DLBDSPBDM01_D Projekt: Data-Mart-Erstellung in SQL

Tutorin: Anna Androvitsanea

Portfolio

Martina Özkan / Business Intelligence (B.Sc.) / Matrikel-Nr. 9227086 / martina.oezkan@iu-study.org

3. Finalisierungsphase – Dokumentation des Datenbankmanagementsystems

Folgende Dokumentation bietet ein Einblick in die wichtigsten Funktionalitäten des ausgewählten Datenbankmanagementsystems PostgreSQL, der Datenbankstruktur sowie relevanten Metadaten.

Funktionalitäten von PostgreSQL

PostgreSQL, als ein objektrelationales Datenbankmanagementsystem, unterstützt sowohl objektrelationale als auch relationale Datenmodelle, was auch die Abbildung von komplexen Datenstrukturen ermöglicht.

Durch seine Erweiterbarkeit bietet den Benutzern die Möglichkeit, auch eigene Datentypen, Indexmethoden oder Funktionen zu erstellen.

Die strenge Einhaltung von SQL-Standards verbessert die Kompatibilität mit anderen Systemen.

Die zuverlässige Datenintegrität wird durch Unterstützung von ACID (Atomicity, Consistency, Isolation, Durability = Atomität, Konsistenz, Isolation, Langlebigkeit) gewährleistet.

Dank Implementierung der MVCC Technik (Multiversion Concurrency Control) wird den Benutzern gleichzeitiger Zugriff auf die Daten ermöglicht, ohne sich gegenseitig zu blockieren oder die Datenbankkonsistenz zu gefährden.

Eine leistungsfähige Speicherung und Verwaltung von Metadaten ist ein zentrales Merkmal von PostgreSQL. Metadaten beschreiben Informationen über Datenquellen, Datenstrukturen, Beziehungen sowie Zugriffsrechten. PostgreSQL bietet nicht nur eine effiziente Speicherung der Metadaten, sondern auch deren gezielte Analysen. Bei datengetriebenen Anwendungen ist eine effektive Metadatenverwaltung für die Transparenz und Nachvollziehbarkeit besonders relevant. Interne Metadaten verwaltet PostgreSQL in systemeigenen Katalogtabelle, Systemkatalogen.

Die Systemkataloge sind der Ort, an dem ein relationales Datenbankmanagementsystem Schema-Metadaten speichert – zum Beispiel Informationen über Tabellen und Spalten sowie interne Verwaltungsinformationen (The PostgreSQL Global Development Group, o. J., S. 2303).

Die Anwender haben die Möglichkeit, über die SQL-Befehle, tiefere Einblicke in die Datenbankstruktur zu erlangen. Außerdem ist es möglich anwendungsspezifische Metadaten zu definieren.

Metadaten und Systemvolumen

- Anzahl der Tabellen = 15

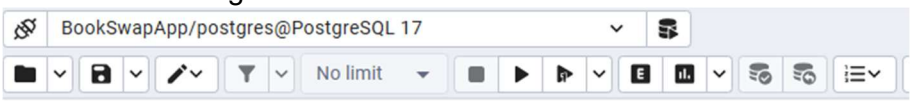
Query | Query History

```sql
--Anzahl der Tabellen im aktuellen Hauptschema _ public-Schema
SELECT COUNT(*)
FROM information_schema.tables
WHERE table_schema = 'public';
```

Data Output | Messages | Notifications

	count bigint
1	15

- Anzahl der Einträge



Query | Query History

```sql
--Schätzung der aktuell vorhandenen Zeilen pro Tabelle (n_Live_tup)
--Für exakte Werte muss COUNT(*) verwendet werden, bei großen Tabellen kann
--es aber zu Performanceschwierigkeiten kommen
SELECT
  relname AS table_name,
  n_live_tup AS row_estimate
FROM pg_stat_user_tables
ORDER BY n_live_tup DESC;
```

Data Output | Messages | Notifications

	table_name name	row_estimate bigint
1	users	11
2	admins	11
3	reservations	11
4	books	11
5	registrations	10
6	authors	10
7	comments	10
8	searches	10
9	loans	10
10	publishinghouses	10
11	shippingoptions	10
12	valuations	10
13	addresses	10
14	categories	10
15	languages	10
16	available_books	4

