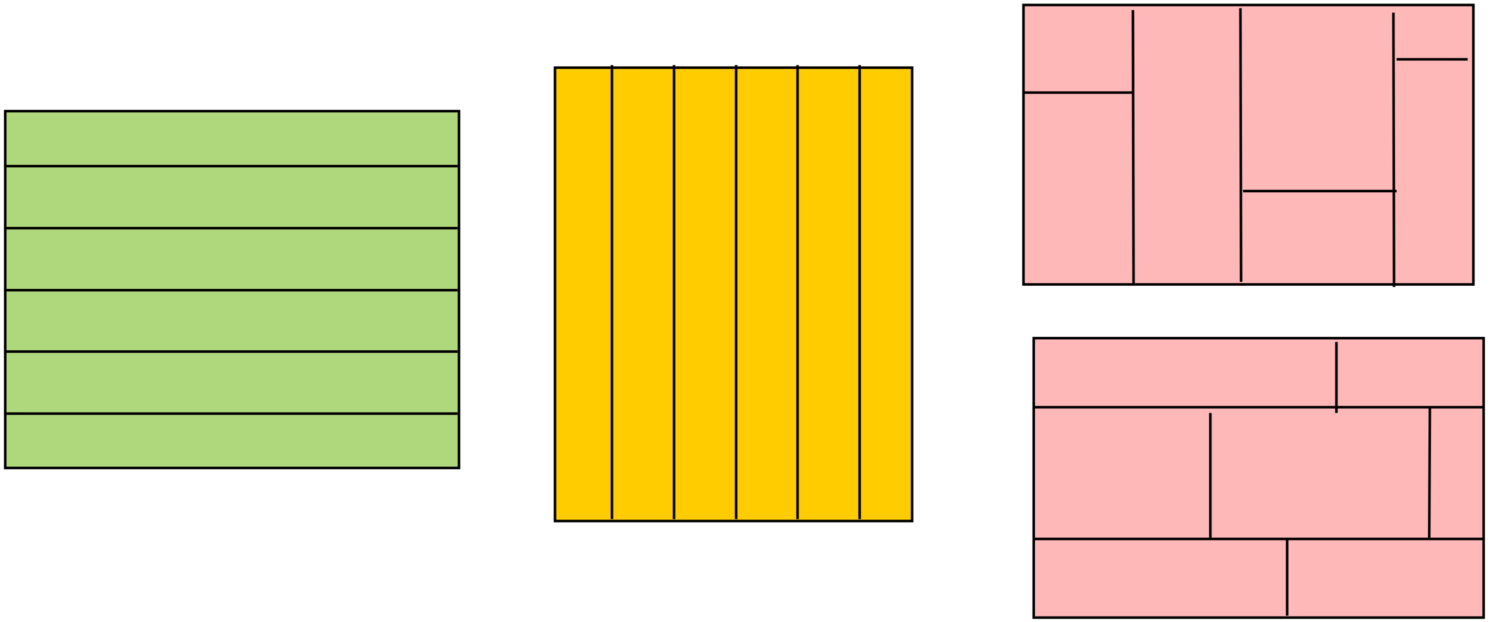
Tutorial 1:
Distributed Databases Design

THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

Alex Polak
E-mail: a.polak@uq.net.au

+ Fragmentation



Horizontal

Vertical

Hybrid/mixed

+ Fragmentation

- Why do we need fragmentation in distributed database?
  - Fully replicate data will cause:
    - Storage issue
    - Query Performance
      - Query time, Index size, Transaction, Lock,...
  - Benefits
    - Parallel process
    - Scalability

+ Fragmentations

- Horizontal Fragmentation
  - Primary Horizontal Fragmentation
    - Using the minterm predicates of itself
  - Derived Horizontal Fragmentation
    - Using the predicates from other relations
- Vertical Fragmentation
  - Each fragmentation contains a subset of attributes

