MySQL (TRIGGER)

```
CREATE DATABASE ukazka_trigger;
USE ukazka_trigger;

CREATE TABLE podniky (id_pod int,
                      nazev varchar(50) NOT NULL,
                      ulice varchar(50) NOT NULL,
                      psc int NOT NULL,
                      telefon char(9),
                      primary key(id_pod));

CREATE TABLE hrusky (id_hru int auto_increment,
                     datum date,
                     kg int,
                     podnik int,
                     foreign key(podnik) references podniky(id_pod),
                     primary key(id_hru));

CREATE TABLE jablka (id_jab int auto_increment,
                     datum date,
                     kg int,
                     podnik int,
                     foreign key(podnik) references podniky(id_pod),
                     primary key(id_jab));

INSERT INTO podniky (id_pod, nazev, ulice, psc, telefon) VALUES (1, 'Slav', 'Husova 1324',
43201, '776555405');
INSERT INTO podniky (id_pod, nazev, ulice, psc, telefon) VALUES (2, 'Gold', 'Golovinova 2356',
43201, '607754245');
INSERT INTO podniky (id_pod, nazev, ulice, psc, telefon) VALUES (3, 'Bonna', 'Golovinova
2121', 43201, '772456123');

INSERT INTO hrusky (id_hru, datum, kg, podnik) VALUES (0, '2016-01-28', 6000, 1);
INSERT INTO hrusky (id_hru, datum, kg, podnik) VALUES (0, '2016-02-28', 9000, 2);
INSERT INTO hrusky (id_hru, datum, kg, podnik) VALUES (0, '2016-02-28', 5000, 1);
INSERT INTO hrusky (id_hru, datum, kg, podnik) VALUES (0, '2016-03-28', 13000, 3);

INSERT INTO jablka (id_jab, datum, kg, podnik) VALUES (0, '2016-01-28', 5000, 2);
INSERT INTO jablka (id_jab, datum, kg, podnik) VALUES (0, '2016-02-28', 7000, 3);
INSERT INTO jablka (id_jab, datum, kg, podnik) VALUES (0, '2016-02-28', 4000, 1);
INSERT INTO jablka (id_jab, datum, kg, podnik) VALUES (0, '2016-03-28', 1500, 2);
```

Milan Mroczkowski http://yesit.cz sata150@gmail.com

Vzorová data

E: podniky

id_pod	nazev	ulice	psc	telefon
1	Slav	Husova 1324	43201	776555405
2	Gold	Golovinova 2356	43201	607754245
3	Bonna	Golovinova 2121	43201	772456123

E: hrusky

id_hru	datum	kg	podnik
1	2016-01-28	6000	1
2	2016-02-28	9000	2
3	2016-02-28	5000	1
4	2016-03-28	13000	3

E: jablka

id_jab	datum	kg	podnik
1	2016-01-28	5000	2
2	2016-02-28	7000	3
3	2016-02-28	4000	1
4	2016-03-28	1500	2

```
SELECT podniky.nazev, jablka.kg AS jablka_kg, jablka.datum
FROM
podniky JOIN jablka ON (podniky.id_pod = jablka.podnik);
```

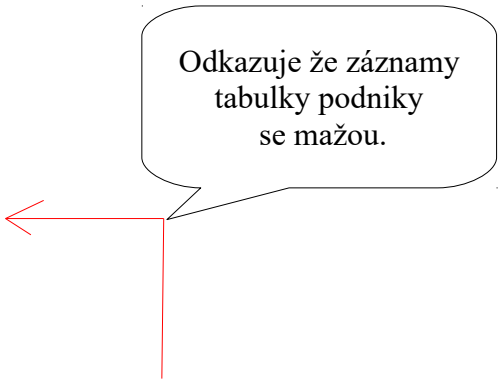
```
SELECT podniky.nazev, hrusky.kg AS hrusky_kg, hrusky.datum
FROM
podniky JOIN hrusky ON (podniky.id_pod = hrusky.podnik);
```

Milan Mroczkowski http://yesit.cz sata150@gmail.com

Řešení:

Cíl: Smazat související záznamy z dalších tabulek, pokud odstraním záznam o nějakém zaměstnanci.

