

Abbildung 1 Titelblatt (Medienakademie, kein Datum)

MODULARBEIT M141

Version 1.2

DATENBANKUMGEBUNG: ONLINESHOP TECHELECTRONICS

Entwicklung eines Online-Shops für Elektronikartikel mit MySQL-Datenbank zur Verwaltung von Bestellungen und Lagerbeständen.

Fabian Manser – Jonas Sieber – Bojan Maljuric
M141

Inhalt

1	Projektbeschreibung	3
1.1	Ausgangslage	3
1.2	Auftrag	3
1.3	Weitere Ideen	3
2	Evaluation DBMS	4
2.1	Begriff Volltextsuche, Transaktion, Replikation	4
2.2	Vergleich verschiedener DBMS	4
2.3	Begründeter Entscheid für ein DBMS	5
3	Technische Dokumentation	5
3.1	Installation	5
3.2	Schnittstellenbeschreibung	8
	Vergleich Tabellentypen	9
3.3	Datenbank einrichten	9
	Benutzerkonten einrichten	11
	Rechtevergabe Views und Stored Procedures	12
4	Testdokumente	13
4.1	Berechtigung testen 4-6:	13
4.2	Views Testen:	13
4.3	Stored Procedure Testen:	14
4.4	Backup testen:	14
4.5	Testfall Logdateien:	15
4.6	Performanceverbesserung	15
4.7	SQL-Abfrageoptimieren:	15
5	Betriebsanweisungen	16
5.1	Datenbank Starten und Stoppen	16
	MySQL Workbench	16
	Konsole	17
5.2	Backup mit MySQL Dump:	17
5.3	Übersicht über Log-dateien:	18
6	Reflexion	19
6.1	Fabian Manser:	19
6.2	Bojan Maljuric:	19
6.3	Jonas Sieber:	19
7	Anhänge	20
7.1	Abbildungsverzeichnis	20

7.2	Tabellenverzeichnis.....	20
7.3	Literaturverzeichnis.....	21
7.4	Arbeitsjournal Jonas Sieber.....	22
7.5	Arbeitsjournal Fabian Manser	22
7.6	Arbeitsjournal Bojan Maljuric	23

1 Projektbeschreibung

1.1 Ausgangslage

Unsere IT-Firma, spezialisiert auf die Entwicklung von Datenbanklösungen, hat einen Auftrag von einem KMU erhalten, einen Onlineshop für Elektronikprodukte zu entwickeln. Der Kunde beabsichtigt, mit dem Onlineshop eine breite Produktpalette anzubieten und eine reibungslose Verwaltung und Optimierung der Lager- und Bestellprozesse zu erreichen. Diese Datenbanklösung soll dem Kunden ermöglichen, Lagerbestände, Bestellungen, Zahlungsabwicklungen und Kundendaten zentral zu verwalten und zu kontrollieren.

Der Onlineshop, der unter dem Namen "**Techelectronics**" laufen soll, soll mit einem Web-Frontend für Kunden und einem Backend für die Verwaltung verbunden sein. Unsere Firma ist innerhalb des Projekts für die Backenddatenbank zuständig und mit Views und Stored Procedures das Front-End zu unterstützen.

1.2 Auftrag

- **Produktverwaltung:** Tabelle mit Produkten in Kategorien, filterbar und sortierbar.
- **Benutzerkonten:** Anmeldesystem, inkl. Passwortschutz
- **Warenkorb und Bestellabwicklung:** Speichern und Bearbeiten von Warenkorbinhalten und sichere Bestellfunktion.
- **Bestandskontrolle:** Echtzeit-Überblick über den Lagerbestand.
- **Zahlungsverwaltung:** Nachverfolgung und Kontrolle der Zahlungseingänge.
- **Datenbanksystem:** Ein schneller leistungsfähiges, relationales Datenbankmodell.

1.3 Weitere Ideen

Um den Onlineshop zu erweitern und langfristig zu optimieren, könnten folgende Funktionen hinzukommen:

- **Analyse-Funktionen** zur Kundenverhaltensauswertung.
- **Automatisierte Lagerprozesse** zur besseren Bestandskontrolle.
- **Erweiterte Benutzerrollen** zur Verwaltung der Zugriffsrechte.
- **Backupkonzept** zum idealen Schutz vor Datenverlust.

2 Evaluation DBMS

2.1 Begriff Volltextsuche, Transaktion, Replikation

Volltextsuche: Sie ermöglicht es, nach Wörtern oder Phrasen im gesamten Text zu suchen, nicht nur in Titeln. Nützlich in grossen Dokumenten oder Datenbanken, wie bei Suchmaschinen.¹

Transaktion: Eine Reihe von Befehlen in einer Datenbank, die entweder vollständig ausgeführt oder bei Fehlern rückgängig gemacht werden. Sie stellt sicher, dass die Daten immer konsistent bleiben.²

Replikation: Das Kopieren von Daten zwischen mehreren Servern, um Ausfallsicherheit und Verfügbarkeit zu gewährleisten, falls einer der Server ausfällt. Zudem dient Replikation auch zur Lastverteilung und zur Leistungsoptimierung.³

2.2 Vergleich verschiedener DBMS

Eigenschaft	MYSQL	Microsoft SQL Server	PostgreSQL
Hersteller	Oracle Corporation ⁴	Microsoft	PostgreSQL Global Development Group
Aktuelle Version	8.4.3 (9.1.0 Innovation)	2022	17
DB-Typ	Relational	Relational	Objektrelationales
Lizenz	Open - Source/Kommerziell	Proprietär	Open-Source
Typische-Anwendung	Webapplikationen	Unternehmensanwendungen	Wissenschaftliche Datenbanken
OS	Plattformübergreifend	Windows, Linux	Plattformübergreifend
Hardwareanforderungen	Moderat	Höher	Moderat
Kosten	Kostenlos bis Enterprise	Kostenpflichtig	Kostenlos
Management-Tool(s)	MySQL Workbench, Heidi SQL, phpMyAdmin	SQL Server Management Studio ⁵	pgAdmin
Volltextsuche	Ja	Ja	Ja
Referentielle Integrität	Ja	Ja	Ja
Transaktionen	Ja	Ja	Ja
Partitionierung/Replikation	Ja	Ja	Ja
Vorteile	Leicht zu bedienen, vielseitig ⁶	Leistungsstark, skalierbar ⁷	Skalierbar, robust
Nachteile	Begrenzte Skalierbarkeit	Hohe Kosten	Komplexere Konfiguration

Tabelle 1 Übersicht DBMS

¹ (persis, 2024)

² (Boekhoven, 2022)

³ (divers, wikipedia, 2024)

⁴ (divers, Wikipedia, 2024)

⁵ (AWS, 2024)

⁶ (Dadhich, 2024)

⁷ (Neovera, 2024)

2.3 Begründeter Entscheid für ein DBMS

Für die Datenbank von Techelectronics haben wir uns aus folgenden Gründen für MySQL entschieden:

