

## Bazy Danych

### Ćwiczenie 6: Podstawy języka SQL, część 1, polecenie SELECT

opracował: dr hab. inż. Artur Gramacki (a.gramacki@issi.uz.zgora.pl)

1. Poruszane zagadnienia: polecenie `SELECT`. klauzule `ORDER BY` oraz `WHERE`, operatory, aliasy, wyrażenia, wartości puste `NULL`
2. Większość ćwiczeń z języka SQL będzie operowała na demonstracyjnym modelu, który każdy student musi samodzielnie zainstalować w swoim serwerze MySQL. Na początku należy więc zainstalować przykładową strukturę relacyjną (10 tabel) wraz z przykładowymi danymi. Postaraj się zrozumieć jakie zależności występują w tej bazie, jaki fragment rzeczywistości został zamodelowany itd. Ewentualnie przedyskutuj te zagadnienia z prowadzącym zajęcia.

Do instrukcji dołączone są dwa pliki. W pierwszym (*model\_demonstracyjny.pdf*) pokazano schematycznie rysunek przykładowej struktury relacyjnej, w drugim natomiast (*model\_demonstracyjny.sql*) znajdziesz wszystkie niezbędne polecenia SQL-a, które są potrzebne aby ten model wprowadzić do MySQL-a oraz zapełnić go przykładowymi danymi.

Podłącz się więc do serwera SQL (na konto użytkownika inne niż *root* - z powodów, o których mowa w pierwszej instrukcji) i wykonaj skrypt *model\_demonstracyjny.sql*. Jeżeli omyłkowo wykonamy go jako użytkownik *root* to generalnie nic złego się nie stanie, jednak należy dbać o to, aby konto administratora nie były zaśmiecanie niepotrzebnymi danymi.

O poprawnym wykonywaniu się skryptu świadczy brak jakichkolwiek komunikatów o błędach. Ekran powinien wyglądać bardzo podobnie do pokazanego niżej (aby wydruk nie był zbyt długi usunięto pewne powtarzające się elementy):

```
mysql> source C:\temp\model_demonstracyjny.sql
Query OK, 0 rows affected (0.04 sec)
Query OK, 0 rows affected (0.03 sec)
Query OK, 0 rows affected (0.04 sec)
...
Query OK, 0 rows affected (0.15 sec) Records: 0 Duplicates: 0 Warnings: 0
Query OK, 0 rows affected (0.17 sec) Records: 0 Duplicates: 0 Warnings: 0
...
Query OK, 1 row affected (0.03 sec)
Query OK, 1 row affected (0.03 sec)
...
Query OK, 1 row affected (0.02 sec)
Query OK, 0 rows affected (0.00 sec)
```

Po załadowaniu skryptu zapoznaj się z obiektami, które zostały utworzone. Jeżeli nie pamiętasz jak to się robi, zaglądnij do wcześniejszych instrukcji i/lub literatury. W ostateczności spytaj się prowadzącego laboratorium. Powinniśmy otrzymać wynik jak poniżej:

```
+-----+
| Tables_in_lab |
+-----+
| customer      |
| dept          |
| emp           |
| inventory     |
| item          |
| ord           |
| product       |
| region        |
| title         |
| warehouse     |
+-----+
10 rows in set (0.13 sec)
```

Szczegóły budowy jednej przykładowej tabeli (*emp*) pokazano poniżej. Postaraj się zrozumieć tą tabelkę a w razie problemów ... patrz wyżej:

```
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id             | int(11)       | NO   | PRI | NULL    | auto_increment |
| last_name      | varchar(25)   | NO   |     |         |                 |
| first_name     | varchar(25)   | YES  |     | NULL    |                 |
| userid         | varchar(8)    | YES  | MUL | NULL    |                 |
| start_date     | datetime     | YES  |     | NULL    |                 |
| comments       | varchar(255)  | YES  |     | NULL    |                 |
| manager_id     | int(11)       | YES  | MUL | NULL    |                 |
| title          | varchar(25)   | YES  | MUL | NULL    |                 |
| dept_id        | int(11)       | YES  | MUL | NULL    |                 |
| salary         | decimal(11,2) | YES  |     | NULL    |                 |
| commission_pct | decimal(4,2)  | YES  |     | NULL    |                 |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.94 sec)
```

