匠人学院™
jiangren.com.au

Final Exam Part 2

INFS 3200/7907

Bitwise operation

100

Distributed Database Design

- 1/5: How to design a distributed database system (data fragmentation and data replication)
- 1/5: How to distribute a database system
- 2/5: Properties (Completeness, Disjoint, Reconstructibility)
- Suggestions: Be familiar with the concepts

Question 1. Distributed Databases (8 marks)

A Semijoin is a special type of join operation that can be used in distributed database design and distributed query processing.

(a) [4 marks] Consider two relations R(A, B) and S(X, A, C), where S.A is the foreign key. Assume that R is horizontally fragmented based on its attribute A into R1 and R2. Please use the semijoin operation to define the derived horizontal fragmentation of S based on the fragmentation of R, and explain how your S fragmentation meets the reconstruction property.

(b) [4 marks] Assume that the relation R(A, B) is located on site 1 and that the relation S(X, A, C) is located on site 2. Consider a join query $R \bowtie_A S$ at site 1. Please give a step-by-step query execution plan using semijoin operations to process this query.

知识点：Distributed Database Design
(a) Horizontal and Vertical fragmentation
(b) Semi join

Approach 1: site 2 --- S --→ site 1

Approach 2: site 1 – all unique values of R.A → site 2
site 2 – S semijoin R.A -→ site 1

Distributed Database Design

- 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

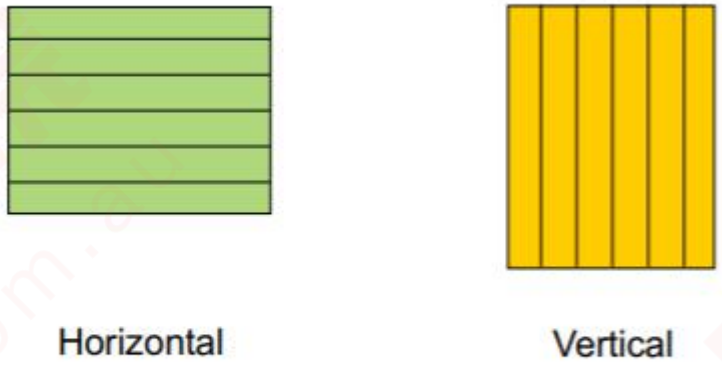
ID	NAME	AGE	SAL
8907	Sally	29	67050
7643	Elvin	22	51980
0986	Sally	46	54000

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

Vertical VS Horizontal Fragmentation

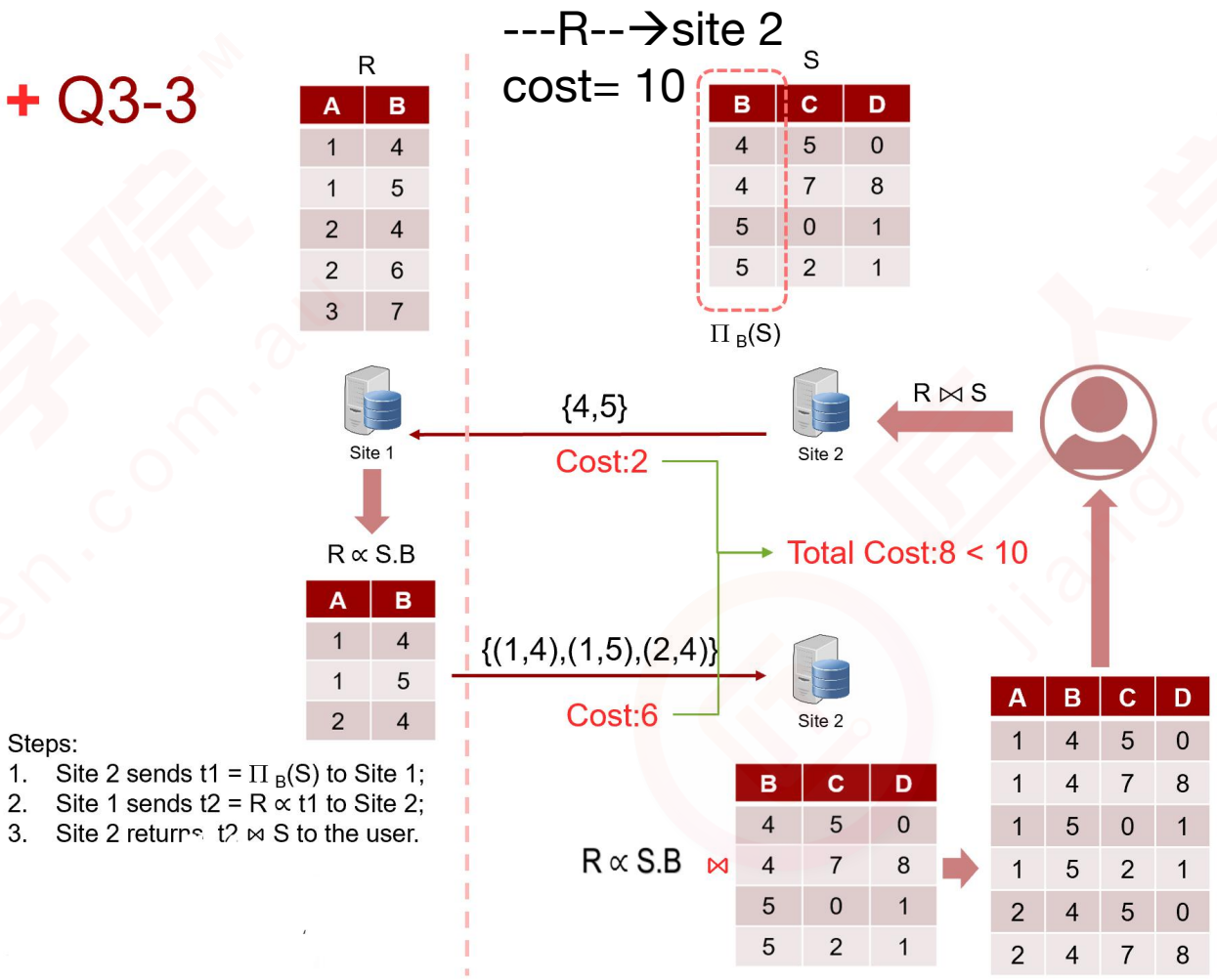


Primary: using predicates on this table only
Derived: using foreign relation predicates

1. Must satisfy 3 properties: disjoint, reconstructibility, completeness
2. For vertical fragmentation, each fragment must include the primary key

Semijoin

+ Q3-3



Steps:

1. Site 2 sends t1 = $\Pi_B(S)$ to Site 1;
2. Site 1 sends t2 = R ∝ t1 to Site 2;
3. Site 2 returns t2 ⋈ S to the user.

Note:
For R semijoin B

B serves as a filter

Question 1. Distributed Databases (3 marks)

Consider relation S(A, B, C, D), where A is the primary key attribute. S is vertically fragmented into S1(A, B) and S2(A, C, D) and allocated at site N1 and N2 respectively.

How to insert a tuple (a, b, c, d) into S? This insert operation must meet the atomicity property.

1. Check if “a” exists in S (to meet primary key constraints).
2. Check with N1 and N2 to see if S1 and S2 can be updated (i.e., if update locks can be granted)
3. Insert (a,b) to S1 and insert (a,c,d) to S2.

Note: We need to use 2PC to guarantee automicity.

2-Phase Commit Protocol

- **Two-Phase Commit** Protocol ≠ 2PL
  - 2PL ⟹ serializability
  - 2PC ⟹ atomic transactions
- Site at which Tx originates is **coordinator**
- Other sites at which it executes are **subordinates**.

- When an Tx wants to commit:
- Coordinator sends **prepare** msg to each subordinate.
- Subordinate writes an **abort** or **prepare** log record and then sends a **no** or **yes** msg to coordinator.

- If coordinator gets **unanimous** yes votes:
  - The coordinator writes a commit log record and sends commit message to all subordinates
- Else
  - writes abort log rec, and sends abort message
- Subordinates write abort/commit log record based on the message they get, then send **ack** message to coordinator
- Coordinator writes end log record after getting all **acks**

Question 2. Distributed Databases (4 marks)

When a relational table is vertically partitioned into two tables, the primary key needs to be duplicated in both partitions. Please construct a simple example to illustrate the problem(s) that can be caused if the primary key is not duplicated.

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

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

Question 1 [11 marks] Data fragmentation and data replication are two important steps of distributed database design.

(a) [4 marks] Suppose that we have the following two relations *MachineType* and *Skills*, where the attribute *Machine* in relation *Skills* is a foreign key referring to the attribute *Machine* in relation *MachineType*. Given the following two predicates, what are the primary horizontal fragments for relation *MachineType* and the derived horizontal fragments for relation *Skills*? (Hint: the answer should contain four tables in total)

- Type = "CPU"
- Type = "Tablet PC"

Skills

Person	Machine
Lisa	AMD A4-3300
John	AMD A4-3300
Brown	AMD A4-3300
John	Samsung700T1A-A01
Young	Samsung700T1A-A01
Kate	Toshiba AT100
Brown	HL-3070CW
John	HL-3070CW

MachineType

Type	Machine
CPU	AMD A4-3300
CPU	Intel Core i7
Tablet PC	Samsung700T1A-A01
Tablet PC	Samsung700T1A-H01
Tablet PC	Toshiba AT100
Printer	HL-3070CW

(b) [2 marks] The correctness of a fragmentation is usually measured by three criteria, namely completeness, disjointness, and reconstructability. Is the above fragmentation correct, why or why not?

