

Linux 2 DEVOPS 2020

Lektion 6

Idag

- Databaser
- SQL- och NoSQL-databaser
- MySQL
- MongoDB
- Mitt-i-kursen-utvärdering

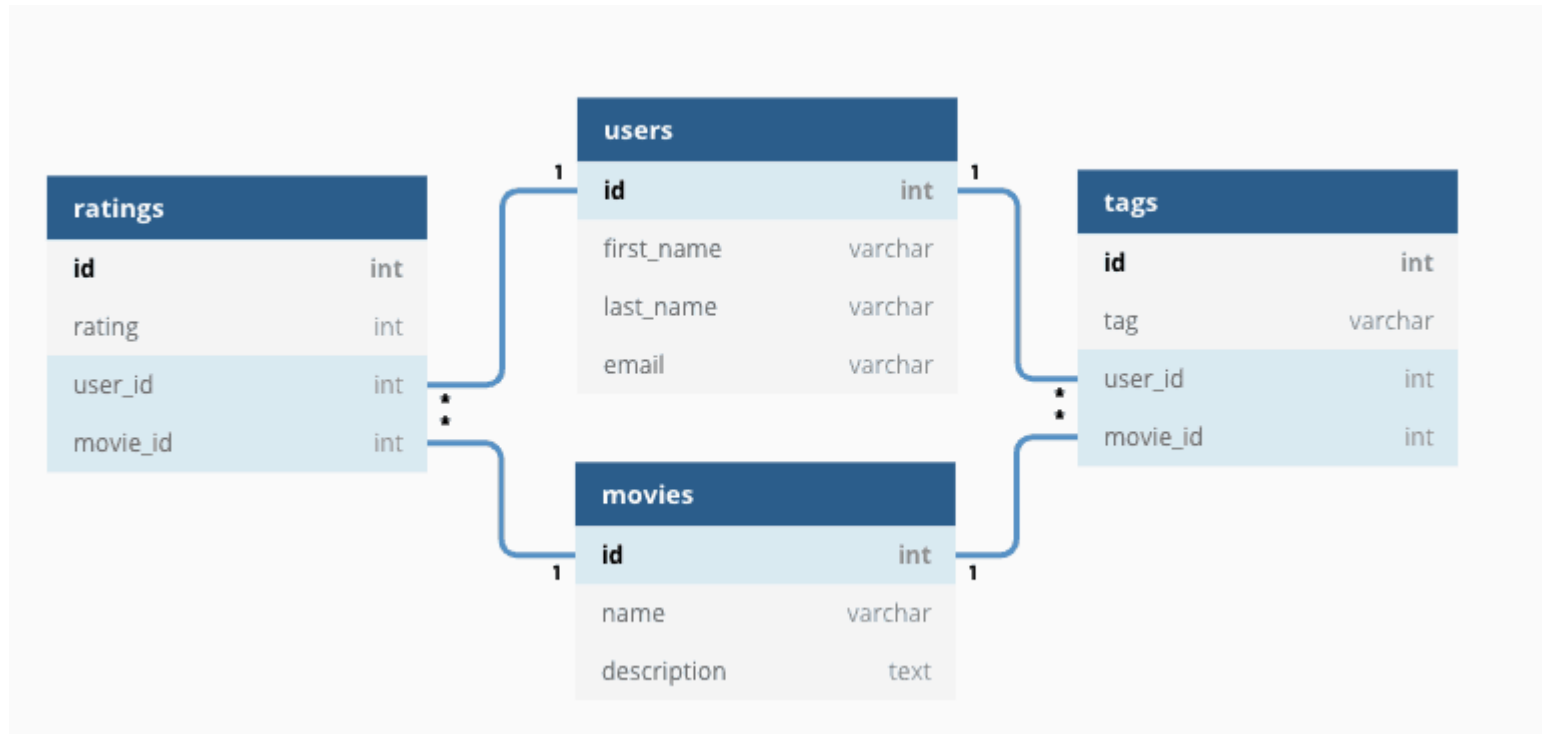
Databaser

- Egentligen varje ansamling av data, t ex textdokument
- I sammanhanget menar man oftast relationsdatabaser (SQL-databaser)
- Även andra sätt att organisera data för sökningar (NoSQL-databaser)

Relationsdatabaser

- Data organiserat i tabeller
 - Rader (poster)
 - Kolumner (fält)
- Relationer mellan tabeller
 - Beroenden mellan fält

Relationsdatabaser



Exempel på enkel struktur för relationsdatabas.
Stulet från omnici.com

Relationsdatabaser

- Normalform
 - Se till att data delas upp i tabeller så att det inte upprepas i onödan och så att varje tabell har ett tydligt dataset
- Underlätta snabba sökningar
- Joins av tabeller

