SQL—Structured Query Language

- Definere tabeller
- Skranker
- Fylle tabeller med data

Lage en tabell med SQL

create table R
(A_1 D_1 [S_1],
. . .
A_n D_n [S_n],
[*liste av skranker*]
);

R er navnet på relasjonen/tabellen
A_i er et atributt
D_j er et domene
S_k er en skranke

[ ] betyr at dette leddet er en valgfri del av setningen

Lage en tabell med SQL—forenklet uten skranker

create table R (
A_1 D_1,
A_2 D_2,
...
A_n D_n
);

R er navnet på relasjonen/tabellen
A_i er et atributt
D_j er et domene

create—eksempel

```
create table EmneStudent (
        brukernavn varchar(8),
        emnekode varchar(7),
        gruppenr integer,
        eksamensdato date
);
```

create—eksempel

```
create table EmneStudent (
        brukernavn varchar(8),
        emnekode varchar(7),
        gruppenr int, -- kan også skrive integer
        eksamensdato date
);
```

Legg merke til likheten med en klassedefinisjon i Java:

```
public class EmneStudent {
        String brukernavn;
        String emnekode;
        int gruppenr;
        Date eksamensdato;
}
```

Datatyper sortert etter hyppighet i IN2090

datatype	*forklaring*
`varchar(n)`	tekst med variabel lengde
`int integer`	heltall
`date`	dato
`time`	klokkeslett
`char(n)`	tekst med fast lengde
`boolean`	boolsk verdi
`real`	flyttall
`numeric(n,d)`	*n* desimale sifre, *d* etter desimalpunktum
`timestamp`	dato og klokkeslett
`bit(n)`	bitstreng med fast lengde
`bit varying(n)`	bitstreng med variabel lengde

create—eksempel med skranker

Student(bnavn, snr, navn, stprog)

```
create table Student (
       bnavn char(8) primary key,
       snr int unique,
       navn varchar(80) not null,
       stprog varchar(40)
);
```

create—eksempel med skranker

Student(bnavn, snr, navn, stprog)

- bnavn er studentens brukernavn, og *primærnøkkel*

```
create table Student (
       bnavn char(8) primary key,
       snr int unique,
       navn varchar(80) not null,
       stprog varchar(40)
);
```

create—eksempel med skranker

Student(bnavn, snr, navn, stprog)

- bnavn er studentens brukernavn, og *primærnøkkel*
- snr er studentens studentnummer, og *supernøkkel*

```
create table Student (
       bnavn char(8) primary key,
       snr int unique,
       navn varchar(80) not null,
       stprog varchar(40)
);
```

create—eksempel med skranker

Student(bnavn, snr, navn, stprog)

- bnavn er studentens brukernavn, og *primærnøkkel*
- snr er studentens studentnummer, og *supernøkkel*
- navn er studentens navn, og kan *ikke være null*

```
create table Student (
       bnavn char(8) primary key,
       snr int unique,
       navn varchar(80) not null,
       stprog varchar(40)
);
```

create—eksempel med skranker

Student(bnavn, snr, navn, stprog)

- bnavn er studentens brukernavn, og *primærnøkkel*
- snr er studentens studentnummer, og *supernøkkel*
- navn er studentens navn, og kan *ikke være null*
- stprog er studieprogrammet studenten er tatt opp til

```
create table Student (
       bnavn char(8) primary key,
       snr int unique,
       navn varchar(80) not null,
       stprog varchar(40)
);
```

Primærnøkler

- Kan deklareres i **create table** sammen med primærnøkkelattributtet (bare hvis attributtet utgjør primærnøkkelen alene)

```
create table Emner (
        emnekode varchar(12) primary key,
        ...
);
```

Primærnøkler

- Kan deklareres i **create table** sammen med primærnøkkelattributtet (bare hvis attributtet utgjør primærnøkkelen alene)

```
create table Emner (
        emnekode varchar(12) primary key,
        ...
);
```

- Kan deklareres separat i **create table** etter attributtdeklarasjonene

```
create table Emner (
        emnekode varchar(12),
        ...
        primary key(emnekode)
);
```

Primærnøkler

- Hvis primærnøkkelen består av flere attributter må definisjonen kome etter attributtdeklarasjonene

```
create table eksamensforsøk (
        brukernavn char(8)
        emne  char(10)
        semester char(5),
        karakter char(1),
        primary key (brukernavn, emne, semester)
);
```

Forsøk på å legge inn flere karakterer for samme student i samme emne i samme semester vil da feile.

