

DDL og CRUD

Modul 9 (uke 7) innhold

Dagens agenda

- 08:30** | **Check-in** (oppmøte + dagens agenda)
+ Klasserommet: **DDL & CRUD**
- 09:30** | Løse DB-oppgavesett (ligger på Canvas)
- 11:30** | *45 min. lunch*
- 12:15** | Klasserommet: **ER modellering, del 1/2**
- 13:00** | Jobbe med DB-oppgaver
- 16:00** | Klasserommet: **ER modellering, del 2/2**
+ Check-out (ferdig **16:30**)

Finne nøkler – utestående oppgave fra i går

City

ID	Name	CountryCode	District	Population
1	Kabul	AFG	Kabul	1780000
2	Qandahar	AFG	Qandahar	237500
3	Herat	AFG	Herat	186800
4	Mazar-e...	AFG	Balkh	127800
5	Amster...	NLD	Noord-H...	731200
6	Rotterd...	NLD	Zuid-Holl...	593321
7	Haag	NLD	Zuid-Holl...	440900
8	Utrecht	NLD	Utrecht	234323

Countrylanguage

CountryCode	Language	IsOfficial	Percentage
ABW	Dutch	T	5.3
ABW	English	F	9.5
ABW	Papiament	F	76.7
ABW	Spanish	F	7.4
AFG	Balochi	F	0.9
AFG	Dari	T	32.1
AFG	Pashto	T	52.4
AFG	Turkmenian	F	1.9

Country

Code	Name	Continent	Region	SurfaceArea	IndepYear	Population	LifeExpectancy	GNP	GNPold	LocalName	GovernmentForm	HeadOfState	Capital	Code2
ABW	Aruba	North America	Caribbean	193.00	NULL	103000	78.4	828.00	793.00	Aruba	Nonmetropolitan T...	Beatrix	129	AW
AFG	Afghani...	Asia	Souther...	652090.00	1919	22720000	45.9	5976.00	NULL	Afganistan/A...	Islamic Emirate	Mohammad Omar	1	AF
AGO	Angola	Africa	Central ...	1246700.00	1975	12878000	38.3	6648.00	7984.00	Angola	Republic	JosÃ© Eduard...	56	AO
AIA	Anguilla	North America	Caribbean	96.00	NULL	8000	76.1	63.20	NULL	Anguilla	Dependent Territor...	Elisabeth II	62	AI
ALB	Albania	Europe	Souther...	28748.00	1912	3401200	71.6	3205.00	2500.00	ShqipÃ«ria	Republic	Rexhep Mejdani	34	AL
AND	Andorra	Europe	Souther...	468.00	1278	78000	83.5	1630.00	NULL	Andorra	Parliamentary Copri...		55	AD
ANT	Netherl...	North America	Caribbean	800.00	NULL	217000	74.7	1941.00	NULL	Nederlandse...	Nonmetropolitan T...	Beatrix	33	AN

1. Hvilke kolonner kan være primærnøkler?
2. Hvilke kolonner kan være fremmednøkler?

SQL DDL: endring av tabeller (ikke data)

- DDL = Data Definition Language.
- En del av SQL, har følgende SQL kommandoer for å modifisere tabeller:
 - CREATE TABLE
 - ALTER TABLE
 - DROP TABLE

CRUD (i SQL sammenheng)

- **CRUD** = de vanlige operasjonene vi gjør for innhold i en database:
 - **C**reate (**INSERT INTO**)
 - **R**ead (**SELECT**)
 - **U**pdate (**UPDATE**)
 - **D**elete (**DELETE FROM**)
- Merk: Alle 4 operasjoner går på **innhold**, ikke tabellene i seg selv.
 - (Altså **IKKE** ~~create table~~, ~~alter table~~ eller ~~drop table~~.)

