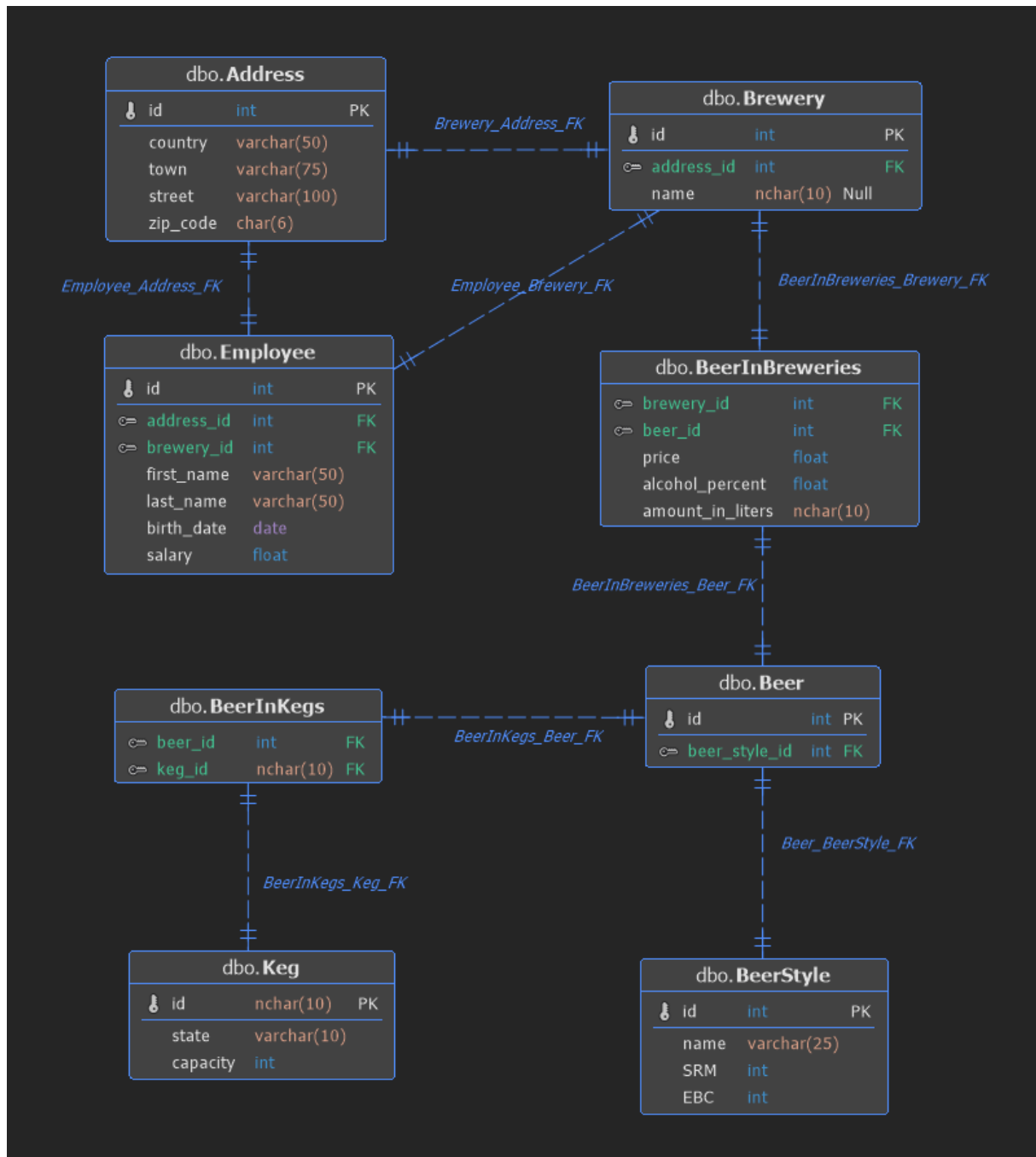


# Report 1

Databases optimization

Attention! Whole project is available at the [repository](#).

