

Projekt Bonusclub Documentation

Views

Trigger

Calculate subtotal

Delete Customer

(Before delete müssen Positionen und Rechnungen gelöscht werden.) oder in Invoices updaten zu einer anderen Zahl

Update sold units

Packages

Customers

P: Update a customer

```
procedure sp_update_customer( n_customer_id_in IN NUMBER,
                             v_name_in IN VARCHAR,
                             v_surname_in IN VARCHAR,
                             v_phone_number IN VARCHAR,
                             d_date_of_birth_in DATE,
                             v_email_in VARCHAR,
                             n_card_no_in Number)
```

F: Turnover

P: Delete (DSGVO)

Nur als Prozedur funktioniert nur wenn der Trigger programmiert worden ist.

Articles

P: Add

P: Update

P: List

pa_bonus_branch

Package name: `pa_bonus_branch`

Entwickler: Manuel Szecsenyi

Beschreibung: Ermöglicht das Aufrufen von Filialdaten.

f_get_branches

Gibt alle Filialen zurück.

```
function f_get_branches RETURN SYS_REFCURSOR
```

Parameter

Diese Funktion nimmt keine Parameter entgegen.

Rückgabewert

`SYS_REFCURSOR` - Alle abgespeicherten Filialen.

f_get_branch

Gibt eine die Filiale zurück mit der angegebenen ID.

```
function f_get_branch(n_branch_id_in IN NUMBER) RETURN SYS_REFCURSOR
```

Parameter

`n_branch_id_in NUMBER` - Eine Filialid

Rückgabewert

`SYS_REFCURSOR` - Die Filiale mit der angegebenen ID.

Invoices

P: Add Invoice

```
PROCEDURE sp_add_invoice(  
    n_customer_id_in IN NUMBER,  
    n_branch_office_id_in IN NUMBER,  
    n_invoice_id_out OUT NUMBER  
)
```

Input

`n_customer_id_in` - Der Kunde zu dem die Rechnung gehört.

`n_branch_office_id_in` - Die Filial-ID in der der Einkauf getätigt wurde.

Output

`n_invoice_id_out` - Die Rechnungs-ID der neu angelegten Rechnung.

Description

Speichert eine neue leere Rechnung zu einem Kunden und einer Filiale ab.

P: Update Invoice

```
PROCEDURE sp_update_invoice(  
    n_invoice_id_in IN NUMBER,  
    n_customer_id_in IN NUMBER,  
    d_purchase_timestamp_in IN TIMESTAMP,  
    n_branch_office_id_in IN NUMBER  
)
```

Input

`n_invoice_id_in` - Die Rechnungs-ID der Rechnung welche geändert werden soll. Die ID kann nicht verändert werden.

`n_customer_id_in` - Der Kunde zu dem die Rechnung gehört.

`n_purchase_timestamp_in` - Der Timestamp des Einkaufs.

`n_branch_office_id_in` - Die Filial-ID in der der Einkauf getätigt wurde.

Output

Description

Aktualisiert die wichtigsten Metainformationen einer Rechnung. Die Rechnungs-ID wird nicht verändert.

P: Add Article

```
PROCEDURE sp_add_article(  
    n_invoice_id_in IN NUMBER,  
    n_article_id_in IN NUMBER,  
    n_quantity_in IN NUMBER  
)
```

Input

`n_invoice_id_in` - Die Rechnungs-ID der Rechnung.

`n_article_id_in` - Die Artikel-ID eines Artikels

`n_quantity_in` - Wie viel Stück des Produkte soll verrechnet werden?

Output

Description

Fügt einen Artikel mit der angegebenen Stückzahl zu einer Rechnung hinzu. Es wird der im Artikel hinterlegte Preis verrechnet.

P: Delete Article

```
PROCEDURE sp_delete_article(  
    n_invoice_id_in IN NUMBER,  
    n_article_id_in IN NUMBER  
)
```

Input

`n_invoice_id_in` - Die Rechnungs-ID der Rechnung.

`n_article_id_in` - Die Artikel-ID eines Artikels

Output

Description

Der angegebene Artikel auf der Rechnung wird entfernt. Die Rechnungspositionen werden nachgerückt.

F: Get all invoices

```
function f_get_invoices_rc RETURN SYS_REFCURSOR
```

Input

Output

`rc_cursor_out` - Alle vorhandenen Rechnungen

Description

Gibt alle Rechnungen zurück. Nützlich für die Ansicht in der GUI.

F: Get invoice

```
function f_get_invoice_rc(n_invoice_id_in in NUMBER) RETURN SYS_REFCURSOR
```

Input

`n_invoice_id_in` - Die Rechnungs-ID der Rechnung.

Output

`rc_cursor_out` - Ein Pointer auf die Rechnung

Description

Gibt die Metadaten einer Rechnung zurück. Nützlich für die Ansicht in der GUI.

F: List articles (invoice_id IN)

```
function f_get_invoice_articles_rc(n_invoice_id_in in NUMBER) RETURN  
SYS_REFCURSOR
```

Input

`n_invoice_id_in` - Die Rechnungs-ID der Rechnung.

Output

`rc_cursor_out` - Alle vorhandenen Positionen

Description

Gibt alle Positionen einer Rechnung zurück. Nützlich für die Ansicht in der GUI.

Statistics

F: Meistverkauften Artikel

```
function f_get_most_bought_article_rt RETURN BONUS_ARTICLES%ROWTYPE;
```

Input

Output

`rt_article_row_out` - Ein Eintrag aus der Tabelle Artikel.

Description

Gibt den Artikel zurück mit dem größten `sold_units` Attribut. Zählt **nicht** die Häufigkeit eines Artikels in den Positionen. Diese Statistik kann mit dem zurücksetzen der verkauften Einheiten eines Produktes verändert werden.

F: Kunde mit dem größten Umsatz

```
function f_get_customer_with_highest_turnover_rt RETURN BONUS_ARTICLES%ROWTYPE;
```

Input

Output

`rt_article_row_out` - Ein Eintrag aus der Tabelle Artikel.

Description

Gibt den Artikel zurück mit dem größten `sold_units` Attribut. Zählt **nicht** die Häufigkeit eines Artikels in den Positionen. Diese Statistik kann mit dem zurücksetzen der verkauften Einheiten eines Produktes verändert werden.

P: Verkaufszahlen zurücksetzen

```
PROCEDURE ps_reset_sold_units(  
    n_article_id_in IN NUMBER  
)
```

Input

Output

`n_article_id_in` - Der Artikel welcher zurückgesetzt werden soll.

Description

Setzt die `sold_units` des Artikels auf null (0).