SQL – opprette tabell

- Skal vi lage en ny tabell, gjøres det med følgende syntaks:
 - (et utvalg av mulighetene)

```
CREATE TABLE tabellnavn
(
  kolonnenavn datatype [UNIQUE|NOT NULL|DEFAULT|...],
  kolonnenavn datatype [UNIQUE|NOT NULL|DEFAULT|...],
  ...,
  [PRIMARY KEY (kolonnenavn)],
  [FOREIGN KEY (kol.navn) REFERENCES tabell_navn (kol.navn)]
);
```

SQL – opprette tabell (forts.)

- Eksempel:


```
CREATE TABLE countryIsh (  
  Code CHARACTER(3) NOT NULL,  
  Name TEXT NOT NULL UNIQUE,  
  Continent TEXT CHECK ((Continent = 'Asia'::TEXT) OR (Continent = 'Europe'::TEXT)),  
  Region TEXT NOT NULL DEFAULT '',  
  SurfaceArea REAL NOT NULL DEFAULT 0.00,  
  IndepYear SMALLINT DEFAULT NULL,  
  Population INTEGER NOT NULL DEFAULT 0,  
  PRIMARY KEY (Code)  
);
```

- NOT NULL betyr at kolonnen må fylles inn.
- DEFAULT brukes for å angi standardverdier.
- UNIQUE hindrer to like verdier (får feilmelding ved innleggelse av duplikat).
- CHECK legger begrensninger på innhold, resultatet blir som en Enum i programmeringsspråk.
- Kjør gjerne denne SQL-en i test-databasen!

Løpenummer

- Løpende rekke av tall, benyttes gjerne som primærnøkkel.
- Sikrer at alle verdier i en kolonne er forskjellige.
 - Ergo også unik primærnøkkel om den benyttes som dette.
- Vi kan angi at en kolonne skal benytte et løpenummer v.h.a. datatypen **SERIAL**.
 - Øker med +1 for hver nye rad. (Gjenbruker ikke tidligere verdier selv om de frigjøres gjennom sletting.)

```
CREATE TABLE student (  
  ID SERIAL NOT NULL,  
  Name TEXT NOT NULL,  
  PRIMARY KEY (ID)  
);
```



- (Samme funksjonalitet benytter nøkkelordene AUTO_INCREMENT i MySQL og IDENTITY(1,1) i SQL Server).
- SQL-syntaksen over forklares på kommende slides.

SQL – opprette tabell (forts.)

- Eksempel #2: en tabell med fremmednøkkel til countryish tabellen:

```
CREATE TABLE cityIsh (  
  ID SERIAL,  
  Name TEXT NOT NULL,  
  CountryishCode CHARACTER(3) NOT NULL,  
  District TEXT NOT NULL,  
  Population INTEGER NOT NULL,  
  PRIMARY KEY (ID),  
  FOREIGN KEY (CountryishCode) REFERENCES CountryIsh(Code)  
);
```

- SERIAL betyr, som alt nevnt, datatype med autogenerated løpenummer.
- FOREIGN KEY (nederst) spesifiserer fremmednøkkel.

SQL – endre tabell

- Eksisterende tabeller kan endres med følgende syntaks:

```
ALTER TABLE tabellnavn  
ADD kolonnenavn datatype,  
ALTER COLUMN kolonnenavn TYPE datatype_ny,  
DROP COLUMN kolonnenavn,  
...
```

Velg en eller fler av disse (avhengig av hva du ønsker å gjøre).

- Eksempel:

```
ALTER TABLE student  
ADD Birthdate TIMESTAMP;
```

SQL – slette tabell

- Å slette en tabell er kort og greit:

```
DROP TABLE tabellnavn;
```

- NB: Men vær sikker på at du sletter rett tabell!
Den blir borte for evig og alltid. O_o

- Eksempel:

```
DROP TABLE student;
```

SQL – legge til innhold i tabell

- Å legge til rader i en tabell kan gjøres på to måter.
 - Med eller uten spesifisering av kolonnene:

```
INSERT INTO tabellnavn  
VALUES (value1, value2, value3, ..., valueN);
```

eller

```
INSERT INTO tabellnavn (kolonnenavn1, kolonnenavn2, ...)   
VALUES (value1, value2, ...),  
      (value1, value2, ...),  
      ...
```

INSERT INTO – eksempel

```
INSERT INTO countryIsh  
VALUES ('WWW', 'Wlandet', 'Asia', 'OnlineRegion', 1337, 2023, 42);
```

```
INSERT INTO cityIsh (name, countryishCode, district, population)  
VALUES ('Byen min', 'WWW', 'OnlineArea', 12345);
```

- Legg merke til at vi ikke legger inn informasjon i alle kolonner, selv om de ikke tillater null.
 - Vi kan gjøre det slik fordi de får satt en DEFAULT-verdi.
- Er det tilfeldig at det legges inn informasjon i tabellen countryIsh først?

