MASSACHUSETTS INSTITUTE OF TECHNOLOGY
SLOAN SCHOOL OF MANAGEMENT

15.565 Integrating Information Systems:

Technology, Strategy, and Organizational Factors

15.578 Global Information Systems:

Communications & Connectivity Among Information Systems

Spring 2002

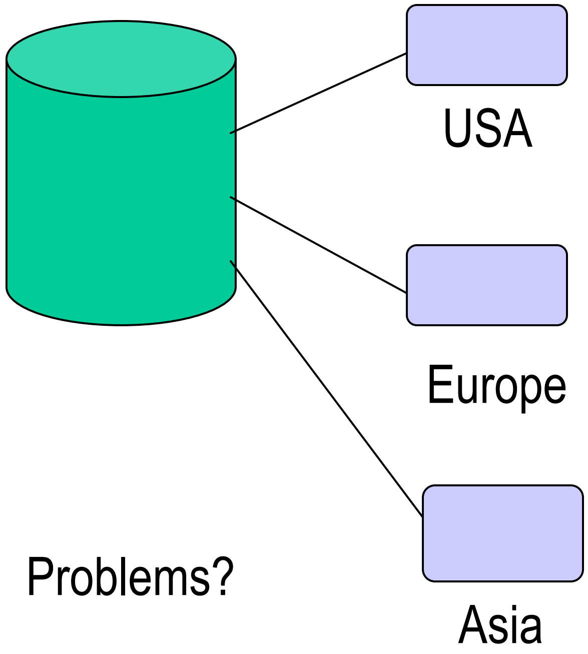
Lecture 13

DISTRIBUTED HOMOGENOUS DATABASES

CENTRALIZE OR DECENTRALIZE ?

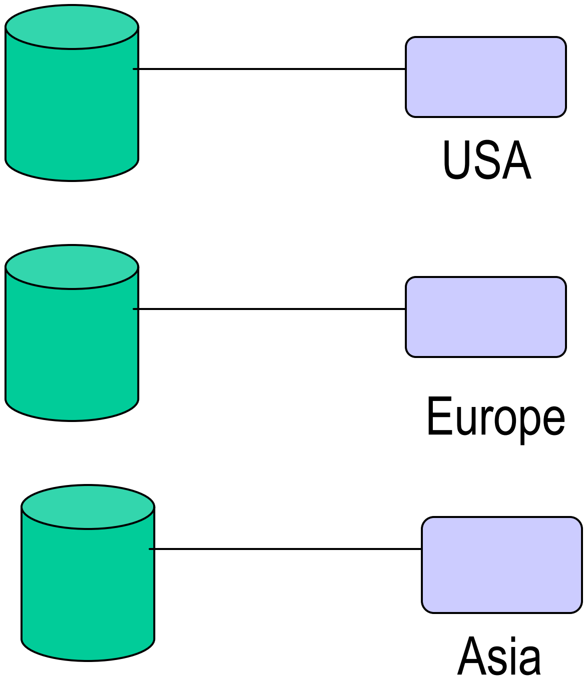
Common design challenge -- Consider account balance database(s) for international bank or foreign exchange operation. What do you do?

Centralized database



Problems?

