# *Database Management I (420-D10-HR)*

# *Lab 10 – Logical To Physical model*

Date assigned: Monday, October 17, 2016

Date due: **Monday, October 17, 2016, 3:50pm**

**Learning Objectives**

Upon successful completion of this lab exercise, the student will be able to:

* Convert logical ERDs to relational tables as part of transitioning to the implementation phase of Database design

**Marks**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Question** | **Mark** | **Out Of** |
| A | 1:N Model |  |  |
|  | Relationship and Keys |  | 8 |
|  | Table and Column definitions |  | 6 |
| B | Associative Entity Model |  |  |
|  | Relationship and keys |  | 12 |
|  | Data tables |  |  |
|  | * Student |  | 4 |
|  | * Course Table |  | 3 |
|  | * Course-Taken Table |  | 7 |
| C | Super/sub-type model - disjoint |  |  |
|  | * Relationships and keys |  | 16 |
|  | * Separate tables - Table details |  | 8 |
|  | * Merged tables- Table details |  | 4 |
| D | Super/sub-type model - overlapping |  |  |
|  | * Relationships and keys |  | 6 |
|  | * Separate tables - Table details |  | 6 |
| E | Research and definitions |  |  |
|  | * 11 definitions and examples |  | 22 |
|  |  |  |  |
|  | Organization |  | 5 |
|  | **Total** |  | **107** |

***To be handed in:***

1. The ***username*\_D10\_L10** documentcontaining the completed lab 10 answers should be submitted to Moodle.

***To Start:***

1. Read the class notes: “N05 Physical Database Design”. (I assume you’ve already read N04 by now). It can be found in Moodle.
2. Rename this lab document to ***username*\_D10\_L10** and insert your answers as indicated.

1. **1:N Model**
   1. Logical Diagram



* 1. Relationships and Keys

| **Table** | **Primary Key** | **Foreign Keys** | **Tables Referenced** |
| --- | --- | --- | --- |
| **Professor** | **Professor\_Number** | **Department\_Number** | **Department** |
| **Department** | **Department\_Number** | **None** | **None** |

* 1. Table/Column definitions

List **all** the columns of each Table

*add rows if required*

* + 1. Professor

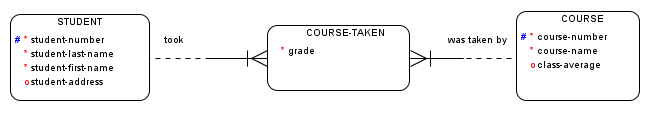
| **Column Name** | **Primary Key (Y/N)** | **Foreign Key (Y/N)** |
| --- | --- | --- |
| **Professor\_number** | **Yes** | **No** |
| **Professor\_name** | **No** | **No** |
| **Office\_phone\_number** | **No** | **No** |
| **Department\_Number** | **No** | **No** |

* + 1. Department table

*Add rows as required*

| **Column Name** | **Primary Key (Y/N)** | **Foreign Keys (Y/N)** | **General Attribute (Y/N)** |
| --- | --- | --- | --- |
| **Department\_Number** | **Yes** | **Yes** | **No** |
| **Department\_name** | **No** | **No** | **No** |

1. **Associative entities** 
   1. Logical Diagram



* 1. Relationships and Keys

| **Table** | **Primary Key** | **Foreign Keys** | **Tables Referenced** |
| --- | --- | --- | --- |
| **Student** | **Student\_Number** | **None** | **None** |
| **Course\_taken** | **No** | **None** | **Course** |
| **Course** | **Course\_Number** | **None** | **None** |

* 1. Data tables

Create your tables with all the defined column and rows based on the following data:

| **Student Last Name** | **Student First Name** | **Student Address** | **Student Grade** | **Course Number** | **Course Name** | **Class Average** |
| --- | --- | --- | --- | --- | --- | --- |
| **Pendous** | **Stu** | **101 Genius Blvd** | **100** | **401** | **Advanced Algebra** | **75** |
| **Pendous** | **Stu** | **101 Genius Blvd** | **100** | **407** | **Calculus** | **71** |
| **Yeungguy** | **Sum** | **12 Sommerset** | **85** | **401** | **Advanced Algebra** | **75** |
| **Yeungguy** | **Sum** | **12 Sommerset** | **76** | **407** | **Calculus** | **71** |
| **Skywalker** | **Luke** | **45 Dagoba** | **98** | **201** | **Intro to Light Sabre** | **75** |
| **Skywalker** | **Leia** | **98 Hoth** | **76** | **201** | **Intro to Light Sabre** | **75** |
| **Yeungguy** | **Sum** | **12 Sommerset** | **65** | **201** | **Intro to Light Sabre** | **75** |
| **Robot** | **Elliot** | **58 Second Street** | **100** | **410** | **Advanced Computer Hacking** | **84** |

Student Table:

*(Define/add columns and rows as required)*

| **Column 1** | **Column 2** |  |  |
| --- | --- | --- | --- |
| **Student\_number** | **Student\_first\_name** | **Student\_last\_name** | **Student\_address** |

Course Table:

*(define/add columns and rows as required)*

| **Column 1** | **Column 2** |  |  |
| --- | --- | --- | --- |
| **Course\_number** | **Course\_name** | **Class\_average** | **Semester** |

