

# COSC344 Assignment 2 Report

---

Leader: Hayden McAlister

Members: Masaaki Fukushima, Jack Heikell, Nat Moore

## Entity-Relationship Model

- Student

Attribute	Simplicity	Num-Values	Data Type
Student_ID	Simple, Not NULL	Single-valued	Int (key Attribute)
Name	Simple, Not NULL	Single-valued	String
Phone	Simple	Single-valued	String
Address	Composite (Street_Number: int[1,10000], Street_Name: Str, Suburb: Str)	Multi-valued	String
Course	Simple	Multi-valued	String
Enrollment_Date	Simple, Not NULL	Single-valued	Date
Graduate_Date	Simple	Single-valued	Date
Graduated_bool	Derived (From existence of Graduate_Date)	Single-valued	Boolean

- Staff

Attribute	Simplicity	Num-Values	Data Type
Staff_ID	Simple, Not NULL	Single-valued	Int (Key Attribute)
Name	Simple, Not NULL	Single-valued	String
Phone	Simple	Single-valued	String
Address	Composite (Street_Number: int[1,10000], Street_Name: Str, Suburb: Str)	Multi-valued	String
Salary	Simple, Not NULL	Single-valued	Float
IRD_Num	Simple, not NULL	Single-valued	Int

- Department

Attribute	Simplicity	Num-Values	Data Type
Name	Simple	Single-valued	String (Key Attribute)
Campus	Simple	Derived (From Campus Reference)	String
Number_of_Employees	Composite (Number_of_academic_staff: int, Number_of_nonacademic_staff: int)	Single-valued	int
Number_of_Students	Derived (from Student references)	Single-valued	int
Address	Derived (from Building references)	Multi-valued	String

- Course

Attribute	Simplicity	Num-Values	Data Type
Name	Simple	Single-valued	String (Key Attribute)
Years_required	Simple	Single-valued	int

Attribute	Simplicity	Num-Values	Data Type
Undergraduate	Simple	Single-valued	boolean
Postgraduate	Simple	Single-valued	boolean
Number_of_Students	Derived (from Student references)	Single-valued	int

- Paper

Attribute	Simplicity	Num-Values	Data Type
Paper_Code	Simple	Single-valued	String (Key Attribute)
Semester	Simple	Multi-valued	String (Enumerated)
Points	Simple	Single-valued	Int

- Campus

Attribute	Simplicity	Num-Values	Data Type
Name	Single-valued	Single-valued	String (Key Attribute)
Main_Office_Address	Single-valued	Single-valued	String
Phone	Simple	Single-valued	String (Candidate Key)
Email	Simple	Single-valued	String (Candidate Key)

- Building

Attribute	Simplicity	Num-Values	Data Type
Address	Composite (Street_Number: int[1,10000], Street_Name: Str, Suburb: Str)	Single-valued	String (Key Attribute)
Postcode	Simple	Single-valued	Int (Four Digit)
Name	Simple	Single-valued	String

- Room

Attribute	Simplicity	Num-Values	Data Type
Room Number	Simple	Single-valued	Int, [1,10000] (Partial Key)
Seating	Simple	Single-valued	Int [1,10000]
Accessibility	Simple	Single-valued	Boolean
Projector	Simple	Single-valued	Boolean

## Mapping to Relational Model

### Step 1: Mapping Regular Entity Types

#### BUILDING

- Decompose composite attribute and add all simple attributes, add weak key to primary key

<u>Street Number</u>	<u>Street Name</u>	<u>Suburb</u>	Postcode	Name
----------------------	--------------------	---------------	----------	------

#### DEPARTMENT

<u>Name</u>	Number_Of_Academic_Staff	Number_Of_Nonacademic_Staff
-------------	--------------------------	-----------------------------

#### COURSE

<u>Name</u>	Years_Required	Postgraduate_Bool	Coordinator_id (REFERENCES Staff)
-------------	----------------	-------------------	-----------------------------------

## PAPER

<u>Paper_code</u>	Semester (Multi-value)	Points
-------------------	------------------------	--------

## CAMPUS

<u>Name</u>	Street_Number	Street_Name	Suburb	Phone	Email
-------------	---------------	-------------	--------	-------	-------

## STUDENT

- Decompose composite attributes and add all simple attributes.

