

## **Table of Contents**

Lage Schema/database.....	2
Legge tabell i databasen.....	3
Sette data i tabellen world.....	5

# Lage Schema/database

sql kode:

```
CREATE SCHEMA `oblig2v23` DEFAULT CHARACTER SET utf8 COLLATE utf8_bin;
```

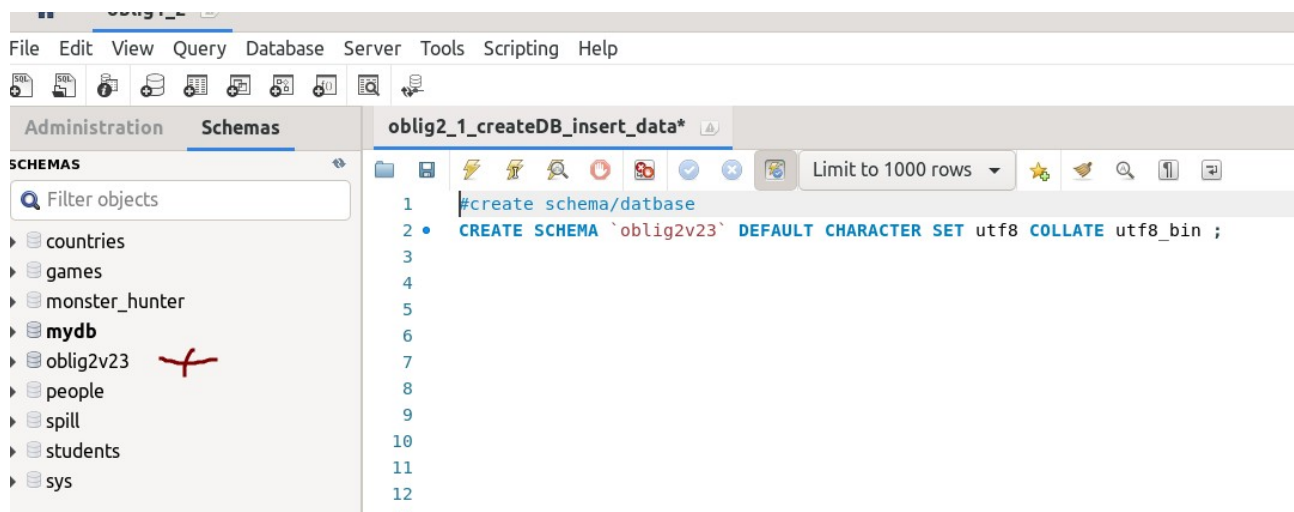
For å lage ny Schema, kan vi bruke enten workbench sin gui eller med sql kode. Her har vi brukt denne koden linjen.

CREATE SCHEMA er nøkkelordene for å lage schema. Etter det gir vi navnet Schema ”oblig2v23”.

DEFAULT CHARACTER SET er nøkkel ord for bestemme hva karakterer som Schema kan innholde. Utf8 er alle mulige bokstaver og tegn brukt over hele verden. Dette går mer på språk. Norsk spesielle bokstaver ”Æ,Å,Ø” er inkludert i utf8.

Utf8\_bin er type utf, men for maskin. Bin er forkortet for binary. Maskin vil kunne skille mellom store og små bokstaver og lignende.

Når vi kjører kode, får vi ny database vist på schemas tab i venstre



# Legge tabell i databasen

```
#use the database
```

```
USE oblig2v23;
```

```
#create a table
```

```
create TABLE world (  
id INT NOT NULL AUTO_INCREMENT,  
Country VARCHAR(25),  
Region VARCHAR(25),  
Population INT,  
Area INT,  
GDP INT,  
PRIMARY KEY(id)  
);
```

For å legge tabell i database, må vi først velge hvilke database vi skal jobbe på. Det er to måter å gjøre dette på. Man kan dobbelklikke på databasen venstre siden i workshop under ”schemas” tab eller kjøre sql kode USE databaseNavn; Her bruker vi USE oblig2v23: for å komme oss i riktig database.

