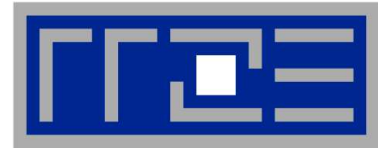


Zugriff auf Firebird-Datenbanken mit PHP

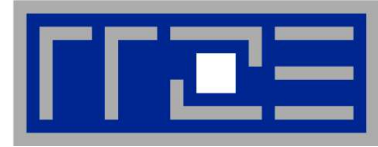
Daniel de West

DB-Campus-Treffen

15. Januar 2004



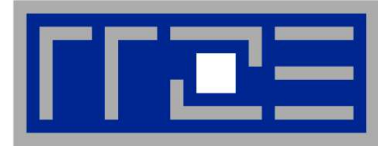
- **PHP und Firebird**
- **Die wichtigsten Befehle**
 - **Verbindungsaufbau**
 - **Übermitteln von Abfragen**
 - **Beenden von Verbindungen**
 - **Persistente Verbindungen**
 - **Transaktionen**
- **Vorstellung der Klasse class_fbird**
 - **Bsp. 1: Verbindungsaufbau, Abfrage, Ende**
 - **Bsp. 2: Ausführen von Stored Procedures mit und ohne Ergebnis**
- **Überblick über Interbase/Firebird-Befehle in PHP**
- **Weiterführende Links**



Aufbau der Verbindung:

```
ibase_connect ( string database [, string  
    username [, string password [, string charset  
    [, int buffers [, int dialect [, string  
    role]]]] ] )
```

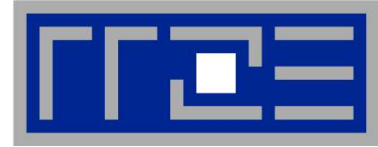
Liefert eine Ressource-ID zurück, die einer Variablen (z. B. `$link`) übergeben werden kann.



Beenden der Verbindung:

```
ibase_close ( [int connection_id] )
```

Obwohl die Verbindung standardmäßig auch durch das Ende des Skripts beendet wird, sollte die Verbindung zur Vermeidung von unnötigen Verbindungen von Hand beendet werden.



Vorbereitung von Abfragen

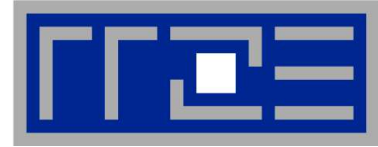
```
ibase_prepare ( [int link_identifizier, string  
query] )
```

Durchführung vorbereiteter Abfragen

```
ibase_execute ( int query [, int bind_args] )
```

Ausführen von Abfragen ohne Vorbereitung

```
ibase_query ( [int link_identifizier, string  
query [, int bind_args]] )
```



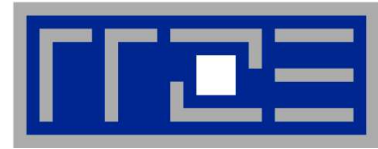
Erstellen Persistenter Verbindungen:

```
ibase_pconnect ( string database [, string  
username [, string password [, string  
charset [, int buffers [, int dialect [,  
string role]]]] ] )
```

Beenden Persistenter Verbindungen:

```
ibase_free_result ( int result_identifier )
```

Persistente Verbindungen müssen unbedingt geschlossen werden, da sie nicht selbständig nach Beendigung des Skripts geschlossen werden. Sie können außerdem nicht mit `ibase_close()` geschlossen werden.



Starten einer Transaktion

```
ibase_trans ( [int trans_args [, int link_identifizier]] )
```

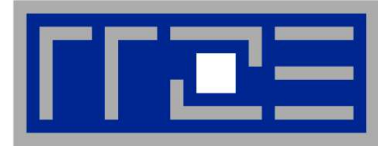
Übermitteln einer Transaktion

```
ibase_commit ( [int link_identifizier, int trans_number] )
```

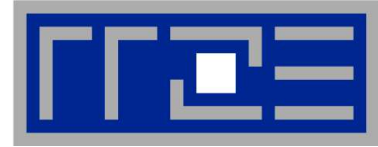
Zurücknehmen einer Transaktion

```
ibase_rollback ( [int link_identifizier, int  
trans_number] )
```

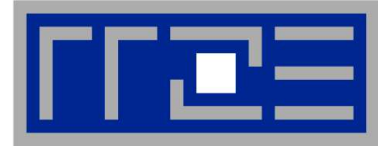
Abfragen innerhalb von Transaktionen können als ganzes übermittelt oder rückgängig gemacht werden.



