Relational Databases: Internals

Responsible:

Dr. Tegawendé F. BISSYANDE

tegawende.bissyande@uni.lu

Course Author:

Eduardo Cunha de Almeida

eduardo@inf.ufpr.br

Teacher Assistant:

Médéric Hurier

mederic.hurier@uni.lu

Course Features

- Sep. 20th - **Introduction to Big Data**

Part 1. Databases and Query Models for Big Data

- Sep. 27th - **Relational Databases: Reminders**
- Oct. 4th - **Relational Databases: Internals**
- Oct. 11th - **NoSQL Databases**
- Oct. 18th - **MapReduce Model**
- Oct. 25th - **Hadoop and Spark**
- Nov. 8th - **Datalog Model**

Part 2. Data Analysis and Machine Learning

- Nov. 15th - **Statistics and Probabilities**
- Nov. 22th - **Communication and Visualization**
- Nov. 29th - **Features Engineering and Supervised Learning**
- Dec. 5st - **Unsupervised and Reinforcement Learning**
- Dec. 12th - **Homework Time**

Section Features

- History
- Database Layout
- Query Processing
- Query Optimization
- Transactions

History

History

1970's

- INGRES Project at Berkeley

  INGRES corp, Sybase, MS SQL Server, PostgreSQL

- System-R Project at IBM

  DB2, Non-StopSQL, HP Allbase

History

1980's

- DB2
- Oracle
- Informix
- Teradata
- Sybase

SQL as the standard query language !

History

1990's

- Postgres
- MySQL
- Illustra (from Postgres)
- BerkeleyDB (embedded K/V)

History

2000's

- MonetDB (as open-source)
- Vertica
- Greenplum
- EnterpriseDB
- Infobright

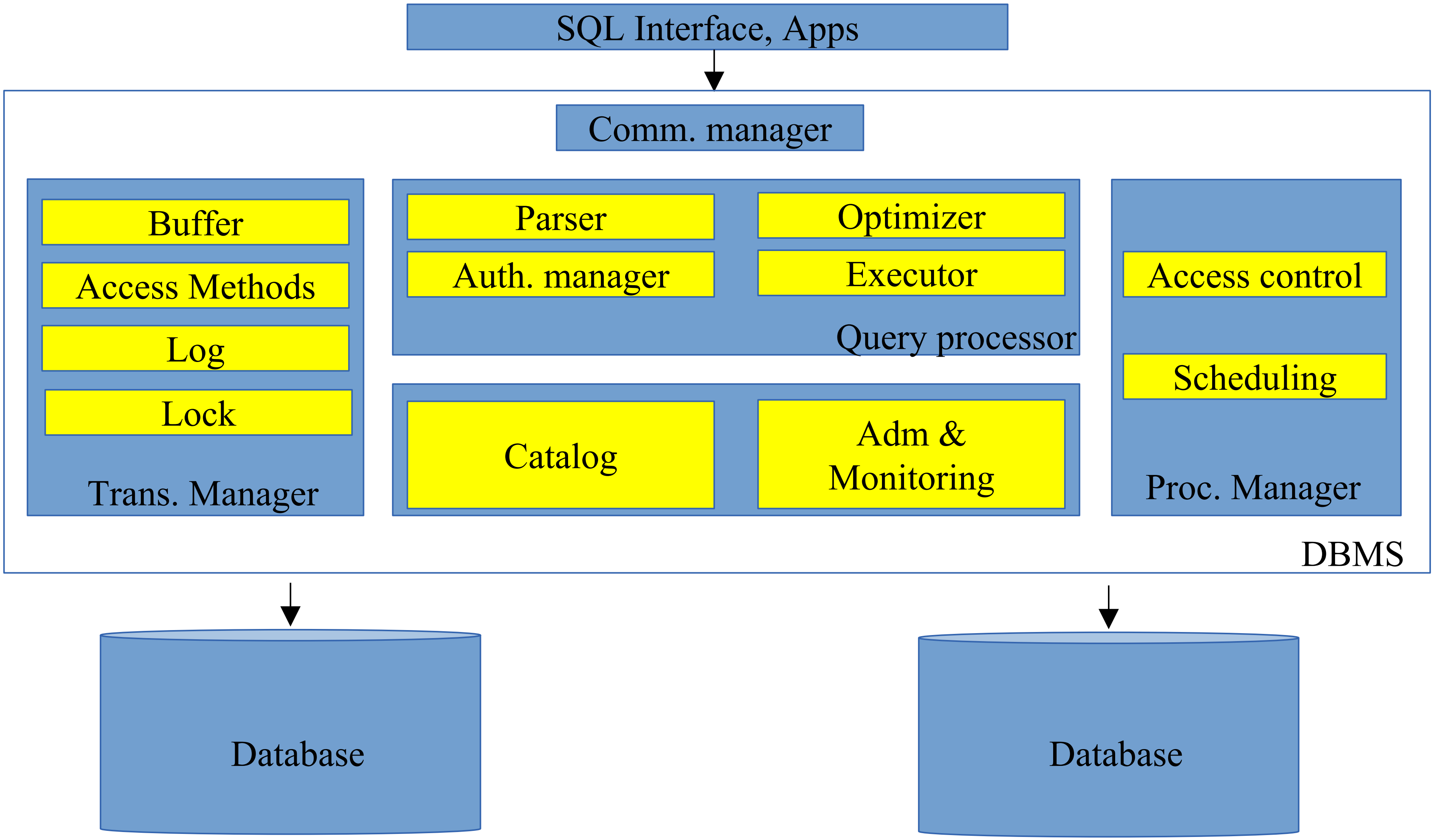
History

- VoltDB (from H-Store)
- Facebook Presto
- Google F1
- Google Megastore

2010's

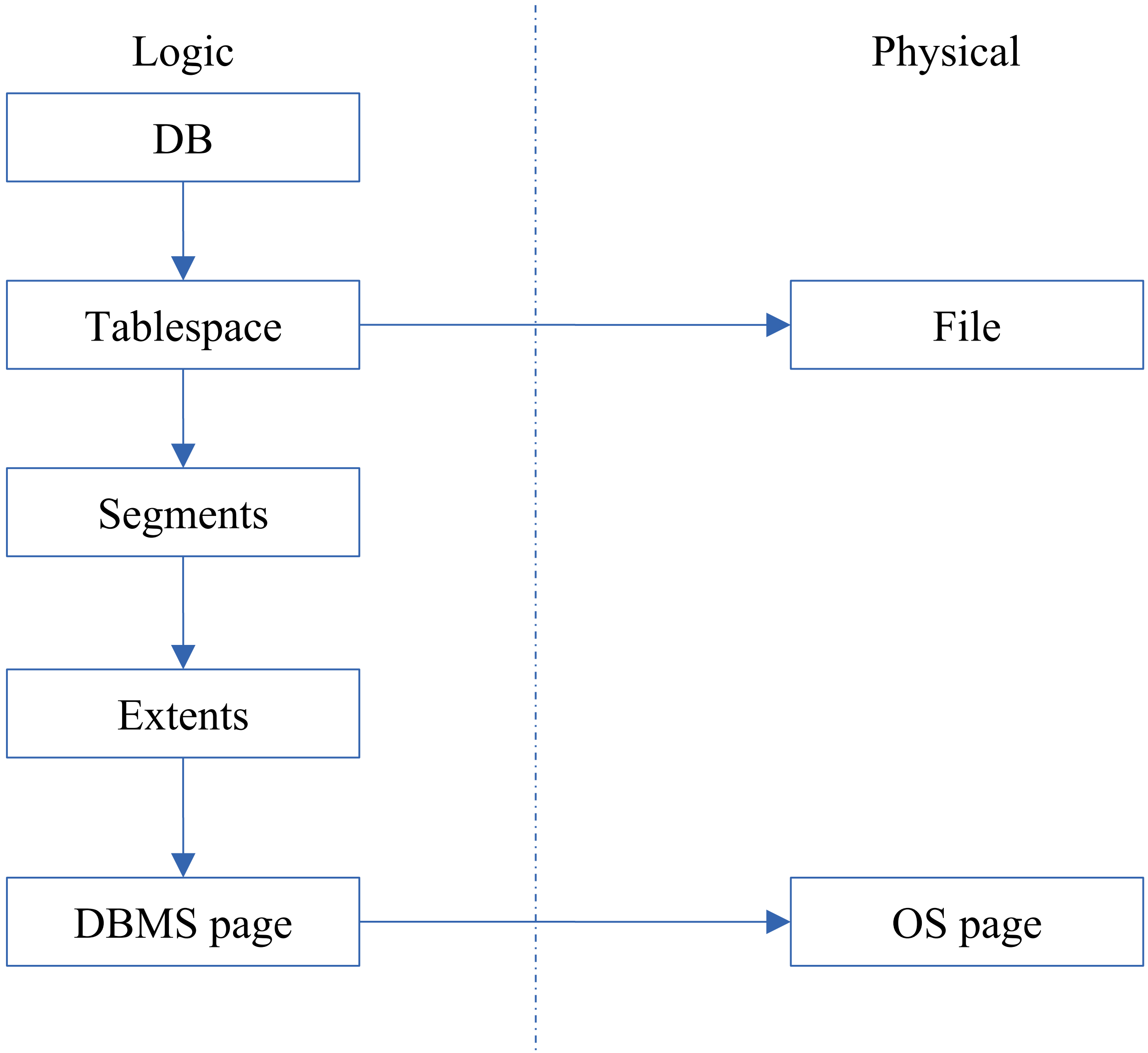
NewSQL and NoSQL architectures !

