L6: Full-Relation Queries

CS1106/CS6503: Intro to Relational Databases

Dr Kieran T. Herley

Semester One, 2023-24

School of Computer Science & Information Technology
University College Cork

Summary

SQL's aggregation functions. Aggregation and grouping.

Aggregation and Grouping

- Our queries to data all relate to individual rows within tables (possibly more than one), not queries that "summarize" data from the entire table (or sections thereof)
- Today's DB contains stats on the countries of the world (population, GDP etc.)
- How to answer questions like:
  - What is the population of Europe?
  - What country has the largest GDP?
  - Which region in the world has the greatest area?

Today's Database

countries

name	*region*	*area*	*population*	*gdp*
Afghanistan	South Asia	652225	26000000	NULL
Albania	Europe	28728	3200000	6656000000
Algeria	Middle East	2400000	32900000	75012000000
⋮	⋮	⋮	⋮	⋮
Ireland	Europe	70182	4000000	137120000000
⋮	⋮	⋮	⋮	⋮
Zimbabwe	Africa	390759	12900000	6192000000

A Closer Look At `countries`

countries

name	*region*	*area*	*population*	*gdp*
Afghanistan	South Asia	652225	26000000	NULL
Albania	Europe	28728	3200000	6656000000
Algeria	Middle East	2400000	32900000	75012000000
⋮	⋮	⋮	⋮	⋮
Ireland	Europe	70182	4000000	137120000000
⋮	⋮	⋮	⋮	⋮
Zimbabwe	Africa	390759	12900000	6192000000

1

name countries name (as VARCHAR)

region surrounding region (as VARCHAR)

area in km^2 (as DECIMAL)

population (as DECIMAL)

gdp Gross Domestic Product in dollars; (as DECIMAL)

[1] Data may be somewhat "stale", but will suffice for target practice.

What Is The Population Of China?

- China's population:

```
SELECT name, population
FROM countries
WHERE name = 'China';
```

What Is The Population Of China?

- China's population:

```
SELECT name, population
FROM countries
WHERE name = 'China';
```

- What does this do?

```
SELECT name, area/population
FROM countries
WHERE region = 'Europe'
ORDER BY area/population DESC;
```

Note the use of *expressions* like `area/population` in `SELECT` and `ORDER BY` clauses

Whole-Table Queries Using Aggregates

How Many Countries Are There?

- Desired result summarizes info. from entire table not just one or two rows.

How Many Countries Are There?

- Desired result summarizes info. from entire table not just one or two rows.
-

```
SELECT COUNT(name)
FROM countries;
```

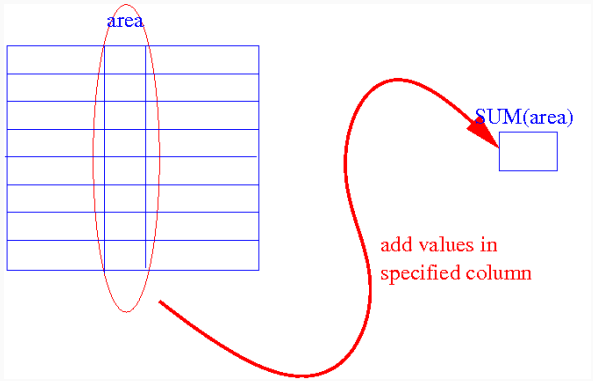
COUNT(name)
189

- Counts the number of values in the `name` column
- Result (shown right) is a 1×1 table!
- Could use `COUNT(*)` instead of `COUNT(name)` here

How Many Countries Are There cont'd

-

```
SELECT SUM(area)
FROM countries;
```



- SQL has five main *aggregation* operators:
  SUM sum of column values
  AVG average of column values
  MIN minimum of column values
  MAX maximum of column values
  COUNT count of values in column with duplicates counted every time they appear
- All these can be used to compute aggregates of *groups* of rows sharing some common characteristics

Some Whole-Table Aggregation Queries

- What is the total population of the world?

```
SELECT SUM (population)
FROM countries;
```

Some Whole-Table Aggregation Queries

- What is the total population of the world?

```
SELECT SUM (population)
FROM countries;
```

- What is the largest GDP?

```
SELECT MAX (gdp)
FROM countries;
```

Some Whole-Table Aggregation Queries

- What is the total population of the world?

```
SELECT SUM (population)
FROM countries;
```

- What is the largest GDP?

```
SELECT MAX (gdp)
FROM countries;
```

Some Whole-Table Aggregation Queries

- How many regions are there?

```
SELECT COUNT(region)
FROM countries;
```

Wrong!

Some Whole-Table Aggregation Queries

- How many regions are there?

```
SELECT COUNT(region)
FROM countries;
```

Wrong!counts the num. values in `region` column including duplicates

- Second go at "How many regions are there?"

```
SELECT COUNT(DISTINCT region)
FROM countries;
```

  - `DISTINCT` ensures each region is counted only once
  - Can use this with any aggregate operator but only makes sense with `COUNT`

GROUPING

What is the Breakdown of Population By Region?