Kiedy mamy już pewność, że model bazy załadował się poprawnie możemy rozglądnąć się po zawartości tabel. Używamy do tego oczywiście polecenie `SELECT` z języka SQL (patrz poprzednie instrukcje). Przykładowo zawartość tabeli *dept* pokazano poniżej.

```
+-----+-----+-----+
| id | name           | region_id |
+-----+-----+-----+
| 50 | Administration | 1         |
| 10 | Finance        | 1         |
| 41 | Operations     | 1         |
| 42 | Operations     | 2         |
| 43 | Operations     | 3         |
| 44 | Operations     | 4         |
| 45 | Operations     | 5         |
```

```

| 31 | Sales          |          1 |
| 32 | Sales          |          2 |
| 33 | Sales          |          3 |
| 34 | Sales          |          4 |
| 35 | Sales          |          5 |
+-----+-----+-----+
12 rows in set (0.01 sec)

```

3. W tym punkcie należy samodzielnie napisać kilkanaście poleceń w języku SQL. Jeżeli będziesz uzyskiwał na ekranie inne wyniki niż te prezentowane w instrukcji, najprawdopodobniej będzie to oznaczało, że twoje zapytanie SQL jest błędne (chodzi o błędy logiczne, bo błędy składniowe zostaną natychmiast wykryte przez serwer MySQL).

Warunkiem uzyskania w ćwiczeniu takich jak poniżej wyników jest praca na demonstracyjnym schemacie relacyjnym, o którym mowa powyżej. Dlatego też należy upewnić się, że odpowiedni skrypt wykonał się bezbłędnie (poprawnie utworzyły się wszystkie tabele oraz załadowały się wszystkie dane). Dla pewności warto przed rozpoczęciem ćwiczenia po prostu utworzyć schemat od nowa, nawet jeżeli wydaje ci się, że ten, który masz aktualnie zainstalowany jest poprawny.

Poniżej zamieszczone zestawienie ilości rekordów w każdej z tabel<sup>1</sup>. W bazie tej przechowywane są różne informacje na temat innych baz. Jest to więc swego rodzaju baza *metadanych*. Ten prosty test poprawności danych należy jednak potraktować bardzo ostrożnie, gdyż pokazuje on tylko ile jest rekordów w poszczególnych tabelach a nie czy są one zgodne z oryginałem.

```

mysql> SELECT table_name, table_rows
      -> FROM information_schema.tables
      -> WHERE table_schema='lab';
+-----+-----+-----+
| table_name | table_rows |
+-----+-----+-----+
| customer  |          15 |
| dept      |          12 |
| emp       |          25 |
| inventory |         114 |
| item      |          62 |
| ord       |          16 |
| product   |          33 |
| region    |           5 |
| title     |           8 |
| warehouse |           5 |
+-----+-----+-----+
10 rows in set (0.02 sec)

```

Ilość rekordów można łatwo sprawdzić wydając dla każdej tabeli polecenie:

```

mysql> SELECT COUNT(*) FROM emp;
+-----+
| COUNT(*) |
+-----+
|         25 |
+-----+
1 row in set (0.03 sec)

```

---

<sup>1</sup> Korzystamy tutaj ze specjalnej „systemowej” bazy danych o nazwie *information\_schema*.

