	Lehrveranstaltung	Grundlagen von Datenbanken			WS 2014/15
	Aufgabenzettel	1			
	STiNE-Gruppe 19	Meimerstorf, Jochens, Töter			
	Ausgabe	Mi. 18.10.2017	Abgabe	Fr. 03.11.2017	

1 Informationssysteme

1.1 Charakterisierung

Ein Informationssystem ist ein informationstechnologisch unterstütztes Anwendungssystem, also im Endeffekt ein Computersystem, welches betriebliche Aufgaben ausführt. Charakterisierend für ein Datenbanksystem ist eine leichte Handhabbarkeit der Daten, die Kontrolle der Datenintegrität und die Kontrolle über die operationalen Daten, außerdem im heutigen Status in Zeiten von Amazon und Google ist auch die Leistung und Skalierbarkeit der Daten wichtig. Die Skalierbarkeit, da eine Datenbank für eine kleine Webanwendung sowie auch für ein großes DB-System wie Amazons Nutzerverwaltung funktionieren muss. **Aufgaben:**

- 1) Organisieren von Arbeitsabläufen
- 2) Paralleler Zugriff von mehreren Stellen
- 3) Hoher Durchsatz und kurze Antwortzeit

1.2 Datenunabhängigkeit


Die Datenunabhängigkeit bezeichnet die Unabhängigkeit der Daten und den dazugehörigen Anwendungsprogrammen. Realisiert wird dies meist, dadurch dass ein DMS, eine Schnittstelle zu den Daten bereitstellt welche die Anwendungsprogramme dann verwenden können.

Daraus folgt auch direkt die physische Unabhängigkeit, welche darstellt, dass die Anwendungsprogramme nicht verändert werden müssen wenn sich die physische Speicherungsart der Daten auf der Festplatte ändert. Da sich die Schnittstelle nicht ändert.

Die logische Unabhängigkeit bezeichnet, die Robustheit der Anwendungsprogramme gegen Änderungen im Datenbankschema. Dies ist auf jeden Fall bei SQL Datenbanken nicht der Fall, da wenn ich mein Schema verändere ich auch meine SQL-Queries verändern muss.

1.3 Beispiele


- a) SAP System in Unternehmen : Arbeitsabläufe leichter machen.
- b) Googles Suchmaschine : Das Internet indizieren und danach mit Anfragen von Usern durchsuchen
- c) Geldautomat: Geld abheben, Kontoauszug

	Lehrveranstaltung	Grundlagen von Datenbanken			WS 2014/15
	Aufgabenzettel	1			
	STiNE-Gruppe 19	Meimerstorf, Jochens, Töter			
	Ausgabe	Mi. 18.10.2017	Abgabe	Fr. 03.11.2017	

2 Miniwelt

2.1 relevante Objekte

- Spieler
 - Anmelden Anwendung
 - Tippgemeinschaft erstellen
 - Wettbewerbe anlegen
 - Begegnungen zu Wettbewerb hinzufügen
 - Ergebnisse eintragen
 - Spieler zu Gemeinschaft hinzufügen/entfernen
 - Tipps abgeben
 - Sich über Begegnung informieren, Punkte einsehen etc.
- Wettbewerb
 - Hat einen Gründer
 - Gehört zu einer Tippspielgemeinschaft
- Tippgemeinschaft
 - Hat verwaltenden Gründer
 - Hat n-Wettbewerbe
 - Hat n-Mitspieler

	Lehrveranstaltung	Grundlagen von Datenbanken WS 2014/15		
	Aufgabenzettel	1		
	STiNE-Gruppe 19	Meimerstorf, Jochens, Töter		
	Ausgabe	Mi. 18.10.2017	Abgabe	Fr. 03.11.2017

- Punktestand für Spieler

- Begegnung

- Gehört zu 1 Wettbewerb

- Hat 1 Ergebnis

- Hat n-Tipps

- Tipp

- Hat einen diskreten Wert

- Gehört zu einer Begegnung

- Wird von Spieler erstellt

2.2 Anforderungen

3 Transaktionen

3.1 Zeitpunkt A


3.1.1 Dateisystem

a) Daten im RAM:

Daten im RAM gehen verloren, die gesamte Transaktion muss wiederholt werden.

b) Daten auf Platte:

Die Daten wurden bereits geschrieben, heiSSt wir haben nun das Problem das wir unseren Vorgang nicht einfach wiederholen können da von KontoID = 5 bereits das Geld abgebucht wird. Beim Dateisystem wissen wir allerdings nicht wie weit unsere Transaktion war. Dadurch ist es schwer die Transaktion ohne weitere Mechanismen zu wiederholen.

