	Lehrveranstaltung	<b>Grundlagen von Datenbanken</b>		WS 2018/19
	Aufgabenzettel	<b>1 (Lösungsvorschläge)</b>		
	Gesamtpunktzahl	<b>40</b>		
	Ausgabe	<b>Mi. 17.10.2018</b>	Abgabe	<b>Fr. 02.11.2018</b>



## 1 Informationssysteme

[10 P.]

### a) Charakterisierung:

Erläutern Sie den Begriff Informationssystem und nennen Sie in diesem Zusammenhang drei relevante Aufgaben eines rechnergestützten Informationssystems. (2 Punkte)

#### Lösungsvorschlag:

Aus dem Skript:

- i) Ein Informationssystem (IS) besteht aus Menschen und Maschinen, die **Informationen erzeugen und/oder benutzen** und die durch Kommunikationsbeziehungen miteinander verbunden sind.
- ii) Ein rechnergestütztes IS ist ein System, bei dem die **Erfassung, Speicherung und/oder Transformation von Informationen** durch den Einsatz von EDV teilweise automatisiert ist. In der betrieblichen Praxis besteht es typischerweise aus einer Menge unabhängiger Systeme, die zusammen die angestrebte Leistung erbringen.

### b) Datenunabhängigkeit:

Definieren Sie kurz den Begriff Datenunabhängigkeit und unterscheiden Sie dabei die logische von der physischen Datenunabhängigkeit. (2 Punkte)

#### Lösungsvorschlag:

- i) **logische Datenunabhängigkeit:** Die logische Datenunabhängigkeit ist die Stabilität des Datenbankschemas gegen Änderungen der Anwendung und umgekehrt. Anwendungen müssen nicht umfassend geändert werden, wenn das Datenbankschema geändert wird. Sie umfasst die Unabhängigkeit der externen Datendarstellung vom verwendeten Datenbankschema. Eine Sicht kann die logische Anordnung der betrachteten Daten den Benutzerwünschen anpassen (z.B. Vertauschung von Attributen).

	Lehrveranstaltung	<b>Grundlagen von Datenbanken</b>		WS 2018/19
	Aufgabenzettel	<b>1 (Lösungsvorschläge)</b>		
	Gesamtpunktzahl	<b>40</b>		
	Ausgabe	<b>Mi. 17.10.2018</b>	Abgabe	<b>Fr. 02.11.2018</b>

ii) **physische Datenunabhängigkeit:** Die physische Datenunabhängigkeit trennt die logische Darstellung der Daten durch das Datenbankschema von der physikalischen Speicherung der Daten. Eine Änderung der physischen Anordnung der Daten (z.B. Reorganisation auf der Festplatte) hat keinen Einfluss auf die logische Darstellung der Daten (das Datenbankschema). Die physische Speicherung ist nach außen transparent und bleibt dem Nutzer somit verborgen.

c) **Beispiele:**

Nennen Sie drei Anwendungsbeispiele für Informationssysteme und beschreiben Sie die jeweils charakteristischen Vorgänge. Vermeiden Sie die Wiederholung von Beispielen aus der Vorlesung. (6 Punkte)

**Lösungsvorschlag:**

**Beispiel 1 - Informationssystem eines Krankenhauses:** Im Krankenhaus werden Verletzungen und Krankheiten von Patienten durch medizinisches Personal behandelt und soweit möglich geheilt. Die persönlichen Daten der Patienten, ihre Krankengeschichte, Untersuchungsergebnisse und Diagnosen sowie Gestalt und Verlauf ihrer Behandlung werden durch das betriebliche Informationssystem gespeichert und können von bestimmten Benutzergruppen innerhalb des Personals erstellt, angezeigt und verändert werden. Das Personal untergliedert sich in verschiedene Fachbereiche und Berufsgruppen, die mit eigenen Benutzerrechten verknüpft sind. Der Arzt ruft Untersuchungsergebnisse von Patienten ab, fordert weitere Untersuchungen an, beurteilt das Krankheitsbild von Patienten und legt den Behandlungsplan fest und verwaltet die Leistungen für die spätere Abrechnung mit den Krankenkassen. Die Pflegekräfte und Therapeuten führen den Behandlungsplan des Arztes aus, bereiten Untersuchungen vor, indem beispielsweise ein Terminplan erstellt wird, geben Untersuchungsergebnisse ein und dokumentieren pflegerische Tätigkeiten. Die Kosten der Behandlung werden über festgelegte Tarife mit den jeweils zuständigen Krankenkassen abgerechnet.

