

Inleiding tot databanken

3. Vraagtaalen – Deel 1

Prof. dr. Paolo Pilozzi



Overzicht

3.1 Vraagtafen

3.2 Relationeel Model

3.3 Relationele Algebra

3.4 SQL (Deel 1)

Herhaling Hoorcollege 1

3.1 Vraagtafen

Herhaling Hoorcollege 1
Hoofdstuk 2.2 & 2.3

- * 3-Schema DBMS architectuur
- * Talen en Interfaces

3-S DBMS arch. & Talen

* 3-schema DBMS architectuur:

SDL: Storage Definition Language (intern)

DDL: Data Definition Language (conceptueel)

VDL: View Definition Language (extern)

* Daarnaast:

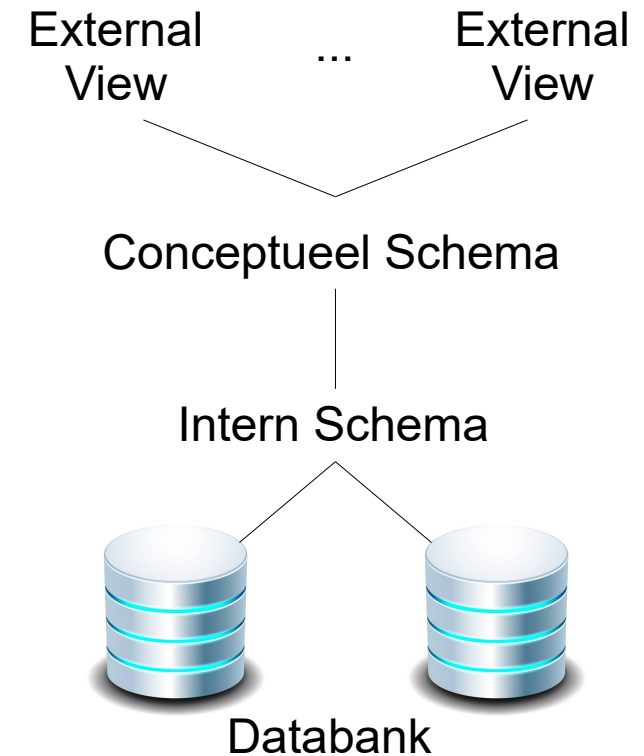
DML: Data Manipulation Language

Vb. ophalen, aanpassen, ...

* Relational DBMS:

SQL (= DDL+VDL+DML)

SQL is een declaratieve taal



(ANSI/SPARC, 1975)

3.2 Relationeel Model

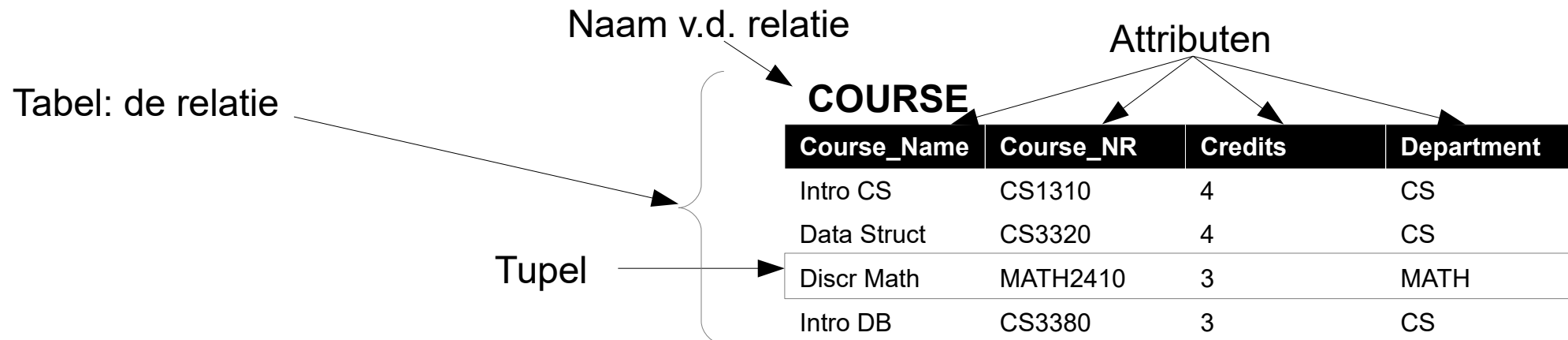
Herhaling Hoorcollege 1
Hoofdstuk 5

- * Concepten
 - Domeinen, attributen, tupels, en relaties
 - Karakteristieken van relaties
 - Notaties
- * Sleutels
- * Integriteit

Relatie

Een **relatie** R op de verzamelingen V_1, \dots, V_n is een deelverzameling van de productverzameling $V_1 \times \dots \times V_n$: $R \subseteq V_1 \times \dots \times V_n$

In **Relationele Model**: Elke rij in de tabel is een **tupel**. De kolomnamen zijn **attributen**. De tabel is een **relatie**. Kolom data types: **domein** van mogelijke waarden voor attributen.



Domein, Attribuut, Tupel en Relatie

- * **Domein D** of **dom** = verzameling atomaire waarden.
- * Een **relatieschema** R , geschreven als $R(A_1, A_2, \dots, A_n)$, bestaat uit een **relatiennaam** R en een lijst van **attributen** A_1, A_2, \dots, A_n .
- * Elk attribuut A_i stelt een **domein** van waarden voor dat deel uitmaakt van R . D is het domein van A_i , geschreven als $\text{dom}(A_i)$. De **ariteit** van R is het aantal attributen, n , in R .

Opgelet: In Hoorcollege 1 maakten we een expliciet verschil tussen het relatie schema (cfr. Rel) en zijn naam. *Vanaf nu niet meer tenzij het niet anders kan!*

Domein, Attribuut, Tupel en Relatie

Tupel over de attributen $X = \{A_1, A_2, \dots, A_n\}$

$t = \{(A_1, w_1), (A_2, w_2), \dots, (A_n, w_n)\}$ met
elke $w_i \in \text{dom}(A_i) \cup \{\text{NULL}\}$

Korte notatie als geen verwarring mogelijk: $\langle w_1, w_2, \dots, w_n \rangle$

Merk op: Attributen zijn niet geordend! (wel in notatie/opslag: lijst)

Relatie r van het relatie schema $R(A_1, A_2, \dots, A_n)$, ook geschreven als $r(R)$,
is een verzameling tupels, $r = \{t_1, t_2, \dots, t_m\}$.

Merk op, ook tuples zijn niet geordend! (wel in notatie/opslag: lijst)

Merk op, een tupel kan maar één keer voorkomen in een relatie!

Notatie v. Relationele Modellen

Een relatie schema R van graad n wordt geschreven als $R(A_1, A_2, \dots, A_n)$.

- * Q, R, S worden gebruikt voor relatienamen.
- * q, r, s worden gebruikt voor relatietoestanden (= verzameling tupels).
 - **De naam, R , van een relatie kan dit ook eenduidig bepalen.**
- * t, u, v worden gebruikt voor tupels.
- * $R.A$ wordt gebruikt voor attribuut A van relatieschema R aan te duiden.
- * Een tupel t in een relatie $r(R)$ wordt geschreven als $\langle w_1, w_2, \dots, w_n \rangle$
waarbij w_i de waarde is die overeenkomt met attribuut A_i .
- * Componenten van een tupel kunnen bekomen worden via:
 - De notatie $t.A_i$ om de waarde w_i in t voor attribuut A_i te bekomen
 - De notatie $t[A_i, A_j, \dots, A_k]$ of $t.(A_i, A_j, \dots, A_k)$ met A_i, A_j, \dots, A_k attributen van R om zo de subtupel $\langle w_i, w_j, \dots, w_k \rangle$ te bekomen.

