DBMS
data
database
distributed
parallel
shared
system
architecture
DBMSs
memory
Distributed
DDBMS
processing
network
performance
systems
processors
access
centralized
multiple
sites
site
applications
computers
nothing
single
one
sometimes
processor
Parallel
may
major
fragments
typically
across
fact
require
physically
within
users
machines
like
Database
control
scalability
approach
improve
distinction
logical
reliability
eliminates
also
many
replicated
make
provide
local
discuss
often
number
dis
user
makes
per
application
management
transparent
Shared
databases
development
definition
nodes
Figure
interconnection
bottleneck
mode
administered
organizational
technology
capable
consists
logically
decentralized
directly
Date
com
disk
stored
computer
least
used
partitioned
global
way
incompatible
associated
provides
collection
structure
based
controlled
defined
remote
large
islands
resources
disks
architectures
scalable
order
split
computing
referred
operational
communication
distribution
must
System
Management

Database System Lab (CSE 3103)

Session 06

Nazmus Sakib, Assistant Professor, Dept. of CSE, AUST

Nowshin Nawar Arony, Adjunct Lecturer, Dept. of CSE, AUST

SQL JOIN OPERATION

- The JOIN keyword is used in an SQL statement to query data from two or more relationship between certain columns in these tables.

- INNER JOIN: Return rows when there is at least one match in both tables

- LEFT JOIN: Return all rows from the left table, even if there are no matches in the right table

- RIGHT JOIN: Return all rows from the right table, even if there are no matches in the left table

- FULL JOIN: Return rows when there is a match in one of the tables

INNER JOIN Keyword

- The INNER JOIN keyword return rows when there is at least one match in both tables.

- Syntax

```
SELECT column_name(s)
FROM table_name1
INNER JOIN table_name2
ON table_name1.column_name=table_name2.column_name
```

Example:

```
SELECT Customer.LastName, Customer.FirstName, CustomerOrder.Bill
FROM Customer INNER JOIN CustomerOrder
ON Customer.CustomerID=CustomerOrder.CustomerID
```

- The INNER JOIN keyword return rows when there is at least one match in both tables. rows in “Customer" that do not have matches in "CustomerOrder", those rows will NOT be listed.

LEFT JOIN Keyword

- The LEFT JOIN keyword returns all rows from left table (table_name1), even if there are no matches in the right table (table_name2).

- Syntax

```
SELECT column_name(s)
FROM table_name1
LEFT JOIN table_name2
ON table_name1.column_name=table_name2.column_name
```

Example:

```
SELECT Customer.LastName, Customer.FirstName, CustomerOrder.Bill
FROM Customer
LEFT JOIN CustomerOrder
ON Customer.CustomerID=CustomerOrder.CustomerID
```

The LEFT JOIN keyword returns all the rows from the left table (Customer), even if there are no matches in the right table (CustomerOrder).

- In some databases LEFT JOIN is called LEFT OUTER JOIN.

RIGHT JOIN Keyword

- The RIGHT JOIN keyword returns all rows from right table (table_name2), even if there are no matches in the left table (table_name1).

- Syntax

```
SELECT column_name(s)
FROM table_name1
RIGHT JOIN table_name2
ON table_name1.column_name=table_name2.column_name
```

Example:

```
SELECT Customer.LastName, Customer.FirstName, CustomerOrder.Bill
FROM Customer
RIGHT JOIN CustomerOrder
ON Customer.CustomerID=CustomerOrder.CustomerID
```

- The RIGHT JOIN keyword returns all the rows from the right table (CustomerOrder), even if there are no matches in the left table (Customer).

- In some databases RIGHT JOIN is called RIGHT OUTER JOIN.

FULL JOIN Keyword

- The FULL JOIN keyword return rows when there is a match in one of the tables.
- Syntax

```
SELECT column_name(s)
FROM table_name1
FULL JOIN table_name2
ON table_name1.column_name=table_name2.column_name
```

Example:

```
SELECT Customer.LastName, Customer.FirstName, CustomerOrder.Bill
FROM Customer
FULL JOIN CustomerOrder
ON Customer.CustomerID=CustomerOrder.CustomerID
ORDER BY Customer.LastName
```

- The FULL JOIN keyword returns all the rows from the left table (Customer), and all right table (CustomerOrder). If there are rows in "Customer" that do not have matches in "CustomerOrder", or if there are rows in "CustomerOrder" that do not have matches in "Customer", those rows will be listed as well.

UNION Operator

- The UNION operator is used to combine the result-set of two or more SELECT statements.

Conditions:

- Each Select Statement within the UNION must have the same number of columns
- Columns must also have similar datatypes
- Columns in each select statement must be in same order

Syntax

```
SELECT column_name(s) FROM table_name1
UNION
SELECT column_name(s) FROM table_name2
```

- UNION operator selects only distinct values by default. To allow duplicate values, use UNION ALL

```
SELECT column_name(s) FROM table_name1
UNION ALL
SELECT column_name(s) FROM table_name2
```

Example

```
SELECT FirstName, LastName FROM Customer_Online
UNION
SELECT  FirstName, LastName FROM Customer_Offline

SELECT FirstName, LastName FROM Customer_Online
UNION ALL
SELECT  FirstName, LastName FROM Customer_Offline
```

- It is not necessary that the corresponding columns in each SELECT statement have the same name, but they do need to be the same corresponding data types.

```
SELECT FirstName, LastName FROM Customer
UNION
SELECT  ClientFirstName, ClientLastName FROM Client
```


DBMS
data
database
distributed
parallel
shared
system
architecture
processing
network
performance
systems
processors
access
centralized
Distributed
DDBMS
DBMSs
memory
multiple
sites
site
single
nothing
Parallel
processor
fragments
may
major
typically
require
physically
within
users
machines
applications
computers
control
scalability
approach
improve
distinction
logical
Database
reliability
eliminates
distribution
communication
also
replicated
make
provide
local
sometimes
one
split
order
scalable
architectures
referred
operational
decentralized
directly
com
Date
large
islands
resources
disks
remote
defined
controlled
based
often
many
discuss
Management
number
dis
user
per
makes
logically
incompatible
way
global
structure
collection
provides
associated
stored
disk
computer
partitioned
used
least
consists
bottleneck
administered
organizational
interconnection
mode
Figure
application
management
transparent
technology
capable
nodes
definition
development
Shared
databases
share
must
System
computing
like
across
fact