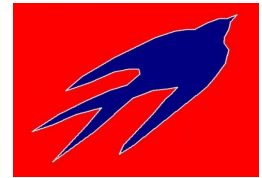


# Check - Liste: DATENBANK - AUSWAHL

Check – Liste zur Auswahl von Datenbanken.



Herausgeber: **TST**

Erste Bearbeitung: August 1998

Letzte Bearbeitung: Juni 2017

## Check - Liste: Datenbank - Auswahl

### Inhaltsverzeichnis

Allgemeine Vorgehensweise.....	3
K.O.-Kriterien.....	3
Marktrepraesentanz des DBMS Herstellers.....	3
DV-Technisches Umfeld des DBMS Herstellers.....	4
Weiterentwicklung.....	4
Preise.....	4
Technische Kriterien.....	5
1. Flexibilitaet.....	5
1.1 Erfuellungsgrad der 12 Anforderungsregeln fuer RDBMS.....	5
1.2. Datenmanipulationsprache (DML) / DML-Werkzeuge.....	5
1.2.1. Konkurrierendes Lesen und Schreiben (Synchronisation paralleler Transaktionen)...	5
1.2.2. Performance beeinflussende Faktoren.....	5
1.2.3. Abhaengigkeiten der Programme vom logischen DB-Design.....	6
1.2.4. Hardwareunterstuetzende Faktoren.....	6

1.2.5. Kompatibilitaet / Portierbarkeit.....	6
1.2.6. DML-Funktionalitaet der 3-GL (embedded) und der 4-GL (Applikationsentwicklungssprache).....	6
1.2.7. Programmelemente der 4-GL Sprache.....	7
1.2.8. DML Werkzeuge.....	7
1.3. Datendefinitionssprache (DDL) / Data Dictionary.....	7
1.4. Integration der SW-Produkte.....	8
1.5. Physische Begrenzung des Datenbanksystems.....	8
2. Netzwerkfaehigkeit.....	8
2.1. Netzwerk.....	8
2.2. Verteilte DB.....	8
3. Datensicherheit.....	9
3.1. Funktionalitaet des Backup.....	9
3.2. Funktionalitaet des Recovery.....	9
4. Datenschutz.....	9
4.1. Hierarchien des Datenschutzes.....	9
4.2. Zugriffsschutz.....	10
5. Betriebsfuehrung.....	10
5.1. Datenbankadministrator-Werkzeuge.....	10
5.2. Laden von Daten in die und aus der DB.....	10
6. Herstellerunterstuetzung.....	10
7. Endbenutzertools.....	11
Mit den besten Empfehlungen von.....	11

## Copyright

Dieses Dokument ist Eigentum von Ruy Kuhlmann.

Es darf auch ohne meiner Genehmigung vervielfaeltigt und auch dritten Personen zugaenglich gemacht werden unter der Voraussetzung, dass „TST / Ruy Kuhlmann“ als Autor erwaehnt wird.

Kontakt ueber meine Homepage: <http://www.ruynk.de>

## Allgemeine Vorgehensweise

Zunächst wird in Abhängigkeit vom zukünftigen DBMS-Umfeld ein Anforderungskatalog erstellt, der die erwarteten Anforderungen an das zukünftige DBMS enthält.

Dieser wird in 3 Abschnitte untergliedert:

- K.O.-Kriterien
- Strategische Kriterien, deren Erfüllungsgrad sich nur schwer durch eine Zahl ausdrücken lässt. (an 3. Stelle?)
- Technische Kriterien. (an 2. Stelle?)

## K.O.-Kriterien

I. A. unternehmensweite strategische Ziele. Einige könnten sein:

- bestehende bzw. gewünschte Hardware, die das DBMS unterstützen muss
- vorhandenes bzw. gewünschtes Betriebssystem, die das DBMS unterstützen muss
- finanzieller Aufwand, der benötigt wird, um das DBMS im Unternehmen einzuführen
- die bestehende Verbreitung des in Frage kommenden DBMS
- Akzeptanzprobleme bei zukünftigen Anwendern, Programmierer oder des DBA (DB-Administrators)
- Strategische Kriterien, die sich nur schwer rein technisch beurteilen lassen.

Meist sind diese Kriterien nur bedingt für die eigentliche DB-Auswahl relevant.

