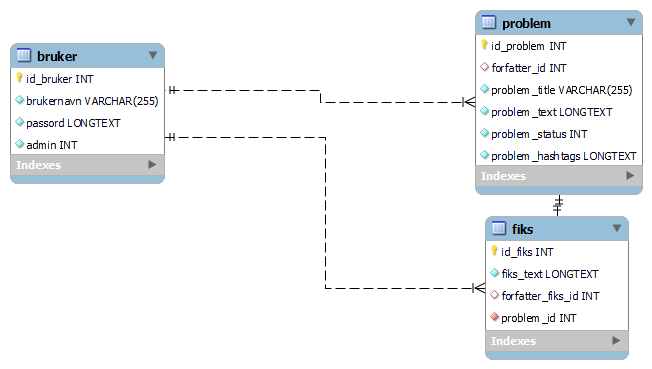
# Database modell:



## Bruker - tabell

En bruker har *id\_bruker*, *brukernavn* og *passord* + *admin*:

*Id\_bruker* er id til bruker

*Brukernavn* er et brukernavn

*Passord* er passordet du skriver inn men hashet eller saltet

*Admin* kan være 0 (kunde), 1 (ansatt), 2 (admin). Hver rolle kommer til å få spesifikke rettigheter: Kunder kan lage en problem, Ansatt kan svare på de problemene og Admin kan slette bruker og problemer.

## Problem - tabell

Et problem har *Id\_problem*, *forfatter\_id*, *problem\_title, problem\_text, problem\_status* og *problem\_hashtags*.

*Id\_problem* er problemet til id (hihi)

*forfatter\_id* er foreign key med iden til en id i bruker-tabellen.

Hvis en bruker slettes blir *bruker\_id* bare null.

Slik kan vi beholde løsningen til problemet og selve problemet, til og med når brukeren er slettet

*Problem\_title* og *problem\_text* er selvforklarende

*Status* kan være: 0 (Sendt), 1 (Løst).

Hvis den er løst vil den lage en fiks i fiks tabellen.

*Problem\_hashtags* er en liste me hastags (f.eks. #Mus).

Dette er sånn at man kan søke på en løsning som har flere hashtags.

Søket på hashtags er styrret av hasttagsene som er i databasen.

Hvis du lager f.eks. #Kuben, blir det mulig å søke #Kuben.

## Fiks – tabell

En fiks har *id­\_fiks*, *fiks\_text*, *forfatter\_fisk\_id* og *problem\_id*

*Id\_fiks* og *fiks\_text* er selvforforklarende.

*Forfatter\_fisk­\_id* er en foreign nøkkel til en bruker som burde alltid være en admin eller ansatt

Hvis brukeren blir slettet blir *Forfatter­\_fisk\_id* bare **null**. Slik at vi kan beholde fikset.

*Problem\_id* er foreign key id-en til problemet.

Hvis originale problemet blir slettet burde denne fikset bli slettet samtidig (slettet gjennom admin).