<u>Student_ID</u>	Name	Phone	Street_Number	Street_Name	Suburb	Enrollment	Graduation	Graduated
-------------------	------	-------	---------------	-------------	--------	------------	------------	-----------

## STAFF

- Decompose composite attributes and add all simple attributes.

<u>Staff_ID</u>	Name	Phone	Street_Number	Street_Name	Suburb	Salary	IRD_Num
-----------------	------	-------	---------------	-------------	--------	--------	---------

## Step 2: Mapping Weak Entity Types

## ROOM

- Add as primary key reference the primary key of building

<u>Street_Number</u> (REFERENCES Building)	<u>Street_Name</u> (REFERENCES Building)	<u>Suburb</u> (REFERENCES Building)	<u>Room_Number</u>	Seating	Accessibility	Projector
---	---	--	--------------------	---------	---------------	-----------

## Step 3: Mapping of binary 1:1 Relationships

## DEAN\_OF (Campus 1:1 Staff)

- The Staff\_ID is added to the campus table to represent a DEAN\_OF relationship, as campus has total participation.

## CAMPUS

<u>Name</u>	Main_Office_Address	Phone	Email	Dean (REFERENCES Staff)
-------------	---------------------	-------	-------	-------------------------

## COORDINATES (Course 1:1 Staff)

- Staff\_ID is added to course as staff members have total participation in the COORDINATES relationship.

## COURSE

<u>Name</u>	Years_Required	Postgraduate_Bool	Coordinator_id (REFERENCES Staff)
-------------	----------------	-------------------	-----------------------------------

## Step 4: Mapping of Binary 1:N Relationships

## OFFERS (Department 1:N Paper)

- Add as foreign key to Paper the primary key of Department

## PAPER

<u>Paper_Code</u>	Semester	Points	Department_Name (REFERENCES Department)
-------------------	----------	--------	--

- The department name is added to the paper table to represent any number of papers belonging to a single department.

## LOCATED\_ON (Building N:1 Campus)

- Add as foreign key to Building the primary key of Campus

## BUILDING

<u>Street Number</u>	<u>Street Name</u>	<u>Suburb</u>	<u>Postcode</u>	<u>Building Name</u>	<u>Campus Name</u> (REFERENCES Campus)
----------------------	--------------------	---------------	-----------------	----------------------	---

OFFICE\_OF (Room 1:N Staff)

- Create a new table OFFICE\_OF that includes the primary key of staff as primary and foreign key, and primary key of room as foreign key
- This avoids many NULLs for staff with no office

OFFICE\_OF

<u>Staff ID</u> (REFERENCES Staff)	<u>Street Number</u> (REFERENCES Building)	<u>Street Name</u> (REFERENCES Building)	<u>Suburb</u> (REFERENCES Building)	<u>Room Number</u>
---------------------------------------	---	---	--	--------------------

LOCATED\_IN (Room N:1 Building)

- Add as foreign key to Room the primary key of Campus
- Already done in step 2 (weak entity mapping), so no extra work is needed

STUDENT\_AT (Student N:1 Campus)

- Reference for the campus a student is located at/in.

STUDENT

<u>Student ID</u>	<u>Name</u>	<u>Phone</u>	<u>Street Number</u>	<u>Street Name</u>	<u>Suburb</u>	<u>Enrollment</u>	<u>Graduation</u>	<u>Graduated</u>	<u>Campus</u>
-------------------	-------------	--------------	----------------------	--------------------	---------------	-------------------	-------------------	------------------	---------------

STAFF\_AT (Student N:1 Campus)

- Reference for the campus a staff member is located at/in.

STAFF

<u>Staff ID</u>	<u>Name</u>	<u>Phone</u>	<u>Street Number</u>	<u>Street Name</u>	<u>Suburb</u>	<u>Salary</u>	<u>IRD_Num</u>	<u>Campus</u>
-----------------	-------------	--------------	----------------------	--------------------	---------------	---------------	----------------	---------------

SUPERVISES (Staff M:1 Student)

- We decided to change this into it's own supervises entity, to eliminate excessive NULL's if many students did not have supervisors.

STAFF\_SUPERVISES\_STUDENT

<u>Staff ID (REFERENCES Staff)</u>	<u>Student ID (REFERENCES Student)</u>
------------------------------------	--

## Step 4.5: Mapping of Binary 2:N Relationships

ENROLLED\_IN (Student 2:N Course)

- Though originally Enrolled\_In was going to be handled by data fields within the Student entity, we decided to model enrollment through a separate entity. This is so a student can enroll in more than one course.

