

INF105 - 21
relasjonelle DB / normalisering

David Grellscheid

ingen forelesning i morgen

Hvordan organisere dette på en bedre måte?

AKVAKULTURTILLATELSER PR. 13.04.2020							
TILL_NR	ORG.NR/PERS	NAVN	ADRESSE	POSTNR	POSTSTED	TILDELINGSTIDS	ART
A A 0001	969159570	NORGES MILJØ- OG BIOVITENSKAPELIGE	POSTBOKS 5003	1432	ÅS	03-10-1991	Regnbueørret
A A 0001	969159570	NORGES MILJØ- OG BIOVITENSKAPELIGE	POSTBOKS 5003	1432	ÅS	03-10-1991	Ørret
A A 0001	969159570	NORGES MILJØ- OG BIOVITENSKAPELIGE	POSTBOKS 5003	1432	ÅS	03-10-1991	Laks
A F 0001	855869942	NORSK INSTITUTT FOR VANNFORSKNING	GAUSTADALLÉEN 21	349	OSLO	05-02-1991	Ørret
A F 0001	855869942	NORSK INSTITUTT FOR VANNFORSKNING	GAUSTADALLÉEN 21	349	OSLO	05-02-1991	Laks
A F 0001	855869942	NORSK INSTITUTT FOR VANNFORSKNING	GAUSTADALLÉEN 21	349	OSLO	05-02-1991	Regnbueørret
A F 0002	855869942	NORSK INSTITUTT FOR VANNFORSKNING	GAUSTADALLÉEN 21	349	OSLO	31-03-2017	Rognkjeks (felles)
A F 0003	855869942	NORSK INSTITUTT FOR VANNFORSKNING	GAUSTADALLÉEN 21	349	OSLO	01-06-2017	Brungylt
A F 0003	855869942	NORSK INSTITUTT FOR VANNFORSKNING	GAUSTADALLÉEN 21	349	OSLO	01-06-2017	Berggylt
A F 0003	855869942	NORSK INSTITUTT FOR VANNFORSKNING	GAUSTADALLÉEN 21	349	OSLO	01-06-2017	Gressgylt
A F 0003	855869942	NORSK INSTITUTT FOR VANNFORSKNING	GAUSTADALLÉEN 21	349	OSLO	01-06-2017	Havabbor
A F 0003	855869942	NORSK INSTITUTT FOR VANNFORSKNING	GAUSTADALLÉEN 21	349	OSLO	01-06-2017	Sandflyndre

DB-modellering

Relasjonell modellering

Relationell modellering

- E.F.Codd, 1970: "Relational model"
- relasjonelle DB ble til de mest populære
- basert på matematisk grunnlag i mengdeteori
- ingen grafisk representasjon, må ikke forveksles med ER

A Relational Model of Data for Large Shared Data Banks

E. F. Codd
IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on *n*-ary relations, a normal form for data base relations, and the concept of a universal data sublanguage are introduced. In Section 2, certain operations on relations (other than logical inference) are discussed and applied to the problems of redundancy and consistency in the user's model.

Relationell modellering

- En database er en samling av ***relasjoner***

- Hver relasjon er en mengde av ***tuples***. Hver av de beskriver noe lignende fra den reelle verden

- En tuple er en sortert samling av attributtene som beskriver ulike aspekter til enheten.

Relation tilsvarer Entity eller Relationship

SUPPLIER

SUPNR	SUPNAME	SUPADDRESS	SUPCITY	SUPSTATUS
21	Deliwines	240, Avenue of the Americas	New York	20
32	Best Wines	660, Market Street	San Francisco	90
37	Ad Fundum	82, Wacker Drive	Chicago	95
52	Spirits & co.	928, Strip	Las Vegas	NULL
68	The Wine Depot	132, Montgomery Street	San Francisco	10
69	Vinos del Mundo	4, Collins Avenue	Miami	92
...				

Tuple
Tuple
Tuple
Tuple

Attribute Attribute Attribute Attribute Attribute

Relation **Attributes**

```
Student (Studentnr, Name, HomePhone, Address)

Professor (SSN, Name, HomePhone, OfficePhone,
E-mail)

Course (CourseNo, CourseName)
```

$$R(A_1, A_2, A_3, \ldots, A_n)$$

Hvert attribut har en ***domene (domain)*** dom(A_i), som er alle mulige verdier til A_i.

Relation Attributes

$$R\,(A_1, A_2, A_3, \cdots, A_n)$$

$$\{$$

Set of tuples

$$t_1 = \langle v_1, v_2, v_3, \cdots, v_n \rangle$$

$$t_2 = \langle v_1, v_2, v_3, \cdots, v_n \rangle$$

$$t_3 = \langle v_1, v_2, v_3, \cdots, v_n \rangle$$

$$\vdots$$

$$t_m = \langle v_1, v_2, v_3, \cdots, v_n \rangle$$

$$\}$$

SUPPLIER

SUPNR	SUPNAME	SUPADDRESS	SUPCITY	SUPS
21	Deliwines	240, Avenue of the Americas	New York	20
32	Best Wines	660, Market Street	San Francisco	90
37	Ad Fundum	82, Wacker Drive	Chicago	95
52	Spirits & co.	928, Strip	Las Vegas	NULL
68	The Wine Depot	132, Montgomery Street	San Francisco	10
69	Vinos del Mundo	4, Collins Avenue	Miami	92
...				

Hver verdi v_i:
enten fra dom(A_i) eller *NULL*

NULL:
verdien mangler / ikke relevant

Relation **Attributes**

$$R\,(A_1, A_2, A_3, \cdots, A_n)$$

ordnet tuple, rekkefølge er relevant

$$\{$$

$$t_1 = \langle v_1, v_2, v_3, \cdots, v_n \rangle$$

$$t_2 = \langle v_1, v_2, v_3, \cdots, v_n \rangle$$

Set of tuples

$$t_3 = \langle v_1, v_2, v_3, \cdots, v_n \rangle$$

uordnet, ingen repetisjon

$$\vdots$$

$$t_m = \langle v_1, v_2, v_3, \cdots, v_n \rangle$$

$$\}$$

ingen multiple verdier

farge

~~"rød, blå"~~

Hver verdi v_i:

enten fra dom(A_i) eller *NULL*

NULL:

verdien mangler / ikke relevant

Nøkler (keys)

- Superkey > **Key**
- Candidate Key / **Primary Key** / Alternative Key
  - = en Key vi *kan* velge / *har* valgt / *ikke* har valgt til identifikasjon av tupler
- **Foreign Key**
  - = et attribut som peker til en annen relasjon sin PK

Ulike nøkler / keys

Superkey

$$R\,(A_1, A_2, A_3, \cdots, A_n)$$

- delmengde av attributene A_i til R slik at ingen to *tuples* har de samme verdiene der $\iff$ superkeys har ***unique*** constraint
- superkey trenger ikke være minimal, kan ha redundante attributter
- `(Studentnr, Name, HomePhone)`
- $\implies$ hele $A_i, \cdots, A_n$ alltid er en superkey