General DBMS Architecture

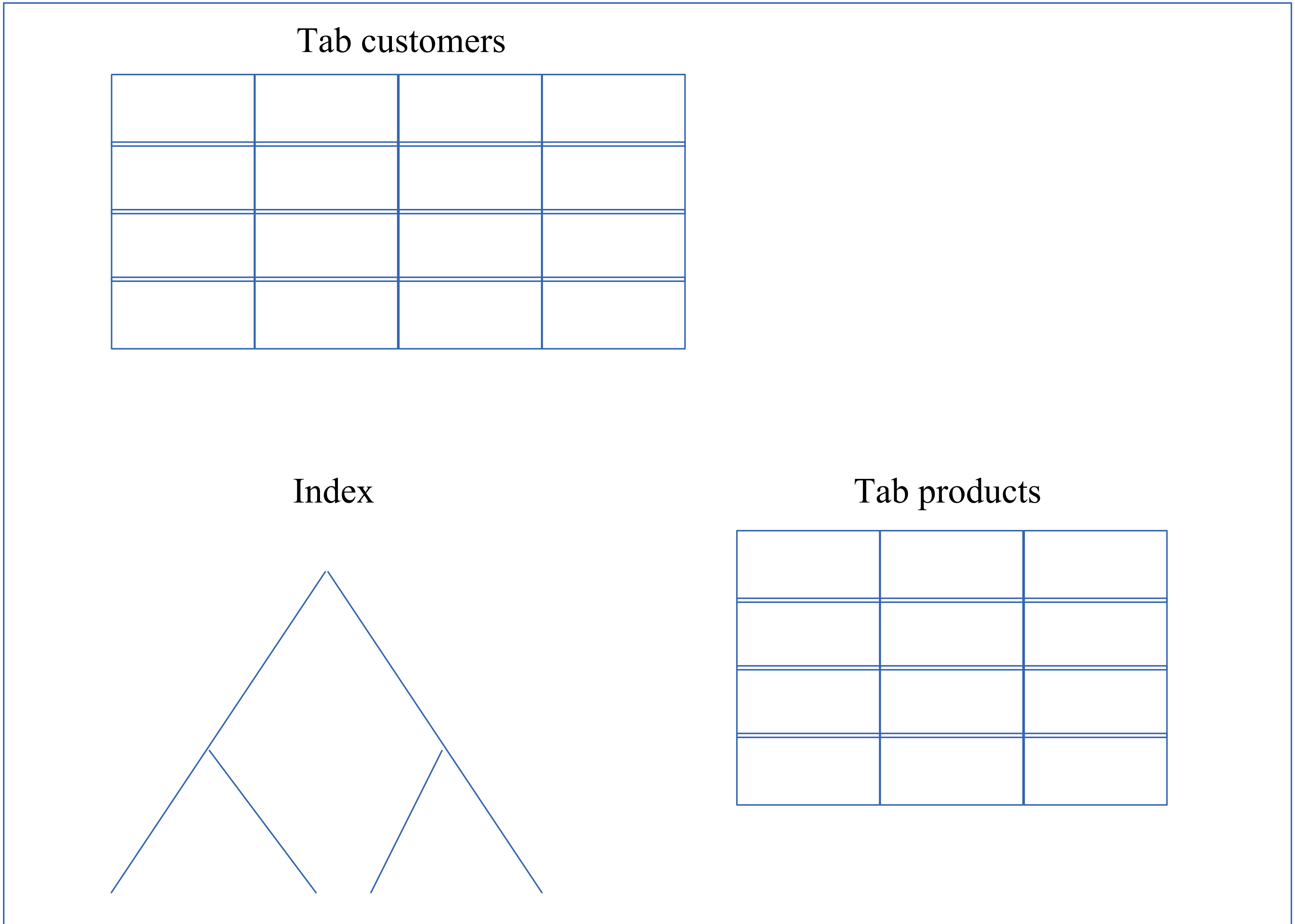


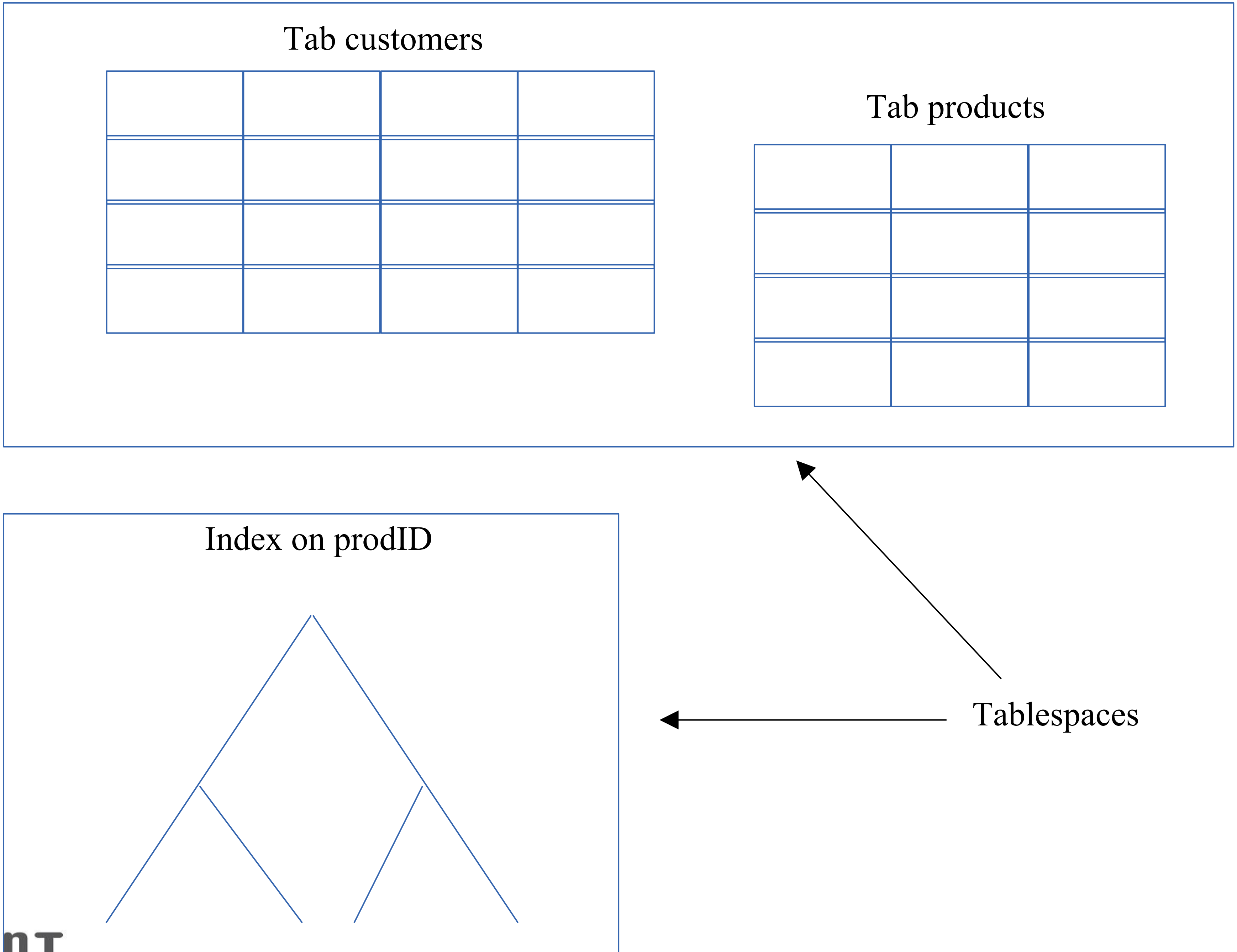
Database Layout

Database Layout

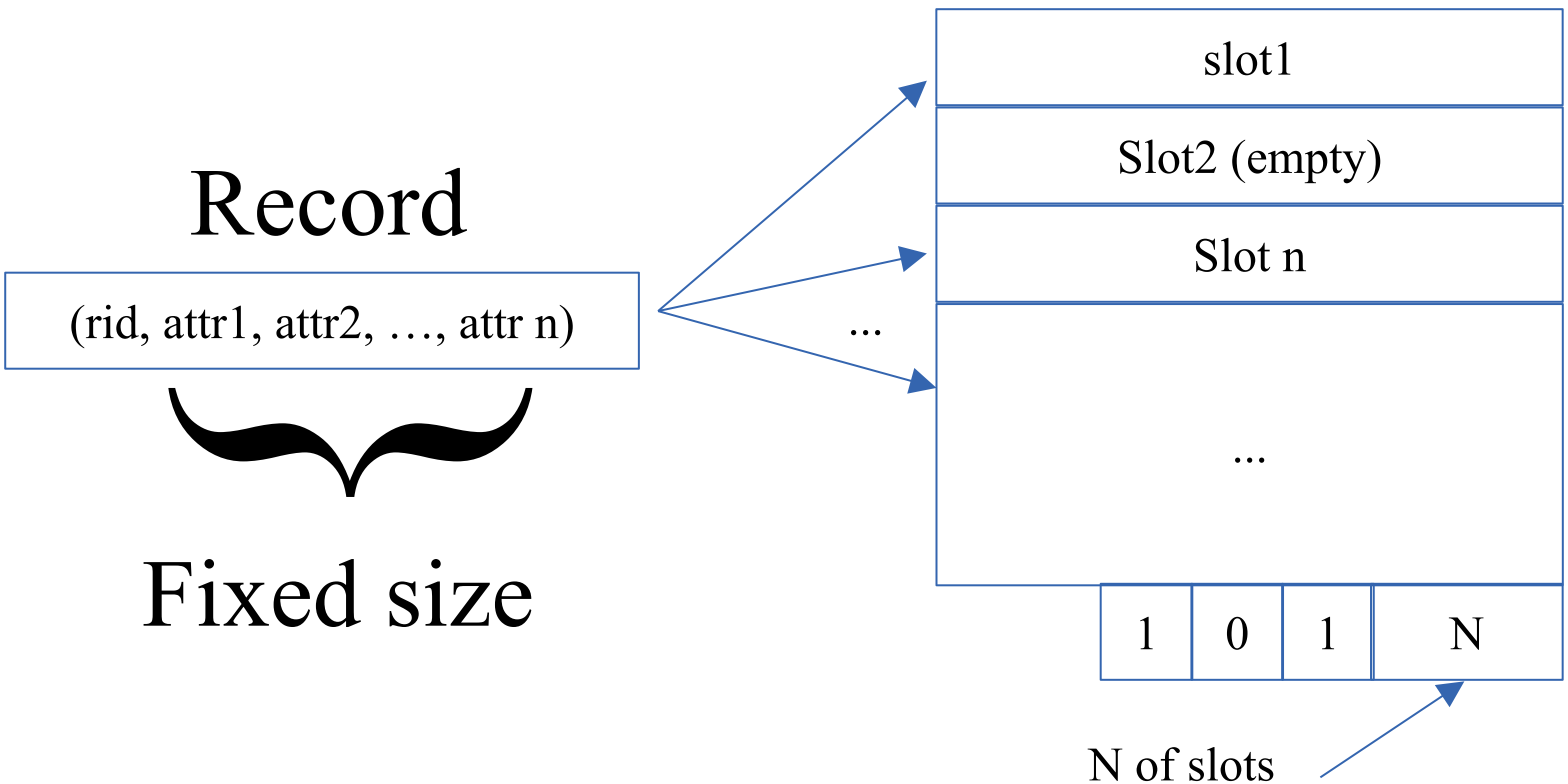


securityandtrust.lu
UNIVERSITÉ DU LUXEMBOURG

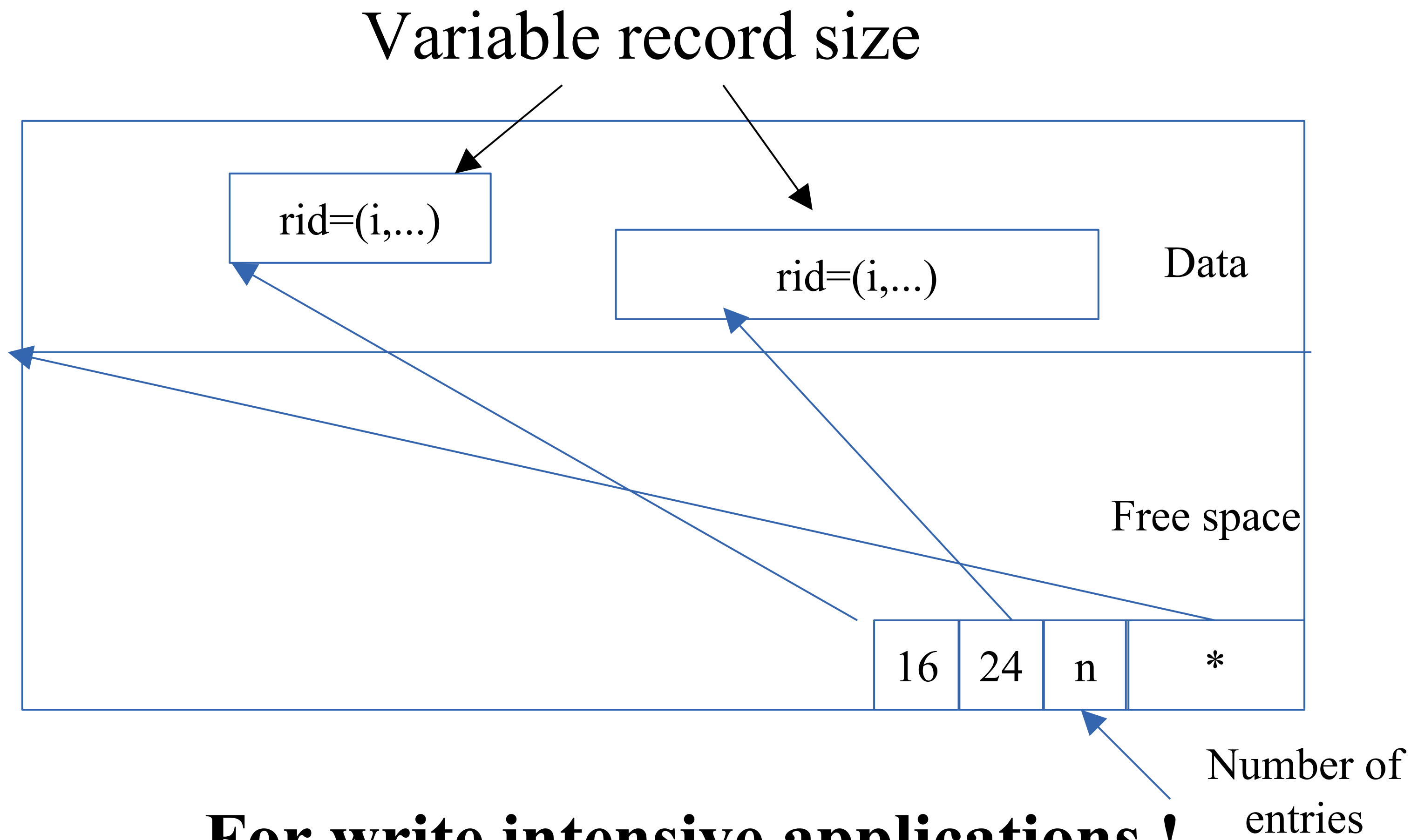

Tab customers
Index
Tab products


Tab customers
Tab products
Index on prodID
Tablespaces

Fixed size pages



Variable size pages
N-ary Storage Model (NSM)



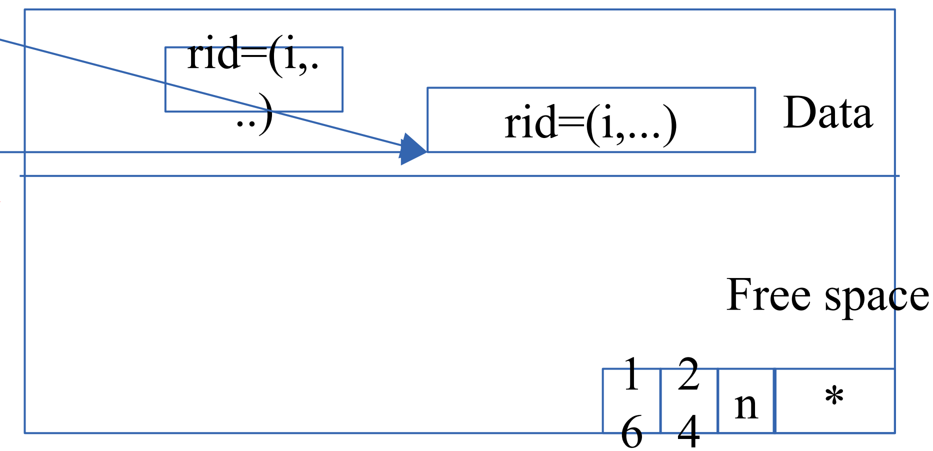
For write intensive applications !

NSM Example

INSERT INTO T
VALUES(2, 'MARIA', 18, CURITIBA, F);

SELECT *
FROM CUSTOMER
WHERE ID=2;

SELECT AVG(AGE)
