All about joins

IMPROVING QUERY PERFORMANCE IN POSTGRESQL

SQL

Amy McCarty
Instructor

Course overview

- Query structure, including joins, subqueries, and temporary tables
- Limiting and aggregating data
- Database storage properties and optimization tools
- Query planning and execution

Query planner

Query

- SQL instructions

Query (execution) plan

- Actual steps

Query planner

What are joins?

- Combine multiple tables

What are joins?

- Combine multiple tables

Why use joins?

- Look up tables
- Combine data

How?

- Inner and outer

Sales ID	Order Dt	Amt	Cust No
01	2019-02-02	145.30	911

ID	Name	Customer Since
911	Jim Smith	2019-01-01

Sales ID	Order Dt	Amt	Name
01	2019-02-02	145.30	Jim Smith

Inner joins

Athlete	Country
Jack	AUT
Aditya	IND
Mikhail	RUS
Javier	MEX

Country	Name	Pop (mil)
AUT	Austria	9
IND	India	1,339
RUS	Russia	145
BRA	Brazil	209

```
SELECT *
FROM athletes a
INNER JOIN countries c
ON a.country = c.country
```

Athlete	Country	Country1	Name	Pop (mil)
Jack	AUT	AUT	Austria	9
Aditya	IND	IND	India	1,339
Mikhail	RUS	RUS	Russia	145

Inner joins

Athlete	Country
Jack	AUT
Aditya	IND
Mikhail	RUS
Javier	**MEX**

Country	Name	Pop (mil)
AUT	Austria	9
IND	India	1,339
RUS	Russia	145
BRA	**Brazil**	**209**

```
SELECT *
FROM athletes a
INNER JOIN countries c
ON a.country = c.country
```

Athlete	Country	Country1	Name	Pop (mil)
Jack	AUT	AUT	Austria	9
Aditya	IND	IND	India	1,339
Mikhail	RUS	RUS	Russia	145

USING inner joins

Athlete	Country
Jack	AUT
Aditya	IND
Mikhail	RUS
Javier	MEX

Country	Name	Pop (mil)
AUT	Austria	9
IND	India	1,339
RUS	Russia	145
BRA	Brazil	209

```
SELECT *
FROM athletes
INNER JOIN countries
USING (country)
```

Athlete	Country	Name	Pop (mil)
Jack	AUT	Austria	9
Aditya	IND	India	1,339
Mikhail	RUS	Russia	145

USING inner joins

Athlete	Country
Jack	AUT
Aditya	IND
Mikhail	RUS
Javier	**MEX**

Country	Name	Pop (mil)
AUT	Austria	9
IND	India	1,339
RUS	Russia	145
BRA	**Brazil**	**209**

```
SELECT *
FROM athletes
INNER JOIN countries
USING (country)
```

Athlete	Country	Name	Pop (mil)
Jack	AUT	Austria	9
Aditya	IND	India	1,339
Mikhail	RUS	Russia	145

Left outer join

Athlete	Country
Jack	AUT
Aditya	IND
Mikhail	RUS
Javier	MEX

Country	Name	Pop (mil)
AUT	Austria	9
IND	India	1,339
RUS	Russia	145
BRA	Brazil	209

```sql
SELECT *
FROM athletes a
LEFT JOIN countries c
ON a.country = c.country
```

Athlete	Country	Country1	Name	Pop (mil)
Jack	AUT	AUT	Austria	9
Aditya	IND	IND	India	1,339
Mikhail	RUS	RUS	Russia	145
Javier	MEX			

Left outer join

Athlete	Country
Jack	AUT
Aditya	IND
Mikhail	RUS
Javier	MEX

Country	Name	Pop (mil)
AUT	Austria	9
IND	India	1,339
RUS	Russia	145
BRA	**Brazil**	**209**

```
SELECT *
FROM athletes a
LEFT JOIN countries c
ON a.country = c.country
```

Athlete	Country	Country1	Name	Pop (mil)
Jack	AUT	AUT	Austria	9
Aditya	IND	IND	India	1,339
Mikhail	RUS	RUS	Russia	145
Javier	MEX			

Right outer join

Athlete	Country
Jack	AUT
Aditya	IND
Mikhail	RUS
Javier	MEX

Country	Name	Pop (mil)
AUT	Austria	9
IND	India	1,339
RUS	Russia	145
BRA	Brazil	209

```
SELECT *
FROM athletes a
RIGHT JOIN countries c
ON a.country = c.country
```

Athlete	Country	Country1	Name	Pop (mil)
Jack	AUT	AUT	Austria	9
Aditya	IND	IND	India	1,339
Mikhail	RUS	RUS	Russia	145
		BRA	Brazil	209

Right outer join

Athlete	Country
Jack	AUT
Aditya	IND
Mikhail	RUS
Javier	**MEX**

Country	Name	Pop (mil)
AUT	Austria	9
IND	India	1,339
RUS	Russia	145
BRA	Brazil	209

```
SELECT *
FROM athletes a
RIGHT JOIN countries c
ON a.country = c.country
```

Athlete	Country	Country1	Name	Pop (mil)
Jack	AUT	AUT	Austria	9
Aditya	IND	IND	India	1,339
Mikhail	RUS	RUS	Russia	145
		BRA	Brazil	209

Full outer join

```
SELECT *
FROM athletes a
FULL OUTER JOIN countries c
ON a.country = c.country
```

Athlete Nme	Country	Country1	Name	Pop (mil)
Jack	AUT	AUT	Austria	9
Aditya	IND	IND	India	1,339
Mikhail	RUS	RUS	Russia	145
Javier	MEX			
		BRA	Brazil	209

- Query (execution) plan

- Constrains query planner

Let's practice!