Typische Vorgänge:

- Aufnahme/Verlegung/Entlassung eines Patienten.
- Bearbeitung allgemeiner Daten des Patienten.
- Eingabe der Gesundheitsdaten des Patienten.
- Anordnung / Vorbereitung einer Untersuchung/Therapie .
- Zuordnung eines Behandelnden Arztes.
- Abrechnung der Behandlung von Patienten.
- Verwaltung von Labordaten.
- Bestellung von Material.
- Personalverwaltung.

	Lehrveranstaltung	<b>Grundlagen von Datenbanken</b>		WS 2018/19
	Aufgabenzettel	<b>1 (Lösungsvorschläge)</b>		
	Gesamtpunktzahl	<b>40</b>		
	Ausgabe	<b>Mi. 17.10.2018</b>	Abgabe	<b>Fr. 02.11.2018</b>

**Beispiel 2 - Informationssystem eines Kinos:** Für die Organisation eines Kinos benötigt man ein Informationssystem, welches die aktuellen Kinofilme mit Namen, kurzer Beschreibung und Filmdauer enthält. Außerdem muss verwaltet werden, welche Werbungen in welchen Kinosälen ausgestrahlt werden sollen und welche Dauer diese haben. Hinzu kommt die Verwaltung von Mitarbeitern mit ihren persönlichen Daten wie Gehalt, Tätigkeit und Beschäftigungszeiten. Über die ausgestrahlten Filme und deren Zuschauer werden Statistiken geführt.

Typische Vorgänge:

- Film mit entsprechenden Eigenschaften (Regisseur, Sprache, Länge etc.) hinzufügen.
- Verwalten verschiedener Kinosäle mit unterschiedlicher Platzzahl.
- Neue Vorstellung anlegen mit entsprechenden Eigenschaften (Kinosaal, Film, Zeit).
- Zuordnen von Vorstellungen zu Filmen.
- Zuordnen von Vorstellungen zu Uhrzeiten.
- Verschieben einer Vorstellung.
- Programmplanung: Saalverwaltung, Reservierung, Film-Anmietung.
- Verkaufen von Karten für bestimmte Vorstellungen.
- Werbung verwalten (Werbung einem Film zuordnen).
- Verwaltung von Mitarbeitern.

**Beispiel 3 - International Shark Attack File (ISAF):** Die ISAF ist eine internationale Datenbank, in der Angriffe von Haien auf Menschen gespeichert werden. Sie enthält wissenschaftliche Daten wie Autopsieberichte, Fotos der Opfer sowie Zeitpunkt und Verlaufsbeschreibung des Angriffs. Nur autorisierte Wissenschaftler haben Zugriff auf die Datenbank.


Typische Vorgänge:

- Eintragen/Abrufen eines Berichts über Haiangriffe.
- Eintragen/Überprüfen/Anmelden eines Wissenschaftlers.
- Erstellen von Statistiken
- Ausgeben von Warnmeldungen bei besonderen Häufungen von Angriffen in einem bestimmten Gebiet.

## 2 Miniwelt

[12 P.]

Für ein Tippspiel der kommenden Fußball-Europameisterschaft soll eine browserbasierte Anwendung implementiert werden. Das Tippspiel läuft dabei wie folgt ab: Jeder Mitspieler soll sich an der Anwendung anmelden und

	Lehrveranstaltung	<b>Grundlagen von Datenbanken</b>		WS 2018/19
	Aufgabenzettel	<b>1 (Lösungsvorschläge)</b>		
	Gesamtpunktzahl	<b>40</b>		
	Ausgabe	<b>Mi. 17.10.2018</b>	Abgabe	<b>Fr. 02.11.2018</b>

eine Tippspieltippgemeinschaft erstellen können. Der jeweilige Gründer verwaltet dabei die Tippspieltippgemeinschaft indem er für diese Wettbewerbe anlegt, Begegnungen zu einem Wettbewerb hinzufügt oder Ergebnisse einträgt. Der Tippspieltippgründer hat außerdem die Möglichkeit, weitere Mitspieler zu dieser Gemeinschaft hinzuzufügen oder auch zu entfernen. Diese hinzugefügten Mitspieler können dann Tipps auf die angelegten Begegnungen abgeben. Nach dem Beenden einer Begegnung trägt der Verwalter das Ergebnis ein und die Mitspieler erhalten für ihre Tipps entsprechende Punkte. Die Mitspieler können sich in der Anwendung jeweils über ihren aktuellen Punktestand innerhalb einer Tippspieltippgemeinschaft informieren oder auch die Ergebnisse zu den Begegnungen einsehen.

