

COMPSCI 105: Lecture #22 Join Relationships

©2015-2019 Dr. William T. Verts

How did we get here?

- Searching and Sorting
 - Looking stuff up
 - Arranging data to make looking stuff up go faster
- Indexing
 - Speeding up searches without actually sorting
- Joins
 - Synthesizing new tables from several old ones.
- Now, **Relationships** (ties everything together!)

©2019 Dr. William T. Verts

Types of Relationships

- Many:Many
 - Neither table indexed on join field
- 1:Many or Many:1
 - One table indexed on join field
- 1:1
 - Both tables indexed on join field

©2015-2017 Dr. William T. Verts

Many:Many Relationships

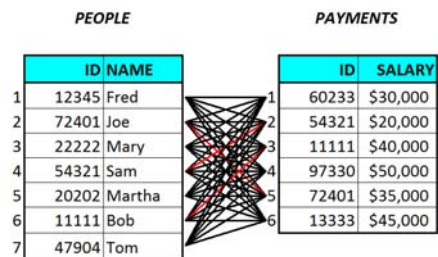
©2019 Dr. William T. Verts

Many:Many Relationships

- Neither table has an index on the join-field.
- Any record in either table may have multiple matches in the other table.
- Every record in one table is compared against all records in the other table.
- For M records in one table and N records in the other, there will be $O(M \times N)$ comparisons.
- It is possible to have $M \times N$ records in the answer (all match), or none (no match).

©2015-2017 Dr. William T. Verts

Many:Many Join

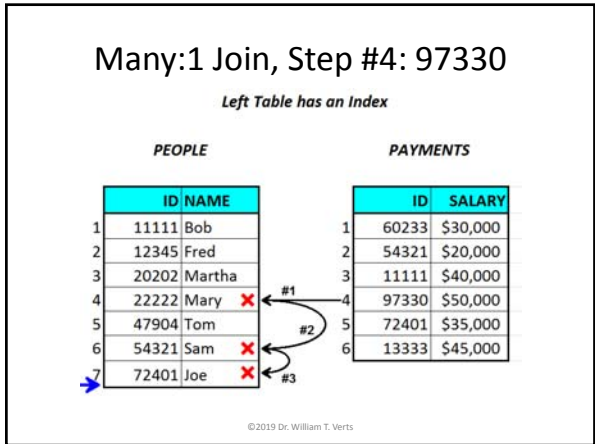
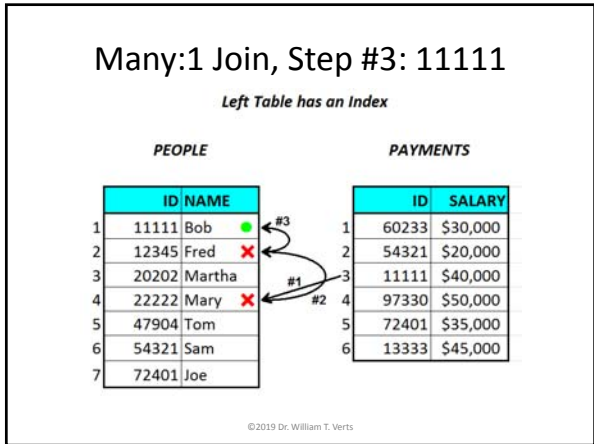
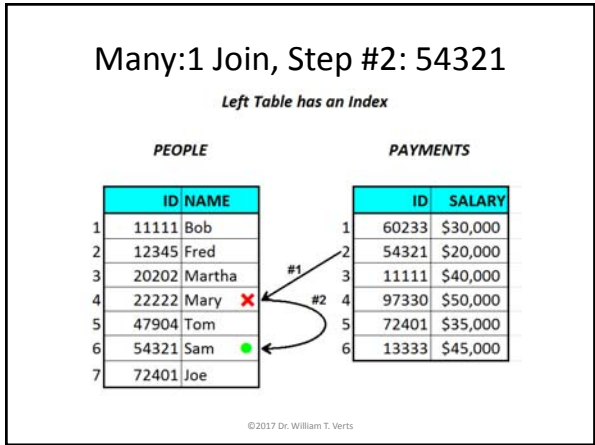
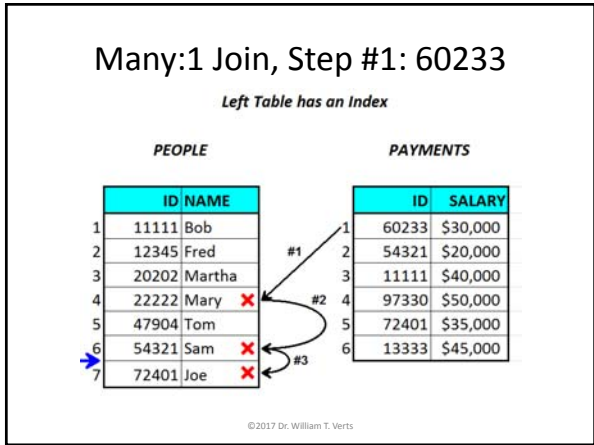