- Datenbankgröße = 9051 kB

Query | Query History

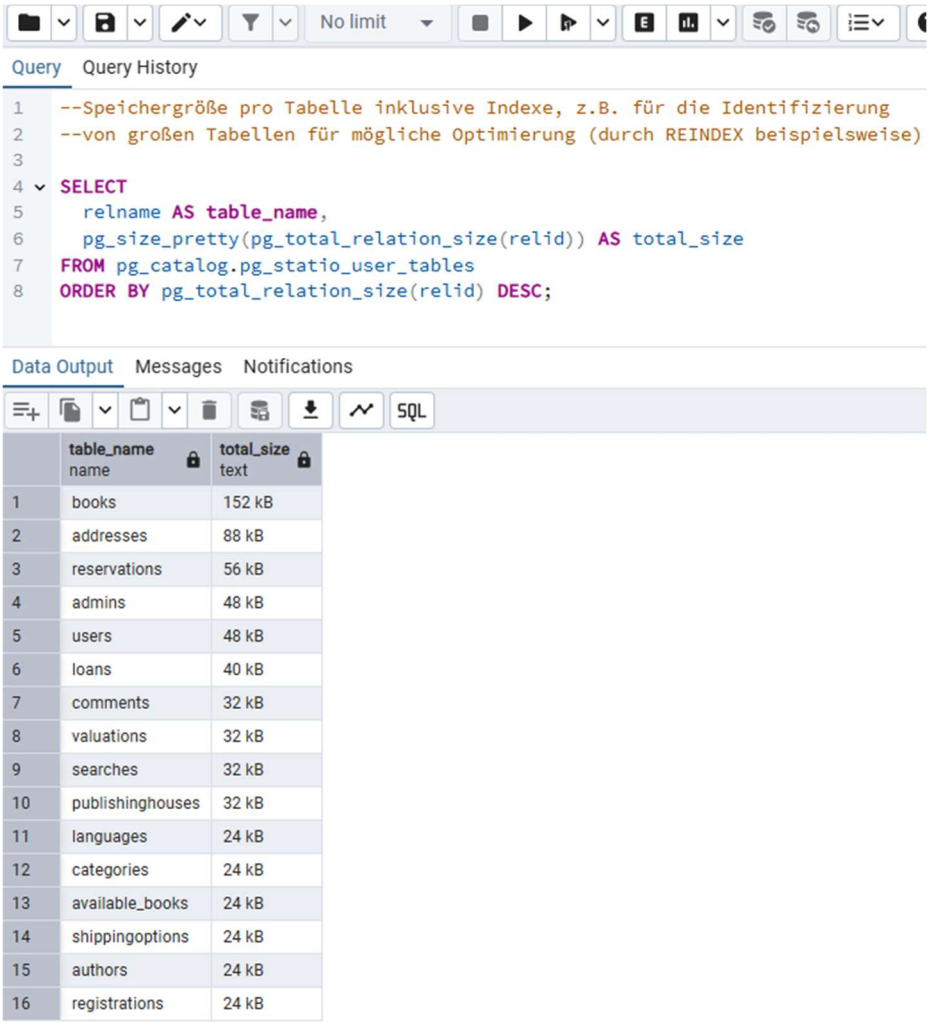
```sql
--Gesamtgröße der Datenbank in lesbaren Format

SELECT pg_size_pretty(pg_database_size(current_database())) AS db_size;
```

