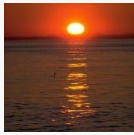
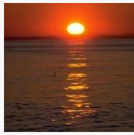
Chapter 29:IBM DB2 Universal Database

Database System Concepts, 6th Ed.



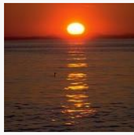
Chapter 29: IBM DB2 Universal Database

- 29.1 Overview
- 29.2 Database-Design Tools
- 29.3 SQL Variations and Extensions
- 29.4 Storage and Indexing
- 29.5 Multidimensional Clustering
- 29.6 Query Processing and Optimization
- 29.7 Materialized Query Tables
- 29.8 Autonomic Features in DB2
- 29.9 Tools and Utilities
- 29.10 Concurrency Control and Recovery
- 29.11 System Architecture
- 29.12 Replication, Distribution, and External Data
- 29.13 Business Intelligence Features
- Bibliographical Notes

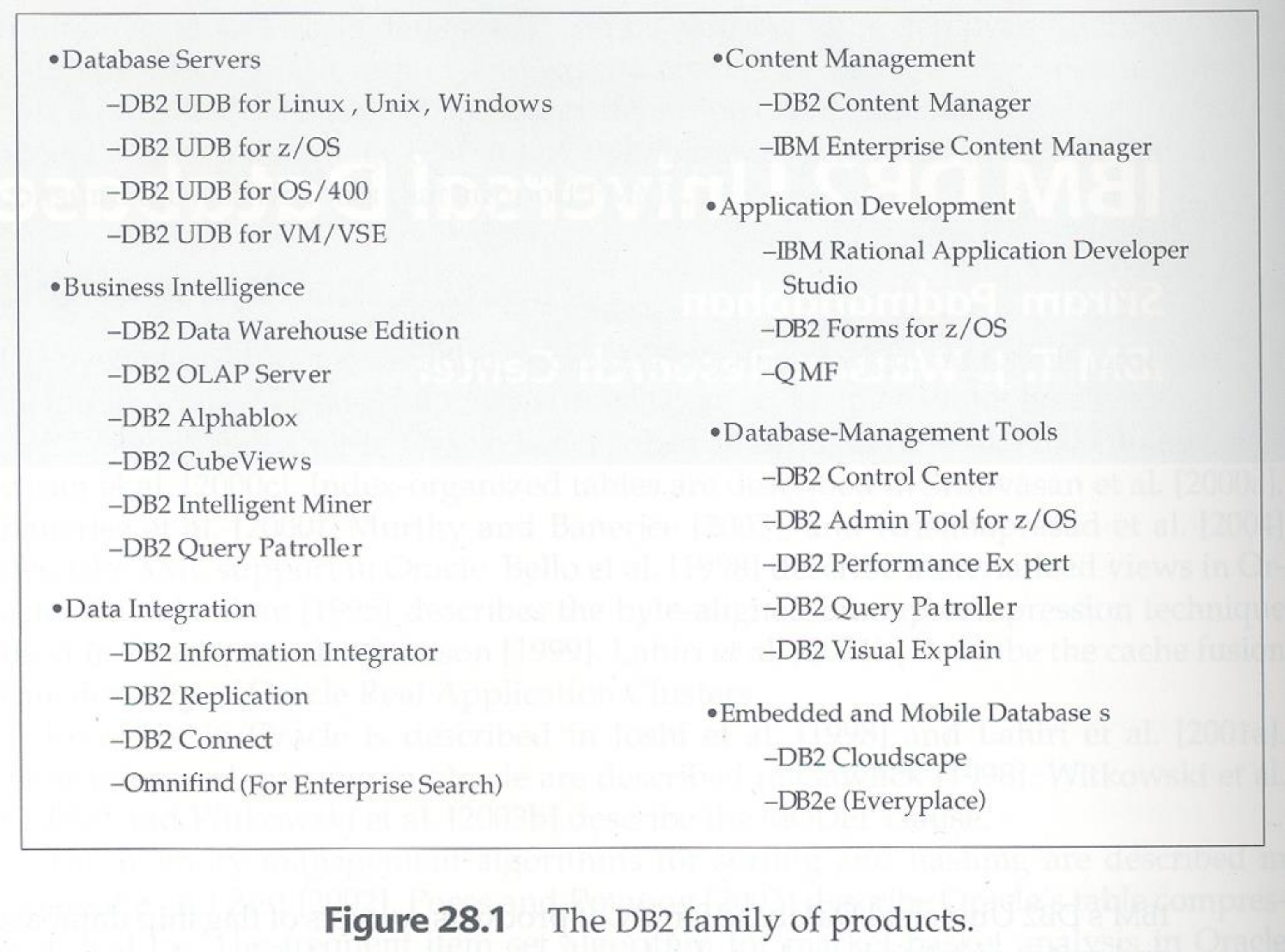


DB2 Overview

- The origin of DB2
  - The System R project at IBM’s Almaden Research Center in 1976
- The first DB2 product was released in 1984
- IBM continually enhanced the DB2 product in areas
  - Transaction processing
  - Query processing and optimization
  - Parallel processing
  - Active database support
  - Advanced query and warehousing techniques
  - Object-relational support
- DB2 database engine consists of four code base types
  - Linux, Unix, and Windows / zOS / VM / OS400
- In this chapter, the focus is on the DB2 Universal Database (UDB) engine that supports Linux, Unix, and Windows
- The latest version of DB2 is version 8.2, which improves the scalability, availability, and general robustness of the DB2 engine



DB2 Product Family

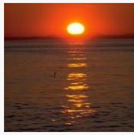


- Database Servers
  - –DB2 UDB for Linux , Unix , Windows
  - –DB2 UDB for z/OS
  - –DB2 UDB for OS/400
  - –DB2 UDB for VM/VSE
- Business Intelligence
  - –DB2 Data Warehouse Edition
  - –DB2 OLAP Server
  - –DB2 Alphablox
  - –DB2 CubeViews
  - –DB2 Intelligent Miner
  - –DB2 Query Patroller
- Data Integration
  - –DB2 Information Integrator
  - –DB2 Replication
  - –DB2 Connect
  - –Omnifind (For Enterprise Search)
- Content Management
  - –DB2 Content Manager
  - –IBM Enterprise Content Manager
- Application Development
  - –IBM Rational Application Developer Studio
  - –DB2 Forms for z/OS
  - –QMF
- Database-Management Tools
  - –DB2 Control Center
  - –DB2 Admin Tool for z/OS
  - –DB2 Performance Expert
  - –DB2 Query Patroller
  - –DB2 Visual Explain
- Embedded and Mobile Databases
  - –DB2 Cloudscape
  - –DB2e (Everyplace)

Figure 28.1 The DB2 family of products.

Chapter 29: IBM DB2 Universal Database

- 29.1 Overview
- 29.2 Database-Design Tools
- 29.3 SQL Variations and Extensions
- 29.4 Storage and Indexing
- 29.5 Multidimensional Clustering
- 29.6 Query Processing and Optimization
- 29.7 Materialized Query Tables
- 29.8 Autonomic Features in DB2
- 29.9 Tools and Utilities
- 29.10 Concurrency Control and Recovery
- 29.11 System Architecture
- 29.12 Replication, Distribution, and External Data
- 29.13 Business Intelligence Features
- Bibliographical Notes



Database-Design Tools

- DB2 provides support for many logical and physical database features using SQL
- The feature includes constraints, triggers, and recursion using SQL constructs
- Physical database features such as tablespaces, bufferpools, and partitioning are also supported by using SQL statements
- DB2 Control Center
  - Includes various design- and administration-related tools
  - Provides a tree-view of a server and its database
  - Allows users to define new objects, create ad-hoc SQL queries, and view query results

Chapter 29: IBM DB2 Universal Database

- 29.1 Overview
- 29.2 Database-Design Tools
- 29.3 SQL Variations and Extensions
- 29.4 Storage and Indexing
- 29.5 Multidimensional Clustering
- 29.6 Query Processing and Optimization
- 29.7 Materialized Query Tables
- 29.8 Autonomic Features in DB2
- 29.9 Tools and Utilities
- 29.10 Concurrency Control and Recovery
- 29.11 System Architecture
- 29.12 Replication, Distribution, and External Data
- 29.13 Business Intelligence Features
- Bibliographical Notes

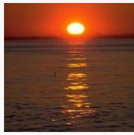
SQL Variations and Extensions

DB2 supports a rich set of SQL features for database processing

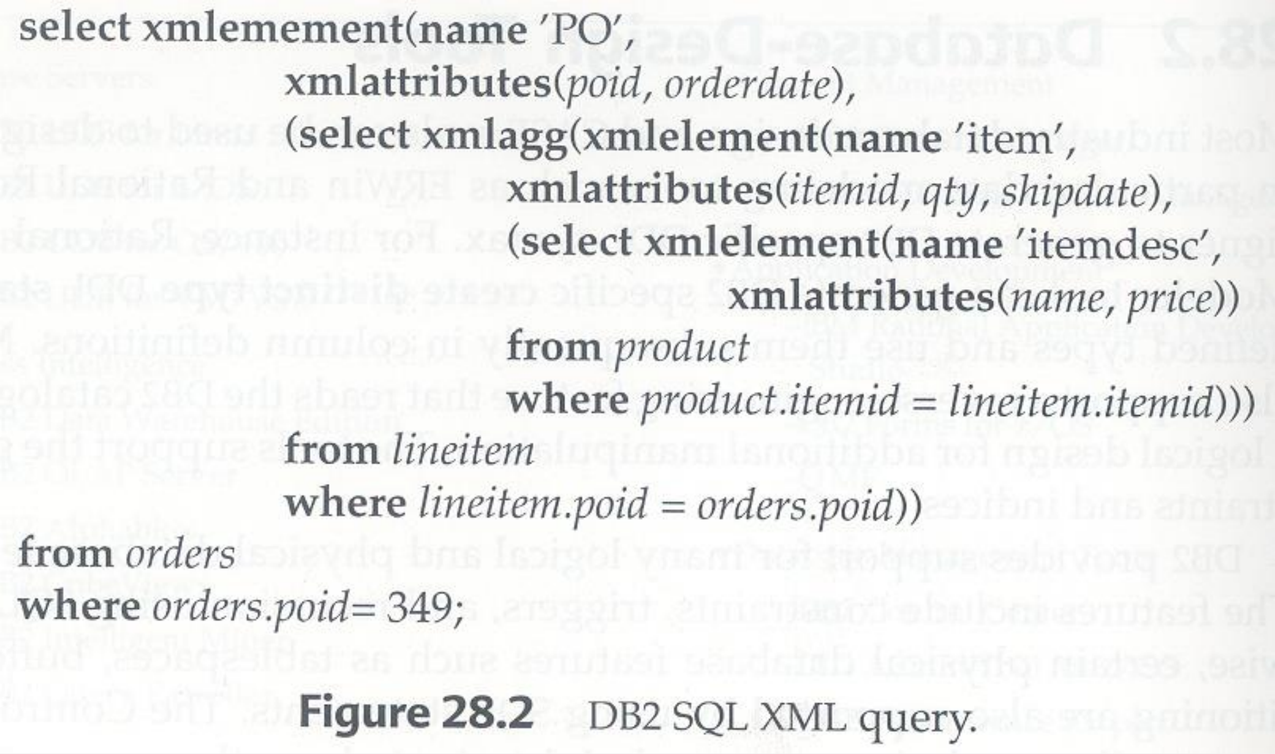
- **XML features**

  A rich set of XML functions have been included in DB2

  - **xmlelement**
  - **xmlattributes**
  - **xmlforest**
  - **xmlconcat**
  - **xmlserialize**
  - **xmlagg**
  - **xml2clob**
- Fig. 28.2 shows an SQL query with XML extensions
- Fig. 28.3 shows its resultant output

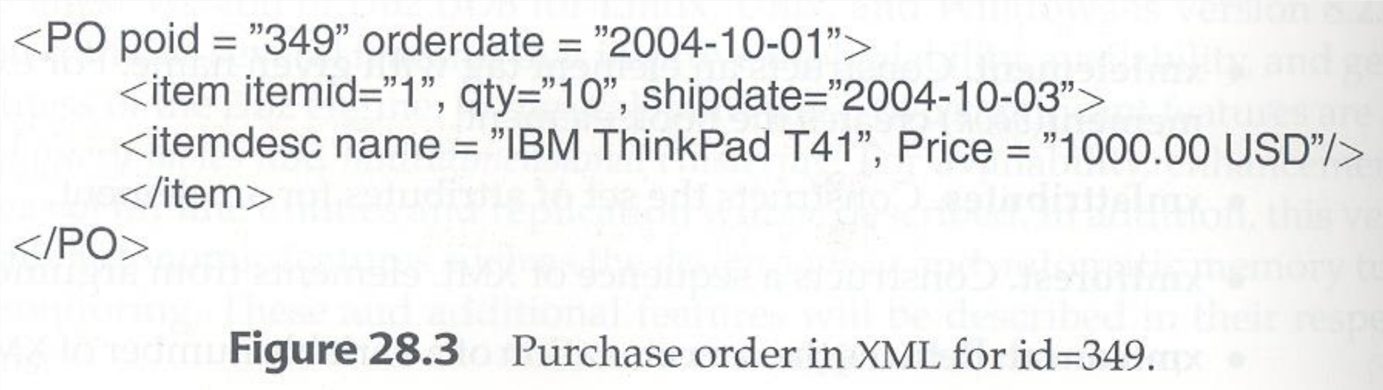


SQL Variations and Extensions (cont.)



```
select xmlemement(name ’PO’,
              xmlattributes(poid, orderdate),
              (select xmlagg(xmlelement(name ’item’,
                          xmlattributes(itemid, qty, shipdate),
                          (select xmlelement(name ’itemdesc’,
                                      xmlattributes(name, price))
                          from product
                          where product.itemid = lineitem.itemid)))
              from lineitem
              where lineitem.poid = orders.poid))
from orders
where orders.poid= 349;
```

Figure 28.2 DB2 SQL XML query.



```
<PO poid = ”349” orderdate = ”2004-10-01”>
      <item itemid=”1”, qty=”10”, shipdate=”2004-10-03”>
      <itemdesc name = ”IBM ThinkPad T41”, Price = ”1000.00 USD”/>
      </item>
</PO>
```

Figure 28.3 Purchase order in XML for id=349.