- a) Leiten Sie aus der beschriebenen Miniwelt die für die Realisierung des Tippspiels relevanten Elemente (Objekttypen) und Vorgänge ab. (4 Punkte)

#### **Lösungsvorschlag:**

- **Mögliche Objekttypen (mit Attributen):**

- Mitspieler: Vorname, Name, Passwort (, Punktestand)
- Verwalter (Gründer): Vorname, Name, Passwort (, Punktestand)
- Tippspieltippgemeinschaft: Name
- Wettbewerb: Name
- Tipp: Ergebnis A, Ergebnis B, Punkte
- Begegnung: Datum, Uhrzeit, Team A, Team B
- Ergebnis: Tore A, Tore B


- **Mögliche Vorgänge:**

i) Verwalter / Gründer:

- Anmelden an der Anwendung
- Tippspieltippgemeinschaft erstellen
- Mitglieder verwalten: hinzufügen, entfernen
- Wettbewerbe verwalten: hinzufügen
- Begegnungen verwalten: hinzufügen, entfernen, Ergebnisse eintragen

ii) Mitspieler:

- Anmelden an der Anwendung
- Ergebnisse von Begegnungen ansehen
- Tipps für Begegnungen abgeben
- Punktestand einsehen

	Lehrveranstaltung	<b>Grundlagen von Datenbanken</b>		WS 2018/19
	Aufgabenzettel	<b>1 (Lösungsvorschläge)</b>		
	Gesamtpunktzahl	<b>40</b>		
	Ausgabe	<b>Mi. 17.10.2018</b>	Abgabe	<b>Fr. 02.11.2018</b>

b) Welche Anforderungen an die Anwendung ergeben sich aus der beschriebenen Miniwelt? Diskutieren Sie die in der Vorlesung genannten allgemeinen Anforderungen an Datenbanksysteme anhand des hier beschriebenen Beispiels. (8 Punkte)

### Lösungsvorschlag:

Entsprechend der Anforderungen an Datenbanksysteme aus dem Skript:

#### i) Kontrolle über die operationalen Daten:

- Ergebnisse von Begegnungen sollen von mehreren Benutzern gleichzeitig getippt sowie deren Ergebnisse eingesehen werden können.
- Verhindern der Abgabe von mehreren Tipps für eine Begegnung von einem einzelnen Mitspieler.
- Verschiedene Benutzerrechte für unterschiedliche Nutzergruppen (Verwalter, Spieler)

#### ii) Leichte Handhabbarkeit der Daten:

- Bearbeiten und Tippen ohne Wissen über die Technik der Datenbank
- Benutzerfreundliche Oberfläche für die Tippspieler und Verwalter
- Einfache Programmierschnittstelle für Entwickler (unabhängig von der physischen Speicherung der Daten)

#### iii) Kontrolle der Datenintegrität:


- Erhaltung der Konsistenz der Daten bei Systemabstürzen (z.B. Wiederherstellung der ursprünglichen Version eines Tippspiels oder Benutzers bei Systemabsturz während seiner Bearbeitung).
- Beim Zugriff auf die Daten sollte es für den Benutzer den Anschein haben, als sei er die einzige Person, die auf die Daten zugreift (logischer Einbenutzerbetrieb).
- Der gleichzeitig Zugriff mehrerer Benutzer und Verwalter soll unterstützt werden, ohne das Konflikte in der Benutzung die Daten inkonsistent werden lassen (Mehrbenutzerbetrieb)

#### iv) Leistung, Skalierbarkeit und Erweiterbarkeit:

- Das System soll den Zugriff von gleichzeitigen Benutzern und Verwaltern ermöglichen
- Der gleichzeitige Zugriff sollte die Leistung des Systems nicht negativ beeinflussen, was auch bei steigender Anzahl von Daten gewährleistet sein muss.

#### v) Hoher Grad an Datenunabhängigkeit:

- Ermöglichen von verschiedenen Systemen zur Veröffentlichung und Bearbeitung von Begegnungen und Tipps (Web-Anwendung für die Tippspieler, Desktop-Anwendung für Verwalter), welche auf dieselben Daten zugreifen.

