# IT102

Normalization: 2NF, 3NF

2nd Normal Form

- A table is in 2NF if

  - ◆ It is in 1NF and

  - ◆ It includes no partial dependencies; that is no attribute is dependent on only a portion of the primary key.

- Since a partial dependency can exist only if a table's primary key is composed of several attributes, a table whose primary key consists of only a single attribute must automatically be in 2NF if it is in 1NF.

- Review the database design below and discuss which table is not in 2NF and why.
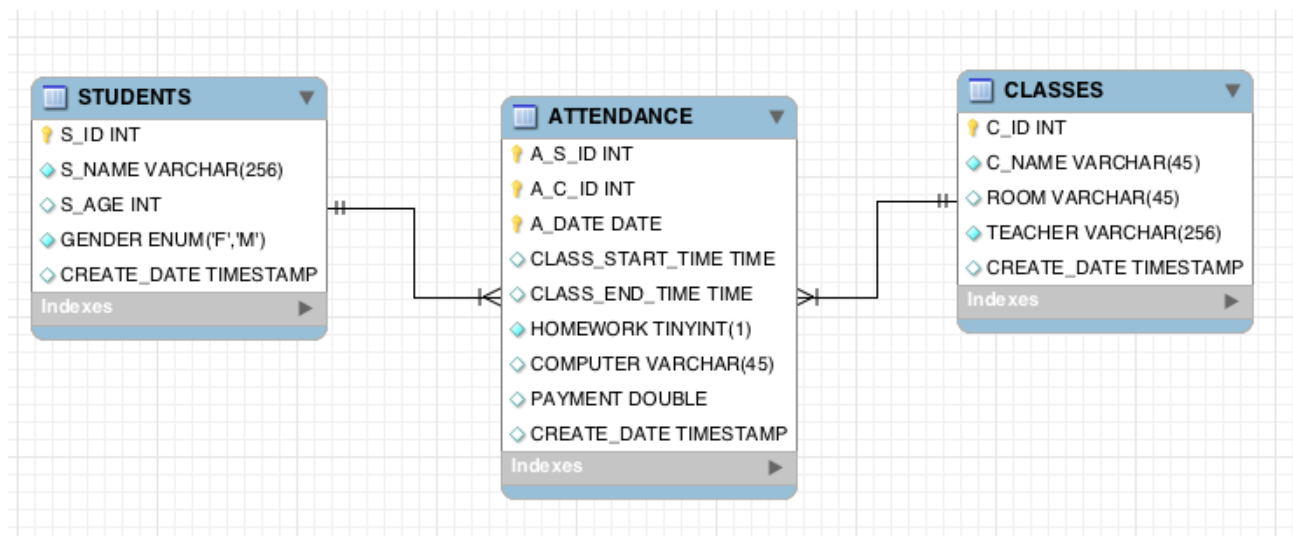
3rd Normal Form

- A table is in 3NF if

  ◆ It is in 2NF and

  ◆ It contains no transitive dependen-cies.

- Transitive dependency means that if we have a primary key A and a non-key attributes B and C where C is more dependent on B than A, and B is directly dependent on A, then C can be considered transitively dependent on A.

- Another way to look at it is a bit like a stepping stone across a river. If we consider the primary key A to be the far bank of the river and our non-key domain C to be our current location, in order to get to A, our primary key, we need to step on a stepping stone B, another non-key domain, to help us get there.

- Review the database design below and discuss which table is not in 3NF and why.
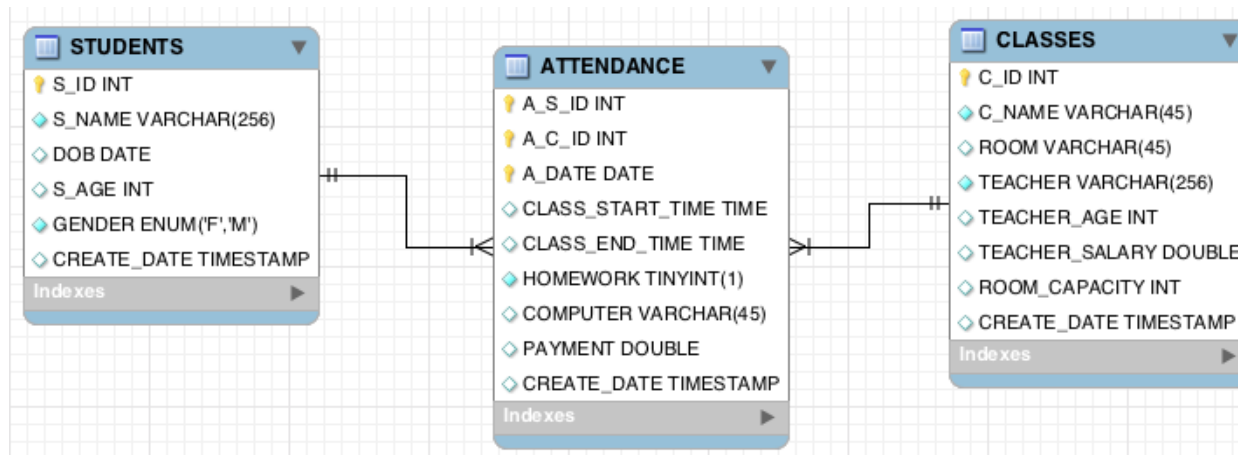
DB Design Best Practices

- Adhere to naming conventions: case, prefixes, word separators (no spaces), singular or plural names;

- Pay attention to automicity: use attributes that cannot be further subdivided.

- Each table should have a primary key, typically an automatically incremented integer type.

- Design for security: don't give everybody admin access.

- Avoid "SELECT *" statements, use "SELECT <column names> instead.

- Normalization must be used as required, to optimize the performance. Under-normalization will cause excessive repetition of data, over-normalization will cause excessive joins across too many tables. Both of them will result in inferior performance.

- Document your database design with ER schemas and instructions.

Homework

- StonyBrook Hospital has hired you to create a database for managing patients billing.

- Review the patient's bill sample.

- Create a database design (on paper) that includes tables, relationships, columns, primary keys and foreign keys, or reuse your HW from last week.

- Normalize your database design to 3NF.

- Create the tables in MySQL based on your design.

- Insert a couple of sample records into your billing DB.

Patient bill						
Patient #: 12345			Date: 7/20/08			
Patient Name: Mary Baker			Date admitted: 7/14/08			
Patient Address: 300 Oak Street			Discharge date: 7/17/08			
City-State-Zip: Boulder, CO 80638						
Cost Center	Cost Name	Date Charged	Item Code	Desc	Charge	Bal Due
100	Room & Board	7/14/08	2000	Semi-prv room	200.00	
		7/14/08	2005	Television	5.00	
		7/15/08	2000	Semi-prv room	200.00	
		7/16/08	2000	semi-prv room	200.00	
				Subtotal		605.00
110	Laboratory	7/14/08	1580	Glucose	25.00	
		7/15/08	1585	Culture	20.00	
				Subtotal		45.00
125	Radiology	7/15/08	3010	X-ray chest	30.00	
				Subtotal		30.00
				Balance due		680.00