

# HIT-Batch-Schulungsunterlagen

*2014*

HIT/ZID-Datenbank  
im Bayerisches Staatsministerium für  
Ernährung, Landwirtschaft und Forsten  
Ludwigstr. 2  
80539 München  
Stand: 17.01.2014

## Inhaltsverzeichnis

Inhaltsverzeichnis.....	2
1. Allgemeines .....	3
1.1. HIT-Testsystem .....	3
1.2. HIT-Kompetenzen.....	4
1.3. HIT – Data Dictionary.....	5
1.3.1. Meldungen .....	5
1.4. Befehlserstellung - Hilfe über Meldeprogramm .....	6
1.4.1.1. Befehle.....	7
2. HIT-Batch .....	8
2.1. Geeigneten Texteditor installieren.....	8
2.2. Technische Voraussetzungen .....	8
2.3. Alternativ: Vereinfachte Version HIT-Batch .....	11
2.4. HIT-Batch-Abfragen .....	12
2.4.1. Operatoren .....	12
2.4.2. HIT-Abfragefunktionen.....	13
2.4.3. Filterfunktionen.....	14
2.4.3.1. Beispiele .....	15

## 1. Allgemeines

### 1.1. HIT-Testsystem

Für Übungszwecke steht das HIT-Testsystem zur Verfügung.

- Echtdaten grundsätzlich nicht ins Testsystem!
- Testeingaben grundsätzlich nicht im Echtssystem!

So ist das HIT-Testsystem zu finden:

- Link zu HitTest auf Startseite <http://www.hi-tier.de>
- oder direkt <http://www.hi-tier.de/HitTest/>

Im HitTest-Hilfetext (Klick auf [HELP?](#)) sind Test-Zugänge beschrieben.

Bei Abfragen/Eingaben immer die Kompetenzen des Betriebstyps beachten!

Testbetriebe am Beispiel von Land 01 (Schleswig-Holstein)		
Betriebsnummer	PIN	Betriebstyp
01 000 000 0001	900001	Landwirt
01 000 000 0002	900002	sonstiger Tierhalter (z.B. Zuchtverband)
01 000 000 0003	900003	Viehhändler
01 000 000 0004	900004	Schlachtbetrieb
01 000 000 0005	900005	Transporteur
Spezielle Benutzergruppen		
Anmeldung als Landwirtschaftsverwaltung		
01 000 000 0006	900006	Landwirtschaftsverwaltung
01 000 000 0007	900007	Landwirtschaftsverwaltung (erweiterte Rechte)
Anmeldung als Veterinärverwaltung		
01 000 000 0008	900008	Veterinärverwaltung
Anmeldung als BSE-Labor		
01 000 000 0081	900081	BSE-Labor
01 000 000 0082	900082	Landeslabor
01 000 000 0084	900084	amtlicher Tierarzt (BSE)
Ohrmarken zum Testen		
DE 03 000 00001	DE 09 810 00000	
DE 03 000 00002	DE 09 810 00001	
DE 03 000 00003	DE 09 810 00002	
DE 03 000 00004	DE 09 810 00003	

Generell ist der Zugangsregel zum HIT-Testsystem so:

Betriebsnummer/PIN : YY000000XXXX/90XXXX

=> YY Bundesland z.B. 09 für BY, XXXX Betriebstyp mit dem man sich anmelden will.

z.B. Veterinärverwaltung BY: 090000000008/900008

## 1.2. HIT-Kompetenzen

Die Betriebstypen in HIT haben unterschiedliche Kompetenzen. Diese sind Online einsehbar und bei Abfragen zu berücksichtigen, siehe Entwicklungsbereich: <http://www.hi-tier.de/Entwicklung/>

**Konzept**

Zurück
Home
Weiter

[HIT Data Dictionary](#)

[HIT-Protokoll](#)

[Betriebs-DB](#)

[Plausi & Datenfluss](#)

[Datensicherheit\\_Kompetenz](#)

[Tiergesundheit](#)

[EU-Belange](#)

[Meldungsverarbeitung](#)

[Sonstiges](#)

**HIT Data D**

Home
Nach ob

**Allgemeines**

Das Data-Dictionary wird in HIT-Meldungen gespeichert und ist mit

**Kompetenz**

Zurück
Home
Nach oben
Weiter

**Kompetenzen zu TYP\_BETR 8, Stand:04.06.2012 15:43:43**

ENTITY	AKTIONNR	Code	TYP_BETR	Code-Text	BEREICH	Code
Anzeigen System: <input checked="" type="radio"/> Test <input type="radio"/> Produktion exakte Suche: <input checked="" type="radio"/> nein <input type="radio"/> ja <span style="float: right;">Download</span>						
<a href="#">#BESTRTGX</a>	2	R	8	VVW - Veterinärverwaltung	99	U
<a href="#">#BEWMELD</a>	2	R	8	VVW - Veterinärverwaltung	99	U
<a href="#">#IORGDATEN</a>	2	R	8	VVW - Veterinärverwaltung	99	U
<a href="#">#TIERGESXX</a>	2	R	8	VVW - Veterinärverwaltung	99	U
<a href="#">#ADDRHOF</a>	2	R	8	VVW - Veterinärverwaltung	99	U
<a href="#">#ADDRPOST</a>	2	R	8	VVW - Veterinärverwaltung	99	U
<a href="#">#BESTREG</a>	2	R	8	VVW - Veterinärverwaltung	99	U
<a href="#">#BESTREX</a>	2	R	8	VVW - Veterinärverwaltung	99	U
<a href="#">#BESTREY</a>	2	R	8	VVW - Veterinärverwaltung	99	U
<a href="#">#BESTRTG</a>	2	R	8	VVW - Veterinärverwaltung	99	U

Kompetenzabfrage für Betriebstyp 8  
(Veterinärverwaltung)

4 / 16

## 1.3. HIT - Data Dictionary

<http://www.hi-tier.de/Entwicklung/Konzept/Default.htm>

Im „**HIT Data Dictionary**“ sind die Melde-Entitäten, Codesets, Plausibilitätsprüfungen usw. zu finden. In „Meld-Elemente“ werden die „Feldnamen“ für die einzelnen Meldungen, die für die Befehle erforderlich sind, aufgelistet.