Decentralized databases



NEXT GENERATION GLOBAL FOREIGN EXCHANGE SYSTEM (FXX) -- REVISITED

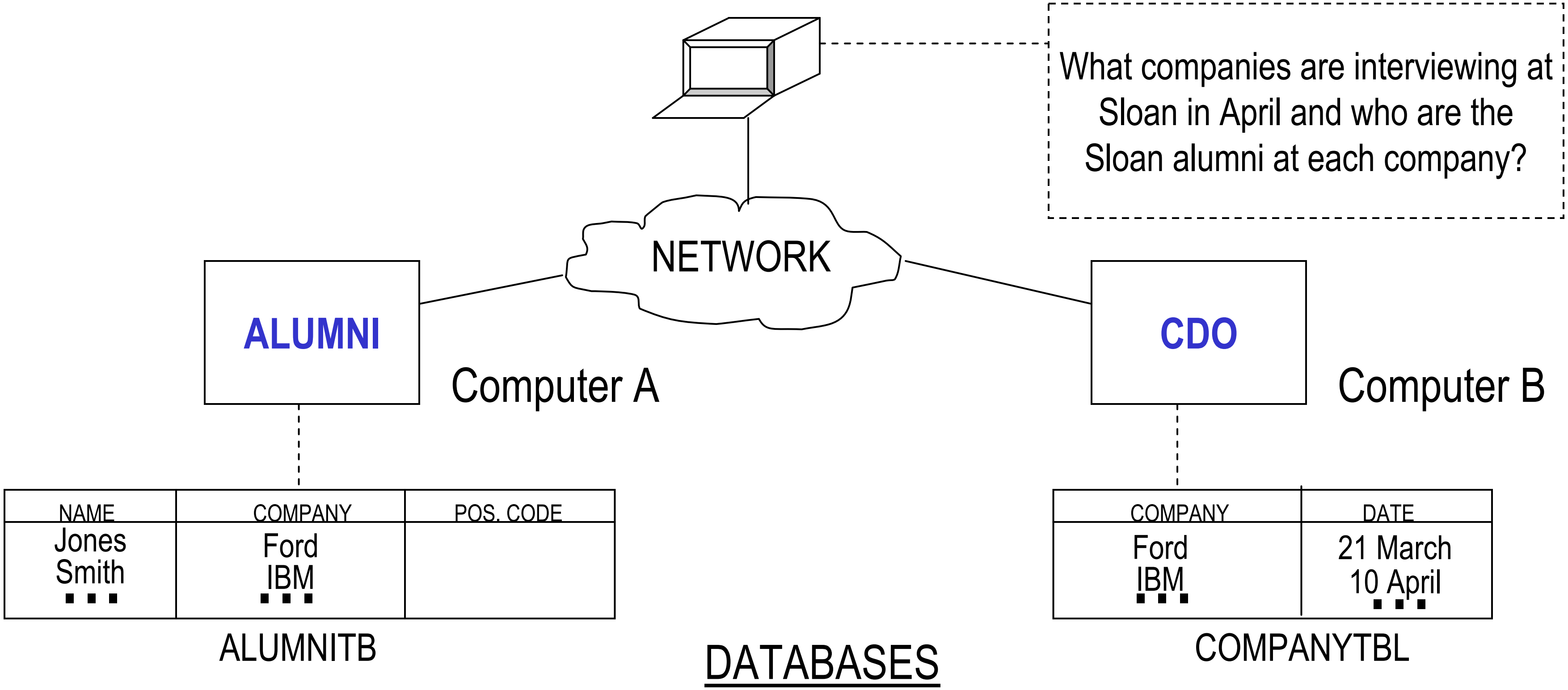
Global Database

The *Global Database* contains a *Replication* of information from each of the Branch and Regional servers based on Local Time Zone and GMT. The Global Database is continuously updated by branches and regions worldwide on an asynchronous, real time basis. The nominal delay attributed to transmission time will not exceed one minute. All tier 1 cities will be linked via *ISDN* to handle network traffic overflow and any network failures.

The data in the Global Database is time stamped to GMT time and the Local time of the branch from which data was received. Time stamping data to GMT time supports query consistency across multiple time zones. Time stamping data to Local time supports per site queries.

The Global Database will include all region/branch deals, End of Day P&L, daily risk figures including Money-at-Risk, partner, and credit information. Global Business Managers and senior management can use the Global Database to manage, monitor, and analyze positions, P&L, risk, partner activity, credit limits, brokerage, margins, and the Global FX Order Manager. Any third party *Executive Information System* ("EIS") tools can be used against the Global Database.

DISTRIBUTED DATABASE -- APPEAR AS CENTRALIZED DATABASE



What companies are interviewing at Sloan in April and who are the Sloan alumni at each company?

NAME	COMPANY	POS. CODE
Jones Smith ■ ■ ■	Ford IBM ■ ■ ■	

ALUMNITB

COMPANY	DATE
Ford IBM ■ ■ ■	21 March 10 April ■ ■ ■

COMPANYTBL

DATABASES

PROBLEMS:

1. WHAT KNOWLEDGE NEEDED?

2. GET WHICH DATA FROM WHERE?

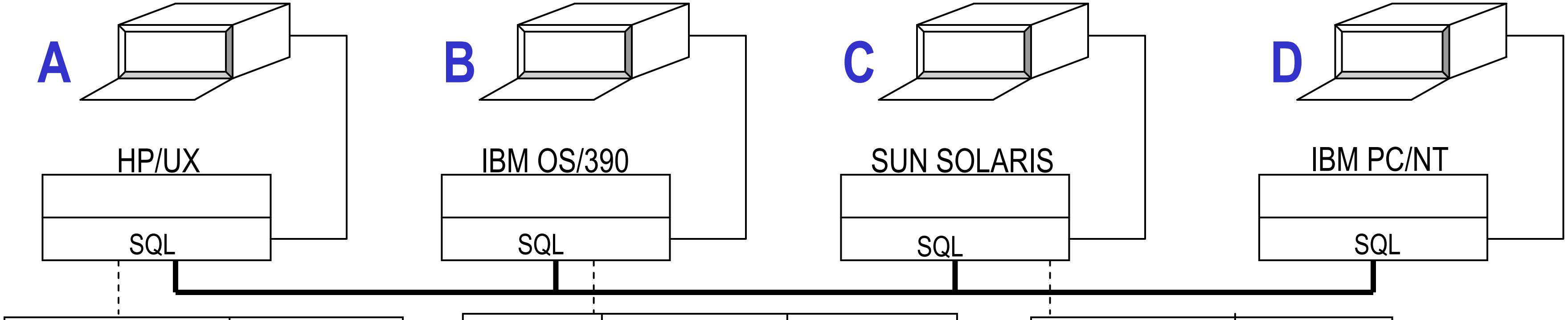
3. HOW TIE DATA TOGETHER?

4. COPY TO WHERE? WHERE IS PROCESSING DONE?

5. HOW MAKE THIS **TRANSPARENT** TO USER AND DEVELOPER?

DISTRIBUTED **HOMOGENOUS** DATABASE SYSTEMS

(e.g., ORACLE/STAR and DISTRIBUTED INGRES -- SAME DBMS ON ALL SYSTEMS)



POS. CODE	POSITION

POSITIONTB **(p)**

NAME	COMPANY	POS. CODE
John	IBM	
Fred	IBM	
Larry	Dell	

ALUMNITB **(a)**

COMPANY	DATE
IBM	April
Dell	April

COMPANYTB **(c)**

DATABASES
(SERVERS)

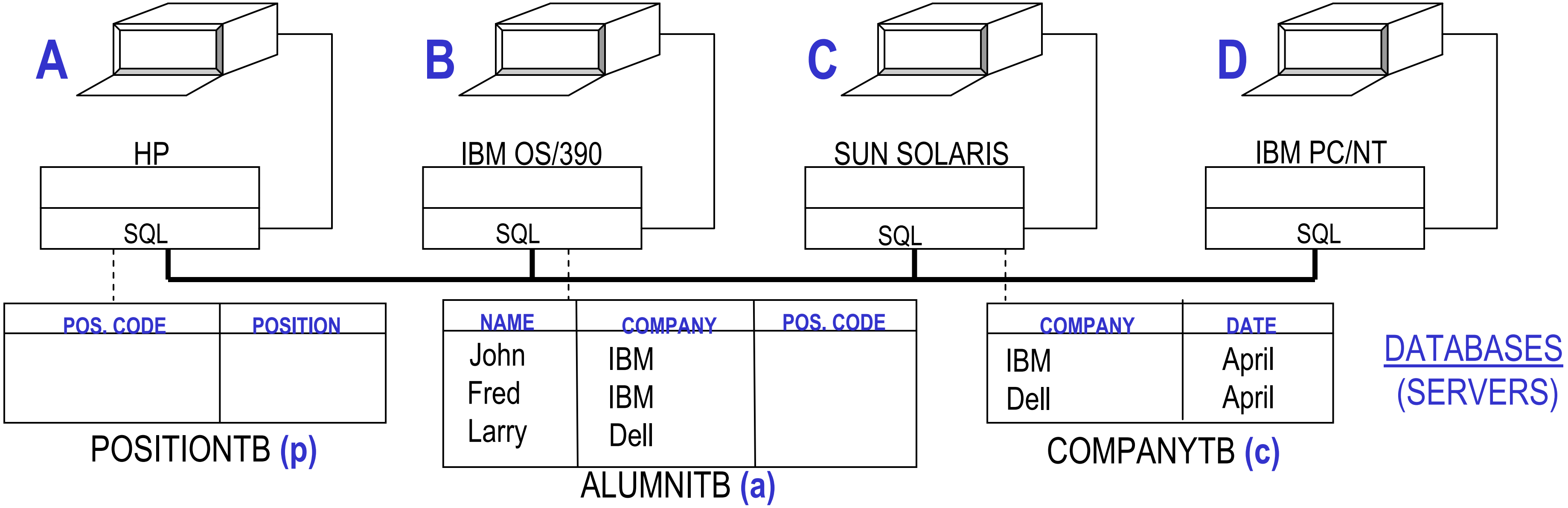
```
Select name, company, date
  from a, c
  where date = ‘April’
  and a.company = c.company
```

