Recap

- Database:
  - collection of data central to some enterprise that is managed by a Database Management System
  - reflection of the current state of the enterprise
  - constantly changes

1

Transaction

- Programs that execute to update the information stored in a database
  - Deposit money into bank account
  - Register for a course
  - Get approval for credit card use

2

Transaction Manager

- Responsible for the consistency of database
  - changes in the real-world are reflected *correctly* in the database
  - every time a real-world event happens, a *transaction* occurs to cause the corresponding changes in the database
- **Definition**: A transaction is an application program with special properties – see next slides – to guarantee it maintains database correctness

3

Properties of Transactions (ACID)

- **A**tomicity: ALL-or-NOTHING execution (a sequence of primitive commands that needs to be executed ALL or NONE). **Deposit money – the bank takes it but does not increase the balance**
- **I**solation: No two transactions should be executed at the same time. **Two withdrawals cannot be done at the same time**
- **D**urability: Effects of a transaction can never be lost **After my deposit, the balance should stay the same until I withdraw some money**
- **C**onsistency: Constraints are satisfied all the time **No more than 300 withdraw at ATM**

4

Transaction Manager

- Log manager: every change in the database is logged separately on the disk (for *recovery* or *durability*)
- Concurrency-control manager: for *isolation* (uses *lock*, similar to *lock* in OS)
- Deadlock resolution: resources control

5

What should be stored in a database?

Examples of databases

- Airline reservation system
- Banking system
- Student registration system
- Supermarket
- Corporate record
- ….

7

Airline reservation system

- Data: Information about flights
  - Flight number, type of aircraft
  - Date, time, departure airport, arrival airport
  - Number of seats (1st, 2nd class if applicable)
  - Lists of travelers, their reservation
  - Ticket prices, number of available seats
- Operations (Queries/Transactions):
  - Customer inquires about the availability of a flight, ticket for a flight
  - Customer makes a reservation
  - Customer cancels a reservation
- Properties:
  - Large number of transactions (*very* frequently)
  - Cannot be processed in batch mode (*on-line* transaction processing)
  - Concurrency required

8

Banking system

- Data: Account information
  - Customer information (name, address, accounts, balances)
  - Relationship between customers and accounts
- Operations (Queries/Transactions):
  - Customer inquires about the balance of one of its accounts
  - Customer makes a deposit
  - Customer withdraws
- Properties:
  - Large number of transactions (*very* frequently)
  - Cannot be processed in batch mode (*on-line* transaction processing)
  - Concurrency required
  - Recovering from failures

9

Student Registration System

- Data: Information about students and courses
  - Student information (name, address, SSN, status, major, minor, courses taken and grade, courses enrolled, balance, picture)
  - Course information (name, call number, number, credit hours, department, instructor, date and time, location, number of students)
- Operations (Queries/Transactions):
  - Students ask for a transcript, list of enrolled classes
  - Adding/Dropping classes
  - Prerequisites enforcement
- Properties:
  - Large number of transactions at the beginning and end of semester
  - Batch mode processing possible (better with *on-line* transaction processing)
  - Concurrency required

10

Databases (Now vs. Then)

- Relational model using SQL - high-level view of data
  - Older systems presented low-level view
- Might contain multimedia data
  - Older systems restricted to alphanumeric data
- On-line: database accessed at time of event
  - Older systems were off-line, batch

11

Databases (Now vs. Then)

- Concurrent - multiple transactions execute simultaneously
  - Older systems processed transactions sequentially
- Distributed computation - different parts of the application execute on different computers
  - Older systems were centralized

12

Databases (Now vs. Then)

- Distributed data - different parts of the data are stored in different databases on different computers
  - Older systems were centralized
- Heterogeneous - involves HW and SW modules from different manufacturers
  - Older systems were homogeneous
- Accessed by everyone (e.g., e-commerce)
  - Older systems restricted to trained personnel

13

Database (System) Requirements

Database (System) Requirements

- **High Availability**: on-line => must be operational while enterprise is functioning
- **High Reliability**: correctly tracks state, does not lose data, controlled concurrency
- **High Throughput**: many users => many transactions/sec
- **Low Response Time**: on-line => users are waiting

15

Requirements (cont.)

- **Long Lifetime**: complex systems are not easily replaced
  - Must be designed so they can be easily extended as the needs of the enterprise change
- **Security**: sensitive information must be carefully protected since system is accessible to many users
  - Authentication, authorization, encryption

16

People in Design, Implementation, and Maintenance of a Database

- **System Analyst** - specifies system using input from customer; provides complete description of functionality from customer's and user's point of view
- **Database Designer** - specifies structure of data that will be stored in database
- **Application Programmer** - implements application programs (transactions) that access data and support enterprise rules

17

People (cont.)

- **Database Administrator** - maintains database once system is operational: space allocation, performance optimization, database security
- **System Administrator** - maintains transaction processing system: monitors interconnection of HW and SW modules, deals with failures and congestion

18

Database System Studies

Design of databases

- **how** to design a database
- **what** should be stored
- **which** structure for the data
- **what** assumptions should be made
- **how** is the connection between data

20

Database programming

- how to write queries on the database
- how to use other capabilities of a DBMS in an application
- how is database programming combined with conventional programming

21

Database System Implementation

- how to build a DBMS (query processing, transaction processing, storage manager etc.)

This will not be discussed in this course.

22

Application of Database

Decision Support System (OLTP vs. OLAP)

- **On-line Transaction Processing** (OLTP)
  - Day-to-day handling of transactions that result from enterprise operation
  - Maintains correspondence between database state and enterprise state
- **On-line Analytic Processing** (OLAP)
  - Analysis of information in a database for the purpose of making management decisions

24

On-Line Analytical Processing

- Analyzes historical data (terabytes) using complex queries
- Due to volume of data and complexity of queries, OLAP often uses a data warehouse
- **Data Warehouse -** (offline) repository of historical data generated from OLTP or other sources
- **Data Mining** - use of warehouse data to *discover* relationships that might influence enterprise strategy

25

Exp – Airline reservation system

- OLTP
  - Event: customer A books tickets from ELP to NY; update database to reflect that event
- OLAP
  - During the last holiday season, how many customers fly from ELP to Dallas and NY?
- Data Mining
  - Are there any airports in which more than 50% of travelers from ELP need to change their flight?

26