

CS 580 Client-Server Programming
Fall Semester, 2000
Doc 20 SQL & Normalization
Contents

Some Data Modeling	2
Normalization	4
First Normal Form (1NF)	4
Example - Office Hours	4
Adding Values	6
Some Issues	8
Second Normal Form (2NF)	13
Third Normal Form (3NF)	17
Other Normal Forms	18
Indexes	19

References

MySQL, Paul DuBois, New Riders Publishing, 2000.

Oracle Design, Ensor & Stevenson, O'Reilly & Associates, Inc., 1997

MySQL Online Manual

<http://www.mysql.com/documentation/mysql/commented/manual.php>

Copyright ©, All rights reserved.

2000 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA.

OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

Some Data Modeling Terms

Entity

A distinct class of things about which something is known

Entity Occurrence

Particular instance of an entity class

In a database entity occurrences are records in a table

Attribute

An abstraction belonging to or characteristic of an entity

Primary Key (unique identifier)

An attribute (or set of attributes) that uniquely define an entity

Relationship

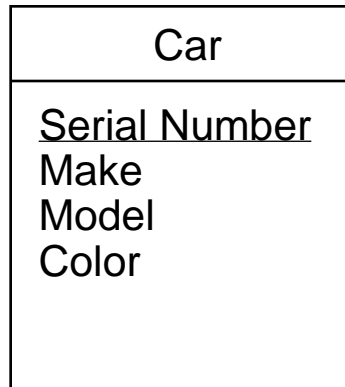
An abstraction belonging to or characteristic of two entities or parts together

Relational databases do not support pointers to entities

Foreign Key

A unique identifier in a record representing another record

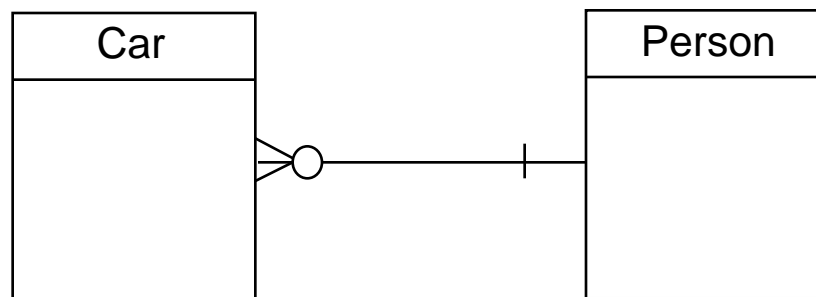
Entity Relationship Diagram (ERD)



Entity (car) with:

Attributes (Color, make, model, serial number)

Primary key (serial number)



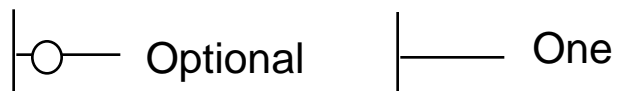
Relationship between Car and Person entities

Car must have one and only one owner

Person may own zero, one or more cars

Person can own many cars

Key



Normalization

Defined by Dr. E. F. Codd in 1970

Normal forms

Reduce redundant data and inconsistencies

First Normal Form (1NF)

An entity is in the first normal form when all its attributes are single valued

Example - Office Hours

Name	OfficeHour1	OfficeHour2	OfficeHour3
Whitney	10:00-11:00 W	17:00-18:30 Tu	15:00-16:00 Fri
Beck	8:00-12:00 M		
Anantha	9:00-10:30 Tu	9:00-10:30 Thu	

What if someone has more than 3 office hours?

