## Der Lernende kennt den Unterschied von relationalen vs. nicht-relationale Datenbanken.

## Nicht-relationale Datenbanken sind speziell für bestimmte Anforderungen optimiert, hauptsächlich hinsichtlich des Erstellens von Daten (Stichwort soziale Netzwerke) und des Abrufens von Daten (Stichwort Suchmaschinen). Während bei relationalen Datenbanken vor allem die Konsistenz der Daten das wichtigste Merkmal ist.

## Der Lernende kennt die Funktionsweise der nicht-relationalen Datenbanken.

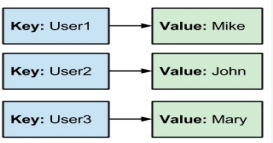
Sie verwenden für die Verwaltung der Daten keine Relationen, sondern andere Datenstrukturen.

Lassen sich in vier Untergruppen Teilen:

* Key-Value-Datenbanken
* dokumentenorientierte Datenbanken
* Graphdatenbanken
* spaltenorientierte Datenbanken

## Der Lernende kennt die Key-Value-Datenbank.

Key-Value-Datenbanken (auch: Key-Value-Stores) verwenden als Datenstruktur für die Speicherung von Daten sogenannte assoziative Arrays. Ein assoziatives Array besteht dabei aus einer Sammlung von Schlüssel-Wert-Paaren, in denen ein Schlüssel

als eindeutige Kennung zum Abrufen eines ihm zugeordneten Wertes dient.

Anwendung: speichern Key-Value-Datenbanken die Daten als eine einzige flache Sammlung ohne

Struktur oder Definition von Beziehungen zwischen Datensätzen.

## Der Lernende kennt die dokumentenorientierte Datenbank.

Ein Bild, das Text, Gerät enthält.

Automatisch generierte Beschreibungsind eine Sonderform von Key-Value-Datenbanken.  Die Werte sind hierbei immer sogenannte Dokumente, wobei nicht Dokumente im Sinne von Word-Dokumenten oder Ähnlichem gemeint sind, sondern Dokumente im Sinne einer strukturierten Zusammenstellung von Daten.

## Der Lernende kennt die Graphdatenbank.

Bei Graphdatenbanken werden die Daten in Form von Graphen gespeichert. Ein Graph besteht aus verschiedenen Knoten, die durch Kanten miteinander verbunden sind. Die Knoten repräsentieren einzelne Datensätze, die Kanten repräsentieren die Beziehungen zwischen diesen Datensätzen, wobei zwischen einseitigen Beziehungen (gerichtete Kanten) und beidseitigen Beziehungen (ungerichtete Kanten) unterschieden wird.

##### Ein Bild, das Tisch enthält. Automatisch generierte BeschreibungDer Lernende kennt die spaltenorientierte Datenbank.

Spaltenorientierte Datenbanken (Column-oriented Databases) sind Datenbanken, bei denen die Daten in Spalten gespeichert werden. Im Unterschied zu relationalen Datenbanken sind die einzelnen Spalten nicht Teil einer Tabelle, sondern werden separat

verwaltet. Spaltenorientierte Datenbanken eignen sich immer dann gut, wenn nur auf bestimmte Eigenschaften bzw. Spalten von Datensätzen zugegriffen werden muss, beispielsweise zur Auswertung von Daten.

## Der Lernende kennt die Anbindung der nicht relationalen Datenbank an einen Express-Server und kann diese mit Node.js umsetzen.

Ein Bild, das Text enthält.

Automatisch generierte BeschreibungNpm install mysql2….Connection sorgt für die verbindung.

##### Der Lernende kennt verschiedene Authentifizierungsmechanismen und kann diese mit Node.js umsetzen.

Basic-, Session Based- und Token Based Authentication

## Der Lernende kennt die Basic Authentication.

Nach erfolgreicher Authentifizierung die Credentials mit jeder Anfrage an den Server gesendet werden

##### Der Lernende kennt die Session Based Authentication.

Nach erfolgreicher Authentifizierung vom Server eine Session-ID generiert wird, die der Client fortan mit jeder Anfrage mitsendet und die der Server eindeutig einer Session zuordnen kann

## Der Lernende kennt die Token Based Authentikation.

Nach erfolgreicher Authentifizierung vom Server ein Token generiert wird, das anschliessend vom Client mit jeder Anfrage an den Server gesendet wird und das die vollständigen Session-Informationen des Nutzers enthält.

##### 

Der Lernende kennt die Authentifizierung mit Passport bei Express und kann diese umsetzen.

##### Der Lernende weiss, wie Webanwendungen getestet werden.

Nach erfolgreicher Authentifizierung vom Server ein Token generiert wird, das anschliessend vom Client mit jeder Anfrage an den Server gesendet wird und das die vollständigen Session-Informationen des Nutzers enthält.  
Der Lernende weiss, was Saubere Schnittstellen, Refactoring und Testbarkeit beim Testen sind.

**Saubere Schnittstellen:** Wenn Sie zuerst die Tests schreiben, bevor Sie mit der Implementierung beginnen machen Sie sich mehr Gedanken darüber, wie die Schnittstelle (API) des zu testenden Codes aussehen muss.

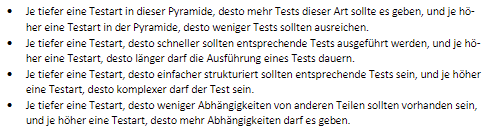
**Refactoring**: Automatisierte Tests stellen sicher, dass bei Änderungen und Optimierungen an der entsprechenden Komponente (dem sogenannten Refactoring) die Funktionalität die gleiche bleibt und noch genauso funktioniert wie in den Tests gefordert.

**Testbarkeit**: Indem Sie zuerst Tests schreiben, stellen Sie sicher, dass Ihr Code einfacher zu testen ist. Code, für den Sie erst nachträglich Tests schreiben, ist häufig schwierig zu testen.

Der Lernende kennt verschiedene Arten von Tests.

Siehe Pyramide. + Kompatibilitätstests, Performancetests, Sicherheitstests

### Der Lernende weiss, was Komponententests, Integrationstests, End-toEnd-Tests sind.

Unit: testen einzelne Komponenten auf Codeebene wie beispielsweise Klassen oder Funktionen.

Integration: testen das Zusammenspiel verschiedener Komponenten einer Anwendung.

End-to-End: testen Sie eine Anwendung von einem Ende (dem Frontend) bis zum anderen Ende (dem Backend)

Der Lernende weiss, was Kompatibilitätstests, Performancetests und Sicherheitstests sind.

Kompatibilitätstests: Da wird getestet, ob die Webanwendung auf verschiedenen Browsern oder Betriebssystemen anwendbar ist.

Performance: Wie sich eine Anwendung bei grosser Auslastung oder extremer Auslastung verhält.

Sicherheitstests: Über Sicherheitstests können sie ermitteln, ob eine Webapplikation Schwachstellen bezüglich der Sicherheit aufweist.

##### Der Lernende weiss, was Unittesting ist und kann das mit dem assert-Modul umsetzen.

## Der Lernende weiss, war die Testpyramide ist.