Ulike nøkler / keys

Key

$$R\,(A_1, A_2, A_3, \cdots, A_n)$$

- En ***key*** er en *superkey* der ***ingen*** av attributtene kan droppes uten å miste uniqueness
- en *key* **ikke** har redundante attributter
- (Studentnr, Name, HomePhone)
- hver relasjon R har minst én key

Ulike nøkler / keys

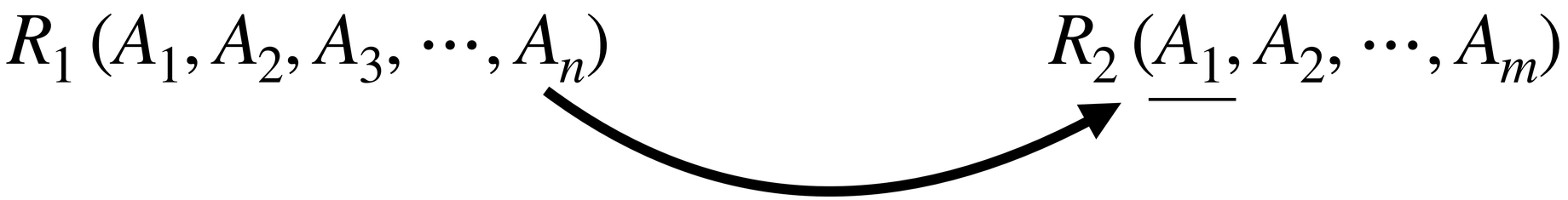
Candidate / primary (PK) / alternative key

$$R\,(A_1, A_2, A_3, \cdots, A_n)$$

- En relasjon kan ha flere mulige *keys*, som vi kaller for ***Candidate keys***
- `(bilskilt, chassisnr, merke, modell)`
- En ***primary key (PK)*** er den som ***vi har valgt*** til å identifisere tuplene
- De andre keys kalles ***alternative keys***
- *primary key* må være NOT NULL

Ulike nøkler / keys

Foreign key (FK)



- En mengde av attributter fra R_1 kalles for en ***foreign key FK*** når
  - attributtene i FK har de samme domenene som domenene til *primary key PK* til R_2
  - **En FK-verdi** til en tuple i R_1
    ***enten* finnes** som PK-verdi til en tuple i R_2
    ***eller* er NULL**

Relational Constraints

Domain constraint	The value of each attribute type A must be an atomic and single value from the domain dom(A).
Key constraint	Every relation has a key that allows to uniquely identify its tuples.
Entity integrity constraint	The attribute types that make up the primary key should always satisfy a NOT NULL constraint.
Referential integrity constraint	A foreign key FK has the same domain as the primary key PK attribute type(s) it refers to and either occurs as a value of PK or NULL.

Relation **Attributes**

```
Student (Studentnr, Name, HomePhone, Address)

Professor (SSN, Name, HomePhone, OfficePhone,
E-mail)

Course (CourseNo, CourseName)
```

CREATE TABLE student (studentnr PRIMARY KEY, name NOT NULL, homephone, address);

CREATE TABLE professor (ssn PRIMARY KEY, name NOT NULL, homephone, officephone, email);

CREATE TABLE course (courseno PRIMARY KEY, coursename NOT NULL);

sqlite3 movies.db

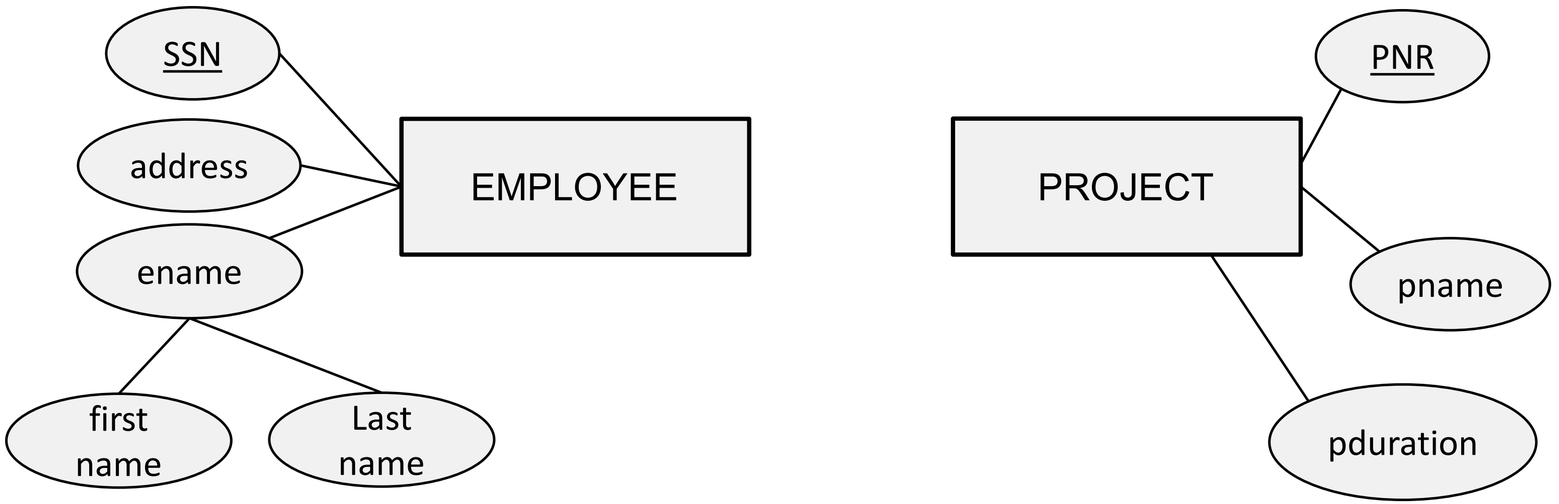
- .schema
- CREATE TABLE movies(
  id INTEGER PRIMARY KEY,
  title TEXT, language TEXT,
  year INTEGER,
  runtime INTEGER,
  budget INTEGER,
  revenue INTEGER);
- CREATE TABLE people(
  id INTEGER PRIMARY KEY,
  name TEXT);
- CREATE TABLE actors(
  movie_id INTEGER,
  actor_id INTEGER,
  character TEXT,

  PRIMARY KEY(movie_id, actor_id),
  FOREIGN KEY(movie_id)
  REFERENCES movies(id),
  FOREIGN KEY(actor_id)
  REFERENCES actors(id));
- sqlite trenger
  PRAGMA foreign_keys = ON;

fra ER til en relasjonell modell

(Lemahieu at.al 6.3/6.4)