The screenshot shows the HIT website interface. On the left, there is a navigation menu with links like 'Aktuelles', 'Informationen', 'Regionalstellen', etc. The main content area is titled 'Herkunftssicherungs- und Informations...' and contains sections for 'Meldeprogramm', 'Zuständige Stellen', and 'Weitere Angebote'. A red box highlights the link 'www.hi-tier.de/Entwicklung/ Bereich für Software-Entwickler'. On the right, there is a 'Hinweise' section and a 'Konzept' section. A red box highlights the 'HIT Data Dictionary' link in the 'Konzept' section. A red arrow points from the 'HIT Data Dictionary' link to the 'HIT Data Dictionary' page shown in the next screenshot.

### 1.3.1. Meldungen

Abfrage der Datenstruktur einer speziellen Meldung (Entity) durch Eingabe in die Spalte „Entity“, z.B. Geb und dann ANZEIGEN

The screenshot shows the 'HIT Data Dictionary' page. The 'Allgemeines' section is active, showing a table of 'Meldungen zu Entity GEB, Stand: 28.03.2012 11:00:32'. A red box highlights the 'Entity' column header and the 'GEB' row. A red arrow points from the 'HIT Data Dictionary' link in the previous screenshot to this page. Another red arrow points from the 'GEB' row to the 'Anzeigen' button. The table below shows the structure of the 'GEB' entity.

Entity	Entityname	Typ	BNR.Komp	Präm.Komp	Delta_Von	Delta_Bis	Delta_Von(2)	Delta_Bis(2)
GEB	Geburtsmeldung	Tier-Bewegung	BNR15		SYS_VON	SYS_BIS		
GEBURTA	Geburtsmeldung, alte zwischen 1.7.98 und 26.9.99	Tier-Bewegung	BNR15		SYS_VON	SYS_BIS		
GEBURTK	Geburt (nur lesend, insbes. für Kalbung)	Abfrage	BNR15		SYS_VON	SYS_BIS		

Klick auf Meldung (z.B. Geburt) führt zu den Meld-Elementen (Feldnamenliste)



### 1.4.1.1. Befehle

Gruppe	Befehl	Erläuterung
1	<u>E</u> xecute	Ausführen von Kommandos, wie z.B. Logon, Logoff, Parameter setzen sowie Ausführen von Meldungen bei denen der Sender nicht weiß, ob Insert oder Update nötig, insbesondere bei Betriebsdaten-Meldungen mit automatischer Historisierung
2	<u>I</u> nsert	Übertragen oder Pflegen von Datensätzen der einzelnen Meldungsarten
	<u>U</u> pdate	
	<u>S</u> torno	
	<u>D</u> elete	
3	<u>R</u> etrieve	Abholen von verschiedenen Daten vom Server
4	<u>C</u> onfirm	Bestätigen bereits abgespeicherter Sätze insbesondere im Konfliktfall (früher XS/S)

#### Schematische Darstellung

\*<Nr>:<Aktions-Code><Stückelung>/<Sub-Code>:<Meldung>/<Feldliste>:<Datenliste>

<http://www.hi-tier.de/Entwicklung/Konzept/HITP/feink040.html>

Syntax und Bestandteile		
Token	Definition	Bemerkung
<Befehl>	<Einleitung>:<Aktion>:<Objekt>:<Datenliste>	4 Haupt-Elemente, mit Doppelpunkt getrennt
<Einleitung>	<Befehls-Kenner><Befehls-Nummer>	
<Befehls-Kenner>	* (Stern)   + (Pluszeichen)	kennzeichnet Befehlsbeginn, + wenn weitere Befehlssteile folgen, * beim letzten oder einzigen Befehlssteil
<Befehls-Nummer>	<Haupt-Nummer>[+<Unter-Nummer>][#<Rowkeys>]	Achte auf Trennung der Unter-Nummer durch '+'
<Haupt-Nummer>	Fortlaufende Nummer des Befehl	dient zur richtigen Zuordnung des Ergebnisses und Sicherung der Reihenfolge und Vollständigkeit, beginnt bei 1 für die Connection
<Unter-Nummer>	Fortlaufende Nummer des Teilbefehls, wenn der Befehl aus mehreren Zeilen besteht	dient zur Sicherung der Reihenfolge und Vollständigkeit wenn Teilbefehle vorliegen, beginnt bei 1 für jede neue Haupt-Nummer
<Rowkeys>	<Rowkey>[<Rowkeys>]	Liste von benutzerdefinierten Keys
<Rowkey>	benutzerdefinierten Key (String-Konstante)	dient zur Identifikation des Befehls
<Aktion>	<Aktions-Code><Stückelung>[<Sub-Codeliste>]	
<Aktions-Code>	X I U S D R	X (=Execute)   I (=Insert)   U (=Update)   S (=Storno)   D (=Delete)   R (=Retrieve)   C (=Confirm)
<Stückelung>	F S B T	F (=Feld)   S (=Satz)   B (=Block)   T (=Transaktion)
<Sub-Codeliste>	<Sub-Code>[<Sub-Codeliste>]	ein Sub-Code oder mit Semikolon getrennt mehrere Sub-Codes
<Sub-Code>	T E S A P O ...	T (=Trotzdem)   E (=ENDE)   S (=SICHER)   A (=Abbruch)   P (=Panik)   O (=NOOP) ... siehe <a href="#">Sub-Codes</a> , für RETRIEVE gibt es noch weitere Sub-Codes, siehe <a href="#">HIT-QL</a>
<Objekt>	[<Meldung>][<Feldliste>]	Meldung kann bei nachfolgenden Feldern der selben Meldung fehlen
<Meldung>	LOGON BETRD GEBURT ....	Feste Strings, siehe Meldung
<Feldliste>	<Feldname>[<Feldliste>]	ein Feld oder mit Semikolon getrennt mehrere Felder
<Feldname>	BNR15 PIN ...	komplette Liste je nach Meldung, siehe Meldungselemente
<Datenliste>	<Datenelement>[<Datenliste>]	ein Datenwert oder mit Semikolon getrennt mehrere Datenwerte

#### Spezielle RS- Abfragen

RS/I	Inclusive stornierte Daten Beispiel: : *2:RS/I:BTR_D/*:BNR15;=;090000000001 Normalerweise werden bei Tabellen, die Historisierung unterstützen, nur aktuelle Sätze angezeigt. Mit dem Subcode I erhält man alle Sätze, inklusive der Stornierten.
RS/Z	Abfrage zu einem bestimmten Zeitpunkt
RS/D	Delta-Abfrage von Daten, die nach der letzten Abfrage hinzugekommen sind
RS/H	Delta-Abfrage und Historie (auch stornierte Meldungen)

## 2. HIT-Batch

### 2.1. Geeigneten Texteditor installieren

z.B. „Notepad++6.0“ von Heise

- <http://www.heise.de/download/notepad.html>
- Notepad auf Laufwerk speichern, auf das man Zugriff hat, installieren

Das Windows-Notepad ist wenig geeignet!



