Podstawy baz danych - ćwiczenie 9

Polecenie SELECT cz 9

18,20

Ćwiczenie 1

Użyj dwóch wersji polecenia INSERT aby dodać 2 pracowników (o takim samym nazwisku – np. twoje nazwisko) do tabeli emp. Pierwsza wymaga podawania wartości dla wszystkich pól, nawet gdy są tam wartości NULL. Druga wersja jest wygodniejsza, gdyż wpisujemy wartości tylko do interesujących nas pól. W obu przypadkach system odmówi wstawienia rekordu, gdy zostanie naruszone chociaż jedno z ograniczeń integralnościowych założonych na tabeli. Jako wynik powinniśmy otrzymać:

```
mysql> SELECT first_name, last_name FROM emp WHERE last_name LIKE 'Kowalski';
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| Jan        | Kowalski  |
| Roch       | Kowalski  |
+------------+-----------+
2 rows in set (0.00 sec)
```

Ćwiczenie 2

Za pomocą jednego polecenia INSERT dodaj 3 pracowników. Jako wynik możemy otrzymać:

```
mysql> SELECT id,first_name, last_name FROM emp WHERE last_name LIKE 'Nowak';
+-----+------------+-----------+
| id  | first_name | last_name |
+-----+------------+-----------+
| 103 | Jan        | Nowak     |
| 104 | Karol      | Nowak     |
| 105 | Jerzy      | Nowak     |
+-----+------------+-----------+
3 rows in set (0.00 sec)
```

Ćwiczenie 3

Dodaj do tabeli region województwo Kujawsko-Pomorskie. Wykorzystaj fakt że dane w kolumnie id są automatycznie inkrementowane. Jako wynik możemy otrzymać:

```
mysql> SELECT * FROM region ORDER BY name;
+----+----------------------+
| id | name                 |
+----+----------------------+
|  3 | Africa / Middle East |
|  4 | Asia                 |
|  5 | Europe               |
|  6 | Kujawsko-Pomorskie   |
|  1 | North America        |
|  2 | South America        |
+----+----------------------+
6 rows in set (0.02 sec)
```

Ćwiczenie 4

Usuń wstawiony wiersz w ćwiczeniu 3, następnie dodaj go ponownie. Zaobserwuj jakie id zostanie „przydzielone” nowowpisanej wartości. Jako wynik możemy otrzymać:

```
mysql> SELECT * FROM region ORDER BY name;
+----+----------------------+
| id | name                 |
+----+----------------------+
|  3 | Africa / Middle East |
|  4 | Asia                 |
|  5 | Europe               |
```

```
|  7 | Kujawsko-Pomorskie   |
|  1 | North America        |
|  2 | South America        |
+----+----------------------+
6 rows in set (0.00 sec)
```

Ćwiczenie 5

Utworzyć tabelę emp2 i dodać do niej wszystkich pracowników, których nazwiska zaczynają się na M,N,O. Jako wynik możemy otrzymać:

```
mysql> SELECT * FROM emp2;
+-----------+-----------+---------+
| imie      | nazwisko  | zarobki |
+-----------+-----------+---------+
| LaDoris   | Ngao      | 1450.00 |
| Midori    | Nagayama  | 1400.00 |
| Roberta   | Menchu    | 1250.00 |
| Colin     | Magee     | 1400.00 |
| Mai       | Nguyen    | 1525.00 |
| Elena     | Maduro    | 1400.00 |
| Akira     | Nozaki    | 1200.00 |
| Chad      | Newman    |  750.00 |
| Alexander | Markarian |  850.00 |
| Jan       | Nowak     |    NULL |
| Karol     | Nowak     |    NULL |
| Jerzy     | Nowak     |    NULL |
+-----------+-----------+---------+
12 rows in set (0.00 sec)
```

Ćwiczenie 6

Uaktualnić dane pracownika, którego dodaliśmy w ćwiczeniu 1. Jako wynik możemy otrzymać:

```
mysql> UPDATE emp SET salary = 3000, start_date = '2016-04-20' WHERE id=100;
mysql> SELECT id,last_name,first_name,salary FROM emp WHERE id=100;
+-----+-----------+------------+---------+
| id  | last_name | first_name | salary  |
+-----+-----------+------------+---------+
| 100 | Kowalski  | Jan        | 3000.00 |
+-----+-----------+------------+---------+
1 row in set (0.00 sec)
```