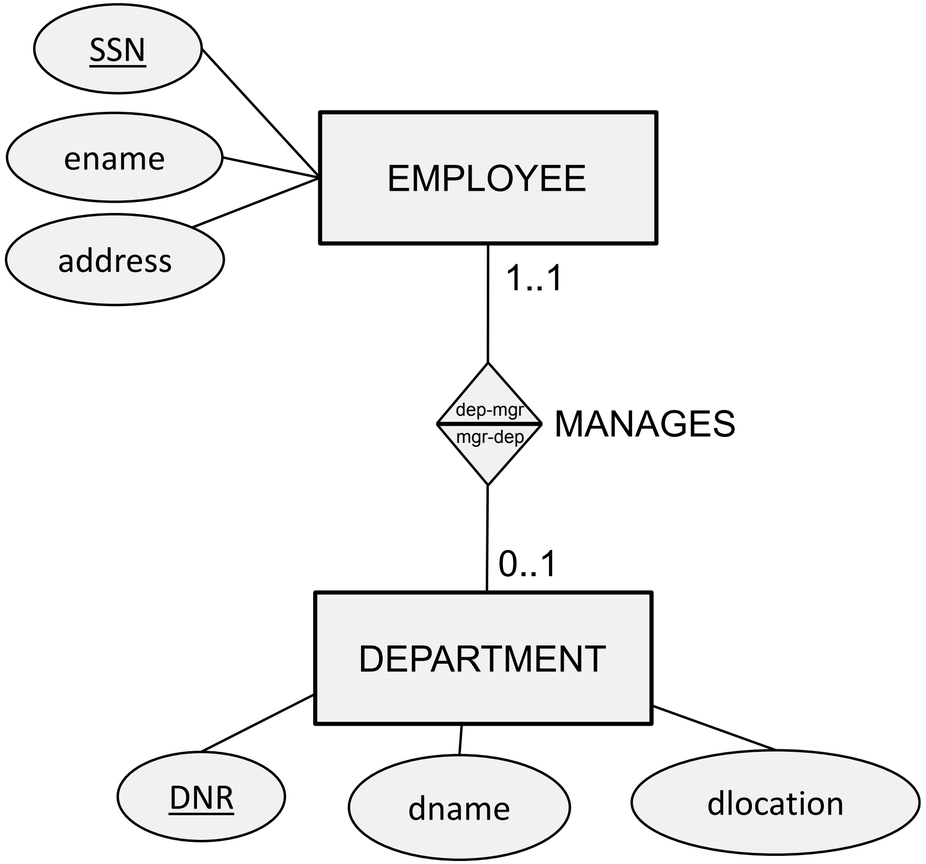
Mapping Entity Types



EMPLOYEE(SSN, address, first name, last name)

PROJECT(PNR, pname, pduration)

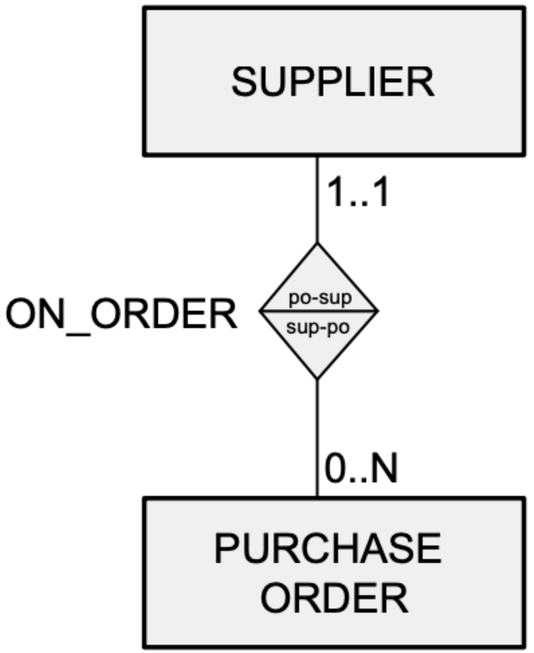
Mapping a Binary 1:1 Relationship Type



EMPLOYEE(SSN, ename, address, *DNR*)
DEPARTMENT(DNR, dname, dlocation)

EMPLOYEE(SSN, ename, address)
DEPARTMENT(DNR, dname, dlocation, *SSN*)

1-N relasjon i ER



SUPPLIER

SUPNR	SUPNAME	SUPADDRESS	SUPCITY	SUPSTATUS
...				
37	Ad Fundum	82, Wacker Drive	Chicago	95
94	The Wine Crate	330, McKinney Avenue	Dallas	75
...				

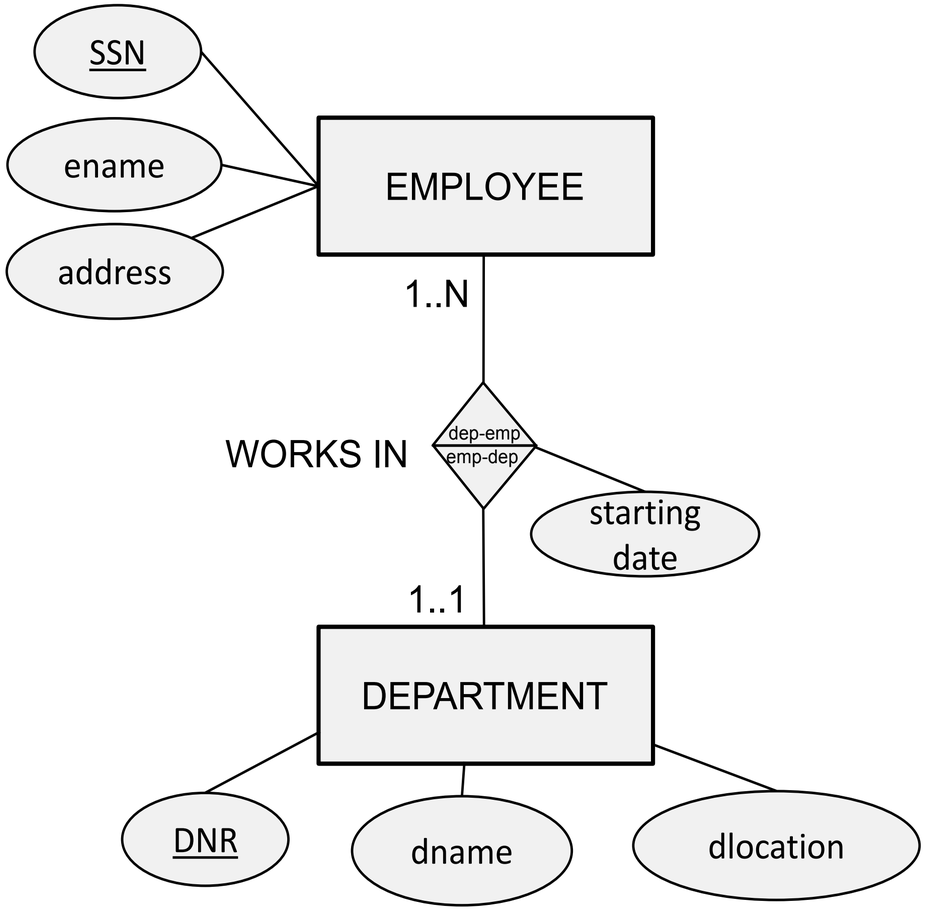
PURCHASE_ORDER

PONR	PODATE	*SUPNR*
1511	2015-03-24	37
1512	2015-04-10	94
...		

