

1.264 Recitation 10/3-4

Darris Hupp

Sol Bisker

Overview

- Announcements
- More SQL from Lecture 8
- Loading Chemical Manufacture Dataset
- Homework 4 (Data Modeling) Help

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; (SalesReps)
 - Range is inclusive (>=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

- 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:**

PartID	Vendor
123	A
234	A
345	B
362	A
2345	C
3464	A
4533	C

Vendor	PartsCount
A	4
B	1
C	2

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;**

Chemical Manufacturer Dataset

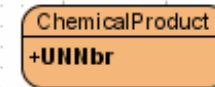
- Download from website
 - Under Materials -> General
 - Save it to C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data
- Open SQL Server Management Studio Express and connect
- In object explorer (right side of screen), right click on Databases and choose “Attach” from the menu
- Click Add, and locate the Homework5F2006.mdf file you just saved on your hard drive
- Under database details, select the log file, and click remove
- Click OK
- If you expand the Databases folder in the object explorer, you should see the Homework5F2006 database you just attached

Using the Chemical Dataset

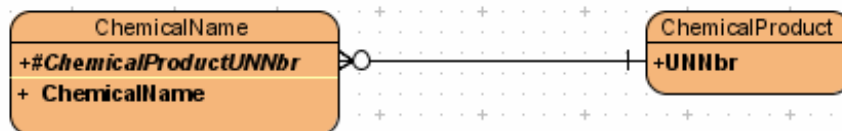
- Click “New Query” in the toolbar to open a new query window
- Type these statements (click Execute in toolbar to run, or press F5... highlight a statement first to execute only that statement):
- Tell SQL Server which database you are using:
 - `USE Homework5F2006`
- Show the entire CHEMICAL_RAW table
 - `SELECT * FROM CHEMICAL_RAW`
- List all the chemicals whose price is more than \$3.00 in descending order by price
 - `SELECT UN_NUMBER, CHEM_NAME, Price FROM CHEMICAL_RAW WHERE Price > 3 ORDER BY Price DESC`
- List the number of chemicals with each price, in descending order by price
 - `SELECT COUNT(*) AS PriceCount, Price FROM CHEMICAL_RAW GROUP BY Price HAVING Price > 3 ORDER BY Price DESC`

Homework 4 Help

1. Define the Chemical Product entity. Use international (UN) number as the primary key. This defines a unique chemical product at the manufacturer level.

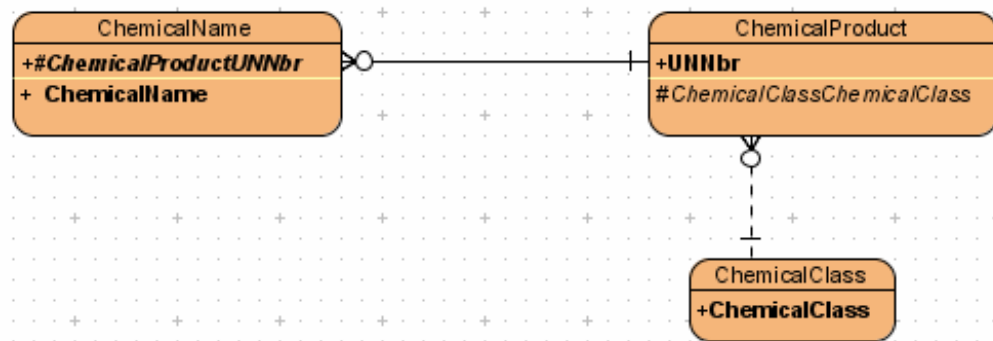


2. Create a separate ChemicalName entity to hold the multiple names associated with each UN number. Don't keep any chemical names in the Chemical Product entity.
 - Build the relationship from ChemicalName to ChemicalProduct. In this entire homework, build the relationships right after building each new entity. Visual Paradigm will help create the foreign/primary key.



Homework 4 Help

3. Create a domain (validation) entity to store the valid values of the chemical class data (1 through 9, with decimal subdivisions). Chemical class should be a foreign key in the Chemical Product entity you define, which Visual Paradigm will create automatically for you when you create the relationship.



Homework 4 Help

- Place the quantity limits in a separate QuantityLimit entity, by passenger and cargo vehicle. These should not be in the Chemical Product entity. The separate entity should hold the UN number, vehicle type (cargo or passenger) and quantity limit.
 - You will need to create a vehicle type domain entity before building the QuantityLimit entity.

