DAT 1000 Eksamen

Oppgave 1 SQL

1-a:

CREATE TABLE Medlem (

Mnr INT NOT NULL,

Fornavn VARCHAR(255) NOT NULL ,

Etternavn VARCHAR(255) NOT NULL ,

Tlf VARCHAR(255) NOT NULL,

PRIMARY KEY (Mnr)

);

CREATE TABLE Påmelding (

TNr INT NOT NULL,

MNr INT NOT NULL,

PRIMARY KEY (TNr, MNr),

FOREIGN KEY (Tnr) REFERENCES Tur(TNr),

FOREIGN KEY (MNr) REFERENCES Medlem(Mnr)

);

1-b:

SELECT \*

FROM tur

WHERE MONTH(StartDato) = 7

AND Pris < 8000

ORDER BY Pris ASC, StartDato;

1-c:

SELECT medlem.Mnr, Fornavn, Etternavn

FROM medlem

RIGHT JOIN påmelding p on medlem.Mnr = p.MNr

RIGHT JOIN tur t on p.TNr = t.TNr

WHERE t.StartHytte = 2;

1-d:

SELECT tur.TNr, Beskrivelse, StartDato, COUNT(p.Mnr) AS "Antall Påmeldte"

FROM tur

RIGHT JOIN påmelding p on tur.TNr = p.TNr

GROUP BY p.TNr

ORDER BY COUNT(p.MNr);

1-e:

CREATE TABLE Hytte (

HNr INT NOT NULL PRIMARY KEY,

Navn VARCHAR(255) NOT NULL ,

AntSenger INT NOT NULL ,

HytteType VARCHAR(255) NOT NULL,

CHECK (HytteType in ('Betjent', 'Selvbetjent', 'Ubetjent'))

);

Teknikken jeg bruker er å bruke funksjonen CHECK for å se hvilke verdier som kommer inn. Og vil da sende en error hvis noen av disse verdiene er feil.

1-f:

INSERT INTO medlem (Fornavn, Etternavn, Tlf) VALUES ('Per', 'Persen', '91314322');

INSERT INTO påmelding (TNr, MNr) VALUES (3, 5);

1-g:

SELECT Fornavn, Etternavn

FROM medlem

LEFT JOIN påmelding p on medlem.Mnr = p.MNr

WHERE p.Mnr IS NULL;

2-a: Se vedlegg 1

2-b: Se vedlegg 2

Oppgave 3 -Normalisering

1. Første normalform, må inneholde kun en verd og det kan ikke finnes repeterende grupper.

Andre normalform, må alle verdier være avhengig av primærnøkkelen.

Tredje normalform, må ikke en verdi være avhengig av en annen verdi som ikke er en primærnøkkel.

1. Denne tabellen er i universal relasjon fra starten.

Dette er tabellen på første normalform. Her har vi da satt det til kun en verdi per felt.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PET ID | PET NAME | PET TYPE | PET AGE | OWNER | VISIT DATE | PROCEDURE |
| 246 | Rover | DOG | 12 | SAM COOK | JAN 13/2002 | 01 - RABIES VACCINATION |
| 246 | Rover | DOG | 12 | SAM COOK | MAR 27/2002 | 10 - EXAMINE and TREAT WOUND |
| 246 | Rover | DOG | 12 | SAM COOK | APR 02/2002 | 05 - HEART WORM TEST |
| 298 | SPOT | DOG | 2 | TERRY KIM | JAN 21/2002 | 08 - TETANUS VACCINATION |
| 298 | SPOT | DOG | 2 | TERRY KIM | MAR 10/2002 | 05 - HEART WORM TEST |
| 341 | MORRIS | CAT | 4 | SAM COOK | JAN 23/2001 | 01 - RABIES VACCINATION |
| 341 | MORRIS | CAT | 4 | SAM COOK | JAN 13/2002 | 01 - RABIES VACCINATION |
| 519 | TWEEDY | BIRD | 2 | TERRY KIM | APR 30/2002 | 20 - ANNUAL CHECK UP |
| 519 | TWEEDY | BIRD | 2 | TERRY KIM | APR 30/2002 | 12 - EYE WASH |

