

C3KDH1

Databashantering, 7,5 hp NGWEK18h, NGWDK18h

Inlämning: aggregering av data med SQL (1hp)

Benny Zhang, s182343

1.

```
SELECT GROUP_CONCAT(gender, ": ", amountGender SEPARATOR ' || ') AS  
genderCount  
FROM (  
    SELECT gender, COUNT(gender) AS amountGender  
    FROM customer  
    GROUP BY gender  
    HAVING COUNT(gender) > 0) AS i
```

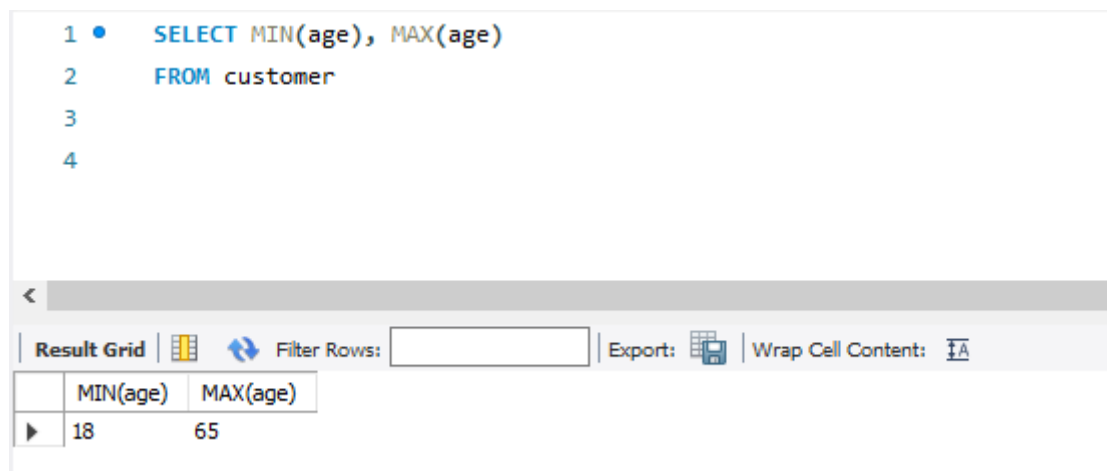


```
1 • SELECT
2     GROUP_CONCAT(gender, ': ', amountGender
3         SEPARATOR ' || ') AS genderCount
4 FROM
5     (SELECT
6         gender, COUNT(gender) AS amountGender
7     FROM
8         customer
9     GROUP BY gender
10    HAVING COUNT(gender) > 0) AS i
11
```

genderCount
Male: 478 Female: 522

2.

```
SELECT MIN(age), MAX(age)
FROM customer
```



```
1 • SELECT MIN(age), MAX(age)
2 FROM customer
3
4
```

MIN(age)	MAX(age)
18	65

3.

```
SELECT COUNT(DISTINCT(age))
FROM customer
```

1 • SELECT COUNT(DISTINCT(age))

2 FROM customer

3

4

5

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

COUNT(DISTINCT(age))
48

4.

```
SELECT group_concat(country, ": ", averageAge SEPARATOR ' || ') AS
averageAgeCountry
```

```
FROM (
```

```
    SELECT country, AVG(age) AS averageAge
```

```
    FROM customer
```

```
    GROUP BY country) AS y
```

1 • SELECT group_concat(country, ": ", averageAge SEPARATOR ' || ') AS averageAgeCountry

2 FROM (

3 SELECT country, AVG(age) AS averageAge

4 FROM customer

5 GROUP BY country) AS y

6

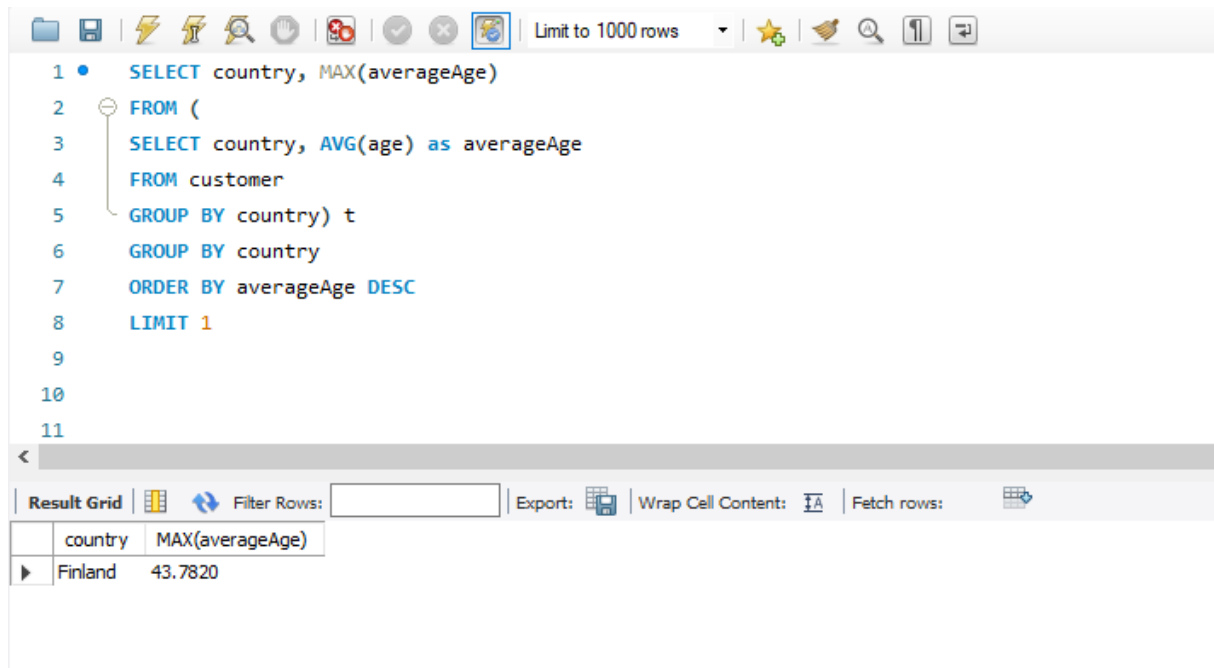
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

