Database Systems

The Relational Model
A Simplified Introduction

Topics

- Brief Background
- SQL vs Model Terms
- Common Terminology
  - Tables, Columns, Rows
  - Primary Keys
  - Foreign Keys
  - Relationships / Referential Integrity
  - Normalization - SKIP for the time being

Background

A Relational Model of Data for Large Shared Data Banks

E. F. Codd
IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed

Background

- Set Theory : foundation of the relational model
- Think in terms of sets of data rather than individual items or rows of data
- Relational algebraic operations (e.g. union, intersect, product, join, etc)
- Tables are analogues of sets: collection of distinct elements having common properties

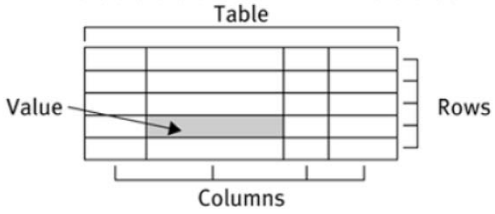
SQL vs Model Terms

Equivalent Terms

SQL	Model	~~Files~~
Table	Relation	~~File~~
Column	Attribute	~~Field~~
Row	Tuple	~~Record~~

Tables

- Each value is atomic (only 1 single value) or unknown (NULL)



Tables

- Example of an actual table

id	fname	lname
A01	Sarah	Buchman
A02	Wendy	Heydemark
A03	Hallie	Hull
A04	Klee	Hull
A05	Keith	Tang
A06		Johnson
A07	John	Doe
.		
.		

Tables

- Database is a collection of >=1 tables
- Uniquely named database structure that holds data as a collection of values
- Contains data about one entity type (e.g. student, course, enrollment, etc)
- 2-dimensional: columns and rows
- >= 1 columns
- >= 0 rows (0 row = empty table)
- **Columns and rows are unordered**

Columns

- Each column represents a specific attribute (or property) of the table's entity type
- Each column has a domain that restricts the set of values allowed - constraints
- Entries in columns are single-valued (atomic)
- Order of columns (left to right) is unimportant
- Each column is uniquely named

Rows

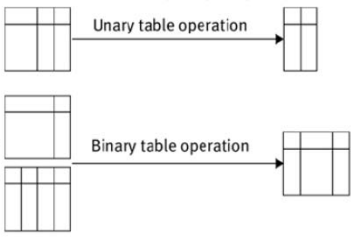
- Each row describes a fact about an entity (i.e. unique instance of an entity type)
- Each row contains a value or NULL for each of the table's columns
- Order of rows are unimportant
- No two rows in a table can be identical (model requirement, but not SQL)
- Each row is uniquely identified by its primary key

Tips - Relevant Facts

- DBMS uses 2 types of tables:
  User tables - for user data
  System tables - for metadata
- System Catalog = all the system tables
- **Closure** property of tables
- SQL Standard defines a hierarchy of RDB structures : catalog.schema.objects
- DBMSes use other terms for the same concepts, and vice versa

Closure Property of Tables

- Guarantees that result is always a table -> allows nesting of operations



Hierarchy of RDB Structures

Object References	
Platform	**Address**
Standard SQL	catalog.schema.object
Access	database.object
SQL Server 2008	server.database.schema.object
Oracle 11g	schema.object
DB2	schema.object
MySQL	Database.object
PostgreSQL	Database.schema.object

Primary Key – Entity Integrity

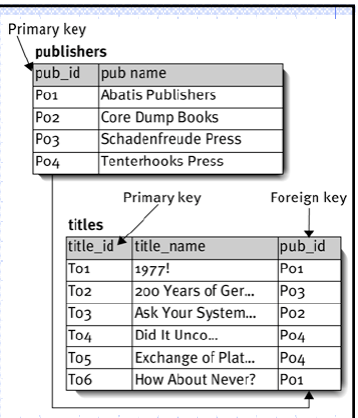
- A primary key is required for each table to access one single row and its column values
- **Entity Integrity** : Each table has at most **one** primary key [PK]
  - **Unique** : no 2 rows can have the same primary key value
  - **Not null**
- Key : Simple (1 column) or composite (>1)
- Minimal : minimum number of columns as necessary for uniqueness

Tips - Relevant Facts

- Use of artificial keys instead of common unique identifiers
- Candidate keys : multiple choices of unique identifiers in a table
- One of the candidate keys is designated as primary key
- Alternate keys : all the remaining candidate keys
- e.g. identifying a book in the library

Foreign Keys [FK]

- Mechanism in Relational DB to associate tables
- Relationship
- **Referential Integrity**



Foreign Keys [FK]

- Column(s) in one table which values, if any, relate to or reference the primary key or unique key values in other table(s) or its own table
- Child table - contains FK column(s)
  Parent table - has PK or unique column
- FK values generally are not unique in their own table
- A table can have 1 PK and >= 0 FKs

Referential Integrity

- Foreign key values in child table must correspond to existing value in parent table, or NULL
- PK (parent) and FK (child) columns can have different names, but must have the same domain
- Self-referencing table:
  FK in one table references PK of same table

Relationships

- Association established between common columns in two tables
- Can be self-referencing
- Relationship between tables can be:
  - One-to-One
  - One-to-Many
  - Many-to-Many
- The One / Many refers to the possible maximum number of rows allowed to exist in that table

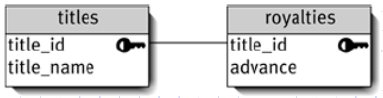
One-to-One

titles

title_id	title_name
T01	1977!
T02	200 Years of Ger...
T03	Ask Your System...
T04	But I Did It Unco...

royalties

title_id	advance
T01	10000
T02	1000
T04	20000



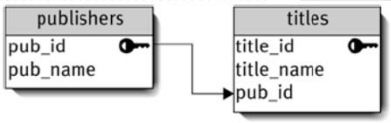
One-to-Many

publishers

pub_id	pub name
P01	Abatis Publishers
P02	Core Dump Books
P03	Schadenfreude Press
P04	Tenterhooks Press

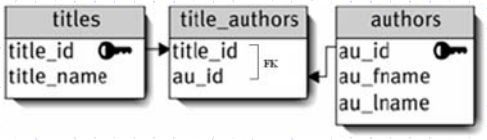
titles

title_id	title_name	pub_id
T01	1977!	P01
T02	200 Years of Ger...	P03
T03	Ask Your System...	P02
T04	But I Did It Unco...	P04
T05	Exchange of Plati...	P04

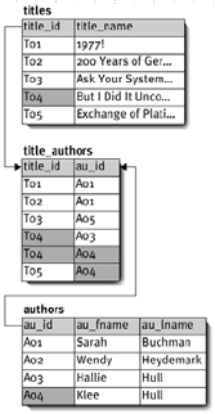


Many-to-Many

- Cannot be implemented directly in the relational model
- Need to be implemented with a pair of one-to-many relationships by adding a Junction or Associating table



Many-to-Many



titles

title_id	title_name
T01	1977!
T02	200 Years of Ger...
T03	Ask Your System...
T04	But I Did It Unco...
T05	Exchange of Plati...

title_authors

title_id	au_id
T01	A01
T02	A01
T03	A05
T04	A03
T04	A04
T05	A04

authors

au_id	au_fname	au_lname
A01	Sarah	Buchman
A02	Wendy	Heydemark
A03	Hallie	Hull
A04	Klee	Hull

Normalization

- Defer detail discussion in this topic until the second half of this term when we will be discussing Database Design
- Can pre-read these pages