Zum Beispiel:

- Derzeitiger Reifegrad der Produkte (Leistungsumfang und Effizienz) der Produktpalette im geplanten Hard- und Softwareumfeld.
- Langfristige Perspektiven der Produkte unter Berücksichtigung zukünftiger Hard- und Softwareentwicklung.
- Grad der Abhängigkeit vom DBMS-Hersteller
- Existenz und Weiterentwicklungsfähigkeit des Herstellers.
- Homogenität der Produkte in Abhängigkeit des schon vorhandenen und zukünftigen Hard- und Softwareumfelds.
- Externes Personalangebot und vorhandenes Know-How für dieses DBS
- Portabilität (Unterstützung von Standards)
- Allgemeine Hersteller- und Produktdaten.

## Marktrepräsentanz des DBMS Herstellers

- Weltmärkte

- Umsatz und Umsatzentwicklung
- Anzahl Mitarbeiter (BRD und Weltweit)
- Anzahl Installationen (BRD und Weltweit)
- Anzahl Installationen auf dem Zielbetriebssystem (BRD und Weltweit)
- Zuwachs der Installationen auf dem Zielbetriebssystem (BRD und Weltweit) bezüglich der letzten Jahre
- Verteilung der Installationen auf die unterstützten Betriebssysteme
- Referenzen fuer annaehernd vergleichbare Anforderungen und Datenvolumen
- Anzahl der Versionswechsel und der zu erwartende Aufwand pro Versionswechsel
- Verfahrensweise bei Behebung von Auftretenden Fehlern im DBMS (Bugs) zwischen zwei Versionen

## DV-Technisches Umfeld des DBMS Herstellers

- Unterstützte Hardware, aufgegliedert nach Rechengroessen
- Notwendige HW und SW - Features
- Nicht oder nur bedingt unterstützte HW-Eigenschaften (z.B Multiprozessoren, gespiegelte Platten, etc.)
- Welche Betriebssysteme werden sonst noch unterstützt?

## Weiterentwicklung

- Welche ist die aktuelle Version?
- Wann ist der naechste, wann war der letzte Versionswechsel?
- Was sind die gravierenden Aenderungen der naechsten Version?
- Einflussmoeglichkeiten der Benutzer, Benutzervereinigungen
- Zugrundeliegende Programmiersprache
- Wo (oertlich) und unter welchem Betriebssystem erfolgt die Weiterentwicklung und Wartung des Systems?
- Kooperationen mit HW- und SW - Herstellern

## Preise

- Einzellizenzen pro CPU
- Evtl. sonstige Kosten fuer benoetigte Tools
- Rabatte fuer Netzwerke
- Laufende Kosten durch Versionswechsel, Hotline, zusaetzliche Manuals, Schulung fuer Programmierer, Anwender und DBA, Beratungen
- Laufzeit Versionen (nur fuer DBMS ohne Entwicklertools)

## Technische Kriterien

### 1. Flexibilitaet

#### 1.1 Erfuellungsgrad der 12 Anforderungsregeln fuer RDBMS

- Informationsregel
- Garantierter Zugriff
- Systematische Behandlung von fehlenden Informationen
- Dynamischer "Online Katalog", basierend auf dem relationalen Modell
- Allesumfassende Sprache
- Datenaenderungen durch Views
- High-Level-Insert, -Update und -Delete
- Physische Datenunabhaengigkeit
- Logische Datenunabhaengigkeit
- Integritaetsunabhaengigkeit
- Verteilungsunabhaengigkeit
- Unterwanderungsverbot

#### 1.2. Datenmanipulationssprache (DML) / DML-Werkzeuge

##### 1.2.1. Konkurrierendes Lesen und Schreiben (Synchronisation paralleler Transaktionen)

- Daten aendern, waehrend andere lesen
- Konsistentes Lesen, waehrend ander updaten
- Ist Update ohne Setzen eines Locks moeglich? (sinnvoll bei Stand-Alone-Dateien/Tabellen)
- Lock auf Feldebene, Satzebene, Pageebene, Tabellenebene, Datenbankebene, Systemebene
- Automatisches Erkennen von Deadlocks
- Aufloesen von Deadlocks

##### 1.2.2. Performance beeinflussende Faktoren

- Kann I/O Cachegroesse variiert werden?
- Cache-Verwaltung (wird nach einem Update der Puffer im I/O Cache sofort physisch geschrieben oder moeglichst lange im Cache gehalten?)
- Koennen Saetze physisch als Cluster gespeichert werden?
- Kann der Datenkompressionsvorgang ein- und abgeschaltet werden?
- Kann der Schluesselkompressionsvorgang ein- und abgeschaltet werden?
- Koenenn mehrere Felder als ein Schluessel definiert werden?
- Kann der Schluessel als eindeutig definiert werden?
- Kann ein phonetischer Schluessel definiert werden
- Direktzugriff ueber:

- B-Baum
- ISAM
- Datenbank-Schlüssel
- Hash
- invertierte Listen
- Werden Datenbankstatements vorübersetzt? (besonders wirksam bei Programmschleifen und OLTP-Anwendungen)
- Ist der Code reentrant? (d.h. wird gleicher DBMS-Code im Hauptspeicher mehrmals gehalten; dies beeinflusst die effektive Auslastung des Hauptspeichers)
- Wieviele prozesse werden durch das DBMS kreiert?
- Wie performance-effektiv wird physischer Speicherplatz verwaltet? (wie performance-effektiv wird nach Datenaenderung in der physischer Datenseite automatisch reorganisiert?)
- Ausbalancierung der Indexbaeume (werden in regelmaessigen Abstaenden die Indexbaeume automatisch reorganisiert?)
- Wie effektiv ist die Speicherplatzverwaltung der Datenbankdaten auf Festplatte? (bezuglich Plattenplatzbedarf und Performance)
- Automatisches Optimieren der Suchabfrage
- Automatisches Optimieren der Suchabfrage, anzeigbar (anzeigbar bedeutet, der gefundene Weg des Optimierers kann dargestellt werden)

### **1.2.3. Abhaengigkeiten der Programme vom logischen DB-Design**

- Wie "nicht-prozedural" ist die mitgelieferte 4GL?
- Wie nicht-prozedural sind die DML-Statements in der 3GL Sprache?
- Welche Programmiersprachen werden unterstuetzt?

### **1.2.4. Hardwareunterstuetzende Faktoren**

- Inwieweit werden Multiprozessor-Computer unterstuetzt? (kann der Datenbankprozess gleichzeitig auf mehreren Knoten oder nur auf einem laufen; inwieweit werden dabei die Datenbankfunktionen unterstuetzt?)
- Inwieweit wird das Prinzip "Gespiegelte Platten" unterstuetzt?

### **1.2.5. Kompatibilitaet / Portierbarkeit**

Kompatibilitaet zu ANSI-SQL und OPEN-SQL (in der 4-GL und 3-GL embedded Sprache)

### **1.2.6. DML-Funktionalitaet der 3-GL (embedded) und der 4-GL (Applikationsentwicklungssprache)**

- Unterstuetzte Datentypen
- Koennen neue, eigene Datentypen definiert werden?
- Nullwerte (ist die Nichtexistenz von Werten durchgaengig realisiert?)
- Konvertierungsfunktionen fuer alle unterstuetzte Datentypen, z.B. Char, Date, Money...
- Arithmetische Anweisungen fuer
  - trigonometrische Funktionen
  - Gruppenfunktionen
  - Grundrechenarten
  - zusaetzliche Datentypen wie Date, Money, BLO etc.

- Existieren Zeichenkettenfunktionen?
- Unterstützung der verschiedenen JOIN-Arten (Inner Join, Outer Join, Auto Join etc.)
- Mengenoperationen bzgl. relationaler Algebra (Bildung von Vereinigungs-, Durchschnitts- und Differenzmengen)
- Unterprogrammverarbeitung (3-GL und 4-GL)
- Mischung von 3-GL und 4-GL Modulen
- Welche DML-Statements können auf Views angewandt werden?
- Unterstützung von Views (wieviele Hierarchien, Check Option, Kontrolle bei Drop View und Drop Table)
- Inwieweit ist die Anzahl der Tabellen pro Abfrage begrenzt?
- Grenzen in der Schachtelungstiefe pro Abfrage (Subselects)
- Unterstützung von objektorientierten Programmiersprachen und Datentypen

### 1.2.7. Programmelemente der 4-GL Sprache

- Variablendefinitionen mit eigenen Datentypen, globale / lokale Variable
- Prozedurendefinition (Welche Arten der Parameterübergabe)
- Schleifenverarbeitung
- Sprunganweisungen
- Können bedingte Anweisungen formuliert werden? (if ... then ... else ...)
- Wird Rekursion unterstützt?