### Zadanie 1

Wyświetlić dane o pracownikach (tabela *emp*). Zwrócić uwagę na to w jaki sposób wyświetlana jest kolumna *start\_date*. Jako wynik powinniśmy otrzymać:

```
+-----+-----+-----+-----+
| first_name | last_name   | start_date       | salary |
+-----+-----+-----+-----+
| Carmen     | Velasquez   | 1990-03-03 00:00:00 | 2500.00 |
| LaDoris    | Ngao        | 1990-03-08 00:00:00 | 1450.00 |
| Midori     | Nagayama    | 1991-06-17 00:00:00 | 1400.00 |
| Mark       | Quick-To-See | 1990-04-07 00:00:00 | 1450.00 |
| Audry      | Ropeburn    | 1990-03-04 00:00:00 | 1550.00 |
| Molly      | Urguhart    | 1991-01-18 00:00:00 | 1200.00 |
| Roberta    | Menchu      | 1990-05-14 00:00:00 | 1250.00 |
| Ben        | Biri        | 1990-04-07 00:00:00 | 1100.00 |
| Antoinette | Catchpole   | 1992-02-09 00:00:00 | 1300.00 |
| Marta      | Havel       | 1991-02-27 00:00:00 | 1307.00 |
| Colin      | Magee       | 1990-05-14 00:00:00 | 1400.00 |
| Henry      | Giljum      | 1992-01-18 00:00:00 | 1490.00 |
| Yasmin     | Sedeghi     | 1991-02-08 00:00:00 | 1515.00 |
| Mai        | Nguyen      | 1992-01-22 00:00:00 | 1525.00 |
| Andre      | Dumas       | 1991-10-09 00:00:00 | 1450.00 |
| Elena      | Maduro      | 1992-02-07 00:00:00 | 1400.00 |
| George     | Smith       | 1990-03-08 00:00:00 | 940.00  |
| Akira      | Nozaki      | 1991-02-09 00:00:00 | 1200.00 |
| Vikram     | Patel       | 1991-08-06 00:00:00 | 795.00  |
| Chad       | Newman      | 1991-07-21 00:00:00 | 750.00  |
| Alexander  | Markarian   | 1991-05-26 00:00:00 | 850.00  |
| Eddie      | Chang       | 1990-11-30 00:00:00 | 800.00  |
| Radha      | Patel       | 1990-10-17 00:00:00 | 795.00  |
| Bela       | Dancs       | 1991-03-17 00:00:00 | 860.00  |
| Sylvie     | Schwartz    | 1991-05-09 00:00:00 | 1100.00 |
+-----+-----+-----+-----+
25 rows in set (0.13 sec)
```

### Zadanie 2

Wyświetlić dane o pracownikach (tabela *emp*). Zwrócić uwagę na trochę być może sztuczny i na siłę' układ i zawartość kolumn. Użyć funkcji scalającej napisy CONCAT. Jako wynik powinniśmy otrzymać:

```
+-----+-----+-----+
| last_name   | last_name   | first_name & last_name |
+-----+-----+-----+
| Velasquez   | Velasquez   | Carmen<->Velasquez   |
| Ngao        | Ngao        | LaDoris<->Ngao        |
| Nagayama    | Nagayama    | Midori<->Nagayama     |
| Quick-To-See | Quick-To-See | Mark<->Quick-To-See   |
| Ropeburn    | Ropeburn    | Audry<->Ropeburn     |
| Urguhart    | Urguhart    | Molly<->Urguhart     |
| Menchu      | Menchu      | Roberta<->Menchu     |
| Biri        | Biri        | Ben<->Biri           |
| Catchpole   | Catchpole   | Antoinette<->Catchpole |
| Havel       | Havel       | Marta<->Havel        |
| Magee       | Magee       | Colin<->Magee        |
+-----+-----+-----+
```

```

| Giljum      | Giljum      | Henry<->Giljum      |
| Sedeghi    | Sedeghi    | Yasmin<->Sedeghi   |
| Nguyen     | Nguyen     | Mai<->Nguyen       |
| Dumas      | Dumas      | Andre<->Dumas      |
| Maduro     | Maduro     | Elena<->Maduro     |
| Smith      | Smith      | George<->Smith     |
| Nozaki     | Nozaki     | Akira<->Nozaki    |
| Patel      | Patel      | Vikram<->Patel     |
| Newman     | Newman     | Chad<->Newman     |
| Markarian  | Markarian  | Alexander<->Markarian |
| Chang      | Chang      | Eddie<->Chang     |
| Patel      | Patel      | Radha<->Patel     |
| Dancs      | Dancs      | Bela<->Dancs      |
| Schwartz   | Schwartz   | Sylvie<->Schwartz  |
+-----+-----+-----+
25 rows in set (0.01 sec)

