Relational Algebra exercises

lunedì 14 marzo 2022 11:49

EMPLOYEE (NAME, SURNAME, CF, BIRTH_DATE, ADDRESS, GENDER, SALARY, SUPERVISOR, DEPT)

DEPARTMENT (DCODE, DNAME, MANAGER, START_DATE)

PROJECT (PNUMBER, PNAME, DCODE)

WORKS_ON (CF, PNUMBER, HOURS)

DEPENDANT (CF, NAME, GENDER, BIRTH_DATE, PARENTAL_RELATIONSHIP)

[→ : FK, — : PART OF PK]

EX. 0: RETURN THE NAME AND SURNAME of ALL EMPLOYEES

$\Pi_{NAME, SURNAME, CF}$ (EMPLOYEE)

OBSERVE THAT WE INCLUDE "CF" IN THE PROJECTION, TO AVOID CLASHES

EX. 0(BIS): RETURN THE CF OF EMPLOYEES WITH A SALARY GREATER THAN 30.000, AND OF GENDER 'FEMALE'

$\sigma_{SALARY > 30.000 \text{ AND } GENDER = 'FEMALE'}$ (EMPLOYEE)

EX. 0(TER): RETURN THE CF OF EMPLOYEES WHO MANAGES A DEPARTMENT

$\Pi_{MANAGER}$ (DEPARTMENT)

NOTE THAT, EVEN IF AN EMPLOYEE MANAGES > 1 DEPARTMENTS, WE RETURN IT ONLY ONCE

EX. 1: RETURN CF, NAME AND SURNAME OF EMPLOYEES WHO MANAGE A DEPARTMENT TOGETHER WITH THE MANAGED DEPARTMENT NAME

$\Pi_{NAME, SURNAME, \ldots}$ (EMPLOYEE $\bowtie_{CF = MANAGER}$ DEPARTMENT)

$$\Pi_{\substack{\text{NAME},\\ \text{SURNAME},\\ \text{CF, DNUMBER}}} (\text{EMPLOYEE} \bowtie_{\text{CF}=\text{MANAGER}} \text{DEPARTMENT})$$

EX. 2: FIND THOSE EMPLOYEES WHOSE SALARY IS > THAN THAT OF THEIR SUPERVISOR

$$\text{EMP1} \leftarrow \rho_{\text{NAME1} \ldots \text{DEPT1} \leftarrow \text{NAME} \ldots \text{DEPT}} (\text{EMPLOYEE})$$

$$T \leftarrow \Pi_{\substack{\text{CF},\\ \text{SALARY},\\ \text{SALARY1}}} (\text{EMPLOYEE} \bowtie_{\substack{\text{SUPERVISOR}\\ =\text{CF1}}} \text{EMP1})$$

$$R \leftarrow \Pi_{\text{CF}} (\sigma_{\substack{\text{SALARY}\\ > \text{SALARY1}}} (T))$$

NOTE THAT WE HAVE RENAMED THE TABLE EMPLOYEE - THIS ALLOWED US TO SPECIFY THE CONDITION IN THE JOIN OPERATION WITHOUT ANY AMBIGUITIES

AS FOR THE JOIN ITSELF, NOTE THAT TUPLES WITH A NULL VALUE FOR SUPERVISOR ALWAYS FAIL THE COMPARISON, THUS THEY WILL NOT BELONG TO THE RESULT SET

EX. 3: RETURN EMPLOYEES WHO DO NOT HAVE ANY DEPENDANTS

$$R \leftarrow \Pi_{\text{CF}} (\text{EMPLOYEE}) - \Pi_{\text{CF}} (\text{DEPENDANT})$$

HERE, WE HAVE REASONED IN TERMS OF FINDING "NEGATIVE CASES" FIRST - WE IDENTIFIED A SET OF POSSIBLE CANDIDATE TUPLES TO RETURN, AND FROM SUCH A SET WE REMOVED THOSE THAT WERE NO GOOD FOR US -

---

EX. 4: RETURN EMPLOYEES THAT HAVE AT LEAST 2 DEPENDANTS

$$D \leftarrow \rho_{\text{CF-E} \ldots \leftarrow \text{CF} \ldots} (\text{DEPENDANT})$$

$$R \leftarrow \Pi_{\text{CF}} (\text{DEPENDANT} \bowtie_{\substack{\text{CF} = \text{CF-E}\\ \text{AND}\\ \text{NAME} \neq \text{NAME-E}}} D)$$

