# Algemeen

## Begrippen

| Canvas | Synoniemen |
|---|---|
| Cursus / Course | Cursus Blok |
| Opdrachtengroep / Assignmentgroup | Module (=onderdeel van blok) |

## Studenten codes

Er is een functie beschikbaar, maar die is om veiligheidsredenen niet gepubliceerd.

## Hidden reports

| | |
|---|---|
| Aantal pogingen per opdracht per student | http://localhost:8080/query/attempts |
| Productiefste dagen van studenten | http://localhost:8080/query/day-of-week |
| Productiefste dagdelen van studenten | http://localhost:8080/query/dagdeel |
| Gemiddelde score per module | http://localhost:8080/query/moeilijk |
| Snel inleveren | http://localhost:8080/query/rapid |
| Access Log | http://localhost:8080/query/log |
| | |

## Nieuwe users

1. Gebruik script import_users.py (in repo onder /import) om users uit een Canvas-cursus te importeren.
   Het script maakt SQL-code.

   Let op gebruik van juiste database!

2. Gebruik Eduarte-docent (easy versie) om een CSV van een klas naar Excel te exporteren.
   Voeg deze regel aan het eind toe en voer queries uit.

```
="update user set klas='2B' where student_nr='"&A2&"';"
```

3. Maak unieke access codes voor de studenten.

   Codes zijn afhankelijk van een 'salt' deze kan alleen worden aangepast door de code aan te passen en moet in principe één keer per jaar worden veranderd waarna iedere student een nieuwe code krijgt.

```
Informatie verwijderd in verband met security
```

# Windows Dev Omgeving

## hosts

`C:\Windows\System32\drivers\etc\hosts`

```
127.0.0.1 c20.cmon.local
127.0.0.1 c21.cmon.local
127.0.0.1 c22.cmon.local
```

## vhosts

`C:\xampp\apache\conf\extra\httpd-vhosts.conf`

```
<VirtualHost *:80>
  ServerName cmon.local
  ServerAlias c21.cmon.local c22.cmon.local c20.cmon.local
  DocumentRoot "C:\Users\maxbi\www\yii2\canvas\web"
  RewriteEngine on
  RewriteCond %{SERVER_NAME} =c22.cmon.ovh [OR]
  RewriteCond %{SERVER_NAME} =cmon.ovh [OR]
  RewriteCond %{SERVER_NAME} =c21.cmon.ovh
  RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,NE,R=permanent]
</VirtualHost>
```

# Create view in DB

(view moet elk jaar worden aangepast)

View is nodig om een overzicht te krijgen van alle nagekeken opdrachten (per docent).

## Old

```
drop view all_submissions;

create view all_submissions as
SELECT *, 'c22' as cohort FROM `canvas-c20`.`submission`
union
SELECT *, 'c23' as cohort FROM `canvas-c21`.`submission`
UNION
SELECT *, 'c24' as cohort FROM `canvas-c22`.`submission`;
```

## New as from c23

```
DROP VIEW IF EXISTS `all_submissions`;

CREATE view all_submissions as

SELECT
d.id as module_id,
d.naam as module_name,
d.pos as module_pos,
d.generiek as generiek,
u.name as student_name,
u.student_nr as student_nr,
u.grade as grading_enabled,
g.name as grader_name,
c.korte_naam as cursus_short_name,
s.submitted_at as submitted_at,
s.graded_at as graded_at,
s.entered_score as entered_score,
r.minpunten as minpunten,
'c22' as cohort
from `canvas-c22`.submission s
inner join `canvas-c22`.assignment a on a.id=s.assignment_id
inner join `canvas-c22`.module_def d on d.id=a.assignment_group_id
inner join `canvas-c22`.user u on u.id=s.user_id
left outer join `canvas-c22`.user g on g.id=s.grader_id
inner join `canvas-c22`.course c on c.id=s.course_id
```

```
inner join `canvas-c22`.resultaat r on r.module_id=d.id and r.student_nummer = u.student_nr

UNION

SELECT
d.id as module_id,
d.naam as module_name,
d.pos as module_pos,
d.generiek as generiek,
u.name as student_name,
u.student_nr as student_nr,
u.grade as grading_enabled,
g.name as grader_name,
c.korte_naam as cursus_short_name,
s.submitted_at as submitted_at,
s.graded_at as graded_at,
s.entered_score as entered_score,
r.minpunten as minpunten,
'c21' COLLATE utf8mb4_general_ci as cohort
from `canvas-c23`.submission s
inner join `canvas-c23`.assignment a on a.id=s.assignment_id
inner join `canvas-c23`.module_def d on d.id=a.assignment_group_id
inner join `canvas-c23`.user u on u.id=s.user_id
Left outer join `canvas-c23`.user g on g.id=s.grader_id
inner join `canvas-c23`.course c on c.id=s.course_id
inner join `canvas-c23`.resultaat r on r.module_id=d.id and r.student_nummer = u.student_nr

union

SELECT
d.id as module_id,
d.naam as module_name,
d.pos as module_pos,
d.generiek as generiek,
u.name as student_name,
u.student_nr as student_nr,
u.grade as grading_enabled,
g.name as grader_name,
c.korte_naam as cursus_short_name,
```

```
s.submitted_at as submitted_at,
s.graded_at as graded_at,
s.entered_score as entered_score,
r.minpunten as minpunten,
'c24' COLLATE utf8mb4_general_ci as cohort
from `canvas-c24`.submission s
inner join `canvas-c24`.assignment a on a.id=s.assignment_id
inner join `canvas-c24`.module_def d on d.id=a.assignment_group_id
inner join `canvas-c24`.user u on u.id=s.user_id
left outer join `canvas-c24`.user g on g.id=s.grader_id
inner join `canvas-c24`.course c on c.id=s.course_id
inner join `canvas-c24`.resultaat r on r.module_id=d.id and r.student_nummer = u.student_nr
```