På andre normalform så kan den ikke ha noen partielle avhengigheter. Dette betyr at den ikke kan ha en verdi som er avhenging av kun deler av primørnøkkelen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PET ID | PET NAME | PET TYPE | PET AGE | OWNER |
| 246 | Rover | DOG | 12 | SAM COOK |
| 246 | Rover | DOG | 12 | SAM COOK |
| 246 | Rover | DOG | 12 | SAM COOK |
| 298 | SPOT | DOG | 2 | TERRY KIM |
| 298 | SPOT | DOG | 2 | TERRY KIM |
| 341 | MORRIS | CAT | 4 | SAM COOK |
| 341 | MORRIS | CAT | 4 | SAM COOK |
| 519 | TWEEDY | BIRD | 2 | TERRY KIM |
| 519 | TWEEDY | BIRD | 2 | TERRY KIM |

|  |  |  |  |
| --- | --- | --- | --- |
| PET ID |  | VISIT DATE | PROCEDURE |
| 246 |  | JAN 13/2002 | 01 - RABIES VACCINATION |
| 246 |  | MAR 27/2002 | 10 - EXAMINE and TREAT WOUND |
| 246 |  | APR 02/2002 | 05 - HEART WORM TEST |
| 298 |  | JAN 21/2002 | 08 - TETANUS VACCINATION |
| 298 |  | MAR 10/2002 | 05 - HEART WORM TEST |
| 341 |  | JAN 23/2001 | 01 - RABIES VACCINATION |
| 341 |  | JAN 13/2002 | 01 - RABIES VACCINATION |
| 519 |  | APR 30/2002 | 20 - ANNUAL CHECK UP |
| 519 |  | APR 30/2002 | 12 - EYE WASH |

Dette blir tabellen for 3. Normalform da den ikke har noen verdier som ikke er avhengig av en annen ikke primærnøkkel verdi.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PET ID | PET NAME | PET TYPE | PET AGE | OWNER |
| 246 | Rover | DOG | 12 | SAM COOK |
| 298 | SPOT | DOG | 2 | TERRY KIM |
| 341 | MORRIS | CAT | 4 | SAM COOK |
| 519 | TWEEDY | BIRD | 2 | TERRY KIM |

|  |  |
| --- | --- |
| ID | PROCEDURE |
| 01 | RABIES VACCINATION |
| 05 | HEART WORM TEST |
| 08 | TETANUS VACCINATION |
| 10 | EXAMINE and TREAT WOUND |
| 12 | EYE WASH |
| 20 | ANNUAL CHECK UP |

|  |  |  |  |
| --- | --- | --- | --- |
| VisitID | PET ID | VISIT DATE | PROCEDURE ID |
| 1 | 246 | JAN 13/2002 | 01 |
| 2 | 246 | MAR 27/2002 | 10 |
| 3 | 246 | APR 02/2002 | 05 |
| 4 | 298 | JAN 21/2002 | 08 |
| 5 | 298 | MAR 10/2002 | 05 |
| 6 | 341 | JAN 23/2001 | 01 |
| 7 | 341 | JAN 13/2002 | 01 |
| 8 | 519 | APR 30/2002 | 20 |
| 9 | 519 | APR 30/2002 | 12 |

Oppgave 4

1. En primær nøkkel er en unik indentifikator for en tabell som bestemmer de andre verdiene. Vi bruker primærnøkler for å løse en rad i en tabell til noen bestemte verdier.

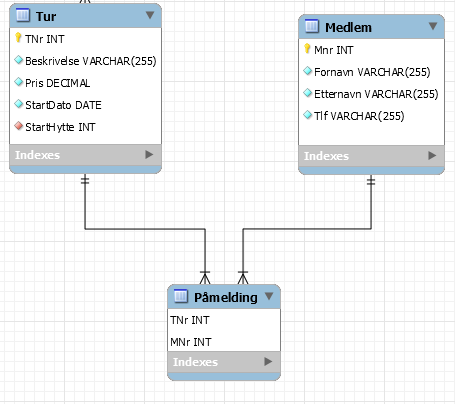
For eksempel i en PostNr og Posted tabell er PostNr primærnøkkel da den bestemmer poststedet.

1. Et utsnitt (View) er en måte å lage en virtuell tabell basert på en SELECT setning.

Utsnitt kan brukes til flere ting da de er raskere enn komplekse SELECT setninger, dette betyr også at man lager en virtuell tabell man kan utføre videre spørringer på og den inneholder kun eksisterende verdier som er i tabellen fra før.

Et eksempel er at man har en gammel tabell som man skal migrere til en ny en. Dette betyr at den gamle tabellen ikke skal eksistere lengre noe som kan skape problemer for applikasjoner der man bruker den gamle tabellen. Derfor kan man bruke den nye tabellen som har flere felter, men lage et view på denne tabellen med navn som den gamle tabellen og ekskluderer de verdiene man ikke skal ha med seg. Derfor reduserer man redudant data, og minimaliserer risikoen ved å lage en ny tabell.

1. En m-t-m relasjon er en hypotetisk relasjon da den egentlig ikke har mulighet for å eksistere uten å bruke en ekstra tabell for å koble tabell 1 og tabell 2 til hverandre. Dette gjør at en mange til mange tabell er mulig ved å bruke en koblingstabell.



Her set vi at vi ønsker at en tur kan ha flere medlemmer og et medlem skal ha flere turer. Derfor legger vi begge to til i en tabell påmelding som knyter dem sammen via primærnøkkel.