

CCT396, Fall 2011

# Database Design and Implementation

Yuri Takhteyev  
University of Toronto



This presentation is licensed under Creative Commons Attribution License, v. 3.0. To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/>. This presentation incorporates images from the Crystal Clear icon collection by Everaldo Coelho, available under LGPL from <http://everaldo.com/crystal/>.

**Week 6**

**Normalization**

# Two Approaches

1. Start with something, then fix it
2. Start by modeling



Normalization

# What Problems?

problem type	cause	solution
update	redundancy	decomposition
insertion	redundancy	decomposition
deletion	redundancy	decomposition

# Normal Forms

5<sup>th</sup> Normal Form

4<sup>th</sup> Normal Form

BC Normal Form

3<sup>rd</sup> Normal Form

2<sup>nd</sup> Normal Form

1<sup>st</sup> Normal Form

fixing  
weird  
issues

you can do it!

trivial!

# 1NF

No multi-valued attributes

pet_id	owner	names
1	Bob	“Slim”, “the Serpent”, “Ribbon”
2	Gwen	“Fluffy”, “Big Dog”

# The Wrong Solution

pet_id	owner	name1	name2	name3
1	Bob	"Slim"	"the Serpent"	"Ribbon"
2	Gwen	"Fluffy"	"Big Dog"	



# The Right Solution

pet_id	owner
1	Bob
2	Gwen

pet_id	names
1	Slim
1	the Serpent
1	Ribbon
2	Fluffy
2	Big Dog

# 2NF and 3NF

Getting rid of  
“functional dependencies”

pet_id	pet_name	owner_email	owner_name
...	...	...	...
...	...	...	...

pet\_id → pet\_name, owner\_email,  
owner\_name

owner\_email → owner\_name

# Functional Dependency

**pet\_id → owner\_name**

“owner\_name is functionally depend on pet\_id”

“pet\_id functionally determines owner\_id”

“owner\_name describes pet\_id”

# Examples

**(movie\_id, title, year, rating)**

movie → title, year, rating

# Examples

**(building\_code, room\_no,  
session\_code, time\_slot,  
course\_code)**

building\_code, room\_no,  
session\_code, time\_slot →  
course\_code

# Examples

**(building\_code, room\_no,  
session\_code, time\_slot,  
course\_code, course\_title)**

building\_code, room\_no,  
session\_code, time\_slot →  
course\_code, course\_title

course\_code → course\_title

# Examples

**(building\_code, room\_no,  
session\_code, time\_slot,  
course\_code, course\_title,  
year, instructor\_utorid,  
instructor\_name)**



# Examples

**(building\_code, room\_no,  
session\_code, time\_slot,  
course\_code, course\_title,  
year, instructor\_utorid,  
instructor\_name, leap\_year,  
instructor\_employee\_no,  
instructor\_rank,  
num\_students, room\_capacity)**

# Full Dependency

**a full key → attribute**

(This is what we want to see.)

“Each attribute must describe the key, the whole key, and nothing but the key. So help me Codd.”

# Partial Dependency

**part of a key → non-key attribute**

(building\_code, room\_number,  
building\_name, capacity)

Keys: (building\_code, room\_number)

A partial dependency:

building\_code → building\_name

# Examples

(pet\_id, pet\_name, owner\_name)

(gadget\_id, agent\_id)

(city\_name, province\_code,  
province\_name, population)

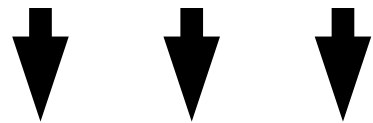
# 2<sup>nd</sup> Normal Form

No partial dependencies

# Getting to 2NF

## Decomposition:

(building\_code, room\_number,  
building\_name, capacity)



(building\_code, room\_number,  
capacity)

(building\_code, building\_name)

# Transitive Dependency

**non-key attribute → non-key attribute** (a simplified definition)

(pet\_id, pet\_name, species\_id, species\_name)

Keys: pet\_id

A transitive dependency:

species\_id → species\_name

# 3<sup>rd</sup> Normal Form

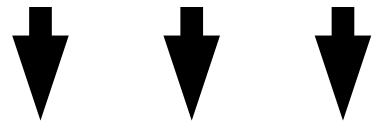
No transitive dependencies



# Getting to 3NF

## Decomposition:

(pet\_id, pet\_name, species\_id,  
species\_name)



(pet\_id, pet\_name, species\_id)  
(species\_id, species\_name)

# Examples

**(building\_code, room\_no,  
session\_code, time\_slot,  
course\_code, course\_title,  
year, instructor\_utorid,  
instructor\_name, leap\_year,  
instructor\_employee\_no,  
instructor\_rank,  
num\_students, room\_capacity)**

# Examples

(building\_code, room\_no, session\_code,  
time\_slot, course\_code, instructor,  
num\_students)

(building\_code, room\_no, room\_capacity)

(session\_code, year)

(year, leap\_year)

(course\_code, course\_title)

(instructor\_utorid, instructor\_name,  
instructor\_employee\_no, instructor\_rank)

# Better Yet

(room\_id, building\_code, room\_no,  
room\_capacity)

(room\_id, session\_code, time\_slot,  
course\_code, instructor, num\_students)

(session\_code, year)

(year, leap\_year)

(course\_code, course\_title)

(instructor\_utorid, instructor\_name,  
instructor\_employee\_no, instructor\_rank)

# Boyce-Codd NF

**“All determinants must be candidate keys.”**

(Closes a “loophole” in 3NF.)

# BCNF

(instructor\_id, instructor\_utorid,  
time\_slot)

Keys:

(instructor\_id, time\_slot)

(instructor\_utoroid, time\_slot)

instructor\_id  $\rightarrow$  instructor\_utorid

instructor\_utorid  $\rightarrow$  instructor\_id

(3NF, but not BCNF)

# 3NF vs. BCNF

**It is always possible to decompose a table into 3NF, but not always into BCNF.**

# 4NF

Similar to BCNF but based on multivalued dependencies ( $\twoheadrightarrow$ ).

Usually implies mistaking a binary relationships for ternary.

Examples tend to look contrived.

Not going to be on the exam.



# 5NF

We could break this table up and then re-create it with a join. (A simplified definition.)

Again, examples tend to look contrived.

Not going to be on the exam.

Questions?

# Alter Table: Add

```
alter table «table»  
add column «column» «type»;
```

---

```
alter table pet  
add column pet_id integer;
```

# Alter Table: Drop

```
alter table «table»  
drop column «column»;
```

---

```
alter table pet  
drop column pet_id;
```

# Alter Table: Modify

```
alter table «table»  
modify column «column» «type»;
```

---

```
alter table pet  
modify column pet_id char(5);
```

# Alter Table: Rename

```
alter table «old_name»  
rename «new_name»;
```

---

```
alter table pet  
rename animal;
```

# Adding Constraints

```
alter table «table»  
add constraint foreign key  
«foreign key constraint»;
```

---

```
alter table pet  
add constraint foreign key  
(owner_id) references  
owner(owner_id);
```

# Alternatively...

Drop the table.

Create a new one.



# The Final Project

Get started!

# Week 3 Exercises

Madame Z's Database Revisited