

# Database Management: A brief Introduction

Brian L. Dos Santos

## Critical Role of Data

- Without data, an organization cannot function
  - particularly true in Ecommerce
- Initially, data was prepared for specific applications
  - payroll data for the payroll system
  - parts lists for the bill of materials system
  - sales data for statistical analysis
- By 1970, clear that data had common properties
- Data for many applications could be stored together in an organized way
  - database instead of separate collections

## What is a Database?

- No formal definition
- A collection of related data allowing:
  - insert (add new data)
  - delete (delete existing data)
  - update (change existing data = delete + insert)
  - query (retrieve all data having a certain property)
- What does “related” mean?

## The Relational Model

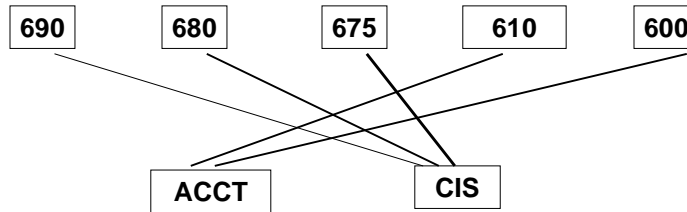
- A set is a collection of unique items  
{ John Paul, John Paul II, John XXIII, Pius XII } Popes since WWII  
{ John Paul, John XXIII, John Paul, John XXIII } NOT A SET
- A relation on two sets A, B is a set of pairs of elements, one from A and one from B  
 $A = \{ 680, 600, 675, 610, 690 \}$   
 $B = \{ \text{CIS}, \text{ACCT} \}$   
 $R = \{ (680, \text{CIS}), (600, \text{ACCT}), (610, \text{ACCT}), (690, \text{CIS}), (675, \text{CIS}) \}$

## The Relational Model

$A = \{ 680, 600, 675, 610, 690 \}$

$B = \{ \text{CIS}, \text{ACCT} \}$

$R = \{ (680, \text{CIS}), (600, \text{ACCT}), (610, \text{ACCT}), (690, \text{CIS}), (675, \text{CIS}) \}$



This is the graph of the relation R

## The Relational Model

$A = \{ 680, 600, 675, 610, 690 \}$

$B = \{ \text{CIS}, \text{ACCT} \}$

$R = \{ (680, \text{CIS}), (600, \text{ACCT}), (610, \text{ACCT}), (690, \text{CIS}), (675, \text{CIS}) \}$

COURSE	SCHOOL
680	CIS
600	ACCT
610	ACCT
690	CIS
675	CIS

CONTAINS ONLY COURSE NUMBERS

CONTAINS ONLY SCHOOL NAMES

This is a table of the relation R

## The Relational Model

Relations are not necessarily binary. May involve many sets:

COURSE	REQ'D	ROOM	#	FACULTY	DEPT
610	Y	123	35	Attaway	ACCT
680	N	123	35	Dos Santos	CIS
600	Y	122	30	Baxendale	ACCT
675	Y	146	35	Dos Santos	CIS
690	N	250	33	Wright	CIS

- Each row is a 6-tuple. Relation on 6 sets.
- No implied ordering of either rows or columns. Sorting is irrelevant
- Note: bad table design since "DEPT" is an attribute of "FACULTY", not "COURSE"

## Tables

- A relation can also be represented as a table
- One row for each tuple in the relation
- Table has implicit order (of rows and columns)
  - But: a relation has no ordering, either of tuples or attributes
- The cardinality of a relation R is the number of tuples it contains = # of rows in its table
- Relational model represents data as a collection of unordered two-dimensional tables

## Keys

- Key: an attribute (or minimum set of attributes) that uniquely defines a tuple
  - In the example relation, “Course” is a key. Note that with Course as the Key, we cannot have multiple sections of the same course.
- A relation may have a key that is made up of multiple attributes.
- A set of attributes that can serve as a key is a candidate key.
- One is chosen as the primary key.
- Keys are used to reference (retrieve) tuples.

## Foreign Keys

- A key from one relation that is an attribute of another relation is a foreign key.
- If we had a “Faculty” relation, then “Faculty” would be a foreign key in the “Courses” relation.
- Foreign keys connect relations together.

