Database Theory and Applications for Biomedical Research and Practice

BMIN 502 / EPID 635
Week 10: More database implementation

John H. Holmes, PhD

Institute for Biomedical Informatics

Agenda for today

- Joins, revisited
- Complex queries
- The Entity-Attribute-Value model

Joins, revisited

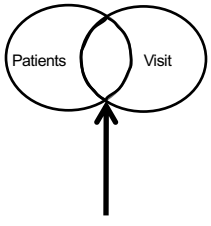
Relationships between tables

- **Join**
  - Association between a field in one table and its counterpart in another table
- **View (or Dynaset)**
  - Sets (virtual tables) created dynamically as the result of joining two or more tables

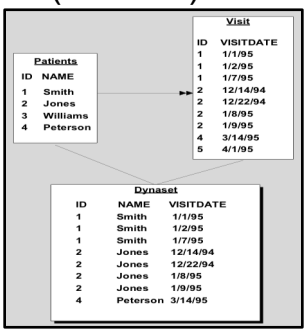
Types of joins

- **Equijoin (Inner join)**
  - Records are combined and added to dynaset only when values are equal for the join fields
- **Left outer join**
  - Records from “left-hand” table are added to dynaset even if none in “right-hand” table match
- **Right outer join**
  - Records from “right-hand” table are added to dynaset even if none in “left-hand” table match

INNER JOIN (or JOIN)

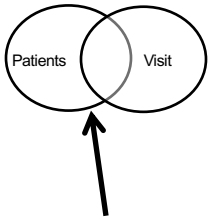


Result of JOIN

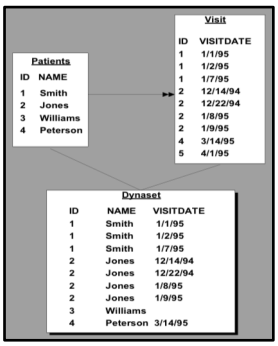



```
SELECT Patients.ID, Visit.VISITDATE
FROM Patients
JOIN Visit on Paitients.ID = Visit.ID
```

LEFT JOIN



Result of LEFT JOIN



Patients

ID	NAME
1	Smith
2	Jones
3	Williams
4	Peterson

Visit

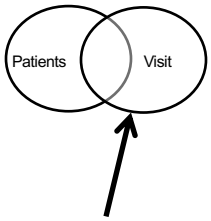
ID	VISITDATE
1	1/1/95
1	1/2/95
1	1/7/95
2	12/14/94
2	12/22/94
2	1/8/95
2	1/9/95
4	3/14/95
5	4/1/95

Dynaset

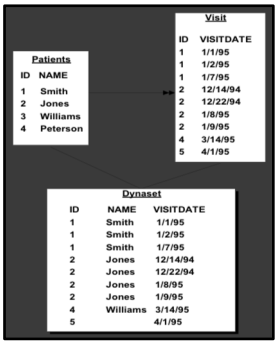
ID	NAME	VISITDATE
1	Smith	1/1/95
1	Smith	1/2/95
1	Smith	1/7/95
2	Jones	12/14/94
2	Jones	12/22/94
2	Jones	1/8/95
2	Jones	1/9/95
3	Williams	
4	Peterson	3/14/95

```
SELECT Patients.ID, Visit.VISITDATE
FROM Patients
LEFT JOIN Visit on Paitients.ID = Visit.ID
```

RIGHT JOIN



Result of RIGHT JOIN



Patients

ID	NAME
1	Smith
2	Jones
3	Williams
4	Peterson

Visit

ID	VISITDATE
1	1/1/95
1	1/2/95
1	1/7/95
2	12/14/94
2	12/22/94
2	1/8/95
2	1/9/95
4	3/14/95
5	4/1/95

Dynaset

ID	NAME	VISITDATE
1	Smith	1/1/95
1	Smith	1/2/95
1	Smith	1/7/95
2	Jones	12/14/94
2	Jones	12/22/94
2	Jones	1/8/95
2	Jones	1/9/95
4	Williams	3/14/95
5		4/1/95

```
SELECT Patients.ID, Visit.VISITDATE
FROM Patients
RIGHT JOIN Visit on Paitients.ID = Visit.ID
```

Complex queries

Complex queries

- Most commonly used to retrieve data from two or more tables
- Typically accomplished with JOINs
  - Left join of two tables:

```
SELECT Patients.ID, Admission.admitdate
FROM Patients
LEFT JOIN Admission on Patients.ID = Admission.ID
```

  - Left join of three tables:

```
SELECT Patients.ID, Admission.admitdate, Radiology.normal
FROM Patients
LEFT JOIN Admission on Patients.ID = Admission.ID
LEFT JOIN Radiology on Admission.ID = Radiology.ID
```

Let's try it!

