Structured Query Language

- Started as "Sequel"
- Now an ANSI (X3H2) / ISO standard
  - SQL2 or SQL-92
  - SQL3 is in the works
- Lots of uses and versions
  - As a direct language our focus
  - As an embedded language in a program in a conventional language
  - As a protocol from client to DB server increasingly important

SQL DDL

- TABLE/SCHEMA/CATALOG
- CREATE/DROP/ALTER
- SQL DATA TYPES
  - DB files are often character-based, descended from COBOL
  - INT, FLOAT, etc.
  - DEC(i,j)
  - CHAR(n), VARCHAR(n)
  - DATE/TIME

90% of what you need to know

SELECT <attributes>
FROM <tables>
WHERE <conditions>
• SELECT is a projection into the output
• FROM is effectively a Cartesian product of all the tables used
• WHERE are conditions, which could be very elaborate and even involve nested SELECTs.

Some Notation

- Attributes can be qualified
- Tables can be renamed
  SELECT S.FNAME, S.LNAME
  FROM EMPLOYEE E, EMPLOYEE S
  WHERE E.SUPERSSN = S.ESSN
- SELECT * means display all columns
- No WHERE means display all rows

More SQL Topics

- Conditionals
  - NOT, AND, OR (in precedence order)
- Tables vs sets
  - Duplicates are NOT eliminated on most operations!
  - DISTINCT keyword eliminates dups in SELECT
Set Operations

• UNION, intersections ("INTERSECT"), difference ("EXCEPT")
  – Do eliminate duplicates
  – Use to connect whole sets (result of queries), not within WHERE
  – EXCEPT not supported by MS Access 97
• Division is not an SQL operation
• Older SQL had a CONTAINS (set inclusion)

IN and EXISTS

• IN
  – Tests for membership in a set: \( x \in A \)
  – Binary operator returning Boolean value
    • used within conditions (WHERE)
    • left side a row (or value construed as a row)
    • right side a table, frequently the result of a (nested) SELECT
• EXISTS
  – Unary operator returning a Boolean
  – Tests a table for non-empty: \( A \neq \emptyset \)

Aggregate Functions

• Five standard functions: COUNT, SUM, MIN, MAX, AVG
• Normally appears in SELECT clause
  SELECT SUM(SALARY)
• Is applied after WHERE
• Result is a column in the output
  – Can sometimes think of as a scalar

Grouping

• GROUP BY <attributes>
  – Causes rows with common values to be grouped together
  – No particular ordering among groups
• Might be used just to improve output presentation, but…
• Most commonly used in connection with aggregate functions

Grouping and Functions

In presence of grouping, aggregate functions apply to each group individually.
  – Output has one row per group
• Order of processing: WHERE conditions, then GROUPING, then SELECT (including agg. functions).
  – What if you want a condition applied after the grouping??

HAVING

• Applies a condition after grouping
  – Condition is applied to each group
• May use aggregate functions
  – HAVING SUM(QUANTITY) > 350