## 1. Schema of the database



## 2. Description how structure is generated

```
/* STRUCTURE */
CREATE TABLE dbo.Address
(
    id INT NOT NULL IDENTITY(1, 1) PRIMARY KEY,
    country VARCHAR(50),
    town VARCHAR(75),
    street VARCHAR(100),
    zip_code CHAR(6),
    CONSTRAINT CHK_zip_code CHECK (zip_code LIKE '[0-9][0-9]-[0-9][0-9][0-9]')
);

CREATE TABLE dbo.Brewery
(
    id INT NOT NULL IDENTITY(1, 1) PRIMARY KEY,
    address_id INT NOT NULL,
    name VARCHAR(100),
    CONSTRAINT FK_Brewery_Address FOREIGN KEY (address_id) REFERENCES Address (id)
);

CREATE TABLE dbo.Employee
(
    id INT NOT NULL IDENTITY(1, 1) PRIMARY KEY,
    address_id INT NOT NULL,
    brewery_id INT NOT NULL,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    birth_date DATE,
    salary DECIMAL(8, 2),
    CONSTRAINT FK_Employee_Brewery FOREIGN KEY (brewery_id) REFERENCES Brewery (id),
    CONSTRAINT FK_Employee_Address FOREIGN KEY (address_id) REFERENCES Address (id)
);

CREATE TABLE dbo.BeerStyle
(
    id INT NOT NULL IDENTITY(1, 1) PRIMARY KEY,
    name VARCHAR(25),
    SRM INT,
    EBC INT
);

CREATE TABLE dbo.Beer
(
    id INT NOT NULL IDENTITY(1, 1) PRIMARY KEY,
    beer_style_id INT NOT NULL,
    price DECIMAL(6, 2),
    alcohol_percent DECIMAL(4, 2),
    amount_in_liters DECIMAL(6, 2),
    CONSTRAINT FK_Beer_Beer_Style FOREIGN KEY (beer_style_id) REFERENCES BeerStyle
(id)
);

CREATE TABLE dbo.Keg
(
    id INT NOT NULL IDENTITY(1, 1) PRIMARY KEY,
    state VARCHAR(10),
    capacity INT
);

CREATE TABLE dbo.BeerInBreweries
(
```

```

        brewery_id INT NOT NULL,
        beer_id INT NOT NULL,
        CONSTRAINT FK_Beer_In_Breweries_Brewery FOREIGN KEY (brewery_id) REFERENCES
Brewery (id) ON DELETE CASCADE,
        CONSTRAINT FK_Beer_In_Breweries_Beer FOREIGN KEY (beer_id) REFERENCES Beer (id)
    );

CREATE TABLE dbo.BeerInKegs
(
    beer_id INT NOT NULL,
    keg_id INT NOT NULL,
    CONSTRAINT FK_Beer_In_Kegs_Beer FOREIGN KEY (beer_id) REFERENCES Beer (id) ON
DELETE CASCADE,
    CONSTRAINT FK_Beer_In_Kegs_Keg FOREIGN KEY (keg_id) REFERENCES Keg (id)
);

```

### 3. Description how the data is generated

The fastest way to fill a database with a large amount of data is to do it by SQL script that will execute at this specific database, so I made a few. The structure of my SQL application, if I could say that, consists of few generators scripts that control the whole process of data-generating and execute construction scripts. Each table has its own construction script that fills it perfectly.

Example of Address table generator. All of the scripts are available on my [repository](#).

```

/* CHECK TABLES */
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'TempCountry')
DROP TABLE TempCountry;
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'TempStreet')
DROP TABLE TempStreet;
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'TempTown')
DROP TABLE TempTown;
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'Address')
DROP TABLE Address;

/* TABLE */
CREATE TABLE dbo.Address
(
    id INT NOT NULL IDENTITY(1, 1) PRIMARY KEY,
    country VARCHAR(50),
    town VARCHAR(75),
    street VARCHAR(100),
    zip_code CHAR(6),
    CONSTRAINT CHK_zip_code CHECK (zip_code LIKE '[0-9][0-9]-[0-9][0-9][0-9]')
);

/* TEMP TABLES */
CREATE TABLE TempCountry
(
    tempID INT NOT NULL IDENTITY(1, 1) PRIMARY KEY,
    tempValue VARCHAR(100) NOT NULL
);

CREATE TABLE TempTown
(

