Functions and Summary Queries

SQL SELECT Statement

- Has 6 main clauses
  - `SELECT` Specifies which columns are to appear in output.
  - `FROM` Specifies table(s) to be used.
  - `WHERE` Filters rows.
  - `GROUP BY` Forms groups of rows with same column value.
  - `HAVING` Filters groups subject to some condition.
  - `ORDER BY` Specifies the order of the output.

Functions

CHAR Function

- Typically used to output characters that can't be typed
- Examples for common control characters
  - Tab: CHAR(9)
  - Line feed: CHAR(10)
  - Carriage return: CHAR(13)

FORMAT function

- Convert a number into a string of characters
  - Uses commas to separate the thousands
  - Rounds to the specified number of decimal places
- Syntax
  FORMAT(number,decimal)
- Examples:
  - `FORMAT(1234567.8901,2)`
    `1,234,567.89`
  - `FORMAT(1234.56,4)`
    `1,234.5600`
  - `FORMAT(1234.56,0)`
    `1,235`

Some String Functions

- CONCAT
- CONCAT_WS(sep,str1[,str2]...)
- LTRIM(str)
- RTRIM(str)
- TRIM(str)
- LENGTH(str)
- LOCATE(find,search[,start])
- LEFT(str,length)
- RIGHT(str,length)
- SUBSTRING_INDEX(str,delimiter, count)
- SUBSTRING(str,start[,length])

More String Functions

- REPLACE(search,find,replace)
- INSERT(str,start,length,insert)
- REVERSE(str)
- LOWER(str)
- UPPER(str)
- LPAD(str,length,pad)
- RPAD(str,length,pad)
- SPACE(count)
- REPEAT(str,count)

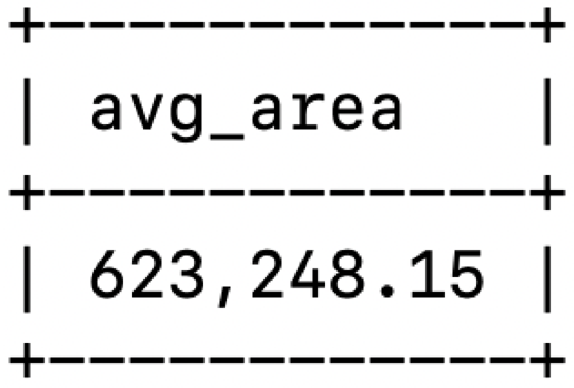
Some numeric functions

- ROUND(number[,length])
- TRUNCATE(number,length)
- CEILING(number)
- FLOOR(number)
- ABS(number)
- SIGN(number)
- SQRT(number)
- POWER(number,power)
- RAND()

Example: Format Function

- Find the average area of all countries

```
SELECT FORMAT(AVG(area),2) AS avg_area
FROM countries;
```



```
+--------------+
| avg_area     |
+--------------+
| 623,248.15 |
+--------------+
```

Current date and time functions

- Return current local date and time based on the system clock
  - NOW()
  - SYSDATE()
  - CURRENT_TIMESTAMP()
- Return the current local date
  - CURDATE()
  - CURRENT_DATE()
- Return the current local time
  - CURTIME()
  - CURRENT_TIME()
- Returns GMT date and time
  - UTC_DATE()
  - UTC_TIME()

CURRENT_TIMESTAMP, CURRENT_DATE and CURRENT_TIME are ANSI standard

Some date/time parsing functions

- DAYOFMONTH(date)
- MONTH(date)
- YEAR(date)
- HOUR(time)
- MINUTE(time)
- SECOND(time)
- DAYNAME(date)
- MONTHNAME(date)

Formatting dates

- DATE_FORMAT(date, format)
- Common codes for date format strings
  - Month, numeric (01…12): %m
  - Month, numeric (1…12): %c
  - Month name (January…December): %M
  - Abbreviated month name (Jan…Dec): %b
  - Day of the month, numeric (00…31): %d
  - Day of the month, numeric (0…31): %e
  - Day of the month with suffix (1st, 2nd, 3rd, etc.): %D
  - Year, numeric, 2 digits: %y
  - Year, numeric, 4 digits: %Y
  - Weekday name (Sunday…Saturday): %W
  - Abbreviated weekday name (Sun…Sat): %a

Examples: Date Formatting

- Format current date as 10/05/21
- Format current date as October 5, 2021
- Format current date as 05-Oct-2021
- Format current date as Tuesday, October 5th