(c) [3 marks] What is data replication? What are the benefits of having data replications, at what costs?

Data Fragmentation

- Type = “CPU”
- Type = “Tablet PC”

Skills

Person	Machine
Lisa	AMD A4-3300
John	AMD A4-3300
Brown	AMD A4-3300
John	Samsung700T1A-A01
Young	Samsung700T1A-A01
Kate	Toshiba AT100
Brown	HL-3070CW
John	HL-3070CW

MachineType

Type	Machine
CPU	AMD A4-3300
CPU	Intel Core i7
Tablet PC	Samsung700T1A-A01
Tablet PC	Samsung700T1A-H01
Tablet PC	Toshiba AT100
Printer	HL-3070CW

Data Replication

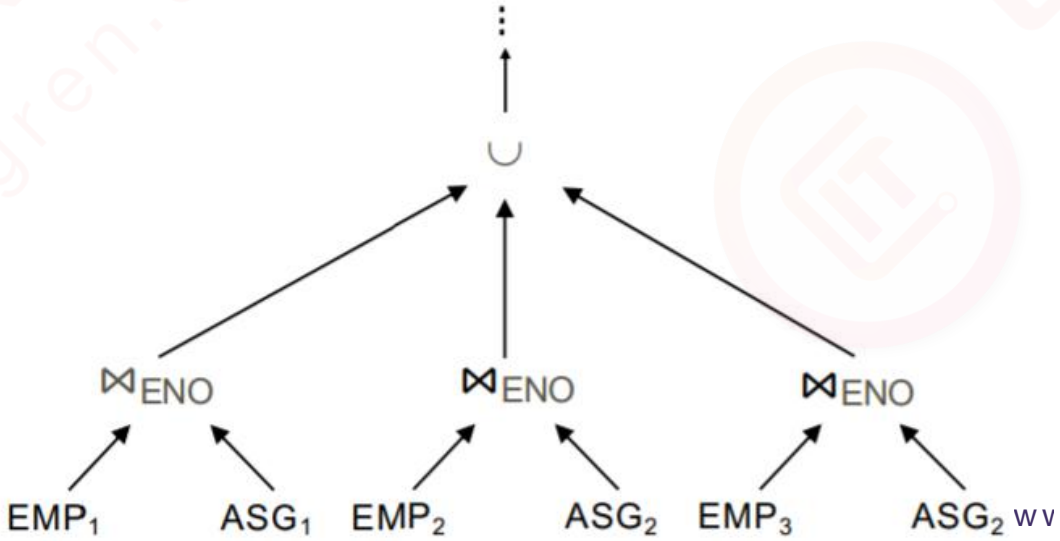
- A centralized system has a single copy of all data
- One way to distribute a database is to copy some or all data, possibly several times
  - Each copy stored at a different node in the network
  - Query routed to “nearest” node
- Fragments can be replicated
  - Full replication, and partial replication
- All copies of replicated data must be updated in a single transaction to maintain atomicity and consistency

Pros: back up, return queried records faster

Cons: data update, storage

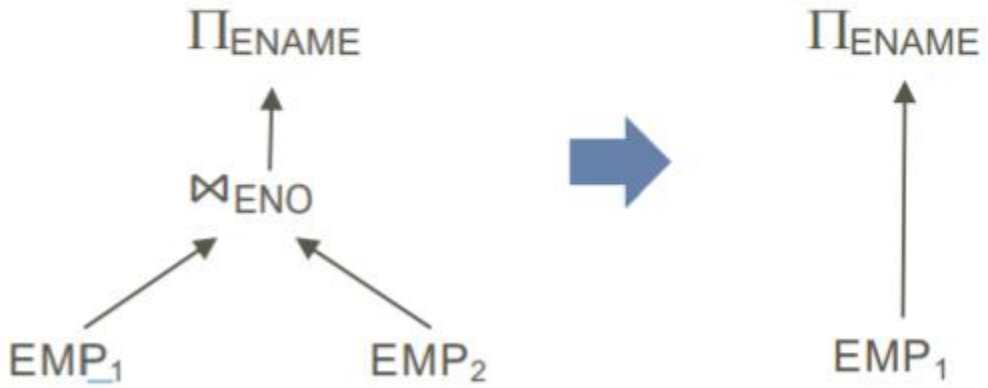
Distributed Query Processing

- 2/5: How to process queries in distributed database systems
- 3/5: Work out query execution plans using notations (similar to INFS2200)
- Suggestions: Practice the steps on the slides. Be familiar with the notations.

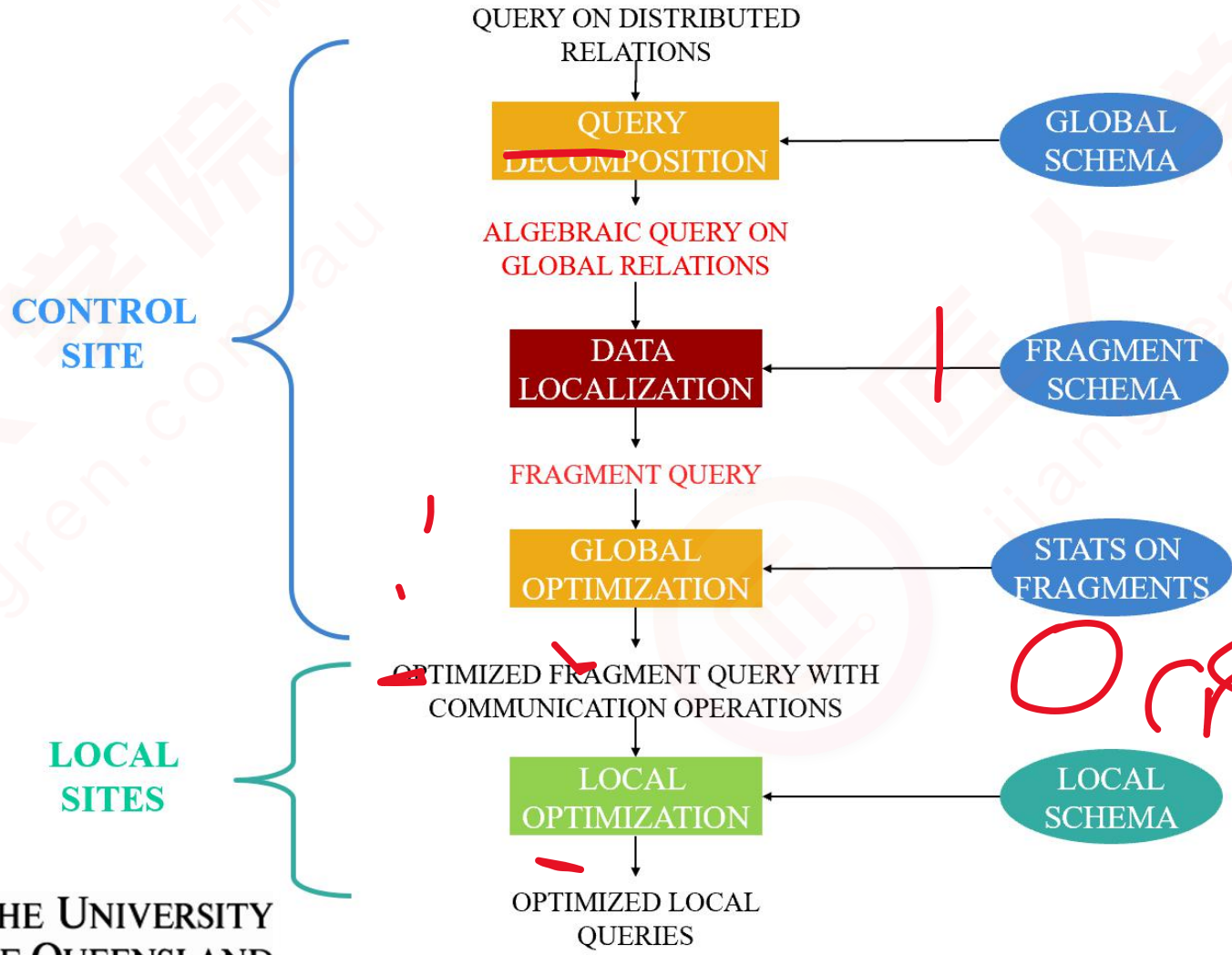


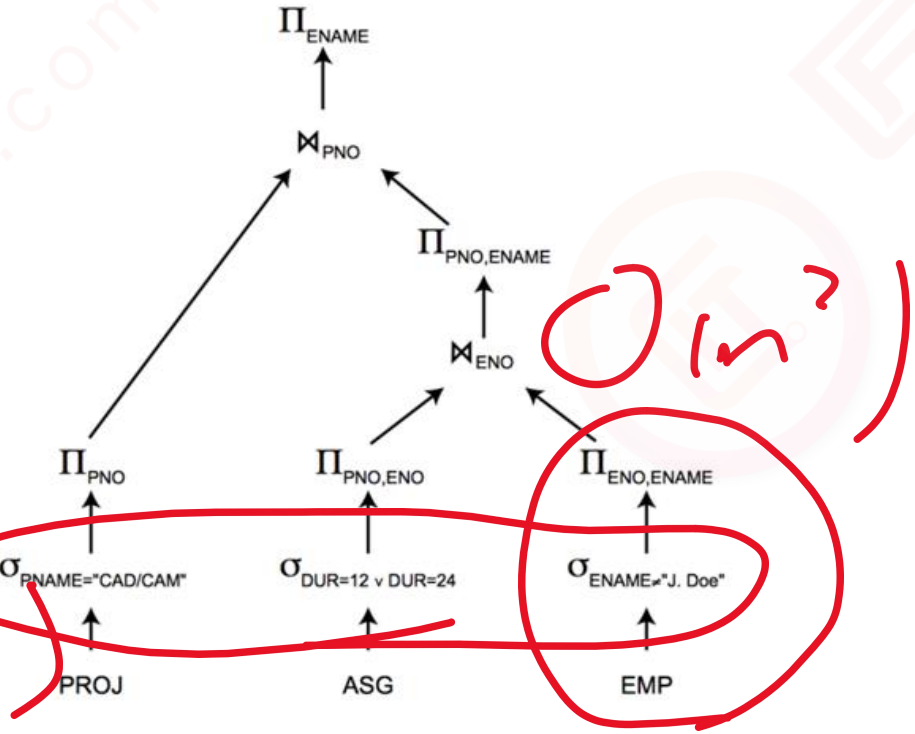
$EMP_1 = \Pi_{ENO,ENAME}(EMP)$; $EMP_2 = \Pi_{ENO,TITLE}(EMP)$

```
SELECT  ENAME
FROM    EMP
```



