

LibreOffice
The Document Foundation

Base

Aktuelle Standardwerte für Datum und Zeit

Copyright

Dieses Dokument unterliegt dem Copyright © 2014. Die Beitragenden sind unten aufgeführt. Sie dürfen dieses Dokument unter den Bedingungen der GNU General Public License (<http://www.gnu.org/licenses/gpl.html>), Version 3 oder höher, oder der Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0/>), Version 3.0 oder höher, verändern und/oder weitergeben.

Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.

Fast alle Hardware- und Softwarebezeichnungen und weitere Stichworte und sonstige Angaben, die in diesem Buch verwendet werden, sind als eingetragene Marken geschützt.

Da es nicht möglich ist, in allen Fällen zeitnah zu ermitteln, ob ein Markenschutz besteht, wird das Symbol (R) in diesem Buch nicht verwendet.

Mitwirkende/Autoren

Robert Großkopf

Rückmeldung (Feedback)

Kommentare oder Vorschläge zu diesem Dokument können Sie in deutscher Sprache an die Adresse discuss@de.libreoffice.org senden.

Vorsicht



Alles, was an eine Mailingliste geschickt wird, inklusive der E-Mail-Adresse und anderer persönlicher Daten, die die E-Mail enthält, wird öffentlich archiviert und kann nicht gelöscht werden. Also, schreiben Sie mit Bedacht!

Datum der Veröffentlichung und Softwareversion

Veröffentlicht am 24.06.2020. Basierend auf der LibreOffice Version 6.4.

Inhalt

Einführung	4
Tabellen	4
Datum_Aenderdatum	4
Zeitstempel_Aenderstempel	5
Ansichten	6
Abfragen	6
DatumZeitvorgabe	7
Zeitdifferenzen	7
Zeitdifferenzen_Formularvorlage	8
Formulare	10
Defaultdatum_Subform_neue_Datensaetze	10
Defaultdatum_Zeit_Subform_neue_Datensaetze	11
Defaultdatum_Aenderung	11
Defaultdatum_Makro_SQL	12
Defaultdatum_Zeit_Makro_SQL	12
Defaultdatum_Makro	13
Defaultdatum_Zeit_Makro	14
Defaultdatum_Makro_Standarddatum	14
Defaultdatum_Makro_Standarddatum_verlegt	15
Defaultdatum_Makro_Standardzeitstempel	15
Defaultdatum_Zeitdifferenz_Makro	16
Berichte	16
Bericht_Zeitdifferenzen	17
Bericht_Zeitdifferenzen_Formularvorlage	19
Makros	19
Update	19
Datum_aktuell	20
Datum_Zeit_aktuell	21
Standarddatum	22
Standarddatum_verlegt	22
StandardZeitstempel	23
UpdateTimestamp	23

Einführung

In Formularen gibt es für Formularfelder die Möglichkeit, Standardwerte zu setzen. Damit werden über die grafische Benutzeroberfläche Wertvorgaben für bestimmte Felder gemacht, die abgespeichert werden, wenn sie nicht überschrieben werden.

Auch bei der Erstellung von Tabellen ist in den Feldeigenschaften ein Feld für den «Defaultwert» vorgesehen. Ein Eintrag an dieser Stelle erzeugt ebenfalls einen festen Vorgabewert für die Eingabe in ein Feld – jetzt aber bei Tabellen. Er hat im übrigen nichts mit dem Default-Wert der eigentlichen Datenbank zu tun. Der Default-Wert, definiert in einer Datenbank, wird nur dann geschrieben, wenn ein Feld als leeres Feld abgespeichert werden soll.

Alle Standardwerte der grafischen Benutzeroberfläche haben den Nachteil, dass sie nicht flexibel sind. Es kann nicht das beim Öffnen des Formulars aktuelle Datum oder die aktuelle Zeit eingegeben werden.

Die folgende Datenbank¹ zeigt an Beispielen, wie es möglich ist, ein aktuelles Datum bzw. eine aktuelle Datum-Zeit-Kombination beim Erstellen von Datensätzen zu erzeugen. Es zeigt außerdem, welche Möglichkeiten es gibt, ein Datum hinterher mit möglichst geringem Aufwand ändern zu lassen. Typische Beispiele wären hier eine Zeitmessung in Form von Arbeitsbeginn und Arbeitsende oder ein Vermerk, wann ein Datensatz erstellt und wann er zuletzt geändert worden ist.

Tabellen

Die Tabellen stellen nur einfache Beispieltabellen dar. Sie sollen nur zeigen, wie aktuelle Standardwerte für ein Datum bzw. ein Datum mit Zeitangabe gesetzt werden können.

Datum_Aenderdatum

Datenziel
Formular: Defaultdatum_Aenderung , Defaultdatum_Makro , Defaultdatum_Makro_Standarddatum , Defaultdatum_Makro_Standarddatum_verlegt , Defaultdatum_Subform_neue_Datensaetze
Abfrage: DatumZeitvorgabe
Bericht, Makro: keine direkt

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld kann auch als Auto-Wert-Feld gesetzt werden.
Name	Text	Der Name einer Person oder irgendeine Bezeichnung. Steht hier nur als Beispieleintrag ohne weitere Bedeutung. Eingabe erforderlich: Nein
Datum	Datum	Hier soll ein Startdatum gespeichert werden, das nachher nicht mehr verändert wird. Eingabe erforderlich: Nein

1 Beispieldatenbank Beispiel_Default_Datum_Zeit.odt

AenderDatum	Datum	Hier soll beim Erstellen des Datensatzes das gleiche Datum wie im Feld "Datum" abgespeichert werden. Dieses Feld wird beim erneuten Aufrufen des Datensatzes aber gegebenenfalls überschrieben. Eingabe erforderlich: Nein
-------------	-------	---

Die Tabelle «Datum_SQLDefault_Aenderdatum» ist von den Feldern her völlig gleich aufgebaut.

Datenziel
Formular: Defaultdatum_Makro_SQL
Abfrage, Bericht, Makro: keine direkt

Hier ist im Unterschied zur Tabelle «Datum_Aenderdatum» allerdings über **Extras → SQL** hinterher ein SQL-Defaultwert eingestellt worden:

```
1 ALTER TABLE "Datum_SQLDefault_Aenderdatum"
2 ALTER COLUMN "Datum"
3 SET DEFAULT CURRENT_DATE;
```

```
1 ALTER TABLE "Datum_SQLDefault_Aenderdatum"
2 ALTER COLUMN "AenderDatum"
3 SET DEFAULT CURRENT_DATE;
```

Damit werden in dieser Tabelle beim Abspeichern eines neuen Datensatzes die Felder "Datum" und "AenderDatum" automatisch mit dem aktuellen Datum versehen, sofern nicht ein anderes Datum eingetragen wurde.

```
1 ALTER TABLE "Datum_SQLDefault_Aenderdatum"
2 ADD CHECK (COALESCE("AenderDatum", "Datum")>="Datum");
```

Außerdem wird durch die Check-Bedingung sicher gestellt, dass der Eintrag in "AenderDatum" immer größer oder gleich dem Eintrag in "Datum" ist.

Zeitstempel_Aenderstempel

Datenziel
Formular: Defaultdatum_Makro_Standardzeitstempel , Defaultdatum_Zeit_Makro , Defaultdatum_Zeit_Subform_neue_Datensaetze
Abfrage: Zeitdifferenzen , Zeitdifferenzen_Formularvorlage
Bericht, Makro: keine direkt

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld kann auch als Auto-Wert-Feld gesetzt werden.
Name	Text	Der Name einer Person oder irgendeine Bezeichnung. Steht hier nur als Beispieleintrag, z.B. zur Ermittlung von Arbeitszeit. Eingabe erforderlich: Nein
Zeitstempel	Datum/Zeit	Hier soll eine Startzeit zusammen mit Datum gespeichert werden (TIMESTAMP), die nachher nicht mehr verändert wird. Eingabe erforderlich: Nein

Zeitstempel_A enderung	Datum/Zeit	Hier soll beim Erstellen des Datensatzes die gleiche Zeit mit Datum wie im Feld "Zeitstempel" abgespeichert werden. Dieses Feld wird beim erneuten Aufrufen des Datensatzes aber gegebenenfalls überschrieben. Eingabe erforderlich: Nein
---------------------------	------------	--

Die Tabelle «Zeitstempel_SQLDefault_Aenderstempel» ist von den Feldern her völlig gleich aufgebaut. Hier ist allerdings über **Extras** → **SQL** hinterher ein SQL-Defaultwert eingestellt worden.

