

**SQL: Part I**

CPS 116  
Introduction to Database Systems

---

---

---

---

---

---

---

---

**Announcements (September 13)**

- ◆ Homework #1 due next Tuesday
  - Do we need a help session on Monday?
- ◆ Course project assigned today
  - Choice of "standard" or "open"
  - One- to three-person teams
  - Two milestones + demo/report
  - Milestone #1 due in 4 weeks, right after fall break

---

---

---

---

---

---

---

---

**SQL**

- ◆ SQL: Structured Query Language
  - Pronounced "S-Q-L" or "sequel"
  - The standard query language supported by most commercial DBMS
- ◆ A brief history
  - IBM System R
  - ANSI SQL89
  - ANSI SQL92 (SQL2)
  - ANSI SQL99 (SQL3)
  - ANSI SQL 2003 (+OLAP, XML, etc.)

---

---

---

---

---

---

---

---

## Creating and dropping tables

❖ CREATE TABLE *table\_name*  
(..., *column\_name*, *column\_type*, ...);

❖ DROP TABLE *table\_name*;

❖ Examples

```
create table Student (SID integer,  
                    name varchar(30), email varchar(30),  
                    age integer, GPA float);  
create table Course (CID char(10), title varchar(100));  
create table Enroll (SID integer, CID char(10));  
drop table Student;  
drop table Course;  
drop table Enroll;  
-- everything from -- to the end of the line is ignored.  
-- SQL is insensitive to white space.  
-- SQL is insensitive to case (e.g., ...Course... is equivalent to  
-- ...COURSE...)
```

---

---

---

---

---

---

---

---

## Basic queries: SFW statement

❖ SELECT  $A_1, A_2, \dots, A_n$   
FROM  $R_1, R_2, \dots, R_m$   
WHERE *condition*;

❖ Also called an SPJ (select-project-join) query

❖ Equivalent (not really!) to relational algebra query

$$\pi_{A_1, A_2, \dots, A_n} (\sigma_{condition} (R_1 \times R_2 \times \dots \times R_m))$$

---

---

---

---

---

---

---

---

## Example: reading a table

❖ SELECT \* FROM Student;

- Single-table query, so no cross product here
- WHERE clause is optional
- \* is a short hand for "all columns"

---

---

---

---

---

---

---

---

### Example: selection and projection

- ◆ Name of students under 18
  - `SELECT name FROM Student WHERE age < 18;`
- ◆ When was Lisa born?
  - `SELECT 2007 - age  
FROM Student  
WHERE name = 'Lisa';`
  - SELECT list can contain expressions
    - Can also use built-in functions such as SUBSTR, ABS, etc.
  - String literals (case sensitive) are enclosed in single quotes

---

---

---

---

---

---

---

---

### Example: join

- ◆ SID's and names of students taking courses with the word "Database" in their titles
  - `SELECT Student.SID, Student.name  
FROM Student, Enroll, Course  
WHERE Student.SID = Enroll.SID  
AND Enroll.CID = Course.CID  
AND title LIKE '%Database%';`
  - LIKE matches a string against a pattern
    - % matches any sequence of 0 or more characters
  - Okay to omit `table_name` in `table_name.column_name` if `column_name` is unique

---

---

---

---

---

---

---

---

### Example: rename

- ◆ SID's of all pairs of classmates
  - Relational algebra query:  
 $\pi_{1.SID, 2.SID} (\rho_{1,1} Enroll \bowtie_{1.CID = 2.CID \wedge 1.SID > 2.SID} \rho_{1,2} Enroll)$
  - SQL:  
`SELECT e1.SID AS SID1, e2.SID AS SID2  
FROM Enroll AS e1, Enroll AS e2  
WHERE e1.CID = e2.CID  
AND e1.SID > e2.SID;`
  - AS keyword is completely optional

---

---

---

---

---

---

---

---