

DAT107 – Databaser, NoSQL

Øving 3: JSON

Oppgave 1 – JSON

Med utgangspunkt i XML-filen «fag.xml» fra øving 2, skal dere konvertere data i filen til JSON og lagre i filen «fag.json».

NB! Dere trenger kun å ta med de 5 første emnene.

Oppgave 2 - JSON

Med utgangspunkt i XML-filen «studenter.xml» fra øving 2, skal dere konvertere data i filen til JSON og lagre i filen «studenter.json».

NB! Dere trenger kun å ta med 3 emner per student, og kun 3 studenter.

Oppgave 3 - JSON i PostgreSQL

- a) Bruk filen «studenter-json.sql» til å opprette tabellen STUDENTER i «forelesning3» skjema i din PostgreSQL database, samt å legge til 3 rader.

NB! Kopier innholdet i filen inn i «Query Tool» i PgAdmin. Merk alt innholdet og kjør alt samtidig, dvs. trykk «execute»/«play» knappen (eller F5).

- b) Lag en spørring som henter ut alle studenter med student_id, fornavn, etternavn og studieprogram.
- c) Endre spørringen fra oppgaven over slik at fornavn og etternavn slås sammen (konkateneres med blankt tegn imellom) i en egen kolonne med alias «navn».
- d) Samme spørring som over, men legg til en kolonne «studiestart». Studiestart skal være en sammenslåing av semester og årstall, f.eks. 'H2024'.

Tips! Bruk parenteser rundt «jsonb_path_query()->>0» hvis du skal slå dette sammen med annen tekst ved hjelp av «||» operatoren.

- e) Hva blir resultatet av denne spørringen (se under)? (Kjør spørringen i pgAdmin)

```
SELECT jsonb_path_query('{"values":[1, 2, 3, 4, 5]}', '$.values[*] ? (@ > 3)');
```

- f) Hva er forskjellen på resultatet av spørringen over, og denne spørringen?

```
SELECT jsonb_path_query('{"values":[1, 2, 3, 4, 5]}', '$.values[*] ? (@ > 3)::text;
```

(Flere oppgaver på neste side)

- g) Lag en spørring som henter ut alle studenter som har tatt emnet «DAT152». Spørringen skal hente ut samme data som i oppgave d), dvs, student_id, fornavn, etternavn, studiestart og studieprogram.

Tips! Bruk "jsonb_path_exists" (I stedet for «jsonb_path_query») som vist i oppgaven over i WHERE filter.

Slik: ... WHERE jsonb_path_exists(student_data, '\$.<path> ? (@.<filter>)');

Husk at vi i filteret må bruke «==» for å sjekke likhet.

- h) Lag en spørring som bruker JSONB_SET til å oppdatere fornavnet til student_id = 101 til «Ola Johnny»

Tips! Se UPDATE fra video om JSON og/eller samme foil (pdf fil).

NB! DE NESTE OPPGAVENE ER EKSTRA VANSKELIG OG IKKE EKSAMENSRELEVANTE. Oppgavene er kun for de som har lyst til å bryne seg litt, og har lyst på en utfordring.

- i) Lag en UPDATE spørring som bruker JSONB_SET til å oppdatere poengene som student_id=101 har fått i emnet DAT100 til 99.

Tips! Du må finne posisjon til DAT100 i arrayen «emner» (starter på 0), og bruke posisjonen slike: jsonb_set(student_data, '{emner, <posisjon, poeng}', '99')

- j) Lag en spørring som henter ut alle fag som student_id=101 har tatt. I tillegg til skrive ut emne «kode» og «poeng» som egne kolonner (se under), skal også det aktuelle emnets posisjon i array «emner» være med som egen kolonne (se under).

Tips!

...

SELECT ..., emne

FROM studenter,

jsonb_array_elements(student_data->'emner') WITH ORDINALITY

AS emne(obj, pos)

WHERE ...

student_id [PK] integer	fornavn text	kode text	poeng text	pos bigint
101	Ola	DAT100	86	1
101	Ola	MAT101	92	2
101	Ola	ING100	77	3
101	Ola	DAT111	81	4