```

```

tempID INT NOT NULL IDENTITY(1, 1) PRIMARY KEY,
tempValue VARCHAR(100) NOT NULL
);

CREATE TABLE TempStreet
(
tempID INT NOT NULL IDENTITY(1, 1) PRIMARY KEY,
tempValue VARCHAR(100) NOT NULL
);

/* TEMPORARY DATA INSERTION FOR RANDOM REAL DATA GENERATION */
INSERT INTO TempCountry(tempValue) VALUES ('Nianemark');
INSERT INTO TempCountry(tempValue) VALUES ('Tarucokmo');
INSERT INTO TempCountry(tempValue) VALUES ('Honengmal');

INSERT INTO TempTown(tempValue) VALUES ('New Franew');
INSERT INTO TempTown(tempValue) VALUES ('Kurfham');
INSERT INTO TempTown(tempValue) VALUES ('Terresnod');
INSERT INTO TempTown(tempValue) VALUES ('Saint Richall');
INSERT INTO TempTown(tempValue) VALUES ('West Twhist');
INSERT INTO TempTown(tempValue) VALUES ('Red Ranrell');
INSERT INTO TempTown(tempValue) VALUES ('Landington');
INSERT INTO TempTown(tempValue) VALUES ('Linsristown');
INSERT INTO TempTown(tempValue) VALUES ('Red Crosshithewelsh');
INSERT INTO TempTown(tempValue) VALUES ('Stonesumxan');
INSERT INTO TempTown(tempValue) VALUES ('Sauing');
INSERT INTO TempTown(tempValue) VALUES ('Red Thampnel');
INSERT INTO TempTown(tempValue) VALUES ('Alrslem Creek');
INSERT INTO TempTown(tempValue) VALUES ('Grand Gibbsall');
INSERT INTO TempTown(tempValue) VALUES ('Port Sodlon');
INSERT INTO TempTown(tempValue) VALUES ('Co');
INSERT INTO TempTown(tempValue) VALUES ('Stokeweland');
INSERT INTO TempTown(tempValue) VALUES ('North Mauwich-In-Surwal');
INSERT INTO TempTown(tempValue) VALUES ('East Tombbid Du Chucolne');
INSERT INTO TempTown(tempValue) VALUES ('Hocourtney River');
INSERT INTO TempTown(tempValue) VALUES ('La Tegoolhav');
INSERT INTO TempTown(tempValue) VALUES ('Mount Landcotes Du Thamsprings');
INSERT INTO TempTown(tempValue) VALUES ('North Bilpeltod');
INSERT INTO TempTown(tempValue) VALUES ('Saint Wkerneread Upon Dealpat');
INSERT INTO TempTown(tempValue) VALUES ('Dgelia With Knareswai');
INSERT INTO TempTown(tempValue) VALUES ('North Rdermal');
INSERT INTO TempTown(tempValue) VALUES ('Dclay');
INSERT INTO TempTown(tempValue) VALUES ('La Basbo');
INSERT INTO TempTown(tempValue) VALUES ('Feagisland');
INSERT INTO TempTown(tempValue) VALUES ('Langdy Heights');
INSERT INTO TempTown(tempValue) VALUES ('Port Cokemed-In-Wilday');
INSERT INTO TempTown(tempValue) VALUES ('Port Vycold');
INSERT INTO TempTown(tempValue) VALUES ('Yustown-In-Stroudzouch');

INSERT INTO TempStreet(tempValue) VALUES ('Stockwell Walk');
INSERT INTO TempStreet(tempValue) VALUES ('Raglan Garden');
INSERT INTO TempStreet(tempValue) VALUES ('Hill Causeway');
INSERT INTO TempStreet(tempValue) VALUES ('Nowell Avenue');
INSERT INTO TempStreet(tempValue) VALUES ('St Davids Corner');
INSERT INTO TempStreet(tempValue) VALUES ('Sydney Meadow');
INSERT INTO TempStreet(tempValue) VALUES ('Old Farm Street');
INSERT INTO TempStreet(tempValue) VALUES ('Bolingbroke Road');
INSERT INTO TempStreet(tempValue) VALUES ('Leys Promenade');
INSERT INTO TempStreet(tempValue) VALUES ('Eckersley Drive');
INSERT INTO TempStreet(tempValue) VALUES ('Mcdermott Road');
INSERT INTO TempStreet(tempValue) VALUES ('Sandfield Strand');
INSERT INTO TempStreet(tempValue) VALUES ('Argus Road');