- MySQL bietet eine hohe Geschwindigkeit und kann grosse Datenmengen effizient verarbeiten, was es ideal für unsere Anforderungen macht.
- Die relationale Struktur von MySQL ermöglicht eine logische und konsistente Speicherung unserer Daten, was die Verwaltung erleichtert.
- MySQL unterstützt sowohl vertikale als auch horizontale Skalierung, sodass wir bei Bedarf die Leistung problemlos anpassen können.
- Die Kompatibilität mit verschiedenen Programmiersprachen und Plattformen erleichtert die Integration in unsere bestehenden Projekte.
- MySQL kann Kostenlos in der Community Edition als auch in der Enterprise Edition genutzt werden, was es auch für kleinere Firmen attraktiv und Privatverwendung attraktiv macht.⁸
- Die grosse Open-Source-Community sorgt für kontinuierliche Verbesserungen und regelmässige Updates, was uns Sicherheit und Unterstützung bietet.⁹

3 Technische Dokumentation

3.1 Installation

Für das Projekt Techelectronics haben wir uns für die neueste LTS-Version (Long Term Support) des MySQL Community Servers (8.4) entschieden und diese auf einem PC installiert. Um die Datenbankverwaltung zu vereinfachen, haben wir das eigene Tool MySQL Workbench (8) eingesetzt. Anschliessend haben wir eine lokale Instanz erstellt, die als Testumgebung diene. Dadurch konnten wir die Datenbank testen und konfigurieren, ohne die Produktionsdatenbank zu gefährden. Nach der Installation habe ich die folgenden Konfigurationen überprüft und angepasst:

Der Status des Servers kann unter dem Menüpunkt **Server -> Server Status** überprüft werden. Wenn dort **«RUNNING(1)»** angezeigt wird, bedeutet dies, dass der Server aktiv ist. In dieser Ansicht wird zudem die installierte Version im oberen Bereich unter **Version(2)** angezeigt. Um den verwendeten Port zu überprüfen, kann man die Zeile **Port(3)** einsehen, die standardmässig auf 3306 eingestellt ist.

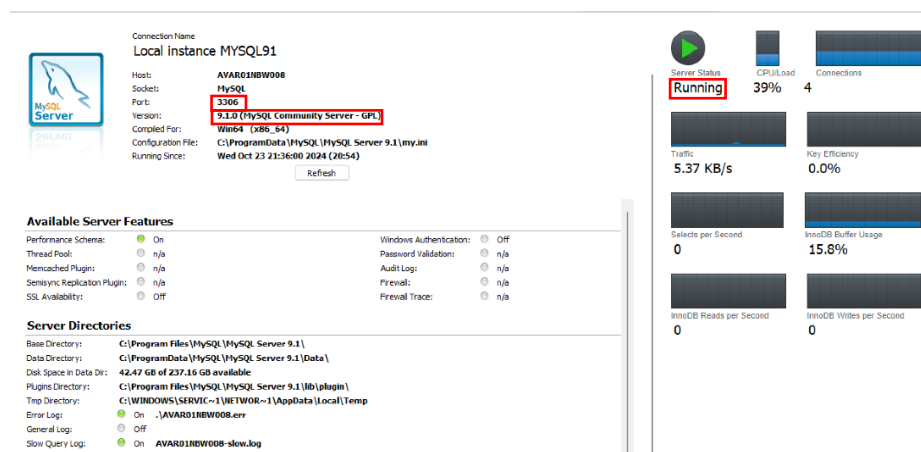


Abbildung 2 Port, Version und Server Status anzeigen

⁸ (Wiggers, 2023)

⁹ (RemoteScout24, 2023)

Angemeldeten Benutzer können mit folgendem Pfad überprüft werden: **Server > User and Privileges**.

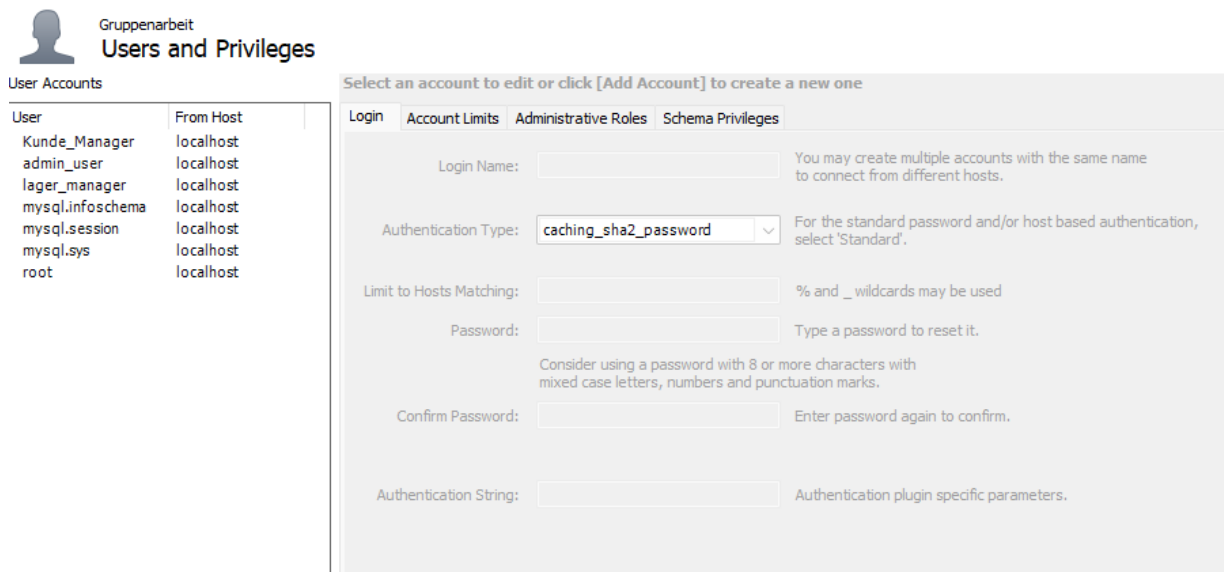


Abbildung 3 Alle Benutzer anzeigen lassen

Zur Überprüfung der Funktionalität des Servers wurde ausserdem eine Testdatenbank (**mytest**) erstellt:

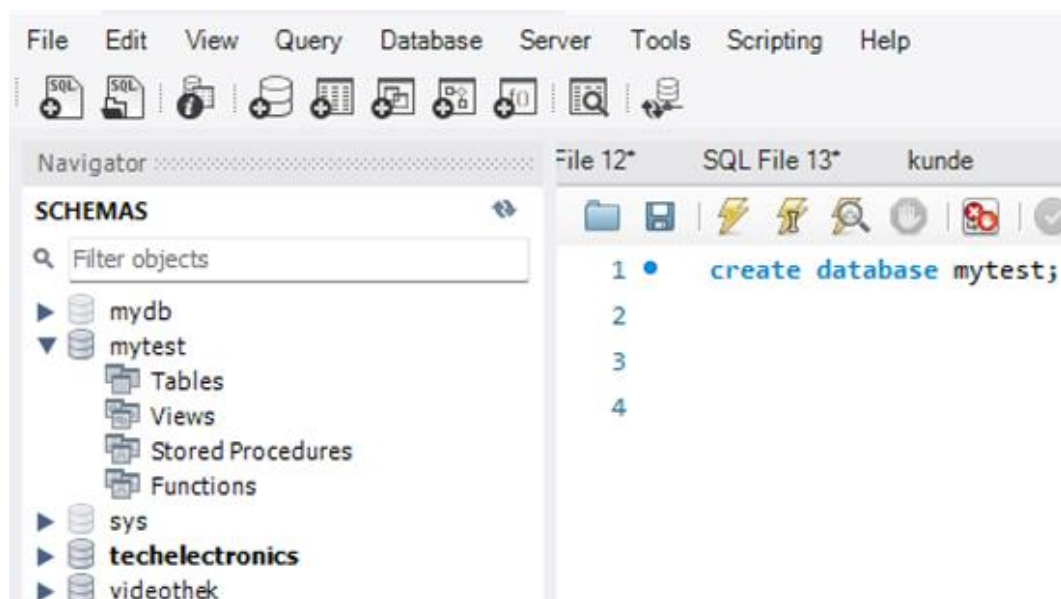


Abbildung 4 Test um Serverfunktionalität zu testen

In einem weiteren Test werden alle Datenbanken angezeigt. Abschliessend wird die Testdatenbank wieder gelöscht:

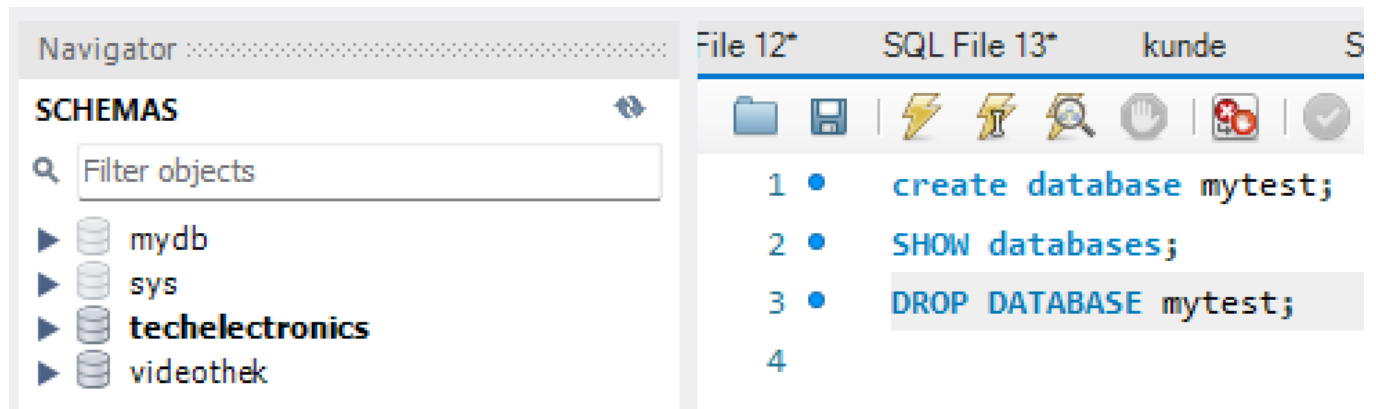


Abbildung 5 Tests der Datenbanken

3.2 Schnittstellenbeschreibung

Komplette Abbildung des Datenmodells der Datenbank Techelectronics:

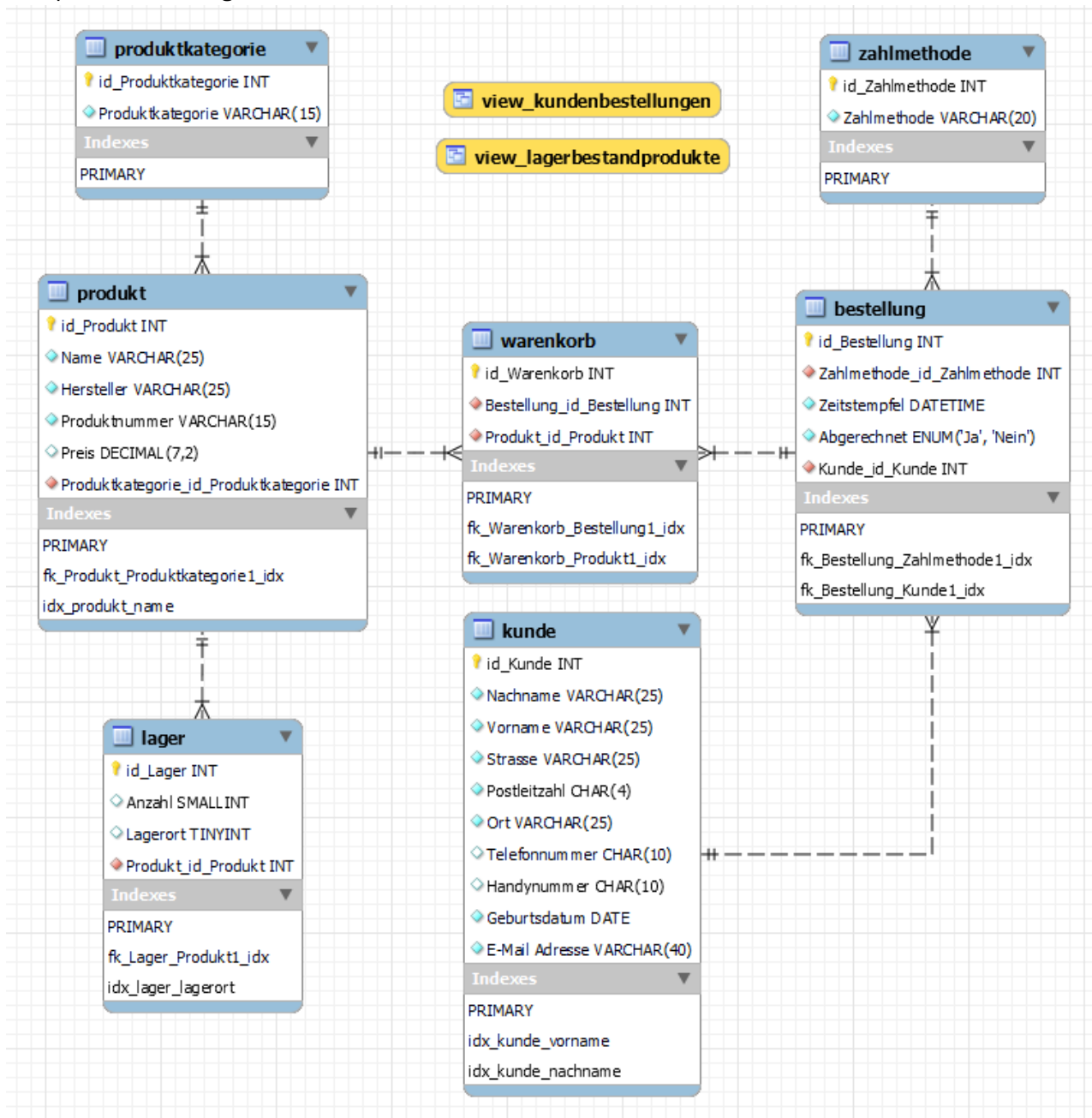


Abbildung 6 Datenmodell von Techelectronics

Das Skript **createDB.sql** erstellt die Datenbank **Techelectronics** samt allen erforderlichen Tabellen, Fremdschlüsselbeziehungen und Indizes. Zu Beginn deaktiviert es vorübergehend die Fremdschlüsselprüfungen, um sicherzustellen, dass die Tabellen in der richtigen Reihenfolge angelegt werden; am Ende des Skripts werden die Prüfungen wieder reaktiviert.

