* **Allgemeines**

[Alles einklappen](https://elearning.edu.szf.at/course/view.php?id=2874)

* + Sie **verwenden Subqueries, Views und virtuelle Tabellen**um Daten detaillierter auswerten zu können!
  + In dieser Lerneinheit werden Sie sich folgende Kompetenzen erarbeiten:
    - Sie erstellen Unterabfragen.
    - Sie speichern Abfrageergebnisse in virtuelle Tabellen.

**Damit werden Sie folgendes Ziel erreichen:**

* + - Sie verwenden Subqueries und Views um Daten in virtuellen Tabellen zu speichern.

**Zeit für diese Einheit ca. 10 Unterrichtseinheiten**

* + **Information**

**Subqueries**in SQL **dienen**dazu, **komplexe Abfragen** auszuführen, indem sie eine **Abfrage innerhalb einer anderen Abfrage** ausführen. Sie **ermöglichen**es einem **Benutzer**, **Daten** aus einer **Tabelle**zu **extrahieren**und diese **Ergebnisse**in einer **anderen Abfrage**zu verwenden, um weitere Informationen zu finden oder **bestimmte Bedingungen**zu erfüllen.

**Subqueries können**in **SELECT-, INSERT-, UPDATE- und DELETE-Statements** **verwendet**werden und bieten eine **flexible Möglichkeit**, **Daten**aus **mehreren Tabellen** zu **kombinieren**, komplexe Filterbedingungen zu erstellen oder aggregierte Informationen zu extrahieren. Ein weiterer Vorteil von **Subqueries**ist, dass sie es Benutzern ermöglichen, a**uf Daten zuzugreifen**, die **normalerweise nicht in einer Abfrage verfügbar** sind, wie z.B. die **Ergebnisse**von Aggregatfunktionen wie **COUNT**, **SUM**oder **AVG**.

Insgesamt bieten Subqueries eine leistungsstarke Möglichkeit, Daten aus verschiedenen Quellen zu extrahieren und zu analysieren, was sie zu einem wichtigen Werkzeug in der SQL-Programmierung macht.

In **SQL**bezieht sich der Begriff "**virtuelle Tabelle**" normalerweise auf eine **temporäre Tabelle,** die **von einer Abfrage generiert** wird und **nur während der Laufzeit der Abfrage existiert**. Diese Art von Tabelle wird auch als Common Table Expression (CTE) bezeichnet.

Eine **virtuelle Tabelle** ermöglicht es einem Benutzer, eine **komplexe Abfrage zu erstellen**, indem er eine oder mehrere Zwischentabellen erstellt, die während der Abfrage verwendet werden. **Virtuelle Tabellen sind nützlich,** wenn eine **Abfrage mehrere Schritte**oder komplexe Filterbedingungen erfordert, **die mit einer einzelnen Abfrage schwer zu erreichen wären**.

Ein weiterer **Vorteil**von virtuellen Tabellen ist, dass sie die **Lesbarkeit**von Abfragen verbessern können, indem sie komplexe Unterabfragen in leicht verständliche, eigenständige Abfragen aufteilen. Dies erleichtert auch die Wartung von Abfragen, da Änderungen an einer virtuellen Tabelle automatisch in der gesamten Abfrage aktualisiert werden.

Um eine **virtuelle** **Tabelle**in **SQL**zu erstellen, wird eine **CTE-Abfrage verwendet**, Der Name der virtuellen Tabelle kann dann in der Hauptabfrage verwendet werden, um auf die Zwischentabelle zuzugreifen.

In **SQL** bezeichnet eine **View eine virtuelle Tabelle**, die aus **einer oder mehreren Tabellen in einer Datenbank erstellt wird**. Eine View ist im Wesentlichen eine benannte Abfrage, die gespeichert wird, um sie bei Bedarf einfach wiederzuverwenden.

