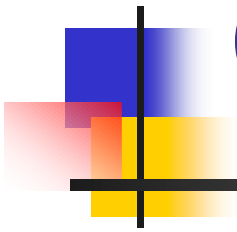


Distributed Databases in Oracle





Distribution in Oracle

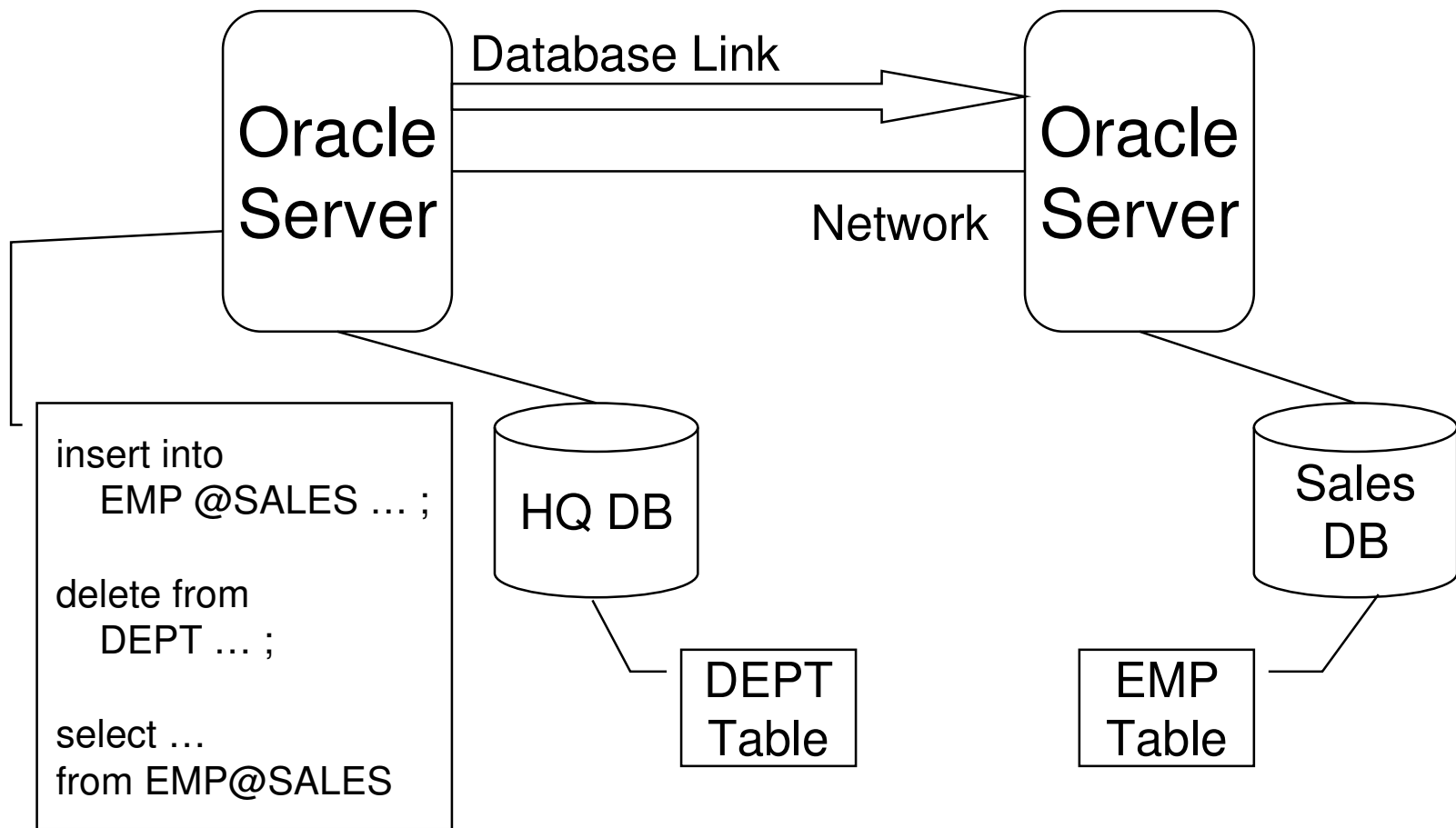
- Oracle supports distribution through a peer-to-peer architecture; there is no global schema.
- Oracle focuses on settings in which all participating databases are relational.
- Databases from different vendors can be included using either:
 - Product-specific gateways (e.g., by purchasing a gateway from Oracle to Sybase).
 - The generic connectivity services supported through ODBC or OLE DB.



