

Primer: FAKULTET - Okidači za prosek studenta.

Napomene:

Prosek studenta se regularno određuje:

- po polaganju ispita INSERT u tabelu POLAGANJE,
- po poništenju ispita UPDATE u tabeli POLAGANJE (kolona Status).

Uz to, moguće su sledeće naknadne ispravke grešaka pri unosu ocena:

- naknadni unos neunete ocene INSERT u tabelu POLAGANJE,
- uklanjanje pogrešno unete ocene DELETE u tabeli POLAGANJE,
- ispravka ocene iz 5 u prolaznu UPDATE u tabeli POLAGANJE (kolona Ocena),
- ispravka ocene iz prolazne u 5 UPDATE u tabeli POLAGANJE (kolona Ocena),
- ispravka ocene iz prolazne u prolaznu UPDATE u tabeli POLAGANJE (kolona Ocena).

Najefikasnije je rešenje sa okidačima koji reaguju na događaje INSERT, DELETE i UPDATE u tabeli POLAGANJE, pri čemu se unutar njih vrši eventualno grananje na konkretne situacije.

Okidači:

```
CREATE TRIGGER InsertPolaganje
AFTER INSERT ON POLAGANJE
REFERENCING NEW ROW AS Novi
FOR EACH ROW
WHEN Novi.Ocena > 5
UPDATE STUDENT
SET BrojOcena = BrojOcena + 1,
    ZbirOcena = ZbirOcena + Novi.Ocena
WHERE IDStu = Novi.IDStu ;
```

Regularni i naknadni unos

```
CREATE TRIGGER DeletePolaganje
AFTER DELETE ON POLAGANJE
REFERENCING OLD ROW AS Stari
FOR EACH ROW
WHEN Stari.Ocena > 5
UPDATE STUDENT
SET BrojOcena = BrojOcena - 1,
    ZbirOcena = ZbirOcena - Stari.Ocena
WHERE IDStu = Stari.IDStu ;
```

Uklanjanje pogrešnog unosa

```

CREATE TRIGGER UpdatePolaganje
AFTER UPDATE ON POLAGANJE
REFERENCING OLD ROW AS Stari
NEW ROW AS Novi
FOR EACH ROW
BEGIN
    IF Novi.Status = 'P' THEN
        UPDATE STUDENT
        SET BrojOcena = BrojOcena - 1,
            ZbirOcena = ZbirOcena - Novi.Ocena
        WHERE IDStu = Novi.IDStu ;
    ELSE
        IF Stari.Ocena = 5 AND Novi.Ocena > 5 THEN
            UPDATE STUDENT
            SET BrojOcena = BrojOcena + 1,
                ZbirOcena = ZbirOcena + Novi.Ocena
            WHERE IDStu = Novi.IDStu ;
        ELSEIF Stari.Ocena > 5 AND Novi.Ocena = 5 THEN
            UPDATE STUDENT
            SET BrojOcena = BrojOcena - 1,
                ZbirOcena = ZbirOcena - Stari.Ocena
            WHERE IDStu = Novi.IDStu ;
        ELSEIF Stari.Ocena > 5 AND Novi.Ocena > 5 THEN
            UPDATE STUDENT
            SET ZbirOcena = ZbirOcena + ( Novi.Ocena - Stari.Ocena )
            WHERE IDStu = Novi.IDStu ;
        END IF
    END IF
END

```

Poništavanje ocene

Ispravka ocene
iz =5 u >5

iz >5 u =5

iz >5 u >5

Drugi način - preko pomoćnih SQL procedura:

```
CREATE TRIGGER InsertPolaganje
AFTER INSERT ON POLAGANJE
REFERENCING NEW ROW AS Novi
FOR EACH ROW
WHEN Novi.Ocena > 5
CALL UvecajProsek ( Novi.IDStu, Novi.Ocena ) ;
```

Regularni i naknadni unos

```
CREATE TRIGGER DeletePolaganje
AFTER DELETE ON POLAGANJE
REFERENCING OLD ROW AS Stari
FOR EACH ROW
WHEN Stari.Ocena > 5
CALL UmanjiProsek ( Stari.IDStu, Stari.Ocena ) ;
```

Uklanjanje pogrešnog unosa

```
CREATE TRIGGER UpdatePolaganje
AFTER UPDATE ON POLAGANJE
REFERENCING OLD ROW AS Stari
NEW ROW AS Novi
FOR EACH ROW
BEGIN
  IF Novi.Status = 'P' THEN
    CALL UmanjiProsek ( Novi.IDStu, Novi.Ocena ) ;
  ELSE
    IF Stari.Ocena = 5 AND Novi.Ocena > 5 THEN
      CALL UvecajProsek ( Novi.IDStu, Novi.Ocena ) ;
    ELSEIF Stari.Ocena > 5 AND Novi.Ocena = 5 THEN
      CALL UmanjiProsek ( Novi.IDStu, Stari.Ocena ) ;
    ELSEIF Stari.Ocena > 5 AND Novi.Ocena > 5 THEN
      CALL IspraviProsek ( Novi.IDStu, Novi.Ocena, Stari.Ocena ) ;
    END IF
  END IF
END
```

Poništavanje ocene

Ispravka ocene

iz =5 u >5

iz >5 u =5

iz >5 u >5

```
PROCEDURE UvecajProsek ( IN :IDStu ..., IN :Ocena ... )
BEGIN
  UPDATE STUDENT
  SET    BrojOcena = BrojOcena + 1,
        ZbirOcena = ZbirOcena + :Ocena
  WHERE IDStu = :IDStu ;
END
```

```
PROCEDURE UmanjiProsek ( IN :IDStu ..., IN :Ocena ... )
BEGIN
  UPDATE STUDENT
  SET    BrojOcena = BrojOcena - 1,
        ZbirOcena = ZbirOcena - :Ocena
  WHERE IDStu = :IDStu ;
END
```

```
PROCEDURE IspraviProsek ( IN :IDStu ..., IN :NovaOcena ..., IN :StaraOcena ... )
BEGIN
  UPDATE STUDENT
  SET    ZbirOcena = ZbirOcena + ( :NovaOcena - :StaraOcena )
  WHERE IDStu = :IDStu ;
END
```