

1. (20 pkt.) Podstawowe polecenie wyboru danych z bazy w SQL ma postać:

SELECT A_1, A_2, \dots FROM R_1, R_2, \dots WHERE F ;

i jest praktycznie równoważne zapytaniu w algebrze relacji:

$$\pi_{A_1, A_2, \dots}(\sigma_F(R_1 \times R_2 \times \dots))$$

Na przykład:

SELECT sok FROM lubi WHERE osoba='Abacki';

SELECT sok FROM podaja,bywa WHERE podaja.bar=bywa.bar AND
bywa.osoba='Abacki';

Oznaczają odpowiednio „soki lubiane przez Abackiego” oraz „soki podawane w barach, w których bywa Abacki”. Różnica polega tylko na tym, że w zapytaniu SQL nie są automatycznie usuwane powtórzenia krotek z wyniku, a w algebrze relacji są. To oznacza, że potrafimy zadać już wiele pytań do bazy danych. Zapisz poniższe zapytania:

- (a) Napisz zapytanie zwracające nazwiska wszystkich studentów (czyli osoby na semestrze co najmniej 1 i co najwyżej 10), którzy mają na imię Piotr i są najwyżej na 3 semestrze (można to wszystko odczytać z tabeli **uzytkownik**).
 - (b) Napisz zapytanie zwracające nazwy przedmiotów i liczbę przypisanych im punktów dla wszystkich przedmiotów z egzaminem (można to wszystko odczytać z tabeli **przedmiot**).
 - (c) Odszukaj kod *baz danych, bieżącego semestru* oraz *baz danych w bieżącym semestrze* — jako odpowiedź podaj znalezione wartości atrybutów **kod_przed**, **semestr_id** oraz **kod_przed_sem**.
 - (d) Napisz zapytanie zwracające nazwiska wszystkich prowadzących dowolne zajęcia z *baz danych w bieżącym semestrze*. Wykorzystaj w tym celu wartość **kod_przed_sem** znaną w poprzednim punkcie i sformułuj zapytanie dotyczące tabel: **grupa** i **uzytkownik**.
2. (30 pkt.) Załóż tablicę *cwbd2009* zawierającą kolumny: *prow_id*, *prow_nazwisko*, *stud_id*, *stud_nazwisko*, *grupa_kod*, *termin*:
- *prow_id*, *prow_nazwisko* — są przeznaczone na kod oraz nazwisko prowadzącego;
 - *stud_id*, *stud_nazwisko* — są przeznaczone na kod oraz nazwisko studenta;

- *grupa_kod* — jest przeznaczone na kod grupy;
- *termin* — jest przeznaczone na termin zapisu do grupy.

Zadbaj, by typy kolumn były zgodne z typami kolumn z oryginalnej bazy zawierających odpowiednie dane. Zdefiniuj także klucze obce gwarantujące wystąpienie w bazie odpowiednich danych (czyli łączące tabelę *cwbd2009* z tabelami *uzytkownik* i *grupa*). Zdefiniuj jawnie wszystkie akcje referencyjne (nawet, gdy są domyślne) i odpowiednie więzy nakładające ograniczenia na wartości w kolumnie.

3. (20 pkt.) Znajdź w bazie następujące dane:

- Wyszukaj kody wszystkich grup *ćwiczenio-pracowni* do baz danych z tego semestru (możesz wykorzystać znaleziony wcześniej kod baz danych w bieżącym semestrze). Podaj te kody, jako odpowiedź na to pytanie.
- Wykorzystaj znalezione kody grup, by zadać pytanie zwracające kody i nazwiska studentów, numery grup i czas zapisu do grupy dla wszystkich studentów, którzy zapisali się do tych grup (jest to zapytanie `select` dotyczące tabel *uzytkownik* i *wybor*). Wynik tego zapytania wpisz do tablicy *cwbd2009* poleceniem:

```
INSERT INTO cwbd2009(stud_id, stud_nazwisko, grupa_kod, termin)
    „Twoje zapytanie SELECT”;
```

.

4. (20 pkt.) Polecenie modyfikacji, to:

```
UPDATE tablica SET atrybut1=wartość1, atrybut2=wartość2,... WHERE
    warunek;
```

Polecenie to wyszukuje w *tabeli* wszystkie krotki spełniające *warunek* i zmienia im wartość *attributu1* na *wartość1*, *attributu2* na *wartość2* itd. Na przykład polecenie:

```
UPDATE lubi SET sok='egzotyczny' WHERE sok='ananas' or sok='banan';
```

zmienia w tabeli *lubi* wpisy o zamiłowaniu do soku ananasowego lub bananowego na wpis o upodobaniu do soku egzotycznego.

Wyszukaj nazwiska i kody prowadzących poszczególne grupy *ćwiczenio-pracowni* do baz danych w bieżącym semestrze. Gdy już je znasz, to zmień zawartość tabeli *cwbd2009* tak, by uzupełnić te nazwiska poleceniami `UPDATE` (jednym poleceniem uzupełnij nazwisko prowadzącego we wszystkich zapisach dotyczących jednej grupy).

5. (30 pkt.) Wybrane z bazy dane można przedstawić w postaci uporządkowanej:

```
SELECT A,B,C FROM R,S WHERE F ORDER BY B,A DESC;
```

powoduje wybranie atrybutów A , B i C krotek z $R \times S$ spełniających warunek F , ale wynik jest porządkowany najpierw według wartości B (rosnąco), a przy jednakowych wartościach B — według A (malejąco). Wykorzystaj to, by znaleźć w każdej grupie osobę, która zapisała się do niej jako pierwsza. Zdefiniuj tabelę **najszybszybd2009** zawierającą trzy kolumny: `kod_grupy`, `stud_kod` i `stud_nazwisko` o typach zgodnych z typami kolumn z analogicznymi danymi z bazy oraz odpowiednich kluczach obcych. Wpisz znalezione wartości do tabeli **najszybszybd2009** pojedynczymi poleceniami

```
INSERT INTO najszybszybd2009 VALUES(...);
```