```

```

INSERT INTO TempStreet(tempValue) VALUES ('Heywood Brae');
INSERT INTO TempStreet(tempValue) VALUES ('Mountain Croft');
INSERT INTO TempStreet(tempValue) VALUES ('Goldfinch Cloisters');
INSERT INTO TempStreet(tempValue) VALUES ('Elder Brae');
INSERT INTO TempStreet(tempValue) VALUES ('Bradford Bank');
INSERT INTO TempStreet(tempValue) VALUES ('St James Promenade');
INSERT INTO TempStreet(tempValue) VALUES ('Heol Drindod');
INSERT INTO TempStreet(tempValue) VALUES ('Victory Brook');
INSERT INTO TempStreet(tempValue) VALUES ('Holland Approach');
INSERT INTO TempStreet(tempValue) VALUES ('Walker Newydd');
INSERT INTO TempStreet(tempValue) VALUES ('Kings Parade');
INSERT INTO TempStreet(tempValue) VALUES ('Harcourt Hollow');
INSERT INTO TempStreet(tempValue) VALUES ('Copse Circus');
INSERT INTO TempStreet(tempValue) VALUES ('James Grange');
INSERT INTO TempStreet(tempValue) VALUES ('Kennet Esplanade');
INSERT INTO TempStreet(tempValue) VALUES ('Bishops Royd');
INSERT INTO TempStreet(tempValue) VALUES ('Carr Bridge');
INSERT INTO TempStreet(tempValue) VALUES ('Glen Villas');
INSERT INTO TempStreet(tempValue) VALUES ('Chelsea Piece');
INSERT INTO TempStreet(tempValue) VALUES ('Morecambe Gardens');
INSERT INTO TempStreet(tempValue) VALUES ('Brambling Top');
INSERT INTO TempStreet(tempValue) VALUES ('Clarke Gate');
INSERT INTO TempStreet(tempValue) VALUES ('Queens Promenade');
INSERT INTO TempStreet(tempValue) VALUES ('Patience Grove');
INSERT INTO TempStreet(tempValue) VALUES ('Blue Lion Close');
INSERT INTO TempStreet(tempValue) VALUES ('Clyde Hollow');
INSERT INTO TempStreet(tempValue) VALUES ('Bradley Maltings');
INSERT INTO TempStreet(tempValue) VALUES ('Balsall Heath Road');
INSERT INTO TempStreet(tempValue) VALUES ('Bryn Arthur');
INSERT INTO TempStreet(tempValue) VALUES ('Tree Road');
INSERT INTO TempStreet(tempValue) VALUES ('Town Drive');
INSERT INTO TempStreet(tempValue) VALUES ('Keats Promenade');
INSERT INTO TempStreet(tempValue) VALUES ('Moss Top');
INSERT INTO TempStreet(tempValue) VALUES ('Cavendish Links');
INSERT INTO TempStreet(tempValue) VALUES ('Cheltenham Elms');
INSERT INTO TempStreet(tempValue) VALUES ('Wheatfield Grange');
INSERT INTO TempStreet(tempValue) VALUES ('Richard Vale');
INSERT INTO TempStreet(tempValue) VALUES ('Town Furlong');
INSERT INTO TempStreet(tempValue) VALUES ('Norristhorpe Lane');
INSERT INTO TempStreet(tempValue) VALUES ('Franklin North');
INSERT INTO TempStreet(tempValue) VALUES ('Gardner Isaf');
INSERT INTO TempStreet(tempValue) VALUES ('Summer Gate');
INSERT INTO TempStreet(tempValue) VALUES ('Macrae Street');
INSERT INTO TempStreet(tempValue) VALUES ('Woodview Orchards');
INSERT INTO TempStreet(tempValue) VALUES ('Victory Cliff');
INSERT INTO TempStreet(tempValue) VALUES ('Tulip Brambles');
INSERT INTO TempStreet(tempValue) VALUES ('Malton Court');
INSERT INTO TempStreet(tempValue) VALUES ('Little Sutton Grove');
INSERT INTO TempStreet(tempValue) VALUES ('Waterloo Woods');
INSERT INTO TempStreet(tempValue) VALUES ('Coleridge Leaze');
INSERT INTO TempStreet(tempValue) VALUES ('Highbury Drive');
INSERT INTO TempStreet(tempValue) VALUES ('Hanbury Broadway');
INSERT INTO TempStreet(tempValue) VALUES ('Trewern');
INSERT INTO TempStreet(tempValue) VALUES ('Kipling Cottages');
INSERT INTO TempStreet(tempValue) VALUES ('Trafalgar Gate');
INSERT INTO TempStreet(tempValue) VALUES ('Dunster Cottages');
INSERT INTO TempStreet(tempValue) VALUES ('David Broadway');
INSERT INTO TempStreet(tempValue) VALUES ('Avon Willows');
INSERT INTO TempStreet(tempValue) VALUES ('Alexandra Mead');
INSERT INTO TempStreet(tempValue) VALUES ('Howard Walk');
INSERT INTO TempStreet(tempValue) VALUES ('Minster Yard');
INSERT INTO TempStreet(tempValue) VALUES ('Raven Meols Lane');