Sleutels

Relatie schema $R(A_1, A_2, \dots, A_n)$ met attributen $X = \{A_1, A_2, \dots, A_n\}$

- * Super-sleutel $K \subseteq X$
= Verzameling attributen die tupel van $r(R)$ ondubbelzinnig bepalen
- * Kandidaat-sleutel K = Super-sleutel K zonder overbodige attributen
 \Rightarrow Er bestaat geen super-sleutel $K' \subset K$ (of $K' \subseteq K$ met $K' \neq K$)
- * Primaire sleutel, PK (Primary Key), is kandidaatsleutel en wordt onderstreept

Een verzameling attributen FK (Foreign Key) van R_1 , of $R_1(A_1, A_2, \dots, A_n)$, is een **verwijssleutel** a.s.a.

- * Attributen van FK hebben zelfde domein als primaire sleutel attributen PK van R_2
- * Elke waarde van FK in R_1 komt voor als waarde van een tupel in R_2 of is NULL
 $\Rightarrow \forall t_1 \in R_1: t_1[\text{FK}]$ is NULL of $\exists t_2 \in R_2: t_1[\text{FK}] = t_2[\text{PK}]$

Bepeningen

Domeinrestricties

- Bepeningen op de mogelijke waarden van een attribuut
- vb. Leeftijd van een mens als postieve integer < 150

Sleutelrestricties

- Primaire sleutels dienen uniek te zijn en mogen niet NULL zijn

Algemene integriteitsrestricties

= Juistheid en volledigheid van een databank

- Statische regels

Slaan op databank toestand (elke mogelijke toestand)

- Dynamische regels

Slaan op toestandsovergangen (elke mogelijke transitie)

Overzicht

3.1 Vraagtafen

3.2 Relationeel Model

3.3 Relationele Algebra

3.4 SQL (Deel 1)

Herhaling Hoorcollege 1

3.3 Relationele Algebra

Relationele Algebra (RA) is de verzameling basisoperaties die samengaan met het Relationele Model (RM)

Hoofdstuk 6 – Oefenzitting 2

- * Inleiding
- * Unaire operaties: SELECT, PROJECT
- * Sequenties van operaties & Hernoemingen: RENAME
- * Binaire operaties
 - Verzamelingenleer: UNION, INTERSECTION, MINUS, CARTESISCH PRODUCT
 - Join operatoren: THETA-JOIN, EQUI-JOIN, NATURAL JOIN
- * Fundamentele operatoren
- * Bijkomend: Aggregaatfuncties & Groeperingen, Recursieve sluiting, OUTER JOIN

Inleiding

Algebraïsche talen

- Steunt op Relationele Algebra
- Operatoren over relaties
- Proceduraal (Hoe)

Calculus talen

- Steunt op Relationele Calculus
- Formele beschrijving:
Predikatenlogica
- Declaratief (Wat)

Queryverwerking
-en optimalisatie

Declarativiteit

Praktisch

SQL (Structured Query Language)

- Vooral declaraties (Wat)
- Alle bewerkingen op databanken

Inleiding - Voorbeeld

Figure 5.7
Referential integrity constraints displayed on the COMPANY relational database schema.

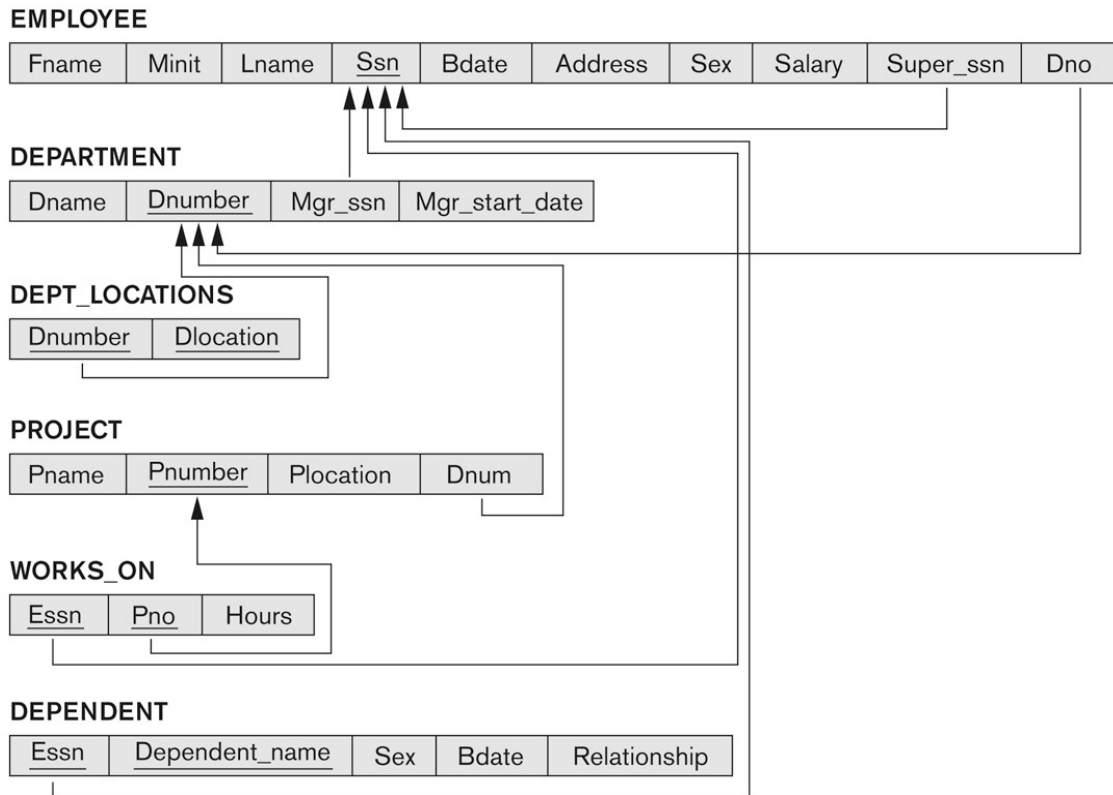


Figure 5.6
One possible database state for the COMPANY relational database schema.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

Unaire operaties: SELECT

Selecteert een subset van tupels uit een relatie, R, op basis van een selectie criterium, F:

$$\sigma_F(R) = \{t \mid t \in R \text{ en } F \text{ onder substitutie van } t \text{ is waar}\}$$

- * Beeldt relatie af op relatie met hetzelfde schema (zelfde attributen)
=> Resultaat is van dezelfde graad.
- * Op één relatie toegepast en selectie criterium op elk tupel individueel
=> Unair operatie
- * De resulterende relatie bevat nooit meer tupels dan de oorspronkelijke!
=> $\sigma_F(R) \subseteq R$ en kardinaliteit stijgt niet $\#\sigma_F(R) \leq \#R$

Unaire operaties: SELECT

Selectiecriteria is een booleaanse vergelijking met clauses van de vorm:

$\langle A_i \rangle \langle \text{vergelijkingsoperator} \rangle \langle C^{te} / A_j \rangle$; met

- * A_i, A_j : namen van een attribuut in R (A_i, A_j zelfde domein);
- * C^{te} : constante van het domein van het attribuut, A_i ; en
- * vergelijkingsoperatoren: =, \neq , <, >, \leq , \geq .

en connectoren: \wedge (AND), \vee (OR), \neg (NOT)

- * =, \neq , <, >, \leq , \geq voor (volledig) geordende domeinen, anders enkel =, \neq
- * Bijkomende vergelijkingsoperatoren: SUB_STRING

Unaire operaties: SELECT

Enkele voorbeelden:

* $\sigma_{Dno=4}$ (EMPLOYEE)

EMPLOYEE									
Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Unaire operaties: SELECT

Enkele voorbeelden:

* $\sigma_{\text{Salary} > 30000}(\text{EMPLOYEE})$

EMPLOYEE									
Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Unaire operaties: SELECT

Enkele voorbeelden:

* $\sigma_{(Dno=4 \wedge Salary>25000) \vee (Dno=5 \wedge Salary>30000)}$ (EMPLOYEE)

EMPLOYEE									
Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Unaire operaties: SELECT

Eigenschappen/Opmmerkingen:

- * Selectieoperator is commutatief:

$$\sigma_{F_1}(\sigma_{F_2}(R)) = \sigma_{F_2}(\sigma_{F_1}(R))$$

- * Selectieoperatoren kunnen samengesteld worden via criterium:

$$\sigma_{F_1}(\sigma_{F_2}(\dots(\sigma_{F_n}(R))\dots)) = \sigma_{F_1 \wedge F_2 \wedge \dots \wedge F_n}(R)$$

- * Hoe vergelijkingsoperatoren $<$, $>$, \leq , \geq behandelen?