ENROLLED\_IN

<u>Student ID</u>	<u>Course</u>
-------------------	---------------

## Step 5: Mapping of Binary M:N Relationships

BASED\_IN (Building M:N Department)

- Create new Relation with primary key of each related entity

DEPT\_BASED\_IN\_BUILDING

<u>Dept Name</u> (REFERENCES Department)	<u>Street Number</u> (REFERENCES Building)	<u>Street Name</u> (REFERENCES Building)	<u>Suburb</u> (REFERENCES Building)
---	---	---	--

LECTURED\_IN (Room M:N Paper)

- Create new Relation with primary key of each related entity

## PAPER\_LECTURED\_IN\_ROOM

<u>Paper Code</u> (REFERENCES Paper)	<u>Street Number</u> (REFERENCES Building)	<u>Street Name</u> (REFERENCES Building)	<u>Suburb</u> (REFERENCES Building)	<u>Room Number</u>
---	---	---	--	--------------------

## WORKS\_FOR(Staff M:N Department)

- Create new Relation with primary key of each related entity
- Originally it was decided that staff members could work for multiple departments, but for realism and ease of use, we have since agreed staff members may only work for one department, and combination departments could be added where necessary. A cleaner would work for the cleaning department, rather than each department that they clean, for example. This is represented below:

## STAFF

<u>Staff ID</u>	Name	Phone	<u>Street Number</u>	<u>Street Name</u>	<u>Suburb</u>	Salary	IRD_Num	Campus	<u>Supervises_Student</u> (REFERENCES Student)	<u>Department</u> (REFERENCES Department)
-----------------	------	-------	----------------------	--------------------	---------------	--------	---------	--------	---	--

## COUNTS\_TOWARD(Paper M:N Course)

- Create new Relation with primary key of each related entity

## PAPER\_COUNTS\_TOWARD\_COURSE

<u>Paper Code</u> (REFERENCES Paper)	<u>Course Name</u> (REFERENCES Course)
---	---

## POSSIBLE\_MAJOR\_FOR(Department M:N Course)

- Create new Relation with primary key of each related entity

## DEPARTMENT\_OFFERS\_MAJOR\_FOR\_COURSE

<u>Department Name</u> (REFERENCES Department)	<u>Course Name</u> (REFERENCES Course)
---	---

## TEACHES (Staff N:M Paper)

- Relationship refers to the keys of both staff and paper.

## TEACHES

<u>Teaching_staff (Staff ID)</u>	<u>Paper (Paper_code)</u>
----------------------------------	---------------------------

## OFFERED\_AT (Paper N:M Campus)

- Relationship refers to both paper and campus.

## OFFERED\_AT

<u>Paper (Paper Code)</u>	<u>Campus Name</u>
---------------------------	--------------------

## TAKES (Student N:M Paper)

- Relationship refers to student and paper.

