```
SET FOREIGN_KEY_CHECKS=OFF;
DROP TABLE IF EXISTS beverage;
DROP TABLE IF EXISTS store;
DROP TABLE IF EXISTS sells;
SET FOREIGN_KEY_CHECKS=ON;

create table `beverage` (
        `code` int,
   `name` varchar(15),
   `size` varchar(10),
   primary key(`code`)
);

INSERT INTO beverage values(121, 'Latte', 'medium');
INSERT INTO beverage values(140, 'Latte', 'large');

INSERT INTO beverage values(122, 'Tea', 'large');
INSERT INTO beverage values(131, 'Tea', 'small');

INSERT INTO beverage values(123, 'Cola', 'medium');
INSERT INTO beverage values(133, 'Cola', 'large');

INSERT INTO beverage values(124, 'Mocha', 'medium');
INSERT INTO beverage values(134, 'Mocha', 'large');

INSERT INTO beverage values(125, 'Pepsi', 'large');
INSERT INTO beverage values(135, 'Pepsi', 'medium');

INSERT INTO beverage values(126, 'Schweppes', 'small');
```

```
INSERT INTO beverage values(136, 'Schweppes', 'medium');

INSERT INTO beverage values(127, 'Dr Pepper', 'medium');

INSERT INTO beverage values(128, 'Fanta', 'large');

INSERT INTO beverage values(129, 'Sprite', 'medium');

INSERT INTO beverage values(130, 'Cappy', 'small');

INSERT INTO beverage values(132, 'Mountain Dew', 'large');

INSERT INTO beverage values(137, 'Nestea', 'medium');

INSERT INTO beverage values(138, 'Lemonade', 'small');

INSERT INTO beverage values(139, 'Budweiser', 'medium');

/* ---------------------------------------------------- */

create table `store` (
        `name` varchar(15) not null,
   `location` varchar (20),
   `telephone` numeric,
   primary key (`name`)
);

INSERT INTO store values('REMA', 'Horsens', '45252163');
INSERT INTO store values('Fotex', 'Aarhus C', '45252153');
```

```
INSERT INTO store values('Bilka', 'Randers', '45784163');
INSERT INTO store values('Netto', 'Aarhus V', '45284263');
INSERT INTO store values('Fakta', 'Horsens', '45215163');

/* ------------------------------------------------- */

create table sells (
        `store_name` varchar(15),
   `code` int,
   `price` numeric,
   primary key (`store_name`, `code`),
   foreign key (`store_name`) references store(`name`),
   foreign key (`code`) references beverage(`code`)
);

INSERT INTO sells values('REMA', 122, '10');
INSERT INTO sells values('Fotex', 121, '15');
INSERT INTO sells values('REMA', 132, '20');
INSERT INTO sells values('Fotex', 122, '25');
INSERT INTO sells values('REMA', 123, '30');
INSERT INTO sells values('Fotex', 123, '40');
INSERT INTO sells values('Bilka', 121, '38');
-- INSERT INTO sells values('Netto', 121, '20');
INSERT INTO sells values('REMA', 124, '100');
INSERT INTO sells values('Fotex', 124, '80');
INSERT INTO sells values('REMA', 125, '5');
-- INSERT INTO sells values('Netto', 122, '5'); -- cheapest drink
-- INSERT INTO sells values('Netto', 123, '60');
INSERT INTO sells values('Fotex', 125, '200');
```

```
INSERT INTO sells values('REMA', 126, '85');
-- INSERT INTO sells values('Netto', 124, '60');
-- INSERT INTO sells values('Netto', 125, '65');
INSERT INTO sells values('Fotex', 126, '87');
INSERT INTO sells values('REMA', 127, '15');
INSERT INTO sells values('Bilka', 122, '20');
INSERT INTO sells values('REMA', 128, '15');
INSERT INTO sells values('Fakta', 121, '20');
INSERT INTO sells values('Fakta', 122, '15');
INSERT INTO sells values('Fakta', 123, '4');
INSERT INTO sells values('Fakta', 124, '15');
INSERT INTO sells values('Fakta', 140, '49'); -- FAKTA latte large, 49 dkk, horsens
INSERT INTO sells values('Fotex', 127, '15');
INSERT INTO sells values('Fotex', 128, '20');
INSERT INTO sells values('Fotex', 129, '15');
INSERT INTO sells values('Fotex', 130, '20');
INSERT INTO sells values('REMA', 129, '15');
INSERT INTO sells values('Fotex', 131, '20');
INSERT INTO sells values('Bilka', 123, '15');
INSERT INTO sells values('Bilka', 124, '20');
INSERT INTO sells values('Bilka', 125, '15');
INSERT INTO sells values('Bilka', 126, '20');
INSERT INTO sells values('Bilka', 140, '35'); -- bilka, large latte, 35 dkk, randers
INSERT INTO sells values('Fotex', 132, '10');
INSERT INTO sells values('REMA', 130, '15');
INSERT INTO sells values('Fotex', 133, '20');
-- INSERT INTO sells values('Netto', 126, '15');
INSERT INTO sells values('Fotex', 134, '20');
INSERT INTO sells values('REMA', 131, '15');
```

```
-- INSERT INTO sells values('Netto', 127, '20');
INSERT INTO sells values('REMA', 140, '40'); -- rema large latte, 40 dkk, horsens
INSERT INTO sells values('Fotex', 135, '20');
-- INSERT INTO sells values('Netto', 128, '15');
-- INSERT INTO sells values('Netto', 129, '20');
-- INSERT INTO sells values('Netto', 130, '15');
INSERT INTO sells values('Fotex', 136, '20');
INSERT INTO sells values('Fotex', 137, '15');

INSERT INTO sells values('Netto', 121, '20');
INSERT INTO sells values('Netto', 122, '20');
INSERT INTO sells values('Netto', 123, '20');
INSERT INTO sells values('Netto', 124, '20');
INSERT INTO sells values('Netto', 125, '20');
INSERT INTO sells values('Netto', 126, '20');