+ Predicate

- A Boolean-Valued Function
  - $P: X \rightarrow \{true, false\}$
    - A predicate on value X
- Simple Predicate
  - A relation $R(A_1, A_2, \ldots, A_N)$
    - A_i is an attribute over domain D_i
  - Predicate P_j defined on R
    - $P_j: A_i\ \theta\ \ Value$
      - $\theta\ \ \in \{=, <, \neq, \leq, >, \geq\}$
      - $Value\ \ \in D_i$

- $P: SAL \geq 30{,}000$
  - $P(5000) \rightarrow false$
  - $P(50000) \rightarrow true$

+ Minterm Predicate

- Minterm:
  - Conjunction of a set of Boolean values
    - $a \wedge b \wedge c$
    - $a \wedge \neg b \wedge c$
    - $a, b, c \in \{0,1\}$
- A set of simple predicate $P = \{P_1, P_2, \ldots, P_n\}$
  - $M_i = P_1^* \wedge P_2^* \wedge \ \ldots \wedge P_n^*$
    - P_j^* is either P_j or $\neg P_j$
      - $P_j: SAL \geq 30{,}000$
      - $\neg P_j: SAL < 30{,}000$
    - 2^n in total

+ Minterm Predicate

- Each minterm predicate defines a **fragmentation**
- Each attribute can have multiple predicates
  - $P_1 : SAL \geq 30{,}000$
  - $P_2 : SAL < 50{,}000$
- Some minterms are useful
  - $m_1 = P_1 \wedge P_2 =$ 30000 $\leq SAL < 50000$
- Some are redundant
  - $m_2 = P_1 \wedge \neg P_2 = SAL \geq 30{,}000 \wedge SAL \geq 50{,}000 \; = \; SAL \geq 50{,}000 = \neg P_2$
  - $m_3 = \neg P_1 \wedge P_2 \; = SAL < 30{,}000 \wedge SAL < 50{,}000 = SAL < 30{,}000 = \neg P_1$
- Some are invalid
  - $m_4 = \neg P_1 \wedge \neg P_2 = SAL < 30{,}000 \wedge SAL \geq 50{,}000 = \emptyset$

+ Correctness

- $R \Rightarrow F = \{F_1, F_2, \dots, F_n\}$
- Completeness
  - $\forall t \in R, \quad \exists F_i \in F, such\ that\ t \in F_i$
- Disjointness
  - $\forall F_i, F_j \in F\ and\ i \neq j \ \Rightarrow F_i \cap F_j = \emptyset$
- Reconstruction
  - $R = \cup_{i=1,\dots,n} F_i$

+ Question 1

- Given the following relation and the predicates $P_1: SAL > 30{,}000$, $P_2: SAL < 30{,}000$

ID	NAME	AGE	SAL
1289	John	24	12000
8907	Sally	29	67050
7643	Elvin	22	51980
0988	Kelly	42	30000
6543	Emily	19	28760
0986	Sally	46	54000
2345	Thomas	23	29999

- How many fragments will we have? (Using simple predicates)

+ Q1: Horizontal Fragmentation

- a) Perform a horizontal fragmentation of the table based on the given predicates

ID	NAME	AGE	SAL
1289	John	24	12000
8907	Sally	29	67050
7643	Elvin	22	51980
0988	Kelly	42	30000
6543	Emily	19	28760
0986	Sally	46	54000
2345	Thomas	23	29999

12000 < 30000

Fragment 1 (SAL < 30000)

ID	NAME	AGE	SAL
1289	John	24	12000
6543	Emily	19	28760
2345	Thomas	23	29999

+ Q1: Horizontal Fragmentation

- b) Is this fragmentation correct? Why?

ID	NAME	AGE	SAL
1289	John	24	12000
8907	Sally	29	67050
7643	Elvin	22	51980
0988	Kelly	42	30000
6543	Emily	19	28760
0986	Sally	46	54000
2345	Thomas	23	29999

Where is Kelly?

