

CIS4301 Notes: Exam 2 Review

Ryan Roden-Corrent

March 21, 2014

1 Things to Review

- Normal Form: **1NF**, **2NF**, **3NF**
- Functional Dependencies
 - 6 Rules, **especially transitive**
- More queries
 - GROUP BY
 - NULLs
 - Aggregates (and their behavior with regards to NULL)
- Indices: When is it a good idea to use indexes?
 - For example, given a set of queries, use indices to speed it up
- Views: virtual vs materialized (how are they updated?)
- Foreign Keys, Check Constraints
- Constraints
 - Table Level
 - Tuple Level
- Triggers: After vs Before:
 - Before: Allows trigger to throw error state and stop further modification
 - After: Ok if not too much to undo

2 Chapter Breakdown

Ch.3, Ch.5-9 are on the test.

2.1 Stuff you don't need to study:

you DON'T need to know the following:
--

Ch.3 Closure FD, set cover, BCNF, 4NF, 5NF

Ch.5

Ch.6 6.6.4-6.6.6

Ch.7 7.4

Ch.8 8.4

Ch.9 9.1 - 9.3, 9.5-9.7

9.4 probably don't need to know stored procedures for quiz, but good to know

3 Indexes (Indices?)

An index can be thought of as a **hashtable** or **b-tree**.

hashtable quick lookups, point queries

- Point Query: Avoid scanning large set, go directly to record

b-tree Range Queries

- very flat, holds alot of information
- useful for e.g. get all items with *attribute > some_value*
- such a query would not work well with hashtable (repeated hash value computation)

3.1 Why not always use Indices?

Some databases kinda do (e.g. MongoDB). However, If you index everything, you are creating alot of duplication. Creating a specialized data structure for every column **consumes excessive space**, and **updates can take a long time**.

3.2 What if something is updated in database but not index

In general, must check if cached value is marked as "dirty".

4 Window Functions

Look them up on postgres docs, especially **rank()**.

Listing 1: rank example

```
SELECT X, avg(salary) OVER (PARTITION BY X)
FROM G
GROUP BY X;
```