FROM CUSTOMER;

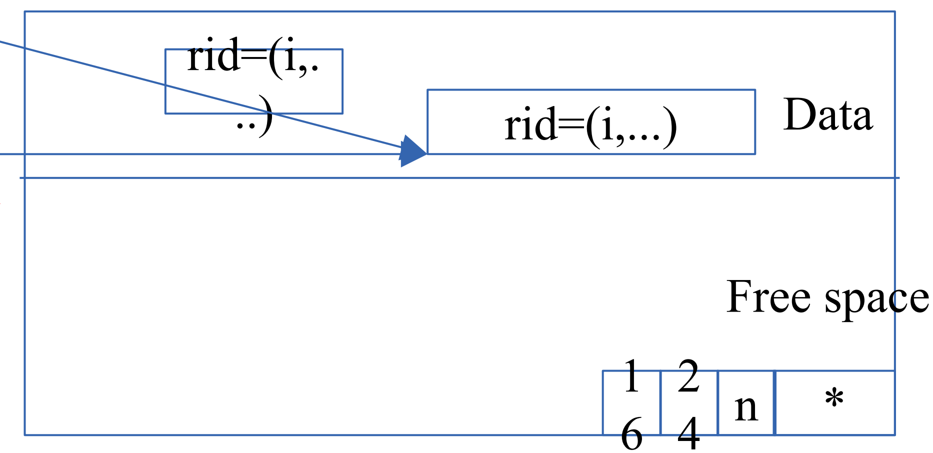


NSM Example

INSERT INTO T
VALUES(2, 'MARIA', 18, CURITIBA, F);

SELECT *
FROM CUSTOMER
WHERE ID=2;

SELECT AVG(AGE)
FROM CUSTOMER;



- Few write operations
- Read the entire tuple at once
- Aggregations may read the entire DB!!

Columnar page

Decomposition Storage Model (DSM) [Ailamaki, VLDB02]

Page: Attr1	Page: Attr2	Page: Attr3
(rid1,attr1)	(rid1,attr2)	(rid1,attr3)
(rid2,attr1)	(rid2,attr2)	(rid2,attr3)
…	…	…
(rid_n,attr1)	(rid_n,attr2)	(rid_n,attr3)

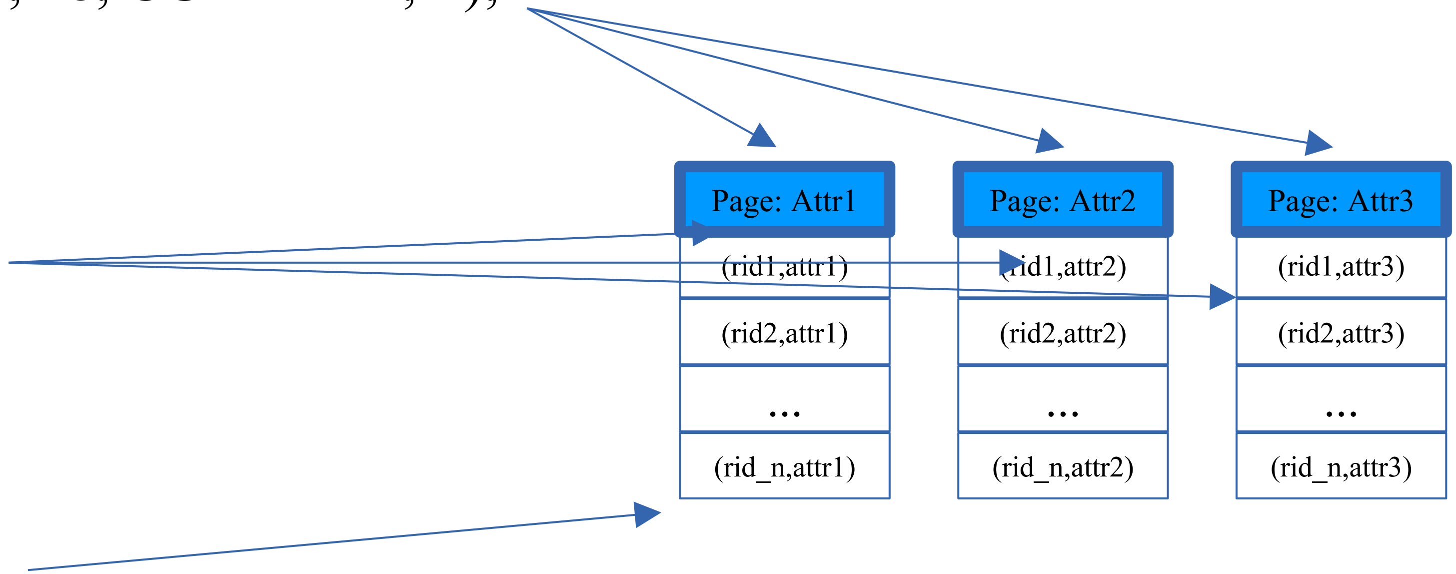
Improves aggregation and compression

For read intensive applications !