Der Tabellename musste für FIREBIRD gekürzt werden, da FIREBIRD für *Tabellennamen und Feldnamen nur maximal 30 Zeichen* zulässt. Dabei sind Standardzeichen gemeint. Sonderzeichen wie Umlaute nehmen in der Regel gleich den Platz für 2 Zeichen ein. In FIREBIRD heißt die Tabelle also «Zeitstempel_SQLD_Aenderstempel».

Datenziel
Formular: Defaultdatum_Zeit_Makro_SQL
Abfrage, Bericht, Makro: keine direkt

```
1 ALTER TABLE "Zeitstempel_SQLDefault_Aenderstempel"
2 ALTER COLUMN "Zeitstempel"
3 SET DEFAULT CURRENT_TIMESTAMP;
```

```
1 ALTER TABLE "Zeitstempel_SQLDefault_Aenderstempel"
2 ALTER COLUMN "Zeitstempel_Aenderung"
3 SET DEFAULT CURRENT_TIMESTAMP;
```

Damit werden in dieser Tabelle beim Abspeichern eines neuen Datensatzes die Felder "Zeitstempel" und "Zeitstempel_Aenderung" automatisch mit der aktuellen Zeit und dem aktuellen Datum versehen, sofern nicht eine andere Zeit und ein anderes Datum eingetragen wurde.

```
1 ALTER TABLE "Zeitstempel_SQLDefault_Aenderstempel"
2 ADD CHECK (COALESCE("Zeitstempel_Aenderung", "Zeitstempel")>="Zeitstempel");
```

Außerdem wird durch die Check-Bedingung sicher gestellt, dass der Eintrag in "Zeitstempel_Aenderung" immer größer oder gleich dem Eintrag in "Zeitstempel" ist.

Ansichten

Diese Datenbank enthält für die HSQLDB keine Ansichten.

In FIREBIRD sind statt der Abfragen [Zeitdifferenzen](#) und [Zeitdifferenzen_Formularvorlage](#) Ansichten notwendig. Die Abfragen sind dort wegen veränderten Codes nur im direkten SQL-Modus ausführbar und damit für die Ausgabe in Berichten nicht sortierbar und auch sonst nur begrenzt brauchbar.

Die Ansichten in FIREBIRD haben die Bezeichnung «Ansicht_Zeitdifferenzen» und «Ansicht_Zeitdifferenzen_Form». Die zweite Benennung musste wegen der Beschränkung auf 30 Zeichen für Tabellennamen gekürzt werden.

Abfragen

Die Datenbank enthält drei Abfragen. Für die Eingabe des Standarddatums ist lediglich die Abfrage «DatumZeitvorgabe» von Bedeutung. Die weiteren Abfragen dienen als Berichtsgrundlage («Zeitdifferenzen») oder als Vorlage für ein Formular, in dem gleichzeitig berechnete Differenzen betrachtet werden sollen («Zeitdifferenzen_Formularvorlage»).

DatumZeitvorgabe

Datenquelle
Tabelle: Datum_Aenderdatum

Datenziel
Formular: Defaultdatum_Subform_neue_Datensaetze , Defaultdatum_Zeit_Subform_neue_Datensaetze

Die Abfrage «DatumZeitvorgabe» dient lediglich dazu, das aktuelle Datum und den aktuellen Zeitstempel von der Datenbank zu erfragen. Damit kann ohne weitere Zuhilfenahme von direkter SQL-Eingabe oder Makros das aktuelle Datum und (etwas begrenzt) auch der Zeitstempel in ein Formular übertragen werden.

```
1 SELECT DISTINCT CURRENT_TIMESTAMP AS "Jetzt",  
2 CURRENT_DATE AS "Heute"  
3 FROM "Datum_Aenderdatum"
```

Für eine Abfrage muss immer eine Tabelle als Basis stehen. Dies ist hier die Tabelle "Datum_Aenderdatum" (Zeile 3). In diesem Fall spielt es keine Rolle, welche Tabelle das ist, da nur Standardwerte aus der Datenbank ermittelt werden sollen: das aktuelle Datum **CURRENT_DATE** und die aktuelle Zeit kombiniert mit dem Datum **CURRENT_TIMESTAMP**. Durch den Zusatz **DISTINCT** (Zeile 1) werden die Werte nur einmal dargestellt. Ansonsten würde die Datenbank so viele Datensätze wiedergeben, wie die Tabelle bereits an Datensätzen hat – mit lauter gleichen Datums- und Zeitstempelangaben.

Zeitdifferenzen

Datenquelle
Tabelle: Zeitstempel_Aenderstempel

Datenziel
Bericht: Bericht_Zeitdifferenzen

Zur Auswertung von Zeiteingaben in einem Bericht wurde die Abfrage «Zeitdifferenzen» erstellt. Da in **FIREBIRD** diese Abfrage nur in etwas abgewandelter Form und in direktem SQL-Modus ausführbar ist (siehe unten) existiert dort stattdessen keine Abfrage sondern eine Ansicht mit der Bezeichnung «Ansicht_Zeitdifferenzen».

```
1 SELECT "ID",  
2 "Name",  
3 "Zeitstempel",  
4 "Zeitstempel_Aenderung",  
5 CAST ( EXTRACT ( YEAR FROM "Zeitstempel" ) || '-' ||  
        RIGHT( '0' || EXTRACT ( MONTH FROM "Zeitstempel" ), 2 ) || '-' ||  
        RIGHT( '0' || EXTRACT ( DAY FROM "Zeitstempel" ), 2 ) AS DATE )  
        AS "StartDatum",  
6 DATEDIFF( 'mi', "Zeitstempel", "Zeitstempel_Aenderung" )  
        AS "Differenz_Minuten"  
7 FROM "Zeitstempel_Aenderstempel"
```

Zuerst wird die komplette Tabelle "Zeitstempel_Aenderstempel" ausgewählt. Das Feld "ID" ist dabei nur notwendig, wenn so eine Abfrage für die Eingabe von Daten benutzt werden soll. Um eine Abfrage editierbar zu halten muss der Primärschlüssel der in der Abfrage enthaltenen Tabellen ebenfalls enthalten sein.

Aus dem Feld "Zeitstempel" ließe sich mit Hilfe der Formatierung auch lediglich das Datum darstellen. In der Abfrage wird zu Demonstrationszwecken gezeigt, wie die Datumsdarstellung auch erreicht werden kann:

Tag, Monat und Jahr können nur über separate Abfragen des Feldes "Zeitstempel" ermittelt werden. Um auch bei einstelligen Tageszahlen und Monatszahlen die vorangestellte '0' zu erhalten werden eine '0' und der ermittelte Tag zusammengefasst (`'0' || DAY("Zeitstempel")`) und dann aus dem entstehenden Text von rechts aus die 2 übrigbleibenden Zeichen ausgelesen. Das Datum wird dann in einem für SQL lesbaren Datumsformat geschrieben: vierstellige Jahreszahl, zweistellige Monatszahl und zweistellige Tageszahl, verbunden mit einem Bindestrich (**YYYY-MM-DD**). Damit ist die Abfrage auch für andere Datumsschreibweisen nutzbar, denn dieser Text wird jetzt in das Datumsformat der Datenbank über `CAST (Datumstext AS DATE)` umgewandelt. Dadurch kann die Ausgabe z.B. im Bericht wieder entsprechend nach lokalen Einstellungen formatiert werden.