```

### Zadanie 3

Wyświetlić dane o pracownikach (tabela *emp*). Tym razem wybieramy tylko pracowników zarabiających powyżej 1500 i wynik sortujemy wg. nazwisk. Jako wynik powinniśmy otrzymać:

```

+-----+-----+-----+-----+
| first_name | last_name | start_date          | salary |
+-----+-----+-----+-----+
| Mai        | Nguyen   | 1992-01-22 00:00:00 | 1525.00 |
| Audry     | Ropeburn | 1990-03-04 00:00:00 | 1550.00 |
| Yasmin    | Sedeghi  | 1991-02-08 00:00:00 | 1515.00 |
| Carmen    | Velasquez | 1990-03-03 00:00:00 | 2500.00 |
+-----+-----+-----+-----+
4 rows in set (0.05 sec)

```

### Zadanie 4

Wyświetlić dane o pracownikach (tabela *emp*). Tym razem wybieramy tylko pracowników, których zarobki mieszczą się w przedziale od 1500 do 2500 (włącznie) i wynik sortujemy wg. Tych zarobków. Na ile różnych sposobów można wykonać to polecenie? Jako wynik powinniśmy otrzymać:

```

+-----+-----+-----+-----+
| first_name | last_name | start_date          | salary |
+-----+-----+-----+-----+
| Yasmin    | Sedeghi  | 1991-02-08 00:00:00 | 1515.00 |
| Mai        | Nguyen   | 1992-01-22 00:00:00 | 1525.00 |
| Audry     | Ropeburn | 1990-03-04 00:00:00 | 1550.00 |
| Carmen    | Velasquez | 1990-03-03 00:00:00 | 2500.00 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

### Zadanie 5

Wyświetlamy dane jak w poprzednim punkcie, ale zmieniamy etykiety. Zwrócić uwagę na wielkość liter w nazwach etykiet. Jako wynik powinniśmy otrzymać:

```

+-----+-----+-----+-----+
| IMIE      | nazwisko  | data zatrudnienia  | ZaRoBkI |
+-----+-----+-----+-----+

```

```

| Yasmin | Sedeghi | 1991-02-08 00:00:00 | 1515.00 |
| Mai | Nguyen | 1992-01-22 00:00:00 | 1525.00 |
| Audry | Ropeburn | 1990-03-04 00:00:00 | 1550.00 |
| Carmen | Velasquez | 1990-03-03 00:00:00 | 2500.00 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

### Zadanie 6

Wyświetlić dane o pracownikach (tabela *emp*). Tym razem wybieramy tylko pracowników, którzy pracują w dziale o numerze 41 (kolumna *dept\_id*). Dodatkowo wyświetlamy kwotę zarobków pomnożoną przez 12. Jak można zmienić nazwę etykiety *salary\*12*? Jako wynik powinniśmy otrzymać:

```

+-----+-----+-----+-----+
| first_name | last_name | salary | salary*12 |
+-----+-----+-----+-----+
| LaDoris | Ngao | 1450.00 | 17400.00 |
| Molly | Urguhart | 1200.00 | 14400.00 |
| Elena | Maduro | 1400.00 | 16800.00 |
| George | Smith | 940.00 | 11280.00 |
+-----+-----+-----+-----+
4 rows in set (0.06 sec)