	Lehrveranstaltung	<b>Grundlagen von Datenbanken</b>		WS 2018/19
	Aufgabenzettel	<b>1 (Lösungsvorschläge)</b>		
	Gesamtpunktzahl	<b>40</b>		
	Ausgabe	<b>Mi. 17.10.2018</b>	Abgabe	<b>Fr. 02.11.2018</b>

### 3 Transaktionen

[8 P.]

Im Folgenden ist eine Überweisung in Pseudocode von einem Konto mit der ID 5 auf das Konto mit der ID 7 skizziert:

```
BEGIN Ueberweisung
  UPDATE Konto
    SET Saldo = Saldo - 1000
  WHERE KontoID = 5
```

-- Zeitpunkt A --

```
UPDATE Konto
  SET Saldo = Saldo + 1000
  WHERE KontoID = 7
```

```
PRINT Konto
  WHERE KontoID = 7
```

-- Zeitpunkt B --

```
PRINT Konto
  WHERE KontoID = 5
```

END

Zum Zeitpunkt A bzw. zum Zeitpunkt B kommt es zu einem Stromausfall. Welche Folgen hat der jeweils resultierende Systemabsturz? Achten sie darauf, dass geänderte Daten nicht notwendigerweise sofort auf die Platte geschrieben werden. Wie können problematische Folgen verhindert werden, wenn der Vorgang in einem Datenbanksystem abgewickelt wird?

#### Lösungsvorschlag:

a) Der Vorgang wird nicht in einem Datenbanksystem abgewickelt und das Anwendungssystem trifft keine zusätzlichen Recovery-Maßnahmen:

- **Systemabsturz zum Zeitpunkt A:**

- **1. Annahme:** Es wurden bereits Änderungen persistent gemacht. Von Konto 5 wurden 1000 Geldeinheiten abgezogen, konnten jedoch aufgrund des Systemabsturzes nicht mehr auf Konto 7 gutgeschrieben werden. Das Geld „verschwindet“ und der Systemzustand wird dadurch inkonsistent.
- **2. Annahme:** Es wurden bislang keine Änderungen persistent gemacht. Es entstehen keine Änderungen am System. Der Systemzustand bleibt konsistent.

	Lehrveranstaltung	<b>Grundlagen von Datenbanken</b>		WS 2018/19
	Aufgabenzettel	<b>1 (Lösungsvorschläge)</b>		
	Gesamtpunktzahl	<b>40</b>		
	Ausgabe	<b>Mi. 17.10.2018</b>	Abgabe	<b>Fr. 02.11.2018</b>

- **Systemabsturz zum Zeitpunkt B:**

- **1. Annahme:** Es wurden bereits Änderungen persistent gemacht. 1000 Geldeinheiten wurden von Konto 5 abgezogen und auf Konto 7 gutgeschrieben. Der eigentliche Überweisungsvorgang ist damit abgeschlossen. Der Systemzustand bleibt konsistent, allerdings können in der realen Welt dennoch Inkonsistenzen auftreten, da der zweite PRINT-Befehl nicht ausgegeben wurde.
- **2. Annahme:** Es wurden bislang keine Änderungen persistent gemacht. Es entstehen keine Änderungen am System. Der Systemzustand bleibt konsistent. Es wurden jedoch teilweise Daten über den PRINT-Befehl an die Außenwelt weitergegeben, die nicht dem Systemzustand entsprechen. Diese können weitere Auswirkungen auf die Realwelt haben.


b) Der gesamte Vorgang wird als Transaktion in einem Datenbanksystem abgewickelt:

- **Systemabsturz zum Zeitpunkt A:**

- **1. Annahme:** Es wurden bereits Änderungen persistent gemacht. Von Konto 5 wurden 1000 Geldeinheiten abgezogen, konnten jedoch aufgrund des Systemabsturzes nicht mehr auf Konto 7 gutgeschrieben werden. Das Datenbanksystem erkennt, dass die Transaktion nicht erfolgreich abgeschlossen wurde, und setzt die geänderten Kontodaten auf ihren Zustand vor Beginn der Transaktion zurück (UNDO-Recovery). Somit wird das System in seinen letzten transaktionskonsistenten Zustand zurückversetzt.
- **2. Annahme:** Es wurden bislang keine Änderungen persistent gemacht. Es entstehen keine Änderungen am System. Der Systemzustand bleibt konsistent. Es sind keine UNDO- oder REDO-Recovery-Maßnahmen nötig.

