

Test report

1. Database schema:

id	name	population	COUNTRY
1	2110 State of Illinois	128120000	USA
2	2115 Mexico	128120000	MEX
3	2118 Mexico	128120000	MEX
4	2120 Mexico	128120000	MEX
5	2122 Mexico	128120000	MEX
6	2124 Mexico	128120000	MEX
7	2126 Mexico	128120000	MEX
8	2128 Mexico	128120000	MEX
9	2130 Mexico	128120000	MEX
10	2132 Mexico	128120000	MEX

id	name	percentage	COUNTRY
1	2110 State of Illinois	12.812	USA
2	2115 Mexico	12.812	MEX
3	2118 Mexico	12.812	MEX
4	2120 Mexico	12.812	MEX
5	2122 Mexico	12.812	MEX
6	2124 Mexico	12.812	MEX
7	2126 Mexico	12.812	MEX
8	2128 Mexico	12.812	MEX
9	2130 Mexico	12.812	MEX
10	2132 Mexico	12.812	MEX

language	count
English	4075
Spanish	239
French	964

2. Queries

- Most popular languages by region (version which works on every database engine)

```
SELECT language.name AS language_name, MAX(language_aggregate.region) AS region,
MAX(language_aggregate.language_population) AS language_population FROM (
```

```
SELECT language_sum.region, MAX(language_sum.language_population) AS
language_population FROM (
```

```
SELECT lang.name, cntr.region, sum((lang.percentage / 100) * cntr.population) AS
language_population
```

```
FROM language lang
```

```
JOIN country cntr
```

```
ON lang.country = cntr.code
```

```
GROUP BY lang.name, cntr.region) language_sum
```

```
GROUP BY language_sum.region) language_aggregate
```

```
JOIN country
```

```
ON country.region = language_aggregate.region
```

```
JOIN language
```

```
ON language.country = country.code
```

```
GROUP BY language.name, country.region
```

```
HAVING SUM((language.percentage / 100) * country.population) =
MAX(language_aggregate.language_population);
```

	LANGUAGE_NAME	REGION	LANGUAGE_POPULATION
1	Javanese	Southeast Asia	83570158
2	Papuan Languages	Melanesia	3792451
3	English	Australia and New Zealand	18695372
4	Russian	Eastern Europe	148136548
5	English	Micronesia/Caribbean	94
6	Arabic	Middle East	89153760
7	German	Western Europe	87155999,6
8	Portuguese	South America	166037997
9	Spanish	Caribbean	21530888
10	English	North America	258821522
11	Chinese	Eastern Asia	1175677671
12	Arabic	Northern Africa	140084044
13	Oromo	Eastern Africa	19395150
14	Kiribati	Micronesia	84235
15	Kongo	Central Africa	11480181
16	Samoan	Polynesia	147108
17	Zulu	Southern Africa	9508689
18	Hausa	Western Africa	29225396
19	Spanish	Central America	122159129
20	Swedish	Nordic Countries	8255142,6
21	English	British Islands	61728266,6
22	Hindi	Southern and Central Asia	405169038
23	Italian	Southern Europe	54303880
24	Lithuanian	Baltic Countries	3047066,4

Database Statistics

POSTGRESQL BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=70, operationsPerSecond=42.857142857142854)
ORACLE BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=82, operationsPerSecond=36.58536585365854)
MYSQL BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=151, operationsPerSecond=19.867549668874172)
H2 BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=1658, operationsPerSecond=1.8094089264173705)

- **Most popular languages by region (version which works only on Oracle and PostgreSQL)**

```
SELECT t.name, t.region, t.language_population
FROM (
SELECT l.name, c.region, ((l.percentage * c.population)/100) language_population,
ROW_NUMBER() OVER (PARTITION BY c.region ORDER BY ((l.percentage *
c.population)/100) DESC) row_number
FROM language l
JOIN country c ON c.code = l.country) t WHERE t.row_number = 1
```

Database Statistics

POSTGRESQL BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=69, operationsPerSecond=43.47826086956521, exception=null)