```

### Zadanie 7

Wyświetlić dane o pracownikach (tabela *emp*). Wybrać tylko tych pracowników, którzy zostali zatrudnieni po 31-12-1991 (kolumna *start\_date*). Wynik posortować wg. daty zatrudnienia. Zwrócić uwagę na format podawanej daty w klauzuli *WHERE*. Jako wynik powinniśmy otrzymać:

```

+-----+-----+-----+
| first_name | last_name | start_date |
+-----+-----+-----+
| Henry | Giljum | 1992-01-18 00:00:00 |
| Mai | Nguyen | 1992-01-22 00:00:00 |
| Elena | Maduro | 1992-02-07 00:00:00 |
| Antoinette | Catchpole | 1992-02-09 00:00:00 |
+-----+-----+-----+
4 rows in set (0.00 sec)

```

### Zadanie 8

Wyświetlić bieżącą datę oraz bieżący czas systemowy. Jako wynik powinniśmy otrzymać:

```

+-----+-----+-----+
| current_date | current_time | current_timestamp |
+-----+-----+-----+
| 2006-01-30 | 23:20:32 | 2006-01-30 23:20:32 |
+-----+-----+-----+
1 row in set (0.00 sec)

```

### Zadanie 9

Wyświetlić bieżącą datę oraz bieżący czas systemowy. W drugiej kolumnie specyficznie sformatować wynik (użyć funkcji *DATE\_FORMAT*). Jako wynik powinniśmy otrzymać:

```

+-----+-----+
| current_timestamp | biezaca data i godzina |
+-----+-----+
| 2006-01-30 23:55:58 | Monday :: January :: 30 :: 2006 :: 23:55:58 |
+-----+-----+
1 row in set (0.00 sec)

```

### Zadanie 10

Wyświetlić dane o pracownikach. Wyświetlić datę zatrudnienia (kolumna *start\_date*) oraz dodatkowo informację o tym, ile dni są już zatrudnieni. Wynik posortować wg. ilości tych dni. Użyć funkcji systemowej *DATEDIFF*. Jako wynik powinniśmy otrzymać:

```

+-----+-----+-----+
| last_name | start_date | ilosc dni |
+-----+-----+-----+
| Velasquez | 1990-03-03 00:00:00 | 5812 |
| Ropeburn | 1990-03-04 00:00:00 | 5811 |
| Ngao | 1990-03-08 00:00:00 | 5807 |
| Smith | 1990-03-08 00:00:00 | 5807 |
| Quick-To-See | 1990-04-07 00:00:00 | 5777 |
| Biri | 1990-04-07 00:00:00 | 5777 |
| Magee | 1990-05-14 00:00:00 | 5740 |
| Menchu | 1990-05-14 00:00:00 | 5740 |
| Patel | 1990-10-17 00:00:00 | 5584 |
| Chang | 1990-11-30 00:00:00 | 5540 |
| Urguhart | 1991-01-18 00:00:00 | 5491 |
| Sedeghi | 1991-02-08 00:00:00 | 5470 |
| Nozaki | 1991-02-09 00:00:00 | 5469 |
| Havel | 1991-02-27 00:00:00 | 5451 |
| Dancs | 1991-03-17 00:00:00 | 5433 |
| Schwartz | 1991-05-09 00:00:00 | 5380 |
| Markarian | 1991-05-26 00:00:00 | 5363 |
| Nagayama | 1991-06-17 00:00:00 | 5341 |
| Newman | 1991-07-21 00:00:00 | 5307 |
| Patel | 1991-08-06 00:00:00 | 5291 |
| Dumas | 1991-10-09 00:00:00 | 5227 |
| Giljum | 1992-01-18 00:00:00 | 5126 |
| Nguyen | 1992-01-22 00:00:00 | 5122 |
| Maduro | 1992-02-07 00:00:00 | 5106 |
| Catchpole | 1992-02-09 00:00:00 | 5104 |
+-----+-----+-----+
25 rows in set (0.00 sec)

