SQL Survival Guide
Joins

Andy Choens

EBCoP

May 18, 2015

Outline

Outline for section 1

Example Data

- Two tables of example data!
- ALL Code provided: sql/04-joins
  - create_tables.sql: Creates the example tables.
  - Everything Else: Other files contain example queries.
- Example data in SQLite: data/04-joins.sqlite.

Table: Departments

DEPT_ID	DEPT_NAME	DEPT_FLOOR
31	Sales	1
33	Engineering	3
34	Clerical	2
35	Marketing	3

Aliased as 'dept'.

Table: Employees

EID	DEPT_ID	LAST_NAME	FIRST_NAME	GENDER
1	31	Rafferty	Gerry	M
3	33	Jones	Jon	M
5	33	Heisenberg	Werner	M
7	34	Robinson	Elizabeth	F
9	34	Smith	Jefferson	M
11	NULL	Williams	Serena	F

Aliased as 'empl'.

Outline for section 2

Dafynitions?

We need a common set of terms to discuss this.
TODO: The slides in this section need a lot of work.

Dafynition: Relational Data

TODO

Dafynition: Normalized Data

TODO

Dafynition: Result Set

TODO

Dafynition: Clause

The following protected SQL commands START a clause:

- SELECT
- FROM
- WHERE
- HAVING

Dafynition: Clause

The following protected SQL commands START a clause:

- SELECT
- FROM
- WHERE
- HAVING

Style:

- Align SQL Clauses
- Predicates should be indented consistently.
- Feel free to ask me why I structure SQL queries the way I do.

Dafynition: Join Clause?

- The FIRST thing you should write.
- Combines records from two or more tables (result set).
- SQL Joins are a difficult skill to master.
- They are necessary for working with normalized, relational data.

Outline for section 3

Order of Operations

What is the answer to this (silly) equation?

$$\sqrt{(2^2 + 2) \cdot 6}$$

Order of Operations

What is the answer to this (silly) equation?

$$\sqrt{(2^2 + 2) \cdot 6}$$

And how were we all able to come up with the same answer?

Order of Operations

Just like math, SQL has an order of operations:

1. FROM
2. WHERE
3. GROUP BY
4. SELECT
5. HAVING
6. ORDER BY

Order of Operations

Some last order of operation notes:

- Sub-queries are run before the outer query.
- The optimizer may reorganize query (relational algebra).
- Sometimes writing things out of order can bite you.

Outline for section 4

Types of Joins

There are **only** 5 ANSI joins:

1. Cross Join
2. Inner Join
3. Left Outer Join
4. Right Outer Join
5. Full Outer Join

1. DO NOT USE!
2. Very Useful.
3. Very Useful.
4. Most of you should not use this.
5. You will rarely use this.

Most SQL queries can be written with only 2 types of joins.

Cross Join: Discussion

- Returns the Cartesian product of the tables in the FROM clause.
- Can be written explicitly and implicitly.
- This is **ALMOST ALWAYS** a bad idea.

Explicit Cross Join: Example

Question:
How many rows will the following query return?

```
1    select *
2    from EMPLOYEES empl cross join DEPARTMENTS dept
3    ;
```

Source: https://github.com/Choens/sql-survival-guide/blob/master/sql/04-joins/cross-joins.sql

Explicit Cross Join: Example

Question:
How many rows will the following query return?

```
1    select *
2    from EMPLOYEES empl cross join DEPARTMENTS dept
3    ;
```

Source: https://github.com/Choens/sql-survival-guide/blob/master/sql/04-joins/cross-joins.sql

Answer:
24 rows

Cross Join: What Happens

TODO: A graphical representation of what it is doing.

Cross Join: Result Set

EID	DEPT_ID	LAST_NAME	FIRST_NAME	GENDER	DEPT_ID	DEPT_NAME	DEPT_FLOOR
1	31	Rafferty	Gerry	M	31	Sales	1
1	31	Rafferty	Gerry	M	33	Engineering	3
1	31	Rafferty	Gerry	M	34	Clerical	2
1	31	Rafferty	Gerry	M	35	Marketing	3
3	33	Jones	Jon	M	31	Sales	1
3	33	Jones	Jon	M	33	Engineering	3
3	33	Jones	Jon	M	34	Clerical	2
3	33	Jones	Jon	M	35	Marketing	3
5	33	Heisenberg	Werner	M	31	Sales	1
5	33	Heisenberg	Werner	M	33	Engineering	3
5	33	Heisenberg	Werner	M	34	Clerical	2
5	33	Heisenberg	Werner	M	35	Marketing	3
7	34	Robinson	Elizabeth	F	31	Sales	1
7	34	Robinson	Elizabeth	F	33	Engineering	3
7	34	Robinson	Elizabeth	F	34	Clerical	2
7	34	Robinson	Elizabeth	F	35	Marketing	3
9	34	Smith	Jefferson	M	31	Sales	1
9	34	Smith	Jefferson	M	33	Engineering	3
9	34	Smith	Jefferson	M	34	Clerical	2
9	34	Smith	Jefferson	M	35	Marketing	3
11	[NULL]	Williams	Serena	F	31	Sales	1
11	[NULL]	Williams	Serena	F	33	Engineering	3
11	[NULL]	Williams	Serena	F	34	Clerical	2
11	[NULL]	Williams	Serena	F	35	Marketing	3