Mit `DATEDIFF('mi', "Zeitstempel", "Zeitstempel_Aenderung")` wird die Zeitdifferenz zwischen der Startzeit und der Endzeit in Minuten ermittelt. Diese Zeitdifferenz kann später im Bericht addiert werden und gegebenenfalls auch durch Umrechnung als eine Angabe von Minuten und Stunden erfolgen.

FIREBIRD benötigt hier eine andere Bezeichnung: `DATEDIFF(minute, "Zeitstempel", "Zeitstempel_Aenderung")`. Diese Schreibweise ohne die Maskierung mit einfachen Anführungszeichen versteht die GUI nicht, so dass dann die Abfrage nur noch im direkten SQL-Modus funktioniert.

Zeitdifferenzen_Formularvorlage

Datenquelle
Tabelle: Zeitstempel_Aenderstempel
Datenziel
Formular: Defaultdatum_Zeitdifferenz_Makro
Bericht: Bericht_Zeitdifferenzen_Formularvorlage

Als Formularvorlage ist die wesentlich umfangreichere Abfrage «Zeitdifferenzen_Formularvorlage» gedacht. Da in **FIREBIRD** diese Abfrage nur in etwas abgewandelter Form und in direktem SQL-Modus ausführbar ist (siehe unten) existiert dort stattdessen keine Abfrage sonder eine Ansicht mit der Bezeichnung «Ansicht_Zeitdifferenzen_Form».

```

1 SELECT "ID",
2       "Name",
3       "Zeitstempel",
4       "Zeitstempel_Aenderung",
5       CONVERT ( EXTRACT ( YEAR FROM "Zeitstempel" ) || '-' ||
6               RIGHT( '0' || EXTRACT ( MONTH FROM "Zeitstempel" ), 2 ) || '-' ||
7               RIGHT( '0' || EXTRACT ( DAY FROM "Zeitstempel" ), 2 ) , DATE )
8       AS "StartDatum",
9       ( DATEDIFF( 'mi', "Zeitstempel", "Zeitstempel_Aenderung" ) -
10      MOD( DATEDIFF( 'mi', "Zeitstempel", "Zeitstempel_Aenderung" ) ,
11      60 ) ) / 60 || ':' || RIGHT( '0' || MOD( DATEDIFF( 'mi',
12      "Zeitstempel", "Zeitstempel_Aenderung" ), 60 ), 2 )
13      AS "Differenz_Stunden",
14      ( ( SELECT SUM( DATEDIFF( 'mi', "Zeitstempel",
15      "Zeitstempel_Aenderung" ) ) FROM "Zeitstempel_Aenderstempel"
16      WHERE "Name" = "a"."Name" ) - MOD( ( SELECT SUM( DATEDIFF( 'mi',
17      "Zeitstempel", "Zeitstempel_Aenderung" ) ) FROM
18      "Zeitstempel_Aenderstempel" WHERE "Name" = "a"."Name" ) ,
19      60 ) ) / 60 || ':' || RIGHT( '0' || MOD( ( SELECT SUM( DATEDIFF(

```

```
'mi', "Zeitstempel", "Zeitstempel_Aenderung" ) ) FROM
"Zeitstempel_Aenderstempel" WHERE "Name" = "a"."Name" ), 60 ), 2
) AS "Gesamt_Stunden"
```

```
8 FROM "Zeitstempel_Aenderstempel" AS "a"
```

Bis zur Abfrage des Startdatums ist diese Abfrage gleich der Abfrage «Zeitdifferenzen»: Danach wird beständig die Zeitdifferenz in Minuten genutzt: **DATEDIFF('mi', "Zeitstempel", "Zeitstempel_Aenderung")** (Zeile 6). Die Zahlenangabe kann nicht in ein Zeitformat der Datenbank umgewandelt werden, da in der Datenbank nur Zeiten bis maximal 24 Stunden verwaltet werden. Es wird hier also lediglich in einer Textform die Zeitdarstellung nachgebildet.

Zuerst werden die Minuten durch 60 dividiert, um die Anzahl in Stunden zu ermitteln. Eigentlich müsste es hier genügen, beim Ergebnis die Nachkommastellen abzuschneiden oder auch abzurunden. Dies gelingt leider nicht, da bei den Befehlen TRUNCATE() und CEILING() das Ergebnis immer als Dezimalzahl mit einer Nachkommastelle ausgegeben wird – auch wenn die Nachkommastelle 0 ist. Um ein Ergebnis ohne Nachkommastellen zu erhalten muss von vornherein der verbleibende Rest der Division **MOD (Minuten, 60)** subtrahiert werden. Dann wird bei den anschließenden Division die korrekte Ganzzahl ausgegeben.

Anschließend wird ein Doppelpunkt über **|| ':'** angehängt und an diesen wiederum der Rest aus der Division der Minuten durch 60. Vor die Division wird eine führende '0' gesetzt, damit auf jeden Fall eine zweistellige Zahl für die Minutendarstellung vorhanden ist. Wie beim Datum werden mit **RIGHT (Text, 2)** einfach die zwei Zeichen des Textes übernommen, die am weitesten rechts stehen.

Mit dem entsprechenden Verfahren wird auch die Gesamtzahl an Stunden ermittelt. Da diese pro Person ermittelt werden soll sind hier für die Ermittlung der Stunden und Minuten korrelierende Unterabfragen notwendig. Dazu ist zuerst einmal der Tabelle der Hauptabfrage "Zeitstempel_Aenderung" der **Alias "a"** zugeordnet. Anschließend wird in der Unterabfrage die Summe **SUM** für die Zeitdifferenz in Minuten (**DATEDIFF('mi', "Zeitstempel", "Zeitstempel_Aenderung")**) gebildet – aber nur bezogen auf den jeweiligen Namen, der im aktuellen Datensatz der Hauptabfrage steht: **WHERE "Name" = "a"."Name"**. Die Berechnungen und Umwandlungen der Ergebnisse aus den korrelierenden Unterabfragen sind gleich den Berechnungen und Umwandlungen aus der direkten Ermittlung der Zeitdifferenz in Minuten für den jeweiligen Datensatz. Das Ergebnis wird als "Gesamt_Stunden" ausgegeben.

In **FIREBIRD** sind hier die Zeilen 6 und 7 etwas anders zu formulieren:

```
6 ( ( DATEDIFF(minute, "Zeitstempel", "Zeitstempel_Aenderung" ) - MOD( DATEDIFF(minute,
"Zeitstempel", "Zeitstempel_Aenderung" ), 60 ) ) / 60 ) || ':' || RIGHT( '0' ||
MOD( DATEDIFF( minute, "Zeitstempel", "Zeitstempel_Aenderung" ), 60 ), 2 ) AS
"Differenz_Stunden",
7 ( ( ( SELECT SUM( DATEDIFF( minute, "Zeitstempel", "Zeitstempel_Aenderung" ) ) FROM
"Zeitstempel_Aenderstempel" WHERE "Name" = "a"."Name" ) - MOD( ( SELECT
SUM( DATEDIFF( minute, "Zeitstempel", "Zeitstempel_Aenderung" ) ) FROM
"Zeitstempel_Aenderstempel" WHERE "Name" = "a"."Name" ), 60 ) ) / 60 ) || ':' ||
RIGHT( '0' || MOD( ( SELECT SUM( DATEDIFF(minute, "Zeitstempel",
"Zeitstempel_Aenderung" ) ) FROM "Zeitstempel_Aenderstempel" WHERE "Name" =
"a"."Name" ), 60 ), 2 ) AS "Gesamt_Stunden"
```

Zum einen ist wieder der Befehl **DATEDIFF** durch die Angabe von **minute** ohne die Maskierung mit einfachen Anführungszeichen erforderlich. Dadurch muss die Abfrage als direkte SQL-Abfrage ausgeführt werden.

