

**Repositório:** <https://www.github.com/Navomeister/bcd-umbrela/tree/main>

## Queries:

1. Consultar o endereço, hora de início (start time) e hora final (end time) dos pontos de serviço da mesma cidade que o usuário cujo ID é 5.

Mostrando registros 0 - 3 (4 no total. Consulta levou 0.0022 segundos.)

```
SELECT service_point.startTime, service_point.endTime FROM service_point WHERE service_point.city IN (SELECT address.city FROM address INNER JOIN user ON address.user_id = user.userId WHERE user.userId = 5);
```

Perfil [ Editar em linha ] [ Editar ] [ Demonstrar SQL ] [ Criar código PHP ] [ Atualizar ]

☐ Mostrar tudo | Número de linhas: 25 | Filtrar linhas: Procurar nesta tabela | Ordenar pela chave: Nenhum

Opções extras

	startTime	endTime
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Remover	10:00 am	5:30 pm
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Remover	10:00 am	5:30 pm
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Remover	10:00 am	6:00 pm
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Remover	9:30 am	5:30 pm

2. Consultar todos os produtos que é do tipo laptop.

```
/* Consultar todos os produtos que é do tipo laptop.*/ SELECT * FROM product WHERE product.type LIKE "%laptop%";
```

[ Editar em linha ] [ Editar ] [ Criar código PHP ]

☐ Mostrar tudo | Número de linhas: 25 | Filtrar linhas: Procurar nesta tabela | Ordenar pela chave: Nenhum

Opções extras

	productid	store_id	name	brand	type	amount	price	colour	customerReview	modelName
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Remover	1	8	ASUS Chromebook 11.6 laptop	Asus	laptop	262	NULL	navy blue	10	C201PA-DS02
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Remover	8	8	Microsoft Surface Pro 4 i5 (128GB) with Wireless M...	Microsoft	laptop	1350	NULL	black	30	CR5-00001

3. Consultar a quantidade total de produtos que foram colocados no carrinho (shopping cart), considerando a loja com storeid (sid) igual a 8.

Mostrando registros 0 - 0 (1 no total. Consulta levou 0.0004 segundos.)

```
/* Consultar a quantidade total de produtos que foram colocados no carrinho (shopping cart), considerando a loja com storeid (sid) igual a 8.*/ SELECT SUM(save_to_shopping_cart.quantity) AS qtProducts FROM save_to_shopping_cart INNER JOIN product ON save_to_shopping_cart.productId = product.productId WHERE product.store_id = 8;
```

[ Editar em linha ] [ Editar ] [ Criar código PHP ]

☐ Mostrar tudo | Número de linhas: 25 | Filtrar linhas: Procurar nesta tabela

Opções extras

qtProducts  
6

4. Consultar os campos name, streetaddr e city (tabela address) de todos os pedidos que foram entregues em 17-02-2017.

Mostrando registros 0 - 0 (1 no total. Consulta levou 0.0030 segundos.)

```
/* Consultar os campos name, streetaddr e city (tabela address) de todos os pedidos que foram entregues em 17-02-2017.*/ SELECT user.name, address.streetAddr, address.city FROM user INNER JOIN address ON user.userId = address.userId INNER JOIN deliver_to ON address.addrId = deliver_to.addrId WHERE deliver_to.timeDelivered = '2017-02-17';
```

[ Editar em linha ] [ Editar ] [ Criar código PHP ]

☐ Mostrar tudo | Número de linhas: 25 | Filtrar linhas: Procurar nesta tabela

Opções extras

name	streetAddr	city
Tynone D. Harvey	Ap #881-9739 In Rd.	Saint-Georges

## 5. Consulte os comentários do produto 123456789;

✓ Mostrando registros 0 - 1 (2 no total, Consulta levou 0.0003 segundos.)