Implicit Cross Join: Example

```
1    select *
2    from EMPLOYEES empl, DEPARTMENTS dept
3    ;
```

Source: https://github.com/Choens/sql-survival-guide/blob/master/sql/04-joins/cross-joins.sql

- Also returns 24 rows.
- What does this query look like?

Cross Join: Results

- Cross Joins Are Dangerous! (Especially the implicit ones.)
- They return the maximum number of rows possible.

Inner Join: Discussion

- Returns all records which have matching records in both tables, according to the join-predicate or WHERE clause.
- Can be written explicitly or implicitly.
- Returns the least number of rows.

Explicit Inner Join: Example

Question:
How many rows will the following query return?

```
1    select *
2    from EMPLOYEES empl inner join DEPARTMENTS dept
3    on empl.dept_id = dept.dept_id
4    ;
```

Source: https://github.com/Choens/sql-survival-guide/blob/master/sql/04-joins/cross-joins.sql

Explicit Inner Join: Example

Question:
How many rows will the following query return?

```
1    select *
2    from EMPLOYEES empl inner join DEPARTMENTS dept
3    on empl.dept_id = dept.dept_id
4    ;
```

Source: https://github.com/Choens/sql-survival-guide/blob/master/sql/04-joins/cross-joins.sql

Answer:
5 rows

Inner Join: What Happens

TODO: A pictoral representation.

Inner Join: Result Set

EID	DEPT_ID	LAST_NAME	FIRST_NAME	GENDER	DEPT_ID	DEPT_NAME	DEPT_FLOOR
1	31	Rafferty	Gerry	M	31	Sales	1
3	33	Jones	Jon	M	33	Engineering	3
5	33	Heisenberg	Werner	M	33	Engineering	3
7	34	Robinson	Elizabeth	F	34	Clerical	2
9	34	Smith	Jefferson	M	34	Clerical	2

Implicit Inner Join: Example

This should look familiar:

```
1    select *
2    from EMPLOYEES empl, DEPARTMENTS dept
3    where empl.dept_id = dept.dept_id
4    ;
```

Source: https://github.com/Choens/sql-survival-guide/blob/master/sql/04-joins/cross-joins.sql

- Also returns 5 rows.

Why Not Use Implicit Join Syntax?

Question:
Implicit Cross Join v Implicit INNER JOIN: What's the difference?

Why Not Use Implicit Join Syntax?

Question:
Implicit Cross Join v Implicit INNER JOIN: What's the difference?

Answer:
The WHERE clause.

- Implicit Join syntax *is* deprecated.
- This style makes it too easy to write a Cross Join (cartesian).
- It makes it harder to learn the SQL Order of Operations.

Outer Joins

- The result set from an Inner Joins includes matching records ONLY.
- The result set from an Outer Join retains more records.
- Types of Outer Joins:
  - ▸ Left Outer Join (Left Join)
  - ▸ Right Outer Join (Right Join)
  - ▸ Full Outer Join
- The only difference is which records become part of the result set.

Left Outer Join: Discussion

- Result set includes all members of the 'left' table.

Left Outer Join: Example

Question:
How many rows will the following query return?

```sql
select *
from EMPLOYEES empl left join DEPARTMENTS dept
on empl.department_id = dept_dept_id
;
```

Source: https://github.com/Choens/sql-survival-guide/blob/master/sql/04-joins/left-joins.sql

Left Outer Join: Example

Question:

How many rows will the following query return?

```
1    select *
2    from EMPLOYEES empl left join DEPARTMENTS dept
3    on empl.department_id = dept_dept_id
4    ;
```

Source: https://github.com/Choens/sql-survival-guide/blob/master/sql/04-joins/left-joins.sql

Answer:

6 rows

Right Outer Join: Discussion

TODO

Right Outer Join: Example

TODO

Full Outer Join: Discussion

TODO

Full Outer Join: Example

TODO

Outline for section 5

Self Join: Discussion

TODO

Self Join: Example

TODO

Outline for section 6

Additional Information

- https://en.wikipedia.org/wiki/Join_(SQL)

Questions?