Formatting times

- TIME_FORMAT(time, format)
- Common codes for date/time format strings (continued)
  - Hour (00…23): %H
  - Hour (0…23): %k
  - Hour (01…12): %h
  - Hour (1…12): %l
  - Minutes (00…59): %i
  - Time, 12-hour (hh:mm:ss AM or PM): %r
  - Time, 24-hour (hh:mm:ss): %T
  - Seconds (00…59): %S
  - AM or PM: %p

Examples: Time Formatting

- Format the current time as
  - 14:45:00
  - 2:45 PM

Calculating dates and times

- DATE_ADD(date, INTERVAL expression unit)
- DATE_SUB(date, INTERVAL expression unit)
- DATEDIFF(date1, date2)
- TO_DAYS(date)
- TIME_TO_SEC(time)

Summary Queries

Aggregate functions

- Operate on a series of values and return a single value
- Query that contains one or more aggregate functions is called a summary query

Aggregate Functions in SQL

- ISO standard defines five aggregate functions:
  - COUNT returns number of values in specified column.
  - SUM returns sum of values in specified column.
  - AVG returns average of values in specified column.
  - MIN returns smallest value in specified column.
  - MAX returns largest value in specified column.
- Each operates on a single column of a table and returns a single value.

Non-numeric Columns

- AVG and SUM can only be used on columns that contain numeric data
- MIN, MAX and COUNT can be used on columns containing numbers, date or string values

SELECT Statement - Aggregates

- Aggregate functions can be used only in SELECT list and in HAVING clause.
- A SELECT list cannot reference a column without with an aggregate function unless there is a GROUP BY clause
- For example, the following is illegal:

```
SELECT country_id, COUNT(area)
FROM countries;
```

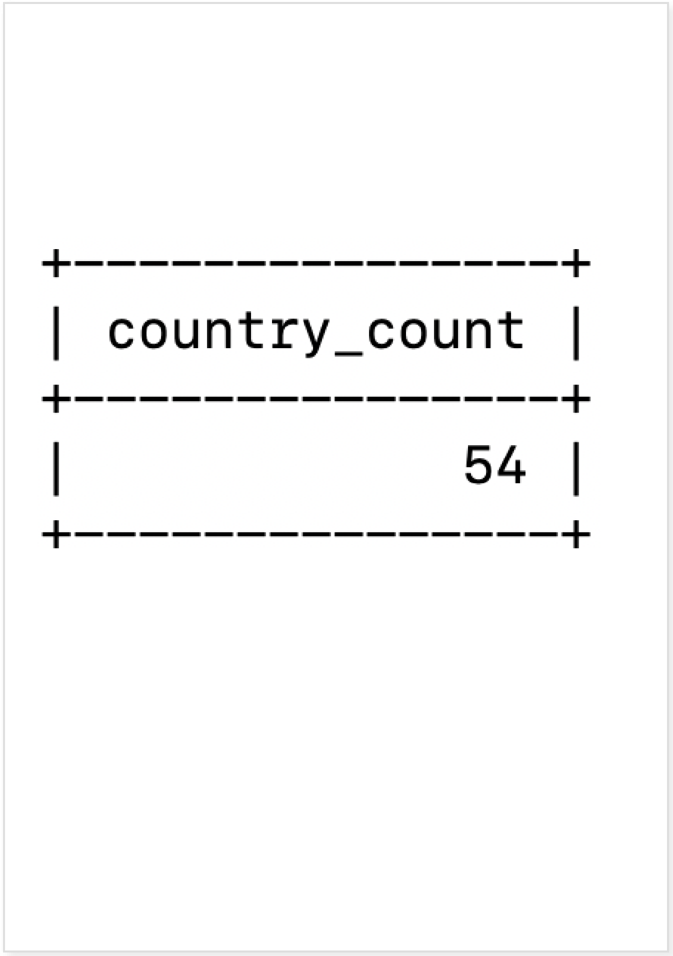
COUNT

- COUNT(*) counts all rows of a table, regardless of whether nulls or duplicate values occur.
- Can use DISTINCT before column name to eliminate duplicates.
- DISTINCT has no effect with MIN/MAX, but may have with SUM/AVG.