DB2 SQL Data Type Extension

- DB2 supports user-defined data types
- Users can define *distinct* or *structured* data types
- User-defined data types: ***distinct***

```
        create distinct type us_dollar as decimal(9,2)
```

- User can create a field in a table with type *us_dollar*

```
        select product from us_sales
        where price > us_dollar(1000)
```

- User-defined data types: ***structured***
  Structured data types are complex objects that usually consist of two or more attributes

```
        create type department_t as
                (deptname varchar(32),
                depthead varchar(32),
        faculty_count integer)
mode db2sql
```

- Structured types can be used to define ***typed tables***

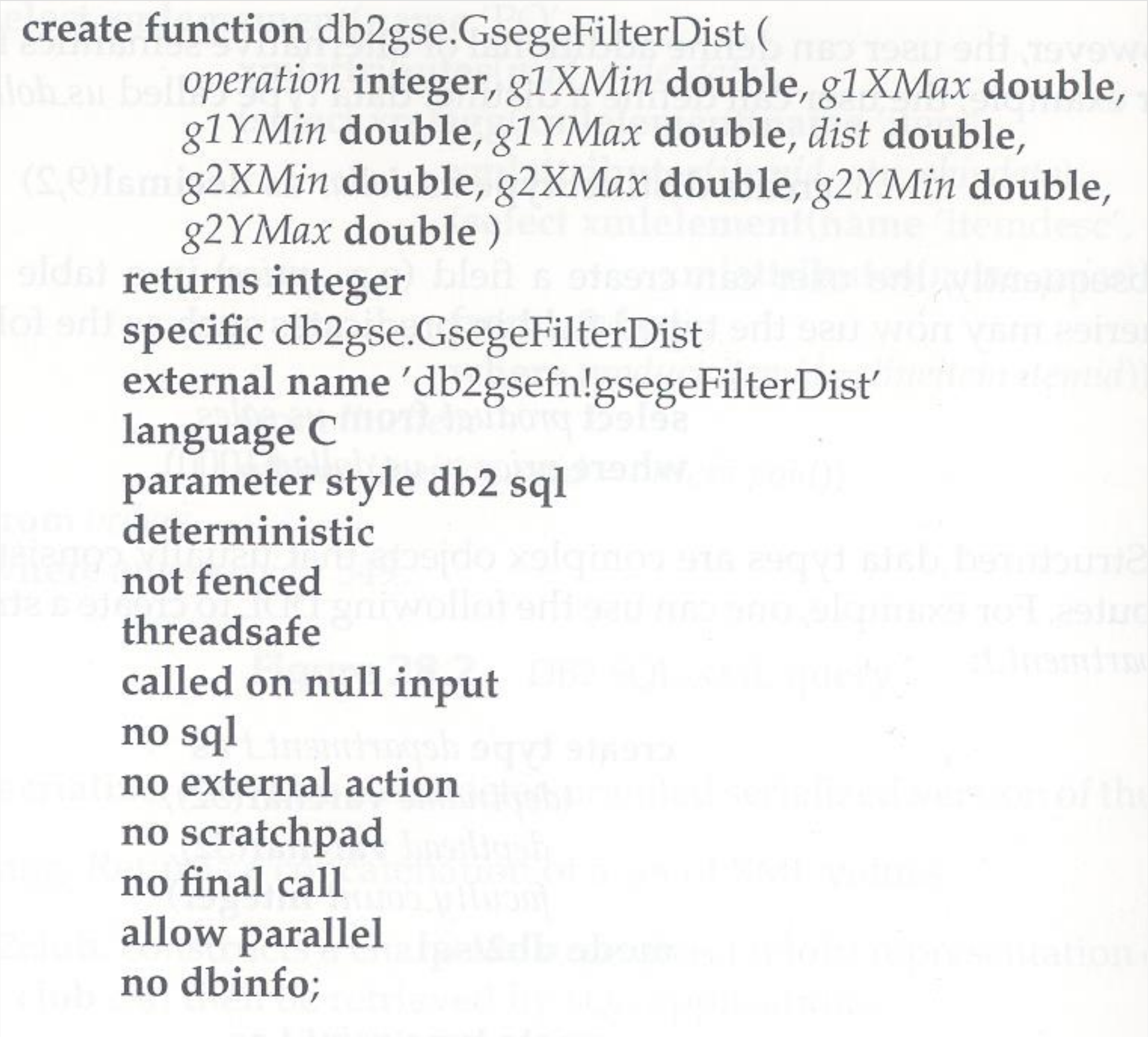
```
        create table dept of department_t
```

DB2 SQL Function and Method [1/2]

- **User-Defined Functions (UDFs) and Methods**
  - Users can define their own functions and methods
  - Functions can generate scalars (single attribute) or tables (multiattribute row) as their result
  - User can register functions using the **create function** statement
  - User-defined functions can operate in fenced or unfenced modes.
    - **In fenced mode:** the functions are executed by a separate thread in its own address space
    - **In unfenced mode:** the database processing agent is allowed to execute the function in the server's address space

DB2 SQL Function and Method [2/2]

- Fig 28.4 shows a definition of UDF, *db2gse.GsegeFilterDist*



```
create function db2gse.GsegeFilterDist (
        operation integer, g1XMin double, g1XMax double,
        g1YMin double, g1YMax double, dist double,
        g2XMin double, g2XMax double, g2YMin double,
        g2YMax double )
    returns integer
    specific db2gse.GsegeFilterDist
    external name 'db2gsefn!gsegeFilterDist'
    language C
    parameter style db2 sql
    deterministic
    not fenced
    threadsafe
    called on null input
    no sql
    no external action
    no scratchpad
    no final call
    allow parallel
    no dbinfo;
```

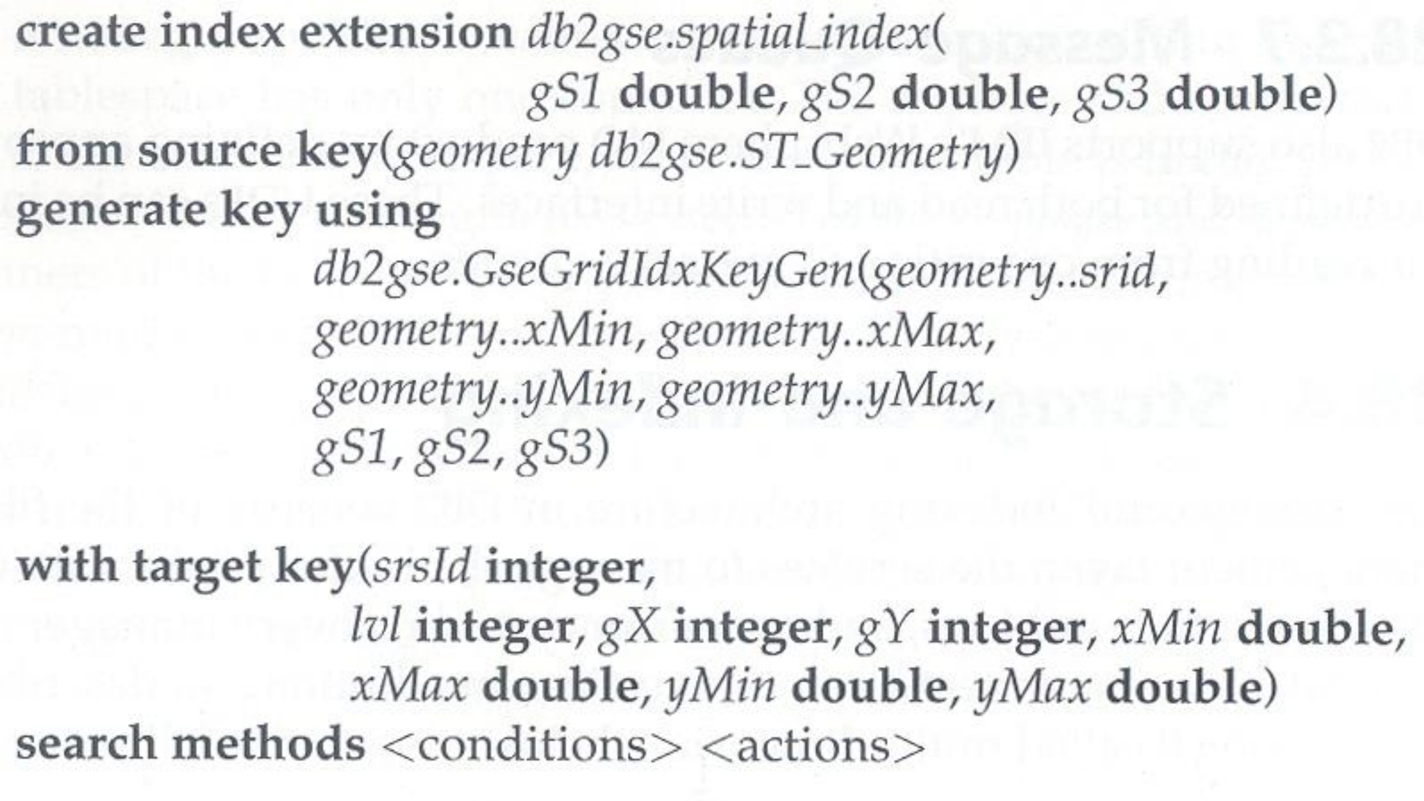
Figure 28.4 Definition of a UDF.

DB2 SQL Large Objects

- **Large Objects**
  - New database applications require the manipulation of text, images, video, and other types of data that are typically quite large
  - DB2 supports these requirements by providing three different large object (LOB) types
    - Binary Large Objects (**blob**s)
    - Single Byte Character Large Objects (**clob**s)
    - Double Byte Character Large Objects (**dbclob**s)

DB2 SQL Indexing and Constraints

- **Indexing Extensions and Constraints**
  - Users can create index extensions to generate keys from structured data types by using **create index extension** statement



```
create index extension db2gse.spatial_index(
                                gS1 double, gS2 double, gS3 double)
from source key(geometry db2gse.ST_Geometry)
generate key using
                  db2gse.GseGridIdxKeyGen(geometry..srid,
                  geometry..xMin, geometry..xMax,
                  geometry..yMin, geometry..yMax,
                  gS1, gS2, gS3)

with target key(srsId integer,
                    lvl integer, gX integer, gY integer, xMin double,
                    xMax double, yMin double, yMax double)
search methods <conditions> <actions>
```

Figure 28.5 Spatial index extension in DB2.