Claimed Benefits of Local Autonomy

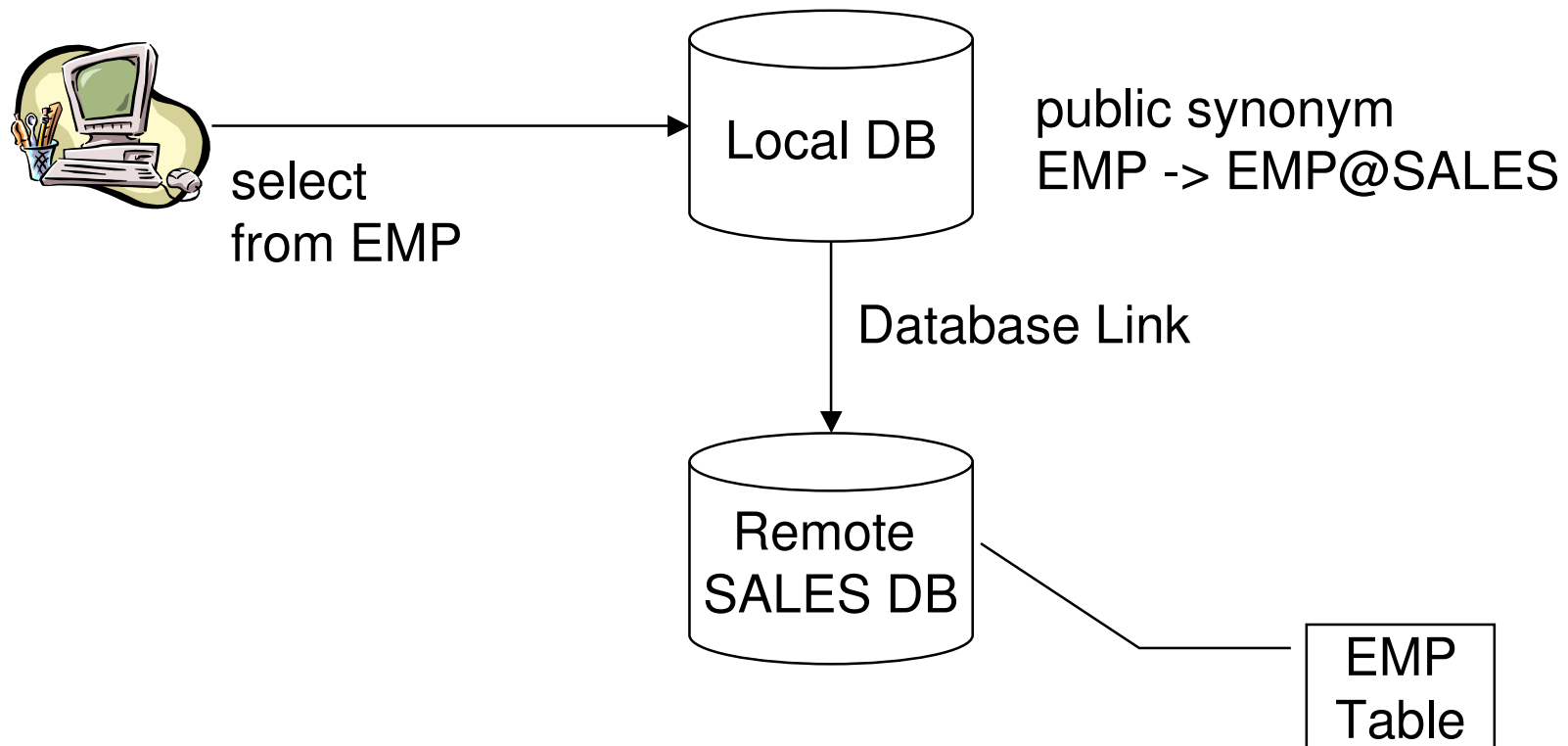
- More manageable database administration, as responsibilities are locally contained.
- No single point of failure; recovery from failures is local.
- No need to provide and maintain a global catalogue.
- Nodes can upgrade software independently.

Basic Architecture



Database Links

- A database link is a one-way communication path from one database server to another.





Oracle Names

- Links require globally unique names for database objects.
- Oracle name servers act as central repositories of information about each database in the system to ease the configuration of distributed database access.
- Each database in a distributed database has its own global database name.
- Oracle forms a database's global database name by prefixing the database's network domain with the individual database's name.