TRANSPARENCY GOALS:

1. **RETRIEVAL** -- SAME RESULTS FROM ANY SITE

2. **UPDATE** -- UPDATE FROM ANY SITE
   TO A SINGLE TABLE

DISTRIBUTED **HOMOGENOUS** DATABASE SYSTEMS



POS. CODE	POSITION

POSITIONTB **(p)**

NAME	COMPANY	POS. CODE
John	IBM	
Fred	IBM	
Larry	Dell	

ALUMNITB **(a)**

COMPANY	DATE
IBM	April
Dell	April

COMPANYTB **(c)**

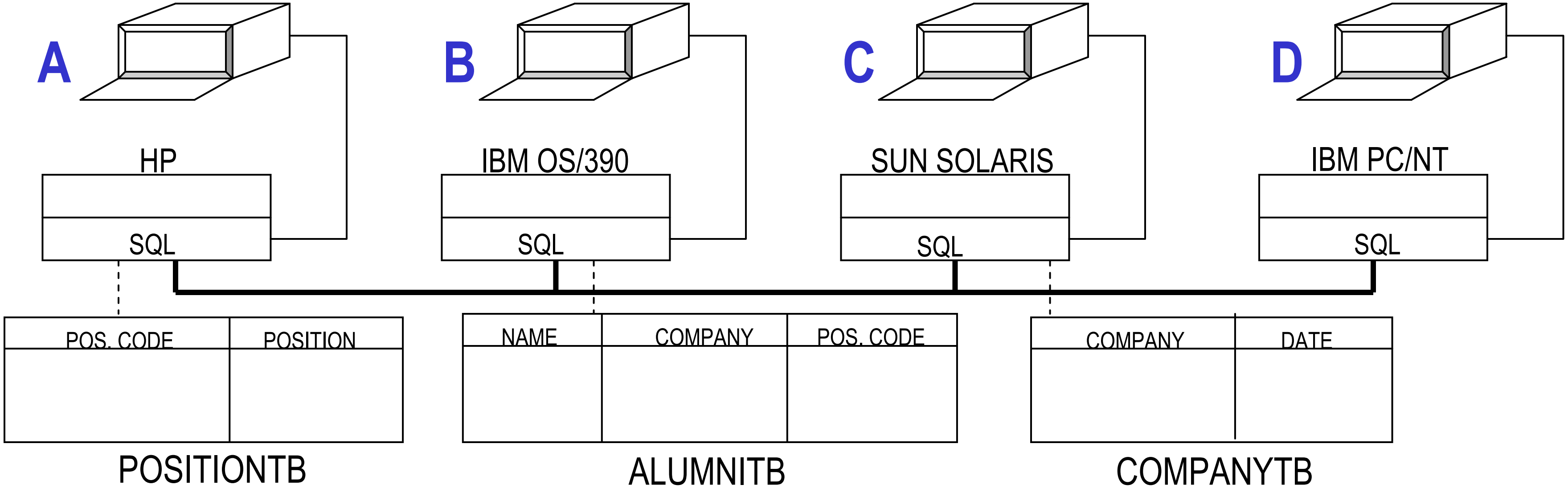
TRANSPARENCY GOALS:

3. **SCHEMA** – CHANGE IN SCHEMA VISIBLE AT ANY SITE (AUTOMATICALLY)

4. **PERFORMANCE OPTIMIZATION** – SAME (BEST) SPEED FROM ANY SITE

```
Select name, company, date
  from a, c
  where date = ‘April’
  and a.company = c.company
```

DISTRIBUTED **HOMOGENOUS** DATABASE SYSTEMS



POS. CODE	POSITION

POSITIONTB

NAME	COMPANY	POS. CODE

ALUMNITB

COMPANY	DATE

COMPANYTB

DATABASES (SERVERS)

TRANSPARENCY GOALS:

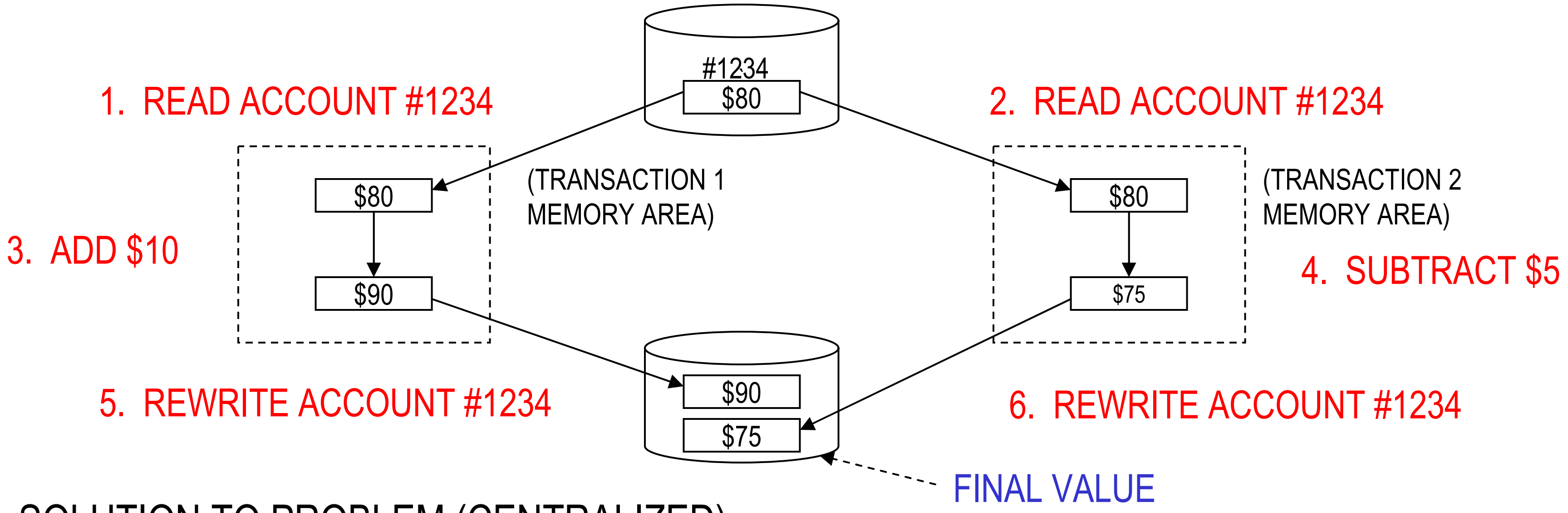
5. **TRANSACTION** -- MULTI-UPDATE DONE CORRECTLY (*SEE NEXT SLIDE*)

6. **COPY** -- REDUNDANT COPIES MAINTAINED AND USED EFFICIENTLY.

DIFFICULTY WITH UPDATE AND COPY TRANSPARENCY

- "CONCURRENCY CONTROL" IN CENTRAL DATABASE -- EXAMPLE:

TRANSACTION 1	TRANSACTION 2
ADD $10 TO ACCOUNT #1234	SUBTRACT $5 FROM ACCOUNT #1234



1. READ ACCOUNT #1234

2. READ ACCOUNT #1234

3. ADD $10

4. SUBTRACT $5

5. REWRITE ACCOUNT #1234

6. REWRITE ACCOUNT #1234

-- SOLUTION TO PROBLEM (CENTRALIZED)

- COORDINATE THE TRANSACTION THROUGH USE OF LOCKS
- MULTIPLE LOCKS REQUIRED DURING TRANSACTION TO COORDINATE ALL DATA ELEMENTS NEEDED (READ AND WRITE)
- DEADLOCK POSSIBILITIES EXIST

ADDITIONAL CONCURRENCY CONTROL PROBLEMS

- IN DISTRIBUTED DATABASES WITH GENERAL UPDATE AND COPY TRANSPARENCY
  - COMPLEXITY OF COORDINATING **DISTRIBUTED LOCKS**
  - **COMMUNICATION OVERHEAD** OF LOCK SETTING (DON'T KNOW IF LOCK REALLY NEEDED)
  - DIFFICULTY OF **DEADLOCK DETECTION**
  - REPLICATED COPIES ADD:
    - UPDATING OVERHEAD (ALL COPIES MUST BE UPDATED)
    - LOCKING OVERHEAD (ALL COPIES MUST BE LOCKED)
    - PERFORMANCE COMPLEXITY (MANY MORE SEQUENCES)
  - DANGER OF SITE FAILURE
    - WHO RESETS LOCK THEN?
  - PARTITIONING AND REINTEGRATION OF NETWORK
- HUNDREDS OF RESEARCH ARTICLES WRITTEN ON THIS SUBJECT

