1.264 Lecture 8

SQL: Basics, SELECT
SQL

• Structured query language (SQL) used for
  – Data definition (DDL): tables and views (virtual tables)
  – Data manipulation (DML): user or program can INSERT, DELETE, UPDATE or retrieve (SELECT) data
  – Access control: security
  – Data sharing: by concurrent users
  – Data integrity: referential integrity and transactions

• Not a complete language like Java, Visual Basic or C++
  – SQL is sub-language of about 30 statements
  – Usually embedded in another language or tool for database access
  – SQL has several inconsistencies; NULLs are problematic
  – Portable across operating systems and somewhat among vendors
Things that vary among SQL implementations

- Error codes
- Data types supported (dates/times, currency, string variations)
- System tables, about the structure of the database itself
- Interactive SQL
- Programming interface: no vendor follows the standard
- Dynamic SQL, used for report writers and query tools
- Implementer-defined variations within the standard
- Database initialization, opening and connection
SQL SELECT

- SELECT constructed of clauses to get columns and rows from one or more tables or views.Clauses must be in order:
  - SELECT columns
  - INTO new table
  - FROM table or view
  - WHERE specific rows or a join is created
  - GROUP BY grouping conditions (columns)
  - HAVING group-property (specific rows)
  - ORDER BY ordering criterion  ASC | DESC
### Example tables

#### Orders

<table>
<thead>
<tr>
<th>OrderNbr</th>
<th>Cust</th>
<th>Prod</th>
<th>Qty</th>
<th>Amt</th>
<th>Disc</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>211</td>
<td>Bulldozer</td>
<td>7</td>
<td>$31,000.00</td>
<td>0.2</td>
</tr>
<tr>
<td>2</td>
<td>522</td>
<td>Riveter</td>
<td>2</td>
<td>$4,000.00</td>
<td>0.3</td>
</tr>
<tr>
<td>3</td>
<td>522</td>
<td>Crane</td>
<td>1</td>
<td>$500,000.00</td>
<td>0.4</td>
</tr>
</tbody>
</table>

#### Customers

<table>
<thead>
<tr>
<th>CustNbr</th>
<th>Company</th>
<th>CustRep</th>
<th>CreditLimit</th>
</tr>
</thead>
<tbody>
<tr>
<td>211</td>
<td>Connor Co</td>
<td>89</td>
<td>$50,000.00</td>
</tr>
<tr>
<td>522</td>
<td>AmaratungaEnterprise</td>
<td>89</td>
<td>$40,000.00</td>
</tr>
<tr>
<td>890</td>
<td>Feni Fabricators</td>
<td>53</td>
<td>$1,000,000.00</td>
</tr>
</tbody>
</table>

#### SalesReps

<table>
<thead>
<tr>
<th>RepNbr</th>
<th>Name</th>
<th>RepOffice</th>
<th>Quota</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>53</td>
<td>Bill Smith</td>
<td>1</td>
<td>$100,000.00</td>
<td>$0.00</td>
</tr>
<tr>
<td>89</td>
<td>Jen Jones</td>
<td>2</td>
<td>$50,000.00</td>
<td>$130,000.00</td>
</tr>
</tbody>
</table>

#### Offices

<table>
<thead>
<tr>
<th>OfficeNbr</th>
<th>City</th>
<th>State</th>
<th>Region</th>
<th>Target</th>
<th>Sales</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Denver</td>
<td>CO</td>
<td>West</td>
<td>$3,000,000.00</td>
<td>$130,000.00</td>
<td>970.586.3341</td>
</tr>
<tr>
<td>2</td>
<td>New York</td>
<td>NY</td>
<td>East</td>
<td>$200,000.00</td>
<td>$300,000.00</td>
<td>212.942.5574</td>
</tr>
<tr>
<td>57</td>
<td>Dallas</td>
<td>TX</td>
<td>West</td>
<td>$0.00</td>
<td>$0.00</td>
<td>214.781.5342</td>
</tr>
</tbody>
</table>
Example schema
Using SQL Server and Management Studio Express

- Your SQL Server database engine should start by default when your system starts
  - Ask TAs for help if needed
- Start Management Studio Express (MSE) from Start->Programs
- Open Lecture7OrderDB.sql with MSE in Windows Explorer
  - Download the .sql file from the MIT Server
- Select ‘Execute’ from toolbar
  - Database should be created and data inserted for exercises during this class
SQL queries: SELECT

• Click ‘New Query’ in MSE; type these statements:
• List the sales reps
  – SELECT Name, Sales, Quota FROM SalesReps;
• Find the amount each rep is over or under quota
  – SELECT Name, Sales, Quota, (Sales-Quota) FROM SalesReps;
• Find the slackers
  – SELECT Name, Sales, Quota, (Sales-Quota) FROM SalesReps WHERE Sales < Quota;