```

### Zadanie 11

Wyświetlić dane o pracownikach posortowane wg. nazwisk. Wyświetlić tylko pięć kolejnych rekordów poczynając od czwartego rekordu licząc od góry (użyć klauzuli *LIMIT*). Jako wynik powinniśmy otrzymać:

```

+-----+-----+
| first_name | last_name |
+-----+-----+
| Bela | Dancs |
| Andre | Dumas |
| Henry | Giljum |
| Marta | Havel |

```

```
| Elena      | Maduro    |
+-----+-----+
5 rows in set (0.00 sec)
```

### Zadanie 12

Wyświetlić dane z tabeli produktów (tabela *product*). Wyświetlić tylko te rekordy, gdzie nazwa produktu (kolumna *name*) rozpoczyna się na literę A, B lub C. Wynik dodatkowo posortować malejąco (od Z do A). Jako wynik powinniśmy otrzymać:

```
+-----+
| name                |
+-----+
| Chapman Helmet     |
| Cabrera Bat        |
| Bunny Ski Pole     |
| Bunny Boot         |
| Black Hawk Knee Pads |
| Black Hawk Elbow Pads |
| Alomar Glove       |
| Alexeyer Pro Lifting Bar |
| Ace Ski Pole       |
| Ace Ski Boot       |
+-----+
10 rows in set (0.00 sec)
```

### Zadanie 13

Wyświetlić dane z tabeli pracowników (tabela *emp*). Wyświetlić tylko te rekordy, gdzie wielkość prowizji (kolumna *commision\_pct*) jest różna od wartości *NULL*. Jako wynik powinniśmy otrzymać:

```
+-----+-----+-----+
| first_name | last_name | COMMISSION_PCT |
+-----+-----+-----+
| Colin      | Magee     | 10.00          |
| Henry      | Giljum    | 12.50          |
| Yasmin     | Sedeghi   | 10.00          |
| Mai        | Nguyen    | 15.00          |
| Andre      | Dumas     | 17.50          |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

### Zadanie 14

Wyświetlić dane z tabeli oddziałów (tabela *dept*). Wynik posortować wg. nazw oddziałów oraz wg. numerów regionów (kolumna *region\_id*). Jako wynik powinniśmy otrzymać:

```
+----+-----+-----+
| id | name                | region_id |
+----+-----+-----+
| 50 | Administration      | 1         |
| 10 | Finance              | 1         |
| 41 | Operations           | 1         |
| 42 | Operations           | 2         |
| 43 | Operations           | 3         |
| 44 | Operations           | 4         |
```



```

| 45 | Operations      |          5 |
| 31 | Sales           |          1 |
| 32 | Sales           |          2 |
| 33 | Sales           |          3 |
| 34 | Sales           |          4 |
| 35 | Sales           |          5 |
+-----+-----+-----+
12 rows in set (0.00 sec)

```

### Zadanie 15

Wyświetlić dane o klientach (tabela *customer*), których zdolność kredytowa (kolumna *credit\_rating*) oceniana jest na dobrą (wartość w kolumnie: *GOOD*). Zwrócić uwagę na sposób formułowania warunku w klauzuli *WHERE*. Użyć raz operatora *=* a raz operatora *LIKE*. Jako wynik powinniśmy otrzymać:

```

+-----+-----+-----+
| name                | credit_rating |
+-----+-----+-----+
| Delhi Sports        | GOOD         |
| Sweet Rock Sports   | GOOD         |
| Muench Sports       | GOOD         |
+-----+-----+-----+
3 rows in set (0.00 sec)

```

### Zadanie 16

Z tabeli *dept* wyświetlić wszystkie nazwy oddziałów. Usunąć powtarzające się duplikaty (użyć klauzuli *DISTINCT*). Jako wynik powinniśmy otrzymać:

```

+-----+
| name      |
+-----+
| Administration |
| Finance    |
| Operations  |
| Sales      |
+-----+
4 rows in set (0.00 sec)

```