`SELECT comments.content FROM product INNER JOIN comments ON product.productId = comments.productId WHERE product.productId = 5;`

☐ Perfil [\[ Editar em linha \]](#) [\[ Editar \]](#) [\[ Demonstrar SQL \]](#) [\[ Criar código PHP \]](#) [\[ Atualizar \]](#)

☐ Mostrar tudo | Número de linhas: 25 ▾ | Filtrar linhas:  | Ordenar pela chave: Nenhum ▾

Opções extras

**content**  
It is the first Go Pro I have had and so far I am ...  
First go pro I have ever purchased. Really impress...

## Tipos dos Campos:

'addrId', 'cardId', 'buyerId', 'orderId', 'itemId', 'creditId', 'debitId', 'addrId', 'sellerId', 'storeId', 'productId', 'buyerId', 'idComm', 'serviceld' e suas respectivas chaves estrangeiras: int(10); são chaves primárias, portanto precisam ser exatas e permanentes. As chaves estrangeiras precisam ser idênticas às primárias, portanto são do mesmo tipo.

address => name, city, postalCode, province: varchar(50); são textos pequenos e variáveis, então não precisam de muito espaço.

streetAddr: varchar(250); é um texto que pode ser grande, precisa de mais espaço disponível.

contactPhoneNumber: char(11); é um número de tamanho fixo, mas não são feitas contas com ele, podendo ser deixado como char.

bank\_card => cardNumber: char(19); também um campo com tamanho fixo que não será feito contas.

bank: varchar(50); pequeno texto de tamanho variável.

expiryDate: date; data que não precisa de horário.

contain => quantity: int(5); pequena quantidade de tamanho variável.

credit\_card => organization: varchar(50); texto pequeno.

deliver\_to => timeDelivered: datetime; data com horário, já que precisa de especificidade.

manage => setupTime: datetime; data com horário, também precisa ser específico

order => creationTime: datetime; apenas o horário deixaria muito aberto

paymentStatus: varchar(50); texto com poucas palavras, não precisa de muito.

totalAmount: int(5); um número com poucos dígitos, dificilmente haverá uma compra grande os suficiente para usá-los.

order\_item => price: float(6,2); um valor decimal com 2 dígitos após a vírgula, para preço em dinheiro.

creationTime: datetime; novamente data com hora, pelos mesmos motivos.

payment => paytime: datetime; mesmo motivo.

product => name: varchar(100); texto para o nome do produto, não sendo nem tão grande nem pequeno demais.

brand: varchar(50); marcas não costumam ser grandes.

type: varchar(50); o tipo do produto normalmente é usado em tags, então não são grandes.

amount: int(5); o estoque não deve ser maior que 99999 unidades.

price: float(6,2); preço decimal para uso em moeda.

colour: varchar(50); apenas o nome da cor.

customerReview: int(3); avaliação de 0 a 100.

save\_to\_shopping\_cart => addTime: datetime; data com especificidade da hora.

quantity: int(3); não precisa ser tão grande.

store => name: varchar(50); nomes de lojas não costumam ser tão grandes.

startTime: datetime; necessário pro input, mesmo não tendo horário.

customerGrade: int(1); número de estrelas, só vai até 5.

streetAddr: varchar(100); rua e número, valor alto por precaução.

city: varchar(50); nome da cidade normalmente é pequeno.

province: varchar(50); nome de estado também é pequeno.

user => name: varchar(100); valor grande para comportar nomes exóticos.

phoneNum: char(13); char com valor fixo, já que número de telefone tem um padrão.

service\_point => streetAddr: varchar(250); endereço completo.

province: varchar(50); apenas nome do estado.

city: varchar(50); apenas o nome da cidade.

startTime: varchar(50); texto para aceitar o modo do input (ex. '5:00 am').

endTime: varchar(50); mesmo motivo acima.

comments => creationTime: datetime; para aceitar tanto data quanto hora.

grade: float(2,1); para aceitar notas de 0,1 até 5,0.

content: varchar(500); tamanho grande para aceitar comentários detalhados.