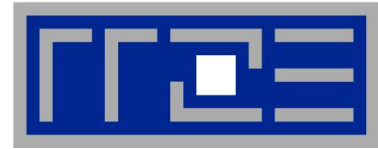
- **ibase_connect** -- Öffnet eine Verbindung zu einer InterBase-Datenbank
- **ibase_pconnect** -- Erzeugt eine permanente Verbindung zu einer InterBase-Datenbank
- **ibase_close** -- Schließt die Verbindung zu einer InterBase-Datenbank
- **ibase_errmsg** -- Gibt Fehlermeldungen zurück
- **ibase_prepare** -- Vorbereitung einer Abfrage für den folgenden Gebrauch von Parameter-Platzhaltern und für die eigentliche Ausführung.
- **ibase_execute** -- Ausführen einer vorbereiteten Abfrage
- **ibase_query** -- Führt eine Abfrage (Query) auf eine InterBase-DB aus
- **ibase_trans** -- Beginnt eine Transaktion
- **ibase_commit** -- Übermittelt eine Transaktion an die Ausführung
- **ibase_rollback** -- Macht eine Transaktion rückgängig



- **ibase_timefmt** -- Bestimmt das Format von Zeitstempel-, Datums- und Zeit-Feldern, die von einer Abfrage zurück gegeben werden
- **ibase_fetch_object** -- Liest einen Datensatz einer InterBase-Datenbank als Objekt ein
- **ibase_fetch_row** -- Liest einen Datensatz aus einer InterBase-Datenbank
- **ibase_field_info** -- Get information about a field
- **ibase_free_query** -- Gibt den Speicher einer vorbereiteten Abfrage wieder frei
- **ibase_free_result** -- Gibt den Speicher eines Abfrage-Ergebnisses frei
- **ibase_num_fields** -- Ermittelt die Anzahl der Felder einer Ergebnis-Liste



- **ibase_blob_add** -- Add data into created blob
- **ibase_blob_cancel** -- Cancel creating blob
- **ibase_blob_close** -- Close blob
- **ibase_blob_create** -- Create blob for adding data
- **ibase_blob_echo** -- Output blob contents to browser
- **ibase_blob_get** -- Get len bytes data from open blob
- **ibase_blob_import** -- Create blob, copy file in it, and close it
- **ibase_blob_info** -- Return blob length and other useful info
- **ibase_blob_open** -- Open blob for retrieving data parts

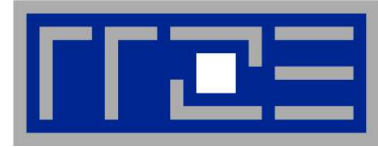


Sinn der Klasse class_fbird ist es:

- **Ständig wiederkehrende Programmteile auszulagern**
- **Fehler bei Standardproblemstellungen zu vermeiden**
- **In begrenztem Umfang Abfragen zu optimieren**
- **Den Umstieg von MySQL auf Firebird zu erleichtern**

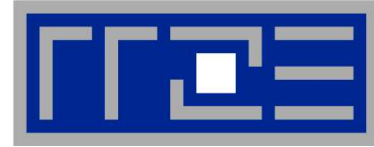
Die Klasse class_fbird kann in der jeweils aktuellsten Version von folgender Seite heruntergeladen werden:

