TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Database lab

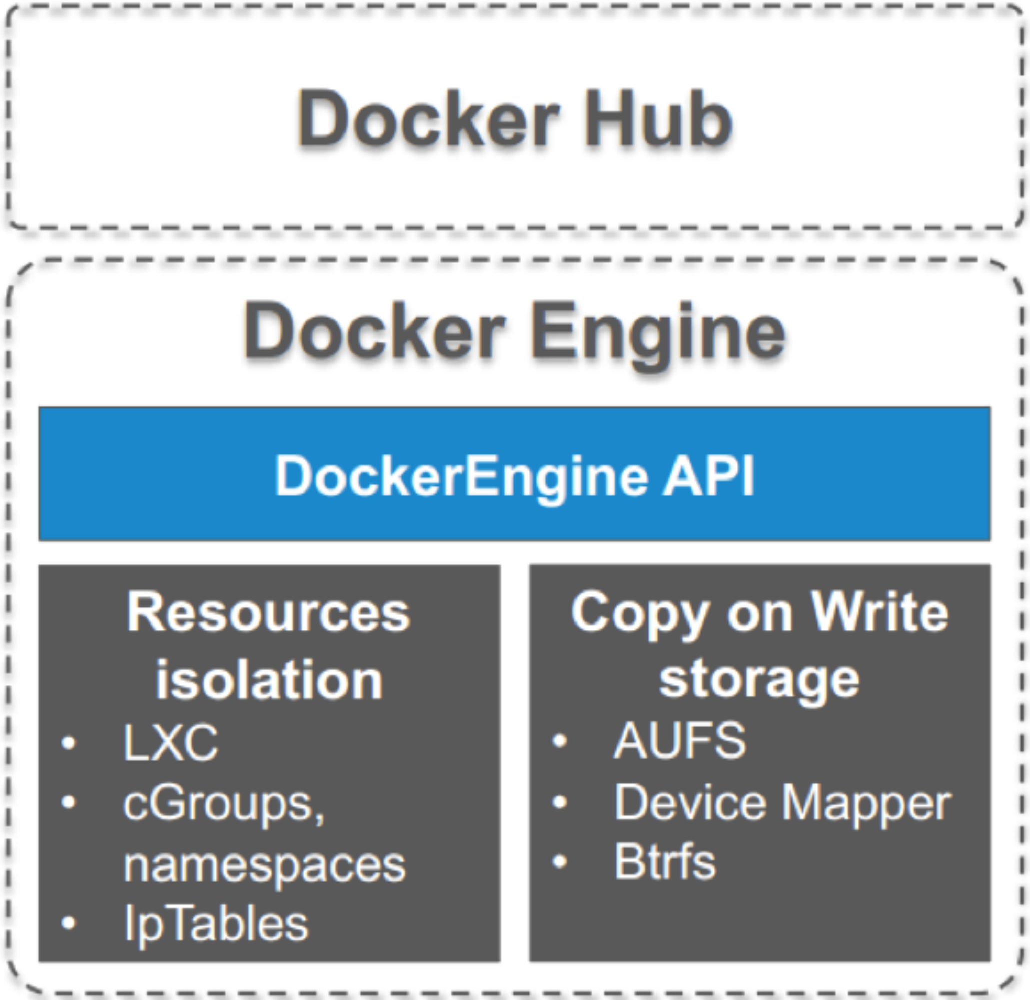
Viet-Trung Tran
http://is.hust.edu.vn/~trungtv/

Preparation

- Join our student group
  - https://www.facebook.com/groups/trungtv.students/
  - For Q&A, announcement, exam scores
- In case of urgency (no emergency)
  - trungtv@soict.hust.edu.vn
- MySQL or Mariadb

What is docker

- « A technology enabling to standardize packaging and deployment operations »
- Key properties
  - Portable
    - Limited adherence with the subsequent OS
    - Only a recent Linux kernel is required
    - Docker lightweight agent to be installed
  - Disposable
    - Reduce building and deployment costs
  - Live
    - Versioned and incremental images
  - Community
    - Public registries