In jeder Tabelle wurden zusätzlich Indizes angelegt, da sie die Abfragezeiten erheblich reduzieren. Durch die Verwendung von Indizes kann die Datenbank ähnlich wie in einem Buch über das Inhaltsverzeichnis direkt auf die relevanten Informationen zugreifen, anstatt alle Inhalte durchsuchen zu müssen. Allerdings erfordern diese Indizes zusätzlichen Speicherplatz und können die Leistung beim Einfügen von Daten leicht beeinträchtigen.

Vergleich Tabellentypen

InnoDB¹⁰:	Unterstützt Transaktionen und Fremdschlüssel. Garantiert Datenintegrität und ist gut für Anwendungen, die viele Schreibvorgänge haben.
MyISAM:	Schnell bei Lesezugriffen, aber keine Transaktionen oder Fremdschlüssel. Gut für leselastige Anwendungen.
HEAP¹¹:	Speichert Daten im RAM, super schnell, aber Daten gehen bei Neustart verloren. Nützlich für temporäre Daten.
ARCHIVE¹²:	Komprimiert Daten, nur lesbar. Gut für Archivierung und alte Daten.
BLACKHOLE:	Speichert keine Daten, leitet sie nur weiter. Nützlich für spezielle Zwecke.

Tabelle 2 Vergleich diverser Tabellentypen

Für die Datenbank Techelectronics haben wir uns für den Tabellentyp InnoDB entschieden, da dieser Transaktionen, Datenintegrität und Fremdschlüssel bietet. Zudem dient dieser gut, für sichere und stabile Datenbanken.

3.3 Datenbank einrichten

Das Skript **importData.sql** wurde erstellt, um Initialdaten in die Tabellen unserer Datenbank zu importieren. Es kann mehrfach ausgeführt werden, ohne dass die referenzielle Integrität verletzt wird oder es zu doppelten Datensätzen kommt. Dabei werden sowohl DELETE- als auch INSERT- und LOAD DATA-Befehle verwendet.

-- Vorhandene Daten in allen Tabellen löschen

```
DELETE FROM Techelectronics.Warenkorb;
DELETE FROM Techelectronics.Lager;
DELETE FROM Techelectronics.Bestellung;
DELETE FROM Techelectronics.Produkt;
DELETE FROM Techelectronics.Produktkategorie;
DELETE FROM Techelectronics.Zahlungsmethode;
DELETE FROM Techelectronics.Kunde;
```

Abbildung 7 Daten von Tabellen löschen

Zu Beginn des Skripts werden alle vorhandenen Daten in den Tabellen gelöscht, um sicherzustellen, dass das Skript mehrfach ausgeführt werden kann, ohne dass bestehende Daten die neuen beeinträchtigen. Die Kundentabelle wird mit dem Befehl LOAD DATA befüllt, um grössere Datenmengen effizient einzulesen. Dabei wird eine CSV-Datei (Kunden.csv) genutzt, die die Kundendaten enthält.

```
-- -----
-- LOAD DATA für 'Kunde'
-- -----
-- Windows Path: C:\ProgramData\MySQL\MySQL Server 8.4\Uploads\

LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.4\\Uploads\\Kunde.csv'
INTO TABLE Techelectronics.Kunde
CHARACTER SET utf8mb4
FIELDS TERMINATED BY ';'
ENCLOSED BY ''
LINES TERMINATED BY '\n'
ignore 1 rows
('Nachname', 'Vorname', 'Strasse', 'Postleitzahl', 'Ort', 'Telefonnummer', 'Handynummer', 'Geburtsdatum', 'E-Mail Adresse');
```

Abbildung 8 IMPORT Befehl

¹⁰ (IONOS, 2021)

¹¹ (MariaDB, 2024)

¹² (dev.mysql, 2024)

Für Tabellen, die nicht mit LOAD DATA befüllt werden, erfolgt der Import über INSERT-Befehle. Beispielhaft dafür ist die Produktetabelle:

```

-----
-- Insert data für 'Produkt'
-----

INSERT INTO Techelectronics.Produkt (Name, Hersteller, Produktnummer, Preis, Produktkategorie_id_Produktkategorie)
VALUES
  ('iPhone 15', 'Apple', '123456', 899.99, 1),
  ('Galaxy S24', 'Samsung', '654321', 799.99, 1),
  ('MacBook Air', 'Apple', '123654', 1199.99, 2),
  ('Surface Pro', 'Microsoft', '456123', 999.99, 3);

```

Abbildung 9 INSERT Befehl

Hinweis: Durch das Leeren der Tabellen bleiben die Primärschlüssel unverändert. So wird gewährleistet, dass die referenzielle Integrität der Datenbank beibehalten wird, auch wenn das Skript mehrfach ausgeführt wird.

Beispiel: Wenn produkt_id der Primärschlüssel der Produktetabelle ist, darf dieser sich bei wiederholtem Ausführen des Skripts nicht ändern.

Bei der Eingabe von Daten ist zudem sicherzustellen, dass die richtige Zeichenkodierung verwendet wird, damit Sonderzeichen wie Umlaute korrekt angezeigt werden. Im Skript wird UTF-8mb4 als Standardkodierung angenommen.

```

-----
-- LOAD DATA für 'Kunde'
-----

-- Windows Path: C:\ProgramData\MySQL\MySQL Server 8.4\Uploads\

LOAD DATA INFILE 'C:\ProgramData\MySQL\MySQL Server 8.4\Uploads\Kunde.csv'
INTO TABLE Techelectronics.Kunde
CHARACTER SET utf8mb4
FIELDS TERMINATED BY ';'
ENCLOSED BY ''
LINES TERMINATED BY '\n'
ignore 1 rows
('Nachname', 'Vorname', 'Strasse', 'Postleitzahl', 'Ort', 'Telefonnummer', 'Handynummer', 'Geburtsdatum', 'E-Mail Adresse');

```

Abbildung 10 richtige Zeichencodierung setzen

Rechte, welche der Benutzer lager_manager@localhost anzeigen lassen mit SHOW GRANTS;

```

Grants for lager_manager@localhost
GRANT USAGE ON *.* TO 'lager_manager'@'localhost'
GRANT SELECT, UPDATE ON `techelectronics`.`lager` TO 'lager_manager'@'localhost'

```

Abbildung 11 Rechte von User anzeigen

Benutzerkonten einrichten

Um zu gewährleisten, dass das Skript mehrfach ausgeführt werden kann, werden bereits existierende Benutzer zuerst gelöscht:

```
-- Vorhandene Benutzer löschen, um das Skript mehrfach auszuführen
DROP USER IF EXISTS 'lager_manager'@'localhost';
DROP USER IF EXISTS 'admin_user'@'localhost';
DROP USER IF EXISTS 'kunde_manager'@'localhost';
```