averageAgeCountry
Sweden: 41.4243 Finland: 43.7820 Norway: 42.7288 Denmark: 42.9375 Iceland: 41.1176

5.

```
select country, MAX(averageAge)
FROM (
    SELECT country, AVG(age) as averageAge
    FROM customer
    GROUP BY country) t
GROUP BY country
```

ORDER BY averageAge DESC
LIMIT 1



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and settings. A dropdown menu shows 'Limit to 1000 rows'. The SQL editor contains the following query:

```
1 • SELECT country, MAX(averageAge)
2 FROM (
3 SELECT country, AVG(age) as averageAge
4 FROM customer
5 GROUP BY country) t
6 GROUP BY country
7 ORDER BY averageAge DESC
8 LIMIT 1
9
10
11
```

Below the editor is a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, a 'Wrap Cell Content' toggle, and a 'Fetch rows' button. The results table has two columns: 'country' and 'MAX(averageAge)'. The first row shows 'Finland' with a value of '43.7820'.

country	MAX(averageAge)
Finland	43.7820

6.
SELECT COUNT(*)
FROM
(
SELECT
country
FROM
customer
GROUP BY country
HAVING count(id) > 100) y

```

1 • SELECT COUNT(*)
2 FROM
3 (
4 SELECT
5 country
6 FROM
7 customer
8 GROUP BY country
9 HAVING count(id) > 100) y
10
11

```

Result Grid

	COUNT(*)
▶	3

7.

```

SELECT shirt_size, COUNT(*) AS customers
FROM customer
WHERE age BETWEEN 25 AND 35
GROUP BY shirt_size
ORDER BY customers DESC
LIMIT 1;

```

```

1 • SELECT shirt_size, COUNT(*) AS customers
2 FROM customer
3 WHERE age BETWEEN 25 AND 35
4 GROUP BY shirt_size
5 ORDER BY customers DESC
6 LIMIT 1;
7
8

```

Result Grid

	shirt_size	customers
▶	L	41

8.

```

SELECT phrase
FROM customer
ORDER BY LENGTH(phrase) DESC

```

LIMIT 1;

The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```
1 • SELECT phrase
2 FROM customer
3 ORDER BY LENGTH(phrase) DESC
4 LIMIT 1;
```

Below the query editor is a result grid. The first row has a column header 'phrase'. The first data row contains the text 'Business-focused demand-driven artificial intelligence'.

phrase
Business-focused demand-driven artificial intelligence

9.
SELECT shirt_size, COUNT(*) as customers
FROM customer
WHERE length(phrase) >= 30
GROUP BY shirt_size
ORDER BY customers DESC
LIMIT 1;

The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```
1 • SELECT shirt_size, COUNT(*) as customers
2 FROM customer
3 WHERE length(phrase) >= 30
4 GROUP BY shirt_size
5 ORDER BY customers DESC
6 LIMIT 1;
```

Below the query editor is a result grid. The first row has column headers 'shirt_size' and 'customers'. The first data row shows '2XL' for shirt_size and '122' for customers.

shirt_size	customers
2XL	122

10.

```
SELECT
    GROUP_CONCAT('Name: ',
        first_name,
        ', ',
        last_name,
        ', From: ',
        country,
        ', Max age: ',
        age
        SEPARATOR ' || ') AS EldestCustomerCountry
FROM
    (SELECT
        x.first_name, x.last_name, x.country, x.age
    FROM
        customer x
    JOIN (SELECT
        country, MAX(age) AS maxAge
    FROM
        customer
    GROUP BY country) AS y ON x.country = x.country
    AND x.age = y.maxAge
    GROUP BY country
    ORDER BY country ASC) AS i
```

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the query from the previous block, with line numbers 1 through 24. The bottom section displays the 'Result Grid' with a table titled 'EldestCustomerCountry'. The table has one row of data, which is a long string concatenating the names and ages of the oldest customer in each country, separated by ' || '.

EldestCustomerCountry
Name: Quilan Crysell, From: Denmark, Max age: 65 Name: Farris Lund, From: Finland, Max age: 65 Name: Ame Richold, From: Iceland, Max age: 64 Name: Cecelia Cometto, From: Norway, Max age: 65 Name: Parry Nowalowska, From: Sweden, Max age: 65