- Volledige ordening: nummers, karakter strings (alfabetisch), datums
- Partiële ordening: matrices (of lijsten); als niet gedefiniëerd: onwaar
- Ongeordend: kleuren (e.g., {rood, groen, blauw}); altijd onwaar

Unaire operaties: PROJECT

Selecteert een subset van attributen, X , uit een relatie, R :

$$\pi_X(R) = \{t \mid \exists t' \in R: t'.X = t\}$$

- * Beeldt relatie af op relatie met verminderd schema (minder attributen)
=> Resultaat is van verminderde graad.
- * Op één relatie toegepast zonder selectie criterium
=> Unaire operatie
- * De resulterende relatie bevat nooit meer tupels dan de oorspronkelijke!
=> Kardinaliteit stijgt niet $\#\pi_X(R) \leq \#R$ (geen dubbels want verzameling)

Unaire operaties: PROJECT

Eigenschappen/Opmerkingen:

* Projectieoperator is NIET commutatief:

$$\pi_X(\pi_Y(R)) \text{ enkel gedefinieerd als } X \subseteq Y$$

* Projectieoperator is idempotent:

$$\pi_X(\pi_X(R)) = \pi_X(R)$$

* Enkel buitenste projectieoperator:

$$\pi_{X_1}(\pi_{X_2}(\dots \pi_{X_n}(R)\dots)) = \pi_{X_1}(R)$$

Unaire operaties: PROJECT

Enkele voorbeelden:

$$\pi_{Lname, Fname, Salary}(\text{EMPLOYEE})$$

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1



Lname	Fname	Salary
Smith	John	30000
Wong	Franklin	40000
Zelaya	Alicia	25000
Wallace	Jennifer	43000
Narayan	Ramesh	38000
English	Joyce	25000
Jabbar	Ahmad	25000
Borg	James	55000

Unaire operaties: PROJECT

Enkele voorbeelden:

$$\pi_{\text{Sex,Salary}}(\text{EMPLOYEE})$$

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1



Sex	Salary
M	30000
M	40000
F	25000
F	43000
M	38000
M	25000
M	55000

Sequenties v. operaties

Operaties kunnen samengesteld worden, vb.:

$$\Pi_{Fname, Lname, Salary}(\sigma_{Dno=5}(EMPLOYEES))$$

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Fname	Lname	Salary
John	Smith	30000
Franklin	Wong	40000
Ramesh	Narayan	38000
Joyce	English	25000

Sequenties v. operaties

Operaties kunnen samengesteld worden, vb.:

$$\pi_{Fname,Lname,Salary}(\sigma_{Dno=5}(EMPLOYEES))$$

Expliciet benoemen van tussenresultaten:

$$EMP_D5 \leftarrow \sigma_{Dno=5}(EMPLOYEES)$$

$$RESULT \leftarrow \pi_{Fname,Lname,Salary}(EMP_D5)$$

SELECT en PROJECT commuteren:

$$\pi_X(\sigma_F(R)) = \sigma_F(\pi_X(R)) \text{ als F enkel attributen in X gebruikt}$$

$$\text{vb. } \pi_{Fname,Lname,Salary}(\sigma_{Salary>30000}(EMPLOYEE)) = \sigma_{Salary>30000}(\pi_{Fname,Lname,Salary}(EMPLOYEE))$$

Hernoeming

Doel: Wijziging van attribuutnamen

$$\text{EMP_D5} \leftarrow \sigma_{\text{Dno}=5}(\text{EMPLOYEES})$$

$$\text{RESULT}(\text{First_name}, \text{Last_name}, \text{Salary}) \leftarrow \pi_{\text{Fname}, \text{Lname}, \text{Salary}}(\text{EMP_D5})$$

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

RESULT

First_name	Last_name	Salary
John	Smith	30000
Franklin	Wong	40000
Ramesh	Narayan	38000
Joyce	English	25000

Hernoeming

Doel: Wijziging van attribuutnamen

$$\text{EMP_D5} \leftarrow \sigma_{\text{Dno}=5}(\text{EMPLOYEES})$$

$$\text{RESULT}(\text{First_name}, \text{Last_name}, \text{Salary}) \leftarrow \pi_{\text{Fname}, \text{Lname}, \text{Salary}}(\text{EMP_D5})$$

Kan ook via **unaire operator RENAME**, ρ .

Toegepast op een relatie, R , van graad n , met attributen A_1, A_2, \dots, A_n :

$$\rho_{S(B_1, B_2, \dots, B_n)}(R)$$

hernoem R naar S en A_1, A_2, \dots, A_n naar B_1, B_2, \dots, B_n

$$\rho_S(R)$$

hernoem R naar S

$$\rho_{(B_1, B_2, \dots, B_n)}(R)$$

hernoem A_1, A_2, \dots, A_n naar B_1, B_2, \dots, B_n

Operaties uit Verzamelingenleer

UNION, INTERSECTION en MINUS (\Rightarrow Binaire operaties)

* Enkel op vergelijkbare relaties (met zelfde type tupels) = unie compatibel

R en S zijn **unie compatibel** a.s.a. voor $R(A_1, A_2, \dots, A_n)$ en $S(B_1, B_2, \dots, B_m)$ geldt dat $n = m$ en $\forall i \in [1, n]: \text{dom}(A_i) = \text{dom}(B_i)$.

- UNION: $R \cup S$ behoudt tupels in R of S of in beide (zonder dubbels)
- INTERSECTION: $R \cap S$ behoudt tupels zowel in R als in S
- MINUS: $R - S$ behoudt tupels in R maar niet in S

* Operatie is "schemabehoudend" (Behoudt attribuutnamen van 1^{ste} relatie)

Operaties uit Verzamelingenleer

Voorbeeld:

- a) De unie compatibele relaties
- b) $STUDENT \cup INSTRUCTOR$
- c) $STUDENT \cap INSTRUCTOR$
- d) $STUDENT - INSTRUCTOR$
- e) $INSTRUCTOR - STUDENT$

Eigenschappen:

- * UNION en INTERSECTION zijn commutatief. MINUS niet.
- * $R \cap S = R \cup S - ((R - S) \cup (S - R))$