©2014 Dr. William T. Verts

1:Many or Many:1 Relationships

©2019 Dr. William T. Verts

- ### 1:Many or Many:1 Relationships
- One table (M records) is without an index, the other (N records) is indexed on the join-field.
 - The indexed table typically has unique information in the indexed field.
 - Every record in the unindexed table does a binary search into the indexed table.
 - A record in the unindexed table has at most one match in the indexed table.
 - A record in the indexed table may have multiple matches in the unindexed table.
 - There are $O(M \times \log_2(N))$ comparisons.
- ©2015 Dr. William T. Verts



Many:1 Join, Step #5: 72401

Left Table has an Index

PEOPLE		PAYMENTS	
ID	NAME	ID	SALARY
1	11111	Bob	
2	12345	Fred	
3	20202	Martha	
4	22222	Mary	✗
5	47904	Tom	
6	54321	Sam	✗
7	72401	Joe	●

ID	SALARY	
1	60233	\$30,000
2	54321	\$20,000
3	11111	\$40,000
4	97330	\$50,000
5	72401	\$35,000
6	13333	\$45,000

©2019 Dr. William T. Verts

Many:1 Join, Step #6: 13333

Left Table has an Index

PEOPLE		PAYMENTS	
ID	NAME	ID	SALARY
1	11111	Bob	
2	12345	Fred	✗
3	20202	Martha	✗
4	22222	Mary	✗
5	47904	Tom	
6	54321	Sam	
7	72401	Joe	

ID	SALARY	
1	60233	\$30,000
2	54321	\$20,000
3	11111	\$40,000
4	97330	\$50,000
5	72401	\$35,000
6	13333	\$45,000

©2017 Dr. William T. Verts

1:1 Relationships

©2019 Dr. William T. Verts

1:1 Relationships

- Both tables have an index on the join-field.
- Joining the tables requires only one pass each over the two tables.
- A record in either table has at most one match in the other table.
- There are at most $O(M+N)$ comparisons.

©2015 Dr. William T. Verts

1:1 Join

Both Tables have Indexes

PEOPLE		PAYMENTS	
ID	NAME	ID	SALARY
1	11111	Bob	
2	12345	Fred	
3	20202	Martha	
4	22222	Mary	
5	47904	Tom	
6	54321	Sam	
7	72401	Joe	

ID	SALARY	
1	11111	\$40,000
2	13333	\$45,000
3	54321	\$20,000
4	60233	\$30,000
5	72401	\$35,000
6	97330	\$50,000

©2014 Dr. William T. Verts

Examples and Conclusions

©2019 Dr. William T. Verts

Suppose $M = N = 1024$ Records

- Many:Many
 - $M \times N = 1,048,576$ comparisons
 - Possibly 1,048,576 records in answer
- 1:Many or Many:1
 - $M \times \log_2(N) = 1024 \times 10 = 10,240$ comparisons
- 1:1
 - $M + N = 2048$ comparisons (maximum, often smaller)

©2015 Dr. William T. Verts

Conclusions

- Indexing speeds up both queries and joins.
- In a join, having one index is faster than having none, having two indexes is even faster.
- For many:many relationships, neither table will have an index so joins will be slow (oh, well), and may generate many more records than are in either source table. This is OK.

©2015 Dr. William T. Verts

©2019 Dr. William T. Verts