

<u>Werken met SQLite in de</u> <u>catalografie</u>

29/04/2025

Doel/agenda

- Hoe exporteer je een Brocadelijst naar een SQLite databank?
- Hoe is een SQLite databank opgebouwd?
- Basissyntax voor zoekopdrachten?
- Een aantal voorbeelden uit de praktijk



Gebruikersproces : catexpsqlite

- Link : <u>https://anet.be/menu/catexpsqlite</u>
- Nodig
 - Bronlijst: lijst met clois
 - Export profiel : kies steeds 'cat' ==> je geeft via het export profiel aan dat je catalografische data (van cloi's) wil exporteren.
 - Kies steeds 'cat'
 - **Export generator** : bepaalt welke data naar buiten gebracht wordt en of deze eventueel verrijkt
 - kies steeds 'catall' voor data uit lvd
 - Alle data; vaste contents maar ook de lokale data (tentoonstellingsgegevens, veilinggegevens, ...)
 - Kies 'abkdg', 'abahs', 'abirua' voor catalografische data van repositories
 - IR's : projectinformatie, vakgroepen, bibliometrische data, ...



Gebruikersproces : catexpsqlite

- Link : <u>https://anet.be/menu/catexpsqlite</u>
- Nodig
 - Bronlijst: lijst met olois
 - **Export profiel** : kies steeds 'catobj' ==> je geeft via het export profiel aan dat je, startend vanuit een lijst met olois, de catalografische data (van de gelinkte cloi's) wil exporteren.
 - **Export generator** : bepaalt welke data naar buiten gebracht wordt en of deze eventueel verrijkt
 - kies steeds 'catall' voor data uit lvd
 - Alle data; vaste contents maar ook de lokale data (tentoonstellingsgegevens, veilinggegevens,
 - Niet van toepassing voor data uit institutional repositories



Brocade (zijsprongetje)

Gebruikersproces : catallsqlite

- Link : <u>https://anet.be/menu/catallsqlite</u>
- Nodig
 - **Bronlijst**: lijst met clois OF catalografisch regelwerk
 - Export profiel : kies steeds 'catobj'
 - Export generator : kies steeds 'catall' ==> Alle data, van clois en olois (uitgebreid objectensysteem!) wordt naar buiten gebracht
 - n.v.t. voor institutional repositories



Aanmaken van een bronlijst

- Manueel
- Na een selectie via een selectieprofiel
- Via een pregedefineerde lijst (<u>https://anet.be/doc/brocade/support/html/listsel.html</u>#)
 - Een aanvulling op gewone lijsten, vaak het resultaat van een selectie (x # handelingen)
 - Sls-lijsten ==> om met 1 instructie een selectie te maken
 - LVD (1): sls:lm:lvd:od ==> lijst met alle cloi's van regelwerk lvd met lidmaatschap 'od'
 - sls: identifier van de lijst
 - Im: code ~ selecteer op een lidmaatschap
 - Ivd : regelwerk/databank waarbinnen je selecteert
 - od: het lidmaatschap waarop je selecteert



Aanmaken van een bronlijst

- Manueel
- Na een selectie via een selectieprofiel
- Via een pregedefineerde lijst (<u>https://anet.be/doc/brocade/support/html/listsel.html</u>#)
 - LVD (1): sls:lm:lvd:od ==> lijst met alle cloi's van regelwerk lvd met lidmaatschap 'od'
 - LVD (2): **sls:select:modified[7]:sls:lm:lvd:zebra** ==> lijst van diezelfde cloi's met lidmaatschap 'zebra' die aangepast zijn de laatste 7 dagen
 - STCV : **sls:lm:stcv:sa** ==> lijst met alle cloi's van regelwerk stcv met lidmaatschap 'sa'
 - IRUA : **sls:select:modified[7]:sls:lm:irua:irua** ==> lijst met alle cloi's van regelwerk irua met lidmaatschap 'irua' ie aangepast zijn de laatste 7 dagen
 - IRKDG : **sls:lm:irkdg:irkdg** ==> lijst met alle cloi's van regelwerk irkdg
 - IRAHS : **sls:lm:irahs:irahs** ==> lijst met alle cloi's van regelwerk irahs



Gebruikersproces : catexpsqlite

- Link : <u>https://anet.be/menu/catexpsqlite</u>
- Demo aanmaak SQLite databank adhv "sls:lm:irahs:irahs"
- In de achtergrond:
 - Aanmaken van diverse brocade lijsten
 - Tabellen compileren tot een SQLite bestand ==> Hiermee gaan jullie zo meteen aan de slag ;-)

Wat is SQLite?

SQL = Structured Query Language

 In deze context: taal die gebruikt wordt om gegevens uit een relationele databank te bevragen.