	Lehrveranstaltung	Grundlagen von Datenbanken			WS 2014/15
	Aufgabenzettel	1			
	STiNE-Gruppe 19	Meimerstorf, Jochens, Töter			
	Ausgabe	Mi. 18.10.2017	Abgabe	Fr. 03.11.2017	

3.1.2 Datenbanksystem

- a) Daten im RAM:
Die Daten im RAM gehen verloren, die gesamte Transaktion muss wiederholt werden.
- b) Daten auf Platte:
Die Daten wurden bereits geschrieben, heißt also das Kontoid = 5 hat bereits 1000 Geld weniger. Das DBM muss nun also genauso wie das Dateisystem von Zeitpunkt A an weitermachen, das geht jedoch gegen das Prinzip der Atomarität (Im Späteren erklärt).

3.2 Zeitpunkt B

3.2.1 Dateisystem


- a) Daten im RAM:
Die Daten gehen verloren die gesamte Transaktion muss wiederholt werden.
- b) Daten auf Platte:
Die Daten wurden geschrieben. Es werden allerdings die Daten nicht noch einmal ausgegeben. Diesen Fall könnte man also ignorieren, da der essentielle Teil der Operation durchgeführt wurde. Und einfach die Daten beim nächsten Versuch nur noch einmal ausgeben.

3.2.2 Datenbanksystem

- a) Daten im RAM:
Die Daten gehen verloren die gesamte Transaktion muss wiederholt werden.
- b) Daten auf Platte:
Die Daten wurden geschrieben, und das Geld erfolgreich verbucht. Die Daten sind also konsistent. Die Transaktion wurde jedoch nicht komplett durchgeführt also muss nach dem Atomaritätsprinzip die Transaktion nochmal durchgeführt werden.

3.3 DBM Mechanismen

- Durch die Atomarität von Transaktionen in einem DBM, muss eine Transaktion immer ganz oder garnicht durchgeführt werden. Wie bei Zeitpunkt B, wo zwar alles schon geschrieben ist und nur die Ausgabe fehlt.

	Lehrveranstaltung	Grundlagen von Datenbanken			WS 2014/15
	Aufgabenzettel	1			
	STiNE-Gruppe 19	Meimerstorf, Jochens, Töter			
	Ausgabe	Mi. 18.10.2017	Abgabe	Fr. 03.11.2017	

- Dadurch dass die Transaktion durch das DBM durchgeführt wird, müssen wir als Nutzer uns nicht darum kümmern (Im Gegensatz zum Dateisystem) dass eine Anfrage wiederholt wird, wenn es zu Fehler kommt, sondern wir können uns darauf verlassen, dass unsere Anfrage komplett ausgeführt wird, wenn wir sie ans DBM übergeben haben.

4 Warm-Up SQL

4.1 Create/Insert

Mit dem Befehl CREATE TABLE wird eine Tabelle mit dem Namen user in der Datenbank gdb_Gruppe erstellt. Diese enthält dem Befehl entsprechend drei verschiedene Einträge, eine ID welche als primärer Key für die Tabelle verwendet wird, außerdem einen 49 Zeichen langen Character der nicht leer sein darf wenn ein Objekt in die Tabelle eingefügt wird, und als letztes noch das password, welches 8 Zeichen lang sein kann und auch nicht leer sein darf.

Mit dem zweiten Befehl INSERT INTO wird ein Datensatz bestehend aus (ID,Name,Passwort) in die vorher erstellte Tabelle eingefügt.

4.2 SELECT/DROP

Mit dem Befehl SELECT * FROM table WHERE, werden alle Datensätze der Tabelle table ausgegeben bei denen die Bedingung welche nach WHERE steht, stimmt. Also in unserem Fall genau unseren Datensatz. Hierbei ist zu beachten, dass durch das * wir alle Spalten des Datensatzes mitgeliefert bekommen und nicht nur eine spezielle Spalte

Mit dem Befehl DROP können Tabellen komplett aus der Datenbank gelöscht werden, wobei hier darauf zu achten ist das DROP tatsächlich im Gegensatz zu TRUNCATE die Tabelle löscht und nicht nur den Inhalt der Tabelle.