Intro to Database Design and SQL Coding

RP Institute 6-11-2018
Brian Murphy
Butte College
Murphybr@butte.edu
(530) 895-2987
DID YOU SEE ANY ERRORS ON THE SPREADSHEET I PUT TOGETHER?

ONLY THREE.

WHAT ARE THEY?

YOUR DATA, YOUR FORMAT, AND YOUR FORMULAS.
ALGORITHMS
BY COMPLEXITY

MORE COMPLEX

LEFTPAD  QUICKSORT  GIT
MERGE  SELF-DIVING  GOOGLE
CAR  SEARCH  BACKEND

SPRAWLING EXCEL SPREADSHEET
BUILT UP OVER 20 YEARS BY A
CHURCH GROUP IN NEBRASKA TO
COORDINATE THEIR SCHEDULING

xkcd
A WEBCOMIC OF ROMANCE,
SARCASM, MATH, AND LANGUAGE.
xkcd UPDATES EVERY MONDAY, WEDNESDAY, AND FRIDAY.
Becoming Normal

There are 6 normal forms or levels + some added flavors.

You need to understand getting to the 3rd level

1st – No repeating columns, no positional values

2nd - Every column depends on the primary key

3rd – Every column only depends on the primary key

4th – No Many to Many relationships within a table
<table>
<thead>
<tr>
<th>ZipCode</th>
<th>City</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>95667</td>
<td>Placerville</td>
<td>CA</td>
</tr>
<tr>
<td>95670</td>
<td>Rancho Cordova</td>
<td>CA</td>
</tr>
<tr>
<td>95130</td>
<td>Susanville</td>
<td>CA</td>
</tr>
<tr>
<td>95928</td>
<td>Chico</td>
<td>CA</td>
</tr>
</tbody>
</table>
Customer
- CustomerID
- FirstName
- LastName
- Address1
- Address2
- Zip
- Phone
- Email

Zip
- ZipCode
- City
- State

Navigation Properties
- Customers
<table>
<thead>
<tr>
<th>ISBN</th>
<th>AuthorID</th>
</tr>
</thead>
<tbody>
<tr>
<td>978-3-16-148410-0</td>
<td>1</td>
</tr>
<tr>
<td>978-3-16-148410-1</td>
<td>2</td>
</tr>
<tr>
<td>978-3-16-148410-3</td>
<td>3</td>
</tr>
<tr>
<td>978-3-16-148410-3</td>
<td>4</td>
</tr>
</tbody>
</table>
Data Types

String
  • Varchar, char, nvarchar

Numeric
  • Int, decimal, float, real, money

Temporal
  • Date, time, datetime, smalldatetime

Other
  • Varchar(max), nvarchar(max), varbinary(max)
Ladies and gentlemen, this is your stewardess speaking. We regret any inconvenience the sudden cabin movement might have caused. This is due to periodic air pockets we encountered. There's no reason to become alarmed and we hope you enjoy the rest of your flight. By the way, is there anyone on board who knows how to fly a plane?
Be sure to include the tables necessary to show the interactions between

- Student
- Course
- Instructor
A word about Data Warehouses

Normalized databases like the ones used in ERP systems are good at:

- Quickly making transactions
- Locking individual records

Are bad at:

- Quickly getting data out
  - Specialized knowledge
  - No calculated values
  - Joins take time to process

Data Warehouses are de-normalized to reduce # of joins and have calculated values stored in the database
Getting Data Out

Finally, what you came for – SQL Code 😊
JOINS

INNER
LEFT OUTER
RIGHT OUTER
FULL OUTER
UNION
UNION ALL
SELECT DISTINCT dbo_STUDENT.STU_FIRST_NAME, dbo_STUDENT.STU_LAST_NAME, StudentsOfColorRetreatAttendees.[Butte College ID], dbo_STUDENT.STU_CUM_GPA, dbo_ENROLLMENT.ENR_TERM, dbo_ENROLLMENT.ENR_CURRENT_REG_FLAG, dbo_SECTION.SEC_NAME, dbo_SECTION.SEC_MODE, dbo_SECTION_MEETING.SEM_DAYS_DISPLAY, dbo_SECTION_MEETING.SEM_TIMES_DISPLAY
FROM (((StudentsOfColorRetreatAttendees INNER JOIN dbo_STUDENT ON StudentsOfColorRetreatAttendees.[Butte College ID] = dbo_STUDENT.STUDENT_ID) INNER JOIN dbo_ENROLLMENT ON dbo_STUDENT.STUDENT_ID = dbo_ENROLLMENT.ENR_STUDENT_ID) INNER JOIN dbo_SECTION ON (dbo_ENROLLMENT.ENR_TERM = dbo_SECTION.SEC_TERM) AND (dbo_ENROLLMENT.ENR_SECTION_ID = dbo_SECTION.SEC_ID)) INNER JOIN dbo_SECTION_MEETING ON dbo_SECTION.SEC_ID = dbo_SECTION_MEETING.SEC_ID WHERE (((dbo_ENROLLMENT.ENR_TERM)="2017FA") AND ((dbo_ENROLLMENT.ENR_CURRENT_REG_FLAG)="Y"));
Select Statements

SELECT
FROM
(Joins)
WHERE
Group By
Having
Order By
Samples

SELECT Gender
FROM Student
WHERE Gender = 'M';

SELECT Distinct Gender
FROM Student
Order by 1;

SELECT Gender, COUNT(studentid)
From Student
Group by Gender
Having Gender = 'M';
SELECT TOP 100 *
FROM STUDENT
INNER JOIN ENROLLMENT ON STUDENT.STUDENT_ID = ENROLLMENT.STUDENT_ID
INNER JOIN SECTION ON ENROLLMENT.SECTION_ID = SECTION.SECTION_ID
INNER JOIN COURSE ON SECTION.COURSEID = COURSE.COURSEID
WHERE TERM = '2017FA'
AND (SUBJECT = 'MATH' OR SUBJECT = 'ENGL')
SELECT TOP 100 *
FROM STUDENT
INNER JOIN ENROLLMENT ON STUDENT.STUDENT_ID = ENROLLMENT.STUDENT_ID
INNER JOIN SECTION ON ENROLLMENT.SECTION_ID = SECTION.SECTION_ID
INNER JOIN COURSE ON SECTION.COURSEID = COURSE.COURSEID
WHERE TERM = '2017FA'
AND SUBJECT IN ('ENGL', 'MATH')
AND GRADE IS NOT NULL;
SELECT *
FROM STUDENT_TERM
LEFT JOIN FINANCIAL_AID ON STUDENT_TERM.STUDENTID = FINANCIAL_AID.STUDENTID
AND STUDENT_TERM.TERM = FINANCIAL_AID.TERM
INNER JOIN STUDENT ON STUDENT_TERM.STUDENTID = STUDENT.STUDENTID
Know thy Data
Fire Up SQL Server Management Studio

murphybradworks.database.windows.net

RP_Institute_2018

RP1nstituteGU3$t

RP1nstitute