

Relational Databases I

Source Work:
Malone, Thomas. *15.561 Information Technology Essentials, Spring 2005*.
(MIT OpenCourseWare: Massachusetts Institute of Technology),
<http://ocw.mit.edu/courses/sloan-school-of-management/15-561-information-technology-essentials-spring-2005>
(Accessed 19 Nov, 2014). License: Creative Commons BY-NC-SA

License:
Health Information Technology by Hye-Chung Kum is licensed under a
[Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](#)

Course URL:
<http://pinformatics.org/phpm631>

1 (3/5/2017)

Outline

- What is a database?
- What is a database management system?
- An Introduction to SQL
 - How to retrieve data from a database
 - How to create a database (optional)

2 (3/5/2017)

Why are we learning this?

- Databases are perhaps the single most important class of corporate applications
- Databases are surprisingly powerful data modeling and analysis tools in situations where spreadsheets fall short
 - Students who plan to work in management consulting will soon find this out
- MS Access is a great example of how easy it is to build powerful applications without the need of a background in technology
 - But we are going to learn SQL directly

3 (3/5/2017)

What is a database

- Boring answer
 - A structured collection of data
 - Example: A telephone directory
- Insightful answer
 - Digital representation of the real world
 - A data-centered mirror of an organization's business processes
 - Structure of data reflects organizational processes
 - Content of data reflects organization's history

4 (3/5/2017)

Representing the Real World as Data: What Data is needed?

- Entity
 - a person, place, thing, or event on which we maintain information
 - Examples: Employees, Patients, Products, Services, Warehouses
- Attribute
 - characteristic or quality of particular entity
 - Examples: Employee's SSN, Patient's Address, Product's Unit Price
- Relationships Among Entities
 - Examples:
 - Patients -Visits -Services(s)
 - Patients -Seen by -Doctors

5 (3/5/2017)

From Spreadsheets to Databases

- Spreadsheets are great for keeping track of data for one type of entities
 - Participants of a conference
 - Students of a class
 - Customers of a company
 -

6 (3/5/2017)

What is the basic spreadsheet “data model”?

- Each row stores data about one entity
- Each column stores data about an attribute
- Each cell stores data about an attribute of an entity



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H
1	patientID	fname	lname	mi	race	sex	addr	phone
2	101	EDWIN	TILLMAN	B	W	M	1804 WHITE OAK CHURCH RD	979-237-0168
3	102	DEANN	JOHNSON	Y	W	F	401 WAVERLY HILLS DR	979-741-0546
4	103	LILA	AUTRY	J	W	F	1220 WHITE OAK CHURCH RD	979-857-9563
5	104	JEANNE	WHITTLESEY	B	B	M	1132 BATCHELOR RD	979-892-6769
6	105	FREDDIE	BRASWELL	E	B	F	7028 JENKS RD	979-550-6488

Spreadsheet limitations

- Things get complicated when we want to keep track of several inter-related entities
- For example:
 - Patients
 - Medications
 - Visits
- You tried it!
 - How was it?
 - You figured out the relevant data
 - But, to keep the sheet to fully incorporate all the relationships, and always correct how much work would it be?

Spreadsheets are awkward for storing relationships

- Main difficulty is that a “visit” is essentially a relationship between one Patient and one or more Services provided, Medications prescribed, and Labs ordered

Storage of information is not even half the story

- The reason we build databases is in order to easily retrieve information to answer questions that support managerial decision-making
- For example:
 - Who are our top 10 most costly patients based on their total costs during hospital visits in 2014?
- Can you do this using a spreadsheet?
- Multiple people at once?

Spreadsheets

- PRO: Human friendly = flexible
 - Data for people (notes)
- CON: not computer friendly
 - Difficult to keep the data strictly following rules (protocols)
 - If rules are not strictly followed, computers can not do much with the data
 - Keeping multiple sheets consistent (relationships) is difficult
 - Concurrent access by many users is difficult
 - Tracking change is difficult
 - Trackback, if something is not right, is difficult

Relational Databases

- A relational DB supports storage of data as a set of inter-related tables
 - Each table stores data about a set of Entities
 - Each table row is a record about one such Entity
 - Each record column is a field specifying an attribute of this Entity
 - Each record has a field that acts as a unique identifier of an entity (usually an ID number)
 - Relationships among entities are specified by referring to this unique identifier from other tables
 - DB Schema: blueprint of the DB
 - Table (entity, attributes) & relationships