### 2.2. Technische Voraussetzungen

**HI-Tier**  
Herkunftssicherungs- und Informationssystem für Tiere

**HI-Tier Entwicklung**  
Das Web "Entwicklung" dient zur Koordination der Entwicklung des Projekts "HI-Tier : Herkunftssicherungs- und Informationssystem für Tiere".

**Programme**  
Allgemeines  
Die zentrale Datenbank des HI-Tier ist so aufgebaut, dass über ein von uns definiertes und offengelegtes Kommunikations-Protokoll (HIT-Protokoll) via Internet Programme, sogenannte HIT-Clients, mit der Zentralen Datenbank (ZDB) des HI-Tier Daten austauschen können. So wie z.B. beliebige I-Programme verschiedenster Hersteller Mail-Servern anderer Hersteller im Internet Daten austauschen können, wenn sie selbste Kommunikations-Protokoll

**Batch-Client**  
Allgemeine Hinweise  
Der Batch-Client ist ein Programm das große Dateien von und zur Zentralen Datenbank überträgt. Das Format oder auch ADIS/ADED benutzt werden. Damit auf jedem Betriebssystem das Java unterstützt von der ZDB kostenlos über das Internet zum Download Softwareherstellern in deren Programm integriert

**Installation**  
Download  
aktuelle Produktions-Version  
Stand: Version 41 vom 23.12.2011  
Dateibereich/HitBatch.jar  
Dateibereich/HitUpros.jar - nur Programme ohne alles weitere  
Dateibereich/HitBatch.zip  
Dateibereich/HitUpros.zip - Programme + Beispiele und Aufrufdateien  
Dateibereich/HitBatchSources.zip  
Dateibereich/HitUprosSources.zip - die Quellen der Programme  
neueste Version (erst noch zu testen)  
Stand: Version 42 vom 29.03.2012  
Dateibereich/beta/HitBatch.jar  
Dateibereich/beta/HitUpros.jar - nur Programme ohne alles weitere  
Dateibereich/beta/HitBatch.zip  
Dateibereich/beta/HitUpros.zip - Programme + Beispiele und Aufrufdateien  
Dateibereich/beta/HitBatchSources.zip  
Dateibereich/beta/HitUprosSources.zip - die Quellen der Programme  
Ab sofort unterstützt der HitBatch nur noch Java-Versionen >= 1.5.

Oder Link direkt: <http://www.hi-tier.de/Entwicklung/Programme/batch005.html>

- Download „HitBatch.zip“ und „Upros.zip“, beide öffnen, ZIP-Inhalt in gemeinsamen Ordner
- ZIP-Datei extrahieren
- HITBATCH.cmd: ausführende Datei („command“) → Start der Abfrage
  - in Texteditor öffnen, unter „ set BASEDIR“ Pfad dieses Ordners angeben, z.B. C:\HIT
- HITBATCH.ini: Initialisierungsdatei - Steuerung und Konfigurationseinstellungen (INI-Parameter, siehe <http://www.hi-tier.de/Entwicklung/Programme/batch002.html> )
  - in Texteditor öffnen
  - Mit !!! Parameter prüfen bzw. anpassen
  - User-ID (Betriebsnummer und PIN), MBN, Mandant anpassen
  - PIN als txt-Datei anlegen, Pfad zur Datei bei PIN angeben  
PIN-Automatik: Für automatisierte Vorgänge => wenn PIN abläuft wird vom System automatisch neue PIN vergeben und in Datei geschrieben. Nur geeignet für „reine“ HIT-Batch-Zugänge, da diese PIN nicht einsehbar.
  - Port wie bei **Testcase.ini** (siehe 2.1.2.1)
  - Schwere 0 => alles, auch Erfolg wird mitgeschrieben
  - Man kann bis zu 50 verschiedene Arbeitsaufträge= SET bearbeiten. Wenn „setcout“ = 1 führt er nur den 1. Set aus
- Datei.in: beinhaltet die Abfrage
  - Zeile 1: <AUSGABEFELDER>
  - Zeile 2 : <BEDINGUNG>
- Datei.log: Transaktionsprotokoll (LOG-Datei prüfen, insbesondere beim Senden von Daten→Fehlermeldungen)

**HITBatch.CMD**

```

1  @echo off
2  REM
3  REM Aufrufscript
4  REM
5
6  cd /d %~dp0
7
8  set BASEDIR=\projekt\HIT
9
10
11  :AGAIN
12  set HBINI=%1
13  if %HBINI%.==. set HBINI=HitBatch.ini
14  if exist %HBINI% goto RUNHB
15  echo Parameter-Datei: %HBINI% nicht gefunden
16  goto FERTIG
17

```

Pfad des Ordners angeben, in dem die herunter geladenen HitBatch.zip“ und „Upros.zip gespeichert sind!