IMPROVING QUERY PERFORMANCE IN POSTGRESQL

Subqueries and common table expressions (cte)

IMPROVING QUERY PERFORMANCE IN POSTGRESQL

SQL

Amy McCarty
Instructor

About subqueries

What?

- Join alternative
- Simple query

Why?

- Can return one result
- Readable
- SQL instructions similar to joins

How?

- In SELECT, FROM, or WHERE clauses

SELECT subquery

row	script_word	word_length
1	goat	4
2	goat	4
3	dog	3
15,782	...	...

row	english_word	word_length
1	goat	4
2	turkey	6
3	ant	3
171,476	...	...

SELECT subquery

```
SELECT AVG(word_length) AS avg_movie
  , (SELECT AVG(word_length)
      FROM english_language)
      AS avg_english
FROM MOVIE
```

avg_movie	avg_english
3	4.5

WHERE subquery

row	script_word	word_length
1	goat	4
2	goat	4
3	dog	3
15,782	...	...

row	english_word	word_length
1	goat	4
2	turkey	6
3	ant	3
171,476	...	...

WHERE subquery

```
SELECT AVG(word_length) AS avg_movie
FROM english_language
WHERE word IN
  (SELECT DISTINCT word FROM movie)
```

avg_movie
3

FROM subquery

```
SELECT AVG(word_length) AS avg_movie
FROM (SELECT * FROM movie)
```

- Decreases readability
- Limits query plan flexibility

About common table expressions (CTEs)

What?

- Join alternative
- Standalone query with temporary results set

Why?

- Can return one result
- Readable
- Creates a temporary table

How?

- WITH statements

CTE structure

```
WITH english_cte AS
(
  SELECT word_length
        , COUNT(word) AS word_count AS english_word_count
    FROM english_language
)
SELECT movie.word_length
   , COUNT(movie.word) AS movie_word_count
   , cte.english_word_count
FROM movie
INNER JOIN english_cte cte
ON movie.word_length = cte.word_length
GROUP BY movie.word_length, cte.english_word_count
```

Let's practice!

IMPROVING QUERY PERFORMANCE IN POSTGRESQL

Working with temporary tables

IMPROVING QUERY PERFORMANCE IN POSTGRESQL

SQL

Amy McCarty
Instructor

datacamp

About temp(orary) tables

What?

- Short-lived table

Why?

- Transient storage
- Database session
- Multiple queries
- User specific
- Slow tables

How?

- CREATE TEMP TABLE name AS

TEMP table structure

holiday	holiday_type	country_code
Epiphany	religious	CZE
Epiphany	religious	FRA
Epiphany	religious	USA
Thanksgiving	secular	USA

```
CREATE TEMP TABLE usa_holidays AS
  SELECT holiday, holiday_type
  FROM world_holidays
  WHERE country_code = 'USA';
```

USA Holidays

holiday	holiday_type
Epiphany	religious
Thanksgiving	secular

Slow, large tables

- Slow because many records

Table Stats	World Holidays	USA Holidays
Type	table	temp table
# Rows	591,444	25

Slow, complicated views

- Slow because view logic

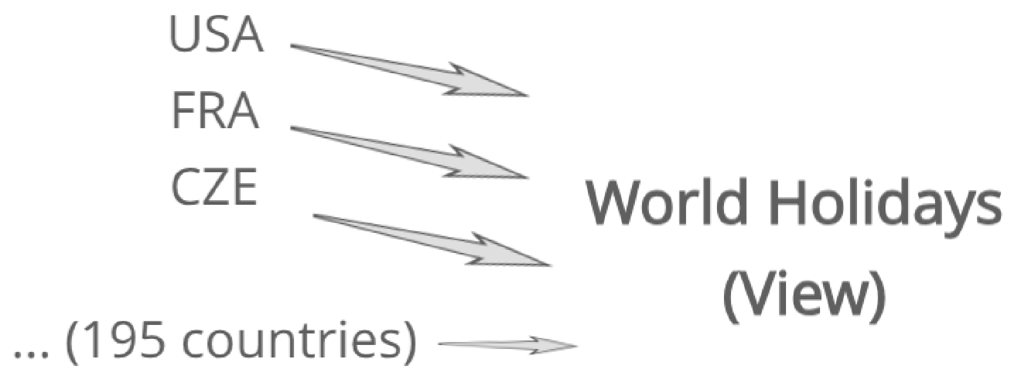

Table Stats	World Holidays	USA Holidays
Type	view	temp_table
# Rows	591,444	25
Sources	195	1

- Tables contain data
- Views contain the directions to data

Joining many tables to one

```
CREATE TEMP TABLE usa_holidays AS
  SELECT holiday, holiday_type
  FROM world_holidays
  WHERE country_code = 'USA';
```

```
WITH religious AS
(   SELECT usa.holiday, r.initial_yr
     , r.celebration_dt
    FROM religious r
    INNER JOIN usa_holidays usa
      USING (holiday)   )
, secular AS
(   SELECT usa.holiday, s.initial_yr
     , s.celebration_dt
    FROM secular s
    INNER JOIN usa_holidays usa
      USING (holiday)   )
, ...
```

ANALYZE

```sql
CREATE TEMP TABLE usa_holidays AS
SELECT holiday, holiday_type
FROM world_holidays
WHERE country_code = 'USA';

ANALYZE usa_holidays;

SELECT * FROM usa_holidays
```

Query planner (execution steps)

- Statistics from pg_statistics
- Runtime estimates

Let's practice!

IMPROVING QUERY PERFORMANCE IN POSTGRESQL