-- B. Crump, A. Jarrell

**Functional Dependencies and 3NF**

Person:

per_id —> name, address+, email, gender, zip_code
(left side is a primary key/ candidate key)
per_id —>—> phone (multi-valued attribute)
address+ —> zip_code (each address is associated with a zip_code)
name, address+, email, gender, zip_code —> per_id
(left side can also be a candidate key)

Company:

comp_id —> address+, zip_code, website, primary_sector
(left side is a primary key/ candidate key)
address+ —> zip_code (each address is associated with a zip_code)
comp_id —>—> specialty (multi-valued attribute)
address+, zip_code, website, primary_sector —> comp_id
(left side can also be a candidate key)

Course:

c_code —> title, level, description, status, retail_price
(left side is a primary key/ candidate key)
title —> description (each title is associated with a description)

Decompose into:
course1(c_code, title, level status, retail_price)
course2(title, description)
since description is not part of a key in course. When we intersect course1 and course2, we get title, which is the superkey of course2.

Section:

c_code, sec_no, semester, year —> offered_by, format, price, complete_date
(left side is a primary key/ candidate key)
c_code, sec_no, semester, year, offered_by, format —> price
(left side is associated with a price)

Job_Profile:

pos_code —> title, description, avg_pay
(left side is a primary key/ candidate key)
pos_code —>—> required_skill (multi-valued attribute)
title —> description (each title is associated with a description)

Decompose into:
jp1(pos_code, title, avg_pay)
jp2(title, description)
since description is not part of a key in job_profile. When we intersect jp1 and jp2, we get title, which is the superkey of jp2.

Job:

job_code —> type, pay_rate, pay_type, company, pos_code
(left side is a primary key/ candidate key)

Knowledge_Skills:

ks_code —> title, description, level
(left side is a primary key/ candidate key)
title —> description (each title is associated with a description)

Decompose into:
ks1(ks_code, title, level)
ks2(title, description)
since description is not part of a key in knowledge_skills. When we intersect ks1 and ks2, we get title, which is the superkey of ks2.

Job_Company:

job_code —> comp_id
job_code, comp_id —> (all attributes associated with each)

Teaches:

ks_code, c_code —> (all attributes associated with each)

Has_Job:

job_code —> per_id
per_id, job_code —> (all attributes associated with each), start_date, end_date

Skills:

ks_code, pos_code —> (all attributes associated with each)

Experience:

per_id, ks_code —> (all attributes associated with each)

Takes:

per_id, c_code, sec_no, semester, year —> (all attributes associated with each)

Offers:

comp_id, c_code, sec_no, semester, year —> (all attributes associated with each)