Multi-Table Queries in SQL

Multi-Table Queries in SQL

- Can use subqueries provided result columns come from same table.
- If result columns come from more than one table must use a join instead.
- To perform join in the FROM clause, use the keyword JOIN between the tables and specify the condition using the ON clause

Aliases

- Possible to use an alias for a table named in FROM clause.
- Alias is separated from table name with a space.
- Alias can be used to qualify column names when there is ambiguity.

Example: Simple Join

- Find all countries, their region name and area

```
SELECT c.name, r.name, area
FROM countries c
   JOIN regions r
   ON c.region_id = r.region_id;
```

name	name	area
Aruba	Caribbean	193.00
Anguilla	Caribbean	96.00
Netherlands Antilles	Caribbean	800.00
Antigua and Barbuda	Caribbean	442.00
Bahamas	Caribbean	13878.00
Barbados	Caribbean	430.00
Cuba	Caribbean	110861.00
Cayman Islands	Caribbean	264.00
Dominica	Caribbean	751.00
Dominican Republic	Caribbean	48511.00
Guadeloupe	Caribbean	1705.00
Grenada	Caribbean	344.00
Haiti	Caribbean	27750.00
Jamaica	Caribbean	10990.00
Saint Kitts and Nevis	Caribbean	261.00
Saint Lucia	Caribbean	622.00

Join Exercises

List region name and the continent name for all regions.

List the country name and population in 2005. Sort by country name. *Note: this requires a where clause.*

Alternate JOIN syntax

- If the columns that you are joining on have the same name in both tables, you can simplify the statement with `USING`.
- Example:

```
SELECT c.name, r.name, area
FROM countries c  JOIN regions r
   USING (region_id);
```

Example: Three Table Join

- List each country with its region name and continent name

```
SELECT c.name, r.name, d.name
FROM countries c
JOIN regions r ON c.region_id =
r.region_id
JOIN continents d ON d.continent_id =
r.continent_id
ORDER BY c.name;
```

name	name	name
Afghanistan	Southern and Central Asia	Asia
Albania	Southern Europe	Europe
Algeria	Northern Africa	Africa
American Samoa	Polynesia	Oceania
Andorra	Southern Europe	Europe
Angola	Central Africa	Africa
Anguilla	Caribbean	North America
Antarctica	Antarctica	Antarctica
Antigua and Barbuda	Caribbean	North America
Argentina	South America	South America
Armenia	Middle East	Asia
Aruba	Caribbean	North America
Australia	Australia and New Zealand	Oceania
Austria	Western Europe	Europe
Azerbaijan	Middle East	Asia
Bahamas	Caribbean	North America
Bahrain	Middle East	Asia
Bangladesh	Southern and Central Asia	Asia

Example: Three Table Join with USING

- List each country with its region name and continent name

```
SELECT
 countries.name as country,
 regions.name as region,
 continents.name as continent
FROM countries
JOIN regions USING(region_id)
JOIN continents using(continent_id)
ORDER BY countries.name;
```

country	region	continent
Afghanistan	Southern and Central Asia	Asia
Albania	Southern Europe	Europe
Algeria	Northern Africa	Africa
American Samoa	Polynesia	Oceania
Andorra	Southern Europe	Europe
Angola	Central Africa	Africa
Anguilla	Caribbean	North America
Antarctica	Antarctica	Antarctica
Antigua and Barbuda	Caribbean	North America
Argentina	South America	South America
Armenia	Middle East	Asia
Aruba	Caribbean	North America
Australia	Australia and New Zealand	Oceania
Austria	Western Europe	Europe
Azerbaijan	Middle East	Asia
Bahamas	Caribbean	North America
Bahrain	Middle East	Asia
Bangladesh	Southern and Central Asia	Asia
Barbados	Caribbean	North America
Belarus	Eastern Europe	Europe

Computing a Join

- Form a Cartesian product of the tables named in FROM clause.
  - Restrict the results by specifying a predicate using JOIN ON
- If there is a WHERE clause, apply the search condition to each row of the product table, retaining those rows that satisfy the condition.
- For each remaining row, determine value of each item in SELECT list to produce a single row in result table.
- If DISTINCT has been specified, eliminate any duplicate rows from the result table.
- If there is an ORDER BY clause, sort result table as required.

