COGS 121
HCI Programming Studio
Week 03 - Tech Lecture
• Assignment #1 extended to Monday night 11:59pm

• Assignment #2 to be released on Tuesday during lecture
Database Management Systems and SQL

Week 03 - Tech Lecture
References and Acknowledgments

https://pgexercises.com

Prof. A. Vaisman (U Toronto)
What Is a DBMS?

• A very large, integrated collection of data describing activities of organizations.

• Models real-world.

  • Entities (e.g., students, courses)

  • Relationships (e.g., Madonna is taking CS564)

• A Database Management System (DBMS) is a software package designed to store and manage databases.
Why Use a DBMS?

• Data independence and efficient access.
• Reduced application development time.
• Data integrity and security. Different users may access different data subsets.
• Uniform data administration.
• Concurrent access, recovery from crashes.
Describing Data: Data Models

- A data model is a collection of concepts and constructs for describing data.

- A schema is a description of a particular collection of data, using the a given data model.

- The relational model of data is the most widely used model today.
  - Main concept: relation, basically a table with rows and columns.
  - Every relation has a schema, which describes the columns, or fields.
The Relational Model (Introduction)

• Central construct: the RELATION : a set of records.

• Data is described through a SCHEMA specifying the name of the relation, and name and type of each field:
  
  • Students(pid: string, name: string, login: string, age: integer, gpa: real)

• Actual data: instance of the relations : a set of tuples, v.g.: {<53666, Jones, jones@cs, 18, 3.4>,
  <53688, Smith, smith@ee, 18, 3.2>,
  <53650, Smith, jones@math, 19, 3.8>, ...}
Example: University Database

- **Conceptual schema:**
  - Students(pid: string, name: string, login: string, age: integer, gpa: real)
  - Courses(cid: string, cname: string, credits: integer)
  - Enrolled(pid: string, cid: string, grade: string)

  —> describes data in terms of the data model of the DBMS

- **Physical schema:**
  - Relations stored as unordered files.
  - Index on first column of Students.

- **External Schema (View):**
  - Course_info(pid: string, enrollment: integer)
Querying a DBMS

• A DBMS provides a Query Language.

• Query languages allow querying and updating a DBMS in a simple way.

• Most popular DML (Data Manipulation Language) : **SQL** (Structured Query Language).

• Queries:
  - List the name of student with pid=A0967546
  - Name and age of students enrolled in COGS121
Basic SQL

- SQL language
  - Considered one of the major reasons for the commercial success of relational databases

- SQL
  - Structured Query Language
  - Statements for data definitions, queries, and updates (both DDL and DML)
  - Core specification
  - Plus specialized extensions
SQL Data Definition and Data Types

• Terminology:
  
  • **Table**, **row**, and **column** used for relational model terms relation, tuple, and attribute

• **CREATE** statement

• Main SQL command for data definition
Schema and Catalog Concepts in SQL

- SQL schema
  - Identified by a schema name
  - Includes an authorization identifier and descriptors for each element
- Schema elements include
  - Tables, constraints, views, domains, and other constructs
- Each statement in SQL ends with a semicolon
CREATE TABLE EMPLOYEE
  ( Fname VARCHAR(15) NOT NULL, 
    Minit CHAR, 
    Lname VARCHAR(15) NOT NULL, 
    Ssn CHAR(9) NOT NULL, 
    Bdate DATE, 
    Address VARCHAR(30), 
    Sex CHAR, 
    Salary DECIMAL(10,2), 
    Super_ssn CHAR(9), 
    Dno INT NOT NULL, 
  PRIMARY KEY (Ssn),
  FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn),
  FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber) );

CREATE TABLE DEPARTMENT
  ( Dname VARCHAR(15) NOT NULL, 
    Dnumber INT NOT NULL, 
    Mgr_ssn CHAR(9) NOT NULL, 
    Mgr_start_date DATE, 
  PRIMARY KEY (Dnumber),
  UNIQUE (Dname),
  FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn) );

CREATE TABLE DEPT_LOCATIONS
  ( Dnumber INT NOT NULL, 
    Dlocation VARCHAR(15) NOT NULL, 
  PRIMARY KEY (Dnumber, Dlocation),
  FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber) );
Attribute Data Types and Domains in SQL

• Basic data types
  • Numeric data types
    • Integer numbers: INTEGER, INT, and SMALLINT
    • Floating-point (real) numbers: FLOAT or REAL, and DOUBLE PRECISION
  • Character-string data types
    • Fixed length: CHAR(n), CHARACTER(n)
    • Varying length: VARCHAR(n), CHAR VARYING(n), CHARACTER VARYING(n)
Attribute Data Types and Domains in SQL