- Can specify grouping of rows using `GROUP BY` and then the aggregates apply to each group separately
-

```
SELECT SUM(population)
FROM countries
GROUP BY region;
```

SUM(population)
668981725
7432900
2173791900
818020000
373882000
433400000
358510000
1462138000

- Conceptually this works as follows:
  - Rows with same region value are bunched together into groups
  - `SUM` applied to each group separately
  - One row per group appears in result
- When we omit the `GROUP BY` clause, default group (the entire table) is used for any aggregate operators

What is the Breakdown cont'd

```
SELECT SUM(population)
FROM countries
GROUP BY region;
```

countries

name	···	*population*	···
Afghanistan		26000000	
Albania	···	3200000	···
Algeria	···	32900000	···
⋮	⋮	⋮	⋮
Ireland	···	4000000	···
⋮		⋮	⋮
Zimbabwe	···	12900000	···

The original table

Notes: the original table remains unaltered– the "grouping" is notional

What is the Breakdown cont'd

```
SELECT SUM(population)
FROM countries
GROUP BY region;
```

name	···	*population*	···
Angola	···	14500000	···
⋮	⋮	⋮	⋮
Zimbabwe	···	12900000	···
Antigua	···	77000	···
⋮	⋮	⋮	⋮
St Vincent	···	121000	···
⋮	⋮	⋮	⋮

Grouped by region

Notes: the original table remains unaltered– the "grouping" is notional

What is the Breakdown cont'd

```
SELECT SUM(population)
FROM countries
GROUP BY region;
```

SUM(pop)
668981725
7432900
2173791900
818020000
433400000
358510000
1462138000

Result containing one row per group, e.g. sum of 'Africa'

What is the Breakdown cont'd

- Can include the grouping attribute in SELECT clause for readability
-

```
SELECT region, SUM(population)
FROM countries
GROUP BY region;
```

region	SUM(populatio
Africa	668981725
Americas	7432900
Asia-Pacific	2173791900
⋮	⋮

- SELECT clause can include only
  1. aggregate terms (e.g. `SUM(. . .)`)
  2. un-aggregated attributes (e.g. `region`), but this only makes sense for attributes mentioned in `GROUP BY` clause

What Country Has The Largest Population?

- First attempt:

```
SELECT name, MAX(population)
FROM countries;
```

What Country Has The Largest Population?

- First attempt:

```
SELECT name, MAX(population)
FROM countries;
```

Wrong!

- Some versions of SQL object (syntax error) to inappropriate mixing of aggregates and non-aggregates in `SELECT` clause with no GROUP columns specified
- Others includes *arbitrary* value from `name` column not necessarily the one corresponding to largest population:

name	**MAX(population)**
Albania	1300000000

- Will return to this later . . .

Some More Examples

- List max, min and average and total GDP for each region

```
SELECT region, MAX(gdp), MIN(gdp), AVG(gdp), SUM(gdp)
FROM countries
GROUP BY region;
```

- List Max, min and average and total GDP for regions Europe and North America

```
SELECT region, MAX(gdp), MIN(gdp), AVG(gdp), SUM(gdp)
FROM countries
WHERE region IN ('Europe', 'North America')
GROUP BY region;
```

Some More Examples

List the regions and size (num of countries) for all regions of size ≥ 10

```
SELECT region, COUNT(*)
FROM countries
WHERE COUNT(*) >= 10
GROUP BY region;
```

Wrong!

- WHERE conditions apply to individual table rows
- Can't use aggregates in `WHERE` clause

HAVING

What Regions Contain At Least 10 Countries?

- Want to filter by some characteristic of groups rather than of individual rows, the group size in the previous example

What Regions Contain At Least 10 Countries?

- Want to filter by some characteristic of groups rather than of individual rows, the group size in the previous example
-

```
SELECT region, COUNT(*)
FROM countries
GROUP BY region;
```

Counts num. countries for all regions

```
SELECT region, COUNT(*)
FROM countries
GROUP BY region
HAVING COUNT(*) >= 10;
```

Counts num. countries for regions with at least ten countries

- Aggregates in HAVING clause apply to the group in question (here `region` groups); The only attributes that can appear in the HAVING clause un-aggregated are those mentioned in the GROUP BY clause (*i.e.* `region` in this case).

Which Regions Have Populations in Excess of a Billion?

- ```
  SELECT region, SUM(population)
  FROM countries
  GROUP BY region
  HAVING SUM(population) > 1000000000;
  ```
  ```

What Is The Average GDP Of Regions With Large Populations

- List the number of countries, average GDP and total area of all regions that have a population greater than one billion
-

```
SELECT region, COUNT(*), AVG(gdp), SUM(area)
FROM countries
GROUP BY region
HAVING SUM(population) > 1000000000;
```

Which Regions Are Economically Homogeneous?

- List the max and min per capita GDP for all regions where the maximum GDP per capita is at most five times the the minimum GDP per capita
-

$$\text{Per capita GDP} = \text{GDP}/\text{population}$$

Which Regions Are Economically Homogeneous?

- List the max and min per capita GDP for all regions where the maximum GDP per capita is at most five times the the minimum GDP per capita
- $$\text{Per capita GDP} = \text{GDP}/\text{population}$$
- ```
  SELECT region, MIN(gdp/population), MAX(gdp/population)
  FROM countries
  GROUP BY region
  HAVING MAX(gdp/population) < 5 * MIN(gdp/population);
  ```
  ```

Which Country Has The Large Population?

- ```
  SELECT name, population
  FROM countries
  WHERE population = MAX(population);
  ```

  Wrong!– Can't use aggregates in `WHERE` clause

- ```
  SELECT name, population
  FROM countries
  HAVING population = MAX(population);
  ```

  Wrong!– Can't use (un-aggregated) attributes in `HAVING` clause other than those in `GROUP BY` clause (here none)

Which Country Has The Largest Population?

- Finally an approach that works:

```
SELECT name, population
FROM countries
WHERE population = (SELECT MAX(population) FROM countr
```

- Uses *subqueries* (seen later); "inner query" (within parentheses) determines max value and "outer query" uses it to determine associated country

Notes and Acknowledgements

The `countries` DB was taken from the website `www.sqlzoo.net`.
The data itself was ultimately taken from the BBC.