DELIMITER - říká jaký bude oddělovač.
TRIGGER - spouštěč, vynucuje integritu
BEFORE DELETE ON - před smazáním záznamu/ů v třídě entit zamestnanci provede
BEGIN .. END




```
DELIMITER |

CREATE TRIGGER test
BEFORE DELETE ON podniky

FOR EACH ROW

BEGIN
DELETE FROM hrusky WHERE podnik = OLD.id_pod;
DELETE FROM jablka WHERE podnik = OLD.id_pod;
END|

DELIMITER ;
```

```
DELETE FROM podniky WHERE id_pod='3';
DELETE FROM podniky WHERE nazev='Gold';
```

```
DROP TRIGGER test;
SHOW TRIGGERS\G
SHOW CREATE TRIGGER test\G
```

Pojmy: INSERT, UPDATE, DELETE
BEFORE, AFTER
OLD, NEW

Milan Mroczkowski http://yesit.cz sata150@gmail.com

Jak nastavit to samé v MS SQL (ON DELETE CASCADE)

```
CREATE DATABASE on_delete_cascade;
USE on_delete_cascade;

CREATE TABLE podniky (id_pod int PRIMARY KEY,
                      nazev varchar(50) NOT NULL,
                      ulice varchar(50) NOT NULL,
                      psc int NOT NULL,
                      telefon char(9));

CREATE TABLE hrusky (id_hru int PRIMARY KEY IDENTITY(1,1),
                     datum date,
                     kg int,
                     podnik int,
                     foreign key(podnik) references podniky(id_pod));

CREATE TABLE jablka (id_jab int PRIMARY KEY IDENTITY(1,1),
                     datum date,
                     kg int,
                     podnik int,
                     foreign key(podnik) references podniky(id_pod) ON
                     DELETE CASCADE);

INSERT INTO podniky (id_pod, nazev, ulice, psc, telefon) VALUES (1, 'Slav', 'Husova 1324',
43201, '776555405');
INSERT INTO podniky (id_pod, nazev, ulice, psc, telefon) VALUES (2, 'Gold', 'Golovinova 2356',
43201, '607754245');
INSERT INTO podniky (id_pod, nazev, ulice, psc, telefon) VALUES (3, 'Bonna', 'Golovinova
2121', 43201, '772456123');

SET IDENTITY_INSERT hrusky ON;
INSERT INTO hrusky (id_hru, datum, kg, podnik) VALUES (1, '2016-01-28', 6000, 1);
INSERT INTO hrusky (id_hru, datum, kg, podnik) VALUES (2, '2016-02-28', 9000, 2);
INSERT INTO hrusky (id_hru, datum, kg, podnik) VALUES (3, '2016-02-28', 5000, 1);
INSERT INTO hrusky (id_hru, datum, kg, podnik) VALUES (4, '2016-03-28', 13000, 3);
SET IDENTITY_INSERT hrusky OFF;

SET IDENTITY_INSERT jablka ON;
INSERT INTO jablka (id_jab, datum, kg, podnik) VALUES (1, '2016-01-28', 5000, 2);
INSERT INTO jablka (id_jab, datum, kg, podnik) VALUES (2, '2016-02-28', 7000, 3);
INSERT INTO jablka (id_jab, datum, kg, podnik) VALUES (3, '2016-02-28', 4000, 1);
INSERT INTO jablka (id_jab, datum, kg, podnik) VALUES (4, '2016-03-28', 1500, 2);
SET IDENTITY_INSERT jablka OFF;

ALTER TABLE hrusky ADD CONSTRAINT test
foreign key (podnik) references podniky(id_pod) ON DELETE CASCADE;

DELETE FROM podniky WHERE id_pod='3';
DELETE FROM podniky WHERE nazev='Gold';
```

Milan Mroczkowski http://yesit.cz sata150@gmail.com