- 1-N relasjon kan modelleres direkte som FK

- Supplier (**Supnr**, Supname, Supaddress, Supcity, Supstatus)
  Purchase_Order (POnr, POdate, ***Supnr***)

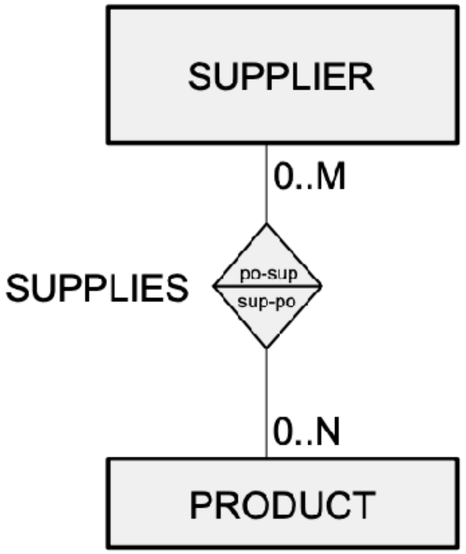
Mapping a Binary 1:N Relationship Type



EMPLOYEE(SSN, ename, address, starting date, *DNR*)

DEPARTMENT(DNR, dname, dlocation)

M-N relasjon i ER



SUPPLIER

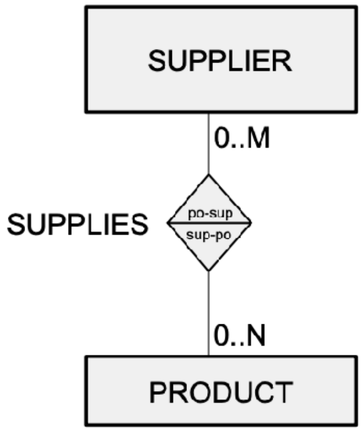
SUPNR	SUPNAME	SUPADDRESS	SUPCITY	SUPSTATUS
21	Deliwines	240, Avenue of the Americas	New York	20
32	Best Wines	660, Market Street	San Francisco	90
...				

PRODUCT

PRODNR	PRODNAME	PRODTYPE	AVAILABLE_QUANTITY
0119	Chateau Miraval, Cotes de Provence Rose, 2015	rose	126
0154	Chateau Haut Brion, 2008	red	111
...		red	5

- M-N relasjon kan ikke modelleres direkte som FK, kan ikke ha multiple verdier
- Løsning: lage en ny relasjons-table

M-N relasjon i ER



SUPPLIER

SUPNR	SUPNAME	SUPADDRESS	SUPCITY	SUPSTATUS
21	Deliwines	240, Avenue of the Americas	New York	20
32	Best Wines	660, Market Street	San Francisco	90
...				

PRODUCT

PRODNR	PRODNAME	PRODTYPE	AVAILABLE_QUANTITY
0119	Chateau Miraval, Cotes de Provence Rose, 2015	rose	126
0154	Chateau Haut Brion, 2008	red	111
...		red	5

SUPPLIES

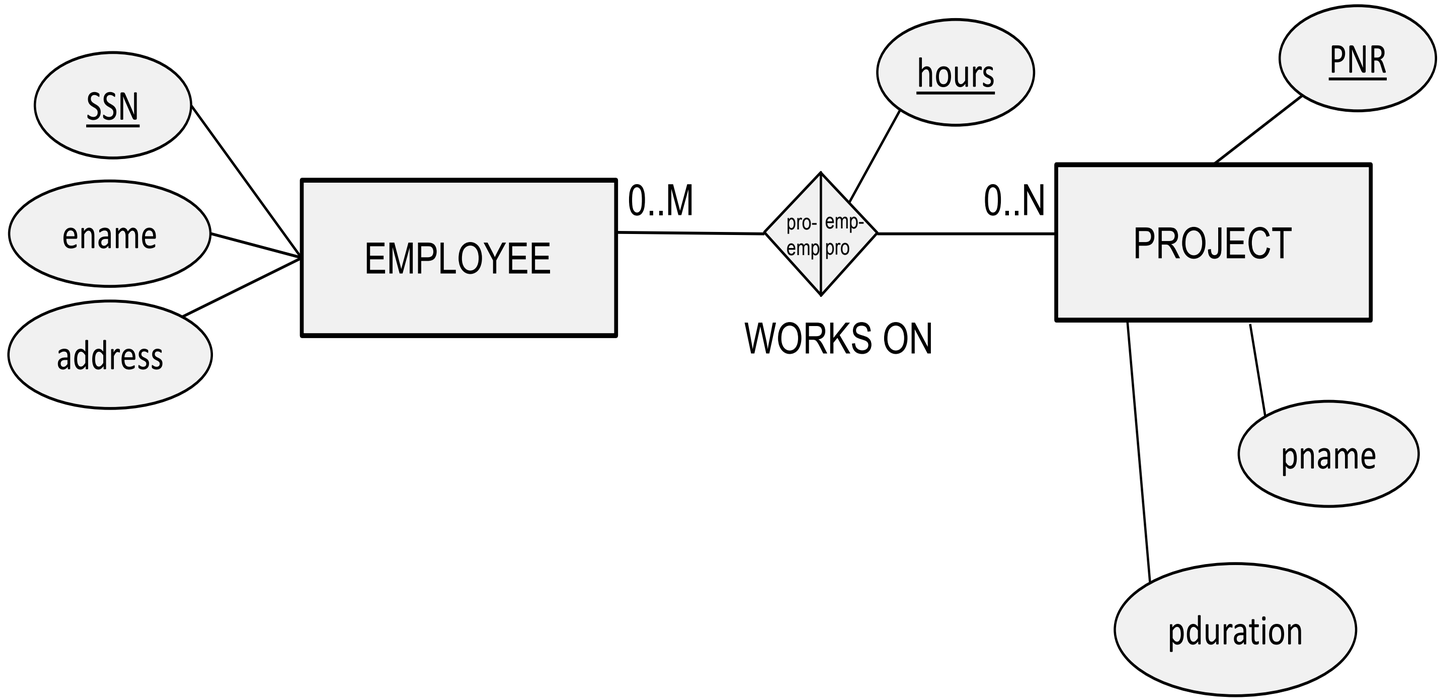
SUPNR	***PRODNR***	PURCHASE_PRICE	DELIV_PERIOD
...			
68	0327	56.99	4
...			
21	0289	17.99	1
21	0327	56.00	6
21	0347	16.00	2
...			
69	0347	18.00	4
84	0347	18.00	4
...			

Supplies (*Supnr*, *Prodnr*, ...)

FK FK

PK

Mapping a Binary M:N Relationship Type

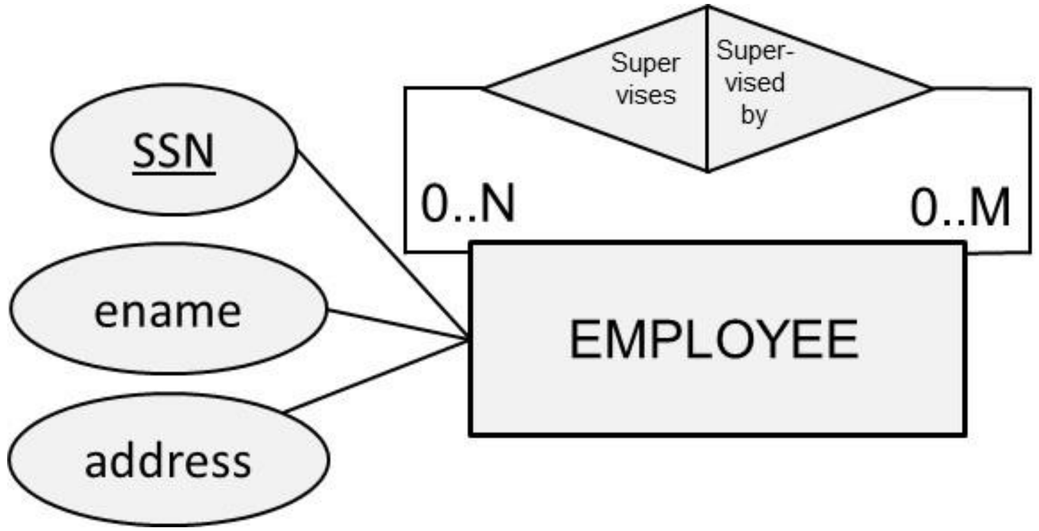


EMPLOYEE(<u>SSN</u>, ename, address)

