

Übungsblatt 4 zu Kapitel 4

- 19) Geben Sie die Namen der Mitarbeiter aus, und zwar zusammen mit der Differenz zum Gehalt des Vorgesetzten. Warum werden nicht alle 9 Mitarbeiter ausgegeben?
Hinweis: Es liegt ein Inner-Join vor!
- 20) Geben Sie die Namen aller Mitarbeiter aus, und zwar zusammen mit der Differenz zum Gehalt des Vorgesetzten. Null-Werte sind unbedingt zu vermeiden und bei Bedarf durch die Zahl 0 zu ersetzen.
Hinweis: Es liegt ein Outer-Join vor! Verwenden Sie die Funktion *Coalesce (Wert1, Wert2)*, um Null-Werte zu vermeiden.
- 21) Geben Sie zu jedem Kunden folgende Daten an:
Kundennr, Kundenname, Umsatz. Hat ein Kunde bisher nichts gekauft (gemäß den Relationen *Auftrag* und *Auftragsposten*), so wird beim Umsatz der Wert 0 eingetragen.
Hinweis: Wieder liegt ein Outer-Join vor! Vermeiden Sie unter allen Umständen die Ausgabe des Wertes NULL! Die Spaltenüberschriften heißen so wie oben angegeben!
- 22) Schreiben Sie ein „Miniatur-Programm“, das einen Mitarbeiternamen anfordert. Anschließend erfolgt die Ausgabe aller Daten dieses Mitarbeiters, inkl. der Gehaltsdifferenz zum Vorgesetzten.
Hinweis: Liegt ein Inner- oder Outer-Join vor? Um kleine Programme zusammen mit Variablen schreiben zu können, sind folgende SQL*PLUS-Befehle interessant. Diese Befehle können auch in SQL-Developer verwendet werden.
- | | |
|----------------------------|---|
| REMARK text | Kommentarzeile, der <i>Text</i> wird überlesen |
| PROMPT text | Der angegebene <i>Text</i> (ohne Hochkomma) wird auf Bildschirm ausgegeben |
| PAUSE text | Der <i>Text</i> wird auf Bildschirm ausgegeben, das Programm wartet bis zur Eingabe von Return |
| ACCEPT variable typ | PROMPT 'text' Gibt den <i>Text</i> auf Bildschirm aus, liest die folgende Eingabe in die <i>Variable</i> des angegebenen <i>Typs</i> ein. Als Typ sind für Zahlen <i>NUMBER</i> und für Strings <i>CHAR</i> erlaubt. Beispiel:
Accept Persnr Number Prompt 'Bitte Personalnr eingeben: ' |
- Die mit Accept definierten Variablen können in folgenden SQL-Anweisungen verwendet werden! Zur Unterscheidung von Spaltennamen muss diesen Variablen dort das **Zeichen '&'** vorangestellt werden (z.B. *&Persnr*).
- Hinweis: Verwenden Sie in SQL-Developer ein eigenes Skript, und führen Sie dann das gesamte Skript aus, nicht nur einen Befehl. Verwenden Sie also den zweiten Button (F5)!
- Hinweis: Mit „Set Verify Off“ wird in SQL-Developer das Ersetzen einer Variable durch den Inhalt nicht angezeigt. Einfach mal ausprobieren. Mit „Set Verify On“ wird das Ersetzen wieder angezeigt.
- 23) Wir erweitern Aufgabe 18 dahingehend, dass die Meistverdienenden sortiert werden, beginnend mit dem höchsten Gehalt. Gleichzeitig verwenden wir eine Variable, die angibt, wie viele Mitarbeiter ausgegeben werden sollen.
Hinweis: Wir verwenden die Order-By-Klausel und den Accept-Befehl (siehe Aufgabe 22).
- 24) Geben Sie alle Lieferanten aus, die mindestens alle Artikel liefern wie Lieferant 3.
Hinweis: Diese Aufgabe benötigt den Divisionsoperator. Bilden Sie die Division aus der Relationalen Algebra mit der Ersatzdarstellung der Division nach.
Bitte beachten Sie, dass die bei der Ersatzdarstellung verwendeten Relationen R1 und R2 bereits mit dem Select-Befehl erzeugte Teilrelationen sind.
Unbedingt beachten: Der EXCEPT-Operator heißt in Oracle MINUS!