

Bazy danych 2 – zajęcia nr 3

Zapoznać się z dokumentacją:

- poleceń (INNER) JOIN, LEFT JOIN, RIGHT JOIN
- poleceń UNION, UNION ALL
- polecenia CREATE TEMPORARY TABLE
- atrybutów NOT_NULL oraz AUTO_INCREMENT

1. W domyślnej bazie na serwerze wierzba.wzks.uj.edu.pl utworzyć przy pomocy interfejsu użytkownika serwisu phpMyAdmin dwie tabele: „Tabela_A” oraz „Tabela_B” o takiej samej strukturze:

	Field	Type	Null	Key	Default	Extra
▶	idTabela_A	int(11)	NO	PRI	NULL	auto_increment
	imie	varchar(45)	YES		NULL	

Zapamiętać znaczenie atrybutów „NOT_NULL” oraz „AUTO_INCREMENT”. Skopiować kod, który został wygenerowany do utworzenia tabel.

2. Za pomocą komendy INSERT INTO wprowadzić następujące dane do tabel:

	idTabela_A	imie
▶	1	Ada
	2	Lola
	3	Stefan
	4	Mateusz
*	NULL	NULL

	idTabela_B	imie
▶	1	Agnieszka
	2	Ada
	3	Mateusz
	4	Wacek
*	NULL	NULL

INSERT INTO Tabela_A (imie) VALUES ('Ada');

3. Porównać działanie klauzul (INNER) JOIN, LEFT JOIN, RIGHT JOIN z warunkiem zgodności imion w obu tabelach.
4. Utworzyć tabelę „Produkty”:

Field	Type	Null	Key	Default	Extra
idprodukt	int(11)	NO	PRI	NULL	auto_increment
produkt	varchar(45)	YES		NULL	
producent	varchar(45)	YES		NULL	

oraz tabelę “Kupujacy”:

Field	Type	Null	Key	Default	Extra
idkupujacy	int(11)	NO	PRI	NULL	auto_increment
idprodukt	int(11)	YES		NULL	
imie	varchar(45)	YES		NULL	
ilosc	int(11)	YES		NULL	

5. Wprowadzić następujące dane do tabel „Kupujący” i „Produkty”:

idkupujący	idprodukty	imie	ilosc	idprodukty	produkt	producent
1	NULL	Karina	NULL	1	buty	CCC
2	1	Magda	2	2	laptop	Acer
3	3	Tola	1	3	telefon	Samsung
4	2	Bolek	1	4	pendrive	Toshiba
5	2	Donek	4	5	spodnie	NULL
6	1	Wacek	4			
7	2	Pola	2			
8	4	Bazyl	1			
9	3	Damian	1			
10	NULL	Iwona	NULL			

6. Wyświetlić dane o tym jaki kupujący kupił jaki produkt i w jakiej ilości.
7. Wyświetlić dane o tym przez kogo dany produkt został zakupiony oraz czy w ogóle został zakupiony.
8. Wyświetlić dane o tym kto kupił produkty danego producenta.
9. Wyświetlić informacje o tych osobach, które kupiły najwięcej produktów danego typu. Kolumny tabeli wynikowej: imie, produkt, ilosc.
10. Utworzyć tabelę „Produkty2” taką jak tabela „Produkty”.

11. Wprowadzić do niej następują dane:

idprodukty	produkt	producent
1	buty	Zara
2	laptop	Lenovo
3	telefon	Nokia
4	dlugopis	Parker
5	olowek	Kohinoor
6	rower	Romet
7	samochod	Syrena

12. Za pomocą operatora „UNION ALL” wyświetlić kolumnę powstałą z połączenia kolumny „produkt” w tabelach „Produkty” i „Produkty2”.
13. Za pomocą operatora „UNION” wyświetlić wszystkie unikalne produkty znajdujące się w tabelach „Produkty” i „Produkty2”.
14. Wyświetlić wszystkie produkty z obu tabel, których nazwa producenta ma więcej niż 5 znaków.

Zapoznaj się z poleceniami:

CREATE PROCEDURE

DELIMITER

DECLARE

SELECT INTO

CALL

SHOW PROCEDURE STATUS

DROP PROCEDURE (IF EXISTS)

15. Utwórz procedurę składowaną wyświetlającą tabelę „gracze”.

```
DELIMITER $$  
CREATE PROCEDURE WyswietlGraczy()  
BEGIN  
    SELECT * FROM gracze;  
END $$  
DELIMITER ;
```

16. Wywołaj powyższą procedurę (poleceniem CALL)

17. Wyświetl przeciętne wagi wszystkich graczy z poszczególnych drużyn.

18. Utwórz procedurę składowaną wyświetlającą przeciętną wagę wszystkich graczy z poszczególnych drużyn.

19. Za pomocą procedury o nazwie „GraczeOPonadprzecietnejWadze()” wyświetl zawodników, których waga jest większa od przeciętnej wagi wszystkich graczy. Przeciętną wagę należy zapisać do zmiennej.

20. Utwórz procedurę „WyswietlWyzszychGraczy()” z parametrem wejściowym „prog_wzrostu” typu INT, która wyświetla zawodników o wzroście większym niż podany.

21. Jaka jest różnica pomiędzy parametrami procedury typu IN, OUT?

22. Napisz procedurę, która w zmiennej sesyjnej zapisuje wzrost najwyższego zawodnika.

23. Wywołaj powyższą procedurę i wyświetlić wynik.