PROJECT(<u>PNR</u>, pname, pduration)

WORKS_ON(<u>*SSN*</u>, <u>*PNR*</u>, hours)

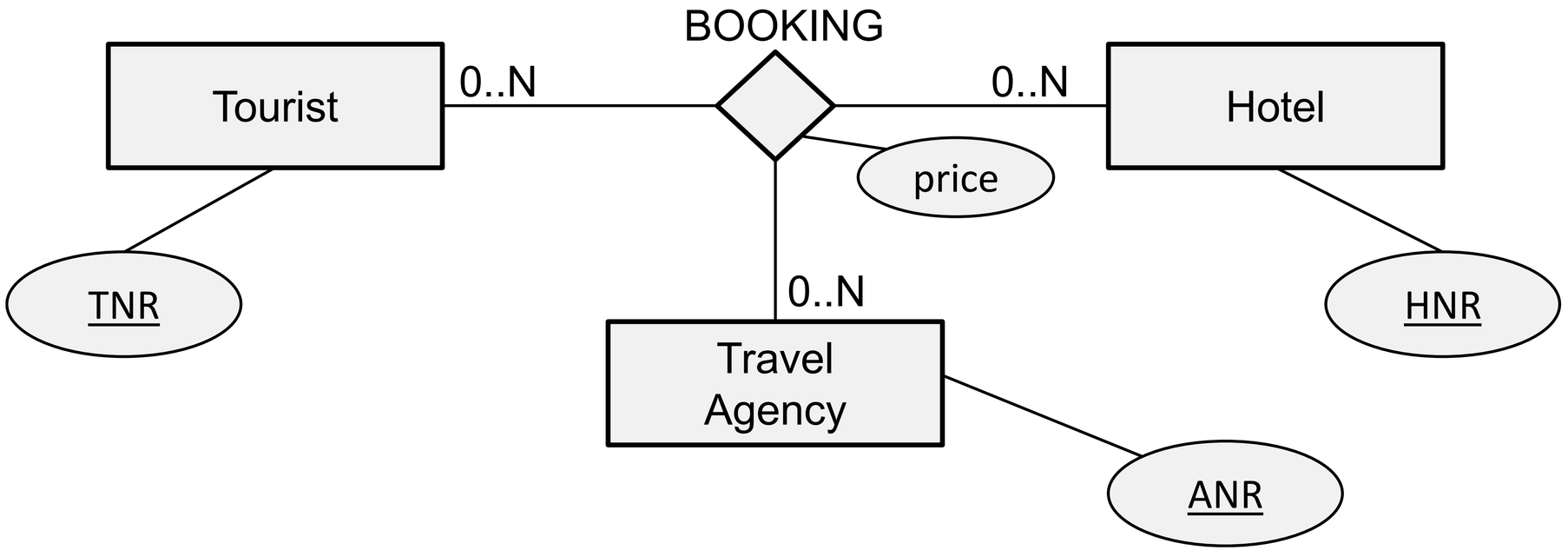
Mapping Unary Relationship Types (M-N)



EMPLOYEE(SSN, ename, address)

SUPERVISION(*Supervisor*, *Supervisee*)

Mapping n-ary Relationship Types



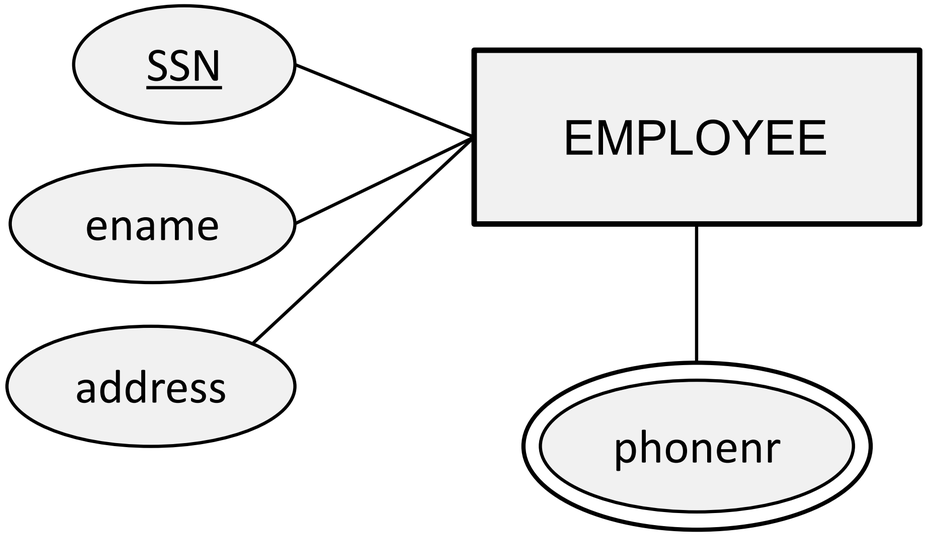
TOURIST(TNR, …)

TRAV_AGENCY(ANR, …)

HOTEL(HNR, …)

BOOKING(*TNR*, *ANR*, *HNR*, price)

Mapping Multivalued Attribute Types



EMPLOYEE(SSN, ename, address)

EMP-PHONE(PhoneNr, *SSN*)

DB-normalisering / DB normalization

AKVAKULTURTILLATELSER PR. 13.04.2020							
TILL_NR	ORG.NR/PERS	NAVN	ADRESSE	POSTNR	POSTSTED	TILDELINGSTIDS	ART
A A 0001	969159570	NORGES MILJØ- OG BIOVITENSKAPELIGE	POSTBOKS 5003	1432	ÅS	03-10-1991	Regnbueørret
A A 0001	969159570	NORGES MILJØ- OG BIOVITENSKAPELIGE	POSTBOKS 5003	1432	ÅS	03-10-1991	Ørret
A A 0001	969159570	NORGES MILJØ- OG BIOVITENSKAPELIGE	POSTBOKS 5003	1432	ÅS	03-10-1991	Laks
A F 0001	855869942	NORSK INSTITUTT FOR VANNFORSKNING	GAUSTADALLÉEN 21	349	OSLO	05-02-1991	Ørret
A F 0001	855869942	NORSK INSTITUTT FOR VANNFORSKNING	GAUSTADALLÉEN 21	349	OSLO	05-02-1991	Laks
A F 0001	855869942	NORSK INSTITUTT FOR VANNFORSKNING	GAUSTADALLÉEN 21	349	OSLO	05-02-1991	Regnbueørret
A F 0002	855869942	NORSK INSTITUTT FOR VANNFORSKNING	GAUSTADALLÉEN 21	349	OSLO	31-03-2017	Rognkjeks (felles)
A F 0003	855869942	NORSK INSTITUTT FOR VANNFORSKNING	GAUSTADALLÉEN 21	349	OSLO	01-06-2017	Brungylt
A F 0003	855869942	NORSK INSTITUTT FOR VANNFORSKNING	GAUSTADALLÉEN 21	349	OSLO	01-06-2017	Berggylt
A F 0003	855869942	NORSK INSTITUTT FOR VANNFORSKNING	GAUSTADALLÉEN 21	349	OSLO	01-06-2017	Gressgylt
A F 0003	855869942	NORSK INSTITUTT FOR VANNFORSKNING	GAUSTADALLÉEN 21	349	OSLO	01-06-2017	Havabbor
A F 0003	855869942	NORSK INSTITUTT FOR VANNFORSKNING	GAUSTADALLÉEN 21	349	OSLO	01-06-2017	Sandflyndre