ORACLE BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=50, operationsPerSecond=60.0, exception=null)

H2 BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=3, operationsPerSecond=1.7976931348623157E308, exception=java.lang.RuntimeException: org.h2.jdbc.JdbcSQLException: Syntax error in SQL statement "SELECT T.NAME, T.REGION, T.LANGUAGE_POPULATION FROM (SELECT L.NAME, C.REGION, ((L.PERCENTAGE * C.POPULATION)/100) LANGUAGE_POPULATION, ROW_NUMBER() OVER (PARTITION[*] BY C.REGION ORDER BY ((L.PERCENTAGE * C.POPULATION)/100) DESC) ROW_NUMBER FROM LANGUAGE L JOIN COUNTRY C ON C.CODE = L.COUNTRY) T WHERE T.ROW_NUMBER = 1 "; expected "); SQL statement: SELECT t.name, t.region, t.language_population FROM (SELECT l.name, c.region, ((l.percentage * c.population)/100) language_population, ROW_NUMBER() OVER (PARTITION BY c.region ORDER BY ((l.percentage * c.population)/100) DESC) row_number FROM language l JOIN country c ON c.code = l.country) t WHERE t.row_number = 1 (42001-197)

MYSQL BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=3, operationsPerSecond=1.7976931348623157E308, exception=java.lang.RuntimeException: com.mysql.jdbc.exceptions.jdbc4.MySQLSyntaxErrorException: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'row_number FROM language l JOIN country c ON c.code = l.country) t WHERE t.ro' at line 3)

- **Cities from 10 biggest countries**

```

SELECT town.*
FROM country
JOIN town
ON town.country = country.code
WHERE (
    SELECT COUNT(*)
    FROM country cntr
    WHERE cntr.surfacearea >= country.surfacearea
) <= 10
ORDER BY town.population DESC;

```

ID	DISTRICT	NAME	POPULATION	COUNTRY
1	1024 Maharashtra	Mumbai (Bombay)	10500000	IND
2	206 São Paulo	São Paulo	9968485	BRA
3	1890 Shanghai	Shanghai	9696300	CHN
4	3580 Moscow (City)	Moscow	8389200	RUS
5	3793 New York	New York	8008278	USA
6	1891 Peking	Peking	7472000	CHN
7	1025 Delhi	Delhi	7206704	IND
8	1892 Chongqing	Chongqing	6351600	CHN
9	207 Rio de Janeiro	Rio de Janeiro	5598953	BRA
10	1893 Tianjin	Tianjin	5286800	CHN
11	3581 Pietari	St Petersburg	4694000	RUS
12	1026 West Bengali	Calcutta [Kolkata]	4399819	IND
13	1894 Hubei	Wuhan	4344600	CHN
14	1895 Heilongjiang	Harbin	4289800	CHN
15	1896 Liaoning	Shenyang	4265200	CHN
16	1897 Guangdong	Kanton [Guangzhou]	4256300	CHN
17	1027 Tamil Nadu	Chennai (Madras)	3841396	IND
18	3794 California	Los Angeles	3694820	USA
19	1898 Sichuan	Chengdu	3361500	CHN
20	130 New South Wales	Sydney	3276207	AUS
21	69 Distrito Federal	Buenos Aires	2982146	ARG
22	1028 Andhra Pradesh	Hyderabad	2964638	IND
23	3795 Illinois	Chicago	2896016	USA
24	1029 Gujarat	Ahmedabad	2876710	IND
25	1899 Jiangsu	Nanking [Nanjing]	2870300	CHN
26	131 Victoria	Melbourne	2865329	AUS
27	1900 Jilin	Changchun	2812000	CHN
28	1901 Shaanxi	Xi'an	2761400	CHN
29	1902 Liaoning	Dalian	2697000	CHN
30	1030 Karnataka	Bangalore	2660088	IND
31	1903 Shandong	Qingdao	2596000	CHN
32	208 Bahia	Salvador	2302832	BRA
33	1904 Shandong	Jinan	2278100	CHN
34	1905 Zhejiang	Hangzhou	2190500	CHN