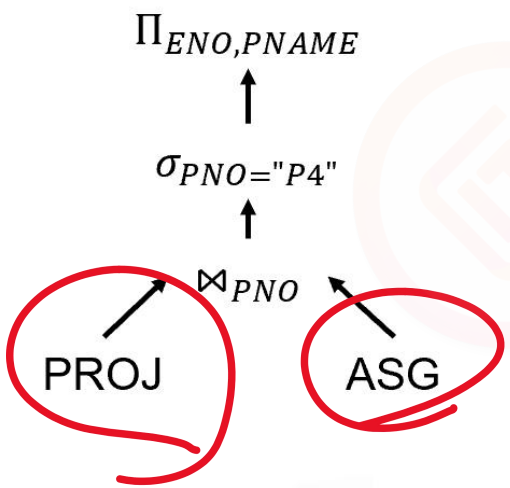
Query Decomposition

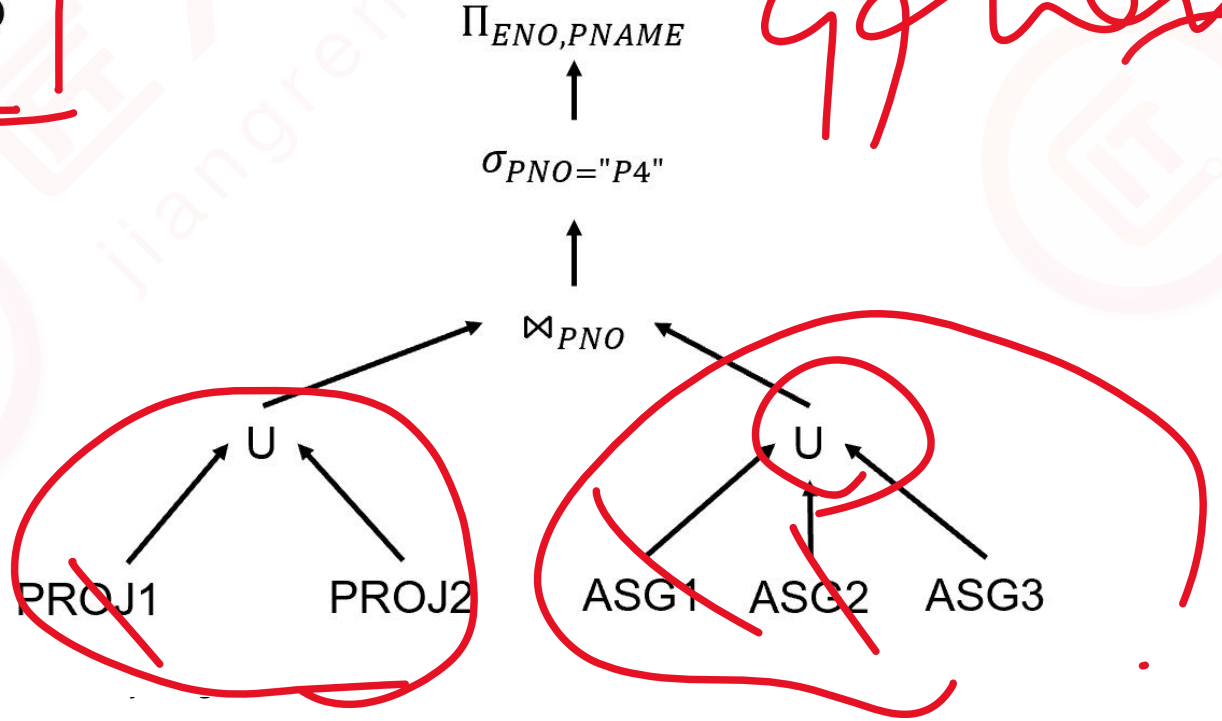




Data Localization

- Localized query
  - SELECT ENO, PNAME
  - FROM PROJ1 U PROJ2, ASG1 U ASG2 U ASG3
  - WHERE
    - PROJ.PNO = ASG.PNO
    - AND PNO = "P4"

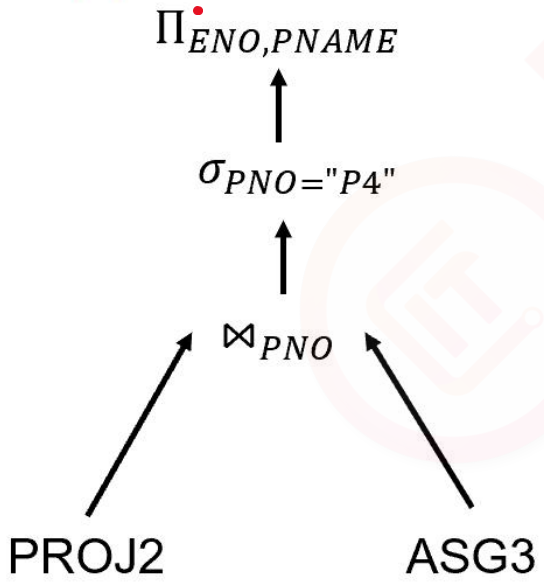


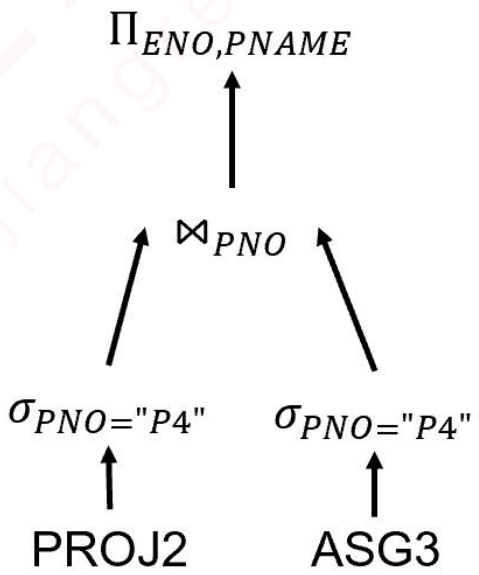


Reduced Query

注意题干要求reduced query还是localized query

- SELECT ENO, PNAME
- FROM PROJ2, ASG3
- WHERE
  - PROJ2.PNO = ASG3.PNO
  - AND PNO = "P4"





Question 2 [5 marks]. Consider a simplified database defined by the following schemas:

STUDENT(SNO, SNAME, PROGRAM)

COURSE(CNO, CNAME, CTITLE)

ENROLLMENT(SNO, CNO, RESULT)

(a) [1 mark] Given the following SQL query, transform it into a query execution tree.

```
SELECT SNO, SNAME, CNAME, RESULT
FROM STUDENT S, COURSE C, ENROLLMENT E
WHERE C.CNO = E.CNO AND S.SNO = E.SNO
```

(b) [2 marks] Assume that relation COURSE is horizontally fragmented as follows:

$COURSE_1 = \sigma_{CNO \leq C100}(COURSE)$

$COURSE_2 = \sigma_{CNO > C100}(COURSE)$

and that relation ENROLLMENT is horizontally fragmented as follows:

$ENROLLMENT_1 = \sigma_{CNO \leq C100}(ENROLLMENT)$

$ENROLLMENT_2 = \sigma_{C100 < CNO \leq C200}(ENROLLMENT)$

$ENROLLMENT_3 = \sigma_{CNO > C200}(ENROLLMENT)$

Show an equivalent query execution tree of the above query after replacing relations with fragments.

(c) [2 marks] Show an optimised query execution tree of the above query after applying reduction rules.

Data Warehouse Design

- 1/5: Concepts of data warehouse
- 3/5: Know how to write table schemas (snowflake schema, star schema)
- 1/5: Definitions of OLAP queries
- 3/5: Give the query result of an OLAP query

Question 2. Data warehouses (11 marks)

Consider a sales fact table with three dimensions (time, location, product).

(a) (3 marks) Explain what a data cube is in data warehousing systems.

(b) (3 marks) Explain what a dicing operation is.

(c) (2 marks) It is not common for data warehousing systems to support update operations. Describe a reason why supporting updates in data warehouses is not a good idea. Briefly justify your answer.

(d) (3 marks) A data warehouse can often make use of materialized views (e.g., using materialized data cubes). Discuss advantages and disadvantages of building materialized views in data warehouses.

