

**TEHTÄVÄNANTO:**

Tee opintorekisteritietokantaan aliohjelma, joka lisää uuden opintosuorituksen. Syötteenä annetaan opiskelijan etunimi, sukunimi, kurssikoodi ja arvosana. Aliohjelman pitää tarkistaa, että opiskelija ja opintojakso ovat olemassa ja että arvosana on välillä 0 - 5.

Alkutilanne: Tavoitteena lisätä opiskelija Ayrtonille uusi arvosana käyttämällä aliohjelmaa.

```
mysql> select opiskelija.etunimi, opiskelija.sukunimi, arviointi.arvosana, opintojakso.nimi
-> FROM opiskelija
-> INNER JOIN arviointi ON arviointi.idOpiskelija = opiskelija.idOpiskelija
-> INNER JOIN opintojakso ON opintojakso.idOpintojakso = arviointi.idOpintojakso
-> WHERE opiskelija.etunimi = 'Ayrton' AND opiskelija.sukunimi = 'Senna';
+-----+-----+-----+-----+
| etunimi | sukunimi | arvosana | nimi                |
+-----+-----+-----+-----+
| Ayrton  | Senna    | 5        | Fysiikan perusteet  |
| Ayrton  | Senna    | 1        | Koneoppimisen perusteet |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

MySQL Workbench -sovelluksessa toteutettu aliohjelma ”*LisaaSuoritus*”:

Name:  The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
1 • CREATE PROCEDURE `LisaaSuoritus` (
2   IN En VARCHAR(45),
3   IN Sn VARCHAR(45),
4   IN KK VARCHAR(45),
5   IN ArvoS INT
6 )
7 LisaaSuoritus:BEGIN
8   declare OpiskeID INT DEFAULT 0;
9   declare OpintoID INT DEFAULT 0;
10
11   SELECT idOpiskelija INTO OpiskeID FROM Opiskelija WHERE Etunimi = En AND Sukunimi = Sn;
12   if OpiskeID = 0 then
13     SELECT 'Virhe: Opiskelijaa ei ole. Kokeile uudelleen.';
14     LEAVE LisaaSuoritus;
15   end if;
16
17   SELECT idOpintojakso INTO OpintoID FROM Opintojakso WHERE Koodi = KK;
18   if OpintoID = 0 then
19     SELECT 'Virhe: Opintojaksoa ei ole. Kokeile uudelleen.';
20     LEAVE LisaaSuoritus;
21   end if;
22
23   if ArvoS > 5 OR ArvoS < 0 then
24     SELECT 'Virhe: Arvosana ei validi. Sallitut arvo 0 - 5.';
25     LEAVE LisaaSuoritus;
26   end if;
27
28   INSERT INTO Arviointi VALUES(NULL,OpiskeID,OpintoID,ArvoS,CURDATE());
29 END
```

Kutsutaan LisaaSuoritus -funktiota:

```
mysql> CALL LisaaSuoritus('Ayrton', 'Senna', 'IN00EP46', 4);  
Query OK, 1 row affected (0.01 sec)
```

Lopputilanne: Päivitetty arviointitaulukko Ayrtonille:

```
mysql> select opiskelija.etunimi, opiskelija.sukunimi, arviointi.arvosana, opintojakso.nimi  
-> FROM opiskelija  
-> INNER JOIN arviointi ON arviointi.idOpiskelija = opiskelija.idOpiskelija  
-> INNER JOIN opintojakso ON opintojakso.idOpintojakso = arviointi.idOpintojakso  
-> WHERE opiskelija.etunimi = 'Ayrton' AND opiskelija.sukunimi = 'Senna';
```

etunimi	sukunimi	arvosana	nimi
Ayrton	Senna	5	Fysiikan perusteet
Ayrton	Senna	1	Koneoppimisen perusteet
Ayrton	Senna	4	Vektorit ja kompleksiluvut

```
3 rows in set (0.00 sec)
```

Virhetulostuksen esimerkki: **tulostaa kahdesti**

```
mysql> CALL LisaaSuoritus('Ayrton', 'Senna', 'IN00EP46', 6);  
+-----+  
| Virhe: Arvosana ei validi. Sallitut arvo 0 - 5. |  
+-----+  
| Virhe: Arvosana ei validi. Sallitut arvo 0 - 5. |  
+-----+  
1 row in set (0.00 sec)
```

1. Query OK, 0 rows affected (0.00 sec)

```
mysql> CALL LisaaSuoritus('Ayrto', 'Senna', 'IN00EP46', 4);  
+-----+  
| Virhe: Opiskelijaa ei ole. Kokeile uudelleen. |  
+-----+  
| Virhe: Opiskelijaa ei ole. Kokeile uudelleen. |  
+-----+  
1 row in set (0.00 sec)
```

2.

Pohdintaa: Tehtävän MySQL-koodissa en ole huomioinut sitä, onko opiskelijalla jo arvosana kyseisestä kurssista. En usko, että se olisi kovin haastavaa toteuttaa, vaikka emme ole käsitelleet MySQL rutiineja aikaisemmin. Tästäpä jatkokehitysidea!