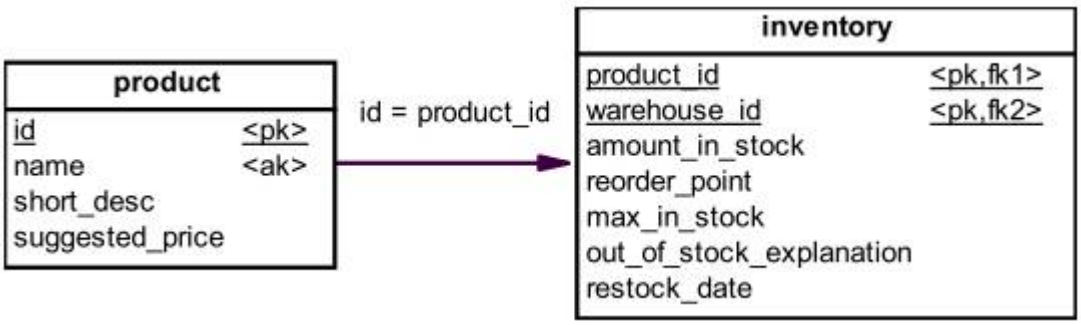
Ćwiczenie 7

Uaktualnić wszystkie rekordy w tabeli emp. W polu comment wpisać login składający się z ciągu ID, nazwiska, pierwszej litery imienia. Jako wynik możemy otrzymać:

```
mysql> SELECT first_name,last_name,comments FROM emp WHERE last_name LIKE 'K%';
+------------+-----------+-------------+
| first_name | last_name | comments    |
+------------+-----------+-------------+
| Jan        | Kowalski  | IDKowalskiJ |
| Roch       | Kowalski  | IDKowalskiR |
+------------+-----------+-------------+
2 rows in set (0.00 sec)
```

Ćwiczenie 8

Obniżyć (jednym zapytaniem SQL- nie ręcznie) cenę tych produktów w bazie, których sprzedano mniej ni» 20 sztuk (różnica wartości w kolumnach max_in_stock i amount_in_stock). Obniżki cen dokonujemy tylko dla produktów z hurtowni w Sao Paolo. Jako wynik możemy otrzymać:



Ceny produktów do uaktualnienia

```
mysql> SELECT name,suggested_price FROM product WHERE id IN( SELECT product_id FROM inventory WHERE max_in_stock - amount_in_stock < 20 AND warehouse_id = (SELECT id FROM warehouse WHERE city = 'Sao Paolo'));
+----------------------+-----------------+
| name                 | suggested_price |
+----------------------+-----------------+
| Black Hawk Knee Pads |            9.00 |
| Black Hawk Elbow Pads |           8.00 |
| Himalaya Tires       |           18.25 |
| Safe-T Helmet        |           60.00 |
| Griffey Glove        |           80.00 |
| Alomar Glove         |           75.00 |
| Winfield Bat         |           50.00 |
+----------------------+-----------------+
7 rows in set (0.00 sec)
```

Ceny produktów po uaktualnieniu (rabat 10%)

```
mysql> SELECT name,suggested_price FROM product WHERE id IN( SELECT product_id FROM inventory WHERE max_in_stock - amount_in_stock < 20 AND warehouse_id = (SELECT id FROM warehouse WHERE city = 'Sao Paolo'));
+----------------------+-----------------+
| name                 | suggested_price |
+----------------------+-----------------+
| Black Hawk Knee Pads |            8.10 |
| Black Hawk Elbow Pads |           7.20 |
| Himalaya Tires       |           16.43 |
| Safe-T Helmet        |           54.00 |
| Griffey Glove        |           72.00 |
| Alomar Glove         |           67.50 |
| Winfield Bat         |           45.00 |
+----------------------+-----------------+
7 rows in set (0.00 sec)
```

Ćwiczenie 9

Przepisać (jednym zapytaniem - nie ręcznie) wszystkich pracowników z wydziału Sales zarabiających ponad 1200 do wydziału Operations zlokalizowanym w regionie North America, zwiększając im jednocześnie płacę o 10% i modyfikując datę zatrudnienia (pole start_date) na bieżącą datę. Pracownicy do przeniesienia:

```
mysql> SELECT id, last_name, salary, dept_id  FROM emp WHERE dept_id IN (SELECT id FROM dept WHERE name = 'Sales') AND salary > 1200;
+----+-----------+---------+---------+
| id | last_name | salary  | dept_id |
+----+-----------+---------+---------+
|  3 | Nagayama  | 1400.00 |      31 |
| 11 | Magee     | 1400.00 |      31 |
```

```
| 12 | Giljum    | 1490.00 |      32 |
| 13 | Sedeghi   | 1515.00 |      33 |
| 14 | Nguyen    | 1525.00 |      34 |
| 15 | Dumas     | 1450.00 |      35 |
+----+-----------+---------+---------+
6 rows in set (0.00 sec)
```