SQLite = databankbeheersysteem

- Datasets eenvoudig te bevragen via SQL
- Hele databank (inhoud + settings + ...) als 1 file; geen server nodig; hangt niet vast aan 1 specifieke toepassing.
- Goed als bewaarformaat (1 vd 4 aangeraden bewaarformaten voor datasets voor LoC)



SQL basics: structuur van de zoekvraag

Specifieke operatoren in een specifieke volgorde

- **SELECT:** selectie van data (veldnamen)
- FROM: uit een bepaald deel van de databank (tabelnamen)
- WHERE: die voldoen aan een bepaalde voorwaarde
- ORDER BY / GROUP BY / LIMIT / ...: en die je op een bepaalde manier wil tonen of beperken.



SQL basics: best practice

- Operatoren in hoofdletters
- Verschillende lijnen
- Query eindigen met ;

1 2	SELECT note_nt FROM note
3	WHERE note_ty = 'cong'
4	ORDER BY note_nt;



SQL basics: SELECT ... FROM ...

Structuur	Effect
SELECT * SELECT impressum_pl	Alles selecteren. De kolom impressum_pl selecteren.
SELECT impressum_ug, impressum_pl,	De kolommen impressum_ ug en impressum_pl selecteren (in die volgorde).
SELECT impressum_pl AS plaats	De kolom impressum_pl selecteren en die de naam 'plaats' geven.
SELECT DISTINCT impressum_pl	De unieke plaatsen selecteren.
SELECT COUNT(impressum_pl)	Het aantal plaatsen selecteren.
SELECT COUNT(DISTINCT impressum_pl)	Het aantal unieke plaatsen selecteren.

FROM impressum;

SQL basics: SELECT ... FROM ... WHERE (NOT)

SELECT impressum_ju1sv AS jaar FROM impressum	Selecteer impressum_ju1sv met de naam 'jaar' uit de tabel impressum
WHERE impressum_ju2sv;	waar er ook een waarde is in kolom impressum_ju2sv.
WHERE jaar > 2020;	waar het jaar groter is dan 2020.
WHERE jaar < 2020;	waar het jaar kleiner is dan 2020.
WHERE jaar >= 2020;	waar het jaar groter is dan of gelijk is aan 2020.
WHERE jaar <= 2020;	waar het jaar kleiner is dan of gelijk is aan 2020.
WHERE jaar <> 2020 (of: WHERE jaar != 2020 of: WHERE NOT jaar = 2020)	waar het jaar niet gelijk is aan 2020.



SQL basics: SELECT ... FROM ... WHERE (NOT)

SELECT impressum_pl AS plaats FROM impressum	Selecteer impressum_pl met de naam 'plaats' uit de tabel impressum
WHERE plaats = 'Antwerpen'	waar plaats Antwerpen is.
WHERE plaats LIKE '%en'; WHERE plaats LIKE 'Ant_erpen';	waar plaats het patroon '%en' heeft. waar plaats het patroon 'Ant_erpen' heeft.
WHERE plaats IN('Antwerpen', 'Amsterdam', 'Leuven');	waar plaats Antwerpen, Amsterdam of Leuven is.
WHERE plaats = """;	waar plaats leeg is.



SQL basics: SELECT ... FROM ... WHERE (NOT)

SELECT impressum_ju1sv AS jaar, impressum_pl AS plaats FROM impressum	Selecteer impressum_ju1sv met de naam 'jaar' en impressum_pl met als naam 'plaats' uit de tabel impressum
WHERE jaar = 2020 OR jaar = 2021;	waar het jaar 2020 of 2021 is.
WHERE jaar BETWEEN 2020 AND 2023;	waar het jaar tussen 2020 (incl.) en 2023 (incl.) is.
WHERE jaar = 2020 AND plaats = 'Antwerpen';	waar het jaar 2020 is en de plaats Antwerpen is.
WHERE (jaar = 2020 OR jaar = 2021) AND (plaats = 'Brugge' OR plaats = 'Leuven');	waar het jaar 2020 of 2021 is, en waar de plaats Brugge of Leuven is.



SQL basics: SELECT ... FROM ... (WHERE) ... ORDER BY

SELECT impressum_ju1sv AS jaar, impressum_pl FROM impressum WHERE jaar BETWEEN 2020 AND 2022	Selecteer impressum_ju1sv met de naam 'jaar' en impressum_pl uit de tabel impressum met als jaar 2020, 2021 of 2022
ORDER BY jaar;	gesorteerd op jaar oplopend.
ORDER BY jaar DESC;	gesorteerd op jaar aflopend.
ORDER BY jaar, impressum_pl DESC;	Gesorteerd op jaar (oplopend), vervolgens op plaats (aflopend).