知识点：data warehouse design

a data cube (or datacube) is a multi-dimensional ("n-D") array of values. The data cube is used to represent data (sometimes called facts) along some measure of interest. (tensor)

Select a subset of all dimensions

Data warehouse mainly stores historical data

Pros and Cons of View Materialization

+ View Materialization in DW

- Advantages
  - OLAP queries are typically aggregate queries, e.g., CUBE
  - Pre-aggregation is essential for interactive response time
    - Pre-calculate expensive joins
    - Speed up online OLAP queries
- Disadvantages
  - Increase storage cost
  - The content of the materialized views must be maintained when the underlying detail tables are modified
  - Need carefully designed strategy to trade-off between query performance and accessibility to up-to-date data

Data Replication has similar pros and cons

(a) [4 marks] Discuss different roles that views play in the following systems. (You should give at least one type of use for each system)

- Relational database systems
- Distributed database systems
- Data warehousing systems
- Federated database systems

Views

- Relational DB:
  - You can query a view like you can a table. A view can combine data from two or more table, using joins, and also just contain a subset of information.
  - Specifying privileges: If the owner A of a relation R wants another account B to be able to retrieve only some fields of R, then A can create a view V of R that includes only those attributes and then grant SELECT on V to B

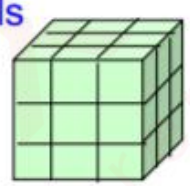
- Distributed DB: used in the bottom-up approach of DB design (data integration).

Views

- Data Warehouse:
  - Store pre-calculated expensive joins to speed up online OLAP queries. (interactive response time)
- Federated database systems:
  - Provide virtual view of integrated data without actually bringing data into a physical centralized database

OLAP Queries

- Pivoting (cross-tabulation)
  - Rotate data cube to show a different orientation of axes
- Roll-up
  - Move up concept hierarchy, grouping into larger units along a dimension with more generalization
- Drill-down
  - Disaggregate to a finer-grained view to show more details
- Slice and dice
  - Perform projection operations on the dimensions



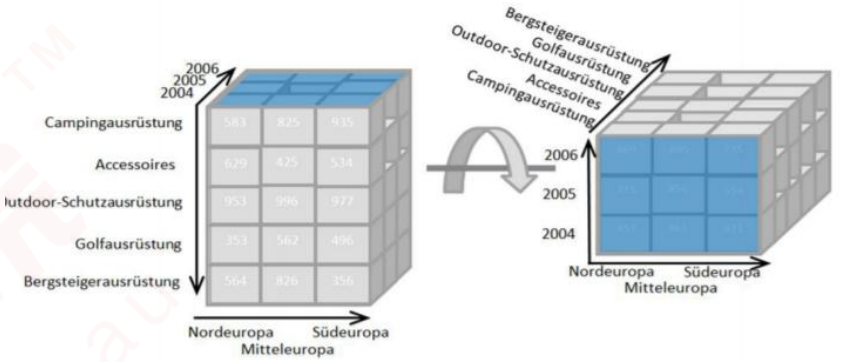
- Other operations, such as arithmetic (to get derived values), sorting, selection.

	Product		Total
Day	**Milk**	**Bread**	**Perishables**
9/2/2012	8952	5836	14788
10/2/2012	7910	8059	15969
	Product Group		Total
Day	**Perishables**	**Canned Goods**	**All Groups**
9/2/2012	14788	55621	206771
10/2/2012	15969	68123	310885

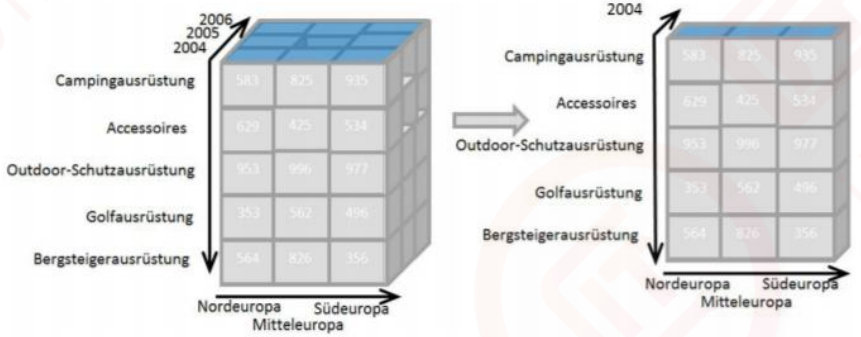
Roll-up

	Product Group		Total
Day	**Perishables**	**Canned Goods**	**All Groups**
9/2/2012	14788	55621	206771
10/2/2012	15969	68123	310885
	Product		Total
Day	**Milk**	**Bread**	**Perishables**
9/2/2012	8952	5836	14788
10/2/2012	7910	8059	15969

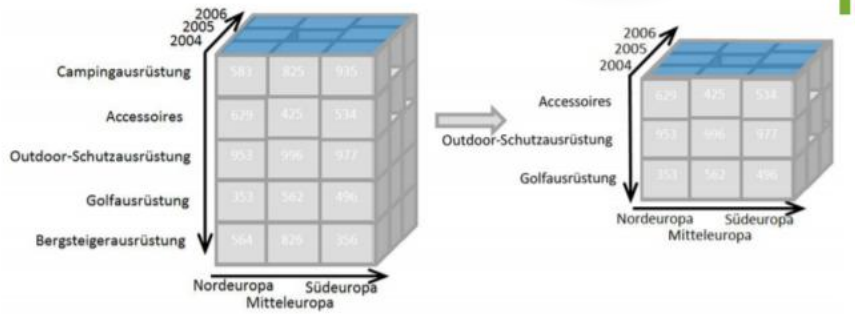
Drill-down



Pivoting



Slicing



Dicing

Question 3 [8 marks] Suppose that a data warehouse for *University* consists of the following three dimensions: *student*, *course*, and *semester*, and one measure *grade*. At the lowest conceptual level (e.g., for a given student, course, and semester combination), the *grade* measure stores the actual course grade of the student. At higher conceptual levels (e.g., for a given student and course combination), *grade* stores the average grade for the given combination.

(a) [4 marks] The data warehouse can be modelled by either a *star schema* or a *snowflake schema*. Briefly describe the similarities and the differences between the two models, and then analyse their advantages compared to one another.

(b) [2 marks] Given the base cuboid on *{student, course, semester}*, how can we obtain the average grade of "Advanced Database Systems" course for each student using OLAP operations? (*Hint: you only need to explain which operations are performed on which dimensions in either SQL or plain English*)

(c) [2 marks] *Bitmap indexing* is a useful technique in data warehousing. Taking this cube as an example, briefly discuss the advantages and problems of using a bitmap index structure.

BitMap

ID	Name	Sex	Nationality
1	Wang Ming	Male	China
2	Natasha	Female	Russia
3	Lucy	Female	Australia
4	Jack	Male	Australia

M	F
1	0
0	1
0	1
1	0

China	Russia	Australia
1	0	0
0	1	0
0	0	1
0	0	1

Pros: small size, speed up query

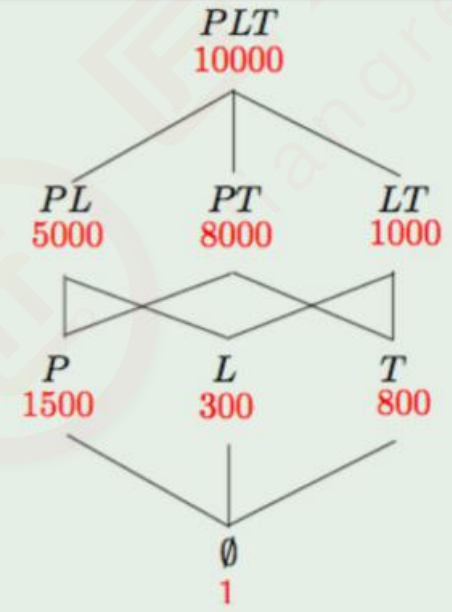
Cons: not suitable for attribute with high cardiality. Not suitable for attributes that are updated frequently.

#columns = #unique values (cardiality)
#rows = #rows of the original table

Data Warehouse Implementation

- 2/5: Different indexing methods
- 2.5/5: View Materialization Concepts
- 3/5: Benefit of materialized views (calculate the benefits)

Q	cost
{PLT}	10000
{PL}	10000
{PT}	10000
{LT}	1000
{P}	10000
{L}	1000
{T}	1000
∅	1000



u

Question 3. Data Warehouses (6 marks)

Materialized cuboids are pre-computed and stored on disk. A data warehouse can often make use of materialized cuboids.

a) (2 marks) Suppose that the cuboid on *{student, semester}* is materialized. Among the following group-by queries, which queries can benefit from this materialized cuboid?

{student, course, semester}

{student, course}

{student, semester}

{course, semester}

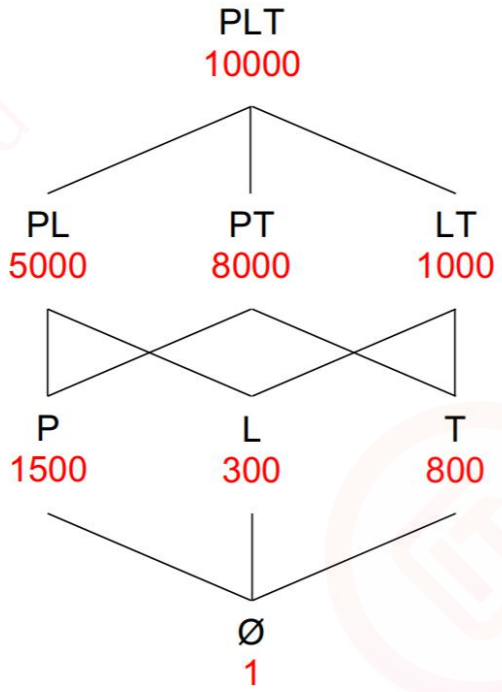
{student}

{course}

{semester}

Ø

b) (4 marks) Suppose that a data warehouse consists of the following three dimensions: *product* (P), *location* (L), and *time* (T), and one measure *sales*. Below is a lattice of all possible cuboids created on the data warehouse dimensions. Each of the numbers shows the cost of using the corresponding cuboid when it is materialized to answer a group-by query. Assume that all the queries are issued with the same frequency, and we have already materialized two cuboids: {*PLT*} and {*PL*}. Which cuboid should be materialized next using the greedy algorithm and why?