Płace i działy pracowników po przeniesieniu

```
mysql> SELECT id, last_name, salary, dept_id  FROM emp WHERE id IN (3,11,12,13,14,15);
+----+-----------+---------+---------+
| id | last_name | salary  | dept_id |
+----+-----------+---------+---------+
|  3 | Nagayama  | 1540.00 |      41 |
| 11 | Magee     | 1540.00 |      41 |
| 12 | Giljum    | 1639.00 |      41 |
| 13 | Sedeghi   | 1666.50 |      41 |
| 14 | Nguyen    | 1677.50 |      41 |
| 15 | Dumas     | 1595.00 |      41 |
+----+-----------+---------+---------+
6 rows in set (0.03 sec)
```

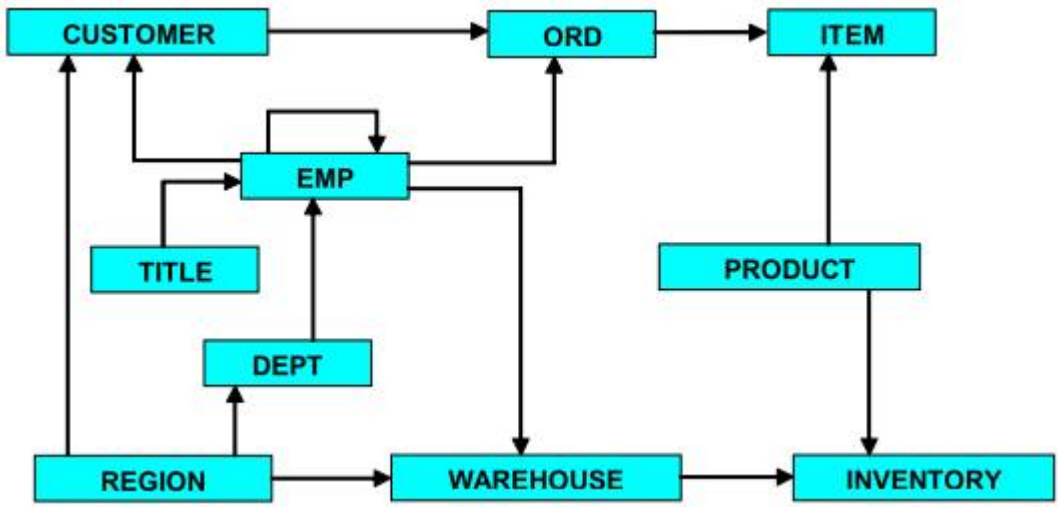
Ćwiczenie 10

Usunąć z bazy pracowników wszystkich o nazwisku Kowalski:

```
mysql> SELECT * FROM emp WHERE last_name LIKE 'Kowalski';
```

Ćwiczenie 11

Wykasować wszystkie tabele




```
DROP TABLE IF EXISTS item;
DROP TABLE IF EXISTS inventory;
DROP TABLE IF EXISTS ord;
DROP TABLE IF EXISTS product;
DROP TABLE IF EXISTS warehouse;
DROP TABLE IF EXISTS customer;
DROP TABLE IF EXISTS emp;
DROP TABLE IF EXISTS dept;
DROP TABLE IF EXISTS region;
DROP TABLE IF EXISTS title;
```

Ćwiczenie 12

Utworzyć tabelę prostą pierwsza tabelę (inaczej: relacja) składa się z czterech kolumn: pierwsza jest typu całkowitego (kolumna uczen_id), kolejne dwie są typu znakowego o zmiennej szerokości (kolumny imie oraz nazwisko), ostatnia kolumna (typ_szkoly_id) jest też typu znakowego, ale o stałej szerokości. Kolumna uczeń_id jest tzw. kluczem głównym oraz jest zdefiniowana jako AUTO_INCREMENT. Wartość tej kolumny przy dodawaniu rekordów może by¢ automatycznie zwiększana . Gdy dodajemy wiersz do tabeli zawierającej kolumnę typu AUTO_INCREMENT, nie podajemy wartości dla tej kolumny, bo odpowiedni numer nada baza danych. Pozwala to na proste i efektywne rozwiązanie problemu generowania unikatowych wartości dla kolejnych wierszy tabeli.