## HITBatch.INI

```

1  ;
2  ;-----
3  ; HitBatch Version 39, Stand 05.08.2011
4  ;-----
5  ;
6  ; Ausführliche Kommentare für HitBatch-Client
7  ;-----
8  ;
9  ; Der Paragraph [Global] wird zuerst ausgewertet
10 ;
11 ; für den Test stehen zunächst für jedes Bundesland verschiedene Userid/PIN-Kombinationen zur Verfügung
12 ; Userid/Betriebsnummern i.d.F 276bb0000000xx
13 ; PIN/Passwort i.d.F. 9000xx
14 ; mit bb=01..16 Bundesland und
15 ;
16 ; Userid / Betriebsnummer der s
17 ; Userid=27609000000015
18 ;
19 ; MBN : Mitbenutzernummer
20 ; Mbn=0
21 ;
22 ; MANDANT : Mandant, für den HI
23 ; Mandant=0
24 ;
25 ; MANDANT : Mandant, für den HIT-Gesamt-Berechtigungen vorliegt und für den hier jetzt agiert werden soll (Default: 0=Keiner)
26 ; Mandant=0
27 ;
28 ; PIN: Passwort für Anmeldung unter Betriebsnummer und ggf. Mitbenutzernummer
29 ; Folgende Angaben sind möglich:
30 ; ? Abfrage des Passwortes über Standardeingabe (STDIN) der Console.
31 ; %datei> Laden der PIN als Klartext aus <datei>
32 ; %<datei> Laden der verschlüsselten PIN aus <datei>
33 ; Der Zugriff auf diese Dateien ist entsprechend
34 ; Variante %datei> ist dringend zu empfehlen.
35 ;
36 ; Pin=#\HIT-Batch-Schulung\PIN
37 ;
38 ;
39 ;
40 ;
41 ;
42 ;
43 ;
44 ;
45 ;
46 ;
47 ;
48 ;
49 ;
50 ;
51 ; Testsystem, Test-BNR und PIN erforderlich
52 ; PrimaryServer=212.18.9.50
53 ; PrimaryPort=2223
54 ; BackupServer=194.94.221.1
55 ; BackupPort=2223
56 ;
57 ; PRIMARYSERVER=hitserver.hi-tier.bybn.de
58 ; PRIMARYPORT=2223
59 ; BACKUPSERVER=hitbackup.hi-tier.bybn.de
60 ; BACKUPPORT=2223
61 ;
62 ;
63 ; Welche Systemmeldungen kommen ins Protokoll/Bildschirm
64 ; 0=Urgent nur dringende Nachrichten,
65 ; 1=Important wichtige Nachrichten
66 ; 2=Informativ die wichtigsten Ablaufschritte
67 ; 3=VeryVerbose sehr geschwätzig, Testausgaben
68 ; 4=Debug gebe auch noch alle SEND-Strings aus (außer Anmeldestring mit PIN)
69 ; 5=DebugX -- " -- inklusive Anmeldestring mit PIN
70 ;
71 ;
72 ; PROTOLEVEL=1
73 ;
74 ;
75 ;
76 ;
77 ;
78 ;
79 ;
80 ;
81 ;
82 ;
83 ;
84 ;
85 ;
86 ;
87 ;
88 ;
89 ;
90 ;
91 ;
92 ;
93 ;
94 ;
95 ;
96 ;
97 ;
98 ;
99 ; Inputtest - dieser Parameter ist seit der Version 39 entfallen, da die CSV-Verarbeitung umgestellt wurde
100 ; InputTest=0
101 ;
102 ;
103 ;
104 ;
105 ;
106 ;
107 ;
108 ;
109 ;
110 ;
111 ;
112 ;
113 ;
114 ;
115 ;
116 ;
117 ;
118 ;
119 ;
120 ;
121 ;
122 ;
123 ;
124 ;
125 ;
126 ;
127 ;
128 ;
129 ;
130 ;
131 ;
132 ;
133 ;
134 ;
135 ;
136 ;
137 ;
138 ;
139 ;
140 ;
141 ;
142 ;
143 ;
144 ;
145 ;
146 ;
147 ;
148 ;
149 ;
150 ;
151 ;
152 ;
153 ;
154 ;
155 ;
156 ;
157 ;
158 ;
159 ;
160 ;
161 ;
162 ;
163 ;
164 ;
165 ;
166 ;
167 ;
168 ;
169 ;
170 ;
171 ;
172 ;
173 ;
174 ;
175 ;
176 ;
177 ;
178 ;
179 ;
180 ;
181 ;
182 ;
183 ;
184 ;
185 ;
186 ;
187 ;
188 ;
189 ;
190 ;
191 ;
192 ;
193 ;
194 ;
195 ;
196 ;
197 ;
198 ;
199 ;
200 ;
201 ;
202 ;
203 ;
204 ;
                
```

**User-ID=** In Testdatenbank die Test-Betriebsnummer mit entsprechender Kompetenz wählen (hier als Regionalstelle- 15); in Produktion die eigene Betriebsnummer eintragen.

Für **PIN** ggf. Ordner anlegen; verschlüsselte PIN-Übertragung mit # → PIN=#:\HIT-Batch-Schulung\PIN

Server prüfen:  
Adressen außerhalb des bayer. Behördennetz:  
**PRIMARYSERVER**=hitserver.hi-tier.de  
**PRIMARYPORT**=2223 (Test) oder 2222 (Produktion)  
**BACKUPSERVER**=hitbackup.hi-tier.de

**Schwere=0** (auch Erfolgsmeldung wird ausgegeben)  
Bei der „Schwere“ wird festgelegt, welche Hinweise/Nachfragen/Fehler in Log-Datei ausgegeben werden

**Setcount=** Anzahl der auszuführenden Aufträge (Set)  
**Startset=** Nummer des Set, bei dem der HitBatch beginnen soll

## 2.3. Alternativ: Vereinfachte Version HIT-Batch

<http://www.hi-tier.de/entwicklung/Programme/batch008.html>

(INI-Datei auf wesentliche Struktur reduziert -ohne Kommentare und Beispielabfragen)

The screenshot shows the website interface with several key elements:

- HI-Tier Entwicklung** header with navigation links: [Neuigk./Historie](#), [Grundlagen](#), [Konzept](#), [Vorträge](#), [Technik](#), [Programme](#), [Instanz](#).
- Programme** section with buttons: [Zurück](#), [Home](#), [Weiter](#).
- Batch-Client** section with buttons: [Zurück](#), [Home](#), [Nach oben](#).
- weitere Beispiele** section containing a table of examples.

