

Lecture 28: Declarative Programming: SQL

- So far, our programs are explicit directions for solving a problem; the problem itself is *implicit* in the program.
- **Declarative** programming turns this around:
 - A "program" is a description of the desired characteristics of a solution.
 - It is up to the system to figure out how to achieve these characteristics.
- Example: Spreadsheets contain formulae indicating what value a cell contains, but they say nothing about the order in which calculate these values or how to keep them up-to-date with changes in the data.
- Example (somewhat impure): SQL (Structured Query Language).
- SQL is a widely used notation for interrogating and modifying **database management systems** (DBMSs).

Last modified: Fri Apr 7 15:04:11 2017

CS66A, Lecture #28 1

Relational Databases

- A DBMS is a collection of data. The kind of DBMS accessed by SQL is *relational*.
- In mathematics, a *relation* is a set of tuples that represent values that *stand in some relationship* to one another.
- In a relational DBMS, relations take the form of **tables** with labeled columns. Each entry (tuple) is called a *row*.

Table Name: students

SID	Last	First	SemEnt	YearEnt	Major	Column Titles ← Column Titles
101	Knowles	Jason	F	2003	EECS	
102	Chan	Valerie	S	2003	Math	
103	Xavier	Jonathan	S	2004	LSUnd	← Row
104	Armstrong	Thomas	F	2003	EECS	
105	Brown	Shana	S	2004	EECS	
106	Chan	Yangfan	F	2003	LSUnd	

↑
Column

Last modified: Fri Apr 7 15:04:11 2017

CS66A, Lecture #28 2

Defining a Table in SQL

```
grades
SID | CCN | Grade
101 | 21228 | B
102 | 21231 | A
101 | 21105 | B+
106 | 21001 | B
103 | 21005 | B+
102 | 21229 | A
```

One way to create this table in SQL:

```
create table grades as
select "101" as SID, 21228 as CCN, "B" as Grade union
select "102", 21231, "A" union
select "101", 21105, "B+" union
select "106", 21001, "B" union
select "103", 21005, "B+" union
select "102", 21229, "A";
```

Warning: This lecture shows atypical use of SQL.

Last modified: Fri Apr 7 15:04:11 2017

CS66A, Lecture #28 3

Some Details on Definition

```
create table grades as
select "101" as SID, 21228 as CCN, "B" as Grade union
select "102", 21231, "A" union
select "101", 21105, "B+" union
select "106", 21001, "B" union
select "103", 21005, "B+" union
select "102", 21229, "A";
```

- This **create** statement is essentially an assignment to a new table variable, **grades**.
- Each **select** is a **table-valued expression** that defines a set of rows (all singleton sets in this case).
- **union** is then the set union operator on tables.
- The unioned tables must be compatible (same columns).
- First **select** establishes column names.

Last modified: Fri Apr 7 15:04:11 2017

CS66A, Lecture #28 4

Selection

- Power of SQL comes from **select** statements with conditions.
- Given table **grades** on left, the queries


```
select Grade, CCN from grades where SID = '101';
create table roster21228 as
select SID from grades where CCN = 21228;
```

create two new tables shown on the right (the first anonymous):

SID	CCN	Grade	Grade	CCN	SID
101	21228	B	B	21228	101
102	21231	A	B+	21105	105
101	21105	B+	A-	21232	104
102	21229	A	B	21001	
102	21105	A-			
101	21232	A-			
104	21228	A-			
102	21001	B+			
104	21105	A-			
105	21228	A			
104	21005	A-			
101	21001	B			

Last modified: Fri Apr 7 15:04:11 2017

CS66A, Lecture #28 5

Selection II

- ```
select Grade, CCN from grades where SID = '101';
create table roster21228 as
select distinct SID from grades where CCN = 21228;
```
- In these statements, the values added to the resulting tables are not constants (as before), but rather **column specifiers**: expressions that extract values from rows of the table **grades**.
  - By default (no **as** clauses), columns in result take their names from the selected columns.
  - SQL is declarative in the sense that we **declare the characteristics** of the table we want, without saying how to conduct the necessary search.
  - In the cases above, the search looks pretty simple, but the system hides the complexity that results when
    - **multiple** tables are involved, or
    - certain columns are **indexed** to speed up searches involving those columns.

