

Окидачи (тригери) – задаци за вежбање

Задатак 1. Дата је база података са следећим табелама:

student_mast(STUDENT_ID, NAME, ST_CLASS),
student_marks(STUDENT_ID, SUB1, SUB2, SUB3, SUB4, SUB5, TOTAL, PER_MARKS, GRADE) и
student_log(USER_ID, DESCRIPTION).

Претпоставимо да табела student_marks садржи податке о ученицима који су се пријавили за полагање теста. Док се тест не заврши, у табели се за сваког ученика налазе само вредности атрибута STUDENT_ID, док су вредности осталих атрибута постављене на подразумеване вредности (0 или NULL). Након завршетка теста, ученици су добили оцене из 5 различитих предмета. Потребно је изменити податке у табели student_marks, при чему је потребно за сваког ученика који је радио тест унети резултате сваког од пет тестова, као и укупан број бодова које је освојио (TOTAL), али и просечну оцену на тесту (PER_MARKS) и свеукупну оцену (GRADE).

Вредности атрибута се рачунају на следећи начин:

Укупан број сакупљених поена: $TOTAL = SUB1 + SUB2 + SUB3 + SUB4 + SUB5$

Просечна оцена: $PER_MARKS = (TOTAL)/5$

Успех: GRADE=

- If $PER_MARKS \geq 90 \rightarrow$ 'EXCELLENT'
- If $PER_MARKS \geq 75$ AND $PER_MARKS < 90 \rightarrow$ 'VERY GOOD'
- If $PER_MARKS \geq 60$ AND $PER_MARKS < 75 \rightarrow$ 'GOOD'
- If $PER_MARKS \geq 40$ AND $PER_MARKS < 60 \rightarrow$ 'AVERAGE'
- If $PER_MARKS < 40 \rightarrow$ 'NOT PROMOTED'

Задатак 2. Школска година се завршила и почела је следећа, па је потребно за све ученике из табеле student_mast променити разред у следећи. Како не би случајно дошло до грешке да нови разред буде мањи од старог, креирати окидач који то спречава. Такође, управа жели да прати који су ученици уписали следећу годину у датој школи, као и који су се ученици пребацили у неку другу школу. Креирати окидач који ће обезбедити да се после сваке измене неке n-торке у табели student_mast дода нова n-торка у табелу student_log (чувају се информације о кориснику – **user_id** који је извршио измену и опис извршене измене). Направити и окидач који након сваког брисања n-торке из табеле student_mast додаје нови ред у табелу student_log са информацијама о кориснику који врши брисање и кратким описом извршеног брисања.

Задатак 3. Дата је табела emp_details, која представља део базе људских ресурса:

emp_details(EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER, HIRE_DATE, JOB_ID, SALARY, COMMISSION_PCT).

Креирати окидач који омогућава да се након сваког уношења података о запосленима у табелу emp_details изврши провера њихове зараде и ако је зарада већа од 20000, провизија тог запосленог (COMMISSION_PCT) се поставља на 0.1. У супротном, поставити вредност провизије на 0.5.

Задатак 4. Дата је апроксимација релационе базе једне издавачке куће следећом схемом:

P (P_SIF, IME, BR_NASLOVA, DRZAVA) – релација Писац

I (I_SIF, NAZIV, STATUS, DRZAVA) – релација Издавач

K (K_SIF, NASLOV, OBLAST) – релација Књига

KP (K_SIF, P_SIF, R_BROJ) – Аутор (веза Књиге и Писца)

KI (K_SIF, I_SIF, IZDANJE, GODINA, TIRAZ) – Издаваштво (веза Књиге и Издавача)

Приликом регистрације новог ауторског дела (унос п-торке у табелу KP), потребно је увећати вредност атрибута BR_NASLOVA за 1 у оквиру табеле P за писца који је аутор дела. Такође, у случају брисања ауторског дела, исти бројач је потребно смањити за 1. У случају да уочена грешка и да је за неку књигу унета погрешна шифра аутора, потребно је то исправити и ажурирати бројаче за оба писца.