Assignment 8: Exercise 1

Subqueries

- Also called nested queries
- You can nest any number of queries
- The inner-most query is executed first, then in succession each in turn to the outer-most
- Queries can be nested inside a SELECT, INSERT, UPDATE, or DELETE statement or inside another subquery.
- A subquery is usually added within the WHERE Clause of another SQL SELECT statement

Example of a nested query

Patients		
patient_ID	dob	sex
1	10/1/1998	1
2	4/16/2004	2
3	3/7/1995	1
4	9/1/2000	1
5	11/15/1993	2

Admissions		
patient_ID	adm_date	hospital
1	10/1/1998	HUP
2	4/16/2004	PUPMC
3	3/7/1995	PAH
3	9/1/2000	PAH
4	11/15/1993	HUP
5	11/15/1993	PAH
5	8/1/1999	HUP
5	3/15/2003	PAH

```
SELECT patient_ID, dob FROM Patients WHERE
patientID=(SELECT patientID FROM Admissions WHERE year(adm_date) >= 2000);
```

Patients	
patient_ID	dob
2	4/16/2004
3	3/7/1995
5	11/15/1993

Let's try it!

Assignment 8: Exercise 2

Functions

NB: *x*=a literal or a field

- Numeric
  - Performs arithmetic operations on a single value, but not across rows in a table
  - Example: ROUND(*x,d*) rounds x to d decimal places
- String
  - Example: LENGTH(*x*) returns the length of *x*
- Date/Time
  - Example: DATE(*x1*, *x2,* INTERVAL *y*) returns the difference between the later date (*x1*) and the more recent (*x2*), in the units expressed in *y*

Let's try it!

Assignment 8: Exercise 3

Aggregate Functions

Perform across records

- Examples:
  - COUNT()
  - AVG()
  - STD()
  - MAX()
  - MIN()
  - SUM()
- Used with SELECT
  - SELECT COUNT(*) FROM Patients
    - Returns number of records in the Patient table

Let's try it!

Assignment 8: Exercise 4

The Entity-Attribute-Value model

"Typical" relational tables

Patients		
patient_ID	dob	sex
1	10/1/1998	1
2	4/16/2004	2
3	3/7/1995	1
4	9/1/2000	1
5	11/15/1993	2

Admissions		
patient_ID	adm_date	hospital
1	10/1/1998	HUP
2	4/16/2004	PUPMC
3	3/7/1995	PAH
3	9/1/2000	PAH
4	11/15/1993	HUP
5	11/15/1993	PAH
5	8/1/1999	HUP
5	3/15/2003	PAH

- Tables = Entities
- Rows represent an instance of an entity
- Columns represent specific attributes

However, sometimes this is a problem

- When you don't know how many columns to fix in a table
- When the data are likely to be sparse
- When you don't know how to model a relational database ☺

The Entity-Attribute-Value model

- Minimum of three columns in a single table!
  - Entity (table)
  - Attribute (name of the data element)
  - Value (the value of the attribute)
- Thus:

Entity	Attribute	Value
Patient	Gender	Female
Patient	BirthDate	3/10/1990
Lab	Hemoglobin	12.1
Lab	Hematocrit	35.7
Lab	Glucose	92
Lab	Potassium	4.3

Comparing traditional relational and EAV models

Relational

Patients

patient_ID	dob	sex
1	10/1/1998	1
2	4/16/2004	2
3	3/7/1995	1
4	9/1/2000	1
5	11/15/1993	2

Admissions

patient_ID	adm_date	hospital
1	10/1/1998	HUP
2	4/16/2004	PUPMC
3	3/7/1995	PAH
3	9/1/2000	PAH
4	11/15/1993	HUP
5	11/15/1993	PAH
5	8/1/1999	HUP
5	3/15/2003	PAH

EAV

Entity	Attribute	Value
Patient	patientID	1
Patient	dob	10/1/1998
Patient	sex	1
Admissions	adm_date	10/1/1998
Admissions	hospital	HUP
Admissions	adm_date	4/16/2004
Admissions	hospital	PUPMC
…	…	…