## STUDENT\_TAKES\_PAPER

<u>Student ID</u>	<u>Paper_code</u>
-------------------	-------------------

## Step 6: Mapping of Multi-valued attributes

## PAPER\_SEMESTERS

<u>Paper_code</u>	<u>Semester</u>
-------------------	-----------------

PAPER

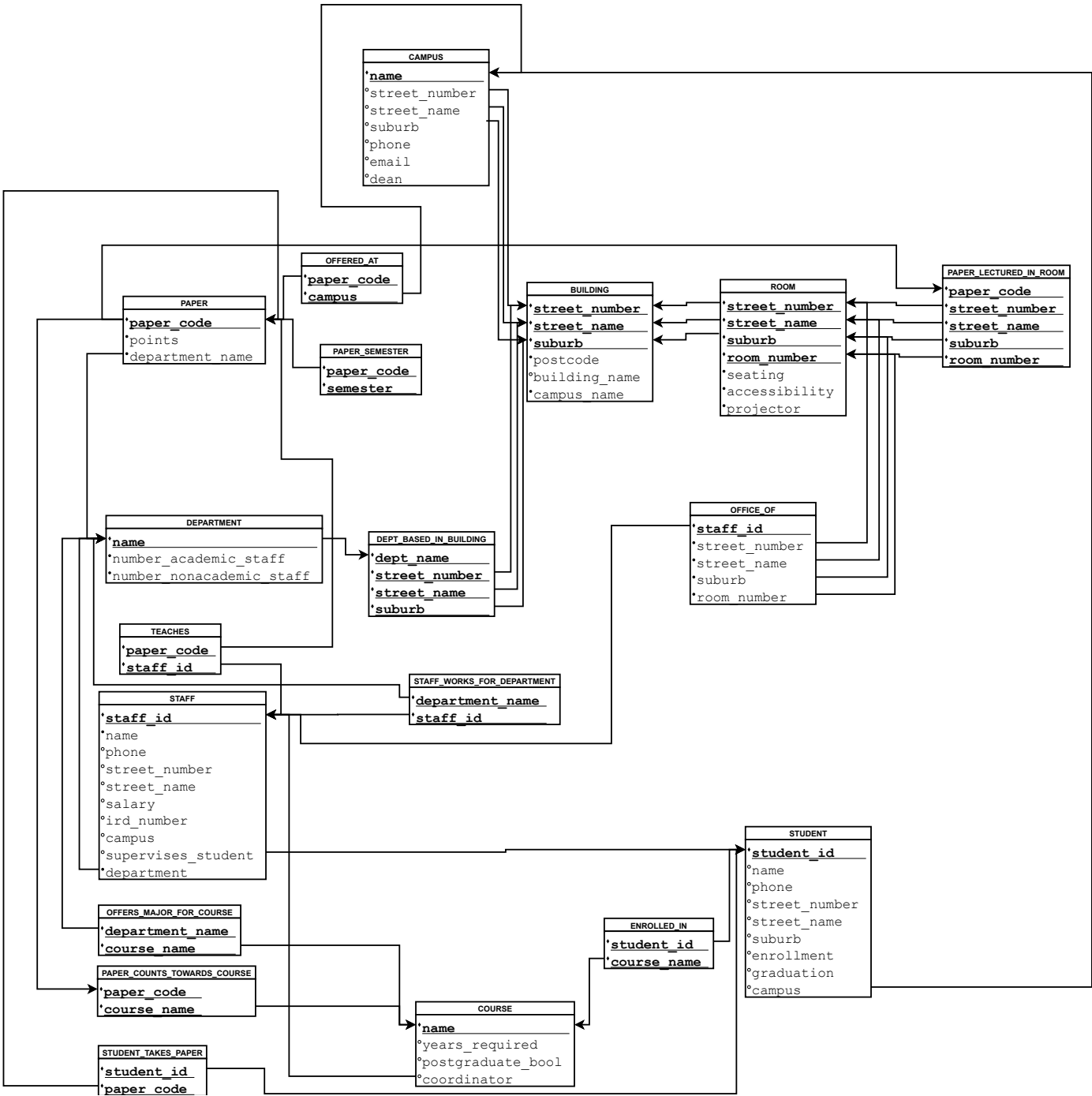
Paper code Points

- Semesters was previously a multi-valued attribute of paper, as a paper could be taught in multiple semesters. The table was split so one represents a paper and all the semesters it is taught in, and another refers to the paper code and the points associated with it.

Step 7: Mapping of N-ary Relationship types

- We had no N-ary relationship types to map

Mapped Relational Model



BUILDING

Street Number	Street Name	Suburb	Postcode	Building Name	Campus Name (REFERENCES Campus)
---------------	-------------	--------	----------	---------------	------------------------------------

## ROOM

<u>Street Number</u> (REFERENCES Building)	<u>Street Name</u> (REFERENCES Building)	<u>Suburb</u> (REFERENCES Building)	<u>Room Number</u>	Seating	Accessibility	Projector
---	---	--	--------------------	---------	---------------	-----------

## DEPARTMENT

<u>Name</u>	<u>Number_Of_Academic_Staff</u>	<u>Number_Of_Nonacademic_Staff</u>
-------------	---------------------------------	------------------------------------

## DEPT\_BASED\_IN\_BUILDING

<u>Dept Name</u> (REFERENCES Department)	<u>Street Number</u> (REFERENCES Building)	<u>Street Name</u> (REFERENCES Building)	<u>Suburb</u> (REFERENCES Building)
---	---	---	--

## CAMPUS

<u>Name</u>	<u>Street Number</u>	<u>Street Name</u>	<u>Suburb</u>	<u>Phone</u>	<u>Email</u>
-------------	----------------------	--------------------	---------------	--------------	--------------

## COURSE

<u>Name</u>	<u>Years_Required</u>	<u>Postgraduate_Bool</u>	<u>Coordinator_id</u> (REFERENCES Staff)
-------------	-----------------------	--------------------------	--