INSERT INTO sells values('Netto', 127, '20');
INSERT INTO sells values('Netto', 128, '20');
INSERT INTO sells values('Netto', 129, '20');
INSERT INTO sells values('Netto', 130, '20');
INSERT INTO sells values('Netto', 131, '20');
INSERT INTO sells values('Netto', 132, '20');
INSERT INTO sells values('Netto', 133, '20');
INSERT INTO sells values('Netto', 134, '20');
INSERT INTO sells values('Netto', 135, '20');
INSERT INTO sells values('Netto', 136, '20');
INSERT INTO sells values('Netto', 137, '20');
INSERT INTO sells values('Netto', 138, '20');
INSERT INTO sells values('Netto', 139, '20');
INSERT INTO sells values('Netto', 140, '20');
```

```
-- 1. Find the names of beverages that are offered in 'maxi' size

select `name` from beverage where size = 'large';


-- 2. Find the names of beverages that come in 'maxi' or 'medium' size

select `name`, size from beverage where size = 'large' or size = 'medium';


-- 3. Find the names of beverages that come in both a 'maxi' and 'medium' size

SELECT a.name, a.size, b.size

FROM beverage a, beverage b

WHERE a.name = b.name

AND a.size = 'large' and b.size = 'medium'

ORDER BY A.name;


-- 4. Find the names and the phone numbers of the stores in "Randers" or "Horsens" that sell a 'maxi'
beverage named "latte"

-- for no more than 45 kr


select a.`name`, a.telephone

from store a, sells b, beverage c

where a.location in ('Randers', 'Horsens')

        and b.`code` = c.`code`

        and b.store_name = a.`name`

        and b.price <= 45 and c.`name` = 'latte' and c.size = 'large';


-- 5. Find the code(s), name(s), and name(s) of store(s), selling the least expensive beverage(s).

select c.`code`, c.`name`, a.`name`

from store a, sells b, beverage c

WHERE
```

```
        b.`code` = c.`code`
        and b.store_name = a.`name`
  and b.price = (
    SELECT
      MIN(sells.price)
    FROM
      sells);

-- Version 2, no aggregation
select b.`code`, b.`name`, t.`name` as "store name"
from (store t, sells s, beverage b)
LEFT JOIN sells s2
  ON s2.price < s.price
        where b.`code` = s.`code` and s.store_name = t.name and s2.price is null;

-- 6.For each store , give its name and the code(s) of the least expensive beverage(s) it sells.

select t.`name`, s.price, b.`code`
from store t, sells s, beverage b
where b.`code` = s.`code` and s.`store_name` = t.`name`
        and not exists (
                select * from sells s1, beverage b1
    where s1.`code` = b1.`code` and s1.store_name=t.`name` and
                        s1.price < s.price
  );

-- 7. For each store, give its name and the price of the least expensive beverage(s) it sells.
-- Repeat, but now (i) include the name(s) of such beverage(s) and (ii) do not use any aggregation operations,
```

```
-- such as like MIN, GROUP BY, ORDER BY, etc.


-- 7.1

select a.store_name, a.price

from sells a, store b

group by a.`store_name`, a.price

having (a.price, a.store_name) in (select min(t.price), t.store_name from sells t, store g where
t.`store_name` = g.`name` group by t.`store_name`)

order by a.price asc;


-- 7.2

select t.`name`, s.price, b.`name`

from store t, sells s, beverage b

where b.`code` = s.`code` and s.`store_name` = t.`name`

        and not exists (

                select * from sells s1, beverage b1

    where s1.`code` = b1.`code` and s1.store_name=t.`name` and

                        s1.price < s.price

  );


-- 8. For each beverage, give its name, size, and its highest price across all stores.

-- Repeat, but now (i) include the name(s) of store(s) selling that beverage at the highest price, and (ii)

-- do not use any aggregation operations,such as MAX, GROUP BY, ORDER BY, etc.


-- 8.1

select b.`name`, b.size, max(s.price)

from beverage b, sells s

where b.`code` = s.`code`

group by b.`name`, b.`code`;
```

```
-- 8.2
select b.`name`, b.size, s.price, t.`name`
from store t, sells s, beverage b
where b.`code` = s.`code` and s.`store_name` = t.`name`
        and not exists (
                select * from sells s1, store t1
    where s1.`code` = b.`code` and s.store_name=t1.`name` and
                        s1.price > s.price
  );

-- 9. Find the names of the stores that offer all beverage codes; do not use COUNT.
select s.`name`
from store s
where not exists (
        select * from beverage b
  where not exists (
                select * from sells e
                where e.`code` = b.`code` and e.`store_name` =  s.`name`));

-- 10. Find the largest price difference between beverages of the same name across all stores, the name(s) of the beverage(s)
-- having that price difference, and the relevant store(s).

select abs(s.price - s1.price) as priceDifference, b.`name`, s.store_name, s1.store_name
from beverage b, sells s, beverage b1, sells s1, store t
where b.`code` = s.`code` and b1.`code` = s1.`code` and t.`name` =  s.store_name and b.`name` = b1.`name`
group by priceDifference desc limit 1
```