Last modified: Fri Apr 7 15:04:11 2017

CS66A, Lecture #28 6

## Multiple Tables

- Searches can involve multiple tables:

| SID | Last      | students |       |       | SemEnt | YearEnt | Major |
|-----|-----------|----------|-------|-------|--------|---------|-------|
|     |           | First    | CCN   | Grade |        |         |       |
| 101 | Knowles   | Jason    | 21228 | B     | 2003   | EECS    |       |
| 102 | Chan      | Valerie  | 21231 | A     | 2003   | Math    |       |
| 103 | Xavier    | Jonathan | 21105 | B+    | 2004   | LSUnd   |       |
| 104 | Armstrong | Thomas   | 21232 | A-    | 2003   | EECS    |       |
| 105 | Brown     | Shana    | 21229 | A     | 2004   | EECS    |       |
| 106 | Chan      | Yangfan  | 21229 | A     | 2003   | LSUnd   |       |

create table report as select Last, First, CCN, Grade  
from grades, students where students.SID = grades.SID;

| Last    | First   | report |       |     |
|---------|---------|--------|-------|-----|
|         |         | CCN    | Grade |     |
| Knowles | Jason   | 21228  | B     |     |
| Chan    | Valerie | 21231  | A     |     |
| Knowles | Jason   | 21105  | B+    |     |
| Chan    | Valerie | 21229  | A     |     |
| ...     | ...     | ...    | ...   | ... |

Last modified: Fri Apr 7 15:04:11 2017

CS61A Lecture #28 7

## Another Example

- Suppose we supply a translation table from grades to points (on the left).

- Now can ask
 

```
select Last, First, CCN, Grade from students, grades, grade-values where
students.SID = grades.SID and Letter = Grade and GP >= 3.7;
```

| grade_values | Letter | GP  | Last      | First   | CCN   | Grade |
|--------------|--------|-----|-----------|---------|-------|-------|
| A+           | A      | 4   | Chan      | Valerie | 21231 | A     |
| A            | A      | 4   | Chan      | Valerie | 21229 | A     |
| A-           | A      | 3.7 | Chan      | Valerie | 21105 | A-    |
| B+           | B      | 3.3 | Knowles   | Jason   | 21232 | A-    |
| B            | B      | 3   | Armstrong | Thomas  | 21228 | A-    |
| B-           | B      | 2.7 | ...       | ...     | ...   | ...   |

Last modified: Fri Apr 7 15:04:11 2017

CS61A Lecture #28 8

## Arithmetic, Etc.

- It is also possible to construct values by computation.
- This table produces the grade points awarded for each letter grade in each course (CCN):

```
create table units as
select "21228" as CCN, 4 as Units union
select "21231", 3 union
select "21105", 1 union
select "21232", 4 union
select "21001", 3;

create table credits as select units.CCN, Letter, GP * Units
from units, grade-values;
```

Last modified: Fri Apr 7 15:04:11 2017

CS61A Lecture #28 9

## Mutation

- We've looked at a functional subset of SQL: we never change a table, just create new ones, as in nondestructive operations.
- In real life, we also change existing tables.

```
create table grades (SID, CCN, Grade);
insert into grades values ("101", 21228, "B");
insert into grades values ("101", 21228, "A");
insert into grades values ("102", 21231, "A");
insert into grades values ("101", 21105, "B+");
insert into grades values ("106", 21001, "B");
insert into grades values ("103", 21005, "B+");
insert into grades values ("102", 21229, "A");

-- Can also insert from a select:
create table selected_report (SID, CCN, Grade);
insert into selected_report select * from report where SID = "106";
insert into selected_report select * from report where SID = "108";
```

- We will not emphasize mutation in this course, however.

Last modified: Fri Apr 7 15:04:11 2017

CS61A Lecture #28 10