## DEPARTMENT\_OFFERS\_MAJOR\_FOR\_COURSE

<u>Department Name</u> (REFERENCES Department)	<u>Course Name</u> (REFERENCES Course)
---	---

## STUDENT

<u>Student ID</u>	<u>Name</u>	<u>Phone</u>	<u>Street Number</u>	<u>Street Name</u>	<u>Suburb</u>	<u>Enrollment</u>	<u>Graduation</u>	<u>Graduated</u>	<u>Campus</u>
-------------------	-------------	--------------	----------------------	--------------------	---------------	-------------------	-------------------	------------------	---------------

## ENROLLED\_IN

<u>Student ID</u>	<u>Course</u>
-------------------	---------------

## STAFF

<u>Staff ID</u>	<u>Name</u>	<u>Phone</u>	<u>Street Number</u>	<u>Street Name</u>	<u>Suburb</u>	<u>Salary</u>	<u>IRD_Num</u>	<u>Campus</u> (REFERENCES Campus)	<u>Supervises_Student</u> (REFERENCES Student)	<u>Department</u> (REFERENCES Department)
-----------------	-------------	--------------	----------------------	--------------------	---------------	---------------	----------------	--------------------------------------	---	--

## STAFF\_SUPERVISES\_STUDENT

<u>Staff ID</u> (REFERENCES Staff)	<u>Student ID</u> (REFERENCES Student)
------------------------------------	--

## PAPER

<u>Paper_code</u>	<u>Points</u>
-------------------	---------------

## PAPER\_SEMESTERS

<u>Paper_code</u>	<u>Semester</u>
-------------------	-----------------

## PAPER\_COUNTS\_TOWARD\_COURSE

<u>Paper Code</u> (REFERENCES Paper)	<u>Course Name</u> (REFERENCES Course)
---	---

## OFFERED\_AT

<u>Paper (Paper Code)</u> (REFERENCES Paper)	<u>Campus Name</u> (REFERENCES Campus)
---	---

## PAPER\_LECTURED\_IN\_ROOM

<u>Paper Code</u> (REFERENCES Paper)	<u>Street Number</u> (REFERENCES Room)	<u>Street Name</u> (REFERENCES Room)	<u>Suburb</u> (REFERENCES Room)	<u>Room Number</u> (REFERENCES Room)
---	---	---	------------------------------------	---

STUDENT\_TAKES\_PAPER

<u>Student ID</u> (REFERENCES Student)	<u>Paper code</u> (REFERENCES Paper)
---	---

TEACHES

<u>Teaching staff (Staff ID)</u>	<u>Paper (Paper code)</u>
----------------------------------	---------------------------

OFFICE\_OF

<u>Staff ID</u> (REFERENCES Staff)	<u>Street Number</u> (REFERENCES Building)	<u>Street Name</u> (REFERENCES Building)	<u>Suburb</u> (REFERENCES Building)	<u>Room Number</u>
---------------------------------------	---	---	--	--------------------

## Normalization

### 1NF

Definition: All values are atomic

- This is true from the mapping process from ERD to Relational Model
- The multi-value attribute of papers, semesters, is already converted to be atomic.

### 2NF

Definition: 1NF and every non-key attribute is fully dependent on the primary key

BUILDING

- Postcode is dependent only on street name and suburb, but not street number
- Remove postcode to its own entity, make a foreign key reference

BUILDING

<u>Street Number</u>	<u>Street Name</u> (REFERENCES postcode)	<u>Suburb</u> (REFERENCES postcode)	<u>Building Name</u>	<u>Campus Name</u> (REFERENCES Campus)
----------------------	---	--	----------------------	---

POSTCODE

<u>Street Name</u>	<u>Suburb</u>	<u>Postcode</u>
--------------------	---------------	-----------------