DSM Example

INSERT INTO T
VALUES(2, 'MARIA', 18, CURITIBA, F);

SELECT *
FROM CUSTOMER
WHERE ID=2;

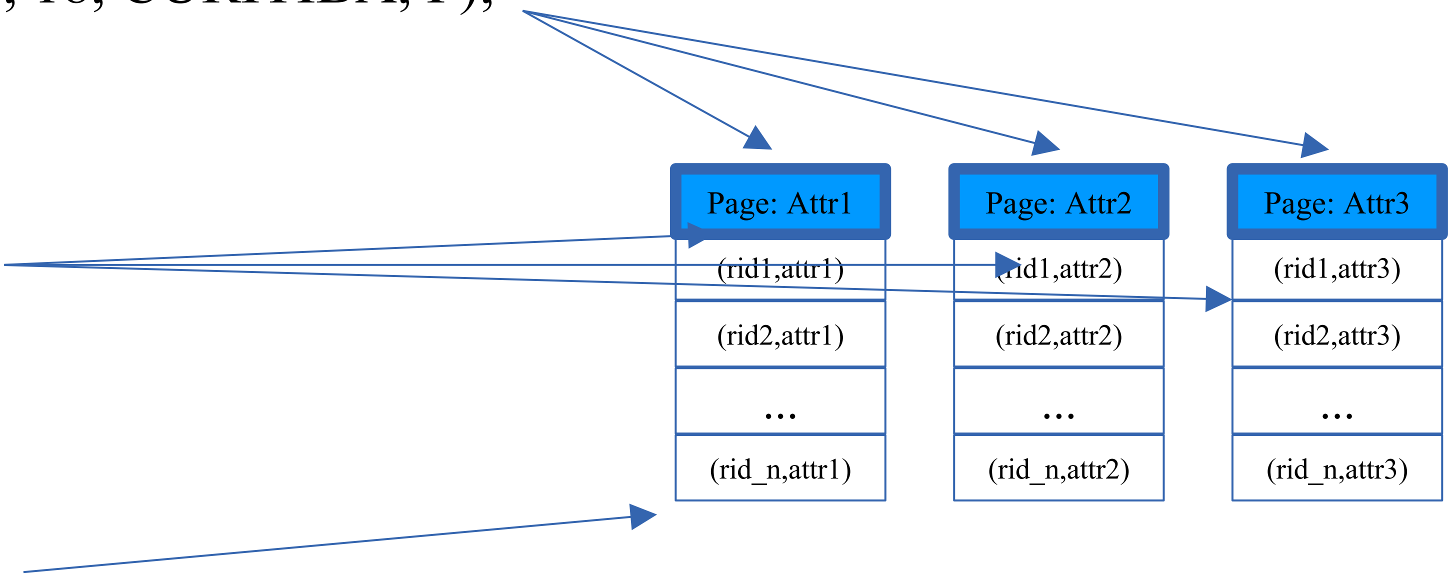


SELECT AVG(AGE)
FROM CUSTOMER;

DSM Example

INSERT INTO T
VALUES(2, 'MARIA', 18, CURITIBA, F);

SELECT *
FROM CUSTOMER
WHERE ID=2;

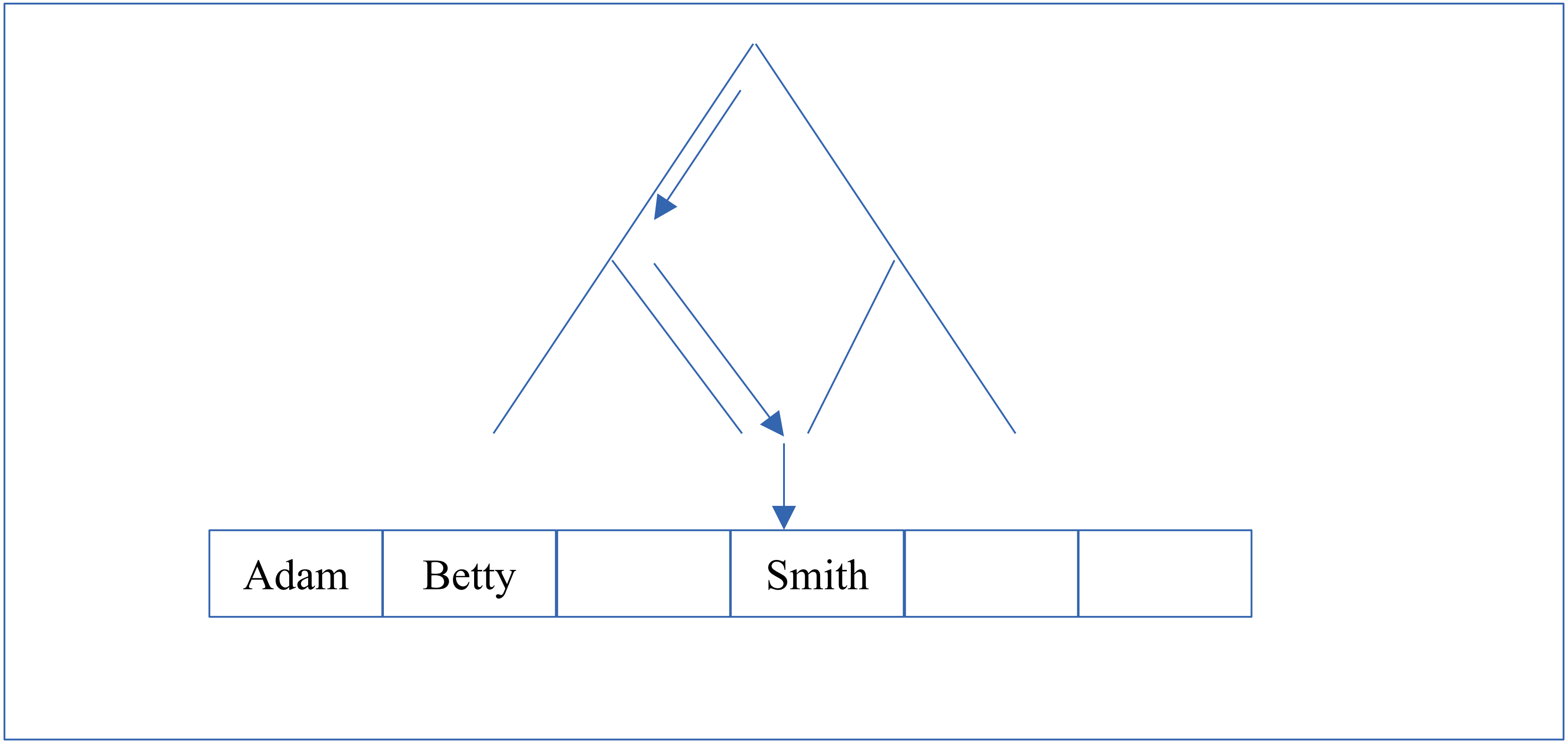


SELECT AVG(AGE)
FROM CUSTOMER;

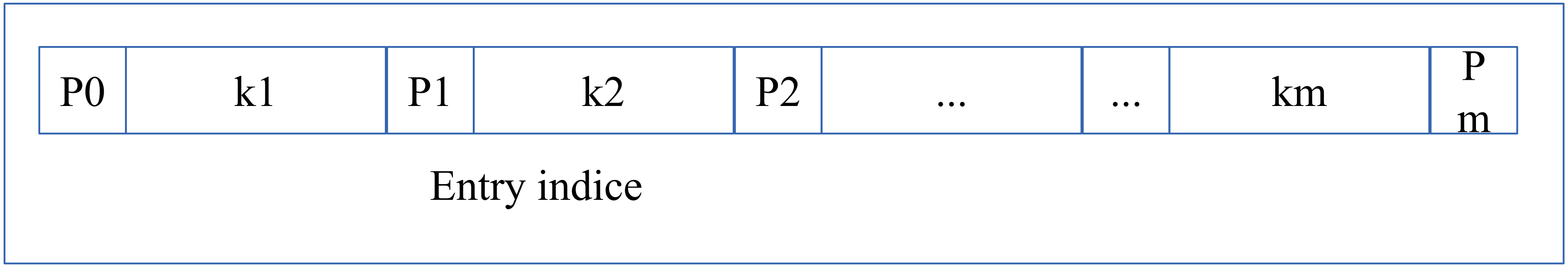
- More write operations than NSM
- Cannot read an entire tuple at once
- Aggregations are fast, do not read the entire DB!!

