# Subselects / verschachtelte Abgragen mit SOME/ANY und ALL

Bei Unterabfragen (Subqueries) werden, je nachdem, ob die Unterabfrage genau ein oder mehere Ergebnistupel liefert unterschiedliche Arten von Befehlen (Prädikaten) verwendet.

Bei skalaren Ergebnis, d.h. wenn nur ein Ergebniswert zurückgegeben wird, können die skalaren Vergleichsoperatoren verwendet werden, die sonst auch für bspw. Vergleiche bei Zahlen verwendet werden. Hierzu gehören: =, !=, <, > usw.

Sobald ein oder mehrere Tupel im Resultat der Unterabfrage enthalten sind, müssen wir eine Mengenbetrachtung vornehmen. Dazu existieren folgende Operatoren: IN, {Vergleichsoperator} + ALL, {Vergleichsoperator} + ANY (oder SOME), EXISTS und UNIQUE

|  |  |
| --- | --- |
| Ausdruck | wird TRUE, wenn |
| Wert IN (Subquery)  Beispiel: select \* from student where matrikelnummer IN (…); | der angegebene Wert in der Subquery enthalten ist. |
| W (Vergleichsoperator) ALL  Beispiel: select id from manager where gehalt <= ALL (Select gehalt from manager); | wenn jedes Tupel im Subquery die Vergleichsoperation erfüllt |
| W (Vergleichsoperator) ANY  select person from kandidaten where qualifikation > ANY (select qualifikation from fussballer); (Suche alle Kandidaten, die eine bessere Qualifikation als irgendein Fußballer der aktuellen Mannschaft haben) | wenn in der Subquery mindestens ein Tupel enthalten ist, für die der Vergleich erfüllt ist |
| EXIST (Subquery)  select Vorname, Nachname from Kunde where exists (select distinct KundenNr from sales where sales.KundenNr = Kunde.KundenNr); | wenn die Subquery ein nicht leere Menge ergibt  Im Beispiel: wenn es mindestens einen Eintrag zum Kunden in der Sales-Tabelle gibt. |
| UNIQUE (Subquery)  select Vorname, Nachname from Kunde where unique (select KundenNr from sales where sales.KundenNr = Kunde.KundenNr); | wenn die Subquery nur ein einziges Ergebnis zurück liefert.  Im Beispiel: suche die Namen aller Kunden, für die bisher nur ein Verkauf registriert ist |

Quellen:

Kudraß, T.: Taschenbuch Datenbanken, 2015, S.112 ff.

https://www.dummies.com/programming/sql/how-to-use-exists-unique-distinct-and-overlaps-in-sql-statements/