Date No

controls = (dNumber, PNumber) Explicit implementation of "controls" Easy & recommended (for now)

has assigned = (dNumber, SSN)

is managed by = (dNumber, SSN, startdate)

works = (SSN, PNumber, hour)

supervisor = (SSN_up, SSN_down)

dependent of = (SSN, fname)

Relational Schema

DEPARTMENT = (dNumber, dName)

DEPT_LOC = (dNumber, dloc)

EMPLOYEE = (SSN, ename, address, salary, birthdate, sex)

DEPENDENT = (fname, SSN, relationship, birthdate, sex)

Relational Algebra

Operators

SELECT symbol = 6 (sigma)

syntax = 6 <selection condition> (R)

where selection condition is boolean expression on the attributes of R

R = relational algebra expression

EMPLOYEE

e_id	ename	es_name	salary
e001	Ece	Ece	F 50
e050	Oya	Ece	F 20
e100	Ece	Ece	F 70
e2	Ali	Omer	M 20

attributes (sonradan eklendi)

Tuple

Use relational algebra to select all employees with salary > 30 out of EMPLOYEE relation

$R_1 \leftarrow \sigma_{\text{<salary > 30>}}(\text{EMPLOYEE})$

Use RA to select all emp. having name = Ece and salary > 20

$R_1 \leftarrow \sigma$ (EMPLOYEE)

$<ename = Ece>$ AND $<salary > 20>$

$R_2 \leftarrow \sigma_{<ename=Ece>}\left(\sigma_{<salary > 20>}(EMPLOYEE)\right)$

Cardinality = number of tuple || $|\sigma_c(R)| \leq |R|$

Degree = number of attribute

degree (EMPLOYEE) = 4

SELECT operator does not change the degree

Card (EMPLOYEE) = 4 (Tuples, rows) $|EMPLOYEE| = 4$

Card (R_1) = 2

Card (R_2) = 1

PROJECT symbol = Π

syntax = Π (R)

list attribute

Use RA to select <e_id, salary> attribute out of EMPLOYEE

$R_3 \leftarrow \Pi_{<e\_id, salary>}$ (EMPLOYEE)

degree = 2

card = 4

Date No

Use RA to project EMPLOYEE over salary

$R_4 \leftarrow \Pi_{\text{salary}}(\text{EMPLOYEE})$ degree $(R_4) = 1$

card $(R_4) = 3$ duplicate sayma

R_4

salary
50
20
70

Use RA to retrieve all info of female employees

$\text{Result} \leftarrow \sigma_{<sex = f>}(\text{EMPLOYEE})$ card (Result) = 3

degree (Result) = 5

Use RA to retrieve all info of employee having fname = Ece

$R_2 \leftarrow \sigma_{<fname = Ece>}(\text{EMPLOYEE})$ card $(R_2) = 2$

degree $(R_2) = 5$

Use RA to retrieve all info of employees having fname = Ece and salary ≥ 50

$R_3 \leftarrow \sigma_{<fname = Ece> \text{ AND } <salary \geq 50>}(\text{EMPLOYEE})$ card $(R_3) = 2$

degree $(R_3) = 5$

Use RA to retrieve fname and salary of all employees

$R_4 \leftarrow \Pi_{<fname, salary>}(\text{EMPLOYEE})$ degree $(R_4) = 2$

card $(R_4) = 4$

Use RA to retrieve fname and salary of female employees

$Temp1 \leftarrow \sigma_{<sex=F>}(EMPLOYEE)$ degree = 5, card = 3

$R_S \leftarrow \Pi_{<fname, salary>}(Temp1)$ degree = 2, card = 3

$R_S \leftarrow \Pi_{<fname, salary>}(\sigma_{sex=F}(EMPLOYEE))$

UNION, INTERSECTION and MINUS Syntax: $R \cup S$, $R \cap S$, $R - S$

R and S must be union compatible. They must have same number of attribute.

$R(A_1, A_2, \ldots, A_n)$ } $n = m$ $dom(A_i) = dom(B_i)$

$S(B_1, B_2, \ldots, B_n)$ } $\forall i = 1 \ldots n$

Kitaptaki örnek

degree (STUDENT) = 2 degree (INSTRUCTOR) = 2

card (STUDENT) = 7 card (INSTRUCTOR) = 5

T = STUDENT ∪ INSTRUCTOR

degree (T) = 2

card (T) = 10

Company Database örneği

Retrieve the SSN of all employees who either work in DNO5 or directly supervise an employee from DNO5

$DEPT5\_EMP \leftarrow \sigma_{<DNO=5>}(EMPLOYEE)$ degree (DEPT5_EMP) = 10

card (DEPT5_EMP) = 4

Date No

$SSN\_DEPTS\_EMP \leftarrow \underset{\langle SSN \rangle}{\Pi}(DEPTS\_EMP)$ degree = 1, card = 4

$SSN\_SUPERVISE \leftarrow \underset{\langle super.ssn \rangle}{\Pi}(DEPTS\_EMP)$ degree = 1, card = 2

$R = SSN\_DEPTS\_EMP \cup SSN\_SUPERVISE$

Cartesian Product

Syntax : $R \times S$

Semantic : $Q = R(A_1, A_2, \ldots, A_n) \times S(B_1, B_2, \ldots, B_n)$

$Q(A_1, A_2, \ldots A_n, B_1, B_2, \ldots, B_n)$

degree (Q) = degree (R) + degree (S)