Primærnøkler, regler

- Maks én primærnøkkeldeklarasjon pr. relasjon

Primærnøkler, regler

- Maks én primærnøkkeldeklarasjon pr. relasjon
- Flere kandidatnøkler, bruk skranken **unique**

Primærnøkler, regler

- Maks én primærnøkkeldeklarasjon pr. relasjon
- Flere kandidatnøkler, bruk skranken **unique**
- Konsekvenser av deklarasjonen:
  - To tupler i relasjonen får ikke stemme overens i alle attributtene i primærnøkkelen. Forsøk på brudd ved **insert** eller **update** skal avvises av DBMSet
  - Attributtene i primærnøkkelen får ikke inneholde **null**

Primærnøkler, regler

- Maks én primærnøkkeldeklarasjon pr. relasjon
- Flere kandidatnøkler, bruk skranken **unique**
- Konsekvenser av deklarasjonen:
  - To tupler i relasjonen får ikke stemme overens i alle attributtene i primærnøkkelen. Forsøk på brudd ved **insert** eller **update** skal avvises av DBMSet
  - Attributtene i primærnøkkelen får ikke inneholde **null**
- Dette må sjekkes av systemet ved hver **insert** og hver **update**

Skranke på ett attributt

- not null
  - **create table** Ansatt (... Fdato int **not null**, ..);
  - Konsekvenser:
    - Kan ikke sette inn tuppel med verdien null i attributtet
    - Kan ikke endre verdien til null senere

Skranke på ett attributt

- not null
  - **create table** Ansatt (... Fdato int **not null**, ..);
  - Konsekvenser:
    - Kan ikke sette inn tuppel med verdien null i attributtet
    - Kan ikke endre verdien til null senere
- check
  - ```
    create table Ansatt (
        ...
        Tittel varchar(15)
        check (Tittel='Selger'
               or Tittel='Konsulent' or ...
              ),
        ...
    );
    ```
  - Angir en betingelse på attributtet. Sjekkes ved hver endring av attributtets verdi
    ```

Skranke på ett attributt

- check
  - ```
    create table Ansatt (
        Navn varchar(40),
        Tittel varchar(15),
        Fnr varchar(11),
        ... ,
        CONSTRAINT sjekkTittel check
           ( Tittel='Selger' or
             Tittel='Konsulent' or ...),
        ...
    );
    ```
    ```

Skranke på ett attributt

- check
  - ```
    create table Ansatt (
        Navn varchar(40),
        Tittel varchar(15),
        Fnr varchar(11),
        ... ,
        CONSTRAINT sjekkTittel check
           ( Tittel='Selger' or
             Tittel='Konsulent' or ...),
        ...
    );
    ```
- check
  - ```
    create table Emne (
        eKode varchar(7),
        eNavn varchar(99),
        ... ,
        CONSTRAINT sjekkEkode check (ekode > 'INF0000'
                                     and ekode < 'INF9999'),
        ...
    );
    ```

Skranke på ett attributt

- check
  - ```
    create table Ansatt (
        Navn varchar(40),
        Tittel varchar(15),
        Fnr varchar(11),
        ... ,
        CONSTRAINT sjekkTittel check
           ( Tittel='Selger' or
             Tittel='Konsulent' or ...),
        ...
    );
    ```
- check
  - ```
    create table Emne (
        eKode varchar(7),
        eNavn varchar(99),
        ... ,
        CONSTRAINT sjekkEkode
                    check (ekode like 'INF____'),
        ...
    );
    ```

Skranker: fremmednøkler

- **Fremmednøkkel**: Ett eller flere attributter som *peker ut/refererer* et tuppel i en annen relasjon.

Personale

Ans#	Navn	Fdato	Pers#	Avd
10	Gro	290264	39201	**null**
9	Berit	131172	35697	Knøttene
8	Bjørn	150571	34322	Knøttene
12	Liv	031079	39201	**null**

Barn

Løpe#	Navn	Fdato	Avd	TilknPers
2	Lisa	180502	Rosa Pantern	**null**
5	Trym	030205	Knøttene	9
4	Anne	301102	Tommeliten	**null**
7	Anne	151204	Knøttene	8

Hva skal vi med fremmednøkler?

- Hva skal vi med fremmednøkler?

- Hva skal vi med fremmednøkler?
- Knytte sammen informasjon som hører sammen (f.eks. til samme objekt i virkeligheten) fra flere tabeller.

- Hva skal vi med fremmednøkler?
- Knytte sammen informasjon som hører sammen (f.eks. til samme objekt i virkeligheten) fra flere tabeller.
- Må vi definere fremmednøkler for å gjøre dette?

- Hva skal vi med fremmednøkler?
- Knytte sammen informasjon som hører sammen (f.eks. til samme objekt i virkeligheten) fra flere tabeller.
- Må vi definere fremmednøkler for å gjøre dette?
- Nei

- Hva skal vi med fremmednøkler?
- Knytte sammen informasjon som hører sammen (f.eks. til samme objekt i virkeligheten) fra flere tabeller.
- Må vi definere fremmednøkler for å gjøre dette?
- Nei
- Hvorfor gjør vi det da?