Etter vi er inn i riktig database, er det mulig legge tabeller i databasen. Vi velger å bruke sql kode lage kolonnene til base. Vi starter alltid med CREATE TABLE, som sier begynn å lage tabell. Også må vi gi tabellen navn som ikke unik i databasen. Kan ikke ha to tabeller med samme navn i samme database. Vi kaller tabell for ”world”. Den skal inneholde alle land på jorda.

I parentes kan vi definere navn på kolonnene vi vil at tabellen skal ha og hva slags datatype hver kolonne har lov til å ta imot. Man legger kolonne ved å definere navn på kolonnen først, etter det datatype kolonnen skal kunne ta imot. Etter er mye man kan velge å legge til kolonnen. Disse mer regler og begrensninger for kolonnen. Dette er altså mal for hva tabellen skal innehold. Derfor vi kaller dette Schema, fordi Schema database objekter som tabeller, indekser og mer. Vi skal gå en for en hva kolonne er tabellen ”world”.

## Første kolonnen: id

```
id INT NOT NULL AUTO_INCREMENT,
```

navnet på første kolonnen er ”id”. Det er datatype int som er heltall. Med NOT NULL, kan tabellen ikke være tom. AUTO\_INCREMENT gjør at tabellen automagisk oppdaterer seg med 1 hver gang ny rad blir lagd til tabellen. Hver rad i tabellen vil på denne måte ha unik id tall.

## Andre og tredje kolonner: Country, Region

Country **VARCHAR(25)**,

Region **VARCHAR(25)**

Andre og tredje kolonner er bygd på samme måte. Andre kolonne har navnet country og har datatype VARCHAR(25). Denne kolonnen skal innehold navnet på land. VARCHAR er variable character. Typisk navn programmering språk for denne datatype er ''string''. Det er tekst type. Tredje er region til landet. Denne er også varchar datatype. Varchar maks lengde på 255 karakterer, men det er mulig å presisere i parentes antall karakterer i heltall. Her har vi valgt 25 karakterer på begge.

## Fjerde, femte og sjette kolonner: Population, Area, GDP

Population **INT**,

Area **INT**,

GDP **INT**,

Disse tre kolonnen er av datatype int, heltall. Population kolonnen innehold befolkning til, Area hvor stor landet er og GDP er GDP på norsk også.