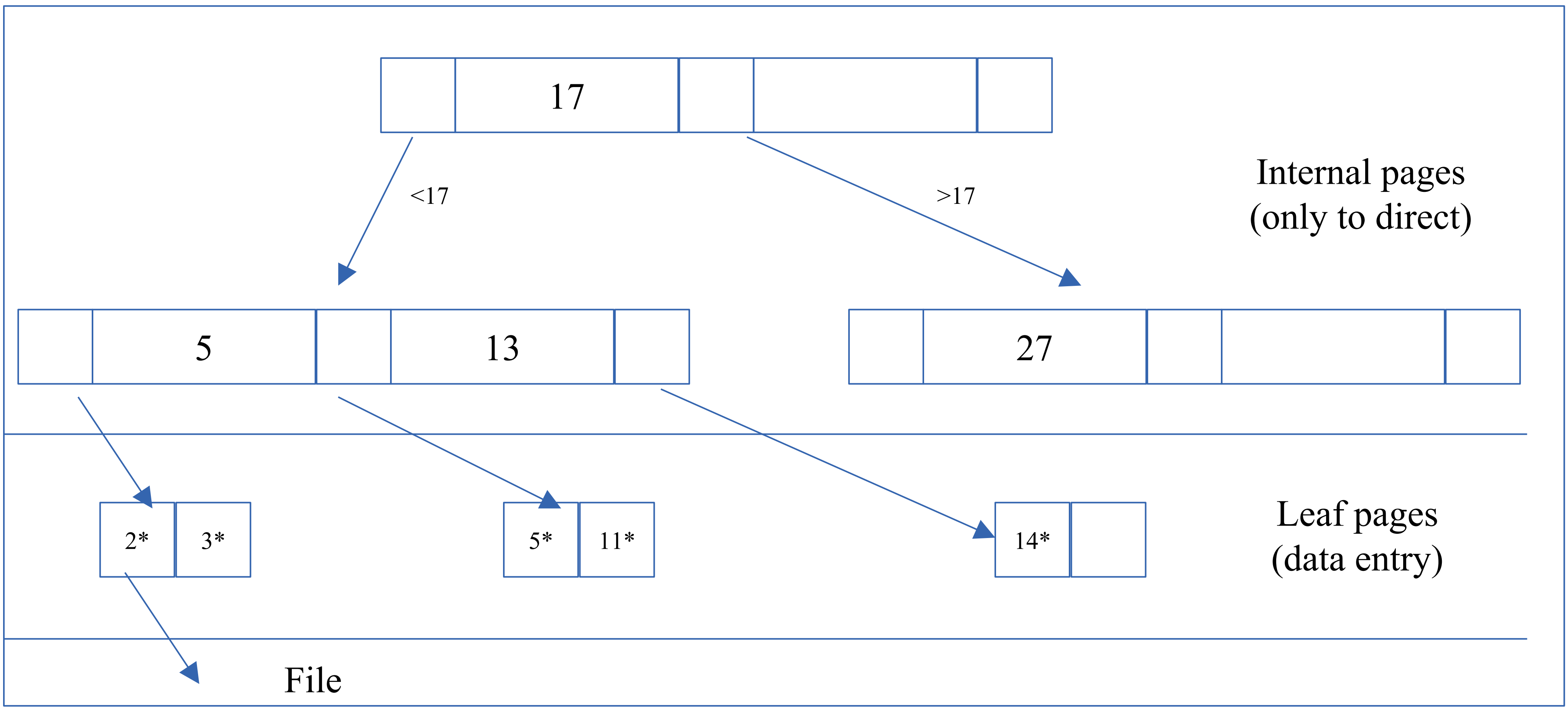
Database Index

Access method: B-tree Index

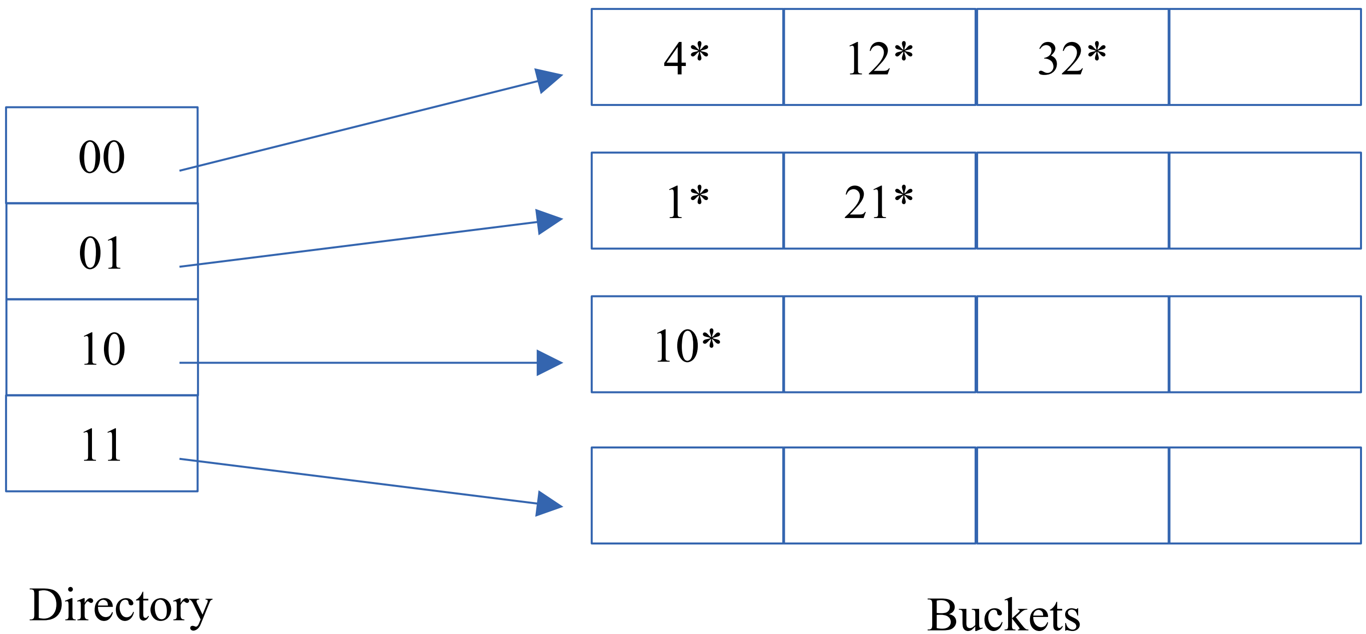


securityandtrust.lu
UNIVERSITÉ DU LUXEMBOURG


P0
k1
P1
k2
P2
...
...
km
P
m
Entry indice

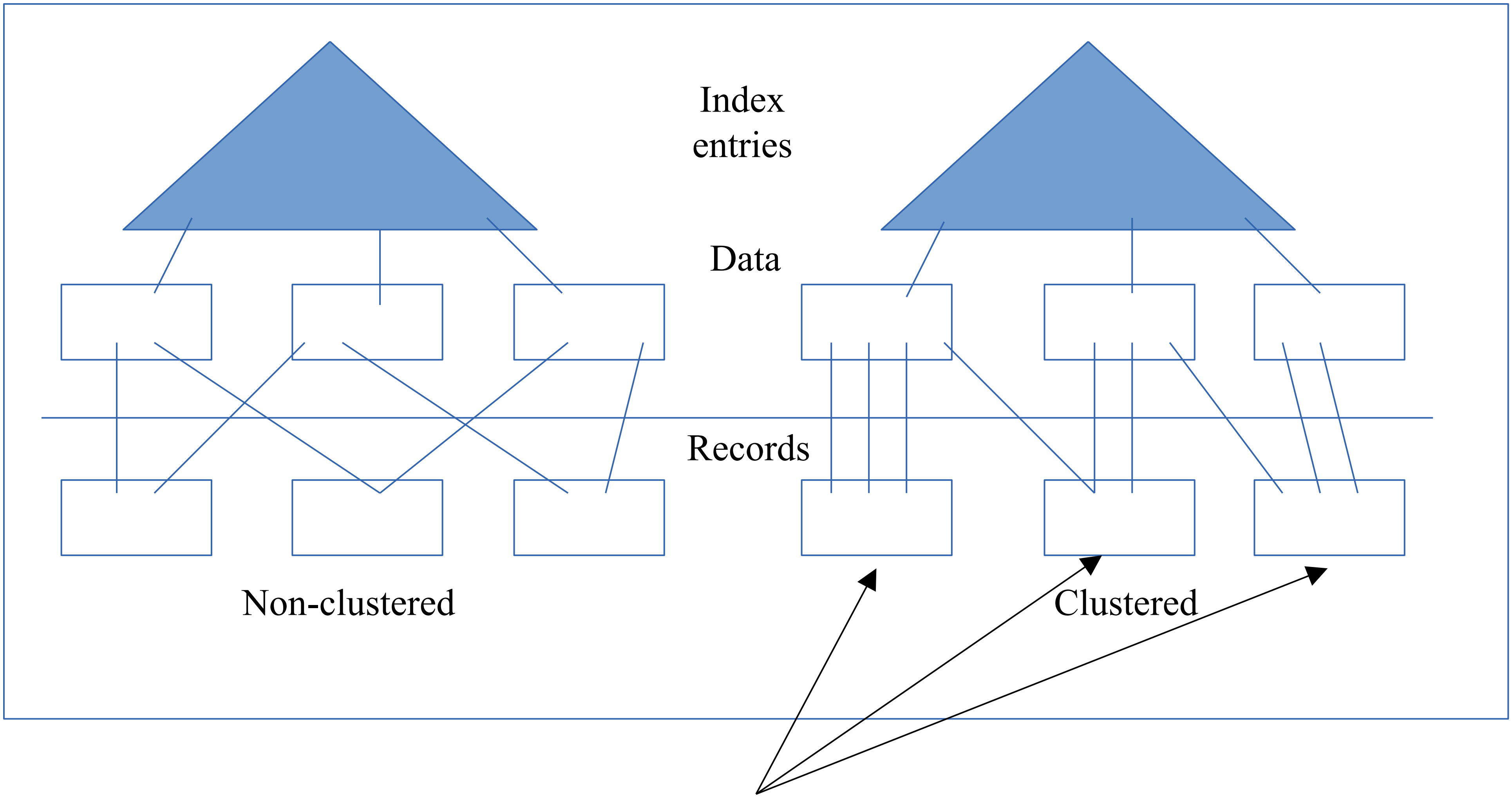

17
<17
>17
Internal pages
(only to direct)
5
13
27
2*
3*
5*
11*
14*
Leaf pages
(data entry)
File