Example: Use of COUNT(*)

- How many countries have an area of less than 1,000 square kilometers

```
SELECT COUNT(*) as
country_count FROM countries
WHERE area < 1000;
```



```
+-----------------+
| country_count   |
+-----------------+
|              54 |
+-----------------+
```

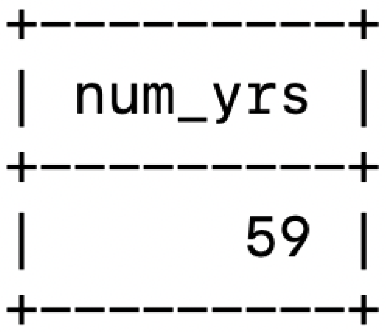
DISTINCT

- The DISTINCT keyword prevents duplicate rows from being included in the result set
- In summary queries, all values are included regardless of whether they are duplicated

Example: Use of COUNT(DISTINCT col)

- How many different years are there in country_stats

```
SELECT count(DISTINCT year) AS num_yrs
FROM country_stats;
```

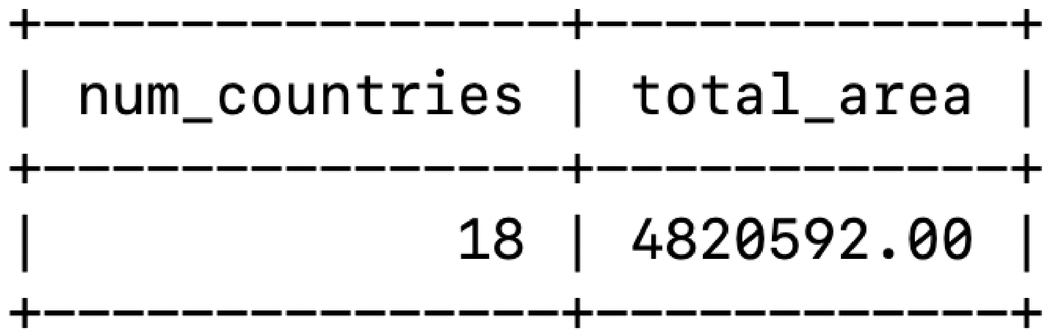


```
+---------+
| num_yrs |
+---------+
|      59 |
+---------+
```

Example: Use of COUNT and SUM

- Find number of countries in and total area of region 5

```
SELECT COUNT(country_id) AS
num_countries,
SUM(area) AS total_area
FROM countries
WHERE region_id = 5;
```

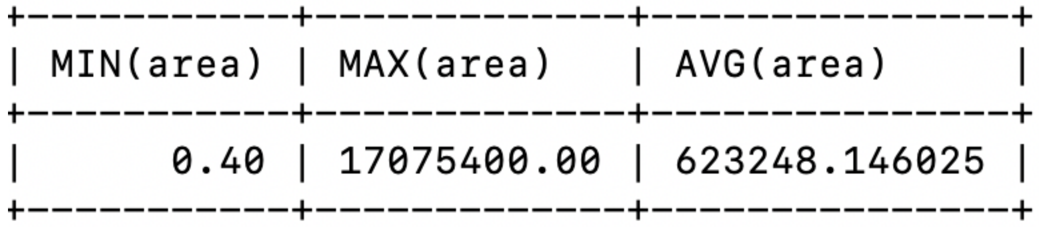


```
+---------------+------------+
| num_countries | total_area |
+---------------+------------+
|            18 | 4820592.00 |
+---------------+------------+
```

Example: Use of MIN, MAX, AVG

- Find the min, max and average area

```
SELECT MIN(area), MAX(area), AVG(area)
FROM countries;
```



```
+-----------+-------------+---------------+
| MIN(area) | MAX(area)   | AVG(area)     |
+-----------+-------------+---------------+
|      0.40 | 17075400.00 | 623248.146025 |
+-----------+-------------+---------------+
```

Exercise - Summary Queries

- Write a query to find the number of languages in the languages table
- Write a query to find the number of different regions in the countries table
- Find the earliest national day of all countries using an aggregate function (Don’t sort and limit).
- Find the number of countries **without** a national day.