Docker Hub

Docker Engine

DockerEngine API

Resources isolation

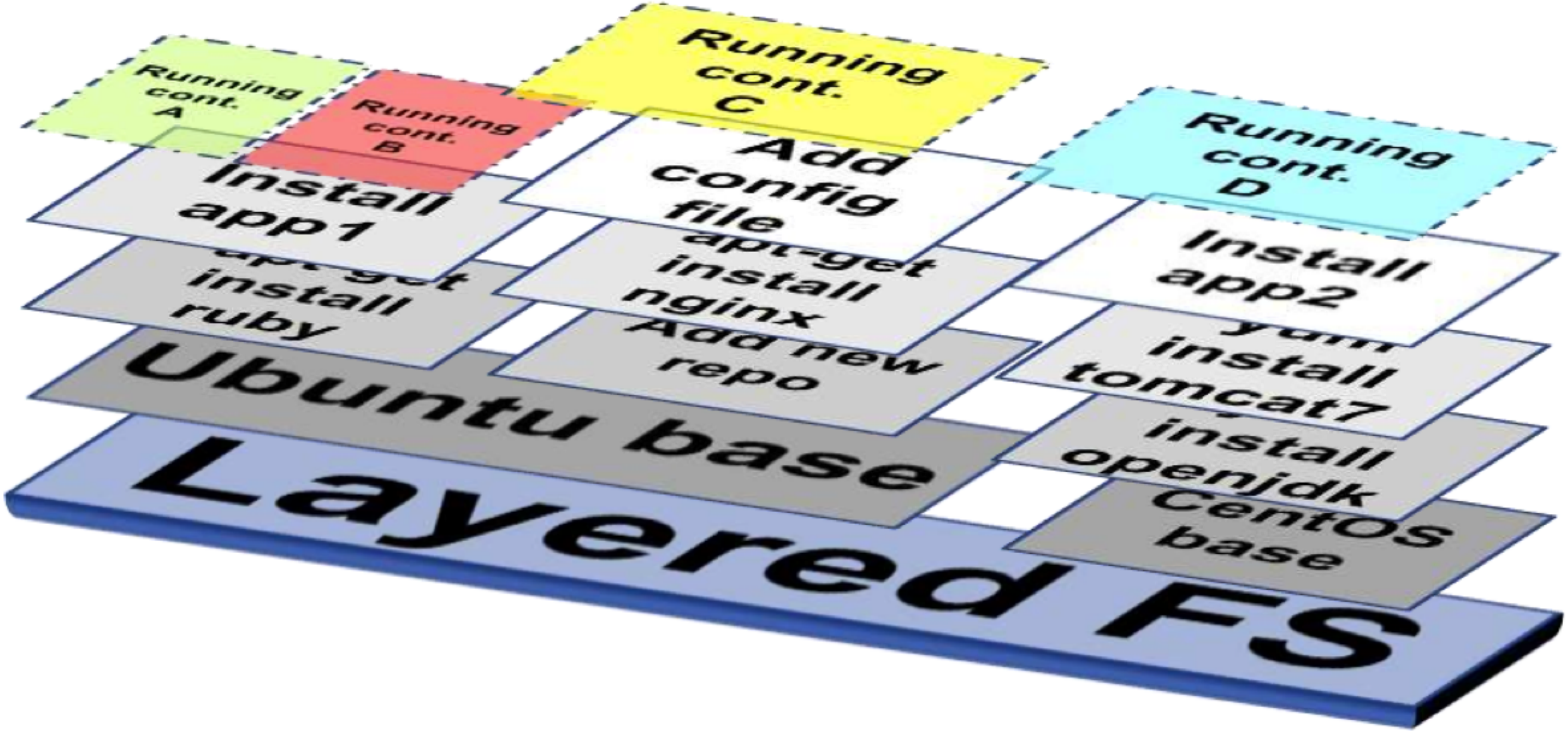
- LXC
- cGroups, namespaces
- IpTables

Copy on Write storage

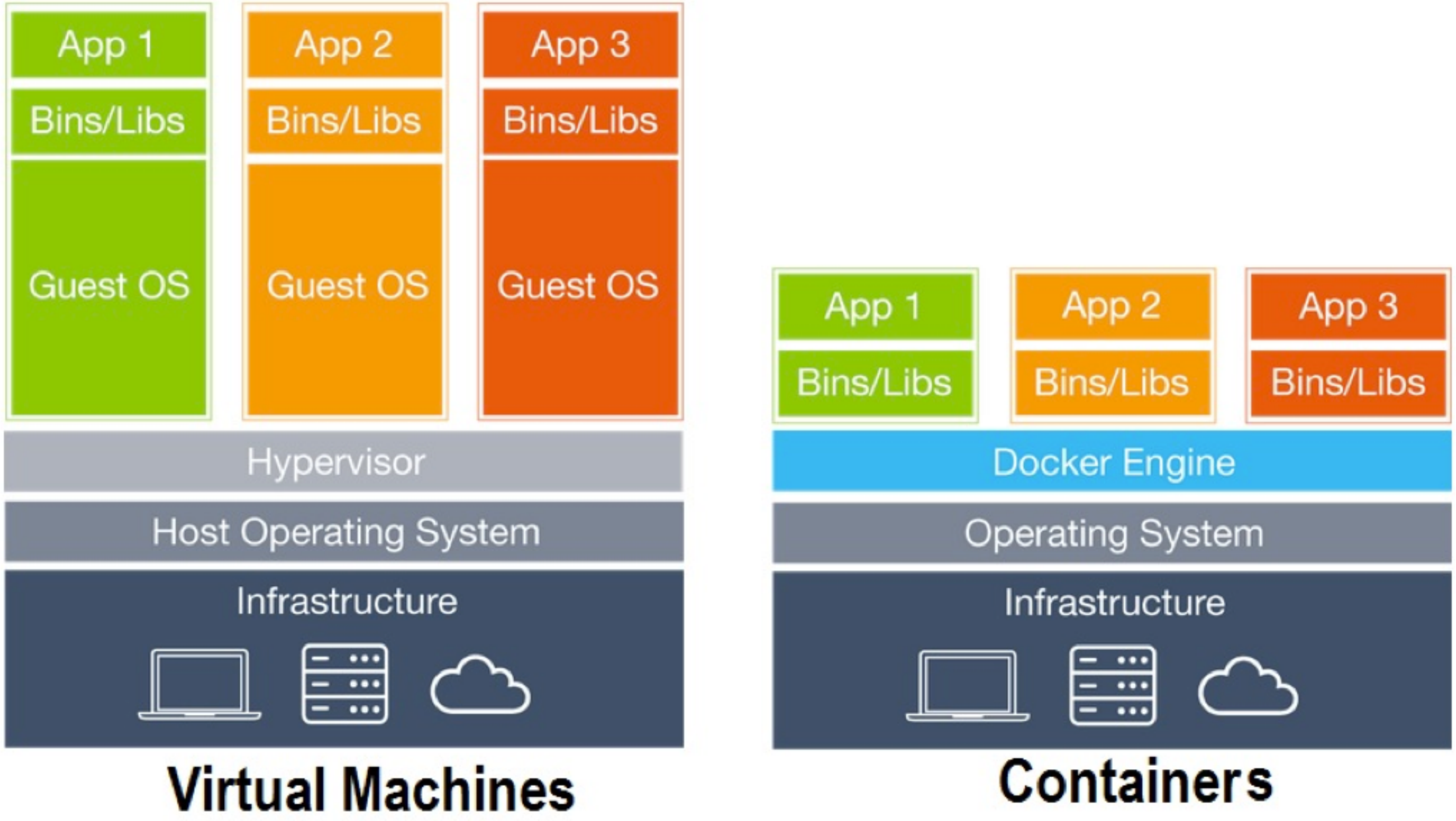
- AUFS
- Device Mapper
- Btrfs

Major actors

Versioning and incremental images

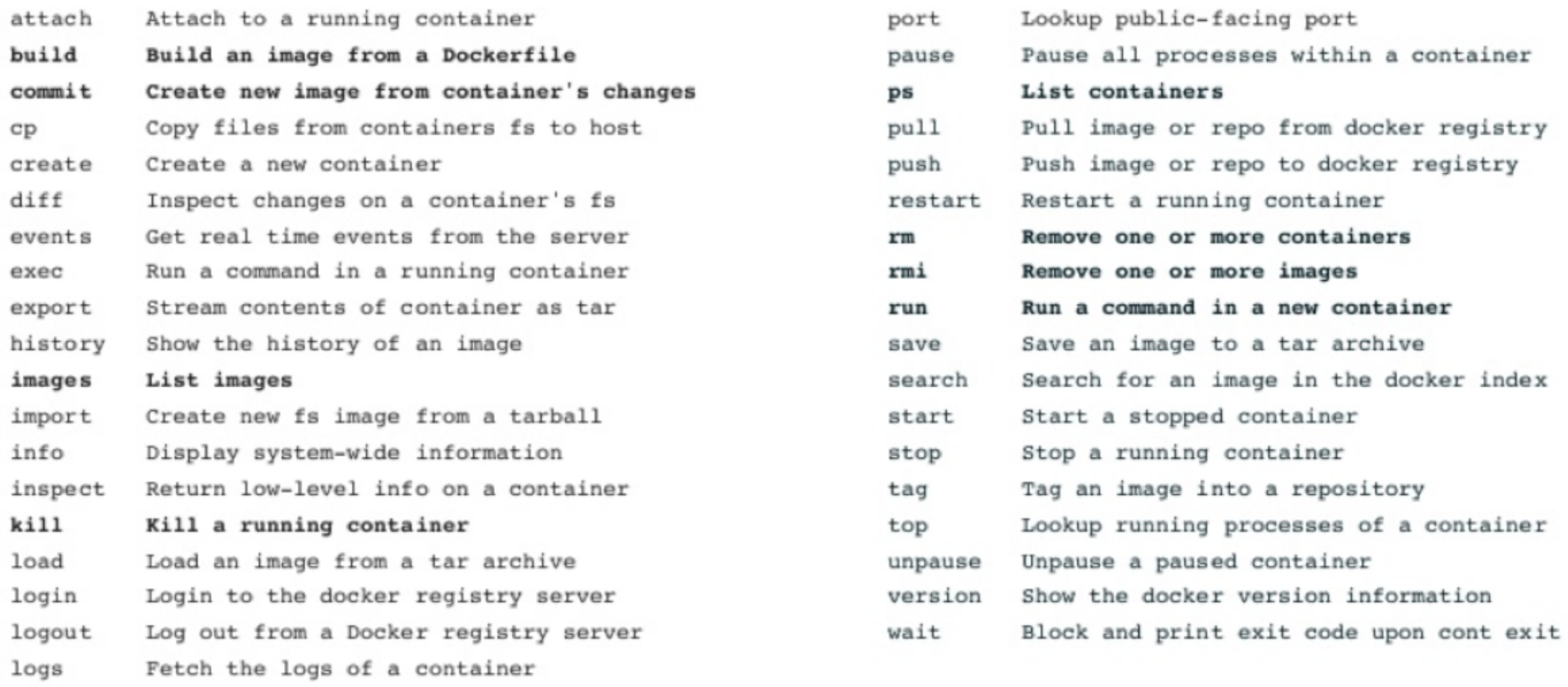


Docker containers vs. virtual machines



Docker installation & commands

- https://www.docker.com/community-edition



```
attach     Attach to a running container
build      Build an image from a Dockerfile
commit     Create new image from container's changes
cp         Copy files from containers fs to host
create     Create a new container
diff       Inspect changes on a container's fs
events     Get real time events from the server
exec       Run a command in a running container
export     Stream contents of container as tar
history    Show the history of an image
images     List images
import     Create new fs image from a tarball
info       Display system-wide information
inspect    Return low-level info on a container
kill       Kill a running container