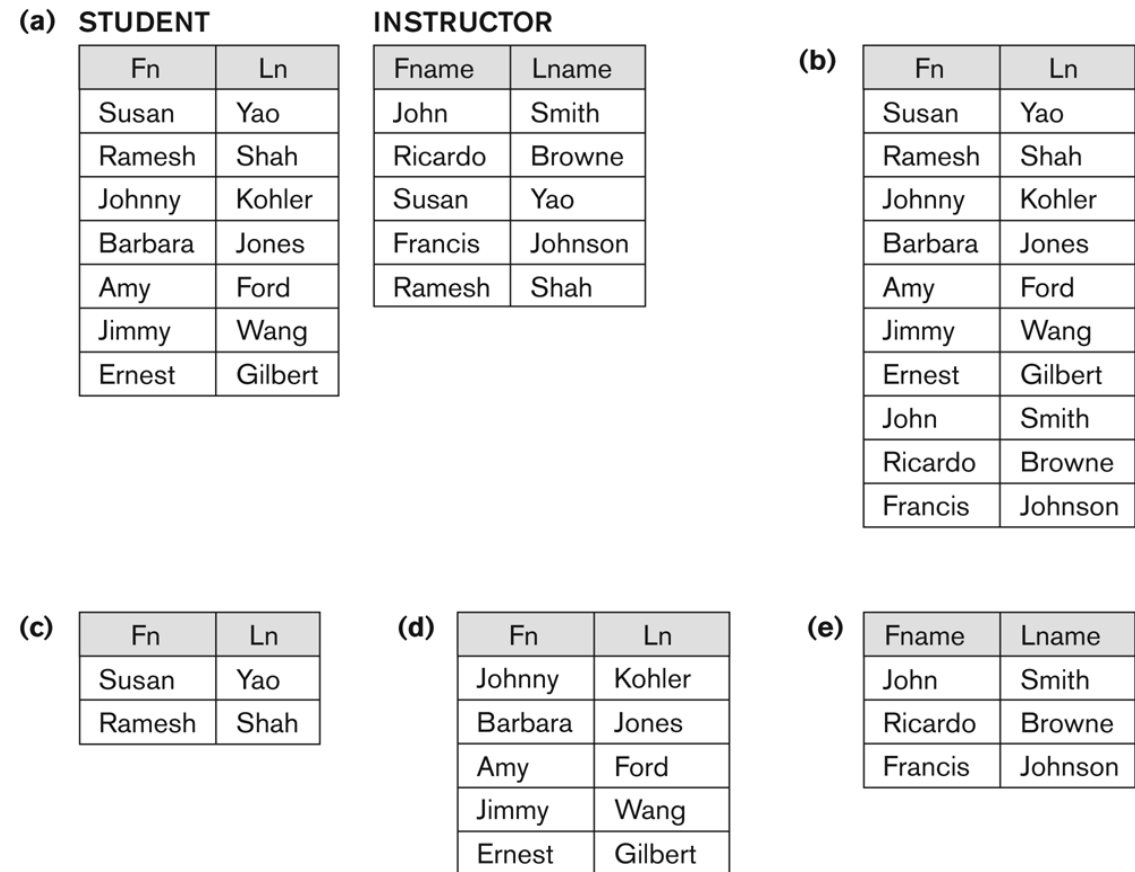


Figure 6.4

The set operations UNION, INTERSECTION, and MINUS. (a) Two union-compatible relations. (b) $STUDENT \cup INSTRUCTOR$. (c) $STUDENT \cap INSTRUCTOR$. (d) $STUDENT - INSTRUCTOR$. (e) $INSTRUCTOR - STUDENT$.

Operaties uit Verzamelingenleer

CARTESISCH PRODUCT (\Rightarrow Binaire operatie)

Voor relaties R en S (niet noodzakelijk unie compatibel), met $R(A_1, A_2, \dots, A_n)$ en $S(B_1, B_2, \dots, B_m)$ als relatieschema's resp., is het cartesisch product van R en S , geschreven als $R \times S$, de relatie Q , met $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$ als relatieschema, zodat elke combinatie van tupels uit R en S in Q zit.

\Rightarrow Als n_R de kardinaliteit v. R is ($n_R = |R| = \#R$) en n_S die v. S , dan is $n_R * n_S$ die v. Q .

\Rightarrow Het n -ary carthesisch product is bij extensie dat over n onderliggende relaties.

Operaties uit Verzamelingenleer

Voorbeeld (Cartesisch product):

Lijst met namen van personen ten laste van vrouwelijke werknemers.

$$F_EMP \leftarrow \sigma_{Sex = 'F'}(EMPLOYEE)$$
$$EMP_N \leftarrow \pi_{Fname, Lname, Ssn}(F_EMP)$$
$$EMP_D \leftarrow EMP_N \times DEPENDENT$$
$$T_RES \leftarrow \sigma_{Ssn = Essn}(EMP_D)$$
$$RESULT \leftarrow \pi_{Fname, Lname, Dependent_name}(T_RES)$$

=> { Niet noodzakelijk
Maar wat weg kan zo snel mogelijk weg.

Operaties uit Verzamelingenleer

Voorbeeld (Cartesisch product):

Lijst met namen van personen
ten laste van vrouwelijke werknemers.
Met hernoeming van tussenresultaten.

$FEMALE_EMPS \leftarrow \sigma_{Sex = 'F'}(EMPLOYEE)$
 $EMP_NAMES \leftarrow \pi_{Fname, Lname, Ssn}(FEMALE_EMPS)$
 $EMP_DEPENDENTS \leftarrow EMP_NAMES \times DEPENDENT$
 $ACTUAL_DEPENDENTS \leftarrow \sigma_{Ssn = Essn}(EMP_DEPENDENTS)$
 $RESULT \leftarrow \pi_{Fname, Lname, Dependent_name}(ACTUAL_DEPENDENTS)$

Figure 6.5
The CARTESIAN PRODUCT (CROSS PRODUCT) operation.

FEMALE_EMPS

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
Alicia	J	Zelaya	999887777	1968-07-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5

EMP_NAMES

Fname	Lname	Ssn
Alicia	Zelaya	999887777
Jennifer	Wallace	987654321
Joyce	English	453453453

EMP_DEPENDENTS

Fname	Lname	Ssn	Essn	Dependent_name	Sex	Bdate	...
Alicia	Zelaya	999887777	333445555	Alice	F	1986-04-05	...
Alicia	Zelaya	999887777	333445555	Theodore	M	1983-10-25	...
Alicia	Zelaya	999887777	333445555	Joy	F	1958-05-03	...
Alicia	Zelaya	999887777	987654321	Abner	M	1942-02-28	...
Alicia	Zelaya	999887777	123456789	Michael	M	1988-01-04	...
Alicia	Zelaya	999887777	123456789	Alice	F	1988-12-30	...
Alicia	Zelaya	999887777	123456789	Elizabeth	F	1967-05-05	...
Jennifer	Wallace	987654321	333445555	Alice	F	1986-04-05	...
Jennifer	Wallace	987654321	333445555	Theodore	M	1983-10-25	...
Jennifer	Wallace	987654321	333445555	Joy	F	1958-05-03	...
Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28	...
Jennifer	Wallace	987654321	123456789	Michael	M	1988-01-04	...
Jennifer	Wallace	987654321	123456789	Alice	F	1988-12-30	...
Jennifer	Wallace	987654321	123456789	Elizabeth	F	1967-05-05	...
Joyce	English	453453453	333445555	Alice	F	1986-04-05	...
Joyce	English	453453453	333445555	Theodore	M	1983-10-25	...
Joyce	English	453453453	333445555	Joy	F	1958-05-03	...
Joyce	English	453453453	987654321	Abner	M	1942-02-28	...
Joyce	English	453453453	123456789	Michael	M	1988-01-04	...
Joyce	English	453453453	123456789	Alice	F	1988-12-30	...
Joyce	English	453453453	123456789	Elizabeth	F	1967-05-05	...

ACTUAL_DEPENDENTS

Fname	Lname	Ssn	Essn	Dependent_name	Sex	Bdate	...
Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28	...

RESULT

Fname	Lname	Dependent_name
Jennifer	Wallace	Abner

Join operatoren

JOIN combineert gerelateerde tupels uit 2 relaties in 1 tuple:

$$R \bowtie S$$

=> Cartesisch product + selectie (=> binaire operatie)

Vb. Het volgende:

$EMP_D \leftarrow EMP_N \times DEPENDENT$

$T_RES \leftarrow \sigma_{Ssn = Essn}(EMP_D)$

kan met een JOIN:

$T_RES \leftarrow EMP_N \bowtie_{Ssn = Essn} DEPENDENT$

Join operatoren - Voorbeeld

Combineer voor alle departementen de informatie over het departement en die over hun manager:

DEPARTMENT ⋈_{Mgr_ssn=Ssn} EMPLOYEE

DEPT_MGR

Dname	Dnumber	Mgr_ssn	...	Fname	Minit	Lname	Ssn	...
Research	5	333445555	...	Franklin	T	Wong	333445555	...
Administration	4	987654321	...	Jennifer	S	Wallace	987654321	...
Headquarters	1	888665555	...	James	E	Borg	888665555	...