SQL

- Hantera data i relationsdatabaser
- Sök, lägg in, ändra, radera

SQL Exempel

- Sök

```
select * from users
```

- Lägg in

```
insert into users (first_name, last_name, email) values  
( 'Archibald', 'Haddock', 'haddock@moulinsart.be' )
```

- Ändra

```
update users set email='tintin@moulinsart.be' where  
first_name='Tintin'
```

- Radera

```
delete from users where last_name='Dupont'
```


Databasmotor

- Programmet som möjliggör själva databasen
 - Databasfunktioner
 - Användarhantering
 - Lagringshantering
- Exempel (SQL): MySQL, Oracle, MS SQL Server, PostgreSQL
- Exempel (NoSQL): MongoDB, CouchDB, Neo4j

MySQL

- Relationsdatabaser
- Snabb och anses enkel
- Används ofta för databaser bakom webbsiter
- Open source

MySQL

- mysqld
- Data lagras per default i /var/lib/mysql
- /etc/mysql/my.cnf
 - Här finns bl a datadir angivet
- Defaultport 3306
- Loggar per default till /var/log/mysql/error.log

Övning 1

- Installera MySQL (det finns paket för apt) och starta den
 - Paketet heter mysql-server
- Kontrollera att du kan gå in i kommandoradsinterfacet
- Titta i /etc/mysql/
 - Särskilt my.cnf

Övning 1

```
mo@sputnik:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 45
Server version: 5.5.62-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

MySQL: databaser och användare

- Skapa databas

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name  
[create_option]
```

- Skapa användare

```
CREATE USER [IF NOT EXISTS] user [auth_option] [, user  
[auth_option]] ...
```

```
DEFAULT ROLE role [, role ] ...
```

```
[REQUIRE {NONE | tls_option [[AND] tls_option] ...}]
```

```
[WITH resource_option [resource_option] ...]
```

```
[password_option | lock_option] ...
```

MySQL: databaser och användare

- Användarrättigheter

GRANT priv_type [(column_list)]

[, priv_type [(column_list)]] ...

ON [object_type] priv_level

TO user [auth_option] [, user [auth_option]] ...

[REQUIRE {NONE | tls_option [[AND]
tls_option] ...}]

[WITH {GRANT OPTION | resource_option} ...]

MySQL: databaser och användare

Exempel

```
mysql> create database mydb;
```

```
mysql> create user myuser1 identified by  
"ohemlig";
```

```
mysql> grant all privileges on mydb.* to  
'myuser1';
```

```
mysql> grant select on mydb.* to 'myuser2';
```


Övning 2

- Skapa en databas testdb1 på din MySQL-server.
- Skapa en användare dbuser1 som får fullständiga rättigheter i databasen testdb1 och även får skapa nya användare där.
- Låt dbuser1 skapa några testtabeller, använd förslagsvis tidigare bild som din databasdesign.

Övning 2

```
mysql> create database testdb1;  
mysql> create user dbuser1 identified by "losen1";  
mysql> grant all privileges on testdb1.* to  
'dbuser1';  
mysql> grant create user on *.* to 'dbuser1';  
  
prompt$ mysql -u dbuser1 -p
```

Övning 2

```
mysql> create table users (id int auto_increment  
primary key, first_name varchar(60), last_name  
varchar(80), email varchar(50));
```

```
mysql> create table movies (id int auto_increment  
primary key, name varchar(60), description text);
```

```
mysql> create table ratings (id int auto_increment  
primary key, rating int, user_id int, movie_id int,  
foreign key (user_id) references users(id), foreign key  
(movie_id) references movies(id));
```

```
mysql> create table tags (id int auto_increment primary  
key, tag varchar(40), user_id int, movie_id int,  
foreign key (user_id) references users(id), foreign key  
(movie_id) references movies(id));
```

Användarrättigheter

- Läsa – select
- Skriva – insert
- Uppdatera – update
- all / "all privileges"
- Skapa dbs, tabeller etc – create