SQL basics: SELECT ... FROM ... (WHERE) ... GROUP BY

SELECT impressum_ju1sv AS jaar,	Selecteer impressum_ju1sv met de naam 'jaar' en het
COUNT(impressum_pl) AS aantal_plaatsen	aantal plaatsen met de naam 'aantal_plaatsen'
FROM impressum	uit de tabel impressum
GROUP BY jaar	groepeer het aantal plaatsen per jaar
ORDER BY jaar;	en sorteer per jaar (oplopend).

	jaar	aantal_plaatsen
188	2018	502
189	2019	590
190	2020	530
191	2021	513
192	2022	440
193	2023	481
194	2024	404
195	2025	39

SQL basics: SELECT ... FROM ... GROUP BY ... HAVING

SELECT impressum_pl AS plaats, COUNT(impressum_pl) AS aantal_keer FROM impressum	Selecteer impressum_pl als 'plaats' en het aantal keer dat de plaatsen voorkomen als 'aantal_keer' uit de tabel impressum
GROUP BY plaats HAVING aantal_keer > 50	groepeer per plaats waarbij het aantal keren dat deze plaats voorkomt groter is dan 50
ORDER BY aantal_keer DESC;	sorteer op aantal in dalende volgorde.

-		
	plaats	aantal_keer
1	Bruxelles	3554
2	Antwerpen	2229
3	Brugge	1404
4	Mechelen	982
5	Brussel	954
6	Gent	863
7	Limal	718

SQL tabellen combineren: LEFT JOIN

- SELECT title_ti, impressum_pl
 FROM title
 LEFT JOIN impressum USING(cloi);
- --> de tabel title
- --> combineer met de tabel impressum op 'cloi' (gemeenschappelijk)
- --> selecteer title_ti en impressum_pl.
- Ook mogelijk:

LEFT JOIN impressum **ON** title.cloi = impressum.cloi



Voorbeelden uit de praktijk: onderwerpen

- Welke onderwerpen gebruiken wij bij deze deelcollectie?
 - Maak een lijst in Brocade die de juiste deelcollectie bevat
 - Exporteer als SQLite
 - Voer volgende query uit:

```
SELECT subject_vw AS 'UDC-onderwerp', COUNT(*) AS aantal
FROM subject
WHERE subject_ty="U"
GROUP BY subject_vw
ORDER BY aantal DESC;
```



Voorbeelden uit de praktijk: exemplaarannotaties

Veilingcatalogi met exemplaarannotatie <PPKK>

- Maak een lijst in Brocade van de veilingcatalogi
- Exporteer als SQLite
- Voer volgende query uit:

SELECT cloi AS record, oloi AS object, object_an AS annotatie FROM holdingobject WHERE annotatie LIKE "%<PPKK>%";



Voorbeelden uit de praktijk: spreadsheet op maat

- Geef me alles van vóór 1701 met volgende velden (...)
 - Maak een lijst in Brocade van c-loi's van jouw instelling van vóór 1701
 - Exporteer als SQLite
 - Voer volgende query uit:



Voorbeelden uit de praktijk: spreadsheet op maat (2)

SELECT symbol.cloi AS record, holding_pk, author_fn, author_vn, author_vw, title_lg, title_ti, impressum_pc, impressum_pl, impressum_uc, impressum_ug, impressum_ju1sv, impressum_ju1ty, impressum_ju2sv, impressum_ju2ty, collation_fm, collation_ka, collation_pg, number_nr, number_ty, fulltext loc, fulltext zurl

FROM symbol

```
LEFT JOIN holding USING(cloi)
```

```
LEFT JOIN author USING(cloi)
```

```
LEFT JOIN title USING(cloi)
```

```
LEFT JOIN impressum USING(cloi)
```

```
LEFT JOIN collation USING(cloi)
```

```
LEFT JOIN number USING(cloi)
```

```
LEFT JOIN fulltext USING(cloi)
```

```
WHERE lib = "MPM" AND (title ty = "h" OR title ty = "p")
```

```
ORDER BY record, holding pk, number ty;
```

OEFENING

 Geef me van alle werken die in Antwerpen zijn uitgegeven de c-loi, de hoofdtitel, de plaats van uitgave, de uitgever en het jaar van uitgave en sorteer alles van recent naar oud.



Meer informatie

- DB Browser for SQLite (download): <u>https://sqlitebrowser.org/dl/</u>
- Tabel met geijkte veldnamen : <u>https://anet.be/doc/anet/importexport/html/bvv-2228.html#tabel-van-geijkte-veldnamen</u> (geeft extra info over de headers van vaste velden in het catalografieformulier)
- <u>https://www.sqlitetutorial.net/</u> (section 1-5)
- https://www.w3schools.com/sql/
- YouTube
- ChatGPT, CoPilot, ...