Course-Taken Table:

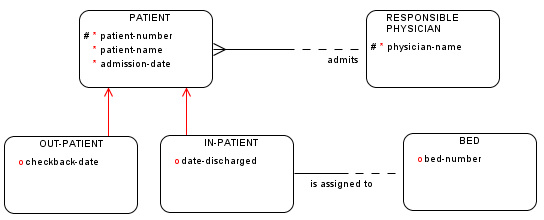
*(define/add columns and rows as required)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Student\_number** | **Course\_number** | **Student\_grade** | **Semester** |

1. **Supertypes and subtypes – Disjoint sub-types**
   1. Logical Diagram

Assume a Patient can only be an in-patient or an out-patient, but not both

Not shown in diagram, but add the following attributes to BED(location, last-replacement-date, bed-size)



* 1. Implement all types as separate tables
     1. Relationships and Keys

| **Table** | **Primary Key** | **Foreign Keys** | **Tables Referenced** |
| --- | --- | --- | --- |
| **Patient** | **Patient\_number** | **Patient\_number** | **None** |
| **Responsible physician** | **Physician\_id** |  | **Patient** |
| **Out-patient** | **Patient\_number** | **Patient\_number** | **Patient** |
| **In-patient** | **Patient\_number** | **Patient\_number** | **Bed** |
| **Bed** | **Bed\_number** |  | **In-patient** |

* + 1. List all tables, columns in text form

*i.e. TABLENAME(column1, column2, column3,….)*

Responsible physician (physician\_id, *physician\_name*)

Patient (*patient\_number, patient\_name, admission\_date, physician\_id*)

Out-patient (patient\_number, *checkback\_date*)

In-patient (patient\_number, *date\_discharged*)

Bed (*bed\_number, patient\_number, location, last\_replacement\_date, bed\_size*)

* 1. Merge sub-types to super type
     1. Relationships and Keys

| **Table** | **Primary Key** | **Foreign Keys** | **Tables Referenced** |
| --- | --- | --- | --- |
| **Patient** | **Patient\_number** | **Patient\_number** | **Bed, physician\_id** |

* + 1. List all tables, columns in text form

*i.e. TABLENAME(column1, column2, column3,….)*

*Responsible physician (physician\_name)*

*Patient (patient\_number, patient\_name, admission\_date, discharge\_date, checkback\_date, physician\_name)*

*Bed(*

1. **Supertypes and subtypes – overlapping sub-types**
   1. Logical Model

Assume any part can be manufactured AND purchased

Incomplete model: (not all physical columns shown)

PART( part-number, description, location, quantity-on-hand)

Sub types:

MANUFACTURED-PART( routing-number)

PURCHASED-PART(shipping-code)

* 1. Implement all types as separate tables
     1. Relationships and Keys

| **Table** | **Primary Key** | **Foreign Keys** | **Tables Referenced** |
| --- | --- | --- | --- |
| **PART** | **Part-number** |  |  |
| **MANUFACTURED-PART** | **Part-number** |  |  |
| **PURCHASED-PART** | **Part-number** |  |  |

* + 1. List all tables, all columns in text form

*i.e. TABLENAME(column1, column2, column3,….)*

Part (part\_number, description, location, quantity\_on\_hand)

Manufactured-part(part\_number, routing\_number)

Purchased-part(part\_number, shipping\_code)

1. **Research and Definitions**

Review the class notes (N04, N05 in Moodle) and answer the following questions

Provide a definition and an example for each of the following terms.

Note: Rephrase in your own words. Cut and Paste from the notes will result in a 0 mark.

| **Term** | **Definition** | **Example** |
| --- | --- | --- |
| **Candidate Key** | **A candidate key is one that is in consideration to become the primary key** | **SIN and employee numbers are candidates for the employee entity** |
| **Key** | **One or many attributes that uniquely define a row in a table** | **Student\_number** |
| **Intelligent key** | **A key that is derived from pre-existing information unique to entity** | **A serial-number in a store’s database** |
| **Composite attribute** | **Consists of one or more attributes that isn’t really an attribute, but acts as one.** | **Address is a composite key of a bunch of information** |
| **Derived attribute** | **An attribute that is found through the output of some sort of calculation performed.** | **Age is based on the number of years since your date of birth.** |
| **Constraint rule** | **Something that prevents an attribute from containing certain values.** | **A date must be valid, you can’t have November 31st as a date or February 30th** |
| **Normalization** | **A technique used to maintain database integrity by reducing the amount of redundant data and inconsistencies** | **Getting rid of repetitive columns inside a table** |
| **Entity integrity** | **An entity that doesn’t contain duplicate data for the same person or multiple entries with different Ids and stuff** | **The same student listed twice in active directory with two different student numbers** |
| **Foreign key** | **A key used by other tables to access a row inside that table** | **Employee\_id could be a foreign key for finding another employees manager** |
| **Composite Primary key** | **A primary key in an associative entity that is composed of the primary keys from multiple other tables** | **Student\_number and course\_number** |
| **Weak entity** | **An associative entity that links two tables together resolving many to many relationships.** | **A student can have multiple courses and a course contains multiple students, which prevents you from accessing data from either table. Enter, student\_course** |