*Tabelle 2: Die Union-Operatoren von SQL*

| Funktion | Beschreibung |
|---|---|
| UNION | Kombiniert die Resultate zweier Abfragen und entfernt die doppelten Zeilen. |
| UNION ALL | Kombiniert die Resultate zweier Abfragen und entfernt die doppelten Zeilen nicht. |
| MINUS | Entfernt alle diejenigen Zeilen einer Abfrage, die auch in einer anderen Abfrage zurückgeliefert werden. |
| INTERSECT | Liefert aus den Resultaten zweier Abfragen nur solche Zeilen, die in beiden vorkommen. |

Das folgende Beispiel einer Union-Abfrage verwendet den MINUS-Operator, um eine Liste aller Tabellen zurückzuliefern, für die noch keine Indizes definiert wurden:

```
SELECT table_name
FROM user_tables
MINUS
SELECT DISTINCT table_name
FROM user_indexes
WHERE table_owner = USER
ORDER BY table_name;
```

Die erste Abfrage liefert eine Liste aller eigenen Tabellen zurück. Die zweite Abfrage liefert eine Liste aller Tabellen, die indiziert sind. Die MINUS-Union-Operation entfernt alle indizierten Tabellen aus der ersten Liste, so daß die nicht-indizierten Tabellen übrigbleiben.

---

HINWEIS

Wenn zwei oder mehr Tabellen mittels Union verrknüpft werden, ist nur eine Klausel ORDER BY erlaubt, die am Ende angegeben werden muß. Nur diejenigen Zeilen werden sortiert, die als Endergebnis zurückgeliefert werden.

---



# Formatierung von Reports

SQL*Plus-Reports sind gewöhnlich spaltenorientiert. SQL*Plus ermöglicht es, Spaltenüberschriften und Anzeigeformate für jede Spalte in einem Report zu definieren. Weiterhin können ebenso Seitenköpfe und Fußzeilen, Seiten- und Zeilenumbrüche sowie Summenberechnungen wie Gesamtsummen und Teilsummen festgelegt werden.

## Spaltenüberschriften

Mittels der Klausel HEADING im Befehl COLUMN können Spaltenüberschriften definiert werden:

```
COLUMN employee_name HEADING "Angestelltenname"
```

Dabei können entweder einfache oder doppelte Anführungszeichen verwendet werden, um den Überschriftentext einzuschließen. Die resultierende Überschrift wird dann wie folgt aussehen:

```
Angestelltenname
-------------
```

Soll eine Überschrift über mehrere Zeilen hinweg angegeben werden, verwendet man den senkrechten Strich ( | ), um die Position des Zeilenumbruchs zu markieren. Zum Beispiel:

```
COLUMN employee_name HEADING "Angestellten-|Name"
```

Die resultierende Mehrzeilen-Überschrift wird wie folgt aussehen:

```
Angestellten-
Name
---------
```

Überschriften von Textspalten sind linksbündig. Überschriften von numerischen Spalten sind hingegen rechtsbündig. Mittels der Klausel JUSTIFY kann dieses Verhalten geändert werden:

```
COLUMN employee_name HEADING "Angestellten-|Name" -
   JUSTIFY RIGHT
COLUMN employee_name HEADING "Angestellten-|Name" -
   JUSTIFY CENTER
```

Mittels SET HEADSEP kann das Zeilenumbruch-Zeichen in ein anderes als den Vertikalstrich geändert werden. Mittels SET UNDERLINE kann das Unterstrich-Zeichen in ein anderes Zeichen als einen Bindestrich geändert werden.

## Spaltenformate

Die Anzeigeformate können mittels der Klausel FORMAT im Befehl COLUMN definiert werden. Formatangaben von numerischen Feldern können sehr detailliert angegeben werden: Angabe der Länge, der Zahl der Dezimalstellen sowie das Format der Dezimaldarstellung. Bei Text- und Datumsfeldern können die Spaltenbreite wie auch der Zeilenumbruch kontrolliert werden. Der spätere Abschnitt »SQL*Plus-Formatelemente« erläutert, wie verschiedene Datentypen formatiert werden können.

## Seitenbreite und -länge

Die Seitenbreite wird durch den Befehl SET LINESIZE angegeben. Standardmäßig beträgt diese Breite 80 Zeichen. Diese kann – auf 60 Zeichen zum Beispiel – durch den folgenden Befehl geändert werden:

```
SET LINESIZE 60
```

Die Einstellung LINESIZE wird von SQL*Plus verwendet, um Seitenüberschriften und Fußzeilen zu zentrieren und rechtsbündig darzustellen.

Die Seitenbreite wird durch die Anweisung SET PAGESIZE kontrolliert. Standardmäßig ist eine Länge von 24 Zeilen pro Seite eingestellt, wobei diese die Seitenüberschrift wie auch die Fußzeilen umfaßt. Der folgende Befehl ändert die Seitenlänge auf 50 Zeilen:

```
SET PAGESIZE 50
```

Das Setzen von PAGESIZE auf Null hat für SQL*Plus eine besondere Bedeutung. Eine PAGESIZE von Null verhindert die Anzeige von Seitenüberschriften, Fußzeilen und Spaltenüberschriften.