If $|R| = n_r$ and $|S| = n_s$ then $|Q| = n_r \times n_s$

Use RA to retrieve the name of the female dependent of each employee together with "fname" and "Lname" of employee

$FEM\_DEP \leftarrow \Pi_{\langle ESSN, Dependent\_name \rangle}(\sigma_{\langle sex = F \rangle}(DEPENDENT))$ degree = 2, card = 4

$Temp \leftarrow FEM\_DEP \times EMPLOYEE$ degree = 10 + 2 = 12, card = 8 × 4

$Result \leftarrow \underset{\langle Fname, Lname, Dependent\_name \rangle}{\Pi}(\sigma_{\langle ESSN = SSN \rangle}(TEMP))$ degree = 3, card = 4

Reflexive relationship = (SSN, superSSN)

Date No

Use RA query to retrieve the LName of all employees working in DNO = 5.

$TEMP \leftarrow \sigma_{DNO=5}(EMPLOYEE)$ degree (TEMP) = 10, card (TEMP) = 4

$R \leftarrow \pi_{LName}(TEMP)$ degree = 1, card = 4

Use RA query to retrieve PName of all project located in Staffo

$R \leftarrow \pi_{PName}(\sigma_{PLocation=Stafford}(PROJECT))$ degree = 1, card = 2

Use RA query to retrieve the SSN of the manager of the Research Department.

$Temp \leftarrow \sigma_{Dname=Research}(DEPARTMENT)$ degree = 4, card = 1

$R \leftarrow \pi_{Mgr\_SSN}(Temp)$ degree = 1, card = 1

Use RA query to retrieve the fName of the manager of the Research Department.

$SSN\_Mng\_Res \leftarrow \pi_{Mgr\_SSN}(Temp)$

$All \leftarrow EMPLOYEE \times SSN\_Mng\_Res$ degree (All) = 11, card (All) = 8 x 1 = 8

Interesting_All ← σ (All)
SSN = Mgr_SSN

All & Interesting_All
↓
Interesting_All ← EMP ⋈ SSN_...
SSN = Mgr_SSN

Result ← π (Interesting_All)
FName

Use RA to retrieve the FName of employees who works less than 30 hours in at least 1 project

less_than_30 ← σ (WORKS_ON) degree (less_than_30) = 3
Hours < 30
card (less_than_30) = 12

ESSN_less_30 ← π (less_than_30) degree (ESSN_less_30) = 1
ESSN
card (ESSN_less_30) = 7

Temp1 ← EMPLOYEE × ESSN_less_30 degree (Temp1) = 11
card (Temp1) = 56

Temp2 ← σ (Temp1) degree (Temp2) = 11
ESSN = SSN
card (Temp2) = 7

Result ← π (Temp2) degree (Result) = 1
FName
card (Result) = 6

EMPLOYEE ⋈ ESSN_less_30
ESSN = SSN

Date No

Functional Dependency: It is a constraint between 2 sets of attributes.

Ex: $R(A_1, A_2, \ldots, A_n)$

$X = \{A_1, A_2\}$ $y = \{A_3, A_4\}$

fd: $X \rightarrow y$

for every 2 tuples, t_1 and t_2 in the snapshot

$t_1[x] = t_2[x] \longrightarrow_{\text{then}} t_1[y] = t_2[y]$

Ex:

In EMPLOYEE $X = \{SSN\}$ $y = \{Bdate\}$

fd1: $X \rightarrow y$ It is correct because if

$t_1[x] = t_2[x]$ I have 2 rows with the same SSN ☆

$t_1[y] = t_2[y]$

Ex: WORKS_ON fd2: $X \rightarrow y$ $X = \{ESSN\}$ $y = \{Hours\}$

If $t_1[x] = t_2[x] \rightarrow t_1[y] = t_2[y]$ wrong

32.5 7.5

Ex:

EMPLOYEE + PROJECT + WORKS ON

SSN	PNumber	Hours	Ename	Pname	Plocation

fd1 = {SSN} → {Ename}

fd2 = {Pnumber} → {Pname, Plocation}

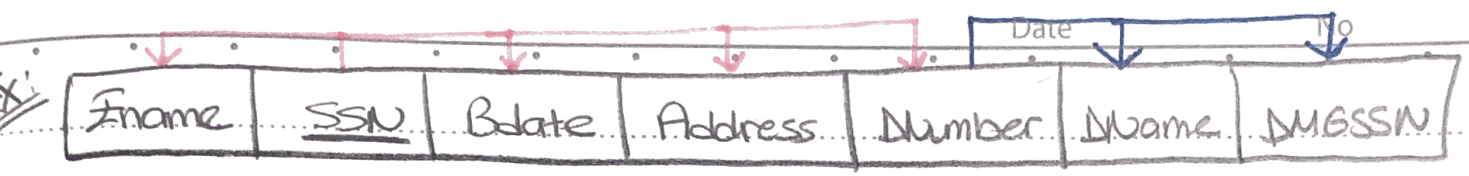
fd3 = {SSN, Pnumber} → {Hours}

SSN	PNumber	Hours

SSN	Ename

Pnumber	Pname	Plocation

Partial Dependency ⇒ ~~2NF~~



fd1 = { SSN } → { Fname, Bdate, Address, DNumber }

fd2 = { DNumber } → { DName, DMGSSN }

Transitive Dependency ⇒ ~~3NF~~

SSN	Fname	Bdate	Address	DNumber

DNumber	DName	DMGSSN