Användarrättigheter

Privilege	Meaning and Grantable Levels
<u>ALL [PRIVILEGES]</u>	Grant all privileges at specified access level except <u>GRANT OPTION</u> and <u>PROXY</u> .
<u>ALTER</u>	Enable use of <u>ALTER TABLE</u> . Levels: Global, database, table.
<u>ALTER ROUTINE</u>	Enable stored routines to be altered or dropped. Levels: Global, database, routine.
<u>CREATE</u>	Enable database and table creation. Levels: Global, database, table.
<u>CREATE ROLE</u>	Enable role creation. Level: Global.
<u>CREATE ROUTINE</u>	Enable stored routine creation. Levels: Global, database.
<u>CREATE TABLESPACE</u>	Enable tablespaces and log file groups to be created, altered, or dropped. Level: Global.
<u>CREATE TEMPORARY TABLES</u>	Enable use of <u>CREATE TEMPORARY TABLE</u> . Levels: Global, database.
<u>CREATE USER</u>	Enable use of <u>CREATE USER</u> , <u>DROP USER</u> , <u>RENAME USER</u> , and <u>REVOKE ALL PRIVILEGES</u> . Level: Global.
<u>CREATE VIEW</u>	Enable views to be created or altered. Levels: Global, database, table.
<u>DELETE</u>	Enable use of <u>DELETE</u> . Level: Global, database, table.
<u>DROP</u>	Enable databases, tables, and views to be dropped. Levels: Global, database, table.
<u>DROP ROLE</u>	Enable roles to be dropped. Level: Global.
<u>EVENT</u>	Enable use of events for the Event Scheduler. Levels: Global, database.
<u>EXECUTE</u>	Enable the user to execute stored routines. Levels: Global, database, routine.
<u>FILE</u>	Enable the user to cause the server to read or write files. Level: Global.
<u>GRANT OPTION</u>	Enable privileges to be granted to or removed from other accounts. Levels: Global, database, table, routine, proxy.
<u>INDEX</u>	Enable indexes to be created or dropped. Levels: Global, database, table.
<u>INSERT</u>	Enable use of <u>INSERT</u> . Levels: Global, database, table, column.
<u>LOCK TABLES</u>	Enable use of <u>LOCK TABLES</u> on tables for which you have the <u>SELECT</u> privilege. Levels: Global, database.
<u>PROCESS</u>	Enable the user to see all processes with <u>SHOW PROCESSLIST</u> . Level: Global.
<u>PROXY</u>	Enable user proxying. Level: From user to user.
<u>REFERENCES</u>	Enable foreign key creation. Levels: Global, database, table, column.
<u>RELOAD</u>	Enable use of <u>FLUSH</u> operations. Level: Global.
<u>REPLICATION CLIENT</u>	Enable the user to ask where source or replica servers are. Level: Global.
<u>REPLICATION SLAVE</u>	Enable replicas to read binary log events from the source. Level: Global.
<u>SELECT</u>	Enable use of <u>SELECT</u> . Levels: Global, database, table, column.
<u>SHOW DATABASES</u>	Enable <u>SHOW DATABASES</u> to show all databases. Level: Global.
<u>SHOW VIEW</u>	Enable use of <u>SHOW CREATE VIEW</u> . Levels: Global, database, table.
<u>SHUTDOWN</u>	Enable use of <u>mysqladmin shutdown</u> . Level: Global.
<u>SUPER</u>	Enable use of other administrative operations such as <u>CHANGE MASTER TO</u> , <u>KILL</u> , <u>PURGE BINARY LOGS</u> , <u>SET GLOBAL</u> , and <u>mysqladmin debug</u> command. Level: Global.
<u>TRIGGER</u>	Enable trigger operations. Levels: Global, database, table.
<u>UPDATE</u>	Enable use of <u>UPDATE</u> . Levels: Global, database, table, column.
<u>USAGE</u>	Synonym for "no privileges"

Från dev.mysql.com

Övning 3

- Lägg nu upp två nya användare, dbuser2 och dbuser3. En av dem skall få skriva och ändra i testdb1, en skall bara få läsa.
- Testa att lägga in data respektive söka data, så att du ser att rättigheterna stämmer.

Övning 3

```
mysql> create user dbuser2 identified by  
"losen2";
```

```
mysql> create user dbuser3 identified by  
"losen3";
```

```
mysql> grant select, insert, update on testdb1.*  
to 'dbuser2';
```

```
mysql> grant select on testdb1.* to 'dbuser3';
```

Koppla till databas från applikation

- Skapa connection med inloggning, skicka sql, få tillbaka data
- Tänk på att i lämpliga fall sätta upp användare för publik app / webbsite / klient respektive för administration av databasens innehåll
- Kom ihåg att använda bind-parametrar, dvs undvika att ta något som skickats in av användaren rakt av
 - Prepared statement

