreiben Sie eine Abfrage, um den Namen (Vorname, Nachname) und das Einstellungsdatum der Mitarbeiter zu finden, die nach „Jones“ eingestellt wurden.**
   * select \* from employees

WHERE last\_name = 'Jones';

* + SELECT employees.first\_name, employees.last\_name, employees.hire\_date

From employees

WHERE employees.hire\_date>'1987-09-20'

Order by employees.hire\_date;

1. **Schreiben Sie eine Abfrage, um den Abteilungsnamen und die Anzahl der Mitarbeiter in der Abteilung zu erhalten**
   * SELECT departments.department\_name, COUNT(employees.employee\_id) AS count\_employees FROM departments

LEFT JOIN employees

ON departments.department\_id = employees.department\_id;

1. **Schreiben Sie eine Abfrage, um die Mitarbeiter-ID, die Stellenbezeichnung und die Anzahl der Tage zwischen dem Enddatum und dem Startdatum für alle Stellen in Abteilung 90 zu finden.**
   * select employee\_id, job\_id, DATEDIFF(CURRENT\_DATE, hire\_date) from employees  
     where department\_id=90;
2. **Schreiben Sie eine Abfrage, um die Abteilungs-ID und den Namen und Vornamen des Managers anzuzeigen.**
   * Name und Vorname Manager fehlen???
3. **Schreiben Sie eine Abfrage, um den Abteilungsnamen, den Managernamen und die Stadt anzuzeigen.**
   * SELECT departments.department\_name, employees.last\_name, employees.manager\_id, locations.city FROM departments

JOIN employees

ON departments.manager\_id = employees.manager\_id

JOIN locations

ON departments.location\_id = locations.location\_id;

1. **Schreiben Sie eine Abfrage, um die Berufsbezeichnung und das Durchschnittsgehalt der Mitarbeiter anzuzeigen.**
   * SELECT job\_id, AVG(salary) AS average\_salary FROM employees

GROUP BY Job\_id;

1. **Schreiben Sie eine Abfrage, um die Berufsbezeichnung, den Namen des Mitarbeiters und die Differenz zwischen dem Gehalt des Mitarbeiters und dem Mindestgehalt für den Job anzuzeigen.**
2. **Schreiben Sie eine Abfrage, um den Jobverlauf anzuzeigen, der von einem Mitarbeiter ausgeführt wurde, der derzeit mehr als 10000 Gehalt bezieht.**
3. **Schreiben Sie eine Abfrage, um den Abteilungsnamen, den Namen (Vorname, Nachname), das Einstellungsdatum und das Gehalt des Managers für alle Manager anzuzeigen, deren Erfahrung mehr als 15 Jahre beträgt.**