Database	Statistics
MYSQL	BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=255, operationsPerSecond=11.76470588235294)
POSTGRES	BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=325, operationsPerSecond=9.23076923076923)
H2	BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=1999, operationsPerSecond=1.5007503751875937)
ORACLE	BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=115, operationsPerSecond=26.08695652173913)

- **Top 3 cities from every region based on population**

```

SELECT town.*, country.region

FROM town

JOIN country

ON town.country = country.code

WHERE (

SELECT COUNT(*)

FROM town twn

JOIN country cntr

ON twn.country = cntr.code

WHERE twn.population >= town.population AND country.region = cntr.region

) <= 3

ORDER BY country.region;

```

ID	DISTRICT	NAME	POPULATION	COUNTRY	REGION
1	132 Queensland	Brisbane	1291117	AUS	Australia and New Zealand
2	130 New South Wales	Sydney	3276207	AUS	Australia and New Zealand
3	131 Victoria	Melbourne	2865329	AUS	Australia and New Zealand
4	2434 Riika	Riga	764328	LVA	Baltic Countries
5	2447 Vilna	Vilnius	577969	LTU	Baltic Countries
6	2448 Kaunas	Kaunas	412639	LTU	Baltic Countries
7	457 England	Birmingham	1013000	GBR	British Islands
8	456 England	London	7285000	GBR	British Islands
9	458 Scotland	Glasgow	619680	GBR	British Islands
10	929 Ouest	Port-au-Prince	884472	HTI	Caribbean
11	587 Distrito Nacional	Santo Domingo de Guzmán	1609966	DOM	Caribbean
12	2413 La Habana	La Habana	2256000	CUB	Caribbean
13	56 Luanda	Luanda	2022000	AGO	Central Africa
14	1803 Littoral	Douala	1448300	CMR	Central Africa
15	2298 Kinshasa	Kinshasa	5064000	COD	Central Africa
16	2517 México	Ecatepec de Morelos	1620303	MEX	Central America
17	2515 Distrito Federal	Ciudad de México	8591309	MEX	Central America
18	2516 Jalisco	Guadalajara	1647720	MEX	Central America
19	756 Addis Abeba	Addis Abeba	2495000	ETH	Eastern Africa
20	1881 Nairobi	Nairobi	2290000	KEN	Eastern Africa
21	3305 Dar es Salaam	Dar es Salaam	1747000	TZA	Eastern Africa
22	1532 Tokyo-to	Tokyo	7980230	JPN	Eastern Asia
23	1890 Shanghai	Shanghai	9696300	CHN	Eastern Asia
24	2331 Seoul	Seoul	9981619	KOR	Eastern Asia
25	3580 Moscow (City)	Moscow	8389200	RUS	Eastern Europe
26	3581 Pietari	St Petersburg	4694000	RUS	Eastern Europe
27	3426 Kiöva	Kyiv	2624000	UKR	Eastern Europe
28	3493 -	Nouméa	76293	NCL	Melanesia
29	2884 National Capital Dis	Port Moresby	247000	PNG	Melanesia
30	764 Central	Suva	77366	FJI	Melanesia
31	2881 Koror	Koror	12000	PLW	Micronesia
32	2688 Chuuk	Weno	22000	FSM	Micronesia
33	2507 Majuro	Dalap-Uliga-Darrit	28000	MHL	Micronesia
34	1365 Baghdad	Baghdad	4336000	IRQ	Middle East
35	3173 Riyadh	Riyadh	3324000	SAU	Middle East
36	3357 Istanbul	Istanbul	8787958	TUR	Middle East
37	3236 Newmaa	Helsinki [Helsingfors]	555474	FIN	Nordic Countries

Database Statistics

MYSQL BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=13134, operationsPerSecond=0.2284148012791229)

POSTGRESQL BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=17164, operationsPerSecond=0.17478443253320902)

H2 BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=18952, operationsPerSecond=0.15829463908822286)

ORACLE BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=18904, operationsPerSecond=0.15869657215404148)

3. Conclusions

Three of the tested databases can be used in production environment, Oracle, MySQL, PostgreSQL. H2 is here only for contrast. As you can see H2 have worst execution times for every query, in first case H2 was about 20 times worst than other databases.