Eine **View**ermöglicht es Benutzern, **eine bestimmte Ansicht von Daten** in einer Datenbank zu erstellen, indem sie eine Abfrage erstellen, die eine oder mehrere Tabellen abfragt und die Ergebnisse in einer separaten virtuellen Tabelle speichert. Die erstellte View wird mit einem eigenen Namen versehen und kann dann von Benutzern abgefragt werden, ohne dass die zugrunde liegenden Tabellen direkt abgefragt werden müssen.

Views bieten eine Reihe von Vorteilen, darunter:

**Vereinfachung von Abfragen:**  
Durch die Erstellung einer View können Benutzer komplexe Abfragen auf einfache Weise wiederholen, ohne jedes Mal die gesamte Abfrage neu schreiben zu müssen.

**Reduktion von Redundanzen:**  
Durch die Verwendung von Views können Benutzer die gleichen Daten in mehreren Tabellen vermeiden und Redundanzen reduzieren, indem sie Abfragen auf eine einzelne View konzentrieren.

**Einschränkung von Datenzugriffen:**  
Views können so konfiguriert werden, dass sie nur einen bestimmten Teil einer Tabelle oder nur bestimmte Spalten anzeigen, um den Zugriff auf sensible Daten zu beschränken.

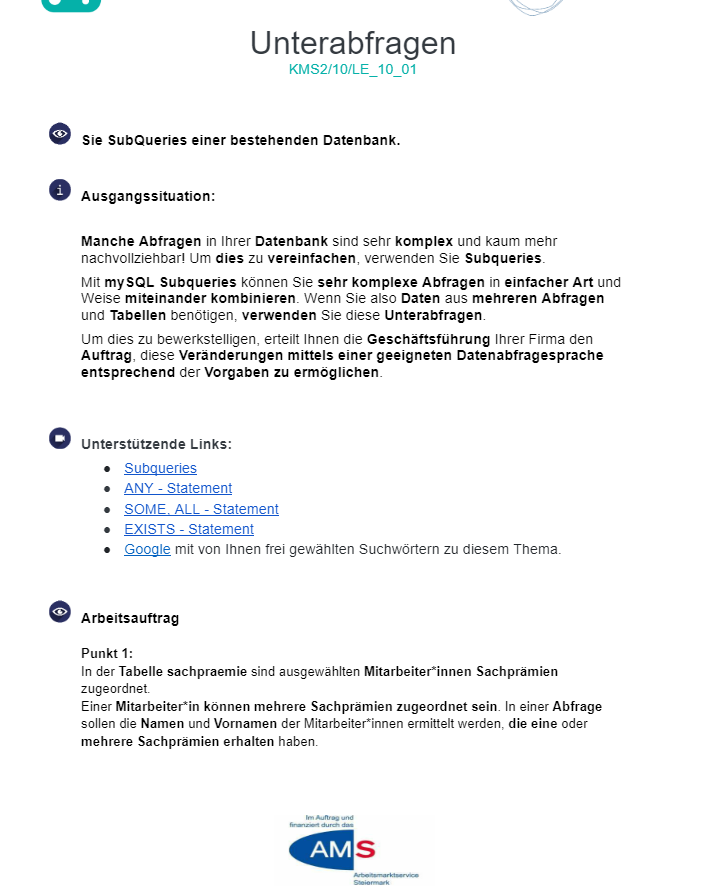
**Erhöhung der Sicherheit:**  
Durch die Verwendung von Views können Benutzer auch komplexe Berechtigungen und Einschränkungen festlegen, um den Zugriff auf Daten zu beschränken oder sicherzustellen, dass nur bestimmte Benutzer auf bestimmte Daten zugreifen können.