# Set Collate in all tables

```
ALTER TABLE assignment
CONVERT TO CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
ALTER TABLE assignment_group
CONVERT TO CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
ALTER TABLE course
CONVERT TO CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
ALTER TABLE check_in
CONVERT TO CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
ALTER TABLE log
CONVERT TO CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
ALTER TABLE login_user
CONVERT TO CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
ALTER TABLE module
CONVERT TO CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
ALTER TABLE module_def
CONVERT TO CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
ALTER TABLE resultaat
CONVERT TO CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
ALTER TABLE submission
CONVERT TO CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
ALTER TABLE `user`
CONVERT TO CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
```

# Database (main entities)

image-1666113019407.png

Onderstreepte entiteiten komen 1:1 uit de Canvas API.

*Resultaat* wordt berekend na de import en is een gegenereerde "de-normalized table" ten behoeve van de eenvoud en performance.

Een course (meestal blok) heeft meerdere assignement_groups. De tabellen module, assignement_group en module_def zijn feitelijk één tabel. De reden dat er drie tabellen zijn heeft te maken met de Canvas API, de module en assignement_group komen 1:1 uit de API en de module_def zijn eigen toevoegingen.

## In schema

image-1666113613887.png

# *Server*

## Backup files (www, home, en apache2)

Backup gaat via Restic naar Object Storage en wordt via crontab dagelijks gedraaid.

```
# backup to eu2.contabostorage.com (s3) and clean once a week
30 23 * * * /home/.../restic/restic-backup.sh > /home/max/log/restic-backup.log 2>&1
# clean up
55 23 * * 0 /home/.../restic/restic-clean.sh  > /home/max/log/restic-clean.log 2>&1
```

Er worden drie backup's gemaakt:

```
restic backup /var/www/       --exclude-file=exclude.txt
restic backup /home/.../      --exclude-file=exclude.txt
restic backup /etc/apache2/   --exclude-file=exclude.txt
```

Voor restoren van een snapshot zoek je eerst *snapshot id* op met `restic-list.sh`. Over het algemeen neem je de onderste uit de list tenzij je verder terug wilt.

Daarna het volgende commando invoeren

```
./restic.sh restore <id> --target /tmp/restore
```

restic.sh

```
#! /bin/bash

export AWS_ACCESS_KEY_ID="xxx-xxx-xxx"
export AWS_SECRET_ACCESS_KEY="xxx-xxx-xxx"
export RESTIC_REPOSITORY="s3:server/bucket"
export RESTIC_PASSWORD="xxx-xxx-xxx"

restic $@
```

restic manual: https://restic.readthedocs.io/en/latest/index.html

# Backup SQL

Gaat via dit script dat via Cron om 23:xx wordt aangeroepen.

```
#!/bin/bash

#####################################################################
##
##   MySQL Database Backup Script
##
#####################################################################

export PATH=/bin:/usr/bin:/usr/local/bin
TODAY=`date +"%d%b%Y"`
TODAY=`date +"%m%d-%H00"`

#####################################################################
################## Update below values  #######################

DB_BACKUP_PATH='/home/.../mysql/Backup-DB'
MYSQL_HOST='localhost'
MYSQL_PORT='3306'
MYSQL_USER='root of wat anders'
MYSQL_PASSWORD='xxxx-xxxx-xxxx'
DATABASE_NAMES='db1 db2 db3 db4 db5'
BACKUP_RETAIN_DAYS=180   ## Number of days to keep local backup copy

#####################################################################
```

```
for DATABASE_NAME in ${DATABASE_NAMES}
do

  echo "Backup started for database - ${DATABASE_NAME}"

  mkdir -p ${DB_BACKUP_PATH}

  mysqldump -h ${MYSQL_HOST} \
     -P ${MYSQL_PORT} \
     -u ${MYSQL_USER} \
     -p${MYSQL_PASSWORD} \
     ${DATABASE_NAME} | gzip > ${DB_BACKUP_PATH}/${DATABASE_NAME}-${TODAY}.sql.gz

  if [ $? -eq 0 ]; then
    echo "Database backup successfully completed"
  else
    echo "Error found during backup"
    exit 1
  fi

done

# remove all files older than 90 days excpet those made on the 1ste of the month
find ${DB_BACKUP_PATH} -type f -not -name '*01-????.sql.gz' -mtime +90 -exec rm {} \;
# remove all files not created at 23:00 and older than 8 days
find ${DB_BACKUP_PATH} -type f -not -name '*300.sql.gz' -mtime +8 -exec rm {} \;

##### Remove backups older than {BACKUP_RETAIN_DAYS} days  #####
find ${DB_BACKUP_PATH}/* -mtime +${BACKUP_RETAIN_DAYS} -exec rm {} \;
```

--

Revision #4
Created 2023-01-16 18:40:34 UTC by Max
Updated 2024-08-21 16:55:21 UTC by Max