Wasted space for those that have fewer office hours

Not is 1NF since office hours are repeated

Faculty

name	faculty_id
Whitney	1
Beck	2
Anantha	3

Office Hours

start_time	end_time	day	faculty_id	office_hour_id
10:00	11:00	Wed	1	1
8:00	12:00	Mon	2	2
17:00	18:30	Tue	1	3
9:00	10:30	Tue	3	4
9:00	10:30	Thu	3	5
15:00	16:00	Fri	1	6

```
CREATE TABLE faculty (  
    name CHAR(20) NOT NULL,  
    faculty_id INT UNSIGNED AUTO_INCREMENT NOT NULL  
    PRIMARY KEY  
)
```

```
CREATE TABLE office_hours (  
    start_time TIME NOT NULL,  
    end_time TIME NOT NULL,  
    day ENUM("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun") NOT  
    NULL,  
    faculty_id INT UNSIGNED NOT NULL,  
    office_hour_id INT UNSIGNED AUTO_INCREMENT NOT NULL  
    PRIMARY KEY  
)
```

Adding Values

```
INSERT INTO  
  faculty  
VALUES  
  ('Whitney', NULL),  
  ('Beck', NULL),  
  ('Anantha', NULL);
```

With positional data all columns must be given a value

If you do not like the idea of positional data you can list the columns

```
INSERT INTO faculty (faculty_id, name) VALUES (NULL, 'Frost');
```

```
INSERT INTO faculty ( name) VALUES ('Vinge');
```

```
INSERT INTO faculty SET name = 'Eckberg';
```

Columns listed get the default values,

Auto_increment columns get next count

See

<http://www.mysql.com/documentation/mysql/commented/manual.php?section=INSERT> for more information on the INSERT command

Office_Hours adding Simple Insert

```
INSERT INTO
  office_hours
SET
  start_time = '10:00:00',
  end_time = '11:00:00',
  day = 'Wed',
  faculty_id = 1;
```

The problem is that we need to know the id for the faculty

Using Select

```
INSERT INTO
  office_hours (start_time, end_time, day, faculty_id )
SELECT
  '8:00:00' AS start_time,
  '12:00:00' AS end_time,
  'Mon' AS day,
  faculty_id AS faculty_id
FROM
  faculty
WHERE
  name = 'Beck';
```

Some Issues

Adding Lots of Data

Adding data into tables via individual SQL commands is tedious

mysqlimport allows you to import files into a table

For more info and examples see:

<http://www.mysql.com/documentation/mysql/commented/manual.php?section=mysqlimport>

See

http://www.mysql.com/documentation/mysql/commented/manual.php?section=Loading_tables

for information about the LOAD DATA command

Using MySQL client

See

http://www.mysql.com/documentation/mysql/commented/manual.php?section=Entering_queries

MySql client input line editing

mysql client allows you to edit input lines

Key	Meaning
Up arrow, Ctrl-P	Recall previous line
Down arrow, Ctrl-N	Recall next line
Ctrl-A	Move to the beginning of line
Ctrl-E	Move to the end of line
Ctrl-K	Erase everything form cursor to end of line
Ctrl-_	Undo last change

Getting Office Hours

```
SELECT
    name, start_time, end_time, day
FROM
    office_hours, faculty
WHERE
    faculty.faculty_id = office_hours.faculty_id
```

name	start_time	end_time	day
Whitney	10:00:00	11:00:00	Wed
Beck	08:00:00	12:00:00	Mon
Whitney	17:00:00	18:30:00	Tue
Whitney	15:00:00	16:00:00	Fri
Anantha	09:00:00	10:30:00	Tue
Anantha	09:00:00	10:30:00	Thu

Some Formatting

```
SELECT
  name AS Instructor,
  CONCAT(
    TIME_FORMAT(start_time, "%r"), " to ",
    TIME_FORMAT(end_time, "%r")
  ) AS Time,
  day AS Day
FROM
  office_hours, faculty
WHERE
  faculty.faculty_id = office_hours.faculty_id
ORDER BY
  Name
```

