

MySQL (TRIGGER)

```
CREATE DATABASE ukazka_trigger;  
USE ukazka_trigger;
```

```
CREATE TABLE zamestnanci (id_zam int,  
                           prijmeni varchar(50) NOT NULL,  
                           jmeno varchar(50) NOT NULL,  
                           ulice varchar(50) NOT NULL,  
                           psc int NOT NULL,  
                           telefon char(9),  
                           primary key(id_zam));
```

```
CREATE TABLE platy (id_pla int auto_increment,  
                    datum date,  
                    castka int,  
                    zamestnanec int,  
                    foreign key(zamestnanec) references zamestnanci(id_zam),  
                    primary key(id_pla));
```

```
INSERT INTO zamestnanci (id_zam, jmeno, prijmeni, ulice, psc, telefon) VALUES (1, 'Milan',  
'Mroczkowski', 'Husova 1324', 43201, '776555405');
```

```
INSERT INTO zamestnanci (id_zam, jmeno, prijmeni, ulice, psc, telefon) VALUES (2, 'Silva',  
'Nová', 'Golovinova 2356', 43201, '607754245');
```

```
INSERT INTO zamestnanci (id_zam, jmeno, prijmeni, ulice, psc, telefon) VALUES (3, 'Anna',  
'Lysá', 'Golovinova 2121', 43201, '772456123');
```

```
INSERT INTO platy (id_pla, datum, castka, zamestnanec) VALUES (0, '2016-01-28', 17000, 1);
```

```
INSERT INTO platy (id_pla, datum, castka, zamestnanec) VALUES (0, '2016-02-28', 18000, 2);
```

```
INSERT INTO platy (id_pla, datum, castka, zamestnanec) VALUES (0, '2016-02-28', 19000, 1);
```

```
INSERT INTO platy (id_pla, datum, castka, zamestnanec) VALUES (0, '2016-03-28', 18500, 3);
```

E: zamestnanci

id_zam	prijmeni	jmeno	ulice	psc	telefon
1	Mroczkowski	Milan	Husova 1324	43201	776555405
2	Nová	Silva	Golovinova 2356	43201	607754245
3	Lysá	Anna	Golovinova 2121	43201	772456123

E: platy

id_pla	datum	castka	zamestnanec
1	2016-01-28	17000	1
2	2016-02-28	18000	2
3	2016-02-28	19000	1
4	2016-03-28	18500	3

SELECT prijmeni, jmeno, datum, castka **FROM** zamestnanci, platy
WHERE zamestnanci.id_zam = platy.zamestnanec;

prijmeni	jmeno	datum	castka
Mroczkowski	Milan	2016-01-28	17000
Nová	Silva	2016-02-28	18000
Mroczkowski	Milan	2016-02-28	19000
Lysá	Anna	2016-03-28	18500

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 zamestnanci

FOR EACH ROW

BEGIN

DELETE FROM platy WHERE zamestnanec = OLD.id_zam;

END|

DELIMITER ;

Odkazuje že záznamy
tabulky zamestnanci
se mažou.

DELETE FROM zamestnanci WHERE id_zam='3';
DELETE FROM zamestnanci WHERE jmeno='Milan';

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

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

Jak nastavit to samé v MS SQL (TRIGGER)

```
CREATE DATABASE ukazka_trigger;  
USE ukazka_trigger;
```

```
CREATE TABLE zamestnanci (id_zam int PRIMARY KEY,  
                           prijmeni varchar(50) NOT NULL,  
                           jmeno varchar(50) NOT NULL,  
                           ulice varchar(50) NOT NULL,  
                           psc int NOT NULL,  
                           telefon char(9));
```

```
CREATE TABLE platy (id_pla int PRIMARY KEY IDENTITY(1,1),  
                    datum date,  
                    castka int,  
                    zamestnanec int,  
                    foreign key(zamestnanec) references zamestnanci(id_zam));
```

Poznámka: IDENTITY(1,1) je auto_increment používaný v MySQL. První jednička je výchozí, která roste o jedničky nahoru, což je increment.

```
INSERT INTO zamestnanci (id_zam, jmeno, prijmeni, ulice, psc, telefon) VALUES (1, 'Milan',  
'Mroczkowski', 'Husova 1324', 43201, '776555405');  
INSERT INTO zamestnanci (id_zam, jmeno, prijmeni, ulice, psc, telefon) VALUES (2, 'Silva',  
'Nová', 'Golovinova 2356', 43201, '607754245');  
INSERT INTO zamestnanci (id_zam, jmeno, prijmeni, ulice, psc, telefon) VALUES (3, 'Anna',  
'Lysá', 'Golovinova 2121', 43201, '772456123');
```

SET IDENTITY_INSERT platy ON;

```
INSERT INTO platy (id_pla, datum, castka, zamestnanec) VALUES (1, '2016-01-28', 17000, 1);  
INSERT INTO platy (id_pla, datum, castka, zamestnanec) VALUES (2, '2016-02-28', 18000, 2);  
INSERT INTO platy (id_pla, datum, castka, zamestnanec) VALUES (3, '2016-02-28', 19000, 1);  
INSERT INTO platy (id_pla, datum, castka, zamestnanec) VALUES (4, '2016-03-28', 18500, 3);
```

SET IDENTITY_INSERT platy OFF;

SET IDENTITY_INSERT platy ON dočasně vypne auto_increment;

E: zamestnanci

id_zam	prijmeni	jmeno	ulice	psc	telefon
1	Mroczkowski	Milan	Husova 1324	43201	776555405
2	Nová	Silva	Golovinova 2356	43201	607754245
3	Lysá	Anna	Golovinova 2121	43201	772456123

E: platy

id_pla	datum	castka	zamestnanec
1	2016-01-28	17000	1
2	2016-02-28	18000	2
3	2016-02-28	19000	1
4	2016-03-28	18500	3

SELECT prijmeni, jmeno, datum, castka **FROM** zamestnanci, platy
WHERE zamestnanci.id_zam = platy.zamestnanec;

prijmeni	jmeno	datum	castka
Mroczkowski	Milan	2016-01-28	17000
Nová	Silva	2016-02-28	18000
Mroczkowski	Milan	2016-02-28	19000
Lysá	Anna	2016-03-28	18500

Řešení:

Tady to neřeší trigger, ale CONSTRAINT pro omezení cizího klíče.

ALTER TABLE platy **ADD CONSTRAINT** test
foreign key (zamestnanec) **references** zamestnanci(id_zam) **ON DELETE CASCADE;**

DELETE FROM zamestnanci **WHERE** id_zam='3';
DELETE FROM zamestnanci **WHERE** jmeno='Milan';

DROP DATABASE ukazka_trigger;
SP_WHO
USE master
kill 51;
GO

pokud nepůjde smazat databáze:
podívám se kdo je připojen, zkusím:

číslo nalezl ve sloupci SPID

Poznámka, lze to řešit i takto:

```
CREATE TABLE platy (id_pla int PRIMARY KEY IDENTITY(1,1),  
    datum date,  
    castka int,  
    zamestnanec int,  
    foreign key(zamestnanec) references zamestnanci(id_zam) ON  
    DELETE CASCADE);
```

Můžeme zadat ihned ON DELETE CASCADE.

Poznámka:

MySQL
SHOW TABLES;

MS SQL
SELECT * FROM ukazka_trigger.INFORMATION_SCHEMA.TABLES WHERE
TABLE_TYPE = 'BASE TABLE';

Najdu v kategorii Views