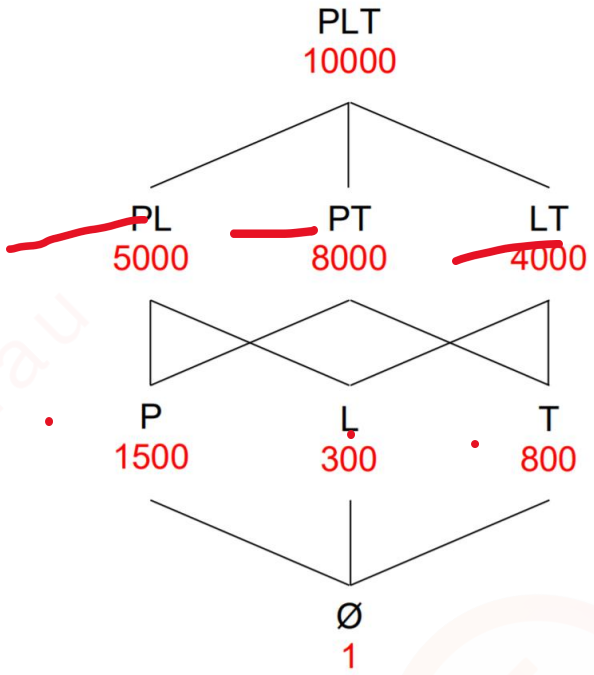


LT

PLT: 10000. query on PLT
PL: 5000. Query on PL
PT: 10000. Query on PLT
LT: 1000. Query on LT
P: 5000. Query on PL
L: 1000. Query on LT (LT < PL)
T: 1000. Query on LT
NULL: 1000. Query on LT

Note:
PLT must be materialised

Question 3 [8 marks]. Suppose that a data warehouse for *Company* consists of the following three dimensions: *product* (P), *location* (L), and *time* (T), and one measure *sales*. Below is a lattice of all possible cuboids created on the data warehouse. Each of the numbers shows the cost of using the corresponding cuboid when it is materialized to answer a group-by query.



Suppose that the frequency distribution of all the group-by queries is as follows:

{PTL (0.05), PL (0.25), PT (0.15), LT (0.1), P (0.2), L (0.1), T (0.1), Ø (0.05)}

What are the first two cuboids that should be materialized in order to minimize total query cost, and why?

We must materialize PLT.

Choose one from PL, PT, LT

Let's say we choose to materialize PL
PLT: 10000 x 0.05
PL: 5000 x 0.25
PT: 10000 x 0.15
LT: 10000 x 0.1
P: 5000 x 0.2
L: 5000 x 0.1
T: 10000 x 0.1
NULL: 5000 x 0.05
Weighted sum

Question 3 [8 marks] A data warehouse is usually represented by either a star schema or a snowflake schema. Various OLAP operations, e.g., roll-up, drill-down, slice, dice, pivot, etc., can be performed on a data warehouse.

(a) [2 marks] For the following snowflake schema, show the equivalent star schema with dimension tables.

Olympiad Table
Olympiad
City
Organizing Committee
Contact address

Venue Table
Venue
Location
Region

Gender Table
Gender

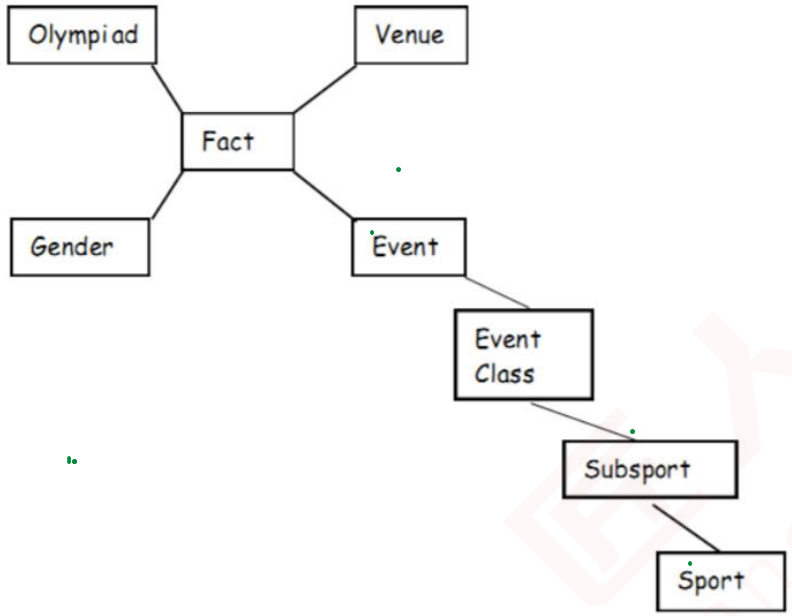
Event Table
Event
Event Class

Event Class Table
Event Class
Subsport

Subsport Table
Subsport
Sport

Sport Table
Sport
Sporting Federation

Fact Table
Olympiad
Venue
Event
Gender
Attendance



(b) [2 marks] What are the advantages and disadvantages of star schema, compared with snowflake schema?

(c) [4 marks] Given the fact table and dimensions in the above snowflake schema, how can we know the attendance of each sport at each venue? (Please explain which OLAP operations are performed on which dimensions in either SQL or plain English)

Star Schema

Day	Month	Qtr	Year
9/2/2012	Feb	1	2012
10/2/2012	Feb	1	2012

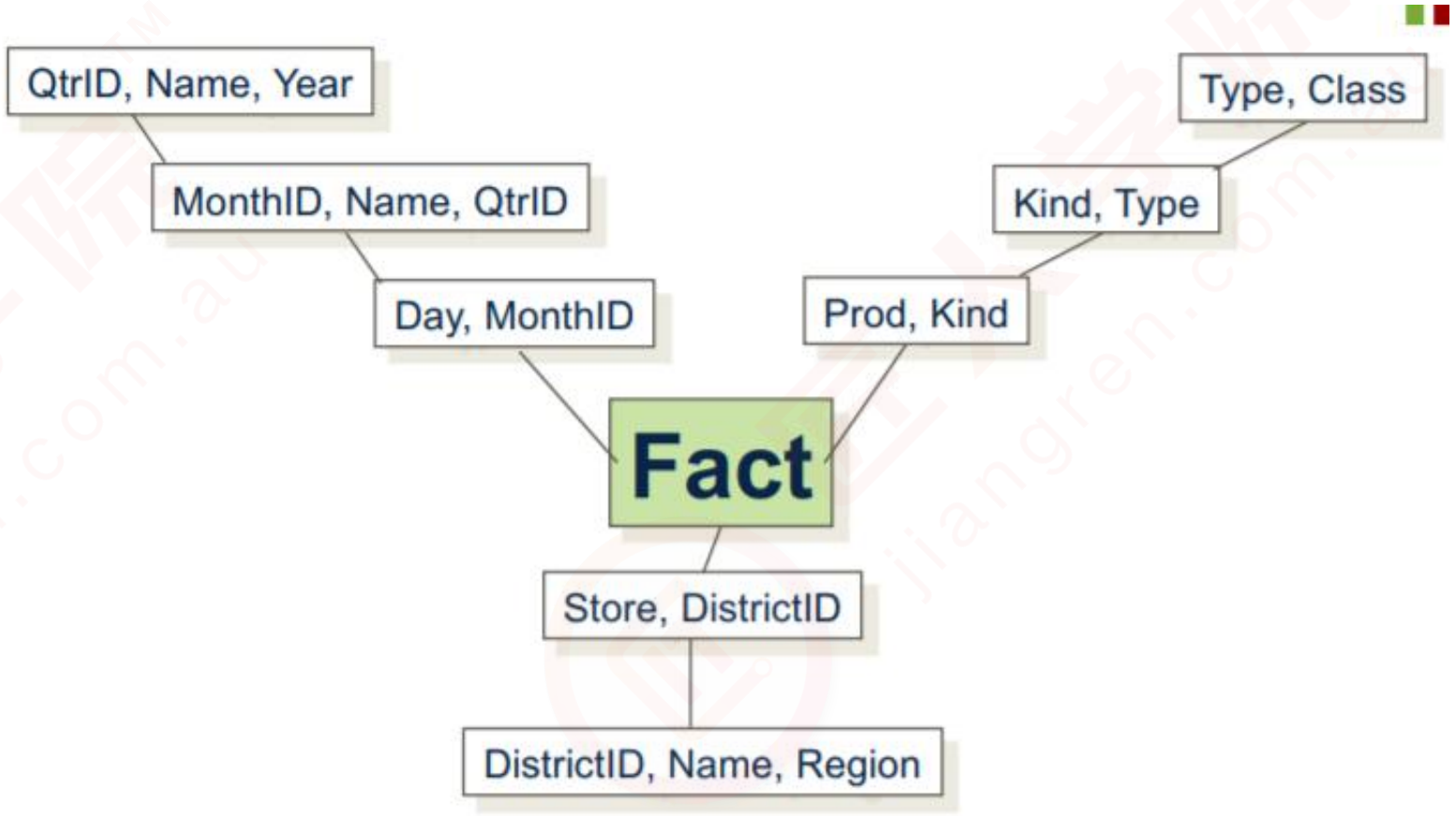
Store	District	Region
Toowong	North	Brisbane
Kenmore	West	Brisbane

Facts

Product	Kind	Type	Class
Milk	Dairy	Perishable	Food
Bread	Bakery	Perishable	Food

The fact table is much larger than dimension tables

Snowflake Schema



Pros: less redundancy

Cons: Costly joins to query the results

The following is a star schema with a fact table and three dimension tables on *Time*, *Location*, and *Goods*.