- Hva skal vi med fremmednøkler?
- Knytte sammen informasjon som hører sammen (f.eks. til samme objekt i virkeligheten) fra flere tabeller.
- Må vi definere fremmednøkler for å gjøre dette?
- Nei
- Hvorfor gjør vi det da?
- For å sikre at alle «pekere» til en annen tabelle peker til noe.

- Hva skal vi med fremmednøkler?
- Knytte sammen informasjon som hører sammen (f.eks. til samme objekt i virkeligheten) fra flere tabeller.
- Må vi definere fremmednøkler for å gjøre dette?
- Nei
- Hvorfor gjør vi det da?
- For å sikre at alle «pekere» til en annen tabelle peker til noe.
- For at DBMSet skal gjøre oppslag i tabeller mer effektivt.

Skranker: fremmednøkler

- *med ett attributt, samme attributtnavn*:

```
create table StudTarEmne (
    .... ,
    emnekode char(10) references Emner ,
    ....
);
```

- *med ett attributt, ulike attributtnavn*:

```
create table StudTarEmne (
    .... ,
    emnekode char(10) references Kurs(kurskode)  ,
    ....
);
```

- *med flere attributter med samme navn*:

```
create table StudFikkKarakterIEmne (
   studid int ,
   emnekode varchar(10) ,
   semester varchar(5) ,
   karakter char(1) ,

   foreign key (studid, emnekode, semester)
          references Resultater

);
```

De refererte attributtene må være primærnøkkel, deklarert **primary key**. (I noen SQL-systemer holder det med at attributtene er deklarert UNIQUE, men ikke i PostgreSQL).

- *med flere attributter med forskjellig navn*:

```
create table StudFikkKarakterIEmne (
  studid int ,
  emnekode varchar(10) ,
  semester varchar(5) ,
  karakter char(1) ,

  foreign key (studid, emnekode, semester)
          references Resultater (bnavn, kurskode, sem)

);
```

De refererte attributtene må være primærnøkkel, deklarert **primary key**. (I noen SQL-systemer holder det med at attributtene er deklarert UNIQUE, men ikke i PostgreSQL).

Fremmednøkler mot flere tabeller brukes for å implementere et mange-til-mange forhold mellom tabeller:

```
create table student (
    bnavn char(8) primary key,
    navn varchar(80),
    ...
);

create table emne (
    ekode char(10) primary key,
    emnenavn varchar(80),
    emneeier varchar(80),
    ...
);

create table antalleksamensforsøk (
    brukernavn char(8) references student(bnavn),
    emne  char(10) references emne(ekode),
    antforsøk integer,
    primary key (brukernavn, emne)
);
```

Legge inn data i tabeller

insert into $R(A_1, A_2, \ldots, A_k)$
values $(v_1, v_2, \ldots, v_k)$;

- Attributtlisten kan sløyfes hvis den dekker samtlige attributter i R og følger attributtenes default rekkefølge

Legge inn data i tabeller

insert into $R(A_1, A_2, \ldots, A_k)$
values $(v_1, v_2, \ldots, v_k)$;

```
insert into Student (snr, navn, bnavn, stprog)
values (133423, 'Liv E. Laga', 'livelaga',
        'Informatikk: programmering og nettverk'
);

insert into Student
values ('alistrak', 133424, 'Ali Straks',
       'Informatikk: design, bruk, interaksjon'
);
```

Husk attributtrekkefølgen fra definisjonen av Student:
Student (bnavn, snr, navn, stprog)

insert

insert into $R(A_1, A_2, \ldots, A_k)$
values $(v_1, v_2, \ldots, v_k)$;

insert into $R(A_1, A_2, \ldots, A_k)$
select-setning;

insert

insert into $R(A_1, A_2, \ldots, A_k)$
values $(v_1, v_2, \ldots, v_k)$;

insert into $R(A_1, A_2, \ldots, A_k)$
select-setning;

- Attributtlisten kan sløyfes hvis den dekker samtlige attributter i R og følger attributtenes default rekkefølge

insert

insert into $R(A_1, A_2, \ldots, A_k)$
values $(v_1, v_2, \ldots, v_k)$;

insert into $R(A_1, A_2, \ldots, A_k)$
select-setning;

- Attributtlisten kan sløyfes hvis den dekker samtlige attributter i R og følger attributtenes default rekkefølge
- **NB**—optimaliseringer i DBMSet kan medføre at tuplene legges inn etterhvert som de beregnes i **select**-setningen. Dette kan ha sideeffekter på beregningen av **select**-setningen