DB2 SQL Web Services

- **Web Services**
  - DB2 can integrate Web services as producer or consumer
  - For example, a Web service invokes the following SQL

```
select trn_id, amount, date
from transactions
where cust_id = <input>
order by date
fetch first 1 row only;
```

  - The following SQL shows DB2 acting as a consumer of a Web service

```
select ticker_id, GetQuote(ticker_id)
from portfolio
```

Chapter 29: IBM DB2 Universal Database

- 29.1 Overview
- 29.2 Database-Design Tools
- 29.3 SQL Variations and Extensions
- 29.4 Storage and Indexing
- 29.5 Multidimensional Clustering
- 29.6 Query Processing and Optimization
- 29.7 Materialized Query Tables
- 29.8 Autonomic Features in DB2
- 29.9 Tools and Utilities
- 29.10 Concurrency Control and Recovery
- 29.11 System Architecture
- 29.12 Replication, Distribution, and External Data
- 29.13 Business Intelligence Features
- Bibliographical Notes

DB2 Storage Structure [1/3]c

- DB2 provides storage abstraction for managing logical database tables usefully in multinode and multidisk environment
- *Nodegroups* can be defined to support table partitioning across a specific set of nodes in a multinode system

  → Flexibility in allocating table partitions

  ex) Large tables may be partitioned across all nodes in a system

  Small tables may reside on a single node
- DB2 uses *tablespaces* to organize tables
- A tablespace consists of one or more *containers,* which are references to directories, devices, or files
- A tablespace may contain zero or more database objects such as tables, indices, or LOBs
- Fig 28.6 shows that
  - Two tablespaces are defined for a nodegroup
  - *Humanres* tablespace is assigned four containers
  - *Sched* tablespace has only one container
  - *Employee* and *department* tables are assigned to the *humanres* tablespace
  - *Project* table is in the *sched* tablespace

DB2 Storage Structure [2/3]

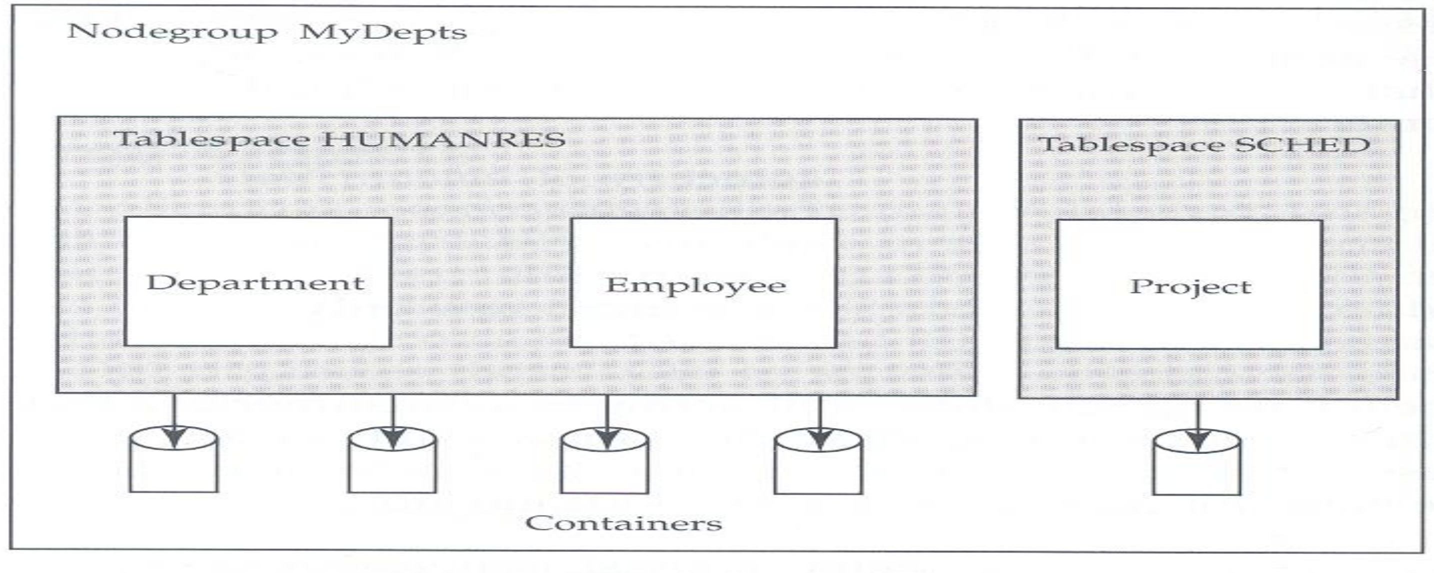


Figure 28.6 Tablespaces and containers in DB2.

- Two kinds of tablespaces
  - System-managed spaces (SMS)

    : Directories or file systems that the underlying operating system maintains
  - Data-managed spaces (DMS)

    : Raw devices or preallocated files that are then controlled by DB2
- DB2 support stripping across the different containers
  - One-by-one extent page in round-robin fashion
  - Benefits are parallel I/O and load balancing

DB2 Buffer Pools

- **Buffer Pools**
  - A buffer pool is a common shared data area that maintains memory copies of objects
  - One or more buffer pools may be associated each tablespace for managing different objects
  - DB2 allows buffer pools to be defined by SQL statements

    create bufferpool <buffer-pool> ….

    alter bufferpool <buffer-pool> **size** <n>

DB2 Tables, Records, and Indices [1/3]

- A table consists of a set of pages, and each page consists of a set of records
  - user data records or special system records
- DB2 uses a space map record called Free Space Control Record (FSCR) to find free space in table
- Fig 28.7 shows the logical view of a table and an associated index

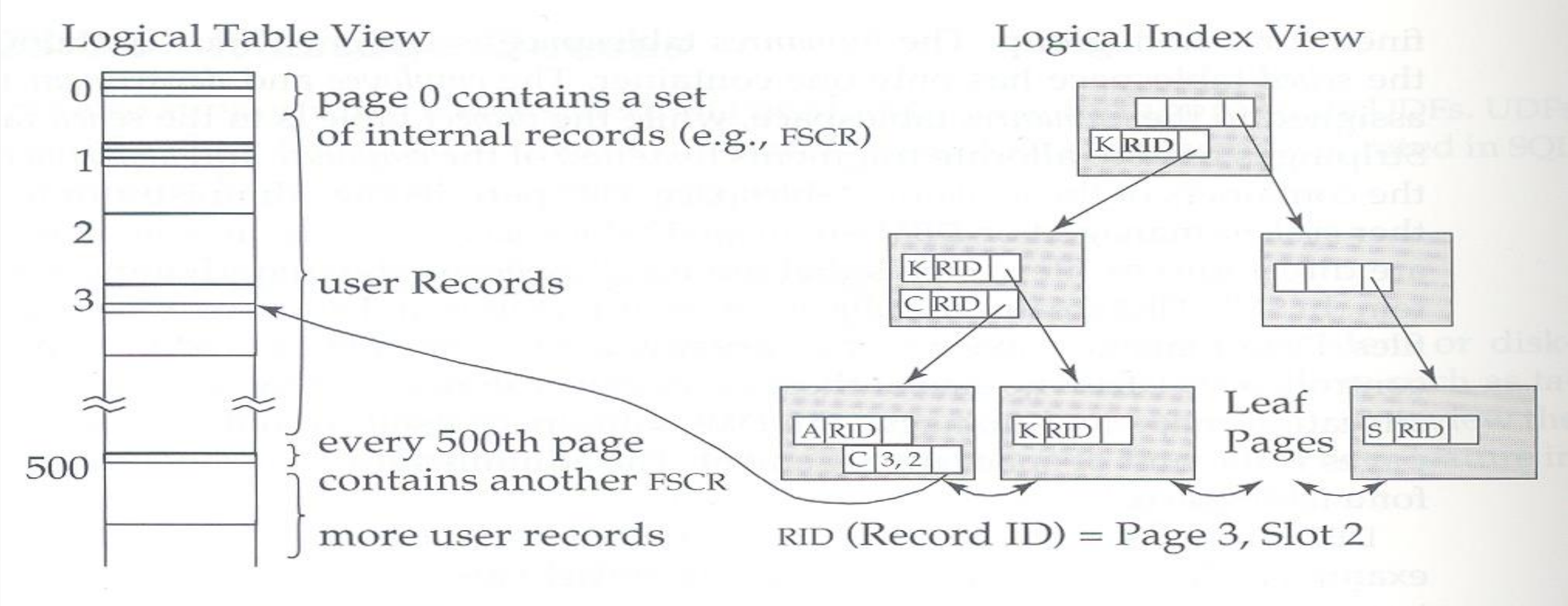

Logical Table View
0
1
2
3
500
page 0 contains a set of internal records (e.g., FSCR)
user Records
every 500th page contains another FSCR
more user records
Logical Index View
K RID
K RID
C RID
A RID
C 3, 2
K RID
S RID
Leaf Pages
RID (Record ID) = Page 3, Slot 2

Figure 28.7 Logical view of tables and indices in DB2.

DB2 Tables, Records, and Indices [2/3]

- Indices are organized as pages containing index records and pointers to child and sibling pages
- DB2 supports B+-tree index mechanisms
  - DB2 contains internal pages and leaf pages
  - Leaf pages contain index entries that point to records in the table
  - Each record in a table can be uniquely identified by using its page and slot information, called a *record ID (RID)*
- DB2 supports “include columns” in the index definition

  create unique index I1 **on** T1(C1) **include** (C2)

  - Specifies that C2 is to be included as an extra column in an index on column C1
  - This enables DB2 use “index-only” query processing techniques

DB2 Tables, Records, and Indices [3/3]

- **Data page:** Fig. 28.8 shows the typical data page format in DB2
  - Each data page contains a header and a slot directory
  - The slot directory is an array of 255 entries that points to record offsets in the page
  - Page 1056 contains record 1 at offset 3700, which is a forward pointer to the record <473, 2>
  - Record <473,2> is an overflow record created as a result of an update operation of the original record <1056,1>
  - DB2 supports different page sizes such as 4,8,16 and 32 KB

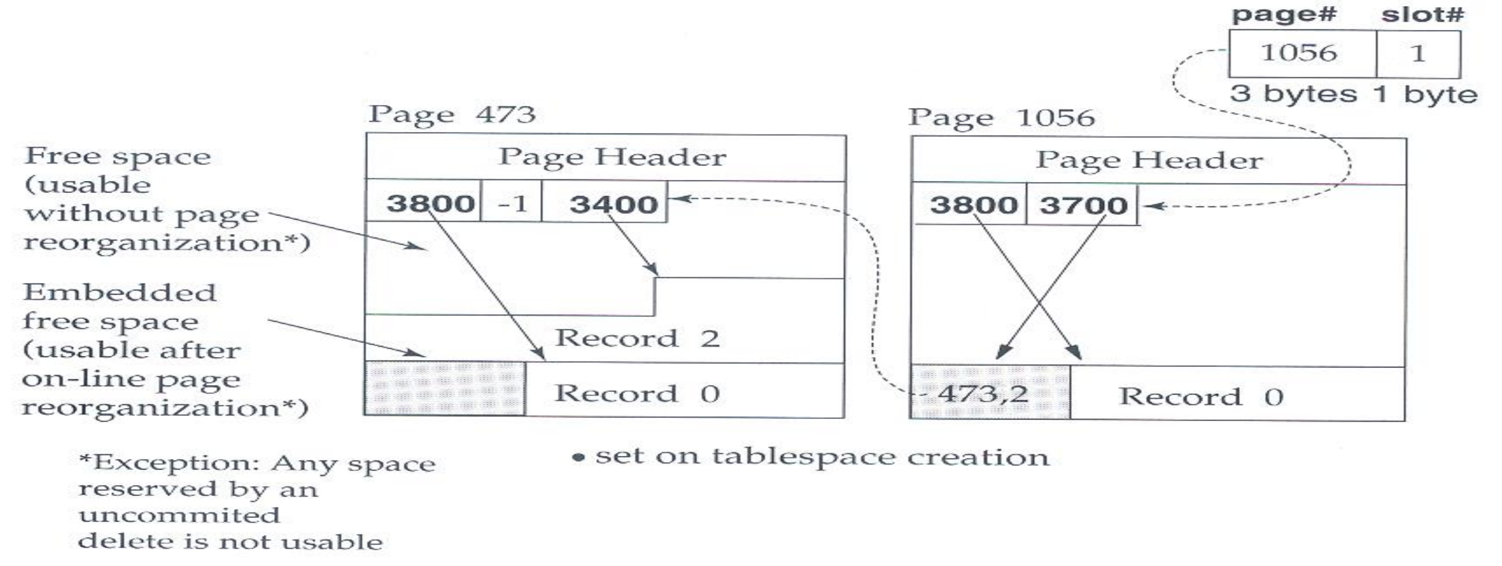

page#
slot#
1056
1
3 bytes
1 byte
Page 473
Page 1056
Page Header
Page Header
3800
-1
3400
3800
3700
Free space (usable without page reorganization*)
Embedded free space (usable after on-line page reorganization*)
Record 2
Record 0
473,2
Record 0
*Exception: Any space reserved by an uncommited delete is not usable
• set on tablespace creation

Figure 28.8 Data page and record layout in DB2.

