

Materiały do zajęć nr 9

Algebra relacyjna

Ściągamy plik:

home.agh.edu.pl/~makopiec/skrypt2.txt

i zapisujemy w katalogu domowym użytkownika student, a następnie wykonujemy go jako skrypt.

Na potrzeby testów tworzymy tabele Student, Przedmiot, Wykładowca:

```
create table Student(  
    Imię      char(15) ,  
    Nazwisko char(25) ,  
    Rok       int unsigned  
);
```

```
insert into Student values  
( 'Adam', 'Kowalski' ,2) ,  
( 'Ewa'  , 'Nowak'   ,3) ,  
( 'Jan'  , 'Kucharski',3);
```

```
create table Przedmiot(  
ID      char(2) ,  
Nazwa   char(25)  
);
```

```
insert into Przedmiot values  
( '10', 'Fizyka' ) ,  
( '11', 'Bazy danych' ) ,  
( '12', 'Statystyka' );
```

```
create table Wykładowca(  
Imię      char(15) ,  
Nazwisko char(25) ,  
ID        char(2)  
);
```

```
insert into Wykładowca values  
( 'Jerzy', 'Adamski' , '10' ) ,  
( 'Anna' , 'Leśniak' , '11' ) ,  
( 'Jakub', 'Nowak'   , '13' );
```

Projekcja

```
select Nazwisko,Rok from Student;
```

Selekcja

```
select * from Student where Rok=3;
```

Iloczyn kartezjański

```
select * from Wykładowca, Przedmiot;          standard SQL-89
```

```
select * from Wykładowca cross join Przedmiot; standard SQL-92
```

Można zrobić nawet iloczyn kartezjański tej samej tabeli, ale trzeba użyć aliasów:

```
select * from Student as S1 cross join Student as S2;
```

Uwaga: `cross join == inner join`

Złączenie naturalne

```
select * from Wykładowca natural join Przedmiot;
```

Można to zrobić również przez złączenie wewnętrzne:

```
select * from Wykładowca inner join Przedmiot using(Id);
```

```
select * from Wykładowca cross join Przedmiot using(Id); (to samo)
```

Złączenie wewnętrzne

Złączenie bez warunku

```
select * from Wykładowca inner join Przedmiot;
```

Złączenie θ (przykład: równozłączenie):

```
select * from Wykładowca inner join Przedmiot where Wykładowca.Id=Przedmiot.Id;
```

(to nie całkiem zgodne z formalną składnią)

```
select * from Wykładowca inner join Przedmiot on Wykładowca.Id=Przedmiot.Id;
```

Złączenie zewnętrzne lewostronne *(outer można pominąć)*

```
select * from Wykładowca left outer join Przedmiot  
on Wykładowca.Id=Przedmiot.Id;
```

```
select * from Wykładowca left join Przedmiot using (Id);
```

Ostatnia składnia zostawia tylko 1 powtarzającą się kolumnę i zmienia porządek kolumn.

Złączenie zewnętrzne prawostronne

```
select * from Wykładowca right outer join Przedmiot  
on Wykładowca1.Id=Przedmiot1.Id;
```

```
select * from Wykładowca right join Przedmiot using (Id);
```

Ostatnia składnia zostawia tylko 1 powtarzającą się kolumnę i zmienia porządek kolumn.

Złączenie zewnętrzne dwustronne (full)

W MySQL nie ma go, ale może być emulowane:

```
(select * from Wykładowca right outer join Przedmiot  
on Wykładowca.Id=Przedmiot.Id)  
union  
(select * from Wykładowca left outer join Przedmiot  
on Wykładowca.Id=Przedmiot.Id);
```

Klucze obce

Ściągamy plik:

home.agh.edu.pl/~makopec/skrypt3.txt

i zapisujemy w katalogu domowym użytkownika student, a następnie wykonujemy go jako skrypt i analizujemy przebieg wykonania.

```
use mytest;
```

```
set foreign key checks=0;
```

pozwała na usunięcie tabel powiązanych kluczami

```
drop table if exists wykładowcy;
```

```
drop table if exists przedmioty;
```

```
drop table if exists zajęcia;
```

```
set foreign key checks=1;
```

przywraca analizę powiązań kluczy

```
create table Przedmioty (  
    IP      int unsigned,  
    Nazwa char(30) not null,  
    primary key (IP)  
);
```

```
create table Wykładowcy (  
    IW      int unsigned,  
    Imię    char(15) not null,  
    Nazwisko char(25) not null,  
    Wydział char(10) not null,  
    primary key (IW)  
);
```

```
create table Zajęcia (  
    --
```

```

Id          int unsigned auto_increment,
Budynek     char(2) not null,
Sala        int unsigned not null,
Data        date,
IW          int unsigned,
IP          int unsigned,
primary key (Id),
foreign key (IP) references Przedmioty(IP) on delete cascade,
foreign key (IP) references Wykładowcy(IW) on delete cascade
);

```

```

desc Wykładowcy;
desc Przedmioty;
desc Zajęcia;

```

```

insert into Przedmioty values
(10,'Fizyka'),
(11,'Statystyka'),
(12,'Bazy danych'),
(13,'Termodynamika');

```

```

insert into Wykładowcy values
(1,'Adam','Kowalski','WEiP'),
(2,'Ewa','Nowak','WFiIS'),
(3,'Jan','Adamski','WIMiR');

```

```

insert into zajęcia values
(NULL,'D4',101,'2015-11-17',1,13),
(NULL,'C6',103,'2015-11-18',2,12),
(NULL,'D9',108,'2015-11-21',3,10),
(NULL,'D9',101,'2015-11-23',3,11);

```

```

select * from Wykładowcy;
select * from Przedmioty;
select * from Zajęcia;

```

```

delete from Wykładowcy where IW=3;

```

```

select * from Zajęcia;

```

Usunięcie danych wykładowcy powiązanych kluczem obcym utworzonym z warunkiem **on delete cascade** powoduje usunięcie zajęć prowadzonych przez tego wykładowcę.

Proszę sprawdzić działanie w przypadku użycia:

on delete restrict

on delete no action

on delete set null

Usunięcie klucza obcego

```

show create table zajęcia;

```

```

alter table zajęcia drop foreign key `zajęcia_ibfk_1`;

```