Figure 6.6

Result of the JOIN operation

Join operatoren

THETA-JOIN is de meest algemene vorm: $R \bowtie_F S$

$F = C_1 \wedge C_2 \wedge \dots \wedge C_n$ met elke C_k van de vorm $A_i \theta B_j$ waarbij

$\theta \in \{ =, <, >, \leq, \geq, \neq \}$ en $\text{dom}(A_i) = \text{dom}(B_j)$

=> Connectoren \vee (OR) en \neg (NOT) niet toegelaten!

=> M.a.w. $\sigma_F(R \times S)$ is expressiever dan $R \bowtie_F S$, maar niet algemener!

=> Merk op dat als join attributen NULL waardes bevatten: onwaar!

=> Merk op dat als θ niet gedefiniëerd is: onwaar!

EQUI-JOIN is de meestgebruikte vorm:

$F = C_1 \wedge C_2 \wedge \dots \wedge C_n$ met elke C_k van de vorm $A_i = B_j$ en $\text{dom}(A_i) = \text{dom}(B_j)$

Join operatoren

NATURAL JOIN is een EQUI-JOIN, geschreven als:

$$R * S$$

waarbij een impliciete $F = C_1 \wedge C_2 \wedge \dots \wedge C_n$ verondersteld wordt met elke C_k van de vorm $A_i = B_j$, $A = B$ en $\text{dom}(A_i) = \text{dom}(B_j)$

=> Stel alle attributen (= join attributen) met zelfde naam uit R en S gelijk

=> Voor de join attributen er maar één (en niet beide)
in het resultaat weergegeven!!

=> Een voorgaande stap is (vaak) hernoemen van attributen opdat zij die moeten overeenkomen, join attributen worden.

Join operatoren

Opgelet: Bij cartesisch product en (eerdere) joins kan het ook gebeuren dat attributen met dezelfde naam al reeds voorkomen in beide relaties!
Dit werd niet beperkt!

Dit is een probleem aangezien we in de resulterende relatie niet langer een verzameling attributen bekomen, wat niet gedefiniëerd is! Dat, of door het verwijderen van dubbele attributen informatie in tupels wegslijten!

=> Hernoemen is de oplossing om het formeel juist te houden.

=> Dit probleem kan vermeden worden door ervoor te zorgen dat attribuutnamen over relaties heen al reeds uniek zijn.

Join operatoren – Voorbeeld

Combineer alle projecten met hun departementen:

$$\text{PROJ_DEPT} \leftarrow \text{PROJECT} * \rho_{(\text{Dname}, \text{Dnum}, \text{Mgr_ssn}, \text{Mgr_start_date})}(\text{DEPARTMENT})$$

Of in twee stappen:

$$\text{DEPT} \leftarrow \rho_{(\text{Dname}, \text{Dnum}, \text{Mgr_ssn}, \text{Mgr_start_date})}(\text{DEPARTMENT})$$

$$\text{PROJ_DEPT} \leftarrow \text{PROJECT} * \text{DEPT}$$

PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

PROJ_DEPT

Pname	<u>Pnumber</u>	Plocation	Dnum	Dname	Mgr_ssn	Mgr_start_date
ProductX	1	Bellaire	5	Research	333445555	1988-05-22
ProductY	2	Sugarland	5	Research	333445555	1988-05-22
ProductZ	3	Houston	5	Research	333445555	1988-05-22
Computerization	10	Stafford	4	Administration	987654321	1995-01-01
Reorganization	20	Houston	1	Headquarters	888665555	1981-06-19
Newbenefits	30	Stafford	4	Administration	987654321	1995-01-01

Join operatoren – Voorbeeld

Combineer alle departementen met hun locaties:

$DEPT_LOCS \leftarrow DEPARTMENT * DEPT_LOCATIONS$

DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

DEPT_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

DEPT_LOCS

Dname	Dnumber	Mgr_ssn	Mgr_start_date	Location
Headquarters	1	888665555	1981-06-19	Houston
Administration	4	987654321	1995-01-01	Stafford
Research	5	333445555	1988-05-22	Bellaire
Research	5	333445555	1988-05-22	Sugarland
Research	5	333445555	1988-05-22	Houston

Fundamentele operatoren

De fundamentele operatoren in relationele algebra zijn:

$$\{\sigma, \pi, \rho, \cup, -, \times\} \quad (\text{en met verfijning } \{\sigma, \pi, \cup, -, \times\})$$

=> SELECT, PROJECT, RENAME, UNION, MINUS, CART.PROD.

Want de andere operatoren kunnen ervan afgeleid worden:

$$R \cap S \equiv R \cup S - ((R - S) \cup (S - R))$$

$$R \bowtie_F S \equiv \sigma_F(R \times S) \quad (\text{Merk op: } F \text{ enkel } \wedge, \text{ maar wetten van De Morgan en } \theta \in \{=, <, >, \leq, \geq, \neq\})$$

* Ze bestaan omdat ze handig zijn, alsook de speciale gevallen van JOIN operaties (met NATURAL JOIN die een RENAME in rekening brengt).

* RENAME kan via PROJECT: maar werd niet exact gedefiniëerd

Bijkomend operaties

Eerdere operatoren schieten soms tekort.

=> Bijkomende operaties:

- * Veralgemeende projectie
- * Aggregaatfuncties en groeperingen
- * Recursieve Sluiting
- * OUTER JOIN (vs. De eerdere JOIN: INNER JOIN)

om expressief vermogen relationele algebra te verbeteren

Veralgemeende projectie

Veralgemeende projectie laat toe om functies te definiëren over geselecteerde attributen: $\pi_{f_1, f_2, \dots, f_n}(R)$;

met f_1, f_2, \dots, f_n functies over de attributen van R en constante waarden.

Doel is bij voorbeeld Rapportage

Gegeven: EMPLOYEE(Ssn, Salary, Deduction, Years_service); maak rapport:

Net_salary = Salary – Deduction Bonus = 2000 * Years_service Tax = 0,25 * Salary

Dat bekomen kan worden via:

$$\text{REPORT}(\text{Ssn}, \text{Net_salary}, \text{Bonus}, \text{Tax}) \leftarrow \pi_{\text{Ssn}, \text{Salary} - \text{Deduction}, 2000 * \text{Years_service}, 0.25 * \text{Salary}}(\text{EMPLOYEE})$$

$$\text{REPORT} \leftarrow \rho_{(\text{Ssn}, \text{Net_salary}, \text{Bonus}, \text{Tax})}(\pi_{\text{Ssn}, \text{Salary} - \text{Deduction}, 2000 * \text{Years_service}, 0.25 * \text{Salary}}(\text{EMPLOYEE}))$$

Aggregaatfuncties & groeperingen

Laten toe om wiskundige aggregaatfuncties te specificeren over collecties van waarden in een databank.

Wordt geschreven als: $\langle \text{groepering} \rangle \mathcal{F}_{\langle \text{functies} \rangle} (R)$

Hier is $\langle \text{groepering} \rangle$ een lijst attributen van R (waarover we gegroepeerd aggregeren \Rightarrow elke combinatie van waarden van attributen in groepering worden samengenomen), en $\langle \text{functies} \rangle$ een lijst van koppels, van de vorm: $\langle \text{functie} \rangle \langle \text{attribuut} \rangle$; met functie SUM, AVERAGE, MAX, MIN, of COUNT en attribuut in R , dat de aggregatie uitdrukt per groepering.