Fragment 1 (SAL < 30000)

ID	NAME	AGE	SAL
1289	John	24	12000
6543	Emily	19	28760
2345	Thomas	23	29999

Fragment 2 (SAL > 30000)

ID	NAME	AGE	SAL
8907	Sally	29	67050
7643	Elvin	22	51980
0986	Sally	46	54000

Q1: Horizontal Fragmentation

- Fragmentation properties
  - Completeness ✗
    - $(0988, Kelly, 42, 30000) \notin F_1$ and $(0988, Kelly, 42, 30000) \notin F_2$
  - Disjointness ✓
    - $F_1 \cap F_2 = SAL > 30000 \cap SAL < 30000 = \emptyset$
  - Reconstructability ✗
    - $F_1 \cup F_2 = R - (Kelly)$

- Solution
  - $P_1: SAL \geq 30{,}000$
  - $P_2: SAL < 30{,}000$

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

+ Q1: Horizontal Fragmentation

- c) Generate a correct horizontal fragmentation using minterm predicates

- Minterm predicates
  - Automatically generate predicates that satisfy the properties
  - Input: A set of simple predicates $P = \{P_1, P_2, P_3, \dots, P_n\}$
    - $P_i = a_i\ \theta$ Value
      - $P_1 : SAL > 30{,}000$ $P_2 : SAL < 30{,}000$
  - Enumerate all possible minterm predicates M_i
    - $M_i = P_1^* \wedge P_2^* \wedge \cdots \wedge P_n^* (P_i^* = P_i\ or\ \neg P_i)$
      - $P_1 : SAL > 30{,}000, P_2 : SAL < 30{,}000$
      - $\neg P_1 : SAL \leq 30{,}000, \neg P_2 : SAL \geq 30{,}000$
  - Eliminate useless ones

Q1: Horizontal Fragmentation

- Predicates
  - $P_1: SAL > 30{,}000, P_2: SAL < 30{,}000$
  - $\neg P_1: SAL \leq 30{,}000, \neg P_2: SAL \geq 30{,}000$
- Minterm predicates
  - $M_1: SAL > 30{,}000 \ \wedge \ SAL < 30{,}000$ ✖
  - $M_2: SAL > 30{,}000 \ \wedge \ SAL \geq 30{,}000$
  - $M_3: SAL \leq 30{,}000 \ \wedge \ SAL < 30{,}000$
  - $M_4: SAL \leq 30{,}000 \ \wedge \ SAL \geq 30{,}000$
- Elimination
  - $M_2: SAL > 30{,}000$, $M_3: SAL < 30{,}000$
  - $M_4: SAL = 30{,}000$

+ Q1: Vertical Fragmentation

- d) Vertically fragment this relation to S1(ID, NAME, AGE) and S2(NAME, SALARY)
  - Is this fragmentation correct?

ID	NAME	AGE	SAL
1289	John	24	12000
8907	Sally	29	67050
7643	Elvin	22	51980
0988	Kelly	42	30000
6543	Emily	19	28760
0986	Sally	46	54000
2345	Thomas	23	29999

Original

ID	NAME	AGE
1289	John	24
8907	Sally	29
7643	Elvin	22
0988	Kelly	42
6543	Emily	19
0986	Sally	46
2345	Thomas	23

Fragment 1

NAME	SAL
John	12000
Sally	67050
Elvin	51980
Kelly	30000
Emily	28760
Sally	54000
Thomas	29999

Fragment 2

+ Q1: Vertical Fragmentation

- Reconstructability?
  - Join: Original = Fragment 1 ⋈ Fragment 2

Fragment 1

ID	NAME	AGE
1289	John	24
8907	Sally	29
7643	Elvin	22
0988	Kelly	42
6543	Emily	19
0986	Sally	46
2345	Thomas	23

Fragment 2

NAME	SAL
John	12000
Sally	67050
Elvin	51980
Kelly	30000
Emily	28760
Sally	54000
Thomas	29999