Access method: Hash Index



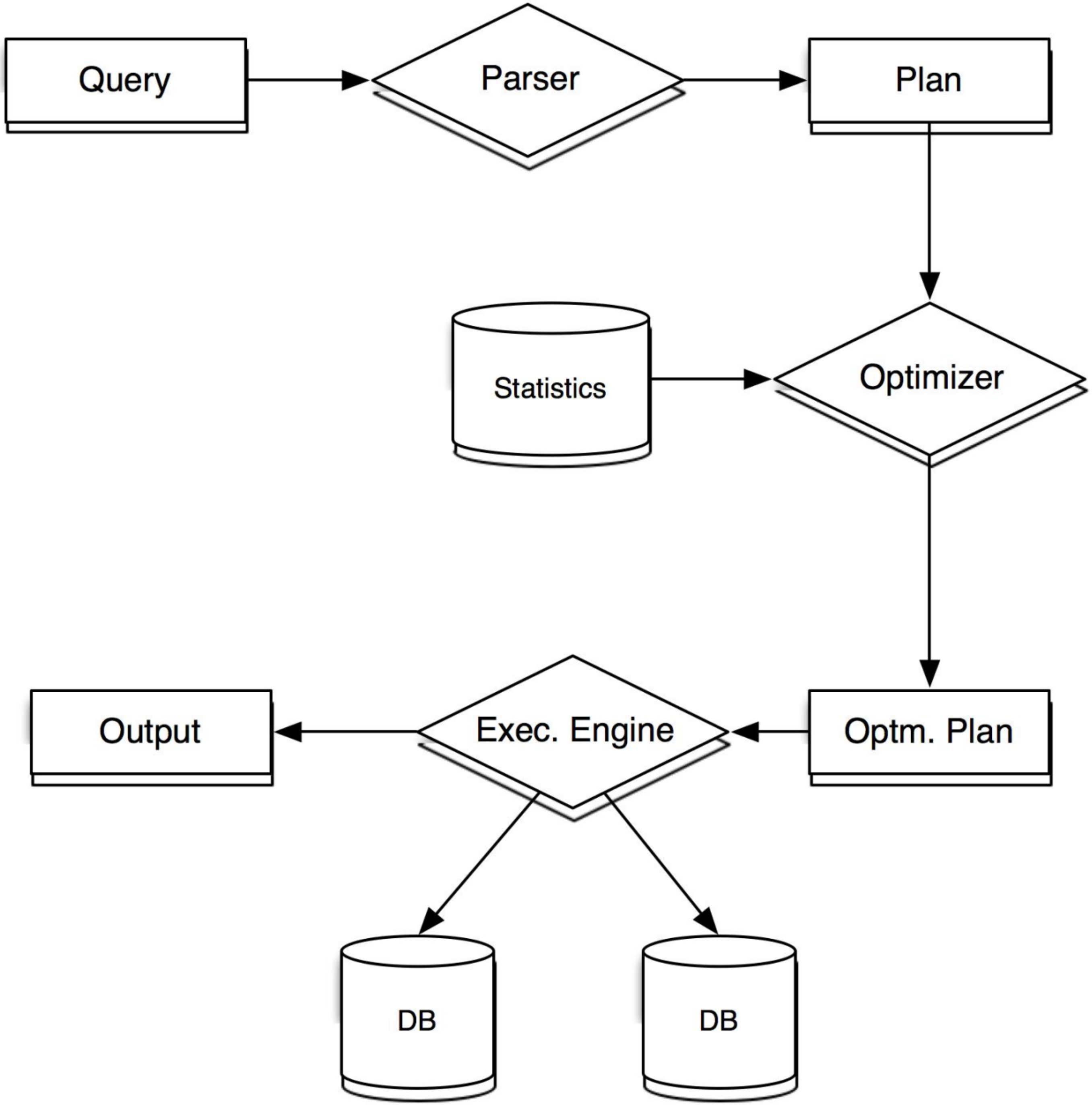
If insert 5*, then 5=101, bucket '01' (read backwards)

Clustered VS Non-clustered index



Physically sorted: only one clustered index per table

Query Processing



(T.F. Bissyandé & M. Hurier)

Query processing

Query

Query expression

SELECT BALANCE
FROM ACCOUNT
WHERE BALANCE > 2500;

$\prod$ BALANCE (σ BALANCE > 2500 (ACCOUNT))

σ BALANCE > 2500

$\prod$ BALANCE (ACCOUNT)

Query processing

Query

SELECT BALANCE
FROM ACCOUNT
WHERE BALANCE > 2500;

Query expression

$\prod$ BALANCE (σ BALANCE > 2500 (ACCOUNT))

σ BALANCE > 2500

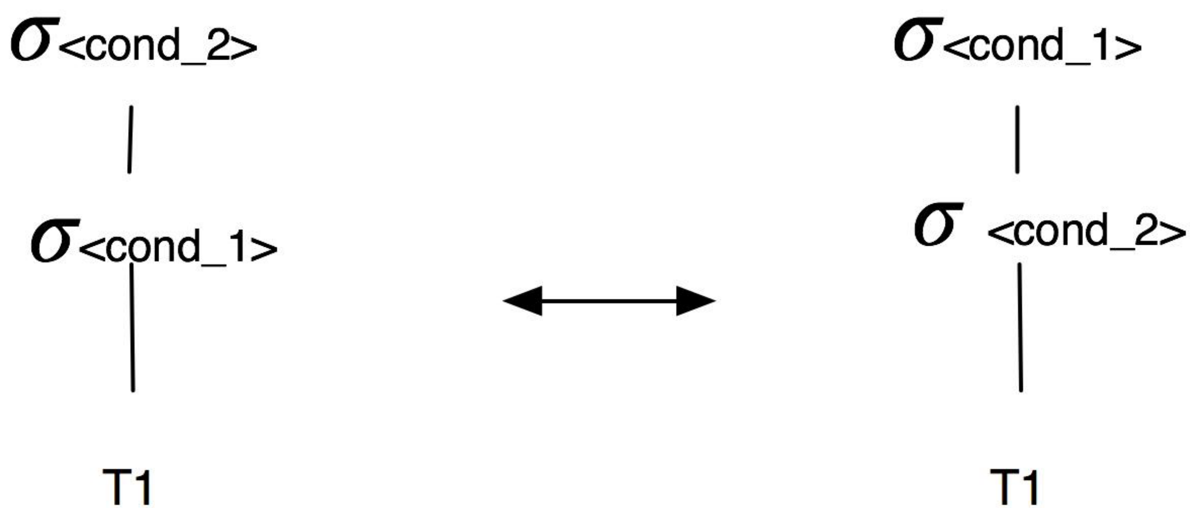
$\prod$ BALANCE (ACCOUNT)

May have n plans!!!

How does the DBMS rewrite a query?

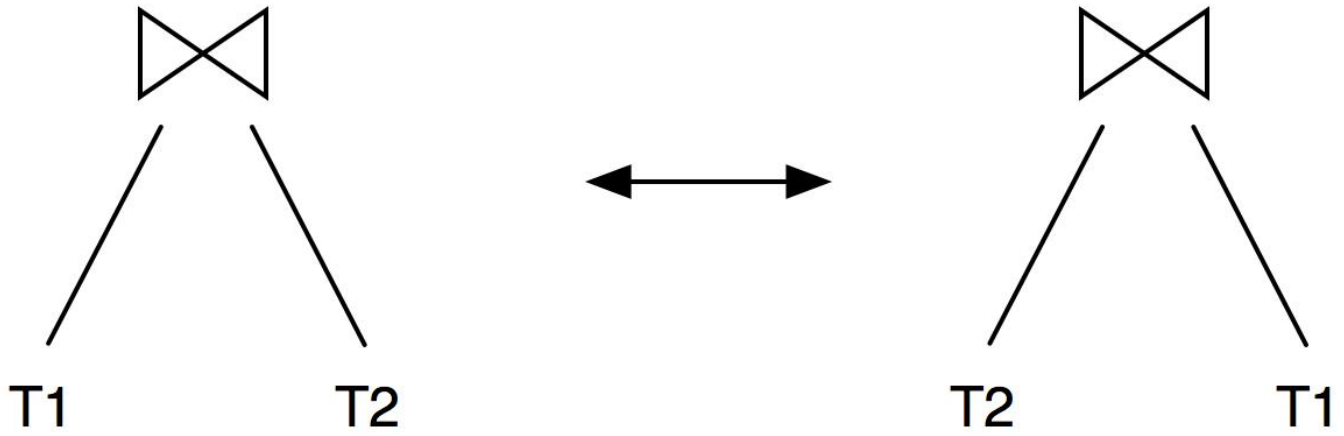
Query Optimization

Query rewrite



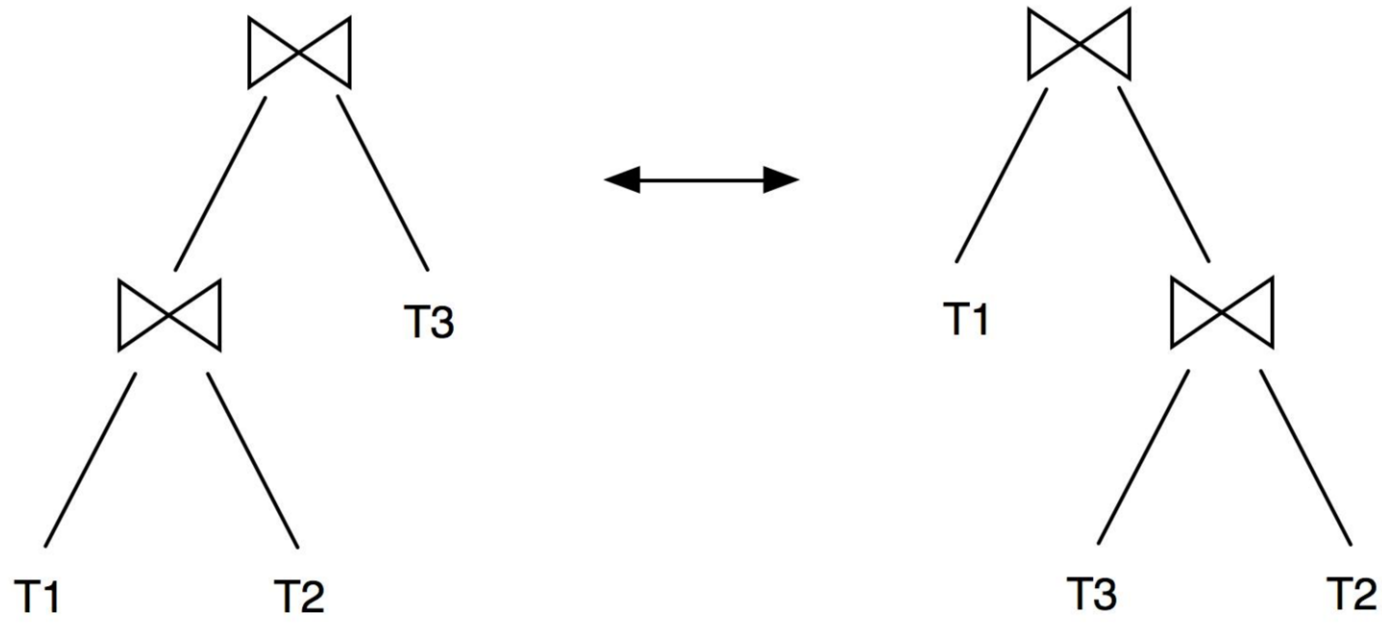
Rule 1: Selection operations are commutative

