

Primer: FAKULTET - Procedura za prijavu ispita.

Napomene:

U okviru procedure PrijavaJednogIspita treba prvo proveriti postoji li rok koji je otvoren za prijave i utvrditi koji je to rok. Ako je to u redu, na osnovu predmeta i roka određuje se koji je ispit u pitanju. Nakon toga redom idu sledeće provere: da li je student slušao taj predmet, da li student ima uslov iz tog predmeta (eventualno), da student već nema važeću prolaznu ocenu iz tog predmeta. U slučaju da su sve provere uspešne, evidentira se prijava ispita i vraća ishod uspešnosti. U suprotnom, vraća se odgovarajući ishod neuspešnosti.

Algoritamska specifikacija:

```
PROCEDURE PrijavaJednogIspita ( >IDStu, >IDPre, <Ishod )
:   Ishod = USPEH
:   ProveraOtvorenostiRoka ( <IDRok, <>Ishod )
:   IF ( Ishod = USPEH )
:   |   OdredjivanjeIspita ( >IDRok, >IDPre, <IDIsp )
:   |   ProveraSlusanosti ( >IDStu, >IDPre, <>Ishod )
:   |   IF ( Ishod = USPEH )
:   |   |   ProveraUslova ( >IDStu, >IDPre, <>Ishod )
:   |   |   IF ( Ishod = USPEH )
:   |   |   |   ProveraNemaPolozeno ( >IDStu, >IDPre, <>Ishod )
:   |   |   |   IF ( Ishod = USPEH )
:   |   |   |   |   EvidentiranjePrijave ( >IDStu, >IDIsp )
:   |   |   |   |   +-
:   |   |   |   |   Ishod = VEC_POLOZENO
:   |   |   |   |   +-
:   |   |   |   |   Ishod = NEMA_USLOVA
:   |   |   |   +-
:   |   |   |   Ishod = NIJE_SLUSANO
:   |   +-
:   |   Ishod = NEMA_ROKA
```

SQL procedure i funkcije:

Osnovna Procedura:

```
PROCEDURE PrijavaJednogIspita ( IN :IDStu ..., IN :IDPre ..., OUT :Ishod ...)
BEGIN
  DECLARE vIDRok      ...,
          vIDisp      ...
  SET :Ishod = USPEH ;
  CALL ProveraOtvorenostiRoka ( vIDRok, :Ishod ) ;
  IF :Ishod = USPEH THEN
    SET vIDisp = OdredjivanjeIspita ( vIDRok, :IDPre ) ;
    SET :Ishod = ProveraSlusanosti ( :IDStu, :IDPre ) ;
    IF :Ishod = USPEH THEN
      SET :Ishod = ProveraUslova ( :IDStu, :IDPre ) ;
      IF :Ishod = USPEH THEN
        SET :Ishod = ProveraNemaPolozeno ( :IDStu, :IDPre ) ;
        IF :Ishod = USPEH THEN
          INSERT INTO PRIJAVA VALUES ( vIDisp, :IDPre, CURRENT_DATE ) ;
        ELSE
          SET :Ishod = VEC_POLOZENO ;
        END IF
      ELSE
        SET :Ishod = NEMA_USLOV ;
      END IF
    ELSE
      SET :Ishod = NIJE_SLUSANO ;
    END IF
  ELSE
    SET :Ishod = NEMA_ROKA ;
  END IF
  RETURN ;
END
```

Pomoćna procedura i funkcije:

```
PROCEDURE ProveraOtvorenostiRoka ( OUT :IDRok ..., OUT :Ishod )
BEGIN
  SELECT IDRok
  INTO :IDRok
  FROM ROK
  WHERE Status = 'O' ;
  IF SQLCODE = NOT_FOUND THEN
    SET :Ishod = NEUSPEH ;
  ELSE
    SET :Ishod = USPEH ;
  END IF
  RETURN ;
END
```

```

FUNCTION OdredjivanjeIspita ( :IDRok ..., :IDPre ... ) RETURNS ...
BEGIN
  DECLARE vIDisp ... ;
  SELECT IDisp
  INTO vIDisp
  FROM ISPIT
  WHERE IDRok=:IDRok AND IDPre=:IDPre ;
  RETURN vIDisp ;
END

```

```

FUNCTION ProveraSlusanosti ( :IDStu ..., :IDPre ... ) RETURNS ...
BEGIN
  DECLARE vIshod ;
  IF EXISTS ( SELECT * FROM SLUSANOST WHERE IDStu=:IDStu AND IDPre=:IDPre ) THEN
    SET vIshod = USPEH ;
  ELSE
    SET vIshod = NEUSPEH ;
  END IF
  RETURN vIshod ;
END

```

```

FUNCTION ProveraUslova ( :IDStu ..., :IDPre ... ) RETURNS ...
BEGIN
  DECLARE vTrebaUslov ...,
         vIshod ... ;
  SET vIshod = USPEH ;
  SELECT TrebaUslov
  INTO vTrebaUslov
  FROM PREDMET
  WHERE IDPre=:IDPre ;
  IF vTrebaUslov THEN
    IF EXISTS ( SELECT * FROM USLOV WHERE IDStu=:IDStu AND IDPre=:IDPre ) THEN
      SET vIshod = USPEH ;
    ELSE
      SET vIshod = NEUSPEH ;
    END IF
  END IF
  RETURN vIshod ;
END

```

```
FUNCTION ProveraNemaPolozeno ( :IDStu, :IDPre ) RETURNS ...
BEGIN
  DECLARE vIshod ...
  SET vIshod = USPEH ;
  IF EXISTS ( SELECT *
              FROM POLAGANJE P, ISPIT I
              WHERE P.IDStu = :IDStu
                    AND P.Ocena > 5
                    AND P.Status = 'V'
                    AND P.IDIsp = I.IDIsp
                    AND I.IDPre = :IDPre ) THEN
    SET vIshod = NEUSPEH ;
  END IF
  RETURN vIshod ;
END
```