Chapter 29: IBM DB2 Universal Database

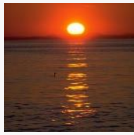
- 29.1 Overview
- 29.2 Database-Design Tools
- 29.3 SQL Variations and Extensions
- 29.4 Storage and Indexing
- 29.5 Multidimensional Clustering
- 29.6 Query Processing and Optimization
- 29.7 Materialized Query Tables
- 29.8 Autonomic Features in DB2
- 29.9 Tools and Utilities
- 29.10 Concurrency Control and Recovery
- 29.11 System Architecture
- 29.12 Replication, Distribution, and External Data
- 29.13 Business Intelligence Features
- Bibliographical Notes

DB2 Multidimensional Clustering (MDC) [1/2]

- DB2 table may be created by specifying one or more keys as dimensions along which to cluster the table's data
- A new clause called **organize by dimensions**
- For example, the following DDL describes a sales table organized by *storedId, year(orderDate), and itemId* as dimensions

```
create table sales(storeId int,
                   orderDate date,
                   shipDate date,
                   receiptDate date,
                   region int,
                   itemId int,
                   price float,
                   yearOd int generated always as year(orderDate))
organize by dimensions (region, yearOd, itemId)
```

- Each of dimensions may consist of one or more columns
- A 'dimension block index' is
  - automatically created for each of the specified dimensions
  - used to access data quickly and efficiently



DB2 Multidimensional Clustering (MDC) [2/2]

- A composite block index is used to maintain the clustering of data over insert and update activity
- Every unique combination of dimension values forms a logical “cell”, which is physically organized as block of pages
- Fig 28.9 shows a simple logical cube with only two values for each dimension attribute

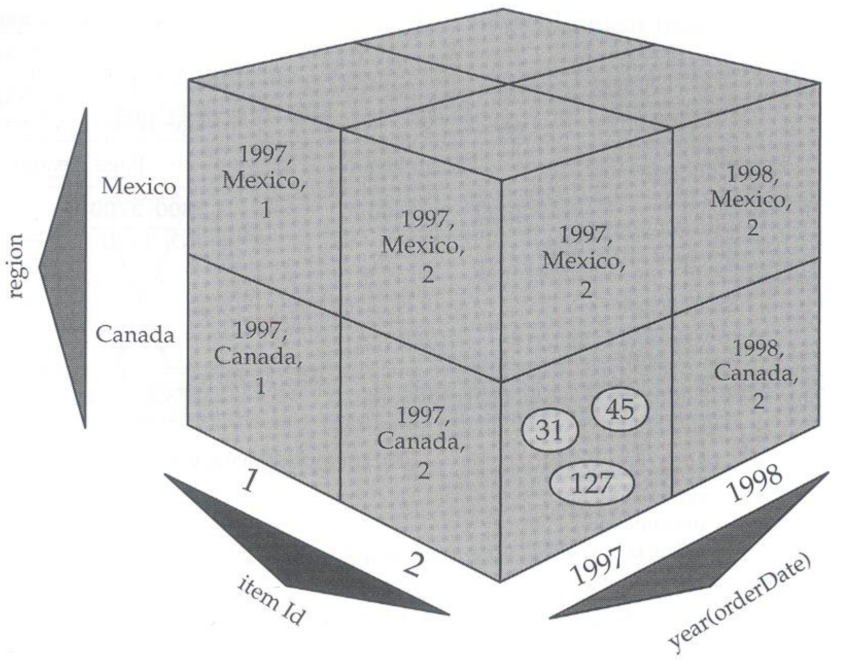
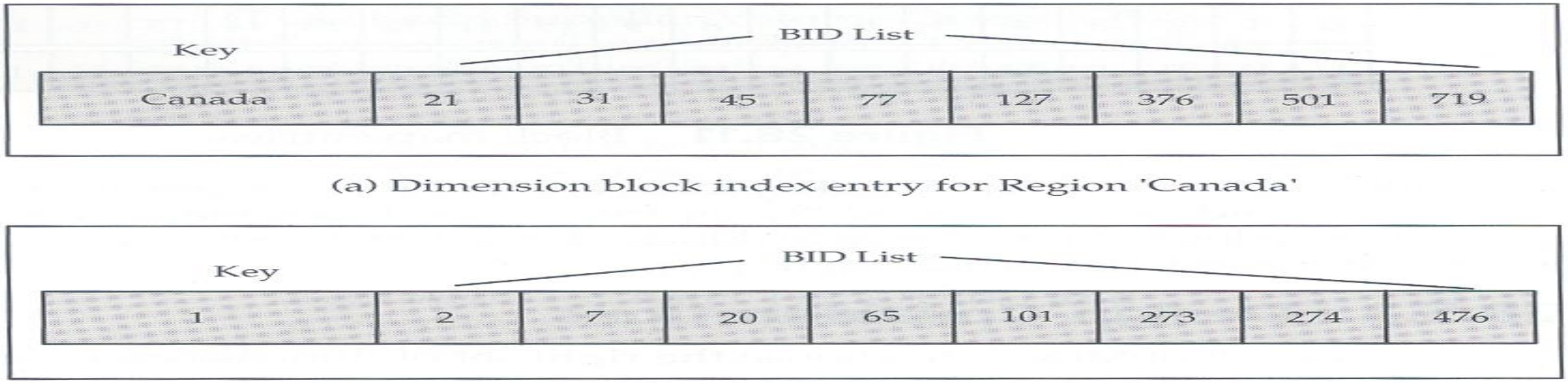


Figure 28.9 Logical view of physical layout of an MDC table.

DB2 Block Index

- **Block Indices**
  - A dimension block index is created on each of the *year(orderDate), region, and itemId* attributes
  - Each is structured in the same manner as a traditional B-tree index
    - except that, at the leaf level, the keys point to a *block identifier* (BID) instead of a RID
  - Fig. 28. 10 illustrates slices of blocks for specific values of *region* and *itemId* dimensions, respectively



Key	BID List							
Canada	21	31	45	77	127	376	501	719

(a) Dimension block index entry for Region 'Canada'

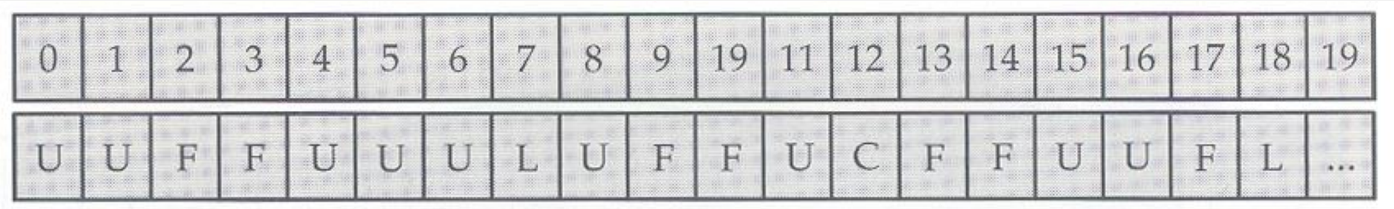
Key	BID List							
1	2	7	20	65	101	273	274	476

(b) Dimension block index entry for item Id = 1

Figure 28.10 Block index key entries.

DB2 Block Map

- **Block Map**
  - A block map
    - Is also associated with the table
    - Records the state of each block belonging to the table
    - May be in such states as
      - **In use**
      - **Free**
      - **Loaded**
      - **Requiring constant enforcement**

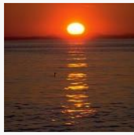


0	1	2	3	4	5	6	7	8	9	19	11	12	13	14	15	16	17	18	19
U	U	F	F	U	U	U	L	U	F	F	U	C	F	F	U	U	F	L	...

Figure 28.11 Block map entries.

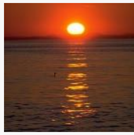
DB2 MDC Design Considerations

- A crucial aspect of MDC is to choose
  - The right set of dimensions for clustering a table
  - The right block size parameter to minimize the space utilization
- If chosen appropriately,
  - Significant performance and maintenance advantages
- If chosen incorrectly,
  - Performance degradation and worse space utilization
- There are a number of tuning knobs that can be exploited to organize the table



Chapter 29: IBM DB2 Universal Database

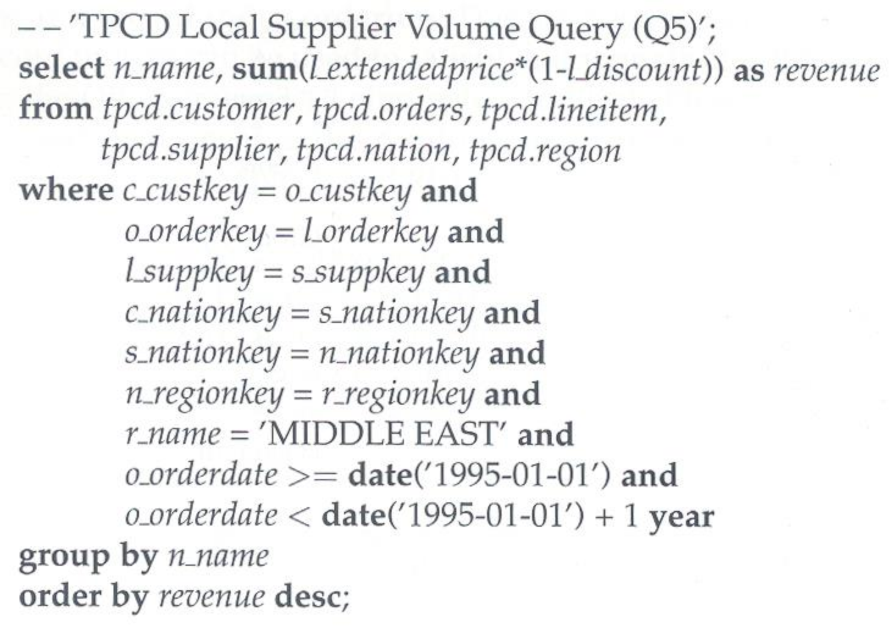
- 29.1 Overview
- 29.2 Database-Design Tools
- 29.3 SQL Variations and Extensions
- 29.4 Storage and Indexing
- 29.5 Multidimensional Clustering
- 29.6 Query Processing and Optimization
- 29.7 Materialized Query Tables
- 29.8 Autonomic Features in DB2
- 29.9 Tools and Utilities
- 29.10 Concurrency Control and Recovery
- 29.11 System Architecture
- 29.12 Replication, Distribution, and External Data
- 29.13 Business Intelligence Features
- Bibliographical Notes



DB2 Query Processing [1/3]

- DB2 queries are transformed into a tree of operations by the query compiler
- The query operator tree is used at execution time for processing
- DB2 supports a rich set of query operators
- Figure 28.12 and 28.13 show a query and its associated query plan
  - The query is a representative complex query from the TPC-H benchmark and contains several joins and aggregations
  - The query plan in this example is rather simple
  - DB2 provides powerful visual explain feature that can help users understand the details of a query execution plan
  - The query plan in the figure is based on the visual explain for the query

DB2 Query Processing [2/3]



```
– – 'TPCD Local Supplier Volume Query (Q5)';
select n_name, sum(l_extendedprice*(1-l_discount)) as revenue
from tpcd.customer, tpcd.orders, tpcd.lineitem,
     tpcd.supplier, tpcd.nation, tpcd.region
where c_custkey = o_custkey and
        o_orderkey = l_orderkey and
        l_suppkey = s_suppkey and
        c_nationkey = s_nationkey and
        s_nationkey = n_nationkey and
        n_regionkey = r_regionkey and
        r_name = 'MIDDLE EAST' and
        o_orderdate >= date('1995-01-01') and
        o_orderdate < date('1995-01-01') + 1 year
group by n_name
order by revenue desc;
```

Figure 28.12 SQL query.