COURSE	REQ'D	ROOM	#	FACULTY
610	Y	123	35	Attaway
680	N	123	35	Dos Santos
600	Y	122	30	Baxendale
675	Y	146	35	Dos Santos
690	N	250	33	Wright

FACULTY	DEPT
Attaway	ACCT
Dos Santos	CIS
Baxendale	ACCT
Dos Santos	CIS
Wright	CIS

## Operations on Relations

- Projection  
List specific attributes L (columns) of R  
e.g. show course number and room

COURSE	REQ'D	ROOM	#	FACULTY	DEPT
610	Y	123	35	Attaway	ACCT
680	N	123	35	Dos Santos	CIS
600	Y	122	30	Baxendale	ACCT
675	Y	146	35	Dos Santos	CIS
690	N	250	33	Wright	CIS

COURSE	ROOM
610	123
680	123
600	122
675	146
690	250

## Operations on Relations

- Selection (extract horizontal slices)
  - List all tuples of relation R whose attributes satisfy condition C
  - E.g. show all tuples with Room = 123

COURSE	REQ'D	ROOM	#	FACULTY	DEPT	FACULTY
610	Y	123	35	Attaway	ACCT	Attaway
680	N	123	35	Dos Santos	CIS	Dos Santos

- Projection & Selection are single table operations

## Structured Query Language (SQL)

- A standard language for manipulating relational databases
- SELECT queries the database
- UPDATE modifies relations
- DELETE removes tuples

### Syntax of the SQL SELECT command:

```
SELECT { attributes }  
FROM { table }  
WHERE { attribute-conditions };
```

## Structured Query Language (SQL)

- Projection
  - Show course number, room
  - SQL: SELECT CourseNo, Room FROM Courses;
- Selection
  - Show all courses in Room 123
  - SELECT \* FROM Courses WHERE Room= "123";

YIELDS DISTINCT  
TUPLES SINCE  
CourseNo  
IS A KEY

MUST ASK FOR DISTINCT TUPLES  
SINCE Room IS NOT A KEY

## Cartesian Product A x B

- Has all attributes of A and all attributes of B

City	State	Pop	% For
Seattle	WA	532900	13.1
Detroit	MI	1027974	3.4

X

Animal	Type
Snake	Reptile
Dog	Mammal
Man	Mammal

=

City	State	Pop	% For	Animal	Type
Seattle	WA	532900	13.1	Snake	Reptile
Detroit	MI	1027974	3.4	Dog	Mammal
Seattle	WA	532900	13.1	Man	Mammal
Detroit	MI	1027974	3.4	Snake	Reptile
Seattle	WA	532900	13.1	Dog	Mammal
Detroit	MI	1027974	3.4	Man	Mammal

$$C(A \times B) = C(A) C(B)$$

## Join

- Cartesian product is often not meaningful. There may be no connection between attributes of a tuple
  - Cities and animals?
- In some cases, the attribute names and values match
- The natural join  $A * B$  consists of tuples with matching attributes (names & values) in **A** and **B**
- Natural join is a way of obtaining information across tables

## Natural Join A \* B

- Attribute names and values must match

City	State	Pop	% For		City	Area	Elevation	Radio
Seattle	WA	532900	13.1	*	Seattle	84	14	40
Detroit	MI	1027974	3.4		Atlanta	130	1050	27
					Detroit	139	601	59

	City	State	Pop	% For	Area	Elevation	Radio
=	Seattle	WA	532900	13.1	84	14	40
	Detroit	MI	1027974	3.4	139	601	59

- Also called “inner join”
- Cartesian product and join are binary operations

## Database Constraints

- **Domain (data validity) constraints**
  - All values in a column must be from the same domain
  - Example: all salaries are positive numeric dollar amounts. “Monthly” is invalid.
- **Entity Integrity**
  - Every entity must have a unique primary key. (Otherwise, can’t access the entity)
- **Referential Integrity**
  - Every foreign key value in a relation must match a primary key in the foreign relation

## RDBMS (Relational Database Management System)

- Provides table definition facilities
- Maintains tables
- Maintains indexes to tables for fast access
- Maintains database integrity
- Provides a language for defining queries
- Report generator for data summarization and output
- Interfaces to other software