Abbildung 12 existierende Benutzer löschen

Anschliessend werden neue Benutzer mit spezifischem Passwort und Berechtigungen erstellt:

```
-- -----
-- Anlegen eines neuen Benutzers für die Verwaltung der Tabelle „Lager“.
-- -----
CREATE USER 'lager_manager'@'localhost' IDENTIFIED BY '1234';
GRANT SELECT, UPDATE ON Techelectronics.lager TO 'lager_manager'@'localhost';

-- -----
-- Erstellen eines neuen Admin-Benutzers mit vollen Rechten für die Datenbank
-- -----
CREATE USER 'admin_user'@'localhost' IDENTIFIED BY '1234';
GRANT ALL PRIVILEGES ON Techelectronics.* TO 'admin_user'@'localhost' WITH GRANT OPTION;

-- -----
-- Anlegen eines neuen Benutzers für die Verwaltung der Tabelle 'kunde'.
-- -----
CREATE USER 'kunde_manager'@'localhost' IDENTIFIED BY '1234';
GRANT SELECT ON Techelectronics.kunde TO 'kunde_manager'@'localhost';
```

Abbildung 13 User mit Passwort und Berechtigung erstellen

Nachdem die Berechtigungen zugewiesen wurden, ist es wichtig, die Änderungen mit dem folgenden Befehl zu aktivieren.

```
-- Berechtigungen neu laden, um sicherzustellen, dass alle Änderungen wirksam sind.
FLUSH PRIVILEGES;
```

Abbildung 14 Änderungen der Benutzer speichern und aktivieren

Rechtevergabe Views und Stored Procedures

Für unsere Datenbank wurden die zwei Views view_kundenbestellungen und view_lagerbestandprodukte erstellt.

Die view_kundenbestellungen zeigt den vor – und Nachname an, den Zeitstempel, die Abrechnung und Zahlungsmethode der Kunden.

Die view_lagerbestandprodukte zeigt den Namen, den Hersteller, die Anzahl und den Lagerort eines Produkts.

Wir haben die drei Stored Procedures createOrder, getCustomerInfo, updateStock erstellt.

Der createOrdner kann man direkt die Kunden – Produkteid hinterlegen wie auch die BestellungsID, um eine Bestellung zu erstellen.

Mit getCustomerInfo kann man den Kunden durch die ID, Vornamen, Nachname und E-Mail-Adresse herausfinden.

Mit updateStock kann schnell und einfach den Lagerbestand eines Produkts updaten und verändern.

Weitere wurden die Berechtigungen an die bestimmten User verteilt.

```
-- -----  
-- Berechtigungen für Views festlegen  
-- -----  
  
-- Berechtigung für Benutzer Kunde_Manager:  
GRANT SELECT ON View_KundenBestellungen TO 'kunde_manager'@'localhost';  
  
-- Berechtigung für Benutzer lager_manager:  
GRANT SELECT ON View_LagerbestandProdukte TO 'lager_manager'@'localhost';  
  
-- Berechtigung für Benutzer admin_user:  
GRANT SELECT ON View_KundenBestellungen TO 'admin_user'@'localhost';  
GRANT SELECT ON View_LagerbestandProdukte TO 'admin_user'@'localhost';  
  
-- -----  
-- Berechtigungen für Stored Procedures festlegen  
-- -----  
  
-- Berechtigung für Benutzer Kunde_Manager:  
GRANT EXECUTE ON PROCEDURE Techelectronics.getCustomerInfo TO 'kunde_manager'@'localhost';  
  
-- Berechtigung für Benutzer lager_manager:  
GRANT EXECUTE ON PROCEDURE Techelectronics.updateStock TO 'lager_manager'@'localhost';
```

Abbildung 15 Rechte von VIEWS und Stored Procedures anpassen

4 Testdokumente

4.1 Berechtigung testen 4-6:

Test-ID	Benutzer	Testfalltyp	Beschreibung	Eingabe	Erwartete Ausgabe	Tatsächliche Aufgabe	Ergebnis
001	Test1	Positiv	Lesen von Daten in der Tabelle 'kunden'	<code>SELECT * FROM kundentabelle;</code>	Daten werden angezeigt	Daten werden angezeigt	Bestanden
002	Test2	Positiv	Einfügen von Daten in die Tabelle 'kunden'	<code>INSERT INTO kundentabelle (name) VALUES ('Test2');</code>	Datensatz wird eingefügt	Datensatz wird eingefügt	Bestanden
003	Test3	Positiv	Löschen von Daten in der Tabelle 'kunden'	<code>DELETE FROM kundentabelle WHERE id = 3;</code>	Datensatz wird gelöscht	Datensatz wird gelöscht	Bestanden

Tabelle 3 Testen von Rechten

4.2 Views Testen:

Test-ID	Benutzer	Testfalltyp	Testbeschreibung	Eingabe	Erwartete Ausgabe	Tatsächliche Ausgabe	Ergebnis
001	Test1	Positiv	Überprüfung der View View1 auf Korrektheit der Daten	<code>SELECT * FROM View1;</code>	Korrekte gefilterte Daten basierend auf WHERE	Gefilterte Daten werden korrekt angezeigt	Bestanden
002	Test2	Positiv	Überprüfung der View View2 mit INNER JOIN	<code>SELECT * FROM View2;</code>	Daten aus mehreren Tabellen korrekt verknüpft	Daten korrekt verknüpft	Bestanden
003	Test3	Negativ	Überprüfung, ob Benutzer Test3 keinen Zugriff auf View1 hat	<code>SELECT * FROM View1;</code>	Fehler: Keine Berechtigung zum Zugriff	Zugriff verweigert	Bestanden
004	Test4	Negativ	Überprüfung, ob Benutzer Test1 keinen Zugriff auf View2 hat	<code>SELECT * FROM View2;</code>	Fehler: Keine Berechtigung zum Zugriff	Zugriff verweigert	Bestanden

Tabelle 4 Kontrolle von den VIEWS

4.3 Stored Procedure Testen:

Test-ID	Benutzer	Testfalltyp	Testbeschreibung	Eingabe	Erwartete Ausgabe	Tatsächliche Ausgabe	Ergebnis
001	Test1	Positiv	Überprüfung der SP getCustomerOrders()	CALL getCustomerOrders();	Alle Bestellungen für Kunden werden korrekt angezeigt	Korrekte Ausgabe der Bestellungen	Bestanden
002	Test2	Negativ	Überprüfung, ob Benutzer Test2 keinen Zugriff auf getCustomerOrders() hat	CALL getCustomerOrders();	Fehler: Keine Berechtigung zum Zugriff	Zugriff verweigert	Bestanden
003	Test3	Positiv	Überprüfung der SP getProductSales()	CALL getProductSales();	Alle Produktverkäufe werden korrekt angezeigt	Korrekte Ausgabe der Produktverkäufe	Bestanden

Tabelle 5 Kontrolle von Stored Procedures

4.4 Backup testen:

Test-ID	Benutzer	Beschreibung	Eingabe	Erwartete Ausgabe	Ergebnis
001	Test1	Backup mit mysqldump	mysqldump --opt --routines --add-drop-database --databases my_database > backup.sql	Backup erfolgreich	Bestanden
002	Test2	Restore der Datenbank	DROP DATABASE my_database; mysql < backup.sql	Restore erfolgreich	Bestanden
003	Test3	Verifizierung der Daten	SELECT * FROM my_database.table;	Alle Daten korrekt wiederhergestellt	Bestanden