Query rewrite



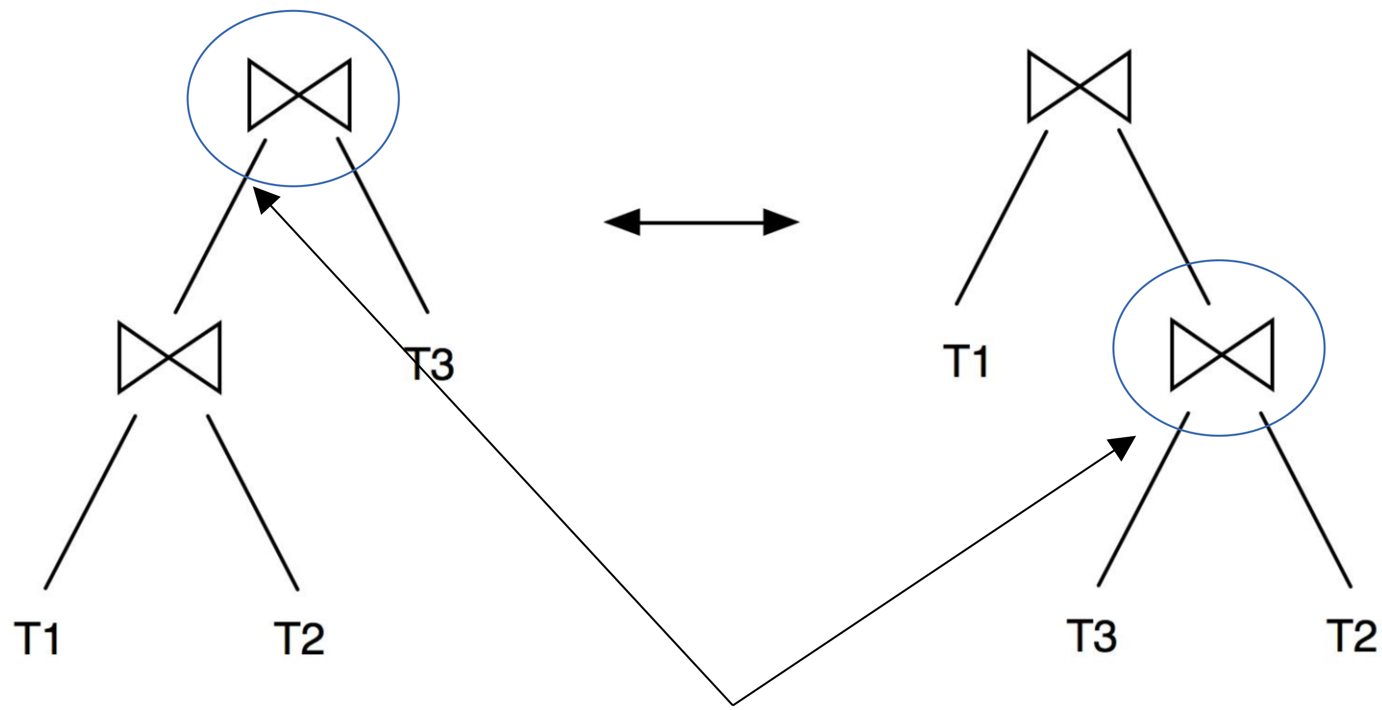
Rule 2: Join operations are commutative
if the order of the attributes is not taken into account

Query rewrite



Rule 3: Natural-joins are associative

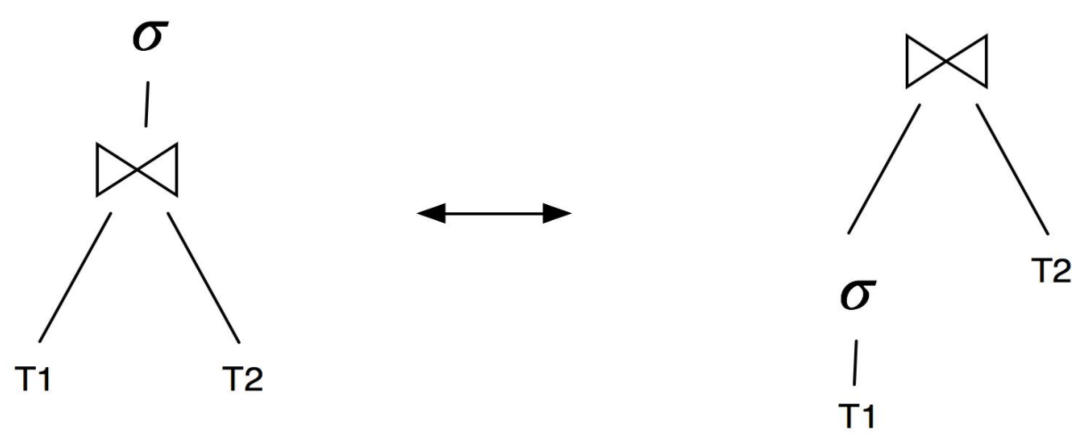
Query rewrite



Rule 4: Natural-joins are associative.

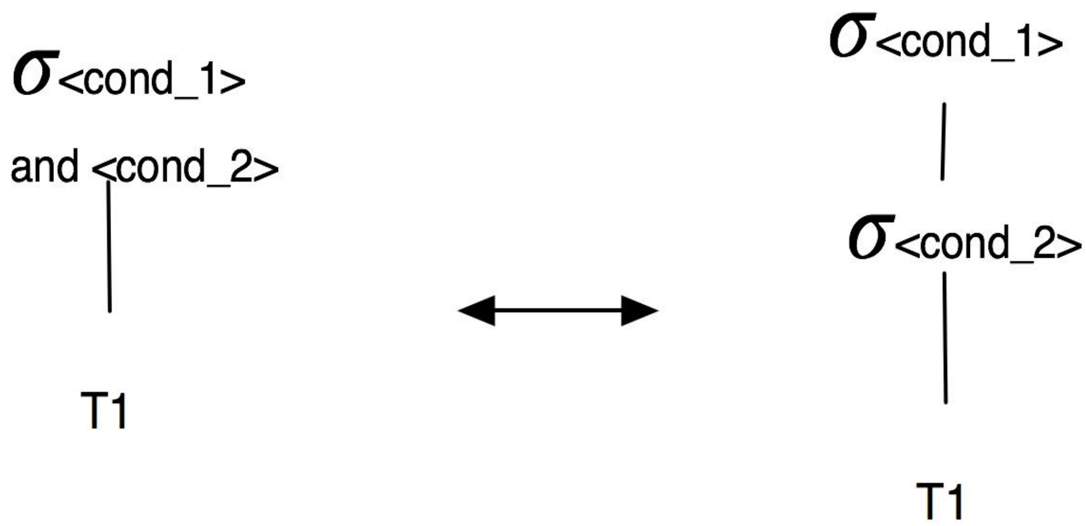
For theta-joins, this only holds
if the following join involves attributes only from T2 and T3.

Query rewrite



Rule 5: Join distributes when all the attributes in the selection condition involve only the attributes of one of the expressions

Query rewrite



Rule 6: Conjunctive selection operations can be split into a sequence of individual selections

Further rules can be found in the readings

Database Catalog

The catalog is a meta-database that stores informations about:

DB and DBMS:
Data structures, Buffer and Page sizes, Indices, Views

Statistics:
Cardinality of tables and indices, Domain values
Page sizes for tables and indices

Database Catalog

ALL_TABLES view from the Oracle catalog

A.1.10 ALL_TABLES

This view provides the following information about tables accessible to the user. The parameters for this view are listed in Table A-10:

Table A-10 ALL_TABLES Parameter

Column	Datatype	NULL ALLOWED	Description
OWNER	VARCHAR2(128)	No	User name of the owner of the table.
TABLE_NAME	VARCHAR2(128)	No	Name of the table.
TABLESPACE_NAME	VARCHAR2(128)	Yes	Name of the catalog or database file containing the table.
CLUSTER_NAME*	VARCHAR2(128)	Yes	Name of the cluster, if any, to which the table belongs.
PCT_FREE*	NUMBER(10)	Yes	Minimum percentage of free space in a block.
PCT_USED*	NUMBER(10)	Yes	Minimum percentage of used space in a block.
INI_TRANS*	NUMBER(10)	Yes	Initial number of transactions.
MAX_TRANS*	NUMBER(10)	Yes	Maximum number of transactions.
INITIAL_EXTENT*	NUMBER(10)	Yes	Size of the initial extent in bytes.
NEXT_EXTENT*	NUMBER(10)	Yes	Size of secondary extents in bytes.
MIN_EXTENTS*	NUMBER(10)	Yes	Minimum number of extents allowed in the segment.
MAX_EXTENTS*	NUMBER(10)	Yes	Maximum number of extents allowed in the segment.
PCT_INCREASE*	NUMBER(10)	Yes	Percentage increase in extent size.
BACKED_UP*	VARCHAR2(1)	Yes	If the table was backed up since last change.
NUM_ROWS*	NUMBER(10)	Yes	Number of rows in the table.
BLOCKS*	NUMBER(10)	Yes	Number of data blocks allocated to the table.
EMPTY_BLOCKS*	NUMBER(10)	Yes	Number of data blocks allocated to the table that contain no data.
AVG_SPACE*	NUMBER(10)	Yes	Average amount of free space (in bytes) in a data block allocated to the table.
CHAIN_CNT*	NUMBER(10)	Yes	Number of rows in the table that are chained from one data block to another, or that have migrated to a new block, requiring a link to preserve the old ROWID.
AVG_ROW_LEN*	NUMBER(10)	Yes	Average length of a row in the table in bytes.