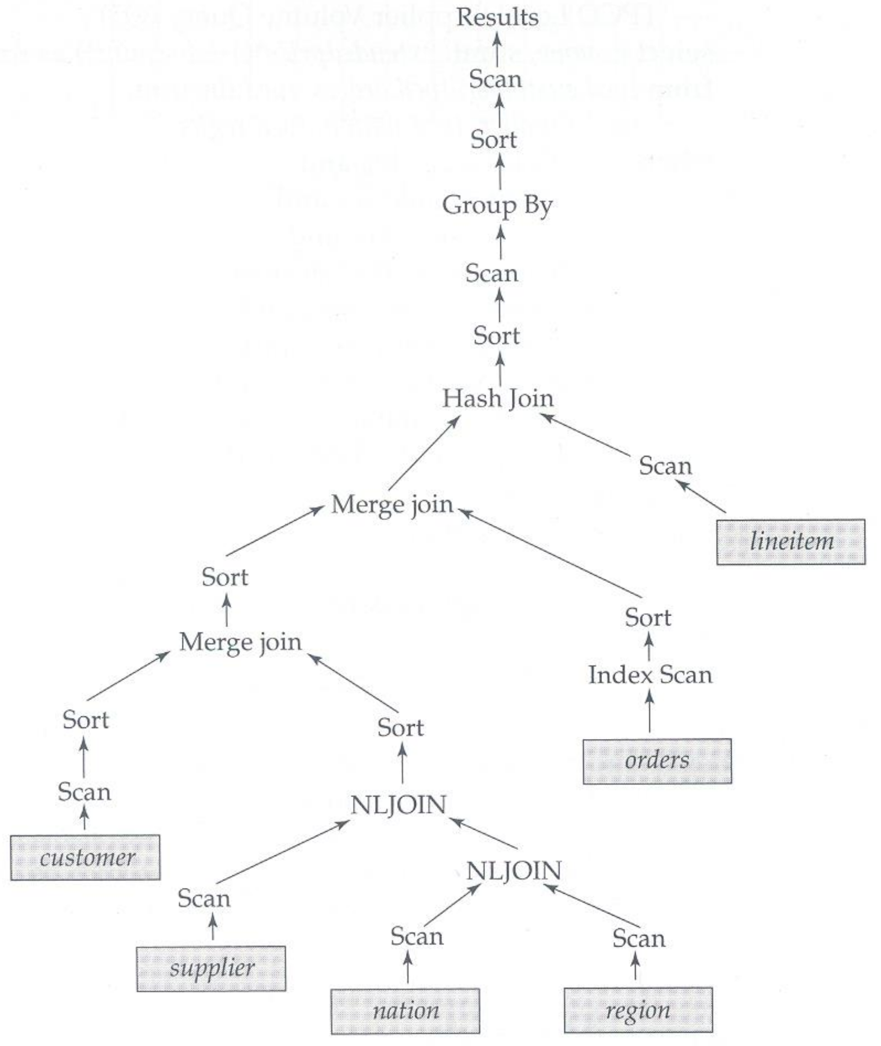


Figure 28.13 DB2 query plan (graphical explain).

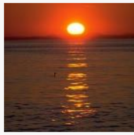
DB2 Query Processing [3/3]

# TPC- H benchmark

- The TPC is a non-profit corporation founded to define transaction processing and database benchmarks and to disseminate objective, verifiable TPC performance data to the industry
- The TPC Benchmark (TPC-H) is a decision support benchmark
- It consists of a suite of business oriented ad-hoc queries and concurrent data modifications
- The queries and the data populating the database have been chosen to have broad industry-wide relevance
- This benchmark illustrates decision support systems that examine large volumes of data, execute queries with a high degree of complexity, and give answers to critical business questions

DB2 Query Processing Access Methods [1/3]

- DB2 supports a comprehensive set of access methods
  - **Table scan**
    - Is the most basic method and performs a page-by-page access of all records in the table
  - **Index scan**
    - DB2 accesses the qualifying records using the RIDs in the index
  - **Block index scan**
    - A new access method for MDC tables
    - The qualifying blocks are accessed and processed
  - **Index only**
    - Is used when the index contains all the attributes required
    - A scan of the index entries is sufficient, and there is no need to fetch the records

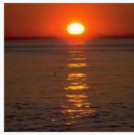


DB2 Query Processing Access Methods [2/3]

- **List prefetch**
  - This access method is chosen for an unclustered index scan with a significant number of RIDs
  - DB2 collects the RIDs of relevant records using an index scan
  - Then sorts the RIDs by page number
  - Finally performs a fetch of the records in sorted order from the data pages
- **Block and record index ANDing**
  - DB2 uses this method when it determines that more than one index can be used to constrain the number of satisfying records in a base table
  - It processes the most selective index to generate a list of BIDs or RIDs
  - It then processes the next selective index to return the BIDs or RIDs
  - A BID or RID qualifies for further processing only if it is present in the intersection (AND operation)
- **Block and record index ORing**
  - Two or more block or record indices can be used to satisfy query predicates that are combined by using the OR operator
  - DB2 eliminates duplicate BIDs or RIDs by performing a sort and then fetches the resulting set of records

DB2 Query Processing for Complex Operations

- **Join, Aggregation, and Set Operations**
  - DB2 can choose between nested-loop, sort-merge, and hash-join techniques
  - In describing the join and set binary operation, we use the notation of “outer” and “inner” tables to distinguish the two input streams
    - Nested-loop technique is useful if the inner table
      - Is very small
      - Or can be accessed by using an index on a join predicate
    - Sort-merge-join and hash-join techniques are used for joins involving large outer and inner tables
    - Merging technique eliminates duplicates in the case of union
    - DB2 implements set operations by using sorting and merging techniques
- **Support for Complex SQL Processing**
  - One of the most important aspects of DB2 is that it uses the query processing infrastructure in an extensible fashion to support complex SQL operation
  - The complex SQL operations include support for deeply nested and correlated subqueries as well as constraints, referential integrity, and triggers
  - DB2 can provide better efficiency and scalability by performing most of the constraint checking and triggering actions as part of the query plan

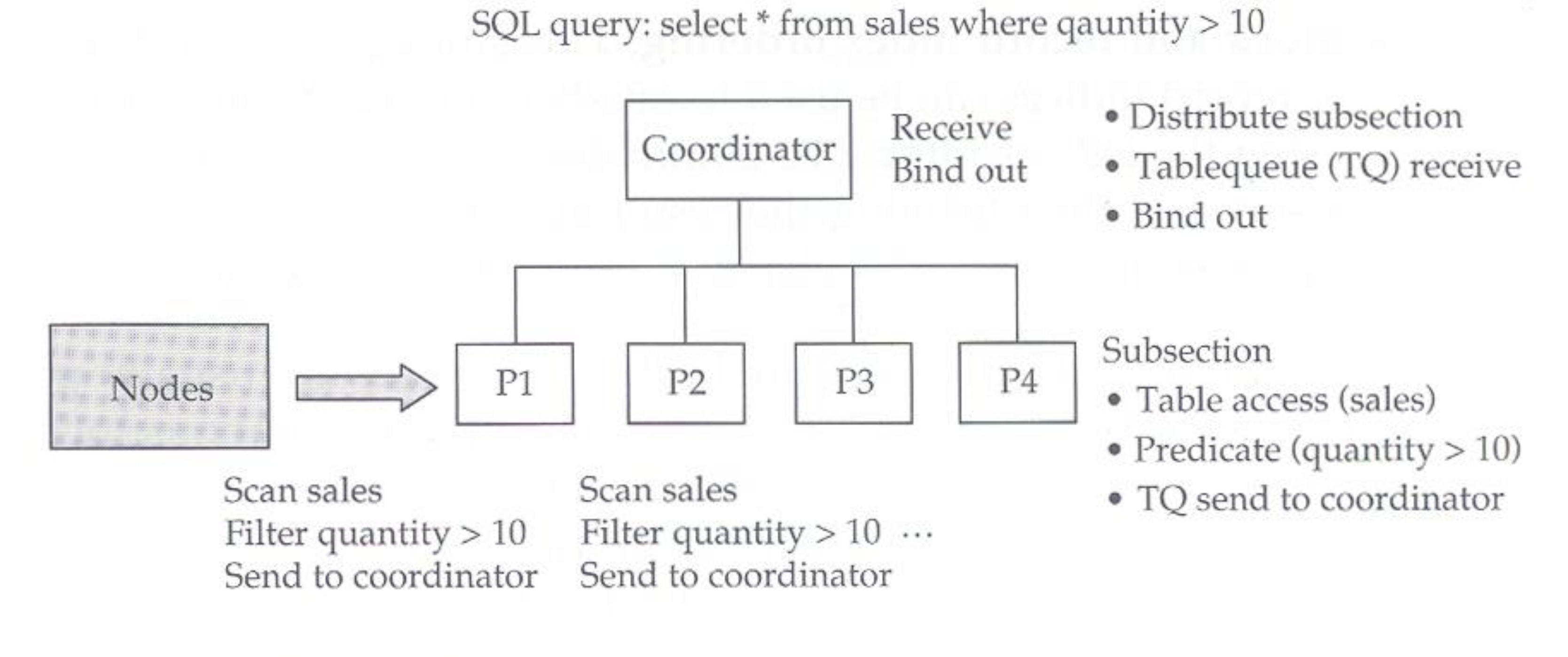


DB2 Multiprocessor Query Processing [1/2]

- **Multiprocessor Query-Processing Features**
  - DB2 extends the base set of query operation with control and data exchange primitives to support SMP, MPP, and SMP-cluster modes of query processing
  - DB2 uses a “tablequeue” abstraction for data exchange between threads on different nodes or on the same node
  - The table queue is a buffer that redirects data to appropriate receivers by using broadcast, one-to-one, or directed multicast methods
  - In all these modes, DB2 employs a coordinator process to control the query operations and final result gathering
  - Coordinator processes can also perform some global database-processing actions

DB2 Multiprocessor Query Processing [2/2]

- **Multiprocessor Query-Processing Features (cont.)**
  - Fig 28.14 shows a simple query executing in a 4-node MPP system



SQL query: select * from sales where qauntity > 10

Coordinator

Receive
Bind out

- Distribute subsection
- Tablequeue (TQ) receive
- Bind out

Nodes

P1 P2 P3 P4

Subsection
- Table access (sales)
- Predicate (quantity > 10)
- TQ send to coordinator

Scan sales
Filter quantity > 10
Send to coordinator

Scan sales
Filter quantity > 10 …
Send to coordinator

Figure 28.14 DB2 MPP query processing using function shipping.

DB2 Query Optimization

- **Query Optimization**

  Procedures

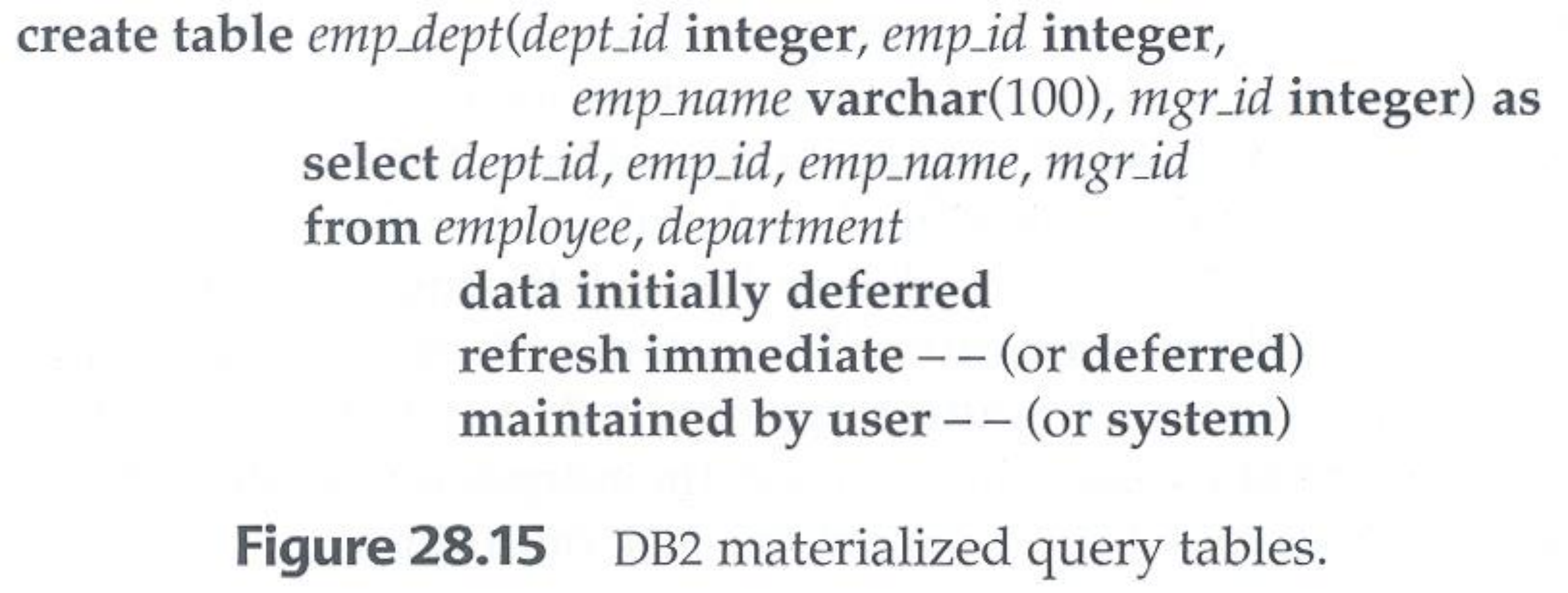
  - Parsing the SQL statement to **QGM** (Query Graph Model)
  - Performing semantic transformations on the QGM to enforce constraints, referential integrity, and triggers → Enhanced QGM
  - Performing *query rewrite* transformations
    - Decorrelation of correlated subqueries
    - Transformation of subqueries into joins using early-out processing
    - Pushing the group by operation below joins
    - Using materialized view for portions of the original query
  - The query optimizer component uses this enhanced and transformed QGM as its input for optimization