Grouping

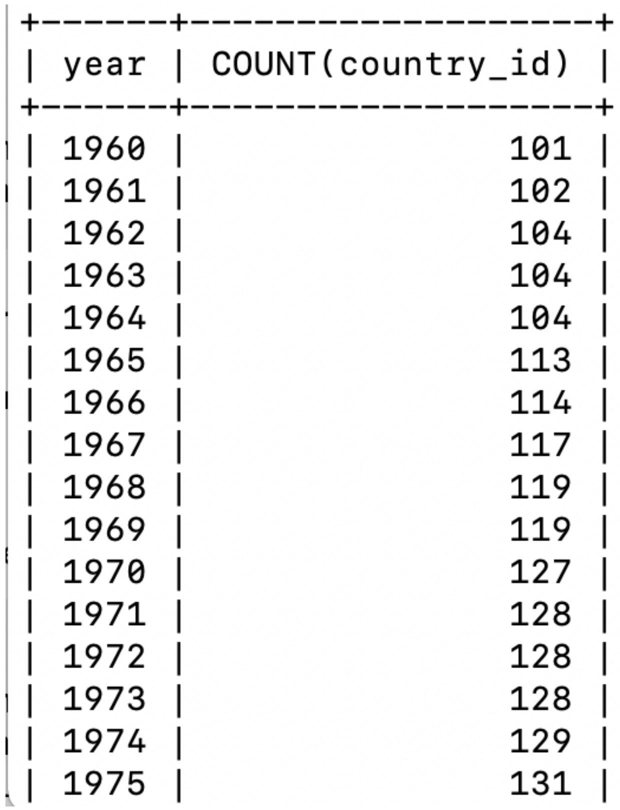
SELECT Statement - Grouping

- Use GROUP BY clause to get subtotals.
- SELECT and GROUP BY closely integrated
  - each item in SELECT list must be single-value per group
- SELECT clause may only contain:
  - column names
  - aggregate functions
  - constants
  - expression involving combinations of the above.
- The GROUP BY clause follows the WHERE clause and precedes the ORDER BY clause
- The default sort order is ascending

Example: Use of GROUP BY

- Find the number of countries with statistics in each year.

```
SELECT year, COUNT(country_id)
FROM country_stats
GROUP BY year;
```



```
+------+-------------------+
| year | COUNT(country_id) |
+------+-------------------+
| 1960 |               101 |
| 1961 |               102 |
| 1962 |               104 |
| 1963 |               104 |
| 1964 |               104 |
| 1965 |               113 |
| 1966 |               114 |
| 1967 |               117 |
| 1968 |               119 |
| 1969 |               119 |
| 1970 |               127 |
| 1971 |               128 |
| 1972 |               128 |
| 1973 |               128 |
| 1974 |               129 |
| 1975 |               131 |
```

SELECT Statement - Grouping

- All column names in SELECT list must appear in GROUP BY clause unless name is used only in an aggregate function.
- If WHERE is used with GROUP BY, WHERE is applied first, then groups are formed from remaining rows satisfying predicate.
- ISO considers two nulls to be equal for purposes of GROUP BY.

Exercise 2 – Grouping Queries

- Find the number of countries in each region.
- Find the average population for each year

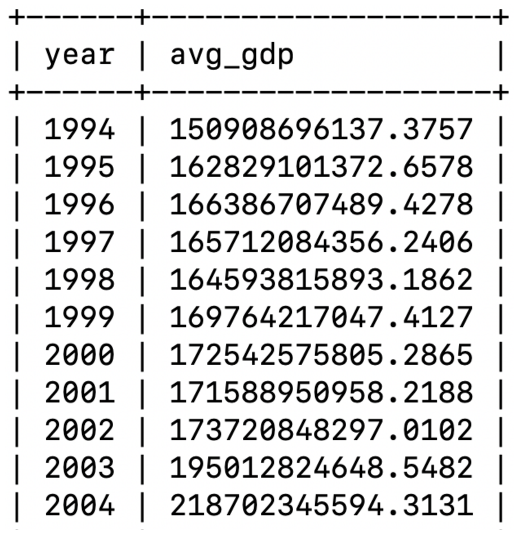
Excluding Groups – HAVING clause

- HAVING clause is designed for use with GROUP BY to restrict groups that appear in final result table.
- Similar to WHERE, but WHERE filters individual rows whereas HAVING filters groups.
- Column names in HAVING clause must also appear in the GROUP BY list or be contained within an aggregate function.

Example: Use of HAVING

- For each year with statistics for more than 180 countries, find the average GDP for each year.

```
SELECT year, AVG(GDP) as
avg_gdp FROM country_stats
GROUP BY year
HAVING COUNT(country_id) > 180;
```



```
+------+-------------------+
| year | avg_gdp           |
+------+-------------------+
| 1994 | 150908696137.3757 |
| 1995 | 162829101372.6578 |
| 1996 | 166386707489.4278 |
| 1997 | 165712084356.2406 |
| 1998 | 164593815893.1862 |
| 1999 | 169764217047.4127 |
| 2000 | 172542575805.2865 |
| 2001 | 171588950958.2188 |
| 2002 | 173720848297.0102 |
| 2003 | 195012824648.5482 |
| 2004 | 218702345594.3131 |
```

Exercise – Having Queries

- Find regions that have more than 10 or more countries and the total area of the region.
- Find the number of countries and the average area for each region. Include only those regions with an average area greater than 200,000 square km.

Exercise - more summary queries

- Find the number of countries with an area less than 1000 square km.
- Find the lowest GDP in the statistics table.
- Find the average area in each region for countries with an area of greater than 5000 square km
- Find those regions where the average area is more than 100,000 square km.