Inhalt	Was (Details ggf. beim Link)	Wie (Zusammenfassung)	Wo (Download)
Muster mit simplen Beispielabfrage	1. Lesen aller Sätze mit allen Feldern der Meldung 'Ernt' DEMO 2. vereinfachtes Lesen über Parameter BEFENH...		<a href="#">DateibereichHitBatchSamples_muster.zip</a>
BVD-STATUS: Eintragen von Betrieben in die Tabelle "BVD-Betriebsstatus"	1. Schritt: Hole Betriebe im LandBezirk 091 mit Betriebsklärung, gültig bis mindestens 31.12.2010 2. Schritt: Hole Betriebe im LandBezirk 091 aus BVD-Betriebsstatus 3. Schritt: Filtere Ergebnisse aus Betriebsklärung um Ergebnisse aus BVD-Betriebsstatus 4. Schritt: Einträge alle noch nicht vorhandenen Betriebe in die Tabelle BVD-Betriebsstatus		<a href="#">DateibereichHitBatchSamplesBVDGSTA_Betriebsstatus.zip</a>
BVD-Status abfragen			(noch kein fertiger Download, Daten siehe unten)
BVD-Tiere mit positive Überläufe mit Nachforschungen			(noch kein fertiger Download, Daten siehe unten)
BHV1 - Betriebe mit Produktionsrichtung Milchkuhhaltung, ohne Bestandsuntersuchung im letzten Jahr	1. Betriebe mit Produktionsrichtung Milchkuhhaltung 2. Ermittle Betriebe mit Bestandsuntersuchung im letzten Jahr 3. Filtere - suche alle in 1) die nicht in 2) vorkommen		(noch kein fertiger Download, Daten siehe unten)
Zerschneidestellen (ZSZ)	1. Abfrage Bestandsregister zur Ermittlung der Tiere mit Kälbern bzw. ZSZ 2. weitere Berechnung z.B. mit Excel		(noch kein fertiger Download, Daten siehe unten)

Download des Batch mit Beispielabfragen

The screenshot shows the contents of the batch files:

```

HitBatch.in | HitBatch.cmd
1 @echo off
2 REM -----
3 REM ggf. wechsele Laufwerk und Pfad ...
4 REM danach sind alle Pfadangaben relativ zu diesem Verzeichnis
5 REM unnötig wenn HitBatch.c
6 REM -----
7 REM cd /D C:\HIT\_muster
8
9 REM Parallekl zum Verzeichnis
10 REM Das sind die entpackter
11
12 set CLASSPATH=..\HitBatch\H
13 java.exe HitBatch.HitBatchM
14
15 pause
16
HitBatch.ini | HitBatch.cmd
50 [SET-1]
51 MELDUNG=DEMO
52 COMMAND=RS
53 INFILE=_muster.in
54 OUF FILE=_muster.out
55 LOGFILE= muster.LOG
56
57 CSVIN=0
58 CSVOUT=0
59 CSVLOG=0
60 FUNCONLY=0
61 OUTAPPEND=0
62 LOGAPPEND=0
    
```

Relative Pfadangabe → damit muss man keinen konkreten Pfad in der INI-Datei angeben

## 2.4. HIT-Batch-Abfragen

<http://www.hi-tier.de/Entwicklung/Programme/batch001.html>

Eine HIT-Abfrage wird über den R(etrieval)-Befehl ausgeführt. Man gibt die Tabelle an, die abgefragt werden soll, dann die Felder, die ausgegeben werden sollen und schließlich noch die Abfragebedingung. Diese vier Teile gibt man wie folgt an:

\*x:<BEFEHL>:<TABELLE>/<AUSGABEFELDER>:<BEDINGUNG>

**Beispiel:** Anzahl Tiere mit BVD-Status P9 - P99 in Land 03, gruppiert nach Status (lebend /tot/sonstiges)

### HitBatch.INI

[SET-1]

Befehl=\*1:RS:TIERSTATX/ST\_LEBEND;\$COUNT(\*):BLAND\_LMON;=;03;AND;BVD\_ERG;BW;409;499;GROUP;ST\_LEBEND

OUFIL=I:\HIT-Batch-Schulung \Abfrage1.OUT (Ergebnisdatei)

LOGFILE=I:\HIT-Batch-Schulung \Abfrage1.LOG (Log-Datei=>zur Erfolgskontrolle bzw. Prüfen von Fehlermeldungen)

OUTAPPEND=0 (legt fest, ob beim Aufruf des SET die Daten im OUFIL überschrieben oder hinten angehängt)

LOGAPPEND=0 (legt fest, ob beim Aufruf des SET die Daten im LOGFILE überschrieben oder hinten angehängt)

CSVIN=0 (Format für Eingabedatei festlegen)

CSVOUT=0 (Format für Ausgabedatei festlegen)

### 2.4.1. Operatoren

<http://www.hi-tier.de/Entwicklung/Konzept/HITP/feink041.html>

**Beispiel:** Lebende Tiere mit BVD-Status P10 und P11 in NRW

\*1: RS: TIERSTATX/LOM:BNR15\_LMON;EQ;05;AND;BVD\_ERG;IN;(410,411);AND;ST\_LEBEND;=;1

#### Operator-Liste

Operator	Vergleichsziel	oder	Definition	Beschreibung	entspricht
EQ	<Feldinhalt> <Konstante>	=	EQUAL	gleich	=
EQX	<Feldinhalt> <Konstante>		EQUAL EXTENDED	gleich, case sensitiv	
NE	<Feldinhalt> <Konstante>	^=	NOT EQUAL	nicht gleich	^=
NEX	<Feldinhalt> <Konstante>		NOT EQX	nicht gleich, case sensitiv	
LT	<Feldinhalt> <Konstante>	<	LESS TO	kleiner als	<
LE	<Feldinhalt> <Konstante>	<=	LESS EQUAL	kleiner gleich	<=
GT	<Feldinhalt> <Konstante>	>	GREATER TO	größer als	>
GE	<Feldinhalt> <Konstante>	>=	GREAT EQUAL	größer gleich	>=
IN	<Konstantenliste>	IN	IN	in nachfolgender Liste	IN ( x , y , z )
NI	<Konstantenliste>	NOT IN	NOT IN	nicht in Liste	NOT IN ( x , y , z )
BW	<Konstante>;<Konstante>	BETWEEN	BETWEEN	zwischen x und y	BETWEEN x AND y
NB	<Konstante>;<Konstante>	NOT BETWEEN	NOT BETWEEN	nicht zwischen	NOT BETWEEN
LI	<Konstante> mit Wildcard "%25"	LIKE	LIKE	gleich mit Wildcard/Joker	
LIX	<Konstante> mit Wildcard "%25"	LIKEX	LIKE EXTENDED	gleich mit Wildcard/Joker, case sensitiv	
NL	<Konstante> mit Wildcard "%25"	NOT LIKE	NOT LIKE	nicht gleich mit Wildcard/Joker	
NLX	<Konstante> mit Wildcard "%25"	NOT LIKEX	NOT LIKE EXTENDED	nicht gleich mit Wildcard/Joker, case sensitiv	
IS	<erweiterte Vergleichsbedingung>	*	Kontext spezifisch	in der Vergleichsbedingung können Werte, Listen und erweiterte Bereiche angegeben werden, darunter auch Landkreisbereichskürzel und KFZ-Kennzeichen. Genaue Beschreibung siehe bei "Komplexe Vergleichsbedingung" (Status: experimentell).	