Let's then focus on these three databases mentioned before, in each case different database won, but in general, Oracle had the most stable results, it of course depends on many factors, for example all of these engines works simultaneously on my local machine, which of course is not the best testing environment.

Another case are indexes, in each queries there was some subqueries, this means at least square complexity, indexes on COUNTRY.POPULATION, COUNTRY.SURFACEAREA and TOWN.POPULATION would for sure speed up a bit all the queries, there can be more indexes of course, but I think, these on mentioned numeric values, are most valuable, as I often searched using these values. Additionally, it is difficult to find two same values for mentioned columns, and that's good from index point of view.

Times after indexes was added on mentioned columns:

- **Most popular languages by region (version which works on every database engine)**

Database	Statistics
H2	BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=442, operationsPerSecond=6.787330316742081, exception=null)
ORACLE	BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=50, operationsPerSecond=61.2244897959183672, exception=null)
POSTGRESQL	BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=49, operationsPerSecond=61.224489795918366, exception=null)
MYSQL	BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=80, operationsPerSecond=37.5, exception=null)

- **Most popular languages by region (version which works only on Oracle and PostgreSQL)**

Database	Statistics
H2	BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=3, operationsPerSecond=1.797691148621157E308, exception=java.lang.RuntimeException: org.h2.jdbc.JdbcSQLException: Syntax error in SQL statement "SELECT T.NAME, T.REGION, T.LANGUAGE, POPULATION FROM (SELECT L.NAME, C.REGION, ((L.PERCENTAGE * C.POPULATION) / 100) LANGUAGE_POPULATION, ROW_NUMBER() OVER (PARTITION BY C.REGION ORDER BY (L.PERCENTAGE * C.POPULATION) / 100) DESC) ROW_NUMBER FROM LANGUAGE L JOIN COUNTRY C ON C.CODE = L.COUNTRY) T WHERE T.ROW_NUMBER = 1 ", expected 77; SQL statement: SELECT t.name, t.region, t.language, population FROM (SELECT l.name, c.region, ((l.percentage * c.population) / 100) language_population, row_number() over (partition by c.region order by ((percentage * c.population) / 100) desc) row_number FROM language l join country c on c.code = l.country) t where row_number = 1 (42001-1977)
ORACLE	BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=25, operationsPerSecond=120.0, exception=null)
POSTGRESQL	BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=56, operationsPerSecond=53.57142857142857, exception=null)
MYSQL	BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=3, operationsPerSecond=1.797691148621157E308, exception=java.lang.RuntimeException: com.mysql.jdbc.exceptions.jdbc4.MySQLException: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'row_number FROM language l join country c on c.code = l.country) t where row' at line 3)

- **Cities from 10 biggest countries**

Database	Statistics
H2	BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=150, operationsPerSecond=20.0, exception=null)
ORACLE	BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=65, operationsPerSecond=46.15384615384615, exception=null)
POSTGRESQL	BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=142, operationsPerSecond=21.126760563380284, exception=null)
MYSQL	BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=135, operationsPerSecond=22.22222222222222, exception=null)

- **Top 3 cities from every region based on population**

Database	Statistics
H2	BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=16738, operationsPerSecond=0.1792328832596487, exception=null)
ORACLE	BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=37243, operationsPerSecond=0.08055205004967377, exception=null)
POSTGRESQL	BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=11849, operationsPerSecond=0.2531859228626888, exception=null)
MYSQL	BenchmarkController.BenchmarkStatistics(totalOperations=3, totalOperationsTime=10465, operationsPerSecond=0.2866698518872432, exception=null)

With indexes on tables I had an average 25% time improvement. But of course this have some impact on tables initialization, on every application bootstrap, databases are filled with new data, now it takes more time than before.