Time

Day	Month	Qtr	Year
9.2.04	Feb	1	2004
10.2.04	Feb	1	2004

Location

Store	District	Region
Toowong	North	Brisbane
Sunnybank	South	Brisbane

Goods

Product	Kind	Type	Class
Milk	Dairy	Perishable	Food
Bread	Bakery	Perishable	Food

Q1.3 (3 marks) For the above star schema, show the equivalent snowflake schema with dimension tables.

Data Integration

- 1/5: Concepts of data integration
- 2/5: Different types of systems (definitions and differences)
- 3/5: Steps of data integration

Question 4 [9 marks]. Data integration is an important pre-processing step in data warehousing and data mining.

(a) [4 marks] List at least four challenges we need to address in data integration, and give one example for each challenge.

(b) [1 mark] Consider the following two University data models:

University A stores student records in one table:

Student(S#, Fname, Lname, Bdate, Program#)

University B stores student records in two tables for programs 01 and 02 separately:

Prog_01(Sid, Fname, Sname, Credit, email)

Prog_02(Sid, Fname, Sname, Credit, email)

It is known that Lname matches Sname, and S# matches Sid. Define the global schema we can construct from these data models.

(c) [4 marks] Write a SQL query to generate the global schema. (Hint: using views)

Data Integration Challenges

+ Challenges in DB Integration

- Each database could be in a different type of DBMS with different data model, query language, etc.
  - Relational, semi-structured, NoSQL
- Schema heterogeneity
  - S1: Employee(ID, name, address, position, salary)
  - S2: Worker(EID, name, address) Position(PID, salary, from, until)
- Data type heterogeneity
  - Employee ID could be a string or an integer
- Value heterogeneity
  - The “cashier” position could be called “cashier” or “associate”
- Semantic heterogeneity
  - Salary is hourly salary before tax
  - Or salary is net, weekly salary with lunch allowance

Question 5 [9 marks] Data integration is an important pre-processing step in data warehousing and data mining.

(a) [4 marks] List at least four challenges we need to address in data integration, and give one example for each challenge.

(b) [1 mark] Consider the following two University data models. University A stores staff records in one table:

Staff(Emp#, Fname, Lname, Bdate, Dept#)

University B stores staff records in two tables for departments 01 and 02 separately:

Dept_01(Eid, Fname, Sname, Position, Phone#)
Dept_02(Eid, Fname, Sname, Position, Phone#)

It is known that *Lname* matches *Sname*, *Fname* matches *Fname*, and *Emp#* matches *Eid*. Define the global schema we can construct from these data models.

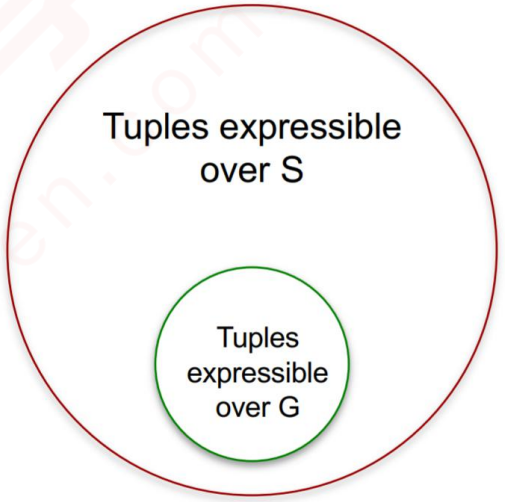
(c) [4 marks] Please write an SQL query to generate the global schema. (*Hint: using views*)

GAV VS LAV

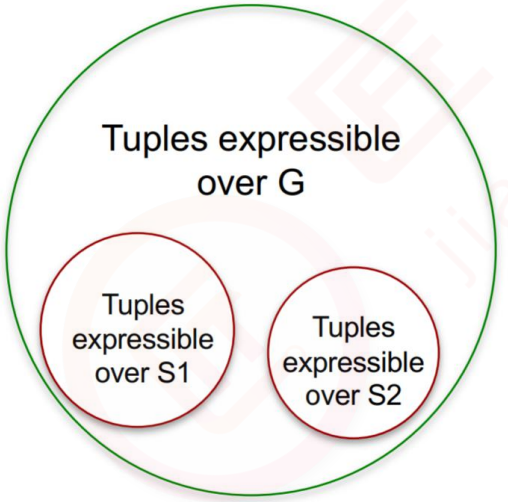
+ View-Based Database Integration

- Problem definition: <*G*, *S*, *M*>
  - *G*: the global schema
  - *S*: a set of local schemas
  - *M*: the mapping to translate queries between *G* and *S*
- A global query is issued over *G* and processed over *S*
- Two popular ways of mapping
  - Global as View (GAV): *G* is a set of views over *S*
    - *M* associates each element in *G* as a query over *S*
    - Employee.EmpID ← Emp.Emp# || DeptXX.S-id
  - Local as View (LAV): *S* is a set of views over *G*
    - *M* associates each element in *S* as a query over *G*
    - Emp.Emp# ← Employee.EmpID

+ A Graphical View



GAV



LAV

- In GAV, the set of possible tuples is defined on *S* while the set of tuples expressible over the sources *S* may be much larger and richer
- In LAV, the set of possible tuples for each *S* is defined on G while the set of tuples expressible over *G* can be much larger (thus, LAV must deal with incomplete answers)

