

MySQL DataBase podstawy odc. 1 by Pasja inforamtyki

1. **SELECT * FROM** pytania - (wybierz* z, * - wszystko, wszystkie kolumny z tabeli);
2. **SELECT** tresc, odpa, odpb,... **FROM** pytania (wybierz z bazy * tabelę tresc, odpa, odpb);

kwerendy - zapytania;

AND & OR - spójniki logiczne, w przypadku zapytania o dwa warunki

AND - prawdziwy gdy spełnione są dwa warunki

OR - prawdziwy gdy spełniony jest jeden warunek

ORDER BY - uporządkuj wg...

ASC (ascending) - rosnąco np **SELECT * FROM** pytania **ORDER BY** tresc **ASC**

DESC (descending) - malejąco

BETWEEN - pomiędzy

LIKE - by SQL podobny do, taki sam

LIMIT – ograniczenie ilości do usuwanych rekordów

Zadania.

1. **SELECT *FROM** pytania **WHERE** id=15;

2.

.

.

.

7. Treści pytań o numerach 10-12: **SELECT** id, tresc **FROM** pytania **WHERE**

id>=10 **AND** id<=12 lub alternatywnie **SELECT** id, tresc **FROM** pytania **WHERE**

id **BETWEEN** 10 **AND** 12

8. Pytania zaczynające się od słów "Jak": **SELECT * FROM** pytania **WHERE** tresc **LIKE** "Jak%"

9. Pytania zawierające frazę "C++": **SELECT * FROM** pytania **WHERE** tresc **LIKE** "%C++%"

OR odpa LIKE "%C++%" OR odpb LIKE "%C++%"

OR odpc LIKE "%C++%" OR odpd LIKE "%C++%"

10. Pytania z programowania oraz systemów operacyjnych i sieci z roku 2012:

SELECT * FROM pytania WHERE (kategoria = "programowanie" OR

kategoria = "systemy operacyjne i sieci") AND

rok="2012"

MySQL odc. 2: *Złożone zapytania SELECT* by Pasja informatyki

Księgarnia online

Książki ułożone alfabetycznie wg tytułu:

SELECT * FROM ksiazki ORDER BY tytuł ASC

Najdroższą książkę w bazie:

SELECT * FROM ksiazki ORDER BY cena DESC LIMIT 1

LIMIT 1 – ogranicz do jednej najdroższej

Wszystkie wysłane zamówienia:

SELECT * FROM zamowienia WHERE status = "wyslano"

Wszystkich klientów o nazwisku Rutkowski

SELECT * FROM klienci WHERE nazwisko = "Rutkowski"

Książki zawierające wyrażenie "PHP" w tytule

SELECT * FROM ksiazki WHERE tytuł LIKE "%PHP%"

Zamówienia ułożone od ostatnio dokonanego

SELECT * FROM zamowienia ORDER BY data DESC

Złożone zapytania SELECT

Zapytania związane z łączeniem tabel:

7. Wyjmij dla wszystkich zamówień: imię i nazwisko klienta zamawiającego, id zamówienia, datę zamówienia:

*Ważne!: Po klauzuli **WHERE** musimy wypisać wszystkie relacje zachodzące pomiędzy używanymi przez nas w zapytaniu złożonymi tabelami!*

```
SELECT klienci.imie, klienci.nazwisko, zamowienia.idzamowienia, zamowienia.data FROM  
klienci, zamowienia WHERE klienci.idklienta = zamowienia.idklienta
```

lub zapis z użyciem **ALIASów**, gdzie zamiast pełnej nazwy klienci lub zamówienie wpisujemy k. lub z., ale z użyciem AS z angielskiego „jako”, wg poniższego przykładu:

```
SELECT k.imie, k.nazwisko, z.idzamowienia, z.data FROM klienci AS k, zamowienia AS z  
WHERE k.idklienta = z.idklienta
```

8. Imiona i nazwiska osób, które zamówiły kiedykolwiek książkę nr 2:

```
SELECT k.imie, k.nazwisko FROM klienci AS k, zamowienia AS z WHERE z.idksiazki = 2  
AND z.idklienta = k.idklienta
```

9. Jakie książki (tytuł, autor) zamówiła osoba: Jan Nowak?

```
SELECT k.tytul, k.imieautora, k.nazwiskoautora FROM ksiazki AS k, zamowienia AS z  
WHERE z.idklienta = 2 AND z.idksiazki = k.idksiazki
```

10. Zamówienia dokonane przez osoby o nazwisku Rutkowski ułożone wg daty od najpóźniej dokonanych (imię i nazwisko osoby zamawiającej, id, datę i status zamówienia, tytuł zamówionej książki)

```
SELECT k.imie, k.nazwisko, z.idzamowienia, z.status, z.data, ks.tytul FROM klienci AS k,  
zamowienia AS z, ksiazki AS ks WHERE k.nazwisko = "Rutkowski" AND z.idksiazki =  
ks.idksiazki AND k.idklienta = z.idklienta ORDER BY z.data DESC
```

MySQL odc. 3: **Zapytania INSERT i UPDATE** by Pasja informatyki

Zmień **Zapytania INSERT**

Zaktualizowanie danych klientów w tabeli klienci:

UPDATE klienci SET nazwisko="Psikuta" WHERE idklienta=4

Zmiana klienta nr 3 na wartość 1 (przykład złego zapytania):

UPDATE klienci SET idklienta=1 WHERE idklienta=3

Zwiększ cenę wszystkich książek zgromadzonych w bazie o 10% (z zaokrągleniem):

UPDATE ksiazki SET cena=ROUND(cena*1.1,2)

Zmniejsz cenę najdroższej książki w bazie o 10 zł.:

UPDATE ksiazki SET cena=cena-10 ORDER BY cena DESC LIMIT 1

Zmień imię i nazwisko klientki Anny Kareniny na Joanna Dostojewska:

UPDATE klienci SET imie="Joanna", nazwisko = "Dostojewska" WHERE idklienta=10

*Ważne!: Przed klauzulą **WHERE** wstawiamy przecinki, pod klauzurze **WHERE** wstawiamy spójniki **AND** lub **OR***

Zmień status zamówień nr 4 i 5 (DB księgarnia) na „wysłano”:

UPDATE zamowienia SET status = "wyslano" WHERE idzamowienia BETWEEN 4 AND 5

Zapytania INSERT

Dodaj nowego klienta do bazy: Franciszek Janowski z Chorzowa

INSERT INTO klienci VALUES (NULL, "Franciszek", "Janowski", "Chorzów")

Dodaj do bazy nowe zamówienie: np Artur Rutkowski kupił książkę „HTML5. Tworzenie witryn”:

INSERT INTO zamowienia(idzamowienia, data, status, idklienta, idksiazki) VALUES (NULL, 2019-04-18, "oczekiwanie", 7, 3)

Wstaw do bazy książkę o tytule „Symfonia C++”, autor o nazwisku Grębosz, ale nie wstawiaj jeszcze imienia autora ani książki:

```
INSERT INTO ksiazki (idksiazki, nazwiskoautora, tytul) VALUES (NULL, "Grębosz", "Symfonia C++")
```

Dodaj dwóch nowych klientów za pomocą jednego zapytania:

```
INSERT INTO klienci VALUES(NULL, "Marek", "Mrugalski", "Jelenia Góra"), (NULL, "Aleksandra", "Monroe", "Gdańsk")
```

Wstaw nową osobę do bazy używając alternatywnego zapisu z klauzulą SET:

```
INSERT INTO klienci SET idklienta=NULL, imie="Anna", nazwisko = "Bugalska", miejscowosc = "Wrocław"
```

MySQL odc. 4: **Modele danych. Relacja. Iloczyn kartezjański** by Pasja informatyki

Model danych – integralny zbiór zasad, opisujący dane i powiązania pomiędzy nimi oraz określający ograniczenia nakładane tak na dane, jak i na operacje wykonywane.

Model musi spełniać 3 role:

- **rola strukturalna** – określenie sposobu przechowywania danych (w sposób uporządkowany);
- **rola integralna** – zagwarantowanie stabilności systemu przechowywanych danych;
- **rola manipulacyjna** – zapewnienie możliwości bezpiecznej modyfikacji danych oraz ewentualnych zmian struktury danych;

Modele danych:

- model jednorodny (wada: redundacja – nadmiarowość);
- model jednorodny (hierarchiczna struktura, brak elastyczności rekordów nadrzędne i podrzędne);
- model obiektowy – oparty na podejściu programowania obiektowego,

rzeczywistość reprezentowana jest przez obiekty

OQL – object query language – obiektowy język zapytań

W SQL **klasę nazywamy encją** (entity) z zestawem atrybutów i metod obiektu, natomiast **obiekt nazywamy instancją** (instance), który jest jednym z reprezentantów (przed

- sieciowy model danych – modyfikacja modelu hierarchicznego, oparta na typach kolekcji oraz typach rekordów, zaś reprezentacja powiązań przyjmowała postać tzw. grafu zorientowanego, nazywanego siecią;
- model relacyjno-obiektowy – model mieszany, w którym mechanizm przechowywania danych jest relacyjny;
- model relacyjny – fundamentem jest matematyczne pojęcie relacji, iloczyn kartezjański (układ osi x i y, $A \times B$)

MySQL odc. 5: **Usuwanie danych: DELETE, TRUNCATE, DROP** by Pasja informatyki

"Usunięcie" zawartości **POLA** tabeli (UPDATE zmiany zawartości):

UPDATE klienci SET nazwisko = " " WHERE idklienta = 3

Kasowanie jednego lub wielu rekordów tabeli = DELETE;

Skasowanie wszystkich rekordów tabeli = TRUNCATE (z ang. ściąć, uciąć, okroić);

Kasowanie STRUKTURY w bazie danych = DROP

Usunięcie zawartości jednego rekordu z tabeli zamówienia z id=2

DELETE*FROM zamówienia WHERE idzamowienia =2

Ważne!!: Brak klauzuli **WHERE** oznacza usunięcie **WSZYSTKICH** rekordów z wybranej tabeli!
Struktura pozostaje.

*Przed wysłaniem do bazy jakichkolwiek nietestowanych zapytań **DELETE, TRUNCATE lub DROP** najlepiej jest wykonać backup (eksport) bazy – w razie wystąpienia problemów z usunięciem niewłaściwych rekordów bądź struktur, możemy odtworzyć zniszczone dane.*

Zapytanie **TRUNCATE** jest szybsze od zapytania **DELETE** bez użycia **WHERE**, ponieważ nie kasuje rekordów kolejno jak **DELETE**, lecz zamiast tego tabela najpierw jest „zdropowana” (usunięta) z bazy, a potem odtworzona na nowo (zapytanie **CREATE**).

TRUNCLE TABLE nazwa_tabeli – usuwanie rekordów w tabeli

DELETE FROM klienci ORDER BY idklienta DESC LIMIT 3 – skasuj dane (rekordy) trzech ostatnio zarejestrowanych klientów

DROP – zrzucić, upuścić,

IF EXISTS – jeżeli istnieje

Np.

DROP TABLE, DROP DATABASE,

DROP INDEX, DROP TABLESPACE

DROP TABLE IF EXISTS nazwa_tabeli