#### Verknüpfungs-Liste

Operator	Definition	Beschreibung
AND	logisches UND	und zugleich
OR	logisches ODER	oder
NAND	logisches UND NICHT	und nicht
NOR	logisches ODER NICHT	oder nicht

## 2.4.2. HIT-Abfragefunktionen

<http://www.hi-tier.de/Entwicklung/Konzept/HITP/feink045.html>

Im Rahmen der Abfragesprache HIT-QL (Query-Language) wurde über HIT-Abfragefunktionen die Möglichkeit geschaffen, zusätzlich zu den Feldern einer abfragbaren Entität davon abgeleitete Informationen, wie z.B. Anzahl Datensätze oder besser lesbare Alphadarstellungen der Ohrmarke/Betriebsnummer zu ermitteln (276000091234567 wird zu DE 09 123 4567).

### Beispiel: Betriebe mit lebenden BHV1-Reagenten in einem Land, z.B. NRW

\*1: RS:

```
TIERSTATX/$BNRKREIS(BNR15_LMON);$ALPHABNR(BNR15_LMON);$ALPHALOM(LOM);GEB_DATR;RASSE;GESCHL_R;BHV1_EMI;BHV1_BDAT
BNR15_LMON;=:05;AND;BHV1_EMI;IN;(20,21,22,23,24,27,28,29,52,55,62);AND;ST_LEBEND;EQ;
1;ORDER;BNR15_LMON;LOM
```

### Beispiel: Zählen der Betriebe und Tiere mit BHV1-Status IM in einem Land, z.B. Niedersachsen

\*1:RS:TIERSTATX/\$COUNTD(BNR15\_LMON/Anz\_Betriebe);\$COUNT(\*/Anz\_Tiere):

```
BNR15_LMON;IS;03;AND;BHV1_EMI;EQ;59
```

Count => zählt die Zeilen

CountD => zählt verschiedene Zeilen (Betrieb wird dadurch nur einmal gezählt)

Funktion	
\$COUNT(*)	Zeilenanzahl
\$COUNTB(*)	Zeilenanzahl, bei sehr großen Datenmengen
\$COUNTF(RASSE)	Zeilenanzahl mit vorhandener Rasse
\$COUNTD(BNR15)	Anzahl verschiedener Betriebe
\$COUNT(*/ANZ_GEBURTEN)	Zeilenanzahl, als Ergebnis-Spalte "ANZ_GEBURTEN"
\$MIN(GEB_DATR)	kleinstes Geburtsdatum, Outputname ist Standard
\$MIN(GEB_DATR/FIRST_DAT)	kleinstes Geburtsdatum, explizite Angabe des Outputnamens
\$YEAR(GEB_DATR/JAHR_GEB)	Jahr aus dem Geburtsdatum, explizite Angabe des Outputnamens
\$MONTH(GEB_DATR)	Monat aus dem Geburtsdatum
\$DAY(GEB_DATR)	Tag aus dem Geburtsdatum
\$DAYS(GEB_DATR)	Laufende Nummer des Tag seit 1.1.1970
\$DATE(SYS_VON)	Datum aus dem Timestamp SYS_VON
\$ALPHALOM(LOM_MUT)	Ohrmarke in aufbereiteter Form
\$LOMSTAAT(LOM)	Staat (numerisch) aus der Ohrmarke, z.B: 276 für DE
\$LOMLAND(LOM)	Land (numerisch) aus der Ohrmarke, z.B: 16 für Thüringen
\$BNRSTAAT(BNR15)	Staat (numerisch), z.B: 276 für DE bei Betrieb 09 177 148 0001
\$BNRLAND(BNR15)	Land (numerisch), z.B: 9 bei Betrieb 09 177 148 0001
\$BNRREGBEZ(BNR15)	Regierungsbezirk (numerisch), z.B: 1 bei Betrieb 09 177 148 0001
\$BNRKREIS(BNR15)	Landkreis (numerisch), z.B: 177 bei Betrieb 09 177 148 0001
\$BNRGEMDE(BNR15)	Gemeinde (numerisch), z.B: 148 bei Betrieb 09 177 148 0001
\$MELD_FRST(ZUGA_DAT)	Berechnet Differenz aus Meldedatum und Ereignisdatum
\$MELD_FR14(ZUGA_DAT)	Differenz aus Meldedatum und Ereignisdatum, beschränkt auf 14 Tage
\$MELD_FRG1(ZUGA_DAT)	Differenz aus Meldedatum und Ereignisdatum, nach Gruppen 1
\$MELD_FRG2(ZUGA_DAT)	Differenz aus Meldedatum und Ereignisdatum, nach Gruppen 2
\$DATE_DIFF(VOK_ADAT;VOK_ADAT)	Differenz aus VOK-Datum und Datum der Anündigung
\$STORNO(SYS_BIS)	Ermittelt ob Datensatz storniert ist, also Wert des Timestamps 31.12.2100
\$AKTIV(SYS_BIS)	Ermittelt ob Datensatz aktiv ist ist, also Wert des Timestamps größer jetzt (current timestamp)
#TIEREIN(BNR15;LOM;ZUGA_DAT)	Tiereingang, default alle Ergebnisspalten
#TIEREIN(*/*)	Tiereingang, explizit alle Ergebnisspalten
#BESTREG(27.9.1999;31.12.1999/*)	Bestandsregister zum Stichtag, alle Ergebnisspalten
#BESTREG(*/*/ANF_TYP;ANF_DAT)	Default-Argument Bestand heute, nur bestimmte Ergebnisspalten

### 2.4.3. Filterfunktionen

<http://www.hi-tier.de/entwicklung/programme/batch002.html>

Da eine direkte Verknüpfung verschiedener Entitäten nicht möglich ist, erfolgt schrittweise eine Abfrage einzelner Tabellen. Anschließend werden die Teilmengen miteinander verschnitten.