Koppla från applikation

Java-exempel

```
public class MySqlBindVariableTest {
    public static void main(String[] args) {
        Connection conn = null;
        PreparedStatement prepStmt = null;
        ResultSet rs = null;
        try {
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            String connectionUrl = "jdbc:mysql://localhost:3306/testdatabase";
            String connectionUser = "testuser";
            String connectionPassword = "testpassword";
            conn = DriverManager.getConnection(connectionUrl, connectionUser, connectionPassword);
            String sqlStmt = "SELECT * FROM employees where last_name=? and first_name like ?";
            System.out.println("SQL Statement:\n\t" + sqlStmt);
            prepStmt = conn.prepareStatement(sqlStmt);
            System.out.println("Prepared Statement before bind variables set:\n\t" + prepStmt.toString());
            prepStmt.setString(1, "Doe");
            prepStmt.setString(2, "%n%");
            System.out.println("Prepared Statement after bind variables set:\n\t" + prepStmt.toString());
            rs = prepStmt.executeQuery();
            while (rs.next()) {
                String id = rs.getString("id");
                String firstName = rs.getString("first_name");
                String lastName = rs.getString("last_name");
                System.out.println("ID: " + id + ", First Name: " + firstName + ", Last Name: " + lastName);
            }
        } catch (Exception e) {
```

Exempel stulet från avajava.com

MySQL: dump av data

- Ta ut data till en fil (t ex för backup eller för att kopiera en databas till en ny server): `mysqldump`

```
mysqldump -u <user> -p --databases <db>
```

- Skriver per default till stdout, så man behöver även skicka resultatet till en fil

Exempel:

```
mysqldump -u root -p --databases mydb >dump.sql
```

- Läs in genom att exekvera sql-kommandona i dumpen, t ex med `mysql -u <user> -p <dump.sql`

Övning 4

- Gör en dump av din databas testdb1 till filen testdb1.sql
- Titta i filen för att se sql-kommandon som återskapar databasen

Övning 4

```
mysqldump -u dbuser1 -p --databases testdb1 >testdb1.sql
```

Övning 5

- Tänk er en databas i MySQL avsedd för en webbshop. Den kommer ha många tabeller, men för övningen räcker det med att ni skapar en tabell för produkter.
- Till shoppen hör, förutom webbsite tillgänglig för hela världen, ett admin-interface för dem som får lägga in och ändra data om produkter i databasen.
- Skapa de användare som är lämpliga för att implementera MySQL-delen av detta scenario. Fundera på vad de skall kunna göra, och ge dem rätt rättigheter för det.

Övning 5

```
mysql> create database webshop;
```

```
mysql> use webshop;
```

```
mysql> create table articles (id int auto_increment primary  
key, name varchar(40), long_name varchar(160), description  
text, price decimal, category int);
```

```
mysql> create user webuser identified by "Aveivdef#1";
```

```
mysql> grant select on webshop.* to 'webuser';
```

```
mysql> create user admuser identified by "Mealof1bos&p";
```

```
mysql> grant select, insert, update, delete on webshop.* to  
'admuser';
```

NoSQL-databaser

- Ikke-relationsdatabaser
- I sin tur olika typer:
 - Document store
 - Key-value
 - Graph

NoSQL: Key-value

"A key–value database, or key–value store, is a data storage paradigm designed for storing, retrieving, and managing associative arrays, and a data structure more commonly known today as a dictionary or hash table." (Wikipedia)

- Nyckel till dataset, sökningar endast på nyckeln
- Vanligt med data i RAM för snabbhetens skull
- Exempel: dbm, Apache Ignite, MemcacheDB

NoSQL: Document store

”A document-oriented database, or document store, is a computer program designed for storing, retrieving and managing document-oriented information, also known as semi-structured data.”
(Wikipedia)

- Ett specialfall av key – value
- Dokument i JSON, BSON, XML...
- Kan göra sökningar utifrån dokumentens struktur
- Exempel: MongoDB, CouchDB, DocumentDB

NoSQL: Graph

"In computing, a graph database (GDB) is a database that uses graph structures for semantic queries with nodes, edges, and properties to represent and store data. A key concept of the system is the graph (or edge or relationship)."
(Wikipedia)

- Noder och länkar ("edges") mellan dessa
- Sociala grafer, intressegrafer m fl
- Exempel: Neo4j, TerminusDB, RedisGraph

MongoDB

- Dokumentdatabas
- Collections uppbyggda i json
- Inga joins etc!
- Open source

MongoDB

- mongod
- Data lagras per default i /var/lib/mongodb
- /etc/mongodb.conf
 - Här finns bl a dbpath angivet
- Defaultport 27017, lokalt webbinterface på 28017
- Loggar per default till
/var/log/mongodb/mongodb.log