<table>
<thead>
<tr>
<th>RepNbr</th>
<th>Name</th>
<th>RepOffice</th>
<th>Quota</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>53</td>
<td>Bill Smith</td>
<td>1</td>
<td>$100,000.00</td>
<td>$0.00</td>
</tr>
<tr>
<td>89</td>
<td>Jen Jones</td>
<td>2</td>
<td>$50,000.00</td>
<td>$130,000.00</td>
</tr>
</tbody>
</table>
SQL queries: calculation, insert, delete, update

- Find the average sale
  - `SELECT AVG(Amt) FROM Orders;`
- Find the average sale for a customer
  - `SELECT AVG(Amt) FROM Orders WHERE Cust = 211;`
- Add an office
  - `INSERT INTO Offices (OfficeNbr, City, State, Region, Target, Sales, Phone) VALUES ('55', 'Dallas', 'TX', 'West', 200000, 0, '214.333.2222');`
- Delete a customer
  - `DELETE FROM Customers WHERE Company = 'Connor Co';`
  - (Syntax is valid but command will fail due to referential integrity)
- Raise a credit limit
  - `UPDATE Customers
    SET CreditLimit = 75000 WHERE Company = 'Amaratunga Enterprises';`
SELECT: * and duplicates

- Select all columns (fields)
  - SELECT * FROM Offices;
- Duplicate rows: query will get two instances of ‘West’
  - SELECT Region FROM Offices;
- Eliminate duplicates:
  - SELECT DISTINCT Region FROM Offices;
NULLs

- NULL values evaluate to NOT TRUE in all cases.
  - Insert ‘NewRep’ with NULL (blank or empty) Quota
  - Write this statement yourself!
- The following two queries will not give all sales reps:
  - SELECT Name FROM SalesReps WHERE Sales > Quota;
  - SELECT Name FROM SalesReps WHERE Sales <= Quota;
  - A new rep with a NULL quota will not appear in either list
- Check for NULLS by:
  - SELECT Name FROM SalesReps WHERE Quota IS NULL;
SELECT Operators

- **SELECT * FROM <table>**
  - WHERE Disc*Amt > 50000;  
    (Orders)
  - WHERE Quota BETWEEN 50000 AND 100000;  
    Range is inclusive (>=50000 and <=100000)
  - WHERE Quota BETWEEN 50000 AND 100000;  
  - WHERE State IN (‘CO’, ‘UT’, ‘TX’);  
    (Offices)
  - WHERE RepNbr IS NOT NULL;  
    (SalesReps)
  - WHERE Phone NOT LIKE ‘21%’;  
    (Offices)

- SQL standard only has 2 wildcards
  - % any string of zero or more characters (* in Access)
  - _ any single character (? in Access)

- Most databases have additional/different wildcards. SQL Server has:
  - [list] match any single character in list, e.g., [a-f]
  - [^list] match any single character not in list, e.g. [^h-m]
SELECT: COUNT, GROUP BY

Parts

<table>
<thead>
<tr>
<th>PartID</th>
<th>Vendor</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>A</td>
</tr>
<tr>
<td>234</td>
<td>A</td>
</tr>
<tr>
<td>345</td>
<td>B</td>
</tr>
<tr>
<td>362</td>
<td>A</td>
</tr>
<tr>
<td>2345</td>
<td>C</td>
</tr>
<tr>
<td>3464</td>
<td>A</td>
</tr>
<tr>
<td>4533</td>
<td>C</td>
</tr>
</tbody>
</table>

• Number of parts from vendor A
  - SELECT COUNT(*) FROM Parts WHERE Vendor = ‘A’;
  - Result: 4

• Number of parts from each vendor
  - SELECT Vendor, COUNT(*) AS PartsCount FROM Parts GROUP BY Vendor;
  - Result:

<table>
<thead>
<tr>
<th>Vendor</th>
<th>PartsCount</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
</tr>
</tbody>
</table>
Exercises

• What is the average credit limit of customers whose credit limit is less than $1,000,000?
• How many sales offices are in the West region?
• Increase the price of bulldozers by 30% in all orders
• Delete any sales rep with a NULL quota
Exercises

• What is the average credit limit of customers whose credit limit is less than $1,000,000?
  – SELECT AVG(CreditLimit) FROM Customers WHERE CreditLimit < 1000000;

• How many sales offices are in the West region?
  – SELECT Count(*) FROM Offices WHERE Region = 'West';

• Increase the price of bulldozers by 30% in all orders
  – UPDATE Orders SET Amt = Amt * 1.3 WHERE Prod = 'Bulldozer';

• Delete any sales rep with a NULL quota
  – DELETE FROM SalesReps WHERE Quota IS NULL;