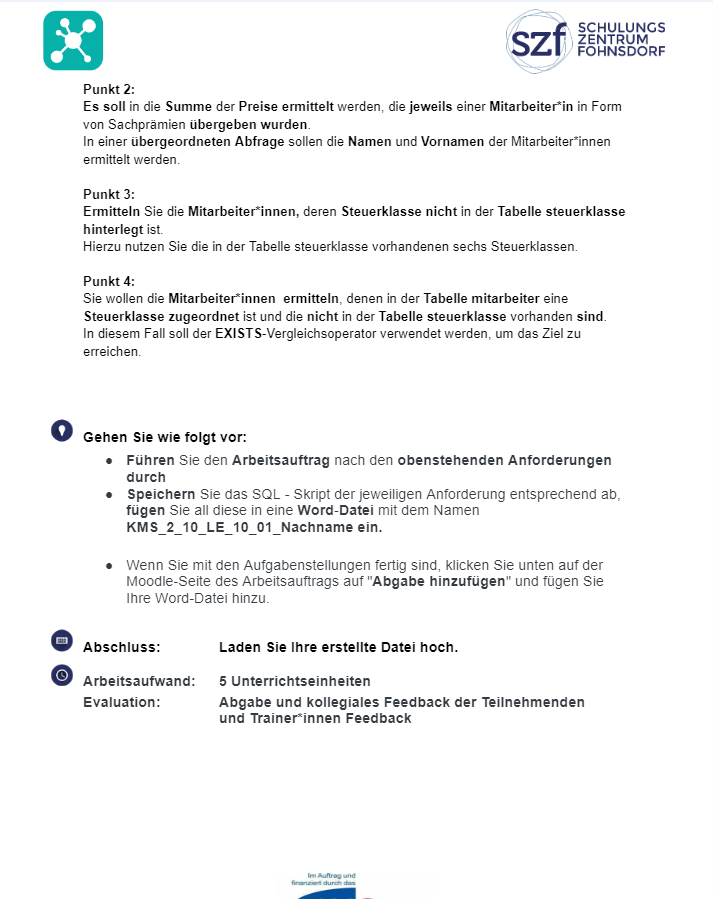
Sie beschäftigen sich mit Fragen wie:

* + - **Was ins Subqueries und warum verwendet man diese?**
    - **Was sind die Vorteil von Views?**
    - **... und vieles mehr 🙂**

Bei Fragen wenden Sie sich wie immer an Ihre **Trainer\*innen**.

Ich wünsche Ihnen für diese Einheit viel Spaß und freue mich auf eine gute Zusammenarbeit. 👍





Počiatočná situácia:: Niektoré požiadavky vo vašej databáze sú veľmi zložité a ťažko pochopiteľné! Aby ste to zjednodušili, používajte poddotazy. S poddotazmi mySQL môžete veľmi zložité požiadavky jednoducho kombinovať. Ak teda potrebujete dáta z viacerých dotazov a tabuliek, použite tieto poddotazy. Na tento účel vám vedenie vašej spoločnosti zadáva úlohu umožniť tieto zmeny pomocou vhodného jazyka pre dopytovanie dát v súlade s pokynmi.

Podporujúce odkazy:

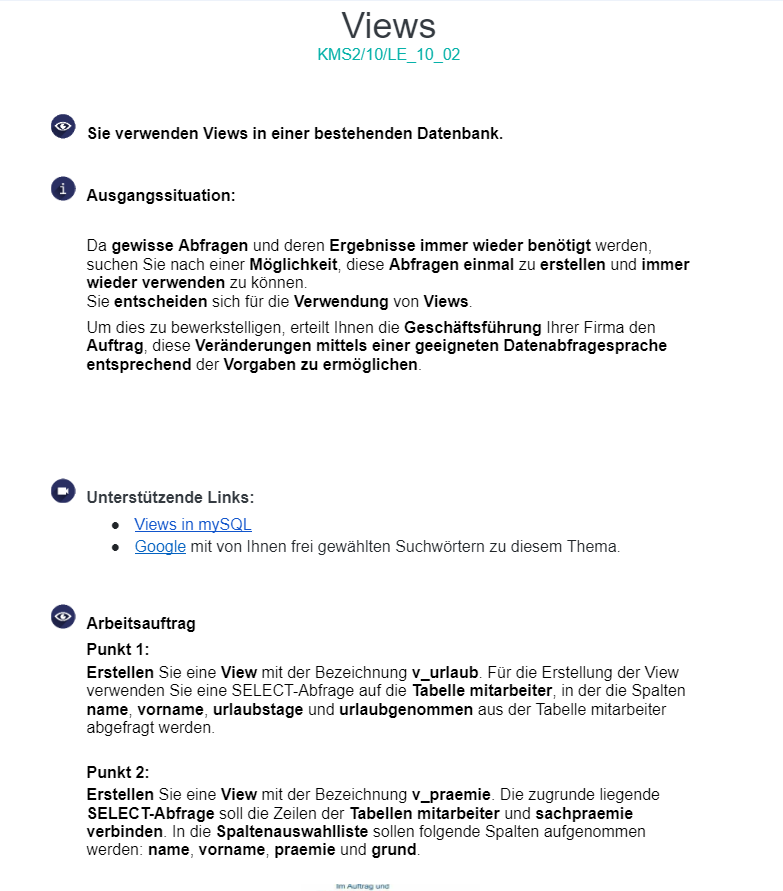
* Poddotazy
* ANY - Príkaz
* SOME. ALL - Príkaz
* EXISTS - Príkaz