Instructor	Time	Day
Anantha	09:00:00 AM to 10:30:00 AM	Tue
Anantha	09:00:00 AM to 10:30:00 AM	Thu
Beck	08:00:00 AM to 12:00:00 PM	Mon
Whitney	10:00:00 AM to 11:00:00 AM	Wed
Whitney	05:00:00 PM to 06:30:00 PM	Tue
Whitney	03:00:00 PM to 04:00:00 PM	Fri

Some Selection

```
SELECT
  name AS Instructor,
  CONCAT(
    TIME_FORMAT(start_time, "%r"), " to ",
    TIME_FORMAT(end_time, "%r")
  ) AS Time,
  day AS Day
FROM
  office_hours, faculty
WHERE
  faculty.faculty_id = office_hours.faculty_id
AND
  name = "Whitney"
ORDER BY
  start_time
```

Instructor	Time	Day
Whitney	10:00:00 AM to 11:00:00 AM	Wed
Whitney	03:00:00 PM to 04:00:00 PM	Fri
Whitney	05:00:00 PM to 06:30:00 PM	Tue

Second Normal Form (2NF)

An entity is in the second normal form if:

- It is in 1NF and
- All non-key attributes must be fully dependent on the entire primary key

Example 1- CDs

Put your collection of CD in a database

cd_title	artist	music_type	cd_id
Songs from the Trilogy	Glass	Modern Classical	1
I Stoten	Falu Spelmanslag	Swedish	2
Photographer	Glass	Modern Classical	3
etc.			

Table is not in 2NF since different CDs

- Can have the same artists
- Can have same music type

Example 2- Course Schedule

Name	Time	Days	Term	Schedule Number
CS635	1700-1815	MW	Spring01	09461
CS651	1700-1815	MW	Spring01	09472
CS672	1700-1815	MW	Spring01	09483
CS683	1830-1945	MW	Spring01	09494
CS696	1530-1645	MW	Spring01	09505
CS696	1830-1945	MW	Spring01	09516
CS696	1530-1645	TTh	Spring01	09520

At SDSU the schedule number uniquely identifies a course in a semester

So the term and schedule number uniquely identifies a course at SDSU

We can use term and schedule as the primary key

The table is in 1NF but not 2NF

Name, Time and Days are not fully dependent on the primary key

Schedule

course_id	time_id	term_id	schedule_number
1	1	2	09461
2	1	2	09472
3	1	2	09483
4	2	2	09494

Courses

course	title	name_id
CS635	Adv Obj Orient Dsgn Prog	1
CS651	Adv Multimedia Systems	2
CS672	Micro Computer Software	3
CS683	Emerging Technologies	4
CS696	Intell Systems & Control	5
CS696	Writing Device Drivers	6
CS696	Sem: Computer Security	7

Time

start_time	end_time	days	time_id
17:00:00	18:15:00	MW	1
18:30:00	19:45:00	MW	2
15:30:00	16:45:00	MW	3
15:30:00	16:45:00	TTh	4
Etc.			

Term

semester	year	term_id
Fall	2000	1
Spring	2001	2

Comments about Previous Slide

The schedule table is now in 2NF

What about the other tables?

If not how would you fix them?

Can you find a better way to decompose the original table?

Third Normal Form (3NF)

An entity is in third normal form if

- It is in 2NF and
- All non-key attributes must only be dependent on the primary key

Customer

Name	Address	City	State Name	State abbreviation	zip	id

State abbreviation depends on State Name

Table is not in 3NF

Other Normal Forms

- Boyce-Codd normal form (BCNF)
- Fourth normal form (4NF)
- Fifth normal form (5NF)

These are beyond the scope of this course

See your local database course/textbook

Indexes

Pro

- Improves retrieval time

Con

- Requires disk space
- Increases the time required to write/update records

Choosing Indexes

- Consider the type of operation on a column

<, <=, =, =>, >, BETWEEN use indexes if exist

Some LIKES use indexes

- Index columns that you search for

Columns in WHERE and join clauses are good candidates

- Use short indexes
- Use unique indexes
- Don't over index