Operators (I/O cost for Selection)

- **Scan on unsorted data**:
  - O(M), where M = number of pages
- **Scan on sorted data:**
  - O(log M), where M = number of pages
- **Index scan (clustered index)**:
  - O(h+1), where h = height of the index tree
- **Index scan (non-clustered index)**:
  - O(h+n), where n = number of fetched tuples
  - n = M if the tuples are spread across all pages
    - (worst case)

Transactions

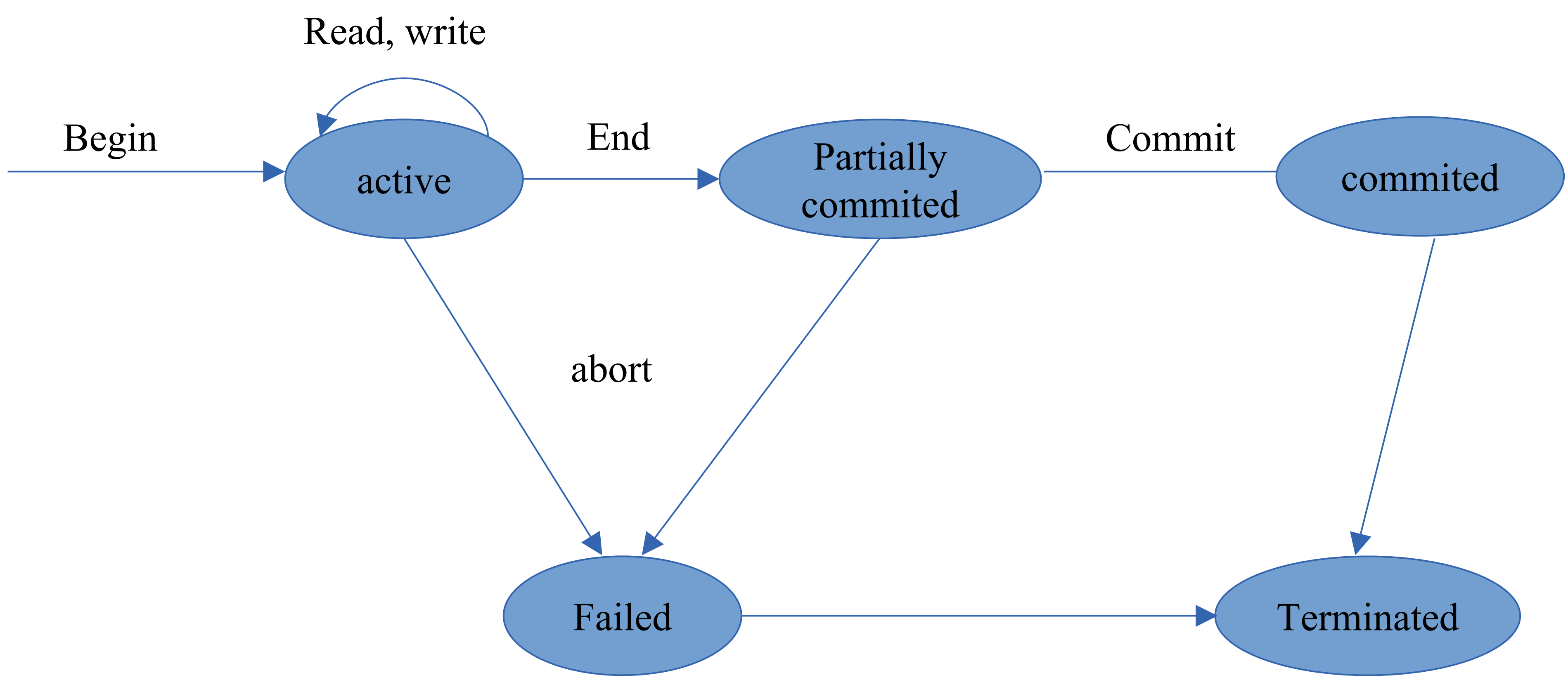
Transactions

A transaction is an atomic unit of work that is either completed in its entirety or not done at all
[Navathe and Elmasri, 2005]

T1	Database
r(balance)	balance = 200
balance += 100	r(x)
w(balance)	balance = 300

Transaction State Transition

[Elmasri and Navathe, 2005]



ACID Properties

ACID = Atomicity, Consistency, Isolation, Durability

Atomicity: if one part of the transaction fails, the entire transaction fails, and the database state is left unchanged

Consistency: ensures that any transaction will bring the database from one valid state to another (constraint rules)

Isolation: the concurrent execution of transactions results in a state that would be obtained if they were executed serially

Durability: once a transaction has been committed, it will remain so, even in the event of power loss, crashes, or errors

Schedules

A schedule orders the execution of concurrent transactions in an interleaving fashion

T1	T2
r(x)	
	r(x)
	w(x)
w(x)	

Concurrency Conflicts

Concurrency problems:
dirty read and lost update

T1	T2
r(balance)	
balance += 100	r(x)
w(balance)	w(x)
w(x)	r(balance)
abort	balance-=100
	w(balance)

T1	T2
r(balance)	
	r(balance)
balance += 100	w(x)
w(x)	balance -= 100
w(balance)	w(balance)

Testing Conflict Serializability

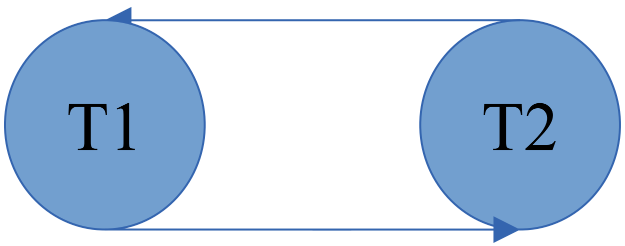
Algorithm:

Create a node for each transaction T in schedule S;
Create an edge Ti -> Tj for each r(x) in Tj after w(x) in Ti
Create an edge Ti -> Tj for each w(x) in Tj after r(x) in Ti
Create an edge Ti -> Tj for each w(x) in Tj after w(x) in Ti
S is serializable if and only if the precedence graph has no cycles

For example:

T1	T2
r(x)	
	r(x)
	w(x)
w(x)	

Not serializable



Readings

- Joseph Hellerstein, Michael Stonebraker and James Hamilton.
- Architecture of a Database System, material at: http://db.cs.berkeley.edu/papers/fntdb07-architecture.pdf
- Hector Garcia-Molina, Jeffrey Ullman, Jennifer Widom. Database Systems: The Complete Book, material at: http://infolab.stanford.edu/~ullman/dscb.html

Thank You !
15 minutes break

Quiz

1. What types of page layout are more likely for the following queries? Why?

1. Q1
   1. SELECT *
   2. FROM CUSTOMERS;

1. Q2
   1. SELECT A.ID, A.BALANCE, C.NAME
   2. FROM CUSTOMERS C, ACCOUNT A
   3. WHERE A.ID=C.ID

1. Q3
   1. SELECT SUM(BALANCE)
   2. FROM ACCOUNT;
   3.

1. Q4

   SELECT C.CITY, SUM(A.BALANCE)
   FROM CUSTOMERS C, ACCOUNT A
   WHERE A.ID=C.ID
   GROUP BY C.CITY;

SNT
securityandtrust.lu

uni.lu
UNIVERSITÉ DU LUXEMBOURG

Quiz

1. Considering the following queries, what types of index need to be implemented? Why?

1. Q1
   1. SELECT NAME, AGE
   2. FROM CUSTOMER
   3. WHERE AGE BETWEEN 18 AND 25;

1. Q2
   1. SELECT NAME, AGE
   2. FROM CUSTOMER
   3. WHERE CUST_ID= 1234;

1. Q3
   1. SELECT NAME, CITY, AGE, SSN
   2. FROM CUSTOMER
   3. WHERE AGE BETWEEN 18 AND 25
   4. AND CITY ;

Quiz

1. Please, rewrite the following expressions using the presented rules and the following relations:

ACCOUNT(BALANCE, ACC_ID, C_ID, B_ID), CUSTOMER(C_ID), BRANCH(B_ID, B_NAME)

1.Q1 $\sigma_{BALANCE > 2500 \text{ AND } C\_ID > 1000}$ (ACCOUNT X CUSTOMER)

1.Q2 $\sigma_{BALANCE > 2500}$ ($\sigma_{B\_ID = 1000}$ (ACCOUNT X BRANCH))

1.Q3 $\prod_{BALANCE, C\_ID}$ ($\sigma_{BALANCE > 2500}$ (ACCOUNT X CUSTOMER))

1.Q4 $\prod_{BALANCE, ACC\_ID}$ (CUSTOMER X ACCOUNT)

1.Q5 $\prod_{BALANCE, ACC\_ID}$ (CUSTOMER X (ACCOUNT X BRANCH))

securityandtrust.lu

Quiz

1.Please, rewrite the following expressions using the presented rules and the following relations:

B_ID),
CUSTOMER(C_ID),
BRANCH(B_ID,
B_NAME)

1.Q1 σ BALANCE > 2500 (σ C_ID > 1000 (ACCOUNT X CUSTOMER))

1.Q2 σ B_ID = 1000 (σ BALANCE > 2500 (ACCOUNT X BRANCH))

1.Q3 σ BALANCE > 2500 (∏ BALANCE, C_ID (ACCOUNT X CUSTOMER))

1.Q4 ∏ BALANCE, ACC_ID (CUSTOMER X ACCOUNT)

1.Q5 ∏ BALANCE, ACC_ID (BRANCH x (CUSTOMER X ACCOUNT))

Quiz

1.Compute of the cost for a query selecting 10% of a table 'R' based on the following information from the catalog:

Relation 'R'	10,000 tuples
Tuples/page ratio	100 tuples

Scan	
Scan on sorted	
Clustered index scan	
Non-clustered index scan	

Quiz

1. Compute of the I/O cost for a query selecting 10% of a table 'R' based on the following information from the catalog:

Relation 'R'	10,000 tuples
Tuples/page ratio	100 tuples

Scan	100 I/O
Scan on sorted	6.6 I/O
Clustered index scan	3 I/O
Non-clustered index scan	103 I/O

Hands-on

Which of the following schedules is conflict serializable?

S1 - T1 : r(x),T3 : r(x),T1 : w(x),T2 : r(x),T3 : w(x)
S2 - T1 : r(x),T2 : r(x),T1 : w(x),T2 : w(x)
S3 - T1 : r(x),T2 : r(y),T3 : w(x),T2 : r(x),T1 : r(x)