Innsettingsregler

- Dersom vi setter inn data i alle kolonner, trenger vi ikke ramse opp kolonnene.
- Skal vi sette inn data i bare noen kolonner, må vi navngi kolonnene som skal ha data.
 - For kolonnenavn som sløyfes fra listen blir det satt inn `null` (eller default verdier).
- Tekstfelt (som TEXT) skal stå i fnutter!
- Husk: innhold i primærnøkkelceller må være unikt!

SQL – oppdatere innhold i tabell

- Oppdatering av innhold i en tabell kan gjøres for alle rader, eller med en where-clause for et utvalg rader:

```
UPDATE tabellnavn  
SET kolonnenavn1 = value1, kolonnenavn2 = value2, ...  
[WHERE ...]
```

- Eksempel:

```
UPDATE countryIsh  
SET Region = 'WRegion', Population = 74  
WHERE Code = 'WWW';
```

SQL – fjerne innhold i tabell

- Fjerning av innhold i en tabell kan gjøres for alle rader, eller med en where-clause, for enkelte rader: (som for update)

```
DELETE FROM tabellnavn  
[WHERE ...]
```

- Eksempel:

```
DELETE FROM countryIsh  
WHERE Code = 'WWW';
```

– NB: Får feilmelding, forklaring på neste slide.

What? DELETE fungerte ikke!?!

```
ERROR:  update or delete on table "countryish" violates foreign key
constraint "cityish_countryishcode_fkey" on table "cityish"
DETAIL:  Key (code)=(WWW) is still referenced from table "cityish".
SQL state: 23503
```

- Vi får ikke til å slette fra countryIsh.
 - Vi får ikke lov fordi byen vi la inn har en fremmednøkkel til dette landet.
- Vi kan fikse dette ved å slette byen først. Men la oss forsøke noe annet:
 - Vi vil legge inn en DELETE CASCADE.
 - Da må vi fjerne FK, og legge den inn på nytt (neste side).

CASCADE DELETE, eksempel

- Først fjerne eksisterende FK: (finder denne i pgAdmin -> cityish tabellen -> Constraints)

```
ALTER TABLE cityIsh  
DROP CONSTRAINT cityish_countryishcode_fkey;
```

- Deretter opprette den på nytt med DELETE CASCADE:

```
ALTER TABLE cityIsh  
ADD CONSTRAINT cityish_countryishcode_fkey  
FOREIGN KEY (countryishCode) REFERENCES CountryIsh(Code) ON DELETE CASCADE;
```

- Forsøker så å slette på nytt: (nå går det bra, og den sletter både fra countryIsh og cityIsh)

```
DELETE FROM countryIsh  
WHERE Code = 'WWW';
```

SQL – fjerne hele tabellen

- Det er altså en viktig forskjell på å slette innhold i en tabell (rader i en tabell) og å fjerne selve tabellen (inklusive alle rader).
- Hvis vi vil rydde opp etter oss i dette eksempelet, og fjerne testtabellene våre:

```
DROP TABLE cityIsh;
```

```
DROP TABLE countryIsh;
```

Reinstallere world

- Det kan jo være at man sletter (eller endrer) innhold i world databasen som ikke skulle vært slettet (eller endret).
- Fortvil ikke! For å reinstallere world:
 - Gjennom pgAdmin, slett alle tabeller i World databasen.
 - Legg tabellene inn på nytt gjennom World sql-scriptet som ligger på Canvas.
- VOILA! (The) world is restored! ;-)