```

```

INSERT INTO TempStreet(tempValue) VALUES ('Bernisdale Close');
INSERT INTO TempStreet(tempValue) VALUES ('Allington Firs');
INSERT INTO TempStreet(tempValue) VALUES ('Crofters Lodge');
INSERT INTO TempStreet(tempValue) VALUES ('Foxes Walk');
INSERT INTO TempStreet(tempValue) VALUES ('Richards Paddock');
INSERT INTO TempStreet(tempValue) VALUES ('Morgan Furlong');
INSERT INTO TempStreet(tempValue) VALUES ('Todholm Terrace');
INSERT INTO TempStreet(tempValue) VALUES ('Gorse Edge');
INSERT INTO TempStreet(tempValue) VALUES ('Raleigh Parkway');
INSERT INTO TempStreet(tempValue) VALUES ('Cemetery Paddocks');
INSERT INTO TempStreet(tempValue) VALUES ('Ridley Cedars');
INSERT INTO TempStreet(tempValue) VALUES ('Helston Promenade');
INSERT INTO TempStreet(tempValue) VALUES ('Cromer Elms');
INSERT INTO TempStreet(tempValue) VALUES ('Whitley Loan');
INSERT INTO TempStreet(tempValue) VALUES ('Johnston Croft');
INSERT INTO TempStreet(tempValue) VALUES ('Newman Grove');
INSERT INTO TempStreet(tempValue) VALUES ('Hardwick Las');
INSERT INTO TempStreet(tempValue) VALUES ('Selborne Paddock');
INSERT INTO TempStreet(tempValue) VALUES ('Alice Broadway');
INSERT INTO TempStreet(tempValue) VALUES ('Moorland Hollies');
INSERT INTO TempStreet(tempValue) VALUES ('Peveril Valley');
INSERT INTO TempStreet(tempValue) VALUES ('King George Lanes');
INSERT INTO TempStreet(tempValue) VALUES ('Brock Heights');
INSERT INTO TempStreet(tempValue) VALUES ('Hammercliffe Road');

/* DATA GENERATION PROCEDURE */
DECLARE @index INT
BEGIN
    SET @index = 0;

    WHILE(@index < 110000)
    BEGIN
        INSERT INTO dbo.Address (country, town, street, zip_code) VALUES (
            (SELECT tempValue FROM TempCountry WHERE tempID = CEILING(RAND() *
            (SELECT COUNT(*) FROM TempCountry))),
            (SELECT tempValue FROM TempTown WHERE tempID = CEILING(RAND() *
            (SELECT COUNT(*) FROM TempTown))),
            (SELECT tempValue FROM TempStreet WHERE tempID = CEILING(RAND() *
            (SELECT COUNT(*) FROM TempStreet))),
            CONCAT((FLOOR(RAND()*(100-10))+10), '-', (FLOOR(RAND()*(1000-
            100))+100))
        );
        SET @index = @index + 1;
    END
END

DROP TABLE TempCountry;
DROP TABLE TempStreet;
DROP TABLE TempTown;

```

## 4. Description of the statistics of the data

- **5 the most expensive beers**

```

SELECT TOP(5) name, price FROM Beer INNER JOIN BeerStyle ON (Beer.beer_style_id
= BeerStyle.id) ORDER BY Beer.price DESC;

```

Witbier	999.98
Pale lager	999.95

Witbier	999.93
Witbier	999.88
Weissbier	999.87

- **5 the cheapest beers**

```
SELECT stylename, beerprice FROM (SELECT * FROM beer INNER JOIN beerstyle USING
(beerstyleID) ORDER BY beerprice ASC) WHERE rownum <= 5;
```

Amber Ale	10,05
Brown Ale	10,14
Imperial Stout	10,17
English Bitter	10,29
Berliner Weisse	10,36