- Bit-string data types
  - Fixed length: BIT(n)
  - Varying length: BIT VARYING(n)
- Boolean data type
  - Values of TRUE or FALSE or NULL
- DATE data type
  - Ten positions
  - Components are YEAR, MONTH, and DAY in the form YYYY-MM-DD
Attribute Data Types and Domains in SQL

• Additional data types
  • Timestamp data type (TIMESTAMP)
  • Includes the DATE and TIME fields
  • Plus a minimum of six positions for decimal fractions of seconds
  • Optional WITH TIME ZONE qualifier
• INTERVAL data type
  • Specifies a relative value that can be used to increment or decrement an absolute value of a date, time, or timestamp
CREATE TABLE PROJECT
  ( Pname       VARCHAR(15)  NOT NULL,
    Pnumber     INT           NOT NULL,
    Plocation   VARCHAR(15),
    Dnum        INT           NOT NULL,
  PRIMARY KEY (Pnumber),
  UNIQUE (Pname),
  FOREIGN KEY (Dnum) REFERENCES DEPARTMENT(Dnumber) );

CREATE TABLE WORKS_ON
  ( Essn       CHAR(9)       NOT NULL,
    Pno        INT           NOT NULL,
    Hours      DECIMAL(3,1)  NOT NULL,
  PRIMARY KEY (Essn, Pno),
  FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn),
  FOREIGN KEY (Pno) REFERENCES PROJECT(Pnumber) );

CREATE TABLE DEPENDENT
  ( Essn       CHAR(9)       NOT NULL,
    Dependent_name VARCHAR(15) NOT NULL,
    Sex          CHAR,         
    Bdate        DATE,        
    Relationship VARCHAR(8),
  PRIMARY KEY (Essn, Dependent_name),
  FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn) );
Specifying Key and Referential Integrity Constraints

- **PRIMARY KEY** clause
  - Specifies one or more attributes that make up the primary key of a relation
  - `Dnumber INT PRIMARY KEY;`

- **UNIQUE** clause
  - Specifies alternate (secondary) keys
  - `Dname VARCHAR(15) UNIQUE;`
Specifying Key and Referential Integrity Constraints (cont’d.)

• **FOREIGN KEY** clause

  • Default operation: reject update on violation

  • Attach referential triggered action clause

  • Options include **SET NULL**, **CASCADE**, and **SET DEFAULT**

  • Action taken by the DBMS for SET NULL or SET DEFAULT is the same for both ON DELETE and ON UPDATE

  • **CASCADE** option suitable for “relationship” relations
Query Languages

<table>
<thead>
<tr>
<th>Employee</th>
<th>Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Dept</td>
</tr>
<tr>
<td>Dept</td>
<td>Manager</td>
</tr>
</tbody>
</table>

SQL

```sql
SELECT Manager
FROM Employee, Department
WHERE Employee.name = "Clark Kent"
  AND Employee.Dept = Department.Dept
```
The SELECT-FROM-WHERE Structure of Basic SQL Queries

• Basic form of the SELECT statement:
  
  • SELECT <attribute list>
  
  • FROM <table list>
  
  • WHERE <condition>;

• where

  • <attribute list> is a list of attribute names whose values are to be retrieved by the query.
  
  • <table list> is a list of the relation names required to process the query.
  
  • <condition> is a conditional (Boolean) expression that identifies the tuples to be retrieved by the query.
The SELECT-FROM-WHERE Structure of Basic SQL Queries (cont’d.)

• Logical comparison operators
  • =, <, <=, >, >=, and <>

• Projection attributes
  • Attributes whose values are to be retrieved

• Selection condition
  • Boolean condition that must be true for any retrieved tuple
Query 0. Retrieve the birth date and address of the employee(s) whose name is ‘John B. Smith’.

Q0: SELECT Bdate, Address
    FROM EMPLOYEE
    WHERE Fname='John' AND Minit='B' AND Lname='Smith';

Query 1. Retrieve the name and address of all employees who work for the ‘Research’ department.

Q1: SELECT Fname, Lname, Address
    FROM EMPLOYEE, DEPARTMENT
    WHERE Dname='Research' AND Dnumber=Dno;

Results of SQL queries when applied to the COMPANY database state shown in Figure 3.6. (a) Q0. (b) Q1. (c) Q2. (d) Q8. (e) Q9. (f) Q10. (g) Q1C.