Data Output | Messages | Notifications

	db_size text
1	9051 kB

- Weitere Metadaten
  - Speichergröße pro Tabelle inkl. Indexe



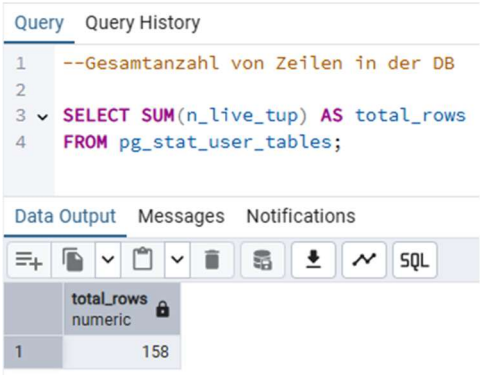
Query | Query History

```
1  --Speichergröße pro Tabelle inklusive Indexe, z.B. für die Identifizierung
2  --von großen Tabellen für mögliche Optimierung (durch REINDEX beispielsweise)
3
4  SELECT
5    relname AS table_name,
6    pg_size_pretty(pg_total_relation_size(relid)) AS total_size
7  FROM pg_catalog.pg_statio_user_tables
8  ORDER BY pg_total_relation_size(relid) DESC;
```

Data Output | Messages | Notifications

	table_name name	total_size text
1	books	152 kB
2	addresses	88 kB
3	reservations	56 kB
4	admins	48 kB
5	users	48 kB
6	loans	40 kB
7	comments	32 kB
8	valuations	32 kB
9	searches	32 kB
10	publishinghouses	32 kB
11	languages	24 kB
12	categories	24 kB
13	available_books	24 kB
14	shippingoptions	24 kB
15	authors	24 kB
16	registrations	24 kB

  - Gesamtanzahl von Zeilen meiner Datenbank = 158



Query | Query History

```
1  --Gesamtanzahl von Zeilen in der DB
2
3  SELECT SUM(n_live_tup) AS total_rows
4  FROM pg_stat_user_tables;
```

Data Output | Messages | Notifications

	total_rows numeric
1	158

- Analyse- und Vacuum-Zeitpunkte pro Tabelle

Query | Query History

```
--Um Performance-Überwacheung oder Entscheidung für Optimierungsmaßnahmen zu erleichtern,
--kann eine Abfrage zur Vacuum- oder Analyse-Zeitpunkt durchgeführt werden:
--VACUUM - Durch Updates oder Delets entstandene "tote" Tupel werden entfernt
--ANALYZE - Statistikdaten über die Tabelleninhalte werden gesammelt
SELECT
  relname AS table_name,
  last_vacuum,
  last_autovacuum,
  last_analyze,
  last_autoanalyze
FROM pg_stat_user_tables;
```

Data Output | Messages | Notifications

	table_name name	last_vacuum timestamp with time zone	last_autovacuum timestamp with time zone	last_analyze timestamp with time zone	last_autoanalyze timestamp with time zone
1	shippingoptions	[null]	[null]	[null]	[null]
2	valuations	[null]	[null]	[null]	[null]
3	addresses	[null]	[null]	[null]	[null]
4	categories	[null]	[null]	[null]	[null]
5	registrations	[null]	[null]	[null]	[null]
6	authors	[null]	[null]	[null]	[null]
7	books	[null]	[null]	[null]	[null]
8	admins	[null]	[null]	[null]	[null]
9	available_books	[null]	[null]	[null]	[null]
10	comments	[null]	[null]	[null]	[null]
11	users	[null]	[null]	[null]	[null]
12	searches	[null]	[null]	[null]	[null]
13	loans	[null]	[null]	[null]	[null]
14	publishinghouses	[null]	[null]	[null]	[null]
15	reservations	[null]	[null]	[null]	[null]
16	languages	[null]	[null]	[null]	[null]

- Abfrage für die Größe der Tabelle Books

Größe der Tabelle inkl. aller Indexe = „pg_total_relation_size“

Größe der Tabelle ohne Indexe = „pg_relation_size“

Query | Query History

```
--Abfrage der Größe der Tabelle Books mit und ohne Indexe

SELECT
  pg_size_pretty(pg_relation_size('BOOKS')) AS nur_tabelle,
  pg_size_pretty(pg_total_relation_size('BOOKS')) AS tabelle_inkl_indexe;
```

Data Output | Messages | Notifications

	nur_tabelle text	tabelle_inkl_indexe text
1	8192 bytes	152 kB

Literaturverzeichnis

The PostgreSQL Global Development Group. (o. J.). *PostgreSQL 17.4 Documentation*.