Chapter 29: IBM DB2 Universal Database

- 29.1 Overview
- 29.2 Database-Design Tools
- 29.3 SQL Variations and Extensions
- 29.4 Storage and Indexing
- 29.5 Multidimensional Clustering
- 29.6 Query Processing and Optimization
- 29.7 Materialized Query Tables
- 29.8 Autonomic Features in DB2
- 29.9 Tools and Utilities
- 29.10 Concurrency Control and Recovery
- 29.11 System Architecture
- 29.12 Replication, Distribution, and External Data
- 29.13 Business Intelligence Features
- Bibliographical Notes

Materialized Query Tables [1/3]

- **Materialized Query Tables (MQTs)**
  - In DB2, the materialized views are called materialized query tables
  - MQTs are specified by using **create table** statement as shown by Fig. 28.15



```
create table emp_dept(dept_id integer, emp_id integer,
                        emp_name varchar(100), mgr_id integer) as
           select dept_id, emp_id, emp_name, mgr_id
           from employee, department
                  data initially deferred
                  refresh immediate – – (or deferred)
                  maintained by user – – (or system)
```

Figure 28.15 DB2 materialized query tables.

Materialized Query Tables [2/3]

- **Query Routing to MQTs**
  - The internal QGM model allows the compiler to
    - match the input query against the available MQT definitions
    - choose appropriate MQTs for consideration
  - Fig. 28.16 shows the entire flow of the reroute and optimization

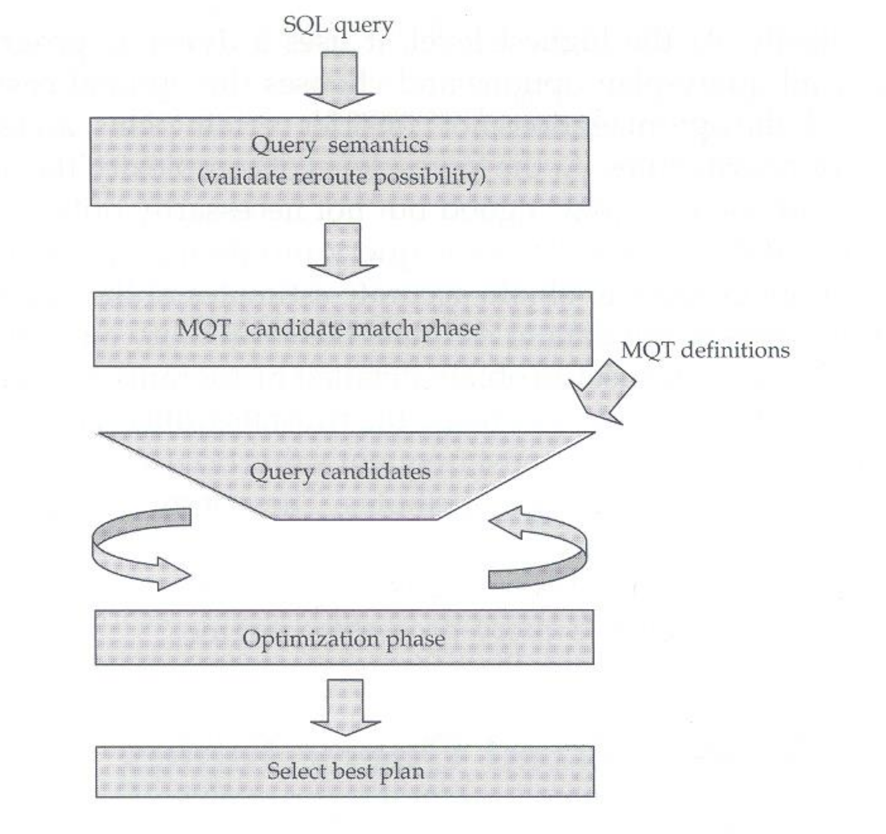
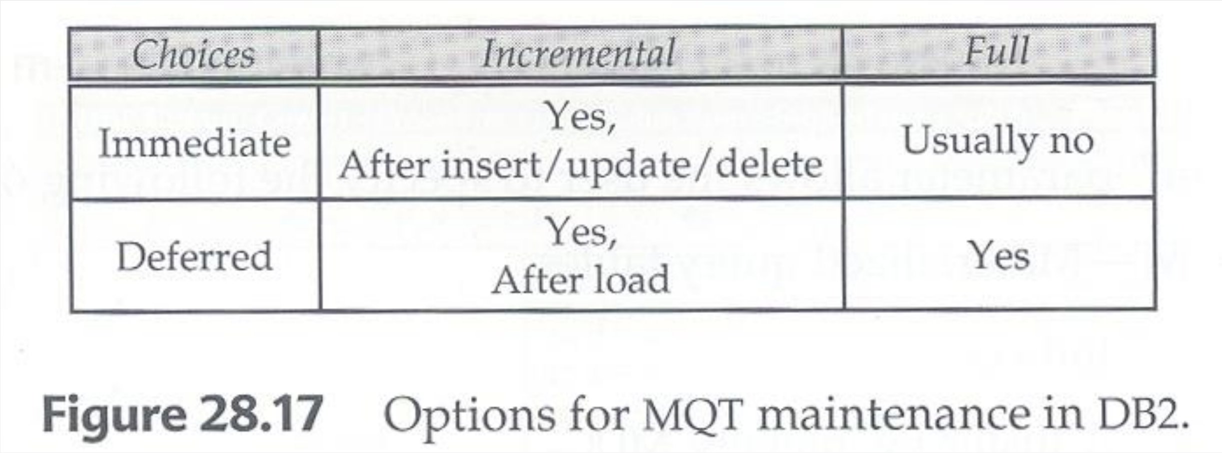


Figure 28.16 MQT matching and optimization in DB2.

Materialized Query Tables [3/3]

- **Maintenance of MQTs**
  - Two dimensions to maintenance: time and cost
    - Time dimension
      - *Immediate*
      - *Deferred*
    - Size dimension
      - *Incremental*
      - *Full*



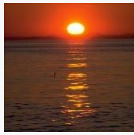
Choices	*Incremental*	*Full*
Immediate	Yes, After insert/update/delete	Usually no
Deferred	Yes, After load	Yes

Figure 28.17 Options for MQT maintenance in DB2.

- Fig. 28.17 shows the two dimensions and the options that are the most useful

Chapter 29: IBM DB2 Universal Database

- 29.1 Overview
- 29.2 Database-Design Tools
- 29.3 SQL Variations and Extensions
- 29.4 Storage and Indexing
- 29.5 Multidimensional Clustering
- 29.6 Query Processing and Optimization
- 29.7 Materialized Query Tables
- 29.8 Autonomic Features in DB2
- 29.9 Tools and Utilities
- 29.10 Concurrency Control and Recovery
- 29.11 System Architecture
- 29.12 Replication, Distribution, and External Data
- 29.13 Business Intelligence Features
- Bibliographical Notes



Autonomic Features in DB2 [1/2]

- Autonomic computing includes a set of techniques that allow the computing environment to manage itself and reduce the external dependencies

- **Configuration**
  - DB2 supports autonomic tuning of various memory and system configuration parameters
  - For instance, parameters such as buffer pool sizes and sort heap sizes can be specified as autonomic

Autonomic Features in DB2 [2/2]

- **Optimization**
  - Important aspects of improving the performance
    - Auxiliary data structures (indices, MQTs)
    - Data organization features (partitioning, clustering)
  - *Design Advisor*
    - Provides workload-based advice
    - Automatically analyze a workload, using optimization techniques
    - ex) db2advis -d <DB name> -i <workloadfile> -m MICP
      - M: Materialized query tables
      - I: Indices
      - C: Clustering, namely, MDC
      - P: Partitioning key selection

Chapter 29: IBM DB2 Universal Database

- 29.1 Overview
- 29.2 Database-Design Tools
- 29.3 SQL Variations and Extensions
- 29.4 Storage and Indexing
- 29.5 Multidimensional Clustering
- 29.6 Query Processing and Optimization
- 29.7 Materialized Query Tables
- 29.8 Autonomic Features in DB2
- 29.9 Tools and Utilities
- 29.10 Concurrency Control and Recovery
- 29.11 System Architecture
- 29.12 Replication, Distribution, and External Data
- 29.13 Business Intelligence Features
- Bibliographical Notes

DB2 Tools and Utilities [1/3]

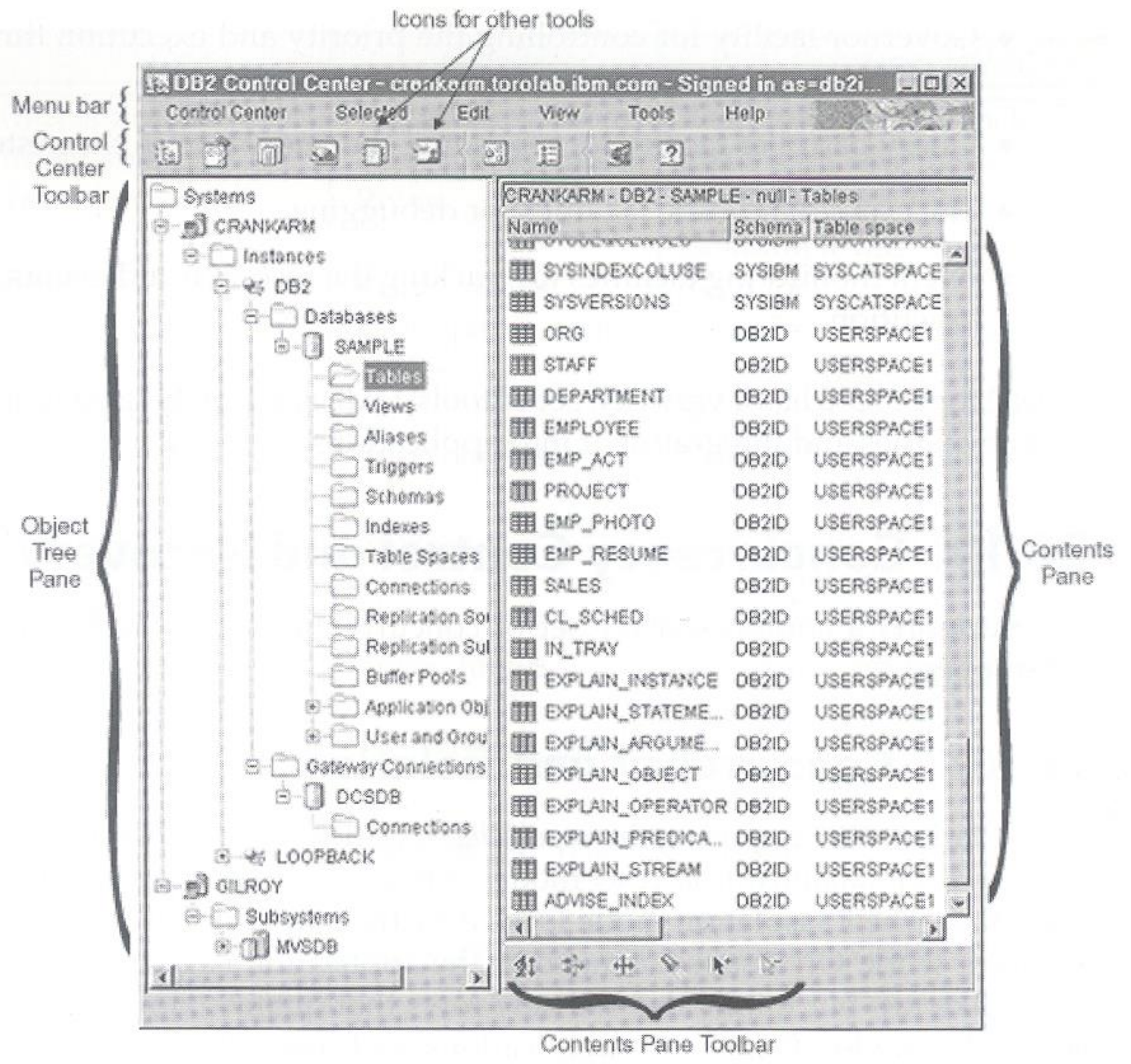
- The DB2 control center is the primary tool for use and administration of DB2 databases