Beschreibung => HITBatch-Filter.pptx; Filtersituation\_INI\_HIT.docx; Filter.xlsx)

spezielle Aktion zum Filtern von Dateien	
<b>FilterArt</b>	<p>Spezielle Angabe ob es sich bei diesem Step um einen normalen HIT-Batch Verarbeitungsschritt handelt bei dem HIT-Protokollbefehle aus dem IN-File an den Server zum Senden, Ändern oder Lesen von Daten geschickt werden oder ob nur lokal auf dem IN-File Kopier- und Filterverarbeitung durchgeführt werden soll:</p> <ul style="list-style-type: none"> <li>0: NORM - Normaler HIT-Batch Verarbeitungsschritt (Default)</li> <li>1: COPY - reines Kopieren der Datensätze aus der Inputdatei (INFILE) in die Outputdatei (OUFILE) ggf. anwenden von post-output-processing aus optionaler POP-Steuerdatei (POPFIL)</li> <li>2: FILTER - Klassischer Filter, kopiere nur Daten aus der Inputdatei (INFILE) in die Outputdatei (OUFILE) wenn die Daten auch in der Filterdatei (FILTERFILE) vorkommen, also "copy IN to OUT if exist in FILTER"</li> <li>3: INVERS - Klassischer Filter (andersrum), kopiere nur Daten aus der Filterdatei (FILTERFILE) in die Outputdatei (OUFILE) wenn die Daten auch in der Inputdatei (INFILE) vorkommen, also "copy FLT to OUT if exist in IN"</li> <li>4: NEGATIV - Negativer Filter, kopiere nur Daten aus der Inputdatei (INFILE) in die Outputdatei (OUFILE) wenn die Daten nicht in der Filterdatei (FILTERFILE) vorkommen, also "copy IN to OUT if not exist in FILTER"</li> <li>5: OR - Oder, kopiere Daten aus der Inputdatei (INFILE) bzw. der Filterdatei (FILTERFILE) nur einmal in die Outputdatei (OUFILE) wenn die Daten in einer oder beiden Dateien vorkommen. Die Dateien sollten gleich strukturiert sein. Wenn in beiden existiert wird nur INFILE genommen</li> <li>6: OR-INV - Oder, analog oben, aber wenn in beiden existiert wird nur FILTERFILE genommen</li> <li>7: XOR - Exklusiv Oder, kopiere nur Daten aus der Inputdatei (INFILE) bzw. der Filterdatei (FILTERFILE) in die Outputdatei (OUFILE) wenn die Daten nur einmal vorkommen</li> <li>8: JOIN - die beiden Tabellen "joinen", also jeden Satz des IN-File mit jedem Satz des FILTER kombinieren und in Ausgabe übernehmen falls Filter-Spalte übereinstimmen</li> <li>9: JOIN-Left - die beiden Tabellen "left outer joinen", also jeden Satz des IN-File mit jedem Satz des FILTER kombinieren und in Ausgabe übernehmen falls Filter-Spalte übereinstimmen oder in FILTER nicht gefunden</li> <li>10: JOIN-Right - die beiden Tabellen "right outer joinen", also jeden Satz des IN-File mit jedem Satz des FILTER kombinieren und in Ausgabe übernehmen falls Filter-Spalte übereinstimmen oder in INPUT nicht gefunden</li> <li>11: JOIN-Full - die beiden Tabellen "right outer joinen", also jeden Satz des IN-File mit jedem Satz des FILTER kombinieren und in Ausgabe übernehmen falls Filter-Spalte übereinstimmen oder in INPUT oder in FILTER nicht gefunden</li> </ul>
<b>Flt_Col_FLT</b>	<p>Angabe der Vergleichsspalte für die Filterdatei (FILTERFILE). Zur Ermittlung ob Daten aus der Inputdatei (INFILE) in der Filterdatei (FILTERFILE) existieren wird die ganze Zeile oder bestimmte Spalten verglichen:</p> <ul style="list-style-type: none"> <li>0: ZEILE - die ganze Zeile wird zum Vergleich herangezogen, sinnvoll insbesondere wenn nur ein Spalte vorhanden</li> <li>1-N: SPALTE - die entsprechende X. Spalte in der Datei wird zum Vergleich benutzt</li> </ul>
<b>Flt_Col_IN</b>	<p>Angabe der Vergleichsspalte für die Inputdatei (INFILE):</p> <ul style="list-style-type: none"> <li>0: ZEILE - die ganze Zeile wird zum Vergleich herangezogen, sinnvoll insbesondere wenn nur ein Spalte vorhanden</li> <li>1-N: SPALTE - die entsprechende X. Spalte in der Datei wird zum Vergleich benutzt</li> </ul>
<b>FilterFile</b>	<p>Datei mit Überschrift und Datenzeilen wenn <b>FilterArt</b> &gt; 0, (kein Default, i.d.R. LEER).</p>
<b>FilterDistinct</b>	<p>Sollen doppelt auftretende Zeilen (bezüglich Vergleichskriterium) beim Filter speziell behandelt werden:</p> <ul style="list-style-type: none"> <li>0: NORMAL - ggf. duplicate - nichts tun (Default)</li> <li>1: DISTINCT - übergehe Doubletten</li> </ul>
<b>FilterNumeric</b>	<p>Wie sollen die Vergleichskriterien (bezüglich angenommener Sortierung und Vergleichsoperation) behandelt werden:</p> <ul style="list-style-type: none"> <li>0: Nein, nicht numerisch sondern TEXT - als Zeichenkette, die Inhalte der Filterspalte(n) sind i.d.R. gleich lang, Sortierfolge muss 1,2,20,21,3 sein (Default)</li> <li>1: Ja, möglichst numerisch als ZAHL - die Inhalte der Filterspalte(n) werden falls reine Zahl numerisch verglichen,</li> </ul>