Multiple Table Joins Exercise

List the country name and the name of the official languages for each country. Order the results by country name.

Example: Multiple Table Grouping

- Find number of countries in each region

```
SELECT r.name,
count(c.country_id) as
num_countries
FROM countries  c JOIN
regions r ON c. region_id =
r.region_id
GROUP by r.region_id
ORDER BY r.name;
```

name	num_countries
Antarctica	5
Australia and New Zealand	5
Baltic Countries	3
British Islands	2
Caribbean	24
Central Africa	9
Central America	8
Eastern Africa	20
Eastern Asia	8
Eastern Europe	10
Melanesia	5
Micronesia	7
Micronesia/Caribbean	1
Middle East	18
Nordic Countries	7

Example: Multiple Table Grouping

- Find number of countries in each region

```
SELECT regions.name,
count(country_id) as
num_countries
FROM countries JOIN regions
USING (region_id)
GROUP by regions.region_id
ORDER BY regions.name;
```

name	num_countries
Antarctica	5
Australia and New Zealand	5
Baltic Countries	3
British Islands	2
Caribbean	24
Central Africa	9
Central America	8
Eastern Africa	20
Eastern Asia	8
Eastern Europe	10
Melanesia	5
Micronesia	7
Micronesia/Caribbean	1
Middle East	18
Nordic Countries	7

Outer Joins

Outer Joins

- If one row of a joined table is unmatched, row is omitted from result table.
- Outer join operations retain rows that do not satisfy the join condition.
- Use keywords LEFT or RIGHT to specify the type of outer join
- The OUTER keyword is optional and is usually omitted

Example: Left Outer Join

- Find countries and their GDP

```
SELECT name, year,  gdp
FROM countries LEFT JOIN country_stats
USING (country_id)
ORDER BY name;
```

Haiti	2015	8724656126
Haiti	2016	7975563430
Haiti	2017	8408252995
Haiti	2018	9658084644
Heard Island and McDonald Islands	NULL	NULL
Holy See (Vatican City State)	NULL	NULL
Honduras	1960	335650000
Honduras	1961	356200000
Honduras	1962	387750000
Honduras	1963	410200000
Honduras	1964	457000000
Honduras	1965	508650000
Honduras	1966	549950000
Honduras	1967	598100000

Example: Find rows with no match in second table

- Find countries with no GDP statistics

```
SELECT name, year,  gdp
FROM countries LEFT JOIN country_stats
USING (country_id)
WHERE gpd IS NULL
ORDER BY name;
```

name	year	gdp
Anguilla	NULL	NULL
Antarctica	NULL	NULL
Bouvet Island	NULL	NULL
British Indian Ocean Territory	NULL	NULL
Christmas Island	NULL	NULL
Cocos (Keeling) Islands	NULL	NULL
Cook Islands	NULL	NULL
East Timor	NULL	NULL
Falkland Islands	NULL	NULL
French Guiana	NULL	NULL
French Southern territories	NULL	NULL
Gibraltar	NULL	NULL

Outer Joins

- Order of tables is important
- Left outer join selects all rows from the table listed first in the query
- Right outer join selects all rows from table listed second in the query

Union, Intersect and Difference

Union, Intersect, and Difference (Except)

- Can use normal set operations of Union, Intersection, and Difference to combine results of two or more queries into a single result table.
- Union of two tables, A and B, is table containing all rows in either A or B or both.
- Intersection is table containing all rows common to both A and B.
- Difference is table containing all rows in A but not in B.
- Two tables must be union compatible.

Unions

- Combines the result sets of two or more SELECT statements into a single result set
- Unions eliminate duplicate rows by default
- Column names are taken from the first SELECT clause
- Sorting is done by the last SELECT statement

Example: UNION

- Produces result tables from both queries and merges both tables together.
- List all contact names (employees and customers) assuming two tables

```
(SELECT firstName, lastName
 FROM employees)
UNION
(SELECT contactFirstName,
contactLastName
 FROM customers);
```

firstName	lastName
Diane	Murphy
Mary	Patterson
Jeff	Firrelli
William	Patterson
Gerard	Bondur
Anthony	Bow
Leslie	Jennings
Leslie	Thompson
Julie	Firrelli
Steve	Patterson
Foon Yue	Tseng