### 3NF

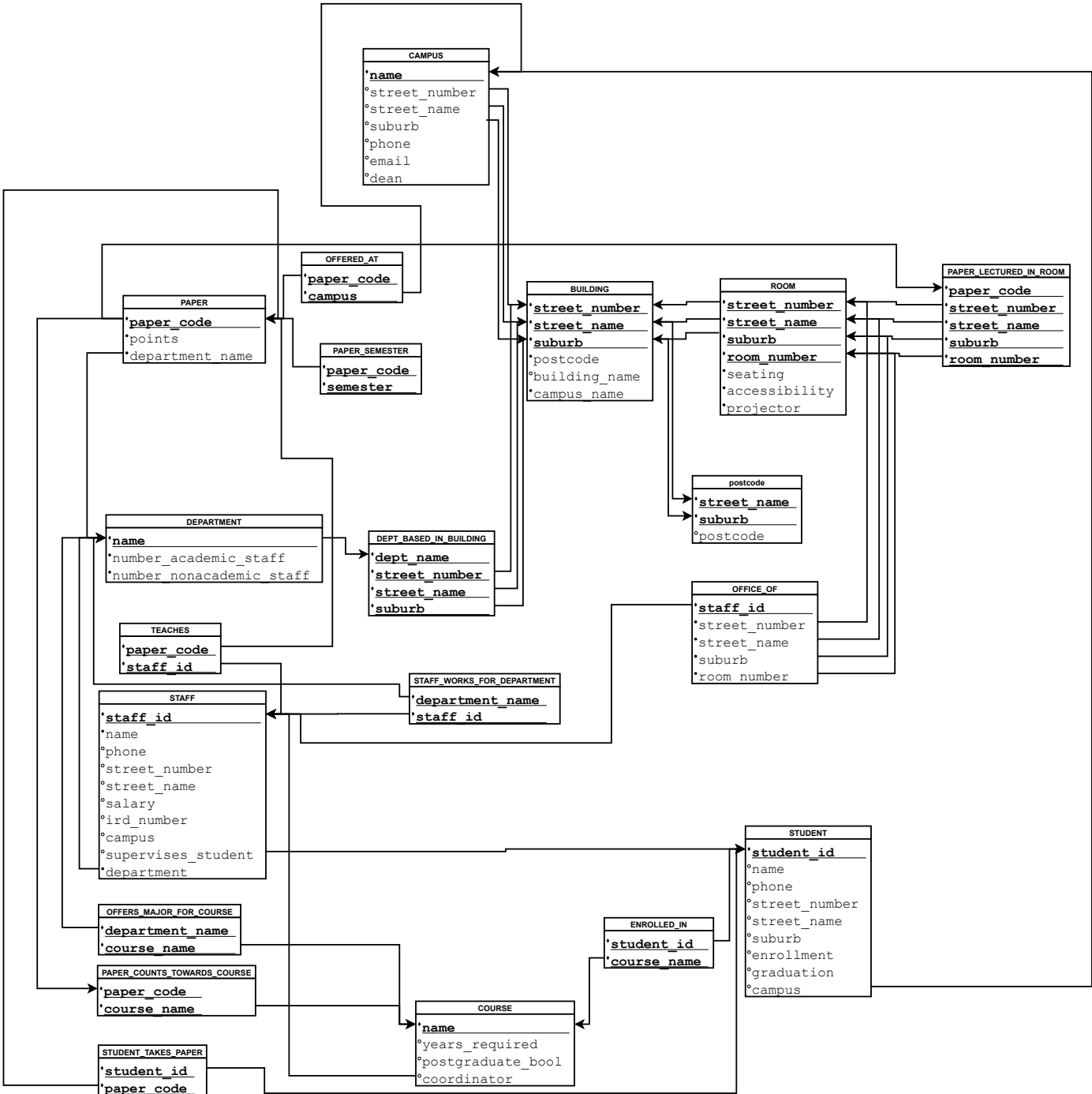
Definition: 2NF and no non-prime attribute is transitively dependent on the primary key

### BCNF

Definition: 3NF and for every non-trivial functional dependency  $X \rightarrow A$ ,  $X$  is a superkey of  $R$

## Normalized Relational Model





POSTCODE

<u>Street Name</u>	<u>Suburb</u>	Postcode
--------------------	---------------	----------

BUILDING

<u>Street Number</u>	<u>Street Name</u> (REFERENCES postcode)	<u>Suburb</u> (REFERENCES postcode)	Building_Name	Campus_Name (REFERENCES Campus)
----------------------	---	--	---------------	------------------------------------

ROOM

<u>Street Number</u> (REFERENCES Building)	<u>Street Name</u> (REFERENCES Building)	<u>Suburb</u> (REFERENCES Building)	<u>Room Number</u>	Seating	Accessibility	Projector
---	---	--	--------------------	---------	---------------	-----------

DEPARTMENT

<u>Name</u>	Number_Of_Academic_Staff	Number_Of_Nonacademic_Staff
-------------	--------------------------	-----------------------------