- En god relasjonell modell bør ***normaliseres:***
  - enklere å skjønne datamodellen og skrive gode spørringer
  - mindre risiko av inkonsistens
    ved insert/delete/update (f.eks her: adresseendring)

SUPPLIES

SUPNR	*PRODNR*	PURCHASE_PRICE	DELIV_PERIOD	*SUPNAME*	*SUPADDRESS*	...	*PRODNAME*	*PRODTYPE*	...
21	0289	17.99	1	*Deliwines*	*240, Avenue of the Americas*		*Chateau Saint Estève de Neri, 2015*	*Rose*	
21	0327	56.00	6	*Deliwines*	*240, Avenue of the Americas*		*Chateau La Croix Saint-Michel, 2011*	*Red*	
...									

PO_LINE

PONR	*PRODNR*	QUANTITY	*PODATE*	*SUPNR*
1511	0212	2	*2015-03-24*	*37*
1511	0345	4	*2015-03-24*	*37*
...				

Det finnes en formell prosedyre som kan overføre en unormalisert modell til en normalisert modell

Supplier

SUPNR	SUPNAME	SUPADDRESS	SUPCITY	SUPSTATUS
21	Deliwines	240, Avenue of the Americas	New York	20
32	Best Wines	660, Market Street	San Francisco	90
...				

Product

PRODNR	PRODNAME	PRODTYPE	AVAILABLE_QUANTITY
0119	Chateau Miraval, Cotes de Provence Rose, 2015	rose	126
0384	Dominio de Pingus, Ribera del Duero, Tempranillo, 2006	red	38
...			

Supplies

SUPNR	PRODNR	PURCHASE_PRICE	DELIV_PERIOD
21	0119	15.99	1
21	0384	55.00	2
...			

Purchase_Order

PONR	PODATE	SUPNR
1511	2015-03-24	37
1512	2015-04-10	94
...		

PO_Line

PONR	PRODNR	QUANTITY
1511	0212	2
1511	0345	4
...		

Normalisering kort oppsummert

- Lag modeller som er enkelt å forklare (gode relasjonsnavn)
- Ikke bland attributter fra ulike enheter i én relasjon
- Unngå behov for attributter med mange *NULL*-verdier

```
SUPPLIER(SUPNR, SUPNAME, SUPTWITTER, …)
```

```
SUPPLIER(SUPNR, SUPNAME, …)
SUPPLIER-TWITTER(SUPNR, SUPTWITTER)
```

Definisjon: funksjonell avhengig
(functionally dependent)

- To mengder med attributter X og Y er funksjonell avhengig $X \rightarrow Y$ når
  - verdien til X unikt bestemmer verdien til Y
    - ansattnummer -> navn
    - prosjektnummer -> prosjektnavn, sted
    - (ansatt, prosjekt) -> antall_timer

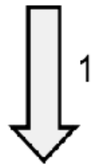
Normaliseringsformer

- 1st normal form (1NF)
- 2nd normal form (2NF)
- 3rd normal form (3NF)
- Boyce-Codd normal form (BCNF / 3.5NF)
- 4th, 5th, ... normal form

1 NF

- Hver attribut i en relasjon må ha en enkel verdi, ikke en kombinasjon ($\Longleftrightarrow$ domain constraint)

DNUMBER	**DLOCATION**	**DMGRSSN**
15	{New York, San Francisco}	110
20	Chicago	150
30	{Chicago, Boston}	100



DEPARTMENT

DNUMBER	***DMGRSSN***
15	110
20	150
30	100

DEP-LOCATION

DNUMBER	**DLOCATION**
15	New York
15	San Francisco
20	Chicago
30	Chicago
30	Boston

2 NF

- En relasjon R **er i 2NF** hvis den er 1NF og
  - alle ikke-key attributene A er *avhengig* av enhver *key* til R

    eller

    det finnes ingen key-attributene som et A ikke er avhengig av

2 NF

R1(SSN, PNUMBER, PNAME, HOURS)

R11(SSN, *PNUMBER*, HOURS)
R12(PNUMBER, PNAME)

(SSN irrelevant)

SSN	PNUMBER	PNAME	HOURS
100	1000	Hadoop	50
220	1200	CRM	200
280	1000	Hadoop	40
300	1500	Java	100
120	1000	Hadoop	120

2 NF

PNUMBER	PNAME
1000	Hadoop
1200	CRM
1500	Java

SSN	*PNUMBER*	HOURS
100	1000	50
220	1200	200
280	1000	40
300	1500	100
120	1000	120

3 NF

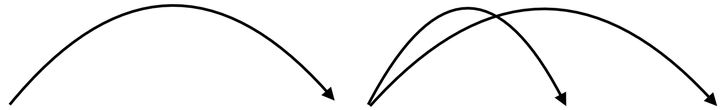
- En relasjon R er i 3NF hvis den er 2NF og
  - ingen ikke-key attributene A er *transitively dependent* på *primary key* til R
    $\iff$
    det finnes ingen ikke-key attributene B der $PK \rightarrow B \rightarrow A$
- Hvis en relasjon ikke er i 3NF, så må den deles opp i flere relasjoner, der B blir en key i én av dem

3 NF

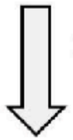
R1(SSN, ENAME, DNUMBER, DNAME, DMGRSSN)

R11(SSN, ENAME, *DNUMBER*)

R12(DNUMBER, DNAME, *DMGRSSN*)



SSN	NAME	DNUMBER	DNAME	*DMGRSSN*
10	O'Reilly	10	Marketing	210
22	Donovan	30	Logistics	150
28	Bush	10	Marketing	210
30	Jackson	20	Finance	180
12	Thompson	10	Marketing	210



3 NF

SSNR	NAME	*DNUMBER*
10	O'Reilly	10
22	Donovan	30
28	Bush	10
30	Jackson	20
12	Thompson	10

DNUMBER	DNAME	*DMGRSSN*
10	Marketing	210
30	Logistics	150
20	Finance	180

BCNF / 3.5NF

- BCNF er strengere en 3NF
- En relasjon R er i BCNF hvis i enhver avhengighet $X \rightarrow Y$, X er en superkey til R



avhengig (men SUPNR ikke SK til R1)