(a) | Bdate | Address       |
    |       |              |
    | 1965-01-09 | 731 Fondren, Houston, TX |

(b) | Fname  | Lname | Address                     |
    |       |       |                             |
    | John   | Smith | 731 Fondren, Houston, TX   |
    | Franklin | Wong | 638 Voss, Houston, TX      |
    | Ramesh | Narayan | 975 Fire Oak, Humble, TX  |
    | Joyce  | English | 5631 Rice, Houston, TX     |
Unspecified WHERE Clause and Use of the Asterisk

• Missing WHERE clause
  • Indicates no condition on tuple selection

• CROSS PRODUCT
  • All possible tuple combinations

Queries 9 and 10. Select all EMPLOYEE SsnS (Q9) and all combinations of EMPLOYEE Ssn and DEPARTMENT Dname (Q10) in the database.

Q9: SELECT Ssn FROM EMPLOYEE;

Q10: SELECT Ssn, Dname FROM EMPLOYEE, DEPARTMENT;
Unspecified WHERE Clause and Use of the Asterisk

• Specify an asterisk (*)

• Retrieve all the attribute values of the selected tuples

Q1 C: SELECT * FROM EMPLOYEE WHERE Dno=5;

Q1 D: SELECT * FROM EMPLOYEE, DEPARTMENT WHERE Dname='Research' AND Dno=Dnumber;

Q10 A: SELECT * FROM EMPLOYEE, DEPARTMENT;
Ordering of Query Results

- Use `ORDER BY` clause
  
  - Keyword `DESC` to see result in a descending order of values
  
  - Keyword `ASC` to specify ascending order explicitly
  
  - `ORDER BY D.Dname DESC, E.Lname ASC, E.Fname ASC`
Substring Pattern Matching and Arithmetic Operators

- **LIKE** comparison operator
  - Used for string **pattern matching**
  - \% replaces an arbitrary number of zero or more characters
  - underscore (_), replaces a single character
- Standard arithmetic operators:
  - Addition (+), subtraction (−), multiplication (∗), and division (/)
- **BETWEEN** comparison operator
Aggregate Functions in SQL

- Used to summarize information from multiple tuples into a single-tuple summary

- **Grouping**
  - Create subgroups of tuples before summarizing

- Built-in aggregate functions
  - `COUNT`, `SUM`, `MAX`, `MIN`, and `AVG`

- Functions can be used in the `SELECT` clause or in a `HAVING` clause
Grouping: The GROUP BY and HAVING Clauses

- Partition relation into subsets of tuples
  - Based on grouping attribute(s)
  - Apply function to each such group independently

- GROUP BY clause
  - Specifies grouping attributes

- If NULLs exist in grouping attribute
  - Separate group created for all tuples with a NULL value in grouping attribute
INSERT, DELETE, and UPDATE Statements in SQL

- Three commands used to modify the database:
  - INSERT, DELETE, and UPDATE
The INSERT Command

- Specify the relation name and a list of values for the tuple

```sql
U1: INSERT INTO EMPLOYEE
VALUES ('Richard', 'K', 'Marini', '653298653', '1962-12-30', '98 Oak Forest, Katy, TX', 'M', 37000, '653298653', 4);

U3B: INSERT INTO WORKS_ON_INFO (Emp_name, Proj_name, Hours_per_week)
SELECT E.Lname, P.Pname, W.Hours
FROM PROJECT P, WORKS_ON W, EMPLOYEE E
WHERE P.Pnumber=W.Pno AND W.Essn=E.Ssn;
```
The DELETE Command

- Removes tuples from a relation
- Includes a \texttt{WHERE} clause to select the tuples to be deleted

```
U4A: DELETE FROM EMPLOYEE WHERE Lname='Brown';
U4B: DELETE FROM EMPLOYEE WHERE Ssn='123456789';
U4C: DELETE FROM EMPLOYEE WHERE Dno=5;
U4D: DELETE FROM EMPLOYEE;
```
The UPDATE Command

- Modify attribute values of one or more selected tuples
- Additional **SET** clause in the **UPDATE** command
- Specifies attributes to be modified and new values

```
U5: UPDATE PROJECT
   SET Plocation = 'Bellaire', Dnum = 5
   WHERE Pnumber=10;
```
Let’s Play

- Download and Install PGAdmin

- Setup two databases
  - Tournament
    - host: ticino.ucsd.edu, port: 5432, database: cogs121
    - username: ‘cogs121’, password ‘sql4cogs121’, schema: cd
  - DELPHI:
    - host: delphidata.ucsd.edu, port: 5432, database: delphibetadb
    - username: ‘cogs121_16_user’, password ‘mcH8Yjs_n#2(xp’, schema: cogs121_16_raw
• username: ‘cogs121’
  password: ‘sql4cogs121’
  schema: cd

• username: ‘cogs121_16_user’
  password: ‘mcH8Yjs_n#2(xp’
  schema: cogs121_16_raw
pgAdmin Demo
TopHat Tournament