## Primary key

**PRIMARY KEY**(id)

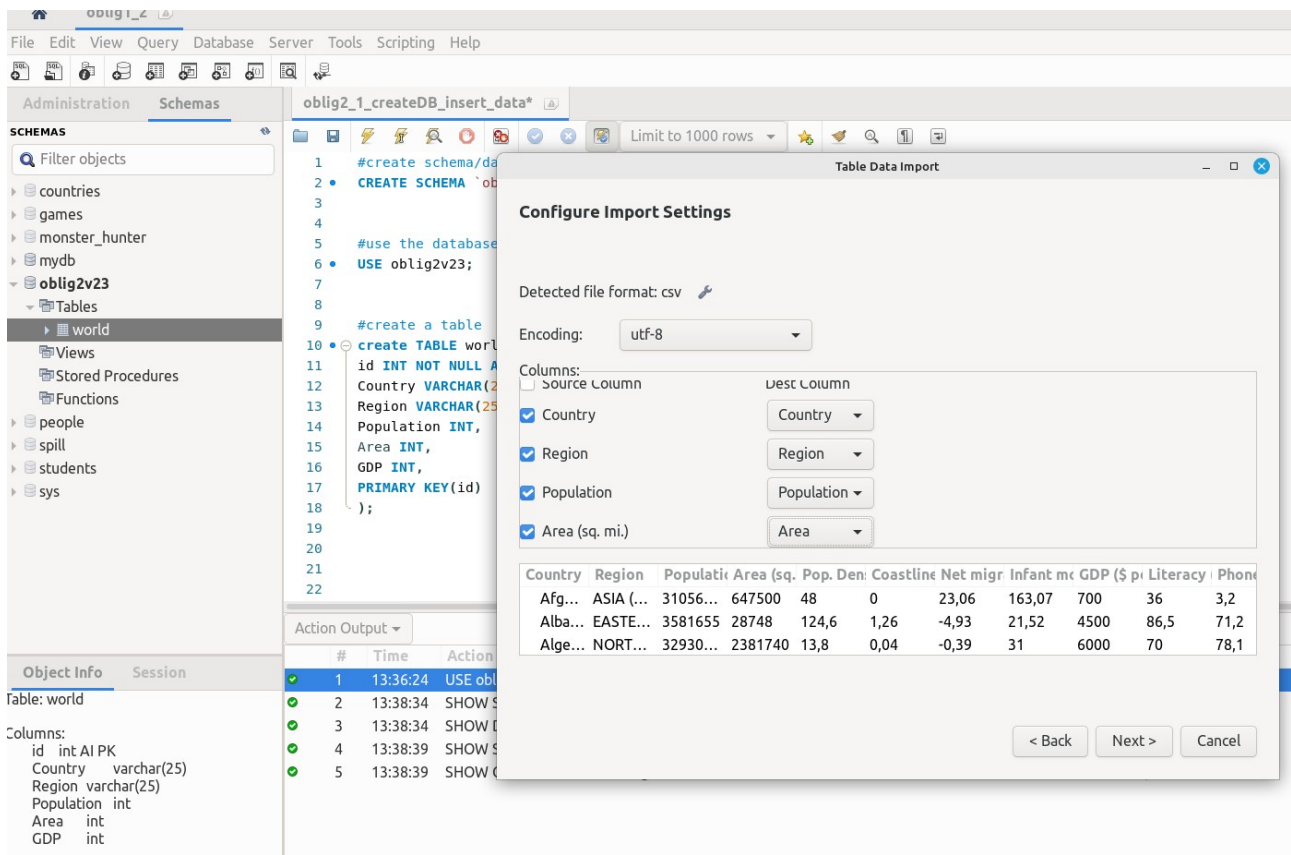
Primary key er det unike nøkkel som andre tabeller kan brukes til å koble med denne tabellen. I parentes kan vi definere hva som blir unik nøkkel for hver rad i tabellen. Her har vi valgt id som den unike nøkkelen. Den øker automatisk og på denne vil måte vil hver rad i tabellen alltid ha unik id. Navn på land er ikke spesiell nok, siden noen land kan samme navn.

Sånn ser det ut når vi legger tabell i database. World er synlig under Tables

The screenshot shows a database management interface with two main panes. The left pane, titled 'Administration' and 'Schemas', displays a tree view of database objects. Under the 'oblig2v23' schema, the 'Tables' folder is expanded, showing a table named 'world' with a red plus sign next to it. The right pane, titled 'oblig2\_1\_createDB\_insert\_data\*', displays the SQL script used to create the table. The script is as follows:

```
1 #create schema/database
2 • CREATE SCHEMA `oblig2v23` DEFAULT CHARACTER SET utf8 COLLATE utf8_bin ;
3
4
5 #use the database
6 • USE oblig2v23;
7
8
9 #create a table
10 • create TABLE world (
11   id INT NOT NULL AUTO_INCREMENT,
12   Country VARCHAR(25),
13   Region VARCHAR(25),
14   Population INT,
15   Area INT,
16   GDP INT,
17   PRIMARY KEY(id)
18 );
19
20
21
22
```

# Sette data i tabellen world



Vi setter data i tabellen via workbench. Vi høyreklikker på tabellen og velger ''Table Data Import Wizard''. Vi følger stegene og velger ut kun kolonnene er definert i tabellen vår. Csv filen vi importerer data fra mer kolonner enn vi ønsker å ha med.

## Første 10 rad i tabellen

```
SELECT * FROM world;
```

Vi henter ut alt i tabellen world med denne SELECT spørringen. Workbench gir oss muligheten å begrense antall rad som spørring kan hente.