	A	B	C	D	E	F	G	H		
1	patientID	fname	lname	mi	race	sex	addr	phone		
2	101	EDWIN	TILLMAN	B	W	M	1804 WHITE OAK CHURCH RD	979-237-0168		
3	102	JOHN	HANSON	A	W	M	1000 CHURCH RD	979-741-0546		
4	103	JERRY	JOHNSON	J	W	M	1000 CHURCH RD	979-857-9563		
Patient Unique ID		Reference to a patient								
5	104	1	visitID	patientID	providerID	dtl_dos	dtl_qtr	diag1	diag2	diag3
6	105	50	86	101	1	11/29/2012	2012QTR4	4019	2720	
7	106	51	87	101	1	12/7/2012	2012QTR4	4019	2720	
8	107	51	86	101	1	12/7/2012	2012QTR4	4019	2720	
9	108	52	88	102	1	12/14/2012	2012QTR4	4019	2720	
10	109	53	140	102	1	1/4/2013	2013QTR1	78830		
11	110	54	141							
12	111	55	142							
13	112	56	143							
14	113	57	144							
15	114	58	201							
16	115	58	201							
17	116	59	206							
18	117	60	238							
19	118	61	244							
20	119	62	264							
21	63	266								
22	64	267								
POPULATION INFORMATIONAL ANALYSIS GROUP										
© 2008										

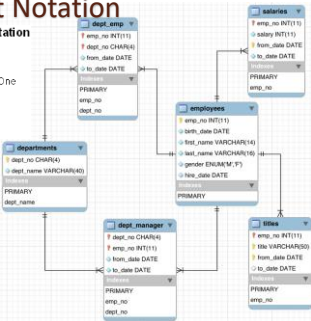
Primary Key & Foreign key

- Combinations of foreign and primary keys are used to implement & enforce relationships
- Primary Key
 - Every table must have a field whose value uniquely identifies each record in the table (e.g. the SSN for people, the ISBN for books, etc.)
 - That field is called the primary key and is marked in the fields list
 - This “represents” the entity of the table
- Foreign Key
 - Used to identify records (entities) in other table according to relation
 - An attribute in a table that uses primary key in another table
- Referential Integrity
 - Based on types of relations keep DB consistent
 - Example: Cascading deletes, not allow inserts

Entity Relationship Diagram Crow's Foot Notation

Summary of Crow's Foot Notation

- One or Zero
- One and only One
- Zero or Many
- One or Many



Outline

- What is a database?
- What is a database management system?
- An Introduction to SQL
 - How to retrieve data from a database
 - How to create a database (optional)

Relational Database Management Systems (DBMS)

- A software that allows the creation of relational databases
- Supports specialized languages (e.g., SQL) for easy retrieval of data from a set of inter-related tables
- Supports easy construction of a Graphical User Interface (GUI) on top of the database
- Allows very large table sizes
- Provides security, fault tolerance, multi-user support, etc.

DBMS

- Different DBMS
 - SQLite
 - MySQL (backend) + php (frontend: web GUI)
 - MS SQL Server, MS Access
 - Oracle
 - DB2 (IBM)
- ODBC: Open DB Connectivity
 - Middleware API to talk to DBs
- Standard Language: SQL
 - Each software has variations to the standard
 - Import/export via standard sql
- Most universal file format
 - Comma separate values: csv
 - Some can handle first line being header

Software stack

DB Applications
(web interface, GUI, e.g., blackboard)


ODBC

mysql

sqlite

Oracle

MS SQL




POPULATION
INFORMATICS
RESEARCH GROUP

19 (3/5/2017)

Take Away I


Relational Databases

- A relational DB supports storage of data as a set of inter-related tables
 - Each table stores data about a set of Entities
 - Each table row is a record about one such Entity
 - Each record column is a field specifying an attribute of this Entity
 - Each record has a field that acts as a unique identifier of an entity (usually an ID number)
 - Relationships among entities are specified by referring to this unique identifier from other tables
- DB Schema: blueprint of the DB
 - Table (entity, attributes) & relationships




POPULATION
INFORMATICS
RESEARCH GROUP

20 (3/5/2017)



Download and Read Assignment/Lab 5




POPULATION
INFORMATICS
RESEARCH GROUP

21 (3/5/2017)

Outline

- What is a database?
- What is a database management system?
- **An Introduction to SQL**
 - **How to retrieve data from a database**
 - How to create a database (optional)



POPULATION
INFORMATICS
RESEARCH GROUP


22 (3/5/2017)

SQL – Structured Query Language

- Every statement yields a table of values as output
 - Sometimes there's only one row in the table!