Tabelle 6 Testen von Backup

4.5 Testfall Logdateien:

Test-ID	Log-Datei	Testbeschreibung	Eingabe	Erwartete Ausgabe	Ergebnis
001	Error log	Syntax-Fehler	My.ini ausgetauscht innodb_log_buffer_sizee = 1M	MySQL Dienst lässt sich nicht mehr starten.	MySQL Dienst lässt sich nicht mehr starten.
002	General query log	Syntax-Fehler	SELECT * FROM Kunde;	Tabelle Kunde existiert nicht	Tabelle Kunde existiert nicht
003	Slow log	Lange Abfrage	SELECT * FROM Kunde;	Abfrage ging 2 Sekunden lang	Abfrage ging 2 Sekunden lang
004	Binary log	Syntax-Fehler	Insert Into Kunde (Vorname, Name) Values ('Hans' , 'Koch');	PLZ darf nicht NULL sein	PLZ darf nicht NULL sein

Tabelle 7 Testfälle von Logdateien

4.6 Performanceverbesserung

Um die Performance einer Datenbank zu verbessern, gibt es folgende Möglichkeiten:

- Verwenden von Indizes
- Optimieren von Abfragen
- Auswählen geeigneter Tabellentypen
- Einfügen von Redundanzen
- Optimieren von Attributdefinitionen

4.7 SQL-Abfrageoptimieren:

Optimierung durch Vermeidung von Wildcards

Output vor der Abfrageoptimierung der Kundentabelle:

Action	Message	Duration / Fetch
SELECT sql_no_cache * FROM kunde LIMIT 10000	8000 row(s) returned	0.000 sec / 0.031 sec

Abbildung 16 Abfrage Output Abfrageoptimierung

SELECT sql_no_cache vorname, nachname FROM kunde LIMIT 10000	8000 row(s) returned	0.000 sec / 0.016 sec
--	----------------------	-----------------------

Abbildung 17 Abfrage Output Abfrageoptimierung

Optimierung durch effizientere Datentypen

Vor der Optimierung (char(20):

Action	Message	Duration / Fetch
SELECT sql_no_cache ORT FROM kunde LIMIT 1000000000	144000 row(s) returned	0.000 sec / 0.187 sec

Abbildung 18 Optimierung von Datentypen vorher

Nach der Optimierung zu char(4):

SELECT sql_no_cache ORT FROM kunde LIMIT 1000000000	144000 row(s) returned	0.015 sec / 0.110 sec
---	------------------------	-----------------------

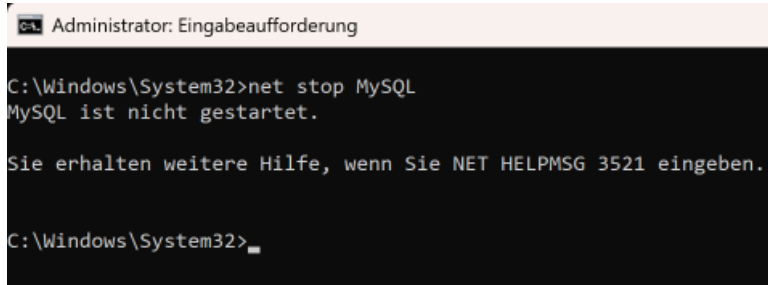
Abbildung 19 Optimierung von Datentyp nachher

5 Betriebsanweisungen

5.1 Datenbank Starten und Stoppen

Hinweis: CMD Muss mit Administratoren Rechte gestartet werden.

Um den MySQL Dienst auf Windows zu stoppen, führt man folgenden Befehl aus: **net stop MySQL**.



```
Administrator: Eingabeaufforderung

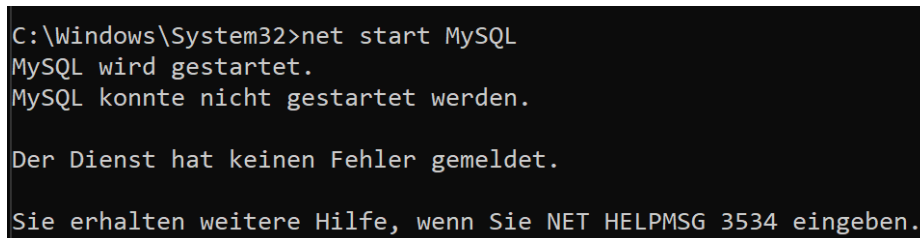
C:\Windows\System32>net stop MySQL
MySQL ist nicht gestartet.

Sie erhalten weitere Hilfe, wenn Sie NET HELPMSG 3521 eingeben.

C:\Windows\System32>
```

Abbildung 20 MySQL Dienst via CMD stoppen

Um den MySQL Dienst auf Windows wieder zu starten, führt man folgenden Befehl aus: **net Start MySQL**



```
C:\Windows\System32>net start MySQL
MySQL wird gestartet.
MySQL konnte nicht gestartet werden.

Der Dienst hat keinen Fehler gemeldet.

Sie erhalten weitere Hilfe, wenn Sie NET HELPMSG 3534 eingeben.
```

Abbildung 21 MySQL via CMD starten

MySQL Workbench

Hinweis: Man muss mit einem User angemeldet sein mit Admin Rechte

Um den Den MySQL Server über das Managementtool zu stoppen, sind folgende Schritte nötig: **Server -> Startup/Shutdown -> Stop Server**

Wird der Server über das Management-Tool gestoppt, fährt die Instanz herunter. Um den Server wieder zu starten, ist es notwendig, das Management-Tool neu zu öffnen oder den Server manuell erneut zu starten.

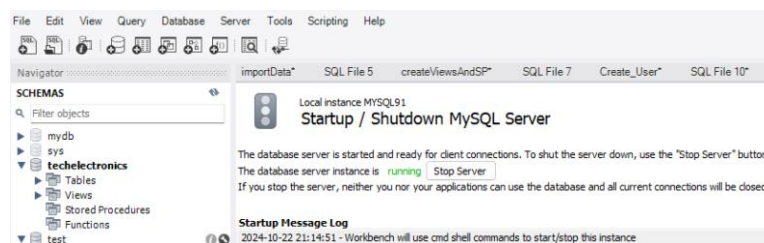


Abbildung 22 Managementtool Fehlerbehebung

Konsole

Hinweis: Man muss in den Pfad "C:\Program Files\MySQL\MySQL Server 9.1\bin" um die Befehle auszuführen können mit «cd» vor den Befehl.

Um den Server in der Konsole zu stoppen, verwendet man den folgenden Befehl:

MySQL Server 8.4\bin> mysqladmin -u root -p shutdown

```

C:\Windows\System32>cd "C:\Program Files\MySQL\MySQL Server 9.1\bin"
C:\Program Files\MySQL\MySQL Server 9.1\bin>mysqladmin -u root -p shutdown
Enter password: ****
  
```

Abbildung 23 Server Shutdown Befehl via CMD

Um den Server in der Konsole zu starten, verwendet man den folgenden Befehl:

mysqld –default-file="C:\ProgramData\MySQL\MySQL Server 9.1\my.ini".

```

C:\Program Files\MySQL\MySQL Server 9.1\bin>mysqld -default-file="C:\ProgramData\MySQL\MySQL Server 9.1\my.ini"
2024-10-23T16:57:48.4021722 [System] [MY-015015] [Server] MySQL Server - start.
2024-10-23T16:57:48.7111222 [System] [MY-010116] [Server] C:\Program Files\MySQL\MySQL Server 9.1\bin\mysqld.exe (mysqld 9.1.0) starting as process 10812
2024-10-23T16:57:48.7519212 [Warning] [MY-010091] [Server] Can't create test file C:\Program Files\MySQL\MySQL Server 9.1\data\mysqld_tmp_file_case_insensitive_test.lower-test
2024-10-23T16:57:48.7524542 [Warning] [MY-010091] [Server] Can't create test file C:\Program Files\MySQL\MySQL Server 9.1\data\mysqld_tmp_file_case_insensitive_test.lower-test
2024-10-23T16:57:48.7530702 [ERROR] [MY-013276] [Server] Failed to set datadir to 'C:\Program Files\MySQL\MySQL Server 9.1\data\' (OS errno: 2 - No such file or directory)
2024-10-23T16:57:48.7552062 [ERROR] [MY-010119] [Server] Aborting
2024-10-23T16:57:48.7563372 [System] [MY-010910] [Server] C:\Program Files\MySQL\MySQL Server 9.1\bin\mysqld.exe: Shutdown complete (mysqld 9.1.0) MySQL Community Server - GPL.
2024-10-23T16:57:48.7563462 [System] [MY-015016] [Server] MySQL Server - end.
  