IN "DEPENDANT" WE HAVE ALL EMPLOYEES WITH AT LEAST

IN DEPENDANT WE HAVE ALL EMPLOYEES WITH AT LEAST ONE DEPENDANT. TO EXTRACT THOSE WITH AT LEAST 2 OF THEM, WE HAVE TO FIND ANOTHER TUPLE IN D THAT REFERS TO THE SAME EMPLOYEE, HOWEVER WITH A DIFFERENT DEPENDANT

TO COUNT TILL 3, 4, 5, ... YOU SIMPLY GO ON REPLICATING TABLES AND PERFORMING JOINS.

EX. 5: RETURN EMPLOYEES THAT HAVE AT MOST TWO DEPENDANTS

$D1 \leftarrow \rho_{CF1 \ldots \leftarrow CF \ldots}(\text{DEPENDANT})$

$D2 \leftarrow \rho_{CF2 \ldots \leftarrow CF \ldots}(\text{DEPENDANT})$

$NO\_GOOD \leftarrow \Pi_{CF}(\text{DEPENDANT} \bowtie_{CF = CF1 \text{ AND } NAME \neq NAME1} D1) \bowtie_{CF1 = CF2 \text{ AND } NAME \neq NAME2 \text{ AND } NAME1 \neq NAME2} D2$

$R \leftarrow \Pi_{CF}(\text{EMPLOYEE}) - NO\_GOOD$

AGAIN, WE HAVE REASONED IN TERMS OF NEGATIVE CASES: WE COLLECTED ALL EMPLOYEES AND FROM THEM WE REMOVED THOSE WITH AT LEAST 3 DEPENDANTS.

IF WE WANTED ALL EMPLOYEES WITH AT MOST 2 DEPENDANTS, AND AT LEAST ONE, IN THE LAST LINE WE WOULD HAVE EXTRACTED OUR CANDIDATES WITH: $\Pi_{CF}(\text{DEPENDANT})$

EX. 6: FIND ALL EMPLOYEES THAT WORK ON ALL PROJECTS EMPLOYEE "RSS" WORKS ON.

- THIS QUERY INVOLVES UNIVERSAL QUANTIFICATION, BUT WE CAN ONLY WRITE QUERIES THAT RELY ON EXISTENTIAL CONDITIONS
- IDEA: TRANSLATE THE UNIVERSAL QUERY INTO AN EXISTENTIAL ONE
- THE "BAD" EMPLOYEES HERE ARE THOSE SUCH THAT THERE EXISTS AT LEAST ONE PROJECT

- THE "BAD" EMPLOYEES HERE ARE THOSE SUCH THAT THERE EXISTS AT LEAST ONE PROJECT OF "RSS" ON WHICH THEY DO NOT WORK

$$RP \leftarrow \Pi_{PNUMBER}(\sigma_{DF='RSS'}(WORKS\_ON))$$

$$SITUATION \leftarrow \Pi_{EF,PNUMBER}(WORKS\_ON)$$

$$ALL\_EMPS \leftarrow \Pi_{EF}(EMPLOYEE)$$

$$REQUIREMENTS \leftarrow ALL\_EMPS \times RP$$

$$NO\_GOOD \leftarrow \Pi_{EF}(REQUIREMENTS - SITUATION)$$

$$R \leftarrow ALL\_EMPS - NO\_GOOD$$

DOING "REQUIREMENTS - SITUATION", WE OBTAIN A PAIR (EMPLOYEE, PROJECT) FOR EACH REQUIREMENT THAT IS NOT SATISFIED — THE GIVEN EMPLOYEE DOES NOT WORK IN THE GIVEN PROJECT, BUT "RSS" DOES —

WHENEVER YOU HAVE A UNIVERSAL CONDITION, YOU CAN REASON IN TERMS OF CANDIDATES TO WHICH YOU SUBTRACT NO GOODS, THAT CAN BE IN TURN DESCRIBED BY MEANS OF AN EXISTENTIAL CONDITION —