GAV VS LAV

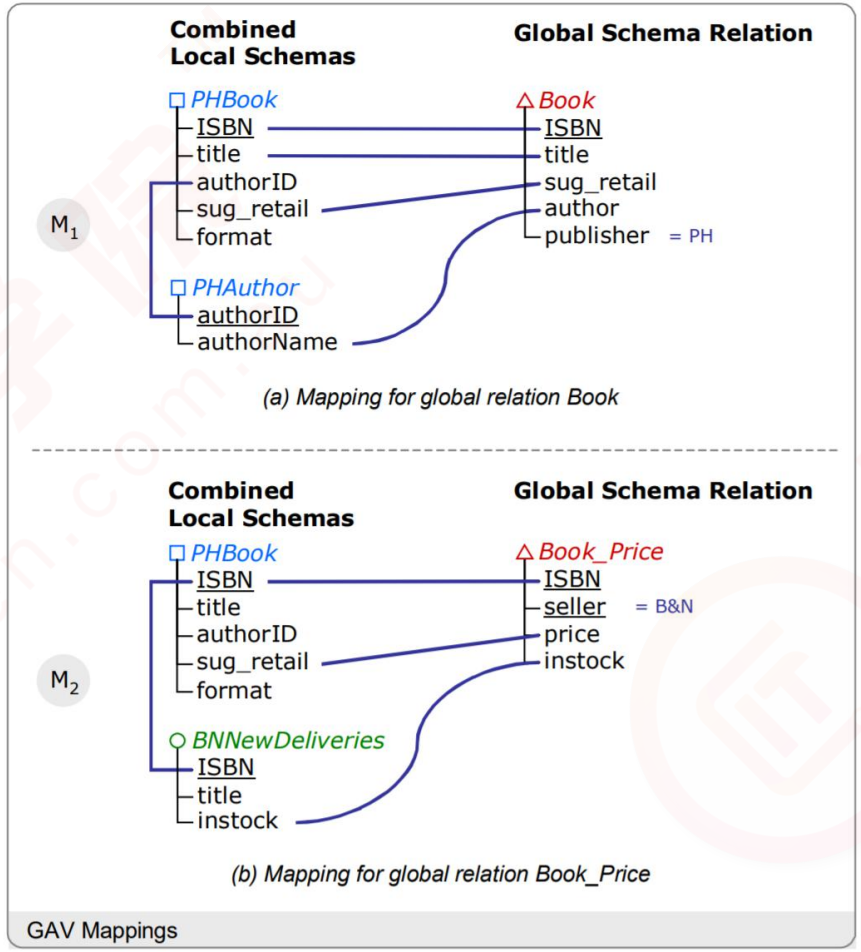


Figure 3: Example of GAV Mappings

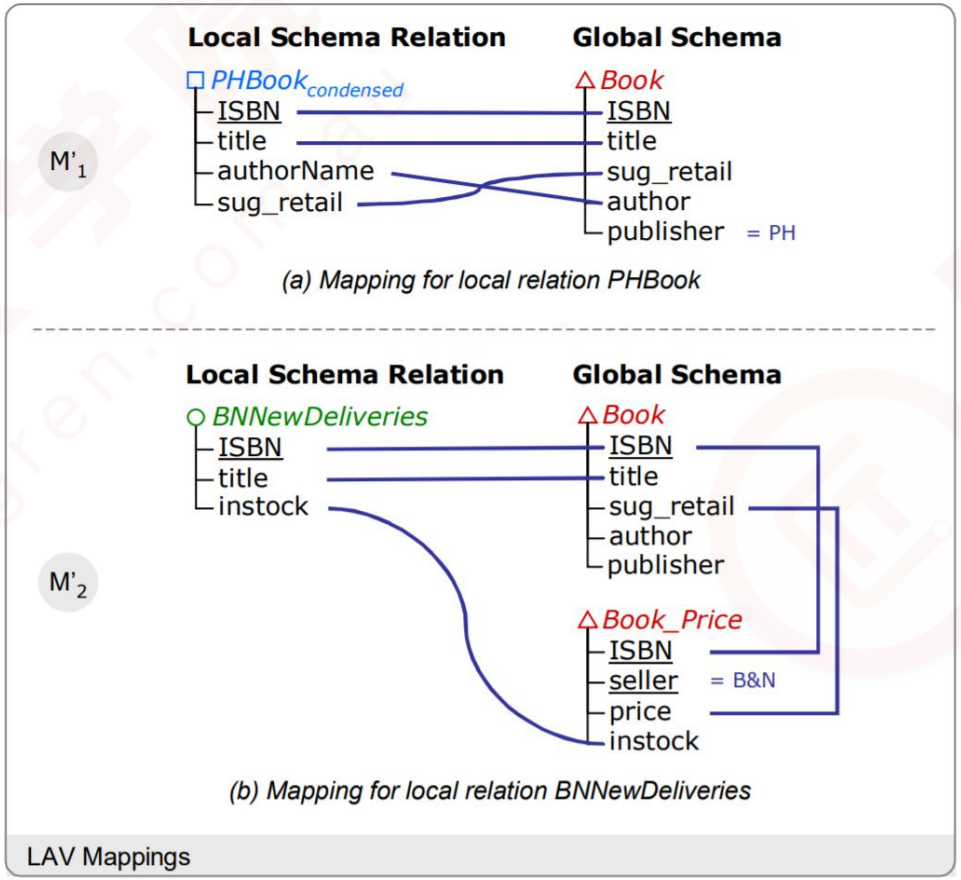


Figure 4: Example of LAV Mappings

SQL Create View

```
CREATE VIEW UNISTAFF AS
SELECT S.Emp#, S.FName, S.Lname
FROM STAFF AS S
UNION
SELECT D.Eid, D.FName, D.Sname
FROM  DEP AS D
```

Data Quality Management

- 1/5: Definitions of data quality (accuracy, consistency, completeness, currency, accessibility, reliability & trust)
- 4/5: Schema Integration
- 1/5: Data Linkage concepts and definitions
- 4/5: Data Linkage approaches (Edit Distance, Q-Gram, Jaccard Coefficients, Cosine Similarities)
- 2/5: Comparisons between these approaches

Question 4. Data Integration (9 marks)

(a) (3 marks) For two strings with *m* and *n* characters respectively, which is the maximum possible edit distance?

(b) (3 marks) What is the edit distance between “maple” and “apple”? Please show the matrix of your calculation.

(c) (3 marks) String similarity can also be measured using Jaccard coefficient based on q-grams. It is a more suitable string similarity measure than the edit distance for two strings that have words in different orders, such as “CEO of Apple” versus “Apple CEO”. Why?

(a)max(m, n)

		k	i	t	t	e	n
	0	1	2	3	4	5	6
s	1	1	2	3	4	5	6
i	2	2	1	2	3	4	5
t	3	3	2	1	2	3	4
t	4	4	3	2	1	2	3
i	5	5	4	3	2	2	3
n	6	6	5	4	3	3	2
g	7	7	6	5	4	4	3

(b)Jaccard coefficient.

Jaccard Distance VS Edit Distance

设计到语序不同的，选择JD

e.g. str1 = “The University of Queensland”

str2 = “The Queensland University”

JD(str1, str2) = 0.6897

ED(str1, str2) = 25

25/ 53 = 0.47

Question 5 [15 marks]. Data quality issues need to be addressed before the data can be released for use by other data analysis applications.

(a) [4 marks] Data quality can be measured from various dimensions. Please list at least four data quality dimensions and give one example of data quality problem for each of these dimensions.

(b) [1 mark] Record linkage is an important task in data quality management. Explain the meaning of record linkage.

(c) [4 marks] Edit distance is a common string similarity measure used in record linkage. Edit distance between two strings is the minimum number of operations (i.e., insert, delete, or replace one character) to transform one string to the other. Compute edit distance between two strings "Serious" and "Ceriers" using the dynamic programming algorithm. Show the calculation step by step in a matrix. What is the edit distance between these two strings?

(d) [2 marks] Jaccard coefficient is another string similarity measure that can be used for record linkage. Assume that we need to use either edit distance or Jaccard coefficient to perform record linkage for a dataset of people's names. Which similarity measure do you suggest to use in the following cases respectively, and why?

- Names are written as either *{first name, last name}* or *{last name, first name}*.
- All the names are written as *{first name, last name}*, but they contain some minor typos.

(e) [4 marks] Efficiency of record linkage should also be considered in practice. Various techniques have been proposed to reduce the number of record comparisons, such as Blocking, Sorted Neighbourhood Approach, Clustering and Canopies, etc. Please explain one of these techniques.

Data Quality Dimensions

+ Data Quality Dimensions

- Accuracy (Erroneous)
  - Postcode “4109” is typed “4019”
- Representational Consistency (Inconsistent)
  - ITEE Vs. Information Technology and Electrical Engineering
- Completeness (Missing)
  - Students don’t have to declare a major till graduation, so major is missing in most enrolments
- Currency (Obsolete)
  - Old phone numbers
- Accessibility (Unavailable)
  - Server down, privacy concerns
- Reliability & Trust (Uncertainty)

(c) [2 marks] Suppose that we need to use edit distance or Jaccard coefficient to perform entity resolution for a dataset of people's names. Which similarity measure do you suggest to use in the following cases respectively, and why?

- Names are written as either *{first name, last name}* or *{last name, first name}*.
- All the names are written as *{first name, last name}*, but they contain some minor typos.

(d) [2 marks] Please give an example that using string similarity alone cannot solve the problem of entity resolution.

Modern Data Management Platforms

- 2.5/5: Cloud Computing Concepts
- 2/5: MapReduce Algorithm

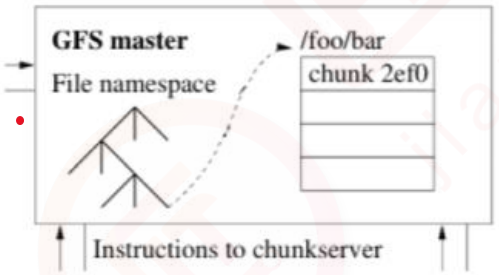
Question 5: Modern Platforms (6 marks)

(a) [3 marks] Explain the main limitation of the Google File System design.

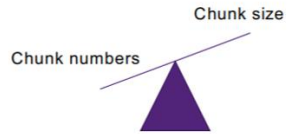
(b) [3 marks] Explain the main efficiency bottleneck of Map/Reduce.

+ The Master

- Maintains all file system metadata
  - Names space, access control info, file to chunk mappings, chunk (including replicas) location, etc.
- Periodically communicates with chunkservers in HeartBeat messages to give instructions and check state
- Helps make sophisticated chunk placement and replication decision, using global knowledge
- For reading and writing, client contacts the Master to get chunk locations, then deals directly with chunkservers
  - Otherwise Master will be a bottleneck for reads/writes
  - However, it can be the single point of failure



Chunks



- Fixed size of 64MB (vs 4kb of cluster size for NTFS)
- Advantages
  - Size of meta data is reduced
  - Involvement of Master is reduced
  - Network overhead is reduced
  - Lazy space allocation avoids internal fragmentation
- Disadvantages
  - Hot spots
    - A small file consists of a small number of chunks, perhaps just one. The chunkservers storing those chunks may become hot spots if many clients are accessing the same file.
    - Solutions: increase the replication factor and stagger application start times; allow clients to read data from other clients

https://support.microsoft.com/en-us/help/140365/default-cluster-size-for-ntfs-fat-and-exfat

Master

- Mater maintains all system metadata
  - Name space, access control info, file to chunk mappings, chunk locations, etc.
- Periodically communicates with chunk servers
  - Through HeartBeat messages
- Advantages:
  - Simplifies the design
- Disadvantages:
  - Single point of failure
- Solution
  - Replication of Master state on multiple machines
  - Operational log and check points are replicated on multiple machines

Question 4. [11 marks] MapReduce is a programming model for processing large-scale datasets with a parallel, distributed algorithm on a cluster of computers. Assume that we have a large census dataset recording information including individual's name, gender, date_of_birth, and country_of_birth. Now we want to use MapReduce to find out the number of people born in each country from the census dataset.

(a) [8 marks] Describe what you need to do in map() and reduce() functions respectively, including the input, output, and the main processing task. (You can use plain English or any programming language).

(b) [1 mark] Explain one way to exchange data between map tasks and reduce tasks.

(c) [2 marks] Some map tasks or reduce tasks may fail due to problems with the node on which they are running. How can such a problem be detected, and how can the MapReduce framework address this problem?

- **"Shuffle" the Map output to the Reduce processors** for all Reduce processors, assigning `K2` key value for each processor with all the Map-generated data associated with that key value

MapReduce

- Map is a function written by the user to take an input key/value pair and produce a set of intermediate key/value pairs
  - Map: `(key1, val1) → list(key2, val2)`
- On completion of the map phase, all the intermediate values for a given output key are combined together into a list and given to a reducer
  - Reduce: `(key2, list(val2)) → list(val2)`

MapReduce

Suppose we have a set of documents (corpus). We want to know how many times each word appears.