```

Abbildung 24 Server Starten via CMD

5.2 Backup mit MySQL Dump:

Zur Datensicherung von MySQL Datenbanken gibt es vier Möglichkeiten, wobei jede Methode vor und Nachteile hat:

mysqldump¹³:	Exportiert Datenbanken als SQL-Datei. Gut für kleinere Datenbanken, langsam bei grossen.
Filecopy:	Kopiert das MySQL-Datenverzeichnis. Schnell, aber nur bei gestoppter Datenbank.
mysqlhotcopy:	Schnell für MyISAM-Tabellen. Funktioniert nur für MyISAM.
mysqlbackup:	Online-Backups für grosse Datenbanken, Teil der kostenpflichtigen MySQL Enterprise Edition.

Tabelle 8 Diverse Arten ein Backup zu erstellen

Tabelle 9 Möglichkeiten zur Datensicherung von Datenbanken

Backup der gesamten Datenbank Techelectronics mit MySQLdump:

```

C:\Program Files\MySQL\MySQL Server 9.1\bin>mysqldump -u test1 -p --opt --routines --add-drop-database --databases techelectronics > C:\HALLO\Test5.sql
Enter password: ****
  
```

Abbildung 25 Backup erstellen

Restore der gesamten Datenbank Techelectronics mit MySQLdump:

¹³ (Donauer, 2011)

```
Administrator: Eingabeaufforderung - cmd clear
C:\Program Files\MySQL\MySQL Server 9.1\bin>mysqldump -u root -p techelectronics < C:\Users\fabia\OneDrive\Dokumente\dum
ps\techelectronics.sql
Enter password: ****
-- MySQL dump 10.13 Distrib 9.1.0, for Win64 (x86_64)
--
-- Host: localhost Database: techelectronics
-----
-- Server version 9.1.0
```

Abbildung 26 Backup restore

5.3 Übersicht über Log-dateien:

Error Log:

- **Eigenschaften:** protokolliert alle kritischen Fehlermeldungen und Server-Abstürze.
- **Einsatzbereich:** Fehlerdiagnose und Überwachung von Server-Problemen.

General Query Log:

- **Eigenschaften:** erfasst jede SQL-Abfrage, die an den Server geschickt wird.
- **Einsatzbereich:** hilft bei der Analyse des Abfrageflusses, Debugging und Fehlersuche.

Slow Query Log:

- **Eigenschaften:** loggt Abfragen, die eine definierte Ausführungszeit überschreiten.
- **Einsatzbereich:** Performance-Optimierung und Identifikation langsamer Abfragen.

Binary Log:

- **Eigenschaften:** speichert alle Änderungen an Datenbanken (Insert, Update, Delete).
- **Einsatzbereich:** Datenwiederherstellung, Replikation und Backup-Strategien.

Test-ID	Log-Datei	Testbeschreibung	Eingabe	Ergebnis
001	Error log	Syntax-Fehler	My.ini das ausgetausch innodb_log_buffer_sizee = 1M	MySQL Dienst lässt sich nicht mehr starten.
002	General query log	Syntax-Fehler	SELECT * FROM Kunde;	Tabelle Kunde existiert nicht
003	Slow log	Lange Abfrage	SELECT * FROM Kunde;	Abfrage ging 2 Sekunden lang
004	Binary log	Syntax-Fehler	Insert Into Kunde (Vorname, Name) Values ('Hans', 'Koch');	PLZ darf nicht NULL sein

Tabelle 10 Testfälle von Logdateien

6 Reflexion

6.1 Fabian Manser:

Das Projekt war im Ganzen meiner Meinung nach ein Erfolg. Es verlief grundsätzlich im Allgemeinen alles reibungslos. Natürlich gab es auch einige Aufgaben, bei denen es Schwierigkeiten gab. Diese Probleme liessen sich aber alle in der Gruppe lösen. Es gab zum Beispiel Probleme bei den VIEWS. Diese wurden zwar erstellt aber hatten keinen Inhalt. Dieses Problem hat uns einen gewissen Moment begleitet, konnte aber danach zum Glück behoben werden. Meiner Meinung nach wäre eine übersichtlichere Aufgabenstellung hilfreich gewesen.

6.2 Bojan Maljuric:

Das MySQL-Projekt war insgesamt eine gute Erfahrung, aber auch recht anstrengend. Es gab einige Herausforderungen, besonders beim Setzen der Benutzerrechte und beim Testen der Funktionen, da nicht immer alles sofort wie erwartet funktionierte. Zudem war die Performance bei grösseren Backups manchmal ein Problem, was das Arbeiten verzögerte.

Trotz dieser Schwierigkeiten fand ich das Projekt sehr lehrreich. Es hat mir ein besseres Verständnis für Datenbanken und ihre Verwaltung gegeben, und ich habe viel über den Umgang mit Benutzerrechten und Backups gelernt. Insgesamt war es eine wertvolle Erfahrung.

6.3 Jonas Sieber:

Das Projekt war insgesamt lehrreich, aber auch herausfordernd. Die Installation und Grundkonfiguration von MySQL Workbench liefen gut, aber bei den Views und Stored Procedures gab es anfangs Probleme. Diese konnten wir nach Rücksprache im Team lösen. Auch das Optimieren der Performance und die Benutzerrechte waren spannend und haben mir geholfen, das System besser zu verstehen. Die abschliessende Dokumentation war zwar sehr aufwendig, brachte aber Klarheit. Alles in allem eine gute Erfahrung, bei der ich viel über Datenbanken mitgenommen habe. Schlussendlich wäre eine übersichtlichere Aufgabenstellung aber hilfreich gewesen.