Jako wynik powinniśmy otrzymać:

```
mysql> describe uczniowie;
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| uczen_id     | int(11)     | NO   | PRI | NULL    | auto_increment |
| imie         | varchar(25) | NO   |     | NULL    |                |
| nazwisko     | varchar(30) | NO   |     | NULL    |                |
| typ_szkoly_id | char(1)    | YES  |     | NULL    |                |
+--------------+-------------+------+-----+---------+----------------+
4 rows in set (0.08 sec)
```

Ćwiczenie 13

Utwórz tabelę temp_emp w oparciu o zapytanie. Tabela powinna zawierać imię nazwisko zarobki i nazwę oddziału gdzie jest zatrudniony dany pracownik. Jako wynik możemy otrzymać tabelę:

```
mysql> SELECT * FROM emp_temp;
+------------+--------------+---------+---------------+
| first_name | last_name    | salary  | name          |
+------------+--------------+---------+---------------+
| Carmen     | Velasquez    | 2500.00 | Administration |
| Audry      | Ropeburn     | 1550.00 | Administration |
| Mark       | Quick-To-See | 1450.00 | Finance        |
| LaDoris    | Ngao         | 1450.00 | Operations     |
| Molly      | Urguhart     | 1200.00 | Operations     |
| Elena      | Maduro       | 1400.00 | Operations     |
| George     | Smith        |  940.00 | Operations     |
| Roberta    | Menchu       | 1250.00 | Operations     |
| Akira      | Nozaki       | 1200.00 | Operations     |
| Vikram     | Patel        |  795.00 | Operations     |
| Ben        | Biri         | 1100.00 | Operations     |
| Chad       | Newman       |  750.00 | Operations     |
| Alexander  | Markarian    |  850.00 | Operations     |
| Antoinette | Catchpole    | 1300.00 | Operations     |
| Eddie      | Chang        |  800.00 | Operations     |
| Marta      | Havel        | 1307.00 | Operations     |
| Bela       | Dancs        |  860.00 | Operations     |
| Sylvie     | Schwartz     | 1100.00 | Operations     |
| Midori     | Nagayama     | 1400.00 | Sales          |
| Colin      | Magee        | 1400.00 | Sales          |
| Henry      | Giljum       | 1490.00 | Sales          |
| Yasmin     | Sedeghi      | 1515.00 | Sales          |
| Mai        | Nguyen       | 1525.00 | Sales          |
| Radha      | Patel        |  795.00 | Sales          |
| Andre      | Dumas        | 1450.00 | Sales          |
+------------+--------------+---------+---------------+
25 rows in set (0.00 sec)
```

```
mysql> DESC emp_temp;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| first_name | varchar(25)  | YES  |     | NULL    |       |
| last_name  | varchar(25)  | NO   |     | NULL    |       |
| salary     | decimal(11,2) | YES  |     | NULL    |       |
| name       | varchar(25)  | NO   |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
4 rows in set (0.05 sec)
```

