4010-350 Personal SE

Introduction to Relational Database Systems

Historical Background

- Early database systems were ad hoc
  - ISAM – Indexed Sequential Access Method
  - Networks of records in a graph
- Regularization by E. Codd
  - Saw that databases were relations that could be operated upon by a small set of operators.
  - Relational algebra & relational calculus.
- One time when theory won:
  - Relational databases easier to understand.
  - Easier to implement and optimize.
- Most production databases today are relational.

The Basics

- A *relational database* has one or more *relations*.
- Each relation is represented by a 2D *table*.
- Each table has a fixed number of *columns*.
- Each column has a *name* for the elements in the column.
- Each column also has a *type* (say string, integer, date, etc.)
- Each table has a varying number of *rows*, with one entry per column per row.
- A rough analogy:
  - A relation is like a class.
  - A column is like an instance variable.
  - A row is like an object in the class.

Example – Baseball Database

Players

id	name	team_id	position
1	D. Jeter	1	SS
2	A.Rodriguez	1	3B
3	J. Varitek	2	C
4	M. Ramirez	3	OF
5	M. Ramirez	2	OF
6	C. Delgado	5	1B
7	D. Wright	5	3B
8	E. Longoria	4	3B

Teams

id	name	ciy
1	Yankees	New York
2	Red Sox	Boston
3	Dodgers	Los Angeles
4	Rays	Tampa Bay
5	Mets	New York

```
$ sqlite3 baseball.sqlite3
SQLite version 3.6.2
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
```

```
create table Players (id integer primary key, name text,  team_id integer, position text) ;
```

```
create table Teams (id integer primary key, name text, city text) ;
```

Example – Adding Rows

Players

id	name	team_id	position
1	D. Jeter	1	SS
2	A.Rodriguez	1	3B
3	J. Varitek	2	C
4	M. Ramirez	3	OF
5	M. Ramirez	2	OF
6	C. Delgado	5	1B
7	D. Wright	5	3B
8	E. Longoria	4	3B
9	K. Youkilis	2	1B

Teams

id	name	city
1	Yankees	New York
2	Red Sox	Boston
3	Dodgers	Los Angeles
4	Rays	Tampa Bay
5	Mets	New York

```
-- This is a comment
-- Insert Kevin Youkilis – the primary key is autoincremented to ensure uniqueness.
insert into Players (name, team_id, position) values ('K. Youkilis', 2, '1B');
```

Example – Listing a Table

Players

id	name	team_id	position
1	D. Jeter	1	SS
2	A.Rodriguez	1	3B
3	J. Varitek	2	C
4	M. Ramirez	3	OF
5	M. Ramirez	2	OF
6	C. Delgado	5	1B
7	D. Wright	5	3B
8	E. Longoria	4	3B
9	K. Youkilis	2	1B

Teams

id	name	city
1	Yankees	New York
2	Red Sox	Boston
3	Dodgers	Los Angeles
4	Rays	Tampa Bay
5	Mets	New York

```
-- List the Players table
select * from Players ;

-- List the Teams table
select * from Teams ;
```

Operator: Select

Select some rows based on criteria

Players

id	name	team_id	position
1	D. Jeter	1	SS
2	A.Rodriguez	1	3B
3	J. Varitek	2	C
4	M. Ramirez	3	OF
5	M. Ramirez	2	OF
6	C. Delgado	5	1B
7	D. Wright	5	3B
8	E. Longoria	4	3B
9	K. Youkilis	2	1B

Teams

id	name	city
1	Yankees	New York
2	Red Sox	Boston
3	Dodgers	Los Angeles
4	Rays	Tampa Bay
5	Mets	New York

```
-- List the Players table
select * from Players where position = "3B" ;

-- List the Teams table
select * from Teams where city = "New York";
```

Operator: Project

Choose only some of the columns

Players

id	name	team_id	position
1	D. Jeter	1	SS
2	A.Rodriguez	1	3B
3	J. Varitek	2	C
4	M. Ramirez	3	OF
5	M. Ramirez	2	OF
6	C. Delgado	5	1B
7	D. Wright	5	3B
8	E. Longoria	4	3B
9	K. Youkilis	2	1B

Teams

id	name	city
1	Yankees	New York
2	Red Sox	Boston
3	Dodgers	Los Angeles
4	Rays	Tampa Bay
5	Mets	New York

```
-- List the name and position of each player
select name, position from Players ;

-- List the name and city of each team
select name, city from Teams ;
```

Operator: Join

Match rows from one table against another

Players

id	name	team_id	position
1	D. Jeter	1	SS
2	A.Rodriguez	1	3B
3	J. Varitek	2	C
4	M. Ramirez	3	OF
5	M. Ramirez	2	OF
6	C. Delgado	5	1B
7	D. Wright	5	3B
8	E. Longoria	4	3B
9	K. Youkilis	2	1B

Teams

id	name	city
1	Yankees	New York
2	Red Sox	Boston
3	Dodgers	Los Angeles
4	Rays	Tampa Bay
5	Mets	New York

Foreign Key Column

```
-- List all combinations of Players & Teams (not very useful)
select * from Players, Teams ;

-- List combinations where Player's team_id = Teams' id (join & select)
select * from Players, Teams where Players.team_id = Teams.id ;
```

Exercises
Join / Select / Project

Players

id	name	team_id	position
1	D. Jeter	1	SS
2	A.Rodriguez	1	3B
3	J. Varitek	2	C
4	M. Ramirez	3	OF
5	M. Ramirez	2	OF
6	C. Delgado	5	1B
7	D. Wright	5	3B
8	E. Longoria	4	3B
9	K. Youkilis	2	1B

Teams

id	name	city
1	Yankees	New York
2	Red Sox	Boston
3	Dodgers	Los Angeles
4	Rays	Tampa Bay
5	Mets	New York

```
-- List the names of the third basemen playing in New York
select Players.name from Players, Teams where Players.team_id = Teams.id and
    Players.position = "3B" and Teams.city = "New York" ;

-- List the cities of all the first basemen
select Teams.city from Players, Teams where Players.team_id = Teams.id and
    Players.position = "1B" ;
```

What About Players w/ Multiple Teams?

- Example: See Manny Ramirez
  - Started season with Boston
  - Ended season with L.A.
- Approach #1:
  - Duplicate records – what we have
  - That is, two or more records for a player – one per position.
  - Issue of redundancy (DRY violation).
- Approach #2:
  - Table for player
  - Table for team
  - Table *linking* players to teams

Linking Table Example

Players

id	name	position
1	D. Jeter	SS
2	A.Rodriguez	3B
3	J. Varitek	C
4	M. Ramirez	LF
5	C. Delgado	1B
6	D. Wright	3B
7	E. Longoria	3B
8	K. Youkilis	1B

PlayersTeams

player_id	team_id
1	1
2	1
3	2
4	2
4	3
5	5
6	5
7	4
8	2

Teams

id	name	city
1	Yankees	New York
2	Red Sox	Boston
3	Dodgers	Los Angeles
4	Rays	Tampa Bay
5	Mets	New York

Can do the same for other duplicates like players with multiple positions

SQLManager for Firefox

- Installation:
  - Visit **http://code.google.com/p/sqlite-manager/**
  - Click on the SQLiteManager_X.Y.Z.xpi link on the right side of the page under featured downloads.
  - Open with Firefox – this installs the tool. You may have to restart Firefox.
- Open the manager – it's in the Firefox Tools menu.
  - You can either open an existing database or create a new one.
  - As you manipulate the database, you'll see the SQL statements that are executed.
  - Joins require you to type the SQL in the Execute SQL tab.