

# Checkpoint: Databasemodellering og SQL

Tid: 2 timer

Hjelpemidler: Google, Canvas og eget materiell, men *IKKE* lov å snakke sammen/hjelpe hverandre.

Gjør så mye du klarer av oppgave 1, deretter så mye du klarer av oppgave 2.

Zip til slutt materialet, lever via Canvas.

## Oppgave 1 (nivå 1 innhold)

Det finnes mange programmeringsspråk. Noen av disse er: (Name, Year, Creator)

- JavaScript, 1997, Brendan Eich
- C#, 2002, Anders Hejlsberg
- Java, 1996, James Gosling
- Python, 1994, Guido van Rossum
- COBOL, 1959, CODASYL

Når man registrerer et programmeringsspråk skal navn alltid være med, men de andre attributtene kan man la være å legge inn. For denne tabellen – Language – ønsker vi en Id satt av databasen, i form av et løpenummer.

Blant verktøy ("Tool") for å skrive programkode finner vi blant annet: (Id, Name, Developer)

- VS, Visual Studio Code, Microsoft
- IJ, IntelliJ, JetBrains

Når man registrerer et nytt programmeringsverktøy skal man alltid føre opp Id og navn. Men hvilken utvikler som står bak verktøyet er valgfritt å ta med. (Og i denne tabellen setter vi altså Id selv.)

Vi ønsker en 1:M (en-til-mange) kopling mellom Tool og Language. Når du legger inn data ønsker vi disse koplet slik at: Visual Studio Code kan benyttes til å programmere C# og JavaScript. IntelliJ kan benyttes til å programmere Java. (Ingen verktøy kan benyttes til Python eller COBOL i denne oppgaven.)

- a) Lag **ER modell** for løsningen beskrevet over. **Modellen skal leveres som en bildefil** (.jpg eller .png). Dette kan være bilde tatt med ditt laptopkamera av en ER modell du tegner for hånd, eller det kan være screenshot av modell tegnet i et verktøy. (Tegne for hånd og ta bilde er antakeligvis den raskeste løsningen. Blir det krøll når du skal ta bilde med laptop-kameraet: Lever inn håndtegningen din.) ER modellen skal minimum inneholde:
  - entiteter (tabeller)
  - attributter (kolonner)
  - forhold mellom tabeller
  - primærnøkler (PKs)
  - fremmednøkler (FKs)
- b) Opprett en **database** med navn "**CheckpointDb**". Opprett tabeller og data ved å bruke SQL-statements, iht. ER modellen du har laget over. Gjør egne valg for detaljer som ikke er beskrevet i oppgaveteksten (bl.a. datatyper for alle kolonner). Alle nødvendige SQL statements for å lage tabeller og fylle de med data skal leveres i en egen .sql fil: **buildDatabase.sql**.

*NB: Klarer du ikke å få til modellen du har tenkt deg, eller klarer du ikke legge inn data iht. oppgaveteksten, gjør egne tilpasninger som gjør at du kommer videre. MEN: Kommenter i så fall dette i sql-filene! (Som '--' kommentarer.)*

*Tips: Så lenge det er ';' (semikolon) bak hver kommando («statement»), kan flere SQL kommandoer kjøres samlet. Etter hvert som du lager og tester dine kommandoer, kopier og lim de over i en ".sql" tekstfil (bare å skifte navn på en .txt fil til en .sql fil). Kommandoene skal stå oppført i den rekkefølgen det er ønskelig å kjøre dem. MERK: 1 fil for CREATE og INSERT, en annen fil for SELECT.*

- c) Opprett **spørringer** («queries») som henter ut data fra din databaseløsning. Alle nødvendige SQL spørringer skal leveres i en egen .sql fil: **queryDatabase.sql**

Lag spørringer som:

1. Viser Year, Name og Creator (kolonnene listet i denne rekkefølgen) for alle rader i Language tabellen, sortert stigende på årstall.
2. Viser alle kolonner i Tool tabellen, men viser kun rad(er) med Visual Studio Code som Name.

## Oppgave 2 (nivå 2 innhold)

- a) Vi fortsetter med **flere spørringer** mot databasen – de må gjerne skrives inn i samme **queryDatabase.sql** fil som ble benyttet i oppgave 1. (Fortsetter derfor på nummereringen.)
3. Viser Tool-Id og summen av språk med alias «NumberOfLanguages». Skal bare vise rader hvor Tool\_Id ikke er NULL.
  4. Føyer sammen (JOIN) to tabeller og lar oss se alle språk sine navn (skal vises som LanguageName i utskriften) og samtidig hvilket verktøy de kan programmeres gjennom, om noe (skal vises som ToolName i utskriften). *Merk:* Om det skulle være språk som ikke har noe verktøy, ønsker vi fortsatt å se dem, men da skal det vises NULL (eller '-Not supported-' om du vet hvordan du implementerer det) for verktøykolonnen.
- b) Vi oppdager at det er mulig å installere utvidelser til våre programmeringsverktøy! IntelliJ kan da benyttes til Python og JavaScript (i tillegg til Java). Visual Studio Code kan benyttes til Python (i tillegg til C# og JavaScript). Det betyr at vi nå har et **M:M forhold** mellom Tool og Language. **Tegn (og lever inn som eget bilde) den nye ER modellen.**
- c) **Implementer den nye strukturen** i databasen. Opprett ekstra tabeller etter behov. For ikke å ødelegge for eksisterende tabeller, spørringer og script, lag en ekstra versjon av eksisterende tabell med '2' bakerst (f.eks. "**Language2**") og **tilsvarende for andre tabeller** der eksisterende tabell trenger å justeres. Legg inn data i de nye tabellene, slik at data og struktur stemmer med informasjonen i punkt b, over (legg gjerne data inn ved hjelp av spørring mot den opprinnelige tabellen der det er hensiktsmessig). Legg til nye SQL statements for dette i **buildDatabase.sql**
- d) Lag **nye spørringer for 3 og 4, over**, slik at de tar høyde for den nye strukturen. Legg til nye SQL queries for dette i **queryDatabase.sql**