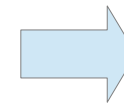
Aggregaatfuncties & groeperingen

Voorbeeld: Gem. inkomen werknemers gegroepeerd per dep.nr.:

Dno \int AVERAGE Salary (EMPLOYEE)

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1



Dno	AVERAGE_salary
5	33250
4	31000
1	55000

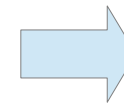
Aggregaatfuncties & groeperingen

Voorbeeld: Gem. inkomen werknemers, gegroepeerd per dep.nr. en geslacht:

Dno, Sex \int AVERAGE Salary (EMPLOYEE)

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1



Dno	Sex	AVERAGE_salary
5	F	25000
5	M	36000
4	F	34000
4	M	25000
1	M	55000

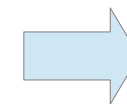
Aggregaatfuncties & groeperingen

Voorbeeld: Aantal werknemers en gem. inkomen, over volledige relatie:

\int COUNT Ssn, AVERAGE Salary (EMPLOYEE)

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1



COUNT_ssn	AVERAGE_salary
8	35125

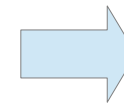
Aggregaatfuncties & groeperingen

Voorbeeld: Hernoemd aantal werknemers en gem. inkomen, gegroepeerd per dep.nr.:

$\rho_{(Dno, Nr_emps, Avg_sal)} \left(\int_{Dno} \text{COUNT Ssn, AVERAGE Salary} (\text{EMPLOYEE}) \right)$

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1



Dno	Nr_emps	Avg_sal
5	4	33250
4	3	31000
1	1	55000

Recursieve sluiting

Voor recursieve relaties tussen tupels van hetzelfde type.

= Transitieve sluiting

Voorbeeld: *Overste-Ondergeschikte*

* Om alle oversten/ondergeschikten van een werknemer te bekomen.

Dit kan in relationele algebra voor een specifiek niveau:

- vb. ouders, grootouders, etc.

Voor alle niveaus in de limiet => oneindige uitdrukking

Recursieve sluiting – Voorbeeld

$BORG_SSN \leftarrow \pi_{Ssn}(\sigma_{Fname='James' \wedge Lname='Borg'}(EMPLOYEE))$
 $SUPERVISION(Ssn1, Ssn2) \leftarrow \pi_{Ssn, Super_ssn}(EMPLOYEE)$

Eerste Niveau:

$RESULT1(Ssn) \leftarrow \pi_{Ssn1}(SUPERVISION \bowtie_{Ssn2=Ssn} BORG_SSN)$

Tweede niveau:

$RESULT2 \leftarrow \pi_{Ssn1}(SUPERVISION \bowtie_{Ssn2=Ssn} RESULT1)$
 $RESULT \leftarrow RESULT1 \cup RESULT2$

Verdere niveaus:

$RESULT<N> \leftarrow \pi_{Ssn1}(SUPERVISION \bowtie_{Ssn2=Ssn} RESULT<N-1>)$
 $RESULT \leftarrow RESULT1 \cup RESULT2 \cup \dots \cup RESULT<N>$

Recursieve sluiting herhaalt dit principe tot RESULT<N> een lege relatie is.

SUPERVISION

(Borg's Ssn is 888665555)
(Ssn) (Super_ssn)

Ssn1	Ssn2
123456789	333445555
333445555	888665555
999887777	987654321
987654321	888665555
666884444	333445555
453453453	333445555
987987987	987654321
888665555	null

RESULT1

Ssn
333445555
987654321

(Supervised by Borg)

RESULT2

Ssn
123456789
999887777
666884444
453453453
987987987

(Supervised by Borg's subordinates)

RESULT

Ssn
123456789
999887777
666884444
453453453
987987987
333445555
987654321

(RESULT1 \cup RESULT2) 51

OUTER JOIN

Staat in contrast met INNER JOIN ($R \bowtie_F S$)

=> D.w.z. de joins die eerder gezien werden.

Het probleem: INNER JOIN geeft geen info over tupels die niet aan de JOIN conditie voldoen of NULL waarden in relevante attributen hebben. Die worden niet geselecteerd.

* Vb. Werknemers die een dept. leiden: geen info over zij die er geen leiden.

Doel: Rapportage in de praktijk wenst die info wel, en OUTER JOINS maken dit mogelijk.

OUTER JOIN

Drie soorten OUTER JOINS:

- * LEFT OUTER JOIN: $R \bowtie_F S$; behoudt alle tupels in R
- * RIGHT OUTER JOIN: $R \bowtie_F S$; behoudt alle tupels in S
- * FULL OUTER JOIN: $R \bowtie_F S$; behoudt alle tupels in R en S

Voorbeeld: Genereer de namen van werknemers en de naam van het departement als ze dat leiden. (Indien niet: NULL).

```
TEMP ← EMPLOYEE ⋈Ssn=Mgr_ssn DEPARTMENT
RESULT ← πFname,Lname,Dname(TEMP)
```

*FULL JOIN hier zelfde resultaat als LEFT JOIN,
RIGHT JOIN hier vergelijkbaar resultaat als INNER JOIN*

Fname	Lname	Dname
John	Smith	NULL
Franklin	Wong	Research
Alicia	Zelaya	NULL
Jennifer	Wallace	Administration
Ramesh	Narayan	NULL
Joyce	English	NULL
Ahmad	Jabbar	NULL
James	Borg	Headquarters

Queries in Relationele Algebra (Hfdst 6.5)

Bekom de naam en het adres van alle werknemers die voor het "Research" departement werken.

$$\pi_{Fname, Lname, Address}(\sigma_{Dname = 'Research'}(DEPARTMENT \bowtie_{Dnumber = Dno} EMPLOYEE))$$

Of:

$$RESEARCH_DEPT \leftarrow \sigma_{Dname = 'Research'}(DEPARTMENT)$$

$$RESEARCH_EMPS \leftarrow (RESEARCH_DEPT \bowtie_{Dnumber = Dno} EMPLOYEE)$$

$$RESULT \leftarrow \pi_{Fname, Lname, Address}(RESEARCH_EMPS)$$

Of: andere volgorde JOIN en SELECT, of met NATURAL JOIN na hernoeming, ...

PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

WORKS_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

DEPARTMENT

<u>Dname</u>	<u>Dnumber</u>	<u>Mgr_ssn</u>	<u>Mgr_start_date</u>
--------------	----------------	----------------	-----------------------

DEPT_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

Queries in Relationele Algebra (Hfdst 6.5)

Vindt de namen van alle werknemers die meer dan twee (≥ 2) personen ten laste hebben.

$$\begin{aligned} \text{DEPS_PER_SSN}(\text{Ssn}, \text{No_of_deps}) &\leftarrow \text{Essn} \int_{\text{COUNT Dependent_name}} (\text{DEPENDENT}) \\ \text{MT2_DEPS_PER_SSN} &\leftarrow \sigma_{\text{No_of_deps} \geq 2} (\text{DEPS_PER_SSN}) \\ \text{RESULT} &\leftarrow \pi_{\text{Lname}, \text{Fname}} (\text{MT2_DEPS_PER_SSN} * \text{EMPLOYEE}) \end{aligned}$$

PROJECT

Pname	Pnumber	Plocation	Dnum
-------	---------	-----------	------

WORKS_ON

Essn	Pno	Hours
------	-----	-------

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
------	----------------	-----	-------	--------------

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	-----	-------	---------	-----	--------	-----------	-----

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
-------	---------	---------	----------------

DEPT_LOCATIONS

Dnumber	Dlocation
---------	-----------

Queries in Relationele Algebra (Hfdst 6.5)

Vindt de namen van alle werknemers zonder personen ten laste.

$$\text{ALL_EMP_SSNS} \leftarrow \pi_{\text{Ssn}}(\text{EMPLOYEE})$$

$$\text{ALL_EMP_WDEPS_SSNS}(\text{Ssn}) \leftarrow \pi_{\text{Essn}}(\text{DEPENDENT})$$

$$\text{ALL_EMP_NODEPS_SSNS} \leftarrow \text{ALL_EMP_SSNS} - \text{ALL_EMP_WDEPS_SSNS}$$

$$\text{RESULT} \leftarrow \pi_{\text{Lname, Fname}}(\text{ALL_EMP_NODEPS_SSNS} * \text{EMPLOYEE})$$

PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

WORKS_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

EMPLOYEE

<u>Fname</u>	<u>Minit</u>	<u>Lname</u>	<u>Ssn</u>	<u>Bdate</u>	<u>Address</u>	<u>Sex</u>	<u>Salary</u>	<u>Super_ssn</u>	<u>Dno</u>
--------------	--------------	--------------	------------	--------------	----------------	------------	---------------	------------------	------------

DEPARTMENT

<u>Dname</u>	<u>Dnumber</u>	<u>Mgr_ssn</u>	<u>Mgr_start_date</u>
--------------	----------------	----------------	-----------------------

DEPT_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

Queries in Relationele Algebra (Hfdst 6.5)

Vindt de namen van managers met personen ten laste.

$$\begin{aligned} \text{MGR_SSNS} &\leftarrow \pi_{\text{Mgr_Ssn}}(\text{DEPARTMENT}) \\ \text{ALL_EMP_WDEPS_SSNS}(\text{Ssn}) &\leftarrow \pi_{\text{Essn}}(\text{DEPENDENT}) \\ \text{ALL_MGR_WDEPS_SSNS} &\leftarrow \text{ALL_EMP_WDEPS_SSNS} \cap \text{MGR_SSNS} \\ \text{RESULT} &\leftarrow \pi_{\text{Lname, Fname}}(\text{ALL_EMP_NODEPS_SSNS} * \text{EMPLOYEE}) \end{aligned}$$

PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

WORKS_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

DEPARTMENT

<u>Dname</u>	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
--------------	----------------	---------	----------------

DEPT_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

Queries in Relationele Algebra (Hfdst 6.5)

Vindt de namen van alle werknemers die op alle projecten van departement 5 werken.

$$\text{DEPT5_PROJS}(Pno) \leftarrow \pi_{Pnumber}(\sigma_{Dnum=5}(\text{PROJECT}))$$

$$\text{EMP_PROJS}(Ssn, EPno) \leftarrow \pi_{Essn, Pno}(\text{WORKS_ON})$$

$$\text{SSNS_ON_PROJS} \leftarrow \pi_{Ssn}(\text{EMP_PROJS})$$

$$\text{ALLCOMB_SSNS_DEPT5_PS} \leftarrow \text{DEPT5_PROJ} \times \text{SSNS_ON_PROJS}$$

$$\text{SSNS_NOTALL_DEPT5_PS} \leftarrow \pi_{Ssn}(\text{ALLCOMB_SSNS_DEPT5_PS} - \text{EMP_PROJS})$$

$$\text{SSNS_ALL_DEPT5_PS} \leftarrow \text{SSNS_ON_PROJS} - \text{SSNS_NOTALL_DEPT5_PS}$$

$$\text{RESULT} \leftarrow \pi_{Fname, Lname}(\text{SSNS_ALL_DEPT5_PS} * \text{EMPLOYEE})$$

PROJECT

Pname	Pnumber	Plocation	Dnum
-------	---------	-----------	------

WORKS_ON

Essn	Pno	Hours
------	-----	-------

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
------	----------------	-----	-------	--------------

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	-----	-------	---------	-----	--------	-----------	-----

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
-------	---------	---------	----------------

DEPT_LOCATIONS

Dnumber	Dlocation
---------	-----------

3.3 SQL (Structured Query Language)

SQL is de standaard taal voor manipulatie in RDBMSs en heeft een groot aandeel in het succes van relationele databanken.

Hoofdstuk 8 – Oefenzitting 2/3

HC3 (Deel 1):

- * Inleiding
- * SQL als vraagtaal (DML): basis

HC4 (Deel 2):

- * SQL als vraagtaal (DML): geavanceerd
- * SQL als DDL en VDL

Inleiding

Algebraïsche talen

- Steunt op Relationele Algebra
- Operatoren over relaties
- Proceduraal (Hoe)

Calculus talen

- Steunt op Relationele Calculus
- Formele beschrijving:
Predikatenlogica
- Declaratief (Wat)

Queryverwerking
-en optimalisatie

Declarativiteit

Praktisch

SQL (Structured Query Language)

- Vooral declaraties (Wat)
- Alle bewerkingen op databanken

Inleiding

SQL is dé industriële standaard, en deel van vele RDBMSs.
Voorganger: SEQUEL (Structured English QUery Language)

De standaarden: SQL1 ('86)), SQL2 (SQL92), **SQL3 (SQL99)**,
SQL2003, SQL2006, SQL2011

Implementaties SQL verschillen tussen RDBMSs en wijken in meerdere en mindere mate af van de standaard: PostgreSQL volgt de standaard goed.

Terminologie SQL t.a.v. Relationeel model en algebra:

schema (= schema), tabel (= relatie), rij (= tuple), kolom (= attribuut)

Verskil SQL t.a.v. RM/RA:

SQL elimineert dubbels niet automatisch => multiset vs. set

SQL – Gegevens opvragen

Syntax v. vraag-query:

```
SELECT    <attributen>  
FROM      <tabellen>  
WHERE     <condities> ;
```

Vergelijking met Relationele Algebra:

```
SELECT    'Projectie'  
FROM      'Cartesisch product' (of join)  
WHERE     'Selectie'
```

SQL – Gegevens opvragen (voorbeeld)

Geef geboortedatum en adres van werknemer John Smith.

```
SELECT    Bdate, Address
FROM      EMPLOYEE
WHERE     Fname='John' AND Lname = 'Smith' ;
```

*Terminal
Output*



Bdate	Address
1965-01-09	731 Fondren, Houston, TX

In RA: $\pi_{Bdate, Address} (\sigma_{Fname='John' \wedge Lname='Smith'} (EMPLOYEE))$

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

SQL – Gegevens opvragen (voorbeeld)

Geef naam en adres van alle werknemers die voor departement 'Research' werken.

```
SELECT    Fname, Lname, Address
FROM      EMPLOYEE, DEPARTMENT
WHERE     Dname='Research' AND Dnumber = Dno;
```

In RA: $\pi_{Fname, Lname, Address}(\sigma_{Dname='Research' \wedge Dnumber = Dno}(\text{DEPARTMENT} \times \text{EMPLOYEE}))$

Of: $\pi_{Fname, Lname, Address}(\sigma_{Dname='Research'}(\text{DEPARTMENT} \bowtie_{Dnumber = Dno} \text{EMPLOYEE}))$

EMPLOYEE									
Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
DEPARTMENT									
Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date						

SQL – Joins opbouwen (Voorbeeld)

Geef namen van projecten gecontroleerd door dept. 5.

```
SELECT Pname  
FROM PROJECT  
WHERE Dnum=5 ;
```

PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

WORKS_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

EMPLOYEE

<u>Fname</u>	<u>Minit</u>	<u>Lname</u>	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
--------------	--------------	--------------	------------	-------	---------	-----	--------	-----------	-----

DEPARTMENT

<u>Dname</u>	<u>Dnumber</u>	<u>Mgr_ssn</u>	<u>Mgr_start_date</u>
--------------	----------------	----------------	-----------------------

DEPT_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

SQL – Joins opbouwen (Voorbeeld)

Geef ssns van werknemers die werken op projecten gecontroleerd door dept. 5.

```
SELECT Essn
FROM PROJECT, WORKS_ON
WHERE Dnum=5 AND Pno=Pnumber ;
```

PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

WORKS_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

DEPARTMENT

<u>Dname</u>	<u>Dnumber</u>	<u>Mgr_ssn</u>	<u>Mgr_start_date</u>
--------------	----------------	----------------	-----------------------

DEPT_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

SQL – Joins opbouwen (Voorbeeld)

Geef voornaam en familienaam van werknemers die werken op projecten gecontroleerd door dept. 5.

```
SELECT    Fname,Lname
FROM      PROJECT, WORKS_ON, EMPLOYEE
WHERE     Dnum=5 AND Pno=Pnumber AND Essn=Ssn ;
```

PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

WORKS_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

DEPARTMENT

<u>Dname</u>	<u>Dnumber</u>	<u>Mgr_ssn</u>	<u>Mgr_start_date</u>
--------------	----------------	----------------	-----------------------