- **Systemabsturz zum Zeitpunkt B:**

- **1. Annahme:** Es wurden bereits Änderungen persistent gemacht. 1000 Geldeinheiten wurden von Konto 5 abgezogen und auf Konto 7 gutgeschrieben. Der eigentliche Überweisungsvorgang ist damit abgeschlossen. Das Datenbanksystem erkennt jedoch, dass die Transaktion nicht erfolgreich abgeschlossen wurde (der PRINT-Vorgang fehlt noch) und setzt die geänderten Kontodaten trotzdem auf ihren Zustand vor Beginn der Transaktion zurück (UNDO-Recovery). Somit wird das System in seinen letzten transaktionskonsistenten Zustand zurück versetzt. Es wurden jedoch teilweise Daten über den PRINT-Befehl an die Außenwelt weitergegeben, die nicht dem Systemzustand entsprechen. Diese könnten ggf. weitere Auswirkungen auf die Realwelt haben (Beispiel: Raketenabschuss).
- **2. Annahme:** Es wurden bislang keine Änderungen persistent gemacht. Es entstehen keine Änderungen am System. Der Systemzustand bleibt konsistent. Es sind keine UNDO- oder REDO-Recovery-Maßnahmen nötig. Es wurden jedoch teilweise Daten über den PRINT-Befehl an die Außenwelt weitergegeben, die nicht dem Systemzustand entsprechen. Diese könnten ggf. weitere Auswirkungen auf die Realwelt haben.

	Lehrveranstaltung	<b>Grundlagen von Datenbanken</b>		WS 2018/19
	Aufgabenzettel	<b>1 (Lösungsvorschläge)</b>		
	Gesamtpunktzahl	<b>40</b>		
	Ausgabe	<b>Mi. 17.10.2018</b>	Abgabe	<b>Fr. 02.11.2018</b>

## 4 Warm-Up MySQL

[10 P.]

Um einige der späteren Übungen auch praktisch durchführen zu können, soll zur Lösung der Aufgaben eine Datenbank benutzt werden. Als verwendete Datenbank kommt dabei eine MySQL-Datenbank zum Einsatz. Laden Sie sich dafür zum Verbinden mit der Datenbank die MySQL-Workbench von

<http://www.mysql.com/downloads/workbench>

herunter, installieren Sie das Tool und machen Sie sich mit Hilfe der in den Übungen ausgegebenen Anleitung mit der Workbench vertraut. Verbinden Sie sich zur Datenbank-Server [vsisls4.informatik.uni-hamburg.de](http://vsisls4.informatik.uni-hamburg.de) (Nur über Fachbereichs-VPN und aus dem Informatikernetz erreichbar!) mit denen in der Übung ausgegebenen Userkennungen.

- a) Führen Sie nach dem ersten Verbinden mit der Datenbank folgende Anfragen aus (ersetzen Sie dabei *gdb\_gruppeXXX* durch die für Ihre Gruppe geltende Kennung!):

```
CREATE TABLE gdb_gruppeXXX.user (
  id INT PRIMARY KEY, name VARCHAR(49) NOT NULL, passwort VARCHAR(8) NOT NULL
);
INSERT INTO gdb_gruppeXXX.user (id, name, passwort)
VALUES (1, "gdbNutzer", "geheim");
```

Ist die Anfrage mit Hilfe der Workbench erfolgreich abgeschlossen, haben Sie ihre erste SQL-Anfrage erfolgreich abgeschlossen. Aktualisieren Sie jetzt die Ansicht in der Workbench und beschreiben Sie in einfachen Worten, was ihrer Meinung nach geschehen ist. (2 Punkte)

### Lösungsvorschlag:

Nach Absenden der Befehle wird eine neue „Tabelle“ *user* angelegt, welche die drei „Spalten“ *id*, *name* und *passwort* besitzt. Nach dem Anlegen wird zusätzlich noch ein neuer Datensatz mit der Id 1, dem Namen *gdbNutzer* und dem Passwort *geheim* in die neue „Tabelle“ eingefügt.