Ćwiczenie 14

Podobnie jak w zadaniu 2 utwórz tabelę temp_emp2, też w oparciu o zapytanie. Dodaj do tabeli region województwo Kujawsko-Pomorskie. Wykorzystaj fakt że dane w kolumnie id są automatycznie inkrementowane. Jako wynik możemy otrzymać:

```
mysql> SELECT * FROM emp_temp2;
+--------------------+--------------------+----------------------------+----------------+
| UPPER(E.first_name) | UPPER(E.last_name) | CONCAT('Zarobki: ',E.salary) | name           |
+--------------------+--------------------+----------------------------+----------------+
| CARMEN             | VELASQUEZ          | Zarobki: 2500.00           | Administration |
| AUDRY              | ROPEBURN           | Zarobki: 1550.00           | Administration |
| MARK               | QUICK-TO-SEE       | Zarobki: 1450.00           | Finance        |
| LADORIS            | NGAO               | Zarobki: 1450.00           | Operations     |
| MOLLY              | URGUHART           | Zarobki: 1200.00           | Operations     |
| ELENA              | MADURO             | Zarobki: 1400.00           | Operations     |
| GEORGE             | SMITH              | Zarobki: 940.00            | Operations     |
| ROBERTA            | MENCHU             | Zarobki: 1250.00           | Operations     |
| AKIRA              | NOZAKI             | Zarobki: 1200.00           | Operations     |
| VIKRAM             | PATEL              | Zarobki: 795.00            | Operations     |
| BEN                | BIRI               | Zarobki: 1100.00           | Operations     |
| CHAD               | NEWMAN             | Zarobki: 750.00            | Operations     |
| ALEXANDER          | MARKARIAN          | Zarobki: 850.00            | Operations     |
| ANTOINETTE         | CATCHPOLE          | Zarobki: 1300.00           | Operations     |
| EDDIE              | CHANG              | Zarobki: 800.00            | Operations     |
| MARTA              | HAVEL              | Zarobki: 1307.00           | Operations     |
| BELA               | DANCS              | Zarobki: 860.00            | Operations     |
| SYLVIE             | SCHWARTZ           | Zarobki: 1100.00           | Operations     |
| MIDORI             | NAGAYAMA           | Zarobki: 1400.00           | Sales          |
| COLIN              | MAGEE              | Zarobki: 1400.00           | Sales          |
| HENRY              | GILJUM             | Zarobki: 1490.00           | Sales          |
| YASMIN             | SEDEGHI            | Zarobki: 1515.00           | Sales          |
| MAI                | NGUYEN             | Zarobki: 1525.00           | Sales          |
| RADHA              | PATEL              | Zarobki: 795.00            | Sales          |
| ANDRE              | DUMAS              | Zarobki: 1450.00           | Sales          |
+--------------------+--------------------+----------------------------+----------------+
25 rows in set (0.00 sec)

mysql> DESC emp_temp2;
+----------------------------+-------------+------+-----+---------+-------+
| Field                      | Type        | Null | Key | Default | Extra |
+----------------------------+-------------+------+-----+---------+-------+
| UPPER(E.first_name)        | varchar(25) | YES  |     | NULL    |       |
| UPPER(E.last_name)         | varchar(25) | NO   |     |         |       |
| CONCAT('Zarobki: ',E.salary) | varchar(22) | YES  |     | NULL    |       |
| name                       | varchar(25) | NO   |     | NULL    |       |
+----------------------------+-------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

Ćwiczenie 15

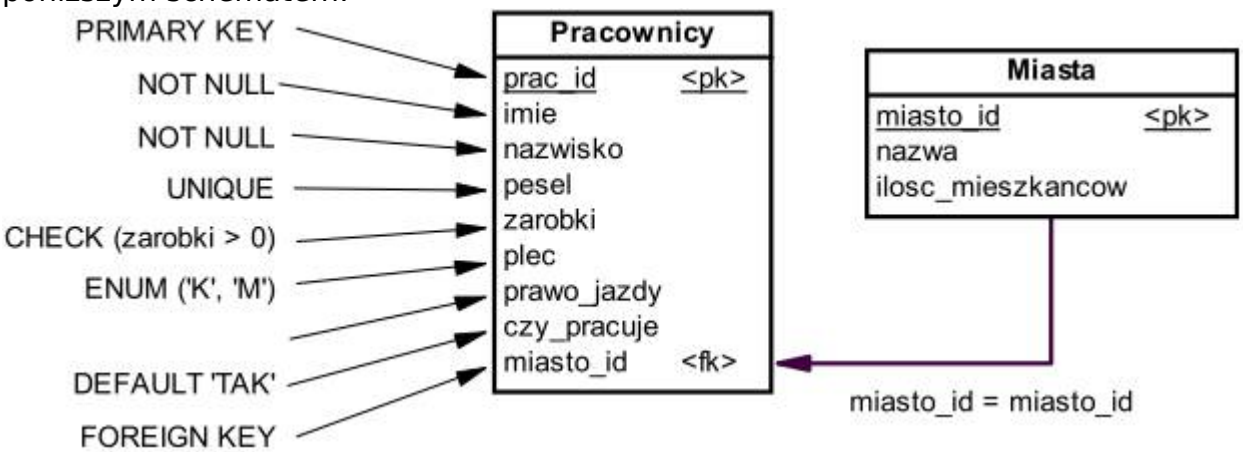
Utwórz widok (VIEW) tabeli emp i dept (first_name, last_name, name). Jako wynik możemy otrzymać:

```
mysql> SELECT * FROM emp_dept_view;
+------------+--------------+----------------+
| first_name | last_name    | name           |
+------------+--------------+----------------+
| Carmen     | Velasquez    | Administration |
| Audry      | Ropeburn     | Administration |
| Mark       | Quick-To-See | Finance        |
| LaDoris    | Ngao         | Operations     |
| Molly      | Urguhart     | Operations     |
| Elena      | Maduro       | Operations     |
| George     | Smith        | Operations     |
| Roberta    | Menchu       | Operations     |
| Akira      | Nozaki       | Operations     |
| Vikram     | Patel        | Operations     |
| Ben        | Biri         | Operations     |
| Chad       | Newman       | Operations     |
| Alexander  | Markarian    | Operations     |
| Antoinette | Catchpole    | Operations     |
| Eddie      | Chang        | Operations     |
| Marta      | Havel        | Operations     |
| Bela       | Dancs        | Operations     |
| Sylvie     | Schwartz     | Operations     |
| Midori     | Nagayama     | Sales          |
| Colin      | Magee        | Sales          |
| Henry      | Giljum       | Sales          |
| Yasmin     | Sedeghi      | Sales          |
| Mai        | Nguyen       | Sales          |
| Radha      | Patel        | Sales          |
| Andre      | Dumas        | Sales          |
+------------+--------------+----------------+
25 rows in set (0.10 sec)