Features of Database Links - 1

- Links are either public or private:
 - A *private* link is specific to the user who created it.
 - A *public* link could be used by any user.



Features of Database Links - 2

- Links differ in the form of connection to the remote database that is created:
 - *Connected user links*: Users connect to the remote database as themselves, and thus must have an account on the remote database.
 - *Fixed user links*: Users connect using a username and password that is a property of the link.



Creating Database Links

- Connected user link, where *sales_service* is the network service to be used:
 - create private database link
SALES.EUROPE.ACME.COM using 'sales_service'.
- Fixed user link:
 - create public database link
SALES.EUROPE.ACME.COM connect to scott identified by tiger using 'sales_service'.



Remote Queries

- A remote query is a query that selects information from one or more remote tables, all of which reside at the same remote node.

```
SELECT *  
FROM scott.dept@sales.us.americas.acme_auto.com;
```



Distributed Queries

- A distributed query retrieves information from two or more nodes.

```
SELECT ename, dname
FROM scott.emp@sales.uk.europe.acme.com e,
     scott.dept@sales.us.americas.acme.com d
WHERE e.deptno = d.deptno;
```



Statement Types

- Standard DML statements can reference remote tables.

SELECT	Queries including joins, aggregates, subqueries, and SELECT ... FOR UPDATE can reference any number of local and remote tables and views.
INSERT UPDATE DELETE	UPDATE, INSERT and DELETE statements can reference both local and remote tables. No programming is necessary to update remote data.



Remote Update

- A remote update is an update that modifies data in one or more tables, all of which are located at the same remote node.

```
UPDATE scott.dept@sales.us.americas.acme_auto.com  
SET loc = 'NEW YORK' WHERE deptno = 10;
```



Distributed Update

- A distributed update is a block that modifies data on two or more nodes.
- Statements in the block are sent to the remote nodes, and execution succeeds or fails as a unit.

```
BEGIN
  UPDATE scott.dept@sales.us.americas.acme_auto.com
  SET loc = 'NEW YORK' WHERE deptno = 10;
  UPDATE scott.emp@sales.uk.europe.acme.com
  SET deptno = 11 WHERE deptno = 10;
END;
```



Synonyms

- **Synonyms provide location transparency.**

```
CREATE PUBLIC SYNONYM emp
FOR scott.emp@sales.us.americas.acme_auto.com
CREATE PUBLIC SYNONYM dept
FOR scott.dept@sales.us.americas.acme_auto.com

SELECT ename, dname
FROM scott.emp@sales.us.americas.acme_auto.com e,
      scott.dept@sales.us.americas.acme_auto.com d
WHERE e.deptno = d.deptno;

SELECT ename, dname FROM emp e, dept d
WHERE e.deptno = d.deptno;
```



Views for Location Transparency

- EMP is stored in a local database. DEPT, is stored in a remote database.
- A view named COMPANY can be created in the local database that joins the data of the local and remote servers:

```
CREATE VIEW company AS
SELECT empno, ename, dname
FROM emp a, dept@hq.acme.com b
WHERE a.deptno = b.deptno;

SELECT * FROM company;
```



Procedures for Location Transparency

- For example:
 - A local synonym can reference a remote procedure.
 - A local procedure can reference remote data.

```
CREATE PROCEDURE fire_emp (enum NUMBER) AS
BEGIN
    DELETE FROM emp@hq.acme.com
    WHERE empno = enum;
END;
```



Remote Procedure Calls

- Stored procedures and functions can also be invoked directly.

```
BEGIN
    emp_mgmt.del_emp@sales.acme.com(1257);
END;
```



Example To Try

- Given the following Databases:
- HQDB:
 - Employee(enum, name)
 - Worksfor(enum, project)
- BuildingDB:
 - Location(project, location)
 - Emp(enum, name, project)
- Create a view at HQDB that relates every employee from HQDB or BuildingDB to his/her location, and give the database link that is required to enable to view to be defined:
 - WorksAt(ename, location)



Summary on Distribution in Oracle

- Oracle provides comprehensive distributed access and update facilities.
- Distribution in Oracle builds on the low level notion of a directed link between databases.
- Link based distribution seems more in line with bottom-up than top-down distributed database development.
- Full support for distributed transactions, and flexible access control are prominent features.



Further Reading

- W. Kim and J. Seo, Classifying Schematic and Data Heterogeneity in Multidatabase Systems, IEEE Computer, 12-18, December, 1991.
- Oracle Database Administrator's Guide, Chapter 29: Distributed Database Concepts.