I love you
I hate you

<Doc ID, content> --- MAP --> [<“I”, 1>, <“love”, 1>, <“you”, 1>, <“I”, 1>, <“hate”, 1>, <“you”, 1>]

---REDUCE--> [<“I”, 2>, <“you”, 2>, <“love”, 1>, <“hate”, 1>

MapReduce

Document word Count

<Doc ID, content> --- MAP --> [<“UK”, 1>, <“US”, 1>, <“US”, 1>, <“China”, 1>, <“China”, 1>, <“Australia”, 1>]

--REDUCE--> [<“Australia”, 1>, <“China”, 2>, <“US”, 2>, <“UK”, 1>

MapReduce

- Inverted Index
  - The map function parses each document, and emits a sequence of <word, document ID> pairs. The reduce function accepts all pairs for a given word, sorts the corresponding document IDs and emits a <word, list(document ID)> pair. The set of all output pairs forms a simple inverted index. It is easy to augment this computation to keep track of word position

<“D1”, “I love you”> --- MAP --> [<“I”, “D1”>, <“love”, “D1”>, <“you”, “D1”>]
<“D2”, “I hate you”> --- MAP --> [<“I”, “D2”>, <“hate”, “D2”>, <“you”, “D2”>]

---REDUCE--> [<“I”, [“D1”, “D2”]>, <“you”, [“D1”, “D2”]>, <“love”, [“D1”]>, <“hate”, [“D2”]>

MapReduce

+ MapReduce: Fault Tolerance

- Handled via re-execution of tasks
  - Task completion committed through the Master
- What happens if a Mapper or Reducer fails?
  - Each node is expected to report back to the Master periodically with completed work and status updates
  - If a node falls (silent for longer than a threshold time), the Master node records the node as dead and sends out the node's assigned work to other nodes
- What happens if Master fails?
  - Potential trouble (many later work attempting to improve the single-point-of-failure problem)

Data Security And Privacy

- 2/5: Concepts about data security
- 4/5: Security approaches (symmetric key, asymmetric key, digital signature, l-anonymity, k-anonymity)
- 2/5: Comparisons between these approaches (symmetric VS asymmetric, l-diversity VS k-anonymity)

Question 6. Privacy (10 marks)

K-anonymity and differential privacy are two common solutions to privacy-preserving data publishing. For each of these two solutions, please explain (1) what they mean, and (2) what changes they need to make to the data before publishing.

(a) [5 marks] K-anonymity.
(b) [5 marks] Differential privacy.

K-anonymity

Name	Age	ZIP
Andy	20	10000
Bob	30	20000
Cathy	40	30000
Diane	50	40000

adversary's knowledge

Age	ZIP	Disease
20	10000	flu
30	20000	dyspepsia
40	30000	pneumonia
50	40000	gastritis

"generalization"

Name	Age	ZIP
Andy	20	10000
Bob	30	20000
Cathy	40	30000
Diane	50	40000

adversary's knowledge

Age	ZIP	Disease
[20,30]	[10000,20000]	flu
[20,30]	[10000,20000]	dyspepsia
[40,50]	[30000,40000]	pneumonia
[40,50]	[30000,40000]	gastritis

medical records

l-diversity

Name	Age	ZIP
Andy	20	10000
Bob	30	20000
Cathy	40	30000
Diane	50	40000

adversary's knowledge

Age	ZIP	Disease
[20,30]	[10000,20000]	breast cancer
[20,30]	[10000,20000]	breast cancer
[40,50]	[30000,40000]	pneumonia
[40,50]	[30000,40000]	gastritis

2-diverse table

Name	Age	ZIP
Andy	20	10000
Bob	30	20000
Cathy	40	30000
Diane	50	40000

adversary's knowledge

Age	ZIP	Disease
[20,30]	[10000,20000]	breast cancer
[20,30]	[10000,20000]	dyspepsia
[40,50]	[30000,40000]	pneumonia
[40,50]	[30000,40000]	gastritis

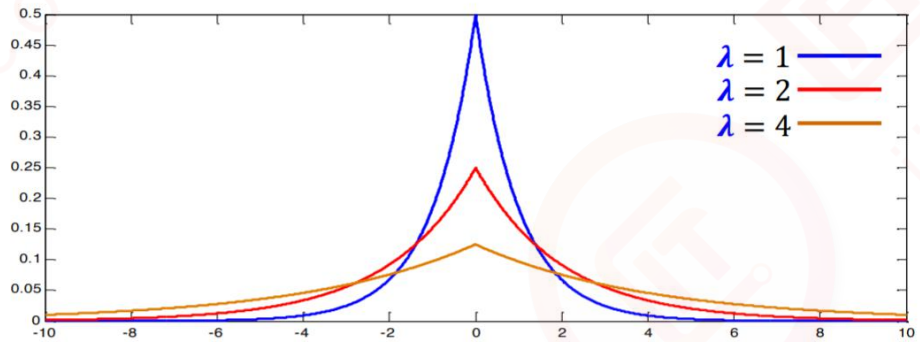
2-diverse table

Differential Privacy

1. We do not publish information that does highly depend on any particular individual record
2. We can introduce some randomness/noises to the data, which does not affect the data utility while giving each individual the refutability
3. We may use a Laplace Distribution to introduce the noises

+ Laplace Distribution Lab(λ)

- $pdf(x|\lambda) = \frac{1}{2\lambda}\exp\left(-\frac{|x|}{\lambda}\right)$;
- variance: $2\lambda^2$; λ is referred as the *scale*



A randomized algorithm A satisfies ε-differential privacy, iff for any two neighboring datasets D and D' and for any output O of A,

$$\Pr[A(D) = O] \leq \exp(\varepsilon) \cdot \Pr[A(D') = O]$$

Distributed Transaction Management

- 1/5: Reviews on the concepts of transactions (INFS2200)
- 2.5/5: New concepts of transactions (in distributed systems)
- 2/5: Different approaches of distributed transactions
- 2/5: Pros and cons of these approaches
- 2/5: Comparisons between similar but different approaches (e.g. synchronous replication VS asynchronous replication)
- Suggestions: be familiar with the concepts, how to use different approaches and their costs/differences.

Question 1 [6 marks]. Data replication is very important in distributed database design.

(a) [2 marks] What are the benefits of having data replications, and at what costs?

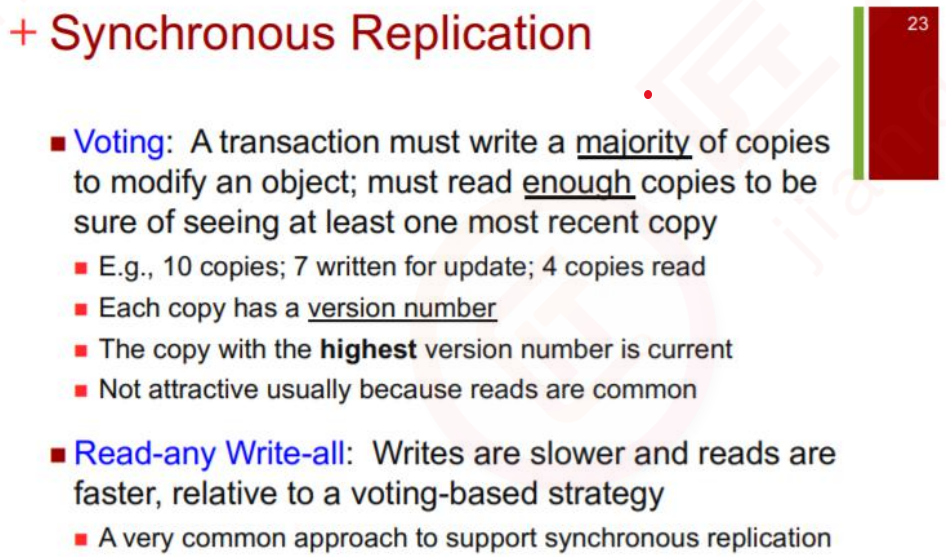
(b) [2 marks] Describe how a voting-based approach works to maintain data consistency among data replications.

(c) [2 marks] If a database is read-intensive with rare updates, should we use a large number of write copies in the voting-based approach? Why or why not?

(a)

pros	cons
Data backup	update
Data retrieval	storage cost

(b)



(c) In the voting approach, we need to read more than one copy

Adopt the Read-any Write-All approach so we only need to read one copy at a time. We rarely need to write because update queries are rare.

Question 4 [9 marks]. Data integration is an important pre-processing step in data warehousing and data mining.

(a) [4 marks] List at least four challenges we need to address in data integration, and give one example for each challenge.

(b) [1 mark] Consider the following two University data models:

University A stores student records in one table:

Student(S#, Fname, Lname, Bdate, Program#)

University B stores student records in two tables for programs 01 and 02 separately:

Prog_01(Sid, Fname, Sname, Credit, email)

Prog_02(Sid, Fname, Sname, Credit, email)

It is known that Lname matches Sname, and S# matches Sid. Define the global schema we can construct from these data models.

(c) [4 marks] Write a SQL query to generate the global schema. (Hint: using views)

```
Create VIEW StudentView AS
SELECT A.S#, A.Fname, A.Lname
FROM Student AS A
UNION
SELECT B.Sid, B.Fname, B.Sname
FROM Prog_01 AS B
UNION
SELECT C.Sid, C.Fname, C.Sname
FROM Rrog_02 AS C
```

Question 6 [7 marks]. Data privacy is a very important issue when publishing data. K-anonymity is a common and simple solution to privacy-preserving data publishing.

(a) [1 mark] What is K-anonymity?

(b) [3 marks] Describe the general approach of K-anonymity.

(c) [1 mark] K-anonymity is still vulnerable in some situations. Explain possible problems of K-anonymity.

(d) [2 marks] L-diversity is a method to reduce the vulnerability of K-anonymity. Describe the general approach of L-diversity, especially its difference with K-anonymity.