

Binary Large Object (BLOB) in Datenbanken

Speicherung großer, unstrukturierter Daten

Was ist ein BLOB?

- BLOB = *Binary Large Object*
- Speichert große binäre Datenmengen
- Typischerweise **nicht textbasiert**, sondern z. B.:
 - Bilder (.jpg, .png)
 - Videos (.mp4)
 - Audiodateien (.mp3)
 - PDF-Dokumente



Eigenschaften von BLOBs

- Daten werden als binäre Sequenz gespeichert
- Keine Interpretation durch das Datenbanksystem
- Kann sehr groß sein (mehrere GB, abhängig vom DBMS)
- BLOBs sind oft nicht durchsuchbar (im Gegensatz zu Text)



Wann verwendet man BLOBs?

- Wenn Dateien **innerhalb der Datenbank** gespeichert werden sollen
- Wenn Daten mit der Anwendung eng verknüpft sind
- Beispiele:
 - **Content-Management-Systeme (CMS)**
 - **Medizinische Archive**
 - **Digitale Bibliotheken**



Vorteile und Nachteile

Vorteile	Nachteile
Einfachere Sicherung (alles in der DB)	Größere Datenbankgröße
Transaktionen möglich	Zugriff langsamer als auf Dateisystem
Datenbank-Zugriffsrechte nutzbar	DB-Backups werden aufwendiger

Datentypen & SQL

- In SQL: BLOB, MEDIUMBLOB, LONGBLOB (je nach Größe)
- Beispiel:

```
CREATE TABLE dokumente (  
  id INT PRIMARY KEY,  
  datei BLOB  
);
```



Zugriff auf BLOBs

- BLOBs werden meist über **Programmiersprachen** verwaltet:
 - Java: `PreparedStatement.setBlob()`
 - Python: `cursor.execute(..., (binary_data,))`
- Häufig mit Streams gelesen/geschrieben

Alternativen zur BLOB-Nutzung

- Speicherung im Dateisystem, Verweis in der DB
- Vorteile:
 - Schnellere Dateizugriffe
 - Geringere DB-Größe
- Nachteile:
 - Höherer Wartungsaufwand
 - Sicherheitsaspekte

Zusammenfassung

- BLOBs ermöglichen die Speicherung großer Binärdaten direkt in der DB
- Praktisch bei vollständiger Datenintegration
- Technisch anspruchsvoller als Textdaten
- Entscheidung abhängig vom Anwendungsfall

Weiterführende Ressourcen

- Read / Write BLOBS in C#:
 - [Read / Write BLOBs from / to SQL Server using C# .NET DataReader](#)
- YouTube: "What is a BLOB in Database?"
- MySQL Doku: <https://dev.mysql.com/doc/>
- PostgreSQL Doku: <https://www.postgresql.org/docs/>