An example of a table in EAV format

Patient ID	Age (years)	Gender	Event Type	Event Name	Event Code	Start Date	End Date	Relevant Data
1031926	31	FEMALE	Diagnostic Test	Platelet	Platelet	3/19/2012		Result: 251 THO/uL,
1031926	31	FEMALE	Diagnostic Test	nrRBCPOP	nrRBCPOP	3/19/2012		Result: Unit of Measure Not Available,
1031926	31	FEMALE	Diagnostic Test	% Neutro Auto	% Neutro Auto	3/19/2012		Result: 65 %,
1031926	31	FEMALE	Diagnostic Test	MCHC	MCHC	3/19/2012		Result: 34 g/dL,
1031926	31	FEMALE	Diagnostic Test	AMC	AMC	3/19/2012		Result: 0 THO/uL,
1031926	31	FEMALE	Diagnostic Test	Chloride	Chloride	3/19/2012		Result: 100 mmol/L,
1031926	31	FEMALE	Diagnostic Test	Alk Phos	Alk Phos	3/19/2012		Result: 47 U/L,
1031926	31	FEMALE	Diagnostic Test	% Mono Auto	% Mono Auto	3/19/2012		Result: 6 %,
1031926	31	FEMALE	Diagnostic Test	ALT	ALT	3/19/2012		Result: 13 U/L,

If you wanted to obtain mean platelet counts across all patients, How would you do it?

Here are the issues for this query

- There is a record for each component of a lab test
- You would need many joins to retrieve each platelet instance, because they are spread out in separate rows
- The values have to be parsed from the units of measurement

Steps to success: 1

1. Import the table as-is into platelets_EAV

2. Create a new table:

```
CREATE TABLE platelets_new (
    patient_ID INT NOT NULL,
    start_date DATETIME,
    event_name VARCHAR(30),
    platelet_count FLOAT,
    PRIMARY KEY (patient_ID, lab_date));
```

Steps to success: 2

3. Convert and transfer the data:

```
INSERT INTO platelets_new
(patient_ID, start_date, event_name, platelet_count)
SELECT patient_ID,
        start_date,
        event_name,
        CASE WHEN event_name='Platelet' THEN
                CAST(relevant_data) AS
        FLOAT( SUBSTR(relevant_data,8,LOCATE(relevant_data,' ',9)
                IN relevant_data))
FROM platelets_EAV
```

How did SUBSTR and LOCATE work?

Patient ID	Age (years)	Gender	Event Type	Event Name	Event Code	Start Date	End Date	Relevant Data
1031926	31	FEMALE	Diagnostic Test	Platelet	Platelet	3/19/2012		Result: 251 THO/uL,
1031926	31	FEMALE	Diagnostic Test	nrRBCPOP	nrRBCPOP	3/19/2012		Result: Unit of Measure Not Available,
1031926	31	FEMALE	Diagnostic Test	% Neutro Auto	% Neutro Auto	3/19/2012		Result: 65 %,
1031926	31	FEMALE	Diagnostic Test	MCHC	MCHC	3/19/2012		Result: 34 g/dL,
1031926	31	FEMALE	Diagnostic Test	AMC	AMC	3/19/2012		Result: 0 THO/uL,
1031926	31	FEMALE	Diagnostic Test	Chloride	Chloride	3/19/2012		Result: 100 mmol/L,
1031926	31	FEMALE	Diagnostic Test	Alk Phos	Alk Phos	3/19/2012		Result: 47 U/L,
1031926	31	FEMALE	Diagnostic Test	% Mono Auto	% Mono Auto	3/19/2012		Result: 6 %,
1031926	31	FEMALE	Diagnostic Test	ALT	ALT	3/19/2012		Result: 13 U/L,

```
Result: 251 THO/uL
123456789012345678
```

1. SUBSTR(relevant_data,9,LOCATE(relevant_data,' ',9))
   ⇒ `251 THO/uL`
2. SUBSTR(relevant_data,9,LOCATE(relevant_data,' ',9))
   ⇒ `251`

For Assignment 9

1. Get the PennOmics data in EAV format .csv file from the Files folder on Canvas
2. The tasks for Assignment 9 are on Canvas
3. Try completing the assignment- it's not due until April 9, but we will discuss it in class on March 26 and April 2