Zum anderen hat **FIREBIRD** Probleme, Ohne eine entsprechende Klammerung Rechnungen von Strings zu unterscheiden. Deshalb ist in Zeile 6 und Zeile 7 eine schließende Klammer hinter **/ 60)** erforderlich. Natürlich muss entsprechend auch eine öffnende Klammer weiter vorne mit eingesetzt werden. Nur so funktioniert die Rechnung, die innerhalb einer Verbindung mit **||** erfolgt.

Formulare

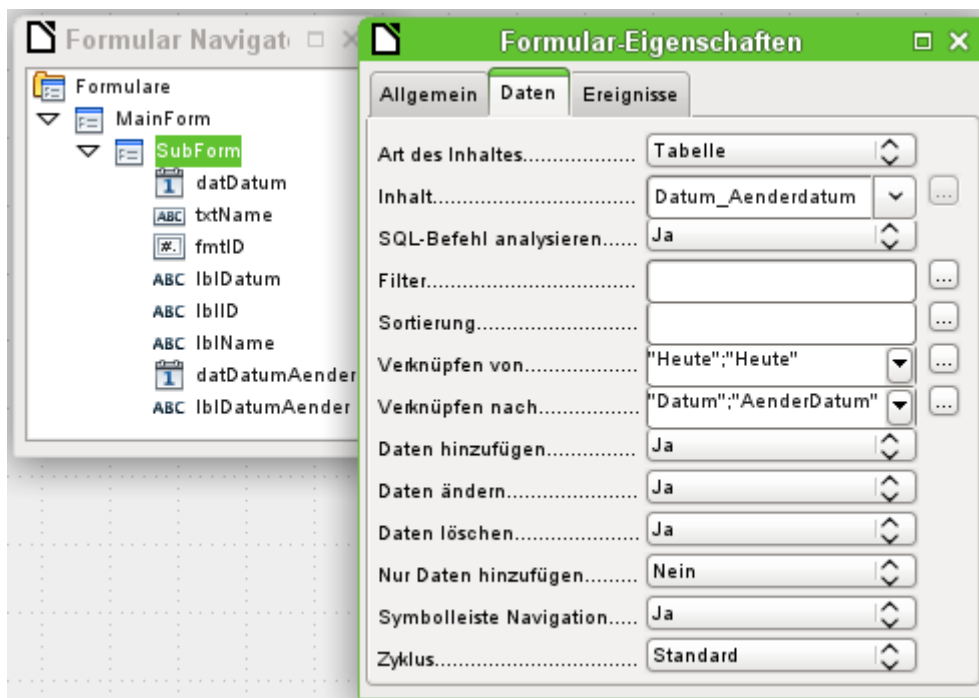
Defaultdatum_Subform_neue_Datensaetze

ID: [] (1)
Name: [] (1.1)
Datum: [19.06.20]
Änderungsdatum: [19.06.20]

1	MainForm (Abfrage: <i>DatumZeitvorgabe</i>)	1.1	SubForm (Tabelle: <i>Datum_Aenderdatum</i>)
---	--	-----	--

Die Formulare «Defaultdatum_Subform_neue_Datensaetze» und «Defaultdatum_Zeit_Subform_neue_Datensaetze» arbeiten nach dem gleichen Prinzip. Wird das Formular geöffnet, so ist bereits das Datum bzw. der Zeitstempel vorgegeben. Im Datumsformular erscheinen die Datensätze des aktuellen Tages und können auch nur Datensätze für den aktuellen Tag eingegeben werden.

Es ist lediglich das Datenfeld für den Primärschlüssel "ID" sowie für das Textfeld "Name" ausfüllbar. Die in Grau gehaltenen Formularfelder sind schreibgeschützt.



Wird das Formular zum Bearbeiten geöffnet und der Formelnavigator aufgerufen, so wird die Verbindung von Hauptformular **MainForm** und Unterformular **SubForm** sichtbar. Das Hauptformular hat als Inhalt die Abfrage «DatumZeitvorgabe». Von dem Hauptformular wird das ermittelte aktuelle Datum an das Unterformular übergeben. Das Feld "Heute" ist mit den Feldern "Datum" und "AenderDatum" des Unterformulars verknüpft.

Im Hauptformular des Datumformulars existieren keine Formularfelder.

Mit Hilfe der Hauptformular-Unterformular-Konstruktion kann auf einfache Weise das aktuelle Datum übertragen und automatisch mit Datensätzen abgespeichert werden. Änderungen von bereits eingegebenen Datumswerten sind nur über die direkte Eingabe per Hand möglich.

Defaultdatum_Zeit_Subform_neue_Datensaetze

The screenshot shows a subform with the following fields and controls:

- ID:
- Name: (The value '1.1' is circled in red)
- Datum und Zeit:
- Änderungsdatum und -zeit:
- Zeitstempel aktualisieren: (The button text '1' is circled in red)

1	MainForm (Abfrage: <i>DatumZeitvorgabe</i>)	1.1	SubForm (Tabelle: <i>Zeitstempel_Aenderstempel</i>)
---	--	-----	--

Im Zeitformular erscheinen auf keinen Fall die vorhergehenden Datensätze, da die Zeit zum Öffnen des Formulars maßgebend ist.

Im Hauptformular des Zeitformulars ist ein Button enthalten. Dieser Button ist in den **Eigenschaften: Schaltfläche → Allgemein → Aktion** auf **Formular aktualisieren** eingestellt. Dadurch werden die Zeiten gegebenenfalls neu eingelesen und übertragen. Würde dies nicht möglich sein, so hätten alle nach dem Öffnen des Formulars erstellten Datensätze den gleichen Zeitstempel.

Bei Zeitangaben muss schon eine laufende Aktualisierung durch einen Button erfolgen. Sie sind in dieser Konstruktion also nur bedingt brauchbar. Änderungen von bereits eingegebenen Zeitstempelwerten sind nur über die direkte Eingabe per Hand möglich.

Defaultdatum_Aenderung

The screenshot shows a subform with the following fields and controls:

- ID:
- Name:
- Datum: (The value '1' is circled in red)
- Änderungsdatum:

1	MainForm (Tabelle: <i>Datum_Aenderdatum</i>)
---	---

Um ein Datum bzw. einen Zeitstempel zu ändern ist ein separates Formular erforderlich, hier das Formular «Defaultdatum_Aenderung». Die Änderung kann nicht in dem vorhergehenden Formular erfolgen, da das Änderungsfeld dort mit dem Hauptformular verbunden ist und bereits eine Datums- bzw. Zeitvorgabe hat.

Defaultdatum_Makro_SQL

ID
1

Name
Robert

Datum
01.05.20

Änderungsdatum
13.05.20

Neuer Datumsstempel

1 MainForm (Tabelle: [Datum_SQLDefault_Aenderdatum](#))

Makros in Feldeigenschaften

1	Button Neuer Datumsstempel (Aktion ausführen)	Module1. Update
---	---	---------------------------------

Die Formulare «Defaultdatum_Makro_SQL» und «Defaultdatum_Zeit_makro_SQL» sind wiederum nach dem gleichen Prinzip aufgebaut.

Bei diesen Formularen erscheint beim Erstellen eines neuen Datensatzes kein Eintrag in den Datums- bzw. Zeitstempel-Feldern. Die Tabelle, die mit dem Formular bearbeitet wird, hat einen Defaultwert über SQL für die Datums- bzw. Zeitstempel-Felder erhalten. Dieser Wert wird beim Abspeichern eines neuen Datensatzes in die Tabelle geschrieben. Der Nutzer des Formulars sieht den Wert also erst, wenn ein Datensatz abgespeichert wurde.

Wenn ein Datensatz geändert werden soll, so kann ein neues Datum bzw. ein neuer Zeitstempel über den Button erstellt werden. Der Button ist hierfür über das **Ereignis → Aktion ausführen** einem Makro verbunden.