## 2.4.3.1. Beispiele

### Beispiel 1: Ermittlung von Milchviehbetrieben ohne Bestandsmilchuntersuchungen

**Wichtig:** Bei Filterfunktionen müssen die Daten gleich sortiert sein; in diesem Bsp. nach Betriebsnummer ORDER;BNR15

**[SET-1]**

**;Betriebe mit BHV1-Bestandsmilchergebnisse**

;

BEFEHL=\*2:RS/C:BHV1BEST/BNR15:BNR15;IS;05;AND;UNTS\_NDAT;BW;01.01.2011;01.01.2014;ORDER;BNR15

DATALINESSOFAR=0

INPUTAFTERSUCCESS=0

VERHALTENBEINACHFRAGE=1

**OUFIL**E=BHV1BEST.csv

LOGFILE=BHV1BEST.log

OUTAPPEND=0

LOGAPPEND=0

CSVIN=0

CSVOUT=0

LOGOUT=0

**[SET-2]**

**;Betriebe mit Produktionsrichtung Milchviehhaltung**

;

BEFEHL=\*2:RS/C:BPROD\_RICHT/BNR15:BNR15;IS;05;AND;PROD\_BIS;=31.12.2100;AND;PROD\_S\_MIK;EQ;1;ORDER;BNR15

DATALINESSOFAR=0

INPUTAFTERSUCCESS=0

VERHALTENBEINACHFRAGE=1

**OUFIL**E=PROD\_RICHT.csv

LOGFILE=PROD\_RICHT.log

OUTAPPEND=0

LOGAPPEND=0

CSVIN=0

CSVOUT=0

LOGOUT=0

**[SET-3]**

**Abgleich Rinderhalter aus PROD\_RICHT (INFILE) gegen Betriebe mit BHV1-Tankmilchuntersuchungen (FILTERFILE)**

**Das Ergebnis sind Milchviehbetriebe ohne BHV1-Bestandsmilchuntersuchungen im OUFIL**E.

;

**INFILE**=PROD\_RICHT.csv

**FILTERFILE**=BHV1BEST.csv

**OUFIL**E=Tankmilchen\_nicht\_vorhanden.csv

LOGFILE=Tankmilchen\_nicht\_vorhanden.log

OUTAPPEND=0

LOGAPPEND=0

CSVIN=0

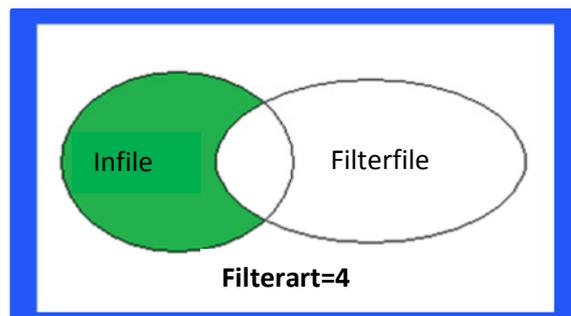
CSVOUT=0

LOGOUT=0

**FILTERART**=4

FLT\_COL\_FLT=1

FLT\_COL\_IN=1



## Beispiel 2: „Verknüpfung“ verschiedener Datentabellen, z.B. BVD-geimpfte Tiere + Tierstammdaten

### [SET-1]

#### Einzelne Impfungen pro Tier

```
BEFEHL=*2:RS/C:IMPfung/LOM;SALPHALOM(LOM/Ohrmarke);BNR15;SALPHABNR(BNR15/Betrieb);IMPf_DAT;IMPf_ZWECK;IMPf_STOFF;IMPf_CHAR;BNR15_TA;BNR15_TA;SALPHABNR(BNR15/Tierarzt):BNR15;IS;09131;AND;IMPf_ZWECK;EQ;6;AND;IMPf_DAT;BW;08.04.2013;08.05.2013;ORDER;LOM;IMPf_DAT
```

OUIFILE=Set1.out

LOGFILE=Set1.log

OUTAPPEND=0

LOGAPPEND=0

CSVIN=0

CSVOUT=0

LOGOUT=0

### [SET-2]

#### alternativ: Anzahl Impfungen pro Tier (daher nur jeweils eine Zeile)

```
BEFEHL=*3:RS:IMPfung/LOM;SALPHALOM(LOM/Ohrmarke);SCOUNT(/Anzahl):BNR15;IS;09131;AND;IMPf_ZWECK;EQ;6;AND;IMPf_DAT;BW;01.12.2013;31.12.2013;GROUP;LOM
```

OUIFILE=Set2.out

....

### [SET-3]

#### Stammdaten aus TIERSTAT zu Ohrmarken (geimpfte Tiere aus Set-2)

MELDUNG=TIERSTAT

COMMAND=RB

DATALINEPSSOFAR=0

INPUTAFTERSUCCESS=0

VERHALTENBEINACHFRAGE=1

INFILE=Set2.out

HEADERTEMPLATE=LOM;GESCHL\_R;GEB\_DATR

INPUTTEMPLATE=LOM;EQ;#[1]

OUIFILE=Set3.out

....

### [SET-4]

Filter 8: Die Tabellen IMPFUNG und TIERSTAT "joinen"=> jeden Satz des IN-File mit jedem Satz des FILTER kombinieren und in Ausgabe übernehmen falls Filter-Spalte übereinstimmen

INFILE=Set1.out

FilterFile=Set3.out

**FilterArt=8**

Flt\_Col\_FLT=1

Flt\_Col\_IN=1

OUIFILE=Set4.out

....

### [SET-5]

Filter 8: Die Tabellen IMPFUNG und TIERSTAT "joinen"=> jeden Satz des IN-File mit jedem Satz des FILTER kombinieren und in Ausgabe übernehmen falls Filter-Spalte übereinstimmen

INFILE=Set2.out

FilterFile=Set3.out

**FilterArt=8**

Flt\_Col\_FLT=1

Flt\_Col\_IN=1

OUIFILE=Set5.out

....