## Seitenüberschriften und Fußzeilen

Seitenüberschriften und Fußzeilen können mittels der Befehle TTITLE und BTITLE definiert werden. TTITLE (*top title*) definiert die Seitenüberschrift. BTITLE (*bottom title*) definiert die Fußzeilen. Die Syntax ist für beide identisch.

### Definition einer Überschrift

Das folgende Beispiel definiert eine Mehrzeilen-Seitenüberschrift mit dem Firmennamen auf der linken und der Seitenzahl auf der rechten Seite.

```
TTITLE LEFT "Meine Firma" CENTER "Aktuelle" -
RIGHT "Seite" FORMAT 999 SQL.PNO SKIP 1 -
CENTER "Angestellenliste" SKIP 4
```

Das Ergebnis sieht wie folgt aus:

```
Meine Firma        Aktuelle          Seite   1
                Angestellenliste
```

Die letzte Klausel SKIP erzeugt drei Leerzeilen zwischen dem Seitenkopf und den Spaltenüberschriften. Dieselben Klauseln sind auch für den Befehl BTITLE möglich, um Fußzeilen zu definieren.

### Einfügen des Datums in die Überschrift

Um das aktuelle Datum in die Überschrift einzufügen, muß man:

1. das Datum in eine Benutzervariable kopieren.
2. die Benutzervariable in den Befehl BTITLE oder TTITLE plazieren.

Die folgenden Anweisungen können in einem SQL*Plus-Skript verwendet werden, um das aktuelle Datum in eine Benutzervariable zu kopieren:

```
SET TERMOUT OFF
COLUMN curdate NEW_VALUE report_date
SELECT TO_CHAR(SYSDATE,'dd-Mon-yyyy') curdate
   FROM DUAL;
SET TERMOUT ON
```

Nach Ausführung der gezeigten Befehle steht das Datum in der benutzerdefinierten Variablen REPORT_DATE zur Verfügung. Der folgende Befehl fügt diesen Wert in die Fußzeile ein:

```
BTITLE LEFT "Report-Datum: " report_date
```

Dieselbe Technik kann auch verwendet werden, um andere Werte aus der Datenbank zu erhalten und entweder in die Seitenüberschrift oder in die Fußzeilen einzufügen.

## *Seitenumbrüche*

Standardmäßig fügt SQL*Plus eine Leerzeile zwischen jede ausgegebene Seite ein. Diese Leerzeile, die dem Parameter PAGESIZE hinzugefügt wird, muß der pyhsikalischen Größe der Druckerseite entsprechen.

Der Befehl SET PAGESIZE kann verwendet werden, um die Anzahl der auszudruckenden Seiten in SQL*Plus zu kontrollieren. SET NEWPAGE kontrolliert in SQL*Plus das Verhalten, wenn ein Seitenumbruch stattfindet. Die Anzahl der Leerzeilen zwischen den Seiten kann mittels der folgenden Anweisung gesetzt werden:

```
SET NEWPAGE 10
```

Durch das Setzen von NEWPAGE auf Null wird ein Seitenvorschub-Zeichen in SQL*Plus eingefügt. Zum Beispiel:

```
SET NEWPAGE 0
```

Die neueren Versionen von SQL*Plus erlauben SET NEWPAGE NONE, wodurch sowohl Leerzeilen als auch Seitenvorschub-Zeichen zwischen den Seiten entfernt werden.

## *Report-Umbrüche*

Die Anweisungen BREAK und COMPUTE können dazu verwendet werden, Umbrüche und Gesamtsummen in einem Report zu definieren. BREAK erlaubt ferner, die Darstellung mehrfacher Spaltenwerte zu unterbinden.



### *Die Anweisung BREAK*

Um die Wiederholung von Spaltenwerten zu unterbinden, wird die Anweisung BREAK verwendet, wie das folgende Beispiel zeigt:

```
SQL> BREAK ON owner
SQL> SELECT owner, table_name
  2    FROM all_tables
  3  ORDER BY owner, table_name;

OWNER      TABLE_NAME
========== ===============
CTXSYS     DR$CLASS
           DR$DELETE
           DR$INDEX
DEMO       CUSTOMER
           DEPARTMENT
           EMPLOYEE
```

Wenn eine Spalte in der Anweisung BREAK aufgelistet wird, druckt SQL*Plus den Wert nur dann, wenn dieser sich ändert. Dabei ist es sehr wichtig, daß die Abfrageresultate für dieselbe Spalte sortiert werden.

Die Anweisung BREAK kann weiterhin dazu verwendet werden, Zeilen zu überspringen oder eine Seite zu umbrechen, wenn sich ein Wert ändert. Zum Beispiel:

```
BREAK ON owner SKIP 1
BREAK ON owner SKIP PAGE
```

Der erste Befehl erzeugt immer dann eine Leerzeile, wenn sich der Wert für `owner` ändert. Der zweite erzeugt in diesem Fall eine neue Seite.

Für einen Report können mehrfache Umbrüche definiert werden, doch dies geschieht meist in einem Befehl. Das folgende Beispiel erzeugt einen Seitenumbruch, wenn sich der Wert von `owner` ändert, und eine Leerzeile, wenn sich der Objekttyp ändert:

```
BREAK ON owner SKIP PAGE ON object_type SKIP 1
SELECT owner, object_type, object_name
   FROM dba_objects
ORDER BY owner, object_type, object_name;
```

Beispiel dba_sequences