CAMPUS

<u>Name</u>	<u>Main_Office_Address</u>	<u>Phone</u>	<u>Email</u>	<u>Dean (REFERENCES Staff)</u>
-------------	----------------------------	--------------	--------------	--------------------------------

COURSE

<u>Name</u>	<u>Years_Required</u>	<u>Undergraduate</u>	<u>Postgraduate</u>	<u>Coordinator (REFERENCES Staff)</u>
-------------	-----------------------	----------------------	---------------------	---------------------------------------

STUDENT

<u>Student ID</u>	<u>Name</u>	<u>Phone</u>	<u>Street_Number</u>	<u>Street_Name</u>	<u>Suburb</u>	<u>Enrollment</u>	<u>Graduation</u>	<u>Graduated</u>	<u>Campus</u>
-------------------	-------------	--------------	----------------------	--------------------	---------------	-------------------	-------------------	------------------	---------------

ENROLLED\_IN

<u>Student ID</u>	<u>Course</u>
-------------------	---------------

STAFF

<u>Staff ID</u>	<u>Name</u>	<u>Phone</u>	<u>Street_Number</u>	<u>Street_Name</u>	<u>Suburb</u>	<u>Salary</u>	<u>IRD_Num</u>	<u>Campus (REFERENCES Campus)</u>	<u>Supervises_Student (REFERENCES Student)</u>	<u>Department (REFERENCES Department)</u>
-----------------	-------------	--------------	----------------------	--------------------	---------------	---------------	----------------	---	--	---

STAFF\_SUPERVISES\_STUDENT

<u>Staff ID (REFERENCES Staff)</u>	<u>Student ID (REFERENCES Student)</u>
------------------------------------	--

DEPT\_BASED\_IN\_BUILDING

<u>Dept Name (REFERENCES Department)</u>	<u>Street Number (REFERENCES Building)</u>	<u>Street Name (REFERENCES Building)</u>	<u>Suburb (REFERENCES Building)</u>
--	--	--	---

DEPARTMENT\_OFFERS\_MAJOR\_FOR\_COURSE

<u>Department Name (REFERENCES Department)</u>	<u>Course Name (REFERENCES Course)</u>
--	--

PAPER

<u>Paper_code</u>	<u>Points</u>
-------------------	---------------

PAPER\_SEMESTERS

<u>Paper_code</u>	<u>Semester</u>
-------------------	-----------------

PAPER\_COUNTS\_TOWARD\_COURSE

<u>Paper Code (REFERENCES Paper)</u>	<u>Course Name (REFERENCES Course)</u>
--	--

OFFERED\_AT

<u>Paper (Paper Code) (REFERENCES Paper)</u>	<u>Campus Name (REFERENCES Campus)</u>
--	--

PAPER\_LECTURED\_IN\_ROOM

<u>Paper Code (REFERENCES Paper)</u>	<u>Street Number (REFERENCES Room)</u>	<u>Street Name (REFERENCES Room)</u>	<u>Suburb (REFERENCES Room)</u>	<u>Room Number (REFERENCES Room)</u>
--	--	--	-------------------------------------	--

STUDENT\_TAKES\_PAPER

<u>Student ID (REFERENCES Student)</u>	<u>Paper code (REFERENCES Paper)</u>
--	--

TEACHES

<u>Teaching_staff (Staff ID)</u>	<u>Paper (Paper code)</u>
----------------------------------	---------------------------

OFFICE\_OF

<u>Staff ID</u>	<u>Street Number</u>	<u>Street Name</u>	<u>Suburb</u>	<u>Room Number</u>
(REFERENCES Staff)	(REFERENCES Building)	(REFERENCES Building)	(REFERENCES Building)	

TEAMWORK SUMMARY

- Hayden McAlister
  - Created template for report format
  - Mapped entities and relationships related to Building, Room
  - Created SQL for tables originating from Building, Room
  - Compiled teams report+SQL fragments into single documents with similar styles
  - Tested combined SQL
- Nat Moore
  - Mapped entities and relationships related to Department, Course
  - Created SQL for tables originating from Department, Course
- Jack Heikell
  - Mapped entities and relationships related to Student, Staff
  - Created SQL for tables originating from Student, Staff
- Masaaki Fukushima
  - Mapped entities and relationships related to Paper, Campus
  - Created SQL for tables originating from Paper, Campus