Bod 1: V tabuľke "sachpraemie" sú vybraným zamestnancom priradené vecné odmeny. Jednému zamestnancovi môže byť priradených viac vecných odmien. V dotaze by sa mali zistiť mená a priezviská zamestnancov, ktorí dostali jednu alebo viac vecných odmien.

Bod 2: Malo by sa zistiť celková suma cien, ktoré boli priradené každému zamestnancovi vo forme vecných odmien. V nadradenom dopyte by sa mali zistiť mená a priezviská zamestnancov.

Bod 3: Zistite zamestnancov, ktorých daňová trieda nie je uložená v tabuľke "steuerklasse". Na tento účel použite šesť daňových tried, ktoré sú k dispozícii v tabuľke "steuerklasse".

Bod 4: Chcete zistiť zamestnancov, ktorým je priradená daňová trieda v tabuľke "mitarbeiter" a ktorí sa nenachádzajú v tabuľke "steuerklasse". V tomto prípade by sa mal použiť operátor porovnávania EXISTS na dosiahnutie cieľa.





Používate zobrazenia v existujúcej databáze.

Východisková situácia: Keďže niektoré dopyty a ich výsledky sa opakujú, hľadáte možnosť vytvorenia týchto dopytov raz a potom ich opakovane použiť. Rozhodnete sa pre použitie zobrazení. Na dosiahnutie tohto cieľa vám vedenie vašej firmy zveruje úlohu umožniť tieto zmeny pomocou vhodného jazyka pre dopytovanie dát v súlade s pokynmi.

Punkt 1: Vytvorte zobrazenie s názvom v\_urlaub. Pre vytvorenie zobrazenia použite SELECT dopyt na tabuľku "mitarbeiter", kde sa vyžadujú stĺpce "name", "vorname", "urlaubstage" a "urlaubgenommen" z tabuľky "mitarbeiter".

Punkt 2: Vytvorte zobrazenie s názvom v\_praemie. Podkladový SELECT dopyt by mal spojiť riadky tabuliek "mitarbeiter" a "sachpraemie". Do zoznamu výberu stĺpcov by mali byť zahrnuté nasledujúce stĺpce: "name", "vorname", "praemie" a "grund".

Punkt 3: Vytvorte zobrazenie s názvom v\_mitarbeiterbonus. Podkladový SELECT dopyt musí prostredníctvom JOIN spojiť riadky tabuliek "mitarbeiter" a "bonus". Okrem toho by malo byť zobrazenie vybavené triedením, ktoré zabezpečí zoradenie riadkov vzostupne podľa hodnôt stĺpca "bonus". Do zoznamu výberu stĺpcov SELECT príkazu pridajte stĺpce "name", "vorname" a "bonus".

Punkt 4: Vytvorte zobrazenie v\_mitarbeiterkrankenkasse, ktoré sa vypýta z tabuľky "mitarbeiter" nasledujúce údaje: "name", "vorname" a "krankenversicherung". Pridajte riadok do zobrazenia! (Pozor na primárny kľúč)

Punkt 5: Teraz odstráňte zobrazenie v\_mitarbeiterkrankenkasse, ktoré ste predtým vytvorili!