- b) Führen Sie als Zweites die folgenden Anfragen aus (ersetzen Sie auch hier *gdb\_gruppeXXX* durch die für Ihre Gruppe geltende Kennung!):

```
SELECT * FROM gdb_gruppeXXX.user WHERE name = "gdbNutzer";
DROP TABLE gdb_gruppeXXX.user;
```

Aktualisieren Sie nach Absenden dieser Befehle wieder die Ansicht in der Workbench und beschreiben Sie in einfachen Worten, was ihrer Meinung nach geschehen ist. (2 Punkte)

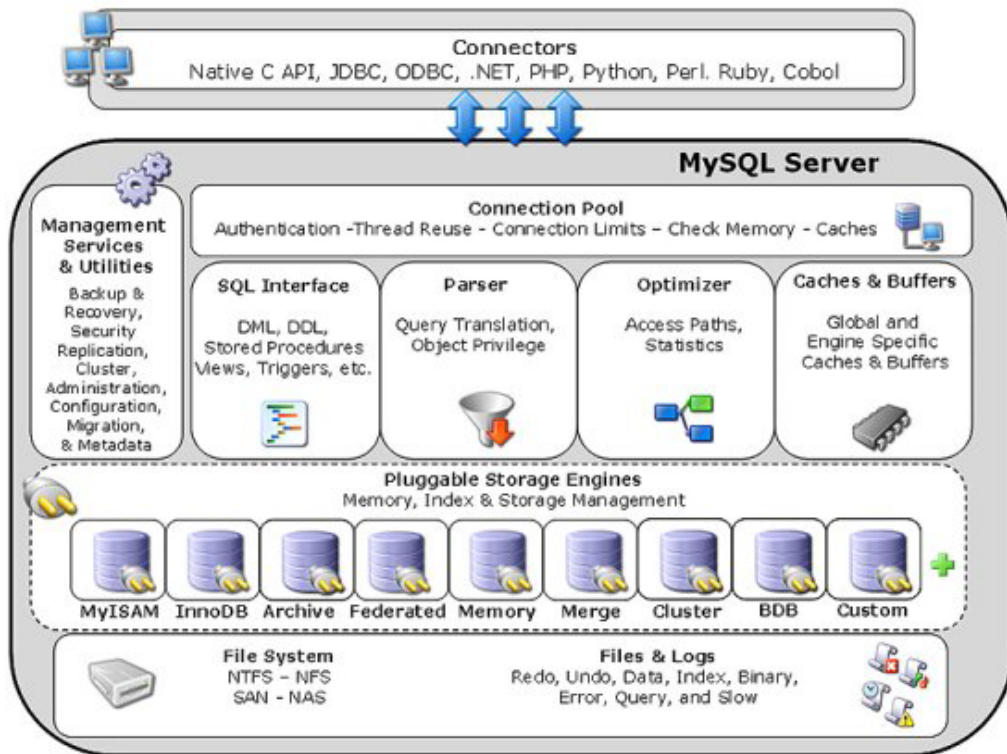
### Lösungsvorschlag:

Nach Absenden der Befehle werden alle „Zeilen“ der „Tabelle“ *user* ausgegeben, welche mit dem Namen *gdbNutzer* übereinstimmen. Anschließend wird die Tabelle gelöscht.



	Lehrveranstaltung	<b>Grundlagen von Datenbanken</b>		WS 2018/19
	Aufgabenzettel	<b>1 (Lösungsvorschläge)</b>		
	Gesamtpunktzahl	<b>40</b>		
	Ausgabe	<b>Mi. 17.10.2018</b>	Abgabe	<b>Fr. 02.11.2018</b>

c) Vergleichen Sie die unterhalb skizzierte Architekturübersicht<sup>1</sup> mit der in der Vorlesung vorgestellten Drei-Schema-Architektur nach ANSI-SPARC (6 Punkte).



### Lösungsvorschlag:

Die ANSI-SPARC Drei-Schema-Architektur baut auf folgenden Schemata auf:

- Einem **internen Schema**, welches die systemspezifische Realisierung der Datenbank der Datenbank spezifiziert. In der MySQL ist dies durch *Storage Engines* (und alles in der Ebene unterhalb der Storage Engines) definiert, welche die systemspezifische Speicherung der Daten im Speicher sowie der Festplatte realisieren.
- Einem **konzeptuellen Schema**, welches das Datenmodell implementierungsunabhängig spezifiziert. SQL ist hierbei die Anfragesprache zur Definition des Datenmodells und zur Erstellung von Anfragen an das Schema. In der Architekturübersicht der MySQL ist dafür auch eine entsprechende SQL-Schnittstelle vorgesehen.
- Einem **externen Schema**, welches anwendungsspezifische Sichten auf das konzeptuelle Schema definiert. Dies ist in der Architekturübersicht der MySQL etwas versteckt, wird aber durch die SQL-Schnittstelle (Views) und durch die Definition von entsprechenden Rechten sowie der Authorisierung über den Connection Pool realisiert.

<sup>1</sup><http://dev.mysql.com/doc/refman/5.1/de/custom-engine-overview.html>, 2010