- It is organized from data objects such as servers, databases, tables, and indices

- It contains task-oriented interfaces to perform commands allows users to generate SQL scripts

- Fig. 28.18 shows a screen shot of the main panel of the Control Center

DB2 Tools and Utilities [2/3]



Name	Schema	Table space
SYSINDEXCOLUSE	SYSIBM	SYSCATSPACE
SYSVERSIONS	SYSIBM	SYSCATSPACE
ORG	DB2ID	USERSPACE1
STAFF	DB2ID	USERSPACE1
DEPARTMENT	DB2ID	USERSPACE1
EMPLOYEE	DB2ID	USERSPACE1
EMP_ACT	DB2ID	USERSPACE1
PROJECT	DB2ID	USERSPACE1
EMP_PHOTO	DB2ID	USERSPACE1
EMP_RESUME	DB2ID	USERSPACE1
SALES	DB2ID	USERSPACE1
CL_SCHED	DB2ID	USERSPACE1
IN_TRAY	DB2ID	USERSPACE1
EXPLAIN_INSTANCE	DB2ID	USERSPACE1
EXPLAIN_STATEME...	DB2ID	USERSPACE1
EXPLAIN_ARGUME...	DB2ID	USERSPACE1
EXPLAIN_OBJECT	DB2ID	USERSPACE1
EXPLAIN_OPERATOR	DB2ID	USERSPACE1
EXPLAIN_PREDICA...	DB2ID	USERSPACE1
EXPLAIN_STREAM	DB2ID	USERSPACE1
ADVISE_INDEX	DB2ID	USERSPACE1

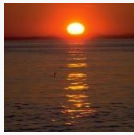
Figure 28.18 DB2 Control Center.

DB2 Tools and Utilities [3/3]

- **Utilities**
  - DB2 provides comprehensive support for load, import, export, reorg, redistribute, and other data-related utilities
  - DB2 supports a number of tools such as
    - Audit facility
    - Governor facility
    - Query patroller facility
    - Trace and diagnostic facilities
    - Event monitoring facilities

Chapter 29: IBM DB2 Universal Database

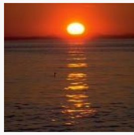
- 29.1 Overview
- 29.2 Database-Design Tools
- 29.3 SQL Variations and Extensions
- 29.4 Storage and Indexing
- 29.5 Multidimensional Clustering
- 29.6 Query Processing and Optimization
- 29.7 Materialized Query Tables
- 29.8 Autonomic Features in DB2
- 29.9 Tools and Utilities
- 29.10 Concurrency Control and Recovery
- 29.11 System Architecture
- 29.12 Replication, Distribution, and External Data
- 29.13 Business Intelligence Features
- Bibliographical Notes



DB2 Concurrency Control [1/3]

Concurrency and Isolation

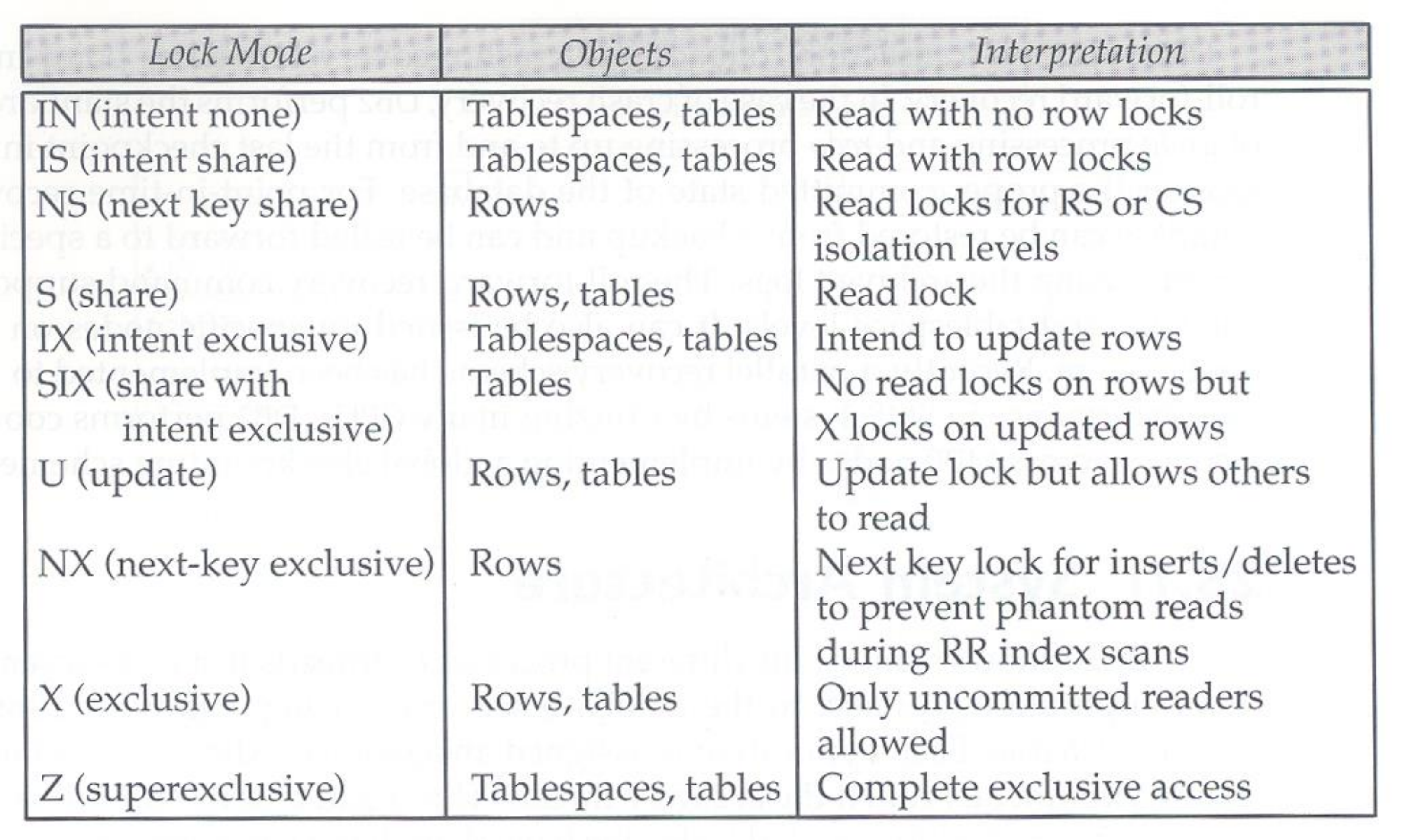
- DB2 supports various modes of concurrency control and isolation
- *Repeatable read* (RR) mode
  - Corresponds closely to the serializable level
  - Ensures that range scans will find the same set of tuples if repeated
- *Read stability* (RS) mode
  - Locks only the rows that an application retrieves in a unit of work
- *Cursor stability* (CS) mode
  - Default isolation level
- *Uncommitted read* (UR) mode



DB2 Concurrency Control [2/3]

- **Concurrency and Isolation (cont.)**
  - The various isolation modes are implemented by using locks
  - DB2 supports record-level and table-level locks
  - DB2 escalates from record-level to table-level locks if the space in the lock table becomes tight
  - DB2 supports a variety of lock modes in order to maximize concurrency
  - Fig. 28.19 shows the different lock modes and their descriptions

DB2 Concurrency Control [3/3]



Lock Mode	*Objects*	*Interpretation*
IN (intent none)	Tablespaces, tables	Read with no row locks
IS (intent share)	Tablespaces, tables	Read with row locks
NS (next key share)	Rows	Read locks for RS or CS isolation levels
S (share)	Rows, tables	Read lock
IX (intent exclusive)	Tablespaces, tables	Intend to update rows
SIX (share with intent exclusive)	Tables	No read locks on rows but X locks on updated rows
U (update)	Rows, tables	Update lock but allows others to read
NX (next-key exclusive)	Rows	Next key lock for inserts/deletes to prevent phantom reads during RR index scans
X (exclusive)	Rows, tables	Only uncommitted readers allowed
Z (superexclusive)	Tablespaces, tables	Complete exclusive access

Figure 28.19 DB2 lock modes.

DB2 Recovery

- **Logging and Recovery**
  - DB2 supports two types of log modes
    - Circular logging : useful for crash recovery or application failure recovery
    - Archival logging : required for roll-forward recovery from an archival backup
  - DB2 supports transaction rollback and crash recovery as well as point-in-time or roll-forward recovery
    - In crash recovery, DB2 performs the standard phases of *undo* and *redo* processing
    - For point-in-time recovery, database can be restored from a backup and can be rolled forward to a specific point in time using the archived logs
    - The roll-forward recovery command supports both database and tablespace levels

Chapter 29: IBM DB2 Universal Database

- 29.1 Overview
- 29.2 Database-Design Tools
- 29.3 SQL Variations and Extensions
- 29.4 Storage and Indexing
- 29.5 Multidimensional Clustering
- 29.6 Query Processing and Optimization
- 29.7 Materialized Query Tables
- 29.8 Autonomic Features in DB2
- 29.9 Tools and Utilities
- 29.10 Concurrency Control and Recovery
- 29.11 System Architecture
- 29.12 Replication, Distribution, and External Data
- 29.13 Business Intelligence Features
- Bibliographical Notes

DB2 Process Model [1/2]

- **DB2 Process Model**

  Fig. 28.20 shows some of the different process or threads in DB2

  - Remote client applications connect to the database server through communication agents such as *db2tcpcm*
  - Each application is assigned an agent called the *db2agent* thread
    - A dedicated server process that serves for each active connection
  - This agent and its subordinate agents perform the application-related tasks
  - There are a set of agents at the level of the server to perform tasks such as crash detection, license server, process creation and control of system resources

DB2 Process Model [2/2]