Övning 6

- Installera MongoDB (det finns paket för apt) och starta den
 - Paketet heter mongod
- Kontrollera att du kan gå in i kommandoradsinterfacet

Övning 6

```
mo@sputnik:~$ mongo
MongoDB shell version: 2.4.9
connecting to: test
> |
```

MongoDB

- En databas skapas så snart man börjar använda den

```
use mydb
```

- Skapa collection (en "tabell" men ändå inte)

```
db.createCollection("<colname>")
```

- En collection skapas också så snart man börjar lägga in data
- Records ("rader" men ändå inte) i en collection, notera att det inte finns några fördefinierade fält

MongoDB

- Sök data med find

```
db.<collection>.find()
```

```
db.<collection>.find( { <villkor> } )
```

- Stoppa in data med insert

```
db.<collection>.insert( { <data> } )
```


MongoDB Exempel

```
> use testdb
```

```
> db.newCol.insert( { "first_name" : "Archibald",  
"last_name" : "Haddock", "email" :  
"haddock@moulinsart.be" } );
```

```
> db.newCol.find()
```

```
> db.newCol.find( { "first_name" : "Tintin" } )
```

Övning 7

- Skapa en databas testdb i MongoDB.
- Skapa en collection för users med first_name, last_name, email i denna databas och lägg in lite data.

Övning 7

```
> use testdb
```

```
> db.users.insert( { "first_name" : "Tintin",  
"last_name" : "Tintin", "email" :  
"tintin@moulinsart.be" } );
```

```
> db.users.insert( { "first_name" : "Milou",  
"last_name" : "Hund", "email" :  
"milou@moulinsart.be" } );
```

Användare i MongoDB

- Skapa användare

```
db.createUser ( { user: "<username>", pwd: "<pwd>" } )
```

- Sätta rättigheter

- När användaren skapas:

```
db.createUser ( { user: "<username>", pwd: "<pwd>",  
roles: [ { role: "<role>", db: "<dbname>" } ] } )
```

- Med grantRolesToUser

```
db.grantRolesToUser( "<username>", [ <roles> ] )
```

Användare i MongoDB, exempel

```
> db.createUser( { user: "testuser1", pwd:
"losen1", roles: [ { role: "readWrite", db:
"testdb" } ] } )
```

```
> db.createUser( { user: "testuser2", pwd:
"losen1", roles: [ "readWrite", "dbAdmin",
"userAdmin" ] } )
```

Övning 8

- Lägg till en användare muser1 som har fullständiga rättigheter i din databas, och en användare muser2 som endast har läsrättigheter.
- Testa att lägga in data respektive söka data, så att du ser att rättigheterna stämmer.

Övning 8

```
> db.createUser( { user: "muser1", pwd:  
"losen1", roles: [ "readWrite", "dbAdmin",  
"userAdmin" ] } )
```

```
> db.createUser( { user: "muser2", pwd:  
"losen2", roles: [ "read" ] } )
```

MongoDB: dump av data

- Ta ut data till en fil (t ex för backup eller för att kopiera en databas till en ny server): `mongodump`
`mongodump`
- Skriver per default ut allt till ett directory dump med underdirectories för respektive databas
- Läs in med `mongorestore`
`mongorestore <directory>`
- Alternativ (mer finkorning) metod: `mongoexport` + `mongoimport`

Övning 9

- Gör en dump av din MongoDB-databas
- Titta på de filer som skapas

Övning 9

mongodump

MySQL – MongoDB

- Skillnader i hur man tänker kring databasdesign
 - MySQL (relationsdatabas) i tabeller enligt hur datat förhåller sig till vartannat
 - MongoDB (dokumentdatabas) i collections där man kan ha nästlat data – men inte slå ihop sökningar av flera collections på en gång
- Likheter i mycket kring drift av databaser ...

Övning 10

- Antag att ni jobbar med både MySQL- och MongoDB-databaser. Ni skall nu göra en checklista för vad man vill ha koll på i drift för dessa.
- Lista saker som skall hanteras för båda typerna
- Lista olikheter i hantering

Övning 10

- Uppsättningar för att se till att rätt saker körs på servern – och att inte onödiga program ligger där.
- Öppna för rätt portar (3306 för MySQL, 27017 och i förekommande fall för vissa kopplingar till 28017 för MongoDB)
- Håll koll på loggar (olika loggar, men samma princip)
- Håll koll på när data växer (olika ställen, men samma princip)
- Backupbehov
- Användare i databaser
- Användare på servern

Mitt-i-kursen-utvärdering



- Vad borde vi sluta göra?
- Vad borde vi börja göra?
- Vad fungerar bra?