

Oppgaver: GitHub + SQL subqueries

Oppgave: GitHub

For å komme i gang med GitHub, se følgende film (16 min.) og gjør deretter oppgavene under:

https://www.youtube.com/watch?v=AW_U0q5BtFI

Merk: Pkt. 1-6 er forklart i filmen, over!

1. Opprett en GitHub konto
2. Lag et «personal access token».
3. Lag et Repository (dvs. et område på GitHub som kan holde på et kommende prosjekt).
4. Bruk Git console kommandoe «clone» for å klonе ditt Git repository («repo») til din laptop.
5. Opprett en ny SQL fil i din lokale Git klonе/mappe.
6. Legg filen til ditt GitHub repo, ved å bruke Git console kommandoene: status, add, stage, commit (*husk å ha med en beskrivende kommentar!*), push.

7. I VS Code, legg til [GitHub Pull Requests and Issues](https://code.visualstudio.com/docs/sourcecontrol/github) extension for å bruke GitHub direkte. Se denne artikkelen for veiledning: <https://code.visualstudio.com/docs/sourcecontrol/github>

8. Åpne sql-filen du lagde i 5, over, i VS Code. Lag et lite utvalg (ca. 5?) SQL statements. De trenger ikke være avanserte, men for hver statement som virker, commit den (gjennom VS Code, husk beskrivende kommentar!) Gjør en sync (pull, så push) etter 2. og etter 5. commit.

<i>NB: Bruk også denne filen (eller en tilsvarende sql-fil som du oppretter og legger til i ditt GitHub repo) når du skal løse SQL subquery oppgavene, under!</i>

9. Gå sammen med minst en annen i klassen. Del GitHub repoene deres med hverandre.
NB: GitHub innstillingene kan være litt knotete å finne frem i, ikke alt er like logisk plassert, helt ok å google hjelp på dette. :-P)
10. Se hvordan din partner har sjekket inn kode i sitt prosjekt. Se på history og commit comments. (For å finne et prosjekt du har fått tilgang på i GitHub må du muligens inn i settings menyen på GitHub websiden.)

Oppgave: Subqueries

Dere skal benytte databasen «world» for disse oppgavene.
--

NB: Bruk sql-filen du lagde i pkt. 5, over (eller en tilsvarende sql-fil som du oppretter og legger til i ditt GitHub repo) når du løser SQL subquery oppgavene under:

1) Når en spørring virker i pgAdmin, kopier den til fila, commit.

2) Når du er ferdig med 2-3 commits, sync (evt. pull så push).

Obs: er du mer glad i Git console enn Git gjennom VS Code? Da er det lov å gjøre commit, pull og push gjennom Git console for disse oppgavene. :-)

- a) Vi ønsker å se alle land i verden som ikke har noen byer. Sorter de alfabetisk på navn.
Tips: benytte en subquery for å spesifisere hvilke landkoder vi *ikke* vil ha med, kombinert med keyword IN eller NOT IN.
- b) Hent ut all info (alle kolonner) for byer som har likt navn som en eller flere andre byer.
Sorter på bynavn.
- c) **VANSKELIG!** Vi ønsker å se prosentandelen som innbyggerne i et land utgjør av jordas totale befolkning. Konkret vil vi ha landets navn, prosentandelen (med kolonnealias "WorldPopPercentage") og kvadratkilometer for dette landet. Sorter svaret på synkende prosentandel.
Problem: For å finne prosentandelen må vi også regne ut hva totalbefolkningen er som en del av spørringen (en subquery).
Tips: subqueries kan plasseres andre steder enn i where clausen. Hvor trenger vi en subquery i dette tilfellet?

Ekstra SQL oppgaver

Bruk «world» for disse SQL oppgavene også. Og legg de til i GitHub repoet ditt.

- a) **Google it:** Vi har IKKE hatt om CASE i timene, men klarer du å finne ut av hvordan du kan benytte begrepet allikevel? (Google er din venn.) Bruk en CASE utvalgsspørring til å hente ut: Landets navn og styresett, der styresett klassifiseres som: «Monarki», «Republikk» eller «Annet». Sorter resultatet på landets navn.
- b) **JOIN revisited:** Subquery oppgave a), over, kan også løses med JOIN – hvordan ser den da ut?