mysql> DESC emp_dept_view;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| first_name | varchar(25) | YES  |     | NULL    |       |
| last_name  | varchar(25) | NO   |     | NULL    |       |
| name       | varchar(25) | NO   |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.03 sec)
```

Ćwiczenie 16

Utworzyć tabelę pracownicy i dodać do niej wszystkie właściwości zgodnie z poniższym schematem:



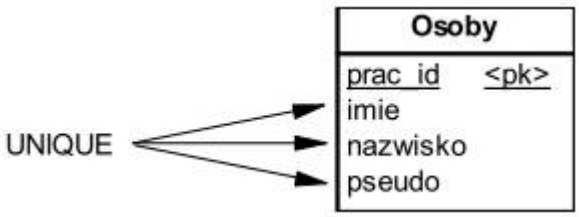
Jako efekt powinniśmy otrzymać:

```
mysql> DESC pracownicy;
+------------+---------------------+------+-----+---------+----------------+
| Field      | Type                | Null | Key | Default | Extra          |
+------------+---------------------+------+-----+---------+----------------+
| prac_id    | int(11)             | NO   | PRI | NULL    | auto_increment |
| imie       | varchar(25)         | NO   |     | NULL    |                |
| nazwisko   | varchar(30)         | NO   |     | NULL    |                |
| pesel      | int(11)             | NO   | UNI | NULL    |                |
| zarobki    | decimal(11,2)       | YES  |     | NULL    |                |
| plec       | enum('M','K')       | NO   |     | NULL    |                |
| prawo_jazdy | set('A','B','C','D') | NO   |     | NULL    |                |
| czy_pracuje | char(3)             | YES  |     | TAK     |                |
| miasto_id  | int(11)             | YES  |     | NULL    |                |
+------------+---------------------+------+-----+---------+----------------+
9 rows in set (0.01 sec)
```

Bardzo często na kolumny w tabeli nakładamy pewne dodatkowe warunki. Przykładowo w kolumnie numerycznej możemy nakazać, aby możliwe było wpisywanie tylko liczb ze zbioru {1, 2, 3, 4}. Podobnie dla np. kolumny tekstowej możemy nakazać wpisywanie tylko i wyłącznie napisów karta płatnicza, gotówka oraz przelew. O kolumnach takich mówimy, że maj¡ one zdefiniowane pewne dodatkowe warunki (nazywane ograniczeniami), które pozwolą nam osiągnąć zamierzony efekt. Nakładanie ograniczeń na kolumny pozwala nam przeprowadza¢ kontrolę wprowadzanych danych na najniższym z możliwych poziomów (na poziomie bazy danych). Oczywiście kontrolę tak¡ można też przeprowadzi¢ na poziomie aplikacji, jednak powinno się traktować jako dobrą zasadę programistyczną aby, jeżeli jest to tylko możliwe, przeprowadzać kontrolę wprowadzanych danych możliwie jak najbliżej serwera bazy danych.
Ograniczenia integralnościowe
NOT NULL - w kolumnie nie można zapisywać wartości NULL (wartość nieznana w tym momencie)
PRIMARY KEY - każda tabela może zawiera¢ tylko jedno takie ograniczenie. Może by¢ zdefiniowane na poziomie jednej kolumnie (tzw. ograniczenie kolumnowe) lub na więcej niż jednej kolumnie (tzw. ograniczenie tablicowe). Zapewnia, że wszystkie wpisane wartości są unikalne i różne od NULL
DEFAULT - określa domyślną wartość używaną podczas wstawiania danych w przypadku, gdy nie zostają jawnie podana żadna wartość dla kolumny
FOREIGN KEY (REFERENCES) - zapewnia tzw. integralność referencyjną. Zapobiega wstawianiu błędnych rekordów w tabelach podrzędnych (po stronie N)
UNIQUE - zapewnia, że wszystkie wpisane wartości są unikalne. Od ograniczenia PRIMARY KEY różni się tym, że dopuszcza wpisywanie wartości NULL
CHECK - pozwala na wpisywanie tylko takich wartości, które spełniają określone warunki (np. zarobki>0). Obecnie w MySQL nie jest zaimplementowane
ENUM - pozwala na wpisanie tylko jednej wartości z wcześniej zdefiniowanego zbioru
SET - pozwala na wpisanie jednej lub wielu wartości z wcześniej zdefiniowanego zbioru.

Ćwiczenie 17

Utworzyć tabelę osoby z ograniczeniem UNIQUE dla pól imie, nazwisko oraz pseudonim.



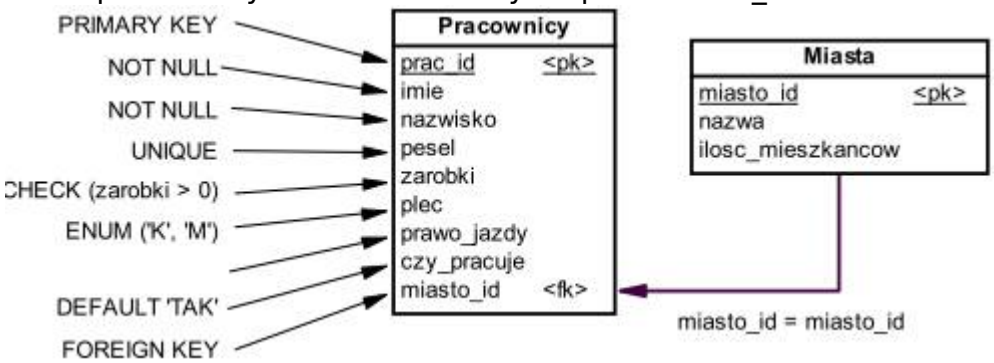
Ćwiczenie 18

Uaktualnić wszystkie rekordy w tabeli emp. W polu comment wpisać login składający się z ciągu ID, nazwiska, pierwszej litery imienia. Jako wynik możemy otrzymać:

```
mysql> UPDATE emp SET comments=CONCAT('ID',last_name,LEFT(first_name,1));
mysql> SELECT first_name,last_name,comments FROM emp WHERE last_name LIKE 'K%';
+------------+-----------+-------------+
| first_name | last_name | comments    |
+------------+-----------+-------------+
| Jan        | Kowalski  | IDKowalskiJ |
| Roch       | Kowalski  | IDKowalskiR |
+------------+-----------+-------------+
2 rows in set (0.00 sec)
```

Ćwiczenie 19

Dodać tabelę miasta (zgodnie ze schematem przedstawionym na rysunku poniżej). W tabeli pracownicy dodać klucz obcy do pola miasto_id.




```
mysql> DESC miasta;
+-------------------+-------------+------+-----+---------+-------+
| Field             | Type        | Null | Key | Default | Extra |
+-------------------+-------------+------+-----+---------+-------+
| miasto_id         | int(11)     | NO   | PRI | NULL    |       |
| nazwa             | varchar(50) | NO   |     | NULL    |       |
| liczba_mieszkancow | int(11)    | YES  |     | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

W tabeli pracownicy dodać klucz obcy do pola miasto_id.

Aby przekona¢ się, że wszystko utworzyło się po naszej myśli możemy wyda¢ poniższe polecenie:

```
mysql> SELECT constraint_name, table_schema, table_name, constraint_type FROM information_schema.table_constraints WHERE table_schema = 'test' and table_name='pracownicy';
+-----------------+--------------+------------+-----------------+
| constraint_name | table_schema | table_name | constraint_type |
+-----------------+--------------+------------+-----------------+
| PRIMARY         | test         | pracownicy | PRIMARY KEY     |
| pesel           | test         | pracownicy | UNIQUE          |
```

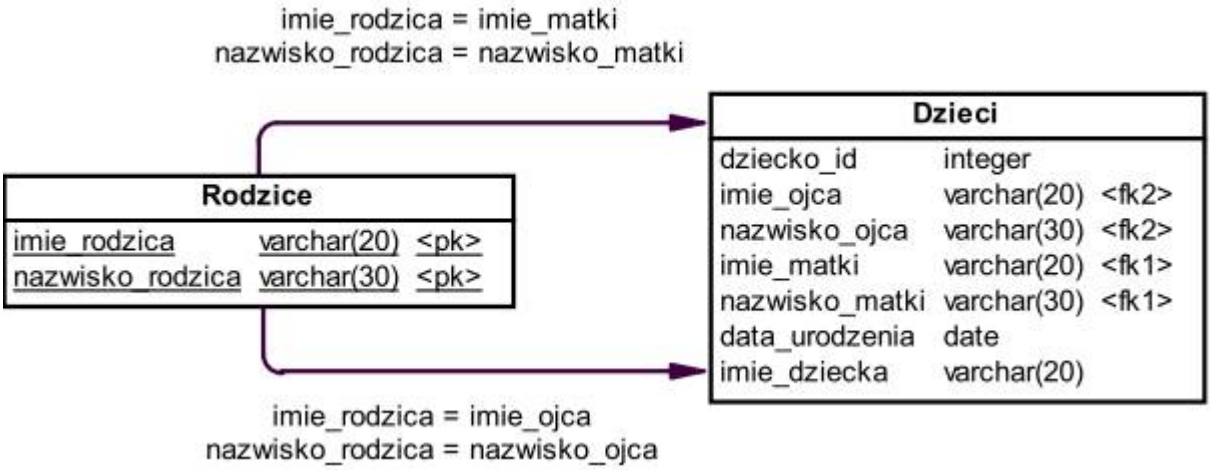
```
| pracownicy_ibfk_1 | test        | pracownicy | FOREIGN KEY     |
+-------------------+-------------+------------+-----------------+
3 rows in set (0.04 sec)
```

Ćwiczenie 20
Usunąć tabelę pracownicy i utworzyć ją ponownie, tym razem dodając klucz obcy (miasto_id).

