

Introduction

Tom Kelliher, CS 318

Jan. 23, 2002

1 Administrivia

Announcements

Assignment

Read Chapters 1 and 2.

Outline

1. Syllabus.
2. Introduction.
3. A closer look: Student Registration System, relational databases, properties of transactions.

Coming Up

Introduction to HTML, PHP, and Unix.

2 Introduction

What is:

1. a database?

2. a DBMS?

Examples: Oracle, Sybase, MS SQL Server, mySQL, PostgreSQL.

3. SQL?

4. a Transaction? (Changes state of database.)

5. a Transaction Processing System? (Transactions, TP monitor, DBMS, database.)

Potentially multiple distributed databases on heterogeneous platforms.

Database characteristics. Envision an airline reservations system.

1. Availability.

2. Reliability.

3. Throughput.

4. Response time.

5. Lifetime.

6. Security.

3 A Closer Look

Common example: Student Registration System. Faculty and/or students can:

1. Authenticate.

2. Register for courses for next semester.
3. Obtain student status reports.
4. Maintain information about students and courses.
5. Enter final grades for completed courses.

This is a starting point — additional transactions needed. Also constraints.

3.1 Relational Databases

1. Table, relation.

View as a predicate — a statement of truth. A set.

2. Row, tuple.

An ordered n-tuple.

3. Column, attribute.

Properties of tuples.

4. Domain of an attribute.

Example relations:

```
CREATE TABLE "student" (  
  "id" integer NOT NULL,  
  "name" character(20) NOT NULL,  
  "address" character(50),  
  "status" character(10) DEFAULT 'Freshman',  
  Constraint "stu_key" Primary Key ("id")  
);
```

```
CREATE TABLE "transcript" (  
  "stuid" integer,  
  "crscode" character(6),
```

```

    "semester" character(6),
    "grade" character(1),
    CONSTRAINT "gradecheck" CHECK ( grade in ('A', 'B', 'C', 'D', 'F')),
    CONSTRAINT "stuidcheck" CHECK (stuid > 0 AND stuid < 1000000000)
);

```

psql demo and sample queries:

```

-- Get name of student with particular Id #.
select Name
from Student
where Id = '987654321';

```

```

-- Get Id and Name of all seniors.
select Id, Name
from Student
where Status = 'Senior';

```

```

-- Get Name, Course, and Grade for all seniors.
-- Must match tuples (join) between two relations.
select Name, CrsCode, Grade
from Student, Transcript
where StuId = Id and Status = 'Senior';

```

3.2 Properties of Transactions

Some integrity constraints for Student Registration System:

1. Student Ids are unique.
2. Students must satisfy course prerequisites before registering for a course.
3. The number of students registered for a course cannot exceed the course cap.
4. Suppose there are two ways to count the number of students registered for a course (aggregate on Transcript relation and attribute of Courses relation). These two ways of counting must yield the same result.

A transaction's ACID properties:

1. Atomicity: All or nothing.
2. Consistency: Integrity constraints are preserved.

Transaction designer assumes database is initially consistent.

3. Isolation: Consider multiple simultaneous transactions. What bad things can happen?

(a) Serial execution.

(b) Transaction schedules: serial, concurrent.

(c) Serializable concurrent schedules.

(d) Isolation definition: *Even though transactions are executed concurrently, the overall effect of the schedule must be the same as if the transactions had executed serially in some order..*

4. Durability: Once a transactions commits, its results are permanent.