Original

ID	NAME	AGE	SAL
1289	John	24	12000
8907	Sally	29	67050
7643	Elvin	22	51980
0988	Kelly	42	30000
6543	Emily	19	28760
0986	Sally	46	54000
2345	Thomas	23	29999
8907	Sally	29	54000
0986	Sally	46	67050

+ Q1: Vertical Fragmentation

- Reconstructability? *solution*
  - Join: Original = Fragment 1 ⋈ Fragment 2

Fragment 1

ID	NAME	AGE
1289	John	24
8907	Sally	29
7643	Elvin	22
0988	Kelly	42
6543	Emily	19
0986	Sally	46
2345	Thomas	23

Fragment 2

ID	SAL
1289	12000
8907	67050
7643	51980
0988	30000
6543	28760
0986	54000
2345	29999

+ Q1: Derived Horizontal Fragmentation

- e) S_ID is a foreign key to Student.ID, perform **derived horizontal fragmentation** using semi-join

ID	NAME	AGE	SAL
1289	John	24	12000
8907	Sally	29	67050
7643	Elvin	22	51980
0988	Kelly	42	30000
6543	Emily	19	28760
0986	Sally	46	54000
2345	Thomas	23	29999

S_ID	COURSE	RESULT
1289	INFS1200	7
1289	INFS2200	6
8907	DECO1400	5
8907	INFS1200	4
8907	INFS2200	4
7643	COMP1002	6
0988	COMP4500	6
0988	INFS2200	5
6543	INFS1200	4
0986	INFS1200	7
2345	INFS1200	7

+ Q1: Derived Horizontal Fragmentation

- Semi-join: $R \ltimes S = \pi_R R \bowtie S$
  - Only attributes in R will appear in the final result
  - S is used to filter the record in R

ID	NAME	AGE	SAL
1289	John	24	12000
0988	Kelly	42	30000
6543	Emily	19	28760
2345	Thomas	23	29999

S

S_ID	COURSE	RESULT
1289	INFS1200	7
1289	INFS2200	6
8907	DECO1400	5
8907	INFS1200	4
8907	INFS2200	4
7643	COMP1002	6
0988	COMP4500	6
0988	INFS2200	5
6543	INFS1200	4
0986	INFS1200	7
2345	INFS1200	7

R

+ Q1: Derived Horizontal Fragmentation

- Derived fragmentation
  - The first table is already fragmented
  - We want the second table to be fragmented the same way as the first one
  - Use semi-join to do the fragmentation
  - S is called owner, R is called member

ID	NAME	AGE	SAL
1289	John	24	12000
0988	Kelly	42	30000
6543	Emily	19	28760
2345	Thomas	23	29999

ID	NAME	AGE	SAL
8907	Sally	29	67050
7643	Elvin	22	51980
0986	Sally	46	54000

S

S_ID	COURSE	RESULT
1289	INFS1200	7
1289	INFS2200	6
8907	DECO1400	5
8907	INFS1200	4
8907	INFS2200	4
7643	COMP1002	6
0988	COMP4500	6
0988	INFS2200	5
6543	INFS1200	4
0986	INFS1200	7
2345	INFS1200	7

R

+ Q1: Derived Horizontal Fragmentation

- Derived fragmentation
  - If the fragmentation is not based on foreign key?

ID	NAME	AGE	SAL
1289	John	24	12000
0988	Kelly	42	30000
6543	Emily	19	28760
2345	Thomas	23	29999

ID	NAME	AGE	SAL
8907	Sally	29	67050
1384	Kelly	25	51980
0986	Sally	46	54000

S

Name	COURSE	RESULT
John	INFS1200	7
Kelly	INFS2200	6
Emily	DECO1400	5
Sally	INFS1200	4
Thomas	INFS2200	4
Emily	COMP1002	6
John	COMP4500	6

R

+ Q1: Derived Horizontal Fragmentation

- Derived fragmentation
  - If the fragmentation is not based on foreign key?