Defaultdatum_Zeit_Makro_SQL

ID
1

Name
Duda

Datum und Zeit
13.10.2019 09:49:00

Änderungsdatum und -zeit
10.11.2019 19:39:00

Neuer Zeitstempel

1 MainForm (Tabelle: [Zeitstempel_SQLDefault_Aenderstempel](#))

Makros in Feldeigenschaften

1	Button Neuer Zeitstempel (Aktion ausführen)	Module1. Update
---	---	---------------------------------

In den Zusatzinformationen des Buttons ist der SQL-Code angegeben, der bei einer Änderung ausgeführt werden soll. Außerdem ist hinter einem Semikolon vermerkt, in welchen der Formular-

felder der Wert für den Primärschlüssel steht. Der Primärschlüssel wird für die Änderung nur des einen Datensatzes benötigt. Ohne den Schlüssel würden Änderungen für alle Datensätze gelten.

Eintrag für den Zeitstempel-Button:

```
UPDATE "Zeitstempel_SQLDefault_Aenderstempel" SET  
"Zeitstempel_Aenderung" = CURRENT_TIMESTAMP;fmtID
```

Eintrag für den Datumsstempel-Button:

```
UPDATE "Datum_SQLDefault_Aenderdatum" SET "AenderDatum" =  
CURRENT_DATE;fmtID
```

Der Button kann also so zum einen abspeichern, zum anderen auch direkt den Wert aus der Datenbank wieder auslesen, so dass die Eingabe auf jeden Fall für die Person, die das Formular bedient, erscheint.

Bei FIREBIRD muss im Befehl für den Zeitstempel der Tabellename noch angepasst werden, da die Bezeichnung über 30 Zeichen lang ist.

Defaultdatum_Makro

The screenshot shows a form with the following fields and values:

- ID: 1
- Name: Angie
- Datum: 02.05.20
- Änderungsdatum: 19.06.20

The value '1' in the Datum field is circled in red.

1	MainForm (Tabelle: <i>Datum_Aenderdatum</i>)
---	---

Makros in Formulareigenschaften		
1	Vor der Datensatzaktion	Module1. <i>Datum_aktuell</i>

In dem Formular befinden sich nur zwei beschreibbare Eingabefelder: Das Primärschlüsselfeld und das Namensfeld. Die Werte in den Feldern für das Speicherdatum und das Änderungsdatum bzw. den Speicher-Zeitstempel und den Änderungs-Zeitstempel werden ohne Eingriff des Nutzers automatisch geschrieben. So wird immer die letzte Änderung in dem Änderungsstempel festgehalten.

Defaultdatum_Zeit_Makro

ID	3
Name	Eva
Datum und Zeit	12.10.2019 08:56:00
Änderungsdatum und -zeit	12.10.2019 15:55:00

1 MainForm (Tabelle: [Zeitstempel_Aenderstempel](#))

Makros in Formulareigenschaften		
1	Vor der Datensatzaktion	Module1. Datum_Zeit_aktuell

Dieses Formular setzt immer automatisch über das entsprechende Makro den Zeitstempel ein. Direkt beim ersten Abspeichern sind so die beiden Felder mit gleichen Zeitstempeln versehen. Beim erneuten Abspeichern, z.B. nach einer Datensatzänderung, wird dann nur das Feld für «Änderungsdatum und -zeit» mit einem neuen Wert versehen.

Defaultdatum_Makro_Standarddatum

ID	
Name	
Datum	19.06.20

1 MainForm (Tabelle: [Datum_Aenderdatum](#))

Makros in Formulareigenschaften		
1	Nach dem Datensatzwechsel	Module1. Standarddatum

Über die Eigenschaften eines Kontrollfeldes lassen sich Standardwerte festlegen. Beim Datumsfeld ist dies mit **Eigenschaften: Datumsfeld → Allgemein → Standarddatum** zu erreichen. Diese Einstellungen sind dann dauerhaft in dem Formular gespeichert. Ein aktuelles Datum oder ein aktueller Zeitstempel können in der grafischen Benutzeroberfläche nicht vorgegeben werden.

Die Standardwerte erscheinen, wie in dem Formular zu sehen, bereits beim Erstellen eines neuen Datensatzes. Sie werden für das Formular allerdings nicht als Datensatzänderung verzeichnet. Erst wenn ein anderes Feld oder eben das Feld mit dem Standardwert geändert wurde nimmt die grafische Benutzeroberfläche das als Änderung wahr und speichert den Datensatz auf die übliche Art und Weise ab (Button »Speichern«, Navigation zum nächsten Datensatz, Navigation zum Unterformular u.ä.).

Bei diesem Formular ist zu Bedenken, dass der Wert beim Erreichen eines neuen Datensatzes bereits im Formular erscheint. Bei Datumswerten ist das allerdings weniger von Bedeutung, es sei denn, die Bearbeitung eines Datensatzes beginnt z.B. um 23:59 Uhr und endet um 00:01 Uhr des Folgetages.

Defaultdatum_Makro_Standarddatum_verlegt

The screenshot shows a form with three fields: 'ID' (empty), 'Name' (containing '1' and circled in red), and 'Datum' (containing '19.04.20').

1	MainForm (Tabelle: Datum_Aenderdatum)
---	--

Makros in Formulareigenschaften		
1	Nach dem Datensatzwechsel	Module1. Standarddatum_verlegt

Beim Datumsfeld legt der Standardwert gleichzeitig fest, mit welcher Monatsansicht die Aufklappfunktion des Datumsfeldes startet. Werden z.B. häufig Datumswerte eingegeben, die im Vormonat liegen, so kann über ein Standarddatum aus dem Vormonat die Auswahl entsprechend eingestellt werden.

Defaultdatum_Makro_Standardzeitstempel

The screenshot shows a form with three fields: 'ID' (containing '3'), 'Name' (containing 'Eva' and circled in red), and 'Zeitstempel' (containing '12.10.2019 08:56:00').

1	MainForm (Tabelle: Zeitstempel_Aenderstempel)
---	--

Makros in Formulareigenschaften		
1	Nach dem Datensatzwechsel	Module1. StandardZeitstempel

Über die Eigenschaften eines Kontrollfeldes lassen sich Standardwerte festlegen. Hier beim formatierten Feld für den Zeitstempel über **Eigenschaften: Formatiertes Feld → Allgemein → Standardwert**.

Bei diesem Formular ist zu Bedenken, dass der Wert beim Erstellen eines neuen Datensatzes bereits im Formular erscheint. Es handelt sich also bei der Zeit nicht um den Zeitpunkt, zu dem die Abspeicherung des Datensatzes erfolgte.

Defaultdatum_Zeitdifferenz_Makro

1	HSQDB: MainForm (Abfrage: Zeitdifferenzen_Formularvorlage)	
1	FIREBIRD: MainForm (Tabelle: Zeitstempel_Aenderstempel)	1.1 FIREBIRD: SubForm (Ansicht: Ansicht_Zeitdifferenzen_Form)

Makros in Formulareigenschaften		
1	Vor der Datensatzaktion	Module1.Datum_Zeit_aktuell
Makros in Feldeigenschaften		
1	Button Neue Endzeit (Aktion ausführen)	Module1.UpdateTimestamp

Dieses Formular zeigt neben dem Zeitstempel für die erste Abspeicherung und dem Zeitstempel für die Änderung des Datensatzes die Differenz zwischen den Zeiten sowie die Differenz zwischen allen Zeiten, die mit der Person verbunden sind, an.

Wird auf den Button «Neue Endzeit» gedrückt, so wird das Änderungsdatum und die Änderungszeit neu gesetzt und die Zeitdifferenzen werden neu eingelesen.