### 1.2.8. DML Werkzeuge

- Funktionalität des Report-Writers
- Wie flexibel ist die Schnittstelle zwischen Maske und DB?
- Freizügigkeit bei der Erstellung des Masken-LayOuts
- Kann die Maske interaktiv erstellt werden?
- Existiert eine default-Maske?
- Ist die Maske in der 3-GL uneingeschränkt verwendbar?
- Existiert eine HELP-Funktion (Mehrsprachig) ?
- Existiert eine interaktive Testeinrichtung (debugging) ?
- Funktionalität der Menüerstellung
- Funktionalität von Graphikfunktionen
- Funktionalität von Textverarbeitungsfunktionen
- Existiert ein leicht zu handhabendes Abfrage-Tool?
- Ist 4-GL im Batch- und Online-Betrieb ablauffähig?
- Funktionalität der Programm- und Maskenverwaltung (Versionskontrolle, Änderungsprotokolle, Rekonstruktionen von Versionen)
- Unterstützung von grossen Entwicklergruppen

## 1.3. Datendefinitionssprache (DDL) / Data Dictionary

- Sind DDL-Statements im laufenden Betrieb anwendbar? (ist dies möglich, ohne dass die DB teilweise oder ganz inaktiviert werden muss?)
- Inwieweit ist DDL in der 3-GL verwendbar?
- Hinzufügen, Ändern von Typ oder Namen sowie Löschen von Feldern/Spalten (ohne Reorganisationsmassnahmen)
- Feld mit Integritätsbedingungen belegen (im Data Dictionary definiert; nicht gemeint sind z.B. Plausibilitätsprüfungen in programm / Maske)

- Kann Tabelle als Ergebnis einer Abfrage angelegt werden? (muss Tabelle explizit im Sinne von "CREATE" definiert werden?)
- Kann Tabellename / Viewname geändert werden?
- Ist Data Dictionary mit vorhandener 4-GL auswertbar?
- Können Synonyme vergeben werden? (betrifft Felder, Schlüssel, Tabellen im Data Dictionary)
- Können mehrere Versionen von Tabellen gehalten werden?
- Werden temporäre Tabellen unterstützt?

## 1.4. Integration der SW-Produkte

- Integration der SW-Produkte untereinander
- Integration mit Betriebssystem-Komponenten des HW-Herstellers
- Wie weit ist die Benutzeroberfläche einheitlich?
- Unterstützung von mehrsprachigen Applikationen

## 1.5. Physische Begrenzung des Datenbanksystems

- Inwieweit ist die Anzahl der Knoten im Netzwerk begrenzt?
- Unbegrenzte DB-Benutzeranzahl?
- Inwieweit ist die Anzahl der DB pro Knoten begrenzt?
- Inwieweit ist die Anzahl der Schlüssel pro Tabelle begrenzt?
- Maximale
  - Feldlänge
  - Schlüssellänge
  - Satzlänge
  - Namenslänge
  - Felderanzahl pro Satz
- Unbegrenzte Anzahl von Tabellen pro DB?
- Maximale Programmgröße (Kb oder Zeilen) der 3-GL und 4-GL

# 2. Netzwerkfähigkeit

## 2.1. Netzwerk

- Unterstützte Kommunikationsprotokolle
- Fernadministration (Start-Stop-Möglichkeiten für die DB, Remote ohne explizites Einloggen, zentrale Steuerung von einem Knoten aus)
- Unterstützung heterogener Knoten (welche Interfaces zu anderen Datenbanksystemen existieren)

## 2.2. Verteilte DB

- Erfüllungsgrad der "Date Regeln"
  1. Lokale Autonomie
  2. Kein Verlass auf einen zentralen Knoten



3. Unterbrechungsfreien Betrieb
  4. Standortunabhaengigkeit
  5. Fragmentierungsunabhaengigkeit
  6. Replikationsunabhaengigkeit
  7. Verteilte Query
  8. Verteiltes Transaktionsmanagement
  9. Hardwareunabhaengigkeit
  10. Betriebssystemunabhaengigkeit
  11. Netzwerkunabhaengigkeit
  12. DBMS-Unabhaengigkeit
- Welche Konsistenzsicherheiten werden geboten?
  - Daten-Extraktion (Kopierfunktionen von Datenbankbereichen von einem Knoten zum anderen)
  - Lesende Verknuepfung mehrerer Knoten in einer Transaktion (nur Ergebnisdaten gehen ueber das Netz, nicht Tabellen vollstaendig)
  - Automatische parallele Abarbeitung einer Transaktion auf verschiedenen Knoten (wird die Abfrage vom Optimierer automatisch zerlegt und parallel abgearbeitet?)
  - Scheduling-Funktionen (inwieweit koennen sowohl selbstaendige, sowie regelmaessig anlaufende, verteilte Transaktionen im Data Dictionary definiert werden?)
  - Updaten mehrerer Knoten in einer Transaktion (Mehr-Phasen-Commit)
  - Funktionalitaet der globalen Lockmechanismen (Deadlock - Erkennung)