Keyword	parameters
select	Iname, fname
from	patients
where	gender='F'
group by	group rows together
order by	Iname, fname

- semicolon: Don't Forget.
 - Tells the computer I am done writing my statement




POPULATION
INFORMATICS
RESEARCH GROUP

23 (3/5/2017)

Choose Columns

- Choosing a subset of columns is sometimes called a "project" operation
- Display first and last name of all patients
 - `SELECT FirstName, LastName FROM patients;`
- TRY: Display diagnosis and visitDate for all patients

```
SELECT Column1, Column2  
FROM Table;
```



POPULATION
INFORMATICS
RESEARCH GROUP

24 (3/5/2017)

Display an Entire Table

- Wildcard
- * : means all columns
- Regular Expression

```
SELECT *  
FROM patients;
```



25 (3/5/2017)

Choose Rows

- Find Hispanic patients
 - SELECT FirstName, LastName
 - FROM Patients
 - WHERE Race = "Hispanic";
- TRY: Find patients born after Jan. 1, 1993
 - YYYY-MM-DD: standard SQL (e.g. '1993-01-01')

```
SELECT Column1, Column2  
FROM Table  
WHERE Condition;
```



26 (3/5/2017)

Choose Rows: conditional statements

- String: WHERE race = "Hispanic";
- String: WHERE fname LIKE "s%";
 - staring with letter s
- Number: WHERE age>18;
- Date: Where svc_dt> '1993-01-01';
 - YYYY-MM-DD: standard SQL
 - Find patients born after Jan. 1, 1993

```
SELECT Column1, Column2  
FROM Table  
WHERE Condition;  
— variable operator value;
```



27 (3/5/2017)

Conditional Operators

Operator	Description
=	Equal
<>	Not equal. Note: In some versions of SQL this operator may be written as !=
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
BETWEEN	Between an inclusive range
LIKE	Search for a pattern (approximately same)
IN	To specify multiple possible values for a column



28 (3/5/2017)

Boolean Logic

Truth Tables (1=T; 0=F)
WHERE gender='Male' AND age>18;
WHERE race='Hispanic' OR race='Black';

x	y	NOT	AND	OR
		~ y	x & y	x y
0	0	1	0	0
0	1	0	0	1
1	0	1	0	1
1	1	0	1	1



29 (3/5/2017)

Try on w3schools

- Which is the correct SQL statement below?
- What is the problem with the incorrect SQL statements ?
- SELECT * FROM Categories
 - where categoryname=beverages;
 - SELECT * FROM Categories
 - where categoryname='beverages';
 - SELECT * FROM Categories
 - where categoryname='Beverages';
- You may also try the free trials at
 - <https://academy.vertabelo.com>



30 (3/5/2017)

Sorting

- Can sort output by contents of a column
 - sort in ascending or descending order
 - sort by more than one column (second one breaks ties)
- Sort patients by last name
 - SELECT * FROM patients
 - ORDER BY lname DESC;
- TRY: What are our 10 most costly visits?

```
SELECT Column1, Column2
FROM Table
ORDER BY Column1 [DESC];
```

Take Away II

SQL – Structured Query Language

- Every statement yields a table of values as output
 - Sometimes there’s only one row in the table!

Keyword	parameters
select	columns and/or expressions
from	Tables
where	conditions on the rows
group by	group rows together
order by	order the rows
;	

Take Away III

Key Concepts in DBMS & SQL

- Keys are used to implement and enforce relationships
- Primary key uniquely identifies records in one table
- Foreign key uniquely identifies records in other table (following relation)
- DB Schema: Entity Relation Diagrams
- Boolean Logic

x	y	NOT	AND	OR
		~ y	x & y	x y
0	0	1	0	0
0	1	0	0	1
1	0	1	0	1
1	1	0	1	1

Resources

- <http://www.w3schools.com/sql/default.asp>
- <https://academy.vertabelo.com>
- <http://www.tutorialspoint.com/sql/index.htm>
 - Basic SQL
- <https://www.sqlite.org/lang.html>
- [Coding ground](#)
 - http://www.tutorialspoint.com/execute_sql_online.php
- [SQLite DB Browser: portableapps](#)
 - http://portableapps.com/apps/development/sqlite_database_browser_portable



Presentations
Reading Log
Lab

Other

- Office Hours
 - Mondays 3-5pm (3/13 & 3/20)
 - Spring break: Thur (3/16) 11-1
- Reading Log
 - Next week: no reading log.
 - FOCUS on learning SQL. Will be on Midterm/Final.
 - Good for resume and real job skills
 - Next week: come with questions about SQL

Lab

- Week 1: Learn SQL
 - W3C online tutorial (individual activity)
 - End of day, how far did you get? Submit lab 1
 - Do quiz and submit results
- Week 2: work on problems
 - Group activity
 - BUT both must type and submit (no copy/paste)
 - Remember you will have to do this in the midterm
 - DUE: BEFORE Spring Break
 - No Late Assignment



37 (3/5/2017)

Using SQLite

- Coding Ground: Cloud
 - .help
 - (dot, no semicolon)
 - SQL stmt;
- SQLite DB browser portableapp
 - Windows & Mac
 - If error message on load, try again



38 (3/5/2017)