Berichte

Für diese Beispieldatenbank wurden zwei Berichte erstellt. Beide greifen auf die gleiche Tabelle «Zeitstempel_Aenderstempel» zu – allerdings über verschiedene Abfragen. Bei dem ersten Bericht ist vor allem mit Formeln innerhalb des Berichtes gearbeitet worden. Beim zweiten Bericht ist das Ergebnis nahezu identisch – nur sind die Inhalte komplett aus der Abfrage übernommen worden, so dass innerhalb des Berichtes keine Rechnung mehr ausgeführt wird.

Bericht_Zeitdifferenzen

Name: Adam

StartDatum	Zeitstempel	Zeitstempel Änderung	Differenz in Minuten
11.10.19	11.10.19 09:08	11.10.19 15:27	379
12.10.19	12.10.19 00:03	13.10.19 01:02	1499
18.10.19	18.10.19 21:15	18.10.19 21:19	4

Zeitsumme in Minuten: 1882

Zeitsumme in Stunden: 31:22

Name: Eva

StartDatum	Zeitstempel	Zeitstempel Änderung	Differenz in Minuten
11.10.19	11.10.19 09:12	11.10.19 15:07	355
12.10.19	12.10.19 08:56	12.10.19 15:55	419
13.10.19	13.10.19 08:31	14.10.19 09:30	1499
18.10.19	18.10.19 21:21	18.10.19 21:23	2
18.10.19	18.10.19 20:56	18.10.19 21:19	23
18.10.19	18.10.19 20:42	18.10.19 20:57	15
19.10.19	19.10.19 09:26	19.10.19 09:50	24

Zeitsumme in Minuten: 2337

Zeitsumme in Stunden: 38:57

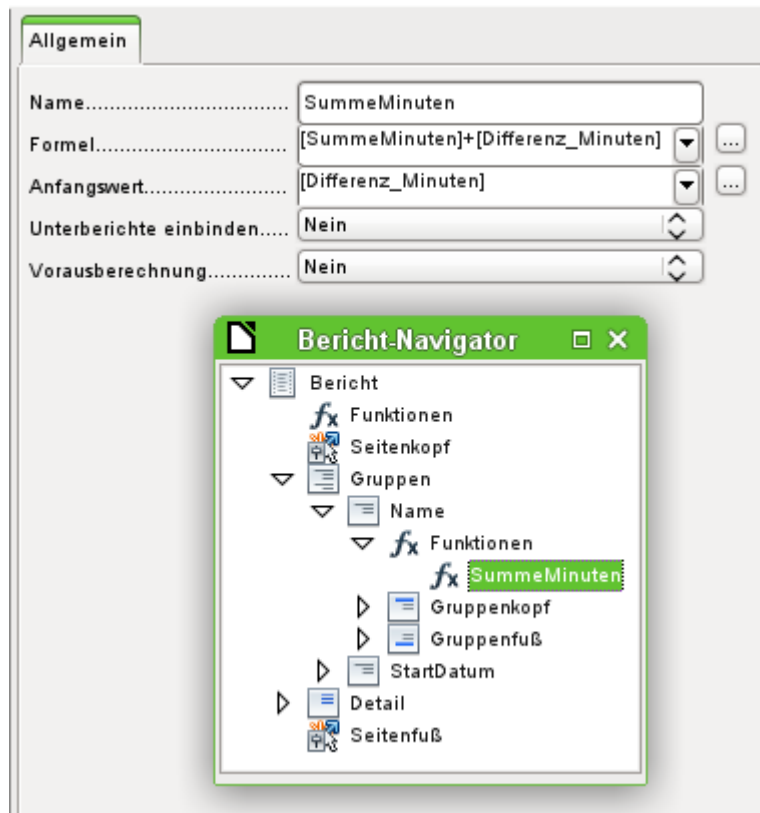
Datenquelle

Abfrage: [Zeitdifferenzen](#) FIREBIRD: [Ansicht_Zeitdifferenzen](#)

Der Bericht zeigt, aufgelistet nach Personen, die Zeitdifferenzen für jedes einzelne Startdatum in der Maßeinheit Minuten. Außerdem wird die Summe der Zeitdifferenzen in Minuten und schließlich auch in Stunden angegeben. So könnte z.B. eine Zusammenstellung über Arbeitszeiten funktionieren.

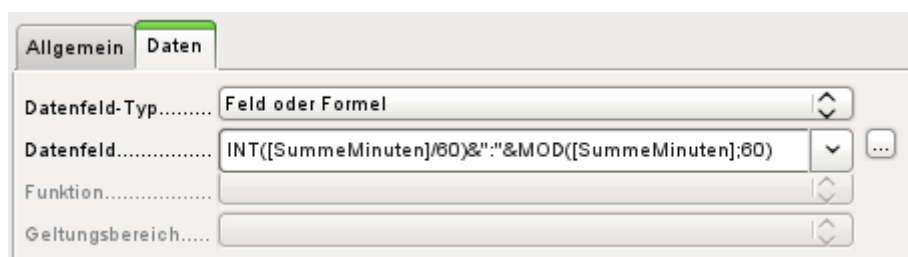
The screenshot shows a report viewer interface. At the top, there is a navigation bar with page numbers 1 through 17. Below this is a search bar and a sidebar with expandable sections. The main content area displays a table with the following columns: StartDatum, Zeitstempel, Zeitstempel Änderung, and Differenz in Minuten. The table content is partially visible, showing data for Adam and Eva, and summary rows for 'Zeitsumme in Minuten' and 'Zeitsumme in Stunden'. The interface includes a search bar and a sidebar with expandable sections for Name, StartDat., and Detail. The table content is partially visible, showing data for Adam and Eva, and summary rows for 'Zeitsumme in Minuten' and 'Zeitsumme in Stunden'.

Gruppierungen und Sortierungen greifen beim Report-Builder ineinander. Der Bericht ist zuerst gruppiert nach dem Namen, zu dem es auch einen Gruppenfuß gibt. Dann erfolgt eine weitere Gruppierung nach dem Startdatum. Diese Gruppierung steht zwar in der Übersicht, wird aber im Bericht nicht gezeigt. Dadurch wird lediglich nach dem Startdatum sortiert. Seitenkopf und Seitenfuß sind hier ausgeblendet und stehen in den Eigenschaften auf **Sichtbar → Nein**.



Im Gruppenfuß befindet sich als erstes das Feld für die Summierung der Summe. Die Funktion wurde händisch erstellt, da die Automatik von LO leider versagte. Über den Berichtsnavigator ist so eine Funktion anschließend erreichbar und kann dann auch noch bearbeitet werden.

[Differenz_Minuten] ist der Feldwert aus der Abfrage «Zeitdifferenzen». Der erste Wert wird gelesen, als Wert der Formel gespeichert und anschließend wird der nächste Wert der Abfrage einfach hinzu addiert. Auf die Formel kann anschließend wie auf das Feld der Abfrage mit eckigen Klammern zugegriffen werden: **[SummeMinuten]**.



Für die Zeitsumme in Stunden wird bei den Daten des Feldes eine Formel eingetragen. Sie liest zuerst den Wert der Funktion **[SummeMinuten]** aus und teilt diesen durch 60. Mit der Funktion **INT** werden daraus nur die Ganzzahlbestandteile übernommen. Anschließend wird mit **&":"&** die Anzahl der verbleibenden Minuten angehängt. **MOD([SummeMinuten];60)** stellt den «Rest» der Division der Minutenanzahl durch 60 dar.