### 3. Datensicherheit

#### 3.1. Funktionalitaet des Backup

- Export / Import auf Datenbank-Ebene, Tabellenebene
- Sicherung der DB waehrend des laufenden Betriebs (wichtig fuer 24-Stundenbetrieb)

#### 3.2. Funktionalitaet des Recovery

- Automatisches Recovery nach Systemabbruch (wichtig fuer operatorlosen Betrieb)
- Optionales Recovery nach Systemabbruch (kann das automatische Recovery abgeschaltet werden?)
- Mehrfache Log-Dateien (kann die Log-Datei (Journal) gleichzeitig z.B. auf Platte und Band geschrieben werden?)
- Komprimierungsrate der Log-Datei (beeinflusst Speicherplatzbedarf und Wiederanlaufzeit)
- Auswertbarkeit der Log-Datei (After Image) mittels 4-GL

### 4. Datenschutz

#### 4.1. Hierarchien des Datenschutzes

Datenschutz auf

- Systemebene
- Datenbankebene
- Tabellenebene
- Pageebene
- Satzebene
- Feldebene

## 4.2. Zugriffsschutz

- Weitergabe von Berechtigungen (kann das Recht, Zugriffsberechtigungen zu vergeben, weitergegeben werden?)
- Koennen Berechtigungen fuer Programme vergeben werden?
- Zugriff von Menuefunktionen benutzerabhaengig?
- Kann die Zugriffsberechtigung Benutzergruppen zugeordnet werden?
- Kann der Zugriff von einer Terminaladresse abhaengig gemacht werden?
- Kann die Tabellenexistenz geheimgehalten werden?
- Koennen Daten chiffriert werden?
- Zugriffsprotokollierung (Auditing)
- Aenderungsaufwand fuer den Datenschutz

## 5. Betriebsfuehrung

### 5.1. Datenbankadministrator-Werkzeuge

- Existieren Werkzeuge fuer das DB-Design?
- Start-Stop-Moeglichkeiten fuer die DB, lokal
- Vorhandensein und Funktionalitaet des Performance-Monitors
- Auswertbarkeit der Daten des Performance-Monitors (sind die Daten des Performance-Monitors mit den vorhandenen 4-GL-Moeglichkeiten und / oder graphisch auswertbar?)
- Aufwand fuer Anpassung von Systemparameter / Konfigurationen
- Laufender Aufwand fuer DBA

### 5.2. Laden von Daten in die und aus der DB

- Werden alle Datentypen unterstuetzt?
- Performance des Laden(s)vorgangs
- Wahlweise Laden als eine Transaktion

## 6. Herstellerunterstuetzung

- Qualitaet der Dokumentation
- Onlineverfuegbarkeit der Dokumentation
- Support durch den Hersteller

## 7. Endbenutzertools

- Qualitaet der Endbenutzertools
- Integration der Endbenutzertools in das Gesamtsystem
- Schnittstellen zu PC-Tools
- Dokumentation der Endbenutzertools (endbenutzergerecht)

Mit den besten Empfehlungen von

### Unternehmen

- **AMTRS** Systemanalyse – Ruy Kuhlmann
- **"ruynk" Projektkoordination** – R. C. N.-Kuhlmann Projektmanager
- **IT-Wissen "Ryusui"** - Kurse und Seminare
- **TST Tsubame Software Tools** - Business Support Software Solutions

### URLs:

- [www.amtrs.de](http://www.amtrs.de) ---> AMTRS: Analysieren, Modellieren, Trainieren, Reorganisieren, Systematisieren
- [www.ruynk.com](http://www.ruynk.com) ---> Projektmanagement, Projektkoordination, FlePA
- [www.ruynk.de](http://www.ruynk.de) ---> Private Homepage
- [www.ryusui.de](http://www.ryusui.de) ---> IT-Wissen – Kurse und Seminare
- [www.tsubame.de](http://www.tsubame.de) ---> Business Support Software Solutions

### Blogs:

- [RUYNK: Über Projektmanagement und FlePA](#)
- [Tsubame SW: Über SW-Entwicklung](#)