ID	NAME	AGE	SAL
1289	John	24	12000
0988	Kelly	42	30000
6543	Emily	19	28760
2345	Thomas	23	29999

ID	NAME	AGE	SAL
8907	Sally	29	67050
1384	Kelly	25	51980
0986	Sally	46	54000

S

Name	COURSE	RESULT
John	INFS1200	7
Kelly	INFS2200	6
Emily	DECO1400	5
Thomas	INFS2200	4
Emily	COMP1002	6
John	COMP4500	6

Name	COURSE	RESULT
Kelly	INFS2200	6
Sally	INFS1200	4

Name	COURSE	RESULT
John	INFS1200	7
Kelly	INFS2200	6
Emily	DECO1400	5
Sally	INFS1200	4
Thomas	INFS2200	4
Emily	COMP1002	6
John	COMP4500	6

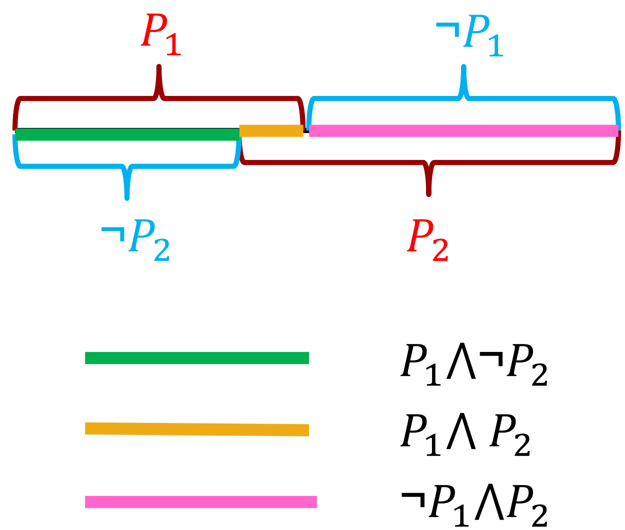
+ Question 2

- A correct fragmentation needs to satisfy the properties of (1) completeness, (2) disjointness and (3) reconstructability.

Please discuss if each of the following fragmentation schemes meet these criteria, and how the original relation can be reconstructed.

Q2: Fragmentation Properties

- a) Primary horizontal fragmentation using minterm predicates.
  - Simple predicates
    - $P_1: SAL < 40{,}000,\ P_2: SAL > 30{,}000$
    - $\neg P_1: SAL \geq 40{,}000, \neg P_2: SAL \leq 30{,}000$
  - Minterm predicates
    - $M_2: SAL \leq 30{,}000, M_3: 30{,}000 < SAL < 40{,}000, M_4: SAL \geq 40{,}000$
  - Completeness
  - Disjointness
  - Reconstructability
    - Union



+ Q2: Fragmentation Properties

- b) Derived horizontal fragmentation using semi-joins with the owner relation correctly fragmented.
  - Completeness
    - Foreign key
  - Disjointness
  - Reconstructability

ID	NAME	AGE	SAL
1289	John	24	12000
0988	Kelly	42	30000
6543	Emily	19	28760
2345	Thomas	23	29999

S.Fragment 1

S_ID	COURSE	RESULT
1289	INFS1200	7
1289	INFS2200	6
0988	COMP4500	6
0988	INFS2200	5
6543	INFS1200	4
2345	INFS1200	7

R.Fragment 1

S_ID	COURSE	RESULT
1289	INFS1200	7
1289	INFS2200	6
8907	DECO1400	5
8907	INFS1200	4
8907	INFS2200	4
7643	COMP1002	6
0988	COMP4500	6
0988	INFS2200	5
6543	INFS1200	4
0986	INFS1200	7
2345	INFS1200	7

R

THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

Q2: Fragmentation Properties

- c) Vertical fragmentation with primary key attributes fully replicated in all fragments.
  - Completeness
  - Disjointness
  - Reconstructability
    - Join