Bericht_Zeitdifferenzen_Formularvorlage

Name: Adam

StartDatum	Zeitstempel	Zeitstempel Änderung	Differenz in Stunden
11.10.19	11.10.19 09:08	11.10.19 15:27	6:19
12.10.19	12.10.19 00:03	13.10.19 01:02	24:59
18.10.19	18.10.19 21:15	18.10.19 21:19	0:04

Zeitsumme in Stunden: 31:22

Name: Eva

StartDatum	Zeitstempel	Zeitstempel Änderung	Differenz in Stunden
11.10.19	11.10.19 09:12	11.10.19 15:07	5:55
12.10.19	12.10.19 08:56	12.10.19 15:55	6:59
13.10.19	13.10.19 08:31	14.10.19 09:30	24:59
18.10.19	18.10.19 21:21	18.10.19 21:23	0:02
18.10.19	18.10.19 20:56	18.10.19 21:19	0:23
18.10.19	18.10.19 20:42	18.10.19 20:57	0:15
19.10.19	19.10.19 09:26	19.10.19 09:50	0:24

Zeitsumme in Stunden: 38:57

Datenquelle

Abfrage: [Zeitdifferenzen_Formularvorlage](#) FIREBIRD: [Ansicht_Zeitdifferenzen_Form](#)

Dieser Bericht ist äußerlich gleich dem vorhergehenden aufgebaut. Nur ist hier die Differenz zwischen den Zeiten direkt in Stunden ausgedrückt.

Dem Bericht liegt statt der Abfrage «Zeitdifferenzen» die Abfrage «Zeitdifferenzen_Formularvorlage» zu Grunde. In dieser Abfrage werden bereits die Zeiten berechnet, auch die Gesamtzeiten gruppiert nach dem Namen. Dadurch ist für den Bericht keine Funktion und keine Formel notwendig.

Makros

Update

Aufruf aus

Formular: [Defaultdatum_Makro_SQL](#), [Defaultdatum_Zeit_Makro_SQL](#)

```
1 SUB Update(oEvent AS OBJECT)
2   DIM oForm AS OBJECT
3   DIM oFeld AS OBJECT
4   DIM oFeldID AS OBJECT
5   DIM stTag AS STRING
6   DIM arList()
7   DIM loRow AS LONG
8   oFeld = oEvent.Source.Model
9   oForm = oFeld.Parent
10  loRow = oForm.getRow
```

```

11  stTag = oFeld.Tag
12  arList = Split(stTag, ";")
13  oFeldID = oForm.getByName(Trim(arList(1)))
14  IF NOT IsEmpty(oFeldID.currentValue) THEN
15      oDatenquelle = ThisComponent.Parent.CurrentController
16      If NOT (oDatenquelle.isConnected()) THEN
17          oDatenquelle.connect()
18      END IF
19      oVerbindung = oDatenquelle.ActiveConnection()
20      oSQL_Anweisung = oVerbindung.createStatement()
21      stSql = arList(0) + " WHERE ""ID"" = " + oFeldID.CurrentValue
22      oSQL_Anweisung.executeUpdate(stSql)
23      oForm.reload
24      oForm.absolute(loRow)
25  END IF
26 END SUB

```

Über den Ursprung des Auslösers wird der Button ermittelt: **oEvent.Source.Model**. Das Formular ist **Parent** zu dem Button (Zeile 9). In Zeile 10 wird die aktuelle Datensatznummer ausgelesen. Die Zusatzinformationen werden aus **oFeld.Tag** ausgelesen und in einem Array in die beiden Teile aufgeteilt (Zeile 11 und 12). Im zweiten Teil steht der Name des Feldes, in dem der Primärschlüssel steht. Mit **Trim(arList(1))** werden eventuelle Leerzeichen vor und hinter dem Eintrag entfernt (Zeile 13).

Ist das Primärschlüsselfeld leer, so wurde der Datensatz noch nicht abgespeichert. Es gibt also auch keine Änderung über den Button zu speichern. Nur wenn in dem Feld ein Wert steht, dann wird ein Kontakt zur Datenbank aufgebaut. Da aber das Formular bereits geöffnet ist müsste dieser Kontakt sowieso bestehen. Die entsprechende Nachfrage ist also nur eine Absicherung (Zeile 15 bis 18).

Schließlich wird der SQL-Code aus den Informationen in den Zusatzinformationen und den zusätzlichen Informationen zu dem Wert in dem Primärschlüsselfeld zusammengestellt (Zeile 21). Der SQL-Code wird an die Datenbank weitergegeben. Anschließend wird das Formular neu geladen und das Ergebnis über **oForm.absolute(loRow)** direkt sichtbar im Formular dargestellt.

Datum_aktuell

Aufruf aus

Formular: *Defaultdatum_Makro*

```

1  SUB Datum_aktuell
2  DIM oDoc AS OBJECT
3  DIM oDrawpage AS OBJECT
4  DIM oForm AS OBJECT
5  DIM oFeld AS OBJECT
6  DIM oFeldAender AS OBJECT
7  DIM unoDate
8  oDoc = thisComponent
9  oDrawpage = oDoc.drawpage
10 oForm = oDrawpage.forms.getByName("MainForm")
11 oFeld = oForm.getByName("datDatum")
12 oFeldAender = oForm.getByName("datDatumAender")
13 unoDate = createUnoStruct("com.sun.star.util.Date")
14 unoDate.Year = Year(Date)
15 unoDate.Month = Month(Date)
16 unoDate.Day = Day(Date)
17 IF isEmpty(oFeld.Date) THEN
18     oFeld.BoundField.updateDate(unoDate)
19 END IF
20 oFeldAender.BoundField.updateDate(unoDate)
21 END SUB

```

Der Zugriff auf die Felder im Formular wird über das Formulardokument erstellt (Zeile 8 bis 12). Anschließend wird das Datum als **Struct** zusammgebaut. In dieser Prozedur wird eine Methode gezeigt, wie die einzelnen Werte den verschiedenen Teilen des **Structs** für ein Datum zugeordnet werden. Ist das Feld mit dem Namen «datDatum» leer (Zeile 17), dann wird das aktuelle Datum über **oFeld.BoundField.updateDate()** in das Feld geschrieben. Ansonsten wird das aktuelle Datum nur in das Feld von «datDatumAender» geschrieben. Die endgültige Abspeicherung erfolgt über das Formular. Schließlich wurde das Makro nur direkt vor der Ausführung einer Datensatzaktion eingesetzt, und diese Aktion bedeutet in diesem Falle ein Schreiben oder Ändern, gegebenenfalls auch löschen eines Datensatzes.

Datum_Zeit_aktuell

Aufruf aus
Formular: <i>Defaultdatum_Zeit_Makro, Defaultdatum_Zeitdifferenz_Makro</i>
Makro: <i>UpdateTimestamp</i>

```

1 SUB Datum_Zeit_aktuell
2   DIM oDoc AS OBJECT
3   DIM oDrawpage AS OBJECT
4   DIM oForm AS OBJECT
5   DIM oFeld AS OBJECT
6   DIM oFeldAender AS OBJECT
7   DIM unoStmp
8   oDoc = thisComponent
9   oDrawpage = oDoc.drawpage
10  oForm = oDrawpage.forms.getByName("MainForm")
11  oFeld = oForm.getByName("fmtDatumZeit")
12  oFeldAender = oForm.getByName("fmtDatumZeitAender")
13  unoStmp = createUnoStruct("com.sun.star.util.DateTime")
14  WITH unoStmp
15    .Year = Year(Date)
16    .Month = Month(Date)
17    .Day = Day(Date)
18    .Hours = Hour(Time)
19    .Minutes = Minute(Time)
20    .Seconds = Second(Time)
21    .NanoSeconds = 000000
22  END WITH
23  IF isEmpty(oFeld.CurrentValue) THEN
24    oFeld.BoundField.updateTimestamp(unoStmp)
25  END IF
26  oFeldAender.BoundField.updateTimestamp(unoStmp)
27 END SUB

```

Das Makro für den Zeitstempel läuft vom Prinzip her gleich ab wie das Makro für das aktuelle Datum «Datum_aktuell». Hier wird lediglich ein anderes **Struct**, nämlich das **DateTime-Struct**, befüllt und der entsprechende Wert abgespeichert. Mit **WITH unoStmp** beginnt die Zusammensetzung des Structes (Zeile 14). **WITH** bedeutet hier nur, dass im Prinzip vor jedes der folgenden Elemente unoStmp gesetzt wird, bis eben **END WITH** auftaucht. Also **unoStmp.Year**, **unoStmp.Month** usw.