DEPT_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

SQL – Joins opbouwen (Voorbeeld)

Geef voor elk project met locatie 'Stafford', het projectnummer, controllerend departementsnummer, en de familienaam van de manager van het departement.

```
SELECT Pnumber, Dnum, Lname
FROM PROJECT, DEPARTMENT, EMPLOYEE
WHERE Plocation='Stafford' AND
Dnum=Dnumber AND Mgr_ssn=Ssn ;
```

PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

WORKS_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

DEPARTMENT

<u>Dname</u>	<u>Dnumber</u>	<u>Mgr_ssn</u>	<u>Mgr_start_date</u>
--------------	----------------	----------------	-----------------------

DEPT_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

SQL – Impliciete vs. Expliciete JOIN

Impliciet:

```
SELECT Pname, Essn
FROM PROJECT, WORKS_ON
WHERE Dnum=5 AND Pno=Pnumber ;
```

Expliciet:

```
SELECT Pname, Essn
FROM (PROJECT JOIN WORKS_ON ON Pno=Pnumber)
WHERE Dnum=5 ;
```

SQL – NATURAL en OUTER JOINS

NATURAL JOIN:

```
SELECT    Dname, Dlocation
FROM      (DEPARTMENT NATURAL JOIN DEPT_LOCATIONS)
WHERE     Dnum=5 ;
```

OUTER JOIN:

```
SELECT    Fname, Lname
FROM      (EMPLOYEE LEFT OUTER JOIN DEPARTMENT
          ON Ssn=Mgr_ssn)
WHERE     Sex='F' ;
```

SQL – Dubbelzinnige attribuutnamen

Verschillende tabellen kunnen attributen met dezelfde naam bevatten. Dit wordt opgelost via: <tabel>.<attribuut>

Vb. als Dno in EMPLOYEE, Dnumber was:

```
SELECT    Lname, Fname
FROM      EMPLOYEE, DEPARTMENT
WHERE     Dname='Research' AND
          DEPARTMENT.Dnumber = EMPLOYEE.Dnumber ;
```

Exactere notatie mag altijd gebruikt worden in alle attributen van query!

SQL – Aliassen

Wat als we dezelfde tabel willen joinen?

Vb. Geef voor elke werknemer voor -en achternaam alsook de achternaam van zijn overste.

```
SELECT      E.Lname, E.Fname, S.Lname
FROM        EMPLOYEE AS E, EMPLOYEE AS S
WHERE       E.Sup_ssn = S.ssn ;
```

Dus met sleutelwoord AS, maar ook onmiddellijk na tabelnaam, en via oplijsting:

```
SELECT      E.Lname, E.Fname, S.Lname
FROM        EMPLOYEE E S
WHERE       E.Sup_ssn = S.ssn ;
```

Moet als dubbelzinnig, maar mag altijd om overzichtelijk te maken.

SQL – Weglaten WHERE of attributen

Weglaten van WHERE? M.a.w. Geen selectie

```
SELECT      Ssn  
FROM        EMPLOYEE ;
```

```
SELECT      Ssn, Dname  
FROM        EMPLOYEE, DEPARTMENT ;    => Cartesisch Product
```

Weglaten van attributen? M.a.w. Geen projectie

```
SELECT      *  
FROM        EMPLOYEE  
WHERE       Dno=5 ;
```

=> Alle attributen selecteren!

SQL – Vermijden van dubbelen

In SQL worden dubbelen niet automatisch verwijderd:

```
SELECT      Salary  
FROM        EMPLOYEE ;
```

```
SELECT      ALL Salary  
FROM        EMPLOYEE ;    => Expliciete versie
```

Deze moeten altijd expliciet verwijderd worden:

```
SELECT      DISTINCT Salary  
FROM        EMPLOYEE ;
```

SQL – Verzamelingoperaties

UNION, INTERSECT, EXCEPT (= MINUS)

- * Tabellen moeten unie-compatibel zijn
 - => Zelfde attributen en volgorde (cfr. RA met verschil volgorde)
- * Dubbelen worden hier wel automatisch verwijderd (cfr. RA)

Toch met multisets werken? ALL toevoegen:

- * UNION ALL
- * INTERSECT ALL
- * EXCEPT ALL

SQL – Verzamelingoperaties

Vb. Alle projecten waar Smith aan meewerkt of waar hij manager is van het controlerende departement.

```
( SELECT DISTINCT Pnumber
  FROM PROJECT, DEPARTMENT, EMPLOYEE
  WHERE Dnum=Dnumber AND Mgr_ssn=Ssn AND Lname='Smith' )
```

UNION

```
( SELECT DISTINCT Pnumber
  FROM PROJECT, WORKS_ON, EMPLOYEE
  WHERE Pno=Pnumber And Essn=Ssn And Lname='Smith' ) ;
```

SQL – String operaties

Wanneer we willen testen op patronen in Strings:

- * Gebruik LIKE in combinatie met '_' voor één willekeurig teken en '%' voor willekeurige rij tekens.
- * Escape (vb. '\%') als karakter voorkomt of speciaal (vb. '\t' of '\n')

Vb. Naam van alle werknemers in Leuven.

```
SELECT      Fname, Lname
FROM        EMPLOYEE
WHERE       Address LIKE '%Leuven%' ;
```

Vb. Naam van alle werknemers geboren in januari van de jaren '90.

```
SELECT      Fname, Lname
FROM        EMPLOYEE
WHERE       Bdate LIKE '199_-01-__' ;           of ('199_-01%')
```

SQL – Rekenkundige operatoren

Kunnen gebruikt worden op numerieke waarden:

- * Als deel van condities in WHERE.
- * Als deel van attributen in SELECT om hun uitkomst te beïnvloeden.

Vb. Geef naam en salaris, verhoogd met 10%, van werknemers werkende op projecten startende met de term 'Product'.

```
SELECT      Fname, Lname, 1.1*Salary AS Incr_sal  
FROM        EMPLOYEE, WORKS_ON, PROJECT  
WHERE       Ssn=Essn And Pno=Pnumber AND Pname LIKE 'Product%';
```

SQL – Andere operatoren

BETWEEN

Vb. Werknemers van dept. 5 met salaris tussen 30000 en 40000.

```
SELECT      *  
FROM        EMPLOYEE  
WHERE       (Salary BETWEEN 30000 AND 40000) AND Dno=5 ;
```

En dan is er nog:

- * String concatenatie: ||
- * Datum, tijd, tijdstempel
 - Verhogen, verlagen met interval
 - Verschil geeft interval
- * ...

SQL – Ordenen van resultaten

De standaard ordening is stijgend! Om dit te beïnvloeden:
ORDER BY <attributenlijst ([ASC|DESC])>

Vb. Werknemers (namen) en projecten waarop ze werken (namen) per departement, en ik elk departement alfabetisch geordend volgens familienaam en voornaam.

```
SELECT      Dname, Fname, Lname, Pname
FROM        DEPARTMENT, EMPLOYEE, WORKS_ON, PROJECT
WHERE       Dnumber=Dno AND Ssn=Essn AND Pno=Pnumber
ORDER BY    Dname, Lname, Fname ;
```

Opgelet: In volgende geval ORDER BY niet nodig!

```
SELECT      Dname, Lname, Fname, Pname
```

...

SQL – Ordenen van resultaten

Vb. Werknemers (namen) en projecten waarop ze werken (namen) per departement, en ik elk departement alfabetisch geordend volgens familienaam en voornaam.

```
SELECT      Dname, Fname, Lname, Pname
FROM        DEPARTMENT, EMPLOYEE, WORKS_ON, PROJECT
WHERE       Dnumber=Dno AND Ssn=Essn AND Pno=Pnumber
ORDER BY    Dname, Lname, Fname ;
```

Expliciet:

```
ORDER BY    Dname ASC, Lname ASC, Fname ASC ;
```

Departementsnamen in dalende orde:

```
ORDER BY    Dname DESC, Lname ASC, Fname ASC ;
```