ID	NAME	AGE	SAL
1289	John	24	12000
8907	Sally	29	67050
7643	Elvin	22	51980
0988	Kelly	42	30000
6543	Emily	19	28760
0986	Sally	46	54000
2345	Thomas	23	29999

ID	NAME	AGE
1289	John	24
8907	Sally	29
7643	Elvin	22
0988	Kelly	42
6543	Emily	19
0986	Sally	46
2345	Thomas	23

ID	SAL
1289	12000
8907	67050
7643	51980
0988	30000
6543	28760
0986	54000
2345	29999

+ Q2: Fragmentation Properties

- d) Vertical fragmentation with primary key attributes replicated in only some fragments.
  - Completeness
  - Disjointness
  - Reconstructability

<u>ID</u>	NAME	AGE	SAL
1289	John	24	12000
8907	Sally	29	67050
7643	Elvin	22	51980
0988	Kelly	42	30000
6543	Emily	19	28760
0986	Sally	46	54000
2345	Thomas	23	29999

<u>ID</u>	NAME	AGE
1289	John	24
8907	Sally	29
7643	Elvin	22
0988	Kelly	42
6543	Emily	19
0986	Sally	46
2345	Thomas	23

NAME	SAL
John	12000
Sally	67050
Elvin	51980
Kelly	30000
Emily	28760
Sally	54000
Thomas	29999

+ Question 3

- After a relation R is fragmented correctly, how to insert a new tuple into R or update an existing tuple? Using primary horizontal fragmentation as an example.

ID	NAME	AGE	SAL
1289	John	24	12000
8907	Sally	29	67050
7643	Elvin	22	51980
0988	Kelly	42	30000
6543	Emily	19	28760
0986	Sally	46	54000
2345	Thomas	23	29999

ID	NAME	AGE	SAL
1289	John	24	12000
0988	Kelly	42	30000
6543	Emily	19	28760
2345	Thomas	23	29999

Fragment 1 (SAL ≤ 30000)

Fragment 2 (SAL > 30000)

ID	NAME	AGE	SAL
8907	Sally	29	67050
7643	Elvin	22	51980
0986	Sally	46	54000

ID	NAME	AGE	SAL
1356	James	25	45386

+ Question 3

- After a relation R is fragmented correctly, how to insert a new tuple into R or update an existing tuple? Using primary horizontal fragmentation as an example.

Fragment 1 (SAL ≤ 30000)

ID	NAME	AGE	SAL
1289	John	24	12000
0988	Kelly	42	30000
6543	Emily	19	28760
2345	Thomas	23	29999

Fragment 2 (SAL > 30000)

ID	NAME	AGE	SAL
8907	Sally	29	67050
7643	Elvin	22	51980
0986	Sally	46	54000
1356	James	25	45386

ID	NAME	AGE	SAL
7643	Kevin	20	21374

+ Question 3

- After a relation R is fragmented correctly, how to insert a new tuple into R or update an existing tuple? Using primary horizontal fragmentation as an example.

ID	NAME	AGE	SAL
1289	John	24	12000
0988	Kelly	42	30000
6543	Emily	19	28760
2345	Thomas	23	29999
7643	Kevin	20	21374

Fragment 1 (SAL ≤ 30000)

ID	NAME	AGE	SAL
8907	Sally	29	67050
7643	Elvin	22	51980
0986	Sally	46	54000
1356	James	25	45386

Fragment 2 (SAL > 30000)

+ Question 3

- After a relation R is fragmented correctly, how to insert a new tuple into R or update an existing tuple? Using primary horizontal fragmentation as an example.
  - Insertion:
    - **If predicates are not on the primary key**, check all the fragments to make sure the primary key does not exist.
    - Compare the value with predicates and insert it into the right fragment.
  - Update: Search the record by WHERE condition, update the value of SAL and **send the record** into the right fragment according to the new value of SAL.