Aqui segue exemplos de como resolver alguns dos problemas mais comuns do MySQL.
Algum dos exemplos usam a tabela de compras, coloque os preços de cada artigo (número de item) de cada negociante. Supondo que cada negociante tem um preço fixo por artigo, então (item, negociante) é uma chave primária aos registros.
Você pode criar a tabela de exemplo como:

```
CREATE TABLE shop (
 article INT(4) UNSIGNED ZEROFILL DEFAULT '0000' NOT NULL,
 dealer  CHAR(20)                 DEFAULT ''     NOT NULL,
 price   DOUBLE(16,2)             DEFAULT '0.00' NOT NULL,
 PRIMARY KEY(article, dealer));

INSERT INTO shop VALUES
(1,'A',3.45),(1,'B',3.99),(2,'A',10.99),(3,'B',1.45),(3,'C',1.69),
(3,'D',1.25),(4,'D',19.95);
```

Assim os dados de exemplo estarão:

```
SELECT * FROM shop

+---------+--------+-------+
| article | dealer | price |
+---------+--------+-------+
|    0001 | A      |  3.45 |
|    0001 | B      |  3.99 |
|    0002 | A      | 10.99 |
|    0003 | B      |  1.45 |
|    0003 | C      |  1.69 |
|    0003 | D      |  1.25 |
|    0004 | D      | 19.95 |
+---------+--------+-------+
```

O valor máximo de uma coluna

"Qual é o artigo que tem o preço mais alto?"

```
SELECT MAX(article) AS article FROM shop

+---------+
| article |
+---------+
|       4 |
+---------+
```

A fila da coluna com o número máximo

"Encontre o número do negociantes, e avalie quem tem o artigo mais caro."

No ANSI SQL isto é facilmente feito com um sub-query:

```
SELECT article, dealer, price
FROM   shop
WHERE  price=(SELECT MAX(price) FROM shop)
```

No MySQL (ainda não faz uma sub-seleção) somente faz isto em dois passos:
1. Obtem o valor máximo e avalia a tabela com uma declaração SELECT.
2. Usando este valor compila a pergunta real:

```
SELECT article, dealer, price
FROM   shop
WHERE  price=19.95
```

Outra solução está em classificar todas filas decrescentes por preço e unicamente obter uma fila usando o MySQL cláusula de LIMIT:

```
SELECT article, dealer, price
FROM   shop
ORDER BY price DESC
LIMIT 1
```

Note: Se há vários artigos caros, a solução de LIMIT mostra unicamente um deles.

Máximo da coluna: por grupo e por valores

"Qual é o preço mais alto por artigo?"

```
SELECT article, MAX(price) AS price
FROM   shop
GROUP BY article
```

```
+---------+-------+
| article | price |
+---------+-------+
|    0001 |  3.99 |
|    0002 | 10.99 |
|    0003 |  1.69 |
|    0004 | 19.95 |
+---------+-------+
```

As filas com grupos de campos de valor máximo

"Para cada artigo, encontre o(s) negociante(s) com o preço mais caro."

No MySQL, podemos fazer isto com um sub-query:

```
SELECT article, dealer, price
FROM   shop s1
WHERE  price=(SELECT MAX(s2.price)
            FROM shop s2
            WHERE s1.article = s2.article)
```

Usando chaves estrangeiras

Você não necessita de chaves estrangeiras para unir 2 tabelas.
O MySQL não faz a checagem de certificar que as chaves da tabela são referências e isto não é feito automaticamente apagando as filas da tabela com uma definição de chave estrangeira. Se você usa as chaves normais, ele trabalhará perfeitamente.

```
CREATE TABLE pessoas (
   id INTEGER NOT NULL AUTO_INCREMENT,
   nome CHAR(60) NOT NULL,
   PRIMARY KEY (id)
);

CREATE TABLE camisetas (
   id INTEGER NOT NULL AUTO_INCREMENT,
   tipo ENUM('camiseta', 'polo', 'regata') NOT NULL,
   cor ENUM('vermelho', 'azul', 'laranja', 'branco', 'preto') NOT NULL,
```

```
    dono INTEGER NOT NULL REFERENCES pessoas,
    PRIMARY KEY (id)
);

INSERT INTO pessoas VALUES (NULL, 'Antonio Paz');

INSERT INTO camisetas VALUES
(NULL, 'polo', 'azul', LAST_INSERT_ID()),
(NULL, 'regata', 'branco', LAST_INSERT_ID()),
(NULL, 'camiseta', 'azul', LAST_INSERT_ID());

INSERT INTO pessoas VALUES (NULL, 'Lilliana Angelovska');

INSERT INTO camisetas VALUES
(NULL, 'regata', 'laranja', LAST_INSERT_ID()),
(NULL, 'polo', 'vermelho', LAST_INSERT_ID()),
(NULL, 'regata', 'azul', LAST_INSERT_ID()),
(NULL, 'camiseta', 'branco', LAST_INSERT_ID());

SELECT * FROM pessoas;
+----+--------------------+
| id | nome               |
+----+--------------------+
| 1  | Antonio Paz        |
| 2  | Lilliana Angelovska |
+----+--------------------+

SELECT * FROM camisetas;
+----+---------+--------+-------+
| id | tipo    | cor    | dono  |
+----+---------+--------+-------+
| 1  | polo    | azul   |     1 |
| 2  | regata  | branco |     1 |
| 3  | camiseta | azul  |     1 |
| 4  | regata  | laranja |     2 |
| 5  | polo    | vermelho |     2 |
| 6  | regata  | azul   |     2 |
| 7  | camiseta | branco |     2 |
+----+---------+--------+-------+

SELECT s.* FROM pessoas p, camisetas s
 WHERE p.nome LIKE 'Lilliana%'
   AND s.dono = p.id
   AND s.cor <> 'branco';

+----+-------+--------+-------+
| id | tipo  | cor    | dono  |
+----+-------+--------+-------+
|  4 | regata | laranja |     2 |
|  5 | polo   | vermelho |     2 |
|  6 | regata | azul   |     2 |
+----+-------+--------+-------+
```