<http://www.rrze.uni-erlangen.de/dienste/firebird/zugriff/php/>



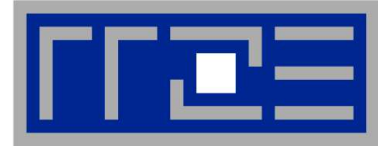
Beispiel 1: Verbindungsaufbau, Abfrage, Ende

```
include_once("class_fbird.php");
$db=new fbird('DBPHOENIX:campus2004', 'DBCAMPUS',
  '*****', 'ISO8859_1',0,3,'DEMO');
$erg=$db->query("SELECT * FROM PERSON WHERE NAME
  LIKE 'A%' AND PLZ > 95000 ORDER BY \"Strasse\"");
while ($zeile=$db->fetch_row($erg)) {
    $x++;
    echo "Zeile $x:<br>";
    print_r($zeile);
    echo "<br><br>";
}
$db->ende();
```



Beispiel 2: Ausführen von Stored Procedures mit und ohne Ergebnis

```
include_once("class_fbird_extended.php");
$db=new fbird('DBPHOENIX:campus2004', 'DBCAMPUS', '*****',
    'ISO8859_1',0,3,'DEMO');
$params=array(-7,'TEST','PHP','','',NULL,'',NULL,NULL, 'RZ123');
$erg=$db->proc_execute("PERSON_IU",$params);
echo "ERG: $erg";
$output_vars=array('ID','NAME','VORNAME','Strasse','E-
    Mail','PLZ','Ort','Land','Homepage','Kundennummer');
$params="";
$limit="FIRST 20";
$erg=$db->query_proc("PERSON_S",$params,$output_vars,$limit);
foreach ($erg as $erg) {
    $x++;
    echo "Zeile $x:<br>";
    print_r($erg);
    echo "<br><br>";
}
$erg=$db->proc_execute("DELMINUS");
$db->ende();
```



function fbird

(\$datei,\$user,\$pass,\$char,\$buffer,\$dialect,\$role='')

Initialisierung und Aufbau der DB-Verbindung

function query(\$abfrage)

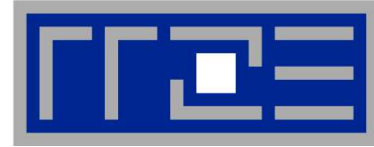
Hier können gültige SQL-Abfragen übergeben werden

function fetch_row(\$erg)

Hier wird die nächste Zeile eines Ergebnisses ausgegeben

function ende()

Schliessen der Datenbankverbindung

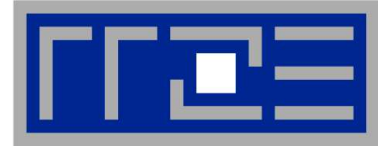


function query_proc(\$proc_name,\$parameters,\$output_vars,\$limit)
Generierung und Ausführung einer SQL-Abfrage, die auf einer Stored-Proc beruht und der ein einzelner oder ein Array von Parametern und Ausgabe-Namen übergeben werden kann. Mit \$limit kann z. B. „FIRST 20“ übergeben werden.

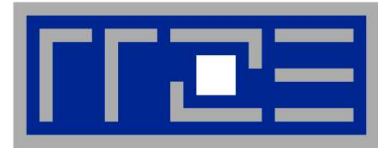
function proc_execute(\$proc_name,\$parameters)
Generierung einer SQL-Abfrage, die auf einer Stored-Proc beruht und der ein einzelner oder ein Array von Parametern übergeben werden kann. Diese Procedure liefert kein Ergebnis zurück.

function us2fb(\$datum)
Ändert das Datumsformat vom MySQL-Typ (2003-01-31) in den Firebird-Typ (2003/31/01)

Function fb2us(\$datum)
Ändert das Datumsformat vom Firebird-Typ (2003/31/01) in den MySQL-Typ (2003-01-31)



- **Beginners Guide to Firebird and PHP**
<http://www.ibphoenix.com/a5056.htm?a=ibphoenix&page=php>
- **Getting Started with PHP and Interbase**
<http://www.ibphoenix.com/a489.htm>
- **Deutsches PHP-Handbuch bei Dynamic-Webpages**
<http://www.dynamic-webpages.de/php/index.php>
- **Deutsche PHP-FAQ**
<http://www.dclp-faq.de/>



**Vielen Dank für Ihre
Aufmerksamkeit!**