7 Anhänge

7.1 **Abbildungsverzeichnis**

Abbildung 1 Titelblatt (Medienakademie, kein Datum)	0
Abbildung 2 Port, Version und Server Status anzeigen.....	5
Abbildung 3 Alle Benutzer anzeigen lassen	6
Abbildung 4 Test um Serverfunktionalität zu testen	6
Abbildung 5 Tests der Datenbanken	7
Abbildung 6 Datenmodell von Techelectronics	8
Abbildung 7 Daten von Tabellen löschen	9
Abbildung 8 IMPORT Befehl	9
Abbildung 9 INSERT Befehl	10
Abbildung 10 richtige Zeichencodierung setzen.....	10
Abbildung 11 Rechte von User anzeigen.....	10
Abbildung 12 existierende Benutzer löschen	11
Abbildung 13 User mit Passwort und Berechtigung erstellen	11
Abbildung 14 Änderungen der Benutzer speichern und aktivieren	11
Abbildung 15 Rechte von VIEWS und Stored Procedures anpassen	12
Abbildung 16 Abfrage Output Abfrageoptimierung	15
Abbildung 17 Abfrage Output Abfrageoptimierung	15
Abbildung 18 Optimierung von Datentypen vorher.....	15
Abbildung 19 Optimierung von Datentyp nachher.....	15
Abbildung 20 MySQL Dienst via CMD stoppen.....	16
Abbildung 21 MySQL via CMD starten	16
Abbildung 22 Managementtool Fehlerbehebung	16
Abbildung 23 Server Shutdown Befehl via CMD	17
Abbildung 24 Server Starten via CMD.....	17
Abbildung 25 Backup erstellen	17
Abbildung 26 Backup restore.....	18

7.2 **Tabellenverzeichnis**

Tabelle 1 Übersicht DBMS	4
Tabelle 2 Vergleich diverser Tabellentypen	9
Tabelle 3 Testen von Rechten	13
Tabelle 4 Kontrolle von den VIEWS	13
Tabelle 5 Kontrolle von Stored Procedures.....	14
Tabelle 6 Testen von Backup	14
Tabelle 7 Testfälle von Logdateien	15
Tabelle 8 Diverse Arten ein Backup zu erstellen	17
Tabelle 9 Möglichkeiten zur Datensicherung von Datenbanken	17
Tabelle Testfälle von Logdateien	18
Tabelle 9 Arbeitsjournal Jonas Sieber	22
Tabelle 10 Arbeitsjournal Fabian Manser	22
Tabelle 11 Arbeitsjournal Bojan Maljuric.....	23

7.3 Literaturverzeichnis

- AWS. (24. 10 2024). *Amazon Webservice*. Von <https://aws.amazon.com/de/compare/the-difference-between-sql-and-mysql/> abgerufen
- Boekhoven, P. (08. 02 2022). *Was ist eine Transaktion und ACID?* Von <https://www.youtube.com/watch?v=fgXvbnkPRzU> abgerufen
- Dadhich, P. (24. 10 2024). *znetlive*. Von <https://www.znetlive.com/blog/mysql-vs-mssql-key-differences/> abgerufen
- dev.mysql. (24. 10 2024). *dev.mysql*. Von <https://dev.mysql.com/doc/refman/8.4/en/archive-storage-engine.html> abgerufen
- divers. (24. 10 2024). *wikipedia*. Von [https://de.wikipedia.org/wiki/Replikation_\(Datenverarbeitung\)](https://de.wikipedia.org/wiki/Replikation_(Datenverarbeitung)) abgerufen
- divers. (24. 10 2024). *Wikipedia*. Von <https://de.wikipedia.org/wiki/PostgreSQL> abgerufen
- Donauer, J. (20. 01 2011). *computerwoche*. Von <https://www.computerwoche.de/article/2861770/mysql-backup-via-terminal.html> abgerufen
- IONOS. (11. 01 2021). *IONOS Digital Guide*. Von <https://www.ionos.at/digitalguide/hosting/hosting-technik/was-ist-innodb/> abgerufen
- MariaDB. (24. 10 2024). *MariaDB*. Von <https://mariadb.com/kb/en/memory-storage-engine/> abgerufen
- Medienakademie. (kein Datum). *ARD ZDF medienakademie*. Abgerufen am 22. Oktober 2024 von ARD ZDF medienakademie: <https://www.ard-zdf-medienakademie.de/seminar/31370>
- Neovera. (24. 10 2024). *Neovera choosing-rdbms*. Von <https://neovera.com/choosing-rdbms-oracle-mysql-sql-server/> abgerufen
- persis. (24. 10 2024). *persis.de/glossar*. Von <https://www.persis.de/glossar/volltextsuche/> abgerufen
- RemoteScout24. (25. 04 2023). *Alles über MySQL*. Von <https://remotescout24.com/de/blog/601-alles-ueber-mysql> abgerufen
- Wiggers, S.-J. (14. 03 2023). *InfoQ*. Von <https://www.infoq.com/news/2023/03/new-azure-mysql-integrations/> abgerufen

7.4 Arbeitsjournal Jonas Sieber

Datum	Arbeit	Aufwand [h]
15.08.2024	Aufnahme der Ausgangslage	0.5
15.08.2024	Konzeptdefinition	0.5
15.08.2024	Evaluation DBMS	0.5
22.08.2024	Installation DBMS	0.5
22.08.2024	Konfiguration MYSQL Workbench	0.5
22.08.2024	Konzipierung und Erstellung ERD	1.5
22.08.2024	Implementierung Datenmodell	2.0
29.08.2024	Import von Initialdaten	1.5
05.09.2024	Benutzerkonten einrichten	1.5
05.09.2024	Views erstellen	2.0
12.09.2024	Stored Procedures erstellen	2.5
19.09.2024	Konzipierung Backup und Restore	1.0
19.09.2024	Leistung Optimieren	1.0
26.09.2024	Einrichtung Logging	0.5
21.10.2024	Korrektur und Kommentierung Scripts	2.0
22.10.2024	Nachführung Dokumentation	2.0
24.10.2024	Korrektur Formatierung Dokumentation	2.0
Total:		22.00

Tabelle 9 11 Arbeitsjournal Jonas Sieber

7.5 Arbeitsjournal Fabian Manser

Datum	Arbeit	Aufwand [h]
22.08.2024	Dokumentation Datenimplementierung	1.5
29.08.2024	Testen / Dokumentieren von Benutzerkonten und Rechten	2.0
05.09.2024	Dokumentieren von Views und Stored Procedures	2.5
12.09.2024	Testen von Stored Procedures	2.0
19.09.2024	Backup und Restore dokumentieren	1.5
26.09.2024	Formatierung der Dokumentation	1.5
21.10.2024	Leistungstests dokumentieren	2.0
22.10.2024	Finale Testberichte schreiben	3.5
24.10.2024	Dokumentation abschliessen	4.0
Total:		20.50

Tabelle 10 12 Arbeitsjournal Fabian Manser

7.6 Arbeitsjournal Bojan Maljuric

Datum	Arbeit	Aufwand [h]
15.08.2024	Technische Implementierung vorbereiten	0.5
22.08.2024	Unterstützung Datenbankkonfiguration	2.0
22.08.2024	Datenmodell überprüfen und optimieren	2.5
29.08.2024	Testing der Datenimplementierung	2.0
05.09.2024	Datenmodelle anpassen	2.5
05.09.2024	Benutzerrechte konfigurieren	2.0
12.09.2024	Stored Procedures überprüfen	2.0
19.09.2024	Leistungsoptimierung der Datenbank	2.5
26.09.2024	Testing von Backup und Restore	1.5
21.10.2024	Skripte optimieren und anpassen	2.0
22.10.2024	Technische Dokumentation nachführen	1.5
24.10.2024	Endgültige Überprüfung der Systemleistung	2.0
Total:		23.00

Tabelle 11.13 Arbeitsjournal Bojan Maljuric