Mas se você quer implementar as regras de limitação (constraints), então deve explicitar as chaves estrangeiras quando realizar a ligação das tabelas.
Obs: SÒ irá funcionar se a sua tabela estiver utilizando o motor (engine) INnoDB e não os demais motores, como o MyISAM.

```
CREATE TABLE pessoas (
    id INTEGER NOT NULL AUTO_INCREMENT,
```

```
    nome CHAR(60) NOT NULL,
    PRIMARY KEY (id)
) ENGINE=INNODB;


CREATE TABLE camisetas (
    id INTEGER NOT NULL AUTO_INCREMENT,
    tipo ENUM('camiseta', 'polo', 'regata') NOT NULL,
    cor ENUM('vermelho', 'azul', 'laranja', 'branco', 'preto') NOT NULL,
    dono INTEGER NOT NULL,
    FOREIGN KEY (dono) REFERENCES pessoas(id) ON DELETE CASCADE,
    PRIMARY KEY (id)
) ENGINE=INNODB;


INSERT INTO pessoas VALUES (NULL, 'Antonio Paz');

INSERT INTO camisetas VALUES
(NULL, 'polo', 'azul', LAST_INSERT_ID()),
(NULL, 'regata', 'branco', LAST_INSERT_ID()),
(NULL, 'camiseta', 'azul', LAST_INSERT_ID());

INSERT INTO pessoas VALUES (NULL, 'Lilliana Angelovska');

INSERT INTO camisetas VALUES
(NULL, 'regata', 'laranja', LAST_INSERT_ID()),
(NULL, 'polo', 'vermelho', LAST_INSERT_ID()),
(NULL, 'regata', 'azul', LAST_INSERT_ID()),
(NULL, 'camiseta', 'branco', LAST_INSERT_ID());

SELECT * FROM pessoas;
+----+--------------------+
| id | nome               |
+----+--------------------+
| 1  | Antonio Paz        |
| 2  | Lilliana Angelovska |
+----+--------------------+

SELECT * FROM camisetas;
+----+---------+--------+-------+
| id | tipo    | cor    | dono  |
+----+---------+--------+-------+
| 1  | polo    | azul   |     1 |
| 2  | regata  | branco |     1 |
| 3  | camiseta | azul  |     1 |
| 4  | regata  | laranja |     2 |
| 5  | polo    | vermelho |     2 |
| 6  | regata  | azul   |     2 |
| 7  | camiseta | branco |     2 |
+----+---------+--------+-------+

SELECT s.* FROM pessoas p, camisetas s
 WHERE p.nome LIKE 'Lilliana%'
   AND s.dono = p.id
   AND s.cor <> 'branco';

+----+-------+--------+-------+
| id | tipo | cor    | dono  |
+----+-------+--------+-------+
|  4 | regata | laranja |     2 |
|  5 | polo  | vermelho |     2 |
|  6 | regata | azul   |     2 |
+----+-------+--------+-------+
```

Exercícios:

1 - Crie uma tabela chamada Veículo, contendo os campos:

Placa – varchar(7) chave primária, não nulo.

Marca – ENUM (Fiat, Chevrolet, VolksWagen, Ford), não nulo.

Modelo – Varchar(50), não nulo

Ano – Integer(4) não nulo

Cor – ENUM (preto, prata, branco, vermelho, azul, outras)

2 – Crie uma tabela chamada Estacionamento, contendo os campos:

Cod – inteiro, auto numeração, chave primária, não nulo

Placa – Varchar(7) chave estrangeira da tabela veículo, não nulo

Nro_vaga – integer(3), não nulo

Hora_entrada – datatime, não nulo

Hora_saída – datatime, não nulo

3 – Insira 3 carros na tabela veículos

4 – Insira cada carro em uma vaga diferente na tabela Estacionamento, considerando todos os carros entrando na mesma hora em vagas diferentes.

5 – Tente inserir um carro em uma vaga na tabela Estacionamento sem ter inserido o veículo na tabela veículos

6 – Altere o segundo veículo inserido em uma vaga, preenchendo o horário de saída com 45 minutos de diferença para a entrada.

7 – Insira um veículo novo na tabela veículo e insira-o na mesma vaga do veículo que saiu.

8 – Faça a exclusão do veículo 1. Verifique se ele foi excluído da tabela Estacionamento

9 – Preencha a saída de todos os veículos que estejam no estacionamento.

10 – Monte uma consulta que mostre quanto tempo cada veículo permaneceu nas vagas.