In dem Struct ist es auch möglich, Nanosekunden weiter zu geben. Die spielen hier aber keine Rolle. Das Struct wird also an dieser Position lediglich mit der entsprechenden Anzahl Nullen aufgefüllt.

Schließlich wird der erstellte Timestamp nur dann in das Feld «fmtDatumZeit» geschrieben, wenn das Feld leer ist. Ansonsten überschreibt der Timestamp nur den alten Timestamp im Feld «fmtDatumZeitAender».

Standarddatum

Aufruf aus

Formular: *Defaultdatum_Makro_Standarddatum*

```
1 SUB Standarddatum
2   DIM oDoc AS OBJECT
3   DIM oDrawpage AS OBJECT
4   DIM oForm AS OBJECT
5   DIM oFeld AS OBJECT
6   DIM unoDate
7   oDoc = thisComponent
8   oDrawpage = oDoc.drawpage
9   oForm = oDrawpage.forms.getByName("MainForm")
10  oFeld = oForm.getByName("datDatum")
11  unoDate = createUnoStruct("com.sun.star.util.Date")
12  unoDate.Year = Year(Date)
13  unoDate.Month = Month(Date)
14  unoDate.Day = Day(Date)
15  oFeld.setPropertyValue("DefaultDate", unoDate)
16 END SUB
```

Das Makro läuft ähnlich ab wie das Makro «Datum_aktuell». Nachdem allerdings das Struct mit dem aktuellen Datum befüllt ist wird jetzt das Standarddatum das Datumsfeldes damit gesetzt. Dies geschieht über den allgemeinen Zugriff auf bestimmte Eigenschaften, nämlich über **setPropertyValue**. Die Eigenschaft heißt beim Datumsfeld «DefaultDate». Der Wert, der hier gespeichert wird, muss vom Typ dem Struct entsprechen (Zeile 15).

Standarddatum_verlegt

Aufruf aus

Formular: *Defaultdatum_Makro_Standarddatum_verlegt*

```
1 SUB Standarddatum_verlegt
2   DIM oDoc AS OBJECT
3   DIM oDrawpage AS OBJECT
4   DIM oForm AS OBJECT
5   DIM oFeld AS OBJECT
6   DIM unoDate
7   DIM daNew AS DATE
8   DIM i AS INTEGER
9   oDoc = thisComponent
10  oDrawpage = oDoc.drawpage
11  oForm = oDrawpage.forms.getByName("MainForm")
12  oFeld = oForm.getByName("datDatum")
13  unoDate = createUnoStruct("com.sun.star.util.Date")
14  i = -2
15  daNew = DateAdd("m", i, Date)
16  unoDate.Year = Year(daNew)
17  unoDate.Month = Month(daNew)
18  unoDate.Day = Day(daNew)
19  oFeld.setPropertyValue("DefaultDate", unoDate)
20 END SUB
```

Die Verlegung des Datums kam als Anfrage in einem Forum. Dort wurden häufig Daten bearbeitet, die die Eingabe eines Datums aus dem Vormonat erforderten. Dann musste in dem aufklappbaren Feld immer erst der Monat zurückgesetzt werden um dann den entsprechenden Datumswert zu setzen.

Wird das Defaultdatum stattdessen direkt auf den Vormonat festgelegt, so kann das Zurückbewegen zum Vormonat entfallen.

Das Makro hat erst einmal den gleichen Aufbau wie «Standarddatum». Für die Umrechnung des Datums wird die Funktion **DateAdd** genutzt. In diesem Fall der in Zeile 14 hinterlegte Wert mittels dieser Funktions zum Monatswert des aktuellen Datums addiert (Zeile 15). Die Funktion macht dann daraus einen gültigen Datumswert in der Zukunft (bei positiven Werte für **i**) oder in der Vergangenheit (bei negativen Werten für **i**).

Schließlich wird der berechnete Wert wie beim Makro «Standarddatum» als vorgegebener Standardwert festgelegt.

StandardZeitstempel

Aufruf aus

Formular: *Defaultdatum_Makro_Standardzeitstempel*

```

1 SUB StandardZeitstempel
2   DIM oDoc AS OBJECT
3   DIM oDrawpage AS OBJECT
4   DIM oForm AS OBJECT
5   DIM oFeld AS OBJECT
6   DIM stStmp AS STRING
7   oDoc = thisComponent
8   oDrawpage = oDoc.drawpage
9   oForm = oDrawpage.forms.getByname("MainForm")
10  oFeld = oForm.getByname("fmtZeitstempel")
11  stStmp = Year(Date) & "-" & Month(Date) & "-" & Day(Date) & " " & Hour(Time) &
        ":" & Minute(Time) & ":" & Second(Time)
12  oFeld.setPropertyValue("EffectiveDefault", stStmp)
13 END SUB

```

Beim Zeitstempel muss der Wert anders festgelegt werden. Die Formularassistenten bilden häufig den Zeitstempel in zwei getrennten Feldern ab: zum einen ein Datumfeld, zum anderen ein Zeitfeld. Hier wurde stattdessen ein formatiertes Feld gewählt. Dort ist der differenzierte Aufbau von Datum und Zeit zusammen nicht als Zeitstempelfeld in den Standardwert übertragbar. Stattdessen wird für den Zeitstempel ein Text mit dem entsprechenden Format erstellt (Zeile 11). Der Aufbau von stStmp ist hier mit der für Datenbanken üblichen Datumsschreibweise versehen: Zuerst die vierstellige Jahreszahl, gefolgt von '-', dann die zweistellige Monatszahl und wieder '-' und schließlich die zweistellige Tageszahl. Zeit und Datum werden durch eine Leertaste getrennt. Als Verbindungselement der Variablen muss hier das «&» gewählt werden. Wird stattdessen, wie sonst bei Texten gewohnt, ein «+» gewählt, dann werden stattdessen die verschiedenen Werte einfach addiert.

Der Standardwert wird hier über **EffectiveDefault** eingefügt (Zeile 12). Zu den entsprechenden Standardwerten siehe unter anderen das Base-Handbuch.

UpdateTimestamp

Aufruf aus

Formular: *Defaultdatum_Zeitdifferenz_Makro*

Benötigt

Makro: *Datum_Zeit_aktuell*

```

1 SUB UpdateTimestamp(oEvent AS OBJECT)
2   DIM oForm AS OBJECT
3   DIM oFeld AS OBJECT
4   DIM loRow AS LONG
5   Datum_Zeit_aktuell

```

```

6     oFeld = oEvent.Source.Model
7     oForm = oFeld.Parent
8     loRow = oForm.getRow
9     IF oForm.IsNew THEN
10        oForm.insertRow
11    ELSE
12        oForm.updateRow
13    END IF
14    oForm.reload
15    oForm.absolute(loRow)
16 END SUB

```

Das Makro «UpdateTimestamp» greift direkt auf das Objekt zu, durch das es ausgelöst wurde: **oEvent AS OBJECT**. Dadurch wird im Folgenden das Formular ermittelt, das mit einer (neuen) Endzeit versehen wird (Zeile 6 und 7). Existiert noch keine Zeiteingabe, so wird die Zeit neu geschrieben. Das Makro «Datum_Zeit_aktuell» wird dafür aus diesem Makro heraus zuerst einmal gestartet (Zeile 5).

Mit **oForm.IsNew** wird abgesichert, ob ein neuer Datensatz eingefügt (**insertRow**) oder ein bestehender Datensatz geändert (**updateRow**) werden muss. Anschließend wird der aktuell sichtbare Datensatz neu in das Formular eingelesen. Dazu wurde in Zeile 8 die aktuelle Datensatznummer ausgelesen. In Zeile 15 wird dann nach einem Neuladen des Formulars diese Datensatznummer Damit wird entsprechend auch die Berechnung sichtbar durchgeführt.