DESIRABLE INDEPENDENCE PROPERTIES

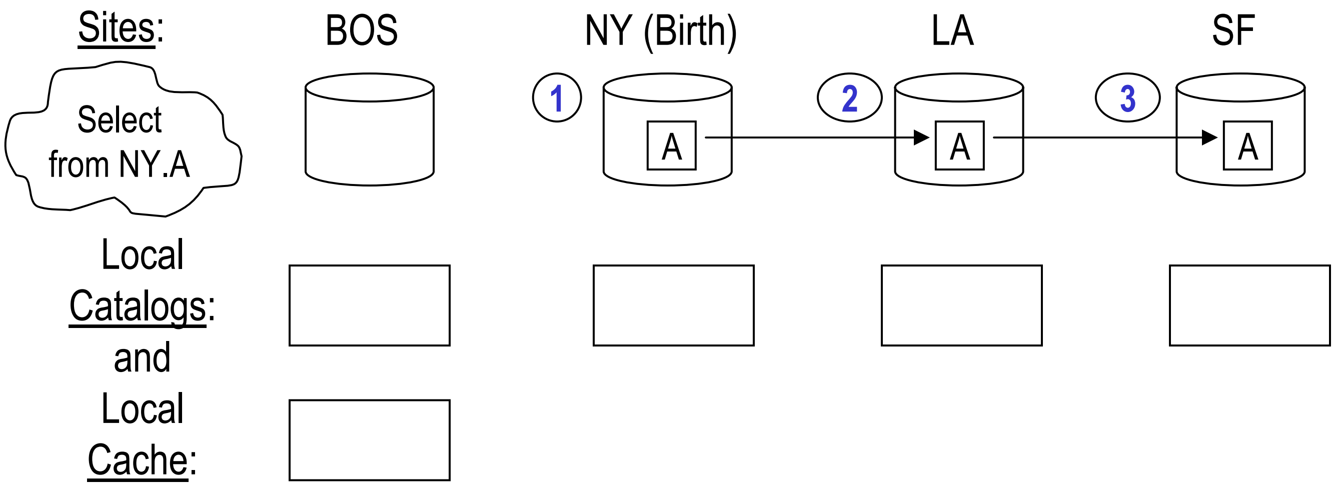
1. **CRASH**
   – ONLY AFFECT “CRASHED” SITE
2. **RECOVERY**
   – RECOVER AUTOMATICALLY
3. **NETWORK**
   – OPERATE OVER ANY AND ALL NETWORKS
4. **HARDWARE/OS**
   – RUN ON RANGE OF EQUIPMENT
5. **SQL VENDOR**
   – ALLOW MULTIPLE SQL VENDORS (E.G., DB2, ORACLE)
6. **DBMS**
   – SUPPORT OTHER DBMS MODELS (HIERARCHICAL, NETWORK) AND LANGUAGES

IBM R* ISSUES AND TECHNIQUES

- DISTRIBUTED CATALOG MANAGEMENT
- REMOTE CATALOG ACCESS
- DISTRIBUTED DATA DEFINITION
- **TABLE MIGRATION**
- DISTRIBUTED QUERY PLANNING
- QUERY DISTRIBUTION
- DISTRIBUTED COMMIT PROTOCOLS
- DISTRIBUTED DEADLOCK DETECTION

R* CATALOG MANAGEMENT AND ALTERNATIVES

- CENTRAL CATALOG (INGRES)
- REPLICATED CATALOGS (ORACLE)
- LOCAL CATALOGS (R*)
  - IDENTIFY “BIRTH” SITE IN REQUEST
  - “CACHES” INFORMATION TO REDUCE REMOTE CATALOG ACCESS
  - CATALOG VERSION # USED TO VALIDATE CORRECTNESS



SOME ADDITIONAL DIFFICULT ISSUES

- VERTICALLY PARTITIONED DATABASES

DB1:

Name	Address

DB2:

Name	Phone

- HORIZONTALLY PARTITIONED DATABASES

EAST DB:

Name	Address

WEST DB:

Name	Address

SUMMARY

- DISTRIBUTED HOMOGENOUS DATABASE MANAGEMENT SYSTEMS ARE A REALITY

- STILL LIMITED PRACTICAL EXPERIENCE THUS FAR

- ALSO NEED TO DEAL WITH DIVERSE ENVIRONMENT AND MANY VARIATIONS FOR QUITE A WHILE (HETEROGENEOUS DBMS)