- **Average price for each beer style**

```
SELECT name, ROUND(AVG(price), 2) AS "Average price" FROM Beer JOIN BeerStyle
ON (BEER.beer_style_id = BeerStyle.id) GROUP BY name;
```

Amber Ale	513.220000
American Pale Ale	500.700000
Baltic Porter	518.560000
Berliner Weisse	516.460000
Biere de Garde	504.710000
Blonde Ale	505.680000
Bock	498.210000
Brown Ale	508.370000
Dark lager	502.650000
Doppelbock	498.740000
Double IPA	505.950000
Dunkel	500.690000
Dunkelweizen	498.270000
English Bitter	508.080000
ESB	504.950000
Foreign Stout	509.260000
Imperial Stout	502.520000
India Pale Ale	498.040000
Irish Dry Stout	503.930000
Maibock	495.910000
Marzen	505.400000
Pale lager	500.510000
Pilsener	507.580000
Porter	501.950000
Saison	505.640000
Stout	497.680000
Vienna lager	507.030000
Weissbier	506.250000
Witbier	508.020000

- **Beer with the biggest alcohol percentage value**

```
SELECT TOP(1) name, alcohol_percent FROM Beer JOIN beerStyle ON
(Beer.beer_style_id = BeerStyle.id) ORDER BY alcohol_percent DESC;
```

Berliner Weisse	12
-----------------	----

- **Beer with the lowest alcohol percentage value**

```
SELECT TOP(1) name, alcohol_percent FROM Beer JOIN beerStyle ON
(Beer.beer_style_id = BeerStyle.id) ORDER BY alcohol_percent ASC;
```

Weissbier	4.00
-----------	------

- **5 breweries with the biggest amount of beer**

```
SELECT TOP(5) Brewery.name, SUM(Beer.amount_in_liters) AS 'Total beer' FROM
Brewery JOIN BeerInBreweries ON (Brewery.id = BeerInBreweries.brewery_id) JOIN
Beer ON (Beer.id = BeerInBreweries.beer_id) GROUP BY Brewery.name ORDER BY
'Total beer' DESC;
```

Dodger Blue Bootle Brewhouse	22805.25
Medium Sea Green Cap Corporation	22212.73
Dark Orange Box Firm	21208.33
Pale Violet Red Kettle Company	21162.35
Khaki Kettle Organization	20987.16

- **10 breweries with the biggest kegs capacity**

```
SELECT TOP(10) Brewery.name, SUM(Keg.capacity) AS 'Kegs capacity' FROM Brewery
JOIN BeerInBreweries ON (Brewery.id = BeerInBreweries.brewery_id) JOIN
BeerInKegs ON (BeerInBreweries.beer_id = BeerInKegs.beer_id) JOIN Keg ON
(BeerInKegs.keg_id = Keg.id) GROUP BY Brewery.name ORDER BY 'Kegs capacity'
DESC;
```

Medium Sea Green Cap Corporation	1930
Misty Rose Bootle Brewhouse	1835
Pink Kettle Corporation	1735
Green Yellow Can Business	1720
Cornflower Blue Opener Business	1715
Alice Blue Can Corporation	1695
Dodger Blue Bootle Brewhouse	1650
Khaki Kettle Organization	1645
Ivory Mug Business	1640
Dark Orange Box Firm	1605

- **10 the best paid employees**

```
SELECT TOP(10) Employee.first_name, Employee.last_name, Employee.salary,
Brewery.name FROM Employee JOIN Brewery ON (Employee.brewery_id = Brewery.id)
ORDER BY Employee.salary DESC;
```

Harding	Brady	14999.96	Dark Gray Mug Brewhouse
Roger	Figuroa	14999.84	Gainsboro Bootle Institution



Lowe	Robles	14999.67	Yellow Green Cap Firm
Kerr	Rodgers	14999.65	Coral Mug Brewery
Filip	Watkins	14999.38	Misty Rose Bootle Brewhouse
Dillan	Thomson	14999.35	Dark Grey Keg Institution
Peter	Mccarthy	14999.09	Navajo White Cap Organization
Mackenzie	Bailey	14999.07	Blue Kettle Brewhouse
Dexter	Maxwell	14998.93	Powder Blue Keg Institution
Ferguson	Hawkins	14998.73	Midnight Blue Mug Corporation