load       Load an image from a tar archive
login      Login to the docker registry server
logout     Log out from a Docker registry server
logs       Fetch the logs of a container
port       Lookup public-facing port
pause      Pause all processes within a container
ps         List containers
pull       Pull image or repo from docker registry
push       Push image or repo to docker registry
restart    Restart a running container
rm         Remove one or more containers
rmi        Remove one or more images
run        Run a command in a new container
save       Save an image to a tar archive
search     Search for an image in the docker index
start      Start a stopped container
stop       Stop a running container
tag        Tag an image into a repository
top        Lookup running processes of a container
unpause    Unpause a paused container
version    Show the docker version information
wait       Block and print exit code upon cont exit
```

Database lab setup

Mariadb overview

- MariaDB, a branch of MySQL created in the wake of Oracle's purchase of Sun Microsystems.
- GPLv2 license, a lot of added community development
- Feature enhanced
- Application compatible & feature compatible with MySQL

Docker install mariadb

- https://hub.docker.com/_/mariadb/
- Pull mariadb image
  - docker pull mariadb
- Start a new container
  - docker run --name some-mariadb -e MYSQL_ROOT_PASSWORD=my-secret-pw -d mariadb:tag
- Access to running container
  - Docker exec –it some-mariadb bash

References

- Getting started:
  - http://www.mysqltutorial.org/basic-mysql-tutorial.aspx
  - http://www.sqlteaching.com/
  - https://www.codecademy.com/courses/learn-sql

- Related tutorials:
  - MySQL-CLI: https://www.youtube.com/playlist?list=PLfdtiltiRHWEw4-kRrh1ZZy_3OcQxTn7P
  - Analyzing Business Metrics: https://www.codecademy.com/learn/sql-analyzing-business-metrics
  - SQL joins infografic: http://cd64.de/sql-joins
- Tools:
  - DataGrip: https://www.jetbrains.com/datagrip/
  - Sequel Pro: http://www.sequelpro.com/

Commands

- Access monitor:`mysql -u [username] -p;`(will prompt for password)
- Show all databases:`show databases;`
- Access database:`mysql -u [username] -p [database]`(will prompt for password)
- Create new database:`create database [database];`
- Select database:`use [database];`
- Determine what database is in use:`select database();`
- Show all tables:`show tables;`
- Show table structure:`describe [table];`
- Create new table with columns:`CREATE TABLE [table] ([column] VARCHAR(120), [another-column] DATETIME);`
- Adding a column:`ALTER TABLE [table] ADD COLUMN [column] VARCHAR(120);`

Commands (con’t)

- Adding a column with an unique, auto-incrementing ID:`ALTER TABLE [table] ADD COLUMN [column] int NOT NULL AUTO_INCREMENT PRIMARY KEY;`
- Inserting a record:`INSERT INTO [table] ([column], [column]) VALUES ('[value]', [value]');`
- MySQL function for datetime input:`NOW()`
- Explain records:`EXPLAIN SELECT * FROM [table];`
- List all users:`SELECT User,Host FROM mysql.user;`
- Create new user:`CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';`
- Grant`ALL`access to user for`*`tables:`GRANT ALL ON database.* TO 'user'@'localhost’;`
- Find out the IP address: `SHOW VARIABLES WHERE Variable_name = 'hostname';`
- More on MySQL cheetsheet (included)