R11(*SUPNR*, PRODNR, QUANTITY)
R12(SUPNR, SUPNAME)

Normaliseringsformer

uformell oppsummert

- **1st normal form (1NF)**
  ingen multiple verdier i et attributt

- **2nd normal form (2NF)**
  ingen attributter som krever bare en del av nøklen

- **3rd normal form (3NF)**
  ingen "mellomledd" som selv kan bli nøkkel til noen attributter

- **Boyce-Codd normal form (BCNF / 3.5NF)**
  for alle $X \rightarrow Y$ *som finnes i en relasjon R, X må være en superkey til R*

Normalisering i praksis - et eksempel

- Definer en dårlig normalisert SQL-tabell som tilsvarer alle CSV-kolonnene
- Last inn fra CSV
- Tenk opp et nytt, normalisert tabellskjema
- Definer nye tabeller fra ulike kolonner av den eksisterende:

  CREATE TABLE ... AS SELECT ...
- Må begynne med de som ikke har noen FK
- Fix problems :-)

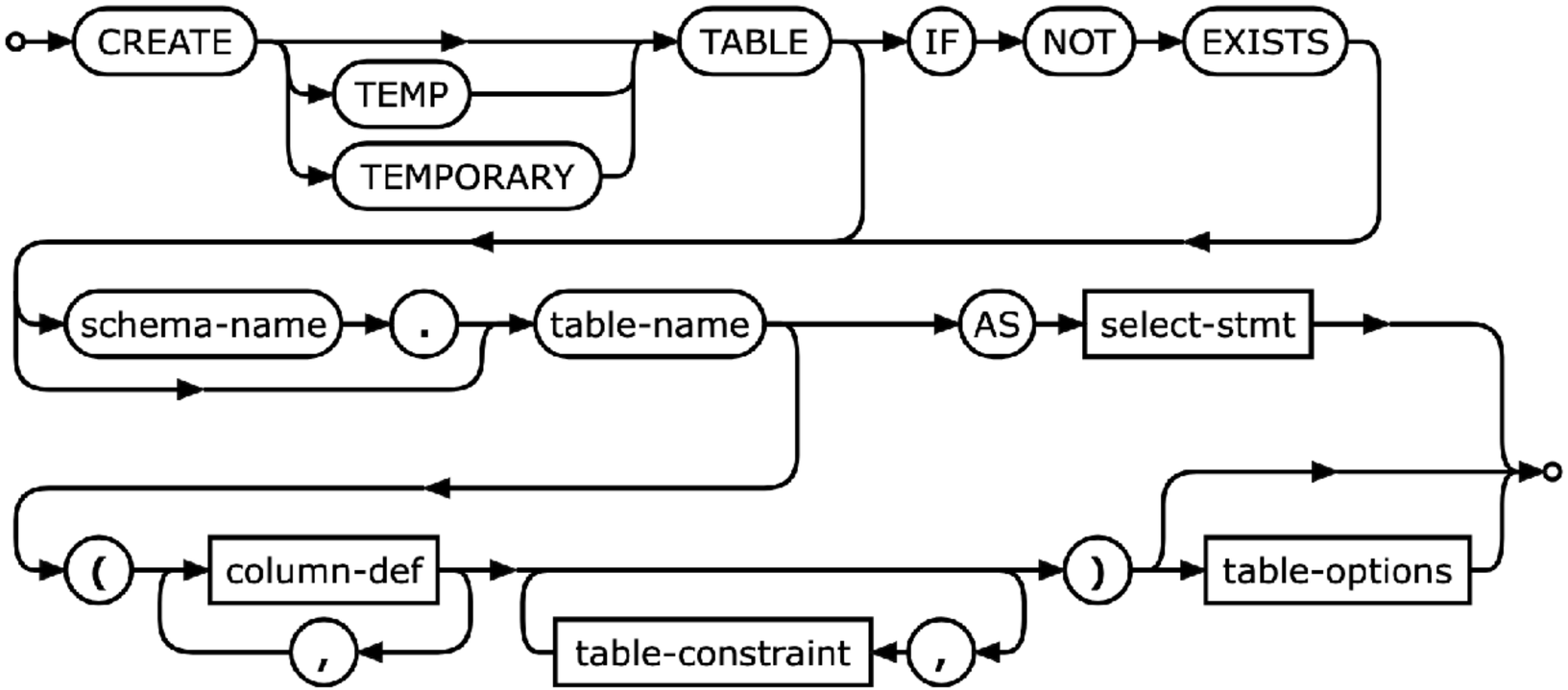
SQL-spørringer

må øves praktisk, nytter ikke å ha i en forelesning

SQL-spørringer *(queries)*