Aby przekona¢ się, że wszystko utworzyło się po naszej myśli możemy wydać poniższe polecenie:

```
mysql> SELECT constraint_name, table_schema, table_name, constraint_type FROM information_schema.table_constraints WHERE table_schema = 'test' and table_name='pracownicy';
+------------------------+-------------+------------+-----------------+
| constraint_name        | table_schema | table_name | constraint_type |
+------------------------+-------------+------------+-----------------+
| PRIMARY                | test        | pracownicy | PRIMARY KEY     |
| pesel                  | test        | pracownicy | UNIQUE          |
| pracownicy_miasta_id_fk | test        | pracownicy | FOREIGN KEY     |
+------------------------+-------------+------------+-----------------+
3 rows in set (0.00 sec)
```

Ćwiczenie 21
Utworzyć tabele rodzice i dzieci wg. poniższego schematu:



Aby przekona¢ się, że wszystko utworzyło się po naszej myśli możemy wydać poniższe polecenie:

```
mysql> SELECT constraint_name, table_schema, table_name, constraint_type FROM information_schema.table_constraints WHERE table_schema = 'test' and table_name='rodzice';
+-----------------+-------------+------------+-----------------+
| constraint_name | table_schema | table_name | constraint_type |
+-----------------+-------------+------------+-----------------+
| PRIMARY         | test        | rodzice    | PRIMARY KEY     |
+-----------------+-------------+------------+-----------------+
1 row in set (0.00 sec)

mysql> SELECT constraint_name, table_schema, table_name, constraint_type FROM information_schema.table_constraints WHERE table_schema = 'test' and table_name='dzieci';
+-----------------+-------------+------------+-----------------+
| constraint_name | table_schema | table_name | constraint_type |
+-----------------+-------------+------------+-----------------+
| PRIMARY         | test        | dzieci     | PRIMARY KEY     |
| dzieci_ibfk_1   | test        | dzieci     | FOREIGN KEY     |
```

```
| dzieci_ibfk_2   | test        | dzieci     | FOREIGN KEY     |
+-----------------+-------------+------------+-----------------+
3 rows in set (0.00 sec)
```

Ćwiczenie 21

Usunąć tabele pracownicy i stworzyć ją od nowa dodając INDEX do pól nazwisko i data_ur.

```
mysql> DESC pracownicy;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| prac_id  | int(11)     | NO   | PRI | NULL    |       |
| imie     | varchar(20) | YES  |     | NULL    |       |
| nazwisko | varchar(30) | YES  | MUL | NULL    |       |
| pseudo   | varchar(10) | YES  | UNI | NULL    |       |
| data_ur  | date        | YES  | MUL | NULL    |       |
+----------+-------------+------+-----+---------+-------+
5 rows in set (0.04 sec)

mysql> SELECT constraint_name, table_schema, table_name, constraint_type FROM information_schema.table_constraints WHERE table_schema = 'test' and table_name='pracownicy';
+-----------------+-------------+------------+-----------------+
| constraint_name | table_schema | table_name | constraint_type |
+-----------------+-------------+------------+-----------------+
| PRIMARY         | test        | pracownicy | PRIMARY KEY     |
| pseudo          | test        | pracownicy | UNIQUE          |
+-----------------+-------------+------------+-----------------+
2 rows in set (0.00 sec)
```

Literatura

Ćwiczenie opracowane na podstawie materiałów przedstawionych na stronie http://www.uz.zgora.pl/~agramack/
1. Lech Banachowski (tłum.). SQL. Język relacyjnych baz danych. WNT Warszawa, 1995.
2.Paul Dubios. MySQL. Podręcznik administratora. Wydawnictwo HELION, 2005.
3.MySQL 5.0 Reference Manual, 2005. http://dev.mysql.com/doc/
4.Richard Stones and Neil Matthew. Od podstaw. Bazy danych i MySQL. Wydawnictwo HELION, 2003.
5.Luke Welling and Laura Thomson. MySQL. Podstawy. Wydawnictwo HELION, 2005.