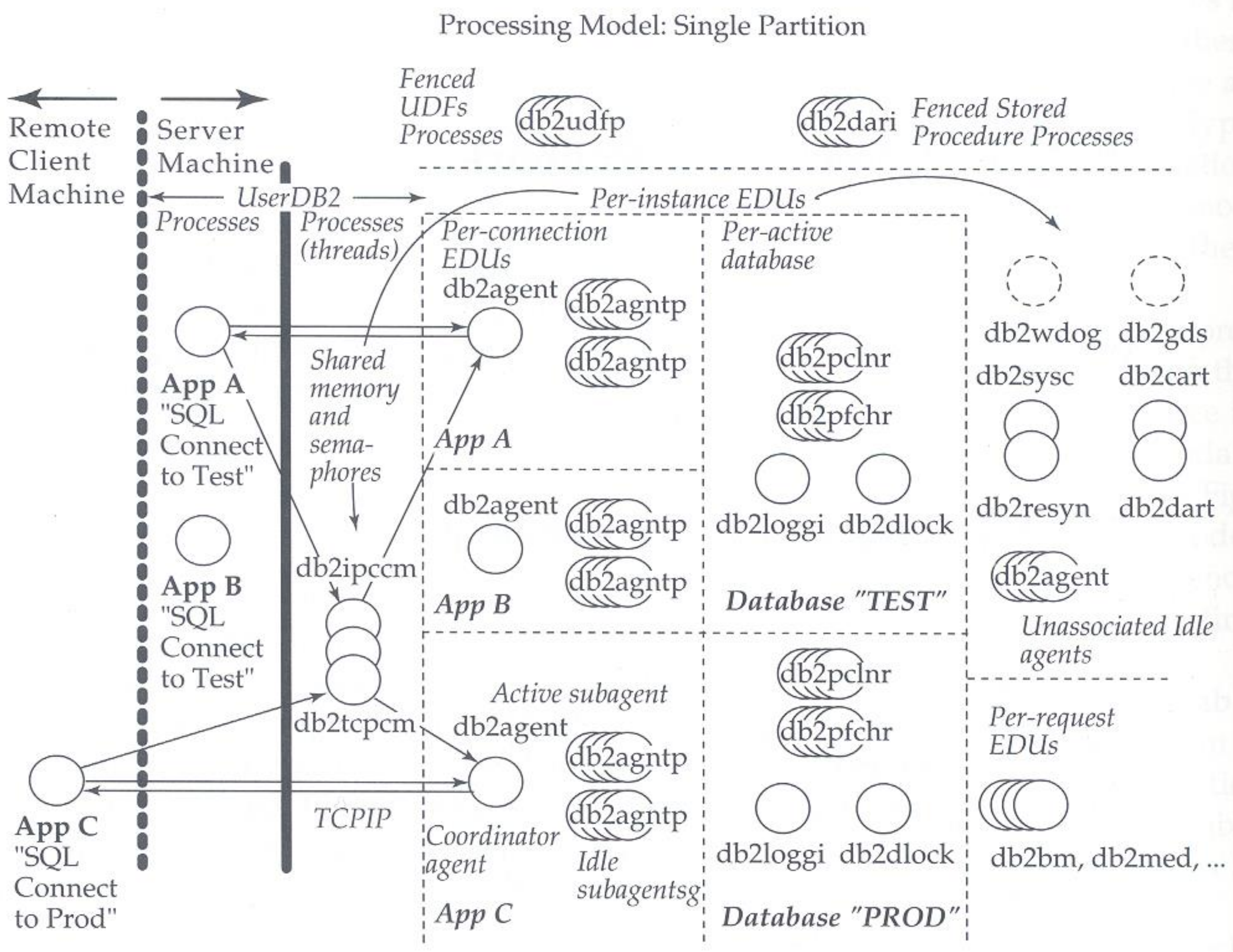


Figure 28.20 Process model in DB2.

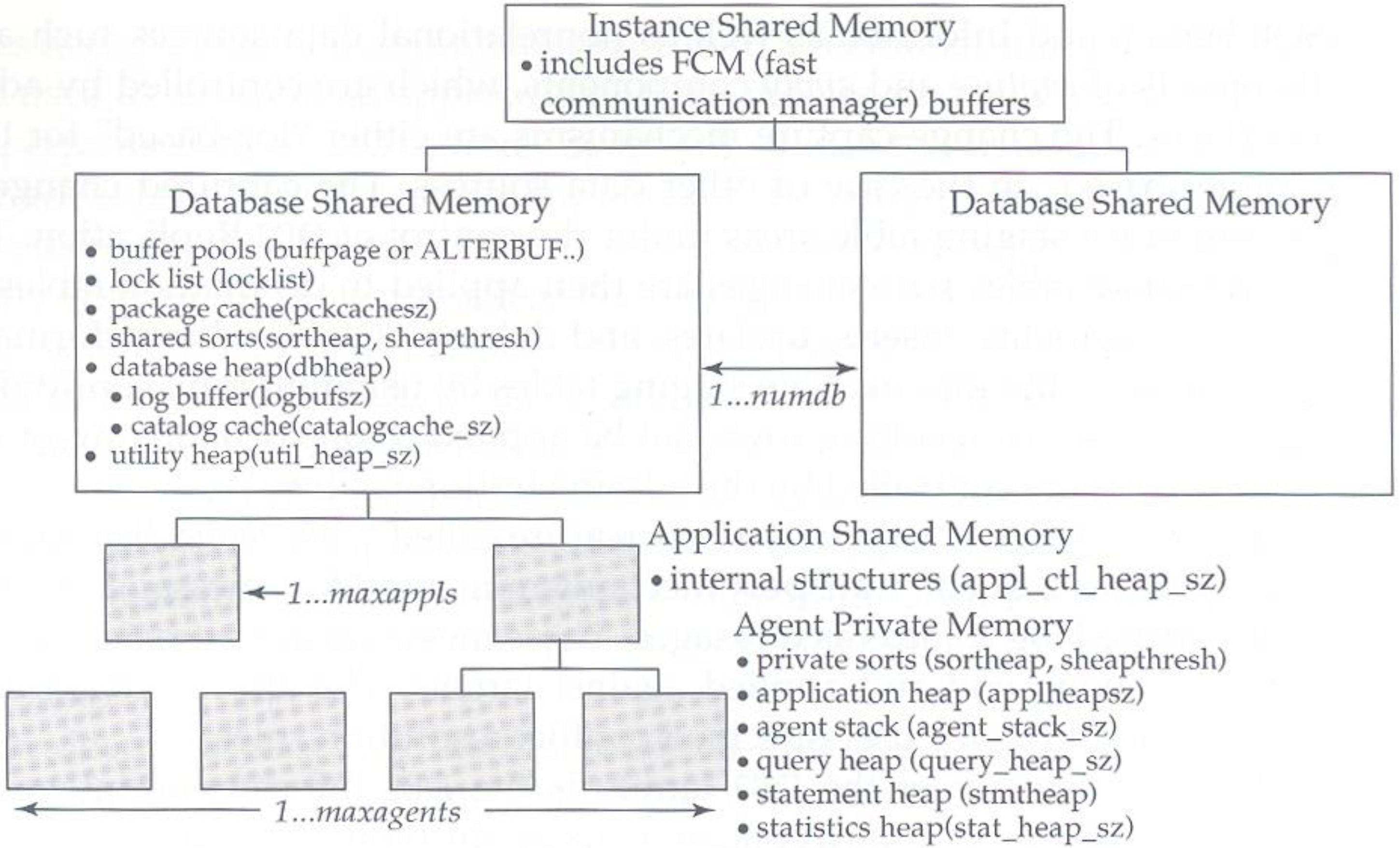
DB2 Memory Model [1/2]

- **DB2 Memory Model**

  Fig. 28.21 shows the different types of memory segment in DB2

  - Private memory in agents or threads is mainly used for local variables and data structures that are relevant only for the current activity
  - Shared memory is partitioned into
    - *database shared memory*
      - contains useful data structures such as buffer pool, lock lists, application package caches, and shared sort areas
    - *server shared memory* and *application shared memory*
      - are primarily used for common data structures and communication buffers

DB2 Memory Model [2/2]



Instance Shared Memory

- includes FCM (fast communication manager) buffers

Database Shared Memory

- buffer pools (buffpage or ALTERBUF..)
- lock list (locklist)
- package cache(pckcachesz)
- shared sorts(sortheap, sheapthresh)
- database heap(dbheap)
  - log buffer(logbufsz)
  - catalog cache(catalogcache_sz)
- utility heap(util_heap_sz)

1...numdb

Database Shared Memory

1...maxappls

Application Shared Memory

- internal structures (appl_ctl_heap_sz)

Agent Private Memory

- private sorts (sortheap, sheapthresh)
- application heap (applheapsz)
- agent stack (agent_stack_sz)
- query heap (query_heap_sz)
- statement heap (stmtheap)
- statistics heap(stat_heap_sz)

1...maxagents

Figure 28.21 DB2 memory model.

Chapter 29: IBM DB2 Universal Database

- 29.1 Overview
- 29.2 Database-Design Tools
- 29.3 SQL Variations and Extensions
- 29.4 Storage and Indexing
- 29.5 Multidimensional Clustering
- 29.6 Query Processing and Optimization
- 29.7 Materialized Query Tables
- 29.8 Autonomic Features in DB2
- 29.9 Tools and Utilities
- 29.10 Concurrency Control and Recovery
- 29.11 System Architecture
- 29.12 Replication, Distribution, and External Data
- 29.13 Business Intelligence Features
- Bibliographical Notes

DB2 Replication, Distribution, and External Data

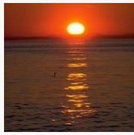
- **DB2 *Replication***
  - A product in the DB2 family that provides replication capabilities among DB2 other relational data sources such as Oracle, MS SQL Server, and other nonrelational data sources
- ***Queue Replication***
  - Creates a queue transport mechanism using IBM's message-queue product
- **DB2 Information-integrator product**
  - Provides federation, replication, and search capabilities
- **User-defined table functions**
  - Enable access to nonrelational and external data sources
  - **create function** statement with the clause **returns table**
- **Two-phase commit protocol**
  - Provides full supports for distributed transaction processing

Chapter 29: IBM DB2 Universal Database

- 29.1 Overview
- 29.2 Database-Design Tools
- 29.3 SQL Variations and Extensions
- 29.4 Storage and Indexing
- 29.5 Multidimensional Clustering
- 29.6 Query Processing and Optimization
- 29.7 Materialized Query Tables
- 29.8 Autonomic Features in DB2
- 29.9 Tools and Utilities
- 29.10 Concurrency Control and Recovery
- 29.11 System Architecture
- 29.12 Replication, Distribution, and External Data
- 29.13 Business Intelligence Features
- Bibliographical Notes

DB2 Business Intelligence Features

- **DB2 Data Warehouse Edition**
  - An offering that incorporates business intelligence features
  - Enhances the DB2 engine with features for ETL, OLAP, mining, and on-line reporting
- **On-line Analytical Processing (OLAP)**
  - *Cube views* provides a mechanism to construct appropriate data structures and MQTs
  - DB2 also provides multidimensional OLAP support using DB2 OLAP server
  - DB2 Alphablox is a new feature that provides on-line, interactive, reporting, and analysis capabilities
  - DB2 Intelligent Miner provides various components for modeling, scoring, and visualizing data



Chapter 29: IBM DB2 Universal Database

- 29.1 Overview
- 29.2 Database-Design Tools
- 29.3 SQL Variations and Extensions
- 29.4 Storage and Indexing
- 29.5 Multidimensional Clustering
- 29.6 Query Processing and Optimization
- 29.7 Materialized Query Tables
- 29.8 Autonomic Features in DB2
- 29.9 Tools and Utilities
- 29.10 Concurrency Control and Recovery
- 29.11 System Architecture
- 29.12 Replication, Distribution, and External Data
- 29.13 Business Intelligence Features
- Summary and Bibliographical Notes

Summary [1/2]

- This chapter provides a very brief synopsis of the features that are available in the DB2 Universal Database System
- In short, the DB2 Universal Database is multi-platform, scalable, object-relational database server
- DB2 Control Center includes various design- and administration-related tools
- DB2 provides support for a rich set of SQL features for various aspects of database processing
- The storage and indexing architecture in DB2 consists of the file-system or disk-management layer, the services to manage the buffer pools, data objects, and concurrency and recovery managers

Summary [2/2]

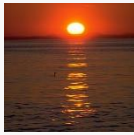
- DB2 provides multidimensional clustering (MDC)
- Materialized views are supported in DB2
- DB2 UDB version 8.2 provides features for simplifying the design and manageability of databases
- DB2 provides a number of tools for ease of use and administration
- DB2 supports a comprehensive set of concurrency-control, isolation, and recovery techniques
- DB2 Data Warehouse Edition is an offering in the DB2 family that incorporates business intelligence features

Bibliographical Notes [1/2]

- The origin of DB2 can be traced back to the System R Project (Chamberlin et al.[1981])

- IBM Research contributions include areas such as transaction processing (write-ahead logging and ARIES recovery algorithms) (Mohan et al.[1992]), query processing optimization (Starbust) (Haas et al.[1990]), parallel processing (DB2 Parallel Edition) (Baru et al.[1995]), active database support (constraints, triggers) (Cochrane et al.[1996]), advanced query and warehousing techniques such as materialized views (Zaharioudakis et al.[2000], Lehner et al.[2000]), multidimensional clustering (Padmanabahn et al.[2003], Bhattacharjee et al.[2003]), autonomic features (Zilio et al.[2004]) and object-relational support (ADTs, UDFs) (Carey et al.[1999])

- Multiprocessor-query-processing details can be found in Baru et al.[1995] or in the DB2 Administration and Performance guides DB2 Online documentation



Bibliographical Notes [2/2]

- Don Chamberlin’s books provide a good review of the SQL and programming features (Chamberlin[1996] and Chamberlin[1998])

- Earlier books by C. J. Date and others provide a good review of the features of DB2 Universal Database for OS/390 (Date[1989] and Martin et al.[1989])

- Recent book such as DB2 for Dummies (Zikopoulos et al.[2000]), DB2 SQL developer’s guide (Sanders[2000]), and the DB2 Administration Certification Guides (Cook et al.[1999]) provide hand-on training for using and administering DB2

- Chamberlin[1998], Zikopoulos et al.[2004] and the DB2 documentation library provide a complete description of the SQL support