SIDENOTE: RELATIONAL ALGEBRA IS NOT A UNIVERSAL QUERY LANGUAGE

• It is impossible in relational algebra to compute the relation R(ANCESTOR, DESCENDANT)

• The proof is by induction. It is reasonably simple, but cumbersome.

• General idea: Any relational algebra query involves a fixed number of copies of relations

• But this limits the distance between ancestors and descendants that can be searched by the query
Natural Join

- For reasons, which we will see later, it is useful to introduce the concept of natural join.
- Suppose we have relations $R(A,B,C)$ and $S(A,C,D)$.
- Then their natural join is: $\pi_{R.A,R.B,R.C,S.D}(\sigma_{R.A=S.A & R.C=S.C}(R \times S))$.
- In effect, we glue the tuples together by equating columns with the same name, and then remove duplicate columns.
- Natural join appears naturally when information about an object is combined from several tables.
Optimization Notes

- Projections and all kinds of selections are cheaper than cartesian products or joins
- In general, equality selections are cheaper than nonequality selections
- Equality selections on an index are cheaper than projections
Relational Algebra is Pragmatic

- It provides a language for writing useful queries
- These queries can be executed by a machine
- It can help us to optimize the performance of queries
Query Optimization: Pushing Selections Inward

- Tables: INFO(Person,Sex,Age), BIRTH(Parent,Child)
- Problem: List all parents who are 30 years old
- The naive query:
  \[ \pi_{\text{parent}}(\sigma_{\text{person}=\text{parent} \& \text{age}=30}(\text{INFO} \times \text{BIRTH})) \]
- The optimized query:
  \[ \pi_{\text{parent}}(\sigma_{\text{person}=\text{parent}}(\sigma_{\text{age}=30}(\text{INFO} \times \text{BIRTH}))) \]
Proving that the Optimization is Correct

• Why do these queries produce the same relation?
  – \( \sigma_{\text{person}=\text{parent} \& \text{age}=30}(\text{INFO} \times \text{BIRTH}) \)
  – \( \sigma_{\text{person}=\text{parent}}(\sigma_{\text{age}=30}(\text{INFO}) \times \text{BIRTH}) \)

• The proof uses the following identities:
  – \( \sigma_{f \& g}(R) = \sigma_f(\sigma_g(R)) \)
  – \( \sigma_g(R \times S) = \sigma_g(R) \times S \), assuming \( g \) refers only to \( R \)

• Conclusion:
  – \( \sigma_{f \& g}(R \times S) = \sigma_f(\sigma_g(R) \times S) \), assuming \( g \) refers only to \( R \)
Topics Covered in this Unit

• Five basic relational algebra operators
• Use of these operators to solve queries
• Using the algebra to optimize queries