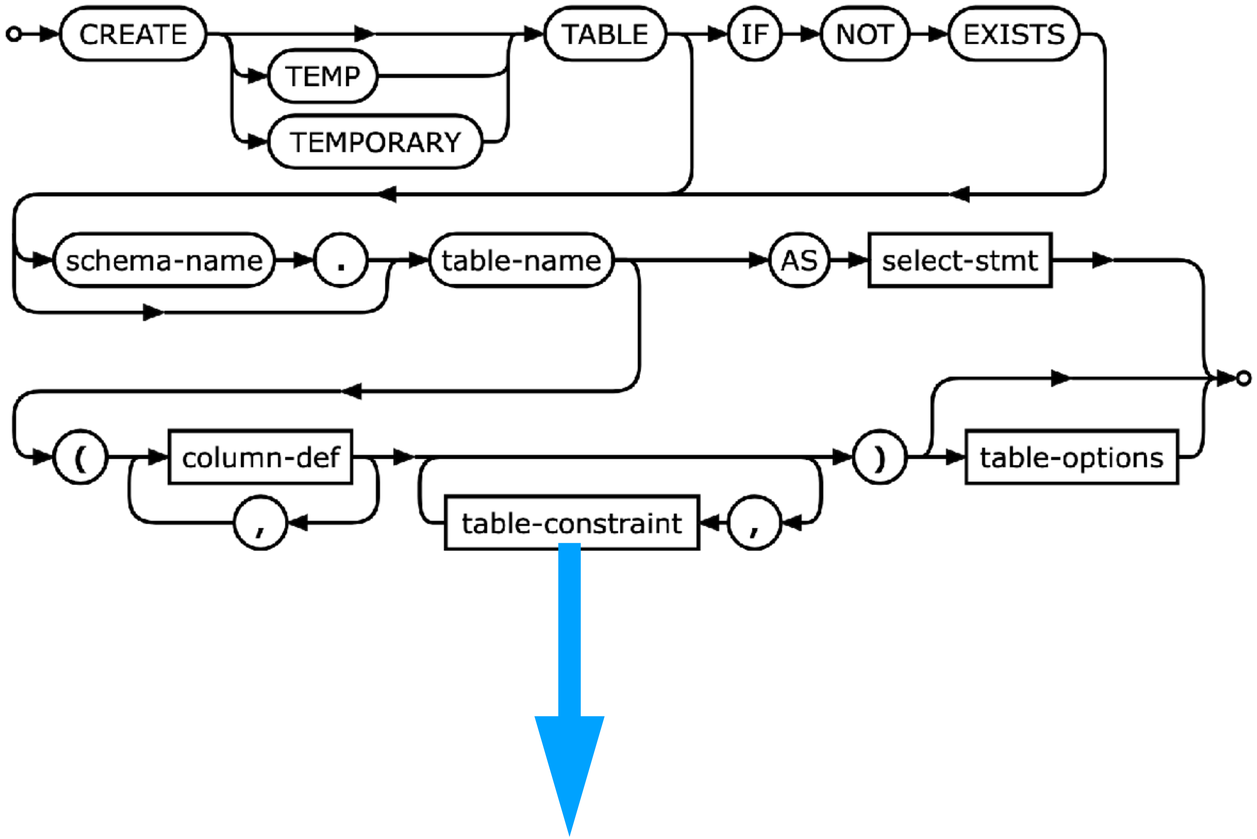
DDL - Data definition language

- CREATE TABLE https://www.sqlite.org/lang_createtable.html



SQL-spørringer *(queries)*

DDL - Data definition language

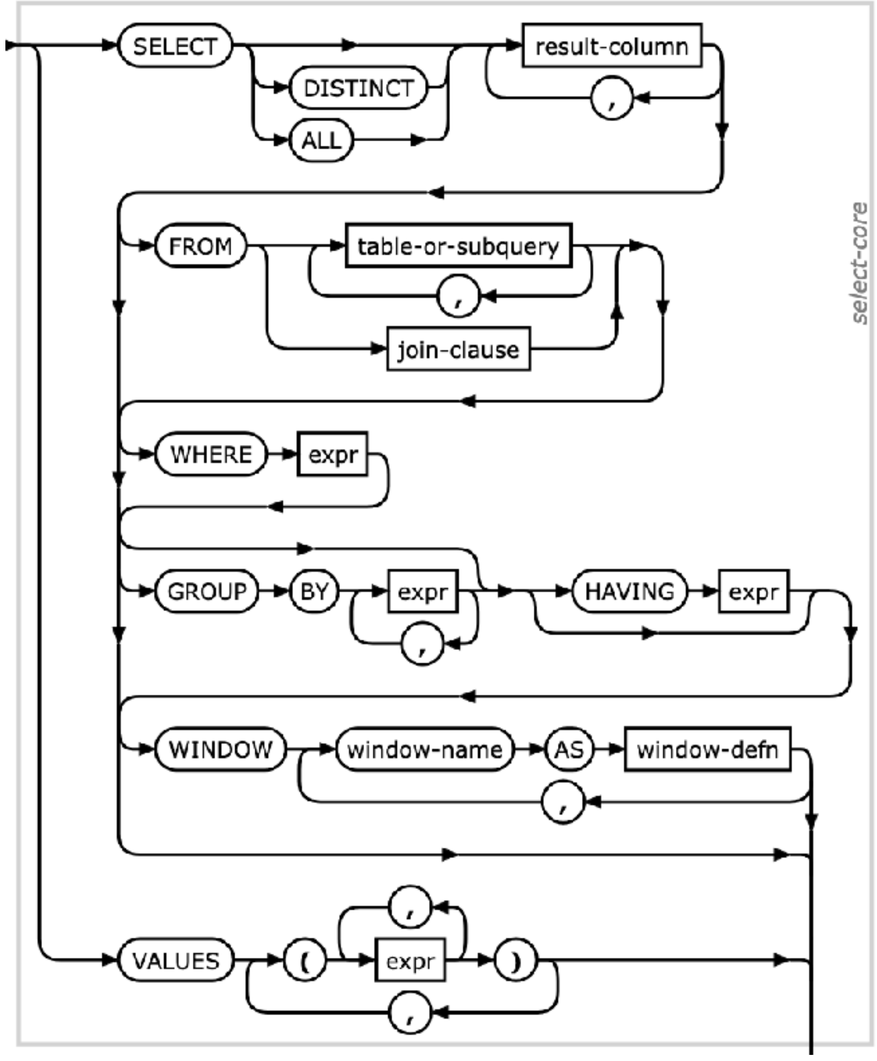


SQLite supports UNIQUE, NOT NULL, CHECK and FOREIGN KEY constraints.

SQL-spørringer *(queries)*

DML - Data manipulation language

- SELECT ... FROM ... WHERE
- ... GROUP BY ... (HAVING) (aggregate functions max,min,count,...)
- ... ORDER BY ... ASC/DESC
- ... LIMIT

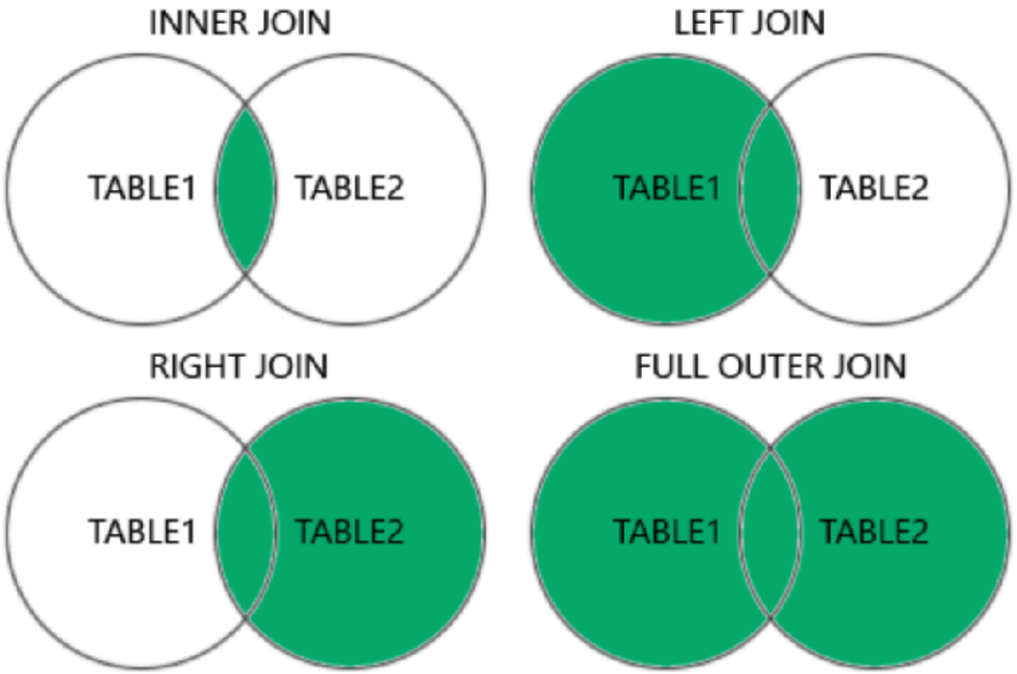


https://www.sqlite.org/lang_select.html

SQL-spørringer *(queries)*

DML - Data manipulation language

- JOIN ... ON
- hva med NULL?
- => inner,outer,left,right



Prøv JOIN på movies.db

noen forslag til spørringer kommer i gruppen

https://www.w3schools.com/sql/sql_join.asp

https://www.sqlitetutorial.net/sqlite-join/

ingen forelesning i morgen

kommende tirsdag:

DB i brukskontekst — ansvar, GDPR, ...