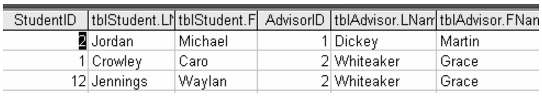
Implementing Table Operations Using Structured Query Language (SQL)

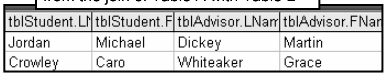
The implementation of table operations in relational database management systems is done through use of SQL, or Structured Query Language, the de facto language allowing users to access and manipulate data in RDBM systems.

Using Multiple Operations



StudentID	tblStudent.LN	tblStudent.F	AdvisorID	tblAdvisor.LNam	tblAdvisor.FNan
2	Jordan	Michael	1	Dickey	Martin
1	Crowley	Caro	2	Whiteaker	Grace
12	Jennings	Waylan	2	Whiteaker	Grace

Show Only certain columns and rows from the join of Table A with Table B



tblStudent.LN	tblStudent.F	tblAdvisor.LNam	tblAdvisor.FNar
Jordan	Michael	Dickey	Martin
Crowley	Caro	Whiteaker	Grace

- This table doesn't exist by itself. It is a view of certain rows and columns from other tables.

Queries: Create Tables From Tables

- CONCEPT: The operations on databases- Restrict/Select, Project, Union, Difference, and Product create tables from tables. These actions are done with a *Query*
- How are queries implemented?
  - □ Database systems come with a "query language" … SQL is the most common one and is the standard for Relational databases
  - □ The most common clauses used in SQL for queries are shown below:

```
SELECT <fields of desired table>     'columns to be retrieved
FROM <list of tables>                 'tables that contain data needed
INNER JOIN <table> ON<keys>          'key constraints (joins) on tables
WHERE <T/F predicate>;       'non key criteria for returning rows
```

Implementing Table Operations With SQL

- SQL stands for Structured Query Language.
- SQL is the de facto query standard for accessing and manipulating data in relational databases
- In Access you can also use a graphical query interface, called the QBE (Query By Example), that generates SQL for you

SQL: Structured Query Language

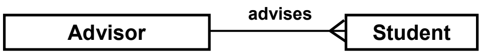
- There are many uses for SQL in database structures.
  - □ SQL can be used to define, or construct, a database
  - □ SQL can be used do basic management of the database
    - check into table content
    - add to table content
    - delete table content
    - etc.
  - □ SQL can be used to query the database
    - create virtual tables or “views” from existing table(s)
    - A view may be selected attributes from various tables

SQL Syntax

- SQL is not case sensitive.
- SQL statements combine several table operations together to display or modify the data
- But note the difference between Select and the table operation Select/Restrict
  - □ The table operation SELECT brings back rows based on some criteria
  - □ SELECT clause in SQL is actually the Projec table operation
    - □ SQL SELECT returns certain columns

A Simple ERD and Database Schema

- Advisor and Student tables
  - □ Each student is allowed a single advisor at any one time
  - □ An advisor may have zero, one or many students to advise



Advisor	Student
AdvisorID	SID
FName	FName
LName	LName
Department	MajorID
HireDate	AdvisorID
PK AdvisorID	PK SID

Basic Data Management

- Checking the Tables Contents

  SELECT <attributes> FROM <table name(s)>;

  - □ Examples:

    SELECT * FROM Student;

  is the same as

    SELECT SID, FName, LName, MajorID, AdvisorID FROM Student;

- This will essentially mimic the table Student and show all current contents in a view of the table

Queries

- Partial Listing of Table Contents

```
SELECT <attributes>
FROM <table name(s)>
WHERE <T/F predicates>;
```

  - □ Examples:

```
SELECT FName, LName, MajorID
FROM Student
WHERE SID = 0023892;

SELECT FName, LName
FROM Student
WHERE MajorID =14;
```

The WHERE clause reduces output of rows based on some specified criteria. It is one implementation of the Select/Restrict Operator

NULL Means Nothing

- A NULL character means that nothing has been entered. This is different from a space or a zero.

```
SELECT LName
FROM Student
WHERE FName IS NULL;
```

ORDER BY… Sorting Outputs

- Sorting in descending order…

```
SELECT StudentID, FName
FROM Student
ORDER BY LName DESC;
```

- Sorting in ascending order…

```
SELECT StudentID, FName
FROM Student
ORDER BY LName ASC;
```

Preparing for a Join….

- Example of a Product and Project Operation:

```
SELECT Student.FName, Student.LName,
Advisor.LName
FROM Student, Advisor;
```

- What is the result? (Using terms from the table operations lecture)

Notice that I indicate the table name with the attribute when I have more than one table in the FROM statement. Specifically when I have attributes with the same name in different tables. This is called Table Qualification

Queries Using Joins

- Example of a Join that includes Product, Project and Restrict:

```
SELECT Student.FName, Student.LName, Advisor.LName
FROM    Student INNER JOIN Advisor ON
            Student.AdvisorID = Advisor.AdvisorID;
```

Comparison Operators

Equals	=	(different from JavaScript)
Not equals	<>	(different from JavaScript)
Greater than	>	
Less than	<	
Greater than or equal to	>=	
Less than or equal to	<=	

Queries

- Use Comparison Operators for further constraints on rows to be returned
  - □ Examples:

```
SELECT FName, Lname
FROM Advisor
WHERE HireDate >= 1987;

SELECT FName, LName, Major
FROM Student
WHERE SID < > 0023892;
```

Queries

- Use logical operators to combine multiple constraints
- Logical Operators: AND, OR, (NOT is also available)
  - □ Examples:

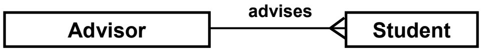
```
SELECT FName, LName
FROM Advisor
WHERE HireDate > 1987 OR
          HireDate < 1962;

SELECT FName, LName
FROM Student
WHERE AdvisorID = 44232 AND
             MajorID =14;
```

Simple Join Queries

What is the SQL statement that will:

- Return the advisor and student name for student id 0001234
- Return the advisor name for student "Joel Martin"



Advisor	Student
AdvisorID	SID
FName	FName
LName	LName
Department	MajorID
HireDate	AdvisorID
PK AdvisorID	PK SID

Just Scratching the Surface

- There are many more commands available in SQL as well as different standards for the language
- You have been shown some common clauses
- In Access you will be provided with a graphical user interface known as QBE, Query by Example, to create queries. But you can look at SQL View to see the SQL clauses that are generated
- Practice interpreting the SQL statements so you can explain what the SQL is doing in one of the queries for Project 3, Part B
- Practice with SQL at: www.w3schools.com/sql