## Functional Dependency

- **Attribute B is functionally dependent on attribute A if the value of A uniquely determines B**
  - One-to-one relationship: two functional dependencies: A depends on B; B depends on A
  - Many-to-one relationship: one functional dependencies: B depends on A
  - Many-to-many relationship: no dependencies: neither A nor B depends on the other
- **Functional dependencies are constraints between attributes or sets of attributes. They must be maintained or error or inconsistency will result.**

## Normalization

- A relation is well-structured if it is non-redundant and allows INSERT, MODIFY and DELETE without error or inconsistency.
- Normalization assists in maintaining functional dependencies and preventing errors and inconsistencies.
- DELETE anomaly:

Student	Email	Course	Room
Dossantos	dossantos@louisville.edu	675	123
Dossantos	dossantos@louisville.edu	680	123
Attaway	Attaway@louisville.edu	600	250

- Deleting "Attaway" removes all information about course 600 (namely that its room is 250)
- In the information is in another table, it shouldn't be here also.

## Normalization

- MODIFY anomaly:

Student	Email	Course	Room
Dossantos	dossantos@louisville.edu	675	123
Dossantos	dossantos@louisville.edu	680	123
Attaway	Attaway@louisville.edu	600	250

- Suppose Dossantos's email address changes. Every line in the table corresponding to Dossantos must be changed or data will be inconsistent.
- An attribute unique to a key should be entered only once in the database.

# Normalization

- Restructuring to produce smaller, well-structured equivalent relations, reduce data replication
- First Normal Form. Make all attributes atomic. No multiple values.

Name	Phone	Dept	Bldg
Wright	83064, 87279	CIS, ACCT	CBPA, ARH
Guan	82597, 87170	CIS, MKTG	CBPA

MULTIPLE VALUES (pointing to Phone, Dept, Bldg in the first row)

MULTIPLE VALUES (pointing to Bldg in the second row)

FIRST NORMAL FORM:

Name	Phone	Dept	Bldg
Wright	83064	CIS	CBPA
Wright	87279	ACCT	ARH
Guan	82597	CIS	CBPA
Guan	87170	MKTG	CBPA

# Second Normal Form

- Eliminate partial functional dependencies. Every non-key attribute must depend on all key attributes (or redundancy can result).

KEY IS (City, State)

Capital DEPENDS ON State ONLY, NOT CITY

NOT IN 2NF:

City	State	Capital	City Pop.
Philadelphia	PA	Harrisburg	1478002
Pittsburgh	PA	Harrisburg	1336449
Detroit	MI	Ann Arbor	1027974

2NF: DECOMPOSE INTO TWO TABLES

City	State	City Pop.
Philadelphia	PA	1478002
Pittsburgh	PA	1336449
Detroit	MI	1027974

State	Capital
PA	Harrisburg
MI	Ann Arbor

## More Normal Forms

- 3NF: No transitive dependencies (between non-key attributes).
- Bryce-Codd NF: Every determinant (left-hand side of a functional dependency) is a candidate key.
- 4NF: No multivalued dependencies.
- 5NF: No lossless JOIN decomposition into two relations. (lossless join = relations that when joined do not result in addition of tuples).
- DKNF (Domain-key Normal Form): Free of modification anomalies. Every constraint is a logical consequence of the definition of keys and domains

## Distributed Databases

- Databases in which data is stored in more than one location but appears local to the user
  - Replicated: multiple copies of database
  - Partitioned: data is split among locations
- Fragmentation
  - Information about fragments is stored in a distributed data catalog (DDC)
  - Horizontal v. vertical fragmentation

## Distributed Databases

- Advantages
  - Reduced load on central DB
  - Lower cost (data spread among small machines)
  - Reliability (machine failure is not fatal)
  - Fast access to local data
  - Ease of growth
- Disadvantages
  - Complexity. Difficult to maintain consistency
  - Security (many access points)
  - Telecommunications required

## Summary

- This is a very basic introduction to relational databases & database management
- Gives you an idea as to some of the issues & problems
- Certainly not enough to design a database in a commercial environment (little knowledge is dangerous).
- Learning to create tables in access does not qualify one to design a database that can be used in a production environment
- Need to (at least) master the topics covered in 3 credit hour course before one can begin to try to design a commercial database.